

BAB 2

TINJAUAN PUSTAKA

2.1 Kecelakaan Lalu Lintas

Kecelakaan lalu lintas berdasarkan UU No. 22 Tahun 2009 adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan/atau kerugian harta benda. Di dalam undang undang ini kecelakaan digolongkan menjadi 3 yaitu :

- a. Kecelakaan lalu lintas ringan yang merupakan kecelakaan yang mengakibatkan kerusakan kendaraan dan/barang.
- b. Kecelakaan lalu lintas sedang yang merupakan kecelakaan yang mengakibatkan luka ringan dan kerusakan kendaraan dan/barang.
- c. Kecelakaan lalu lintas berat yang merupakan kecelakaan yang mengakibatkan korban meninggal dunia dan/luka berat.

2.1.1 Faktor Penyebab Kecelakaan

Menurut Austroad (2002), Warpani (1999), dan Pignataro (1973) secara umum faktor yang paling berkontribusi dalam kecelakaan lalu lintas antara lain faktor manusia (Pengemudi dan pejalan kaki), kendaraan, jalan dan lingkungan jalan. Pignataro juga menyatakan bahwa kecelakaan diakibatkan oleh kombinasi dari beberapa faktor perilaku buruk dari pengemudi atau pejalan kaki, jalan, kendaraan, cuaca buruk ataupun pandangan yang buruk. Kecelakaan lalu lintas ditimbulkan oleh adanya pergerakan dari alat alat angkutan karena adanya kebutuhan perpindahan manusia dan atau barang. Kecelakaan timbul jika salah satu unsur pembentuk lalu lintas tidak berperan sebagaimana mestinya Unsur pembentuk lalu lintas antara lain pemakai jalan, kendaraan, jalan, dan lingkungan. Jadi dapat disimpulkan bahwa kecelakaan terjadi akibat dari salah satu faktor atau kombinasi dua faktor penyebab kecelakaan atau lebih. Faktor penyebab kecelakaan dapat dikelompokkan menjadi 4 yaitu faktor manusia, faktor prasarana (jalan), faktor sarana (kendaraan), dan faktor lingkungan atau cuaca.

2.1.1.1 Faktor Manusia

Penyebab kecelakaan lalu lintas di Indonesia paling banyak disebabkan oleh faktor manusia yaitu sebesar 91% (Direktorat Keselamatan Transportasi Darat atau DKTD (2006) Faktor manusia dapat dikelompokkan menjadi dua yaitu kondisi pengemudi dan usia pengemudi.

a. Kondisi Pengemudi

Lima faktor yang menyebabkan kecelakaan yaitu fisik pengemudi, tingkat kedisiplinan dan pemahaman berlalu lintas masih rendah, kecakapan pengemudi, jarak pandang yang kurang (dalam mengambil jarak aman antar kendaraan), dan pelanggaran nilai batas kecepatan maksimum kendaraan (*Speeding*).

b. Usia Pengemudi

Berdasarkan usia pelaku kecelakaan lalu lintas, sebagian besar berusia antara 22 S.d 30 tahun, kemudian disusul usia antara 31 s.d 40 tahun, dimana pada rentang usia tersebut tergolong sebagai usia tingkat emosinya paling stabil, tingkat kecekatan dan reflek yang lebih baik dibanding golongan usia lainnya, namun biasanya pada usia golongan ini tingkat mobilitasnya di jalan juga sangat tinggi. Jika pelaku kecelakaangolongan ini juga sekaligus menjadi korban, maka hal ini sekaligus merupakan golongan usia yang paling produktif. World health Organization (WHO) mencatat hamper 1,2 juta orang di seluruh dunia setiap tahun tewas akibat kecelakaan di jalan. Dari jumlah itu, 40 persen berusia di bawah 25 tahun. Jutaan lagi mengalami cedera dan sebagian lagi mengalami cacat seumur hidup. (Direktorat Jenderal Perhubungan Darat atau Ditjen Hubdat, 2004).

2.1.1.2 Faktor Sarana (Kendaraan)

Jenis kendaraan yang terlibat dalam kecelakaan lalu lintas sebagian besar adalah sepeda motor dengan persentase pada empat tahun terakhir rata-rata sebesar 62,62 % kemudian diikuti oleh jenis kendaraan mobil penumpang sebesar 36%, kendaraan barang 29,62% dan bus sebesar 10,56%. Tingkat resiko terjadinya kecelakaan akibat ketidaklayakan kendaraan cukup tinggi, Sehingga diperlukan

ketegasan dari aparat penegak hukum untuk menindak pelanggaran tersebut. Kendaraan dapat menjadi faktor penyebab kecelakaan apabila tidak dapat dikendalikan sebagaimana mestinya yaitu sebagai akibat kondisi teknisnya yang tidak layak jalan atau pengguna yang tidak sesuai dengan ketentuan. (Gito dan Mina (2015) .

2.1.1.3 Faktor Prasarana (Jalan)

Faktor yang disebabkan oleh faktor jalan dapat diklasifikasikan sebagai berikut :

- a. Kecelakaan lalu lintas yang disebabkan oleh perkerasan jalan Kondisi permukaan perkerasan dalam hal ini berhubungan dengan permasalahan keselamatan dan kenyamanan sangat erat kaitannya dengan aspek kelicinan dan kecepatan. Dengan dapat menambah tingkat resiko kecelakaan lalu lintas. Kelicinan dapat terjadi karena berkurangnya koefisien gesekan yang bisa ditimbulkan terutama oleh cuaca serta kotoran lumpur dan tumpahan minyak.
- b. Kecelakaan lalu lintas yang disebabkan oleh alinyemen jalan.
- c. Kecelakaan lalu lintas yang disebabkan oleh pemeliharaan jalan.
- d. Kecelakaan lalu lintas yang disebabkan oleh penerangan jalan.
- e. Kecelakaan lalu lintas yang disebabkan oleh tanjakan dan turunan.

2.1.1.4 Faktor Lingkungan

Faktor lingkungan jalan sangat berpengaruh dalam kegiatan lalu lintas. Hal ini mempengaruhi pengemudi dalam mengatur kecepatan (mempercepat, konstan, memperlambat atau berhenti). Faktor-faktor yang mempengaruhi kondisi lingkungan (Oglesby dan Hick, 1999), antara lain :

- a. Di dalam kota, misalnya di daerah pasar, pertokoan perkantoran,sekolah, perumahan, dan lain sebagainya
- b. Di luar kota, misalnya di daerah datar, perdesaan pegunungan atau sebagainya
- c. Di tempat khusus, misalnya di depan tempat ibadah, rumah sakit, tempat wisata dan sebagainya.

2.2 Landasan Teori

Landasan teori bertujuan sebagai acuan atau pedoman agar setiap materi yang digunakan sesuai dengan fakta yang ada, beberapa landasan teori yang akan dibahas diantaranya adalah sebagai berikut :

2.2.1 Sensor Pada Perangkat Android

Komatineni dan MacLean (2012) mendefinisikan sensor pada Android sebagai Bagian dari hardware yang ditanamkan dalam perangkat untuk mengirim data dari lingkungan fisik ke aplikasil. Definisi lainnya tentang sensor pada Android dikemukakan oleh Milette dan Stroud (2012) yang menyatakan sensor sebagai Sebuah kemampuan yang dapat merekam pengukuran terhadap perangkat dan lingkungan eksternalnyall.

Milette dan Stroud (2012) memaparkan jenis sensor pada perangkat Android sebagai berikut :

- a. Sensor Lokasi: Menentukan lokasi perangkat menggunakan berbagai sensor termasuk GPS.
- b. Sensor Fisik: Mendeteksi sifat perangkat secara spesifik seperti orientasi, akselerasi, dan rotasi beserta sifat lingkungan seperti cahaya, medan magnet, dan tekanan udara.
- c. NFC Scanner: Mendeteksi tag Near Field Communication (NFC) terdekat dan berbagi data dengan NFC perangkat Android lainnya yang aktif.
- d. Kamera: Mengumpulkan gambar visual.
- e. Mikropon: Merekam suara.
- f. Speech recognition: Mengonversi suara menjadi teks dengan menggunakan kombinasi dari suara hasil rekaman mikropon dan algoritma recognition.
- g. Sensor eksternal: Setiap sensor yang terhubung melalui mekanisme Android Open Accessory (AOA).

2.2.2 Kelas Sensor

Kelas (dalam pemrograman berorientasi objek) dapat didefinisikan sebagai cetak biru (blueprint) atau prototipe / kerangka yang mendefinisikan variabel-variabel (data) dan method-method (perilaku) umum dari sebuah objek tertentu

(Raharjo, Heryanto dan Haryono 2012). Sebagai contoh, mobil memiliki data seperti warna, tahun, merk, tipe, nomor polisi, dan sebagainya. Selain data atau ciri fisik tersebut, mobil juga memiliki perilaku-perilaku spesifik yang dapat membedakan antara mobil yang satu dengan yang lainnya, seperti sistem pengereman, perubahan roda gigi (persneling), dan sebagainya.

Kelas sensor merupakan penggambaran (representasi) dari hardware sensor pada perangkat Android (Milette dan Stroud 2012). Kelas ini menjelaskan informasi tentang sensor seperti: jangkauan (range) maksimum, delay minimum, nama, daya, resolusi, jenis, vendor, dan versi. Sensor yang direpresentasikan termasuk dalam jenis sensor fisik. Dalam website Android Developers (2014) dijelaskan bahwa secara garis besar ada tiga kategori sensor yang terdapat pada platform Android yaitu :

a. Sensor Gerak

Sensor ini mengukur gaya percepatan dan gaya rotasi sepanjang tiga sumbu (x, y, z). Kategori ini meliputi akselerometer, sensor gravitasi, giroskop, dan sensor vektor rotasi.

b. Sensor Lingkungan

Sensor ini mengukur berbagai parameter lingkungan seperti suhu dan tekanan udara sekitar, pencahayaan, dan kelembaban. Kategori ini meliputi barometer, fotometer, dan termometer

c. Sensor Posisi

Sensor ini mengukur posisi fisik dari perangkat. Kategori ini meliputi sensor arah / orientasi dan magnetometer.

Beberapa sensor pada android berbasis hardware sedangkan yang lainnya berbasis software. Sebagaimana dijelaskan dalam situs Android Developers (2014), sensor berbasis hardware adalah komponen fisik yang terpasang dalam sebuah handset atau tablet. Komponen ini memperoleh data secara langsung dengan mengukur sifat / properti lingkungan tertentu, seperti percepatan, kekuatan medan magnet, maupun perubahan sudut.

Sensor berbasis software bukanlah komponen fisik, meskipun menyerupai sensor berbasis hardware. Sensor berbasis software memperoleh datanya dari satu

atau lebih sensor yang berbasis hardware dan biasa disebut sensor virtual atau sensor sintesis. Sensor percepatan linear dan sensor gravitasi merupakan contoh dari sensor berbasis software.

Tabel 2. 1 Jenis Sensor yang Didukung Platform Android(Android Developers 2014)

Sensor	Basis	Deskripsi	Kegunaan Umum
TYPE_ACCELEROMETER	<i>Hardware</i>	Mengukur gaya percepatan dalam m/s ² pada perangkat di ketiga sumbu fisik (x, y, dan z) termasuk gaya gravitasi.	Mendeteksi pergerakan (<i>shake, tilt, dan lainnya</i>).
TYPE_AMBIENT_TEMPERATURE	<i>Hardware</i>	Mengukur suhu ruangan sekitar dalam derajat Celcius (°C).	Memantau suhu udara.
TYPE_GRAVITY	<i>Software</i> atau <i>Hardware</i>	Mengukur gaya gravitasi dalam m/s ² pada perangkat di ketiga sumbu fisik (x, y, dan z).	Mendeteksi pergerakan (<i>shake, tilt, dan lainnya</i>).
TYPE_GYROSCOPE	<i>Hardware</i>	Mengukur tingkat rotasi dalam rad/s pada tiap sumbu fisik (x, y, dan z).	Mendeteksi rotasi (<i>spin, turn, dan sebagainya</i>).
TYPE_LIGHT	<i>Hardware</i>	Mengukur level cahaya sekitar (iluminasi) dalam lx.	Mengontrol kecerahan layar.
TYPE_LINEAR_ACCELERATION	<i>Software</i> atau <i>Hardware</i>	Mengukur gaya percepatan dalam m/s ² pada perangkat di ketiga sumbu fisik (x, y, dan z), tanpa gaya gravitasi.	Memantau percepatan sepanjang sumbu tunggal.
TYPE_MAGNETIC_FIELD	<i>Hardware</i>	Mengukur medan geomagnetik sekitar di ketiga sumbu fisik (x, y, dan z) dalam μ T.	Membuat kompas.

Sensor	Basis	Deskripsi	Kegunaan Umum
---------------	--------------	------------------	----------------------

TYPE_ORIENTATION	<i>Software</i>	Mengukur sudut rotasi pada perangkat di ketiga sumbu fisik (x, y, dan z). Pada API level 3 kita bisa memperoleh matriks inklinasi dan matriks rotasi dengan menggunakan sensor gravitasi dan sensor medan geomagnetik bersama dengan metode <i>getRotationMatrix()</i> .	Menentukan posisi perangkat.
TYPE_PRESSURE	<i>Hardware</i>	Mengukur tekanan udara sekitar dalam hPa atau mbar.	Memantau perubahan tekanan udara.
TYPE_PROXIMITY	<i>Hardware</i>	Mengukur kedekatan objek dalam cm terhadap layar perangkat. Sensor ini biasanya digunakan untuk menentukan apakah <i>handset</i> ada di dekat telinga seseorang.	Posisi telepon selama panggilan.
TYPE_RELATIVE_HUMIDITY	<i>Hardware</i>	Mengukur kelembaban relatif sekitar dalam persen (%).	Pemantauan titik embun, absolut, dan kelembaban relatif.
TYPE_ROTATION_VECTOR	<i>Software</i> atau <i>Hardware</i>	Mengukur orientasi perangkat dengan menyediakan tiga unsur vektor rotasi perangkat.	Mendeteksi gerakan dan rotasi.

Sensor	Basis	Deskripsi	Kegunaan Umum
--------	-------	-----------	---------------

TYPE _TEMPERATURE	<i>Hardware</i>	Mengukur suhu perangkat dalam derajat Celcius (°C). Implementasi sensor ini bervariasi di seluruh perangkat dan sensor ini diganti dengan sensor TYPE_AMBIENT_TEMPERATURE di API level 14.	Memantau suhu.
----------------------	-----------------	--	----------------

2.2.3 Internet

Internet merupakan singakatan dari Interconnecting Networking. Internet ialah meruapakan hubungan antara berbagai jenis computer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya dimana hubungan tersebut memanfaatkan kemajuan komunikasi (telepon dan satelit) yang menggunakan protocol stansar dalam berkomunikasi yaitu protocol TCP/IP (Transmission Control/Internet Protocol) [8].

Dalam prakteknya, internet memunculkan istilah baru, yakni dunia maya. Sedangkan dunia di mana kita hidup disebut dunia nyata. Internet berada di antara keduanya. Karena salah satu fungsi internet adalah sebagai penghubung antara dunia nyata dan dunia maya. Dunia maya adalah tempat para pengguna internet berkomunikasi. Sehingga internet menjadi sebuah jaringan komunikasi global. Berjuta orang di seluruh dunia menggunakan internet untuk berbagai hal, mulai keperluan pribadi, organisasi, sampai keperluan perusahaan. Masyarakat Indonesia di berbagai daerah juga sudah banyak yang menggunakan internet, Tidak hanya di perusahaan, penggunaan internet juga masuk ke sekolah-sekolah sebagai sarana dalam kegiatan pembelajaran. Akses internet bahkan sudah mudah digunakan di rumah-rumah [8].

Karena begitu banyak manfaat yang dapat diperoleh dengan menggunakan internet, maka keberadaan internet telah menjangkau seluruh dunia. Sebagai sumber daya informasi yang sangat luas dan sangat besar, internet tidak dapat ditangani sendiri oleh satu orang, satu organisasi, atau satu negara pun. Kenyataannya, tidak ada satu orang yang mampu memahami seluruh seluk beluk internet [8].

2.2.4 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. [12]

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. [12]

Beberapa pengertian lain dari Android, yaitu:

1. Merupakan platform terbuka (*open source*) bagi para *programmer* untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli Google Inc dari Android Inc.

Bukan bahasa pemrograman, akan tetapi hanya menyediakan lingkungan hidup atau run time *environment* yang di sebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk *device* dengan sistem memori yang kecil.

2.2.4.1 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, diantaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. Hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. [12]

Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi android terbaru yaitu versi 4.0. (Ice Cream Sandwich). Android juga sudah bergabung dengan beberapa smart mobile seperti LG, Samsung, Sony Ericsson, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM

yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010). [12]

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (Mobile) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. [12]

2.2.4.2 Versi Android

Versi Android yang telah dikeluarkan hingga 20 Oktober 2019 adalah sebagai berikut :

1. Android versi 1.1

Pertama kali platform Android diluncurkan pada tahun 2009, setahun sebelumnya Google telah merilis versi beta yang diperkenalkan kepada khalayak ramai. Android versi pertama dikenal dengan Android 1.1. Berikut adalah fitur-fitur yang tersedia:

- Mampu menyimpan attachment dalam pesan
- Waktu default screen in-call lebih lama bila menggunakan speakerphone plus kemampuan untuk menampilkan atau menyembunyikan dialpad. [12]

2. Android versi 1.5 (Cupcake)

Pada tanggal 27 April 2009 Google juga merilis Android versi Cupcake. Cupcake adalah versi Android yang memulai tradisi penamaan kue untuk rilis Android. Ini dapat menambahkan beberapa fitur dan peningkatan baru dibanding versi sebelumnya.

Berikut adalah fitur-fitur yang tersedia:

- Dukungan untuk tampilan aplikasi widget-miniatur yang bisa disematkan di aplikasi lain (seperti home screen) dan menerima update berkala
- Rekaman video ditambahkan ke kamera bersamaan dengan kemampuan untuk langsung mengupload video ke YouTube.

3. Android versi 1.6 (Donut)

Google merilis Android 1.6 Donut pada bulan September tahun 2009. Penambahan fitur terbesar adalah dukungan untuk CDMA yang memperluas pasar pengguna baru ke Android. CDMA adalah teknologi yang digunakan oleh jaringan mobile Amerika pada saat itu..

Berikut adalah fitur-fitur yang tersedia:

- Navigasi Google Maps ditambahkan bersamaan dengan dukungan navigasi satelit.
- Donut menyertakan fitur gallery untuk memperlancara penangkapan media
- Pengenalan Search Box
- Toggling cepat antara kamera

4. Android versi 2.0 - 2.1 (Éclair)

Pada bulan Oktober 2009, sekitar setahun setelah peluncuran Android 1.0, Google merilis versi 2.0 dari OS, dengan nama Android Eclair. Versi ini adalah pertama yang menambahkan dukungan Text-to-Speech.

Berikut adalah fitur-fitur yang tersedia:

- Dukungan multi-touch ditambahkan ke Android.
- Dukungan untuk mencari di dalam pesan teks.
- Eclair membawa kotak masuk terpadu ke Android. Dukungan untuk beberapa akun Google telah ditambahkan.

5. Android versi 2.2.3 (Froyo)

Android versi terbaru kembali dirilis pada 20 Mei 2010. Google menamainya dengan Froyo. Nama Froyo ini diambil dari singkatan frozen yogurt. Smartphone pertama yang membawa merek Google Nexus, Nexus One, dirilis dengan Android 2.1 dari kotak pada awal tahun 2010, namun dengan cepat menerima update *over-the-air* ke Froyo akhir tahun. Pada android versi ini mulai dilengkapi dengan fitur *friendly* user seperti opsi untuk mematikan akses data pada jaringan seluler.

Berikut adalah fitur-fitur yang tersedia:

- *Hotspot WiFi portabel* untuk berbagi koneksi 3G perangkat dengan *gadget* lainnya.
- Setelan bergabung dengan kontak dan email untuk memback up ke *server* Google yang memungkinkan pembaca mengembalikan segalanya secara otomatis ke perangkat baru.
- *Flash* telepon juga bisa digunakan dalam *video*.

6. Android versi 2.3 - 2.3.7 (Gingerbread)

Gingerbread dirilis pada tahun 2010 . Pada 13 September 2017, Google menunjukkan bahwa hanya 0,6 persen dari semua perangkat Android yang saat ini menjalankan beberapa versi Gingerbread. Fitur utama termasuk dukungan *NFC*, *SIP* untuk panggilan Internet.

Berikut adalah fitur-fitur yang tersedia:

- *UI overhaul* untuk menghindari *screen burn-in* dan meningkatkan daya tahan baterai.
- Dukungan kamera menghadap depan untuk panggilan *video*.
- *Download manager* untuk mengawasi *download* Pembaca.
- Peningkatan pada *keyboard* layar dengan cara pintas dan kursor untuk membantu *copy paste*.

7. Android versi 3.0 - 3.2.6 (Honeycomb)

Honeycomb diluncurkan pada bulan Mei 2011 pada dasarnya untuk memperluas Android untuk mendukung layar tablet. Versi Android ini paling diabaikan dari semua. Karena dirilis khusus untuk tablet dan tidak pernah sampai ke ponsel.

Berikut adalah fitur-fitur yang tersedia:

- Beberapa perbaikan UI memanfaatkan layar besar.
- Tombol perangkat keras dijatuhkan untuk mendukung tombol di layar.
- Browser web mengenalkan *tabbed browsing*.
- Widget yang lebih besar.
- Aplikasi seperti Gmail dan YouTube dirancang ulang untuk menggunakan layar besar.

8. Android versi 4.0 - 4.0.4 (Ice Cream Sandwich)

Android Ice Cream Sandwich dirilis pada bulan Oktober 2011, versi Android Ice Cream Sandwich menghadirkan sejumlah fitur baru bagi pengguna. Ini menggabungkan banyak fitur versi Honeycomb tablet saja dengan smartphone yang berorientasi pada Gingerbread.

Berikut adalah fitur yang menyertainya:

- Desain ulang terbesar ke Android dengan tema Holo.
- Pengalaman browsing lebih cepat.
- Ruang penyimpanan multi untuk aplikasi.
- Pengenalan wajah untuk membuka kunci telepon.

9. Android versi 4.1 - 4.3.1 (Jelly Bean)

Jellybean dimulai pada bulan Juni 2012 dengan dirilis Android 4.1. Google dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di bawah label Jelly Bean, masing-masing pada bulan Oktober 2012 dan Juli 2013.

Berikut adalah fitur yang menyertainya:

- Google Now, tool bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian.
- Project Butter untuk mendukung frame rate yang lebih tinggi saat menggesek menu dan layar rumah.
- Mampu melihat foto dengan menggesek dari kamera untuk menuju ke filmstrip.
- Widget menyetel ulang diri mereka untuk menambahkan yang baru.
- Pemberitahuan fitur yang lebih banyak.
- Fitur gerak dan aksesibilitas baru.

10. Android versi 4.4 (Kitkat)

Android KitKat dirilis pada November 2013 ,Android 4.4 adalah satu-satunya versi OS yang benar-benar menggunakan nama sepotong permen.dan menjadi salah satu versi Android yang paling disukai oleh pengguna Smartphone di dunia.

KitKat memiliki fitur yang istimewa dari OS Android sebelumnya. Berikut adalah fitur yang menyertainya:

- Immersive mode untuk konsumsi konten yang lebih baik.

- Bar navigasi yang lebih baik untuk masuk dan keluar dari mode Immersive.
- Dukungan widget layar kunci.
- Dialer baru dengan fitur Caller ID.
- Wallpaper layar penuh.
- Emoji keyboard untuk emoticon.
- Aplikasi Hangouts dan perpesanan terpadu
- Dukungan cloud print yang lebih baik.
- Integrasi Google Now yang lebih cerdas dan handsfree.

11. Android versi 5.0 (Lollipop)

Android 5.0 Lollipop pertama kali diperkenalkan pada Mei 2014., Android lollipop merupakan perancangan ulang terbesar untuk Android. Smartphone Google Nexus 6, bersama dengan tablet Nexus 9-nya, merupakan perangkat pertama yang memiliki Lollipop yang telah terpasang sebelumnya.

Peningkatan terbesar yang dilakukan oleh Lollipop adalah pengenalan Material Design yang dengan cepat menjadi bahasa desain terpadu yang diterapkan di seluruh produk Google. Berikut adalah fitur yang dimilikinya:

- Dukungan pengaturan cepat yang lebih baik.
- Masa pakai baterai yang disempurnakan dengan mode Battery
- Layar kunci baru.
- Fitur Smart Lock melalui Layanan Google Play.
- Mode tamu untuk berbagi perangkat.
- Pemasangan tombol.

12. Android versi 6.0 (Marshmallow)

Android 6.0 (Marshmallow) Di rilis pada tahun 2015. Ini Perangkat pertama yang dikirim bersama Marshmallow yang telah terpasang sebelumnya adalah smartphone Google Nexus 6P dan Nexus 5X, dengan tablet Pixel C-nya. Tujuan marshmallow memoles sudut kasar dan membuat versi Lollipop lebih baik lagi.

Berikut adalah fitur yang dimilikinya:

- Dukungan sidik jari resmi untuk perangkat.
- Dukungan untuk pembayaran seluler melalui Android Pay.
- Model perizinan yang lebih baik untuk aplikasi.
- Google Now di Tap.
- Deep menghubungkan Apps.

13. Android versi 7.0 (Nougat)

Android 7.0 (Nougat) Dirilis pada Tahun ,2016. Sebelum Nougat terungkap "Android N" dirujuk secara internal oleh Google sebagai "New York Cheesecake".

Berikut adalah fitur yang dimilikinya:

- Doze on the Go untuk waktu siaga yang lebih baik lagi.
- Multi Window untuk penggunaan dua aplikasi secara bersamaan.
- Aplikasi Setelan yang Lebih Baik.
- Hapus semua di layar aplikasi baru-baru ini.
- Balas langsung ke pemberitahuan.
- Notifikasi dibundel.
- Pengaturan Cepat akan mengubah kustomisasi.

14. Android versi 8.0 (Oreo)

Pada bulan Maret 2017, Google Rilis Android 8.0 Oreo, bulan Agustus, Google mengkonfirmasi Oreo akan menjadi nama publik untuk Android 8.0. Sperti yang kita ketahui Ini adalah kedua kalinya Google memilih nama merek dagan untuk Android (Oreo dimiliki oleh Nabisco).

Adapaun sekarang versi ini adalah semua yang baru yang dimilikinya, diantaranya adalah sebagai berikut:

- Pemberitahuan untuk prioritas dan kategorisasi yang lebih baik.
- Pengelolaan warna lebih baik.
- Android O memiliki koleksi emoji baru yang telah didesain ulang.
- Waktu boot lebih cepat: Pada perangkat Pixel, sekarang bisa mengalami waktu boot dua kali lebih cepat dibandingkan dengan Nougat.

Mengisi otomatis dan mengingat kata sandi dalam aplikasi.

2.2.5 Software Pendukung

Dalam pembangunan aplikasi pada tugas akhir ini penulis menggunakan *software* pendukung yang digunakan adalah sebagai berikut :

Android Studio

Android studio adalah lingkungan pengembangan terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi android, yang didasarkan pada IntelliJ IDE. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktifitas anda dalam membuat aplikasi android [12], seperti :

- a. Sistem build berbasis Gradle yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan terpadu tempat anda bisa mengembangkan aplikasi untuk semua perangkat Android
- d. Terapkan perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- e. Template kode dan integrasi GitHub untuk membantu anda membuat fitur aplikasi umum dan mengimpor kode sampel
- f. Framework dan fitur pengujian yang lengkap
- g. Fitur lint untuk merekan performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- h. Dukungan C++ dan NDK

Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

2.2.6 Bahasa Pemograman yang digunakan

2.2.6.1 Java

Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang baha computer untuk perangkat consumer seperti *cable-tv Box*. Dikarenakan perangkat tersebut tidak memiliki banyak memori, Bahasa harus berukuran kecil dan mengandung kode yang padat. Juga karena manufaktur-manufaktur berbeda memilih *processor* yang berbeda pula, maka Bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode “Green” [12].

Karena banyak orang-orang di proyek Green berbasis C++ dan bukan pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan procedural. Mulanya Bahasa yang diciptakan diberi nama “Oak” oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama Bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA, nama JAVA sendiri terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal biji kopi. Akhirnya mereka sepakat untuk memberikan nama Bahasa pemrograman tersebut dengan nama JAVA [12].

Java telah mengakomodasi hampir seluruh fitur-fitur penting Bahasa-bahasa pemrograman yang ada semenjak perkembangan komputasi modern manusia :

1. Dari SIMULA, Bahasa pada tahun 65-an, Bahasa yang paling mempengaruhi JAVA sekaligus C++. Dari Bahasa ini diadopsi bentuk-bentuk dasar dari pemrograman berorientasi objek.
2. Dari LISP – Bahasa tahun 55-an. Diadopsi fasilitas *garbage collection*, serta kemampuan untuk meniru *generic list processing*, meski fasilitas ini jarang yang memanfaatkannya.
3. Dari Algol – Bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
4. Dari C++, diadopsi sintaks, sebagian semantics dan *exception handling*.
5. Dari Bahasa Ada, diambil *strongly type*, dan *exception handling*,
6. Dari Objective C, diambil fasilitas interface.
7. Dari Bahasa SmallTalk, diambil pendekatan *single root class hierarchie*, dimana *object* adalah satu kesatuan hirarki pewarisan.

Dari Bahasa Eiffel, fasilitas *assertion* yang mulai diterapkan di sebagian JDK 1.4.

2.2.7 Global Position System (GPS)

GPS adalah singkatan dari *Global Position Sistem*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari

satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti :

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi yaitu Latitude, Longitude, dan Elevasi.
3. Kecepatan. Ketika berpindah tempat GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
5. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.

2.2.7.1 Akurasi GPS

Posisi yang ditunjukkan oleh suatu GPS mempunyai faktor kesalahan atau juga disebut tingkat akurasi. Sebagai contoh suatu alat GPS menunjukkan titik koordinat dengan tingkat akurasi 5 meter, itu berarti posisi pengguna bisa berada dalam range radius 5 meter dari titik yang ditunjukkan tersebut [8].

Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antar lain :

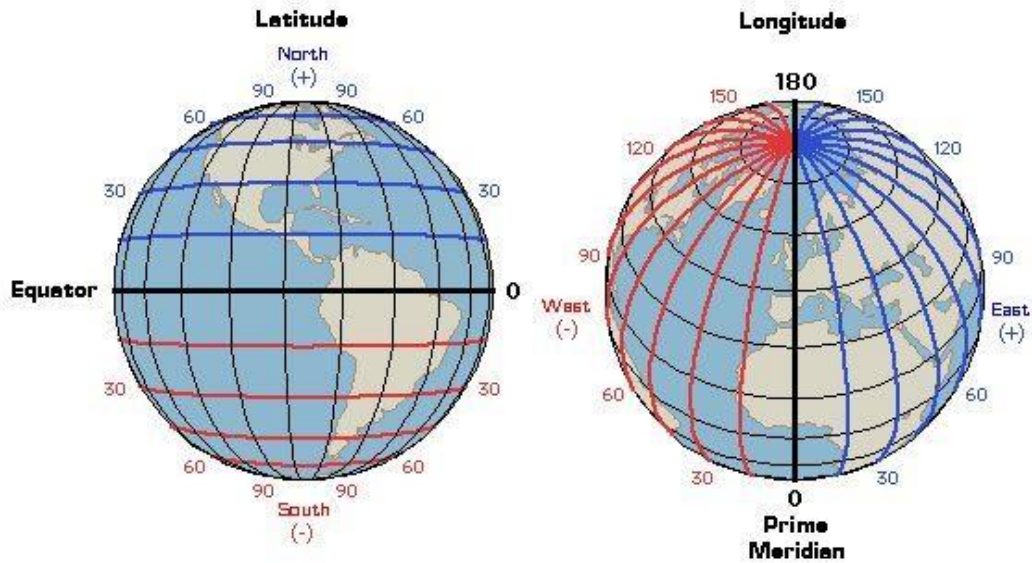
1. Kesalahan Ephemeris. Terjadi jika satelit tidak dapat mentransmisikan posisinya di orbit dengan tepat.
2. Keadaan Ionosphere. Ionosphere berada pada jarak sekitar 43-50 mil di atas permukaan bumi. Satelit yang melewati ionosphere akan menjadi lambat dikarenakan adanya plasma (gas dengan kepadatan rendah). Walaupun GPS *receiver* berusaha untuk mengoreksi/memperbaiki faktor keterlambatan yang terjadi tetap saja aktifitas tertentu dari plasma bisa menyebabkan kesalahan perhitungan.
3. Keadaan Troposphere. Troposphere adalah bagian bagian terendah dari atmosfer sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada temperature, tekanan, dan kelembaban bisa menyebabkan perbedaan kecepatan penerimaan gelombang radio.

4. Kesalahan Waktu. Karena penempatan jam atom pada setiap GPS *receiver* tidak berjalan sebagaimana mestinya. Kesalahan waktu dari GPS *receiver* yang tidak presisi dapat menimbulkan ketidakakurasian.
5. Kesalahan Multipath. Terjadi karena sinyal satelit membentur permukaan keras (seperti bangunan atau tebing) sebelum mencapai GPS *receiver*. Hal tersebut bisa menyebabkan terjadinya delay sehingga perhitungan jarak menjadi tidak akurat.
6. Buruknya Sinyal Satelit. Keadaan langit yang terhalang akan menyebabkan GPS diterima dari satelit. Sebuah sinyal satelit yang pada hari tertentu dengan kualitas yang sama walaupun user berdiri pada tempat yang sama. Hal tersebut dikarenakan posisi dari satelit yang terus bergerak atau bisa juga disebabkan factor penghalang lain seperti pohon, Gedung bertingkat dan sebagainya.

2.2.7.2 Latitude dan Longitude

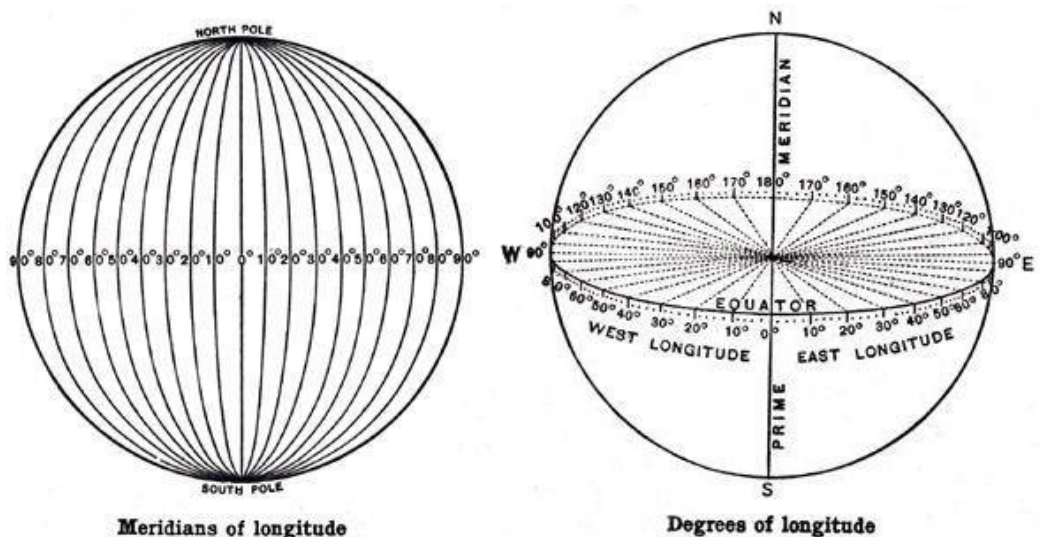
1. Latitude adalah garis yang melintang dari kutub utara dan kutub selatan. Titik 0 adalah sudut ekuator, tanda + menunjukkan arah ke atas menuju kutub utara, sedangkan tanda minus di kordinat Latitude menuju ke kutub selatan. Titik yang dipakai dari 0 ke 90 derajat arah kutub utara, dan 0 ke -90 derajat ke kutub selatan [9].

Gambar 2. 1 Latitude



2. Longitude adalah garis lintang. Angka dari sudut bundar bumi horizontal. Titik diawali dari 0 ke 180 derajat dan 0 ke -180 ke arah sebaliknya. Titik 0 dimulai dari garis negara inggris. Mengarah ke Indonesia akan menjadi angka positif. Kebalikannya koordinat longitude minus adalah arah kebalikan [9].

Gambar 2. 2 Longitude



2.2.7.3 Location-Based Service (LBS)

Location Based Service (LBS) adalah layanan informasi yang dapat diakses menggunakan piranti *mobile* melalui jaringan internet dan seluler serta memanfaatkan kemampuan penunjukan lokasi pada piranti *mobile* [4].

2.2.7.4 Komponen Location-Based Service (LBS)

Terdapat empat komponen pendukung utama dalam teknologi *location based service* [4], antara lain :

1. Piranti *Mobile*
2. Jaringan Komunikasi
3. Komponen *Positioning* (Penunjukan posisi/lokasi)
4. Penyedia layanan dan aplikasi
5. Penyedia data dan konten

2.2.8 Database

Dikutip di buku Simarmata dan Paryudi, ada beberapa pengertian *Database* sebagai berikut :

1. Menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data.
2. Menurut Silberschatz, dkk (2002), mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.
3. Menurut Ramakrishnan dan Gehrke (2003) menyatakan basisdata kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.
4. Menurut McLeod, dkk (2001), adalah kumpulan seluruh sumber daya berbasis computer milik organisasi.

2.2.8.1 Database Management System (DBMS)

Database Management Sistem (DBMS) adalah paket perangkat lunak yang dirancang untuk mendefinisikan, memanipulasi, mengambil dan mengelola data dalam database. Suatu DBMS umumnya memanipulasi data itu sendiri, format data, nama bidang, struktur catatan dan struktur file. Itu juga mendefinisikan aturan untuk memvalidasi dan memanipulasi data ini. DBMS mengurangi pengguna dari program pembingkaihan untuk pemeliharaan data. Bahasa Query generasi keempat, seperti SQL, digunakan Bersama dengan paket DBMS untuk berinteraksi dengan database. Beberapa contoh DBMS yaitu :

- MySQL
- SQL Server
- Oracle

- dBASE
- FoxPro










Sistem manajemen basis data menerima instruksi dari *Database Administration* (DBA) dan demikian menginstruksikan sistem untuk melakukan perubahan yang diperlukan. Perintah-perintah ini dapat memuat, mengambil atau memodifikasi data yang ada dari sistem. DBMS selalu memberikan independensi data. Setiap perubahan dalam mekanisme dan format penyimpanan dilakukan tanpa memodifikasi seluruh aplikasi. Ada empat jenis utama dari organisasi basis data :

1. Database Relasional : Data disusun sebagai tabel yang secara logis independent. Hubungan antar tabel ditunjukkan melalui data Bersama. Data dalam satu tabel dapat merujuk data yang serupa di tabel lain, yang menjaga integritas tautan di antara mereka. Fitur ini disebut sebagai integritas referensial.
2. Database Flat : Data disusun dalam satu jenis catatan dengan jumlah bidang tetap. Tipe database ini menjumpai lebih banyak kesalahan karena sifat data yang berulang.
3. Database Berorientasi Objek : Data disusun dengan kemiripan dengan konsep pemrograman berorientasi objek. Suatu objek terdiri dari data dan metode, sedangkan kelas mengelompokkan objek yang memiliki data dan metode yang serupa.
4. Database Hirarki : Data diorganisasikan dengan hubungan hierarkis. Ini menjadi jaringan yang kompleks jika hubungan satu ke banyak dilanggar.

2.2.9 Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segemen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Gambar 2. 3 Simbol *Flowchart*

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

2.2.10 *Unified Model Language* (UML)

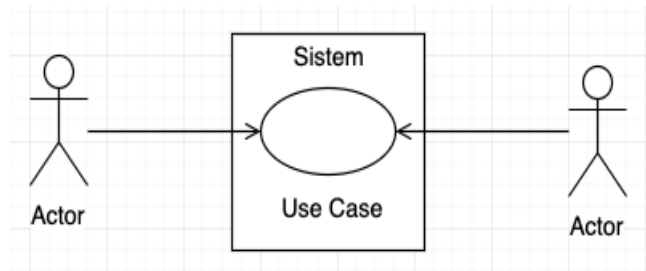
Unified Modeling Language (UML) adalah Bahasa pemodelan standar pada rekayasa perangkat lunak. Dengan menggunakan UML akan berdampak kepada peningkatan produktifitas dan kualitas serta pengurangan biaya dan waktu, kerumitan arsitektur dalam pengembangan [erangkat lunak bisa diatasi dengan menggambarkan cetak biru system tersebut [24].

2.2.10.1 Diagram UML

UML menyediakan 4 macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek [24], yaitu :

2.2.10.2 Use Case Diagram

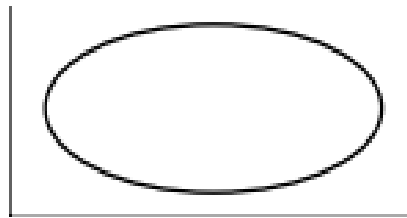
Use Case Diagram digunakan untuk menangkap aspek dinamis dari system. Secara lebih spesifik use case diagram digunakan untuk mengumpulkan kebutuhan dari sebuah system baik karena pengaruh internal maupun eksternal. Gambar dibawah ini menunjukkan *Use Case Diagram* dalam UML.

Gambar 2. 4 Use Case Diagram

Berikut ini adalah bagian dari sebuah *use case diagram* :

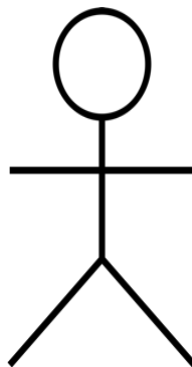
1. Use Case

Use Case adalah deskripsi fungsi dari sebuah system dari perpektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Gambar dibawah ini menunjukkan bentuk *Use Case* dalam UML.

Gambar 2. 5 Use Case

2. Actors

Actor adalah *abstraction* dari orang dan system yang lain yang mengaktifkan fungsi dari target system. Orang atau system bisa muncul dalam beberapa [eran. Gambar dibawah ini menunjukkan *actor* dalam UML.

Gambar 2. 6 Actors

3. Relationship

Relationship adalah hubungan antar *use case* dengan *actor*.

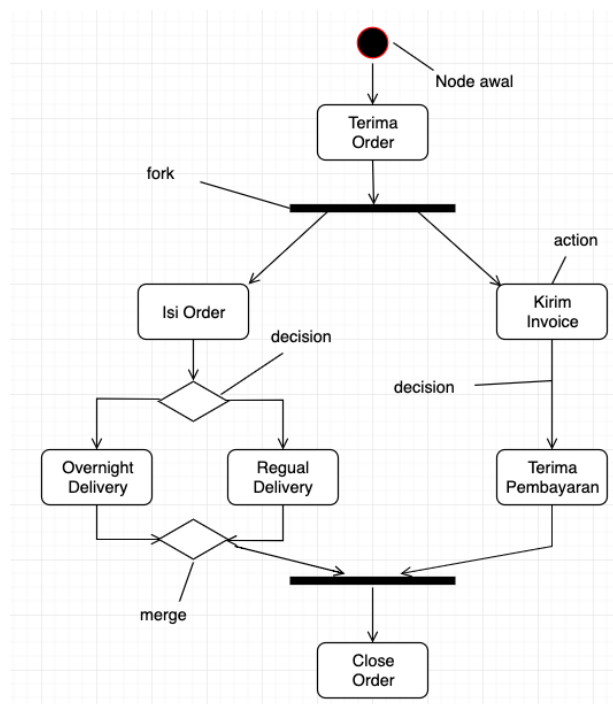
Relationship dalam use case diagram meliputi :

- a. Asosiasi antar *actor* dan *use case*
- b. Asosiasi antar 2 *use case*
- c. Generalisasi antara 2 *actor*
- d. Generalisasi antar 2 *use case*

2.2.10.3 Activity Diagram

Activity Diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari system. Logika procedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. Gambar dibawah ini menunjukkan *Activity Diagram* dalam UML.

Gambar 2. 7 Activity Diagram



Berikut ini merupakan komponen dalam *activity diagram*, yaitu :

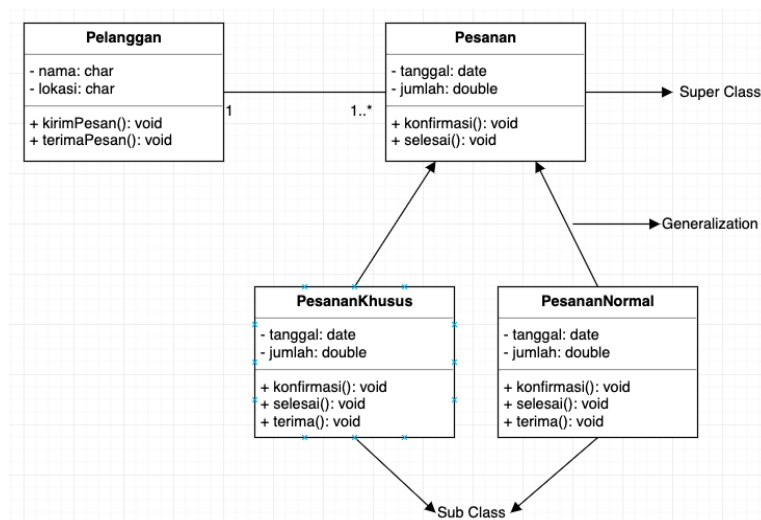
- a. *Activity Node* menggambarkan bentuk notasi dari beberapa proses yang beroperasi dalam control dan nilai data
- b. *Activity Edge* menggambarkan bentuk *edge* yang menghubungkan aliran aksi secara langsung dimana menghubungkan *input* dan *output* dari aksi tersebut

- c. *Initial State* adalah Bentuk lingkaran berisi penuh melambangkan awal dari suatu proses
- d. *Decision* adalah Bentuk wajib dengan suatu *flow* yang masuk beserta dua atau lebih *activity node* yang keluar. *Activity Node* yang keluar ditandai untuk menghasilkan beberapa kondisi
- e. *Fork* adalah Satu bar hitam dengan satu *activity node* yang masuk beserta dua atau lebih *activity node* yang keluar
- f. *Join* adalah Satu bar hitam dengan dua atau lebih *activity node* yang masuk beserta satu *activity node* yang keluar, tercatat pada akhir dari proses secara bersamaan. Semua *actions* yang menuju *join* harus sebelum proses dapat berlanjut.
- g. *Final State* adalah Bentuk lingkaran berisi penuh yang berada di dalam lingkaran kosong menunjukkan akhir dari suatu proses

2.2.10.4 Class Diagram

Class Diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan dan mendokumentasikan berbagai aspek system tetapi juga membangun kode eksekusi, dari aplikasi perangkat lunak. Gambar dibawah ini menunjukkan *class diagram* dalam UML.

Gambar 2. 8 Class Diagram



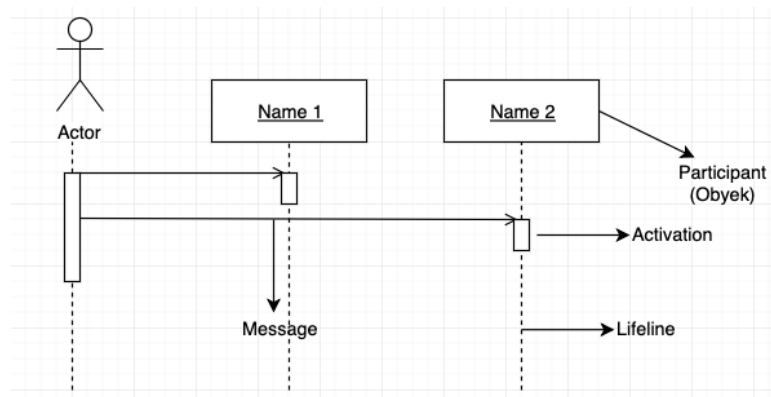
Class diagram mempunyai 3 relasi dalam penggunaannya, yaitu :

- Assosiation* adalah sebuah hubungan yang menunjukkan adanya interaksi antar *class*. Hubungan ini dapat ditunjukkan dengan garis dengan mata panah terbuka di ujungnya yang mengindikasikan adanya aliran pesa dalam satu arah.
- Generalization* adalah sebuah hubungan antar *class* yang bersifat dari khusus ke umum
- Constraint* adalah sebuah hubungan yang digunakan dalam system untuk memberi Batasan pada system sehingga didap aspek yang tidak fungsional.

2.2.10.5 Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek-objek di dalam *use case*. Gambar dibawa ini menunjukkan *Sequence Diagram* dalam UML.

Gambar 2. 9 Sequence Diagram



Berikut merupakan komponen dalam *sequence diagram* :

- Activation* menjelaskan tentang eksekusi dari fungsi yang dimiliki oleh suatu objek
- Actor* menjelaskan tentang peran yang melakukan serangkaian aksi dalam suatu proses
- Collaboration Boundary* menjelaskan tentang tempat untuk lingkungan percobaan dan digunakan untuk memonitor objek
- Parallel Vertical Lines* menjelaskan tentang suatu garis proses yang menunjukkan pada suatu state
- Processes* menjelaskan tentang tindakan/aksi yang dilakukan oleh aktor dalam suatu waktu

- f. *Window* menjelaskan tentang halaman yang sedang ditampilkan dalam suatu proses
- g. *Loop* menjelaskan tentang model logika yang berpotensi untuk diulang beberapa kali

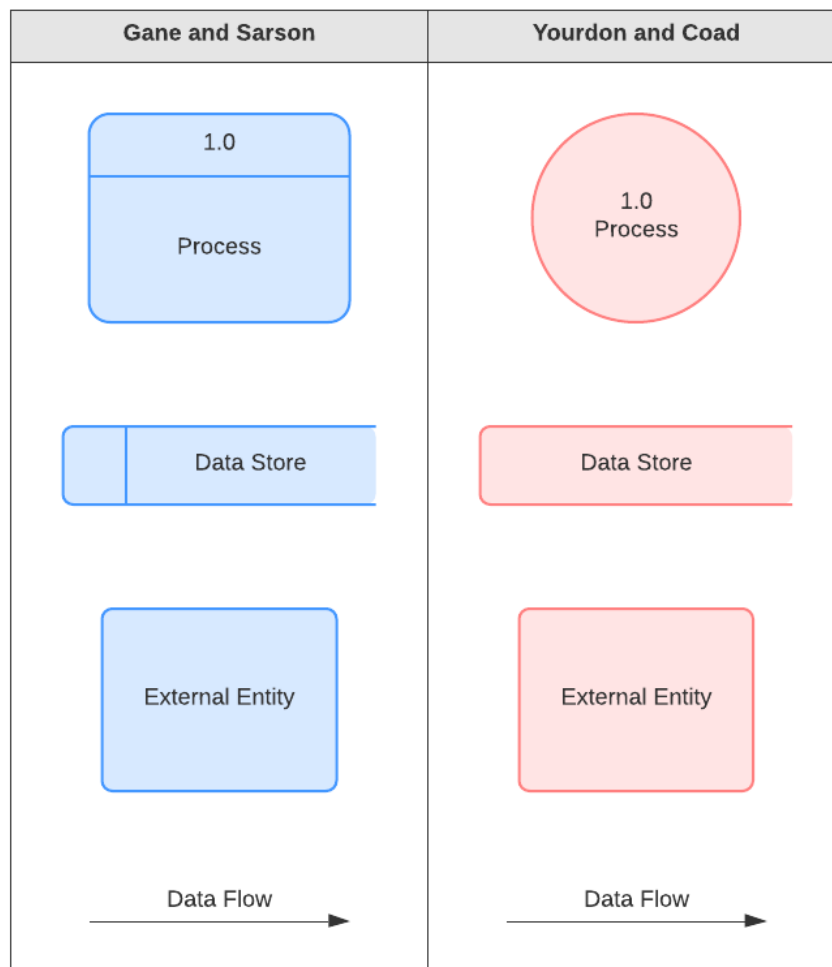
2.2.11 Data Flow Diagram (DFD)

Data Flow Diagram atau DFD menurut Sukamto dan Shalahudin [27] adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan dan keluaran. 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.

2.2.11.1 Simbol Data Flow Diagram

Data Flow Diagram (DFD) merupakan salah satu alat untuk mengetahui aliran data yang mengalir dalam sistem. DFD menggambarkan beberapa simbol sebagai berikut :

Gambar 2. 10 Simbol Data Flow Diagram



- a. *Process* adalah kegiatan atau pekerjaan yang dilakukan oleh semua orang, mesin, atau komputer dari arus data yang masuk ke proses dan akan dihasilkan arus data yang keluar dari proses. Arus data adalah arus yang mengalir dari proses atau arus yang mengalir dari proses menuju proses lain.
- b. *Data Store* digunakan untuk menunjukkan simpanan dari data yang dapat berupa suatu file atau database di sistem komputer, suatu arsip atau catatan manual
- c. *External Entity* dapat berupa orang, organisasi, departemen didalam organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

- d. *Data Flow* menunjukkan arah mengalirnya data. Arus data mengalir menuju proses dan atau meninggalkan data. Arus data mengalir menuju proses dan atau meninggalkan proses. Arus data ini dapat berupa masukan untuk sistem atau hasil dari proses sistem.

2.2.11.2 Tahapan Perencanaan *Data Flow Diagram*

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD :

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

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

2. Membuat *DFD Level 1*

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

3. Membuat *DFD Level 2*

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

4. Membuat *DFD Level 3* dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada *DFD Level* di atasnya. *Breakdown* pada level 3, 4, 5 dan seterusnya aturannya sama persis dengan *DFD Level 1* atau *level 2*. Pada suatu diagram *DFD* sebaiknya jumlah modul tidak melebihi dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit.