BAB 2
LANDASAN TEORI

Landasan teori merupakan teori-teori yang relevan dan dapat digunakan utnuk menjelaskjan variable-variabel penelitian. Landasan teori ini juga berfungsi sebagai sasar intuk member jawaban sementara terhadap rumusan yang diajukan serta membantu dalam menyusun *instrument* penelitian.

2.1 Pengenalan Perangkat Lunak (software)

Perangkat lunak merupakan sekumpulan instruksi dalam bahasa pemrograman yang disusun pemrogram untuk dikerjakan oleh komputer. Istilah perangkat lunak pertama kali dipopulerkan oleh Mr. Bauer pada konferensi NATO tahun 1968 **[**4**]**.

2.1.1 Perangkat Lunak Aplikasi

Perangkat lunak aplikasi merupakan perangkat lunak yang digunakan untuk menyediakan layanan kepada pelanggan (*user*) dan menjalankan bisnis agar lebih efisien. Contohnya seperti sistem akuntansi, sistem manajemen informasi dan sistem perencanaan sumber daya perusahaan.

Beberapa perangkat lunak aplikasi yang digabung bersama menjadi suatu paket kadang disebut suatu paket aplikasi (*application suite*), contohnya mocrosoft office dan openoffice.org yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya.

Adapun contoh dari perangkat lunak adalah sebagai berikut:

1. Peraangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

2.1.2 Perangkat Lunak Sistem

Perangkat lunak sistem merupakan perangkat lunak yang digunakan untuk mengoperasikan dan mempertahankan sistem komputer. Hal ini memungkinkan pelanggan (*user*) untuk menggunakan sumber daya komputer secara langsung dan melalui perangkat lunak lain. Contohnya seperti sistem operasi, *compiler* dan sistem utilitas.

2.1.3 Dasar Dasar Pembuatan Perangkat Lunak

Adapun dasar dasar dalam pembuatan perankat lunak adalah sebagai berikut:

1. Memahami masalah yang akan diangkat dalam pembuatan perangkat lunak
2. Berkomunikasi dengan baik dengan pelanggan (*user*)
3. Memilih metode perancangan yang tepat
4. Mempelajari masalah sebelum diterjemahkan ke dalam bahasa pemrograman
5. Melakukan strategi dan taktik yang baik dalam pengujian

2.1.4 Teknik Pembuatan Perangkat Lunak

Teknik dalam pembuatan perangkat lunak adalah sebagai berikut:

1. Menciptakan perangkat lunak berdasarkan solusi terbaik, hemat biaya, efektif dan praktis dalam penyelesaian masalah untuk pelayanan umat manusia.
2. Mengetahui prinsip – prinsip dasar ilmu computer dan matematika untuk mencapai efektifitas biaya dan solusi terbaik dalam pembuatan perangkat lunak.

2.1.5 Tahapan Pembangunan Perangkat Lunak

1. Analisis

   Tahapan analisis adalah tahap yang menjelaskan persyaratan kebutuhan perangkat lunak yang akan dibangun. Secara umum tahapan analisis adalah sebagai berikut :

   a. Konsep dasar
   b. Persyaratan teknis
   c. Kebutuhan non fungsional
   d. Kebutuhan fungsional

2. Design

   Tahapan design adalah sebuah representasi perangkat lunak yang dibuat untuk memfasilitasi analisis, perencanaan, pelaksanaan dan pengambilan keputusan.

3. Coding / Pengembangan

   Tahapan coding merupakan ekspresi design yang diterjemahkan ke dalam suatu bahasa pemrograman tertentu berdasarkan spesifikasi program. Tahapn coding disebut juga pekerjaan utama dari sebuah pembangunan perangkat lunak.

4. Pengujian (*testing*)

   Tahapan pengujian adalah tahap dimana perangkat lunak yang dibangun diperiksa apakah perangkat lunak memenuhi persyaratan atau tidak. Tujuan dari tahap pengujian adalah untuk mengetahui sebanyak mungkin masalah yang mungkin terdapat dalam perangkat lunak tersebut.

5. Implementasi

   Tahapan implementasi dilakukan di hadapan pengguna (*user*), perangkat lunak yang dibangun akan diterima oleh pengguna (*user*) apabila dalam tahap imlementasi tidak ada cacat pada fungsional utama, kinerja criteria dan toleransi tidak lebih dan tidak kurang dari 2% dan tidak ada masalah dalam penggunaan.

6. Pemeliharaan (*maintenance*)

   Tahapan pemeliharaan dilakukan untuk melakukan perubahan apabila diperlukan karena beberapa alasan, seperti: ada masalah dalam perangkat lunak yang harus diperbaiki, persyaratan baru yang harus dipenuhi serta perubahan dalam kebijakan yang telah disepakati di awal.

2.2 Pengenalan Internet

Internet adalah sejumlah komputer di dunia yang terhubung dalam suatu jaringan yang menerapkan konsep *client-server* (*client* = komputer yang dilayani; *server* = komputer yang melayani *client*).

Internet pada awalnya adalah sebuah jaringan komputer rancangan militer Amerika Serikat (AS) untuk memantapkan kinerja sistem komunikasinya dalam menghadapi kekuatan musuh. Hal ini berlangsung ketika terjadi perang dingin dengan Uni Soviet. Pertimbangannya adalah jika salah satu pusat sistem komunikasi berhasil dihancurkan, maka pihak militer ingin memastikan bahwa lokasi-lokasi yang masih mampu dipertahankan diri harus dapat selalu saling berkomunikasi.

Karena jaringan yang asli adalah rancangan Advanced Research Projects Agency (ARPA) dari militer AS maka internet disebut juga ARPAnet. Namun pada generasi selanjutnya internet lebih dikembangkan ke arah pendidikan, penelitian, komersial dan bidang-bidang lain yang diprakarsai oleh National Research and Education Network (NREN) di tahun 1991.

Kemudian pada tahun-tahun sesudahnya internet menjadi sangat populer dan meluas yang dapat meliputi seluruh aspek kehidupan dengan begitu banyak macam manfaat dan akses yang tak terbatas. Pada tahapan ini ruang, jarak dan waktu tidak dapat membatasi lagi untuk akses informasi global.

2.2.1 Manfaat Internet

Banyak manfaat yang diberikan dengan adanya jaringan internet, antara lain:

1. Gudang Informasi

   Dengan adanya internet, dunia ilmu pengetahuan semakin terbuka lebar, penyebaran informasipun semakin cepat, segala informasi di belahan dunia manapun dapat diperoleh dalam sekejap. Informasi yang tadinya sulit diperoleh, saat ini sudah bukan sesuatu yang sulit lagi. Ini semua dimungkinkan dengan adanya fasilitas *search engine* atau mesin pencari dalam dunia internet, yang artinya adalah pencarian segala informasi yang kita perlukan, yang bisa berupa data, file, gambar, music, maupun film.

Search engine adalah suatu web khusus yang menyediakan pelayanan untuk mengorganisasi, menyusun index berdasarkan kategori, dari beberapa website yang telah mendaftarkan sitenya, serta meberikan rate berdasarkan dari seringnya site tersebut dikunjungi. Hal tersebut akan sangat membantu untuk menemukan halaman web yang dibutuhkan, cukup hanya dengan mengetikkan kata kunci pada form yang telah disediakan.

2. Belanja Online

   Membeli sesuatu secara instan, hal ini dimungkinkan dengan adanya internet karena sekarang telah banyak halaman web yang menyediakan toko online di internet. Tinggal memilih produk yang akan dibeli kemudian dibayar dengan kartu kredit atau dengan transfer via atm atau internet banking, barang segera dikirm dan esoknya barang langsung sampai tepat di depan pintu rumah.

3. Pendidikan

   Dengan internet sekarang dapat membantu mengembangkan ilmu pengetahuan dan pendidikan dengan hanya mengunjungi alamat website yang mengkhususkan memberikan informasi tentang pendidikan, seperti: www.apasich.com. Website ini mencakup seluruh informasi, data serta statistik yang sangat dibutuhkan dalam perkembangan dunia pendidikan dan ilmu pengetahuan di Indonesia.

2.3 Pengenalan E-Mail

Email atau *electronic mail* (surat elektronik) merupakan metode penyimpanan dan pengiriman yang melibatkan aktivitas seperti mengirim, menerima dan menyimpan surat melalui system komunikasi elektronik pada internet.

Konsep email seperti mengirim surat dengan pos biasa, dimana mengirimkan surat ke kantor pos dengan dibubuhi alamat yang kita tuju. Dari kantor pos tersebut akan disampaikan ke kantor pos yang terdekat dengan alamat yang dituju dan akhirnya sampai ke alamat tersebut. Dan penerima hanya membuka kotak pos nya saja yang ada didepan rumah.

Dengan email data dikrim secara elektronik sehingga sampai di tujuan dengan sangat cepat. Email juga dapat mengirim file file berupa program, gambar, grafik, dan lain lain. email juga dapat mengirim ke lebih dari satu alamt email pada saat yang bersamaan.

Sebuah alamt surat (*mail address*) adalah gabungan dari nama dan alamat. Dalam internet mail hal ini didefinisikan dengan nama user dan alamat host. Biasanya bentuk dasar dari *mail address* adalah: ***nama@hostaddress*** dimana nama adalah nama user dan hostaddress adalah *server* tempat mail disimpan.

2.3.1 Struktur Pesan Email

Pesan email terdiri dari sekumpulan karakter ASCII yang terdiri dari beberapa baris. Setiap baris dipisahkan oleh dua karakter berturut-turut, yaitu karakter carriage return (CR) dan karakter line feed (LF). Karakter CR merupakan karakter berkode ASCII 13, sedangkan karakter LF berkode ASCII 10.

Pesan email terdiri dari bagian header dan bagian body yang dipisahkan oleh satu baris kosong. Bagian header merupakan kumpulan dari baris yang menyatakan informasi dari pesan. Sedangkan, body adalah bagian yang berupa isi dari pesan. Contoh struktur pesan email dapat dilihat di bawah ini:

```
From: <alamat email pengirim>
To: <alamat email tujuan>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
This is a message
      just to say hello.
```

Untuk mengirimkan pesan email selain teks ASCII, dibutuhkan pemakaian header MIME (Multi Purpose Mail Extensions) pada email yang dikirimkan. Header utama dari MIME adalah sebagai berikut:

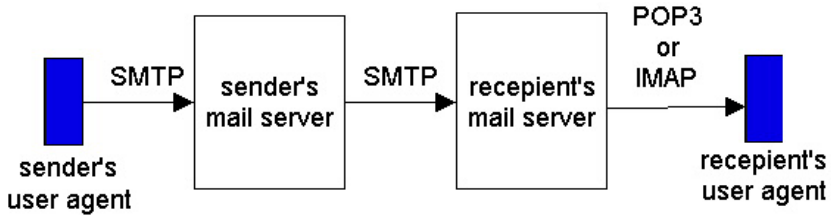
1. Content-Type: header yang berisi informasi mengenai tipe data atau file yang dikirimkan.
2. Content-Transfer-Encoding: header yang berisi informasi mengenai metode encoding yang digunakan pada pengiriman data atau file.

Kedua informasi diatas dibutuhkan oleh user agent penerima untuk menentukan metode yang tepat untuk menampilkan isi dari pesan email.

Pesan email dikirimkan berdasarkan informasi alamat email tujuan. Alamat email berbentuk string ASCII dan terbagi atas dua bagian yang dipisahkan oleh karakter "@". Bagian pertama adalah locally interpreted yang merupakan penanda dari mailbox tujuan. Sedangkan, bagian kedua adalah internet domain yang merupakan alamat dari mail server.

2.3.2 Sistem Kerja Email

Pertama-tama user agent pengirim mengirimkan pesan kepada mail server milik pengirim. Mail server pengirim kemudian memeriksa tujuan email tersebut dan meneruskannya kepada mail server penerima. Email tersebut disimpan oleh mail server penerima hingga pengguna mengambil sendiri pesan tersebut dengan menggunakan user agent-nya. Untuk lebih jelasnya dapat dilihat pada gambar berikut ini:



Gambar 0.1 Sistem Kerja Email

2.3.3 Protokol E-Mail

Protokol email dibagi menjadi 3:

1. *Simple Mail Transfer Protocol (SMTP)*

   SMTP merupakan sebuah protokol email yang hanya mengijinkan penggunanya untuk tujuan mengirim email saja. Port untuk SMTP adalah port 25.

2. *Post Office Protocol (POP/POP3)*

   POP adalah protokol dasar yang digunakan oleh pengguna untuk mentransfer (*download*) seluruh pesan email dari server melalui koneksi TCP/IP. Dengan menggunakan metode ini, pesan email yang ada di server akan dihapus dan ditransfer ke PC pengguna, tetapi pengguna bisa mengarahkan server untuk menyimpan sementara pesan tersebut untuk periode waktu yang sudah ditentukan. POP digunakan karena pengguna dapat membaca email tanpa koneksi internet. Port dasar untuk POP adalah port 110.

3. Internet Message Access Protocol (IMAP)

IMAP dan POP/POP3 memiliki kesamaan yaitu keduanya merupakan protokol untuk mentransfer seluruh pesan email dari server, namun protokol IMAP hanya mentransfer seluruh email ke PC pengguna tanpa menghapus email yang berada di server kecuali pengguna itu sendiri yang menghapus email tersebut. Port yang digunakan untuk IMAP adalah port 143.

2.4 Pengenalan Push Mail

Push e-mail digunakan untuk menggambarkan system surat elektronik yang menyediakan kemampuan selalu siaga, dimana setiap surat baru dengan segera dipindahkan secara aktif (didorong atau pushed) oleh agen pengantar surat / MDA (Mail Delivery Agent) ke agen pengguna surat / MUA (Mail User Agent), disebut juga sebagai pelanggan surat. Surat pelanggan di antaranya termasuk telepon pintar (smart phones) dan merupakan Aplikasi IMAP (Internet Message Access Protocol) pada computer personal.

2.5 Pengenalan Yahoo Messenger

Yahoo messenger (YM) merupakan program pengirim pesan instan popular yang disediakan oleh Yahoo!. Yahoo messenger tersedia secara gratis dan dapat diunduh serta diakses menggunakan Yahoo! ID yang biasa digunakan untuk layanan akses Yahoo! yang lainnya, seperti Yahoo! Mail. Penggunaan ID ini juga mengakibatkan pengguna dapat langsung diberitahu bila mendapat sebuah email.

Yahoo! telah mengumumkan kerjasama dengan Microsoft untuk bergabung dalam jaringan instant messenger. Hal ini mengakibatkan Yahoo! Messenger dapat berhubungan dengan layanan NET Messenger milik Microsoft. Layanan ini mulai berfungsi sejak 13 Juli 2006).

2.5.1 Yahoo Messenger SDK

YM SDK (*Yahoo Messenger Development Kit*) merupakan *development tools* resmi milik Yahoo untuk membangun aplikasi Yahoo Messenger *client.* YM SDK berbasis HTTP dan mengatur format *request* dan *response* untuk mengakses server YM. Format yang diatur meliputi : alamat URL *request*, data paramater, data yang dipertukarkan, dan metode *request* pada HTTP seperti GET, POST, PUT, atau DELETE **[**5**]**.

YM SDK diakses dengan langkah-langkah sebagai berikut :

1. Mengakses situs http://developer.yahoo.com untuk mendapatkan *Open Authentication* (*OAuth*) *key* yang akan digunakan oleh aplikasi untuk mengakses *server* YM. *Oauth key* terdiri dari dua data utama : *Oauth consumer key* dan *consumer secret key*.
2. Melakukan *authentication* kepada *server* YM dengan menggunakan protokol *Direct OAuth*. Bagian ini dilakukan oleh aplikasi YM *client* yang dibangun.
3. Aplikasi mengakses YM API untuk menggunakan layanan yang disediakan oleh YM SDK seperti : *login*, mengirim pesan, *logout*, dll.

2.5.2 Yahoo Messenger Direct OAuth

YM SDK menggunakan protokol OAuth (*Open Authentication*) untuk *authentication* sebelum dapat mengakses YM API. OAuth merupakan *open protocol* yang didefinisikan pada RFC 5849 untuk *authorization secure* API yang dapat diakses dari aplikasi *desktop* ataupun web.

Mekanisme *authentication* OAuth dilakukan untuk mendapatkan *access token* yang digunakan pada setiap pemanggilan YM API. Langkah-langkah dalam Direct Oauth dilakukan dalam dua proses. Proses pertama dilakukan untuk mendapatkan *pre-approved request token.*

Sedangkan proses kedua untuk mendapatkan *access token* dan *access session handle*. *Access token* mempunyai durasi masa berlaku yang biasanya bernilai 3600 detik. Sebelum masa berlaku berakhir, *access token* wajib diperbaharui dengan mengirimkan pesan *refresh access token* [**6**].

2.5.3 Yahoo Messenger API

YM API (*YM Application Programming Interface*) merupakan kumpulan format pesan HTTP yang digunakan untuk mengakses layanan yang disediakan oleh YM. Layanan tersebut antara lain : *login*, menerima daftar kontak, mengirim pesan, menerima pesan, *logout*, dll [**7**].

Pesan HTTP yang dipertukarkan dalam YM API berformat JSON (*JavaScript Object Notation*). JSON adalah format pertukaran data yang dapat dengan mudah dibaca, di-*parsing*, dan di-*generate* baik oleh mesin maupun oleh manusia. Walaupun JSON berbasis bahasa JavaScript, tetapi JSON dapat diimplementasikan dalam berbagai bahasa pemrograman.

Setiap *request* API dalam YM SDK membutuhkan parameter bernama "*Authorization*" yang ditempatkan pada *header* HTTP.

Terdapat banyak layanan yang disediakan oleh YM API untuk membangun YM *client.* Tapi beberapa yang paling sering digunakan adalah *session management, message management, buddy list authorization management,* dan *notification management*.

Setiap request API dalam YM SDK membutuhkan parameter bernama "Authorization" yang ditempatkan pada header HTTP. Berikut adalah formatnya:

```
Authorization: OAuth
realm="yahooapis.com",
oauth_consumer_key=”<api_key>”,
oauth_nonce=”<random_value>”,
oauth_signature_method="PLAINTEXT",
oauth_timestamp=”<unix_timestamp>”,
oauth_token=”<access_token>”,
oauth_version=”1.0”,
oauth_signature=”<shared_secret>&<oauth_token_secret>”
```

2.5.3.1 Session Management API

API ini berfungsi untuk melakukan login, logout, dan refresh session pada Yahoo Messenger. Untuk melakukan login pesan dikirimkan pada URL "http://developer.messenger.yahooapis.com/v1/session".

Sedangkan untuk logout dan refresh session pesan dikirimkan pada URL "http://<server>/v1/session", dimana "<server>" merupakan nilai dari parameter server yang didapatkan pada response ketika melakukan login.

2.5.3.2 Message Management API

API ini digunakan untuk mengirimkan pesan kepada kontak lain pada YM. Request pada API ini dikirimkan pada alamat URL "http://<server>/v1/message/yahoo/<target>". Dimana "<target>" merupakan id Yahoo tujuan pesan. Sedangkan "<server>" adalah nilai yang didapatkan ketika response login.

2.5.3.3 Buddylist Authorization Management API

API ini berfungsi untuk mengontrol *id* Yahoo yang akan ditambahkan pada daftar kontak pengguna. Pesan *Buddylist Authorization Management* dikirimkan pada alamat URL "http://<server>/v1/buddyrequest/yahoo/<adder>". Dimana "<adder>" adalah *id* Yahoo yang sebelumnya mengirimkan kontak *request*.

2.5.3.4 Notification Management API

API ini berfungsi untuk mengambil data notifikasi yang ditujukan untuk pengguna. Notifikasi yang dimaksud antara lain : pesan, update kontak, request kontak, email alerts, file transfer invite, dll. Pesan dari Notification Management API dikirimkan pada alamat "http://<notifyServer>/v1/pushChannel/<loginID>".

Dimana "<notifyServer>" adalah nilai yang didapatkan dari response login. Sedangkan "<loginID>" adalah id Yahoo yang digunakan ketika login.

Notification Management API menggunakan sistem Comet-Style Push. Comet-Style Push bekerja dengan cara mengirimkan request HTTP GET yang long-lived kepada server notifikasi Yahoo. Hal ini membuat koneksi persistent virtual antara client dan server. Beberapa pedoman dalam menggunakan sistem Comet-Style Push Yahoo adalah sebagai berikut:

1. Setiap session mempunyai maksimal satu koneksi long-lived HTTP GET terhadap server notifikasi Yahoo.
2. Koneksi terputus jika server telah membalas notifikasi atau terjadi timeout pada sisi client.
3. Setelah koneksi terputus, client diharapkan untuk membuat koneksi long-lived HTTP GET yang lain.
4. Waktu timeout pada koneksi long-lived HTTP GET maksimal 2 menit.
5. Client disarankan untuk memasukkan nilai random pada URL untuk menghidari cache.
6. Setiap request harus menyertakan cookie bernama "IM" pada header HTTP. Nilai dari "IM" didapatkan pada response header HTTP ketika melakukan login atau update session.

2.6 Tools Analisis

Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas pemodelan yang membawanya kepada suatu spesifikasi lengkap dari persyaratan representasi desain yang komprehensif bagi perangkat lunak yang akan dibangun. Model analisis, yang sebenarnya merupakan serangkaian model representasi teknis dari sistem. Saat ini ada dua yang mendominasi *landscape* pemodelan analisis. Yang pertama analisis terstruktur, adalah pemodelan klasik dan yang kedua adalah analisis berorientasi objek.

2.6.1 Pengenalan Metode Terstruktur

Pendekatan Terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Pendekatan terstruktur merupakan pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan untuk kemudian dapat disatukan kembali menjadi satu kesatuan yang dapat dipergunakan untuk memecahkan masalah. Pendekatan terstruktur dalam pengembangan sistem informasi adalah proses yang berorientasi kepada teknik yang digunakan untuk merancang dan menulis program secara jelas dan konsisten.

Prinsip dari pendekatan terstruktur adalah jika suatu proses telah sampai pada suatu langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya. Ciri-ciri pendekatan terstruktur, yaitu:

1. Merancang berdasar modul. Modularisasi adalah proses yang membagi suatu system menjadi beberapa modul yang dapat beroperasi secara independen.
2. Bekerja dengan pendekatan top-down. Dimulai dari level atas (secara global) kemudian diuraikan sampai ke tingkat modul (rinci).
3. Dilakukan secara iterasi.
4. Kegiatan dilakukan secara paralel.

2.6.1.1 DFD (Data Flow Diagram)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. *DFD* sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan alir data, yaitu :

1. Kejelasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram alir data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

DFD terdiri dari *context diagram* dan diagram rinci (*DFD leveled*), *context diagram* berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *DFD leveled* menggambarkan sistem jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran data penyimpanan data, model ini hanya memodelkan system dari sudut pandang fungsi.

Beberapa simbol yang akan digunakan di dalam *DFD* antara lain adalah sebagai berikut :

1. Kesatuan luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu system dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan dilingkungan luar sistem dapat berupa orang, organisasi atau system lainnya yang berada dilingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

Kesatuan luar ini kebanyakan adalah salah satu dari berikut ini :

a. Suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang sedang dikembangkan.
b. Orang atau sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan.
c. Suatu organisasi atau orang di luar organisasi.
d. Sistem informasi yang lain di luar sistem yang sedang dikembangkan.
e. Sumber asli dari suatu transaksi.
f. Penerimaan akhir dari suatu laporan yang dihasilakn oleh sistem.

2. Aliran Data (*Data Flow*)

Aliran data di *DFD* diberi simbol suatu panah. Aliran data ini mengalir diantara proses (*process*), simpan data (*data store*) dan kesatuan luar (*external entity*). Aliran data ini menunjukkan aliran dari data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

3. Proses

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses yang digambarkan secara umum. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut – sudutnya tumpul.

4. Berkas atau Simpanan Data (*Data Store*)

Berkas atau simpanan data merupakan simpanan dari data yang dapat berupa :

a. Suatu file atau database di sistem komputer.
b. Suatu arsip atau catatan manual.
c. Suatu kotak tempat data di meja seseorang.
d. Suatu tabel acuan manual.

2.6.1.2 ERD (Entity Relationship Diagram)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. *ERD* digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relative kompleks. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. *ERD* menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan yaitu :

1. *Entity*

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.

2. Atribut

Entiti mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entiti.

3. Hubungan

Relationship sebagaimana halnya entiti maka dalam hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isis dari hubungan itu sendiri. Relasi antara dua *file* atau dua tabel dapat dikategorikan menjadi tiga macam, yaitu :

a. One to One Relationship

Yang berarti entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

b. One to Many Relationship

Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada satu himpunan entitas B, tetapi tidak sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

c. Many to Many Relationship

Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada satu himpunan entitas B dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan banyak entitas pada himpunan entitas A.

2.7 Tool Aplikasi

Tool aplikasi merupakan alat atau cara yang digunakan untuk pembangunan perangkat lunak, meliputi bahsa pemrograman yang digunakan dan compiler untuk menjalankan, penyusunan *coding* serta *interface* perangkat lunak.

2.7.1 Pengenalan Java

Java adalah bahasa pemrograman dengan basis C dan C++. Sehingga apabila sudah terbiasa dengan C atau C++, tidak akan sulit untuk mempelajari bahasa pemrograman java. Sejarah *java* berawal pada 1991 ketika perusahaan *Sun Microsystem* memulai *Green Project*, yakni projek penelitian untuk membuat

bahasa yang akan digunakan pada chip-chip *embedded* untuk *device intelligent cunsomer electronic*. Bahasa tersebut haruslah bersifat *multiplatform*, artinya tidak tergantung pada *vendor* yang memanufaktur chip tersebut.

Dalam penelitiannya, *Projek Green* berhasil membuat *prototype* semacam PDA (*Personal Data Assistance*) yang dapat berkomunikasi antara satu dengan yang lain dan diberi nama *Star 7*. Ide berawal untuk membuat sistem operasi bagi *Star 7* berbasis *C* dan *C++*. Setelah berjalan beberapa lama, James Gosling, salah seorang anggota tim merasa kurang puas dengan beberapa karakteristik dan kedua bahasa tersebut dan berusaha mengembangkan bahasa lain. bahasa tersebut kemudian dinamakan Oak, diinspirasi ketika dia meliahat pohon di seberang kaca ruang kantornya. Belakangan *Oak* beralih nama menjadi *Java*.

Karena pada awalnya ditujukan untuk pemrograman *device* kecil, *java* memiliki karakterisitik berukuran kecil, efisien, dan *portable* untuk berbagai *hardware*. *Projek Green* sempat terancam terhenti karena dalam perkembangannya, *device* ini belum memiliki pasar seperti yang diramalkan semula. Selanjutnya *Java* diarahkan untuk pemrograman internet. Secara kebetulan, fitur-fitur *Java* yang telah disebutkan sebelumnya sangat sesuai bagi pengembangan internet sehingga dalam beberapa tahun belakangan ini *Java* telah menjadi primadona untuk pemrograman yang berbasis internet.

2.7.1.1 Keunggulan Java

Java memiliki keunggulan apabila dibandingkan dengan bahsa pemrograman yang lain. berikut adalah beberapa keunggulan dari java:

1. *Java* bersifat sederhana dan relatif mudah

   Java dimodelkan sebagian dari bahasa *C++*, namun dengan memperbaiki beberapa karakteristik *C++*, seperti mengurangi kompleksitas beberapa fitur, penambahan fungsionalitas, serta penghilangan beberapa aspek pemicu ketidakstabilan system pada *C++*.

2. *Java* berorientasi pada objek (*object oriented*)

   Java adalah bahasa pemrograman yang berorientasi objek (OOP), bukan seperti *Pascal*, *Basic*, atau *C* yang berbasis prosedural. Dalam memecahkan masalah, *java* membagi program menjadi objek-objek, kemudian memodelkan

sifat dan tingkah laku masing-masing. Selanjutnya, java menentukan dan mengatur interaksi antara objek yang satu dengan yang lain.

3. *Java* bersifat terdistribusi

Pada decade awal perkembangan PC (*Personal Computer*), komputer hanya bersifat sebagai workstation tunggal, tidak terhubung satu sama lain. Saat ini, system komputerisasi cenderung terdistribusi, mulai dari *workstation client, email server, database server, web server, proxy server* dan sebagainya.

4. *Java* bersifat *multi platform*

Banyak platform Operating System, mulai dari *windows, apple*, serta berbagai varian *UNIX* dan *LINUX*. Pada umumnya, program yang dibuat dan dikompile di suatu platform hanya bisa dijalankan di platform tersebut. Namun *java* bersifat *multiplatform*, yang artinya dapat diterjemahkan oleh *java interpreter* pada berbagai system operasi.

5. *Java* bersifat *multi trhead*

Thread adalah proses yang dapat dikerjakan oleh program dalam suatu waktu. *Java* bersifat *multithread*, artinya dapat mengerjakan beberapa proses dalam waktu yang hamper bersamaan.

2.7.1.2 Ruang Lingkup Java

Platform Java memiliki tiga buah edisi yang berbeda, yaitu J2EE (Java2 *Enterprise Edition*), J2ME (Java2 *Micro Edition*), dan J2SE (Java2 *Second Edition*). J2EE adalah kelompok dari beberapa API dari Java dan teknologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dianggap sebagai *middleware* atau teknologi yang berjalan di *server*, namun sebenarnya J2EE tidak hanya terbatas untuk itu. J2EE juga mencakup teknologi yang dapat digunakan disemua lapisan dari sebuah sistem informasi. Implementasi J2EE menyediakan kelas dasar dan API dari Java yang mendukung pengembangan dan rutin standar untuk aplikasi *client* maupun *server*, termasuk aplikasi yang berjalan di *web browser*. J2SE adalah lingkungan dasar dari Java.

2.7.1.2.1 Java 2 Standard Edition (J2SE)

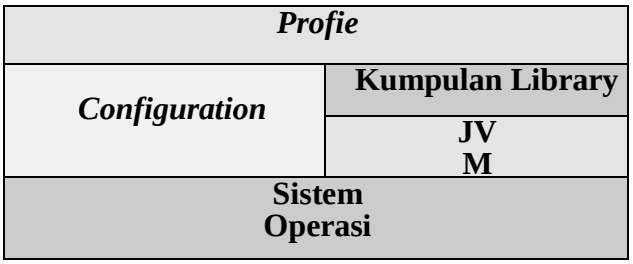
J2SE adalah ini dari bahasa pemrograman Java. Java Development Kit (JDK) Adalah salah satu tool dari J2SE untuk mengompilasi dan menjalankan

program Java. Didalamnya terdapat tool untuk mengompilasi program Java dan JRE.

2.7.1.2.2 Java 2 Micro Edition (J2ME)

Java2 *Micro Edition* atau J2ME adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak Java pada perangkat elektronik beserta perangkat pendukung lainnya. Pada J2ME, jika perangkat lunak berfungsi dengan baik pada sebuah perangkat maka belum tentu juga berfungsi dengan baik pada perangkat lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop*. J2ME biasa digunakan pada telepon selular, *pager*, *personal digital assistants* (PDA's) dan sejenisnya.

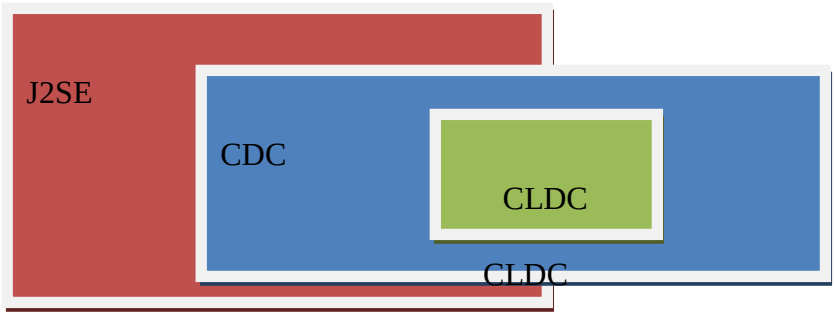
J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai *library* khusus yang tidak dimilikiJ2SE. Arsitektur J2ME dapat di lihat pada gambar berikut.



Gambar 0.2 Arsitektur J2ME

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat (device) yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas (limited). Sebagian besar ponsel tidak mengizinkan aplikasi J2ME menulis pada *file* karena alasan keamanan.

Configuration merupakan Java *library* minimum dan kapabilitas yang dipunya oleh para pengembang J2ME, yang maksudnya adalah sebuah *mobile device* dengan kemampuan Java akan dioptimalkan agar menjadi sesuai. *Configuration* hanyalah mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar-*device*. Dalam J2ME telah didefinisikan dua buah konfigurasi yaitu *Conected Limited Device Configuration* (CLDC) untuk perangkat kecil dan *Connected Device Configuration* (CDC) untuk perangkat yang lebih besar.



Gambar 0.3 Ruang Lingkup Configuratin

Profile berbeda dengan *Configuration*, *profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Dalam J2ME terdapat dua buah *profile* yaitu MIDP dan *Foundation Profile*. Keterhubungan antara *configuration* dan *profile* yang ada pada J2ME beserta jenis mesin virtualnya dapat dilihat pada gambar berikut.

Aplikasi J2ME (MIDlet)	Aplikasi J2ME	Applet/Aplikasi J2SE
KVM	CVM	JVM
MIDP	Foundation Profile	J2SE
CLDC	CDC	
kompleks		

Table 0-1 Hubungan J2ME dengan J2SE

2.7.1.2.3 Java Aplication Description (JAD)

Digunakan untuk mendeskripsikan isi aplikasi untuk pekerluan pemetaan. File JAD berisi deskripsi file JAR (Java Archieve) dan pemetaan atribut MIDlet, sedangkan file JAR berisi kumpulan kelas dan resource.

2.8 Pengenalan Netbeans

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan platform *NetBeans*.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (Java *archive*) yang memuat kelas-kelas Java untuk berinetraksi dengan *NetBeans Open API* dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerful*.

2.8.1 Platform NetBeans

Platform NetBeans adalah framework yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi desktop. Ketika aplikasi berbasis platform NetBeans dijalankan, kelas main dari platform dieksekusi. Modul-modul yang tersedia ditempatkan di sebuah *registry* di dalam memori, dan tugas *startup* modul dijalankan. Secara umum, kode modul dimuatkan ke dalam memori hanya ketika ia diperlukan.

Aplikasi dapat menginstal modul secara dinamis. Aplikasi dapat memasukkan modul *Update Center* untuk mengijinkan pengguna aplikasi men-*download digitally-signed upgrade* dan fitur-fitur baru secara langsung ke dalam aplikasi yang berjalan. Penginstalan kembali sebuah upgrade atau rilis baru tidak memaksa pengguna untuk men-*download* keseluruhan aplikasi lagi.

Platform NetBeans menawarkan layanan-layanan yang umum bagi aplikasi desktop, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur yang disediakan oleh platform NetBeans:

1. Manajemen antarmuka (misal: menu & toolbar)
2. Manajemen pengaturan pengguna
3. Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
4. Manajemen jendela
5. *Wizard framework* (mendukung dialog langkah demi langkah)

2.8.2 NetBeans IDE

NetBeans IDE adalah IDE (integrated development environment) open source yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan *refactoring*.

2.8.2.1 NetBeans Mobility Pack

NetBeans Mobility Pack adalah alat untuk mengembangkan aplikasi yang berjalan pada perangkat bergerak (mobile), umumnya telepon seluler, tetapi juga mencakup PDA, dan lain-lain.

NetBeans Mobility Pack dapat digunakan untuk menulis, menguji, dan debugging aplikasi untuk perangkat bergerak yang menggunakan teknologi berplatform Java Micro Edition (platform Java ME). Paket ini mengintegrasikan dukungan terhadap Mobile Information Device Profile (MIDP) 2.0, Connected Limited Device Configuration (CLDC) 1.1, dan Connected Device Configuration (CDC). Emulator dari pihak ketiga dapat diintegrasikan dengan mudah untuk lingkungan pengujian yang lebih kokoh. NetBeans Mobility Pack saat ini tersedia dalam dua klaster yang berbeda, yang satu memuat CDC dan yang lainnya CLDC.

2.8.2.2 NetBeans Profiler

NetBeans Profiler adalah alat untuk mengoptimalkan aplikasi Java, membantu menemukan kebocoran memori dan mengoptimalkan kecepatan.

Profiler ini berdasarkan sebuah proyek riset *Sun Laboratories* yang dahulu bernama Jfluid. Riset tersebut mengungkap teknik tertentu yang dapat digunakan untuk menurunkan *overhead* proses *profiling* aplikasi Java. Salah satu dari teknik tersebut adalah instrumentas i kode byte dinamis, yang berguna untuk *profiling* aplikasi Java yang besar. Dengan menggunakan instrumentasi kode byte dinamis dan algoritma-algoritma tambahan, *Netbeans Profiler* mampu mendapatkan informasi *runtime* aplikasi yang terlalu besar atau kompleks bagi *profiler* lain. NetBeans IDE 6.0 akan mendukung *Profiling Point* yang memungkinkan kita memprofilkan titik yang tepat dari eksekusi dan mengukur waktu eksekusi.

2.8.2.3 NetBeans C/C++ Pack

NetBeans C/C++ Pack menambahkan dukungan terhadap pengembang C/C++ ke NetBeans IDE 5.5. Paket ini memperbolehkan pengembang menggunakan sekumpulan kompiler dan alat sendiri bersama dengan NetBeans IDE untuk membangun aplikasi *native* untuk MS Windows, Linux, dan Solaris. Paket ini membuat editor mengenali bahasa C/C++ dan menyediakan *project template*, browser kelas yang dinamis, dukungan pembuatan file dan fungsionalitas *debugger*. Para pengembang juga dapat mengembangkan paket tersebut dengan fungsionalitas tambahan mereka sendiri.

2.8.2.4 NetBeans Enterprise Pack

NetBeans Enterprise Pack memperluas dukungan terhadap pengembangan aplikasi perusahaan dan *web service* mulai NetBeans IDE 5.5.

Enterprise Pack ini mengembangkan kemampuan untuk menulis, menguji, dan *debug* aplikasi dengan arsitektur berorientasi layanan (*Service-Oriented Architecture*) menggunakan XML, BPEL, dan Java *web service*. Paket ini menambahkan alat desain visual untuk pemodelan UML, skema XML, dan *web service orchestration*, juga dukungan untuk *web service* dengan menggunakan identitas yang aman. Paket ini juga menginstal dan mengkonfigurasi *runtime* yang diperlukan, termasuk mesin BPEL dan server manajemen identitas yang terintegrasi dengan *Sun Java System Application Server*.

2.8.2.5 NetBeans Ruby Pack

Versi NetBeans 6.0 mengijinkan pengembangan IDE menggunakan Ruby dan Jruby, sebagaimana Rails untuk dua implementasi Ruby yang lain. *Preview NetBeans Ruby Pack* tersedia sejak rilis *Milestone 7* NetBeans 6.

Ruby Pack memasukkan fungsionalitas editor seperti:

1. Pengeditan dasar
2. Pewarnaan sintaks untuk Ruby
3. Pelengkapan kode
4. *Occurence highlighting*
5. Pop-up dokumentasi yang terintegrasi untuk pemanggilan Ruby API
6. Analisis semantik dengan *highlighting* parameter dan variabel lokal yang tidak terpakai

2.8.2.7 NetBeans JavaScript Editor

NetBeans JavaScript Editor menyediakan perluasan dukungan terhadap JavaScript dan CSS. Fitur-fiturnya antara lain:

1. Editor JavaScript
   a. *Syntax highlighting*
   b. Pelengkapan kode untuk objek dan fungsi *native*
   c. Semua fitur dalam editor netbeans
   d. Pembuatan kerangka kelas javascript secara otomatis
   e. Pembuatan pemanggilan ajax dari *template*
2. Ekstensi editor CSS
   a. Pelengkapan kode untuk nama-nama *style*
   b. Navigasi cepat melalui panel navigator
   c. Penampilan deklarasi aturan css di *list view*
   d. Penampilan struktur file di *tree view*

Mengurutkan *outline view* berdasarkan nama, tipe, atau urutan deklarasi (*list & tree*).