

BAB 2

TINJAUAN PUSTAKA

2.1. Profil Dinas Perhubungan Kota Sukabumi

Profil Dinas Perhubungan Kota Sukabumi meliputi sejarah, logo, visi, misi dan pemaparan dari struktur organisasi berikut dengan tugas dan tanggung jawab setiap divisinya.

2.1.1 Sejarah Dinas Perhubungan Kota Sukabumi

Keberadaan Dinas Perhubungan Kota Sukabumi dimulai pada sekitar tahun 1960 dengan terbentuknya Departemen Pekerjaan Umum yang mempunyai bidang garapan mengenai Sarana dan Prasarana Jalan. Kemudian tahun 1960-1970 dibentuklah DLLD (Djawatan Lalu Lintas Djalan), dan setelah adanya ejaan yang disempurnakan, pada tahun 1970-1980 DLLD diubah menjadi DLLAJR (Dinas Lalu Lintas Angkutan Jalan Raya), dan dari tahun 1980 sampai tahun 2000 nama DLLAJR tersebut diubah lagi menjadi DLLAJ.

Dengan adanya Undang-undang nomor 22 tahun 1999 tentang Pemerintah Daerah, maka sejak tahun 2000 untuk urusan Pekerjaan Umum bidang jalan dan urusan Perhubungan Darat menjadi urusan Dinas Perhubungan. Dalam melaksanakan tugasnya, Dinas Perhubungan Kota Sukabumi mempunyai kewenangan yang telah diatur berdasarkan Peraturan Daerah nomor 14 tahun 2000 (lembaran Daerah Tahun 2000 Nomor 18 Seri D – 9) sebagaimana telah diubah dengan Peraturan Daerah Kota Sukabumi Nomor 3 Tahun 2003 (Lembaran Daerah Tahun 2003 Nomor 25 Seri D – 10). tentang Kewenangan Pemerintah Kota Sukabumi dan dengan Keputusan Walikota Sukabumi Nomor 32 Tahun 2016 Tentang Kedudukan, Tugas Pokok Fungsi dan Tata Kerja Dinas Perhubungan Kota Sukabumi.

Dinas Perhubungan Kota Sukabumi merupakan gabungan (merger) antara Dinas Lalu Lintas dan Angkutan Jalan (DLLAJ) dengan Dinas Pekerjaan Umum Bina Marga (DPU Bina Marga). Sebagaimana yang telah diatur dalam Peraturan Daerah Nomor 8 Tahun 2003 tentang pembentukan dan susunan organisasi perangkat Daerah dan sekretariat Dewan Perwakilan Rakyat Kota Sukabumi, maka

Dinas Pekerjaan Umum Bina Marga mempunyai kedudukan sebagai Bidang Jalan pada Dinas Perhubungan dan DLLAJ menjadi Bidang Lalu Lintas dan Angkutan serta Bidang Keselamatan Lalu Lintas Perlengkapan Jalan Sarana dan Prasarana.

Selanjutnya berdasarkan PP No. 41 Tahun 2004 yang diperkuat melalui Perda Kota Sukabumi Nomor 6 Tahun 2008, maka terjadi kembali perubahan terhadap struktur organisasi Dinas Perhubungan Kota Sukabumi dengan bergabungnya Bidang Jalan dan Jembatan dibawah naungan Dinas Pekerjaan Umum, maka Bidang tugas Dinas Perhubungan hanya meliputi Bidang Teknik, Sarana dan Prasarana Lalu Lintas serta Bidang Lalu Lintas dan Angkutan.

Pada awal tahun 2013 Dinas Perhubungan Kota Sukabumi kembali bergabung antara Dinas Perhubungan dengan Dinas Pekerjaan Umum Bina Marga. Sebagaimana yang telah diatur dalam Peraturan Daerah Nomor 16 Tahun 2012 tentang pembentukan dan susunan organisasi perangkat daerah serta Peraturan Walikota Sukabumi Nomor 38 Tahun 2016 tentang Kedudukan, Tugas Pokok, Fungsi dan Tata Kerja Dinas Perhubungan Kota Sukabumi.

2.1.2 Logo Dinas Perhubungan Kota Sukabumi

Logo merupakan merupakan suatu gambar atau sekadar sketsa dengan arti tertentu, dan mewakili suatu arti dari perusahaan, daerah, organisasi, produk, negara, lembaga, dan hal lainnya membutuhkan sesuatu yang singkat dan mudah diingat sebagai pengganti dari nama sebenarnya [13].

Menurut Andreas Ledewyk Sagala logo adalah tanda, lambang, ataupun simbol yang mengandung makna dan digunakan sebagai identitas sebuah organisasi, perusahaan atau individu agar mudah diingat oleh orang lain [14].

Logo Dinas Perhubungan Kota Sukabumi sendiri menggunakan standar logo dari keluarga besar Dinas Perhubungan di Berbagai kota. Berikut logo DISHUB kota Sukabumi pada gambar dibawah :



Gambar 2.1 Logo dinas perhubungan

Logo Perhubungan adalah suatu bentuk simbolis yang menggambarkan keluarga besar Perhubungan. Logo terdiri dari bentuk lingkaran mempunyai unsur-unsur roda bergigi, jangkar, burung Garuda, dan bulatan bumi. Arti dari unsur logo ialah [15] :

- a. Roda bergigi berarti matra Perhubungan Darat
- b. Jangkar berarti matra Perhubungan Laut
- c. Burung Garuda berarti matra Perhubungan Udara
- d. Bulatan bumi berarti lingkup pelayanan jasa Perhubungan
- e. Warna logo terdiri dari warna biru langit (*cerulean blue*) berarti kedamaian dan kuning berarti keagungan

2.1.3 Visi Dinas Perhubungan Kota Sukabumi

Perumusan dan penjelasan terhadap Visi dimaksud, menghasilkan pilar-pilar Visi yang diterjemahkan pengertiannya, sebagaimana tabel di bawah ini:

- a. Menjadi perekat dan menyatukan berbagai gagasan strategi.
- b. Memiliki orientasi ke masa depan.
- c. Menumbuhkan komitmen seluruh jajaran Dinas Perhubungan Kota Sukabumi dan juga *stake holders*.
- d. Menjamin kesinambungan kepemimpinan pada Dinas Perhubungan kota Sukabumi.

2.1.4 Misi Dinas Perhubungan Kota Sukabumi

Sesuai Visi dan Misi yang telah ditetapkan dan tugas yang harus diemban dan dilaksanakan oleh Dinas Perhubungan, telah disusun pula Misi Dinas yang akan dipergunakan sebagai landasan tujuan utama ke arah mana perencanaan/program Dinas ingin dicapai. Misi Dinas Perhubungan Kota Sukabumi :

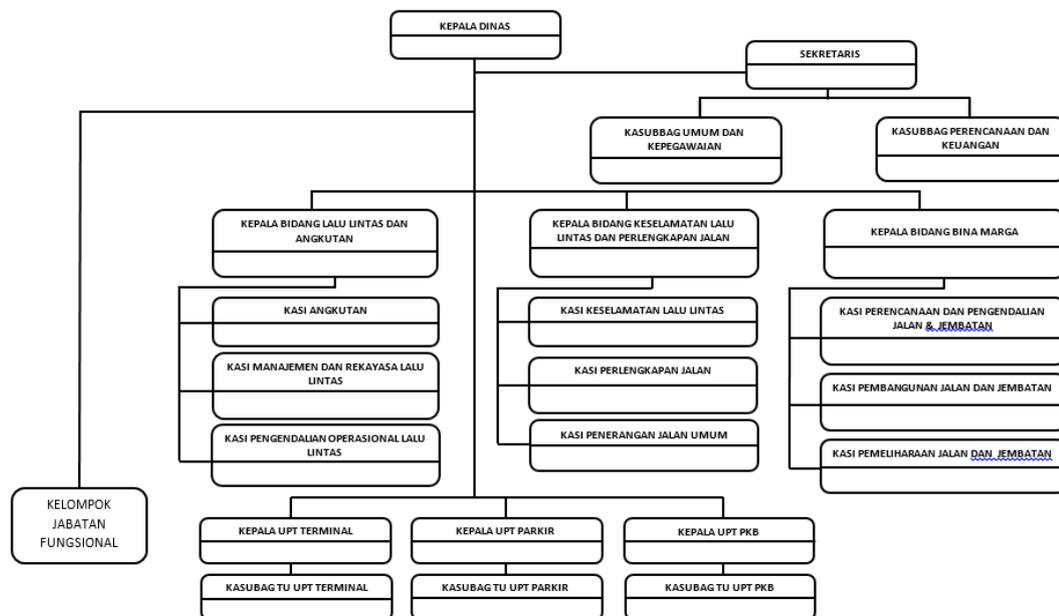
1. Mewujudkan insan perhubungan yang profesional.
2. Meningkatkan sarana, prasarana dan fasilitas pendukung transportasi yang berkualitas serta berkelanjutan.
3. Meningkatkan sistem pelayanan transportasi yang selamat, tertib, nyaman serta berwawasan lingkungan.

2.1.5 Struktur Organisasi Dinas Perhubungan Kota Sukabumi

Struktur Organisasi adalah suatu susunan dan hubungan antara setiap bagian dan juga posisi yang ada pada suatu perusahaan dalam menjalankan suatu kegiatan operasional untuk mencapai suatu tujuan. Struktur Organisasi juga menggambarkan dengan jelas pemisahan suatu kegiatan pekerjaan antara yang satu dengan yang lainnya. Dalam struktur organisasi harus ada yang berwenang siapa yang melapor [16]. Ada empat elemen yang ada dalam struktur organisasi yaitu :

1. Adanya spesialisasi kerja
2. Adanya standarisasi kegiatan kerja
3. Terbentuknya koordinasi dalam kegiatan kerja
4. Seluruh organisasi memiliki tanggung jawab yang sama

Struktur organisasi juga dapat membantu manajemen suatu perusahaan dalam mencapai tujuannya. Struktur organisasi juga menjelaskan bagaimana tugas akan dibagi dan dikelompokkan dan dikoordinasikan. Dalam struktur organisasi di Dinas Perhubungan Kota Sukabumi terdapat Kepala Dinas yang bertanggung jawab untuk memimpin dinas perhubungan di bantu oleh sekretaris dan kepala bidang pada tiap bagian.



Gambar 2.2 Struktur organisasi DISHUB kota Sukabumi

2.1.6 Tugas dan Tanggung Jawab

Tugas dan tanggung jawab kedudukan dalam struktur organisasi Dinas Perhubungan Kota Sukabumi adalah sebagai berikut :

1. Kepala Dinas

Dinas Perhubungan Kota Sukabumi dipimpin oleh seorang Kepala Dinas. Kepala Dinas mempunyai tugas sebagai berikut :

- a. Membantu Kepala Daerah dalam merumuskan kebijakan umum dan teknis bidang perhubungan yang meliputi penanganan lalu lintas, angkutan, teknik, sarana lalu lintas, prasarana jalan, dan penerangan jalan umum.
- b. Memimpin, mengkoordinasikan, mengawasi, mengevaluasi dan mengendalikan semua kegiatan Dinas.
- c. Membuat Rencana Strategis (RENSTRA), rencana kerja (RENJA), laporan akuntabilitas kinerja dalam rangka pelaksanaan tugas Dinas, laporan Penyelenggaraan Pemerintahanan Daerah, dan laporan kinerja pertanggungjawaban.
- d. Mengadakan kerja sama dengan dinas/instansi/lembaga terkait lainnya untuk kelancaran pelaksanaan tugas.

- e. Membina dan memotivasi kepada seluruh pegawai di Lingkungan Dinas dalam upaya meningkatkan efektivitas dan produktivitas kerja.
- f. Menyelenggarakan pembinaan teknis di bidang perhubungan.
- g. Mengkaji dan menyiapkan bahan penetapan kebijakan Kepala Daerah di bidang perhubungan.
- h. Mengkaji dan Mengoreksi perizinan/rekomendasi dan sertifikasi di bidang perhubungan.
- i. Memberi saran dan pertimbangan kepada Kepala Daerah di bidang perhubungan.
- j. Mempertanggungjawabkan tugas Dinas secara teknis operasional kepada Kepala Daerah melalui Wakil Kepala Daerah.
- k. Mempertanggungjawabkan tugas Dinas secara administratif kepada Kepala Daerah melalui Sekretaris Daerah dan
- l. Melaksanakan tugas lain yang diberikan oleh Kepala Daerah.
Dalam melaksanakan tugasnya, Kepala Dinas dibantu oleh:
 - a. Sekretariat.
 - b. Bidang Lalu Lintas dan Angkutan.
 - c. Bidang Keselamatan Lalu Lintas dan Penerangan Jalan Umum.
 - d. Bidang Bina Marga.
 - e. UPT dan
 - f. Kelompok Jabatan Fungsional.

2. Sekretariat

Sekretariat dipimpin oleh seorang Sekretaris, mempunyai tugas membantu dan bertanggung jawab kepada Kepala Dinas dalam hal :

- a. Pemberian pelayanan administrasi umum, perlengkapan kepegawaian, perencanaan, dan keuangan di lingkungan Dinas.
- b. Penyusunan Rencana Kerja Sekretariat.
- c. Penyediaan dan pengolahan data untuk penyusunan Rencana strategis (Renstra), Rencana Kerja (Renja), dan laporan akuntabilitas kinerja Dinas (LAKIP), laporan Penyelenggaraan Pemerintahanan Daerah, dan laporan kinerja pertanggungjawaban.

- d. Pengkoordinasian penyusunan rencana kerja dan penyelenggaraan tugas bidang secara terpadu.
- e. Penyusunan anggaran dan penatausahaan keuangan Dinas.
- f. Pengumpulan dan penyusunan peraturan di bidang perhubungan.
- g. Pengadaan, pemeliharaan, dan pengelolaan perlengkapan barang inventaris Dinas.
- h. Pembinaan dan pengendalian di bidang administrasi umum, perlengkapan, kepegawaian, perencanaan, dan keuangan.
- i. Penyelenggaraan dan pengelolaan system informasi keuangan.
- j. Pengelolaan kebersihan, keamanan, dan ketertiban Dinas.
- k. Pemantauan, evaluasi dan pelaporan tugas Sekretariat.
- l. Pengkoordinasian dan konsultasi dengan dinas/instansi/lembaga terkait untuk kelancaran pelaksanaan tugas dan.
- m. Pelaksanaan tugas lain yang diberikan atasan sesuai dengan tugas dan fungsinya.

Dalam menyelenggarakan tugas sebagaimana tersebut diatas, Sekretariat dibantu oleh :

- a. Sub-bagian Umum dan Kepegawaian.
- b. Sub-bagian Perencanaan dan Keuangan.

3. Bidang Lalu Lintas dan Angkutan

Bidang Lalu Lintas dan Angkutan dipimpin oleh seorang Kepala Bidang, mempunyai tugas membantu dan bertanggungjawab kepada Kepala Dinas dalam hal :

- a. Penyusunan bahan perumusan dan penyusunan kebijakan bidang keselamatan lalu lintas dan perlengkapan jalan.
- b. Penyusunan rencana kerja bidang.
- c. Perencanaan dan melaksanakan dalam penanganan angkutan orang dan barang, manajemen dan rekayasa lalu lintas, serta pengawasan dan pengendalian oprasional lalu lintas.
- d. Perumusan bahan kebijakan bidang lalu lintas dan angkutan.

- e. Penyiapan bahan dan pemberian perijinan rekomendasi, sertifikasi lalu lintas dan angkutan.
- f. Pengkoordinasian dan kerja sama dengan instansi lainnya dalam rangka kelancaran pelaksanaan tugas dan
- g. Pemonitoring, evaluasi dan pelaporan pelaksanaan tugas dan
- h. Pelaksanaan tugas lain yang diberikan oleh atasan sesuai dengan tugas dan fungsinya.

Dalam menyelenggarakan tugas sebagaimana dimaksud di atas, Bidang Lalu Lintas dan Angkutan dibantu oleh :

- a. Seksi Angkutan.
 - b. Seksi Manajemen dan Rekayasa Lalu Lintas.
 - c. Seksi Pengendalian Operasional Lalu Lintas.
4. Bidang Keselamatan Lalu Lintas dan Perlengkapan Jalan
- Bidang Keselamatan Lalu Lintas dan Perlengkapan Jalan dipimpin oleh seorang Kepala Bidang, mempunyai tugas membantu dan bertanggung jawab kepada Kepala Dinas dalam hal :
- a. Penyiapan bahan perumusan dan penyusunan kebijakan bidang keselamatan lalu lintas dan perlengkapan jalan.
 - b. Penyusunan rencana kerja bidang.
 - c. Perencanaan, pengadaan, dan pemeliharaan sarana lalu lintas dan penerangan jalan umum.
 - d. Penyusunan petunjuk teknis dan sosialisasi keselamatan lalu lintas.
 - e. Pelaksanaan audit keselamatan dan investigasi kecelakaan lalu lintas.
 - f. Penyiapan pemberi rekomendasi penyelenggaraan sekolah mengemudi dan bengkel umum kendaraan.
 - g. Penyiapan pemberian akreditasi, sertifikasi pengujian kendaraan bermotor dan pengesahan hasil uji berkala kendaraan bermotor yang dilaksanakan oleh swasta.
 - h. Penentuan lokasi dan evaluasi fasilitas parkir umum.
 - i. Pengkoordinasian dengan instansi lainnya dalam rangka kelancaran pelaksanaan tugas;

- j. Pemonitoran, evaluasi dan pelaporan pelaksanaan tugas dan
- k. Pelaksanaan tugas lain yang diberikan oleh atasan sesuai dengan tugas dan fungsinya.

Dalam menyelenggarakan tugas sebagaimana dimaksud di atas, Bidang Keselamatan Lalu Lintas dan Perlengkapan Jalan dibantu oleh :

- a. Seksi Perlengkapan Jalan.
- b. Seksi Keselamatan Lalu Lintas.
- c. Seksi Penerangan Jalan Umum.

5. Bidang Bina Marga

Bidang Bina Marga dipimpin oleh seorang Kepala Bidang, mempunyai tugas membantu dan bertanggungjawab kepada Kepala Dinas dalam hal :

- a. Pelaksanaan teknis operasional di bidang bina marga yang meliputi pengendalian, operasional, pembangunan/ peningkatan kualitas jalan dan jembatan serta pemeliharannya.
- b. Penetapan kondisi jalan kota.
- c. Perencanaan peningkatan jalan dan jembatan.
- d. Pemeliharaan Jalan.
- e. Pemberi izin pemanfaatan/penggunaan daerah manfaat jalan.
- f. Melakukan penyusunan perencanaan pembangunan transportasi kota yang seluruh wilayahnya berada dalam wilayah Daerah.
- g. Pengkoordinasian dan konsultasi dengan dinas/instansi/lembaga lainnya dalam rangka kelancaran pelaksanaan tugas.
- h. Evaluasi dan laporan hasil pelaksanaan tugas dan
- i. Pelaksanaan tugas lain yang diberikan oleh atasan sesuai dengan tugas dan fungsinya.

Dalam melaksanakan tugas sebagaimana dimaksud diatas, Kepala Bidang Prasarana Jalan dan Jembatan dibantu oleh :

- a. Seksi Perencanaan Jalan dan Jembatan.
- b. Seksi Pembangunan Jalan dan Jembatan.
- c. Seksi Pemeliharaan Jalan dan Jembatan.

6. UPT (Unit Pelaksana Teknis) Terdiri dari 3 UPT, yaitu :

- a. UPT Terminal.
- b. UPT Parkir dan
- c. UPT Pengujian Kendaraan Bermotor (PKB).

2.2 Sistem

Ada dua pengelompokan pendekatan tentang definisi sistem (Fanny dan Eko), ada pengelompokan pendekatan pada prosedurnya dan pengelompokan pendekatan pada komponen atau elemennya [17].

- a. Pengelompokan pendekatan sistem pada prosedurnya “Sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan untuk melakukan suatu kegiatan dalam menyelesaikan suatu sasaran tertentu”.
- b. Pengelompokan pendekatan sistem pada komponennya “Sistem merupakan kumpulan suatu elemen yang berhubungan untuk mencapai suatu tujuan tertentu”.

2.3 Parkir

Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggal oleh pengemudinya. Menurut Hobbs, parkir diartikan sebagai suatu kegiatan untuk meletakkan atau menyimpan kendaraan di suatu tempat tertentu yang lamanya tergantung kepada selesainya keperluan dari pengendaraan tersebut [18].

Menurut undang-undang nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, Bab I Ketentuan Umum, pada Pasal 1 angka 15 dan 16 tertulis sebagai berikut : Pasal 1 Dalam Undang-Undang ini yang dimaksud dengan : 15. Parkir adalah keadaan Kendaraan berhenti atau tidak bergerak untuk beberapa saat dan ditinggalkan pengemudinya. 16. Berhenti adalah keadaan Kendaraan tidak bergerak untuk sementara dan tidak ditinggalkan pengemudinya.

Ada 2 jenis parkir yang dikemukakan oleh Martadipura (2013), yaitu Parkir yang berada di badan jalan (on street parking) dan parkir yang berada luar badan jalan (off street parking). Pada On Street parking membutuhkan badan jalan untuk digunakan sebagai tempat parkir, contoh adalah parkir di depan pertokoan dimana kendaraan berhenti di pinggir jalan. Sedangkan untuk off street parking tidak

membutuhkan badan jalan untuk sebagai tempat parkir melainkan dipersiapkan sebuah lahan khusus untuk menampung kendaraan parkir [19].

2.3.1 Sistem Parkir

Sistem parkir di Indonesia dapat dibagi menjadi dua jenis yaitu sistem parkir di badan jalan (*on-street parking*) dan sistem parkir di dalam pelataran parkir (*off-street parking*). Pada sistem di dalam pelataran parkir dapat dibagi lagi menjadi dua jenis yaitu gedung parkir (*parking building*) dan parkir di bawah tanah (*basement parking*), berdasarkan pengaturan posisi kendaraan, yaitu paralel, parkir tegak lurus, dan parkir serong [20]. Pada umumnya komponen-komponen yang terdapat pada sebuah sistem parkir seperti di atas yaitu petugas parkir, PC (*personal computer*) yang ditempatkan pada pos masuk dan pos keluar. Petugas parkir pada pos masuk bertugas untuk memasukkan data berupa nomor plat mobil ke dalam database pada PC dan memberikan tiket pada *costumer*. Sedangkan petugas parkir pada pos keluar bertugas untuk mengambil tiket dan uang parkir dari *costumer*.

Sistem parkir terkomputerisasi, yang menggunakan bantuan komputer untuk mengotomatisasi proses parkir. Kedua jenis sistem parkir ini memiliki kelebihan dan kelemahan. Untuk sistem parkir konvensional, kelebihannya adalah penggunaannya yang sudah dikenal, dianggap mudah, dan tidak membutuhkan investasi teknologi. Pada sisi yang lain, sistem ini mempunyai kelemahan yaitu seringnya terjadi kesalahan dalam penulisan karcis oleh petugas parkir, sehingga berdampak waktu pelayanan lama akibat proses duplikasi penulisan ulang karcis.

Kelemahan lainnya yaitu pemborosan kertas parkir dan data transaksi tidak terekam. Sistem parkir terkomputerisasi juga memiliki kelebihan. Kelebihannya antara lain permasalahan pengelolaan dengan sistem parkir konvensional dapat tereliminasi [21].

2.3.2 Kendaraan

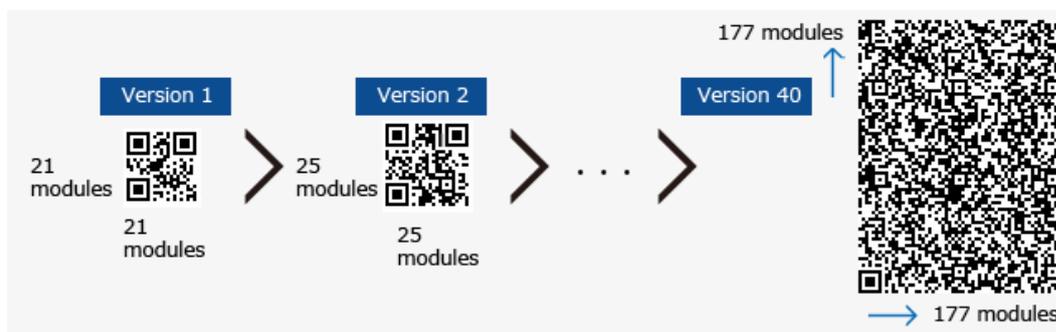
Kendaraan adalah alat transportasi, baik yang digerakkan oleh mesin maupun oleh makhluk hidup. Kendaraan ini biasanya buatan manusia (mobil, motor, kereta, perahu, dan pesawat). Kendaraan tidak bermotor dapat juga digerakkan oleh manusia atau ditarik oleh hewan, seperti gerobak [22].

2.4 Teknologi Matriks *Barcode* 2D

Sebuah kode batang atau suatu kumpulan data optik yang dibaca mesin. Sebenarnya, kode batang ini mengumpulkan data dalam lebar (garis) dan spasi garis paralel dan dapat disebut sebagai kode batang atau simbologi linear atau 1D (1 dimensi). Tetapi juga memiliki bentuk persegi, titik, heksagon dan bentuk geometri lainnya di dalam gambar yang disebut kode matriks atau simbologi 2D (2 dimensi). Selain tak ada garis, sistem 2D sering juga disebut sebagai kode batang.

2.4.1 *QR Code (Quick Response Code)*

QR Code adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994. Tidak seperti *barcode* yang hanya satu sisinya saja yang mengandung data, *QR Code* mempunyai dua sisi yang berisi data, dan ini membuat *QR Code* lebih banyak memuat informasi dibandingkan *barcode* (Denso Wave Inc, 2011) [23]. Ada 40 versi dalam *QR Code*, 4 level koreksi kesalahan, dan ukuran simbol maksimum (versi tertinggi) dapat mengkodekan 7089 data numerik atau 4296 data alfanumerik. Tingkat koreksi kesalahan tertinggi memungkinkan pemulihan 30% dari kata-kata kode simbol [24].



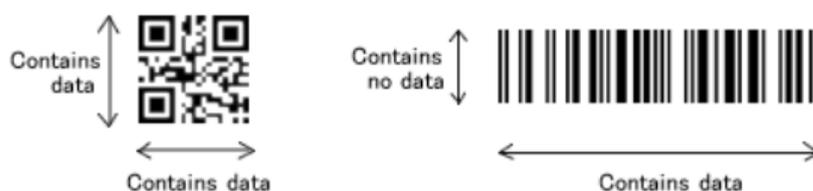
Gambar 2.3 Beberapa versi *QR Code*

QR Code juga memiliki fitur koreksi kesalahan, yang secara otomatis akan mengembalikan data jika beberapa dari mereka rusak atau tidak dapat dibaca. Skema koreksi kesalahan ini menggunakan algoritma Koreksi Kesalahan Reed-Solomon yang terdiri dari 4 level koreksi : Low (L), Medium (M), Quartile (Q), dan High (H), di mana masing-masing level memiliki kemampuan sendiri bagaimana banyak data dapat dipulihkan.

Tabel 2.1 Level koreksi kesalahan

Level Koreksi Kesalahan	Kemampuan Level koreksi Kesalahan
L	Memulihkan 7% data
M	Memulihkan 15% data
Q	Memulihkan 25% data
H	Memulihkan 30% data

QR Code terdiri dari modul hitam (titik persegi) diatur dalam kotak persegi pada latar belakang putih yang dapat dibaca oleh perangkat pencitraan seperti kamera dan diolah menggunakan koreksi kesalahan Reed-Solomon hingga gambar dapat dengan tepat di interpretasikan. Data tersebut kemudian diekstraksi dari pola yang ada dari kedua komponen horizontal dan komponen vertical pada gambar [25].

Gambar 2.4 *QR Code*Gambar 2.5 Perbedaan *QR Code* dan *Barcode*

Setiap modul gelap dan terang pada *QR code* mewakili 0 dan 1 sehingga memungkinkan *QR code* di baca oleh mesin. Modul *QR code* memiliki beberapa fungsi, sebagian modul berisi data sedangkan bagian lainnya merupakan modul yang berfungsi sebagai function pattern yang meningkatkan performa pembacaan, error correction dan kompensasi distorsi. *Timing pattern* menyimpan informasi

ukuran dari simbol *QR code* yang nantinya di gunakan oleh pembaca *QR code*. Selain itu *QR code* juga membutuhkan “*Quiet Zone*” yaitu buffer seluas 4 modul yang tidak berisi data untuk memastikan teks ataupun tanda di sekitar *QR code* tidak di salah kenali sebagai data *QR code* [26]. Selain *QR Code* beberapa jenis kode 2D lainnya telah dikembangkan. Tabel berikut menunjukkan berbagai jenis kode 2D beserta kapasitas datanya.

Tabel 2.2 Jenis kode 2D dan kapasitasnya

		QR Code	PDF417	Datamatrix	Maxi Code
					
Pengembang (Negara)		DENSO (Jepang)	Symbol Technologies (AS)	RVSI Acuity CiMatrix (AS)	UPS (AS)
Tipe		Matriks	Stacked Barcode	Matriks	Matriks
Kapasitas Data	Numerik	7089	2710	3116	138
	Alfanumerik	4296	1850	2355	93
	Biner	2953	1018	1556	
	Kanji	1817	554	778	

2.4.2 Fungsi dan cara kerja *QR Code*

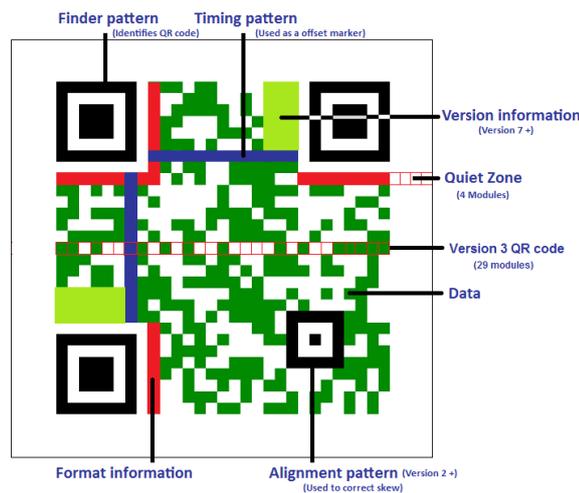
QR Code berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan URL, nomer telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, pada tanda tanda bus, kartu nama ataupun media lainnya. *QR Code* adalah sebuah kode yang berisi informasi yang dapat dibaca dengan menggunakan cara sebagai berikut [5]:



Gambar 2.6 Cara kerja *QR Code* pada *smarthphone*

- Pembaca *QR Code* (*QR Code Reader*) melalui kamera ponsel akan menyimpan informasi pada *QR Code*, informasi yang dimaksud berupa alamat web (URL), nomor telepon, teks dan SMS.
- Setelah di *scan*, sebuah *link* URL yang berisi informasi pada *QR Code* akan ditampilkan pada layar ponsel.

QR Code memiliki bagian-bagian struktur yang akan penulis jelaskan pada gambar 2.7 dibawah ini [27] :



Gambar 2.7 Struktur *QR Code*

- Finding Pattern* merupakan pola untuk mendeteksi posisi dari *QR Code*.
- Timing pattern* merupakan pola yang digunakan untuk identifikasi koordinat pusat dari *QR Code*, dibuat dalam bentuk modul hitam putih bergantian.

3. *Version Information* merupakan Versi dari sebuah *QR Code*, versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (177 x 177) modul.
4. *Quiet Zone* merupakan organisasi kosong dibagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor CCD.
5. *QR Code version* merupakan versi *QR Code*. Pada contoh gambar, versi yang digunakan adalah versi 3 (29 x 29 modul).
6. *Data* merupakan organisasi tempat data tersimpan atau data dikodekan.
7. *Alignment Pattern* merupakan pola yang digunakan untuk memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
8. *Format information* merupakan informasi tentang error correction level dan mask pattern.

2.4.2.1 Kapasitas *QR Code*

Jumlah data yang dapat disimpan pada simbol *QR Code* tergantung dari tipe data (mode, atau kumpulan karakter masukan), versi (1, ..., 40, mengindikasikan keseluruhan dimensi simbol), dan tingkat kode koreksi kesalahan. Kapasitas penyimpanan maksimum saat ini adalah untuk simbol 40-L (versi 40 dengan koreksi kesalahan tingkat *low*) [25].

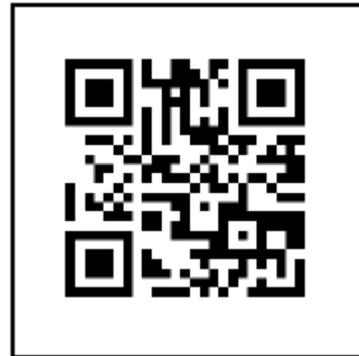
Tabel 2.3 Kapasitas Maksimum Penyimpanan Karakter (40-L)

Input mode	Maksimal karakter	Bits / karakter	Default encoding
Numeric only	7.089	$3^{1/3}$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric	4.296	$5^{1/2}$	0–9, A–Z (upper-case only), space, \$, %, *, +, -, ., /, :
Binary/byte	2.953	8	ISO 8859-1
Kanji/kana	1.817	13	Shift JIS X 0208

Setiap versi akan mempunyai ukuran yang berbeda dengan versi lainnya. Gambar 2.8 sampai Gambar 2.13 adalah contoh simbol *QR Code* dengan versi yang berbeda.



Gambar 2.8 *QR Code* Versi 1
(21x21)



Gambar 2.9 *QR Code* Versi 2
(25x25)



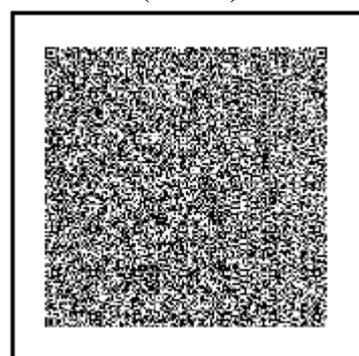
Gambar 2.10 *QR Code* Versi 3
(29x29)



Gambar 2.11 *QR Code* Versi 4
(33x33)



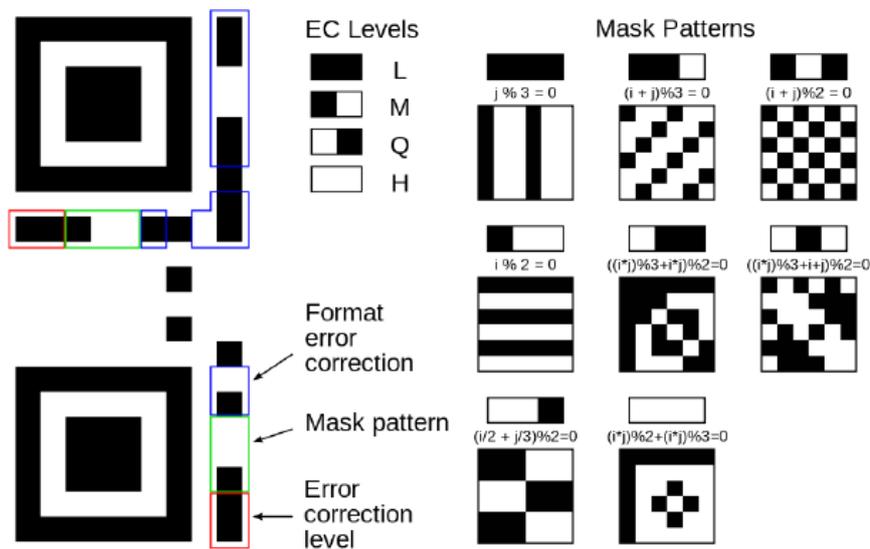
Gambar 2. 1 *QR Code* Versi 25
(117x117)



Gambar 2. 2 *QR Code* Versi 40
(177x177)

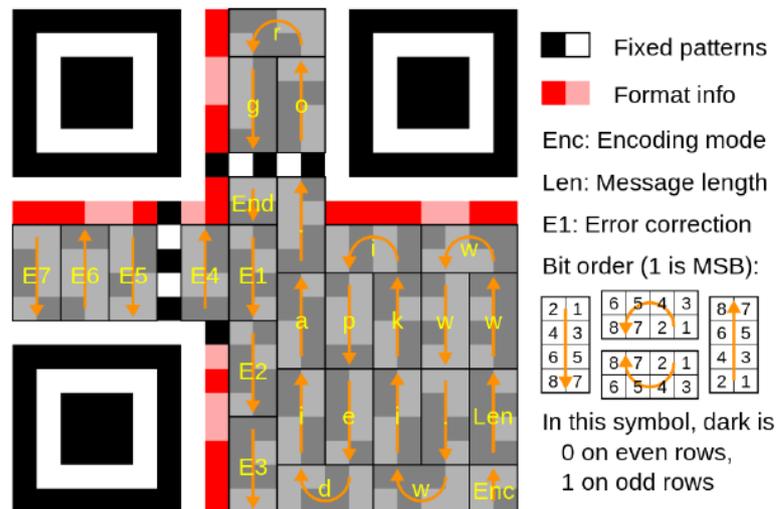
2.4.2.2 Encoding

Informasi format mencatat dua hal: tingkat koreksi kesalahan dan *mask pattern* yang digunakan untuk simbol. *Masking* digunakan untuk memecah pola di area data yang mungkin membingungkan *scanner*, seperti daerah kosong besar atau fitur menyedatkan yang terlihat seperti tanda *locator*. *Mask pattern* didefinisikan pada grid yang diulang sediperlukannya untuk menutupi seluruh simbol. Modul yang sesuai dengan daerah gelap mask terbalik. Informasi format dilindungi dari kesalahan dengan kode BCH, dan dua salinan lengkap termasuk dalam setiap simbol *QR*.



Gambar 2.12 Maksud dari informasi format

Data pesan ditempatkan dari kanan ke kiri dalam pola zigzag. Dalam simbol yang lebih besar, cukup rumit karena adanya pola keselarasan dan penggunaan beberapa blok *interleaved* koreksi kesalahan.



Gambar 2.13 Penempatan Pesan dan Simbol QR Code

Indikator 4 bit digunakan untuk memilih mode *encoding* dan menyampaikan informasi lainnya. Mode pengkodean dapat dicampur sesuai kebutuhan dalam simbol QR.

Tabel 2.4 Menu *encoding*

Indikator	Keterangan
0001	<i>Numeric encoding (10 bits per 3 digits)</i>
0010	<i>Alphanumeric encoding (11 bits per 2 characters)</i>
0100	<i>Byte encoding (8 bits per character)</i>
1000	<i>Kanji encoding (13 bits per character)</i>
0011	<i>Structured append (used to split a message across multiple QR symbols)</i>
0111	<i>Extended Channel Interpretation (select alternate character set or encoding)</i>
0101	<i>FNC1 in first position (see Code 128 for more information)</i>
1001	<i>FNC1 in second position</i>
0000	<i>End of message</i>

Setelah setiap indikator yang memilih mode *encoding*, panjang *field* lah yang memberitahu berapa banyak karakter dikodekan dalam mode tersebut. Jumlah bit pada panjang *field* tergantung pada *encoding* dan versi simbol.

Tabel 2.5 Jumlah bit per panjang field

Encoding	Ver.1–9	10–26	27–40
Numerik	10	12	14
Alfanumerik	9	11	13
Byte	8	16	16
Kanji	8	10	12

Mode *encoding* alfanumerik menyimpan pesan lebih kompak daripada yang mode byte, tetapi mode alfanumerik tidak bisa menyimpan huruf-huruf non-kapital dan memiliki pilihan tanda baca terbatas yang cukup untuk alamat web yang belum sempurna. Dua karakter dikodekan dalam nilai 11-bit dengan rumus ($V = 45 \times C_1 + C_2$)

Tabel 2.6 Kode karakter alfanumerik

Kode	Karakter	Kode	Karakter
00	0	23	N
01	1	24	O
02	2	25	P
03	3	26	Q
04	4	27	R
05	5	28	S
06	6	29	T
07	7	30	U
08	8	31	V
09	9	32	W
10	A	33	X
11	B	34	Y
12	C	35	Z
13	D	36	SP
14	E	37	\$
15	F	38	%
16	G	39	*
17	H	40	+
18	I	41	–

19	J	42	.
20	K	43	/
21	L	44	:
22	M	45	

2.4.2.3 Decoding

Decoding adalah proses pembacaan *QR Code* untuk menghasilkan informasi dari data yang ada pada *QR Code*. *Decoding* merupakan kebalikan dari proses *encoding* yang merupakan proses untuk mengubah data ke dalam bentuk *QR Code* [25]. Tahapan *Decoding QR Code* dari pembacaan sampai menjadi *output* data merupakan kebalikan dari prosedur *encoding*, tahapan – tahapan *decoding* adalah sebagai berikut :

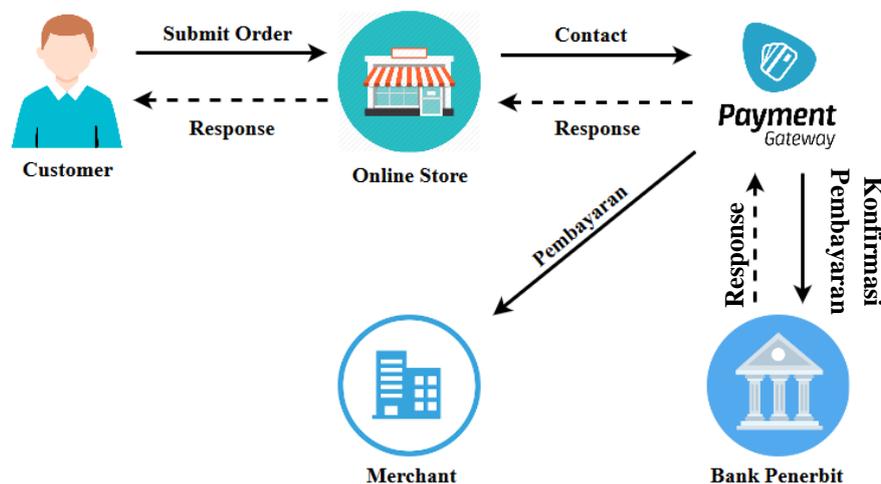
1. Cari dan temukan citra simbol, kenali modul gelap dan terang sebagai *array* “0” dan “1”.
2. Baca informasi format.
3. Baca *version information* dan tentukan *version* dari *QR code*.
4. Lepaskan *masking* dengan cara menaplikasi operasi XOR pada daerah yang di *masking* dengan *pattern masking* yang di dapatkan dari ekstraksi informasi format.
5. Baca simbol karakter sesuai dengan aturan penempatan dan kembalikan data dan *error correction codeword* dari pesan.
6. Deteksi *error* menggunakan *error correction codeword*, jika di temukan *error* , perbaiki.
7. Pisahkan data *codeword* menjadi *segment* sesuai dengan *mode indicator* dan *character count indicator*.
8. *Decode* karakter sesuai dengan mode yang digunakan dan keluarkan hasil.

2.5 Payment Gateway

Menurut (Gulati & Srivastava, 2007) *Internet E-commerce Payment gateway* adalah komponen infrastruktur penting untuk memastikan transaksi berlangsung tanpa hambatan dan terlindungi total melalui jaringan *internet*. *Payment Gateway* adalah sebuah akses poin ke dalam jaringan perbankan nasional. Semua transaksi

secara *online* harus melalui *Payment Gateway* untuk diproses. Secara teorinya, *payment gateway* bertindak sebagai jembatan antara pemilik *website* dan institusi keuangan yang melakukan proses transaksi. *Payment gateway* membuktikan dan mengarahkan detail pembayaran didalam lingkungan teraman antara berbagai pihak dan bank yang terkait. Fungsi *payment gateway* pada dasarnya sebagai saluran yang terenkripsi, yang secara aman mengirimkan detail transaksi dari pembeli yang menggunakan *personal computer* ke bank untuk disetujui [9].

Payment Gateway merupakan *merchant service* yang disediakan oleh penyedia layanan aplikasi e-commerce yang mengesahkan kartu kredit atau pemrosesan pembayaran langsung untuk e-bisnis, *online retailers*, bisnis *online* dan *offline (brick and click)*. *Payment Gateway* mungkin disediakan oleh bank kepada pelanggannya, tetapi dapat disediakan oleh penyedia layanan keuangan khusus sebagai layanan terpisah, seperti penyedia layanan pembayaran [28]. Berikut cara kerja dari *Payment Gateway* :



Gambar 2.14 Cara kerja *payment gateway*

2.6 Android

Android adalah sebuah *system operasi* untuk perangkat *mobile* berbasis linux yang mencakup *system operasi*, *middleware* dan aplikasi [29]. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34

perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*.

2.6.1 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

a. *Applications* dan *Widgets*

Applications dan *Widgets* ini adalah layer di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. *Applications Frameworks*

Applications Frameworks ini adalah layer di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-provider* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications* dan *Frameworks* adalah sebagai berikut :

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

c. *Libraries*

Libraries ini adalah layer di mana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, layer ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta :

1. *libraries* media untuk pemutaran media audio dan video.
2. *libraries* untuk manajemen tampilan.
3. *libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.

4. *libraries* SQLite untuk dukungan *database*.
5. *libraries* SSL dan WebKit terintegrasi dengan *web browser* dan *security*.
6. *libraries* LiveWebcore mencakup modern web browser dengan *engine embeded web view*.
7. *libraries* 3D yang mencakup implementasi OpenGL ES 1.0 *API's*.

d. *Android Run Time*

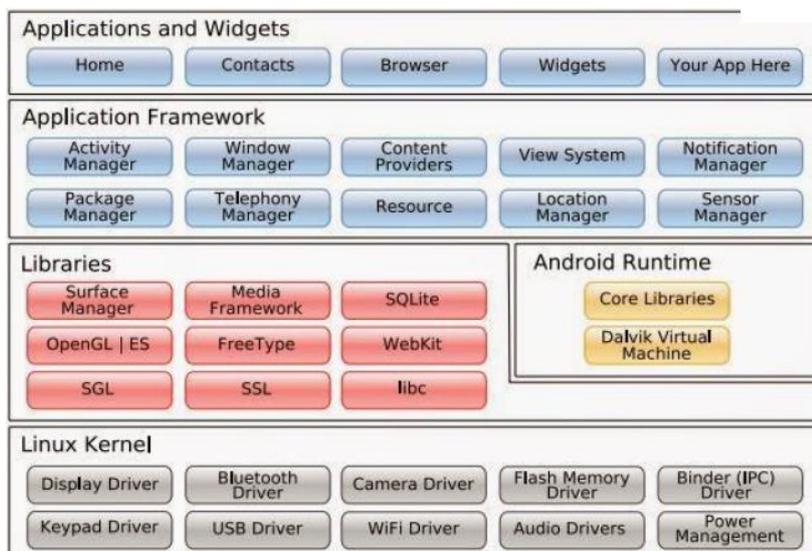
Android Run Time merupakan layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu :

1. *Core Libraries* : berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries*. e. Linux Kernel.
2. *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.

e. *Linux Kernel*

Linux kernel adalah layer dimana inti dari *operating system* dari Android itu berada. Berisi *file-file system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan Android adalah linux kernel realease 2.6.

Berikut ini adalah gambar dari Arsitektur Android yang dapat dilihat pada gambar dibawah ini.



Gambar 2.15 Gambar arsitektur android

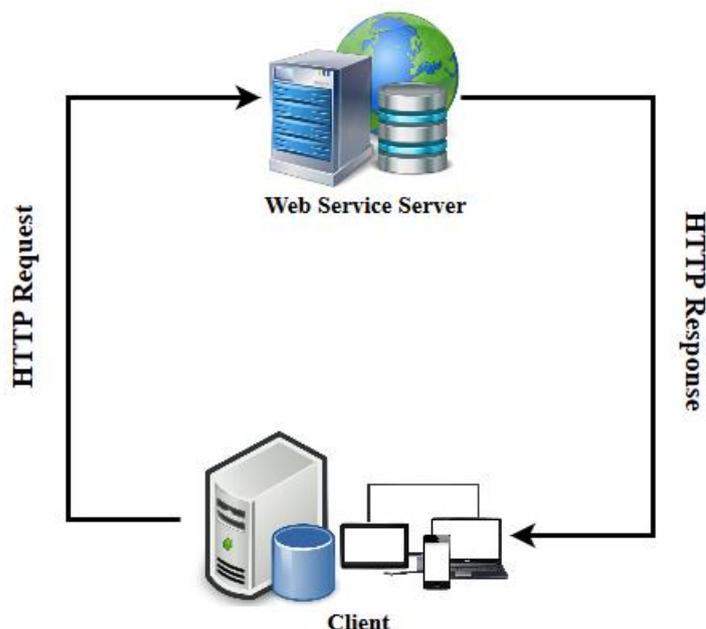
2.7 Android Studio

Android Studio adalah *Integrated Development Environment (IDE)* resmi Google untuk mengembangkan aplikasi Android. Android Studio diperkenalkan oleh Google pada tahun 2013 pada konferensi pengembang tahunan mereka Google I / O. Setelah lebih dari satu tahun setengah pengembangan aktif, Android Studio akhirnya mencapai versi stabil dan resmi merilis versi 1.0 pada Desember 2014.

2.8 Web Services

Menurut Salter-Jennings, *web service* merupakan sebuah teknologi yang dapat digunakan untuk mengimplementasikan *service*. Menurut Hwang et al (2008), *web service* secara fakta telah menjadi standar untuk melakukan ekspos fungsi dari aplikasi bisnis. *Web service* akan menjadi blok bangunan untuk pengembangan aplikasi generasi mendatang [30].

Menurut O'Brien (2010, p157), Web service merupakan komponen software yang berbasis framework web dan standar object-oriented dan teknologi untuk penggunaan web yang secara elektronik menghubungkan aplikasi user yang berbeda dan platform yang berbeda. Web service dapat menghubungkan fungsi bisnis untuk pertukaran data secara real time dalam aplikasi berbasis web.



Gambar 2.16 Arsitektur *web services*

Untuk lebih jelasnya mengenai web server, berikut ini adalah contoh dari cara kerja web server secara konkret [31]:

1. User mengawali dengan melakukan koneksi terhadap internet.
2. Setelah komputer terhubung ke dalam internet, maka user akan memasukkan alamat internet URL dengan base www di dalam browsernya.
3. Ketika user menekan enter, maka secara teknis user sudah melakukan request atau permintaan dari konten tertentu yang akan dia akses.
4. Ketika user sudah melakukan request tersebut, maka request tersebut akan diteruskan ke dalam web server, dan kemudian akan diproses oleh web server.
5. Pada saat *web server* menerima *request*, maka *web server* akan mengolah request tersebut, dan mencari semua informasi yang dibutuhkan oleh user pada saat proses request berlangsung.
6. Setelah pemrosesan yang dilakukan oleh web server selesai, maka web server akan mengirimkan kembali hasil dari pemrosesan data yang sudah dilakukan.
7. Hasil dari pemrosesan data yang sudah dilakukan oleh web server kemudian akan muncul pada layar dan juga tampilan browser yang dimiliki oleh user.

Pada dasarnya web server adalah salah satu elemen vital pada sebuah website, dan sudah pasti memiliki banyak sekali fungsi utama. Berikut ini adalah beberapa fungsi utama dari sebuah web server [31]:

1. Untuk memproses permintaan user di sebuah web
2. Sebagai penyedia data dan juga informasi yang dimiliki oleh sebuah web
3. Menyimpan konten sebuah website
4. Penentu kecepatan pemrosesan informasi dari sebuah halaman web

2.9 Sublime Text 3

Sublime Text merupakan *text editor* berbagai bahasa pemrograman mulai dari bahasa c hingga *java*. Sublime Text merupakan *editor* HTML yang profesional untuk mendesain, menulis kode program, dan mengembangkan ke halaman *website*, halaman *web*, dan aplikasi *web*. Sublime Text 3 memiliki beberapa kelebihan yaitu :

1. *Multi Platform*

Kelebihan dari Sublime Text adalah *software* ini tersedia dalam berbagai *platform* sistem operasi, antara lain Windows, Linux, dan MacOS.

2. *Plugin*

Plugin-nya sangat beragam, sehingga bisa memudahkan programmer dalam mengembangkan *software*-nya. Sublime Text juga memiliki sangat banyak *package*. Untuk menginstall *package* kita bisa menjalankan *Package Manager* kemudian akan muncul *list package*, kita tinggal mencari (tentu saja menggunakan *fuzzy search* juga) *package* yang diinginkan.

3. Tema dan *color scheme* yang bervariasi

Di Sublime Text, kita bisa mengganti *Color Scheme* (untuk area editor) maupun *Theme* (untuk *interface sidebar, tab, console, search* dll). Untuk mengedit *color scheme*, kita harus membuka *file xml* berekstensi *.tmTheme* dan mengedit isinya, atau bisa juga menggunakan *.tmTheme* editor.

4. *Go To Anything*

Dengan *Go to Anything*, kita bisa membuka file di dalam *project* dengan cepat, tinggal tekan *Ctrl + P* kemudian ketik nama filenya. Untuk mencari nama *file* tidak harus mengetik secara tepat, karena adanya algoritma *fuzzy*.

5. *Drag & Drop*

Menyeret dan melepas *file* teks ke dalam editor akan membuka tab baru secara otomatis. Kita juga bisa menentukan lokasi tab pada saat menyeret *file* teks tersebut.

6. Membuka *File* Besar

Sublime Text mampu membuka dan mengedit sebuah *file* teks yang sangat besar tanpa masalah.

7. *Command Palette*

Sublime Text memiliki tampilan yang lebih *simple* dan sangat minim *menu*, bahkan tidak ada *toolbar* sama sekali. Kebanyakan perintah-perintahnya bisa kita akses menggunakan *Shortcut*, atau kalau belum hafal *shortcut* kita bisa mengakses *Command Palette* (tekan *Ctrl + Shift + P*) kemudian cari perintah yang kita inginkan (menggunakan *fuzzy search*).

8. *Column Editing*

Baik Notepad++ maupun Sublime Text memiliki fitur ini. Di Notepad++ tekan *Alt* kemudian *mouse* didrag ke bagian yang ingin diedit. Di Sublime Text menggunakan *drag* tombol *mouse* tengah, atau bisa juga dengan *Shift + drag* tombol *mouse* kanan.

9. *Split Editing*

Jika and memiliki monitor yang lebar maka kedua editor ini sangat cocok karena sama-sama memiliki fitur untuk *Split Editing* (membuka 2 atau lebih file).

10. *Auto- Completion*

Sublime Text memiliki auto complete untuk beberapa bahasa yang saya pakai seperti PHP, CSS, Javascript. Fitur ini juga mendukung *fuzzy search* sehingga tidak harus menyetik secara tepat. Notepad++ juga memiliki *auto complete*, namun tidak senyaman Sublime Text dan harus diaktifkan dulu melalui *Settings -> Preferences -> Backup/Auto-Completion*.

11. *Mini map/document map*

Sublime Text memiliki *Minimap*, semacam versi *mini* dari *file* untuk mempermudah melihat *file* secara keseluruhan.

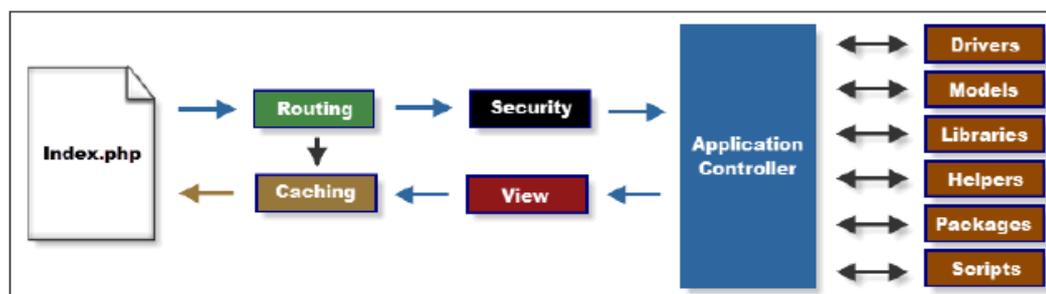
12. *Go To Definition*

Ini adalah fitur baru Sublime Text 3 , fitur ini sangat membantu menemukan *function/class* di dalam *project*, caranya dengan meletakkan *cursor* di nama *function/class* kemudian tekan F12, maka *file* yang berisi definisi *function/class* tersebut akan terbuka. Untuk melihat list semua *function/class* bisa menggunakan *Go to Symbol in Project* (tekan *Ctrl + Shift + R*) [32].

2.10 Codeigniter

Codeigniter merupakan salah satu macam kerangka kerja pemrograman web berbasis bahasa pemrograman PHP yang menggunakan pendekatan MVC (*Model-View-Controller*). Codeigniter memiliki banyak fitur dengan kebutuhan sistem yang ringan dan dapat mempermudah dalam pengembangan sebuah aplikasi web yang kaya akan content (*rich application*). Dengan menggunakan perangkat kerangka kerja Codeigniter *framework*, dapat mengurangi jumlah baris kode

program, sehingga dapat meminimalisasi kesalahan penulisan kode program, mengurangi besarnya ukuran file, dan mempercepat eksekusi program (Upton, 2007). Codeigniter *framework* memiliki banyak *library* dengan kelebihan lainnya, terutama untuk menangani keamanan data melalui proses POST *data filtering* saat eksekusi dan kemampuan untuk melakukan kompresi dokumen (EllisLab Inc., 2011)[33].



Gambar 2.17 Arsitektur *codeigniter framework*

Cara kerja *framework* Codeigniter dimulai dari awal saat *client* melakukan *request* ke *server*, kemudian *framework* akan melakukan cek terlebih dahulu pada *routing database* untuk mengetahui *controller* apa yang akan dieksekusi. Jika pernah dieksekusi dan disimpan dalam *cache*, *framework* akan melakukan *redirect* ke *cache* tersebut, sehingga dapat mengurangi waktu proses. Data input akan melalui *security layer* dimana akan dilakukan *filtering* dan sanitasi data, sehingga input data yang masuk ke dalam sistem sudah bersih dan valid, tidak mengandung kode asing yang dapat membahayakan sistem. Berikutnya, sistem menjalankan *controller* yang diminta, sekaligus menyediakan *library* yang dibutuhkan. Terakhir, *output* data ditampilkan melalui proses *View*.

2.11 PHP (*Pheriperal Hypertext Preprocessor*)

PHP merupakan singkatan dari *Pheriperal Hypertext Preprocessor*, sebuah bahasa *scripting* yang terpasang pada HTML [34]. Sebagian besar sintaks mirip dengan bahasa C, Java, asp dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang *web* untuk menulis halaman *web* dinamik. Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan sistem *database* di dalam *web*. Sistem *database* yang dapat didukung oleh PHP

adalah Oracle, MySQL, Ms.Access Sybase, PostgreSQL dan lainnya.

Keunggulan lainnya dari PHP adalah juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 bahkan HTTP. Berikut adalah kelebihan PHP [34]:

1. *Web Server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
2. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak.
3. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah system.

2.12 MYSQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [34]. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. MySQL biasanya digunakan atau diinstall bersamaan dengan XAMPP sehingga untuk melihat isi tabel bias menggunakan PHPmyAdmin.

Sebagai *software database* dengan konsep *database* modern, MySQL memiliki banyak kelebihan antara lain :

1. *Protability*

MySQL dapat digunakan dengan stabil tanpa kendala, berarti pada berbagai sistem operasi diantaranya seperti Windows, Linux, Mac OS X Server, Solaris, Amiga HP-UX dan masih banyak lagi. *Open source* MySQL didistribusikan

secara *open source* di bawah lisensi GPL, sehingga dapat memperoleh menggunakannya secara cuma-cuma tanpa dipungut biaya sepeserpun.

2. *Multiuser*

MySQL dapat digunakan untuk menangani beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini akan memungkinkan sebuah database server MySQL dapat diakses *client* secara bersamaan dalam waktu yang bersamaan pula.

3. *Performance Tuning*

MySQL memiliki kecepatan yang cukup menakjubkan dalam menangani query sederhana, serta mampu memproses lebih banyak SQL persatuan waktu.

4. *Column Types*

MySQL didukung tipe kolom (tipe data) yang sangat kompleks.

5. *Command dan Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

6. *Scalability dan Limits*

Dalam hal batas kemampuan, MySQL terbukti mampu menangani *database* dalam skala yang besar dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

7. *Interface*

Sama halnya dengan software database lainnya, MySQL memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

8. Struktur tabel

Struktur tabel MySQL cukup baik, serta cukup fleksibel. Misalnya ketika menangani *Alter Table*, dibandingkan database lainnya semacam ProgresSQL ataupun Oracle.

2.13 UML (*Unified Modeling Language*)

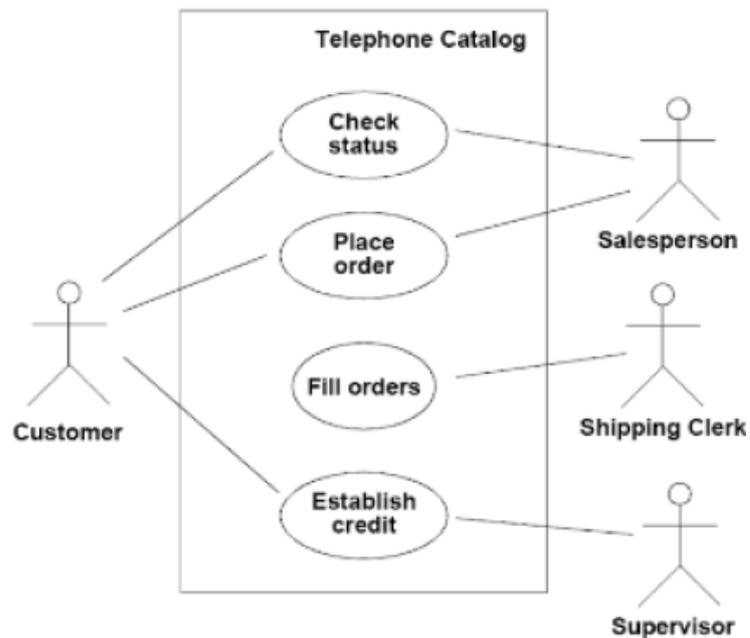
UML (*Unified Modeling Language*) merupakan sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, deskripsi, atau *software*) yang terdapat dalam sistem *software*. UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OOP yang telah ada sebelumnya, yaitu *Booch*, OMT (*Object Modeling Technique*), dan OOSE (*Object-Oriented Software Engineering*). UML merupakan kesatuan dari ketiga pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode tersebut. UML dikeluarkan oleh OMG (*Object Management Group, Inc*) yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer*, dan para *user* sistem komputer.

Dengan adanya UML, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. UML diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini[35]. Tujuan UML diantaranya adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

2.13.1 *Use Case Diagram*

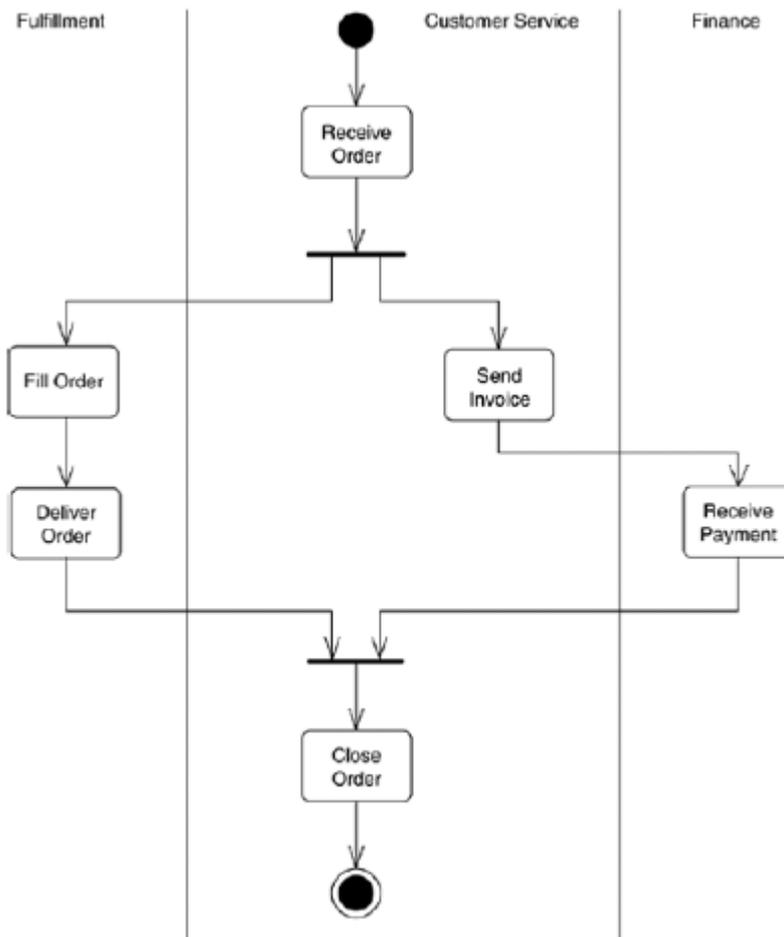
Use Case Diagram merupakan diagram yang menggambarkan dan merepresentasikan *actor*, *use cases*, dan *dependencies* suatu proyek dimana tujuan dari *diagram* ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar [36]. Berikut contoh *use case diagram* pada gambar dibawah ini.



Gambar 2.18 Contoh *use case* diagram

2.13.2 Activity Diagram

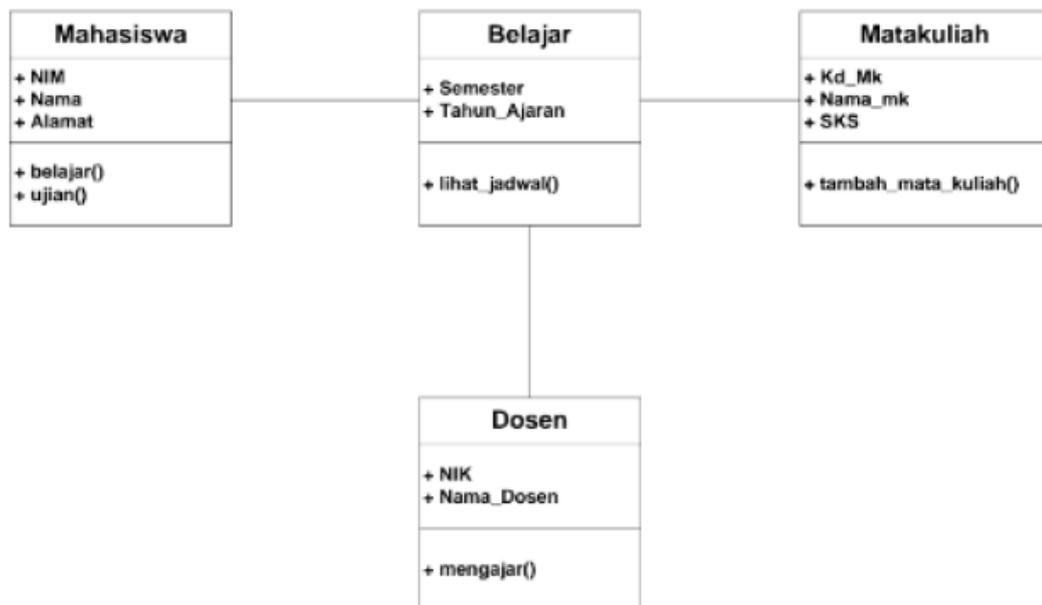
Activity Diagram adalah diagram uml yang digunakan untuk menggambarkan alur aktivitas dari satu proses. *Activity* Diagram memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya, dengan kata lain diagram hanya menyebutkan aturan – aturan rangkaian dasar yang harus diikuti [36]. Berikut contoh *activity* diagram pada gambar dibawah ini.



Gambar 2.19 Contoh *activity* diagram

2.13.3 Class Diagram

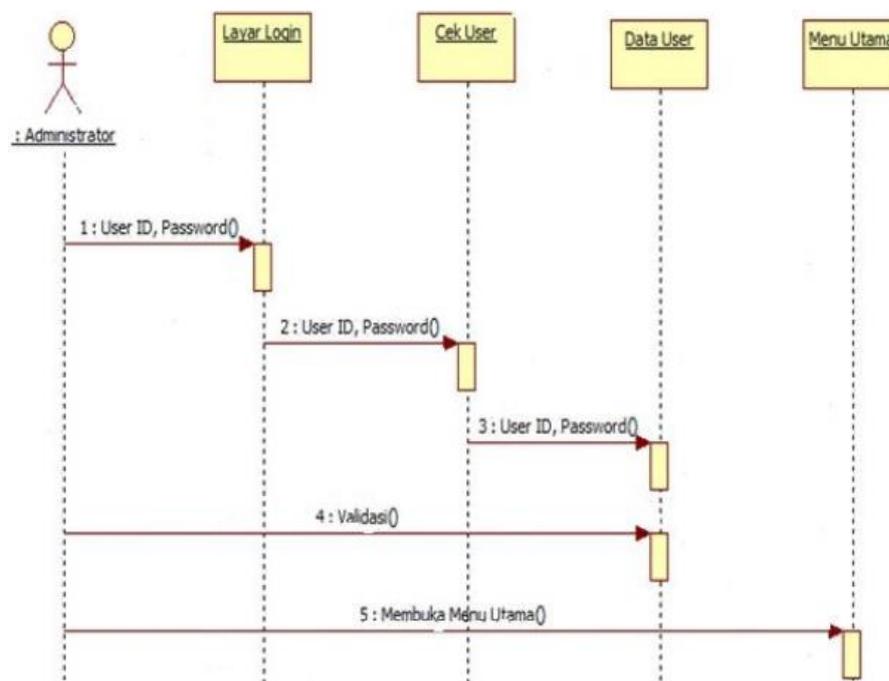
Class Diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen – komponen kelas dan hubungan antara masing – masing kelas. Selain itu, *Class* Diagram mendeskripsikan jenis – jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class* Diagram juga menunjukkan *property* dan operasi sebuah kelas serta batasan – batasan yang terdapat dalam hubungan – hubungan objek tersebut [36]. Berikut contoh *activity* diagram pada gambar dibawah ini.



Gambar 2.20 Contoh *class* diagram

2.13.4 Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan interaksi antar objek. *Sequence* Diagram secara khusus menjabarkan *behavior* sebuah *scenario* tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan – pesan yang melewati objek ini dalam sebuah *use case* [36].



Gambar 2.21 Contoh *sequence* diagram

