BAB II
LANDASAN TEORI

2.1. Pengertian Perangkat Lunak (*Software*)

Perangkat lunak memiliki pengertian menunjuk pada program dan alat bantu lain yang bersifat menambah kemampuan komputer sebagai alat untuk melaksanakan tugas atau operasi tertentu. Program aplikasi dapat dibuat secara khusus untuk memenuhi kebutuhan khusus pula (*tailor-made*) atau berupa paket yang mempunyai aplikasi umum.

Disebut juga dengan perangkat lunak, merupakan kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya. perangkat lunak ini merupakan catatan bagi mesin komputer untuk menyimpan perintah, maupun dokumen serta arsip lainnya.

Merupakan data elektronik yang disimpan sedemikian rupa oleh komputer itu sendiri, data yang disimpan ini dapat berupa program atau instruksi yang akan dijalankan oleh perintah, maupun catatan-catatan yang diperlukan oleh komputer untuk menjalankan perintah yang dijalankannya. Untuk mencapai keinginannya tersebut dirancanglah suatu susunan logika, logika yang disusun ini diolah melalui perangkat lunak, yang disebut juga dengan program beserta data-data yang diolahnya. Pengeloahan pada *software* ini melibatkan beberapa hal, diantaranya adalah sistem operasi, program, dan data. *Software* ini mengatur sedemikian rupa sehingga logika yang ada dapat dimengerti oleh mesin komputer.

2.2. Konsep Dasar Sistem

Pengertian sistem menurut Jogiyanto H.M yang dalam bukunya menyatakan :

"*Sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan"*

Sistem dapat didefinisikan juga sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau sub-sistem yang berinteraksi untuk mencapai suatu tujuan. Sistem adalah kumpulan dari bagian apapun baik fisik maupun non fisik

yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

Dari pengertian diatas dapat diambil kesimpulan bahwa didalam sistem terdiri dari sistem-sistem bagian (subsistem). Masing-masing subsistem dapat terdiri dari subsistem yang lebih kecil lagi. Subsistem ini saling berinteraksi dan saling berhubungan membentuk satu kesatuan (terintegrasi) sehingga tujuan sistem tersebut dapat tercapai.

2.2.1. Pengertian Dasar Sistem

Suatu sistem pada dasarnya merupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan yang melaksanakan dan mempermudah pelaksanaan kegiatan utama dari suatu organisasi. Menurut Jogiyanto *"Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, nerkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu".*

Sistem itu sendiri mempunyai tujuan yang sama untuk menghasilkan sesuatu yang lebih bermanfaat ada yang menyebut mencapai suatu tujuan *(goal)* dan ada yang menyebutkan untuk mencapai sasaran *(object)*. Jadi dapat disimpulkan pengertian sistem adalah kumpulan dari berbagai macam elemen yang berbeda namun mempunyai tujuan yang sama untuk mencapai sasaran atau objek.

Beberapa ciri dari sistem yaitu sebagai berikut :

1. Mengarah Pada Tujuan

   Cara kerja sistem ini adalah merangkai dan mengkoordinasikan fakta-fakta untuk mencapai tujuan dengan menggunakan aturan-aturan tertentu.

2. Merupakan Suatu Keseluruhan

   Sistem merupakan suatu keseluruhan yang bulat dan utuh, dimana tujuan masing-masing dari bagian yang membentuk sistem akan saling menunjang dan mencapai tujuan dari sistem secara keseluruhan, dan ini berarti bahwa pencapaian tujuan dari salah satu bagian tidak dapat

dilakukan dengan mengabaikan pencapaian tujuan dengan bagian yang lainnya.

3. Adanya Keterbatasan

   Sistem memiliki sifat yang terbuka, dimana suatu sistem dapat berinteraksi dengan sistem lainnya yang lebih besar.

4. Adanya Proses Transformasi

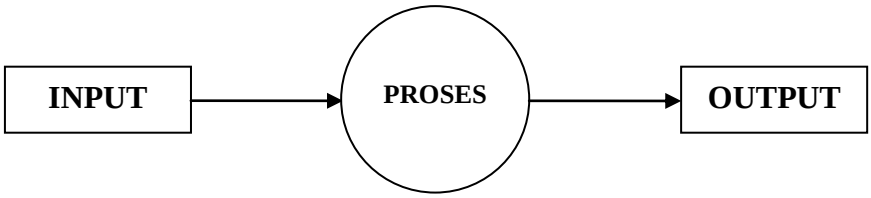
   Suatu sistem mempunyai atau melakukan proses transformasi kegiatan yang mengubah suatu *input* atau masukan menjadi suatu *output* untuk mencapai suatu tujuan.

5. Saling Berkaitan

   Sistem terdiri dari elemen-elemen yang saling berkaitan satu elemen dengan elemen yang lain.

2.2.2. Bentuk Umum Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*), dalam bentuk umum sistem ini bisa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan rencana yang telah ditentukan sebelumnya.



Gambar 2.1 Bentuk Umum Sistem

2.2.3. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai sebagai berikut :

1. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama untuk membentuk satu kesatuan.

Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

   Batas sistem 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 (*Environment*)

   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 meupakan energi dari sistem dan harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lainnya. Keluaran (*output*) dari suatu subsistem akan menjadi masukan (*input*) untuk subsistem yang 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 beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

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 supra sistem.

7. Pengolahan (*Process*)

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

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

   Suatu sistem pasti mempunyai tujuan atau sasaran. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.4. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai *"penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya."* Analisis dan Desain : Jogiyanto

2.3. Konsep Dasar Informasi

Definisi umum informasi dalam pemakaian sistem informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berguna dan lebih berarti bagi yang memakai di dalam mengambil keputusan. Jadi sistem pengolahan informasi pengolahan data dari bentuk tak berguna menjadi berguna bagi penerimanya.

2.3.1. Pengertian Sumber Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang berguna untuk pengambilan keputusan.

Sumber dari Informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan saat tertentu.

Informasi adalah data yang telah diproses kedalam suatu bentuk yang lebih berarti bagi penerima dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau keputusan mendatang.

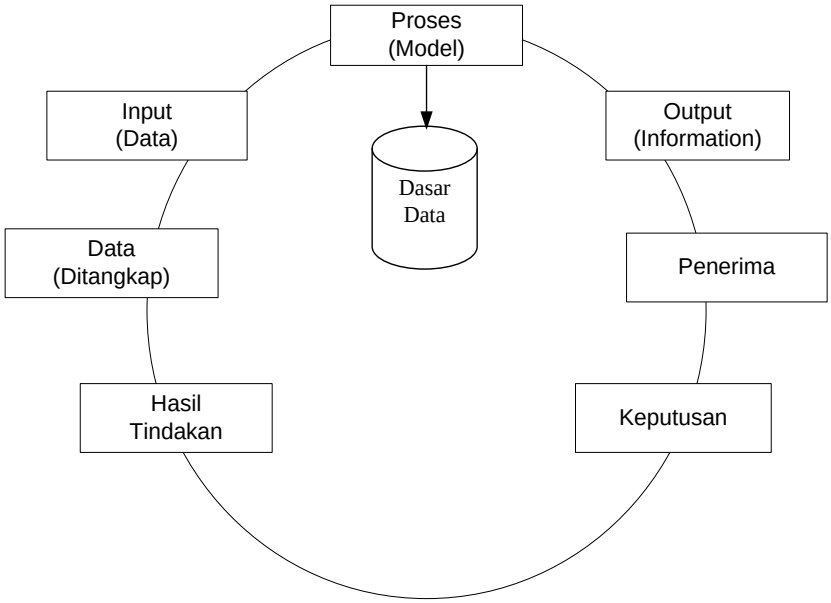
Informasi yang bersumber dari proses data harus merupakan informasi yang terstruktur. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.3.2. Siklus Informasi

Data merupakan bentuk yang masih mentah sehingga perlu diolah untuk proses lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tetentu agar dapat lebih berguna dalam bentuk informasi.

Data yang diolah melalui suatu model menjadi informasi, penerima kamudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan dan menghasilkan suatu tindakan yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali melalui suatu model sampai mmbentuk suatu siklus informasi.

Siklus informasi dapat dilihat seperti pada gambar 2.3 berikut :



Gambar 2.2 Siklus Informasi
Sumber : Jogiyanto H.M, *Pengenalan Komputer*

2.3.3. Kualitas Informasi

Kualitas dari suatu informasi yang diharapkan tergantung tiga hal pokok, yaitu :

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan yang biasanya terjadi dan selain itu harus jelas maksud dan tujuannya sehingga *output*/keluaran biasa dipertanggung jawabkan.

2. Tepat Waktu

   Informasi yang dihasilkan atau dibuthkan tidajk boleh terlambat (usang) informasi yang usang tidak akan mempunyai nilai yang baik, sehingga jika digunakan sebagai dasar dari pengambilan keputusan akan bersifat fatal atau mengalami kesalahan dalam pengambilan keputusan dan tindakan.

3. Relevan

   Berarti informasi tersebut mempunyai manfaat dan informasi yang diterima si pemakai dan pemakai lainya bisa berbeda-beda.

2.3.4. Nilai Informasi

Nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *Cost Effectiveness* atau *Cost Benefit.*

2.3.5 Konsep Dasar Sistem Informasi

Sistem informasi merupakan suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan stategis dari suatu organisasi dan menyediakan informasi untuk pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Sistem informasi dapat didefinisikan sebagai berikut :

"*Sistem informasi adalah suatu sistem didalam suau organisasi yang mempertemukan kebutuhan pengolahan transaksi yang mendukung operasi bersifat manajerial dan kegiatan stategi-stategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".*

2.3.6 Komponen Sistem Informasi

Sistem informasi mempunyai beberapa komponen yaitu :

1. Perangkat keras *(Hardware)* yaitu semua alat komputeryang secara fisik dapat diliah dan diraba dengn jelas, seperti: *keyboard*, *harddisk*, *monitor*, alat *printer*, *mouse*, dan lain-lain.
2. Perangkat lunak *(Software)* yatitu semua pengakat komputer yang dapat mebuat perangkat keras komputer bekerja menjalankan fungsinya.
3. Manusia *(Brainware)* yaitu orang yang mengoperasikan komputer.

2.4 Alat-alat Pemodelan Sistem

Pemodelan sistem merupakan hal yang penting bagi kelangsungan sistem itu sendiri. Pemodelan sistem adalah suatu upaya untuk menjaga efektifitas sistem dalam memenuhi kebutuhan pengguna sistem.Pemodelan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sudah ada.

Alat-alat yang digunakan dalam pemodelan sistem antara lain adalah:

1. Diagram Konteks.
2. Flowmap
3. *Entity Relationship Diagram* (ERD)
4. *Data Flow Diagram* (DFD).
5. Kamus Data *(Data Dictionary).*

2.5 Pendekatan Analisis Sistem

Untuk melangkah ke tahap berikutnya yaitu tahap desain, maka sebelumnya akan melakukan tahap-tahap sebagai berikut :

1. **Diagram Konteks**

Diagram konteks menggambarkan aplikasi dalam satu lingkaran dan hubungan dengan entitas luar. Dimana lingkaran tersebut menggambarkan keseluruhan proses dalam aplikasi. Dalam penggambaran ini, sistem dianggap sebagai sebuah objek yang tidak dijelaskan secara rinci, karena yang ditekankan adalah interaksi sistem dengan lingkungan yang mengaksesnya.

2. **Flow Map**

Bagan alir atau *flow map* adalah bagan yang menunjukan alir didalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan dokumentasi.

Bagan alir dokumen atau sering disebut juga *Flow Map* merupakan bagan alir yang menunjukan arus dari aliran formulir termasuk tembusan-tembusannya.

3. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram adalah salah satu model atau teknik pendekatan yang dapat menyatakan atau mengambarkan hubungan antara *entity* di dalam sebuah sistem di mana sebuah hubungan tersebut dinyatakan sebagai *one to one, one to many, many to one* dan *many to many.*

Entity Relationship Diagram memiliki komponen-komponen utama, yaitu:

1. Entitas (*Entity*)

Entitas adalah sebuah barang atau objek yang dapat dibedakan dari objek lain. Entitas direpresentasikan dengan empat persegi panjang.

2. Atribut (*Attribute*)

Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Atribut direpresentasikan dengan bentuk *elips*.

3. Relasi (*Relationship*)

Relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Relasi direpresentasikan dengan jajaran genjang. Relasi antar tabel dikategorikan menjadi tiga macam, yaitu :

4. Kardinalitas (*Cardinality*)

Kardinalitas adalah angka yang menunjukkan banyaknya kemunculan suatu objek terkait dengan kemunculan objek lainnya pada suatu relasi.

Jenis-jenis kardinalitas yaitu sebagai berikut :

a. *One to One*

Hubungan *one to one* adalah antara *file* pertama dengan *file* kedua adalah satu banding satu.

b. *One to Many*

Hubungan *one to many* adalah sebuah relasi merupakan tahap dimana hubungan antara *file* pertama dengan *file* kedua adalah satu banding banyak.

c. *Many to One*

Hubungan *many to one* adalah sebuah relasi merupakan tahap dimana hubungan antara *file* pertama dengan *file* kedua adalah banyak banding satu.

d. *Many to Many*

Hubungan *many to many* adalah sebuah relasi antara suatu *file* satu dengan *file* yang kedua memepunyai relasi banyak berbanding banyak.

4. DFD (*Data Flow Diagram*)

Data Flow Diagram adalah diagram sistem yang menggambarkan cara kerja aplikasi secara *logic*. Mulai dari tingkat paling tinggi sampai dengan tingkat paling rendah. Pada perancangan ini terdiri dari perancangan awal *(preliminary design*) dan perancangan rinci *(detailed design)* sesuai dengan tahap-tahap rekayasa perangkat lunak. Adapun penjelasan dari perancangan awal adalah perancangan sistem yang menggambarkan tentang hubungan antara sistem dengan lingkungan luar sistem.

Hubungan ini dapat digambarkan dengan menggunakan diagram konteks. Sedangkan perancangan rinci adalah perancangan sistem yang menggambarkan tentang proses yang terjadi pada sistem serta arus data yang mengalir antar proses. DFD merupakan alat yang digunakan pada metode pengembangan sistem yang terstruktur, DFD ini menggambarkan arus data di dalam sistem yang terstruktur dan jelas serta merupakan dokumentasi sistem yang baik.

5. Kamus Data *(Data Dictionary)*

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data dapat digunakan pada tahap analisa dan perancangan sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang masukan, merancang laporan-laporan dan *database.*

Dengan adanya kamus data, didapat definisi-definisi dari bentuk-bentuk yang tidak dimengerti dalam DFD yaitu aliran data, *file*, proses dan elemen-elemen data. Arus data pada DFD bersifat global, hanyaditujukan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari arus data, secara lebih lengkap dapat dilihat di kamus data.

2.6 Basis Data

2.6.1 Pengertian Data

Data merupakan ”fakta atau keterangan yang belum mempunyai arti atau nilai, serta data dapat dijadikan kajian analisis atau kesimpulan. Data biasanya terdiri dari beberapa elemen data (data item). Elemen data adalah unit terkecil dari data yang ada artinya bagi pengguna *(user).*

2.6.2 Pengolahan Data

Pengolahan data dengan komputer terkenal dengan nama pengolahan data elektronik. Data adalah kumpulan kejadian yang diangkat dari suatu kenyataan. Data dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus atau gabungan darinya. Pengolahan data adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti, berupa suatu informasi.

2.6.3 Siklus Pengolahan Data

Suatu proses pengolahan data terdiri dari tiga tahapan dasar yang disebut dengan siklus pengolahan data (*data processing cycle*) yaitu *input*, *processing* dan *output*.

1. *Input*, tahap ini merupakan proses memasukkan data ke dalam proses komputer lewat alat input (*input device*).
2. *Process*, tahap ini merupakan proses pengolahan dari data yang sudah dimasukkan yang dilakukan oleh alat pemroses (*processing data*), yang dapat berupa proses menghitung, membandingkan, mengklasifikasikan, mengurutkan, mengendalikan atau mencari di *storage*.
3. *Output*, tahp ini merupakan proses menghasilkan *output* dari hasil pengolahan data ke alat *output* (*output device*), yaitu berupa informasi.

2.6.4 Konsep Dasar Basis Data

Basis Data (*Database)* dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis Data menunjukkan suatu kumpulan data yang dipakai

dalam suatu lingkungan perusahaan atau instansi-instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data (*database system).*

2.6.5 Pengertian Basis Data

Basis data terdiri dari kata basis dan data. Basis dapat diartikan gudang atau tempat bersarang dan data yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

2.6.6 Tujuan Basis Data

Basis Data (*Database*) pada prinsipnya ditujukan untuk pengaturan data agar terdapat kemudahan dalam pengambilan kembali data tersebut. Berikut ini terdapat beberapa tujuan dari Basis Data diantaranya yaitu :

1. Kecepatan dan kemudahan (*Speed)*
2. Efisiensi ruang penyimpanan (*Space)*
3. Keakuratan (*Accuracy)*
4. Ketersediaan (*Avaibility)*
5. Kelengkapan (*Completeness)*
6. Keamanan (*Security*)
7. Kebersamaan (*Sharability*)

2.6.7 Pengguna Basis Data

Pengguna basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem diantaranya yaitu :

1. Programmer Aplikasi (*Application Programmer*)

   Pemakai yang berinteraksi dengan basis data dengan menggunakan *Data Manipulation Language* (DML) untuk membuat aplikasi dengan menggunakan bahasa pemrograman.
2. *User* Mahir (*Casual User*)

Pemakai yang berinteraksi dengan sistem tanpa membuat modul program.

3. *User* Umum (*Naïve User*)

   Pemakai berinteraksi dengan aplikasi basis data yang telah dibuat atau disediakan oleh sistem.

4. *User* Khusus (*Specialized User*)

   Pemakai yang membuat program aplikasi basis data untuk keperluan tertentu atau khusus.

2.6.8 *Database Management System* (DBMS)

Diperlukan suatu sistem untuk diintegrasikan data file ke dalam suatu file sehingga bisa melayani berbagai *user* yang berbeda. Perangkat keras dan lunak serta prosedur yang mengelola *database* manajemen sistem.

DBMS memungkinkan untuk membentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan.

Fungsi yang penting dari DBMS adalah sebagai berikut :

1. Menyediakan sistem akses cepat.
2. Mengurangi kerangkapan data atau *redundancy* data.
3. Memungkinkan adanya *updating* secara bersamaan.
4. Menyediakan sistem yang memungkinkan dilakukannya pengembangan *database*.
5. Memberikan perlindungan dari pihak pemakai tidak berhak.

2.6.9 NetBeans IDE 6.8

NetBeans mengacu pada kedua platform yang kerangka untuk aplikasi Java desktop, dan sebuah lingkungan pengembangan terpadu (IDE) untuk berkembang dengan Java, JavaScript, PHP, Python, Groovy, C, C + + , Scala Clojure dan lain-lain. NetBeans IDE 7.0 tidak lagi mendukung Ruby dan Ruby on Rails.

NetBeans IDE ditulis dalam Java dan dapat dijalankan di mana saja sebuah JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris. Sebuah JDK dibutuhkan untuk fungsi pengembangan Java, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.

Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari serangkaian modular komponen perangkat lunak yang disebut *modul.* Aplikasi berbasis pada platform NetBeans (termasuk NetBeans IDE) dapat diperpanjang dengan pengembang pihak ketiga .

NetBeans dimulai pada tahun 1996 sebagai Xelfi (kata bermain di *Delphi*), Java IDE proyek mahasiswa di bawah bimbingan Fakultas Matematika dan Fisika di Universitas Charles di Praha . Pada tahun 1997 membentuk perusahaan Staněk Romawi di sekitar proyek dan menghasilkan versi komersial NetBeans IDE hingga dibeli oleh Sun Microsystems pada tahun 1999. Sun open-source IDE NetBeans pada bulan Juni tahun berikutnya. Sejak itu, komunitas NetBeans terus berkembang.

NetBeans IDE 6.0 memperkenalkan dukungan untuk mengembangkan modul IDE dan aplikasi-aplikasi client kaya berdasarkan platform NetBeans, Swing Java GUI builder (sebelumnya dikenal sebagai "Project Matisse"), meningkatkan CVS dukungan, WebLogic 9 dan JBoss 4 dukungan, dan perangkat tambahan banyak redaksi. NetBeans 6 tersedia dalam repositori resmi distribusi Linux utama.

NetBeans IDE 6.5, dirilis pada bulan November 2008, memperpanjang yang sudah ada Java EE fitur (termasuk dukungan Java Persistence, EJB 3 dan JAX-WS). Selain itu, NetBeans Enterprise Pack mendukung pengembangan aplikasi perusahaan Java EE 5, termasuk SOA alat desain visual, skema XML alat, layanan web orkestrasi (untuk BPEL), dan UML pemodelan. The NetBeans IDE Bundle untuk C / C + + mendukung C / C + + dan FORTRAN pembangunan.

NetBeans IDE 6.8 adalah IDE pertama untuk memberikan dukungan lengkap dari Java EE 6 dan GlassFish Enterprise Server v3 . Pengembang open-source hosting yang mereka proyek-proyek kenai.com tambahan manfaat dari instant messaging dan pelacakan masalah integrasi dan navigasi yang benar dalam IDE, dukungan

untuk pengembangan aplikasi web dengan PHP 5.3 dan kerangka Symfony, dan code completion ditingkatkan, layout, petunjuk dan navigasi dalam proyek-proyek JavaFX.

NetBeans IDE 6.9, dirilis pada bulan Juni 2010, menambahkan dukungan untuk OSGi , Spring Framework 3.0, Java EE ketergantungan injeksi (JSR-299), Zend Framework untuk PHP , dan lebih mudah navigasi kode (seperti "ditimpa / Diimplementasikan" penjelasan), format , petunjuk, dan refactoring di beberapa bahasa. NetBeans IDE 7.0 dirilis pada April 2011.

2.6.10 Database Microsoft Access 2007

Microsoft Office Access adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk

aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan Microsoft Internet Information Services (IIS) dan menggunakan Microsoft Active Server Pages (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada Microsoft SQL Server yang memiliki kemampuan yang lebih tinggi.

Beberapa pengembang aplikasi profesional menggunakan Microsoft Access untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid Application Development*/*RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para *salesman*.

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka Access (*form*, *report*, *query*, dan kode Visual Basic) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya Microsoft Jet Database Engine (yang secara *default* digunakan oleh Microsoft Access), Microsoft SQL Server, Oracle Database, dan beberapa produk lainnya yang mendukung ODBC.

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statemen SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine.

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, Microsoft JET Database Engine tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari Microsoft Access 2000 yang menggunakan Microsoft Jet Database Engine versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperi halnya sebuah stored procedure, meskipun prosesur tersebut dibatasi hanya untuk sebuah pernyataan tiap prosedurnya. Access juga mengizinkan *form* untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan form tersebut, dan merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*pass-through* dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya.

Dalam berkas Access Database Project (ADP) yang didukung oleh Microsoft Access 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau Microsoft SQL Server, ketimbang menggunakan Microsoft JET Database Engine. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampur semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints* dan *trigger*, *view*, *stored procedure*, dan *UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah *form*, *report*, *macro*, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut.

Access mengizinkan pengembangan yang relatif cepat karena semua tabel basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya (*.MDB). Untuk membuat Query, Access menggunakan Query Design Grid, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa pemrograman SQL. DI dalam Query Design Grid, para pengguna dapat memperlihatkan tabel basis data sumber dari *query*, dan memilih *field-field* mana yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam *grid*. *Join* juga dapat dibuat dengan

cara mengklik dan menyeret field-field dalam tabel ke dalam *field* dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan.

Bahasa pemrograman yang tersedia di dalam Access adalah Microsoft Visual Basic for Applications (VBA), seperti halnya dalam beberapa aplikasi Microsoft Office. Dua buah pustaka komponen Component Object Model (COM) untuk mengakses basis data pun disediakan, yakni Data Access Object (DAO), yang hanya terdapat di dalam Access 97, dan ActiveX Data Objects (ADO) yang tersedia dalam versi-versi Access terbaru.

Koneksi Java NetBeans Ke Microsoft Access

Untuk menggabungkan bahasa pemograman Java NetBeans dan Microsoft Accsess dibutuhkanbeberapa perintah khusus, disini penulis mempunyai contoh coding untuk mengkoneksikan program tersebut, yaitu :

```
private Object[][] getData(){
Object [][]data1=null;
Connection conn;
try{
//Memanggil Driver
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
//Membuat variabel bertipe Connection
//Kelas Connection berfungsi menyimpan sebuah koneksi yang sudah dilakukan oleh DriverManger
conn=DriverManager.getConnection("jdbc:odbc:perpustakaan");
//Membuat Variabel bertipe Statement
//Kelas Statement berfungsi mengirimkan statement Query Ke Database
Statement
st=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,Result
Set.CONCUR_UPDATABLE);
//Membuat Variabel Bertipe ResulSet
```

```
//Kelas Resultset Berfungsi Menyimpan Dataset(Sekumpulan Data) hasil statement Query
ResultSet rs=st.executeQuery("Select * from Anggota");
rs.last();
int rowcount=rs.getRow();
rs.beforeFirst();
data1=new Object[rowcount][3];
int no=-1;
// Menyimpan Data dari Tabel Ke Variabel Array
while(rs.next()){
no=no+1;
data1[no][0]=rs.getString("NoAnggota");
data1[no][1]=rs.getString("Nama");
data1[no][2 ]=rs.getString("Alamat");
}
// Tutup Koneksi
st.close();
conn.close();
}catch (ClassNotFoundException cnfe) {
// Ketika Gagal Memanggil Driver
System.out.println("Class Driver tidak ditemukan.. : " + cnfe);
System.exit(0);
}catch (SQLException sqle) {
// Ketika Gagal Sql
System.out.println("Proses Query Gagal = " + sqle);
System.exit(0);
}catch(Exception e){
System.out.println("Koneksi Access Gagal " +e.getMessage());
System.exit(0);
}
return data1;
```

```
}
```

3. Buat Method tampilTabel();

Source Code Lengkap Sebagai Berikut :

4 . Tambahkan perintah-perintah berikut pada FAnggota.

```
private void tampilTabel(){
String [] JudulKolom={"No Anggota","Nama Anggota","Alamat"};
tabel=new JTable();
tabMode = new DefaultTableModel(getData(), JudulKolom){
boolean[] canEdit = new boolean [] { false, false, false };
@Override
public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}
};
tabel.setModel(tabMode);
jScrollPane1.setViewportView(tabel);
}
```

Pada praktek ini :

Pastikan Import nya seperti ini

Tambahkan variable

table dan tabMode

Panggil method

tampilTabel();

5 . Buat Event actionPerformed pada Tombol Simpan

```
private void BSimpanActionPerformed(java.awt.event.ActionEvent evt)
{
// TODO add your handling code here:
Connection conn;
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
conn=DriverManager.getConnection("jdbc:odbc:perpustakaan");
```

```
Statement
st=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,Result
Set.CONCUR_UPDATABLE);
int rs=st.executeUpdate("insert into Anggota
values("+"'"+txtNoAnggota.getText()+
"','"+txtNama.getText()+"','"+txtAlamat.getText()+"')");
String
Data[]={txtNoAnggota.getText(),txtNama.getText(),txtAlamat.getText()};
if(rs>0){
JOptionPane.showMessageDialog(this,"Input Berhasil");
// Menambah data ke jTable
tabMode.addRow(Data);
}st.close();
conn.close();
}catch (ClassNotFoundException cnfe) { // Ketika Gagal Memanggil
Driver
System.out.println("Class Driver tidak ditemukan.. : " + cnfe);
System.exit(0);
}catch (SQLException sqle) {
System.out.println("Input Gagal = " + sqle);
System.exit(0);
}catch(Exception e){
System.out.println("Koneksi Gagal " +e.getMessage());
System.exit(0);
}
}
```