

**SISTEM INFORMASI JALUR PENDAKIAN GUNUNG
DI INDONESIA BERBASIS *WEB***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan
Jenjang Strata Satu (S1)
Pada Program Studi Sistem Informasi**

Oleh

**Mochamad Regi Gumilang
(352031004)**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI
2022**

LEMBAR PENGESAHAN

**SISTEM INFORMASI JALUR PENDAKIAN GUNUNG
DI INDONESIA BERBASIS *WEB***

Oleh
Mochamad Regi Gumilang
352031004

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

SARJANA SISTEM INFORMASI

Pada
**PROGRAM STUDI SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI**

Bandung, Januari 2022
Disahkan oleh

Ketua Program Studi,

Dosen Pembimbing,

Moch. Ali Ramdhani, S.T., M.Kom
NIDN. 0403097701

Chairuddin, Ir., M.T., M.M., Dr.
NIDN. 0426076901

LEMBAR PERSETUJUAN REVISI

**SISTEM INFORMASI JALUR PENDAKIAN GUNUNG
DI INDONESIA BERBASIS *WEB***

Oleh
Mochamad Regi Gumilang
352031004

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

Bandung, Januari 2022
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Chairuddin, Ir., M.T., M.M., Dr.	Pembimbing	
2.	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 1	
3.	Haryoso Wicaksono, S.SI., M.M., M.KOM.	Penguji 2	

Mengetahui,
Ketua Program Studi Sistem Informasi

Moch. Ali Ramdhani, S.T., M.Kom
NIDN. 0403097701

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah skripsi 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, Januari 2022
Yang membuat pernyataan

Mochamad Regi Gumilang
352031004

ABSTRAK

SISTEM INFORMASI JALUR PENDAKIAN GUNUNG DI INDONESIA BERBASIS *WEB*

Oleh :

Mochamad Regi Gumilang
352031004

Sistem Informasi jalur pendakian gunung di indonesia membantu para calon pendaki memperluas informasi mengenai jalur pendakian. Bagi para pendaki pemula yang awam dengan informasi seputar jalur pendakian gunung yang akan dituju diperlukan informasi tentang jalur pendakian, hal ini dikarenakan masih banyak orang yang tidak tahu kondisi langsung di area pendakian. Oleh sebab itu dibutuhkan suatu perancangan sistem informasi jalur pendakian gunung di Indonesia menggunakan PHP berbasis web yang dapat diakses semua pengguna khususnya para pendaki. Web adalah kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (Uniform Resource Locator). PHP adalah bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. dari penelitian yang dilakukan dapat disimpulkan bahwa sistem informasi ini dapat mempermudah pendaki untuk mendapatkan informasi seputar jalur pendakian, denah pendakian dan mengetahui jalur pendakian yang akan ditempuh.

Kata kunci : sistem informasi, jalur pendakian, pegunungan, web, pendaki,.

ABSTRACT

MOUNTAINEERING PATH INFORMATION SYSTEM IN INDONESIA BASED ON WEB

By :

Mochamad Regi Gumilang
352031004

Information systems for mountain climbing routes in Indonesia help prospective climbers expand information about hiking trails. For beginner climbers who are unfamiliar with information about the mountain climbing route to be addressed, information about the climbing route is needed, this is because there are still many people who do not know the immediate conditions in the climbing area. Therefore, it is necessary to design an information system for mountain climbing routes in Indonesia using web-based PHP that can be accessed by all users, especially climbers. The web is a collection of web pages that have been published on the internet and have a domain/URL (Uniform Resource Locator). PHP is a scripting programming language designed for building web applications. From the research conducted, it can be concluded that this information system can make it easier for climbers to get information about climbing routes, climbing plans and knowing the climbing routes to be taken.

Keywords: information system, hiking trails, mountains, web, climbers.

KATA PENGANTAR

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji syukur kami panjatkan kehadirat Alloh SWT, yang atas rahmat-Nya dapat menyelesaikan penyusunan laporan yang berjudul **“SISTEM INFORMASI JALUR PENDAKIAN GUNUNG DI INDONESIA BERBASIS *WEB*”**. Laporan ini dimaksudkan untuk memenuhi persyaratan kelulusan pada Program Studi Sistem Informasi di STMIK-IM Bandung.

Dalam Kesempatan ini, penulis juga mengucapkan terima kasih kepada orang tua, istri serta semua pihak yang telah memberikan *support* dalam menyelesaikan penelitian ini. Penulis menyadari bahwa dalam menyajikan penelitian ini masih terdapat kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun dan membantu penulis guna pembelajaran dikemudian hari.

Demikian penulis sampaikan, dengan harapan agar penelitian ini dapat memberikan manfaat bagi semua pihak. Akhir kata, penulis berharap agar penelitian ini mendapat berkah dan ridho dari Allah SWT.

Wasalamu'alaikum Warrahmatullahi Wabarrakatuh

Bandung, Januari 2022

Mochamad Regi Gumilang

UCAPAN TERIMA KASIH

Pada kesempatan kali ini, penulis menyampaikan rasa berterima kasih kepada rekan – rekan, pembimbing, serta pihak yang ikut serta membantu penulis dalam penyusunan laporan ini, antara lain :

1. Bapak Dr. Chairuddin, Ir., M.M., M.T. selaku Ketua STMIK-IM Bandung dan pembimbing yang telah memberikan ilmu dan masukan dalam penelitian.
2. Bapak Moch. Ali Ramdhani, S.T., M.Kom. selaku selaku Ketua Program Studi Sistem Informasi.
3. Istri dan anak tercinta yang selalu memberikan dukungan penuh untuk penulis.
4. Bapak & Ibu dosen yang telah memberikan penulis ilmu selama penulis melakukan kegiatan belajar mengajar di STMIK-IM Bandung.
5. Rekan – rekan di STMIK-IM Bandung.

Bandung, Januari 2022

Mochamad Regi Gumilang

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN REVISI.....	ii
SURAT PERNYATAAN	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
UCAPAN TERIMA KASIH.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Metode Penelitian	4
1.5.1 Teknik Pengumpulan Data.....	4
1.5.2 Metode Pengembangan Perangkat Lunak.....	4
1.6 Sistem Penulisan	6
BAB II.....	8
LANDASAN TEORI.....	8
2.1 Pengertian Sistem.....	8
2.1.1 Komponen Sistem	8
2.1.2 Klasifikasi Sistem	10
2.2 Pengertian Informasi	14
2.3 HTML	15
2.4 PHP	16
2.5 <i>Framework</i>	18
2.5.1 Fungsi <i>Framework</i>	18

2.6 MySQL	20
2.6.1 SQL	22
2.7 UML.....	24
2.7.1 Building blocks	25
2.7.2 Diagram.....	29
2.8 Data Flow Diagram (DFD)	39
2.9 Entity Relationship Diagram (ERD)	44
2.10 Flow Chart	46
2.11 <i>Website</i>	48
2.12 Gunung.....	49
2.12.1 Jenis Pegunungan di Indonesia	50
BAB III	52
ANALISA MASALAH DAN PERANCANGAN PROGRAM.....	52
3.1 Analisa	52
3.1.1 Analisa Masalah.....	52
3.1.2 Analisa Sistem Berjalan	52
3.1.3. Analisa Kebutuhan Non Fungsional	54
3.1.3 Analisa Kebutuhan Fungsional	56
3.2 Perancangan Sistem	58
3.2.1 Desain Sistem dengan <i>Unified Modeling Language</i> (UML)	58
3.2.2 Desain Database Sistem	71
3.2.3 Desain <i>Interface</i> Sistem	75
BAB IV	83
IMPLEMENTASI DAN UJI COBA	83
4.1 Implementasi.....	83
4.1.1 Implementasi Perangkat Keras.....	83
4.1.2 Implementasi Perangkat Lunak.....	83
4.1.3 Implementasi <i>Database</i>	84
4.1.4 Implementasi Antarmuka.....	86
BAB V	101
KESIMPULAN DAN SARAN.....	101
5.1 Kesimpulan	101
5.2 Saran	102
DAFTAR PUSTAKA	103

LAMPIRAN – LAMPIRAN	105
1. Login_admin.php	105
2. Admin_datagunung.php.....	110
3. Ambil_komentar.php	118

DAFTAR TABEL

Tabel 2 . 1 Simbol <i>Use case</i> (Rosa dan Shalahuddin, 2015:156)	30
Tabel 2 . 2 Simbol <i>Class Diagram</i> (Rosa dan shalahuddin, 2016:146).....	32
Tabel 2 . 3 Simbol <i>Sequence Diagram</i> (Rosa dan shalahuddin, 2016:165).....	34
Tabel 2 . 4 Simbol <i>State Chart Diagram</i> (Rosa dan Salahuddin, 2016:163).....	36
Tabel 2 . 5 Simbol <i>Activity diagram</i> (Rosa dan salahuddin, 2016:162)	37
Tabel 2 . 6 Notasi DFD (Edward Yourdon dan Tom DeMarco,1980)	41
Tabel 2 . 7 Simbol pada ERD (Peter Chen, 1970)	45
Tabel 2 . 8 Simbol pada <i>Flow chart</i> (Peter Chen, 1970).....	47
Tabel 3 . 1 Penjelasan aktor	58
Tabel 3 . 2 Penjelasan <i>Use case</i> diagram	59
Tabel 3 . 3 Penjelasan <i>Activity</i> diagram	62
Tabel 3 . 4 Penjelasan <i>State chart</i> diagram	70
Tabel 3 . 5 Login Admin	73
Tabel 3 . 6 Komentar.....	73
Tabel 3 . 7 Daftar gunung	73
Tabel 4 . 1 Pengujian Login	94
Tabel 4 . 2 Pengujian daftar gunung	96
Tabel 4 . 3 Pengujian data komentar.....	98
Tabel 4 . 4 Pengujian daftar gunung	98
Tabel 4 . 5 Pengujian komentar.....	99

DAFTAR GAMBAR

Gambar 1 . 1 Model <i>Waterfall</i> (Juniardi Dermawan & Sistem,2017)	5
Gambar 2. 1 Sistem tertutup (wahyudin,2009)	12
Gambar 2. 2 Sistem relatif tertutup (wahyudin,2009)	12
Gambar 2. 3 Sistem terbuka (wahyudin,2009).....	12
Gambar 2. 4 Sistem pengendalian umpan balik sebuah sistem (wahyudin,2009) 13	
Gambar 2. 5 Contoh DFD yang dikembangkan (Chris gane dan Trish sarson,1979)	39
Gambar 3 . 1 Flowchart Analisa Sistem berjalan.....	53
Gambar 3 . 2 Use case diagram.....	59
Gambar 3 . 3 Activity Diagram.....	61
Gambar 3 . 4 Sequence Admin Login.....	63
Gambar 3 . 5 Sequence Admin Input informasi jalur pegunungan.....	64
Gambar 3 . 6 Sequence Admin Input tips pendakian.....	65
Gambar 3 . 7 Sequence Admin menerima komentar	66
Gambar 3 . 8 Sequence User melihat menu informasi jalur pegunungan.....	67
Gambar 3 . 9 Sequence User melihat tips pendakian.....	67
Gambar 3 . 10 Sequence User melihat dan mengisi kolom berbagi cerita dan komentar.....	68
Gambar 3 . 11 <i>State chart</i> diagram	69
Gambar 3 . 12 ERD Sistem Informasi Jalur pegunungan	72
Gambar 3 . 13 Desain menu login.....	75
Gambar 3 . 14 Desain tambah daftar gunung.....	76
Gambar 3 . 15 Desain tambah daftar gunung.....	77
Gambar 3 . 16 Desain edit dan hapus daftar gunung	78
Gambar 3 . 17 Desain hapus komentar	79
Gambar 3 . 18 Desain beranda pada <i>user</i>	79
Gambar 3 . 19 Desain daftar gunung pada <i>user</i>	80
Gambar 3 . 20 Desain detail dan lokasi pada menu daftar gunung <i>user</i>	81
Gambar 3 . 21 Desain berbagi cerita <i>user</i>	82
Gambar 3 . 22 Desain tips pendakian <i>user</i>	82
Gambar 4 . 1 Login admin	84
Gambar 4 . 2 Daftar.....	85
Gambar 4 . 3 Komentar.....	86
Gambar 4 . 4 Halaman Login.....	87
Gambar 4 . 5 Tambah daftar gunung pada admin.....	88
Gambar 4 . 6 edit dan hapus daftar gunung pada admin.....	88

Gambar 4 . 7 menghapus komentar pada admin	89
Gambar 4 . 8 tampilan beranda <i>user</i>	89
Gambar 4 . 9 daftar gunung pada <i>user</i>	90
Gambar 4 . 10 peta titik kordinat detail dan lokasi pada daftar gunung <i>user</i>	91
Gambar 4 . 11 Informasi detail dan lokasi pada daftar gunung <i>user</i>	91
Gambar 4 . 12 Informasi detail dan lokasi pada daftar gunung <i>user</i>	92
Gambar 4 . 13 berbagi cerita <i>user</i>	92
Gambar 4 . 14 berbagi komentar <i>user</i>	93
Gambar 4 . 15 tips pendakian pada <i>user</i>	93

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah Negara yang kaya akan alamnya. termasuk didalamnya yaitu pegunungan. Pegunungan merupakan kumpulan atau gugusan beberapa gunung besar dan juga kecil yang memanjang dan sambung menyambung menjadi satu antara satu dengan yang lainnya. Pegunungan atau disebut juga Barisan dan Banjaran merupakan suatu area geografis berupa rentetan dari beberapa gunung-gunung atau bukit yang terkait secara geologis yang membentuk suatu deret atau bentangan. Pegunungan merupakan salah satu bentang alam yang sudah banyak kita temui. Di Indonesia sendiri, pegunungan ini jumlahnya ada banyak sekali. Pegunungan yang ada di Indonesia hampir berada di setiap pulau di Indonesia. Selain pegunungan membawa potensi kekayaan alam yang besar , pegunungan juga memberikan pemandangan yang sangat indah. Hal inilah yang menyebabkan pegunungan seringkali dijadikan sebagai tempat wisata bagi orang- orang yang tinggal di kota dan ingin merasakan sejuknya alam Indonesia. Tidak mengherankan bahwa daerah pegunungan ini seringkali dibanjiri wisatawan, baik pada musim liburan maupun setiap akhir pekan tiba.

Pegunungan di Indonesia menjadi salahsatu wisata alam yang sangat diminati oleh kebanyakan orang terutama yang punya hobi mendaki. Bagi para pendaki pemula yang awam dengan informasi seputar jalur pendakian gunung yang akan dituju diperlukan informasi tentang jalur pendakian, hal ini dikarenakan masih banyak orang yang tidak tahu kondisi langsung di area pendakian. Akibatnya terdapat beberapa kasus kehilangan, terjatuh, hipotermia hingga kecelakaan ketika mendaki. Dengan masalah yang dihadapi, perlu adanya sebuah sistem informasi berbasis *web* mengenai jalur pendakian yang akan membantu para calon pendaki memperluas informasi mengenai jalur pendakian.

Oleh karena itu, pada tugas akhir ini berdasarkan permasalahan tersebut akan dikaji dan dilakukan system informasi berbasis *web* yang dapat diterapkan pada jalur pendakian di Indonesia dan dituangkan dalam bentuk laporan penelitian yang berjudul “SISTEM INFORMASI JALUR PENDAKIAN GUNUNG DI INDONESIA BERBASIS *WEB*”.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat diidentifikasi masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun Sistem Informasi berbasis *web* untuk jalur pendakian pegunungan di Indonesia?

2. Bagaimana menginformasikan kepada para calon pendaki dapat mengetahui kondisi jalur pendakian yang akan ditempuh?

1.3 Tujuan Penelitian

Mengimplementasikan system informasi jalur pendakian menggunakan aplikasi berbasis *web*. Serta menjawab identifikasi masalah diatas yakni:

1. Merancang aplikasi dengan Bahasa pemograman *web* yaitu PHP dan database MYSQL untuk penerapan system informasi jalur pendakian berbasis *web*.
2. Merancang sistem untuk membuat informasi seputar jalur pendakian, kondisi jalur pendakian, denah jalur pendakian dan informasi pegunungan.

1.4 Batasan Masalah

Dalam pembuatan Tugas Akhir ini penulis membatasi permasalahan sebagai berikut:

1. Aplikasi *web* ini membantu dalam proses informasi seputar jalur pendakian, denah pendakian dan informasi pegunungan.
2. Aplikasi jalur pendakian pegunungan ini membantu para pendaki untuk mengetahui kondisi jalur pendakian sehingga pendaki dapat mempersiapkan keperluan yang akan dibawa.

1.5 Metode Penelitian

Beberapa metode penelitian yang penulis gunakan seperti dibawah:

1.5.1 Teknik Pengumpulan Data

Dalam penelitian ini metode yang digunakan adalah dengan menggunakan metode action program dengan teknik pengumpulan data sebagai berikut:

1. Observasi

Observasi yaitu pengumpulan data dan informasi yang dilakukan dengan cara mengamati langsung ke objek yang akan diteliti.

2. Wawancara.

Wawancara yaitu pengumpulan data dengan cara melakukan tanya jawab dengan pihak-pihak terkait.

3. Studi perpustakaan

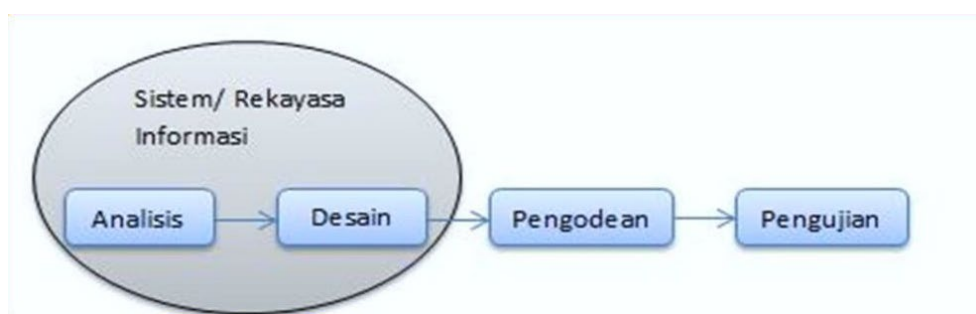
Melakukan pencarian melalui buku di perpustakaan, referensi dari internet yang secara langsung terkait dengan permasalahan yang ada.

1.5.2 Metode Pengembangan Perangkat Lunak

Metode untuk merancang sistem yang digunakan penulis dalam penelitian ini yaitu metode waterfall. Air terjun (waterfall) sering juga disebut model

sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support) (Rosa dan Shalahuddin, 29:2013).

Berikut adalah gambar model air terjun:



Gambar 1 . 1 Model *Waterfall* (Juniardi Dermawan & Sistem,2017)

A. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

B. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus 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 perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

C. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

D. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik 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.

E. Pendukung (Support) atau Pemeliharaan (Maintenance)

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.

1.6 Sistem Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan.:

BAB I: PENDAHULUAN

Berisi latar belakang, identifikasi dan batasan masalah, manfaat, metodologi penulisan dan sistematika penulisan.

BAB II: LANDASAN TEORI

Pada bab ini memuat landasan teori yang berisi Pustaka yang berhubungan dengan masalah-masalah yang dibahas dalam penelitian.

BAB III: ANALISA DAN PERANCANGAN

Pada bab ini menjelaskan mengenai masalah yang di jabarkan, sesuai dengan tujuan penelitian pada BAB 1 dan dengan metode penelitian pada BAB 1 yaitu metode waterfall.

BAB IV: IMPLEMENTASI DAN PENGUJIAN

Pada bab ini menjelaskan kesimpulan yang didapatkan setelah melakukan penelitian serta sara-saran.

BAB V: KESIMPULAN DAN SARAN

Pada bab ini menjelaskan kesimpulan yang didapatkan setelah melakukan penelitian serta sara-saran.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri atas komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika sering kali bisa dibuat.

Sistem merupakan sekumpulan elemen, himpunan dari suatu unsur, komponen fungsional yang saling berhubungan dan berinteraksi satu sama lain untuk mencapai tujuan yang diharapkan. Menurut Fatansyah (2015:11) bahwa Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.

2.1.1 Komponen Sistem

Sistem adalah suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama

lain dan terpadu, sistem terdiri atas objek-objek atau unsur-unsur atau komponen-komponen yang berkaitan dan berhubungan satu sama lainnya, Sutabri (2012:3).

Terdapat enam blok yang saling berinteraksi satu dengan lainnya yang membentuk satu kesatuan untuk mencapai sasaran yaitu:

1. *Input*

Input merupakan masukan yang bisa berupa fakta atau data apapun yang bisa mendukung tercapainya tujuan suatu sistem. *Input* merupakan bagian dari sistem yang bertugas untuk menerima data masukan yang digunakan sebagai komponen penggerak/ menangkap data/ pemberi tenaga dimana sistem itu dioperasikan atau yang akan dimasukan yang berupa dookumen-dokumen dasar.

2. *Proses*

Proses merupakan tahapan pengelolaan *input* sistem sehingga dapat menjadi suatu informasi yang bermanfaat sehingga tujuan sistem dapat tercapai.

3. *Output*

Output merupakan hasil yang didapatkan dari sebuah sistem. Dimana *output* ini dapat berupa informasi penting ataupun *output* lainnya yang dapat menggambarkan kesuksesan pencapaian tujuan dari suatu sistem.

4. *Kendala*

Kendala merupakan masalah yang dapat terjadi saat sistem dijalankan, sehingga perlu adanya solusi untuk menyelesaikan masalah ini. Komponen yang berisiko aturan atau batas – batas yang berlaku. Akan membuat tujuan menjadi lebih bermanfaat. Adanya suatu kendala atau batasan yang jelas, akan mampu mengidentifikasi apa yang harus diantisipasi dalam mencapai tujuan sistem.

5. Kontrol

Kontrol merupakan pemantauan sistem yang dilakukan agar tidak ada masalah yang terjadi.

6. Umpan balik

Umpan balik merupakan respon yang diberikan oleh sistem ketika sedang aktif. Komponen yang memberikan respon atau berjalannya suatu sistem. Kegiatan tersebut seperti perbaikan atau pemeliharaan sistem.

2.1.2 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya. (Tata Sutabri, 2012)

1. Sistem Abstrak dan Sistem Fisik

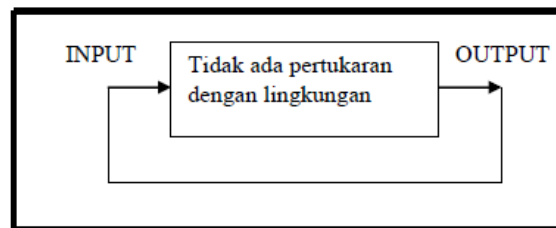
- Sistem Abstrak (*abstract system*), adalah sistem yang berisi gagasan atau konsep-konsep. Contohnya adalah sistem teologi atau keagamaan yaitu suatu sistem yang mengatur hubungan antara manusia dengan Tuhannya, antara alam dan Allah sebagai pencipta alam semesta.
- Sistem Fisik (*physical system*), adalah sistem yang secara fisik dapat dilihat. Contohnya sistem Komputer, sistem transportasi, sistem perguruan tinggi, sistem akuntansi dan lain-lain.

2. Sistem Tertentu (*deterministic*) dan tidak tertentu (*Probabilistik*)

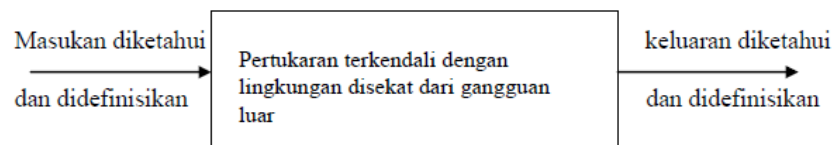
- Sistem *Deterministik* (*deterministic system*) adalah sistem yang operasinya dapat diprediksi secara tepat. Contohnya adalah sistem komputer. Sistem ini kita dapat memberikan input sesuai dengan tujuan *output* tertentu.
- Sistem Probabilistik (*probabilistic system*), adalah sistem yang tidak dapat diprediksi atau diramal dengan pasti karena mengandung unsur probabilitas atau kemungkinan-kemungkinan. Contohnya adalah, sistem evapotranspirasi, sistem serapan hara, sistem fotosintesis dan lain-lain.

3. Sistem Tertutup dan Sistem Terbuka

- Sistem Tertutup (*closed system*), adalah sistem yang tidak berhubungan dengan lingkungan dan tidak dipengaruhi oleh lingkungannya (bekerja secara otomatis).

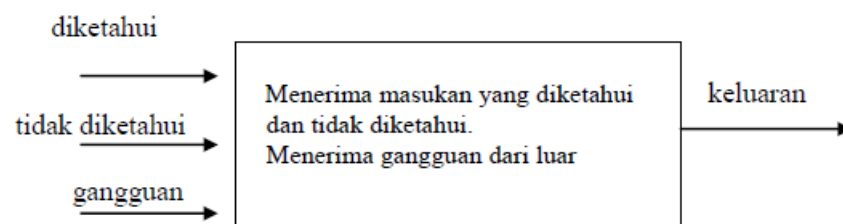


Gambar 2. 1 Sistem tertutup
(wahyudin,2009)

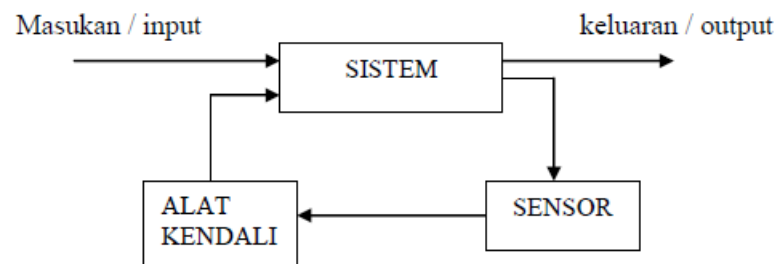


Gambar 2. 2 Sistem relatif tertutup
(wahyudin,2009)

- Sistem Terbuka (*open system*), adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Ciri-cirinya adalah, sistem menerima masukan yang diketahui, yang bersifat acak, maupun gangguan.



Gambar 2. 3 Sistem terbuka (wahyudin,2009)



Gambar 2. 4 Sistem pengendalian umpan balik sebuah sistem (wahyudin,2009)

4. Sistem Alamiah dan Sistem Buatan Manusia

- Sistem Alamiah (*natural system*), adalah sistem yang terjadi secara alamiah tanpa campur tangan manusia, contohnya sistem tata Surya.
- Sistem Buatan Manusia (*human made system*), adalah sistem yang dibuat oleh manusia, contohnya sistem komputer, sistem mobil, sistem telekomunikasi.

5. Sistem Sederhana dan Sistem Kompleks

- Sistem Sederhana adalah sistem yang tidak rumit atau sistem dengan tingkat kerumitan rendah, contohnya sistem sepeda, sistem mesin ketik, sistem infiltrasi tanah.
- Sistem Kompleks adalah sistem yang rumit, contohnya sistem otak manusia, sistem komputer, sistem keseimbangan hara esensial dalam tanah dan lain-lain. Sistem informasi dapat tergolong sebagai sistem kompleks atau sederhana tergantung pada implementasinya.

2.2 Pengertian Informasi

Informasi merupakan hasil pengolahan data, sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung pada saat itu juga atau secara tidak langsung pada saat mendatang, (Edhy Sutanta, 2011).

Menurut Jogiyanto H.M. (2001), informasi adalah hasil pengolahan data menjadi bentuk yang lebih bermanfaat bagi si penerima dan menggambarkan peristiwa nyata yang dapat digunakan dalam mengambil keputusan.

Dalam proses pembangunan sistem informasi, terdapat beberapa komponen yang perlu diperhatikan dalam mengambil keputusan untuk bisnis dan strategi pemasaran. Sistem Informasi sendiri terdiri atas komputer, manusia, fakta, instruksi, dan kumpulan prosedur yang dapat dikategorikan sebagai berikut.

1. Sistem Informasi Manajemen

Sistem informasi manajemen adalah sebuah sistem perencanaan yang melibatkan bagian internal perusahaan yang meliputi pemanfaatan teknologi, prosedur, dan interaksi manusia untuk memecahkan permasalahan bisnis seperti layanan, biaya produksi, atau penentuan strategi bisnis yang tepat.

2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah salah satu bagian dalam sistem informasi berbasis komputer yang berfungsi untuk mengambil sebuah keputusan yang tepat bagi perusahaan, organisasi, atau instansi terkait.

3. Sistem Informasi Eksekutif

Executive Information System (EIS) merupakan salah satu jenis dari sistem informasi manajemen untuk memudahkan dalam mengambil kebijakan yang dibutuhkan oleh eksekutif senior dengan menampilkan akses informasi yang tepat dan relevan.

4. Sistem Pemrosesan Transaksi

Sistem Pemrosesan Transaksi adalah bagian dari sistem informasi yang memiliki tugas untuk menjalankan dan mencatat setiap kebutuhan transaksi secara berkelanjutan (kontinu) yang diperlukan dalam bisnis. Tujuan utamanya adalah untuk menjawab berbagai pertanyaan rutin dan memonitoring setiap transaksi yang dilakukan oleh organisasi terkait.

2.3 HTML

HTML atau *Hypertext Markup Language* adalah bahasa markup yang digunakan untuk membuat struktur halaman *website* agar dapat ditampilkan pada *web browser*. HTML dapat dianalogikan sebagai pondasi awal dalam menyusun

kerangka halaman *web* secara terstruktur sebelum membahas terkait tampilan desain dan sisi fungsionalitas, (Muhammad Robith Adani, 2021).

Berikut kerangka inti yang membentuk struktur bahasa markup, yaitu:

1. *Tag*

Tag merupakan perintah atau awalan instruksi yang nantinya akan dibaca pada *web browser*. Misalnya adalah penggunaan *tag* ``, yang berfungsi untuk memberikan ketebalan pada teks paragraf.

2. *Element*

Element merupakan proses keseluruhan kode yang dimulai dari *tag* pembuka (`< >`) dan diakhiri dengan *tag* penutup (`</>`), sebagai contoh *element* `<p>Ini adalah website saya</p>`.

3. *Attribute*

Attribute merupakan informasi atau perintah tambahan yang dimasukkan ke dalam sebuah *elemen*. Misalnya saja ``, sehingga atributnya adalah *class*, *src* dan *alt image*.

2.4 PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman yang berjalan pada *serverside scripting* dan bersifat *open source* (sumber terbuka). Untuk

penggunaan dari bahasa ini sering digunakan untuk pembuatan aplikasi berbasis *website* yang berjalan secara dinamis, sehingga dapat terintegrasi dengan basis data (database), (Muhammad Robith Adani, 2021).

PHP banyak digunakan untuk pembuatan *website* untuk kebutuhan *ecommerce*, sistem informasi, maupun *landing page*. Karena PHP tergolong dalam bahasa pemrograman berbasis *server-side*, maka *script* yang digunakan akan diproses oleh *server*. Jenis *server* yang sering dipakai adalah *Apache*, *Nginx*, dan *LiteSpeed*.

PHP memiliki beberapa keunggulan, yaitu:

1. Terintegrasi dengan Sistem *Database*

PHP dapat terintegrasi secara langsung dengan *server* serta sistem basis data. Dimana, tentu PHP merupakan bahasa yang bersifat *open source* dan sangat mudah untuk terkoneksi dengan *database* dari *MySQL*, *Postgre*, *Oracle*, dan lain sebagainya.

2. Mudah Dipelajari

PHP bukan bahasa pemrograman yang eksklusif karena bisa dipelajari dari mana saja. Bahasa ini juga masuk ke kelas *entry level* sehingga orang awam sekali pun bisa mempelajarinya. Apalagi banyak sekali tutorial gratis yang bisa didapatkan di internet.

3. Pengembangan Produk Berjalan Cepat

Dibandingkan dengan bahasa pemrograman lainnya, PHP memiliki kemampuan untuk membuat program atau *website* berjalan dengan lebih mudah dan tidak perlu membuat semua dari awal. Ada beberapa *library* yang dipakai untuk membuat program secara cepat.

4. Program yang Ringkas

PHP dikenal sebagai program yang cukup ringkas dalam proses pengembangannya. dimulai dari proses unduh sampai instalasi berjalan dengan cepat. Selain itu, dalam proses pengembangan dan mencoba *website* tidak butuh waktu kompilasi yang lama.

2.5 Framework

Framework adalah kerangka kerja untuk mengembangkan aplikasi berbasis *website* maupun desktop. Kerangka kerja disini sangat membantu developer dalam menuliskan sebuah dengan lebih terstruktur dan tersusun rapi. Kerangka kerja diciptakan untuk mempermudah kinerja dari programmer. Sehingga, seorang programmer tidak perlu untuk menuliskan kode secara berulang – ulang, (Muhammad Robith Adani, 2021)

2.5.1 Fungsi Framework

Proses pengerjaan aplikasi dapat dilakukan dengan menggunakan *framework* yang tepat dan sesuai dengan kebutuhan project. Berikut merupakan beberapa fungsi kerangka kerja dalam *web development*.

1. Kode program lebih terstruktur

Fungsi *framework* yang utama adalah membuat *source code* menjadi lebih terstruktur. Program yang dibuat akan dimasukkan ke dalam setiap komponen sesuai dengan fungsi nya masing – masing.

2. Membantu kinerja dari developer

Fungsi yang kedua adalah membantu kinerja dari developer sendiri. *Framework* diciptakan untuk memudahkan kinerja dari developer dalam segi efisiensi waktu serta *resource* yang dibutuhkan. Selain itu, apabila dikerjakan oleh beberapa tim maka *framework* akan sangat membantu efektivitas kerja dan sinkronisasi. Setiap dapat langsung memahami dan mengerjakan tugas masing – masing tanpa perlu mengecek satu persatu komponen baik dari segi *back end* maupun *front end*.

3. Meningkatkan keamanan perangkat lunak / *website*

Selain membantu kinerja dari developer, dari segi keamanan *website* maupun *software* juga akan meningkat. Karena dalam *framework* telah diidentifikasi oleh beberapa developer ahli dari berbagai negara.

4. Pemeliharaan dan dokumentasi dari *website* lebih mudah

Dengan melakukan *maintenance*, pengguna dapat merubah versi *website* tersebut dan menambahkan beberapa fitur dengan lebih mudah dan aman. Sehingga, ketika ada perbaikan pada *website* maka terdapat notifikasi atau pesan bagi pengguna bahwa *website* masih dalam perbaikan.

5. Mempercepat dalam proses pembuatan *website*

Fungsi terakhir adalah untuk mempercepat dalam proses pembuatan *website*. Dalam hal ini, bukan berarti pembuatan *website* yang baik dikerjakan dengan cepat saja. Tetapi, proses pembuatan dapat dilakukan dengan lebih cepat dan menghasilkan produk yang berkualitas dan sesuai dengan kebutuhan *customer*. Developer dapat mengembangkan aplikasi dengan menggunakan komponen – komponen yang telah tersedia dalam *framework*. Sehingga, tidak perlu untuk menyusun ulang dari awal kode program.

2.6 MySQL

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah

MySQL *FreeSoftware* yang berada dibawah Lisensi GNU/GPL (*General Public License*), (Haris Saputro, 2003).

Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu Bahasa permintaan yang bernama SQL (*Structured Query Language*).

Database MySQL memiliki beberapa kelebihan dibanding database lain, diantaranya:

1. MySQL merupakan *Database Management System* (DBMS)
2. MySQL sebagai *Relation Database Management System* (RDBMS) atau disebut dengan *database relational*
3. MySQL Merupakan sebuah *database server* yang *free*, artinya pengguna bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya
4. MySQL merupakan sebuah *database client*
5. MySQL mampu menerima *query* yang bertupuk dalam satu permintaan atau *MultiThreading*.
6. MySQL merupakan *Database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *GigaByte* sekalipun.

7. MySQL didukung oleh *driver* ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa *visual* seperti *visual Basic* dan *Delphi*.
8. MySQL adalah *database* menggunakan enkripsi *password*, jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.
9. MySQL merupakan *Database Server* yang *multi user*, artinya *database* ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
10. MySQL mendukung *field* yang dijadikan sebagai kunci *primer* dan kunci unik (*Unique*).
11. MySQL memiliki kecepatan dalam pembuatan *table* maupun *update an table*.

2.6.1 SQL

SQL (*Structured Query Language*) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. SQL secara de facto merupakan bahasa standar yang digunakan dalam RDBMS (*relational database management system*). Saat ini hampir semua *server* basis data yang ada mendukung bahasa SQL untuk melakukan manajemen datanya. SQL merupakan bahasa baku (ANSI/SQL), non prosedural dan berorientasi himpunan (*set oriented language*) SQL dapat

digunakan baik secara interaktif atau ditempelkan (*embedded*) pada sebuah program aplikasi. SQL dibagi menjadi tiga bentuk Query yaitu DDL, DML dan DCL, (Haris Saputro, 2003).

1. DDL (*Data Definition Language*)

DDL adalah sebuah metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *Database*, *Query* yang dimiliki DDL adalah :

- *CREATE*: Digunakan untuk membuat *Database* dan Tabel
- *Drop*: Digunakan untuk menghapus Tabel dan *Database*
- *Alter*: Digunakan untuk melakukan perubahan *struktur* tabel yang telah dibuat, baik menambah *Field* (*Add*), mengganti nama *Field* (*Change*) ataupun menamakannya kembali (*Rename*), dan menghapus *Field* (*Drop*).

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* DML ini untuk melakukan pemanipulasian *database* yang telah dibuat. *Query* yang dimiliki DML adalah :

- *INSERT*: Digunakan untuk memasukkan data pada Tabel *Database*
- *UPDATE*: Digunakan untuk pengubahan terhadap data yang ada pada Tabel *Database*

- *DELETE*: Digunakan untuk Penhapusan data pada tabel *Database*

3. DCL (*Data Control Language*)

DCL adalah sebuah metode *Query SQL* yang digunakan untuk memberikan hak otorisasi mengakses *Database*, mengalokasikan *space*, pendefinisian *space*, dan pengauditan penggunaan *database*. *Query* yang dimiliki DCL adalah:

- *GRANT*: Untuk mengizinkan User mengakses Tabel dalam *Database*.
- *REVOKE*: Untuk membatalkan izin hak *user*, yang ditetapkan oleh perintah *GRANT*
- *COMMIT*: Menapkan penyimpanan *Database*
- *ROLLBACK* : Membatalkan penyimpanan *Database*

2.7 UML

Menurut Nugroho (2010: 6) menyatakan bahwa “UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek.

Menurut Rembaugh, dkk (2005: 3) sebagai pencipta UML menyatakan bahwa “*The Unified Modeling Language (UML) is general-purpose visual*

modeling that is used to specify, visualize, construct, and document the artifact of a software system.

UML adalah salah satu *tool* / model untuk merancang pengembangan *software* yang berbasis object oriented. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen – komponen yang diperlukan dalam sistem *software*.

2.7.1 Building blocks

Menurut Booch (2005:7), UML merupakan bahasa standar yang digunakan dalam perancangan sebuah sistem. Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML, seperti building block, aturan-aturan yang menyatakan bagaimana building block diletakkan secara bersamaan, dan beberapa mekanisme umum (common). Tiga macam yang terdapat dalam building block adalah:

1. Benda/*Things*

Adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen – elemen lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda / thing adalah sebagai berikut:

- *Classes*

Classes yang diuraikan sebagai sekelompok dari *object* yang mempunyai *atribute*, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*. Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, *atribute*, dan metoda pengoperasiannya.

- *Interfaces*

Interfaces merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. *Interface* / antar-muka mendefinisikan sebuah set / kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen.

- *Collaboration*

Collaboration yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas / elemen-elemen yang bekerja secara bersama-sama. *Collaborations* mempunyai struktur dan dimensi. Pemberian sebuah kelas memungkinkan berpartisipasi

didalam beberapa *collaborations* dan digambarkan dengan sebuah ‘*elips*’ dengan garis terpotong-potong.

- *Use cases*

Use Cases adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. ‘*usecase*’ digunakan untuk membentuk tingkah-laku benda / *things* dalam sebuah model serta di realisasikan oleh sebuah *collaboration*. Umumnya ‘*usecase*’ digambarkan dengan sebuah ‘*elips*’ dengan garis yang solid, biasanya mengandung nama.

- *Nodes*

Nodes merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalah sebuah komputer, umumnya mempunyai sedikitnya *memory* dan *processor*. Sekelompok komponen mungkin terletak pada sebuah *node* dan juga mungkin akan berpindah dari *node* satu ke *node* lainnya. Umumnya *node* ini digambarkan seperti kubus serta hanya mengandung namanya.

2. Hubungan/Relationships

Ada 4 macam hubungan didalam penggunaan UML, yaitu;

- *Dependency*

Dependency adalah hubungan semantik antara dua benda/*things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus putus.

- *Association*

Association hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *association* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.

- *Generalizations*

Generalizations adalah menggambarkan hubungan khusus dalam obyek anak/child yang menggantikan obyek parent / induk. Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah.

- *Realizations*

Realizations merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau

elements, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan *realization*.

3. Bagan/*Diagrams*

Bagan/*Diagrams* Sebagai kumpulan / *group* dari benda-benda/*things*.

2.7.2 Diagram

Rosa dan Shalahudin (2016:133), menjelaskan bahwa UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan design, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian gambar dan diagram yang sangat baik.

UML mempunyai 9 diagram, yaitu:

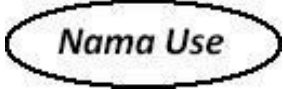


1. Diagram *Use Case*



Diagram *Use Case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya.

Diagram *Use Case* dekat kaitannya dengan kejadian-kejadian. Kejadian (*scenario*) merupakan contoh apa yang terjadi Ketika seseorang berinteraksi dengan sistem.

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa dan Shalahuddin, 2015:156)

Tabel 2 . 1 Simbol *Use case* (Rosa dan Shalahuddin, 2015:156)

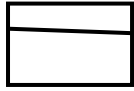
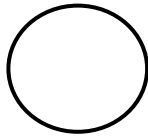

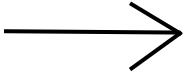

Simbol	Keterangan
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>Use case</i></p>
<p>Aktor / actor</p> 	<p>Orang, proses, atau sistem lain 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, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.</p>
<p>Asosiasi / association</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

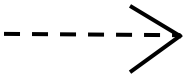
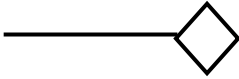
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inherence</i> pada pemograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>

2. Diagram *Class*

Diagram *Class* memberikan pandangan secara luas dari suatu *system* dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis (menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan).

Tabel 2 . 2 Simbol *Class Diagram* (Rosa dan shalahuddin, 2016:146)

Simbol	Nama	Keterangan
	<i>Class</i>	Kelas pada struktur
	<i>Anterface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
	<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
	<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
	<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (whole-part)


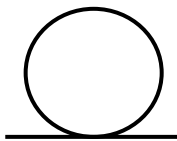
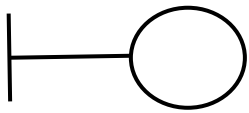
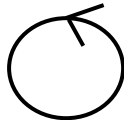
3. Diagram *Package*


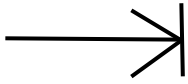
Untuk mengatur pengorganisasian diagram *Class* yang kompleks, dapat dilakukan pengelompokan kelas-kelas berupa *package* (paket - paket). Package adalah kumpulan elemen-elemen logika UML.

4. Diagram *Sequence*

Diagram *Class* dan diagram *Object* merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti Diagram *Interaction*. Diagram *sequence* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Tabel 2 . 3 Simbol *Sequence Diagram* (Rosa dan shalahuddin, 2016:165)

Simbol	Nama	Keterangan
	<i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan system
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel

	<i>A Focus Of Control</i>	Menggambarkan tempat mulai dan berakhirnya sebuah message (pesan)
	<i>A Line Of Life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi


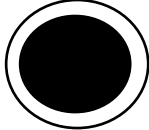
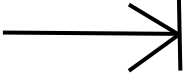

5. Diagram *Collaboration*

Diagram *Collaboration* juga merupakan diagram *interaction*. Diagram membawa informasi yang sama dengan diagram *Sequence*, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

6. Diagram *StateChart*

Behaviors dan *state* dimiliki oleh obyek. Keadaan dari suatu obyek bergantung pada kegiatan dan keadaan yang berlaku pada saat itu. Diagram *StateChart* menunjukkan kemungkinan dari keadaan obyek dan proses yang menyebabkan perubahan pada keadaannya.

Tabel 2 . 4 Simbol *State Chart Diagram* (Rosa dan Salahuddin, 2016:163)



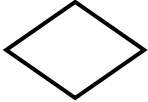

Simbol	Nama	Deskripsi
	<i>Start Awal</i>	Start atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
	<i>Start Akhir</i>	End atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu <i>system</i>
	<i>Event</i>	Adalah kegiatan yang menyebabkan berubahnya status mesin
	<i>State</i>	<i>State</i> atau status adalah keadaan sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu memicu perubahan tersebut

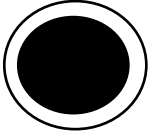
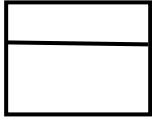
7. Diagram *Activity*

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada obyek yang dalam suatu proses (atau proses

menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

Tabel 2 . 5 Simbol *Activity diagram* (Rosa dan salahuddin, 2016:162)

Simbol	Nama	Deskripsi
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu

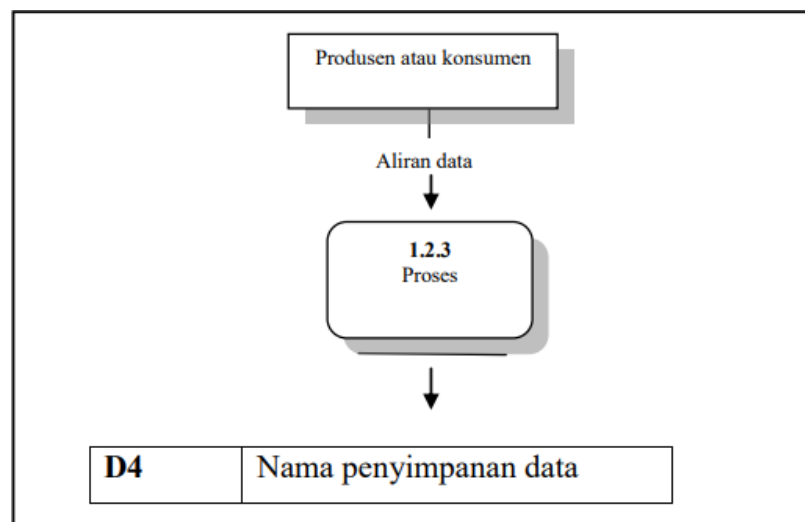
		aktivitas digabungkan menjadi satu.
	Status Akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

8. Diagram *Deployment*

Component adalah sebuah *code module* (kode-kode modul). Diagram *Component* merupakan fisik sebenarnya dari diagram *Class*. Diagram *Deployment* menerangkan bahwa konfigurasi fisik *software* dan hardware.

2.8 Data Flow Diagram (DFD)

Sumanto dan Shalahuddin (2015:69) yang dimaksud dengan *Data Flow Diagram* awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem. Berikut adalah contoh *Data Flow Diagram* yang dikembangkan oleh Chris Gane dan Trish Sarson:



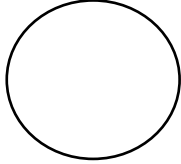

Gambar 2. 5 Contoh DFD yang dikembangkan (Chris gane dan Trish sarson,1979)



Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an dimana mengubah persegi dengan sudut lengkung (pada DFD Chris dan Tris Sarson) dengan lingkaran untuk menotasikan.

DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

Sukanto dan Shalahuddin (2015:71) notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :

Tabel 2 . 6 Notasi DFD (Edward Yourdon dan Tom DeMarco,1980)

NO	NOTASI	K ETERANGAN
1.		<p>Proses atau fungsi atau prosedur pada perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur didalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>), pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-</p>

		tabel pada basis data ERD, <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDF).
3.		Entitas luar (<i>External Entity</i>) atau masukan (<i>Input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.
4.		Aliran data, merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data missal “data peserta”.

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan *Diagram Flow Diagram* (DFD):

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem yang lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di *breakdown* menjadi DFD Level 2. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di *breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di *breakdown*.

4. Membuat DFD Level 3 dan seterusnya

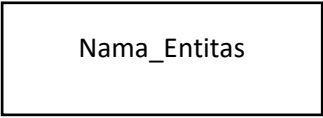
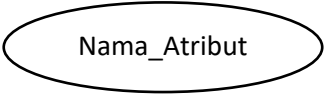
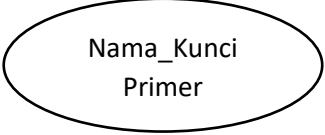
DFD Level 3,4,5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di atasnya. *Breakdwon* pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

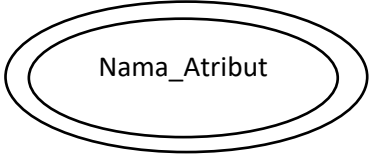
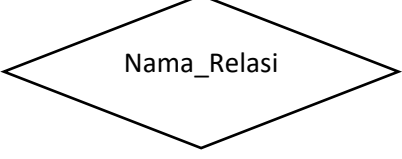
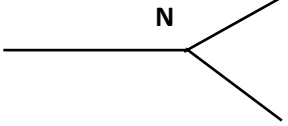
Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit.

2.9 Entity Relationship Diagram (ERD)

Menurut Sukamto dan Shalahuddin (2015:289) “*Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen). Barket (dikembangkan oleh Richard Barket, Ian Palmen, Harry Ellis), notasi Crow’s Foot, dan beberapa notasilain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2 . 7 Simbol pada ERD (Peter Chen, 1970)


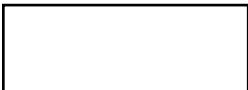



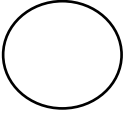
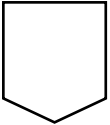

SIMBOL	DESKRIPSI
<p>Entitas / Entity</p> 	<p>Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p>Atribut Kunci Primer</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record 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).</p>

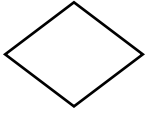



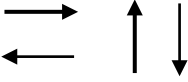
<p>Atribut multinilai/multivalue</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam entitas yang dapat memiliki nilai dari satu.</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.</p>
<p>Asosiasi</p> 	<p>Penghubung antar relasi dan entitas dimana dikedua 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 <i>one to many</i> menghubungkan entitas A dan entitas B.</p>

2.10 Flow Chart

Menurut Indrajani (2015:36) yang dimaksud *Flow Chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Simbol-simbol dalam *Flow Chart* adalah sebagai berikut:

Tabel 2 . 8 Simbol pada *Flow chart* (Peter Chen, 1970)

NO	SIMBOL	KETERANGAN
1.		Simbol <i>Start</i> atau <i>End</i> yang mengidentifikasi awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Simbol masukan atau keluaran dari atau ke sebuah pita <i>magnetic</i> .
5.		Simbol <i>Input/Output</i> yang mendefinisikan masukan dan keluaran proses.
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama.
7.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda.
8.		Simbol masukan atau keluaran dari atau ke sebuah dokumen.

9.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
10.		Simbol <i>database</i> atau basis data.
11.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll.
12.		Simbol yang mendefinisikan proses yang dilakukan secara manual.
13.		Simbol untuk menghubungkan antar proses atau antar simbol.

2.11 Website

Menurut Bekti (2015:35) menyimpulkan bahwa: *Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing masing dihubungkan dengan jaringan-jaringan halaman.

Sebuah situs *web* biasanya ditempatkan setidaknya pada sebuah server *web* yang dapat diakses melalui jaringan seperti internet, ataupun jaringan lokal (LAN)

melalui alamat internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai Waring Wera Wanua atau lebih dikenal dengan singkatan WWW.

Meskipun setidaknya halaman beranda situs internet umumnya dapat diakses publik secara bebas, pada prakteknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs *web* mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota, atau bahkan meminta pembayaran untuk dapat menjadi anggota untuk dapat mengakses isi yang terdapat dalam situs *web* tersebut. Misalnya situs-situs yang menampilkan berita, layanan email danlainnya. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi atau karena tujuan komersial tertentu.

2.12 Gunung

Gunung adalah bagian yang ada di permukaan bumi yang menjulang lebih tinggi dibandingkan dengan wilayah sekitarnya. Barisan atau kumpulan gunung yang berjajar disebut dengan pegunungan. Definisi lebih rinci tentang gunung diberikan oleh *Encyclopedia Britannica* dengan menyebutkan bahwa dataran dengan ketinggian 2000 kaki atau setara 600 meter baru bisa dikatakan sebagai gunung. Kemudian, dalam Kamus Besar bahasa Indonesia (KBBI) disebutkan bahwa gunung adalah bukit yang amat besar dan tinggi yang umumnya mencapai lebih dari 600 meter di atas permukaan laut (mdpl). Gunung dapat terbentuk sebagai akibat dari aktivitas tektonik.

Kumpulan atau rangkaian gunung dan bukit disebut pegunungan. Ketinggiannya lebih dari seribu meter di atas permukaan air laut. Pegunungan dapat berupa daerah gunung berapi. Bisa juga berupa gunung tidak berapi. Contohnya, Pegunungan Jayawijaya, Pegunungan Bukit Barisan, Tengger, Kapuas Hulu, dan sebagainya. Pegunungan sendiri terbentuk setelah melalui tiga tahap. Tahap pertama adalah sedimentasi, yaitu endapan terbawa dari daratan oleh sungai atau terlempar dari gunung berapi. Endapan ini kemudian mengendap dan menjadi lapisan tebal, biasanya terjadi di dalam lautan, dan lalu menjadi mampat menjadi batuan endapan. Tahap kedua adalah adanya pergerakan lapisan kerak yang akan mendesak batu-batuan dan mendorongnya sehingga terlipat. Kemudian, tahap ketiga adalah tekanan yang sangat besar ini menyebabkan lapisan batuan terangkat dan membentuk pegunungan.

2.12.1 Jenis Pegunungan di Indonesia

Di Indonesia sudah banyak dijumpai pegunungan. Jika dilihat secara sekilas, antara pegunungan yang satu dengan pegunungan yang lain tidak ada perbedaan atau tampak mirip. Namun, pegunungan-pegunungan ini dapat dibedakan menjadi beberapa jenis, terutama jika ditinjau dari ketinggian yang dimiliki pegunungan yang bersangkutan.

Berikut beberapa jenis-jenis pegunungan:

1. Pegunungan rendah, pegunungan ini merupakan jenis pegunungan yang memiliki ketinggian tidak terlalu tinggi. Pegunungan dikatakan termasuk

kategori pegunungan rendah jika hanya memiliki ketinggian antara 500 meter sampai dengan 1.500 meter di atas permukaan air laut.

2. Pegunungan tinggi adalah kebalikan dari pegunungan rendah. Disebut pegunungan tinggi jika pegunungan itu memiliki ketinggian di atas pegunungan rendah, atau memiliki ketinggian lebih dari 1.500 meter dari atas permukaan air laut.

BAB III

ANALISA MASALAH DAN PERANCANGAN PROGRAM

3.1 Analisa

3.1.1 Analisa Masalah

Berdasarkan pengalaman dan hasil pengamatan penulis, bahwasanya informasi mengenai jalur pendakian terutama bagi pendaki pemula sangat dibutuhkan, hal ini untuk mengurasi resiko terjadinya beberapa kasus kehilangan, terjatuh, hipotermia hingga kecelakaan ketika mendaki.

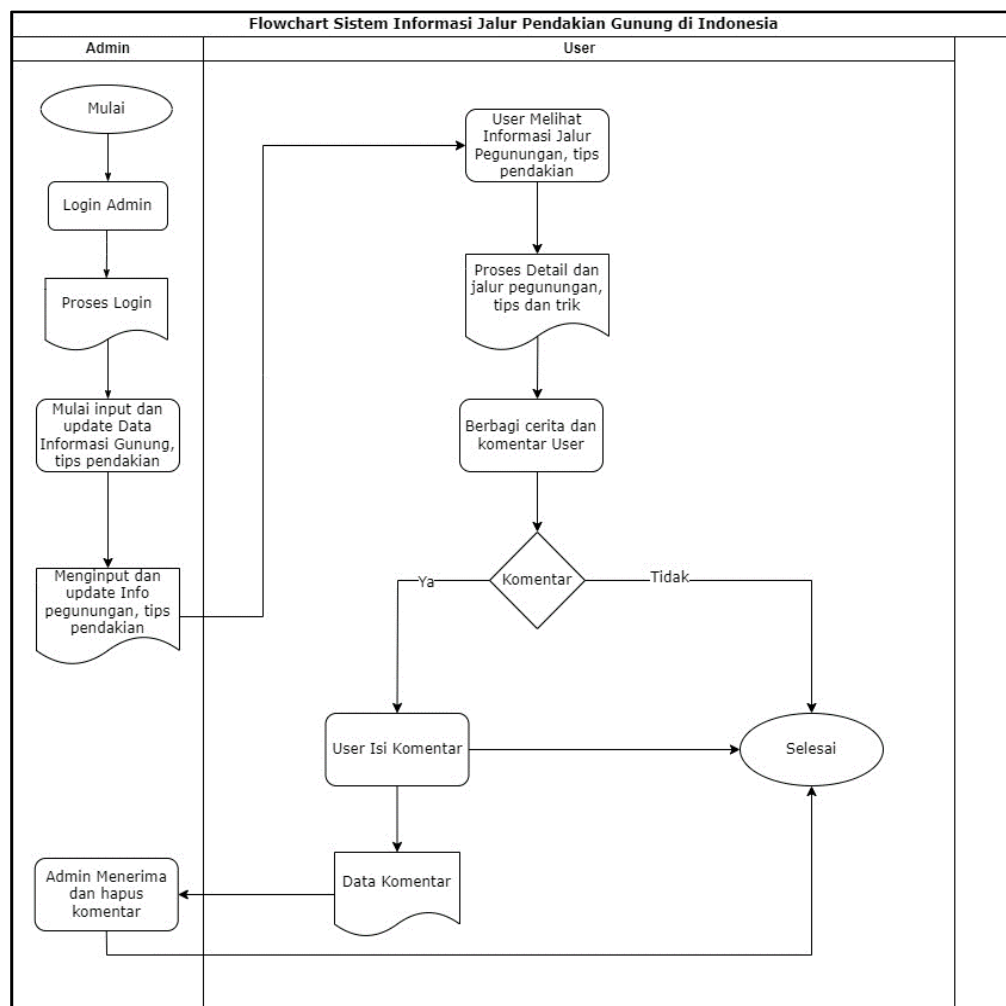
Berdasarkan hal tersebut, penulis tertarik untuk membuat sistem informasi jalur pendakian menggunakan aplikasi berbasis *web*. Sehingga dapat meminimalisir resiko untuk para pendaki pemula dengan memberikan berbagai informasi seputar gunung dan jalur pendakian yang akan di tuju.

3.1.2 Analisa Sistem Berjalan

Dari hasil pengamatan penulis berikut adalah prosedur yang berjalan pada sistem informasi :

1. *Admin* melakukan *Login* untuk masuk kedalam *web*.

2. *Admin* menginput dan *update* daftar gunung.
3. *User* melihat daftar gunung Informasi jalur pegunungan.
4. *User* melihat, berbagi cerita dan mengisi komentar seputar informasi jalur pegunungan
5. *Admin* dapat menerima dan menghapus komentar dari *user*
6. *User* melihat tips pendakian



Gambar 3 . 1 Flowchart Analisa Sistem berjalan

Didalam Sistem ini, tersedia dua menu yang dapat diakses admin yaitu data gunung dan data komentar. Data gunung didalam admin berfungsi untuk menambahkan informasi gunung seperti nama gunung, jalur pendakian, info gunung hingga lokasinya. sedangkan data komentar berfungsi menghapus komentar-komentar yang masuk dari user.

Data user berfungsi untuk melihat informasi jalur gunung yang berada di menu daftar gunung. didalamnya user dapat melihat beberapa gunung dengan informasi seputar jalur, lokasi, waktu trek yang ditempuh dan fasilitas yang berada dibeberapa jalur tersebut. Sehingga para pendaki dapat memilih jalur yang dapat ditempuh sesuai dengan kebutuhan dan medannya. User juga dapat memilih menu berbagi cerita dan tips pendakian. Menu berbagi cerita memberikan ruang untuk user berbagi cerita seputar pendakiannya, berkomentar dan bertanya kepada userlainnya. Dan didalam menu tips pendakian, user dapat melihat beberapa tips seputar pendakian seperti perlengkapan yang harus dibawa, rekomendasi gunung untuk pemula dan lainnya.

3.1.3. Analisa Kebutuhan Non Fungsional

Adapun kebutuhan secara spesifik yang dibutuhkan agar dapat menunjang kelancaran dalam penggunaan sistem pendukung keputusan penilaian kinerja tersebut. Ada 3 macam kebutuhan yang dapat di jabarkan, antara lain :

- a. Kebutuhan Hardware

Hardware atau perangkat keras yang dibutuhkan oleh user untuk menunjang kelancaran dalam penggunaan sistem yaitu PC atau laptop dan modem. Berikut adalah minimal spesifikasi detail mengenai perangkat keras yang di butuhkan :

1. Processor : Dual-Core 2.7 Ghz.
2. RAM : 2 GB DDR3
3. Disk : Free Space 30 GB
4. Graphic Card : Graphics card that support DirectX 9.0c 128 MB
5. Modem : HSDPA 7.2 Mbps/ HSUPA 5.76 Mbps

b. Kebutuhan Software

Software atau perangkat lunak yang dibutuhkan untuk menunjang kelancaran dalam penggunaan sistem antara lain :

1. Sistem Operasi Windows 7, 8, 8.1 atau 10.
2. Browser.
3. Web Server
4. Draw.io

c. Kebutuhan Brainware

Brainware atau pengguna dalam sistem ini ada 2 level. Berikut adalah detail peran mengenai pembagian level dari *user* :

1. *User*

User disini dapat melihat informasi, tips pendakian, berbagi cerita dan mengisi komentar seputar informasi jalur pegunungan.

2. Admin

Admin disini memiliki hak akses yang lebih dari *user*, Admin memiliki akses *Login* untuk *control* keseluruhan fitur yang ada di sistem seperti menambahkan, menghapus, mengubah Informasi jalur gunung dan komentar dari *User*.

3.1.3 Analisa Kebutuhan Fungsional

Analisa fungsional ini menggambarkan kebutuhan yang dijalankan oleh sistem sesuai dengan fungsi yang ada. Berikut adalah gambaran mengenai fungsi - fungsi yang ada pada sistem :

a. Login

1. Admin melakukan Login untuk masuk kedalam *web*.

b. Daftar Gunung

1. Admin melakukan input data Informasi gunung didalam menu daftar gunung. jika ada pembaharuan data, admin dapat mengupdate dan menghapus data tersebut.
2. *User* dapat melihat informasi gunung.

c. Berbagi cerita

1. *User* dapat mengisi kolom cerita untuk menceritakan pengalamannya..
2. *User* dapat berkomentar didalam kolom komentar yang disediakan.
3. Admin dapat menerima komentar yang masuk dari *user* dan dapat menghapus komentar tersebut.

d. Tips Pendakian

1. Admin melakukan input dan mengupdate data tips pendakian untuk dilihat *user*.
2. *User* dapat melihat tips pendakian.

3.2 Perancangan Sistem

3.2.1 Desain Sistem dengan *Unified Modeling Language* (UML)

Perancangan yang dilakukan oleh penulis terhadap alur sistem yang dibuat akan di implementasikan menggunakan UML yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *state chart diagram*.

3.2.1.1 Use Case Diagram

Pada use case diagram ini menjelaskan interaksi antara sistem dengan Admin dan user.

1. Aktor

Tabel 3 . 1 Penjelasan aktor

No	Aktor	Penjelasan
1.	Admin	Login admin, Input dan update data gunung, terima dan hapus komentar <i>user</i>
2.	<i>User</i>	Melihat Informasi gunung, berbagi cerita, berkomentar dan melihat tips pendakian.

2. Use case



Gambar 3 . 2 Use case diagram

Penjelasan mengenai *Use case* Diagram dapat dilihat pada Tabel 3.2.

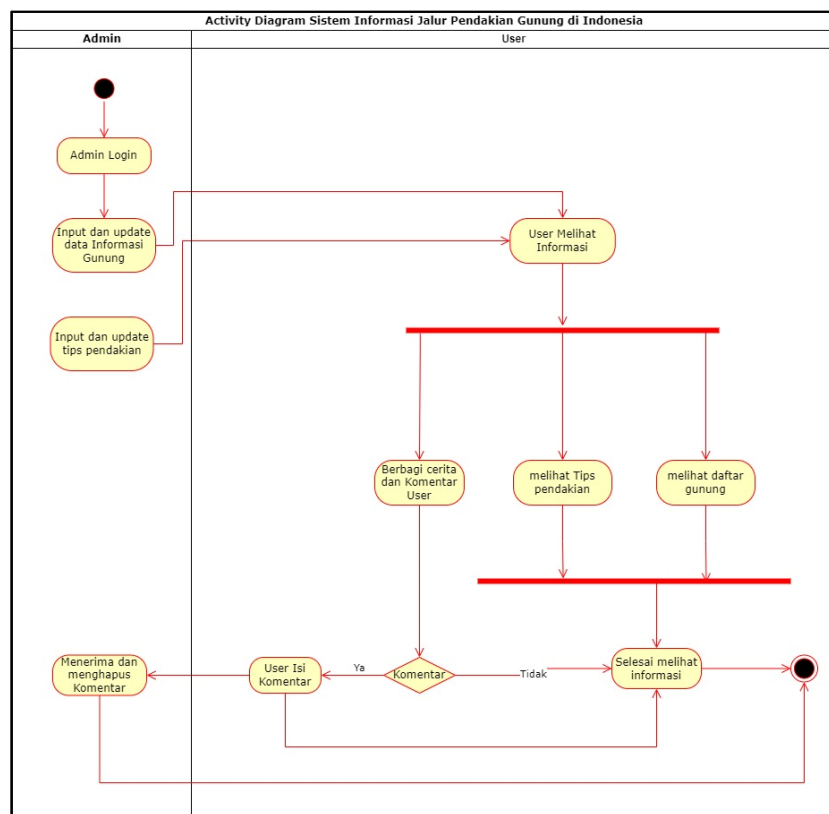
Tabel 3 . 2 Penjelasan *Use case* diagram

No	Use case	Penjelasan
1.	Melakukan Login	Admin dapat melakukan login untuk masuk kedalam <i>web</i>

2.	Input, update dan hapus data gunung	Admin dapat menginput, update dan hapus data gunung didalam daftar gunung.
3.	Terima dan hapus komentar	Admin dapat menerima dan menghapus komentar yang masuk di dalam berbagi cerita.
4.	Input, update dan hapus tips pendakian	Admin dapat Input, update dan hapus tips pendakian.
5.	Melihat Informasi jalur pegunungan	<i>User</i> dapat melihat informasi jalur pegunungan di daftar pegunungan.
6.	Komentar <i>User</i>	<i>User</i> dapat berkomentar dikolom komentar yang berada di berbagi cerita.
7.	Melihat tips pendakian	<i>User</i> dapat melihat tips pendakian.

3.2.1.2 Activity Diagram

Pada diagram aktivitas ini menjelaskan bagaimana proses-proses yang ada pada sistem berdasarkan dengan fungsi yang dijalankan oleh Admin maupun *user*. Aktivitas yang digambarkan dalam *activity diagram* ini merupakan gambaran bagaimana alur penggunaan sistem informasi jalur pegunungan yang sedang berjalan dapat dilihat pada diagram 3.3.



Gambar 3 . 3 Activity Diagram

Penjelasan mengenai *Activity* Diagram dapat dilihat pada Tabel 3.3.

Tabel 3 . 3 Penjelasan *Activity* diagram

No	Activity	Penjelasan
1.	Admin : Melakukan Login	Admin dapat melakukan login untuk masuk kedalam <i>web</i>
2.	Input, update dan hapus data gunung	Admin dapat menginput, update dan hapus data gunung didalam daftar gunung.
3.	Input, update dan hapus tips pendakian	Admin dapat Input, update dan hapus tips pendakian.
4.	Admin: Terima dan hapus komentar	Admin dapat menerima dan menghapus komentar yang masuk di dalam komentar
5.	<i>User</i> : Memilih untuk berkomentar	<i>User</i> dapat memilih untuk berkomentar atau tidak dalam kolom komentar yang berada di berbagi cerita. Jika <i>user</i> memilih Ya, maka <i>user</i> mengisi komentar, namun jika tidak, <i>user</i> selesai melihat informasi.
6.	<i>User</i> : Melihat tips pendakian	<i>User</i> dapat melihat tips pendakian.

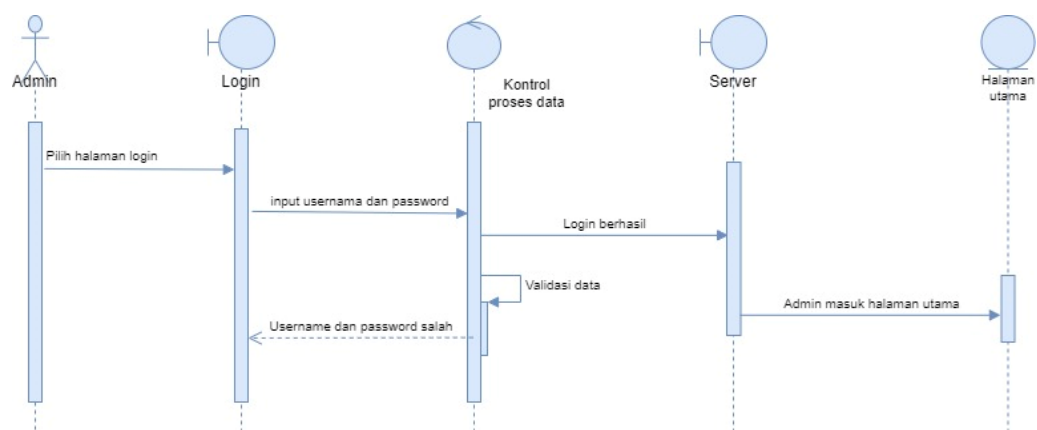
7.	<i>User:</i> Melihat Informasi jalur pegunungan	<i>User</i> dapat melihat informasi jalur pegunungan yang sudah dimasukkan di daftar gunung.
----	---	--

3.2.1.3 Sequence Diagram

Pada diagram ini menjelaskan bagaimana interaksi antara objek pada sistem yang sedang berjalan berdasarkan urutan waktunya. Berikut adalah *sequence* diagram pada sistem yang dibuat :

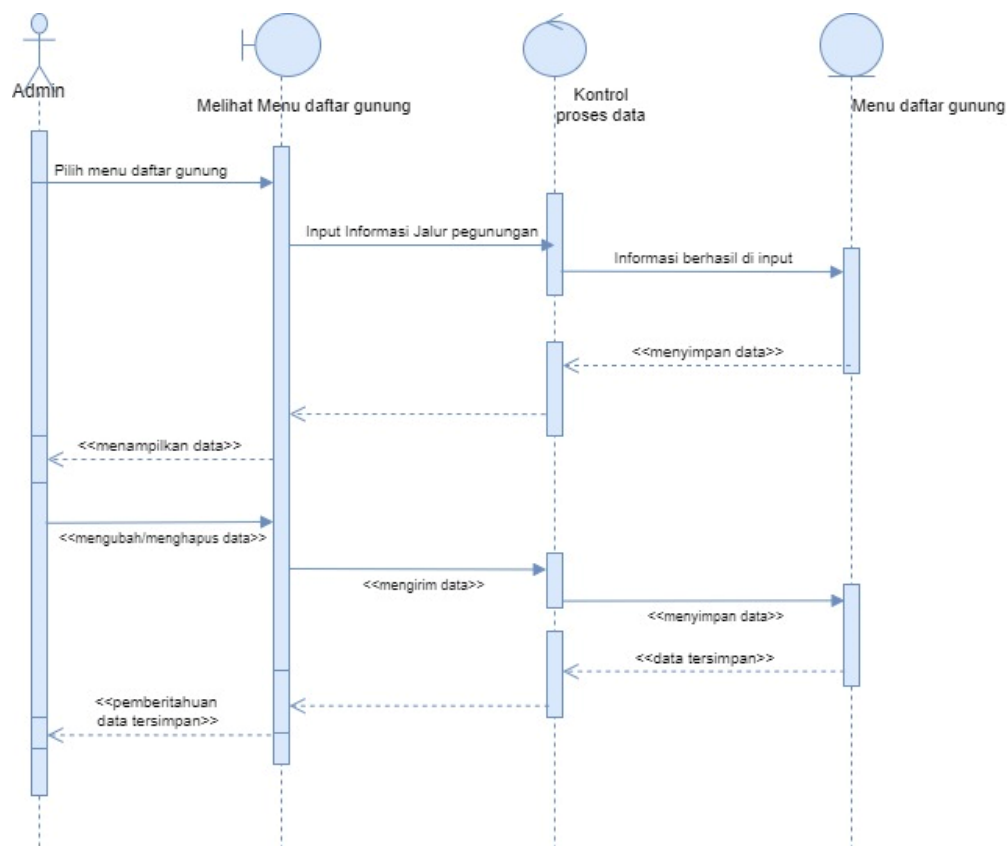
1. Interaksi yang terjadi pada saat admin melakukan proses *login* ke sistem.

Admin masuk ke halaman *login*, admin memasukkan *username* dan *password*. Jika *username* dan *password* benar, maka admin akan masuk ke halaman utama. Namun jika *username* dan *password* salah, akan Kembali ke kolom *login*.



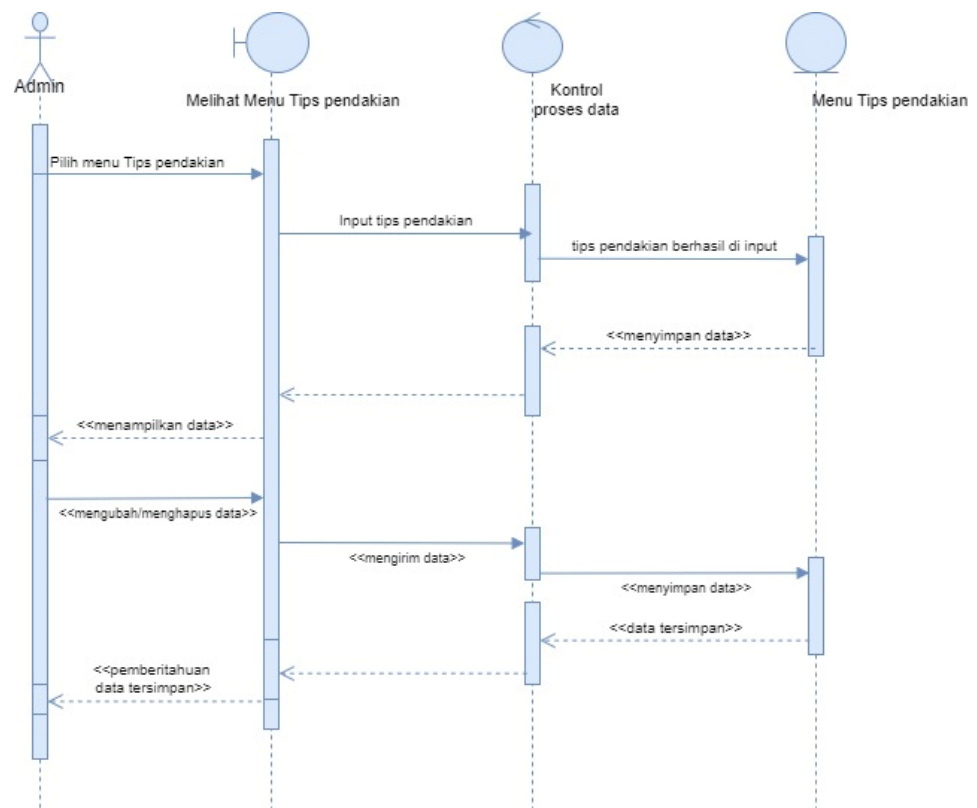
Gambar 3 . 4 Sequence Admin Login

2. Interaksi yang terjadi pada saat admin melakukan proses input, admin memilih menu daftar gunung dan menginput kedalamnya informasi jalur pegunungan. setelah proses input berhasil admin dapat mengupdate dan menghapus data tersebut.



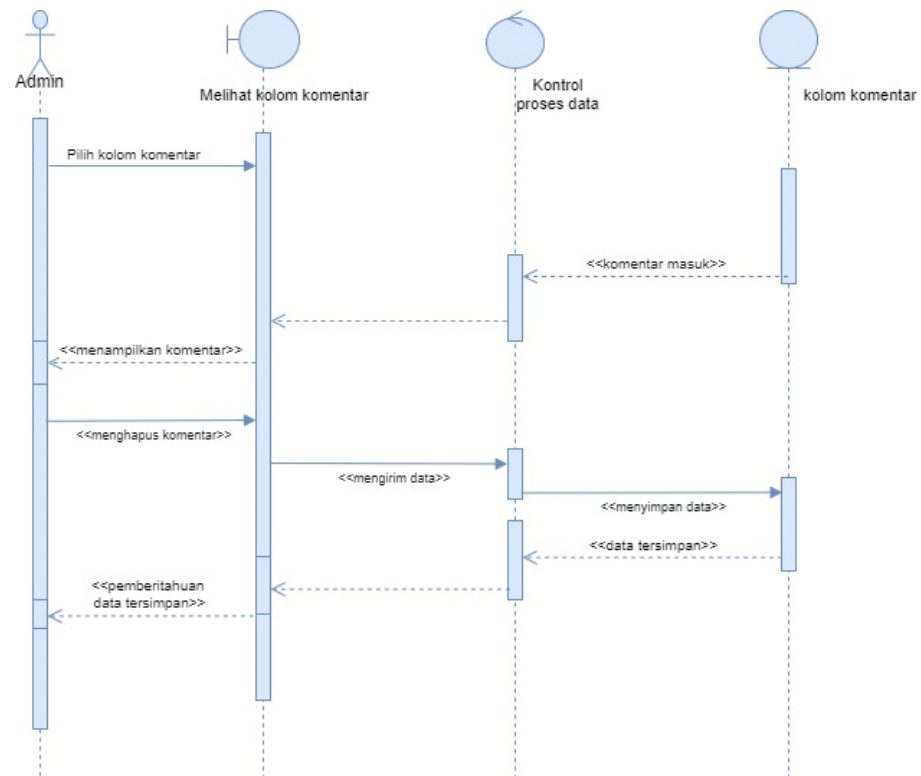
Gambar 3 . 5 Sequence Admin Input informasi jalur pegunungan

3. Interaksi yang terjadi pada saat admin melakukan proses input tips pendakian . Admin memilih menu tersebut, lalu admin menginput tips pendakian. Setelah proses input berhasil, admin dapat mengupdate dan menghapus data tersebut.



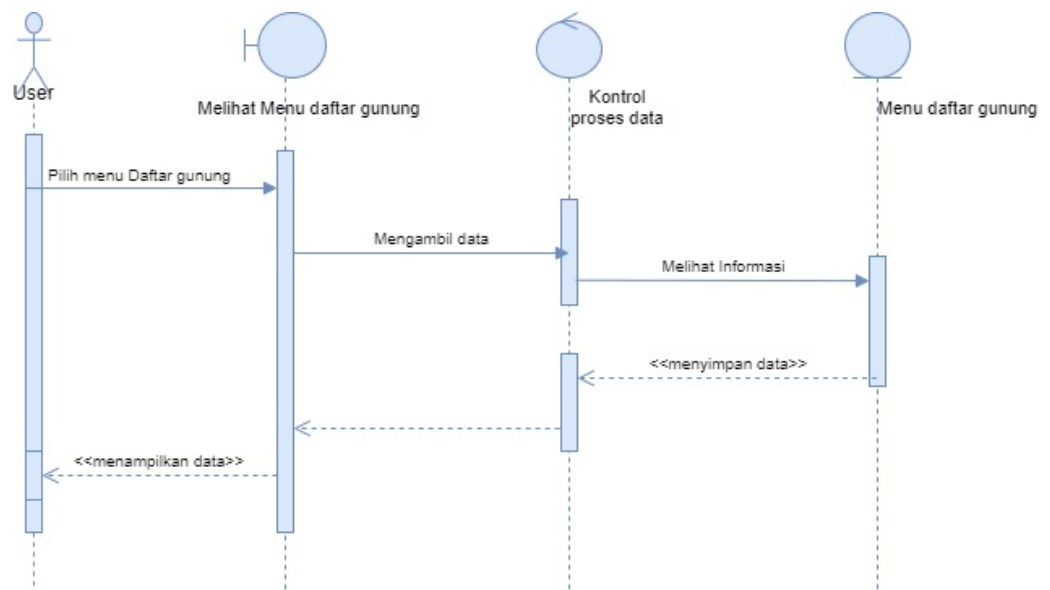
Gambar 3 . 6 Sequence Admin Input tips pendakian

4. Interaksi yang terjadi pada saat admin menerima komentar dari *user*. Admin melihat dan menerima komentar yang masuk dari *user*. Setelah admin menerima komentar, admin juga dapat menghapus komentar tersebut.



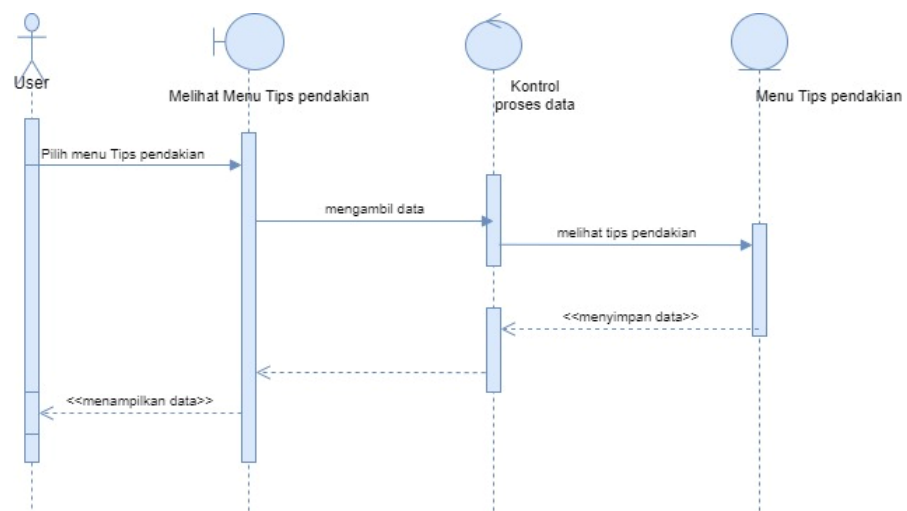
Gambar 3 . 7 Sequence Admin menerima komentar

5. Interaksi yang terjadi pada saat user melihat informasi jalur pegunungan.
User dapat memilih menu daftar gunung . Setelah *user* berada dalam menu tersebut, *user* dapat melihat seluruh informasi jalur pegunungan dan detail maps pegunungan tersebut.



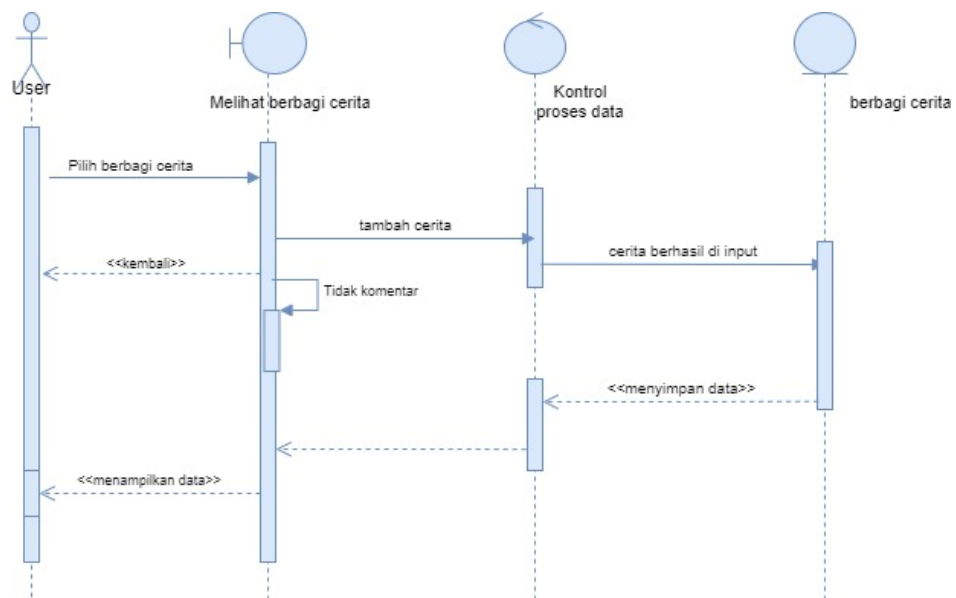
Gambar 3 . 8 Sequence *User* melihat menu informasi jalur pegunungan

6. Interaksi yang terjadi pada saat user melihat tips dan trik. User dapat memilih menu tips dan trik. Setelah user berada dalam menu tersebut, user dapat melihat tips pendakian.



Gambar 3 . 9 Sequence *User* melihat tips pendakian

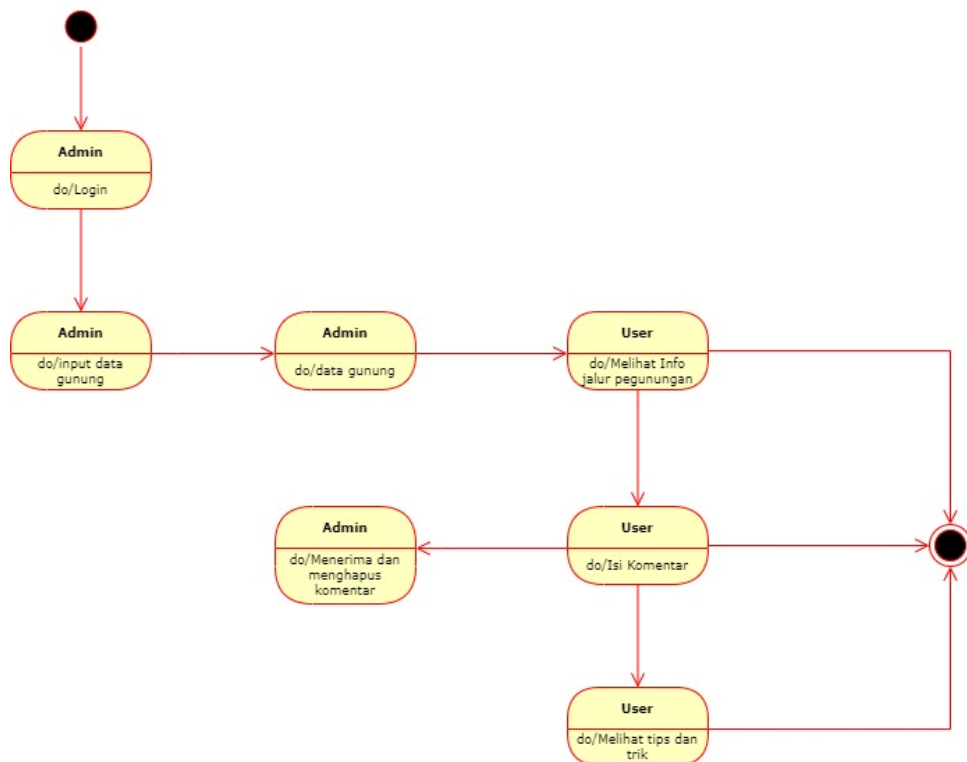
7. Interaksi yang terjadi pada saat *user* memilih kolom berbagi cerita dan kolom komentar pada menu berbagi cerita. *User* dapat mengisi kolom berbagi cerita dan kolom komentar . Jika *user* tidak ingin mengisinya *user* dapat kembali ke menu semula.



Gambar 3 . 10 Sequence *User* melihat dan mengisi kolom berbagi cerita dan komentar

3.2.1.4 StateChart Diagram

Pada diagram ini menjelaskan transisi dan perubahan keadaan (dari satu state ke state lainnya). Penjelasan *State Chart* diagram terdapat pada Diagram 3.11



Gambar 3 . 11 *State chart* diagram

Tabel 3 . 4 Penjelasan *State chart* diagram

No	<i>State chart</i>	Penjelasan
1.	Admin : Melakukan Login	Admin dapat melakukan login untuk masuk kedalam <i>web</i>
2.	Input, update dan hapus data gunung	Admin dapat menginput, update dan hapus data gunung didalam daftar gunung.
3.	Input, update dan hapus tips pendakian	Admin dapat Input, update dan hapus tips pendakian.
4.	Admin: Terima dan hapus komentar	Admin dapat menerima dan menghapus komentar yang masuk di dalam komentar
5.	<i>User</i> : Memilih untuk berkomentar	<i>User</i> dapat memilih untuk berkomentar atau tidak dalam kolom komentar yang berada di berbagi cerita. Jika <i>user</i> memilih Ya, maka <i>user</i> mengisi komentar, namun jika tidak, <i>user</i> selesai melihat informasi.
6.	<i>User</i> : Melihat tips pendakian	<i>User</i> dapat melihat tips pendakian.

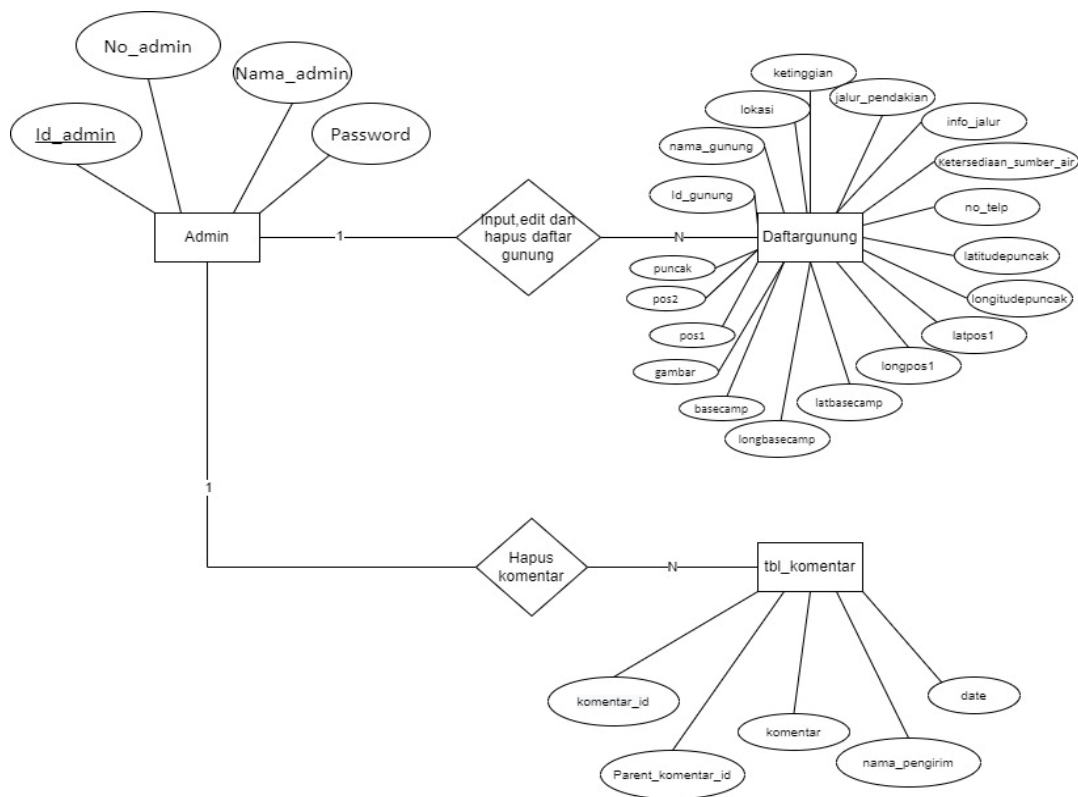
7.	<i>User:</i> Melihat Informasi jalur pegunungan	<i>User</i> dapat melihat informasi jalur pegunungan yang sudah dimasukkan di daftar gunung.
----	---	--

3.2.2 Desain Database Sistem

Selain menggunakan UML untuk membuat sebuah desain sistem, desain *database* pun perlu diperlukan untuk memperjelas bagaimana hubungan antara entitas yang ada. Desain *database* yang digunakan antara lain ERD dan struktur tabel.

3.2.2.1 Entity Relationship Database (ERD)

Entity Relationship Database atau (ERD) yang digunakan dalam pembuatan sistem informasi ini adalah sebagai berikut :



Gambar 3 . 12 ERD Sistem Informasi Jalur pegunungan

3.2.2.2 Struktur Tabel

Berikut adalah struktur tabel yang digunakan dalam pembuatan sistem informasi jalur pendakian.

Tabel 3 . 5 Login Admin

Field	Tipe data
Id_admin	Int(3)
No_admin	Varchar(10)
Nama_admin	Varchar(25)
password	Varchar(50)

Tabel 3 . 6 Komentar

Field	Tipe data
komentar_id	Int (11)
Parent_komentar_id	Int(11)
komentar	Varchar(200)
nama_pengirim	Varchar(40)
date	timestamp

Tabel 3 . 7 Daftar gunung

Field	Tipe data
Id_gunung	Int(8)
Nama_gunung	Varchar(255)
Lokasi	Varchar(5)
ketinggian	Varchar(255)
Jalur_pendakian	Mediumtext
Info_jalur	Mediumtext
Ketersediaan_sumber_air	Varchar(50)
No_telp	Varchar(255)
Latitude puncak	Varchar(50)

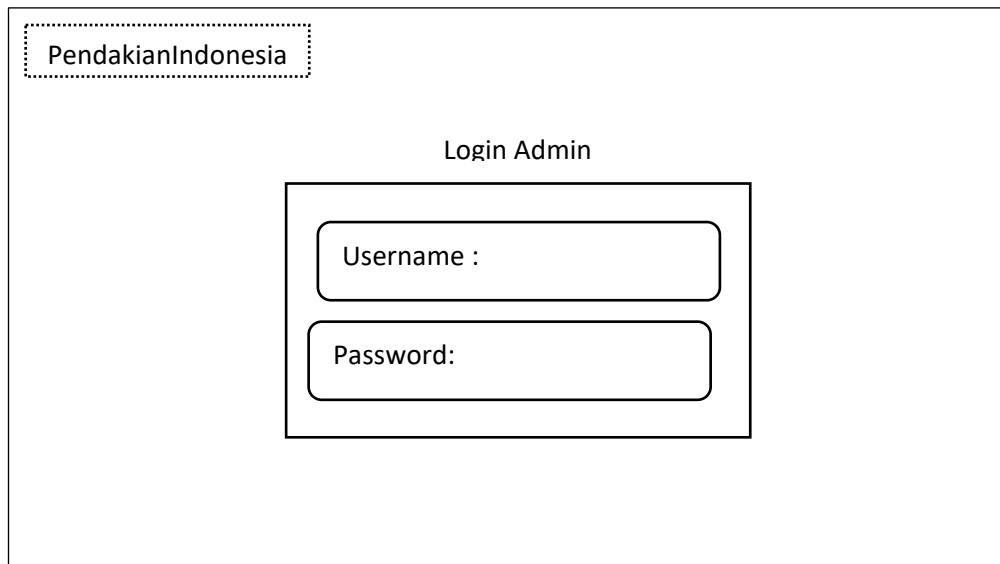
Longitudepuncak	Varchar(50)
Latpos1	Varchar(50)
Longpos1	Varchar(50)
Latpos2	Varchar(50)
Longpos2	Varchar(50)
Latpos3	Varchar(50)
Longpos3	Varchar(50)
Latpos4	Varchar(50)
Longpos4	Varchar(50)
Latpos5	Varchar(50)
Longpos5	Varchar(50)
Latpos6	Varchar(50)
Longpos6	Varchar(50)
Latpos7	Varchar(50)
Longpos7	Varchar(50)
Latpos8	Varchar(50)
Longpos8	Varchar(50)
Latpos9	Varchar(50)
Longpos9	Varchar(50)
latbasecamp	Varchar(50)
longbasecamp	Varchar(50)
gambar	Varchar(50)
basecamp	Varchar(50)
Pos1	Varchar(50)
Pos2	Varchar(50)
Pos3	Varchar(50)
Pos4	Varchar(50)

Pos5	Varchar(50)
Pos6	Varchar(50)
Pos7	Varchar(50)
Pos8	Varchar(50)
Pos9	Varchar(50)
puncak	Varchar(50)

3.2.3 Desain *Interface* Sistem

Sebelum membuat program tentunya ada desain tampilan yang dimiliki oleh sistem. Berikut adalah desain tampilan dari sistem yang dibuat.

1. Tampilan menu login admin



The image shows a web application interface for 'PendakianIndonesia'. In the top left corner, the text 'PendakianIndonesia' is enclosed in a dashed rectangular box. Centered on the page is the title 'Login Admin'. Below this title is a rectangular login form. Inside the form, there are two input fields: the first is labeled 'Username :' and the second is labeled 'Password:'. Both labels are positioned to the left of their respective input boxes.

Gambar 3 . 13 Desain menu login

2. Tampilan menu daftar gunung pada admin untuk menambah daftargunung.

PendakianIndonesia

Daftar gunung ▼

Berbagi cerita ▼

Simpan data

No	Field	Isi data
1.	Id_gunung	
2.	Nama_gunung	
3.	Lokasi	
4.	Ketinggian	
5.	Jalur_pendakian	
6.	Info_jalur	
7.	Ketersediaan_sumber_air	
8.	No_telp	
9.	Latitudepuncak	
10.	Longitudepuncak	
11.	Latpos1	
12.	Longpos1	
13.	Latpos2	
14.	Longpos2	
15.	Latpos3	
16.	Longpos3	
17.	Latpos4	
18.	Longpos4	
19.	Latpos5	
20.	Longpos5	

Gambar 3 . 14 Desain tambah daftar gunung

PendakianIndonesia

Daftar gunung ▼

Berbagi cerita ▼

No	Field	Isi data
21.	Latpos6	
22.	Longpos6	
23.	Latpos7	
24.	Longpos7	
25.	Latpos8	
26.	Longpos8	
27.	Latpos9	
28.	Longpos9	
29.	latbasecamp	
30.	longbasecamp	
31.	gambar	
32.	basecamp	
33.	Pos1	
34.	Pos2	
35.	Pos3	
36.	Pos4	
37.	Pos5	
38.	Pos6	
39.	Pos7	

Gambar 3 . 15 Desain tambah daftar gunung

3. Tampilan menu daftar gunung pada admin untuk edit dan hapus.

PendakianIndonesia

Daftar gunung ▼

Berbagi ▼

Tambah

No	Nama Gunung	Ketinggian	Kota	Kontak Basecamp	Aksi
1	Gunung Semeru	3675 mdpl	Kab Malang dan Kab Lumajang, Jawa Timur	TNBTS (0341) 491828 ; Fax. (0341) 490885 CP : 085238659182/0341787055 (Mas Gofur), Resort Ranupane : 08283930822 081227920297 - Mahmudin (petugas)	<div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Edit</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Hapus</div>
2	Gunung Guntur	2249 mdpl	Kab. Garut, Jawa Barat	Jalur Citiis : @bcguntur.ummytati / 085223705155 @basecamp_villaguntursari / 085294125342 Jalur Cikahuripan : @gunung_guntur / 085222382442 / 08989316234	<div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Edit</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Hapus</div>
3	Gunung Prau	2565 mdpl	Kab. Wonosobo, Jawa Tengah	Jalur Patak Banteng : @patakbanteng / 085228283428 @prau_patakbanteng / 081226709022 @gunung_prau_2565mdpl / 081555833303 @gunung_prau_dieng / 082242905557 Jalur Kali Lembu : @baseamp.g.prau.via.kalilembu / 082322076215 Jalur Dieng Wetan : @basecamp_	<div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Edit</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; display: inline-block;">Hapus</div>

Gambar 3 . 16 Desain edit dan hapus daftar gunung

4. Tampilan menu berbagi cerita pada admin untuk hapus komentar

PendakianIndonesia

Daftar gunung ▼

Berbagi cerita ▼

Hapus

No	Nama pengirim	Komentar	Aksi
1			
2			
3			

Gambar 3 . 17 Desain hapus komentar

5. Tampilan menu beranda pada *user*.

PendakianIndonesia

Beranda ▼

Daftar gunung ▼

Berbagi cerita ▼

Tips ▼

Gambar Gunung

Ketahui jalur yang akan dilalui

Cari informasi sebelum melewati jalu

Lebih lanjut

Gambar 3 . 18 Desain beranda pada *user*

6. Tampilan menu daftar gunung pada *user*

PendakianIndonesia

Beranda

Daftar gunung

Berbagi cerita

Tips

Gambar Gunung

No	Nama Gunung	Ketinggian	Kota	Kontak Basecamp	Aksi
1	Gunung Semeru	3675 mdpl	Kab Malang dan Kab Lumajang, Jawa Timur	TNBTS (0341) 491828 ; Fax. (0341) 490885 CP : 085238659182/0341787055 (Mas Gofur), Resort Ranupane : 08283930822 081227920297 - Mahmudin (petugas)	Detail dan lokasi
2	Gunung Guntur	2249 mdpl	Kab. Garut, Jawa Barat	Jalur Citiis : @bcguntur.ummytati / 085223705155 @basecamp_villaguntursari / 085294125342 Jalur Cikahuripan : @gunung_guntur / 085222382442 / 08989316234	Detail dan lokasi
3	Gunung Prau	2565 mdpl	Kab. Wonosobo, Jawa Tengah	Jalur Patak Banteng : @patakbanteng / 085228283428 @prau_patakbanteng / 081226709022 @gunung_prau_2565mdpl / 081555833303 @gunung_prau_dieng / 082242905557 Jalur Kali Lembu : @baseamp.g.prau.via.kalilembu / 082322076215 Jalur Dieng Wetan : @basecamp_	Detail dan lokasi

Gambar 3 . 19 Desain daftar gunung pada *user*

7. Tampilan detail dan lokasi pada menu daftar gunung *user*.

PendakianIndonesia


Beranda

Daftar gunung

Berbagi cerita

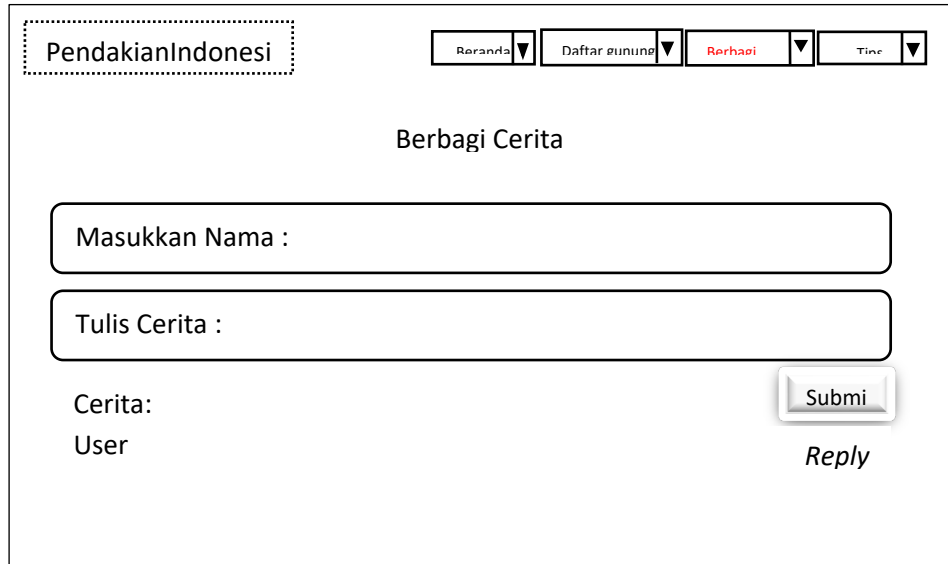
Tips

Gunung Semeru

Item	Detail
Nama Gunung	Gunung Semeru
Lokasi	Kab. Wonosobo, Jawa Tengah
Ketinggian	2565 MDPL
Jalur pendakian	<p>b.Jalur pendakian</p> <ul style="list-style-type: none"> •Jalur Pranten, di Kecamatan Bawang Kabupaten Batang (5-6jam) <p>Medan jalur pertama terbelah landai dengan trek yang cukup ekstrem. Namun jalur setelahnya terdapat naik turun bukit dan harus berhati-hati.</p> <ul style="list-style-type: none"> •Gunung Prau Via Pathak Banteng di Wonosobo (2jam) <p>Medan jalur tanah dengan trek cukup terjal dan menanjak. Jarak tempuh jalur ini sangat singkat yaitu 2 jam.</p> <ul style="list-style-type: none"> •Gunung Prau Via Wates <p>Medan jalur landai dan cukup ekstrem. Dijalur ini akan bertemu dengan air terjun sebagai sumber mata air.</p> <ul style="list-style-type: none"> •Gunung Prau via Kalilembu <p>Medan jalur tanah dan cukup mudah dilalui. Terdapat 3 pos melewati kali serayu.</p> <ul style="list-style-type: none"> •Gunung Prau via Dieng Kulon (Dwarawati) <p>Medan jalur ini tanah dan mudah dilalui. Apabila melewati jalur ini maka bisa sekaligus berwisata Telaga Pengilon, Candi Arjuna dan Telaga Warna.</p>
Info Pendakian	Gunung Prau terletak di kawasan dataran tinggi Dieng, Kabupaten Wonosobo, Desa Patakbanteng dengan ketinggian 2.565 MpdL. Dengan jalur pendakian yang relatif mudah dan cukup jelas bagi para pendaki serta view yang istimewa menjadikan Gunung Prau menjadi gunung favorit bagi para pendaki saat ini. Untuk mendaki Puncak Gunung Prau, pendaki hanya membutuhkan sekitar 3-4 jam trekking. Ada 2 jalur pendakian yang menjadi favorit para pendaki yaitu melalui jalur Dieng ataupun Patakbanteng. Jalur pendakian Via Dieng cenderung lebih santai dan tidak berbahaya jika dibandingkan dengan jalur Patak Banteng yang jalurnya cukup terjal meskipun waktu tempuhnya lebih singkat.
Foto Gunung	

Gambar 3 . 20 Desain detail dan lokasi pada menu daftar gunung *user*

8. Tampilan menu berbagi cerita pada *user*.



The interface for sharing a story is titled "Berbagi Cerita". It features a header with the site name "PendakianIndonesia" and a navigation bar with links: "Beranda", "Daftar gunung", "Berhaci", and "Tinc". The main form consists of two input fields: "Masukkan Nama :" and "Tulis Cerita :". Below these fields, the text "Cerita:" and "User" is displayed. A "Submi" button is positioned to the right of the "Tulis Cerita" field. A "Reply" link is located at the bottom right of the form.

Gambar 3 . 21 Desain berbagi cerita *user*

9. Tampilan menu tips pendakian pada *user*.



The interface for hiking tips is titled "Tips". It features a header with the site name "PendakianIndonesia" and a navigation bar with links: "Berand", "Daftar", "Berbagi", and "Tios". The main content area is a large rectangle labeled "Isi Tips".

Gambar 3 . 22 Desain tips pendakian *user*

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1 Implementasi

4.1.1 Implementasi Perangkat Keras

Sesuai dengan apa yang dijabarkan pada bab 3, perangkat keras yang digunakan dalam penerapan aplikasi ini memiliki minimum spesifikasi sebagai berikut :

1. Processor : Dual-Core 2.7 Ghz.
2. RAM : 2 GB DDR3
3. Disk : Free Space 30 GB
4. Graphic Card : Graphics card that support DirectX 9.0c 128 MB
5. Modem : HSDPA 7.2 Mbps/ HSUPA 5.76 Mbps

4.1.2 Implementasi Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan guna melancarkan penerapan














aplikasi ini, antara lain :

1. Sistem Operasi Windows 7, 8, 8.1 atau 10.
2. Browser.
3. Web Server
4. Draw.io

4.1.3 Implementasi *Database*

Sesuai dengan apa yang dijabarkan pada bab 3, berikut adalah implementasi pembuatan tabel pada *database* yang digunakan oleh aplikasi :

1. Struktur tabel admin pada *database* yang digunakan. Table admin berisi username dan password untuk login admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id 	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/> 2	username	varchar(50)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/> 3	password	varchar(50)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/> 4	nama	varchar(200)	utf8mb4_general_ci		No	None			 Change  Drop  More

Gambar 4 . 1 Login admin

2. Struktur tabel daftar gunung pada *database* datagunung berisi informasi gunung dan titik kordinat antar pos.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_gunung 📌	int(8)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nama_gunung	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3 lokasi	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 ketinggian	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	5 jalur_pendakian	mediumtext	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	6 info_jalur	mediumtext	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	7 ketersediaan_sumber_air	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	8 no_telp	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	9 latitudepuncak	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	10 longitudepuncak	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	11 latpos1	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	12 longpos1	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	13 latpos2	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	14 longpos2	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	15 latpos3	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	16 longpos3	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	17 latpos4	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	18 longpos4	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	19 latpos5	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	20 longpos5	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	21 latbasecamp	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	22 longbasecamp	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	23 gambar	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	24 basecamp	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	25 pos1	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	26 pos2	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	27 pos3	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	28 pos4	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	29 pos5	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	30 puncak	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Gambar 4 . 2 Daftar

3. Struktur tabel komentar pada database tbl_komentar berisi komentar dan nama pengirim yang nantinya akan diisi oleh *user*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	komentar_id			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	parent_komentar_id			No	None			Change Drop More
<input type="checkbox"/>	3	komentar	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4	nama_pengirim	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5	date			No	current_timestamp()			Change Drop More

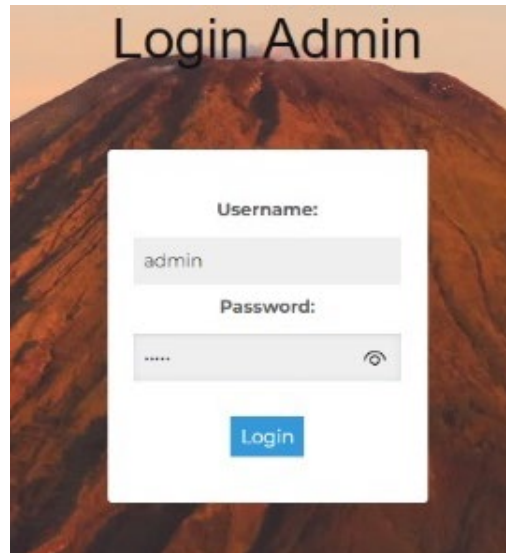
Gambar 4 . 3 Komentar

4.1.4 Implementasi Antarmuka

Pada implementasi ini penulis menyesuaikan tampilan yang ada pada aplikasi dengan rancangan tampilan yang ada pada bab 3. Berikut adalah tampilan yang ada pada aplikasi yang telah dibuat :

1. Tampilan halaman Login.

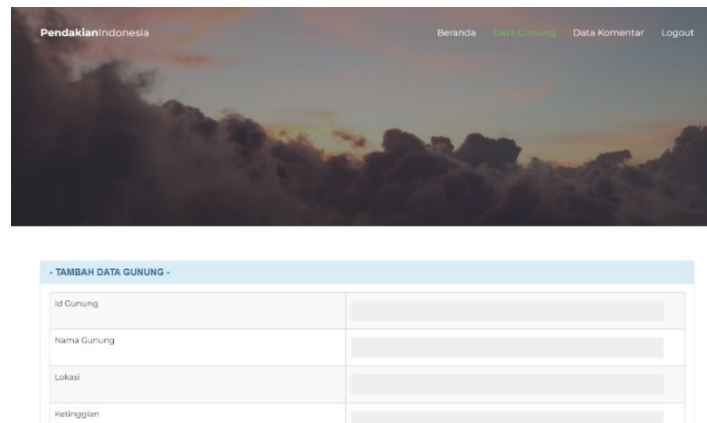
Pemilihan halaman login merupakan halaman yang menampilkan form untuk login admin. Berikut tampilan halaman login.



Gambar 4 . 4 Halaman Login

2. Tampilan tambah daftar gunung pada admin.

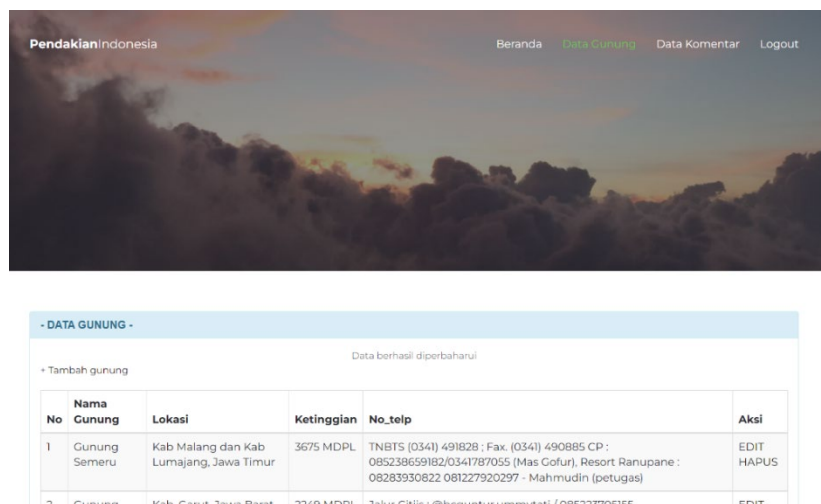
Menu daftar gunung pada admin merupakan halaman yang menampilkan menu daftar gunung. didalam menu terdapat tombol tambah data gunung yang berfungsi menambah informasi gunung.



Gambar 4 . 5 Tambah daftar gunung pada admin

3. Tampilan edit dan hapus daftar gunung pada admin

Menu edit dan hapus daftar gunung pada admin merupakan halaman informasi gunung yang dapat di edit dan di hapus dengan cara memilih tombol edit atau hapus.

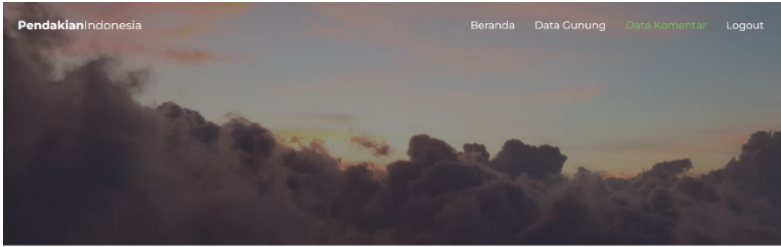


No	Nama Gunung	Lokasi	Ketinggian	No_telp	Aksi
1	Gunung Semeru	Kab Malang dan Kab Lumajang, Jawa Timur	3675 MDPL	TNBS (0341) 491828 : Fax: (0341) 490885 CP : 085238659182/0341787055 (Mas Cofur), Resort Ranupane : 08283930822 081227920297 - Mahmudin (petugas)	EDIT HAPUS
2	Gunung	Kab. Caringi, Jawa Barat	2249 MDPL	Talir Citris : 085227705155	EDIT

Gambar 4 . 6 edit dan hapus daftar gunung pada admin

4. Tampilan menu data komentar.

Menu data komentar merupakan halaman yang menampilkan data gunung yang dapat dihapus oleh admin dengan cara memilih tombol hapus.



PendakianIndonesia Beranda Data Gunung Data Komentar Logout

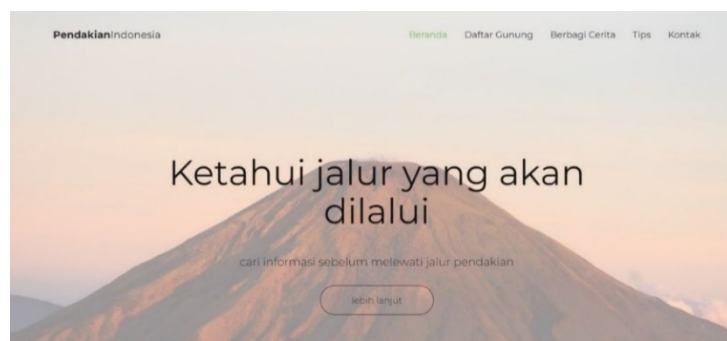
- DATA KOMENTAR -

No	Nama Pengirim	Komentar / Cerita	Tanggal	Aksi
1	Komputer	balasan komentar 1	2019-03-22 18:57:13	HAPUS
2	Dini	balasan anak komentar 1	2019-03-22 18:57:34	HAPUS
3	Dono	Balasan Komentar 2	2019-03-22 18:57:49	HAPUS

Gambar 4 . 7 menghapus komentar pada admin

5. Tampilan menu beranda pada *user* .

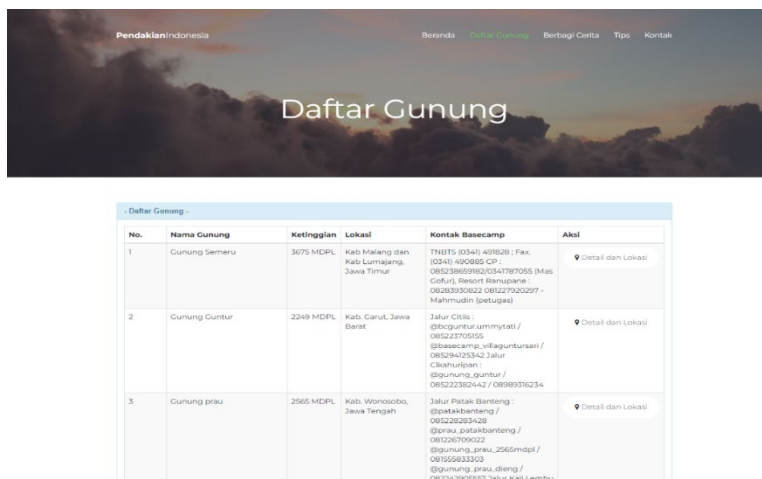
Menu beranda pada *user* merupakan halaman yang menampilkan penjelasan singkat kegunaan system informasi ini.



Gambar 4 . 8 tampilan beranda *user*

6. Tampilan menu daftar gunung pada *user*.

Menu daftar gunung pada *user* merupakan halaman yang menampilkan informasi jalur pegunungan dengan dilengkapi data gunung, informasi jalur, ketinggian, detail dan lokasi.

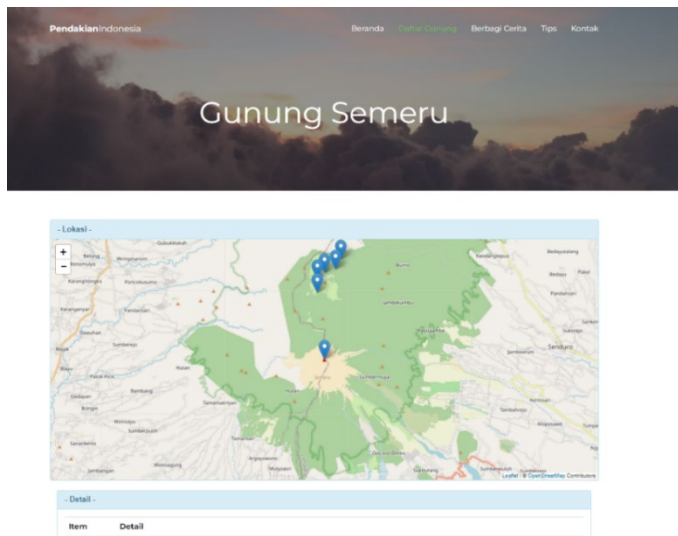


No.	Nama Gunung	Ketinggian	Lokasi	Kontak Basecamp	Aksi
1	Gunung Semeru	3675 MDPL	Kab Malang dan Kab Lumajang, Jawa Timur	TNBT5 (0341) 409828 : Fax (0341) 400895 CP : 081238659862/0347897055 (Mas Gofur), Resort Ranupane : 08240333422 081227952297 - Mahmudin (petugas)	Detail dan Lokasi
2	Gunung Guntur	2249 MDPL	Kab. Garut, Jawa Barat	Jalur Ciri : @bcguntur.ummytati / 08122705955 @basecamp_vilafagunturani / 081234102542 Jalur Cikahuripan : @gunung_guntur / 081223365442 / 08188931234	Detail dan Lokasi
3	Gunung prau	2565 MDPL	Kab. Wonorejo, Jawa Tengah	Jalur Patah Banteng : @patakbanteng / 081226283428 @prau_patakbanteng / 081226709022 @gunung_prau_2565mdpl / 081556833303 @gunung_prau_diang / 082242905557 Jalur Kali Lembu	Detail dan Lokasi

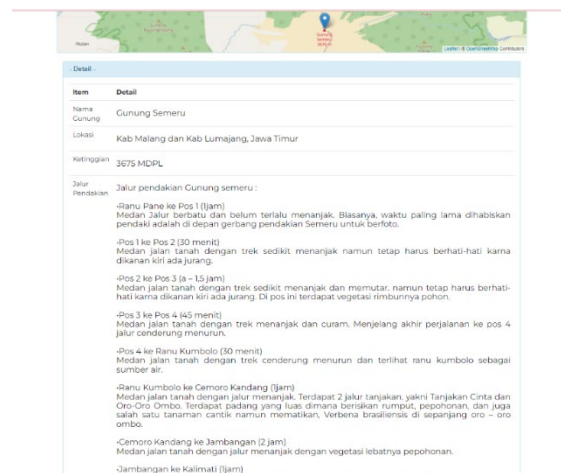
Gambar 4 . 9 daftar gunung pada *user*

7. Tampilan detail dan lokasi pada menu daftar gunung.

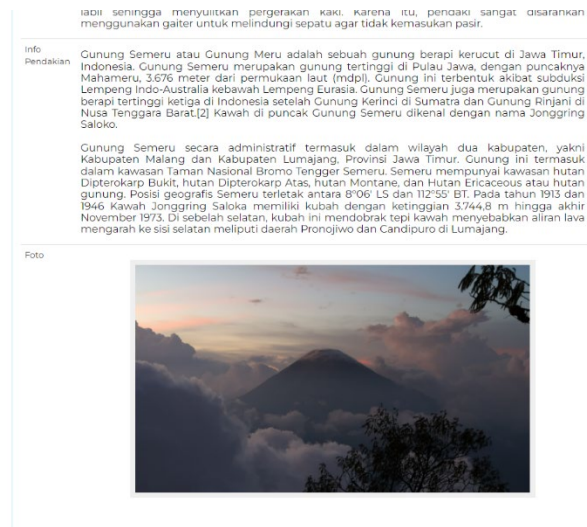
Form tampilan detail dan lokasi pada menu daftar gunung merupakan halaman yang menampilkan informasi detail gunung ketika *user* memilih tombol detail dan lokasi dari gunung yang dipilih.



Gambar 4 . 10 peta titik kordinat detail dan lokasi pada daftar gunung *user*



Gambar 4 . 11 Informasi detail dan lokasi pada daftar gunung *user*



Gambar 4 . 12 Informasi detail dan lokasi pada daftar gunung *user*

8. Tampilan menu berbagi cerita pada *user*

Menu berbagi cerita pada *user* merupakan halaman yang menampilkan kolom cerita dan komentar yang dapat diisi oleh *user*. *User* dapat bercerita tentang pengalaman, informasi danlainnya seputar gunung dan pendakian.

PendakianIndonesia Beranda Daftar Gunung **Berbagi Cerita** Tips Kontak

Berbagi Cerita

Masukkan Nama

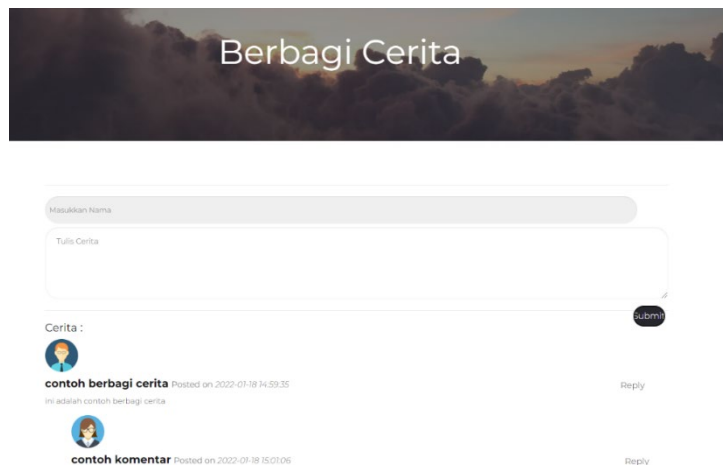
Tulis Cerita

Cerita : Submit

Gambar 4 . 13 berbagi cerita *user*

9. Tampilan komentar pada menu berbagi cerita.

Pada menu berbagi cerita, *User* dapat berkomentar didalam cerita yang sudah dibagikan oleh *user* yanglainnya.



Gambar 4 . 14 berbagi komentar *user*

10. Tampilan menu tips pendakian pada *user*.

Menu tips pada user merupakan halaman yang menampilkan berbagai tips pendakian.



Gambar 4 . 15 tips pendakian pada *user*


4.2 Uji Coba


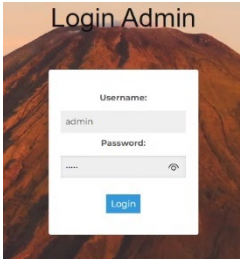
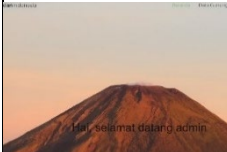
Metode uji coba yang digunakan oleh penulis yaitu *Black Box Testing*. Metode *Black Box* ini menguji fungsi-fungsi yang ada pada aplikasi secara keseluruhan. Berikut adalah hasil dari pengujian yang dilakukan terhadap aplikasi ini :

1. Hasil pengujian pada menu login.

Pengujian pada system dilakukan dengan memasukkan password yang benar dan memasukkan password yang salah. Jika admin memasukkan password atau username yang salah, maka system akan Kembali dan meminta diulangi lagi. Namun jika data password dan username benar, maka akan muncul beranda pada web pendakian indonesia. Berikut data hasil pengujian login.

Tabel 4 . 1 Pengujian Login



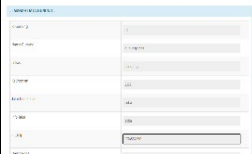



No	Skenario Pengujian	Yang diharapkan	Output	Kesimpulan
1.	Menggunakan email yg benar & katasandi yang salah atau sebaliknya. <i>Test Case :</i>	Sistem memerintahkan untuk memeriksa kembali email & katasandi,	Sesuai yang diharapkan. <i>Test Result :</i> 	Valid

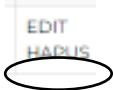
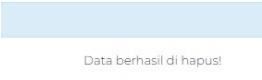
				
2.	<p>Menggunakan email yg benar & katasandi yang benar</p> <p><i>Test Case :</i></p> 	<p>Login berhasil dan sistem akan menampilkan halaman aplikasi sesuai dengan level nya.</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid

2. Hasil pengujian pada menu daftar gunung admin

Pengujian pada menu daftar gunung yang dilakukan oleh admin adalah dengan menambahkan data gunung, mengedit dan menghapus data. Hasil uji valid. Berikut data hasil pengujian daftar gunung admin.

Tabel 4 . 2 Pengujian daftar gunung


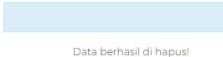
No	Skenario Pengujian	Yang diharapkan	Output	Kesimpulan
1.	<p>Mengklik tombol tambah gunung.</p> <p><i>Test Case:</i></p> 	<p>Sistem menampilkan form untuk mengisi daftar gunung</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid
2.	<p>Input data gunung.</p> <p><i>Test Case:</i></p> 	<p>Sistem memberitahu kan data berhasil di tambah</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid
3.	<p>Klik ubah data gunung.</p> <p><i>Test Case:</i></p> 	<p>Sistem memberitahu kan data berhasil di ubah</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid

4.	Klik hapus data gunung. <i>Test Case:</i> 	Sistem memberitahu kan data berhasil di hapus	Sesuai yang diharapkan. <i>Test Result :</i> 	Valid
----	---	---	---	-------

3. Hasil pengujian pada menu data komentar admin

Pengujian pada menu data komentar yang dilakukan oleh admin adalah dengan menghapus data komentar , Hasil uji valid. Berikut data hasil pengujian daftar gunung admin.

Tabel 4 . 3 Pengujian data komentar

No	Skenario Pengujian	Yang diharapkan	Output	Kesimpulan
1.	Klik hapus komentar <i>Test Case:</i> 	Sistem memberitahukan data berhasil di hapus	Sesuai yang diharapkan. <i>Test Result :</i> 	Valid

4. Hasil pengujian pada menu daftar gunung *user*

Pengujian pada menu daftar gunung pada *user* yang dilakukan adalah dengan menekan tombol detail dan lokasi , Hasil uji valid.

Berikut data hasil pengujian daftar gunung *user*.

Tabel 4 . 4 Pengujian daftar gunung

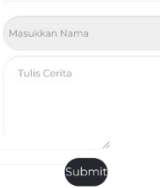
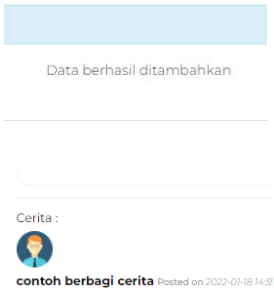
No	Skenario Pengujian	Yang diharapkan	Output	Kesimpulan
1.	Klik detail dan lokasi	Sistem menampilkan	Sesuai yang diharapkan.	Valid

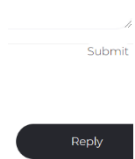
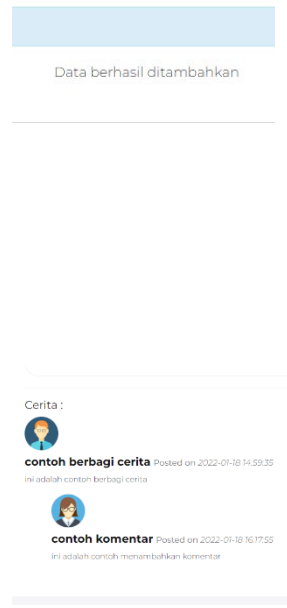
	<p><i>Test Case:</i></p> 	<p>n form daftar gunung</p>	<p><i>Test Result :</i></p> 	
--	--	-----------------------------	--	--

5. Hasil pengujian pada menu berbagi cerita *user*

Pengujian pada menu berbagi cerita dilakukan oleh *user*. *User* dapat mengisi cerita dan komentar.

Tabel 4 . 5 Pengujian komentar

No	Skenario Pengujian	Yang diharapkan	Output	Kesimpulan
1.	<p>Input cerita dan klik submit.</p> <p><i>Test Case:</i></p> 	<p>Sistem memberitahukan data berhasil di tambahkan</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid

2.	<p>Input komentar dan klik reply</p> <p><i>Test Case:</i></p>  <p>contoh komentar</p> <p>ini adalah contoh menambahkan komentar</p>	<p>Sistem memberitahukan data berhasil di tambahkan</p>	<p>Sesuai yang diharapkan.</p> <p><i>Test Result :</i></p> 	Valid

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan, diantaranya:

1. Dengan dibuatnya perancangan sistem informasi jalur pegunungan ini dapat membantu para pendaki untuk mendapatkan informasi seputar jalur pendakian, denah pendakian dan kondisi jalur pendakian. Sistem berbasis *Web* telah dibuat dan di implementasi serta uji coba sistem yang telah dilakukan dengan hasil sudah mampu memuat informasi tersebut.
2. Dengan sistem informasi jalur pegunungan ini pendaki dapat melihat dan memilih jalur pendakian gunung yang akan ditempuh dan fasilitas yang berada di beberapa jalur tersebut. Terdapat menu berbagi cerita untuk user dapat bercerita pengalaman selama pendakian, berkomentar. Dan menu tips pendakian yang berisi tips seputar keperluan dan informasi penting pendakian.

5.2 Saran

Sistem informasi jalur pegunungan ini lebih fokus kepada jalur dan kondisi pegunungan yang akan didaki. Oleh karena itu ada beberapa saran yang dapat disampaikan guna mengembangkan lebih baik lagi. Adapun saran - saran yang dapat disampaikan adalah sebagai berikut:

1. Sistem informasi ini dapat membuat sebuah komunitas untuk para pendaki bisa berbagi ilmu atau ingin mengadakan acara pendakian bersama.
2. Sistem informasi ini dapat bekerjasama dengan toko sewa peralatan pendakian. Sehingga memudahkan para pendaki yang belum lengkap untuk menyewa peralatannya di toko yang sudah bekerjasama tersebut.
3. Sistem informasi ini dapat lebih banyak memberikan informasi jalur pegunungan yang berada di seluruh Indonesia.
4. Sistem informasi ini dapat bekerjasama dengan pihak jasa travel gunung sehingga semakin mempermudah calon pendaki yang akan mendaki ke berbagai pegunungan di Indonesia baik sendiri atau berkelompok.
5. Perancangan Sistem Informasi ini dapat diimplementasikan dengan peta jalur pendakian dan trek pos augmented reality di dalam android.

DAFTAR PUSTAKA

- Bekti, Bintu Humairah. 2015. Mahir Membuat Website dengan Adobe Dreamweaver CS6, CSS dan JQuery. Yogyakarta: ANDI
- Bpakhm.unp (2018): Konsep dasar dan pengertian sistem, <http://bpakhm.unp.ac.id/konsep-dasar-dan-pengertian-sistem/> . Diakses 4 Desember 2021.
- Fatma, Desy (2016): Pengertian Pegunungan: Proses terbentuknya dan jenisnya, <https://ilmugeografi.com/ilmu-bumi/gunung/pengertian-pegunungan>. Diakses 3 Desember 2021.
- Hadinata, Ryan. 2018, Sistem Informasi Kepegawaian Berbasis Web pada PT.Tris Pratama Konsultan, Sekolah Tinggi Manajemen Sistem Informasi Dan Komputer Indonesia Mandiri, Bandung.
- Indrajani. 2015. Database Design (Case Study All in One). Jakarta: PT Elex Media Komputindo.
- Maulana, Kiki. 2021, Rancang Bangun Aplikasi Pendaftaran Online Turnamen Player Unkown Battle Ground Di The Pillars E-Sport Berbasis Web Fakultas Sistem Informasi, Sekolah Tinggi Manajemen Sistem Informasi Dan Komputer Indonesia Mandiri, Bandung.
- Nur, Al lillah (2019): Pengertian Sistem Beserta Ciri, Karakteristik, Komponen dan Contoh Sistem, <https://www.nesabamedia.com/pengertian-sistem/>. Diakses 4 Desember 2021.
- Nairinne, Herlin (2014): Komponen – Komponen (Elemen) dari Sistem, <https://herlinnairine.wordpress.com/2014/11/02/komponen-komponen-elemen-dari-sistem/> . Diakses 14 Desember 2021.

- Robith, Muhammad (2021): Pengertian Sistem Informasi dan Contoh Penerapan pada Dunia Industri, <https://www.sekawanmedia.co.id/sistem-informasi/>.Diakses 14 Desember 2021
- Rosa, Salahuddin, dalam Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek, Edisi ke Dua, Bandung: Informatika, 2014
- S, Rosa, A, Shalahuddin, M. 2016. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung.
- Zakiyudin, Ais. (2019): Sistem informasi manajemen edisi 2, Jakarta: Mitra Wacana Media, 105 halaman.

LAMPIRAN – LAMPIRAN

Listing Program

1. Login_admin.php

```
<!DOCTYPE html>

<html>

<head>

    <title>Login Admin | Pendakian Indonesia</title>

    <link rel="stylesheet" type="text/css" href="css/style.css">

    <!-- Basic Page Needs

===== -->

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>Pendakian Indonesia</title>

    <meta name="description" content="">

    <meta name="author" content="">

    <meta name="keywords" content="">

    <!-- Mobile Specific Metas

===== -->

    <meta name="viewport" content="width=device-width, minimum-scale=1.0,
maximum-scale=1.0">

    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

<!-- Fonts -->

<link
href="https://fonts.googleapis.com/css?family=Montserrat:300,300i,400,600,700"
rel="stylesheet">

<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.4.0/css/font-awesome.min.css">

<!-- Favicon
===== -->

<link rel="apple-touch-icon" sizes="180x180" href="assets/img/apple-touch-
icon.png">

<link rel="icon" type="image/png" sizes="16x16" href="assets/img/favicon.png">

<!-- Stylesheets
===== -->

<!-- Bootstrap core CSS -->
<link href="assets/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="assets/css/style.css" rel="stylesheet">
<link href="assets/css/responsive.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries
-->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
<header id="masthead" class="site-header">

```



```

<nav id="primary-navigation" class="site-navigation">
    <div class="container">

        <div class="navbar-header">

            <a class="site-title"><span>Pendakian</span>Indonesia</a>

        </div><!-- /.navbar-header -->

    </div>
</nav><!-- /.site-navigation -->
</header><!-- /#masthead -->
<div id="hero" class="hero over">
    <div class="hero-content">
        <div class="hero-text">
            <center><h3><?php
                if(isset($_GET['pesan'])){
                    if($_GET['pesan'] == "gagal"){
                        echo "Login gagal! username dan password salah!";
                    }else if($_GET['pesan'] == "logout"){
                        echo "Anda telah berhasil logout";
                    }else if($_GET['pesan'] == "belum_login"){
                        echo "Anda harus login untuk mengakses halaman admin";
                    }
                }
            ?></h3></center>

            <center><h2>Login Admin</h2></center>

        <br/>
        <div class="login">
        <br/>

```

```

<form action="login.php" method="post" onSubmit="return validasi()">
  <div>
    <label>Username:</label>
    <input type="text" name="username" id="username" />
  </div>
  <div>
    <label>Password:</label>
    <input type="password" name="password" id="password" />
  </div>
  <div>
    <input type="submit" value="Login" class="tombol">
  </div>
</form>
</div>
</div><!-- /.hero-text -->
</div><!-- /.hero-content -->
</div><!-- /.hero -->

```

```

<script type="text/javascript">
function validasi() {
  var username = document.getElementById("username").value;
  var password = document.getElementById("password").value;
  if (username != "" && password!="") {
    return true;
  }else{
    alert("Username dan Password harus di isi !");
    return false;
  }
}

```

```

</script>

<footer id="colophon" class="site-footer">

  <div class="copyright">

    <div class="container">

      <div class="row">

        <div class="col-xs-8">

          <div class="social-links">

            <a class="twitter-bg" href="#"><i class="fa fa-twitter"></i></a>

            <a class="facebook-bg" href="#"><i class="fa fa-facebook"></i></a>

            <a class="pinterest" href="#"><i class="fa fa-pinterest"></i></a>

            <a class="linkedin-bg" href="#"><i class="fa fa-linkedin"></i></a>

          </div><!-- /.social-links -->

        </div>

        <div class="col-xs-4">

          <div class="text-right">

            <p>&copy; Pendakian Indonesia</p>

            <p>All Rights Reserved</p>

          </div>

        </div>

      </div>

    </div><!-- /.copyright -->

  </footer><!-- /#footer -->

</body>

<!-- Bootstrap core JavaScript
===== -->

<!-- Placed at the end of the document so the pages load faster -->

```

```

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

<script src="assets/js/bootstrap.min.js"></script>

<script src="assets/js/bootstrap-select.min.js"></script>

<script src="assets/js/jquery.slicknav.min.js"></script>

<script src="assets/js/jquery.countTo.min.js"></script>

<script src="assets/js/jquery.shuffle.min.js"></script>

<script src="assets/js/script.js"></script>

</html>

```

2. Admin_datagunung.php

```

<?php
$title = "DATA GUNUNG";
include_once "../koneksi.php"; ?>

<html lang="en">

<head>

<!-- Basic Page Needs
===== -->

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Daftar Gunung | PendakianIndonesia</title>

<meta name="description" content="">
<meta name="author" content="">
<meta name="keywords" content="">

```

```

<!-- Mobile Specific Metas
===== -->

<meta name="viewport" content="width=device-width, minimum-scale=1.0,
maximum-scale=1.0">

<meta name="apple-mobile-web-app-capable" content="yes">

<!-- Fonts -->

<link
href="https://fonts.googleapis.com/css?family=Montserrat:300,300i,400,600,700"
rel="stylesheet">

<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.4.0/css/font-awesome.min.css">

<!-- Favicon
===== -->

<link rel="apple-touch-icon" sizes="180x180" href="../assets/img/apple-touch-
icon.png">

<link rel="icon" type="image/png" sizes="16x16"
href="../assets/img/favicon.png">

<!-- Stylesheets
===== -->

<!-- Bootstrap core CSS -->

<link href="../assets/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->

<link href="../assets/css/style.css" rel="stylesheet">

<link href="../assets/css/responsive.css" rel="stylesheet">

</head>

<body>

```

```

<header id="masthead" class="site-header site-header-white">
  <nav id="primary-navigation" class="site-navigation">
    <div class="container">
      <div class="navbar-header">
        <a href="index.html" class="site-
title"><span>Pendakian</span>Indonesia</a>
      </div><!-- /.navbar-header -->
      <div class="collapse navbar-collapse" id="agency-navbar-collapse">
        <ul class="nav navbar-nav navbar-right">
          <li><a href="index_admin.php">Beranda</a></li>
          <li class="active"><a href="admin_datagunung.php">Data
Gunung</a></li>
          <li><a href="admin_datakomentar.php">Data Komentar</a></li>
          <li><a href="Logout.php">Logout</a></li>
        </ul>
      </div>
    </div>
  </nav><!-- /.site-navigation -->
</header><!-- /#masthead -->
<div id="hero" class="hero overlay subpage-hero portfolio-hero">
  <div class="hero-content">
    </div>
  </div>

<main id="main" class="site-main">
  <section class="site-section subpage-site-section section-portfolio">
    <div class="container">
      <div class="row">
        <div class="col-md-12">

```



```

    }
    ?>
</p>
<?php endif; ?>
<a
href="tambah.php">+ Tambah gunung</a>

<br/>
<br/>
<thead>
    <tr>

<th>No</th>

<th>Nama Gunung</th>

<th>Lokasi</th>

<th>Ketinggian</th>

<th>No_telp</th>

<th>Aksi</th>

    </tr>
</thead>

<?php

include '../koneksi.php';

$no = 1;

$data = mysqli_query($koneksi,"SELECT * from datagunung");

while($d = mysqli_fetch_array($data)){

```

```
?>
```

```
<tr>
```

```
<td><?php echo $no++; ?></td>
```

```
<td><?php echo $d['nama_gunung']; ?></td>
```

```
<td><?php echo $d['lokasi']; ?></td>
```

```
<td><?php echo $d['ketinggian']; ?></td>
```

```
<td><?php echo $d['no_telp']; ?></td>
```

```
<td>
```

```
<a href="edit.php?id_gunung=<?php echo $d['id_gunung']; ?>">EDIT</a>
```

```
<a href="hapus.php?id_gunung=<?php echo $d['id_gunung']; ?>">HAPUS</a>
```

```
</td>
```

```
</tr>
```

```
<?php
```

```
}
```

```
?>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
</main>
```

```
</body>
```

```
<footer id="colophon" class="site-footer">
  <div class="copyright">
    <div class="container">
      <div class="row">
        <div class="col-xs-8">
          <div class="social-links">
            <a class="twitter-bg" href="#"><i class="fa fa-twitter"></i></a>
            <a class="facebook-bg" href="#"><i class="fa fa-
facebook"></i></a>
            <a class="pinterest" href="#"><i class="fa fa-pinterest"></i></a>
            <a class="linkedin-bg" href="#"><i class="fa fa-
linkedin"></i></a>
          </div><!-- /.social-links -->
        </div>
        <div class="col-xs-4">
          <div class="text-right">
            <p>&copy; Pendakian Indonesia</p>
            <p>All Rights Reserved</p>
          </div>
        </div>
      </div>
    </div>
  </div><!-- /.copyright -->
</footer><!-- /#footer -->
```

```
<!-- Bootstrap core JavaScript
```

```

===== -->

<!-- Placed at the end of the document so the pages load faster -->

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

<script src="../../assets/js/bootstrap.min.js"></script>

<script src="../../assets/js/bootstrap-select.min.js"></script>

<script src="../../assets/js/jquery.slicknav.min.js"></script>

<script src="../../assets/js/jquery.countTo.min.js"></script>

<script src="../../assets/js/jquery.shuffle.min.js"></script>

<script src="../../assets/js/script.js"></script>

</body>
</html>

```

3. Ambil_komentar.php

```

<?php
session_start();
include 'koneksi2.php';
include 'csrf.php';
$output="";

$query = "SELECT * FROM tbl_komentar WHERE parent_komentar_id = '0'
ORDER BY komentar_id DESC";
$dewan1 = $db1->prepare($query);
$dewan1->execute();
$res1 = $dewan1->get_result();
while ($row = $res1->fetch_assoc()) {
    $output .= '
    <div class="media border p-3 mb-2">

```

```

        <div class="media-body">

        <div class="row">

        <div class="col-sm-10">

            <h4><b>'. $row["nama_pengirim"] .'</b> <small> Posted on
<i>'. $row["date"] .'</i></small></h4>

            <p>'. $row["komentar"] .'</p>

        </div>

        <div class="col-sm-2" align="right">

            <button type="button" class="btn btn-primary reply"
id="' . $row["komentar_id"] . "'>Reply</button>

        </div>

    </div>

</div>

';

$output .= ambil_reply($db1, $row["komentar_id"]);
}

```

```

echo json_encode([$output]);

```

```

function ambil_reply($db1, $parent_id = 0, $marginleft = 0){
    $output="";
    $query = "SELECT * FROM tbl_komentar WHERE parent_komentar_id=?";
    $dewan1 = $db1->prepare($query);
    $dewan1->bind_param("s", $parent_id);
    $dewan1->execute();
    $res1 = $dewan1->get_result();

```

```

$count = $res1->num_rows;

if($parent_id == 0) {
    $marginleft = 0;
} else {
    $marginleft = $marginleft + 48;
}

```

```

$tingkat = $marginleft/48+1;

```

```

if($count > 0){
    while ($row = $res1->fetch_assoc()) {
        $output .= '
            <div class="media border p-3 mb-2" style="margin-left:'. $marginleft. 'px">
                
                <div class="media-body">
                    <div class="row">
                        <div class="col-sm-10">
                            <h4><b>'. $row["nama_pengirim"] .'</b> <small> Posted on
<i>'. $row["date"] .'</i></small></h4>
                            <p>'. $row["komentar"] .'</p>
                        </div>';
                    </div>

                    if($tingkat < 4){
                        $output .= '
                            <div class="col-sm-2" align="right">
                                <button type="button" class="btn btn-primary reply"
id="'. $row["komentar_id"] .' ">Reply</button>
                            </div>';
                    }
                </div>
            </div>';
        }
    }
}

```

```
$output .= '  
    </div>  
    </div>  
    </div>  
';  
$output .= ambil_reply($db1, $row["komentar_id"], $marginleft);  
}  
}  
  
return $output;  
}  
?>
```