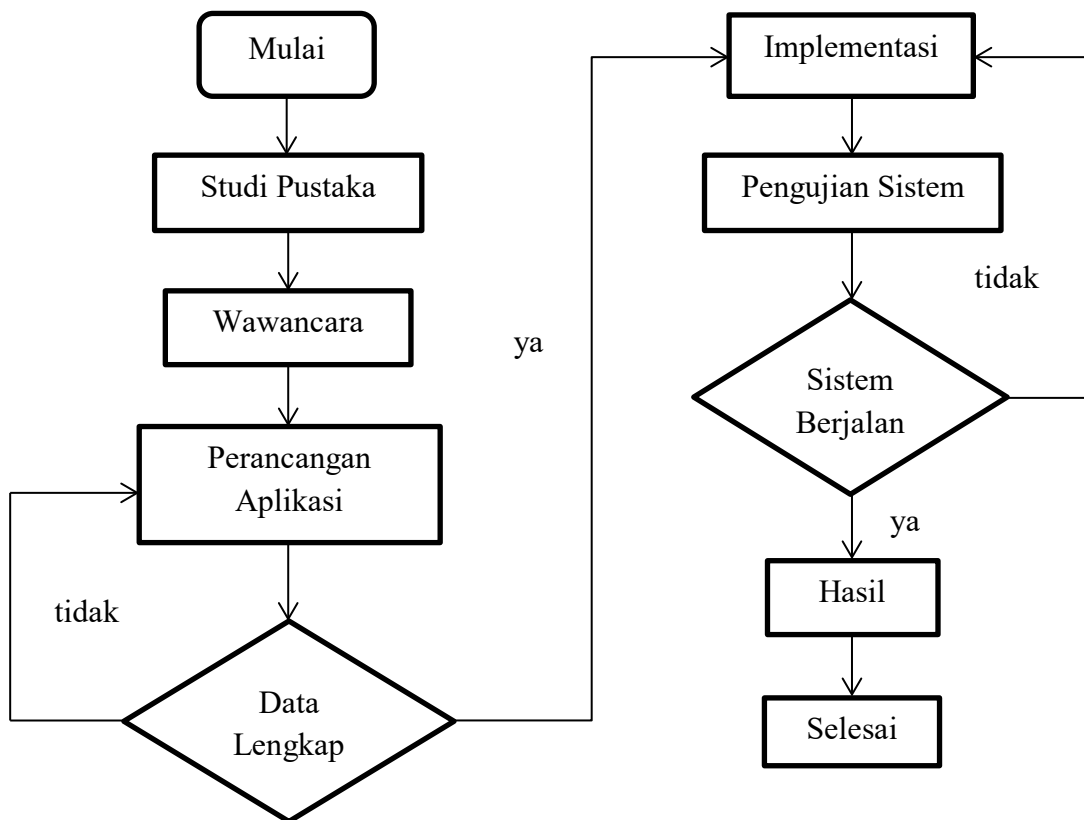


BAB III

METODELOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian dilakukan oleh peneliti untuk mempermudah dalam melakukan penelitian. Desain penelitian digambarkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk memperoleh data-data yang dibutuhkan dalam penelitian ini adalah:

1. Wawancara

Melakukan wawancara dengan dokter spesialis anak yang terdapat pada RSUD Dr. H. Abdul Moeloek. Wawancara dilakukan dengan tujuan untuk memperoleh data, informasi dan keterangan-keterangan tentang objek

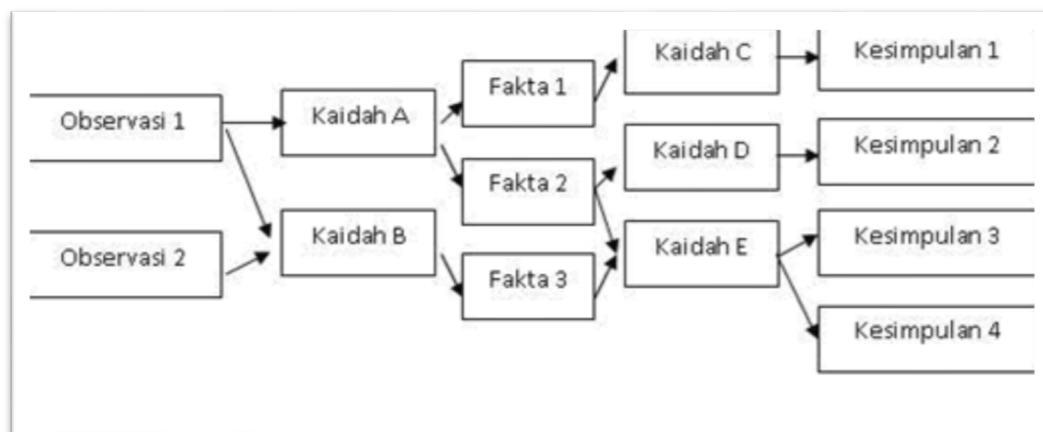
penelitian yang dipilih. Dari proses wawancara ini akan diperoleh data-data apa saja yang diperlukan oleh subjek penelitian dan sistem seperti apa yang diinginkan oleh pengguna.

2. Studi Pustaka

Dilakukan dengan membaca buku-buku, mengutip dan membangun catatan yang bersumber pada bahan-bahan pustaka yang mendukung dan berkaitan dengan penelitian ini. Selanjutnya dengan cara mempelajari dan memahami jurnal-jurnal yang bersumber dari internet yang berhubungan dengan masalah yang akan di bahas dalam karya ilmiah ini.

3.3 Metode *Forward Chaining*

Metode *forward chaining* adalah suatu metode pengambilan keputusan yang umum digunakan dalam *system* pakar. Proses pencarian dengan metode *forward chaining* berangkat dari kiri ke kanan, yaitu dari premis menuju kepada kesimpulan akhir, metode ini sering disebut *data driven* yaitu pencarian dikendalikan oleh data yang diberikan. (Hartati & Iswanti 2008) dalam (Ismail, 2017).



Gambar 3.2 Rumus Metode *Forward Chaining* (Jamkhandi & Disouza, 2012)

Pemecahan masalah pada *forward chaining* dimulai dari fakta masalah yang diberikan dan kumpulan langkah-langkah sah atau aturan-aturan untuk berpindah *state*. Tanda-tanda atau kunci keberhasilan akan terkumpul dengan sendirinya ketika mulai mengarah ke kesimpulan (Sijabat, 2018).

3.4 Metode Pengembangan Perangkat Lunak

Metode yang di gunakan dalam mengembangkan perangkat lunak ini adalah model *prototype*. Model *prototype* memiliki 5 buah tahapan yaitu sebagai berikut.

3.4.1 *Communication*

Dalam penelitian ini, tahap *communication* adalah suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya dan terjadwalkan data-data yang didapat saat penelitian di RSUD Dr. H. Abdul Moeloek. Adapun langkah-langkah metode ilmiah yaitu sebagai berikut:

1. Merumuskan Masalah

Tahapan ini merupakan langkah yang pertama. Merumuskan masalah bertujuan untuk memperjelas masalah dengan mengajukan beberapa atau serangkaian pertanyaan terhadap masalah yang ada.

2. Melakukan Penyusunan Rencana Penelitian

Langkah kedua adalah penyusunan rencana. Rencana penelitian dibuat dengan membuat tujuan penelitian agar rencana penelitian lebih jelas. Hal ini tentu saja dilakukan dengan membuat tinjauan pustaka sehingga diperoleh data-data yang berhubungan dan metode penelitian yang akan dilakukan.

3. Melakukan Penelitian

Langkah yang ketiga adalah melakukan penelitian. Langkah ini merupakan langkah metode ilmiah yang dilakukan setelah rencana penelitian atau proposal telah diterima. Penelitian sendiri tergantung pada langkah penelitian atau metodologi penelitian yang akan digunakan.

3.4.2 *Quick Plan*

Pada tahap *quick plan* ini dilakukan untuk menetapkan bagaimana perangkat lunak akan di operasikan. Hal ini berkaitan untuk menentukan perangkat keras, perangkat lunak, tampilan program dan *form-form* yang akan dipakai dalam pembuatan *prototype*.

3.4.2.1 Analisis dan Definisi Persyaratan

Adapun analisis kebutuhan perangkat lunak dan perangkat keras yang di gunakan untuk membangun aplikasi ini adalah sebagai berikut.

1. Analisis *Software*

Adapun *software* yang di gunakan untuk membantu dalam pembuatan aplikasi ini yaitu sebagai berikut:

- a. Sistem operasi *Microsoft Windows 8.1 Professional*.
- b. *Adobe Dreamweaver*, di gunakan untuk membuat *web*.
- c. XAMPP, di gunakan sebagai *server*.

2. Analisis *Hardware*

Adapun *hardware* yang di gunakan dalam pembuatan aplikasi yang akan di buat, yaitu sebagai berikut:

- a. *Processor: Intel(R) Celeron(R)*.
- b. *Installed memory (RAM): 2,00 GB*.

3.4.2.2 Skema Representasi

1. Kebutuhan Data

Kebutuhan data dalam membangun aplikasi sistem pakar yaitu menggunakan akuisisi pengetahuan. Akuisisi Pengetahuan adalah suatu proses untuk mengumpulkan data dari pakar yang di perlukan untuk memahami, merumuskan, dan memecahkan persoalan. Setelah data terkumpul, barulah bisa menentukan penyakit, gejala-gejala yang di alami, basis aturan, dan pohon keputusan.

a. Penyakit

Data penyakit dapat di lihat pada tabel 3.1 sebgai berikut.

Tabel 3.1 Data Penyakit

Kode	Penyakit
P01	<i>Rubella</i>

b. Gejala

Berikut merupakan gejala-gejala yang ada pada *rubella*. Dapat di lihat pada tabel 3.2 sebagai berikut.

Tabel 3.2 Data Gejala

Kode	Gejala
G001	Demam
G002	Sakit Kepala
G003	Hidung tersumbat atau pilek
G004	Tidak nafsu makan
G005	Mata merah
G006	Pembengkakan kelenjar limfa pada telinga dan leher
G007	Ruam berbentuk bintik-bintik kemerahan
G008	Nyeri sendi (terutama pada remaja wanita)

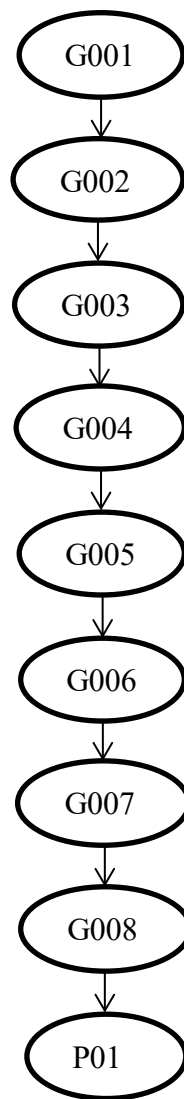
c. Basis Aturan

Tabel 3.3 Aturan (*rule*)

Kode	Keterangan
P01	<i>Rubella</i>
Gejala	G001, G002, G003, G004, G005, G006, G007, G008
Definisi	<i>Rubella</i> atau di sebut juga campak jerman, yaitu campak 3 hari yang di sebabkan oleh virus <i>rubella</i> .
Solusi	Segera berobat atau konsultasi dengan dokter.

d. Pohon Keputusan

Pohon keputusan di gunakan sebagai alat pendukung keputusannya. Setelah mendapatkan data penyakit dan gejala-gejalanya, maka dapat di tentukan pohon keputusannya yaitu sebagai berikut.



Gambar 3.3 Pohon Keputusan

3.4.3 Modelling

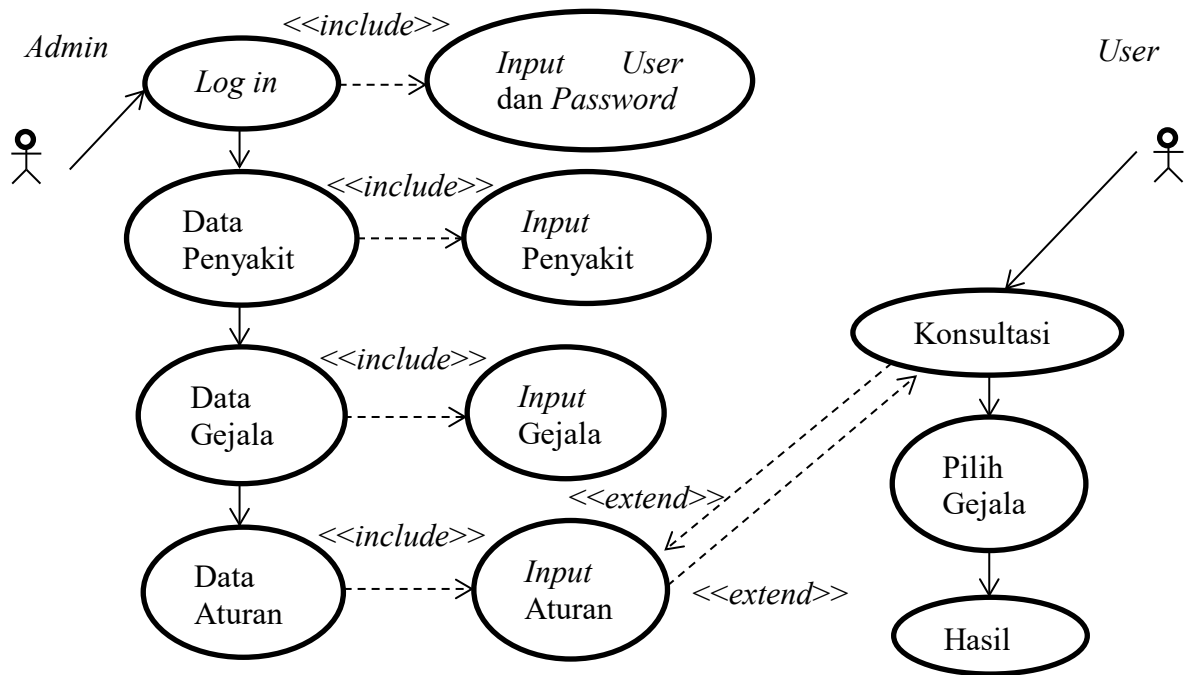
3.4.3.1 Unified Modelling Language (UML)

Adapun penerapan *Unified Modelling Language* (UML) untuk perancangan desain *inteface*, yaitu sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang di harapkan dari sebuah sistem. Adapun diagram di bawah ini yang menunjukkan fungsi

sebuah sistem, bagaimana sistem tersebut dapat berinteraksi dengan pengguna (*user*).



Gambar 3.4 Use Case Diagram

- a) Nama *use case*: Menu konsultasi.

Actor: User (pengguna).

Tujuan: Untuk berkonsultasi.

Deskripsi: Pada menu ini, *user* akan di beri pilihan “ya atau tidak” tentang gejala-gejala *rubella* yang di tampilkan pada aplikasi ini.

- b) Nama *use case*: Menu hasil.

Actor: User (pengguna).

Tujuan: Untuk mendapatkan informasi.

Deskripsi: Pada menu hasil, akan di tampilkan hasil *user* terkena *rubella* atau tidak serta menampilkan solusi untuk *user*.

- c) Nama *use case*: Menu *log in*.

Actor: Admin (pengelola).

Tujuan: Untuk mengantarkan *admin* menuju halaman *admin* yang ada di dalam aplikasi diagnosis *rubella*.

Deskripsi: Pada menu ini, *admin* memasukkan *username* dan *password* agar bisa berada di dalam halaman *admin*.

- d) Nama *use case*: Menu data penyakit.

Actor: *Admin* (pengelola).

Tujuan: Untuk memasukkan nama-nama penyakit.

Deskripsi: Pada menu ini, *admin* akan menginput nama penyakit yaitu penyakit *rubella* untuk di tampilkan di halaman *user*.

- e) Nama *use case*: Menu data gejala.

Actor: *Admin* (pengelola).

Tujuan: Untuk memasukkan gejala-gejala.

Deskripsi: Pada menu ini, *admin* akan menginput gejala-gejala yang berhubungan dengan *rubella*.

- f) Nama *use case*: Menu data aturan (*rule*).

Actor: *Admin* (pengelola).

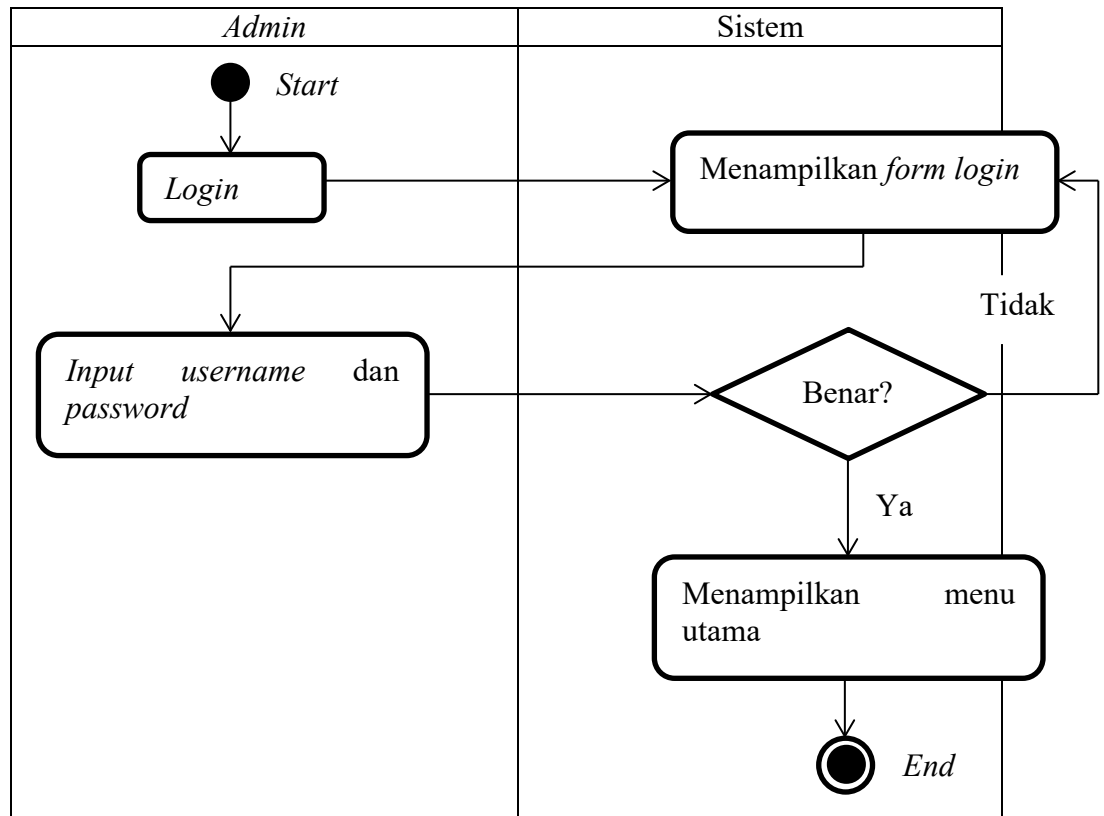
Tujuan: Untuk menentukan aturan-aturan yang akan di gunakan dalam aplikasi ini.

Deskripsi: Pada menu *rule* ini, *admin* akan menginput *rule-rule* yang akan di gunakan untuk berkonsultasi dalam aplikasi ini.

2. *Activity Diagram*

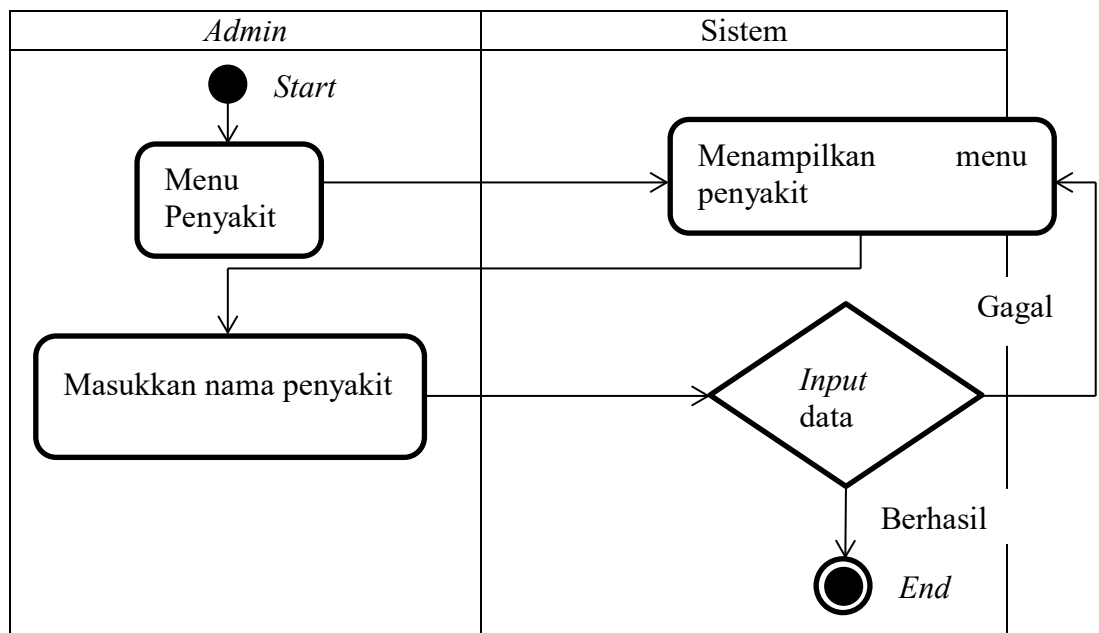
Suatu diagram aktivitas menggunakan sebuah kotak berisi lengkung untuk menggambarkan fungsi tertentu yang ada dalam suatu sistem atau perangkat lunak yang akan di kembangkan, sementara itu tanda panah menggambarkan aliran di dalam sistem atau perangkat lunak, bentuk intan di gunakan untuk menggambarkan keputusan – keputusan bercabang (masing – masing tanda panah yang memancar dari tanda intan di beri label), dan garis horizontal tebal mengindikasikan bahwa aktivitas – aktivitas paralel di dalam sistem atau perangkat lunak sedang terjadi.

a) Activity diagram ketika admin melakukan login



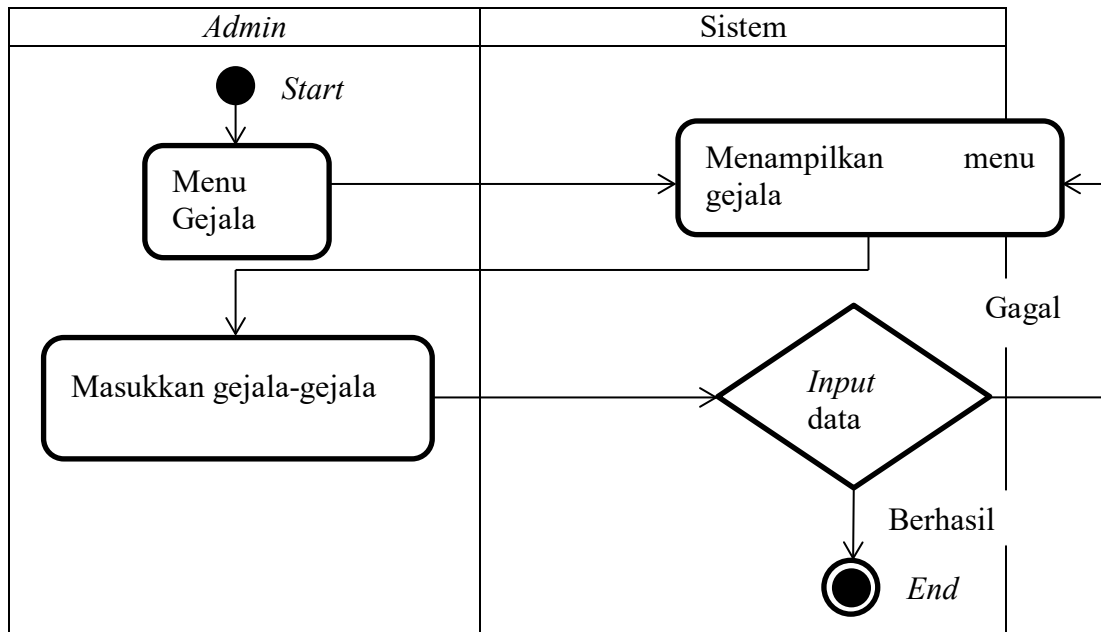
Gambar 3.5 Activity Diagram Login

b) Activity diagram ketika admin menginput data penyakit



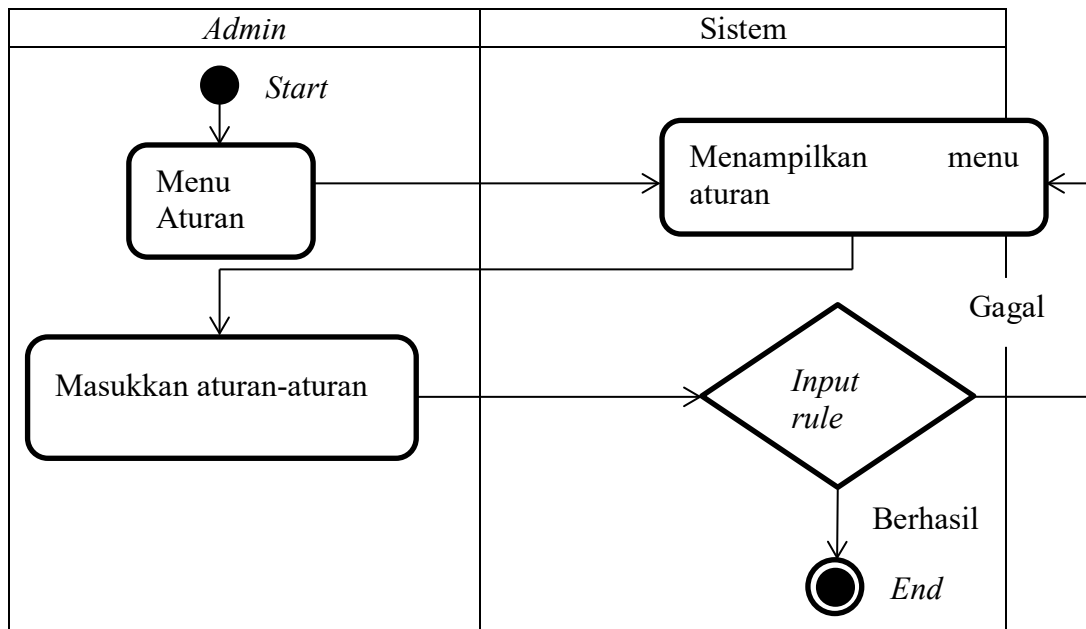
Gambar 3.6 Activity Diagram Menu Penyakit

c) *Activity diagram ketika admin menginput gejala*



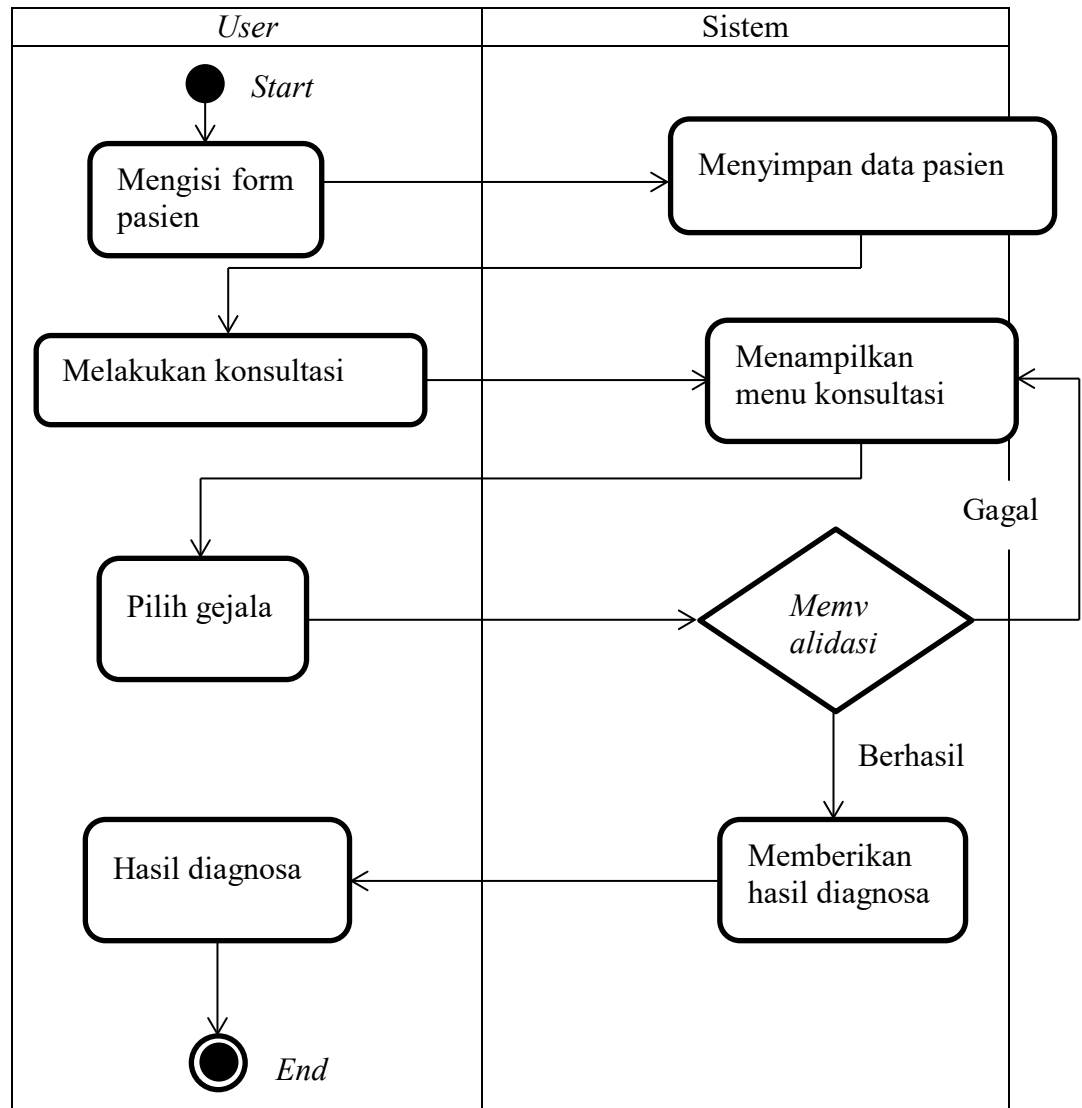
Gambar 3.7 Activity Diagram Menu Gejala

d) *Activity diagram ketika admin menginput rule (aturan)*



Gambar 3.8 Activity Diagram Menu Aturan (Rule)

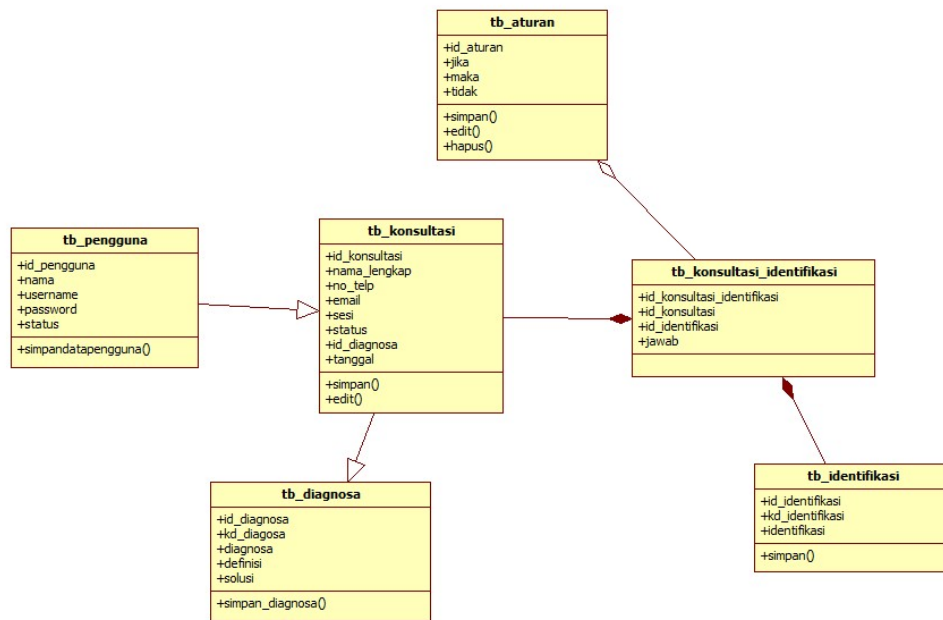
e) *Activity diagram* ketika *user* melakukan konsultasi



Gambar 3.9 *Activity Diagram* Konsultasi

3. *Class Diagram*

Class diagram menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. *Class diagram* dapat di lihat pada gambar berikut ini.



Gambar 3.10 Susunan *Class Diagram*

3.4.3.2 Struktur Database

Struktur *database* dari sistem pakar dalam mendiagnosa penyakit *rubella* adalah sebagai berikut.

1. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_pengguna.

Fungsi: Mengelola dan menyimpan data admin.

Primary key: id_pengguna.

Struktur *database* tb_pengguna dapat di lihat pada tabel 3.4 berikut ini.

Tabel 3.4 Tabel Pengguna

No.	Field	Type	Lenght	Constrant
1.	id_pengguna	int	11	Primary key
2.	Nama	varchar	50	
3.	Username	varchar	20	
4.	Password	varchar	50	
5.	status	int	1	

2. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_identifikasi.

Fungsi: Untuk mengidentifikasi gejala-gejala apa saja yang ada pada *rubella*.

Primary key: id_identifikasi.

Struktur *database* tb_identifikasi dapat di lihat pada tabel 3.5 berikut ini.

Tabel 3.5 Tabel Identifikasi

No.	Field	Type	Lenght	Constrant
1.	id_identifikasi	int	11	Primary key
2.	kd_identifikasi	char	5	
3.	identifikasi	varchar	150	

3. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_aturan.

Fungsi: Menyimpan aturan atau *rule* yang di *inputkan* oleh *admin*.

Primary key: id_aturan.

Struktur *database* tb_aturan dapat di lihat pada tabel 3.6 berikut ini.

Tabel 3.6 Tabel Aturan

No.	Field	Type	Lenght	Constrant
1.	id_aturan	int	11	Primary key
2.	Jika	varchar	5	
3.	Maka	varchar	5	
4.	Tidak	varchar	5	

4. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_konsultasi.

Fungsi: Menyimpan data pasien.

Primary key: id_konsultasi.

Struktur *database* tb_konsultasi dapat di lihat pada tabel 3.7 berikut ini.

Tabel 3.7 Tabel Konsultasi

No.	Field	Type	Lenght	Constrant
1.	id_konsultasi	int	11	Primary key
2.	nama_lengkap	varchar	50	
3.	no_telp	varchar	15	
4.	Email	varchar	50	
5.	Sesi	varchar	15	
6.	Status	int	1	
7.	id_diagnosa	int	11	
8.	Tanggal	datetime	-	

5. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_konsultasi_identifikasi.

Fungsi: Untuk mengidentifikasi data pasien yang berkonsultasi.

Primary key: id_konsultasi_identifikasi.

Struktur *database* tb_konsultasi_identifikasi dapat di lihat pada tabel 3.8 berikut ini.

Tabel 3.8 Tabel Konsultasi Identifikasi

No.	Field	Type	Lenght	Constrant
1.	id_konsultasi_identifikasi	int	11	Primary key
2.	id_konsultasi	int	11	
3.	id_identifikasi	int	11	
4.	Jawab	int	1	

6. Nama *database*: adev_sc_sp_forward_chaining.

Nama tabel: tb_diagnosa.

Fungsi: Untuk mendiagnosa pasien terkena *rubella* atau tidak.

Primary key: id_diagnosa.

Struktur *database* tb_diagnosa dapat di lihat pada tabel 3.9 berikut ini.

Tabel 3.9 Tabel Diagnosa

No.	Field	Type	Lenght	Constrant
1.	id_diagnosa	Int	11	Primary key
2.	kd_diagnosa	Char	5	
3.	Diagnosa	varchar	100	
4.	Definisi	Text	-	
5.	Solusi	Text	-	

3.4.4 Construction

Tahapan *construction* pada penelitian ini merupakan pembuatan skrip *coding*.

3.4.4.1 Rancangan *Input Output*

Proses ini menghasilkan sebuah arsitektur perangkat lunak sehingga dapat diterjemahkan ke dalam kode-kode program. Berikut merupakan rancangan-rancangan *interface* dari aplikasi pendiagnosa *rubella* yaitu sebagai berikut.

a) Menu *login* (*admin*)

Keterangan
<div style="border: 1px solid black; width: 80%; margin: 10px auto; padding: 5px;"><i>Username</i></div> <div style="border: 1px solid black; width: 80%; margin: 10px auto; padding: 5px;"><i>Password</i></div> <div style="border: 1px solid black; width: 30%; margin: 10px auto; padding: 5px;"><i>Login</i></div>

Gambar 3.11 Rancangan *Interface* Menu *Login*

b) Menu data penyakit (*admin*)

LOGO	Data Diagnosa													
<div style="border: 1px solid black; padding: 5px; width: 80px; margin: 0 auto;">Data Penyakit</div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%; text-align: center; padding: 5px;">Pencarian</td> <td style="width: 20%;"></td> <td style="width: 40%; text-align: center; padding: 5px;"> <div style="border: 1px solid black; padding: 5px; display: inline-block;">Cari</div> </td> </tr> <tr> <td style="text-align: center; padding: 5px;">No</td> <td style="text-align: center; padding: 5px;">Kode</td> <td style="text-align: center; padding: 5px;">Penyakit</td> <td style="text-align: center; padding: 5px;">Definisi</td> <td style="text-align: center; padding: 5px;">+</td> </tr> <tr> <td style="text-align: center; padding: 5px;">1.</td> <td style="text-align: center; padding: 5px;">XXX</td> <td></td> <td></td> <td style="text-align: center; padding: 5px;">Edit hapus</td> </tr> </table>	Pencarian		<div style="border: 1px solid black; padding: 5px; display: inline-block;">Cari</div>	No	Kode	Penyakit	Definisi	+	1.	XXX			Edit hapus
Pencarian		<div style="border: 1px solid black; padding: 5px; display: inline-block;">Cari</div>												
No	Kode	Penyakit	Definisi	+										
1.	XXX			Edit hapus										

Gambar 3.12 Rancangan *Interface* Menu Data Penyakit

c) Menu gejala (*admin*)

LOGO	Data Identifikasi									
Data Gejala	<div>Pencarian <input type="text"/> <input type="button" value="Cari"/></div>									
	<table border="1"> <tr> <th>No</th> <th>Kode</th> <th>Gejala</th> <th>+</th> </tr> <tr> <td>1.</td> <td>XXX</td> <td></td> <td>Edit hapus</td> </tr> </table>			No	Kode	Gejala	+	1.	XXX	
No	Kode	Gejala	+							
1.	XXX		Edit hapus							

Gambar 3.13 Rancangan *Interface* Menu Gejalad) Menu aturan (*admin*)

LOGO	Data Aturan (<i>Rule</i>)								
Aturan	<table border="1"> <tr> <th>No</th> <th>Aturan (<i>rule</i>)</th> <th>+</th> </tr> <tr> <td>1.</td> <td> Jika ... Maka ... Jika tidak maka ... </td> <td>Edit hapus</td> </tr> </table>			No	Aturan (<i>rule</i>)	+	1.	Jika ... Maka ... Jika tidak maka ...	Edit hapus
	No	Aturan (<i>rule</i>)	+						
1.	Jika ... Maka ... Jika tidak maka ...	Edit hapus							

Gambar 3.14 Rancangan *Interface* Menu Aturan (*Admin*)e) Menu konsultasi (*user*)

1.

LOGO	Konsultasi Mulai	
Konsultasi	ISI DATA DIRI!	
	Nama lengkap:	<input type="text"/>
	Nomor telp:	<input type="text"/>
	Email:	<input type="text"/>
	<input type="button" value="Lanjutkan"/>	<input type="button" value="Batal"/>

Gambar 3.15 Rancangan *Interface* Menu Konsultasi (*User*)

2.

LOGO	Konsultasi Identifikasi
Konsultasi	<p>Identifikasi jawaban yang sesuai!</p> <p>001 – Apakah demam?</p> <p><input type="radio"/> YA <input type="radio"/> TIDAK</p> <p>Lanjutkan</p>

Gambar 3.16 Rancangan *Interface* Menu Konsultasi (*User*)

3.

LOGO	Konsultasi Hasil								
Konsultasi	<p>Hasil konsultasi anda!</p> <p>Pemilihan identifikasi anda:</p> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Identifikasi</th> <th>Pilihan</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>XXX</td> <td>Gejala</td> <td>Misal : Ya</td> </tr> </tbody> </table> <p>Diagnosa:</p> <p>Jawaban</p> <p>Selesai</p>	No	Kode	Identifikasi	Pilihan	1.	XXX	Gejala	Misal : Ya
No	Kode	Identifikasi	Pilihan						
1.	XXX	Gejala	Misal : Ya						

Gambar 3.17 Rancangan *Interface* Menu Konsultasi (*User*)

3.4.5 Deployment

Tahapan *deployment* di lakukan setelah semua tahapan dari *communication*, *quick plan*, *modelling*, dan *construction* telah di lakukan. Pada tahap *deployment*, sistem indentifikasi akan di tes semua fungsi-fungsi tombol dan penerapan metode *forward chaining*nya.