BAB II
TINJAUAN PUSTAKA

2.1 *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.

2.2 *Futsal*

Futsal berasal dari bahasa spanyol yaitu dari kata *futbol sala*, artinya sepak bola ruangan. Pada tahun 1930 di Montevideo, Uruguay, futsal diperkenalkan oleh seorang pelatih sepak bola yang bernama Juan Carlos Ceriani. Futsal terbentuk dari ketidakmampuan masyarakat akan lahan untuk bisa bermain sepak bola. Perbedaan futsal dengan sepak bola adalah dari jumlah pemain yang ada dilapangan, ukuran bola, ukuran lapangan dan peraturan yang diberlakukan. Perkembangan futsal sendiri cukup pesat dikarenakan dari peralatan dan peraturan yang sederhana memudahkan masyarakat yang umumnya sulit untuk bermain sepak bola outdoor yang semakin hari semakin sulit ditemukan lapngan yang luas untuk menggelar sebuah pertandingan sepak bola.

2.2.1. Jenis – Jenis Pertandingan Futsal

Futsal sendiri memiliki jenis – jenis pertandingannya tesendiri, yang dimaksud untuk mengenal futsal apa yang sedang kita mainkan termasuk kedalam jenis yang mana. Dan ini adalah jenis – jenis pertandingan futsal tersebut :

1. Fun Futsal

   Contoh pertandingan yang termasuk dalam fun futsal adalah ketika sebuah perkumpulan seperti teman kelas, teman kantor, sebuah komunitas bahkan sebuah instansi yang melakukan pertandingan non-resmi yang tidak menjadikan skor akhir pertandingan sebagai patokan, tidak menjadikan prestasi sebagai tujuan utamanya dan dalam 1 tim bermain hanya mempunyai 2 posisi yaitu posisi penjaga gawang dan pemain yang tidak memiliki posisi yang pasti.

2. Sparring

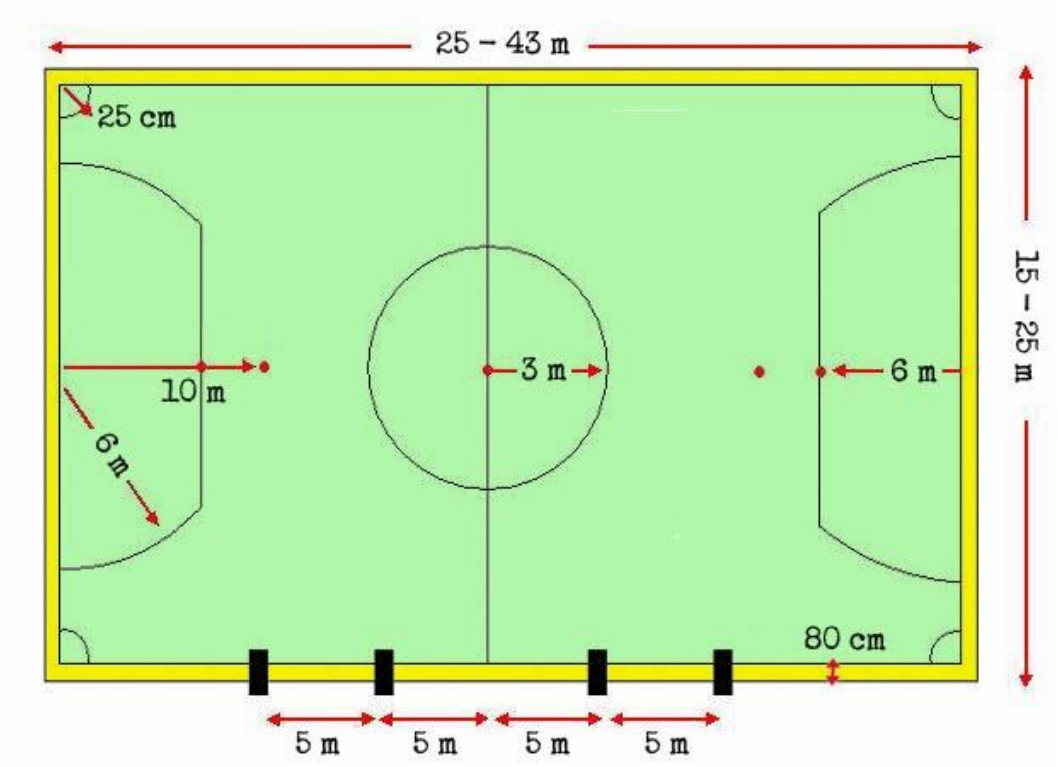
   Sparring sendiri memiliki sebuah artian khusus dimana adanya sebuah pertandingan resmi antar klub atau komunitas yang didalamnya menjadikan skor akhir menjadi penentu bahwa klub yang memenangi pertandingan mempunyai keunggulan dibanding dengan klub yang kalah dan disetiap pemainnya memiliki kemampuan dan posisi yang sesuai dengan gaya bermain masing masing individu.

3. Tournament

   Tournament futsal biasanya mempertemukan antar klub dalam sebuah kompetisi untuk bisa mencari prestasi dan eksistensi sebuah klub yang memenangi tournament tersebut, biasanya klub yang mengikuti sudah mempunyai pengakuan klub yang utuh dari hal materi pemain, jumlah pemain dan manajerial.

2.2.2. Lapangan Futsal

Lapangan Futsal sendiri memiliki standar internasional yang dikeluarkan oleh FIFA sebagai induk futsal itu sendiri, yang dijelaskan pada gambar 2.1 :



Gambar 2. 1 Lapangan Futsal

1. Ukuran Lapangan Futsal

   Lapangan futsal sendiri memiliki panjang dari 25 sampai dengan 43 meter dan memiliki lebar lapangan sepanjang 15 hingga 25 meter

2. Garis Batas Lapangan Futsal

   Garis selebar 8 *centimeter* yaitu garis sentuh di sisi, garis gawang di ujung ujung, dan garis melintang tengah lapangan, dan garis sepanjang 3 meter untuk jari jari dari lingkaran tengah.

3. Ukuran Daerah Penalti

   Busur berukuran 6 meter dari masing – masing tiang gawang, titik penalti sepanjang 6 meter dari titik tengah garis gawang dan titik penalti kedua sepanjang 10 meter dari titik tengah garis gawang.

4. Zona Pergantian Pada Permainan Futsal

   Daerah 5 meter (5 meter dari garis tengah lapangan) pada sisi tribun dari pelemparan.

5. Ukuran Gawang Futsal

   Gawang dalam futsal memiliki tinggi 2 meter dan lebar sepanjang 3 meter

2.2.3. Pembentukan tim futsal

Tim futsal terdiri dari 5 atau lebih pemain yang memiliki kesamaan terhadap suatu jenis tertentu dan untuk mencari pemain yang sesuai untuk membentuk suatu tim futsal tidaklah mudah. Hal ini dikarenakan oleh banyak faktor, yaitu :

1. Lokasi

   Lokasi sangat menentukan tim futsal di dalam berlatih. Hal ini dikarenakan apabila terdapat salah satu anggota tim futsal yang berbeda daerah apalagi jauh dari lapangan futsal itu sendiri, hal ini dapat mempersulit komunikasi antara satu anggota tim futsal dengan anggota lainnya dari tim futsal tersebut.

2. Jenis Pertandingan Futsal

   Jenis pertandingan futsal merupakan salah satu faktor yang sangat mempengaruhi di dalam pembentukan suatu tim futsal. Karena tim futsal merupakan kumpulan para pemain futsal yang sama-sama bermain di dalam satu jenis game yang sama.

3. Posisi Pemain

   Di dalam tim futsal setiap pemain memiliki fungsi dan perannya masing-masing. Pembagian peran atau tugas ini berdasarkan keahlian dari masing-masing pemain futsal tersebut. Pembagian peran ini bertujuan agar di dalam setiap pertandingannya ada yang memiliki strategi ataupun hanya senang senang dalam bermain futsal.

2.2.4. Posisi Pemain

Setiap tim futsal memiliki gaya permainan yang berbeda, namun pada dasarnya setiap pemain futsal memiliki peran dan fungsi masing-masing di dalam timnya. Setiap pemain futsalmemiliki tugas yang berbeda satu sama lain, namun peran tersebut terbagi menjadi 2 bagian utama, yaitu :

1. Penjaga Gawang (GK)

   Pemain futsal di posisi ini memiliki peran penting dalam sebuah tim futsal untuk bisa menjaga gawangnya dengan memakai seluruh tubuhnya agar bisa mengagalkan tim lawan untuk mencetak skor dan memiliki tingkat reflek yang baik dan adapun perlengkapan untuk seorang kiper agar bisa terhindar dari cidera seperti memakai *kneepad, armpad, finger protection* dan *glove* (sarung tangan).

2. Pemain Bertahan (Defender)

   Posisi ini tidak kalah penting seperti penjaga gawang, pemain bertahan memiliki kriteria seperti membatasi ruang gerak pemain lawan untuk melakukan penyerangan ke gawang dan biasanya memiliki postur yang kuat untuk bisa berani dalam benturan yang terjadi dengan pemain lawan, pemain bertahan harus mempunyai skill dalam hal keberanian dan ketenangan dalam bermain.

3. Playmaker

   Tugas dari playmaker adalah mengatur alur permainan tim agar bisa menjadikan permainan tidak *stuck* dalam sebuah situasi tertekan, mempunyai *skill* dalam mengolah bola diatas rata – rata dan mempunyai akurasi operan dan *shooting* yang juga diatas pemain lainnya.

4. Pivot

   Mempunyai insting mencetak gol yang tinggi dan orang pertama yang meredam dibelakang pertahanan lawan ketika ada serangan balik (counter attack).

2.3 *Sistem Informasi*

Sistem adalah kumpulan dari komponen-komponen peralatan model requirements, function dan interface (Mathiassen, Lars, 2000). Sedangkan Sistem Informasi adalah sistem yang menggunakan teknologi komputer untuk mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi. Sistem Informasi merupakan kombinasi teratur apapun dari sumber daya manusia, *hardware*, *software*, data, dan jaringan.

2.3.1. Komponen Sistem Informasi

Sistem informasi dapat digambarkan sebagai sistem yang terdiri dari berbagai komponen. Komponen ini saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran:

1. Komponen Masukan

   Mewakili sejumlah data yang masuk ke dalam sistem informasi. Input tersebut termasuk metode-metode dan media untuk memperoleh data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen Proses

   Terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di database dengan cara tertentu untuk menghasilkan keluaran (output) yang diinginkan.

3. Komponen Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi dan dokumentasi yang dapat digunakan untuk semua tingkatan manajemen dan semua pemakai sistem.

4. Komponen Teknologi

   Teknologi merupakan *tool-box* dalam sistem informasi.Teknologi digunakan untuk menerima input,menjalankan model, menyimpan dan mengakses data,menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari system secara keseluruhan.

5. Komponen Hardware

Hardware berperan penting sebagai suatu peralatan vital bagi sistem informasi yang digunakan untuk menyimpan dan mengolah informasi.

6. Komponen Software

Software berfungsi sebagai kumpulan perintah komputerisasi yang berguna untuk mengolah,menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen Basis data

Database merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan pada perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Pengelolaan database umumnya dikenal dengan nama DBMS *(Database Management System).*

8. Komponen Kendali

Bagian pengendalian dirancang dan diterapkan untuk memelihara system dari hal-hal yang dapat merusaknya, seperti faktor-faktor alamiah temperatur, air, api, debu, virus, sabotase/hijacking, dan sebagainya. [3]

2.3.2. Kualitas Sistem

Kualitas informasi adalah suatu fungsi menyangkut nilai dari keluaran yang dihasilkan oleh suatu sistem yang dirasakan oleh pengguna (Negas et. al. 2003). Oleh karena itu sistem yang akan diimplementasikan harus berkualitas, agar pengguna merasa nyaman dalam penggunaan dan pemanfaatan sistem tersebut (Negash et. al. 2003) menggunakan *interactivity* dan *accses* untuk pengukuran kualitas sistem. *Interactivity* adalah tingkat dimana para pengguna sistem dapat mengambil bagian dalam memodifikasi isi dan format dari suatu lingkungan yang didasarkan pada media yang meliputi umpan balik yang cepat, berbagai alternatif dan tampilan yang dapt berubah. Sedangkan *accses* adalah ketersediaan dari sistem ketika pelanggan mencoba untuk mendapatkan kembali informasi, dan mudah dalam menggunakan penghubung untuk menghubungi orang yang diperlukan (yang mendukung) yang meliputi

kecepatan waktu respon, mudah mendapatkan informasi dan mudah berhubungan dengan manajemen.

2.3.3. Kualitas Informasi

Kualitas sistem adalah mengukur proses informasi melalui sistem yang digunakan, atau keterkaitan anatara karakteristik sistem dengan keberhasilan implementasi (Negash et al, 2003). Oleh karena itu sistem yang akan diimplementasikan harus berkualitas, agar pengguna merasa nyaman dalam penggunaan dan pemanfaatan sistem tersebut. Negash et. al. (2003) menggunakan interactivity dan accses untuk pengukuran kualitas sistem. Interactivity adalah tingkat dimana para pengguna sistem dapat mengambil bagian dalam memodifikasi isi dan format dari suatu lingkungan yang didasarkan pada media. Interctivity meliputi umpan balik yang cepat, berbagai alternatif, dan tampilan yang dapat berubah. Acces adalah ketersediaan dan sistem ketika pelanggan mencoba untuk mendapatkan kembali informasi, dan mudah dalam menggunakan alat penghubung untuk menghubungi orang yang diperlukan (yang m endukung). Access meliputi kecepatan waktu respon, mudah mendapat informasi, mudah berhubungan dengan manajemen. Pada penelitian ini peneliti menggunakan indikator yang sama digunakan oleh Nurniah (2005), yaitu menggunakan indikator gabungan yang dikembangkan Dol and Torkzadeh yang digunakan Li Xiao dan Dasgupta yang mengukur kepuasan penggguna pada sistem informasi berbasis web dan indikator yang digunakan Bailey dan Pearson (1983) yang digunakan mengukur dan menganalisa kepuasan pengguna komputer. Indikator tersebut dijelaskan di bawah ini:

1. *Accuracy* (ketepatan), ketepatan dari suatu sistem dalam mengelola data untuk menghasilkan informasi yang tidak menyesatkan.
2. *Confidence in sistem* (kepercayaan terhadap sistem), sistem yang digunakan oleh pengguna tidak menimbulkan keraguan dalam pengoerasiannya.
3. *Completeness* (kelengkapan), dapat memenuhi seluruh kebutuhan pengguna sistem.

4. *Flexibility* (fleksibel), program yang ada dapat ditambah atau dikurangi sesuai dengan keperluan.
5. *Ease of use* (mudah digunakan), kemudahan dalam pengoperasian system akan memudahkan pengguna dalam menggunakan sistem tersebut.
6. *Integration of system*, kesiapan sistem untuk menghubungkan data dalam areal yang berbeda.
7. *Understanding of systems*, tingkat pemahaman yang dimiliki oleh pengguna terhada sistem atau layanan yang tersedia.
8. *Respon time* (kecepatan respon), waktu yang dibutuhkan oleh sistem untuk merespon input.

2.4 *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:

2.4.1. Android Life Cycle

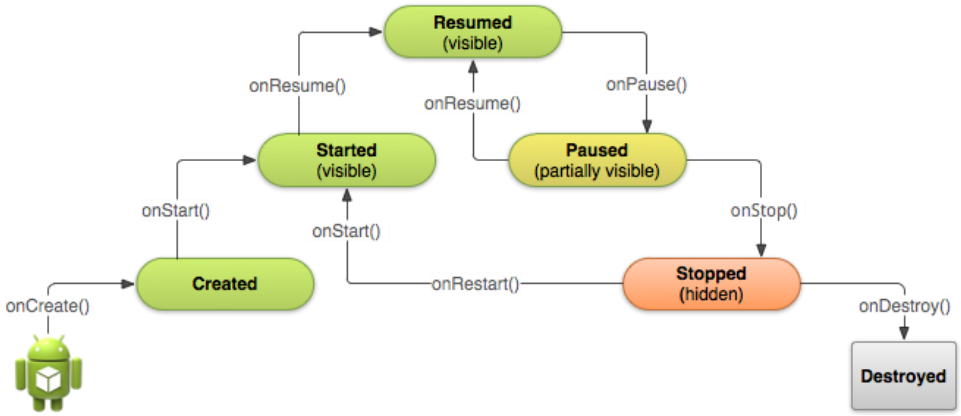
Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada,klasifikasi tersebut berdasarkan kelas-kelas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

Tabel 2. 1 Komponen Aplikasi Mobile

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*.

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 .



Gambar 2. 2 Siklus Hidup Android

Terdapat beberapa *state* dalam siklus hidup android yang terjadi seperti diilustrasikan pada gambar 2.2 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().

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 .

3. Stopped

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

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().

2.4.2. Fitur Android

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.

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*.

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

2.*Touch Gestures* dan *Multi-touch*

Touchscreen adalah *user interface* intuitif yang digunakan banyak *smartphone* didunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. *Multi-touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu. Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu, pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut.

3.*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.

2.4.3. 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 2.3 *Mutiple Assets*, 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.



Gambar 2. 3 *Multiple Assets*

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.



Gambar 2. 4 *Touch Feedback*

2.5 *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.

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.

2.6 *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. [4]

2.7 *GPS*

GPS adalah singkatan dari *Global Positioning System* yang Merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh departemen pertahanan amerika ini digunakan untuk kepentingan militer maupun sipil (survey dan pemetaan). Sistem GPS yang nama aslinya adalah NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System*), mempunyai tiga segmen yaitu:

1. Satellite

Bertugas untuk menerima dan menyimpan data yan ditransmisikan oleh stasiun-stasiun pengontrol, menyimpan dan menjaga informasi waktu ketelitian tinggi (ditentukan dengan jam atomic di satelit), dan memancarkan sinyal dan informasi secara kontinyu ke pesawat penerima (receiver)dari pengguna.

2. Pengontrol

Bertugas untuk mengendalikan dan mengontrol satelit dari bumi baik untuk mengecek kesehatan satelit, penentuan dan prediksi orbit dan waktu, sinkronisasi waktu antar satelit, dan mengirim data ke satelit.

3. Penerima

Bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi (posisi tiga dimensi yaitu koordinat di bumi plus ketinggian), arah jarak dan waktu yang diperlukan oleh pengguna. Ada dua macam tipe penerima yaitu tipe NAVIGASI dan tipe GEODETIC.

Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktip bekerja dan 3 buah sisanya adalah cadangan.

Dalam bidang survey dan pemetaan untuk wilayah terumbu karang, GPS dapat digunakan untuk menentukan posisi titik-titik lokasi penyelaman maupun transek. Posisi yang diperoleh adalah posisi yang benar terhadap sistem koodinat bumi. Dengan mengetahui posisinya yang pasti, lokasi-lokasi penyelaman maupun transek dapat diplotkan kedalam peta kerja.

2.8 *Google Maps*

Google Maps adalah sebuah jasa peta *globe* virtual gratis dan *online* disediakan oleh Google dapat ditemukan di http://maps.google.com. Teknologi tersebut menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia.

Google Maps merupakan Peta *Online* atau Membuka peta secara online. kini dapat dilakukan secara mudah melalui servis gratis dari Google ini. Bahkan, servis ini menyediakan API (Application Programming Interface) yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Tampilan GoogleMaps pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja. [3]

2.9 *JavaScript Object Notation (JSON)*

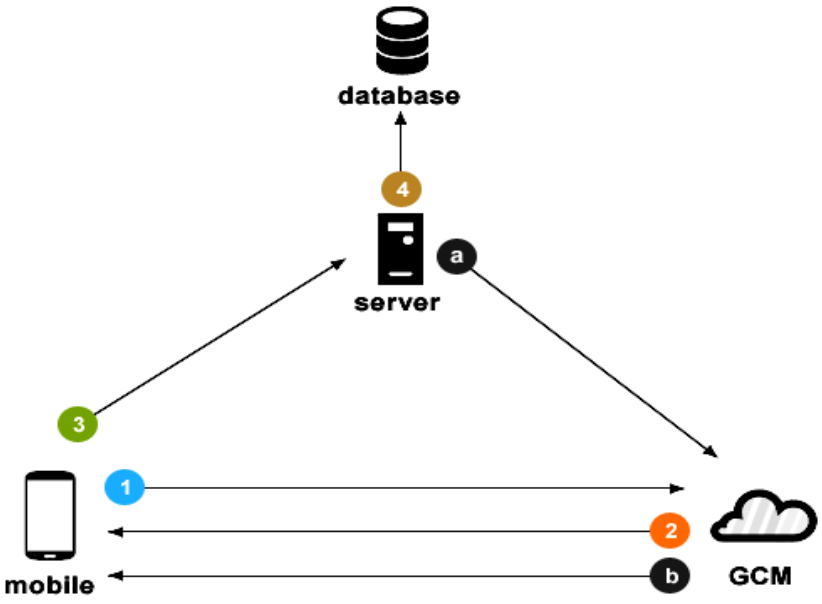
JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript,

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 [9].
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).

2.10 *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*. [6]



Gambar 2. 5 GCM Model

Berdasarkan ilustrasi dari gambar 2.7 GCM Model terdapat dua server yang terlibat dalam proses *push messaging*, pada setiap server yang terlibat terdapat proses pengiriman data yang berbeda-beda diantaranya :

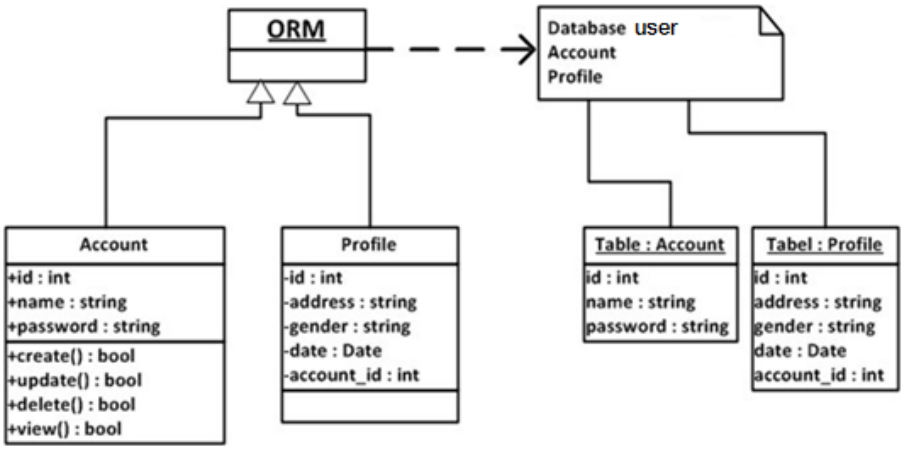
1. Device android mengirimkan sender id dan application id menuju GCM Server untuk registrasi.
2. Setelah berhasil registrasi, GCM server akan mengirimkan registration id menuju device android.
3. Setelah device android mendapatkan registration id, device tersebut akan mengirimkan registration id menuju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

2.11 *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 :

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

Berikut adalah contoh gambar ORM dari penjelasan yang telah dijelaskan sebelumnya [7].



Gambar 2. 6 *Object Relational Mapping(ORM)*

2.12 *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.

2.12.1. Unified Modeling Languange (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 :

1. **Use Case Diagram**

   Use case diagram merupakan suatu pemodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. Use case diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, use case diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponen dalam use case diagram antaranya:

   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 didalamnya. 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.

b. Use Case

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari 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 system.

2. Use Case Scenario

Sebuah diagram yang menunjukkan *use case* dan aktor mungkin menjadi titik awal yang bagus, tetapi tidak memberikan detail yang cukup untuk desainer sistem untuk benar-benar memahami persis bagaimana sistem dapat terpenuhi. Cara terbaik untuk mengungkapkan informasi penting ini adalah dalam bentuk penggunaan *use case scenario* berbasis teks per *use case*-nya. Berikut adalah dasar format penulisan *use case scenario*. Dasar pembangunan *use case scenario* dapat dilihat pada tabel 2.2 Dasar Pembangunan *Use Case Scenario* :

Tabel 2. 2 Dasar Pembangunan *Scenario Use Case*

Use Case Name	Berisi nama dari *Use case* yang akan digunakan	
Goal In Context	Menjelaskan apa yang aktor coba untuk dapatkan dari *Use case*	
Description	Menjelaskan gambaran dari *Use case*	
Related Use Case	Daftar *Use case* yang berhubungan dengan *Use case* tersebut	
Successful End Condition	Kondisi *Use case* jika berhasil	
Failed End Condition	Kondisi *Use case* jika gagal	
Actors	Daftar aktor yang dapat mengakses *Use case*	
Trigger	Aktifitas yang dilakukan untuk mengawali *Use case*	
Main Flow	*Step*	*Action*
	1	Deskripsi urutan aksi dari aktifitas *Use case*
	2	
	3	
Extension	*Step*	*Branching Action*
	2.1	Deskripsi urutan aksi lain selain urutan aksi utama
	2.2	

3. 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 untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private, dan protected.

4. 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. [8]

2.13 *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*.

2.13.1. 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*.

2.13.2. 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 (*tool*) 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. [9]

2.13.3. Cara Kerja LBS

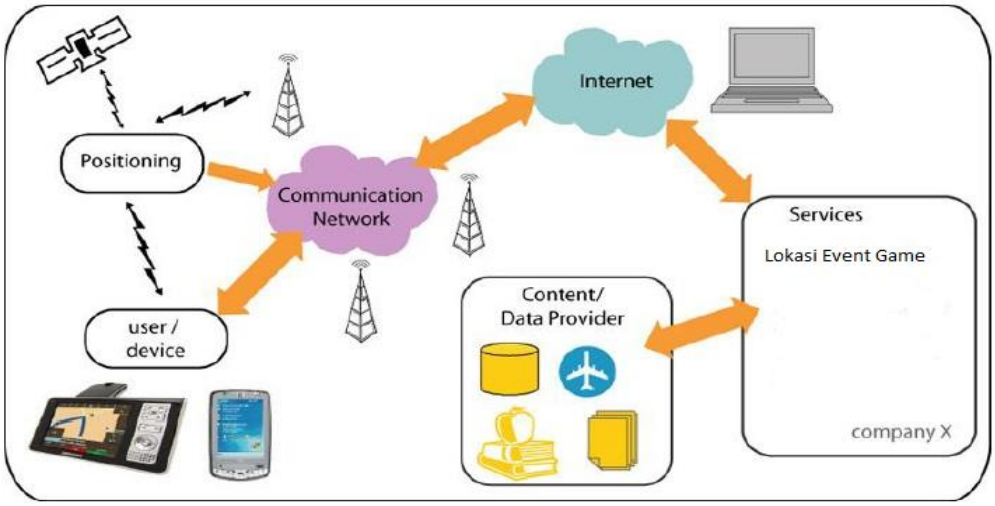
Untuk menggambarkan cara kerja LBS pada perangkat lunak yang akan dibangun pada fitur pencarian lokasi *event games* dapat dijelaskan sebagai berikut :

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 2. 7 Cara Kerja *Location Based Services* (LBS)

2.14 *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.

2.15 *Pengujian White Box*

Ujicoba Whitebox merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Dengan menggunakan metode ujicoba whitebox, para pengembang software dapat menghasilkan kasus-kasus uji yang :

1. Menjamin bahwa seluruh independent paths dalam modul telah dilakukan sedikitnya satu kali
2. Melakukan seluruh keputusan logikal baik dari sisi benar maupun salah
3. Melakukan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya

4. Menguji struktur data internal untuk memastikan validitasnya

Pengujian dilakukan berdasarkan bagaimana suatu software menghasilkan output dari input . Pengujian ini dilakukan berdasarkan kode program. Disebut juga struktural testing atau glass box testing, teknik pengujiannya :

1. Menggambarkan kode program ke dalam graph yaitu node & edge. Jika berhubungan bernilai 1, bila tidak bernilai nol. Dalam pengujian ini akan diperoleh hasil :

a. Kemungkinan source code yang dieksekusi

b. Waktu yang dibutuhkan

c. Memori yang digunakan

d. Sumber daya yang digunakan

2. Basic path, yaitu pengukuran kompleksitas kode program dan pendefinisian alur yang akan dieksekusi.

Digambarkan sequence, if, atau while nya

Uji coba basis path adalah teknik uji coba white box yg diusulkan Tom McCabe. Metode ini memungkinkan perancang test case mendapatkan ukuran kekompleksan logical dari perancangan prosedural dan menggunkan ukuran ini sbg petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Test case yg didapat digunakan untuk mengerjakan basis set yg menjamin pengerjaan setiap perintah minimal satu kali selama uji coba. Data flow testing, untuk mendeteksi penyalahgunaan data dalam sebuah program.

1. Cyclomatic Complexity

Cyclomatic Complexity merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Pada Basis Path Testing, hasil dari cyclomatic complexity digunakan untuk menentukan banyaknya independent paths. Independent path adalah sebuah kondisi pada program yang menghubungkan node awal dengan node akhir.

Terdapat 2 persamaan yang digunakan, yaitu:

V(G)= E - N + 2 atau V(G)= P + 1

Keterangan:

V(G)= cyclomatic complexity untuk flow graph G

E=Jumlah edge(panah)

N=Jumlah node(lingkaran)

P=Jumlah predicate node

Kelebihan White Box Testing

Kesalahan logika. Digunakan pada sintaks ‘if’ dan pengulangan. Dimana White Box Testing akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti. Ketidaksesuaian asumsi. Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki. Kesalahan ketik. Mendeteksi bahasa pemrograman yang bersifat case sensitive. [10]

2.16 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 (ST), Tidak Setuju (T), Tidak Memutuskan (N), 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 [11]

Tabel 2. 3 Panduan Pemberian Skor

Jenis Pernyataan	Bobot pendapat				
	SS	S	N	T	TS
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. Untuk lebih jelasnya dapat dilihat pada rumus dibawah ini.

$$\overline{X} = \frac{\sum Total}{n}$$

Dimana,

$\overline{X} \geq 3\ bersikap\ positif$

$\overline{X} < 3\ bersikap\ negatif$

Keterangan :

$\overline{X}$ = rata-rata

$\sum Total = jumlah\ seluruh\ nilai\ setelah\ dikalika\ dengan\ bobot$

n = total responden

2.17 Metode Haversine

Rumus Haversine adalah persamaan pada navigasi, yang memberikan jarak lingkaran besar antara dua titik bumi berdasarkan garis bujur (longitude) dan garis lintang (latitude). Perhitungan dari rumus ini juga cukup akurat dimana rumus ini mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi. Secara matematis dapat dituliskan didalam persamaan berikut :

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Rumus 2. 1 Rumus Haversine

Keterangan :

$\Delta\sigma$ = Haversine Formula

$\Delta\emptyset$ = Lat1 (Latitude awal) – Lat2 (Latitude Tujuan)

$\emptyset_s$ = Latitude1 (Latitude Awal)

$\emptyset_f$ = Latitude2 (Latitude Tujuan)

$\Delta\lambda$ = Lon1 (Longitude Awal) – Lon2 (Longitude Akhir)

Analisis perhitungan jarak ini menggunakan rumus *haversine* untuk menghitung jarak antara dua titik dengan mengabaikan kondisi geografis seperti lembah ataupun bukit berdasarkan persamaan 1 dimana sudut harus dalam satuan radian untuk menggunakan fungsi trigonometri. [12]

Berikut sejumlah data sarana musik yang digunakan sebagai contoh untuk proses perhitungan jarak :

Tabel 2. 4 Tabel Lokasi Sarana Musik

Sarana Musik	Alamat	latitude	longitude
Sarana 1	Jl. Indramayu Raya No. 68	-6.919695	107.657364
Sarana 2	Jl. Pungkur 97	-6.927554	107.607798
Sarana 3	Jl. Soka no 4	-6.915286	107.628294
Sarana 4	Jl. Surapati No 197	-6.898055	107.633097
Sarana 5	Jl. Depok Raya No 1	-6.914517	107.655424

Berikut merupakan data lokasi pengguna :

Tabel 2. 5 Lokasi Awal

Lokasi Awal	Alamat	latitude	longitude
Lokasi Pengguna	Jl. Purwakarta	-6.917557	107.660445

Dari sejumlah data sarana musik yang ada akan dihitung jarak antara lokasi pengguna dengan setiap lokasi sarana untuk mendapatkan lokasi terdekat yang dalam hal ini terdekat dalam radius 2 km. Untuk mendapatkan lokasi terdekat perlu dilakukan perhitungan jarak pada setiap lokasi yang ada dengan lokasi pengguna setelah itu dilakukan pengecekan jarak dalam radius 2 Km.

Berikut contoh perhitungan lokasi pengguna (Jl. Purwakarta) dengan salah satu lokasi sarana (Jl. Indramayu) pada tabel diatas menggunakan rumus *haversine* pada persamaan 1.

Langkah awal dihitung selisih lokasi pengguna dengan lokasi sarana musik:

Δlat= (-6.919695 – (-6.917557))

= -0.000037315139 (dalam radian)

Δlon= (107.657364 - 107.660445)

= -0.000053773594 (dalam radian)

Kemudian menghitung perpotongan sumbu untuk mengetahui jarak dengan menghitung :

a = $\sin^2$(-0.000037315139 / 2) + (cos-6.917557) * (cos(-6.919695) * $\sin^2$(-0.000053773594 / 2)

= 0.00000000106051512354642 (dalam radian)

c= 2 * Asin2 (√0.000000728901324047745)

= 0.0000651311023681102

Hasil diatas masih dalam radian maka perlu dikalikan dengan radius permukaan bumi sehingga didapat hasil seperti berikut :

d = 6371 * 0.0000651311023681102
= 0.41 Km (pembulatan dua angka di belakang koma)

Perhitungan tersebut dilakukan terhadap lokasi pengguna dengan lokasi – lokasi sarana yang ada sehingga akan didapatkan hasil (jarak) antara lokasi pengguna dan sarana musik yang kemudian di cek ke dalam radius 2 Km untuk mengetahui lokasi mana saja yang terdekat seperti pada tabel berikut :

Tabel 2. 6 Tabel Kesimpulan Perhitungan Jarak

Lokasi Awal	Lokasi Tujuan	Jarak	Kesimpulan
Lokasi Pengguna	**Sarana1**	**0.41 Km**	**Didalam radius 2 Km**
Lokasi Pengguna	Sarana 2	5.92 Km	Diluar radius 2 Km
Lokasi Pengguna	Sarana 3	3.56 Km	Diluar radius 2 Km
Lokasi Pengguna	Sarana 4	3.72 Km	Diluar radius 2 Km
Lokasi Pengguna	**Sarana 5**	**0.65 Km**	**Didalam radius 2 Km**

2.18*Metode TOPSIS*

TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan

prioritas alternatif bisa dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien,dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan [13]. Adapun langkah - langkah untuk menentukkan sebuah nilai dengan menggunakan metode TOPSIS :

1. **TOPSIS dimulai dengan membangun sebuah matriks keputusan**

   Matriks keputusan X mengacu terhadap m alternative yang akan di evaluasi berdasarkan n kriteria. Matriks keputusan X dapat dilihat pada rumus dibawah ini.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i^m x_{ij}^2}}$$

Rumus 2. 2 Topsis – Mencari Nilai Keputusan

Dengan i = 1,2,…….,m dan j = 1,2,…..,n

2. **Membuat solusi ideal positif dan solusi ideal negatif**

   Persamaan yang digunakan untuk dapat menentukan solusi ideal positif dan solusi negatif dapat ditentukan berdasarkan rating bobot ternomalisasi (y_{ij}) :

$$y_{ij} = w_i r_{ij}$$

Rumus 2. 3 Topsis – Mencari Bobot Ternormalisasi

Dengan i = 1,2,…….,m dan j = 1,2,…….,n

$A^+ = (y_1^+, y_2^+, \dots\dots\dots, y_n^+)$

$A^- = (y_1^-, y_2^-, \dots\dots\dots, y_n^-)$

Dengan

$$y_j^+ = \begin{cases} \max_i y_{ij}; \ \ jika\ j\ adalah\ atribut\ keuntungan \\ \min_i y_{ij};\ jika\ j\ adalah\ atribut\ biaya \end{cases}$$

$$y_j^- = \begin{cases} \min_i y_{ij}; \ \ jika\ j\ adalah\ atribut\ keuntungan \\ \max_i y_{ij};\ jika\ j\ adalah\ atribut\ biaya \end{cases}$$

3. **Membuat jarak alternatif Ai dengan solusi ideal positif**

   Jarak antara alternatif Ai dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^{n} (y_i^+ - y_j^i)^2}$$

Rumus 2. 4 Topsis – Solusi Ideal Positif

4. **Membuat jarak alternatif Ai dengan solusi ideal negatif**

   Jarak antara alternatif Ai dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^{n} (y_{ij}^- - y_j^-)^2}$$

Rumus 2. 5 Topsis – Solusi Ideal Negatif

5. **Menentukan nilai preferensi untuk setiap preferensi**

   Menentukan nilai preferensi untuk setiap alternatif yang dirumuskan sebagai berikut :

$$V_i = \frac{D_i}{D_i^- + D_i^+}$$

Rumus 2. 6 Topsis – Nilai Preferensi

6. **Merangking Alternatif**

Alternatif diurutkan dari nilai V_i terbesar ke nilai terkecil.

Alternatif dengan nilai V_i terbesar merupakan solusi yang terbaik

1.