BAB II
LANDASAN TEORI

II.1 Internet

Internet (*Inter-Network*) adalah sebutan untuk sekumpulan jaringan komputer yang menghubungkan situs akademik, pemerintahan, komersial, organisasi, maupun perorangan. Internet menyediakan akses untuk layanan telekomnunikasi dan sumber daya informasi untuk jutaan pemakainya yang tersebar di seluruh dunia. Layanan internet meliputi komunikasi langsung (email, chat), diskusi (Usenet News, email, milis), sumber daya informasi yang terdistribusi (World Wide Web, Gopher), remote login dan lalu lintas file (Telnet, FTP), dan aneka layanan lainnya [3].

II.1.1 *Hypertext Transfer Protocol* (HTTP)

Hypertext Transfer Protocol atau HTTP merupakan suatu protokol yang dipakai oleh www dimana http itu sendiri memberikan definisi terhadap bagaimana pesan dapat disampaikan dan diformat dari server menuju klien. Selain itu, HTTP juga digunakan untuk mendistribusikan sistem informasi berbasis *hypertext*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses HTML. HTTP diprakarsai oleh *World Wide Web* sistem informasi yang menyeluruh sejak tahun 1990 [3].

II.1.2 *Hypertext Transfer Protocol Secure* (HTTPS)

Hypertext Transfer Protocol Secure atau HTTPS adalah sebuah protokol web yang dikembangkan oleh Netscape dan dibangun di dalam browsernya sehingga dapat melakukan enkripsi dan dekripsi permintaan laman dari pengguna sama halnya dengan laman yang dikembalikan oleh server web. HTTPS merupakan penggabungan antara Hypertext Transfer Protocol (HTTP) dengan SSL / TLS protokol. Semua komunikasi yang dilakukan melalui HTTPS akan dienkripsi dengan tujuan untuk keamanan saat terjadi transaksi data di internet [3].

II.1.3 *File Transfer Protocol* (FTP)

Fasilitas ini memungkinkan para pengguna internet untuk melakukan pengiriman (*upload*) atau menyalin (*download*) sebuah file antara komputer lokal dengan computer lain yang terhubung dalam jaringan internet. Protokol standar yang digunakan untuk keperluan ini disebut sebagai *File Transfer Protocol* (FTP). FTP umumnya dimanfaatkan sebagai sarana pendukung untuk kepentingan pertukaran maupun penyebarluasan sebuah file melalui jaringan internet. FTP juga dimanfaatkan untuk melakukan prose *upload* suatu halaman web ke *webserver* agar dapat diakses oleh pengguna internet lainnya [3].

II.1.4 *Electronic Mail / Email / Messaging*

Electronic Mail / Email / Messaging dalam istilah Indonesia diartikan sebagai aplikasi yang memungkinkan para pengguna internet untuk saling berkirim pesan melalui alamat elektronik di internet. Para pengguna email memilki sebuah *mailbox* (kotak surat) elektronik yang tersimpan dalam suatu *mailserver*. Suatu *Mailbox* memiliki sebuah alamat sebagai pengenal agar dapat berhubungan dengan *mailbox* lainnya, baik dalam bentuk penerimaan maupun pengiriman pesan. Pesan yang diterima akan ditampung dalam *mailbox*, selanjutnya pemilik *mailbox* sewaktu-waktu dapat mengecek isinya, menjawab pesan, menghapus, atau menyunting dan mengirimkan pesan email [3].

II.2 *Database*

Kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data dan juga update yang rumit. Sedangkan untuk proses memasukan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang

disebut dengan sistem manajemen basis data atau lebih dikenal dengan istilah *Database Management System* (DBMS) [4].

II.2.1 Fungsi Database

Adapun fungsi database diantaranya adalah sebagai berikut [4]:

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menentukan informasi.
2. Menentukan kualitas informasi. Informasi dapat dikatakan bernilai apabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
3. Mengurangi duplikasi data *(data redudancy).*
4. Hubungan data dapat ditingkatkan *(data relatability).*
5. Mengurangi pemborosan tempat simpanan luar.

II.2.2 Kriteria Database

Dari penjelasan sebelumnya dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu [4]:

1. Bersifat data *oriented* dan bukan *program oriented*
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah, dapat digunakan dengan cara-cara yang berbeda.

II.2.3 *Database Management System* (DBMS)

DBMS merupakan sistem perangkat lunak yang memungkinkan *user*untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh DBMS. Berikut adalah beberapa fungsi yang harus ditangani DBMS, yaitu [4] :

1. Mengolah pendefinisian data
2. Dapat menangani permintaan pemakai untuk mengakses data
3. Memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (*Database Administrator*)

4. Menangani kegagalm dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun *disk* dan menangani unjuk kerja semua fungsi secara efisien

II.3 *Game*

Game diambil dari bahasa Inggris yang diterjemahkan yang artinya adalah permainan. Pengertian *game* menurut beberapa ahli :

1. Menurut Chris Crawford, seorang *game designer* :
   "*Game pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada pelaku aktif, ada lawan anda."*
2. Menurut David Parlett, seorang *gamer profesional* :
   "Game adalah sesuatu yang memiliki akhir dan cara mencapainya, artinya ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya".
3. Menurut Roger Caillois, seorang sosiolog Perancis, dalam bukunya berjudul *Les jeux et les hommes* menyatakan :
   "Game adalah aktivitas yang mencakup karakteristik berikut : fun (bebas bermain adalah pilihan bukan kewajiban), separate (terpisah), uncertain, non-productive, governed by rules (ada aturan), fictitious (pura-pura)".

Dari pengertian diatas dapat disimpulkan bahwa *game* adalah sebuah aktivitas yang interaktif dan memiliki tujuan dalam pencapaiannya yang diatur oleh sebuah aturannya sendiri, adanya pelaku aktif dan memiliki lawan main serta memiliki sebuah akhir. Sedangkan, Game atau main dalam kamus besar bahasa indonesia adalah melakukan permainan untuk menyenangkan hati dengan menggunakan alat-alat tertentu atau tidak atau melakukan perbuatan untuk bersenenang-senang dengan alat-alat tertentu atau tidak dalam keadaan berlangsung atau mempertunjukkan tontonan.

II.3.1 Klasifikasi Game

The Entertainment Software Rating Board (ESRB) adalah sebuah organisasi *regulator* mandiri yang menilai permainan *video*, panduan pengiklanan, prinsip privasi *online* permainan *video* dan *software* hiburan lainnya di Kanada dan Amerika Serikat. Berikut ini adalah klasifikasi *game* berdasarkan pembatas usia, menurut *Entertainment Software Rating Board* (ESRB), yaitu [5] :

Tabel II.1 Klasifikasi Game [5]

Simbol	Keterangan
	***Early Childhood*** **(3+)** memiliki isi yang mungkin cocok untuk usia 3 tahun atau lebih tua.
	***Everyone*** **(6+)** memiliki konten yang mungkin cocok untuk usia 6 tahun atau lebih tua. Judul-judul dalam kategori ini mungkin berisi sedikit kartun, fantasi atau kekerasan ringan.
	***Everyone*** **(10+)** memiliki konten yang mungkin cocok untuk usia 10 tahun atau lebih tua. Judul-judul dalam kategori ini mungkin berisi lebih banyak kartun, fantasi, atau kekerasan ringan, bahasa ringan dan *minimal* tema sugestif.
	***Teen*** **(13+)** memiliki konten yang mungkin cocok untuk usia 13 tahun dan lebih tua. Judul-judul dalam kategori ini mungkin mengandung kekerasan, tema sugestif, humor kasar, sedikih darah dan simulasi perjudian.
	***Mature*** **(17+)** memiliki konten yang mungkin cocok untuk usia orang usia 17 tahun atau lebih. Judul-judul dalam kategori ini mungkin berisi intens kekerasan, darah dan konten seksual.

	***Adults Only* (18+)** memiliki konten yang hanya boleh dimainkan oleh orang-orang berusia 18 tahun atau lebih tua. Judul-judul dalam kategori ini mungkin termasuk adegan yang intens dengan kekerasan dan grafik konten seksual.
	***RP (Rating Pending)*** simbol ini muncul dalam iklan sebelum permainan sampai kepada pasar.

II.3.2 Jenis - Jenis Game

Game sendiri memiliki jenis-jenisnya tersendiri, yang dimaksudkan untuk mengenal *game* apa yang sedang kita mainkan termasuk kedalam jenis yang mana. Dan ini adalah jenis-jenis *game* tersebut [5] :

1. Simulasi

   Contoh permainan yang termasuk dalam *game* simulasi adalah simulasi konstruksi dan manajemen, simulasi kendaraan seperti yang diterapkan pada permainan balapan dan perang luar *edugames* angkasa.

2. Edukasi

   Contohnya adalah *edugames* yang dibuat dengan tujuan spesifik sebagai alat pendidikan, baik itu untuk belajar mengenal warna untuk balita, mengenal huruf dan angkat, matematika, sampai belajar bahasa.

3. *Entertaiment*

a) Aksi – *Shooting*, (tembak-tembakan, atau hajar-hajaran bisa juga tusuk tusukan, tergantung cerita dan tokoh di dalamnya). *Game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan dan juga waktu.

b) *Fighting* (pertarungan), *game* ini memerlukan keceptan reflek dan koordinasi mata-tangan, tetapi inti dari *game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya).

c) Petualangan, *game* murni pertualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara *visual*, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan

percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat.

d) *Role Playing*, *game* jenis ini memiliki penekanan pada tokoh atau peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan oleh pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya level.
e) *Casual Games*, *game* ini merupakan jenis game yang dimainkan dengan rileks, simple dan tidak kompleks serta mudah untuk dipelajari. Selain itu, game jenis ini ukurannya sedikit dan tidak lebih dari 100MB.
f) *Multiplayer Online*, game yang dapat dimainkan secara bersamaan oleh lebih dari 2 orang (bahkan dapat mencapai jutaan orang dalam satu waktu tertentu) membuat pemain dapat bermain bersama dalam satu dunia *virtual* dari sekedar *chatting* biasa hingga saling menyerang dan menjatuhkan antar pemain.

II.4 *Gamers*

Gamers adalah sebutan untuk orang yang bermain *game* dengan menggunakan perangkat elektronik. *Gamers online* yaitu orang yang bermain *game* dengan perangkat komputer dan terhubung dengan jaringan internet, dimana para *gamers* dapat terhubung satu sama lain tanpa dibatasi oleh ruang dan waktu. *Gamers* tidak mengenal batasan usia, status, jenis kelamin dan jenis pekerjaan. Semua orang yang bermain *game* dengan menggunakan perangkat elektronik yang ada dapat dikatakan sebagai seorang *gamers* [6].

II.5 *eSport*

eSport merupakan olahraga yang dimainkan dengan menggunakan media elektronik, dimana dibeberapa negara, seperti : Korea Selatan, Swedia dan Indonesia *game* sudah dikategorikan sebagai salah satu cabang olahraga dan memiliki lembaga resmi untuk menaungi para pemain *game* tersebut, dibawah kementerian olahraga di negara-negara tersebut. eSport adalah salah satu lini yang

paling berkembang dalam beberapa tahun ke belakang. Walaupun begitu, usia eSport bisa dibilang hampir setua sejarah dunia gaming sendiri. Saat mode *multiplayer* mulai diperkenalkan pada konsumen, saat itu pula para *gamers* seperti mendapatkan sebuah dunia baru yang disebut dengan dunia *virtual*, dimana para *gamers* bisa bermain, bersosialisasi dan saling berkompetisi dengan *gamers* lainnya. [7].

II.5.1 *eSport Team*

(Albert F, 2010) *eSport Team* atau tim eSport merupakan kumpulan dari *gamers-gamers* yang bermain bersama dalam kategori *game* tertentu dan memiliki visi dan misi yang sama dalam mencapai tujuan dari tim *eSport* tersebut. Para *gamers* tersebut memainkan *game* yang sama dalam satu divisi *eSport Team*, namun *eSport Team* bisa memiliki lebih dari satu divisi *eSport Team* dengan kategori *game* yang berbeda, misalnya : salah satu *eSport Team* di Indonesia yang telah memiliki banyak penghargaan baik penghargaan lokal maupun internasional, yaitu **nextlevelgaming** atau lebih dikenal dengan nama **team nxl>** memiliki beberapa divisi *eSport Team* diantaranya : divisi CS:GO, X-SHOT dan DOTA2.

II.5.2 Pembentukan *eSport Team*

eSport Team terdiri dari 2 atau lebih *gamers* yang memiliki kesamaan terhadap suatu jenis tertentu dan untuk mencari *gamers* yang sesuai untuk membentuk suatu *eSport Team* tidaklah mudah. Hal ini dikarenakan oleh banyak faktor, yaitu :

1. Lokasi

   Lokasi sangat menentukan *eSport Team* di dalam berlatih. Hal ini dikarenakan apabila terdapat salah satu anggota *eSport Team* yang berbeda kota hal ini dapat menghambat komunikasi antara satu anggota *eSport Team* dengan anggota *eSport Team* yang lain.

2. Jenis Game

   Jenis Game merupakan salah satu faktor yang sangat mempengaruhi di dalam pembentukan suatu *eSport Team*. Karena *eSport Team* merupakan

kumpulan *gamers* yang sama-sama bermain di dalam satu jenis game yang sama.

3. Posisi Pemain

   Di dalam *eSport Team* setiap pemain memiliki fungsi dan perannya masing-masing. Pembagian peran atau tugas ini berdasarkan keahlian dari masing-masing *gamer* tersebut. Pembagian peran ini bertujuan agar di dalam setiap pertandingan setiap strategi yang telah ditentukan dapat berjalan sesuai dengan rencana.

4. Usia

   Usia juga dapat mempengaruhi terbentuknya suatu *eSport Team* karena usia dapat mempengaruhi tingkat mentalitas, kecepatan respon mata dan reflek seseorang. Karena untuk jenis *game* tertentu seperti jenis *game* FPS *(First Person Shooter)* diperlukan kejelian mata di dalam melihat setiap pergerakan lawan.

II.5.3 Posisi Pemain

Setiap *game* memiliki gaya permainan yang berbeda, namun pada dasarnya setiap *gamers* memiliki peran dan fungsi masing-masing di dalam timnya. Setiap *player* memiliki tugas yang berbeda satu sama lain, namun peran tersebut terbagi menjadi 5 bagian utama, yaitu :

1. *IGL (In Game Leader)*

Player ini merupakan otak dari setiap strategi yang akan diterapkan di dalam permainan, dimana IGL merupakan ketua tim yang akan mengarahkan para anggotanya agar dapat bermain dengan baik agar sesuai dengan strategi yang akan diterapkan.

2. Penyerang / Killer

Player ini bertugas untuk menyerang tim lawan dan mengobrak – abrik pertahanan lawan. *Player* ini memiliki *skill* / kemampuan lebih diantara *player* lainnya. Penyerang / *killer* memiliki peranan yang sangat penting di dalam memangkan pertandingan, karena di dalam *game* bergenre FPS dan MOBA

semakin banyak penyerang / *killer* membunuh tim lawan maka kemungkinan besar tim tersebut akan memenangkan pertandingan.

3. Pembantu / *Backupper*

   Player ini bertugas untuk membantu *killer* di dalam melakukan tugasnya untuk menyerang tim lawan. Apabila *killer* terbunuh di dalam pertandingan maka *backupper* bertugas untuk menggantikan tugas dari pada *killer*.

4. *Backstab*

   Player ini bertugas untuk menjaga tim dari serangan belakang lawan. Apabila tim sedang menyerang ke suatu tempat maka *backstab* bertugas untuk menjaga sisi belakang tim agar tidak di serang dari belakang.

5. *Support*

   Player ini bertugas untuk membantu *killer* di dalam menjalankan tugasnya. *Support* bertugas untuk membukakan jalan untuk *killer* sebelum *killer* menyerang tim lawan.

II.6 *Mobile* Teknologi

Mobile teknologi adalah istilah kolektif yang digunakan untuk menggambarkan berbagai jenis teknologi komunikasi seluler. *Handphone* yang dulunya hanya digunakan sebagai alat untuk berkomunikasi satu sama lain sudah berubah dengan berkembangnya teknologi dan membuat *handphone* saat ini tidak hanya sebagai alat untuk berkomunikasi saja, namun memiliki banyak fungsi, seperti : bermain *game*, menonton TV, mencari informasi berita dan mencari lokasi suatu daerah. Banyak ahli berpendapat bahwa masa depan teknologi komputer terletak pada komputasi *mobile* dengan jaringan nirkabel. Sehingga membuat teknologi pada *mobile* akan terus berkembang dari waktu ke waktu.

II.7 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis

secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008 [8] :

II.7.1 Android Life Cycle

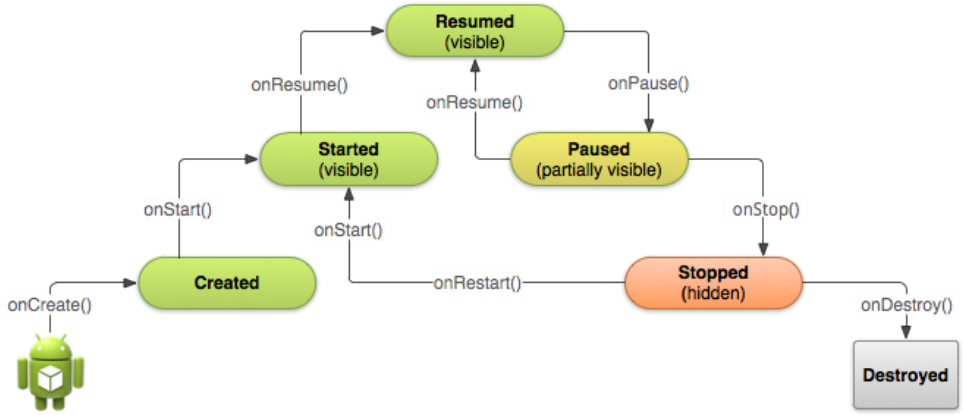
Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada,klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [9].

Tabel II.2 Komponen Aplikasi Android

Functionality	Java Base Class	Examples
Focused thing a user can do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Receive messages	BroadcastReceiver	Trigger alarm upon event
Store and retrieve data	ContentProvider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service*, *BroadcastReceiver*, *ContentProvider* [9].

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam method Activity dengan menerapkan metode callback tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi Android memiliki siklus hidup yang berbeda dengan aplikasi desktop ataupun web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang cukup tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai penggunan [8].



Gambar II.1 Siklus Hidup Android [7]

Terdapat beberapa *state*dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar II.1 Siklus Hidup Android, akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya:

1. **Resumed**

Resumed terjadi ketika aplikasi berjalan setelah state *paused* . State ini akan menjalankan perintah program yang ditulis pada method onResume() [8].

2. **Paused**

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini [7].

3. **Stopped**

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground [7].

State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cyclecallback* berikutnya. Artinya, setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), krmudian diikuti oleh onResume() [9].

II.7.2 Fitur

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah:

1. *Multi-proses* dan *App Widgets*

Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara *multitasking.* Keuntungan yang didapat adalahketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan *game*, proses lain dapat berjalan di belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan [7].

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widgets dapat menjalankan prosesrequest seperti musik streaming atau mendeteksi suhu ruangan secara *background* [7].

Multi-proses dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaantidak benar [8].

2. *Hard* dan *Soft Keyboard*

Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi [8].

II.7.3 Prinsip Desain

Android memiliki beberapa prinsip desain yang dapat menjadi acuan dalam membuat desain aplikasi android diantaranya adalah :

1. *Multiple Assets*

Android mendukung jutaan smartphone, tablet dan perangkat lain dalam berbagai ukuran layar dan ukuran, untuk itu *Multiple Assets*sangat disarankan digunakan untuk mengatasi fragmentasi pada android. Seperti ilustrasi pada Gambar II.2 Klasifikasi Ukuran Ikon, android menciptakan beberapa klasifikasi ukuran icon yaitu : MDPI, HDPI, XHDPI, XXHDPI dan XXXHDPI. MDPI dan HDPI dikhususkan untuk icon yang akan digunakan pada device berukuran smartphone sedangkan untuk XHDPI, XXHDPI dan XXXHDPI digunakan pada device berukuran tablet [9].



Gambar II.2 Klasifikasi Ukuran Ikon

2. *Touch Feedback*

Touch Feedback dalam android digunakan sebagai respon setiap objek yang ditekan pengguna. Hal ini bertujuan untuk memberi tahu pengguna objek mana yang berinteraksi dengan penggunan [9].



Gambar II.3 Touch Feedback

II.8 *User-Generated Content* (UGC)

User-generated Content atau disingkat UGC, dikenal juga dengan istilah *consumer-generated* media (CGM) atau *user-created content* (UCC) adalah merujuk pada berbagai tipe konten materi atau media yang tersedia untuk umum dan diproduksi oleh end-user atau para non-media profesional. Kemajuan teknologi web memungkinkan konten website tidak lagi dimonopoli oleh profesional media, melainkan dapat dibuat oleh para penggunanya. UGC merupakan salah satu ciri dominan Web 2.0. Contoh penerapan pada UGC adalah YouTube hampir semua konten yang dimiliki situs tersebut dibuat dan diupload oleh penggunanya [10].

Konsep UGC pada dasarnya telah banyak merubah cara berinteraksi pengguna dengan internet begitu juga dalam media periklanan. Bagi media periklanan jejaring sosial dengan konsep UGC memiliki potensi besar menyediakan market yang lebih terarah dan terpusat bagi mereka [10].

II.9 *Web Services*

W3C mendefinisikan web service sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML atau JSON melalui protokol internet. Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, menyediakan method-method yang dapat diakses oleh network. Web Service juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda. Beberapa karakteristik dari web service adalah:

1. *Message-based*
2. *Standards-based*
3. *Programming language independent*
4. *Platform-neutral*

Beberapa key standard didalam web service adalah: JSON, XML, SOAP, WSDL and UDDI [11].

II.10 *JavaScript Object Notation* (JSON)

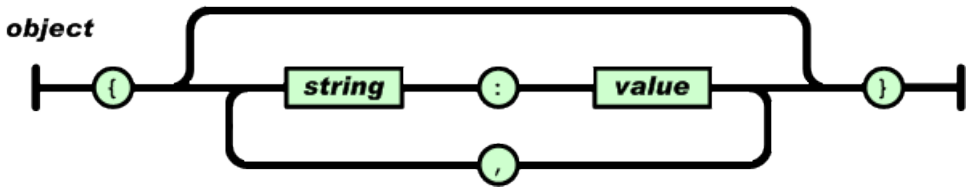
JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang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

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array [12].
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence) [12].

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini.JSON menggunakan bentuk sebagai berikut:

1. Objek

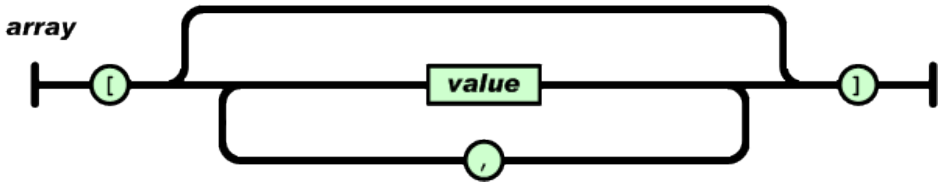
Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON [12].



Gambar II.4 Objek JSON

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma). Larik dalam JSON dapat digunakan sebagai value dari JSON object hal ini dapat berguna jika JSON menyimpan data bertingkat [12].

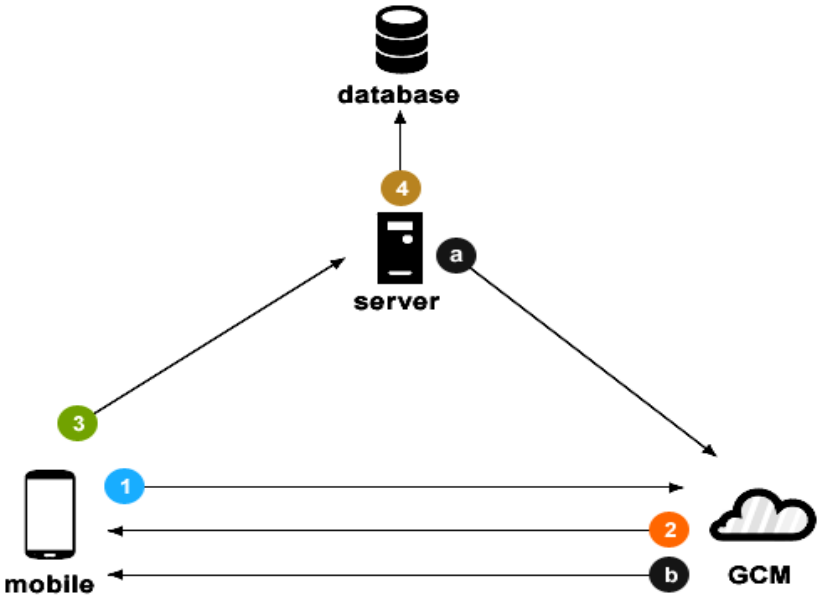


Gambar II.5 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Bollean dan nilai NULL [12].

II.11 *Google Cloud Messaging*

Google Cloud Messaging adalah teknologi untuk membantu pengembang mengirim data dari server menuju device android secara langsung melalui server *google*, pemanfaatan teknologi ini biasanya digunakan untuk *push messaging.* Pengiriman pesan menggunakan teknologi ini mengharuskan adanya API key, key ini terkait langsung dengan akun google yang digunakan untuk mempublikasikan aplikasi melalui *google play* [13].



Gambar II.6 GCM Model [13]

Berdasarkan ilustrasi dari Gambar II.6 GCM Model [13] terdapat dua server yang terlibat dalam proses *push messaging*, pada setiap server yang terlibat terdapat proses pengiriman data yang berbeda-beda diantaranya [13] :

1. Device android mengirimkan sender id dan application id menuju GCM Server untuk registrasi.
2. Setelah berhasil registrasi, GCM server akan mengirimkan registration id menuju device android.
3. Setelah device android mendapatkan registration id, device tersebut akan mengirimkan registration id menuju server aplikasi.
4. Server aplikasi akan menyimpan registrasi id yang dikirimkan untuk digunakan sebagai id saat menggunakan *push messaging*.

II.12 *Object Oriented Analysis Desain*

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaiut analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD

adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem [14] :

II.12.1 *Unified Modeling Language* (UML)

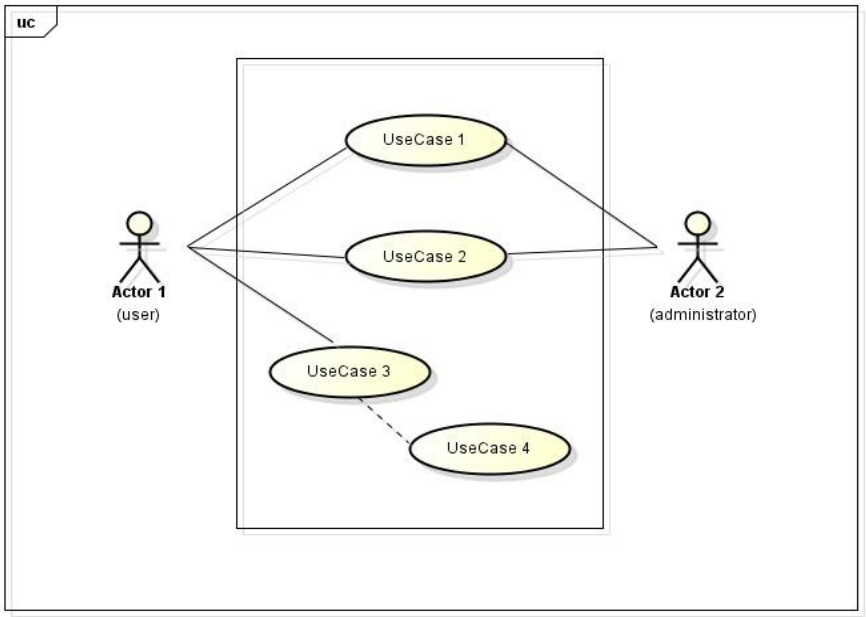
Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal, Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented*(OO). UML merupakan standar terbuka yang dikelola oleh Open Management Group (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah [14] :

1. *Use Case Diagram*

Use case diagram merupakan suatu pemodelan yang menitik beratkan padafungsionalitas utama dalam suatu konteks sistem. Use case diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi denganobjek diluar sistem tersebut. Selain itu, use case diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponendalam use case diagram antaranya [14] :

a. Aktor

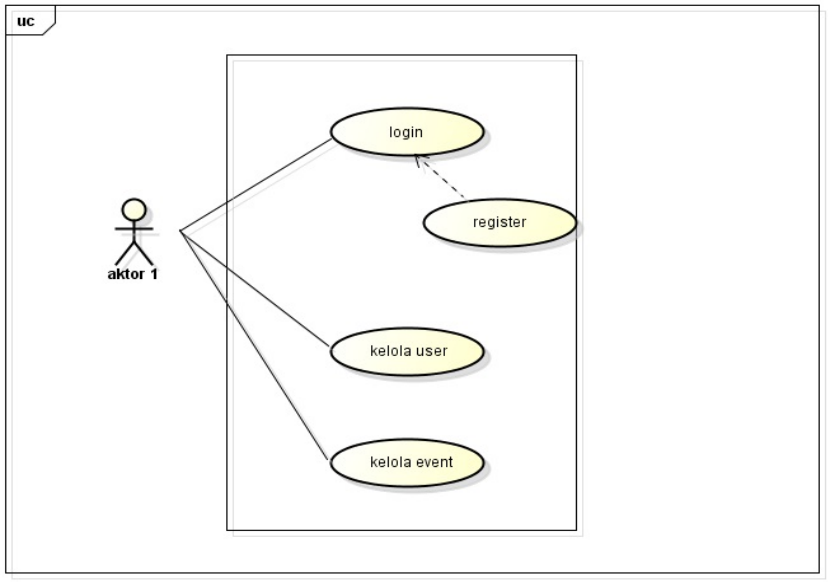
Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas di dalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut [14]. Berikut adalah contoh aktor yang terlibat pada perangkat lunak yang akan dibangun :



Gambar II.7 Aktor

b. *Use Case*

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisidari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem [14]. Berikut adalah contoh use case dari perangkat lunak yang akan dibangun :



Gambar II.8 *Use Case*

2. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi properti dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untukmenggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas.Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private, dan protected [14]. Berikut adalah contoh class diagram :

```
public class event_games
{


}
```

3. *Sequence Diagram*

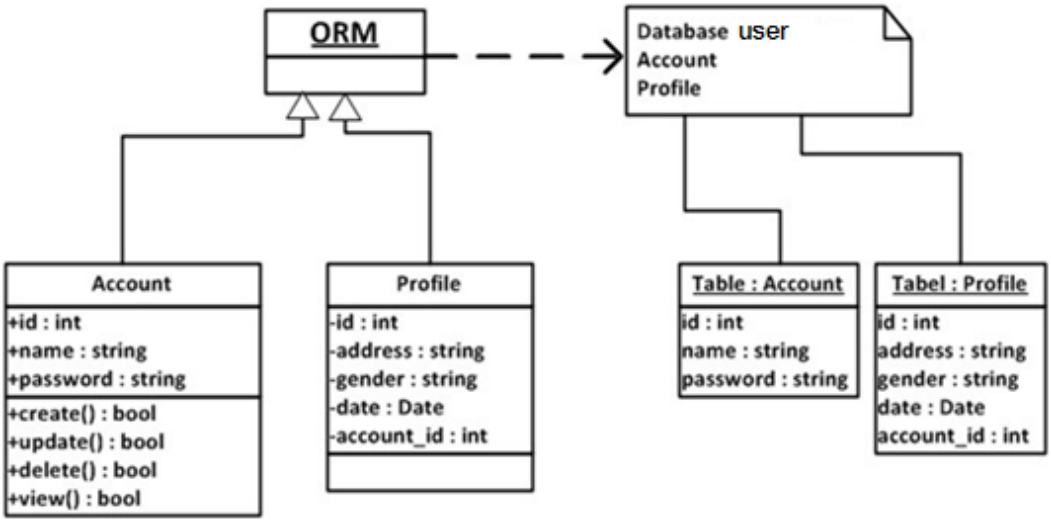
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case : interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi [14] :

II.13 Object Relational Mapping (ORM)

Object Relational Mapping adalah salah satu teknik untuk memetakan basis data relasional ke model objek. Pemetaan dapat dikategorikan menjadi 4, yaitu [136] :

1. *Basic mapping*
2. *Relationship mapping*
3. *Composition mapping*
4. *Inheritance mapping*

Berikut adalah contoh gambar ORM dari penjelasan yang telah dijelaskan sebelumnya.



Gambar II.9 *Object Relational Mapping(ORM)*

II.14 *Location Based Services* (LBS)

Location Based Service (LBS) atau layanan berbasis lokasi adalah sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak

tersebut. *Location Based Service* dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu, seperti menemukan lokasi *event games*.

II.14.1 Unsur Utama LBS

Berikut adalah dua unsur utama dari *Location Based Service*, yaitu :

1. ***Location Manager* (*API Maps*) :** Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programming Interface* (API) *Maps* menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada "com.google.android.maps;".
2. ***Location Providers* (*API Location*) :** Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada paket Android yaitu dalam paket "android.location". Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditentukan melalui *Location Manager*.

II.14.2 Komponen LBS

Terdapat lima komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain :

1. **Piranti *Mobile*,** adalah salah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu (*tools*) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Piranti mobile yang dapat digunakan bisa berupa PDA, *smartphone*, laptop. Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.
2. **Jaringan Komunikasi,** Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. **Komponen *Positioning* (Penunjuk Posisi/Lokasi),** Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi mobile atau juga menggunakan *Global Positioning System* (GPS).
4. **Penyedia layanan dan aplikasi,** merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di *Yellow Pages* sesuai dengan permintaan, dan masih banyak lagi yang lainnya.
5. **Penyedia data dan konten**, Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya.

II.14.3 Cara Kerja LBS

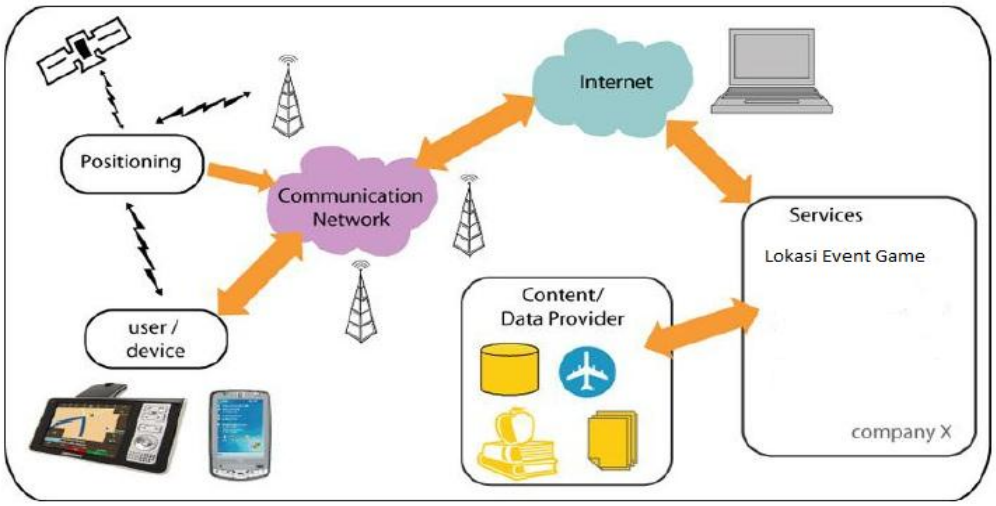
Untuk menggambarkan cara kerja LBS pada perangkat lunak yang akan dibangun pada fitur pencarian lokasi *event games* dapat dijelaskan sebagai berikut :

1. Anggaplah sekarang fungsi pencarian telah diaktifkan, posisi pengguna sebenarnya dari perangkat mobile diperoleh dari *positioning service*. Hal ini dapat dilakukan baik oleh perangkat menggunakan GPS sendiri atau layanan posisi jaringan yang berasal dari *provider* (*Cell Tower)*. Setelah

itu perangkat *mobile* pengguna mengirimkan permintaan informasi, yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke *gateway* telekomunikasi.

2. *Gateway* memiliki tugas untuk bertukar pesan di antara jaringan komunikasi selular dan internet. Oleh karena itu dia mengetahui alamat *web* dari beberapa aplikasi *server* dan rute permintaan ke spesifik *server* tertentu. *Gateway* akan menyimpan juga informasi tentang perangkat *mobile* yang telah meminta informasi.
3. Aplikasi *server* membaca permintaan dan mengaktifkan layanan yang terkait.
4. Kemudian *service* akan menemukan bahwa pengguna membutuhkan informasi tentang lokasi *event games* dari *database yellow pages* pada wilayah tertentu dan kemudian *service* tersebut akan meminta penyedia data untuk memberikan data tersebut.
5. Setelah semua informasi *service* telah melakukan *buffer* spasial dan *query routing* untuk mendapatkan lokasi dari *event game* tersebut. Setelah itu, informasi lokasi yang didapat akan dikirim kembali ke pengguna melalui *internet, gateway* dan jaringan *mobile.*
6. Kemudian informasi mengenai lokasi *event game* akan diterima kepada pengguna dalam bentuk peta digital.

Berikut adalah contoh gambar cara kerja dari LBS seperti yang telah dijelaskan sebelumnya :



Gambar II.12 Cara Kerja *Location Based Services* (LBS)

II.15 XAMPP

Menurut Bunafit Nugroho (2008 : 2) XAMPP adalah suatu bundel *web server* yang populer digunakan untuk coba-coba di *Windows* karena kemudahan instalasinya. Bundel program *open source* tersebut berisi antara lain server web, Apache, Interpreter PhP, dan basis data MySQL. Setelah menginstall XAMPP, kita bisa memulai pemrograman PhP di komputer sendiri maupun mencoba menginstall aplikasi-aplikasi *web*.

II.16 Sublime Text 3

Sublime Text 3 adalah sebuah *syntax editor* yang menggunakan Python API yang biasa digunakan oleh para programmer, khususnya *Web Developer*. Sublime Text mempunyai fitur *plugin* tambahan yang memudahkan. Tak kalah dengan fitur, desainnya yang simpel dan bagus menjadikan Sublime Text terkesan elegan untuk sebuah *syntax editor*. Sublime text mendukung sejumlah bahasa pemrograman diantaranya C, C++, C#, PHP, CSS, HTML, ASP.

II.17 PHP

PHP adalah singkatan dari *Hypertext Prepocessor*, yaitu sebuah bahasa pemrograman yang berupa kode atau script yang bisa ditambahkan ke dalam Bahasa Pemrograman HTML , PHP itu sendiri sering kali digunakan untuk hal merancang, membuat dan juga memprogram sebuah website. PHP juga sangat sering digunakan untuk membuat sebuah ataupun beberapa CMS, CMS ialah sebuah software atau perangkat lunak yang mempunyai kegunaan untuk memanipulasi semua atau beberapa isi dari sebuah halaman website.

II.18 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Bahasa Inggris: *Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia.. MySQL adalah *server database* SQL (*Structured Query Language)* yang paling banyak diminati karena kecepatan kerja dan kemudahan dalam penggunaannya.

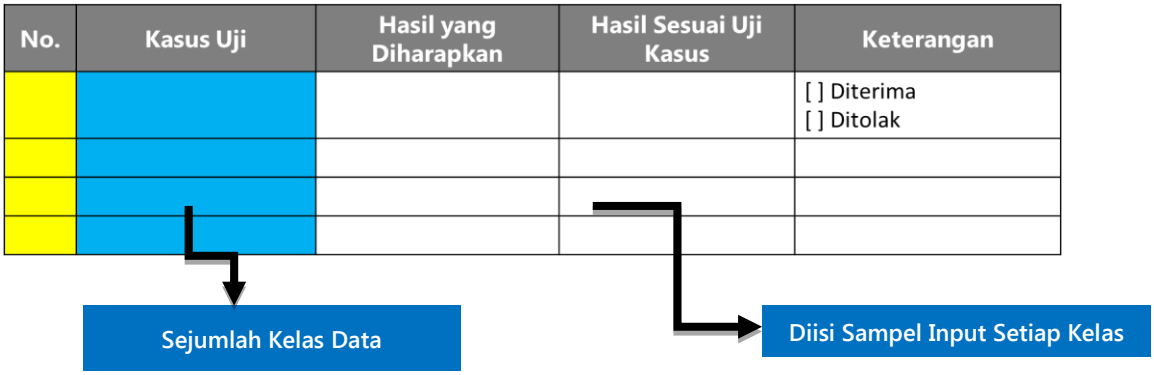
II.19 Pengujian *Black Box*

Metode pengujian *blackbox* fokus pada keperluan penelusuran kesalahan fungsional dari software. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Teknik pengujian black box terdiri dari 10 jenis diantaranya *Equivalence Partitioning, Boundary Value Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing.* Salah satunya yang akan dibahas adalah Equivalence partitioning.

Equivalence partioning merupakan metode ujicoba blackbox yang membagi domain input dari program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Kasus uji penanganan single yang ideal menemukan sejumlah kesalahan (misalnya : kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati. Format pengujian blackbox dapat dilihat pada Tabel II.3



II.20 Pengujian Kuesioner

Kuesioner adalah sebuah daftar pernyataan yang harus diisi oleh orang yang akan dievaluasi (responden) . Metode yang digunakan dalam kuesioner pada penelitian ini adalah skala Likert. Dalam skala likert, responden diminta untuk membaca dengan seksama setiap pernyataan yang disajikan, kemudian ia diminta untuk menilai pernyataan-pernyataan tersebut.

Derajat penilaian responden terhadap suatu pernyataan terbagi dalam 5 kategori yang tersusun secara bertingkat, mulai dari Sangat Tidak Setuju (STS), Tidak Setuju (TS), Ragu-Ragu (R), Setuju (S), dan Sangat Setuju (SS). Atau dapat pula sebaliknya. Pernyataan tiap kuesioner dibuat berdasarkan aspek-aspek yang diteliti. Bobot pemberian skor yang digunakan dapat dilihat pada Tabel II.4

Tabel II.4 Panduan Pemberian Skor

Jenis Pernyataan	Bobot pendapat				
	SS	S	R	TS	STS
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Skor yang telah dihitung pada setiap pernyataan kemudian dikalikan dengan masing-masing bobot tersebut sesuai dengan skenario kuesioner yang telah dibuat. Setelah itu total kan seluruh bobot jawaban tersebut kemudian bagi dengan total responden yang nantinya menjadi nilai rata-rata. Nilai rata-rata inilah yang diambil sebagai acuan sikap dimana jika nilai rata-rata kurang dari 3, maka dapat diartikan responden bersikap negatif dan jika nilai rata-rata lebih dari sama dengan 3, maka dapat diartikan responden bersikap positif terhadap tujuan yang ingin peneliti capai.