

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi. Landasan teori yang dibahas yaitu pengertian *peringatan dini* , *longsor* , *cloud messaging* , *Android*, *Java*, *Web Service*, *PHP*, *MySQL*, Metode yang digunakan dan bahasa pemrograman yang digunakan dalam pembangunan aplikasi.

2.2 Peringatan Dini

Sistem peringatan dini bencana adalah elemen yang sangat penting dalam upaya pengurangan risiko bencana. Dengan adanya peringatan dini bencana, maka masyarakat dapat melakukan respon yang sesuai untuk melakukan penyelamatan dan menghindari korban jiwa serta mengurangi dampak bencana tersebut. Agar sistem peringatan dini dapat berjalan secara efektif maka dibutuhkan partisipasi aktif masyarakat yang berada di daerah berisiko, memfasilitasi kegiatan-kegiatan penyadaran publik dan kesiapsiagaan masyarakat, serta penyampaian peringatan yang terpercaya.

Peringatan dini adalah serangkaian kegiatan pemberian peringatan sesegera mungkin kepada masyarakat tentang kemungkinan terjadinya bencana pada suatu tempat oleh lembaga yang berwenang. Masyarakat memiliki hak untuk berpartisipasi dalam pengambilan keputusan terhadap kegiatan penanggulangan bencana, khususnya yang berkaitan dengan diri dan komunitasnya serta berkewajiban untuk memberikan informasi yang benar kepada publik tentang penanggulangan bencana.

Peringatan dini sebagai salah satu bagian dalam penyelenggaraan penanggulangan bencana dalam situasi terdapat potensi bencana dilakukan untuk mengambil tindakan cepat dan tepat dalam rangka mengurangi risiko terkena bencana serta mempersiapkan tindakan tanggap darurat. Agar dapat berjalan

efektif, sistem peringatan dini harus dikelola secara terpadu dan menyeluruh, serta melibatkan secara aktif masyarakat dan para pemangku kepentingan terkait.

Syarat sebuah peringatan dini yang lengkap dan efektif serta berpusat pada masyarakat (people-centered) adalah terpenuhinya empat komponen yaitu pengetahuan risiko, pemantauan bahaya dan layanan peringatan, penyebaran dan komunikasi dan kemampuan respon. Tujuan utama sistem peringatan dini berbasis masyarakat adalah menguatkan individu dan masyarakat yang terancam bahaya untuk bertindak secara tepat waktu dan benar sehingga dapat mengurangi kemungkinan terjadinya kerusakan fisik seseorang dan kematian.

2.3 Longsor/gerakan Tanah

Longsoran merupakan salah satu jenis gerakan massa tanah atau batuan, atau percampuran keduanya, menuruni atau keluar lereng akibat dari terganggunya kestabilan tanah atau batuan penyusun lereng tersebut.

Tanah longsor dapat disebabkan oleh:

1. Penggundulan hutan; yang biasanya akan mengakibatkan berkurangnya daya ikat tanah;
2. Getaran pada lereng akibat gempa bumi ataupun ledakan, penggalian, getaran alat/kendaraan;
3. Peningkatan beban yang melampaui daya dukung tanah;
4. Pemotongan kaki lereng secara sembarangan yang mengakibatkan lereng kehilangan gaya penyangga.

Gejala terjadinya tanah longsor antara lain:

1. Munculnya retakan memanjang atau lengkung pada tanah atau pada konstruksi bangunan, yang biasanya terjadi setelah hujan;
2. Terjadinya penggembungan pada lereng atau tembok bangunan;
3. Tiba-tiba muncul rembesan atau mata air pada lereng;
4. Apabila pada lereng sudah terdapat rembesan air/mata air, air tersebut tiba-tiba menjadi keruh bercampur lumpur;

5. Pohon-pohon atau tiang-tiang miring searah kemiringan lereng;
6. Terdengar suara gemuruh atau suara ledakan dari atas lereng;
7. Terjadi runtuh atau aliran butiran tanah/kerikil secara mendadak dari atas lereng.

2.4 Cloud Messaging

Firestore Cloud Messaging biasa disebut FCM, sebelumnya dikenal sebagai Google Cloud Messaging (GCM), merupakan solusi dari cloud lintas platform untuk pesan dan layanan pemberitahuan untuk Android, iOS, dan aplikasi web, yang saat ini dapat digunakan tanpa biaya.

Layanan Firestore Cloud Messaging ini disediakan oleh Firestore, yang merupakan anak perusahaan dari Google. Pada tanggal 21 Oktober 2014, Firestore mengumumkan telah diakuisisi oleh Google untuk jumlah yang tidak diungkapkan. Situs web Google Cloud Messaging resmi menunjuk ke Firestore Cloud Messaging (Firestore Cloud Messaging) sebagai versi baru Google Cloud Messaging (GCM). Pihak Google menyarankan untuk aplikasi yang masih menggunakan Google Cloud Messaging untuk segera migrasi ke Firestore Cloud Messaging.

Fitur-fitur yang diberikan oleh Firestore Cloud Messaging sebenarnya tidak terlalu jauh berbeda dengan Google Cloud Messaging. Dengan Firestore Cloud Messaging kita bisa memberikan pemberitahuan dan membuat komunikasi dua arah antara perangkat.

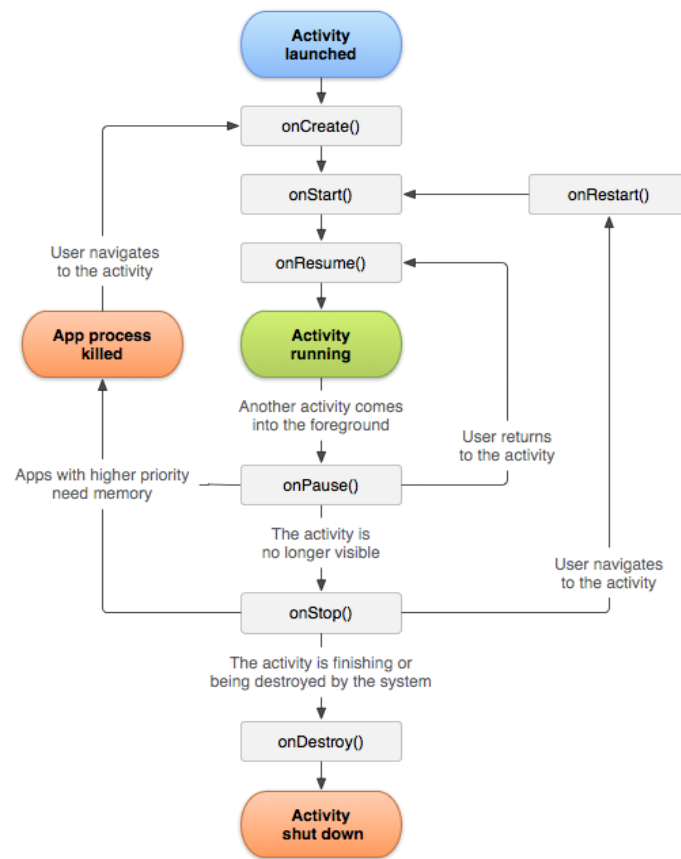
2.5 Android

Android adalah sebuah sistem operasi untuk perangkat seluler berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk seluler yang berbasis Linux. Android menyediakan wadah terbuka bagi para development untuk membuat aplikasinya sendiri. Pada awalnya diciptakan oleh *Android.Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google.Inc*. Untuk pengembangannya,

dibentuklah Open Handset Alliance (OHA), asosiasi dari 34 perusahaan yang termasuk Google, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

2.5.1. Android Life Cycle

Android memiliki pola pemrograman lain tidak seperti pola pemrograman biasa di mana aplikasi yang dijalankan pada fungsi `main()`, sistem android menjalankan kode dalam metode *Activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang ada dalam sistem *smartphone* android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web. Hal ini dikarenakan aplikasi ponsel memiliki tingkat interupsi proses yang lumayan tinggi seperti menerima panggilan masuk aplikasi diharuskan menghentikan proses sementara. Siklus hidup juga berguna untuk memastikan aplikasi agar tidak menguras daya baterai.



Gambar 2.1 Android Life Cycle

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.1 Siklus Hidup Android, akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya:

1. *Resumed*

Resumed terjadi saat aplikasi berjalan setelah state *paused* . State ini akan menjalankan perintah program yang ditulis pada *method* *onResume()*.

2. *Paused*

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas

lebih tinggi seperti panggilan telepon. Aplikasi tidak bisa menjalankan perintah apapun ataupun menampilkan apapun dalam *state* ini.[7]

3. *Stopped*

Dalam keadaan ini, aplikasi tidak dapat menampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service dibackground*.

State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan *state* berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem *OnCreate()* dipanggil, dengan cepat sistem akan memanggil method *OnStart()*, kemudian diikuti oleh *onResume()*.

2.5.2. Flowchart

Flowchart adalah gambaran langkah-langkah dan urutan prosedur dari suatu program. Flowchart juga dapat membantu untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart juga dapat mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti :

1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan *range* dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama.

Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol flowchart yang standar.

2.5.3. JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data *universal*. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

2.5.4. Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun

pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang menjual, maka diganti menjadi Java. Pada Desember 1998, Sun memperkenalkan nama “Java 2” (J2) sebagai generasi kedua dari java platform. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME).

Ada tiga platform Java yang masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda:

1. Standard Edition (J2SE) : J2SE merupakan inti dari bahasa pemrograman Java. J2SE didesain untuk jalan pada komputer desktop dan komputer workstations.
2. Enterprise Edition (J2EE): Dengan built-in mendukung untuk servlets, JSP, dan XML, edisi ini ditujukan untuk aplikasi berbasis server.
3. Micro Edition (J2ME) : Didesain untuk piranti dengan memori terbatas, layar tampilan terbatas dan power pemrosesan yang juga terbatas.

2.5.5. Web services

Service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

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

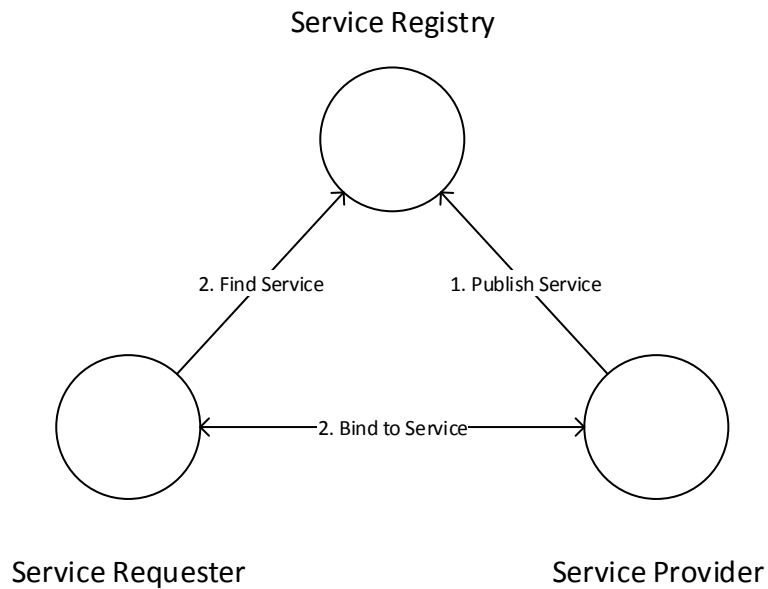
Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:[8]

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

2.5.6. Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

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



Gambar 2.2 Web Service

1. *Service Provider* berfungsi untuk menyediakan layanan dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
2. *Service Registry* berfungsi sebagai lokasi central yang mendeskripsikan semua layanan yang telah didaftarkan.
3. *Service Requestor* meminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.5.7. Processor Hypertext Protocol (PHP)

PHP merupakan salah satu pemrograman server-side di antara beberapa pemrograman yang ada. Sejak di luncurkan, PHP mendapat respon yang sangat baik dari kalangan pengembang aplikasi web kemudahannya untuk di pahami, serta sintaksnya yang mirip dengan Bahasa C menjadikan pemrograman ini cepat di kenal di kalangan luas.

Tanpa mengurangi kemampuan yang sudah ada, PHP 5 hadir dengan fitur lebih kompleks, dan merupakan rilis terbaru yang di dikeluarkan oleh pengembangnya secara besar besaran pengembang PHP merombak bug yang sering di jumpai dalam PHP 4, hal ini akan sangat jelas terasa ketika anda membuat aplikasi berbasis objek.

Dengan tujuan untuk lebih meningkatkan kinerja aplikasi, tanpa bermaksud membuang fitur yang ada, PHP 5 menambahkan fitur – fitur baru seperti dukungan terhadap MYSQL. Dukungan ini di wujudkan dalam bentuk fungsi lama dan di tulis ulang.

2.5.8. HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web. Website yang dibuat dengan HTML ini, dapat di lihat oleh semua orang yang terkoneksi dengan internet. Tentunya dengan menggunakan aplikasi penjelajah. HTML merupakan standarisasi penulisan untuk pembuatan suatu website, dan CSS merupakan pendamping HTML sebagai pengganti cara memformat tampilan, sedangkan Java Script adalah salah satu Bahasa Scripting yang dapat digunakan untuk melakukan proses lainnya yang tidak dapat dilakukan oleh HTML dan CSS[7].

Tiap kali kita mengakses dokumen web, maka sesungguhnya kita mengakses dokumen seseorang yang ditulis dengan menggunakan format HTML. Beberapa orang merasa keberatan jika dikatakan HTML adalah sebuah bahasa pemrograman karena struktur yang dimilikinya dianggap terlalu sederhana, kode-kode dibaca oleh penjelajah baris per baris, dari atas ke bawah. HTML juga tidak memiliki ‘looping’ seperti bahasa pemrograman lain.

Pada HTML dipergunakan *hypertext* link atau hubungan antara teks dan dokumen lain. Dengan demikian pembaca dokumen bisa melompat dari satu dokumen ke dokumen yang lain dengan mudah.

2.5.9. MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya

Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat open source (tidak berbayar). MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP.

2.6. Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). UML merupakan standar yang relatif terbuka yang dikontrol oleh Object Management Group (OMG).

2.6.1. Use Case Diagram

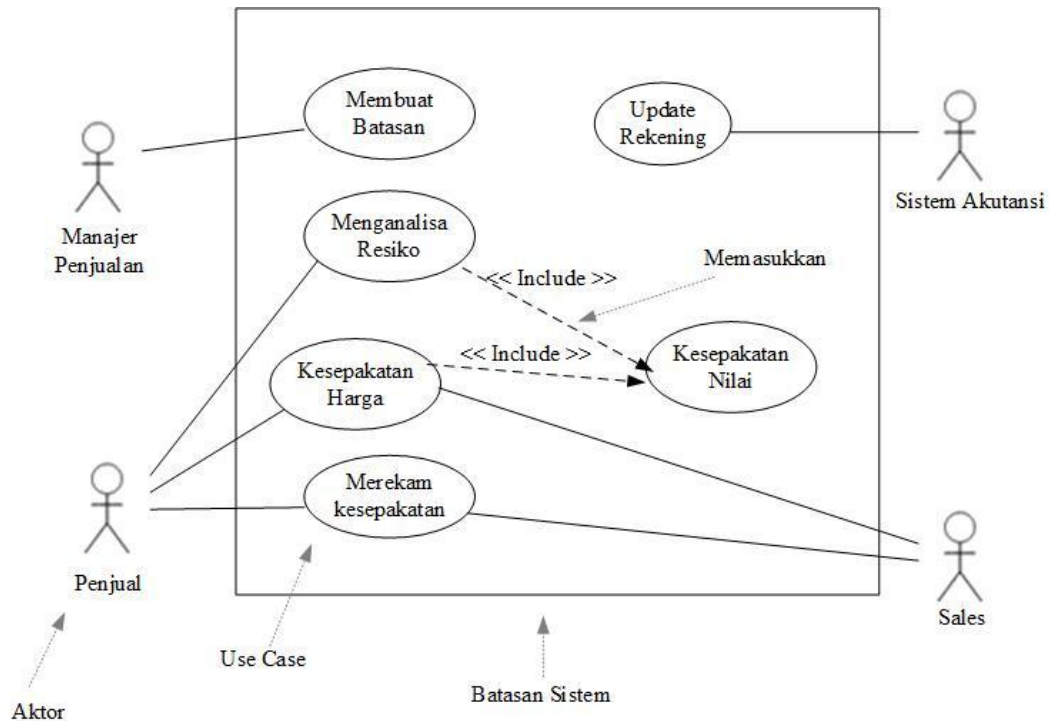
Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Sebuah *use case* adalah serangkaian skenario yang dikemas menjadi satu oleh tujuan pengguna umum. Dalam bahasa *use case*, para pengguna disebut sebagai aktor. Aktor merupakan seseorang yang berperan sebagai pengguna sistem. Aktor tidak harus manusia. Jika sebuah sistem melakukan layanan untuk sebuah sistem komputer lain, sistem lain tersebut merupakan aktor.

Use case diagram menampilkan aktor, *use case*, dan hubungan antara mereka :

- a. Aktor mana yang menggunakan *use case* mana.

b. *Use case* mana yang memasukkan *use case* lain.

Berikut ini adalah contoh dari diagram *use case* :

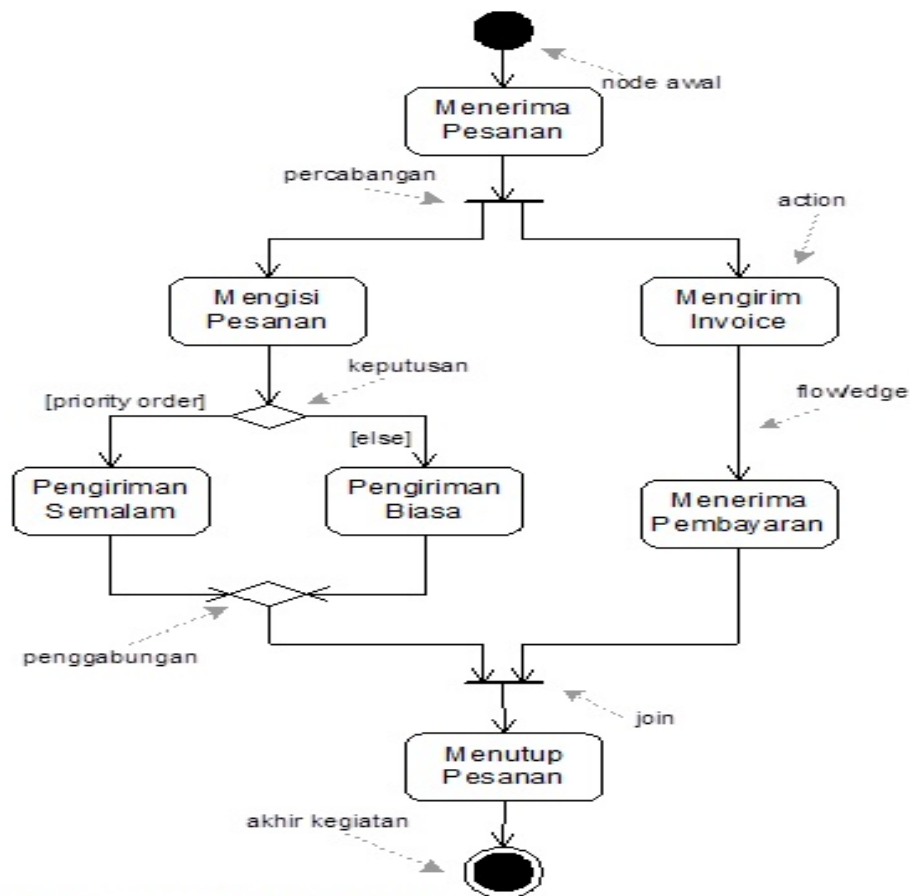


Gambar 2.3 Contoh diagram Use Case

[Cockburnm use case] menjelaskan sebuah skema tingkatan use case. Inti use case berada pada tingkat sea level. Use case sea level khususnya mewakili sebuah interaksi diskrit antara aktor utama dan sistem. Use case yang ada disana hanya karena mereka dimasukkan oleh use case sea level adalah fish level. Lebih tinggi, use case kite level menampilkan bagaimana use case sea level sesuai dengan interaksi bisnis yang lebih luas. Use case kite level biasanya merupakan use case bisnis, sedangkan se case sea dan fish level merupakan use case sistem. Kebanyakan use case Anda berada pada tingkatan sea level.

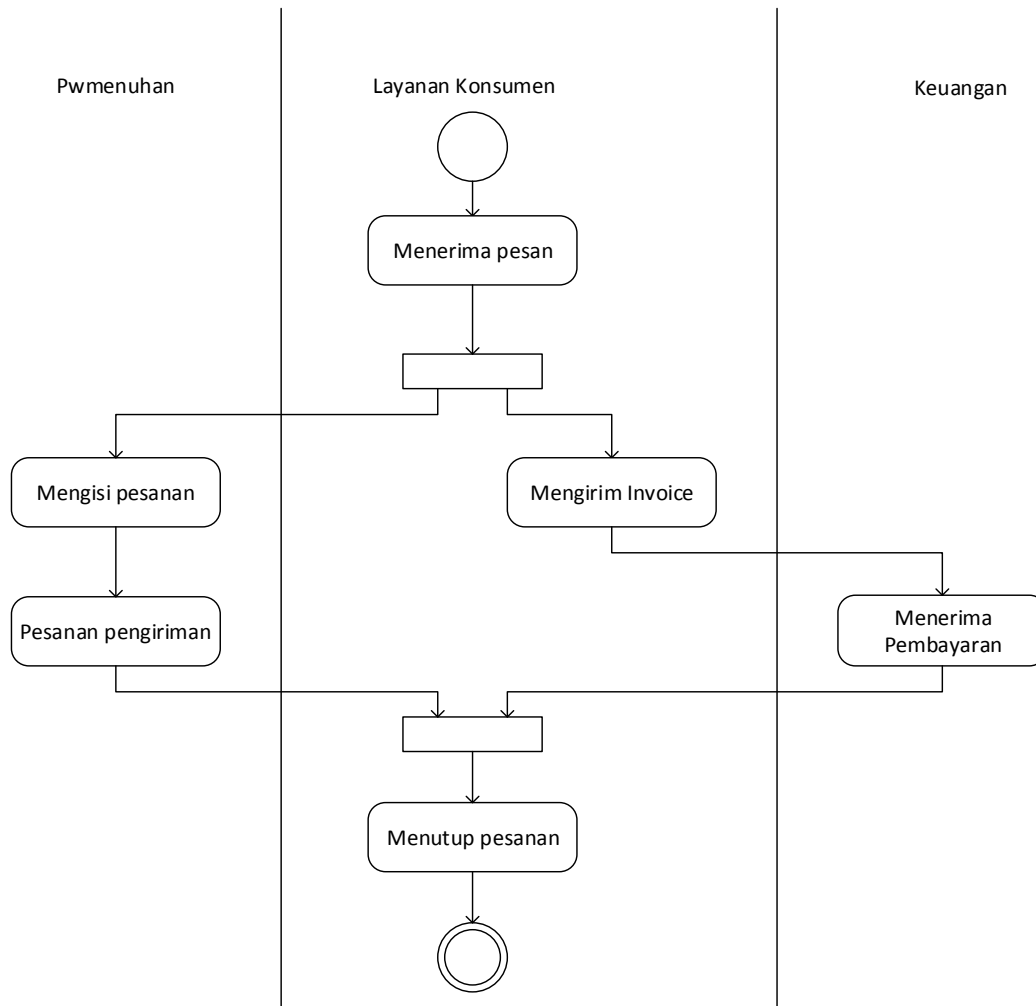
2.6.2. Activity Diagram

Diagram *activity* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja.



Gambar 2.4 Contoh Diagram Aktivitas

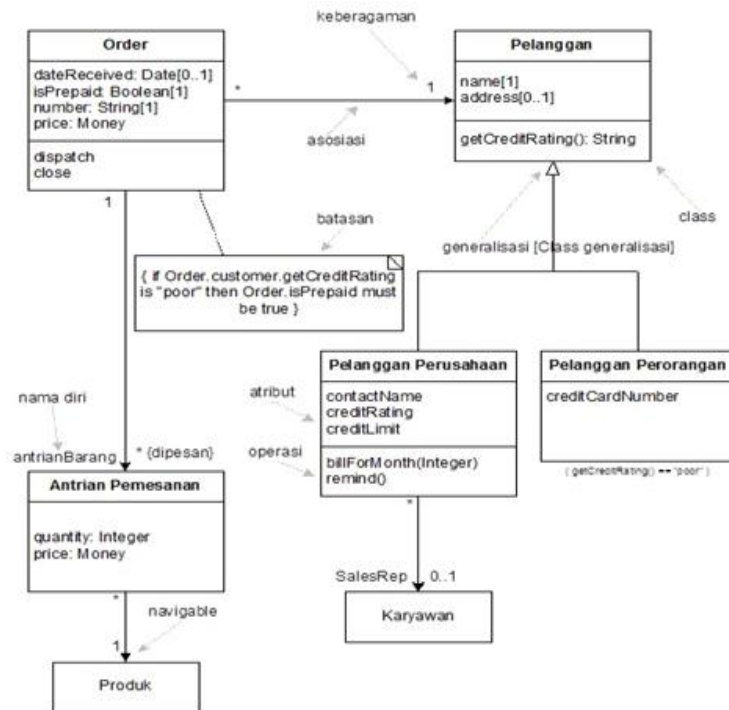
Activity diagram memberi tahu anda tentang apa yang terjadi, tetapi diagram ini tidak memberi tahu anda tentang siapa yang melakukan apa. Jika anda ingin menunjukkan siapa melakukan apa, anda dapat memisahkan *activity* diagram ke dalam partisi-partisi yang menampilkan *action* mana yang dilakukan oleh sebuah *class* atau organisasi. Pembuatan partisi pada Gambar 2.5 merupakan pembuatan partisi sederhana satu dimensi. Model ini sering disebut sebagai *swim lanes*.



Gambar 2.5 Partisi Pada Activity Diagram

2.6.3. Diagram Kelas

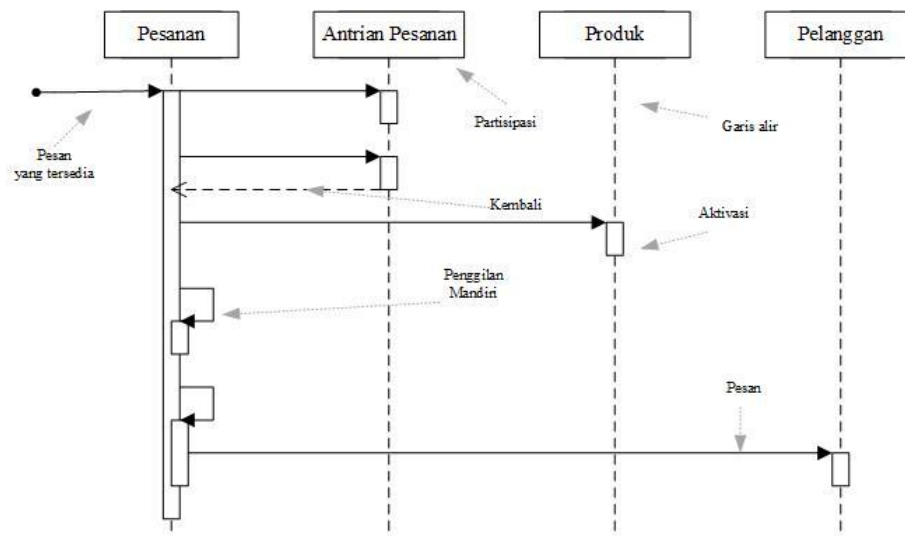
Diagram *class* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram *class* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat pada sebuah hubungan-hubungan objek. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*. Didalam *class* dibagi menjadi tiga bagian. Nama *class*, atributnya, dan operasinya.



Gambar 2.6 Contoh Diagram Class

2.6.4. Sequence Diagram

Diagram *sequence* secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek di dalam *use case*. Diagram *sequence* menunjukkan interaksi dengan menampilkan setiap partisipan dengan garis alur secara vertikal dan pengurutan pesan dari atas ke bawah.



Gambar 2.7 Contoh Diagram Sequence

2.6.4.1. Pengujian Aplikasi

Pengujian bertujuan untuk mencari kesalahan. Pengujian yang baik adalah pengujian yang memiliki kemungkinan besar dalam menentukan kesalahan. Berikut adalah penjelasan mengenai pengujian *white box* dan *black box*.

2.6.4.2. White Box Testing

Pengujian *white box* didasarkan pada pemeriksaan yang teliti terhadap detail prosedural. Jalur logis diseluruh perangkat lunak dan kerja sama antar komponen diuji dengan menguji serangkaian kondisi dan atau *loop* spesifik. Pengujian *white box* akan membawa pada kebenaran program, yang perlu dilakukan adalah mendefinisikan semua jalur logis, mengembangkan *test case* untuk menguji dan mengevaluasi hasilnya. *Test case* menjamin bahwa semua jalur independen didalam modul telah dieksekusi sedikitnya satu kali, melaksanakan semua keputusan logis pada sisi benar dan yang salah, melaksanakan semua *loop* pada batas dan dalam batas-batas operasional mereka, dan melakukan struktur data internal.

2.6.4.3. Black Box Testing

Pengujian *black box* disebut juga dengan pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black box* bukan teknik alternatif untuk *white box*. Sebaliknya, *black box* merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap oleh metode *white box*. Pengujian *black box* ada beberapa kategori untuk menemukan kesalahan, yaitu:

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Kesalahan perilaku atau kinerja.
5. Kesalahan inisialisasi dan penghentian.

Tidak seperti pengujian *white box*, yang dilakukan pada awal proses pengujian, pengujian *black box* dilakukan pada tahap pengujian selanjutnya. Pengujian *black box* dirancang untuk menjawab pertanyaan berikut:

1. Bagaimana validitas fungsional diuji?
2. Bagaimana perilaku dan kinerja sistem diuji?
3. Kelas-kelas masukan apakah yang akan membentuk test case yang baik?
4. Apakah sistem sangat sensitif terhadap nilai masukan tertentu?
5. Bagaimana batas-batas kelas data diisolasi?
6. Beberapa kecepatan dan volume data yang dapat ditolerir oleh sistem?
7. Apakah pengaruh kombinasi spesifik data pada operasi sistem?

2.6.4.4. Pengujian Beta

Pengujian beta merupakan pengujian objektif yang dilakukan langsung ke lapangan dengan menggunakan kuesioner mengenai tanggapan pengguna sibenar terhadap aplikasi yang telah dibangun. Pengujian beta terdiri dari kuesioner.

2.6.4.5. Skala Likert

Skala data yang digunakan untuk pengukuran variabel independen adalah skala likert. Skala likert adalah skala yang digunakan untuk mengukur sikap dan pendapat seseorang tentang fenomena sosial. Data yang telah terkumpul melalui angket, kemudian penulis olah kedalam bentuk kuantitatif, yaitu dengan cara menetapkan skor jawaban dari pertanyaan yang telah dijawab oleh responden, dimana pemberian skor tersebut didasarkan pada ketentuan Sugiyono menyatakan bahwa variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya.

