

***NETWORK MONITORING SYSTEM* BERBASIS WEB
DENGAN SISTEM PERINGATAN DINI DAN *MAPPING*
JARINGAN DI STMIK INDONESIA MANDIRI**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk kelulusan
Jenjang Strata Satu (S1)
Pada Program Studi Teknik Informatika**

Oleh :

UJI HAMDANI
362141018



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA &
KOMPUTER INDONESIA MANDIRI
BANDUNG
2023**

LEMBAR PENGESAHAN

***NETWORK MONITORING SYSTEM* BERBASIS
WEB DENGAN SISTEM PERINGATAN DINI DAN
MAPPING JARINGAN DI STMIK INDONESIA
MANDIRI**

Oleh

Uji Hamdani
362141018

Tugas Akhir ini telah diterima dan disahkan untuk memenuhi persyaratan
mencapai gelar

STRATA SATU TEKNIK INFORMATIKA

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI**

Bandung, 26 Juni 2023
Disahkan oleh

Ketua Program Studi,



Asto Purwanto, S.T., M.Kom
NIDN : 0404078903

Dosen Pembimbing,



Patah Herwanto, S.T., M.Kom
NIDN : 0426076901

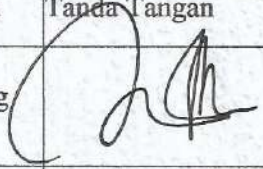


LEMBAR PERSETUJUAN REVISI

NETWORK MONITORING SYSTEM BERBASIS WEB DENGAN SISTEM PERINGATAN DINI DAN MAPPING JARINGAN DI STMIK INDONESIA MANDIRI

Oleh
Uji Hamdani
362141018

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan masukan pada saat sidang tugas akhir

Bandung,
Menyetujui,

No	Nama	Keterangan	Tanda Tangan
1	Patah Herwanto, S.T., M.Kom.	Pembimbing	
2	Dr. Chairuddin, M.M., MT.	Penguji 1	
3	Yudhi W. Arthana R., S.T., M.Kom.	Penguji 2	

Mengetahui,

Ketua Program Studi Teknik Informatika,



Asto Purwanto, S.T., M.Kom
NIDN : 0404078903

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa :

- (1) Naskah skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di sekolah tinggi manajemen informatika dan komputer indonesia mandiri maupun perguruan tinggi lainnya.
- (2) Naskah ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ini ada bantuan atau arahan dari pihak lain maka telah disebutkan identitas dan jenis bantuannya didalam lembar ucapan terimakasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya seni saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak lain.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 10 Juni 2023
Yang membuat pernyataan



Uji Hamdani
362141018

ABSTRAK

***NETWORK MONITORING SYSTEM* BERBASIS WEB DENGAN SISTEM PERINGATAN DINI DAN *MAPPING* JARINGAN DI STMIK INDONESIA MANDIRI**

Oleh

Uji Hamdani
NIM 362141018

Sistem monitor jaringan adalah salah satu bagian terpenting dalam sebuah jaringan, dimana pemantauan jaringan menjadi suatu hal yang sangat menentukan dalam beroperasinya kerja dari elemen jaringan seperti *switch*, *router*, *server*, *transmission* dan sebagainya. Fungsi pemantauan jaringan adalah sebagai peringatan dini jika terjadi kendala operasional dan konektifitas dari peralatan dan sumber daya yang ada dalam jaringan. Sistem monitor jaringan juga bisa berfungsi sebagai sumber laporan dari aktifitas jaringan dimana data tersebut bisa digunakan untuk perencanaan peningkatan dan perubahan pada elemen jaringan. Sistem monitor jaringan berbasis web ini dibangun dengan tujuan untuk memudahkan proses pemecahan masalah yang terjadi diarea kampus Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri. Hasil dari Sistem monitor jaringan ini berbentuk grafik pada dashboard aplikasi dan notifikasi yang dikirimkan ke telegram tim support dan administrator jaringan. Dengan adanya Sistem monitor jaringan ini dapat meningkatkan layanan jaringan di lingkungan kampus Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

kata kunci : STMIK IM, Sistem Monitor Jaringan, NMS, Web, Zabbix

ABSTRACT

WEB-BASED NETWORK MONITORING SYSTEM WITH EARLY WARNING SYSTEM AND NETWORK MAPPING AT STMIK INDONESIA MANDIRI

By

Uji Hamdani
NIM 362141018

Network monitoring system (NMS) is one of the essential part in a computer network, ensuring the proper functioning of network equipment such as switch, router, server, and transmission. It provides an early warning mechanism during an operational and connectivity problems arising from a network equipment. This system also functions as a network activity data logger where network improvement and changes can be based upon. A web-based NMS is designed to facilitate network troubleshooting at Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK IM) in form of application dashboard which can give notifications to network administrator and support team Telegram. With this web-based NMS network service in STMIK IM can be improved.

Key words: STMIK IM, Network Monitoring System, NMS, Web, Zabbix

KATA PENGANTAR

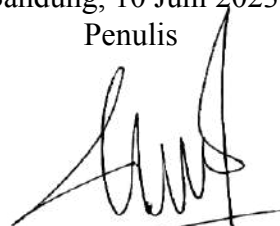
Bismillahirrahmaanirrahim, Puji dan Syukur penulis panjatkan kepada Allah Subhanahu Wa Ta'ala karena atas taufiq dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “*NETWORK MONITORING SYSTEM BERBASIS WEB DENGAN SISTEM PERINGATAN DINI DAN MAPPING JARINGAN DI STMIK INDONESIA MANDIRI*”.

Skripsi ini disusun sebagai syarat untuk menyelesaikan jenjang pendidikan Strata Satu (S1) di Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Penulis menyadari dalam penulisan laporan skripsi ini baik bentuk maupun isinya masih terdapat banyak kekurangan karena keterbatasan pengetahuan dan kempuan serta pengalaman yang penulis miliki. Untuk itu dengan kerendahan hati penulis mengharapkan kritik dan saran yang membangun untuk kemajuan penulis dikemudian hari. Penulis berharap skripsi ini bisa menjadi salah satu sumber yang dapat bermanfaat.

Bandung, 10 Juni 2023

Penulis



Uji Hamdani

362141018

UCAPAN TERIMAKASIH

Puji dan syukur penulis panjatkan kepada Allah SWT, karena atas limpahan rahmat, taufik, hidayah dan karunia-Nya tugas akhir ini dapat terselesaikan dengan baik. Tak lupa sholawat teriring salam semoga tercurah limpahkan kepada baginda Rasulullah Muhammad SAW beserta keluarga dan para sahabatnya, yang telah menjadi suri tauladan yang baik bagi umat muslim diseluruh dunia.

Penyusunan skripsi ini tidak lepas dari dukungan dan bimbingan dari berbagai pihak, maka pada kesempatan ini peneliti ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada :

1. Bapak Patah Herwanto, S.T., M.Kom. selaku Wakil Ketua Bid. Akademik sekaligus sebagai Dosen pembimbing yang selalu meluangkan waktu, pikiran dan tenaga dalam memberikan bimbingan, masukan dan saran-sarannya.
2. Ibu Asto Purwanto, S.T., M.Kom. sebagai ketua Program Studi Teknik Informatika di STMIK IM Bandung.
3. Bapak dan Ibu Dosen, staff dan karyawan Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri yang telah mendidik dan membantu dalam memberikan informasi seta motivasi dalam proses studi maupun tugas akhir berlangsung.
4. Teruntuk kedua atasan penulis di PT Integra Kreasitama Solusindo Bapak Eki Gandana dan Bapak Ahmad Farisy yang selalu mendorong dan memberikan motivasi untuk menyelesaikan skripsi ini.
5. Teruntuk Kedua Orang Tua Tercinta yang tiada hentinya memberikan do'a terbaik untuk penulis agar dilancarkan dalam proses menyelesaikan skripsi ini.

6. Teruntuk kedua Adikku Utari Hambawati dan Nur Annisa yang selalu memberikan motivasi kepada penulis.
7. Teruntuk Istri tercinta yang selalu memberikan do'a, motivasi yang sangat kuat serta selalu mengingatkan penulis dalam menyelesaikan skripsi ini.
8. Kedua anakku Affan dan Hafizh yang selalu dijadikan motivasi oleh penulis dalam menyelesaikan skripsi ini.
9. Serta masih banyak lagi pihak-pihak yang sangat berpengaruh dalam proses penyelesaian skripsi ini yang penulis tidak bisa sebutkan satu persatu.

Semoga Allah SWT senantiasa membalas semua kebaikan yang telah diberikan. Semoga penelitian ini dapat bermanfaat bagi orang yang membacanya.

DAFTAR ISI

SURAT PERNYATAAN.....	i
ABSTRAK.....	ii
ABSTRACT.....	iii
KATA PENGANTAR.....	iv
UCAPAN TERIMAKASIH.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Metode Penelitian.....	5
1.5.1 Metode Pengumpulan Data.....	5
1.5.2 Metode Pengembangan Sistem.....	6
1.6 Sistematika Penulisan.....	8
BAB 2 LANDASAN TEORI.....	10
2.1 Pengertian Sistem.....	10
2.1.1 Bentuk Dasar Sistem.....	10
2.1.2 Karakteristik Sistem.....	11
2.2 Pengertian <i>Network</i>	14
2.2.1 Berdasarkan Jangkauan Geografis.....	15
2.2.2 Berdasarkan Media Transmisi Data.....	16
2.2.3 Berdasarkan Peranan Komputer.....	16
2.2.4 Berdasarkan Topologi Jaringan.....	17
2.3 Pengertian Sistem Monitoring.....	20
2.4 Pengertian <i>Network Monitoring</i>	20
2.5 Pengertian <i>Network Monitoring System</i>	21
2.6 Tujuan <i>Network Monitoring System</i>	21
2.7 Pengertian <i>Internet</i>	22
2.8 Model Referensi OSI.....	22
2.9 Protokol Jaringan.....	24
2.10 <i>Internet Protocol (IP)</i>	24
2.10.1 Pengalamatan IP.....	25
2.10.2 Alamat IP (<i>IP Address</i>).....	25
2.10.3 Pembagian Kelas Alamat IP (<i>Class-based IP Address</i>).....	25
2.11 <i>Internet Control Message Protocol (ICMP)</i>	27
2.11.1 Pesan ICMP.....	27
2.11.2 PING.....	28
2.11.3 Traceroute.....	29
2.12 <i>Address Resolution Protocol (ARP)</i>	30
2.13 PORT.....	30
2.14 <i>User Datagram Protocol (UDP)</i>	31

2.15 Simple Network Management Protocol (SNMP).....	31
2.15.1 <i>Management Information Base (MIBS)</i>	32
2.15.2 <i>Arsitektur SNMP</i>	33
2.16 <i>Transmission Control Protocol (TCP)</i>	34
2.17 <i>Telnet</i>	35
2.18 <i>File Transfer Protocol (FTP)</i>	36
2.19 <i>Simple Mail Transfer Protocol (SMTP)</i>	36
2.20 <i>Hyper-Text Transfer Protocol (HTTP)</i>	36
2.21 <i>Secure Shell (SSH)</i>	37
2.22 <i>Pengertian Web</i>	38
2.22.1 <i>Jenis-Jenis Web</i>	38
2.22.2 <i>Jenis-Jenis Web berdasarkan sifat atau style</i>	39
2.22.3 <i>Berdasarkan dari segi bahasa pemrograman</i>	39
2.23 <i>Web Server</i>	40
2.24 <i>Apache</i>	41
2.25 <i>HyperText Markup Language (HTML)</i>	41
2.26 <i>Hypertext Preprocessor (PHP)</i>	42
2.27 <i>Javascript</i>	43
2.28 <i>Python</i>	44
2.29 <i>MySQL</i>	44
2.30 <i>Telegram</i>	45
2.31 <i>Zabbix</i>	45
2.32 <i>Pengertian Waterfall</i>	47
2.32.1 <i>Kelebihan Waterfall</i>	48
2.33 <i>Unified Modeling Language (UML)</i>	49
2.34 <i>Pengujian Sistem</i>	56
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	58
3.1 <i>Communication</i>	58
3.1.1 <i>Metode Pengumpulan Data</i>	58
3.1.2 <i>Analisis Sistem</i>	60
3.1.3 <i>Analisis Sistem Jaringan</i>	61
3.1.4 <i>Topologi Jaringan Saat Ini</i>	62
3.1.5 <i>Analisis Sistem Monitoring Yang Sedang Berjalan</i>	62
3.1.6 <i>Evaluasi Sistem Yang Berjalan</i>	64
3.1.7 <i>Topologi Jaringan Yang Diusulkan</i>	64
3.1.8 <i>Analisis Sistem Monitoring Yang Diusulkan</i>	65
3.1.9 <i>Analisa Kebutuhan Perangkat Server</i>	68
3.1.10 <i>Analisa Kebutuhan Untuk Perangkat Keras</i>	69
3.2 <i>Planning (Estimating, Scheduling, Tracking)</i>	69
3.3 <i>Modeling (Analysis and Design)</i>	71
3.3.1 <i>Perancangan Sistem</i>	71
3.3.2 <i>Tujuan Perancangan Sistem</i>	71
3.3.3 <i>Use Case Diagram</i>	72
3.3.4 <i>Spesifikasi Use Case</i>	73
3.3.5 <i>Activity Diagram</i>	76
3.3.6 <i>Class Diagram</i>	81
3.3.7 <i>Sequence Diagram</i>	82

3.3.8 Perancangan Database.....	85
3.3.9 Entity Relationship Diagram.....	92
3.3.10 Perancangan Antar Muka.....	94
BAB 4 IMPLEMENTASI DAN UJI COBA.....	98
4.1 Implementasi Aplikasi.....	98
4.1.1 Physical Database.....	98
4.1.2 Implementasi Perangkat Lunak.....	100
4.1.3 Implementasi Perangkat Keras.....	101
4.2 Implementasi Antarmuka.....	102
4.2.1 Implementasi Sistem Antarmuka.....	102
4.3 Pengujian Sistem.....	108
4.4 Pengujian NMS.....	119
4.4.1 Hasil Pengujian.....	126
BAB 5 PENUTUP.....	127
5.1 Kesimpulan.....	127
5.2 Saran.....	127
DAFTAR PUSTAKA.....	129
LAMPIRAN.....	132

DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i> (Aprianti and Maliha 2016).....	49
Tabel 2.2 Simbol <i>Activity Diagram</i> (Aprianti and Maliha 2016).....	50
Tabel 2.3 Simbol <i>Sequence Diagram</i> (Aprianti and Maliha 2016).....	52
Tabel 2.4 Simbol <i>Entity Relationship Diagram</i>	53
Tabel 3.1 Tabel Studi Literatur.....	58
Tabel 3.2 Penjadwalan Penelitian.....	70
Tabel 3.3 Definisi Aktor.....	73
Tabel 3.4 Spesifikasi Use Case Admin.....	74
Tabel 3.5 Spesifikasi Use Case Menambah Perangkat.....	74
Tabel 3.6 Spesifikasi Use Case Menambahkan Alert.....	74
Tabel 3.7 Spesifikasi Use Case Login Tim Support.....	75
Tabel 3.8 Spesifikasi Use Case Monitor Perangkat.....	75
Tabel 3.9 Spesifikasi Use Case Lihat Alert.....	75
Tabel 3.10 Tabel User.....	86
Tabel 3.11 Tabel User Groups.....	86
Tabel 3.12 Tabel Hosts.....	86
Tabel 3.13 Tabel Host Group.....	88
Tabel 3.14 Tabel Interface.....	88
Tabel 3.15 Tabel Event.....	88
Tabel 3.16 Tabel Graph.....	88
Tabel 3.17 Tabel Items.....	89
Tabel 3.18 Tabel Graph Items.....	91
Tabel 3.19 Tabel History.....	91
Tabel 3.20 Tabel Alert.....	91
Tabel 3.21 Tabel Triggers.....	92
Tabel 4.1 Hasil Pengujian Sistem Login.....	109
Tabel 4.2 Hasil Pengujian Sistem Data Perangkat.....	110
Tabel 4.3 Hasil Pengujian Sistem Alert.....	115
Tabel 4.4 Hasil Pengujian Sistem User.....	116
Tabel 4.5 Hasil Pengujian Sistem Pengguna.....	118
Tabel 4.6 Tabel Proses Fungsi Kerja NMS – Gangguan Access Point.....	120
Tabel 4.7 Tabel Proses Fungsi NMS – Delay akses ke arah Internet.....	122
Tabel 4.8 Tabel Proses Fungsi NMS – Info Penyimpanan Web Server.....	124
Tabel 4.9 Tabel Hasil Pengujian.....	126

DAFTAR GAMBAR

Gambar 1.1 Metode <i>Waterfall</i>	6
Gambar 2.1 Bentuk Dasar Sistem.....	11
Gambar 2.2 Karakteristik Sistem.....	14
Gambar 2.3 Topologi Bus.....	17
Gambar 2.4 Topologi Star.....	18
Gambar 2.5 Topologi Ring.....	19
Gambar 2.6 Topologi Mesh.....	19
Gambar 2.7 Proses Dalam <i>Monitoring</i>	20
Gambar 2.8 Referensi OSI.....	23
Gambar 2.9 Pembagian Kelas IP.....	26
Gambar 2.10 Contoh Hasil Perintah PING.....	29
Gambar 2.11 Contoh Hasil Perintah <i>Traceroute</i>	30
Gambar 2.12 Proses Kerja Telnet.....	35
Gambar 2.13 Proses Kerja HTTP.....	36
Gambar 2.14 Proses Kerja SSH.....	38
Gambar 2.15 Contoh syntax PHP.....	43
Gambar 2.16 Contoh syntax javascript.....	43
Gambar 2.17 Contoh syntax Python.....	44
Gambar 2.18 Contoh syntax MySQL.....	45
Gambar 3.1 Topologi STMIK Indonesia Mandiri Saat Ini.....	62
Gambar 3.2 Flow Map Metode Pengecekan Jaringan Saat Ini.....	63
Gambar 3.3 Topologi STMIK Indonesia Mandiri Yang Diusulkan.....	64
Gambar 3.4 Flow Map Sistem Yang Diusulkan.....	66
Gambar 3.5 <i>Use Case Diagram</i>	73
Gambar 3.6 Activity Diagram Login.....	76
Gambar 3.7 Activity Diagram Menambahkan Perangkat.....	77
Gambar 3.8 Activity Diagram Menambahkan Alert.....	78
Gambar 3.9 Activity Diagram Input Menambahkan Tim Support.....	79
Gambar 3.10 Activity Diagram Input Melihat Alert.....	80
Gambar 3.11 Activity Diagram Monitor Perangkat.....	81
Gambar 3.12 Class Diagram Network Monitoring System.....	82
Gambar 3.13 Sequence Diagram Login.....	83
Gambar 3.14 Sequence Diagram Menambahkan Perangkat.....	83
Gambar 3.15 Sequence Diagram Menambahkan Alert.....	84
Gambar 3.16 Sequence Diagram Menambahkan Tim Support.....	84
Gambar 3.17 Sequence Diagram Melihat Alert.....	85
Gambar 3.18 Sequence Diagram Monitor Perangkat.....	85
Gambar 3.19 Entity Relation Diagram NMS.....	93
Gambar 3.20 Perencanaan Tampilan Login NMS.....	94
Gambar 3.21 Perencanaan Tampilan Dashboard.....	95
Gambar 3.22 Perencanaan Tampilan Grafik.....	95
Gambar 3.23 Perencanaan Tampilan List User.....	96
Gambar 3.24 Perencanaan Tampilan Alert.....	96
Gambar 3.25 Perencanaan Tampilan Form Input User.....	97

Gambar 3.26 Perencanaan Tampilan Form Input Perangkat.....	97
Gambar 4.1 Physical Database.....	99
Gambar 4.2 Operating System CentOS 7.....	100
Gambar 4.3 Kernel Operating System.....	100
Gambar 4.4 Versi Web Server.....	100
Gambar 4.5 Versi PHP.....	101
Gambar 4.6 Versi Database Server.....	101
Gambar 4.7 CPU Server.....	101
Gambar 4.8 RAM Server.....	102
Gambar 4.9 Storage Server.....	102
Gambar 4.10 Implementasi Antarmuka Halaman Login.....	103
Gambar 4.11 Implementasi Antarmuka Halaman Dashboard.....	103
Gambar 4.12 Trafik Distribusi Bandwidth STMIK IM.....	104
Gambar 4.13 Peta Jaringan Kampus STMIK IM.....	104
Gambar 4.14 Monitoring Web Server STMIK IM.....	105
Gambar 4.15 Monitoring Alert Jaringan STMIK IM.....	105
Gambar 4.16 Implementasi Menambah Users.....	106
Gambar 4.17 Implementasi Melihat Tabel Users.....	106
Gambar 4.18 Implementasi Menambah Perangkat.....	107
Gambar 4.19 Implementasi Melihat Perangkat.....	107
Gambar 4.20 Implementasi Notifikasi Telegram.....	108

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia semakin maju dengan adanya perkembangan teknologi dan jaringan. Perkembangan tersebut memudahkan komunikasi antara satu dengan lainnya yang dibangun sesuai dengan kebutuhan institusi maupun kebutuhan pribadi. Dalam sebuah instansi, jaringan selalu dibutuhkan untuk memenuhi kebutuhan transfer data dan sinkronasi dari setiap perangkat yang dipakai. Terlebih lagi di tengah pandemi COVID-19 memberikan dampak bagi semua kalangan, ditambah lagi kebijakan yang diputuskan oleh pemerintah untuk menghibau segala kegiatan dilakukan dari rumah atau dengan istilah WFH (Work From Home).(Tegar and Afandi 2019).

Salah satu tugas manajemen yang sangat penting adalah memantau jaringan untuk mengetahui apakah jaringan masih memadai atau memerlukan peningkatan kapasitas. *Monitoring* juga dapat membantu administrator merancang ulang jaringan. Lalu lintas jaringan (*load traffic*) yang melewati *router* atau antarmuka komputer adalah salah satu dari banyak aspek jaringan yang dapat diamati. Monitoring dapat dilakukan melalui protokol SNMP standar. Kondisi jaringan, seperti status perangkat jaringan yang terlihat *up down* harus dipantau selain lalu lintas jaringan. Alat ping memungkinkan Anda melakukannya. (Rahmat et al. 2022).

Berkembangnya teknologi di Indonesia, didukung oleh kemajuan ilmu pengetahuan dan teknologi jaringan telekomunikasi, membuat pentingnya *monitoring* menjadi penting. *Monitoring* tidak hanya digunakan untuk menemukan masalah dan anomali dalam jaringan; analisis jaringan juga penting untuk pengembangan tim

teknik yang lebih baik. Untuk melacak perangkat dan layanan jaringan, diperlukan fasilitas pendukung seperti monitoring jaringan sistem. Sistem ini memungkinkan *administrator* jaringan untuk memantau perangkat dan layanan tersebut bahkan saat mereka tidak berada di komputer.(Fauzi and Desmulyati 2020). Dengan menerapkan sistem *monitoring* ini, diharapkan setiap masalah jaringan dapat dideteksi dengan cepat, yang memungkinkan penanganan lebih cepat dan efisien.(Kuswanto 2018).

Temuan penelitian ini diawali dengan adanya diskusi dengan tim yang bertanggung jawab menangani jaringan di kampus STMIK Indonesia Mandiri. Masalah-masalah yang sering terjadi cukup beragam seperti terputusnya koneksi kearah *internet*, perangkat yang bermasalah. Untuk proses perbaikan sedikit membutuhkan waktu, karena tidak ada alat yang bisa difungsikan untuk mendeteksi peringatan dini berupa *alert* dan *mapping* perangkat. Selain itu juga tidak ada laporan *performance* untuk penggunaan *bandwidth internet*, *resource* perangkat di kampus.

Karena dengan pentingnya NMS untuk manajemen disebuah jaringan saya selaku penulis dan mahasiswa bermaksud untuk membuat NMS yang bisa membantu untuk mengamati atau memonitor jaringan yang berjalan dimana nantinya NMS tersebut dapat mendeteksi jika terjadi permasalahan di kampus, sebagai contoh jika koneksi kearah internet mengalami *down* maka NMS akan menginformasikan pesan atau notifikasi kepada tim IT dikampus. Dalam kesempatan ini penulis akan membuat NMS untuk kebutuhan manajemen jaringan di kampus dengan judul “**NETWORK MONITORING SYSTEM BERBASIS WEB DENGAN SISTEM PERINGATAN DINI DAN MAPPING JARINGAN DI STMIK INDONESIA MANDIRI**”.

1.2 Identifikasi Masalah

Berdasarkan informasi yang didapatkan oleh peneliti, berikut adalah beberapa masalah yang sering dihadapi pada jaringan kampus saat ini :

1. Sulitnya mengidentifikasi permasalahan jaringan secara tepat.
2. Keterbatasan pemantauan jaringan secara *Real-Time* Sehingga menyebabkan tidak adanya notifikasi atau peringatan saat terjadi gangguan.
3. Tidak adanya pemetaan jaringan yang terintegrasi antar perangkat.
4. Tidak adanya laporan dan analisis yang komprehensif.
5. Tidak adanya acuan yang dijadikan standar untuk menjalankan aktifitas jaringan (Perbaikan, Pembaharuan dan Perawatan).

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk membuat sebuah sistem yang memberikan pengaruh positif dalam memonitor masalah pada jaringan dikampus diantaranya :

1. Pendeteksian masalah dengan cepat.
2. Monitoring jaringan secara *Real-Time*.
3. Pemetaan jaringan yang Komprehensif.
4. Manajemen yang Proaktif.
5. Dapat meningkatkan efisiensi operasional jaringan.

1.4 Batasan Masalah

Langkah-langkah yang diambil akan memastikan system terarah dan berhasil dalam upaya untuk memastikan penyusunan tugas akhir ini tetap fokus pada masalah

yang ada dan mencapai tujuannya dengan baik, maka dalam penelitian tentang *Network Monitoring System*, berikut adalah beberapa batasan masalah yang dapat diterapkan :

1. Lingkup penelitian : Penelitian ini akan difokuskan pada implementasi *Network Monitoring System* di kampus STMIK Indonesia Mandiri.
2. Fokus pada *Web-based System* : Penelitian ini akan berfokus pada *Network Monitoring System* berbasis *web*. Ini berarti sistem yang dikembangkan akan diakses melalui *web browser* dan memiliki antarmuka pengguna yang berbasis *web*.
3. Fitur Notifikasi : Penelitian ini akan memfokuskan pada pengembangan fitur peringatan dini atau notifikasi pada *dashboard* dan aplikasi chat *telegram* yang dapat menginformasikan tim IT tentang gangguan atau anomali yang terjadi pada jaringan.
4. Monitoring Jaringan: Penelitian ini akan difokuskan pada pemantauan jaringan yang meliputi identifikasi masalah, pemantauan kinerja, penggunaan bandwidth, dan resource perangkat jaringan.
5. Pemetaan Jaringan : Penelitian ini akan mencakup pengembangan fitur pemetaan jaringan, yang memungkinkan visualisasi topologi jaringan.
6. Penggunaan Alat dan Teknologi : Penelitian ini akan menggunakan alat, teknologi, atau protokol yang spesifik sesuai dengan kebutuhan dan ketersediaan di kampus STMIK Indonesia Mandiri.

1.5 Metode Penelitian

Dalam penelitian ini, pendekatan yang digunakan penulis untuk melakukan penelitian akan dijelaskan. Metode penelitian digunakan untuk menemukan masalah, mengumpulkan data tentang keadaan aktual, dan mencoba menemukan solusi.

1.5.1 Metode Pengumpulan Data

Metode pengumpulan data dilakukan dengan cara studi lapangan untuk mengumpulkan dan menelaah informasi mengenai sistem yang sedang berjalan pada jaringan kampus dengan metode observasi dan studi dokumentasi.

1. Observasi

Observasi adalah teknik pengumpulan data yang melibatkan banyak aspek yang kompleks dalam prosesnya. Metode observasi digunakan untuk mengukur sikap responden dan merekam berbagai fenomena. Penelitian yang mempelajari perilaku manusia, proses kerja, dan gejala alam cocok dengan teknik pengumpulan data observasi.

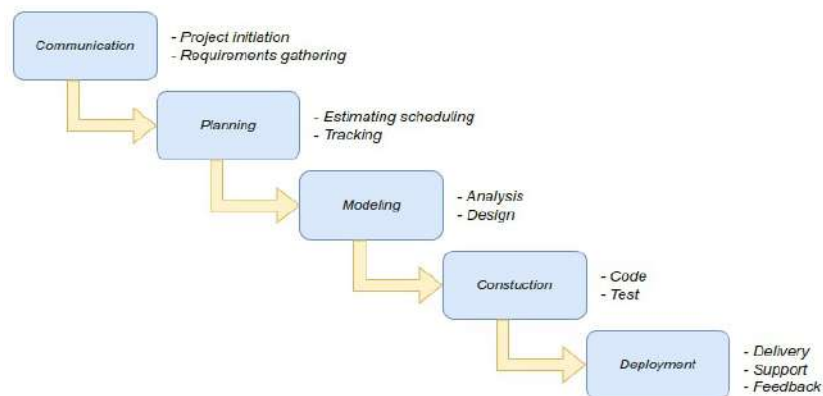
2. Studi Dokumen

Metode pengumpulan data yang dikenal sebagai studi dokumen tidak melibatkan berinteraksi langsung dengan subjek penelitian. Metode ini berpusat pada penelitian dan analisis berbagai jenis dokumen yang terkait dengan masalah tersebut. Dokumen ini digunakan sebagai sumber data penting untuk analisis dan diskusi.

1.5.2 Metode Pengembangan Sistem

Model *waterfall* adalah model pengembangan perangkat lunak yang paling populer. Model ini menggunakan pendekatan linear yang berurutan dari tahap perencanaan awal hingga tahap pemeliharaan akhir system.(Pricillia and Zulfachmi 2021).

Metode yang digunakan penulis dalam rancang bangun *Network Monitoring System* adalah menggunakan metode *waterfall*. Untuk membangun sistem pemantauan jaringan, metode *waterfall* sangat sesuai digunakan. Pendekatan sistematis dan berurutan dalam membangun system adalah alasan mengapa metode ini dipilih. Metode *waterfall* memerlukan sistem dibuat secara berurutan. Metode ini menghasilkan sistem yang berkualitas tinggi karena implementasinya dilakukan secara bertahap tanpa berkonsentrasi pada satu tahapan. Metode *waterfall* mencakup :



Gambar 1.1 Metode *Waterfall*

1. *Communication (Project Initiation & Requirement Gathering)*

Sebelum memulai pekerjaan teknis, sangat penting untuk berkomunikasi dengan pelanggan untuk memahami dan mencapai tujuan yang diinginkan.

Analisis masalah dan pengumpulan data yang diperlukan, serta bantuan dalam mendefinisikan fitur dan fungsi perangkat lunak, adalah hasil dari komunikasi yang mendorong pelaksanaan proyek. Jurnal, *article*, dan sumber informasi lainnya, seperti *internet*, dapat menawarkan data tambahan.(Risald 2021).

2. *Planning (Estimating, Scheduling, & Tracking)*

Perencanaan adalah tahap selanjutnya, di mana estimasi tugas teknis yang akan dilakukan, identifikasi risiko yang mungkin terjadi, penentuan sumber daya yang diperlukan untuk membangun sistem, definisi produk kerja yang ingin dicapai, penjadwalan pelaksanaan pekerjaan, dan pemantauan proses pengembangan sistem.(Risald 2021).

3. *Modelling (Analysis & Design)*

Fase perencanaan dan permodelan arsitektur sistem ini fokus pada desain struktur data, arsitektur perangkat lunak, antarmuka pengguna, dan algoritma program. Tujuannya adalah untuk mendapatkan pemahaman yang komprehensif tentang pekerjaan yang akan dilakukan secara keseluruhan. (Risald 2021).

4. *Construction (Code & Test)*

Mengubah desain menjadi bentuk atau kode yang dapat dipahami mesin adalah proses yang termasuk dalam tahapan pembangunan ini. Setelah proses pengkodean selesai, system dan kode diuji. Tujuannya adalah untuk menemukan dan mengidentifikasi kesalahan sehingga dapat diperbaiki. (Risald 2021).

5. *Deployment (Delivery, Support & Feedback)*

Implementasi perangkat lunak kepada pelanggan, pemeliharaan rutin, perbaikan, evaluasi, dan pengembangan berdasarkan umpan balik adalah semua bagian dari tahapan deployment. Tahap ini dilakukan dengan tujuan memastikan sistem tetap beroperasi dan berkembang sesuai dengan fungsinya. (Risald 2021).

1.6 Sistematika Penulisan

Penulisan Tugas Akhir dibagi menjadi lima bab untuk memfasilitasi pemahaman dan diskusi. Bab-bab ini mencakup :

BAB I PENDAHULUAN

Bab ini membahas latar belakang, masalah yang dihadapi, tujuan penelitian, rumusan, batasan, dan manfaat penelitian. Ini juga membahas metode penelitian yang digunakan, teknik pengambilan data, dan prosedur penulisan laporan.

BAB II LANDASAN TEORI

Bab ini akan membahas konsep dasar atau landasan teori yang digunakan untuk melakukan penelitian. Ini mencakup penjelasan tentang program aplikasi yang akan digunakan untuk merancang system serta kerangka pemikiran atau teori-teori yang terkait dengan pokok masalah yang diteliti.

BAB III ANALISIS MASALAH DAN PERANCANGAN SISTEM

Bab ini akan membahas analisis masalah jaringan yang ditemukan di STMIK Indonesi Mandiri. Selain itu, akan diberikan penjelasan tentang solusi yang dapat

diambil dengan memanfaatkan data dan informasi yang telah dikumpulkan sebelumnya. Selain itu, bab ini akan membahas desain dan pelaksanaan sistem pemantauan jaringan yang digunakan di kampus.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini akan membahas cara melaksanakan perancangan sistem pemantauan jaringan. Ini mencakup pemantauan kondisi jaringan yang sedang berjalan, seperti pemantauan *bandwidth* trafik, sumber daya perangkat seperti *router*, *server*, dan perangkat lainnya. Selain itu, akan dijelaskan bagaimana menggunakan *chat Telegram* dan *dashboard* aplikasi untuk memberikan peringatan dini jika terjadi masalah atau gangguan jaringan.

BAB V PENUTUP

Kesimpulan dan saran yang harus dipertimbangkan saat mengembangkan system di masa depan disajikan dalam bab ini.

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem adalah kumpulan elemen yang saling berinteraksi dan saling mempengaruhi, yang membentuk suatu kesatuan atau keseluruhan yang memiliki sifat dan karakteristik yang lebih besar daripada hanya sekedar penjumlahan elemen-elemennya. (Kuswanto 2018).

Sedangkan menurut (Erawati 2019) Sistem adalah jaringan elemen yang terintegrasi, yang berinteraksi satu sama lain dalam mencapai suatu tujuan tertentu. Sistem dapat berupa sistem fisik (misalnya mesin), sistem sosial (misalnya organisasi), atau kombinasi keduanya.

2.1.1 Bentuk Dasar Sistem

Pada jurnal (Risald 2021) suatu sistem umumnya terbagi menjadi tiga bagian, yaitu diantaranya :

1. Input: Bagian input dalam suatu sistem menerima masukan atau data dari lingkungan atau sumber eksternal. Input ini dapat berupa informasi, bahan mentah, energi, atau input lainnya yang diperlukan oleh sistem untuk beroperasi. Input ini kemudian akan diproses dan dimanfaatkan oleh sistem dalam tahap selanjutnya.
2. Proses: Bagian proses dalam suatu sistem merupakan bagian di mana masukan (input) diolah, diproses, dan diubah menjadi keluaran (output). Proses ini melibatkan aktivitas dan interaksi komponen-komponen dalam sistem untuk mencapai tujuan yang diinginkan. Dalam tahap ini, transformasi atau

pengolahan data atau bahan mentah terjadi sehingga dapat menghasilkan output yang diinginkan.

3. Output: Bagian output dalam suatu sistem merupakan hasil atau keluaran dari proses yang telah dilakukan. Output ini dapat berupa informasi, produk, layanan, atau hasil dari aktivitas sistem yang kemudian akan diterima oleh lingkungan atau penerima eksternal. Output ini menjadi hasil akhir dari sistem dan dapat digunakan atau dimanfaatkan oleh penerima eksternal atau menjadi input untuk sistem lainnya.

2.1.2 Karakteristik Sistem

Komponen, batasan, lingkungan, penghubung, masukan, keluaran, pengolah, tujuan, dan sasaran adalah beberapa karakteristik unik yang dimiliki sistem. (Dharmawan, Purwaningtyas, and Risdiansyah 2018).

1. Komponen Sistem

Sebuah sistem terdiri dari bagian-bagian yang bekerja sama untuk membentuk satu kesatuan yang utuh. Ini mungkin subsistem atau bagian-bagian sistem. Setiap subsistem memiliki fitur unik yang memungkinkannya melakukan fungsi tertentu dan berdampak pada sistem secara keseluruhan. (Dharmawan et al. 2018).



Gambar 2.1 Bentuk Dasar Sistem

2. Batas Sistem (*Boundary*)

Batas sistem adalah area yang membedakan satu sistem dari sistem lainnya atau dari lingkungan sekitarnya. Dengan batas ini, sistem dapat dianggap sebagai entitas yang lengkap. Batas sistem mempengaruhi luas sistem dan menentukan batasan yang ada. (Dharmawan et al. 2018).

3. Lingkungan Sistem (*Environment*)

Semua elemen di luar batas sistem yang memengaruhi operasinya termasuk dalam lingkungan luar sistem. Lingkungan luar dapat memiliki efek positif yang harus dipertahankan dan digunakan sebagai sumber energi bagi sistem, atau efek negatif yang harus dikontrol dan dibatasi. (Dharmawan et al. 2018).

4. Penghubung Sistem (*Interface*)

Subsistem-subsistem dalam system terhubung melalui konektivitas. Sumber daya dapat mengalir dari satu subsistem ke subsistem lainnya melalui koneksi ini. Dengan menggunakan *contact person*, *output* dari satu subsistem dapat dikirim ke subsistem lainnya. Subsistem dapat saling berinteraksi dan membentuk sebuah kesatuan yang utuh berkat koneksi ini. (Dharmawan et al. 2018).

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut *input*. Ada dua jenis masukan: masukan perawatan (*input maintenance*) dan masukan sinyal. Energi yang diberikan ke sistem untuk menjaga operasinya dikenal sebagai masukan perawatan. Namun, *energy* yang diproses sistem untuk menghasilkan keluaran adalah masukan sinyal. (Dharmawan et al. 2018).

6. Keluaran Sistem (*Output*)

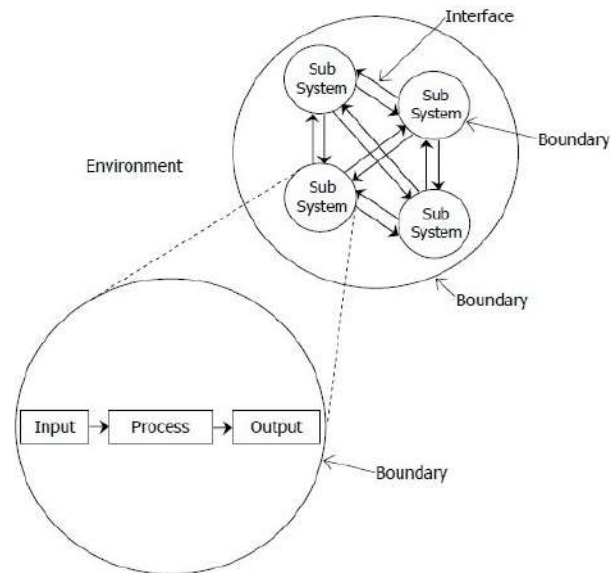
Output adalah produk dari proses pengolahan energi yang dilakukan oleh sistem. Keluaran ini terbagi menjadi dua kategori: keluaran yang bermanfaat dan limbah yang dibuang. Keluaran sampingan yang tidak digunakan atau harus dibuang sistem disebut limbah, sedangkan hasil yang diinginkan atau berguna dari sistem disebut keluaran sampingan. (Dharmawan et al. 2018).

7. Pengolah Sistem

Ada bagian pengolah sistem yang sangat penting untuk mengubah masukan menjadi keluaran. Komponen pengolah ini mengolah *input* dengan berbagai metode atau algoritma. Tujuannya adalah untuk menghasilkan keluaran yang sesuai dengan fungsi sistem. Untuk mengubah atau mengubah data atau energi, bagian pengolah berfungsi sebagai pusat aktivitas sistem. (Dharmawan et al. 2018).

8. Sasaran Sistem

Setiap sistem memiliki tujuan yang harus dicapai untuk memulai operasionalnya. Sasaran memberikan arah dan tujuan khusus bagi sistem. Jika tidak ada sasaran, sistem tidak akan memiliki arah yang jelas dan operasinya mungkin tidak menguntungkan. Sejauh mana tujuan atau sasaran tercapai dapat digunakan untuk mengukur keberhasilan sistem. Salah satu cara utama untuk menilai kesuksesan dan efektivitas suatu sistem adalah melakukan evaluasi terhadap pencapaian sasaran. (Dharmawan et al. 2018).



Gambar 2.2 Karakteristik Sistem

2.2 Pengertian *Network*

Sistem komunikasi yang terdiri dari satu atau lebih sistem telekomunikasi yang terhubung satu sama lain melalui alat komunikasi dan jalur transmisi disebut jaringan. Jaringan, yang dibentuk melalui koneksi antara sistem-sistem tersebut, memungkinkan komunikasi dan pertukaran data antara pengguna. (Wibowo 2022). Berbagi sumber daya yang terhubung ke jaringan adalah manfaat dari menggunakan jaringan komputer. Perangkat keras dan perangkat lunak yang ada di dalam jaringan termasuk dalam kategori sumber daya yang dapat dibagi.

Jaringan komputer terdiri dari sejumlah komputer yang terhubung satu sama lain melalui media dan protokol komunikasi tertentu. Memungkinkan komputer yang terhubung ke jaringan untuk berkomunikasi dan berbagi data adalah tujuan dari jaringan ini. (Mananggell, Mewengkang, and Djamen 2021).

Memastikan transmisi data yang akurat dan aman antara pengirim dan penerima serta menjaga koneksi yang konsisten adalah tujuan utama dalam pengembangan jaringan komputer. Berdasarkan standar tertentu, jaringan komputer dapat dibagi menjadi empat bagian, yang mencakup :

2.2.1 Berdasarkan Jangkauan Geografis

1. Jaringan LAN (*Local Area Network*)

Jaringan lokal, atau LAN, adalah jaringan kecil yang dimiliki oleh gedung atau kampus. Komputer pribadi dan *workstation* yang terletak di dalam gedung atau kantor terhubung melalui jaringan lokal (LAN), yang memungkinkan mereka untuk berbagi printer dan berbagi data. (Wibowo 2022).

2. Jaringan MAN (*Metropolitan Area Network*)

Metropolitan Area Network (MAN) adalah jaringan lokal yang lebih besar daripada LAN dan using teknologi yang sama. MAN dapat digunakan untuk keperluan pribadi atau umum, dan dapat mencakup beberapa kantor di sekitar atau di kota. MAN dapat mengirimkan data dan suara, dan bahkan dapat terhubung ke jaringan TV kabel. (Wibowo 2022).

3. Jaringan WAN (*Wide Area Network*)

Jaringan wilayah luas (WAN) adalah jaringan LAN dan MAN yang saling terhubung dengan cakupan yang luas, seringkali lebih dari satu negara atau bahkan sepuluh benua. WAN terdiri dari banyak *server* yang saling terhubung di seluruh dunia. (Wibowo 2022).

2.2.2 Berdasarkan Media Transmisi Data

1. Jaringan Kabel (*Wired Network*)

Jenis jaringan ini membutuhkan kabel jaringan untuk menghubungkan komputer. Kabel jaringan berperan dalam mentransfer informasi, dalam bentuk sinyal listrik, antar komputer di jaringan. (Wibowo 2022).

2. Jaringan Nirkabel (*Wireless Network*)

Teknologi nirkabel adalah jenis jaringan yang menggunakan frekuensi radio untuk menghubungkan dua atau lebih komputer. Perkembangan teknologi nirkabel dimulai dengan munculnya alat-alat seperti radio dua arah, kontrol remot, dan *walkie-talkie*. Selain itu, jaringan komputer yang menggunakan teknologi nirkabel memungkinkan pengguna untuk terhubung ke jaringan tanpa menggunakan kabel. Teknologi nirkabel memiliki banyak keuntungan, termasuk fleksibilitas, mobilitas, efisiensi waktu, dan biaya instalasi yang rendah karena memperluas atau memindahkan jaringan tidak memerlukan pengaturan atau perpindahan kabel. (Wibowo 2022).

2.2.3 Berdasarkan Peranan Komputer

1. Jaringan *Client-Server*

Satu komputer dalam jaringan *client-server* bertanggung jawab atas semua fungsi jaringan komputer; ini termasuk komunikasi, penggunaan perangkat keras dan perangkat lunak bersama, dan pengendalian jaringan. Komputer ini disebut *server*. Selain itu, komputer lain yang terhubung ke jaringan kecuali *server* disebut sebagai *client*. (Wibowo 2022).

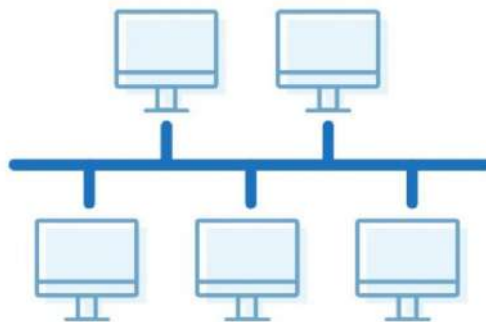
2. Jaringan *Peer-to-peer*

Setiap komputer dalam jaringan ini memiliki posisi yang sama. Tidak ada komputer yang berfungsi sebagai *server* utama. Oleh karena itu, setiap komputer di jaringan memiliki kemampuan untuk berkomunikasi satu sama lain dan berbagi perangkat keras dan perangkat lunak. (Wibowo 2022).

2.2.4 Berdasarkan Topologi Jaringan

1. Topologi Bus

Jalur komunikasi data utama atau *backbone*, yang biasanya menggunakan kabel sebagai media transmisi, menghubungkan semua komputer dalam jaringan satu sama lain. Kabel koaksial adalah jenis kabel yang umum digunakan. Pada ujung-ujung kabel, terminator digunakan untuk mengakhiri jalur komunikasi. (Wibowo 2022).



Gambar 2.3 Topologi Bus

2. Topologi Bintang (*Star*)

Setiap komputer dalam jaringan dalam topologi ini terhubung ke sebuah konsentrator melalui jalur yang berbeda. Hal ini memastikan bahwa jika salah satu komputer mengalami gangguan, jaringan secara keseluruhan tidak akan terpengaruh.

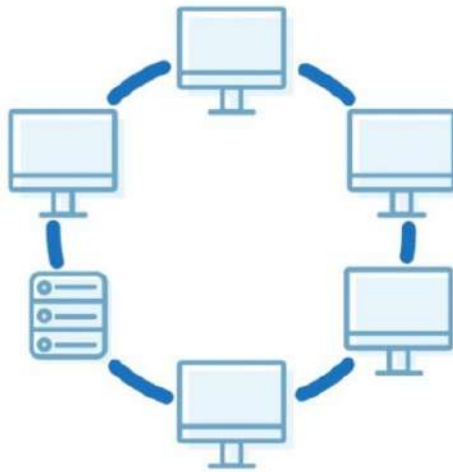
Konsentrator, baik berupa *hub* maupun *switch*, bertanggung jawab mengatur komunikasi di dalam jaringan. (Wibowo 2022).



Gambar 2.4 Topologi Star

3. Topologi Cincin (*Ring*)

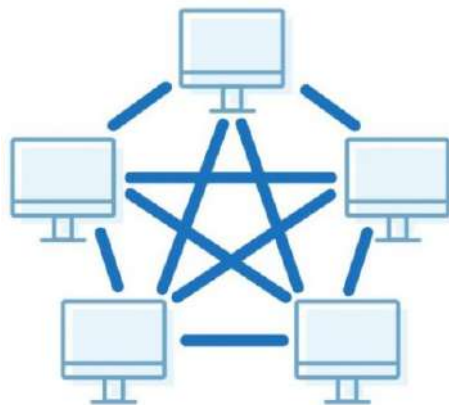
Topologi ini menghubungkan komputer-komputer satu sama lain secara berkelanjutan membentuk suatu cincin. Kedua ujung cincin dihubungkan dalam struktur sistem jaringan ini, yang menyerupai rangkaian listrik seri. Salah satu komputer dapat mengganggu jaringan secara keseluruhan. Data dalam sistem ini dikirim melalui jalur berputar di sepanjang cincin. Dengan demikian, setiap komputer yang ingin mengirimkan data ke komputer lain harus melewati cincin ini. (Wibowo 2022).



Gambar 2.5 Topologi Ring

4. Topologi *Mesh*

Dalam topologi ini, setiap perangkat memiliki koneksi langsung *point-to-point* dengan perangkat lainnya. Setiap perangkat harus dilengkapi dengan *port input* dan *port output* untuk mengakomodasi koneksi *point-to-point* tersebut. (Wibowo 2022).



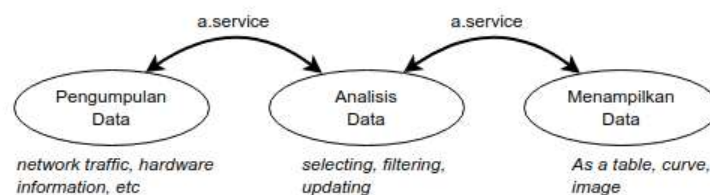
Gambar 2.6 Topologi Mesh

2.3 Pengertian Sistem Monitoring

Monitoring adalah aktivitas yang melibatkan melacak dan memperhatikan informasi yang ingin diketahui. Pemantauan dilakukan dengan tingkat ketelitian yang tinggi agar dapat mengukur pergerakan menuju atau menjauh dari tujuan seiring berjalannya waktu (Jakaria and Dede Solahudin 2019).

Dengan mempertimbangkan definisi sebelumnya, *monitoring* dapat didefinisikan sebagai proses pengawasan atau pemantauan suatu kegiatan untuk mendapatkan informasi yang diperlukan untuk pengambilan keputusan di masa mendatang.

Secara keseluruhan, langkah-langkah yang diperlukan untuk menjalankan sistem pemantauan dapat dibagi menjadi tiga proses utama, seperti yang ditunjukkan pada gambar di bawah ini. (Ramayasa and Arnawa 2015).



Gambar 2.7 Proses Dalam *Monitoring*

2.4 Pengertian *Network Monitoring*

Pemantauan jaringan adalah proses melihat dan memantau jaringan yang saling terhubung. Karena peran *server* dan klien sangat penting dalam jaringan komputer, mereka harus dipantau secara teratur melalui analisis kinerja dan pengendalian komponen secara manual. Ketika jaringan berkembang secara signifikan, hal ini menjadi lebih penting. (Aulia, Kusuma, and Rumani 2018).

2.5 Pengertian *Network Monitoring System*

Sistem Pengawasan Jaringan, juga dikenal sebagai NMS, adalah layanan yang menawarkan perangkat, aplikasi, dan alat untuk membantu manusia mengatur dan memonitor jaringan. *Administrator* atau pengguna dapat dengan mudah memantau dan merawat jaringan lingkungannya dengan menggunakan NMS yang sesuai. (Prasetyo, Budiman, and Putra 2019).

2.6 Tujuan *Network Monitoring System*

Tujuan pemantauan jaringan komputer adalah untuk mengumpulkan data penting tentang komponen jaringan sehingga dapat digunakan untuk mengelola dan mengendalikan jaringan. Oleh karena itu, diharapkan masalah atau gangguan pada jaringan dapat segera terdeteksi dan diperbaiki untuk menjamin stabilitas jaringan. (Prasetyo et al. 2019). Berikut ini adalah 10 alasan untuk menggunakan *Network Monitoring System*..:

1. Solusi NMS selalu memberikan data tentang status peralatan, koneksi, dan sumber daya jaringan untuk memahami kondisi jaringan saat ini.
2. Untuk merencanakan perbaikan dan perubahan pada peralatan jaringan.
3. Mengidentifikasi dan menganalisis masalah jaringan.
4. Menjadi dasar untuk memenuhi persyaratan perjanjian tingkat layanan (SLA).
5. Memahami kapan saat yang tepat untuk menerapkan sistem pemulihan bencana, juga dikenal sebagai pemulihan bencana, untuk mengatasi masalah yang muncul.
6. Membuat sistem aman secara keseluruhan dan beroperasi secara optimum.
7. Akses status jaringan dari jarak jauh.

8. Menjamin jaringan komputer tetap aktif untuk pengguna yang bergantung padanya dapat mengaksesnya tanpa hambatan.
9. Menghemat uang dengan mempercepat proses analisis masalah dan mengurangi durasi jaringan yang tidak aktif.

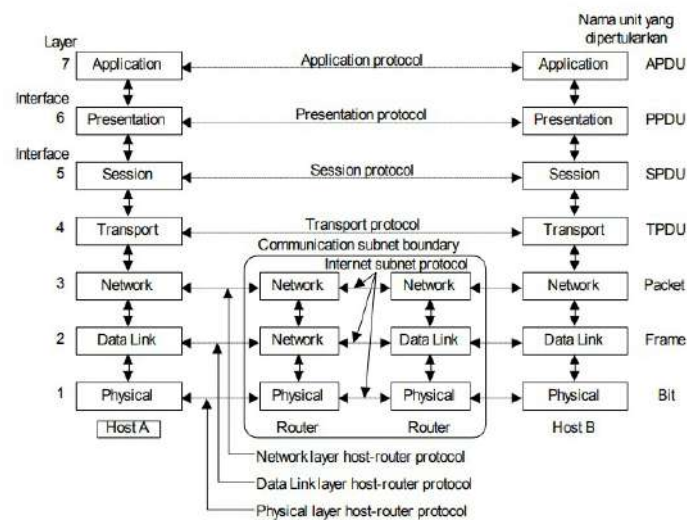
2.7 Pengertian *Internet*

Internet adalah kependekan dari *interconnecting networking*, sebuah sistem global yang terdiri dari banyak jaringan komputer yang saling terhubung melalui paket protokol *Internet* standar (TCP/IP). Jutaan orang di seluruh dunia menggunakan *internet*. (Wibowo 2022). Pengguna yang terhubung melalui komputer mereka ke *Internet* dapat memanfaatkan layanan yang disediakan oleh komputer lain, seperti berbagi gambar, video, dan audio serta bertukar informasi secara langsung. Protokol formal yang diakui oleh semua jaringan saat ini sangat penting untuk mengatasi peningkatan jumlah komputer yang terhubung ke jaringan. *Internet* juga dapat digunakan untuk berkomunikasi dan mendapatkan informasi.

2.8 Model Referensi OSI

Model untuk referensi Sebuah representasi yang disebut *Open System Interconnection* (OSI) menunjukkan bagaimana data dari aplikasi perangkat lunak di satu komputer dapat mencapai aplikasi perangkat lunak lain di komputer lain melalui jaringan. Model referensi OSI mengatur konsep ini ke dalam 7 lapisan yang berfungsi secara spesifik dalam jaringan, sebagaimana dijelaskan dalam Gambar 2.1 (tanpa media fisik). Organisasi Standar Internasional (ISO) mengusulkan model ini sebagai langkah pertama menuju standarisasi protokol internasional di berbagai tingkatan,

Tujuan dari model ini, yang dikenal sebagai Model Referensi ISO OSI, adalah untuk menghubungkan sistem terbuka, atau sistem yang memiliki kemampuan untuk berkomunikasi dengan sistem lainnya. Untuk keperluan singkat, model ini sering disebut model OSI. (Yustika 2011).



Gambar 2.8 Referensi OSI

Model *Open Systems Interconnection* (OSI) memiliki tujuh lapisan. Setiap lapisan mengikuti aturan berikut:

1. Sebuah lapisan harus dibuat jika diperlukan tingkat abstraksi yang berbeda.
2. Setiap lapisan harus memiliki tugas tertentu yang telah ditetapkan.
3. Semua fungsi harus dipilih dengan hati-hati sesuai dengan standar protokol internasional yang berlaku.
4. Batasan antara lapisan dibuat sedemikian rupa sehingga aliran data yang melewati antarmuka menjadi lebih sedikit.

5. Jumlah lapisan harus cukup banyak untuk menghindari penggabungan fungsi yang berbeda dalam satu lapisan yang tidak sesuai. Namun, agar arsitektur jaringan tetap mudah digunakan, jumlah lapisan harus diminimalkan.

2.9 Protokol Jaringan

Protokol merupakan peraturan resmi dan kesepakatan yang mengatur cara komputer saling berkomunikasi melalui jaringan. Protokol ini dapat diterapkan pada salah satu atau beberapa lapisan dalam model OSI. (Yustika 2011).

2.10 *Internet Protocol (IP)*

Protokol Internet (IP) terletak di lapisan internet dalam model *Open Systems Interconnection* (OSI), atau TCP/IP. Semua aplikasi jaringan TCP/IP membutuhkan protokol *Internet* untuk berfungsi dengan baik karena IP adalah komponen penting dari jaringan TCP/IP. Oleh karena itu, mempelajari protokol IP dan fungsinya sangat penting bagi siapa saja yang ingin memahami TCP/IP.

IP adalah sebuah protokol yang mengatur pengenalan dan pengiriman data antara komputer-komputer menggunakan IP *address*. IP merupakan protokol yang tidak memerlukan koneksi (*connectionless protocol*). Saat membangun koneksi baru sebelum data dikirimkan, IP tidak melakukan proses *handshake* atau deteksi dan pemulihan kesalahan dalam situasi ini. Dalam hal ini, IP bergantung pada lapisan lain dalam model untuk melakukan proses *handshake*. (Primarta 2020).

2.10.1 Pengalamatan IP

IP *address* adalah angka biner berukuran 32 bit yang terbagi menjadi empat bagian, masing-masing dengan 8 bit. Dalam teori, tidak mungkin ada dua atau lebih *host* yang menggunakan alamat IP yang sama untuk terhubung ke *Internet*. (Primarta 2020).

2.10.2 Alamat IP (IP Address)

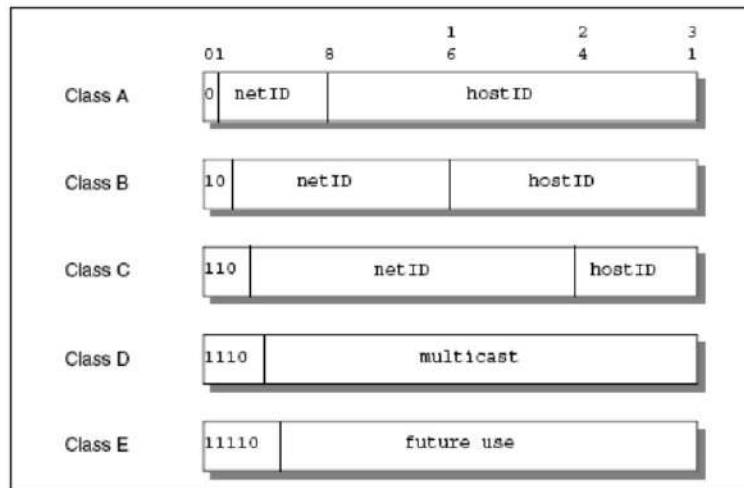
Informasi mengenai pengalamatan IP dapat ditemukan dalam RFC 1166 - *Internet Number*. Setiap *host* di *Internet* diberikan IP address atau alamat *internet* untuk mengidentifikasinya. Jika *host* tersebut terhubung ke lebih dari satu jaringan, hal tersebut disebut sebagai *multi-homed*, dimana setiap antarmuka memiliki satu IP *address*. IP *address* terdiri dari beberapa komponen yang spesifik. (Sukaridhoto 2014)

:

IP Address = <nomor *network*><nomor *host*>

2.10.3 Pembagian Kelas Alamat IP (Class-based IP Address)

Bit pertama dari alamat IP digunakan untuk menentukan sifat dan pembagian alamat IP yang tersisa. Selain itu, bit tersebut juga digunakan untuk memisahkan alamat IP menjadi dua bagian, yaitu alamat jaringan dan alamat *host*. Selain netID, alamat jaringan (*network address*) juga disebut hostID. Pembagian alamat IP terbagi menjadi lima kelas yang umum digunakan, yaitu:



Gambar 2.9 Pembagian Kelas IP

- Kelas A: Alamat jaringan 7 bit dan alamat *host* 24 bit. Dimungkinkan untuk mendukung hingga 2^7-2 (126) jaringan dengan $2^{24}-2$ (16,777,214) *host*, atau lebih dari 2 juta alamat, dengan cara ini.
- Kelas B: Alamat jaringan berukuran empat belas bit dan alamat *host* berukuran enam belas bit. Dengan cara ini, sekitar $2^{14}-2$ (16,328) *host* dan $2^{16}-2$ (65,534) *host*, atau sekitar 1 juta alamat, dapat didukung oleh jaringan.
- Kelas C memiliki 21 bit alamat jaringan dan 8 bit alamat *host*. Dengan cara ini, sekitar $2^{21}-2$ (2,097,150) jaringan dengan 2^8-2 (254) *host*, atau sekitar setengah juta alamat, dapat didukung.
- Kelas D: Alamat ini hanya digunakan untuk *multicast*.
- Kelas E: Alamat ini dibuat untuk digunakan di masa mendatang.

Jaringan dengan banyak *host* menggunakan Kelas A. Jaringan yang lebih kecil dengan jumlah *host* kurang dari 254 menggunakan Kelas C. Namun, kelas B harus digunakan untuk jaringan dengan lebih dari 254 *host*.

2.11 *Internet Control Message Protocol (ICMP)*

Internet Control Message Protocol (ICMP) adalah protokol yang digunakan saat *router* atau *host* tujuan ingin mengirimkan informasi tentang kerusakan pada IP datagram. Beberapa karakteristik dari ICMP meliputi (Sukaridhoto 2014) :

1. Protokol IP digunakan oleh ICMP untuk menyampaikan informasi tentang kerusakan pada datagram IP.
2. ICMP melaporkan masalah atau kerusakan saat pengiriman paket data melalui jaringan.
3. ICMP tidak use pesan ICMP untuk melaporkan kerusakan untuk menghindari pengulangan.
4. Pesan ICMP hanya dikirim pada data yang terfragmentasi untuk melaporkan kerusakan pada fragmentasi pertama.
5. Pesan ICMP tidak menanggapi data *broadcast* atau *multicast*.
6. ICMP tidak akan menanggapi *datagram* IP tanpa *header* IP pengirim.
7. Pesan ICMP dapat menunjukkan kerusakan IP *datagram*.
8. Spesifikasi ICMP tersedia dalam RFC 792, dan pembaruan dapat ditemukan dalam RFC 950.

2.11.1 *Pesan ICMP*

1. *Echo dan Echo Reply*

Penggunaan fitur *Echo* adalah untuk memeriksa apakah suatu *host* aktif atau tidak. Jika *host* tersebut aktif, maka akan dikirimkan balasan berupa pesan *Echo Reply*. (Sukaridhoto 2014).

2. *Destination Unreachable*

Router mengirim pesan untuk menunjukkan bahwa *host* target tidak dapat dijangkau. (Sukaridhoto 2014).

3. *Source Quench*

Ketika tidak ada *buffer* yang cukup untuk meneruskan *datagram*, *router* akan mengirimkan pesan ini. (Sukaridhoto 2014).

4. *Redirect*

Setelah *router* mengirimkan pesan ini, *host* diminta untuk mengirimkan *datagram* berikutnya yang ditetapkan dalam pesan ICMP ke *router* di jaringan. (Sukaridhoto 2014).

5. *Time Exceeded*

Router mengirimkan pesan yang menunjukkan bahwa pesan yang dikirim telah mencapai batas waktu (TTL). *Ping* dan *Traceroute* adalah beberapa program yang menggunakan protokol ICMP. (Sukaridhoto 2014).

2.11.2 PING

Ping adalah salah satu program sederhana dalam aplikasi TCP/IP. Dengan menggunakan *Ping*, kita dapat mengirimkan IP *datagram* ke tujuan tertentu, menghitung waktu perjalanan pulang-pergi (jam perjalanan), dan menerima tanggapan dari tujuan tersebut. Singkatan dari *Packet InterNet Groper*, *Ping* digunakan untuk memeriksa konektivitas jaringan (Sukaridhoto 2014).

Ping menggunakan pesan *echo* dengan *echo response* ICMP. Selain itu, Anda juga dapat using Ping untuk memastikan bahwa alamat IP diinstal pada sebuah host.

Ini adalah apa yang dapat Anda lakukan:

1. Mengirimkan pesan *echo* ICMP ke alamat IP yang dimaksud.
2. Menunggu tanggapan ICMP *echo respond* dari tujuan.
3. Menerima tanggapan dan menghitung waktu perjalanan pulang-pergi.
4. Analisis respons untuk memastikan bahwa instalasi alamat IP berhasil.

Langkah-langkah ini memungkinkan Ping untuk memeriksa ketersediaan *host* dan koneksi jaringan.

```
uji-heAljazari-Desk:~$ ping google.com
PING google.com (142.251.10.102) 56(84) bytes of data:
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=1 ttl=56 time=18.4 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=2 ttl=56 time=18.4 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=3 ttl=56 time=18.8 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=4 ttl=56 time=18.5 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=5 ttl=56 time=18.9 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=6 ttl=56 time=18.8 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=7 ttl=56 time=18.6 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 18.393/18.605/18.851/0.172 ms
```

Gambar 2.10 Contoh Hasil Perintah PING

2.11.3 Traceroute

Aplikasi yang disebut *Traceroute* memungkinkan Anda melacak jalur yang dilalui oleh paket data selama perjalanan mereka menuju tujuan yang telah ditentukan.

Traceroute bekerja dengan mengirimkan pesan-pesan dengan nilai TTL (*Time-to-Live*) awal yang kecil. Ketika pesan tersebut mencapai suatu titik tertentu, Pengirim akan menerima pesan “*time exceeded*” sebagai tanggapan ketika TTL menjadi 0. Setelah itu, *host* pengirim akan mengirimkan pesan ICMP baru yang memiliki nilai TTL yang lebih tinggi. Proses ini akan terus berlanjut hingga paket data mencapai

host tujuan yang dituju. Dengan cara ini, *traceroute* membantu mengidentifikasi jalur-jalur yang dilewati oleh paket data melalui jaringan (Sukaridhoto 2014).

```

My traceroute [v0.93]
Aljazari-Desk (10.100.10.6) 2023-01-14T23:35:55+0700
Keys: Help Display mode Restart statistics Order of fields quit
      Packets
Host      Loss%  Snt  Last  Avg  Best  Wrst StDev
1. _gateway 0.0%  16   2.5   2.4   1.7   3.3   0.4
2. 192.168.1.1 0.0%  16   2.7   2.2   1.7   3.0   0.4
3. 45.126.80.161 0.0%  16   3.3   3.4   3.1   4.2   0.3
4. 103.61.248.209 0.0%  16   7.0   7.2   6.1  10.9   1.4
5. (waiting for reply)
6. 210.210.161.98.cbn.net.id 0.0%  16   7.3   8.1   7.1   9.8   1.0
7. 210.210.161.101.cbn.net.id 0.0%  16   7.7   8.3   6.9   9.6   0.9
8. ip67-106.cbn.net.id 0.0%  16   7.5   8.7   7.0  10.7   1.1
9. s2-211-242.190.203.detik.com 0.0%  16   6.8   6.8   6.0   8.1   0.7

```

Gambar 2.11 Contoh Hasil Perintah *Traceroute*

2.12 Address Resolution Protocol (ARP)

Protokol ARP memautkan alamat IP ke alamat MAC (alamat fisik) dalam jaringan lokal. Perangkat pengirim menggunakan protokol ARP mengubah alamat IP menjadi alamat MAC (Sukaridhoto 2014).

2.13 PORT

Port digunakan untuk berkomunikasi dengan *Port* jaringan TCP/IP digunakan untuk proses komunikasi antar *host* dan berfungsi dengan protokol lainnya. Nomor 16-bit digunakan oleh *port* untuk mengidentifikasi aplikasi atau layanan yang beroperasi di *host* tersebut. Ada dua jenis *port* yang biasa digunakan (Sukaridhoto 2014) :

- **Well-known** : Seperti telnet yang menggunakan *port Well-known*, *port server* berkisar dari 1 hingga 1023. IANA mengatur *port* yang dikenal, yang diberikan kepada proses sistem yang membutuhkan akses khusus dengan hak istimewa.

- **Ephemeral** : Klien tidak menggunakan *port* yang diketahui karena mereka telah setuju sebelumnya untuk menggunakan *port* tertentu saat berkomunikasi dengan *server*. *Port ephemeral* berkisar dari 1023 hingga 65535. Sebagai aturan, dua aplikasi tidak dapat menggunakan satu nomor port secara bersamaan.

2.14 User Datagram Protocol (UDP)

Pada dasarnya, *User Datagram Protocol* adalah Aplikasi IP, seperti protokol *Internet*, memungkinkan perangkat lunak di satu komputer untuk mengirim pesan ke komputer lain melalui jaringan tanpa komunikasi awal. Jadi ketika Anda mengirim pesan menggunakan UDP, komputer pengirim dan penerima tidak perlu bernegosiasi saat bertukar data. (Sukaridhoto 2014).

Berikut adalah beberapa aplikasi yang menggunakan protokol UDP :

- *Trivial File Transfer Protocol* (TFTP)
- *Simple Network Management Protocol* (SNMP)
- *Remote Procedure Call* (RPC) pada *Network File System* (NFS)
- *Domain Name System* (DNS)
- *Lightweight Directory Access Protocol* (LDAP)

2.15 Simple Network Management Protocol (SNMP)

Salah satu protokol yang dikembangkan oleh *Internet Engineering Task Force* (IETF) dalam *Internet Protocol Suite* adalah SNMP (*Simple Network Management Protocol*). Aplikasi di lapisan 7 (*layer 7*) yang digunakan oleh sistem manajemen jaringan untuk memantau perangkat jaringan adalah SNMP. Dengan menggunakan

SNMP, sistem ini dapat menyediakan informasi yang dibutuhkan oleh pengelola jaringan. (Sukaridhoto 2014).

2.15.1 Management Information Base (MIBS)

Sebuah basis data yang disebut MIB (*Management Information Base*) digunakan untuk mengatur perangkat jaringan. Objek jaringan seperti *router* dan *switch* termasuk dalam basis data ini. Struktur untuk Informasi Manajemen Versi 2 (SMIv2) adalah nama untuk komponen MIB yang didefinisikan using *Abstract Syntax Notation One* (ASN.1). Untuk menganalisis MIB, ada program yang disebut kompiler MIB. (Sukaridhoto 2014).

Beberapa RFC yang membahas topik ini adalah RFC1155, yang membahas struktur dan identifikasi Informasi Manajemen untuk jaringan berbasis TCP/IP, RFC1213, yang mengatur Basis Data Informasi Manajemen untuk Manajemen Jaringan berbasis TCP/IP, dan RFC1157, yang membahas protokol Manajemen Jaringan Sederhana.

Protokol manajemen jaringan sederhana, atau SNMP, digunakan untuk menghubungkan station manajemen, seperti konsol, dengan objek manajemen, seperti *router*, *gateway*, dan *switch*, MIB. Seorang petugas SNMP, yang merupakan perangkat lunak yang memiliki kemampuan untuk berkomunikasi dengan manajer SNMP, bertanggung jawab untuk mengambil data selama proses ini.

2.15.2 Arsitektur SNMP

SNMP terdiri dari beberapa *framework* diantaranya :

1. *Master Agent*

Perangkat lunak yang dikenal sebagai *master agent* bekerja pada perangkat yang mendukung SNMP dan bertanggung jawab untuk menanggapi permintaan dari stasiun manajemen melalui SNMP. *Master agent* kemudian meneruskan permintaan tersebut ke subagent yang lebih memahami manajemen objek tertentu. Subagent ini dapat memberikan informasi yang relevan tentang fungsi manajemen tertentu. (Sukaridhoto 2014).

2. *Subagent*

Master agent adalah sebuah program yang berjalan pada perangkat yang mendukung SNMP dan memiliki kemampuan untuk melayani permintaan yang diterima dari stasiun manajemen melalui protokol SNMP. Setelah *master agent* menerima permintaan, dia akan meneruskan permintaan ke *sub agent* yang lebih memahami tugas manajemen tertentu. *Sub agent* bertugas memberikan informasi manajemen kepada *master agent*, yang selanjutnya dikirim ke stasiun manajemen. (Sukaridhoto 2014) :

- Mengumpulkan data dari objek yang dikelola
- Mengatur konfigurasi data objek yang dikelola
- Respon terhadap permintaan master
- Menghasilkan trap atau alarm

3. *Management Station*

Sebagai klien, *management station* memulai permintaan dan menerima notifikasi (*trap*) dari *server* SNMP. (Sukaridhoto 2014).

2.16 *Transmission Control Protocol (TCP)*

Protokol TCP adalah jenis protokol yang dapat diandalkan dan berbasis koneksi, yang memungkinkan aliran byte dari satu mesin dikirimkan tanpa kesalahan. TCP membagi *byte* menjadi pesan-pesan dan kemudian merakitnya kembali saat diterima. Selain itu, Untuk memastikan pengirim yang cepat tidak mengirimkan terlalu banyak pesan kepada penerima yang lambat, TCP juga mengatur pengendalian aliran. Komunikasi TCP melakukan beberapa fungsi, seperti (Sukaridhoto 2014) :

- *Open* : membuka koneksi dengan memasukan beberapa parameter antara lain :
 - 1 Aktif / Pasif
 - 2 Informasi soket tujuan
 - 3 Nomor *port* lokal
 - 4 Nilai *timeout*
 - 5 *Send* : mengirimkan *buffer* data ke tujuan
- *Receive* : Menerima dan menyalin data kepada *buffer* milik pengguna
- *Close* : Menutup koneksi
- *State* : Melihat informasi
- *Abort* : Membatalkan semua kegiatan *send* atau *receive*

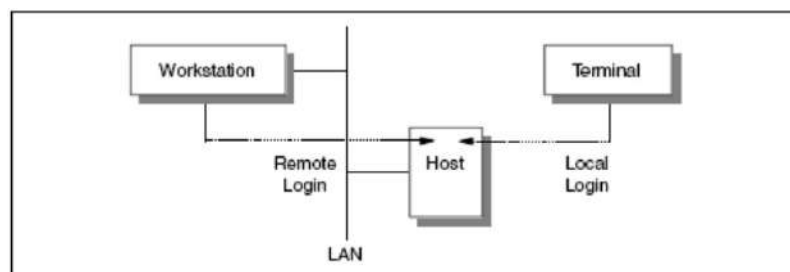
Beberapa contoh umum aplikasi atau layanan jaringan yang menggunakan TCP adalah :

- Telnet *default port 23*
- *File Transfer Protocol (FTP) default port 21*
- *Simple Mail Transfer Protocol (SMTP) default port 25*
- *Hyper-Text Transfer Protocol (HTTP) default port 80*
- *Secure Shell (SSH) default port 22*

2.17 Telnet

Telnet adalah protokol standar yang ditetapkan oleh STD nomor 8. Rincian tentang protokol Telnet dijelaskan dalam RFC 854 yang berjudul "*Telnet Protocol Specification*" dan RFC 855 yang berjudul "*Telnet Option Specifications*" (Sukaridhoto 2014).

Telnet memungkinkan antarmuka program di satu host (klien Telnet) untuk mengakses sumber daya di *host* lain (*server Telnet*). Ini memungkinkan pengguna klien memiliki pengalaman yang sebanding dengan menggunakan *host* mereka sendiri.



Gambar 2.12 Proses Kerja Telnet

2.18 *File Transfer Protocol (FTP)*

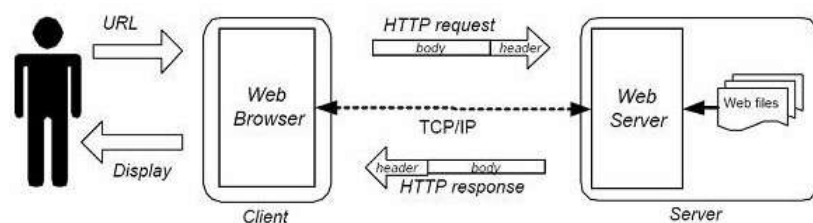
FTP adalah protokol internet yang digunakan untuk berkomunikasi data antara klien dan *server* di jaringan. FTP berfungsi pada lapisan aplikasi, memungkinkan pengguna mengirim dan menerima file melalui koneksi TCP/IP. (Sukaridhoto 2014).

2.19 *Simple Mail Transfer Protocol (SMTP)*

Selama proses pengiriman *email*, protokol SMTP berfungsi untuk berkomunikasi antara klien *email* lokal dan *server email*. Klien email dapat mengirim pesan melalui koneksi TCP/IP ke *server email* untuk dikirim ke tujuan tertentu dengan menggunakan SMTP. Melalui proses ini, agen *transfer* surat (MTA) mengelola *email* di *mail server* Anda sebelum akhirnya dikirim ke *server* surat penerima. (Sukaridhoto 2014).

2.20 *Hyper-Text Transfer Protocol (HTTP)*

HTTP adalah protokol yang mengatur komunikasi informasi antara klien dan *server*. *Client* berfungsi sebagai penerima informasi dan mengirimkan permintaan ke *server*, sementara *server* berfungsi sebagai penyedia informasi dan menanggapi permintaan dengan mengirimkan data yang diminta ke *client*. (Sukaridhoto 2014).



Gambar 2.13 Proses Kerja HTTP

Pada dasarnya, operasi HTTP relatif mudah. Saat peramban *web* digunakan untuk mengakses situs *web*, HTTP berfungsi untuk menghubungkan pengguna ke *World Wide Web* (WWW) dan memproses alamat URL (*Uniform Resource Locator*). Selanjutnya, HTTP mengambil data menggunakan protokol TCP/IP. Setelah itu, pengguna dapat menggunakan peramban untuk melihat situs *web* atau halaman dengan kontennya. Proses ini mengirimkan data melalui HTTP :

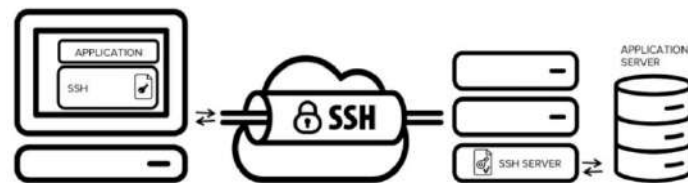
- Komputer klien memulai koneksi dan mengirimkan informasi ke *server web*.
- Permintaan tersebut dijawab oleh *server* HTTP.
- Untuk menanggapi permintaan, *web server* akan mengirimkan kode status data.
- Setelah memproses permintaan klien, *Web server* akan mengakhiri atau menutup sambungan.

2.21 *Secure Shell* (SSH)

SSH adalah sebuah protokol *transfer* yang memungkinkan pengguna untuk mengontrol perangkat secara *remote* melalui koneksi *internet*. Protokol ini menawarkan tiga teknologi enkripsi yang meliputi (Sukaridhoto 2014) :

- **Enkripsi Simetris** : Menggunakan kunci rahasia untuk mengenkripsi dan mendekripsi data klien dan *host*. Kunci rahasia disimpan di *server* dan klien. Kunci tidak ditransfer selama *transfer*.
- **Enkripsi Asimetris** : Menggunakan dua kunci terpisah untuk mengenkripsi dan mendekripsi data, membuat pasangan kunci publik-pribadi. Informasi yang dienkripsi dengan kunci publik hanya dapat ditulis dengan kunci pribadi pasangannya.

- **Hashing** : Ini adalah teknik transmisi satu arah yang tidak dapat diuraikan. Tujuannya adalah untuk memastikan bahwa peretas dan pihak lain tidak mengubah data yang dikirim melalui SSH.



Gambar 2.14 Proses Kerja SSH

2.22 Pengertian Web

Sistem yang disebut “*web*” memungkinkan orang menyajikan dan mengakses dokumen dalam format teks, gambar, multimedia, dan konten lainnya melalui jaringan *internet*. (Sibero 2013).

Dapat diambil kesimpulan oleh penulis bahwa *web* itu adalah sebuah *platform* yang memungkinkan berbagai jenis informasi digital, termasuk teks, gambar, audio, video, dan animasi, diakses dan ditampilkan melalui jaringan *internet*.

2.22.1 Jenis-Jenis Web

Perkembangan *web* mengalami peningkatan yang signifikan seiring dengan kemajuan teknologi informasi yang pesat. Fungsi, fitur, gaya desain, dan bahasa pemrograman yang digunakan adalah faktor utama dalam pembagian *website* menjadi jenis. Adapun jenis-jenis *web* dibawah ini:

2.22.2 Jenis-Jenis Web berdasarkan sifat atau *style*

Berdasarkan sifatnya, *website* dapat dibagi menjadi dua, yaitu diantaranya :

1. *Website Statis*

Website statis adalah website yang mempunyai tampilan tetap tidak berubah. Konten dari website statis juga sama, jarang sekali terjadi update.(Admin 2021).

2. *Website Dinamis*

Website dinamis adalah website yang kontennya selalu diperbarui secara berkala. Perubahan atau penambahan konten bisa dilakukan oleh seorang developer atau dilakukan oleh beberapa orang yang memiliki akses untuk memperbarui konten. Kebanyakan website saat ini berupa website dinamis karena pengelolaannya yang mudah.(Admin 2021).

2.22.3 Berdasarkan dari segi bahasa pemrograman

Ditinjau dari bahasa pemrograman *website* terbagi atas :

1. *Server side*

Website yang beroperasi di sisi *server*, atau di sisi *server*, menggunakan bahasa pemrograman yang berbeda-beda tergantung pada jenis *server* yang ada. PHP, ASP, .NET, dan lainnya adalah contohnya. Kehadiran *server* sangat penting karena *website* yang dibangun dengan bahasa pemrograman tersebut tidak akan berfungsi dengan baik jika tidak ada *server*.(Supono and Putratama 2018).

2. *Client side*

Website client-side memiliki kemampuan untuk berfungsi tanpa memerlukan *server* secara langsung. Hal ini memungkinkan pengguna mengakses dan mengoperasikan situs *web* langsung dari perangkat mereka tanpa mengandalkan *server*. Dalam situasi ini, klien bertanggung jawab untuk memproses dan menampilkan situs *web*, menggunakan bahasa pemrograman seperti HTML, CSS, dan JavaScript. *Website* dari sisi klien memungkinkan pengguna berinteraksi secara langsung dengan konten *website* tanpa melibatkan permintaan dan *responds server*. (Supono and Putratama 2018).

2.23 *Web Server*

Perangkat lunak yang dikenal sebagai *web server* berfungsi sebagai *server* dan menerima permintaan klien untuk situs *web* melalui protokol HTTP atau HTTPS. Klien biasanya adalah *web browser*. Dengan mengirimkan konten dari situs *web* yang diminta kembali ke klien, *web server* bertanggung jawab untuk menjawab permintaan tersebut. *Web server* memiliki kemampuan untuk menangani banyak permintaan secara bersamaan, menyediakan layanan *hosting* untuk situs *web*, menyimpan data dan *file*, dan mengelola interaksi dengan *database* atau aplikasi lainnya. *Web server* kemudian mengembalikan hasil respon berupa *website* yang biasanya terdiri dari dokumen-dokumen dengan menggunakan bahasa pemrograman seperti PHP, ASP dan .NET. (Supono and Putratama 2018)

Dapat disimpulkan dari pengertian diatas bahwa *web server* adalah bagian yang paling utama untuk menjalankan sebuah aplikasi berbasis web untuk menerima

perintah atau permintaan dari *client*, *server internet* yang digunakan sebagai koneksi dan *transfer* data HTML, PHP, ASP, .NET dan lainnya. Selain itu *web server* adalah aplikasi terpenting dari *world wide web* (WWW).

2.24 Apache

Apache adalah perangkat lunak *web server* yang menangani permintaan dan tanggapan HTTP. Selain itu, Apache juga memiliki kemampuan *logging* untuk mencatat detail aktivitas. Apache juga dikenal sebagai *web server* yang mudah digunakan dan modular yang menggunakan protokol HTTP biasa. Keunggulan ini membuat Apache menjadi salah satu *web server* yang paling disukai pengguna. Hal ini dapat dilihat dari jumlah pengguna Apache yang jauh lebih banyak dibandingkan dengan pesaingnya. Berdasarkan survei oleh *Netcraft* pada bulan Januari 2005, pangsa pasar Apache mencapai tidak kurang dari 68% dari total *web server* yang berjalan di Internet. Artinya, jika semua *web server* selain Apache digabungkan, jumlahnya masih belum dapat menyaingi Apache. (Agusvianto 2017).

2.25 HyperText Markup Language (HTML)

HTML (*Hypertext Markup Language*) adalah bahasa dasar yang digunakan dalam pengembangan *web* dengan sifatnya sebagai *scripting client-side*. Dengan menggunakan HTML, kita dapat menampilkan informasi berupa teks, grafik, multimedia dan juga link antar halaman *web* melalui *hyperlink*. Tidak diperlukan editor khusus untuk menggunakan kode HTML ini. Kita dapat menggunakan program penyuntingan yang memiliki antarmuka pengguna grafis (GUI), seperti Notepad, Edit Plus, atau program lainnya. Antarmuka berbasis ikon dapat melakukan semua

perintah yang kita butuhkan, jadi dengan menggunakan program-editor ini, kita tidak perlu mengetik kode HTML secara manual. Memahami kode HTML sangat penting sebagai seorang pengembang aplikasi *web* karena akan memungkinkan kita untuk menguasai program editor lainnya. Saat mempelajari kode HTML secara mendalam, disarankan untuk menggunakan editor teks seperti Notepad.(Agusvianto 2017).

Penulis dapat sampai pada kesimpulan berikut berdasarkan pendapat para ahli di atas salah satu jenis bahasa yang disebut HTML adalah yang digunakan untuk mengatur tampilan dan struktur halaman *web*, mengirimkan data, dan membuat aplikasi. Kami dapat membuat dokumen yang dapat diakses melalui *browser* dengan menggunakan HTML. Ini akan memungkinkan konten seperti teks, gambar, video, audio, dan elemen lainnya.

2.26 Hypertext Preprocessor (PHP)

PHP adalah bahasa skrip sisi *server* yang dimaksudkan untuk pengembangan *web*. PHP juga merupakan bahasa pemrograman umum. Rasmus Lerdorf membangun PHP pertama kali pada tahun 1995, dan saat ini dikelola oleh PHP Group. Halaman PHP resmi dapat ditemukan di <http://www.php.net>. Karena kode PHP diproses di komputer *server*, PHP disebut sebagai bahasa pemrograman sisi *server*. Ini tidak sama dengan bahasa pemrograman sisi klien seperti JavaScript yang digunakan oleh *browser*.(Agusvianto 2017).

Dari pemahaman di atas, dapat disimpulkan bahwa PHP adalah salah satu bahasa pemrograman yang paling umum digunakan untuk mengolah konten *web*

dinamis. Ini digunakan bersama dengan bahasa pemrograman HTML yang berjalan di atas *web server*.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

Gambar 2.15 Contoh syntax PHP

2.27 Javascript

JavaScript adalah bahasa skrip yang digunakan dalam dokumen HTML dan telah menjadi bahasa skrip yang paling banyak digunakan di *internet* sejak awal. Bahasa ini berfungsi sebagai bahasa pemrograman dan memiliki fitur tambahan pada bahasa HTML dengan memungkinkan perintah dilaksanakan oleh pengguna, yaitu dalam browser daripada di *server web*. (Suryana and Koesheryatin 2014).

Dari pendapat penulis di atas dapat disimpulkan bahwa JavaScript adalah bahasa pemrograman atau *scripting* yang berisi instruksi-instruksi untuk dijalankan oleh kode HTML.

```
<button onclick="myFunction()">Click Me!</button>

<script>
function myFunction() {
  let x = document.getElementById("demo");
  x.style.fontSize = "25px";
  x.style.color = "red";
}
</script>
```

Gambar 2.16 Contoh syntax javascript

2.28 Python

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja. Python biasa dipakai dalam pengembangan situs dan perangkat lunak, membuat analisis data, visualisasi data dan otomatisasi tugas. Karena sifatnya yang relatif mudah dipelajari, bahasa pemrograman ini digunakan secara luas oleh non-programmer seperti ilmuwan dan akuntan untuk melakukan tugas harian mereka. Misalnya, dalam mengatur keuangan.(Intern 2023).

```
a = 33
b = 200
if b > a:
    print("b is greater than a")
```

Gambar 2.17 Contoh syntax Python

2.29 MySQL

MySQL adalah sistem manajemen database relasional (RDBMS) open-source berbasis SQL yang bekerja dengan model client-server. Kalau DBMS adalah sistem manajemen database secara umum, RDBMS merupakan software pengelolaan database berdasarkan model relasional.(Christy 2023).

MySQL adalah *Database* pertama yang didukung oleh bahasa *scripting web* PHP dan Perl adalah MySQL. PHP dan MySQL dianggap sebagai perangkat lunak terbaik untuk pengembangan aplikasi *web*. MySQL umumnya digunakan dalam

pembuatan aplikasi berbasis *web*. Pengembangan aplikasi biasanya menggunakan bahasa skrip PHP.

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Gambar 2.18 Contoh syntax MySQL

2.30 Telegram

Telegram adalah aplikasi pesan instan lintas platform yang gratis, berbasis cloud, dan tidak menghasilkan uang. Aplikasi Telegram tersedia untuk berbagai perangkat dan sistem operasi, seperti Android, iOS, Windows Phone, Ubuntu Touch, serta Windows, MacOS X, dan Linux untuk komputer. Pengguna dapat mengirim pesan teks, foto, video, file audio, dokumen, stiker, dan jenis file lainnya melalui Telegram. (Klusaite 2023).

2.31 Zabbix

Zabbix adalah perangkat lunak sumber terbuka yang memungkinkan Anda memantau dan memeriksa status layanan, *server*, dan perangkat keras jaringan lainnya.. Dengan Zabbix, pengguna dapat dengan mudah melihat status dan kondisi jaringan serta menerima peringatan dini melalui notifikasi di *dashboard*, *email*, SMS, dan juga melalui *platform* pesan instan seperti *Telegram*. Zabbix juga mendukung metode *polling* dan *trapping*. Perangkat lunak ini menyediakan antarmuka grafis (GUI) yang memudahkan pengguna dalam mengelola dan mengatur sistem, termasuk tampilan peta (*map*) dan grafik untuk mempermudah administrasi dan pemantauan. (Mediyanto and Mahendra 2017).

Selain semua manfaatnya, Zabbix dapat membantu administrator sistem dan jaringan dalam melakukan semua tugasnya, dan juga dapat berfungsi sebagai alat penting untuk memonitor infrastruktur jaringan. Zabbix memiliki fitur berikut :

- Prestasi dan kapasitas yang luar biasa—mampu memantau ratusan ribu perangkat sekaligus.
- Penemuan otomatis perangkat jaringan dan *server*.
- Pemantauan tersebar dengan manajemen *web* terpusat.
- Dukungan untuk mekanisme pengumpulan dan *polling*.
- Perangkat lunak klien Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, OS X, Tru64/OSF1, Windows 200, Windows server 2003, Windows XP, Windows Vista, Windows Server 2008, dan Windows 7) dengan tingkat kinerja tinggi.
- Pemantauan *web*
- Otentikasi pengguna yang aman
- Izin pengguna yang dapat disesuaikan
- Antarmuka berbasis *web*
- SLA, dan ITIL KPI pada pelaporan
- Pemberitahuan ke platform chat Telegram yang fleksibel
- Sumber daya tingkat tinggi yang dilihat melalui layar konsol visual dan dashboard visual yang diatur oleh pengguna
- *Log Audit*

2.32 Pengertian Waterfall

Salah satu model yang paling umum digunakan dalam pengembangan perangkat lunak adalah model *waterfall*. Model ini menggunakan metode linear, dimulai dari tahap perencanaan hingga tahap pemeliharaan. Tahapan pengembangan sistem mengikuti urutan yang terstruktur, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya.(Pricillia and Zulfachmi 2021). berikut dibawah ini adalah tahapan dalam metode *waterfall*:

A. ***Communication (Project Initiation & Requirement Gathering)***

Sebelum memulai pekerjaan teknis, sangat penting untuk berkomunikasi dengan pelanggan untuk memahami dan mencapai tujuan yang diinginkan. Analisis masalah dan pengumpulan data yang diperlukan, serta bantuan dalam mendefinisikan fitur dan fungsi perangkat lunak, adalah hasil dari komunikasi yang mendorong pelaksanaan proyek. Jurnal, *article*, dan sumber informasi lainnya, seperti *internet*, dapat menawarkan data tambahan (Risald 2021).

B. ***Planning (Estimating, Scheduling, & Tracking)***

Perencanaan adalah tahap selanjutnya, di mana estimasi tugas teknis yang akan dilakukan, identifikasi risiko yang mungkin terjadi, penentuan sumber daya yang diperlukan untuk membangun sistem, definisi produk kerja yang ingin dicapai, penjadwalan pelaksanaan pekerjaan, dan pemantauan proses pengembangan sistem. (Risald 2021).

C. ***Modelling (Analysis & Design)***

Fase perencanaan dan permodelan arsitektur sistem ini fokus pada desain struktur data, arsitektur perangkat lunak, antarmuka pengguna, dan algoritma

program. Tujuannya adalah untuk mendapatkan pemahaman yang komprehensif tentang pekerjaan yang akan dilakukan secara keseluruhan (Risald 2021).

D. *Construction (Code & Test)*

Mengubah desain menjadi bentuk atau kode yang dapat dipahami mesin adalah proses yang termasuk dalam tahapan pembangunan ini. Setelah proses pengkodean selesai, sistem dan kode diuji. Tujuannya adalah untuk menemukan dan mengidentifikasi kesalahan sehingga dapat diperbaiki (Risald 2021).

E. *Deployment (Delivery, Support & Feedback)*

Implementasi perangkat lunak kepada pelanggan, pemeliharaan rutin, perbaikan, evaluasi, dan pengembangan berdasarkan umpan balik adalah semua bagian dari tahapan *deployment*. Tahap ini dilakukan dengan tujuan memastikan system tetap beroperasi dan berkembang sesuai dengan fungsinya (Risald 2021).

2.32.1 Kelebihan Waterfall

Kelebihan metode *waterfall* adalah sebagai berikut, kemampuan untuk melakukan kontrol dengan baik.(Supandi et al. 2018).

1. Kualitas sistem akan baik karena implementasi dilakukan secara bertahap..
2. Untuk mengurangi kesalahan, proses pengembangan model dilakukan secara bertahap.

3. Dokumen pengembangan sistem sangat terorganisir karena setiap fase harus selesai sebelum melanjutkan.




2.33 Unified Modeling Language (UML)








UML adalah Analisis dan desain, definisi persyaratan, dan penjelasan arsitektur dalam pemrograman berorientasi objek adalah standar bahasa industri yang populer. (Putra and Andriani 2019).

1. Use Case Diagram

Sebuah model yang digunakan untuk menggambarkan perilaku sistem informasi yang akan dibangun disebut *use case diagram*. *Use case diagram* menggambarkan interaksi yang biasa terjadi antara pengguna sistem dan sistem itu sendiri melalui skenario penggunaan. (Putra and Andriani 2019).

Tabel 2.1 Simbol *Use Case Diagram* (Aprianti and Maliha 2016)


No	Simbol	Nama	Keterangan
1		<i>Actor</i>	mengidentifikasi jenis peran yang dimainkan pengguna saat berinteraksi dengan kasus
2		<i>Dependency</i>	hubungan di mana elemen yang bergantung pada elemen yang tidak independen dipengaruhi oleh perubahan elemen independen (<i>independent</i>).
3		<i>Generalization</i>	hubungan di mana objek berbagi perilaku struktur data dengan objek induk.


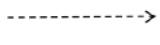


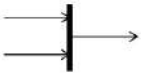
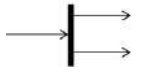
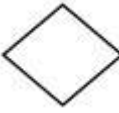
4		<i>Include</i>	Mencantumkan spesifikasi sumber secara eksplisit
5		<i>Extend</i>	Menentukan bahwa target kasus penggunaan memperluas perilaku sumber kasus penggunaan dari titik tertentu.
6		<i>Association</i>	Apa yang menghubungkan satu objek dengan objek lainnya.
7		<i>System</i>	Menentukan paket yang berisi sistem terbatas.
8		<i>Use Case</i>	deskripsi dari urutan tindakan yang ditampilkan oleh sistem yang menghasilkan hasil yang dapat diukur untuk seorang aktor
9		<i>Collaboration</i>	Deskripsi urutan tindakan yang ditunjukkan oleh sistem yang memberikan hasil terukur bagi operator.
10		<i>Note</i>	Komponen fisik yang ada saat aplikasi dijalankan dan berfungsi sebagai representasi sumber daya komputasi.

2. *Activity Diagram*

Activity diagram adalah Sebuah diagram aktivitas digunakan untuk menunjukkan alur kerja atau aktivitas sistem dalam perangkat lunak. (Putra and Andriani 2019).

Tabel 2.2 Simbol *Activity Diagram* (Aprianti and Maliha 2016)

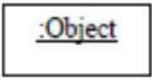




No	Simbol	Nama	Keterangan
1		<i>Activity</i>	menunjukkan cara masing-masing kelas antarmuka berinteraksi satu sama lain.

2		<i>Control flow</i>	Menunjukkan urutan eksekusi
3		<i>Object flow</i>	Menunjukkan aliran objek dari tindakan ke tindakan
4		<i>Start Point</i>	menunjukkan bahwa sebuah hal dimulai atau dibentuk
5		<i>End Point</i>	menunjukkan bahwa sebuah objek memiliki bentuk atau akhir
6		<i>Join/ Penggabungan</i>	Menunjukkan niat untuk menggabungkan kembali tindakan atau tugas yang berjalan bersamaan
7		<i>Fork</i>	menunjukkan untuk membagi tindakan menjadi tindakan atau aktivitas yang berbeda
8		<i>Decision</i>	menunjukkan pilihan atau tindakan yang harus diambil dalam kondisi tertentu

3. *Sequence Diagram*

Sequence diagram menunjukkan interaksi antara objek dalam suatu *use case* dengan menunjukkan waktu hidup objek dan pesan yang dikirim dan diterima. Diagram ini direpresentasikan dengan memuat urutan aktivitas dari objek-objek tersebut. Untuk mencakup seluruh interaksi pesan yang terjadi dalam *use case* yang telah didefinisikan, jumlah *sequence diagram* yang perlu dibuat sebanding dengan jumlah *use case* yang ada. (Putra and Andriani 2019).

Tabel 2.3 Simbol *Sequence Diagram* (Aprianti and Maliha 2016)








No	Simbol	Nama	Keterangan
1		<i>Object</i>	Objek adalah turunan dari kelas dan ditulis secara horizontal. Direpresentasikan sebagai kelas (kotak) dengan titik koma di nama objek.
2		<i>Actor</i>	Aktor juga memiliki kemampuan untuk berinteraksi dengan objek, sehingga aktor juga dapat diurutkan ke dalam kolom. Ikon aktor sama dengan ikon aktor pada diagram use case.
3		<i>Lifeline</i>	lifeline menunjukkan keberadaan objek dalam jangka waktu tertentu. Penanda lifeline adalah garis putus-putus vertikal yang digambar oleh suatu objek.
4		<i>Activation</i>	Aktivasi disebut sebagai kotak persegi panjang yang digambar di garis hidup. Aktivasi menunjukkan objek yang melakukan tindakan.
5		<i>Message</i>	Pesan diwakili oleh panah horizontal di antara aktivasi. Pesan menunjukkan komunikasi antar objek.

4. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah sebuah diagram grafis yang digunakan dalam pembuatan database untuk menghubungkan data satu dengan yang lain. ERD berfungsi sebagai alat bantu untuk merancang database dan memberikan gambaran tentang bagaimana database tersebut akan beroperasi. Diagram ini terdiri dari tiga elemen dasar, yaitu entitas, atribut, dan relasi. Dengan ERD, dapat dijelaskan

struktur data, hubungan antar entitas, dan atribut yang terkait dalam sebuah sistem *database*.(Afiifah, Azzahra, and Anggoro 2022).

Tabel 2.4 Simbol *Entity Relationship Diagram*

Simbol	Nama	Keterangan
	<i>Entity</i>	Objektif di luar sistem berkomunikasi dengan sistem baik sebagai pemakai maupun sumber data.
	<i>Relationship line</i>	Diagram ini menggunakan garis penghubung.
	<i>Option Symbol</i>	Digunakan untuk <i>relasi</i> fungsional
	<i>One Symbol</i>	Digunakan pada <i>relasi</i> menunjukan 1 (<i>one</i>)
	<i>Many</i>	Digunakan pada <i>relasi</i> untuk menunjukan banyak (<i>many</i>)
	<i>Composite entity</i>	Menggambarkan sebuah entitas yang komposit
	<i>Weak entity</i>	entitas yang kemunculannya dipengaruhi oleh entitas yang lebih kuat

Tiga komponen yang akan dibentuk selama pembuatan ERD adalah:

1. Entitas

Entitas adalah objek yang menjadi fokus dalam sebuah *database*. Entitas dapat berupa manusia, tempat, benda, atau kondisi yang berkaitan dengan data yang diperlukan. Dalam *Entity Relationship Diagram* (ERD), entitas direpresentasikan dengan simbol berbentuk persegi panjang. (Afiifah, Azzahra, and Anggoro 2022).

2. Hubungan (relasi/*relationship*)

Relasi dalam ERD adalah hubungan antara dua entitas atau lebih. Relasi ini menunjukkan bagaimana entitas-entitas terkait satu sama lain dalam *database*. Dalam *Entity Relationship Diagram* (ERD), relasi direpresentasikan dengan simbol berbentuk belah ketupat. (Afiifah, Azzahra, and Anggoro 2022).

3. Atribut

Atribut adalah informasi yang terkait dengan suatu entitas. Setiap entitas memiliki atribut yang menggambarkan karakteristiknya, termasuk atribut utama (*primary key*) yang menjadi identitas unik entitas tersebut, serta atribut deskriptif lainnya. Atribut dapat berada dalam tabel entitas atau dapat diwakili secara terpisah. Dalam *Entity Relationship Diagram* (ERD), atribut direpresentasikan dengan simbol berbentuk elips. (Afiifah, Azzahra, and Anggoro 2022).

a. Atribut *key*

Atribut kunci adalah satu atau kombinasi atribut yang dapat mengidentifikasi secara unik setiap baris data (*baris/record*) dalam sebuah tabel. Dianggap unik jika atribut kunci tidak dapat memiliki baris data dengan nilai yang sama. Contoh:

Nomor Induk Mahasiswa (NPM), NIM, dan nomor induk lainnya

b. Atribut *simple*

Atribut yang nilainya merupakan nilai atomik tidak dapat dipisah/dipisahkan lagi. Contoh:

Alamat, penerbit, tahun terbit, judul buku.

c. Atribut *multivalued*

Nilai suatu fitur yang memiliki lebih dari satu nilai fitur yang bersangkutan. Sebagai contoh, dari sebuah buku, yang memiliki beberapa pengarang.

d. Atribut *composite*

Salah satu atribut majemuk adalah atribut yang terdiri dari banyak atribut yang lebih kecil yang memiliki makna yang sama, yang dapat dibagi lagi atau mengandung sub-atribut. Contoh:

Kumpulan nama, mis. nama depan, nama tengah dan nama belakang

e. *Atribut derivatif*

Atribut yang tidak membutuhkan penyimpanan *database* Contohnya, sepenuhnya atau sifat yang berasal dari hubungan atau atribut lainnya. Fungsi ini ditunjukkan dengan garis putus-putus berbentuk oval .

dibawah ini notasi kardinalitas pada *diagram* ER (Jayanti and Sumiari 2018):

a. *Exactly one* (1:1)

Entitas A hanya memiliki satu *record* yang berhubungan dengan entitas B.

b. *Zero, One or More* (0:>1)

Satu atau lebih *record* di entitas A berhubungan dengan satu atau lebih *record* di entitas B atau mungkin tidak berhubungan sama sekali.

c. *One or More* (1:>1)

Satu atau lebih *record* di entitas B pasti berhubungan dengan satu *record* di entitas A.

d. *Zero or One* (1:>1)

Satu catatan di entitas A berhubungan dengan satu catatan di entitas B atau tidak berhubungan sama sekali.

2.34 Pengujian Sistem

Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Perangkat lunak atau yang sering dikenal dengan sebutan software hanyalah satuan elemen dari sistem berbasis komputer yang lebih besar. Biasanya,

perangkat lunak dihubungkan dengan perangkat lunak dan perangkat keras lainnya. (Kurniawati 2018).

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya. (Kurniawati 2018).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 *Communication*

Pada tahapan ini melakukan kegiatan untuk memperoleh informasi yang berkaitan dengan penelitian, melakukan analisi permasalahan dan menyelesaikan permasalahan.

3.1.1 Metode Pengumpulan Data

Tujuan dari metode pengumpulan data adalah untuk mendapatkan informasi yang relevan dengan masalah. Ada beberapa teknik yang digunakan untuk mengumpulkan data, antara lain:

1. Wawancara

Untuk mengumpulkan data, wawancara ini mengadakan tanya jawab dengan orang-orang yang memiliki kemampuan untuk memberikan data dan informasi terkait dengan masalah yang diteliti.

2. Studi Literatur

Sumber referensi dikumpulkan dari buku dan jurnal yang relevan tentang sistem pemantauan jaringan (NMS) berbasis *web*, serta informasi lain yang relevan dengan penelitian.

Tabel 3.1 Tabel Studi Literatur

No	Literatur	Pembahasan
1	Jurnal Teknologi informasi judul : Implementasi Sistem <i>Monitoring</i> Jaringan Menggunakan Zabbix Berdasarkan SNMP Penulis : Arya Pradana,	Tujuan penelitian ini adalah untuk melihat dan menggunakan <i>router</i> yang menampilkan data jaringan, Permasalahan kurangnya tenaga <i>network administrator</i> menjadi salah satu tujuan dari penelitian ini, karena dapat membantu untuk

	<p>Indrastanti R.Widiasari , Rissal Efendi Data Jurnal : AITI: Jurnal Teknologi Informasi, Volume 19 No. 2 Agustus 2022, 248- 262 ISSN 1693-8348 E-ISSN 2615-7128</p>	<p>memecahkan sebuah masalah, aplikasi tersebut menggunakan basis SNMP untuk menampilkan data dari perangkat yang akan dimonitor.</p>
2	<p>Jurnal Teknologi Informasi judul : Implementasi Sistem Pemantauan Jaringan Menggunakan Librenms Pada Jaringan Kampus Universitas Udayana Penulis : I Wayan Krisna Saputra, Dewa Made Wiharta, Nyoman Putra Sastra Data Jurnal : Jurnal SPEKTRUM Vol. 7, No. 2 Juni 2020</p>	<p>Penelitian ini bertujuan untuk memantau kondisi jaringan yang berada dilingkungan kampus agar siswa dapat menggunakan sistem pembelajaran dengan nyaman dan efektif. Aplikasi yang dikembangkan menggunakan libreNMS yang berfungsi untuk memonitoring <i>traffic</i> dan penggunaan sumber daya di setiap perangkat jaringan yang diawasi. Sistem peringatan fleksibel dan auto-deteksi adalah beberapa fitur LibreNMS dengan <i>dashboard</i> yang dapat disesuaikan dengan kebutuhan <i>monitoring</i>.</p>
3	<p>Jurnal Teknologi Informasi Judul : Analisis Dan Perancangan <i>Monitoring</i> Dan Notifikasi <i>System Web Application Firewall</i> Menggunakan Zabbix Penulis : Haryono Stefanus Eko Prasetyo Data Jurnal : https://journal.uib.ac.id/index. php/combines Volume 1 No 1 (2021)</p>	<p>Penelitian ini menjelaskan tentang Analisis dan perancangan Zabbix untuk monitoring dan notifikasi <i>Web Application Firewall</i>, menjelaskna bahwa aplikasi NMS ini sangat memberikan dampak positif untuk perkembangan pada jaringan, selain dari fungsinya sebagai alat untuk monitoring, NMS juga bisa dijadikan sebagai alat untuk memberikan laporan terkait dengan <i>upgrade</i> perangkat, perbaikan perangkat dan dijadikan sebagai alat untuk menerima pelaporan jaringan.</p>
4	<p><i>Journal of Internet and Software Engineering (JISE)</i> Judul : Pemantauan <i>Router CPE</i> pada jaringan metro <i>ethernet</i> menggunakan zabbix berbasis raspberry PI Penulis : Aris Hartono, Unan Yusmaniar Oktiawati Data Jurnal : Vol. 2 No. 1 July 2021</p>	<p>Penelitian ini menggunakan perangkat lain untuk memonitor setiap <i>site</i> yaitu dengan menggunakan raspberry PI, setiap lokasi selalu dipasang perangkat raspberry PI dimana perangkat tersebut sudah terinstall aplikasi NMS menggunakan zabbix.</p>
5	<p>Jurnal Sistem Informasi Judul : Analisis dan</p>	<p>Penelitian ini membahas tentang nms menggunakan fitur yang ada pada mikrotik</p>

<p>Implementasi Sistem Monitoring Koneksi <i>Internet</i> Menggunakan The Dude Di STIKOM Al Khairiyah Penulis : Sutarti, Alif Alfiyansyah Data Jurnal : Volume.4, Agustus 2017 p-ISSN:2406-7768 e-ISSN: 2581-2181</p>	<p>di Kampus STIKOM AL Khairiyah. Sistem monitoring menggunakan Dude hanya untuk melihat status perangkat apakah dalam kondisi <i>up</i> atau kondisi <i>down</i>. Setelah itu informasi gangguan di kirim ke <i>email</i>.</p>
---	---

Dengan mempertimbangkan literatur sebelumnya, kita dapat sampai pada kesimpulan berikut menjelaskan seberapa pentingnya keberadaan NMS disuatu jaringan, terutama pada jaringan yang sangat besar (Instansi, kampus, Hotel, dan lain sebagainya). Dengan adanya NMS dalam sebuah jaringan akan sangat membantu tugas dari seorang *network administrator* atau *system administrator* karena akan memudahkan pengecekan ketika ada masalah dalam jaringan. Selain memudahkan pengecekan, NMS juga bisa dijadikan sebagai *tools* untuk pengolahan data laporan bulanan misalkan untuk memunculkan info penggunaan *bandwidth internet, resource server* (CPU, Memory, Storage, traffic interface).

3.1.2 Analisis Sistem

Tahap analisis sistem melibatkan pengumpulan data dan analisis dokumen yang berkaitan dengan sistem yang beroperasi. Tahap ini bertujuan untuk memudahkan evaluasi kekurangan sistem yang ada sehingga dapat dilakukan perbaikan dan pembuatan sistem baru. Analisis juga mencakup hal-hal yang tidak fungsional, seperti analisis perangkat lunak dan perangkat keras.

3.1.3 Analisis Sistem Jaringan

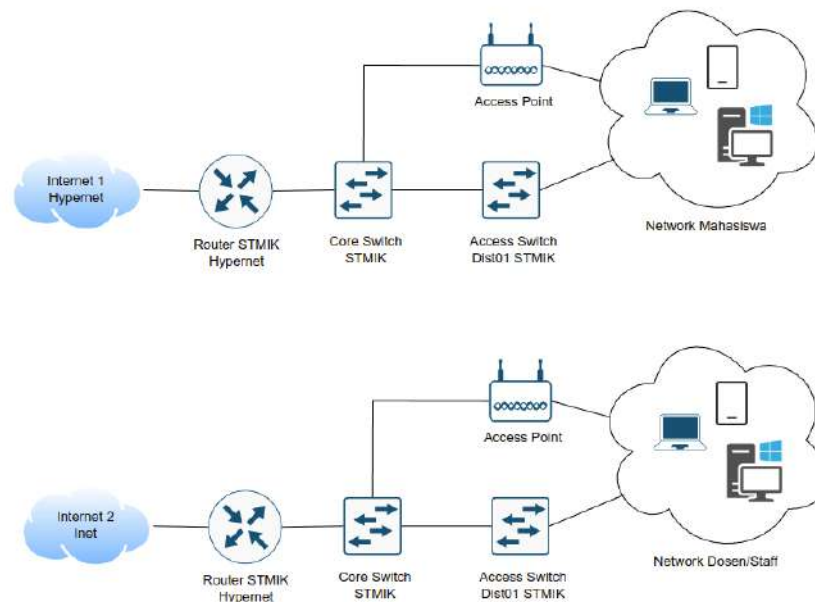
Selama penulis menganalisis dari hasil wawancara, kondisi jaringan di lingkungan STMIK Indonesia mandiri menemukan beberapa masalah yang telah dianalisa antara lain :

1. Sulitnya mengidentifikasi permasalahan jaringan secara tepat.
2. Keterbatasan pemantauan jaringan secara *Real-Time* Sehingga menyebabkan tidak adanya notifikasi atau peringatan saat terjadi gangguan.
3. Tidak adanya pemetaan jaringan yang terintegrasi antar perangkat.
4. Tidak adanya laporan dan analisis yang komprehensif.
5. Tidak adanya acuan yang dijadikan standar untuk menjalankan aktifitas jaringan (Perbaikan, Pembaharuan dan Perawatan).

Hal tersebut bisa menjadi penyebab terhambatnya proses analisis permasalahan, *troubleshoot*, dan bisa membuat *downtime* jaringan lebih lama, karena tidak ada *tools* atau sistem yang berfungsi sebagai alat *monitoring*.

3.1.4 Topologi Jaringan Saat Ini

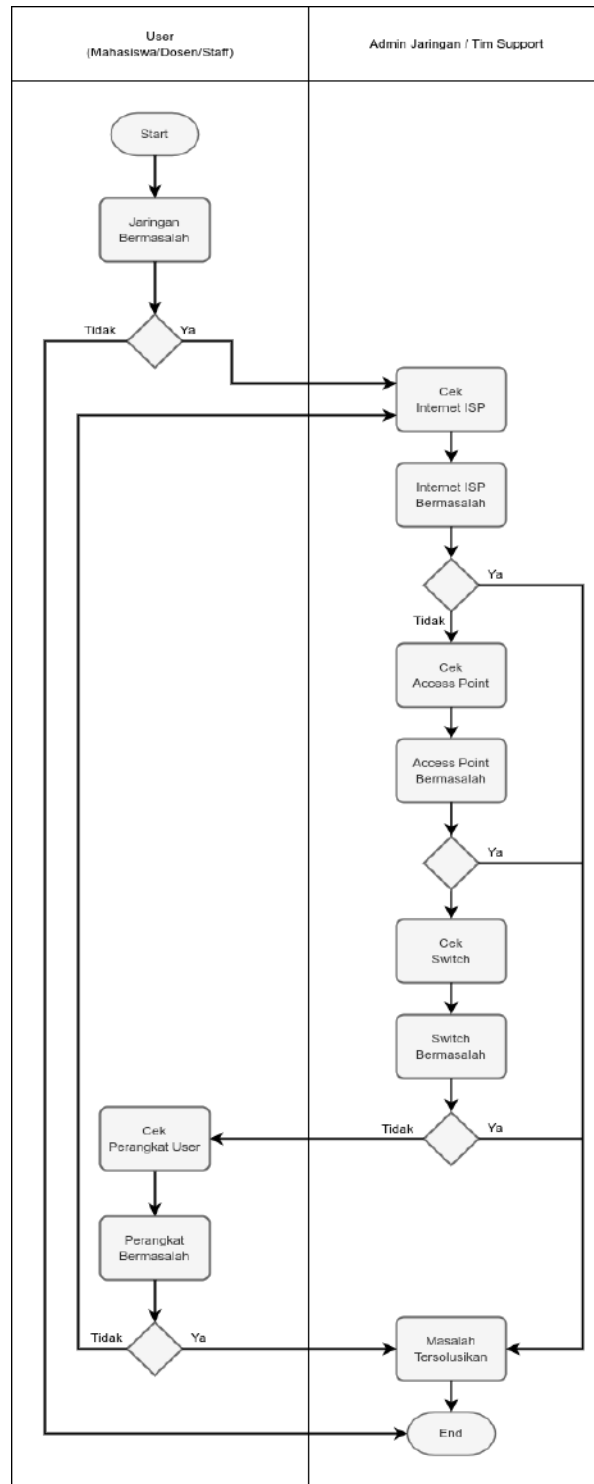
Topologi adalah desain jaringan komputer. Gambar 3.1 menunjukkan topologi jaringan yang digunakan di STMIK Indonesia Mandiri. dibawah ini.



Gambar 3.1 Topologi STMIK Indonesia Mandiri Saat Ini

3.1.5 Analisis Sistem Monitoring Yang Sedang Berjalan

Pada sistem yang sedang berjalan ini, proses *monitoring* jaringan di kampus STMIK Indonesia Mandiri masih menggunakan metode manual dan tradisional, dimana admin jaringan atau tim *support* hanya menunggu komplain tanpa ada peringatan dini atau *alert* yang memberitahukan bahwa kondisi jaringan mengalami gangguan. Berikut dibawah ini *flow map* untuk proses pengecekan jika ada kendala :



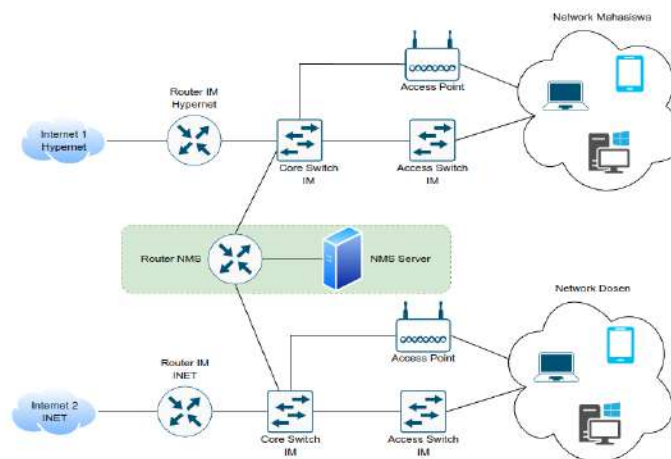
Gambar 3.2 Flow Map Metode Pengecekan Jaringan Saat Ini

3.1.6 Evaluasi Sistem Yang Berjalan

Sistem yang berjalan saat ini, apabila dilihat dari prosedur pengecekan ketika ditemukan permasalahan akan membutuhkan waktu yang cukup lama, karena harus melewati beberapa proses pengecekan. Yang menjadi evaluasi dari pengecekan jaringan secara manual adalah ketika kita menemukan sebuah masalah dan mencari sumber permasalahan dari jaringan tersebut. Harus dilakukan bertahap pengecekannya dimulai dari pengecekan koneksi ke arah *internet*, perangkat jaringan *router*, *access point* dan *switch/hub*. Jaringan *internet* diarea kampus mempunyai urgensi yang sangat tinggi sehinggal dalam proses *recovery* jaringan harus dilakukan dengan *downtime* yang tidak terlalu lama karena bisa mengganggu aktivitas kampus.

3.1.7 Topologi Jaringan Yang Diusulkan

Terdapat perubahan pada topologi yang diusulkan, yaitu penempatan *Router* dan *Server NMS* pada jaringan saat ini. Seperti diperlihatkan pada gambar 3.2 dibawah ini :

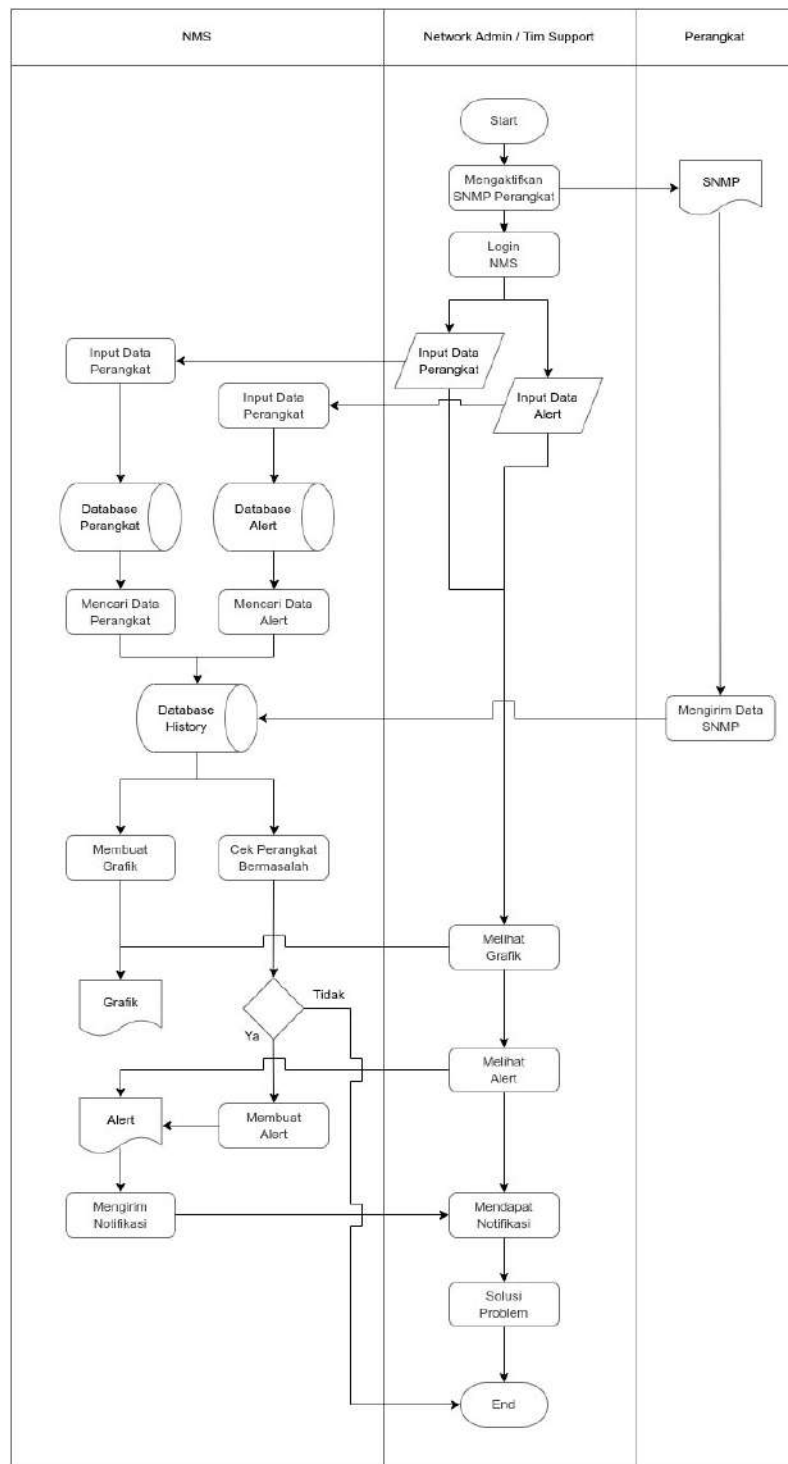


Gambar 3.3 Topologi STMIK Indonesia Mandiri Yang Diusulkan

3.1.8 Analisis Sistem Monitoring Yang Diusulkan

Tujuan utama pembuatan sistem pemantauan jaringan ini adalah untuk meningkatkan efisiensi dan kemudahan identifikasi masalah yang terjadi dalam jaringan. Dalam sistem pemantauan jaringan yang lama pemeriksaan kondisi jaringan jika terpantau ada kendala dilakukan secara manual, sedangkan dalam sistem pemantauan jaringan yang baru akan dibuat sebuah sistem pemantauan jaringan menggunakan sebuah program *Network Monitoring System* yang bertujuan untuk memudahkan proses pemeriksaan dan analisa sehingga *downtime* jauh lebih rendah.

Lihat gambar 3.2 dibawah ini :



Gambar 3.4 Flow Map Sistem Yang Diusulkan

Deskripsi dari *Flow Map* sistem yang diusulkan adalah sebagai berikut :

1. *Network Administrator* mengaktifkan *service* SNMP pada perangkat yang akan dimonitor oleh NMS
2. *Network Administrator* melakukan *login* ke NMS dan melakukan *input* data perangkat dan *input* data *alert*.
3. NMS menerima masukan data berupa verifikasi dari SNMP Community.
4. NMS menerima masukan data berupa verifikasi dari SNMP Community.
5. Data atau laporan yang telah didapat dari setiap perangkat akan dimasukkan ke *Database* pada NMS.
6. Data yang didapat dari perangkat akan dipetakan sesuai dengan id *item* dan id perangkat.
7. Data yang telah dimasukkan ke *database* akan disajikan oleh NMS dalam bentuk grafik.
8. Apabila komunikasi antara NMS dengan perangkat terputus, maka NMS akan membuat laporan berupa *alert* ke *Network Administrator* atau tim *support* yang menjelaskan bahwa ada salah satu perangkat atau *node* yang mengalami masalah.
9. Setelah *Network Administrator* atau tim *support* mendapatkan *alert* maka segera akan dilakukan investigasi dan analisa terkait dengan gangguan yang terjadi sehingga dapat ditemukan solusi terbaik untuk penanganan dari permasalahan tersebut.

3.1.9 Analisa Kebutuhan Perangkat Server

1. Perangkat Lunak NMS *Server*

Aplikasi Sistem Pengawasan Jaringan ini dibuat menggunakan perangkat lunak berikut :

- a. Sistem Operasi Centos 7.8
- b. Sublime Text 4 Build 4121
- c. PHP *Version* 7.2.31
- d. Zabbix-Release 3.4.15
- e. MariaDB 10.5.5
- f. Apache2 *Version* 2.4
- g. Python 2.7
- h. openssh-server *Version* 7.4p1
- i. snmp-server 5.7

2. Perangkat Lunak *User* (Pengguna)

Berikut ini adalah perangkat lunak yang digunakan oleh pengguna untuk menggunakan NMS :

- a. *Web Browser* (untuk akses ke aplikasi NMS)
- b. *Telegram* (Sebagai alat untuk menerima *alert*)
- c. *Email Client* (Sebagai alat untuk menerima *alert*)

3.1.10 Analisa Kebutuhan Untuk Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi Sistem Pengawasan Jaringan. Karena NMS ini adalah aplikasi yang bersifat *client-server* maka dibutuhkan perangkat keras untuk menunjang kerja dari NMS berikut server yang dipergunakan untuk kebutuhan NMS *Server* dilingkungan jaringan STMIK Indonesia Mandiri :

1. Processor : Intel(R) Xeon(R) E5450 @3.00Ghz 4 Core
2. RAM : 16GB
3. Harddisk : 250GB
4. Perangkat pendukung lainnya : Router Mikrotik RB750r2.

3.2 Planning (Estimating, Scheduling, Tracking)

Pada tahap ini, banyak proses yang harus diselesaikan, mulai dari tahap komunikasi hingga tahap konstruksi. Akibatnya, penjadwalan yang tepat diperlukan untuk memastikan penelitian ini selesai pada waktunya. Jadwal penelitian berikut didasarkan pada kegiatan yang dilakukan dengan skala waktu mingguan. Tabel 3.1 akan memperlihatkan jadwal pengerjaan dalam penelitian ini.

Tabel 3.2 Penjadwalan Penelitian

Tahap	No	Aktifitas	Januari				Februari				Maret				Mei				Mei				Juni			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Communication	1	Pengumpulan Data	■	■	■																					
	2	Analisis Permasalahan dan Kebutuhan				■	■																			
	3	Analisis Sistem				■	■																			
	4	Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras				■	■																			
Estimating, Scheduling, Tracking	1	Penjadwalan						■																		
Analysis and Design	1	Perancangan Sistem Menggunakan UML dan ERD							■	■																
	2	Perancangan <i>User Interface</i>									■	■														
Construction	1	Pengkodean											■	■	■	■	■	■	■	■						
	2	Pengujian (<i>Black Box</i>)																			■	■				

3.3 Modeling (Analysis and Design)

3.3.1 Perancangan Sistem

Proses membangun sistem baru menggunakan sistem lama yang sedang berjalan dikenal sebagai perancangan sistem. Perancangan sistem dilakukan untuk menyelesaikan masalah yang ada pada sistem lama dengan membuat sistem baru. Setelah analisis dan persiapan, tahap perancangan sistem dilakukan. Tujuan merancang dan membangun implementasi sistem adalah untuk menunjukkan bagaimana sistem tersebut akan dibentuk. Penggambaran, perancangan, dan pembuatan sketsa atau pengaturan komponen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi adalah semua bagian dari pekerjaan ini. Konfigurasi komponen perangkat keras dan perangkat lunak sistem juga termasuk dalam perancangan sistem.

3.3.2 Tujuan Perancangan Sistem

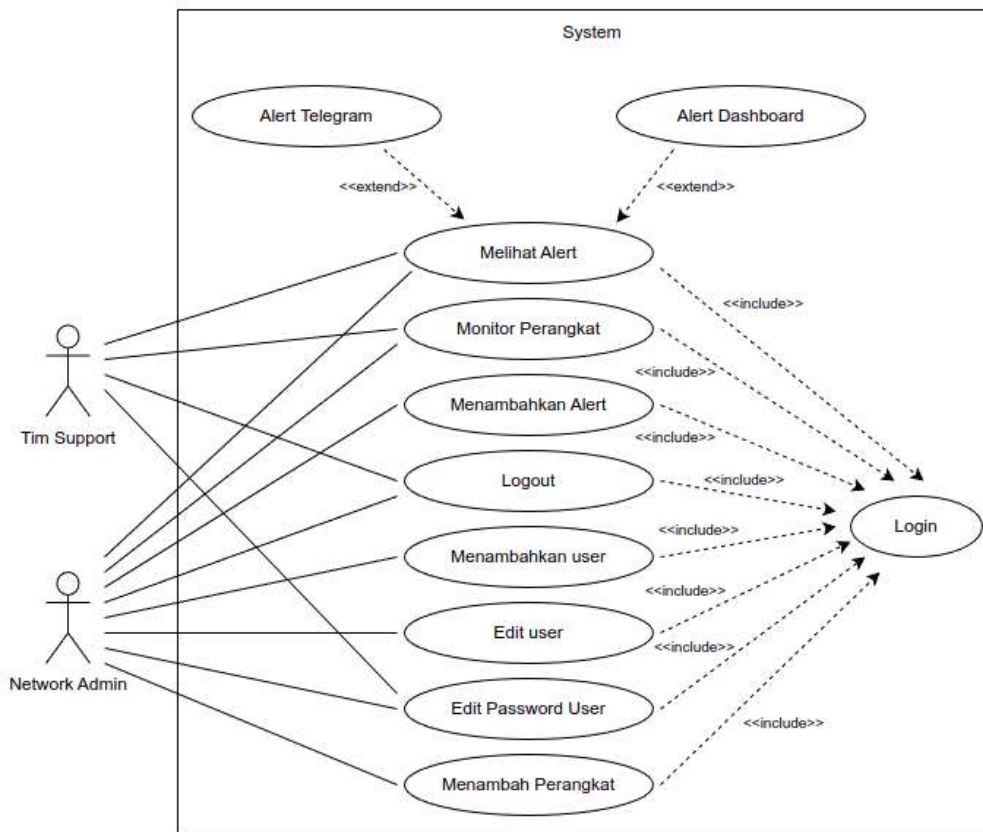
Tujuan dari proyek sistem ini adalah membuat aplikasi sistem pendukung keputusan yang terkomputerisasi. Aplikasi ini dirancang untuk membantu administrator atau tim support mengidentifikasi masalah yang terjadi pada jaringan mereka. Mereka dapat mengidentifikasi dan menyelesaikan masalah jaringan dengan menggunakan aplikasi ini, serta dapat memudahkan dalam membuat laporan performansi pada perangkat jaringan seperti penggunaan *bandwidth*, *resource server*, *switch* dan *router*. Tujuan dari tahap per 3.1 Diagram Koancangan sistem ini adalah menghasilkan desain sistem yang rinci dan terorganisir. Rancangan arsitektur sistem, pemodelan data, desain antarmuka pengguna, dan spesifikasi teknis komponen sistem

semua termasuk dalam desain ini. Perancangan sistem juga mencakup pemilihan teknologi yang tepat dan pengaturan konfigurasi yang ideal untuk memenuhi kebutuhan sistem baik yang berfungsi maupun yang tidak berfungsi. “*Network Monitoring System Berbasis Web* dengan sistem peringatan dini Dan *Mapping Jaringan Di STMIK Indonesia Mandiri*”. Komponen sistem informasi yang dirancang termasuk beberapa komponen penting, seperti :

1. Perancangan Proses (*Flowmap, usecase, activity diagram, sequence diagram* dan *class diagram*).
2. Perancangan *Database* (Tabel Relasi).
3. Perancangan Program (Perancangan Arsitektur, Perancangan Antar Muka).

3.3.3 Use Case Diagram

Untuk menunjukkan persyaratan fungsional yang diharapkan dari sebuah sistem, model diagram Use Case (UML) digunakan. Dalam lingkungan sistem perangkat lunak yang akan dibangun dan ditunjukkan dalam Use Case Diagram. Berdasarkan analisis yang telah dijelaskan sebelumnya, diagram use case berikut dapat dibuat untuk menjelaskan hubungan antara fungsi yang disediakan oleh sistem :



Gambar 3.5 Use Case Diagram

Tabel 3.3 Definisi Aktor

No	Aktor	Definisi
1	Network Admin	Pihak yang melakukan hak akses admin untuk menambahkan perangkat, menambahkan <i>alert</i> , dan menambahkan <i>user</i>
2	Tim Support	Pihak yang mendapatkan notifikasi <i>alert email</i> dan <i>telegram</i> dan dapat memonitor perangkat

3.3.4 Spesifikasi Use Case

Untuk keterangan spesifikasi *use case* diterangkan dalam **Tabel 3.4** sampai dengan **Tabel 3.9** berikut :

Tabel 3.4 Spesifikasi Use Case Admin

<i>Use Case Name</i>	Login
<i>Actor</i>	Admin
<i>Brief Description</i>	<i>Use Case</i> ini digunakan untuk <i>login</i> , sebagai pintu utama.
<i>Basic Flow</i>	Admin mengisi <i>username</i> dan <i>password</i> pada halaman <i>login</i> Admin
<i>Alternate Flow</i>	1. Jika <i>username</i> atau <i>password</i> salah maka tidak bisa masuk <i>home admin</i> akan kembali ke halaman <i>login</i> dan menampilkan login gagal 2. Jika <i>username</i> dan <i>password</i> benar maka ke tampilan <i>home admin</i> _
<i>Post Condition</i>	Admin masuk ke halaman home admin

Tabel 3.5 Spesifikasi Use Case Menambah Perangkat

<i>Use Case Name</i>	Menambah Perangkat
<i>Actor</i>	Admin
<i>Brief Description</i>	<i>Use Case</i> digunakan untuk melakukan <i>input</i> data perangkat
<i>Basic Flow</i>	1. Admin memilih data perangkat 2. Admin melakukan <i>input</i> atau menambahkan data perangkat 3. Untuk <i>edit</i> data perangkat admin memilih <i>edit</i> 4. Untuk tambah data perangkat admin memilih tambah admin 5. Untuk hapus data perangkat admin memilih hapus
<i>Post Condition</i>	Admin melakukan input data perangkat

Tabel 3.6 Spesifikasi Use Case Menambahkan Alert

<i>Use Case Name</i>	Menambah <i>Alert</i>
<i>Actor</i>	Admin
<i>Brief Description</i>	<i>Use Case</i> digunakan untuk melakukan <i>input</i> data <i>alert</i>

<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Admin memilih data <i>alert</i> 2. Admin melakukan <i>input</i> atau menambahkan data <i>alert</i> 3. Untuk <i>edit</i> data <i>alert</i> admin memilih <i>edit</i> 4. Untuk tambah data <i>alert</i> admin memilih tambah <i>alert</i> 5. Untuk hapus data <i>alert</i> admin memilih hapus
<i>Post Condition</i>	Admin melakukan input data <i>alert</i>

Tabel 3.7 Spesifikasi Use Case Login Tim Support

<i>Use Case Name</i>	Menambah <i>Alert</i>
<i>Actor</i>	Tim <i>Support</i>
<i>Brief Description</i>	<i>Use Case</i> digunakan untuk <i>login</i> , sebagai pintu utama.
<i>Basic Flow</i>	Tim <i>Support</i> mengisi <i>username</i> dan <i>password</i> pada halaman <i>login</i>
<i>Alternate flow</i>	<ol style="list-style-type: none"> 1. Jika <i>username</i> atau <i>password</i> salah maka tidak bisa masuk <i>dashboard</i> akan kembali ke halaman <i>login</i> dan menampilkan <i>login</i> gagal. 2. Jika <i>username</i> dan <i>password</i> benar maka ke tampilan <i>dashboard</i>
<i>Post Condition</i>	Admin masuk ke halaman <i>dashboard</i>

Tabel 3.8 Spesifikasi Use Case Monitor Perangkat

<i>Use Case Name</i>	Monitor Perangkat
<i>Actor</i>	Tim <i>Support</i>
<i>Brief Description</i>	<i>Use Case</i> digunakan untuk memonitor atau lihat perangkat
<i>Basic Flow</i>	Pengguna memilih grafik
<i>Alternate flow</i>	<ol style="list-style-type: none"> 1. Tim <i>Support</i> memilih grup 2. Tim <i>Support</i> memilih <i>host</i>/perangkat 3. Tampil grafik
<i>Post Condition</i>	Pengguna memilih grafik

Tabel 3.9 Spesifikasi Use Case Lihat Alert

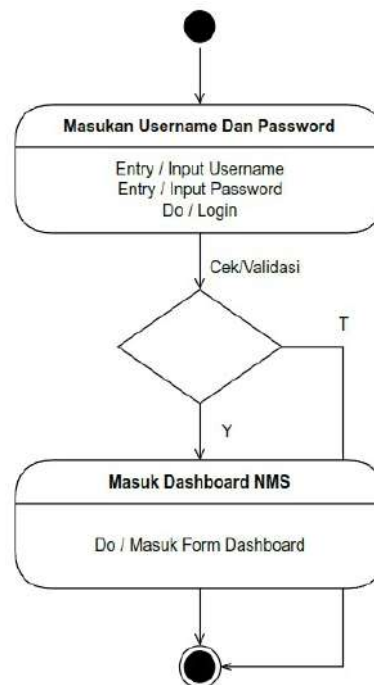
<i>Use Case Name</i>	Monitor Perangkat
<i>Actor</i>	Tim <i>Support</i>

<i>Brief Description</i>	<i>Use Case</i> ini digunakan untuk mendapatkan notifikasi <i>alert</i>
<i>Basic Flow</i>	Tim <i>support</i> mendapat notifikasi <i>alert</i>
<i>Alternate flow</i>	1. Perangkat dalam keadaan masalah 2. Notifikasi dikirim 3. Tim <i>Support</i> mendapat notifikasi dari <i>email</i> dan <i>telegram</i>
<i>Post Condition</i>	Tim <i>Support</i> mendapat notifikasi <i>alert</i> dari sistem

3.3.5 Activity Diagram

Aktivitas diagram menunjukkan aliran kerja (workflow) atau aktivitas dari sistem atau proses bisnis.

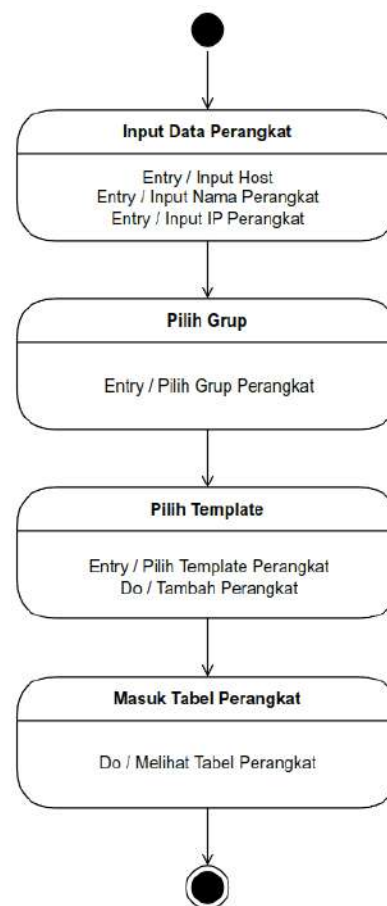
1. Activity Diagram Login (Admin, Tim Support)



Gambar 3.6 Activity Diagram Login

Pada *activity* diagram ini, menggambarkan proses login dimana tim *support* atau admin mengirimkan input *username* dan *password* yang akan divalidasi oleh *Database* NMS, apabila *username* dan *password* tersebut benar maka sistem akan menampilkan *form dashboard*, apabila salah maka sistem akan memunculkan pesan *error*.

2. Activity Diagram Menambahkan Perangkat (Admin)

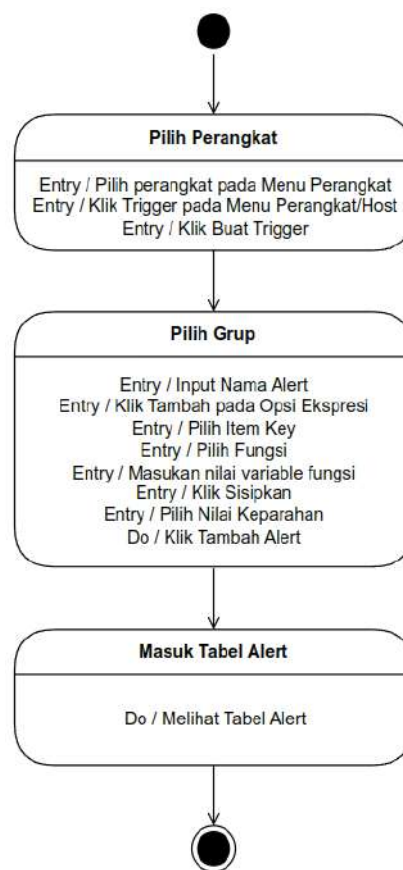


Gambar 3.7 Activity Diagram Menambahkan Perangkat

Pada *activity* diagram ini, menggambarkan proses menambahkan perangkat, dimana Admin masuk *menu host*, kemudian melakukan *input* nama perangkat atau *Host*,

Nama terlihat, *Input* IP perangkat atau *Host*, lalu Admin juga diharuskan untuk memilih grup dan *template* untuk perangkat tersebut, Setelah data terinput lengkap Admin bisa melakukan klik pada tombol tambah perangkat, Apabila data sudah terinput dengan benar data akan tersimpan di *database*, namun apabila salah akan terdapat pesan *error*.

3. Activity Diagram Menambahkan Alert (Admin)

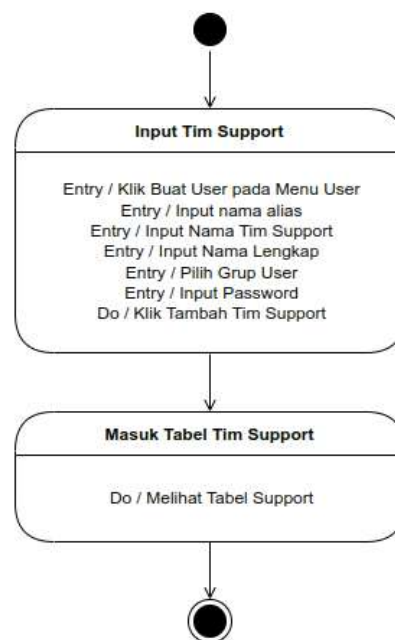


Gambar 3.8 Activity Diagram Menambahkan Alert

Pada *activity* diagram ini, menggambarkan proses menambahkan *alert*, dimana Admin masuk menu *host* atau perangkat, lalu melakukan klik pada tombol buat trigger, kemudian melakukan *input* nama *alert*, lalu klik tambah pada opsi ekspresi,

kemudian pilih *item key* dan pilih fungsi, serta masukan nolai *variable* fungsi, setelah itu klik sisipkan dan kemudian pilih nilai keparahan, setelah data terinput lengkap admin bisa melakukan klik pada tombol tambah perangkat, apabila data sudah terinput dengan benar data akan tersimpan di *database*, namun apabila salah akan terdapat pesan *error*.

4. Activity Diagram Menambahkan Tim Support (Admin)

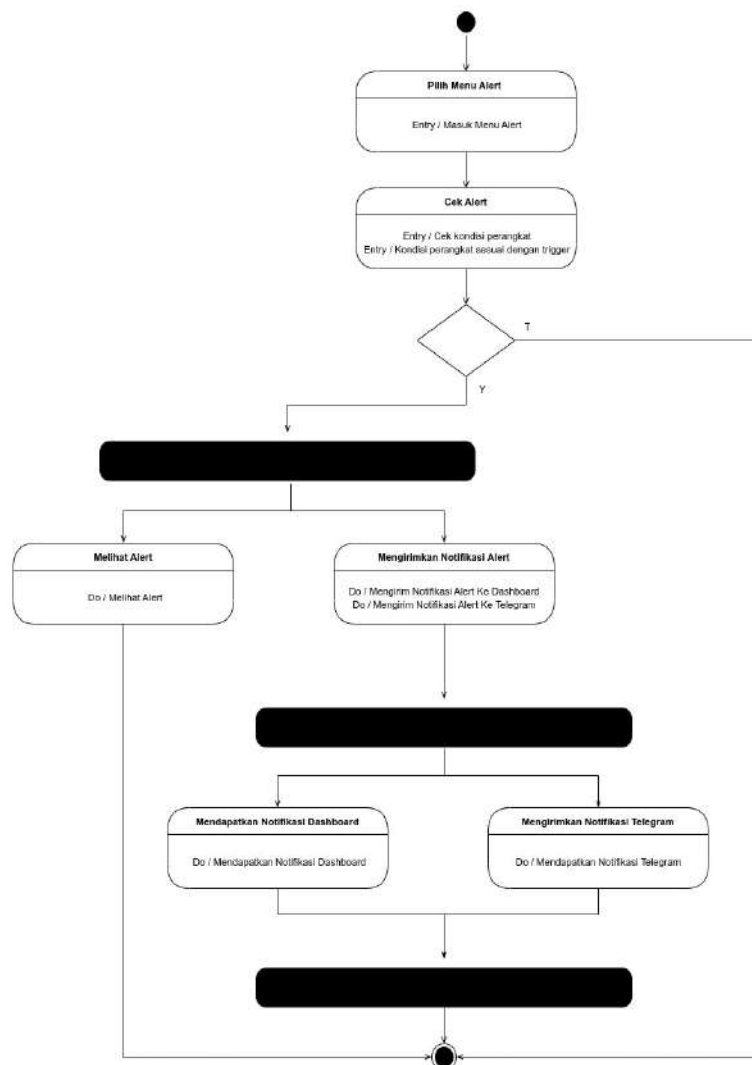


Gambar 3.9 Activity Diagram
Input Menambahkan Tim
Support

Pada *activity* diagram ini, menggambarkan proses menambahkan data pengguna, dimana Admin masuk menu Administrasi, lalu melakukan klik pada menu user, kemudian melakukan input nama *alert*, lalu klik tambah pada opsi ekspresi, kemudian pilih *item key* dan pilih fungsi, serta masukan nilai *variable* fungsi, setelah itu klik sisipkan dan kemudian pilih nilai keparahan, setelah data terinput lengkap admin bisa

melakukan klik pada tombol tambah perangkat, apabila data sudah terinput dengan benar data akan tersimpan di *database*, namun apabila salah akan terdapat *error*.

5. Activity Diagram Melihat Alert (Tim Support)

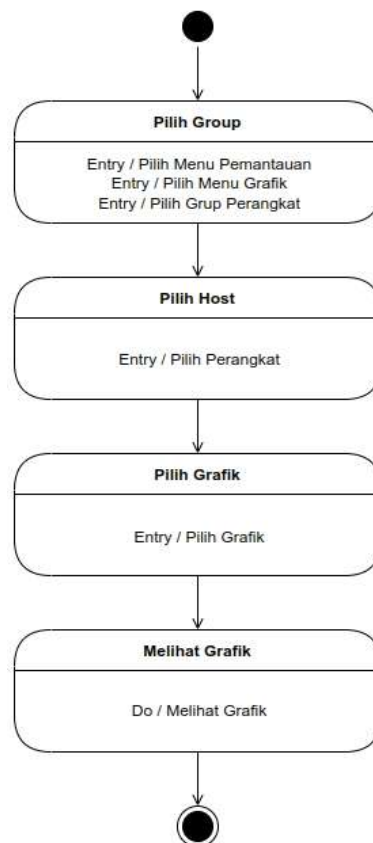


Gambar 3.10 Activity Diagram Input Melihat Alert

Pada *activity* diagram ini, menggambarkan proses melihat *alert*, dengan cara pengguna ataupun admin masuk ke menu *alert*, kemudian NMS melakukan cek pada perangkat yang terhubung melalui agen SNMP atau Zabbix, apabila terdapat masalah

pada perangkat tersebut, maka NMS akan menampilkan *alert* pada menu *alert* dan juga akan mengirimkan notifikasi melalui *dashboard* dan *telegram*.

6. Activity Diagram Monitor Perangkat (Tim Support)

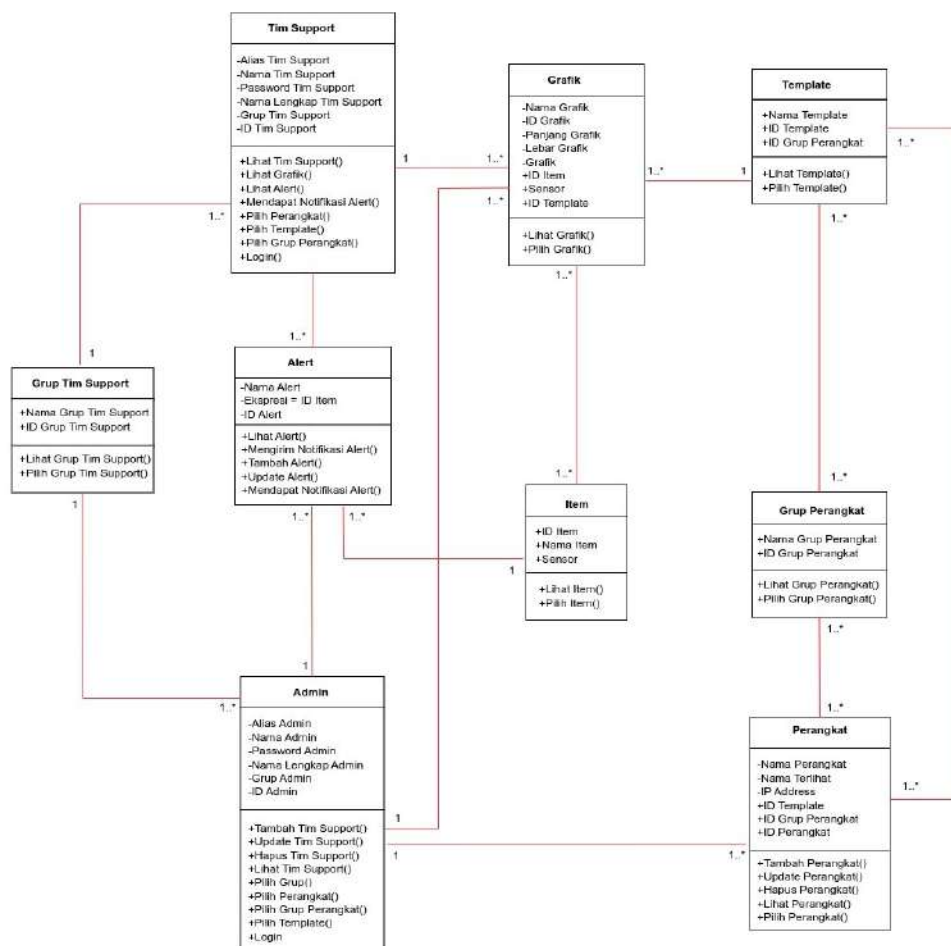


Gambar 3.11 Activity Diagram Monitor Perangkat

Pada *activity* diagram ini, menggambarkan proses monitor perangkat, dengan cara masuk ke menu pemantauan grafik, pengguna ataupun admin memilih grup, perangkat, dan grafik. Setelah dipilih grafik akan tampil.

3.3.6 Class Diagram

Diagram kelas, juga dikenal sebagai diagram kelas, adalah proses menggambarkan struktur sistem dari sudut pandang penjelasan kelas yang akan dibuat untuk membangun sistem kelas, serta atribut, metode, dan kinerjanya. Ini adalah kelas diagram sistem pemantauan jaringan :



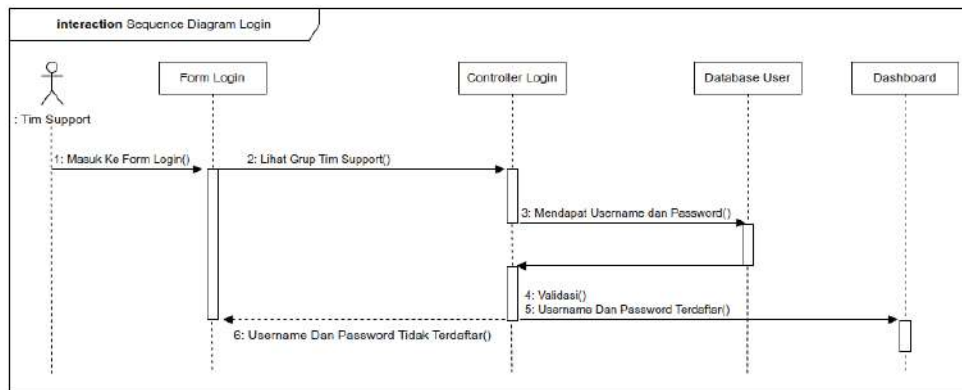
Gambar 3.12 Class Diagram Network Monitoring System

3.3.7 Sequence Diagram

Sequence Diagram menjelaskan bagaimana suatu operasi dijalankan, apa yang dikirim, dan kapan dijalankan. *Sequence Diagram* menampilkan gambaran tentang

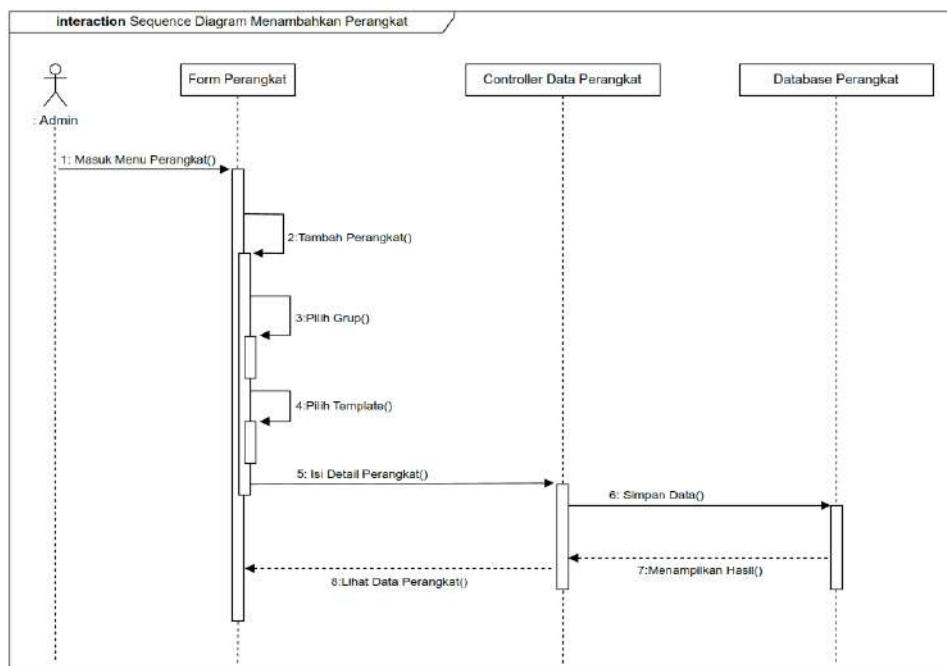
interaksi objek pada sistem dengan menjelaskan kinerja objek dan pesan yang dikirim dan diterima antara objek. Sequence Diagram untuk Sistem Pengawasan Jaringan ditunjukkan di bawah ini:

1. Sequence Diagram Login



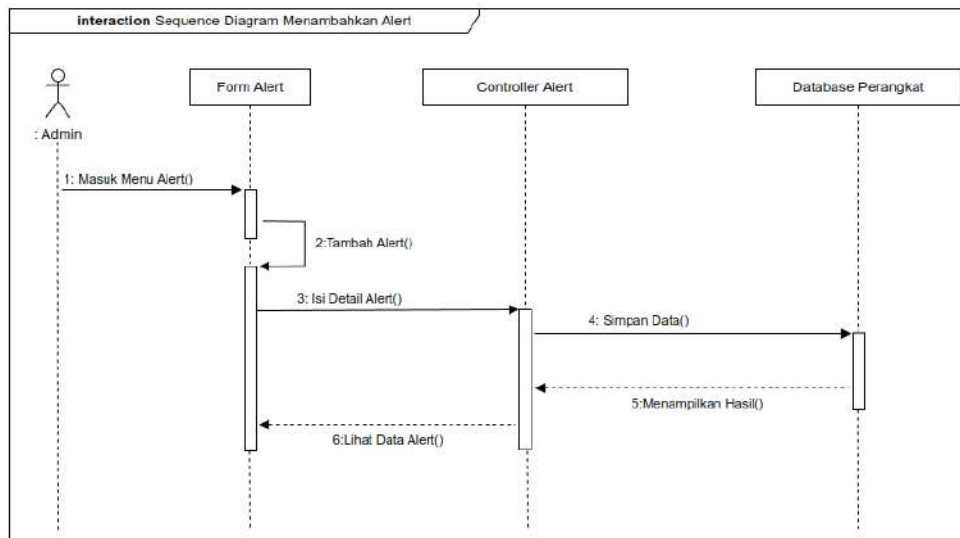
Gambar 3.13 Sequence Diagram Login

2. Sequence Diagram Menambah Perangkat



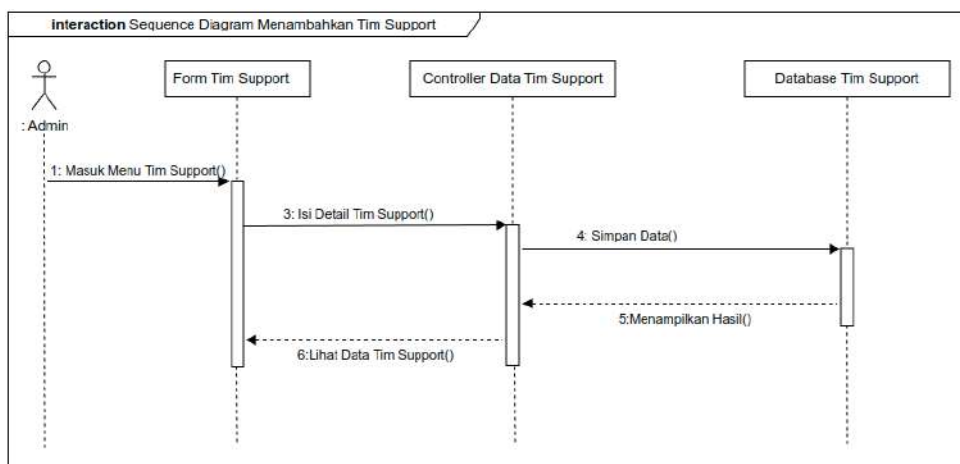
Gambar 3.14 Sequence Diagram Menambahkan Perangkat

3. Sequence Diagram Menambahkan Alert



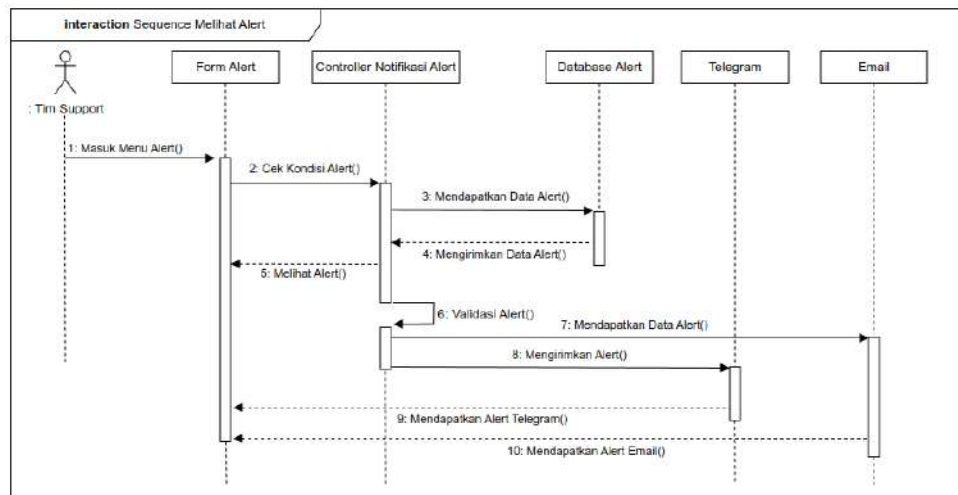
Gambar 3.15 Sequence Diagram Menambahkan Alert

4. Sequence Diagram Menambahkan Tim Support



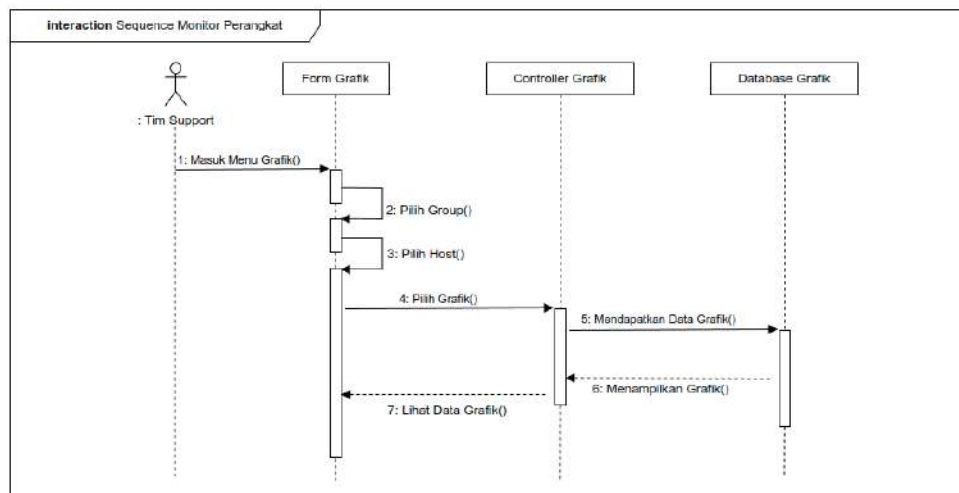
Gambar 3.16 Sequence Diagram Menambahkan Tim Support

5. Sequence Diagram Melihat Alert



Gambar 3.17 Sequence Diagram Melihat Alert

6. Sequence Diagram Monitor Perangkat



Gambar 3.18 Sequence Diagram Monitor Perangkat

3.3.8 Perancangan Database

Dalam setiap perancangan sebuah aplikasi/program pada umumnya diperlukan sebuah perancangan *database* sehingga untuk mendapatkan sistem pengolahan

database yang terdistribusi, berikut adalah tabel 3.10 sampai dengan tabel 3.21 yang merupakan struktur *database* yang digunakan dalam *Network Monitoring System* :

Tabel 3.10 Tabel User

No	Nama	Tipe Data	Length	Keterangan
1	<i>userid</i>	Bigint	20	<i>Primary Key</i> , id pengguna
2	<i>alias</i>	varchar	100	Alias atau <i>username</i>
3	<i>name</i>	varchar	100	Nama pengguna
4	<i>surname</i>	varchar	100	Nama lengkap pengguna
5	<i>passwd</i>	varchar	32	<i>Password</i> pengguna
6	<i>url</i>	varchar	255	url
7	<i>autologin</i>	Int	11	<i>Login</i> otomatis
8	<i>autologout</i>	varchar	32	<i>Logout</i> otomatis
9	<i>lang</i>	varchar	5	Bahasa
10	<i>refresh</i>	varchar	32	<i>Refresh</i>
11	<i>type</i>	Int	11	Tipe
12	<i>theme</i>	varchar	128	Tema
13	<i>attempt_failed</i>	Int	11	Gagal login
14	<i>attempt_ip</i>	varchar	39	Ip login
15	<i>attempt_clock</i>	Int	11	Waktu login
16	<i>row_per_page</i>	Int	11	Baris per halaman

Tabel 3.11 Tabel User Groups

No	Nama	Tipe Data	Length	Keterangan
1	<i>id</i>	Bigint	20	<i>Primary Key</i> , ID Grup Pengguna
2	<i>usrgrpid</i>	Bigint	20	Id Grup User
3	<i>userid</i>	Bigint	20	Id Pengguna

Tabel 3.12 Tabel Hosts

No	Nama	Tipe Data	Length	Keterangan
1	<i>hostid</i>	Bigint	20	<i>Primary Key</i> , IdPerangkat
2	<i>proxy_hostid</i>	Bigint	20	Id Proxy Perangkat
3	<i>host</i>	varchar	128	Perangkat
4	<i>status</i>	Int	11	Status Perangkat
5	<i>disable_until</i>	Int	11	Disable Perangkat
6	<i>error</i>	varchar	2048	Error text agen zabbix

7	<i>available</i>	Int	11	Zabbix agen tersedia
8	<i>errors_from</i>	Int	11	Waktu terakhir error pendeteksian via agen zabbix
9	<i>lastaccess</i>	Int	11	Akses terakhir
10	<i>ipmi_authtype</i>	Int	11	Tipe Authentikas IPMI
11	<i>ipmi_privilege</i>	Int	11	Hak IPMI
12	<i>ipmi_username</i>	varchar	16	Username IPMI
13	<i>ipmi_password</i>	varchar	20	Password IPMI
14	<i>ipmi_disable_until</i>	Int	11	Disable IPMI
15	<i>ipmi_available</i>	Int	11	IPMI tersedia
16	<i>snmp_disable_until</i>	Int	11	SNMP disable
17	<i>snmp_available</i>	Int	11	SNMP tersedia
18	<i>maintenanceid</i>	Bigint	20	ID Maintenance
19	<i>maintenance_status</i>	Int	11	Status Maintenance
20	<i>maintenance_type</i>	Int	11	Tipe Maintenance
21	<i>maintenance_form</i>	Int	11	Waktu Maintenance
22	<i>ipmi_errors_from</i>	Int	11	Waktu terakhir error pendeteksian via IPMI
23	<i>snmp_errors_from</i>	Int	11	Waktu terakhir error pendeteksian via SNMP
24	<i>ipmi_errors</i>	varchar	2048	Pesan error IPMI
25	<i>snmp_errors</i>	varchar	2048	Pesan error SNMP
26	<i>jmx_disable_until</i>	Int	11	JMX Disable
27	<i>jmx_available</i>	Int	11	JMX Tersedia
28	<i>jmx_errors_from</i>	Int	11	Waktu terakhir error pendeteksian via JMX
29	<i>jmx_error</i>	varchar	2048	Pesan error JMX
30	<i>name</i>	varchar	128	Nama
31	<i>flags</i>	Int	11	Asal Host
32	<i>templateid</i>	Bigint	20	Id Template
33	<i>description</i>	Text		Deskripsi
34	<i>tls_connect</i>	Int	11	SSL/TLS Konek ke perangkat
35	<i>tls_accept</i>	Int	11	SSL/TLS Konek dari perangkat
36	<i>tls_issuer</i>	varchar	1024	Isu Sertifikat
37	<i>tls_subject</i>	varchar	1024	Subjek Sertifikat
38	<i>tls_psk_identity</i>	varchar	128	Identitas password SSL/TLS
39	<i>tls_psk</i>	varchar	512	Password SSL/TLS
40	<i>proxy_address</i>	varchar	255	
41	<i>auto_compress</i>	Int	11	

Tabel 3.13 Tabel Host Group

No	Nama	Tipe Data	Length	Keterangan
1	<i>Hostgroupid</i>	Bigint	20	<i>Primary Key, ID Grup Perangkat</i>
2	<i>Hostid</i>	Bigint	20	Id Perangkat
3	<i>Groupid</i>	Bigint	20	Id Grup

Tabel 3.14 Tabel Interface

No	Nama	Tipe Data	Length	Keterangan
1	<i>Interfaceid</i>	Bigint	20	<i>Primary Key, Id Interface</i>
2	<i>Hostid</i>	Bigint	20	Id Perangkat
3	<i>Main</i>	Int	11	Perangkat <i>Default</i>
4	<i>Type</i>	Int	11	Tipe
5	<i>Useip</i>	Int	11	Koneksi via IP
6	<i>IP</i>	varchar	64	<i>IP Address</i>
7	<i>DNS</i>	varchar	64	DNS
8	<i>Port</i>	varchar	64	Port
9	<i>Bulk</i>	Int	11	<i>Bulk Request SNMP</i>

Tabel 3.15 Tabel Event

No	Nama	Tipe Data	Length	Keterangan
1	<i>Eventid</i>	Bigint	20	<i>Primary Key, Id Event</i>
2	<i>Source</i>	Int	11	Tipe pembuatan event
3	<i>Object</i>	Int	11	Tipe Objek
4	<i>Objectid</i>	Bigint	20	Id Objek
5	<i>Clock</i>	Int	11	Waktu Event
6	<i>Value</i>	Int	11	Nilai
7	<i>acknowledged</i>	Int	11	ACK dari event
8	<i>Ns</i>	Int	11	<i>Nanoseconds</i>

Tabel 3.16 Tabel Graph

No	Nama	Tipe Data	Length	Keterangan
1	Graphid	Bigint	20	<i>Primary Key, id Grafik</i>
2	<i>Name</i>	Varchar	128	Nama Grafik
3	<i>Width</i>	Int	11	Lebar Grafik
4	<i>Height</i>	Int	11	Tinggi Grafik
5	<i>Yaxismin</i>	Double	16,4	Nilai minimal Yaxis

6	<i>Yaxismax</i>	Double	16,4	Nilai maximal Yaxis
7	<i>Templateid</i>	Bigint	20	Id Template
8	<i>show work peroid</i>	Int	11	Menampilkan periode
9	<i>show triggers</i>	Int	11	Menampilkan <i>trigger</i>
10	<i>Graphtype</i>	Int	11	Tipe Grafik
11	<i>show legend</i>	Int	11	Menampilkan <i>legend</i>
12	<i>show 3d</i>	Int	11	Menampilkan 3d
13	<i>percent left</i>	Double	16,4	Persentase kiri
14	<i>percent right</i>	Double	16,4	Persentase kanan
15	<i>ymin type</i>	Int	11	Tipe Y minimal
16	<i>ymax type</i>	Int	11	Tipe Y maximal
17	<i>ymin itemid</i>	Bigint	20	Item id Y minimal
18	<i>ymax itemid</i>	Bigint	20	Item id Y maximal
19	<i>Flags</i>	Int	11	Asal grafik

Tabel 3.17 Tabel Items

No	Nama	Tipe Data	Length	Keterangan
1	<i>Itemid</i>	Bigint	20	<i>Primary Key</i> , Id item
2	<i>type</i>	Int	11	Tipe Item
3	<i>snmp community</i>	varchar	64	SNMP Community
4	<i>snmp old</i>	varchar	512	Oid SNMP
5	<i>hostid</i>	Bigint	20	Id Perangkat
6	<i>name</i>	varchar	255	Nama Item
7	<i>key</i>	varchar	255	Sensor Item
8	<i>delay</i>	varchar	1024	Delay
9	<i>history</i>	varchar	255	Data History
10	<i>trends</i>	varchar	255	Tren
11	<i>status</i>	Int	11	Status
12	<i>value type</i>	Int	11	Tipe Nilai
13	<i>trapper hosts</i>	varchar	255	Perangkat Trapper
14	<i>units</i>	varchar	255	Unit
15	<i>snmpv3_securityname</i>	varchar	64	Nama sekuriti snmp versi 3
16	<i>snmpv3_securitylevel</i>	Int	11	Sekuriti level snmp versi 3
17	<i>snmpv3_authpassphrase</i>	varchar	64	Autentikasi snmp versi 3
18	<i>snmpv3_privpassphrase</i>	varchar	64	Password snmp versi 3
19	<i>formula</i>	varchar	255	Formula
20	<i>logtimefmt</i>	varchar	64	Formula waktu log
21	<i>templateid</i>	Bigint	20	Id Template
22	<i>valuemapid</i>	Bigint	20	Nilai map id

23	<i>params</i>	mediumtext		parameter
24	<i>ipmi_sensor</i>	varchar	128	Sensor IPMI
25	<i>authtype</i>	Int	11	Tipe autentikasi
26	<i>username</i>	varchar	64	Username
27	<i>password</i>	varchar	64	Password
28	<i>publickey</i>	varchar	64	Kunci publik
29	<i>privatekey</i>	varchar	64	Kunci private
30	<i>flags</i>	Int	11	Asal Item
31	<i>interfaceid</i>	Bigint	20	Id Interface
32	<i>port</i>	varchar	64	Port
33	<i>description</i>	mediumtext		Deskripsi
34	<i>inventory_link</i>	Int	11	Link Inventory
35	<i>lifetime</i>	varchar	255	Lifetime
36	<i>snmpv3_authprotocol</i>	Int	11	Protokol autentikasi snmp ver 3
37	<i>snmpv3_privprotocol</i>	Int	11	protokol private snmp ver 3
38	<i>snmpv3_contextname</i>	varchar	255	
39	<i>evaltype</i>	Int	11	Tipe eval
40	<i>jmx_endpoint</i>	varchar	255	
41	<i>master_itemid</i>	Bigint	20	
42	<i>timeout</i>	varchar	255	
43	<i>url</i>	varchar	2048	
44	<i>query_fields</i>	varchar	2048	
45	<i>posts</i>	mediumtext		
46	<i>status_codes</i>	varchar	255	
47	<i>follow_redirects</i>	Int	11	
48	<i>post_type</i>	Int	11	
49	<i>http_proxy</i>	varchar	255	
50	<i>headers</i>	mediumtext		
51	<i>retrieve_mode</i>	Int	11	
52	<i>request_method</i>	Int	11	
53	<i>output_format</i>	Int	11	
54	<i>ssl_cert_file</i>	varchar	255	
55	<i>ssl_key_file</i>	varchar	255	
56	<i>ssl_key_password</i>	varchar	64	
57	<i>verify_peer</i>	Int	11	
58	<i>verify_host</i>	Int	11	
59	<i>allow_traps</i>	Int	11	

Tabel 3.18 Tabel Graph Items

No	Nama	Tipe Data	Length	Keterangan
1	<i>gitemid</i>	Bigint	20	<i>primary key</i> , id item grafik
2	<i>graphid</i>	Bigint	20	Id Grafik
3	<i>itemid</i>	Bigint	20	Id Item
4	<i>drawtype</i>	Int	11	Tipe gambar grafik
5	<i>sortorder</i>	Int	11	Posisi Item di Grafik
6	<i>color</i>	varchar	6	Warna
7	<i>yaxisside</i>	Int	11	Posisi skala item Y pada grafik
8	<i>calc_fnc</i>	Int	11	Nilai <i>Item</i>
9	<i>type</i>	Int	11	Tipe

Tabel 3.19 Tabel History

No	Nama	Tipe Data	Length	Keterangan
1	<i>itemid</i>	Bigint	20	Id item
2	<i>clock</i>	Int	11	Waktu
3	<i>value</i>	Double	16,4	Nilai
4	<i>ns</i>	Int	11	Nanoseconds

Tabel 3.20 Tabel Alert

No	Nama	Tipe Data	Length	Keterangan
1	<i>alertid</i>	Bigint	20	<i>Primary Key</i> , Id Alert
2	<i>actionid</i>	Bigint	20	Id Action
3	<i>eventid</i>	Bigint	20	Id Event
4	<i>userid</i>	Bigint	20	Id User
5	<i>clock</i>	Int	11	Waktu Event
6	<i>mediatypeid</i>	Bigint	20	Id Media Tipe
7	<i>sendto</i>	varchar	1024	Tujuan
8	<i>subject</i>	varchar	255	Subjek Pesan
9	<i>message</i>	mediumtext		Pesan
10	<i>status</i>	Int	11	Status pengiriman pesan
11	<i>retries</i>	Int	11	Jumlah pengiriman pesan
12	<i>error</i>	varchar	2048	Pesan error
13	<i>esc_step</i>	Int	11	Eskalasi setelah terjadi alert
14	<i>alerttype</i>	Int	11	Tipe alert
15	<i>p_eventid</i>	Bigint	20	
16	<i>acknowledgeid</i>	Bigint	20	
17	<i>parameters</i>	mediumtext		

Tabel 3.21 Tabel Triggers

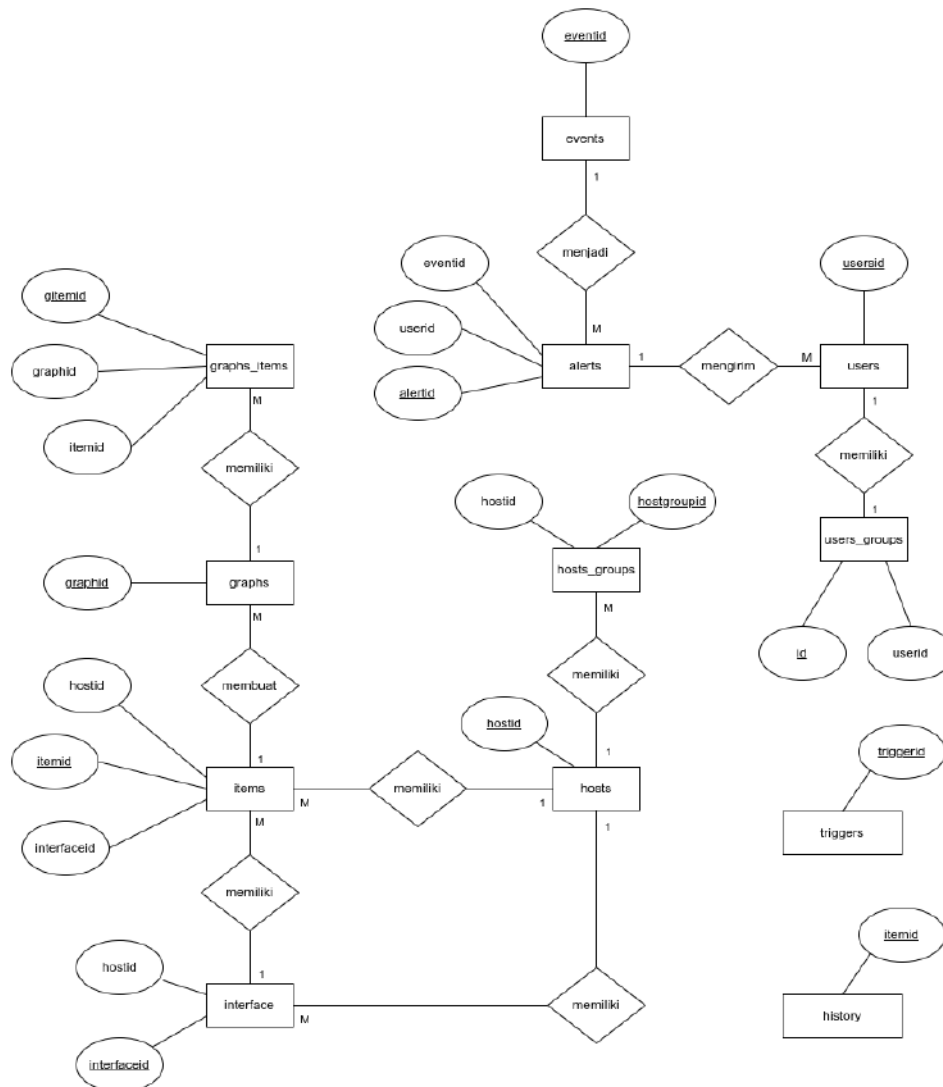
No	Nama	Tipe Data	Length	Keterangan
1	<i>triggerid</i>	Bigint	20	<i>Primary Key, Id trigger</i>
2	<i>expression</i>	varchar	2048	ekspresi
3	<i>description</i>	varchar	255	Deskripsi alert
4	<i>url</i>	varchar	255	url
5	<i>status</i>	Int	11	status
6	<i>value</i>	Int	11	nilai
7	<i>priority</i>	Int	11	prioritas
8	<i>lastchange</i>	Int	11	perubahan terakhir
9	<i>comments</i>	mediumtext		komentar
10	<i>error</i>	varchar	2048	Error
11	<i>templateid</i>	Bigint	20	Id Template
12	<i>type</i>	Int	11	Tipe
13	<i>state</i>	Int	11	State
14	<i>flags</i>	Int	11	Asal Trigger
15	<i>recovery mode</i>	Int	11	
16	<i>recovery expression</i>	varchar	2048	
17	<i>correlation mode</i>	Int	11	
18	<i>correlation tag</i>	varchar	255	
19	<i>manual close</i>	Int	11	
20	<i>opdata</i>	varchar	255	

3.3.9 Entity Relationship Diagram

Entity Relationship Diagram merupakan model data yang menggambarkan hubungan antara penyimpanan melalui notasi grafis dalam pemodelan data konseptual. Model adalah kumpulan teknik dan alat yang digunakan untuk menjelaskan hubungan dan konsistensi data. Model hubungan entitas dan model relasional membentuk model data.

Entity Relationship Diagram digunakan untuk membuat model data konseptual, mensimulasikan struktur dan hubungan antar data, dan mengimplementasikan basis data dengan DBMS. *Entity Relationship Diagram* ini memungkinkan kami untuk menguji model dengan mengabaikan langkah-langkah

yang harus dilakukan. Diagram hubungan entitas dapat membantu menjawab pertanyaan tentang data yang diperlukan dan hubungannya satu sama lain.



Gambar 3.19 Entity Relation Diagram NMS

Penjelasan Gambar 3.19 :

1. Satu *item* bisa membuat satu atau lebih grafik
2. Satu grafik bisa memiliki beberapa *item* grafik
3. Satu *user* atau pengguna hanya bisa masuk ke satu grup pengguna

4. Satu *event* bisa menjadi satu atau lebih *alert*
5. Satu *alert* bisa dikirim ke satu atau lebih pengguna
6. Satu perangkat memiliki beberapa item
7. Satu perangkat bisa masuk atau memiliki satu atau lebih grup perangkat
8. Satu perangkat memiliki beberapa *interface*.

3.3.10 Perancangan Antar Muka

Interface, juga dikenal sebagai antarmuka, adalah tampilan suatu program aplikasi yang berfungsi sebagai alat komunikasi dan memungkinkan program berkomunikasi dengan pengguna. Diharapkan desain antar muka yang akan dibangun dapat membuat antar muka mudah digunakan dan dipahami oleh pengguna. Perancangan antar muka dari NMS ini akan ditampilkan berdasarkan gambar 3.20 sampai dengan gambar 3.26 sebagai berikut :



The image shows a web browser window with the following elements:

- Browser tab: nms-IM
- Address bar: <https://nms.stmik-im.ac.id>
- Form content:
 - Placeholder: LOGO
 - Label: Username
 - Input field:
 - Label: Password
 - Input field:
 - Button: Masuk

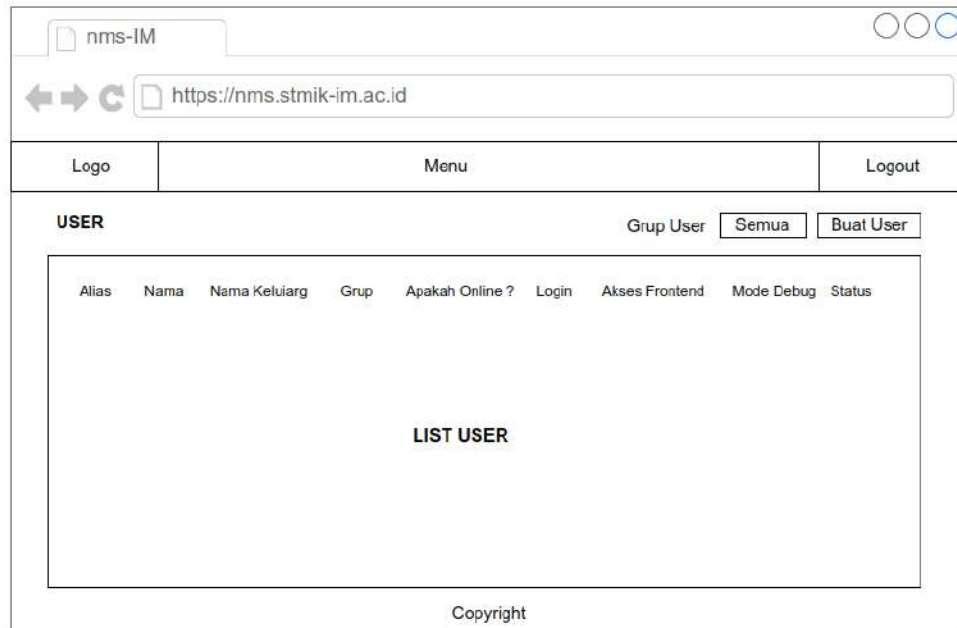
Gambar 3.20 Perencanaan Tampilan Login NMS



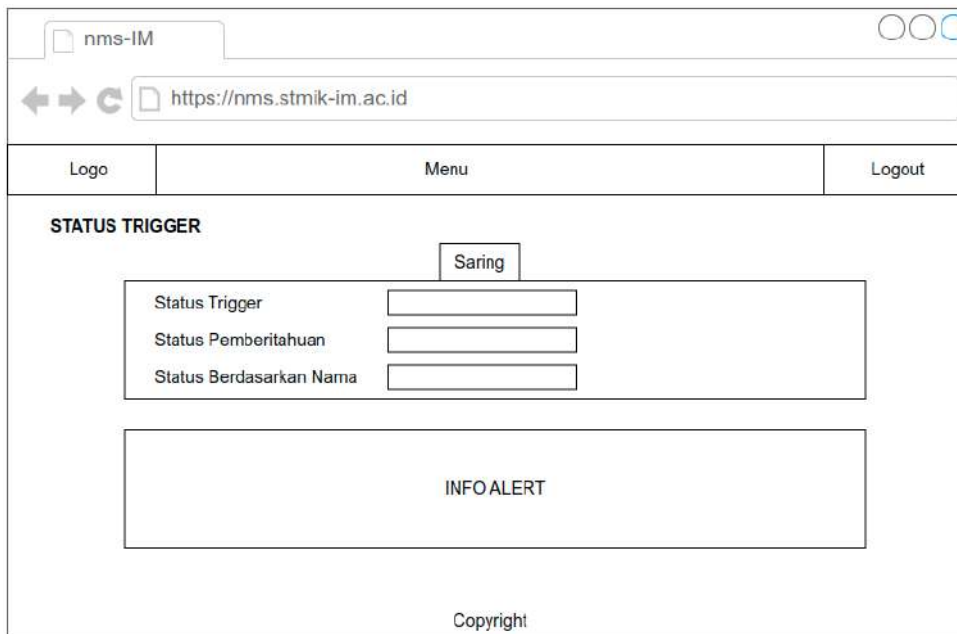
Gambar 3.21 Perencanaan Tampilan Dashboard



Gambar 3.22 Perencanaan Tampilan Grafik



Gambar 3.23 Perencanaan Tampilan List User



Gambar 3.24 Perencanaan Tampilan Alert

nms-IM

https://nms.stmik-im.ac.id

Logo Menu Logout

USER

Alias

Name

Grup

Password

Ulangi

Bahasa

Theme

Login otomatis

Logout otomatis

Segarkan

Baris Per Halaman

Uri setelah masuk

Copyright

Gambar 3.25 Perencanaan Tampilan Form Input User

nms-IM

https://nms.stmik-im.ac.id

Logo Menu Logout

HOST

Nama Host

Nama Terlihat

Grup

Agent Interface Hapus

SNMP Interface Hapus

Copyright

Gambar 3.26 Perencanaan Tampilan Form Input Perangkat

BAB IV

IMPLEMENTASI DAN UJI COBA

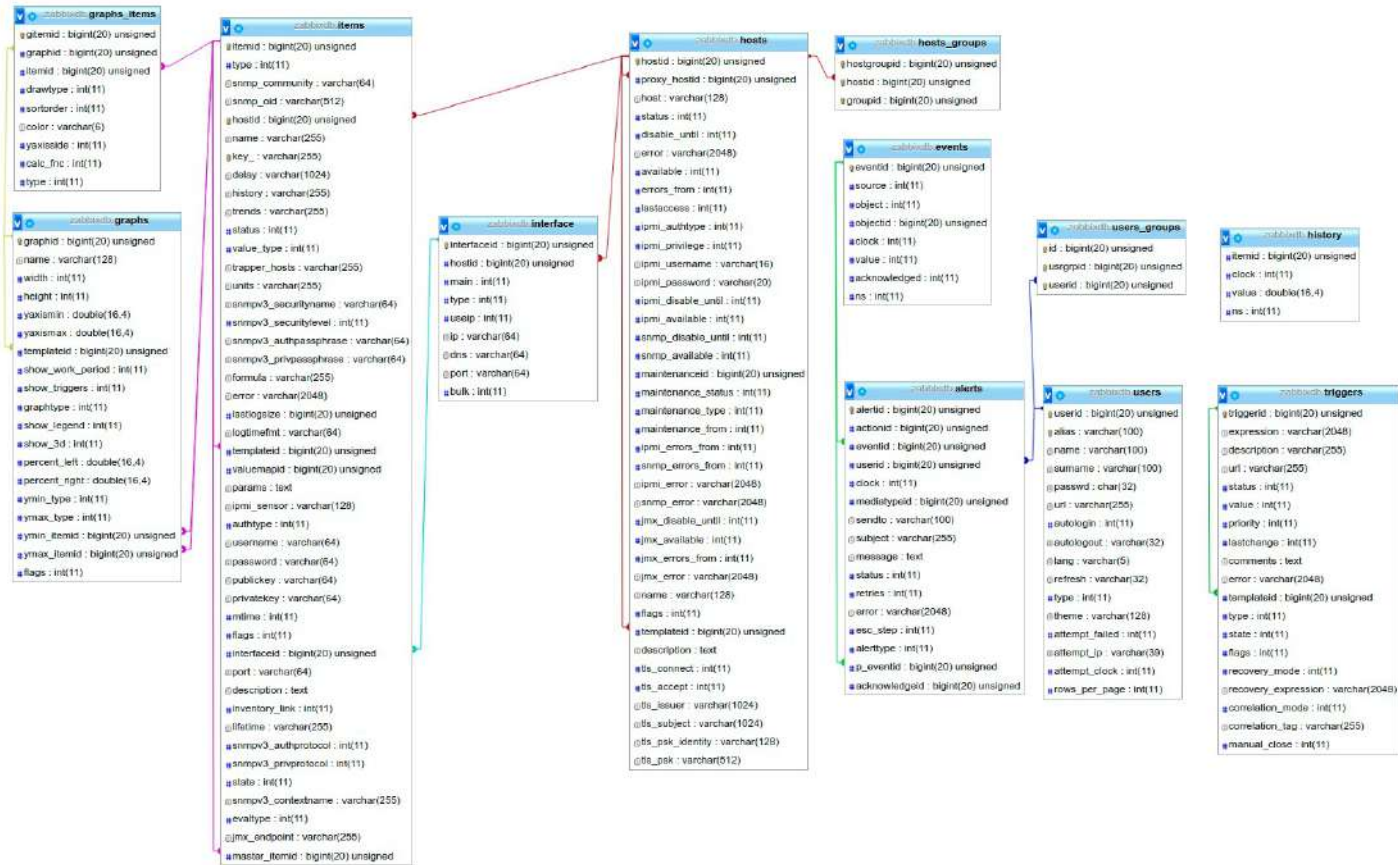
Pengkodean dari hasil analisis dan perancangan yang telah dibahas pada BAB III ANALISA DAN PERANCANGAN SISTEM dibahas dalam bab ini.

4.1 Implementasi Aplikasi

Untuk kampus Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri, diskusi tentang implementasi Sistem Pengawasan Jaringan akan dibagi menjadi 4 (empat) subtopik: database fisis, perangkat lunak, perangkat keras, dan antarmuka.

4.1.1 Physical Database

Gambar berikut menunjukkan implementasi database fisik untuk Sistem Pengawasan Jaringan di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri:



Gambar 4.1 Physical Database

4.1.2 Implementasi Perangkat Lunak

Hasil analisis menunjukkan bahwa aplikasi ini membutuhkan perangkat lunak yang diperlukan untuk menjalankannya. Berdasarkan temuan ini, keputusan untuk menerapkan perangkat lunak tersebut dibuat. Sebagai contoh, program berikut digunakan pada server :

1. Operating System Centos7 dengan kernel 3.10.0-693.el7.x86_64

```
[root@nms-im-new ~]# cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

Gambar 4.2 Operating System CentOS 7

```
[root@nms-im-new ~]# uname -r
3.10.0-693.el7.x86_64
```

Gambar 4.3 Kernel Operating System

2. Web Server menggunakan apache versi 2.4.6

```
Server version: Apache/2.4.6 (CentOS)
Server built: Apr  5 2023 17:18:30
Server's Module Magic Number: 20120211:24
Server loaded: APR 1.4.8, APR-UTIL 1.5.2
Compiled using: APR 1.4.8, APR-UTIL 1.5.2
Architecture: 64-bit
Server MPM: prefork
  threaded: no
  forked: yes (variable process count)
```

Gambar 4.4 Versi Web Server

3. PHP menggunakan versi 7.2.34

```
[root@nms-im-new ~]# php -v
PHP 7.2.34 (cli) (built: Feb 14 2023 16:46:38) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
```

Gambar 4.5 Versi PHP

4. Database menggunakan MariaDB versi 10.5.19

```
[root@nms-im-new ~]# mysql -v
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 778
Server version: 10.5.19-MariaDB MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

Gambar 4.6 Versi Database Server

Sedangkan implementasi pada sisi *client* diuji dengan menggunakan :

1. *Operating System* menggunakan Ubuntu 20.04 LTS
2. *Browser* menggunakan *firefox / chrome*

4.1.3 Implementasi Perangkat Keras

Untuk memastikan bahwa aplikasi dapat berjalan dengan baik, evaluasi kebutuhan perangkat keras menentukan penerapan perangkat keras. Berikut ini adalah perangkat keras yang digunakan pada server :

1. CPU menggunakan Intel Xeon E5450 3.00 Ghz

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 23
model name    : Intel(R) Xeon(R) CPU           E5450  @ 3.00GHz
stepping      : 6
microcode     : 0x60c
cpu MHz       : 2015.525
cache size    : 6144 KB
physical id   : 0
siblings      : 4
core id       : 0
cpu cores     : 4
```

Gambar 4.7 CPU Server

2. RAM 8GB

```
[root@nms-im-new ~]# free -h
              total        used          free      shared  buff/cache   available
Mem:           7.6G         341M         7.1G           9.1M         252M         7.1G
Swap:          3.9G           0B           3.9G
```

Gambar 4.8 RAM Server

3. Storage HDD 200GB

```
[root@nms-im-new ~]# fdisk -l | grep Disk
Disk /dev/sda: 214.7 GB, 214748364800 bytes, 419430400 sectors
Disk label type: dos
Disk identifier: 0x000146a0
Disk /dev/mapper/centos-root: 16.1 GB, 16106127360 bytes, 31457280 sectors
Disk /dev/mapper/centos-swap: 4160 MB, 4160749568 bytes, 8126464 sectors
Disk /dev/mapper/centos-var: 193.4 GB, 193399357440 bytes, 377733120 sectors
```

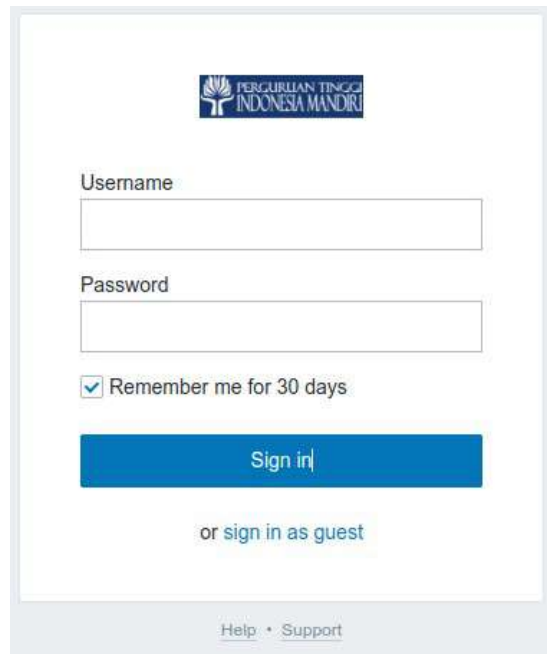
Gambar 4.9 Storage Server

4.2 Implementasi Antarmuka

Beberapa bagian tangkapan layar aplikasi menunjukkan hasil implementasi rancangan antarmuka yang dapat digunakan.

4.2.1 Implementasi Sistem Antarmuka

Untuk memperjelas bentuk dari perancangan *interface* berikut adalah gambar 4.10 sampai dengan gambar 4.20, tampilan implementasi sistem NMS yang telah dibuat mulai dari implementasi halaman login, implementasi *Dashboard Admin*, *Alert Admin*, *Alert Support*, Monitoring trafik admin, monitoring trafik support, halaman tabel user, halaman penambahan user, halaman penambahan perangkat, notifikasi telegram.



PERGURUAN TINGGI
INDONESIA MANDIRI

Username

Password

Remember me for 30 days

Sign in

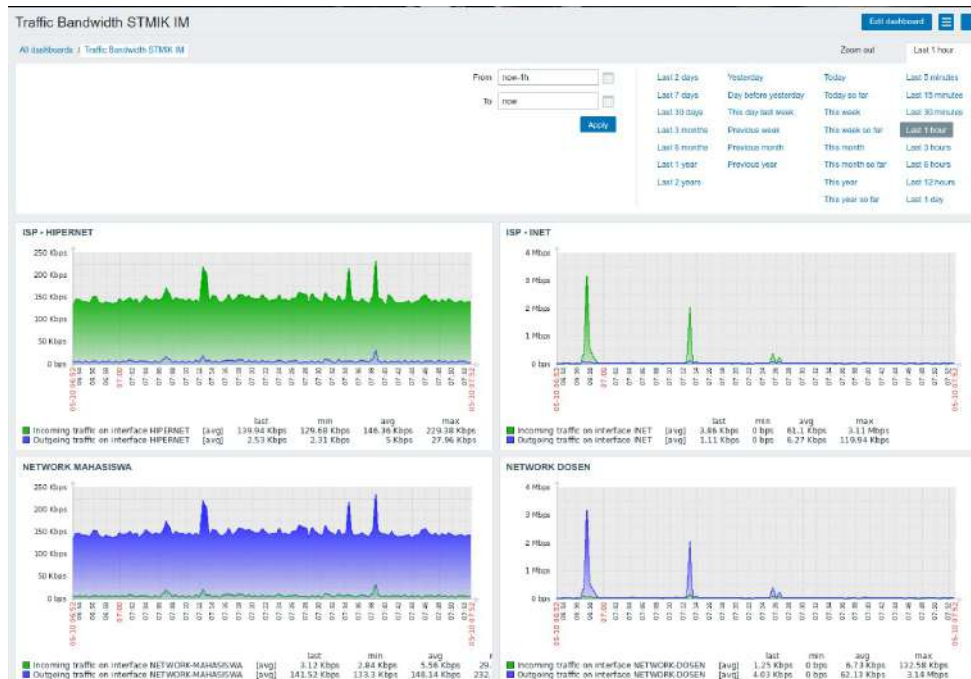
or sign in as guest

[Help](#) • [Support](#)

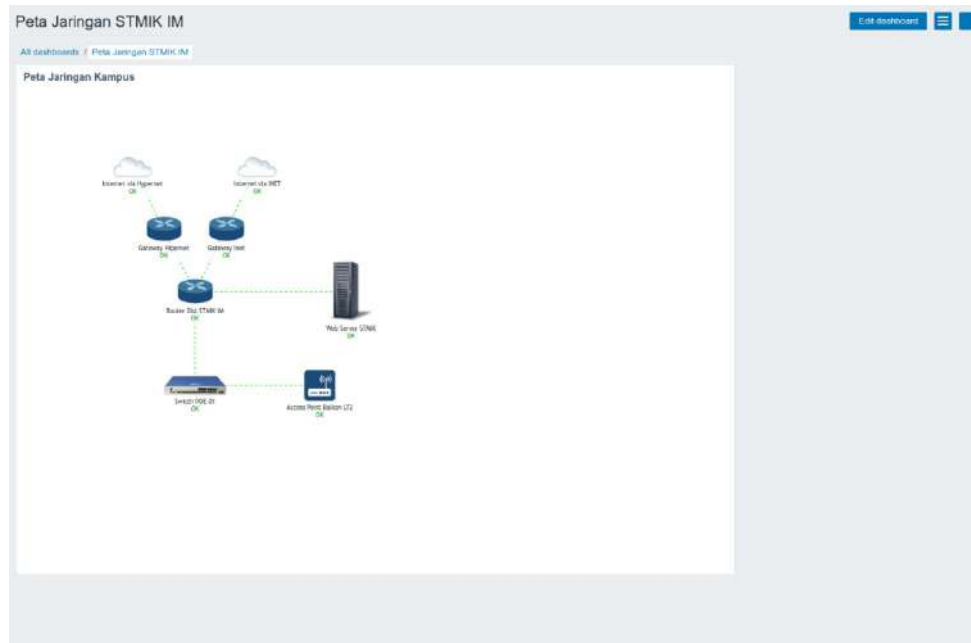
Gambar 4.10 Implementasi Antarmuka Halaman Login



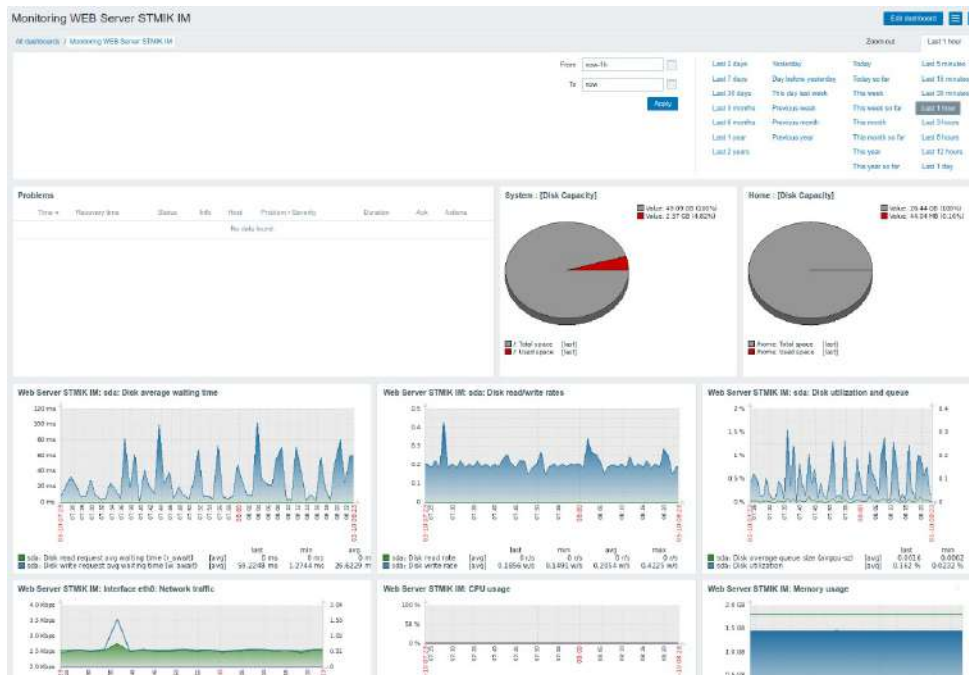
Gambar 4.11 Implementasi Antarmuka Halaman Dashboard



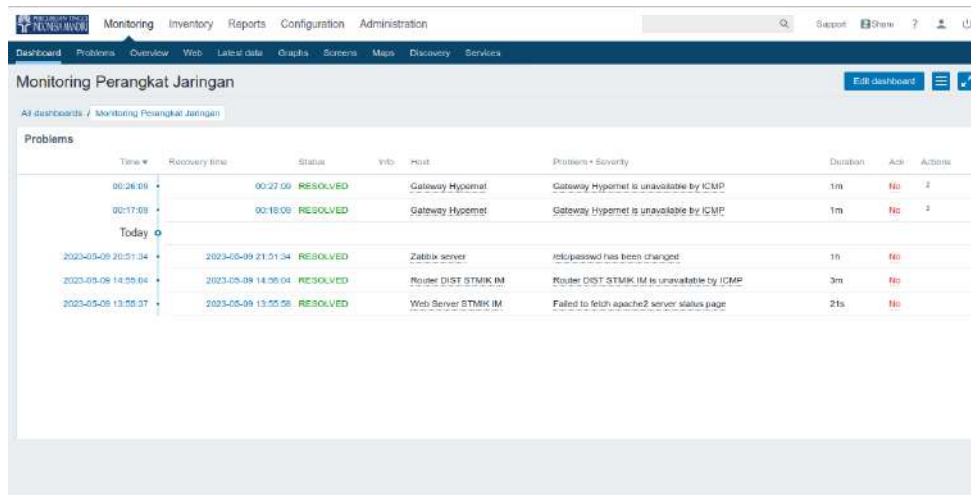
Gambar 4.12 Trafik Distribusi Bandwidth STMIK IM



Gambar 4.13 Peta Jaringan Kampus STMIK IM



Gambar 4.14 Monitoring Web Server STMIK IM



Gambar 4.15 Monitoring Alert Jaringan STMIK IM

The screenshot shows the 'Users' configuration page in the Zabbix Administration interface. The page has a navigation bar with 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. Below the navigation bar, there are tabs for 'General', 'Profiles', 'Authentication', 'User groups', 'Users', 'Media types', 'Scripts', and 'Queue'. The 'Users' tab is active.

The 'Users' form includes the following fields and options:

- * Alias: Text input field
- Name: Text input field
- Surname: Text input field
- * Groups: Search box with 'Type here to search' and a 'Select' button
- * Password: Password input field
- * Password (once again): Password input field
- Language: Dropdown menu (English (en_GB))
- Theme: Dropdown menu (System default)
- Auto-logout: Check box (unchecked)
- Auto-logout: Input field (10m)
- * Refresh: Input field (30s)
- * Rows per page: Input field (50)
- URL (after login): Text input field

Buttons at the bottom of the form are 'Add' and 'Cancel'.

Gambar 4.16 Implementasi Menambah Users

The screenshot shows the 'Users' table in the Zabbix Administration interface. The page has a navigation bar with 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. Below the navigation bar, there are tabs for 'General', 'Profiles', 'Authentication', 'User groups', 'Users', 'Media types', 'Scripts', and 'Queue'. The 'Users' tab is active.

The 'Users' table has a 'User group' dropdown menu set to 'All' and a 'Create user' button. The table has a 'Filter' button and a search bar. The table columns are: Alias, Name, Surname, User type, Groups, Is online?, Login, Frontend access, Debug mode, and Status.

Alias	Name	Surname	User type	Groups	Is online?	Login	Frontend access	Debug mode	Status
Admin	Zabbix	Administrator	Zabbix Super Admin	Zabbix administrators	Yes (2023-05-10 14:17:56)	OK	System default	Disabled	Enabled
guest			Zabbix User	Disabled, Guests	No	OK	Internal	Disabled	Disabled

Buttons at the bottom of the table are '0 selected', 'Unlock', and 'Delete'. The text 'Displaying 2 of 2 found.' is shown at the bottom right.

Gambar 4.17 Implementasi Melihat Tabel Users

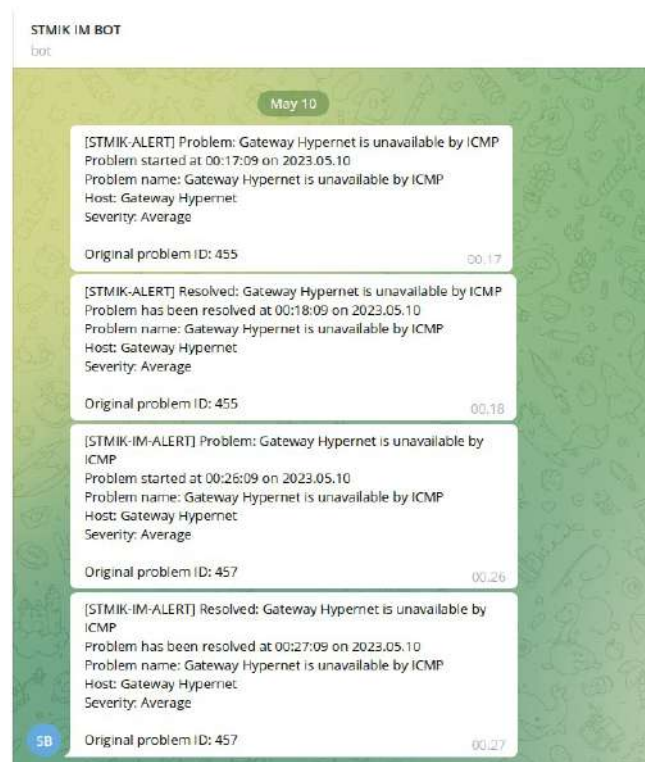
The screenshot shows the Zabbix 'Hosts' configuration page. The 'Host name' field is empty. The 'Visible name' field is also empty. The 'Groups' section has a search box and a 'Select' button. Below this, a note states 'At least one interface must exist.' The 'Agent interfaces' section contains a table with one entry: IP address '127.0.0.1', DNS name (empty), Connect to 'IP', Port '10050', and Default 'Nagios'. There are 'Add' and 'Remove' buttons for this entry. Below are sections for 'SNMP interfaces', 'JMX interfaces', and 'IPMI interfaces', each with an 'Add' button. A 'Description' text area is empty. At the bottom, there are dropdowns for 'Monitored by proxy' (set to 'no proxy') and 'Enabled' (checked), along with 'Add' and 'Cancel' buttons.

Gambar 4.18 Implementasi Menambah Perangkat

The screenshot shows the Zabbix 'Hosts' list page. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Hosts' section has a search bar and a 'Filter' button. Below the search bar are filters for 'Name', 'Templates', 'DNS', 'IP', and 'Port'. There are also filters for 'Monitored by' (Agent, Server, Proxy), 'Proxy', and 'Tags'. The main content is a table of hosts with columns for Name, Applications, Items, Triggers, Graphs, Discovery, Web Interface, Proxy, Templates, Status, Availability, Agent encryption, Info, and Tags. The table contains several rows of host configurations.

Name	Applications	Items	Triggers	Graphs	Discovery	Web Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
Access Point Dalkom LZ	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 192.168.226.251:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Gateway Hypernet	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 43.252.72.60:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Gateway INET	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 27.111.32.161:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Internet via Hypernet	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 8.8.4.4:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Internet via INET	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 9.9.9.9:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Router DIST STMIK IM	Applications 8	Items 147	Triggers 17	Graphs 18	Discovery 3	Web 10.0.111.80:161		Mikrotik-QUEUE, Template ICMP Ping, Template SNMP Mikrotik	Enabled	OK	OK	OK	NONE
Switch POE61	Applications 1	Items 3	Triggers 3	Graphs	Discovery	Web 192.168.226.250:10000		Template ICMP Ping	Enabled	OK	OK	OK	NONE
Web Server STMIK IM	Applications 18	Items 131	Triggers 33	Graphs 24	Discovery 3	Web 10.222.1.5:10000		Template App Apache2 Zabbix agent, Template App Zabbix Server, Template ICMP Ping, Template OS Linux by Zabbix agent /Flomate	Enabled	OK	OK	OK	NONE

Gambar 4.19 Implementasi Melihat Perangkat



Gambar 4.20 Implementasi Notifikasi Telegram

4.3 Pengujian Sistem

Metode black box untuk uji coba aplikasi memungkinkan perekayasa perangkat lunak membuat konfigurasi input yang memenuhi semua persyaratan fungsional program. Kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan struktur data atau akses ke basis data eksternal, kesalahan kinerja, dan kesalahan inisialisasi dan terminal adalah semua tujuan uji ini. Perekayasa sistem dapat menemukan kesalahan dalam kategori berikut dengan menggunakan metode pengujian black box:

1. Fungsi tidak benar atau hilang
2. Kesalahan antar muka

3. Kesalahan pada struktur data (Pengaksesan *database*)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan kinerja

Berikut adalah tampilan hasil pengujian blackbox testing yang ditampilkan pada tabel

4.1 sampai dengan tabel 4.5 :

Tabel 4.1 Hasil Pengujian Sistem Login

No	Kriteria Pengujian	Yang Diharapkan	Hasil Pengujian	Keterangan
1	<p>Pengujian fungsi <i>login</i> dengan <i>input username</i> dan <i>password</i> sesuai</p> <p>Aksi : Mengunjungi halaman <i>login</i> kemudian memberikan nilai pada kolom : <i>username</i> : Admin <i>password</i> : zabbix</p>	<p><i>Login</i> berhasil kemudian masuk ke halaman <i>dashboard NMS</i></p>	<p>Masuk ke halaman <i>Dashboard</i></p>	Berhasil
2	<p>Pengujian fungsi <i>login</i> dengan <i>input</i> parameter salah</p> <p>Aksi : Mengunjungi halaman <i>login</i> kemudian memberikan nilai pada kolom : <i>username</i> : admin-zabbix <i>password</i> : gagal</p>	<p><i>Login</i> gagal, kemudian menampilkan pesan <i>username</i> atau <i>password</i> tidak sesuai dan kembali ke halaman <i>login</i></p>	<p><i>Login</i> gagal dan kembali ke halaman <i>login</i></p>	Berhasil

3	<p>Pengujian fungsi <i>login</i> dengan <i>input</i> parameter kosong</p> <p>Aksi : Mengunjungi halaman <i>login</i> kemudian tidak memberikan nilai pada kolom <i>username</i> dan <i>password</i></p>	<p><i>Login</i> gagal, kemudian menampilkan pesan <i>username</i> atau <i>password</i> tidak sesuai dan kembali ke halaman <i>login</i></p>	<p><i>Login</i> gagal dan kembali ke halaman <i>login</i></p>	Berhasil
---	---	---	---	----------

Tabel 4.2 Hasil Pengujian Sistem Data Perangkat

No	Kriteria Pengujian	Yang Diharapkan	Hasil Pengujian	Keterangan
1	<p>Pengujian fungsi tambah data perangkat dengan <i>input</i> melengkapi <i>field</i> yang menjadi <i>mandatory</i>.</p> <p>Aksi : Mengunjungi halaman tambah data perangkat / <i>host</i> kemudian memasukan nilai sesuai dengan parameter pada field <i>hostname</i>, <i>groups</i>, <i>interfaces</i> <i>tools</i>, dengan nilai yang diberikan :</p> <p><i>Hostname</i> : Router STMIK IM</p>	<p>Data perangkat sukses disimpan ke <i>Database</i></p>	<p>Data perangkat sukses disimpan ke <i>Database</i></p>	Berhasil

	<p><i>Groups : Select Host Group</i></p> <p><i>Interface tool : Select SNMP Interfaces</i> masukan IP 10.0.11.80</p>			
2	<p>Pengujian fungsi tambah data perangkat dengan tidak mengisi salah satu <i>field</i> yang menjadi <i>mandatory</i>.</p> <p>Aksi : Mengunjungi halaman tambah data perangkat / <i>host</i> kemudian mengosongkan salah satu <i>field</i> yang menjadi <i>mandatory</i>.</p>	Data perangkat gagal Disimpan ke <i>Database</i> .	Data perangkat gagal disimpan ke <i>Database</i>	Berhasil
3	<p>Pengujian fungsi tambah data perangkat dengan <i>input</i> melengkapi <i>field</i> yang menjadi <i>mandatory</i> dengan membuat parameter yang tidak sesuai pada <i>field interfaces tools</i>.</p> <p>Aksi : Mengunjungi halaman tambah</p>	Data perangkat gagal Disimpan ke <i>Database</i> .	Data perangkat gagal Disimpan ke <i>Database</i> .	Berhasil

	<p>data perangkat / host kemudian mengisi salah satu <i>field interfaces tools</i> dengan parameter yang berbeda :</p> <p>parameter seharusnya :</p> <p>IP Address : 10.0.111.18 (format pengisian IP Address)</p> <p>pengetestan :</p> <p>IP Address : router.stmik-im.com (dalam bentuk <i>domain name</i>)</p>			
4	<p>Pengujian fungsi tambah data perangkat dengan <i>input select</i> lebih dari satu grup.</p> <p>Aksi : Mengunjungi halaman tambah perangkat / host dengan menambahkan lebih dari satu grup</p>	Data perangkat sukses disimpan ke <i>Database</i>	Data perangkat sukses disimpan ke <i>Database</i>	Berhasil
5	<p>Pengujian fungsi <i>edit</i> data perangkat dengan menambahkan <i>templates</i>.</p>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Berhasil

	<p>Aksi : Mengunjungi halaman tambah perangkat / <i>host</i> kemudian masuk ke menu <i>template</i> dan menambahkan data <i>template</i> lebih dari satu <i>template</i>.</p>			
6	<p>Pengujian fungsi <i>edit</i> data perangkat dengan menambahkan <i>grup</i>.</p> <p>Aksi : Mengunjungi halaman tambah perangkat / <i>host</i> kemudian menambahkan <i>grup</i>.</p>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Berhasil
4	<p>Pengujian fungsi <i>edit</i> data perangkat dengan mengganti <i>interfaces tools</i>.</p> <p>Aksi : Mengunjungi halaman tambah perangkat / <i>host</i> dengan mengganti <i>interface tools</i> dari <i>Agent Interfaces</i> ke <i>SNMP Interfaces</i></p>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Data perangkat sukses dirubah dan disimpan ke <i>Database</i>	Berhasil
5	Pengujian fungsi	Data perangkat	Data	Berhasil

	<p>hapus data perangkat.</p> <p>Aksi : Mengunjungi halaman tambah perangkat / <i>host</i> kemudian melakukan <i>checklist</i> pada daftar perangkat yang ada dengan cara hapus satu persatu atau hapus lebih dari satu</p>	<p>sukses di hapus dari <i>Database</i></p>	<p>perangkat sukses di hapus dari <i>Database</i></p>	
6	<p>Pengujian fungsi hapus data grup pada data perangkat.</p> <p>Aksi : Mengunjungi halaman data perangkat melakukan penghapusan semua grup pada salah satu data perangkat.</p>	<p>Data perangkat gagal dihapus dari <i>database</i> karena <i>field</i> grup pada setiap data perangkat tidak boleh dikosongkan</p>	<p>Data perangkat gagal dihapus dari <i>database</i> karena <i>field</i> grup pada setiap data perangkat tidak boleh dikosongkan</p>	Berhasil
7	<p>Pengujian fungsi hapus <i>interfaces tools</i> pada data perangkat.</p> <p>Aksi : Mengunjungi halaman data perangkat melakukan penghapusan semua <i>inteface</i></p>	<p>Data perangkat gagal di hapus dari <i>database</i> karena <i>field interfaces tools</i> pada setiap data perangkat tidak boleh dikosongkan.</p>	<p>Data perangkat gagal di hapus dari <i>database</i> karena <i>field interfaces tools</i> pada setiap data perangkat tidak boleh dikosongkan.</p>	Berhasil

	<i>tools</i>			
--	--------------	--	--	--

Tabel 4.3 Hasil Pengujian Sistem Alert

No	Kriteria Pengujian	Yang Diharapkan	Hasil Pengujian	Keterangan
1	<p>Pengujian fungsi tambah <i>alert</i> dengan melengkapi <i>field</i> yang menjadi <i>mandatory</i>.</p> <p>Aksi : Mengunjungi halaman tambah <i>alert</i> kemudian melengkapi <i>field</i> yang menjadi <i>mandatory</i>.</p>	Data <i>alert</i> sukses disimpan ke <i>Database</i> .	Data <i>alert</i> sukses disimpan ke <i>Database</i> .	Berhasil
2	<p>Pengujian fungsi tambah <i>alert</i> dengan tidak mengisi nama <i>alert</i>.</p> <p>Aksi : Mengunjungi halaman tambah <i>alert</i> mengisi semua <i>field</i> yang menjadi <i>mandatory</i> kecuali nama <i>alert</i>.</p>	Data <i>alert</i> tidak bisa disimpan ke <i>Database</i> karena <i>field</i> nama <i>alert</i> tidak diisi.	Data <i>alert</i> tidak bisa disimpan ke <i>Database</i> karena <i>field</i> nama <i>alert</i> tidak diisi.	Berhasil
3	<p>Pengujian fungsi <i>edit alert</i> lengkap dan sesuai dengan ketentuan</p> <p>Aksi : Mengunjungi halaman <i>alert</i></p>	Data <i>alert</i> sukses dirubah kemudian disimpan ke <i>Database</i> .	Data <i>alert</i> sukses dirubah kemudian disimpan ke <i>Database</i> .	Berhasil



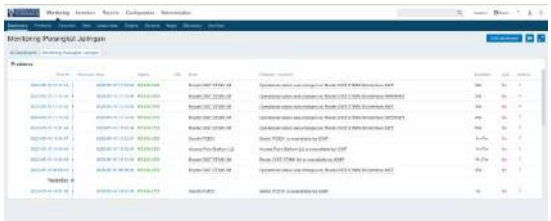

	kemudian melakukan perubahan pada data <i>alert</i> .			
4	Pengujian fungsi edit <i>alert</i> dengan menghapus nama <i>alert</i> . Aksi : Mengunjungi halaman <i>alert</i> kemudian menghapus nama <i>alert</i> .	Data <i>alert</i> gagal disimpan ke <i>Database</i> karena <i>field mandatory</i> tidak diisi.	Data <i>alert</i> gagal disimpan ke <i>Database</i> karena <i>field mandatory</i> tidak diisi.	Berhasil
5	Pengujian fungsi hapus <i>alert</i> . Aksi : Mengunjungi halaman <i>alert</i> kemudian melakukan penghapusan <i>alert</i>	Data <i>alert</i> sukses dihapus dari <i>Database</i>	Data <i>alert</i> sukses dihapus dari <i>Database</i>	Berhasil

Tabel 4.4 Hasil Pengujian Sistem User

No	Kriteria Pengujian	Yang Diharapkan	Hasil Pengujian	Keterangan
1	Pengujian fungsi tambah <i>user</i> . Aksi : Mengunjungi halaman <i>user</i> kemudian memasukan data <i>user</i> lengkap sesuai dengan ketentuan	Data <i>user</i> sukses disimpan ke <i>Database</i>	Data <i>user</i> sukses disimpan ke <i>Database</i>	Berhasil

2	<p>Pengujian fungsi tambah user.</p> <p>Aksi : Mengunjungi halaman <i>user</i> kemudian memasukan data <i>user</i> dengan tidak melengkapi <i>field mandatory</i>.</p>	<p>Data <i>User</i> gagal disimpan ke <i>Database</i> karena ada <i>field mandatory</i> kosong.</p>	<p>Data <i>User</i> gagal disimpan ke <i>Database</i> karena ada <i>field mandatory</i> kosong.</p>	Berhasil
3	<p>Pengujian fungsi <i>edit user</i>.</p> <p>Aksi : Mengunjungi halaman <i>user</i> kemudian mengubah grup, <i>password</i> sesuai dengan <i>validasi password</i>, dan juga menambahkan data nama alias</p>	<p>Data <i>user</i> berhasil dirubah dan disimpan ke <i>Database</i></p>	<p>Data <i>user</i> berhasil dirubah dan disimpan ke <i>Database</i></p>	Berhasil
4	<p>Pengujian fungsi hapus <i>user</i>.</p> <p>Aksi : Mengunjungi halaman <i>user</i> kemudian menghapus salah satu user yang ada pada daftar <i>users</i></p>	<p>Data <i>user</i> berhasil dihapus dari <i>Database</i></p>	<p>Data <i>user</i> berhasil dihapus dari <i>Database</i></p>	Berhasil

Tabel 4.5 Hasil Pengujian Sistem Pengguna

No	Data Masukan	Hasil Pengujian	Keterangan
1	Monitor Perangkat	<p>Grafik MRTG</p> 	<p>Sesuai</p>
		<p>Pemantauan Web Server</p> 	<p>Sesuai</p>
2	Melihat Alert	<p>Alert</p> 	<p>Sesuai</p>
		<p>Notifikasi Telegram</p> <p>Telegram (50343)</p> 	<p>Sesuai</p>

4.4 Pengujian NMS

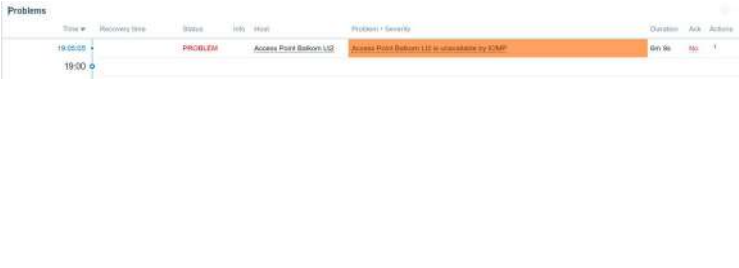
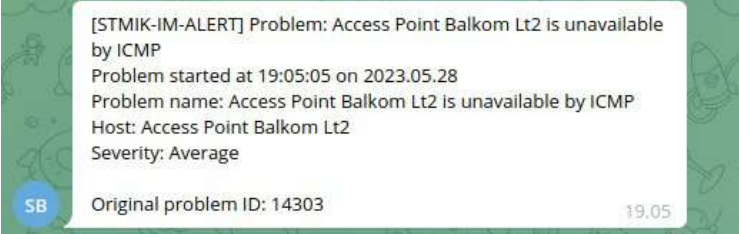
Perangkat router STMIK IM digunakan untuk menjalankan aktivitas manajemen SNMP selama pengujian ini. Di mana pengujian dilakukan melalui pengawasan terus-menerus untuk mengumpulkan data tentang kondisi host, sistem, dan aplikasi.

Tabel 4.5 merupakan hasil pengujian aplikasi manajemen SNMP terhadap *router* STMIK IM, dimana perangkat tersebut di-*manage* oleh NMS. NMS menginformasikan kondisi perangkat secara berkelanjutan dalam bentuk grafik atau dalam bentuk notifikasi informasi. Dimana laporan tersebut dapat dijadikan sebagai acuan untuk perbaikan dan peningkatan kualitas jaringan.


Dibawah ini adalah tabel 4.6 yang berisi dari proses fungsi kerja NMS dari pertama muncul gangguan sampai titik masalah ditemukan.


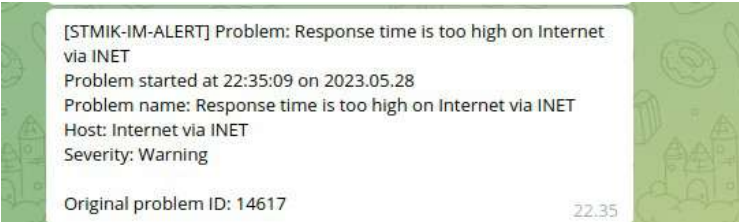
Tabel 4.6 Tabel Proses Fungsi Kerja NMS – Gangguan Access Point

No	Perangkat/ Host	Waktu Masalah Perangkat / Layanan	Aksi Notifikasi	Keterangan
1	Access Point Balkon Lt2	28 Mei 2023 pukul 19:03	Beberapa user (Dosen dan Mahasiswa) di area Balkon Lt 2 mengalami gangguan pada perangkat mereka.	Pukul 19:03 Kondisi user yang berada pada jangkauan atau terkoneksi ke Access Point yang berada di Balkon Lt 2 mengalami permasalahan pada Access Internet.
		28 Mei 2023 pukul 19:05		Pukul 19:05 Termonitor pada peta mapping jaringan terdapat alarm dengan tanda icon merah dan menjelaskan status Access Point Balkon Lt 2 dalam kondisi service ICMP yang tidak tersedia, artinya perangkat tersebut tidak bisa dijangkau oleh NMS dengan layanan ICMP.


	28 Mei 2023 pukul 19:05	<pre>ping -c 3 192.168.220.251; case \$? in (01) true;; *) false;; esac PING 192.168.220.251 (192.168.220.251) 56(84) bytes of data. --- 192.168.220.251 ping statistics --- 3 packets transmitted, 0 received, 100% packet loss, time 1955ms</pre>	<p>Pukul 19:05</p> <p>Tim support melakukan verifikasi dengan cara melakukan ping ke ip Access point tersebut dan hasilnya 100% packet Loss</p>
	28 Mei 2023 pukul 19:05		<p>Pukul 19:05</p> <p>Tim support mendapatkan info pada log perangkat, yang menjelaskan telah terjadi masalah pada perangkat Access Point Balkon Lt 2</p>
	28 Mei 2023 pukul 19:05		<p>Pukul 19:05</p> <p>Network Administrator yang sedang berada diluar kampus mendapatkan notifikasi di telegram yang berisi tentang informasi masalah pada perangkat Access Point Balkon Lt 2</p>

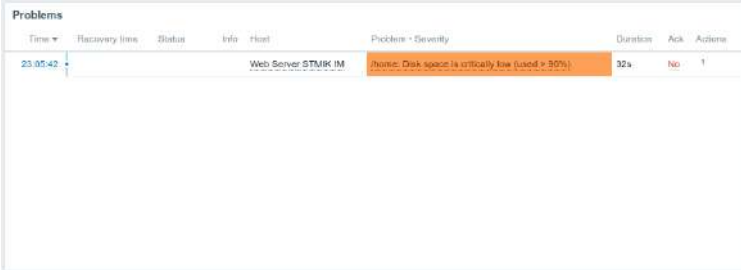

Tabel 4.7 Tabel Proses Fungsi NMS – Delay akses ke arah Internet

No	Perangkat/ Host	Waktu Masalah Perangkat / Layanan	Aksi Notifikasi	Keterangan
1	Layanan Internet via ISP INET	28 Mei 2023 pukul 22:34	User penggunaan internet (Dosen) mengalami masalah ketika akses ke internet, akses terasa sangat lambat.	Pukul 22:34 User pengguna (Dosen) merasakan akses ke internet bermasalah, tidak bisa melakukan aktifitas belajar mengajar sangat terganggu.
		28 Mei 2023 pukul 22:35		Pukul 22:35 Termonitor pada peta mapping jaringan terdapat alarm dengan tanda icon merah dan menjelaskan status kearah internet via INET mengalami high latency, artinya ada beberapa kendala yang menyebabkan hal tersebut. Diantaranya : <ol style="list-style-type: none"> 1. Full Traffic Bandwidth 2. Gangguan pada ISP

	28 Mei 2023 pukul 22:35	<pre>ping -c 10 -s 4096 9.9.9.9; case \$? in [01]) true;; *) false;; esac PING 9.9.9.9 (9.9.9.9) 4096(4124) bytes of data: 4104 bytes from 9.9.9.9: icmp_seq=1 ttl=60 time=5212 ms 4104 bytes from 9.9.9.9: icmp_seq=2 ttl=60 time=5877 ms 4104 bytes from 9.9.9.9: icmp_seq=3 ttl=60 time=6042 ms 4104 bytes from 9.9.9.9: icmp_seq=4 ttl=60 time=7210 ms 4104 bytes from 9.9.9.9: icmp_seq=5 ttl=60 time=7884 ms 4104 bytes from 9.9.9.9: icmp_seq=7 ttl=60 time=8590 ms --- 9.9.9.9 ping statistics --- 10 packets transmitted, 6 received, 40% packet loss, time 8999ms rtt min/avg/max/mdev = 5212.779/6886.306/8590.596/1150.605 ms, pipe 7</pre>	<p>Pukul 22:35</p> <p>Tim support melakukan verifikasi dengan cara melakukan ping ke arah internet ip 9.9.9.9 dengan hasil terlihat packet loss 40% dan latency 4 digit.</p>
	28 Mei 2023 pukul 22:35		<p>Pukul 22:35</p> <p>Tim support mendapatkan info pada log perangkat, yang menjelaskan telah terjadi masalah pada jalur internet via ISP INET</p>
	28 Mei 2023 pukul 22:35		<p>Pukul 22:35</p> <p>Network Administrator yang sedang berada diluar kampus mendapatkan notifikasi di telegram yang berisi tentang informasi masalah jalur internet via ISP INET</p>

Tabel 4.8 Tabel Proses Fungsi NMS – Info Penyimpanan Web Server

No	Perangkat/ Host	Waktu Masalah Perangkat / Layanan	Aksi Notifikasi	Keterangan
1	Web Server STMIK Indonesia Mandiri	28 Mei 2023 pukul 23:03	Tim developer aplikasi mendapatkan info dari user pengguna aplikasi bahwa tidak bisa melakukan update atau penambahan data ke aplikasi mahasiswa.	Pukul 23:03 User pengguna aplikasi merasakan adanya permasalahan akses ke aplikasi mahasiswa.
	Web Server STMIK Indonesia Mandiri	28 Mei 2023 pukul 23:03		Pukul 23:05 Termonitor pada peta mapping jaringan terdapat alarm dengan tanda icon merah dan menjelaskan status Web Server

		28 Mei 2023 pukul 23:05		<p>Pukul 23:05</p> <p>Tim support mendapatkan info pada log perangkat, yang menjelaskan telah terjadi masalah pada penyimpanan web server, penyimpanan server sudah mencapai > 90%</p>
		28 Mei 2023 pukul 23:05		<p>Pukul 23:05</p> <p>Network Administrator yang sedang berada diluar kampus mendapatkan notifikasi di telegram yang berisi tentang informasi masalah pada penyimpanan server yang sudah mencapai > 90%</p>

4.4.1 Hasil Pengujian

Berikut dibawah ini adalah ringkasan dari hasil pengujian NMS diatas :

Tabel 4.9 Tabel Hasil Pengujian

No	Jenis Masalah	Waktu Masalah	Notifikasi Masalah	Durasi Identifikasi
1	Masalah Access Point.	28 Mei 2023 pukul 19:03	28 Mei 2023 pukul 19:05	2 Menit
2	Masalah Koneksi Internet	28 Mei 2023 pukul 22:34	28 Mei 2023 pukul 22:35	1 Menit
3	Masalah Web Server STMIK Indonesia Mandiri	28 Mei 2023 pukul 23:03	28 Mei 2023 pukul 23:05	2 Menit

Dari hasil pengujian diatas dapat dipastikan setiap permasalahan pada perangkat atau layanan yang ada di jaringan untuk proses identifikasi masalah kurang dari 5 menit, sangat membantu untuk mempercepat proses perbaikan permasalahan.

BAB V

PENUTUP

5.1 Kesimpulan

Ada beberapa kesimpulan yang dapat dibuat berdasarkan diskusi dan penelitian yang telah dilakukan, termasuk :

1. Pembuatan NMS berbasis web ini sangat membantu tim support internal kampus untuk melakukan identifikasi jika terjadi permasalahan pada jaringan yang ada di area kampus.
2. Dengan adanya NMS ini, bisa dijadikan dasar oleh PUSKOM atau tim IT di kampus sebagai untuk peningkatan layanan jaringan yang ada dikampus. Karena NMS ini bisa mengumpulkan laporan bulanan terkait dengan kondisi perangkat *switch*, *router* dan *server*. Selain dari perangkat NMS juga bisa membuat laporan penggunaan *bandwidth internet*.
3. Dengan adanya NMS ini PUSKOM atau tim IT dikampus dapat melakukan tindakan preventif sebelum terjadi gangguan.

5.2 Saran

Sistem monitor jaringan ini masih sangat perlu dikembangkan untuk tahap penelitian selanjutnya, selain dari sistem monitoring jaringan perlu dilakukan perbaikan dan optimalisasi dari topologi jaringan yang ada dilingkungan kampus, berikut saran dari penulis antara lain sebagai berikut :

1. Sistem monitor jaringan ini masih belum tersedia dalam versi mobilnya, untuk pengembangannya akan dibuat dalam versi mobilnya. Kondisi saat ini notifikasi yang masuk ke mobile melalui media chat telegram.
2. Jaringan dikampus sebaiknya harus dilakukan beberapa pengembangan, kondisi saat ini kampus sudah mempunyai dua *upstream internet* akan tetapi belum ada penerapan *loadbalancing failover* dimana keduanya belum bisa saling membackup jika ada salah satu upstream yang bermasalah.
3. Sudah disarankan untuk melakukan pengembangan perangkat *access point* dari access point saat ini yang masih berbasis *standalone* ke *controller base*.

DAFTAR PUSTAKA

- Admin. 2021. "Jenis-Jenis Website Dan Penjelasan Lengkapnya." *MySCH*. Retrieved February 19, 2023 (<https://www.mysch.id/blog/detail/61/jenis-jenis-website-dan-penjelasan-lengkapnya>).
- Afiifah, Khoulah, Zaimah Fira Azzahra, and Azaroby Dwi Anggoro. 2022. "Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database Sebuah Literature Review." *Intech* 3(2):18–22. doi: 10.54895/intech.v3i2.1682.
- Agusvianto, Hendra. 2017. "Sistem Informasi Inventori Gudang Untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus : PT . Alaisys Sidoarjo." 01:40–46.
- Aprianti, Winda, and Umi Maliha. 2016. "Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati Kabupaten Tanah Laut." 2(2013):21–28.
- Christy, Ariata. 2023. "Apa Itu MySQL? Pengertian MySQL, Cara Kerja, Dan Kelebihannya." Retrieved (<https://www.hostinger.co.id/tutorial/apa-itu-mysql>).
- Dharmawan, Weiskhy Steven, Deasy Purwaningtias, and Deni Risdiansyah. 2018. "Penerapan Metode SDLC Waterfall Dalam Perancangan Sistem Informasi Administrasi Keuangan Berbasis Desktop." VI(2).
- Erawati, Wati. 2019. "Perancangan Sistem Informasi Penjualan Dengan Pendekatan Metode Waterfall." *Jurnal Media Informatika Budidarma* 3(1):1. doi: 10.30865/mib.v3i1.987.
- Fauzi, Rizky, and Desmulyati. 2020. "Implementasi Network Monitoring System Menggunakan Nagios Dan Nagvis Pada Pt . Pelni (Persero)." *Journal of Information System , Informatics and Computing* 4(1):92–98.
- Intern, Dicoding. 2023. "No Title." Retrieved (<https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>).
- Jayanti, Ni Ketut Dewi Ari, and Ni Kadek Sumiari. 2018. *Teori Basis Data*. Yogyakarta: Penerbit Andi.
- Klusaite, Laura. 2023. "Apakah Telegram Aman?" Retrieved (<https://nordvpn.com/id/blog/telegram-adalah/>).

- Kurniawati, Peni. 2018. "Penguujian Sistem." Retrieved (<https://medium.com/skyshidigital/penguujian-sistem-52940ee98c77>).
- Kuswanto, Herman. 2018. "Sistem Monitoring Perangkat Jaringan Menggunakan Protokol SNMP Dengan Notifikasi Email." *Jurnal Teknik Komputer* 4(2):99–104. doi: 10.31294/jtk.v4i2.3447.
- Mananggell, Arther Valentino, Alfrina Mewengkang, and Arje Cerullo Djamen. 2021. "Perancangan Jaringan Komputer Di SMK Menggunakan Cisco Packet Tracer." *EduTIK: Jurnal Pendidikan Teknologi Informasidan Komunikasi* 1(April):119–31. doi: <https://doi.org/10.53682/edutik.v1i2.1124>.
- Mediyanto, Beni, and Irfan Mahendra. 2017. "Penerapan Metode Utaut Untuk Memprediksi Behavioral Intentions User Dalam Menggunakan Aplikasi Zabbix." *Jurnal Ilmu Pengetahuan Dan Teknologi Komputer* 3(Mediyanto):10.
- Prasetyo, Bambang, Edy Budiman, and Gubtha Mahendra Putra. 2019. "Implementasi Network Monitoring System (NMS) Sebagai Sistem Peringatan Dini Pada Router Mikrotik Dengan Layanan SMS Gateway (Studi Kasus: Universitas Mulawarman)." *Prosiding Seminar Nasional Ilmu Komputer Dan Teknologi Informasi* 4(1):6–10.
- Pricillia, Titania, and Zulfachmi. 2021. "Perbandingan Metode Pengembangan Perangkat Lunak." X(01):6–12. doi: <https://doi.org/10.52771/bangkitindonesia.v10i1.153>.
- Primarta, Rifkie. 2020. "Modul Mata Kuliah Jaringan Komputer." (1):11.
- Putra, Dede Wira Trise, and Rahmi Andriani. 2019. "Unified Modelling Language (UML) Dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD." *Jurnal Teknof* 7(1):32. doi: 10.21063/jtif.2019.v7.1.32-39.
- Ramayasa, I. Putu, and Ida Bagus Ketut Surya Arnawa. 2015. "Perancangan Sistem Monitoring Pengerjaan Skripsi Pada Stmik Stikom Bali Berbasis Web." *Konferensi Nasional Sistem Dan Informatika* 760–65.
- Risald, Risald. 2021. "Implementasi Sistem Penjualan Online Berbasis E-Commerce Pada Usaha UKM IKE SUTI Menggunakan Metode Waterfall." *Journal of Information and Technology* 1(1):37–42. doi: 10.32938/jitu.v1i1.1393.
- Sibero, Alexander F. .. 2013. *Web Programming Power Pack*. Yogyakarta: Mediakom.

- Sukaridhoto, Sritrusta. 2014. "Buku Jaringan Komputer I."
- Supandi, Fandli, Wahit Desta P, Yuni Ambar, and S. Mat. 2018. "ANALISIS RESIKO PADA PENGEMBANGAN PERANGKAT LUNAK YANG MENGGUNAKAN METODE WATERFALL DAN PROTOTYPING ." 2018(Senadi):83–86.
- Supono, and Virdiandry Putratama. 2018. *Pemrograman Web Dengan Menggunakan PHP Dan Framework Codeigniter*. Yogyakarta: Deepublish.
- Suryana, Tata, and Koesheryatin. 2014. *Aplikasi Internet Menggunakan HTML, CSS & Javascript*. Jakarta: PT Elex Media Komputindo.
- Tegar, Alfian, and Putra Afandi. 2019. "Implementasi Network Monitoring System Menggunakan Librenms Berbasis Docker Container." *Jurnal UNESA* 13(Vol 13 No No 01 (2021): Vol 13 No 01 2021):1–13.
- Wibowo, Agus. 2022. "Jaringan Sistem Komputer Jilid1." *Yayasan Prima Agus Teknik Bekerja Sama Dengan Universitas Sains & Teknologi Komputer (Universitas STEKOM)*.
- Yustika, Kamaldila Puji. 2011. "Model Referensi OSI." 1–10.

LAMPIRAN

1. Lampiran Kode

1.A General.login.php

```
<?php
define('ZBX_PAGE_NO_HEADER', 1);
define('ZBX_PAGE_NO_FOOTER', 1);
$message = CHtml::encode(getRequest('message', '')) ;
// remove debug code for login form message, trimming not in regex to
relay only on [ ] in debug
message.
$message = trim(preg_replace('/\s+/', '', $message));
require_once dirname(__FILE__).'/../page_header.php';
$error = ($message !== '') ? (new CDiv($message))-
>addClass(ZBX_STYLE_RED) : null;
$guest = (CWebUser::$data['userid'] > 0)
? (new CListItem(['or ', new CLink('sign in as guest',
ZBX_DEFAULT_URL)]))
->addClass(ZBX_STYLE_SIGN_IN_TXT)
: null;
global $ZBX_SERVER_NAME;
(new CDiv([
(isset($ZBX_SERVER_NAME) && $ZBX_SERVER_NAME !== '')
? (new CDiv($ZBX_SERVER_NAME))->addClass(ZBX_STYLE_SERVER_NAME)
: null,
(new CDiv([
(new CDiv()->addClass(ZBX_STYLE_SIGNIN_LOGO),
(new CForm())
->cleanItems()
->addItem(hasRequest('request') ? new CVar('request',
getRequest('request')) :
null)
->addItem(
(new CList())
->addItem([
new CLabel(_('Username'), 'name'),
(new CTextBox('name'))->setAttribute('autofocus',
'autofocus'),
$error
])
->addItem([new CLabel(_('Password'), 'password'), (new
CTextBox('password'))->setType('password')])
->addItem(
new CLabel([
(new CCheckBox('autologin'))-
>setChecked(getRequest('autologin', 1) == 1),
_('Remember me for 30 days')
], 'autologin')
)
->addItem(new CSubmit('enter', _('Sign in')))
->addItem($guest)
)
]))->addClass(ZBX_STYLE_SIGNIN_CONTAINER),
```



```

$_REQUEST['name'] = $_SERVER['PHP_AUTH_USER'];
}
else {
access_deny(ACCESS_DENY_PAGE);
}
}
// login via form
if (isset($_REQUEST['enter']) && $_REQUEST['enter'] == _('Sign in'))
{
// try to login
$autoLogin = getRequest('autologin', 0);
DBstart();
$loginSuccess = CWebUser::login(getRequest('name', ''),
getRequest('password', ''));
DBend(true);
if ($loginSuccess) {
// save remember login preference
$user = ['autologin' => $autoLogin];
if (CWebUser::$data['autologin'] != $autoLogin) {
API::User()->updateProfile($user);
}
$request = getRequest('request');
if (!zbx_empty($request)) {
$url = $request;
}
elseif (!zbx_empty(CWebUser::$data['url'])) {
$url = CWebUser::$data['url'];
}
else {
$url = ZBX_DEFAULT_URL;
}
redirect($url);
exit;
}
// login failed, fall back to a guest account
else {
CWebUser::checkAuthentication(null);
}
else {
// login the user from the session, if the session id is empty -
login as a guest
CWebUser::checkAuthentication(CWebUser::getSessionCookie());
}
// the user is not logged in, display the login form
if (!CWebUser::$data['alias'] || CWebUser::$data['alias'] ==
ZBX_GUEST_USER) {
switch ($config['authentication_type']) {
case ZBX_AUTH_HTTP:
echo _('User name does not match with DB');
break;
case ZBX_AUTH_LDAP:
case ZBX_AUTH_INTERNAL:
if (isset($_REQUEST['enter'])) {

```

```

$_REQUEST['autologin'] = getRequest('autologin', 0);
}
if ($messages = clear_messages()) {
$messages = array_pop($messages);
$_REQUEST['message'] = $messages['message'];
}
$loginForm = new CView('general.login');
$loginForm->render();
}
}
else {
redirect(zbx_empty(CWebUser::$data['url']) ? ZBX_DEFAULT_URL :
CWebUser::$data['url']);
}
}

```

1.C Dashconf.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/hosts.inc.php';
require_once dirname(__FILE__).'/include/triggers.inc.php';
require_once dirname(__FILE__).'/include/html.inc.php';
$page['title'] = _('Dashboard configuration');
$page['file'] = 'dashconf.php';
$page['scripts'] = ['multiselect.js'];
$page['type'] = detect_page_type(PAGE_TYPE_HTML);
ob_start();
require_once dirname(__FILE__).'/include/page_header.php';
// VAR TYPE OPTIONAL FLAGS
VALIDATION EXCEPTION
$fields = [
'filterEnable' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'grpswitch' => [T_ZBX_INT, O_OPT, P_SYS, BETWEEN(0, 1),
null],
'groupids' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'hidegroupids' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'trgSeverity' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'trigger_name' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'maintenance' => [T_ZBX_INT, O_OPT, P_SYS, BETWEEN(0, 1), null],
'extAck' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'form_refresh' => [T_ZBX_INT, O_OPT, P_SYS, null,
null],
'update' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'cancel' => [T_ZBX_STR, O_OPT, P_SYS, null,
null]
];
check_fields($fields);

```

```

/*
* Actions
*/
if (hasRequest('update')) {
// filter
$filterEnable = getRequest('filterEnable', 0);
CProfile::update('web.dashconf.filter.enable', $filterEnable,
PROFILE_TYPE_INT);
if ($filterEnable == 1) {
// groups
CProfile::update('web.dashconf.groups.grpswitch',
$_REQUEST['grpswitch'],
PROFILE_TYPE_INT);
if ($_REQUEST['grpswitch'] == 1) {
// show groups
$groupIds = getRequest('groupids', []);
$result = true;
DBstart();
$result &= CFavorite::remove('web.dashconf.groups.groupids');
foreach ($groupIds as $groupId) {
$result &= CFavorite::add('web.dashconf.groups.groupids', $groupId);
}
// hide groups
$hideGroupIds = getRequest('hidegroupids', []);
$result &= CFavorite::remove('web.dashconf.groups.hide.groupids');
foreach ($hideGroupIds as $hideGroupId) {
$result &= CFavorite::add('web.dashconf.groups.hide.groupids',
$hideGroupId);
}
DBend($result);
}
// hosts
CProfile::update('web.dashconf.hosts.maintenance',
getRequest('maintenance', 0),
PROFILE_TYPE_INT);
// triggers
CProfile::update('web.dashconf.triggers.severity',
implode(';', array_keys(getRequest('trgSeverity', []))),
PROFILE_TYPE_STR
);
CProfile::update('web.dashconf.triggers.name',
getRequest('trigger_name', ''),
PROFILE_TYPE_STR);
// events
$config = select_config();
if ($config['event_ack_enable']) {
CProfile::update('web.dashconf.events.extAck', getRequest('extAck',
0),
PROFILE_TYPE_INT);
}
}
jsredirect(ZBX_DEFAULT_URL);
}
elseif (hasRequest('cancel')) {

```

```

ob_end_clean();
redirect(ZBX_DEFAULT_URL);
}
ob_end_flush();
$data = [
'config' => select_config()
];
if (hasRequest('form_refresh')) {
$data['isFilterEnable'] = getRequest('filterEnable', 0);
$data['maintenance'] = getRequest('maintenance', 0);
$data['extAck'] = getRequest('extAck', 0);
$data['severity'] = getRequest('trgSeverity', []);
$data['severity'] = array_keys($data['severity']);
$data['trigger_name'] = getRequest('trigger_name', '');
// groups
$data['grpswitch'] = getRequest('grpswitch', 0);
$data['groupIds'] = getRequest('groupids', []);
$data['groupIds'] = zbx_toHash($data['groupIds']);
$data['hideGroupIds'] = getRequest('hidegroupids', []);
$data['hideGroupIds'] = zbx_toHash($data['hideGroupIds']);
}
else {
$data['isFilterEnable'] = CProfile::get('web.dashconf.filter.enable',
0);
$data['maintenance'] =
CProfile::get('web.dashconf.hosts.maintenance', 1);
$data['extAck'] = CProfile::get('web.dashconf.events.extAck', 0);
$data['severity'] = CProfile::get('web.dashconf.triggers.severity',
'0;1;2;3;4;5');
$data['severity'] = zbx_empty($data['severity']) ? [] : explode(';',
$data['severity']);
$data['trigger_name'] = CProfile::get('web.dashconf.triggers.name',
'');
// groups
$data['grpswitch'] = CProfile::get('web.dashconf.groups.grpswitch',
0);
$data['groupIds'] = CFavorite::get('web.dashconf.groups.groupids');
$data['groupIds'] = zbx_objectValues($data['groupIds'], 'value');
$data['groupIds'] = zbx_toHash($data['groupIds']);
$data['hideGroupIds'] =
CFavorite::get('web.dashconf.groups.hide.groupids');
$data['hideGroupIds'] = zbx_objectValues($data['hideGroupIds'],
'value');
$data['hideGroupIds'] = zbx_toHash($data['hideGroupIds']);
TRIGGER_SEVERITY_NOT_CLASSIFIED,
TRIGGER_SEVERITY_INFORMATION,
TRIGGER_SEVERITY_WARNING,
TRIGGER_SEVERITY_AVERAGE,
TRIGGER_SEVERITY_HIGH,
TRIGGER_SEVERITY_DISASTER
];
if ($data['grpswitch']) {
// show groups
$data['groups'] = API::HostGroup()->get([

```

```

'groupids' => $data['groupIds'],
'output' => ['groupid', 'name']
]);
CArrayHelper::sort($data['groups'], [
['field' => 'name', 'order' => ZBX_SORT_UP]
]);
foreach ($data['groups'] as &$group) {
$group['id'] = $group['groupid'];
unset($group['groupid']);
}
unset($group);
// hide groups
$data['hideGroups'] = API::HostGroup()->get([
'groupids' => $data['hideGroupIds'],
'output' => ['groupid', 'name']
]);
CArrayHelper::sort($data['hideGroups'], [
['field' => 'name', 'order' => ZBX_SORT_UP]
]);
foreach ($data['hideGroups'] as &$group) {
$group['id'] = $group['groupid'];
unset($group['groupid']);
}
unset($group);
}
// render view
$dashconfView = new CView('monitoring.dashconf', $data);
$dashconfView->render();
$dashconfView->show();
require_once dirname(__FILE__).'/include/page_footer.php';
$data['severity'] = zbx_toHash($data['severity']);
$data['severities'] = [
TRIGGER_SEVERITY_NOT_CLASSIFIED,
TRIGGER_SEVERITY_INFORMATION,
TRIGGER_SEVERITY_WARNING,
TRIGGER_SEVERITY_AVERAGE,
TRIGGER_SEVERITY_HIGH,
TRIGGER_SEVERITY_DISASTER
];
if ($data['grpswitch']) {
// show groups
$data['groups'] = API::HostGroup()->get([
'groupids' => $data['groupIds'],
'output' => ['groupid', 'name']
]);
CArrayHelper::sort($data['groups'], [
['field' => 'name', 'order' => ZBX_SORT_UP]
]);
foreach ($data['groups'] as &$group) {
$group['id'] = $group['groupid'];
unset($group['groupid']);
}
unset($group);
// hide groups

```

```

$data['hideGroups'] = API::HostGroup()->get([
'groupids' => $data['hideGroupIds'],
'output' => ['groupid', 'name']
]);
CArrayHelper::sort($data['hideGroups'], [
['field' => 'name', 'order' => ZBX_SORT_UP]
]);
foreach ($data['hideGroups'] as &$group) {
$group['id'] = $group['groupid'];
5a
unset($group['groupid']);
}
unset($group);
}
// render view
$dashconfView = new CView('monitoring.dashconf', $data);
$dashconfView->render();
$dashconfView->show();
require_once dirname(__FILE__).'/include/page_footer.php';

```

1.D zabbix.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';

```

1.E charts.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/hosts.inc.php';
require_once dirname(__FILE__).'/include/graphs.inc.php';
$page['title'] = _('Custom graphs');
$page['file'] = 'charts.php';
$page['scripts'] = ['class.calendar.js', 'gtlc.js',
'flickerfreescreen.js'];
$page['type'] = detect_page_type(PAGE_TYPE_HTML);
define('ZBX_PAGE_DO_JS_REFRESH', 1);
ob_start();
require_once dirname(__FILE__).'/include/page_header.php';
// VAR TYPE OPTIONAL FLAGS VALIDATION EXCEPTION
$fields = [
'groupid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'graphid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'period' => [T_ZBX_INT, O_OPT, P_SYS, null, null],
'stime' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'fullscreen' => [T_ZBX_INT, O_OPT, P_SYS, IN('0,1'), null],
// ajax
'favobj' => [T_ZBX_STR, O_OPT, P_ACT, null, null],
'favid' => [T_ZBX_INT, O_OPT, P_ACT, null, null]
];
check_fields($fields);
/*

```

```

* Permissions
*/
if (getRequest('groupid') && !API::HostGroup()-
>isReadable([$REQUEST['groupid']])) {
access_deny();
}
if (getRequest('hostid') && !API::Host()-
>isReadable([$REQUEST['hostid']])) {
access_deny();
}
if (getRequest('graphid')) {
$graphs = API::Graph()->get([
'graphids' => [$REQUEST['graphid']],
'output' => ['graphid']
]);
if (!$graphs) {
access_deny();
}
}
$pageFilter = new CPageFilter([
'groups' => ['real_hosts' => true, 'with_graphs' => true],
'hosts' => ['with_graphs' => true],
'groupid' => getRequest('groupid'),
'hostid' => getRequest('hostid'),
'graphs' => ['templated' => 0],
'graphid' => getRequest('graphid')
]);
if (isset($REQUEST['favobj'])) {
if (getRequest('favobj') === 'timelinefixedperiod' &&
hasRequest('favid')) {
CProfile::update('web.screens.timelinefixed', getRequest('favid'),
PROFILE_TYPE_INT);
}
}
if (!empty($REQUEST['period']) || !empty($REQUEST['stime'])) {
CScreenBase::calculateTime([
'profileIdx' => 'web.screens',
'profileIdx2' => $pageFilter->graphid,
'updateProfile' => true,
'period' => getRequest('period'),
'stime' => getRequest('stime')
]);
}
$curl = (new CUrl())
->removeArgument('period')
->removeArgument('stime');
ob_end_clean();
DBstart();
CProfile::flush();
DBend();
redirect($curl->getUrl());
}
ob_end_flush();
if ($page['type'] == PAGE_TYPE_JS || $page['type'] ==
PAGE_TYPE_HTML_BLOCK) {

```

```

require_once dirname(__FILE__).'/include/page_footer.php';
exit;
}
$data = [
'pageFilter' => $pageFilter,
'graphid' => $pageFilter->graphid,
'fullscreen' => $_REQUEST['fullscreen']
];
// render view
$chartsView = new CView('monitoring.charts', $data);
$chartsView->render();
$chartsView->show();
require_once dirname(__FILE__).'/include/page_footer.php';

```

1.F trstatus.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
$page['file'] = 'tr_status.php';
$page['title'] = _('Status of triggers');
$page['scripts'] = ['class.cswitcher.js'];
$page['type'] = detect_page_type(PAGE_TYPE_HTML);
if ($page['type'] == PAGE_TYPE_HTML) {
define('ZBX_PAGE_DO_REFRESH', 1);
}
require_once dirname(__FILE__).'/include/page_header.php';
// VAR TYPE OPTIONAL FLAGS VALIDATION EXCEPTION
$fields = [
'groupid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'fullscreen' => [T_ZBX_INT, O_OPT, P_SYS, IN('0,1'), null],
'btnSelect' => [T_ZBX_STR, O_OPT, null, null, null],
// filter
'filter_rst' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'filter_set' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'show_triggers' => [T_ZBX_INT, O_OPT, null, null, null],
'show_events' => [T_ZBX_INT, O_OPT, P_SYS, null, null],
'ack_status' => [T_ZBX_INT, O_OPT, P_SYS, null, null],
'show_severity' => [T_ZBX_INT, O_OPT, P_SYS, null, null],
'show_details' => [T_ZBX_INT, O_OPT, null, null, null],
'show_maintenance' => [T_ZBX_INT, O_OPT, null, null, null],
'status_change_days' => [T_ZBX_INT, O_OPT, null, BETWEEN(1,
DAY_IN_YEAR * 2), null],
'status_change' => [T_ZBX_INT, O_OPT, null, null, null],
'txt_select' => [T_ZBX_STR, O_OPT, null, null, null],
'application' => [T_ZBX_STR, O_OPT, null, null, null],
'inventory' => [T_ZBX_STR, O_OPT, null, null, null],
// sort and sortorder
'sort' => [T_ZBX_STR, O_OPT, P_SYS,
IN('"description","lastchange","priority"),
null],
'sortorder' => [T_ZBX_STR, O_OPT, P_SYS,
IN(''.ZBX_SORT_DOWN.'","'.ZBX_SORT_UP.''),
null]

```



```

];
check_fields($fields);
if (getRequest('groupid') && !API::HostGroup()-
>isReadable([getRequest('groupid']])) {
access_deny();
}
if (getRequest('hostid') && !API::Host()-
>isReadable([getRequest('hostid']])) {
access_deny();
}
$config = select_config();
$pageFilter = new CPageFilter([
'groups' => [
'monitored_hosts' => true,
'with_monitored_triggers' => true
],
'hosts' => [
'monitored_hosts' => true,
'with_monitored_triggers' => true
],
'hostid' => getRequest('hostid'),
'groupid' => getRequest('groupid')
]);
$_REQUEST['groupid'] = $pageFilter->groupid;
$_REQUEST['hostid'] = $pageFilter->hostid;
// filter set
if (hasRequest('filter_set')) {
CProfile::update('web.tr_status.filter.show_triggers',
getRequest('show_triggers',
TRIGGERS_OPTION_RECENT_PROBLEM),
PROFILE_TYPE_INT
);
CProfile::update('web.tr_status.filter.show_details',
getRequest('show_details', 0), PROFILE_TYPE_INT);
CProfile::update('web.tr_status.filter.show_maintenance',
getRequest('show_maintenance', 0),
PROFILE_TYPE_INT);
CProfile::update('web.tr_status.filter.show_severity',
getRequest('show_severity',
TRIGGER_SEVERITY_NOT_CLASSIFIED),
PROFILE_TYPE_INT
);
CProfile::update('web.tr_status.filter.txt_select',
getRequest('txt_select', ''), PROFILE_TYPE_STR);
CProfile::update('web.tr_status.filter.status_change',
getRequest('status_change', 0), PROFILE_TYPE_INT);
CProfile::update('web.tr_status.filter.status_change_days',
getRequest('status_change_days', 14),
PROFILE_TYPE_INT
);
CProfile::update('web.tr_status.filter.application',
getRequest('application'), PROFILE_TYPE_STR);
// show events
$showEvents = getRequest('show_events', EVENTS_OPTION_NOEVENT);

```

```

if ($config['event_ack_enable'] == EVENT_ACK_ENABLED || $showEvents !
= EVENTS_OPTION_NOT_ACK) {
CProfile::update('web.tr_status.filter.show_events', $showEvents,
PROFILE_TYPE_INT);
}
// ack status
if ($config['event_ack_enable'] == EVENT_ACK_ENABLED) {
CProfile::update('web.tr_status.filter.ack_status',
getRequest('ack_status', ZBX_ACK_STS_ANY),
PROFILE_TYPE_INT);
}
// update host inventory filter
$inventoryFields = [];
$inventoryValues = [];
foreach (getRequest('inventory', []) as $field) {
if ($field['value'] === '') {
continue;
}
$inventoryFields[] = $field['field'];
$inventoryValues[] = $field['value'];
}
CProfile::updateArray('web.tr_status.filter.inventory.field',
$inventoryFields, PROFILE_TYPE_STR);
CProfile::updateArray('web.tr_status.filter.inventory.value',
$inventoryValues, PROFILE_TYPE_STR);
}
elseif (hasRequest('filter_rst')) {
DBStart();
CProfile::delete('web.tr_status.filter.show_triggers');
CProfile::delete('web.tr_status.filter.show_details');
CProfile::delete('web.tr_status.filter.show_maintenance');
CProfile::delete('web.tr_status.filter.show_events');
CProfile::delete('web.tr_status.filter.ack_status');
CProfile::delete('web.tr_status.filter.show_severity');
CProfile::delete('web.tr_status.filter.txt_select');
CProfile::delete('web.tr_status.filter.status_change');
CProfile::delete('web.tr_status.filter.status_change_days');
CProfile::delete('web.tr_status.filter.application');
CProfile::deleteIdx('web.tr_status.filter.inventory.field');
CProfile::deleteIdx('web.tr_status.filter.inventory.value');
DBend();
}
$showTriggers = CProfile::get('web.tr_status.filter.show_triggers',
TRIGGERS_OPTION_RECENT_PROBLEM);
$showDetails = CProfile::get('web.tr_status.filter.show_details', 0);
$showMaintenance =
CProfile::get('web.tr_status.filter.show_maintenance', 1);
$showSeverity = CProfile::get('web.tr_status.filter.show_severity',
TRIGGER_SEVERITY_NOT_CLASSIFIED);
$txtSelect = CProfile::get('web.tr_status.filter.txt_select', '');
$showChange = CProfile::get('web.tr_status.filter.status_change', 0);
$statusChangeDays =
CProfile::get('web.tr_status.filter.status_change_days', 14);
$ackStatus = ($config['event_ack_enable'] == EVENT_ACK_DISABLED)

```

```

? ZBX_ACK_STS_ANY : CProfile::get('web.tr_status.filter.ack_status',
ZBX_ACK_STS_ANY);
$showEvents = CProfile::get('web.tr_status.filter.show_events',
EVENTS_OPTION_NOEVENT);
// check event acknowledges
if ($config['event_ack_enable'] == EVENT_ACK_DISABLED && $showEvents
== EVENTS_OPTION_NOT_ACK) {
$showEvents = EVENTS_OPTION_NOEVENT;
}
// fetch filter from profiles
$filter = [
'application' => CProfile::get('web.tr_status.filter.application',
''),
'inventory' => []
];
foreach (CProfile::getArray('web.tr_status.filter.inventory.field',
[]) as $i => $field) {
$filter['inventory'][] = [
'field' => $field,
'value' => CProfile::get('web.tr_status.filter.inventory.value',
null, $i)
];
}
$sortField = getRequest('sort', CProfile::get('web.'.
$page['file'].'.sort', 'lastchange'));
$sortOrder = getRequest('sortorder', CProfile::get('web.'.
$page['file'].'.sortorder', ZBX_SORT_DOWN));
CProfile::update('web.'. $page['file'].'.sort', $sortField,
PROFILE_TYPE_STR);
CProfile::update('web.'. $page['file'].'.sortorder', $sortOrder,
PROFILE_TYPE_STR);
$triggerWidget = (new CWidget()->setTitle(_('Status of triggers'));
$triggerWidget->setControls((new CForm('get'))
->addVar('fullscreen', $_REQUEST['fullscreen'])
->addItem((new CList())
->addItem([
new CLabel(_('Group'), 'groupid'),
(new CDiv()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
$pageFilter->getGroupsCB()
])
->addItem([
new CLabel(_('Host'), 'hostid'),
(new CDiv()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
$pageFilter->getHostsCB()
])
->addItem(get_icon('fullscreen', ['fullscreen' =>
$_REQUEST['fullscreen']]))
)
);
// filter
$filterFormView = new CView('common.filter.trigger', [
'overview' => false,
'filter' => [
'filterid' => 'web.tr_status.filter.state',

```

```

'showTriggers' => $showTriggers,
'ackStatus' => $ackStatus,
'showEvents' => $showEvents,
'showSeverity' => $showSeverity,
'statusChange' => $showChange,
'statusChangeDays' => $statusChangeDays,
'showDetails' => $showDetails,
'txtSelect' => $txtSelect,
'application' => $filter['application'],
'inventory' => $filter['inventory'],
'showMaintenance' => $showMaintenance,
'hostId' => getRequest('hostid'),
'groupId' => getRequest('groupid'),
'fullScreen' => getRequest('fullscreen')
],
'config' => $config
]);
$filterForm = $filterFormView->render();
$triggerWidget->addItem($filterForm);
/*
 * Form
 */
$triggerForm = (new CForm('get', 'zabbix.php'))
->setName('tr_status')
->addVar('backurl', $page['file'])
->addVar('acknowledge_type', ZBX_ACKNOWLEDGE_PROBLEM);
$switcherName = 'trigger_switchers';
if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
$showHideAllButton = (new CColHeader(
(new CSimpleButton())
->addClass(ZBX_STYLE_TREEVIEW)
->setId($switcherName)
->addItem((new CSpan())->addClass(ZBX_STYLE_ARROW_RIGHT))
))->addClass(ZBX_STYLE_CELL_WIDTH);
}
else { $showHideAllButton = null;}
if ($config['event_ack_enable']) {
$headerCheckBox = (new CColHeader(
(new CCheckBox('all_eventids'))
->onClick("checkAll('".$triggerForm->GetName()."', 'all_eventids',
'eventids');")
))->addClass(ZBX_STYLE_CELL_WIDTH);
}
else {
$headerCheckBox = null;
}
$triggerTable = (new CTableInfo())
->setHeader([
$showHideAllButton,
$headerCheckBox,
make_sorting_header(_('Severity'), 'priority', $sortField,
$sortOrder),
_('Status'),

```

```

_('Info'),
make_sorting_header(_('Last change'), 'lastchange', $sortField,
$sortOrder),
_('Age'),
($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) ? _('Duration') : null,
$config['event_ack_enable'] ? _('Ack') : null,
_('Host'),
make_sorting_header(_('Name'), 'description', $sortField,
$sortOrder),
_('Description')
]);
// get triggers
$options = [
'output' => ['triggerid', $sortField],
'monitored' => true,
'skipDependent' => true,
'sortfield' => $sortField,
'limit' => $config['search_limit'] + 1
];
if ($pageFilter->hostsSelected) {
if ($pageFilter->hostid > 0) {
$options['hostids'] = $pageFilter->hostid;
}
elseif ($pageFilter->groupid > 0) {
$options['groupids'] = $pageFilter->groupid;
}
}
else {
$options['hostids'] = [];
}
// inventory filter
if ($filter['inventory']) {
$inventoryFilter = [];
foreach ($filter['inventory'] as $field) {
$inventoryFilter[$field['field']][] = $field['value'];
}
$hosts = API::Host()->get([
'output' => ['hostid'],
'hostids' => isset($options['hostids']) ? $options['hostids'] : null,
'searchInventory' => $inventoryFilter
]);
$options['hostids'] = zbx_objectValues($hosts, 'hostid');
}
// application filter
if ($filter['application'] !== '') {
$applications = API::Application()->get([
'output' => ['applicationid'],
'hostids' => isset($options['hostids']) ? $options['hostids'] : null,
'search' => ['name' => $filter['application']]
]);
$options['applicationids'] = zbx_objectValues($applications,
'applicationid');
}
}

```

```

if (!zbx_empty($txtSelect)) {
$options['search'] = ['description' => $txtSelect];
}
if ($showTriggers == TRIGGERS_OPTION_RECENT_PROBLEM) {
$options['only_true'] = 1;
}
elseif ($showTriggers == TRIGGERS_OPTION_IN_PROBLEM) {
$options['filter'] = ['value' => TRIGGER_VALUE_TRUE];
}
if ($ackStatus == ZBX_ACK_STS_WITH_UNACK) {
$options['withUnacknowledgedEvents'] = 1;
}
if ($ackStatus == ZBX_ACK_STS_WITH_LAST_UNACK) {
$options['withLastEventUnacknowledged'] = 1;
}
if ($showSeverity > TRIGGER_SEVERITY_NOT_CLASSIFIED) {
$options['min_severity'] = $showSeverity;
}
if ($showChange) {
$options['lastChangeSince'] = time() - $statusChangeDays *
SEC_PER_DAY;
}
if (!$showMaintenance) {
$options['maintenance'] = false;
}
$triggers = API::Trigger()->get($options);
order_result($triggers, $sortField, $sortOrder);
$url = (new CUrl('tr_status.php'))
->setArgument('fullscreen', getRequest('fullscreen'))
->setArgument('groupid', $pageFilter->groupid)
->setArgument('hostid', $pageFilter->hostid);
$paging = getPagingLine($triggers, $sortOrder, $url);
$triggers = API::Trigger()->get([
'triggerids' => zbx_objectValues($triggers, 'triggerid'),
'output' => API_OUTPUT_EXTEND,
'selectHosts' => [
'hostid', 'name', 'description', 'status', 'maintenanceid',
'maintenance_status', 'maintenance_type'
],
'selectItems' => ['itemid', 'hostid', 'name', 'key_', 'value_type'],
'selectDependencies' => API_OUTPUT_EXTEND,
'selectLastEvent' => ['eventid', 'objectid', 'clock', 'ns'],
'preservekeys' => true
]);
10a
$triggers = CMacrosResolverHelper::resolveTriggerUrls($triggers);
if ($showDetails) {
foreach ($triggers as &$trigger) {
$trigger['expression_orig'] = $trigger['expression'];
}
unset($trigger);
$triggers =
CMacrosResolverHelper::resolveTriggerExpressions($triggers,

```

```

['html' => true, 'resolve_usermacros' => true, 'resolve_macros' =>
true]
);
foreach ($triggers as &$trigger) {
$trigger['expression_html'] = $trigger['expression'];
$trigger['expression'] = $trigger['expression_orig'];
unset($trigger['expression_orig']);
}
unset($trigger);
}
order_result($triggers, $sortField, $sortOrder);
// sort trigger hosts by name
foreach ($triggers as &$trigger) {
if (count($trigger['hosts']) > 1) {
order_result($trigger['hosts'], 'name', ZBX_SORT_UP);
}
}
unset($trigger);
$triggerIds = zbx_objectValues($triggers, 'triggerid');
// get editable triggers
$triggerEditable = API::Trigger()->get([
'triggerids' => $triggerIds,
'output' => ['triggerid'],
'filter' => ['flags' => ZBX_FLAG_DISCOVERY_NORMAL],
'editable' => true,
'preservekeys' => true
]);
// get events
if ($config['event_ack_enable']) {
// get all unacknowledged events, if trigger has unacknowledged event
=> it has events
$eventCounts = API::Event()->get([
'source' => EVENT_SOURCE_TRIGGERS,
'object' => EVENT_OBJECT_TRIGGER,
'countOutput' => true,
'groupCount' => true,
'objectids' => $triggerIds,
'filter' => [
'acknowledged' => 0,
'value' => TRIGGER_VALUE_TRUE
]
]);
foreach ($eventCounts as $eventCount) {
$triggers[$eventCount['objectid']]['hasEvents'] = true;
$triggers[$eventCount['objectid']]['event_count'] =
$eventCount['rowcount'];
}
// gather ids of triggers which don't have unack. events
$triggerIdsWithoutUnackEvents = [];
foreach ($triggers as $tnum => $trigger) {
if (!isset($trigger['hasEvents'])) {
$triggerIdsWithoutUnackEvents[] = $trigger['triggerid'];
}
}
if (!isset($trigger['event_count'])) {

```

```

$triggers[$tnum]['event_count'] = 0;
}
}
if (!empty($triggerIdsWithoutUnackEvents)) {
// for triggers without unack. events we try to select any event
$allEventCounts = API::Event()->get([
'source' => EVENT_SOURCE_TRIGGERS,
'object' => EVENT_OBJECT_TRIGGER,
'countOutput' => true,
'groupCount' => true,
'objectids' => $triggerIdsWithoutUnackEvents
]);
$allEventCounts = zbx_toHash($allEventCounts, 'objectid');
foreach ($triggers as $tnum => $trigger) {
if (!isset($trigger['hasEvents'])) {
$triggers[$tnum]['hasEvents'] =
isset($allEventCounts[$trigger['triggerid']]);
}
}
}
if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
foreach ($triggers as &$trigger) {
$trigger['display_events'] = false;
$trigger['events'] = [];
}
unset($trigger);
$options = [
'output' => ['eventid', 'objectid', 'clock', 'value'],
'source' => EVENT_SOURCE_TRIGGERS,
'object' => EVENT_OBJECT_TRIGGER,
'objectids' => zbx_objectValues($triggers, 'triggerid'),
'time_from' => time() - $config['event_expire'] * SEC_PER_DAY,
'time_till' => time(),
'sortfield' => ['clock', 'eventid'],
'sortorder' => ZBX_SORT_DOWN
];
if ($config['event_ack_enable']) {
$options['select_acknowledges'] = API_OUTPUT_COUNT;
$options['output'][] = 'acknowledged';
}
$events = API::Event()->get($options);
foreach ($events as $event) {
$triggers[$event['objectid']]['events'][] = $event;
if ($showEvents == EVENTS_OPTION_ALL) {
$triggers[$event['objectid']]['display_events'] = true;
}
elseif (!$event['acknowledged'] && $event['value'] ==
TRIGGER_VALUE_TRUE) {
$triggers[$event['objectid']]['display_events'] = true;
}
}
}
}
}

```



```

else {
foreach ($triggers as &$trigger) {
$trigger['display_events'] = false;
}
unset($trigger);
}
// get host ids
$hostIds = [];
foreach ($triggers as $tnum => $trigger) {
foreach ($trigger['hosts'] as $host) {
$hostIds[$host['hostid']] = $host['hostid'];
}
}
// get hosts
$hosts = API::Host()->get([
'output' => ['hostid', 'status'],
'hostids' => $hostIds,
'preservekeys' => true,
'selectGraphs' => API_OUTPUT_COUNT,
'selectScreens' => API_OUTPUT_COUNT
]);
// get host scripts
$scriptsByHosts = API::Script()->getScriptsByHosts($hostIds);
// get trigger dependencies
$dbTriggerDependencies = DBselect(
'SELECT triggerid_down,triggerid_up'.
' FROM trigger_depends'.
' WHERE '.dbConditionInt('triggerid_up', $triggerIds)
);
$triggerIdsDown = [];
while ($row = DBfetch($dbTriggerDependencies)) {
$triggerIdsDown[$row['triggerid_up']][] =
intval($row['triggerid_down']);
}
$maintenanceids = [];
foreach ($triggers as $trigger) {
foreach ($trigger['hosts'] as $host) {
if ($host['maintenance_status'] == HOST_MAINTENANCE_STATUS_ON) {
$maintenanceids[$host['maintenanceid']] = true;
}
}
}
if ($maintenanceids) {
$maintenances = API::Maintenance()->get([
'maintenanceids' => array_keys($maintenanceids),
'output' => ['name', 'description'],
'preservekeys' => true
]);
}
foreach ($triggers as $trigger) {
$trigger['groupid'] = $pageFilter->groupid;
$trigger['hostid'] = $pageFilter->hostid;
$description = [];
if (!empty($trigger['dependencies'])) {

```

```

$dependenciesTable = (new CTable())
->setAttribute('style', 'min-width:
'.ZBX_TEXTAREA_STANDARD_WIDTH.'px;')
12a
->addRow(_('Depends on').':');
foreach ($trigger['dependencies'] as $dependency) {
$dependenciesTable->addRow(' -
'.CMacrosResolverHelper::resolveTriggerNameById($dependency['triggerid']));
}
$description[] = (new CSpan())
->addClass(ZBX_STYLE_ICON_DEPEND_DOWN)
->addClass(ZBX_STYLE_CURSOR_POINTER)
->setHint($dependenciesTable);
}
$dependency = false;
$dependenciesTable = (new CTable())
->setAttribute('style', 'min-width:
'.ZBX_TEXTAREA_STANDARD_WIDTH.'px;')
->addRow(_('Dependent').':');
if (array_key_exists($trigger['triggerid'], $triggerIdsDown) &&
$triggerIdsDown[$trigger['triggerid']]) {
$depTriggers =
CMacrosResolverHelper::resolveTriggerNameByIds($triggerIdsDown[$trigger['triggerid']]);
foreach ($depTriggers as $depTrigger) {
$dependenciesTable->addRow(SPACE.'-'.SPACE.
$depTrigger['description']);
$dependency = true;
}
}
if ($dependency) {
$description[] = (new CSpan())
->addClass(ZBX_STYLE_ICON_DEPEND_UP)
->addClass(ZBX_STYLE_CURSOR_POINTER)
->setHint($dependenciesTable);
}
unset($img, $dependenciesTable, $dependency);
// Trigger has events.
if ($trigger['lastEvent']) {
$event = [
'clock' => $trigger['lastEvent']['clock'],
'ns' => $trigger['lastEvent']['ns']
];
}
// Trigger has no events.
else {
$event = [
'clock' => $trigger['lastchange'],
'ns' => '999999999'
];
}
}

```

```

$description[] = (new
Cspan(CMacrosResolverHelper::resolveEventDescription(zbx_array_merge(
$trigger,
$event))))
->addClass(ZBX_STYLE_LINK_ACTION)
->setMenuPopup(CMenuPopupHelper::getTrigger($trigger));
if ($showDetails) {
$description[] = BR();
$description[] = $trigger['expression_html'];
}
// host js menu
$hostList = [];
foreach ($trigger['hosts'] as $host) {
// fetch scripts for the host js menu
$scripts = [];
if (isset($scriptsByHosts[$host['hostid']])) {
foreach ($scriptsByHosts[$host['hostid']] as $script) {
$scripts[] = $script;
}
}
$host_name = (new Cspan($host['name']))
->addClass(ZBX_STYLE_LINK_ACTION)
->setMenuPopup(CMenuPopupHelper::getHost($hosts[$host['hostid']],
$scripts));
// add maintenance icon with hint if host is in maintenance
if ($host['maintenance_status'] == HOST_MAINTENANCE_STATUS_ON) {
$maintenance_icon = (new Cspan())
->addClass(ZBX_STYLE_ICON_MAINT)
->addClass(ZBX_STYLE_CURSOR_POINTER);
if (array_key_exists($host['maintenanceid'], $maintenances)) {
$maintenance = $maintenances[$host['maintenanceid']];
$hint = $maintenance['name'].' ['.($host['maintenance_type']
? _('Maintenance without data collection')
: _('Maintenance with data collection')).'];
if ($maintenance['description']) {
$hint .= "\n".$maintenance['description'];
}
$maintenance_icon->setHint($hint);
}
$host_name = (new Cspan([$host_name, $maintenance_icon]))-
>addClass(ZBX_STYLE_REL_CONTAINER);
}
$hostList[] = $host_name;
$hostList[] = ', ';
}
array_pop($hostList);
// status
$statusSpan = new Cspan(trigger_value2str($trigger['value']));
// add colors and blinking to span depending on configuration and
trigger parameters
addTriggerValueStyle(
$statusSpan,
$trigger['value'],
$trigger['lastchange'],

```

```

$config['event_ack_enable'] ? ($trigger['event_count'] == 0) : false
);
// open or close
// acknowledge
if ($config['event_ack_enable']) {
if ($trigger['hasEvents']) {
$ack_checkbox = new CCheckBox('eventids['.$trigger['lastEvent']
['eventid'].'],'
,$trigger['lastEvent']['eventid']
);
if ($trigger['event_count']) {
$ackColumn = [
(new CLink(_('No'),
'zabbix.php?action=acknowledge.edit'.
'&acknowledge_type='.ZBX_ACKNOWLEDGE_PROBLEM.
'&eventids[]=$trigger['lastEvent']['eventid'].
'&backurl='.$page['file']
))
->addClass(ZBX_STYLE_LINK_ALT)
->addClass(ZBX_STYLE_RED),
CViewHelper::showNum($trigger['event_count'])
];
}
else {
$ackColumn = (new CLink(_('Yes'),
'zabbix.php?action=acknowledge.edit'.
'&acknowledge_type='.ZBX_ACKNOWLEDGE_PROBLEM.
'&eventids[]=$trigger['lastEvent']['eventid'].
'&backurl='.$page['file']
))
->addClass(ZBX_STYLE_LINK_ALT)
->addClass(ZBX_STYLE_GREEN);
}
}
else {
$ack_checkbox = '';
$ackColumn = (new CCol(_('No events'))->addClass(ZBX_STYLE_GREY);
}
}
else {
$ack_checkbox = null;
$ackColumn = null;
}
if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
$openOrCloseButton = $trigger['display_events']
? (new CSimpleButton())
->addClass(ZBX_STYLE_TREEVIEW)
->setAttribute('data-switcherid', $trigger['triggerid'])
->addItem((new CSpan())->addClass(ZBX_STYLE_ARROW_RIGHT))
: '';
}
else {
$openOrCloseButton = null;

```

```

}
// comments
if (isset($triggerEditable[$trigger['triggerid']])) {
    $comments = new CLink(zbx_empty($trigger['comments']) ? _('Add') :
        _('Show'),
        'tr_comments.php?triggerid='.$trigger['triggerid']);
}
else {
    $comments = zbx_empty($trigger['comments'])
        ? new CSpan('')
        : new CLink(_('Show'), 'tr_comments.php?triggerid='.$
            $trigger['triggerid']);
}
$triggerTable->addRow([
    $openOrCloseButton,
    $ack_checkbox,
    getSeverityCell($trigger['priority'], $config, null, !
        $trigger['value']),
    $statusSpan,
    ($trigger['state'] == TRIGGER_STATE_UNKNOWN) ?
        makeUnknownIcon($trigger['error']) : '',
    ($trigger['lastchange'] == 0)
        ? _('Never')
        : new CLink(zbx_date2str(DATE_TIME_FORMAT_SECONDS,
            $trigger['lastchange']),
            'events.php?filter_set=1&triggerid='.$
            $trigger['triggerid'].'&source='.EVENT_SOURCE_TRIGGERS.
            '&stime='.date(TIMESTAMP_FORMAT,
            $trigger['lastchange'])."&period=".ZBX_PERIOD_DEFAULT
        ),
    ($trigger['lastchange'] == 0) ? '' :
        zbx_date2age($trigger['lastchange']),
    ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
        EVENTS_OPTION_NOT_ACK) ? '' : null,
    $ackColumn,
    $hostList,
    $description,
    $comments
]);
if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
    EVENTS_OPTION_NOT_ACK) {
    $next_event_clock = time();
    foreach (array_slice($trigger['events'], 0,
        $config['event_show_max']) as $enum => $event) {
        if ($showEvents == EVENTS_OPTION_NOT_ACK) {
            if ($event['acknowledged'] || $event['value'] != TRIGGER_VALUE_TRUE)
            {
                continue;
            }
        }
        $eventStatusSpan = new CSpan(trigger_value2str($event['value']));
        // add colors and blinking to span depending on configuration and
        trigger parameters
    }
}

```

```

addTriggerValueStyle($eventStatusSpan, $event['value'],
$event['clock'],
$config['event_ack_enable'] && $event['acknowledged']);
$clock = new CLink(zbx_date2str(DATE_TIME_FORMAT_SECONDS,
$event['clock']),
'tr_events.php?triggerid='.$trigger['triggerid'].'&eventid='.
$event['eventid']);
if ($enum != 0) {
$next_event_clock = $trigger['events'][$enum - 1]['clock'];
}
$triggerTable->addRow(
(new CRow([
(new CCol())->setColSpan($config['event_ack_enable'] ? 3 : 2),
(new CCol($eventStatusSpan))->setColSpan(2),
$clock,
zbx_date2age($event['clock']),
zbx_date2age($next_event_clock, $event['clock']),
$config['event_ack_enable'] ? getEventAckState($event, $page['file'])
: null,
(new CCol())->setColSpan(3)
])))
->setAttribute('data-parentid', $trigger['triggerid'])
->addStyle('display: none;');
);
}
}
}
$footer = null;
if ($config['event_ack_enable']) {
$footer = new CActionButtonList('action', 'eventids', [
'acknowledge.edit' => ['name' => _('Bulk acknowledge')]
]);
}
$triggerForm->addItem([$triggerTable, $paging, $footer]);
$triggerWidget->addItem($triggerForm)->show();
zbx_add_post_js('jqBlink.blink();');
zbx_add_post_js('var switcher = new CSwitcher(\''.
$switcherName.\'\');');
require_once dirname(__FILE__).'/include/page_footer.php';

```

1.G hosts.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/forms.inc.php';
if (hasRequest('action') && getRequest('action') == 'host.export' &&
hasRequest('hosts')) {
$page['file'] = 'zbx_export_hosts.xml';
$page['type'] = detect_page_type(PAGE_TYPE_XML);
$exportData = true;
}
else {
$page['title'] = _('Configuration of hosts');
$page['file'] = 'hosts.php';

```

```

$page['type'] = detect_page_type(PAGE_TYPE_HTML);
$page['scripts'] = ['multiselect.js'];
$exportData = false;
}
require_once dirname(__FILE__).'/include/page_header.php';
// VAR TYPE OPTIONAL FLAGS VALIDATION EXCEPTION
$fields = [
'hosts' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'groups' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'new_groups' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'hostids' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'groupids' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'applications' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'groupid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
'isset({form}) && {form} == "update"'],
'clone_hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
'isset({form}) && {form} == "full_clone"'],
'host' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'isset({add}) || isset({update})', _('Host name')],
'visiblename' => [T_ZBX_STR, O_OPT, null, null,
'isset({add}) || isset({update})'],
'description' => [T_ZBX_STR, O_OPT, null, null, null],
'proxy_hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'status' => [T_ZBX_INT, O_OPT, null,
IN([HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED]), null],
'newgroup' => [T_ZBX_STR, O_OPT, null, null,
null],
'interfaces' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'isset({add}) || isset({update})', _('Agent or SNMP or JMX or IPMI
interface')],
'mainInterfaces' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'templates' => [T_ZBX_INT, O_OPT, null, DB_ID,
null],
'add_template' => [T_ZBX_STR, O_OPT, null, null, null],
'add_templates' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'templates_rem' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'clear_templates' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'ipmi_authtype' => [T_ZBX_INT, O_OPT, null, BETWEEN(-1, 6), null],
'ipmi_privilege' => [T_ZBX_INT, O_OPT, null, BETWEEN(0, 5), null],
'ipmi_username' => [T_ZBX_STR, O_OPT, null, null, null],
'ipmi_password' => [T_ZBX_STR, O_OPT, null, null, null],
'tls_connect' => [T_ZBX_INT, O_OPT, null,
IN([HOST_ENCRYPTION_NONE, HOST_ENCRYPTION_PSK,
HOST_ENCRYPTION_CERTIFICATE]), null],
'tls_accept' => [T_ZBX_INT, O_OPT, null,
BETWEEN(0, (HOST_ENCRYPTION_NONE | HOST_ENCRYPTION_PSK |
HOST_ENCRYPTION_CERTIFICATE)),

```

```

null],
'tls_subject' => [T_ZBX_STR, O_OPT, null, null, null],
'tls_issuer' => [T_ZBX_STR, O_OPT, null, null, null],
'tls_psk_identity' => [T_ZBX_STR, O_OPT, null, null, null],
'tls_psk' => [T_ZBX_STR, O_OPT, null, null,
null],
'flags' => [T_ZBX_INT, O_OPT, null,
IN([ZBX_FLAG_DISCOVERY_NORMAL, ZBX_FLAG_DISCOVERY_CREATED]), null],
'mass_replace_tpls' => [T_ZBX_STR, O_OPT, null, null, null],
'mass_clear_tpls' => [T_ZBX_STR, O_OPT, null, null, null],
'inventory_mode' => [T_ZBX_INT, O_OPT, null,
IN(HOST_INVENTORY_DISABLED.','.HOST_INVENTORY_MANUAL.','.HOST_INVENTO
RY_AUTOMATIC), null],
'host_inventory' => [T_ZBX_STR, O_OPT, P_UNSET_EMPTY, null, null],
'macros' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'visible' => [T_ZBX_STR, O_OPT, null, null,
null],
'show_inherited_macros' => [T_ZBX_INT, O_OPT, null, IN([0,1]), null],
// actions
'action' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT,
IN("host.export","host.massdelete","host.massdisable","host.massenab
le","host.massupdate"'.
',"host.massupdateform"'.
),
null
],
'add_to_group' => [T_ZBX_INT, O_OPT, P_SYS|P_ACT, DB_ID, null],
'delete_from_group' => [T_ZBX_INT, O_OPT, P_SYS|P_ACT, DB_ID, null],
'unlink' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'unlink_and_clear' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'add' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'update' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'masssave' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'clone' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'full_clone' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'delete' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'cancel' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'form' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'form_refresh' => [T_ZBX_INT, O_OPT, null, null, null],
// filter
'filter_set' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'filter_rst' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'filter_host' => [T_ZBX_STR, O_OPT, null, null, null],
'filter_ip' => [T_ZBX_STR, O_OPT, null, null,
null],

```



```

'filter_dns' => [T_ZBX_STR, O_OPT, null, null,
null],
'filter_port' => [T_ZBX_STR, O_OPT, null, null, null],
// sort and sortorder
'sort' => [T_ZBX_STR, O_OPT, P_SYS, IN('"name","status"'),
null],
'sortorder' => [T_ZBX_STR, O_OPT, P_SYS,
IN(''.ZBX_SORT_DOWN.'", "'.ZBX_SORT_UP.'"),
null]
];
check_fields($fields);
if (getRequest('groupid') && !API::HostGroup()-
>isWritable([$REQUEST['groupid']])) {
access_deny();
}
if (getRequest('hostid') && !API::Host()-
>isWritable([$REQUEST['hostid']])) {
access_deny();
}
$hostIds = getRequest('hosts', []);
if ($exportData) {
$export = new CConfigurationExport(['hosts' => $hostIds]);
$export->setBuilder(new CConfigurationExportBuilder());
$export-
>setWriter(CExportWriterFactory::getWriter(CExportWriterFactory::XML)
);
$exportData = $export->export();
if (hasErrorMessages()) {
show_messages();
}
else {
print($exportData);
}
exit;
}
if (hasRequest('filter_set')) {
CProfile::update('web.hosts.filter_ip', getRequest('filter_ip', ''),
PROFILE_TYPE_STR);
CProfile::update('web.hosts.filter_dns', getRequest('filter_dns',
'), PROFILE_TYPE_STR);
CProfile::update('web.hosts.filter_host', getRequest('filter_host',
'), PROFILE_TYPE_STR);
CProfile::update('web.hosts.filter_port', getRequest('filter_port',
'), PROFILE_TYPE_STR);
}
elseif (hasRequest('filter_rst')) {
DBStart();
CProfile::delete('web.hosts.filter_ip');
CProfile::delete('web.hosts.filter_dns');
CProfile::delete('web.hosts.filter_host');
CProfile::delete('web.hosts.filter_port');
DBend();
}
$filter['ip'] = CProfile::get('web.hosts.filter_ip', '');

```

```

$filter['dns'] = CProfile::get('web.hosts.filter_dns', '');
$filter['host'] = CProfile::get('web.hosts.filter_host', '');
$filter['port'] = CProfile::get('web.hosts.filter_port', '');
// remove inherited macros data (actions: 'add', 'update' and 'form')
$macros = cleanInheritedMacros(getRequest('macros', []));
// remove empty new macro lines
foreach ($macros as $idx => $macro) {
if (!array_key_exists('hostmacroid', $macro) && $macro['macro'] ===
'' && $macro['value'] === '') {
unset($macros[$idx]);
}
}
if (isset($_REQUEST['add_template']) &&
isset($_REQUEST['add_templates'])) {
$_REQUEST['templates'] = getRequest('templates', []);
$_REQUEST['templates'] = array_merge($_REQUEST['templates'],
$_REQUEST['add_templates']);
}
if (isset($_REQUEST['unlink']) ||
isset($_REQUEST['unlink_and_clear'])) {
$_REQUEST['clear_templates'] = getRequest('clear_templates', []);
$unlinkTemplates = [];
if (isset($_REQUEST['unlink'])) {
// templates_rem for old style removal in massupdate form
if (isset($_REQUEST['templates_rem'])) {
$unlinkTemplates = array_keys($_REQUEST['templates_rem']);
}
elseif (is_array($_REQUEST['unlink'])) {
$unlinkTemplates = array_keys($_REQUEST['unlink']);
}
}
else {
$unlinkTemplates = array_keys($_REQUEST['unlink_and_clear']);
$_REQUEST['clear_templates'] =
array_merge($_REQUEST['clear_templates'],
$unlinkTemplates);
}
foreach ($unlinkTemplates as $templateId) {
unset($_REQUEST['templates'][array_search($templateId,
$_REQUEST['templates'])]);
}
}
elseif ((hasRequest('clone') || hasRequest('full_clone')) &&
hasRequest('hostid')) {
$_REQUEST['form'] = hasRequest('clone') ? 'clone' : 'full_clone';
$groupids = getRequest('groups', []);
if ($groupids) {
// leave only writable groups
$_REQUEST['groups'] = array_keys(API::HostGroup()->get([
'output' => [],
'groupids' => $groupids,
'editable' => true,
'preservekeys' => true
]));
}
}

```

```

}
if (hasRequest('interfaces')) {
$interfaceid = 1;
foreach ($_REQUEST['interfaces'] as &$interface) {
$interface['interfaceid'] = (string) $interfaceid++;
unset($interface['locked'], $interface['items']);
}
unset($interface);
}
if (hasRequest('full_clone')) {
$_REQUEST['clone_hostid'] = $_REQUEST['hostid'];
}
unset($_REQUEST['hostid'], $_REQUEST['flags']);
}
elseif (hasRequest('action') && getRequest('action') ==
'host.massupdate' && hasRequest('masssave')) {
$hostIds = getRequest('hosts', []);
$visible = getRequest('visible', []);
$_REQUEST['proxy_hostid'] = getRequest('proxy_hostid', 0);
$_REQUEST['templates'] = getRequest('templates', []);
try {
DBstart();
// filter only normal and discovery created hosts
$hosts = API::Host()->get([
'output' => ['hostid'],
'hostids' => $hostIds,
'selectInventory' => ['inventory_mode'],
'filter' => ['flags' => [ZBX_FLAG_DISCOVERY_NORMAL,
ZBX_FLAG_DISCOVERY_CREATED]]
]);
$properties = [
'proxy_hostid', 'ipmi_authtype', 'ipmi_privilege', 'ipmi_username',
'ipmi_password', 'description'
];
$newValues = [];
foreach ($properties as $property) {
if (isset($visible[$property])) {
$newValues[$property] = $_REQUEST[$property];
}
}
if (isset($visible['status'])) {
$newValues['status'] = getRequest('status',
HOST_STATUS_NOT_MONITORED);
}
if (array_key_exists('encryption', $visible)) {
$newValues['tls_connect'] = getRequest('tls_connect',
HOST_ENCRYPTION_NONE);
$newValues['tls_accept'] = getRequest('tls_accept',
HOST_ENCRYPTION_NONE);
if ($newValues['tls_connect'] == HOST_ENCRYPTION_PSK ||
($newValues['tls_accept'] & HOST_ENCRYPTION_PSK)) {
$newValues['tls_psk_identity'] = getRequest('tls_psk_identity', '');
$newValues['tls_psk'] = getRequest('tls_psk', '');
}
}
}

```

```

if ($newValues['tls_connect'] == HOST_ENCRYPTION_CERTIFICATE
|| ($newValues['tls_accept'] & HOST_ENCRYPTION_CERTIFICATE))
{
$newValues['tls_issuer'] = getRequest('tls_issuer', '');
$newValues['tls_subject'] = getRequest('tls_subject', '');
}
}
$templateids = [];
if (isset($visible['templates'])) {
$templateids = $_REQUEST['templates'];
}
// add new or existing host groups
$newHostGroupIds = [];
if (isset($visible['new_groups']) && !empty($_REQUEST['new_groups']))
{
if (CWebUser::getType() == USER_TYPE_SUPER_ADMIN) {
foreach ($_REQUEST['new_groups'] as $newGroup) {
if (is_array($newGroup) && isset($newGroup['new'])) {
$newGroups[] = ['name' => $newGroup['new']];
}
else {
$newHostGroupIds[] = $newGroup;
}
}
if (isset($newGroups)) {
if (!$createdGroups = API::HostGroup()->create($newGroups))
{
throw new Exception();
}
}
$newHostGroupIds = $newHostGroupIds
? array_merge($newHostGroupIds,
$createdGroups['groupids'])
: $createdGroups['groupids'];
}
}
else {
$newHostGroupIds = getRequest('new_groups');
}
}
if (isset($visible['groups'])) {
if (isset($_REQUEST['groups'])) {
$replaceHostGroupIds = $newHostGroupIds
? array_unique(array_merge(getRequest('groups'),
$newHostGroupIds))
: $_REQUEST['groups'];
}
elseif ($newHostGroupIds) {
$replaceHostGroupIds = $newHostGroupIds;
}
if (isset($replaceHostGroupIds)) {
$newValues['groups'] = API::HostGroup()->get([
'groupids' => $replaceHostGroupIds,
'editable' => true,
'output' => ['groupid']

```



```

}
$add = [];
if ($templateids && isset($visible['templates'])) {
$add['templates'] = $templateids;
}
// add new host groups
if ($newHostGroupIds && (!isset($visible['groups']) || !
isset($replaceHostGroups))) {
$add['groups'] = zbx_toObject($newHostGroupIds, 'groupid');
}
if ($add) {
$hostsids = zbx_objectValues($hosts, 'hostid');
$add['hosts'] = zbx_toObject($hostsids, 'hostid');
$result = API::Host()->massAdd($add);
if ($result === false) {
throw new Exception();
}
}
DBend(true);
uncheckTableRows();
show_message(_('Hosts updated'));
unset($_REQUEST['masssave'], $_REQUEST['form'], $_REQUEST['hosts']);
}
catch (Exception $e) {
DBend(false);
show_error_message(_('Cannot update hosts'));
}
}
elseif (hasRequest('add') || hasRequest('update')) {
try {
DBstart();
$hostId = getRequest('hostid', 0);
if ($hostId != 0) {
$create = false;
$msgOk = _('Host updated');
$msgFail = _('Cannot update host');
$dbHost = API::Host()->get([
'output' => API_OUTPUT_EXTEND,
'hostids' => $hostId,
'editable' => true
]);
$dbHost = reset($dbHost);
}
else {
$create = true;
$msgOk = _('Host added');
$msgFail = _('Cannot add host');
}
// host data
if (!$create && $dbHost['flags'] == ZBX_FLAG_DISCOVERY_CREATED) {
$host = [
'hostid' => $hostId,
'status' => getRequest('status', HOST_STATUS_NOT_MONITORED),
'description' => getRequest('description', ''),

```

```

'inventory' => (getRequest('inventory_mode') ==
HOST_INVENTORY_DISABLED)
? []
: getRequest('host_inventory', [])
];
}
else {
// templates
$templates = [];
foreach (getRequest('templates', []) as $templateId) {
$templates[] = ['templateid' => $templateId];
}
// interfaces
$interfaces = getRequest('interfaces', []);
foreach ($interfaces as $key => $interface) {
if (zbx_empty($interface['ip']) && zbx_empty($interface['dns'])) {
unset($interface[$key]);
continue;
}
if ($interface['type'] == INTERFACE_TYPE_SNMP &&
!isset($interface['bulk'])) {
$interfaces[$key]['bulk'] = SNMP_BULK_DISABLED;
}
else {
$interfaces[$key]['bulk'] = SNMP_BULK_ENABLED;
}
if ($interface['isNew']) {
unset($interfaces[$key]['interfaceid']);
}
unset($interfaces[$key]['isNew']);
$interfaces[$key]['main'] = 0;
}
$mainInterfaces = getRequest('mainInterfaces', []);
foreach ([INTERFACE_TYPE_AGENT, INTERFACE_TYPE_SNMP,
INTERFACE_TYPE_JMX,
INTERFACE_TYPE_IPMI] as $type) {
if (array_key_exists($type, $mainInterfaces)) {
$interfaces[$mainInterfaces[$type]]['main'] =
INTERFACE_PRIMARY;
}
}
// new group
$groups = getRequest('groups', []);
$newGroup = getRequest('newgroup');
if (!zbx_empty($newGroup)) {
$newGroup = API::HostGroup()->create(['name' => $newGroup]);
if (!$newGroup) {
throw new Exception();
}
}
$groups[] = reset($newGroup['groupids']);
}
$groups = zbx_toObject($groups, 'groupid');
// host data
$host = [

```

```

'host' => getRequest('host'),
'name' => getRequest('visiblename'),
'status' => getRequest('status', HOST_STATUS_NOT_MONITORED),
'description' => getRequest('description'),
'proxy_hostid' => getRequest('proxy_hostid', 0),
'ipmi_authtype' => getRequest('ipmi_authtype'),
'ipmi_privilege' => getRequest('ipmi_privilege'),
'ipmi_username' => getRequest('ipmi_username'),
'ipmi_password' => getRequest('ipmi_password'),
'tls_connect' => getRequest('tls_connect', HOST_ENCRYPTION_NONE),
'tls_accept' => getRequest('tls_accept', HOST_ENCRYPTION_NONE),
'groups' => $groups,
'templates' => $templates,
'interfaces' => $interfaces,
'macros' => $macros,
'inventory_mode' => getRequest('inventory_mode'),
'inventory' => (getRequest('inventory_mode') ==
HOST_INVENTORY_DISABLED)
? []
: getRequest('host_inventory', [])
];
if ($host['tls_connect'] == HOST_ENCRYPTION_PSK ||
($host['tls_accept'] &
HOST_ENCRYPTION_PSK)) {
$host['tls_psk_identity'] = getRequest('tls_psk_identity', '');
$host['tls_psk'] = getRequest('tls_psk', '');
}
if ($host['tls_connect'] == HOST_ENCRYPTION_CERTIFICATE
|| ($host['tls_accept'] & HOST_ENCRYPTION_CERTIFICATE)) {
$host['tls_issuer'] = getRequest('tls_issuer', '');
$host['tls_subject'] = getRequest('tls_subject', '');
}
if (!$create) {
$host['templates_clear'] = zbx_toObject(getRequest('clear_templates',
[]), 'templateid');
}
if ($create) {
$hostIds = API::Host()->create($host);
if ($hostIds) {
$hostId = reset($hostIds['hostids']);
}
else {
throw new Exception();
}
add_audit_ext(AUDIT_ACTION_ADD, AUDIT_RESOURCE_HOST, $hostId,
$host['host'],
null, null, null);
}
else {
$host['hostid'] = $hostId;
if (!API::Host()->update($host)) {
throw new Exception();
}
}

```



```

$dbHostNew = API::Host()->get([
'output' => API_OUTPUT_EXTEND,
'hostids' => $hostId,
'editable' => true
]);
$dbHostNew = reset($dbHostNew);
add_audit_ext(AUDIT_ACTION_UPDATE, AUDIT_RESOURCE_HOST,
$dbHostNew['hostid'],
$dbHostNew['host'], 'hosts',
$dbHost, $dbHostNew);
}
// full clone
if (getRequest('form', '') === 'full_clone' &&
getRequest('clone_hostid', 0) != 0) {
$srcHostId = getRequest('clone_hostid');
// copy applications
if (!copyApplications($srcHostId, $hostId)) {
throw new Exception();
}
// copy items
if (!copyItems($srcHostId, $hostId)) {
throw new Exception();
}
// copy web scenarios
if (!copyHttpTests($srcHostId, $hostId)) {
throw new Exception();
}
// copy triggers
$dbTriggers = API::Trigger()->get([
'output' => ['triggerid'],
'hostids' => $srcHostId,
'inherited' => false
]);
if ($dbTriggers) {
if (!copyTriggersToHosts(zbx_objectValues($dbTriggers, 'triggerid'),
$hostId, $srcHostId)) {
throw new Exception();
}
}
// copy discovery rules
$dbDiscoveryRules = API::DiscoveryRule()->get([
'output' => ['itemid'],
'hostids' => $srcHostId,
'inherited' => false
]);
if ($dbDiscoveryRules) {
$copyDiscoveryRules = API::DiscoveryRule()->copy([
'discoveryids' => zbx_objectValues($dbDiscoveryRules,
'itemid'),
'hostids' => [$hostId]
]);
if (!$copyDiscoveryRules) {
throw new Exception();
}
}
}

```

```

}
// copy graphs
$dbGraphs = API::Graph()->get([
'output' => API_OUTPUT_EXTEND,
'selectHosts' => ['hostid'],
'selectItems' => ['type'],
'hostids' => $srcHostId,
'filter' => ['flags' => ZBX_FLAG_DISCOVERY_NORMAL],
'inherited' => false
]);
foreach ($dbGraphs as $dbGraph) {
if (count($dbGraph['hosts']) > 1) {
continue;
}
if (httpItemExists($dbGraph['items'])) {
continue;
}
if (!copyGraphToHost($dbGraph['graphid'], $hostId)) {
throw new Exception();
}
}
}
$result = DBend(true);
if ($result) {
uncheckTableRows();
}
show_messages($result, $msgOk, $msgFail);
unset($_REQUEST['form'], $_REQUEST['hostid']);
}
catch (Exception $e) {
DBend(false);
show_messages(false, $msgOk, $msgFail);
}
}
elseif (hasRequest('delete') && hasRequest('hostid')) {
DBstart();
$result = API::Host()->delete([getRequest('hostid')]);
$result = DBend($result);
if ($result) {
unset($_REQUEST['form'], $_REQUEST['hostid']);
uncheckTableRows();
}
show_messages($result, _('Host deleted'), _('Cannot delete host'));
unset($_REQUEST['delete']);
}
elseif (hasRequest('action') && getRequest('action') ==
'host.massdelete' && hasRequest('hosts')) {
DBstart();
$result = API::Host()->delete(getRequest('hosts'));
$result = DBend($result);
if ($result) {
uncheckTableRows();
}
show_messages($result, _('Host deleted'), _('Cannot delete host'));
}

```

```

}
elseif (hasRequest('action') && str_in_array(getRequest('action'),
['host.massenable', 'host.massdisable']) &&
hasRequest('hosts')) {
$enable =(getRequest('action') == 'host.massenable');
$status = $enable ? TRIGGER_STATUS_ENABLED : TRIGGER_STATUS_DISABLED;
$sactHosts = API::Host()->get([
'hostids' => getRequest('hosts'),
'editable' => true,
'templated_hosts' => true,
'output' => ['hostid']
]);
$sactHosts = zbx_objectValues($sactHosts, 'hostid');
if ($sactHosts) {
DBstart();
$result = updateHostStatus($sactHosts, $status);
$result = DBend($result);
if ($result) {
uncheckTableRows();
}
$updateCount = count($sactHosts);
$messageSuccess = $enable
? _n('Host enabled', 'Hosts enabled', $updateCount)
: _n('Host disabled', 'Hosts disabled', $updateCount);
$messageFailed = $enable
? _n('Cannot enable host', 'Cannot enable hosts', $updateCount)
: _n('Cannot disable host', 'Cannot disable hosts', $updateCount);
show_messages($result, $messageSuccess, $messageFailed);
}
}
$pageFilter = new CPageFilter([
'groups' => [
'real_hosts' => true,
'editable' => true
],
'groupid' => getRequest('groupid')
]);
$_REQUEST['groupid'] = $pageFilter->groupid;
$_REQUEST['hostid'] = getRequest('hostid', 0);
$config = select_config();
if ((getRequest('action') === 'host.massupdateform' ||
hasRequest('masssave')) && hasRequest('hosts')) {
$data = [
'hosts' => getRequest('hosts'),
'visible' => getRequest('visible', []),
'mass_replace_tpls' => getRequest('mass_replace_tpls'),
'mass_clear_tpls' => getRequest('mass_clear_tpls'),
'groups' => getRequest('groups', []),
'newgroup' => getRequest('newgroup', ''),
'status' => getRequest('status', HOST_STATUS_MONITORED),
'description' => getRequest('description'),
'proxy_hostid' => getRequest('proxy_hostid', ''),
'ipmi_authtype' => getRequest('ipmi_authtype',
IPMI_AUTHTYPE_DEFAULT),

```

```

'ipmi_privilege' => getRequest('ipmi_privilege',
IPMI_PRIVILEGE_USER),
'ipmi_username' => getRequest('ipmi_username', ''),
'ipmi_password' => getRequest('ipmi_password', ''),
'inventory_mode' => getRequest('inventory_mode',
HOST_INVENTORY_DISABLED),
'host_inventory' => getRequest('host_inventory', []),
'templates' => getRequest('templates', []),
'inventories' => zbx_toHash(getHostInventories(), 'db_field'),
'tls_connect' => getRequest('tls_connect', HOST_ENCRYPTION_NONE),
'tls_accept' => getRequest('tls_accept', HOST_ENCRYPTION_NONE),
'tls_issuer' => getRequest('tls_issuer', ''),
'tls_subject' => getRequest('tls_subject', ''),
'tls_psk_identity' => getRequest('tls_psk_identity', ''),
'tls_psk' => getRequest('tls_psk', '')
];
// sort templates
natsort($data['templates']);
// get groups
$data['all_groups'] = API::HostGroup()->get([
'output' => API_OUTPUT_EXTEND,
'editable' => true
]);
order_result($data['all_groups'], 'name');
// get proxies
$data['proxies'] = DBfetchArray(DBselect(
'SELECT h.hostid,h.host'.
' FROM hosts h'.
' WHERE h.status IN
('HOST_STATUS_PROXY_ACTIVE.', '.HOST_STATUS_PROXY_PASSIVE.')'
));
order_result($data['proxies'], 'host');
// get templates data
$data['linkedTemplates'] = null;
if (!empty($data['templates'])) {
$getLinkedTemplates = API::Template()->get([
'templateids' => $data['templates'],
'output' => ['templateid', 'name']
]);
foreach ($getLinkedTemplates as $getLinkedTemplate) {
$data['linkedTemplates'][] = [
'id' => $getLinkedTemplate['templateid'],
'name' => $getLinkedTemplate['name']
];
}
}
$hostView = new CView('configuration.host.massupdate', $data);
}
elseif (hasRequest('form')) {
$data = [
// Common & auxiliary
'form' => getRequest('form', ''),
'hostid' => getRequest('hostid', 0),
'clone_hostid' => getRequest('clone_hostid', 0),

```

```

'groupid' => getRequest('groupid', 0),
'flags' => getRequest('flags', ZBX_FLAG_DISCOVERY_NORMAL),
// Host
'host' => getRequest('host', ''),
'visiblename' => getRequest('visiblename', ''),
'groups' => getRequest('groups', []),
'newgroup' => getRequest('newgroup', ''),
'interfaces' => getRequest('interfaces', []),
'mainInterfaces' => getRequest('mainInterfaces', []),
'description' => getRequest('description', ''),
'proxy_hostid' => getRequest('proxy_hostid', 0),
'status' => getRequest('status', HOST_STATUS_NOT_MONITORED),
// Templates
'templates' => getRequest('templates', []),
'clear_templates' => getRequest('clear_templates', []),
'original_templates' => [],
'linked_templates' => [],
// IPMI
'ipmi_authtype' => getRequest('ipmi_authtype',
IPMI_AUTHTYPE_DEFAULT),
'ipmi_privilege' => getRequest('ipmi_privilege',
IPMI_PRIVILEGE_USER),
'ipmi_username' => getRequest('ipmi_username', ''),
'ipmi_password' => getRequest('ipmi_password', ''),
// Macros
'macros' => $macros,
'show_inherited_macros' => getRequest('show_inherited_macros', 0),
// Host inventory
'inventory_mode' => getRequest('inventory_mode',
$config['default_inventory_mode']),
'host_inventory' => getRequest('host_inventory', []),
'inventory_items' => [],
// Encryption
'tls_connect' => getRequest('tls_connect', HOST_ENCRYPTION_NONE),
'tls_accept' => getRequest('tls_accept', HOST_ENCRYPTION_NONE),
'tls_issuer' => getRequest('tls_issuer', ''),
'tls_subject' => getRequest('tls_subject', ''),
'tls_psk_identity' => getRequest('tls_psk_identity', ''),
'tls_psk' => getRequest('tls_psk', '')
];
if (!hasRequest('form_refresh')) {
if ($data['hostid'] != 0) {
$dbHosts = API::Host()->get([
'output' => ['hostid', 'proxy_hostid', 'host', 'name', 'status',
'ipmi_authtype', 'ipmi_privilege',
'ipmi_username', 'ipmi_password', 'flags', 'description',
'tls_connect', 'tls_accept', 'tls_issuer',
'tls_subject', 'tls_psk_identity', 'tls_psk'
],
'selectGroups' => ['groupid'],
'selectParentTemplates' => ['templateid'],
'selectMacros' => ['hostmacroid', 'macro', 'value'],
'selectDiscoveryRule' => ['itemid', 'name'],
'selectInventory' => true,

```

```

'hostids' => [$data['hostid']]
]);
$dbHost = reset($dbHosts);
$data['flags'] = $dbHost['flags'];
if ($data['flags'] == ZBX_FLAG_DISCOVERY_CREATED) {
$data['discoveryRule'] = $dbHost['discoveryRule'];
}
// Host
$data['host'] = $dbHost['host'];
$data['visiblename'] = $dbHost['name'];
$data['groups'] = zbx_objectValues($dbHost['groups'], 'groupid');
$data['interfaces'] = API::HostInterface()->get([
'output' => ['interfaceid', 'main', 'type', 'useip', 'ip', 'dns',
'port', 'bulk'],
'selectItems' => ['type'],
'hostids' => [$data['hostid']],
'sortfield' => 'interfaceid'
]);
$data['description'] = $dbHost['description'];
$data['proxy_hostid'] = $dbHost['proxy_hostid'];
$data['status'] = $dbHost['status'];
// Templates
$data['templates'] = zbx_objectValues($dbHost['parentTemplates'],
'templateid');
$data['original_templates'] = array_combine($data['templates'],
$data['templates']);
// IPMI
$data['ipmi_authtype'] = $dbHost['ipmi_authtype'];
$data['ipmi_privilege'] = $dbHost['ipmi_privilege'];
$data['ipmi_username'] = $dbHost['ipmi_username'];
$data['ipmi_password'] = $dbHost['ipmi_password'];
// Macros
$data['macros'] = $dbHost['macros'];
// Interfaces
foreach ($data['interfaces'] as &$interface) {
if ($data['flags'] == ZBX_FLAG_DISCOVERY_CREATED) {
$interface['locked'] = true;
}
else {
// check if interface has items that require specific
interface type, if so type cannot be changed
$interface['locked'] = false;
foreach ($interface['items'] as $item) {
$type = itemTypeInterface($item['type']);
if ($type !== false && $type !=
INTERFACE_TYPE_ANY) {
$interface['locked'] = true;
break;
}
}
}
$interface['items'] = (bool) $interface['items'];
}
unset($interface);

```

```

// Host inventory
$data['inventory_mode'] = array_key_exists('inventory_mode',
$dbHost['inventory'])
? $dbHost['inventory']['inventory_mode']
: HOST_INVENTORY_DISABLED;
$data['host_inventory'] = $dbHost['inventory'];
unset($data['host_inventory']['inventory_mode']);
// Encryption
$data['tls_connect'] = $dbHost['tls_connect'];
$data['tls_accept'] = $dbHost['tls_accept'];
$data['tls_issuer'] = $dbHost['tls_issuer'];
$data['tls_subject'] = $dbHost['tls_subject'];
$data['tls_psk_identity'] = $dbHost['tls_psk_identity'];
$data['tls_psk'] = $dbHost['tls_psk'];
// display empty visible name if equal to host name
if ($data['host'] === $data['visiblename']) {
$data['visiblename'] = '';
}
}
else {
$data['status'] = HOST_STATUS_MONITORED;
}
if (!$data['groups'] && $data['groupid'] != 0) {
$data['groups'][] = $data['groupid'];
}
}
else {
if ($data['hostid'] != 0) {
$dbHosts = API::Host()->get([
'output' => ['flags'],
'selectParentTemplates' => ['templateid'],
'selectDiscoveryRule' => ['itemid', 'name'],
'hostids' => [$data['hostid']]
]);
$dbHost = reset($dbHosts);
$data['flags'] = $dbHost['flags'];
if ($data['flags'] == ZBX_FLAG_DISCOVERY_CREATED) {
$data['discoveryRule'] = $dbHost['discoveryRule'];
}
$templateids = zbx_objectValues($dbHost['parentTemplates'],
'templateid');
$data['original_templates'] = array_combine($templateids,
$templateids);
}
foreach ([INTERFACE_TYPE_AGENT, INTERFACE_TYPE_SNMP,
INTERFACE_TYPE_JMX,
INTERFACE_TYPE_IPMI] as $type) {
if (array_key_exists($type, $data['mainInterfaces'])) {
$interfaceid = $data['mainInterfaces'][$type];
$data['interfaces'][$interfaceid]['main'] = '1';
}
}
}
}
}

```

```

if ($data['hostid'] != 0) {
// get items that populate host inventory fields
$data['inventory_items'] = API::Item()->get([
'output' => ['inventory_link', 'itemid', 'hostid', 'name', 'key_'],
'hostids' => [$dbHost['hostid']],
'filter' => ['inventory_link' => array_keys(getHostInventories())]
]);
$data['inventory_items'] = zbx_toHash($data['inventory_items'],
'inventory_link');
$data['inventory_items'] =
CMacrosResolverHelper::resolveItemNames($data['inventory_items']);
}
if ($data['flags'] == ZBX_FLAG_DISCOVERY_CREATED) {
if ($data['proxy_hostid'] != 0) {
$data['proxies'] = API::Proxy()->get([
'output' => ['host'],
'proxyids' => [$data['proxy_hostid']],
'preservekeys' => true
]);
}
else {
$data['proxies'] = [];
}
}
else {
$data['proxies'] = API::Proxy()->get([
'output' => ['host'],
'preservekeys' => true
]);
order_result($data['proxies'], 'host');
}
foreach ($data['proxies'] as &$proxy) {
$proxy = $proxy['host'];
}
unset($proxy);
if ($data['show_inherited_macros']) {
$data['macros'] = mergeInheritedMacros($data['macros'],
getInheritedMacros($data['templates']));
}
$data['macros'] = array_values(order_macros($data['macros'],
'macro'));
if (!$data['macros'] && $data['flags'] != ZBX_FLAG_DISCOVERY_CREATED)
{
$macro = ['macro' => '', 'value' => ''];
if ($data['show_inherited_macros']) {
$macro['type'] = MACRO_TYPE_HOSTMACRO;
}
$data['macros'][] = $macro;
}
// groups with RW permissions
$data['groupsAllowed'] = API::HostGroup()->get([
'output' => [],
'editable' => true,
'preservekeys' => true

```



```

]);
// all available groups
$data['groupsAll'] = API::HostGroup()->get(['output' => ['groupid',
'name']]);
CArrayHelper::sort($data['groupsAll'], ['name']);
if ($data['templates']) {
$data['linked_templates'] = API::Template()->get([
'output' => ['templateid', 'name'],
'templateids' => $data['templates']
]);
CArrayHelper::sort($data['linked_templates'], ['name']);
$data['writable_templates'] = API::Template()->get([
'output' => ['templateid'],
'templateids' => $data['templates'],
'editable' => true,
'preservekeys' => true
]);
}
$hostView = new CView('configuration.host.edit', $data);
}
else {
$sortField = getRequest('sort', CProfile::get('web.'.
$page['file'].'.sort', 'name'));
$sortOrder = getRequest('sortorder', CProfile::get('web.'.
$page['file'].'.sortorder', ZBX_SORT_UP));
CProfile::update('web.'. $page['file'].'.sort', $sortField,
PROFILE_TYPE_STR);
CProfile::update('web.'. $page['file'].'.sortorder', $sortOrder,
PROFILE_TYPE_STR);
// get Hosts
$hosts = [];
if ($pageFilter->groupsSelected) {
$hosts = API::Host()->get([
'output' => ['hostid', $sortField],
'groupids' => ($pageFilter->groupid > 0) ? $pageFilter->groupid :
null,
'editable' => true,
'sortfield' => $sortField,
'limit' => $config['search_limit'] + 1,
'search' => [
'name' => ($filter['host'] === '') ? null : $filter['host'],
'ip' => ($filter['ip'] === '') ? null : $filter['ip'],
'dns' => ($filter['dns'] === '') ? null : $filter['dns']
],
'filter' => [
'port' => ($filter['port'] === '') ? null : $filter['port']
]
]);
}
order_result($hosts, $sortField, $sortOrder);
$url = (new CUrl('hosts.php'))
->setArgument('groupid', $pageFilter->groupid);
$pageLine = getPagingLine($hosts, $sortOrder, $url);
$hosts = API::Host()->get([

```

```

'hostids' => zbx_objectValues($hosts, 'hostid'),
'output' => API_OUTPUT_EXTEND,
'selectParentTemplates' => ['hostid', 'name'],
'selectInterfaces' => API_OUTPUT_EXTEND,
'selectItems' => API_OUTPUT_COUNT,
'selectDiscoveries' => API_OUTPUT_COUNT,
'selectTriggers' => API_OUTPUT_COUNT,
'selectGraphs' => API_OUTPUT_COUNT,
'selectApplications' => API_OUTPUT_COUNT,
'selectHttpTests' => API_OUTPUT_COUNT,
'selectDiscoveryRule' => ['itemid', 'name'],
'selectHostDiscovery' => ['ts_delete']
]);
order_result($hosts, $sortField, $sortOrder);
// selecting linked templates to templates linked to hosts
$templateids = [];
foreach ($hosts as $host) {
$templateids = array_merge($templateids,
zbx_objectValues($host['parentTemplates'],
'templateid'));
}
$templateids = array_keys(array_flip($templateids));
$templates = API::Template()->get([
'output' => ['templateid', 'name'],
'templateids' => $templateids,
'selectParentTemplates' => ['hostid', 'name'],
'preservekeys' => true
]);
// selecting writable templates IDs
$writable_templates = [];
if ($templateids) {
foreach ($templates as $template) {
$templateids = array_merge($templateids,
zbx_objectValues($template['parentTemplates'], 'templateid'));
}
$writable_templates = API::Template()->get([
'output' => ['templateid'],
'templateids' => array_keys(array_flip($templateids)),
'editable' => true,
'preservekeys' => true
]);
}
// get proxy host IDs that that are not 0
$proxyHostIds = [];
foreach ($hosts as &$host) {
// Sort interfaces to be listed starting with one selected as 'main'.
CArrayHelper::sort($host['interfaces'], [
['field' => 'main', 'order' => ZBX_SORT_DOWN]
]);
if ($host['proxy_hostid']) {
$proxyHostIds[$host['proxy_hostid']] = $host['proxy_hostid'];
}
}
unset($host);

```

```

$proxies = [];
if ($proxyHostIds) {
$proxies = API::Proxy()->get([
'proxyids' => $proxyHostIds,
'output' => ['host'],
'preservekeys' => true
]);
}
$data = [
'pageFilter' => $pageFilter,
'hosts' => $hosts,
'paging' => $pagingLine,
'filter' => $filter,
'sortField' => $sortField,
'sortOrder' => $sortOrder,
'groupId' => $pageFilter->groupid,
'config' => $config,
'templates' => $templates,
'writable_templates' => $writable_templates,
'proxies' => $proxies
];
$hostView = new CView('configuration.host.list', $data);
}
$hostView->render();
$hostView->show();
require_once dirname(__FILE__).'/include/page_footer.php';

```

1.H triggers.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/hosts.inc.php';
require_once dirname(__FILE__).'/include/triggers.inc.php';
require_once dirname(__FILE__).'/include/forms.inc.php';
$page['title'] = _('Configuration of triggers');
$page['file'] = 'triggers.php';
require_once dirname(__FILE__).'/include/page_header.php';
28a
// VAR TYPE OPTIONAL FLAGS VALIDATION EXCEPTION
$fields = [
'groupid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'hostid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
null],
'triggerid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID,
'isset({form}) && ({form} == "update")'],
'copy_type' => [T_ZBX_INT, O_OPT, P_SYS, IN([COPY_TYPE_TO_HOST,
COPY_TYPE_TO_TEMPLATE, COPY_TYPE_TO_HOST_GROUP]), 'isset({copy})'],
'copy_mode' => [T_ZBX_INT, O_OPT, P_SYS, IN('0'), null],
'type' => [T_ZBX_INT, O_OPT, null, IN('0,1'), null],
'description' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY, 'isset({add}) ||
isset({update})', _('Name')],
'expression' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'isset({add}) || isset({update})', _('Expression')],
'priority' => [T_ZBX_INT, O_OPT, null, IN('0,1,2,3,4,5'),

```

```

'isset({add}) || isset({update})'],
'comments' => [T_ZBX_STR, O_OPT, null, null,
'isset({add}) || isset({update})'],
'url' => [T_ZBX_STR, O_OPT, null, null,
'isset({add}) || isset({update})'],
'status' => [T_ZBX_STR, O_OPT, null, null,
null],
'input_method' => [T_ZBX_INT, O_OPT, null, NOT_EMPTY,
'isset({toggle_input_method})'],
'expr_temp' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'(isset({add_expression}) || isset({and_expression}) ||
isset({or_expression}) ||
isset({replace_expression}))', _('Expression')],
'expr_target_single' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'(isset({and_expression}) ||
isset({or_expression}) || isset({replace_expression}))'],
'dependencies' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'new_dependency' => [T_ZBX_INT, O_OPT, null, DB_ID.'{>0',
'isset({add_dependency})'],
'g_triggerid' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'copy_targetid' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'copy_groupid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, 'isset({copy}) &&
(isset({copy_type}) && {copy_type} == 0)'],
'visible' => [T_ZBX_STR, O_OPT, null, null, null],
// filter
'filter_set' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'filter_rst' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'filter_priority' => [T_ZBX_INT, O_OPT, null,
IN([
-1, TRIGGER_SEVERITY_NOT_CLASSIFIED, TRIGGER_SEVERITY_INFORMATION,
TRIGGER_SEVERITY_WARNING,
TRIGGER_SEVERITY_AVERAGE, TRIGGER_SEVERITY_HIGH,
TRIGGER_SEVERITY_DISASTER
]), null
],
'filter_state' => [T_ZBX_INT, O_OPT, null, IN([-1,
TRIGGER_STATE_NORMAL,
TRIGGER_STATE_UNKNOWN]), null],
'filter_status' => [T_ZBX_INT, O_OPT, null,
IN([-1, TRIGGER_STATUS_ENABLED, TRIGGER_STATUS_DISABLED]), null
],
// actions
'action' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT,
IN('"trigger.masscopyto", "trigger.massdelete", "trigger.massdisable", '
.
'"trigger.massenable", "trigger.massupdate", "trigger.massupdateform"
),
null
],
'toggle_input_method' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'add_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'and_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'or_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'replace_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],

```

```

'remove_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'test_expression' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'add_dependency' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'group_enable' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'group_disable' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'group_delete' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'copy' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'clone' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'add' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'update' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'massupdate' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'delete' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'cancel' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'form' => [T_ZBX_STR, O_OPT, P_SYS, null, null],
'form_refresh' => [T_ZBX_INT, O_OPT, null, null, null],
// sort and sortorder
'sort' => [T_ZBX_STR, O_OPT, P_SYS,
IN("description","priority","status"), null],
'sortorder' => [T_ZBX_STR, O_OPT, P_SYS,
IN("'.ZBX_SORT_DOWN.'","'.ZBX_SORT_UP.'"), null]
];
check_fields($fields);
$_REQUEST['status'] = isset($_REQUEST['status']) ?
TRIGGER_STATUS_ENABLED : TRIGGER_STATUS_DISABLED;
29a
$_REQUEST['type'] = isset($_REQUEST['type']) ?
TRIGGER_MULT_EVENT_ENABLED : TRIGGER_MULT_EVENT_DISABLED;
// Validate permissions to single trigger.
$triggerId = getRequest('triggerid');
if ($triggerId !== null) {
$trigger = API::Trigger()->get([
'output' => ['triggerid'],
'triggerids' => [$triggerId],
'filter' => ['flags' => ZBX_FLAG_DISCOVERY_NORMAL],
'editable' => true
]);
if (!$trigger) {
access_deny();
}
}
// Validate permissions to a group of triggers for mass
enable/disable actions.
$triggerIds = getRequest('g_triggerid', []);
$triggerIds = zbx_toArray($triggerIds);
if ($triggerIds && !API::Trigger()->isWritable($triggerIds)) {
access_deny();
}
// Validate permissions to group.
$groupId = getRequest('groupid');
if ($groupId && !API::HostGroup()->isWritable([$groupId])) {
access_deny();
}
// Validate permissions to host.

```

```

$hostId = getRequest('hostid');
if ($hostId && !API::Host()->isWritable([$hostId])) {
access_deny();
}
$exprAction = null;
if (isset($_REQUEST['add_expression'])) {
$_REQUEST['expression'] = $_REQUEST['expr_temp'];
$_REQUEST['expr_temp'] = '';
}
elseif (isset($_REQUEST['and_expression'])) {
$exprAction = 'and';
}
elseif (isset($_REQUEST['or_expression'])) {
$exprAction = 'or';
}
elseif (isset($_REQUEST['replace_expression'])) {
$exprAction = 'r';
}
elseif (getRequest('remove_expression')) {
$exprAction = 'R';
$_REQUEST['expr_target_single'] = $_REQUEST['remove_expression'];
}
elseif (isset($_REQUEST['clone']) && isset($_REQUEST['triggerid'])) {
unset($_REQUEST['triggerid']);
$_REQUEST['form'] = 'clone';
}
elseif (hasRequest('add') || hasRequest('update')) {
$trigger = [
'expression' => getRequest('expression'),
'description' => getRequest('description'),
'url' => getRequest('url'),
'status' => getRequest('status'),
'priority' => getRequest('priority'),
'comments' => getRequest('comments'),
'type' => getRequest('type'),
'dependencies' => zbx_toObject(getRequest('dependencies', []),
'triggerid')
];
if (hasRequest('update')) {
// update only changed fields
$sold_triggers = API::Trigger()->get([
'output' => ['expression', 'description', 'url', 'status',
'priority',
'comments', 'type'],
'selectDependencies' => ['triggerid'],
'triggerids' => getRequest('triggerid')
]);
if (!$sold_triggers) {
access_deny();
}
$sold_triggers =
CMacrosResolverHelper::resolveTriggerExpressions($sold_triggers);
$sold_trigger = reset($sold_triggers);

```

```

Sold_trigger['dependencies'] =
zbx_toHash(zbx_objectValues($old_trigger['dependencies'],
'triggerid'));
$newDependencies = $trigger['dependencies'];
$oldDependencies = $old_trigger['dependencies'];
unset($trigger['dependencies'], $old_trigger['dependencies']);
$triggerToUpdate = array_diff_assoc($trigger, $old_trigger);
$triggerToUpdate['triggerid'] = getRequest('triggerid');
// dependencies
$updateDependencies = false;
if (count($newDependencies) != count($oldDependencies)) {
$updateDependencies = true;
}
else {
foreach ($newDependencies as $dependency) {
if (!isset($oldDependencies[$dependency['triggerid']])) {
$updateDependencies = true;
}
}
}
if ($updateDependencies) {
$triggerToUpdate['dependencies'] = $newDependencies;
}
$result = API::Trigger()->update($triggerToUpdate);
show_messages($result, _('Trigger updated'), _('Cannot update
trigger'));
}
else {
$result = API::Trigger()->create($trigger);
show_messages($result, _('Trigger added'), _('Cannot add trigger'));
}
if ($result) {
unset($_REQUEST['form']);
uncheckTableRows(getRequest('hostid'));
}
}
elseif (isset($_REQUEST['delete']) && isset($_REQUEST['triggerid']))
{
DBstart();
$result = API::Trigger()->delete([getRequest('triggerid')]);
$result = DBend($result);
if ($result) {
unset($_REQUEST['form'], $_REQUEST['triggerid']);
uncheckTableRows(getRequest('hostid'));
}
show_messages($result, _('Trigger deleted'), _('Cannot delete
trigger'));
}
elseif (isset($_REQUEST['add_dependency']) &&
isset($_REQUEST['new_dependency'])) {
if (!isset($_REQUEST['dependencies'])) {
$_REQUEST['dependencies'] = [];
}
foreach ($_REQUEST['new_dependency'] as $triggerid) {

```

```

if (!uint_in_array($triggerid, $_REQUEST['dependencies'])) {
array_push($_REQUEST['dependencies'], $triggerid);
}
}
}
elseif (hasRequest('action') && getRequest('action') ===
'trigger.massupdate'
&& hasRequest('massupdate') && hasRequest('g_triggerid')) {
$result = true;
$visible = getRequest('visible', []);
if ($visible) {
$triggersToUpdate = [];
foreach (getRequest('g_triggerid') as $triggerid) {
$trigger = ['triggerid' => $triggerid];
if (isset($visible['priority'])) {
$trigger['priority'] = getRequest('priority');
}
if (isset($visible['dependencies'])) {
$trigger['dependencies'] = zbx_toObject(getRequest('dependencies',
[]), 'triggerid');
}
$triggersToUpdate[] = $trigger;
}
$result = (bool) API::Trigger()->update($triggersToUpdate);
}
if ($result) {
unset($_REQUEST['form'], $_REQUEST['g_triggerid']);
uncheckTableRows(getRequest('hostid'));
}
show_messages($result, _('Trigger updated'), _('Cannot update
trigger'));
}
elseif (hasRequest('action') && str_in_array(getRequest('action'),
['trigger.massenable',
'trigger.massdisable']) && hasRequest('g_triggerid')) {
$enable = (getRequest('action') == 'trigger.massenable');
$status = $enable ? TRIGGER_STATUS_ENABLED : TRIGGER_STATUS_DISABLED;
$update = [];
// get requested triggers with permission check
$dbTriggers = API::Trigger()->get([
'output' => ['triggerid', 'status'],
'triggerids' => getRequest('g_triggerid'),
'editable' => true
]);
if ($dbTriggers) {
foreach ($dbTriggers as $dbTrigger) {
$update[] = [
'triggerid' => $dbTrigger['triggerid'],
'status' => $status
];
}
}
$result = API::Trigger()->update($update);
}
else {

```



```

$result = true;
}
$update = count($update);
$messageSuccess = $enable
? _n('Trigger enabled', 'Triggers enabled', $updated)
: _n('Trigger disabled', 'Triggers disabled', $updated);
$messageFailed = $enable
? _n('Cannot enable trigger', 'Cannot enable triggers', $updated)
: _n('Cannot disable trigger', 'Cannot disable triggers', $updated);
if ($result) {
  uncheckTableRows(getRequest('hostid'));
  unset($_REQUEST['g_triggerid']);
}
show_messages($result, $messageSuccess, $messageFailed);
}
elseif (hasRequest('action') && getRequest('action') ==
'trigger.masscopyto' && hasRequest('copy') &&
hasRequest('g_triggerid')) {
  if (hasRequest('copy_targetid') && getRequest('copy_targetid') > 0 &&
  hasRequest('copy_type')) {
    // hosts or templates
    if (getRequest('copy_type') == COPY_TYPE_TO_HOST ||
    getRequest('copy_type') ==
    COPY_TYPE_TO_TEMPLATE) {
      $hosts_ids = getRequest('copy_targetid');
    }
    // host groups
    else {
      $hosts_ids = [];
      $group_ids = getRequest('copy_targetid');
      $db_hosts = DBselect(
        'SELECT DISTINCT h.hostid'.
        ' FROM hosts h,hosts_groups hg'.
        ' WHERE h.hostid=hg.hostid'.
        ' AND '.dbConditionInt('hg.groupid', $group_ids)
      );
      while ($db_host = DBfetch($db_hosts)) {
        $hosts_ids[] = $db_host['hostid'];
      }
    }
    DBstart();
    $result = copyTriggersToHosts(getRequest('g_triggerid'), $hosts_ids,
    getRequest('hostid'));
    $result = DBend($result);
    $triggers_count = count(getRequest('g_triggerid'));
    if ($result) {
      uncheckTableRows(getRequest('hostid'));
      unset($_REQUEST['g_triggerid']);
    }
    show_messages($result,
    _n('Trigger copied', 'Triggers copied', $triggers_count),
    _n('Cannot copy trigger', 'Cannot copy triggers', $triggers_count)
  );
}
}

```

```

else {
show_error_message(_('No target selected'));
}
}
elseif (hasRequest('action') && getRequest('action') ==
'trigger.massdelete' && hasRequest('g_triggerid')) {
$result = API::Trigger()->delete(getRequest('g_triggerid'));
if ($result) {
uncheckTableRows(getRequest('hostid'));
}
show_messages($result, _('Triggers deleted'), _('Cannot delete
triggers'));
}
if ((getRequest('action') === 'trigger.massupdateform' ||
hasRequest('massupdate')) &&
hasRequest('g_triggerid')) {
$data = getTriggerMassupdateFormData();
$data['action'] = 'trigger.massupdate';
$triggersView = new CView('configuration.triggers.massupdate',
$data);
$triggersView->render();
$triggersView->show();
}
elseif (isset($_REQUEST['form'])) {
$triggersView = new CView('configuration.triggers.edit',
getTriggerFormData($exprAction));
$triggersView->render();
$triggersView->show();
}
elseif (hasRequest('action') && getRequest('action') ==
'trigger.masscopyto' && hasRequest('g_triggerid')) {
$data = getCopyElementsFormData('g_triggerid', _('CONFIGURATION OF
TRIGGERS'));
$data['action'] = 'trigger.masscopyto';
$triggersView = new CView('configuration.copy.elements', $data);
$triggersView->render();
$triggersView->show();
}
else {
$sortField = getRequest('sort', CProfile::get('web.'.
$page['file'].'.sort', 'description'));
$sortOrder = getRequest('sortorder', CProfile::get('web.'.
$page['file'].'.sortorder', ZBX_SORT_UP));
CProfile::update('web.'. $page['file'].'.sort', $sortField,
PROFILE_TYPE_STR);
CProfile::update('web.'. $page['file'].'.sortorder', $sortOrder,
PROFILE_TYPE_STR);
if (hasRequest('filter_set')) {
CProfile::update('web.triggers.filter_priority',
getRequest('filter_priority', -1),
PROFILE_TYPE_INT);
CProfile::update('web.triggers.filter_state',
getRequest('filter_state', -1),
PROFILE_TYPE_INT);
}
}
}

```

```

CProfile::update('web.triggers.filter_status',
getRequest('filter_status', -1),
PROFILE_TYPE_INT);
}
elseif (hasRequest('filter_rst')) {
CProfile::delete('web.triggers.filter_priority');
CProfile::delete('web.triggers.filter_state');
CProfile::delete('web.triggers.filter_status');
}
$config = select_config();
$data = [
'filter_priority' => CProfile::get('web.triggers.filter_priority', -
1),
'filter_state' => CProfile::get('web.triggers.filter_state', -1),
'filter_status' => CProfile::get('web.triggers.filter_status', -1),
'triggers' => [],
'sort' => $sortField,
'sortorder' => $sortOrder,
'config' => $config
];
$data['pageFilter'] = new CPageFilter([
'groups' => ['with_hosts_and_templates' => true, 'editable' => true],
'hosts' => ['templated_hosts' => true, 'editable' => true],
'triggers' => ['editable' => true],
'groupid' => getRequest('groupid'),
'hostid' => getRequest('hostid')
]);
$data['groupid'] = $data['pageFilter']->groupid;
$data['hostid'] = $data['pageFilter']->hostid;
// get triggers
if ($data['pageFilter']->hostsSelected) {
$options = [
'editable' => true,
'sortfield' => $sortField,
'limit' => $config['search_limit'] + 1
];
if ($sortField === 'status') {
$options['output'] = ['triggerid', 'status', 'state'];
}
else {
$options['output'] = ['triggerid', $sortField];
}
if ($data['filter_priority'] != -1) {
$options['filter']['priority'] = $data['filter_priority'];
}
switch ($data['filter_state']) {
case TRIGGER_STATE_NORMAL:
$options['filter']['state'] = TRIGGER_STATE_NORMAL;
$options['filter']['status'] = TRIGGER_STATUS_ENABLED;
break;
case TRIGGER_STATE_UNKNOWN:
$options['filter']['state'] = TRIGGER_STATE_UNKNOWN;
$options['filter']['status'] = TRIGGER_STATUS_ENABLED;
break;

```

```

default:
if ($data['filter_status'] != -1) {
$options['filter']['status'] = $data['filter_status'];
}
}
if ($data['pageFilter']->hostid > 0) {
$options['hostids'] = $data['pageFilter']->hostid;
}
elseif ($data['pageFilter']->groupid > 0) {
$options['groupids'] = $data['pageFilter']->groupid;
}
}
$data['triggers'] = API::Trigger()->get($options);
}
$_REQUEST['hostid'] = getRequest('hostid', $data['pageFilter']->hostid);
// sort for paging
if ($sortField === 'status') {
orderTriggersByStatus($data['triggers'], $sortOrder);
}
else {
order_result($data['triggers'], $sortField, $sortOrder);
}
// paging
$url = (new CUrl('triggers.php'))
->setArgument('groupid', $data['groupid'])
->setArgument('hostid', $data['hostid']);
$data['paging'] = getPagingLine($data['triggers'], $sortOrder, $url);
$data['triggers'] = API::Trigger()->get([
'output' => ['triggerid', 'expression', 'description', 'status',
'priority', 'error',
'templateid', 'state'],
'selectHosts' => ['hostid', 'host', 'name'],
'selectDependencies' => ['triggerid', 'description'],
'selectDiscoveryRule' => ['itemid', 'name'],
'triggerids' => zbx_objectValues($data['triggers'], 'triggerid')
]);
// sort for displaying full results
if ($sortField === 'status') {
orderTriggersByStatus($data['triggers'], $sortOrder);
}
else {
order_result($data['triggers'], $sortField, $sortOrder);
}
}
$depTriggerIds = [];
foreach ($data['triggers'] as $trigger) {
foreach ($trigger['dependencies'] as $depTrigger) {
$depTriggerIds[$depTrigger['triggerid']] = true;
}
}
}
$dependencyTriggers = [];
if ($depTriggerIds) {
$dependencyTriggers = API::Trigger()->get([
'output' => ['triggerid', 'description', 'status', 'flags'],
'selectHosts' => ['hostid', 'name'],

```

```

'triggerids' => array_keys($depTriggerIds),
'preservekeys' => true
]);
foreach ($data['triggers'] as &$trigger) {
order_result($trigger['dependencies'], 'description', ZBX_SORT_UP);
}
unset($trigger);
foreach ($dependencyTriggers as &$dependencyTrigger) {
order_result($dependencyTrigger['hosts'], 'name', ZBX_SORT_UP);
}
unset($dependencyTrigger);
}
$data['dependencyTriggers'] = $dependencyTriggers;
// get real hosts
$data['realHosts'] = getParentHostsByTriggers($data['triggers']);
// Select writable template IDs.
$hostids = [];
foreach ($data['realHosts'] as $realHost) {
$hostids = array_merge($hostids, zbx_objectValues($realHost,
'hostid'));
}
$data['writable_templates'] = [];
if ($hostids) {
$data['writable_templates'] = API::Template()->get([
'output' => ['templateid'],
'templateids' => $hostids,
'editable' => true,
'preservekeys' => true
]);
}
// do not show 'Info' column, if it is a template
if ($data['hostid']) {
$data['showInfoColumn'] = (bool) API::Host()->get([
'output' => [],
'hostids' => $data['hostid']
]);
}
else {
$data['showInfoColumn'] = true;
}
// render view
$triggersView = new CView('configuration.triggers.list', $data);
$triggersView->render();
$triggersView->show();
}
require_once dirname(__FILE__).'/include/page_footer.php';

```

1.1 users.php

```

<?php
require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/triggers.inc.php';
require_once dirname(__FILE__).'/include/media.inc.php';
require_once dirname(__FILE__).'/include/users.inc.php';

```

```

require_once dirname(__FILE__).'/include/forms.inc.php';
require_once dirname(__FILE__).'/include/js.inc.php';
$page['title'] = _('Configuration of users');
$page['file'] = 'users.php';
require_once dirname(__FILE__).'/include/page_header.php';
$themes = array_keys(Z::getThemes());
$themes[] = THEME_DEFAULT;
// VAR TYPE OPTIONAL FLAGS VALIDATION EXCEPTION
$fields = [
// users
'userid' => [T_ZBX_INT, O_NO, P_SYS, DB_ID,
'isset({form}) && {form} == "update"'],
'group_userid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'filter_usrgrpid' => [T_ZBX_INT, O_OPT, P_SYS, DB_ID, null],
'alias' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY,
'isset({add}) || isset({update})', _('Alias')],
'name' => [T_ZBX_STR, O_OPT, null, null, null,
_x('Name', 'user first name')],
'surname' => [T_ZBX_STR, O_OPT, null, null, null,
_('Surname')],
'password1' => [T_ZBX_STR, O_OPT, null, null,
'(isset({add}) || isset({update})) && isset({form}) && {form} !=
"update" &&
isset({change_password})'],
'password2' => [T_ZBX_STR, O_OPT, null, null,
'(isset({add}) || isset({update})) && isset({form}) && {form} !=
"update" &&
isset({change_password})'],
'user_type' => [T_ZBX_INT, O_OPT, null, IN('1,2,3'), 'isset({add}) ||
isset({update})'],
'user_groups' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY, null],
'user_groups_to_del' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'user_medias' => [T_ZBX_STR, O_OPT, null, NOT_EMPTY, null],
'user_medias_to_del' => [T_ZBX_INT, O_OPT, null, DB_ID, null],
'new_groups' => [T_ZBX_STR, O_OPT, null, null, null],
'new_media' => [T_ZBX_STR, O_OPT, null, null, null],
'enable_media' => [T_ZBX_INT, O_OPT, null, null, null],
'disable_media' => [T_ZBX_INT, O_OPT, null, null, null],
'lang' => [T_ZBX_STR, O_OPT, null, null, null],
'theme' => [T_ZBX_STR, O_OPT, null,
IN(''.implode(", ", $themes).''), 'isset({add}) ||
isset({update})'],
'autologin' => [T_ZBX_INT, O_OPT, null, IN('1'), null],
'autologout' => [T_ZBX_INT, O_OPT, null, BETWEEN(90, 10000), null,
_('Auto-logout
(min 90 seconds)')],
'autologout_visible' => [T_ZBX_STR, O_OPT, null, IN('1'), null],
'url' => [T_ZBX_STR, O_OPT, null, null,
'isset({add}) || isset({update})'],
'refresh' => [T_ZBX_INT, O_OPT, null, BETWEEN(0, SEC_PER_HOUR),
'isset({add}) || isset({update})', _('Refresh (in seconds)')],
'rows_per_page' => [T_ZBX_INT, O_OPT, null, BETWEEN(1,
999999), 'isset({add}) ||
isset({update})', _('Rows per page')],

```

```

// actions
'action' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT,
IN("user.massdelete","user.massunblock"), null],
'register' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, IN("add
permission","delete permission"), null],
'add' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'update' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'delete' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null,
null],
'delete_selected' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'del_user_group' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'del_user_media' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'del_group_user' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'change_password' => [T_ZBX_STR, O_OPT, P_SYS|P_ACT, null, null],
'cancel' => [T_ZBX_STR, O_OPT, P_SYS,
null, null],
35a
// form
'form' => [T_ZBX_STR, O_OPT, P_SYS, null,
null],
'form_refresh' => [T_ZBX_INT, O_OPT, null, null, null],
// sort and sortorder
'sort' => [T_ZBX_STR, O_OPT, P_SYS,
IN("alias","name","surname","type"), null],
'sortorder' => [T_ZBX_STR, O_OPT, P_SYS,
IN("'.ZBX_SORT_DOWN.'","'.ZBX_SORT_UP.'"), null]
];
check_fields($fields);
if (isset($_REQUEST['userid'])) {
$users = API::User()->get([
'userid' => getRequest('userid'),
'output' => API_OUTPUT_EXTEND,
'editable' => true
]);
if (!$users) {
access_deny();
}
}
if (getRequest('filter_usrgrpid') && !API::UserGroup()-
>isWritable([$_REQUEST['filter_usrgrpid']])) {
access_deny();
}
if (hasRequest('action')) {
if (!hasRequest('group_userid') || !
is_array(getRequest('group_userid'))) {
access_deny();
}
else {
$usersChk = API::User()->get([
'output' => ['userid'],
'userid' => getRequest('group_userid'),
'countOutput' => true,
'editable' => true

```

```

]);
if ($usersChk != count(getRequest('group_userid'))) {
access_deny();
}
}
}
$config = select_config();
if (isset($_REQUEST['new_groups'])) {
$_REQUEST['new_groups'] = getRequest('new_groups', []);
$_REQUEST['user_groups'] = getRequest('user_groups', []);
$_REQUEST['user_groups'] += $_REQUEST['new_groups'];
unset($_REQUEST['new_groups']);
}
elseif (isset($_REQUEST['new_media'])) {
$_REQUEST['user_medias'] = getRequest('user_medias', []);
array_push($_REQUEST['user_medias'], $_REQUEST['new_media']);
}
elseif (isset($_REQUEST['user_medias']) &&
isset($_REQUEST['enable_media'])) {
if (isset($_REQUEST['user_medias'][$_REQUEST['enable_media']])) {
$_REQUEST['user_medias'][$_REQUEST['enable_media']]['active'] = 0;
}
}
elseif (isset($_REQUEST['user_medias']) &&
isset($_REQUEST['disable_media'])) {
if (isset($_REQUEST['user_medias'][$_REQUEST['disable_media']])) {
$_REQUEST['user_medias'][$_REQUEST['disable_media']]['active'] = 1;
}
}
elseif (hasRequest('add') || hasRequest('update')) {
$isValid = true;
$usrgrps = getRequest('user_groups', []);
// authentication type
if ($usrgrps) {
$authType = getGroupAuthenticationType($usrgrps,
GROUP_GUI_ACCESS_INTERNAL);
}
else {
$authType = hasRequest('userid')
? getUserAuthenticationType(getRequest('userid'),
GROUP_GUI_ACCESS_INTERNAL)
: $config['authentication_type'];
}
// password validation
if ($authType != ZBX_AUTH_INTERNAL) {
if (hasRequest('password1')) {
show_error_message(_s('Password is unavailable for users with %1$s.',
authentication2str($authType)));
$isValid = false;
}
else {
if (hasRequest('userid')) {
$_REQUEST['password1'] = null;
$_REQUEST['password2'] = null;
}
}
}
}

```



```

36a
}
else {
$_REQUEST['password1'] = 'zabbix';
$_REQUEST['password2'] = 'zabbix';
}
}
}
else {
$_REQUEST['password1'] = getRequest('password1');
$_REQUEST['password2'] = getRequest('password2');
}
if ($_REQUEST['password1'] != $_REQUEST['password2']) {
if (isset($_REQUEST['userid'])) {
show_error_message(_('Cannot update user. Both passwords must be
equal.'));
}
else {
show_error_message(_('Cannot add user. Both passwords must be
equal.'));
}
}
$isValid = false;
}
elseif (isset($_REQUEST['password1']) && $_REQUEST['alias'] ==
ZBX_GUEST_USER &&
!zbx_empty($_REQUEST['password1'])) {
show_error_message(_('For guest, password must be empty'));
$isValid = false;
}
elseif (isset($_REQUEST['password1']) && $_REQUEST['alias'] !=
ZBX_GUEST_USER &&
zbx_empty($_REQUEST['password1'])) {
show_error_message(_('Password should not be empty'));
$isValid = false;
}
}
if ($isValid) {
$user = [];
$user['alias'] = getRequest('alias');
$user['name'] = getRequest('name');
$user['surname'] = getRequest('surname');
$user['passwd'] = getRequest('password1');
$user['url'] = getRequest('url');
$user['autologin'] = getRequest('autologin', 0);
$user['autologout'] = hasRequest('autologout_visible') ?
getRequest('autologout') : 0;
$user['theme'] = getRequest('theme');
$user['refresh'] = getRequest('refresh');
$user['rows_per_page'] = getRequest('rows_per_page');
$user['type'] = getRequest('user_type');
$user['user_medias'] = getRequest('user_medias', []);
$user['usrgrps'] = zbx_toObject($usrgrps, 'usrgrpId');
if (hasRequest('lang')) {
$user['lang'] = getRequest('lang');
}
}

```

```

DBstart();
if (hasRequest('userid')) {
$user['userid'] = getRequest('userid');
$result = API::User()->update([$user]);
if ($result) {
$result = API::User()->updateMedia([
'users' => $user,
'medias' => $user['user_medias']
]);
}
$messageSuccess = _('User updated');
$messageFailed = _('Cannot update user');
$auditAction = AUDIT_ACTION_UPDATE;
}
else {
$result = API::User()->create($user);
$messageSuccess = _('User added');
$messageFailed = _('Cannot add user');
$auditAction = AUDIT_ACTION_ADD;
}
if ($result) {
add_audit($auditAction, AUDIT_RESOURCE_USER,
'User alias ['. $_REQUEST['alias'].'] name ['. $_REQUEST['name'].']
surname ['. $_REQUEST['surname'].']'
);
unset($_REQUEST['form']);
}
$result = DBend($result);
if ($result) {
uncheckTableRows();
}
show_messages($result, $messageSuccess, $messageFailed);
37a
}
}
elseif (isset($_REQUEST['del_user_media'])) {
foreach (getRequest('user_medias_to_del', []) as $mediaId) {
if (isset($_REQUEST['user_medias'][$mediaId])) {
unset($_REQUEST['user_medias'][$mediaId]);
}
}
}
elseif (isset($_REQUEST['del_user_group'])) {
foreach (getRequest('user_groups_to_del', []) as $groupId) {
if (isset($_REQUEST['user_groups'][$groupId])) {
unset($_REQUEST['user_groups'][$groupId]);
}
}
}
elseif (isset($_REQUEST['delete']) && isset($_REQUEST['userid'])) {
$user = reset($users);
$result = API::User()->delete([$user['userid']]);
unset($_REQUEST['userid'], $_REQUEST['form']);
if ($result) {

```

```

uncheckTableRows();
}
show_messages($result, _('User deleted'), _('Cannot delete user'));
}
elseif (hasRequest('action') && getRequest('action') ==
'user.massunblock' && hasRequest('group_userid')) {
$groupUserId = getRequest('group_userid');
DBstart();
$result = unblock_user_login($groupUserId);
if ($result) {
$users = API::User()->get([
'userids' => $groupUserId,
'output' => API_OUTPUT_EXTEND
]);
foreach ($users as $user) {
info('User '.$user['alias'].' unblocked');
add_audit(AUDIT_ACTION_UPDATE, AUDIT_RESOURCE_USER, 'Unblocked user
alias
['.$user['alias'].'] name ['.$user['name'].'] surname ['.
$user['surname'].']);
}
}
$result = DBend($result);
if ($result) {
uncheckTableRows();
}
show_messages($result, _('Users unblocked'), _('Cannot unblock
users'));
}
elseif (hasRequest('action') && getRequest('action') ==
'user.massdelete' && hasRequest('group_userid')) {
$result = API::User()->delete(getRequest('group_userid'));
if ($result) {
uncheckTableRows();
}
show_messages($result, _('User deleted'), _('Cannot delete user'));
}
$_REQUEST['filter_usrgrpid'] = getRequest('filter_usrgrpid',
CProfile::get('web.users.filter_usrgrpid', 0));
CProfile::update('web.users.filter_usrgrpid',
$_REQUEST['filter_usrgrpid'], PROFILE_TYPE_ID);
if (!empty($_REQUEST['form'])) {
$userId = getRequest('userid', 0);
$data = getUserFormData($userId, $config);
$data['userid'] = $userId;
$data['form'] = getRequest('form');
$data['form_refresh'] = getRequest('form_refresh', 0);
$data['autologout'] = getRequest('autologout');
// render view
$usersView = new CView('administration.users.edit', $data);
$usersView->render();
$usersView->show();
}
else {

```

```

$sortField = getRequest('sort', CProfile::get('web.'.
$page['file'].'.sort', 'alias'));
$sortOrder = getRequest('sortorder', CProfile::get('web.'.
$page['file'].'.sortorder', ZBX_SORT_UP));
CProfile::update('web.'. $page['file'].'.sort', $sortField,
PROFILE_TYPE_STR);
CProfile::update('web.'. $page['file'].'.sortorder', $sortOrder,
PROFILE_TYPE_STR);
$data = [
'config' => $config,
'sort' => $sortField,
'sortorder' => $sortOrder
];
// get user groups
38a
$data['userGroups'] = API::UserGroup()->get([
'output' => API_OUTPUT_EXTEND
]);
order_result($data['userGroups'], 'name');
// get users
$data['users'] = API::User()->get([
'usrgrpids' => ($REQUEST['filter_usrgrpids'] > 0) ?
$REQUEST['filter_usrgrpids'] : null,
'output' => API_OUTPUT_EXTEND,
'selectUsrgrps' => API_OUTPUT_EXTEND,
'getAccess' => 1,
'limit' => $config['search_limit'] + 1
]);
// sorting & paging
order_result($data['users'], $sortField, $sortOrder);
$data['paging'] = getPagingLine($data['users'], $sortOrder, new
CUrl('users.php'));
// set default lastaccess time to 0
foreach ($data['users'] as $user) {
$data['usersSessions'][$user['userid']] = ['lastaccess' => 0];
}
$dbSessions = DBselect(
'SELECT s.userid,MAX(s.lastaccess) AS lastaccess,s.status'.
' FROM sessions s'.
' WHERE '.dbConditionInt('s.userid', zbx_objectValues($data['users'],
'userid')).
' GROUP BY s.userid,s.status'
);
while ($session = DBfetch($dbSessions)) {
if ($data['usersSessions'][$session['userid']]['lastaccess'] <
$session['lastaccess']) {
$data['usersSessions'][$session['userid']] = $session;
}
}
// render view
$usersView = new CView('administration.users.list', $data);
$usersView->render();
$usersView->show();
}

```

```
require_once dirname(__FILE__).'/include/page_footer.php';
```

1.J config.inc.php

```
<?php
setlocale(LC_CTYPE, [
'C', 'POSIX', 'en', 'en_US', 'en_US.UTF-8', 'English_United
States.1252', 'en_GB', 'en_GB.UTF-8'
]);
require_once dirname(__FILE__).'/classes/core/Z.php';
try {
Z::getInstance()->run(ZBase::EXEC_MODE_DEFAULT);
}
catch (DBException $e) {
(new CView('general.warning', [
'header' => 'Database error',
'messages' => [$e->getMessage()],
'theme' => ZBX_DEFAULT_THEME
]))->render();
exit;
}
catch (ConfigFileException $e) {
switch ($e->getCode()) {
case CConfigFile::CONFIG_NOT_FOUND:
redirect('setup.php');
exit;
case CConfigFile::CONFIG_ERROR:
(new CView('general.warning', [
'header' => 'Configuration file error',
'messages' => [$e->getMessage()],
'theme' => ZBX_DEFAULT_THEME
]))->render();
exit;
}
}
catch (Exception $e) {
(new CView('general.warning', [
'header' => $e->getMessage(),
'messages' => [],
'theme' => ZBX_DEFAULT_THEME
]))->render();
exit;
}
CProfiler::getInstance()->start();
global $ZBX_SERVER, $ZBX_SERVER_PORT, $page;
39a
$page = [
'title' => null,
'file' => null,
'scripts' => null,
'type' => null,
'menu' => null
];
```

1.K zabbix.conf.php

```
<?php
// Zabbix GUI configuration file.
global $DB;
$DB['TYPE'] = 'MYSQL';
$DB['SERVER'] = 'localhost';
$DB['PORT'] = '3306';
$DB['DATABASE'] = 'nms';
$DB['USER'] = 'nms';
$DB['PASSWORD'] = 'nms12qwaszx';
// Schema name. Used for IBM DB2 and PostgreSQL.
$DB['SCHEMA'] = '';
$ZBX_SERVER = 'nms';
$ZBX_SERVER_PORT = '10051';
$ZBX_SERVER_NAME = 'mulyadi';
$IMAGE_FORMAT_DEFAULT = IMAGE_FORMAT_PNG;
```

1.L db.inc.php

```
<?php
function DBconnect(&$error) {
global $DB;
if (isset($DB['DB'])) {
$error = _('Cannot create another database connection.');
```

```
return false;
}
$result = true;
$DB['DB'] = null; // global db handler
$DB['TRANSACTIONS'] = 0; // level of a nested transaction
$DB['TRANSACTION_NO_FAILED_SQLS'] = true; // true - if no statements
failed in transaction, false -
there are failed statements
$DB['SELECT_COUNT'] = 0; // stats
$DB['EXECUTE_COUNT'] = 0;
if (!isset($DB['TYPE'])) {
$error = 'Unknown database type.';
$result = false;
}
else {
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$DB['DB'] = @mysqli_connect($DB['SERVER'], $DB['USER'],
$DB['PASSWORD'], $DB['DATABASE'], $DB['PORT']);
if (!$DB['DB']) {
$error = 'Error connecting to database:
'.trim(mysqli_connect_error());
$result = false;
}
elseif (mysqli_autocommit($DB['DB'], true) === false) {
$error = 'Error setting auto commit.';
$result = false;
}
}
```

```

else {
DBexecute('SET NAMES utf8');
}
if ($result) {
$dbBackend = new MysqlDbBackend();
}
break;
case ZBX_DB_POSTGRESQL:
$pg_connection_string =
(!empty($DB['SERVER']) ?
'host=\'' . pg_connect_escape($DB['SERVER']) . '\ ' : '').
'dbname=\'' . pg_connect_escape($DB['DATABASE']) . '\ ' .
(!empty($DB['USER']) ?
'user=\'' . pg_connect_escape($DB['USER']) . '\ ' : '').
(!empty($DB['PASSWORD']) ?
'password=\'' . pg_connect_escape($DB['PASSWORD']) . '\ ' : '').
(!empty($DB['PORT']) ?
'port=' . pg_connect_escape($DB['PORT']) . ' : ');
$DB['DB'] = @pg_connect($pg_connection_string);
if (!$DB['DB']) {
$error = 'Error connecting to database.';
$result = false;
}
else {
$schemaSet = DBexecute('SET search_path =
.zbx_dbstr($DB['SCHEMA'] ? $DB['SCHEMA'] : 'public'), true);
if (!$schemaSet) {
clear_messages();
$error = pg_last_error();
$result = false;
}
else {
if (false !== ($pgsql_version =
pg_parameter_status('server_version'))) {
if ((int) $pgsql_version >= 9) {
// change the output format
for values of type bytea from hex (the default) to escape
DBexecute('SET bytea_output =
escape');
}
}
}
}
if ($result) {
$dbBackend = new PostgresqlDbBackend();
}
break;
case ZBX_DB_ORACLE:
$connect = '';
if (!empty($DB['SERVER'])) {
$connect = '//'. $DB['SERVER'];
if ($DB['PORT'] != '0') {
$connect .= ':' . $DB['PORT'];
}
}

```

```

if ($DB['DATABASE']) {
$connect .= '/' . $DB['DATABASE'];
}
}
$DB['DB'] = @oci_connect($DB['USER'], $DB['PASSWORD'], $connect);
if ($DB['DB']) {
DBexecute('ALTER SESSION SET
NLS_NUMERIC_CHARACTERS='.zbx_dbstr('. '));
}
else {
SociError = oci_error();
$error = 'Error connecting to database:
' . $SociError['message'];
$result = false;
}
if ($result) {
$dbBackend = new OracleDbBackend();
}
break;
case ZBX_DB_DB2:
$connect = '';
$connect .= 'DATABASE=' . $DB['DATABASE'] . ';';
$connect .= 'HOSTNAME=' . $DB['SERVER'] . ';';
$connect .= 'PORT=' . $DB['PORT'] . ';';
$connect .= 'PROTOCOL=TCPIP;';
$connect .= 'UID=' . $DB['USER'] . ';';
$connect .= 'PWD=' . $DB['PASSWORD'] . ';';
$DB['DB'] = @db2_connect($connect, $DB['USER'], $DB['PASSWORD']);
if (!$DB['DB']) {
$error = 'Error connecting to database:
'.db2_conn_errormsg();
$result = false;
}
else {
$options = [
'db2_attr_case' => DB2_CASE_LOWER,
];
db2_set_option($DB['DB'], $options, 1);
if (isset($DB['SCHEMA']) && $DB['SCHEMA'] != '') {
DBexecute('SET CURRENT
SCHEMA=' . zbx_dbstr($DB['SCHEMA']));
}
}
if ($result) {
$dbBackend = new Db2DbBackend();
}
break;
case ZBX_DB_SQLITE3:
if (file_exists($DB['DATABASE'])) {
init_sqlite3_access();
lock_sqlite3_access();
try{
$DB['DB'] = @new SQLite3($DB['DATABASE'],
SQLITE3_OPEN_READWRITE);
}
}
}

```



```

}
catch (Exception $e) {
$error = 'Error connecting to database.';
$result = false;
}
unlock_sqlite3_access();
}
else {
$error = 'Missing database';
$result = false;
}
if ($result) {
$dbBackend = new SqliteDbBackend();
}
break;
default:
$error = 'Unsupported database';
$result = false;
}
}
if ($result && (!$dbBackend->checkDbVersion() || !$dbBackend->checkConfig())) {
$error = $dbBackend->getError();
$result = false;
}
if (false == $result) {
$DB['DB'] = null;
}
return $result;
}
function DBclose() {
global $DB;
$result = false;
if (isset($DB['DB']) && !empty($DB['DB'])) {
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$result = mysqli_close($DB['DB']);
break;
case ZBX_DB_POSTGRESQL:
$result = pg_close($DB['DB']);
break;
case ZBX_DB_ORACLE:
$result = oci_close($DB['DB']);
break;
case ZBX_DB_DB2:
$result = db2_close($DB['DB']);
break;
case ZBX_DB_SQLITE3:
lock_sqlite3_access();
$dbBackend->close();
unlock_sqlite3_access();
$result = true;
break;
}
}

```

```

}
unset($DB['DB']);
return $result;
}
function DBstart() {
global $DB;
$result = false;
if ($DB['TRANSACTIONS'] != 0) {
info('POSSIBLE ERROR: Used incorrect logic in database processing,
started
subtransaction!');
return $result;
}
$DB['TRANSACTIONS']++;
$DB['TRANSACTION_NO_FAILED_SQLS'] = true;
if (!isset($DB['DB']) || empty($DB['DB'])) {
return $result;
}
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$result = DBexecute('BEGIN');
break;
case ZBX_DB_POSTGRESQL:
$result = DBexecute('BEGIN');
break;
case ZBX_DB_ORACLE:
$result = true;
break;
case ZBX_DB_DB2:
$result = db2_autocommit($DB['DB'], DB2_AUTOCOMMIT_OFF);
break;
case ZBX_DB_SQLITE3:
lock_sqlite3_access();
$result = DBexecute('BEGIN');
break;
}
return $result;
}
function DBend($doCommit = true) {
global $DB;
$result = false;
if (!isset($DB['DB']) || empty($DB['DB'])) {
return $result;
}
if ($DB['TRANSACTIONS'] == 0) {
info('POSSIBLE ERROR: Used incorrect logic in database processing,
transaction not
started!');
return $result;
}
$DBresult = $doCommit && $DB['TRANSACTION_NO_FAILED_SQLS'];
if ($DBresult) {
$DBresult = DBcommit();
}
}

```

```

else {
DBrollback();
}
$DB['TRANSACTIONS'] = 0;
return (!is_null($doCommit) && $DBresult) ? $doCommit : $DBresult;
}
function DBcommit() {
global $DB;
$result = false;
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$result = DBexecute('COMMIT');
break;
case ZBX_DB_POSTGRESQL:
$result = DBexecute('COMMIT');
break;
case ZBX_DB_ORACLE:
$result = oci_commit($DB['DB']);
break;
case ZBX_DB_DB2:
$result = db2_commit($DB['DB']);
if ($result) {
db2_autocommit($DB['DB'], DB2_AUTOCOMMIT_ON);
}
break;
case ZBX_DB_SQLITE3:
$result = DBexecute('COMMIT');
unlock_sqlite3_access();
break;
}
return $result;
}
function DBrollback() {
global $DB;
$result = false;
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$result = DBexecute('ROLLBACK');
break;
case ZBX_DB_POSTGRESQL:
$result = DBexecute('ROLLBACK');
break;
case ZBX_DB_ORACLE:
$result = oci_rollback($DB['DB']);
break;
case ZBX_DB_DB2:
$result = db2_rollback($DB['DB']);
db2_autocommit($DB['DB'], DB2_AUTOCOMMIT_ON);
break;
case ZBX_DB_SQLITE3:
$result = DBexecute('ROLLBACK');
unlock_sqlite3_access();
break;
}
}

```

```

return $result;
}
function DBselect($query, $limit = null, $offset = 0) {
global $DB;
$result = false;
if (!isset($DB['DB']) || empty($DB['DB'])) {
return $result;
}
// add the LIMIT clause
if (!$query = DBaddLimit($query, $limit, $offset)) {
return false;
}
$time_start = microtime(true);
$DB['SELECT_COUNT']++;
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
if (!$result = mysqli_query($DB['DB'], $query)) {
error('Error in query ['. $query. '] ['. mysqli_error($DB['DB']). ']);
}
break;
case ZBX_DB_POSTGRESQL:
if (!$result = pg_query($DB['DB'], $query)) {
error('Error in query ['. $query. '] ['. pg_last_error(). ']);
}
break;
case ZBX_DB_ORACLE:
if (!$result = oci_parse($DB['DB'], $query)) {
$e = @oci_error();
error('SQL error ['. $e['message']. '] in ['. $e['sqltext']. ']);
}
elseif (!@oci_execute($result, ($DB['TRANSACTIONS'] ? OCI_DEFAULT :
OCI_COMMIT_ON_SUCCESS))) {
$e = @oci_error($result);
error('SQL error ['. $e['message']. '] in ['. $e['sqltext']. ']);
}
break;
case ZBX_DB_DB2:
$options = [];
if ($DB['TRANSACTIONS']) {
$options['autocommit'] = DB2_AUTOCOMMIT_OFF;
}
if (!$result = db2_prepare($DB['DB'], $query)) {
$e = @db2_stmt_errormsg($result);
error('SQL error ['. $query. '] in ['. $e. ']);
}
elseif (true !== @db2_execute($result, $options)) {
$e = @db2_stmt_errormsg($result);
error('SQL error ['. $query. '] in ['. $e. ']);
$result = false;
}
break;
case ZBX_DB_SQLITE3:
if ($DB['TRANSACTIONS'] == 0) {
lock_sqlite3_access();
}
}
}

```

```

}
if (false === ($result = $DB['DB']->query($query))) {
error('Error in query ['. $query. '] Error code ['. $DB['DB']->
lastErrorCode(). '] Message ['. $DB['DB']->lastErrorMsg(). ']');
}
if ($DB['TRANSACTIONS'] == 0) {
unlock_sqlite3_access();
}
break;
}
// $result is false only if an error occurred
if ($DB['TRANSACTION_NO_FAILED_SQLS'] && !$result) {
$DB['TRANSACTION_NO_FAILED_SQLS'] = false;
}
CProfiler::getInstance()->profileSql(microtime(true) - $time_start,
$query);
return $result;
}
function DBaddLimit($query, $limit = 0, $offset = 0) {
global $DB;
if ((isset($limit) && ($limit < 0 || !zbx_ctype_digit($limit))) ||
$offset < 0 ||
!zbx_ctype_digit($offset)) {
$moreDetails = isset($limit) ? ' Limit ['. $limit. '] Offset ['.
$offset. ']' : ' Offset
['. $offset. ']';
error('Incorrect parameters for limit and/or offset. Query ['.
$query. ']' . $moreDetails);
return false;
}
// Process limit and offset
if (isset($limit)) {
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
case ZBX_DB_POSTGRES SQL:
case ZBX_DB_SQLITE3:
$query .= ' LIMIT '.intval($limit).' OFFSET '.intval($offset);
break;
case ZBX_DB_ORACLE:
case ZBX_DB_DB2:
$still = $offset + $limit;
$query = 'SELECT * FROM ('. $query. ') WHERE rownum BETWEEN
'.intval($offset).' AND '.intval($still);
break;
}
}
return $query;
}
function DBexecute($query, $skip_error_messages = 0) {
global $DB;
if (!isset($DB['DB']) || empty($DB['DB'])) {
return false;
}
$result = false;

```

```

$time_start = microtime(true);
$DB['EXECUTE_COUNT']++;
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
if (!$result = mysqli_query($DB['DB'], $query)) {
error('Error in query ['. $query.'] ['.mysqli_error($DB['DB']).']');
}
break;
case ZBX_DB_POSTGRESQL:
if (!$result = (bool) pg_query($DB['DB'], $query)) {
error('Error in query ['. $query.'] ['.pg_last_error().']');
}
break;
case ZBX_DB_ORACLE:
if (!$result = oci_parse($DB['DB'], $query)) {
$e = @oci_error();
error('SQL error ['. $e['message'].'] in ['. $e['sqltext'].']');
}
elseif (@oci_execute($result, ($DB['TRANSACTIONS'] ? OCI_DEFAULT :
OCI_COMMIT_ON_SUCCESS))) {
$e = oci_error($result);
error('SQL error ['. $e['message'].'] in ['. $e['sqltext'].']');
}
else {
$result = true; // function must return boolean
}
break;
case ZBX_DB_DB2:
if (!$result = db2_prepare($DB['DB'], $query)) {
$e = @db2_stmt_errormsg($result);
error('SQL error ['. $query.'] in ['. $e.']);
}
elseif (true !== @db2_execute($result)) {
$e = @db2_stmt_errormsg($result);
error('SQL error ['. $query.'] in ['. $e.']);
}
else {
$result = true; // function must return boolean
}
break;
case ZBX_DB_SQLITE3:
if ($DB['TRANSACTIONS'] == 0) {
lock_sqlite3_access();
}
if (!$result = $DB['DB']->exec($query)) {
error('Error in query ['. $query.'] Error code ['. $DB['DB']->lastErrorCode().'] Message ['. $DB['DB']->lastErrorMsg().']');
}
if ($DB['TRANSACTIONS'] == 0) {
unlock_sqlite3_access();
}
break;
}
if ($DB['TRANSACTIONS'] != 0 && !$result) {

```

```

$DB['TRANSACTION_NO_FAILED_SQLS'] = false;
}
CProfiler::getInstance()->profileSql(microtime(true) - $time_start,
$query);
return (bool) $result;
}
function DBfetch($cursor, $convertNulls = true) {
global $DB;
$result = false;
if (!isset($DB['DB']) || empty($DB['DB']) || is_bool($cursor)) {
return $result;
}
switch ($DB['TYPE']) {
case ZBX_DB_MYSQL:
$result = mysqli_fetch_assoc($cursor);
if (!$result) {
mysqli_free_result($cursor);
}
break;
case ZBX_DB_POSTGRESQL:
if (!$result = pg_fetch_assoc($cursor)) {
pg_free_result($cursor);
}
break;
case ZBX_DB_ORACLE:
if ($row = oci_fetch_assoc($cursor)) {
$result = [];
foreach ($row as $key => $value) {
$field_type = strtolower(oci_field_type($cursor, $key));
// Oracle does not support NULL values for string fields, so
if the string is empty, it will return NULL
// convert it to an empty string to be consistent with other
databases
$value = (str_in_array($field_type, ['varchar', 'varchar2',
'blob', 'clob']) && is_null($value)) ? '' : $value;
if (is_object($value) && (strpos($field_type, 'lob') !==
false)) {
$value = $value->load();
}
$result[strtolower($key)] = $value;
}
}
break;
case ZBX_DB_DB2:
if (!$result = db2_fetch_assoc($cursor)) {
db2_free_result($cursor);
}
else {
// cast all of the values to string to be consistent with other DB
drivers: all of them return
// only strings.
foreach ($result as &$value) {
if ($value !== null) {
$value = (string) $value;
}
}
}
}
}

```

```

}
}
unset($value);
}
break;
case ZBX_DB_SQLITE3:
if ($DB['TRANSACTIONS'] == 0) {
lock_sqlite3_access();
}
if (!$result = $cursor->fetchArray(SQLITE3_ASSOC)) {
unset($cursor);
}
else {
// cast all of the values to string to be consistent with other DB
drivers: all of them return
// only strings.
foreach ($result as &$value) {
$value = (string) $value;
}
unset($value);
}
if ($DB['TRANSACTIONS'] == 0) {
unlock_sqlite3_access();
}
break;
}
if ($result) {
if ($convertNulls) {
foreach ($result as $key => $val) {
if (is_null($val)) {
$result[$key] = '0';
}
}
}
return $result;
}
return false;
}
function zbx_sql_mod($x, $y) {
global $DB;
switch ($DB['TYPE']) {
case ZBX_DB_SQLITE3:
return ' (('.$x.') % ( '.$y.' ))';
default:
46a
return ' MOD('.$x.', '.$y.' )';
}
}
function get_dbid($table, $field) {
// PGSQL on transaction failure on all queries returns false..
global $DB;
if ($DB['TYPE'] == ZBX_DB_POSTGRESQL && $DB['TRANSACTIONS'] && !
$DB['TRANSACTION_NO_FAILED_SQLS']) {
return 0;
}
}

```



```

}
$found = false;
$min = 0;
$max = ZBX_DB_MAX_ID;
do {
$dbSelect = DBselect('SELECT i.nextid FROM ids i WHERE
i.table_name='.zbx_dbstr($table).'
AND i.field_name='.zbx_dbstr($field));
if (!$dbSelect) {
return false;
}
$row = DBfetch($dbSelect);
if (!$row) {
$row = DBfetch(DBselect('SELECT MAX('.$field.') AS id FROM '.$table.'
WHERE
'.$field.' BETWEEN '.$min.' AND '.$max));
if (!$row || ($row['id'] == 0)) {
DBexecute("INSERT INTO ids (table_name,field_name,nextid) VALUES
('$table','$field',$min)");
}
else {
DBexecute("INSERT INTO ids (table_name,field_name,nextid) VALUES
('$table','$field','.$row['id'].')");
}
continue;
}
else {
$ret1 = $row['nextid'];
if (bccomp($ret1, $min) < 0 || !bccomp($ret1, $max) < 0) {
DBexecute('DELETE FROM ids WHERE table_name='.zbx_dbstr($table).' AND
field_name='.zbx_dbstr($field));
continue;
}
$sql = 'UPDATE ids SET nextid=nextid+1 WHERE
table_name='.zbx_dbstr($table).'
AND field_name='.zbx_dbstr($field);
DBexecute($sql);
$row = DBfetch(DBselect('SELECT i.nextid FROM ids i WHERE
i.table_name='.zbx_dbstr($table).' AND
i.field_name='.zbx_dbstr($field)));
if (!$row || is_null($row['nextid'])) {
// should never be here
continue;
}
else {
$ret2 = $row['nextid'];
if (bccomp(bcadd($ret1, 1, 0), $ret2, 0) == 0) {
$found = true;
}
}
}
}
while (false == $found);
return $ret2;

```

```

}
function zbx_db_distinct($sql_parts) {
if (count($sql_parts['from']) > 1) {
return ' DISTINCT ';
}
else {
return ' ';
}
}
function zbx_db_search($table, $options, &$sql_parts) {
list($table, $tableShort) = explode(' ', $table);
$tableSchema = DB::getSchema($table);
if (!$tableSchema) {
info(_s('Error in search request for table "%1$s".', $table));
}
$start = is_null($options['startSearch']) ? '%' : '';
$exclude = is_null($options['excludeSearch']) ? '' : ' NOT ';
$glue = (!$options['searchByAny']) ? ' AND ' : ' OR ';
$search = [];
foreach ($options['search'] as $field => $patterns) {
if (!isset($tableSchema['fields'][$field]) || zbx_empty($patterns)) {
continue;
}
if ($tableSchema['fields'][$field]['type'] != DB::FIELD_TYPE_CHAR
&& $tableSchema['fields'][$field]['type'] != DB::FIELD_TYPE_TEXT) {
continue;
}
$fieldSearch = [];
foreach ((array) $patterns as $pattern) {
if (zbx_empty($pattern)) {
continue;
}
// escaping parameter that is about to be used in LIKE statement
$pattern = str_replace("!", "!!", $pattern);
$pattern = str_replace("%", "!%", $pattern);
$pattern = str_replace("_", "!_", $pattern);
if (!$options['searchWildcardsEnabled']) {
$fieldSearch[] =
' UPPER('.$tableShort.'.'.$field.') '.
$exclude.' LIKE '.
zbx_dbstr($start.mb_strtoupper($pattern).'%').
" ESCAPE '!";
}
else {
$pattern = str_replace("*", "%", $pattern);
$fieldSearch[] =
' UPPER('.$tableShort.'.'.$field.') '.
$exclude.' LIKE '.
zbx_dbstr(mb_strtoupper($pattern)).
" ESCAPE '!";
}
}
$search[$field] = '(' . implode($glue, $fieldSearch) . ')';
}
}

```

```

if (!empty($search)) {
if (isset($sql_parts['where']['search'])) {
$search[] = $sql_parts['where']['search'];
}
$sql_parts['where']['search'] = '(' . implode($glue, $search) . ')';
return true;
}
return false;
}
function check_db_fields($dbFields, &$args) {
if (!is_array($args)) {
return false;
}
foreach ($dbFields as $field => $def) {
if (!isset($args[$field])) {
if (is_null($def)) {
return false;
}
else {
$args[$field] = $def;
}
}
}
return true;
}
function dbConditionInt($fieldName, array $values, $notIn = false,
$sort = true) {
$MAX_EXPRESSIONS = 950; // maximum number of values for using "IN
(id1>,<id2>,...,<idN>)"
$MIN_NUM_BETWEEN = 4; // minimum number of consecutive values for
using "BETWEEN <id1> AND <idN>"
if (is_bool(reset($values))) {
return '1=0';
}
$values = array_keys(array_flip($values));
if ($sort) {
natsort($values);
$values = array_values($values);
}
$betweens = [];
$data = [];
for ($i = 0, $size = count($values); $i < $size; $i++) {
$between = [];
// analyze by chunk
if (isset($values[$i + $MIN_NUM_BETWEEN])
&& bccomp(bcadd($values[$i], $MIN_NUM_BETWEEN), $values[$i +
$MIN_NUM_BETWEEN]) == 0) {
for ($sizeMinBetween = $i + $MIN_NUM_BETWEEN; $i < $sizeMinBetween;
$i++) {
$between[] = $values[$i];
}
$i--; // shift 1 back
// analyze by one
for (; $i < $size; $i++) {

```

```

if (isset($values[$i + 1]) && bccomp(bcadd($values[$i], 1),
$values[$i
+ 1]) == 0) {
$between[] = $values[$i + 1];
}
else {
break;
}
}
$betweens[] = $between;
}
else {
$data[] = $values[$i];
}
}
// concatenate conditions
$dataSize = count($data);
$betweenSize = count($betweens);
$condition = '';
$operatorAnd = $notIn ? ' AND ' : ' OR ';
if ($betweens) {
$operatorNot = $notIn ? 'NOT ' : '';
foreach ($betweens as $between) {
$between = $operatorNot.$fieldName.' BETWEEN
'.zbx_dbstr($between[0]).' AND
'.zbx_dbstr(end($between));
$condition .= $condition ? $operatorAnd.$between : $between;
}
}
if ($dataSize == 1) {
$operator = $notIn ? '!=' : '=';
$condition .= ($condition ? $operatorAnd : '').$fieldName.
$operator.zbx_dbstr($data[0]);
}
else {
$operatorNot = $notIn ? ' NOT' : '';
$data = array_chunk($data, $MAX_EXPRESSIONS);
foreach ($data as $chunk) {
$chunkIns = '';
foreach ($chunk as $value) {
$chunkIns .= ','.zbx_dbstr($value);
}
$chunkIns = $fieldName.$operatorNot.' IN ('.substr($chunkIns, 1).)';
$condition .= $condition ? $operatorAnd.$chunkIns : $chunkIns;
}
}
return (($dataSize && $betweenSize) || $betweenSize > 1 || $dataSize
> $MAX_EXPRESSIONS) ?
>('.$condition.'): $condition;
}
function dbConditionString($fieldName, array $values, $notIn = false)
{
switch (count($values)) {
case 0:

```

```

return '1=0';
case 1:
return $notIn
? $fieldName.'!='.zbx_dbstr(reset($values))
: $fieldName.'='.zbx_dbstr(reset($values));
}
$in = $notIn ? ' NOT IN ' : ' IN ';
$concat = $notIn ? ' AND ' : ' OR ';
$items = array_chunk($values, 950);
$condition = '';
foreach ($items as $values) {
$condition .= !empty($condition) ? ')'. $concat. $fieldName. $in. '(' :
'';
$condition .= implode(',', zbx_dbstr($values));
}
return '('.$fieldName.$in.'('$condition.))';
}
function DBfetchArray($cursor) {
$result = [];
while ($row = DBfetch($cursor)) {
$result[] = $row;
}
return $result;
}
function DBfetchArrayAssoc($cursor, $field) {
$result = [];
while ($row = DBfetch($cursor)) {
$result[$row[$field]] = $row;
}
return $result;
}
function DBfetchColumn($cursor, $column, $asHash = false) {
$result = [];
while ($dbResult = DBfetch($cursor)) {
if ($asHash) {
$result[$dbResult[$column]] = $dbResult[$column];
}
else {
$result[] = $dbResult[$column];
}
}
return $result;
}
function init_sqlite3_access() {
global $DB;
$DB['SEM_ID'] = sem_get(ftok($DB['DATABASE'], 'z'), 1, 0660);
}
function lock_sqlite3_access() {
global $DB;
sem_acquire($DB['SEM_ID']);
}
function unlock_sqlite3_access() {
global $DB;
sem_release($DB['SEM_ID']);
}

```

```

}
function idcmp($id1, $id2) {
return (string) $id1 === (string) $id2;
}
function pg_connect_escape($string) {
return addslashes($string, "\\");
}
function zbx_dbstr($var) {
global $DB;
if (!isset($DB['TYPE'])) {
return false;
}
switch ($DB['TYPE']) {
case ZBX_DB_DB2:
if (is_array($var)) {
foreach ($var as $vnum => $value) {
$var[$vnum] = "" . db2_escape_string($value) . "";
}
return $var;
}
return "" . db2_escape_string($var) . "";
case ZBX_DB_MYSQL:
if (is_array($var)) {
foreach ($var as $vnum => $value) {
$var[$vnum] = "" . mysqli_real_escape_string($DB['DB'],
$value) . "";
}
return $var;
}
return "" . mysqli_real_escape_string($DB['DB'], $var) . "";
case ZBX_DB_ORACLE:
if (is_array($var)) {
foreach ($var as $vnum => $value) {
$var[$vnum] = "" . preg_replace('/\\/\'/', '\\\'', $value) . "";
}
return $var;
}
return "" . preg_replace('/\\/\'/', '\\\'', $var) . "";
case ZBX_DB_POSTGRESQL:
if (is_array($var)) {
foreach ($var as $vnum => $value) {
$var[$vnum] = "" . pg_escape_string($value) . "";
}
return $var;
}
return "" . pg_escape_string($var) . "";
case ZBX_DB_SQLITE3:
if (is_array($var)) {
foreach ($var as $vnum => $value) {
$var[$vnum] = "" . $DB['DB']->escapeString($value) . "";
}
return $var;
}
return "" . $DB['DB']->escapeString($var) . "";
}

```

```

default:
return false;
}
}
function zbx_dbcast_2bigint($field) {
global $DB;
if (!isset($DB['TYPE'])) {
return false;
}
switch ($DB['TYPE']) {
case ZBX_DB_DB2:
case ZBX_DB_POSTGRESQL:
case ZBX_DB_SQLITE3:
return 'CAST('.$field.' AS BIGINT)';
case ZBX_DB_MYSQL:
return 'CAST('.$field.' AS UNSIGNED)';
case ZBX_DB_ORACLE:
return 'CAST('.$field.' AS NUMBER(20))';
default:
return false;
}
}
}

```

1.M zbxtg.py

```

#!/usr/bin/python2.7
# coding: utf-8

import sys
import os
import time
import random
import requests
import json
import re
import stat
from os.path import dirname
import zbxtg_settings

class TelegramAPI():
    tg_url_bot_general = "https://api.telegram.org/bot"

    def http_get(self, url):
        res = requests.get(url, proxies=self.proxies)
        answer = res.text
        answer_json = json.loads(answer.decode('utf8'))
        return answer_json

    def __init__(self, key):
        self.debug = False
        self.key = key
        self.proxies = {}

```

```

        self.type = "private" # 'private' for private chats or
'group' for group chats
        self.markdown = False
        self.html = False
        self.disable_web_page_preview = False
        self.disable_notification = False
        self.reply_to_message_id = 0
        self.tmp_uids = None

    def get_me(self):
        url = self.tg_url_bot_general + self.key + "/getMe"
        me = self.http_get(url)
        return me

    def get_updates(self):
        url = self.tg_url_bot_general + self.key + "/getUpdates"
        if self.debug:
            print_message(url)
        updates = self.http_get(url)
        if self.debug:
            print_message("Content of /getUpdates:")
            print_message(updates)
        if not updates["ok"]:
            print_message(updates)
            return updates
        else:
            return updates

    def send_message(self, to, message):
        url = self.tg_url_bot_general + self.key + "/sendMessage"
        message = "\n".join(message)
        params = {"chat_id": to, "text": message,
"disable_web_page_preview": self.disable_web_page_preview,
                "disable_notification": self.disable_notification}
        if self.reply_to_message_id:
            params["reply_to_message_id"] = self.reply_to_message_id
        if self.markdown or self.html:
            parse_mode = "HTML"
            if self.markdown:
                parse_mode = "Markdown"
            params["parse_mode"] = parse_mode
        if self.debug:
            print_message("Trying to /sendMessage:")
            print_message(url)
            print_message("post params: " + str(params))
        res = requests.post(url, params=params, proxies=self.proxies)
        answer = res.text
        answer_json = json.loads(answer.decode('utf8'))
        if not answer_json["ok"]:
            print_message(answer_json)
            return answer_json
        else:
            return answer_json

```



```

def send_photo(self, to, message, path):
    url = self.tg_url_bot_general + self.key + "/sendPhoto"
    message = "\n".join(message)
    params = {"chat_id": to, "caption": message,
"disable_notification": self.disable_notification}
    if self.reply_to_message_id:
        params["reply_to_message_id"] = self.reply_to_message_id
    files = {"photo": open(path, 'rb')}
    if self.debug:
        print_message("Trying to /sendPhoto:")
        print_message(url)
        print_message(params)
        print_message("files: " + str(files))
    res = requests.post(url, params=params, files=files,
proxies=self.proxies)
    answer = res.text
    answer_json = json.loads(answer.decode('utf8'))
    if not answer_json["ok"]:
        print_message(answer_json)
        return answer_json
    else:
        return answer_json

def get_uid(self, name):
    uid = 0
    if self.debug:
        print_message("Getting uid from /getUpdates...")
    updates = self.get_updates()
    for m in updates["result"]:
        if "message" in m:
            chat = m["message"]["chat"]
        elif "edited_message" in m:
            chat = m["edited_message"]["chat"]
        if chat["type"] == self.type == "private":
            if "username" in chat:
                if chat["username"] == name:
                    uid = chat["id"]
            if (chat["type"] == "group" or chat["type"] ==
"supergroup") and self.type == "group":
                if "title" in chat:
                    if chat["title"] == name.decode("utf-8"):
                        uid = chat["id"]
    return uid

def error_need_to_contact(self, to):
    if self.type == "private":
        print_message("User '{0}' needs to send some text bot in
private".format(to))
    if self.type == "group":
        print_message("You need to mention your bot in '{0}'
group chat (i.e. type @YourBot)".format(to))

def update_cache_uid(self, name, uid, message="Add new string to
cache file"):

```

```

        cache_string = "{0};{1};{2}\n".format(name, self.type,
str(uid).rstrip())
        # FIXME
        if self.debug:
            print_message("{0}: {1}".format(message, cache_string))
        with open(self.tmp_uids, "a") as cache_file_uids:
            cache_file_uids.write(cache_string)

    def get_uid_from_cache(self, name):
        uid = 0
        if os.path.isfile(self.tmp_uids):
            with open(self.tmp_uids, 'r') as cache_file_uids:
                cache_uids_old = cache_file_uids.readlines()

                for u in cache_uids_old:
                    u_splitted = u.split(";")
                    if name == u_splitted[0] and self.type ==
u_splitted[1]:
                        uid = u_splitted[2]

        return uid

class ZabbixAPI():
    def __init__(self, server, username, password):
        self.debug = False
        self.server = server
        self.username = username
        self.password = password
        self.proxies = {}
        self.verify = True
        self.cookie = None

    def login(self):

        if not self.verify:
            requests.packages.urllib3.disable_warnings()

        data_api = {"name": self.username, "password": self.password,
"enter": "Sign in"}
        req_cookie = requests.post(self.server + "/", data=data_api,
proxies=self.proxies, verify=self.verify)
        cookie = req_cookie.cookies
        if len(req_cookie.history) > 1 and
req_cookie.history[0].status_code == 302:
            print_message("probably the server in your config file
has not full URL (for example "
                        "'{0}' instead of
'{1}').".format(self.server, self.server + "/zabbix"))
            if not cookie:
                print_message("authorization has failed, url:
{0}".format(self.server + "/"))
                cookie = None

```

```

        self.cookie = cookie

    def graph_get(self, itemid, period, title, width, height,
tmp_dir):
        file_img = tmp_dir + "{0}.png".format(itemid)

        title = requests.utils.quote(title)

        zbx_img_url = self.server + "/chart3.php?period={1}&name={2}"
\
"&width={3}&height={4}&graphtype=0&legend=1" \
                                "&items[0][itemid]={0}&items[0]
[sortorder]=0" \
                                "&items[0][drawtype]=5&items[0]
[color]=00CC00".format(itemid, period, title,
width, height)
        if self.debug:
            print_message(zbx_img_url)
            res = requests.get(zbx_img_url, cookies=self.cookie,
proxies=self.proxies, verify=self.verify)
            res_code = res.status_code
            if res_code == 404:
                print_message("can't get image from
'{0}'".format(zbx_img_url))
                return False
            res_img = res.content
            with open(file_img, 'wb') as fp:
                fp.write(res_img)
            return file_img

    def api_test(self):
        headers = {'Content-type': 'application/json'}
        api_data = json.dumps({"jsonrpc": "2.0", "method":
"user.login", "params":
                                {"user": self.username, "password":
self.password}, "id": 1})
        api_url = self.server + "/api_jsonrpc.php"
        api = requests.post(api_url, data=api_data,
proxies=self.proxies, headers=headers)
        return api.text

def print_message(string):
    string = str(string) + "\n"
    filename = sys.argv[0].split("/")[-1]
    sys.stderr.write(filename + ": " + string)

def list_cut(elements, symbols_limit):
    symbols_count = symbols_count_now = 0
    elements_new = []
    element_last = None

```

```

element_last_list = []
for e in elements:
    symbols_count_now = symbols_count + len(e)
    if symbols_count_now > symbols_limit:
        limit_idx = symbols_limit - symbols_count
        e_list = list(e)
        for idx, ee in enumerate(e_list):
            if idx < limit_idx:
                element_last_list.append(ee)
            else:
                break
        break
    else:
        symbols_count = symbols_count_now + 1
        elements_new.append(e)
if symbols_count_now < symbols_limit:
    return elements, False
else:
    element_last = "".join(element_last_list)
    elements_new.append(element_last)
    return elements_new, True

def main():

    tmp_dir = zbxtg_settings.zbx_tg_tmp_dir

    tmp_cookie = tmp_dir + "/cookie.py.txt"
    tmp_uids = tmp_dir + "/uids.txt"
    tmp_need_update = False # do we need to update cache file with
uids or not

    rnd = random.randint(0, 999)
    ts = time.time()
    hash_ts = str(ts) + "." + str(rnd)

    log_file = "/dev/null"

    zbx_to = sys.argv[1]
    zbx_subject = sys.argv[2]
    zbx_body = sys.argv[3]

    tg = TelegramAPI(key=zbxtg_settings.tg_key)

    tg.tmp_uids = tmp_uids

    if zbxtg_settings.proxy_to_tg:
        tg.proxies = {
            "http": "http://{0}/".format(zbxtg_settings.proxy_to_tg),
            "https":
"https://{0}/".format(zbxtg_settings.proxy_to_tg)
        }

```

```

    zbx = ZabbixAPI(server=zbxtg_settings.zbx_server,
username=zbxtg_settings.zbx_api_user,
                    password=zbxtg_settings.zbx_api_pass)

    if zbxtg_settings.proxy_to_zbx:
        zbx.proxies = {
            "http":
"http://{0}/".format(zbxtg_settings.proxy_to_zbx),
            "https":
"https://{0}/".format(zbxtg_settings.proxy_to_zbx)
        }

    try:
        zbx_api_verify = zbxtg_settings.zbx_api_verify
        zbx.verify = zbx_api_verify
    except:
        pass

    zbxtg_body = (zbx_subject + "\n" + zbx_body).splitlines()
    zbxtg_body_text = []

    settings = {
        "zbxtg_itemid": "0", # itemid for graph
        "zbxtg_title": None, # title for graph
        "zbxtg_image_period": "3600",
        "zbxtg_image_width": "900",
        "zbxtg_image_height": "200",
        "tg_method_image": False, # if True - default send images,
False - send text
        "tg_chat": False, # send message to chat or in private
        "tg_group": False, # send message to chat or in private
        "is_debug": False,
        "is_channel": False,
        "disable_web_page_preview": False,
    }
    settings_description = {
        "itemid": {"name": "zbxtg_itemid", "type": "int"},
        "title": {"name": "zbxtg_title", "type": "str"},
        "graphs_period": {"name": "zbxtg_image_period", "type":
"int"},
        "graphs_width": {"name": "zbxtg_image_width", "type": "int"},
        "graphs_height": {"name": "zbxtg_image_height", "type":
"int"},
        "graphs": {"name": "tg_method_image", "type": "bool"},
        "chat": {"name": "tg_chat", "type": "bool"},
        "group": {"name": "tg_group", "type": "bool"},
        "debug": {"name": "is_debug", "type": "bool"},
        "channel": {"name": "is_channel", "type": "bool"},
        "disable_web_page_preview": {"name":
"disable_web_page_preview", "type": "bool"},
    }

    for line in zbxtg_body:
        if line.find(zbxtg_settings.zbx_tg_prefix) > -1:

```

```

        setting = re.split("[\s\:\=]+", line, maxsplit=1)
        key = setting[0].replace(zbxtg_settings.zbx_tg_prefix +
";", "")
        if len(setting) > 1 and len(setting[1]) > 0:
            value = setting[1]
        else:
            value = True
        if key in settings_description:
            settings[settings_description[key]["name"]] = value
    else:
        zbxtg_body_text.append(line)

    tg_method_image = bool(settings["tg_method_image"])
    tg_chat = bool(settings["tg_chat"])
    tg_group = bool(settings["tg_group"])
    is_debug = bool(settings["is_debug"])
    is_channel = bool(settings["is_channel"])
    disable_web_page_preview =
bool(settings["disable_web_page_preview"])

    # experimental way to send message to the group
https://github.com/ableev/Zabbix-in-Telegram/issues/15
    if sys.argv[0].split("/")[-1] == "zbxtg_group.py" or "--group" in
sys.argv or tg_chat or tg_group:
        tg_chat = True
        tg_group = True
        tg.type = "group"

    if "--debug" in sys.argv or is_debug:
        is_debug = True
        tg.debug = True
        zbx.debug = True
        print_message(tg.get_me())
        print_message("Cache file with uids: " + tg.tmp_uids)
        log_file = tmp_dir + ".debug." + hash_ts + ".log"
        #print_message(log_file)

    if "--markdown" in sys.argv:
        tg.markdown = True

    if "--html" in sys.argv:
        tg.html = True

    if "--channel" in sys.argv or is_channel:
        tg.type = "channel"

    if "--disable_web_page_preview" in sys.argv or
disable_web_page_preview:
        if is_debug:
            print_message("'disable_web_page_preview' option has been
enabled")
        tg.disable_web_page_preview = True

    if not os.path.isdir(tmp_dir):

```

```

        if is_debug:
            print_message("Tmp dir doesn't exist, creating new
one...")
        try:
            os.makedirs(tmp_dir)
            open(tg.tmp_uids, "a").close()
            os.chmod(tmp_dir, stat.S_IRWXU | stat.S_IRWXG |
stat.S_IRWXO)
            os.chmod(tg.tmp_uids, stat.S_IRWXU | stat.S_IRWXG |
stat.S_IRWXO)
        except:
            tmp_dir = "/tmp"
            if is_debug:
                print_message("Using {0} as a temporary
dir".format(tmp_dir))

    uid = None

    if tg.type == "channel":
        uid = zbx_to
    else:
        zbx_to = zbx_to.replace("@", "")

    if not uid:
        uid = tg.get_uid_from_cache(zbx_to)

    if not uid:
        uid = tg.get_uid(zbx_to)
        if uid:
            tmp_need_update = True
    if not uid:
        tg.error_need_to_contact(zbx_to)
        sys.exit(1)

    if tmp_need_update:
        tg.update_cache_uid(zbx_to, str(uid).rstrip())

    if is_debug:
        print_message("Telegram uid of {0} '{1}':
{2}".format(tg.type, zbx_to, uid))

    # add signature, turned off by default, you can turn it on in
config
    try:
        if zbxtg_settings.zbx_tg_signature:
            zbxtg_body_text.append("--")
            zbxtg_body_text.append(zbxtg_settings.zbx_server)
    except:
        pass

    # replace text with emojis
    if hasattr(zbxtg_settings, "emoji_map"):
        zbxtg_body_text_emoji_support = []
        for l in zbxtg_body_text:

```

```

        l_new = l
        for k, v in zbxtg_settings.emoji_map.iteritems():
            l_new = l_new.replace("{}" + k + "}", v)
            zbxtg_body_text_emoji_support.append(l_new)
        zbxtg_body_text = zbxtg_body_text_emoji_support

    if not tg_method_image:
        result = tg.send_message(uid, zbxtg_body_text)
        if not result["ok"]:
            if result["description"].find("migrated") > -1 and
result["description"].find("supergroup") > -1:

                migrate_to_chat_id = result["parameters"]
["migrate_to_chat_id"]
                tg.update_cache_uid(zbx_to, migrate_to_chat_id,
message="Group chat is migrated to supergroup, updating cache file")
                uid = migrate_to_chat_id
                result = tg.send_message(uid, zbxtg_body_text)
            else:
                zbx.login()
                if not zbx.cookie:
                    print_message("Login to Zabbix web UI has failed, check
manually...")
                else:
                    zbxtg_file_img = zbx.graph_get(settings["zbxtg_itemid"],
settings["zbxtg_image_period"],
settings["zbxtg_image_width"],
settings["zbxtg_image_height"], tmp_dir)
                    #zbxtg_body_text, is_modified = list_cut(zbxtg_body_text,
200)
                    result = tg.send_message(uid, zbxtg_body_text)
                    message_id = result["result"]["message_id"]
                    tg.reply_to_message_id = message_id
                    tg.disable_notification = True
                    if not zbxtg_file_img:
                        tg.send_message(uid, ["Can't get graph image, check
script manually, see logs, or disable graphs"])
                        print_message("Can't get image, check URL manually")
                    else:

                        zbxtg_body_text = ""
                        """
                        if is_modified:
                            print_message("probably you will see
MEDIA_CAPTION_TOO_LONG error, "
symbols, "
in-Telegram/issues/9"
                                "the message has been cut to 200
                                "https://github.com/ableev/Zabbix-
                                "#issuecomment-166895044")
                        """

```



```

        if tg.send_photo(uid, zbxtg_body_text,
zbxtg_file_img):
            os.remove(zbxtg_file_img)

```

```

if __name__ == "__main__":
    main()

```

1.N zbxtg_settings.py

```

# -*- coding: utf-8 -*-

tg_key = "5963256920:AAEd9jklp7pBblhiBf6dntb6vdXdMZcnEM" # telegram
bot api key

zbx_tg_prefix = "zbxtg" # variable for separating text from script
info
zbx_tg_tmp_dir = "/tmp/" + zbx_tg_prefix # directory for saving
caches, uids, cookies, etc.
zbx_tg_signature = False # the signature in your telegram message
for example your company name

zbx_server = "http://10.222.1.3/zabbix" # zabbix server full url (ip
or dns)
zbx_api_user = "api" #user baraye zabbix
zbx_api_pass = "api" #password of user
zbx_api_verify = True # True - do not ignore self signed
certificates, False - ignore


proxy_to_zbx = None
proxy_to_tg = None

#proxy_to_zbx = "proxy.local:3128"
#proxy_to_tg = "proxy.local:3128"

emoji_map = {
    "ok": " ",
    "problem": " ",
    "info": " ",
    "warning": " ",
    "disaster": " ",
    "bomb": " ",
    "fire": " ",
    "hankey": " "
}

```

2. Plagiarism Check (turnitin.com)



Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: **ujititikhamdani@gmail.com 1**
 Assignment title: **No Repository**
 Submission title: **BAB1-BAB5_Revisi_Uji Hamdani**
 File name: **BAB1-BAB5.pdf**
 File size: **3.37M**
 Page count: **132**
 Word count: **16,350**
 Character count: **98,460**
 Submission date: **24-Jun-2023 05:59AM (UTC+0300)**
 Submission ID: **2121693988**

BAB1
PENDAHULUAN

1.1. Latar belakang

Dalam rangka upaya digitalisasi pembelajaran berbasis E-Learning, Perencanaan Sistem Informatika (Perencanaan Sistem Informasi) sebagai salah satu aspek yang sangat penting dalam dunia bisnis, terutama di era digital ini. Perkembangan teknologi informasi yang pesat telah membawa dampak signifikan terhadap berbagai aspek kehidupan manusia, terutama di era digital ini. Salah satu aspek yang paling signifikan adalah transformasi digital yang telah mengubah cara kita berinteraksi, bekerja, dan belajar. Salah satu aspek yang paling signifikan adalah transformasi digital yang telah mengubah cara kita berinteraksi, bekerja, dan belajar. Salah satu aspek yang paling signifikan adalah transformasi digital yang telah mengubah cara kita berinteraksi, bekerja, dan belajar.

Copyright 2023 Turnitin. All rights reserved.

Turnitin Originality Report

Processed on: 24-Jun-2023 05:59 +03
 ID: 2121693988
 Word Count: 16350
 Submitted: 1

BAB1-BAB5_Revisi_Uji Hamdani By
 ujititikhamdani@gmail.com 1

Similarity Index	Similarity by Source	
24%	Internet Sources:	23%
	Publications:	6%
	Student Papers:	11%