

BAB II

LANDASAN TEORI

A. Pengertian Sistem Informasi Manajemen

Sistem informasi dapat didefinisikan sebagai satu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Para pemakai biasanya tergabung dalam suatu entitas organisasi formal, seperti Departemen atau Lembaga suatu Instansi. Sistem informasi memuat berbagai informasi penting mengenai orang, tempat, dan segala sesuatu kejadian yang ada di dalam atau di lingkungan sekitar organisasi. Informasi sendiri mengandung suatu arti yaitu data yang telah diolah kedalam suatu bentuk yang lebih memiliki arti dan dapat digunakan untuk memberikan informasi yang benar kepada yang membutuhkan. Data sendiri merupakan fakta-fakta yang mewakili suatu keadaan, kondisi, atau peristiwa yang terjadi atau ada di dalam atau di lingkungan fisik organisasi. Data tidak dapat langsung digunakan untuk kepentingan tertentu, melainkan harus diolah lebih dahulu agar dapat dipahami, lalu dimanfaatkan dalam pengambilan data informasi yang benar dan tepat. Informasi harus dikelola dengan baik dan memadai agar memberikan manfaat yang maksimal. Penerapan sistem informasi di dalam suatu organisasi dimaksudkan untuk memberikan dukungan informasi yang dibutuhkan, khususnya

Sistem informasi yang digunakan oleh para pengguna dari berbagai tingkatan manajemen ini biasa disebut sebagai: Sistem Informasi Manajemen.

Sistem informasi manajemen merupakan serangkaian dari sub-sistem informasi yang menyeluruh dan terkoordinasi yang secara rasional mampu mentransformasikan data sehingga menjadi informasi yang terintegrasi (terpadu). Sebuah sistem terpadu berdasarkan pada anggapan bahwa harus ada integrasi antara data dan pengolahan. Integrasi data dicapai melalui "data base". Pada sebuah sistem pengolahan informasi, "data base" terdiri dari semua data yang dapat dijangkau oleh sistem. Pada SIM (Sistem Informasi Manajemen) berdasarkan komputer, istilah "data base" biasanya dipakai khusus untuk data yang dapat dijangkau secara langsung oleh komputer. Manajemen sebuah "data base" adalah sebuah sistem perangkat lunak komputer yang disebut sebagai sebuah sistem manajemen basis data.

Sistem informasi manajemen mengandung tiga aktivitas dasar di dalamnya, yaitu: aktivitas masukan (*input*), pemrosesan (*processing*), dan keluaran (*output*). Tiga aktivitas dasar ini menghasilkan informasi yang dibutuhkan organisasi untuk pengambilan informasi yang jelas dan tepat. Masukan berperan di dalam pengumpulan bahan mentah (*raw data*), baik yang diperoleh dari dalam maupun dari lingkungan sekitar organisasi. Pemrosesan berperan untuk mengkonversi bahan mentah menjadi bentuk yang lebih memiliki arti. Sedangkan, keluaran dimaksudkan untuk mentransfer informasi yang diproses kepada pihak-pihak yang akan menggunakannya. Sistem informasi juga membutuhkan umpan balik (*feedback*) yaitu

untuk dasar evaluasi dan perbaikan di tahap input berikutnya. Dewasa ini, sistem informasi yang digunakan lebih berfokus pada sistem informasi berbasis komputer (*computer-based information system*). Harapan yang ingin diperoleh di sini adalah bahwa dengan penggunaan teknologi informasi atau sistem informasi berbasis komputer, informasi yang dihasilkan dapat lebih akurat, berkualitas, dan tepat waktu, sehingga pengambilan data informasi dapat lebih efektif dan efisien. Meskipun sistem informasi berbasis komputer menggunakan teknologi komputer untuk memproses data menjadi informasi yang memiliki arti, ada perbedaan yang cukup tajam antara komputer dan program komputer di satu sisi dengan sistem informasi di sisi lainnya. Komputer dan perangkat lunak komputer yang tersedia merupakan fondasi teknis, alat, dan material dari sistem informasi modern. Komputer dapat dipakai sebagai alat untuk menyimpan dan memproses informasi. Program komputer atau perangkat lunak komputer merupakan seperangkat instruksi operasi yang mengarahkan dan mengendalikan pemrosesan informasi.

B. Teknik Perancangan Database

Teknik Perancangan Database merupakan cara untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem.

Perancangan sistem terjadi pada dua tingkat, yaitu :

- Pada tingkat pertama, perencanaan sistem, analisis dan rancangan umum

... ..

database ini melibatkan tahap *front-end*, bebas dari perancangan database tertentu atau Database Management System (DBMS).

- Pada tingkat kedua, rancangan umum, seperti diagram entitas relasi tingkat tinggi, ditransformasikan (atau didekomposisikan) ke dalam perancangan database rinci untuk sebuah DBMS tertentu yang akan digunakan untuk mengimplementasikan sistem total. Tiga model database yang cukup dikenal adalah :

- a. Model Hierarkikal
- b. Model Jaringan
- c. Model Relasional

Pada masa lalu banyak penjual (*vendors*) menawarkan Database Management Systems (DBMS) yang berdasarkan pada Model Hierarkikal dan Model Jaringan. Saat ini Model Relasional adalah dominan. Karena itu hampir semua penjual perangkat lunak database menawarkan produk perangkat lunak Relational Database Management Systems (RDBMS). RDBMS dibuat dengan struktur tiga skema, Struktur lapisan, ini mendefinisikan data perusahaan pada tingkat yang berbeda.

Skema Eksternal mendefinisikan bagaimana pemakai mengakses dan melihat output dari RDBMS, bebas dari bagaimana data disimpan atau diakses secara fisik. Akses dan manipulasi seperti ini dilaksanakan oleh pemakai dengan memperkerjakan bahasa prosedural, seperti COBOL atau bahasa query, seperti Structured Query

Skema Konseptual yang mendefinisikan model database relasional terdiri dari sekumpulan tabel yang dinormalisasi. Skema konseptual adalah rancangan dari database yang merupakan subyek utama dari bab ini.

Skema Internal terdiri dari organisasi fisik dari data (mis. sekuensial, indeks sekuensial, langsung) dalam hal struktur fisik data dan metode-metode pengaksesan dari sistem operasi komputer.

Sifat-sifat Tabel :

1. Duplikasi baris tidak diperbolehkan. Untuk melaksanakan sifat ini, harus terdapat paling sedikit satu atribut atau kombinasi beberapa atribut yang mengidentifikasi secara unik tiap baris dari tabel. Atribut atau kombinasi beberapa atribut yang melaksanakan tugas ini disebut Kunci Primer (*Primary Key*). Contoh : Nomor_Mahasiswa, adalah kunci primer yang mengidentifikasi tiap mahasiswa secara unik.
2. Database relasional adalah nilai *Primary Key* tidak boleh mempunyai duplikat atau NIL (NULL, yaitu nilai tidak diketahui).
3. Keterhubungan (*relationship*) antara dua tabel. Jika Tabel R2 mempunyai sebuah Kunci Asing (*Foreign Key*) yang cocok dengan kunci primer dari Tabel R1, maka untuk setiap nilai *Foreign Key* harus terdapat sebuah nilai

C. Entity Relationship Diagram (ERD)

Model Entity Relationship

Model entity relationship adalah suatu penyajian data dengan menggunakan Entity dan Relationship.

I. Entity

- Entity adalah obyek yang dapat dibedakan dalam dunia nyata
- Entity set adalah kumpulan dari entity yang sejenis
- Entity set dapat berupa :
 - Obyek secara fisik : Rumah, Kendaraan, Peralatan
 - Obyek secara konsep : Pekerjaan , Perusahaan, Rencana

II. Relationship

Relationship adalah hubungan yang terjadi antara satu atau lebih entity contohnya hubungan antara entity dosen dan entity matakuliah seperti terlihat pada gambar 2.1 di bawah ini :



Gambar 2.1 Contoh Hubungan Antar Entity

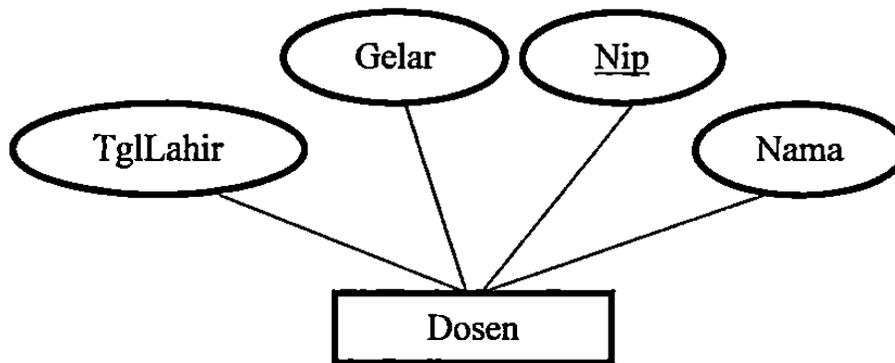
- Atribut

Atribut adalah karakteristik dari entity atau relationship, yang menyediakan

Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

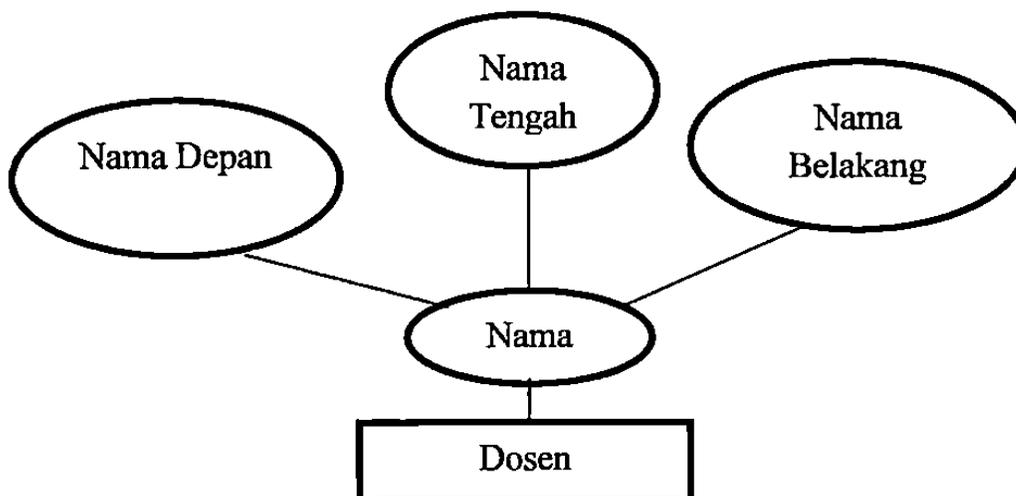
Jenis-jenis atribut :

- Key : Atribut yang digunakan untuk menentukan suatu entity secara unik.
- Atribut Simple : Atribut yang bernilai tunggal.
- Atribut Multivalued : Atribut yang memiliki sekelompok nilai untuk setiap instan entity. Contohnya seperti terlihat pada gambar 2.2 di bawah :



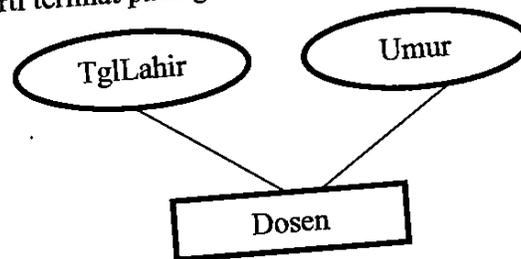
Gambar 2.2 Atribut Yang Memiliki Sekelompok Nilai Untuk Setiap Entity

- Atribut Composite : Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu, seperti terlihat pada gambar 2.3 di bawah:



Gambar 2.3 Atribut Yang Terdiri Dari Beberapa Atribut Yang Lebih Kecil

- Atribut Derivatif : Suatu atribut yang dihasilkan dari atribut yang lain.
Contohnya seperti terlihat pada gambar 2.4 di bawah :



Gambar 2.4 Atribut Yang Dihasilkan Dari Atribut Yang Lain

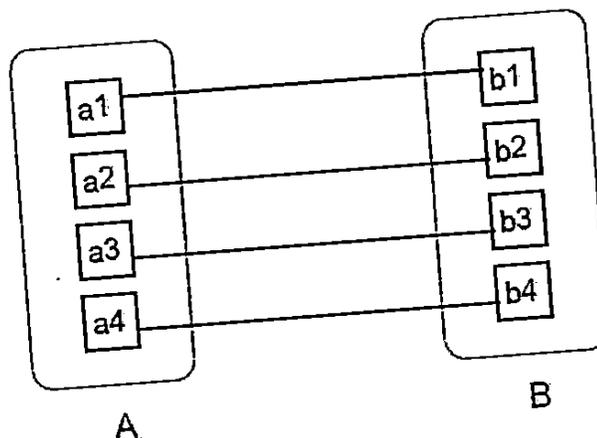
Cardinality Ratio Constraint

Menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

Cardinality Ratio terdiri dari 4 jenis yaitu :

- 1 : 1 (One-To-One)

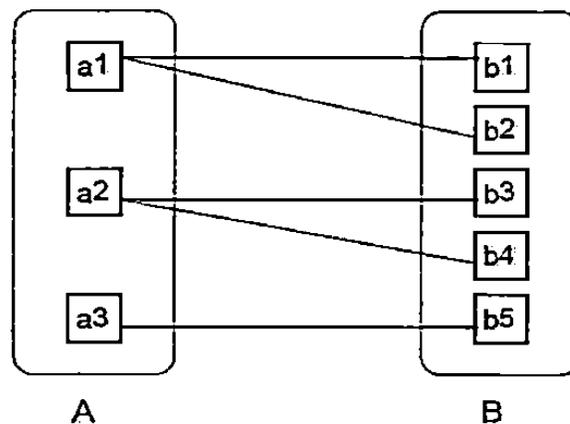
Sebuah entity A diasosiasikan pada sebuah entity B, dan sebuah entity B diasosiasikan dengan paling banyak sebuah entity A. Untuk lebih jelasnya lihatlah gambar 2.5 di baah ini :



Gambar 2.5 Cardinality Ratio 1 : 1 (One-To-One)

- 1 : N (One-To-Many)

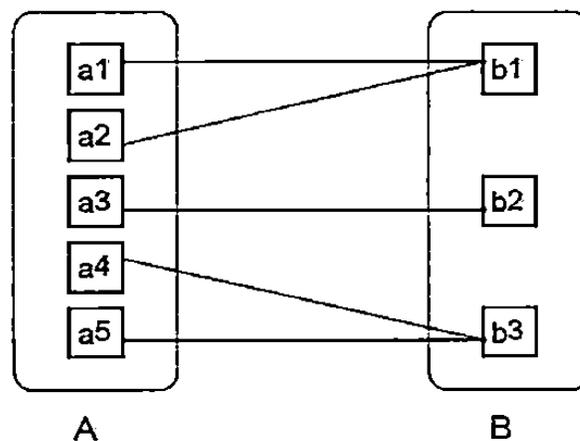
Sebuah entity A diasosiasikan dengan sejumlah entity B, tetapi entity B dapat diasosiasikan paling banyak satu entity A, seperti yang terlihat pada gambar 2.6 di bawah ini :



Gambar 2.6 Cardinality Ratio 1 : N (One-To-Many)

- N : 1 (Many-To-One)

Suatu entity A dapat diasosiasikan dengan paling banyak sebuah entity B, tetapi entity B dapat diasosiasikan dengan sejumlah entity di A, seperti terlihat pada gambar 2.7 di bawah ini :



Gambar 2.7 Cardinality Ratio N : 1 (Many To One)

Ada beberapa langkah dalam proses pembuatan suatu program atau software diantaranya adalah :

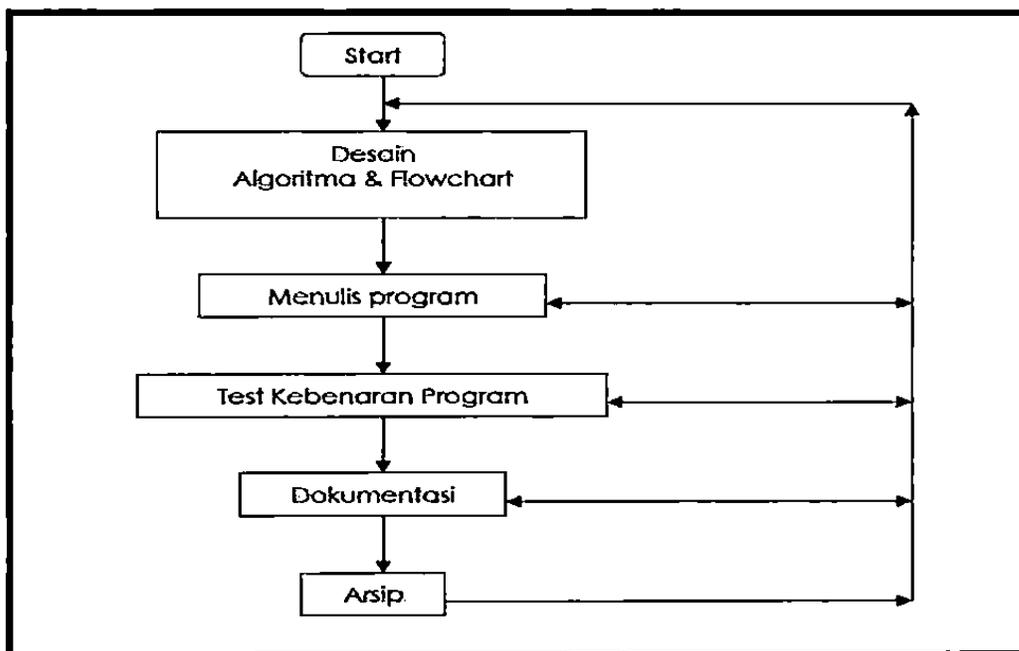
I. Mendefinisikan masalah dan menganalisisnya.

Langkah awal yang harus dilakukan adalah *mengidentifikasi masalah*.

Setelah identifikasi dari suatu permasalahan dari program maka dilakukan proses analisa untuk bisa menentukan beberapa hal antara lain :

- Tujuan dari pembuatan program.
- Parameter-parameter yang digunakan dalam program.
- Fasilitas apa saja yang akan disediakan oleh program.
- Kemudian menentukan metode atau algoritma apa yang akan diterapkan untuk menyelesaikan masalah tersebut.
- Terakhir menentukan bahasa program yang digunakan untuk pembuatan program.

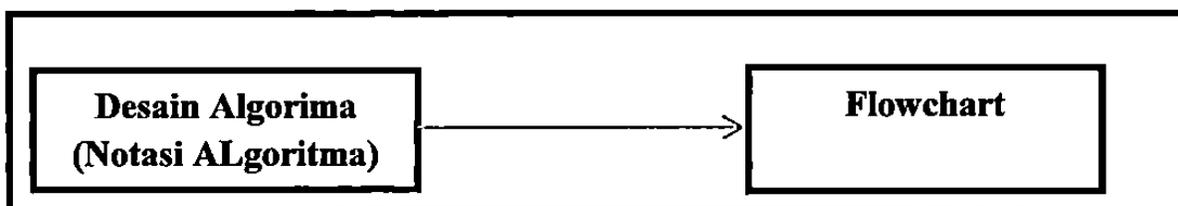
II. Merealisasikan dengan langkah-langkah berikut :



Gambar 2.9 Langkah-Langkah Untuk Merealisasikan Algoritma

Penjelasan mengenai langkah-langkah untuk merealisasikan dari algoritma pemrograman adalah sebagai berikut :

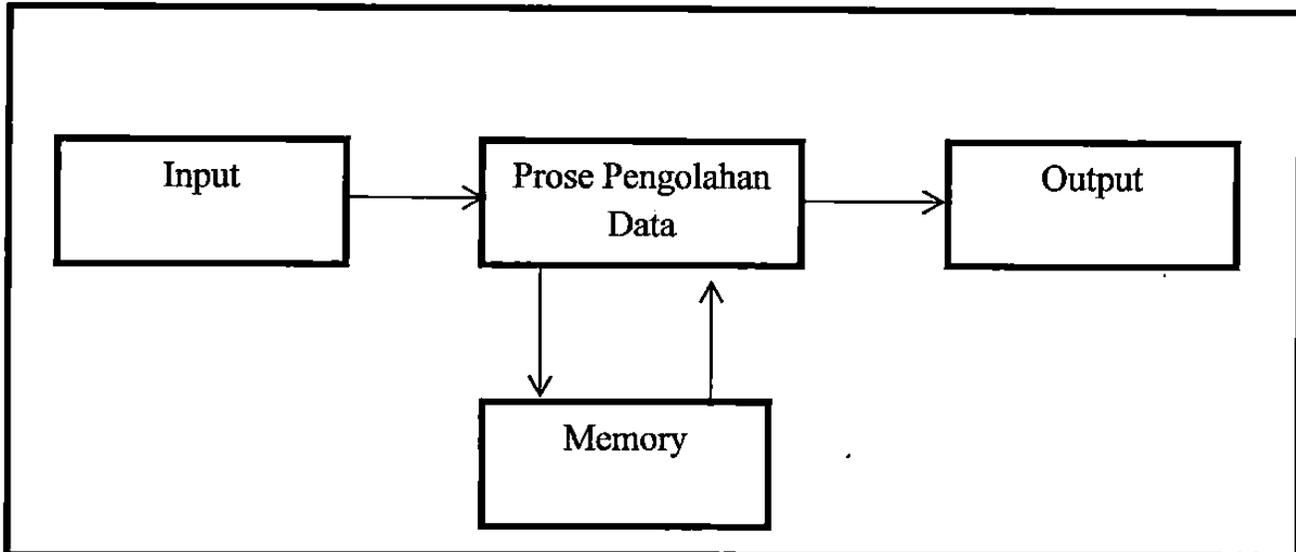
- *Start* : adalah untuk memulai algoritma program disini adalah proses menganalisa permasalahan seperti yang sudah di bahas pada bagian A sebelumnya. Pada bagian ini harus bisa melakukan identifikasi awal dari permasalahan/program yang akan dibuat, karena sangat menentukan desain dari algoritma logic yang akan dibuat.
- *Desain Algoritma dan Flowchart* : setelah permasalahan dapat diidentifikasi maka harus dilakukan desain dari algoritma yang akan diambil untuk penyelesaian permasalahan dan di tuliskan dalam bentuk Flowchart program. Untuk desain algoritma terdapat beberapa tahap yang harus dilakukan, yaitu mulai pembentukan Notasi-Notasi Algoritma (Notasi 1,Notasi 2,Notasi 3,dst). Pada bagian ini merupakan hal yang paling penting dan awal dari belajar programming, karena melatih cara berfikir secara algoritma logic bukan hanya secara logika. Jika digambarkan secara diagram merupakan hal terpisah antara Desain Algoritma dan Flowchart seperti yang terlihat pada gambar 2.10 di bawah ini :



- *Menulis program* : Adalah menuliskan flowchart yang sudah dibuat dalam bentuk bahasa pemrograman. Dalam penulisan program terdapat aturan-aturan yang sudah ditentukan oleh produk bahasa pemrograman itu sendiri (*Delphi, Vb, C#, C++, Asembly, Pascal, dll*). *Beda bahasa program hanyalah beda bahasa yang penting desain algoritma dan flowchart sudah benar.*
- *Test Kebenaran Program* : Merupakan fungsi *debuging*, yaitu menjalankan program dan dilakukan pengecekan apakah program sudah berjalan seperti yang diinginkan atau belum, jika belum maka bisa dilakukan langkah-langkah awal dari proses. *Jika tes program masih salah ada beberap kemungkinan, yaitu :*
 - *Kesalahan Algoritma.*
 - *Kesalahan penulisan/sintak dalam penulisan program.*
- *Dokumentasi* : Dalam pembuatan program hal-hal yang berhubungan dengan variabel-variabel input ataupun variabel yang berubah-ubah harus di dokumentasikan dan menggunakan kata/kalimat yang umum, sehingga jika melakukan rebuild/perbaikan program akan mempermudah seorang programmer.
- *Arsip* : Semua proses diatas dari awal sampai akhir harus diarsipkan dan disimpan sebaikbaiknya, sehingga mempermudah seorang programmer jika mau

E. Struktur Program

Secara umum, struktur suatu program terdiri dari beberapa bagian seperti gambar.2.11. berikut:



Gambar 2.11 Struktur Program

1. Input

Bagian ini merupakan proses untuk memasukkan data ke komputer melalui device yang ada misalnya keyboard, mouse, scanner dll. Program melakukan proses membaca data yang akan diolah dari device tersebut.

2. Output

Bagian ini merupakan proses untuk menampilkan data yang telah diolah, melaporkan hasil pengolahan data melalui device seperti monitor, printer dll.

Program melakukan proses mencetak data ke device tersebut

3. Proses Pengolahan Data

Bagian ini merupakan proses mengolah data yang diinputkan dengan menerapkan metode-metode, teknik-teknik, algoritma-algoritma yang ada. Proses ini menghasilkan data output yang akan dioutputkan kepada pengguna program.

4. Penyimpanan Data (Memory)

Bagian ini merupakan proses menyimpan data dalam memori atau device penyimpanan data seperti disket, harddisk, CD dll