



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya dilakukan oleh Imran Lubis dan Divi Handoko tahun 2019 dengan judul “Analisa dan Rancangan Sistem Informasi Manajemen Administrasi Pemberdayaan Kesejahteraan Keluarga Kota Sibolga”. Proses administrasi yang belum tersistem dengan baik menyebabkan kurang optimal dan efisien dalam penyediaan data dan informasi. Dalam penelitian ini, teknik pengembangan Sistem Informasi Manajemen PKK menerapkan metode Rapid Application Development (RAD), dimana metode ini memiliki keunggulan karena tahapan yang singkat dan cepat. Hasil dari perancangan SIM PKK dapat memberikan kemudahan dalam penyediaan data yang akurat, dan cepat bagi pengambilan kebijakan dan juga mampu meningkatkan kinerja dan efisiensi dalam organisasi PKK [1].

Penelitian lainnya dilakukan oleh Joko Handoyo tahun 2021 dengan judul “Sistem Informasi Program Kerja Pemberdayaan dan Kesejahteraan Keluarga (PKK) Di Desa Cabean, Kecamatan Cepu, Kabupaten Blora”. Perancangan sistem informasi dimodelkan ke dalam diagram dalam metode pengembangan UML (Unified Modeling Language) dan System Development Life Cycle (SDLC) dengan model Waterfall. Tujuan penelitian ini adalah membangun sebuah sistem informasi program kerja PKK yang diharapkan mampu menyelesaikan permasalahan yang berkaitan dengan kualitas informasi, berupa: tersedianya informasi dalam bentuk komputersisasi sehingga dapat diperoleh dengan cepat [2].

Pada tahun 2018, Doni Indrawan dan Jaja melakukan penelitian dengan judul “Sistem Informasi Manajemen Program Kerja Pemberdayaan dan Kesejahteraan Keluarga (PKK)”. Hasil dari rancang bangun ini adalah sebuah aplikasi berbasis web bernama SIM-Program Kerja yang dapat dijalankan pada semua sistem operasi. Dengan dikembangkannya Sistem Informasi Manajemen Program Kerja pada Pemberdayaan dan Kesejahteraan Keluarga (PKK) berbasis web, dapat mempermudah dalam melakukan pemantauan dari hasil kegiatan dan program kerja yang sedang berjalan serta dapat mengumpulkan data-

data untuk melakukan pelaporan dengan dibuatnya pengelolaan data program, realisasi dan Program Kerja [3].

Penelitian pada tahun 2018 yang dilakukan oleh Noor Latifah dengan judul “Perancangan Sistem Pengelolaan Kegiatan Pemberdayaan Kesejahteraan Keluarga Di Universitas Muria Kudus Berbasis Web”. Tujuan penelitian ini adalah untuk membantu dalam mengelola kegiatan termasuk di dalamnya kegiatan menabung, mendokumentasikan hasil kegiatan serta membantu dalam promosi hasil ketrimampilan yang pernah dilaksanakan yang secara tidak langsung bisa membantu meningkatkan kesejahteraan keluarga melalui peran wanita [4].

Pada tahun 2019 Chendy Okstania, Mega Novita, dan Khoiriya Latifah melakukan penelitian dengan judul “Sistem Informasi Media Transparansi Program Pemberdayaan Kesejahteraan Keluarga (PKK) dan Event Reminder Dengan Api Telegram Berbasis Web Pada Rw 3 Kelurahan Karang Tempel Semarang”. Tujuan dari sistem ini adalah untuk menghasilkan suatu informasi yang cepat, tepat dan akurat sebagai dasar dalam pengambilan keputusan. Penelitian ini menggunakan metode waterfall. Dengan adanya sistem ini diharapkan dapat mempermudah pengelolaan data PKK [5].

Perbedaan dalam penelitian yang akan dikembangkan dengan penelitian sebelumnya yaitu saya ingin membuat sistem informasi pengelolaan data PKK yang bertujuan untuk memberikan fasilitas pencatatan dan pelaporan data PKK yang mudah dioperasikan. Metode yang digunakan dalam penelitian ini adalah metode prototype. Sistem dirancang untuk menginput data umum PKK, data kegiatan PKK, serta data Pokja PKK dari masing-masing kelompok PKK RW di Kelurahan Mertasinga

2.2 Landasan Teori

2.2.1 Sistem Informasi

Definisi sistem informasi dalam bukunya Abdul Kadir yang berjudul Pengenalan Sistem Informasi, yaitu : “Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan”. (Kadir, 2003)

Penjelasan tersebut menerangkan bahwa sistem informasi dapat mempermudah perusahaan dalam mencapai sasaran yang telah ditargetkan dengan mengkoordinasikan manusia dan komputer sebagai

sumber daya untuk mengubah masukan menjadi pengeluaran yang diinginkan.

Sistem informasi dalam mendukung beberapa komponen yang fungsinya sangat vital di dalam sistem informasi. Komponen-komponen sistem informasi tersebut adalah Hardware, software, prosedur, pengguna dan database. Komponen-komponen tersebut sangat penting dalam suatu sistem informasi, apabila salah satu komponen tidak ada maka sistem informasi tidak akan berjalan. Penggunaan sistem informasi dalam suatu organisasi atau sektor pemerintahan dapat meningkatkan kinerja dalam pelayanan publik agar suatu pelayanan dapat berjalan efektif dan efisien.

Secara rinci komponen-komponen sistem informasi dapat dijelaskan sebagai berikut :

- a. Perangkat keras (Hardware), mencakup peranti-peranti fisik seperti monitor dan printer.
- b. Perangkat lunak (software) atau program: sekumpulan intruksi yang memungkinkan perangkat keras untuk dapat memproses data
- c. Prosedur: sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembamasyarakatn keluaran yang dikendaki.
- d. Pengguna: semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. Database: merupakan kumpulan dari data yang saling berhubungan dengan data lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulsinya, diantaranya; data, user dan sistem. (Kadir, 2003:70) [6].

2.2.2 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu communication, requirements capturing (analisa kebutuhan pengguna), spesifikasi (menentukan spesifikasi dari kebutuhan pengguna), desain, coding, testing sampai maintenance (pemeliharaan sistem) setelah digunakan.

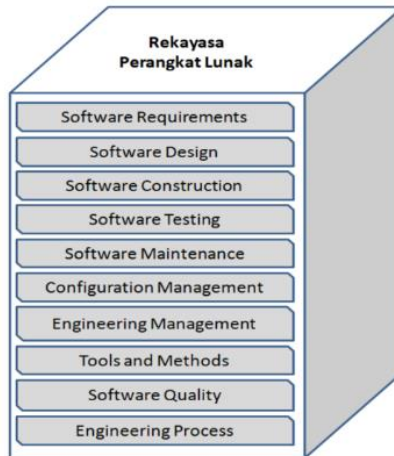
Tujuan rekayasa perangkat lunak adalah sebagai berikut :

1. Menghasilkan perangkat lunak yang berunjuk kerja tinggi, andal serta tepat waktu.

2. Menghasilkan perangkat lunak yang memiliki kekuatan dalam menghadapi berbagai ancaman dan gangguan, baik yang bersumber dari internal maupun eksternal.
3. Menghasilkan perangkat lunak dengan biaya produksi yang rendah.
4. Menghasilkan perangkat lunak yang biaya pemeliharaan yang rendah.
5. Menghasilkan perangkat lunak yang bisa bekerja di berbagai macam platform.

Rekayasa perangkat lunak merupakan serangkaian proses yang amat panjang untuk membuat atau menciptakan suatu perangkat lunak yang berkualitas, bukan merupakan cabang ilmu komputer yang mempelajari tentang technical coding.

Apabila dilihat dari persepsi wilayah cakupannya RPL memiliki wilayah operasional yang dapat dilihat pada Gambar 2.1 berikut ini.



Gambar 2. 1 Wilayah Cakupan RPL

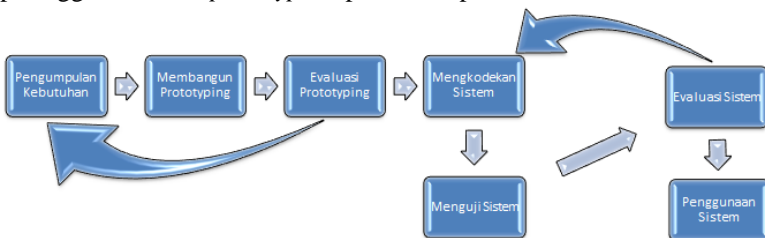
Keterangan :

1. *Software Requirements* meliputi serangkaian tugas, yang membantu menentukan dampak perangkat lunak pada organisasi, kebutuhan pelanggan, dan bagaimana pengguna akan berinteraksi dengan perangkat lunak yang dikembangkan.
2. *Software Design* meliputi proses penampilan arsitektur, komponen, antar muka (interface), dan karakteristik lain dari suatu perangkat lunak.

3. *Software Construction* meliputi aktifitas yang berhubungan dengan hal-hal detail dalam pengembangan perangkat lunak, termasuk algoritma, pengkodean, pencarian kesalahan (debug) dan pengujian (testing).
4. *Software Testing* meliputi pengujian pada kinerja perangkat lunak secara keseluruhan.
5. *Software Maintenance* mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
6. *Software Configuration Management (SCM)* adalah proses yang terpisah dari proses pengembangan karena sebagian besar model pengembangan tidak dapat mengakomodasi perubahan kapan saja selama pengembangan.
7. *Software Engineering Management* yang berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
8. *Software Engineering Tools and Methods* yang mencakup kajian teoritis tentang alat bantu (tools) dan metode RPL.
9. *Software Quality* yang menitikberatkan pada kualitas dan daur hidup perangkat lunak.
10. *Software Engineering Process* berhubungan dengan implementasi, definisi, pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.

A. Metode Pengembangan Sistem

Metode yang digunakan untuk pengembangan sistem ini adalah metode *prototype*. Menggunakan metode *prototype* karena dalam proses pengembangan sistem lebih cepat dan sesuai dengan keinginan pelanggan. Metode *prototype* dapat dilihat pada Gambar 2.2



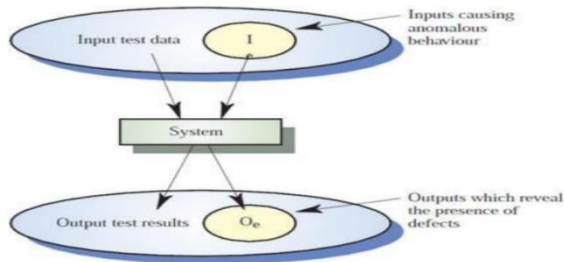
Gambar 2. 2 Metode Prototype

Fase fase dalam *Prototyping* adalah sebagai berikut [7]:

1. Pengumpulan kebutuhan
Di tahap ini pengembang melakukan identifikasi *software* dan semua kebutuhan sistem yang akan dibuat.
2. Membangun *prototyping*
Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format *output*).
3. Evaluasi *prototyping*
Evaluasi ini dilakukan untuk mengetahui apakah *prototyping* sudah sesuai dengan harapan pelanggan.
4. Mengkodekan sistem
Pada tahap ini *prototyping* yang sudah disetujui akan diubah ke dalam bahasa pemrograman.
5. Menguji sistem
Di tahap ini dilakukan untuk menguji sistem perangkat lunak yang sudah dibuat. Pengujian
6. Evaluasi Sistem
Perangkat lunak yang sudah siap jadi akan dievaluasi oleh pelanggan untuk mengetahui apakah sistem sesuai dengan yang diharapkan.
7. Menggunakan sistem
Perangkat lunak yang sudah diuji dan disetujui oleh pelanggan siap digunakan.

B. Metode Pengujian Sistem

Pengujian sistem yang digunakan dalam pengembangan sistem ini adalah *black-box testing*. *Black-box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi. Pengujian ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan *output* yang benar. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Metode ujicoba *black-box* memfokuskan pada keperluan fungsional dari *software*, karena memungkinkan pengembang *software* untuk membuat kondisi input yang akan melatih seluruh syarat fungsional suatu program. Ujicoba *black-box* bukan merupakan alternatif dari ujicoba *white-box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya [8].



Gambar 2. 3 Pengujian Black-Box

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya [8] :

- a. Fungsi yang salah atau hilang
- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja
- e. Kesalahan inisialisasi dan terminasi

Ciri-ciri pengujian *black-box* [8] :

1. Pengujian *black-box* berfokus pada kebutuhan fungsional pada *software*.
2. Pengujian *black-box* bukan teknik alternatif dari pengujian *white-box*, merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode pengujian *white-box*.
3. Pengujian *black-box* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang diuji.

Keunggulan pengujian *black-box* [8] :

1. Black-box dapat menguji keseluruhan fungsionalitas perangkat lunak.
2. Black-box dapat memilih subset test yang secara efektif dan efisien dapat menemukan cacat.

Kekurangan pengujian *black-box* [8] :

1. Penguji tidak yakin apakah perangkat lunak yang diuji telah benar.
2. Penguji tidak yakin bahwa perangkat lunak yang diuji telah lolos pengujian.

2.2.3 Program Berorientasi Objek (PBO)

Pemrograman berorientasi objek merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya [9].

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu [9] :

- a. *Class*
Class didefinisikan sebagai sebuah blueprint (denah), atau prototipe, yang mendefinisikan variabel-variabel dan metode-metode yang umum untuk semua objek dari n jenis tertentu (n maksudnya jumlah tertentu).
- b. *Object*
Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun *object* lain. *Object* adalah implementasi dari *class*. Secara sederhananya, dapat dikatakan terdiri dari *attribute* dan *method*.
- c. *Attribute*
Attribute adalah variabel global yang dimiliki sebuah kelas, yang dapat memiliki hak akses private, public maupun protected. Sebuah atribut yang dinyatakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.
- d. *Method*
Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu *Class*. Pada sebuah *method* di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas.

Adapun konsep dasar dalam PBO [9] :

- a. *Abstraction*
Abstraction merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti.

b. *Encapsulation*

Merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut.

c. *Inheritance*

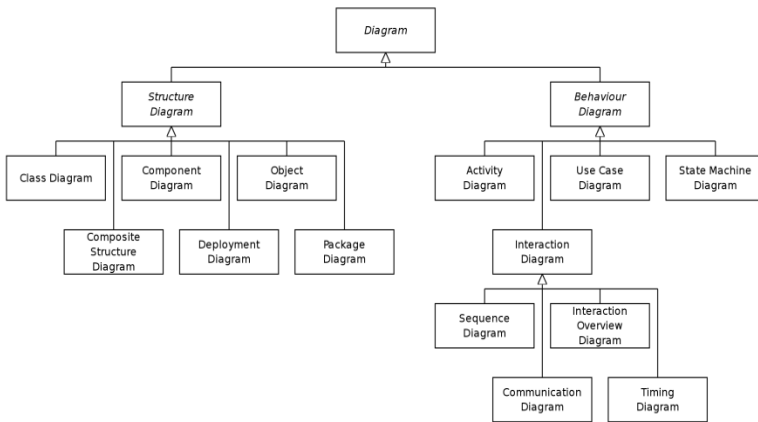
Merupakan konsep mewariskan *attribute* dan *method* yang dimiliki oleh sebuah *class* kepada *class* turunannya. Dengan konsep ini *class* yang dibuat cukup mendefinisikan *attribute* dan *method* yang spesifik didalamnya, sedangkan *attribute* dan *method* yang lebih umum akan didapatkan dari *class* yang menjadi induknya.

d. *Polymorphism*

Merupakan konsep yang memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu *object* agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda.

2.2.4 *Unified Modelling Language (UML)*

Unified Modelling Language merupakan alat perancangan sistem yang berorientasi pada objek, karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program [10].



Gambar 2. 4 Diagram UML


UML mempunyai tiga kategori utama yaitu [10] :



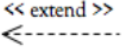

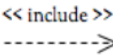
- *Structure diagram*, menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.
- *Behaviour diagram*, menggambarkan ciri-ciri metode/ fungsi dari sebuah sistem.
- *Interaction diagram*, bagian dari *behavior diagram* yang menggambarkan interaksi objek.

A. *Use Case Diagram*

Use case diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. *Use case diagram* memberikan deskripsi jelas dan konsisten dari apa yang harus dilakukan. Berikut adalah simbol-simbol yang ada pada *use case diagram* [11].

Tabel 2. 1 Simbol-simbol *Use Case Diagram*


No.	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Disediakan sistem sebagai unit yang saling bertukar pesan antar aktor. Biasanya dinyatakan dengan kata kerja.



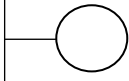

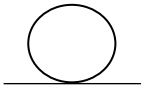
2.		<i>Actor</i>	Merupakan orang, sistem, ataupun proses yang berinteraksi dengan sistem yang akan dibuat diluar dari sistem itu sendiri. Biasanya dinyatakan dengan menggunakan kata benda.
3.		<i>Association</i>	Komunikasi antara aktor dengan <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor.
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> dapat berdiri sendiri tanpa <i>use case</i> tambahan.
5.		<i>Generalization</i>	Hubungan antara dua buah <i>use case</i> dimana fungsi yang satu lebih umum dari lainnya.
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya.

B. *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Berikut adalah simbol-simbol yang ada pada *sequence diagram* [11].

Tabel 2. 2 Simbol-simbol *Sequence Diagram*

No	Simbol	Fungsi
1		<i>Objek entity</i> , antarmuka yang saling berinteraksi.



2		Digunakan untuk menggambarkan <i>user/Pengguna</i> .
3	Message() 	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
4		Digunakan untuk menggambarkan sebuah <i>form</i> .
5		Digunakan untuk menghubungkan <i>Boundary</i> dengan tabel.
6		Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

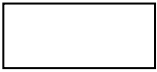
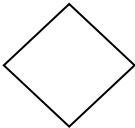






2.2.5 Flowchart



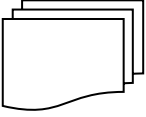

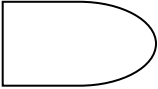
Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Dengan adanya flowchart, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu, ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan flowchart ini.

Ada beberapa fungsi flowchart yaitu merancang proyek baru, mengelola alur kerja, memodelkan proses bisnis, mendokumentasikan setiap proses, merepresentasikan algoritma, serta mengaudit proses. Berikut adalah simbol-simbol yang ada pada flowchart [12].

Tabel 2. 3 Simbol-simbol *Flowchart*

No	Gambar Simbol	Nama	Keterangan
1		<i>Terminal point</i>	Menunjukkan mulai atau berakhirnya suatu proses.
2		<i>Flow direction</i>	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi

			juga untuk menunjukkan alur sebuah proses.
3		Proses	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem.
4		<i>Decision</i>	Simbol yang digunakan untuk memilih sebuah keputusan atau proses sesuai dengan kondisi yang ada.
5		<i>Input output</i>	Menunjukkan sebuah <i>input</i> atau <i>output</i> yang ada tidak bergantung pada jenis peralatannya.
6		<i>Predefined process</i>	Menunjukkan pelaksanaan suatu bagian prosedur, bagian prosedur yang terinformasi belum detail dan akan diperinci ditempat lain.
7		<i>Connector (On-page)</i>	Menghubungkan suatu simbol pada satu halaman yang letaknya berjauhan.
8		<i>Connector (Off-page)</i>	Menghubungkan simbol yang berada dalam halaman yang berbeda.
9		<i>Preparation</i>	Simbol ini menunjukkan persiapan penyimpanan kedalam <i>storage</i> .
10		<i>Manual input</i>	Menunjukkan proses <i>input</i> yang dilakukan secara <i>manual</i> menggunakan <i>online keyboard</i> .

11		<i>Manual operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa menggunakan komputer.
12		<i>Document</i>	Menunjukkan <i>input</i> berupa dokumen dalam bentuk kertas atau <i>output</i> yang harus dicetak.
13		<i>Multiple document</i>	Sama seperti simbol <i>document</i> hanya saja dokumen yang digunakan lebih dari satu.
14		<i>Display</i>	Simbol yang menunjukkan adanya penggunaan peralatan <i>output</i> .
15		<i>Delay</i>	Menunjukkan bahwa adanya proses <i>delay</i> .

2.2.6 Basis Data

Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama tanpa adanya pengulangan (redundansi) data. Keutamaan basis data adalah pengaturan, pemilihan, pengelompokan, pengoperasian data yang akan disimpan sesuai fungsi dan jenisnya. Pada intinya basis data adalah media penyimpanan data agar dapat diakses dengan cepat dan mudah. Sistem informasi tidak dapat dipisahkan dengan basis data, baik berupa teks ataupun *Database Management System (DBMS)* [13].

Keuntungan menggunakan basis data antara lain :

- a. Pengendalian terhadap data terpusat.
- b. Mengurangi redundansi data.
- c. Terciptanya data yang konsisten.
- d. Integritas data dapat terpelihara.
- e. Independensi data.

Kerugian menggunakan basis data antara lain :

- a. Biaya yang dibutuhkan sangat mahal, karena membutuhkan *hardware*, *software*, dan *brainware* yang berkualitas.
- b. Bersifat kompleks dalam penggunaan *hardware* yang berkapasitas besar sehingga dibutuhkan tingkat keahlian yang tinggi [13].

A. ***Database Management System (DBMS)***

Database Management System (DBMS) merupakan perangkat lunak yang digunakan untuk memudahkan dan mengefisienkan pemasukan, pengeditan, penghapusan dan pengambilan informasi terhadap basis data. *Software* yang tergolong dalam DBMS antara lain : *Microsoft SQL, MySQL, Oracle, MS. Access, Padox* [13].

Konsep yang terjadi dalam DBMS adalah :

- a. User melakukan pengaksesan basis data untuk informasi yang diperlukan menggunakan bahasa query yang biasa disebut dengan SQL.
- b. DBMS menerima permintaan dari user dan menganalisis permintaan tersebut.
- c. DBMS memeriksa skema eksternal user, skema konseptual dan struktur penyimpanan.
- d. DBMS mengeksekusi operasi-operasi yang diperlukan untuk memenuhi permintaan user.

Suatu sistem dikatakan DBMS jika memenuhi syarat :

1. Menyediakan fasilitas untuk mengelola data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data.
4. Mampu menangani *backup* data.

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

Data definition Language (DDL) merupakan suatu metode query SQL yang digunakan untuk mendefinisikan data pada suatu database. DDL juga dapat digunakan untuk membuat dan memodifikasi suatu obyek database [13]. Query yang dimiliki DDL adalah :

1. *Create* : Digunakan untuk membuat database dan tabel.

Contoh :

```
create database db_universitas;
create table mata_kuliah (kdmk varchar(8), nama_mata_kuliah
varchar(64), sks int);
```


2. *Drop* : Digunakan untuk menghapus database dan tabel.

Contoh :

```
drop database db_universitas;
```

3. *Alter* : Digunakan untuk melakukan perubahan struktur tabel yang sudah dibuat.

Contoh :

```
alter table nilai add akhir int;
```

4. *Rename* : Digunakan untuk menamakan kembali suatu tabel.

Contoh :

```
rename table db_name.table1 to new_db_name,
db_name.table2 to new_db_name;
```

C. *Data Manipulation Language (DML)*

Data Manipulation Language (DML) merupakan query SQL yang dapat digunakan apabila DDL sudah terjadi, sehingga fungsi dari query DML ini untuk memanipulasi data yang ada didalam database [13]. Query yang dimiliki DML adalah :

1. *Select* : Digunakan untuk memasukan data pada tabel database.

Contoh :

```
select * from mahasiswa;
select * from mahasiswa where umur='20';
```

2. *Insert* : Digunakan untuk memasukan data pada tabel database.

Contoh :

```
insert into mahasiswa (npm, nama_lengkap, jenis_kelamin,
alamat) values ('190202044', 'Syela', 'P', 'Cilacap');
```

3. *Update* : Digunakan untuk mengubah data yang ada pada tabel.

Contoh :

```
update mahasiswa set nama_lengkap='Syela Febrianti',
where npm='190202044';
```

4. *Delete* : Digunakan untuk menghapus data pada tabel.


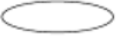


Contoh :

```
delete from mahasiswa where npm='190202044';
```

2.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan pemodelan basis data untuk menggambarkan relasi antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas. Berikut adalah simbol-simbol yang ada pada ERD [13].

Tabel 2. 4 Simbol-simbol ERD

No.	Simbol	Nama	Keterangan
1.		<i>Entity</i>	Kumpulan dari objek yang dapat diidentifikasi secara unik.
2.		Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
3.		Relasi	Hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain: <i>one to one</i> , <i>one to many</i> , dan <i>many to many</i> .
4.		<i>Association</i>	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

Halaman sengaja dikosongkan