

BAB 2

TINJAUAN PUSTAKA

2.1 Profile Perusahaan

Travel Tebo Mandiri Baru merupakan angkutan umum antar kota yang beroperasi di Provinsi Jambi dan berlokasi di Jln. Kapt. Pattimura No 50 RT 07 Kelurahan Rawasari, Kec. Kota Baru, Prov. Jambi.

2.1.1 Logo Tebo Mandiri Baru

Berikut ini adalah logo dari PO. CV. Tebo Mandiri Baru:



Gambar 2.1 Logo Tebo Mandiri Baru

2.1.2 Sejarah Tebo Mandiri Baru

PO. CV. Tebo Mandiri Baru berasal dari nama daerah pemekaran kabupaten baru yang ada di provinsi Jambi, perusahaan ini bergerak di bidang jasa angkutan umum antar kota yang ada di provinsi Jambi. Perusahaan ini didirikan oleh bapak Syofyan Sa'ari yang mencakup sebagai pimpinan perusahaan, lokasi kegiatan usaha PO. CV. Tebo Mandiri Baru adalah di jln. Kapt. Pattimura No 50 RT.07 Kelurahan Rawasari, Kecamatan Kota Baru Jambi.

Adapun daerah oprasional PO. CV. Tebo Mandiri Baru meliputi beberapa daerah kota dan kabupaten yang ada di provinsi Jambi yaitu sebagai berikut :

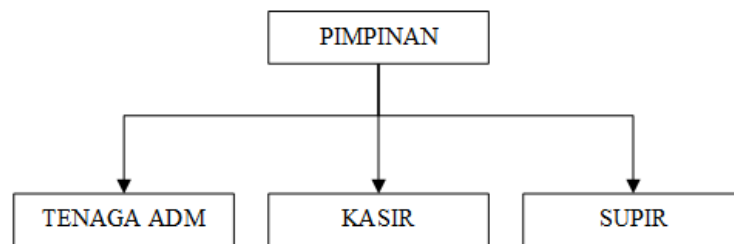
Tabel 2. 1 Daerah Oprasional Tebo Mandiri Baru

<i>NO</i>	<i>Nama Tempat</i>
<i>1</i>	<i>Batas Sumbar</i>
<i>2</i>	<i>Bangko</i>
<i>3</i>	<i>Jambi</i>
<i>4</i>	<i>Kuamang Kuning</i>
<i>5</i>	<i>Muara Bulian</i>
<i>6</i>	<i>Muara Bungo</i>
<i>7</i>	<i>Muara Tebo</i>
<i>8</i>	<i>Muara Tabir</i>
<i>9</i>	<i>Muara Tembesi</i>
<i>10</i>	<i>Pulau Temiang</i>
<i>11</i>	<i>Sarolangun</i>
<i>12</i>	<i>Sungai Bengkal</i>
<i>13</i>	<i>Tanah Garo</i>
<i>14</i>	<i>Tanjung Samalidu</i>

Pada waktu pendiriannya perusahaan ini hanya memiliki 8 armada, namun seiring berjalannya waktu perusahaan ini mulai berkembang dengan memiliki 30 karyawan dan 25 armada.

2.1.3 Struktur Organisasi Tebo Mandiri Baru

Struktur organisasi ini merupakan penggambaran secara grafik yang menggambarkan struktur kerja dari setiap departemen yang mempunyai wewenang dan tanggung jawab masing – masing yang ada di lingkungan perusahaan. Struktur organisasi yang ada di Tebo Mandiri Baru adalah sebagai berikut.



Gambar 2. 1 Struktur Organisasi

2.1.4 Deskripsi dan Tanggung Jawab

Dalam tataran praktisnya, masing - masing bagian memiliki tugas dan tanggung jawab yang berbeda - beda, yaitu sebagai berikut:

1. Pimpinan

Pimpinan atau pemilik merupakan fungsi jabatan tertinggi dalam sebuah perusahaan yang secara garis besar bertanggung jawab mengatur perusahaan secara keseluruhan. Tugas dan tanggung jawab pimpinan adalah sebagai berikut:

- a. Memutuskan dan menentukan peraturan atau kebijakan tertinggi perusahaan.
- b. Bertanggung jawab dalam memimpin dan menjalankan perusahaan.
- c. Bertanggung jawab atas kerugian dan keuntungan perusahaan.
- d. Menetapkan strategi-strategi strategis untuk mencapai visi dan misi perusahaan.
- e. Mengangkat dan memberhentikan karyawan perusahaan.

2. Tenaga ADM

Tenaga ADM atau staf administrasi dalam sebuah perusahaan adalah memastikan segala kegiatan yang bersifat administratif / ketatausahaan perusahaan agar dapat berjalan dengan baik dan lancar.

3. Kasir

Kasir adalah staf keuangan yang memiliki tugas untuk mengelolah keuangan serta melayani pembelian tiket travel.

4. Supir

Tanggung jawab seorang supir adalah:

- a. Menjaga keselamatan jiwa diri sendiri serta penumpang.
- b. Menjaga dan merawat aset perusahaan.
- c. Menjaga kerahasiaan perusahaan.
- d. Menjaga stamina diri agar tetap sehat dan selalu fresh.

2.2 Landasan Teori

Landasan Teori memberikan gambaran dan teori terkait dengan penggunaan sistem. Landasan teori ini akan menjelaskan tentang kantuk, kelelahan, detak jantung, beban kerja, mi band, bluetooth, dan android.

2.2.1 Kantuk

Kantuk didefinisikan sebagai sebuah proses yang dihasilkan dari ritme sirkadian dan kebutuhan untuk tidur. Titik awal terjadinya kantuk tidak dapat diprediksi secara pasti. Pengemudi juga tidak dapat memprediksi akan datangnya serangan kantuk. Keadaan mengantuk ketika mengemudi tergolong kedalam salah satu tindakan yang berbahaya. Kantuk dalam pekerjaan seperti mengemudi dipengaruhi oleh tiga faktor utama yaitu faktor pekerjaan, karakteristik individu, dan tidur yang dilakukannya (14).

Kantuk adalah kecenderungan untuk jatuh tertidur. Keadaan mengantuk dapat menurunkan performansi manusia, seperti berkurangnya kecepatan reaksi, penurunan kewaspadaan, dan menurunkan kemampuan memproses informasi. Oleh karena itu, keadaan mengantuk saat mengemudi tergolong dalam satu tindakan yang berbahaya [14]. Kantuk dapat mempengaruhi proses pada saat mengemudi dan performa dari pengemudi dapat turun hingga dibawah standar yang dibutuhkan oleh situasi lalu lintas.

Secara garis besar faktor yang mempengaruhi tingkat kantuk terutama pada saat mengemudi antara lain adalah:

1. Kuantitas tidur

Tidur sebelum bekerja adalah faktor yang paling mempengaruhi keadaan sadar dan tingkat kewaspadaan pengemudi. Kurang tidur yang bersifat kronis merupakan akibat dari kurangnya waktu tidur dalam jangka waktu yang lama,

sedangkan kurang tidur yang bersifat akut merupakan akibat dari kurangnya waktu tidur dalam waktu satu malam yang pendek. Kebutuhan jam tidur biasanya dipengaruhi oleh usia, semakin bertambah usia seseorang maka semakin berkurang jam tidurnya.

2. Kualitas Tidur

Kualitas tidur yang baik akan ditandai dengan tidur yang tenang, merasa sangat segar saat bangun tidur di pagi hari dan individu merasa penuh semangat untuk melakukan aktivitas hidup lainnya.

3. Status Gizi

Hasil penelitian menyatakan bahwa Indeks Massa Tubuh (IMT) tertinggi dimiliki oleh responden yang memiliki durasi tidur yang kurang sering mengantuk. Penelitian lain menyatakan adanya peningkatan IMT seiring dengan berkurangnya durasi tidur.

4. Masa Kerja

Pengemudi yang professional atau dengan masa kerja yang lebih lama, karena sifat dari pekerjaannya telah menghabiskan sebagian besar jam kerja mereka untuk mengendarai kendaraan. Selain jangka waktu mengemudi, pengemudi yang professional banyak bekerja dalam shift, memiliki jam kerja tidak teratur, dan bekerja pada malam hari.

5. Durasi Mengemudi

Durasi mengemudi telah diketahui sejak lama sebagai salah satu faktor resiko terpenting pada kejadian jatuh tertidur saat mengemudi. Meskipun durasi mengemudi bukan merupakan faktor yang krusial, sebagai contoh 60% dari kecelakaan yang berakibat fatal pada pengemudi terjadi dalam waktu 1 jam mengemudi.

2.2.2 Kelelahan

Kelelahan adalah suatu mekanisme perlindungan tubuh agar tubuh terhindar dari kerusakan lebih lanjut sehingga terjadi pemulihan setelah istirahat. Kelelahan sering dihubungkan dengan kondisi kurang tidur, kondisi akibat tidur yang

terganggu, atau kebutuhan kuat untuk tidur yang berhubungan dengan panjangnya waktu kerja, dan stres kerja yang bervariasi.

Kelelahan kerja akan menurunkan kinerja dan menambah tingkat kesalahan kerja. Meningkatnya kesalahan kerja akan memberikan peluang terjadinya kecelakaan kerja dalam industri. Pembebanan otot secara statis (static muscular loading) jika dipertahankan dalam waktu yang cukup lama akan mengakibatkan RSI (Repetition Strain Injuries), yaitu nyeri otot, tulang, tendon, dan lain-lain yang diakibatkan oleh jenis pekerjaan yang bersifat berulang.

Ada beberapa pendapat mengenai tipe kelelahan akibat kerja, yang diantaranya adalah sebagai berikut [2] :

1. Kelelahan otot (muscular fatigue), disebabkan oleh aktivitas yang membutuhkan tenaga fisik yang banyak dan berlangsung lama. Tipe ini berhubungan dengan perubahan biokimia tubuh dan dirasakan individu dalam bentuk sakit yang akut pada otot. Kelelahan ini dapat dikurangi dengan mendesain prosedur kerja baru yang melindungi individu dari pekerjaan yang terlalu berat, misalnya dengan mendesain ulang peralatan atau penemuan alat-alat baru serta melakukan sikap kerja yang lebih efisien.
2. Kelelahan mental (mental fatigue), berhubungan dengan aktivitas kerja yang monoton. Kelelahan ini dapat membuat individu kehilangan kendali akan pikiran dan perasaan, individu menjadi kurang ramah dalam berinteraksi dengan orang lain, pikiran dan perasaan yang seharusnya ditekan karena dapat menimbulkan konflik dengan individu lain menjadi lebih mudah diungkapkan. Kelelahan ini diatasi dengan mendesain ulang pekerjaan sehingga membuat karyawan lebih bersemangat dan tertantang untuk menyelesaikan pekerjaan.
3. Kelelahan emosional (emotional fatigue), dihasilkan dari stres yang hebat dan umumnya ditandai dengan kebosanan. Kelelahan ini berasal dari faktor-faktor luar di tempat kerja, perusahaan dapat mengatasi kelelahan ini dengan memberikan pelayanan konseling bagi karyawan agar kelelahan emosional yang dirasakan karyawan dapat teratasi dan performansi kerja karyawan meningkat.

2.2.3 Detak Jantung

Detak jantung biasanya mengacu pada jumlah waktu yang dibutuhkan oleh detak jantung per satuan waktu, secara umum direpresentasikan sebagai bpm (*beats per minute*). Detak jantung normal untuk setiap individu berbeda-beda tergantung pada kapan waktu mengukur detak jantung tersebut (saat istirahat atau setelah berolahraga). Variasi dalam detak jantung sesuai dengan jumlah oksigen yang diperlukan oleh tubuh saat itu.

Rentang denyut nadi normal orang dewasa yang mengacu pada standar menurut WHO (*World Health Organization*), yaitu berkisaran antara 60-100 bpm.[3] Ada banyak faktor yang mempengaruhi jumlah detak jantung seseorang, yaitu aktifitas fisik atau tingkat kebugaran seseorang, suhu udara disekitar, emosi, ukuran tubuh serta obat yang sedang dikonsumsi. Denyut nadi memiliki korelasi dengan keadaan tubuh manusia. Berdasarkan penjelasan dr. Ulfi Umroni, pada saat nadi terdeteksi di bawah normal, maka hal ini menunjukkan suplai darah kesistem tubuh berkurang. Dengan berkurangnya suplai darah ini, maka nutrisi dan oksigen yang mengalir dalam darah tidak optimal didistribusikan untuk kebutuhan sistem tubuh dan sel, hal ini akan menyebabkan kondisi tubuh lemas, dan mengantuk.

2.2.4 Beban Kerja

Tubuh manusia dirancang untuk dapat melakukan aktivitas pekerjaan sehari-hari. Adanya massa otot yang bobotnya hampir lebih dari separuh berat tubuh, memungkinkan kita untuk dapat menggerakkan tubuh dan melakukan pekerjaan. Pekerjaan di satu pihak mempunyai arti penting bagi kemajuan dan peningkatan prestasi, sehingga mencapai kehidupan yang produktif sebagai salah satu tujuan hidup. Di pihak lain, dengan bekerja berarti tubuh akan menerima beban dari luar tubuhnya. Dengan kata lain bahwa setiap pekerja merupakan beban bagi yang bersangkutan. Beban tersebut dapat berupa beban fisik maupun beban mental. Setiap beban kerja yang diterima oleh seseorang harus sesuai atau seimbang baik terhadap kemampuan fisik, kemampuan kognitif maupun keterbatasan manusia yang menerima beban tersebut [13]

Beban kerja dipengaruhi oleh berbagai faktor yang sangat kompleks, baik faktor internal maupun faktor eksternal.

1. Beban kerja karena faktor eksternal

Faktor eksternal beban kerja adalah beban kerja yang berasal dari luar tubuh pekerja. Yang termasuk beban kerja eksternal adalah tugas (task) itu sendiri, organisasi dan lingkungan kerja. Ketiga aspek ini sering disebut sebagai stressor.

- a. Tugas-tugas (tasks) yang dilakukan baik yang bersifat fisik seperti, stasiun kerja, tata ruang tempat kerja, alat dan sarana kerja, kondisi atau medan kerja, sikap kerja, cara angkat-angkut, beban yang diangkat-angkut, alat bantu kerja, sarana informasi termasuk display dan control, alur kerja dll. Sedangkan tugas-tugas yang bersifat mental seperti, kompleksitas pekerjaan atau tingkat kesulitan pekerjaan yang mempengaruhi tingkat emosi pekerja, tanggung jawab terhadap pekerjaan dll.
- b. Organisasi kerja yang dapat mempengaruhi beban kerja seperti, lamanya waktu kerja, waktu istirahat, kerja bergilir, kerja malam, sistem pengupahan, sistem kerja, musibah kerja, model struktur organisasi, pelimpahan tugas dan wewenang dll.
- c. Lingkungan kerja yang dapat memberikan beban tambahan kepada pekerja di antaranya seperti lingkungan kerja fisik, lingkungan kerja kimiawi, lingkungan kerja biologis dan lingkungan kerja psikologis.

2. Beban kerja karena faktor internal

Faktor internal beban kerja adalah faktor yang berasal dari dalam tubuh itu sendiri sebagai akibat adanya reaksi dari beban kerja eksternal. Reaksi tubuh tersebut dikenal sebagai strain. Berat ringannya strain dapat dinilai baik secara objektif maupun subjektif. Penilaian secara objektif yaitu melalui perubahan reaksi fisiologis. Sedangkan penilaian subjektif dapat dilakukan melalui perubahan reaksi psikologis dan perubahan perilaku. Karena itu strain secara subjektif berkait erat dengan harapan, keinginan, kepuasan dan penilaian subjektif lainnya. Secara lebih ringkas faktor internal meliputi:

- a. Faktor somatis (jenis kelamin, umur, ukuran tubuh, kondisi kesehatan, status gizi).
- b. faktor psikis (motivasi, persepsi, kepercayaan, keinginan, kepuasan dll.).

2.2.5 Mi Band 2

Mi Band 2 adalah smartwatch yang diproduksi oleh perusahaan Xiaomi. Fitur yang digunakan pada smartwatch ini yaitu sebagai berikut:[4]

1. Water Resistant (anti air) dengan spesifikasi IP-67 yang juga tahan dengan debu.
2. Memiliki sensor detak jantung.
3. Sport Tracker yang dapat memperlihatkan aktivitas pengguna.
4. Sleep tracking yang dapat mendeteksi seberapa besar kualitas tidur pengguna.
5. Calories Tracker yang dapat melihat jumlah kalori yang terbakar akibat aktivitas pengguna.

Mi Band 2 sendiri hanya mendukung perangkat yang berbasis minimal android 5.0 dan bluetooth 4.0.

2.2.6 Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2.4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping transceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan terbatas. Bluetooth 4.0 adalah teknologi Bluetooth versi terbaru dengan keunggulan utama hemat energy yang resmi diadopsi oleh Bluetooth Special Interest Group (SIG) pada tahun 2010 [5]. Bluetooth 4.0 memiliki beberapa keunggulan yaitu:

1. Konsumsi daya yang lebih kecil sehingga hemat baterai atau listrik.
2. Waktu pemakaian yang lebih lama sampai tahunan dengan hanya menggunakan baterai kapasitas kecil (coin cell batteries).
3. Biaya produksi yang rendah.
4. Jarak yang lebih panjang. Mampu menjangkau lebih dari 100 meter.

5. Kecepatan transfer yang lebih baik dengan kemampuan sekitar 1 Mbps.
6. Sistem keamanan yang lebih terkontrol dibandingkan dengan sistem keamanan pada Bluetooth generasi-generasi sebelumnya.

2.3 Android

Android merupakan sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi.

2.3.1 Sejarah Android

Android dikembangkan oleh Android Inc. Perusahaan ini kemudian di beli oleh Google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi Open Handset Alliance tahun 2007. Selain Google, beberapa nama-nama besar juga ikut serta dalam Open Handset Alliance, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel.[6]

2.3.2 Versi Android

Adapun versi android yang pernah dirilis adalah sebagai berikut :

1. Android versi 1.1
Versi 1.1 dirilis pada 9 maret 2009, android ini dilengkapi dengan pembaruan estetis pada aplikasi, alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
2. Android versi 1.5 (Cupcake)
Pada pertengahan bulan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit). Terdapat banyak pembaruan yakni mampu merekam dan menonton video dengan menggunakan kamera, mengupload video ke youtube dan gambar ke picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung otomatis ke headset Bluetooth.
3. Android v1.6 Donut

Pada September 2009, google kembali merilis android versi 1.6 (Donut). Ini merupakan rilis minor dengan fitur utama berupa dukungan teknologi CDMA, gesture, text-to-speech engine, resolusi VWGA, VPN, dan fitur pencarian cepat.

4. Android v2.0-2.1 Eclair

Pada 3 Desember 2009 Google kembali meluncurkan Android dengan versi 2.0/2.1 (Eclair). Mengusung fitur baru yaitu live wallpaper, pengoptimalan hardware, peningkatan Google Maps 3.1.2, Bluetooth 2.1, dan profil bluetooth baru yaitu Object Push Profile (OPP) dan Phone Book Access Profile (PBAP).

5. Android v2.2 Frozen Yoghurt (Froyo)

Pada 20 Mei 2010 dirilis Android 2.2 (Froyo). Fitur baru yang dibawa adalah Global Positioning System (GPS), kompas, kamera, dan Accelorometer.

6. Android v2.3 Gingerbread

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Mengusung beberapa fitur baru, antara lain dukungan banyak kamera, Near field Communication (NFC), download manager *service*, dukungan terhadap sensor lain seperti giroskop, dan barometer.

7. Android v3.0-3.2 Honeycomb

Android Honeycomb dirancang khusus untuk tablet diluncurkan pada 22 Februari 2011. Android versi ini mendukung ukuran layar yang lebih besar. Pada Honeycomb mendukung multi prosesor dan akselerasi perangkat keras (Hardware) untuk grafis.

8. Android v4.0 Ice Cream Sandwich

Pada 19 Oktober 2011, Android versi 4.0 (Ice Cream Sandwich) diluncurkan. Pada versi ini membawa perubahan besar pada tampilan User Interface Android 3.0 agar cocok pada layar kecil. Fitur baru seperti Android Beam dan Wi-Fi Direct juga ditambahkan.

9. Android v4.1-4.3 Jelly Bean

Pada 27 Juni 2012, Android versi 4.1 (Jelly Bean) diluncurkan. Pada versi ini merupakan rilis minor yang membawa perbaikan fungsionalitas dan performa

rendering User Interface (UI). Versi 4.2 mengusung fitur baru seperti gesture typing dan dukungan multiuser di piranti tablet. Versi 4.3 membawa perbaikan berupa dukungan Bluetooth Low Energy dan OpenGL ES 3.0. beberapa fitur baru yaitu dukungan terhadap bahasa internasional dan penulisan teks dua arah.

10. Android v4.4 Kitkat

Android versi 4.4 (Kitkat) diluncurkan pada 3 September 2013. Pada versi ini mengusung sejumlah perbaikan dan fitur baru yaitu dukungan Near Field Communication melalui host card emulation, pencetakan printer nirkabel, WebView dengan rendering engine Chromium, dan dukungan yang lebih baik bagi piranti yang menggunakan Random Access Memory (RAM) rendah.

11. Android versi 5.0 (Lollipop)

Pertama kali diperkenalkan di bawah codename "Android L" pada 25 Juni 2014 selama presentasi keynote pada konferensi pengembang Google I / O. Di samping Lollipop, presentasi difokuskan pada sejumlah platform Android yang berorientasi dan teknologi baru, termasuk Android TV, pada platform Android Auto, dpt dipakai pada platform komputasi Android Wear, dan platform pelacakan kesehatan Google Fit

12. Android Marshmallow

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

2.3.3 Arsitektur Android

Android terdiri dari lapisan software, yaitu lapisan aplikasi, lapisan framework, Android run-time sebagai lapisan perantara (middleware), serta kernel linux yang membungkus hardware. Berikut penjelasan arsitektur dari android:

1. Aplikasi

Aplikasi yang dibuat berada pada level tertinggi dalam hierarki struktur software di android.

2. Framework Aplikasi

Di bawah aplikasi terdapat sejumlah software pendukung, meliputi layanan pengelola Activity, View, sumber daya, pemberitahuan, dan lain-lain.

3. Pustaka

Android menyediakan sejumlah pustaka dasar seperti pustaka 2D dan 3D, multimedia playback, browser engine, pencetakan font, *Database*, dan lain-lain. Aplikasi yang memanfaatkan ini melalui framework aplikasi.

4. Android Runtime

Android menyediakan pustaka inti bagi pemrograman Java. Di bawahnya terdapat Dalvik Virtual Machine (Dalvik VM) yang akan menjalankan aplikasi. Tiap aplikasi akan dilajankan pada proses terpisah dengan VM yang berbeda sehingga terisolasi satu sama lain. Aplikasi masih dapat berkomunikasi dengan aplikasi lain melalui Framework aplikasi. Dalvik VM bergantung pada kernel linux untuk multi-threading dan pengelolaan memori tingkat rendah.

5. Kernel Linux

Kernel Linux bertanggung jawab menyediakan layanan dasar seperti keamanan, pengelolaan, proses, pengelolaan file, pengelolaan sumber daya memori, dan hardware.

2.4 *Web service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk

menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *Web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.[11]

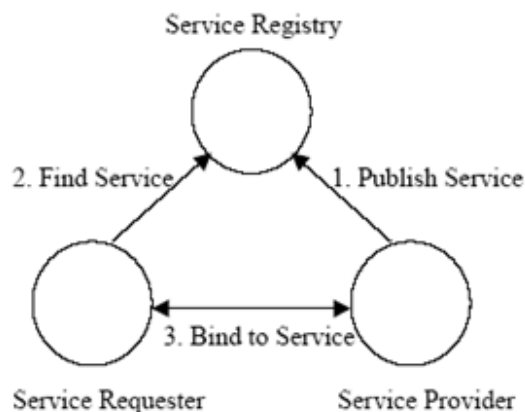
Beberapa alasan mengapa digunakannya *Web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau *Class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *Web service* tidak memerlukan konfigurasi khusus di sisi firewall.

2.4.1 Arsitektur *Web service*

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. *Service Requester* (peminta layanan)
2. *Service Provider* (penyedia layanan)
3. *Service Registry* (daftar layanan)



Gambar 2. 2 Arsitektur Web service

1. *Service Provider*: Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
2. *Service Registry*: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-register.
3. *Service Requestor*: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

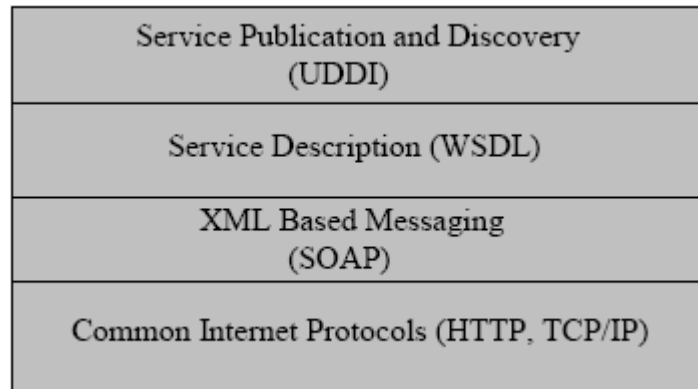
2.4.2 Operasi - Operasi Web service

Secara umum, *Web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1. *Publish/Unpublish*: Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*.

2.4.3 Komponen Web service

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di bawah, yaitu:



Gambar 2. 3 Komponen Web service

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3: *Web service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layer 4 :*Universal Description, Discovery, and Integration* (UDDI), merupakan lokasi direktori berisikan *service* dan bersifat bebas platform

2.5 Struktur Analisis dan Disain

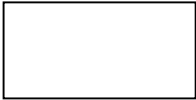
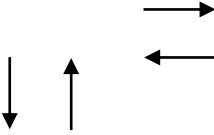
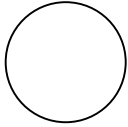
Structured analysis and design adalah pendekatan sistem analisis dan desain sistem informasi secara hierarki. SADM diproduksi untuk CCTA, kantor pemerintah Inggris yang berkaitan dengan penggunaan teknologi di pemerintahan dari tahun 1980 dan seterusnya[16].

2.5.1 Diagram Konteks

Diagram Context, adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem)[17].

Berikut ini merupakan simbol - simbol yang ada pada diagram konteks.

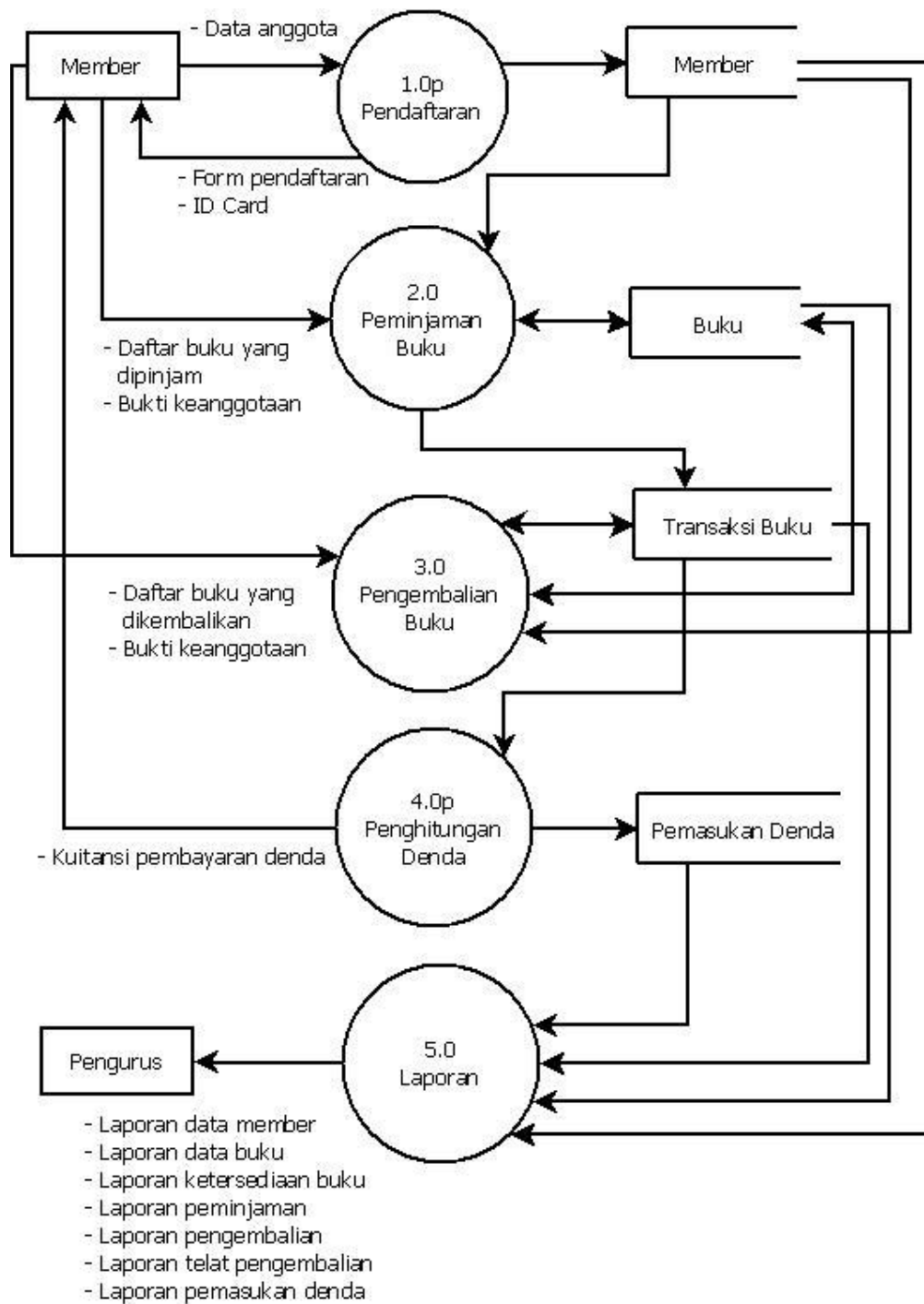
Tabel 2. 2 Simbol Diagram Konteks

Simbol	Pengertian	Keterangan
	Eksternal <i>Entity</i>	Menunjukkan bagian luar sistem atau sumber <i>Input</i> dan <i>output</i>
	Garis aliran	Menunjukkan arus sata antar simbol/proses
	Proses transformasi	Proses yang mengubah <i>Input</i> menjadi <i>output</i>

2.5.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya

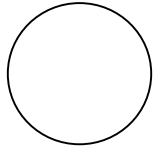

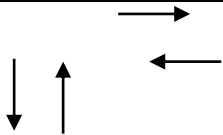
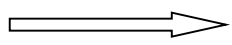
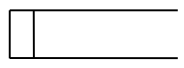

bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program[17]. Berikut ini merupakan contoh data flow diagram:


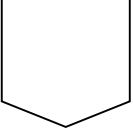


Gambar 2. 4 Contoh Data Flow Diagram

Berikut ini merupakan simbol - simbol yang ada pada data flow diagram atau DFD.

Tabel 2. 3 Simbol Data Flow Diagram DFD

Simbol	Pengertian	Keterangan
	Sistem	Menunjukkan sistem
	Eksternal <i>entity</i>	Menunjukkan bagian luar sistem atau sumber <i>Input</i> dan <i>output</i> data
	Garis aliran	Menunjukkan arus data antar simbol / proses
	Garis aliran	Aliran material
	Data Storage	Digunakan untuk menyimpan arus data atau arsip seperti file transaksi, file induk atau file referensi dan lain-lain
	Proses	Suatu proses yang dipicu atau didukung oleh data.

	Conector (On-page connector)	Digunakan untuk penghubung dalam satu halaman
	Conector (Off-page connector)	Digunakan untuk penghubung berbeda halaman

2.5.3 Spesifikasi Proses

Spesifikasi Proses adalah tabel yang berisi keterangan atau deskripsi dari semua proses yang terdapat pada DFD. Logika proses yang ditulis pada spesifikasi proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau *pseudocode* tetapi tidak boleh mengkombinasikan kedua bahasa tersebut[17].

Tabel 2. 4 Contoh Spesifikasi Proses

<i>No</i>	<i>Proses</i>	<i>Keterangan</i>
1.	<i>No Proses</i>	<i>Menyatakan no proses</i>
2.	<i>Nama Proses</i>	<i>Menyatakan nama proses</i>
3.	<i>Source (Sumber)</i>	<i>(Menyatakan sumber data Input menuju proses)</i>
4.	<i>Input</i>	<i>(menyatakan isi data yang masuk ke proses)</i>
5.	<i>Output</i>	<i>(menyatakan informasi yang keluar dari proses)</i>
6.	<i>Destination</i>	<i>(menatakan tujuan informasi output dari proses)</i>
7.	<i>Logika Proses</i>	<i>(menyatakan algoritma dari proses)</i>

2.5.4 Kamus Data

Kamus data digunakan untuk memperjelas aliran data yang di gambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami secara umum.

Tabel 2. 5 Contoh Kamus Data

<i>Nama aliran data</i>	<i>Data Siswa Nama dari data yang digunakan</i>
<i>WHERE used/How used</i>	<i>Daftar pada poin proses apa data digunakan</i>
<i>Keterangan</i>	<i>Uraian singkat dari data yang digunakan</i>
<i>Struktur Data</i>	<i>Daftar komponen data yang ada pada data yang digunakan</i>
<i>Deskripsi</i>	<i>Jenis data dalam representasi komputer untuk masing-masing data.</i>

2.6 Pemograman Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem. Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstarkasikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek yang akan dienkapsulasikan menjadi suatu kesatuan. Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu [7].

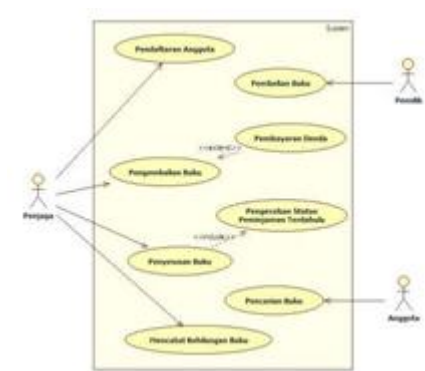
2.7 UML (Unified modeling language)

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan

singkatan dari *Unified modeling language*. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML. Diagram UML yang akan digunakan dalam memvisualisasikan perangkat lunak yang akan dibangun antara lain : *Use case diagram, Class diagram, Activity Diagram, Sequence diagram*. [18]

2.7.1 Use case diagram








Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) system secara keseluruhan yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut dan yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case* menjelaskan secara sederhana fungsi sistem dari sudut pandang user. Berikut ini merupakan contoh *Use case diagram*:




Gambar 2. 5 Contoh Use case diagram

Berikut ini merupakan simbol - simbol yang ada pada *Use case diagram*.

Tabel 2. 6 Simbol Use Case Diagram

GAMBAR	NAMA	KETERANGAN
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	Include	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

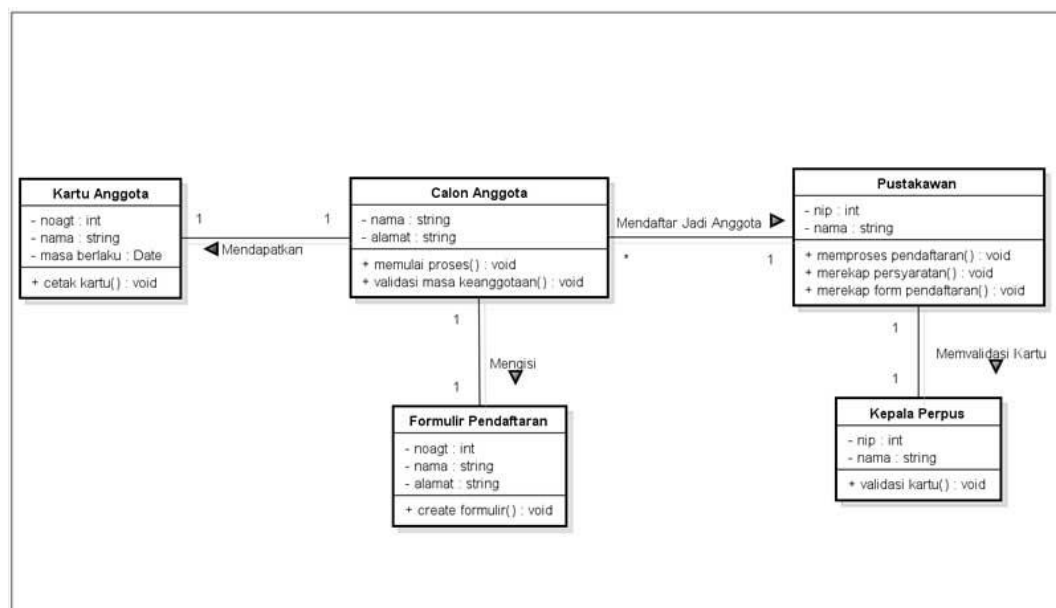
	Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
---	----------	---

2.7.2 Class diagram

Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi.[18] Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti *public*, *private* dan *protected*. Selama tahap desain, *Class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok:

1. Nama (dan stereotype)
2. Atribut
3. Metoda

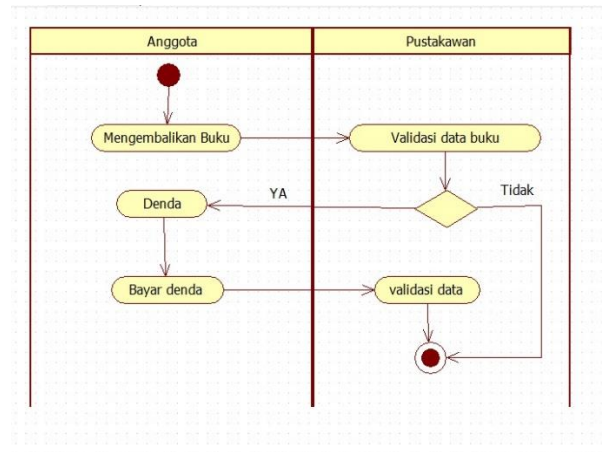
Berikut ini merupakan contoh dari *Class diagram*:



Gambar 2. 6 Contoh Class diagram

2.7.3 Activity Diagram

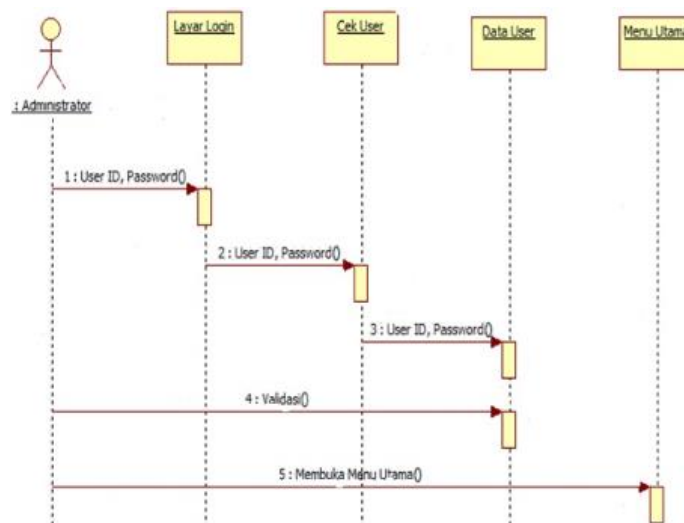
Activity Diagram menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Berikut contoh activity diagram



Gambar 2. 7 Contoh Activity Diagram

2.7.4 Sequence diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *Sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *Use case diagram*, berikut contoh *Sequence diagram*.



Gambar 2. 8 Contoh Sequence diagram

2.8 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*“Database management system”* DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan user untuk memelihara, mengontrol, dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh DBMS. Ada beberapa fungsi yang harus ditangani DBMS yaitu mengolah pendefinisian data, dapat menangani permintaan pemakai untuk mengakses data, memeriksa *security* dan *integrity* data yang didefinisikan oleh DBA (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang

disebabkan oleh kerusakan sistem maupun disk, dan menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada user (pengguna). Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil (akses) secara efisien. Pertimbangan efisien disini adalah bagaimana merancang struktur data yang kompleks tetapi masih tetap bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya. *Database* dibagi menjadi dua jenis yaitu:

1. Basis Data Flat-File

Basis data flat-file ideal untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada dasarnya, mereka tersusun dari sekumpulan string dalam satu atau lebih file yang dapat diurai untuk mendapatkan informasi yang disimpan. Basis data flat-file baik digunakan untuk menyimpan daftar atau data yang sederhana dan dalam jumlah kecil. Basis data flat-file akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks walaupun dimungkinkan pula untuk menyimpan data semacam itu. Salah satu masalah menggunakan basis data jenis ini adalah rentan pada korupsi data karena tidak adanya penguncian yang melekat ketika data digunakan atau dimodifikasi.

2. Basis Data Relasional

Basis data ini mempunyai struktur yang lebih logis terkait cara penyimpanan. Kata "relasional" berasal dari kenyataan bahwa tabel-tabel yang berada di basis data dapat dihubungkan satu dengan lainnya. Basis data relasional menggunakan sekumpulan tabel dua dimensi yang masing-masing tabel tersusun atas baris (tupel) dan kolom (atribut). Untuk membuat hubungan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu primary key di salah satu tabel dan foreign key di tabel yang lain. Saat ini, basis data relasional menjadi pilihan karena keunggulannya. Beberapa kelemahan yang mungkin dirasakan untuk basis data jenis ini adalah implementasi yang lebih sulit untuk data dalam jumlah besar dengan tingkat kompleksitasnya yang tinggi dan proses pencarian informasi yang lebih lambat karena perlu

menghubungkan tabel-tabel terlebih dahulu apabila datanya tersebar di beberapa tabel.

2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *Database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.[8]

Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu *MySQL* AB. *MySQL* AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL* AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Adapun fitur-fitur *MySQL* antara lain :

1. *Relational Database System*. Seperti halnya *software Database* lain yang ada di pasaran, *MySQL* termasuk RDBMS.
2. *Arsitektur Client-Server*. *MySQL* memiliki arsitektur *client-server* dimana *server Database MySQL* terinstal di *server*. *Client MySQL* dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan *internet*.
3. Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software Database*. *MySQL* mendukung SQL versi SQL:2003.
4. Mendukung Sub *SELECT*. Mulai versi 4.1 *MySQL* telah mendukung *SELECT* dalam *SELECT (sub SELECT)*.
5. Mendukung *Views*. *MySQL* mendukung *views* sejak versi 5.0
6. Mendukung *Stored Prosedured (SP)*. *MySQL* mendukung SP sejak versi 5.0.

7. Mendukung *Triggers*. *MySQL* mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang *MySQL* berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung *foreign key*.

Perintah - perintah yang terdapat di SQL antara lain adalah:

1. Menambah data (*INSERT*)

Perintah SQL yang digunakan untuk menambah data ke *Database* adalah perintah *INSERT*. Perintah ini mempunyai bentuk umum sebagai berikut:

INSERT [*INTO*] nama_tabel [(daftar *field*/kolom)] *VALUES* (daftar_data).

2. Mengakses data (*SELECT*)

Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*.

Bentuk umum perintah ini adalah sebagai berikut:

SELECT daftar_ *SELECT FROM* daftar_tabel

[*WHERE* kondisi_pencarian]

[*GROUP BY* daftar_group_by]

[*HAVING* kondisi_pencarian]

[*ORDER BY* daftar_order[*ASC* | *DESC*]]

3. Mengedit/Mengubah data (*Update*)

Perintah SQL yang digunakan untuk mengubah data adalah perintah

UPDATE SET. Perintah ini mempunyai bentuk umum sebagai berikut:

UPDATE nama_table *SET* field1=databaru1

[data2=databaru2]

[*WHERE* kondisi_update].

4. Menghapus data (*Delete*)

Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

DELETE FROM tabel_hapus *WHERE* kondisi_hapus.

2.10 Perangkat Lunak Pendukung

Berikut adalah perangkat lunak pendukung dalam pembangunan aplikasi yang di bangun :

2.10.1 Android Studio

Android Studio adalah *Integrated Development Environment* resmi untuk pengembangan aplikasi Android dan bersigat open *source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android . Android Studio digunakan pada pembangunan aplikasi ini di karenakan IDE resmi untuk pembangunan aplikasi android .

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup [9]:

1. Modul aplikasi Android
2. Modul Pustaka
3. Modul Google App Engine

Secara default, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek anda.

2.10.1.1 Fitur Android Studio

Android Studio juga menawarkan fitur lebih banyak untuk meningkatkan produktivitas anda saat membuat aplikasi android, misalnya:

1. Sistem versi berbasis gradle yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.

5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

2.10.1.2 Versi Rilis Android Studio

Android Studio yang kita kenal selama ini terdiri empat jenis rilis yaitu *canary*, *beta*, *rc*, dan *stable*. Perbedaan masing-masing versi rilis Android Studio adalah:

1. Versi Canary
2. Versi Canary akan mendapatkan semua jenis rilis. Artinya saat ada fitur baru yang dibuat maka versi Canary akan bisa langsung mencoba fitur tersebut tanpa perlu menunggu lama. *Update* untuk Canary biasanya dirilis setiap minggu. Di versi ini kita kemungkinan akan menemukan bug di sana sini karena fitur-fitur baru yang eksperimental dan belum sempurna sudah langsung didapatkan.
3. Versi Beta
4. Jika ingin menggunakan Android Studio yang memiliki fitur baru tapi dengan resiko *bug* yang kecil, gunakanlah versi Beta. Rilis Beta biasanya sudah mengikutsertakan fitur-fitur baru namun dengan bug yang jauh lebih sedikit.
5. Versi Stable
6. Setelah semua fitur mengalami pengujian secara penuh melalui versi-versi sebelumnya, maka kita dapat menggunakan Android Studio yang minim masalah lewat versi Stable. Melalui versi ini kita diberikan garansi Android Studio yang memiliki stabilitas tinggi dan performa yang baik.

2.10.2 Google Maps API

Google maps merupakan layanan dari google yang mempermudah penggunaanya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan *Google Maps API* memungkinkan pengembangan untuk mengintegrasikan Google Maps ke dalam situs web. Dengan menggunakan *Google Maps API* memungkinkan untuk menanamkan situs Google Maps ke dalam situs eksternal, dimana situs data tertentu dapat dilakukan overlay Meskipun pada awalnya hanya Javascript Api, API Maps sejak diperluas untuk menyertakan sebuah API untuk Adobe Flash aplikasi, layanan untuk mengambil gambar peta status, dan layanan web untuk melakukan geocoding, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil elevasi.

Kelas kunci dalam perpustakaan Maps adalah *MapView*, sebuah *subClass* dari *ViewGroup* dalam standar perpustakaan Android. Sebuah *MapView* menampilkan peta dengan data yang diperoleh dari layanan Google Maps. Bila yang ditekan dan gerakan sentuh untuk pan dan zoom peta secara otomatis, termasuk penanganan permintaan jaringan untuk ubin peta tambah. Ini juga menyediakan semua elemen UI yang diperlukan bagi pengguna untuk mengendalikan peta. Aplikasi tersebut juga dapat menggunakan metode *MapView* kelas untuk mengontrol *MapView* secara terprogram dan menarik sejumlah jenis tampilan di atas peta.

Secara umum, kelas *MapView* menyediakan pembungkus di Google Maps API yang memungkinkan aplikasi tersebut memanipulasi data Google Maps melalui metode kelas, dan itu memungkinkan dikerjakan dengan data Maps seperti jenis lain Views.

Google Maps API sendiri menyediakan fungsi yang sangat banyak, berikut ini adalah pembagiannya

1. Google maps Javascript API
2. Google Places API Webservice

Google Places API *Web service* adalah layanan yang mengembalikan informasi tentang tempat - yang didefinisikan dalam API ini sebagai tempat

usaha, lokasi geografis, atau tempat menarik yang menonjol - dengan menggunakan permintaan HTTP.

3. Google Maps API Direction

Google Maps Directions API Adalah layanan yang menghitung arah antar lokasi. Anda bisa menelusuri arah untuk beberapa moda transportasi, termasuk angkutan umum, mengemudi, berjalan atau bersepeda.

4. Google Maps API Geocoding

Geocoding adalah proses mengonversi alamat (seperti alamat jalan) menjadi koordinat geografis (seperti lintang dan bujur), yang bisa Anda gunakan untuk menempatkan marker pada peta, atau memosisikan peta.

5. Google Maps Distance Matrix API

Google Maps Distance Matrix API Adalah layanan yang menyediakan waktu dan jarak perjalanan untuk matriks tempat asal dan tujuan, berdasarkan rute yang direkomendasikan di antara titik awal dan akhir.

6. Google Maps Elevation API

Google Maps Elevation API menyediakan data elevasi untuk semua lokasi di permukaan bumi, termasuk lokasi kedalaman di dasar laut.