

BAB III

LANDASAN TEORI

3.1 Pengertian Sistem Informasi

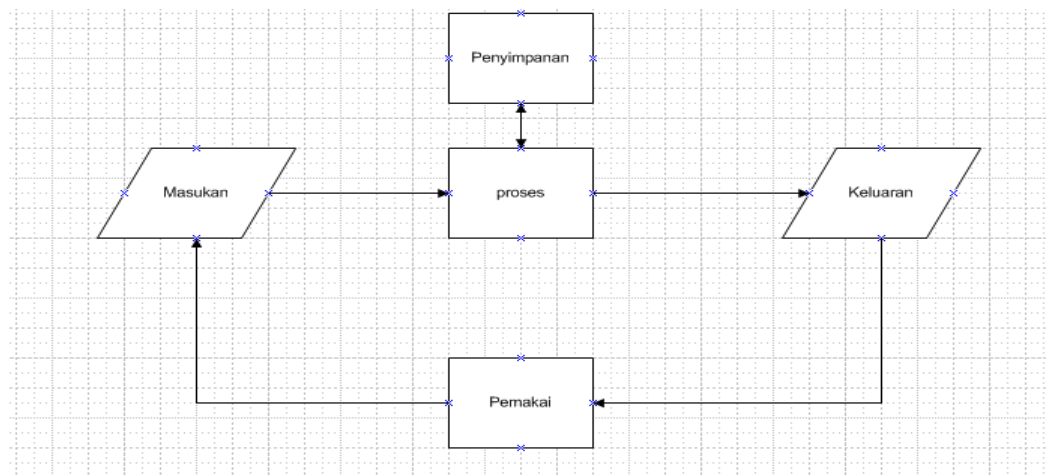
Dalam merancang sistem perlu dikaji tentang konsep dan definisi dari sistem. Menurut Davis (1984: 68) sebuah sistem terdiri dari bagian-bagian saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Berarti, sebuah sistem bukanlah seperangkat unsur yang tersusun secara tak teratur, tetapi terdiri dari unsur yang dapat dikenal sebagai saling melengkapi karena satunya maksud, tujuan, atau sasaran. Sistem fisik lebih dari sekedar bentuk konseptual, karena dapat memperlihatkan kegiatan atau perilaku.

Menurut James A.Hall (2004), sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama. Sedangkan informasi adalah fakta yang menyebabkan penggunanya melakukan tindakan yang tidak akan dapat dilakukannya, jika tidak ada fakta tersebut. Jadi sistem informasi adalah serangkaian prosedur formal di mana data dikumpulkan, diproses menjadi informasi dan didistribusikan ke para pengguna.

Menurut Hartono (1999:1) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan. Berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Menurut Hartono (1999:8) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Data merupakan bentuk yang masih mentah yang belum

dapat bercerita banyak, sehingga perlu lebih lanjut “Sistem Informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dalam laporan-laporan yang diperlukan.”(Leitch, 1983)

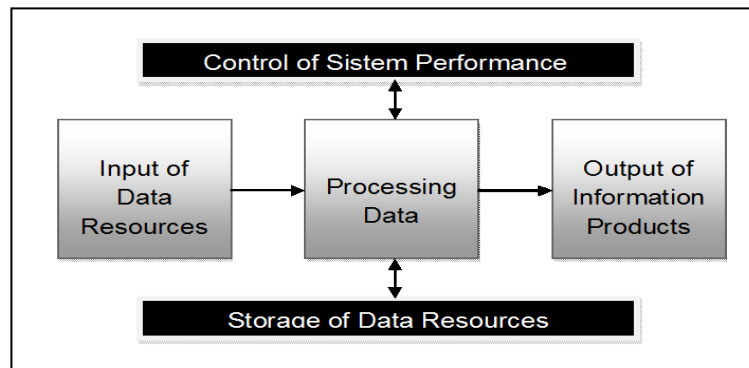
Sedangkan menurut Lani Sidharta (1995), sistem informasi adalah sistem buatan manusia yang berisi himpunan terintegrasi dari komponen-komponen manual dan komponen-komponen terkomputerisasi yang bertujuan untuk mengumpulkan data, menyimpan data, memproses data, dan menghasilkan informasi untuk pemakai. Gambar 3.1 menunjukkan komponen fungsional dari sebuah sistem informasi.



Gambar 3.1 Komponen Sistem Informasi

Pengertian sistem informasi tergantung pada latar belakang cara pandang orang yang mencoba mendefinisikannya. Herlambang dan Tanuwijaya (2005: 47) mendefinisikan bahwa sistem informasi terdiri dari *input*, proses dan *output*, seperti

yang terlihat pada Gambar 3.1 Pada bagian proses terdapat hubungan timbal balik dengan 2 (dua) elemen, yaitu kontrol dari kinerja sistem dan sumber-sumber penyimpanan data. *Input* yang akan diproses berupa data, baik karakter-karakter huruf maupun berupa numerik. Saat ini data bisa berupa suara atau *audio* maupun gambar atau *video*. Data ini diproses dengan metode-metode tertentu dan akan menghasilkan *output* yang berupa informasi. Informasi yang dihasilkan dapat berupa laporan atau *report* maupun solusi dari proses yang telah dijalankan.



Gambar 3.2 Proses Sistem Informasi
(Sumber: Herlambang dan Tanuwijaya, 2005: 46)

3.2 Pengertian Sistem Informasi Penmaru

Sistem informasi penmaru adalah sistem yang berfungsi untuk pengolahan data calon mahasiswa baik tamu pengunjung, calon mahasiswa daftar, serta calon mahasiswa yang melakukan registrasi. Sistem ini bersifat *client-server* karena menggunakan *database* MySQL sehingga untuk mengembangkan jaringan akan lebih mudah untuk diimplementasikan.

3.3 Analisa dan Perancangan Sistem

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Menurut Kendall (2003:7), Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.


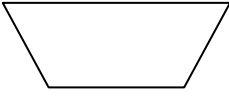

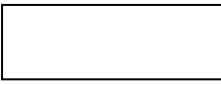

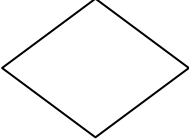

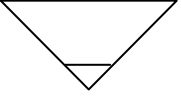
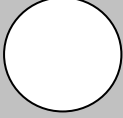
Berikut ini adalah proses dalam analisis dan perancangan sistem:

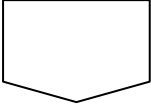
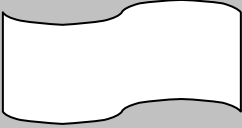
3.3.1 System Flow

System flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Hartono, 1998: 10).

Terdapat beberapa simbol yang digunakan untuk merancang sebuah desain dari sistem, simbol-simbol *system flow* tersebut dapat dilihat pada tabel 3.3 di bawah ini:

Tabel 3.3 Simbol-Simbol *System Flow*

Simbol	Keterangan/ Fungsi
	Terminator Sebagai tanda dimulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu sistem.
	Manual Operation Menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan komputer sebagai mediana (menggunakan proses manual).
	Document <i>Document</i> merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik.
	Process Merupakan sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi.
	Database <i>Database</i> digunakan sebagai media penyimpanan data yang bersifat komputerisasi.
	Decision Merupakan operator logika digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar atau salah.
	Manual Input Melakukan proses <i>input</i> ke dalam <i>database</i> melalui <i>keyboard</i> .
	Off-line Storage Merupakan media penyimpanan dokumen secara manual (arsip).
	On-page Reference Merupakan simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada letaknya terlalu jauh.

Simbol	Keterangan/ Fungsi
	<p>Off-page Reference Simbol ini digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda.</p>
	<p>Paper Tape Merupakan simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (misal: uang) dalam transaksi yang ada pada sistem yang dirancang.</p>


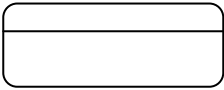
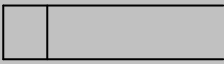

3.3.2 Data Flow Diagram

Pada tahap ini, penggunaan notasi dapat membantu komunikasi dengan pemakai/*user* sistem untuk memahami sistem tersebut secara logika. Diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem ini dikenal dengan nama Diagram Arus Data (*Data Flow Diagram*). DFD berfungsi untuk menggambarkan proses aliran data yang terjadi di dalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan untuk melakukan dekomposisi, mempartisi atau membagi sistem kedalam bagian-bagian yang lebih kecil dan yang lebih sederhana. DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2003:241).

Menurut Leman (1998) *Data Flow Diagram* (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. Kita dapat menggunakan DFD untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru.

Simbol-simbol yang digunakan dalam mendeskripsikan DFD dapat dilihat pada tabel 3.4

Tabel 3.4 Simbol-Simbol *Data Flow Diagram*

Simbol	Keterangan/ Fungsi
	Simbol Entitas Eksternal/ Terminator Menggambarkan asal atau tujuan data di luar sistem.
	Simbol Persegi/ Lingkaran Menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar.
	Simbol File/ Data Store Menggambarkan tempat aliran data disimpan,
	Simbol Aliran Data Menggambarkan aliran data.

3.3.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefinisikan didalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*.

Rahmad (2005) menyatakan bahwa ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada dasarnya ada 3 (tiga) simbol yang digunakan, yaitu:

a. *Entity*

Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar *atribut* diwakili oleh simbol *elips*.

c. Hubungan/ Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

1. *One to One*

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2. *One to Many*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. *Many to Many*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Begitu juga pada entitas B, dapat berhubungan dengan banyak entitas A.

ERD ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*. Untuk itu ERD dibagi menjadi dua jenis model, yaitu:

a. *Conceptual Data Model (CDM)*

Conceptual Data Model (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

b. *Physical Data Model (PDM)*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisik.

3.4 Program Penunjang

Untuk membuat sistem informasi penmaru melalui kriteria lulusan sekolah dari luar Pondok Pesantren Mamba'us Sholihin Suci Manyar Gresik, dibutuhkan beberapa perangkat lunak untuk memudahkan perancangan desain maupun sistem. Perangkat Lunak tersebut antara lain:

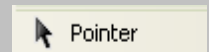

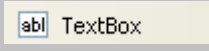
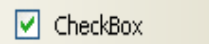

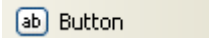


3.4.1 Power Designer

Power Designer merupakan suatu *tool* berupa *software* untuk mendesain sistem dan rancangan ERD yang dikembangkan oleh Sybase Inc. Terdapat dua model, yaitu *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)*.

3.4.2 Microsoft Visual Basic.Net 2008

Microsoft Visual Basic .Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .Net Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan *tool* ini, para pembuat program dapat membangun aplikasi *Windows Form*. Pada tabel 3.5 dapat dilihat berbagai macam komponen yang digunakan untuk membangun sebuah program dalam Microsoft Visual Basic .Net.

Tabel 3.5 Komponen Microsoft Visual Basic .Net 2008

Bentuk	Keterangan
 Pointer	Penunjuk control yang berfungsi memindahkan atau mengubah ukuran control yang ada pada form.
 Label	Menampilkan teks, tetapi pemakai tidak bias berinteraksi dengannya.
 TextBox	Menempatkan teks pada form dan pemakai dapat mengedit teks tersebut.
 CheckBox	Membuat kotak <i>check</i> yang dapat memilih satu atau banyak keadaan.
 RadioButton	Memilih dan mengaktifkan satu keadaan dari banyak pilihan yang ada.
 Button	Membuat tombol pelaksana suatu perintah atau tindakan saat digunakan.
 DataGridView	Menampilkan data dalam bentuk baris dan kolom.
 ComboBox	Tempat menginputkan pilihan atau memilih suatu pilihan dengan <i>Drop-Down-List</i> .

3.4.3 SQL Server 2005

SQL Server 2005 Express Edition adalah produk *database* yang mudah dipakai dan dibuat berdasarkan teknologi SQL Server 2005. Kemudahan pakainya bisa dilihat mulai dari antar muka program setup yang memandu pengguna dalam melalui proses instalasi. Tool berbasis GUI (graphical user interface) yang hadir menyertai SQL Server 2005 Express Edition memang sangat minim, hanya ada SQL Server Configuration Manager dan SQL Server Surface Area Configuration. Akan tetapi, kita bisa men-*download tool* untuk mengelola *instance database* SQL Server 2005 Express Edition, yaitu SQL Server Management Studio Express Edition (SSMS-EE), secara cuma-cuma integrasi yang sangat erat antara SQL Server 2005 Express Edition dengan Visual Studio 2008 juga akan sangat membantu kita dalam mengembangkan aplikasi *database*.

SQL Server 2005 Express Edition menggunakan *engine database* dan API untuk mengakses data yang sama seperti versi SQL Server 2005 lainnya. Perbedaannya, antara lain adalah:

1. Tidak ada dukungan untuk fitur-fitur yang dibutuhkan oleh kalangan *enterprise*.
2. Hanya mendukung satu *processor*.
3. Memory untuk *buffer pool* dibatasi hingga 1 GB.
4. Ukuran maksimal *database* dibatasi sampai 4 GB saja.

Absennya fitur-fitur untuk kalangan *enterprise* seperti *high availability* dan *business intelligence* di SQL Server 2005 Express Edition bukanlah masalah besar, karena aplikasi yang dikembangkan di atas SQL Server 2005 Express Edition dapat bekerja secara seamless di SQL Server 2005 Workgroup Edition, *Standard Edition*,

dan tentunya juga di SQL Server 2005 Enterprise Edition. Dengan demikian, *scalability* dari aplikasi *database* yang dikembangkan menggunakan SQL Server 2005 meneruskan tradisi dalam menghadirkan produk *database* yang mudah dipakai dan tersedia untuk semua kalangan, Microsoft menghadirkan SQL Server 2005 Express Edition sebagai produk penerus Microsoft SQL Server 2000 Desktop Engine (MSDE 2000). Sebagai produk yang disediakan secara cuma-cuma, apa saja kelebihan dan kekurangannya Express Edition dapat terjamin dan implementasinya dapat disesuaikan dengan kebutuhan saat ini.

3.5 Database

Menurut Yuswanto (2005:2), *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database Relational* dan *Non Relational*. Pada *database Non Relational*, sebuah *database* hanya merupakan sebuah *file*.

Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah – masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan dalam mengakses data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai),

masalah keamanan (*security*), masalah integrasi (kesatuan), dan masalah data *independence* (kebesaran data).

Database adalah kumpulan *file-file* yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari setiap *file* yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan, instansi. Dalam merancang aplikasi *database*, selain harus mengatur rutin program agar kinerjanya maksimum, juga harus memperhatikan tata letak penyimpanan data, baik secara fisik maupun *logis*. Perancangan *database* yang baik memberikan waktu pencarian minimum dalam menentukan lokasi *record-record* tertentu. *Database* yang juga harus menyimpan data seefisien mungkin agar *database* tidak mengembang terlalu besar. Dengan demikian data dapat diperbarui dengan mudah.

Perencanaan *database* yang baik melibatkan 7 aktivitas kunci yaitu :

1. Membuat model aplikasi.
2. Menentukan data yang diperlukan aplikasi.
3. Mengorganisir data ke dalam tabel.
4. Menetapkan hubungan antar tabel.
5. Menetapkan kebutuhan *indeks*, validasi dan integritas bagi data.
6. Membuat penyimpanan *query* yang perlu untuk aplikasi.
7. Meninjau ulang rancangan *database*.