BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Profil Perusahaan

Alam TV lahir dari cinta dan keindahan dari Pulau Bali. Bali itu unik, tepat disebut sebagai pulau Dewata. Rasa keindahan yang menanamkan pulau ini menuntut penghormatan terhadap Tanah, Air, Tradisi dan Bali. Sebuah pulau utopia impian banyak orang. Alam TV berusaha untuk menjaga keseimbangan sistem yang mudah untuk di analisis secara cepat. Alam TV akan terus hidup dengan cinta dan kesetiaan untuk mengawal tiga pilar utama dimana kualitas hidup ditentukan, yaitu lingkungan hidup dengan dasar Tri Hita Kirana, pluralisme dalam payung Bhinneka Tunggal Ika yang humanis, dan tradisi budaya dengan ragam kearifan likal yang mencerahkan.

Memasuki tahun kedua, Alam TV Bali memberikan kontribusi untuk mempromosikan kelestarian lingkungan, konservasi kemanusiaan dan budaya Bali yang lebih baik dari televisi lainnya di Indonesia yang pernah terselenggara di daerah-daerah. Dengan penekanan pada jiwa Bali "TAKSU" yang berarti getaran dan pendidikan umum untuk penduduk dan turis. Alam TV Bali memiliki prinsip sosial untuk strategi dan kesuksesan. Target utama berikutnya adalah untuk mendapatkan dukungan finansial yang signifikan untuk mempertahankan momentum yang dicapai sejauh ini.

Visi dan Misi Alam TV Bali :

1. Visi Alam TV Bali
    a. Mempromosikan kelestarian lingkungan hidup di Bali
    b. Mempromosikan kemanusiaan di Bali
    c. Mempromosikan budaya di Bali
2. Misi Alam TV Bali

    Alam TV akan berusaha untuk menjadi perusahaan media massa yang memfasilitasi aktivasi dan pengiriman suara yang memahami kebutuhan untuk kelestarian lingkungan dan budaya di Bali.

2.1.2 Logo Perusahaan



Gambar 2.1 Logo Perusahaan

2.1.3 Struktur Organisasi Perusahaan



Gambar 2.2 Struktur Organisasi

2.2 Landasan Teori

Dalam sebuah laporan penelitian perlu dijelaskan mengenai teori-teori yang digunakan dalam penulisan.

2.2.1 Jurnalistik

Definisi jurnalistik sangat banyak. Namun pada hakekatnya sama, para tokoh komuniikasi atau tokoh jurnalistik mendefinisikan berbeda-beda. Jurnalistik secara harfiah, jurnalistik (*journalistic*) artinya kewartawanan atau hal-ihwal pemberitaan. Kata dasarnya "jurnal" (*journal*), artinya laporan atau catatan, atau "jour" dalam bahasa Prancis yang berarti "hari" (*day*) atau "catatan harian" (*diary*). Dalam bahasa Belanda *journalistiek* artinya penyiaran catatan harian.

Menurut A.Muis (pakar hukum komunikasi) mengatakan bahwa definisi tentang jurnalistik cukup banyak. Namun dari definisi-definisi tersebut memiliki kesamaan secara umum. Semua definisi juranlistik memasukan unsur media massa, penulisan berita, dan waktu yang tertentu (aktualitas). Menurut Edwin Emery juga sama mengatakan dalam jurnalistik selalu harus ada unsur kesegaran waktu (timeliness atau aktualitas). Dan Emery menambahkan bahwa seorang jurnalis memiliki dua fungsi utama. Pertama, fungsi jurnalis adalah melaporkan berita. Kedua, membuat interpretasi dan memberikan pendapat yang didasarkan pada beritanya.

2.2.2 *Citizen Journalism*

Citizen Journalism (Jurnalisme Warga) atau jurnalisme partisipatip atau jurnalisme jalanan merupakan bentuk jurnalisme di mana warga masyarakat turut berperan serta dalam proses pengumpulan, pelaporan, menganalisis dan menyebarkan berita dan informasi dalam dan melalui berbagai media massa. Berbagai media tersebut, seperti yang dikategorikan J.D. Lasica, dalam Online Journalism Review (2003) adalah dalam dan melalui:

1. Audience participation (seperti komenter user yang diattach pada kisah-kisah berita, blog-blog pribadi, foto, atau video footage yang diambil dari handycam pribadi, atau berita lokal yang ditulis oleh anggota komunitas).
2. Situs web berita atau informasi independen (Consumer Reports, Drudge Report).
3. Situs berita partisipatoris murni (OhmyNews).

4. Situs media kolaboratif (Slashdot, Kuro5hin).
5. Bentuk lain dari media ‘tipis’ (mailing list, newsletter e-mail).
6. Situs penyiaran pribadi (situs penyiaran video, seperti KenRadio).

Perkembangan Citizen Journalism bermula pada era 1980-an di Amerika Serikat, ketika munculnya kecenderungan bahwa jurnalisme dan kegiatan peliputan berita dan informasi bukan lagi hanya menjadi tugas para pewarta berita professional, tetapi juga merupakan tindakan dan kegiatan warga masyarakat secara umum dan luas. Seperti gayung bersambut, citizen journalism berkembang ke berbagai belahan dunia termasuk Indonesia, seiring perkembangan media teknologi komunikasi dan informasi.

2.2.3 Definisi Sistem

Sistem merupakan istilah yang umum digunakan dalam berbagai disiplin ilmu, untuk menerangkan suatu metode atau tata cara yang digunakan secara luas dan pesat dalam berbagai aspek kehidupan baik dalam tatanan hidup sederhana seperti kehidupan bermasyarakat sehari - hari maupun di dalam suatu dunia manajemen kerja modern yang mempunyai cakupan lebih luas. Berikut ini akan dijelaskan beberapa pengertian dari sistem, yaitu sebagai berikut:

Kusrini (2007) mengungkapkan :

”Sistem adalah sebuah tatanan yang terdiri dari atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama - sama bertujuan untuk memenuhi suatu proses tertentu”.

Menurut Teguh Wahyono (2004) menjelaskan :

”Sistem adalah kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu”.

Dari uraian di atas dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama - sama untuk mencapai tujuan tertentu.

2.2.3.1 Karakteristik Sistem

Menurut Tata Sutabri (2005)

Sistem memiliki karakteristik atau sifat-sifat tertentu. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

   Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

   Media yang menghubungkan sistem dengan subsistem lain. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

   Energi yang dimasukkan ke dalam sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer. ”Program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan ”data” adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

   Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (O*bjective*)

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran. Maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.2.3.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut:

1. Sistem abstrak dan sistem fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem

komputer, sistem produksi, sistem penjualan, sistem administrasi personalia dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system.* Sistem informasi berbasis komputer merupakan contoh *human-machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem deterministik dan sistem probabilistik

   Sistem deterministik beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem terbuka dan sistem tertutup

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.2.4 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.

2.2.4.1 Data

Banyak pengertian data yang dirangkum dari berbagai sumber. Teguh Wahyono dalam bukunya Sistem Informasi: Konsep Dasar, Analisis Desain, dan Implementasi menjelaskan beberapa pengertian data sebagai berikut :

1. Menurut berbagai kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata *"datum"* yang berarti fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: *"Business data is an organization's description of thing (resources) and events (transactions) that it face"*. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resources*) dan kejadian (*transactions*) yang terjadi.
3. Pengertian yang lain mengatakan bahwa *"data is the description of things and events that we face"*. Jadi data merupakan deskripsi dari sesuatu dan kejadian yang dihadapi.
4. Gordon B. Davis dalam bukunya *Management Informations System: Conceptual Foundations, Structures, And Development* menyebutkan data sebagai bahan mentah dari informasi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dari keempat pengertian di atas, dapat diambil kesimpulan bahwa data adalah suatu bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter-karakter dapat berupa alfabet, angka, maupun symbol khusus seperti *, #, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.

2.2.4.2 Informasi

Masih dari Teguh Wahyono, dalam bukunya menjelaskan beberapa pengertian informasi dari berbagai sumber, sebagai berikut :

1. Menurut Gordon B. Davis dalam bukunya *Management Informations System: Conceptual Foundations, Structures, and Deveopment* menyebutkan informasi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa yang akan datang.
2. Menurut Barry E. Cushing dalam buku *Accounting Information System And Business Organization*, dikatakan bahwa informasi merupakan sesuatu yang menunjukan hasil pengolahan data yang diorganisasikan dan berguna kepada orang yang menerimanya.
3. Menurut Robert N. Anthony dan John Dearden dalam buku *Management Control System*, menyebutkan informasi sebagai suatu kenyataan, data, item yang menambah pengetahuan bagi penggunanya.

   Dari beberapa pengertian di atas dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat Bantu untuk pengambilan suatu keputusan.

2.2.5 Basis Data

2.2.5.1 Definisi Basis Data

Basis data terdiri dari dua kata, yaitu Basis dan Data. **Basis** kurang-lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan **Data** adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya. Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikina rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2. Kumpulan data yang salin berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.2.5.2 Tujuan Basis Data

Tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh/menemukan kembali data (yang kita cari) dengan mudah dan cepat. Secara lebih lengkap pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan seperti berikut ini :

1. Kecepatan dan Kemudahan (*Speed)*

   Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika kita menyimpan data secara manual (non elektronis) atau secara elektronis (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk *spread sheet* atau dokumen teks biasa).
2. Efisiensi Ruang Penyimpanan (*Space)*

   Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.
3. Keakuratan (*Accuracy)*

   Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (*Availability*)

   Pertumbuhan data (baik dari sisi jmlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi kita gunakan, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi *off-line*) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan *offline* (seperti *removable disk* atau *tape*).

5. Kelengkapan (*Completeness*)

   Dalam sebuah basis data, di samping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek, seperti struktur file/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan (*Security*)

   Memang ada sejumlah sistem (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu, kita dapat menentukan pemakai yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

7. Kebersamaan Pemakaian (*Sharability*)

   Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada

saat yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

2.2.5.3 Sistem Basis Data

Basis data hanyalah sebuah objek yang pasif/mati. Ia ada karena ada pembuatnya. Ia tak akan pernah berguna jika tidak ada pengelola. Yang menjadi pengelola secara langsung adalah program/aplikasi (software). Gabungan keduanya (basis data dan pengelolanya) menghasilkan sebuah sistem. Karena itu, secara umum sebuah sistem basis data merupakan sebuah sistem yang terdiri atas kumpulan file (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan seku,pulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi file-file (tabel-tabel) tersebut.

Lebih jauh lagi, dalam sebuah sistem basis data, secara lengkap akan terdapat komponen-komponen utama sebagai berikut :

1. Perangkat Keras (*Hardware*).
2. Sistem Operasi (*Operating System*).
3. Basis Data (*Database*).
4. Sistem Pengelola Basis Data (*Database Management System/DBMS*).
5. Pemakai (*User*).
6. Aplikasi (Perangkat Lunak) lain (bersifat opsional).

2.2.5.4 Abstraksi Data

Salah satu tujuan dari DBMS adalah untuk menyediakan fasilitas/antarmuka (*interface*) dalam melihat/menikmati data (yang lebih ramah/*user oriented*) kepada pemakai/*user*. Untuk itu, sistem tersebut akan menyembunyikan detail tentang bagaiman data disimpan dan dipelihara. Karena itu, seringkali data yang terlihat oleh pemakai sebelumnya berbeda dengan yang tersimpan secara fisik. Abstraksi data merupakan tingkatan/leveldalam bagaimana melihat data dalamsebuah sistem basis data.

Ada tiga level abstraksi data:

1. Level Fisik (*Physical Level*)

   Merupakan level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada level ini, pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri. Pemakai juga berkompeten dalam mengetahui bagaimana representasi fisik dari penyimpanan/pengorganisasian data. Pada level ini kita berurusan dengan data sebagai teks, sebagai angka, atau bahkan melihatnya sebagai himpunan bit data.

2. Level Lojik/Konseptual (*Conseptual Level*)

   Merupakan level berikutnya dalam abstraksi data yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data dan hubungannya dengan data yang lain. Pemakai pada level ini yang misalnya mengetahui bahwa data pegawai disimpan/direpresentasikan dalam beberapa file/tabel, seperti file pribadi, file pendidikan, file pekerjaan, file keluarga, dan sebagainya.

3. Level Penampakan (*View Level*)

   Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data. Banyak *user* dalam sistem basis data tidak akan terlibat (*concern*) dengan semua data/informasi yang ada/disimpan. Para *user* umumnya hanya membutuhkan sebagian data/informasi dalam basis data yang kemunculannya di mata pemakai diatur oleh aplikasi *end-user*. Aplikasi ini juga yang mengkonversi data asli/fisik menjadi data bermakna/lojik pada pemakai. Data yang'dinikmati' pemakai juga bahkan sama sekali berbeda dengan representasi fisiknya, misalnya untuk data yang dapat divisualkan sebagai gambar, data yang dapat diperdengarkan sebagai suara, dan sebagainya. Data yang diperlihatkan juga bisa saja tidak berasal dari hanya sebuah tapi tapi mewakili relasi antar tabel, tapi bagi pemakai yang menggunakannya terasa sebagai satu kesatuan data yang kompak.

2.2.5.5 Bahasa Basis Data

DBMS merupakan perantara bagi pemakai dengan basis data dalam *disk*. Cara berinteraksi/berkomunikasi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat kita sebut sebagai Bahasa Basis Data yang terdiri atas sejumlah perintah yang diformulasikan dan dapat diberikan *user* dan diproses oleh DBMS untuk melakukan suatu pekerjaan tertentu. Sebuah Bahasa Basis Data biasanya dapat dipilah ke dalam dua bentuk yaitu:

1. Data Definition Language (DDL)

   Struktur/skema basis data yang menggambarkan desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language* (DDL). Dengan bahasa inilah kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya. Hasil dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut Kamus Data (*Data Dictionary*).

   Kamus Data merupakan suat metadata (superdata) yaitu data yang mendeskripsikan data sesungguhnya. Kamus Data ini akan selalu diakses dalam suatu operai basis data sebelum suatu file data yang sesungguhnya diakses.

2. Data Manipulation Language (DML)

   Merupakan bentuk Bahasa Basis Data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa:

   a. Penyisipan/penambahan data baru ke suatu basis data.
   b. Penghapusan data dari suatu basis data.
   c. Pengubahan data di suatu basis data.

   Data Manipulation Language (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada dua jenis DML yaitu:

1. Prosedural, yang mensyaratkan agar pemakai menentukan data apa yang diinginkan sera bagaimana cara mendapatkannya.
2. Nonprosedural, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

2.2.5.6 Model Data Relasional

Model Data Relational sering pula disebut sebagai Model Relasional atau Basis Data Relasional. Model data ini ditemukan/diperkenalkan pertama kali oleh E.F Codd. Model Basis Data menunjukkan suatu cara/mekanisme yang digunakan untuk mengelola/mengorganisasi data secara fisik dalam memori sekunder yang akan berdampak pula pada bagaimana kita mengelompokkan dan membentuk keseluruhan data yang terkait dalam sistem yang sedang kita tinjau.

a. Istilah-istilah dalam model data relasional
    1. Relasi

       Sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris.
        1. Atribut

           Kolom pada sebuah relasi (field).
        2. Tupel

           Baris pada sebuah relasi (record).
        3. Domain

           Kumpulan nilai yang valid untuk satu atau lebih atribut
        4. Derajat (degree)

           Jumlah atribut dalam sebuah relasi (jumlah field)
        5. Cardinality

           Jumlah tupel dalam sebuah relasi (jumlah record)

b. Relational Key
    1. Super key

       Satu atribut/kumpulan atribut yang secara unik mengidentifikasi sebuah tupel di dalam relasi (satu atau lebih field yang dapat dipilih untuk membedakan antara 1 record dengan record lainnya).

2. Candidate key

   Atribut di dalam relasi yang biasanya mempunyai nilai unik (super key dengan jumlah field yang paling sedikit).

3. Primary key

   Candidate key yang dipilih untuk mengidentifikasikan tupel secara unik dalam relasi.

4. Alternate key

   Candidate key yang tidak dipilih sebagai primary key.

5. Foreign key

   Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain atribut tersebut hanya sebagai atribut biasa.

6. Relational Integrity Rules

   1. Null

      Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk baris (tuple) tersebut.

   2. Entity Integrity

      Tidak ada satu komponen primary key yang bernilai null.

   3. Referential Integrity

      Suatu domain dapat dipakai sebagai kunci primer bila merupakan atribut tunggal pada domain yang bersangkutan.

2.2.5.7 Normalisasi

Normalisasi adalah suatu proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam dua buah tabel atau lebih, yang tidak lagi memiliki masalah tersebut. Masalah tersebut biasanya merupakan suatu ketidakkonsistenan (tidak normal) apabila dilakukan penghapusan *(delete)*, pengubahan *(update)* dan pembacaan *(retrieve)* pada suatu basis data.

Bentuk normalisasi adalah suatu aturan yang dikenakan pada tabel-tabel dalam basis data dan harus dipenuhi oleh tabel-tabel tersebut pada level-level normalisasi. Ada macam-macam bentuk normalisasi, diantaranya adalah bentuk

tidak normal, bentuk normal pertama, bentuk normal kedua dan bentuk normal ketiga.

Aturan-aturan dalam masing-masing bentuk normalisasi tersebut adalah sebagai berikut:

a. *Bentuk tidak normal*

   Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

b. *Bentuk normal pertama*

   Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom (tidak perlu ada indeks dalam memberi nama kolom).

c. *Bentuk normal kedua*

   Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

d. *Bentuk normal ketiga*

   Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

2.2.6 Kompresi

Dalam ilmu komputer dan teori informasi , kompresi data atau sumber pengkodean adalah proses encoding informasi dengan menggunakan lebih sedikit bit (atau unit informasi-bantalan lainnya) dari sebuah unencoded representasi akan menggunakan, melalui penggunaan khusus pengkodean skema. Dalam komputasi, deduplication data adalah teknik kompresi data khusus untuk menghilangkan data-grained berlebihan kasar, biasanya untuk meningkatkan utilisasi storage.

Seperti komunikasi apapun, dikompresi komunikasi data hanya bekerja jika kedua pengirim dan penerima informasi memahami skema pengkodean. Misalnya, teks ini masuk akal hanya jika penerima mengerti bahwa itu adalah dimaksudkan untuk ditafsirkan sebagai karakter yang mewakili bahasa Inggris demikian pula, data terkompresi hanya dapat dipahami jika metode decoding diketahui oleh penerima.

Kompresi berguna karena membantu mengurangi konsumsi sumber daya mahal, seperti hard disk space atau transmisi bandwidth . Pada sisi negatifnya, data dikompresi harus didekompresi untuk digunakan, dan ini pengolahan tambahan mungkin merugikan beberapa aplikasi. Sebagai contoh, skema kompresi untuk video mungkin memerlukan perangkat keras mahal untuk video yang akan didekompresi cukup cepat untuk dilihat karena sedang decompressed (pilihan untuk dekompresi video secara penuh sebelum menonton mungkin nyaman, dan membutuhkan ruang penyimpanan untuk decompressed video). Rancangan skema kompresi data sehingga melibatkan trade-off antara berbagai faktor, termasuk tingkat kompresi, jumlah distorsi memperkenalkan (jika menggunakan skema kompresi lossy), dan sumber daya komputasi yang dibutuhkan untuk kompres dan uncompress data.

a. Lossy

Lossy kompresi citra digunakan dalam kamera digital , untuk meningkatkan kapasitas penyimpanan dengan minimal penurunan kualitas gambar. Demikian pula, DVD menggunakan lossy MPEG-2 Video codec untuk kompresi video. Dalam lossy kompresi audio , metode psychoacoustics digunakan untuk menghapus non-terdengar (atau kurang terdengar) komponen dari sinyal. Kompresi berbicara manusia sering dilakukan dengan teknik khusus bahkan lebih, sehingga " pidato kompresi "atau" suara coding "kadang-kadang dibedakan sebagai suatu disiplin yang terpisah dari" kompresi audio ". audio yang berbeda dan kompresi standar pidato terdaftar di bawah codec audio . Suara kompresi akan digunakan dalam telepon Internet misalnya, sementara kompresi audio yang

digunakan untuk CD ripping dan diterjemahkan oleh pemain audio. Berikut ciri-ciri lossy:

1) Terdapat informasi yang hilang pada saat sampai pada telinga dan mata manusia.
2) Digunakan pada kompresi objek audio, image, video dimana keakuratan data absolut tidak diperlukan.
3) Contoh: bila video image dikompres dengan basis frame-by- frame hilangnya data pada satu frame tidak mempengaruhi penglihatan.
4) Aplikasi: medical screening systems, video conferencing, dan multimedia messaging systems.
5) Metode kompresi yang banyak digunakan adalah standar JPEG.

b. Lossless

Berikut ini ciri-ciri lossless:

1) Data tidak berubah atau hilang pada proses kompresi atau dekompresi
2) Membuat satu replika dari objek asli
3) Menghilangkan perulangan karakter
4) Digunakan pada data teks dan image
5) Pada saat dilakukan dekompres, perulangan karakter diinstal kembali

c. **Kompresi *lossless versus lossy***

Losseless algoritma kompresi memanfaatkan redundansi biasanya statistik sedemikian rupa untuk mewakili pengirim data lebih singkat tanpa kesalahan. kompresi Lossless dimungkinkan karena sebagian besar dunia nyata telah redundansi data statistik. Sebagai contoh, dalam teks bahasa Inggris, 'e' huruf jauh lebih umum daripada huruf 'z', dan probabilitas bahwa 'q' huruf akan diikuti oleh huruf 'z' sangat kecil. Kompresi jenis lain, disebut *kompresi lossy data* atau persepsi coding , adalah mungkin jika beberapa kehilangan kesetiaan diterima. Umumnya, sebuah kompresi data lossy akan dipandu oleh penelitian tentang bagaimana orang melihat data tersebut. Sebagai contoh, mata manusia lebih sensitif terhadap variasi

halus dalam terang daripada variasi warna. JPEG kompresi gambar yang bekerja di sebagian oleh "pembulatan" beberapa informasi penting ini-kurang. Lossy kompresi data menyediakan cara untuk mendapatkan kesetiaan terbaik untuk jumlah yang diberikan kompresi. Dalam beberapa kasus, *transparan* (unnoticeable) kompresi yang diinginkan, dalam kasus lain, kesetiaan adalah dikorbankan untuk mengurangi jumlah data sebanyak mungkin.

Skema kompresi Lossless adalah reversibel sehingga data asli dapat direkonstruksi, sementara skema lossy menerima beberapa hilangnya data untuk mencapai kompresi yang lebih tinggi. Namun, algoritma kompresi lossless data akan selalu gagal untuk kompres beberapa file, memang, setiap algoritma kompresi tentu akan gagal untuk kompres data tidak berisi pola-pola yang jelas. Upaya untuk kompres data yang telah dikompres biasanya sudah demikian akan menghasilkan sebuah ekspansi, seperti yang akan mencoba untuk menekan semua tapi yang paling sepele dienkripsi data.

Dalam prakteknya, data lossy kompresi juga akan datang ke titik di mana memadatkan lagi tidak bekerja, walaupun suatu algoritma yang sangat lossy, seperti misalnya selalu mengeluarkan byte terakhir dari sebuah file, akan selalu kompres file sampai ke titik di mana ia kosong. Contoh kompresi lossy vs lossless adalah string berikut:

25.888888888

String ini dapat dikompresi sebagai:

25.[9]8

Diartikan sebagai, "25 poin 9 delapan", string aslinya diciptakan sempurna, hanya ditulis dalam bentuk yang lebih kecil. In a lossy system, using Dalam sistem lossy, menggunakan 26 Sebaliknya, data asli pasti hilang, di manfaat dari file yang lebih kecil.

2.2.7 Metode *Object Oriented*

Pada metode pengembangan sistem ini akan menggunakan *object oriented* (OO) yang merupakan paradigma dalam rekayasa software yang didasarkan pada objek dan kelas. Object-oriented merupakan metodologi terbaik dalam rekayasa software dan mencakup bidang aplikasi yang sangat luas. Karena luasnya cakupan object-oriented makan terdapat beberapa hal yang membingungkan berkenaan dengan istilah dan konsep object-oriented. Istilah dan konsep yang berkenaan dengan objectoriented ini adalah :

a. *Object oriented Analysis* adalah metode analisis yang memeriksa requirement (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelaskelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan.
b. *Object oriented Design* adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Object oriented analysis dan *design* merupakan pendekatan yang menekankan pada solusi logic berbasis objek. Beberapa konsep dasar dalam OOAD :

1. Objek

   Objek adalah "benda", secara fisik atau konseptual dapat kita temui di sekeliling kita. Hardware, Software, dokumen, manusia dan bahkan konsep semuanya adalah konsep objek. Sebuah objek memiliki keadaaan sesaat *(state)* dan perilaku *(behaviour)*. State dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut.
2. Kelas

   Kelas adalah definisi umum untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku dan atribut objek-objek tersebut.

Object oriented merupakan metode yang paling baik dalam rekayasa perangkat lunak, diantaranya *procedure oriented, object-oriented, data strukturoriented,* data *flow-oriented,* dan *constraint-oriented*. Sehingga dengan metode objectoriented ini dapat diaplikasikan dalam seluruh ruang lingkup rekayasa software.

Untuk memhami keunggulan OOAD maka kita harus memahami masalah yang dihadapi antara lain :

1. Software sulit untuk dimodifikasi bila memerlukan pengembangan.
2. Proses pembuatan software memerlukan waktu yang cukup lama sehingga kadang kala melebihi anggaran dalam pembuatannya.
3. Para programmer selalu membuat software dari dasar karena tidak adanya kode yang bias digunakan ulang (*reuse*).

Dari beberapa keunggulan diatas, dengan menggunakan object-oriented maka sangat menguntungkan bagi programmer karena programmer dapat mendesign program dalam bentuk objek-objek dan hubungan antara objek-objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Keuntungan yang lain adalah proses pembuatan software dapat dilakukan dengan lebih cepat karena software dibangun dalam objek-objek standard, sehingga dapat digunakan secara berulang-ulang.

Bahasa pemograman yang menggunakan *object oriented* diantaranya yang terkenal C++, Visual Basic dan Java.

2.2.8 *Unified Modelling Language* (*UML*)

Menurut Nugroho (2009), *Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk

menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.2.8.1 Teknik-Teknik UML

Dalam tahap perancangan satu sistem diperlukan adanya teknik-teknik penyusunana sistem untuk menganalisa dan mempelajari objek yang ada unuk mengetahui apakah mereka dapat digunakan kembali atau diadaptasi untuk pemakaina baru atau menetukan satu objek baru atau yang dimodifikasi yang digabung dengan objek yang sudah ada ke dalam suatu aplikasi komputasi. Teknik-teknik tersebut adalah sebagai berikut:

1. **Use Case Diagram**

   Diagram ini menunjukan sekumpulan kasus fungsional dan actor (jenis kelas khusus) dan keterhubungannya diagram ini dapat sangat membantu bila kita sedang menyusun requitment sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada *sistem*.

2. **Sequence Diagram**

   Diagram ini menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

3. **Collaboration Diagram**

   Diagram ini merupakan diagram interaksi. Diagram ini menekankan pada organisasi stuktur dari objek-objek yang mengirim dan menerima pesan.

4. **Class Diagram**

   Diagram ini adalah seebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek yang merupakan inti pengembangan dan desain

yang berorientasi objek. Kelas mengambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

5. **Statechart Diagram**

   Statechart diagram menunjukkan transisi dan perubahan keadaan suatu objek pada sistem sebagai akibat dari stimulasi yang diterima. Dalam UML, state digambarkan berbentuk segi empat dengan sudut tumpul dan memiliki nama sesuai dengan kondisi saat itu.

6. **Activity Diagram**

   Actifity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses parallel yang mungkin erjadi pada beberapa eksekusi.

7. **Component Diagram**

   Component Diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan di antaranya.

8. **Deployment Diagram**

   Deployment diagram menggambarkan detail bagaimana komponen dideploy dalam infrastruktur sistem, dimana komponen akan diletakkan (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal-hal lain yang bersifat fisikal.

2.2.9 Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak computer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya

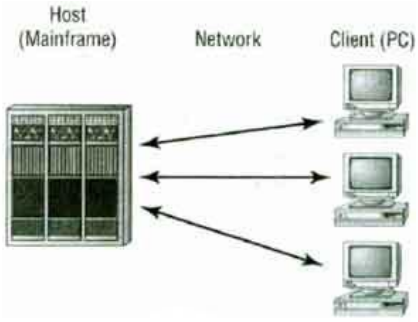
adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini 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.

2.2.9.1 Arsitektur Aplikasi

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu sistem ditempatkan dan bagaimana mereka berkomunikasi. Arsitektur terdistribusi – sebuah istilah yang relatif baru untuk menjelaskan arsitektur aplikasi – berarti bahwa pemrosesan dari suatu aplikasi terjadi pada lebih dari satu mesin. Terdapat beberapa macam arsitektur aplikasi, yaitu :

1. Standalone (*One-Tier*)

   Pada arsitektur ini semua pemrosesan dilakukan pada mainframe. Kode aplikasi, data dan semua komponen sistem ditempatkan dan dijalankan pada host. Seperti terlihat pada gambar 2.3.



Gambar 2.3 Arsitektur Aplikasi *One-Tier*

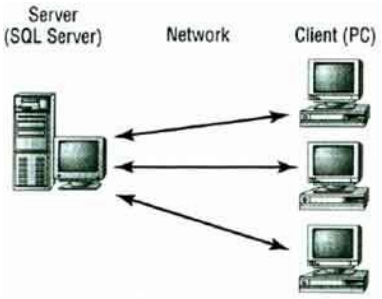
Walaupun komputer client dipakai untuk mengakses mainframe, tidak ada pemrosesan yang terjadi pada mesin ini, dan karena mereka

"dump-client" atau "dump-terminal". Tipe model ini, dimana semua pemrosesan terjadi secara terpusat, dikenal sebagai berbasis-host. Sekilas dapat dilihat kesalahan pada model ini. Ada dua masalah pada komputasi berbasis host: Pertama, semua pemrosesan terjadi pada sebuah mesin tunggal, sehingga semakin banyak user yang mengakses host, semakin kewalahan jadinya. Jika sebuah perusahaan memiliki beberapa kantor pusat, user yang dapat mengakses mainframe adalah yang berlokasi pada tempat itu, membiarkan kantor lain tanpa akses ke aplikasi yang ada.

Pada saat itu jaringan sudah ada namun masih dalam tahap bayi, dan umumnya digunakan untuk menghubungkan terminal dump dan mainframe. Internet baru saja dikembangkan oleh pemerintah US dan pada saat itu dikenal sebagai ARPANET. Namun keterbatasan yang dikenakan pada user mainframe dan jaringan telah mulai dihapus.

2. Client/Server (*Two-Tier)*

Dalam model client/server, pemrosesan pada sebuah aplikasi terjadi pada client dan server. Client/server adalah tipikal sebuah aplikasi *two-tier* dengan banyak client dan sebuah server yang dihubungkan melalui sebuah jaringan, seperti terlihat dalam gambar 2.4. Aplikasi ditempatkan pada komputer client dan mesin database dijalankan pada server jarak-jauh. Aplikasi client mengeluarkan permintaan ke database yang mengirimkan kembali data ke client-nya.



Gambar 2.4 Arsitektur Aplikasi *Two-Tier*

Dalam client/server, client-client yang cerdas bertanggung jawab untuk bagian dari aplikasi yang berinteraksi dengan user, termasuk logika

bisnis dan komunikasi dengan server database. Tipe-tipe tugas yang terjadi pada client adalah :

a. Antarmuka pengguna
b. Interaksi database
c. Pengambilan dan modifikasi data
d. Sejumlah aturan bisnis
e. Penanganan kesalahan

Server database berisi mesin database, termasuk tabel, prosedur tersimpan, dan trigger (yang juga berisi aturan bisnis). Dalam sistem client/server, sebagian besar logika bisnis biasanya diterapkan dalam database. Server database manangani :

a. Manajemen data
b. Keamanan
c. Query, trigger, prosedur tersimpan
d. Penangan kesalahan

Arsitektur client/server merupakan sebuah langkah maju karena mengurangi beban pemrosesan dari komputer sentral ke komputer client. Ini berarti semakin banyak user bertambah pada aplikasi client/server, kinerja server file tidak akan menurun dengan cepat. Dengan client/server user dair berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal. Namun masih terdapat kelemahan pada model ini. Selain menjalankan tugas-tugas tertentu, kinerja dan skalabilitas merupakan tujuan nyata dari sebagian besar aplikasi. Model client/server memiliki sejumlah keterbatasan :

a. Kurangnya skalabilitas
b. Koneksi database dijaga
c. Tidak ada keterbaharuan kode
d. Tidak ada tingkat menengah untuk menangani keamanan dan transaksi

Aplikasi-aplikasi berbasis client/server memiliki kekurangan pada skalabilitas. Skalabilitas adalah seberapa besar aplikasi bisa menangani suatu kebutuhan yang meningkat – misalnya, 50 user tambahan yang

mengakses aplikasi tersebut. Walaupun model client/server lebih terukur daripada model berbasis host, masih banyak pemrosesan yang terjadi pada server. Dalam model client/server semakin banyak client yang menggunakan suatu aplikasi, semakin banyak beban pada server.

Koneksi database harus dijaga untuk masing-masing client. Koneksi menghabiskan sumber daya server yang berharga dan masing-masing client tambahan diterjemahkan ke dalam satu atau beberapa koneksi. Logika kode tidak bisa didaur ulang karena kode aplikasi ada dalam sebuah pelaksanaan executable monolitik pada client. Ini juga menjadikan modifikasi pada kode sumber sulit. Penyusunan ulang perubahan itu ke semua komputer client juga membuat sakit kepala.

Keamanan dan transaksi juga harus dikodekan sebagai pengganti penanganan oleh COM+/MTS. Bukan berarti model client/server bukanlah merupakan model yang layak bagi aplikasi-aplikasi. Banyak aplikasi yang lebih kecil dengan jumlah user terbatas bekerja sempurna dengan model ini. Kemudahan pengembangan aplikasi client/server turut menjadikannya sebuah solusi menarik bagi perusahaan.

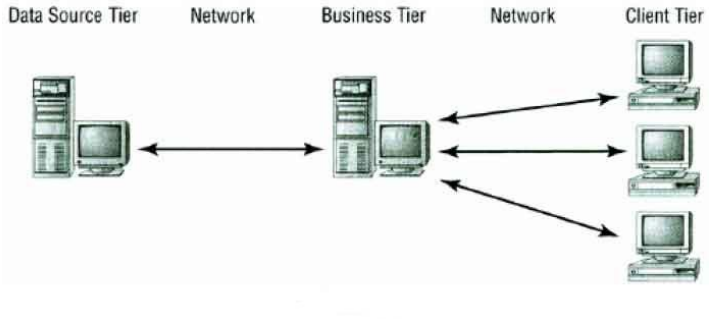
Pengembangan umumnya jauh lebih cepat dengan tipe sistem ini. Siklus pengembangan yang lebih cepat ini tidak hanya menjadikan aplikasi meningkat dan berjalan dengan cepat namun juga lebih hemat biaya.

3. Three-Tier / Multi-Tier

Model *three-tier* atau *multi-tier* dikembangkan untuk menjawab keterbatasan pada arsitektur client/server. Dalam model ini, pemrosesan disebarkan di dalam tiga lapisan (atau lebih jika diterapkan arsitektur *multi-tier*). Lapisan ketiga dalam arsitektur ini masing-masing menjumlahkan fungsionalitas khusus. Yaitu :

a. Layanan presentasi (tingkat client)
b. Layanan bisnis (tingkat menengah)
c. Layanan data (tingkat sumber data)

Layanan presentasi atau logika antarmuka pengguna ditempatkan pada mesin client. Logika bisnis dikeluarkan dari kode client dan ditempatkan dalam tingkat menengah. Lapisan layanan data berisi server database. Setiap tingkatan dalam model three-tier berada pada komputer tersendiri, seperti pada gambar 2.5.



Gambar 2.5 Aplikasi *Three-Tier*

Konsep model three-tier adalah model yang membagi fungsionalitas ke dalam lapisan-lapisan, aplikasi-aplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

2.2.9.2 On-Device Portal (ODP)

On-Device Portal atau ODP adalah suatu teknologi aplikasi Mobile, yang digunakan untuk mendistribusikan konten-konten melalui suatu aplikasi portal khusus yang diinstalasikan dalam perangkat ponsel di sisi klien.

Portal-portal dapat menggunakan teknologi ODP yang menawarkan konten yang tersedia pada portal mampu dapat diunduh dan berada pada perangkat ponsel. ODP bisa digunakan ketika pengguna sedang online maupun offline. ODP juga mampu mengurangi delay atau drop yang seringkali terjadi pada jaringan.

ODP merupakan teknologi yang sedikit mengadaptasi paradigma browser yang memberikan navigasi/penggunaan yang sederhana dan intuitive bagi para penyedia konten dan fungsi-fungsi dasar yang saling berhubungan pada perangkat ponsel. Dengan ODP maka konten dapat tersedia di perangkat ponsel, sehingga mampu meningkatkan penggunaan konten yang tersedia oleh pengguna,

memperdalam pemahaman terhadap konten yang tersedia oleh pengguna dan juga mendorong pengguna untuk lebih jauh masuk ke dalam aplikasi agar lebih memahami konten-konten yang tersedia pada portal.

ODP merupakan user, fungsi, atau penyedia spesifik dan memungkinkan pengguna untuk mengakses, merekam, mengirim atau bertukar informasi dengan lebih cepat dan mudah dalam user yang lebih ramah dan cara yang seragam daripada menggunakan telepon individu OS atau layanan berbasis WAP yang memiliki pelayanan yang berubah-ubah secara signifikan oleh produsen dan operator.

2.2.10 World Wide Web

World Wide Web (biasa disingkat sebagai "Web") adalah sistem dokumen hypertext yang dapat diakses melalui iInternet. Dengan browser Web, kita dapat melihat halaman web yang dapat berisi teks, gambar, video, dan multimedia dan navigasi antara mereka menggunakanhyperlink. Menggunakan konsep hypertext dari sebelumnya, sistem World Wide Web dimulai pada tahun 1989 oleh ahli fisika Inggris Sir Tim Berners-Lee, yang kini Direktur World Wide Web Consortium, dan kemudian oleh Robert Cailliau, seorang ilmuwan komputer Belgia, sedangkan yang kedua bekerja di CERN di Jenewa, Swiss. Di tahun 1990, mereka diusulkan membangun "web dari node" menyimpan "hypertext halaman" dilihat oleh "browser" pada jaringan, dan web yang dirilis pada bulan Desember. yang ada oleh Hubungkan Internet, situs-situs lain yang dibuat, di seluruh dunia, standar internasional untuk menambahkan nama domain & bahasa HTML. Sejak itu, Berners-Lee telah memainkan peran aktif dalam mengawal pengembangan standar Web (seperti bahasa markup di halaman web), dan dalam beberapa tahun terakhir telah mengadvikasi visi beliau yang itu semantik web.

World Wide Web memungkinkan penyebaran informasi melalui internet menjadi mudah digunakan dengan format yang fleksibel. Sehingga ia bermain peranan penting dalam mempopulerkan penggunaan Internet. Walaupun dua istilah tersebut populer digunakan, World Wide Web tidak identik dengan internet. Web adalah aplikasi yang dibangun dan di jalankan di atas internet.

2.2.10.1 Unsur-Unsur dalam Penyediaan Website atau Situs

Untuk menyediakan sebuah website, maka kita harus menyediakan unsurunsur penunjangnya, seperti halnya Nama domain (Domain name/URL - Uniform Resource Locator). Nama domain atau biasa disebut dengan Domain Name atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah website, atau dengan kata lain domain name adalah alamat yang digunakan untuk menemukan sebuah website pada dunia internet. Contoh : http://www.nama situs .com.

Nama domain diperjualbelikan secara bebas di internet dengan status sewa tahunan. Setelah Nama Domain itu terbeli di salah satu penyedia jasa pendaftaran, maka pengguna disediakan sebuah kontrol panel untuk administrasinya. Jika pengguna lupa/tidak memperpanjang masa sewanya, maka nama domain itu akan di lepas lagi ketersediaannya untuk umum. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan website tersebut. Contoh nama domain ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Contoh nama domain ber-ekstensi lokasi Negara Indonesia adalah :

a. co.id : Untuk Badan Usaha yang mempunyai badan hukum sah
b. ac.id : Untuk Lembaga Pendidikan
c. go.id : Khusus untuk Lembaga Pemerintahan Republik Indonesia
d. mil.id : Khusus untuk Lembaga Militer Republik Indonesia
e. or.id : Untuk segala macam organisasi yand tidak termasuk dalam kategori “ac.id”,”co.id”,”go.id”,”mil.id” dan lain lain
f. war.net.id : untuk industri warung internet di Indonesias
g. ch.id : khusus untuk Lembaga Pendidikan yang menyelenggarakan pendidikan seperti SD, SMP dan atau SMU
h. web.id : Ditujukan bagi badan usaha, organisasi ataupun perseorangan yang melakukan kegiatannya di World Wide Web.

2.2.11 Definisi Web Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

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 detil pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

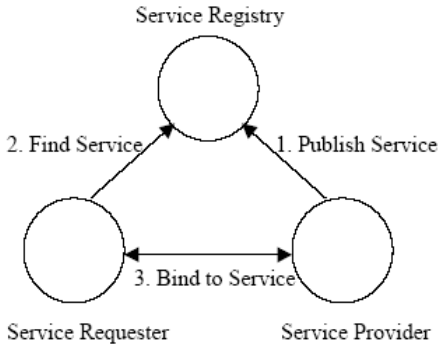
1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

2.2.11.1 Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. Service Requester (peminta layanan)
2. Service Provider (penyedia layanan)

3. Service Registry (daftar layanan)



Gambar 2.6 Arsitektur Web Service

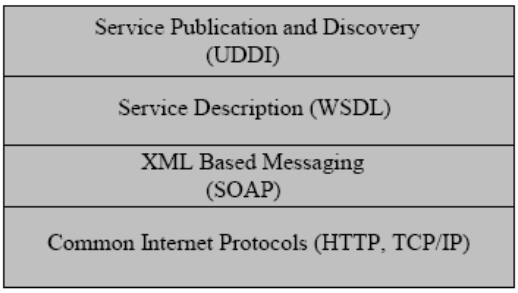
a. Service Provider berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
b. Service Registry berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.
c. Service Requestor meminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.2.11.2 Operasi-Operasi Web Service

Secara umum, web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1. Publish/Unpublish berfungsi untuk menerbitkan/menghapus layanan ke dalam atau dari registry.
2. Find berfungsi sebagai service requestor mencari dan menemukan layanan yang dibutuhkan.
3. Bind berfungsi sebanagi service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

2.2.11.3 Komponen-Komponen Web Service



Gambar 2.7 Komponen Web Service

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di atas, yaitu:

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

2.2.12 Layanan Berbasis Lokasi

Layanan berbasis lokasi atau yang dikenal LBS (*location-based service*) adalah aplikasi komputasi *mobile* yang memberikan layanan kepada pengguna berdasarkan lokasi geografis. Contohnya, "temukan stasiun terdekat", sebuah aplikasi yang memberikan informasi stasiun kereta terdekat.

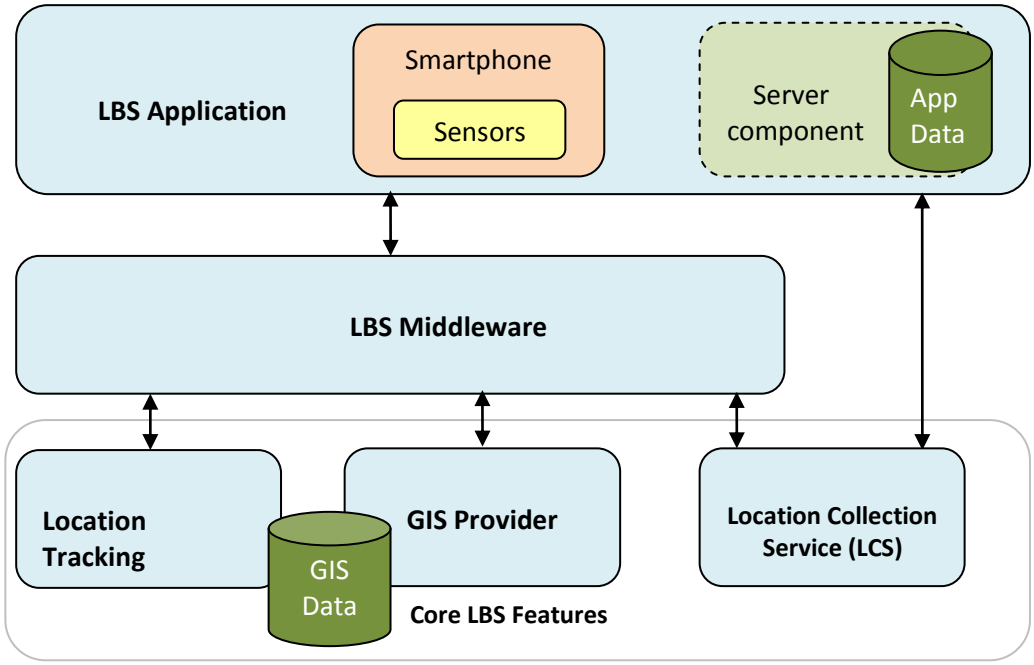
Generasi pertama LBS, reaktif dan fokus pada *client-server*; pengguna akan bertanya pada aplikasi atau sistem untuk sebuah informasi dan kemudian mendapatkan respon. Dengan kemampuan mekanisme *push notification*, peningkatan akses internet mobile dan dikolaborasikan dengan kemampuan Web 2.0, kemampuan LBS akan lebih proaktif dan interaktif. Contohnya :

1. Informasi yang dikirim kepada pengguna berdasarkan lokasi keberadaan sehingga pengguna dapat mencari informasi sesuai kebutuhannya.

2. Orang-orang dapat berbagi (*sharing*) foto dan ulusan informasi dengan model *peer-to-peer*. Sumber data lebih bervariasi, dengan layanan provider yang banyak dibandingkan jika hanya satu sumber.

2.2.12.1 Teknologi dan Komponen-komponen LBS

Layanan Berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan mengunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS terdiri dari beberapa komponen seperti peta (*maps*) dan sistem informasi geografis (GIS), layanan kolektor lokasi (*Location Collective Service*), dan subkomponen spesifik dari aplikasi LBS. Arsitektur dari sebuah LBS dapat digeneralisasi pada gambar 2.8. Deskripsi dari setiap komponen juga ditabulasikan pada tabel.



Gambar 2.8 Diagram Komponen LBS

Tabel 2.1 Komponen – komponen LBS

Komponen	Deskripsi
LBS Application	Mendeskripsikan sebuah aplikasi yang spesifik seperti aplikasi pencarian teman ("*find my friend*"). Aplikasi ini terdiri dari komponen smartphone, mempunyai beberapa sensor, dan komponen *server* yang potensial termasuk data spesifik

	aplikasi (seperti informasi lokasi keberadaan).
LBS middleware	Bagian inti dari fitur-fitur LBS (pelacakan lokasi, layanan GIS, dan layanan pengumpulan lokasi (*location collective services*) untuk memberikan asistensi antarmuka pada aplikasi LBS.
Location Tracking	Komponen ini menyimpan lokasi pelacakan dari pengguna. Merepresentasikan komponen dasar dalam next-generasi LBS seperti data yang memungkinkan rute pengguna ditentukan dan diprediksi. Secara khusus, komponen ini mendukung fungsionalitas sebagai berikut : • Tetap merekam posisi pengguna baik lokasi sekarang maupun lokasi sebelumnya. • Memberikan notifikasi pada komponen lainnya ketika pengguna berpindah dari suatu tempat ke tempat yang lain. Mendukung lokasi berdasarkan notifikasi yang dikirim ke pengguna. • Menentukan lokasi ketika pengguna berada dalam lokasi yang telah ditentukan. Mendukung fitur *geocasting*. • Queri dari jejak lokasi untuk men-generasi model perpindahan pengguna.
GIS Provider	Komponen ini menyediakan fungsionalitas *geospatial* untuk beberapa LBS termasuk peta informasi, peta visualisasi, dan layanan direktori. Google maps dengan API nya dapat dianggap sebagai penyedia GIS. Contohnya, deCarta dan OpenRouteService.org
Location Collection Service (LCS)	Komponen ini mengumpulkan lokasi untuk mendapatkan garis lintang (*latitude*) dan garis bujur (*longitude*) untuk lokasi spesifik pengguna. Berdasarkan pada teknologi, komponen ini mungkin diakses melalui *middleware* LBS (misalnya, triangulation jaringan *mobile* melalui penyedia layanan) atau secara langsung (contohnya, via GPS *receiver* dalam *smartphone*).

2.2.13 *Global Positioning System (GPS)*

2.2.13.1 Pengenalan *Global Positioning System* (GPS)

GPS adalah singkatan dari Global Positioning System, sistem satelit yangdapat memberikan posisi Anda di mana pun di dunia ini. Satelit GPS tidakmentransmisikan informasi posisi Anda, yang ditransmisikan satelit adalahposisi satelit dan jarak penerima GPS Anda dari satelit. Informasi ini diolah alat penerima GPS Anda dan hasilnya ditampilkan kepada Anda.GPS (Global Positioning System) dapat diartikan sebagai cara untukmenentukan posisi secara

global berdasarkan referensi tertentu. Cara menentukan posisi ini menggunakan system radio navigasi dan satelit.

Referensi yang digunakan dalam hal ini adalah datum World Geodetic Surveyth 1984 (WGS 84). Sistem GPS ini dapat digunakan oleh banyak orangsekaligus dan didesain untuk memberikan ketepatan posisi 3D (longitude,latitude, altitude / X, Y, X) dan informasi waktu secara kantiyu diseluruhdunia. Dengan GPS memungkinkan setiap meter persegi di permukaan bumiuntuk diberi alamat yang unik.

GPS sekarang digunakan untuk keperluan-keperluan yang beragam-sebagianmalah tidak pernah terbayangkan oleh pembuat GPS. GPS bisa digunakan dihampir semua lokasi di dunia kecuali di tempat-tempat di mana sinyal satelit GPS tidak dapat diterima, misalnya, di dalam gedung, di dalam goa, di bawahtanah, dan di bawah permukaan air.

GPS sebenarnya adalah proyek Departemen Pertahanan Amerika Serikat(AS) yang memberinya nama resmi NAVSTAR (NAVigation Satellite TimingAnd Ranging). Bagian utama dari sistem GPS adalah 24 satelit yangmengorbit Bumi di ketinggian 20.200 kilometer . Orbit satelit dirancangsehingga setiap titik di Bumi dapat melihat paling sedikit empat satelit padasetiap saat.

Tiap satelit mengitari bumi kira-kira sekali dalam 12 jam dengan kecepatansekitar 11.000 kilometer per jam. Satelit GPS mempunyai panel-panelpengumpul tenaga Matahari untuk membangkitkan energi listrik yangdiperlukannya. Selain itu juga ada baterai yang menyimpan tenaga listrik danmempergunakannya saat satelit tidak memperoleh sinar matahari.

Satelit GPS pertama diluncurkan tahun 1978 dan konstelasi 24 satelit berhasildilengkapi tahun 1994. Setelah itu satelit-satelit baru rutin diluncurkan untukmeng-upgrade satelit lama atau mengganti satelit yang rusak/tidak berfungsilagi. Tiap satelit mentransmisikan data navigasi dalam sinyal CDMA (Code Division Multiple Access)-sama seperti jenis sinyal untuk telepon seluler CDMA. Sinyal CDMA menggunakan kode pada transmisinya sehinggapenerima GPS tetap bisa mengenali sinyal navigasi GPS walaupun adagangguan pada

frekuensi yang sama. Frekuensi yang digunakan adalah L1(1575,42 MHz) dan L2 (1227,6 MHz).

Informasi yang ditransmisikan dari satelit ke penerima GPS terdiri dari dua jenis. Yang pertama disebut "almanak", yaitu posisi dari semua satelit GPS.Jenis informasi kedua disebut "efemeris", yaitu koreksi data almanak. "Kalau'almanak' di-update kira-kira seminggu sekali, data 'eferemis' biasanya di-update tiap setengah jam. Alat penerima GPS yang dinyalakan kembalisetelah seharian dimatikan masih bisa menggunakan data almanak sebelumnya.

Fungsi GPS ketika dikembangkan pertama kali adalah untuk menentukanposisi dan navigasi tetapi sejalan dengan perkembangan Sistem Informasi Geografis (SIG) maka fungsi GPS bertambah dengan memberikan informasikenampakan (data cupture), dapat memberikan gambaran dan menghitungluas suatu bentuk feature tertentu, dan merupakan suatu system informasiyang mampu diakseske berbagai software GIS.Perkembangan GPS ini tentu saja menuntut keseragaman tertentu untukpenentuan posisi dan jarak dalam system yang digunakan yaitu referensi geografis yang digunakan, dalam hal ini adalah WGS 84 (World Geodetic Survey 84) yang digunakan sekarang dibagai dalam tiga segmen:

1. Segmen Kontrol

   Merupakan inti dari system GPS yaitu stasiun pengontrol satelit yangselalu melakukan monitor untuk mengetahui secara pasti posisi,ketinggian, dan kecepatan satelit. Informasi itu Informasi itu dikirimkankembali ke satelit, dan satelit akan mengirimkan informasi tersebut kereceiver di bumi. Station pengontrol ini dioperasikan oleh Departemen Pertahanan Amerika Serikat.

2. Segmen Angkasa

   Susunan satelit yang memancarkan sinyal GPS ke bumi, saat inisudah ada 24 satelit NAVSTAR (Navigation Satelite for Time and Ranging) yang mengorbit bumi dan digunakan untuk GPS.

3. Segmen Pemakai

   Sinyal GPS dari satelit dimanfaatkan oleh pemakai di bumi untuk melakukan perhitungan posisi guna berbagai macam keperluan.

2.2.13.2 Cara Kerja GPS

Setiap satelit mentransmisikan dua sinyal yaitu L1 (1575.42 MHz) dan L2(1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal pseudo-random yaitu kode P (Protected) dan kode C/A (coarse/aquisition). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehinggapenerima (perangkat GPS) dapat mengidentifikasi sinyal dari setiap satelit.Pada saat fitur ” Anti-Spoofing ” diaktifkan, maka kode P akan dienkripsi danselanjutnya dikenal sebagai kode P(Y) atau kode Y.Perangkat GPS yang dikhususkan buat sipil hanya menerima kode C/A padasinyal L1 (meskipun pada perangkat GPS yang canggih dapat memanfaatkansinyal L2 untuk memperoleh pengukuran yang lebih teliti.

Perangkat GPS menerima sinyal yang ditransmisikan oleh satelit GPS. Dalammenentukan posisi, kita membutuhkan paling sedikit 3 satelit untuk penentuanposisi 2 dimensi (lintang dan bujur) dan 4 satelit untuk penentuan posisi 3dimensi (lintang, bujur, dan ketinggian). Semakin banyak satelit yangdiperoleh maka akurasi posisi kita akan semakin tinggi. Untuk mendapatkan sinyal tersebut, perangkat GPS harus berada di ruang terbuka. Apabila perangkat GPS kita berada dalam ruangan atau kanopi yang lebat dandaerah kita dikelilingi oleh gedung tinggi maka sinyal yang diperoleh akansemakin berkurang sehingga akan sukar untuk menentukan posisi dengan tepat atau bahkan tidak dapat menentukan posisi.

2.2.13.3 Ketelitian Posisi GPS

Ketelitian posisi yang didapat dengan pengamatan GPS secara umum akanbergantung pada beberapa faktor yaitu :

a. Kualitas Receiver GPS
b. Jumlah satelit
c. Lama pengamatan

d. Metode penentuan posisi yang digunakan (absolute & differential positioning, kinematic, dll)
e. Strategi Pemrosesan data (Rial time & Post processing, dll)

Luas spektrum ketelitian posisi yang diberikan oleh GPS adalah salah satukeindahan dari GPS, karena pemakai GPS punya keleluasaan dalammelaksanakan penentuan posisi sesuai dengan tingkat ketelitian yangdiperlukan secara optimal dan efisien (baik waktu maupun biaya). Olehsebab itu tidak dapat dipungkiri bahwa GPS dapat melayani cukup banyak aplikasi, dengan tuntutan ketelitian yang beragam.

2.2.14 Metode Penentuan Posisi

Sejak tahun 1978, terdapat 59 satelit GPS yang berhasil di orbitkan mengelilingi bumi, walaupun pada tahun 2010, hanya 30 satelit yang dinyatakan masih beroperasi dengan normal. Kemampuan manusia untuk menentukan lokasi dengan tepat telah dikenal lama dengan menggunakan asap sebagai sinyal untuk menandakan suatu tempat atau lokasi.

GPS dirancang khusus untuk sistem navigasi di bumi. Jika digambarkan dalam peta dunia, maka posisi kita dapat ditandai dengan sebuah titik koordinat lokasi tempat dimana kita berada, seperti pada gambar 2.7**,** titik yang ditandai terdiri atas dua komponen, yaitu *latitude* dan *longitude,* yang menginformasikan keberadaan kita di bumi melalui perangkat lunak GPS. Ketika titik koordinat lokasi diketahui, maka informasi ini dapat digunakan oleh program GPS untuk mendapatkan informasi lebih banyak untuk pengguna, seperti bisnis terdekat, dan informasi kemacetan lalu lintas. Ketika titik koordinat telah ditentukan, aplikasi akan menggunakan proses *geolocation* untuk mendapatkan informasi mengenai lokasi disekitar pengguna.

Tentunya penentuan lokasi tidak hanya berasal dari sistem GPS. Informasi yang digunakan dan bagaimana proses yang terjadi pada perangkat dan mem-*parsing* informasi lokasi adalah berdasarkan pada jenis perangkat yang digunakan. Terdapat beberapa metode untuk menentukan lokasi pada perangkat komputasi

dan tidak semua berdasarkan pada satelit GPS. Berikut beberapa cara penentuan lokasi:

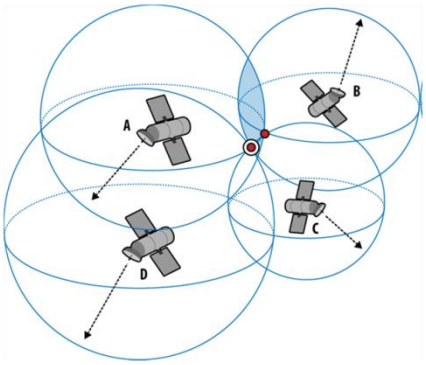


Gambar 2.9 Pinpoint Lokasi Keberadaan Dimanapun di Bumi

1. *Global Positioning Service* (GPS)

   Satelit GPS secara terus-menerus mengirim informasi sehingga perangkat GPS dapat mengurainya (*parse*), contohnya: susunan GPS yang masih berfungsi dengan baik, dimana semua satelit berada dalam orbit yang tepat atau jalur transmisi satelit, dan waktu transmisi. Perangkat *receiver* akan mengkalkulasikan posisinya dengan waktu pengiriman sinyal dengan beberapa satelit dalam *array* yang terlihat (tanpa hambatan ke angkasa).

   Alat penerima menentukan waktu yang dibutuhkan untuk menerima setiap pesan kemudian menghitung jarak dari setiap satelit berdasarkan informasi ini. Jarak pada setiap satelit dari alat penerima, orbit, dan perhitungan *trilateration* menginformasikan alat penerima mengenai posisi sekarang. Sementara tiga transmitter radio *triangulation* cukup untuk menentukan lokasi yang masuk akal (*reasonable location*), terdapat faktor waktu yang menjadi perbandingan dengan satelit yang sedang orbit. Proses ini membutuhkan waktu, mungkin beberapa detik agar sinyal satelit menjangkau bumi – setiap kesalahan kecil jam pada satelit, dikalikan dengan waktu sekarang dapat menjadi kesalahan besar penentuan lokasi. Menggunakan keempat satelit mengurangi kesalahan dalam persamaan. Dalam kasus yang paling sering terjadi, sebuah alat *receiver* akan menggunakan empat satelit atau lebih untuk menghitung posisinya.



Gambar 2.10 Sistem GPS Terhubung ke Satelit

2. *IP Address*

IP (*Internet Protocol*) *address* merupakan angka unik yang diberikan pada setiap perangkat yang terhubung dengan internet sehingga dapat berkomunikasi dengan perangkat lainnya. Ketika perangkat terhubung dengan internet, perangkat dapat mengirim data kepada perangkat lain dan kemudian menerima respon kembali. Meskipun sebuah perangkat menerima alamat yang lebih atau selalu berubah-ubah, *IP address* dapat berubah statis (permanen) atau dinamis (temporar). Tanpa memperhatikan jenis alamat yang dimiliki oleh sebuah perangkat, alamat tersebut selalu akan terdiri dari empat kumpulan angka yang dipisahkan oleh titik, misalnya : 123.123.123.123.

Umumnya, sebuah *IP Address* ditentukan oleh ISP (*Internet Service Provider*) dalam bagian-bagian berdasarkan wilayah oleh lembaga *registry* lokal. Oleh karena itu, negara, wilayah, dan bahkan sebuah kota dapat dengan mudah di identifikasi dengan sebuah *IP Address*. Selanjutnya, kemajuan saat ini dalam pengumpulan data dan pengelolaannya oleh ISP, perangkat lokasi geografis mampu menentukan sebuah lokasi dengan tingkat akurasi tinggi. Yang menjadi tantangan terbesar dengan penggunaan IP Address untuk menentukan sebuah lokasi adalah bahwa terdapat ratusan institusi regional yang akan diproses untuk mendapatkan data.

3. GSM/CDMA Cell ID

Cell ID merupakan angka unik yang digunakan untuk mengidentifikasi setiap perangkat *mobile* pada jaringan seluler, umumnya seperti *IP Address* mengidentifikasikan sebuah perangkat dalam jaringan.

Ada dua tipe jaringan yang paling popular digunakan yaitu *Global System for Mobile Communication* (GSM) dan *Code Division Multiple Access* (CDMA). Jenis layanan seluler yang dibutuhkan didasarkan pada daerah cakupan sinyal.

GSM merupakan teknologi jaringan *mobile* yang paling tertua dan oleh karena itu peningkatan pelayanan dari teknologi GSM tersedia luas dibanding teknologi lainnya. GSM adalah teknologi 2G yang tersedia lebih dari 200 negara, dan jumlah *transfer* data pada jaringan ini lebih dari 75% dari seluruh dunia. Kemampuan jaringan GSM untuk dilakukan migrasi ke layanan jaringan 3G dan 4G (*Evolved High Speed Packet Access* (HSPA+), *Long Term Evolution* (LTE), dan *Service Architecture Evolution* (SAE) juga sangat cepat, sehingga dapat melayani hampir empat juta pelanggan. Meskipun teknologi GSM tidak dapat menangani jumlah data yang sama seperti teknologi lainnya, jaringan GSM memiliki tingkat kualitas jaringan yang tinggi karena kemampuannya untuk menempatkan repeater didalam dan diluar bangunan bahkan jumlah penggunaan repeater yang semakin dikurangi. Hal ini menjadikan jaringan GSM sebagai operator dengan layanan jaringan yang baik dengan tingkat gangguan jaringan yang kecil.

CDMA merupakan teknologi terbaru setelah GSM, baik CDMA dan GSM sama-sama merupakan layanan jaringan 2G (cdmaOne(TM)), dan 3G (CDMA2000(R) dan WCDMA). Keuntungan dari teknologi CDMA adalah dapat mengijinkan banyak pengguna dalam waktu yang bersamaan dengan porsi frekuensi yang diberikan oleh *band*, dimana tidak dapat dilakukan pada jaringan GSM. Jaringan CDMA, seperti GSM juga dapat dilakukan migrasi ke teknologi LTE untuk mendukung jaringan 4G.

Tanpa memperhatikan jenis teknologi dari perangkat *mobile* yang digunakan, prinsip dasarnya adalah mempunyai *identifier unique* dalam jaringan. Menggunakan *triangulation*, *latitude* dan *longitude* Cell ID dapat diidentifikasi, dengan dilakukan *geolocated*. Penggunaan beberapa tower untuk melakukan triangulation posisi dari perangkat mobile, mempunyai tingkat akurasi tinggi. Hal ini mengapa *geolocation* yang bergantung pada

jenis teknologi, bekerja lebih baik dalam lingkungan urban (perkotaan) – dimana keberadaan tower yang satu dengan lainnya berdekatan dibanding pada area pedesaan.

4. *Wifi* dan *Bluetooth MAC Address*

   Wifi dan *Bluetooth MAC address* mempunyai kinerja seperti *IP Address. Media Access Control (MAC) address* merupakan pemberian angka unik pada suatu perangkat, biasanya dilakukan oleh pabrik pembuat perangkat tersebut. Angka ini sifatnya permanen dan sebagai penanda global, namun pada perangkat modern, angka ini dapat diganti secara manual; dikenal dengan *MAC spoofing* (teknik untuk mengganti default *MAC address* dari antarmuka jaringan pada perangkat jaringan). Contoh sebuah MAC Address : 12-34-56-78-9A-BC. *MAC address* untuk router wifi adalah alamat yang mudah untuk ditemukan atau ditandai pada perangkat nirkabel (*wireless*), begitupun dengan MAC *address* untuk perangkat Bluetooth. Menggunakan alamat ini dengan cara yang sama sebagai alamat IP, *latitude* (garis lintang) dan *longitude* (garis bujur), dan lokasi fisik, dapat ditemukan.

2.2.15 *Geotagging*

Geotagging adalah proses penambahan identifikasi metadata geografis terhadap berbagai media contohnya, *geotagged photografi*, atau video, websites, pesan SMS, *QR code*, atau *RSS feed* dan merupakan bentuk dari metada *geospatial*. Data tersebut biasanya terdiri dari koordinat garis lintang (*latitude*) dan garis bujur (*longitude*), dan mungkin juga termasuk ketinggian (*altitude*), bantalan poros (*bearing*), jarak, data akurasi, dan nama tempat.

Geotagging dapat membantu pengguna untuk menemukan informasi posisi keberadaan spesifik dari lingkungan yang luas. Contohnya, seseorang dapat menemukan gambar terdekat dengan memberikan informasi koordinat *latitude* dan *longitude* ke dalam mesin pencari gambar yang sesuai. *Geotagging* dapat memberikan informasi layanan yang potensial digunakan untuk menemukan lokasi berdasarkan berita, website, dan sumber lainnya. *Geotagging* dapat memberitahukan kepada pengguna lokasi dari konten dari gambar atau media

lainnya atau titik pandangan (*point of view*), dan sebaliknya pada beberapa platform media menunjukkan media yang relevan ke lokasi tertentu.

Terkait dengan istilah *geocoding* yang mengacu pada proses menentukan geografis *non* kordinat berdasarkan pengidentifikasi geografi, seperti alamat jalan, dan penemuan yang diasosiakan dengan koordinat geografi. banyak teknik yang dapat digunakan bersama dengan *geotagging* untuk menyediakan teknik pencarian alternatif.

2.2.16 *Geolocation*

Geolocation adalah identifikasi lokasi geografis suatu objek pada dunia nyata. Contohnya, radar, telepon genggam, atau terminal koneksi internet komputer. *Geolocation* mempunyai kaitan erat dengan *positioning*, perbedaannya adalah *geolocation* lebih spesifik dalam menentukan sebuah lokasi (misalnya alamat jalan) dibandingkan dengan *positioning* yang hanya mencakup sekumpulan koordinat geografis. Secara spesifik melibatkan penggunaan sistem lokasi frekuensi radio (RF), misalnya, perbedaan waktu kedatangan (TDOA = *Time Different of Arrival*) dimana memungkinkan spesifikasi lokasi yang baik. Sistem TDOA memanfaatkan pemetaan (*mapping*) atau sistem informasi geografis lainnya. Hal ini berbeda dengan teknologi *radiolocation* (pencarian lokasi dengan menggunakan gelombang radio) konvensional, misalnya pencarian langsung (*direct finding*) dimana sebuah garis perpindahan (*line of bearing)* ke pemancar dicapai dan bukan spesifik lokasi.

Internet dan komputer memungkinkan *geolocation* dapat dilakukan dengan menggabungkan lokasi geografis dengan *internet protocol* (IP) *address*, MAC *Address*, RFID (*Radio-frequency identification*), perangkat keras dengan nomor produksinya, angka seri perangkat lunak (contohnya, UUID, Exif/IPTC/XMP atau modern Steganography), *invoice*, *wifi*, GPS, dan masih banyak lagi. *Geolocation* umumnya bekerja secara otomatis mencari alamat IP pada layanan WHOIS (query atau respon protokol yang sangat luas digunakan dalam query database untuk menyimpan pengguna terdaftar atau bertindak sebagai sumber data internet,

seperti nama domain, IP address, atau sistem *autonomous*) dan mengambil *physical address* terdaftar.

Data geolocation alamat IP mencakup informasi seperti negara, wilayah, kota, kode pos, *latitude*, *longitude*, dan zona waktu. Sekumpulun data yang lebih khusus dapat menentukan parameter lainya seperti nama domain, kecepatan koneksi, ISP, bahasa, *proxy*, nama perusahaan, US DMA/MSA, NAICS codes, dan rumah ataupun bisnis.

2.2.16.1 *Latitude* (Garis Lintang)

Dalam geografi, garis lintang adalah garis khayal yang digunakan untuk menentukan lokasi di bumi terhadap garis khatulistiwa (utara atau selatan). Posisi lintang biasanya dinotasikan dengan simbol huruf Yunani (φ). Posisi lintang merupakan perhitungan sudut dari 0^0 di khatulistiwa sampai ke $+90^0$ di kutub utara dan -90^0 di kutub selatan.

2.2.16.2 *Longitude* (Garis Bujur)

Bujur kadangkala dinotasikan oleh huruf abjad Yunani (λ), menggambarkan lokasi sebuah tempat di timur atau barat daya Bumi dari sebuah garis utara-selatan yang disebut Meridian Utama. Longitude diberikan berdasarkan pengukuran sudut yang berkisar dari 0^0 di Meridian Utama ke $+180^0$ arah timur dan -180^0 arah barat. Tidak seperti lintang yang memiliki equator sebagai posisi awal alami, tidak ada posisi awal alami, tidak ada posisi awal untuk bujur. Oleh karena itu, sebuah dasar meridian harus dipilih. Meskipun kartografer Britania Raya telah lama menggunakan meridian Observatorium Greenwich di London, referensi lainnya digunakan di tempat yang berbeda, termasuk Ferro, Roma, Kopenhagen, Yerusalem, Saint Petersburg, Pisa, Paris, Philadelphia, dan Washington, D.C.. Pada tahun 1884, Konferensi Meridian Internasional mengadopsi meridian Greenwich sebagai Meredian utama universal atau titik nol bujur.

2.2.17 Google Maps

Google maps adalah teknologi dan aplikasi layanan web pemetaan yang disediakan oleh Google, sifat bebas (tidak untuk penggunaan komersial), yang banyak digunakan layanan berbasis peta, mencakup website Google Maps, Google Ride Finder, Google Transit, dan peta yang digunakan oleh penyedia layanan pihak ketiga melalui Google Maps API. Google maps menawarkan peta jalan (*street maps*), sebuah perencana rute (*route planner*) untuk perjalanan dengan menggunakan mobil, pejalan kaki, sepeda atau angkutan umum dan pencari bisnis perkotaan untuk beberapa negara di seluruh dunia. Satelit Google Maps, gambar tidak diperbaharui dalam waktu nyata (realtime); mungkin beberapa minggu atau tahun lamanya.

Google Maps menggunakan varian yang mirip dengan *Mercator projection* (proyeksi peta silender diperkenalkan oleh *geographer* Belgian dan *cartographer* Gerardus Mercator, pada tahun 1569) sehingga tidak dapat menampilkan wilayah kutub. Produk yang berelasi dengan Google Maps adalah Google Earth, program yang berdiri sendiri (*stand-alone*) yang menawarkan fitur pandangan globe, termasuk area kutub.

2.2.17.1 Google Maps API

Diperkenalkan oleh Google pada Juni 2005, Google Maps API adalah fitur yang memungkinkan para pengembang meng-integrasikan Google Maps ke dalam website mereka. Google Maps API adalah layanan gratis (free). Penggunaan Google Maps API, dimungkinkan untuk menanamkan Google Maps kedalam website luar, untuk itu data spesifik dapat ditambahkan. Walaupun pada awalnya hanya JavaScript API, Maps API telah dikembangkan termasuk API untuk aplikasi *adobe flash*, layanan untuk mengumpulkan gambar peta static, dan layanan web *(web service*) untuk melakukan *geocoding*, menghasilkan arah kemudi (*driving directions*), dan informasi elevasi.

2.2.17.2 *Google Maps for Mobile*

Pada tahun 2006, Google memperkenalkan aplikasi Java yang diberi nama *Google Maps for mobile*, ditujukan untuk telepon genggam berbasis Java atau perangkat mobile lainnya. Pada tanggal, 28 november 2007, *Google Maps for Mobile* 2.0 dirilis. Fitur baru dalam versi ini adalah layanan lokasi *non*-GPS, layanan yang tidak membutuhkan perangkat penerima GPS. Fitur "*my location*" bekerja dengan memanfaatkan lokasi GPS dari perangkat *mobile*, jika tersedia. Informasi ini dilengkapi dengan perangkat lunak untuk menentukan jaringan nirkabel (*wireless*) dan *cell site* (tower BTS) terdekat. Perangkat lunak akan mencari lokasi dari tower BTS menggunakan database pengenal jaringan *wireless* dan *cell site*. Metode *cell side* menggunakan triangulasi perbedaan kekuatan sinyal dari *cell transmitter* yang berbeda dan selanjutnya menggunakan properti lokasi (diperoleh dari database cell site online) untuk membantu "My Location" menentukan lokasi pengguna. Metode jaringan *wireless* (jaringan nirkabel) mengkalkulasikan dengan menjangkau hotspot wifi terdekat, dan menggunakan property lokasi (diperoleh dari database wifi online,cara yang sama dengan database cell site) untuk kemudian menemukan lokasi pengguna. Berikut proses urutannya :

1. *GPS-based Services*
2. *WLAN, Wifi-based Services*
3. *Cell Transmitter-based Services*

2.2.17.3 Google Maps Android 2.0

Saat ini, penggunaan telepon genggam sangat meningkat sebagai alat bantu navigasi. Google *Navigation for* Android 2.0 adalah layanan gratis.

Fitur-fitur yang terdapat dalam aplikasi adalah sebagai berikut :

1. Pencarian dalam bahasa Inggris
2. Pencarian dengan suara
3. Pandangan lalu lintas
4. Pencarian sepanjang rute
5. Pandangan satelit

6. Pandangan jalan
7. *Car dock mode*

2.2.18 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-*Mobile*, dan Nvidia.

2.2.18.1 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android juga sudah bergabung dengan beberapa smart *Mobile* seperti Nokia, Sony Ericsson, dan lainnya.

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

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*Mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

1. **Android versi 1.1**

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

2. **Android versi 1.5 (Cupcake)**

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

3. **Android versi 1.6 (Donut)**

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indicator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada

kamera, camcorder dan galeri yang dintegrasikan pada CDMA / EVDO, 802.1x, VPN, Gestures, dan *Textto- speech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

4. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan *Google Maps* 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *Mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

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

6. **Android versi 2.3 (Gingerbread)**

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

7. **Android versi 3.0 (Honeycomb)**

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

8. **Android versi 4.0 (ICS :Ice Cream Sandwich)**

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

2.2.18.2 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk di dalamnya Symbian, iOS, Windows *Mobile*, BlackBerry, Java *Mobile* Edition, Linux *Mobile* (LiM), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut:

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royaliti. Sementara pengembang *software* menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.
2. Arsitektur komponen dasar android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun *game* yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual* Dalvik, sehingga kode program *portabel* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.2.19 JSON

JSON (*JavaScript Object Notation*) 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**

   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).

2. **Larik**

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

3. **Nilai**

   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.

4. **String**

   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.

5. **Angka**

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

2.2.20 Java

Java adalah salah satu bahasa pemrograman berorientasi objek (OOP-Object Oriented Programming). Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek.

Pemrograman Berorientasi Obyek (OOP)**,** Pemisalan Objek dalam OOP Objek-objek dalam dunia nyata, mempunyai 2 karakteristik khusus : Status dan Perilaku. Contohnya, sepeda punya status (jumlah gir, jumlah pedal, dua buah ban) dan perilaku (mengerem, mempercepat, ubah gir). Bahasa yang berorientasi pada objek pun mempunyai karakteristik yang sama dengan objek-objek di dunia nyata. Yaitu status yang dalam bahasa pemrograman biasanya disimpan sebagai Variabel dan perilaku yang diimplementasikan sebagai Method.

2.2.20.1 Sejarah Java

Dimulai pada tahun 1991 dibentuknya suatu tim yang diberi nama green Tim ini dipimpin oleh Patrick Naughton dan James Gosling (Sekelompok insinyur Sun). Awalnya mereka ingin membuat suatu bahasa komputer yang dapat digunakan oleh TV kabel (Cable TV Box) yang memiliki memori kecil dan setiap

perusahaan memiliki tipe yang berbeda. Untuk mengaplikasikan hal tersebut mereka menggunakan hal yang pernah dicoba oleh bahasa pascal.

Sebelumnya Niclaus Wirth telah membuat sebuah bahasa yang portable yang akan digunakan dalam mesin bayangan selanjutnya ini disebut sebagai Virtual Mesin namun saat itu belum diumumkan. Akhirnya tim green mengunakan ide ini akhirnya mereka menciptakan Java Virtual Mesin. Vitual mesin inilah yang menbuat java dapat dijalankan diberbagai platform.

Pada tahun 1992 tim green membuat produknya yang diberi nama *7 (Star Seven) Namun produk ini gagal dipasaran. Setelah itu dibuat produk yang baru. Inilah java, pada awalnya bahasa pemrograman yang dibuat tersebut diberi nama “oak “ (Kemungkinan nama ini diambil dari nama pohon yang ada didepan jendela James Gosling) Namun setelah diperiksa ternyata nama oak sudah pernah digunakan. Maka lahirlah nama “ Java “.

Pada tahun 1993 sampai pertengahan 1994 tim green yang mempunyai nama baru First person. Inc hanya menghabiskan waktu dengan memperhatikan penjualan dari produknya. Ternyata tidak satupun orang ditemukan. Akhirnya pada tahun 1994 First Person dibubarkan. Pada tahun itu Web berkembang sangat pesat saat itu browser yang banyak digunakan adalah Mosaic suatu broser yang gratis.

Dalam suatu wawancara pada pertengahan tahun 94 James Gosling mengatakan “We could build a real cool browser. It was one of the few things in the client/serve mainstream that needed some of the weird things we’d done: architecture neutral, real-time,reliable, secure—issues that weren’t terribly important in the workstation world. So we built a browser.” Akhirnya dibuatlah suatu browser oleh Patrick Naughton dan Jonathan Payne. Browser itu diberi nama Hot Java. Browser ini dibuat dengan menggunakan bahasa java untuk menunjukkan kemapuan dari java.

Yang tidak kalah hebatnya mereka juga membuat teknologi yang disebut Aplet sehingga browser dapat menjalankan kode program didalam browser. Inilah bukti dari teknologi itu yang ditunjukkan java pada tanggal 25 mei 1995 Sejak saat itu browser java memberikan lisensinya kepada browser lain untuk bisa

menjalankan Applet dan pada tahun 1996 diberikan pada Internet Explorer. Akhirnya pada tahun 1996 dikeluarkan versi pertama dari java dengan mengeluarkan java 1.02

Pada tahun 1998 dikeluarkan java 1.2 dan java mulai mengeluarkan slogannya "Write once, Run anywhere.

2.2.20.2 Edisi Java

Java adalah bahasa yang dapat dijalankan dimanapun dan di sembarang platform apapun, di beragam lingkungan : internet, intranets,consumer electronic products, dan computer applications.The java 2 platform tersedia dalam 3 edisi untuk keperluan berbeda. Untuk beragam aplikasi yang dibuat dengan bahasa Java, java dipaketkan dalam edis – edisi berikut :

a. Java 2 Standard Edition (J2SE)
b. Java 2 Enterprise dition (J2EE)
c. Java 2 Micro Edition (J2ME)

Masing – masing edisi berisi Java 2 Software Development Kit (J2SDK) untuk mengembangkan aplikasi dan Java 2 Runtime Environent (J2RE) untuk menjalankan aplikasi.

2.2.20.3 Kelebihan Java

Beberapa kelebihan Java adalah sebagai berikut :

1. Multiplatform. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi

menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan bytecode tersebut.

2. Objek Oriented Programming (OOP) yang artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut objek. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antara komputer sekalipun.
3. Perpustakaan kelas Yang Lengkap. Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program-program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.21 MySQL

MySQL adalah implementasi dari sistem menejemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, dan pemakaiannya tidak untuk kepentingan komersial. MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*), adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.2.21.1 Sejarah MySQL

MySQL diciptakan oleh MySQL AB, sebuah perusahaan yang didirikan oleh Michael Widenius (Monty), David Axmark, dan Allan Larsson, di Swedia pada tahun 1995. Kemudian pada tahun 2008, Sun Microsystem,Inc. mengakusisi perusahaan tersebut dengan perkiraan harga $1 billion US dollar. Pada tanggal, 27 januari 2010, Sun Microsystem, Inc diakusisi oleh Oracle Corporation, setelah kedua perusahaan tersebut menandatangi persetujuan pada tanggal 20 april 2009, Sun Microsystem, Inc kemudian bergabung (*merged*) dengan Oracle USA, Inc. kemudian perusahaan tersebut berubah nama menjadi Oracle America,Inc.

2.2.21.2 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Microsoft Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-*user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Peningkatan performa (*performance tuning*). MySQL memiliki kecepatan yang menakjubkan dalam manangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki lapisan keamanan seperti level subnetmask, nama *host*, dan izin akses user dengan sistem perizinan yang mendatail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan *protokol* TCP/IP, Unix *socket* (UNIX), atau *Named Pipes* (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur Tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.22 PHP

Adalah bahasa *script server side* didesain untuk pengembangan web untuk menghasilkan sebuah web dinamis. Untuk itu, kode PHP disisipkan kedalam skrip HTML dan diproses atau diterjemahkan oleh web server dengan sebuah modul processor PHP untuk kemudian menghasilkan halaman web.

2.2.22.1 Sejarah PHP

Pada awalnya PHP merupakan singkatan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengola data formulis dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP.FI. dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Dalam tahun yang sama, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih., lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP, *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.2.22.2 Kelebihan PHP

Beberapa kelebihan PHP dari bahasa pemrogram web, antara lain :

1. Bahasa pemrograman PHP adalah sebuah skrip yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lightpd hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.2.23 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdari dari program Apache, HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), A (Apache), M (MySQL), P (PHP), P (Perl). Program ini tersedia dibawah lisensi GNU Public Lisence dan bertindak sebagai web server yang mengelola halaman web dinamis. XAMPP tersedia untuk sistem operasi Microsoft Windows, Linux, Solaris, dan Mac OS. Umumnya perangkat lunak ini digunakan untuk projek pengembangan web.

2.2.24 Eclipse

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

1. *Multi-platform*: Dapat berjalan pada berbagai platform seperti, Microsoft Windows, Linux, Solaris, AIX, HP-UX, dan Mac OS X.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Phyton, 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 yang difavoritkan karena gratis dan *open source*, yang berarti setiap orang boleh melihat kode program perangkat lunak. Selain itu, kelebihan dari Eclipse yang membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponan yang dinamakan *plug-in*.

2.2.24.1 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 lebih lanjut dan pengaturan organisasinya.

2.2.24.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkut 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*

Standard pengoperasian 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 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.

2.2.24.3 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 pengembangan (*deployment*) dan perawatan (*maintenance*) untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

Tabel 2.2 Perkembangan Versi Eclipse

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

2.2.25 *Software Development Kit* (SDK)

Software Development Kit (SDK atau *devkit*) adalah sekumpulan alat pengembangan yang memungkinkan untuk menciptakan sebuah aplikasi untuk paket perangkat lunak tertentu (*software package*), framework perangkat lunak, platform perangkat keras, komputer sistem, konsol video game, sistem operasi, atau platform yang sama.

2.2.26 *Java Development Kit* (JDK)

Java Development Kit atau disingkat JDK merupakan produk dari Oracle Corporation yang ditujukan untuk pengembangan perangkat lunak berbasis Java. Sejak Java diperkenalkan, sampai saat ini Java SDK telah banyak digunakan. Pada tanggal, 17 November 2006, Sun memperkenalkan JDK dibawah lisensi GNU *General Public License* (GPL), sehingga penggunaannya bebas (gratis).

JDK juga dilengkapi dengan JRE (*Java Runtime Environment*), biasanya disebut *private*. JRE terdiri dari JVM (*Java Virtual Machine*) dan semua *library class* yang terdapat lingkungan produksi, sama baiknya dengan penambahan hanya library-library yang berguna bagi *developers*, seperti *library internationalization* dan *library* IDL.

2.2.27 *Android Development Tools* (ADT)

Android Development Tools (*ADT*) adalah plugin untuk Eclipse *Intergrated Development Environment* (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android.

ADT memperluas kemampuan Eclipse untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom XML editor, dan debug panel ouput. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.