

BAB II

TINJAUAN PUSTAKA

II.1. Sejarah *Chatting*

Istilah komunikasi melalui internet biasa dikenal dengan *chatting*. Istilah *chatting* sebenarnya berasal dari nama IRC (*Internet Relay Chat*), dimana seseorang dapat saling berkomunikasi melalui internet dengan menginstal sebuah perangkat lunak (*software*) bantu. IRC adalah suatu bentuk komunikasi langsung melalui internet. IRC didesain untuk komunikasi grup pada forum diskusi yang dikenal dengan *channel*, tetapi mengizinkan pula untuk melakukan komunikasi secara individu. IRC dibuat oleh *Jarko Oikarinen* pada bulan Agustus 1988 untuk menggantikan sebuah program yang disebut *Multiuser Talk* (MUT) pada *Buletin Board System* (BBS). IRC terinspirasi Bitney Relay Chat yang dioperasikan pada jaringan Bitnet.

Sampai sekarang *chatting* masih digunakan bahkan perkembangannya semakin pesat. Sebagian orang memilih menggunakan *chatting* karena selain efektif juga menghemat biaya. Selama ada jaringan internet di sekitar kita, komunikasi akan dapat berjalan tanpa memikirkan biaya yang dikeluarkan tidak seperti jika kita menggunakan telepon. Banyak manfaat dan kelebihan dari *Chatting* yang dapat kita gunakan.

- a. Sebagai sarana untuk mencari teman / relasi.
- b. Tempat berkumpulnya orang yang memiliki kegemaran yang sama.
- c. Menghemat biaya dalam berkomunikasi.

- a. Dapat digunakan sebagai media *teleconference*.
- b. Sebagai wadah untuk bertukar pikiran (*sharing*) pendapat secara interaktif.
- c. Menghemat biaya perusahaan dan ruangan untuk bertemu.

Chatting dibagi menjadi 2 macam, yaitu berupa teks dan suara. Contoh *chatting* berbasis teks adalah *mIRC*, *yahoo messenger*, dan sebagainya sedangkan *chatting* berbasis suara disebut VoIP adalah *Skype*, *Google Talk* dan *Yahoo messenger with voice*. (Budi Sutedjo, 2007 hal 52).

II.2. Pengertian Aplikasi

Menurut Whitten Perancangan Sistem adalah “Proses dimana keperluan pengguna dirubah ke dalam bentuk paket perangkat lunak dan atau kedalam spesifikasi pada komputer yang berdasarkan pada sistem informasi.”

(Kristanto, 1994, hal 60).

II.3. Pengertian Perancangan

Pembuatan sistem informasi akuntansi dibutuhkan adanya perancangan tentang apa yang akan dibuat dan apa yang akan dihasilkan. Adanya suatu rancangan dalam sistem informasi akuntansi, maka kita akan tahu kemana tujuan kita.

Definisi perancangan menurut Al-Bahra (2005:51) yang terdapat dalam buku yang berjudul Analisis dan Desain Sistem Informasi, menjelaskan bahwa: “perancangan adalah kemampuan untuk membuat beberapa alternatif pemecahan

masalah.” Azhar Susanto (2004:332) menjelaskan dalam buku yang berjudul Sistem Informasi Manajemen Konsep dan Pengembangannya yaitu: “perancangan

adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis.” Berdasarkan dua definisi perancangan tersebut, maka penulis dapat menyimpulkan bahwa perancangan merupakan suatu alternatif untuk memecahkan masalah dan yang telah dipilih selama tahap analisis dalam pemecahan masalah yang dihadapi perusahaan.

II.4. Alat Bantu Perancangan System

Alat bantu yang digunakan dalam perancangan atau pengembangan sistem yang digunakan dalam penulisan ini adalah sebagai berikut:

II.4.1 CLDC(System Development Life Cycle)

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Pengembangan sistem (SDLC) diperlukan untuk menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada hal ini dikarenakan adanya permasalahan di sistem lama dan pertumbuhan organisasi.

Dengan telah dikembangkannya sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di sistem yang baru. Peningkatan-peningkatan ini berhubungan dengan PIECES yaitu sebagai berikut : *Performance* (kinerja), peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* dan *response time*. *Throughput* adalah jumlah dari pekerjaan yang dapat dilakukan suatu saat tertentu. *Response*

time adalah rata-rata waktu yang tertunda diantara dua transaksi atau pekerjaan ditambah dengan waktu *response* untuk menanggapi pekerjaan tersebut. *Information* (informasi), peningkatan terhadap kualitas informasi yang disajikan. *Economy* (ekonomis), peningkatan terhadap manfaat-manfaat, keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi.

SDLC (*System Development Life Cycle*) adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan *programmer* dalam membangun sistem informasi. Langkah yang digunakan meliputi : melakukan survei dan menilai kelayakan proyek pengembangan sistem informasi, mempelajari dan menganalisis sistem informasi yang sedang berjalan, menentukan permintaan pemakai sistem informasi, memilih solusi atau pemecahan masalah yang paling baik, menentukan perangkat keras (*hardware*) dan perangkat lunak (*software*), merancang sistem informasi baru, membangun sistem informasi baru, mengkomunikasikan dan mengimplementasikan sistem informasi baru, memelihara dan melakukan perbaikan/peningkatan sistem informasi baru bila diperlukan.

Dalam sebuah siklus SDLC, terdapat lima langkah. Jumlah langkah SDLC pada referensi lain mungkin berbeda, namun secara umum adalah sama. Langkah tersebut adalah sebagai berikut :

1. Perancangan Sistem

Tujuan perancangan sistem adalah untuk menentukan dan mendefinisikan sistem informasi apa yang akan dikembangkan sehingga dapat memberikan keuntungan dan nilai bagi kegiatan bisnis secara keseluruhan.

2. Analisa Sistem

Analisa sistem dapat didefinisikan sebagai pengguna dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikan. Tahap ini merupakan tahap yang kritis dan penting karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap berikutnya. Langkah-langkah dasar yang harus dilaksanakan oleh analis sistem yaitu :

1. Mengidentifikasi masalah
2. Memahami sistem yang membatasinya
3. Alternatif - alternatif apa saja yang ada untuk mencapai sasaran dan untuk memodifikasi atau mengubah sistem
 - a. Pilih satu dari alternatif yang telah diidentifikasi pada tahap sebelumnya
 - b. Implementasikan alternatif yang dipilih
 - c. Evaluasi masalah dari perubahan yang kita buat dalam sistem

3. Rancangan Sistem

Alternatif yang telah dipilih dalam langkah analisa sistem merupakan dasar dari rancangan sistem. Rancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Tahap ini menyangkut

konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak sistem sehingga setelah menginstalasi sistem akan benar-benar akan memuaskan spesifikasi sistem yang telah ditetapkan pada akhir analisa sistem.

4. Implementasi Sistem

Tahap dari implementasi sistem adalah :

- a. Membangun dan menguji jaringan *database*
- a. Membangun dan menguji program
- c. Instalasi dan menguji sistem yang baru
- d. Penyerahan sistem yang telah dibuat

5. Perawatan dan Pengembangan Sistem

Diperlukan adanya kegiatan tambahan setelah sistem yang baru dijalankan, seperti merawat dan menjaga agar sistem tetap berjalan sesuai dengan apa yang dikehendaki. Perlu juga diperhatikan akibat adanya kebijaksanaan yang baru yaitu perubahan-perubahan prosedur, agar sistem tetap menjalankan fungsinya sehingga pengembangan sistem diperlukan.

II.4.2 Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowchart menolong analisis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil serta menolong dalam menganalisis alternatif-alternatif dalam pengoperasian.

System flowchart adalah urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.

Program flowchart adalah suatu bagian dengan symbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses lainnya dalam suatu program

Pedoman-pedoman dalam membuat flowchart, ada beberapa petunjuk yang harus diperhatikan seperti:

- a. Flowchart digunakan dari atas ke bawah dan dari kiri kekanan
- b. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan didefinisi ini harus dapat dimengerti oleh pembacanya.
- c. Kapan aktivitas dimulai dan berakhir harus secara jelas.
- d. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja,
- e. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
- f. Lingkup dan range dari aktivitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang

digambarkan tidak perlu digambarkan pada flowchart yang sama. Symbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya jika percabangannya tidak berkaitan dengan system

1. Defenisi *Flowchart*

Flowchart merupakan bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

Tujuan Membuat *Flowchart* :

- a. Menggambarkan suatu tahapan penyelesaian masalah
- b. Secara sederhana, terurai, rapi dan jelas
- c. Menggunakan simbol-simbol standar

2. Simbol *Flowchart*

Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok :

a. *Flow direction symbols*

Digunakan untuk menghubungkan simbol satu dengan yang lain disebut juga *connecting line*

b. *Processing symbols*

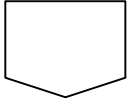
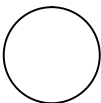
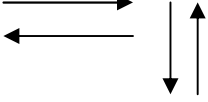
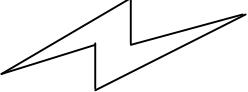
Menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur

c. *Input / Output symbols*



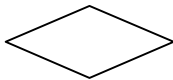
Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.


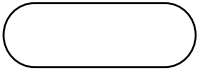

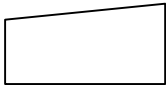
Simbol – symbol yang digunakan dapat dilihat pada tabel II.1

Tabel II.1. *Flow direction symbols*


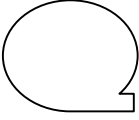

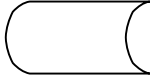
	<p>Symbol Off-line Connector (Simbol untuk keluar/masuk <i>procedure</i> atau proses dalam lembar/halaman yang lain)</p>
	<p>Symbol Connector (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama)</p>
	<p>Simbol arus / <i>flow</i>, yaitu menyatakan jalannya arus suatu proses</p>
	<p>Simbol <i>communication link</i>, yaitu menyatakan transmisi data dari satu lokasi ke lokasi lain</p>

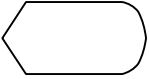
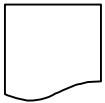
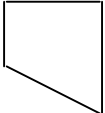
Processing symbols

	<p><i>Symbol Process</i> (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)</p>
	<p>Symbol Manual Operation (Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer)</p>
	<p><i>Symbol Decision</i> (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi)</p>

	<p><i>Symbol Predefined Process</i> (Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>)</p>
	<p><i>Symbol Terminal</i> (Simbol untuk permulaan atau akhir dari suatu program)</p>
	<p><i>Symbol Off-line Storage</i> (Simbol yang menunjukkan bahwa data di dalam simbol ini akan disimpan)</p>
	<p><i>Symbol Manual Input</i> (Simbol untuk pemasukan data secara <i>manual on-line keyboard</i>)</p>

Input / Output symbols

	<p><i>Symbol input-output</i> (Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya)</p>
	<p><i>Symbol magnetig-tape unit</i> (Simbol yang menyatakan input berasal pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i>)</p>
	<p><i>Symbol punched card</i> (Simbol yang menyatakan input berasal dari kartu atau output ditulis ke kartu)</p>
	<p><i>Symbol disk and on-line storage</i> (Simbol untuk menyatakan input berasal dari disk atau <i>output</i> disimpan ke <i>disk</i>)</p>

	<p><i>Symbol display</i> (Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, <i>plotter</i>, <i>printer</i>, dan sebagainya)</p>
	<p><i>Symbol document</i> (simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas)</p>
	<p><i>Symbol transmittal tape</i> (Simbol untuk menyatakan <i>input</i> berasal dari mesin jumlah/hitung)</p>

Dalam penulisan Flowchart dikenal dua model, yaitu Sistem *Flowchart* dan Program *Flowchart*.

1. Sistem *Flowchart*

Bagan Yang memperlihatkan urutan prosedur dan proses dari beberapa file di dalam media tertentu. Melalui *flowchart* ini terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Selain itu juga menggambarkan *file* yang dipakai sebagai input dan output, tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, dan hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.

2. Program *Flowchart*

Bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Dua jenis metode penggambaran program *flowchart* :

- a. *Conceptual flowchart*, menggambarkan alur pemecahan masalah secara global
 - b. *Detail flowchart*, menggambarkan alur pemecahan masalah secara rinci.
- (sumber: <http://ilmukomputer.org/wp-content/uploads/2009/06/anharku-flowchart.pdf>. Di akses pada tanggal 16 Nov 2012).

II.5. Sejarah Java

Java lahir karena ketidakpuasan seorang insinyur di SUN *Micro System* bernama James Gosling. Ia tidak puas dengan *compiler* C++ (yang ia gunakan untuk membuat *software* yang *di-embed* pada peralatan elektronik) karena dinilai terlalu banyak menghasilkan *bug*, berbiaya besar, sangat bergantung pada *platform*. Gosling merasa perlu membuat *kompiler* baru sebagai solusi terhadap sejumlah kelemahan C++ tersebut.

kompiler tersebut diberi nama dengan Oak. *kompiler* ini mirip dengan C++ tapi dengan sejumlah pengurangan fitur yang dianggap kurang menguntungkan dalam pengembangan, seperti *multiple inheritance*, konversi tipe secara otomatis, penggunaan *pointer* dan *magemen memori*.

Pada tahuun 1994 Oak diubah namanya menjadi *java*. Pada era ini, *Java* divisikan sebagai bahasa yang memiliki dukungan baik terhadap *web*.

II.6. Bahasa Pemrograman Java

Program java adalah sekumpulan perintah-perintah (instruction) yang diatur secara sistematis untuk menyelesaikan suatu masalah. Salahsatu bahasa program

komputer adalah bahasa pemrograman java. Bahasa pemrograman java adalah bahasa pemrograman berorientasi objek (PBO) atau Objek Oriented Programming (OOP).

Java bersifat netral, tidak bergantung pada suatu platform, dan mengikuti prinsip WORA (Write Once and Run Anywhere). Beberapa bahasa pemrograman seperti basic, Fortran, Pascal, COBOL, C, Prolog, dan yang lainnya harus bergantung pada suatu platform karena mengharuskan anda mengkompilasikan program tersebut pada setiap sistem operasi yang ditentukan jika ingin menggunakan hasil program tersebut.

Java merupakan bahasa pemrograman yang dikembangkan dengan menggunakan bahasa C++, sehingga sintaknya tidak jauh berbeda dengan bahasa C++. Bagi programmer yang terbiasa menggunakan bahasa C++ tidak mengalami kesulitan untuk beralih ke Java. Sifat-sifat dari bahasa pemrograman Java yakni:

1. Java berorientasi Objek

Pemrograman Java merupakan pemrograman yang berorientasi objek. Artinya, penulisan program harus dibuat dalam bentuk objek-objek, kemudian memodelkan sifat masing-masing objek dalam program. Java mengatur dan menentukan interaksi antara objek.

2. Java bersifat Terdistribusi

Program Java sesuai dengan perkembangan teknologi saat ini, yaitu teknologi jaringan lokal seperti jaringan internet. Dalam hal ini Java dapat digunakan untuk mendistribusikan data atau informasi dalam workstation client, e-mail server, database server, dan webserver dalam jaringan lokal dan internet.

3. Java bersifat MultiPlatform

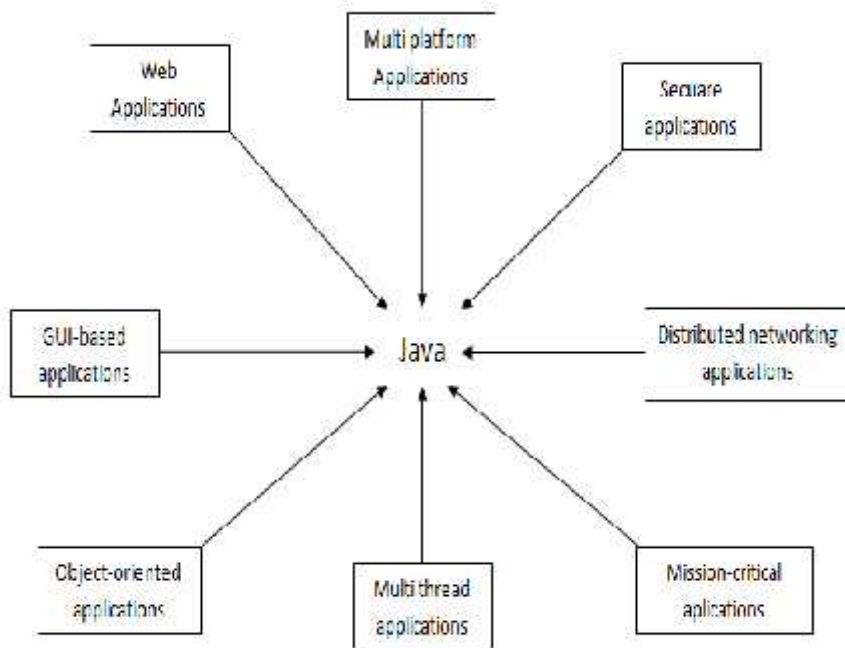
Program Java yang sudah di terjemahkan(dikompilasi) dengan benar dapat dijalankan pada platform yang lain atau sistem operasi yang lain. Java menggunakan prinsip WORA(Write Once and Run Anywhere). Artinya, sekali menulis program dalam bahasa Java dan mengkompilasinya maka program tersebut dapat dijalankan pada berbagai sistem operasi, seperti DOS, Windows, Linux, MacOS dan Solaris.

4. Java bersifat Multithread

Thread merupakan unit dasar penggunaan Central Processing Unit(CPU) yang terdiri dari Thread_ID, program counter, register set dan stack. Multithread adalah banyaknya pekerjaan yang dapat dikerjakan dalam satuan waktu yang hampir bersamaan. Jika webserver hanya memiliki satu thread, berarti client yang lain harus menunggu. Oleh karena itu, diperlukan multithread untuk dapat melayani client server.

II.7 Mengenal Teknologi Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas platform, dikembangkan oleh SUN Micro System dengan sejumlah keunggulan yang memungkinkan java dijadikan sebagai bahasa pemrograman enterprise. Gambar keunggulan - keunggulan java dapat dilihat pada gambar II.1



Gambar II.1 keunggulan – keunggulan pada java.

Berikut kita perjelas keunggulan – keunggulan java yang tampak pada gambar diatas.

Berbasis GUI

Kita bias membuat tampilan berbasis grafik (Grafik user interface/GUI) untuk memudahkan pemakai berinteraksi dengan pemrograman.

.Berorientasi Objek

Konsep pemrograman berorientasi objek tak lain dirancang agar dapat memandang pemrograman sebagai suatu kehidupan nyata. Ini membuat pengembangan software menjadi lebih mudah karena seolah-olah berhubungan dengan kehidupan nyata, bukan lagi kehidupan dunia lain. Nah, java merupakan

salah satu bahasa yang memiliki dukungan penuh terhadap konsep pemrograman berorientasi objek ini.

Aplikasi Web

Saat ini web merupakan sarana yang tidak dipisahkan dari IT. Mungkin ini disebabkan fitur web sebagai sarana komunikasi dihubungkan dengan konsep bahwa manusia itu adalah makhluk sosial yang perlu berkomunikasi. Java merupakan bahasa pemrograman yang memiliki dukungan sangat baik terhadap aplikasi web. Hal ini wajar sebab memang pada awalnya java dilahirkan sebagai solusi untuk menjawab kebutuhan bahasa pemrograman yang mendukung aplikasi berbasis jaringan .

Diantara teknologi java yang mendukung aplikasi web adalah Applet,JSP,(untuk aplikasi web berbasis server), CORBA(untuk aplikasi terdistribusi), dan lain-lain. Lebih dari itu, java memiliki SDK (software Developmen Kit) yang ditujukan untuk mendukung aplikasi enterprise, yang identik dengan aplikasi jaringan, yaitu J2EE(Java2 Enterprise Edition).

Multiplatform

Jika membuat pemrogram dengan bahasa pemrograman seperti C, C++ atau pascal, program yang di kompilasi hanya bias berjalan di satu flatform saja. Jika dikompilasi di windows, program itu hanya bisa dijalankan pada windows. Jika dikompilasi pada Linux, maka hanya bisa dijalankan di linux. Jika ingin menjalankan diatas windows, kemudian suatu waktu ingin menjalankannya pada Linux, harus mati-matian untuk menyesuaikan kode program itu.

Dengan java, hal itu tidak perlu terjadi. Sekali membuat program, maka bisa dijalankan pada komputer dengan platform yang berbeda, asalkan JVM telah terinstall untuk platform itu. Bila membuat program java pada windows, maka java bisa dijalankan pada Linux, Unix, dan lain-lain.

Keamanan

Salah satu alasan mengapa java dapat digunakan untuk menangani kebutuhan aplikasi enterprise adalah kemampuannya dalam soal keamanan. Fitur keamanan java ada dua paket, yaitu pada JDK dan pada java cryptography Extension (JCE). Fitur yang tersedia pada paket itu meliputi signature (untuk menandatangani dokumen), message digest, pembangkitan kunci, autentikasi, enkripsi dan bilangan besar (Big Number).

Bisa Digunakan pada Aplikasi Jaringan Terdistribusi

Aplikasi yang berjalan pada jaringan terdistribusi melibatkan sejumlah komputer yang berkomunikasi secara transparan, yaitu seolah sejumlah komputer itu merupakan sebuah komputer yang bersatu untuk menjalankan sebuah perintah. Dengan fitur ini bisa membuat program untuk menyelesaikan sebuah kasus tertentu dengan melibatkan beberapa komputer. Ini akan membuat penyelesaian masalah menjadi lebih cepat.

Mendukung Software Mission-Critical

Java bisa digunakan untuk pengembangan software dimana tingkat eror yang terjadi sangat diperhatikan, dimana sebuah eror pada eksekusi program bisa mengakibatkan kerusakan fatal pada system secara keseluruhan. sebagai contoh bisa dibayangkan apa yang akan terjadi jika software pengendali pesawat terbang rusak. Sangat fatal akibatnya bukan?.

Sebagai dukungan terhadap software seperti ini, java menghilangkan sebuah fitur C++ yang berpotensi menghasilkan error yang fatal, seperti pointer, konversi tipe tanpa dicek, dan lain-lain.

Multithread

Fitur multithread digunakan untuk menjalankan sejumlah proses secara bersamaan. Dengan menggunakan fasilitas ini anda bisa membuat program java yang menjalankan beberapa perintah sekaligus, sehingga tidak perlu menunggu sebuah perintah selesai dikerjakan untuk mengerjakan perintah lain.

(Rijalul Fikri, Ipam Fuadina Adam, Imam Prakoso 2005 hal 15)

II.8 Pengenalan NetBeans

NetBeans IDE adalah *IDE 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*.

Versi terbaru saat ini adalah *NetBeans IDE 0.7* yang dirilis Mei 2010 mengembangkan fitur-fitur *Java EE* yang sudah ada (termasuk *Java Persistence support*, EJB-3 dan JAX-WS). Sementara paket tambahannya, *NetBeans Enterprise Pack* mendukung pengembangan aplikasi perusahaan *Java EE 5*, meliputi alat desain *visual SOA*, skema XML, *web service* dan pemodelan UML. *NetBeans C/C++ Pack* mendukung proyek C/C++.

Modularitas: Semua fungsi *IDE* disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman *Java*, editing, atau dukungan bagi CVS. *NetBeans* memuat semua modul yang diperlukan dalam pengembangan *Java* dalam sekali *download*, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengizinkan *NetBeans* untuk bisa dikembangkan. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, *Sun Studio*, *Sun Java Studio Enterprise*, dan *Sun Java Studio Creator* dari *Sun Microsystems* semuanya berbasis *NetBeans IDE*.

Untuk membuat dialog atau user-interface, kita tidak perlu membuat teks program secara manual baris per baris, tetapi cukup klik pada component-palette. Teks program akan dihasilkan secara otomatis. *NetBeans* mencakup compiler atau builder, dan debugger internal. Hal ini sangat memudahkan proses paska perancangan program. Proses deployment dan atau tanpa tes dapat juga dilakukan dari dalam *NetBeans*.**8**

(Rijalul Fikri, Ipam Fuadina Adam, Imam Prakoso 2005 hal 19)

II.9 IDE NetBeans

Di dalam NetBeans, semua perancangan dan pemrograman dilakukan didalam kerangka sebuah proyek. Proyek NetBeans merupakan sekumpulan file yang dikelompokan didalam satu kesatuan.

Pada umumnya sebuah proyek, beserta file-file isinya, akan ditempatkan didalam suatu direktori. Meskipun demikian ada juga proyek yang dapat menangani file-file yang terpisah direktorinya.

Apakah file-file ini disatukan atau terpisah direktorinya, tergantung dari jenis proyeknya.ada proyek yang standar dan ada proyek yang bebas (*free-form*).

proyek yang standar memakai skrip Ant (otomatis dibuat oleh NetBeans),sedangkan proyek *free form* memakai skrip Ant yang dapat dibuat sendiri.

Yang dimaksud dengan script Ant disini adalah jakarta Ant. Ant dibangun oleh apache software foundation. (<http://ant.apache.org>). Ant dipakai untuk otomatisasi tugas-tugas yang rutin yang dilakukan oleh developer, seperti kompilasi, tes, debug, *running* dan *packaging*. Disini Ant dipakai untuk memodifikasi proses build dan *run* dari *proyek*. jakarta Ant sebagai *build-tools* pada saatini boleh dikatakan dominan dalam lingkungan java/XML.

Ant dibangun dengan java sehingga memiliki keuntungan bersifat *multuplatform*.sedangkan skrip Ant sendiri ditulis dalam format XML.

Jika tidak tertarik maka kita tidakperlu tahu mengenai skrip Ant untuk membuat aplikasi java dalam NetBeans. Telah tersedia bermacam-macam template standar

yang disajikan dalam bentuk wizard, yang akan memudahkan kita untuk membuat proyek java tanpa harus ditemani Ant.

Ant dapat melakukan modifikasi misalnya menentukan target Ant. Atau membuat sendiri skrip Ant untuk menentukan spesifikasi proyek yang anda inginkan.

Jadi skript Ant sama sekali bukan merupakan suatu keharusan untuk dipahami dalam pemrograman NetBeans. Tanpa memahami Ant sama sekali, kita dapat membuat aplikasi java seutuhnya.

(G. Sri Hartati, Bherry Suharto, M.Soesilo Wijono 2007 hal 9).