

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori memberikan gambaran dan teori terkait dengan membangun sistem Aplikasi *Live Score Flag Football*. Untuk membangun sebuah sistem, maka perlu dikemukakan hal-hal atau teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan tugas akhir ini.

2.1.1 *Flag Football*

Flag Football adalah salah satu variasi dari American Football. Analoginya, *Flag Football* bisa dibilang sebagai futsalnya *American Football*. Terdapat beberapa perbedaan antara *Flag Football* dengan *American Football*, namun yang paling mendasar adalah teknik *tackle* (menjatuhkan) yang digantikan dengan sebuah teknik yang dinamakan *deflagging* (melepas bendera / flag dari tubuh pemain lawan). Pada pertandingan *American Football*, pembawa bola dinyatakan *down* (permainan berhenti sementara karena lutut atau tangan pembawa bola jatuh mengenai permukaan tanah) dan permainan berhenti apabila pemain bertahan atau defender berhasil menjatuhkan pembawa bola hingga lutut atau tangannya menyentuh tanah. Pada *Flag Football*, pembawa bola (*ball carrier*) dinyatakan "down" dan permainan berhenti apabila defender (pemain bertahan) berhasil mencabut setidaknya satu bendera (*flag*) yang terpasang di pinggang pembawa bola tersebut.

Permainan *flag football* dimainkan diatas lapangan berukuran 80x50 yards. Biasanya permainan ini dilakukan di lapangan yang berumput, bis itu lapangan bola biasa atau lapang berumput lainnya. Sebagai pembatas, setiap 10 yard akan dipasang pembatas. Pembatas yang digunakan biasanya adalah *traffic cone* (pembatas jalan yang berbentuk kerucut dan berwarna jingga).

Setiap tim yang bermain di atas lapangan harus terdiri dari 8 pemain. Pembagian posisi pemain dibebaskan kepada tim yang bermain sesuai dengan strategi apa yang akan mereka gunakan. Permainan akan berlangsung selama 2x20 menit. Pertandingan flag football terdiri dari beberapa set. Setiap set yang dimainkan, salah satu tim akan bertindak sebagai *attacker* (tim yang menyerang) dan lawannya akan bertindak sebagai *defender* (tim yang bertahan). Tim *attacker* akan diberi kesempatan 4 kali menyerang. Ketika tim *attacker* telah 4 kali *down*, maka tim *defender* akan berganti peran menjadi *attacker*, peraturan ini terus berlaku hingga waktu 2x20 menit berakhir.

Permainan *flag football* menggunakan bola yang berbentuk lonjong. Pemegang bola pertama ketika *kick-off* (pertandingan dimulai) yaitu ada pada pemain yang berposisi sebagai *deffensive lineman*. Bola dipegang dengan cara menempelkan bola secara tegak lurus menyentuh tanah. Posisi semua pemain harus berada di tengah-tengah lapangan. Ketika peluit pertandingan dibunyikan, pemain yang memegang bola akan memberikan bola kepada pemain di posisi *quarterback* yang berdiri di belakangnya dengan cara memberikan bola melewati kolong diantara kedua kaki *defensive lineman*.

Peraturan selanjutnya yaitu dalam satu kali kesempatan menyerang, setiap tim hanya boleh melakukan 1 kali operan/perpindahan bola terhitung dari ketika bola sudah dipegang oleh *quarterback*. Operan bisa dilakukan dengan cara hanya memberikan secara langsung kepada pemain lain atau dengan cara dilempar menuju pemain lain. Pertandingan akan terus berlanjut hingga siapapun pemain pemegang bola telah berhasil sampai pada area *endzone* lawan (10 yard terakhir yang berada pada area permainan lawan) dan dinyatakan *touchdown* (istilah gol dalam *flag football*). Dan apabila ketika permainan sedang berlangsung si pemegang bola berhasil dicabut *flag* (bendera) yang ada pada tubuhnya maka dinyatakan *down* (permainan berhenti sementara karena lutut atau tangan pembawa bola jatuh mengenai permukaan tanah) dan dilanjutkan ke kesempatan berikutnya. Kesempatan selanjutnya akan dimulai di area dimana pemain sebelumnya berhasil di *deflagging* (tercabutnya bendera yang ada di bagian tubuh pemain).

2.1.1.1 Sejarah *Flag Football*

Flag football awalnya dinamai *Touch and Flag Football*. Olahraga ini pertama kali dimainkan pada tahun 1930an. Awalnya olahraga ini diciptakan oleh para anggota militer Amerika Serikat sebagai olahraga hiburan mereka. Seiring berjalannya waktu, pada rentang waktu 1960 hingga 1970, olahraga ini selalu dilihat dengan nama *National Collegiate Flag Football Championship* yang secara rutin diselenggarakan setiap bulan di *New Orleans University* dan diikuti oleh beberapa universitas pada masa itu.

Kompetisi mayor pertama dibentuk pada tahun 1988 oleh Mike Cihon dan diberi nama USTFL (*United States Flag Touch Football League*). USTFL merupakan suatu kompetisi terbesar di Amerika Serikat yang selalu diselenggarakan di Orlando, pada tahun 2002 kompetisi ini menyelenggarakan turnamen dengan jumlah 175 tim. Pada tahun 1999, PFFL (*Professional Flag Football League*) dibentuk dan diselenggarakan hingga sekarang.

Di Indonesia, IFFA kependekan dari Indonesian Flag Football Association, saat ini adalah organisasi penyelenggara kegiatan flag football dan wadah pecinta football terbesar di Indonesia. IFFA berdiri sejak 14 Februari 2009, sebagai kelanjutan dari Indonesian Flag Football League yang vakum sejak 2001. Saat ini IFFA sudah memiliki kepengurusan yang tetap dan anggota yang cukup banyak. IFFA memiliki kepengurusan di Jakarta, dan bermain di Hockey Field Senayan.

2.1.1.2 Peraturan Dalam *Flag Football*

Di Indonesia, *Flag Football* diselenggarakan dengan menggunakan peraturan resmi (*Official Rulebook*) yang dikeluarkan oleh IFFA (*Indonesian Flag Football Association*). Aturan tersebut dapat dilihat secara lengkap di bab lampiran pada sub bab *Rulebook Flag Football*.

2.1.2 Teori Dasar

Sub bab ini membahas semua landasan teori yang berkaitan dengan pembangunan aplikasi live score flag football ini.

2.1.2.1 *Live Score*

Live Score adalah jenis layanan yang ditawarkan oleh banyak situs web dan penyiar yang terkait dengan olahraga serta operator taruhan olahraga online. Tujuan utama dari *Live Score* adalah memberikan informasi secara *real time* tentang hasil olahraga dari berbagai disiplin ilmu/olahraga. Informasi *Live Score* biasanya diberikan secara gratis dan sangat populer di kalangan penggemar taruhan olahraga, karena mereka memungkinkan untuk melihat data yang dikumpulkan pada banyak acara olahraga. Di masa lalu, layanan *live score* hanya tersedia di TV melalui teleteks atau di radio. Sekarang ada banyak situs yang menyediakan *Live Score*. Hal ini memungkinkan untuk mendapatkan informasi hasil pertandingan secara langsung dari banyak pertandingan pada saat bersamaan. Beberapa situs memberikan informasi tambahan, seperti daftar pemain, detail kartu, substitusi dan obrolan *online* di mana penggemar olahraga dapat mengumpulkan dan mendiskusikan acara yang sedang berlangsung.

2.1.2.2 Aplikasi Mobile

Menurut Buyens aplikasi mobile berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, dan penggunaan [2]. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau

aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Maka aplikasi *mobile* dapat diartikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat diakses melalui perangkat nirkabel, pager, PDA, telepon seluler, *smartphone*, dan perangkat sejenisnya.

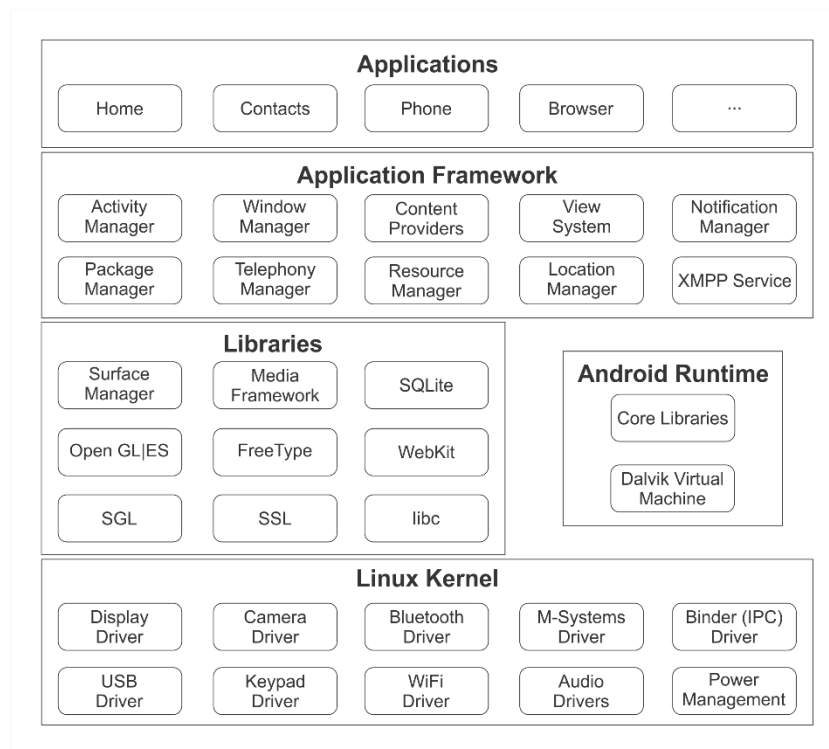
2.1.2.3 Android

Android merupakan sistem operasi mobile yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainnya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan diatas perangkat hardware yang ada. Akan tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga.

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

2.1.2.3.1 Arsitektur Android

Android terdiri dari beberapa stack software yang terdiri dari: *Applications*, *Application Framework*, *Libraries*, *Android Runtime* dan *Kernel Linux*. Arsitekturlengkap platform ini dapat dilihat pada gambar berikut.



Gambar 2.1 Struktur Android

1. Application

Application merupakan program yang langsung berhubungan dengan *user*. Baik program yang merupakan bawaan dari *Android* sendiri maupun program yang dibuat oleh *developer* menggunakan bahasa pemrograman *java*.

2. Application Framework

Lapisan ini berisi sekumpulan API yang dapat digunakan oleh *programmer* maupun *core application* dari *android*. Lapisan ini dirancang untuk memudahkan pengguna komponen dari *android* sendiri. Aplikasi maupun dalam *android* dapat berbagi fungsi sehingga aplikasi lain dapat memanfaatkannya.

3. Libraries

Android mendukung beberapa library C/C++ yang digunakan pada berbagai komponen Android. Kemampuan ini dapat diakses oleh *developer* melalui Android application framework. Beberapa library diantaranya adalah: a. System C library. Implementasi library C standar (*libc*).

- b. Media Libraries. Mendukung berbagai format multimedia (termasuk MPEG4,H.264, MP3, AAC, AMR, JPG, PNG).
- c. Surface Manager. Mengatur akses ke subsistem display.
- d. LibWebCore. Engine web browser modern.
- e. SGL. Engine grafis 2D.
- f. 3D Library. Implementasi OpenGL ES 1.0 yang mendukung akselerasi hardware.
- g. FreeType. Rendering untuk bitmap dan vector font.
- h. SQLite. Basis data relasional yang kecil namun sangat ampuh.

4. Android Runtime

Tiap aplikasi pada Android memiliki proses-nya masing-masing. Tiap aplikasi tersebut memiliki instans dari Dalvik virtual machine (VM). Dalvik virtual machine dirancang agar suatu device dapat menjalankan beberapa VM secara efisien. Dalvik VM mengeksekusi file dengan format Dalvik Executable format (.dex) yang dirancang untuk meminimalkan memory footprint .

Dalvik VM berbasis register, dan dapat menjalankan kelas-kelas yang dikompilasi dengan bahasa pemrograman java dan ditransformasikan menjadi format .dex. Dalvik VM sendiri bergantung pada Kernel Linux untuk fungsi dasarnya, seperti threading dan manajemen memori secara low-level.

5. Linux Kernel

Android menggunakan Kernel Linux versi 2.6 sebagai sistem utama. Fungsi kernel yang digunakan antara lain untuk keamanan, manajemen memori, manajemen proses, manajemen jaringan dan driver model. Kernel juga berfungsi sebagai layer abstrak antara hardware dan lapisan lainnya pada software stack.

2.1.2.3.2 Sejarah Android

Pada Juli 2000, *Google* bekerjasama dengan *Android Inc.*, perusahaan yang berada di Palo Alto, California, Amerika Serikat. Para

pendiri *Android Inc.* bekerja pada *Google*, diantaranya Andy Rubin, Rich Miner, Nick Sears dan Chris White. Saat itu banyak yang menganggap fungsi *Android Inc.* hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa *Google* hendak memasuki pasar telepon seluler. Di perusahaan *Google*, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh *Kernel Linux*. Hal ini menunjukkan indikasi bahwa *Google* sedang bersiap menghadapi persaingan dalam pasara telepon seluler. Versi *android* terbaru yaitu versi 5.0. (*KitKat*). *Android* juga sudah bergabung dengan beberapa *smart mobile* seperti *LG*, *Samsung*, *Sony Ericsson*, *HTC*, *Motorola* dan lainnya.

Sekitar September 2007 sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler (akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar GSM yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporotaion* dan tersedia di pasaran pada 5 Januari 2010).

Pada Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc.* *Garmin Ltd*, *Softbank*, *Sony Ericsson Toshiba Corp* dan *Vodafone Group Plc*. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, *Android*, perangkat bergerak (*Mobile*) yang merupakan modifikasi *kernel Linux 2.6*. sejak *Android* dirilis telah dilakukan berbagai pembaruan berupa perbaikan *bug* dan penmbahan fitur baru. [3]

2.1.2.3.3 Versi Android

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android*

adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi *android* seperti dibawah ini :

1. Android Versi 1.1

Pada 9 Maret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan Email.

2. Android Versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan pada CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

4. Android Versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah mengoptimalkan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan

UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth 2.1*. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps*-aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah Shazam, *Backgrounds* dan *WeatherBug*. Sistem operasi *Android* dalam situs internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *MySpace* dan *Facebook*.

5. Android Versi 2.2 (*Froyo : Frozen Yoghurt*)

Pada 20 Mei 2010, *Android* versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi dengan V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel dan kemampuan *auto update* dalam aplikasi Android Market.

6. Android Versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, *Android* versi 2.3 (*Gingerbread*)
Perubahan-perubahan umum yang didapat dari versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM. efek audio baru (*reverb, equalization, headphone virtualization, dan bass boost*), dukungan kemampuan *Near Field Communication* (NFC) dan dukungan jumlah kamera yang lebih dari satu.

7. Android Versi 3.1 (*Honeycomb*)

Android Honeycomb dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet peratam yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

8. Android Versi 4.0 (*Ice Cream Sandwich*)

Pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline* dan berbagi informasi dengan menggunakan NFC.

9. Android Versi 4.1 (*Jelly Bean*)

Pada tanggal 27 Juni 2012 Google mengumumkan *Android* versi 4.1 (*Jelly Bean*) dalam konferensi Google I/O berdasarkan kernel Linux 3.0.31. *Jelly Bean* adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuak pengguna (*User Interface*). Pembaruan ini diwujudkan dalam proyek *Butter*, perbaikan ini termasukantisipasi sentuh, *triple buffering*, perpanjangan waktu vsync dan peningkatan *frame rate* hingga 60fps untuk menciptakan UI yang lebih halus. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

10. Android Versi 4.4 (*KitKat*)

Android versi 4.4 (*KitKat*) dirilis Oktober 2013. Setelah sebelumnya beredar rumor bahwa *Android* versi berikutnya setelah *Jelly Bean* diperkirakan akan diberi nomor 5.0 dan dinamai *Key Lime Pe*.

11. Android Lollipop Version 5.0

Android Lollipop adalah versi stabil terbaru dengan versi antara 5.0 dan 5.1. Diresmikan pada 25 Juni 2014 saat Google I / O, dan tersedia secara resmi melalui over-the-air (OTA) update pada tanggal 12 November 2014, untuk memilih perangkat yang menjalankan distribusi Android dilayani oleh Google (seperti perangkat Nexus dan Google Play edition).

2.1.2.4 DBMS

DBMS (Database Management System) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani backup data.

2.1.2.5 Konsep Dasar Berorientasi Object

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata [4].

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek akan dikapsulasi (dihubungkan) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dinkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (model/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*Object*)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi akan dibuat sebuah objek. Objek dilihat dari segi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehingga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*Method*)

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Contoh metode atau operasi misalnya *Read*, *Write*, *Move*, *Copy*, dan sebagainya.

4. Atribut (*Attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*Abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*Encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*Inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*Interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

Komunikasi antarobjek dilakukan lewat pesan (message) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*Polymorphism*)

Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

2.1.2.6 GPS (Global Positioning System)

Global Positioning System adalah sistem navigasi berbasis satelit terdiri dari jaringan 24 satelit ditempatkan ke orbit oleh Departemen Pertahanan Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978. Layanan GPS dahulu hanya dipergunakan untuk keperluan militer namun mulai terbuka untuk publik. 24 satelit GPS tersebut berada sekitar 12.000 mil di atas bumi bergerak mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, memiliki baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari dan memiliki roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS *receiver*, sehingga harus selalu dipelihara agar posisinya tepat. Posisi satelit-satelit tersebut selalu dipantau oleh stasiun pengendali. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS *receiver* mengambil informasi tersebut dan melakukan perhitungan triangulation untuk menentukan lokasi pengguna dengan tepat. GPS receiver membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit. Dengan mengetahui jarak tersebut, GPS receiver dapat melakukan perhitungan dan menentukan posisi pengguna dan menampilkan dalam peta elektronik. Setelah menentukan posisi pengguna, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS sangat presisi dalam mengirim informasi karena satelit tersebut menggunakan jam atom. Jam atom yang ada pada satelit

merupakan partikel atom yang diisolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Keistimewaan GPS adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat GPS bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi Wide Area Augmentation System (WAAS) keakuratannya sampai 3 meter. Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan di tempat terbuka.

Penggunaan di dalam ruangan, atau di tempat yang menghalangi arah satelit (di angkasa), maka GPS tidak akan bekerja secara akurat dan maksimal. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi. Semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data posisi yang dihasilkan. Selain itu, ketinggian juga mempengaruhi proses kerja GPS, karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit.

Sinyal yang diterima oleh alat penerima (receiver) di permukaan, dimana GPS receiver ini akan mengumpulkan informasi dari satelit GPS, seperti :

- a. Waktu, GPS receiver menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
- b. Lokasi, GPS memberikan informasi lokasi dalam tiga dimensi:
 1. Latitude
 2. Longitude
 3. Elevasi
- c. Kecepatan, ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
- d. Arah perjalanan, GPS dapat menunjukkan arah tujuan.
- e. Simpan lokasi, tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS receiver.

- f. Komulasi data, GPS receiver dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya.

2.1.2.7 LBS (Location Based-Service)

Salah satu ciri mobile phone atau tablet pc adalah portabilitas, sehingga tidak mengherankan bahwa beberapa fitur Android sangat menarik seperti layanan yang memungkinkan anda menemukan, mencari serta memvisualisasikan posisi kita ke dalam peta lokasi fisik seperti GoogleMaps. Kita dapat membuat peta berbasis GoogleMaps dan menjadikannya sebagai element dalam layout User Interface (UI) yang kita rancang. Kita dapat melakukan akses penuh ke peta GoogleMaps, dan memungkinkan kita untuk mengontrol pengaturan tampilan, mengubah tampilan zoom lokasi, dan memindahkan lokasi tampilan.

Location Based Service adalah service yang berfungsi untuk mencari dengan teknologi Global Positioning Service (GPS) dan Google's cell-based location. Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis, namun sebagai user kita membutuhkan alamat atau posisi realtime kita bukan nilai lintang dan bujur. Android menyediakan geocoder yang mendukung forward dan reverse geocoding. Menggunakan geocoder, anda dapat mengkonversi nilai lintang bujur menjadi alamat dunia nyata atau sebaliknya.

Location based service 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:

- a. Location Manager (API Maps)

Menyediakan tools/resource 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`.

b. Location Providers (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh device/perangkat. API Location berhubungan dengan data GPS dan data lokasi realtime. API Location berada pada paket android yaitu dalam paket `android.location`. Dengan Location Manager, kita dapat menentukan lokasi kita saat ini, Track gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

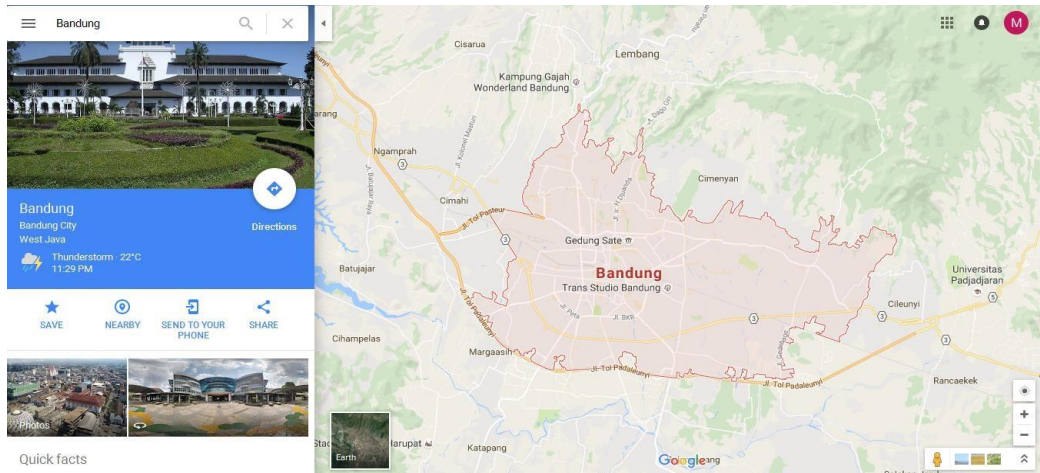
2.1.2.8 Google Maps

GoogleMaps adalah peta online atau membuka peta secara online, dapat dilakukan secara mudah melalui layanan gratis dari Google. Bahkan layanan ini menyediakan API (Application Programming Interface) yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buaatannya. Tampilan GoogleMaps pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.

GoogleMaps adalah layanan gratis yang diberikan oleh Google dan sangat populer. GoogleMaps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah seperti pada gambar 2.3. Dengan kata lain, GoogleMaps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur GoogleMaps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan GoogleMaps API. GoogleMaps API adalah suatu library yang berbentuk JavaScript.

Cara membuat GoogleMaps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi Internet yang sangat stabil. Dengan menggunakan GoogleMaps API, kita dapat menghemat waktu dan biaya

untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia.



Gambar 2.2 Google Maps

Pada GoogleMaps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah :

- a. ROADMAP, untuk menampilkan peta biasa 2 dimensi.
- b. SATELITE, untuk menampilkan foto satelit.
- c. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
- d. HYBRID, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota).

2.1.2.9 Google Geocoding

Google Maps sudah memiliki fitur yang cukup lengkap, termasuk mencari lokasi kota pada peta (*geocoding*). *Geocoding* adalah proses menemukan koordinat geografis, yaitu bujur dan lintang, dari data geografis lain, misalnya alamat. Fitur *geocoding* dapat dimanfaatkan untuk

menambahkan informasi *latitude* dan *longitude* pada foto yang tidak memiliki data *geotagging*.

2.1.2.10 UML (*Unified Modeling Language*)

Unified Modeling Language adalah keluarga notasi grafis yang didukung oleh mera-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP) [5].

Pemodelan visual adalah suatu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata. Model berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (customer, ahli dibidangnya, analis, desainer, dan lain-lain), memodelkan perusahaan, menyiapkan dokumentasi, merancang program, dan merancang basis data.

2.1.2.10.1 Use Case Diagram

Usecase diagram digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (or bagian sistem: *subsistem* atau *class*) ke pemakai. *Usecase* dapat dilingkupi dengan batasan sistem yang diberi label nama sistem. *Usecase* adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

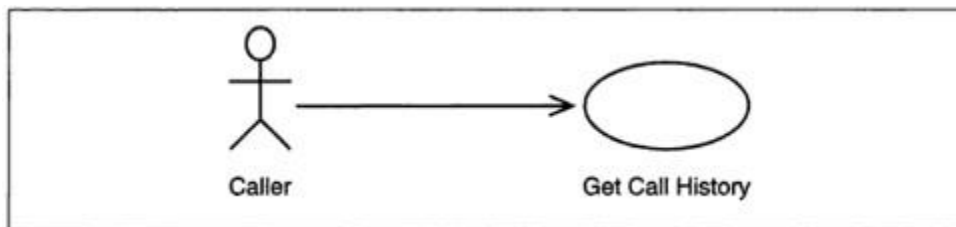
Usecase: aktor, dan asosiasi mereka dapat ditampilkan pada diagram use case. Satu diagram ini, kita melihat seorang aktor yang disebut "pelanggan" dan usecase disebut "*get call history*" arah panah menunjukkan bahwa pelanggan memulai komunikasi.

Use case Diagram memberikan pandangan dan memberikan perspectives yang berbeda pada model usecase tunggal. diagram *usecase* mungkin hanya berisi aktor, hanya usecase, atau kombinasi dari keduanya. umum digunakan usecase diagram adalah sebagai berikut:

1. Diagram gambaran yang menunjukkan semua usecase dan aktor

2. Diagram ringkasan aktor menampilkan serangkaian aktor terkait konseptual
3. Aktor diagram perspektif menunjukkan semua usecase melibatkan aktor
4. Diagram ringkasan penggunaan-kasus yang menunjukkan seperangkat usecase konseptual terkait
5. Usecase diagram perspektif menggambarkan bagaimana usecase yang berhubungan dengan aktor dan kasus pengguna lain yang patut.

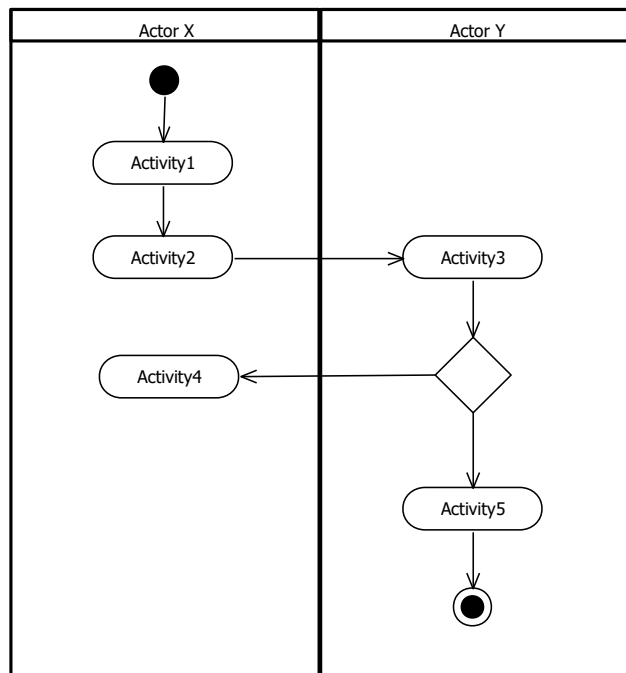
Usecase dibentuk untuk merepresentasikan bagaimana suatu sistem bekerja atau berjalan. Suatu sistem tidak lepas dari bagian-bagian tertentu untuk membentuk sistem tersebut dibentuk .



Gambar 2.3 Contoh Usecase diagram sederhana

2.1.2.10.2 Activity Diagram

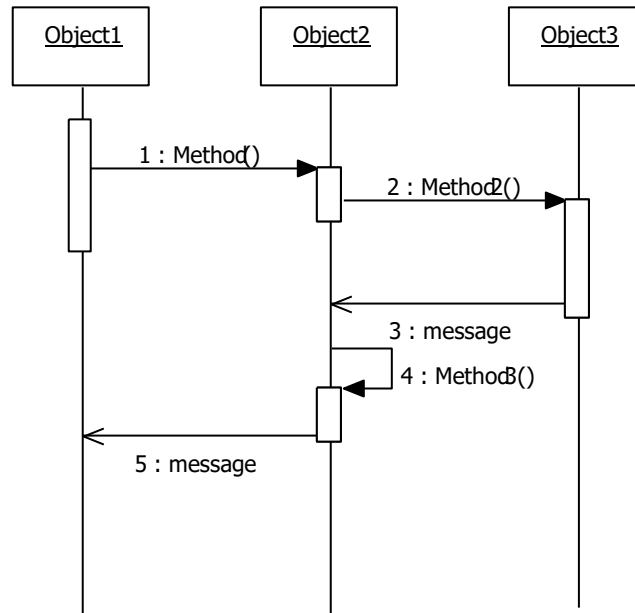
Diagram Aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa *diagram* aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. Dibawah ini merupakan *activity diagram* gambar berikut :



Gambar 2.4 Contoh Bentuk *Activity Diagram*

2.1.2.10.3 Sequence Diagram

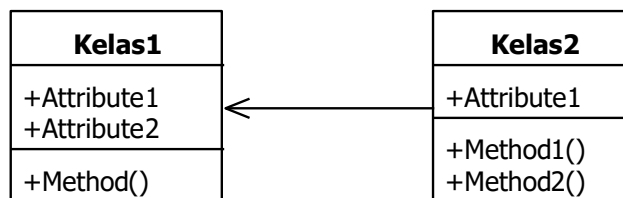
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Dibawah ini merupakan *sequence diagram* gambar berikut.



Gambar 2.5 Contoh Bentuk *Sequence Diagram*

2.1.2.10.4 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Pada Gambar 2.8 adalah contoh *class diagram* untuk menggambarkan struktur sistem yang akan dibuat untuk membangun sistem.



Gambar 2.6 Contoh Bentuk *Class Diagram*

2.2.11 Google Rest Api

Rest Api adalah suatu jenis *web service* yang menerapkan konsep perpindahan antara *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perpindahan HTTP yang bisa digunakan adalah fungsi GET, POST, PUT atau DELETE. Balasan yang dikirim adalah bentuk XML sederhana tanpa ada protokol pemetaan data, sehingga informasi yang diterima lebih mudah dibaca dan dipasang disisi *client*.

2.2.12 Simple Additive Weighting (SAW)

Metode Simple Additive Weighting (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Pahlevy. 2010). Metode SAW membutuhkan proses normalisasi matriks keputusan X ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut (Kusumadewi, Harjoko, dan Wardoyo. 2006):

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana:

r_{ij} = rating kinerja ternormalisasi dari alternatif A_i ($i=1,2,\dots,m$) Max_i = nilai maksimum dari setiap baris dan kolom.

Min_i = nilai minimum dari setiap baris dan kolom. x_{ij} = baris dan kolom dari matriks.

Formula untuk mencari nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai (Kusumadewi, Harjoko, dan Wardoyo.2006):

Dimana:

V_i = Nilai akhir dari alternatif W_i = Bobot yang telah ditentukan

r_{ij} = Normalisasi matriks.

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.1.3 Tools Pendukung

Dalam pembangunan sistem program terdapat beberapa *tools* perangkat pendukung yang di gunakan untuk membangun sebuah sistem ini ssebagai berikut:

2.1.3.1 Notepad ++

Notepad++ adalah sebuah text editor yang sangat berguna bagi setiap orang dan khususnya bagi para developer dalam membuat program. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi Microsoft Windows.

Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad ++ juga dilisensikan sebagai perangkat free. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena sourceforge.net sebagai layanan yang memfasilitasi Notepad ++ membebaskannya untuk digunakan. Beberapa daftar bahasa program yang didukung oleh Notepad++ adalah C, C++, Java, C#, XML, HTML, PHP, Javascript.

2.1.3.2 Processor Hypertext Protocol (PHP)

PHP adalah suatu bahasa *server-side* yang didesain khusus untuk aplikasi web [6]. PHP dapat disisipkan diantara HTML, dan karena bahasa *server-side*, maka bahasa PHP akan dieksekusi di server, yang dikirimkan ke browser adalah “hasil jadi” bentuk HTML, dan kode PHP tidak akan terlihat.

PHP dahulu merupakan proyek pribadi dari Rasmus Lerdorf tahun 1994 (dengan dikeluarkannya PHP versi 1 yang digunakan untuk membuat home page pribadi yang dipublikasikan 8 juni 1995). Versi pertama ini berupa kumpulan script PERL. Untuk versi kedua Ramus menulis ulang script-script PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk form HTML dan koneksi MYSQL. Adapun PHP didapat dari singkatan *PHP Hypertext Preprocessor*.

Setelah mengalami perkembangan, Zeev Suraski dan Andi Gutmans (termasuk Rasmus) yang tergabung dalam suatu kelompok open source, maka mereka memulai versi 3 PHP nya, PHP telah menampakkan keunggulan sebagai salah satu bahasa *server scripting* yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini PHP disebut sebagai “PHP: Hypertext Preprocessor”. Pada bulan Mei 2000 dirilis PHP versi 4 yang dikeluarkan oleh Zend Engine yaitu sebuah produk yang diciptakan oleh Suraski dan Gutmans. Kehandalan PHP diperlengkap dengan dirilisnya PHP versi 5 pada 13 Juli 2004 yang mendukung OOP serta berbagai manajemen database yang luas dan penanganan error yang mudah. Sintak yang digunakan berasal dari bahasa C, Java maupun PERL. Sampai tulisan ini dibuat versi PHP yang terbaru adalah versi 5.2.0 yang dirilis pada November 2006.

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. Oleh karena itu, spesifikasi server lebih berpengaruh pada eksekusi dari script php daripada spesifikasi client. Namun tetap diperhatikan bahwa web yang dihasilkan tentunya harus dapat dibuka oleh browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client. PHP termasuk dalam *Open Source Product*. Jadi dapat menambah *source code* dan dapat

mendistribusikannya secara bebas. PHP juga diedarkan secara gratis. Kita bisa mendapatkannya secara gratis. PHP juga dapat berjalan diberbagai web server semisal IIS, Apache, PWS.

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web .

```
<?php      header('Access-Control-Allow-
Origin: *');
require_once("../class/about.php");
        $obj = new About();
echo json_encode($obj);
?>
```

Gambar 2.7 Contoh Skrip PHP

2.1.3.3 Database Mysql

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. MySQL adalah implementasi dari system manajemen basis data relasional (RDBMS). MySQL dibuat oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL.

MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael "Monty". Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

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

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

Sistem database menyediakan bahasa untuk mendefinisikan database Data Definition Language (DDL) serta bahasa untuk memanipulasi database Data Manipulation Language (DML) untuk melakukan operasi-operasi tertentu pada database dan juga Data Control Language (DCL).

a. Data Definition Language (DDL)

Kumpulan perintah SQL yang digunakan untuk membangun database, bisa juga memiliki arti lain perintah untuk membuat struktur dasar database dan tabel. Perintah-perintah yang termasuk DDL adalah statment connect, create, show, describe, drop, rename, dan alter.

b. Data Manipulation Language (DML)

Kumpulan statement yang digunakan sebagai perintah untuk mengelola data dalam sebuah tabel. Perintah DML digunakan setelah menggunakan DDL antara lain seperti statement insert, update, delete, dan select.

c. Data Control Language (DCL)

Perintah yang digunakan untuk keperluan keamanan (security) database dengan membuat hak akses tertentu bagi setiap user. Statment pada DCL antara lain grant, revoke, set, dan lock table.

Dalam MySQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD sebagai berikut:

1. *Create*

Create adalah operasi penambahan data baru kedalam table. Terdapat 2 Query untuk menambah data, yang pertama adalah penambahan yang tidak menspe

sifikasikan nama kolom yang akan ditambahkan dan hanya memberikan isi dari tabelnya. Query nya adalah sebagai berikut:

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

Untuk yang kedua adalah menginputkan data dengan mencantumkan nama kolom yang akan diisikan dengan isi di dalamnya. Query nya adalah sebagai berikut:

```
INSERT INTO table_name (column1,column2,column3,...)
```

```
VALUES (value1,value2,value3,...);
```

2. *Read*

Read adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam database. Query nya adalah sebagai berikut:

```
SELECT column_name,column_name  
FROM table_name;
```

3. *Update*

Update adalah operasi untuk mengubah data yang ada di dalam database. Query nya adalah sebagai berikut:

```
UPDATE table_name SET column1=value1,column2=value2,... WHERE  
some_column=some_value;
```

4. *Delete*

Delete adalah operasi untuk menghapus data yang ada di dalam table. Query nya adalah sebagai berikut:

```
DELETE FROM table_name  
WHERE some_column=some_value;
```

2.1.3.4 Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java. Pada Desember 1998, Sun

memperkenalkan nama “Java 2” (J2) sebagai generasi kedua dari java platform. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME) .

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

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

2.1.3.4.1 Kelebihan Java

Adapun beberapa kelebihan dari bahasa pemrograman Java adalah sebagai berikut :

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program dengan Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memberi anda kemampuan untuk menuangkan semua ide, karena bahasa pemrograman ini bukan merupakan scripting language (bahasa naskah) yang menghilangkan kemampuan kita untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

2. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasiaplikasi yang dibangun dengan bahasa Java sangat handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat secure, yaitu kasus-kasus seperti buffer everflow yang umumnya menjadi lubang keamanan aplikasiaplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan securitynya yang bagus.

3. Berorientasi Objek

Paradigma pemrograman berorientasi objek merupakan paradigma pemrograman masa depan. Java merupakan bahasa pemrograman berorientasi objek. Java bukan turunan langsung dari bahasa pemrograman manapun, juga sama sekali tidak kompetibel dengan semuanya. Java memiliki keseimbangan, Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memberi anda kemampuan untuk menuangkan semua ide, karena bahasa pemrograman menyediakan mekanisme peng-class-an sederhana, dengan model antar muka dinamik yang intuitif hanya jika diperlukan.

4. Kokoh

Java membatasi anda dari beberapa hal kunci supaya anda dapat menemukan kesalahan lebih cepat saat mengembangkan program. Java langsung memeriksa program saat anda menuliskannya, dan sekali lagi ketika program di jalankan. Karena Java adalah bahasa yang sangat ketat dalam hal tipe data dan deklarasi, banyak kesalahan umum terjadi saat kompilasi. Hal ini akan lebih menghemat waktu jika dibandingkan dengan keharusan menjalankan program terlebih dahulu dan memeriksa semua bagian program untuk melihat ketidakcocokan dinamis selama program berjalan. Ini adalah contoh di mana Java lebih luwes dan kokoh dari beberapa bahasa lain, tetapi dengan imbalan yang layak untuk kelebihan itu.

5. Interaktif

Java memiliki beberapa kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan kita untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

6. Netral terhadap Berbagai Arsitektur

Java telah mengambil beberapa keputusan yang sulit dalam pembuatan bahasa Java dan bagaimana program dijalankan, jadi anda dapat sepenuhnya percaya “tulis sekali, jalan di mana saja, kapan saja, selamanya”.

7. Terinterpretasi dan Berkinerja Tinggi

Java dilengkapi keajaiban lintas platform yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java (Java byte-code), yang dapat diterjemahkan oleh sistem manapun yang memiliki program Java didalamnya. Java, bagaimanapun dirancang untuk tetap berkinerja baik pada CPU yang tidak terlalu kuat. Walaupun Java merupakan bahasa terinterpretasi, kodekode Java telah dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk menghasilkan kinerja yang tinggi.

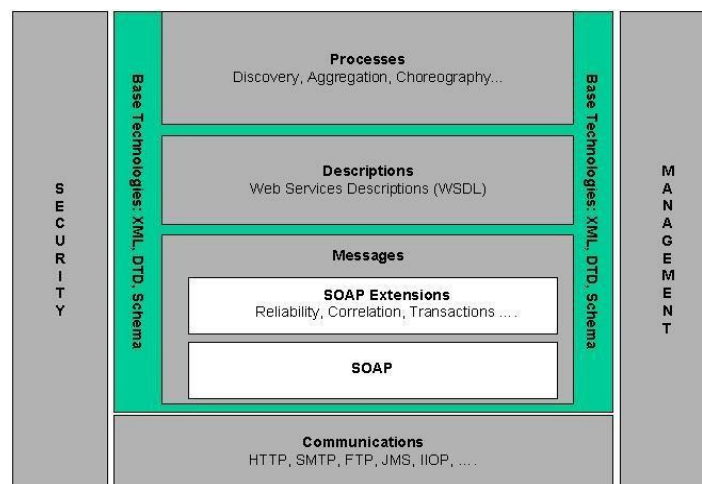
Sistem program Java yang melakukan optimasi tepat waktu tersebut tidak kehilangan keuntungan dari program yang netral terhadap platform. “lintas platform berkinerja tinggi” bukan sekedar omong-kosong. Dalam aplikasi Java (*.class) merupakan Java bytecode yang berjalan di atas jvm (Java Virtual Machine), yang kemudian jvm-lah yang akan menginterpretasikan kode-kode tersebut ke kode native atau kode mesin dari arsitektur yang bersangkutan. Hal sangat menarik karena urusan arsitektur mesin bukan jadi masalah bagi programmer tapi menjadi urusan kompiler pada bahasa pemrograman Java.

2.1.3.5 Web Service

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam *platform* dan antarmuka bahasa pemrograman yang independen. Tujuan pengembangannya adalah untuk menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standart protokol yang ditetapkan oleh *web service*. Hal

ini bisa terjadi, karena standart protokol itu tidaklah terikat pada suatu *platform* atau bahasa pemograman. Protokol itu sendiri dibangun oleh *Extensible Markup Language (XML)* yang memang kenyataannya telah didukung oleh banyak *platform*, bahasa pemograman, dan oleh *developer* di seluruh dunia.

Arsitektur *web service* dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Banyak cara untuk memvisualisasikan *service*, sama banyaknya dengan cara untuk membangun dan menggunakan *web service* gambar berikut hanyalah salah satu cara untuk menggambarkan arsitekturnya.



Gambar 2.8 Arsitektur Web Service

Web service disusun oleh tiga komponen standart, yaitu :

1. *Simple Object Access Protocol (SOAP)*, yaitu protokol yang bertanggung jawab dalam pertukaran informasi dalam lingkungan jaringan terdistribusi.
2. *Web Service Definition Language (WSDL)*, dokumen standart yang dituliskan pada format XML, dan mendefinisikan kehadiran web service dalam suatu jaringan.
3. *Universal Description, Discovery and Integration (UDDI)*, yaitu suatu lokasi direktori yang berisikan *service* (layanan) dan bersifat bebas

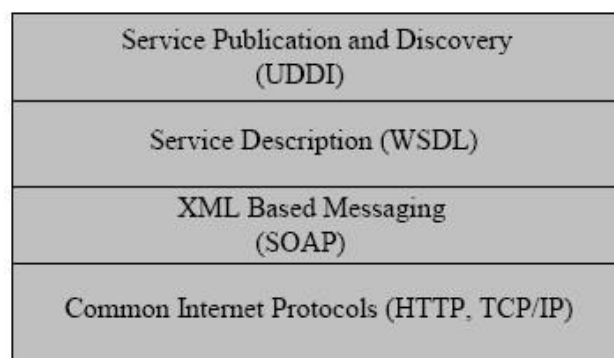
platform (platform independent), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam dan luar jaringan.

pada dalam suatu pekerjaan. Adanya standart tersebut membuat web *service* mudah di akses melalui berbagai antarmuka dan juga memberi peluang dimungkinkannya berbagai sistem yang dibangun pada platform yang berbeda dan bahasa yang berbeda untuk berkolaborasi

2.1.3.5.1 Komponen – Komponen Web Service

Web Service secara keseluruhan memiliki empat layer komponen yaitu sebagai berikut :

1. Layer 1 : Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2 : *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3 : *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.



Gambar 2.9 Komponen – Komponen Web Service

2.1.3.6 Xampp

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer. XAMPP juga dapat disebut juga CPanel server virtual, yang dapat membantu melakukan

priview sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

