

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. Perancangan Sistem**

Perancangan sistem dibuat sebagai tahapan untuk mempersiapkan proses implementasi sistem, dan untuk menggambarkan secara jelas proses-proses yang diinginkan oleh pemakai (*user*). Perancangan sistem memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang terdapat di Klinik Padjadjaran yang beralamat di Jl. PH.H Mustofa No. 28 dalam menunjang aktivitas pelayanan.

##### **4.1.1. Tujuan Perancangan Sistem**

Tujuan dari perancangan sistem ini adalah untuk merancang sistem yang diusulkan setelah melewati proses analisis dan evaluasi permasalahan dari sistem yang sedang berjalan, sehingga sistem yang diusulkan dapat mengatasi berbagai masalah yang ada pada sistem yang sedang berjalan.

Tujuan perancangan sistem yang diusulkan adalah sebagai berikut:

1. Sistem Informasi Pelayanan pada Klinik Padjadjaran, ditujukan untuk mempercepat proses-proses yang menyangkut pengolahan data agar

terkomputerisasi serta meningkatkan ketepatan, kecepatan, dan keakuratan informasi yang dibutuhkan.

2. Dengan adanya Aplikasi Pelayanan ini diharapkan dapat meningkatkan pelayanan terhadap pasien yang berkunjung dan memberikan kemudahan dalam proses pengolahan dan pelaporan data pada Klinik Padjadjaran.
3. Membangun aplikasi berbasis web yang terintegrasi dengan sistem database, yang dapat mempublikasikan dan mendokumentasikan data pasien, data dokter, data petugas baik poliklinik maupun pendaftaran dan data obat.
4. Untuk memberikan fasilitas terhadap petugas pelayanan dan juga dokter agar mudah melihat atau membaca data perkembangan dari pasien, sehingga dapat dijadikan alat bantu untuk pengambilan keputusan.

#### **4.1.2. Gambaran Umum Sistem yang Diusulkan**

Gambaran umum sistem yang diusulkan adalah berupa sistem aplikasi pelayanan berbasis web, yang dapat dijadikan sebagai sarana penunjang sistem informasi pelayanan yang sedang berjalan di Klinik Padjadjaran. Dalam sistem aplikasi pelayanan ini terdapat proses untuk menghasilkan informasi pelayanan diantaranya yaitu informasi pelayanan, dokter, pasien, petugas, biaya medis, serta tindakan yang dilakukan oleh dokter.

Dalam penggunaannya, sistem aplikasi pelayanan digunakan oleh lima kategori pengguna (*user*), yaitu admin, pasien, petugas pendaftaran, petugas poliklinik dan

dokter. Sistem aplikasi pelayanan ini selalu dapat diakses dari berbagai tempat dan kapan saja dengan menggunakan komputer yang terhubung dengan jaringan internet.

Sistem aplikasi ini dinamakan Sistem Aplikasi Pelayanan Berbasis Web. Adapun proses-proses yang dapat dilakukan oleh sistem aplikasi ini adalah sebagai berikut :

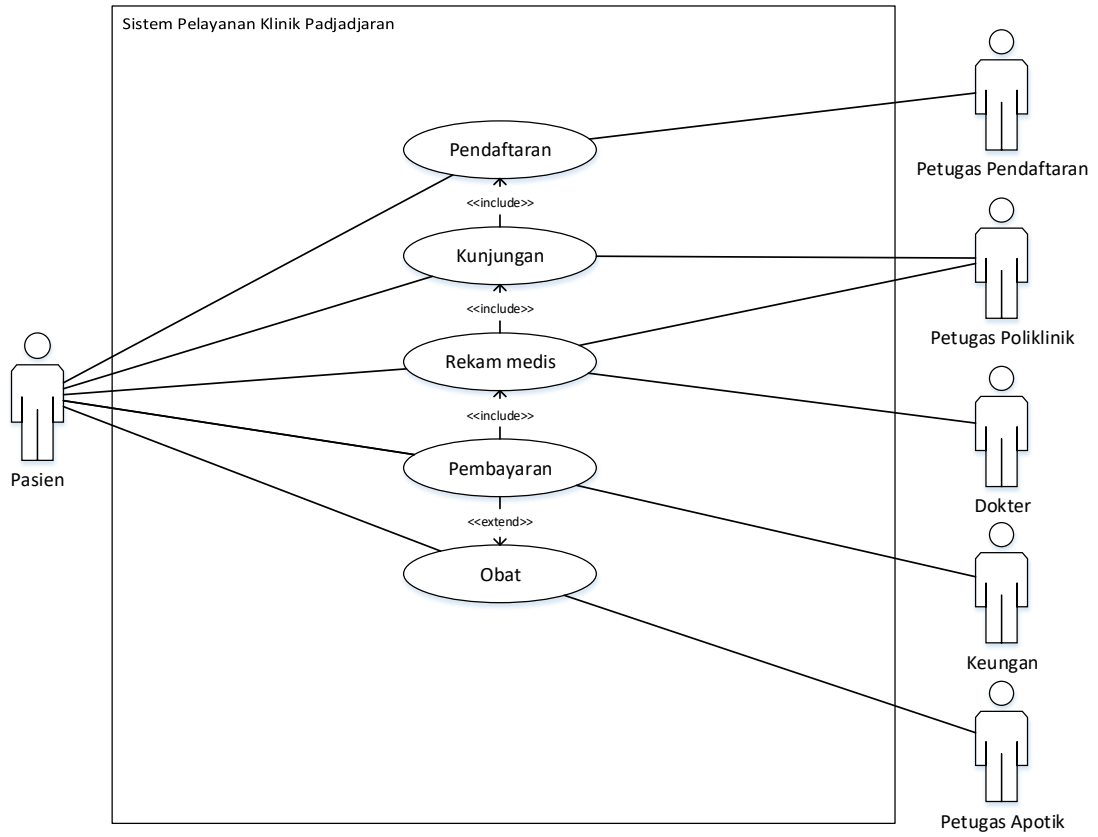
1. Pasien dapat menggunakan fasilitas yang ada di dalam sistem aplikasi ini baik untuk pendaftaran maupun untuk melihat history pelayanan yang telah dilalui oleh pasien yang bersangkutan.
2. Petugas pendaftaran dapat mencetak kartu pelayanan terhadap pasien yang telah melakukan pendaftaran.
3. Petugas pendaftaran dapat menginput nota pemeriksaan sesuai poliklinik yang dituju oleh pasien dan dapat mencetak langsung kartu berobat.
4. Petugas poliklinik dapat melihat atau mengecek kartu berobat untuk menyamakan dengan yang ada di dalam database dimana telah di input oleh petugas pendaftaran sebelumnya
5. Setelah petugas poliklinik melakukan pemeriksaan awal, petugas poliklinik dapat mencetak nota pemeriksaan dan kartu pelayanan.
6. Setelah dokter melakukan pemeriksaan dokter menginput nota pemeriksaan dan dapat langsung mencetak resep sesuai jenis penyakit yang diderita oleh pasien.
7. Admin dapat membuat rekapitulasi berupa laporan dari semua kegiatan yang terjadi didalam aplikasi pelayanan ini.
8. Admin dapat melakukan proses *update* (tambah, ubah, hapus) terhadap data-data master didalam *database*.

### **4.1.3. Perancangan Prosedur yang Diusulkan**

Perancangan prosedur yang diusulkan bertujuan untuk menentukan kebutuhan dari sistem yang diusulkan atau dirancang. Sesuai dengan metode pendekatan sistem yang digunakan, maka penggambaran atau pemodelan sistem yang diusulkan akan dipresentasikan menggunakan notasi UML (Unified Modeling Language), meliputi: aktor, diagram use case, skenario use case, diagram aktivitas, diagram sekuensial, diagram kelas, diagram objek, diagram komponen, diagram *deployment*.

#### **4.1.3.1. Use Case diagram**

Diagram *use case* atau *use case* diagram menyajikan interaksi antara *use case* dan aktor. Permodelan ini dimaksudkan untuk menggambarkan proses Admin, Pasien, Petugas Pendaftaran, Petugas Poliklinik dan Dokter, proses dan hubungan yang terjadi antara aktor dan *use case* di dalam sistem yang diusulkan. Diagram *use case* sistem yang diusulkan dapat dilihat pada gambar di bawah ini.



### Gambar 4.1 Diagram *Use Case* yang Diusulkan

#### a. Definisi aktor dan deskripsinya

Aktor adalah orang yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Simbol aktor digambarkan dengan simbol orang. Aktor belum tentu merupakan orang, biasanya dinyatakan dengan menggunakan kata benda diawal frase nama aktor.

**Tabel 4.1 Definisi Aktor dan Deskripsinya**

No.	Aktor	Deskripsi
1	Pasien	Pihak yang mendapatkan perawatan setelah melakukan pendaftaran.
2	Petugas Pendaftaran	Pihak yang bertugas mendaftarkan biodata serta membimbing pasien dalam mengikuti proses pelayanan di klinik padjadjaran, dari pendaftaran sampai proses pembayaran.

3	Petugas Poliklinik	Pihak yang melakukan pemeriksaan awal terhadap pasien untuk menentukan Poli sesuai dengan jenis penyakit pasien.
4	Dokter	Pihak yang diberi kepercayaan penuh untuk melakukan pengobatan terhadap pasien yang telah mendaftar atau mengikuti proses pelayanan.
5	Keuangan	Pihak yang menghitung semua biaya administrasi dari Pendaftaran, biaya pelayanan sampai biaya obat.
6	Petugas Apotik	Pihak yang memberikan obat sesuai resep yang diberikan Dokter.
7	Admin	Pihak yang dapat mengelolah semua data yang terdapat didalam aplikasi pelayanan, serta pihak yang membuat laporan.

### **b. Definisi *Use Case* dan Deskripsinya**

*Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor. Fungsionalitas ini biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case*. *Use case* mendeskripsikan apa yang akan sistem kerjakan tapi tidak memberi spesifikasi dan tidak memiliki kompetensi bagaimana dia bisa melakukannya.

**Tabel 4.2 Definisi *Use Case* dan Deskripsinya**

No	<i>Use Case</i>	Deskripsi
1.	Pendaftaran	Mendaftarkan biodata pasien kedalam sistem informasi pelayanan.
2.	Kunjungan	Pencatatan biodata pasien disaat pasien datang lebih dari satu kali dengan syarat pasien harus terdaftar terlebih dahulu.
3.	Rekam medis	Sebagai dokumentasi untuk catatan dokter dan pasien disaat pasien menjalani pengobatan.
4.	Pembayaran	Proses pembayaran semua administrasi yang dikenakan terhadap pasien.
5.	Obat	Proses untuk menebus obat di apotik sesuai resep yang diberikan oleh dokter.
6.	Laporan	Sebagai tempat untuk merekam atau mendokumentasikan semua aktifitas yang telah terjadi didalam aplikasi pelayanan.



#### 4.1.3.2. Skenario *Use Case*

Skenario *use case* bertujuan untuk mendeskripsikan atau menjelaskan diagram *use case*. Berikut adalah skenario *use case* dari diagram *use case* yang diusulkan.

**Tabel 4.5 Skenario *Use Case* Pendaftaran**

<b>Identifikasi</b>	
Nama	Pendaftaran
Tujuan	Mendaftarkan biodata pasien kedalam sistem informasi pelayanan.
<b>Deskripsi</b>	
Tipe	Aktivitas utama
Aktor	Petugas Pendaftaran, Pasien
<b>Kondisi awal</b>	
Pasien belum terdaftar di dalm system informasi pelayanan.	
<b>Main Flow</b>	
<b>Aksi Actor Pasien</b>	<b>Reaksi Sistem</b>
1. Menyerahkan kartu identitas kepada petugas pendaftaran.	
<b>Aksi Actor Petugas Pendaftaran</b>	
1. Menerima kartu identitas pasien	
2. Membuka aplikasi pelayanan.	
3. Memilih menu pasien.	
	4. Menampilkan form input pasien
5. Menginput data pasien sesuai kartu identitas yang diterima.	
	6. Menyimpan data pasien kadalam database.
7. Memilih menu pendaftaran	

	8. Menampilkan menu pendaftaran untuk pasien
9. Memilih kode pasien dan menentukan poliklinik serta dokter yang akan menangani pasien.	
10. Setelah mengisi form pendaftaran dengan lengkap petugas pendaftaran mensubmit atau memilih tombol daftar	
	11. Menyimpan data pasien kedalam <i>database</i> pendaftaran.
<b>Kondisi Akhir</b>	
Pasien telah terdaftar kedalam aplikasi pelayanan.	

**Tabel 4.6 Skenario Use Case Kunjungan**

<b>Identifikasi</b>	
Nama	Kunjungan
Tujuan	Pencatatan biodata pasien disaat pasien datang lebih dari satu kali dengan syarat pasien harus terdaftar terlebih dahulu.
<b>Deskripsi</b>	
Tipe	Umum
Aktor	Pasien, Petugas Pendaftaran
<b>Kondisi awal</b>	
Pasien belum berkunjung	
<b>Main Flow</b>	
<b>Aksi Actor Pasien</b>	<b>Reaksi Sistem</b>
1. Pasien datang kerumah sakti dan membawa kartu pelayanan untuk diberikan kepada petugas poliklinik	
<b>Aksi Petugas Poliklinik</b>	

1. Petugas poliklinik meminta kartu pelayanan kepada pasien.	
2. Petugas poliklinik memilih menu kunjungan didalam aplikasi pelayanan.	
	3. Menampilkan form kunjungan.
4. Menginput nama pasien sesuai kode pasien sebagai tanda bahwa pasien telah berkunjung, serta menentukan poliklinik dan dokter yang menangani pasien yang berkunjung.	
	Menyimpan data kunjungan kedalam <i>database</i> .
<b>Kondisi Akhir</b>	
Pasien telah berkunjung dan data kunjungan tersimpan didalam <i>database</i> kunjungan.	

Tabel 4.7 Skenario *Use Case* Rekam Medis

<b>Identifikasi</b>	
Nama	Rekam Medis
Tujuan	Sebagai dokumentasi untuk catatan dokter dan pasien disaat pasien menjalani pengobatan.
<b>Deskripsi</b>	
Tipe	Utama
Aktor	Dokter, Petugas Poliklinik, Pasien
<b>Kondisi awal</b>	

Perkembangan keluhan pasien belum diketahui	
Main Flow	
Aksi Actor Pasien	Reaksi Sistem
1. Berkunjung ke poliklinik dengan membawa kartu pelayanan.	
Aksi Petugas Poliklinik	
2. Petugas Poliklinik meminta kartu pelayanan dan melakukan pemeriksaan awal sebelum pasien diserahkan kepada dokter yang bersangkutan.	
3. Petugas Poliklinik menyerahkan pasien kepada dokter yang bersangkutan	
Aksi Actor Dokter	
4. Memeriksa ulang kondisi pasien	
5. Dokter melakukan pengobatan terhadap pasien dan mencatat keluhan pasien serta tindakan apa yang telah diberikan terhadap pasien dengan cara membuka aplikasi pelayanan dan memilih menu pelayanan.	
	6. Menampilkan form pelayanan.
7. Menginput kode pasien, keluhan serta tindakan yang dilakukan dokter terhadap pasien	
8. Setelah menginput semua data yang diminta oleh sistem, dokter menekan tombol simpan atau submit.	
	9. Menyimpan data pelayanan kedalam database.
Kondisi Akhir	
Keluhan pasien telah diketahui dan telah diberi tindakan oleh dokter.	

Tabel 4.8 Skenario *Use Case* Pembayaran

<b>Identifikasi</b>	
Nama	Pembayaran
Tujuan	Proses pembayaran semua administrasi yang dikenakan terhadap pasien.
<b>Deskripsi</b>	
Tipe	Aktivitas Utama
Aktor	Pasien, Petugas Keuangan
<b>Kondisi awal</b>	
Pasien belum membayar biaya administrasi	
<b>Main Flow</b>	
<b>Aksi Actor Pasien</b>	<b>Reaksi Sistem</b>
1. Datang ke bagian keuangan dengan membawa kartu pelayanan.	
<b>Aksi Aktor Keuangan</b>	
2. Petugas keuangan membuka system informasi pelayanan.	
3. Memilih menu pembayaran	
	4. Menampilkan form pembayaran
5. Melihat semua biaya administrasi pasien dengan cara menginput kode pasien.	
	6. Menampilkan detail biaya yang harus dibayar oleh pasien.
7. Bagian keuangan menginput jumlah yang dibayar pasien sesuai dengan biaya detail yang ditampilkan oleh sistem.	
	8. Menyimpan transaksi pembayaran ke dalam database.
	9. Menampilkan kuitansi pembayaran.
10. Bagian keuangan mencetak kuitansi pembayaran dan di berikan kepada pasien.	

<b>Kondisi Akhir</b>
Semua biaya administrasi telah dibayar oleh pasien

Tabel 4.9 Skenario *Use Case* Obat

<b>Identifikasi</b>	
Nama	Obat
Tujuan	Proses untuk menebus obat di apotik sesuai resep yang diberikan oleh dokter.
<b>Deskripsi</b>	
Tipe	Umum
Aktor	Pasien, Petugas Apotik
<b>Kondisi awal</b>	
Pasien belum mendapatkan obat	
<b>Main Flow</b>	
<b>Aksi Aktor Pasien</b>	<b>Aksi Actor Petugas Apotik</b>
1. Pasien mengunjungi apotik dengan membawa resep yang diberikan oleh dokter	
2. Pasien memberikan resep kepada petugas apotik.	
	3. Petugas apotik menerima resep.
	4. Petugas apotik mengecek resep yang diterima dari pasien apakah pembayaran valid atau tidak. A1: Pembayaran valid A2: Pembayaran tidak valid.
	5. Petugas apotik memberikan obat kepada pasien sesuai resep yang diterima dari pasien
6. Pasien menerima obat dari apotik sesuai resep yang dibawa.	

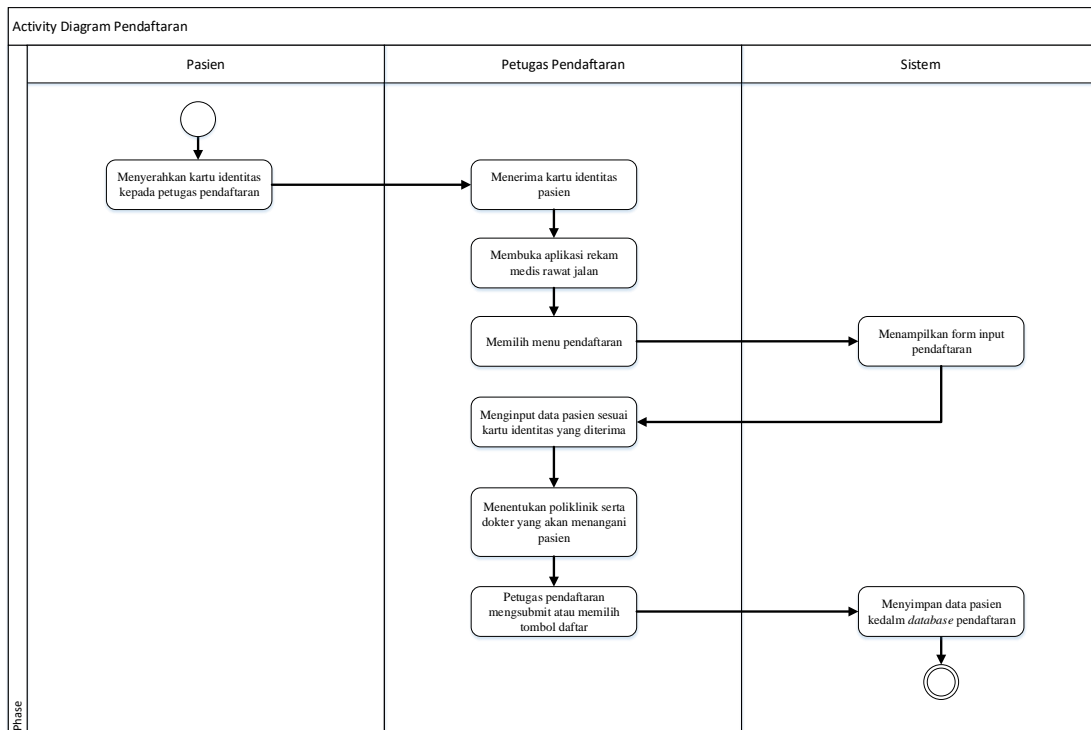
Aliran Alternatif 1 (A1)	
	1. Aliran lanjut ke skenario obat no. 5
Aliran Alternatif 2 (A2)	
	1. Memberikan pesan kepada pasien bahwa resep tidak valid.
2. Mengkonfirmasi pesan	
	3. Aliran kembali ke skenario pembayaran langkah ke 1.
<b>Kondisi Akhir</b>	
Pasien telah menerima obat dari apotik	

#### 4.1.3.3 Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan aliran fungsionalitas sistem. Dalam diagram ini akan digambarkan berbagai aliran aktivitas dalam sistem, yang bertujuan untuk mengetahui alur <sup>proses</sup> pada sistem yang diusulkan.

Berikut adalah diagram aktivitas yang mengacu pada setiap skenario use case yang sudah dibuat sebelumnya :

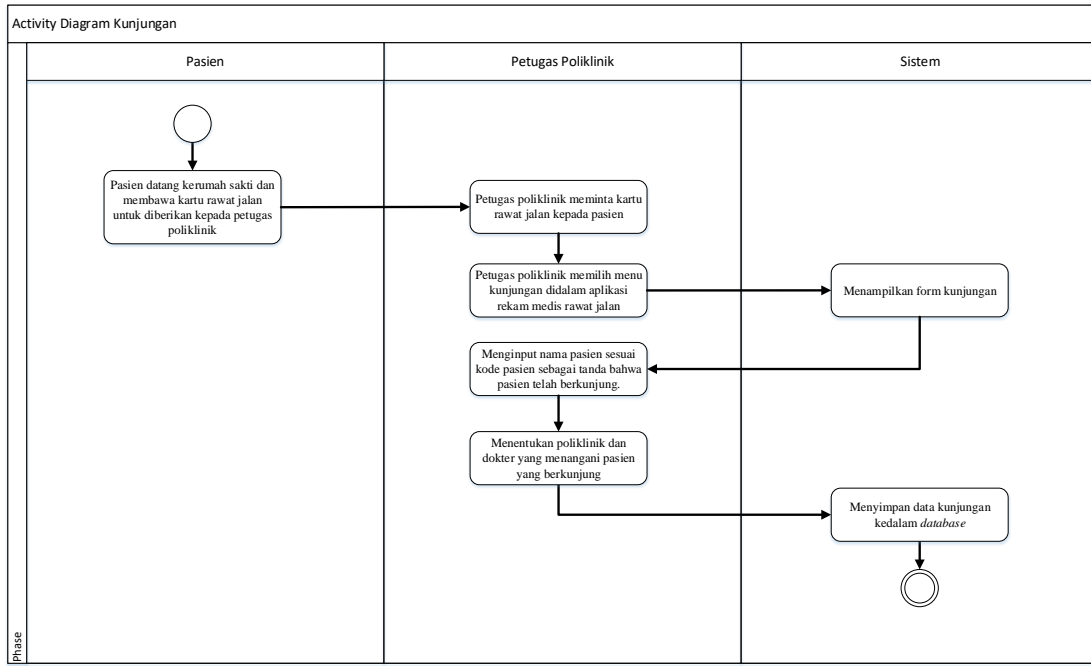
## 1. Activity Diagram Pendaftaran



Gambar 4.4 Activity Diagram Pendaftaran

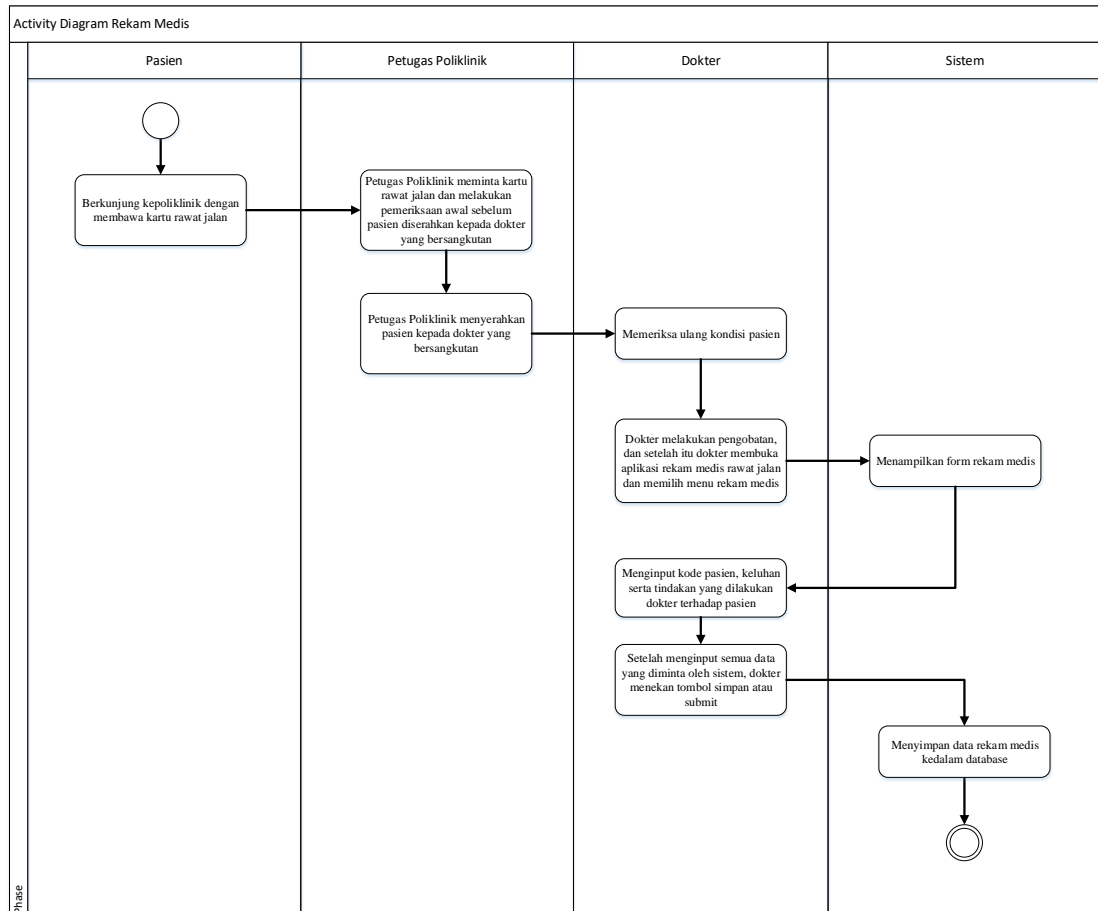


## 2. Activity Diagram Kunjungan



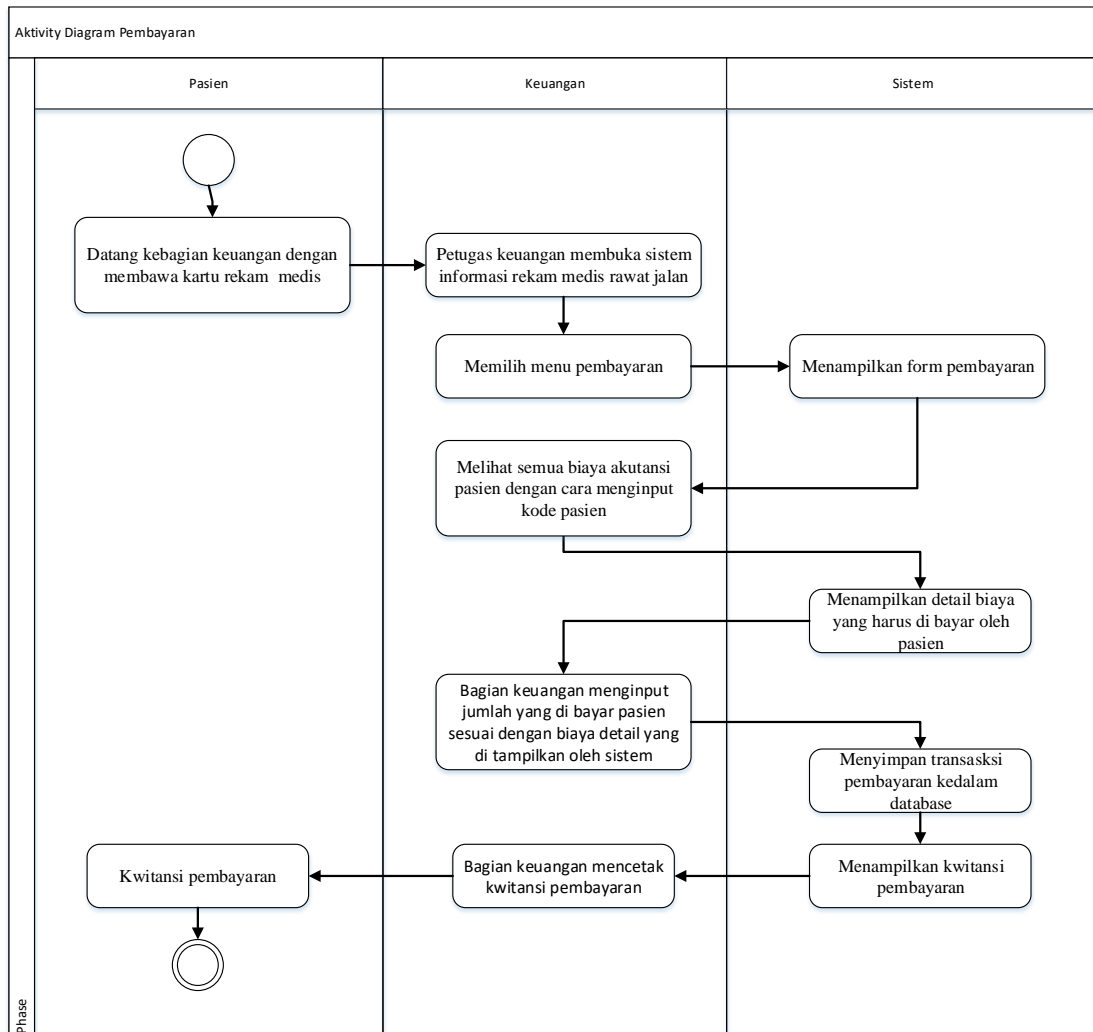
Gambar 4.5 Activity Diagram Kunjungan

### 3. Activity Diagram Rekam Medis



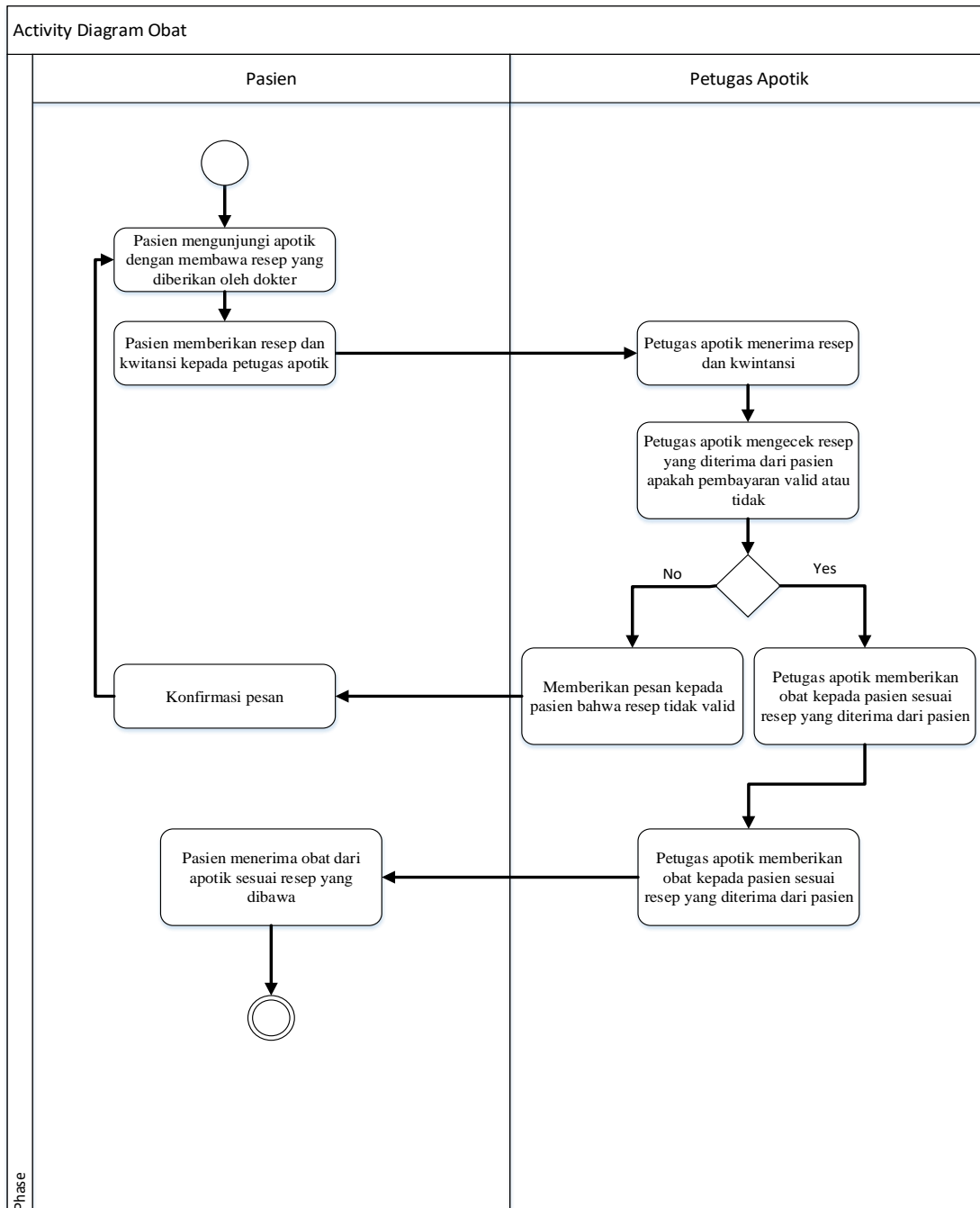
**Gambar 4.6 Activity Diagram Rekam Medis**

#### 4. Activity Diagram Pembayaran



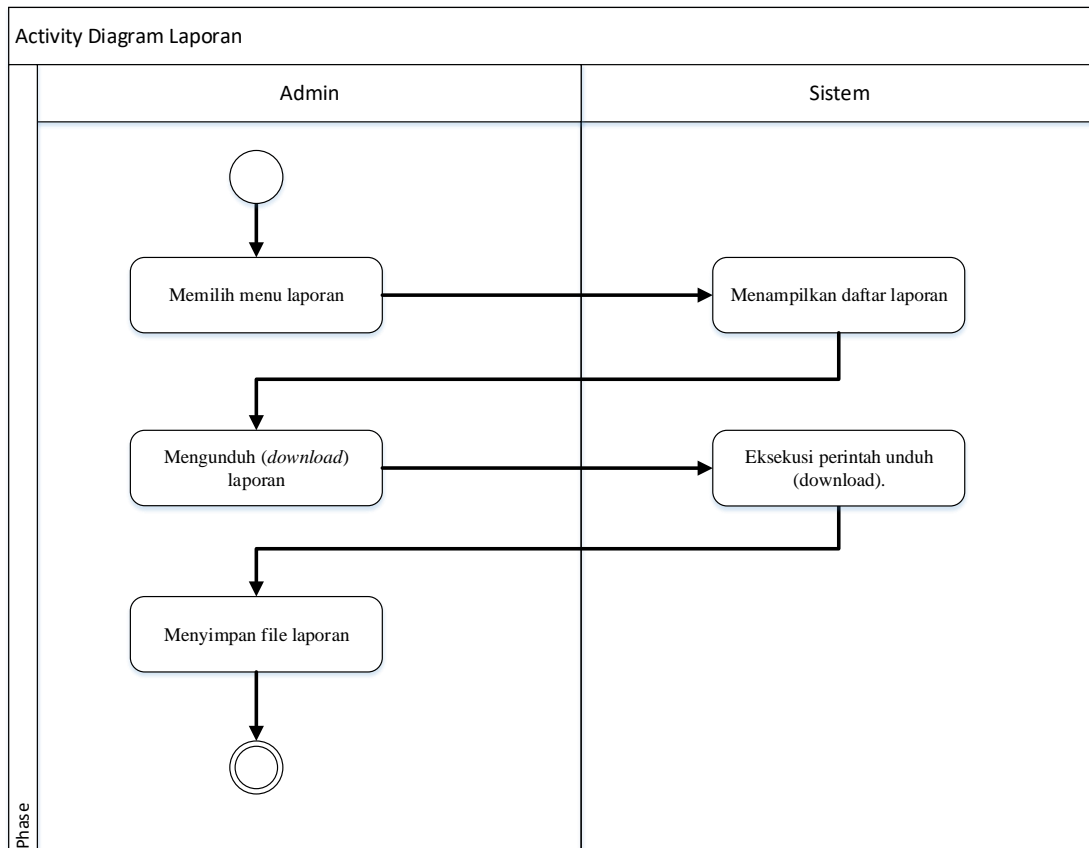
Gambar 4.7 Activity Diagram Pembayaran

## 5. Activity Diagram Obat



Gambar 4.8 Activity Diagram Obat

## 6. Activity Diagram Laporan



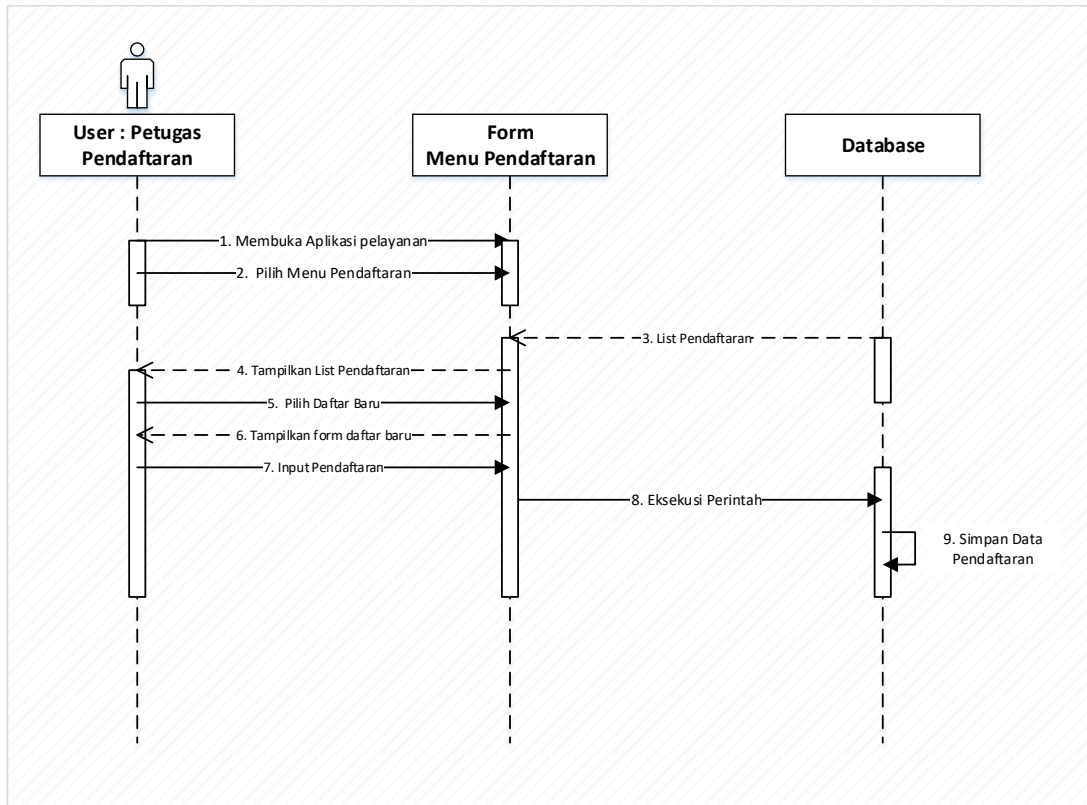
**Gambar 4.9** Activity Diagram Laporan

### 4.1.3.4 Sequence Diagram

Diagram sekuensial atau sequence diagram digunakan untuk menunjukkan aliran fungsionalitas dalam use case. Sequence diagram digunakan untuk memberikan gambaran detail dari setiap use case diagram yang telah dibuat sebelumnya. Setiap objects yang terlibat dalam sebuah use case digambarkan dengan garis putus-putus vertical, kemudian message yang dikirim oleh object digambarkan dengan garis horizontal secara kronologis dari atas ke bawah.

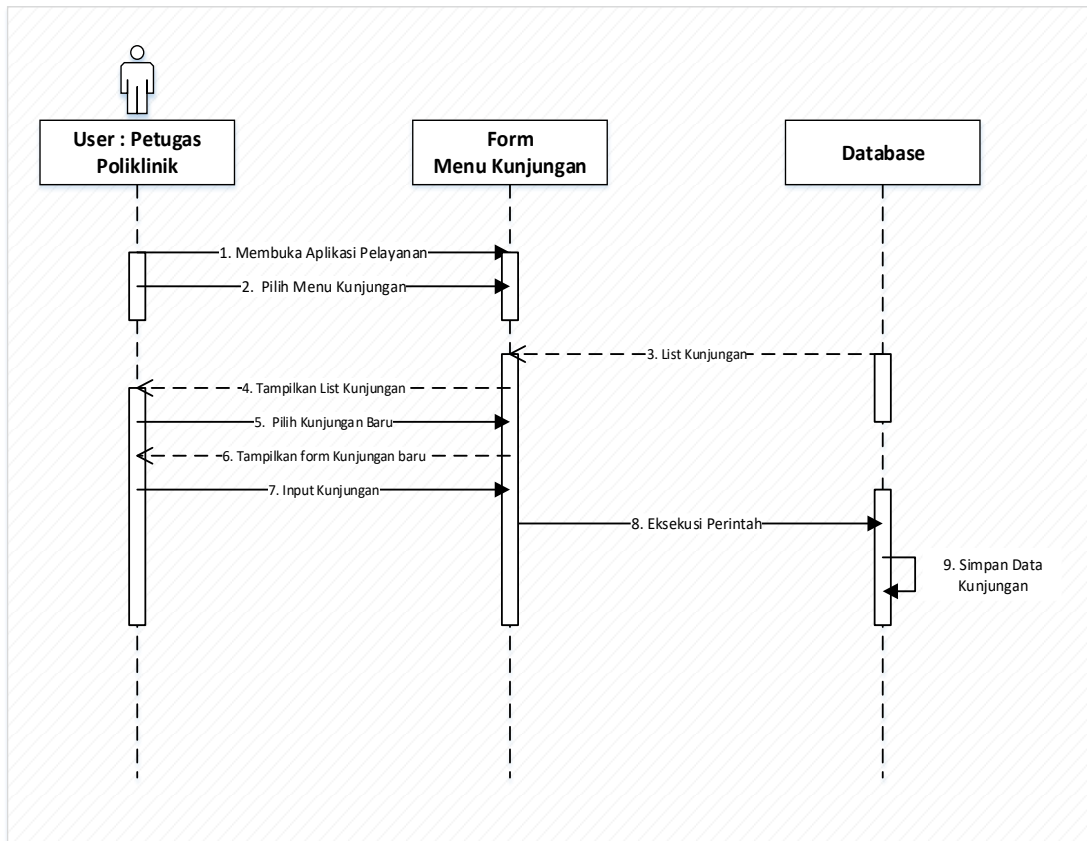
Berikut adalah diagram sekuensial yang menggambarkan fungsionalitas sistem aplikasi pelayanan di klinik.

### 1. Sequence Pendaftaran



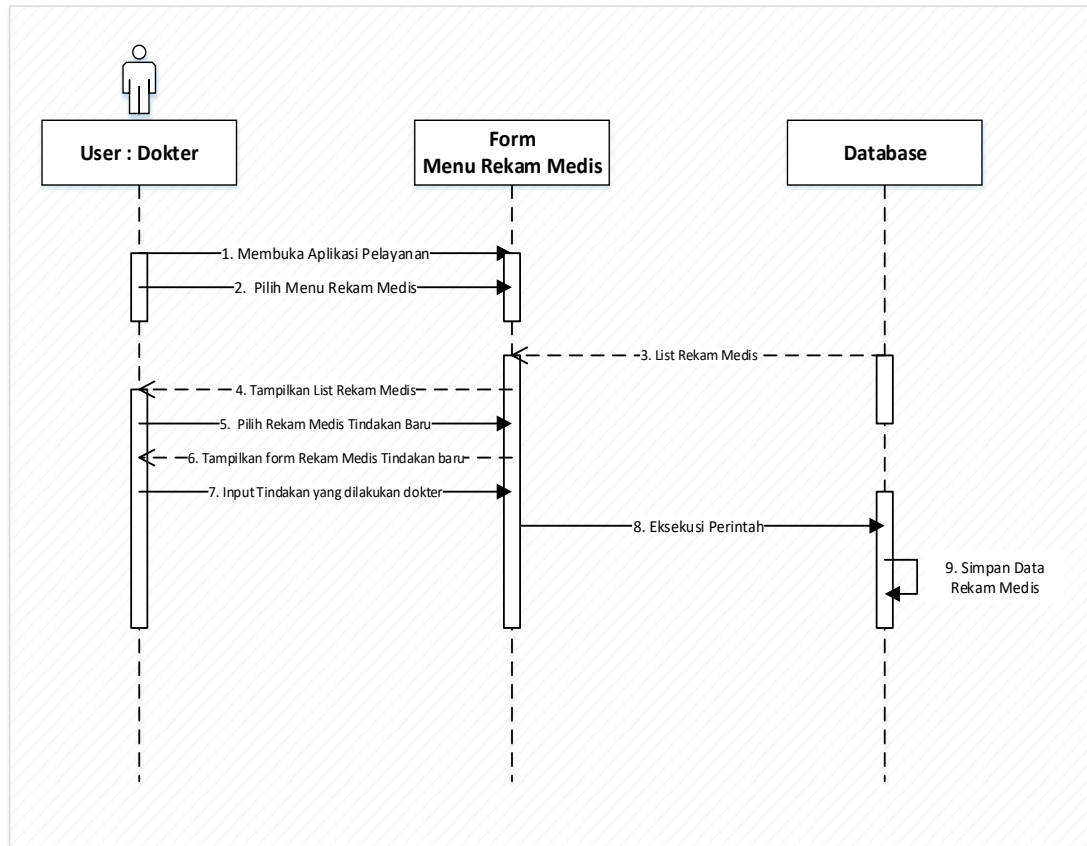
Gambar 4.12 Sequence Pendaftaran

## 2. Sequence Kunjungan



Gambar 4.13 Sequence Kunjungan

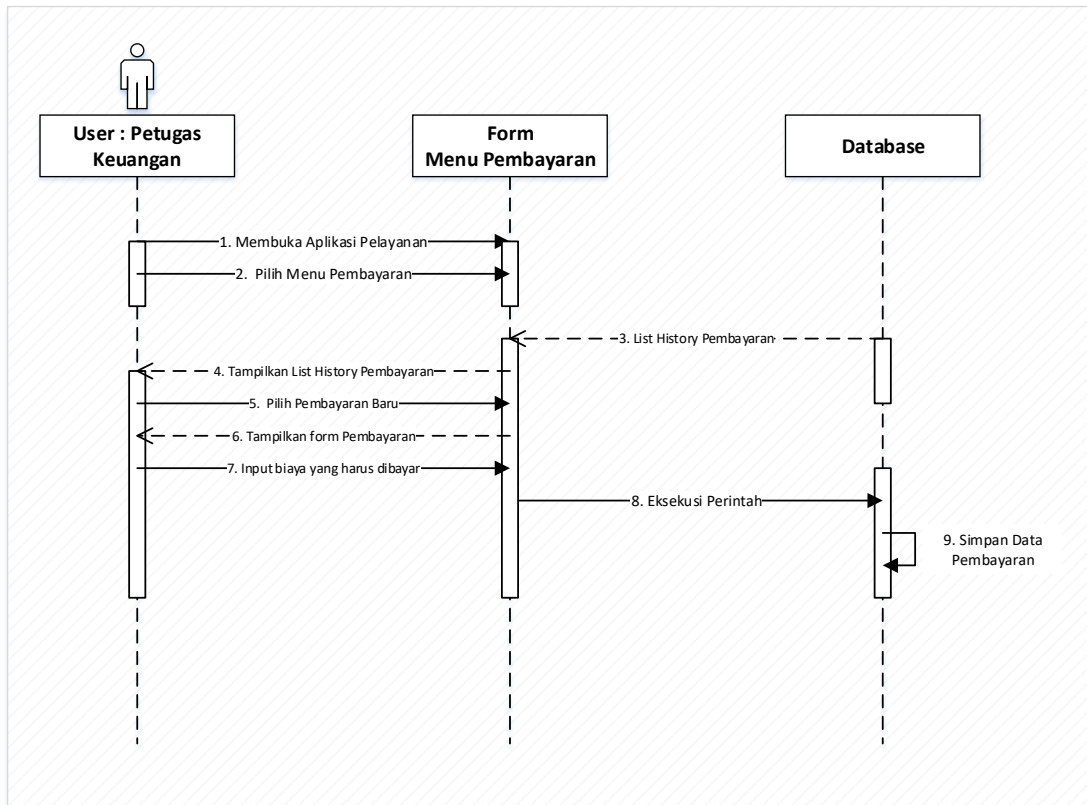
### 3. Sequence Rekam Medis



Gambar 4.14 Sequence Rekam Medis

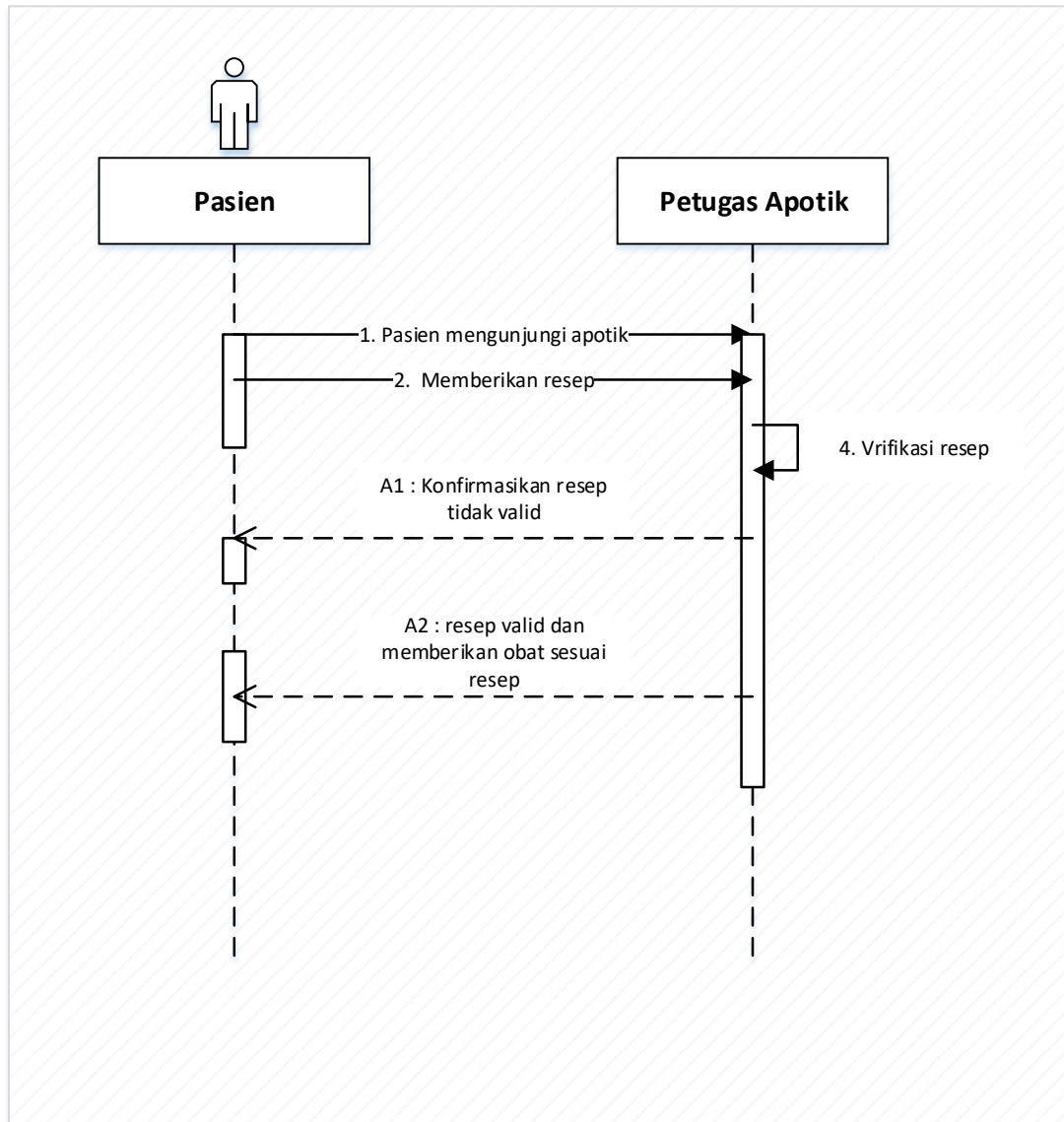


#### 4. Sequence Pembayaran



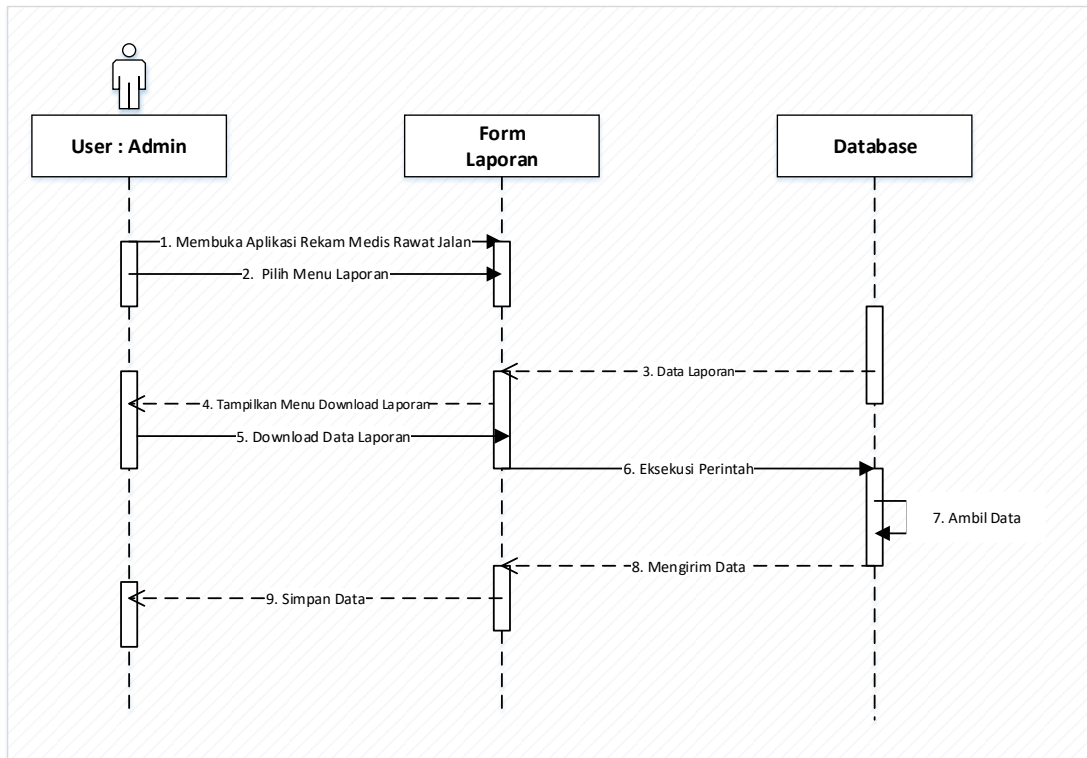
Gambar 4.15 Sequence Pembayaran

## 5. Squence Obat



**Gambar 4.16** Sequence Obat

## 6. Sequence Laporan



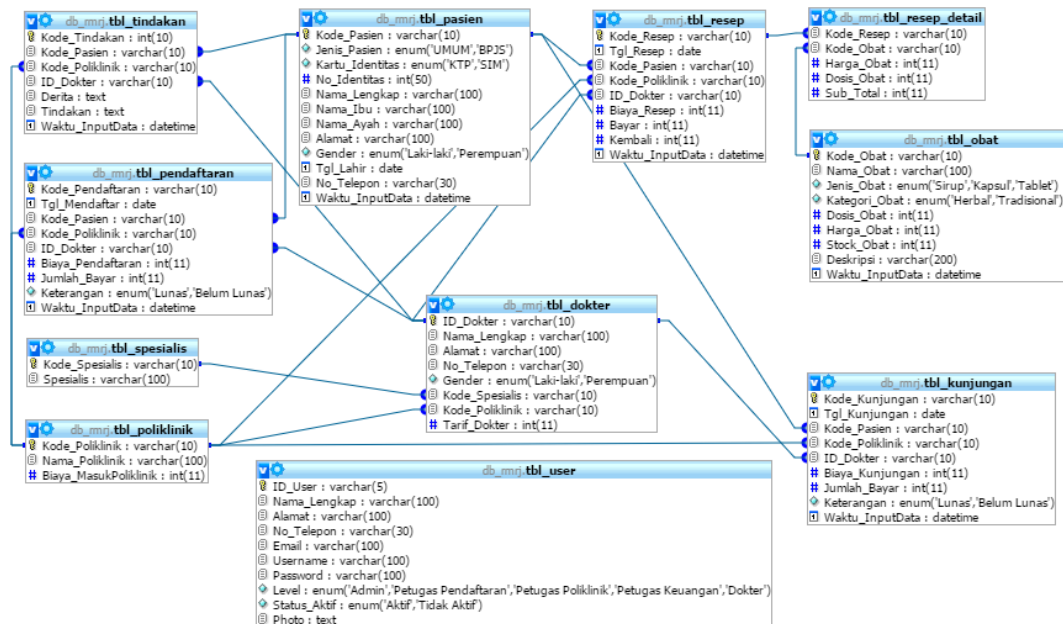
Gambar 4.17 Sequence Laporan

### 4.1.4. Perancangan Data

#### 4.1.4.1. Class Diagram

Diagram kelas atau class diagram menunjukkan interaksi antara kelas dalam system. Class diagram dibangun berdasarkan use case diagram, sequence diagram yang telah dibuat sebelumnya. Diagram kelas merupakan suatu diagram yang menggambarkan atau memvisualisasikan struktur sistem dari kelas-kelas serta

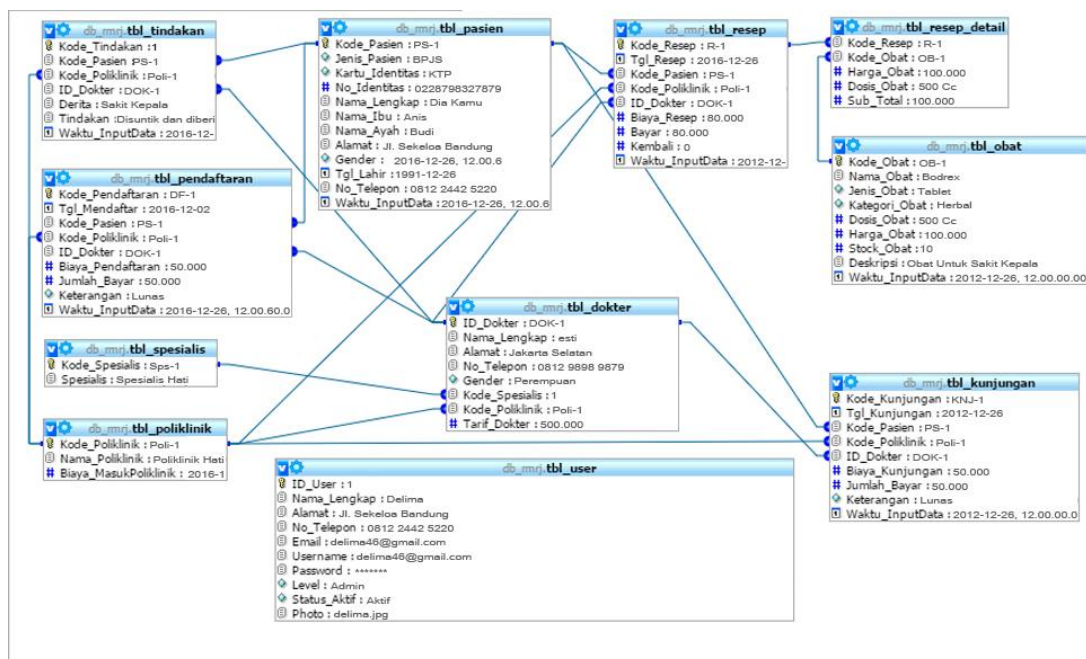
hubungannya. Diagram kelas ini juga menampilkan interaksi dalam kelas-kelas tersebut, atribut apa yang dimiliki atau operasi/metode apa yang dimiliki kelas itu. Diagram kelas sistem aplikasi pelayanan pada klinik padjadjaran dapat dilihat dibawah ini :



Gambar 4.18 Class Diagram

#### 4.1.4.2. Object Diagram

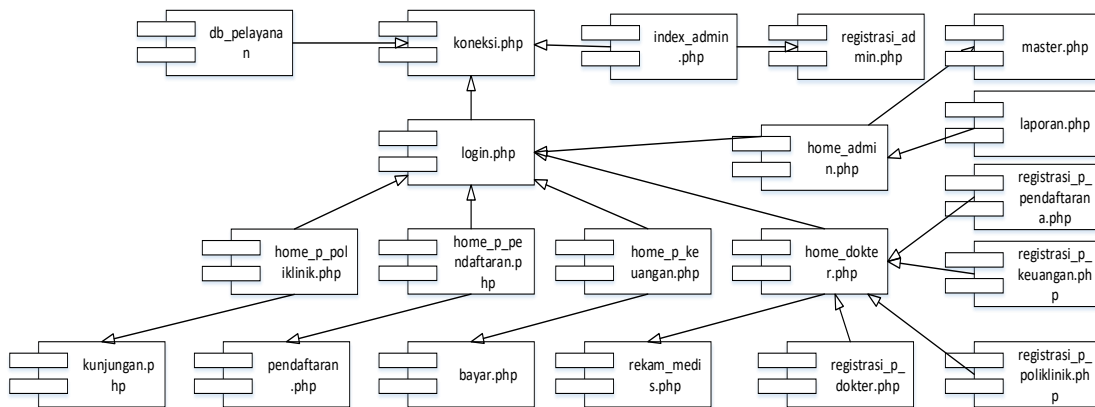
Object Diagram atau Diagram Objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Diagram objek juga merupakan diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu. Diagram objek lebih konkrit daripada kelas diagram, dan sering digunakan untuk memberikan contoh-contoh, ataupun dalam menguji kasus untuk diagram kelas. *Object Diagram* sistem aplikasi pelayanan pada klinik padjadjaran dapat dilihat dibawah ini :



Gambar 4.19 Object Diagram

### 4.1.4.3. Component Diagram

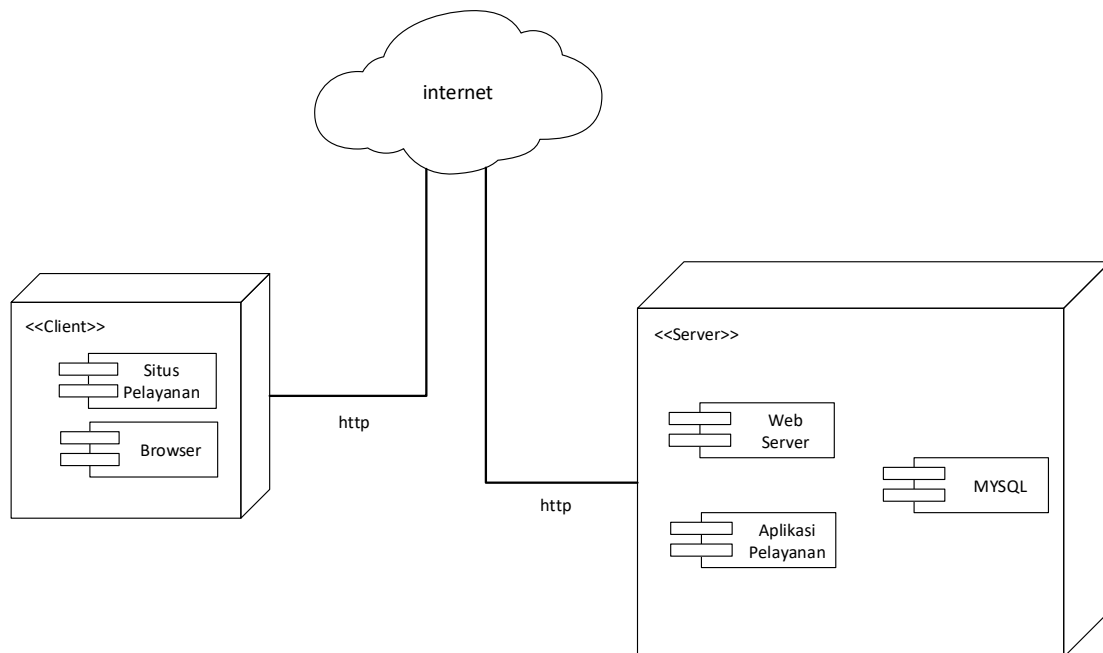
Diagram komponen atau component diagram menunjukkan model secara fisik komponen perangkat lunak pada sistem dan hubungannya. *Component Diagram* sistem aplikasi pelayanan pada klinik padjadjaran dapat dilihat dibawah ini :



**Gambar 4.20 Component Diagram**

#### 4.1.4.4. Deployment Diagram

Diagram deployment atau *deployment diagram* menampilkan rancangan fisik jaringan dimana berbagai komponen akan terdapat disana. *Deployment Diagram* sistem aplikasi pelayanan pada klinik padjadjaran dapat dilihat dibawah ini :



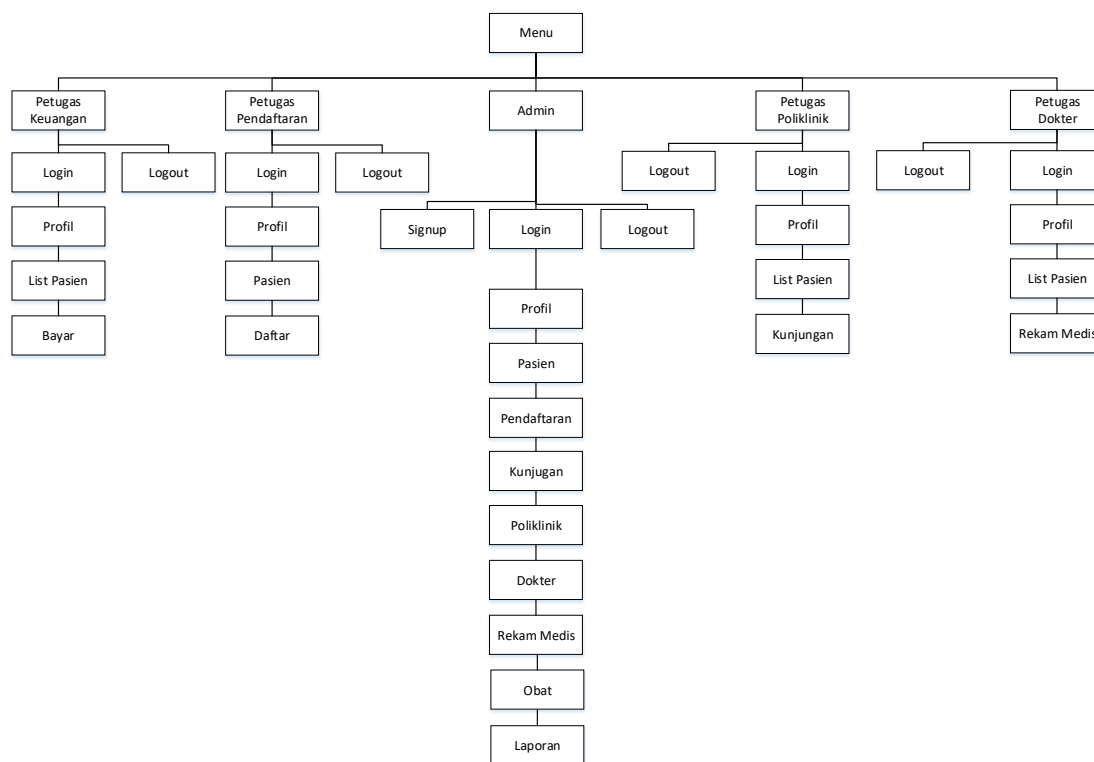
**Gambar 4.21 Deployment Diagram**

## 4.2. Perancangan Antar Muka

Perancangan antar muka merupakan penggambaran tampilan (interface) sebuah sistem yang akan digunakan secara langsung oleh pengguna, dan juga merupakan penggambaran interaksi yang dapat dilakukan oleh pengguna dalam sistem. Adapun

dalam perancangan antar muka ini terdapat beberapa bagian yang harus dilakukan, yaitu dari menentukan rancangan input dan output pada setiap fungsi yang telah ditentukan.

#### 4.2.1 Struktur Menu



**Gambar 4.22 Struktur Menu**

#### 4.2.2 Perancangan Input

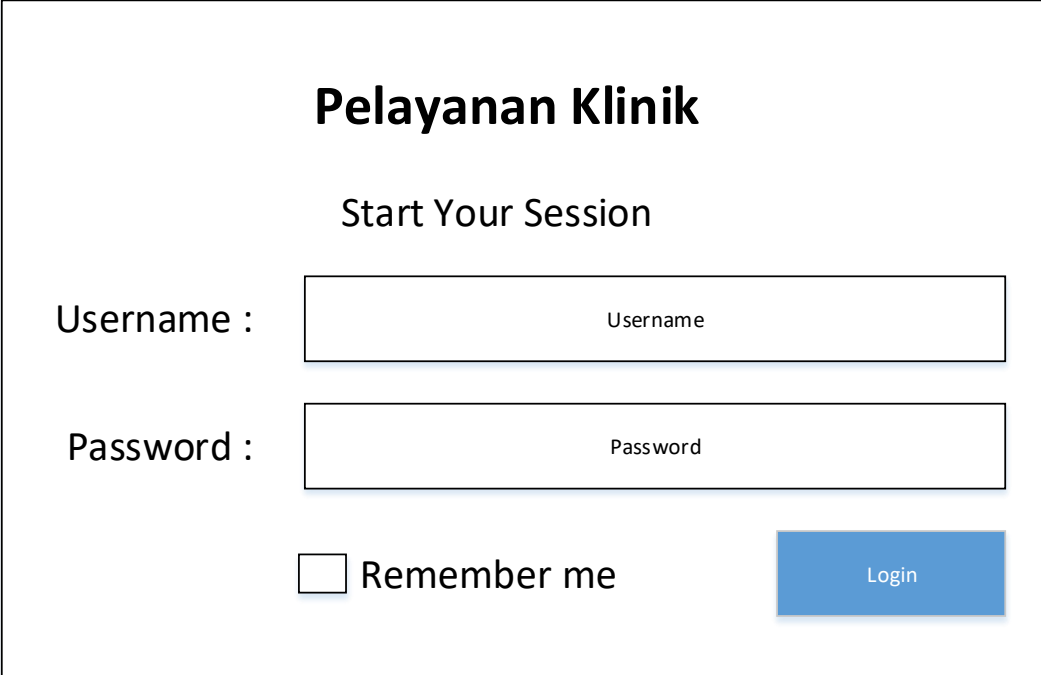
Perancangan input yaitu sebuah tampilan yang dirancang sebagai tempat untuk memasukkan data-data yang diberikan oleh pengguna sistem (user), yang merupakan sumber untuk proses pengolahan yang akan menghasilkan sebuah informasi yang



berguna bagi pengguna. Pengguna dalam aplikasi dibagi ke dalam 5 (tiga) tipe yaitu Admin, Petugas Pendaftaran, Petugas Poliklinik, Petugas Keuangan dan Dokter.

### 1. Perancangan Form Login User

Form yang digunakan untuk masuk kedalam sistem aplikasi pelayanan pada klinik padjadjaran sesuai status login.



The image shows a login form for a clinic service. The form is titled "Pelayanan Klinik" and "Start Your Session". It contains two input fields: "Username" and "Password". Below the password field, there is a checkbox labeled "Remember me" and a blue "Login" button.

**Pelayanan Klinik**

Start Your Session

Username :

Password :

Remember me

**Gambar 4.23 Perancangan Form Login User**

## 2. Perancangan Form Registrasi User

### Form Registrasi User

Nama Lengkap :

Alamat :

No Telepon :

Email :

Level :

Username :

Password :

**Gambar 4.24 Perancangan Form Registrasi User**

### 3. Perancangan Form Pasien

## Pasien

Jenis Pasien :

Kartu Identitas :

No Identitas:

Nama Lengkap :

Nama Ibu :

Nama Ayah :

Alamat :

Gender :  Laki-laki  Perempuan

Tgl Lahir :

No Telepon :

**Gambar 4.25 Perancangan Form Pasien**

#### 4. Perancangan Form Pendaftaran

### Pendaftaran

Pasien :

Poliklinik :

Dokter :

Biaya Pendaftaran :

Jumlah Bayar :

**Gambar 4.26 Perancangan Form Pendaftaran**

## 5. Perancangan Form Kunjungan

The image shows a web form titled "Kunjungan". It contains four input fields, each with a label to its left and a placeholder text inside the field:

- Pasien :
- Poliklinik :
- Dokter :
- Biaya Kunjungan :

At the bottom right of the form is a blue button labeled "Simpan".

**Gambar 4.27 Perancangan Form Kunjungan**

## 6. Perancangan Form Poliklinik

### Poliklinik

Nama Poliklinik :

Biaya Masuk Poliklinik :

**Gambar 4.28 Perancangan Form Poliklinik**

## 7. Perancangan Form Dokter

### Dokter

Nama Lengkap :

Alamat :

No Telepon :

Gender :  Laki-laki  Perempuan

Spesialis :

Poliklinik :

Tarif Dokter :

**Gambar 4.29 Perancangan Form Dokter**

## 8. Perancangan Form Rekam Medis

### Rekam Medis

Pasien :

Poliklinik :

Dokter :

Keluhan :

Tindakan :

**Gambar 4.30 Perancangan Form Rekam Medis**



## 9. Perancangan Form Obat

### Obat

Nama Obat :

Jenis Obat :

Kategori Obat :  Herbal  Tradisional

Dosis Obat :

Harga Obat :

Stock Obat :

Deskripsi :

**Gambar 4.31 Perancangan Form Obat**

## 10. Perancangan Form Resep

### Resep

Pasien :

Poliklinik :

Dokter :

Biaya Resep

Nama Obat	Harga Obat	Dosis Obat	Sub Total
<input type="text" value="Nama Obat"/>	<input type="text" value="Harga Obat"/>	<input type="text" value="Dosis Obat"/>	<input type="text" value="Sub Total"/>
<input type="text" value="Nama Obat"/>	<input type="text" value="Harga Obat"/>	<input type="text" value="Dosis Obat"/>	<input type="text" value="Sub Total"/>
<input type="text" value="Nama Obat"/>	<input type="text" value="Harga Obat"/>	<input type="text" value="Dosis Obat"/>	<input type="text" value="Sub Total"/>

Gambar 4.32 Perancangan Form Resep

## 11. Perancangan Form Pembayaran

### Pembayaran

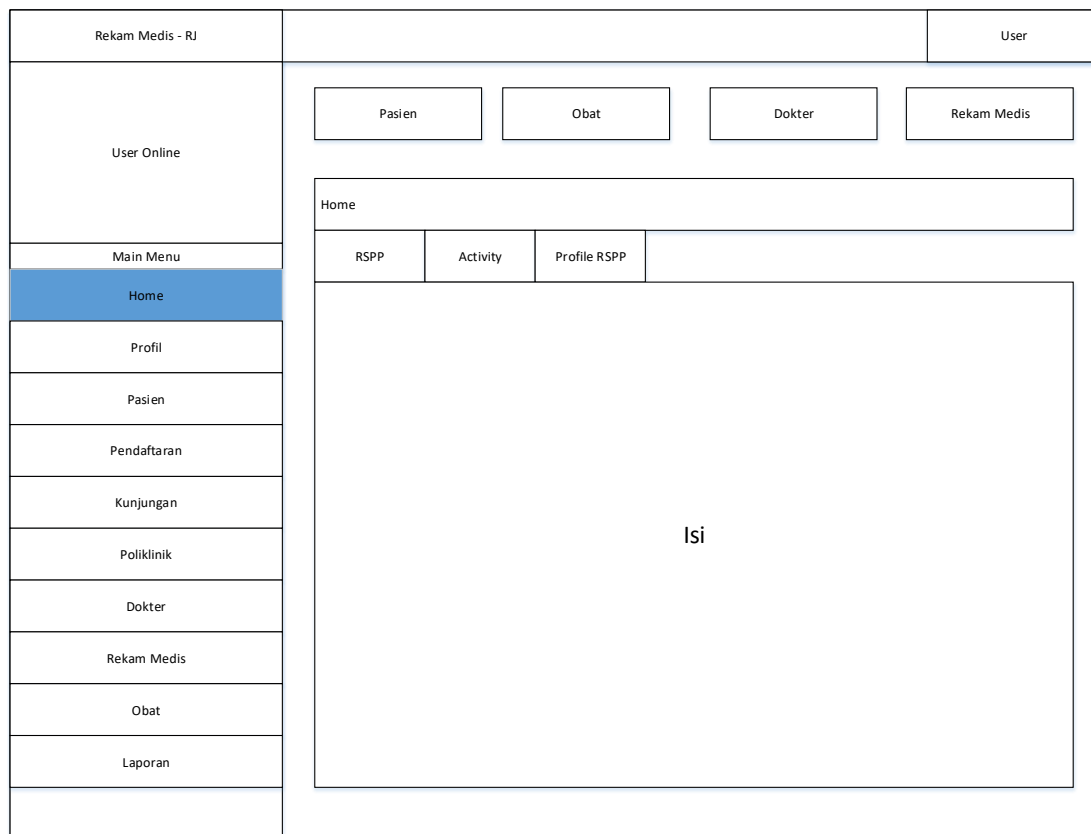
P. Keuangan :	Petugas Keuangan
Pasien :	Pasien
Biaya Kunjungan :	Biaya Kunjungan
Biaya Resep :	Biaya Resep
Jumlah Bayar :	Jumlah Bayar
Bayar :	Bayar
Kembali :	Kembali

**Gambar 4.33 Perancangan Form Pembayaran**

### 4.2.3 Perancangan Output

Perancangan output atau keluaran merupakan hasil dari pengolahan data yang diterima dari proses masukan data berupa informasi yang berguna bagi pengguna atau user. Adapun beberapa rancangan output dari Sistem Aplikasi *e-learning* Politeknik LP3I Medan ini adalah sebagai berikut:

#### 1. Perancangan Output Halaman Utama



**Gambar 4.34 Perancangan Output Halaman Utama**

## 2. Perancangan Output Pasien

Rekam Medis - RJ							User																				
User Online	Pasien	Obat	Dokter	Rekam Medis																							
Main Menu	Pasien																										
Home	Data Pasien	Tambah Pasien																									
Profil	<table border="1"> <thead> <tr> <th>Kode Pasien</th> <th>Jenis Pasien</th> <th>K. Identitas</th> <th>No Identitas</th> <th>Nama Lengkap</th> <th>Alamat</th> <th>Gender</th> </tr> </thead> <tbody> <tr> <td>PS-1</td> <td>BPJS</td> <td>KTP</td> <td>112293844</td> <td>Putri</td> <td>Bandung</td> <td>Perempuan</td> </tr> <tr> <td>PS2</td> <td>Umum</td> <td>SIM</td> <td>228839879</td> <td>Putra</td> <td>Bandung</td> <td>Laki-laki</td> </tr> </tbody> </table>						Kode Pasien	Jenis Pasien	K. Identitas	No Identitas	Nama Lengkap	Alamat	Gender	PS-1	BPJS	KTP	112293844	Putri	Bandung	Perempuan	PS2	Umum	SIM	228839879	Putra	Bandung	Laki-laki
Kode Pasien	Jenis Pasien	K. Identitas	No Identitas	Nama Lengkap	Alamat	Gender																					
PS-1	BPJS	KTP	112293844	Putri	Bandung	Perempuan																					
PS2	Umum	SIM	228839879	Putra	Bandung	Laki-laki																					
Pasien																											
Pendaftaran																											
Kunjungan																											
Poliklinik																											
Dokter																											
Rekam Medis																											
Obat																											
Laporan																											

**Gambar 4.35 Perancangan Output Pasien**

### 3. Perancangan Output Pendaftaran

Rekam Medis - RJ							User																					
User Online	Pasien	Obat	Dokter	Rekam Medis																								
Main Menu	Pendaftaran																											
Home	Data Pendaftaran	Tambah Pendaftaran																										
Profil	<table border="1"> <thead> <tr> <th>Kode Pendaft</th> <th>Tgl Daftar</th> <th>Kode Pasien</th> <th>Kode Poliklinik</th> <th>Kode Dokter</th> <th>Biaya Pendaftaran</th> <th>Jumlah Bayar</th> </tr> </thead> <tbody> <tr> <td>DF-1</td> <td>2016-12-12</td> <td>PS-1</td> <td>Poli-1</td> <td>Dok-1</td> <td>Rp. 100.000</td> <td>Rp. 500.000</td> </tr> <tr> <td>DF-2</td> <td>2016-12-13</td> <td>PS-2</td> <td>Poli-2</td> <td>Dok-2</td> <td>Rp. 100.000</td> <td>Rp. 500.000</td> </tr> </tbody> </table>							Kode Pendaft	Tgl Daftar	Kode Pasien	Kode Poliklinik	Kode Dokter	Biaya Pendaftaran	Jumlah Bayar	DF-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp. 100.000	Rp. 500.000	DF-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp. 100.000	Rp. 500.000
Kode Pendaft	Tgl Daftar	Kode Pasien	Kode Poliklinik	Kode Dokter	Biaya Pendaftaran	Jumlah Bayar																						
DF-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp. 100.000	Rp. 500.000																						
DF-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp. 100.000	Rp. 500.000																						
Pasien																												
Pendaftaran																												
Kunjungan																												
Poliklinik																												
Dokter																												
Rekam Medis																												
Obat																												
Laporan																												

**Gambar 4.36 Perancangan Output Pendaftaran**

#### 4. Perancangan Output Kunjungan

Rekam Medis - RJ						User																		
User Online	Pasien	Obat	Dokter	Rekam Medis																				
Main Menu	Kunjungan																							
Home	Data Kunjungan	Tambah Kunjungan																						
Profil	<table border="1"> <thead> <tr> <th>Kode Kunju</th> <th>Tgl Kunjungan</th> <th>Kode Pasien</th> <th>Kode Poliklinik</th> <th>Kode Dokter</th> <th>Jumlah Bayar</th> </tr> </thead> <tbody> <tr> <td>KJN-1</td> <td>2016-12-12</td> <td>PS-1</td> <td>Poli-1</td> <td>Dok-1</td> <td>Rp. 400.000</td> </tr> <tr> <td>KJN-2</td> <td>2016-12-13</td> <td>PS-2</td> <td>Poli-2</td> <td>Dok-2</td> <td>Rp. 400.000</td> </tr> </tbody> </table>						Kode Kunju	Tgl Kunjungan	Kode Pasien	Kode Poliklinik	Kode Dokter	Jumlah Bayar	KJN-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp. 400.000	KJN-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp. 400.000
Kode Kunju	Tgl Kunjungan	Kode Pasien	Kode Poliklinik	Kode Dokter	Jumlah Bayar																			
KJN-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp. 400.000																			
KJN-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp. 400.000																			
Pasien																								
Pendaftaran																								
<b>Kunjungan</b>																								
Poliklinik																								
Dokter																								
Rekam Medis																								
Obat																								
Laporan																								

**Gambar 4.37 Perancangan Output Kunjungan**

## 5. Perancangan Output Poliklinik

Rekam Medis - RJ				User									
User Online	Pasien	Obat	Dokter	Rekam Medis									
Main Menu	Poliklinik												
Home	Data Poliklinik	Tambah Poliklinik											
Profil	<table border="1"> <thead> <tr> <th>Kode Polikli</th> <th>Nama Poliklinik</th> <th>Biaya Masuk Poliklinik</th> </tr> </thead> <tbody> <tr> <td>Poli-1</td> <td>Poliklinik Mata</td> <td>Rp. 200.000</td> </tr> <tr> <td>Poli-2</td> <td>Poliklinik Hati</td> <td>Rp. 300.000</td> </tr> </tbody> </table>				Kode Polikli	Nama Poliklinik	Biaya Masuk Poliklinik	Poli-1	Poliklinik Mata	Rp. 200.000	Poli-2	Poliklinik Hati	Rp. 300.000
Kode Polikli	Nama Poliklinik	Biaya Masuk Poliklinik											
Poli-1	Poliklinik Mata	Rp. 200.000											
Poli-2	Poliklinik Hati	Rp. 300.000											
Pasien													
Pendaftaran													
Kunjungan													
Poliklinik													
Dokter													
Rekam Medis													
Obat													
Laporan													

**Gambar 4.38 Perancangan Output Poliklinik**



## 6. Perancangan Output Dokter

Rekam Medis - RJ	User																										
User Online	Pasien	Obat	Dokter	Rekam Medis																							
Main Menu	Dokter																										
Home	Data Dokter	Tambah Dokter																									
Profil	<table border="1"> <thead> <tr> <th>Kode Dokter</th> <th>Nama Dokter</th> <th>Alamat</th> <th>No Telepon</th> <th>Gender</th> <th>Spesialis</th> <th>Poliklinik</th> </tr> </thead> <tbody> <tr> <td>Dok-1</td> <td>Esti</td> <td>Jakarta</td> <td>08129834798</td> <td>Perempuan</td> <td>Sps-1</td> <td>Poli-1</td> </tr> <tr> <td>Dok-2</td> <td>Esta</td> <td>Bandung</td> <td>08138793749</td> <td>Laki-laki</td> <td>Sps-2</td> <td>Poli-2</td> </tr> </tbody> </table>						Kode Dokter	Nama Dokter	Alamat	No Telepon	Gender	Spesialis	Poliklinik	Dok-1	Esti	Jakarta	08129834798	Perempuan	Sps-1	Poli-1	Dok-2	Esta	Bandung	08138793749	Laki-laki	Sps-2	Poli-2
Kode Dokter	Nama Dokter	Alamat	No Telepon	Gender	Spesialis	Poliklinik																					
Dok-1	Esti	Jakarta	08129834798	Perempuan	Sps-1	Poli-1																					
Dok-2	Esta	Bandung	08138793749	Laki-laki	Sps-2	Poli-2																					
Pasien																											
Pendaftaran																											
Kunjungan																											
Poliklinik																											
<b>Dokter</b>																											
Rekam Medis																											
Obat																											
Laporan																											

**Gambar 4.39 Perancangan Output Dokter**

## 7. Perancangan Output Rekam Medis

Rekam Medis - RJ						User																		
User Online	Pasien	Obat	Dokter	Rekam Medis																				
Main Menu	Rekam Medis																							
Home	Data Tindakan	Tambah Tindakan																						
Profil	<table border="1"> <thead> <tr> <th>No</th> <th>Kode Pasien</th> <th>Kode Poliklinik</th> <th>Kode Dokter</th> <th>Keluhan Pasien</th> <th>Tindakan</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PS-1</td> <td>Poli-1</td> <td>Dok-1</td> <td>Sakit Kepala</td> <td>Dijarum dan diberi obat</td> </tr> <tr> <td>2</td> <td>PS-2</td> <td>Poli-2</td> <td>Dok-2</td> <td>Demam</td> <td>Dijarum dan diberi obat</td> </tr> </tbody> </table>						No	Kode Pasien	Kode Poliklinik	Kode Dokter	Keluhan Pasien	Tindakan	1	PS-1	Poli-1	Dok-1	Sakit Kepala	Dijarum dan diberi obat	2	PS-2	Poli-2	Dok-2	Demam	Dijarum dan diberi obat
No	Kode Pasien	Kode Poliklinik	Kode Dokter	Keluhan Pasien	Tindakan																			
1	PS-1	Poli-1	Dok-1	Sakit Kepala	Dijarum dan diberi obat																			
2	PS-2	Poli-2	Dok-2	Demam	Dijarum dan diberi obat																			
Pasien																								
Pendaftaran																								
Kunjungan																								
Poliklinik																								
Dokter																								
Rekam Medis																								
Obat																								
Laporan																								

**Gambar 4.40 Perancangan Output Rekam Medis**

## 8. Perancangan Output Obat

Rekam Medis - RJ							User																					
User Online	Pasien	Obat	Dokter	Rekam Medis																								
Main Menu	Obat																											
Home	Data Obat	Tambah Obat																										
Profil	<table border="1"> <thead> <tr> <th>No</th> <th>Kode Obat</th> <th>Nama Obat</th> <th>Jenis</th> <th>Dosis</th> <th>Harga</th> <th>Stock</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Obt-1</td> <td>Bodrex</td> <td>Sirup</td> <td>500</td> <td>Rp. 10.000</td> <td>10</td> </tr> <tr> <td>2</td> <td>Obt-2</td> <td>Parasetamol</td> <td>Tablet</td> <td>500</td> <td>Rp. 20.000</td> <td>20</td> </tr> </tbody> </table>							No	Kode Obat	Nama Obat	Jenis	Dosis	Harga	Stock	1	Obt-1	Bodrex	Sirup	500	Rp. 10.000	10	2	Obt-2	Parasetamol	Tablet	500	Rp. 20.000	20
No	Kode Obat	Nama Obat	Jenis	Dosis	Harga	Stock																						
1	Obt-1	Bodrex	Sirup	500	Rp. 10.000	10																						
2	Obt-2	Parasetamol	Tablet	500	Rp. 20.000	20																						
Pasien																												
Pendaftaran																												
Kunjungan																												
Poliklinik																												
Dokter																												
Rekam Medis																												
Obat																												
Laporan																												

**Gambar 4.41 Perancangan Output Obat**

## 9. Perancangan Output Resep

Rekam Medis - RJ							User																					
User Online	Pasien	Obat	Dokter	Rekam Medis																								
Main Menu	Resep																											
Home	Data Resep	Buat Resep																										
Profil	<table border="1"> <thead> <tr> <th>No</th> <th>Kode Resep</th> <th>Tgl Resep</th> <th>Kode Pasien</th> <th>Kode Poliklinik</th> <th>Kode Dokter</th> <th>Biaya Resep</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>R-1</td> <td>2016-12-12</td> <td>PS-1</td> <td>Poli-1</td> <td>Dok-1</td> <td>Rp.200.000</td> </tr> <tr> <td>2</td> <td>R-2</td> <td>2016-12-13</td> <td>PS-2</td> <td>Poli-2</td> <td>Dok-2</td> <td>Rp.300.000</td> </tr> </tbody> </table>							No	Kode Resep	Tgl Resep	Kode Pasien	Kode Poliklinik	Kode Dokter	Biaya Resep	1	R-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp.200.000	2	R-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp.300.000
No	Kode Resep	Tgl Resep	Kode Pasien	Kode Poliklinik	Kode Dokter	Biaya Resep																						
1	R-1	2016-12-12	PS-1	Poli-1	Dok-1	Rp.200.000																						
2	R-2	2016-12-13	PS-2	Poli-2	Dok-2	Rp.300.000																						
Pasien																												
Pendaftaran																												
Kunjungan																												
Poliklinik																												
Dokter																												
Rekam Medis																												
Obat																												
Resep																												

**Gambar 4.42 Perancangan Output Resep**

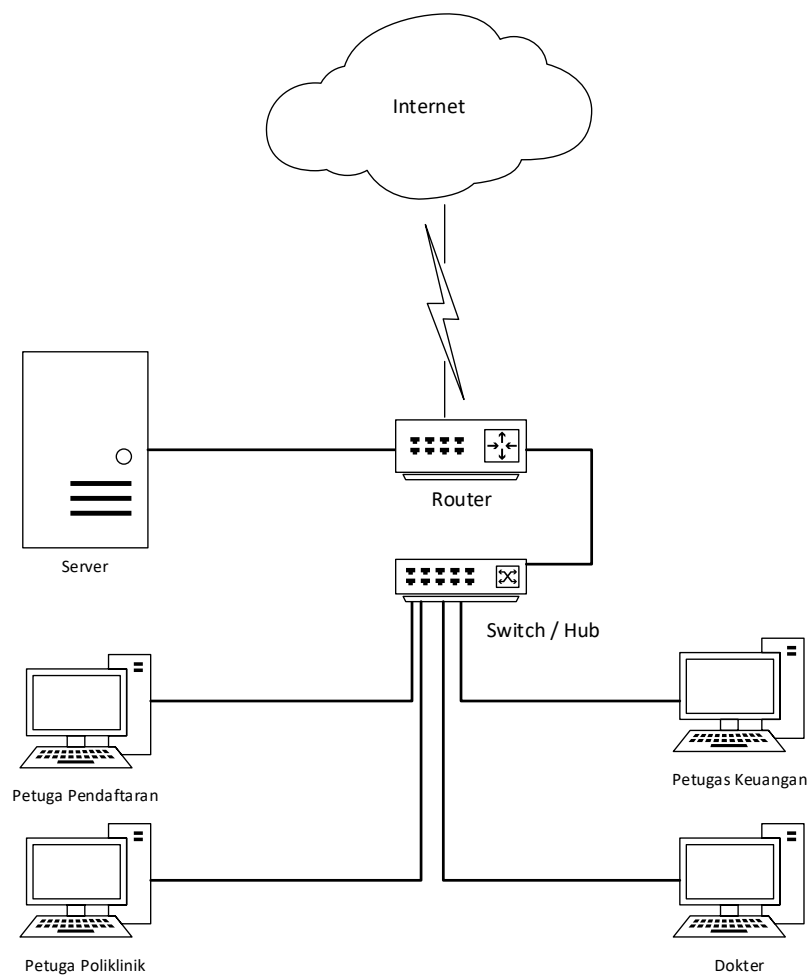
## 10. Perancangan Output Pembayaran

Rekam Medis - RJ						User																		
User Online	Pasien	Obat	Dokter	Rekam Medis																				
Main Menu	Pembayaran																							
Home	Data Pembayaran	Tambah Pembayaran																						
Profil	<table border="1"> <thead> <tr> <th>No</th> <th>Petugas Keuangan</th> <th>Kode Pasien</th> <th>Biaya Kunjungan</th> <th>Biaya Resep</th> <th>Jumlah Bayar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pku-1</td> <td>PS-1</td> <td>Rp.500.000</td> <td>Rp.200.000</td> <td>Rp.700.000</td> </tr> <tr> <td>2</td> <td>Pku-1</td> <td>PS-2</td> <td>Rp.400.000</td> <td>Rp.200.000</td> <td>Rp.600.000</td> </tr> </tbody> </table>						No	Petugas Keuangan	Kode Pasien	Biaya Kunjungan	Biaya Resep	Jumlah Bayar	1	Pku-1	PS-1	Rp.500.000	Rp.200.000	Rp.700.000	2	Pku-1	PS-2	Rp.400.000	Rp.200.000	Rp.600.000
No	Petugas Keuangan	Kode Pasien	Biaya Kunjungan	Biaya Resep	Jumlah Bayar																			
1	Pku-1	PS-1	Rp.500.000	Rp.200.000	Rp.700.000																			
2	Pku-1	PS-2	Rp.400.000	Rp.200.000	Rp.600.000																			
Pasien																								
Pendaftaran																								
Kunjungan																								
Poliklinik																								
Dokter																								
Rekam Medis																								
Obat																								
Pembayaran																								

**Gambar 4.43 Perancangan Output Pembayaran**

### 4.3 Perancangan Arsitektur Jaringan

Perancangan arsitektur jaringan adalah bentuk rancangan suatu konfigurasi jaringan yang diusulkan untuk diterapkan di dalam pelaksanaan aplikasi pelayanan klinik padjajaran dapat dilihat pada gambar berikut ini :



**Gambar 4.44 Perancangan Arsitektur Jaringan**

Pada gambar diatas menunjukkan arsitektur jaringan dimana aplikasi pelayanan ini bisa diakses oleh komputer pengguna baik itu Admin, P. Pendaftaran, P. Poliklinik, P. Keuangan, dan juga Dokter yang sudah terhubung dengan internet secara langsung melalui *browser* dengan mengirim request ke web server.

#### **4.4 Pengujian**

Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

##### **4.4.1 Rencana Pengujian**

Pengujian perangkat lunak ini menggunakan metode pengujian *black box*. Pengujian *black box* ini tidak perlu tahu apa yang sesungguhnya terjadi dalam sistem atau perangkat lunak, yang diuji adalah masukan serta keluarannya. Dengan berbagai masukan yang diberikan, apakah sistem atau perangkat lunak memberikan keluaran seperti yang kita harapkan atau tidak.

Pengujian aplikasi pelayanan menggunakan data uji berupa sebuah data masukan dari *user*. Berikut ini adalah rencana pengujian aplikasi pelayanan :

**Tabel 4.11 Rencana Pengujian**

Item yang Diuji	Detail Pengujian	Jenis Pengujian
Login Website	<ol style="list-style-type: none"> <li>1. Menampilkan form login.</li> <li>2. Mengisi form login dengan menginput username dan password.</li> <li>3. Menampilkan notifikasi jika belum terdaftar.</li> <li>4. Menampilkan notifikasi username atau password salah.</li> </ol>	<i>Black Box</i>
Registrasi	<ol style="list-style-type: none"> <li>1. Mengisi form registrasi</li> <li>2. Menampilkan notifikasi jika data tidak lengkap.</li> <li>3. Menampilkan notifikasi jika kode/id user tidak aktif.</li> </ol>	<i>Black Box</i>
Input Pasien	<ol style="list-style-type: none"> <li>1. Mengisi form pasien.</li> <li>2. Menampilkan notifikasi jika pasien yang diinput sudah ada.</li> <li>3. Menampilkan notifikasi jika data pasien tidak lengkap.</li> </ol>	<i>Black Box</i>
Input Pendaftaran	<ol style="list-style-type: none"> <li>1. Mengisi form pendaftaran.</li> <li>2. Menampilkan notifikasi jika pasien sudah pernah mendaftar.</li> </ol>	<i>Black Box</i>
Input Data Kunjungan	<ol style="list-style-type: none"> <li>1. Mengisi form input data kunjungan pasien.</li> <li>2. Menampilkan notifikasi jika pasien belum melakukan pendaftaran.</li> </ol>	<i>Black Box</i>
Input Poliklinik	<ol style="list-style-type: none"> <li>1. Mengisi form input poliklinik.</li> <li>2. Menampilkan notifikasi jika poliklinik sudah ada.</li> </ol>	<i>Black Box</i>
Input Data Dokter	<ol style="list-style-type: none"> <li>1. Mengisi form input data dokter.</li> <li>2. Menampilkan notifikasi jika dokter sudah pernah diinput.</li> </ol>	<i>Black Box</i>



Pelayanan	<ol style="list-style-type: none"> <li>1. Mengisi form pelayanan.</li> <li>2. Menampilkan semua kode pasien.</li> <li>3. Menampilkan id dokter yang menangani.</li> </ol>	<i>Black Box</i>
Obat	<ol style="list-style-type: none"> <li>1. Mengisi form input data obat.</li> <li>2. Menampilkan notifikasi jika obat sudah pernah ditambahkan.</li> <li>3. Menampilkan notifikasi jika data obat tidak lengkap.</li> </ol>	<i>Black Box</i>
Resep	<ol style="list-style-type: none"> <li>1. Mengisi form resep yang akan diberikan kepada pasien.</li> <li>2. Menampilkan kode pasien yang telah melakukan pendaftaran.</li> <li>3. Menampilkan id dokter yang menangani pasien.</li> <li>4. Menampilkan data obat yang tersedia</li> <li>5. Menghitung biaya dari semua obat yang dipilih.</li> </ol>	<i>Black Box</i>
Pembayaran	<ol style="list-style-type: none"> <li>1. Mengisi form pembayaran yang akan dibebankan kepada pasien.</li> <li>2. Menampilkan kode pasien yang telah melakukan pendaftaran atau kunjungan.</li> <li>3. Menampilkan biaya kunjungan.</li> <li>4. Menampilkan biaya resep.</li> <li>5. Menghitung biaya dari semua biaya kunjungan dan juga biaya resep.</li> </ol>	<i>Black Box</i>
Laporan Pasien	Menampilkan data pasien yang telah mendaftar	<i>Black Box</i>
Laporan Dokter	Menampilkan data dokter yang terdaftar didalam pelayanan.	<i>Black Box</i>
Laporan Obat	Menampilkan data obat yang tersedia.	<i>Black Box</i>

Laporan absensi	Menampilkan data pelayanan termasuk tindakan dokter terhadap pasien.	<i>Black Box</i>
Logout	Menghapus session login.	<i>Black Box</i>

#### 4.4.2 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian diatas, kegiatan selanjutnya adalah melakukan pengujian. Berikut ini adalah pengujian aplikasi sesuai dengan rencana pengujiannya;

##### 1. Pengujian login

**Tabel 4.12 Pengujian Login**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Masuk ke alamat website	Menampilkan form login	Menampilkan form login	[X] Diterima [ ] Ditolak
Mengisi form login dengan username dan password	Login berhasil dan menampilkan menu utama sesuai hak akses	Login berhasil dan menampilkan menu utama sesuai hak akses	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Mengosongkan data salah satu atau semua data	Memberikan pesan 'Gagal!'	Memberikan pesan 'Gagal!'	[X] Diterima [ ] Ditolak
Memasukkan username atau password yang salah	Memberikan pesan 'Gagal!'	Memberikan pesan 'Gagal!'	[X] Diterima [ ] Ditolak

Tabel 4.13 Pengujian Input Data Pasien

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form input data pasien	Dapat mengisi form input data pasien.	Pasien telah berhasil ditambahkan	[X] Diterima [ ] Ditolak
Klik tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Mengosongkan data salah satu atau semua data	Memberikan pesan 'Gagal!'	Memberikan pesan 'Gagal!'	[X] Diterima [ ] Ditolak

Tabel 4.14 Pengujian Input Data Pendaftaran

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form input data pendaftaran	Dapat mengisi form input data pendaftaran.	Form data pendaftaran telah terisi	[X] Diterima [ ] Ditolak
Klik tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klik tombol batal	Membatalkan inputan. Kembali ke halaman kelas	Input batal dan kembali ke halaman kelas.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			

Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan ‘Gagal!’ data sudah ada	Tampil pesan ‘Gagal!’ dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak
---	--	--	-----------------------------

**Tabel 4.15 Pengujian Input Data Kunjungan**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form input data kunjungan	Dapat mengisi form input data kunjungan.	Form data kunjungan telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman kunjungan	Input batal dan kembali kehalaman kunjungan.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan ‘Gagal!’ data sudah ada	Tampil pesan ‘Gagal!’ dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak

**Tabel 4.16 Pengujian Input Data Poliklinik**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>

Mengisi form input data poliklinik	Dapat mengisi form input data poliklinik.	Form data poliklinik telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman poliklinik	Input batal dan kembali kehalaman poliklinik.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan 'Gagal!' data sudah ada	Tampil pesan 'Gagal!' dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak

**Tabel 4.17 Pengujian Input Data Dokter**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form input data dokter	Dapat mengisi form input data dokter.	Form data dokter telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman dokter	Input batal dan kembali kehalaman dokter.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			

Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan ‘Gagal!’ data sudah ada	Tampil pesan ‘Gagal!’ dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak
---	--	--	-----------------------------

**Tabel 4.18 Pengujian Input Data Pelayanan**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form pelayanan.	Dapat mengisi form input data pelayanan.	Form data pelayanan telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman pelayanan.	Input batal dan kembali kehalaman pelayanan.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan ‘Gagal!’ data sudah ada	Tampil pesan ‘Gagal!’ dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak

**Tabel 4.19 Pengujian Input Data Obat**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form data obat.	Dapat mengisi form input data obat.	Form data obat telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman obat.	Input batal dan kembali kehalaman obat.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan 'Gagal!' data sudah ada	Tampil pesan 'Gagal!' dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak

**Tabel 4.20 Pengujian Input Data Resep**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form resep.	Dapat mengisi form input data resep.	Form data resep telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan	Tampil pesan bahwa data telah tersimpan	[X] Diterima [ ] Ditolak

	kedalam database.	kedalam database.	
Klick tombol batal	Membatalkan inputan. Kembali ke halaman resep.	Input batal dan kembali kehalaman resep.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan 'Gagal!' data sudah ada	Tampil pesan 'Gagal!' dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak

**Tabel 4.21 Pengujian Input Data Pembayaran**

<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Aktifitas yang Dilakukan</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil yang Didapatkan</b>	<b>Keterangan</b>
Mengisi form pembayaran.	Dapat mengisi form input data pembayaran.	Form data pembayaran telah terisi	[X] Diterima [ ] Ditolak
Klick tombol simpan	Menampilkan pesan data berhasil disimpan kedalam database.	Tampil pesan bahwa data telah tersimpan kedalam database.	[X] Diterima [ ] Ditolak
Klick tombol batal	Membatalkan inputan. Kembali ke halaman pembayaran.	Input batal dan kembali kehalaman pembayaran.	[X] Diterima [ ] Ditolak
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data yang diinput sama dengan data yang sudah ada didalam database.	Memberikan pesan 'Gagal!' data sudah ada	Tampil pesan 'Gagal!' dan data tidak tersimpan kedalam database.	[X] Diterima [ ] Ditolak



#### **4.4.3. Kesimpulan Hasil Pengujian**

Berdasarkan hasil uji sampel diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsionalitas menghasilkan keluaran sesuai dengan yang diharapkan oleh penulis.

### **4.5 Implementasi**

Tahap implementasi sistem merupakan tahap penerjemahan perancangan berdasarkan hasil analisis kedalam suatu Bahasa pemrograman tertentu. Pada bagian ini akan dijelaskan spesifikasi perangkat keras dan perangkat lunak dimana program tersebut diimplementasikan.

#### **4.5.1. Implementasi Perangkat Lunak**

Perangkat lunak yang digunakan pada sistem yang digunakan untuk membangun program adalah sebagai berikut :

##### **1. Sistem Operasi**

Sistem operasi yang digunakan saat melakukan pengembangan program adalah *Microsoft window 10 pro*.

##### **2. *Integrated Development Envaironment (IDE)***

IDE yang digunakan untuk membuat program ini yaitu Notepad ++ yang memiliki fasilitas untuk mengembangkan Bahasa pemrograman berbasis *Website*.

### 3. Browser

Browser yang digunakan untuk menguji dan menjalankan program adalah *google Chrome*.

### 4. XAMPP

XAMPP merupakan *software bundle* yang didalamnya sudah ada beberapa *software* yang digunakan untuk pengembangan aplikasi berbasis server *side*. XAMPP digunakan dalam pengembangan aplikasi ini karena memiliki fitur *web server Apache* dan database *Mysql* yang dapat digunakan sebagai pengelolaan database dalam pembangunan aplikasi.

#### 4.5.2. Implementasi perangkat keras

Implementasi perangkat keras akan menjelaskan kebutuhan minimum perangkat keras yang diperlukan untuk menjalankan atau mengimplementasikan aplikasi yang dibuat. Berikut akan dijelaskan spesifikasi minimum perangkat keras yang bisa digunakan untuk mendukung aplikasi ini. Tetapi semakin tinggi spesifikasi perangkat keras yang digunakan maka akan semakin baik dalam menjalankan aplikasi. Kebutuhan minimumnya yaitu :

##### 1. Komputer/Laptop

Komputer digunakan penyusun untuk pembuatan program (*coding*). Adapun komputer yang digunakan adalah dengan spesifikasi berikut :

- a) *Processor* : kecepatan minimum 1.8 GHZ
- b) *Memory / RAM* : *minimum* 2 MB

- c) *Harddisk* : minimum kapasitas 50 GB
- d) VGA : dengan kecepatan minimum 32 MB
- e) Modem : minimal jaringan 3G

## 2. *Smartphone*

*Smartphone* digunakan untuk mengakses secara *mobile* dengan menggunakan *browser*. *Smartphone* harus memiliki spesifikasi minimal yaitu :

- a) Mendukung dalam menjalankan bahasa pemrograman berbasis *Web* (PHP dan HTML).
- b) Mampu terkoneksi dengan jaringan dengan jaringan minimal 3G.

### 4.5.3. Implementasi Basis Data

Implementasi basis data dilakukan dengan menggunakan bahasa SQL, dimana aplikasi database yang digunakan adalah MySQL. Pada pengimplementasian basis data ini ditulis sintaks sebagai berikut;

#### 1. Sintaks pembuatan database

```
CREATE DATABASE IF NOT EXIST 'db_pelayanan_klinik' DEFAULT
CHARACTER SET latin1 COLLATE latin1_general_ci;
USE 'db_rmrj';
```

#### 2. Sintaks pembuatan tabel admin

```
CREATE TABLE IF NOT EXISTS `tbl_admin` (
`ID_Admin` int(10) NOT NULL,
```

```

`Nama_Lengkap` varchar(50) NOT NULL,
`Alamat` varchar(100) NOT NULL,
`No_Telepon` varchar(30) NOT NULL,
`Email` varchar(100) NOT NULL,
`Level` enum('Admin','Super Admin') NOT NULL,
`Status_Aktif` enum('Y','N') NOT NULL,
`username` varchar(50) NOT NULL,
`password` varchar(50) NOT NULL,
`Photo` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

### 3. Sintaks pembuatan tabel dokter

```

CREATE TABLE IF NOT EXISTS `tbl_dokter` (
`ID_Dokter` varchar(10) NOT NULL,
`Nama_Lengkap` varchar(100) NOT NULL,
`Alamat` varchar(200) NOT NULL,
`No_Telepon` varchar(30) NOT NULL,
`Gender` enum('Laki-laki','Perempuan') NOT NULL,
`Kode_Spesialis` int(11) NOT NULL,
`Kode_Poliklinik` int(11) NOT NULL,
`Tarif_Dokter` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

## 4. Sintaks pembuatan tabel poliklinik

```
CREATE TABLE IF NOT EXISTS `tbl_poliklinik` (
  `Kode_Poliklinik` varchar(10) NOT NULL,
  `Nama_Poliklinik` varchar(50) NOT NULL,
  `Biaya_MasukPoliklinik` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;
```

## 5. Sintaks pembuatan tabel spesialis

```
CREATE TABLE IF NOT EXISTS `tbl_spesialis` (
  `Kode_Spesialis` varchar(10) NOT NULL,
  `Spesialis` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;
```

## 6. Sintaks pembuatan tabel pasien

```
CREATE TABLE IF NOT EXISTS `tbl_pasien` (
  `Kode_Pasien` varchar(10) NOT NULL,
  `Jenis_Pasien` enum('BPJS','Umum') NOT NULL,
  `Kartu_Identitas` enum('KTP','SIM') NOT NULL,
  `No_Identitas` int(30) NOT NULL,
  `Nama_Lengkap` varchar(50) NOT NULL,
  `Nama_Ibu` varchar(50) NOT NULL,
```

```

`Nama_Ayah` varchar(50) NOT NULL,
`Alamat` varchar(100) NOT NULL,
`Gender` enum('Laki-laki','Perempuan') NOT NULL,
`Tgl_Lahir` date NOT NULL,
`No_Telepon` varchar(30) NOT NULL,
`Waktu_InputData` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

#### 7. Sintaks pembuatan tabel pendaftaran

```

CREATE TABLE IF NOT EXISTS `tbl_pendaftaran` (
`Kode_Pendaftaran` varchar(10) NOT NULL,
`Tgl_Mendaftar` date NOT NULL,
`Kode_Pasien` varchar(10) NOT NULL,
`Kode_Poliklinik` varchar(10) NOT NULL,
`ID_Dokter` varchar(10) NOT NULL,
`Biaya_Pendaftaran` int(11) NOT NULL,
`Jumlah_Bayar` int(11) NOT NULL,
`Keterangan` enum('Lunas','Belum Lunas') NOT NULL,
`Waktu_InputData` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

#### 8. Sintaks pembuatan tabel kunjungan

```

CREATE TABLE IF NOT EXISTS `tbl_kunjungan` (
  `Kode_Kunjungan` varchar(10) NOT NULL,
  `Tgl_Kunjungan` date NOT NULL,
  `Kode_Pasien` varchar(10) NOT NULL,
  `Kode_Poliklinik` varchar(10) NOT NULL,
  `ID_Dokter` varchar(10) NOT NULL,
  `Biaya_Kunjungan` int(11) NOT NULL,
  `Jumlah_Bayar` int(11) NOT NULL,
  `Waktu_InputData` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

#### 9. Sintaks pembuatan tabel tindakan

```

CREATE TABLE IF NOT EXISTS `tbl_tindakan` (
  `Kode_Tindakan` int(10) NOT NULL,
  `Kode_Pasien` varchar(10) NOT NULL,
  `Kode_Poliklinik` varchar(10) NOT NULL,
  `ID_Dokter` varchar(10) NOT NULL,
  `Keluhan` text NOT NULL,
  `Tindakan` text NOT NULL,
  `Waktu_InputData` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10;

```

## 10. Sintaks pembuatan tabel obat

```

CREATE TABLE IF NOT EXISTS `tbl_obat` (
  `Kode_Obat` `int`(10) NOT NULL,
  `Nama_Obat` `varchar`(100) NOT NULL,
  `Jenis_Obat` `enum`('Sirup','Kapsul','Tablet') NOT NULL,
  `Kategori_Obat` `enum`('Herbal','Tradisional') NOT NULL,
  `Dosis_Obat` `int`('10') NOT NULL,
  `Harga _Obat` `int`('10') NOT NULL,
  `Stock_Obat` `int`('10') NOT NULL,
  `Deskripsi` `varchar`('200') NOT NULL,
  `Waktu_InputData` `datetime` NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

```

## 11. Sintaks pembuatan tabel resep

```

CREATE TABLE IF NOT EXISTS `tbl_resep` (
  `Kode_Resep` `int`(10) NOT NULL,
  `Tgl_Resep` `date` NOT NULL,
  `Kode_Pasien` `varchar`(10) NOT NULL,
  `Kode_Poliklinik` `varchar`(10) NOT NULL,
  `ID_Dokter` `varchar`(10) NOT NULL,
  `Biaya_Resep` `int`(11) NOT NULL,
  `Keterangan` `enum`('Lunas','Belum Lunas') NOT NULL,

```



```
` Waktu_InputData ` datetime NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10;
```

## 12. Sintaks pembuatan tabel resep detail

```
CREATE TABLE IF NOT EXISTS `tbl_resepdetail` (
```

```
`Kode_Resep` int(10) NOT NULL,
```

```
`Kode_Obat` int(10) NOT NULL,
```

```
`Harga_Obat` int(11) NOT NULL,
```

```
`Dosis_Obat` int(10) NOT NULL,
```

```
`Sub_Total` int(10) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10;
```

## 13. Sintaks pembuatan tabel pembayaran

```
CREATE TABLE IF NOT EXISTS `tbl_pembayaran` (
```

```
`Kode_Pembayaran` int(10) NOT NULL,
```

```
`Tgl_Pembayaran` date NOT NULL,
```

```
`Biaya_Kunjungan` int(11) NOT NULL,
```

```
`Biaya_Resep` int(11) NOT NULL,
```

```
`Jumlah_Bayar` int(11) NOT NULL,
```

```
`Bayar` int(11) NOT NULL,
```

```
`Kembali` int(11) NOT NULL,
```

```
`Keterangan` enum('Lunas','Belum Lunas') NOT NULL,
```

```
`Waktu_InputData` date NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;
```

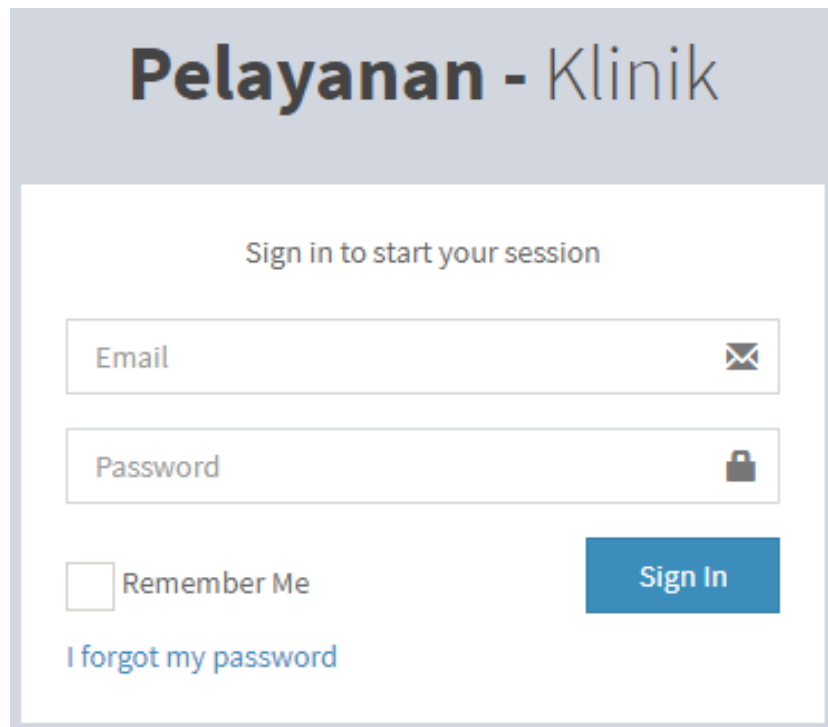
#### 4.5.4. Implementasi Antar Muka

Implementasi antar muka dilakukan dengan setiap halaman yang dibuat.

Berikut ini adalah beberapa tampilan antar muka yang telah dihasilkan.


##### 1. Halaman *Login* Untuk *User* dan *Admin*


Tampilan halaman *login* yang akan digunakan untuk masuk kemenu utama admin atau *user*.



**Pelayanan - Klinik**

Sign in to start your session

Email 

Password 

Remember Me

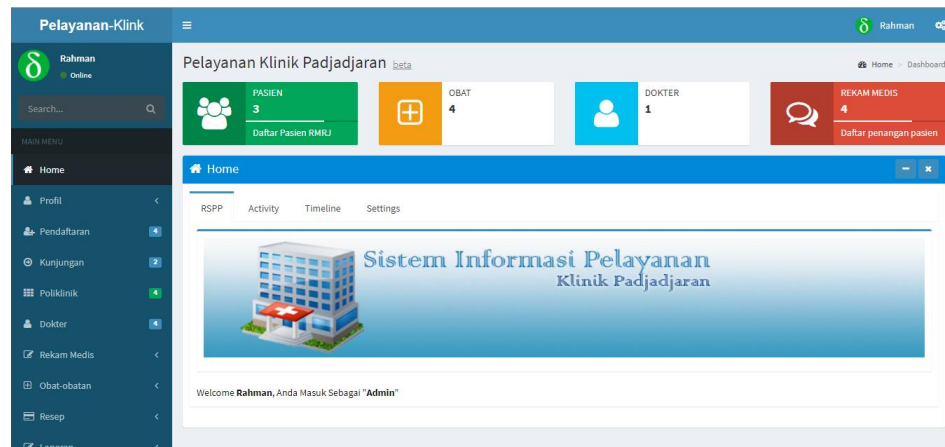
[I forgot my password](#)

**Sign In**

**Gambar 4.45** Halaman *Login* User dan Admin

## 2. Halaman Utama Admin

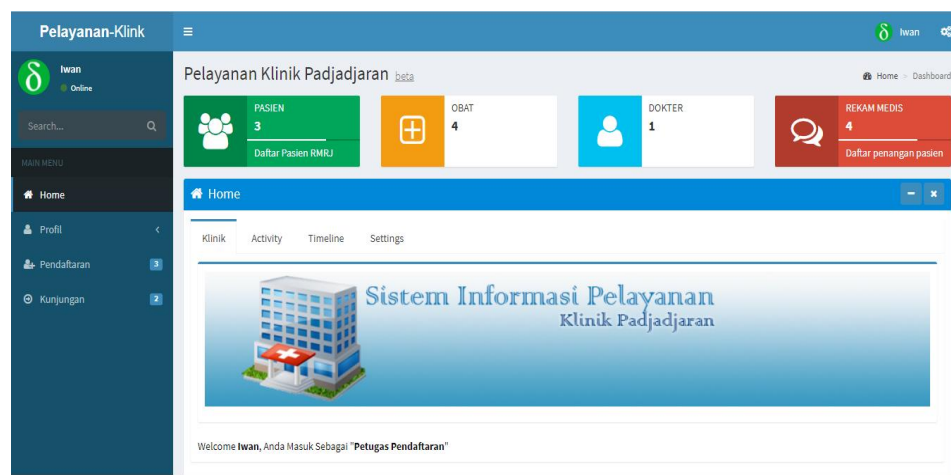
Tampilan halaman admin yang akan digunakan untuk mengakses menu sesuai hak akses admin yang telah disediakan.



Gambar 4.46 Halaman Utama Admin

## 3. Halaman Utama Petugas Pendaftaran

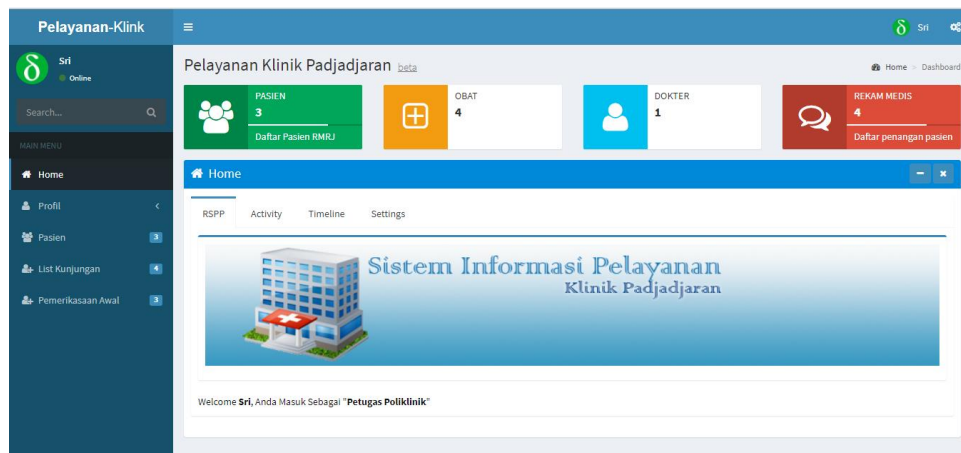
Tampilan halaman petugas pendaftaran yang akan digunakan untuk mengakses menu sesuai hak akses petugas pendaftaran yang telah disediakan.



Gambar 4.47 Halaman Utama Petugas Pendaftaran

#### 4. Halaman Utama Petugas Poliklinik

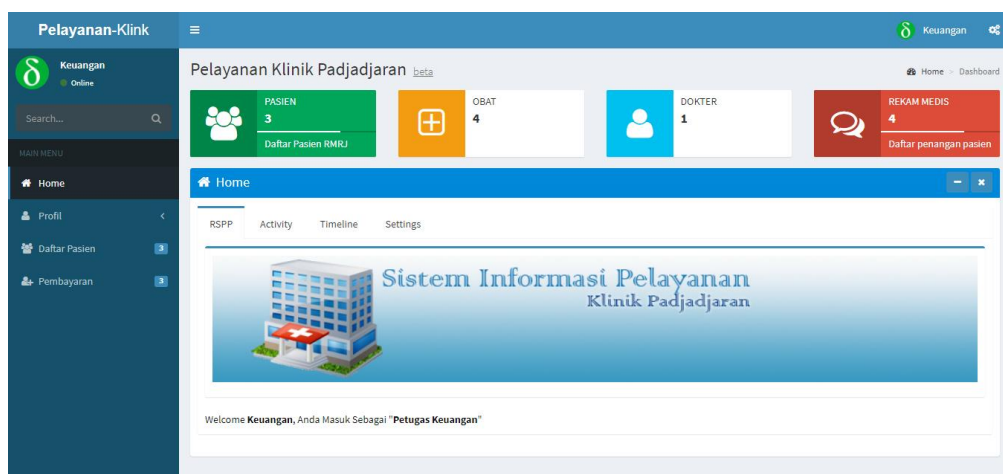
Tampilan halaman petugas poliklinik yang akan digunakan untuk mengakses menu sesuai hak akses petugas poliklinik yang telah disediakan.



Gambar 4.48 Halaman Utama Petugas Poliklinik

#### 5. Halaman Utama Petugas Keuangan

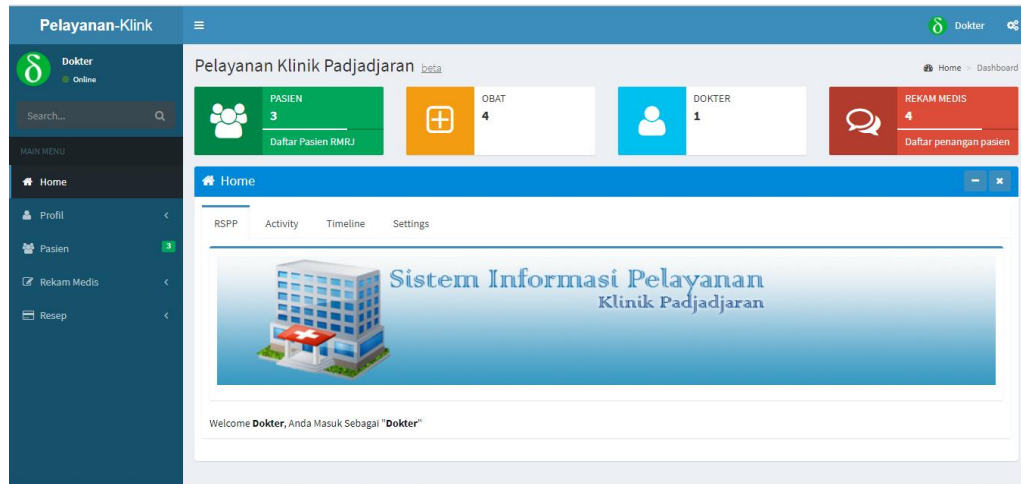
Tampilan halaman petugas keuangan yang akan digunakan untuk mengakses menu sesuai hak akses petugas keuangan yang telah disediakan.



Gambar 4.49 Halaman Utama Petugas Keuangan

## 6. Halaman Utama Dokter

Tampilan halaman dokter yang akan digunakan untuk mengakses menu sesuai hak akses dokter yang telah disediakan.

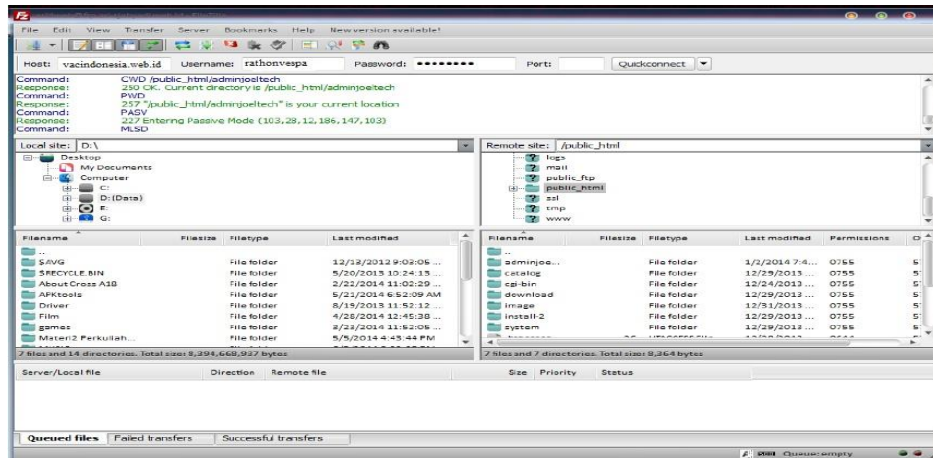


**Gambar 4.50 Halaman Utama Dokter**

### 4.5.5. Implementasi instalasi program

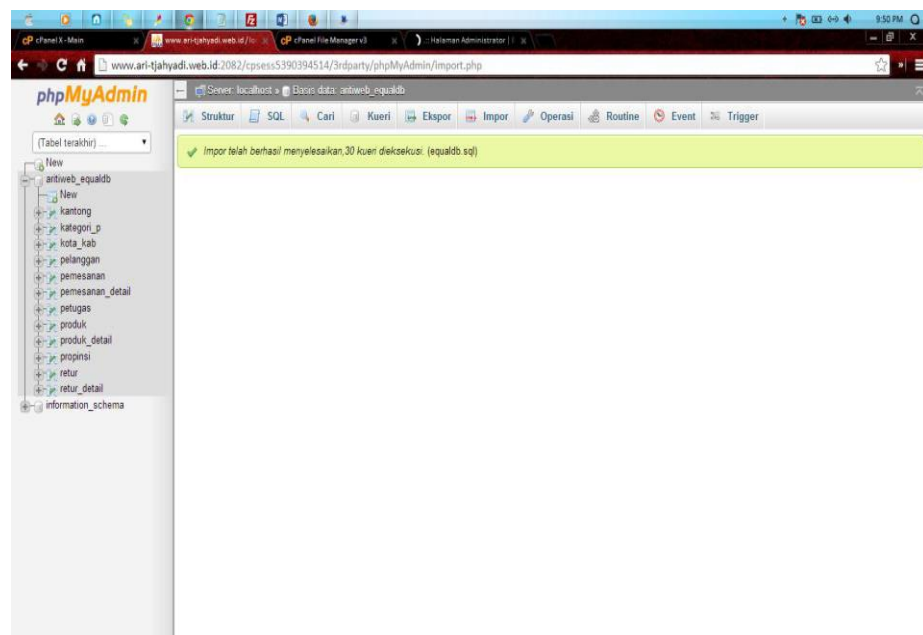
Berikut ini merupakan implementasi instalasi program bagaimana cara menggunakan program Sistem Informasi pelayanan klinik padjajaran, langkah-langkahnya adalah sebagai berikut:

1. Kompres file aplikasi web kedalam bentuk.zip, kemudian upload file aplikasi web yang telah berbentuk.zip dengan membuka program filezilla dan masukan nama *host*, *username*, *password*, dan *port*. Berikut adalah gambar jelasnya sebagai berikut:



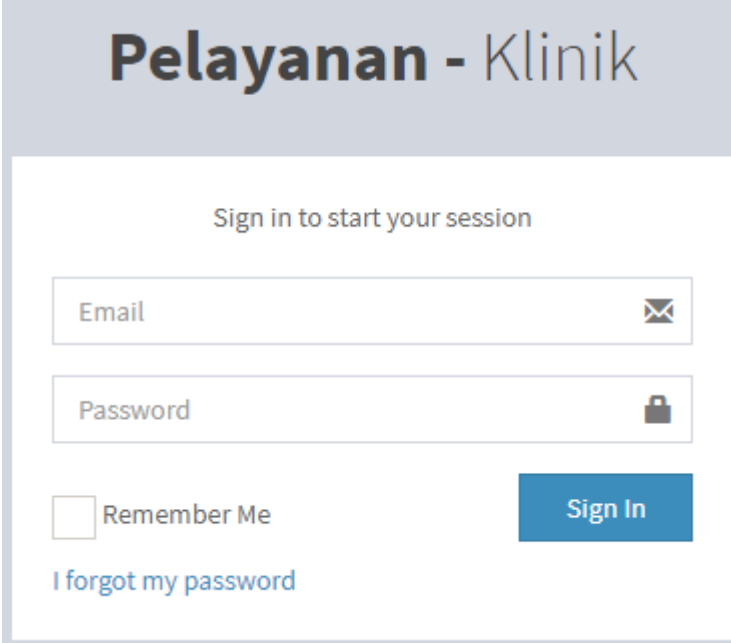
**Gambar 4.89** Upload Aplikasi Web Server ke Hosting

- Setelah Upload file aplikasi web nya ke *server hosting*, *setting username* dan *password* untuk *database* website nya di *server hosting* setelah itu *import* file *database.sql* ke *phpmyadmin* di *server hosting*, berikut adalah gambar setelah *database.sql* berhasil di *import*:



**Gambar 4.90** Pengaturan Basis Data di Web Hosting

3. Setelah *database.sql* berhasil di *import* ketikkan alamat website di browser, seperti gambar tampilan halaman login admin sistem informasi pelayanan klinik padjadjaran setelah di hosting:



The image shows a web browser window displaying the admin login page for a system titled "Pelayanan - Klinik". The page has a light gray header with the title in a large, bold, dark font. Below the header, the text "Sign in to start your session" is centered. There are two input fields: "Email" with an envelope icon and "Password" with a lock icon. Below the "Email" field is a checkbox labeled "Remember Me". To the right of the "Remember Me" checkbox is a blue button labeled "Sign In". Below the "Remember Me" checkbox is a link that says "I forgot my password".

**Gambar 4.91** Halaman Login Admin Setelah Upload ke Server Hosting