BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Pengertian suatu sistem tentu mempunyai beberapa persyaratan umum, persyaratan umum tersebut adalah bahwa sistem harus mempunyai unsur lingkungan, interaksi unsur dengan suatu tujuan yang akan dicapai.

Berikut ini pengertian sistem menurut beberapa ahli:

Menurut *Raymond McLeod* (2004 : 9) yang dimaksud dengan sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

Menurut *Jogiyant*o (2000 : 683) yang dimaksud dengan sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Menurut (Davis, 1985) yang dimaksud dengan sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud.

Menurut (Lucas, 1989) yang dimaksud dengan sistem sebagai suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain dan terpacu".

2.1.1. Elemen Sistem

Sebuah sistem mempunyai tujuan atau sasaran. Sistem mempunyai elemen-elemen penyusunan diantaranya adalah :

1. Tujuan

Sasaran atau tujuan dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan yang akan dihasilkan sistem, karena suatu sistem akan dikatakan berhasil apabila mengenai sasaran dan tujuan.

2. Batasan

Dalam mencapai tujuan dari sistem dibutuhkan batasan –batasan suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Kontrol

Merupakan pengawasan dari pelaksanaan pencapaian tujuan sistem yang dapat berupa asal masukan, frekuensi pemasukkan data, jenis pemasukkan dan lain-lain.

2.1.2 Karakteristik Sistem

Karakteristik sistem menurut Jogiyanto (2005:3) adalah untuk mencapai tujuannya, suatu sistem harus memiliki sifat-sifat tertentu atau suatu karakteristik sebagai berikut :

1. Komponen *(Components)*

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama membentuk satu kesatuan.

2. Batas sistem *(Boundary)*

Merupakan daerah yang membatasi antara satu sistem dengan sistem lainnya atau dengan lingkungan luar.

3. Lingkungan Luar Sistem *(Environments)*

Merupakan segala sesuatu yang berada diluar batas sistem yang bisa mempengaruhi operasi sistem baik itu yang bersifat merugikan ataupun menguntungkan.

4. Penghubung *(Interface)*

Merupakan media penghubung antar subsistem yang memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan *(Input)*

Merupakan energy yang dimasukan kedalam sistem, yang dapat berupa masukan perawatan *(Maintenance Input)* dan masukan sinyal *(Signal Input)*.

6. Keluaran *(Output)*

Merupakan hasil dari energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna dari sisa pembuangan.

7. Pengolah Sistem *(Process)*

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

8. Sasaran *(Objective)* atau Tujuan *(Goal)*

Suatu sistem pasti mempunyai tujuan (goal), jika suatu sistem tidak mempunyai tujuan yang jelas, maka semua operasi sistem tidak ada gunanya.

2.2. Konsep Dasar Informasi

Informasi sangat penting dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh dan berakhir. Menurut Jogiyanto (2001 : 8) yang dimaksud dengan informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya.

2.2.1. Siklus Informasi

Siklus informasi untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi.

2.2.2. Kualitas dan Nilai Informasi

Menurut Jogiyanto (2001 : 10), kualitas dari suatu informasi (*quality ofinformation*) tergantung dari tiga hal, yaitu informasi yang harus akurat (*accurate),* tepat pada waktunya (*timeliness*) dan relevan (*relevance*).

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat pada waktunya, bearti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah using tidak akan mempunyai nilai lagi.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap –tiap orang satu dengan yang lainnya berbeda.

2.3. Pengertian Sistem Informasi

Setelah diketahui bahwa informasi merupakan sangat penting bagi suatu organisasi atau manajemen untuk mengambil keputusan, maka dapat didefinisikan sebagai berikut :

a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam berorganisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan atau untuk mengendalikan informasi.
c. Suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.4. Kosep Dasar Sistem Informasi

Menurut Jogiyanto (2001 : 11) yang dimaksud Sistem Informasi adalah suatu sistem yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi terdiri dari beberapa komponen, antara lain :

a. *Hardware* : CPU, disk, terminal, printer.
b. *Software* : sistem operasi, Sistem Basis Data, program pengontrol komunikasi, program aplikasi

Personal : yang mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem. Data : data yang tersimpan dalam jangka waktu tertentu.

2.5 Java

2.5.1 Pengenalan Java Language

Bahasa Java dikembangkan oleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang-barang elektronik seperti televisi, *VCR*, *toaster* dan mesin - mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program-program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program - program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.

Sekarang, *Sun* telah mengeluarkan berbagai program Java yang dapat digunakan seperti *Java API,* atau JDK atau *JAVA Developer Kit* . Selain itu, banyak juga program-program lain yang dapat digunakan untuk membuat program Java, seperti *NetBeans, JBuilder*, *JCreator, J++*, dan sebagainya. Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*

3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas.

J2SE diutamakan untuk membuat java *Applet* dan aplikasi yang *stand alone*. J2EE digunakan untuk membuat sistem *enterprise*, dimana sebuah aplikasi dapat disebarkan pada jaringan computer dan dapat diakses dari jauh. J2EE mengandung semua fungsi J2SE dan diutamakan untuk membuat aplikasi server. Pada aplikasi ini penulis akan memakai platform J2SE.

2.5.2 Sejarah Java

Bahasa pemrograman Java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh *Patrick Naughton*, beserta sembilan programmer lainnya dari *Sun Microsystem*.Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (Star Seven).Setelah era *Star Seven* selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek *The Green Project*.

Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer. Pada

sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Sayang terjadi perpecahan di antara mereka suatu hari. Tiga dari pimpinan utama proyek,dari *Sun Microsystems* bersama Marc Andreessen, membentuk *Netscape*. Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "bapak java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java".

2.6. Kelebihan Dan Kekurangan Java

2.6.1. Kelebihan Java

1. Kelebihan utama dari *Java* ialah dapat dijalankan di beberapa *platform* /sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja.Dengan kelebihan ini pemrogram cukup menulis sebuah program *Java* dandikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin /*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpaperubahan. Kelebihan ini memungkinkan sebuah program berbasis *Java* dikerjakan diatas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah *Microsoft Windows, Linux, Mac OS*, dan *Sun Solari*. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs *Java*) untuk menginterpretasikan *bytecode* tersebut.
2. Semua aspek yang terdapat di *Java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data

diturunkan dari kelas dasar yang disebut 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 pemograman termudah, bahkan untuk fungsi fungsi yang *advance* seperti komunikasi antara komputer sekalipun.

3. Perpustakaan yang lengkap *Java* terkenal dengan kelengkapan *library* (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi
4. Bergaya C++ Memiliki sintaks seperti bahasa pemrograman [C++] sehingga menarik banyak pemrogram C++ untuk pindah ke *Java*. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke *Java*.

2.6.2 Kekurangan Java

1. Dekompilasi adalah proses membalikan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode di *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 atau *direverse-engineer*.

2. Penggunaan memori untuk program berbasis *Java* jauh lebih besar dari pada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.7. UML (Unified Modeling Language)

UML (Unified Modeling Language)adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.7.1 Bagian – Bagian UML

Bagian utama dari UML adalah view, diagram, model element, dan general mechanism.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antara lain: use case view, logical view, component view, concurrency view,dan deployment view.

b. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya. View ini digambarkan dalam use case diagram dan kadang-kadang dengan activity diagrams. Viewini digunakan terutama untuk pelanggan, perancang (designer), pengembang (developer), dan penguji sistem (tester).

c. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object,danrelationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. View ini digambarkan dalam class diagrams untuk struktur statis dan dalam state, sequence, collaboration, dan activity diagram untuk model dinamisnya. View ini digunakan untuk perancang (designer) dan pengembang (developer).

d. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).

e. Concurrency view

Membagi sistem ke dalam proses dan prosesor.View ini digambarkan dalam diagram dinamis (state, sequence, collaboration, dan activity diagrams) dan diagram implementasi (component dan deployment diagrams) serta digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

f. Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (nodes) dan bagaimana hubungannya dengan lainnya. View ini digambarkan dalam deployment diagramsdan digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

g. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Use casemerupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

3. Component Diagram

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. Komponent merupakan implementasi software dari sebuah atau lebih class. Komponent dapat berupa source code, komponent biner, atau executable component. Sebuah komponent berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.Sehingga component diagram merepresentasikan dunia riil yaitu component software yang mengandung component, interface dan relationship.

4. Deployment Diagram

Menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, executeable component dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

5. State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh stateyang berbeda.

6. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antaraobject, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequence diagrams. Dalam menunjukkan pertukaran pesan, collaboration diagram menggambarkan object dan hubungannya (mengacu ke konteks) . Jika penekannya pada waktu atau

urutan gunakan sequence diagrams, tapi jika penekanannya pada konteks gunakan collaboration diagram.

8. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use caseatau interaksi.

Tujuan Penggunaan UML

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. Memberikan model yang siap pakai, bahsa pemodelan visual yang
5. UML bisa juga berfungsi sebagai sebuah (blue print) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (reserve enginering).

Perangkat lunak yang mendukung pembuatan diagaram UML

1. StarUML

   StarUML adalah sebuah proyek open source untuk mengembangkan cepat, fleksibel, extensible, featureful, dan bebas-tersedia UML / platform MDA berjalan pada platform Win32.Tujuan dari proyek StarUML adalah untuk

membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat UML komersial seperti Rational Rose, Bersama dan sebagainya

2. Acceleo

Acceleo adalah generator kode yang mengubah model menjadi kode. Acceleo mudah digunakan dan menyediakan "dari rak" generator (Jee,. Bersih, Php …) dan template editor untuk Eclipse.

3. Cargo UML

Cargo UML adalah open source UML modeling tool terkemuka dan termasuk dukungan untuk semua diagram UML standar 1,4. Ini berjalan pada setiap platform Java dan tersedia dalam bahasa sepuluh. ArgoUML ditulis seluruhnya di Jawa dan menggunakan Java Kelas Foundation.Hal ini memungkinkan ArgoUML untuk berjalan di hampir semua platform

2.8. Neatbeans

Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger.



Gambar 2.1 Netbeans

Netbeans juga digunakan oleh sang programmer untuk menulis, meng-compile, mencari kesalahan dan menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional desktop, enterprise, web, and mobile applications dengan Java language, C/C++, dan bahkan dynamic languages seperti PHP, JavaScript, Groovy, dan ruby.

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun netbeans memiliki 2 produk yaitu Platform Netbeans dan Netbeans IDE. Platform Netbeans merupakan framework yang dapat digunakan kembali (reusable) untuk menyederhanakan pengembangan aplikasi deskto dan Platform NetBeans juga menawarkan layanan-layanan yang umum bagi aplikasi.

2.8.1 Fitur – Fitur Netbeans

Fitur-fitur dari Platform Netbeans antara lain:

- Manajemen antarmuka (misal: menu & toolbar)
- Manajemen pengaturan pengguna
- Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
- Manajemen jendela
- Wizard framework (mendukung dialog langkah demi langkah)

Netbeans IDE merupakan sebuah 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 5.5.1 yang dirilis Mei 2007 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. 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 mengijinkan 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 Microsystem semuanya berbasis NetBeans IDE.

Fitur fitur yang terdapat dalam netbeans antara lain:

1. Smart Code Completion: untuk mengusulkan nama variabel dari suatu tipe, melengkapi keyword dan mengusulkan tipe parameter dari sebuah method.
2. Bookmarking: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. Go to commands: fitur yang digunakan untuk jump ke deklarasi variabel, source code atau file yang ada pada project yang sama.
4. Code generator: jika kita menggunakan fitur ini kita dapat meng-generate constructor, setter and getter method dan yang lainnya.
5. Error stripe: fitur yang akan menandai baris yang eror dengan memberi highlight merah.