

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian tentang sistem peminjaman ruangan sebelumnya sudah pernah dilakukan oleh Rizky Ridho Prasetyo dan Rio Wirawan (2018) dengan judul Perancangan Sistem Informasi Peminjaman Ruangan Berbasis Web Pada Universitas Pembangunan Nasional “Veteran” Jakarta. Penelitian ini dilakukan karena sebelumnya proses peminjaman ruangan masih dilakukan secara manual. Dengan adanya sistem ini maka peminjaman ruang akan lebih mudah dan memberikan penyelesaian permasalahan yang ada. Metode yang digunakan dalam menganalisa permasalahan pada penelitian ini menggunakan metode PIECES (*PerForm ance, InFormation, Economic, Control, Efficiency, Service*). Menggunakan bahasa pemrograman PHP dan MySQL sebagai *database* sistemnya [4].

Penelitian lain tentang sistem peminjaman ruangan dilakukan juga oleh Anak Agung Raka P.W.A, Muhammad Huzaimi Maulana, Cindya Dewi Andini dan Faridatun Nadziroh (2018) dengan judul Sistem Peminjaman Ruangan Online (SPRO) Dengan Metode *Unfield Modeling Language* (UML). Sistem ini mempermudah dalam melakukan pengelolaan peminjaman secara online baik melalui desktop, smartphone ataupun tablet agar pihak internal kampus ataupun external kampus dapat lebih mudah untuk melakukan peminjaman ruang ataupun barang. Metode yang digunakan dalam pengembangan sistemnya adalah metode *Waterfall* dan metode *Unfield Modeling Language* (UML) untuk perancangan sistem serta metode pengujian menggunakan metode *Black Box* [5].

Dita Isnania Rahma (2019) melakukan penelitian dengan judul Sistem Peminjaman Alat Inventaris Dan Ruangan Laboratorium Fakultas Teknologi Informasi Dan Komunikasi Universitas Semarang. Sistem ini berfungsi untuk meningkatkan efisiensi kerja dalam pendataan inventori Teknisi Laboratorium Fakultas Teknologi Informasi Dan Komunikasi Universitas Semarang. Model pengembangan sistem yang digunakan dalam merancang sistem adalah model pengembangan *Waterfall* sedangkan perancangan dan analisa sistem menggunakan metode *Unfield Modeling Language* (UML). Sistem ini dikembangkan menggunakan PHP dan MySQL sebagai *database* penyimpanannya [6].

Pada tahun 2020 M. Nishom dan Dega Surono Wibowo juga melakukan penelitian yang berjudul Sistem Informasi Laboratorium Berbasis *Website* di Program Studi D IV Teknik InFormatika Politeknik Harapan Bersama Tegal. Sebelumnya terdapat berbagai permasalahan yang muncul karena belum adanya sistem yang terintegrasi untuk dapat melihat informasi jadwal penggunaan ruangan laboratorium, sistem peminjaman ruangan laboratorium dan permasalahan lainnya. Oleh karena itu dibuatlah sistem ini sebagai solusi untuk menangani permasalahan yang ada serta dapat membantu manajemen laboratorium dalam mengelola jadwal, manajemen alat dan manajemen pelaporan untuk memberikan pelayanan prima kepada *stakeholder*. Pengembangan sistem dalam penelitian ini menggunakan *framework* CI dan pemodelan sistem dilakukan menggunakan pendekatan UML yang bersifat *Object Oriented Design* serta metode *blackbox* untuk pengujian sistem [7].

Ali Mulyanto dan Arjun Gunawan (2021) melakukan penelitian dengan judul Implementasi Metode Prototype Pada Sistem Peminjaman Alat Kerja Berbasis Web di PT SK Metalindo. Sebelumnya sistem peminjaman alat kerja sudah ada di PT SK Matalindo tetapi masih menggunakan cara manual yang sering mengakibatkan salah pencatatan. Oleh karena itu dibangun sistem ini yang bertujuan sebagai sarana untuk mencatat data peminjaman alat kerja oleh karyawan sehingga menunjang kelancaran dalam pengelolaan alat kerja. Pengembangan sistem dalam penelitian ini menggunakan *framework* Laravel dengan menggunakan metode pengembangan sistem *prototype* [8].

Pada penelitian ini penulis bermaksud untuk merancang dan membangun sebuah Sistem Informasi Peminjaman Ruangan dan Peralatan di Politeknik Negeri Cilacap. Sistem ini dibangun berbasis *website* dengan menggunakan Bahasa pemrograman PHP dan metode pengembangan sistem *waterfall*. Sistem ini akan digunakan oleh Politeknik Negeri Cilacap dan dibuat berbasis *website*. Kelebihan dari sistem ini dibandingkan dengan sistem sebelumnya yaitu tidak hanya terbatas pada peminjaman ruangan saja namun juga terdapat peminjaman peralatan. Sistem ini juga dapat digunakan di dalam maupun di luar jam perkuliahan dan di tambahkan dengan fitur – fitur yang disesuaikan dengan kondisi saat ini. Harapan dari sistem ini nantinya dapat mempercepat dan mempermudah proses peminjaman ruangan dan peralatan bagi civitas akademika di Politeknik Negeri Cilacap sehingga seluruh kegiatan dapat difasilitasi dengan baik.

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 Sistem Informasi

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan melaksanakan suatu fungsi untuk mencapai tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem.

Informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya dan untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan.

Sistem informasi merupakan suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [9].

2.2.2 Basis Data

Basis data adalah kumpulan *file -file* yang mempunyai kaitan antara satu *file* dengan *file* lain sehingga membentuk satu bangun data untuk menginformasikan suatu perusahaan instansi, dalam bahasan tertentu.

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language (SQL)*. MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL Free SoftwareI* yang berada dibawah Lisensi *General Public License (GPL)*:

MySQL juga dapat didefinisikan sebagai sebuah *database server*, dapat juga berpersion sebagai *client* sehingga sering disebut *database*

client / server yang *open source* dengan kemampuan dapat berjalan baik di Operasi Sistem maupun dengan *Platform Windows* maupun *Linux*.

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1994-1995. MySQL sudah ada sejak 1979. MySQL termasuk jenis *Relation Database Management System* (RDBMS) digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Sebuah basis data yang terdapat pada MySQL mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah suatu bahasa (*language*) yang digunakan untuk mengakses data di dalam sebuah *database* relasional. SQL sering juga disebut dengan istilah *query*, dan bahasa SQL secara praktiknya digunakan sebagai bahasa standar untuk manajemen *database* relasional. Hingga saat ini hampir seluruh *server database* atau software *database* mengenal dan mengerti bahasa SQL. Dalam penggunaan SQL terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam *database*. Secara garis besar, SQL Server mempunyai 3 (Tiga) jenis perintah SQL yaitu:

1. **Data Definition Language (DDL)**

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah *database*, dalam hal ini *database* dan tabel. Terdapat tiga perintah penting dalam DDL, yaitu:

a. **CREATE**

Perintah ini digunakan untuk membuat, termasuk di dalamnya membuat *database* baru, tabel baru view baru, dan kolom baru. Contoh: `CREATE DATABASE nama_database`.

b. **ALTER**

Perintah **ALTER** berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan

memberikan atribut pada kolom. Contoh: *ALTER TABLE* nama_tabel *ADD* nama_kolom *datatype*.

c. *DROP*

Perintah *DROP* berfungsi untuk menghapus *database* atau tabel. Contoh: *DROP DATABASE* nama *database*.

2. *Data Manipulation Language (DML)*

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam *database* yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu:

a. *INSERT*

Perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika *database* dan tabel sudah dibuat. Contoh: *INSERT INTO* nama_tabel *VALUES* (data1, data2, dst...);

b. *SELECT*

Perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: *SELECT* nama_kolom1, nama_kolom2 *FROM* nama_tabel;

c. *UPDATE*

Perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: *UPDATE* nama_tabel *SET* kolom1=data1, kolom2=data2,... *WHERE* kolom=data;

d. *DELETE*

Perintah delete digunakan untuk menghapus data dari sebuah tabel. Contoh: *DELETE FROM* nama_tabel *WHERE* kolom=data;

3. *Data Control Language (DCL)*

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan *server databasenya*, seperti manipulasi *user* dan hak akses (*priviledges*). Yang termasuk perintah dalam DCL ada 2 (Dua), yaitu:

a. *GRANT*

Perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu *user* atau pengguna. Hak akses tersebut bisa berupa hak membuat (*CREATE*), mengambil data (*SELECT*),

menghapus data (*DELETE*), mengubah data (*UPDATE*), dan hak khusus lainnya yang berhubungan dengan sistem *database*.

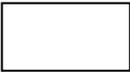
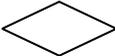
b. REVOKE

Perintah ini digunakan untuk mencabut hak akses yang telah diberikan kepada *user*. Dalam ini merupakan kebalikan dari perintah *GRANT*.

4. *Entity Relationship Diagram (ERD)*

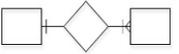
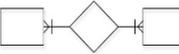
ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya. Berikut merupakan simbol-simbol dari ERD:

Tabel 2. 1 Simbol ERD

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	Link		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

Tabel 2. 2 Macam-Macam Kardinalitas

No	Nama	Simbol	Keterangan
1	Relasi Satu ke Satu (<i>One to One</i>)		Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
2	Relasi Satu ke Banyak (<i>One to Many</i>)		Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3	Relasi Banyak ke Banyak (<i>Banyak to Many</i>)		Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya

2.2.3 Rekayasa Perangkat Lunak (RPL)

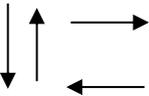
1. Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Ada beberapa jenis-jenis *flowchart* diantaranya:

- Bagan alir sistem (*system flowchart*)
- Bagan alir dokumen (*document flowchart*)
- Bagan alir skematik (*schematic flowchart*)
- Bagan alir program (*program flowchart*)
- Bagan alir proses (*process flowchart*)

Simbol-simbol dalam *flowchart* dapat dilihat pada Table 2.3

Tabel 2. 3 Simbol Flowchart

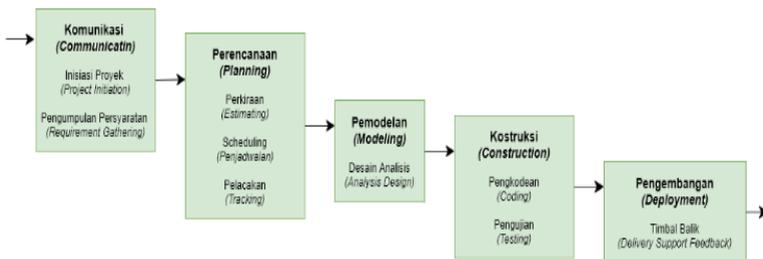
No	Simbol	Keterangan
1	 <p><i>Flow Direction Symbol</i></p>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connecting line</i> .
2	 <p><i>Terminator Symbol</i></p>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3	 <p><i>Processing Symbol</i></p>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 <p><i>Manual Operation Symbol</i></p>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
5	 <p><i>Decision Symbol</i></p>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5	 <p><i>Input -Output Symbol</i></p>	Yaitu simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dari jenis peralatannya.

6	 <p style="text-align: center;">Document Symbol</p>	Yaitu simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.

2. Metode Waterfall

Metode yang digunakan dalam pembuatan sistem ini adalah menggunakan pendekatan berorientasi data atau terstruktur yaitu *Linier Sequential Model*, model proses ini sering disebut juga sebagai Metode Waterfal.

Menurut Roger S. Pressman, *Waterfall Model* atau biasa disebut klasik *Life Cycle* adalah model klasik yang bersifat sistematis, berurutan dalam membangun piranti lunak. Aktivitas-aktivitas dalam *waterfall* model adalah sebagai berikut [3]:



Gambar 2. 1 Metode Waterfall

a. Komunikasi (*Communication*)

Pada tahap ini akan dilakukan inisiasi proyek, seperti menganalisis masalah yang ada dan tujuan yang akan dicapai.

b. Perencanaan (*Planning*)

Tahap ini merupakan tahap dimana akan dilakukan estimasi mengenai kebutuhan-kebutuhan yang diperlukan untuk membuat sebuah sistem. Selain itu, penjadwalan dalam proses pengerjaan juga

ditentukan pada tahap ini.

c. Pemodelan (*Modeling*)

Kemudian mulai masuk pada tahap pemodelan dimana perancang menerjemahkan kebutuhan sistem kedalam representasi untuk menilai kualitas sebelum tahap selanjutnya dikerjakan.

d. Konstruksi (*Construction*)

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemah desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh *user*. Tahap ini yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan pengujian (*Testing*) terhadap sistem yang telah dibuat tadi. Tujuan pengujian adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki [3].

Pengujian sistem dilakukan untuk mengetahui kesalahan yang muncul ketika program selesai dibuat untuk selanjutnya dapat diperbaiki. Pengujian ini secara fungsional dilakukan untuk meminimalkan kesalahan dan memastikan bahwa semua bagian lulus pengujian dan hasil *output* sesuai dengan keinginan pengguna. Pengujian yang dilakukan pada penelitian ini adalah pengujian *black-box* untuk memperlihatkan fungsi perangkat lunak dapat beroperasi dengan baik.

Pengujian *black-box* didasarkan pada detail dari sistem seperti antarmuka, fungsi dan kesesuaian alur dengan permintaan pengguna. Berbeda dengan pengujian *white-box* yang menguji kode program, pengujian *black-box* tidak dapat menguji bahkan melihat kode program. Selain itu, *black-box Testing* juga menguji *input* serta *output* yang dihasilkan dengan menggunakan semua persyaratan fungsional suatu program. Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan pada tahap akhir dikarena pengujian *black-box* memperhatikan struktur - struktur kontrol, sehingga perhatian berfokus pada domain Informasi. Pengujian *black-box* dapat menemukan kesalahan dengan menggunakan kategori sebagai berikut:

- 1) Fungsi yang tidak benar atau hilang
- 2) Kesalahan antarmuka

- 3) Kesalahan struktur data atau akses *database* eksternal
- 4) Kesalahan kinerja
- 5) Kesalahan inisialisasi ataupun terminal Pengujian *black-box* lebih baik dilakukan pada tahap akhir.

Peng uji harus merancang skenario pengujian yang dapat memberitahu mengenai ada atau tidaknya kesalahan melalui beberapa proses yang ada, diantaranya yaitu:

- 1) Menganalisa kebutuhan dan spesifikasi dari perangkat lunak
 - 2) Pemilihan jenis *input* yang menghasilkan *output* dengan benar serta *input* yang mungkin menghasilkan *output* yang salah
 - 3) Menentukan *output* untuk setiap *input* -an
 - 4) Pengujian dilakukan dengan *input*-an yang benar-benar telah diseleksi
 - 5) Melakukan pengujian
 - 6) Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan
 - 7) Menentukan fungsional yang seharusnya ada pada perangkat lunak
- e. Pengembangan (*Deployment*)

Tahap ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian software yang telah dibuat dilakukan pemeliharaan secara berkala.

2.2.4 Data Flow Diagram (DFD)

Data Flow Diagram merupakan gambaran perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisa maupun rancangan sistem. Diagram ini menggambarkan secara rinci tentang proses yang terdapat pada diagram jenjang. Penggunaan DFD ini membantu untuk memahami sistem secara logika, terstruktur dan jelas. Dalam DFD juga menggambarkan penyimpanan data dan proses mentranForm asikan data.

Fungsi dari adanya DFD adalah pemodelan yang mungkin untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Menurut Pressman DFD memperlihatkan gambaran tentang masukan – proses – keluaran dari sebuah sistem. Dalam DFD terdapat 3 level yang harus diketahui, yaitu:

1. Diagram Konteks

Diagram konteks adalah sebuah diagram yang menggambarkan seluruh proses yang terdapat dalam sistem. Diagram konteks ini merupakan level tertinggi dalam DFD sehingga sebelum DFD dibuat, dibutuhkan diagram konteks. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan. Diagram konteks hanya menggunakan 3 simbol, yaitu entitas eksternal, aliran data dan proses.

2. Diagram Level 1

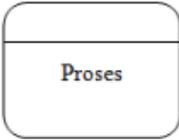
Diagram level 1 merupakan tahap lanjutan dari diagram 0, dimana pada diagram ini semua proses yang ada pada diagram konteks akan dirinci dengan lengkap sehingga lebih detail. Proses utama akan dipecah menjadi sub-proses sesuai dengan fungsinya masing-masing. Pada DFD level 1 ini akan digambarkan aliran data secara lebih kompleks di setiap prosesnya hingga terbentuklah datastore dan aliran data.

3. Diagram Level 2

Diagram level 2 merupakan diagram aliran data yang mana menjadi rincian proses dari DFD level 1 yang tugasnya adalah menguraikan proses-proses yang ada dalam lingkup sistem. Setiap proses yang ada pada DFD level 1 dapat diperinci menjadi sebuah DFD lagi. Proses DFD nantinya akan berhenti hingga cukup mendetail.

Simbol-simbol dalam DFD dapat dilihat pada tabel 2.4

Tabel 2. 4 Simbol Data Flow Diagram

No	Simbol	Fungsi
1	 <p><i>Entitas Eksternal</i></p>	<p>Simbol yang menunjukkan entitas luar yang berinteraksi langsung dengan sistem bisa dalam bentuk memberikan data ke sistem, menerima data dari sistem ataupun keduanya</p>
2	 <p><i>Proses</i></p>	<p>Simbol yang digunakan untuk menunjukkan kegiatan yang dilakukan oleh pengguna, komputer dari hasil suatu arus yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses atau untuk mengubah <i>input</i> menjadi <i>output</i>.</p>
3	 <p><i>Data Store</i></p>	<p>Simbol yang digunakan untuk menunjukkan simpanan data yang dapat berupa <i>file</i> atau <i>database</i> pada sebuah sistem</p>
4	 <p><i>Alur Data</i></p>	<p>Simbol yang menunjukkan data mengalir melalui sistem, dimulai dengan sebgaiian <i>input</i> dan diubah atau diproses menjadi <i>output</i>.</p>

~Halaman Ini Sengaja Dikosongkan~