BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Adapun pengertian definisi sistem, karakteristik sistem dan klasifikasi sistem sebagai berikut:

2.1.1 Definisi Sistem

Menurut Jogiyanto (2005 : 1) “Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.”

Pada umumnya sistem merupakan kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan.

2.1.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, menurut Jogiyanto (2005 :1) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi antara lain sebagai berikut :

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berintegrasi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa

suatu subsistem atau bagian-bagian dari sistem. Setiap subsitem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruahan.

2. Batasan Sistem

   Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem.

3. Lingkungan Luar Sistem

   Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsitem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsitem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

   Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem diproses dan akhirnya dikeluarkan berupa informasi yang dibutuhkan.

6. Keluaran Sistem

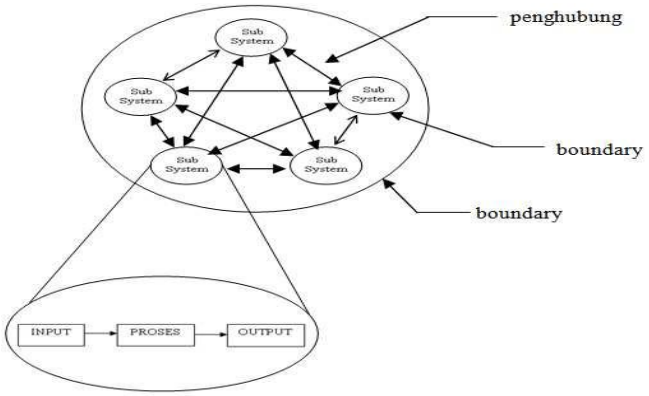
   Keluaran (*output*) adalah hasil dari energi yang diolah dan di diklasifikasikan menjadi informasi yang berguna.

7. Pengolahan Sistem

   Pengolahan sistem merupakan suatu bagian yang mengolah masukan (*input*) dan memprosesnya agar menjadi *output* informasi yang berguna.

8. Sasaran Sistem

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Jika suatu tidak mempunyai sasaran maka operasi sistem tidak akan berguna. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem. Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya.



Gambar 2.1 Karakteristik Sistem

Sumber : Jogiyanto (2005 : 6)

2.1.3. Klasifikasi Sistem

Sistem juga dapat diklasifikasikan dari beberapa sudut pandang, klasifikasi sistem menurut Jogiyanto (2004 : 687) diantaranya adalah sebagai berikut :

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alami, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia, sistem ini melibatkan interaksi antara manusia dengan mesin.

3. Sistem Tertentu dan Sistem Tak Tentu

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi antara bagian-bagiannya dapat dideteksi dengan pasti. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Sistem Terbuka

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhungan dan terpanguruh dengan lingkungan luarnya.

2.2. Pengertian Informasi

Adapun pengertian definisi informasi, kualitas informasi, nilai informasi, dan siklus informasi sebagai berikut:

2.2.1. Definisi Informasi

Dalam kehidupan sehari-hari sebuah informasi yang dapat diartikan sebagai data yang memiliki nilai guna. Data tersebut merupakan nilai, keadaan, dan memiliki sifat berdiri sendri lepas dari konteks apapun.

Menurut Jogiyanto (2005 : 8) “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.”

2.2.2. Kualitas Informasi

Kualitas dari suatu informasi tergantung pada tiga hal pokok yaitu:

a. Akurat (*accurarcy*)

   Informasi bisa dikatakan berkualitas apabila seluruh kebutuhan informasi telah tersampaikan (*complenteness*), semua pesan benar/sesuai (*corretness*), serta pesan yang disampikan sesuai dengan sistem yang diinginkan oleh *user*.

b. Tepat waktu (*timeliness*)

   Berbagai proses dapat diselesaikan dengan tepat waktu, laporan yang dibutuhkan dapat disampaikan tepat waktu.

c. Relevan (*relevancy*)

Informasi harus mempunyai manfaat untuk pemakainya, dimana relevansi informasi untuk individu berbeda tergantung pada yang membutuhkan.

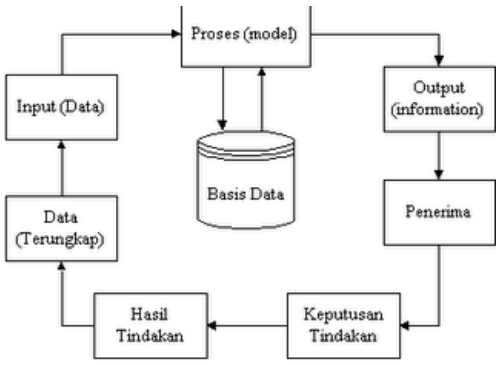
2.2.3. Nilai Informasi

Nilai informasi (*value of information*) ditentukan oleh dua hal yaitu manfaat dan biaya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem umumnya digunakan untuk beberapa keguanaan sehingga tidak mungkin atau sulit untuk menghubungkan antara informasi tentang suatu masalah dengan biaya untuk memperolehnya, karena sebagain besar informasi digunakan tidak hanya satu pihak saja didalam perusahaan.

2.2.4. Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk informasi, maka perlu diolah melalui suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus ini disebut dengan siklus

informasi (*information cycle*) atau disebut juga siklus pengolahan data (*processing cycles*).



Gambar 2.2 Siklus Informasi

Sumber : Tata Sutabri (2007)

2.3 Pengertian Sistem Informasi

Adapun pengertian sistem informasi, komponen sistem informasi, perancangan sistem informasi, pengolahan sistem informasi, pengendalian sistem informasi dan penilaian sistem informasi sebagai berikut:

2.3.1 Pengertian Sistem Informasi

Menurut Robert A. Leitch dan K. Roscoe Davis dalam buku Jogiyanto (2005:11) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari waktu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut Hall dalam buku Abdul Kodir (2003:11) sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokan, diproses menjadi informasi, dan didistribusikan kepada pemakai.

Dari kedua pengertian tersebut, dapat di simpulkan bahwa sistem informasi adalah perangkat unsur yang secara teratur saling berkaitan dengan tujuan untuk mengelola data sehingga menghasilkan informasi yang berguna. Sistem informasi juga mempunyai beberapa komponen, yaitu :

1. *Hardware*
2. *Software*
3. *Brainware*
4. Data
5. Prosedur atau metode-metode

2.3.2 Komponen Sistem Informasi

Menurut Jogiyanto (2005:12) Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok, yaitu :

1. Blok Masukan

   Blok masukan merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa dokumen-dokumen dasar yang dapat diolah menjadi suatu informasi tertentu.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data input untuk menghasilkan suatu informasi yang dibutuhkan.

3. Blok Keluaran

   Blok keluaran merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

4. Blok Teknologi

   Blok teknologi merupakan penunjang utama dalam berlangsunganya sistem informasi. Yang memiliki beberapa komponen yaitu alat memasukan data (*input device*), alat untuk menyimpan dan mengakses data (*storege device*), alat untuk menghasilkan dan mengirimkan keluaran (*output divice*) dan alat untuk membantuk pengendalian sistem secara keseluruan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainare*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, agar informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasistas penyimpanannya.

6. Blok Kendali

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal–hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi kesalahan-kesalahan kerusakan dalam penggunaan sistem.

2.3.3 Perencanaan Sistem Informasi

Menurut Tata Sutabri (2003:49) untuk dapat menerapkan sistem informasi yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan, dan evaluasi sesuai dengan keinginan dan nilai masing-masing organisasi.



Gambar 2.3 Tingkatan Perencanaan Sistem Informasi

(Sumber: Tata Sutabri, 2003:50)

2.3.4 Pengolahan Sistem Informasi

Menurut Tata Sutabri (2003:50) pengelolaan sistem informasi adalah bagian yang tak dapat dipisahkan dari studi manajemen sebagaimana pengelolaan ketenagan, keuangan, organisasi, dan tata laksana.

2.3.5 Pengendalian Sistem Informasi

Menurut Tata Sutabri (2003:52) pengendalian sistem informasi merupakan bagian yang tak dapat dipisahkan dari pengelolaan sistem informasi, bahkan ia melaksanakan fungsi yang sangat penting karena mengamati setiap tahapan dalam proses pengelolaan informasi.

2.3.6 Penilaian Sistem Informasi

Menurut Tata Sutabri (2005:54) komponen produk ditandai oleh hasil-hasil yang dicapai oleh sistem yang mutu dan jumlah yang tinggi. Untuk mengetahui hingga mana komponen-komponen telah beroperasi dengan baik sebagaimana yang diharapkan, komponen penilaian pada gilirannya menempati kedudukan dan fungsi yang sangat strategis, bahkan dapat dinilai sebagai sangat menentukan keberhasilan keseluruhan pengelolaan sistem informasi itu sendiri.

2.4 Sistem Informasi Penjualan dan Pembelian

Sistem informasi penjualan dan pembelian merupakan suatu sistem yang terdiri dari kumpulan orang, peralatan dan prosedur yang memadukan antara pekerjaan mesin (komputer) dan manusia yang menyajikan keakuratan informasi bagi para pemakai dalam membuat keputusan untuk memecahkan masalah didalam perusahaan.

2.4.1 Tujuan Sistem Informasi Penjualan dan Pembelian

Sistem informasi penjualan merupakan salah satu dari sistem informasi yang terpenting pada perusahaan umumnya. Sistem informasi penjualan ini bertujuan untuk membantu manajer dalam berbagai hal seperti :

a) Membantu manajemen dalam pengambilan keputusan.
b) Manajemen dapat menerima laporan lebih sering dan terperinci.
c) Manajemen dapat memonitor prestasi produk, pasar, karyawan, penjualan dan berbagai unit pemasaran lainnya.

Sistem informasi penjualan dan pembelian ini sangat berperan di dalam setiap perusahaan, agar aktivitas penjualan yang dilakukan dapat cepat serta akurat diselesaikan dan informasi yang tersaji dapat tepat waktu pada saat dibutuhkan.

2.5 *UML*

Menurut Sri Dharwiyanti dan Romi Satria Wahono, 2003, *Pengantar Unified Modeling Language (UML)*, Kuliah Umum Ilmu Komputer, 2-10 *Unified Modelling Language (UML)* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras,

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

Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax*/semantik. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Abstraksi konsep dasar *UML* yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari diagram. *Main concepts* bisa kita pandang sebagai *term* yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagaram tersebut.

1. *Use Case Diagram*

   Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case*

merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

2. *Class Diagram*

   Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

3. *Statechart Diagram*

   Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

4. *Activity Diagram*

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

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi

memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

7. *Package Diagram*

   Package Diagram utamanya digunakan untuk mengelompokkan elemen diagram *UML* yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. *Package diagram* pada dasarnya adalah *class diagram* yang hanya menampilkan paket, disamping kelas, dan hubungan ketergantungan, disamping hubungan khas yang ditampilkan pada *class diagram*.

8. *Component Diagram*

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

9. *Deployment Diagram*

   Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana

kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya *TCP/IP*) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.6 *Java*

Bahasa *Java* dikembangkan oleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan perangkat lunak bagi *consumer* barang-barang elektronik seperti *televise, 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 *Eclipse, 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*

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

2.7 *Netbeans*

NetBeans adalah *Integrated Development Environment (IDE)* berbasiskan *Java* dari *Sun Microsystems* yang berjalan di atas *Swing*. *Swing* merupakan sebuah teknologi *Java* untuk pengembangan aplikasi *Desktop* yang dapat bejalan di berbagai macam *platforms* seperti *Windows*, *Linux*, *Mac OS X* dan *Solaris*.

Suatu *IDE* adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface (GUI),* suatu *text* atau kode editor, suatu *compiler* atau *interpreter* dan suatu *debugger*.

Netbeans merupakan *software development* yang *Open Source*, dengan kata lain *software* ini di bawah pengembangan bersama, bebas biaya. *NetBeans* merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. *Sun Microsystems* mendirikan proyek kode terbuka *NetBeans* pada bulan Juni 2000 dan terus menjadi sponsor utama.

Saat ini terdapat dua produk : *NetBeans IDE* dan *NetBeans Platform*. The *NetBeans IDE* adalah sebuah lingkungan pengembangan untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. *Netbeans IDE* ditulis dalam *Java*, namun dapat mendukung bahasa pemrograman lain.

Terdapat banyak modul untuk memperluas *Netbeans IDE*. *Netbeans IDE* adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

Tersedia juga *NetBeans Platform* sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi *desktop* yang besar. Mitra *ISV* menyediakan *plug-in* bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam *Platform* dan dapat juga digunakan untuk membuat kakas dan solusi sendiri.

Kedua produk adalah kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan *Lisensi Common Development and Distribution License* (*CDDL*).

2.5 *XAMPP*

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License*, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.Untuk mendapatkanya dapat men-*download* langsung dari *website* resminya.