

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau *variable* yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Selain itu suatu sistem tidak bisa lepas dari lingkungan sekitarnya maka umpan balik (*feed back*) dapat berasal dari lingkungan sistem yang dimaksud. (Tata Sutabri : 2012 : 10).

II.1.1. Karakteristik Sistem

Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Components*).

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

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber – sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan signal (*signal input*).

6. Keluaran Sistem (*output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolah sistem (*Proses*)

Sudah sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi kelauran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus

yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide – ide yang tampak secara fisik, misalnya komputer, sistem produksi sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system*.

3. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program – program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem terbuka dan Sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan lainnya. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. (Tata Sutabri : 2012 : 21-22).

II.2. Konsep Dasar Informasi

II.2.1. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya. (Tata Sutabri : 2012 : 29).

Kualitas informasi tergantung dari 3 hal yaitu :

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan – kesalahan dan tidak menyesatkan.

2. Tepat waktu (*Timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lain berbeda, misalnya informasi sebab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan dan akan lebih relevan bila ditunjukkan ahli teknik perusahaan. (Tata Sutabri : 2012 : 41).

II.2.2. Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi

operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar dengan laporan-laporan yang diperlukan (Tata Sutabri : 2012 : 46). Sistem informasi terdiri dari komponen – komponen yang disebut blok bangunan (*building block*) yaitu :

1. Blok masukan (input block)

Input mewakili data yang masuk kedalam sistem informasi, Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen – dokumen dasar.

2. Blok model (model block)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang di inginkan.

3. Blok keluaran (output block)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (technology block)

Teknologi merupakan “tool box” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, teknologi terdiri dari 3 bagian utama, yaitu teknisi (brainware), Perangkat lunak (software), perangkat keras (hardware).

5. Blok basis data (*database block*)

Basis data atau database merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan – kecurangan, kegagalan – kegagalan. (Tata Sutabri : 2012 : 47-48).

II.3. *PHP*

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa script server-side dalam pengembangan web yang disisipkan pada dokumen HTML.

Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan software *Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di download secara bebas dari situs resminya. Salah satu fitur yang dapat diandalkan oleh *PHP* adalah dukungannya terhadap banyak *database*, seperti *Adabas D, dBase, Direct MS-SQL, Empress, FilePro, FrontBase, Hyperware, IBM DB2, Informix, MSOL, MySql* dan masih banyak lagi (Dadan Sutisna : 2008 : 40).

Bagi pemula belajar *PHP*, istilah ini sangat penting untuk mendukung dalam pembuatan program, tetapi bagi yang telah belajar bahasa pemrograman

seperti bahasa *C*, *Pascal* dan *Java*, istilah ini sudah sering didengar. Kita Perlu mengingat perbedaan tipe data, variabel, konstanta dan operator yang terdapat dalam *PHP*. Tipe data merupakan jenis data, setiap data pasti memiliki jenis data tertentu. Jenis data berguna untuk menghindari ekspresi yang tipe datanya bercampur. Sebagai contoh 'PHP' + 4.0 (jenis data *string PHP* mau ditambah dengan tipe data *numerik 4.0*). PHP mengenal 3 kelompok jenis data yaitu :

1. Integer (bilangan bulat)

Tipe data integer merupakan tipe data yang direpresentasikan berbentuk angka dan tidak memakai tanda titik desimal. Contoh : \$angka=2-;

2. Tipe data Double (bilangan pecahan)

Tipe data double merupakan tipe data yang direpresentasikan memakai tanda titik desimal. Contoh : \$angka=20.50;

3. Tipe data String (teks)

Tipe data string merupakan tipe data yang direpresentasikan diapit tanda petik tunggal (' ') atau petik ganda (" "). Contoh : \$kantor="CV.Noornet";
(Ir.Yuniar Supardi :2010 : 17-18).

II.3.1. Pemrograman Berorientasi Objek dalam PHP

Berikut beberapa contoh untuk memasukkan skrip PHP.

(Dadan Sutisna : 2008 : 41):

1. Penulisan PHP dimulai dengan <? Dan diakhiri dengan ?>

</

Disini skrip PHP

```
?>
```

2. Setiap perintah pada PHP diakhiri dengan titik koma (;)

```
<?
```

```
Perintah 1;
```

```
Perintah 2;
```

```
Dst;
```

```
?>
```

3. Variabel PHP ditulis dengan tanda dollar (\$) dan kemudian diikuti nama variable tanpa spasi.

```
<?
```

```
$namavariabel = "ini adalah variable";
```

```
$hitung = 2 + 5;
```

```
?> (Dadan Sutisna : 2008 : 41).
```

II.3.2. Sistem Pendukung Keputusan

Pada tahun 1970, Scoot Morton menyampaikan konsep utama dari "Sistem Pendukung Keputusan/Decision Support Sistem (DSS) yang mendefenisikan DSS suatu sistem interaktif yang berbasis komputer yang membantu pengambilan keputusan dengan menggunakan data dan model untuk memecahkan masalah yang tidak terstruktur". Definisi DSS menurut Keen dan Scoot Morton adalah "gabungan sumber daya individual yang intelektual dengan kemampuan komputer untuk meningkatkan kualitas keputusan yang diambil yaitu sebuah sistem yang dibantu oleh komputer untuk pembuatan keputusan yang berhubungan dengan

masalah semi terstruktur”. Untuk lebih jelasnya pemahaman masing-masing fase pada proses pengambilan keputusan, maka berikut ini diuraikan secara jelas dari masing-masing fase tersebut.

1. Intelligence, Fase Intelligence dimulai dengan pengidentifikasian tujuan dan organisasi serta objektif yang berhubungan pada isu-isu yang sedang ditangani.
2. Design, Fase ini meliputi penemuan, pengembangan dan penganalisisan aksi-aksi yang mungkin, mencakup pemahaman masalah dan pengujian solusi.
3. Choice, Proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam pengambilan keputusan.
4. Implementation, Pelaksanaan tindakan dalam proses pengambilan keputusan.
(Jurnal Teknologi Informasi & Pendidikan:Madison:2012:2).

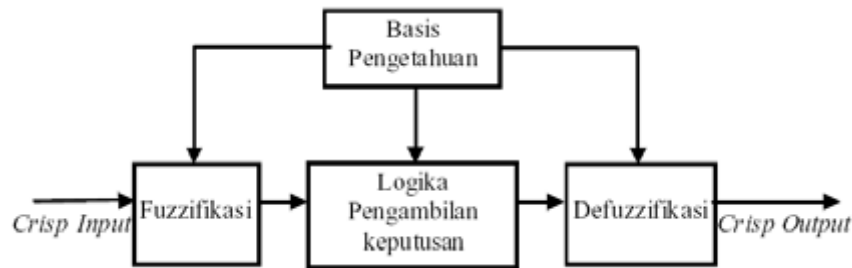
II.3.3. Logika Fuzzy

Logika *fuzzy* merupakan pengembangan dari teori himpunan *fuzzy* yang diprakasai oleh Prof. Lofti Zadeh dari Universitas California USA, pada tahun 1965. Logika *Fuzzy* berbeda dengan logika digital biasa, dimana logika digital biasa hanya mengenal dua keadaan yaitu : Ya_Tidak atau ON_OFF atau High_Low atau “1”_”0”. Sedangkan logika *fuzzy* meniru cara berfikir manusia dengan menggunakan konsep sifat kesamaran suatu nilai. Pada Logika *Fuzzy* dapat memberikan suatu nilai dari nol secara kontiniu sampai nilai satu. Logika

Fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *Output*. (Jurnal Teknologi Informasi & Pendidikan:Madison:2012:4).

Konfigurasi dasar dari sistem logika fuzzy

yang digunakan seperti terlihat pada Gambar II.1 berikut :



Gambar II.1 Konfigurasi dasar sistem logika fuzzy

Sumber (Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor DC:Sutikno:2010:27)

Konfigurasi sistem logika *fuzzy* terdapat 4 komponen utama, yaitu unit fuzzifikasi, basis pengetahuan yang terdiri dari basis data dan basis aturan, logika pengambilan keputusan, dan unit defuzzifikasi. Proses fuzzifikasi dipergunakan untuk mengubah data masukan tegas bentuk derajat keanggotaan. Basis pengetahuan dipergunakan untuk menghubungkan himpunan masukan dengan himpunan keluaran. Logika pengambilan keputusan dipergunakan untuk mengkombinasi aturan-aturan yang terdapat pada basis aturan suatu pemetaan dari suatu himpunan *fuzzy input* ke suatu himpunan *fuzzy output*. Defuzzifikasi adalah langkah terakhir dalam suatu sistem logika *fuzzy* dengan tujuannya adalah mengkonversi setiap hasil dari *inference engine* yang diekspresikan dalam bentuk himpunan *fuzzy* ke suatu bilangan *real*. (Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor DC:Sutikno:2010:27)

Orang yang belum pernah mengenal logika *fuzzy* pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang amat rumit dan tidak menyenangkan. Namun, sekali seseorang mulai mengenal *fuzzy*, ia pasti akan sangat tertarik dan akan menjadi pendatang baru untuk ikut serta mempelajari logika *fuzzy*. Logika *fuzzy* dikatakan sebagai logika baru yang lama. Sebab ilmu tentang logika *fuzzy* modern dan metodis baru ditemukan beberapa tahun yang lalu. Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang *output*. Sebagai contoh :

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi setiap hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tindakan pelayan yang diberikan.
3. Anda mengatakan kepada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.
4. Penumpang taksi berkata pada sopir taksi seberapa cepat laju kendaraan yang diinginkan, sopir taksi akan mengatur pijakan gas taksinya. (Rika Rosnely:2012:63-64).

Ada beberapa alasan mengapa orang menggunakan logika fuzzy, antara lain :

1. Konsep logika fuzzy mudah dimengerti, konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.

2. Logika fuzzy sangat fleksibel
3. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika fuzzy dapat berkerja sama dengan teknik teknik kendali secara konvensional.
7. Logika fuzzy didasarkan pada bahasa alami. (Rika Rosnely:2012:65).

II.4. Alat Bantu Pengembangan Sistem

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

II.4.1. UML (Unified Modelling Language)

Unified Modelling Language (UML) merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem, mau tidak mau pasti akan menjumpai *UML*, baik kita sendiri yang membuat atau sekedar membaca diagram *UML* buatan orang lain. Sebelum ada *UML*, para pengembang bahasa pemrograman berorientasi objek sulit untuk

berkomunikasi satu sama lain. Para pengguna notasi yang berlainan ini saling berebut pengaruh agar notasi yang mereka gunakan jadi notasi standar.

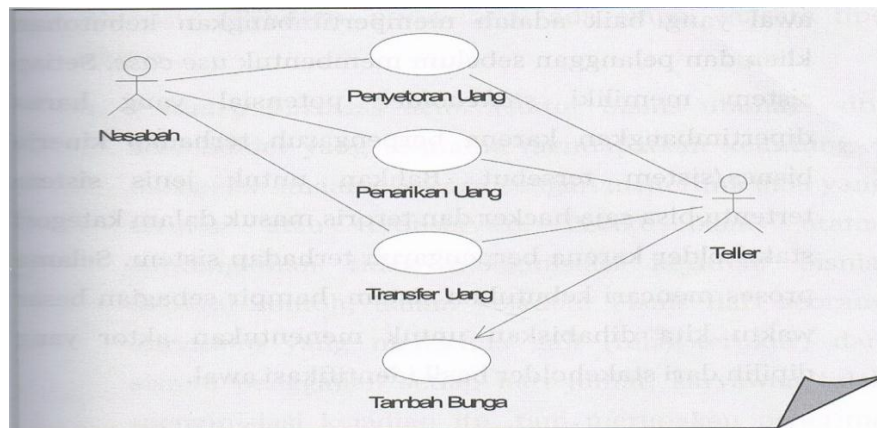
Sejak tahun 1997, divisi *Revision Task Force (RTF)* milik *OMG* beberapa kali merevisi *UML*. Revisi dimaksudkan untuk memperkuat konsistensi notasi, meningkatkan kekompakan antara *user* dan pengembang perangkat lunak. Akan tetapi *UML* terpaksa mengikuti perkembangan *software-software* berbasis objek yang ada. Setelah dilakukan perubahan secara sistematis, akhirnya dihasilkan *UML 2.0* pada tahun 2003. (Prabowo Pudjo Widodo ; 2011 : 7-9).

II.4.2. Diagram – diagram Pada Metode UML

1. Use Case Diagram

Menurut (Philone:2005:bab 7.1) *use case* adalah menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten:2004:258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait, baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. (Prabowo Pudjo Widodo ; 2011 : 21).

Notasi *use case* menunjukkan 3 aspek dari sistem yaitu *actor use case* dan *system / sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Ilustrasi *actor*, *use case* dan *system* ditunjukkan pada gambar II.2

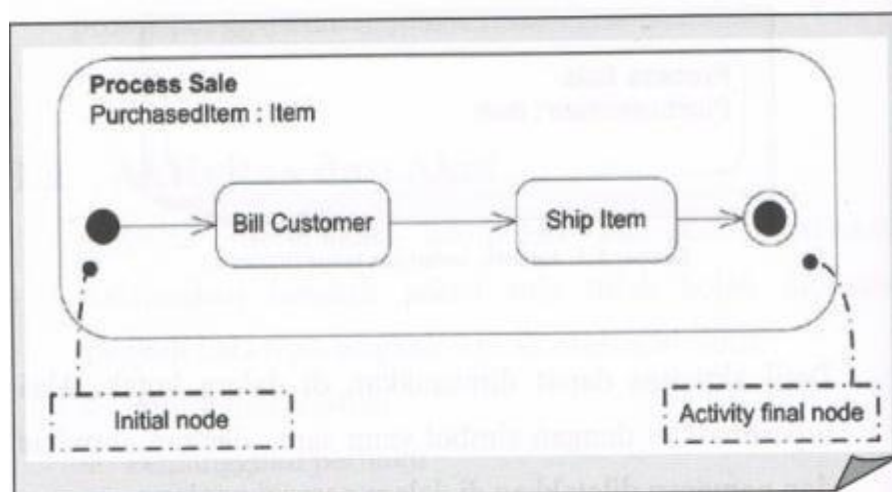


Gambar II.2. Use Case Diagram

Sumber : (Prabowo Pudjo Widodo ; 2011 : 17)

2. Activity diagram

Activity diagram lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan *software* melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas dalam bentuk kumpulan aksi-aksi. (Prabowo Pudjo Widodo ; 2011 : 143). Berikut gambar dari sederhana dari *activity diagram*.

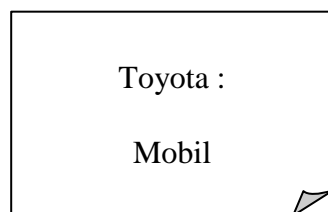


Gambar II.3 Contoh Activity Diagram Sederhana

Sumber : (Prabowo Pudjo Widodo ; 2011 : 146)

3. Class Diagram

Diagram kelas atau class diagram adalah inti dari proses pemodelan objek baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model. (Prabowo Pudjo Widodo :2011 : 37).



Gambar II.4 Contoh Activity Diagram Sederhana

Sumber : (Prabowo Pudjo Widodo ; 2011 : 41)

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*.

Komponen utama *Sequence diagram* terdiri dari atas objek yang dituliskan dengan kotak segiempat bernama. *Messege* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

II.5. Sistem Basis Data

Basis adalah markas, gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek

seperti manusia (dosen, mahasiswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Basis data adalah kumpulan data yang saling berhubungan yang disimpan/diorganisasi secara bersama, dalam bentuk sedemikian rupa, dan tanpa redundansi (pengulangan) yang tidak perlu supaya dapat dimanfaatkan kembali dengan cepat dan mudah untuk memenuhi berbagai kebutuhan. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 3).

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

DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. DBMS merupakan antarmuka pengguna basis data (baik merupakan pengguna langsung maupun aplikasi) dengan data yang tersimpan. Penyimpanan data oleh DBMS disesuaikan dengan bentuk model datanya, beberapa contoh DBMS adalah *PostgreSQL*, *MYSQL*, *DB2*, *Oracle*, *SQLserver* dan lain lain. Sebelum ada DBMS, data pada umumnya disimpan dalam bentuk *flat file*, yaitu *file teks* yang ada pada sistem operasi. Sampai sekarang pun masih banyak aplikasi yang

menyimpan dalam bentuk *flat file* secara langsung. Contoh yang sederhana adalah misalnya dalam sistem operasi linux terdapat *file password* yang digunakan untuk menyimpan nama penggunanya. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 5).

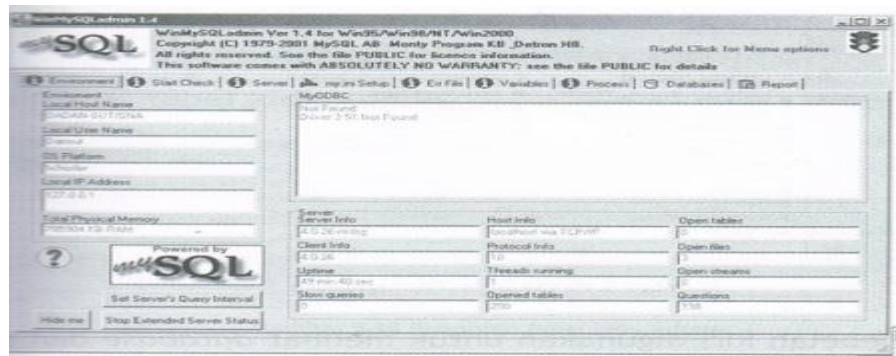
II.6. *MySQL*

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya).

Database digunakan untuk penyimpanan data, demikian pula dengan *MYSQL*. Kita akan memanggil data pada *MYSQL* melalui *PHP*, kemudian hasilnya dikirim kekomputer *klien* untuk ditampilkan pada *browser*. Data pada *MYSQL* dapat dipanggil, dihapus atau di tambah melalui *query* .(Dadan Sutisna : 2008 : 46)

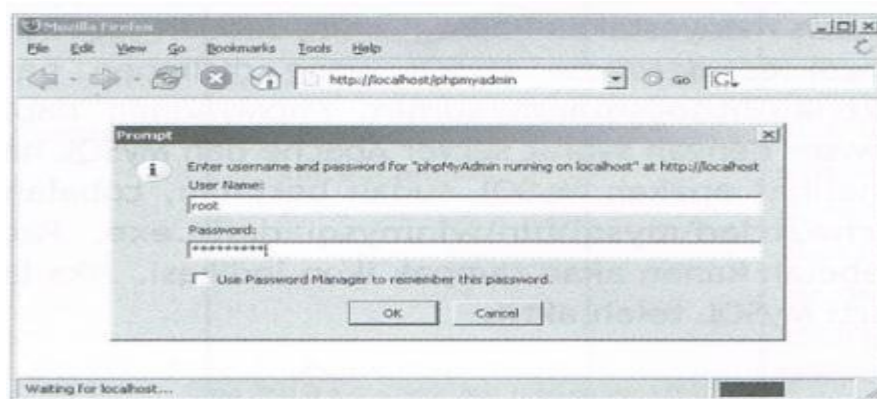
Langkah untuk menjalankan *webservice local* adalah :

1. Buka program *browser*, misalnya ***Mozilla Firefox*** atau ***Internet Explorer***.
2. Dalam jendela *browser*, ketil ***localhost*** dibagian ***Address***, kemudian tekan ***Enter***. Jika proses instalasi benar maka akan tampil halaman *localhost* dapat dilihat pada gambar II.5.



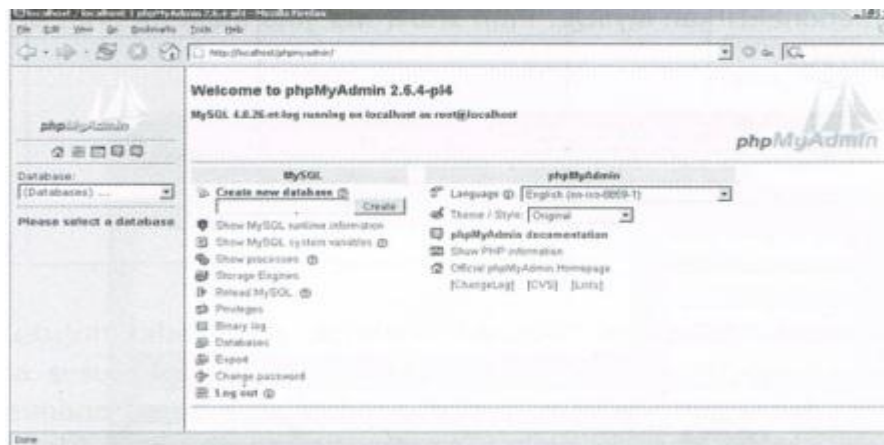
Gambar II.5. Halaman Awal Localhost
Sumber : (Dadan Sutisna : 2008 : 57)

3. Gambar II.5 juga menampilkan informasi 4 komponen pendukung *webserver* yang telah terinstal pada komputer sekaligus menampilkan versi yang digunakan.
4. Berikutnya program akan meminta memasukkan *user* dan *password* dari server *MySQL*. Ketik *user* nama **root** pada kotak **User Name**, dan masukkan *password*-nya pada kotak **Password** sesuai dengan *password* pada saat penginstalan. Tampilan konfirmasi permintaan *username* dan *password* server *MySQL* dapat dilihat pada gambar II.6.



Gambar II.6. Konfirmasi Permintaan Username dan Password Server MySQL
Sumber : (Dadan Sutisna : 2008 : 58)

Jika *username* dan *password* yang dimasukkan benar, maka tampil halaman *phpMyAdmin* tampak pada gambar II.7.



Gambar II.7. Tampilan Halaman PhpMyAdmin

Sumber : (Dadan Sutisna : 2008 : 58)

II.7. Konsep DataBase

Dalam konsep perancangan *database* terdapat pendekatan yang perlu diperhatikan yaitu sebagai berikut :

II.7.1 Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel yang menunjukkan entitas sekaligus relasinya.

Tahap normalisasi terdiri dari beberapa bentuk :

1. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal Kesatu (1NF/ *First Normal Form*)

Bentuk normal kesatu mempunyai ciri : setiap data dibentuk dalam *file file* (*file* datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa "atomic value". Tidak ada set atribut yang berulang atau atribut bernilai ganda (*multivalued*). Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukan pecahan kata sehingga artinya lain. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi, maka ia tidak memiliki sifat induknya.

3. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Bentuk normal kedua memiliki syarat : bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung fungsi pada kunci utama/*primary key* sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya. Tahap normalisasi terdiri dari beberapa bentuk:

4. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Untuk menjadi bentuk normal ketiga, relasi haruslah dalam bentuk normal kedua dan semua atribut dalam primer tidak punya hubungan yang transif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh. Contoh pada bentuk kedua di atas termasuk juga bentuk normal ketiga seluruh atribut yang ada disitu bergantung penuh pada kunci primernya.

5. Bentuk Normal *Boyce Codd* (BCNF/ *Boyce Codd Normal Form*)

Boyce Codd Normal Form mempunyai paksaan yang lebih kuat dari pada bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey*. (Ema Utami dan Anggit Dwi Hartanto ; 2012 :40-42).

II.7.2 Contoh Kasus Normalisasi

Untuk lebih jelasnya dapat dilihat normalisasi pada kasus pengobatan di rumah sakit sebagai berikut :

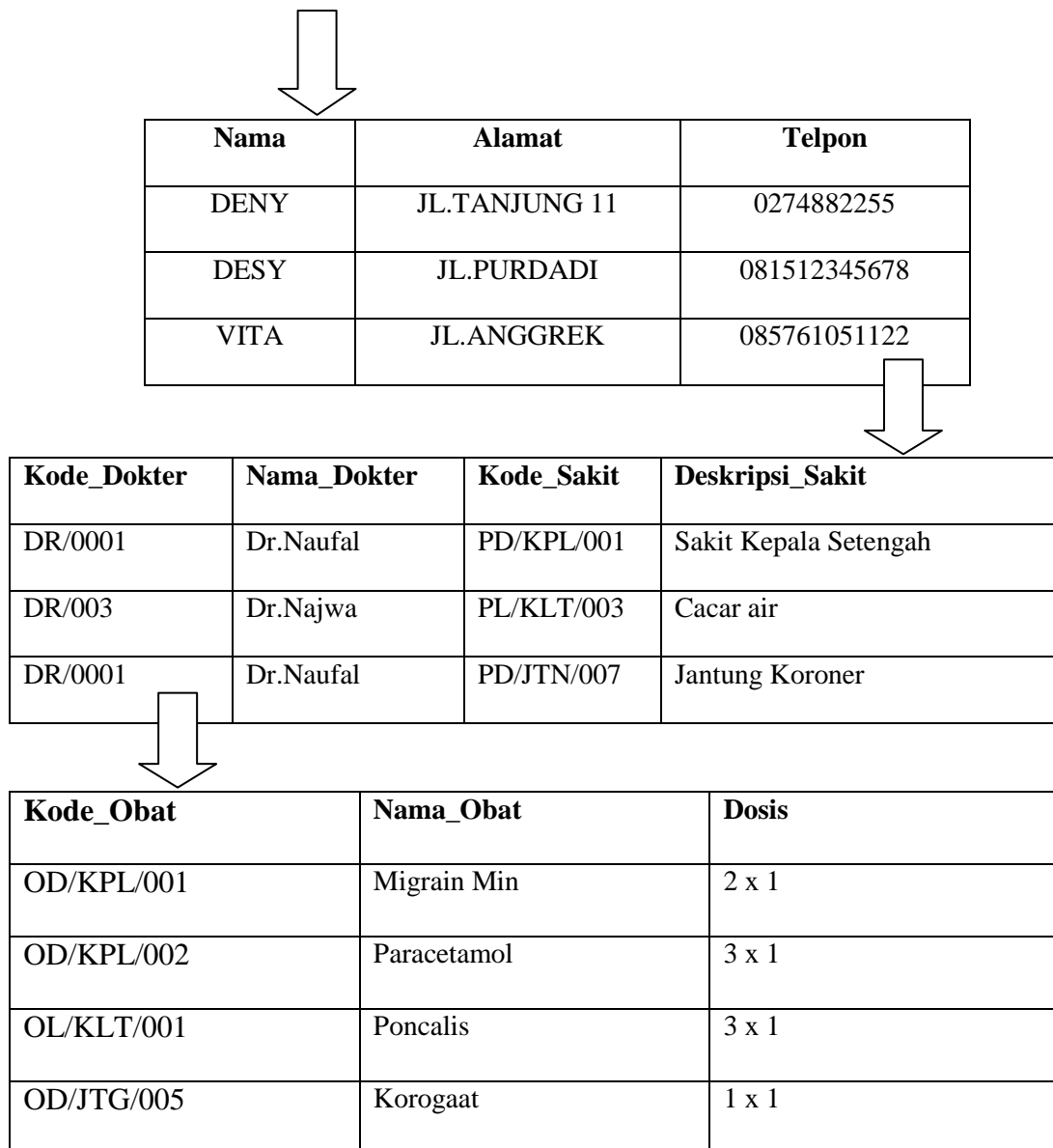
Tabel II.1 Bentuk Unnormal Form

No_Pasien
Nama_Pasien
No_Pendaftaran
Tanggal_Daftar
Alamat
Kode_Pos
Telpon
Tanggal_Periksa 1...n
Kode_Dokter 1...n
Nama_Dokter 1...n
Kode_Sakit 1...n
Deskripsi_Sakit 1...n
Kode_Obat 1...n
Nama_Obat 1...n
Dosis 1...n

Sumber (Ema Utami Dan Anggit Dwi Hartanto ; 2012 : 43)

Tabel II.2 Contoh Bentuk Unnormal Form

No_Pendaftaran	Tgl_Daftar	Tgl_Periksa	No_Pasien
D/01/0001	15/07/2005	19/07/2005	PS/0001
D/01/0002	18/08/2005	18/08/2005	PS/0002
D/01/0004	29/09/2005	01/10/2005	PS/0004



Sumber (Ema Utami Dan Anggit Dwi Hartanto ; 2012 : 44)

Tabel di atas dalam bentuk tidak normal karena :

1. Mempunyai bentuk *non flat file* karena data disimpan sesuai dengan datangnya sehingga tidak memiliki struktur yang sama/tertentu, terjadi duplikasi atau tidak lengkap.

2. Memuat set atribut berulang (*non single value*).
3. Memuat atribut *non atomic value*.

II.8 E-R Diagram (ERD)

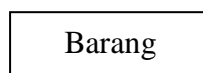
ERD merupakan notasi grafis dalam pemodelan data *konseptual* yang mendeskripsikan hubungan antar penyimpanan. *ERD* juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lainnya dalam dunia nyata. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 18)

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar dua data. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. *Entity*

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh adalah barang, pemasok, pekerja dan lain-lain.

Seandainya adalah A maka barang A adalah isi dari barang, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan. Karena itu harus dibedakan antara entitas sebagai bentuk umum dari deskripsi tertentu dan isi entitas seperti A dan B dalam contoh diatas. Entitas dapat digambarkan dalam bentuk persegi empat.

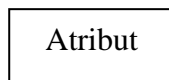


Gambar II.8. Entitas

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 19)

2. *Atribut*

Entitas mempunyai elemen yang disebut *atribut* dan berfungsi mendeskripsikan karakter *entitas*, misalnya atribut nama barang dari *entitas* barang. Setiap *ERD* bisa berisi lebih dari satu *atribut*. *Entitas* digambarkan dalam bentuk elips.

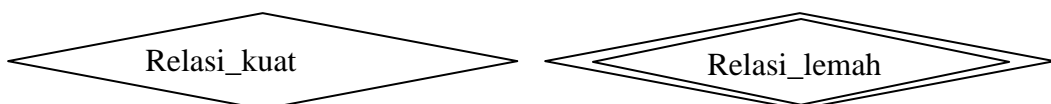


Gambar II.9 Atribut

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 20)

3. Hubungan / *relationship*

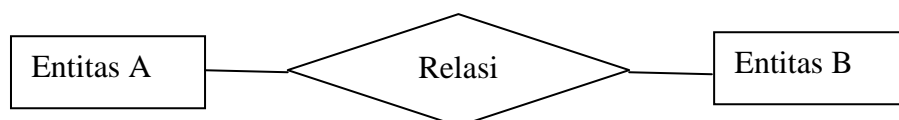
Belah ketupat merupakan penggambaran hubungan (relasi) antarentitas atau sering disebut kerelasian. Ada dua macam penggambaran relasi yaitu relasi kuat dan relasi lemah. Relasi kuat biasanya menghubungkan antarentitas kuat, sedangkan relasi lemah untuk menghubungkan antara entitas kuat dengan entitas lemah. *Relationship* digambarkan sebagai berikut :



Gambar II.10. Relationship

Sumber : ((Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

Untuk menghubungkan entitas-kerelasian-entitas digunakan garis lurus seperti gambar berikut ini :

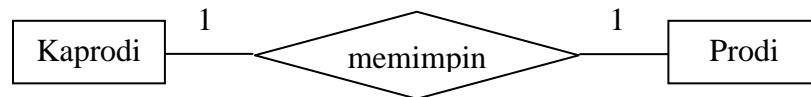


Gambar II.11. Kerelasian Antar Entitas

Sumber : ((Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

Jenis-jenis hubungan:

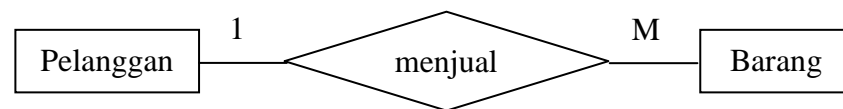
- a. Satu ke satu, misalnya suatu perusahaan mempunyai aturan satu supir hanya boleh menangani satu kendaraan karena alasan tertentu.



Gambar II.12. Relational 1 to 1

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

- b. Satu ke banyak atau banyak ke satu, misalnya suatu perusahaan selalu berasumsi bahwa satu pelanggan dapat membeli banyak barang.



Gambar II.13. Relational 1 to Many

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 25)