

BAB II

TINJAUAN PUSTAKA

II.1. Sistem dan Analisa Sistem

II.1.1. Sistem

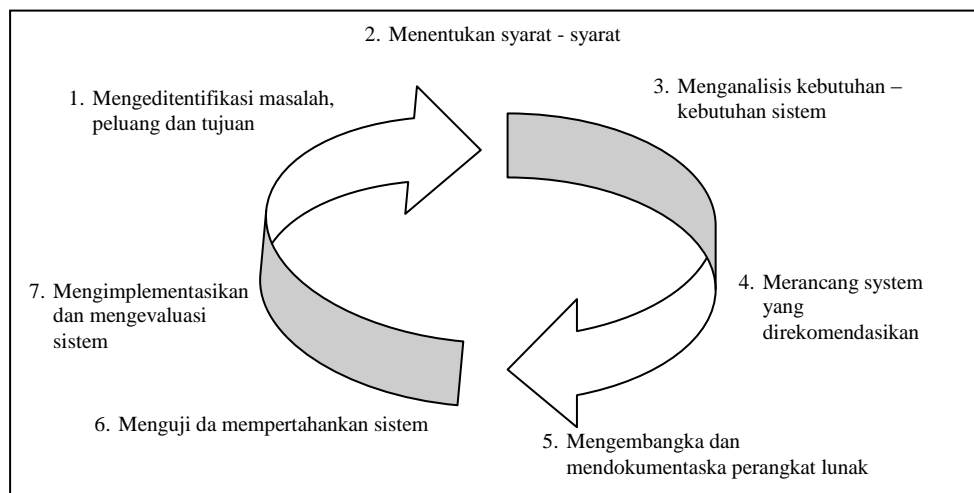
Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas/fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Sebagai contoh, sistem kendaraan terdiri dari: komponen *starter*, komponen pengapian, komponen penggerak, komponen pengerem, komponen kelistrikan-spedometer, lampu dan lain-lain. Komponen-komponen tersebut diatas memiliki tujuan yang sama yaitu untuk membuat kendaraan tersebut bisa dikendarai dengan nyaman dan aman. Contoh lain yaitu sistem perguruan tinggi, yang terdiri dari dosen, mahasiswa, kurikulum, dan lain-lain. Sistem ini bertujuan untuk menghasilkan mahasiswa-mahasiswa yang memiliki kemampuan di bidang ilmunya. (Kusrini; 2008 : 1)

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Menurut Jerry FithGerald : Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. (Kusrini; 2008 :1)

II.1.2. Siklus Hidup Pengembangan Sistem

Siklus hidup pengembangan sistem ini adalah pengembangan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah di kembangkan dengan sangat baik melalui pengembangan siklus kegiatan penganalisisan dan pemakaian secara spesifik (Kusrini; 2008 : 8)

Adapun siklus hidup pengembangan sistem data pada gambar II.1:

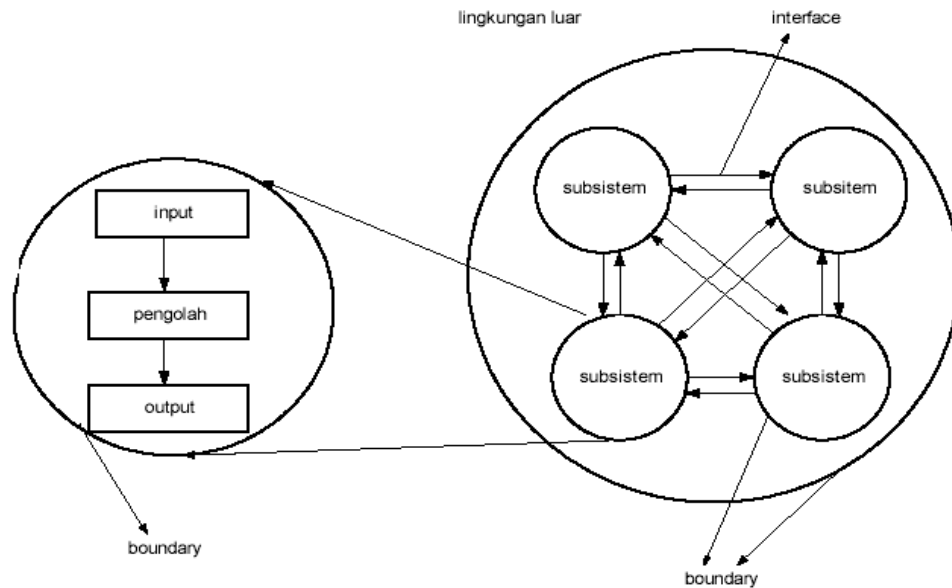


Gambar II.1 Siklus Hidup Pengembangan Sistem

(Sumber: Kusrini; 2008 : 9)

II.1.3. Karakteristik Sistem

Karakteristik sistem dapatlah digambarkan pada gambar II.2 :



Gambar II.2 Karakteristik Sistem

(Sumber: Kusriani; 2008 : 2)

Keterangan Gambar II.2 :

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, 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 sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan/berfungsi sebagaimana

mestinya. Tentunya sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

2. Batas Sistem (*Boundary*)

Batas 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 tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*)

untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supersistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain

menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

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

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Perbedaan suatu sasaran (*objectives*) dan suatu tujuan (*goal*) adalah, *goal* biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem utama, seperti misalnya sistem bisnis perusahaan, maka istilah *goal* lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem lainnya yang merupakan bagian atau subsistem dari sistem bisnis, maka istilah *objectives* yang lebih tepat. Jadi tergantung dari ruang lingkup mana memandang sistem tersebut. Seringkali tujuan (*goal*) dan sasaran (*objectives*) digunakan bergantian dan tidak dibedakan.

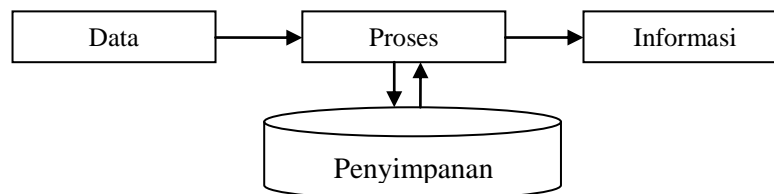
II.1.4. Informasi

Informasi adalah data yang dapat dianalogikan dengan data – data , yang belum di kelola dan harus diolah untuk menjadi informasi yang akurat. (Kusrini; 2008 : 12). Agar informasi yang penulis sajikan lebih bermanfaat maka terlebih dahulu dibuat aliran informasi yang lebih jelas dan lengkap. Berkaitannya dengan

penyedia informasi bagi manajemen dalam mengambil suatu keputusan, yang diperoleh harus berkualitas. Maka kualitas dari informasi tergantung pada:

1. Akurat : akurat berarti bahwa informasi harus bebas dari kesalahan – kesalahan dan tidak biasa (menyesatkan) dan jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerimaan informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi atau merusak informasi tersebut.
2. Relevansi : relevansi berarti bahwa informasi benar – benar berguna bagi suatu tindakan dan keputusan oleh seseorang.
3. Tepat waktu : tepat waktu berarti bahwa informasi yang datang pada penerimaan tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan dalam pengambilan keputusan. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu di dapat,

Untuk lebih jelasnya informasi merupakan hasil atau output dari proses informasi data. Hal ini dapat dilihat seperti gambar II.3 dibawah ini :



Gambar II.3 Proses Data Menjadi Informasi

(Sumber: Kusriani; 2008 : 12)

II.2. Sistem Informasi Geografis

Sistem informasi berbasis pemetaan dan geografis adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkait erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa-peristiwa yang terjadi dipermukaan bumi. Teknologi *GIS* mengintegrasikan operasi pengolahan data berbasis database yang digunakan saat ini pengambilan data berdasarkan kebutuhan, serta analisa statistik dengan menggunakan visualisasi yang khas berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya.

Teknologi Sistem informasi Geografis dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, SIG bisa membantu perencana untuk secara cepat menghitung waktu tanggap darurat saat terjadi bencanaalam, atau SIG dapat digunakan untuk mencari lahan basah (wetlands) yang membutuhkan perlindungan dari polusi. (Eko Budiyanto; 2002 : 5)

II.2.1. Peta Digital

Peta digital (digital map), yaitu peta yang berupa gambaran permukaan bumi yang diolah dengan bantuan media komputer. Data yang diperoleh berupa data digital dan hasil dari gambaran tersebut dapat disimpan dalam suatu media seperti disket, CD, maupun media penyimpanan lainnya, serta dapat ditampilkan kembali pada layar monitor komputer. Biasanya peta digital ini dibuat dengan menggunakan software GIS (Geography Information system). Ilmu yang

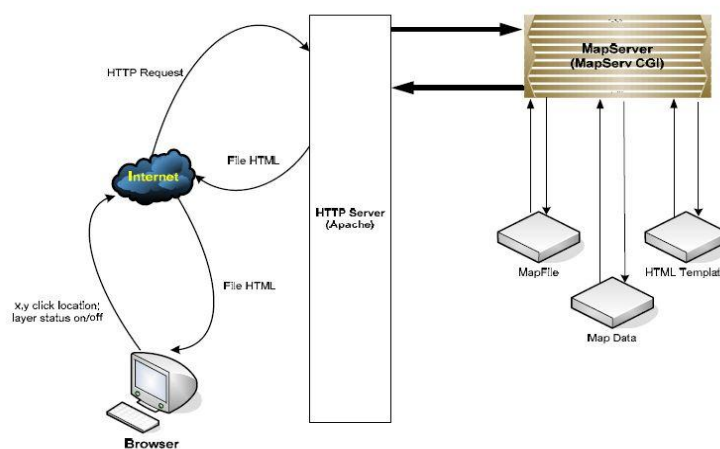
mempelajari tentang peta dan pemetaan disebut dengan kartografi dan orang yang ahli dalam bidang peta dan pemetaan disebut kartograf.

II.2.2. Tahap Analisis Sistem

Metode sekuensial linear atau waterfall mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Berikut ini adalah tahapan-tahapan dari metode waterfall (Sommerville, 2000).

II.2.3. Arsitektur Gis

Secara umum diagram pengembangan sistem dimulai dengan user melakukan action pada web browser yaitu meminta layanan ke web browser yaitu meminta layanan mapserver ke webserver dan diterjemahkan dalam CGI yang diteruskan oleh mapscript untuk meminta Dataset GIS pada program external sehingga didapatkan output yang ditampilkan pada web browser. Sistem aplikasi yang dibangun dengan MapServer memiliki arsitektur yang ditunjukkan pada gambar II.4 :



**Gambar II.4. Arsitektur Sistem
(Sumber: DM.Solution Group)**

II.2.4. ArcView/Map Server

ArcView merupakan salah satu software yang digunakan untuk dalam pengolahan data geografi untuk menampilkan informasi spasial, membuat peta yang telah banyak dijumpai dipasaran. Masing-masing perangkat lunak ini mempunyai kelebihan dan kekurangan dalam menunjang analisis informasi geografi. Salah satu yang sering digunakan saat ini adalah ArcView. ArcView yang merupakan salah satu perangkat lunak Sistem Informasi geografi yang dikeluarkan oleh ESRI (Environmental Systems Research Institute). ArcView dapat melakukan pertukaran data, operasi-operasi matematik, menampilkan informasi spasial maupun atribut secara bersamaan, membuat peta tematik, menyediakan bahasa pemrograman (script) serta melakukan fungsi-fungsi khusus lainnya dengan bantuan extensions seperti spasial analyst dan image analyst ESRI (Eko Budiyanto, 2003: 3).

ArcView dalam operasinya menggunakan, membaca dan mengolah data dalam format Shapefile, selain itu ArcView juga dapat memanggil data-data dengan format BSQ, BIL, BIP, JPEG, TIFF, BMP, GeoTIFF atau data grid yang berasal dari ARC/INFO serta banyak lagi data-data lainnya. Setiap data spasial yang dipanggil akan tampak sebagai sebuah Theme dan gabungan dari theme-theme ini akan tampil dalam sebuah view. ArcView mengorganisasikan komponen-komponen programnya (view, theme, table, chart, layout dan script) dalam sebuah project. Project merupakan suatu unit organisasi tertinggi di dalam ArcView.

Salah satu kelebihan dari ArcView adalah kemampuannya berhubungan dan berkerja dengan bantuan extensions. Extensions (dalam konteks perangkat lunak SIG ArcView) merupakan suatu perangkat lunak yang bersifat “plug-in” dan dapat diaktifkan ketika penggunanya memerlukan kemampuan fungsionalitas tambahan (Prahasta). Extensions bekerja atau berperan sebagai perangkat lunak yang dapat dibuat sendiri, telah ada atau dimasukkan (di-instal) ke dalam perangkat lunak ArcView untuk memperluas kemampuan-kemampuan kerja dari ArcView itu sendiri. Contoh-contoh extensions ini seperti Spasial Analyst, Edit Tools v3.1, Geoprocessing, JPGE (JFIF) Image Support, Legend Tool, Projection Utility Wizard, Register and Transform Tool dan XTools Extensions. Arc View biasa digunakan antara lain untuk :

1. Digitasi data citra dari layer monitor (on screen digitizing).
2. Reaktifasi citra dengan bantuan ekstensi image analysis.
3. Editing tema dengan drag and drop atau cut and paste.
4. Editing tema dengan query item pada table.
5. Konvesri data dari MS-EXCEL atau MS-ACCESS menjadi tema baru pada data spasial yang telah ada.
6. Pembuatan kontur dengan bantuan ekstensi image analysis dan spasial analis.
7. Pembuatan peta 3D dan perhitungan volume dengan bantuan 3D analysis.
8. Pengubahan system proyeksi dengan projection utility.

9. Kemudahan konversi data ke perangkat lunak lain, seperti :
AUTOCAD, MAPINFO. (Eko Budiyanto, 2003: 3).

II.3. Alat Bantu Pengembangan Sistem

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

II.3.1. UML (Unified Modelling Language)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

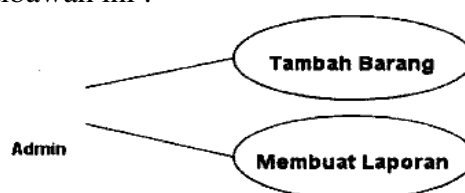
UML merupakan kesatuan bahasa dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OTM) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady booch sangat terkenal dengan nama metode *Design Object oriented*. Metode ini menjadikan proses analisis dan design kedalam empat tahapan interaktif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen (Munawar; 2005 : 16).

II.3.2. Diagram UML(*Unified Modelling Language*)

UML (*Unified Modelling Language*) mendefinisikan diagram-diagram berikut ini :

a. *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 *actor* 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. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Hal ini dapat dilihat pada gambar II.4 dibawah ini :

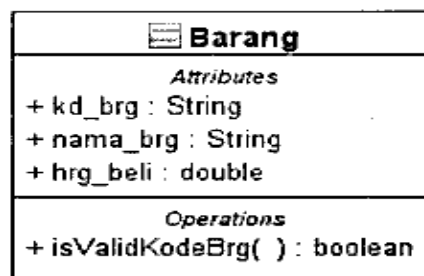


Gambar II.5 *Use case diagram*

(Sumber: Miftakhul Huda: 2010 : 138)

b. *Class diagram*

Class adalah sebuah spesifikasi objek, yang memiliki atribut/properti dan layanan/fungsional. *Class Diagram* menggambarkan struktur dan deskripsi kelas, package dan objek beserta hubungan satu sama lain, seperti *containment*, pewarisan, *asosiasi* dan lain-lain. *Class* memiliki tiga hal pokok: nama (dan *stereotype*), atribut dan metode. *Class diagram* dapat dilihat seperti gambar II.5 dibawah ini :

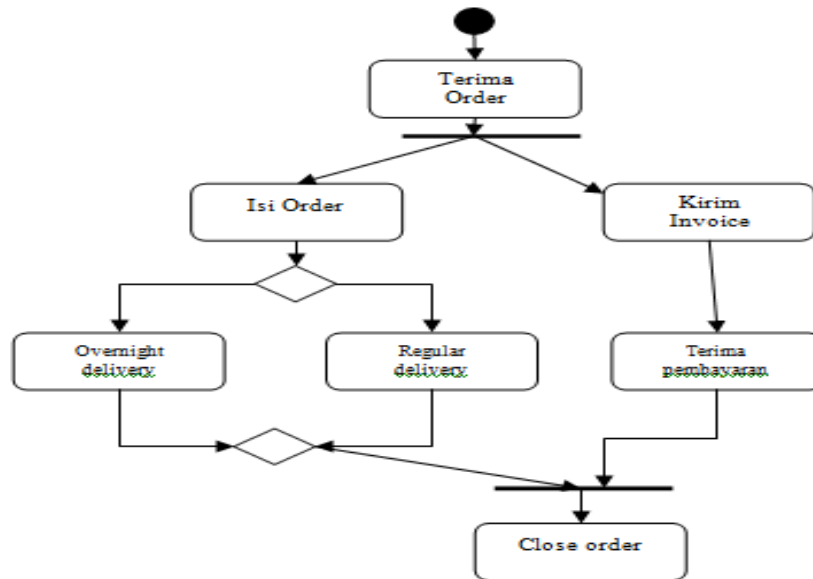


Gambar II.6 Class diagram

(Sumber: Miftakhul Huda; 2010 : 139)

c. *Activity diagram*

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *Activity Diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity Diagram* dapat dilihat seperti gambar II.6 :



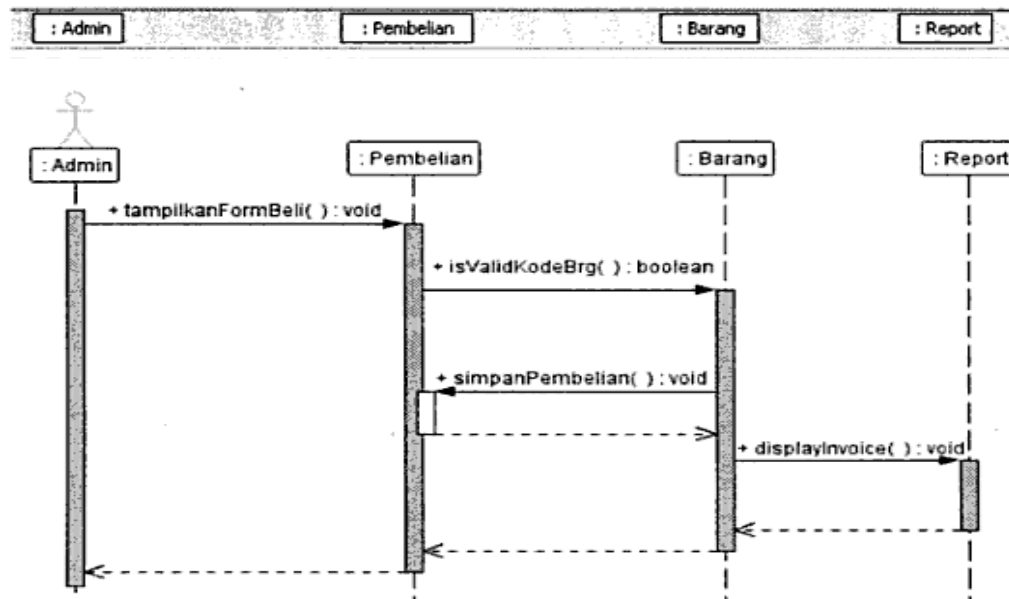
Gambar II.7 Activity Diagram

(Sumber: Munawar; 2005 : 111)

d. *Sequence diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kota segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Hal itu dapat dilihat seperti gambar II.7 :



Gambar II.8 *Sequence diagram*

(Sumber: Miftakhul Huda; 2010 : 143)

II.4. Sistem Basis Data (Database)

Basis data adalah kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan oleh pemakainya (Linda Marlinda; 2004 : 1). Komponen-komponen pada sebuah sistem basis data antara lain:

1. Perangkat keras
2. Sistem operasi
3. Basis data
4. DBMS (Database Management System)
5. Pemakai
6. Aplikasi lain

Pada sebuah institusi, data merupakan salah satu hal yang sangat penting. Setiap bagian/divisi dari institusi memiliki data sendiri-sendiri. Tapi setiap bagian pun membutuhkan sebagian data dari bagian yang lain. Hal ini yang biasa dikenal sebagai “shared data”. Setiap divisi memiliki aplikasi sendiri-sendiri dalam melakukan manipulasi dan pengambilan data tersebut. Setiap aplikasi memiliki file-file dalam sistem operasi yang digunakan untuk menyimpan data-data. Seiring dengan berkembangnya institusi, bertambahnya bagian/divisi, bertambah pula data dan aplikasi yang digunakan. Bertambahnya aplikasi, bertambah pula file-file yang dibuat.

Menyimpan data dalam bentuk file yang berbeda-beda, memiliki kekurangan-kekurangan:

1. Data redundancy dan inconsistency.

Dikarenakan programmer yang berbeda membuat file dan aplikasi masing-masing, menyebabkan beragam format dan aplikasi yang dibuat. Bahkan, aplikasi pun dibuat menggunakan bahasa pemrograman yang berbeda-beda. Lebih jauh lagi, data atau informasi yang sama bisa terdapat dalam beberapa file yang berbeda. Ini yang disebut dengan redundancy. Redundancy data ini lama kelamaan akan menyebabkan inconsistency dari data.

2. Kesulitan dalam pengaksesan data.

Dikarenakan setiap aplikasi memiliki file tersendiri untuk penyimpanan dan pengambilan data, maka jika suatu bagian dari institusi membutuhkan data dari bagian lain, akan menemui kesulitan. Hal ini

dikarenakan aplikasi yang dimiliki bagian tersebut, tidak dapat membaca file yang terdapat di bagian lain.

3. Isolasi data.

Dikarenakan data tersebar dalam berbagai macam file, dan file tersebut dalam beragam format, pembuatan aplikasi baru akan terasa sulit ketika harus membaca format dari masing-masing file tersebut.

4. Masalah integritas.

Data yang disimpan harus memenuhi hal yang dinamakan dengan consistency constraint. Jika sebuah constraint berubah, maka seluruh aplikasi yang digunakan harus mengakomodasinya. Masalah akan muncul, jika constraint melibatkan beberapa data dari file yang berbeda-beda.

5. Masalah keamanan.

Tidak semua pengguna dari basis data dapat mengakses semua data. Hal ini akan sulit dilakukan jika menggunakan gaya penyimpanan data dalam file.

Tujuan utama dari sistem basis data adalah untuk menyediakan fasilitas untuk view data secara abstrak bagi penggunanya. Namun bagaimana sistem menyimpan dan mengelola data tersebut, hanya diketahui oleh sistem itu sendiri. Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data (Kusrini; 2008 : 2). Berikut ini tiga level abstraksi data:

1. Level fisik

Merupakan level terendah pada abstraksi data yang menunjukkan bagaimana sesungguhnya data disimpan. Pada level ini pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.

2. Level logik

Merupakan level berikutnya pada abstraksi data, menggambarkan data apa yang disimpan pada basis data dan hubungan apa saja yang ada di antara data tersebut.

3. Level view

Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data. Banyak user dalam sistem basis data tidak akan terlibat dengan semua data atau informasi yang ada atau yang disimpan. Para user umumnya hanya membutuhkan sebagian data atau informasi dalam basis data yang kemunculannya di mata user diatur oleh aplikasi end user.

II.4.1. *Entity Relationship Diagram (ERD)*

ERD adalah sebuah model data yang didasarkan atas persepsi dunia nyata, yang terdiri atas object dasar yang disebut dengan entitas dan hubungan (relation) antar object-object tersebut. Tujuan dari ERD adalah untuk mewakili object data dan hubungan mereka.

Sesuai dengan namanya ada 2 komponen utama pembentuk model keterhubungan entitas yaitu entity (entitas) dan relasi (relation). Entitas menyatakan suatu object yang mempresentasikan suatu himpunan atau sesuatu

didunia nyata yang mempunyai peranan dalam sistem yang sedang dibangun, sedangkan relasi merupakan sebuah kumpulan dari beberapa entitas atau relasi yang memiliki tipe sama.

Pada model entity relationship diagram hubungan antar file direlasikan dengan kunci relasi (relation key), yang merupakan kunci utama dari masing-masing file. Untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu file, yaitu :

1. One to one relationship

Hubungan antara file pertama dan file kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan table dan relasi antar keduanya digambarkan dengan panah tunggal.

2. One to Many relationship


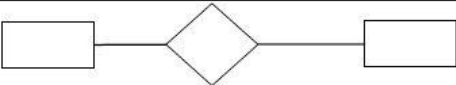




Hubungan antara file pertama dan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Hubungan tersebut dapat digambarkan dengan panah banyak untuk menunjukkan hubungan banyak tersebut.

3. Many to many relationship

Hubungan antara file pertama dan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan panah ganda untuk menunjukkan hubungan banyak tersebut (Kusrini; 2008 : 2).

Berikut tabel 22 akan menunjukkan simbol-simbol yang digunakan dalam ERD.

Tabel II.1 Simbol – Simbol ERD

Gambar	Keterangan
	Entitas
	Hubungan
	Penghubung (link)
	One to Many
	One to One
	Many to Many

(Sumber : Jogiyanto ; 2005 : 795)

II.4.2. Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store.

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara user dan analis sistem tentang data yang mengalir di dalam sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh user.

Sementara itu, pada tahap perancangan sistem kamus data digunakan untuk merancang input, laporan dan database.

Pembentukan kamus data didasarkan atas alur data yang terdapat pada DFD. Alur data pada DFD ini bersifat global, dalam arti hanya menunjukkan nama alur datanya tanpa menunjukkan struktur dari alur data itu. Untuk menunjukkan struktur dari alur data secara terinci maka dibentuklah kamus data yang didasarkan pada alur data di dalam DFD.

Form kamus data Suatu sistem dapat diuraikan ke dalam 4 form kamus data yang menerangkan isi database sistem dalam bentuk hirarki seperti yang digambarkan sebagai berikut :

Data Flow Dictionary Entry	Data Store Dictionary Entry
Data Structure Dictionary Entry	
Data Element Dictionary Entry	

Gambar II.9 Hirarki dari Form Kamus Data

(Sumber : Miftakhul Huda; 2008 : 129)

Dari gambar di atas tampak bahwa data flow dan data store ada pada level tertinggi. Di sini lebih baik menganggap data flow dan data store sebagai file dari data. Selanjutnya struktur data yang ada pada data flow dan data store terletak pada level kedua atau middle level. Di sini struktur data dianggap sebagai record data. Yang terakhir adalah data element yang terletak pada level terendah, karena data element merupakan bagian dari struktur data. Di sini data element dianggap sebagai field.

Data Flow Dictionary Entry

Data flow dictionary entry ini menerangkan setiap data flow pada DFD.

Data flow ini dapat berupa :

- a. Satu struktur yang terdiri dari satu elemen data tunggal.
- b. Satu struktur yang terdiri dari satu paket elemen data.
- c. Multiple struktur.

Berdasarkan uraian di atas, maka hubungan antara alur data pada DFD dan alur data pada elemen kamus data adalah one to one relationship (relasi satu-satu). Jika DFD berisi 40 alur data, maka kamus data harus mempunyai 40 elemen alur data (Hartono : 2003).

II.4.3 Normalisasi

Normalisasi merupakan proses pengelompokan attribute-atribute dan suatu relasi sehingga membentuk WELL-STRUKTUR RELATION. Normalisasi merupakan proses pengelompokan elemen data menjadi table-tabel yang menunjukkan entity dan relasinya. Normalisasi ditemukan pada tahun 1970 oleh E. F. CODD (Linda Marlinda: 2004: 115). Bentuk normalisasi yang biasanya digunakan dalam normalisasi adalah bentuk :

- a. Bentuk normalisasi I/1-NF (First Normal Form)

Suatu relasi memenuhi normal 1-NF jika setiap kolom bernilai tunggal untuk setiap barisnya.

- b. Bentuk normalisasi II/2-NF (Second Normal Form)

Suatu relasi memenuhi normal 2-NF jika memenuhi 1-NF, dan setiap kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Disebut

tergantung sepenuhnya terhadap kunci primer jika suatu kolom selalu bernilai sama untuk nilai kunci yang sama.

c. Bentuk normalisasi III/3-NF (Third Normal Form)

Suatu relasi memenuhi normal 3-NF jika relasi tersebut memenuhi 2-NF, dan setiap kolom bukan kunci tidak tergantung secara fungsional kepada kolom bukan kunci yang lain dalam relasi tersebut. Dengan kata lain setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer (Linda Marlinda: 2004: 120).

II.5. Website

Website atau disingkat web atau dikenal juga dengan nama homepage kini telah menjadi trend. Ribuan bahkan jutaan website kini telah meramaikan dunia maya seiring dengan perkembangan internet. Bila telah terhubung dengan internet, sebuah perusahaan dapat mempromosikan produknya ke segala penjuru dunia lewat website perusahaan. Dengan www.yahoomail.com kita bisa berkirim email ke teman kita di luar negeri tanpa harus repot repot menulis di kertas, memasukkan ke dalam amplop, dan mengirimnya lewat kantor pos. Dengan website pula kita bisa mengetahui berita mutakhir lewat www.detik.com, dan dengan website pula kita bisa menambah teman dengan mendaftar ke www.friendster.com.

Kita pun bisa membuat website seperti itu sendiri dengan bantuan software web developer yang sudah ada seperti Microsoft Frontpage atau Macromedia Dreamweaver. Kalaupun program tersebut tidak terinstal di komputer anda,

asalkan ada program Notepad, Internet Explorer, atau web browser sejenis anda masih bias membuat web pribadi anda.

Website bisa dikembangkan oleh perorangan, group tertentu, institusi pendidikan, universitas, organisasi nirlaba, institusi bisnis dan lain sebagainya. Walaupun skala organisasi berbeda-beda, tiap orang memiliki kesempatan untuk membuat website.(Ali Zaky; 2006 : 3)

II.6. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius (Achmad Solihin; 2010 : 8).

Fitur-fitur MySQL antara lain :

1. Relational Database System.

Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.

2. Mendukung Sub Select.

Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).

3. Mendukung Views.

MySQL mendukung views sejak versi 5.0

4. Mendukung Stored Prosedured (SP).

MySQL mendukung SP sejak versi 5.0

5. Mendukung Triggers.

MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.

6. Mendukung replication.

7. Mendukung transaksi.

8. Mendukung foreign key.

9. Tersedia fungsi GIS.

10. Free (bebas didownload).

11. Stabil dan tangguh.

12. Fleksibel dengan berbagai pemrograman.

13. Security yang baik (Achmad Solihin; 2010 : 8).

II.7. Java

Pengertian Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan (Bambang Hariyanto; 2005:6).

. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

- Programming-language specification
- Application-programming interface
- Virtual-machine specification

Java API terdiri dari tiga bagian utama:

- Java Standard Edition (SE), sebuah standar API untuk merancang aplikasi desktop dan applets dengan bahasa dasar yang mendukung grafis, M/K, keamanan, konektivitas basis data dan jaringan.
- Java Enterprise Edition (EE), sebuah inisiatif API untuk merancang aplikasi server dengan mendukung untuk basis data.
- Java Micro Edition (ME), sebuah API untuk merancang aplikasi yang jalan pada alat kecil seperti telepon genggam, komputer genggam dan pager.

Java Virtual Machine (JVM) adalah sebuah spesifikasi untuk sebuah komputer abstrak. JVM terdiri dari sebuah kelas pemanggil dan sebuah interpreter Java yang mengeksekusi kode arsitektur netral. Kelas pemanggil memanggil file .class dari kedua program Java dan Java API untuk dieksekusi oleh interpreter Java. Interpreter Java mungkin sebuah perangkat

lunak interpreter yang menterjemahkan satu kode byte pada satu waktu, atau mungkin sebuah just-intime (JIT) kompiler yang menurunkan bytecode arsitektur netral kedalam bahasa mesin untuk host computer.

Sistem Operasi Java Sistem operasi biasanya ditulis dalam sebuah kombinasi dari kode bahasa C dan assembly, terutama disebabkan oleh kelebihan performa dari bahasa tersebut dan memudahkan komunikasi dengan perangkat keras. Satu kesulitan dalam merancang sistem basis bahasa adalah dalam hal proteksi memori, yaitu memproteksi sistem operasi dari pemakai program yang sengaja memproteksi pemakai program lainnya. Sistem operasi tradisional mengharapkan pada tampilan perangkat keras untuk menyediakan proteksi memori. Sistem basis bahasa mengandalkan pada tampilan keamanan dari bahasa. Sebagai hasilnya, sistem basis bahasa menginginkan pada alat perangkat keras kecil, yang mungkin kekurangan tampilan perangkat keras yang menyediakan proteksi memori (Bambang Hariyanto; 2005:6).

II.8. Netbeans

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.

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 desktop, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi.

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. 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 (Bambang Hariyanto; 2005).