BAB 2
LANDASAN TEORI

2.1 PT. Coronet Crown

PT. Coronet Crown adalah perusahaan farmasi yang memproduksi banyak jenis obat, baik untuk obat dengan resep dokter (*ethical*) atau obat bebas tanpa resep dokter (OTC- *Over The Counter*). Perusahaan farmasi ini telah berdiri sejak tahun 1960 dengan nama "PT CORONET SYNDYCATE" yang beralamat di Jalan Bibir 14 Surabaya. Perusahaan farmasi ini hanya memproduksi bedak dan vitamin saja. Saat ini PT. Coronet Crown telah memiliki 3 distributor utama yaitu PT. Serdam Farma, PT. Cempaka Indah Murni, dan PBF Coronet Bandung. Sejak tahun 1988 sampai sekarang berlokasi di Jalan Raya Taman km 15, Sidoarjo, Surabaya, dengan Presiden Direktur Bapak Singgih Darmawan. Semboyan dari PT Coronet Crown adalah "Mutu adalah Nomor 1". Berbagai penghargaan telah diraih oleh Herocyn, antara lain apresiasi Top Brand Award mulai tahun 2007 sampai saat ini. PT. Coronet Crown terus melakukan upaya-upaya agar dapat lebih dekat dengan para pengguna setia, antara lain dengan mengadakan atau menjadi *sponsorship event fun bike*, dunia anak, maupun bentuk-bentuk promosi lainnya.

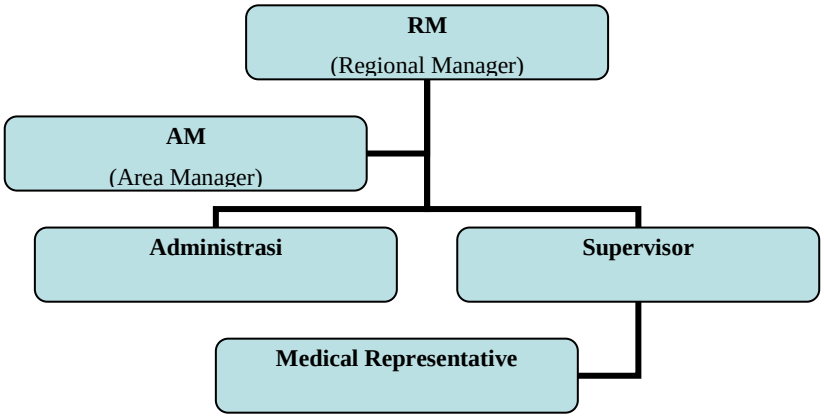
Pertumbuhan dan perkembangan yang terjadi pada dunia usaha di Indonesia akhir-akhir semakin pesat, sehingga menyebabkan terjadinya persaingan yang ketat diantara perusahaan. Oleh karena itu, PT. Coronet Crown harus memiliki strategi bisnis yang tepat agar dapat mempertahankan bahkan meningkatkan penjualan produk-produk yang dimiliki. Selain strategi bisnis yang mumpuni, PT. Coronet Crown pun harus memiliki orang-orang yang ulet dan telaten untuk dijadikan sebagai *medical representative* atau lebih dikenal dengan sebutan medrep.

Medical Representative bisa didefinisikan sebagai duta perusahaan atau seseorang yang dipercaya untuk mewakili perusahaan guna mempromosikan produk secara profesional, kredibel, dan berintegritas. Istilah lainnya adalah *detailer*, karena dalam melaksanakan tugasnya, seorang *medical*

representative harus mampu menjelaskan seluk beluk produk yang ditawarkan secara detail kepada kostumernya. *Medical representative* ini dibutuhkan oleh perusahaan farmasi sebagai perpanjangan tangan agar produknya diketahui oleh dokter atau apotek [2].

Salah satu tugas utama seorang *medical representative* adalah melakukan kunjungan secara rutin ke dokter. Ada dokter yang dikunjungi seminggu sekali, ada juga yang cukup 2 kali sebulan. Bahkan untuk dokter-dokter tertentu, ada yang dikunjungi lebih dari sekali dalam seminggunya.

2.2 Struktur Organisasi



Gambar 2. 1 Struktur Organisasi Perusahaan

2.3 Deskripsi Jabatan

Adapun tugas dari masing-masing departemen dari struktur organisasi diatas diantaranya:

1. ***Regional manager***
    a. Mengkoordinator semua kegiatan operasional yang berada di bawah tanggungjawabnya, seperti order, stok penjualan, *display*, penjualan, keamanan serta kepuasan konsumen
    b. Melakukan *briefing* kepada *supervisor*
2. ***Area Manager***
    a. Mengawasi kehadiran serta keberadaan semua karyawan yang bertugas
    b. Melakukan briefing kepada *coordinator counter* dan timnya

c. Melakukan penilaian berkala kinerja semua karyawan
d. Membuat laporan penjualan mingguan dan bulanan

3. *Administration*

a. Menginput data karyawan
b. Mencatat absen kehadiran karyawan
c. Membuat laporan pertanggung jawaban absen kehadiran karyawan setiap bulannya
d. Memasukan sales masing masing karawan dan mengirim ke pusat

4. *Supervisor*

a. Mengawasi seluruh kegiatan *medical representative.*
b. Menerima laporan hasil pekerjaan *medical representative.*
c. Memastikan sarana dan prasarana yang ada di lapangan masih berfungsi dengan baik
d. Melakukan *survey competitor* dan melakukan analisa perbaikan
e. Membuat laporan penjualan dan omset migguan dan bulanan

5. *Medical representative*

a. Mensurvey tempat dan alamat dokter praktek di area masing masing
b. Mensurvey ke setiap apotek untuk mengetahui resep dokter yang masuk ke apotek
c. Mensurvey produk kompetitor di setiap apotek dan klinik di setiap area
d. Mengunjungi dokter dan melakukan tawar menawar untuk tujuan kerjasama dengan pihak PT. Coronet Crown Bandung

2.4 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*Instruction*) atau pernyataan (*Statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

Aplikasi merupakan perangkat lunak yang berada di dalam sistem dan digunakan untuk mengolah data untuk menjadi suatu informasi yang berguna bagi penggunanya. Aplikasi juga dapat diartikan sebagai suatu program yang siap

untuk digunakan yang dibuat untuk fungsi – fungsi tertentu dengan harapan dapat membantu memudahkan penggunanya.

2.5 Data absensi

Data absesnsi yang sedang berjalan saat ini di PT. Coronet Crown masih berbentuk manual menggunakan kartu kunjungan. Setiap medrep harus membawa seluruh kartu kunjungan dokter serta data-data dokter untuk kemudian ditandatangani sebagai bukti bahwa medrep tersebut telah melakukan kunjungan.

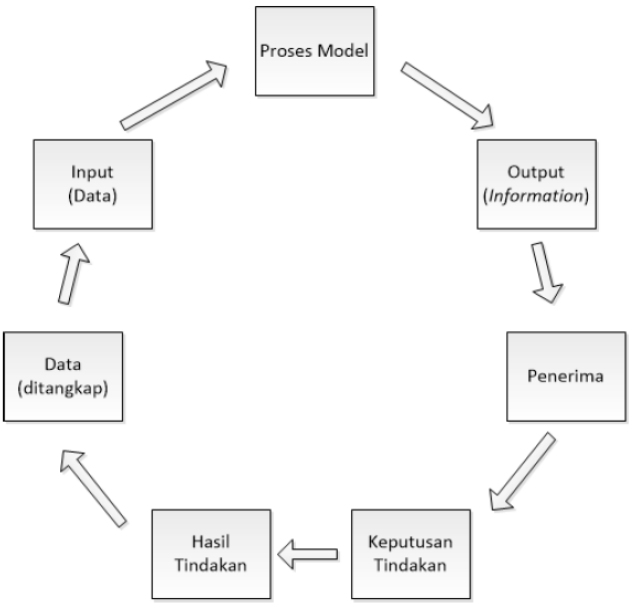
2.5.1 Data

Data merupakan sesuatu yang tidak mempunyai arti bagi penggunanya dan masih memerlukan adanya suatu pengolahan terhadap data tersebut. Data bisa berbentuk suatu kejadian, benda, aktifitas, gambar, huruf dan sebagainya. Data adalah bahan mentah untuk menjadi suatu informasi.

2.5.2 Informasi

Informasi merupakan data yang telah diproses atau diolah sehingga memiliki nilai tertentu bagi penggunanya. Keberadaan data sangatlah penting karena data merupakan bahan utama untuk didapatkannya suatu informasi yang dibutuhkan.

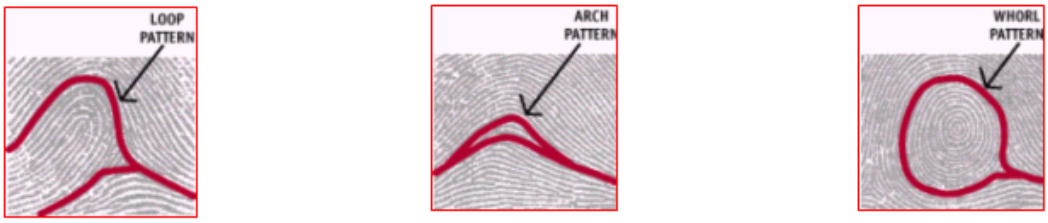
Informasi juga diartikan sebagai hasil proses, manipulasi, dan penataan dari sekelompok data yang memiliki sebuah nilai pengetahuan bagi penggunanya. Sumber dari suatu informasi adalah data. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali [3]. Data tersebut akan ditangkap sabagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi ini dapat digambarkan sebagai berikut :



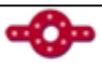
Gambar 2. 2 Siklus Informasi

2.6 *Fingerprint*

Fingerprint adalah alat yang digunakan sebagai pengenal identitas, melindungi komputer dari data atau pemakaian komputer oleh orang yang tidak bertanggung jawab. Dengan menggunakan *Fingerprint* sebagai pengganti tanda tangan dapat mengetahui identitas pemilik dari sidik jari. Sidik jari adalah gurat-gurat yang terdapat di kulit ujung jari [4].



Gambar 2. 3 Proses Deteksi Pola Sidik Jari

 Ridge	Ridge	Mempunyai ketegasan jarak ganda dari permulaan ke-akhir, sebagai lebar ridges satu dengan lainya
 Evading Ends	Evading Ends	Dua ridge dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 Bifurcation	Bifurcation	Dua ridge dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 Hook	Hook	Ridges merobek; satu ridges tidaklah lebih panjang dibanding 3mm
 Fork	Fork	Dua ridges dihubungkan oleh sepertiga ridges tiddak lebih panjang dibanding 3mm
 Dot	Dot	Bagian ridges adalah tidak lagi dibanding ridges yang berdekatan
 Eye/Island	Eye	ridges merobek dan menggabungkan lagi di dalam 3mm
 Eye/Island	Island	Ridges merobek dan tidak ber menggabung lagi, kurang dari 3mm dan tidak lebih dari 6mm. Area yang terlampir adalah Ridge.
 Enclosed Ridge	Enclosed Ridge	Ridges tidak lebih panjang dibanding 6mm antara dua ridges
 Enclosed Loop	Enclosed Loop	Yang tidak mempola menentukan pengulangan antar dua atau lebih ridges paralel
	Specialties	Rare ridge membentuk seperti tanda tanya dan sangkutan pemotong

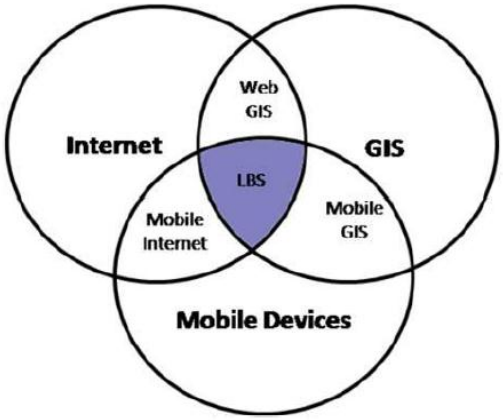
Gambar 2. 4 Pola deteksi pada sensor

2.7 GPS

Global Positioning System (GPS) merupakan sistem navigasi yang menggunakan satelit dalam penggunaannya. GPS terdiri dari tiga bagian yaitu sistem kontrol, satelit dan pengguna. Sistem kontrol adalah bagian yang mengontrol pergerakan satelit – satelit yang ada dan saling berinteraksi satu sama lain kemudian pengguna adalah alat navigasi yang digunakan seperti perangkat mobile yang kini sudah memiliki fitur GPS didalamnya. GPS biasanya digunakan untuk menunjukan suatu lokasi yang berada di permukaan bumi dengan tingkat akurasi yang cukup baik yaitu kurang dari 10 meter selama tidak ada benda padat yang dapat menghambat sinyal untuk mendapatkan lokasi pengguna.

2.8 Location Based Service

Location Based Service (LBS) atau layanan berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi dengan kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS memungkinkan untuk melakukan komunikasi dan interaksi dua arah. LBS ini dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu *Geographic Information System* (GIS), Internet *Service* dan *mobile device*, seperti pada gambar 2.2 : [5]



Gambar 2. 5 Teknologi Location Based Service

2.8.1 Metode Penentu Lokasi

LBS merupakan layanan yang mampu bereaksi aktif pada suatu perubahan posisi sehingga dapat mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut. Pada teknologi LBS berbasis jaringan seluler, penentuan posisi pada sebuah peralatan komunikasi ditentukan berdasarkan posisi relatif terhadap lokasi BTS (*Base Transceiver Station*). Dalam menentukan posisi dari sebuah perangkat mobile yang sedang aktif, secara umum terdapat tiga tingkat metode yang digunakan saat ini, diantaranya adalah sebagai berikut : [5]

1. Metode *Basic Positioning* berbasis pada *Cell Identification* (Cell ID)

   Penentuan posisi didasarkan pada daerah geografis yang tercakup oleh sebuah *Cell* berhubungan dengan daerah cakupan dari sinyal radio. Ketika sebuah *handphone* terhubung secara aktif dengan sebuah *base station*,

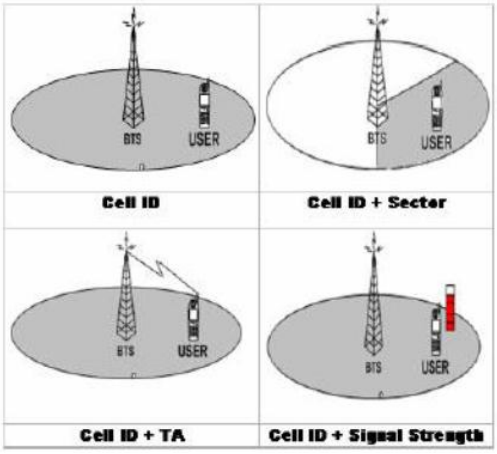
berarti *handphone* tersebut diasumsikan berada dalam *cell* dari *base station* tersebut. Untuk mengukur jarak dan arah *handset* dari *base station* tidak dapat diketahi dengan pasti, oleh karena itu untuk lebih meningkatkan lagi akurasi hasil pencarian pada metode ini seringkali dikombinasikan dengan metode lain seperti:

1. *Timing Advance* (TA)

   Dengan menggunakan TA ini metode Cell ID akan ditambahkan sebuah fungsionalitas untuk menghitung *Round Trip Time* (RTT), yaitu waktu transmisi sebuah *frame* (dari *base station* ke *handphone*) dan waktu sebuah *frame* (dari *handphone* ke *station*). Dengan tambahan ini jarak antara *handphone* dan *base station* dapat ditentukan dengan keakuratan 50 m.

2. *Network Measurement Report* (NMR)

   Berdasarkan pada besar kecilnya sinyal (*Receiver Signal Strength*) yang diterima *handphone* yang ada di suatu "*sector cell*", maka posisi itu dapat ditentukan dengan lebih akurat.

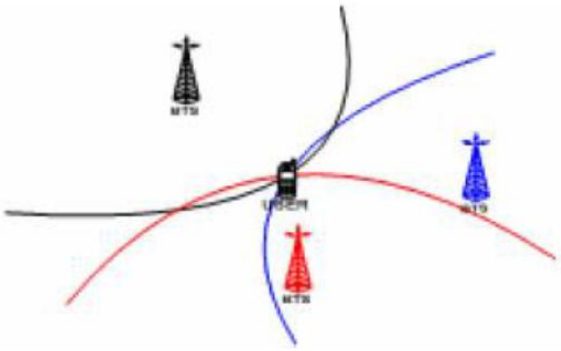


Gambar 2. 6 ***Network Measurement Report***

3. Metode Enhanced Posittioning

Metode ini menggunakan pendekatan *Observe Time Difference* (OTD) dengan jaringan GSM yang sering digunakan seperti *Enhanced*-OTD (E-OTD). E-OTD adalah metode pencarian posisi yang berdasarkan pada

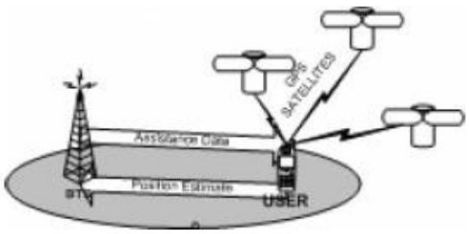
waktu. Untuk menentukan posisi *relative*, sebuah handphone harus aktif terhadap tiga *base station* dan perlu ditentukan terlebih dahulu jarak *handphone* terhadap masing- masing *base station* berdasarkan waktu yang ditempuh oleh sebuah sinyal dari handphone ke masing – masing *base station*. Untuk menentukan posisi dari handphone yang sedang aktif digunakan rumus matematika untuk triangulasi. Dengan metode ini akurasinya akan meningkat hingga ketilitian kurang dari 50 m.



Gambar 2. 7 *Enhanced Positioning*

2. Metode *Advance Positioning*

Metode ini menggunakan *teknologi Assisted-Global Positioning System* (A-GPS). A-GPS ini merupakan metode yang berbasis pada waktu. Pada metode ini, akan dilakukan pengukuran waktu tiba dari sebuah sinyal yang dikirim dari tiga buah satelit GPS. Hal ini berarti handset harus memiliki fasilitas untuk mengakses GPS. A-GPS juga menghasilkan akurasi secara *vertical* dan estimasi jarak yang baik. Akurasinya pun kurang dari 10 m.



Gambar 2.1 Advance Positioning

2.8.2 Komponen *Location Based Service*

Dalam menggunakan *Location Based Service* atau layanan berbasis lokasi terdapat beberapa elemen yang diperlukan , antara lain :

1. *Mobile Devices*

   Sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan seperti perangkat *mobile smartphone* yang memiliki fasilitas navigasi.
2. *Communication Network*

   Jaringan selular yang mengirimkan data pengguna dan permintaan layanan.
3. *Positioning Component*

   Untuk dapat mengolah layanan harus menentukan lokasi pengguna. Posisi pengguna ini dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS).
4. *Service and Application Provider*

   Penyedia layanan pengguna selular yang bertanggung jawab untuk memproses suatu layanan.
5. *Data and Content Provider*

   Penyedia layanan informasi data yang dapat diminta oleh pengguna.

2.8.3 *Latitude* (Garis Lintang)

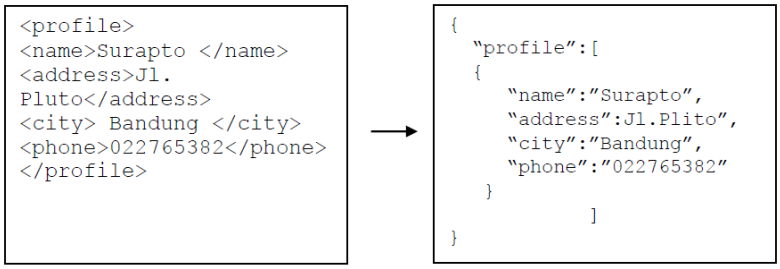
Latitude adalah garis yang melintang di antara kutub utara dan kutub selatan yang menghubungkan sisi bagian bumi dari timur dan barat. Garis lintang memiliki posisi membentangi bumi seperti halnya garis khatulistiwa tetapi dengan kondisi nilai tertentu. *Latitude* dibedakan menjadi dua wilayah yaitu lintang utara dan selatan dimana nilai koordinat di bagian utara selalu positif dan koordinat selatan selalu negatif.

2.8.4 *Longitude* (Garis Bujur)

Longitude adalah garis bujur yang menghubungkan antara sisi utara dan sisi selatan permukaan bumi. Garis bujur digunakan untuk mengukur sisi barat dan timur koordinat suatu titik di belahan bumi. *Longitude* dibedakan menjadi dua wilayah yaitu bujur timur dan bujur barat dimana nilai koordinat yang berada di timur selalu negatif dan nilai koordinat dibagian barat selalu positif.

2.9 JSON (Javascript Object Notation)

JSON atau *Javascript Object Notation* adalah format untuk pertukaran data yang lebih ringan,mudah dibaca dan ditulis oleh manusia serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON lebih unggul dari pada format pertukaran data yang lain seperti XML. Mulai dari kecepatan, penulisan yang lebih mudah dan *coding parsing* yang lebih ringkas dan sederhana. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman lain / apapun karena menggunakan bahasa pemrograman yang umum digunakan yaitu keluarga C seperti C, C#, C++, Java, Javascript, Perl, Python dan lain – lain. [7]
Berikut adalah contoh konversi data XML ke dalam format JSON :



Gambar 2. 8 JSON (*Javascript Object Notation)*

2.10 Google *Maps*

Google *Maps* adalah sebuah layanan gratis dari Google tentang berbagai macam teknologi pemetaan. Google *Maps* menampilkan peta dunia secara digital sehingga lebih memudahkan ketika ingin mengetahui suatu lokasi tertentu. Google *Maps* menyediakan beberapa keunggulan seperti peta jalan, menunjukan rute untuk suatu perjalanan yang diinginkan baik itu melalui jalur mobil sampai pejalan kaki.

2.11 Google *Maps* API

Google *Maps* API adalah sebuah layanan yang memungkinkan Google *Maps* agar dapat diintegrasikan ke dalam aplikasi yang memang membutuhkan fitur dari Google *Maps* ini. Google *Maps* API adalah sebuah *library Javascript*. Dengan menggunakan Google *Maps* API ini dapat memudahkan pengembang dalam membuat aplikasi yang membutuhkan teknologi peta di dalamnya dan tidak

perlu memikirkan cara membuatnya secara manual karena dengan menggunakan Google *Maps* API ini sudah semua yang dibutuhkan terkait teknologi peta digital sudah cukup lengkap dan dapat dimanfaatkan untuk aplikasi yang membutuhkannya [8].

Saat ini Google *Maps* API telah mengalami perkembangan, hingga saat ini Google Maps API telah mencapai versi 3. Namun, Google *Maps* API versi 3 ini masih hanya bisa digunakan oleh pemrograman berbasis web dengan menggunakan *Javascript*. Untuk penggunaan Google *Maps* API pada perangkat *mobile* seperti misalnya android Google Maps API yang bisa digunakan saat ini telah mencapai versi 2. Perkembangan pada Google Maps Versi 2 ini adalah dengan memiliki fitur – fitur yang lengkap dari versi sebelumnya salah satunya adalah dapat menampilkan peta dengan sudut yang diingkan dan berupa 3D *view*. Selain itu pada Google *Maps* API terdapat beberapa pilihan model peta yang ditampilkan seperti *Roadmap, Satellite, Terrain* dan *Hybrid*.

2.12 Object Oriented Programming (OOP)

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan atau *development* suatu *software* dimana dalam struktur *software* tersebut didasarkan kepada interaksi *object* dalam penyelesaian suatu proses atau tugas. Interaksi tersebut mengambil form dari pesan-pesan dan mengirimkannya kembali antar *object* tersebut. *Object* akan merespon pesan tersebut menjadi sebuah tindakan atau metode [9]. Adapun konsep pada pemrograman berorientasi objek ini yaitu:

1. Object

   Object adalah sebuah struktur yang menggabungkan data dan prosedur untuk bekerja bersama-sama.

2. Abstraction

   Ketika membangun *objects* dalam aplikasi OOP, adalah penting untuk menggabungkan konsep *abstraction* ini. Jika membangun aplikasi shipping, maka harus membangun *object* produk dengan atribut seperti ukuran dan berat. Warna adalah contoh informasi yang tidak ada

hubungannya dan harus dibuang. Tetapi ketika akan membangun *order-entry application*, warna menjadi penting dan harus termasuk atribut *object* produk.

3. Enkapsulasi

   Sebuah proses dimana tidak ada akses langsung ke data yang diberikan, bahkan hidden. Jika ingin mendapat data, harus berinteraksi dengan *object* yang bertanggung jawab atas data tersebut.

4. Polymorphisms

   Kemampuan dua buah *object* yang berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik. Contohnya, dapat mengirim pesan *print* ke *object* printer yang akan mencetak pada printer, dan juga dapat mengirim pesan yang sama ke *object screen* yang akan menuliskan pada *screen* monitor. Dalam OOP, diterapkan tipe *polymorphism* melalui proses yang disebut *overloading*. Dapat dilakukan dalam implementasi metode yang berbeda pada sebuah *object* yang mempunyai nama yang sama.

5. Inheritance

   Penggunaan *inheritance* dalam OOP untuk mengklasifikasikan *objects* dalam program sesuai karakteristik umum dan fungsinya. Hal ini akan membuat pekerjaan bersama *object* lebih mudah dan lebih intuitif. Hal ini juga membuat programming lebih mudah karena memungkinkan untuk mengkombinasikan karakteristik umum kedalam *object parent* dan mewariskan karakteristik ini ke *child object*.

2.13 Unified Modeling Language (UML)

Unified Modelling language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataanya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan

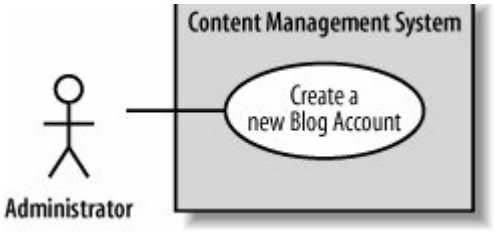
persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif [10].

Bahasa pemrograman grafis telah ada di industri perangkat lunak sejak lama. Pemicu utaman di balik semuanya adalah bahasa pemrograman berada pada tingkat abstraksi yang tidak terlalu tinggi untuk diskusi tentang desain. UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Commom Object Request Broker Architecture*). Uml lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an.

2.13.1 Use Case Diagram

Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Sebuah use case juga menggambarkan cara sistem berperilaku untuk memenuhi kebutuhan. Use *case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang sistem tersebut digunakan. Dalam bahasan *Use case*, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem. Aktor tidak harus manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem lain tersebut merupakan aktor [10].

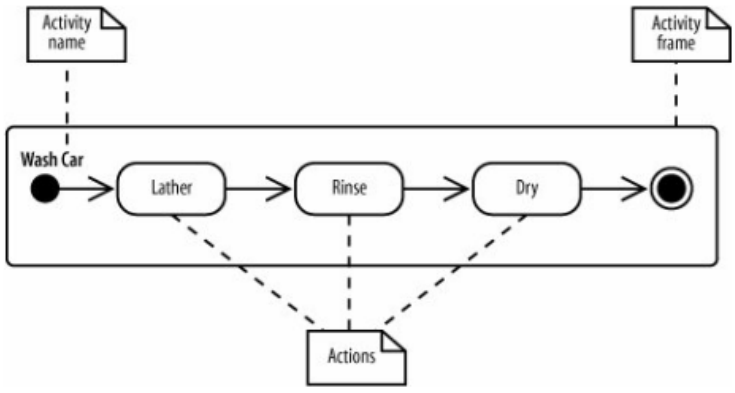
Berikut contoh bentuk *Use Case Diagram* yang mengkomunikasikan antara aktor *Use Case* di dalam sebuah sistem :



Gambar 2. 9 Contoh Use Case Diagram

2.13.2 Activity Diagram

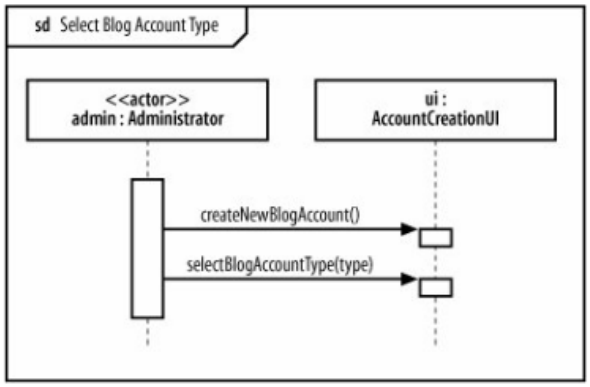
Activity diagram adalah teknik untuk menggambrakan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antar diagram ini dan notasi diagram alir adalah diagram ini mendukung sifat pararel. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya. Dengan kata lain, diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus kita ikuti [11]. Hal ini penting untuk pemodelan bisnis, karena proses-proses sering muncul secara pararel. Ini juga berguna pada algoritma yang bersamaan, di mana urutan-urutan independen dapat melakukan hal-hal secara pararel. Contoh sederhana penggunaan *activity diagram* pada kasus proses mencuci mobil dapat dilihat pada gambar 2.10 :



Gambar 2. 10 Contoh Activity Diagram

2.13.3 Sequence Diagram

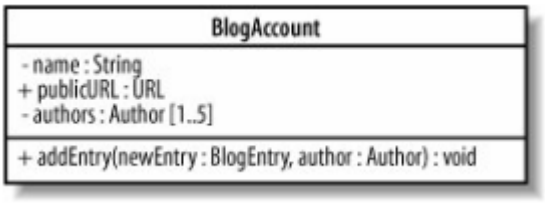
Sequence Diagram menunjukan interaksi antar diagram yang berkolaborasi dalam beberapa sifat. Sebuah sequence diagram secara khusus menjabarkan sifat skenario tunggal [11]. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek - objek ini di dalam Use case. Berikut contoh bentuk dari sequence diagram :



Gambar 2. 11 Contoh Sequence Diagram

2.13.4 Class Diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. *Class diagram* juga menunjukan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut [12]. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah class.



Gambar 2. 12 Contoh Class Diagram

2.14 Android

Android merupakan sistem operasi mobile yang berbasis Linux. Perkembangannya saat ini telah menjadikan android menjadi platform yang sangat cepat dalam melakukan inovasi [13]. Setelah diakuisisi oleh Google, android kini menjadi *platform* yang terdiri dari sistem operasi Linux dan juga GUI (*Graphical User Interface*) dimana memudahkan untuk para pengembang untuk membuat aplikasi berbasis android.

2.14.1 Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi termasuk Google, HTC, Intel Motorola, Qualcomm, T-Mobile, dan Nvidia [14].

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode – kode Android di bawah lisensi Apache, Sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Di dunia ini terdapat dua jenis distributor system operasi android. Pertama yang mendapat dukungan penuh dari Google atau Google *Mail Services* (GMS) dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan android sebagai sistem operasinya. Telepon seluler ini diproduksi oleh HTC *Corporation* dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja android ARM Holdings, Atheros *Communications*, diproduksi oleh Asustek

Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dab Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. Sejak android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bus dan penambahan fitur baru. Pada masa saat ini sebagian besar *vendor – vendor smartphone* sudah memproduksi *smartphone* berbasis android, vendor – vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Phillips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi android. Hal ini karena android itu adalah sistem operasi yang *open source* sehingga bebas disistribusikan dan dipakai oleh *vendor* manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan android selain faktor yang disebutkan di atas adalah karena android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, aplikasi dan Tool Pengembangannya, market aplikasi android serta dukungan yang sangat tinggi dari komunitas open source di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

2.14.2 Versi Android

Terdapat keunikan dari nama sistem operasi android yaitu digunakannya nama – nama makanan penutup (*Dessert*) dan setiap kenaikan versi nama – nama sistem operasinya memiliki huruf awalan berutan sesuai abjad. Berikut versi – versi android yang pernah dirilis :

1. Android 1.1

   Pertama kali dirilis adalah pada 9 Maret 2009. Awalnya versi sistem operasi android ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan update dari untuk memperbaiki beberapa bugs dan mengganti API serta menambahkan beberapa fitur lain.

2. Android 1.5 (Cupcake)

   Versi ini pertama kali dirilis pada pertengahan Mei 2009. Dimulai dari versi ini penamaan versi android mulai menggunakan nama makanan pencuci mulut (dessert). Sistem operasi ini berbasiskan kernel Linux 2.6.27 dan menambahkan beberapa update serta UI baru dari versi android yang sebelumnya.

3. Android 1.6 (Donut)

   Dirilis pertama kali pada September 2009. Memiliki peningkatan fitur pencarian dan UI yang lebih user friendly. Pada versi ini sudah mendukung teknologi CDMA/EVDO, 802.1x, VPNs kemudian support layar dengan resolusi WVGA.

4. Android 2.0/2.1 (Éclair)

   Versi ini dirilis pertama kali pada 3 Desember 2009. Terjadi penambahan fitur untuk pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5. Beberapa versi updatenya antara android 2.0 kemudian 2.0.2 dan terakhir 2.1.

5. Android 2.2 (Froyo : Frozen Yoghurt)

   Dirilis pertama kali pada bulan Mei 2010 pada smartphone Google Nexus One. Pada versi ini sudah support terhadap Adobe Flash Player 10.1, peningkatan pada kecepatan membuka dan menutup aplikasi serta penggunaan SD Card sebagai media penyimpanan aplikasi.

6. Androidd 2.3 (Gingerbread)

   Pertama kali diperkenalkan pada Desember 2010. Dirancang untuk memaksimalkan kemampuan aplikasi dan game serta mulai digunakannya Near Field Communication (NFC). Memiliki perbaikan terhadap dukungan layar dengan resolusi WXGA dan diatasnya.

7. Android 3.0/3.1 (Honeycomb)

   Dirilis pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Versi ini didesain khusus untuk pengoptimalan penggunaan tablet PC.

8. Android 4.0 (Ice Scream Sandwidch)

   Dirilis pada 19 Oktober 2011 dan pertama kali digunakan pada smartphone Samsung Galaxy Nexus. Secara teori semua perangkat android versi sebelumnya, Gingerbread dapat di-update ke android 4.0 Ice Scream Sandwidch.

9. Android 4.1 (Jelly Bean)

   Android Jelly Bean dibuat dengan memiliki keunggulan dan fitur baru. Penambahan baru itu diantaranya adalah meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru serta pencarian melalui Voice Search yang lebih cepat. Android Jelly Bean ini pertama kali digunakan pada produk tablet Asus yaitu Google Nexus7.

2.14.3 Arsitektur Android

Arsitektur android terdiriri dari beberapa lapisan yang mendukung fungsi–fungsi spesifik dari sistem operasi. Berikut lapisan – lapisan arsitektur android :

1. *Applications and Widgets*

   Merupakan layer dimana kita berhubungan dengan aplikasi saja. Biasanya dengan mendownload aplikasi kemudian dilakukan instalasi dan menjalankan aplikasi tersebut. Pada layer terdapat aplikasi ini seperti klien email, program SMS, kalender, peta, browser, kontak dan sebagainya dan semua aplikasi dibuat dengan bahasa pemrograman Java.

2. Applications Frameworks

   Android termasuk kedalam *Open Development Platform* yang menawarkan kepada pengembang untuk membangun aplikasi bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi

resources, menjalankan service background , mengatur alarm, menambah status notification dan lain – lain. Contoh dari *service* yang berjalan di background adalah *Asyncronous Task* yang merupakan sebuah *class* abstrak yang digunakan untuk melakukan operasi pada background dan menampilkannya di UI. Pengembang juga memiliki akses menuju API *framework* seperti yang dilakukan aplikasi pada kategori inti.

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

a. Views
b. Content Provider
c. Resource Manager
d. Notification Manager
e. Activity Manager

3. Libraries

Merupakan layer dimana fitur – fitur android berada. Android menyediakan satu set *library-library* dalam bahasa C/C++ yang dapat digunakan oleh berbagai komponen android. Berikut beberapa library yang ada di android :

a. *Libraries* media utnuk pemutaran media gambar, audio maupun video.
b. *Libraries* untuk manajeman tampilan
c. *Libraries Graphics* SGL dan OpenGL untuk grafis 2D dan 3D
d. *Libraries* SQLite untuk dukungan database
e. *Libraries* SSL dan WebKit terintegrasi dengan web browser dan security.
f. *Libraries* LiveWebcore mencakup modern web browser dengan engine embedded web view.
g. *Libraries* 3D untuk implementasi OpenGL ES 1.0 API's.

4. Android *Runtime*

Merupakan layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) adalah mesin yang membentuk dasar kerangka aplikasi android. Terdapat dua bagian dalam android Runtime, yaitu :

a. Core Libraries

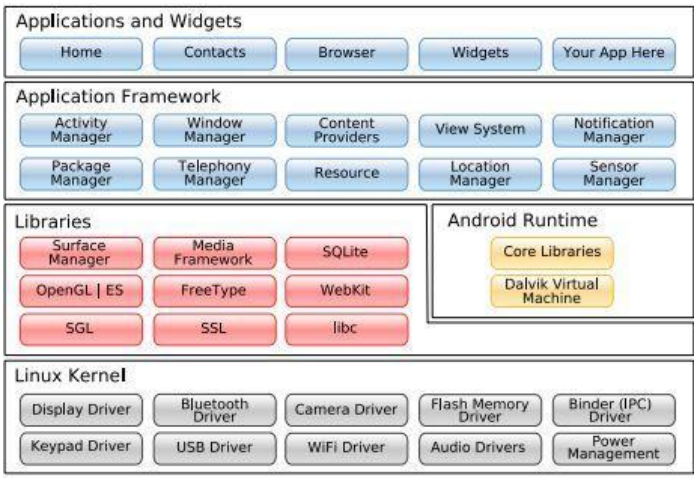
Aplikasi android dibuat menggunakan bahasa pemrograman Java. Dalvik Virtual Machine bukan merupakan Virtual Machine Java sehingga dibutuhkan sebuah libraries yang dapat menterjemahkan bahasa Java/C yang ditangan oleh bagian Core Libraries ini.

b. Dalvik Virtual Machine (DVM)

Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi – fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

c. Linux Kernel

Merupakan layer yang dimana inti dari system operasi android itu berada. Berisi file – file sistem yang mengatur sistem processing, memory, resources, drivers dan sistem – sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6. Untuk lebih mudah arsitektur android dapat dilihat pada gambar 2.13:



Gambar 2. 13 Arsitektur Android

2.14.4 Komponen Android

Aplikasi android dibangun menggunakan bahasa pemrograman Java. Untuk membuat aplikasi android terdapat beberapa komponen utama didalamnya, yaitu :

1. Activites

   Merupakan suatu tampilan (*User Interface*) untuk pengguna melakukan interaksi dengan aplikasi android. Sebuah aplikasi android biasanya memiliki lebih dari satu *activity* tergantung tujuan dan desain aplikasi tersebut. Untuk berpindah dari satu activity ke *activity* lain dapat dilakukan dengan suatu event seperti klik tombola tau menggunakan trigger. Dalam hirarki *activity* dinyatakan dengan method *Activity*.setContentView().

2. Service

   Service tidak memiliki *Graphic User Interface* (GUI) tetapi *service* berjalan secara *background*. Sebagai contoh media player sedang memutar musik dari daftar lagu yang ada, aplikasi ini akan memiliki dua atau lebih activity yang memungkinkan pengguna untuk melakukan aktifitas lain seperti menulis sms selama musiknya masih berjalan. Untuk melakukan itu, *activity* media *player* dapat menjalankan *service* dimana *service* tersebut akan dijalankan pada thread utama dari proses aplikasi.

3. *Broadcast Receiver*

   Broadcast Receiver adalah komponen aplikasi yang menanggapi *system-wide broadcast announcements*. Beberapa contoh *broadcast* yang dikirimkan oleh sistem operasi android adalah munculnya notifikasi ketika adanya aplikasi baru yang ter-install/ter-*update*, layar mati , *booting* sistem operasi selesai, SMS atau telepon masuk, dan sebagainya. *Broadcast Receiver* juga tidak memiliki tampilan, tapi dia dapat menjalankan suatu *activity* atau menampilkan notifikasi di *Notification* Bar.

4. Content Provider

   Sebuah *content provider* mengatur sekumpulan data aplikasi yang terbagi (shared). Dengan content provider dapat menyimpan data di filesystem,

sebuah database SQLite, di web, atau di metode penyimpanan data lainnya yang bisa diakses oleh aplikasi. Melalui *content provider*, aplikasi lain bisa memberikan *query* atau bahkan bisa memodifikasi, tentunya jika content *provider* mengijinkan aksesnya. Sebagai contoh, sistem android menyediakan *content provider* yang mengatur informasi kontak user.

5. Intent

   Intent merupakan pesan Asynchronous yang memungkinkan aplikasi untuk memberikan request secara fungsional dari komponen yang berbeda – beda di dalam sistem android.

2.15 PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor (PHP) adalah bahasa pemrograman *server side* yang dirancang khusus untuk pengembangan aplikasi berbasis *web*. *Script* PHP disisipkan kedalam *script* HTML yang kemudian akan di proses pada suatu *web server* dan akan menghasilkan suatu halaman *web*. PHP termasuk kedalam *open source product* sehingga mengubah *source code* dan mendistribusikannya secara bebas.

Awalnya PHP diciptakan oleh Andi Gutmans untuk menghitung jumlah pengunjung yang mengakses homepage yang dibuatnya. Namun seiring dengan perkembangan internet, kini telah dirilis PHP/FI, PHP2, PHP3, PHP4, dan selanjutnya sudah ada PHP5 yang mampu digunakan untuk membangun aplikasi web dengan koneksi basis data yang cukup banyak. PHP5 adalah versi pengembangan dari PHP4 dengan menambahkan fungsi – fungsi seperti Zend Engine sehingga akses lebih cepat, kuat, stabil dan mudah berinteraksi dengan berbagai aplikasi pendukung lainnya [15].

PHP banyak mendukung basis data, seperti MySQL, *PostgresSQL*, *Interbase*, ODBC, mSQL, *Oracle* dan *Sybase*. Kini PHP4 banyak digunakan oleh para *web developer* untuk membangun aplikasi *web* karena memang terbukti dapat bekerja dengan baik.

Untuk menjalankan sistem PHP dibutuhkan 3 komponen utama, diantaranya :

a. *Web Server*
b. *Script PHP*
c. *Database Server*

2.15.1 *Script* PHP

Penulisan *script* PHP diawali dan diakhiri dengan sintaks khusus. Ada beberapa cara yang dapat digunakan untuk menuliskan *script* PHP, yaitu :

1. Dengan sintaks “<?PHP” dan “?>”; sintaks tersebut adalah yang paling umum dari PHP.
2. Dengan sintaks “<%” dan “%>”; sintaks tersebut sebenarnya adalah sintaks dari ASP, tetapi dapat digunakan sebagai sintaks PHP apabila konfigurasi PHP pilihan asp_tag diaktifkan.
3. Dengan sintaks “<SCRIPT LANGUAGE = “php” dan diakhiri dengan “</SCRIPT>”; sintaks itu digunakan untuk mendeklarasikan sebuah perintah pada html.

2.15.2 Kelebihan PHP

PHP memiliki kelebihan yang tidak dimiliki oleh bahasa – bahasa lain yang sejenis. PHP mudah dibuat dan cepat dijalankan. PHP dapat berjalan dalam *web server* yang berbeda dan dalam system operasi yang berbeda pula. PHP dapat berjalan di system operasi *UNIX*, Windows 98, Windows NT dan *Machintosh*.

PHP diterbitkan secara gratis dan dapat mendownload *source code* PHP tanpa perlu mengeluarkan uang. PHP juga dapat berjalan pada server Microsoft *Personal Web Server*, *Apache*, IIS dan sebagainya. PHP juga termasuk bahasa yang *embedded* (bisa diletakkan di dalam HTML).

PHP yang ditulis dengan menggunakan bahasa C dapat di kembangkan sendiri. Jika telah menguasai bahasa C, dapat dengan mudah menambah fungsi – fungsi baru. PHP adalah bahasa yang sangat popular. Sebagian besar sintaks PHP mirip dengan sintaks dalam bahasa C dan Perl. [15]

2.16 MySQL

MySQL atau yang sering dibaca dengan ejaan "*mai es que el*" ini adalah sebuah *database* yang berbasiskan *server database*. Kemampuannya dalam menangani RDBMS (*Relasional Database Management System*) mengakibatkan *database* ini menjadi *database* yang sangat popular saat ini. *Database* MySQL mampu menangani data yang sangat besar hingga ukuran Giga Byte, dengan kemampuan daya tampung data ini maka MySQL sangat cocok digunakan untuk mengcover data perusahaan baik kecil sampai perusahaaan besar [16].

2.17 *Eclipse*

Eclipse merupakan sebuah IDE (*Integrated Development Environment*) untuk mengembangkan suatu aplikasi dan dapat dijalankan di berbagai *platform* (*Platform-Independent*). *Eclipse* termasuk kedalam tools yang bersifat *open source* [17]. Eclipse memiliki beberapa sifat diantaranya adalah sebagai berikut :

1. *Multi-platform*

   Target *system* operasi eclipse ini adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. *Multi-language*

   Dikembangkan dengan bahasa pemrograman Java tetapi eclipse juga mendukung pengembangan aplikasi dengan bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP dan lain – lain.

3. *Multi-role*

   Eclipse dapat digunakan untuk aktifitas dalam siklus pengembangan aplikasi seperti dokumentasi, pengujian, pengembangan dan sebagainya.

2.17.1 Arsitektur *Eclipse*

Eclipse merupakan sebuah kernel yang berarti bahwa *eclipse* adalah fungsi dari *pluf-in* yang sudah dipasang (di-*install*). Basis dari eclipse dinamakan *Rich Client Platform* (RCP). Berikut adalah komponen yang membentuk RCP :

1. *Core Platform*
2. *OSGi*

3. *SWT (Standard Widget Toolkit)*
4. *JFace*
5. *Eclipse Workbench*

Eclipse dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat *eclipse* dapat kompatibel untuk pengembangan program Java dan PDE (*Plug-in Development Environment*). Eclipse beserta plug-in yang dimilikinya diimplementasikan dalam bahasa pemrograman Java [18].

Eclipse tidak hanya untuk mengembangkan program Java, tetapi juga untuk berbagai macam keperluan yang dibutuhkan. Untuk melakukan perluasan apapun didalam eclipse cukup dengan meng-install *plug-in* yang dibutuhkan.

2.17.2 Versi *Eclipse*

Eclipse Foundation telah mengkoordinasikan peluncuran *Eclipse* secara rutin yang dikenal dengan nama *simultaneous Release*. Setiap versi yang diluncurkan terdiri dari *eclipse platform* dan sejumlah proyek yang terlibat langsung dalam pembuatan *proyek eclipse* [19].

Hampir setiap tahun ke tahun *eclipse* selalu dijadwalkan untuk meluncurkan versi terbarunya. Tujuannya adalah untuk menyediakan distribusi eclipse dengan fitur – fitur dan versi yang sudah terstandarisasi. Berikut berbagai versi eclipse yang sudah diluncurkan saat ini, yaitu :

Tabel 2. 1 Versi *Eclipse*

Kode Peluncuran	Tanggal Peluncuran	Platform	Nama Proyek
Eclipse 3.0	21 Juni 2004	3.0	-
Eclipse 3.1	28 Juni 2005	3.1	-
Callisto	30 Juni 2006	3.2	Callisto Project
Europa	29 Juni 2007	3.3	Europa Project
Ganymede	25 Juni 2008	3.4	Ganymede Project
Galileo	24 Juni 2009	3.5	Galileo Projecy
Helios	23 Juni 2010	3.6	Indigo Project
Juno	27 Juni 2012	4.2	Juno Project

2.18 *Software Development Kit* (SDK)

Software Development Kit (SKD) adalah sekumpulan *tools* yang yang memungkinkan untuk membuat dan menjalankan aplikasi. SDK juga merupakan kerangka kerja perangkat lunak, platform perangkar keras, sistem komputer, konsol video game, sistem operasi dan sebagainya.

2.19 *Development Kit* (JDK)

Java Development Kit (JDK) merupakan produk dari *Sun Microsystems* yang ditujukan untuk pengembang *Java*. Sejak Java diperkenalkan, hingga saat ini banyak yang telah menggunkan Java SDK. Pada tanggal 17 November 2006 Sun *Microsystems* mengumumkan bahwa akan dirilis JDK dibawah lisensi GNU (*General Public License*) sehingga membuat penggunaannya menjadi gratis.

2.20 Metode Pengujian Sistem

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.20.1 Pengujian *Blackbox*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut[20]:

1. Fungsi tidak benar atau hilang.
2. Kesalahan *interface* atau antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.20.2 Skala Likert

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena [21].

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negative yang berfungsi untuk mengukur sikap negative objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (TS), 2 untuk tidak setuju (S), 3 untuk raguragu (CS), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. Skala Likert digunakan untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan Skala Likert :

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan Skala Likert :

1. Mudah dibuat dan di terapkan.

2. Skala Likert lebih mudah membuatnya dibanding lain sepertti skala Thurstone.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa alternative, sehingga informasi mengenai item tersebut diperjelas.