BAB 2

TINJAUAN PUSTAKA

2.1 Profil CV. PROJECTRON CORPORATION

CV. PROJECTRON CORPORATION adalah suatu bentuk perusahaan Persekutuan Komanditer / CV (*Commanditaire Venootschaap*) yang bergerak dalam ruang lingkup bidang usaha perdagangan besar berdasarkan balas jasa (FEE) kontrak.

CV. PROJECTRON CORPORATION memiliki ruang lingkup bidang usaha yaitu, *Instrumens Maintenance*, Desain Elektronika dan Elektrikal Mekanikal, *IT Software Developer*, dan Perdagangan Umum dan Jasa.

2.2 Struktur Organisasi

Struktur organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur Organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi.



Gambar 2.1 Struktur Organisasi CV. PROJECTRON CORPORATION

2.3 Deskripsi Tugas

Deskripsi tugas digunakan untuk mengetahui tugas, wewenang tanggung jawab dari masing-masing bagian. Deskripsi tugas yang ada di CV. PROJECTRON CORPORATION dijelaskan pada tabel 2.1.

Tabel 2.1 Deskripsi Tugas

No	Jabatan	Deskripsi Tugas
1	Komanditer	Bertugas sebagai pengawas dari berjalannya suatu CV. PROJECTRON CORPORATION
2	Direktur	Bertindak sebagai perwakilan perusahaan dalam hubungannya dengan dunia luar perusahaan.
3	Manajer Operasional	Bertugas mengelola dan meningkatkan efektivitas dan efisiensi operasi perusahaan, serta survei lapangan.
4	Manajer Teknik dan Pemasaran	Bertugas untuk menentukan barang atau produk apa yang telah disepakati dengan perusahaan lain, dan menetapkan harga yang sesuai dengan kualitas dan kuantitas barang.
5	Manajer Keuangan dan ADM	Bertugas menetapkan kebutuhan keuangan lembaga sekarang dan akan datang dan rencana menutupinya (termasuk rencana jangka pendek, menengah, dan panjang), serta bertanggung jawab terhadap perencanaan, pelaksanaan, pengendalian, pelaporan kegiatan lembaga dalam aspek keuangan.

2.4 Logo CV. Projectron Corporation

Logo CV. Projectron Corporation dapat dilihat pada Gambar 2.2



Gambar 2.2 Logo CV. PROJECTRON CORPORATION

2.5 Definis Pengukuran

Pengertian Pengukuran menurut para ahli[5]:

1. Menurut Ign. Masidjo (1995: 14) pengukuran adalah suatu kegiatan menentukan kuantitas suatu objek melalui aturan-aturan tertentu sehingga kuantitas yang diperoleh benar-benar mewakili sifat dari suatu objek yang dimaksud.
2. Pengukuran bisa diartikan sebagai proses memasangkan fakta-fakta suatu objek dengan fakta-fakta satuan tertentu (Djaali & Pudji Muljono, 2007).
3. Menurut Endang Purwanti (2008:4) pengukuran dapat diartikan sebagai kegiatan atau upaya yang dilakukan untuk memberikan angka-angka pada suatu gejala atau peristiwa, atau benda, sehingga hasil pengukuran akan selalu berupa angka.
4. Pengukuran dapat diartikan dengan kegiatan untuk mengukur sesuatu. Pada hakekatnya, kegiatan ini adalah membandingkan sesuatu dengan atau sesuatu yang lain (Anas Sudiono, 2001).
5. Pengukuran adalah suatu proses atau kegiatan untuk menentukan kuntitas sesuatu (Zaenal Arifin, 2012).

2.6 Definis Ketinggian

Pengertian tinggi menurut para ahli adalah besaran yang menunjukkan kedudukan sesuatu terhadap poisis dasar. Menurut pengertian ini, posisi dasarlah menjadi titik acuan. Artinya, selalu ada titik nol yang menjadi acuan untuk menentukan batasan tinggi. Dalam bidang keilmuan ahli geografi, menggunakan istilah tinggi untuk mengukur kedudukan permukaan bumi yang tidak rata. Yang menjadi titik acuan atau titik nol pengukurannya adalah permukaan air laut bila dalam keadaan tenang.

2.7 Definis Kemiringan

Keadaan sudut yangg terbentuk oleh perpotongan antara bidang lapisan batuan dengan bidang datar yangg dihitung berdasarkan bidang vertikal.

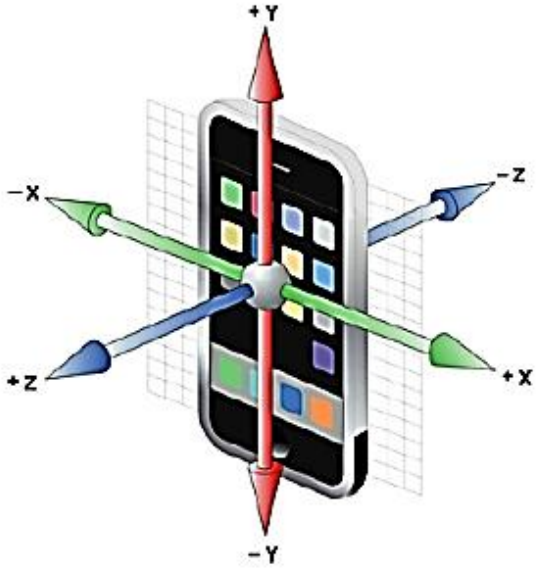
2.8 Definis Accelerometer

Accelerometer adalah sebuah sensor/tranduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. Pada penelitian ini, *Accelerometer* digunakan sebagai mengambil nilai sudut kemiringan dari suatu objek. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga bisa digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut juga percepatan (*acceleration*). Jika kecepatan semakin berkurang daripada kecepatan sebelumnya, disebut *deceleration*. Percepatan juga bergantung pada arah/orientasi karena merupakan penurunan kecepatan yang merupakan besaran vektor. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula.

Pada *smartphone*, *Accelerometer* merupakan sensor yang bisa membaca pergerakan sehingga dapat mengubah tampilan layar dari posisi *landscape* ke *portrait* atau sebaliknya dengan cukup memiringkan badan ponsel secara otomatis.

Pada *smartphone* berbasis *android* terdapat sensor manager yang berfungsi untuk mengaktifkan sensor *accelerometer* dalam mencari nilai koordinat x,y,z dengan memiringkan handphone[6].



Gambar 2.3 Sumbu *Accelerometer*[6]

2.8.1 Prinsip Kerja *Accelerometer*

Prinsip kerja dari tranduser ini berdasarkan hukum fisika bahwa apabila suatu konduktor digerakkan melalui suatu medan magnet, atau jika suatu medan magnet digerakkan melalui suatu konduktor, maka akan timbul suatu tegangan induksi pada konduktor tersebut. *Accelerometer* yang diletakan di permukaan bumi dapat mendeteksi percepatan 1g (ukuran gravitasi bumi) pada titik vertikalnya, untuk percepatan yang dikarenakan oleh pergerakan horizontal maka *accelerometer* akan mengukur percepatannya secara langsung ketika bergerak secara horizontal. Hal ini sesuai dengan tipe dan jenis sensor *accelerometer* yang digunakan karena setiap jenis sensor berbeda-beda sesuai dengan spesifikasi yang dikeluarkan oleh perusahaan pembuatnya. Saat ini hampir semua sensor/tranduser *accelerometer* sudah dalam bentuk digital (bukan dengan sistem mekanik) sehingga cara kerjanya hanya bedasarkan temperatur yang diolah secara digital dalam satu chip[6].

2.8.2 Tipe Accelerometer

a. *Capacitive*

   Lempengan metal pada sensor memproduksi sejumlah kapasitansi, perubahan kapasitansi akan mempengaruhi percepatan.

b. *Piezoelectric*

   Kristal *piezoelectric* yang terdapat pada *accelerometer* jenis ini mengeluarkan tegangan yang selanjutnya dikonversi menjadi percepatan.

c. *Piezoresistive*

   Lempengan yang secara resistan akan berubah sesuai dengan perubahan percepatan.

d. *Hall effect*

   Percepatan yang dirubah menjadi sinyal elektrik dengan cara mengukur setiap perubahan pergerakan yang terjadi pada daerah yang terinduksi magnet.

e. *Magnetoresistive*

   Perubahan percepatan diketahui berdasarkan resistivitas material karena adanya daerah yang terinduksi magnet.

f. *Heat Transfer*

   Percepatan dapat diketahui dari lokasi sebuah benda yang dipanaskan dan diukur ketika terjadi percepatan dengan sensor temperatur.

2.9 Handphone

Handphone disebut juga dengan telepon seluler (ponsel) atau telepon genggam (telgam) merupakan sebuah perangkat komunikasi elektronik yang memiliki kemampuan dasar sama dengan telepon konvensional saluran tetap, tetapi bersifat *portable* atau *mobile* (dapat dibawa kemana-mana) sehingga tidak perlu disambungkan dengan jaringan telepon kabel. Sampai saat ini di Indonesia sudah terdapat dua jenis jaringan nirkabel atau *wireless* untuk telepon selular, yakni GSM (*Global System for Mobile Telecommunications*) dan CDMA (*Code Division Multiple Acces*). Kegunaan ponsel sangatlah beragam, tidak hanya untuk melakukan panggilan saja, melainkan sekarang sudah berkembang bisa digunakan

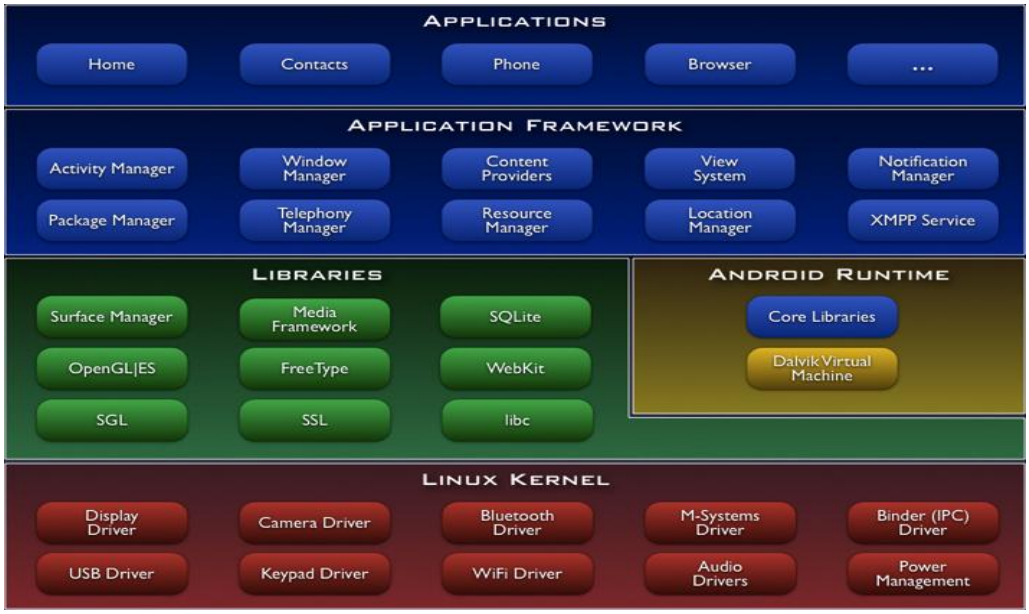
untuk melakukan pengiriman dan penerimaan pesan singkat atau SMS (*Short Message Service*). Semakin berkembang handphone semakin dilengkapi dengan berbagai kecanggihan fitur yang menarik dan membuat penggunanya bertambah pesat, yakni menangkap siaran radio dan televisi, pemutar video dan audio, kamera, game, dan sekarang semakin diperlengkap dengan jaringan internet[7].

2.10 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Pada penelitian ini, *Android* digunakan sebagai media pembangunan suatu aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. 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[8].

2.10.1 Arsitektur Android

Google menggambarkan *Android* seperti sebuah tumpukan software. Setiap lapisan dari tumpukan ini terdiri dari beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Dalam paket system operasi android tediri dari beberapa unsure seperti tampak pada gambar 2.4. Secara sederhana arsitektur *android* merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu Framework yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.4 Arsitektur *Android*[8]

Berikut adalah penjelasan mengenai masing-masing lapisan pada arsitektur *android*:

1. Linux Kernel

Android dibangun di atas kernel Linux. Namun secara keseluruhan *android* bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada *android* hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan *driver* layar, kamera, *keypad, WiFi, Flash Memory*, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

1. Media *Library* untuk memutar dan merekam berbagai macam format audio dan video.
2. *Surface Manager* untuk mengatur hak akses *layer* dari berbagai aplikasi.

3. *Graphic Library* termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
4. *SQLite* untuk mengatur relasi *database* yang digunakan pada aplikasi.
5. *SSl* dan *WebKit* untuk *browser* dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi *Android* 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

3. Android Runtime

Pada *android* tertanam paket pustaka inti yang menyediakan sebagian besarfungsi *android*. Inilah yang membedakan *android* dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi *android* menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya:

1. Pustaka Inti, *android* dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti *android* menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus *android*.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtualyang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan danpengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

4. Aplication Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi *android*. Selain itu, juga menyediakan abstraksiuntuk

mengakses perangkat, serta mengatur tampilan *User Interface* dan sumberdaya aplikasi. Bagian terpenting dalam kerangka aplikasi *android* adalah sebagai berikut:

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *Resource Manager* berfungsi untuk mengatur sumber daya yang ada dalam program. Seperti karakter, grafik dan *file* layout.
4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenail lokasi perangkat android berada.
5. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

5. Aplication Layer

Puncak dari diagram arsitektur *android* adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketikamenjalankan program. Pengguna hanya akan melihat program ketika digunakantanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalandalam *Android runtime* dengan menggunakan kelas dan *service* yang tersedia pada Framework aplikasi. Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasilainnya. Pada android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.11 Basic4Android

Basic4android adalah development tool sederhana yang *powerfull* untuk membangun aplikasi *Android*. Bahasa *Basic4android* mirip dengan bahasa *Visual Basic* dengan tambahan dukungan untuk objek. Aplikasi *Android* (APK) yang dicompile oleh *Basic4Android* adalah aplikasi *Android* native/asli dan tidak ada extra runtime seperti di *Visual Basic* yang ketergantungan file msvbvm60.dll,

yang pasti aplikasi yang dicompile oleh *Basic4Android* adalah *NO DEPENDENCIES* (tidak ketergantungan file oleh lain). IDE Basic4Android hanya fokus pada *development Android*[9].

Pada penelitian ini, *Basic4Android* digunakan sebagai perangkat kompilasi kode-kode program. *Basic4Android* sendiri merupakan tools RAD (*Rapid Application Development*) yang terdiri dari framework, library, dan IDE (*Integrated Development Environment*) yang terintegrasi dengan java dan *android SDK*. *Basic4Android* didesain sedemikian rupa sehingga memudahkan developer untuk mengembangkan aplikasi *android* menggunakan bahasa *Visual Basic* dan *IDE* yang mudah untuk digunakan.

Basic4Android juga menyediakan library-library lengkap untuk mengakses berbagai macam fitur yang ada pada *smartphone* seperti sensor, kamera, GPS, dan sebagainya[10].

2.11.1 Kelebihan dan Fitur Basic4Android

a. Simple dan *Powerfull* RAD (*Rapid Application Development*) tools untuk mengembangkan aplikasi native android tanpa harus menghabiskan waktu mempelajari pemrograman Eclipse/Java.
b. IDE (*Integrated Development Environment*) lengkap yang fokus 100% pada pengembangan aplikasi *Android*.
c. Dikompile kedalam native *bytecode*, tidak ada tambahan runtime library yang dibutuhkan. APK file yang dihasilkan sama dengan APK yang dihasilkan ketika anda membuat aplikasi menggunakan Eclipse/Java.
d. Performa dari aplikasi yang dihasilkan sama dengan aplikasi yang dibuat dengan Java.
e. *Object Oriented Programming Language* (sintaks sama dengan Visual Basic).
f. Tidak perlu menggunakan XML untuk *Layout*.
g. Sangat *ekstensible* dengan dukungan *custom java library*.

h. Satu-satunya visual editor untuk *android* yang benar-benar WYSIWYG. Visual editor mendukung banyak screen dan resolusi.
i. Mempunyai fitur *designer script*, membuat anda dapat dengan mudah membuat *layout* yang *sophisticated*.
j. Fitur *Basic4Android* UI *Cloud Services* dapat digunakan untuk mengetes aplikasi anda pada smartphone atau tablet asli lewat *cloud*.
k. Mendukung semua *android phone* dan tablet dari versi 1.6 sampai dengan 4x.
l. IDE yang modern dengan dukungan *autocomplete*, *built-in* dokumentasi, internal *index*, dan fitur *advance* yang lain.
m. Fitur *step by step debugger* yang *powerfull*.
n. Fitur *built-in code obfuscation.*
o. Mendukung semua *android core* (GPS, SQL *Database*, *Widgets*, Live Wallpaper, Bluetooth, USB, *Web Services*, *Camera*, JSON, XML, Excel CSV, *Multitouch*, NFC, *Views Animation*, *Push Notification*, *AdMob*, *OpenGL*, *Graphic and Chart*, Sensor, dll)

2.12 UML (*Unified Modeling Language*)

UML adalah kependekan dari *Unified Modeling Language* yang merupakan suatu cara untuk menyelesaikan suatu masalah dengan mendeskripsikannya yang telah menjadi standar dalam dunia industri untuk memvisualisasikan, merancang dan mendokumentasikan sistem perangkat lunak. Dengan menggunakan UML kita dapat membangun model untuk segala bentuk dan jenis aplikasi perangkat lunak, yang mana aplikasi yang dibangun dapat berjalan pada perangkat lunak dengan sistem operasi dan jaringan apapun. Selain itu, dengan menggunakan UML, programmer dapat mengerti, memahami dengan jelas maksud, tujuan dan arah rancangan sistem, sehingga sistem perngakat lunak yang dirancang dapat dibangun dengan bahasa program apapun. Namun karena UML merupakan suatu model pengembangan sistem perangkat lunak yang berbasis *object oriented* sehingga menggnuakan bentuk class dan operation dalam konsep dasarnya, maka lebih cocok dalam pemrogramannya dengan menggunakan bahasa-bahasa

pemrograman yang berbasis OOP (*Object Oriented Programming*) seperti C, C++, Java, VB dan lain sebagainya[11].

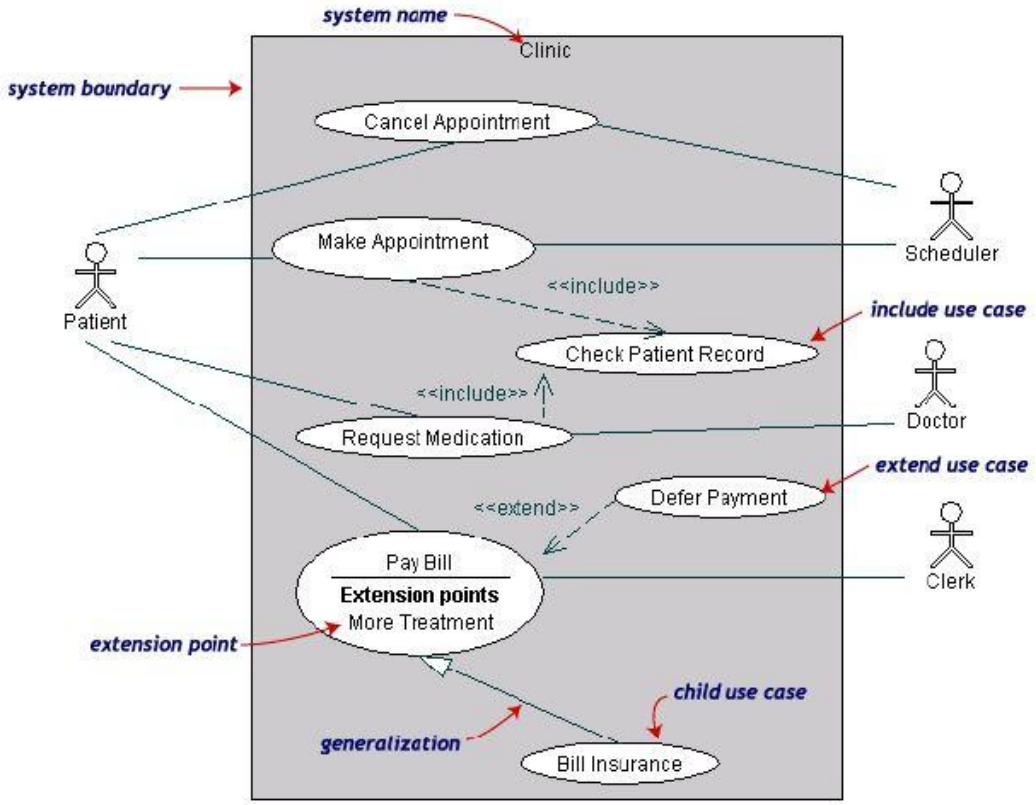
Secara lebih mendalam, UML lebih dari sekedar sebuah standard dan penemuan dari suatu notasi-notasi yang disatukan, tetapi juga berisi konsep-konsep baru yang menarik yang tidak ditemukan secara umum dalam komunitas object oriented. Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau simbol untuk merepresentasikan elemen-elemen dalam sistem. Bentuk diagram yang digunakan untuk merepresentasikannya adalah sebagai berikut:

1. *Use-case Diagram*
2. *Class Diagram*
3. *Sequence Diagram*
4. *Collaboration Diagram*
5. *Activity Diagram*
6. *Component Diagram*
7. *Deployment Diagram*

2.12.1 Use-Case Diagram

Use-case diagram merupakan suatu bentuk diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dirancang. Dalam *Use-case diagram* penekanannya adalah "apa" yang diperbuat oleh sistem, dan bukan "bagaimana". Sebuah use-case akan merepresentasikan sebuah interaksi antara pelaku atau actor dengan sistem.

Use-case diagram yang digunakan dalam mercancang suatu sistem dapat sangat membantu pada saat menyusun *requirement* sebuah sistem, mengomunikasikannya dengan klien, dan merancang pengujian untuk semua fitur yang terdapat dalam sistem. Dalam suatu sistem aplikasi database, use-case diagram sangat membantu requierement apa saja yang diperlukan.



Gambar 2.5 Contoh *Use-case Diagram*[11]

2.12.2 Class Diagram

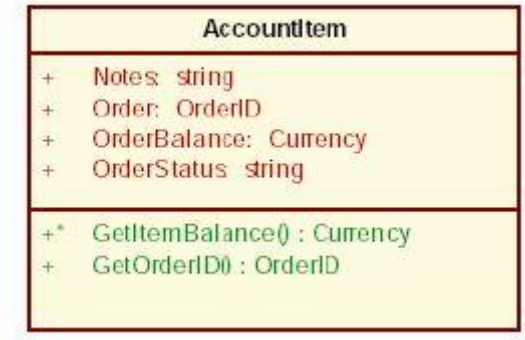
Sebuah *Class Diagram* menunjukkan struktur yang statis dari beberapa class dalamsuatu sistem. *Class-class* merepresentasikan suatu keadaan (atribut/properti) dan yang akan dikerjakan oleh sistem (metoda/fungsi). Class memiliki tiga area pokok:

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

Atribut dan metoda dalam class diagram dapat memiliki salah satu sifat seperti berikut di bawah ini:

1. Private, hanya dapat diakses oleh class itu sendiri.
2. Protected, hanya dapat diakses oleh class itu sendiri dan turunan dari class tersebut.
3. Public, dapat diakses oleh class selain dari class yang bersangkutan.

Class dapat direpresentasikan dalam sebuah *interface* atau sebaliknya merupakan implementasi dari sebuah *interface* yang berupa class abstrak yang hanya tidak memiliki *attribute* dan hanya memiliki metoda. Berikut merupakan bentuk class diagram secara umum:

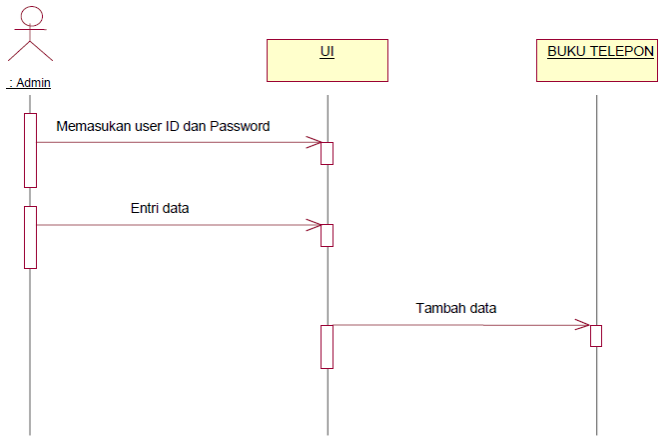


Gambar 2.6 Contoh *Class Diagram*[11]

2.12.3 Sequance Diagram

Sequence Diagram merupakan diagram yang mengambarkan kolaborasi yang dinamisa ntara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antar obyek di dalam dan di sekitar sistem yang berupa pesan atau instruksi yang berurutan.

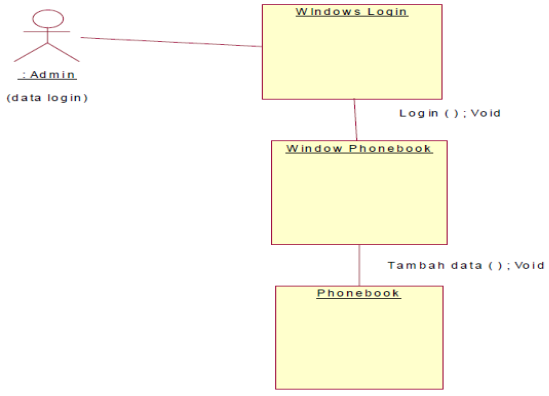
Sequence diagram umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah-langkah yang dilakukan baik oleh actor maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau output.



Gambar 2.7 Contoh *Sequance Diagram*[11]

2.12.4 Collaboration Diagram

Sebuah *collaboration diagram* menunjukkan kolaborasi yang dinamis yang mirip dengan sequence diagram. *Collaboration diagram* digambarkan sebagai sebuah object diagram dimana sejumlah obyek ditunjukkan disekitarnya dengan hubungan-hubungannya.

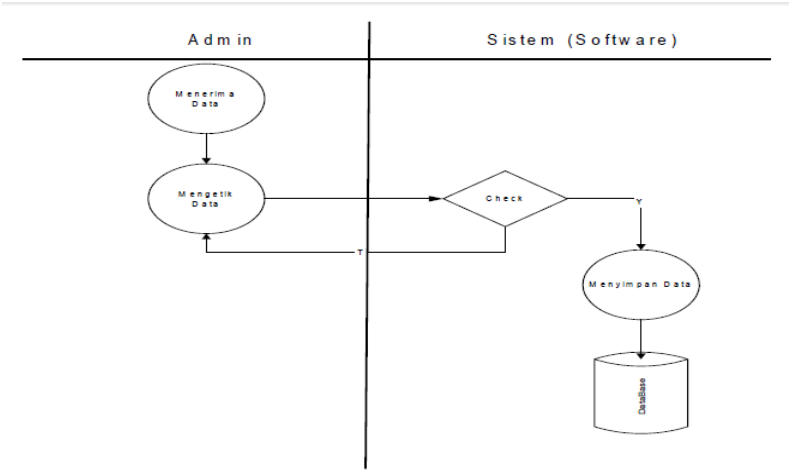


Gambar 2.8 Contoh *Collaboration Diagram*[11]

2.12.5 Activity Diagram

Sebuah *Activity Diagram* menunjukkan suatu alur kegiatan secara berurutan. *Activity Diagram* digunakan untuk mendiskripsikan kegiatan-kegiatan dalam

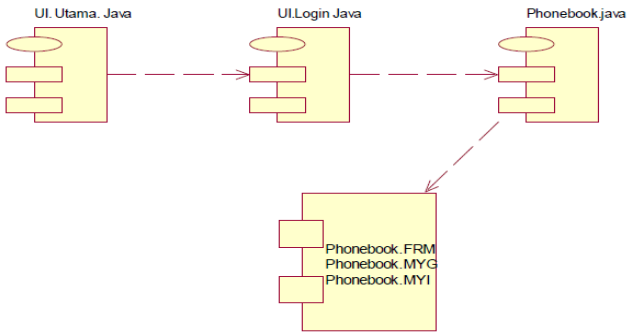
sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alurkegiatan yang lainnya seperti use case atau suatu interaksi.



Gambar 2.9 Contoh *Activity Diagram*[11]

2.12.6 Component Diagram

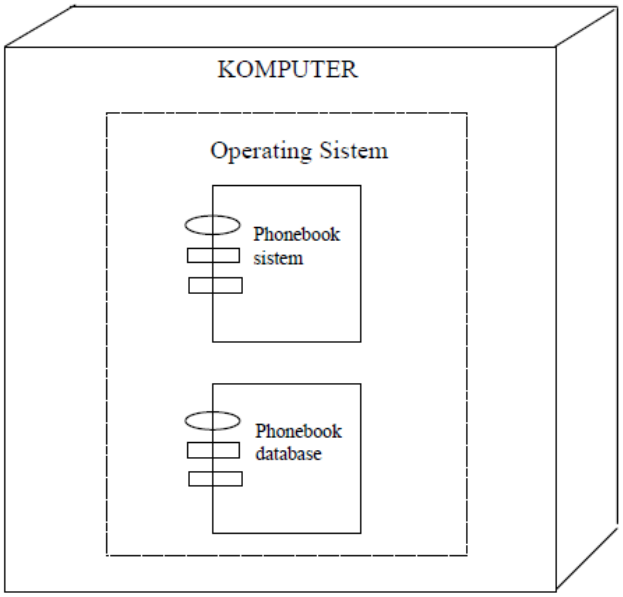
Component Diagram menunjukkan struktur dan hubungan antar komponen software termasuk ketergantungan (*dependency*) diantara komponen-komponen tersebut. Komponen pada piranti lunak adalah berupa modul-modul yang berisikan code, baik *library* maupun *executable*. Umumnya komponen yang terbentuk dari beberapa class dan/atau package, atau juga dapat dari komponen-komponen yang lebih kecil.



Gambar 2.10 Contoh *Component Diagram*[11]

2.12.7 Deployment Diagram

Deployment Diagram menunjukkan arsitektur fisik pada hardware dan software pada suatu sistem yang dirancang. *Deployment diagram* juga dapat menunjukkan perngkat-perangkat dan nodes diantara hubungan yang dimilikinya antar komponen.



Gambar 2.11 Contoh *Deployment Diagram*[11]

2.13 GPS (*Global Positioning System*)

Global Positioning System (GPS) adalah suatu sistem navigasi atau penentu posisi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika Serikat (US DoD = *United States Department of Defense*). Sistem ini didesain untuk memberikan posisi dan informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa tergantung waktu dan cuaca. Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (*longitude/latitude*)[12].

Pada dasarnya GPS terdiri atas tiga segmen yaitu segmen angkasa, kontrol/pengendali, dan pengguna, ketiga segmen tersebut mempunyai fungsi masing – masing diantaranya :

a. Segmen angkasa: terdiri dari 24 satelit yang berfungsi untuk mengirimkan posisi dan waktu kepada pengguna di seluruh dunia.
b. Segmen Kontrol/Pengendali: pusat pengendali utama yang terdapat di Colorodo Springs, dan 5 stasiun pemantau lainnya dan 3 antena yang tersebar di bumi ini.
c. Segmen Pengguna: Pada sisi pengguna dibutuhkan *receiver* GPS (perangkat GPS) untuk dapat menerima signal satelit GPS dan kemudian menghitung posisi, kecepatan dan waktu.

Untuk dapat menggunakan GPS sebagaimana fungsinya setidaknya GPS harus memliki 3 signal satelit untuk menghitung posisi 2D dan dibutuhkan 4 atau lebih signal satelit untuk menghitung posisi 3D *position* (*longituge*, *latitude*, dan *altitude* (ketinggian).

2.13.1 Titik Koordinat

Titik koordinat adalah titik yang berpedoman pada garis latitude dan longitude suatu daerah. Kaitannya dengan latitude dan longitude adalah, kedua garis lintang dan bujur inilah (*latitude* = garis lintang, *longitude* = garis bujur) yang menentukan di perolehnya suatu nilai derajat dari suatu titik yang diukur.

Koordinat pada umumnya dituliskan dengan format DD (*Degree Decimal*), terkadang juga dituliskan dengan format DMS (*Degree-Minute-Second*), jika berada dibelahan bumi utara dan timur maka bernilai positif, dalam hal ini untuk nilai Lintang atau *Latitude* dan Bujur atau *Longitude*.

2.13.2 Konversi Derajat Bujur dan Lintang

Perlu di ketahui pula konversi derajat menit (‘) dan detik (“) pada letak astronomis suatu tempat dalam satuan jarak (km dan m). Dalam pemakaian GPS sering didapatkan titik koordinat suatu posisi lokasi dalam bentuk *Latitude* (Lintang), *Longitude* (Bujur) atau *Degree* (Derajat). Dalam hal ini ada suatu formula yang digunakan untuk mengkonversi bilangan tersebut dari bentuk *Degree Minute Second* (DMS) menjadi *Degree Decimal* (DD), atau sebaliknya konversi *Degree Decimal* (DD) menjadi *Degree Minute Second* (DMS)[13].

a. Contoh perhitungan Konversi *Degree Decimal* (DD) menjadi *Degree Minute Second* (DMS):

   Bila diperoleh angka 109.036889 maka diubah menjadi 109˚.

   Angka dibelakang koma dikalikan dengan bilangan desimal 60, maka 0,036389 x 60 = 2,18334, urutan kedua yaitu 2 menit.

   Sisa angka dibelakang koma diatas (18334) dikalikan dengan bilangan desimal 60, maka 0,18334 x 60 = 11,0004, urutan ketiga yaitu 11 detik.

   Hasil dari konversi , secara lengkap ditulis 109˚2'11", hal ini juga berlaku pada Latitude.

b. Contoh perhitungan Konversi *Degree Minute Second* (DMS) menjadi *Degree Decimal* (DD):

   Bila diperoleh angka 7˚13'13", maka dapat diubah menjadi,

   = (Nilai Degree) + (Nilai Minute/60) + (Nilai Second/3600)

   = 7 + (13/60) + (13/3600)

   = 7.220278

   Maka hasil konversi DMS ke DD adalah 7.220278.

2.13.3 Menghitung Jarak Antar Koordinat

Jika menggunakan Google -Earth akan langsung dapat diketahui jarak antar titik-titik koordinat dengan menarik garis dari satu sisi titik ke titik yang diinginkan . Jika tidak maka bisa dihitung menggunakan rumus 1' = 1.825 m, 1" = 30,416 m. Selanjutnya selisih angka N menghasilkan sumbu vertical, sedangkan selisih angka E menghasilkan sumbu *horizontal*. Adapun jarak antar titik atau koordinat adalah merupakan garis miring sebuah sisi segitiga siku. Berikut adalah contoh perhitungan jarak antara Lapang AGATIS Tanjung Selor dengan Tempat Tinggal Massugeng[14].

Tabel 2. 2 Contoh Perhitungan Jarak Koordinat

	Tempat Tinggal Massugeng	Lapang AGATIS	Selisih	Dikalikan 30,416 m
N	2˚50'44.42"	2˚50'26.49"	17.93"	545,36
E	117˚22'20.86"	117˚22'26.06"	5.10"	155,12

Maka jarak tempat tinggal Massugeng dengan Lapang AGATIS adalah $\sqrt{(545{,}36^2 + 155.12^2)\text{m}^2}$ = 566,99 m.

2.14 Trigonometri

Trigonometri adalah bagian dari ilmu matematika yang mempelajari tentang hubungan antara sisi dan sudut suatu segitiga serta fungsi dasar yang muncul dari relasi tersebut. Trigonometri merupakan nilai perbandingan yang didefinisikan pada koordinat kartesius atau segitiga siku-siku. Bagi para siswa, trigonometri identik dengan fungsi trigonometri yang meliputi sinus (sin), cosinus (cos), tangen (tan), cosecan (cosec), secan (sec), dan cotangen (cotan) yang kesemuanya b merupakan cara untuk menentukan suatu sisi sebuah segitiga atau sudut yang terbentuk dari dua buah sisi dalam sebuah segitiga[15].

Trigonometri merupakan ilmu matematika yang sangat penting dalam kehidupan. Aplikasi ilmu trigonometri dalam kehidupan mencangkup segala bidang seperti astronomi, geografi, teori musik, elektronik, ekonomi, medical, teknik, dan masih banyak lagi. Dengan trigonometri kita bisa mengukur jarak suatu bintang diangkasa tanpa harus pergi kesana. Dengan trigonometri kita bisa mengukur sudut ketinggian tebing tanpa harus memanjatnya. Bisa mengukur lebar suatu sungai tanpa harus menyeberanginya. Itulah manfaat dari mempelajari trigonometri dalam kehidupan sehari-hari.

Trigonometri adalah sebuah konsep. Hal pertama yang perlu dimengerti dalam memahami konsep dasar trigonometri adalah mengetahui, mengerti dan memahami bentuk dan rumus-rumus sebuah segitiga, terutama segitiga siku-siku. Pada dasarnya sebuah segitiga selalu terdiri dari 3 sisi, yaitu sisi miring, sisi samping, dan sisi depan. Dan tiga buah sudut yaitu sudut tegak lurus, sudut depan

dan sudut samping. Dimana jika di tambahkan jumlah sudut sebuah segitiga haruslah 180 derajat.

Tujuan utama mempelajari trigonometri dalam ilmu matematika adalah untuk menemukan nilai sebuah sudut atau panjang sebuah sisi sebuah segitiga. Untuk tujuan tersebut diatas maka trigonometri memiliki 2 nilai fungsi, yaitu:

1. Nilai fungsi trigonometri unuk sudut istimewa.
   Sudut istimewa disini adalah sudut yang besarnya 0, 30, 45, 60, 90 derajat. Untuk menentukan nilai fungsi sudut istimewa digunakan konsep geometri.
2. Nilai fungsi trigonometri untuk sudut lainnya.
   Untuk menentukan nilai fungsi trigonometri sudut tidak istimewa biasanya menggunakan tabel atau scientific kalkulator yang dilengkapi dengan fungsi trigonometri.

2.14.1 Identitas Trigonometri

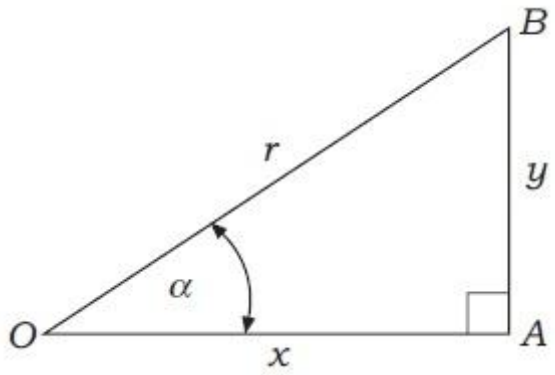
Dari nilai fungsi trigonometri tersebut kemudian diperoleh identitas trigonometri. Identitas trigonometri adalah suatu persamaan dari fungsi trigonometri yang bernilai benar untuk setiap sudutnya dengan kedua sisi ruasnya terdefinisi. Identitas trigonometri terbagi 3, yaitu Identitas Kebalikan, Identitas Perbandingan dan Identitas Phytagoras yang masing-masing memiliki fungsi dasar, yaitu:

Tabel 2.3 Identitas Trigonometri

Identitas Kebalikan	**Identitas Perbandingan**	**Identitas Phytagoras**
Cosec A = 1/ sin A	Tan A = Sin A/ Cos A	$Cos^2 A + Sin^2 A = 1$
Sec A = 1/cos A	Cot A = Cos A / Sin A	$1 + Tan^2 A = Sec^2 A$
Cot A = 1/ tan A		$1 + Cot^2 A = Cosec^2 A$

2.14.2 Perbandingan Trigonometri

Gambar berikut menunjukkan segitiga OAB siku-siku di A dengan besar sudut O = α°



Gambar 2. 12 Perbandingan Trigonometri[15]

dimana:

x = sisi siku-siku samping sudut (proyeksi)

y = sisi siku-siku depan sudut (proyektor)

r = sisi miring (proyektum)

Perbandingan trigonometri untuk sudut α pada segitiga siku-siku OAB didefinisikan sebagai berikut.

$$\sin\alpha = \frac{y}{r} \qquad \csc\alpha = \frac{r}{y}$$

$$\cos\alpha = \frac{x}{r} \qquad \sec\alpha = \frac{r}{x}$$

$$\tan\alpha = \frac{y}{x} \qquad \cot\alpha = \frac{x}{y}$$

dengan:

sin α dibaca sinus α

cos α dibaca cosinus α

tan α dibaca tangen α

csc α dibaca cosecan α

sec α dibaca secan α

cot α dibaca cotangen α

Dari perbandingan di atas, kita memperoleh hubungan sebagai berikut

$$\cos \alpha = \frac{1}{\sin \alpha}$$

$$\sec \alpha = \frac{1}{\cos \alpha}$$

$$\cot \alpha = \frac{1}{\tan \alpha}$$