BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan konsep dasar dan teori-teori yang mendukung pembangunan aplikasi . Selain itu, pada bab ini juga dijelaskan tools-tools yang dipakai dalam pembangunan aplikasi pemandu turis.

2.1 Pemandu Turis

Menurut Kamus Besar Bahasa Indonesia (KBBI) pemandu turis adalah penunjuk jalan bagi para wisatawan. Pemandu turis selain sebagai penunjuk jalan juga bertugas memberikan informasi mengenai lokasi yang dikunjungi. Pemandu turis memahami informasi mengenai lokasi yang dikunjungi mulai dari nama, alamat dan detail informasi lokasi tersebut [18].

2.2 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [17].

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga

memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.3 Data dan Informasi

Untuk menuju pada pengertian Sistem Informasi secara utuh, diperlukan pemahaman yang tepat tentang konsep data dan informasi. Keterkaitan data dan informasi sangatlah erat sebagaimana antara sebab dan akibat. Bahwa data merupakan dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

2.3.1 Data

Menurut Al-Bahra Bin Ladjamudin, data dapat didefenisikan sebagai deskripsi dari suatu dan kejadian yang kita hadapi. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan sebagai *file* dalam database. Data akan menjadi bahan dalam suatu proses pengolahan data. Oleh karena itu, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut [1].

Adapun definisi dari kata data adalah suatu istilah majemuk dari *datum* yang berarti fakta atau bagian dari kata yang mengandung arti, yang berhubungan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata angka-angka, huruf-huruf atau simbol-simbol yang menunjukkan ide, objek, kondisi atau situasi. Jelasnya data itu

dapat berupa apa saja dan dapat ditemui dimana saja. Kegunaan data adalah sebagai bahan dasar yang objektif dalam proses penyusunan kebijakan dan keputusan. Dalam kaitannya dengan pengolahan data dengan komputer, pengertian data dapat dibatasi pada fakta-fakta yang dapat direkam. Dalam setiap pengolahan data, data merupakan sumber informasi yang dapat dihasilkan.

2.3.2 Informasi

Menurut Gordon B. Davis, informasi adalah data yang telah diolah menjadi sebuah bentuk yang penting bagi penerima dan mempunyai nilai yang nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau yang akan datang. Informasi akan memiliki arti manakala informasi tersebut memiliki unsur-unsur sebagai berikut: [2]

1. Relevan artinya Informasi yang diinginkan benar-benar ada relevansi dengan masalah yang dihadapi.
2. Kejelasan artinya terbebas dari istilah-istilah yang membingungkan.
3. Akurasi artinya bahwa informasi yang hendak disajikan harus secara teliti dan lengkap.
4. Tepat waktu artinya data yang disajikan adalah data terbaru dan mutahir.

2.4 Basis Data

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka,

deretan karakter, atau simbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut [7]:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disk (disket atau hard disk). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani oleh manusia, sementara basis data dikelola/ditangani melalui perantara alat/mesin pintar elektronis (yang dikenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metoda/cara yang dapat digunakan dalam upaya penyimpanan.

2.4.1 Tujuan Basis Data

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat sebuah basis data yang baik adalah sebagai berikut [7]:

1. Mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali.
2. Tidak adanya redundansi dan menjaga konsistensi data.
3. Pengaturan dalam pemilahan data sesuai dengan fungsi dan jenisnya.

2.4.2 Manfaat dan Kelebihan Basis Data

Manfaat atau kelebihan basis data antara lain [7]:

1. Kecepatan dan Kemudahan (*speed*)

   Dengan menggunkan basis data pegambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokan, mengurutkan bahkan perhitungan dengan matematika.

2. Keakuratan (*Accurate*)

   Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain menekan ketidakakuratan dalam pemasukan atau penyimpanan data.

3. Pemusatan Kontrol Data

Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja. Jika ada perubahan data alamat mahasiswa misalnya, maka tidak perlu kita memperbarui semua data di masing-masing bagian, tetapi cukup hanya disatu basis data.

4. Ketersediaan (*Availability*)

   Dengan basis data kita dapat mem-backup data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi suatu organisasi dari waktu ke waktu membutuhkan media penyimpanan yang semakin besar.

5. Keamanan (*security*)

   Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang mengaksesnya.

6. Kebebasan Data (*Data Independence*)

   Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi atau struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

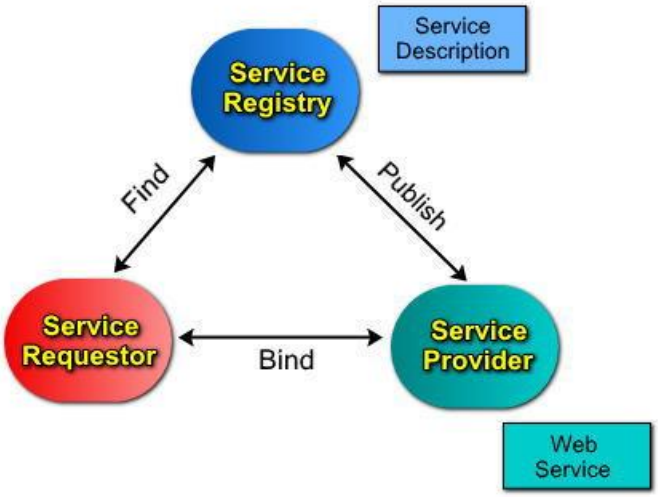
2.5 Web Service

Web service adalah salah satu sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam *format* yang dapat dibaca oleh mesin. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa *compiler*. Namun dalam pengembangannya XML tidak lagi menjadi satu-satunya *format* pertukaran data, JSON(*Java Script Object Notation*) merupakan salah satu format pertukaran data yang dapat diimplementasikan pada *web service* selain XML [21].

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

2.5.1 Arsitektur *Web Service*

Web service dibangun dari tiga komponen utama yaitu *service provider, service registry*, dan *service requestor*. Komponen-komponen tersebut saling berinteraksi melalui artifak web service, yang berupa deskripsi dan implementasi layanan. Terdapat macam tiga operasi yang memungkinkan komponen-komponen tersebut untuk dapat saling berinteraksi, yaitu *publish, find,* dan *bind.*



Gambar 2. 1Arsitektur web service

Pada gambar diatas terdapat 3 komponen utama yang terdapat dalam *web service* yaitu [21]:

1. *Service* Provider, merupakan komponen yang menyediakan akses operasi layanan *web service*.
2. *Service Requestor*, platform yang bertindak sebagai client dari *web service* dan memberikan akses layanan.
3. *Service Registry*, merupakan tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *service registry* bersifat *optional*. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.

2.6 *Location Based Service* (LBS)

LBS sendiri memiliki arti dalam bahasa indonesia adalah layanan berbasis lokasi, yaitu layanan informasi yang didasarkan pada lokasi pengguna saat itu berada. Sistem pada LBS mampu bereaksi aktif pada perubahan entitas posisi dan memberikan informasi letak objek di sekitar pengguna berada. Informasi yang disediakan LBS antara lain [19]:

1. Latitude (sudut jarak dari khatulistiwa)
2. Bujur (sudut jarak dari Greenwich meridian)
3. Ketinggian (di atas permukaan laut)
4. Orientasi (sudut jarak dari utara tiang)

2.6.1 Komponen LBS

Terdapat 5 komponen dalam teknologi LBS, namun hanya 4 diantaranya yang merupakan pendukung utama dalam teknologi LBS, antara lain [19]:

1. Piranti Mobile

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(*Personal Digital Assistant)*, *smartphone*, *laptop*. Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

2. Jaringan Komunikasi

Komponen kedua adalah 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)

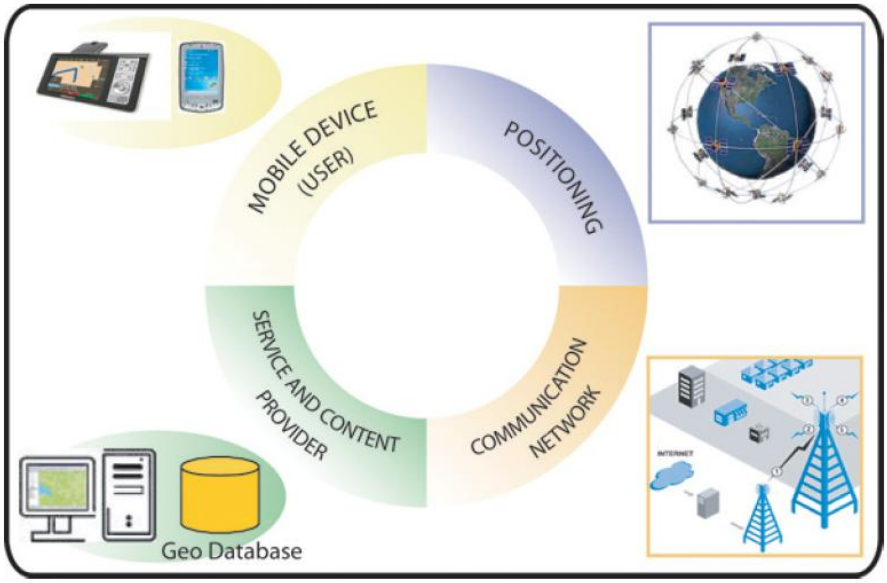
Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

4. Penyedia layanan dan aplikasi

Penyedia layanan 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. Secara lengkap keempat komponen pendukung utama LBS tersebut dapat dilihat pada gambar berikut.



Gambar 2. 2 Komponen pendukung utama teknologi LBS

2.7 Global Positioning System (GPS)

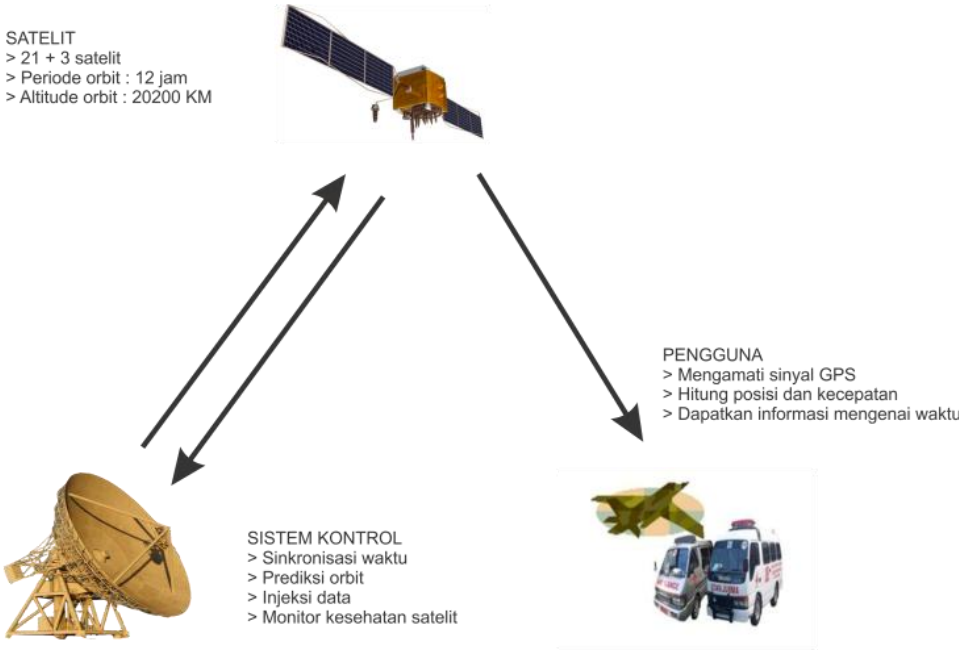
Pembahasan mengenai GPS akan dijelaskan pada sub bab berikut.

2.7.1 Pengertian GPS

GPS adalah sebuah sistem navigasi berbasis radio yang menyediakan informasi berupa koordinat posisi, kecepatan dan waktu kepada pengguna dengan bantuan sinkronisasi satelit. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga dimensi serta informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa tergantung waktu dan cuaca, kepada banyak orang secara simultan. Sistem ini menggunakan satelit yang berfungsi sebagai pengirim sinyal yang berisi informasi koordinat lokasi, kecepatan, arah dan waktu pada alat penerima sinyal GPS(*receiver*) dipermukaan bumi [2].

2.7.2 Arsitektur GPS

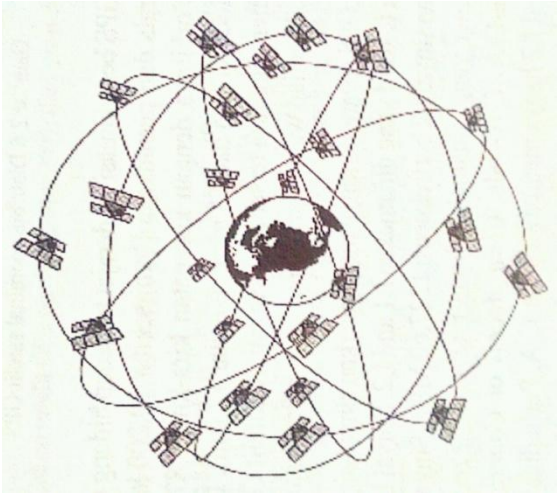
Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyal ke bumi dan ditangkap oleh alat penerima di permukaan bumi. Arsitektur GPS terdiri dari tiga bagian yaitu *Space Segment*, *Control Segment* dan *User Segment* [2].



Gambar 2. 3 Skema sistem penentuan posisi global

1. Space Segment

Space Segment adalah bagian yang terdiri dari kumpulan-kumpulan satelit diluar angkasa yang berada pada orbit bumi. Jarak antara satelit dengan permukaan bumi biasanya sekitar 20.000 Km diatas permukaan bumi. Satelit-satelit ini diatur sedemikian rupa sehingga GPS receiver menerima sedikitnya data dari 6 satelit yang berbeda, namun berada dalam jangkauan orbitnya.



Gambar 2. 4 *Space segment* GPS

Sinyal satelit dapat melewati awan, kaca maupun plastik, tetapi tidak dapat melewati gedung dan gunung. Satelit mempunyai jam atom, dan juga akan memancarkan informasi berupa waktu/jam saat ini. Data ini dipancarkan dengan kode *pseudo-random*. Masing-masing satelit memiliki kodenya sendiri-sendiri. Nomor kode ini biasanya akan ditampilkan pada alat navigasi atau *GPS receiver*, maka dengan nomor kode tersebut pengguna dapat mengidentifikasi sinyal satelit yang sedang diterima alat tersebut. Data ini berguna bagi alat navigasi untuk mengukur jarak antara alat navigasi dengan satelit, yang akan digunakan untuk mengukur koordinat lokasi. Kekuatan sinyal satelit juga akan membantu alat dalam penghitungan. Kekuatan sinyal ini lebih dipengaruhi oleh lokasi satelit, sebuah alat akan menerima sinyal lebih kuat dari satelit yang berada tepat diatasnya dibandingkan dengan satelit yang berada di garis cakrawala.

Ada dua jenis gelombang yang saat ini dipakai untuk alat navigasi berbasis satelit. Pada umumnya, yang pertama lebih dikenal dengan sebutan L1 pada 1575.42

MHz. Sinyal L1 ini yang akan diterima oleh alat navigasi. Satelit juga mengeluarkan gelombang L2 pada frekuensi 1227.6 Mhz. Gelombang L2 ini digunakan untuk tujuan militer dan bukan untuk umum [2].

2. Control Segment

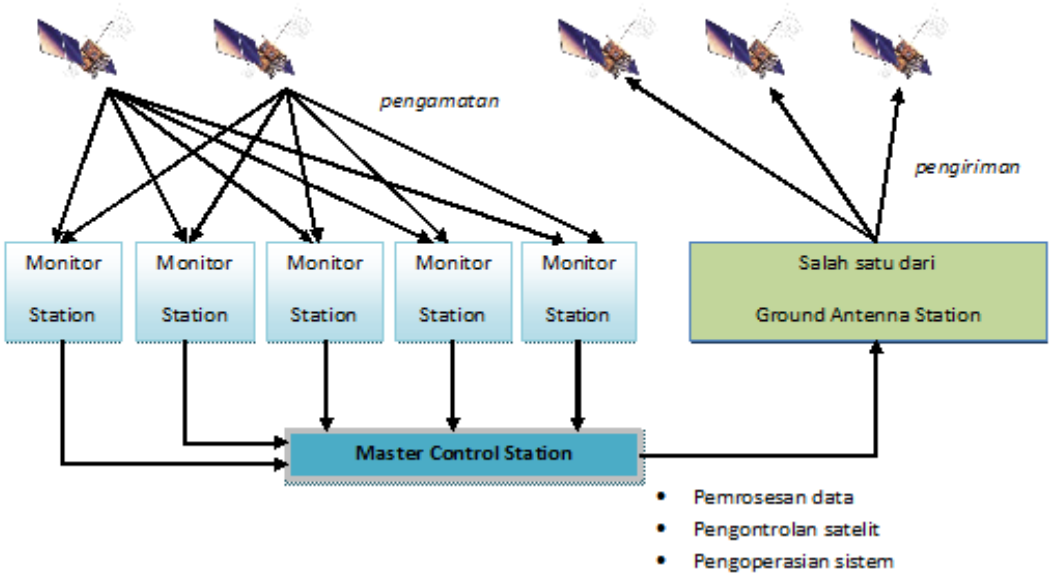
Segmen sistem control GPS berfungsi mengontrol dan memantau operasional semua satelit GPS dan memastikan bahwa semua satelit berfungsi sebagai mana mestinya. Secara lebih spesifik tugas utama dari segmen system control GPS adalah [2]:

1. Secara kontinyu memantau dan mengontrol sistem satelit.
2. Menentukan dan menjaga waktu system GPS.
3. Memprediksi empheris satelit serta karakteristik jam satelit.
4. Secara periodik meremajakan (*update*) *navigation message* dari setiap satelit.
5. Melakukan manuver satelit agar tetap berada dalam orbitnya, atau melakukan relokasi untuk menggantikan satelit yang tidak 'sehat', seandainya diperlukan.

Secara spesifik, segmen sistem control terdiri dari *Ground Antenna Stations* (GAS), *Monitor Stations* (MS), *Prelaunch Compatibility Station* (PCS), dan *Master Control Station* (MCS) [*Bagley and Lamons*, 1992]. Dalam segmen sistem kontrol GPS, MS bertugas mengamati secara kontinyu seluruh satelit GPS yang terlihat. Pada prinsipnya stasiun MS tidak melakukan pengolahan data, tetapi hanya mengirimkan

data pseudorange serta pesan navigasi yang dikumpulkan ke MCS untuk diproses secara *real-time*.

Seluruh data yang dikumpulkan oleh MS kemudian dikirimkan ke MCS untuk diproses guna memperoleh parameter-parameter dari orbit satelit dan waktu, serta parameter-parameter penting lainnya. Disamping tugas pengolahan data tersebut, MCS juga bertanggung jawab dalam pengontrolan pergerakan satelit dalam orbitnya serta status kesehatannya. Hasil perhitungan tersebut kemudian dikirimkan ke salah satu GAS, dimana untuk selanjutnya informasi tersebut beserta data lainnya dikirimkan ke satelit-satelit GPS yang dapat dilihat oleh stasiun GAS.



Gambar 2. 5 Skema kerja sistem control GPS

3. User Segment

Segmen pengguna ini terdiri dari para pengguna satelit GPS, baik di darat, laut, udara, maupun di angkasa. Dalam hal ini, alat penerima sinyal GPS(*GPS receiver*)

diperlukan untuk menerima dan memperoses sinyal dari satelit GPS untuk digunakan dalam penentuan posisi, kecepatan, waktu maupun parameter turunan lainnya [2].

Satelit akan memancarkan data almanak dan ephemeris yang akan diterima oleh alat navigasi secara teratur. Data almanak berisikan perkiraan lokasi (approximate location) satelit yang dipancarkan terus menerus oleh satelit. Data ephemeris dipancarkan oleh satelit, dan valid untuk sekitar 4-6 jam. Untuk menunjukkan koordinat sebuah titik (dua dimensi), alat navigasi memerlukan paling sedikit sinyal dari 3 buah satelit. Untuk menunjukkan data ketinggian sebuah titik (tiga dimensi), diperlukan tambahan sinyal dari 1 buah satelit lagi.

Dari sinyal-sinyal yang dipancarkan oleh kumpulan satelit tersebut, alat navigasi akan melakukan perhitungan-perhitungan, dan hasil akhirnya adalah koordinat posisi yang akan diteralat tersebut. Makin banyak jumlah sinyal satelit yang diterima oleh sebuah alat, akan membuat alat tersebut menghitung koordinat posisinya dengan lebih tepat.

Karena alat navigasi ini bergantung penuh pada satelit, maka sinyal satelit menjadi sangat penting. Alat navigasi berbasis satelit ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit [2].

2.7.3 Metode Penentuan Lokasi

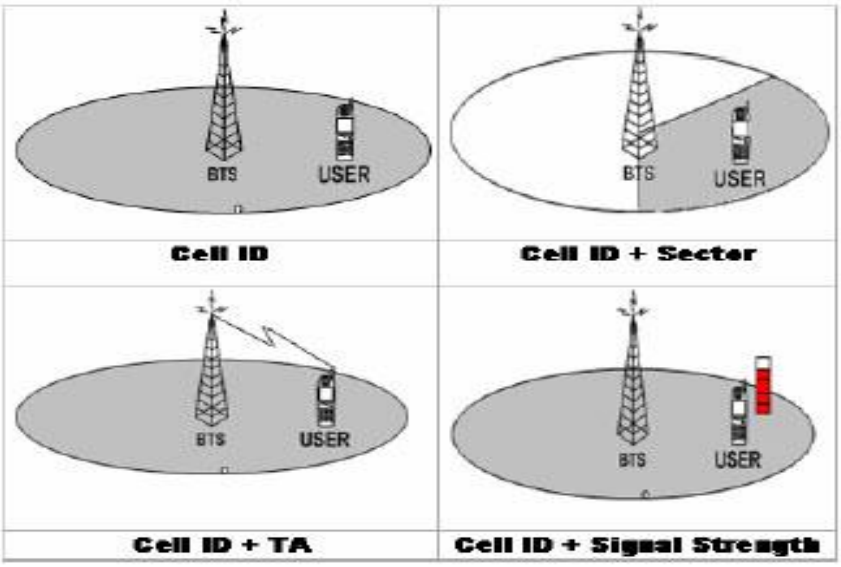
Ada 3 metode dalam penentuan lokasi yang hingga saat ini masih terus dipakai. Ketiga metode tersebut adalah *Metode Basic Postioning* yang berbasis pada *Cell Identification*, *Enhanced Positioning* dan *Advanced Positioning* [5].

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

Metode *Cell ID* merupakan metode yang paling sederhana dalam menentukan keberadaan *handset*. Penentuan posisinya didasarkan pada kenyataan bahwa daerah geografis yang tercakup oleh sebuah *cell* berhubungan dengan daerah cakupan dari sinyal radio. Ketika sebuah handset terhubung secara aktif dengan *base station*, berarti handset tersebut diasumsikan berada dalam daerah yang mampu dicakup oleh *base station* tersebut atau berada dalam sebuah *cell* dari *base station* tersebut. Untuk mengukur jarak dan arah handset dari *base station* tidak dapat diketahui dengan pasti. Oleh karena itu, untuk lebih meningkatkan lagi akurasi pencarian, metode Cell ID sering dikombinasikan dengan metode lain misalnya [5]:

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

2. *Network Measurement Report* (*NMR*), dengan berdasar pada besar kecilnya sinyal (*Received Signal Strength*) yang diterima handphone yang ada di suatu "*sector cell*", maka posisi itu dapat ditentukan lebih akurat.

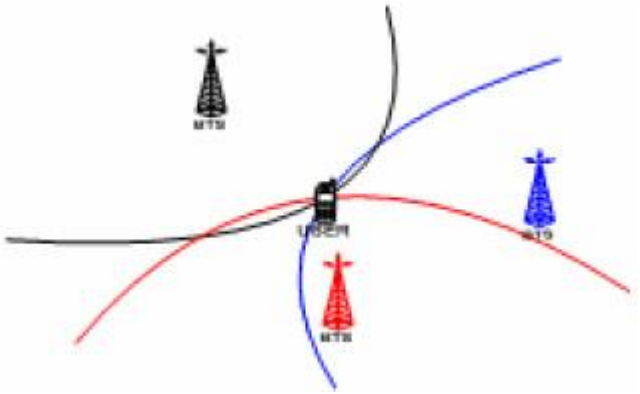


Gambar 2. 6 Penentuan lokasi menggunakan metode Cell ID

b) Metode *Echanced Positioning.*

Pada umumnya metode ini menggunakan pendekatan *Observe Time Difference* (OTD). Dalam jaringan GSM yang sering digunakan adalah *Enhanced*-OTD (E-OTD). E-OTD adalah metode pencarian posisi yang berdasarkan pada waktu. Untuk menentukan posisi handset aktif terdapat 3 *base station*, perlu ditentukan terlebih dahulu jarak *handset* terhadapa masing-masing *base station* berdasarkan waktu yang ditempuh oleh sebuah sinyal ke *handset* oleh masing-masing *base station* [5].

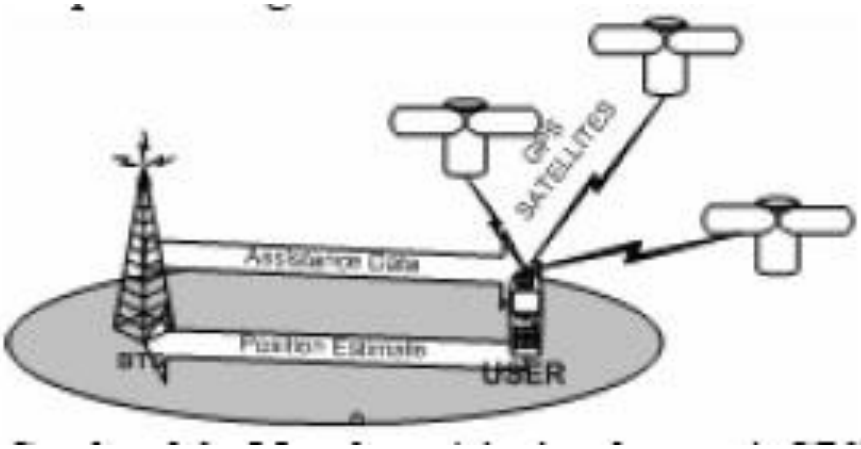
Untuk selanjutnya, dengan menggunakan rumus tringulasi matematika maka dapat ditentukan posisi dari *handset* yang sedang aktif tersebut. Dengan menggunakan metode ini akurasi akan meningkat hingga memiliki ketelitian sampai kurang dari 50m.



Gambar 2. 7 Penentuan lokasi menggunakan metode E-OTD

c) Metode *Advanced Positioning*.

Metode *Advanced Positioning* merupakan metode yang paling tinggi akurasinya dalam menentukan lokasi dibandingkan metode lainnya. Metode ini pada umumnya menggunakan teknologi *Assisted Global-Positioning System*(A-GPS). A-GPS merupakan metode yang berbasis pada waktu. Pada metode ini akan dilakukan pengukuran waktu tiba dari sebuah sinyal yang dikirimkan dari 3 buah satelit GPS. A-GPS jaul lebih efisien dan efektif dalam mengakses informasi dari satelit karena tidak perlu mencari data satu persatu dari keseluruhan satelit GPS yang ada, namun A-GPS telah mengetahui satelit mana yang dibutuhkan atau dituju. Metode ini hanya bisa diterapkan khusus pada *handset* yang dilengkapi dengan fasilitas GPS [5].



Gambar 2. 8 Alur sistem metode *advanced positioning*

2.7.4 Kelebihan dan Kekurangan GPS

Teknologi GPS merupakan teknologi yang sangat fenomenal dalam bidang penentuan posisi karena mampu memberikan informasi mengenai posisi secara *real-time* dan kontinu, dimana saja dan kapan saja. Di samping itu, ada beberapa hal yang membuat GPS sangat baik dalam sistem pelacakan [2]:

1. Tidak tergantung waktu dan cuaca.
2. Meliputi wilayah yang luas.
3. Tidak terpengaruh topografis.
4. Memberikan ketelitian akurasi yang cukup.
5. Penggunaan tidak dikenakan biaya.

Namun, GPS juga memliki kekurangan yaitu mengharuskan GPS *receiver* dalam *Line-of-Sight* setidaknya 3 buah satelit GPS untuk menentukan lokasi. Dengan kekurangan yang dimiliki oleh GPS tersebut, maka GPS hanya ideal untuk digunakan dalam *outdoor positioning*.

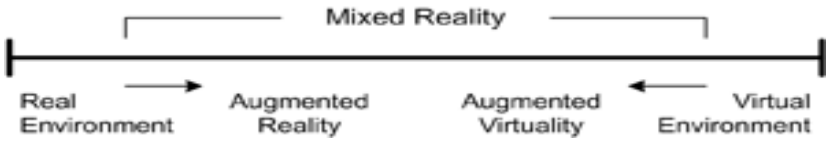
2.8 Augmented Reality

2.8.1 Pengertian Augmented Reality

Augmented Reality adalah penggabungan objek-objek digital baik itu berupa dua dimensi maupun tiga dimensi dengan dunia nyata yang bersifat interaktif secara real time. Objek-objek digital yang menampilkan informasi tidak dapat diterima pengguna oleh inderanya sendiri. Informasi yang ditampilkan oleh objek-objek digital hanya sebagai alat yang membantu persepsi dan interaksi penggunanya dengan dunia nyata. Sehingga informasi tersebut membantu penggunanya melaksanakan kegiatan dalam dunia nyata [11].

2.8.2 Arsitektur Augmented Reality

Pada tahun 1994, Milgram dan Kishino merumuskan kerangka kemungkinan penggabungan dan peleburan dunia nyata dan dunia maya ke dalam sebuah ***v****irtuality continuum.* Dalam kerangka tersebut,*augmented reality* lebih dekat ke sisi kiri yang menjelaskan bahwa lingkungan bersifat nyata dan benda bersifat maya. Sebaliknya *augmented virtuality* lebih dekat ke sisi kanan dalam kerangka tersebut, yang menjelaskan bahwa lingkungan bersifat maya dan benda bersifat nyata. Sehingga jika terjadi penggabungan antara a*ugmented reality* dengan *virtual reality* akan tercipta *mixed reality* [11].



Gambar 2. 9 Konsep kerangka *augmented reality*

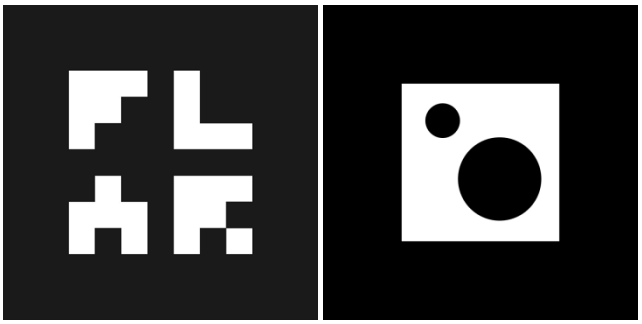
2.8.3 Konsep Augmented Reality

Augmented reality memiliki 2 metode yang sangat signifikan berkembang dalam beberapa tahun belakangan ini yaitu *marker based tracking* dan *markerless augmented reality.*

2.8.3.1 Marker Based Tracking

Augmented reality pada *marker based tracking* merupakan metode yang dikembangkan sejak tahun 80an. *Marker* merupakan ilustrasi dari hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* dan menciptakan dunia *virtual* 3D yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z [6].

Salah satu pengembang *augmented reality* yang populer dengan menggunakan metode *marker based tracking* saat ini adalah ARToolkit. ARToolkit telah dikembangkan sejak tahun 90an. Salah satu pengembangan dari ARToolkit adalah FLARToolkit.

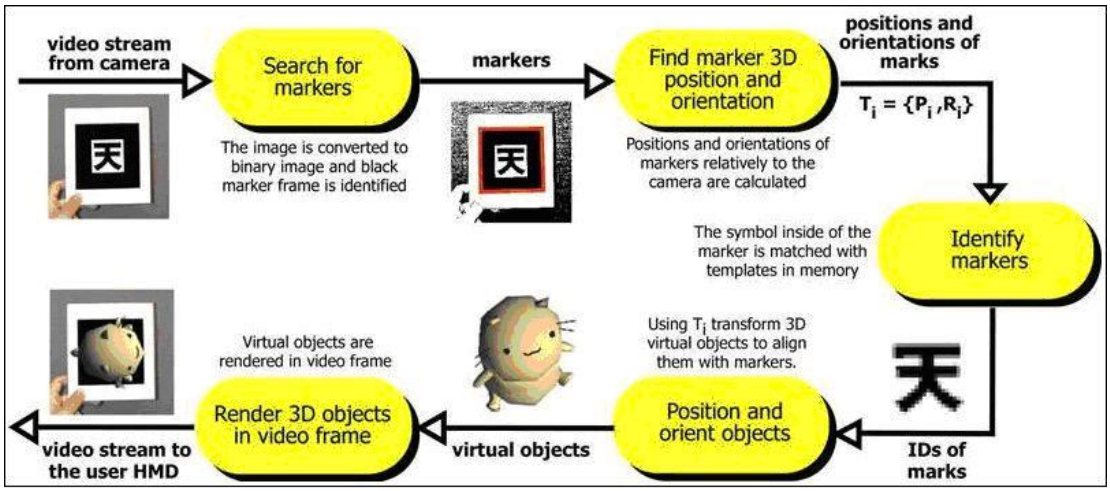


Gambar 2. 10 Macam bentuk *marker augmented reality*

1) ARToolkit

ARToolkit merupakan *library* perangkat lunak yang dikembangkan oleh Dr.Hirokazu Kato. ARToolkit dapat mengenali pola *marker* yang berbentuk persegi hitam dan putih yang ditengah nya memiliki *pola image pattern* yang beraturan dengan batas hitam tebal dan latar belakang putih [16].

a. Cara Kerja ARToolkit



Gambar 2. 11 Konsep marker *augmented reality*

Pada Gambar 2.11 merupakan alur proses pendeteksian *marker* untuk dapat merender objek *augmented reality*. Penjelasannya sebagai berikut :

1. Kamera menangkap gambar *marker* dari dunia nyata secara *real time.*
2. Perangkat lunak dalam komputer mencari *marker* pada masing-masing *frame* video.
3. Jika ditemukan sebuah garis tepi dari *marker*, maka digunakan perhitungan matematika untuk menghitung posisi kamera relatif terhadap kotak warna hitam.
4. Tiap kali posisi kamera diketahui maka model grafis hasil rekayasa komputer digambarkan pada posisi yang sama.
5. Model tersebut ditampilkan di atas gambar video dunia nyata dan tampak menempel pada *marker* yang berbentuk kotak.
6. Hasil akhir menunjukkan jika user melihat pada tampilan, maka akan tampak objek virtual tambahan pada dunia nyata.

b. Algoritma Gaussian Detrivative

Sebelum mendeteksi bentuk *marker* ARToolkit melakukan proses thresholding agar dapat dibaca dengan mudah bentuk dan tata letak *marker* di dunia nyata. Proses *thresholding* adalah proses dimana gambar yang ditangkap di ubah menjadi warna hitam dan putih sehingga mempermudah penyeleksian pola bentuk *marker*. Dengan menggunakan persamaan Gaussian detrivative, ARToolklit akan mendeteksi wilayah yang ditangkap kamera kemudian dirubah menjadi gambar berbentuk threshold(hitam putih). Setelah gambar sekitar diolah menjadi thresholding maka pendeteksian bentuk pengenalan *marker* menjadi lebih mudah, karena warna pola dan bentuk pada

marker berwarna hitam dan putih. Persamaan Gaussian detriative kernel yang digunakan yaitu [6]:

[-3 -5 0 5 3] * A

Jika garis tepi *marker* telah terbaca, kemudian persamaan baru dimasukkan untuk dapat membaca objek dan background pada *marker*.

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

Keterangan :

G_y = Objek pixel

G_x= *Background* pixel

c. Algoritma RANSAC

Untuk mendeteksi garis tepi pada *marker*, ARToolkit juga menggunakan algoritma RANSAC(*Random Sample Consesus*). Dengan algoritma RANSAC, ARToolkit dapat mendeteksi lebih dari 1 *marker* karena RANSAC merupakan algoritma yang dapat membaca setiap garis tepi melalui persamaan yang telah ditetapkan [6].

d. Kekurangan ARToolkit

Algoritma-algoritma yang digunakan ARToolkit juga memiliki beberapa kelemahan, yaitu:

1. Tidak begitu cepat dalam menangkap pola dan bentuk *marker*.
2. Tidak begitu baik dalam mendeteksi garis tepi dan garis sudut dari sebuah *marker*.

2) FLARToolkit

FLARToolkit merupakan hasil portingan ARToolkit dari JAVA ke FLASH. Seperti pada ARToolkit yang memakai bahasa pemrograman JAVA dalam pengembangannya, FLARToolkit memakai Action Script untuk mengembangkan *augmented reality* yang akan dibangun [6].

2.8.4 *Marker*less Augmented Reality

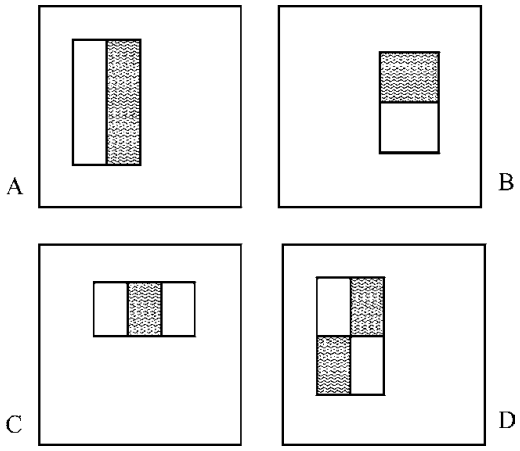
Berbeda dengan *Marker Based Tracking*, metode *Markerless Augmented Reality* tidak lagi memerlukan penggunaan sebuah *marker* untuk menampilkan objek-objek digital. Metode ini memiliki beberapa teknik-teknik khusus yaitu *Face Tracking, 3D Object Tracking, Motion Tracking* dan *GPS Based Tracking.*

1. Face Tracking

Ciri pada wajah setiap manusia berbeda-beda, namun pasti setiap manusia memilki mata, hidung dan mulut. Penentuan titik koordinat bentuk pola wajah menggunakan algoritma Viola-Jones. Dengan algoritma Viola-Jones, teknik ini dapat mengenali pola pada mata, hidung dan mulut dan mengabaikan objek sekitarnya sehingga objek digital dapat diimpelementasikan pada wajah melalui sebuah kamera.

A. Algoritma Viola-Jones *Face Detection*

Pendeteksian bentuk pola wajah dengan menggunakan algoritma ini adalah berdasarkan penentuan wajah dengna menggunakan *image features. Image features* adalah pengelompokkan bentuk pola wajah dengan menggunakan bentuk objek persegi panjang yang didalamnya dipilah berdasarkan warna putih dan abu-abu.



Gambar 2. 12 Konsep integral image untuk face tracking

Dapat dilihat pada gambar diatas, wajah akan diolah dengan persegi panjang yang mana jumlah *pixel* persegi panjang yang berwarna putih dikurangi dengan jumlah *pixel* yang berwarna abu. Objek persegi panjang yang berperan sebagai perantara tata letak wajah tersebut mempercepat kinerja perhitungan dalam menentukan posisi dari setiap bentuk pola wajah.

B. Kelebihan dan Kekurangan Algoritma Viola-Jone

Kelebihan yang dimiliki algoritma Viola-Jone dalam penerapannya pada teknologi *augmented reality* adalah :

1. Perhitungan integral gambar dengan menggunakan penentuan objek berdasarkan persegi panjang cepat. Sehingga penentuan posisi wajah dapat dengan cepat dikenali.
2. Dapat dengan cepat menolak bentuk wajah, karena dukungan integral gambar dengan objek persegi panjang.

Kekurangan dari algoritma Viola-Jones :

1. Sangat mudah dan kuat untuk mendeteksi wajah depan yang berada di bawah sinar cahaya. Pendeteksian di bawah cahaya yang sangat mudah cepat tidak begitu terlihat significan ketika berada di daerah yang tidak terlalu didukung oleh cahaya, sehingga pendeteksian wajah tidak terlalu maksimal hasilnya.
2. Penerapan Algoritma komputasi sangat mahal.
3. Tidak Cocok diimplementasikan ke *platform mobile*.

2. 3D Object Tracking

Berbeda dengan *Face Tracking* yang hanya mengenali pola wajah secara umum, teknik *3D object Tracking* dapat mengenali semua bentuk benda yang ada disekitarnya seperti televisi, handphone, mobil, rumah dan lain-lain. Teknik ini dapat memvisualisasikan hal-hal umum yang salah satu contohnya memberikan informasi petunjuk penggunaan bagaimana membuka katup AC pada mobil.



Gambar 2. 13 Penerapan augmented reality dengan teknik 3D object tracking

3D object Tracking menggunakan hubungan geometris sebagai pendeteksi bentuk objek 3D. Hubungan geometris digunakan karena pada *3D object Tracking* hanya menggunakan fungsi kamera untuk mendeteksi benda-benda nyata yang ada di sekitar sebagai sebuah *marker*. Ada 2 Metode pengembangan *3D object Tracking* yang dipakai yaitu *bottom-up approach*(BUA) dan *top-down approac*h(TDA). 2 metode ini cocok untuk dipakai dalam penentuan titik koordinat bentuk objek 3D berbasis kamera.

1. *Bottom-up Approach*(BUA)

Bottom-up Approach harus melewati 2 langkah untuk menentukan posisi objek 3d, yaitu:

1. Pelacakan titik koordinat

   Pelacakan titik koordinat sebuah objek dicocokkan pada titik koordinat yang telah di simpan di dalam database. Ketika titik posisi sebuah objek 3D dikenali maka BUA akan dengan cepat mengenalinya. BUA adalah sebuah metode berbasis titik yang dapat dengan mudah terdeteksi seperti titik sudut atau titik tepi sebagai titik acuan sebuah objek 3D.

2. Perhitungan parameter kamera

   Untuk dapat memperkirakan pergerakan kamera maka posisi objek 3D yang telah memiliki titik koordinat acuan dibaca oleh system. Ketika setidaknya tiga titik acuan yang terdeteksi, maka akan dilakukan perhitungan dari objek yang ditangkap oleh kamera untuk memprediksi pergerakan kamera.

Metode BUA dapat dengan cepat menghitung koordinat titik-titik objek 3d dengan sangat akurat dan biaya yang dikeluarkan dalam pengembangannya juga relative rendah. Namun, BUA sulit untuk mendeteksi dan melacak posisi titik acuan objek 2D karena ketika kamera bergerak titik-titik yang telah menjadi titik acuan sebuah objek 2D tidak akan sama dengan titik-titik koordinat yang telah ada di dalam database. Hal itu menyebabkan kesalahan dalam pelacakan sebuah objek, sehingga objek yang dijadikan sebagai *marker* tidak dapat menampilkan sebuah bentuk objek augmented reality.

2. *Top-down Approach*(TDA)

Metode TDA dapat melacak sebuah objek ketika objek tersebut tertutup sesuatu(tidak sepenuhnya tertutup), jadi metode ini dapat memperkirakan kalau objek tersebut adalah objek yang dijadikan *marker* walaupun objek tersebut tertutup sesuatu atau ketika kamera digerakkan secara cepat. Hal itu didukung oleh *Inertial Orientation Sensors*, yang merupakan salah satu keunggulan dari metode TDA ini. *Inertial Orientation Sensors* dapat mengenali posisi awal sebuah objek, jadi walaupun kamera tersebut digerakkan *Inertial Orientation Sensors* masih dapat membaca titik koordinat sebuah objek.

3. Motion Tracking

Metode *marker*less dengan teknik ini yaitu dengan menangkap gerakan yang dilakukan pengguna. Salah satu contoh Augmented Reality dengan menggunakan teknik ini yaitu pada film Avatar. Dunia film sudah menggunakan teknologi ini dengan sangat ekstensif karena dapat mensimulasikan gerakan secara real-time.



Gambar 2. 14 Penerapan motion tracking simulasi mengendara mobil

Motion tracking lebih dekat ke dunia *virtual reality*, karena di sini hasil yang tampak hampir keseluruhan dunia virtual. Dalam penerapannya, *motion tracking* memerlukan sebuah alat pendeteksi gerakan dari tubuh penggunanya. Salah satu alat tersebut adalah *Head Mounted Display*(HMDs), alat ini berfungsi sebagai indra penglihatan dunia *virtual*.

Metode pengembangan yang dipakai dalam *motion tracking* yaitu metode TDA. Metode ini dibutuhkan karena *inertial orientation sensors* yang menjadi solusi dalam menangkap sebuah gerakan.

4. GPS Based Tracking

Pengembangan teknik ini lebih diarahkan pada *smartphone*, karena teknologi GPS dan kompas yang tertanam pada *smartphone* tersebut. Dengan memanfaatkan fitur GPS yang berfungsi sebagai penentu lokasi pengguna pada saat itu berada sehingga lokasi terdekat yang ingin dituju dapat dilihat melalui implementasi augmented reality.

Teknik ini berguna sebagai pemandu selayaknya fungsi GPS, namun dilengkapi dengan *marker* informasi arah yang dituju. Dalam implementasinya, teknik ini mengharuskan tersambungnya koneksi GPS dan kebutuhan paket data yang

ada pada *smartphone*, karena data-data lokasi yang dimiliki GPS memiliki akses langsung dari satelit agar cepat mendeteksi wilayah yang telah dijadikan sebuah objek *marker* informasi pada *Augmented Reality*. Akses internet memiliki fungsi sebagai pemanggilan data-data berupa *latitude, longitude*, serta informasi yang mendukung setiap lokasi yang disimpan pada *server* sehingga beban ukuran aplikasi dapat diminimalisir [8].

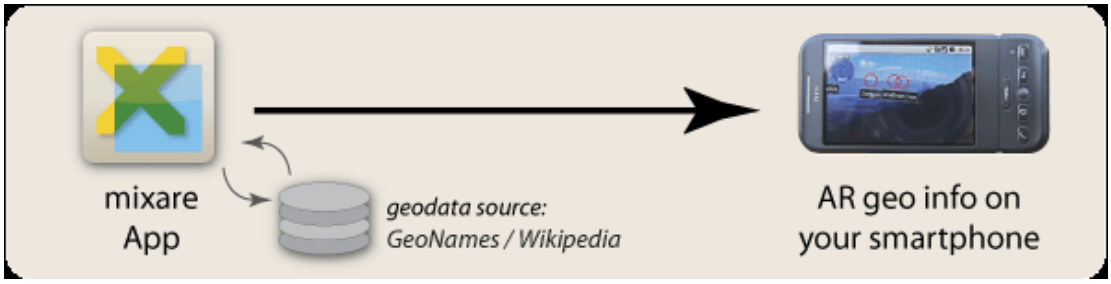
Teknik GPS *based tracking* sebenarnya membutuhkan peran kompas dan akselerometer sebagai pengatur ukuran layar secara *horizontal* dan *vertical* agar marker lokasi dapat dilihat ketika kamera *handset* berada posisi yang sesuai dengan lokasi tersebut. Namun ketika *handset* tidak berada dalam sudut pandang lokasi tersebut maka *marker* tersebut tidak akan tampak.

Kebanyakan teknik GPS *based tracking* sudah memiliki *engine* pembantu(AR *browser*) yang telah dikembangkan oleh beberapa perusahaan sehingga mempermudah untuk mengembangkan teknik ini sesuai dengan keinginan. Pada tugas akhir ini digunakan AR Mixare *Browser* sebagai engine untuk mengembangkan aplikasi menjadi lebih mudah digunakan.

a. Mixare

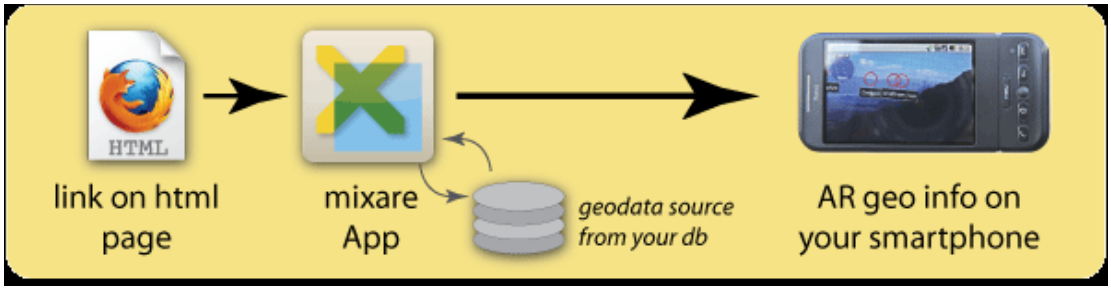
Mixare adalah sebuah browser augmented reality open source yang dipublikasikan dibawah lisensi GPLv3. Mixare dapat dengan mudah diintegrasikan dengan data yang dimiliki sendiri maupun dengan data yang sudah tersedia seperti Wikipedia. Pengembangan Mixare dapat dilihat sebagai berikut [20]:

1. Mixare merupakan aplikasi yang dapat berdiri sendiri dengan *point of interest* Wikipedia sebagai pusat informasinya.



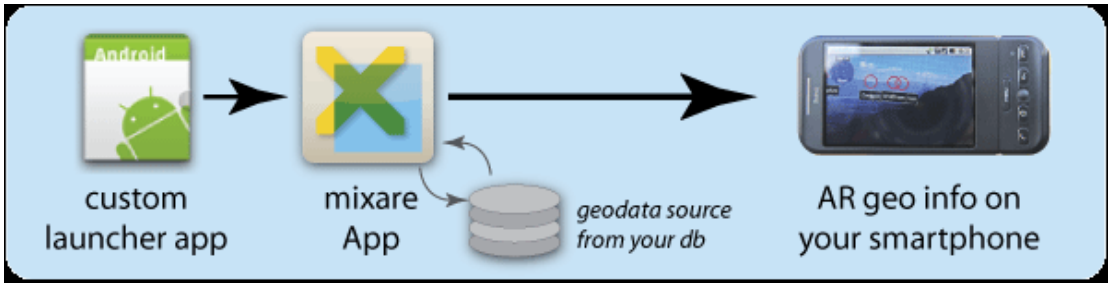
Gambar 2. 15 Mixare dengan Wikipedia sebagai database servernya

2. mixare dapat diakses melalui link pada sebuah situs HTML, dimana sumber data yang ditransfer ke aplikasi.



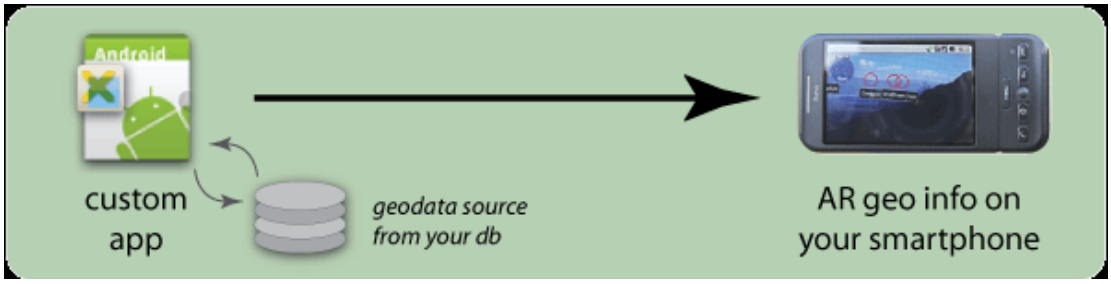
Gambar 2. 16 Mixare diakses melalui sebuah link pada situs HTML

3. Mixare dapat diintegrasikan dengan aplikasi yang dibangun sendiri, dengan sumber data yang ditransfer ke aplikasi.



Gambar 2. 17 Mixare dapat diintegrasikan dengan aplikasi yang telah dibangun sendiri.

4. Mixare dapat secara bebas dikembangkan dan bahkan dapat diubah menjadi sebuah aplikasi individu.



Gambar 2. 18 Mixare dapat dikembangkan dan diubah menjadi aplikasi individu.

b. Penentuan Jarak Lokasi Tujuan Terdekat

Dalam menentukan jarak lokasi tujuan terdekat dibutuhkan persamaan yang mampu menkalkulasikan antara 2 lokasi yaitu lokasi pengguna berada dan lokasi yang ada disekitar pengguna. Lokasi terdekat biasanya menjadi tujuan utama pengguna untuk mencapai lokasi tersebut, sehingga untuk mendapatkan jarak terdekat lokasi tersebut dengan pengguna digunakan metode *Great Circle Distance*.

Great Circle Distance adalah metode untuk melakukan menentukan jarak terpendek antara 2 lokasi di muka bumi. Salah satu persamaan untuk menghitung greatest distance antara 2 titik lokasi yaitu dengan menggunakan persamaan haversine.

Persamaan Haversine merupakan persamaan yang dapat mengkalkulasikan jarak antara 2 titik lokasi di muka bumi. Perhitungan dari rumus ini juga cukup akurat dimana rumus ini mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi.

Persamaan Haversine

$$d = \text{acos}(\sin(lat_1).\sin(lat_2)+\cos(lat_1).\cos(lat_2).\cos(long_2-long_1)).R$$

Dimana :

Lat 1 = Nilai latitude lokasi pengguna

Lat 2 = Nilai latitude lokasi poi

Long 1 = Nilai longitude lokasi pengguna

Long 2 = Nilai longitude lokasi poi

R = Radius bumi (rata-rata radius = 6,371kilometer)

Dengan menggunakan persamaan ini maka jarak antara lokasi pengguna dan setiap lokasi yang ingin dituju akan tampak pada visualisasi marker augmented reality. Sehingga lokasi POI yang terdekat dapat diketahui oleh pengguna.

2.8.5 Manfaat Augmented Reality

Dengan teknologi Augmented Reality, banyak manfaat yang diperoleh dari berbagai macam bidang, seperti kedokteran, manufaktur, hiburan, pendidikan, militer dan navigasi.

2.8.5.1 Kedokteran

Pada dunia kedokteran, peran *Augmented Reality* menjadi semakin dibutuhkan, dikarenakan pencitraan pada simulasi operasi, simulasi pembuatan vaksin virus, simulasi kerangka tubuh dan lain-lain yang berhubungan dengan dunia kedokteran.

2.8.5.2 Manufaktur dan Reparasi

Augmented Reality pada bidang ini diaplikasikan sebagai pemasangan, pemeliharaan dan reparasi mesin-mesin berstruktur kompleks, seperti mesin pada mobil. Informasi-informasi yang dibutuhkan dapat dengan mudah dimengerti yaitu dengan cara dengan cara menampilkan gambar-gambar 3D diatas peralatan-peralatan yang nyata. Gambar-gambar ini menampilkan langkah-langkah yang harus dilakukan agar dapat menyelesaikan bagian-bagian belum dimengerti untuk diselesaikan. Gambar 3D pun dapat dianimasikan, sehingga instruksi pemasangan, pemeliharaan dan reparasi menjadi semakin jelas.

2.8.5.3 Hiburan

Dunia hiburan adalah salah satu bidang yang sudah cukup lama menerapkan teknologi Augmented Reality. Salah satu contohnya pada siaran langsung pertandingan sepak bola, dimana pada bagian atas kiri biasanya ditempatkan informasi skor dan waktu pertandingan.

2.8.5.4 Militer

Seperti dunia hiburan, militer juga sudah betahun-tahun menerapkan teknologi ini pada pelatihan simulasi pesawat dan perang mereka. Contohnya, seorang tentara yang menggunakan perlengkapan tersebut dapat melihat helikopter yang datang. Dalam peperangan, tampilan medan perang yang nyata dapat digabungkan dengan informasi catatan dan sorotan untuk memperlihatkan unit musuh yang tidak terlihat tanpa perlengkapan ini.

2.8.5.5 Navigasi Telepon Genggam

Dampak teknologi ini benar-benar terasa pada 1 tahun belakangan ini, dikarenakan integrasinya pada sistem GPS dan *handphone*. Dengan didukung GPS lokasi-lokasi yang sulit ditemukan, dapat dengan mudah diarahkan.

Implementasi teknologi ini sangat membantu bagi para turis yang baru mengenal wilayah yang akan ditujunya. GPS akan menunjukkan arah lokasi saat ini berada dan mengarahkan pada lokasi tempat tujuan yang akan dituju. Sedangkan teknologi Augmented Reality pada navigasi berfungsi sebagai *marker* informasi tempat yang diwujudkan dalam layar *handphone* melalui sebuah kamera. Arah yang dituju dapat dengan mudah ditempuh sesuai dengan data *latitude* dan *longitude* yang diberikan GPS melalui satelit.

2.9 Android

Pembahasan mengenai android dan *tools* yang digunakan dalam pembuatan aplikasi, akan dijelaskan pada sub bab berikut.

2.9.1 Pengertian Android

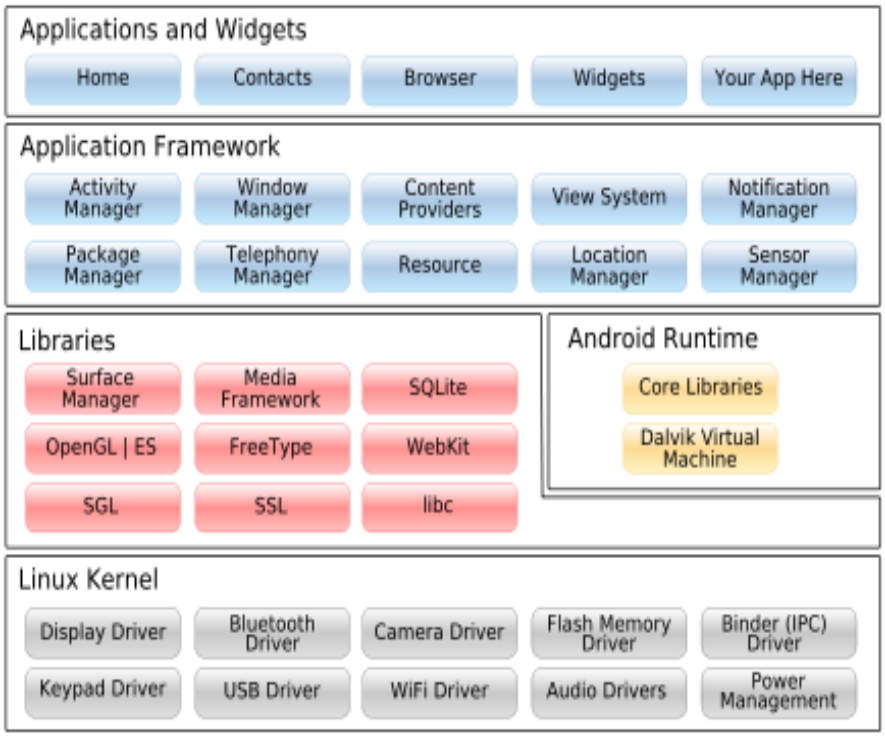
Android merupakan sistem operasi yang ditujukan pada perangkat bergerak (*mobile*) baik itu berupa *handphone* maupun *netbook*. Android dibangun diatas Linux Kernel yang memberikan keterbukaan dari sisi pengembang, sehingga *developer* pengembang Android tidak hanya untuk kalangan tertentu saja.

Android dikembangkan oleh Google bersama *Open Handset Allience* (OHA). Open handset Allience merupakan aliansi perangkat selular terbuka yang terdiri dari

47 perusahaan *Hardware, Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

2.9.2 Arsitektur Android

Arsitektur pada sistem Android memiliki 5 tingkat lapisan *layer*. Setiap lapisan *layer* tersebut menghimpun beberapa program yang yang mendukung fungsi-fungsi spesifik dari sistem operasi. Pemisahan *layer* tersebut bertujuan untuk memberikan abstraksi sehingga memudahkan pengembang aplikasi.



Gambar 2. 19 Arsitektur android

Layer Linux Kernel merupakan lapisan *layer* paling bawah yang berperan sebagai abstraction layer antara hardware dan keseluruhan software.. Google menggunakan Linux Kernel versi 2.6 untuk membangun sistem operasi Android,

yang mencakup *memory management*, *security setting*, *power management* dan beberapa *driver hardware*.

Layer Libraries berisi berisi library-library dalam bahasa C yang digunakan oleh berbagai komponen yang ada pada sistem operasi Android. Beberapa library tersebut antara lain :

1. System C library, diturunkan dari implementasi standard C *system library* (libc) milik BSD, dioptimasi untuk piranti *embedded* berbasis Linux.
2. *Media library* digunakan untuk berbagai format gambar, audio, dan video.
3. *Surface Manager* digunakan untuk mengatur tampilan.
4. *LibWebCore* merupakan *engine* yang digunakan untuk *web browser*.
5. *SGL(Scalable Graphics Library)* merupakan *engine* yang digunakan untuk grafik 2D.
6. *Libraries 3D* merupakan implementasi berdasarkan OpenGL ES 1.0 APIs; *library* ini menggunakan hardware 3D *acceleration* dan *highly optimized* 3D software rasterizer.
7. *FreeType* digunakan untuk rendering huruf *bitmap* dan *vector*.
8. *Library SQL Lite* merupakan *engine* yang digunakan untuk database relasional yang *lightweight*.

Layer Android Runtime berisi *Core Libraries* yang mencakup serangkaian inti *library* Java, artinya Android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemrograman Java. Di *layer* ini juga terdapat *Dalvik Virtual Machine*, dimana

setiap aplikasi Android akan berjalan pada prosesnya sendiri dengan *instance* dari *Dalvik Virtual Machine*. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi *Virtual Machine* dengan *efisien***.** Dalvik VM dapat mengeksekusi file dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual Machine ini *register-based*, dan menjalankan class-class yang *dicompile* menggunakan *compiler* Java yang kemudian ditransformasi menjadi format .dex menggunakan "dx" tool yang telah disertakan. Dalvik Virtual Machine (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti *threading* dan *low-level memory management*.

Layer Application Framework merupakan layer yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. *Application Framework* merupakan serangkaian tool dasar seperti alokasi *resource* smartphone, aplikasi telepon, pergantian antar - proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Layer Application adalah layer dimana aplikasi Android berjalan seperti *email client*, SMS *program*, kalendar, *browser*, peta, dan lain-lain. *Layer* ini juga merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi. Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada Android semua aplikasi, baik

aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.10 Unified Modelling Language (UML)

UML dalam sebuah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan *artifact* dari sistem *software*, untuk memodelkan bisnis, dan sistem non-*software* lainnya. UML merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan artifact yang terdapat dalam sistem. *Artifact* adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifact* dapat berupa model, deskripsi atau *software*.

2.10.1 Use Case Diagram

Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (Aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use Case Diagram* dapat digunakan selama proses analisis untuk menagkap requitment sistem dan untuk memahami bagaimana sistem bekerja.

2.10.2 Class Diagram

Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram*

memperlihatakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem.

Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, *class diagram* berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

2.10.3 Behavior Diagram

Behavior diagram dapat dikelompokan menjadi tiga diagram, yaitu :

a. Statechart Diagram

Statechart Diagram berfungsi untuk memodelkan prilaku dinamis satu kelas satu objek.

b. Activity Diagram

Activity Diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktifitas dalam suatu proses.

c. Interaction Diagram

Interaction Diagram dibagi menjadi dua model diagram yaitu :

2.10.4 Sequence Diagram

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan use case. *Sequence diagram*, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*.

2.10.5 Colaboration Diagram

Melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam *collaboration diagram* terdapat beberapa objek, *link,* dan *message*.

2.10.6 Implementation Diagram

Implementation diagram dibagi menjadi dua diagram, yaitu :

1. *Component Diagram* menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code*, *binary code*, dan komponen tereksekusi (*execute components*)
2. *Deployment Diagram* memperlihatkan pemetaan *software* kepada *hardware*. Dimana akan berjalan (di *server/multitier*, *standalone* atau lainnya), dan menggambarkan model koneksi dan kemampuan jaringan dan hal lainya yang bersifat fisik.

2.11 MySQL

MySQL adalah sebuah sistem manajemen database relasi *(relational database management system)* yang bersifat "terbuka" *(open source)*. Terbuka maksudnya adalah MySQL boleh di-*download* oleh siapa saja, baik versi kode program aslinya maupun versi binernya dan dapat digunakan secara (relatif) gratis baik untuk

dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

2.11.1 Sejarah MySQL

Sejarah MySQL yang merupakan hasil buah pikiran dari Michael "Monty" Widenius, David Axmark, dan Alan Larson dimulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal ditulisnya program MySQL adalah untuk mengembangkan aplikasi web yang digunakan salah satu klien MySQL AB.

2.11.2 Keistimewaan MySQL

Sebagai *database server* yang memiliki konsep *database modern*, MySQL memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL.

1. *Interface*

   Interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*)

2. *Portability*

   MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, dan lain-lain.

3. *Open Source*

   MySQL didistribusikan secara *Open Source*, sehingga dapat digunakan secara bebas.

4. *Multi-user*

   MySQL dapat digunakan oleh beberapa *User* dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

5. *Performance Tuning*

   MySQL memiliki kecepatan yang tinggi dalam menangani *query*, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

6. *Column Types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, timestamp, year, set* dan *enum.*

7. *Command* dan *Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query.*

8. *Security*

   MySQL memiliki lapisan lapisan sekuritas seperti *level subnetmask,* nama *host*, dan izin akses *User* dengan *system* perizinan yang mendetail serta *password* terenkripsi.

9. *Scalability* dan Limits

   MySQL mampu menangani *database* dengan skala besar, dengan jumlah records lebih dari 50 juta dan 60 juta table serta 5 miliar baris.

Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

10. *Connectivity*

    MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, Unix Socet (UNIX), atau Named Pipes (NT).

11. *Localisation*

    Deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.

12. *Client* dan *Tools*

    Dilengkapi dengan berbagai *Tool* yang dapat digunakan untuk administrasi *database,* dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur Tabel

Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding database lainnya semacam PostgreSQL ataupun Oracle.

2.12 Personal Home Page Tool (PHP)

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

2.12.1 Sejarah PHP

PHP merupakan teknologi yang diperkenalkan pada tahun 1994 oleh Rasmus Lerdorf berupa versi awal yang tidak dipublikasikan, digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup onlinenya. Versi pertama digunakan oleh pihak lain pada awal tahun 1995 dan dikenal dengan nama *Personal Home Page Tools*. di dalamnya terkandung sebuah parser *engine* (mesin pengurai) yang sangat disederhanakan, yang hanya mampu mengolah macro khusus dan beberapa utilitas yang sering digunakan pada pembuatan homepage, serta buku tamu, pencacah dan hal semacamnya, parser tersebut ditulis ulang pada tahun 1995 dan dinamakan PHP/FI version 2. FI (form Interpreter) sendiri berasal dari kode lain yang juga ditulis oleh rasmus, yang menterjemahkan HTML dari data. ia menggabungkan *Personal*; *Homepage tool* dengan *form* interpreter dan menambahkan dukungan terhadap *server* database yang menggunakan format mysql, sehinnbgga lahirlah PHP/FI.PHP/FI tumbuh dengan pesat hingga sekarang.

2.12.2 Keistimewaan PHP

Sebagai bahasa skrip yang bersifat *server side* PHP memiliki beberapa keunggulan antara lain:

1. Tidak diperlukan kompabilitas *browser* atau harus menggunakan *browser* tertentu karena *serverlah* yang akan mengerjakan skrip PHP tersebut
2. *Freeware* artinya dapat didistribusikan dengan bebas

Memiliki kemampuan koneksi keberbagai macam database seperti: MySQL, PostgreSQL, Oracle, dBase, Sybase dan banyak lagi.

2.12.3 Wampserver

Dalam dunia teknologi informasi, kata server sudah tidak asing lagi karena untuk menjalankan sebuah sistem informasi atau aplikasi berbasiskan internet harus memiliki paling sedikit satu *server*. Baik itu *server online* atau pun yang *offline*. Agar aplikasi bisa berjalan dengan kinerja yang maksimum.

Salah satu *server offline* yang ada sekarang ini adalah WAMP yang merupakan kependekan dari Windows, Apache, MySQL, dan PHP. WAMP dibuat hanya untuk bekerja pada sistem operasi *windows*.

Apache adalah sebuah *web server* yang membuat akses pengguna yang menggunakan web browser seperti internet explorer atau firefox untuk connet ke komputer dan melihat informasi yang terdapt pada komputer melalui halaman web.

WAMP server digunakan karena *software* lainnya yang membangun aplikasi merupakan software yang mendukung WAMP server.

2.13 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.13.1 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.13.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

1. *Core platform*
2. OSGi
3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh

Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

2.13.3 Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse *Platform* dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem *enterprise*. Adapun versi eclipse yang telah diluncurkan yaitu :

1. Eclipse 3.0
2. Eclipse 3.1
3. Callisto
4. Europa
5. Ganymede
6. Galileo
7. Helios
8. Indigo

2.14 JSON(JavaScript Object Notation)

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 pemprograman 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.

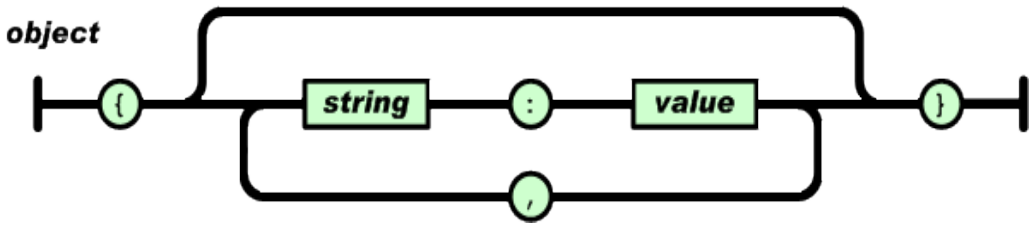
JSON terbuat dari dua struktur:

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*.
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*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen 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.

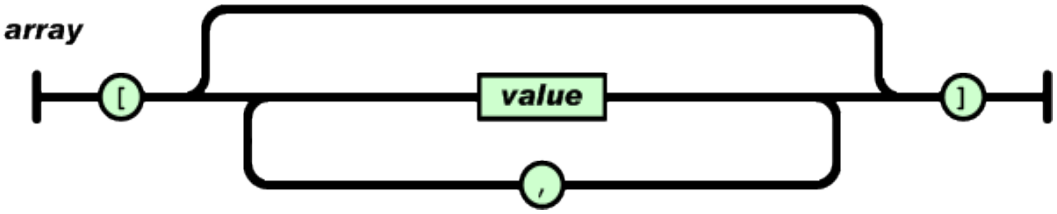
JSON menggunakan bentuk sebagai berikut:

1. Objek 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).



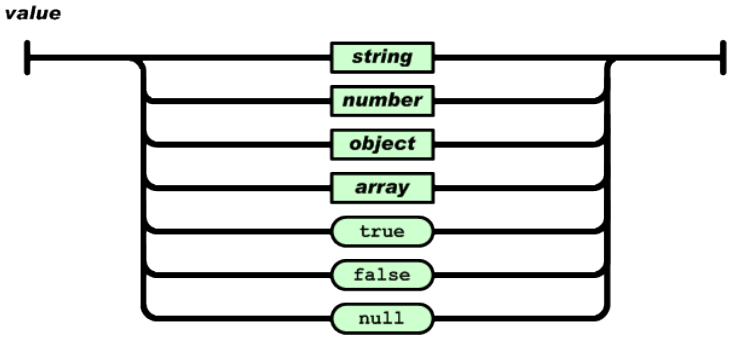
Gambar 2. 20 Objek JSON

2. Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan `[` (kurung kotak buka) dan diakhiri dengan `]` (kurung kotak tutup). Setiap nilai dipisahkan oleh `,` (koma).



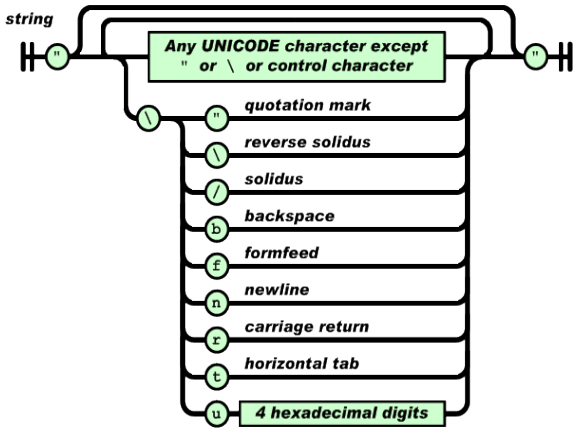
Gambar 2. 21 Larik JSON

3. Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau *angka*, atau `true` atau `false` atau `null`, atau sebuah *objek* atau sebuah *larik*. Struktur-struktur tersebut dapat disusun bertingkat.



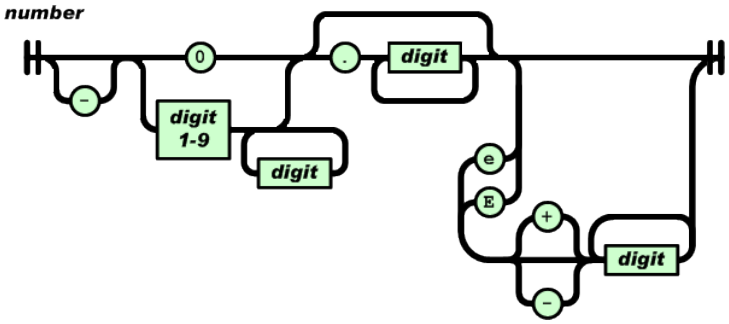
Gambar 2. 22 Nilai JSON

4. String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar 2. 23 String JSON

5. Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2. 24 Angka JSON