

**PERANCANGAN APLIKASI ABSENSI SISWA SECARA  
DARING DI LINGKUNGAN SMK YADIKA  
TANJUNGSARI**

**SKRIPSI**

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

**Oleh**

Maulia Fahrudilah

362041003



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN  
KOMPUTER INDONESIA MANDIRI  
2021**

**LEMBAR PENGESAHAN**

**PERANCANGAN APLIKASI ABSENSI SISWA SECARA  
DARING DI LINGKUNGAN SMK YADIKA  
TANJUNGSARI**

Oleh

Maulia Fahrudillah  
362041003

Skripsi ini telah diterima dan disahkan untuk  
memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada

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

Bandung, Juni 2021  
Disahkan oleh

Ketua Program Studi,

Dosen Pembimbing,

Chalifa Chazar, S.T., M.T.  
NIDN. 0421098704

Novi Rukhviyanti, ST.M.SI  
NIDN. 04247602

**LEMBAR PERSETUJUAN REVISI**

**PERANCANGAN APLIKASI ABSENSI SISWA SECARA  
DARING DI LINGKUNGAN SMK YADIKA  
TANJUNGSARI**

Oleh

Maulia Fahrudillah  
362041003

Telah melakukan sidang skripsi dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang skripsi.

Bandung, Juni 2021  
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Novi Rukhviyanti, S.T., M.SI.	Pembimbing	
2.	Chairuddin, Ir., M.T., M.M., Dr.	Penguji 1	
3.	Haryoso Wicaksono, S.SI., M.M., M.Kom.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika

Chalifa Chazar, S.T., M.T.  
NIDN. 0421098704

## **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) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian 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, Juni 2021  
Yang membuat pernyataan

Maulia Fahrudillah  
362041003

## ABSTRAK

### PERANCANGAN APLIKASI ABSENSI SISWA SECARA DARING DI LINGKUNGAN SMK YADIKA TANJUNGSARI

Oleh

Maulia Fahrudillah

362041003

*Absensi* dapat dikatakan suatu tanda kehadiran yang merupakan bagian dari aktifitas pelaporan yang ada di setiap sekolah untuk menandai kehadiran siswa, pada umumnya absensi masih dilakukan secara manual, dimana hal ini sangat tidak efisien apalagi dikarenakan situasi pandemik Covid19 yang memberikan perubahan cukup besar di semua sektor pendidikan karena dibatasinya interaksi sosial. Namun kegiatan pendidikan harus tetap berjalan oleh sebab itu sangat perlu untuk diterapkan sebuah teknologi yang dapat membantu proses absensi di sekolah agar lebih mudah dan efisien. Dalam penelitian ini dirancang sebuah sistem absensi yang dapat dilakukan menggunakan *smartphone android* dengan memanfaatkan sistem kamera dan *Global Positioning System* (GPS) untuk menyampaikan informasi proses absensi secara daring.

Kata Kunci : Absensi, Android, *Global Positioning System*

## **ABSTRACT**

### **DESIGN OF ONLINE STUDENT ATTENDANCE APPLICATION AT SMK YADIKA TANJUNGSARI**

Oleh

Maulia Fahrudillah

362041003

Attendance can be said to be a sign of attendance which is part of reporting activities in every school to mark student attendance, in general attendance is still done manually, which is very inefficient especially due to the Covid19 pandemic situation which has made quite a big change in all education sectors. because of limited social interaction. However, educational activities must continue, therefore it is very necessary to apply a technology that can help the attendance process in schools to make it easier and more efficient. In this research, an attendance system is designed that can be done using an android smartphone by utilizing the camera system and the Global Positioning System (GPS) to convey information.process attendance online.

**Keywords :** Attendance, Android, Global Positioning System

## KATA PENGANTAR

Segala puji serta syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat, hidayah serta kekuatan. Tak lupa juga solawat beserta salam semoga selalu terlimpah curahkan kepada jungjungan alam HABBIBANA WANNABIANA MUHAMMAD S.A.W., kepada keluarganya, kepada saudaranya, kepada sahabat-sahabatnya, dan semoga sampai kepada kita selaku umatnya aamin ya rabbal alamin.

Alhamdulillah pada kesempatan kali ini penulis telah menyelesaikan pembuatan skripsi dengan judul “Perancangan Aplikasi Absensi Siswa Secara Daring Di Lingkungan SMK Yadika Tanjungsari”. Semoga apa yang telah penulis buat dapat bermanfaat khususnya bagi penulis dan umumnya bagi pembaca. Dalam penelitian ini tidak lepas dari bantuan dan dukungan dari berbagai pihak, baik moril, materil maupun spiritual. Oleh karena itu, dalam kesempatan ini saya sampaikan penghormatan dan ucapan terima kasih yang sebesar-besarnya kepada:

1. ALLAH S.W.T., karena berkat, rahmat dan ridho-NYA bisa melaksanakan perkuliahan dengan baik dan lancar.
2. Orang Tua tercinta, karena atas izin dan motivasi merekalah dapat melaksanakan perkuliahan dan penelitian dengan penuh semangat.
3. Istri tercinta Gilang Citra dan anak-anaku Ismail, Ishaq yang telah memberikan dukungan dan semangat serta motivasi.

4. Ibu Novi Rukhviyanti, ST.M.SI. juga selaku Dosen pembimbing yang telah membantu dalam memberi arahan serta masukan dalam penelitian yang di lakukan.
5. Ibu Chalifa Chazar ST., MT. selaku Ketua Program Studi Teknik Informatika STMIK Indonesia Mandiri Bandung.
6. Bapak Chairuddin Ir., M.T., M.M., Dr. selaku Ketua STMIK-IM Bandung
7. Bapak Endi Rohendi M.Pd Selaku Kepala Sekolah SMK Yadika Tanjungsari, beserta staff dan guru yang telah membantu dan memberikan izin penelitian.
8. Tak lupa juga kepada rekan-rekan seperjuangan jurusan Teknik Informatika STMIK Indonesia Mandiri Bandung yang selama ini telah menyemangati dalam melaksanakan perkuliahan dan penelitian.
9. Dengan segala kerendahan hati penulis mengucapkan banyak terima kasih kepada semua pihak-pihak yang tidak bisa saya sebutkan satu persatu yang telah membantu dalam menyelesaikan skripsi ini.

Penulis menyadari skripsi ini masih terdapat kekurangan dan jauh dari kesempurnaan, dikarenakan kurangnya kemampuan dan pengetahuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun. Dan semoga skripsi ini dapat memberikan manfaat di kemudian hari.

Bandung, Juni 2021  
Penulis,

Maulia Fahrudillah  
362041003



## DAFTAR ISI

SURAT PERNYATAAN.....	i
ABSTRAK.....	ii
ABSTRACT.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	1
BAB IPENDAHULUAN.....	2
1. 1 Latar Belakang .....	2
1. 2 Identifikasi Masalah.....	6
1. 3 Tujuan Penelitian.....	6
1. 4 Batasan Masalah .....	7
1. 5 Metoda Penelitian .....	7
1. Analisis.....	7
2. Desain.....	8
3. Pengkodean.....	8
4. Pengujian.....	8
1. 6 Sistematika Penulisan .....	9
BAB IILANDASAN TEORI.....	11
2. 1 Absensi.....	11
2. 2 Siswa.....	12
2. 3 Guru.....	13
2. 4 Kepala Sekolah.....	14
2. 5 Aplikasi.....	15
2. 6 Perangkat keras (Hardwere).....	16
2. 7 Perangkat lunak (Softwere).....	17
2. 8 Daring.....	17
2. 9 Google (Google Form Dan Google Sheets).....	18
2. 10 Waterfall ( Metode Air Terjun ).....	19
1. Analisis.....	20
2. Desain.....	20
3. Pengodean.....	21
4. Pengujian.....	21
2. 11 Flowmap.....	22
2. 12 Diagram Konteks.....	24
2. 14 <i>Use Case Diagram</i> .....	25
2. 15 <i>Class Diagram</i> .....	27
2. 16 <i>Activity Diagram</i> .....	28
2. 17 <i>Basis Data (Data base)</i> .....	28
2. 18 <i>Entity Relationship Diagram (ERD)</i> .....	30
2. 19 <i>Black Box Testing</i> .....	32
2. 20 <i>Global Positioning System (GPS)</i> .....	35
BAB IIIANALISIS MASALAH DAN PERANCANGAN PROGRAM.....	38
3.1 Analisis.....	38
3.1.1 Analisis Data.....	38
3.1.2 Analisis Sistem yang Berjalan.....	45
3.1.3 Analisis Masalah.....	46
3.1.4 Analisis Sistem yang Akan Dibangun.....	46

3.2 Desain.....	48
3.2.1 Deskripsi Aktor.....	48
3.2.2 Deskripsi <i>Use Case</i> .....	49
3.2.3 <i>Use Case Diagram</i> .....	50
3.2.4 <i>Skenario Use Case</i> .....	52
3.2.5 <i>Activity Diagram</i> .....	56
3.2.6 <i>Sequence Diagram</i> .....	62
3.2.7 <i>Class Diagram</i> .....	69
3.2.8 <i>Entity Relationship Diagram (ERD)</i> .....	70
3.2.9 <i>Perancangan Antar Muka</i> .....	70
3.2.10 <i>Konstruksi dan Pembangunan (Coding and Testing)</i> .....	72
BAB IV IMPLEMENTASI DAN UJI COBA.....	74
4.1 Implementasi.....	74
4.1.1 Perangkat Keras ( <i>Hardware</i> ).....	74
4.1.2 Perangkat Lunak ( <i>Software</i> ).....	75
4.2 Pengkodean.....	76
4.2.1 Kamus Data.....	76
4.2.2 Relasi Tabel.....	80
4.3 Implementasi Antarmuka ( <i>Interface</i> ).....	81
4.3.1 Tampilan Antar Muka Menu Login.....	81
4.3.2 Tampilan Antar Muka Main Menu.....	82
4.3.3 Tampilan Antar Muka Input Data Staff.....	82
4.3.4 Tampilan Antar Muka Input Data Siswa.....	83
4.3.5 Tampilan Antar Muka Input Jadwal Mata Pelajaran.....	83
4.3.6 Tampilan Menu Utama di Smartphone.....	84
4.3.7 Tampilan Menu Lihat Jadwal Siswa Di Smartphone.....	85
4.3.8 Tampilan Menu Absensi Siswa Di Smartphone.....	85
4.3.9 Tampilan Menu Report Absensi Siswa.....	86
4.4 Uji Coba ( <i>Testing</i> ).....	87
4.4.1 Pengujian Login dan Logout User.....	87
4.4.2 Pengujian Proses Sistem Absensi.....	88
BAB V PENUTUP.....	90
5.1 Kesimpulan.....	90
5.2 Saran.....	90
DAFTAR PUSTAKA.....	92
LAMPIRAN LIST PROGRAM.....	94

## DAFTAR TABEL

Tabel 3. 1	Data <i>User Requirement Cycle</i> Pertama Pada Responden.....	41
Tabel 3. 2	Deskripsi Aktor.....	48
Tabel 3. 3	Deskripsi <i>Use Case</i> .....	49
Tabel 3. 4	<i>Scenario Use Case Login</i> .....	52
Tabel 3. 5	<i>Scenario Use Case Logout</i> .....	53
Tabel 3. 6	<i>Scenario Use Case Absensi</i> .....	53
Tabel 3. 7	<i>Scenario Use Case Kelola Gedung</i> .....	53
Tabel 3. 8	<i>Scenario Use Case Kelola Staff</i> .....	54
Tabel 3. 9	<i>Scenario Use Case Kelola Kelas</i> .....	54
Tabel 3. 10	<i>Scenario Use Case Kelola Mata Pelajaran</i> .....	54
Tabel 3. 11	<i>Scenario Use Case Kelola Siswa</i> .....	55
Tabel 3. 12	<i>Scenario Use Case Kelola Mata Pelajaran</i> .....	55
Tabel 3. 13	<i>Scenario Use Case Kelola Report Absensi</i> .....	56
Tabel 4. 14	Struktur Tabel m_siswa.....	76
Tabel 4. 15	Struktur Tabel m_matpel.....	77
Tabel 4. 16	Struktur Tabel m_gedung_sekolah.....	77
Tabel 4. 17	Struktur Tabel m_kelas.....	77
Tabel 4. 18	Struktur Tabel m_staff.....	78
Tabel 4. 19	Struktur Tabel tr_absensi.....	78
Tabel 4. 20	Struktur Tabel tr_jadwal_matpel.....	79
Tabel 4. 21	Tabel Pengujian Login dan Logout User.....	87
Tabel 4. 22	Tabel Pengujian Sistem Absensi Siswa Dan Guru.....	88

## DAFTAR GAMBAR

Gambar 2. 1 Tahapan Model Air Terjun ( <i>Waterfall</i> ).....	20
Gambar 2. 2 Tabel Simbol <i>Flowmap</i> .....	24
Gambar 2. 3 Tabel Simbol Diagram Konteks.....	25
Gambar 2. 4 Tabel Simbol Use Case Diagram.....	26
Gambar 2. 5 Tabel Simbol Class Diagram.....	27
Gambar 2. 6 Tabel Simbol <i>Activity Diagram</i> .....	28
Gambar 2. 7 Tabel Simbol-simbol ERD notasi Chen.....	31
Gambar 2. 8 Tabel Simbol-simbol ERD notasi Chen.....	31
Gambar 2. 9 Bentuk hubungan relasi ERD.....	32
Gambar 3. 1 Diagram hasil User Requirement siswa- Cycle Pertama.....	41
Gambar 3. 2 Grafik hasil <i>User Requirement</i> guru-Cycle Pertama.....	43
Gambar 3. 3 <i>Flowmap</i> Sistem Yang Berjalan di SMK Yadika Tangjungsari.....	45
Gambar 3. 4 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun.....	47
Gambar 3. 5 Use case diagram siswa.....	51
Gambar 3. 6 Use case diagram Guru Mata Pelajaran, Wali kelas, Kepala Sekolah, dan Admin.....	51
Gambar 3. 7 <i>Activity Diagram</i> Login Siswa Guru, Wali Kelas, Admin, Kepala Sekolah.....	56
Gambar 3. 8 <i>Activity Diagram</i> Logout.....	57
Gambar 3. 9 <i>Activity Diagram</i> Melihat Mata Pelajaran Siswa.....	57
Gambar 3. 10 <i>Activity Diagram</i> Absensi Siswa.....	58
Gambar 3. 11 <i>Activity Diagram</i> Admin Input Data Gedung.....	58
Gambar 3. 12 <i>Activity Diagram</i> Admin Input Data Staff Pengajar.....	59
Gambar 3. 13 <i>Activity Diagram</i> Admin Input Data Kelas.....	59
Gambar 3. 14 <i>Activity Diagram</i> admin input data mata Pelajaran.....	60
Gambar 3. 15 Diagram Aktiviti Kelola Data Siswa (Admin, Walikelas).....	60
Gambar 3. 16 <i>Activity Diagram</i> Admin Dan Wali Kelas Input Data Jadwal Mata Pelajaran.....	61
Gambar 3. 17 <i>Activity Diagram</i> report absensi wali kelas dan kepala sekolah.....	61
Gambar 3. 18 <i>Activity Diagram</i> Kelola Data Absensi Guru.....	62
Gambar 3. 19 <i>Sequence Diagram</i> admin input data gedung sekolah.....	63
Gambar 3. 20 <i>Sequence Diagram</i> admin input data staff.....	63
Gambar 3. 21 <i>Sequence Diagram</i> Admin Input Data Kelas.....	64
Gambar 3. 22 <i>Sequence Diagram</i> Admin Input Data Mata Pelajaran.....	64
Gambar: 3.23. Gambar 3. 23 <i>Sequence Diagram</i> Admin Input Data Siswa.....	65
Gambar 3. 24 <i>Sequence Diagram</i> Admin Input Data Jadwal Mata Pelajaran.....	65
Gambar 3. 25 <i>Sequence Diagram</i> Report Data Absensi Oleh Wali Kelas Dan Kepala Sekolah.....	66
Gambar 3. 26 <i>Sequence Diagram</i> Kelola Absensi Oleh Guru Mata Pelajaran.....	66
Gambar 3. 27 <i>Sequence Diagram</i> Absensi Siswa.....	67
Gambar 3. 28 <i>Sequence Diagram</i> Lihat Mata Pelajaran.....	67
Gambar 3. 29 <i>Sequence Diagram</i> Absensi Siswa.....	68
Gambar 3. 30 <i>Sequence Diagram</i> Login Absensi Siswa.....	68
Gambar 3. 31 <i>Class Diagram</i> absensi Siswa.....	69
Gambar 3. 32 Entity Relationship Diagram (ERD) Absensi Siswa.....	70
Gambar 3. 33 Tampilan Antar Muka Menu Login.....	71
Gambar 3. 34 Tampilan Antar Muka Tampilan Menu.....	71
Gambar 3. 35 Tampilan Antar Muka Menu Aplikasi Android.....	72
Gambar 4. 1 Relasi Tabel sistem aplikasi absensi siswa.....	80
Gambar 4. 2 Gambar Antar Muka Menu Login.....	81
Gambar 4. 3 Gambar Tampilan Antar Muka Main Menu Sistem Absensi.....	82
Gambar 4. 4 Gambar Tampilan Antar Muka Menu Input Data Staff Pengajar.....	82

<i>Gambar 4. 5 Gambar Tampilan Antar Muka Menu Input Data Staff Pengajar.....</i>	<b>83</b>
<i>Gambar 4. 6 Gambar Tampilan Antar Muka Menu Input Data Staff.....</i>	<b>83</b>
<i>Gambar 4. 7 Gambar Tampilan Antar Muka Menu Login Data Siswa.....</i>	<b>84</b>
<i>Gambar 4. 8 Gambar Tampilan Antar Muka Menu Utama Siswa.....</i>	<b>84</b>
<i>Gambar 4. 9 Gambar Tampilan Antar Muka Menu Lihat Jadwal Mata Pelajaran Siswa Di Smartphone.....</i>	<b>84</b>
<i>Gambar 4. 10 Gambar Tampilan Antar Muka Menu Absensi Siswa Melalui SmartPhone.....</i>	<b>85</b>
<i>Gambar 4. 11 Tampilan Menu Ambil Foto Dan Lakukan Absensi Siswa Di Smatrphone.....</i>	<b>86</b>
<i>Gambar 4. 12. Laporan Hasil Absensi Siswa.....</i>	<b>86</b>

## **BAB I**

### **PENDAHULUAN**

#### **1. 1 Latar Belakang**

Pandemik COVID-19 yang mengepung dunia belum bisa diredam. Semua orang lantas mengambil jarak demi memutus rantai penularan COVID-19. Tempat-tempat ibadah kini mulai sepi, agenda-agenda massa dihilangkan, sekolah dan kampus dibatasi aktifitasnya. COVID-19 juga memberikan dampak serius di sektor pendidikan, baik di Indonesia maupun secara global.

Presiden Jokowi dalam Rapat Terbatas yang diselenggarakan pada Selasa (24/3) bersama menteri terkait, sudah ketok palu. Hasilnya, pemerintah mengumumkan Ujian Nasional (UN) di tahun 2020 resmi ditiadakan. Mulai dari tingkat Sekolah Dasar (SD) hingga setingkat Sekolah Menengah Atas (SMA). “Presiden Joko Widodo memutuskan meniadakan Ujian Nasional (UN) untuk tahun 2020 yang sebelumnya sudah ada kesepakatan UN dihapus mulai tahun 2021,” ujar Juru Bicara Presiden Jokowi, Fadjoel Rachman, dalam rilisnya, Selasa (24/3) berakibat Tenaga dan peserta didik di seluruh dunia merasakan betul

dampak yang luar biasa dari wabah virus COVID-19 yang pertama kali muncul di China. Akibat pandemik yang sudah menyebar ke 156 negara itu, banyak sekolah-sekolah terpaksa diliburkan. Karena dengan adanya pengurangan dan peniadaan Kegiatan Belajar Mengajar (KBM) di sekolah dinilai menjadi keputusan yang tepat untuk mengurangi kontak dan mengurangi kerumunan massa (*Social and Physical Distancing*) pada masa pandemi Covid-19.

(<https://kumparan.com/kumparansains/imbas-pandemi-virus-corona-bagi-dunia-pendidikan-indonesia-dan-global-1t5YVXRYAbo/full>, diakses tanggal 14/3/2021.19.00).

Hal ini sesuai dengan Menteri Pendidikan dan Kebudayaan Republik Indonesia terkait Surat Edaran Nomor 4 Tahun 2020 tentang Pelaksanaan Kebijakan Pendidikan dalam Masa Darurat Penyebaran Corona Virus Disease (COVID-19) (<https://www.kemdikbud.go.id/main/blog/2020/03/se-mendikbud-pelaksanaan-kebijakan-pendidikan-dalam-masa-darurat-penyebaran-covid19>, diakses tanggal 14/3/2021.20.00).

Semua kegiatan belajar mengajar seperti pembelajaran, diskusi dan kegiatan lain yang sejenis diupayakan untuk tetap berjalan dengan metode jarak jauh atau lebih di kenal dengan istilah *daring* (dalam jaringan) dan banyak orang yang bekerja di rumah atau dengan istilah *Work From Home* (WFH).

Banyak kegiatan dilaksanakan dengan menggunakan metode *daring* dan *Work From Home* dengan menggunakan metode *synchronous*, seperti *Webex*, *Zoom*, *Skype*, *Microsoft teams*, *Googlemet* dan lain-lain. Juga *Asynchronous*, semisal *eLisa*, *Elok*, *Elearning Farmasetika*, *Whatsapp Group* dan lain-lain,

dengan tetap memperhatikan kelanjutan proses belajar mengajar dan ketercapaian mutu pembelajaran yang telah direncanakan.

Termasuk di SMK Yadika Tanjungsari, kegiatan belajar mengajar dilaksanakan dengan menggunakan metode *daring* dengan menggunakan *WhatsApp*, *Timestamp camera pro*, *Google meet*, *Google classroom*, *Google Sheets* dan *Google form*. Sistem pembelajaran *daring* (dalam jaringan) merupakan sistem pembelajaran tanpa tatap muka secara langsung antara guru dan siswa tetapi dilakukan melalui *online* yang menggunakan jaringan *internet*. Guru harus memastikan kegiatan belajar mengajar tetap berjalan, meskipun siswa berada di rumah. Solusinya, guru dituntut dapat mendesain media pembelajaran sebagai inovasi dengan memanfaatkan media *daring* (dalam jaringan).

Akibat perubahan metode pembelajaran secara *daring* maka sistem kehadiran (*absensi*) siswa juga harus dilakukan secara *daring*. *Absensi* tak terlepas dari kegiatan belajar mengajar, dimana *absensi* akan memberikan banyak fungsi yang penting berkaitan dengan siswa untuk mengikuti proses belajar mengajar dan mengikuti ujian. Kegiatan *absensi* sebelum masa pandemik umumnya masih dilakukan dengan menggunakan kertas, lalu siswa dipanggil satu persatu oleh guru setiap mata pelajaran.

Di SMK Yadika selama pandemik sistem absensi dilakukan dengan cara siswa pada awal semester harus mengirimkan foto menggunakan aplikasi *Timestamp camera Pro* untuk merekam lokasi siswa, selanjutnya *absensi* dilakukan menggunakan *Google Form* setiap pelajaran berlangsung.

Sistem pendidikan dapat berjalan dengan baik bila siswa dapat hadir dalam proses belajar mengajar sehingga diperlukan pengawasan kedisiplinan yang dilakukan oleh sekolah. Pengawasan yang dilakukan tidak serta-merta menjadi tanggung jawab pihak sekolah, namun peran serta secara aktif perlu dilakukan juga oleh para orang tua/wali murid. Dalam prakteknya, peran pengawasan terhadap kehadiran siswa hanya dibebankan kepada pihak sekolah, namun banyak juga terjadi kesalahan pengawasan yang dilakukan oleh pihak sekolah yakni belum adanya mekanisme pelaporan secara terkomputerisasi yang berhubungan dengan kehadiran siswa.

Masalah ini terjadi karena setiap siswa melakukan *absensi* dengan menggunakan *Google Form* untuk setiap pelajaran. Pada akhir pelajaran setiap guru harus merekapitulasi setiap absensi yang dilakukan oleh siswa, selanjutnya melaporkan hasilnya kepada setiap wali kelas. Selanjutnya wali kelas akan merekap kehadiran siswa untuk dilaporkan kepada kepala sekolah. Hal ini menyebabkan ketidak efisienan dalam laporan absensi, karena masih dilakukan secara manual.

Oleh karena itu penelitian ini bertujuan untuk mengembangkan aplikasi absensi yang mempermudah siswa dalam melakukan absensi, dan juga mempermudah guru, wali kelas dan kepala sekolah untuk memantau kehadiran siswa.



## 1. 2 Identifikasi Masalah

Dari beberapa uraian yang dikemukakan pada latar belakang, maka dapat diidentifikasi masalah-masalah sebagai berikut :

1. Bagaimana mempermudah siswa untuk melakukan absensi dalam masa pembelajaran secara *daring*?
2. Bagaimana mempermudah guru dalam merekapitulasi kehadiran siswa?
3. Bagaimana mempermudah laporan rekapitulasi kehadiran bagi wali kelas dan kepala sekolah untuk proses pemantauan?

## 1. 3 Tujuan Penelitian

Berdasarkan hasil identifikasi masalah yang telah disebutkan di atas, maka tujuan penelitian ini adalah:

1. Membangun aplikasi absensi yang memudahkan siswa untuk melakukan absensi secara *daring* melalui *smart phone*.
2. Membangun aplikasi *absensi* yang dapat secara otomatis membuat rekapitulasi laporan absensi setiap siswa dalam pelajaran untuk mempermudah kinerja guru.
3. Membangun aplikasi *absensi* yang dapat mempermudah pelaporan *absensi* siswa bagi guru untuk diserahkan kepada wali kelas dan kepala sekolah untuk melakukan pemantauan kehadiran siswa.

#### **1.4 Batasan Masalah**

Berdasarkan identifikasi masalah di atas, diperlukan batasan masalah untuk membatasi ruang lingkup penelitian ini. Batasan masalah penelitian ini antara lain:

1. Aplikasi absensi ini hanya digunakan untuk merekapitulasi absensi siswa di lingkungan SMK Yadika Tanjungsari
2. Aplikasi ini tidak dapat memberikan laporan kepada orang tua siswa
3. Metode pengembangan aplikasi yang digunakan adalah Metode *Waterfall*, tetapi hanya sampai tahapan pengujian

#### **1.5 Metoda Penelitian**

Metode penelitian yang digunakan dalam penelitian ini mengacu kepada metode pengembangan perangkat lunak. Metode pengembangan perangkat lunak yang digunakan adalah Metode *Waterfall*. Berikut ini adalah tahapan Metode *Waterfall*, yaitu:

##### **1. Analisis**

Analisis kebutuhan perangkat lunak merupakan proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh sekolah.

Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

## 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain ini juga perlu didokumentasikan.

## 3. Pengkodean

Pada tahapan ini adalah pembuatan kode program yang mana desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

## 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logis dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. Pendukung (*Support*)

Tahap pendukung (*support*) atau pemeliharaan (*maintenance*) ini diperlukan karena tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru (Rosa A dan Shalahuddin, 2019).

## **1. 6 Sistematika Penulisan**

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan skripsi ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bab I menguraikan tentang latar belakang melakukan penelitian, identifikasi masalah, menentukan tujuan dan kegunaan penelitian, pembatasan masalah dan metode yang digunakan dalam penelitian serta sistematika penulisan.

### **BAB II LANDASAN TEORI**

Berisi referensi terkait dengan topik penelitian yang ditujukan untuk menunjang penulisan skripsi. Pada Bab ini, mahasiswa wajib mencantumkan

5(lima) buah referensi ilmiah yang memiliki latar belakang yang kuat minimal lima tahun terakhir dengan judul yang dipilih. Referensi dapat diambil dari jurnal, karya ilmiah, *prosiding*, buku teks, *white paper*, majalah ilmiah dan tesis yang relevan dengan topik peneliti.

### **BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM**

Berisi tentang proses perancangan program berdasarkan analisa permasalahan sesuai dengan metoda pengembangan/metoda perancangan/metoda penelitian yang dipilih.

### **BAB IV IMPLEMENTASI DAN UJI COBA**

Berisi tentang penjelasan dalam pengoperasian program secara bertahap. Tuliskan juga hasil evaluasi implementasi program, termasuk kelebihan dan kekurangannya. Untuk evaluasi dapat menggunakan metode *kualitatif*, *kuantitatif*, atau metode evaluasi lain yang sesuai.

### **BAB V PENUTUP**

Berisi kesimpulan yang didapatkan dari hasil penelitian. Tuliskan juga saran yang bertujuan untuk peningkatan topik skripsi yang dapat digunakan di masa mendatang.

## BAB II

### LANDASAN TEORI

#### 2.1 Absensi

*Absensi* adalah sebuah kegiatan pengambilan data guna mengetahui jumlah kehadiran pada suatu acara. Setiap kegiatan yang membutuhkan informasi mengenai peserta tentu akan melakukan *absensi*. (<http://e-journal.uajy.ac.id/1692/2/1TF05139.pdf>) (diakses tanggal 26/03/2021.18.00)

(<https://jurnal.untirta.ac.id/index.php/jis/article/view/498>) (diakses tanggal 26/03/2021.18.30) *absensi* adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen *institusi* itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan.

Kita mengenal beberapa jenis *absensi*. Yang membedakan jenis jenis *absensi* tersebut adalah cara penggunaannya, dan tingkat daya gunanya secara umum jenis-jenis absensi dapat di kelompokkan menjadi dua, yaitu;

##### 1. Absensi Manual

*Absensi* manual adalah cara pengentrian kehadiran dengan cara menggunakan pena (tanda tangan)

## 2. Absensi Non Manual (Dengan Menggunakan Alat)

*Absensi* non manual adalah suatu cara pengentrian kehadiran dengan menggunakan sistem *terkomputerisasi*, bisa menggunakan kartu dengan *barcode*, *finger print* ataupun dengan mengentrikan NIP ( Nomor Induk Pegawai) dan sebagainya.

Pengelolaan *absensi* dengan memanfaatkan kelebihan teknologi informasi adalah:

1. *Absensi* dapat dilakukan dengan lebih mudah dan cepat
2. *Informasi* hasil pencarian yang disajikan lebih lengkap
3. Mempermudah dalam melakukan *Entry absensi*
4. Memudahkan pembuatan laporan *absensi*

### 2.2 Siswa

Siswa dalam istilah adalah peserta didik pada jenjang pendidikan menengah pertama dan menengah atas. Siswa adalah komponen masukan dalam sistem pendidikan yang selanjutnya diproses dalam proses pendidikan.

Menurut Prof. Dr. Shafique Ali Khan, Siswa merupakan orang yang datang ke suatu lembaga untuk memperoleh atau mempelajari beberapa tipe pendidikan, selanjutnya orang ini disebut pelajar atau orang yang mempelajari ilmu pengetahuan berapapun usianya, dari manapun, siapa pun, dalam bentuk apapun, dengan biaya apapun untuk meningkatkan pengetahuan dan moral pelaku belajar.

Menurut UU RI No. 20 tahun 2003 pasal 1 ayat 4 Siswa adalah anggota masyarakat yang berusaha meningkatkan kualitas dirinya dengan melalui proses pendidikan tertentu.

Menurut Wikipedia Siswa merupakan anggota masyarakat yang berusaha meningkatkan potensi diri melalui proses pembelajaran pada jalur pendidikan baik pendidikan formal maupun nonformal, pada jenjang pendidikan dan jenis pendidikan tertentu.

(<https://www.dosenpendidikan.co.id/pengertian-siswa-menurut-para-ahli>)  
dikutip tanggal 27/04/2021 jam 17.00 )

### **2.3 Guru**

Menurut kamus besar bahasa Indonesia Guru adalah orang yang pekerjaannya (mata pencahariannya, profesinya) mengajar. Dengan demikian guru adalah orang yang bisa menyampaikan pengetahuannya kepada orang lain (murid atau peserta didik).

Guru-guru seperti ini harus mempunyai semacam kualifikasi formal. Dalam definisi yang lebih luas, setiap orang yang mengajarkan suatu hal yang baru dapat juga dianggap seorang guru. Beberapa istilah yang juga menggambarkan peran guru, antara lain : Dosen, Mentor, Tentor, Tutor

Guru adalah pendidik dan pengajar pada pendidikan anak usia dini jalur sekolah atau pendidikan formal, pendidikan dasar, dan pendidikan menengah.  
([http://id.wikipedia.org/wiki/Guru#Arti\\_umum](http://id.wikipedia.org/wiki/Guru#Arti_umum)).(diakses tanggal 26 / 03 / 2021 .  
18.30)



## 2.4 Kepala Sekolah

Menurut kamus besar bahasa Indonesia kepala sekolah yang terdiri dari dua kata yaitu “kepala” dan “sekolah”, kata “kepala” dapat diartikan “ketua” atau “pemimpin” dalam suatu organisasi atau lembaga. Sedang sekolah adalah sebuah lembaga di mana menjadi tempat menerima dan memberi pelajaran.

Dalam organisasi kata pemimpin, mengandung konotasi: “menggerakkan, mengarahkan membimbing, melindungi, membina, memberikan teladan, memberikan dorongan, memberikan bantuan, dan masih banyak lagi tentang pengertian pemimpin”.

Keberhasilan suatu lembaga pendidikan sangat tergantung pada kepemimpinan kepala sekolah. Karena kepala sekolah sebagai pemimpin di lembaganya, maka dia harus mampu membawa lembaganya kearah tercapainya tujuan yang telah ditetapkan. kepala sekolah harus mampu melihat adanya perubahan serta mampu melihat masa depan dalam kehidupan globalisasi yang lebih baik. Kepala sekolah harus bertanggung jawab atas kelancaran dan keberhasilan semua urusan pengaturan dan pengelolaan secara formal kepada atasannya atau informal kepada masyarakat yang telah menitipkan anak didiknya.

Kepala sekolah sebagai penentu kebijakan di sekolah mempunyai peran penting sehingga mampu memimpin sekolah dengan bijak dan terarah demi meningkatkan kualitas dan mutu pendidikan di sekolahnya, sehingga kualitas lulusan anak didiknya dapat membanggakan dan menyiapkan masa depan yang lebih baik bagi dirinya sendiri dan bagi lingkungannya.

Kepala sekolah harus mempunyai wawasan, keahlian managerial, mempunyai karisma kepemimpinan dan juga pengetahuan yang luas tentang tugas dan fungsi sebagai kepala sekolah. Dengan kemampuan yang dimiliki seperti itu, kepala sekolah tentu saja akan mampu mengantarkan dan membimbing segala komponen yang ada di sekolahnya dengan baik dan efektif menuju ke arah cita-cita sekolah.

Menurut Mulyono dalam bukunya Manajemen Administrasi dan Organisasi Pendidikan bahwa kepemimpinan merupakan ruh yang menjadi pusat sumber gerak organisasi untuk mencapai tujuan. Kepemimpinan yang berkaitan dengan kepala sekolah dalam meningkatkan kesempatan untuk mengadakan pertemuan secara efektif dengan para guru dalam situasi yang kondusif. Perilaku kepala sekolah harus mendorong kinerja para guru dengan menunjukkan rasa bersahabat, dekat, dan penuh pertimbangan terhadap para guru, baik sebagai individu maupun secara kelompok. Perilaku instrumental merupakan tugas-tugas yang diorientasikan dan secara langsung diklarifikasi dalam peranan.

(dikutip <http://www.jejakpendidikan.com/2017/01/pengertian-kepala-sekolah.Html>) (diakses tanggal 28/04/2021 18.30)

## **2.5 Aplikasi**

Menurut Jogiyanto Aplikasi merupakan penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

Menurut Kamus Kamus Besar Bahasa Indonesia (KBBI) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan

atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari user (pengguna).

Menurut Wikipedia Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Menurut Rachmad Hakim S aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (game) dan sebagainya.

Menurut Harip Santoso aplikasi adalah suatu kelompok file (form, class, report) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset.

Menurut Ibis aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya.

Menurut Hengky W. Pramana aplikasi merupakan suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia (Yosua P.W Simaremare, Apol Pribadi S dan Radityo Prasetianto Wibowo”Perancangan dan Pembuatan Aplikasi Manajemen Publikasi Ilmiah Berbasis Online pada Jurnal SISFO)

## **2. 6 Perangkat keras (Hardware)**

Perangkat keras merupakan perangkat yang dapat dilihat dan disentuh secara langsung oleh manusia yang berfungsi untuk mendukung proses komputerisasi. Perangkat keras bagi suatu *sistem* terdiri atas seluruh bagian fisik komputer (pusat pengolah, unit masukan/keluaran), peralatan penyimpanan data, dan terminal masukan/keluaran.

## 2.7 Perangkat lunak (Software)

Sedangkan perangkat lunak (*software*) tidak dapat disentuh dan dilihat secara fisik, *software* memang tidak berwujud benda tapi kita bisa mengoperasikannya. Perangkat lunak dapat dibagi dalam 3 jenis utama :

1. Bahasa pemrograman : merupakan perangkat lunak yang bertugas mengkonversikan arsitektur dan algoritma. contoh : *Basic, Cobal, Pascal, C++, Fortran, Java,*
2. Sistem Operasi : saat komputer pertama kali di hidupkan, sistem operasilah yang pertama kali di jalankan, sistem operasi yang mengatur seluruh proses. contoh : *DOS, Unix, Windows 95, 98, 2000, Xp, Vista , IMB OS/2, Apple's System 7*
3. Utility : sistem operasi merupakan perangkat lunak sistem dengan fungsi tertentu, misalnya pemeriksaan perangkat keras (*hardware troubleshooting*), memeriksa disket yang rusak (bukan rusak fisik), mengatur ulang isi harddisk (*partisi, defrag*), contoh *Utility* adalah *Norton Utility*.

## 2.8 Daring

Menurut KBBI Kemendikbud, Daring adalah akronim dari dalam jaringan. Artinya terhubung melalui jaringan komputer, melalui *internet*. Merinci kegiatan-kegiatan *daring* di antaranya, *webinar*, kelas *online*, KKN *online*, hingga kuliah *online*. Seluruh kegiatan dilakukan menggunakan jaringan *internet* dan komputer.

## 2.9 Google (Google Form Dan Google Sheets)

*Google form* atau *google formulir* adalah sebuah aplikasi yang disediakan oleh *google* untuk pekerjaan administratif survei yang tergabung pada aplikasi *google drive* seperti *google doc*, *spreadsheet*, dan *slide*

*Google Forms* adalah alat yang memungkinkan mengumpulkan informasi dari pengguna melalui survei atau kuis yang dipersonalisasi. Informasi tersebut kemudian dikumpulkan dan secara otomatis terhubung ke *spreadsheet*.

*Spreadsheet* adalah sebuah program komputer yang digunakan untuk menyimpan, menampilkan, serta mengolah data dalam bentuk baris dan kolom. Baris biasanya menggunakan label angka 1,2,3 dan seterusnya. Sedangkan kolom menggunakan label abjad seperti A, B, C, dan seterusnya.

Pengolahan data dalam *spreadsheet* disimpan dalam sebuah *sel*. Pemberian nama *sel* tersebut sesuai dengan penggunaan label pada kolom dan baris yang digunakan, contohnya A1, A2, A3 dan seterusnya.

Definisi dari *spreadsheet* sendiri merupakan tabel-tabel dalam komputer yang berisi baris dan kolom yang dapat digunakan untuk mengolah serta mengatur sebuah data.

Pengolah angka merupakan arti dari kata *spreadsheet* itu sendiri, karena *spreadsheet* memang berfungsi untuk mengolah angka-angka yang terdapat di dalam *sel*. *Sel* dapat berisi nilai numerik, rumus, maupun teks alfanumerik.

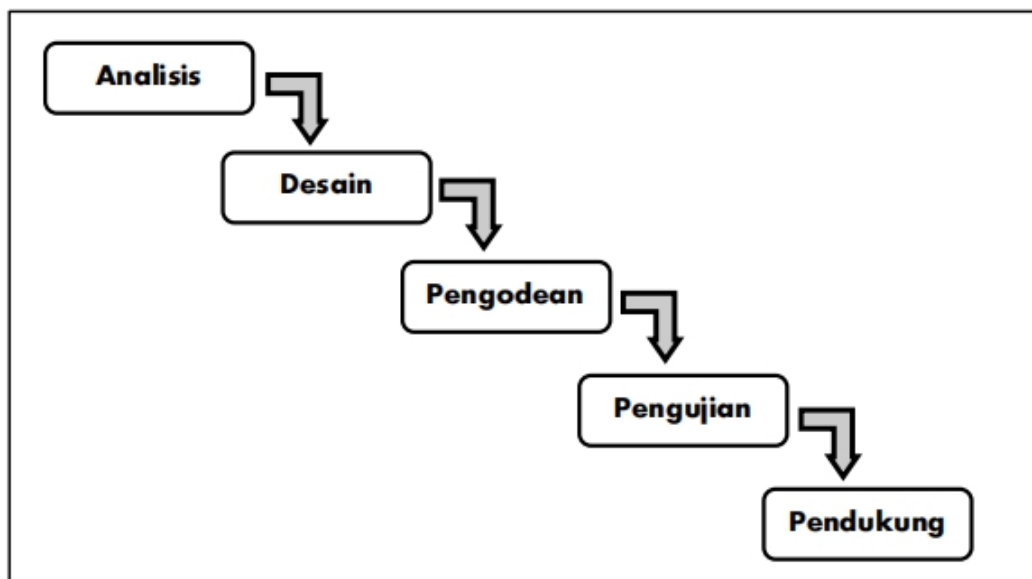
## 2. 10 Waterfall ( Metode Air Terjun )

Pada awal pengembangan perangkat lunak, para pembuat program (*programmer*) langsung melakukan *pengkodean* perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak.

*Software Development Life Cycle* atau *System Development Life Cycle (SDLC)* dimulai dari tahun 1960-an, untuk 23 mengembangkan sistem skala besar secara fungsional untuk para konglomerat pada zaman itu. Sistem-sistem yang dibangun mengelola informasi kegiatan dan *rutinitas* dari perusahaan-perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya.

SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan *metodologi* yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Salah satu model yang dimiliki SDLC adalah Model SDLC air terjun (*Waterfall*) yang sering disebut juga model *sekuensial linier (sequential linier)* atau alur hidup klasik (*Classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara *sekuensial* atau berurutan dimulai dari *analisis, desain, pengkodean, pengujian, dan tahap pendukung (support)*. Tahapan model air terjun (*waterfall*) dapat dilihat pada Gambar 2.1 dibawah ini.



Gambar 2. 1 Tahapan Model Air Terjun (*Waterfall*)

(R.A. Sukamto & Salahudin , diakses tanggal 27/03/2021.18.30)

Berikut ini adalah penjelasan dari tahapan-tahapan yang dilakukan dalam Model air terjun (*Waterfall*):

#### 1. Analisis

Analisis kebutuhan perangkat lunak merupakan proses pengumpulan kebutuhan dilakukan secara *intensif* untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami. Perangkat lunak seperti apa yang dibutuhkan oleh pengguna (*user*). Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

#### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang *focus* pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain ini juga perlu didokumentasikan.

### 3. Pengodean

Pada tahapan ini adalah pembuatan kode program yang mana desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 4. Pengujian

Pengujian fokus pada perangkat lunak secara logis dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalkan kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung (*Support*)

Tahap pendukung (*support*) atau pemeliharaan (*maintenance*) ini diperlukan karena tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke pengguna (*user*).



Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari *analisis spesifikasi* untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab berikut:




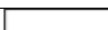




- a. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pembangunan.
- b. Sangat sulit bagi pengguna (*user*) untuk mendefinisikan semua di awal alur pembangunan. pengguna (*user*) sering kali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
- c. Pengguna (*user*) tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Model air terjun sangat cocok digunakan kebutuhan pengguna (*user*) yang sudah sangat paham dengan metode ini sehingga kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun adalah struktur tahap pengembangan *sistem* jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan sehingga tidak ada tumpang tindih pelaksanaan tahapan.

## 2.11 Flowmap

*Flowmap* adalah campuran peta dan *flow chart*, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. *Flowmap* menolong analisis dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoprasian. Kegunaan dari *flowmap* adalah sebagai berikut :

1. Menggambarkan aktivitas apa saja yang sedang berjalan.
2. Menjabarkan aliran dokumen yang terlihat.
3. Menjelaskan hubungan-hubungan data dan informasi dengan bagian bagian dalam aktivitas.
4. Mendefinisikan hubungan antara bagian (pelaku proses), proses manual atau berbasis komputer.
5. Aliran data (dalam bentuk dokumen keluaran dan masukan)




NO.	SIMBOL	NAMA	KETERANGAN
1.		Terminator	Awal dan akhir <i>Flowmap</i> .
2.		Dokumen	Menunjukkan dokumen <i>input</i> atau <i>output</i> untuk proses manual atau komputer.
3.		Proses manual	Kegiatan proses yang dilakukan dengan manual.
4.		Proses komputer	Kegiatan proses yang dilakukan dengan komputerisasi.
5.		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
6.		Garis alir	Menunjukkan alir data dari atau ke proses.
7.		Data storage	Menunjukkan penyimpanan arsip atau dokumen non komputer.
8.		Arsip	Menunjukkan penyimpanan data yang dilakukan secara manual.

Gambar 2. 2 Tabel Simbol *Flowmap*

(*R.A.Sukamto & Shalahudin, diakses tanggal 26/4/2021. 18.30*)

## 2. 12 Diagram Konteks

Diagram konteks digunakan sebagai diagram guna merancang fungsi-fungsi yang dijalankan dalam perangkat lunak nantinya. Oleh karena itu, akan sangat penting adanya kesesuaian antara perancangan kode dengan program yang akan dibuat. Sehingga disaat perangkat lunak sudah tidak ditangani pengembangan pertamanya maka perangkat lunak tetap dapat dikembangkan karena kesesuaian kode program (*R.A.Sukamto & Shalahudin, diakses tanggal 26/4/2021. 18.30*).

NO.	SIMBOL	NAMA	KETERANGAN
1.		Entitas	Menunjukkan asal atau tujuan suatu data atau informasi.
2.		Proses	Pemrosesan pengolahan suatu data.
3.		Arus Informasi	Menggambarkan aliran data yang berjalan.

Gambar 2. 3 Tabel Simbol Diagram Konteks

(R. A. Sukamto & Shalahudin, diakses tanggal 26/4/2021. 18.30)





### 2. 13 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan test-test pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada *metodelogi* tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (R.A. Sukamto & Shalahudin, diakses tanggal 26/4/2021. 18.30).

### 2. 14 *Use Case Diagram*

*Use Case Diagram* adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian aktor dan *use case* (Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.00).

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau faktor.

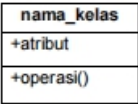






NO	SIMBOL	NAMA SIMBOL	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2.	<<include>>	<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
3.	<<extend>>	<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4.		<i>Association</i>	Apa yang menghubungkan antar objek satu dengan objek lainnya.
5.		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Gambar 2. 4 Tabel Simbol Use Case Diagram

(Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.30)

## 2.15 Class Diagram

*Class Diagram* atau diagram *class* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas (Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.30).


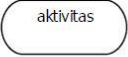



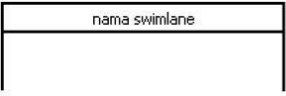


Simbol	Deskripsi
Kelas 	kelas pada struktur sistem
Antarmuka / <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
kebergantungan / <i>dependency</i> 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	relasi antar kelas dengan makna semuabagian ( <i>whole-part</i> )

Gambar 2.5 Tabel Simbol Class Diagram

(Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.45)

## 2.16 Activity Diagram

Diagram activity atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.45).

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / <i>decision</i> 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / <i>join</i> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
<i>fork</i> , 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
<i>join</i> , 	digunakan utk menunjukkan kegiatan yg digabungkan

Gambar 2. 6 Tabel Simbol *Activity Diagram*

(Sukamto & Shalahudin, diakses tanggal 26/4/2021. 19.45)

## 2.17 Basis Data (Data base)

Basis data digunakan sebagai media penyimpanan data dan tempat pengolahan data menjadi informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang *terintegrasi*.

Menurut Sukamto dan Shalahuddin (2019:43), “sistem basis data” adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Sedangkan, menurut Lubis (2016:2) mengemukakan bahwa “basis data” merupakan gabungan file data yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat *independen*”.

Begitu juga menurut Yanto (2016:11) mengemukakan bahwa basis data merupakan “himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah”. Dapat ditarik kesimpulan bahwa basis data merupakan sekumpulan data yang tersimpan di tempat penyimpanan data dan dapat diolah menjadi informasi serta digunakan kembali jika dibutuhkan kembali.

Penulis menggunakan basis data dengan bahasa *Structured Query Language* (SQL) dan MySQL sebagai aplikasi pengolah basis data. *Structured Query Language* (SQL) adalah bahasa pemrograman ini sering digunakan karena kompatibel dan dapat berlaku hampir semua media pengolah basis data.

Menurut Sukamto dan Shalahuddin (2015:46) “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”.

Berdasarkan kutipan dari para ahli di atas dapat disimpulkan bahwa



*Structured Query Language* (SQL) merupakan suatu bahasa pemrograman yang berguna untuk mengelola basis data

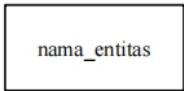
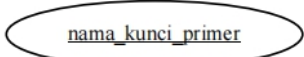
## 2.18 *Entity Relationship Diagram (ERD)*

Menurut Rosa (2018) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, bisa berupa *file* teks ataupun *Database Management System* (DBMS).

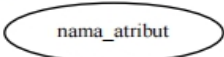
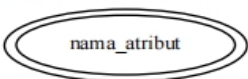
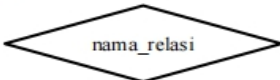

Kebutuhan basis data dalam sistem informasi meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah tersimpan. Tujuan dari dibuatnya tabel-tabel adalah untuk menyimpan data ke dalam tabel-tabel agar mudah di akses. Oleh karena itu untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom. *Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang paling banyak digunakan dan dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *Object Oriented DBMS* (OODBMS) maka perancangan tidak perlu menggunakan ERD.

ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis),

notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen, Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen dapat dilihat pada Tabel dibawah.

Simbol	Deskripsi
Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)

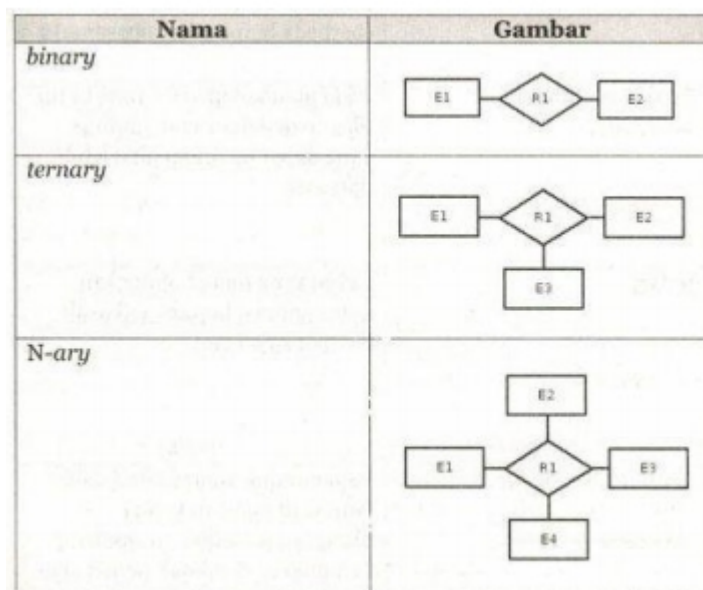
Gambar 2. 7 Tabel Simbol-simbol ERD notasi Chen

Simbol	Deskripsi
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut multivalai/ <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/ <i>association</i> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.

Gambar 2. 8 Tabel Simbol-simbol ERD notasi Chen

*Sukanto & Shalahudin, diakses tanggal 26/4/2021. 19.45*

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah *entitas*). Beberapa metode perancangan ERD menoleransi hubungan *relasi ternary* (satu relasi menghubungkan tiga buah relasi) atau N-ary (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD tidak mengizinkan hubungan ternary atau N-ary. Contoh dari bentuk hubungan relasi dalam ERD dapat dilihat pada Gambar. dibawah ini.



Gambar 2. 9 Bentuk hubungan relasi ERD

(Rosa dan Shalahuddin, 2019:52)

## 2. 19 *Black Box Testing*

Pengujian sistem merupakan pengujian program perangkat lunak yang lengkap dan terintegrasi. Perangkat lunak atau yang lebih 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.

Pengujian terhadap perangkat lunak dapat dibedakan menjadi dua yaitu *Black Box Testing* dan *White Box Testing*.

a. *Black Box Testing*.

*Black Box Testing* atau yang lebih sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program.

Pada *Black Box Testing* ini dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji *source code program*. *Black-box testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi *domain*.

Tes “Kotak Hitam atau *Black Box*” hanya terdiri dari meninjau fungsi-fungsi dari aplikasi, yaitu jika melakukan apa yang seharusnya, tidak peduli bagaimana melakukannya. Struktur dan fungsi internalnya tidak dipelajari. Dengan demikian penguji perlu mengetahui apa peran sistem itu, dan fungsinya, tetapi tidak tahu mekanisme internalnya. Dia memiliki profil “pengguna”.

Jadi metode ini bertujuan untuk memeriksa, setelah tahap akhir proyek, apakah perangkat lunak atau aplikasi berfungsi dengan baik, dan melayani

penggunaanya secara efisien. Biasanya, penguji mencari fungsi yang hilang atau salah; antarmuka, kinerja, *inisialisasi* program dan kesalahan keluar; struktur data atau kesalahan akses basis data eksternal.

Untuk ini, mereka menyiapkan skenario berdasarkan jalur pengguna yang berbeda yang dapat diikuti pada sistem yang akan diuji. Semua fungsi harus diperhitungkan, agar semuanya telah diuji pada akhir operasi. Setiap tes mengikuti skenario, memeriksa untuk setiap fungsi, jika *input* yang *valid* diterima, yang tidak valid ditolak, dan tentu saja, setiap waktu, jika output atau hasil yang diharapkan memang diperoleh. Itu adalah metode “coba-coba”.

Jadi untuk melakukan *Black Box Testing* seseorang tidak harus memiliki kemampuan *coding* atau *TDD coding*, Tes ini bisa dilakukan oleh pengguna, tim marketing bahkan *user* sebagai kuisisionernya.

Teknik-teknik *Black-box Testing*:

1. *Equivalence Partitioning*
2. Cara kerja teknik ini adalah dengan melakukan *partition* atau pembagian menjadi beberapa partisi dari input data.
3. *Boundary Value Analysis*
4. Teknik ini lebih fokus kepada *boundary*, adakah *error* dari luar atau sisi dalam *software*, minimum, maupun maksimum nilai dari *error* yang didapat.
5. *Fuzzing*

*Fuzz* merupakan teknik untuk mencari *bug* atau gangguan

dari *software* dengan menggunakan injeksi data yang terbilang cacat ataupun sesi semi-otomatis.

6. *Cause-Effect Graph*

7. Ini adalah teknik testing dimana menggunakan *graphic* sebagai acuannya. Dimana dalam grafik ini menggambarkan relasi antara efek dan penyebab dari *error* tersebut.

8. *Orthogonal Array Testing*

9. Dapat digunakan jika input domain yang relatif terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.

10. *All Pair Testing*

11. Dalam teknik ini, semua pasangan dari *test case* di desain sedemikian rupa agar dapat dieksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar input parameternya. Tujuannya *testing* ini adalah memiliki pasangan *test case* yang mencakup semua pasangan tersebut.

12. *State Transition*

13. *Testing* ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dari UI dalam bentuk grafik.

## 2. 20 *Global Positioning System (GPS)*

*Global Positioning System* (GPS) merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Cara kerja GPS secara logis ada 5 langkah:

1. Memakai perhitungan "*triangulation*" dari satelit.
2. Untuk perhitungan "*triangulation*", GPS mengukur jarak menggunakan *travel time* sinyal radio.
3. Untuk mengukur *travel time*, GPS memerlukan memerlukan akurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya.
5. Terakhir harus mengkoreksi *delay signal* waktu perjalanan di atmosfer sampai diterima *receiver*.

Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS. Sebuah GPS *receiver* harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Jika GPS *receiver* dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude, longitude* dan *altitude*).

Jika sudah dapat menentukan posisi *user*, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan lain lain. Sinyal yang dikirimkan oleh satelit ke GPS akan digunakan untuk menghitung waktu perjalanan (*travel time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival* (TOA). Sesuai dengan prinsip fisika, bahwa untuk mengukur

jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal. Dari beberapa pemakaian GPS di atas dikategorikan menjadi:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi:
  - 1) *Latitude*
  - 2) *Longitude*
  - 3) *Altitude*
3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
5. Komulasi data. GPS *receiver* dapat menyimpan informasi *track*, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya.
6. *Tracking*. Membantu untuk memonitoring pergerakan objek. Membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat.



## **BAB III**

### **ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

Langkah-langkah penelitian ini merujuk pada metode pengembangan perangkat lunak, yaitu Metode Waterfall. Berikut adalah langkah-langkah pada Metode Waterfall:

#### **3.1 Analisis**

Pada tahapan ini dilakukan proses pengumpulan kebutuhan data untuk menspesifikasikan kebutuhan perangkat lunak. Proses pengumpulan data yang dilakukan antara lain:

1. Analisis data
2. Analisis sistem yang berjalan
3. Analisis masalah
4. Analisis sistem yang akan dibangun
5. Analisis kebutuhan sistem

##### **3.1.1 Analisis Data**

Profil singkat SMK YADIKA Tanjungsari Sumedang merupakan Sekolah Menengah Kejuruan yang berlokasi di Jl. Raya Tanjungsari no

394.A Rt 03/02 Desa Gudang Kecamatan Tanjungsari Kab. Sumedang .  
SMK YADIKA TANJUNGSARI SUMEDANG adalah salah satu unit dari Bidang Pendidikan Yadika yang ke 25. Mulai beroperasi sejak tahun pelajaran 2010/2011 dengan membuka 3 buah jurusan yaitu Jurusan Teknik Komputer Jaringan, Teknik Kendaraan Ringan, dan juga Otomatisasi Tata Kelola Perkantoran (OTKP).

Proses analisis data dilakukan dengan cara melakukan proses wawancara secara langsung kepada guru, wali kelas dan kepala sekolah SMK Yadika Tanjungsari.

Pengumpulan data bertujuan untuk mengumpulkan data-data mengenai sistem absensi yang sering di gunakan sekolah SMK Yadika Tanjungsari. Tujuannya adalah mengumpulkan data-data terkait absensi dan bagaimana proses atau alur sebuah sistem absensi.

Hasil komunikasi tersebut didapatkan data bahwa dikarenakan situasi dan kondisi Covid-19 yang membatasi setiap orang berkerumun maka sistem absensi menggunakan *id card*, *finger print* atau secara manual menjadi sangat tidak efektif. Saat ini secara keseluruhan semua menggunakan *google form* sebagai sistem absensi yang sering digunakan saat ini sebagai sarana absensi secara *daring* (Dalam Jaringan) atau online menggunakan *internet*. *Google form* akan mengumpulkan data dengan baik dan tentunya real time.

Beberapa di antaranya adalah memudahkan Anda untuk membuat sekaligus mengumpulkan sebuah data. Anda tidak perlu meluangkan

banyak tenaga untuk memberikan pertanyaan satu persatu pada yang bersangkutan, apalagi sampai mencetak pertanyaan di kertas demi *absensi* atau *quisioner*. Teknologi telah memudahkan semuanya. Anda hanya perlu membagikan sebuah *link* dari *google form* yang telah dibuat, kemudian biarkan orang mengisi absensi atau *kuisisioner* secara *online*. Dengan demikian, Anda tidak perlu membuang *media* berupa kertas, tinta, buang waktu, dan juga tenaga.

Peneliti menggunakan *model spiral* dan melakukan berbagai tahapan-tahapan yang terdapat pada teori rekayasa perangkat lunak dalam mengembangkan *sistem absensi* online di SMK Yadika antara lain, *communication, planning (estimation, scheduling), modelling (design), construction (code and test)* serta model spiral memiliki beberapa *cycle* dalam pengembangannya.

Pada tahap ini peneliti melakukan komunikasi dengan pihak terkait, seperti siswa, Guru, Wali kelas dan Akademik. Peneliti melakukan wawancara untuk mendapatkan *User Requirement* sebagai salah satu bahan untuk mengembangkan sistem. Proses pengambilan *user requirement* dilakukan pada setiap *cycle* untuk mendeteksi apakah adanya kebutuhan baru atau kesalahan di *cycle* sebelumnya.

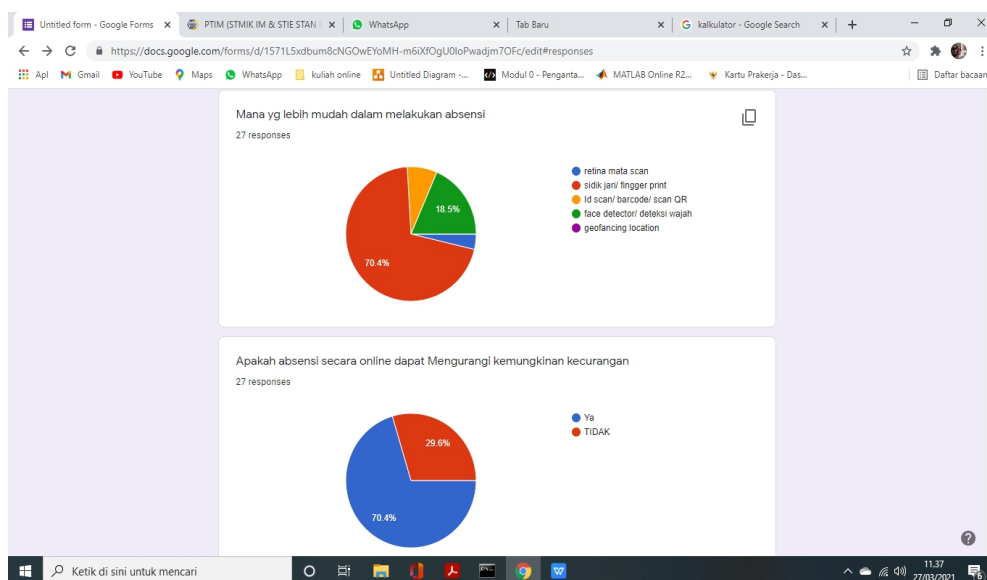
#### 1. *Cycle Pertama*

Pada tahap *cycle* pertama, peneliti memfokuskan untuk mengetahui alur sistem yang akan dibuat. *User requirement* diberikan kepada siswa yang pernah menjadi Penanggung Jawab absensi Kelas, guru dan

Akademik. Responden terdiri dari 27 orang siswa, Fasilitas yang diharapkan responden terhadap absensi online untuk siswa/mahasiswa dapat dilihat pada tabel 3.1. dibawah ini

Tabel 3. 1 Data *User Requirement Cycle* Pertama Pada Responden

No.	Fasilitas yang Diharapkan	Jumlah Responden
1	Pentingnya Pengisian absensi bagi siswa/mahasiswa	27
2	Proses absensi secara online terbilang efektif dan efisien	23
3	Mengurangi penggunaan kertas (paperless)	26
4	Apakah penting pengisian form nama atau nim bagi siswa/mahasiswa	27
5	Sistem absensi online tergolong sangat cepat tanpa perlu menulis nama dan noreg lagi	25
6	Mana yg lebih mudah dalam melakukan absensi	
7	Apakah absensi secara online dapat Mengurangi kemungkinan kecurangan	19



Gambar 3. 1 Diagram hasil User Requirement siswa- Cycle Pertama

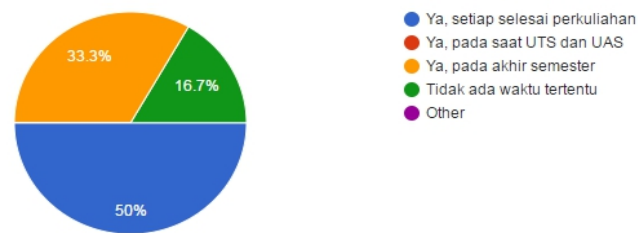
Dari hasil responden 70,4% atau 19 orang mengatakan bahwa finger print sangat cepat dan efektif untuk melakukan absensi. Sistem pencatatan kehadiran yang aman dan efisien sejauh ini banyak yang menggunakan *fingerprint*. Namun sistem ini membutuhkan biaya yang tidak murah dikarenakan harga alat yang lumayan mahal serta sistem ini tidak dapat dipindah-pindah sehingga setiap ruang kelas harus memiliki alat tersebut. Dengan memanfaatkan teknologi perangkat bergerak dan internet, *diimplementasikan* sebuah aplikasi sistem kehadiran siswa dapat melakukan *absensi/presensi* pada *smartphone* masing-masing.

Responden dari *User Requirement* untuk Guru dan Wali kelas berjumlah 16 orang. Peran guru pada sistem absensi siswa merupakan sebagai pemantau absensi yang nantinya akan berpengaruh pada nilai siswa tersebut. Alur guru dalam memantau proses absensi siswa/mahasiswa terdapat pada tabel 3.2. Dibawah ini

Tabel 3. 2 Data User Requirement Cycle Pertama pada Guru

No	Cara guru/dosen memantau absensi siswa/mahasiswa	Jumlah
1	Memantau melalui penanggung jawab kelas	2
2	Menghitung jumlah mahasiswa yang ada di ruang kelas	14
3	Memanggil nama mahasiswa satu persatu	6
4	Memantau melalui aplikasi online	10

Selain mengetahui bagaimana guru, wali kelas, kepala sekolah memantau absensi, peneliti juga menanyakan seberapa sering guru/dosen memantau daftar hadir siswa/mahasiswa dan didapat hasil sebagai berikut:



Gambar 3. 2 Grafik hasil *User Requirement* guru-Cycle Pertama

Dari informasi tersebut sebagian besar guru memilih untuk memantau melalui kelas dan menghitung jumlah mahasiswa yang berada di kelas serta memanggil nama mahasiswa satu persatu jika cukup waktu. Maka dari itu, sistem absensi online akan menerapkan alur di mana guru dapat memantau proses absensi sendiri dan secara *mobile*. Guru juga dapat memvalidasi apabila terjadi kecurangan pada proses absensi.

Admin yang menjadi narasumber dalam *User Requirement* kali ini berjumlah satu orang yang bertugas sebagai admin *Absensi* yang merekap absen untuk para guru atau untuk arsip data. Admin hanya membuat jadwal yang telah ditentukan dan mengatur kelas. Admin mempunyai peran dalam mengatur jumlah siswa dalam satu kelas.

Jika terdapat kelas yang melebihi kapasitas maka admin berhak membuka kelas baru dan mengatur daftar siswa untuk kelas tersebut.

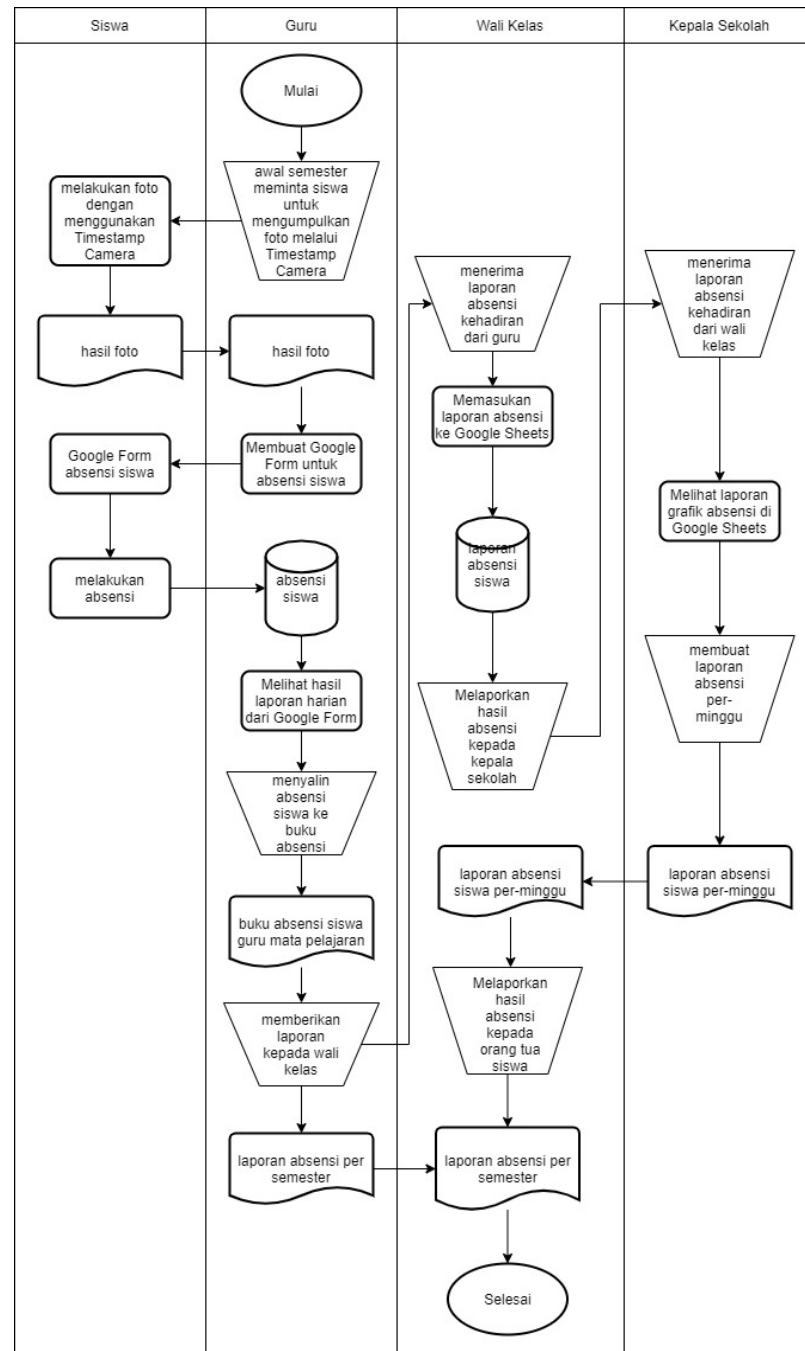
Maka dari itu, peneliti membuat suatu hubungan antara admin absensi dengan sistem absensi *online* yang peneliti kembangkan. Hubungannya antara lain, daftar siswa setiap kelas pada admin absensi akan menjadi *input* daftar absen yang ada pada sistem *absensi online*. Selengkapnya dalam bagan pada tahapan perencanaan. Setelah memperoleh informasi untuk input dan output sistem, peneliti pada *cycle* pertama ini akan membuat tampilan atau *mockup* mengikuti alur yang telah diberikan admin lalu akan diteruskan pada tahap *cycle* dua.

## 2. *Cycle Kedua*

Terdapat beberapa fitur yang ditambahkan untuk melengkapi *cycle* pertama, yaitu adanya fitur bagi kelas berdasarkan nomor registrasi genap atau ganjil pada simulasi sistem admin dan penambahan tanggal di setiap kolom pertemuan untuk absensi siswa. Perbaikan beberapa tampilan juga menjadi masukan bagi peneliti di *cycle* kedua ini. Tujuan dari *cycle* kedua ini ialah memperbaiki beberapa kesalahan di *cycle* pertama dan menyelesaikan fungsionalitas dari sistem.

### 3.1.2 Analisis Sistem yang Berjalan

Proses analisis sistem yang berjalan digambarkan dengan menggunakan diagram *Flowmap*. Berikut ini adalah gambaran sistem yang sedang berjalan untuk proses absensi di SMK Yadika Tanjungsari.



Gambar 3. 3 *Flowmap* Sistem Yang Berjalan di SMK Yadika Tanjungsari



### **3.1.3 Analisis Masalah**

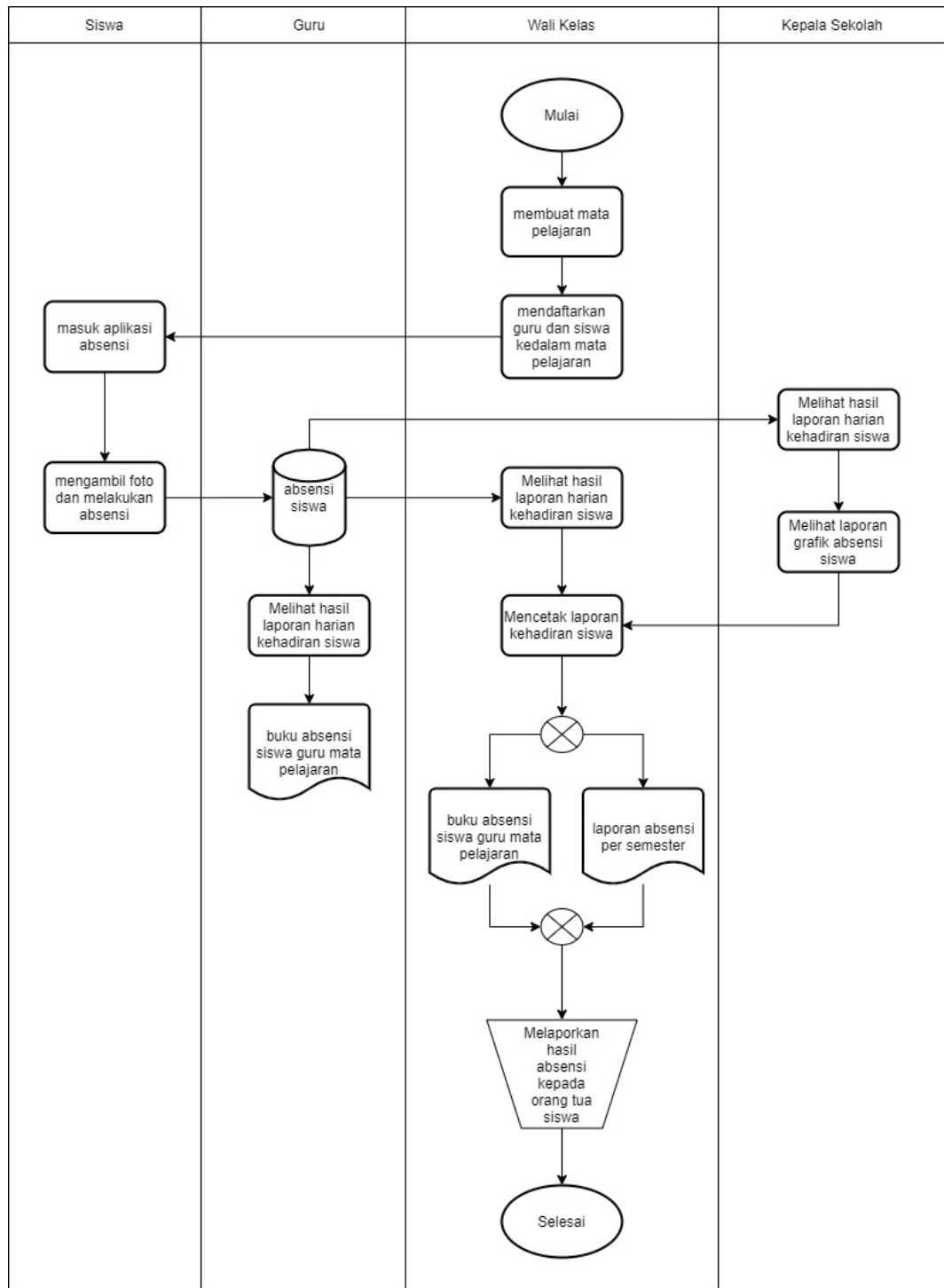
Berdasarkan hasil analisis sistem yang sedang berjalan saat ini pada proses absensi di SMK Yadika Tanjungsari, terdapat beberapa masalah antara lain:

1. Tidak efektifnya proses pelaporan absensi, karena setiap guru harus melakukan rekapitulasi dari Google Form ke buku absensi siswa guru mata pelajaran, kemudian melaporkan hasil absensi siswa kepada wali kelas melalui whatsapp.
2. Wali kelas akan menerima setiap laporan setiap absensi siswa dari guru mata pelajaran melalui aplikasi whatsapp, kemudian wali kelas harus merekapitulasi hasil laporan guru mata pelajaran di aplikasi Google Sheets. Hal ini menyebabkan bertambahnya tugas wali kelas.
3. Kepala sekolah akan melihat grafik kehadiran siswa berdasarkan hasil aplikasi Google Sheets dari wali kelas.
4. Selanjutnya, kepala sekolah akan melakukan penggabungan hasil absensi siswa selama 1 minggu untuk selanjutnya di berikan kepada orang tua siswa sebagai hasil pengawasan keberjalanan kegiatan pembelajaran. Hal ini juga memberikan tambahan tugas kepada kepala sekolah.

### **3.1.4 Analisis Sistem yang Akan Dibangun**

Berdasarkan hasil analisis masalah yang telah di analisis, maka akan dibangun sebuah aplikasi absensi siswa secara daring di SMK Yadika

Tanjungsari. Gambaran sistem aplikasi absensi yang akan di bangun digambarkan dengan menggunakan *Flowmap*.



Gambar 3. 4 *Flowmap* Analisis Sistem Yang Akan Dibangun

## 3.2 Desain

Untuk mendesain perangkat lunak, digunakan model *Unified Modeling Language* (UML) yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya.

### 3.2.1 Deskripsi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi sistem absensi secara daring di SMK Yadika Tanjungsari.

Tabel 3. 2 Deskripsi Aktor

No	Aktor	Deskripsi
1.	Siswa	Seorang siswa atau peserta didik yang melakukan proses belajar di sebuah sekolah SMK Yadika Tanjungsari.
2.	Guru	Seorang tenaga pendidik profesional yang mendidik, mengajarkan suatu ilmu, membimbing, melatih, memberikan penilaian, serta melakukan evaluasi kepada peserta didik baik secara <i>luring</i> ataupun <i>daring</i> , di SMK Yadika Tanjungsari.
3.	Wali kelas	Seorang tenaga pendidik profesional yang mendidik, mengajarkan suatu ilmu, membimbing, melatih, memberikan penilaian, serta melakukan evaluasi kepada peserta didik dan mendapat tugas tambahan sebagai penanggung jawab dinamika pembelajaran di dalam kelas di SMK Yadika Tanjungsari.
4.	Kepala Sekolah	Kepala sekolah adalah pemimpin yang mempunyai peranan sangat besar dalam mengembangkan mutu pendidikan di sekolah SMK Yadika Tanjungsari.

No	Aktor	Deskripsi
5.	Admin	Adalah seorang petugas, yang mempunyai peranan untuk menginput dan mengelola data aplikasi.

### 3.2.2 Deskripsi Use Case

Berikut adalah deskripsi pendefinisian *use case* siswa pada aplikasi sistem absensi secara daring di SMK Yadika Tanjungsari.

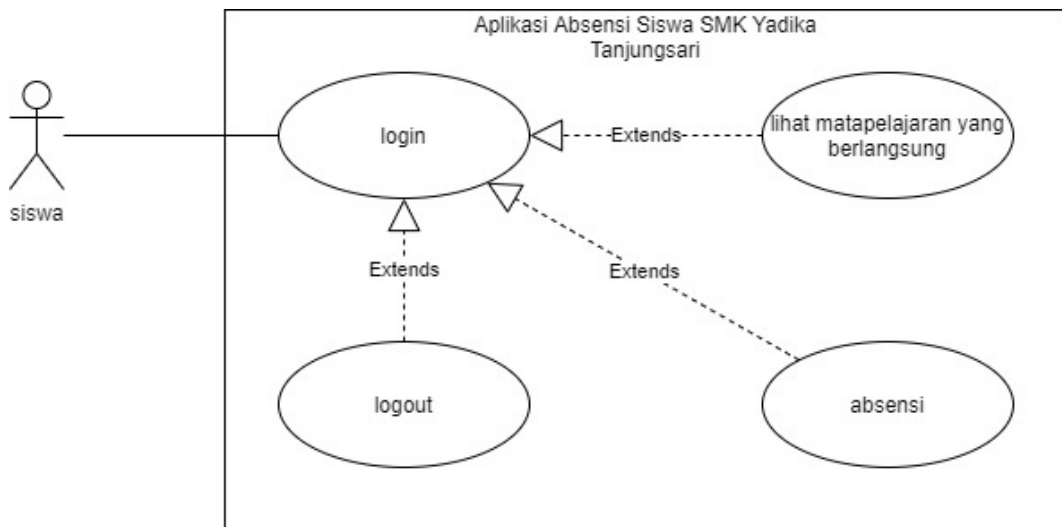
Tabel 3. 3 Deskripsi Use Case

No	Use Case	Deskripsi
1	Login	Merupakan proses <i>verifikasi</i> data di mana pengguna dapat masuk ke sebuah sistem atau aplikasi
2	Logout	Merupakan proses <i>verifikasi</i> data di mana pengguna keluar dari sebuah sistem atau aplikasi yang sedang digunakan
3	Lihat mata pelajaran yang berlangsung	Siswa atau peserta didik melihat tampilan jadwal mata pelajaran yang disediakan setiap harinya
4	Absensi	Siswa melakukan absensi sebagai bukti kehadiran di setiap mata pelajaran berlangsung dengan cara mengambil foto dan menentukan point GPS
5	Login	Merupakan proses verifikasi data dimana pengguna dapat masuk ke sebuah sistem atau aplikasi
6	Logout	Merupakan proses verifikasi data di mana pengguna keluar dari sebuah sistem atau aplikasi yang sedang digunakan
7	Kelola Gedung	Admin melakukan proses untuk menentukan lokasi dan menginputkan data spesifik mengenai lokasi sekolah
8	Kelola staff pengajar	Admin melakukan input data pengajar untuk

No	Use Case	Deskripsi
		pengaturan jadwal guru
9	Kelola Kelas	Admin melakukan input data dan pengaturan jadwal kelas bagi siswa dan tenaga pengajar
10	Kelola Mata Pelajaran	Admin melakukan penginputan data mata pelajaran untuk dilakukan pengaturan jadwal mata pelajaran bagi siswa dan tenaga pendidik
11	Kelola Siswa	Admin melakukan pengaturan jadwal untuk mengolah data siswa
12	Kelola jadwal mata pelajaran	Admin melakukan pengaturan jadwal mata pelajaran bagi siswa dan guru mata pelajaran
13	Report Absensi	Wali kelas, guru, kepala sekolah dapat mengolah data absensi siswa untuk pembuatan pelaporan
14	Kelola absensi	Guru membuat laporan absensi sebagai bentuk laporan kepada wali kelas dan kepala sekolah

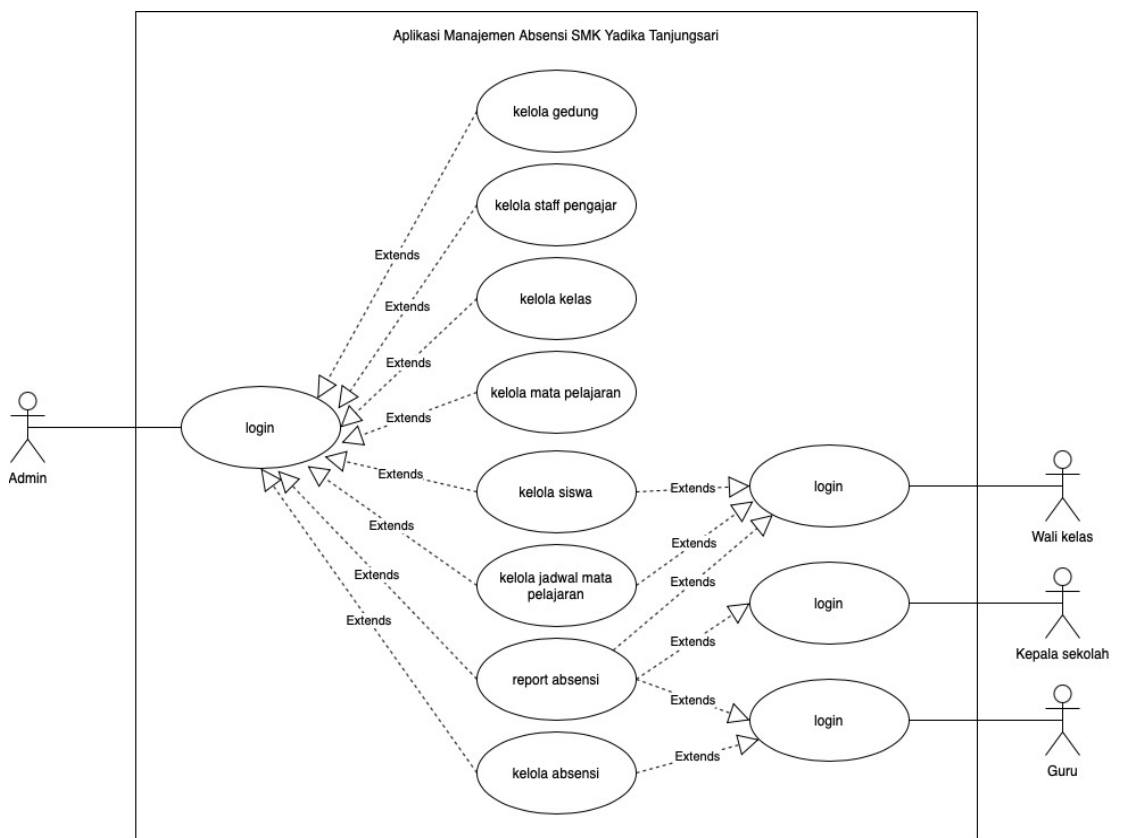
### 3.2.3 Use Case Diagram

*Use case* diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem di mana seorang siswa selaku aktor melakukan *login* untuk dapat masuk ke sebuah aplikasi, setelah melakukan *login* maka siswa dapat melihat mata pelajaran yang berlangsung pada setiap harinya, setelah memilih satu mata pelajaran maka siswa dapat melakukan sistem absensi sebagai bukti kehadiran dan mengikuti mata pelajaran yang berlangsung. setelah semua kegiatan belajar berakhir maka setiap siswa dapat melakukan *logout* untuk keluar dari aplikasi sebagai tanda berakhirnya pelajaran di setiap harinya. Berikut gambaran *use case* diagram siswa



Gambar 3. 5 Use case diagram siswa

Berikut gambaran *Use case* diagram aplikasi manajemen absensi SMK Yadika Tanjung Sari



Gambar 3. 6 Use case diagram Guru Mata Pelajaran, Wali kelas, Kepala Sekolah, dan Admin

### 3.2.4 Skenario Use Case

Alur skenario inilah yang nantinya menjadi landasan pembuatan *sequence* diagram / diagram *sequence* pada sistem absensi secara daring di SMK Yadika Tanjungsari.

Tabel 3. 4 *Scenario Use Case Login*

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik <i>Login Button</i>	
	2. Menampilkan <i>Form Login</i>
3. Memasukan NIS dan <i>Password</i>	
	4. Memeriksa <i>valid</i> atau tidaknya data masukan dengan memeriksa ke <i>table user</i>
	5. Masuk ke aplikasi sistem
<b>Skenario alternatif</b>	
1. Klik <i>Login Button</i>	
	2. Menampilkan <i>Form Login</i>
3. Memasukan NIS dan <i>Password</i>	
	4. Memeriksa <i>valid</i> atau tidaknya data masukan dengan memeriksa ke <i>table user</i>
	5. Menampiklan pesan login tidak valid
6. Memasukan NIS dan <i>Password yang valid</i>	
	7. Memeriksa <i>valid</i> atau tidaknya data masukan dengan memeriksa ke <i>table user</i>
	8. Masuk ke aplikasi sistem

Tabel 3. 5 *Scenario Use Case Logout*

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik <i>Logout Button</i>	
	2. Keluar dari aplikasi sistem

Tabel 3. 6 *Scenario Use Case Absensi*

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Setelah siswa berhasil masuk sistem aplikasi,	
	2. Menampilkan Form jadwal mata pelajaran
3. memilih jadwal mata pelajaran	
	4. Sistem melakukan verifikasi dan menampilkan informasi posisi kelas dan jam mata pelajaran
5. Siswa masuk ke ruang kelas dengan melakukan foto identifikasi dan melakukan cek position	
	6. Sistem menyimpan data siswa sebagai tanda absensi siswa

Tabel 3. 7 *Scenario Use Case Kelola Gedung*

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola gedung	
	2. Menampilkan menu kelola gedung
3. Masukan alamat sekolah, lokasi gedung dan lokasi GPS	
	4. Menyimpan hasil input alamat sekolah, lokasi gedung dan lokasi GPS



Tabel 3. 8 *Scenario Use Case* Kelola Staff

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola staff/ tenaga pengajar	
	2. Menampilkan menu kelola staff/ tenaga pengajar
3. Masukan NIK, nama, status, jabatan	
	4. Menyimpan hasil input NIK, nama, status, jabatan

Tabel 3. 9 *Scenario Use Case* Kelola Kelas

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola kelas	
	2. Menampilkan menu kelola kelas
3. Masukan kode kelas, nama gedung sekolah, status dan nama wali kelas	
	4. Menyimpan Masukan kode kelas, nama gedung sekolah, status dan nama wali kelas

Tabel 3. 10 *Scenario Use Case* Kelola Mata Pelajaran

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola mata pelajaran	
	2. Menampilkan menu kelola mata pelajaran

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Masukan kode mata pelajaran dan nama mata pelajaran	
	4. Menyimpan Masukan kode mata pelajaran dan nama mata pelajaran

Tabel 3. 11 *Scenario Use Case* Kelola Siswa

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola siswa	
	2. Menampilkan menu kelola siswa
3. Masukan NIS, Nama siswa, alamat siswa, kelas, alamat rumah, titik GPS rumah	
	4. Menyimpan Masukan NIS, Nama siswa, alamat siswa, kelas, alamat rumah, titik GPS rumah

Tabel 3. 12 *Scenario Use Case* Kelola Mata Pelajaran

<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik masuk menu kelola jadwal mata pelajaran	
	2. Menampilkan menu kelola jadwal mata pelajaran
3. Masukan kode mata pelajaran dan nama mata pelajaran	
	4. Menyimpan Masukan kode mata pelajaran dan nama mata pelajaran

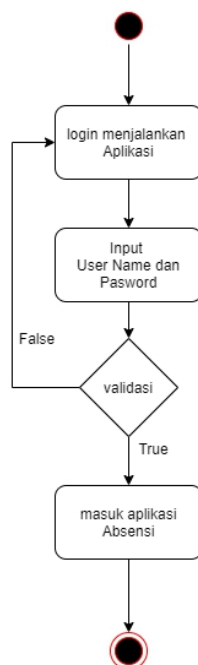
Tabel 3. 13 *Scenario Use Case* Kelola Report Absensi

Aksi Aktor	Reaksi Sistem
1. Klik report absensi	
	2. Menampilkan hasil report absensi

### 3.2.5 *Activity Diagram*

Berikut adalah diagram aktivitas atau *activity* diagram untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari aplikasi sistem absensi secara daring di SMK Yadika Tanjungsari.

1. Diagram *Activity Login* (Siswa, Guru, Wali Kelas, Admin, Kepala Sekolah)



Gambar 3. 7 *Activity Diagram Login* Siswa Guru, Wali Kelas, Admin, Kepala Sekolah

## 2. Diagram *Activity Logout*



Gambar 3. 8 *Activity Diagram Logout*

## 3. Diagram *Activity Melihat Mata Pelajaran Siswa*



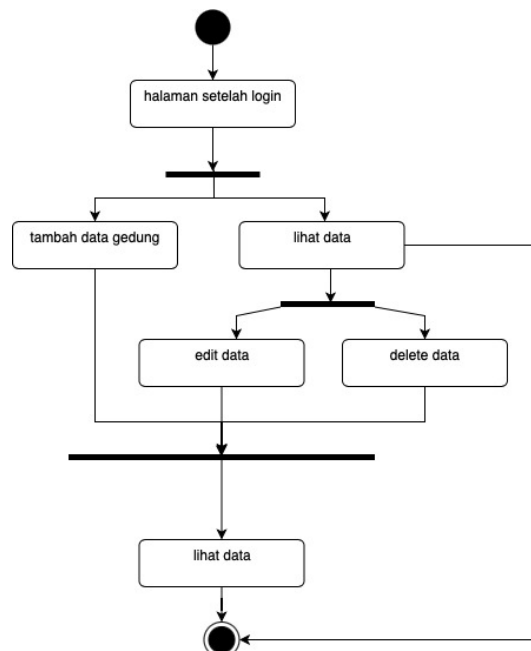
Gambar 3. 9 *Activity Diagram Melihat Mata Pelajaran Siswa*

#### 4. Diagram *Activity* Absensi Siswa



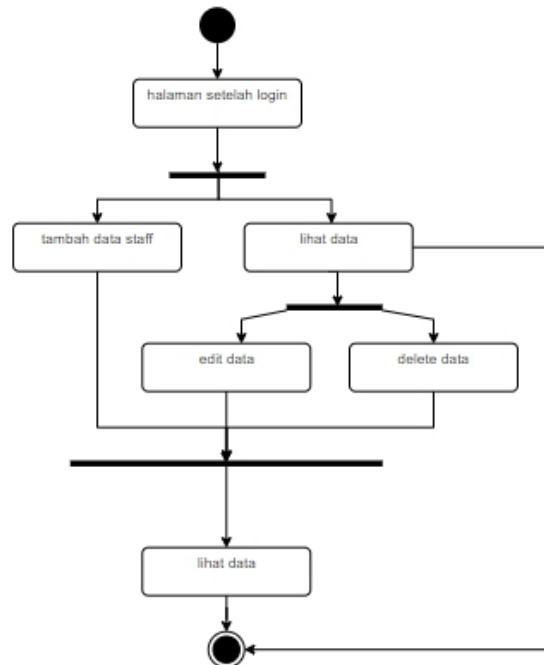
Gambar 3. 10 *Activity* Diagram Absensi Siswa

#### 5. *Diagram Activity* Kelola Gedung (Admin)



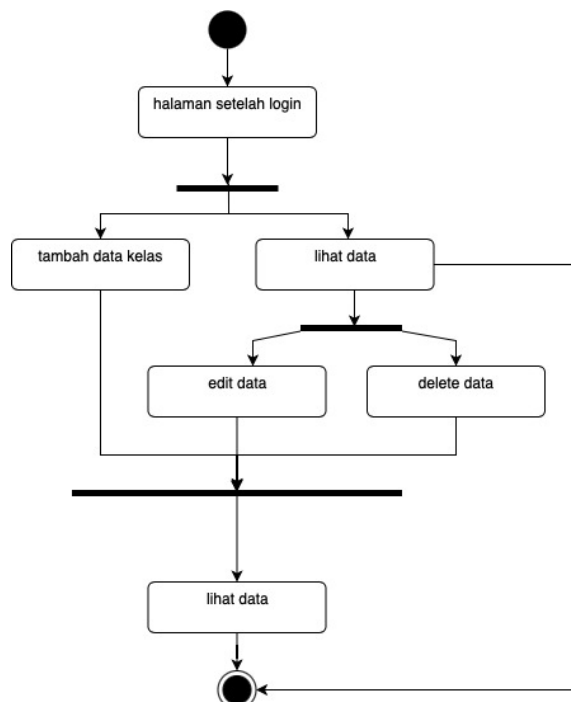
Gambar 3. 11 *Activity* Diagram Admin *Input* Data Gedung

## 6. Diagram Aktivitas Kelola Data Staff (Admin)



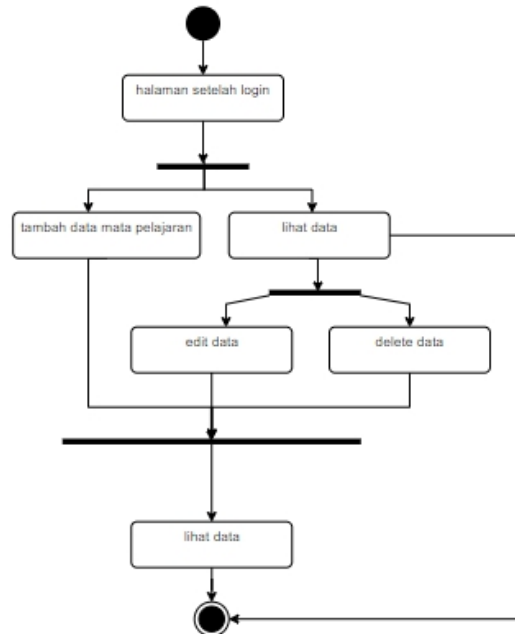
Gambar 3. 12 Activity Diagram Admin Input Data Staff Pengajar

## 7. Diagram Aktivitas Kelola Data Kelas (Admin)



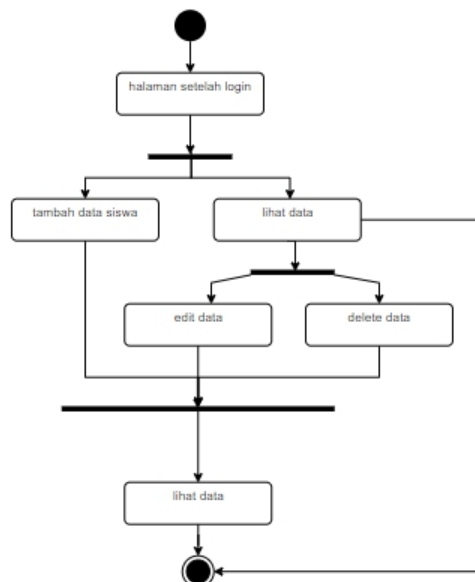
Gambar 3. 13 Activity Diagram Admin Input Data Kelas

8. *Diagram Aktivitas* Kelola Data Mata Pelajaran (Admin)



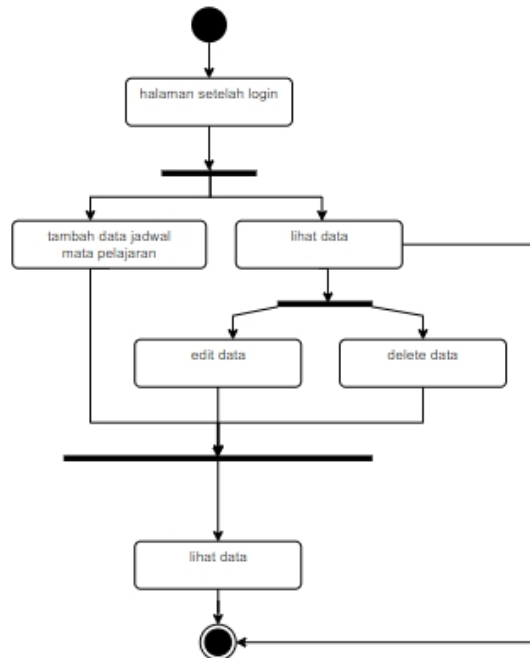
Gambar 3. 14 *Activity Diagram* admin *input* data mata Pelajaran

9. *Diagram Aktivitas* Kelola Data Siswa (Admin, Walikelas)



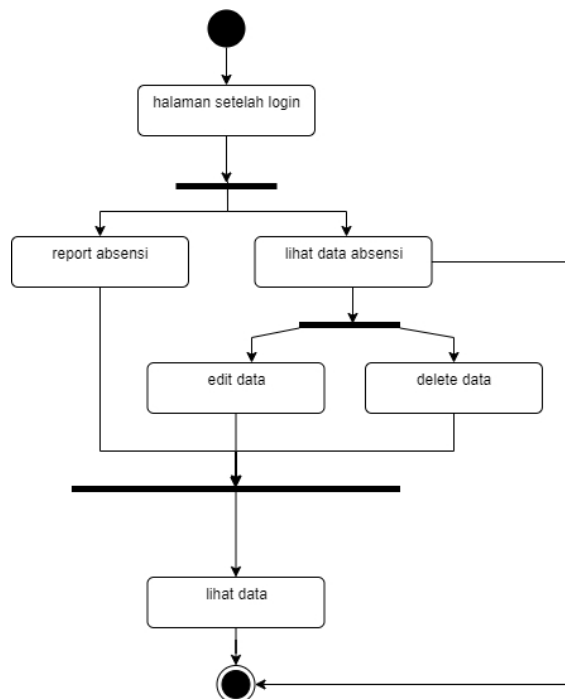
Gambar 3. 15 *Diagram Aktivitas* Kelola Data Siswa (Admin, Walikelas)

10. *Diagram Aktivitas* Kelola Data Mata Pelajaran (Admin, Guru)



Gambar 3. 16 *Activity Diagram* Admin Dan Wali Kelas *Input* Data Jadwal Mata Pelajaran

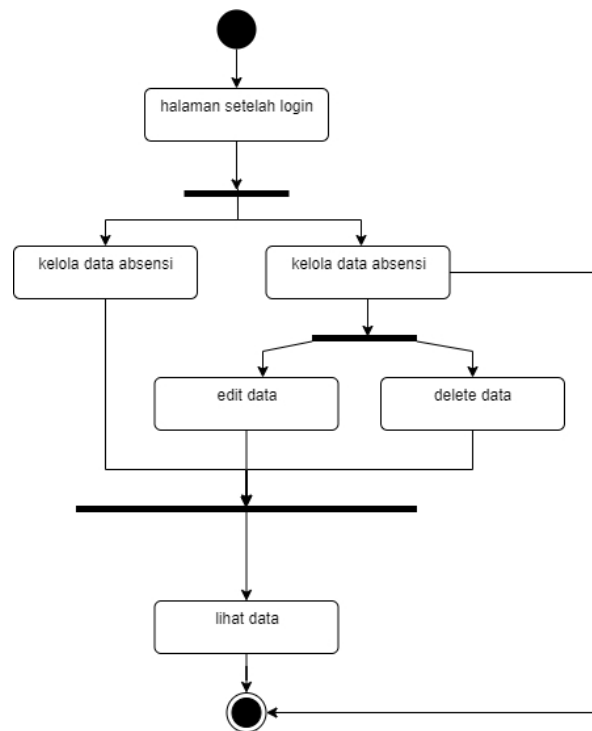
11. *Diagram Aktivitas* Report Absensi Wali Kelas Dan Kepala Sekolah



Gambar 3. 17 *Activity Diagram* report absensi wali kelas dan kepala sekolah



## 12. Diagram Aktivitas Kelola Data Absensi Guru

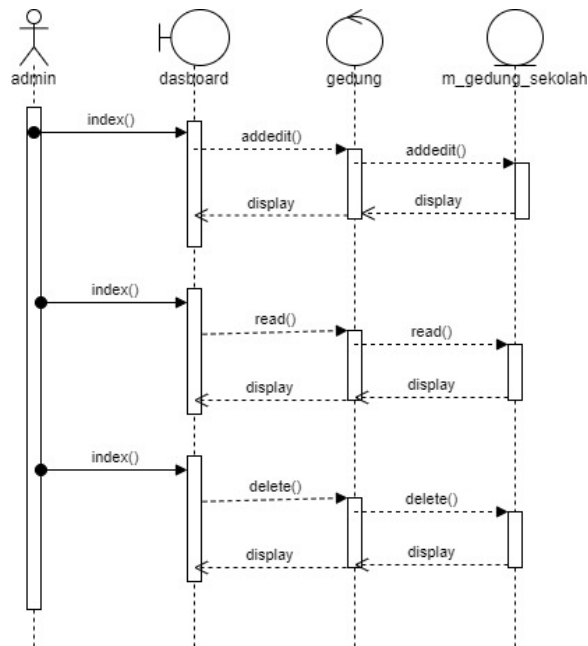


Gambar 3. 18 *Activity* Diagram Kelola Data Absensi Guru

### 3.2.6 *Sequence Diagram*

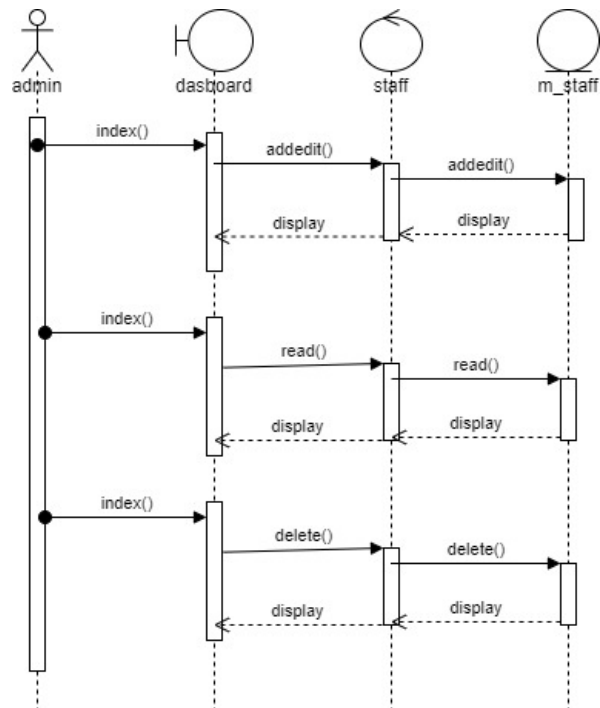
Dalam buku yang berjudul sistem monitoring lulusan perguruan tinggi dalam memasuki dunia kerja menggunakan tracer study (Harianto, Pratiwi, 64 Suhariaydi, 2019) *sequence diagram* memiliki pengertian yaitu menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang di kirimkan dan di terima antara objek.

1. *Sequence Diagram Input Data Gedung Oleh Admin*



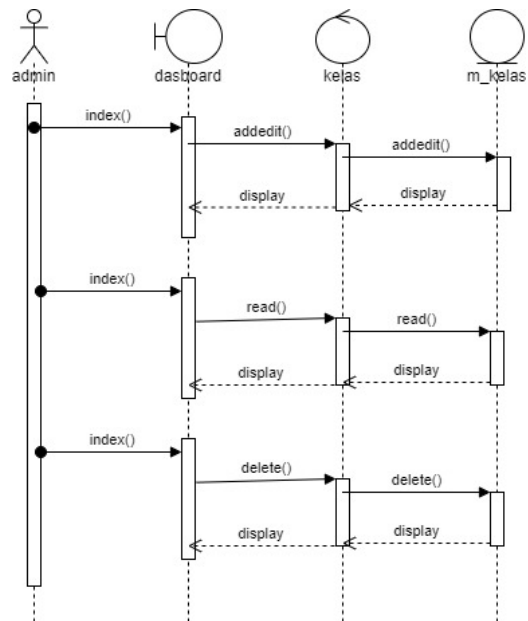
Gambar 3. 19 *Sequence Diagram admin input data gedung sekolah*

2. *Sequence diagram input data staff/ tenaga Pengajar oleh admin*



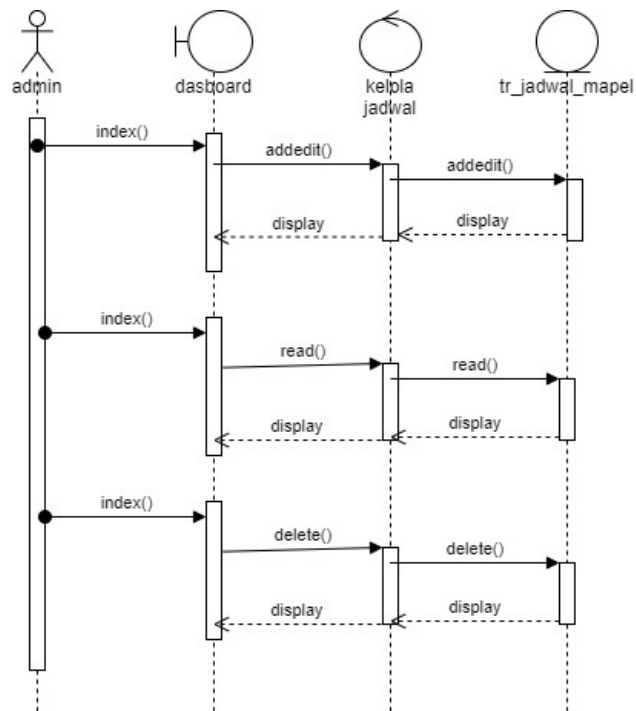
Gambar 3. 20 *Sequence Diagram admin input data staff*

### 3. Sequence Diagram Input Data Kelas Oleh Admin



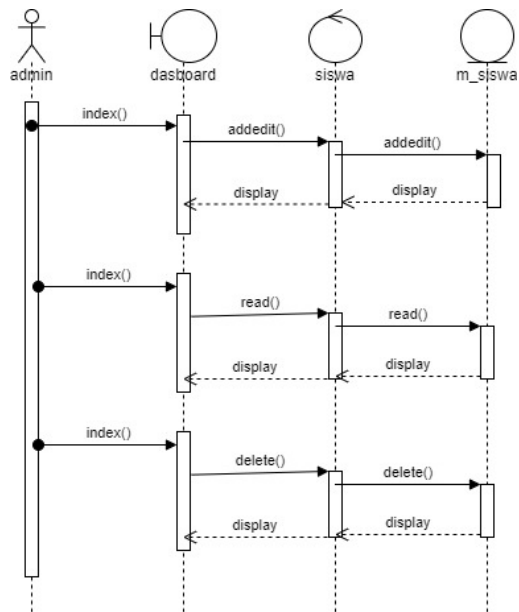
Gambar 3. 21 Sequence Diagram Admin Input Data Kelas

### 4. Sequence Diagram Input Data Mata Pelajaran Oleh Admin



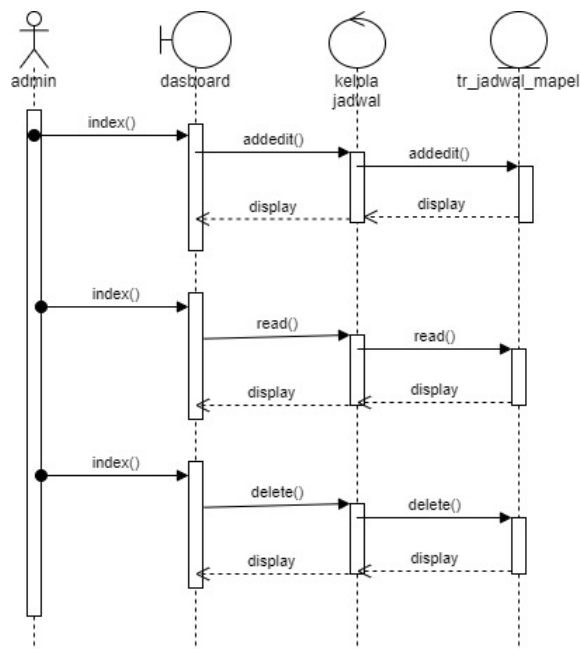
Gambar 3. 22 Sequence Diagram Admin Input Data Mata Pelajaran

### 5. Sequence Diagram *Input Data Siswa Oleh Admin*



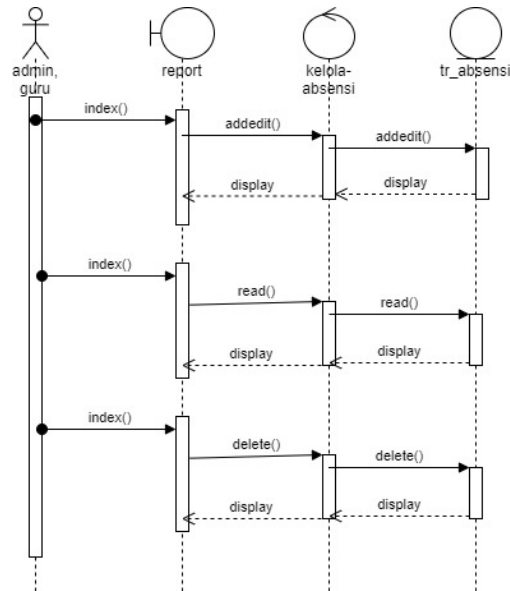
Gambar: 3.23. Gambar 3. 23 Sequence Diagram Admin *Input Data Siswa*

### 6. Sequence Diagram *Input Data Jadwal Mata Pelajaran Oleh Admin*



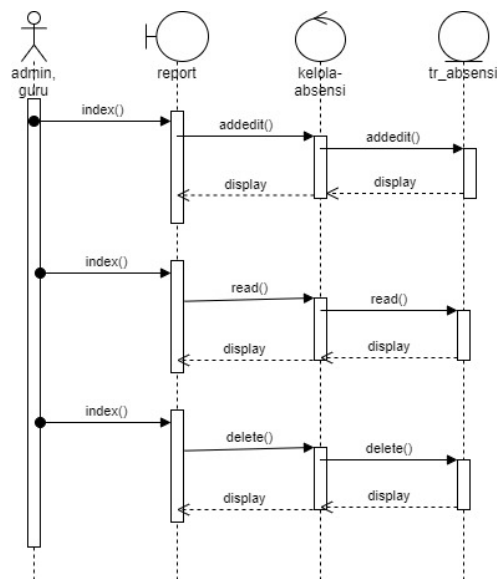
Gambar 3. 24 Sequence Diagram Admin *Input Data Jadwal Mata Pelajaran*

7. *Sequence Diagram Report Data Absensi Oleh Wali Kelas Dan Kepala Sekolah*



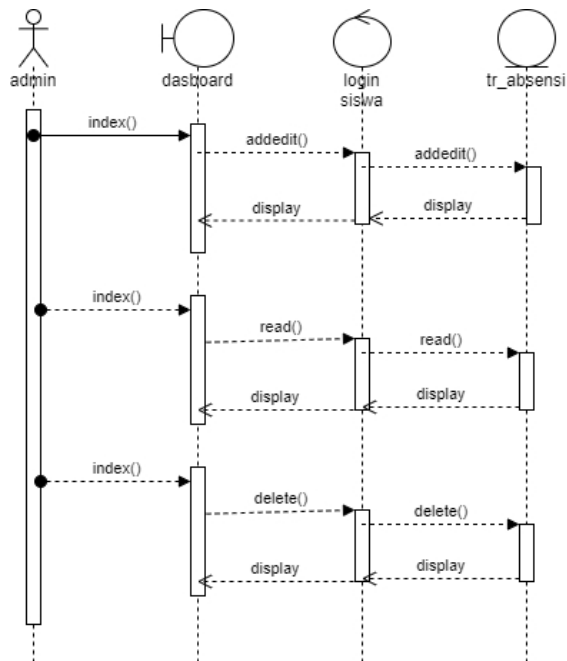
Gambar 3. 25 *Sequence Diagram Report Data Absensi Oleh Wali Kelas Dan Kepala Sekolah*

8. *Sequence Diagram Input Kelola Absensi Oleh Guru Mata Pelajaran*



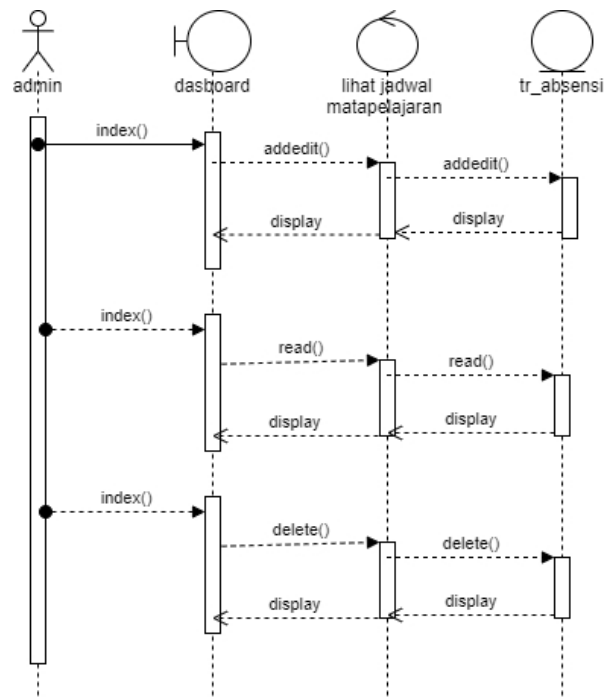
Gambar 3. 26 *Sequence Diagram Kelola Absensi Oleh Guru Mata Pelajaran*

### 9. Sequence Diagram Login Siswa



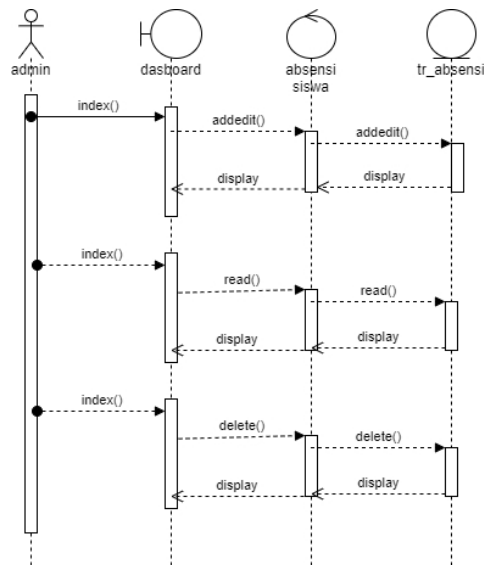
Gambar 3. 27 Sequence Diagram Absensi Siswa

### 10. Sequence Diagram Lihat Mata Pelajaran



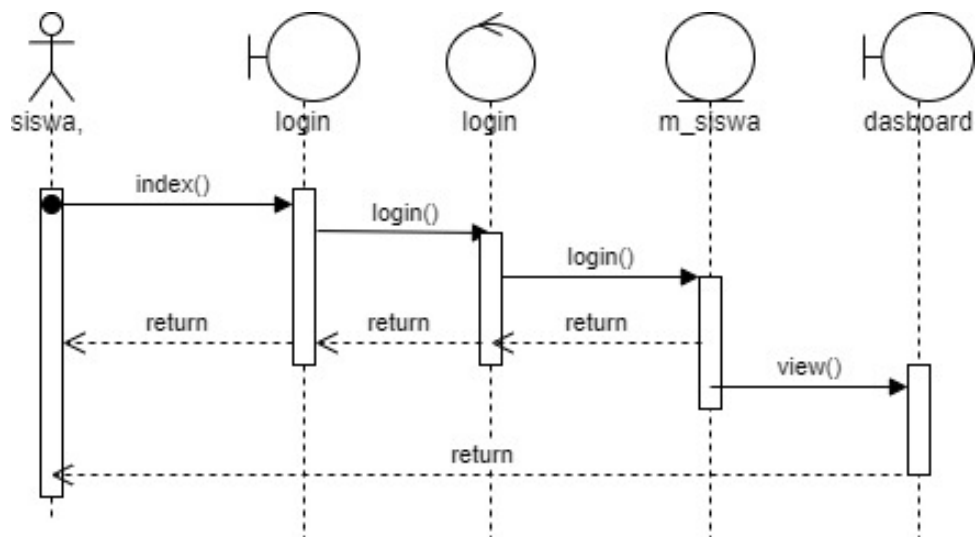
Gambar 3. 28 Sequence Diagram Lihat Mata Pelajaran

### 11. Sequence Diagram Absensi Siswa



Gambar 3. 29 Sequence Diagram Absensi Siswa

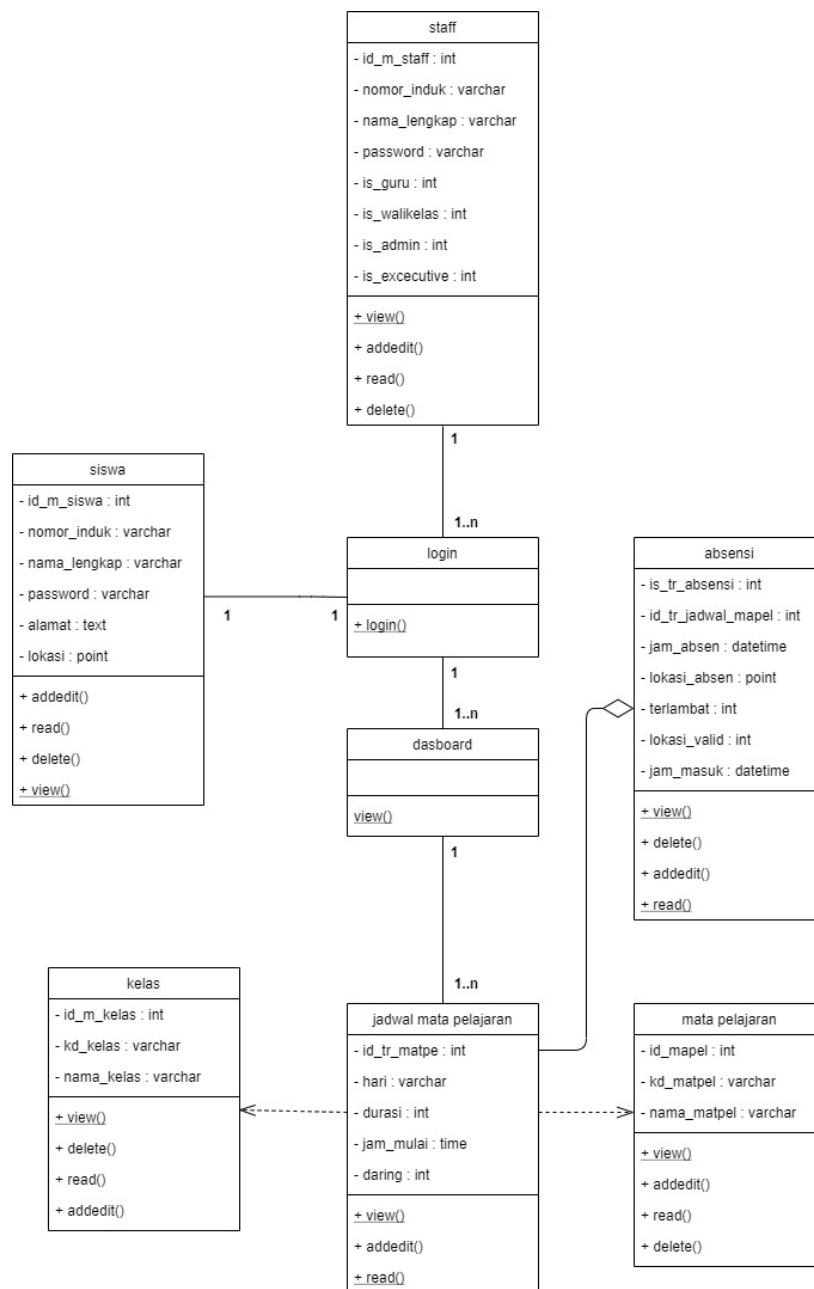
### 12. Sequence Diagram Login Absensi Siswa



Gambar 3. 30 Sequence Diagram Login Absensi Siswa

### 3.2.7 Class Diagram

*Class diagram* merupakan diagram yang menggambarkan pendefinisian kelas-kelas yang akan di buat, dalam *class diagram* juga di gambarkan hubungan statis antar kelas.

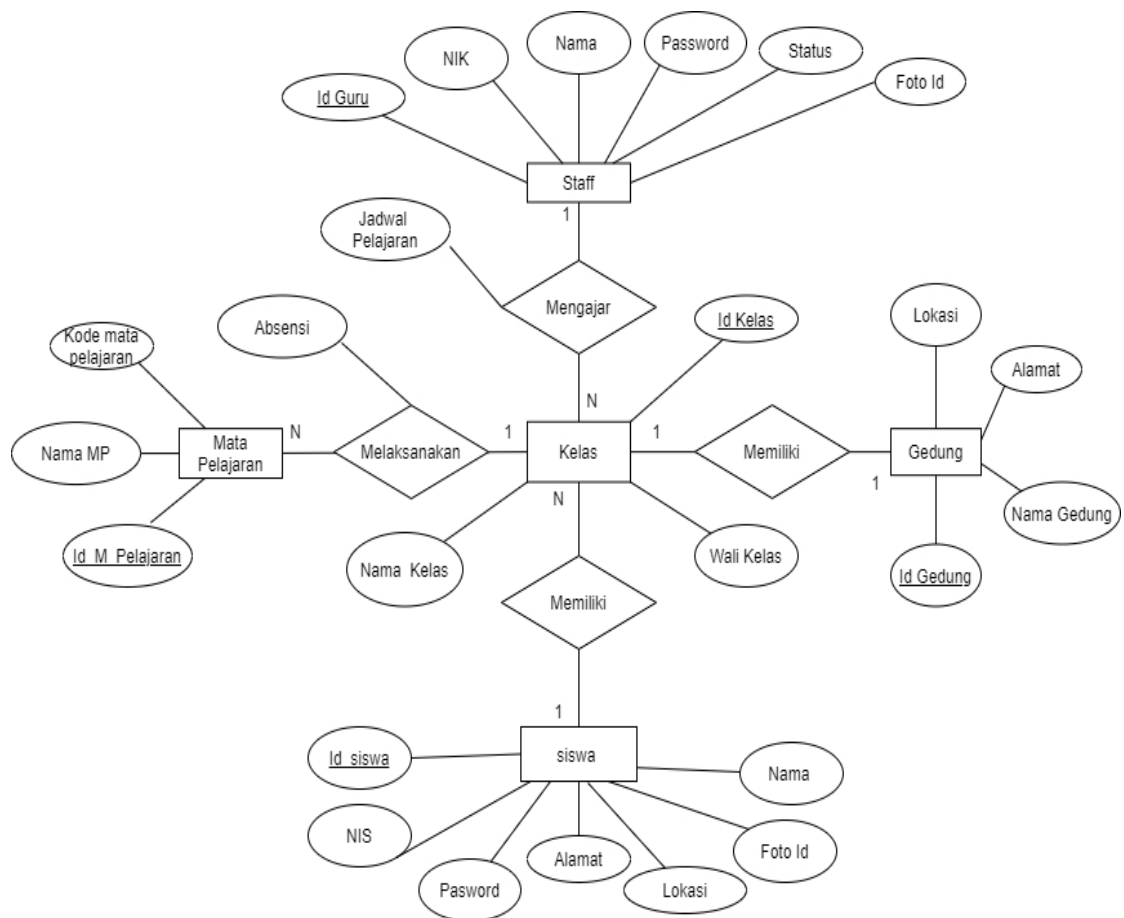


Gambar 3. 31 *Class Diagram* absensi Siswa



### 3.2.8 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan pemodelan basis data yang menggambarkan relasi antar entitas. Berikut adalah ERD dari sistem absensi siswa :



Gambar 3. 32 Entity Relationship Diagram (ERD) Absensi Siswa

### 3.2.9 Perancangan Antar Muka

Perancangan antar muka menggunakan laptop bentuk tampilan Login di absensi secara daring Adapun perancangan aplikasi adalah sebagai berikut

## 1. Tampilan *Login*

The wireframe shows a rectangular box representing a login form. At the top center, the text "Sign In" is displayed. Below this, there are two input fields: "User Id" and "Pasword". Underneath the "Pasword" field is a checkbox followed by the text "remember me". At the bottom of the form is a button labeled "Sign me in".

Gambar 3. 33 Tampilan Antar Muka Menu Login

## 2. Tampilan Menu

The wireframe depicts a menu interface. On the left side, there is a vertical sidebar containing a "logo sekolah" at the top, followed by a "dashboard menu" section with several buttons: "gedung sekolah", "staff", "kelas", "mata pelajaran", "siswa", and "jadwal pelajaran". The main area of the interface is divided into a "header" at the top right, which includes a "user id" box, and a large "halaman" (page) area below it. The "halaman" area contains a large rectangular box labeled "MAIN MENU TAMPILAN".

Gambar 3. 34 Tampilan Antar Muka Tampilan Menu

### 3. Tampilan menu Android



Gambar 3. 35 Tampilan Antar Muka Menu Aplikasi Android

#### 3.2.10 *Konstruksi dan Pembangunan (Coding and Testing)*

Tahapan-tahapan dalam proses konstruksi dan pembangunan sistem absensi online adalah sebagai berikut:

a. *Membuat Basis Data*

*Basis data* dibuat dengan *phpMyAdmin* sesuai dengan desain ERD.

b. *Implementasi Desain Simulasi Sistem Admin*

Implementasi desain sistem menggunakan *framework codeigniter* dan CSS berdasarkan desain awal sistem sebelumnya.

c. *Implementasi Pemrograman Sistem Admin*

Pemrograman menggunakan *framework codeigniter* dengan metode *Model-View-Controller* atau MVC.

d. Membuat API untuk Akses *Basis Data* Sistem Absen Online

Pembuatan API dengan bahasa pemrograman PHP untuk mengirim atau menerima data pada *basisdata*.

e. Implementasi Desain Aplikasi *Android*

Desain aplikasi android telah diimplementasikan ke dalam bentuk XML pada saat peneliti membuat *mockup*. Pada tahap ini peneliti memperbaiki *mockup* yang kurang dan menghubungkan layout dan fungsi pada *activity* di android.

f. Implementasi Metode Pertukaran Data antara Aplikasi Android dengan *Basis Data*

Peneliti meingplementasikan API yang telah dibuat dan menghubungkannya dengan aplikasi android. Pertukaran data pada versi web dilakukan dengan metode GET/POST. Sedangkan pada aplikasi android, pertukaran data dilakukan dengan bantuan aplikasi ionic yang berfungsi sebagai media perantara dalam mengakses API yang telah peneliti buat.

## **BAB IV**

### **IMPLEMENTASI DAN UJI COBA**

#### **4.1 Implementasi**

Pada tahapan ini sesuai dengan tahapan metode pengembangan sistem yang digunakan yaitu *System Development Life Cycle (SLDC)* model *Waterfall* adalah pengodean atau pembuatan kode program yang merupakan implementasi dari rancangan pembangunan aplikasi klasifikasi penduduk yang telah dibuat sebelumnya.

##### **4.1.1 Perangkat Keras (Hardware)**

Perangkat keras (*Hardware*) yang digunakan untuk membangun dan menjalankan perancangan aplikasi absensi siswa secara daring di lingkungan SMK Yadika Tanjungsari menggunakan laptop azus dengan spesifikasi sebagai berikut:

1. Processor Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz
2. RAM 6 GB
3. Kapasitas Harddisk 463 GB

Serta menggunakan android Samsung Galaksi A02 dengan spesifikasi sebagai berikut:

1. Chipset & OS Mediatek MT6739W GPU Power VR GE8100
2. Android 10, One UI 2.0

### 3. Memori 3GB RAM + 32GB ROM

Minimum kebutuhan Perangkat keras (*Hardware*) yang digunakan untuk menjalankan aplikasi absensi siswa secara daring di lingkungan SMK Yadika Tanjungsari menggunakan laptop dengan spesifikasi sebagai berikut:

1. Processor Intel(R) Core(TM) i3 1.70GHz
2. RAM 2 GB
3. Kapasitas Harddisk 128 GB

Serta minimum penggunaan android dengan spesifikasi sebagai berikut:

1. CPU Okta-core 2,0 GHz
2. Android 7,
3. Memori 2GB RAM + 16GB ROM

#### **4.1.2 Perangkat Lunak (*Software*)**

Selain perangkat keras, digunakan juga beberapa perangkat lunak (*Software*) yang merupakan pendukung sistem yang terdiri dari sistem operasi dan aplikasi database. Perangkat lunak yang digunakan dalam pembangunan aplikasi absensi siswa secara daring di lingkungan SMK Yadika Tanjungsari adalah sebagai berikut:

1. Sistem Operasi Windows 10
2. Draw IO untuk perancangan UML dan ERD
3. Visual Studio Code Versi 1.56 April 2021

4. Bahasa Pemrograman java menggunakan NetBeans, PHP MyAdmin 5.0
5. Data Base menggunakan MySQL
6. Mozilla Firefox
7. HTTP Screen Scren

## 4.2 Pengkodean

Pengkodean adalah suatu teknik yang dilakukan untuk memberikan penegasan kepada proses yang terlibat transmisi data. Dan dalam proses tersebut perlu untuk diperhatikan segala fasilitas komunikasi dan media yang tersedia. Berikut tahapan pengkodean

### 4.2.1 Kamus Data

Berdasarkan pemodelan ERD dan rancangan tabel-tabel yang telah dibuat sebelumnya, maka implementasi basis data (*database*) untuk aplikasi absensi siswa secara daring di lingkungan SMK Yadika Tanjungsari menggunakan web server dari XAMPP versi 3.2.4 dengan MySQL. Berikut implementasi basis data aplikasi absensi secara daring di Smk Yadika :

1. Implementasi struktur basis data (*database*)

Tabel 4. 14 Struktur Tabel m\_siswa

No	Field	Tipe Data	Size	Index
1.	id_m_siswa	int	11	Primary Key
2.	nomor_induk	varchar	100	Forgent Key
3.	nama_lengkap	varchar	100	

No	Field	Tipe Data	Size	Index
4.	password	varchar	100	
5.	alamat	Text		
6.	lokasi	point		
7.	id_m_kelas	int	11	

Tabel 4. 15 Struktur Tabel m\_matpel

No	Field	Tipe Data	Size	Index
1.	id_m_matpel	int	11	Primary Key
2.	kd_matpel	varchar	10	
3.	nm_matpel	varchar	100	Forgent Key

Tabel 4. 16 Struktur Tabel m\_gedung\_sekolah

No	Field	Tipe Data	Size	Index
1.	id_m_gedung_sekolah	int	11	Primary Key
2.	nm_gedung_sekolah	varchar	100	Forgent Key
3.	alamat	text		
4.	lokasi	point		

Tabel 4. 17 Struktur Tabel m\_kelas

No	Field	Tipe Data	Size	Index
1.	id_m_kelas	int	11	Primary Key
2.	kd_kelas	varchar	10	Forgent Key
3.	nama_kelas	varchar	100	



No	Field	Tipe Data	Size	Index
4.	id_wali_kelas	int	11	
5.	id_m_gedung_sekolah	int	11	

Tabel 4. 18 Struktur Tabel m\_staff

No	Field	Tipe Data	Size	Index
1.	id_m_staff	int	11	Primary Key
2.	nama_lengkap	varchar	100	
3.	photo	varchar	100	
4.	nomor_induk	varchar	100	Forgent Key
5.	password	varchar	100	
6.	is_walikelas	tinyint	1	
7.	is_guru	tinyint	1	
8.	is_executive	tinyint	1	
9.	is_administrator	tinyint	1	
10.	uid	varchar	32	

Tabel 4. 19 Struktur Tabel tr\_absensi

No	Field	Tipe Data	Size	Index
1.	id_tr_absensi	int	11	Primary Key
2.	id_m_siswa	int	11	
3.	id_tr_jadwal_matpel	int	11	
4.	jam_absens	datetime		

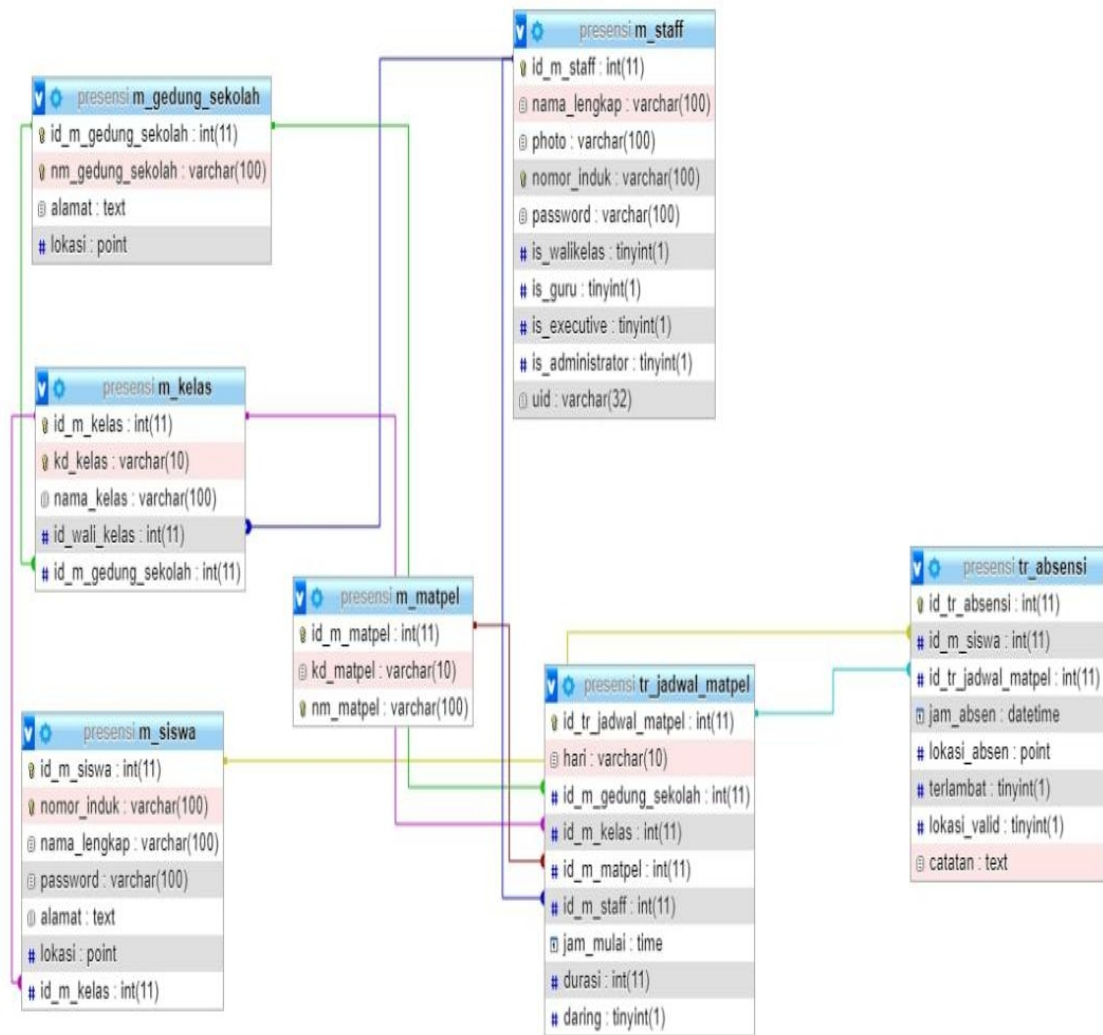
No	Field	Tipe Data	Size	Index
5.	lokasi_absen	point		
6.	terlambat	tinyint	1	
7.	lokasi_valid	tinyint	1	
8.	catatan	text		

Tabel 4. 20 Struktur Tabel tr\_jadwal matpel

No	Field	Tipe Data	Size	Index
1.	tr_jadwal_matpel	int	11	Primary Key
2.	hari	varchar	10	
3.	id_m_gedung_sekolah	int	11	
4.	id_m_kelas	int	11	Foreign Key
5.	id_m_matpel	int	11	
6.	id_m_staff	int	11	
7.	Jam_mulai	time		
8.	durasi	int	11	
9.	daring	tinyint	1	

## 4.2.2 Relasi Tabel

Berikut merupakan relasi table sistem absensi siswa :

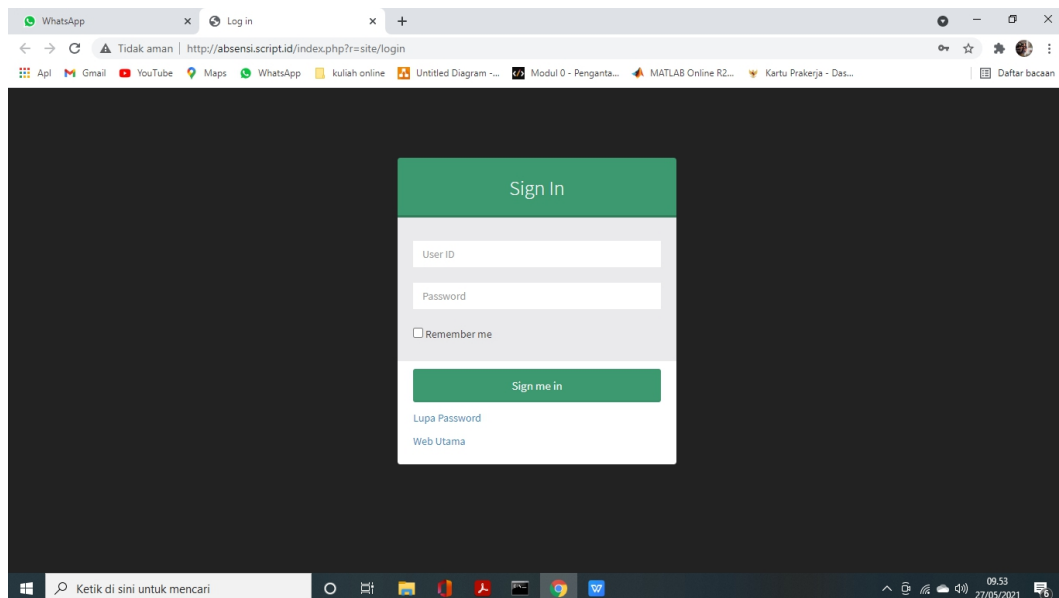


Gambar 4. 1 Relasi Tabel sistem aplikasi absensi siswa

### 4.3 Implementasi Antarmuka (Interface)

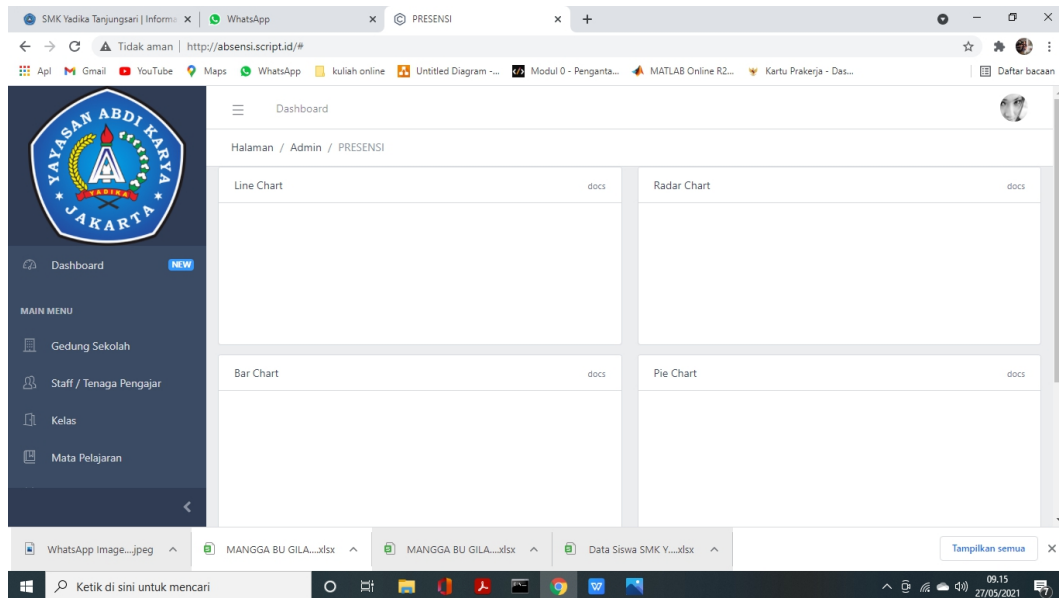
Aplikasi saya buat dengan mengikuti kriteria-kriteria yaitu mudah dioperasikan (user friendly), serta sesuai dengan permasalahan yang dibahas, sehingga sistem aplikasi yang dikembangkan diharapkan mampu menjawab segala permasalahan yang terjadi di SMK Yadika Tanjungsari.

#### 4.3.1 Tampilan Antar Muka Menu Login



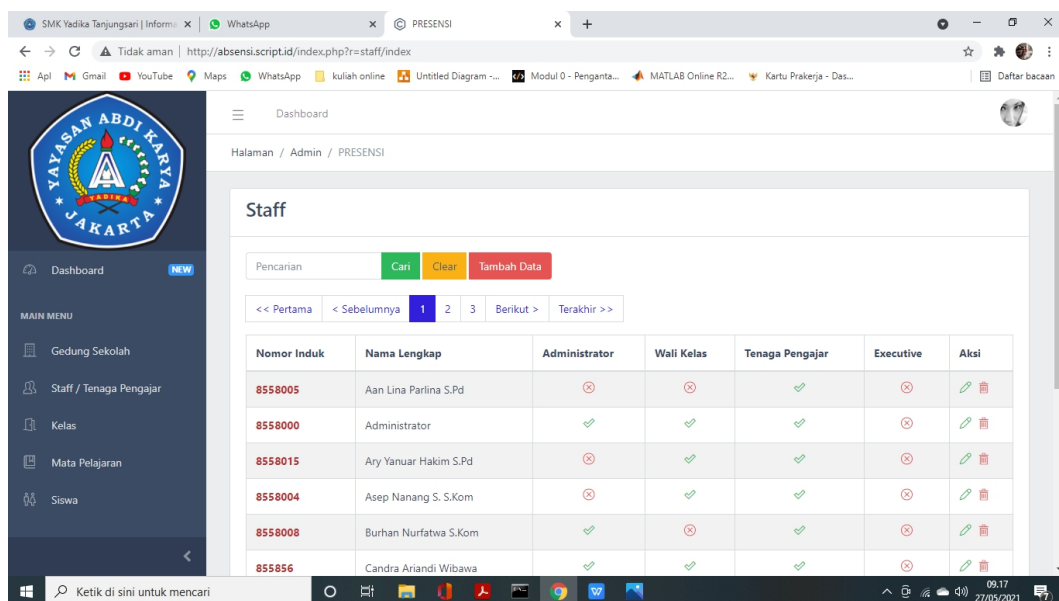
Gambar 4. 2 Gambar Antar Muka Menu *Login*

### 4.3.2 Tampilan Antar Muka Main Menu



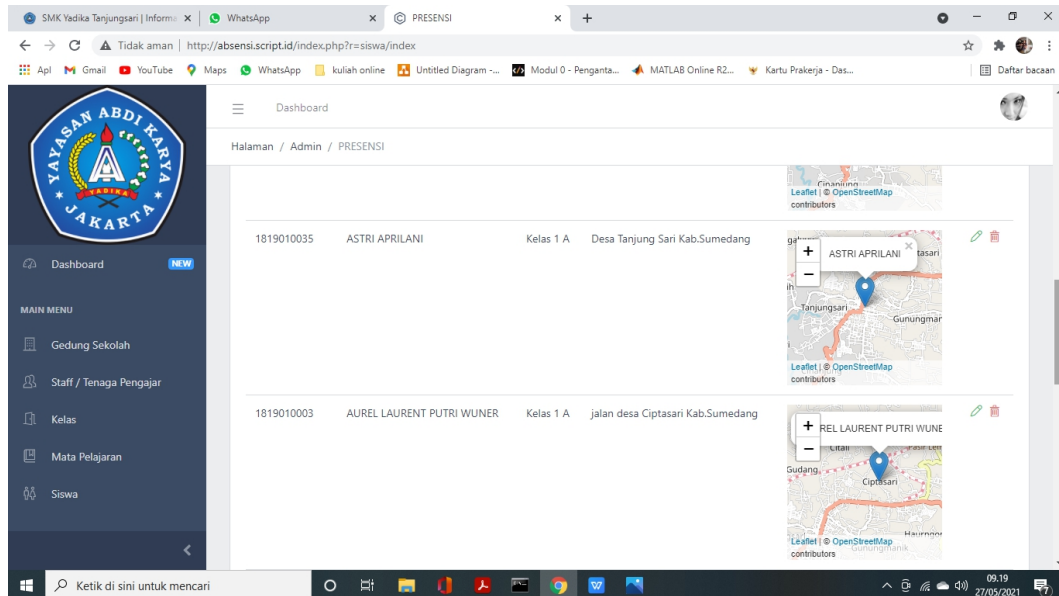
Gambar 4. 3 Gambar Tampilan Antar Muka Main Menu Sistem Absensi

### 4.3.3 Tampilan Antar Muka Input Data Staff



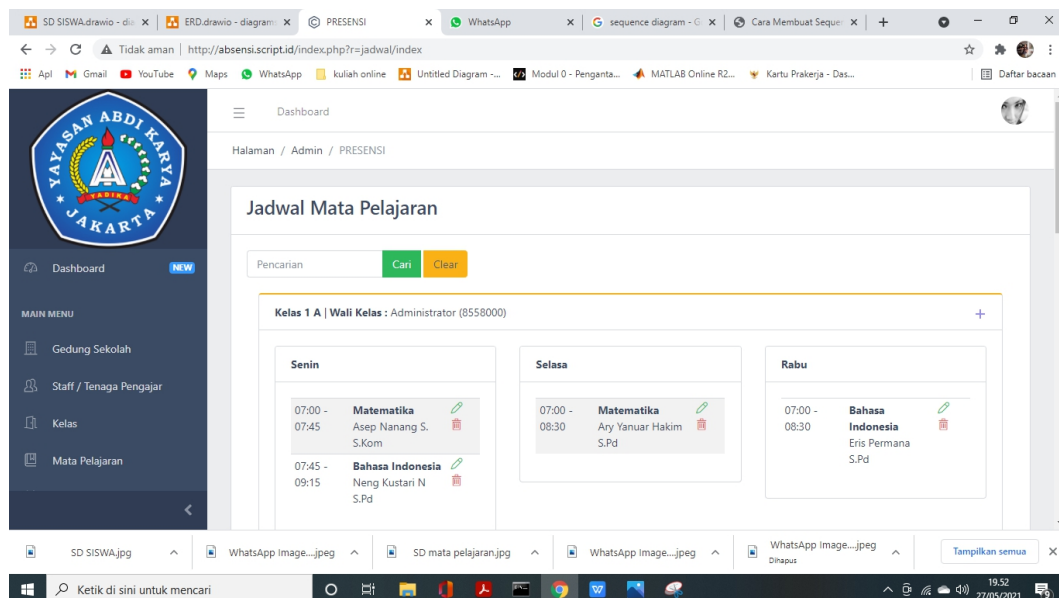
Gambar 4. 4 Gambar Tampilan Antar Muka Menu Input Data Staff Pengajar

### 4.3.4 Tampilan Antar Muka Input Data Siswa



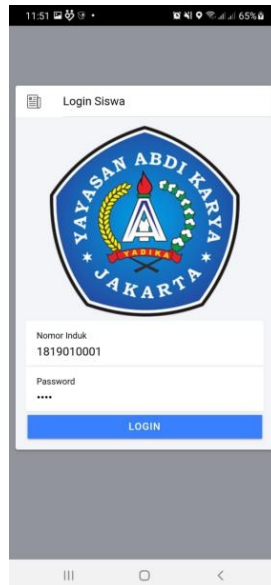
Gambar 4. 5 Gambar Tampilan Antar Muka Menu Input Data Staff Pengajar

### 4.3.5 Tampilan Antar Muka Input Jadwal Mata Pelajaran

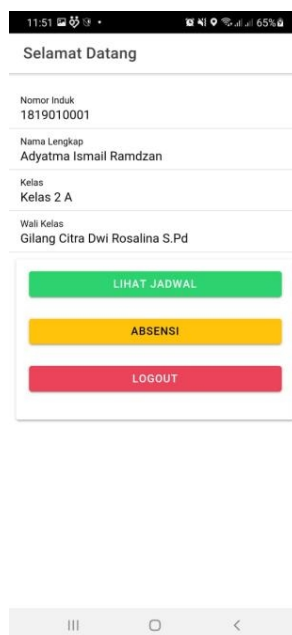


Gambar 4. 6 Gambar Tampilan Antar Muka Menu *Input* Data Staff

#### 4.3.6 Tampilan Menu Utama di Smartphone

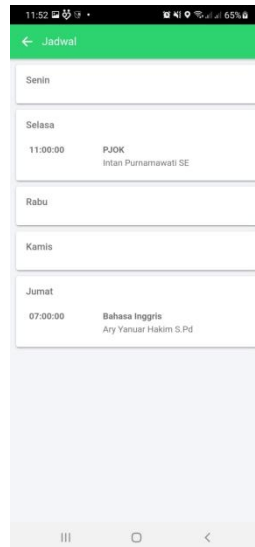


Gambar 4. 7 Gambar Tampilan Antar Muka Menu Login Data Siswa



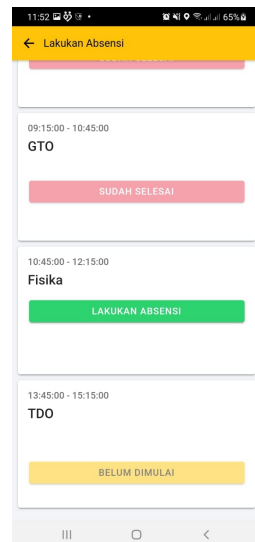
Gambar 4. 8 Gambar Tampilan Antar Muka Menu Utama Siswa  
Di Smartphone

#### 4.3.7 Tampilan Menu Lihat Jadwal Siswa Di Smartphone



Gambar 4. 9 Gambar Tampilan Antar Muka Menu Lihat Jadwal Mata Pelajaran Siswa Di Smartphone

#### 4.3.8 Tampilan Menu Absensi Siswa Di Smartphone



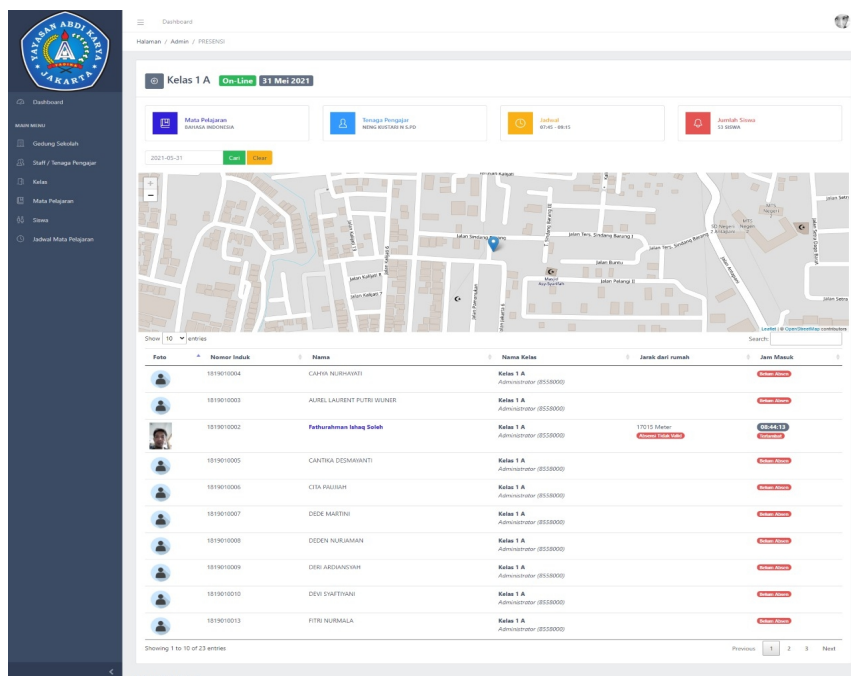
Gambar 4. 10 Gambar Tampilan Antar Muka Menu Absensi Siswa Melalui SmartPhone





Gambar 4. 11 Tampilan Menu Ambil Foto Dan Lakukan Absensi Siswa Di Smartphone

#### 4.3.9 Tampilan Menu Report Absensi Siswa



Gambar 4. 12. Laporan Hasil Absensi Siswa

#### 4.4 Uji Coba (Testing)

Pada tahap ini dilakukan uji coba atau pengujian aplikasi yang sudah dibangun. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan (*input*), dan keluaran (*output*) dari perangkat lunak sesuai dengan spesifikasi yang diharapkan. Metode atau pendekatan pengujian yang digunakan dalam tahapan ini adalah dengan *Black-Box Testing* (pengujian kotak hitam) yang merupakan pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Berikut adalah hasil dari pengujian menggunakan metode *Blackbox Testing* :

##### 4.4.1 Pengujian Login dan Logout User

Tabel 4. 21 Tabel Pengujian Login dan Logout User

No	Deskripsi Pengujian Kasus	Masukan/ Aksi	Hasil yang Diharapkan	Kesimpulan yang di dapat
1.	Menguji login siswa	Proses melakukan pengisian nomor identitas siswa (NIS)	Proses login berhasil Menampilkan NIS harus di isi	sesuai
2.	Menguji login siswa	Tidak melakukan pengisian NIS	Proses login gagal, NIS harus di isi	sesuai
3.	Menguji login guru	Proses melakukan pengisian Nomor Identitas Pengajar (NIP)	Proses login berhasil Menampilkan NIS harus di isi	sesuai

No	Deskripsi Pengujian Kasus	Masukan/ Aksi	Hasil yang Diharapkan	Kesimpulan yang di dapat
4.	Menguji login guru	Tidak melakukan pengisian NIP	Proses login gagal, NIP harus di isi	sesuai
5	Menguji logout aplikasi	Menu logout di klik	Proses logout berhasil keluar dari aplikasi	sesuai

#### 4.4.2 Pengujian Proses Sistem Absensi

Tabel 4. 22 Tabel Pengujian Sistem Absensi Siswa Dan Guru

No	Deskripsi pengujian Kasus	Masukan/ Aksi	Hasil yang diharapkan	Kesimpulan Yang di dapat
1.	Menguji sistem absensi siswa	Proses siswa melakukan identifikasi melalui photo dan cek lokasi	Proses identifikasi berhasil menampilkan nama siswa dan mencatat point lokasi	sesuai
2.	Laporan absensi siswa/guru	Print laporan absensi	Proses laporan berhasil di print	sesuai
3.	Input data gedung sekolah	Masukan nama dan alamat sekolah	Proses berhasil di inputkan	sesuai
4.	Input data staff / tenaga pengajar	Masukan nama staff/tenaga pengajar, status guru	Proses berhasil di inputkan	sesuai

No	Deskripsi pengujian Kasus	Masukan/ Aksi	Hasil yang diharapkan	Kesimpulan Yang di dapat
5.	Input data kelas	Masukan data data kelas dan input data walikelas	Proses berhasil di inputkan	sesuai
6.	Input data mata pelajaran	Masukan data mata pelajaran dan staff/ tenaga pengajar	Proses berhasil di inputkan	sesuai
7.	Input data siswa	Masukan data siswa, alamat, lokasi GPS rumah	Proses berhasil di inputkan	sesuai
8.	Input data jadwal mata pelajaran	Masukan jadwal mata pelajaran, data kelas, dan data tenaga pengajar	Proses berhasil di inputkan	sesuai

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari penelitian yang telah dilakukan di SMK Yadika Tanjungsari mengenai sistem aplikasi absensi secara daring didapat kesimpulan sebagai berikut:

1. Dengan dibangunnya aplikasi absensi secara daring di SMK Yadika Tanjungsari mempermudah siswa dalam melakukan absensi secara daring
2. Dengan adanya aplikasi ini pengaturan absensi, jadwal kelas, dan penempatan ruang kelas menjadi lebih mudah.
3. Dengan dibangunnya aplikasi absensi secara daring di SMK Yadika Tanjungsari memudahkan dalam memantau keberadaan siswa selama berjalannya kegiatan belajar dan mengajar, sehingga baik wali kelas maupun kepala sekolah dapat memonitoring aktivitas belajar siswa secara berkelanjutan.

#### **5.2 Saran**

Penelitian ini masih terdapat banyak kekurangan, untuk peningkatan dan pengembangan sistem aplikasi ini di masa mendatang penulis mempunyai beberapa saran, antara lain :

1. Aplikasi absensi secara daring perlu ditambahkan fitur-fitur untuk menunjang kebutuhan pengguna dalam pengoprasian sistem yang dibuat.

2. Aplikasi absensi yang dibuat memerlukan akurasi tinggi mengenai *Global Positioning System* (GPS) sehingga dibutuhkan alat alat penunjang yang lebih mumpuni untuk mengoperasikan sistem absensi ini. Keterbatasan sarana dan prasarana dalam membuat aplikasi menjadikan koordinat GPS masih banyak yang menyimpang.

## DAFTAR PUSTAKA

- Rosa A, S. and Shalahuddin, M. (2019) *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. 2nd-Revisi edn. Edited by S. Rosa A and M. Shalahuddin. Bandung: Informatika Bandung.
- Sandro Alfeno , Ririn Eka Cipta Devi (2017) ” Implementasi *Global Positioning System* (GPS) dan *Location Based Service* (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek” JURNAL SISFOTEK GLOBAL ISSN : 2088 – 1762 Vol. 7 No. 2, September 2017
- Novita Retno P L, Sarwosri, dan Dwi Sunaryono (2017) “ Rancang Bangun Aplikasi *Mobile* Android Sistem Kehadiran Mahasiswa melalui Pencocokan Tanda Tangan Studi Kasus Jurusan Teknik Informatika ITS” JURNAL TEKNIK ITS Vol. 6, No. 2 (2017), 2337-3520 (2301-928X Print)
- Sukanto & Shalahudin Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. . Bandung: Informatika Bandung.
- Eko Budi Setiawan, Bobi Kurniawan “Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan Radio Frequency Identification (RFid)” Jurnal CoreIT, Vol.1, No.2, Desember 2015 ISSN: 2460-738X (Cetak)
- Santoso, H., & Yulianto, A. (2017). ANALISA DAN PERANCANGAN SISTEM ABSENSI SISWA BERBASIS WEB DAN SMS GATEWAY. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*,
- Siswo Wardoyo, Romi Wiryadinata, Raya Sagita* “ Sistem Presensi Berbasis Algoritma Eigenface Dengan Metode Principal Component Analysis” <https://jurnal.untirta.ac.id/index.php/jis/article/view/498>
- <https://kumparan.com/kumparansains/imbaspandemi-virus-corona-bagi-dunia-pendidikan-indonesia-dan-global-1t5YVXRYAbo/full>.
- <https://www.kemdikbud.go.id/main/blog/2020/03/se-mendikbud-pelaksanaan-kebijakan-pendidikan-dalam-masa-darurat-penyebaran-covid19>.
- Baenil Huda, Bayu Priyatna “Penggunaan Aplikasi Content Manajement System (CMS) Untuk Pengembangan Bisnis Berbasis E-Commerce” SYSTEMATICS, Vol. 1, No. 2, Desember 2019, pp 81-88
- (Harianto, Pratiwi, 64 Suhariaydi, 2019) Sistem Monitoring Lulusan Perguruan Tinggi dalam memasuki dunia kerja menggunakan tracer study *sequence diagram*”

Yosua P.W Simaremare, Apol Pribadi S dan Radityo Prasetianto Wibowo  
“Perancangan dan Pembuatan Aplikasi Manajemen Publikasi Ilmiah  
Berbasis Online pada Jurnal SISFO Jurusan Sistem Informasi, Fakultas  
Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)



## LAMPIRAN LIST PROGRAM

```
<?php

/**
 * This is the model class for table "m_staff".
 *
 * The followings are the available columns in table 'm_staff':
 * @property integer $id_m_staff
 * @property string $nama_lengkap
 * @property string $photo
 * @property string $nomor_induk
 * @property string $password
 * @property integer $is_walikelas
 * @property integer $is_guru
 * @property integer $is_executive
 * @property integer $is_administrator
 * @property string $uid

 *
 * The followings are the available model relations:
 * @property MKelas[] $mKelas
 * @property TrJadwalMatpel[] $trJadwalMatpels
 */
class MStaff extends CActiveRecord
{
    /**
     * Returns the static model of the specified AR class.
     * @param string $className active record class name.
     * @return MStaff the static model class
     */

    public $m_staff_id_m_staff="";
    public $m_staff_nama_lengkap="";
    public $m_staff_photo="";
    public $m_staff_nomor_induk="";
    public $m_staff_password="";
    public $m_staff_is_walikelas="";
    public $m_staff_is_guru="";
    public $m_staff_is_executive="";
    public $m_staff_is_administrator="";
    public $m_staff_uid="";
    public $modulename = "MStaff";

    private $sqlwhere = " WHERE
```

```

        `id_m_staff` LIKE :id_m_staff OR
        `nama_lengkap` LIKE :nama_lengkap OR
        `photo` LIKE :photo OR
        `nomor_induk` LIKE :nomor_induk OR
        `password` LIKE :password OR
        `is_walikelas` LIKE :is_walikelas OR
        `is_guru` LIKE :is_guru OR
        `is_executive` LIKE :is_executive OR
        `is_administrator` LIKE :is_administrator OR
        `uid` LIKE :uid OR
";

public static function model($className=__CLASS__)
{
    return parent::model($className);
}

/**
 * @return string the associated database table name
 */
public function tableName()
{
    return 'm_staff';
}

/**
 * @return array validation rules for model attributes.
 */
public function rules()
{
    // NOTE: you should only define rules for those attributes that
    // will receive user inputs.
    return array(
        array('nama_lengkap, nomor_induk', 'required'),
        array('is_walikelas, is_guru, is_executive, is_administrator', 'numerical',
'integerOnly'=>true),
        array('nama_lengkap, photo, nomor_induk, password', 'length', 'max'=>100),
        array('uid', 'length', 'max'=>32),
        // The following rule is used by search().
        // Please remove those attributes that should not be searched.
        array('id_m_staff, nama_lengkap, photo, nomor_induk, password, is_walikelas,
is_guru, is_executive, is_administrator, uid', 'safe', 'on'=>'search'),
    );
}

/**
 * @return array relational rules.
 */
public function relations()
{
    // NOTE: you may need to adjust the relation name and the related
    // class name for the relations automatically generated below.

```

```

        return array(
            'mKelas' => array(self::HAS_MANY, 'MKelas', 'id_wali_kelas'),
            'trJadwalMatpels' => array(self::HAS_MANY, 'TrJadwalMatpel', 'id_m_staff'),
        );
    }

    /**
     * @return array customized attribute labels (name=>label)
     */
    public function attributeLabels()
    {
        return array(
            'id_m_staff' => 'Id M Staff',
            'nama_lengkap' => 'Nama Lengkap',
            'photo' => 'Photo',
            'nomor_induk' => 'Nomor Induk',
            'password' => 'Password',
            'is_walikelas' => 'Is Walikelas',
            'is_guru' => 'Is Guru',
            'is_executive' => 'Is Executive',
            'is_administrator' => 'Is Administrator',
            'uid' => 'Uid',
        );
    }

    /**
     * Retrieves a list of models based on the current search/filter conditions.
     * @return CActiveDataProvider the data provider that can return the models based on the
    search/filter conditions.
     */
    public function search()
    {
        // Warning: Please modify the following code to remove attributes that
        // should not be searched.

        $criteria=new CDbCriteria;

        $criteria->compare('id_m_staff',$this->m_staff_id_m_staff);
        $criteria->compare('nama_lengkap',$this->m_staff_nama_lengkap,true);
        $criteria->compare('photo',$this->m_staff_photo,true);
        $criteria->compare('nomor_induk',$this->m_staff_nomor_induk,true);
        $criteria->compare('password',$this->m_staff_password,true);
        $criteria->compare('is_walikelas',$this->m_staff_is_walikelas);
        $criteria->compare('is_guru',$this->m_staff_is_guru);
        $criteria->compare('is_executive',$this->m_staff_is_executive);
        $criteria->compare('is_administrator',$this->m_staff_is_administrator);
        $criteria->compare('uid',$this->m_staff_uid,true);

        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }

```

```

/* This section use by Auth Manager
=====
*/

    public function isView()
    {
        return Yii::app()->getModule('auth')->isView($this->modulename,Yii::app()-
>getModule('auth')->getPostID());
    }

    public function isAdd()
    {
        return Yii::app()->getModule('auth')->isAdd($this->modulename,Yii::app()-
>getModule('auth')->getPostID());
    }

    public function isEdit()
    {
        return Yii::app()->getModule('auth')->isEdit($this->modulename,Yii::app()-
>getModule('auth')->getPostID());
    }

    public function isDelete()
    {
        return Yii::app()->getModule('auth')->isDelete($this->modulename,Yii::app()-
>getModule('auth')->getPostID());
    }
/* End Section
=====
*/

    public function RetrieveByID($id_m_staff)
    {

        $sql = "SELECT * FROM `m_staff` WHERE `id_m_staff` = :id_m_staff LIMIT 1";

        $connection=Yii::app()->db;
        $command=$connection->createCommand($sql);
        $command->bindParam(":id_m_staff", $id_m_staff, PDO::PARAM_INT);
        $dataReader=$command->query();
        foreach($dataReader as $row) {
            $this->m_staff_id_m_staff = $row['id_m_staff'];
            $this->m_staff_nama_lengkap = $row['nama_lengkap'];
            $this->m_staff_photo = $row['photo'];
            $this->m_staff_nomor_induk = $row['nomor_induk'];
            $this->m_staff_password = $row['password'];
            $this->m_staff_is_walikelas = $row['is_walikelas'];
            $this->m_staff_is_guru = $row['is_guru'];
            $this->m_staff_is_executive = $row['is_executive'];
        }
    }

```

```
$this->m_staff_is_administrator = $row['is_administrator'];  
$this->m_staff_uid = $row['uid'];
```

```
    }  
}
```

```
public function SaveData($id_m_staff)  
{  
    if($id_m_staff==0){  
        $sql = 'INSERT INTO  
'm_staff'(`id_m_staff`,`nama_lengkap`,`photo`,`nomor_induk`,`password`,`is_walikelas`,`is_guru`,`  
'is_executive`,`is_administrator`,`uid`)  
VALUES(:id_m_staff,:nama_lengkap,:photo,:nomor_induk,:password,:is_walikelas,:is_guru,:is_ex  
ecutive,:is_administrator,:uid)';  
    }else{  
        $sql = 'UPDATE `m_staff` SET  
'id_m_staff`=:id_m_staff,`nama_lengkap`=:nama_lengkap,`photo`=:photo,`nomor_induk`=:nomo  
r_induk,`password`=:password,`is_walikelas`=:is_walikelas,`is_guru`=:is_guru,`is_executive`=:is_e  
xecutive,`is_administrator`=:is_administrator,`uid`=:uid WHERE `id_m_staff` = :id_m_staff';  
    }  
    try {  
        $connection=Yii::app()->db;  
        $command=$connection->createCommand($sql);  
        $command->bindParam(':id_m_staff', $this-  
>m_staff_id_m_staff,PDO::PARAM_INT);  
        $command->bindParam(':nama_lengkap', $this-  
>m_staff_nama_lengkap,PDO::PARAM_STR);  
        $command->bindParam(':photo', $this->m_staff_photo,PDO::PARAM_STR);  
        $command->bindParam(':nomor_induk', $this-  
>m_staff_nomor_induk,PDO::PARAM_STR);  
        $command->bindParam(':password', $this->m_staff_password,PDO::PARAM_STR);  
        $command->bindParam(':is_walikelas', $this->m_staff_is_walikelas,PDO::PARAM_INT);  
        $command->bindParam(':is_guru', $this->m_staff_is_guru,PDO::PARAM_INT);  
        $command->bindParam(':is_executive', $this-  
>m_staff_is_executive,PDO::PARAM_INT);  
        $command->bindParam(':is_administrator', $this-  
>m_staff_is_administrator,PDO::PARAM_INT);  
        $command->bindParam(':uid', $this->m_staff_uid,PDO::PARAM_STR);  
  
        $dataReader=$command->execute();  
        return true;  
    }  
    catch( CDbException $Exception ) {  
        return false;  
    }  
}
```

```
public function DeleteData($id_m_staff)
```

```

{
    $sql = 'DELETE FROM `m_staff` WHERE `id_m_staff` = :id_m_staff LIMIT 1';

    try {
        $connection=Yii::app()->db;
        $command=$connection->createCommand($sql);
        $command->bindParam(':id_m_staff',$id_m_staff, PDO::PARAM_INT);

        $dataReader=$command->execute();
        return true;
    }
    catch( CDbException $Exception ) {
        return false;
    }
}

```

```

public function RawSQL($param)
{
    if($param==""){
        $sql = 'SELECT * FROM `m_staff` WHERE `id_m_staff` = :id_m_staff';
    }else{
        $sql= "";
    }
    $data = $this->findAllBySql($sql);
    return new CActiveDataProvider($this, array(
        'data'=>$data,
        'pagination'=>true,
    ));
}

```

```

public function jsonCBO($param=0)
{
    if($param==0)
    {
        $where = "";
    }else
    {
        $where = '
            `id_m_staff` = :id_m_staff AND
            `nama_lengkap` = :nama_lengkap AND
            `photo` = :photo AND
            `nomor_induk` = :nomor_induk AND
            `password` = :password AND
            `is_walikelas` = :is_walikelas AND
            `is_guru` = :is_guru AND
            `is_executive` = :is_executive AND
            `is_administrator` = :is_administrator AND

```

```

        `uid` = :uid AND
";
        ';
    }
    $sql = "SELECT * FROM `m_staff` ". $where;
    $connection=Yii::app()->db;
    $command=$connection->createCommand($sql);
    if(trim($param)>0){
        $command->bindParam(":id_m_staff", $param ,
PDO::PARAM_INT);
        $command->bindParam(":nama_lengkap", $param , PDO::PARAM_STR);
        $command->bindParam(":photo", $param , PDO::PARAM_STR);
        $command->bindParam(":nomor_induk", $param , PDO::PARAM_STR);
        $command->bindParam(":password", $param , PDO::PARAM_STR);
        $command->bindParam(":is_walikelas", $param , PDO::PARAM_INT);
        $command->bindParam(":is_guru", $param , PDO::PARAM_INT);
        $command->bindParam(":is_executive", $param , PDO::PARAM_INT);
        $command->bindParam(":is_administrator", $param , PDO::PARAM_INT);
        $command->bindParam(":uid", $param , PDO::PARAM_STR);
    }

    return '{"info":' . CJSON::encode($command->query()) . '}';
}

```

```

public function showData()
{
    $sql = 'SELECT * FROM `m_staff`';
    $connection=Yii::app()->db;
    $command=$connection->createCommand($sql);
    $dataReader=$command->queryAll();
    return $dataReader;
}

```

```

public function showDataCBO($id)
{
    $str = "";
    $sql = 'SELECT * FROM `m_staff` ORDER BY `nama_lengkap` asc';
    $connection=Yii::app()->db;
    $command=$connection->createCommand($sql);
    $dataReader=$command->queryAll();
    $str .=";
    foreach ($dataReader as $row){
        if ($row['id_m_staff']==$id){
            $s ='selected="selected"';
        }else{
            $s =";
        }
    }
}

```

```

        $str .= '<option value="" . $row['id_m_staff'] . "' . $s.'>' . $row['nama_lengkap'] . '
('.$row['nomor_induk'].)' . '</option>';
    }
    return $str;
}

```

```

public function showDataCBOWaliKelas($id)
{
    $str = "";
    $sql = 'SELECT * FROM `m_staff` where is_walikelas = true AND id_m_staff NOT
IN(SELECT id_wali_kelas from m_kelas) ORDER BY `nama_lengkap` asc';
    $connection=Yii::app()->db;
    $command=$connection->createCommand($sql);
    $dataReader=$command->queryAll();
    $str .="";
    foreach ($dataReader as $row){
        if ($row['id_m_staff']==$id){
            $s='selected="selected"';
        }else{
            $s="";
        }
        $str .= '<option value="" . $row['id_m_staff'] . "' . $s.'>' . $row['nama_lengkap'] . '
('.$row['nomor_induk'].)' . '</option>';
    }
    return $str;
}

```

```

public function Countdata($param)
{
    $str = "";
    if(trim($param)==""){
        $where = "";
    }else{
        $where = $this->sqlwhere;
    }
    $sql = "SELECT count(`id_m_staff`) as c FROM `m_staff`
" . $where . " ";

    $connection=Yii::app()->db;
    $command=$connection->createCommand($sql);
    if(trim($param)<>""){
        $command->bindParam(":id_m_staff", $param ,
PDO::PARAM_INT);
        $command->bindParam(":nama_lengkap", $param , PDO::PARAM_STR);
        $command->bindParam(":photo", $param , PDO::PARAM_STR);
        $command->bindParam(":nomor_induk", $param , PDO::PARAM_STR);
        $command->bindParam(":password", $param , PDO::PARAM_STR);
    }
}

```



```

        $command->bindParam(":is_walikelas", $param , PDO::PARAM_INT);
        $command->bindParam(":is_guru", $param , PDO::PARAM_INT);
        $command->bindParam(":is_executive", $param , PDO::PARAM_INT);
        $command->bindParam(":is_administrator", $param , PDO::PARAM_INT);
        $command->bindParam(":uid", $param , PDO::PARAM_STR);
    }
    $dataReader=$command->query();
    foreach($dataReader as $row) {
        $results = $row['c'];
    }
    return $results;
}

public function getAll()
{
    $r = array();
    $a = $this->findAll();
    if ($a <> array())
    {
        foreach ($a as $row)
        {
            $r[$row->id_m_staff]=ucwords($row->id_m_staff);
        }
    }
    return $r;
}
//End Model
/* USE FOR POST
$this->id_m_staff = $_POST['id_m_staff'];
$this->nama_lengkap = $_POST['nama_lengkap'];
$this->photo = $_POST['photo'];
$this->nomor_induk = $_POST['nomor_induk'];
$this->password = $_POST['password'];
$this->is_walikelas = $_POST['is_walikelas'];
$this->is_guru = $_POST['is_guru'];
$this->is_executive = $_POST['is_executive'];
$this->is_administrator = $_POST['is_administrator'];
$this->uid = $_POST['uid'];
*/

/* USE FOR XML POST CONTROLLER

public function actionXMLData()
{
    $id = 0;
    $str = '<result>';
    $model = new MStaff;
    $a = "";
    if(!isset($_POST['data'])){exit();}
    $data = preg_replace('#&(?!=[a-z_0-9]+=)#', '&', $_POST['data']);

```

```

$xml = simplexml_load_string($data);
foreach($xml->xpath('/DATAROOT/m_staff/*') as $child)
{
    if($child->getName()=='id_m_staff'){ $model->m_staff_id_m_staff = 0; $id = $child;}
    if($child->getName()=='nama_lengkap'){ $model->m_staff_nama_lengkap = $child;}
    if($child->getName()=='photo'){ $model->m_staff_photo = $child;}
    if($child->getName()=='nomor_induk'){ $model->m_staff_nomor_induk = $child;}
    if($child->getName()=='password'){ $model->m_staff_password = $child;}
    if($child->getName()=='is_walikelas'){ $model->m_staff_is_walikelas = $child;}
    if($child->getName()=='is_guru'){ $model->m_staff_is_guru = $child;}
    if($child->getName()=='is_executive'){ $model->m_staff_is_executive = $child;}
    if($child->getName()=='is_administrator'){ $model->m_staff_is_administrator = $child;}
    if($child->getName()=='uid'){ $model->m_staff_uid = $child;}
    if($child->getName()=='save'){
        if(trim($model->m_staff_SOMECOLUMNNAME)<>"") {
            if($model->SaveData(0)){
                $str .= '<id value="" . $id . "" />';
            }else{
                $str .="";
            }
        }else{
            $str .= '<id value="" . $id . "" />';
        }
    }
}
echo $str .='</result>';

}

*/
}

```

<?php

```

class StaffController extends Controller
{
    public function actionAdd()
    {
        $this->render('add');
    }

    public function actionclearsearch()
    {
        unset(Yii::app()->session['search']);
        $this->redirect('index.php?r=staff/index');
    }
}

```

```

public function actionIndex()
{
    $crit = array();
    $sqlstr = "";
    if(isset($_POST['search'])){
        Yii::app()->session['search'] = $_POST['search'];
    }
    (isset(Yii::app()->session['search'])) ? $search = Yii::app()->session['search'] : $search = "";

    if(!empty($search))
    {
        $sqlstr .= " AND (nomor_induk LIKE :search OR nama_lengkap LIKE :search) ";
        $crit[':search']="%"$search%";
    }

    //$criteria->select= "t.*,s_users.fullname as fn";
    //$criteria->with = array('addBy');
    //$criteria->join = 'LEFT JOIN s_users ON s_users.id_users= t.add_by';
    //$x = Param::model()->findByPk(1);
    //echo $x->paramValue;
    $criteria = new CDbCriteria();
    //$sqlstr .= " AND loginattempt > " . $x->paramValue . " ";
    $criteria->with = array('mKelases');
    $criteria->condition = ' 1 = 1 ' . $sqlstr;
    $criteria->order = ' t.nama_lengkap ASC';
    //$criteria->params = array (':id'=>$id);
    $criteria->params = $crit;
    //print_r($criteria->params . '<br/>');
    //print_r( $criteria);
    //exit();
    $item_count = MStaff::model()->count($criteria);
    $pages = new CPagination($item_count);
    $pages->setPageSize(Yii::app()->params['listPerPage']);
    $pages->applyLimit($criteria); // the trick is here!
    $this->render('index',array(
        'title'=>"Staff",
        'model'=> MStaff::model()->findAll($criteria), // must be the same as $item_count
        'item_count'=>$item_count,
        'page_size'=>Yii::app()->params['listPerPage'],
        'items_count'=>$item_count,
        'pages'=>$pages,
    ));
}

```

```

public function actionaddedit()
{
    if(isset($_GET['id'])){ $id=$_GET['id'];}else{$id=0;}
    $m = MStaff::model()->findByPk($id);
    if($m<>array()){ $model = $m;}else{$model= new MStaff;}
}

```

```

        if(isset($_POST['MStaff']))
        {
            $model->attributes = $_POST['MStaff'];

            if($model->attributes['is_walikelas']=='on'){ $model->is_walikelas=1;}else{ $model->is_walikelas=0;}
            if($model->attributes['is_guru']=='on'){ $model->is_guru=1;}else{ $model->is_guru=0;}
            if($model->attributes['is_administrator']=='on'){ $model->is_administrator=1;}else{ $model->is_administrator=0;}
            if($model->attributes['is_executive']=='on'){ $model->is_executive=1;}else{ $model->is_executive=0;}

            if($id==0){
                if($_POST['password']){
                    $model->password = md5('yiidefault' . $_POST['password']);
                }else{
                    echo '<div class="alert alert-danger" role="alert">';
                    echo '<p> <i class="c-icon cil-x-circle text-danger"></i> &nbsp; Kolom password harus di isi.</p></div>';
                    exit();
                }
            }else{
                $model->password = md5('yiidefault' . $_POST['password']);
            }
            //exit();

            if($model->save())
            {
                echo '<div class="alert alert-success" role="alert">';
                echo '<p><i class="c-icon cil-check text-success"></i>&nbsp; Data telah berhasil di simpan.</p>';
                echo '</div>';
            };

            }else{
                echo '<div class="alert alert-danger" role="alert">';
                foreach ($model->getErrors() as $key => $value) {
                    echo '<p> <i class="c-icon cil-x-circle text-danger"></i> &nbsp; ' . $value[0] . '</p>';
                }
                echo '</div>';
            }
        }else{
            //$this->layout = 'admin';
            $this->render('addedit',array('model'=>$model));
        }
    }
}

```

```

public function actiondelete()
{
    if(isset($_GET['id']))
    {
        $model = new MStaff();
        if($_GET['id']==1){
            echo false;
            exit();
        }
        if($model->deleteByPk((int)$_GET['id'])){echo true;}else{echo false;}
    }else{
        echo false;
    }
}

// Uncomment the following methods and override them if needed
/*
public function filters()
{
    // return the filter configuration for this controller, e.g.:
    return array(
        'inlineFilterName',
        array(
            'class'=>'path.to.FilterClass',
            'propertyName'=>'propertyValue',
        ),
    );
}

public function actions()
{
    // return external action classes, e.g.:
    return array(
        'action1'=>'path.to.ActionClass',
        'action2'=>array(
            'class'=>'path.to.AnotherActionClass',
            'propertyName'=>'propertyValue',
        ),
    );
}
*/
}

```