

**LAPORAN KERJA PRAKTEK**  
**MEMBANGUN APLIKASI PERPUSTAKAAN**  
**MUTIARA ILMU DENGAN MICROSOFT ACCESS DI**  
**DESA CIPARAY**

Diajukan untuk memenuhi persyaratan kelulusan  
Matakuliah TIF335 Kerja Praktek

oleh :  
ASRI HARDIYANTI / C1A160023



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BALE BANDUNG**  
**2019**

# **LEMBAR PENGESAHAN**

## **PROGRAM STUDI TEKNIK INFORMATIKA**

### **MEMBANGUN SISTEM INFORMASI PERPUSTAKAAN MUTIARA ILMU DENGAN MICROSOFT ACCESS DI DESA CIPARAY**

oleh :

ASRI HARDIYANTI / C1A160023

disetujui dan disahkan sebagai

Laporan Kerja Praktek

Bandung, Juli 2019

Koordinator Kerja Praktek

Yaya Suharya ,S.Kom MT.

NIK.0407047706

# **LEMBAR PENGESAHAN**

## **PERPUSTAKAAN DESA CIPARAY**

### **MEMBANGUN SISTEM INFORMASI PERPUSTAKAAN MUTIARA ILMU DENGAN MICROSOFT ACCESS DI DESA CIPARAY**

oleh :

ASRI HARDIYANTI / C1A160023

disetujui dan disahkan sebagai

Laporan Kerja Praktek

Bandung, Juli 2019

Kepala Perpustakaan Mutiara Ilmu,

YADHIE WAHYU Q

## **Abstraksi**

Aplikasi Perpustakaan adalah sistem yang berfungsi untuk pengolahan data perpustakaan dari stok buku-buku yang tersedia, pengunjung sampai yang melakukan peminjaman dan pengembalian buku yang dipinjam. Sistem ini menggunakan database Microsoft Access sehingga untuk mengembangkan jaringan akan lebih mudah untuk diimplementasikan.

Dengan menggunakan sistem aplikasi ini kemudahan proses pengelolaan data perpustakaan dan pengontrolannya akan lebih mudah. Setiap data stok buku, pengunjung dan peminjam.

Penelitian ini telah menghasilkan sebuah sistem Aplikasi Perpustakaan yang membantu kerja dari para petugas perpustakaan dan dapat mempermudah pengguna untuk melakukan proses Input data dan Membuat Laporan agar pengelolaan input data dapat diolah secara efektif dan efisien, sehingga bisa langsung di akses serta informasi berupa Laporan dapat tersampaikan dengan baik.

*Kata kunci: Aplikasi, Input, Laporan, Perpustakaan.*

## **Kata Pengantar**

Puji syukur saya ucapkan kepada Allah SWT, dengan rahmat dan karunianya penulis dapat menyelesaikan Laporan Kerja Praktek dengan judul “Membangun Sistem Informasi Perpustakaan Mutiara Ilmu dengan Microsoft Access Di Desa Ciparay”. Dari hasil kerja praktek penulis di Perpustakaan Desa Ciparay yang beralamat di : Jl. Bojong-Parigi No.111 Desa Ciparay Kecamatan Ciparay.

Pada kesempatan ini, penulis mengucapkan terimakasih atas segala kerjasama dan dukungannya kepada :

1. Keempat orang tua serta Suami tercinta, yang tanpa lelah selalu memberikan dukungan, do'a dan semangat serta menjadi motivasi bagi penulis untuk tetap berjuang sampai hari ini.
2. Bapak Yudi Herdiana, S.T, M.T, sebagai Dekan Fakultas Teknologi Informasi.
3. Bapak Ahmad Faozan Muntaha, S.Kom selaku dosen pembimbing Kerja Praktek
4. Bapak Yadhie Wahyu Q, sebagai pembimbing lapangan tempat penulis melaksanakan kerja praktek danSeluruh rekan di Perpustakaan Mutiara Ilmu.
5. Dan untuk semua pihak yang tidak dapat penulis sebutkan satu-persatu yang telah turut membantu penulis dalam menyelesaikan Laporan Kerja Praktek ini.

Dengan penuh kesadaran diri dan segala kerendahan hati, penulis menyadari bahwa laporan kerja praktek ini masih sangat jauh dari sempurna.Oleh karena itu, penulis senantiasa mengharapkan kritik dan saran, semoga Laporan Kerja Praktek ini bermanfaat bagi pembaca.

Bandung, 2019

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PENGESAHAN .....	iii
Abstraksi .....	i
Kata Pengantar .....	ii
DAFTAR ISI .....	iii
DAFTAR GAMBAR .....	v
DAFTAR TABEL .....	vii
DAFTAR PUSTAKA .....	viii
Bab I PENDAHULUAN .....	I-1
I.1    Latar belakang .....	I-1
I.2    Lingkup .....	I-2
I.3    Batasan Masalah .....	I-2
I.4    Tujuan .....	I-2
Bab II ORGANISASI ATAU LINGKUNGAN KP .....	II-1
II.1    Struktur Organisasi .....	II-1
II.2    Visi dan Misi Perpustakaan .....	II-2
II.3    Tujuan Perpustakaan Mutiara Ilmu .....	II-2
II.4    Lingkup Pekerjaan .....	II-2
II.5    Deskripsi Pekerjaan .....	II-3
II.6    Jadwal Kerja .....	II-3
Bab III PENGETAHUAN TEORI PENUNJANG KP .....	III-1
III.1    Teori Penunjang KP .....	III-1
III.1.1    Sistem Informasi .....	III-1
III.1.2    Konsep Sistem .....	III-2
III.1.3    Konsep Informasi .....	III-2
III.1.4    Microsoft Access 2010 .....	III-3
III.1.5    Perancangan .....	III-7
III.1.6    Data Base .....	III-8
III.1.7    Analisis .....	III-9
III.1.8    SDLC ( <i>Systems Development Life Cycle</i> ) .....	III-10

III.1.9	Metode <i>Water Fall</i> .....	III-13
III.1.10	UML ( <i>Unified Modeling Language</i> ) .....	III-18
III.1.11	<i>Use Case Diagram</i> .....	III-23
III.1.12	<i>Activity Diagram</i> .....	III-29
III.1.13	Klasifikasi Buku .....	III-33
<b>Bab IV PELAKSANAAN KERJA PRAKTEK</b> .....		IV-1
IV.1	Input .....	IV-1
IV.2	Proses .....	IV-1
IV.2.1	Analisis Sistem yang sedang berjalan .....	IV-1
IV.2.2	Analisis Kebutuhan Sistem .....	IV-2
IV.3	Output .....	IV-3
IV.3.1	Perancangan Aplikasi .....	IV-3
IV.3.2	Perancangan Basis Data .....	IV-11
IV.3.3	Perancangan Tampilan User Face .....	IV-14
IV.3.4	Tampilan User Interface .....	IV-20
IV.3.5	Pencapaian Hasil .....	IV-27
<b>Bab V PENUTUP</b> .....		V-1
V.1	Kesimpulan dan saran mengenai pelaksanaan KP .....	V-1
V.1.1	Kesimpulan Pelaksanaan Kerja Praktek .....	V-1
V.1.2	Saran Pelaksanaan KP .....	V-2
V.2	Kesimpulan dan saran mengenai substansi yang digeluti selama KP ...	V-2
V.2.1	Kesimpulan .....	V-2
V.2.2	Saran .....	V-3
Lampiran A. TOR (TERM OF REFERENCE) .....		A-1
Lampiran B. Log Activity .....		B-1

## DAFTAR GAMBAR

<i>Gambar 2.1 Struktur Organisasi Perpustakaan.....</i>	<i>II-1</i>
<i>Gambar 3.1 Hirarki Data Base.....</i>	<i>III-9</i>
<i>Gambar 3.2 Tahapan Metode Waterfall.....</i>	<i>III-14</i>
<i>Gambar 3.3 Start Point/ Initial State.....</i>	<i>III-31</i>
<i>Gambar 3.4 Simbol Activity.....</i>	<i>III-31</i>
<i>Gambar 3.5 Simbol Action Flow.....</i>	<i>III-31</i>
<i>Gambar 3.6 Simbol Decision.....</i>	<i>III-32</i>
<i>Gambar 3.7 Simbol Synchronization.....</i>	<i>III-32</i>
<i>Gambar 3.8 Simbol Merge Event.....</i>	<i>III-33</i>
<i>Gambar 4.1 Use Case Diagram Admin.....</i>	<i>IV-4</i>
<i>Gambar 4.2 Activity Diagram Login.....</i>	<i>IV-6</i>
<i>Gambar 4.3 Activity Diagram Input data Buku.....</i>	<i>IV-6</i>
<i>Gambar 4.4 Activity Diagram Input data anggota.....</i>	<i>IV-7</i>
<i>Gambar 4.5 Activity Diagram Input data transaksi.....</i>	<i>IV-7</i>
<i>Gambar 4.6 Activity Diagram Input data kunjungan.....</i>	<i>IV-8</i>
<i>Gambar 4.7 Activity Diagram untuk melihat profil perpustakaan.....</i>	<i>IV-8</i>
<i>Gambar 4.8 Activity Diagram Untuk melihat Struktur Organisasi.....</i>	<i>IV-9</i>
<i>Gambar 4.9 Activity Diagram Untuk melihat data master perpustakaan.....</i>	<i>IV-9</i>
<i>Gambar 4.10 Activity Diagram Logout.....</i>	<i>IV-10</i>
<i>Gambar 4.11 Activity Diagram untuk menampilkan Laporan.....</i>	<i>IV-11</i>
<i>Gambar 4.12 Perancangan Tampilan Switch Board.....</i>	<i>IV-14</i>
<i>Gambar 4.13 Perancangan Tampilan Switch Board.....</i>	<i>IV-14</i>
<i>Gambar 4.14 Perancangan Tampilan Login.....</i>	<i>IV-15</i>
<i>Gambar 4.15 Perancangan Tampilan Menu Utama.....</i>	<i>IV-16</i>
<i>Gambar 4.16 Perancangan Tampilan Struktur Organisasi.....</i>	<i>IV-17</i>
<i>Gambar 4.17 Perancangan tampilan Data master.....</i>	<i>IV-18</i>
<i>Gambar 4.18 Perancangan tampilan Input Data.....</i>	<i>IV-19</i>
<i>Gambar 4.19 Perancangan Tampilan Laporan.....</i>	<i>IV-20</i>
<i>Gambar 4.20 Halaman Switch Board.....</i>	<i>IV-21</i>



<i>Gambar 4.21 Halaman Switch Board.....</i>	<i>IV-21</i>
<i>Gambar 4.22 Gambar implementasi Form login .....</i>	<i>IV-21</i>
<i>Gambar 4.23 implementasi Form Menu Utama .....</i>	<i>IV-22</i>
<i>Gambar 4.24 Gambar implementasi Form profil perpustakaan. ....</i>	<i>IV-22</i>
<i>Gambar 4.25 Implementasi Form Struktur Organisasi .....</i>	<i>IV-23</i>
<i>Gambar 4.26 Gambar implementasi Form Data Master .....</i>	<i>IV-23</i>
<i>Gambar 4.27 Gambar Tampilan Data Master.....</i>	<i>IV-24</i>
<i>Gambar 4.28 Gambar implementasi Form Input data Buku .....</i>	<i>IV-24</i>
<i>Gambar 4.29 Gambar implementasi Form Input data Anggota. ....</i>	<i>IV-25</i>
<i>Gambar 4.30 Gambar implementasi Form Input data Transaksi .....</i>	<i>IV-25</i>
<i>Gambar 4.31 Gambar implementasi Form Input data kunjungan.....</i>	<i>IV-26</i>
<i>Gambar 4.32 Gambar implementasi Form Laporan Harian.....</i>	<i>IV-26</i>

## DAFTAR TABEL

<i>Tabel 3.1 Simbol-Simbol Use Case Diagram.....</i>	<i>III-25</i>
<i>Tabel 3.2 Simbol-Simbol Activity DiagramActivity Diagram.....</i>	<i>III-30</i>
<i>Tabel 4.1 Spesifikasi Perangkat .....</i>	<i>IV-3</i>
<i>Tabel 4.2 Deskripsi Use Case Diagram .....</i>	<i>IV-4</i>
<i>Tabel 4.3 Keterangan Rancangan Tabel Anggota .....</i>	<i>IV-11</i>
<i>Tabel 4.4 Keterangan Rancangan Tabel Buku .....</i>	<i>IV-12</i>
<i>Tabel 4.5 Keterangan Rancangan Tabel kunjungan .....</i>	<i>IV-13</i>
<i>Tabel 4.6 Keterangan Rancangan Tabel Transaksi Peminjaman.....</i>	<i>IV-13</i>
<i>Tabel B.1 Lampiran Log Activity.....</i>	<i>B-1</i>

## DAFTAR PUSTAKA

- [1] Adipurnomo R. 2012. Membuat Aplikasi Database Perpustakaan Menggunakan Access (Online).  
(<http://rikiadipur.blogspot.com/2012/05/membuat-database-perpustakaan-sederhana.html>. Diakses tanggal 16 Mei 2019).
- [2] Andi. (2007). Seri Panduan Lengkap Microsoft Access 2007. Madiun: MADCOMS.
- [3] Chandra A, Lukman HP. Sistem Informasi Perpustakaan Berbasis Microsoft Access pada Sekolah Tinggi Kejuruan ALAMNANIYAH di Ngawi.
- [4] Hakim L. 2012. Aplikasi Database Perpustakaan dengan Access (online).  
(<http://luqmanhak.blogspot.com/2012/11/aplikasi-database-perpustakaan-dengan.html>. Diakses tanggal 17 April 2019)
- [5] Saputro T. 2012. Pengertian Microsoft Access (Online), (<http://trisaputro1996.blogspot.com/2012/11/pengertian-microsoft-access.html>. Diakses tanggal 09 April 2019).

# **Bab I**

## **Pendahuluan**

### **I.1 Latar belakang**

Perpustakaan Desa atau yang biasa disingkat Perpusdes adalah bukan hanya sebagai tempat membaca ataupun meminjam buku saja, namun merupakan juga bisa digunakan sebagai sarana untuk meningkatkan kesejahteraan masyarakatnya. Hanya ada beberapa desa yang sangat peduli dengan hal ini, tentunya kalau mau untuk membangun desanya masing banyak hal yang di dapat dari pembangunan perpustakaan daerah.

Dengan adanya Perpustakaan Desa ini, masyarakat tidak hanya mampu mengeluarkan program dan juga produk, melainkan mampu juga melakukan marketing onlinenya. Sehingga masyarakat dapat meningkatkan penjualan dengan jangkauan promosi yang lebih luas. Sebagian besar masyarakat di desa berprofesi sebagai petani atau pekebun, dan dalam hal ini Perpustakaan Desa mampu memberikan pencerahan bagi masyarakat untuk bisa lebih maju dan modern. Dengan adanya computer/laptop dan internet yang mendukung, tentunya ini akan berdampak positif bagi masyarakatnya. Sehingga bisa menghasilkan bibit yang lebih unggul serta mampu menambah penghasilan.

Pada sebuah perpustakaan pasti sangat dibutuhkan sistem yang berfungsi untuk pengolahan data perpustakaan itu sendiri, mulai dari stok buku yang tersedia, pengunjung, anggota perpustakaan, sampai transaksi peminjaman dan pengembalian buku. Dengan menggunakan sistem aplikasi ini kemudahan proses pengolahan dan pengontrolan data perpustakaan akan lebih mudah. Begitu pula dengan pembuatan laporan harian perpustakaan akan lebih mudah dan akurat.

## **I.2 Lingkup**

Lingkup materi kerja praktek yang dilaksanakan di Perpustakaan Mutiara Ilmu adalah pembuatan aplikasi Perpustakaan Microsoft Acces. Saya Menangani semua data yang menyangkut beberapa hal , diantaranya:

- Data Profil Perpustakaan
- Data Buku
- Data Anggota Perpustakaan
- Data Transaksi Peminjaman
- Data Kunjungan
- Laporan

Aplikasi ini merupakan aplikasi system informasi yang dibangun untuk memudahkan Admin atau petugas perpustakaan untuk meng-input data data perpustakaan secara keseluruhan.

## **I.3 Batasan Masalah**

Agar penulisan laporan ini tidak menyimpang dari tujuan semula serta menghindari permasalahan-permasalahan yang lebih kompleks, maka penulis membatasinya. Adapun batasan permasalahan yang diambil penulis adalah mengenai proses pengolahan buku, anggota, transaksi peminjaman, laporan, serta data petugas perpustakaan dengan menggunakan Microsoft Access 2010.

## **I.4 Tujuan**

Tujuan pelaksanaan kerja prakek di Perpustakaan Mutiara Ilmu Desa Ciparay ini adalah Membangun aplikasi sistem Informasi untuk Mempermudah dan membantu petugas perpustakaan dalam proses pendataan transaksi pemijaman buku, pendataan anggota perpustakaan, dan data buku buku yang tersedia. Sehingga petugas perpustakaan dapat dengan mudah memberikan dan atau membuat laporan untuk dilaporkan kepada kepala perpustakaan.

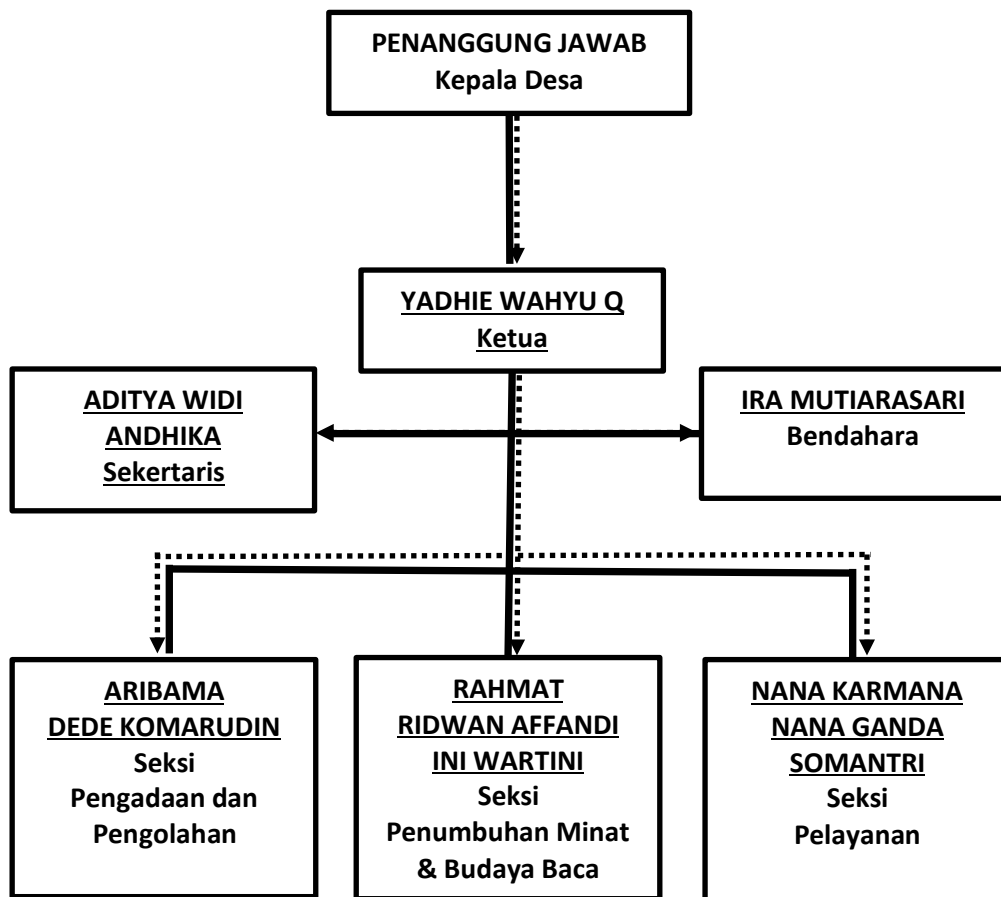
## Bab II

### Organisasi atau Lingkungan Kerja Praktek

#### II.1 Struktur Organisasi

Perpustakaan Mutiara Ilmu yang beralamat di Kp.Bojong-Parigi No.111 desa Ciparay kecamatan Ciparay adalah sebuah organisasi di bidang pendidikan.

Struktur organisasi Perpustakaan Mutiara Ilmu dapat dilihat pada gambar di bawah ini. Dalam melaksanakan kerja praktek, didapatkan bimbingan secara langsung dari Bapak Yadhie Wahyu Q selaku ketua Perpustakaan. Dengan demikian, pengembangan aplikasi ini berada di bawah lingkup Ketua Perpustakaan.



*Gambar 2.1 Struktur Organisasi perpustakaan*

## **II.2 Visi dan Misi Perpustakaan**

### **1. Visi Perpustakaan Mutiara Ilmu**

"Perpustakaan Desa Sebagai Sumber Informasi dan Tempat Belajar Masyarakat "

### **2. Misi Perpustakaan Mutiara Ilmu**

- Menumbuhkan Minat membaca Masyarakat Desa.
- Meningkatkan Kesadaran Budaya Literasi Masyarakat Desa.
- Meningkatkan Peran Serta Masyarakat Dalam Pengembangan dan Pemberdayaan Perpustakaan Desa.

## **II.3 Tujuan Perpustakaan Mutiara Ilmu**

- Menyediakan sumber informasi yang cepat, tepat, dan murah bagi masyarakat, terutama informasi mengenai topik yang berguna bagi mereka dan yang sedang hangat dalam kalangan masyarakat.
- untuk melayani masyarakat desa dalam memenuhi kebutuhan informasi masyarakat tersebut.
- Menyediakan wahana mencerdaskan kehidupan masyarakat desa dan menumbuhkan daya kreasi, prakarsa dan swakarsa masyarakat melalui peningkatan gemar membaca dan semangat belajar masyarakat.
- Menyediakan berbagai ilmu pengetahuan dan pengalaman kepada masyarakat dalam berbagai bidang.
- Menyediakan kebutuhan sarana edukasi, rekreasi, penerangan, informasi dan penelitian bagi warga desa.

## **II.4 Lingkup Pekerjaan**

Perpustakaan Mutiara Ilmu ini mempunyai lingkup pekerjaan sebagai penyedia dan pelayan masyarakat dalam hal penyediaan buku-buku edukatif untuk mempermudah masyarakat dalam mencari berbagai sumber ilmu, informasi dan pengetahuan untuk meningkatkan kualitas hidup masyarakat serta menambah minat baca masyarakat sekitar desa Ciparay.

Dalam pelaksanaan kerja praktek dilakukan pengolahan data data yang berkaitan dengan Perpustakaan seperti data buku yang tersedia, data data anggota Perpustakaan, data kunjungan, dan data-data Transaksi peminjaman dan pengembalian buku perpustakaan.

## **II.5 Deskripsi Pekerjaan**

Deskripsi pekerjaan yang dilakukan selama kerja praktek di Perpustakaan Mutiara Ilmu adalah Membangun Aplikasi system Informasi dan menangani hal berikut:

### **1. Komunikasi**

Melakukan wawancara dengan petugas Perpustakaan seputar Keperluan Perpustakaan, khususnya aplikasi yang memudahkan Pekerjaan Petugas Perpustakaan dalam mengelola data data perpustakaan.

### **2. Analisa**

Setelah melakukan komunikasi dan wawancara, didapatkan beberapa masalah dalam hal pengolahan data, yaitu pengolah data masih dilakukan secara manual di Microsoft Excel. Contoh nya dalam proses peng-*input-an* data yang di input satu persatu di Microsoft Excel sehingga memakan banyak waktu.

### **3. Perencanaan**

maka dari itu petugas perpustakaan dan kepala perpustakaan menyarankan untuk membuat aplikasi perpustakaan di Microsoft Access. Karena dirasa lebih membantu dan mudahkan petugas perpustakaan dalam hal pengolahan data perpustakaan dan pembuatan laporan.

### **4. Implementasi**

Dilakukan pengumpulan data data perpustakaan terlebih dahulu secara keseluruhan yang terdapat di Microsoft Excel sebelumnya. Yang kemudian di konversikan ke Microsoft Access, lalu membuat form-form menu dan form-form input,dan lain-lain. Sehingga aplikasi dapat digunakan oleh operator atau petugas perpustakaan dengan baik.

## **II.6 Jadwal Kerja**

Kerja praktek dilaksanakan dari tanggal 18 februari 2019 sampai dengan 21 maret 2019 selama kurang lebih 4 minggu. Waktu kerja praktek adalah dari hari Senin dan kamis, pukul 08.00 sampai dengan pukul 15.00 WIB.



Secara umum, kegiatan yang dilakukan selama kerja praktek adalah sebagai berikut:

1. Minggu pertama :
  - Pengenalan lingkungan kerja.
  - Wawancara dengan petugas perpustakaan mengenai apa saja yang dibutuhkan di perpustakaan.
2. Minggu kedua :
  - Pengumpulan data perpustakaan yang dibutuhkan dalam pengerjaan aplikasi
  - Mempersiapkan Tools yang akan dipakai untuk membuat Aplikasi yaitu Microsoft Access 2010.
  - Mempelajari beberapa fitur yang diperlukan dalam pembuatan aplikasi di Microsoft access 2010 yang belum diketahui sebelumnya.
3. Minggu ketiga :
  - Mengkonversi beberapa data yang telah diberikan dari pihak perpustakaan yang berbentuk data Microsoft Excel menjadi data Microsoft Access.
  - Melakukan penyesuaian ulang seluruh data data.
  - Membuat beberapa form input.
  - Mendesain tampilan Aplikasi.
4. Minggu keempat :
  - Membuat Form Login dengan *encode*-an di Visual Basic.
  - Melakukan pengujian, perbaikan dan optimalisasi.
  - Membuat Laporan dan evaluasi kerja praktek.

Secara keseluruhan, realisasi jadwal kerja sesuai dengan rencana yang telah disusun. Selama pelaksanaan kerja praktek diadakan beberapa kali review dengan petugas dan ketua perpustakaan. Bapak Ari Bama dan Bapak Yadhie Wahyu, untuk menyampaikan kemajuan yang telah diperoleh.

## **Bab III**

### **Pengetahuan Teori penunjang KP**

#### **III.1 Teori Penunjang KP**

##### **III.1.1 Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam 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 tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri, 2004).

Sistem Informasi Perpustakaan Berbasis Ms Access

Menurut Mc leod (2001), Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi. Menurut O'Brien (2005), sistem informasi adalah suatu kombinasi terartur apapun dari people (orang), hardware (perangkat keras), software (piranti lunak), computer networks and data communications (jaringan komunikasi), dan database (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi.

Suatu sistem informasi pada dasarnya terbentuk melalui suatu kelompok kegiatan operasi yang tetap, yaitu:

1. Mengumpulkan data
2. Mengelompokkan data
3. Menghitung
4. Menganalisa
5. Menyajikan laporan.

Sistem informasi secara umum merupakan sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

### **III.1.2 Konsep Sistem**

Sistem adalah hubungan antara unit yang satu dengan unit yang lainnya yang saling berhubungan satu dengan yang lainnya dan tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan sebelumnya. Menurut Jogianto (2005) sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi.

### **III.1.3 Konsep Informasi**

Informasi adalah data yang telah tersaring dan diolah melalui sistem pengolahan sehingga memiliki arti dan nilai guna bagi seseorang untuk digunakan dalam mengambil suatu keputusan. Informasi menurut Jogiyanto (2005) adalah “sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.” Data hasil olahan yang berupa informasi hendaknya berkualitas bagi penerima informasi. Jogiyanto (2005) menyatakan bahwa kualitas suatu informasi tergantung dari tiga hal, yaitu:

1. Informasi harus akurat Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak biasa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus dapat akurat karena proses pencapaian informasi dari sumber informasi ke penerima kemungkinan banyak mengalami gangguan yang dapat menyebabkan informasi tersebut rusak atau berubah.
2. Informasi harus tepat waktu Tepat waktu berarti informasi yang datang untuk penerima tidak boleh terlambat, karena informasi yang sudah usang tidak memiliki nilai lagi. Informasi dijadikan sebagai landasan dalam pengambilan keputusan, sehingga informasi yang dikirim tidak boleh terlambat diterima oleh penerima informasi.

3. Informasi harus relevan Relevan berarti informasi yang dikirim harus mempunyai makna untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda satu sama lain. Sistem

#### **III.1.4 Microsoft Access 2010**

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

Microsoft merilis Microsoft Access 1.0 pada bulan November 1992 dan dilanjutkan dengan merilis versi 2.0 pada tahun 1993. Microsoft menentukan spesifikasi minimum untuk menjalankan Microsoft Access 2.0 adalah sebuah komputer dengan sistem operasi Microsoft Windows 3.0, RAM berkapasitas 4 *megabyte* (6 *megabyte* lebih disarankan) dan ruangan kosong hard disk yang dibutuhkan 8 *megabyte* (14 *megabyte* lebih disarankan). Versi 2.0 dari Microsoft Access ini datang dengan tujuh buah disket *floppy* 3½ inci berukuran 1.44 *megabyte*.

Perangkat lunak tersebut bekerja dengan sangat baik pada sebuah basis data dengan banyak *record* tapi terdapat beberapa kasus di mana *data* mengalami kerusakan. Sebagai contoh, pada ukuran basis data melebihi 700 *megabyte* sering mengalami masalah seperti ini (pada saat itu, memang *hard disk* yang beredar masih berada di bawah 700 *megabyte*). Buku manual yang dibawanya memperingatkan bahwa beberapa kasus tersebut disebabkan oleh *driver* perangkat yang kuno atau konfigurasi yang tidak benar.

Nama kode (*codename*) yang digunakan oleh Access pertama kali adalah Cirrus yang dikembangkan sebelum Microsoft mengembangkan Microsoft Visual Basic, sementara mesin pembuat *form* antarmuka yang digunakannya dinamakan dengan Ruby. Bill Gates melihat purwarupa (*prototype*) tersebut dan memutuskan bahwa komponen bahasa pemrograman BASIC harus dikembangkan secara bersama-sama sebagai sebuah aplikasi terpisah tapi dapat diperluas. Proyek ini dinamakan dengan Thunder. Kedua proyek tersebut dikembangkan secara terpisah, dan mesin pembuat *form* yang digunakan oleh keduanya tidak saling cocok satu sama lainnya. Hal tersebut berakhir saat Microsoft merilis Visual Basic for Applications (VBA).

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan Microsoft Internet Information Services (IIS) dan menggunakan Microsoft Active Server Pages (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada Microsoft SQL Server yang memiliki kemampuan yang lebih tinggi.

Beberapa pengembang aplikasi profesional menggunakan Microsoft Access untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid*

*Application Development/RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para *salesman*.

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka Access (*form*, *report*, *query*, dan kode Visual Basic) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya Microsoft Jet Database Engine (yang secara *default* digunakan oleh Microsoft Access), Microsoft SQL Server, Oracle Database, dan beberapa produk lainnya yang mendukung ODBC.

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statement SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini MAL dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine.

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, Microsoft JET Database Engine tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari Microsoft Access 2000 yang menggunakan Microsoft Jet Database Engine versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperti

halnya sebuah *stored procedure*, meskipun prosedur tersebut dibatasi hanya untuk sebuah pernyataan tiap prosedurnya. Access juga mengizinkan *form* untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan form tersebut, dan merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*pass-through* dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya.

Dalam berkas Access Database Project (ADP) yang didukung oleh Microsoft Access 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (\*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau Microsoft SQL Server, ketimbang menggunakan Microsoft JET Database Engine. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampir semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints* dan *trigger*, *view*, *stored procedure*, dan *UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah *form*, *report*, *macro*, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut.

Access mengizinkan pengembangan yang relatif cepat karena semua tabel basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya. Untuk membuat Query, Access menggunakan Query Design Grid, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa pemrograman SQL. Di dalam Query Design Grid, para pengguna dapat memperlihatkan tabel basis data sumber dari *query*, dan memilih *field-field* mana yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam *grid*. *Join* juga dapat dibuat dengan cara mengklik dan menyeret *field-field* dalam tabel ke dalam

*field* dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan.

Bahasa pemrograman yang tersedia di dalam Access adalah Microsoft Visual Basic for Applications (VBA), seperti halnya dalam beberapa aplikasi Microsoft Office. Dua buah pustaka komponen Component Object Model (COM) untuk mengakses basis data pun disediakan, yakni Data Access Object (DAO), yang hanya terdapat di dalam Access 97, dan ActiveX Data Objects (ADO) yang tersedia dalam versi-versi Access terbaru.

### **III.1.5 Perancangan**

Perancangan adalah penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi (Syifaun Nafisa, 2003). Perancangan system dapat dirancang dalam bentuk bagan alir system (system flowchart), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukan urutan-urutan proses dari system.

Menurut Al Bahra bin Ladjamudin (2013 : 375) ada dua tahapan dari perancangan, yaitu:

#### **1. Perancangan Masukan**

Merupakan awal dimulai proses pengolahan data. Bahan mentah dari informasi merupakan data yang muncul/terjadi berbagai (seluruh) transaksi yang dilakukan oleh seorang atau sekelompok orang. Data-data transaksi akan menjadi masukan bagi system informasi. Hasil dari system informasi yang diperoleh tidak akan menyimpang dari data yang dimasukan. Kualitas masukan menentukan kualitas keluaran.

#### **2. Perancangan Keluaran**

Pada tahap perancangan keluaran secara umum hanya dimaksudkan untuk menentukan bentuk output yang akan



dihasilkan oleh system yang akan dirancang, lengkap dengan struktur data dan tampilan layarnya.

### III.1.6 Data Base

Database merupakan salah satu komponen yang sangat penting dalam sistem operasi, karena merupakan basis dalam penyediaan informasi bagi para pemakai. Penerapan database dalam sistem informasi disebut database sistem. Database sistem adalah suatu sistem informasi yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan yang lainnya.

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis untuk memperoleh informasi dari basisdata tersebut. Berikut pengertian database menurut para ahli:

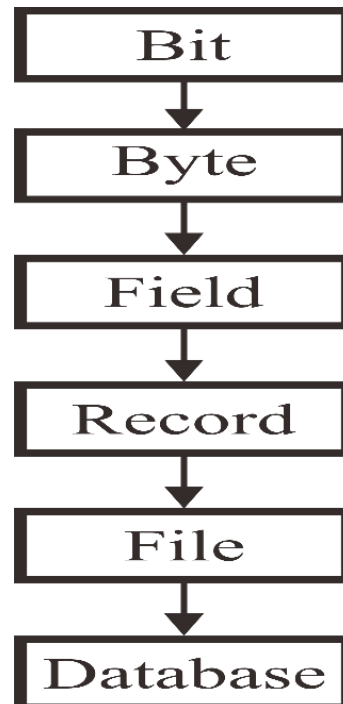
- Database adalah kumpulan data-data dalam bentuk tabel yang saling berhubungan yang disimpan dalam media perangkat keras, contohnya *Harddisk* yang dapat diambil lagi sebagai informasi. (Abdul Kadir, 2002).
- Database adalah suatu perorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. (Abdul Kadir, 2002)

Adapun hirarki data pada database adalah sebagai berikut:

1. **Bit**, merupakan bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte*.
2. **Byte**, merupakan atribut dari *field* yang berupa karakter yang membentuk nilai dari sebuah field.
3. **Field**, merupakan atribut dari *record* yang menunjukkan satu *item* dari data seperti: nama, nip, alamat dan lain sebagainya.
4. **Record**, merupakan kumpulan dari field yang menggambarkan satu unit dari individu tertentu.
5. **File**, merupakan kumpulan dari *record* yang saling terkait dan memiliki format field yang sama dan sejenis.

6. **Database**, kumpulan dari beberapa *file*/tabel yang saling berhubungan antara *file* yang satu dengan *file* yang lainnya.

Adapun gambar hirarki data pada database sebagai berikut:



*Gambar 3.1 Hirarki Data Base*

### III.1.7 Analisis

Pengertian analisis diartikan sebagai penguraian suatu pokok atas berbagai penelahan bagian itu sendiri, serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan (Prastowo dan Julianty, 2002). Menurut Komarudin, 2001, analisis adalah kegiatan berpikir untuk menguraikan suatu keseluruhan untuk menjadi komponen sehingga dapat mengenal tanda-tanda komponen, hubungannya satu sama lain dan fungsi masing-masing dalam satu keseluruhan yang terpadu.

Dari pendapat diatas dapat disimpulkan bahwa analisis atau analisa adalah kegiatan untuk menguraikan suatu pokok hal menjadi bagian-bagian atau komponen sehingga dapat diketahui cirri atau tanda tiap bagian, kemudian hubungan satu sama lain serta fungsi masing-masing dari keseluruhan.

### III.1.8 SDLC (*Systems Development Life Cycle*)

SDLC (*Systems Development Life Cycle*, Siklus Hidup Pengembangan Sistem) atau *Systems Life Cycle* (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*). Dalam rekayasa perangkat lunak angkyat Ä, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni: siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan prototyping (*life cycle using prototyping*), dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*).

SDLC adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi. Langkah yang digunakan meliputi:

- 1) Melakukan survei dan menilai kelayakan proyek pengembangan sistem informasi.
- 2) Mempelajari dan menganalisis sistem informasi yang sedang berjalan
- 3) Menentukan permintaan pemakai sistem informasi
- 4) Memilih solusi atau pemecahan masalah yang paling baik
- 5) Menentukan perangkat keras (*hardware*) dan perangkat lunak (*software*)
- 6) Merancang sistem informasi baru
- 7) Membangun sistem informasi baru

- 8) Mengkomunikasikan dan mengimplementasikan sistem informasi baru
- 9) Memelihara dan melakukan perbaikan/peningkatan sistem informasi baru bila diperlukan

*System Development Lyfe Cycle* (SDLC) adalah keseluruhan proses dalam membangun sistem melalui beberapa langkah. Ada beberapa model SDLC. Model yang cukup populer dan banyak digunakan adalah *waterfall*. Beberapa model lain SDLC misalnya *fountain*, *spiral*, *rapid*, *prototyping*, *incremental*, *build & fix*, dan *synchronize & stabilize*.

Dengan siklus SDLC, proses membangun sistem dibagi menjadi beberapa langkah dan pada sistem yang besar, masing-masing langkah dikerjakan oleh tim yang berbeda.

Dalam sebuah siklus SDLC, terdapat lima langkah. Jumlah langkah SDLC pada referensi lain mungkin berbeda, namun secara umum adalah sama. Berikut ini adalah Fase-fase *System Development Life Cycle* (SDLC) meliputi:

#### 1. Perencanaan Sistem (*Systems Planning*)

Lebih menekankan pada aspek studi kelayakan pengembangan system (*feasibility study*).Aktivitas-aktivitas yang ada meliputi:

- Pembentukan dan konsolidasi tim pengembang.
- Mendefinisikan tujuan dan ruang lingkup pengembangan.
- Mengidentifikasi apakah masalah-masalah yang ada bisa diselesaikan melalui pengembangan sistem.
- Menentukan dan evaluasi strategi yang akan digunakan dalam pengembangan sistem.
- Penentuan prioritas teknologi dan pemilihan aplikasi.

## 2. Analisis Sistem (*Systems Analysis*)

Analisa sistem adalah tahap di mana dilakukan beberapa aktivitas berikut:

- Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani oleh sistem.
- Brainstorming dalam tim pengembang mengenai kasus mana yang paling tepat dimodelkan dengan sistem.
- Mengklasifikasikan masalah, peluang, dan solusi yang mungkin diterapkan untuk kasus tersebut.
- Analisa kebutuhan pada sistem dan membuat batasan sistem.
- Mendefinisikan kebutuhan sistem.

## 3. Perancangan Sistem (*Systems Design*)

Pada tahap ini, features dan operasi-operasi pada sistem dideskripsikan secara detail. Aktivitas-aktivitas yang dilakukan adalah:

- Menganalisa interaksi obyek dan fungsi pada sistem.
- Menganalisa data dan membuat skema database.
- Merancang user interface.

## 4. Implementasi Sistem (*Systems Implementation*)

Tahap berikutnya adalah implementasi yaitu mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba.

Dalam implementasi, dilakukan aktivitas-aktivitas sebagai berikut:

- Pembuatan database sesuai skema rancangan.
- Pembuatan aplikasi berdasarkan desain sistem.
- Pengujian dan perbaikan aplikasi (*debugging*).

### 5. Pemeliharaan Sistem (*Systems Maintenance*)

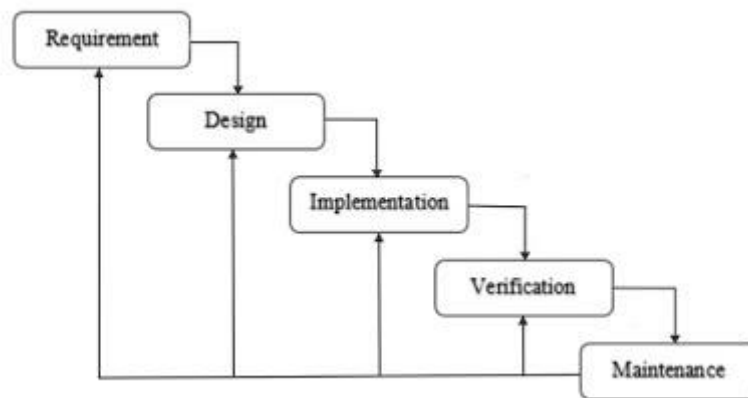
Dilakukan oleh admin yang ditunjuk untuk menjaga sistem tetap mampu beroperasi secara benar melalui kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan.

Siklus SDLC dijalankan secara berurutan, mulai dari langkah pertama hingga langkah keenam. Setiap langkah yang telah selesai harus dikaji ulang, kadang-kadang bersama expert user, terutama dalam langkah spesifikasi kebutuhan dan perancangan sistem untuk memastikan bahwa langkah telah dikerjakan dengan benar dan sesuai harapan. Jika tidak maka langkah tersebut perlu diulangi lagi atau kembali ke langkah sebelumnya.

Kaji ulang yang dimaksud adalah pengujian yang sifatnya quality control, sedangkan pengujian di langkah kelima bersifat quality assurance. Quality control dilakukan oleh personal internal tim untuk membangun kualitas, sedangkan quality assurance dilakukan oleh orang di luar tim untuk menguji kualitas sistem. Semua langkah dalam siklus harus terdokumentasi. Dokumentasi yang baik akan mempermudah pemeliharaan dan peningkatan fungsi sistem.

#### **III.1.9 Metode *Water Fall***

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012). Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini.



**Gambar 3.2 Tahapan Metode Waterfall**

- **Tahapan Metode Waterfall**

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, pemeliharaan. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut :

### **1. Requirement Analysis**

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

### **2. System Design**

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

### **3.Implementation**

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

### **4.Integration & Testing**

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

### **5.Operation & Maintenance**

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

- **Kelebihan Metode *Waterfall***

Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

- **Kekurangan Metode *Waterfall***

Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk



kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya.

- Prakteknya di Dunia Nyata

Secara teori penggunaan metode waterfall ini akan meminimalisir kesalahan pengerjaan sistem di awal karena antara klien dan pihak *developer* sudah saling berkomunikasi diawal pada saat menyusun inisiasi proyek. Tetapi apakah demikian?

Pada kenyataannya tidak semua klien dapat menjelaskan apa maksud dan tujuan dari sistem yang ingin dibuat. Karena hal itulah pihak *developer* bisa salah menterjemahkan ide dari klien terhadap sistem yang ingin dibuatnya. Dari pihak *developer* pun juga terkadang kurang bisa menggali kebutuhan sistem yang akan dibangunnya.

Waktu yang dibutuhkan untuk mengumpulkan data yang lengkap sebelum mulai suatu proyek cukup panjang, hal inilah yang sering menjadi kendala karena biasanya baik klien ataupun tim *developer* sama-sama tidak sabar. Dari sisi klien tidak sabar karena pihak *developer* terkesan banyak tanya dan tidak paham dengan sistem yang akan dibuatnya, sedangkan dari pihak *developer* tidak sabar karena biaya operasional yang dibutuhkan selama masa inisiasi proyek akan membengkak.

Waktu pengerjaan sangat sering menjadi kendala dalam proses pembuatan sebuah sistem. Sering kali klien menginginkan waktu penyelesaian sebuah proyek dalam tempo yang relatif singkat. Alih-alih waktu yang singkat, proses inisiasi proyek dan desain sistem tidak dapat dilakukan secara maksimal. Akibatnya proses implementasi sistem kejar tayang dan akhirnya dapat ditebak, sistem yang dibuat kurang sesuai dengan keinginan klien.

*Improvement* di fase implementasi sistem sering kali terjadi karena biasanya hal-hal terkait *improvement* ini tidak terlihat di saat inisiasi proyek. Hal ini bisa terjadi karena 2 hal, yaitu:

1. Kebutuhan klien yang terus bertambah terhadap sistem yang ingin dibangunnya.
2. Dari sisi *developer* sendiri yang terkadang menemukan kendala ataupun merasa ada yang kurang dari sistem yang sedang dibangunnya yang kemudian melakukan *improvement*.

*Improvement* yang dilakukan terus menerus selama fase implementasi ini sering menyebabkan waktu penyelesaian proyek menjadi mundur. Waktu penyelesaian mundur maka biaya operasional *developer* bertambah. Jika hal ini terjadi sebenarnya secara teori solusinya sederhana, semua *improvement* masuk ke tahap berikutnya, namun kenyataannya terkadang tidak demikian.

Tahap testing dilakukan di fase-fase akhir merupakan salah satu kendala yang sering dihadapi. Tim kontrol kualitas (*QA*) dan klien baru bisa melakukan testing saat semua sistem sudah selesai dikerjakan. Pada saat testing inilah terkadang klien baru menyadari bahwa proses yang diharapkan dari sistem yang dibangun tidak sesuai dengan ide awalnya.

- Bagaimana Solusinya?

Melihat prakteknya di lapangan, berarti metode waterfall tidak layak digunakan sebagai model pendekatan pengembangan software. Menurut saya kuncinya ada di inisiasi proyek dan tahap testing. Lalu bagaimanakah penerapannya?

1. Lakukan proses inisiasi proyek sedetil mungkin. Lakukan beberapa kali pertemuan dengan klien dan juga internal tim.

2. Libatkan tim QA pada tahap inisiasi proyek ini, sehingga mereka bisa mengetahui atau mungkin menyarankan sesuatu terkait pengembangan proyek tersebut.
3. Dokumentasikan hasil inisiasi proyek ini sedetail-detailnya dan minta klien untuk menyepakati hasil tersebut.

Setelah semua langkah itu terpenuhi barulah proyek bisa dimulai. Selain itu pihak *developer* dan klien harus mematuhi prosedur yang telah disepakati bersama, sehingga apabila ada *improvement* yang ingin dikerjakan supaya dimasukkan ke fase berikutnya.

### III.1.10 UML (*Unified Modeling Language*)

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

*Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang

menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG).

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

- Use Case Diagram untuk memodelkan proses bisnis.
- Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
- Sequence Diagram untuk memodelkan pengiriman pesan (*message*) antar *objects*.
- Collaboration Diagram untuk memodelkan interaksi antar *objects*.
- State Diagram untuk memodelkan perilaku *objects* di dalam sistem.
- Activity Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
- Class Diagram untuk memodelkan struktur kelas.
- Object Diagram untuk memodelkan struktur *object*.

- Component Diagram untuk memodelkan komponen *object*.
- Deployment Diagram untuk memodelkan distribusi aplikasi.

### **Tujuan atau fungsi dari penggunaan UML**

Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang diantaranya:

- Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
- Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
- Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
- Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai coding suatu program.
- Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
- Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

Menurut Adi Nugroho (2005), “bangunan dasar metodologi *Unified Modeling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan system/perangkat lunak yang akan dikembangkan yaitu:

- ***Sesuatu (things)***  
Ada 4 (empat) things dalam *unified modelling language* (UML) yaitu:
- ***Strucrutal Things***  
Merupakan bagian dari relative statis dalam model *unified modeling language* (UML). Bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

- ***Behavioral Things***

Merupakan bagian yang dinamis pada model *unified modeling language* (UML). Biasanya merupakan kata kerja dari model *unified modeling language*, yang mencerminkan perilaku sepanjang ruang atau waktu.

- ***Grouping Things***

Merupakan bagian pengoperasian dalam *unified modelling language* (UML). Dalam penggambaran model yang rumit kadang diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model dan subsistem-subsistem.

- ***Annotational Things***

Merupakan bagian yang menjelaskan model *unified modelling language* (UML) dan dapat berupa komentar-komentar yang menjelaskan fungsi serta cirri-ciri elemen dalam model *unified modelling language* (UML).

- ***Relasi (relationship)***

Ada 4 (empat) macam relationship dalam *unified modelling language* (UML), yaitu:

1. **Kebergantungan**

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elem yang bergantung padanya elemen yang tidak mandiri.

2. **Asosiasi**

Merupakan apa yang menghubungkan anatar objek satu dengan objek lainnya, bagaimana hubungan suatu objek dengan objek lainnya. Suatu bentuk asosiasi adalah agredasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

3. **Generalisasi**

Merupakan hubungan dimana objek anak (*descendent*) berbagai perilaku dan struktur data dari objek yang ada diatasnya objek

induk (*ancestor*). Arah dari atas kebawah dari objek induk ke objek anak dinamakan spesialisasi, sedangkan arah berlawanan sebaliknya dari arah bawah ke atas dinamakan generalisasi.

#### 4. Realisasi

Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

#### Langkah-langkah penggunaan *Unified Modeling Language* (UML)

1. Buatlah daftar *business process* dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk setiap *business process* dan definisikan dengan tepat fungsional yang harus disediakan oleh sistem, kemudian perhalus *use case diagram* dan lengkapi dengan *requirement*, *constraints* dan catatan-catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain non fungsional, *security* dan sebagainya yang juga harus disediakan oleh sistem.
5. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.
6. Definisikan obyek-obyek level atas *package* atau domain dan buatlah *sequence* dan/atau *collaboration* untuk tiap alur pekerjaan, jika sebuah *use case* memiliki kemungkinan alur normal dan *error*, buat lagi satu diagram untuk masing-masing alur.
7. Buatlah rancangan *user interface model* yang menyediakan antar muka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau domain dipecah menjadi hirarki class lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap class dibuat unit test untuk menguji fungsionalitas class dan interaksi dengan class lain.
9. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokkan class menjadi komponen-komponen karena itu

buatlah component diagram pada tahap ini. Juga, definisikan test integrasi untuk setiap komponen meyakinkan ia bereaksi dengan baik.

10. Perhalus *deployment diagram* yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan dan sebagainya. Petakan komponen ke dalam node.
11. Mulailah membangun sistem. Ada dua pendekatan yang tepat digunakan : Pendekatan *use case* dengan mengassign setiap *use case* kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan test dan pendekatan komponen yaitu mengassign setiap komponen kepada tim pengembang tertentu.

### ***III.1.11 Use Case Diagram***

*Use Case* adalah sebuah kegiatan atau juga interaksi yang saling berkesinambungan antara aktor dan juga sistem. Atau dengan kata lain teknik secara umum digunakan, guna mengembangkan software / sistem informasi, guna memperoleh kebutuhan fungsional dari sistem yang ada.

Komponen tersebut kemudian menjelaskan komunikasi antara actor, dengan sistem yang ada. Dengan demikian, use case dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen.

*Use case* ini adalah layanan atau juga fungsi yang ada pada sistem untuk para penggunanya. Sedangkan Use case Diagram, adalah gambaran efek fungsionalitas yang diharapkan oleh sistem.

Komponen ini pastinya sangat membantu ketika Anda sedanh menyusun requitment pada sebuah sistem, kemudian mengkomunikasikan rancangan aplikasi tersebut pada konsumen, dan juga merancang test case untuk berbagai fiture yang ada di sistem. Use case ini dapat digunakan atau diaplikasikan ke use case lainnya, sehingga duplikasi fungsionalitas dapat



dihindari, caranya adalah dengan menarik keluar fungsional yang common.

(*Martin Fowler, 2005*), *Use case* adalah table grafik yang berisi rangkaian use case yang digunakan untuk menjelaskan sebuah pemodelan. Use case digunakan untuk membentuk tingkah laku benda/things dalam sebuah model serta di Realisasikan oleh sebuah collaboration. Umumnya use case digambarkan dengan sebuah elips dengan garis yang solid, biasanya mengandung nama. Use case menggambarkan proses system (kebutuhan sistem dari sudut pandang user).

Secara umum use case adalah:

1. Pola prilaku sistem
2. Urutan transaksi yang berhubungan yang dilakukan oleh suatu actor.

Use case sendiri juga mempunyai fungsi serta manfaat yang baik untuk para pembuat dan juga para konsumen.

Fungsi use case diagram ini cukup banyak, antara lain :


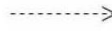

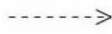






- Dapat memperlihatkan urutan aktifitas proses yang ada pada sistem
- Dapat menggambarkan prose bisnis dan juga urutan aktivitas yang ada dalam sebuah proses

Sedangkan manfaat dari Use case sendiri adalah ;

- Untuk memudahkan komunikasi dengan menggunakan domain expert dan juga end user.
- Adanya Interface yang harus dimiliki oleh sebuah sistem.
- Memberikan kepastian pemahaman yang pas, tentang requirement atau juga kebutuhan sebuah sistem.
- Dapat digunakan untuk mengidentifikasi, siapa yang sedang berinteraksi dengan sistem, dan juga apa yang harus dilakukan untuk sistem tersebut.

- Biasanya digunakan untuk verifikasi.

**Tabel 3.1 Simbol-Symbol Use Case Diagram.**

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Komponen use case sendiri hanya ada 3 macam, yaitu Sistem, Aktor dan juga use case sendiri. Berikut ini adalah penjelasan mengenai Use case diagram :

### 1. Sistem

Sebagai batasan sistem yang ada pada relasi dengan actor actor yang biasa menggunakannya dari luar sistem, dan fitur-fitur yang harus disediakan dalam sistem. Perangkat ini akan digambarkan dengan pola segi empat yang akan membatasi semua use case yang ada dalam sistem terhadap pihak yang mana sistem tersebut akan berinteraksi.

Sistem tersebut akan diberi label, sesuai dengan nama label tersebut. Tapi umumnya sistem ini tidak diberi gambar karena tidak terlalu memberikan arti bagi diagram tersebut.

## 2. *Actor*

*Actor* disini sebenarnya bukan bagian dari diagram, akan tetapi actor ini adalah salah satu alat untuk menciptakan use case diagram jadi lebih mudah. Actor ini akan digunakan untuk menjelaskan sesuatu atau juga seseorang yang sedang berinteraksi dengan sistem.

Dimana *actor* ini akan memberikan informasi pada sistem, hanya akan menerima informasi dari sistem, atau juga keduanya (yaitu memberikan informasi dari sistem, atau menerima informasi dari sistem). Disini Actor juga tidak akan memiliki kontrol terhadap use case itu sendiri, namun hanya diberikan gambaran umum atau juga spesifik, guna memudahkan Anda dalam menggunakan hubungan atau relationship yang ada.

Ada beberapa hal yang memungkinkan actor berhubungan dengan sistem lainnya, antara lain :

- Ketika seseorang ataupun pihak lain akan mengelola sistem tersebut.
- Adanya kepentingan terhadap sistem, dimana arus informasi baik untuk si penerima maupun inputan sistem saling berkepentingan.
- Adanya external resource yang digunakan oleh sistem tersebut.
- Adanya sistem lain yang tengah berinteraksi dengan sistem yang dibuat.

## 3. *Use case*

Use Case sendiri adalah gambaran fungsional dari sebuah sistem. Dengan demikian, antara konsumen dan juga pengguna pada sistem tersebut, akan mengerti atau paham mengenai fungsi sistem yang tengah dibangun.

Ada beberapa relasi dalam use case yang harus Anda ketahui, antara lain :

### ***1. Association***

Ini adalah teknik mengidentifikasi interaksi yang dilakukan oleh actor tertentu dengan use case tertentu pula. Hal ini digambarkan dengan garis antara actor terhadap use case tersebut. Asosiasi ini biasanya ditandai dengan garis 1 arah (ditandai dengan garis anak panah), jika terjadi komunikasi 2 arah, maka tanda panah tidak akan diperlukan.

### ***2. Generalization***

Ini adalah teknik mengidentifikasi relasi antara 2 aktor dan juga 2 Use case, dimana salah satunya akan meng-inherit dan juga menambah atau override sifat dari perangkat lainnya. Untuk teknik penggambarannya menggunakan garis bermata panah yang kosong. Garis akan diambil dari yang meng-inherit kemudian mengarah ke yang di-inherit.

### ***3. Dependency***

Dependency ini terbagi menjadi 2 macam, yaitu include dan juga extend.

Yang dimaksud dengan Include, adalah :

- Berfungsi untuk mengidentifikasi hubungan antara 2 use case, dimana use case yang satu akan memanggil use case yang lainnya.
- Apabila ada beberapa use case dengan aktifitas yang sama, maka bagian aktifitas tersebut akan dijadikan aktifitas tersendiri, dengan relasi dependensi use case semula ke use case yang baru.
- Biasanya digambarkan dengan garis putus-putus dengan mata panah notasi include yang pada garis. Kemudian arahkan mata panah sesuai dengan arah yang memanggil.

Yang dimaksud dengan extend, adalah :

- Apabila pemanggilan, memerlukan kondisi tertentu maka akan berlaku dependensi.
- Konsep extend ini berbeda dengan konsep extend dalam program Java.
- Dependensi ini akan digambarkan dengan dependensi include namun dengan arah panah yang berlawanan.

#### **4. *Aggregation***

Aggregation ini adalah sebuah bentuk association yang maka elemen yang satu akan berisi elemen lainnya.

Untuk membuat sebuah use case, ada langkah-langkah membuat Use Case yang harus Anda perhatikan dengan baik, antara lain :

- Mengidentifikasi para pelaku bisnis
- Mengidentifikasi use case sebagai persyaratan bisnis
- Membuat diagram sebagai model use case
- Mendokumentasikan naratif use case sebagai persyaratan bisnis

Adapun practical guidance yang perlu diperhatikan, ketika membangun diagram use case adalah ;

- Set konteks untuk target sistem
- Mengidentifikasi semua actor
- Mengidentifikasi semua use case
- Mendefinisikan semua asosiasi dari setiap actor dan setiap use case
- Mengevaluasi setiap actor dan juga setiap use case , guna mendapatkan kemungkinan perbaikan
- Mengevaluasi setiap use case guna dependensi include
- Mengevaluasi setiap use case guna dependensi exclude
- Mengevaluasi setiap actor dan juga setiap use case untuk generalisasi.

### ***III.1.12 Activity Diagram***




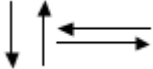
*Activity* diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

*Activity* diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dari sebagian besar transisi di trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan behaviour internal sebuah system (dan interaksi antara subsistem) secara eksak, tetap lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih.

Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana actor menggunakan system untuk melakukan aktivitas. Sama seperti state, standard UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas.

Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses parallel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik garis horizontal atau vertical. *Activity* diagram dapat dibagi menjadi beberapa objek swimlane untuk menggambarkan objek mana bertanggung jawab untuk aktivitas tertentu.

**Tabel 3.2 Simbol-Simbol Activity Diagram**

No	Simbol	Deskripsi
1.		Bagaimana objek dibentuk atau diawali
2.		Bagaimana objek dibentuk dan diakhiri
3.		Digunakan untuk suatu keputusan / atau tindakanyang harus diambil pada kondisi tertentu
4.		Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Tujuan utama dari Activity Diagram adalah menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses agar lebih mudah dipahami. Activity diagram juga digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya... Adapun tujuan pembuatan Activity Diagram lainnya yaitu :

- Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
- Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis
- Membantu memahami proses secara keseluruhan
- Menggambarkan aliran paralel, bercabang dan bersamaan dari system

Komponen yang ada pada Activity Diagram antara lain :

- ***Start Point/Initial State***(Titik Mulai)

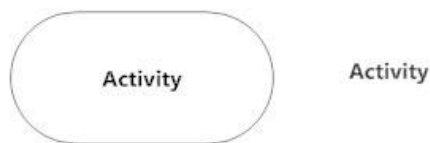
Start Point merupakan lingkaran hitam kecil, yang menandakan tindakan awal atau titik awal aktivitas untuk setiap diagram aktivitas.



***Gambar 3.3 Start Point/ Initial State***

- ***Activity***(Aktivitas)

Activity menunjukkan aktivitas yang dilakukan atau yang sedang terjadi dalam activity diagram.



***Gambar 3.4 Simbol Activity***

- ***Action Flow***(Arah)

Action Flow digunakan untuk transisi dari suatu tindakan ke tindakan yang lain atau menunjukkan aktivitas selanjutnya setelah aktivitas sebelumnya.

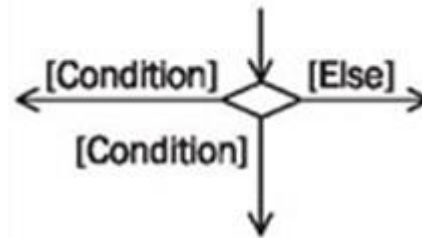


***Gambar 3.5 Simbol Action Flow***



- **Decision (Keputusan)**

Decision adalah suatu titik atau point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

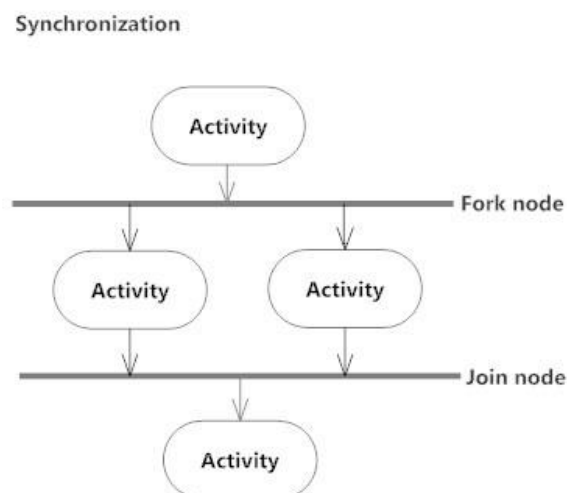


**Gambar 3.6 Simbol Decision**

- **Synchornization**

Synchornization dibagi menjadi 2 yaitu fork dan join.

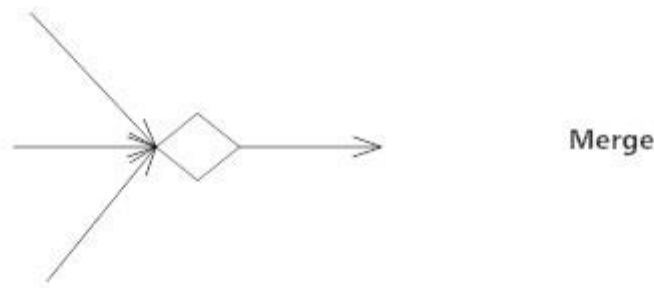
- Fork (percabangan) digunakan untuk memecah behaviour menjadi activity atau action yang paralel.
- Join (penggabungan) untuk menggabungkan kembali activity atau action yang paralel.



**Gambar 3.7 Simbol Synchronization**

- **Merge Event(Menggabungkan)**

Merge Event berfungsi untuk menggabungkan flow yang dipecah oleh decision.



**Gambar 3.8 Simbol Merge Event**

- **Swimlanes**

Swimlanes berfungsi untuk memecah activity diagram menjadi baris dan kolom untuk membagi tanggung jawab obyek-obyek yang melakukan aktivitas.

- **Final State/End Point(Titik Akhir)**

Final State menunjukan bagian akhir dari aktivitas.

### III.1.13 Klasifikasi Buku

**Klasifikasi** merupakan kata serapan dari bahasa Belanda, *classificatie*, yang sendirinya berasal dari bahasa Prancis *classification*. Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan.

Di dalam KBBI, klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Secara harafiah bisa pula dikatakan bahwa **klasifikasi** adalah pembagian sesuatu menurut kelas-

kelas. Menurut Ilmu Pengetahuan, Klasifikasi adalah Proses pengelompokkan benda berdasarkan ciri-ciri persamaan dan perbedaan.

Dalam perpustakaan koleksi harus dikelola dengan baik. Khususnya dalam hal penyajian, sangat perlu diperhatikan dengan baik bagaimana menata serta mengelompokkan koleksi perpustakaan. Hal ini bertujuan untuk mempermudah dalam pengelolaan serta juga memberikan kenyamanan juga kemudahan bagi pengguna perpustakaan agar dapat mencari atau menemukan apa yang dicari.

Anda dapat mencari dan menemukan buku yang dicari dengan cukup mudah. Penataan koleksi yang ada di perpustakaan memang tidak sembarangan. Koleksi di perpustakaan harus ditata menurut panduan khusus, salah satu panduan atau standar dalam mengelompokkan koleksi terutama buku di perpustakaan adalah Sistem *Dewey Decimal Classification* (DDC).

Sistem *Dewey Decimal Classification* (klasifikasi desimal *Dewey*) ini diciptakan oleh *Melvil Dewey* seorang pustakawan dari Amerika. Berikut merupakan pengelompokan koleksi perpustakaan berdasarkan sistem *Dewey Decimal Classification* (*Sistem Desimal Dewey*) contoh:

- 000 - Komputer, Informasi Dan Referensi Umum
- 100 - Filsafat Dan Psikologi
- 200 - Agama
- 300 - Ilmu sosial
- 400 - Bahasa
- 500 - Sains dan Matematika
- 600 - Teknologi
- 700 - Kesenian dan Rekreasi
- 800 - Sastra
- 900 - Sejarah dan Geografi

Penataan koleksi di perpustakaan harus mengikuti standar tertentu agar memberikan kemudahan dalam pengelolaan dan penyajian. Sistem pengelolaan koleksi perpustakaan dengan *Dewey Decimal Classification* ini merupakan

salah satu pilihan standar dalam mengelompokkan dan mengatur koleksi yang ada di perpustakaan. Sistem ini sudah banyak diterapkan di perpustakaan-perpustakaan di berbagai negara, membuktikan bahwa sistem ini mampu memberikan perpustakaan sistem yang mudah diaplikasikan namun tetap dapat menyajikan koleksi perpustakaan dengan baik.

Sistem *Dewey Decimal Classification* ini juga telah beberapa kali mengalami perubahan yang mengikuti perkembangan. Pembaruan terhadap sistem ini akan menjadikan sistem ini fleksibel dan cocok diterapkan pada banyak perpustakaan baik lama maupun perpustakaan baru. Sistem ini tentunya juga bisa diterapkan dalam sebuah toko buku untuk menyajikan buku-buku atau cetakan-cetakan lain yang hendak dijual sehingga calon pembeli dapat dengan mudah menemukan apa yang hendak dicari jika ditata atau dikelompokkan dengan sistem atau standar tertentu.

## **Bab IV**

### **Pelaksanaan Kerja Praktek**

#### **IV.1 Input**

Perpustakaan Desa Ciparay yang memiliki jumlah buku yang bertambah setiap tahun, baik berupa Sumbangan, maupun pengadaan sendiri oleh pihak desa dan pemerintah. Meskipun koleksi yang ada pada perpustakaan ini cukup lengkap, namun pengelolaannya belum efektif karena sistem pengelolaannya yang masih diketik secara manual di Microsoft Excel. misalnya; memasukan data buku, memasukkan data anggota, memasukkan data peminjaman dan pengembalian buku oleh anggota, membuat laporan peminjaman buku, dan lain-lain. Hal ini mengakibatkan pemborosan tenaga dan waktu, belum lagi jika ditinjau dari sisi keakuratan data yang tingkat kesalahan lebih mudah terjadi. Rencana pengembangan perangkat lunak diberikan oleh Bapak Ari Bama yaitu petugas perpustakaan dan bapak Yadhie Wahyu Q selaku kepala perpustakaan mutiara ilmu , baik secara tertulis maupun secara lisan. Maka dari itu dibuat sistem informasi perpustakaan berbasis MS Access yang diharapkan bisa mengelola data perpustakaan dengan lebih efektif dan efisien sehingga informasi yang dihasilkan bisa lebih akurat, tepat waktu dan relevan.

#### **IV.2 Proses**

Pada tahapan ini ditentukan masalah-masalah dan kebutuhan yang timbul pada perencanaan aplikasi yang sedang berjalan di Perpustakaan Mutiara Ilmu Desa ciparay dan akan dibuatkan solusi dari permasalahan yang ditemukan.

##### **IV.2.1 Analisis Sistem yang sedang berjalan**

Sebelum dilakukan perancangan, terlebih dahulu dilakukan analisis terhadap proses pengolahan data perpustakaan. Berikut adalah proses analisis data perpustakaan:

- a. Mencari data ketersediaan buku yang ada di Perpustakaan.

- b. Mencari data Anggota Perpustakaan.
- c. Mengumpulkan semua data yang ada.Selanjutnya data di olah untuk mendapatkan hasil yang diinginkan.

#### **IV.2.2 Analisis Kebutuhan Sistem**

Pembangunan aplikasi yang dilakukan dimulai dengan analisis kebutuhan perangkat lunak, adapun kebutuhan yang diperlukan adalah sebagai berikut :

##### **1. Kebutuhan Masukan**

- a. *User Name* dan *Password* yang merupakan Masukan saat Login aplikasi system Informasi Perpustakaan Mutiara Ilmu di Desa Ciparay.
- b. Data Anggota yang merupakan data Identitas Anggota perpustakaan.
- c. Data Buku Merupakan data seluruh buku yang tersedia di Perpustakaan yang dihimpun melalui Data Buku Induk yang ada di Perpustakaan
- d. Data Transaksi Merupakan Data yang diperoleh dari From Input Data Transaksi dengan memasukkan Nama Anggota dan buku yang dipinjam sehingga menghasilkan sebuah transaksi peminjaman Buku.

##### **2. Kebutuhan Proses**

Berikut ini adalah kebutuhan proses Sistem aplikasi perpustakaan Mutira Ilmu Desa Ciparay:

- a. Proses Login adalah Proses Memasukan data, dimana data yang dimasukkan adalah data anggota berupa Nomor Identitas (ID) yang dipilih menjadi admin, dan passwordnya untuk dapat masuk ke halaman menu utama.
- b. Proses Pengisian data Anggota yaitu proses memasukkan data anggota seperti Nama, alamat, jabatan, dan lain lain.
- c. Proses pengisian data Buku yaitu proses memasukkan data buku yang baru datang ke perpustakaan.

- d. Proses pengisian data Transaksi yaitu proses pemasukan data transaksi peminjaman buku dengan memasukkan ID peminjam dan Kode buku yang dipinjam.

### 3. Kebutuhan Perangkat

Perangkat dan *tools* yang digunakan dalam pembangunan Aplikasi Perpustakaan antara lain:

1. PC (Personal Computer)

***Tabel 4.1 Spesifikasi Perangkat***

Laptop	Acer ES 11
Prosesseor	Intel ® celeron® Proceessor N3050
RAM	2 GB DDR3 L Memory
Harddisk	500 GB
VGA	Intel ® HD Graphics

2. Sistem Operasi : Windows 10
3. Microsoft Access 2010

## IV.3 Output

Setelah diketahui data-data yang dibutuhkan sebagai sarana pendukung dalam pembuatan sistem informasi ini, maka akan dilanjutkan dengan perancangan sistem. Di sini perancangan sistem menggunakan Use Case Diagram sebagai gambaran sistemnya dan Activity diagram sebagai rancangan relasi tabelnya Selanjutnya dilakukan pemrograman dengan pemanfaatan MS.Access.

### IV.3.1 Perancangan Aplikasi

Untuk mempermudah gambaran sistem informasi Perpustakaan ini dibuatlah desain perancangan sebagai berikut :

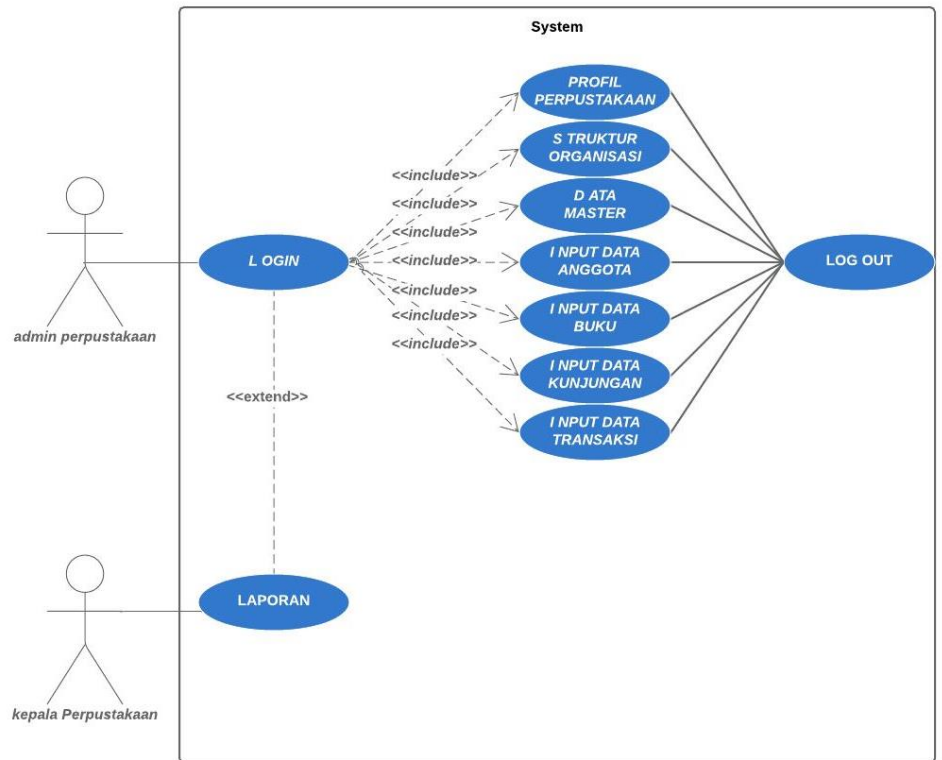
#### 1. Use Case Diagram

- a. Definisi Aktor pada *Use Case Diagram*

Definisi aktor pada *use case diagram* menggambarkan seorang yang berinteraksi dengan sistem, di mana hanya bisa

menginputkan informasi dan menerima informasi dari system.



b. Use Case Diagram Admin












**Gambar 4.1 Use Case Diagram Admin**

Aktor dalam gambar di atas adalah Petugas Perpustakaan yang bertindak sebagai admin. Admin memiliki hak akses penuh untuk mengelola seluruh data dalam sistem informasi. Hak akses yang dimiliki oleh admin antara lain: mengelola data Perpustakaan secara keseluruhan.

**Tabel 4.2 Deskripsi Use Case Diagram**

No	Use Case	Deskripsi
1	 Aktor	Merupakan <i>use case</i> untuk mengelola Data Perpustakaan.
2	 login	Merupakan <i>use case</i> untuk Login atau masuk kedalam system.

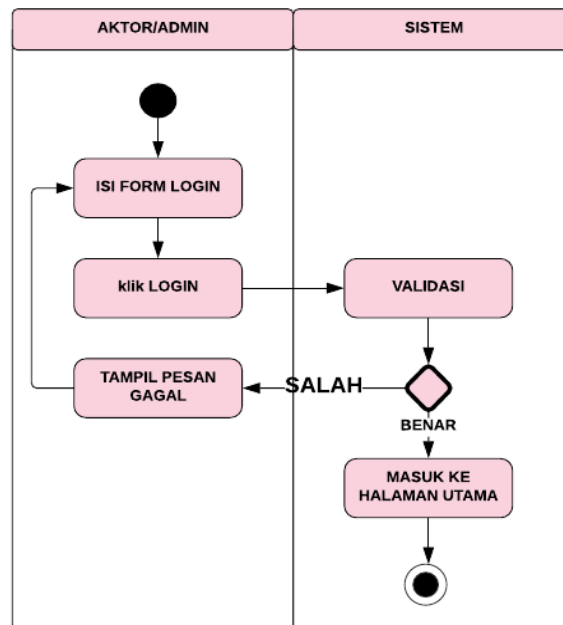


3		Merupakan <i>use case</i> untuk memasukan data buku.
4		Merupakan <i>use case</i> untuk memasukan data Anggota Perpustakaan.
5		Merupakan <i>use case</i> untuk memasukan data Transaksi Peminjaman Buku.
6		Merupakan <i>use case</i> untuk masuk ke data master.
7		Merupakan <i>use case</i> untuk Membuat Laporan
8		Merupakan <i>usecase</i> untuk masuk ke profil perpustakaan
9		Merupakan <i>usecase</i> untuk masuk ke struktur organisasi perpustakaan
10		Apa yang menghubungkan antara objek satu dengan objek lainnya
11		Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan

#### 4. Activity Diagram

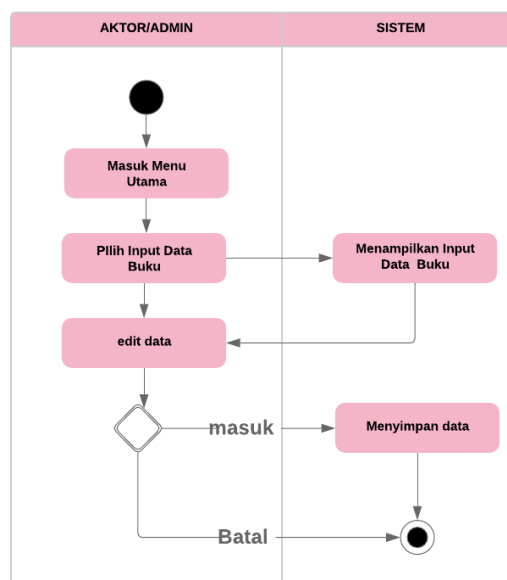
Berdasarkan *use case diagram* di atas, maka dihasilkan *Activity Diagram*. Salah satu *Activity Diagram* yaitu :

1. *Activity Diagram* login yang tersaji pada gambar dibawah ini :



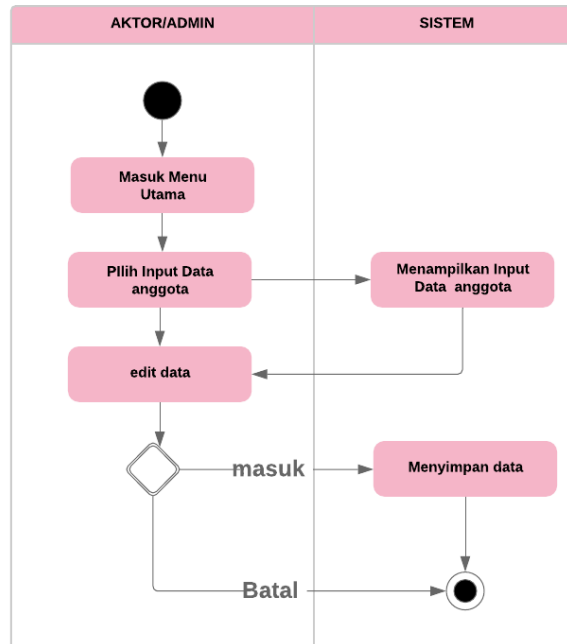
**Gambar 4.2 Activity Diagram Login**

2. *Activity Diagram* Input data Buku



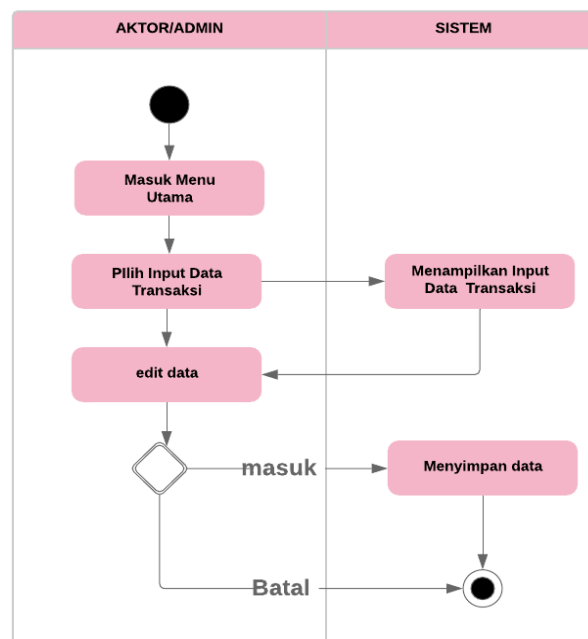
**Gambar 4.3 Activity Diagram Input data Buku**

### 3. Activity Diagram Input data anggota



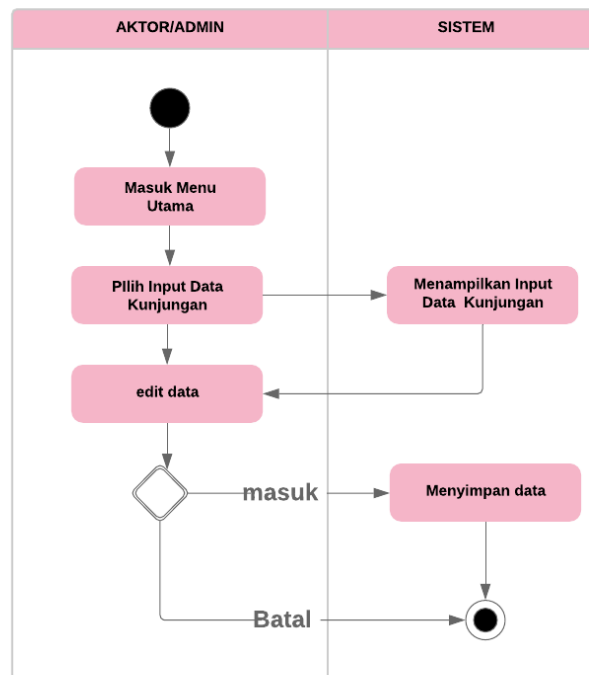
**Gambar 4.4 Activity Diagram Input data anggota**

### 4. Activity Diagram input data Transaksi



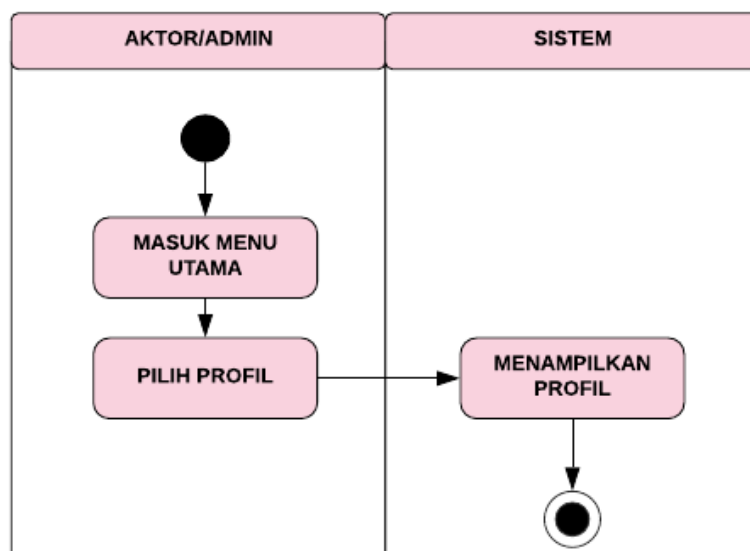
**Gambar 4.5 Activity Diagram Input data transaksi**

5. *Activity Diagram* Input data Kunjungan



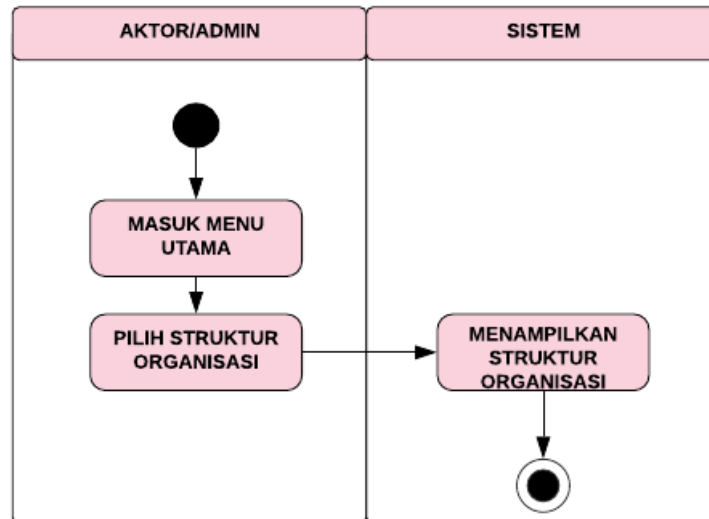
**Gambar 4.6** *Activity Diagram* Input data kunjungan

6. *Activity diagram* Untuk melihat Profil Perpustakaan.



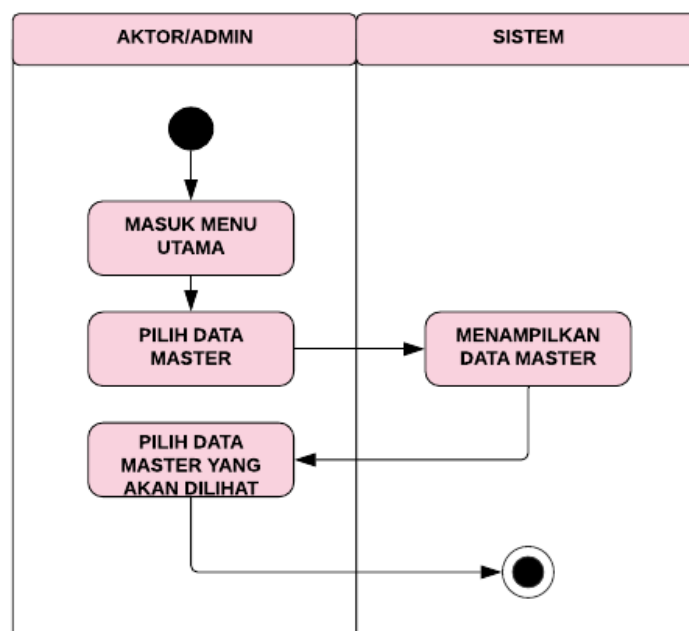
**Gambar 4.7** *Activity Diagram* untuk melihat profil perpustakaan

7. *Activity diagram* Untuk melihat Struktur Organisasi Perpustakaan.



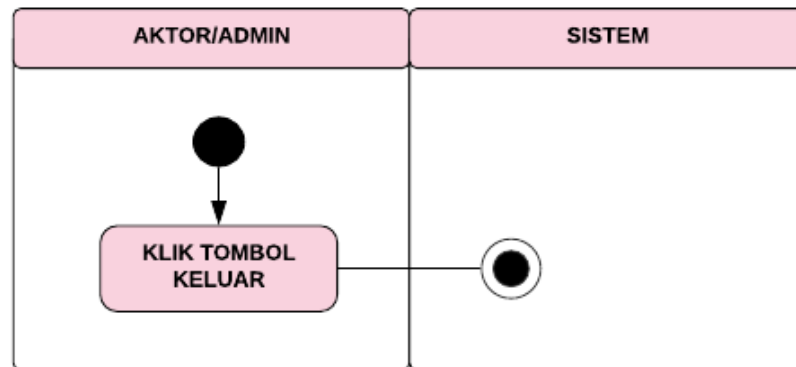
**Gambar 4.8 Activity Diagram Untuk melihat Struktur Organisasi Perpustakaan**

8. *Activity diagram* Untuk melihat Data Master Perpustakaan.



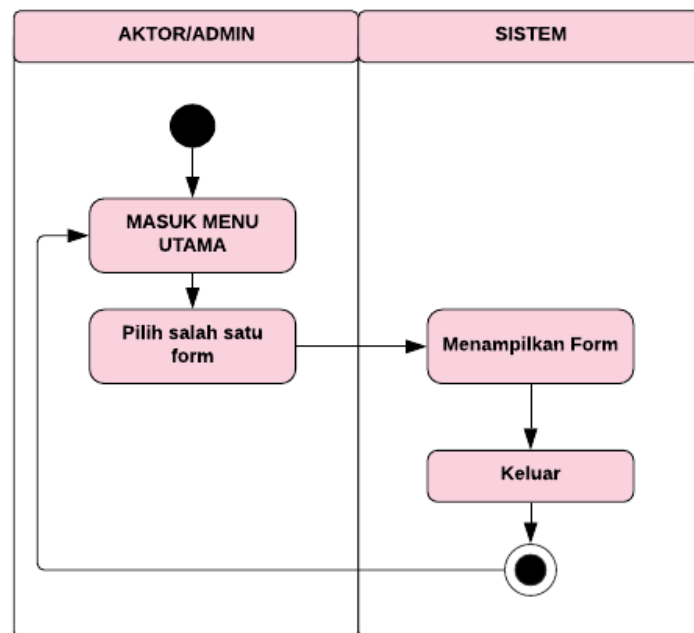
**Gambar 4.9 Activity Diagram Untuk melihat data master perpustakaan**

9. *Activity diagram* Untuk logout



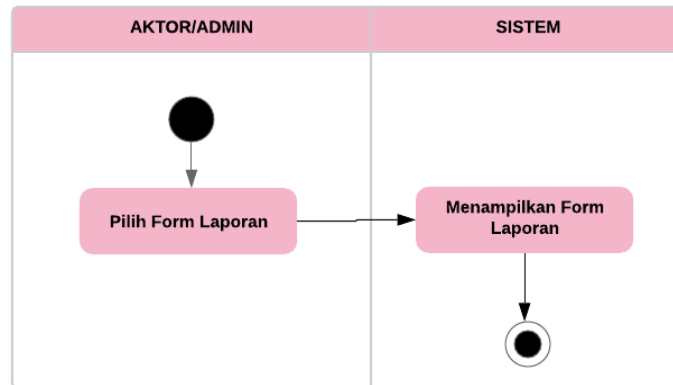
**Gambar 4.10 Activity Diagram Logout**

10. *Activity Diagram* Untuk Keluar dari Form.



**Gambar 4.11 Activity Diagram untuk keluar dari form**

### 11. Activity Diagram Untuk menampilkan Laporan



**Gambar 4.12 Activity Diagram untuk menampilkan Laporan**

Activity Diagram diatas menggambarkan proses login, logout, tambah data, lihat data, ubah data, hapus data, dan cetak data. Data yang dimaksud adalah data yang digunakan dalam sistem informasi Perpustakaan Mutiara Ilmu.

#### IV.3.2 Perancangan Basis Data

Basis Data dibutuhkan untuk menyimpan semua data-data pokok yang dibutuhkan untuk dijadikan informasi yang ditampilkan pada perancangan aplikasi manajemen data praktikum. Dibawah ini adalah beberapa rancangan tabel yang akan dibuat pada database perancangan aplikasi manajemen data praktikum.

##### 1. Rancangan Tabel Anggota

**Tabel 4.1 Keterangan Rancangan Tabel Anggota**

No	Field	Tipe	Ukuran	Keterangan
1	ID anggota	Text	7	Menyimpan Data ID Anggota
2	Nama	Text	50	Menyimpan data Nama anggota

3	Alamat	Text	50	Menyimpan data Alamat Anggota.
4	Status	LongInteger	13	Menyimpan data Status Anggota
5	Keterangan	Text	255	Memberikan keterangan

## 2. Rancangan Tabel Buku

***Tabel 4.2 Keterangan Rancangan Tabel Buku***

No	Field	Tipe	Ukuran	Keterangan
1	Tanggal	Date&Time	12	Menyimpan Data Tanggal
2	Kode Buku	Text	10	Menyimpan data Kode Buku
3	Kode Klasifikasi	Text	10	Menyimpan data Kode klasifikasi
4	Pengarang	Text	150	Menyimpan data nama pengarang buku
5	Nama Buku	Text	250	Menyimpan data nama buku
6	Tahun Buku	Number	4	Menyimpan data tahun buku.
7	Kualitas Buku	Long Int	4	Menyimpan data kualitas buku
8	Penerbit	Text	50	Menyimpan data penerbit buku
9	Asal buku	Long int	9	Menyimpan data asal buku
10	Harga	Number	10	Menyimpan data harga buku



11	Keterangan	Text	255	Menyimpan keterangan
----	------------	------	-----	----------------------

### 3. Rancangan Tabel Kunjungan

***Tabel 4.3 Keterangan Rancangan Tabel kunjungan***

No	Field	Tipe	Ukuran	Keterangan
1	Nama Anggota	Text	100	Menyimpan data Nama Anggota yang berkunjung
2	Alamat	Text	50	Menyimpan data alamat Anggota yang berkunjung
3	Tanggal	Date&Time	12	Menyimpan data Tanggal berkunjung
4	Jam	Date&Time	4	Menyimpan data Jam berkunjung
5	Status	Long Int	13	Menyimpan data Status pengunjung

### 4. Rancangan Tabel Transaksi Peminjaman

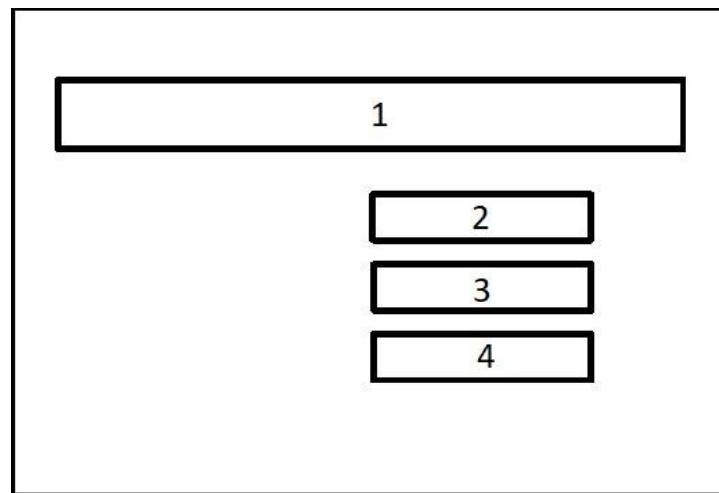
***Tabel 4.4 Keterangan Rancangan Tabel Transaksi Peminjaman***

No	Field	Tipe	Ukuran	Keterangan
1	ID Anggota	Text	15	Menyimpan data ID Anggota yang melakukan transaksi peminjaman Buku
2	Rak	LongInt	5	Menyimpan data rak Buku
3	Kode Buku	text	10	Menyimpan data kode Buku

4	Tanggal pinjam	Date&Time	12	Menyimpan data Tanggal peminjaman buku
5	Tanggal kembali	Date&Time	12	Menyimpan data pengembalian buku

### IV.3.3 Perancangan Tampilan User Face

#### 1. Perancangan Tampilan Switch Board.



***Gambar 4.13 Perancangan Tampilan Switch Board***

Di halaman awal atau halaman Switch Board terdapat 4 Kolom Opsional yaitu :

- 1) Kolom Judul Kerja Praktek atau Nama Aplikasi
- 2) Kolom Login adalah Tombol yang dipilih untuk Login dan selanjutnya akan masuk Ke Menu utama.
- 3) Kolom Laporan, jika kita memilih tombol Laporan kita akan masuk ke halaman Laporan.
- 4) Kolom Logout, adalah tombol yang dipilih Untuk Log Out atau keluar dari Aplikasi.

## 2. Perancangan Tampilan Login

The diagram shows a login form layout within a rectangular frame. It consists of five numbered boxes arranged vertically and horizontally:

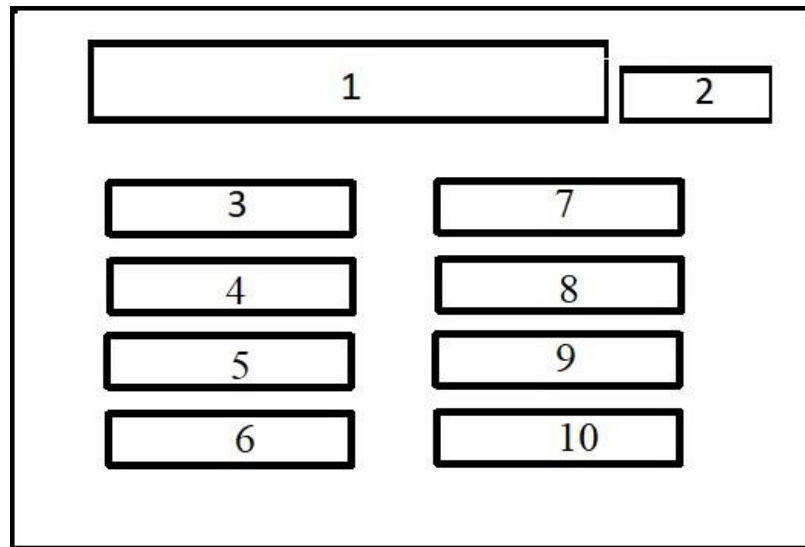
- Box 1: A wide rectangular box at the top center.
- Box 2: A smaller rectangular box centered below Box 1.
- Box 3: A rectangular box centered below Box 2.
- Box 4: A small rectangular box on the left, centered below Box 3.
- Box 5: A small rectangular box on the right, centered below Box 3.

***Gambar 4.14 Perancangan Tampilan Login***

Jika dihalaman awal atau Form Switch Board Admin atau User memilih untuk Login, maka akan muncul Form Login. Login memiliki 5 Kolom, yaitu :

- 1) Kolom Judul Login
- 2) Kolom ID User adalah kolom yang perlu diisi dengan ID User Admin untuk masuk ke Menu Utama.
- 3) Kolom Password untuk memasukkan Password atau Kata sandi nya.
- 4) Kolom Login, jika kita sudah memasukkan ID User dan password dengan benar, selanjutnya admin menekan tombol Login maka akan masuk ke Menu Utama. Tapi sebaliknya, jika ID User atau password yang dimasukkan salah, maka akan ada pemberitahuan bahwa admin tidak terdaftar, dan selanjutnya admin akan keluar dari Form Login dan kembali ke Switch Board.
- 5) Kolom Cancel adalah Kolom yang digunakan untuk membatalkan Login, jika tombol cancel dipilih maka admin akan kembali ke Form Switch Board.

### 3. Perancangan Tampilan Menu utama



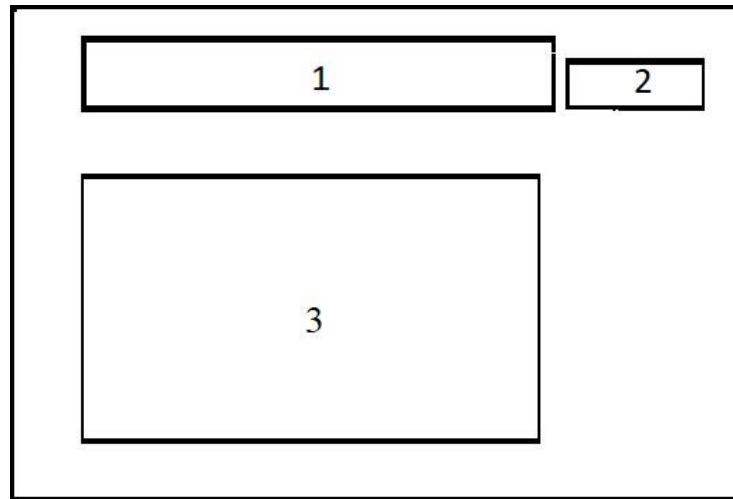
***Gambar 4.15 Perancangan Tampilan Menu Utama***

Selanjutnya, jika Admin telah Login maka akan muncul Form Menu utama, Dalam Form Menu utama terdapat 10 Kolom, diantaranya:

- 1) Kolom Judul aplikasi
- 2) Kolom Log Out, jika admin akan keluar dari menu utama.
- 3) Kolom Profil adalah Kolom yang menampilkan Profil Perpustakaan.
- 4) Kolom Struktur Organisasi adalah Kolom yang menampilkan Struktur Organisasi Perpustakaan.
- 5) Kolom Data Master adalah Kolom yang menampilkan seluruh data data yang telah diinput di form Input data.
- 6) Kolom Laporan adalah kolom yang menampilkan Form Laporan.
- 7) Kolom Input Data Buku, adalah Kolom yang digunakan untuk memasukkan data-data buku yang masuk.
- 8) Kolom Input Data Anggota, adalah Kolom yang digunakan untuk memasukkan data-data Anggota Baru.
- 9) Kolom Input Data Transaksi, adalah Kolom yang digunakan untuk memasukkan data-data Transaksi Peminjaman Buku.

10) Kolom Input Data Kunjungan, adalah Kolom yang digunakan untuk memasukkan data-data Anggota yang berkunjung ke perpustakaan.

#### 4. Perancangan Tampilan Profil Perpustakaan

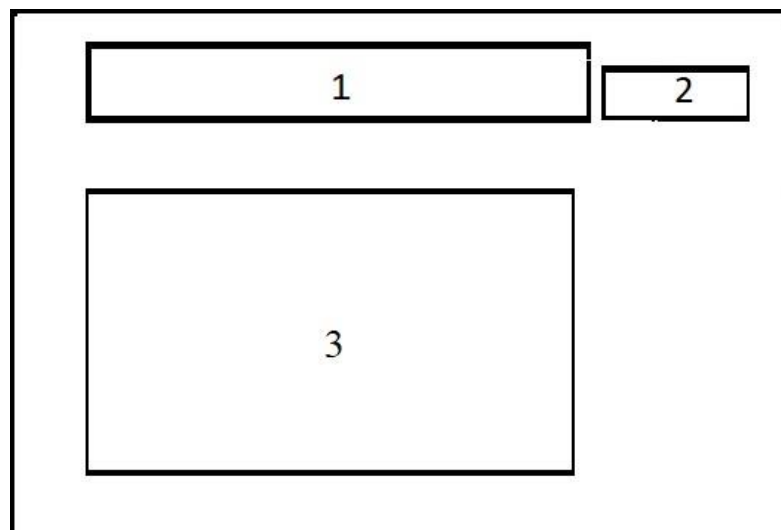


**Gambar 4.16 Perancangan Tampilan Profil Perpustakaan**

Jika di Menu Utama, kita memilih Kolom Profil, maka akan tampil Form Profil Perpustakaan yang memiliki 3 Kolom, Yaitu :

- 1) Kolom Judul Form Profil
- 2) Kolom KELUAR adalah Kolom yang digunakan untuk keluar dari form Profil.
- 3) Tampilan data Profil Perpustakaan.

#### 5. Perancangan Tampilan Struktur Perpustakaan



**Gambar 4.17 Perancangan Tampilan Struktur Perpustakaan**

Jika di Menu Utama, kita memilih Kolom Struktur Organisasi, maka akan tampil Form Struktur Organisasi Perpustakaan yang memiliki 3 Kolom, Yaitu :

- 1) Kolom Judul Form Struktur Organisasi Perpustakaan.
- 2) Kolom KELUAR adalah Kolom yang digunakan untuk keluar dari form Struktur Organisasi.
- 3) Tampilan Struktus Organisasi Perpustakaan.

#### 6. Perancangan Tampilan Data Master

The diagram illustrates the layout of the Data Master Form. It consists of a large rectangular frame containing several smaller rectangular boxes. At the top, there are two boxes: a wide one labeled '1' and a smaller one labeled '2' to its right. Below these, on the left side, there is a vertical stack of four boxes labeled '3', '4', '5', and '6' from top to bottom.

***Gambar 4.18 Perancangan Tampilan Data Master***

Jika di Menu Utama, kita memilih Kolom Data Master, maka akan tampil Form Data Master Perpustakaan yang memiliki 6 Kolom, Yaitu :

- 1) Kolom Judul Form Data Master.
- 2) Kolom KELUAR adalah Kolom yang digunakan untuk keluar dari form Data Master.
- 3) Kolom Data Buku adalah Kolom yang menampilkan seluruh data data Buku yang terdapat diperpustakaan yang telah berhasil disimpan ketika data buku diinput di Form Input data.
- 4) Kolom Data Anggota adalah Kolom yang menampilkan seluruh data data Anggota Perpustakaan yang telah berhasil disimpan ketika data Anggota diinput di Form Input data.
- 5) Kolom Data Transaksi adalah Kolom yang menampilkan seluruh data data Transaksi peminjaman buku yang telah

berhasil disimpan ketika data transaksi diinput di Form Input data.

- 6) Kolom Data Kunjungan adalah Kolom yang menampilkan seluruh data data Kunjungan Anggota Perpustakaan yang telah berhasil disimpan ketika data Kunjungan diinput di Form Input data.

#### 7. Perancangan Tampilan Input Data

The diagram illustrates the layout of the input data form. It consists of a large rectangular container. Inside this container, at the top, is a horizontal rectangular box labeled '1'. Below box '1' is a larger rectangular box labeled '2'. At the bottom of the container, there is a row of five small square buttons labeled '3', '4', '5', '6', and '7' from left to right. Directly below these buttons is a single horizontal rectangular box labeled '8'.

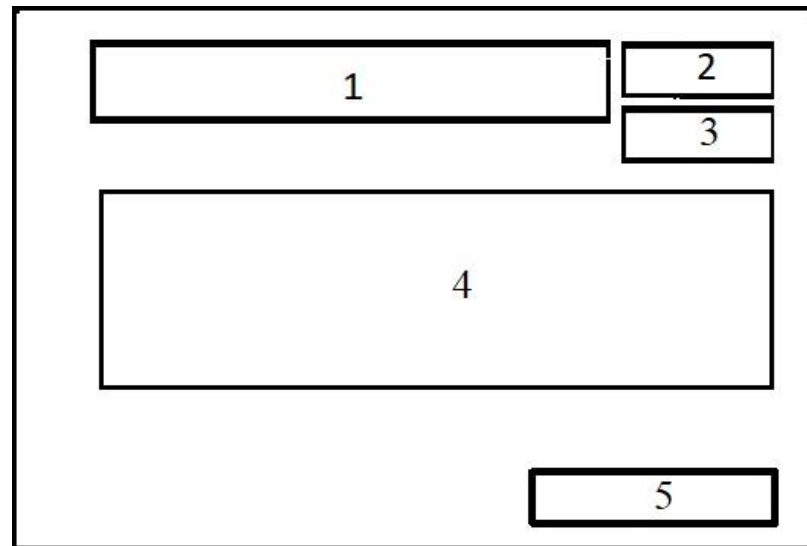
**Gambar 4.19 Perancangan Tampilan Input Data**

Jika di Menu Utama, kita memilih salah satu Kolom Input Data, diantaranya: Input data Buku, Input Data Anggota , Input data Transaksi, atau Input data Kunjungan, maka akan tampil Form Data Master Perpustakaan yang memiliki 8 Kolom, Yaitu :

- 1) Kolom Judul Input Data.
- 2) Kolom Pengisian data.
- 3) Tombol *Previous* adalah tombol untuk melihat data yang telah diinput sebelumnya.
- 4) Kolom Tambah adalah kolom yang digunakan untuk menambah data baru.
- 5) Kolom Hapus adalah kolom yang digunakan untuk menghapus data yang telah diinput.
- 6) Kolom Simpan adalah Kolom yang digunakan untuk menyimpan data yang telah diinput.

- 7) Tombol *Next* adalah tombol yang digunakan untuk melihat data selanjutnya yang telah berhasil disimpan.
- 8) Kolom Keluar adalah kolom yang digunakan untuk keluar dari Form Input data.

#### 8. Perancangan Tampilan Laporan



***Gambar 4.20 Perancangan Tampilan Laporan***

Jika di Menu Utama ataupun di Swtch Board kita memilih Kolom Laporan, maka akan tampil Form Laporan yang memiliki 5 Kolom, Yaitu :

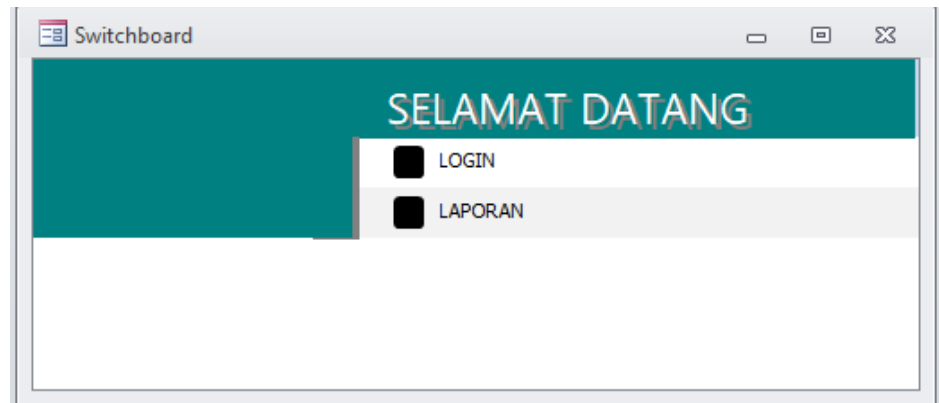
- 1) Kolom Judul Laporan
- 2) Kolom Cetak adalah kolom yang digunakan untuk mencetak Laporan.
- 3) Kolom Keluar adalah kolom yang digunakan untuk keluar dari Form Laporan.
- 4) Kolom Tampilan Laporan.
- 5) Kolom Tanggal Laporan.

### **IV.3.4 Tampilan User Interface**

#### **1. Halaman Switchboard**

Halaman Switch Board merupakan halaman awal sistem informasi. Dimana user bisa memilih Login untuk masuk ke system atau memilih laporan jika hanya ingin melihat laporan.

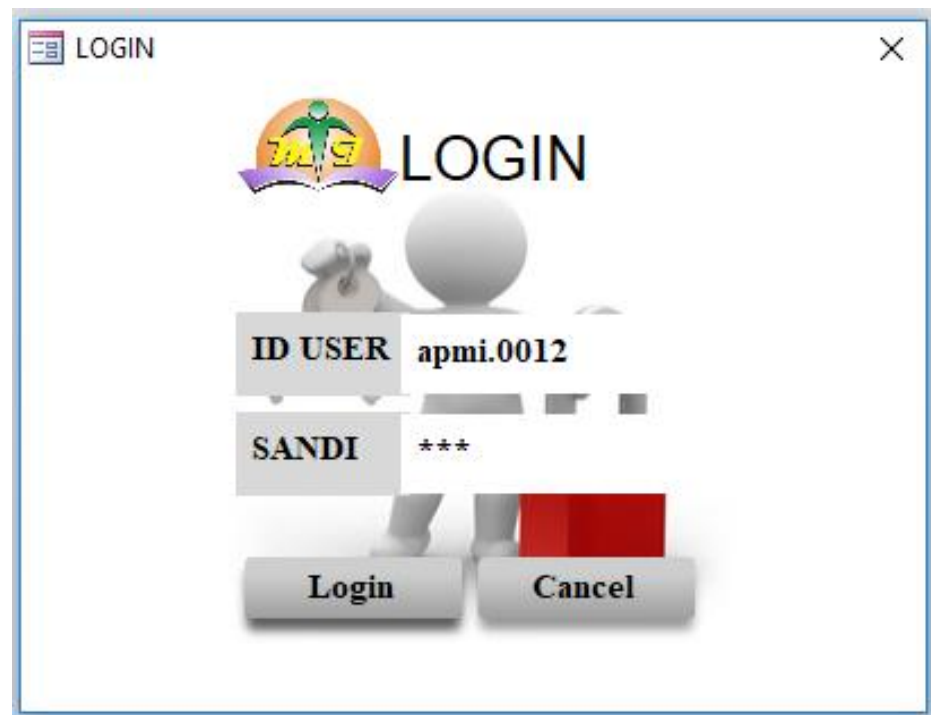




**Gambar 4.21 Halaman Switch Board**

## 2. Halaman login

Halaman login ditujukan pada Gambar IV.12 Admin harus login menggunakan *username* dan *password*.



**Gambar 4.22 Gambar implementasi Form login**

## 3. Menu Utama

Setelah mengisi kolom username dan password kemudian klik tombol Login. Jika pengisian benar, tampilan akan berpindah menjadi tampilan menu utama. Pada Implementasi Form Menu Utama, Terdapat beberapa Menu yang tersedia seperti yang terdapat pada gambar IV.13 dibawah ini:



**Gambar 4.23 implementasi Form Menu Utama**

Ada beberapa pilihan didalam menu utama yaitu antara lain:

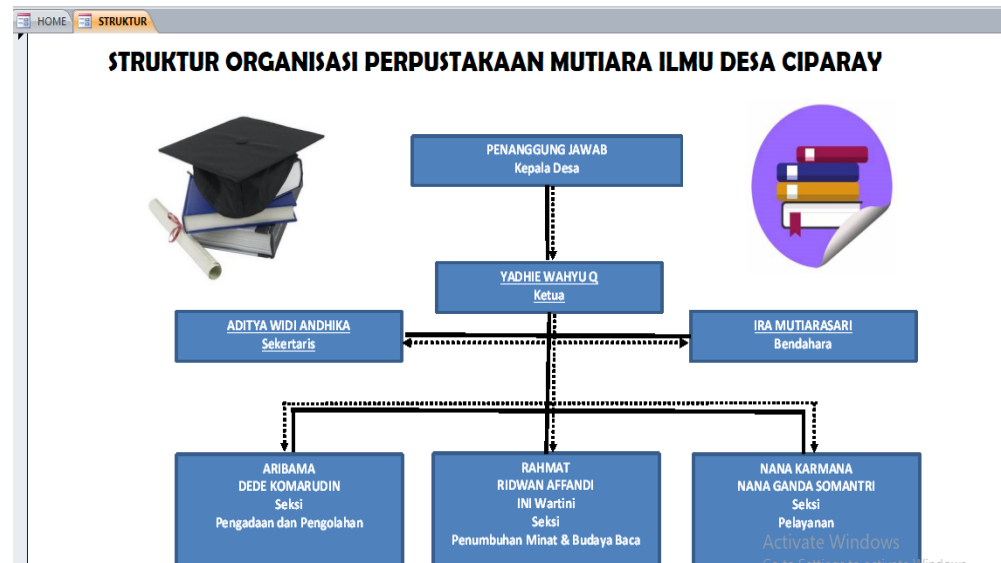
- a. Profil perpustakaan.

Profil perpustakaan memuat data data perpustakaan seperti gambar dibawah ini:

1. NAMA PERPUSTAKAAN	: "PERPUSTAKAAN MUTIARA ILMU"
2. DESA/ KELURAHAN	: CIPARAY
3. KECAMATAN	: CIPARAY
4. KABUPATEN/ KOTA	: BANDUNG
5. PROVINSI	: JAWA BARAT
6. ALAMAT	: Jl. Bojong-Parigi No.111 Desa Ciparay
FACEBOOK	:
INSTAGRAM	:
TWEETER	:
BLOG	:
E-MAIL	:
7. DIDIRIKAN TAHUN	:
BERDASARKAN SK NO	:
8. EFEKTIF BEROPRASI	: JULI,2012
9. NAMA PETUGAS	: YADHIE WAHYU Q
10. PENANGGUNG JAWAB	: H. DACHYA MULYANA

**Gambar 4.24 Gambar implementasi Form profil perpustakaan.**

## b. Struktur Organisasi



Gambar 4.25 Implementasi Form Struktur Organisasi

## c. Tampilan Data Master

Data master adalah data yang menghimpun seluruh data yang berhasil disimpan di form input sebelumnya. Berikut adalah tampilan data master.



Gambar 4.26 Gambar implementasi Form Data Master

Dan ketika User memilih salahsatu tombol, misalkan tombol data anggota maka tampilan akan menjadi tampilan table seperti pada gambar berikut:

NO	ID ANGGOTA	NAMA	ALAMAT	TANGGAL	JAM	STATUS
1	APML0001	H.DACHYA MULVANA	Kp. Bojong RW.06			PERANGKAT DESA
2	APML0002	ASEP KUSMANA	Kp. Hujung RW.01			
3	APML0003	S. NUGRAHA	Kp. Hujung RW.01			
4	APML0004	RAHMAT	Kp. Babakan Caringin RW.05			
5	APML0005	RIDWAN AFFANDI	Kp. Parigi RW 07			
6	APML0006	DEDE KOMARUDIN	Kp. Lio RW 04			
7	APML0007	NANA GANDA SOMANTRI	Kp. Hujung RW.01			
8	APML0008	NANA KARMANA	Kp. Bojong RW.06			
9	APML0009	NIKNIK PERMANIK	Kp. Bojong RW.06			
10	APML0010	NENG ROHAETI	Kp. Babakan Caringin RW.05			
11	APML0011	IRA MUTIARA SARI	Kp. Bojong RW.06			
12	APML0012	ARI BAMA	Kp. Lio RW 09			ADMIN
13	APML0013	MAMAN SURYAMAN	Kp. Bojong RW.06			

**Gambar 4.27 Gambar Tampilan Data Master**

d. Form Input Data Buku

Pilih salah satu menu yang ditampilkan di Home, misalnya User akan memasukan data Buku, klik tombol Input Data Buku, maka akan tampil form pengisian data seperti pada gambar berikut ini:

**Gambar 4.28 Gambar implementasi Form Input data Buku**

e. Form Input data Anggota

Form input data anggota digunakan untuk menambahkan nama anggota baru ,menyimpan nya, ataupun menghapus nya, berikut tampilan input data anggota:

**INPUT DATA ANGGOTA**  
**PERPUSTAKAAN MUTIARA ILMU**

NO: 1  
ID ANGGOTA: APML0001  
NAMA: H.DACHVA MULYANA  
ALAMAT: Kp. Bojong RW.06  
TANGGAL:   
JAM:   
STATUS: **PERANGKAT DE**  
KETERANGAN:   
ADMIN  
BPD  
KARANG TARUNA  
LPMD  
MASVAKAT  
MUI  
PELAJAR  
PERANGKAT DEIA  
PKK

HAPUS SIMPAN  
KELUAR

Activate Windows  
Go to Settings to activate Windows

**Gambar Error! No text of specified style in document..1 Gambar implementasi Form Input data Anggota.**

f. Input data transaksi peminjaman

Form input data transaksi peminjaman digunakan untuk menginput data transaksi peminjaman buku dan tanggal pengembaliannya. Berikut ini adalah gambar form input data transaksi peminjaman buku:

**INPUT DATA TRANSAKSI PEMINJAMAN**  
**PERPUSTAKAAN MUTIARA ILMU**

NO: 1  
NAMA ANGGOTA: APML0001  
NAMA BUKU: 000-1  
TANGGAL PINJAM: 000-1  
TANGGAL KEMBALI: 000-1

TAMBAH HAPUS

000-1 Internet Masuh  
000-2 Terampil Mengs  
000-3 Peliharalah KOI  
000-4 Microsoft Word  
000-5 Trib mengomp1  
000-6 Mahir Menggun  
152-7 Emosi  
153-8 Orang Desa Bin  
297-9 Amaliah DI Buk  
297-10 Riba Dalam Isla  
297-11 Makanan Yang  
297-12 Mengenal Sister  
297-13 Ananiah Ghada  
297-14 Meneladani Wa

Activate Windows  
Go to Settings to activate Windows

**Gambar 4.30 Gambar implementasi Form Input data Transaksi**

g. Input data Kunjungan

Form Input data kunjungan digunakan untuk memasukan data anggota yang berkunjung ke perpustakaan. Berikut adalah gambar form data kunjungan:

**INPUT DATA KUNJUNGAN  
PERPUSTAKAAN MUTIARA ILMU**

NO:

NAMA ANGGOTA:

ALAMAT:

TANGGAL:

JAM:

STATUS:

KETERANGAN:

TAMBAH HAPUS SIMPAN

KELUAR

**Gambar 4.31 Gambar implementasi Form Input data kunjungan**

h. Laporan

Form laporan menampilkan laporan harian perpustakaan, diantaranya: siapa saja yang meminjam buku, buku apa saja yang dipijam, dan lama peminjaman. Berikut adalah gambar form Laporan Harian perpustakaan.

**LAPORAN HARIAN TRANSAKSI PEMINJAMAN BUKU**

CETAK

KELUAR

NAMA ANGGOTA	NAMA BUKU	TANGGAL PINJAM	TANGGAL KEMBALI
APMII.0009	000-1	01-Oct-19	03-Oct-19
APMII.0002	000-4	24-Oct-19	25-Oct-19
APMII.0002	000-2	01-Oct-19	07-Oct-19

Thursday, October 24, 2019  
1:10:09 PM

**Gambar 4.32 Gambar implementasi Form Laporan Harian**

#### **IV.3.5 Pencapaian Hasil**

Adapun hasil yang dicapai dari kerja praktek di Perpustakaan Mutiara Ilmu Desa Ciparay ini berupa Aplikasi system Informasi Basis data Microsoft Access. Aplikasi ini menawarkan fungsionalitas proses klinis sebagai berikut:

- *Data View*
- *Delete Data*
- *Edit Data*
- *Create Data*
- *Data Print*

Setelah sistem atau aplikasi perpustakaan selesai dibuat maka sistem tersebut diujicobakan pada pengguna. Pada tahap ini masih ada beberapa saran dan perbaikan yang dilakukan seperti sistem peng-kodean (kode buku) yang digunakan, tambahan data field, ketepatan data dan lain-lain. Perbaikan dilakukan hingga system berjalan sebagaimana yang diinginkan.

## **Bab V**

### **Penutup**

#### **V.1 Kesimpulan dan saran mengenai pelaksanaan KP**

Dari penjelasan diatas dapat disimpulkan bahwa kegiatan Praktik Kerja Lapangan sangat bermanfaat untuk mahasiswa/siwi ataupun untuk pihak industri. Selain itu kegiatan Praktik Kerja Lapangan juga menjadi salah satu sarana untuk mengasah keterampilan khususnya dalam hal praktik dimana mereka bisa belajar lebih luas dalam dunia kerja serta sebagai ajang untuk melatih mahasiswa/siswi menjadi generasi muda yang bertanggung jawab dan profesional.

##### **V.1.1 Kesimpulan Pelaksanaan Kerja Praktek**

1. Mahasiswa dapat mengaplikasikan ilmu yang diperoleh selama perkuliahan untuk menyelesaikan permasalahan di dunia nyata.
2. Mahasiswa dapat mengetahui ilmu dan keterampilan yang dibutuhkan untuk memasuki dunia kerja di era globalisasi, seperti:
  - Keterampilan berkomunikasi dan bekerja sama dengan orang lain.
  - Ilmu dasar mengenai bidang spesifik yang diperoleh selama perkuliahan. Misalnya ilmu dasar di bidang informatika, ilmu dasar di bidang ekonomi, dan sebagainya.
  - Keterampilan menganalisis permasalahan untuk dicari solusinya.
  - Ilmu pengetahuan umum.
  - Keterampilan mempelajari hal yang baru dalam waktu relatif singkat.
3. Mahasiswa menyadari pentingnya etos kerja yang baik, disiplin, dan tanggung jawab dalam menyelesaikan suatu pekerjaan.
4. Kerja praktek dapat melatih mahasiswa untuk bekerja sama dalam suatu tim, baik antar peserta kerja praktek maupun dengan petugas Perpustakaan.
5. Mahasiswa memperoleh tambahan ilmu yang tidak diperoleh di proses perkuliahan.



### **V.1.2 Saran Pelaksanaan KP**

Adapun saran mengenai pelaksanaan kerja praktek antara lain:

1. Perlu ditumbuhkan kebiasaan belajar secara mandiri (*self-learning*) di kalangan mahasiswa, khususnya dalam mempelajari teknologi secara aplikatif. Salah satu fasilitas yang tersedia yang mendukung proses pembelajaran secara mandiri ini adalah koneksi internet yang cukup cepat.
2. Perlu adanya kemampuan mahasiswa untuk menggabungkan seluruh ilmu yang pernah didapat di perkuliahan dalam proses pembangunan perangkat lunak.
3. Perlu adanya bimbingan secara lebih intensif bagi mahasiswa kerja praktek.
4. Jika memungkinkan, dalam pelaksanaan kerja praktek mahasiswa dapat dilibatkan dalam suatu proyek di mana mahasiswa dapat bekerja sama dengan pegawai lain.

## **V.2 Kesimpulan dan saran mengenai substansi yang digeluti selama KP**

### **V.2.1 Kesimpulan**

Berdasarkan hasil penelitian dan pengalaman yang didapatkan, dapat disimpulkan bahwa Sistem informasi perpustakaan berbasis komputer dengan menggunakan MS Access dibangun untuk memenuhi kebutuhan pengguna yang mengelola data perpustakaan Mutiara Ilmu Desa Ciparay, sehingga dihasilkan sebuah program aplikasi yang dinamakan “Aplikasi Sistem Informasi Perpustakaan Mutiara Ilmu Desa Ciparay”. Perancangan aplikasi ini dimulai dengan pembuatan nama database yang dilanjutkan dengan perancangan objek database dengan konsep diagram UML, Use Case Diagram, dan Activity Diagram. Table, Form, Report dan Macro dibuat sesuai analisis sistem. Aplikasi yang dihasilkan telah diujicobakan pada perpustakaan Mutiara Ilmu desa Ciparay dengan beberapa saran dan perbaikan.

### **V.2.2 Saran**

Sistem ini masih bersifat desktop dan belum memanfaatkan jaringan internet sebagai media komunikasi data. Aksesibilitas sistem informasi ini akan menjadi semakin luas tentunya dengan teknologi internet. Pengembangan sistem ke arah web-based system akan memberikan manfaat yang lebih besar lagi bagi mahasiswa dan Perpustakaan Mutiara Ilmu Desa Ciparay.

## **Lampiran A. TOR (TERM OF REFERENCE)**

Sebelum melaksanakan kerja praktek penulis melakukan beberapa metode penelitian yaitu diantaranya observasi, *interview*, dan studi pustaka. Setelah mengamati dan mempelajari lokasi kerja praktek yang telah ditentukan dan si setuju oleh instansi tempat kerja praktek. Setelah kepala instansi menyetujui penulis melakukan kerja praktek tersebut. Penulis menjelaskan bahwa penulis memiliki tugas yang harus dikerjakan di lokasi selama kerja praktek yaitu:

1. Membantu dalam penataan pemberkasan data Perpustakaan.
2. Membantu Pendataan panitia KPPS.
3. membantu mengajarkan pengunjung perpustakaan tentang beberapa Aplikasi dasar Komputer
4. Membantu Membersihkan dan merapikan Perpustakaan dan tempat pelatihan computer.

Bandung, 21 April 2019

Disetujui Oleh:

Peserta Kerja Praktek

Ketua Perpustakaan,

**Asri Hardiyanti**

**Yadhie Wahyu Q**

## Lampiran B. Log Activity

***Tabel B.1 Log Activity***

Minggu/Tgl	Kegiatan	Hasil
Senin, 18 Februari 2019	- Pengenalan lingkungan Perpustakaan	-Mendapatkan informasi mengenai lingkungan perpustakaan.
Kamis, 21 Februari 2019	-Wawancara dengan petugas perpustakaan mengenai apa saja yang dibutuhkan di perpustakaan.	-Mendapatkan pembimbing lapangan. -Mendapatkan Informasi kebutuhan perpustakaan.
Senin, 25 Februari 2019	-Pengumpulan data perpustakaan yang dibutuhkan dalam pengerjaan aplikasi.	-Mendapatkan Data data perpustakaan.
Kamis, 28 Februari 2019	-Survei ruangan Perpustakaan dan membantu merapikan Buku buku.	-Mendapatkan informasi mengenai ruangan-ruangan yang ada dipergustakaan.
Senin, 4 Maret 2019	-Survei ruangan Pelatihan Komputer dan membantu Membersihkan Ruangan pelatihan tersebut.	-Mendapatkan Informasi mengenai kegiatan yang dilakukan di perpustakaan.
Kamis, 7 Maret 2019	-Membantu Pendataan Panitia KPPS untuk Pemilihan Presiden.	-Mendapatkan Informasi dan pengetahuan mengenai pendataan panitia KPPS.
Senin, 11 Maret 2019	membantu mengajarkan pengunjung perpustakaan tentang beberapa Aplikasi dasar Komputer.	-Mendapatkan pengetahuan mengenai pengajaran aplikasi dasar computer.
Kamis, 21 Maret 2019	Presentasi Aplikasi serta evaluasi kekurangan Aplikasi.	-Laporan.

