BAB 2
LANDASAN TEORI

2.1 Profil tempat penelitian

2.1.1 Visi Pusat Mikroelektronika ITB

Visi Memacu pengembangan Teknologi Informasi dan Mikroelektronika melalui pembentukan ekosistem yang kreatif dan inovatif untuk kemakmuran bangsa Indonesia.

2.1.2 Misi Pusat Mikroelektronika ITB

Misi Menciptakan sinergi dengan mengembangkan program-program dalam ekosistem yang menghasilkan produk-teknologi-layanan yang mencakup lembaga pendidikan dan penelitian, industri, dan pemerintah.

2.1.3 Posisi Strategi Pusit Mikroelektronika ITB

Pusat Mikroelektronika merupakan suatu pusat unggulan yang menjadi indikator kemajuan dan posisi tawar suatu perguruan tinggi. Mengacu kepada keputusan senat akademik ITB bahwa ITB harus menuju kepada *research university* menyiratkan agar semua dosen harus memiliki kualitas dan budaya *research*. Fokus penelitian Pusat Mikroelektronika mengacu kepada *roadmap* berkelanjutan dari program kegiatan pengembangan dan penguasaan teknologi. Ditinjau dari skala sempit/laboratorium maka Pusat Mikroelektronika merupakan suatu tempat akumulasi infrastruktur *research* dan teknologi mikroelektronika. Dilihat dari posisi yang lebih tinggi secara *corporate* yaitu ITB sebagai perguruan tinggi, Pusat Mikroelektronika memiliki tiga peranan penting yaitu:

a. Sebagai *scientific accumulator*

b. Sebagai *technology accumulator*

c. Pusat solusi, produk teknologi dan produk jasa

Ditinjau dari segi peran strategis, teaching laboratorium berada pada suatu Kelompok Keakhlian (KK) yang berperan sebagai *curriculumn enforcement*, sedangkan *research laboratory* berada pada lintas UKS yang berfungsi sebagai *competency enforment* and *solution centre* untuk menuju terwujudnya ITB menjadi *the centre of excellent*. Ditinjau dari segi sumber daya manusia, di dalam

pusat Mikroelektronika terdapat tiga kelompok tenaga akhli yaitu *resarch and technology director* yang terdiri atas para pakar dari kalangan dosen, peneliti tetap yang terdiri atas tenaga peneliti tetap kontrak yang berfungsi sebagai *technology accumulator*, serta peneliti akademik yang terdiri atas mahasiswa pasca sarjana sebagai strategic resources untuk menguasai *the scientific and technology state of the art*. Dengan demikian Pusat Mikroelektronika menjadi penjamin kualitas (Quality Assurance) dilaksanakannya kontrak bisnis dengan dunia luar, serta menjadi *intellectual and technology products generator*.

Dengan demikian pusat Mikroelektronika merupakan *polecy instrument* bagi perguruan tinggi untuk menuju *research university* dengan kultur *technopreuneur*, yaitu suatu atmosphere dimana seluruh unit bernuansa flexible dalam pengembangan program dan pembiayaan rutin, sedang porsi budget atau pembiayaan dari ITB merupakan suatu *policy* instrument untuk meningkatkan posisi tawar institusi secara nasional maupun global.

2.1.4 Tujuan dan Program Pusat Mikroelektronika ITB

Tujuan dibentuknya pusat Mikroelektronika adalah untuk membentuk *techno-industrial cluster*, dimana terjadi konsentrasi dan interaksi antara lembaga riset, perguruan tinggi, dan industri, sehingga terjadi pertumbuhan daya saing ekonomi dan kemampuan teknologi. Misi program Pusat Mikroelektronika adalah:

a. Mengorientasikan kemampuan Iptek yang telah terakumulasi di Pusat penelitian danperguruan tinggi sebagai unsur pembentuk daya saing perekonomian nasional.
b. Membentuk jaringan kerjasama antara sejumlah industri, lembaga Penelitiaan dan perguruan tinggi dalam pengembangan produk dan proses produksi dengan menggunakan teknologi *state-of-the-art*.
c. Menumbuhkan kapasitas inovasi dengan memanfaatkan kemajuan Iptek, terutama dalam membentuk kemampuan manghasilkan produk-produk teknologi informasi dan mikroelektronika yang dapat bersaing di pasar internasional.

d. Membentuk kemampuan nasional untuk penguasaan perkembangan Iptek berdasarkan *technology roadmap* yang berlaku di dunia internasional, dengan memperhatikan berbagai faktor seperti HaKI, standarisasi, dan berbagai faktor teknologi lain yang telah merupakan bagian dari sistem perdagangan internasional.
e. Mendorong terbentuknya *technology-industrial cluster* yang memiliki jaringan luas, baik di dalam maupun luar negeri.

Misi tersebut diatas dilaksanakan untuk mencapai visi Program Kegiatan R&D yaitu:

a. Meningkatkan jejaring kerjasama dengan menggandeng forum-forum
    a) Menghasilkan ekosistem/*techno-industrial cluster* yang lebih efektif
b. Melakukan pembinaan dan akumulasi kepakaran
    a) Menghasilkan sumber daya manusia dengan kepakaran yang tinggi
    b) Menghasilkan sumber daya manusia berjiwa wirausaha
c. Melakukan penelitian dan pengembangan
    a) Menghasilkan sistem produk-teknologi- layanan yang inovatif
d. Melakukan inkubasi bisnis
    a) Menghasilkan *start-up company* yang tangguh berbasis teknologi
    b) Konsep Strategis yang diusung adalah:
e. Membangun critical-mass peneliti dan penelitian
f. Mengefektifkan ekosistem yang mencakup perguruan tinggi, lembaga penelitian, industry, dan pemerintah

Mengembangkan teknologi dan aplikasi di bidang Teknologi Informasi dan Mikroelektronika (TIMe).

2.1.5 Sasaran Umum dari Pusat Mikroelektronika ITB

Sasaran umum R&D pada bidang Teknologi Informasi dan Mikroelektronika mengacu kepada tolak ukur keberhasilan sebagai berikut :

a. Melakukan kegiatan R&D pada bidang mikroelektronika sebagai upaya akusisi teknologi dengan *deliverable* berupa paper dan publikasi

b. melakukan ektraksi topic-topik strategis yang layak dikembangkan menjadi produk kompetitif dengan strategi roadmapping
c. Keberhasilan pembuatan produk yang masuk pasar atau proses produksi yang dipakai oleh industri sesuai dengan spesifikasi yang ditentukan (*mass production*).
d. Jumlah paten dan makalah yang dihasilkan
e. Jumlah mitra dalam dan luar negeri yang terlibat, khususnya mitra industry
f. Jumlah pembiayaan dari mitra industri yang berpartisipasi
g. Jumlah mitra industri yang melaksanakan produk
    a) Sedangkan dalam jangka panjang sasarannya adalah
h. Keberhasilan produk yang dihasilkan memasuki pasar dalam dan luar negeri
i. Keberhasilan menjalankan serta mengefektifkan *techno-industrial cluster*
j. Kemajuan penguasaan teknologi *roadmap*

Pada tingkat nasional bidang Teknologi Informasi dan Mikroelektronika (TIMe) menjadi salah satu unggulan pembangunan bidang sosial ekonomi Indonesia yang diharapkan meningkatkan penghasilan devisa dan penciptaan lapangan pekerjaan tingkat rendah, menengah, dan tinggi.

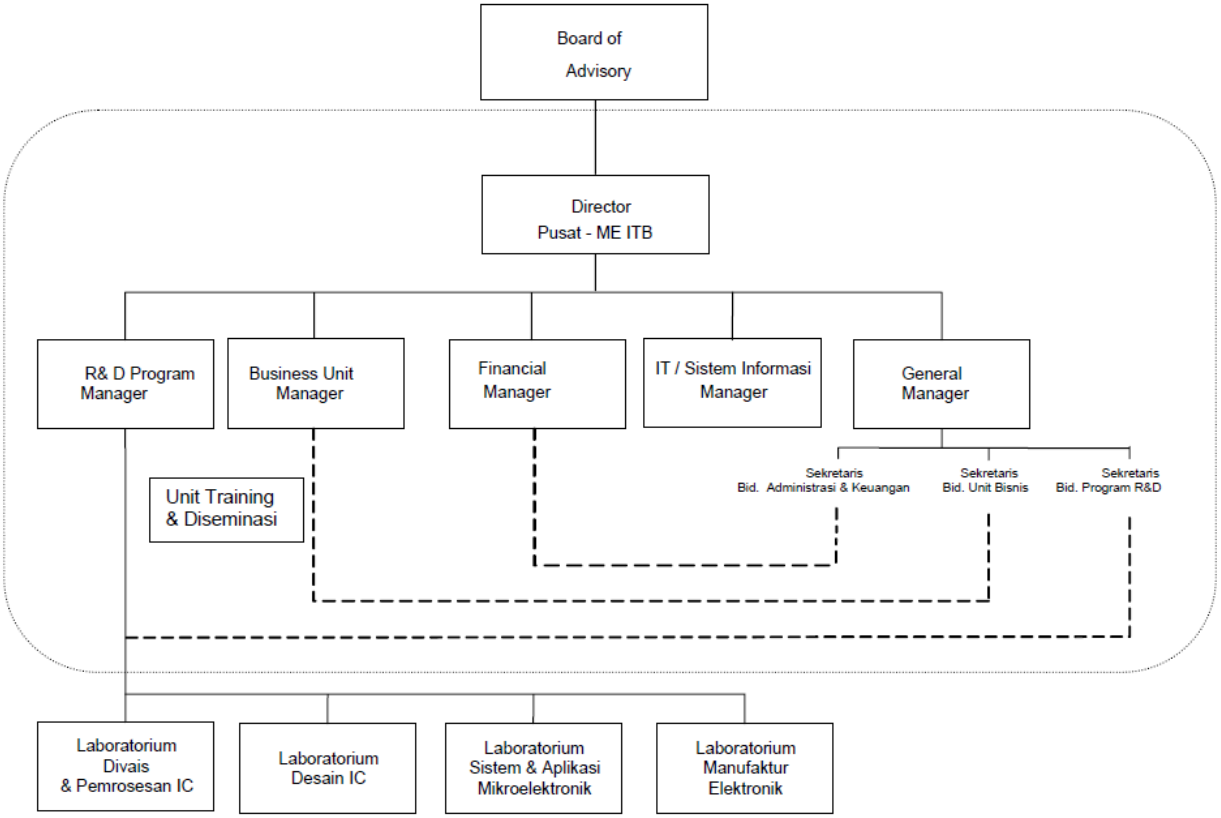
2.1.6 Program dan Aktivitas Laboratorium

Saat ini Pusat Mikroelektronika fokus kegiatan pengembangan dan penelitian di Program Sensor & Mikrosistem, Desain IC & Authentication, Elektronika Industri, Internet economy, Wireless Multimedia Internet, dan Manufaktur Elektronika.

Aktivitas Pusat Mikroelektronika terdistribusi ke dalam Laboratorium-laboratorium yakni diantaranya:

a. Lab Sistem dan Aplikasi, menggunakan IC untuk mengembangkan produk dan aplikasi. Dalam banyak kasus, mikroelektronik devices, produk dan sistem diproduksi secara massal.

b. Lab Desain, mendevelop technology untuk mendesain IC dengan menggunakan Computeraided design (CAD) dan karakterisasi dari komponen dan devices.
c. Lab Devices dan Proses, memperoleh dan mendevelop teknologi untuk analisis, karakterisasi, desain, *developing*, dan pemrosesan komponen dan *devices* mikroelektronika yang akan digunakan dalam IC.
d. Lab Manufaktur, mempelajari dan men-*develop* teknologi untuk manufaktur produk elektronika.



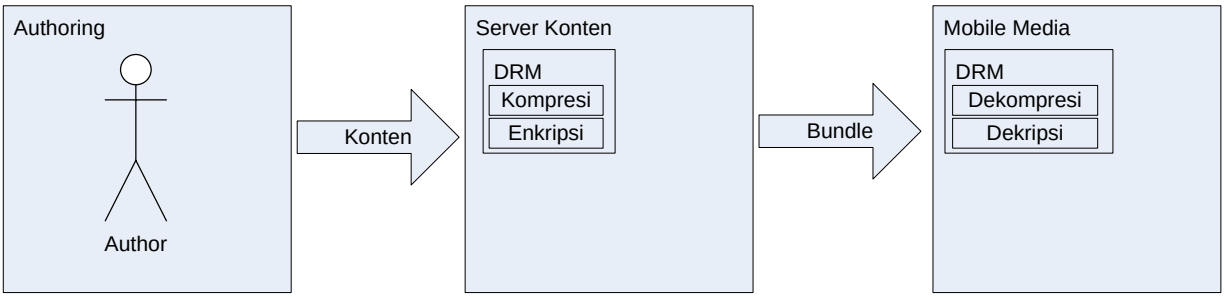
Gambar 2-1 Strukturk Organisasi Fungsional

2.2 Digital Publishing

Penerbitan (*Publishing*) merupakan usaha untuk memperoleh, mempersiapkan, atau menciptakan dan menjual produk informasi secara tercetak. Industri penerbitan adalah salah satu industri yang tidak pernah padam, karena

kebutuhan akan suatu produk informasi masih tinggi hingga saat ini. Dengan muncul dan berkembangnya sistem informasi digital dan internet membuat ruang lingkup dari penerbitan meluas dengan mencakup sumber daya elektronik seperti versi elektronik dari buku, koran, dan majalah, yang kemudian di sebut dengan *digital publishing* (penerbitan digital).

Digital publishing merupakan publikasi digital dari e-books, digital magazine, dan pengembangan dari perpustakaan digital dan katalog. Digital publishing saat ini sudah menjadi hal umum dalam proses penerbitan ilmiah di mana telah diusulkan bahwa peer-review jurnal ilmiah dalam proses digantikan oleh penerbitan elektronik. Selain itu digital publishing menjadi hal yang umum dalam hal pendistribusian buku, majalah dan koran ke konsumen melalui gadget (tablet, smartphone), karena pasar yang berkembang oleh jutaan setiap tahun, yang dihasilkan oleh vendor online seperti apple itunes bookstore, toko buku amazon untuk kindle, dan buku di android market. Riset pasar menunjukkan bahwa setengah dari semua majalah dan sirkulasi surat kabar akan melalui pengiriman digital pada akhir 2013 dan bahwa setengah dari semua bacaan akan dilakukan tanpa kertas pada tahun 2015[3].



Gambar 2-2 Sistem Digital Publishing

2.3 Digital Right Management

Pada gambar 2-2 terlihat bahwa salah satu komponen dalam sistem digital publishing adalah digital right management, dimana dengan komponen tersebut konten-konten yang didistribusikan lebih terproteksi, sehingga lebih aman dari tindak pembajakan.

DRM atau digital right management sendiri adalah teknologi yang memungkinkan penyedia konten digital untuk mendistribusikan, mempromosikan dan menjual konten digital dalam cara yang aman. Teknologi DRM memungkinkan distributor konten untuk mengontrol penggunaan konten digital (seperti musik, wallpaper, majalah digital, dll) dengan cara membatasi akses untuk akses, menyalin atau konversi ke format lain oleh end-user[4].

Proteksi konten oleh DRM dicapai dengan mendefinisikan hak penggunaan yang berbeda untuk setiap konten yang menerapkan DRM. DRM menciptakan landasan yang penting berupa kepercayaan antara author dan konsumen yang merupakan prasyarat untuk pengembangan pasar yang lebih kuat. Selain itu DRM dapat membantu melindungi kepentingan pemilik konten, karena DRM tidak hanya berguna untuk melindungi konten, tetapi juga menjual dan mendistribusikan konten.

2.4 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (dalam bahasa inggris: object-oriented programming, biasa disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek, jadi semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek.

Berbeda dengan pemrograman terstruktur pemrograman berorientasi objek setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Dalam OOP sistem dibangun dengan membuat objek objek dan menginteraksikannya untuk menyelesaikan suatu pekerjaan.

OOP berputar pada konsep dari obyek yang merupakan elemen dasar dari program. Ilustrasinya adalah : Objek mobil mempunyai atribut tipe transmisi, warna dan manufaktur, mobil juga mempunyai tingkah laku berbelok, mengerem dan berakselerasi. Pada perangkat lunak, menggunakan atribut sebagai data dan tingkah laku sebagai method.

Pemrograman berorientasi objek memliki beberapa keuntungan seperti :

a. Maintenance, program lebih mudah dibaca dan Maintenance, program lebih mudah dibaca dan dipahami, dan pemrograman berorientasi obyek

mengontrol kerumitan program hanya dengan mengijinkan rincian yang dibutuhkan untuk programmer.

b. Pengubahan program (berupa penambahan atau penghapusan fitur tertentu) disesuaikan dengan keperluan berdasarkan objeknya.

Pada paradigma OOP tidak lepas dari istilah *class*. *Class* sendiri merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsiyang menunjuk pada berbagai macam perilaku/turunan dari anjing. Class suatu struktur dalam program yang berisi satu atau beberapa metode (dapat berupa fungsi dan/atau prosedur) dan atribut (dapat berupa data ataupun variabel yang akan dipoerasikan dalam metode). Dengan demikian atribut merupakan anggota kelas yang bersifat statis, sedangkan metode merupakan anggota kelas yang bersifat dinamis.

Selain *class* terdapat juga istilah objek pada paradigma OOP. Obyek adalah sebuah instance (keturunan) dari class, dimana Setiap obyek dibangun dari sekumpulan data (atribut) yang disebut variabel untuk menjabarkan karakteristik khusus dari obyek, dan juga terdiri dari sekumpulan method yang menjabarkan tingkah laku dari obyek. Bisa dikatakan bahwa obyek adalah sebuah perangkat lunak yang berisi sekumpulan variabel dan method yg berhubungan

2.5 Pemrograman Terstruktur

Pemrograman Terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis, dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami.

Berbeda dengan OOP, pemrograman terstruktur adalah sebuah cara pemrosesan data yang terstruktur (sistematis) dalam analisa, cara dan penulisan pemrograman. Selain itu dibandingkan dengan OOP Pemrograman terstruktur unggul dalam melakukan pemrograman sederhana karena lebih efisien dan lebih mudah dalam hal perawatannya.

Dalam pemrograman yang terstruktur, pendekatan pembuatan program adalah dengan menganut konsep "top-down". pada konsep ini, program dimulai

dengan gambaran global, yang dinyatakan dengan nama-prosedur (sub-rutin) dan bukan isi detailnya. Selanjutnya prosedur sendiri bisa dipecah-pecah lagi menjadi prosedur yang lain. Konsep ini sangat memudahkan dalam pemodifian program.

2.6 Teori Pengujian

Ada dua macam pengujian perangkat lunak yang dapat dilakukan, yaitu metode *white box* dan *blackbox. White box testing* dilakukan di awal program, sedangkan *black box testing* baru dilakukan pada tahap berikutnya. Pengujian dilakukan untuk melakukan penilaian tentang keberhasilan program dan menentukan apakah prgram sudah sesuai dengan keinginan *user* atau belum.

2.6.1 Pengujian *White Box*

Pengujian *White-box* yaitu metode desain *test case* yang menggunakan stuktur kontrol desain posedural unntuk memperoleh *test case. Test case* dapat diperoleh dengan:

a. Menjamin bahwa semua *independet path* pada suatu modul telah digunakan minimal satu kali
b. Menggunakan semua keputusan logis pada sisi *true* dan *false.*
c. Mengeksekusi semua *loop* dalam batasannya dan pada batas operasionalnya
d. Mengunakan struktur data internal untuk menjamin validitasnya

Pengujian ini harus memenuhi kriteria sebagai berikut :

a. Mengurangi pelaksanaan *test case* untuk mencapai hasil pengujian yang diinginkan
b. *Test case* akan menunjukkan ketidaksesuaian (ketidaksinkronan) beberapa kesalahan, tetapi kurang menunjukkan detil kesalahan

Software yang dibangun tidak selalu sempurna. Terkadang ada kesalahan yang baru akan diketahui saat pengujian atau bahkan saat implementasi. Berikut ini kriteria *software* yang "cacat" :

a. Kesalahan logika dan asumsi yang salah berbanding terbalik dengan probabilitas jalur program yang akan dieksekusi. Kecenderungan kesalahan ini terjadi pada tahap desain dan implementasi fungsi, kondisi

atau kontrol yang berada di luar pikiran. Kesalahan ini tejadi jika pemrosesan yang rutin sudah dikerjakan dengan baik tetapi pemrosesan yang khusus cenderung diabaikan.

b. Selalu ada keyakinan bahwa *logical path* tidak akan dieksekusi pada basis regular. Kesalahan ini terjadi karena adanya kesalahan asumsi tentang aliran data dan kontrol.
c. Kesalahan tipologis yang merupakan kesalahan yang acak atau random. Perpindahan dari bahasa pemrograman satu ke bahasa pemrograman lain menyebabkan timbulnya kesalahan sintak.

Path Testing adalah metode pengujian yang memungkinkan desainer *test case* mengukur kompleksitas logika desain posedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi. Metode ini menjamin bahwa setiap statement akan dilalui minimal sekali dalam proses pengujian. Tujuannya adalah meyakinkan bahwa himpunan *test case* akan menguji setiap *path* pada suatu program paling sedikit satu kali. Titik awal *path testing* adalah suatu program *flow graph* yang menunjukkan node – node yang menyatakan keputusan program, misalnya : kondisi if – then – else dan busur yang menyatakan alur kontrol. Statement yang memiliki kondisi ada pada node – node dalam *flow graph*. *Path testing* menggambarkan alur kontrol di mana setiap cabang ditunjukkan oleh path yang terpisah sedangkan loop ditunjukkan oleh arrows looping dan kembali ke loop kondisi node.

2.6.2 Pengujian *Black Box*

Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangk at lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Black-box testing berfokus pada persyaratan fungsional perangkat lunak. Metode ini memungkinkan *software developer* untuk mendapatkan serangkaian

kondisi input yang mempergunakan semua persyaratan fungsional program. Black-Box testing bukan alternatif *white-box testing*, namun merupakan pelengkap yang mampu mengungkap kesalahan, jika dibandingkan metode *white-box testing.*

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

2.7 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa standar yang bekerja dalam *object-oriented* untuk menentukan, memvisualisasikan, merancang, dan mendokumentasikan elemen-elemen informasi yang terdapat dalam sistem *software*. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh grady Booch, Jim Rumbaugh dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem[6].



Gambar 2-3 Logo UML

2.7.1 Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

a. *Use Case Diagram* untuk memodelkan proses bisnis

b. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi

c. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*

d. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*

e. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem

f. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *Objects* di dalam sistem

g. *Class Diagram* untuk memodelkan struktur kelas

h. *Object Diagram* untuk memodelkan struktur object

i. *Component Diagram untuk memodelkan komponen object*

j. *Deployment Diagram* untuk memodelkan distribusi aplikasi

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu :

a. *Use Case* diagram, untuk memodelkan proses bisnis dan merepresentasikan sebuah interaksi antara aktor dengan sistem

b. *Sequence* diagram, untuk menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok yakni nama (dan stereotype) , atribut , dan metoda .

c. *Statechart diagram* , untuk memodelkan perilaku objects di dalam sistem atau menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem. Pada umumnya statechart diagram menggambarkan class tertentu .

d. *Activity diagram* , untuk memodelkan perilaku *Use Cases* dan *objects* di dalam system.

e. *Sequence diagram* , untuk memodelkan pengiriman pesan (*message*) antar objects dan juga digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

f. *Collaboration diagram* , untuk memodelkan interaksi antar objects seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*

g. *Component diagram* , untuk memodelkan komponen object atau menggambarkan struktur dan hubungan antar modul yang berisi code, termasuk ketergantungan (*dependency*) di antaranya.

h. *Deployment diagram*, untuk memodelkan distribusi aplikasi.

2.8 Data Flow Diagram

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut[7].

Kita dapat menggunakan DFD untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru.

Empat simbol yang digunakan :

Ada 3 (tiga) jenis DFD, yaitu ;

1. Context Diagram (CD)
2. DFD Fisik
3. DFD Logis

DFD Level

DFD dapat digambarkan dalam Diagram Context dan Level n. Huruf n dapat menggambarkan level dan proses di setiap lingkaran.

a. Diagram Context
b. Diagram Level n
    a) DFD Logis
    b) DFD Fisik

Context Diagram (CD)

Jenis pertama Context Diagram, adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD;

1. Terminologi sistem :
    a. *Batas Sistem* adalah batas antara “daerah kepentingan sistem”.
    b. *Lingkungan Sistem* adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.
    c. *Interface* adalah aliran yang menghubungkan sebuah sistem dengan linkungan sistem tersebut.
2. Menggunakan satu simbol proses,
3. Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut,

4. Antara Entitas Eksternal/Terminator tidak diperbolehkan komunikasi langsung
5. Jika terdapat termintor yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda asterik (*) atau garis silang (#).
6. Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipermainkan personil tersebut.
7. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

Diagram Level *n* / Data Flow Diagram Levelled

Dalam diagram n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram diagram logis. Dimana Diagram Level *n* merupakan hasil pengembangan dari *Context Diagram* ke dalam komponen yang lebih detail tersebut disebut dengan top-down partitioning. Jika kita melakukan pengembangan dengan benar, kita akan mendapatkan DFD-DFD yang seimbang. Sebagai contoh, gambar 1.1, gambar 1.2, gambar 1.3, gambar 1.4 dan gambar 1.5. Beberapa hal yang harus diperhatikan dalam membuat DFD ialah:

1. Pemberian Nomor pada diagram level *n* dengan ketentuan sebagai berikut:
    a. Setiap penurunan ke level yang lebih rendah harus mampu merepresentasikan proses tersebut dalam sepesifikasi proses yang jelas. Sehingga seandainya belum cukup jelas maka seharusnya diturunkan ke level yang lebih rendah.
    b. Setiap penurunan harus dilakukan hanya jika perlu.
    c. Tidak semua bagian dari sistem harus diturunkan dengan jumlah level yang sama karena yang kompleks bisa saja diturunkan, dan yang sederhana mungkin tidak perlu diturunkan. Selain itu, karena tidak semua proses dalam level yang sama punya derajat kompleksitas yang sama juga.
    d. Konfirmasikan DFD yang telah dibuat pada pemakai dengan cara top-down.

e. Aliran data yang masuk dan keluar pada suatu proses di level n harus berhubungan dengan aliran data yang masuk dan keluar pada level n+1. Dimana level n+1 tersebut mendefinisikan sub-proses pada level n tersebut.
f. Penyimpanan yang muncul pada level n harus didefinisikan kembali pada level n+1, sedangkan penyimpanan yang muncul pada level n tidak harus muncul pada level n-1 karena penyimpanan tersebut bersifat lokal.
g. Ketika mulai menurunkan DFD dari level tertinggi, cobalah untuk mengidentifikasi external events dimana sistem harus memberikan respon. External events dalam hal ini berarti suatu kejadian yang berkaitan dengan pengolahan data di luar sistem, dan menyebabkan sistem kita memberikan respon.

2. Jangan menghubungkan langsung antara satu penyimpanan dengan penyimpanan lainnya (harus melalui proses).
3. Jangan menghubungkan langsung dengan tempat penyimpanan data dengan entitas eksternal / terminator (harus melalui proses), atau sebaliknya.
4. Jangan membuat suatu proses menerima input tetapi tidak pernah mengeluarkan output yang disebut dengan istilah “black hole”.
5. Jangan membuat suatu tempat penyimpanan menerima input tetapi tidak pernah digunakan untuk proses.
6. Jangan membuat suatu hasil proses yang lengkap dengan data yang terbatas yang disebut dengan istilah “magic process”.
7. Jika terdapat terminator yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda asterik (*) atau garis silang (#), begitu dengan bentuk penyimpanan.
8. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

DFD Fisik

Adalah representasi grafik dari sebuah sistem yang menunjukan entitas-entitas internal dan eksternal dari sistem tersebut, dan aliran-aliran data ke dalam

dan keluar dari entitas-entitas tersebut. Entitas-entitas internal adalah personel, tempat (sebuah bagian), atau mesin (misalnya, sebuah komputer) dalam sistem tersebut yang mentransformasikan data. Maka DFD fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan. (Tidak Bahas).

Perlu diperhatikan didalam memberikan keterangan di lingkaran-lingkaran (simbol proses) dan aliran-aliran data (simbol aliran data) dalam DFD fisik menggunakan label/keterangan dari kata benda untuk menunjukan bagaimana sistem mentransmisikan data antara lingkaran-lingkaran tersebut.

Misal :

Aliran Data : Kas, Formulir 66W, Slip Setoran

Proses : Cleck Penjualan, Kasir, Pembukuan, dll.

DFD Logis

Adalah representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan DFD logis untuk membuat dokumentasi sebuah sistem informasi karena DFD logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan.

Keuntungan dari DFD logis dibandingkan dengan DFD fisik adalah dapat memusatkan perhatian pada fungsi-funsi yang dilakukan sistem.

Perlu diperhatikan di dalam pemberian Keterangan/ Label;

a. Lingkaran-lingkaran (simbol proses) menjelaskan apa yang dilakukan sistem
   Misal : Menerima Pembayaran, Mencatat Penjualan, Membandingkan kas dan Daftar Penerimaan, Mempersiapkan Setoran, dll.
b. Aliran-aliran data (simbol aliran data) menggambarkan sifat data.
   Misal : Pembayaran (bukan “Cek”, “Kas”, “ Kartu Kredit”
   Jurnal Penjualan (bukan “Buku Penjualan”), dll

Usulan dari analis (berupa DFD dalam bab 4), beberapa hal yang umum yang mendapat perhatian dalam mendesain baru tersebut ialah:

a. Menggabungkan beberapa tugas menjadi Satu
b. Master Detail Update
c. Meminimalkan tugas-tugas yang tidak penting
d. Menghilangkan tugas-tugas yang duplikat
e. Menambahkan proses baru
f. Meminimalkan proses input

Menetapkan bagian mana yang harus dikerjakan komputer dan bagian mana yang harus dikerjakan manual.

2.9 Entity Relationship Diagram

Basisdata Relasional adalah kumpulan dari relasi-relasi yang mengandung seluruh informasi berkenaan suatu entitas/ objek yang akan disimpan di dalam database. Tiap relasi disimpan sebagai sebuah file tersendiri. Perancangan basisdata merupakan suatu kegiaatan yang setidaknya bertujuan sebagai berikut:

a. Menghilangkan redundansi data
b. Meminimumkan jumlah relasi di dalam basis data
c. Membuat relasi berada dalam bentuk normal, sehingga dapat meminimumkan permasalahan berkenaan dengan penambahan, pembaharuan dan penghapusan.

ERD adalah suatu pemodelan dari basisdata relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut entity dan hubungan yang dimilikinya disebut relationship. Suatu entity bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya. Contoh : entity Mahasiswa, mempunyai atribut nama, umur, alamat, dan nim. Diagram E-R terdiri dari:

a. Kotak persegi panjang, menggambarkan himpunan entitas
b. Elip, menggambarkan atribut-atribut entitas
c. Diamon, menggambarkan hubungan antara himpunan entitas
d. Garis, yang menghubungkan antar objek dalam diagram E-R

E-R Diagram merupakan suatu bahasa pemodelan yang dimana posisinya dapat dianalogikan dengan stroy board dalam industri film, blu print arsitektur suatu bangunan, miniatur, dan lain-lain. Dalam praktiknya, membangun suatu sistem terlebih dahulu dilakukannya suatu perencaan. Pemodelan merupakan suatu sub bagian dari perencanaan secara keseluruhan sebagai salah satu upaya feedback evaluasi perampungan suatu perencanaan. E-R Diagram sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

a. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat.
b. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan developer.
c. Menghasilkan dokumentasi yang baik untuk client sebagai bahan diskusi dengan bentuk E-R Diagram itu sendiri, dan
d. Kamus data bagi bagi para pengembang database.

Struktur dari E-R Diagram secara umum ialah terdiri dari:

a. Entitas merupakan objek utama yang informasi akan disimpan, biasanya berupa kata benda, ex; Mahasiswa, Dosen, Nasabah, Mata Kuliah, Ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.
b. Atribut merupakan deskripsi dari objek yang bersangkutan.

Relationship merupakan suatu hubungan yang terjalin antara dua entitas yang ada.

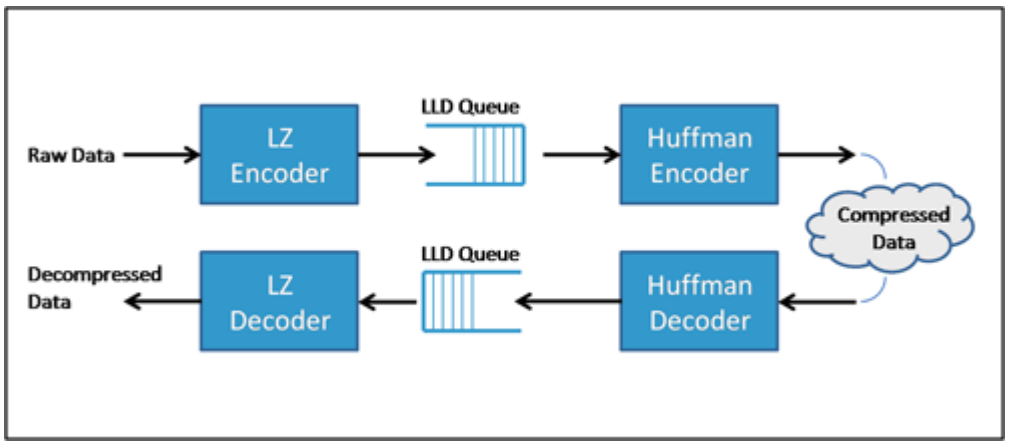
2.10 DEFLATE

Kompresi DEFLATE yang merupakan kompresi data lossless yang menggunakan kombinasi dari teknik kompresi LZ77 dan Huffman coding. Efisiensi kompresi DEFLATE ini adalah variabel berdasarkan aliran data terkompresi, dan tolak ukur umum menggunakan file berbasis teks menunjukkan rasio kompresi dari 2,5:1 sampai kisaran 3,5:1. Kompresi sebesar ini memiliki keunggulan operasional yang jelas di penyimpanan data dan data jaringan.

Dalam media penyimpanan, pengurangan ukurang file yang tersimpan berbanding lurus dengan pengurangan kapasitas media yang diperlukan untuk

menyimpan file. Hal ini sangat berguna dalam sistem penyimpanan atau backup data seperti tape backup, VTL (virtual type library), disk system backups, flash-based disk drives, dan sistem penyimpanan de-duplikasi.

Dalam jaringan data, pengurangan ukuran file yang ditransimisikan berimbas pada penurunan bandwidth jaringan yang diperlukan untuk mengirimkan file. Hal ini berguna dalam server web, load balancing, optimasi gateway WAN, bahkan router dimana kompresi HTTP digunakan. Dengan menerapkan suatu kompresi pada proses pertukaran data, suatu sistem bisa mendapatkan keuntungan besar, dimana dengan penggunaan bandwidth yang rendah dapat memaksimalkan layanan yang diberikan.



Gambar 2-4 Gambaran proses deflate

Data mentah masuk ke encoder LZ dimana pencocokan string dilakukan dengan menggunakan algoritma hash, tabel hash, dan history buffer. Output dari encoder LZ adalah antrian yang disebut antrian LLD (literals and long / distance), yang menyimpan symbol-simbol yang dihasilkan selama tahap pengkodean LZ. Output dari antrian LLD diproses oleh encoder Huffman yang menghasilkan aliran bit terkompresi.

Pada proses dekompresi, data yang terkompresi diterjemahkan pada decoder Huffman untuk membangun aliran simbol yang dibutuhkan oleh decoder LZ. Decoder LZ beroperasi langsung mematikan aliran simbol tersebut untuk merekonstruksi data asli, dan tidak memerlukan hashing atau tabel hash.

Kompresi menggunakan deflate dilakukan secara blok demi blok. Blok terkompresi dapat memiliki panjang yang berubah-ubah sampai dengan 64KB.

2.11 AES

AES atau Advanced Encryption Standard merupakan spesifikasi untuk enkripsi data elektronik yang dibentuk oleh National Institute of Standards and Technology (NIST) pada tahun 2001, dimana standar ini merupakan varian dari algoritma Rijndael, yang merupakan blok cipher simetris yang dapat mengolah data blok 128 bit, menggunakan kunci cipher dengan panjang 128, 192, dan 256 bit.

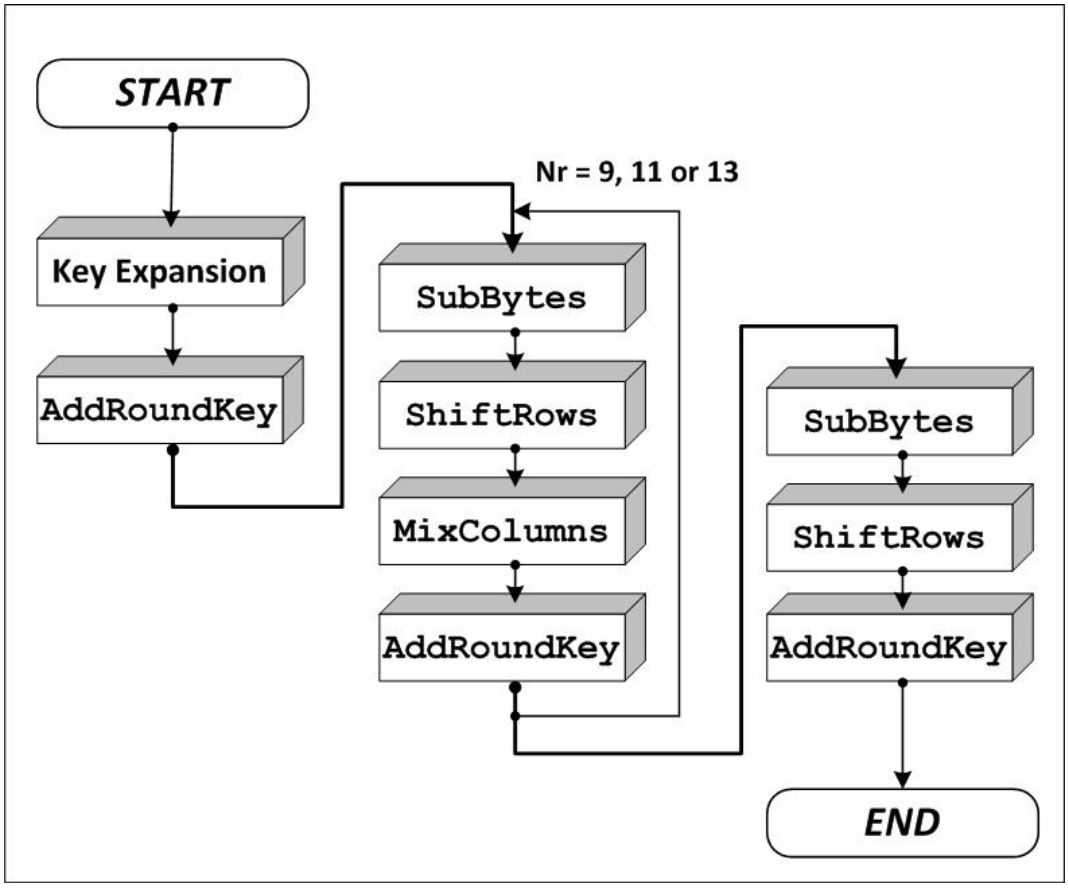
AES didasarkan pada prinsip desain yang dikenal sebagai jaringan substitusi-permutasi, dan penggunaannya cepat dalam perangkat keras maupun perangkat lunak. Tidak seperti pendahulunya DES, AES tidak menggunakan jaringan Feistel. Ukuran utama yang digunakan AES untuk menentukan jumlah pengulangan putaran transformasi yang mengkonversi input menjadi output adalah sebagai berikut :

a. 10 siklus pengulangan untuk kunci 128-bit
b. 12 siklus pengulangan untuk kunci 192-bit
c. 14 siklus pengulangan untuk kunci 256-bit.

Blok cipher AES beroperasi pada 4x4 array byte (128bit), dimana merupakan ukuran dari blok masukan, blok keluaran dan state. Hal ini diwakili oleh nilai Nb=4, yang mencerminkan jumlah kata 32-bit (jumlah kolom) dalam array state. Panjang yang diperbolehkan dari key cipher adalah 128, 192, dan 256 bit. Panjang kunci diwakili oleh Nk=4,6,8 yang mencerminkan jumlah kata 32-bit dalam array key cipher.

State dienkripsi atau didekripsi dengan menerapkan transformasi byte berorientasi untuk jumlah dan putaran tertentu. Jumlah puataran yang akan dilakukan tergantung pada ukuran kunci. Jumlah putaran diwakili oleh Nr. Algoritma AES menentukan cipher dan kebalikannya untuk kelengkapan siklus enkripsi dan dekripsinya. Forward cipher mengambil teks biasa sebagai masukan bersama dengan cipher-key dan outputnya adalah data dienkripsi atau ciphertext. Inverse cipher mengambil ciphertext sebagai masukan dan mendekripsi kembali ke plaintext yang sama menggunakan cipher key yang digunakan untuk enkripsi. Algoritma AES terdiri dari tahap berikut:

1. Key Expansion = Round key berasal dari kunci cipher menggunakan Rijndael.
2. Initial Round – AddRoundKey = masing-masih byte awal state dikombinasikan dengan Round Key menggunakan operasi bit-wise
3. Middle Round- Nr = r sampai Nr = r-1 = Operasi pengulangan yang melakukan transformasi berikut :
    a. SubBytes = Langkah substitusi non-linear dimana setiap byte diganti dengan yang lain sesuai dengan table lookup.
    b. ShiftRows = Langkah transposisi dimana setiap baris state digeser sesuai jumlah siklus langkah.
    c. MixColumn = Operasi pencampuran yang beroperasi pada kolom state, menggabungkan empat byte dalam setiap kolom’
    d. AddRoundKey = masing-masih byte awal state dikombinasikan dengan Round Key menggunakan operasi bit-wise
4. Final Round
    a. SubByte = Langkah substitusi non-linear dimana setiap byte diganti dengan yang lain sesuai dengan table lookup.
    b. ShiftRows = Langkah transposisi dimana setiap baris state digeser sesuai jumlah siklus langkah.
    c. AddRoundKey = masing-masih byte awal state dikombinasikan dengan Round Key menggunakan operasi bit-wise



Gambar 2-5 Gambaran proses AES

2.12 HTML5

HTML5 adalah revisi kelima dari HTML (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997 dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan standar baru untuk XHTML, HTML dan DOM HTML. Sejak munculnya HTML versi 4.01, perkembangan website dunia semakin berkembang. Memang saat ini HTML5 masih dalam masa pengembangan, dan hanya ada beberapa browse sudah mendukung HTML5.

HTML5 merupakan hasil proyek dari W3C (World Wide Web Consortium) dan WHATWG (Web Hypertext Application Technology Working Group). Dimana

WHATWG bekerja dengan bentuk web dan aplikasi dan W3C merupakan pengembang dari XHTML 2.0 pada tahun 2006, kemudian mereka memutuskan untuk bekerja sama dan membentuk versi baru dari HTML.

Adapun kelebihan yang paling menonjol dari HTML 5 adalah kemudahan akses yang lebih baik. Tag Baru seperti header, footer, nav, section, aside dll memungkinkan browser untuk mengakses konten dengan mudah. Sebelumnya, kita hanya menentukan hal tersebut dengan tag dan menggunakan atribut id ataupun class. Dengan tag html5 yang baru, browser mampu menjelajah dokumen HTML dengan lebih baik lagi. Selain itu dukungan yang lebih baik terhadap video dan audio. Beberapa Peramban yang sudah mendukung HTML 5 yaitu Safari, Chrome, Firefox, dan Opera.

Kelebihan lain yang dimiliki HTML 5 antara lain:

1. Dukungan yang lebih baik untuk penyimpanan secara offline
2. Unsur kanvas untuk menggambar
3. Video dan elemen audio untuk media pemutaran file multimedia
4. Elemen konten yang lebih spesifik, seperti artikel, footer, header, nav, section
5. Bentuk kontrol form seperti kalender, tanggal, waktu, email, url, search.

2.13 Android

Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC. Android juga merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan *key-application* yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API (*Application Programming* Interface) diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

2.13.1 Sejarah Android

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Keunggulan utama

Android adalah gratis dan *open source*. Keuntungan open source, banyak 14 pengembang software yang bisa melihat dan memanfaatkan kode itu serta bisa membuat aplikasi baru di dalamnya. Berbagai aplikasi android diwadahi dalam sebuah portal, yaitu Android Market, sehingga pengguna tinggal meng-*install* aplikasi pilihannya.

a. Produk Awal

September 2007 Google mengajukan hak paten aplikasi telepon seluler.Android yang digunakan pada perangkat bergerak (mobile) merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis dilakukan berbagai pembaharuan berupa perbaikan bug dan penambahan fitur baru.

b. Android Versi 1.1

Dirilis oleh Google pada bulan Maret 2009. Android versi ini di lengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan *email*.

c. *Android Cupcake (*1.5)

Hanya selang 1 bulan Google merilis Android 1.5 **(***Cupcake***)** dengan SDK (*Software Development Kit*). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler yakni kemampuan merekam dan menonton video dengan modus kamera, menggugah 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 disesuai dengan system.

d. *Android Donut (*1.6)

Versi ini dirilis pada September dengan menampilkan proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol *applet* VPN (*Virtual Private* 15 *Network*) atau jaringan pribadi. Fitur lainnya adalah galeri yang memungkinkan

pengguna untuk memilih foto yang akan dihapus, kamera, camcorder, dan galeri yang diintegrasikan : CDMA/EVDO, 802.1x, VPN, Gestures dan Text to speech engine, kemampuan dial kontak, teknologi text to change speech (tidak tersedia dalam ponsel), pengadaan resolusi VWGA.

e. Android Eclair (2.0 / 2.1)

Versi ini dirilis 3 desember 2009 yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan user interface dan merupakan versi Android yang pertama kali mendukung format HTML5. Perubahan yang lain yaitu : pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru , daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth* 2.1.

f. Android Froyo / *Frozen Yoghurt* (2.2)

Versi ini dirilis 20 mei 2010, android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan kinerja, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash 10.1, aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 Javascript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD card, kemampuan WiFi *Hotspot portable*, dan kemampuan *auto update* dalam aplikasi Android *Market*.

g. *Android Gingerbread* (2.3)

Versi ini dirilis 6 desember 2010. Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur *soft keyboard* & *copy/paste, power management*, dan support *Near Field Communication*, peningkatan kemampuan permainan (*gaming*), layanan antarmuka (*User Interface*) didesain ulang, dukungan format 16 video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization* dan *bass boost*), dan dukungan jumlah kamera yang lebih dari satu.

h. *Android Honeycomb* (3.0, 3.1 dan 3.2)

Merupakan versi Android yang ditujukan untuk gadget / device dengan layar besar seperti Tablet PC; Fitur baru Honeycomb yaitu dukungan terhadap prosessor multicore dan grafis dengan hardware acceleration.

i. *Android Ice Cream Sandwich* (4.0)

Android 4.0 Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang *Google* 26 *I/O Developer Conference* (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. Android versi 4.0 ini dapat digunakan di smartphone ataupun tablet. Fitur utama yang ditambahkan di Android 4.0 ialah *Face Unlock*, *Android Beam*, perubahan major *User Interface*, dan ukuran layar standar (*native screen*) beresolusi 720p *(high definition)*, perangkat tambahan fotografi, mencari email secara offline dan berbagai informasi dengan menggunakan NFC.

j. *Jelly Bean* (4.1 dan 4.2)

Google memperkenalkan android 4.1 pada 27 juni 2013, sedangkan jelly bean versi 4.2 pada 29 oktober 2012. Jelly bean merupakan versi dengan incremental update yang memiliki tujuan untuk meningkatkan funsionalitas dan performansi dari user interface. Peningkatan performa ini disebut “project butter”, yang mana menerapkan touch anticipation, triple buffering, extended vsync timing dan fixed frame rate pada 60fps untuk menciptakan user interface yang lebih halus dan lancer.

2.13.2 Fitur-fitur android

Berikut ini adalah fitu-fitur yang ada pada Android:

a. **Framework Aplikasi** yang mendukung penggantian komponen dan *reusable*

b. **Mesin virtual Dalvik** dioptimalkan untuk perangkat *mobile*

c. **Integrated browser** berdasarkan *engine open source WebKit*

d. **Grafis** yang dioptimalkan dan didukung oleh *library* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).

e. **SQLite** untuk penyimpanan data

f. **Media Support** yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

g. **GSM Telephony** (tergantung hardware)

h. **Bluetooth, EDGE, 3G, dan WiFi** (tergantung hardware)

i. **Kamera, GPS, kompas, dan accelerometer** (tergantung hardware)

j. **Lingkungan Development** yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk Eclipse IDE.

k. **Dukungna Perangkat Tambahan** : Android dapat memanfaatkan kamera, layar sentuh, *accelerometers*, *magnetometers*, GPS, akselerasi 2D (dengan perangkat orientasi, scalling,konversi format piksel), dan akselerasi grafis 3D.

l. ***Multi-Touch*** : kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

m. **Market** : handphone memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di *download* dan di-*install* melalui internet.

2.13.3 Open Handset Alliance

Dibentuklah aliansi bisnis yang bernama *Open Handset Alliance* (5 November 2007), yang dikepalai/dipimpin oleh Google, terdiri dari 78 perusahaan, dimana didalamnya terlibat banyak perusahaan-perusahaan besar. Tujuan dibentuknya OHA yaitu untuk bersaing dengan platform perangkat ponsel lainnya, seperti yang dikembangkan oleh Apple, Microsoft, Nokia, Palm, Research In Motion, Symbian, and bada.

Trend Saat ini trend penggunaan ponsel Android semakin meningkat, terbukti pada tahun 2010 penjualan ponsel dengan OS Android di US telah mencapai peringkat pertama (33%), kemudian disusul oleh ponsel dengan OS BlackBerry (28%), dan yang ketiga oleh ponsel dengan iOS (22%).

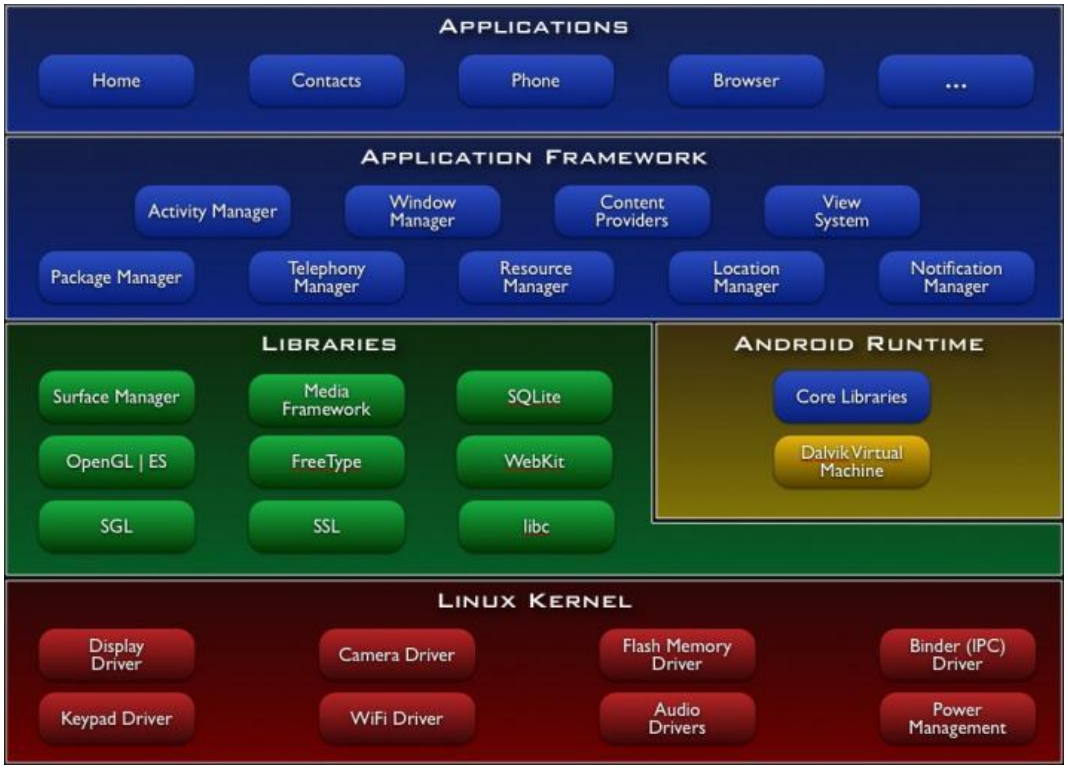
Developer (*Pengembang*) Bahasa yang digunakan untuk membuat aplikasi Android adalah Java. Oleh karena itu jangan heran jika Android memiliki komunitas pengembang aplikasi yang sangat luas dan besar. Tentunya dengan besarnya komunitas ini fungsionalitas dari perangkat yang menggunakan OS Android juga meningkat. Saat ini telah terdapat **70,000** aplikasi Android yang

tersedia, dan menjadikan Android sebagai *environment* pengembangan aplikasi *mobile* **terpopuler** kedua saat ini.

Lisensi Sebagian besar source code yang dirilis oleh Android berada dalam lisensi Apache, serta lisensi free software dan open source.

2.13.4 Arsitektur Android

Sistem operasi Android memiliki 4 lapisan (layer) yang merupakan komponen system Android. Gambar berikut merupakan lapisan arsitektur system operasi android.



Gambar 2-6 Arsitektur Sistem Operasi Android

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari system operasi. Berikut ini susunan dari lapisan – lapisan tersebut dilihat dari lapisan dasar hingga lapisan teratas. Linux Kernel Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun system Android, yang mencakup memory management, security setting, power management dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara

hardware dan keseluruhan software. Sebagai contoh : HTC GI dilengkapi dengan kamera. Kernel Android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah kepada hardware kamera.

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, WiFi, *Flash Memory*, audio dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan. *Libraries* Bertempat di level yang sama dengan android *runtime* adalah *libraries*. Android menyertakan satu set *library-library* dalam bahasan C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem android. Kemampuan ini dapat diakses oleh programmer melewati android *application* framework. Sebagai contoh android mendukung pemutaran format audio, video dan gambar. Berikut beberapa core library tersebut:

1. *System C Library* – variasi dari implemetasi standard C sistem library (libc) milik BSD, dioptimasi untuk piranti embedded berbasis linux.
2. *Media Libraries* – berdasarkan packet video‟s openCORE, library-library ini mendukung playback dan recording dari berbagai format audio dan video popular, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
3. *Surface Manager* – Mengelola atau mengatur akses pada display dan lapisan composites 2D dan grafis 3D dari berbagai aplikasi.

4. *LibWebCore* – web browser engine modern yang mensupport android browser maupun embeddable web view.
5. SGL – Mendasari grafis 2D
6. 3D libraries – implementasi berdasarkan OpenGL ES 1.0 APIs, library ini menggunakan perangkat keras akselerasi 3D dan pengoptimalan 3D *software rasterizer*.

7. *FreeType* – bitmap dan vector font rendering
8. SQLite – mesin database yang kuat dan ringan tersedia untuk semua aplikasi.

Library - library tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi android 1.5 pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan NDK (*Native Development Toolkit*). ***Android Runtime***

Lapisan berikutnya adalah android *runtime* yang berisi *core libraries* dan *dalvik virtual machine*. *Core libraries* mencakup serangkaian inti library java, artinya android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemprograman java. Dalvik adalah java virtual machine yang memberikan kekuatan pada sistem android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiao aplikasi yang berjalan pada android berjalan pada prosesnya sendiri, dengan *instance* dari dalvik *virtual machine*. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi *virtual machine* dengan efisien. Dalvik VM dapat mengeksekusi file dengan format .dex (*Dalvik Executable*) yang telah dioptimasi untuk menggunakan minimal memory *footprint*. *Virtual machine* ini *register-based*, dan menjalankan *class-class* yang *decompile* menggunakan "dx" *tool* yang telah disertakan. Dalvik virtual machine menggunakan kernel linux untuk menjalankan fungsi – fungsi seperti *threading* dan *low level memory management*. ***Framework Aplikasi*** Lapisan selanjutnya adalah *framework* aplikasi, yang mencakup program untuk mengatur fungsi – fungsi dasar smartphone. Application framework merupakan serangkaian tool dasar seperti alokasi resources smartphone, aplikasi telepon, pergantian antar proses atau program dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada *tool – tool* dasar tersebut dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks. Programmer mendapatkan akses penuh untuk memanfaatkan API (*Android Protocol Interface*) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen – komponen, dimana setiap aplikasi dapat

menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen yang dikehendaki. Di dalam semua aplikasi terdapat servis dan sistem yang meliputi :

a. Satu set *Views* yang dapat digunakan untuk membangun aplikasi meliputi *lists, grids, text boxes, buttons,* dan *embeddable web browser*.

b. *Content Providers* yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain misalnya *contact* atau membagi data yang dimiliki.

c. *Resources Manager,* menyediakan akses ke *non-code resources* misalnya *localized strings, graphics* dan *layout files*.

d. *Notification Manager* yang memungkinkan semua aplikasi untuk menampilkan *custom alerts* pada status bar.

e. *Activity Manager* yang memanage *life cycle* dari aplikasi dan menyediakan *common navigation backstack.*

Applications Di lapisan teratas terdapat aplikasi itu sendiri. Di lapisan inilah dapat ditemukan fungsi – fungsi dasar smartphone seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain – lain. Sebagian penggunan pada lapisan inilah yang paling sering diakses, dimana pengaksesan fungsi – fungsi dasar tersebut melalui *user interface*.

2.14 Java

Di antara sekian banyak perangkat lunak yang sudah ada, kita sering mendengar istilah Java, atau disebut juga Java J2ME (Java 2 Micro Edition), yang merupakan turunan dari bahasa Java, di mana fungsi-fungsinya difokuskan untuk pembuatan aplikasi pada lingkungan telepon seluler.

Java merupakan suatu teknologi perangkat lunak yang di dalamnya mencakup bahasa pemrograman. Selain itu Java juga merupakan suatu platform yang memiliki virtual machine dan library yang diperlukan untuk menulis dan menjalankan suatu program. Pada awalnya java dikembangkan pada lingkungan komputer oleh Sun Microsystem Inc. dengan tujuan untuk menghasilkan suatu bahasa komputer sederhana tanpa harus terikat pada arsitektur tertentu.

Pada tahun 1995 Sun meluncurkan sebuah browser berbasis Java dengan julukan Hot Java, kemudian diikuti Netscape yang memutuskan untuk membuat browser dengan dilengkapi bahasa Java. Setelahnya, ikut bergabung pula berbagai pengembang ternama diantaranya IBM dan Microsoft. Pada tahun berikutnya, Sun Microsystem Inc. merilis Java Software Development Kit (JDK) pertamanya, yaitu JDK 1.1. Kemudian terus berkembang dari pemrograman applet yang berjalan di browser menjadi pemrograman kelas dunia yang banyak digunakan untuk pengembangan aneka ragam aplikasi. Bahasa pemrograman Java sendiri secara garis besar dikelompokan menjadi 3, yaitu

a. J2SE (*Java 2 Standar Edition*), digunakan untuk mengembangkan aplikasi-aplikasi desktop dan applet.

b. J2EE (*Java 2 Enterprise Edition*) dipergunakan untuk mengembangkan aplikasi-aplikasi berbasis client/server berskala enterprise

c. J2ME (*Java 2 Micro Edition*). J2ME diaplikasikan pada berbagai perangkat kecil dengan jumlah memori, kapasitas penyimpan dan user interface terbatas, seperti ponsel dan PDA.

Hingga saat ini ada dua versi platform Java J2ME yang umum digunakan pada telepon genggam yaitu MIDP 1.0 yang masih memiliki spesifikasi sederhana dan menyediakan fungsi dasar untuk aplikasi mobile, di antaranya basic user interface dan keamanan jaringan. Dan MIDP 2.0 yang sudah ditambahkan berbagai fasilitas seperti *game*, multimedia, dukungan berbagai jenis konektivitas, maupun OTA.

2.14.1 Kelebihan dan kekurangan java

Kelebihan java:

a. *Multiplatform*. Kelebihan utama dari java ialah dapat dijalankan di beberapa platform/sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan dimana 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 sistem linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, linux, Mrac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri untuk menginterpretasikan *bytecode* tersebut.

b. OOP (*Object Oriented Programming*) yang artinya semua aspek yang terdapat di java adalah objek. Java merupakan salah satu bahasan pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. 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 pemrograman termudah, bahkan untuk fungsi yang advance seperti komunikasi antara komputer sekalipun

c. *Library* yang lengkap, dimana java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program-program yang disertakan dalam pemrogram 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.

d. 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.

e. 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).

Kekurangan Java:

a. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini memungkinkan karena kode jadi java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi seperti nama-

nama kelas, metode dan tipe data. Hal yang sama juga terjadi pada Microsoft.Net Platform. Dengan demikian , algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.

b. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnyaseperti C/C++ dan Pascal (lebih spesifik lagi delphi dan *Object Pascal*). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru, tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.15 PHP

PHP adalah sebuah bahasa pemograman yang berjalan dalam sebuah web-server (server side). PHP diciptakan oleh programmer unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan sebuah script dalam wesite pribadinya dengan tujuan untuk memonitor siapa saja yang pernah mengunjungi website-nya.

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI pada sekitar tahun 1995, dan diperkenalkan kepada beberapa programmer pemula dengan alasan bahasa yang digunakan oleh PHP cukup sederhana dan mudah dipahami. Selanjutnya Rasmus menulis ulang PHP dengan bahsa C untuk meningkatkan kecepatan aksesnya.

Mulai bulan September sampai Oktober 1995, kode PHP ditulis ulang dan digabungkan menjadi PHP/F1. Baru di akhir tahun 1995 dirilis bagi umum secara gratis. Mengapa Rasmus membagikan ke publik secara gratis ? Rasmus berangapan apabila kode PHP ini berguna bagi dirinya, tentu juga akan bermanfaat untuk oranglain. Toh pada akhirnya akan kembali bermanfaat bagi dirinya sendiri.

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.

Pada tahun 1997, 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 dirubah 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.
Hingga saat ini (Januari 2011) PHP sudah merilis versi **5.3.5**

2.15.1 Prinsip kerja PHP

System kerja dari PHP diawali dengan permintaan yang beasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh webserver.

Selanjutnya webserver akan mencarikan berkas yang diminta dan menampilkan isinya di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya.

Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung script PHP ? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke web-server, web-server akan memeriksa tipe file yang diminta user. Jika tipe file yang diminta adalah pHP, maka akan memeriksa isi script dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung script PHP, permintaan user akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan script-script PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user.

2.16 MySQL

MySQL adalah sebuah perangkat lunak Pembuat database yang bersifat terbuka atau open source dan berjalan disemua platform baik Linux maupun Si Windows, **MySQL** merupakan program pengakses database yang bersifat network sehingga dapat digunakan untuk aplikasi Multi User (Pengguna Banyak). **MySQL adalah** Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial.



Gambar 2-7 Logo MySQL

MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

2.17 Tools pendukung pembuatan aplikasi android

2.17.1 Java Development Kit (JDK)

JDK adalah software development kit merupakan seperangkat kode, library, form dan paket yang berguna untuk membuat aplikasi atau applet java. Di dalam JDK terdapat *Java Runtime Environment* (JRE) yang akan memproses *byte code* yang dihasilkan dari interpenter java. Tugas dari interpenter java ini adalah

untuk mengcompile source.java menjadi source.class yang selanjutnya source.class inilah yang akan dijalankan oleh JRE.

2.17.2 Sotware Development Kit (SDK)

SDK adalah suatu kumpulan dari library dan toolkit seperti emulator untuk mengembangkan atau menciptakan aplikasi untuk suatu perangkat lunak, perangkat computer, system operasi atau platform. Di dalam SDK terdapat *tools* yang dibutuhkan dalam pengembangan android, diantaranya yaitu adb shell yang merupakan singkatan dari android development bridge yang dapat menjalankan terminal android seperti terminal pada system operasi linux, dan *command* yang terdapat dalam adb shell sendiri seperti command linux pada umumnya.

2.17.3 Eclipse

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

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

b. **Mulit-language**: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

c. **Multi-role**: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu Kelebihan dari eclipse yang membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plugin*. 46

Secara standar Eclipse selalu dilengkapi dengan *Java Development Tools* (JDE), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan *Plug-in Development Environment* (PDE) untuk 27 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. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*).