

# **SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB**

Novi Rukhviyanti<sup>1</sup>, Danu Syahfrizal<sup>2</sup>

Program Studi Teknik Informatika

STMIK Indonesia Mandiri, Jl. Belitung No. 7 Bandung

Email : [novi.rukhviyanti@stmik-im.ac.id](mailto:novi.rukhviyanti@stmik-im.ac.id), [danusyahfrizal001@gmail.com](mailto:danusyahfrizal001@gmail.com)

## ***Abstract***

*At the moment it is happening where the emergence of a new disease outbreak is very large and has reached all over the world. The outbreak in question is Coronavirus Disease 2019 or commonly referred to as COVID-19. The spread is already very widespread or the whole world is experiencing an outbreak of this new disease, and this new disease is very contagious. Many people are confused as to whether or not he has the disease because it is very contagious. And where today many governments do restrictions on activities outside the home so that people are confused to consult a doctor or to a health clinic. Regarding this new disease many health experts to create a journal announcing this new disease through a trusted health web.*

**Keywords:** *Expert, Infectious Diseases, COVID-19, WEB*

## **Abstrak**

Pada saat ini sedang terjadinya dimana munculnya wabah penyakit baru yang sangat besar dan sudah menjangkau seluruh dunia. Wabah yang dimaksud yaitu adalah Coronavirus Disease 2019 atau yang biasa disebut dengan **COVID-19**. Penyebarannya sudah sangat luas sekali atau seluruh dunia sedang mengalami wabah penyakit baru ini, dan penyakit baru ini sangat mudah menular. Banyak orang mengalami kebingungan apakah dia terkena penyakit tersebut atau tidak sebab penyakit ini sangat mudah sekali menular. Dan dimana sekarang ini banyak pemerintahan melakukan pembatasan kegiatan diluar rumah sehingga orang-orang bingung untuk berkonsultasi ke dokter atau ke klinik kesehatan. Perihal penyakit baru ini banyak pakar-pakar kesehatan untuk membuat jurnal mengumumkan penyakit baru ini melalui web kesehatan yang terpercaya.

**Kata Kunci:** Pakar, Penyakit Menular, **COVID-19, WEB**

## 1. PENDAHULUAN

Pada belakangan ini telah muncul permasalahan baru yang dialami oleh seluruh dunia tidak terkecuali negara Indonesia yaitu merebahnya wabah penyakit menular terbaru yaitu *corona virus disease 2019* atau yang bisa disebut dengan **COVID-19**. Jenis penyakit ini adalah penyakit mudah menular ke sesama manusia, virus ini pertama kali ditemukan atau teridentifikasi di wilayah Wuhan, Tiongkok di pasar hewan Wuhan. Dan sampai saat ini belum ditemukannya vaksin (masih dalam tahap pengembangan) yang dapat menyembuhkan penyakit ini namun gejalanya dapat dirasakan. Tetapi tidak berlaku untuk yang daya tahan imunnya kuat sehingga bisa saja tertular tetapi tidak ada sedikit gejala yang ditimbulkan.

Sistem pakar adalah suatu cabang dari *Artificial Intelligent (AI)* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar menurut (Arhami, 2005). Selama beberapa dekade terakhir, sistem pakar telah menjadi aplikasi praktek yang utama dari riset **AI**. Dewasa ini, ada banyak sistem yang berguna dalam hampir setiap bidang operasional diseluruh dunia. Mulai dari *gadget* sederhana seperti *handphone* sampai robot-robot dalam industri manufaktur dan medis. Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* yaitu suatu aplikasi komputer yang ditunjukan untuk membantu pengambilan keputusan atau pemecahan persoalan dalam bidang yang spesifik. Sistem ini bekerja dengan menggunakan pengetahuan dan metode analisis yang telah didefinisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya (Hayadi, 2018).

Dalam sistem pakar, metode *forward chaining* merupakan salah satu metode penalaran yang digunakan dalam mesin inferensi, dimana metode penalaran ini dimulai dengan data dan alasan untuk menuju pada suatu jawaban atau kesimpulan (Wadi, 2020). Algoritma *forward chaining* adalah satu dari dua metode utama reasoning (pemikiran) ketika menggunakan *inference engine* (mesin pengambil keputusan) dan bisa secara logis dideskripsikan sebagai aplikasi pengulangan dari *modus ponens* (satu set aturan inferensi dan argumen yang valid). Lawan dari *forward chaining* adalah *backward chaining*.

*Website* adalah keseluruhan halaman-halaman **WEB** yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman **WEB** yang saling berhubungan. Hubungan antara satu halaman **WEB** dengan halaman **WEB** lainnya disebut dengan *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

## **2. METODE PENELITIAN**

### **2.1. Teknik Pengumpulan Data**

#### 1. Studi lapangan meliputi

- a. Observasi langsung yaitu mengadakan pengamatan secara langsung bagaimana objek pada rute yang akan di tentukan.
- b. Wawancara secara tidak langsung yaitu teknik pengumpulan data dengan cara berbicara secara tidak langsung kepada pihak-pihak tertentu dengan menggunakan aplikasi resmi tentang yang berkaitan.

#### 2. Studi literatur

Pada tahap ini dilakukan studi literatur tentang informasi terkait dengan melakukan studi kepustakaan terhadap referensi yang berkaitan dengan penelitian.

### **2.2. Metode Perancangan**

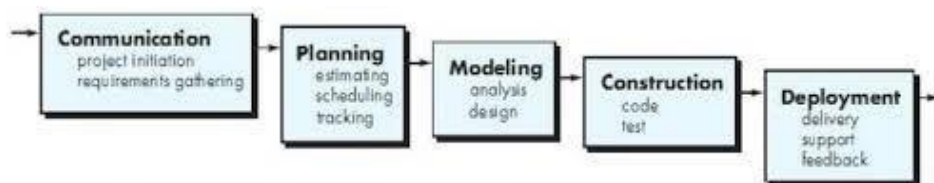
Metode perancangan yang digunakan dalam tugas akhir ini adalah menggunakan metode *forward chaining* bisa disebut juga runut maju atau pencarian yang dimotori data (*data driven search*). Jadi pencarian dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *derived information* (*then*).

### **2.3. Metode Pengembangan**

Metode *Waterfall* adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Dalam proses

implementasi metode *Waterfall* ini, sebuah langkah akan diselesaikan terlebih dahulu dimulai dari tahapan yang pertama sebelum melanjutkan ke tahapan yang berikutnya. Adapun keuntungan menggunakan metode *waterfall* ini yaitu requirement harus didefinisikan lebih mendalam sebelum proses *coding* dilakukan, selain itu proses implementasinya dilakukan secara bertahap dari tahap pertama hingga tahap terakhir secara berurutan. Disamping itu metode *Waterfall* ini juga memungkinkan sedikit mungkin perubahan yang dilakukan oleh proyek berlangsung.

Fase-fase dalam *Waterfall* Model menurut referensi Pressman :



**GAMBAR: 2.1.** Waterfall Pressman (Pressman, 2015)

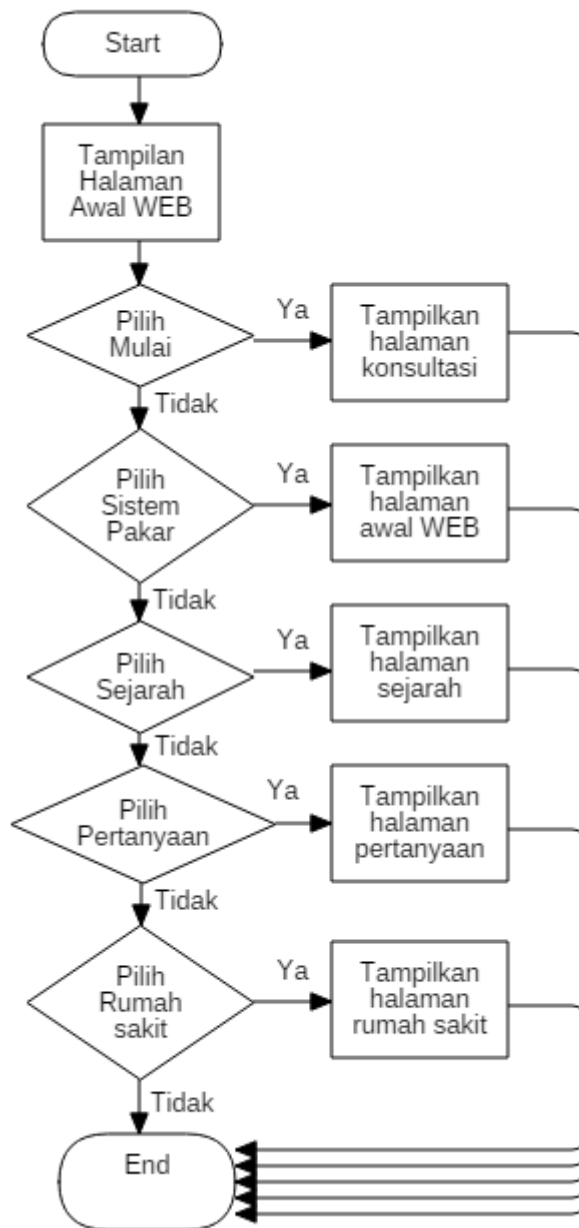
### **3. HASIL DAN PEMBAHASAN**

#### **3.1. Analisis Masalah**

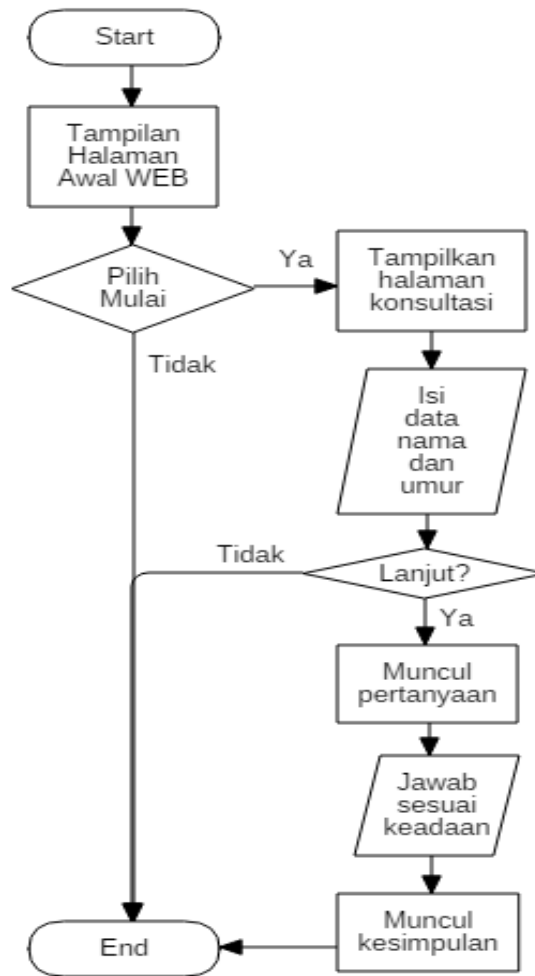
Analisis masalah merupakan langkah dimana langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang sedang berjalan. Adapun identifikasi masalah pada penelitian ini adalah bagaimana cara meningkatkan kembali ketertarikan dan menganalisis secara tepat mengenai terindikasi tidaknya penyakit baru ini dengan diagnosis gejala menggunakan konsep **WEB**.

#### **3.2. Analisis Sistem yang Diusulkan**

Analisis ini menjelaskan tentang kebutuhan atau kondisi yang harus dipenuhi dalam suatu sistem.



**GAMBAR: 3.1.** Flowchart Media Untuk Memahami Yang Diusulkan



**GAMBAR 3.2.** *Flowchart* Sistem Munculnya Pertanyaan Diagnosa

Pada *flowchart* diatas adalah usulan sistem hingga memunculkan tampilan pertanyaan diagnosa penyakit yang akan mengetes terkena atau tidak terkena penyakit tersebut.

### 3.3. *Modelling*

#### 3.3.1. Desain Perangkat Lunak

Desain atau perancangan merupakan tahap untuk memenuhi kebutuhan *user* mengenai gambaran yang jelas tentang perancangan dari sistem yang akan dibuat setiap diimplementasikan. Adapun dalam tahap perancangan ini terdapat sistem dengan notasi **UML**, dan perancangan *interface*.

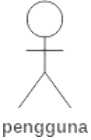
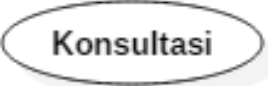

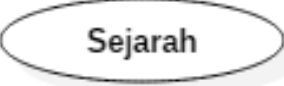
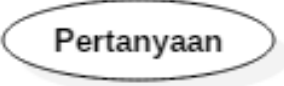
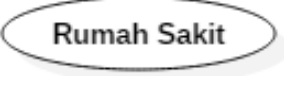

### **3.3.2. Perancangan Sistem**

Dalam perancangan sistem perangkat lunak, penulis menggunakan model berorientasi objek dengan menggunakan notasi **UML** (*Unified Modelling Language*), diantaranya *Use case* Diagram, *Activity* Diagram, *Class* Diagram, *Statechart* Diagram, *Sequence* Diagram, *Component* Diagram, *Deployment* Diagram.

#### **3.3.2.1. Use case Diagram**

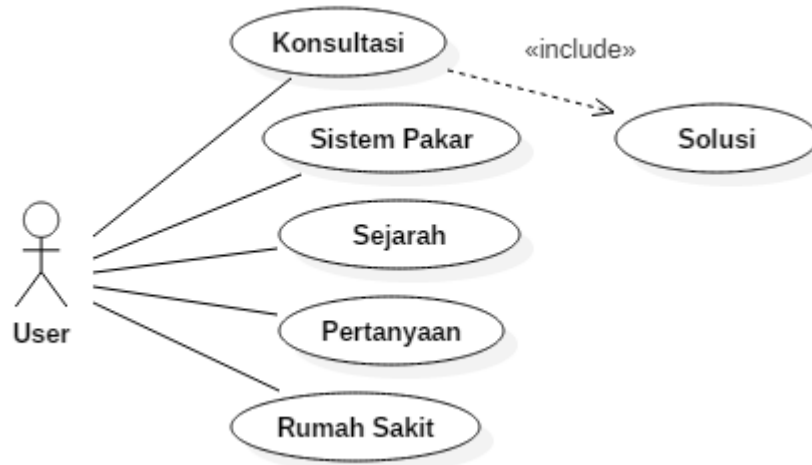
*Use case* Diagram merupakan gambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *Use case* Diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. *Use case* Diagram juga menggambarkan interaksi aktor dengan sistem yang akan dibangun guna mengetahui fungsi apa saja yang tersedia didalamnya. Aktor dalam aplikasi ini, yaitu pengguna. Pengguna disini merupakan semua kalangan. Adapun yang dapat dilakukan oleh pengguna yaitu melihat tampilan **WEB** sistem pakar dan isinya. Berikut ini merupakan tabel deskripsi perancangan *Use case* Diagram yang digambarkan pada tabel 3.1.

**TABEL: 3.1. Deskripsi Perancangan Use case Diagram**

No	Notasi	Deskripsi
1.		Merupakan aktor yang dapat menjalankan semua fitur.
2.		Merupakan fitur dimana pengguna dapat berkonsultasi.
3.		Merupakan fitur dimana pengguna dapat melihat tampilan awal <b>WEB</b> .
4.		Merupakan fitur dimana pengguna dapat melihat sejarah penyakit pertama ditemukan.
5.		Merupakan fitur dimana pengguna dapat melihat seputar pertanyaan tentang penyakit.
6.		Merupakan fitur dimana pengguna dapat melihat rumah sakit rujukan di Indonesia.
7.		Merupakan fitur yang dapat menampilkan hasil dari konsultasi sebelumnya.



Adapun perancangan *Use case* Diagram dalam aplikasi sistem pakar **COVID-19** ini di gambarkan pada gambar 3.3. berikut ini :



**GAMBAR: 3.3.** *Use case* Diagram Aplikasi Sistem Pakar Penyakit **COVID-19**

Berdasarkan gambar diatas, maka dibuatlah deskripsi dengan skenario dari setiap proses yang ada pada *Use case* Diagram tersebut kedalam tabel.

**TABEL: 3.2.** Skenario *Use case* Diagram

<b>Nama Use case</b>	Menampilkan Sistem Pakar Penyakit <b>COVID-19</b>		
<b>Aktor</b>	Pengguna		
<b>Deskripsi</b>	Digunakan untuk melihat seputar dengan penyakit <b>COVID-19</b>		
<b>Tujuan</b>	<i>User</i> dapat melihat tentang seputar penyakit dalam bentuk <b>WEB</b>		
<b>Post Condition</b>	Muncul pertanyaan diagnosa		
<b>Scenario</b>	<b>Step</b>	<b>Actor Activity</b>	<b>Respon System</b>
	1	<i>User</i> memilih menu konsultasi dengan mengklik <i>button</i> “MULAI”	Menampilkan halaman mengisikan data diri dan muncul beberapa pertanyaan.

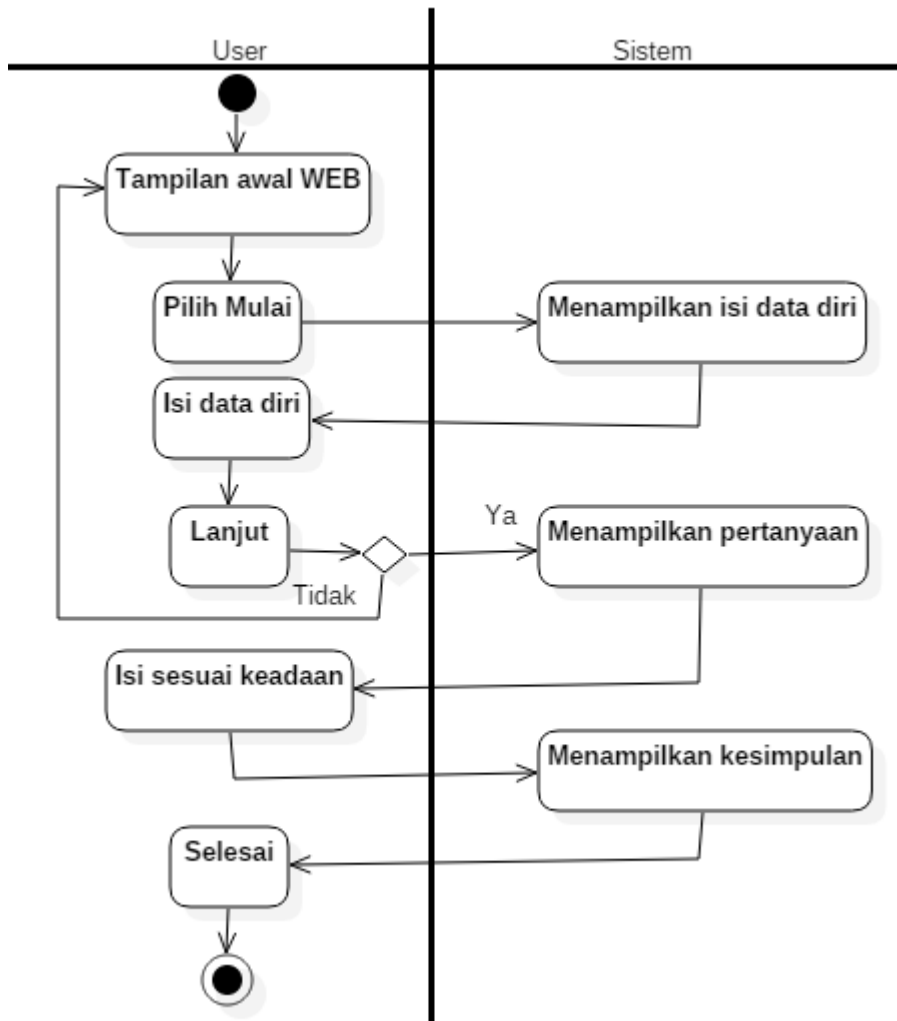
**Lanjutan TABEL: 3.2. Skenario Use case Diagram**

<i>Scenario</i>	<i>Step</i>	<i>Actor Activity</i>	<i>Respon System</i>
	2	<i>User</i> memilih beberapa menu <i>bar</i> seperti sistem pakar, sejarah, seputar pertanyaan, dan rumah sakit rujukan	Menampilkan apa yang diinputkan <i>user</i> di halaman tersebut.
	3	<i>User</i> mendapatkan hasil diagnosa yaitu solusi dan kesimpulan.	Menampilkan hasil diagnosa setelah <i>user</i> mengisi beberapa pertanyaan.

### **3.3.2.2. Activity Diagram**

1. *Activity* diagram identifikasi menampilkan pertanyaan diagnosa.

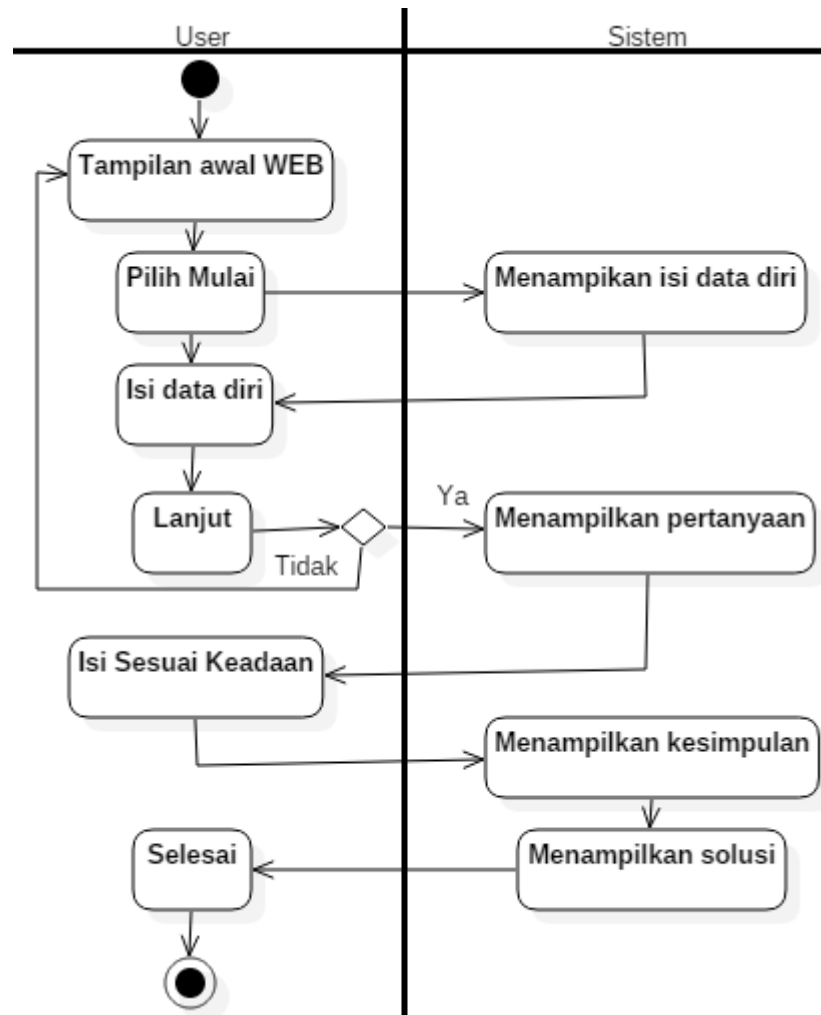
*Activity* diagram pada gambar 3.4. menggambarkan proses dalam menampilkan pertanyaan hingga proses selesai.



**GAMBAR: 3.4.** Activity Diagram Menampilkan Pertanyaan Diagnosa

2. *Activity* diagram identifikasi menampilkan solusi.

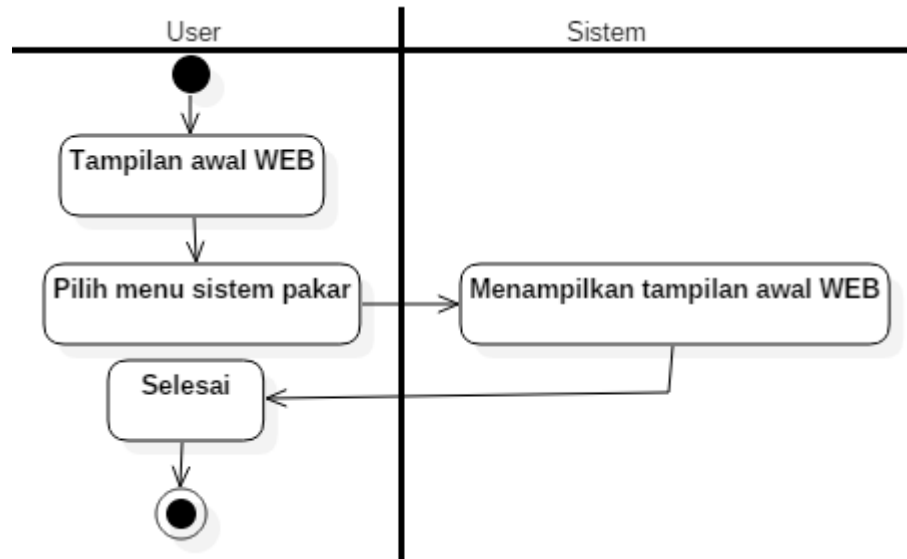
*Activity* diagram pada gambar 3.5. menggambarkan proses dalam menampilkan pertanyaan hingga proses solusi.



**GAMBAR: 3.5.** *Activity* Diagram Menampilkan Solusi

### 3. Activity diagram Menampilkan halaman awal WEB

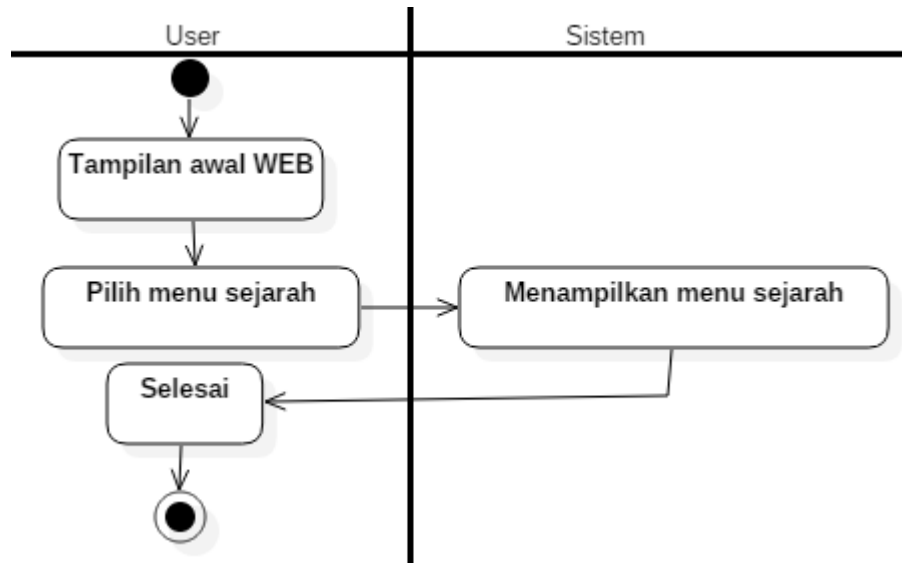
Activity Diagram pada gambar 3.6. menggambarkan proses dari menampilkan rumah sakit rujukan COVID-19.



**GAMBAR: 3.6.** Activity Diagram Menampilkan Tampilan Awal WEB

### 4. Activity diagram menampilkan sejarah penyakit COVID-19

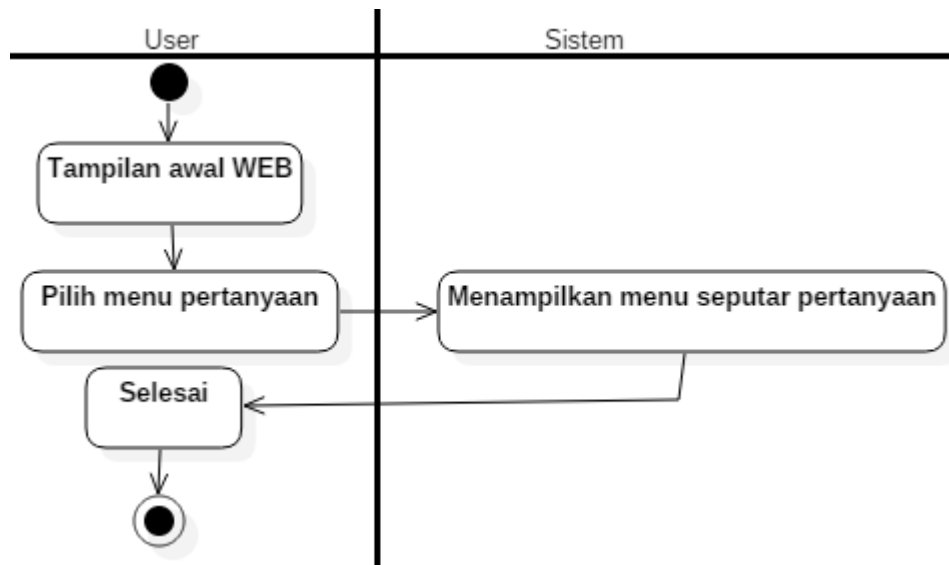
Activity diagram pada gambar 3.7. menggambarkan proses dari menampilkan info sejarah penyakit COVID-19.



**GAMBAR: 3.7.** Activity Diagram Info Sejarah Penyakit COVID-19

4. Activity diagram menampilkan seputar pertanyaan mengenai COVID-19

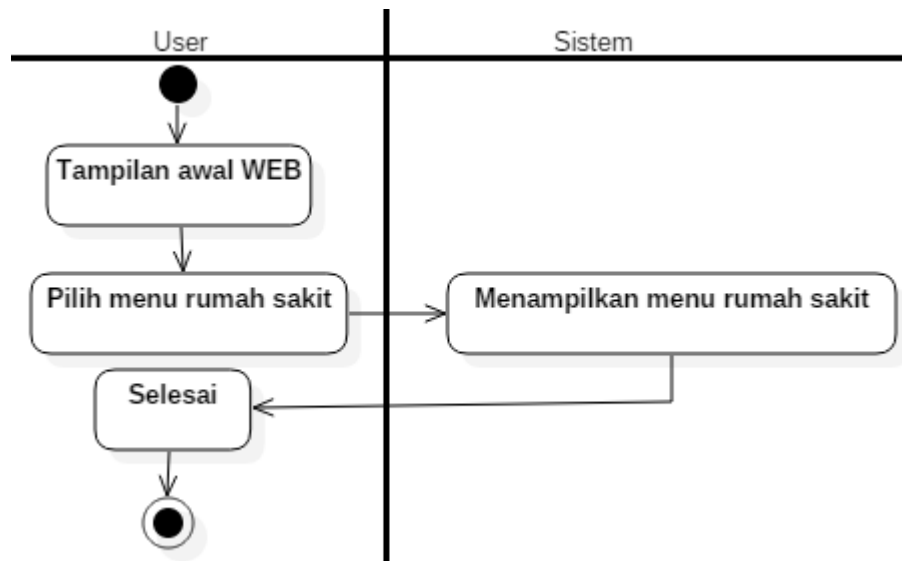
Activity diagram pada gambar 3.8. menggambarkan proses dari menampilkan seputar pertanyaan mengenai COVID-19.



**GAMBAR: 3.8.** Activity Diagram Menampilkan Seputar Pertanyaan **COVID-19**

5. Activity diagram menampilkan rumah sakit rujukan

Activity diagram pada gambar 3.9 menggambarkan proses dari menampilkan rumah sakit rujukan **COVID-19**.



**GAMBAR: 3.9.** Activity Diagram Menampilkan Rumah Sakit Rujukan  
**COVID-19**

### 3.3.2.3. Class Diagram

*Class Diagram* digunakan untuk menampilkan beberapa kelas yang ada di dalam perangkat lunak yang akan dikembangkan. *Class Diagram* menggambarkan struktur dan deskripsi beserta hubungan kelas yang satu dengan yang lainnya. Berikut ini merupakan class diagram dari aplikasi sistem pakar **COVID-19** di gambarkan pada gambar 3.10.





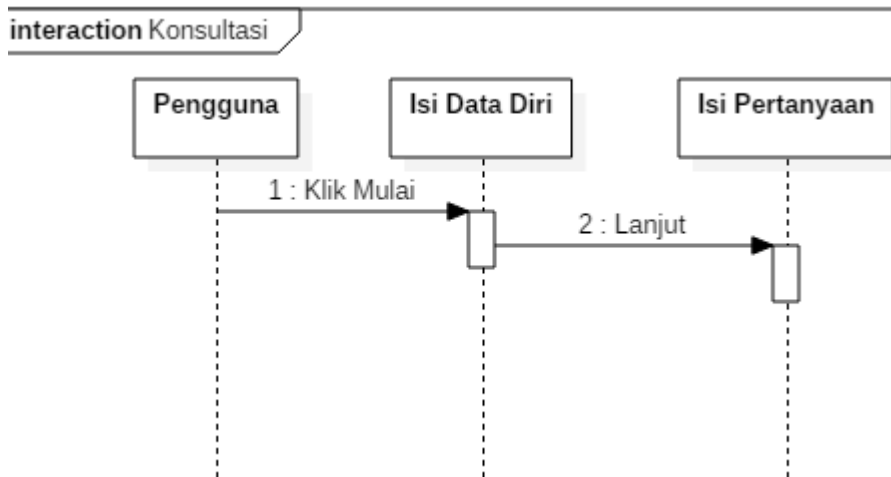
**GAMBAR: 3.10.** Class Diagram Aplikasi Sistem Pakar COVID-19

### 3.3.2.4. Sequence Diagram

*Sequence* diagram merupakan gambaran interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Berikut merupakan *Sequence* diagram pada aplikasi sistem pakar COVID-19 dijelaskan pada gambar 3.11. sampai gambar 3.16. :

#### 1. *Sequence* Diagram Pertanyaan Diagnosa.

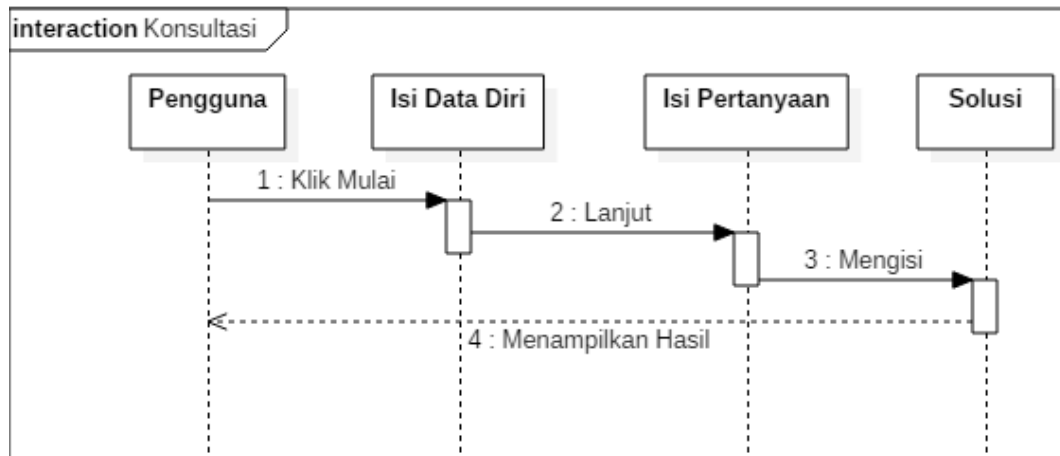
*Sequence* diagram pada gambar 3.11. menggambarkan urutan pesan yang ada dalam *Use case* diagram identifikasi pertanyaan diagnosa.



**GAMBAR: 3.11.** *Sequence* Diagram Pertanyaan Diagnosa

## 2. Sequence Diagram Menampilkan Solusi.

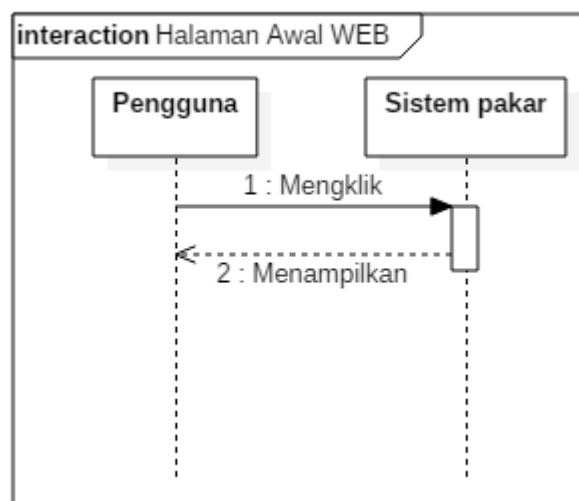
Sequence diagram pada gambar 3.12. menggambarkan urutan pesan yang ada dalam Use case diagram menampilkan solusi.



**GAMBAR: 3.12.** Sequence Diagram Solusi

## 3. Sequence Diagram Menampilkan Menu Tampilan Awal.

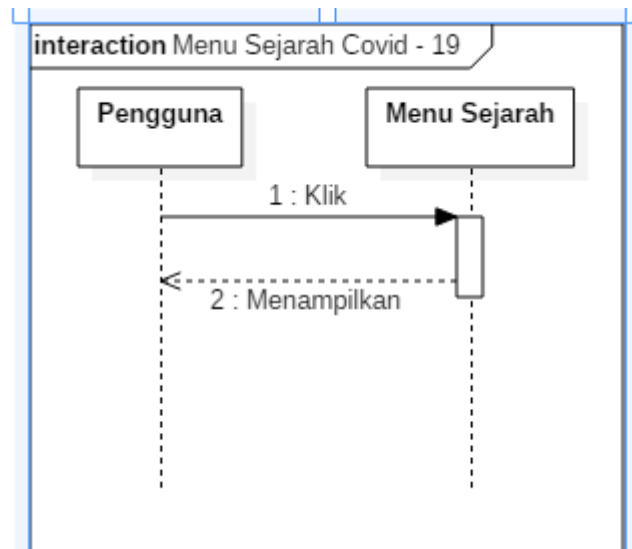
Sequence diagram pada gambar 3.13. menggambarkan urutan pesan yang ada dalam Use case diagram menampilkan menu tampilan awal **WEB**.



**GAMBAR: 3.13.** Sequence Diagram Menampilkan Menu Tampilan Awal **WEB**

#### 4. *Sequence* Diagram Menampilkan Menu Sejarah.

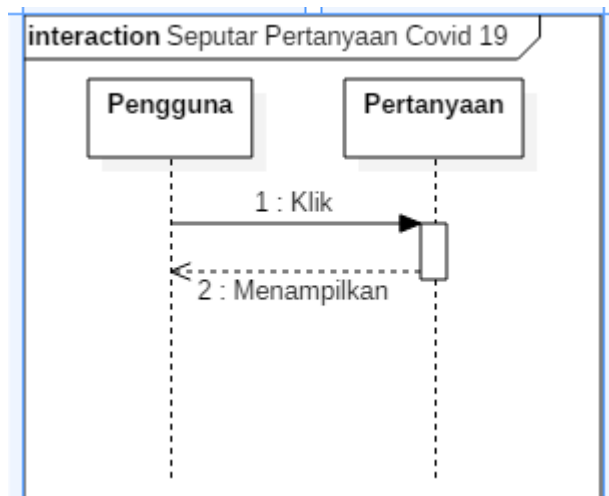
*Sequence* diagram pada gambar 3.14. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan menu sejarah.



**GAMBAR: 3.14.** *Sequence* Diagram Menampilkan Menu Sejarah COVID-19

#### 5. *Sequence* Diagram Menampilkan Menu Seputar Pertanyaan COVID-19.

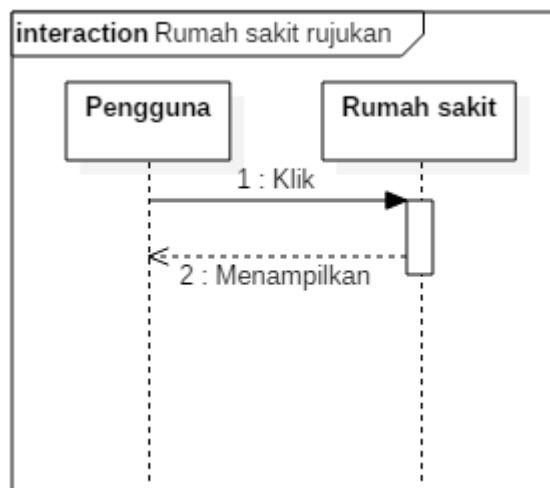
*Sequence* Diagram pada gambar 3.15. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan seputar pertanyaan.



**GAMBAR: 3.15.** *Sequence* Diagram Menampilkan Menu Seputar Pertanyaan COVID-19

6. *Sequence* Diagram Menampilkan Menu Rumah Sakit Rujukan COVID-19.

*Sequence* diagram pada gambar 3.16. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan rumah sakit rujukan.



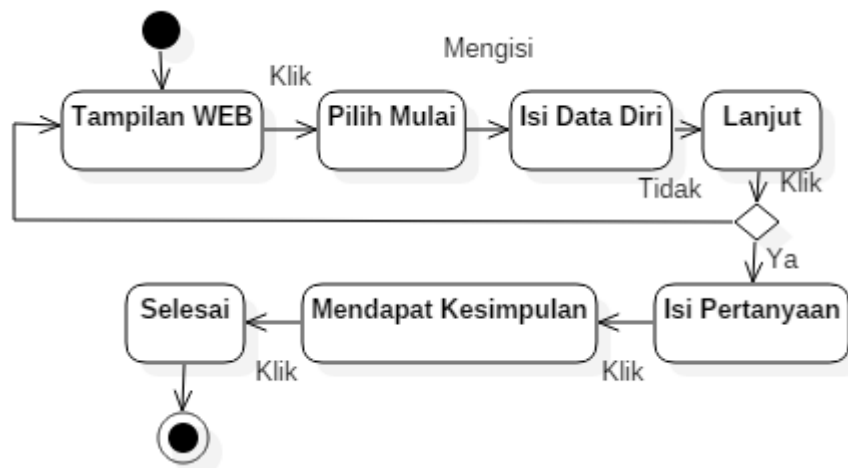
**GAMBAR: 3.16.** *Sequence* Diagram Menampilkan Rumah Sakit Rujukan COVID-19

### 3.5.2.5. Statechart Diagram

*Statechart* diagram merupakan proses yang menggambarkan perubahan status atau transisi dari suatu sistem (dari *State* satu ke *State* lainnya). Berikut merupakan *Statechart* diagram pada aplikasi sistem pakar **COVID-19** dijelaskan pada gambar 3.17. sampai gambar 3.22. :

#### 1. *Statechart* Diagram Pertanyaan Diagnosa

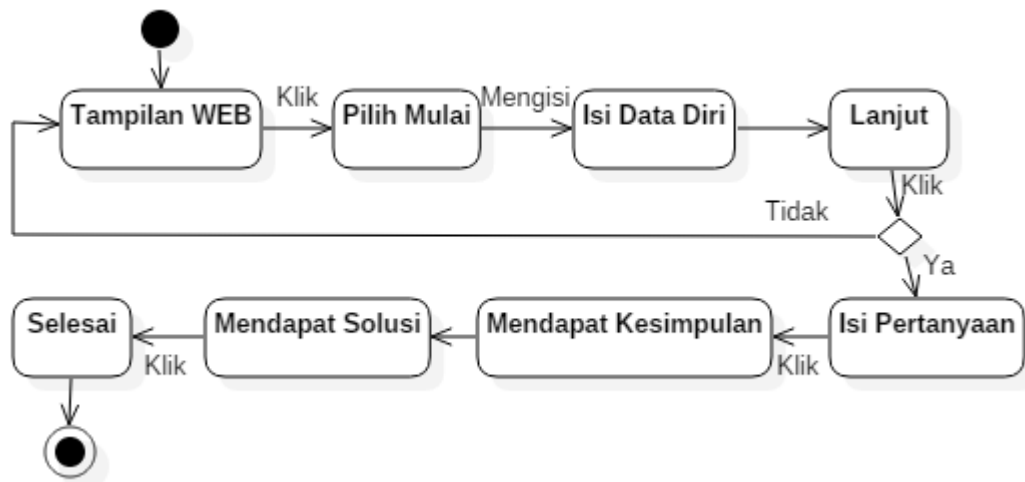
*Statechart* diagram pertanyaan diagnosa menggambarkan dari tampilan awal ke proses pertanyaan diagnosa untuk tercapainya kesimpulan terkena atau tidak terkena terhadap penyakit **COVID-19**.



**GAMBAR: 3.17.** *Statechart* Diagram Pertanyaan Diagnosa

#### 2. *Statechart* Diagram Solusi

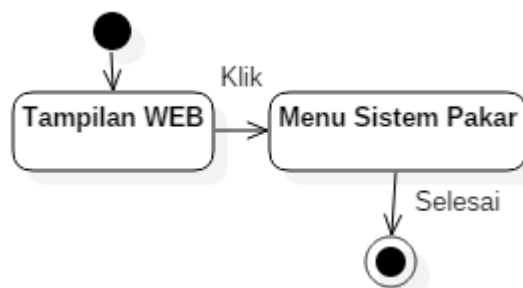
*Statechart* diagram menu solusi menggambarkan saat proses dilakukannya pilihan untuk mendapatkan solusi.



**GAMBAR: 3.18.** Statechart Diagram Solusi

### 3. Statechart Diagram Menu Awal WEB

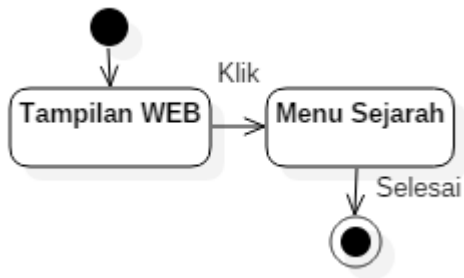
Statechart diagram menu awal WEB menggambarkan saat proses dilakukannya pilihan untuk menampilkan tentang tampilan awal WEB.



**GAMBAR: 3.19.** Statechart Diagram Menu Awal WEB

### 4. Statechart Diagram Menu Sejarah COVID-19

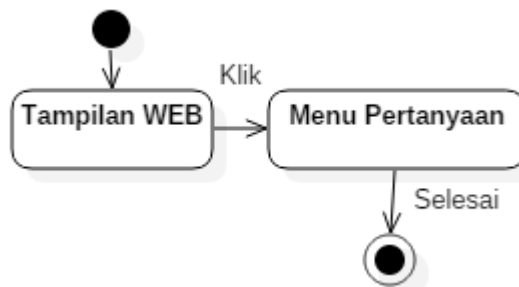
Statechart diagram menu sejarah menggambarkan saat proses dilakukannya pilihan untuk menampilkan tentang sejarah COVID-19.



**GAMBAR: 3.20.** *Statechart* Diagram Menu Sejarah **COVID-19**

5. *Statechart* Diagram Menampilkan Menu Pertanyaan Seputar **COVID-19**

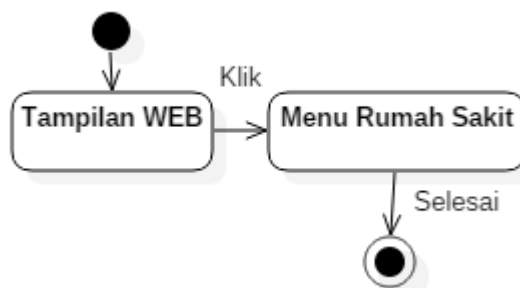
*Statechart* diagram menampilkan menu pertanyaan seputar **COVID-19**.



**GAMBAR: 3.21.** *Statechart* Diagram Menampilkan Pertanyaan Seputar **COVID-19**

6. *Statechart* Diagram Menampilkan Menu Rumah Sakit Rujukan **COVID-19**

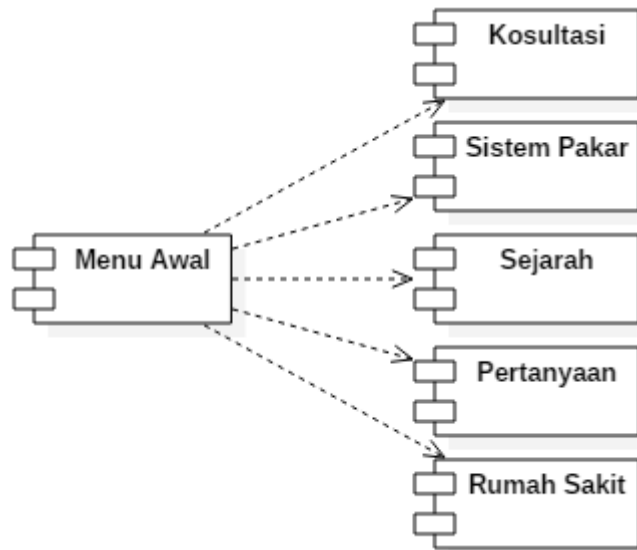
*Statechart* diagram menampilkan menu rumah sakit rujukan **COVID-19**.



**GAMBAR: 3.22.** *Statechart* Diagram Menampilkan Menu Rumah Sakit Rujukan **COVID-19**

### 3.3.2.6. Component Diagram

Diagram komponen yang digunakan pada aplikasi sistem pakar **COVID-19** ini terdiri dari komponen aplikasi yaitu komponen main menu yang saling terhubung dengan komponen yang lainnya seperti komponen menu sejarah **COVID-19**, komponen menu tentang pertanyaan seputar **COVID-19**, dan komponen menu rumah sakit rujukan **COVID-19** di Indonesia. Berikut adalah diagram komponen dari aplikasi sistem pakar **COVID-19** yang di jelaskan pada gambar 3.23.

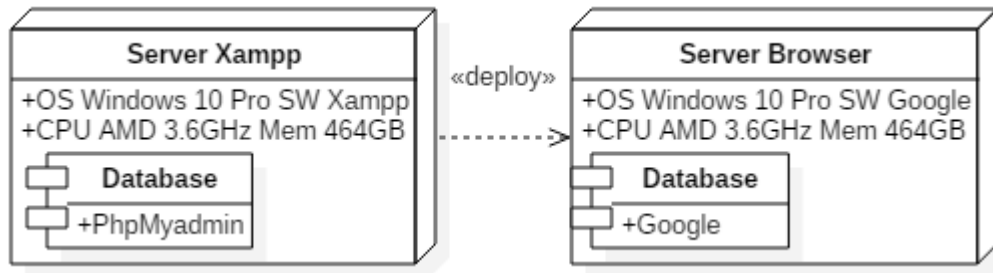


**GAMBAR: 3.23.** Component Diagram Aplikasi Sistem Pakar **COVID-19**

### 3.3.2.7. Deployment Diagram

*Deployment* diagram pada aplikasi sistem pakar **COVID-19** ini menjelaskan hubungan antar *user*, aplikasi. *User* menggunakan *device* berupa perangkat laptop dalam menjalankan aplikasi ini. Model dari *Deployment* diagram aplikasi sistem pakar **COVID-19** yang diusulkan pada gambar 3.24.

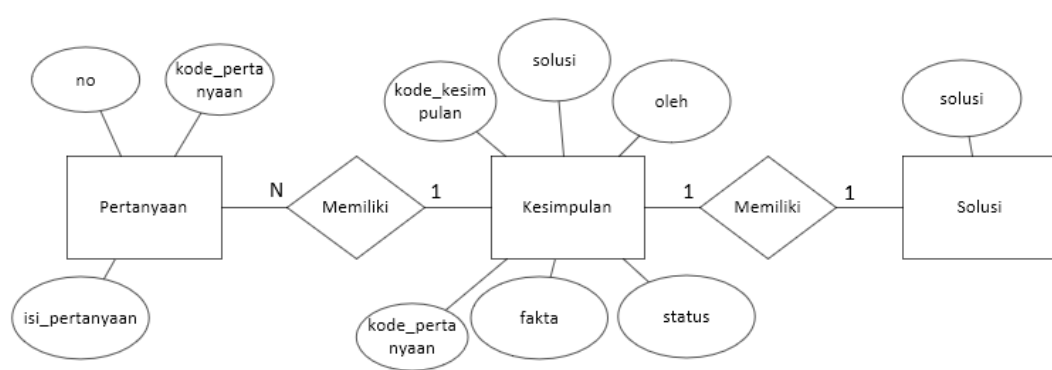




**GAMBAR: 3.24.** *Deployment Diagram* Aplikasi Sistem Pakar COVID-19

### 3.3.3. *Entity Relationship Diagram*

**ERD** (*Entity Relationship Diagram*) adalah sebuah konsep yang mendeskripsikan hubungan antara penyimpanan (*database*) dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek-objek tersebut. Berikut merupakan **ERD** dari aplikasi sistem pakar COVID-19 yang di jelaskan pada gambar 3.25.



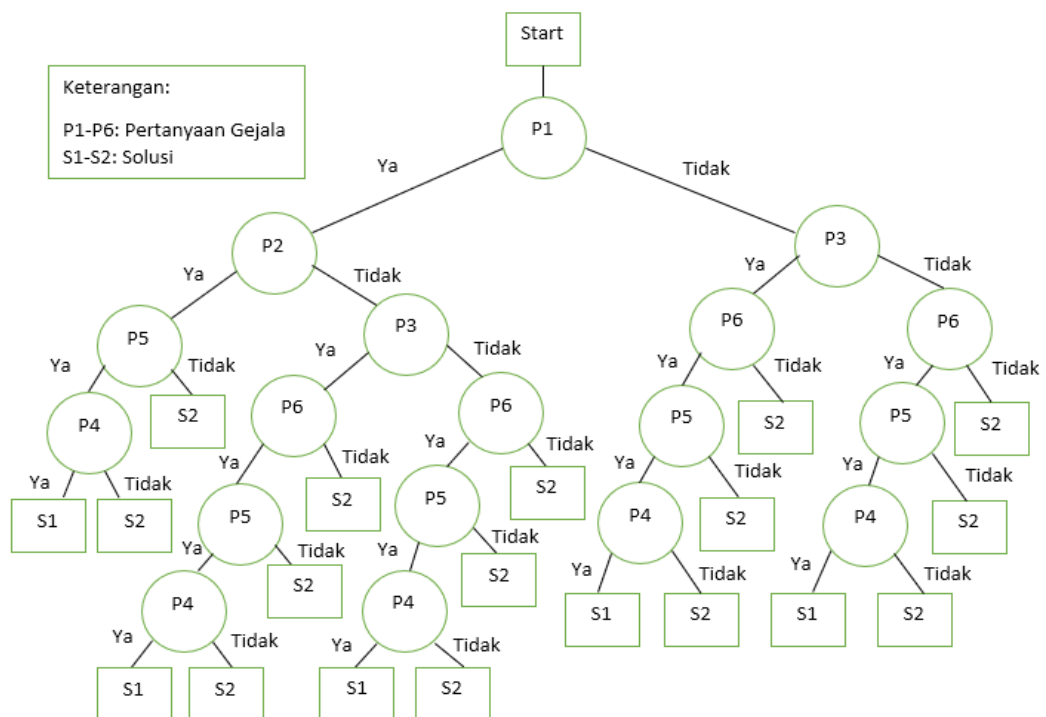
**GAMBAR: 3.25.** *Entity Relationship Diagram* Aplikasi Sistem Pakar COVID-19

Pada gambar diatas memiliki tiga entitas yaitu pertanyaan, kesimpulan dan, solusi serta dua relasi yang menghubungkan ketiganya. Entitas pertanyaan, memiliki atribut no, kode\_pertanyaan, dan isi\_pertanyaan. Dan memiliki sub atribut yaitu pertanyaan-pertanyaan, dan harus memiliki satu kesimpulan. Entitas kesimpulan memiliki atribut yaitu kode\_kesimpulan, solusi, oleh, kode\_pertanyaan, fakta, dan status. Pada atribut fakta memiliki sub atribut yaitu segera lakukan **SWAB test** dan jaga kesehatan. Dan entitas solusi memiliki atribut solusi dan memiliki dua sub atribut yaitu terindikasi dan tidak terindikasi. Jadi jika

sudah menjawab banyak pertanyaan mendapatkan satu kesimpulan beserta mendapatkan satu solusi. Entitas pertanyaan ke kesimpulan menggunakan relasi *Many to One* sedangkan entitas kesimpulan ke solusi menggunakan relasi *One to One*.

### 3.3.4. Pohon Keputusan

Dengan di temukannya gejala-gejala penyakit **COVID-19** dan metode inferensi yang digunakan *forward chaining* yang timbul atau tampak maka akan mempermudah dalam pembuatan *decision tree* atau pohon keputusan tentang penentuan diagnosa di bawah ini, merupakan penentuan diagnosa berdasarkan gejala-gejala yang muncul.



**GAMBAR: 3.26.** Pohon Keputusan Aplikasi Sistem Pakar **COVID-19**

### 3.3.5. Rule/Aturan

Aturan (*rule*) biasanya dituliskan dalam bentuk jika-maka (*IF-THEN*). *Rule* ini dapat dikatakan sebagai hubungan implikasi dua bagian yaitu bagian premise (jika) dan bagian konklusi (maka). Apabila bagian premise dipenuhi maka bagian

konklusi juga akan bernilai benar. Berikut beberapa *rule* dalam menganalisis penyakit virus **COVID-19**.

*Rule 1*

*IF* Demam

*AND* Kehilangan Rasa/Bau

*AND* Kelelahan

*AND* Sesak Nafas

*THEN* Anda Terindikasi

*Rule 2*

*IF* Demam

*AND* Kehilangan Rasa/Bau

*AND* Kelelahan

*AND* Tidak Sesak Nafas

*THEN* Anda Tidak Terindikasi

*Rule 3*

*IF* Demam

*AND* Kehilangan Rasa/Bau

*AND* Tidak Kelelahan

*THEN* Anda Tidak Terindikasi

*Rule 4*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Sesak Nafas

*AND* Pilek

*AND* Kelelahan

*AND* Sesak Nafas

*THEN* Anda Terindikasi

*Rule 5*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Sesak Nafas

*AND* Pilek

*AND* Kelelahan

*AND* Tidak Sesak Nafas

*THEN* Anda Tidak Terindikasi

*Rule 6*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Sesak Nafas

*AND* Pilek

*AND* Tidak Kelelahan

*THEN* Anda Tidak Terindikasi

*Rule 7*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Sesak Nafas

*AND* Tidak Pilek

*THEN* Anda Tidak Terindikasi

*Rule 8*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Tidak Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Sesak Nafas

*THEN* Anda Terindikasi

*Rule 9*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Tidak Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Tidak Sesak Nafas

*THEN* Anda Tidak Terindikasi

*Rule 10*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Tidak Batuk

*AND* Pilek

*AND* Tidak Kelelahan

*THEN* Anda Tidak Terindikasi

*Rule 11*

*IF* Demam

*AND* Tidak Kehilangan Rasa/Bau

*AND* Tidak Batuk

*AND* Tidak Pilek

*THEN* Anda Tidak Terindikasi

*Rule 14*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Tidak Kelelahan

*THEN* Anda Tidak Terindikasi

*Rule 12*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Sesak Nafas

*THEN* Anda Terindikasi

*Rule 15*

*IF* Tidak Demam

*AND* Batuk

*AND* Tidak Pilek

*THEN* Anda Tidak Terindikasi

*Rule 13*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Tidak Sesak Nafas

*THEN* Anda Tidak Terindikasi

*Rule 16*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Sesak Nafas

*THEN* Anda Terindikasi

*Rule 17*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Kelelahan

*AND* Tidak Sesak Nafas

*THEN* Anda Tidak Terindikasi

*Rule 18*

*IF* Tidak Demam

*AND* Batuk

*AND* Pilek

*AND* Tidak Kelelahan

*THEN* Anda Tidak Terindikasi

*Rule 19*

*IF* Tidak Demam

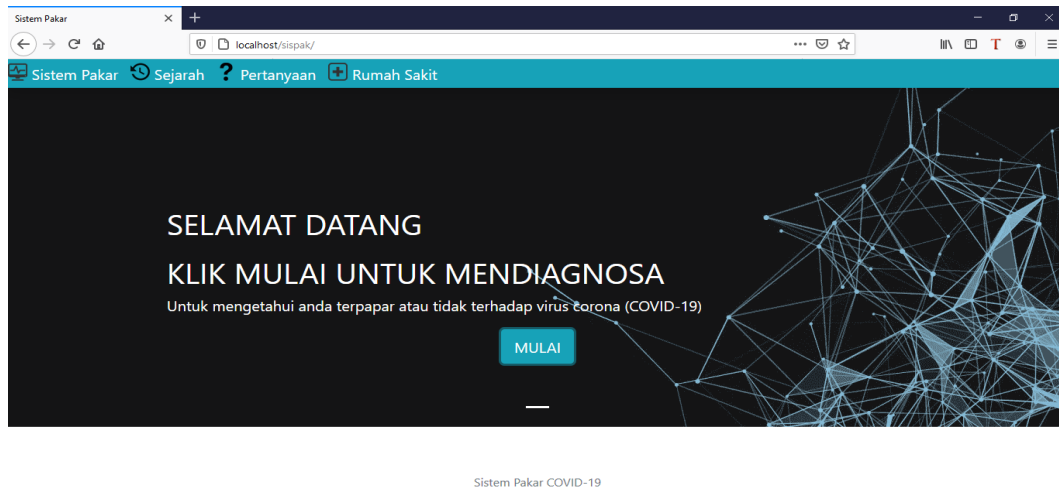
*AND* Batuk

*AND* Tidak Pilek

*THEN* Anda Tidak Terindikasi

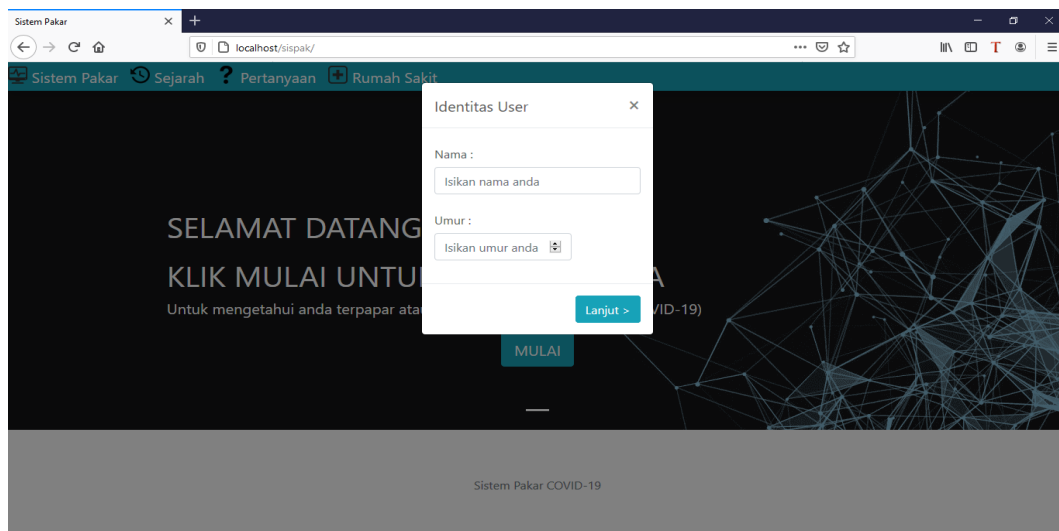
### 3.3.6. Implementasi Antarmuka

Pada gambar ini menampilkan tampilan awal **WEB**.



**GAMBAR: 3.27.** Tampilan Awal **WEB**

Jika ingin memulai, klik tombol *button* yang bertuliskan Mulai maka tampilannya akan seperti berikut ini.



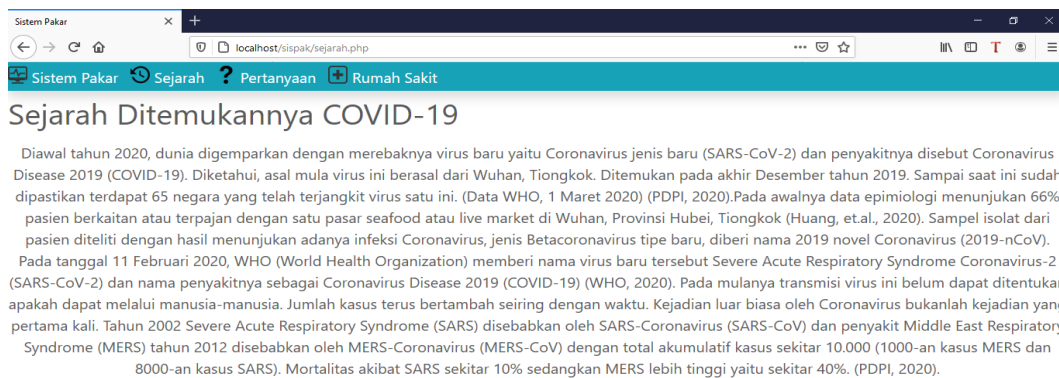
**GAMBAR: 3.28.** Tampilan Untuk Memulai Diagnosa

Jika sudah mengisi data nama dan umur maka selanjutnya akan muncul tampilan pertanyaan diagnosa seperti berikut ini.



### **GAMBAR: 3.29.** Tampilan Pertanyaan Diagnosa

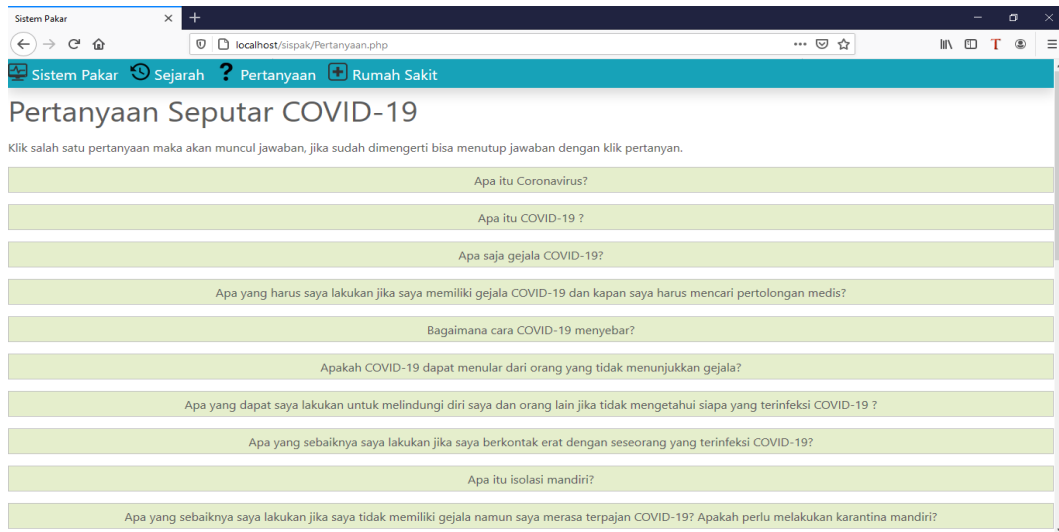
Pada gambar berikut ini tampilan jika mengklik menu sejarah.



### **GAMBAR: 3.30.** Tampilan Menu Sejarah



Pada gambar berikut ini tampilan jika mengklik menu seputar Pertanyaan.



**GAMBAR: 3.31.** Tampilan Menu Pertanyaan

### 3.3.7 *Testing*

*Testing* merupakan proses menganalisa sesuatu entitas aplikasi untuk mendeteksi perbandingan antara keadaan yang terdapat dengan keadaan yang diinginkan (*defects/error/bugs*) serta mengevaluasi fitur-fitur dari entitas *software* menurut (standar *ANSI/ IEEE*, 1059).

#### **A. *Black Box Testing***

*Black box testing* juga dikenal sebagai *behavioral testing*, metode ini merupakan pengujian terhadap fungsionalitas atau kegunaan sebuah aplikasi.

**TABEL: 3.3. Rencana Pengujian**


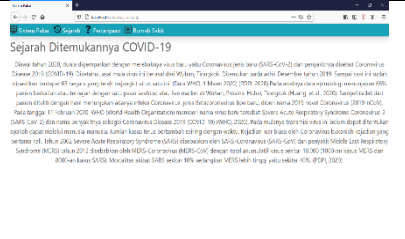

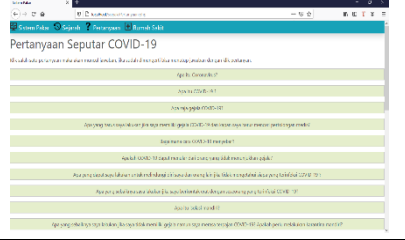

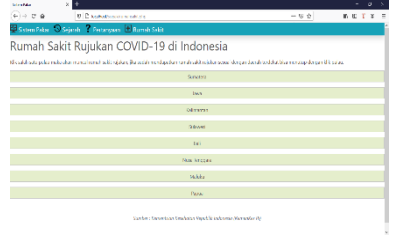
No	Kelas Uji	Butir Pengujian	Jenis Pengujian
1	Menu/Button	Memilih <i>button</i> Mulai	<i>Black Box</i>
		Memilih <i>button</i> Sistem Pakar	<i>Black Box</i>
		Memilih <i>button</i> Sejarah	<i>Black Box</i>
		Memilih <i>button</i> Pertanyaan	<i>Black Box</i>
		Memilih <i>button</i> Rumah Sakit	<i>Black Box</i>
		Memilih <i>button</i> Lanjut	<i>Black Box</i>
		Memilih <i>button</i> Silang	<i>Black Box</i>
		Memilih <i>button</i> Ya atau Tidak	<i>Black Box</i>
		Memilih <i>button</i> Akhiri	<i>Black Box</i>
2	Isi data	Mengisi data Nama dan Umur	<i>Black Box</i>
		Tidak mengisi data Nama dan Umur	<i>Black Box</i>
		Mengisi data Nama saja	<i>Black Box</i>
		Mengisi data Umur saja	<i>Black Box</i>

Setelah dilakukan pengujian dengan menggunakan metode *Black box*, maka didapatkan hasil pengujian pada tabel berikut ini:







**TABEL: 3.4. Tabel Hasil Pengujian *Black box***

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
1	Menu	Memilih <i>button</i> Mulai		Masuk kedalam menu isi data diri	Menampilkan menu data diri untuk diisi sebagai syarat untuk tes diagnosa		Berhasil
2		Memilih <i>button</i> Sistem Pakar		Masuk ke tampilan awal <b>WEB</b>	Menampilkan awal <b>WEB</b>		Berhasil



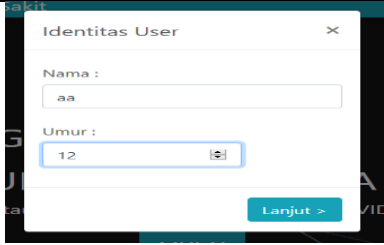


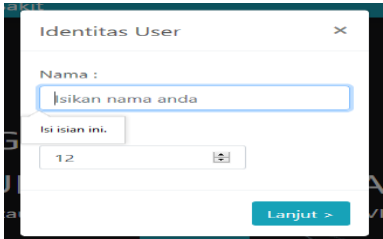
Lanjutan dari TABEL: 3.4. Tabel Hasil Pengujian *Black box*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
3	Menu	Memilih <i>button</i> Sejarah		Masuk ke menu tentang sejarah COVID-19	Menampilkan sejarah COVID-19		Berhasil
4		Memilih <i>button</i> Pertanyaan		Masuk ke menu tentang pertanyaan seputar COVID-19	Menampilkan list berbagai pertanyaan seputar COVID-19		Berhasil
5		Memilih <i>button</i> Rumah Sakit		Masuk ke menu rumah sakit rujukan COVID-19	Menampilkan list berbagai rumah sakit rujukan COVID-19 di Indonesia		Berhasil




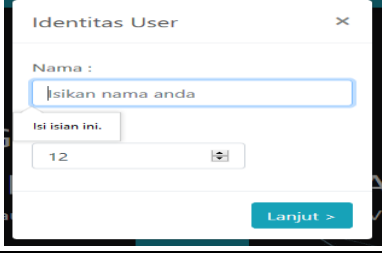
Lanjutan dari TABEL: 3.4. Tabel Hasil Pengujian *Black box*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
6	Menu	Memilih <i>button</i> Lanjut		Masuk ke menu pertanyaan diagnosa	Menampilkan beberapa pertanyaan diagnosa untuk mengambil keputusan		Berhasil
7		Memilih <i>button</i> Silang		Membatalkan untuk tes diagnosa	Menampil-kan awal <b>WEB</b>		Berhasil
8		Memilih <i>button</i> Ya atau Tidak		Pilihan jawaban sesuai dengan keadaan	Menampilkan pertanyaan selanjutnya hingga selesai		Berhasil

Lanjutan dari TABEL: 3.4. Tabel Hasil Pengujian *Black box*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
9		Memilih <i>button</i> Akhiri		Mengakhiri proses tes diagnosa	Menampilk-an tampilan awal <b>WEB</b>		Berhasil
10	Isi Data	Mengisi data Nama dan Umur		Data nama dan umur terisi	Menampilkan menu sesi pertanyaan diagnosa		Berhasil
11		Tidak mengisi data Nama dan Umur		Data nama dan umur tidak terisi	Menampilkan <i>pop up</i> , harus mengisi data yang belum diisi		Berhasil

Lanjutan dari TABEL: 3.4. Tabel Hasil Pengujian *Black box*

No	Item Uji	Skenario Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
12	Isi Data	Mengisi data Nama saja		Data nama terisi sedangkan umur tidak terisi	Menampilkan <i>pop up</i> , harus mengisi data umur		Berhasil
13		Mengisi data Umur saja		Data nama tidak terisi sedangkan umur terisi	Menampilkan <i>pop up</i> , harus mengisi data nama		Berhasil

#### 4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut :

Dengan adanya aplikasi sistem pakar diagnosa penyakit **COVID-19**, maka dapat mengetahui terkena atau tidaknya penyakit ini melalui gejala yang ditimbulkan.

#### 5. REFERENSI

Arhami, M., (2005). Konsep Dasar Sistem Pakar, Andi. Yogyakarta (12 Januari 2021).

Astuti, Puji. (2018). Penggunaan Metode Black Box Testing (Boundary Value Analysis) Pada Sistem Akademik (SMA/SMK). [Online]. 11(2), 186-195. Tersedia di: [https://journal.lppmunindra.ac.id/index.php/Faktor\\_Exacta/article/view/2510](https://journal.lppmunindra.ac.id/index.php/Faktor_Exacta/article/view/2510) (12 Februari 2021).

Carfi, A., Bernabei, R., & Landi, F. (2020). Persistent symptoms in patients after acute COVID-19. *Jama*, 324(6), 603-605 <https://europepmc.org/article/pmc/pmc7349096>. (6 Januari 2021)

Chan, J. F. W., Yuan, S., Kok, K. H., To, K. K. W., Chu, H., Yang, J., ... & Yuen, K. Y., (2020). A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. *The Lancet*, 395(10223), 514-523 <https://www.sati.org.ar/files/Coronavirus/THELANCET-1.pdf>. (6 Januari 2021)

Fauci, A. S., Lane, H. C., & Redfield, R. R. (2020). COVID-19—navigating the uncharted <https://www.nejm.org/doi/full/10.1056/nejme2002387>. (6 Januari 2021)



Hamid, A. R. A. H., (2020). Social responsibility of medical journal: a concern for COVID-19 pandemic. *Medical Journal of Indonesia*, 29(1), 1-3 <https://mji.ui.ac.id/journal/index.php/mji/article/download/4629/1699>. (6 Januari 2021)

Hayadi, B. H., (2018). *Sistem Pakar*, Deepublish. Yogyakarta (16 Mei 2020).

Muslihudin, M., Oktafianto. (2016): *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, C.V Andi Offset. Yogyakarta (30 Mei 2020).

Pressman, R.S. (2015). *Rekayasa Perangkat Lunak : Pendekatan Praktisi Buku I*, Andi. Yogyakarta (24 Februari 2021).

Raharjo. D. S. J., Damiyana. D. & Hidayatullah. M. (2016). *Sistem Pakar Diagnosa Penyakit Lambung dengan Metode Forward Chaining Berbasis Android*. [Online]. 6(2), 1-8. Tersedia di : <https://stmikglobal.ac.id/journal/index.php/sisfotek/article/view/102>. (25 Februari 2021).

Romeo. (2003). *Testing dan Implementasi Sistem*. (Edisi Pertama) [Online]. Tersedia di: [https://repository.dinus.ac.id/docs/ajar/MATERI\\_KULIAH\\_IMPLEMENTASI\\_DAN\\_PENGUJIAN\\_SISTEM.pdf](https://repository.dinus.ac.id/docs/ajar/MATERI_KULIAH_IMPLEMENTASI_DAN_PENGUJIAN_SISTEM.pdf) (12 Februari 2021).

Sitorus, Lamhot. (2015). Algoritma dan Pemrograman, C.V Andi Offset. Yogyakarta (30 Mei 2020).

Wadi, Hamzan. (2020). Sistem Pakar Forward Chaining dengan Java GUI & MySQL : Studi kasus diagnosa penyakit ikan air tawar, Turida Publisher. (16 Februari 2021).

WHO. (2020). Coronavirus [https://www.who.int/health-topics/coronavirus#tab=tab\\_1](https://www.who.int/health-topics/coronavirus#tab=tab_1). (27 Mei 2020).