BAB II

LANDASAN TEORI

2.1 Perguruan Tinggi

Pendidikan tinggi terdiri dari (1) pendidikan akademik yang memiliki fokus dalam penguasaan ilmu pengetahuan dan (2) pendidikan vokasi yang menitikberatkan pada persiapan lulusan untuk mengaplikasikan keahliannya. Institusi Pendidikan Tinggi yang menawarkan pendidikan akademik dan vokasi dapat dibedakan berdasarkan jenjang dan program studi yang ditawarkan seperti akademi, politeknik, sekolah tinggi, institut dan universitas.

Akademi adalah perguruan tinggi yang menyelenggarakan program pendidikan vokasi dalam satu cabang atau sebagian cabang ilmu pengetahuan, teknologi atau kesenian tertentu. Sedangkan Politeknik adalah perguruan tinggi yang menyelenggarakan program pendidikan profesional dalam sejumlah bidang pengetahuan khusus. Kedua bentuk pendidikan tinggi ini menyediakan pendidikan pada level diploma. Contoh pendidikan tinggi seperti ini adalah Akademi Bahasa dan Politeknik Pertanian. Sekolah Tinggi adalah perguruan tinggi yang menyelenggarakan pendidikan vokasi dan akademik dalam lingkup satu disiplin ilmu pengetahuan, teknologi atau kesenian tertentu. Oleh karena itu, sekolah Tinggi ini menawarkan pendidikan baik pada level diploma maupun sarjana. Namun, ketika sebuah sekolah tinggi memenuhi persyaratan mereka dapat menawarkan pendidikan tingkat lanjut setelah level sarjana. Sekolah Tinggi ilmu Komputer merupakan salah satu contoh dari jenis pendidikan tinggi ini. Institut dan Universitas adalah institusi perguruan tinggi yang menyediakan pendidikan tinggi yang mengarah kepada level sarjana. Institut menawarkan pendidikan

akademik dan/atau vokasi dalam kelompok disiplin ilmu pengetahuan, teknologi dan/atau seni tertentu. Disisi lain, Universitas menawarkan pendidikan akademik dan/atau vokasi dalam berbagai kelompok disiplin ilmu pengetahuan, teknologi dan/atau seni. Institusi pendidikan tinggi ini dapat juga melayani pendidikan pada level profesional. Institut Seni adalah salah satu contohnya. (DIKTI, 2011)

2.1.1 Jenjang Pendidikan dan Syarat Belajar

Institusi pendidikan tinggi menawarkan berbagai jenjang pendidikan baik berupa pendidikan akademis maupun pendidikan vokasi. Perguruan tinggi yang memberikan pendidikan akademis dapat menawarkan jenjang pendidikan Sarjana, Program Profesi, Magister (S2), Program Spesialis (SP) dan Program Doktoral. Sedangkan pendidikan vokasi menawarkan program Diploma I, II, II dan IV. Untuk menyelesaikan pendidikan Sarjana (S1), seorang mahasiswa diwajibkan untuk mengambil 144-160 Satuan Kredit Semester (SKS) yang diambil selama delapan sampai dua belas semester. Pada jenjang S2 atau program Pasca Sarjana, seorang mahasiswa harus menyelesaikan 39 sampai 50 SKS selama kurun waktu empat sampai sepuluh semester dan 79 samapi 88 SKS harus diselesaikan dalam jangka waktu delapan samapi empat belas semester bagi program doktoral.

2.1.2 Metode Pembelajaran dan Jadwal Akademik

Pendidikan tinggi dapat diterapkan dalam beberapa bentuk: reguler atau tatap muka dan pendidikan jarak jauh. Pendidikan reguler diterapkan dengan menggunakan komunikasi langsung diantara dosen dan mahasiswa, sedangkan pendidikan jarak jauh dilaksanakan dengan menggunakan berbagai jenis media komunikasi seperti surat menyurat, radio, audio/video, televisi, dan jaringan

computer. Baik pendidikan reguler maupun pendidikan jarak jauh memulai aktivitas akademis atau jadwal akademikpada bulan September setiap tahunnya. Satu tahun akademik terbagi atas minimal dua semester yang terdiri dari setidak-tidaknya 16 minggu. Institusi pendidikan tinggi juga dapat melangsungkan semester pendek diantara dua semester reguler. Penerimaan mahasiswa pada perguruan tinggi didasarkan atas beberapa persyaratan dan prosedur serta objek penyeleksian yang tidak diskriminatif. Hal tersebut diatur oleh Senat masing-masing institusi pendidikan tinggi. Penerimaan mahasiswa merupakan tanggung jawab dari masing-masing perguruan tinggi. Calon mahasiswa D1,D2,D3,D4 dan S1 harus menamatkan pendidikan menengah atas atau yang sederajat dan lulus pada ujian masuk masing-masing perguruan tinggi. Kandidat mahasiswa S2 harus memiliki ijazah Sarjana (S1) atau yang sederajat dan lulus ujian seleksi masuk perguruan tinggi. Untuk S3, Mahasiswa harus memiliki Ijazah S2 atau yang sederajat dan lulus seleksi masuk.

2.1.3 Jaringan kerja

Direktorat Jenderal Pendidikan Tinggi menjalankan fungsi koordinasi terhadap perguruan tinggi di Indonesia, baik negeri maupun swasta. Jaringan kerja Ditjen Dikti secara garis besar terbagi atas dua yaitu terhadap perguruan tinggi negeri dan terhadap perguruan tinggi swasta. Saat ini Ditjen Dikti melakukan pengawasan langsung terhadap 88 perguruan tinggi negeri yang tersebar di seluruh provinsi di Indonesia. Adapun koordinasi dengan perguruan tinggi swasta dilakukan Ditjen Dikti melalui Kopertis (Koordinasi Perguruan Tinggi Swasta). Kopertis adalah unit pelaksana teknis Ditjen Dikti yang berada di 12 wilayah, yaitu:

1) Kopertis Wilayah I di Medan yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Sumatera Utara dan Nanggroe Aceh Darusalam.
2) Kopertis Wilayah II di Palembang yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Sumatera Selatan, Bengkulu, Lampung, Bangka Belitung.
3) Kopertis Wilayah III di Jakarta yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi DKI Jakarta.
4) Kopertis Wilayah IV di Bandung yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Jawa Barat dan Banten.
5) Kopertis Wilayah V di Yogyakarta yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi D.I. Jogjakarta.
6) Kopertis Wilayah VI di Semarang yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Jawa Tengah.
7) Kopertis Wilayah VII di Surabaya yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Jawa Timur dan Madura.
8) Kopertis Wilayah VIII di Denpasar yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Bali, NTB, dan NTT.
9) Kopertis Wilayah IX di Makassar yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Sulawesi Selatan, Sulawesi Tenggara, Sulawesi Tengah, Sulawesi Utara, Gorontalo, Sulawesi Barat.
10) Kopertis Wilayah X di Padang yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Sumatera Barat, Jambi, Riau, Kepulauan Riau.

11) Kopertis Wilayah XI di Banjarmasin yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Kalimatan Selatan, Kalimantan Timur, Kalimantan Tengah, Kalimantan Barat.
12) Kopertis Wilayah XII di Ambon yang mengkoordinasikan Perguruan Tinggi Swasta (PTS) di Provinsi Maluku, Maluku Utara, Papua, Papua Barat.

2.2 *Platform Google Android*

2.2.1 The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (Harahap, 2011). Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaannya dengan *Java Virtual Machine* (JVM) seperti Java ME (*Java mobile Edition*), tetapi Android menggunakan *Virtual Machine* sendiri yang menurut saya dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat mobile.

Dalvik Virtual Machine (DVM) adalah ”register bases” sementara *Java Virtual Machine* (JVM) adalah ”*stack based*”, DVM didesain oleh Dan Bornsten dan beberapa *engineers* Google lainnya. DVM menggunakan kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C / C + sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel Linux yang digunakan dalam Android tersebut.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

2.2.2 Android SDK (Software Development Kit)

Android SDK adalah tools API *(Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java (Harahap, 2011). Android merupakan subset perangkat lunak untuk *smartphone* yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang *direlease* oleh Google.

Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberi Anda kesempatan untuk membuat aplikasi yang bukan merupakan aplikasi bawaan Smartphone.

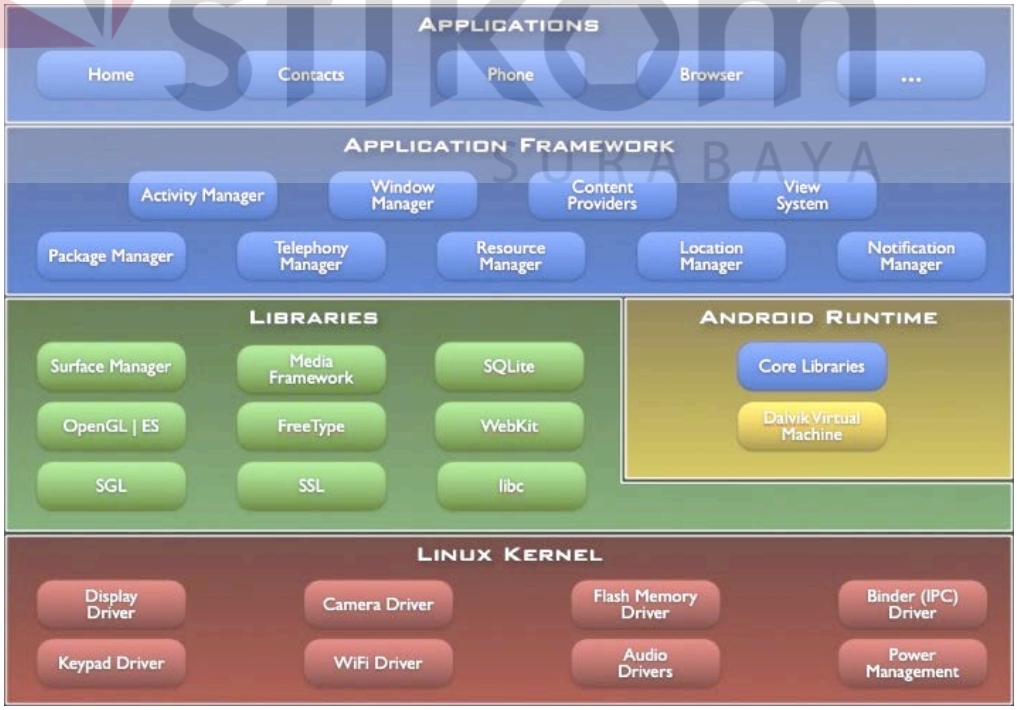
Beberapa fitur-fitur Android yang paling penting adalah:

1. Framework Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin Virtual Dalvik dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *engine open sourceWebKit*.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0 (Opsional akselerasi hardware).

5. SQLite untuk menyimpan data.
6. Media *Support* yang mendukung audio, video, dan gambar.
7. Bluetooth, EDGE, dan WiFi (tergantung hardware).
8. Kamera, GPS, kompas, dan accelerometer (tergantung hardware)
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tool* untuk debugging, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

2.2.3 Arsitektur Android

Arsistektur sistem terdiri atas 5 layer, pemisahan layer bertujuan untuk memberikan abstraksi sehingga memudahkan pengembangan aplikasi. Layer-layer tersebut adalah layer aplikasi, layer *framework* aplikasi, *layer libraries*, *layer runtime, dan layer kernel*. Gambar 6 memberikan gambaran umum komponen-komponen dalam arsitektur sistem operasi Android.



Gambar 2.1. Arsitektur Android

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

1. *Applications* dan *Widgets*

   Applications dan *Widgets* ini adalah layar dimana berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut, di layer inilah terdapat seperti aplikasi inti termasuk klien email, program sms, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman java.

2. Aplikasi *Frameworks*

   Android adalah ”*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada penembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resource*, menjalankan *servicebackground*, mengatur alarm, dan menambahkan tambahan seperti status notificaations, dan masih banyak lagi. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh apikasi yang berkategori inti. Arsitekturaplikasi dirancang supaya kita dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*).

3. Libraries

   Libraries ini adalah layer dimana *feature-feature* Android berada, biasanya para pembuat aplikasi kebanyakan mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas kernel, layer ini meliputi berbagai library C / C++ inti seperti Libc dan SSL.

4. Android *Runtime*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

5. Linux Kernel

   Linux kernel adalah layer dimana inti dari operating sistem dari Android itu sendiri, berisi file-file *system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operating Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel *release* 2.6.

2.3 Google Maps

Google Maps adalah sebuah layanan gratis peta digital dari Google berbasis web yang dapat digunakan dan ditempatkan pada website tertentu dengan menggunakkan Google Maps API (Google Inc, 2011).

Google Maps sendiri mempunyai fitur-fitur antara lain navigasi peta dengan *dragging mouse*, *zoom-in*dan*zoom-ou t*untuk menunjukkan informasi peta secara detil memberi penanda pada peta dan memberi informasi tambahan. Mode viewing pada Google Maps berupa "*Map*" (peta topografi dan jalan), "*satelite*" (peta berupa foto satelit dan foto resolusi tinggi dari udara), "*Hybrid*" (peta berupa foto satelit dan peta jalan berada diatasnya) dan "*Street View*", fasilitas ini secara resmi diperkenalkan oleh Google pada Mei 2007.

Seperti layanan aplikasi Google lainnya. Google Maps dibangun dengan menggunakan javascript, seperti ketika *user* menggeser pada peta, sepetak peta akan didownload dari *server* dan ditampilkan pada *user* tanpa harus *refresh* seluruh halaman web. Sebuah lokasi yang ditunjukkan dengan sebuah pin sebenarnya adalah berupa file PNG transparan yang diletakkan diatas peta. Teknik

yang digunakan untuk memberikan interaktifitas yang tinggi dengan cara melakukan *request* secara *asynchronous* dengan Javascript dan XML yang juga dikenal dengan AJAX.

2.3.1 Google Maps API

Google telah membuat Google Maps API untuk memfasilitasi para *developer* untuk mengintegrasikan Google Maps pada websitenya. Ini merupakan layanan gratis yang sementara tidak mengandung iklan, tetapi Google menyatakan pada perjanjian menggunakan layanan bahwa mereka berhak untuk menampilkan iklan dimasa yang akan datang (Google Inc, 2011). Dengan menggunakan Google Maps API kita dapat menampilkan seluruh fasilitas Google Maps pada *website* kita. Dimulai dengan membuat API key (API *Key* ini berfungsi sebagai kunci akses untuk website kita) dan kita sudah dapat menggunakan fungsi-fungsiyang ada pada Google Maps API untuk *website* kita.

2.3.2 Google Maps API pada Android

Location Based Service (LBS) atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. Dua unsur utama LBS adalah :

1. *Location Manager (API Maps)*

   Meneyediakan *tool/ source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan, memanipulasi *maps*/ peta beserta *feature-feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada *com.google.android.maps.*

2. *Location Providers (API Locations)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/ perangkat. *API Location* berhubungan dengan data GPS dan data lokasi *real-time*. Dengan *Location Providers* kita dapat menentukan lokasi kita saat ini, *Track* gerakan/ perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

Android memberikan akses aplikasi ke layanan lokasi yang didukung oleh perangkat melalui kelas-kelas dalam *package android.location*. komponen utama dari kerangka lokasi adalah sistem layanan *LocationManager*, yang menangani akses ke layanan lokasi. *LocationManager* tidak secara langsung dibuat objeknya, akan tetapi digunakan permintaan ke sistem dengan memanggil *getSystemService (Context.LOCATION_SERRVICE)* sehingga akan didapatkan *handler* untuk objek *LocationManager*. Ketika telah didapatkan objek *LocationManager* maka dapat melakukan query ke semua daftar *LocationProviders* yang dikenal oleh *LocationManager*sebagai lokasi terakhir.

Ketika bekerja dengan *emulator* maka fungsi untuk lokasi tidak dapat digunakan karena tidak tersedia GPS nyata pada *emulator* sehingga diperlukan data lokasi tiruan menggunakan *ddms* pada eclipse maupun melalui adb. Melalui *ddms* dapat dilakukan secara manual pengiriman koordinat bujur/ lintang ke perangkat, menggunakan file GPX menjelaskan rute untuk pemutaran ke perangkat, menggunakan file KML menjelaskan letak individu untuk pemutaran *sequencing* ke perangkat.

Geocoding memungkinkan menerjemahkan antara alamat jalan dengan bujur/ lintang koordinat peta. Hal ini dapat memberikan konteks dikenali untuk lokasi dan koordinat yang digunakan dalam layanan berbasis lokasi dan peta

berbasis *Activity*. Pencarian *geocoding*dilakukan di server, sehingga aplikasi akan meminta untuk memasukkan sebuah izin penggunaan internet. Kelas *geocoder* menyediakan akses untuk dua fungsi *geocoding* :

1. Forward Geocoding : mencari lintang dan bujur alamat.
2. Reverse Geocoding : Mencari alamat dan jalan untuk sesuai lintang dan bujur.

2.4 UML (*Unified Modeling Language*)

2.4.1 Pengertian UML

UML (Unified Modelling Language) adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan *artifacts* dari sistem software, untuk memodelkan bisnis, dan sistem nonsoftware lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks **(Suhender & Gunadi, 2002)** .

2.4.2 *Artifact* UML

Artifact adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software. Artifact dapat berupa model, deskripsi, atau software. Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

A. *Use-case diagram*

Use-case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

B. *Class diagram*

Class diagram membantu untuk visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain (dalam *logical view*) dari suatu sistem.

C. Behavior diagram

i. Statechart diagram

Statechart diagram memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas kepada yang lainnya dan aksi yang menyebabkan perubahan satu *state* atau aktivitas. *Statechart diagram* khususnya digunakan untuk memodelkan taraf-taraf diskrit dari sebuah siklus hidup objek sedangkan *activity diagram* paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses.

ii. Activity diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas kedalam keadaan sesaat (*state*). *Activity diagram* juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use-case* berinteraksi.

iii. Interaction diagram:

1. Sequence diagram

Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use-case*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use-case*. Tipe diagram ini sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti.

2. Collaboration diagram

Collaboration diagram melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam satu *collaboration diagram* terdapat beberapa *object*, *link*, dan *message*. *Collaboration diagram* digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

D. Implementation diagram:

i. Component diagram

Componen diagram menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *sourcecode*, *binarycode* dan komponen tereksekusi (*executable components*).

ii. Deployment diagram

Setiap model hanya memiliki satu *deployment diagram*. Diagram ini memperlihatkan pemetaan *software* kepada *hardware*.

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Dibuatnya berbagai jenis diagram diatas karena:

a) Setiap sistem yang kompleks selalu paling baik jika didekati melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencakupi untuk melihat sistem yang besar dan kompleks.
b) Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa.

Diagram-diagram tersebut dibuat agar model yang dibuat semakin mendekati realitas.

2.4.3 Faktor yang Mendorong Dibuatnya UML

1) Pentingnya Model
   a) Membangun model untuk suatu sistem software (terlebih software untuk suatu organisasi bisnis) sangat bergantung pada kosntruksinya atau kemudahan dalam memperbaikiinya. Oleh karena itu, membuat model sangatlah penting sebagaimana pentingnya kita memiliki cetak biru untuk suatu bangunan yang besar.

b) Model yang bagus sangat penting untuk menghasilkan komunikasi yang baik antar anggota tim dan untuk meyakinkan sempurnanya arsitektur sistem yang dibangun.

c) Jika kita membangun suatu model dari suatu sistem yang kompleks, tidak mungkin kita dapat memahaminya secara keseluruhan. Dengan meningkatnya komplektisitas sistem, visualisasi dan pemodelan menjadi sangat penting. UML dibuat untuk merespon kebutuhan tersebut.

2) Kecenderungan Dunia Industri terhadap *Software*

a) Sebagai suatu nilai yang strategis bagi pasar software adalah dengan meningkatnya kebutuhan dunia industri untuk memiliki teknik otomatisasi dengan software. Oleh karena itu, penting sekali adanya teknik rekayasa software yang dapat meningkatkan kualitas dan mengurangi biaya dan waktu. Termasuk dalam teknik rekayasa software adalah teknik *visual programming*, juga pembuatan *framework* dan pola.

b) Komplektisitas dunia industri semakin meningkat karena bertambah luasnya runag lingkup dan tahapan proses. Satu motivasi kunci bagi para pembangun UML adalah untuk membuat suatu himpunan semantik dan notasi yang mampu menangani kerumitan arsitektural dalam semua ruang lingkup.

3) Terjadinya Konvergensi Metode dan *Tool* Pemodelan

Sebelum adanya UML, terdapat ketidakjelasan bahasa pemodelan apa yang paling unggul. Para user harus memilih diantara bahasa pemodelan dan *tool* (software) pemodelan yang banyak dan mirip UML dibuat untuk membuat integrasi baru dalam bahasa pemodelan antar *tool* dan “proses”.

2.4.4 Tujuan UML

Tujuan utama 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.

2.5 GPS

2.5.1 Pengertian GPS (*Global Positioning System*)

GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh di dunia saat ini (Cellular Telecommunications Industry Association, 2011). GPS menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah GPS *receiver* menerima informasi dari tiga atau lebih satelit tersebut untuk menentukan posisi. GPS *receiver* harus berada dalam *line-of-sight* (LoS) terhadap keempat satelit tersebut untuk menentukan posisi, sehingga GPS hanya ideal untuk digunakan dalam *outdoor positioning*.

2.5.2 Arsitektur GPS

Arsitektur GPS terdiri atas 3 segmen, yaitu: *space segment, control segment, dan user segment*. *Space segment* adalah berupa 27 satelit-satelit yang terus mengorbit bumi. Sebanyak 24 satelit digunakan untuk operasional

sedangkan 3 sisanya digunakan sebagai cadangan. Dua puluh empat satelit tersebut dibagi atas kumpulan 4 satelit yang mengorbit pada 6 jalur lintasan. Orbit lintasan ini telah diatur sedemikian rupa sehingga setiap titik di bumi pasti tercakup dalam LoS dari sedikitnya 6 satelit. Deskripsi *space segment* di GPS dapat dilihat pada Gambar 2.1. Setiap satelit GPS mengorbit pada ketinggian 20.200 kilometer dari permukaan bumi dengan periode evolusi 12 jam (El-Rabbany, 2002). Sejak September 2007 kini GPS memiliki 31 satelit dimana satelit tambahan digunakan untuk meningkatkan ketelitian GPS *receiver*.

Control segment berupa stasiun di bumi yang memonitor satelit-satelit GPS. Stasiun ini mengontak setiap satelit GPS secara berkala untuk memberikan *update* navigasi. *Update* ini berupa sinkronisasi jam atomik satelit dengan satelit lainnya dan mengkoreksi lintasan orbit setiap satelit.

Gambar 2.2. Space Segment pada GPS

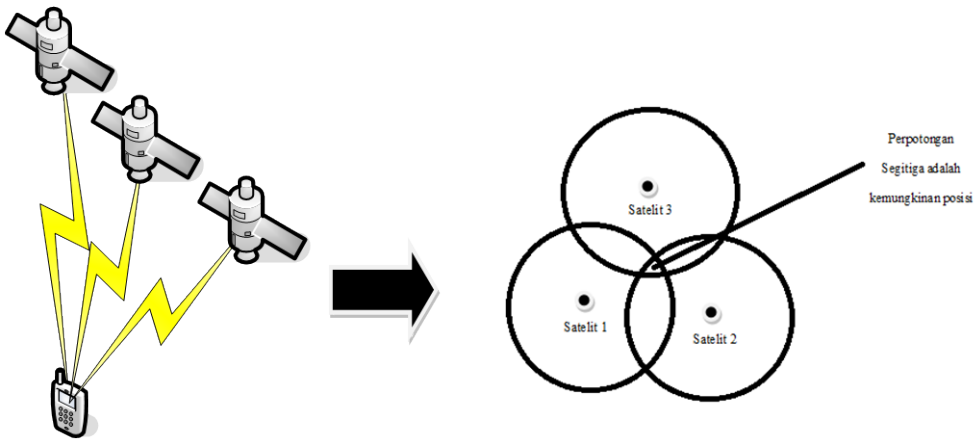
User segment adalah berupa *GPS receiver*. Secara umum, *GPS receiver* terdiri sebuah antena yang dapat menangkap frekuensi yang ditransmisikan satelit

GPS, sebuah prosesor, dan sebuah jam yang sangat stabil. *GPS receiver* memiliki atribut *channel* yaitu jumlah satelit yang dapat dimonitornya dalam suatu waktu (sekarang umumnya jumlah channel berkisar antara 12 -20). Kebanyakan *GPS receiver* dapat meneruskan datanya ke perangkat lain melalui koneksi serial, USB atau Bluetooth menggunakan protokol NMEA.

2.5.3 Cara Penentuan Lokasi pada GPS

Satelit GPS mengorbit bumi dua kali dalam sehari dengan lintasan yang sangat presisi dan mentrasmisikan sinyal informasi secara kontinu ke bumi. *GPS receiver* memanfaatkan informasi ini dengan berperan sebagai sebuah alat pengukur yang menghitung jarak antara antenna *receiver* dengan berbagai satelit GPS. Kemudian *GPS receiver* mendeduksi posisi melalui proses trilaterasi dengan mencari perpotongan tiap vector satelit-satelit tersebut. Jarak antara antenna *receiver* dan satelit diukur dengan membandingkan waktu yang terdapat pada sinyal dengan waktu ketika sinyal diterima.

Sebuah GPS *receiver* setidaknya harus menangkap sinyal dari 3 buah satelit untuk menghitung posisi 2 dimensi (latitude dan longitude) dan pergerakannya. Dengan 4 buah satelit atau lebih, *receiver* dapat menghitung posisi 3 dimensi (latitude, longitude, dan altitude).



Gambar 2.3. Trilaterasi dalam GPS

Gambar 3. memberikan gambaran sederhana yang memodelkan proses trilaterasi dalam GPS untuk menentukan lokasi. Lingkaran melambangkan ruang wilayah cakupan sebuah satelit dan titik di tengahnya adalah satelit itu sendiri.

2.5.4 Kelebihan dan Kekurangan GPS

Teknologi GPS sangat fenomenal dalam bidang penentuan posisi karena mampu memberikan informasi mengenai posisi secara *real-time*dan*continue*, di mana saja dan kapan saja. Selain itu ada beberapa hal yang membuat GPS sangat baik dalam sistem pelacakan:

1. Tidak bergantung waktu dan cuaca,
2. Meliputi wilayah yang luas,
3. Tidak terpengaruh topografis,
4. Memberikan ketelitian akurasi yang cukup,
5. Pengunaan tidak dikenakan biaya, alias gratis.

Kekurangan GPS adalah mengharuskan *GPS receiver* berada dalam *Line-of-Sight* dengan setidaknya 3 buah satelit GPS untuk menentukan lokasi.

2.5.5 GPS Tracking

Istilah GPS *tracking* digunakan dalam konteks Tugas Akhir ini sebagai pengiriman informasi lokasi perangkat mobile saat pengguna melakukan *query* terhadap aplikasiyang kemudiandilanjutkandengan pengiriman informasi lokasi perangkat *mobile* per setiap periode waktu tertentuuntukdisimpan di *web server*. Berdasarkan lingkup wilayah pantauannya, system penelusuran dan pelacakan terbagi atas *wide-area* dan *local-area*. Sistem penelusuran dan pelacakan *wide-area* pada umumnya menggunakan *GPS receiver* dikarenakan wilayah pantauannya yang sangat luas. Penggunaan GPS juga memungkinkan pengguna dapat meminta informasi posisi setiap saat, namun GPS memiliki keterbatasan pada ruang tertutup dikarenakan factor *Line-of-Sight* tadi.

2.5.6 Akurasi GPS

GPS menyediakan posisi dengan ketepatan akurasi hingga 15 meter, yang berarti jika *GPS receiver* memberikan koordinat terhadap suatu lokasi tertentu, maka boleh diharapkan lokasi sebenarnya berada dalam radius 15 meter dari koordinat tersebut (El-Rabbany, 2002). Ketepatan GPS bergantung daripada lokasi *GPS receiver*-nya dan halangan terhadap sinyal satelit GPS.Meski secara umum, GPS menawarkan tingkat ketelitian 15 meter, namun akurasi ini dapat ditingkatkan dengan berbagai teknik, seperti *Assisted GPS* (A-GPS*), Differential GPS* (D-GPS), atau *Wide Area Augmentation System* (WAAS). Pada kenyataan di lapangan, akurasi dapat bervariasi sesuai keadaan.

2.5.7 *Assisted GPS* (A-GPS*)*

Pada Smartphone & Tablet generasi terbaru kita bisa menemukan fitur GPS (Global Positioning System), yaitu fitur untuk menentukan lokasi kita melalui koneksi satelit. Selain itu ada perangkat GPS original yang hanya berfungsi tunggal GPS saja. GPS sendiri memiliki beberapa perbedaan teknologi & cara kerja. Pada Smartphone & Tablet dikenal dua tipe GPS, yaitu GPS Satellite & Assisted-GPS (A-GPS). Secara sederhana, GPS satellite adalah GPS orisinal yang menggunakan controller & mekanisme GPS murni dan langsung terhubung satelit. Sedangkan A-GPS, selain satelit juga harus menggunakan teknologi GSM (atau menara BTS) untuk menentukan lokasi.

Assisted Global Positioning System (A-GPS) akan membantu perangkat Anda menemukan satelit. Saat menggunakan A-GPS, perangkat Anda akan menerima informasi satelit dari server data melalui jaringan seluler. Apabila perangkat Anda tidak menerima data tersebut, perangkat akan mencoba mendeteksi satelit lain. Dengan bantuan data tersebut, perangkat Anda akan mendeteksi satelit yang tepat.A-GPS mempercepat penandaan kalkulasi lokasi.A-GPS merupakan layanan jaringan, yang digunakan pada perangkat smartphone ataupun tablet yang mendukung A-GPS.A-GPS tersedia di seluruh negara dan tidak tergantung pada layanan operator terkait. A-GPS didesain agar perangkat dapat terhubung dengan satelit dengan lebih cepat dan lebih dapat diandalkan daripada menggunakan GPS tunggal. Sebagai contoh, penentuan posisi dapat lebih cepat diperoleh meskipun pada musim dingin, atau saat koneksi GPS dimatikan dalam waktu lama, atau apabila pengguna melakukan perjalanan ke negara yang berbeda.Dengan A-GPS, perkiraan waktu untuk membangun koneksi

GPS dapat ditingkatkan secara signifikan.Perangkat akan membutuhkan waktu beberapa menit untuk membangun koneksi GPS pada situasi dingin, A-GPS akan mengurangi waktu yang dibutuhkan sampai 10 detik.Sebagai tambahan, A-GPS mengurangi waktu yang dibutuhkan oleh perangkat untuk menemukan posisinya, yang dikenal sebagai 'Time To First Fix' (TTFF), untuk sebagian besar lokasi geografis di seluruh dunia.GPS memiliki beberapa keterbatasan yang juga dimiliki oleh A-GPS. Sebagai contoh, ketersediaan dan kualitas sinyal yang dapat dipengaruhi oleh halangan dari gedung-gedung yang tinggi, kondisi cuaca, dan lokasi pengguna. Apabila Anda berada di dalam ruangan, sebaiknya Anda keluar dan berada di halaman terbuka. Pastikan bahwa tangan Anda tidak menghalangi GPS receiver, yang berada pada bagian bawah perangkat Anda.

A-GPS menggunakan jaringan sesluler 3G dan 2G serta koneksi paket data GPRS dan EGPRS. Anda juga harus memiliki titik akses internet yang didefinisikan oleh perangkat Anda.Titik akses Wireless LAN (WLAN) tidak didukung apabila Anda sedang menggunakan A-GPS.Di mana saya dapat menggunakan A-GPS?A-GPS dapat digunakan di seluruh dunia, selama Anda memiliki akses ke jaringan seluler dan koneksi data. Anda juga dapat menggunakan layanan ini di luar jaringan lokal Anda(roaming). A-GPS tidak tergantung pada layanan operator yang spesifik.Berapa biaya yang dibutuhkan untuk menggunakan A-GPS?Proses mendownload data memerlukan transmisi melalui provider layanan jaringan Anda. Pada umumnya, untuk mendapatkan koneksi GPS dengan A-GPS memerlukan minimum 10kB data yang ditransfer.Seluruh biaya transmisi data akan dibebankan kepada pengguna.

2.6 Test Case

Test case merupakan suatu tes yang dilakukan dengan berdasar pada suatu inisialisasi, masukan, kondisi atau pun hasil yang telah ditentukan sebelumnya. Adapun kegunaan *test case* adalah sebagai berikut (Romeo, 2003):

1. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi produk. *Test case* yang digunakan untuk testing ini adalah *black box testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap desain. *Test case* yang digunakan untuk testing ini adalah *white box testing*.

2.6.1 Black Box Testing

Black box testing merupakan *testing* yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* juga disebut sebagai *behavioral testing*, *specification-based testing*, *input output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional *software*, yang berdasar pada spesifikasi kebutuhan *software*. Kategori *error* yang akan diketahui menggunakan *black box testing* adalah (Romeo, 2003):

1. Fungsi yang hilang atau tidak benar.
2. *Error* dari *interface*.
3. *Error* dari struktur data atau akses eksternal database.
4. *Error* dari kinerja atau tingkah laku sistem.
5. *Error* dari *inisialisasi* dan terminasi.

2.7 Angket

Angket atau disebut juga *questionnaire* adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon, sesuai dengan permintaan pengguna. Tujuan penyebaran angket adalah mencari informasi dari responden tanpa khawatir bila responden memberikan jawaban yang tidak sesuai dengan keyataan (Riduwan, 2005). Dalam penelitian ini, angket dibutuhkan untuk mengukur tingkat kelayakan penggunaan aplikasi.

Menurut Riduwan (2005), para ahli membedakan dua tipe skala pengukuran menurut gejala sosial yang diukur, yaitu:

1. Skala pengukuran untuk mengukur perilaku susila dan kepribadian, antara lain skala sikap, skala moral, tes karakter, dan skala partisipasi sosial.
2. Skala pengukuran untuk mengukur berbagai aspek budaya lain dan lingkungan sosial, antara lain skala mengukur status sosial ekonomi, lembaga swadaya masyarakat (sosial), kemasyarakatan, kondisi rumah tangga dan lain-lain.

Masih menurut Riduwan (2005), skala sikap dibagi menjadi lima bentuk, yaitu skala *Likert*, skala *Guttman*, skala *Defferensial Simantict*, *Rating Scale* dan skala *Thurstone*.

2.7.1 Skala Likert

Skala *Likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok tentang kejadian atau gejala sosial. Pengukuran sikap, pendapat dan persepsi seseorang harus melalui proses pengolahan data. Angket yang sebelumnya telah diisi kemudian direkapitulasi sehingga dapat dilakukan perhitungan skor.

Perhitungan skor penilaian untuk setiap pertanyaan (QS) didapatkan dari jumlah pengguna (PM) dikalikan dengan skala nilai (N). Jumlah skor tertinggi (STtot) didapatkan dari skala tertinggi (NT) dikalikan jumlah pertanyaan (Qtot) dikalikan total pengguna(Ptot). Sedangkan nilai persentase akhir (Pre) diperoleh dari jumlah skor hasil pengumpulan data (JSA) dibagi jumlah skor tertinggi (STtot) dikalikan 100%. Persamaan yang digunakan untuk melakukan perhitungan skor pada setiap pertanyaan dapat dilihat pada Persamaan 2.1. Persamaan 2.2 digunakan untuk menghitung jumlah skor tertinggi. Persamaan 2.3 menghasilkan nilai persentase yang akan digunakan dalam proses analisis.

$$QS(\mathrm{n}) = PM \times N \quad (2.1)$$

$$STtot = NT \times Qtot \times Ptot \quad (2.2)$$

$$Pre = \frac{JSA}{STtot} \times 100\% \quad (2.3)$$

dengan:

QS(*n*) = skor pertanyaan ke-*n*

PM = jumlah pengguna yang menjawab

N = skala nilai

STtot = total skor tertinggi

NT = skala nilai tertinggi

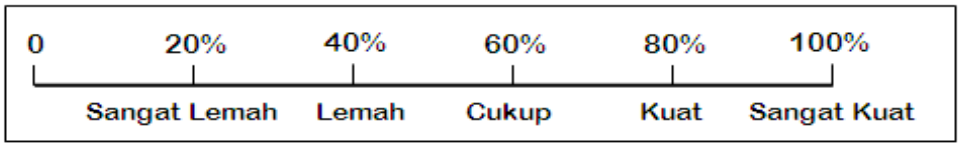
Qtot = total pertanyaan

Ptot = total pengguna

Pre = persentase akhir (%)

JSA = jumlah skor akhir

Analisis dilakukan dengan melihat persentase akhir dari proses perhitungan skor Nilai persentase kemudian dicocokkan dengan kriteria interpretasi skor, seperti yang terlihat pada Gambar 2.4.



Gambar 2.4 Kriteria Interpretasi Skor (Riduwan, 2005: 15)