

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1. Analisa Kebutuhan *Software*

A. Tahapan Analisis

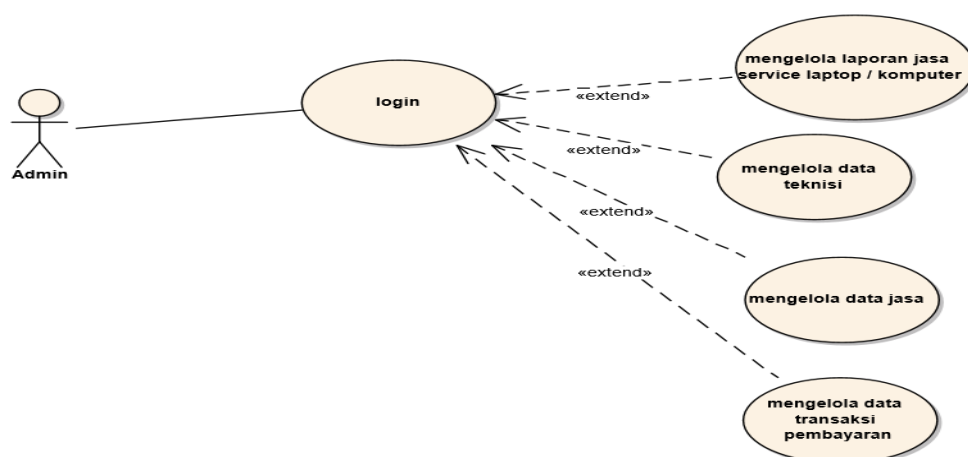
Aplikasi sistem jasa dan *service* komputer / laptop adalah salah satu media untuk membantu menyelesaikan masalah (biasanya yang berhubungan dengan komputer). Berikut ini spesifikasi kebutuhan (*system requirement*) dari sistem aplikasi jasa dan *service* komputer / laptop:

Halaman Administrasi.

- B1. Administrasi dapat mengelola data Teknisi
- B2. Administrasi dapat mengelola data jasa
- B3. Administrasi dapat mengelola data transaksi pembayaran
- B4. Administrasi dapat mengelola data laporan jasa *service* komputer / laptop

B. Use Case Diagram.

1 Use Case Diagram Aplikasi Admin



Sumber : Bzs Komputer (2016)

Gambar IV.1 Use Case Diagram Halaman Admin

a. Deskripsi Use Case Mengelola Data Teknisi

Tabel IV.1 Deskripsi *Use Case Diagram* Mengelola Data Teknisi

Use Case Name	Mengelola data Teknisi
<i>Requirements</i>	B1
<i>Goal</i>	Admin dapat menambah, mengedit dan menghapus data Teknisi
<i>Pre-conditions</i>	Admin telah login
<i>Post-conditions</i>	Data Teknisi, tersimpan atau terhapus
<i>Failed end condition</i>	Gagal menyimpan atau menghapus data Teknisi
<i>Primary actors</i>	Admin
<i>Main Flow / Basic Path</i>	1. Admin melihat data Teknisi yang ada 2. Admin menambah data Teknisi 3. Admin menyimpan data Teknisi
<i>Invariant 1</i>	2a. Admin mengedit data Teknisi.
<i>Invariant 2</i>	2b. Admin menghapus data Teknisi

b. Deskripsi Use Case Mengelola Data Jasa

Tabel IV.2 Deskripsi *Use Case Diagram* mengelola data Jasa

Use Case Name	Mengelola data Jasa
<i>Requirements</i>	B2
<i>Goal</i>	Admin dapat menambah, mengedit dan menghapus Jasa
<i>Pre-conditions</i>	Admin telah login
<i>Post-conditions</i>	Data Jasa tersimpan, ter-update atau terhapus
<i>Failed end condition</i>	Gagal menyimpan data Jasa
<i>Primary actors</i>	Admin
<i>Main Flow / Basic Path</i>	1. Admin melihat data Jasa 2. Admin menambah data Jasa 3. Admin menyimpan data Jasa

<i>Invariant 1</i>	2a. Admin mengedit data Jasa
<i>Invariant 2</i>	2b. Admin menghapus data Jasa

c. Deskripsi *Use Case* Mengelola Data Transaksi Pembayaran

Tabel IV.3 Deskripsi *Use Case Diagram* Mengelola Data Transaksi Pembayaran

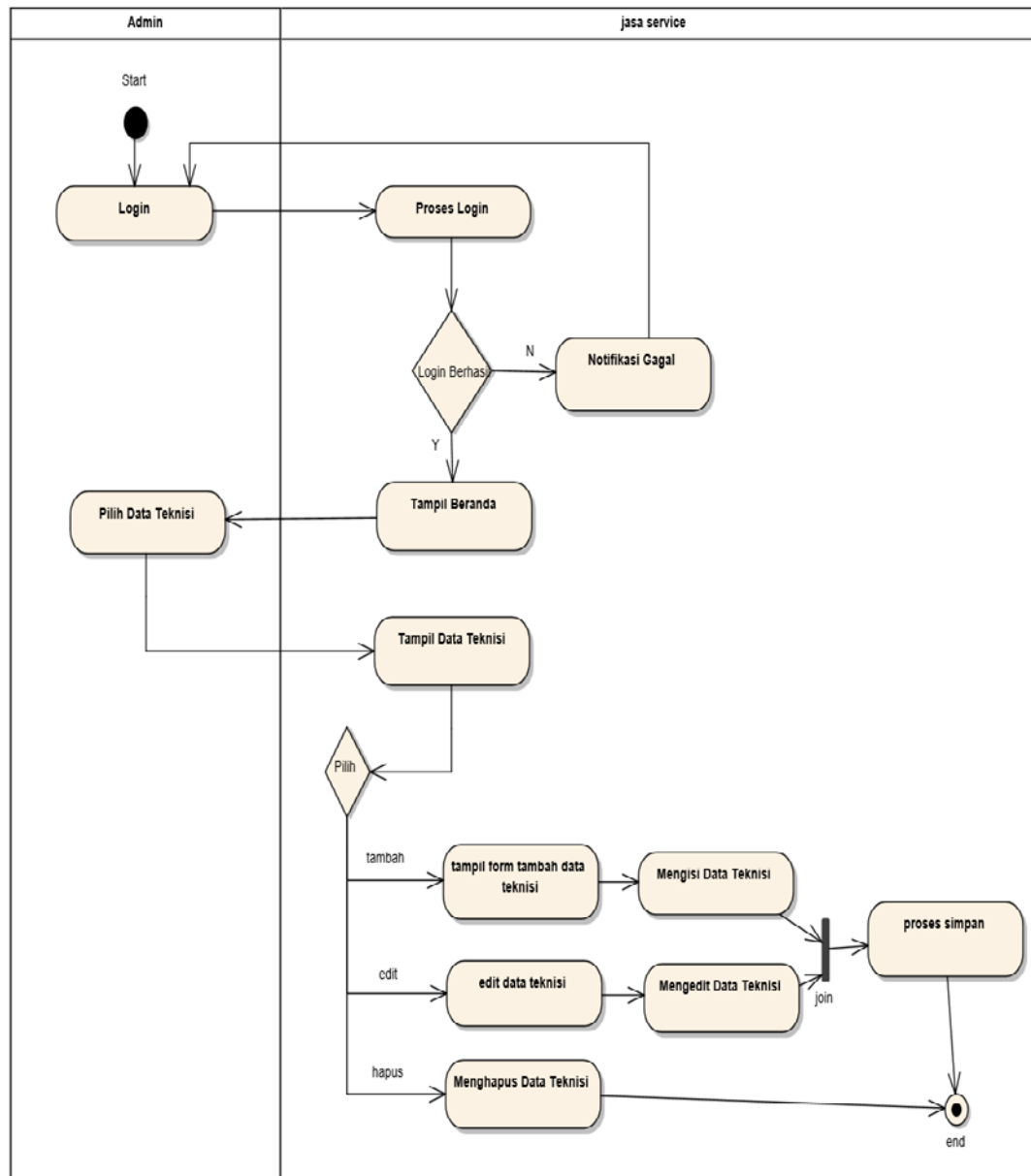
Use Case Name	Mengelola data transaksi Pembayaran
<i>Requirements</i>	B4
<i>Goal</i>	Admin dapat menambah data transaksi Pembayaran
<i>Pre-conditions</i>	Admin telah login
<i>Post-conditions</i>	Data pembayaran penjualan terupdate
<i>Failed end condition</i>	Gagal menyimpan data transaksi pembayaran
<i>Primary actors</i>	Admin
<i>Main Flow / Basic Path</i>	1. Admin melihat data pembayaran
<i>Invariant</i>	-2a. Admin mengedit data pembayaran

d. Deskripsi *Use Case* Mengelola Data Laporan

Tabel IV.4 Deskripsi *Use Case Diagram* mengelola data laporan

Use Case Name	Mengelola data laporan
<i>Requirements</i>	B6
<i>Goal</i>	Admin mengolah data laporan by system
<i>Pre-conditions</i>	Admin telah login
<i>Post-conditions</i>	Data laporan ter-update setiap ada pembayaran
<i>Failed end condition</i>	Data laporan error
<i>Primary actors</i>	Admin
<i>Main Flow / Basic Path</i>	Admin memilih data laporan per bulan
<i>Invariant</i>	-

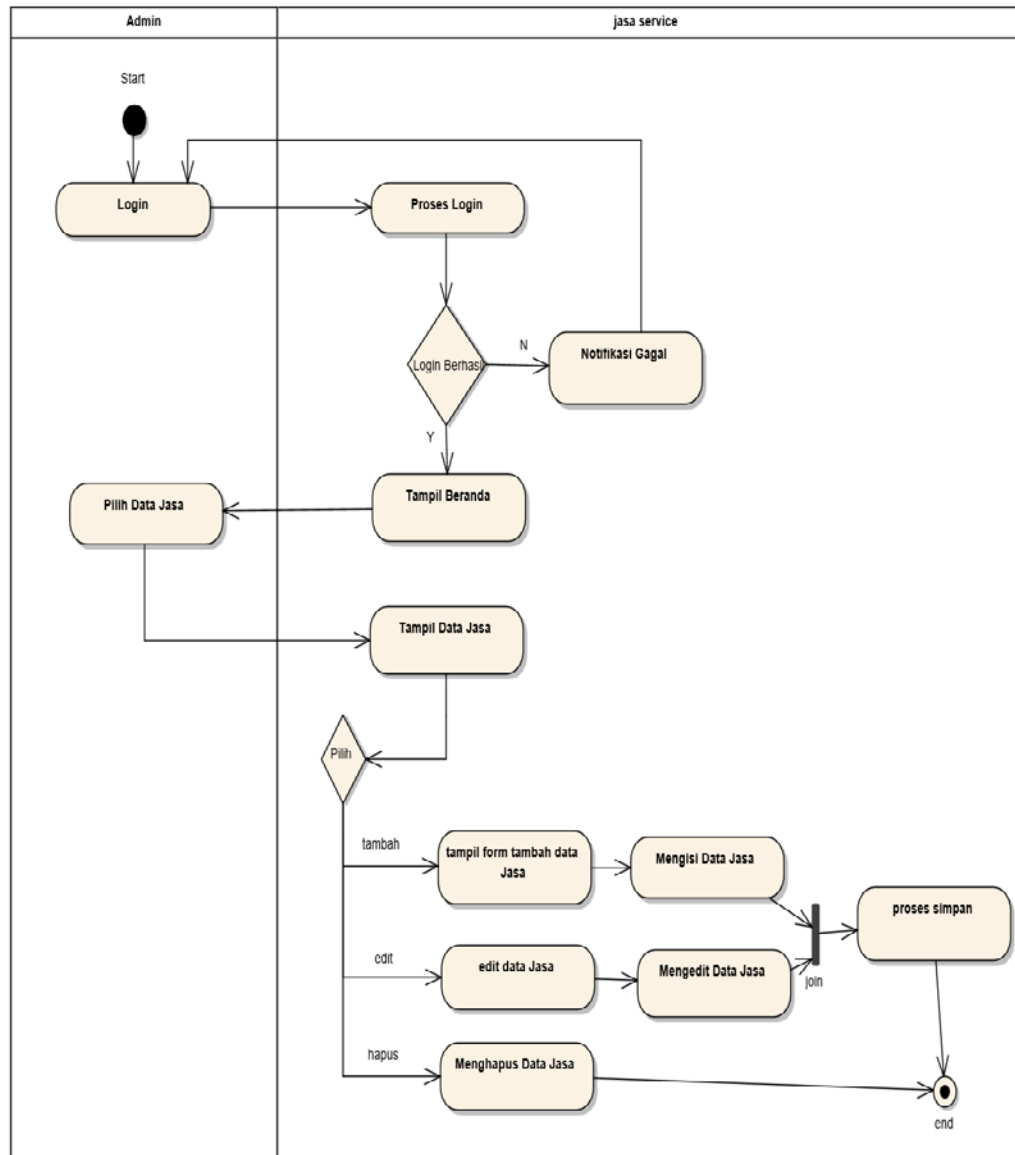
2. Activity Diagram admin mengelola data teknisi



Sumber : Bzs Komputer (2016)

Gambar IV.2 Activity Diagram admin mengelola data teknisi

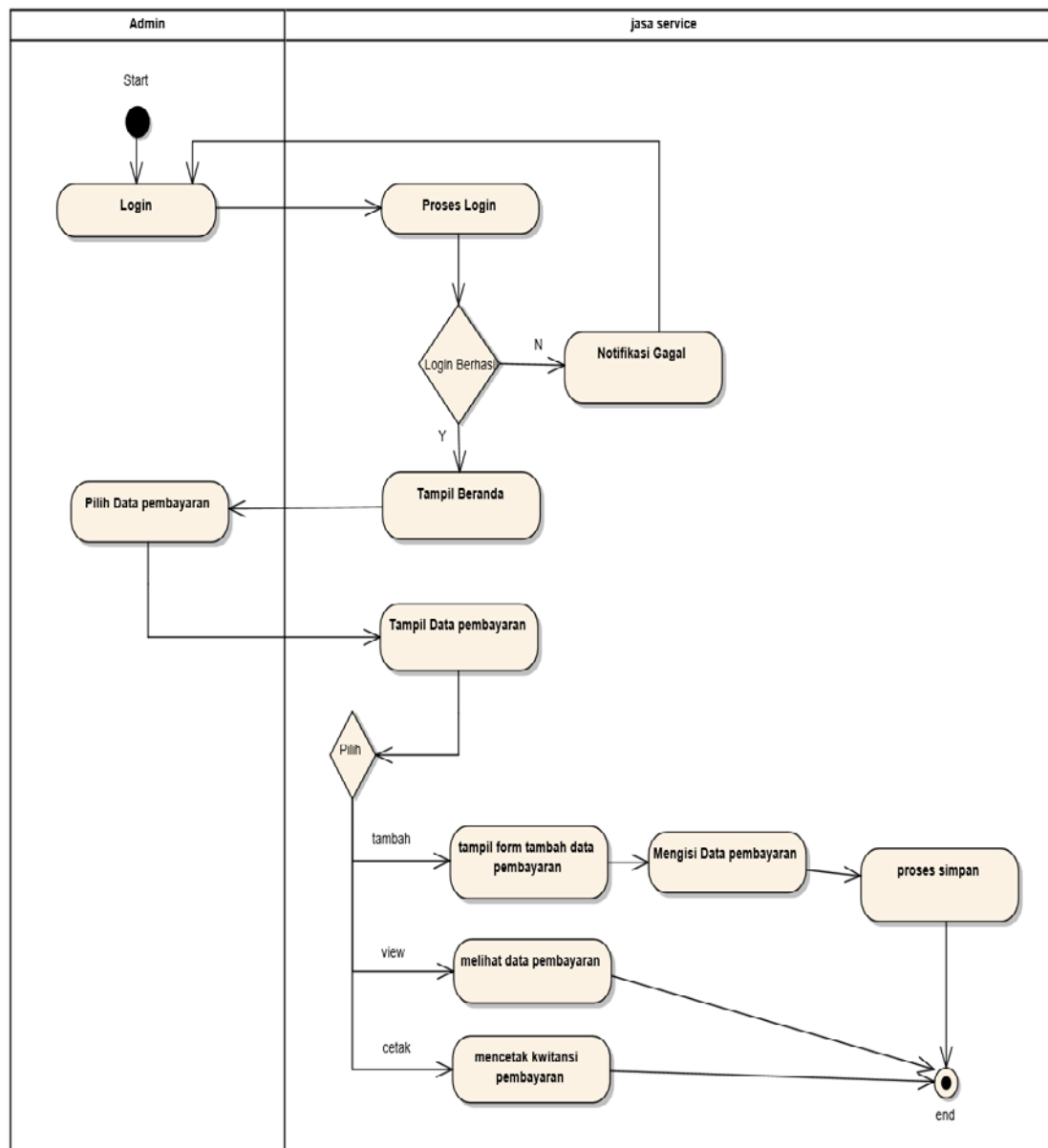
3. Activity Diagram admin mengelola data Jasa



Sumber : Bzs Komputer (2016)

Gambar IV.3 Activity Diagram admin mengelola data Jasa

4. *Activity Diagram* admin mengelola data pembayaran



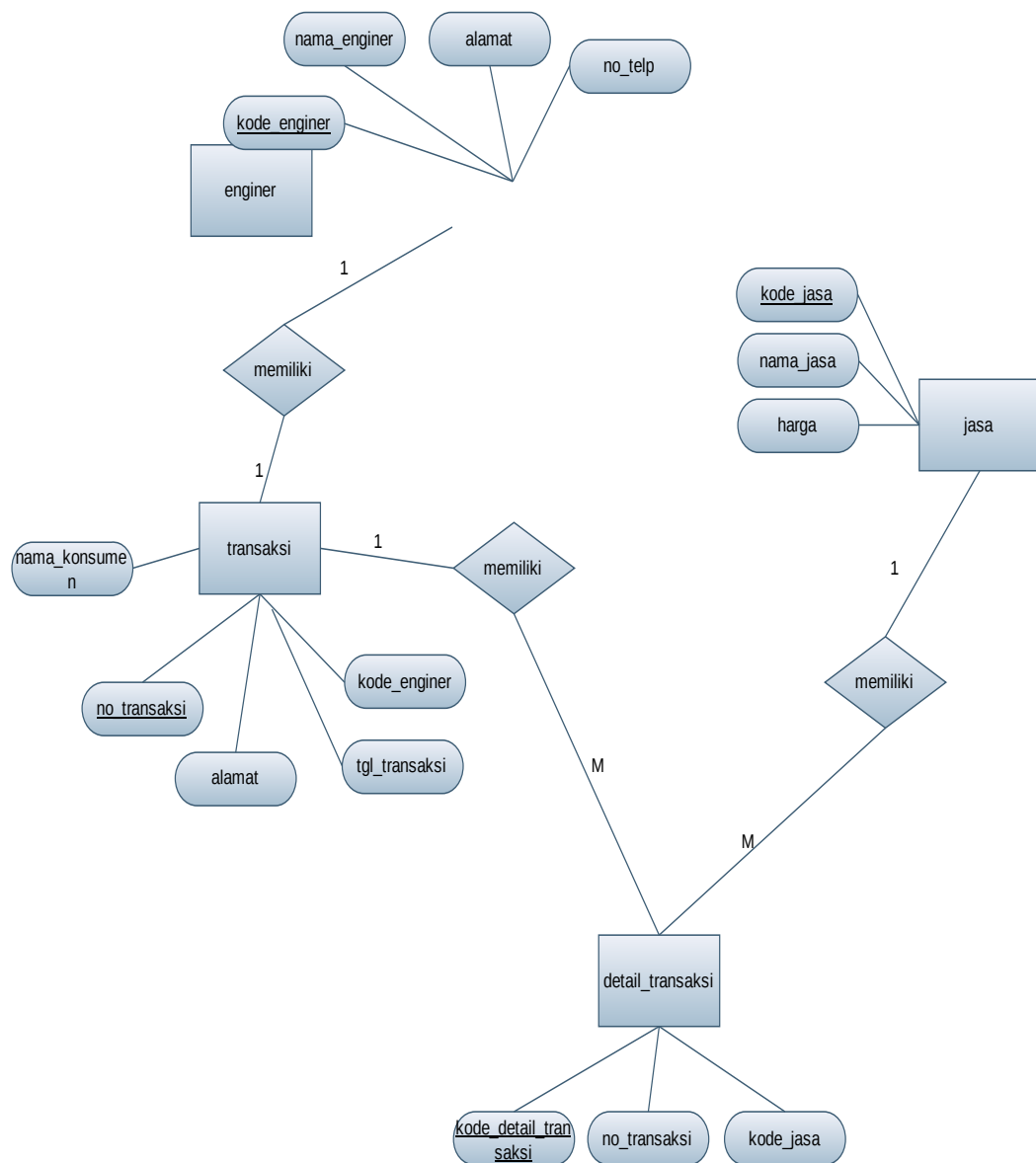
Sumber : Bzs Komputer (2016)

Gambar IV.4 *Activity Diagram* admin mengelola data Pembayaran

4.2 Desain

4.2.1. Database.

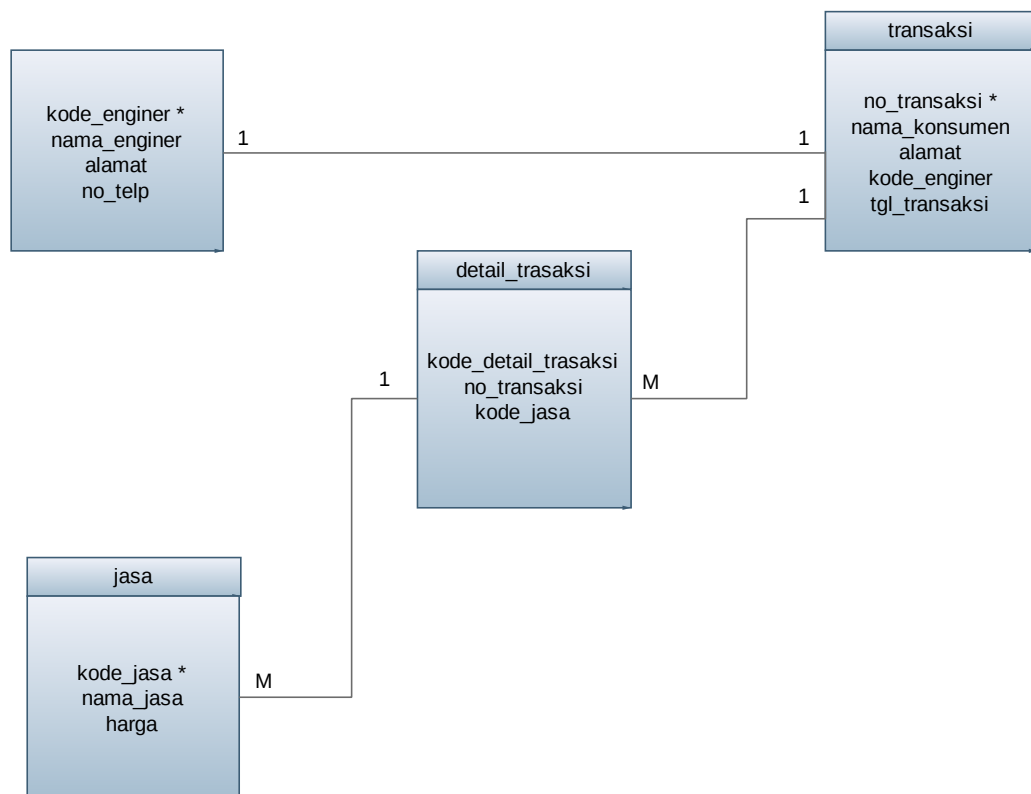
1. Entity Relationship Diagram.



Sumber : Bzs Komputer (2016)

Gambar IV.6 Entity Relationship Diagram

2. *LRS (Logical Record Structure).*



Sumber : Bzs Komputer (2016)

Gambar IV.7 *Logical Record Structure*

3. Spesifikasi File.

a. Spesifikasi file table wilayah.

Nama Database : aplikasi

Nama File : enginer

Akronim : enginer.myd

Tipe File : file master

Akses File : *index sequential*

Panjang Record : 67 karakter

Kunci Field : kode_enginer

Tabel IV.5 Spesifikasi File table Enginer

No	Elemen data	Nama field	Type	Size	Keterangan
1	Kode Enginer	Kode_enginer	Int	4	<i>Primary Key</i>
2	Nama Enginer	Nama_enginer	Varchar	50	
3	Alamat Enginer	Alamat	Text		
4	Nomor Telepon	No_telp	Varchar	13	

b. Spesifikasi file table jasa.

Nama Database : aplikasi

Nama File : jasa

Akronim : jasa.myd

Tipe File : file master

Akses File : *index sequential*

Panjang Record : 62 karakter

Kunci Field : kode_jasa

Tabel IV.6 Spesifikasi File table jasa

No	Elemen data	Nama field	Type	Size	Keterangan
1	Kode Jasa	kode_jasa	Int	4	<i>Primary Key</i>
2	Nama Jasa	nama_jasa	Varchar	50	
3	Harga	Harga	Int	8	

c. Spesifikasi file table transaksi.

Nama Database : aplikasi

Nama File : transaksi

Akronim : transaksi.myd

Tipe File : file transaksi

Akses File : *random*

Panjang Record : 63 karakter

Kunci Field : no_transaksi

Tabel IV.7 Spesifikasi File table transaksi

No	Elemen data	Nama field	Type	Size	Keterangan
1	Nomor Transaksi	no_transaksi	Varchar	18	<i>Primary Key</i>
2	Nama Konsumen	Nama_konsumen	Varchar	50	
3	Alamat Konsumen	Alamat	Text		
4	Kode Enginer	Kode_enginer	Varchar	5	

d. Spesifikasi file table detail_transaksi.

Nama Database : aplikasi

Nama File : detail_transaksi

Akronim : detail_transaksi.myd

Tipe File : file transaksi

Akses File : *random*

Panjang Record : 32 karakter

Kunci Field : kode_detail_transaksi

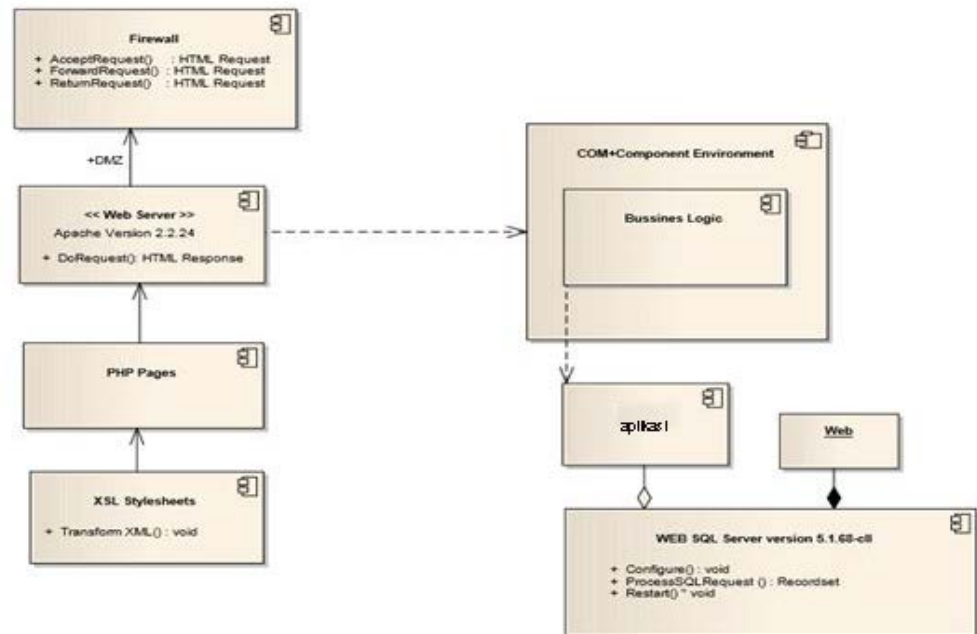
Tabel IV.8 Spesifikasi File table detail_transaksi

No	Elemen data	Nama field	Type	Size	Keterangan
1	Kode Detail Transaksi	kode_detail_transaksi	Int	4	<i>Primary Key</i>
2	Nomor Transaksi	no_transaksi	Varchar	18	<i>Foreign Key</i>
3	Kode Jasa	Kode_jasa	Varchar	10	<i>Foreign Key</i>

4.2.2. Software Architecture.

1. Component Diagram.

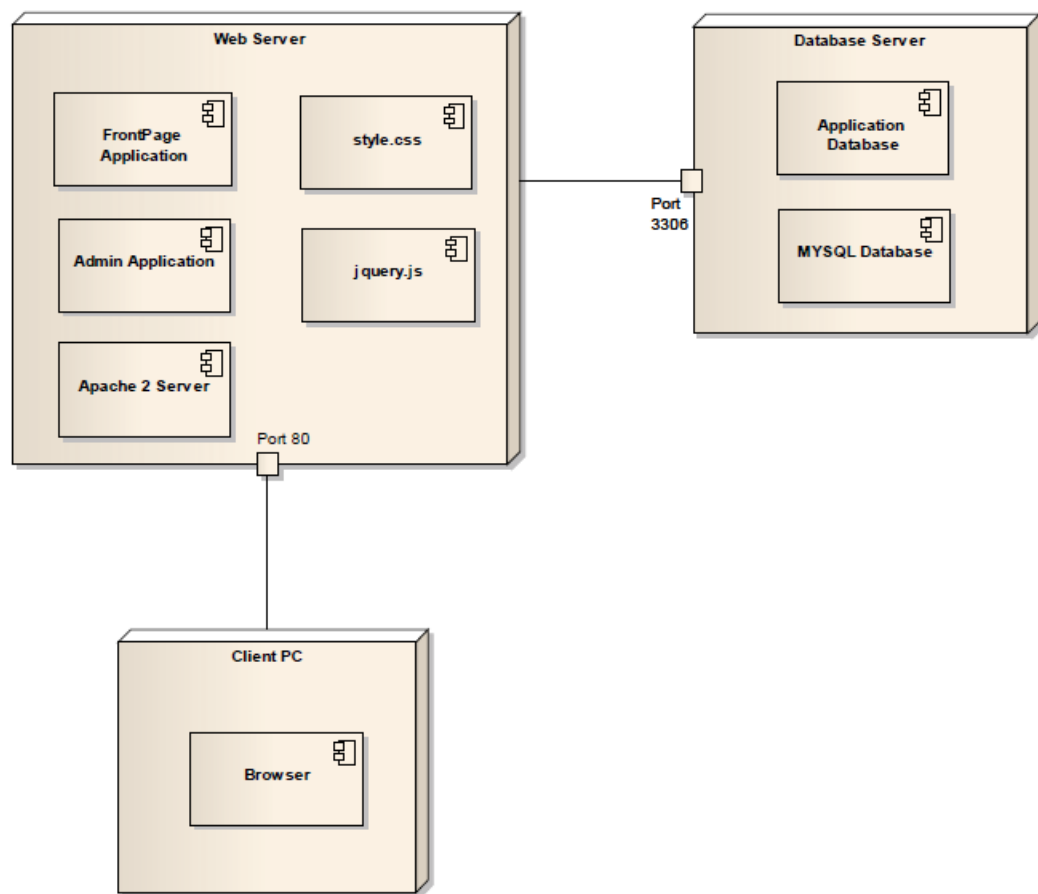
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan diantaranya.



Gambar IV.8 Component Diagram

2. Deployment diagram.

Diagram *Deployment* (*Deployment Diagram*) bersifat statis, diagram ini memperlihatkan konfigurasi saat aplikasi di jalankan (*run-time*), memuat simpul-simpul beserta komponen-komponen yang ada didalamnya.

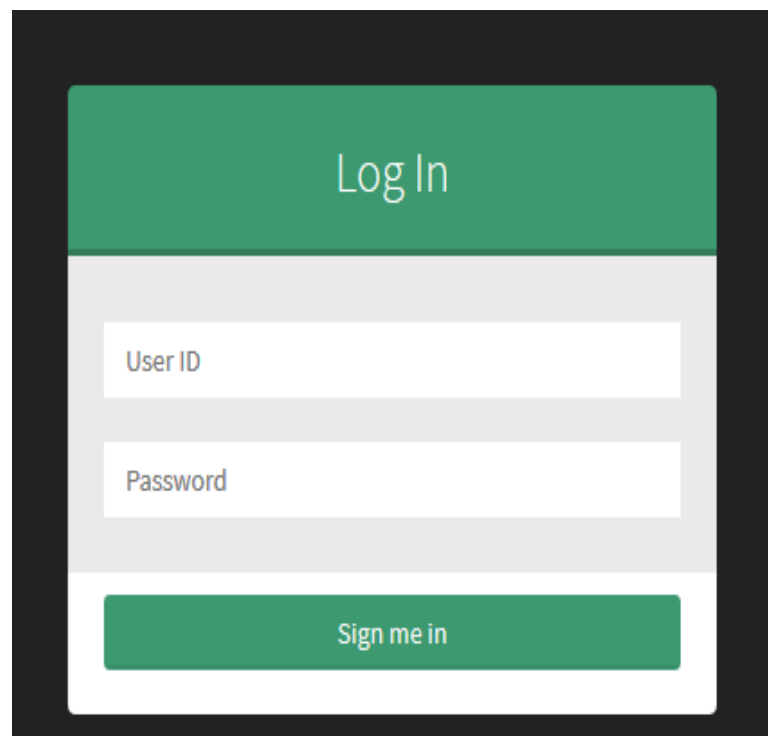


Gambar IV.9 *Deployment Diagram*

4.2.3. *User Interface.*

Antarmuka pemakai (*User Interface*) merupakan mekanisme komunikasi antara pengguna dengan sistem. Antarmuka pemakai (*User Interface*) dapat menerima informasi dari pengguna (*user*) dan memberikan informasi kepada pengguna (*user*) untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi.

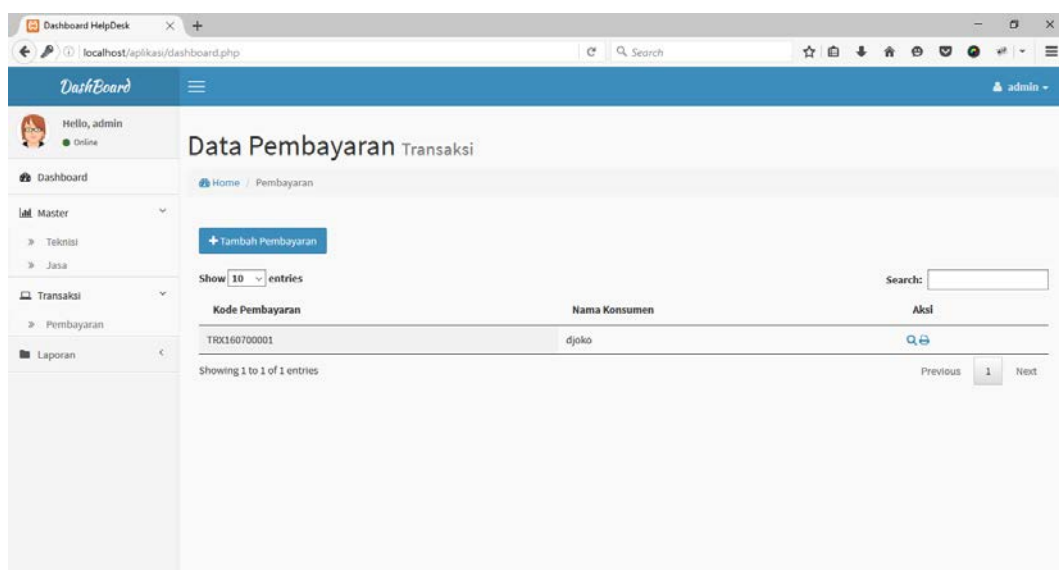
1. Halaman Login Admin.



The image shows a login page with a dark background. At the top, there is a green rectangular box with the text "Log In" in white. Below this, there are two white input fields: the first is labeled "User ID" and the second is labeled "Password". At the bottom of the form, there is a green rectangular button with the text "Sign me in" in white.

Gambar IV.10 Tampilan Halaman Login Admin

2. Halaman Transaksi Pembayaran.



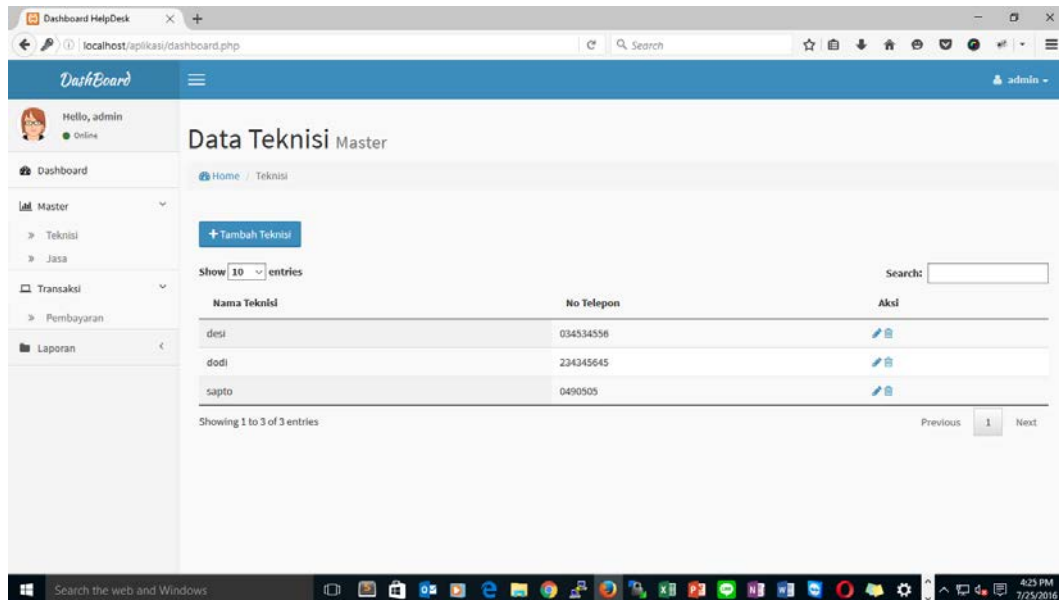
The image shows a web application interface for a payment transaction. The top navigation bar is blue and contains the text "Dashboard HelpDesk" and "admin". The left sidebar is white and contains a list of menu items: "Dashboard", "Master" (with sub-items "Teknisi" and "Jasa"), "Transaksi" (with sub-item "Pembayaran"), and "Laporan". The main content area is white and has the title "Data Pembayaran Transaksi". Below the title, there is a blue button labeled "+ Tambah Pembayaran". A search bar is located on the right side of the main content area. Below the search bar, there is a table with the following data:

Kode Pembayaran	Nama Konsumen	Aksi
TRX160700001	djoko	View

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom right, there are navigation buttons: "Previous", "1", and "Next".

Gambar IV.11 Tampilan Transaksi Pembayaran

3. Halaman Data Teknisi.



Gambar IV.12 Tampilan Teknisi

4.3. Code generation.

Berikut ini merupakan bahasa pemrograman terstruktur yang penulis gunakan untuk membuat aplikasi psikotest ini :

1. *Code generation* login admin.

```
<!DOCTYPE html>
<html class="bg-black">
  <head>
    <meta charset="UTF-8">
    <title>LogIn Helpdesk</title>
    <meta content='width=device-width, initial-scale=1, maximum-
scale=1, user-scalable=no' name='viewport'>
    <link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />
    <link href="css/font-awesome.min.css" rel="stylesheet"
type="text/css" />
    <!-- Theme style -->
    <link href="css/AdminLTE.css" rel="stylesheet" type="text/css" />
```

```

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements
    and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via
    file:// -->
    <!--[if lt IE 9]>
        <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
        <script
src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->
</head>
<body class="bg-black">

    <div class="form-box" id="login-box">
        <div class="header">Log In</div>
        <form action="loginprc.php" method="post">
            <div class="body bg-gray">
                <div class="form-group">
                    <input type="text" name="username" class="form-control"
placeholder="User ID"/>
                </div>
                <div class="form-group">
                    <input type="password" name="password" class="form-
control" placeholder="Password"/>
                </div>
            </div>
            <div class="footer">
                <button type="submit" class="btn bg-olive btn-block">Sign me
in</button>
            </div>
        </form>
    </div>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js" type="text/javascript"></script>

</body>
</html>

```

2. *Code generation* Transaksi Pembayaran

```

<script src="js/jquery.js"></script>
<script src="js/jquery-1.7.2.min.js"></script>
<link rel="stylesheet" href="css/jquery.dataTables.min.css"></style>
<script src="js/jquery.dataTables.min.js"></script>

<script type="text/javascript">
    $(document).ready(function(){

```

```

        $('#myTable').dataTable();
    });

    function hapusTiket(kd){
        $.post("module/tiket/actiontiket.php?action=hapus&kd=" + kd,
            function(data) {
                if (data == "OK") {
                    $('#isiKonten').load('module/tiket/index.php');
                }
                else {
                    alert(data)
                }
            });
        //$('#isiKonten').load('module/cabang/editCabang.php?kd='+kd);
    }
</script>

<section class='content-header'>
    <h1>
        Data Tiket
        <small>Transaksi</small>
    </h1>
    <ol class='breadcrumb'>
        <li><a href='#'><i class='fa fa-dashboard'></i> Home</a></li>
        <li class='active'>Tiket</li>
    </ol>
</section>

<!-- Main content -->
<section class='content'>
    <button class="btn btn-primary" onclick="TambahTiket()">
        <i class="fa fa-plus"></i>
        Tambah Pembayaran
    </button>
    <br><br>

    <div class="table-responsive">
        <table id="myTable" class="display table" width="100%" >
            <thead>
                <tr>
                    <th>Kode Pembayaran</th>
                    <th>Nama Konsumen</th>
                    <th>Aksi</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    include_once "../db.mysql.php";

```

```

include_once "../connectdb.php";
$s = "select * from tiket left join engine on
tiket.kode_engine=engine.kode_engine where status='1' order by
tiket.tgl_mulai asc";

$r = _query($s);
while($w = _fetch_array($r)){
    ?>
    <tr>
        <td><?php
$w['no_transaksi'];?></td>
        <td><?php
$w['nama_konsumen'];?></td>
        <td>
            <a
href="javascript:editTiket('<?php echo $w['no_tiket'];?>');"><i class="fa fa-pencil"
title='Edit'></i></a>
            <?php if ($w['status']=='1') {
?>
            <a
href="javascript:hapusTiket('<?php echo $w['no_tiket'];?>');"><i class="fa fa-
trash-o" title='Hapus'></i></a>
            <?php } ?>
            <a
href="javascript:editTiket('<?php echo $w['no_tiket'];?>');"><i class="fa fa-print"
title='Cetak'></i></a>
        </td>
    </tr>
    <?php
    }
    ?>
</tbody>
</table>
</div>
</section><!-- /.content -->

```

3. Code generation Teknisi

```

<script src="js/jquery.js"></script>
<script src="js/jquery-1.7.2.min.js"></script>
<link rel="stylesheet" href="css/jquery.dataTables.min.css"></style>
<script src="js/jquery.dataTables.min.js"></script>
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />

<script type="text/javascript">
    $(document).ready(function(){
        $('#myTable').dataTable();
    });

```

```

function hapusTiket(kd){
$.post("module/tiket/actiontiket.php?action=hapus&kd=" + kd,
function(data) {
if (data == "OK") {
$('#isiKonten').load('module/tiket/index.php');
}
else {
alert(data)
}
});
//$('#isiKonten').load('module/cabang/editCabang.php?kd='+kd);
}
</script>

<section class='content-header'>
<h1>
Data Berita Acara
<small>Transaksi</small>
</h1>
<ol class='breadcrumb'>
<li><a href='#'><i class='fa fa-dashboard'></i> Home</a></li>
<li class='active'>Berita Acara</li>
</ol>
</section>

<!-- Main content -->
<section class='content'>

<div class="table-responsive">
<table id="myTable" class="display table" width="100%" >
<thead>
<tr>
<th>Kode Tiket</th>
<th>Nama Engineer</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php
include_once "../db.mysql.php";
include_once "../connectdb.php";
$s = "select * from tiket left join engineer on
tiket.kode_engineer=engineer.kode_engineer order by tiket.tgl_mulai asc";
$r = _query($s);
while($w = _fetch_array($r)){
?>
<tr>

```


A. Form Login.

Tabel IV.9

Hasil Pengujian Black Box Testing Form Login

No	Scenario Pengujian	Test Case	Hasil Yang di harapkan	Hasil pengujian	Kesimpulan
1	Mengkosongkan semua isian data login, lalu langsung klik tombol login	Username : (kosong) Password : (kosong)	Sistem akan menolak akses login dan menampilkan pesan "anda belum mengisi username"	Sesuai Harapan	Valid
2	Hanya mengisi username dan mengosongkan passwod lalu langsung klik tombol login	Username : admin Password : (kosong)	Sistem akan menolak akses login dan menampilkan pesan "anda belum mengisi password"	Sesuai Harapan	Valid
3	Mengimputkan dengan kondisi salah satu data benar dan satu lagi salah lalu langsung klik tombol login	Username : admin (benar) Password : dodi (salah)	Sistem akan menolak akses login dan menampilkan pesan "Login gagal kombinasi username dan password salah"	Sesuai Harapan	Valid
4	Mengimputkan data benar lalu langsung klik tombol login	Username : admin (benar) Password : admin (benar)	Sistem menerima akses login kemudian langsung menampilkan tampilan home	Sesuai Harapan	Valid

B. Pengujian Terhadap *Form* Tambah Engineer

Tabel IV.10.

Hasil Pengujian *Black Box Testing Form* Tambah Engineer

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Salah satu <i>field</i> belum terisi kemudian klik daftar	Nama Engineer : (kosong) <i>Field</i> lain : terisi	Muncul peringatan "Nama Engineer Tidak boleh Kosong" dan kembali ke <i>form</i> tambah engineer	Sesuai harapan	Valid
2	Semua <i>field</i> terisi kemudian klik daftar	<i>Field</i> : terisi	Tersimpan kedalam database dan langsung ke halaman data engineer"	Sesuai harapan	Valid

C. Pengujian Terhadap *Form* Tambah Jasa

Tabel IV.11.
Hasil Pengujian *Black Box Testing Form* Tambah Jasa

No	Skenario Pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Salah satu <i>field</i> belum terisi kemudian klik daftar	Nama wilayah : (kosong) <i>Field</i> lain : terisi	Muncul peringatan “Nama Jasa Tidak boleh Kosong “ dan kembali ke <i>form</i> tambah produk	Sesuai harapan	Valid
2	Semua <i>field</i> terisi kemudian klik daftar	<i>Field</i> : terisi	Tersimpan kedalam database dan langsung ke halaman data jasa”	Sesuai harapan	Valid

4.5. Support.

Dalam membangun sebuah sistem yang nantinya akan digunakan pada *Personal Computer* (PC) spesifikasi *hardware* yang dibutuhkan adalah: *Prosesor Dual Core*, *Memori 1 GB DDR 2 RAM*, *Hardisk 80 GB HDD* dan *VGA Std* atau *onboard*. Sedangkan untuk *Operating System* berbasis *Windows*. Untuk *software* yang akan digunakan nanti dapat menggunakan *Macromedia Dreamweaver 8*, *Apache2Triad* dan *Web Browser* menggunakan *Mozilla Firefox* dan *Google Chrome*.

Dalam Pembuatan Aplikasi Web ini penulis menggunakan software sebagai berikut :

- Program : *Macromedia Dreamweaver*
- Database* : *Aphache2triad MySQL 5.0.18*
- Browser* : *Mozilla firefox* Versi 21.0
- Software plugin* : *Firebug*

Dan spesifikasi hardware untuk mendukung pembuatan aplikasi meliputi :

- a. *Processor* : Intel Core2Duo 2,10 GHz
- b. *Memory* : 2 Gb
- c. *Harddisk* : 250 Gb,
- d. *Keyboard* : Standar
- e. *Mouse* : *Optical mouse*

4.6. Spesifikasi *Hardware dan Software*.

Untuk Spesifikasi Hardware dan Software dalam melakukan pengaksesan web ini adalah sebagai berikut :

Tabel IV.12 Spesifikasi Hardware dan Software

Kebutuhan	Keterangan
Sistem operasi	Windows XP dan sesudahnya
<i>Processor</i>	Minimal Core 2 Duo 2.1 Ghz
RAM	2 GB
<i>Harddisk</i>	160 GB
Monitor	SVGA 14"
<i>Keyboard</i>	108 key
Printer	Dot matrix
<i>Mouse</i>	Standar
<i>Browser</i>	Mozilla Firefox, Internet Explorer, Google Chrome

4.7. Spesifikasi Dokumen Sistem Usulan.

a. Dokumen Masukan

1. Nama Dokumen: Kwitansi Pembayaran

Fungsi : Untuk Bukti Pembayaran

Sumber : Admin

Tujuan : Konsumen

Media : Kertas
Jumlah : 1 lembar
Frekuensi : Setiap ada Pembayaran Baru
Bentuk : Lihat lampiran B.1

B. Output

1. Nama Dokumen: Laporan

Fungsi : Untuk Laporan ke pemilik perusahaan
Sumber : Admin
Tujuan : Owner
Media : Kertas
Jumlah : 1 lembar
Frekuensi : Setiap Bulan
Bentuk : Lihat lampiran B.2