

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

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh kelulusan Jenjang

Strata Satu (S1)

Pada program Studi Teknik Informatika

Oleh

Danu Syahfrizal

361601015



SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER

INDONESIA MANDIRI

2021

LEMBAR PENGESAHAN

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB EXPERT SYSTEMS FOR DIAGNOSIS OF COVID-19 VIRUS IN WEB

Oleh :

Danu Syahfrizal
361601015

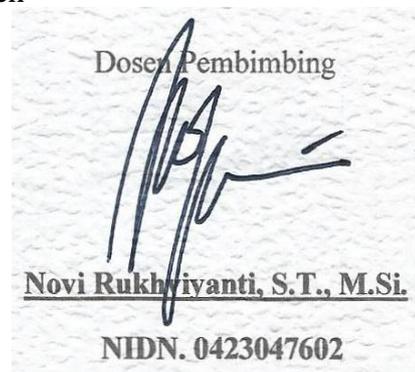
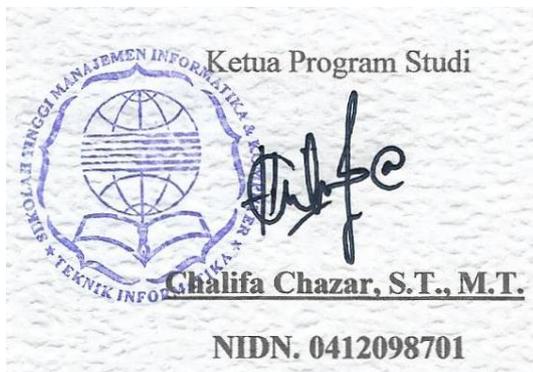
Tugas Akhir Ini Telah Diterima Dan Disahkan Untuk
Memenuhi Persyaratan Mencapai Gelar
Sarjana Teknik Informatika

Pada

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI

Bandung, 24 Februari 2021

Disetujui Oleh



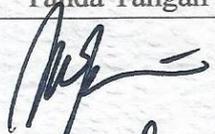
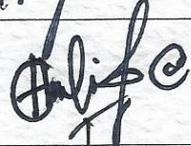
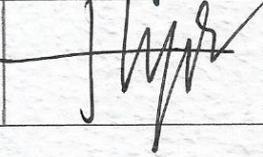
LEMBAR PERSETUJUAN REVISI

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB

Oleh :
Danu Syahfrizal
361601015

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada sidang tugas akhir.

Bandung, Maret 2021
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	Novi Rukhviyanti., S.T., M.Si.	Dosen Pembimbing	
2	Chalifa Chazar, S.T., M.T.	Penguji 1	
3	Haryoso Wicaksono, S.Si., M.M., M.Kom.	Penguji 2	

Mengetahui
Ketua Program Studi Teknik Informatika



Chalifa Chazar., S.T., M.T.
NIDN. 0412098701

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Tugas akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik. Baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
2. Tugas akhir ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terimakasih.
3. Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya seni saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 24 Februari 2021



ABSTRAK

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB

Danu Syahfrizal

361601015

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**

ABSTRACT

EXPERT SYSTEMS FOR DIAGNOSIS OF COVID-19 VIRUS IN WEB

Danu Syahfrizal

361601015

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*

KATA PENGANTAR

Dengan mengucapkan Alhamdulillah sebagai wujud syukur ke Hadirat Allah SWT, yang senantiasa memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan penelitian tugas akhir ini dengan baik dan tepat pada waktunya.

Tugas akhir ini, berjudul **SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB**, disusun untuk melengkapi tahapan akhir studi yang dijalani di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Tugas akhir ini berisi mengenai perancangan sebuah aplikasi berbasis web sebagai media untuk mendiagnosa terkena penyakit menular **COVID-19** dengan menggunakan metode sistem pakar dimana data-data yang dimiliki sudah tervalidasi oleh pakar-pakar kedokteran di bidang keahliannya masing-masing, dari sistem yang diusulkan dengan harapan dapat mengatasi masalah yang diidentifikasi.

Dengan segala keterbatasan tentunya diharapkan aplikasi ini dapat bermanfaat bagi berbagai pihak, khususnya bagi penulis sendiri.

Bandung, 24 Februari 2021

Penulis

Danu Syahfrizal
361601015

UCAPAN TERIMA KASIH

Dengan mengucapkan syukur Alhamdulillah, penelitian ini dapat diselesaikan untuk memenuhi syarat tugas akhir. Laporan penelitian dalam tugas akhir ini di ajukan untuk memenuhi dan melengkapi salah satu syarat akademik dalam kelulusan jenjang Strata Satu (S1) jurusan Teknik Informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Penyusunan tugas akhir ini tidak lepas dari dukungan dan bimbingan dari berbagai pihak, maka pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Ibu Novi Rukhviyanti., S.T., M.Si. selaku Dosen pembimbing yang selalu meluangkan waktu, fikiran dan tenaga dalam memberikan bimbingan, masukan dan saran-sarannya.
2. Bapak Dr. Chairuddin, M.T., M.M. selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
3. Ibu Chalifa Chazar, S.T., M.T. selaku Ketua program studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
4. Seluruh Dosen, Staff dan Karyawan Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM) yang telah mendidik dan membantu dalam memberika informasi serat motivasi dalam proses studi maupun tugas akhir berlangsung.

5. Teruntuk Kedua Orang Tua Tercinta Bapak Machmudi dan Ibu Dwi Hartati yang sangat penulis sayangi dan cintai. Terima kasih selalu memberikan nasehat, dukungan, didikan, kasih sayang, serta Do'a yang penuh dan tulus.
6. Teruntuk adik penulis Sekar Artianingrum terima kasih selalu memberikan dukungan, kasih sayang, serta doa yang tulus.
7. Kakek (Alm), nenek, paman, bibi, dan serta keluarga besar penulis, yang sangat penulis sayangi dan cintai, terima kasih selalu telah mendukung, dan mendoakan penulis selalu.
8. Sahabat-sahabat penulis yaitu angkatan 2016 program studi teknik informatika yang sama-sama berjuang untuk menyelesaikan laporan penelitian tugas akhir ini.
9. Seluruh rekan STMIK, STIE, dan STAN Indonesia Mandiri angkatan 2016 yang sama-sama berjuang untuk terus meraih impian, yang saling membantu juga memberikan semangat, dan seringkali menjadi tempat *sharing*, baik yang berhubungan dengan materi perkuliahan maupun lainnya.
10. Seluruh rekan sahabat, dan pihak-pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu Penulis baik secara langsung maupun tidak langsung memberikan semangat kepada Penulis dalam menyelesaikan laporan penelitian tugas akhir.

Penulis menyadari bahwa masih banyak kekurangan yang mendasar pada laporan penelitian tugas akhir ini. Oleh karena itu penulis mengundang pembaca untuk memberikan saran serta kritik yang dapat membangun penulis. Penulis berharap adanya kritik konstruktif dan saran yang membangun dari semua pihak.

Akhir kata saya, berharap semoga dengan selesainya laporan penelitian Tugas Akhir ini dapat memberikan manfaat bagi semua pihak serta menambah wawasan bagi pemikiran kita semua. Terima kasih.

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN REVISI	ii
SURAT PERNYATAAN	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
UCAPAN TERIMA KASIH	vii
DAFTAR ISI	x
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	4
1.3. Tujuan Penelitian	5
1.4. Batasan Masalah	5

1.5.	Metode Penelitian	5
1.5.1.	Teknik Pengumpulan Data	5
1.5.2.	Metode Perancangan	6
1.6.	Sistematika Penulisan	6
BAB II LANDASAN TEORI		8
2.1.	Sistem Pakar	8
2.1.1.	Pengertian Sistem Pakar	8
2.1.2.	Kelebihan dan Kekurangan Sistem Pakar	9
2.1.3.	Ciri-ciri Sistem Pakar	10
2.1.4.	Konsep Dasar Sistem Pakar	10
2.1.5.	Teknik Inferensi <i>Forward Chaining</i>	11
2.1.6.	Basis Pengetahuan	15
2.2.	Diagnosis	16
2.2.1.	Pengertian Diagnosis	16
2.2.2.	Batasan Diagnosis	17
2.2.3.	Jenis-jenis Diagnosis	17
2.3.	<i>Coronavirus Disease 2019 (COVID-19)</i>	18
2.3.1.	COVID-19	18

2.3.2.	Sejarah COVID-19	19
2.3.3.	Gejala Yang Ditimbulkan Saat Terkena COVID-19	20
2.3.4.	Cara Mencegah Terkena COVID-19	22
2.3.5.	Perkembangan Penanganan COVID-19 di Indonesia	22
2.3.5.1.	Data Penyebaran COVID-19 di Jawa Barat	24
2.4.	Aplikasi Berbasis WEB	26
2.4.1.	Pengertian Aplikasi Berbasis WEB	26
2.4.2.	Jenis-jenis WEB	27
2.4.3.	Pengertian PHP (<i>Hypertext Preprocessor</i>)	28
2.4.4.	Pengertian HTML (<i>Hypertext Markup Language</i>)	28
2.4.5.	Pengertian MySQL (<i>My Structure Query Language</i>)	29
2.5.	<i>Flowchart</i>	29
2.6.	UML (<i>Unified Modeling Language</i>)	31
2.6.1.	<i>Use case Diagram</i>	32
2.6.1.1.	Tujuan <i>Use case Diagram</i>	32
2.6.1.2.	Fungsi <i>Use case Diagram</i>	32
2.6.1.3.	Manfaat <i>Use case Diagram</i>	33
2.6.1.4.	Karakteristik <i>Use case Diagram</i>	33

2.6.1.5.	Simbol <i>Use case</i> Diagram	34
2.6.2.	<i>Activity</i> Diagram	35
2.6.2.1.	Tujuan <i>Activity</i> Diagram	35
2.6.2.2.	Simbol <i>Activity</i> Diagram	35
2.6.3.	<i>Class</i> Diagram	36
2.6.3.1.	Fungsi dan Manfaat <i>Class</i> Diagram	37
2.6.3.2.	Simbol <i>Class</i> Diagram	37
2.6.4.	<i>Sequence</i> Diagram	38
2.6.4.1.	Tujuan <i>Sequence</i> Diagram	39
2.6.4.2.	Simbol <i>Sequence</i> Diagram	39
2.6.5.	<i>Component</i> Diagram	41
2.6.5.1.	Fungsi <i>Component</i> Diagram	41
2.6.5.2.	Simbol <i>Component</i> Diagram	42
2.6.6.	<i>Deployment</i> Diagram	43
2.6.6.1.	Tujuan <i>Deployment</i> Diagram	43
2.6.6.2.	Simbol <i>Deployment</i> Diagram	44
2.6.7.	<i>Communication</i> Diagram	44
2.6.7.1.	Tujuan <i>Communication</i> Diagram	45

2.6.7.2.	Simbol <i>Communication</i> Diagram	45
2.6.8.	<i>Statechart</i> Diagram	46
2.6.8.1.	Simbol <i>Statechart</i> Diagram	47
2.7.	ERD (<i>Entity Relationship</i> Diagram)	48
2.7.1.	Simbol <i>Entity Relationship</i> Diagram	49
2.7.2.	Derajat Relasi dan Kardinalitas	50
2.8.	Pengembangan Perangkat Lunak	50
2.8.1.	Metode Pengembangan Perangkat Lunak	50
2.9.	<i>Testing</i>	53
2.9.1.	Pengertian <i>Testing</i>	53
2.9.2.	<i>Behavioral Test (Black Box Test)</i>	53
BAB III PEMBAHASAN		54
3.1.	<i>Communication</i>	54
3.1.1.	Teknik Pengumpulan Data	54
3.1.1.1.	Studi Lapangan	54
3.1.1.2.	Studi Literatur	57
3.1.2.	Analisis Sistem	59
3.1.3.	Analisis Proses	59

3.1.4.	Analisis Masalah	61
3.1.5.	Analisis Sistem yang Diusulkan	61
3.1.5.1.	Analisis Pengguna	64
3.1.5.2.	Analisis Perangkat	64
3.2.	<i>Planning</i>	66
3.3.	<i>Modeling</i>	68
3.3.1.	Desain Perangkat Lunak	68
3.3.2.	Perancangan Sistem	68
3.3.2.1.	<i>Use case</i> Diagram	68
3.3.2.2.	<i>Activity</i> Diagram	72
3.3.2.3.	<i>Class</i> Diagram	78
3.3.2.4.	<i>Sequence</i> Diagram	79
3.3.2.5.	<i>Statechart</i> Diagram	84
3.3.2.6.	<i>Component</i> Diagram	88
3.3.2.7.	<i>Deployment</i> Diagram	88
3.3.3.	<i>Entity Relationship</i> Diagram	89
3.3.4.	Pohon Keputusan	91
3.3.5.	<i>Rule/Aturan</i>	92

3.3.6. Perancangan Antarmuka	95
BAB IV IMPLEMENTASI DAN UJI COBA	99
4.1. <i>Construction (Code & Test)</i>	99
4.1.1. Implementasi	99
4.1.1.1. Implementasi <i>Hardware & Software</i>	99
4.1.1.2. Implementasi Antar Muka	100
4.1.1.3. Implementasi <i>Database</i>	103
4.1.1.4. <i>Testing</i>	104
BAB V PENUTUP	111
5.1. Kesimpulan	111
5.2. Saran	111
DAFTAR PUSTAKA	112
LAMPIRAN-LAMPIRAN	116

DAFTAR TABEL

TABEL: 2.1. Simbol-simbol <i>Use case</i> Diagram	34
TABEL: 2.2. Simbol-simbol <i>Activity</i> Diagram	36
TABEL: 2.3. Simbol-simbol <i>Class</i> Diagram	38
TABEL: 2.4. Simbol-simbol <i>Sequence</i> Diagram	40
TABEL: 2.5. Simbol-simbol <i>Component</i> Diagram	42
TABEL: 2.6. Simbol-simbol <i>Deployment</i> Diagram	44
TABEL: 2.7. Simbol-simbol <i>Communication</i> Diagram	45
TABEL: 2.8. Simbol-simbol <i>Statechart</i> Diagram	47
TABEL: 2.9. Simbol-simbol <i>Entity Relationship</i> Diagram	49
TABEL: 3.1. Referensi Penelitian	58
TABEL: 3.2. Penjadwalan Penelitian (<i>Planning</i>)	67
TABEL: 3.3. Deskripsi Perancangan <i>Use case</i> Diagram	69
TABEL: 3.4. Skenario <i>Use case</i> Diagram	71
TABEL: 4.1. Rencana Pengujian	105
TABEL: 4.2. Tabel Hasil Pengujian <i>Black Box</i>	106

DAFTAR GAMBAR

GAMBAR: 2.1. Model <i>Forward Chaining</i> Berbasis Aturan	13
GAMBAR: 2.2. Operasi Sistem <i>Forward Chaining</i>	14
GAMBAR: 2.3. Peta Penyebaran COVID-19 di Jawa Barat	24
GAMBAR: 2.4. Jumlah Kasus COVID-19 di Jawa Barat	25
GAMBAR: 2.5. Jumlah Kasus COVID-19 di Jawa Barat	25
GAMBAR: 2.6. Jumlah Kasus COVID-19 di Jawa Barat	26
GAMBAR: 2.7. Simbol-simbol Pada Diagram <i>Flowchart</i>	32
GAMBAR: 2.8. Model <i>Waterfall Pressman</i>	51
GAMBAR: 3.1. Hasil Wawancara Secara Tidak Langsung Dengan Pakar Kesehatan	55
GAMBAR: 3.2. Hasil Wawancara Secara Tidak Langsung Dengan Pakar Kesehatan	56
GAMBAR: 3.3. Hasil Wawancara Secara Tidak Langsung Dengan Pakar Kesehatan	57
GAMBAR: 3.4. <i>Flowchart</i> Proses Memahami Yang Sedang Berjalan	60
GAMBAR: 3.5. <i>Flowchart</i> Media Untuk Memahami Yang Diusulkan	62

GAMBAR: 3.6. <i>Flowchart</i> Sistem Munculnya Pertanyaan Diagnosa	63
GAMBAR: 3.7. <i>Use case</i> Diagram Aplikasi Sistem Pakar Penyakit COVID-19	70
GAMBAR: 3.8. <i>Activity</i> Diagram Menampilkan Pertanyaan Diagnosa	73
GAMBAR: 3.9. <i>Activity</i> Diagram Menampilkan Solusi	74
GAMBAR: 3.10. <i>Activity</i> Diagram Menampilkan Tampilan Awal WEB	75
GAMBAR: 3.11. <i>Activity</i> Diagram Info Sejarah Penyakit COVID-19	76
GAMBAR: 3.12. <i>Activity</i> Diagram Menampilkan Seputar Pertanyaan COVID-19	77
GAMBAR: 3.13. <i>Activity</i> Diagram Menampilkan Rumah Sakit Rujukan COVID-19 19	78
GAMBAR: 3.14. Class Diagram Aplikasi Sistem Pakar COVID-19	79
GAMBAR: 3.15. <i>Sequence</i> Diagram Pertanyaan Diagnosa	80
GAMBAR: 3.16. <i>Sequence</i> Diagram Solusi	80
GAMBAR: 3.17. <i>Sequence</i> Diagram Menampilkan Menu Tampilan Awal WEB	81
GAMBAR: 3.18. <i>Sequence</i> Diagram Menampilkan Sejarah COVID-19	82
GAMBAR: 3.19. <i>Sequence</i> Diagram Menampilkan Menu Seputar Pertanyaan COVID-19	83

GAMBAR: 3.20. <i>Sequence</i> Diagram Menampilkan Rumah Sakit Rujukan COVID-19	84
GAMBAR: 3.21. <i>Statechart</i> Diagram Pertanyaan Diagnosa	85
GAMBAR: 3.22. <i>Statechart</i> Diagram Solusi	85
GAMBAR: 3.23. <i>Statechart</i> Diagram Menu Awal WEB	86
GAMBAR: 3.24. <i>Statechart</i> Diagram Menu Sejarah COVID-19	86
GAMBAR: 3.25. <i>Statechart</i> Diagram Menampilkan Pertanyaan Seputar COVID-19	87
GAMBAR: 3.26. <i>Statechart</i> Diagram Menampilkan Menu Rumah Sakit Rujukan COVID-19	87
GAMBAR: 3.27. <i>Component</i> Diagram Aplikasi Sistem Pakar COVID-19	88
GAMBAR: 3.28. <i>Deployment</i> Diagram Aplikasi Sistem Pakar COVID-19	89
GAMBAR: 3.29. <i>Entity Relationship</i> Diagram Aplikasi Sistem Pakar COVID-19	90
GAMBAR: 3.30. Pohon Keputusan Aplikasi Sistem Pakar COVID-19	91
GAMBAR: 3.31. Perancangan Tampilan Main Menu Aplikasi	95
GAMBAR: 3.32. Perancangan Tampilan Isi Data	96
GAMBAR: 3.33. Perancangan Tampilan Pertanyaan Diagnosa	96
GAMBAR: 3.34. Perancangan Tampilan Hasil Diagnosa	97

GAMBAR: 3.35. Perancangan Tampilan Sejarah Tentang COVID-19	97
GAMBAR: 3.36. Perancangan Tampilan Seputar Pertanyaan COVID-19	98
GAMBAR: 3.37. Perancangan Tampilan Rumah Sakit Rujukan COVID-19	98
GAMBAR: 4.1. Tampilan Awal WEB	100
GAMBAR: 4.2. Tampilan Untuk Memulai Diagnosa	101
GAMBAR: 4.3. Tampilan Pertanyaan Diagnosa	101
GAMBAR: 4.4. Tampilan Menu Sejarah	102
GAMBAR: 4.5. Tampilan Menu Pertanyaan	102
GAMBAR: 4.6. Tampilan Menu Rumah Sakit	103
GAMBAR: 4.7. Relasi Tabel Aplikasi Sistem Pakar COVID-19	103
GAMBAR: 4.8. Tabel Pertanyaan Aplikasi Sistem Pakar COVID-19	104
GAMBAR: 4.9. Tabel Kesimpulan Aplikasi Sistem Pakar COVID-19	104
GAMBAR: 4.10. Tabel Solusi Aplikasi Sistem Pakar COVID-19	104

BAB I

PENDAHULUAN

1.1. Latar Belakang

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.

Menurut **WHO** (*World Health Organization*), gejala umum yang dapat dirasakan jika terjangkit virus ini adalah demam, rasa lelah, dan batuk kering. Beberapa pasien mungkin mengalami rasa nyeri dan sakit, hidung tersumbat, pilek, hilangnya rasa atau bau, sakit tenggorokan atau diare, Gejala-gejala yang dialami biasanya bersifat ringan dan muncul secara bertahap. Beberapa orang yang terinfeksi tidak menunjukkan gejala apa pun dan tetap merasa sehat.

Sebagian besar (sekitar 80%) orang yang terinfeksi berhasil pulih tanpa perlu perawatan khusus. Sekitar 1 dari 6 orang yang terjangkit **COVID-19** menderita sakit parah dan kesulitan bernapas. Orang-orang lanjut usia (lansia) dan orang-orang dengan kondisi medis yang sudah ada sebelumnya seperti tekanan darah tinggi, gangguan jantung atau diabetes, punya kemungkinan lebih besar mengalami sakit lebih serius. Mereka yang mengalami demam, batuk dan kesulitan bernapas sebaiknya mencari pertolongan medis.

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).

Pada saat ini virus tersebut sudah merebak ke hampir seluruh dunia menjadi hal yang sangat ditakuti oleh semua orang. Banyaknya para tenaga medis atau dokter yang mampu dijadikan salah satu pilihan untuk melakukan konsultasi kesehatan bukanlah hal yang susah untuk ditemukan. Namun disamping itu juga

para pasien kadang kala harus berfikir panjang untuk datang langsung ke rumah sakit atau klinik, hal ini dikarenakan bahayanya virus tersebut sangat mudah menular sehingga pakar kesehatan atau dokter untuk menganjurkan setiap orang membatasi kegiatan di luar rumah untuk mencegah terjadinya penularan virus.

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*.

Forward chaining mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Mesin inferensi yang menggunakan *forward chaining* mencari aturan-aturan inferensi sampai menemukan satu dari antecedent (dalil hipotesa atau klausa *IF - THEN*) yang benar. Ketika aturan tersebut ditemukan maka mesin pengambil keputusan dapat membuat kesimpulan, atau konsekuensi (klausa *THEN*), yang menghasilkan informasi tambahan yang baru dari data yang disediakan. Mesin akan mengulang melalui proses ini sampai sasaran ditemukan.

Forward chaining adalah contoh konsep umum dari pemikiran yang dikendalikan oleh data (*data-driven*) yaitu, pemikiran yang mana fokus

perhatiannya dimulai dari data yang diketahui. *Forward chaining* bisa digunakan didalam agen untuk menghasilkan kesimpulan dari persepsi-persepsi yang datang, seringkali tanpa *query* yang spesifik.

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*.

Pada permasalahan diatas maka penulis akan membangun aplikasi berbasis **WEB** dengan bertujuan untuk konsultasi layaknya dengan seorang pakar kesehatan untuk mendiagnosa penyakit virus **COVID-19** melalui *website* yang terhubung dengan koneksi internet. Yang mempunyai metode *forward chaining* yang dimana data-data yang dikumpulkan sesuai fakta yang terjadi saat ini dengan sudah tervalidasi oleh pakar kesehatannya secara langsung. Maka judul yang akan diambil adalah “**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT VIRUS COVID-19 BERBASIS WEB**”.

1.2. Identifikasi Masalah

Berdasarkan uraian latar belakang masalah yang dikemukakan diatas yang menjadi pokok masalah dalam tugas akhir ini adalah :

Bagaimana cara untuk mengetahui terindikasi penyakit **COVID-19**?

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

Untuk mengetahui terkena atau tidak terhadap penyakit **COVID-19**.

1.4. Batasan Masalah

Agar masalah yang diteliti tidak keluar dan menyimpang, maka diperlukannya suatu batasan masalah tersebut adalah :

1. Aplikasi ini hanya untuk mendiagnosa penyakit menular **COVID-19** pada manusia saja.
2. Aplikasi ini mendukung berbasis dekstop tidak *mobile phone*.
3. Harus terhubung dengan internet.

1.5. Metode Penelitian

1.5.1. 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.

1.5.2. 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)*.

1.6. Sistematika Penulisan

Sistematika penulisan ini meliputi 5 bagian yaitu terdiri dari :

BAB I : PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, identifikasi masalah, tujuan penelitian, batasan masalah, metode penelitian, metode perancangan, dan sistematika penulisan tugas akhir.

BAB II : LANDASAN TEORI

Pada bab ini akan dijelaskan beberapa teori yang mendasari pemecahan masalah dan teori-teori yang berkaitan langsung dengan pemecahan masalah serta pembahasannya diperoleh

dari buku-buku referensi dan catatan-catatan lainnya yang dikembangkan.

BAB III : ANALISIS MASALAH DAN PERANCANGAN PROGRAM

Pada bab ini berisi tentang analisis sistem untuk membuat aplikasi meliputi langkah-langkah pembangunan media untuk mendiagnosis penyakit virus **COVID-19** pada manusia berbasis *web* yang meliputi : *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*, *State chart diagram*, *component diagram* dan *deployment diagram* struktur menu aplikasi.

BAB IV : IMPLEMENTASI DAN UJI COBA

Pada bab ini berisi tentang pengujian akhir dengan menggunakan *testing* aplikasi *black box*, serta tampilan *layout* aplikasi yang dibuat dalam tugas akhir.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisikan kesimpulan dan saran-saran dari hasil analisis dan pembahasan yang dapat diajukan sehubungan dengan permasalahan tugas akhir.

BAB II

LANDASAN TEORI

2.1. Sistem Pakar

2.1.1. Pengertian Sistem Pakar

Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* yaitu suatu aplikasi komputer yang ditujukan 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. Sistem ini disebut sistem pakar karena fungsi dan perannya sama seperti seorang ahli yang harus memiliki pengetahuan, pengalaman dalam memecahkan suatu persoalan. Sistem biasanya berfungsi sebagai kunci penting yang akan membantu suatu sistem pendukung keputusan atau sistem pendukung eksekutif. Sistem pakar adalah suatu cabang dari (AI) Artificial Intelligent yang membuat penggunaan secara luas Knowledge yang khusus untuk penyelesaian masalah tingkat manusia yang pakar (Arhami, 2005).

2.1.2. Kelebihan dan Kekurangan Sistem Pakar

A. Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan (Hayadi, 2018), diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seseorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Mempermudah akses pengetahuan seorang pakar.
6. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
7. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

B. Kekurangan sistem pakar

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.

2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

2.1.3. Ciri-ciri Sistem Pakar

Ciri-ciri sistem pakar adalah sebagai berikut:

1. Terbatas pada *domain* keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah atau *rule* tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.

2.1.4. Konsep Dasar Sistem Pakar

Sistem pakar terdiri dari beberapa konsep yang harus dimilikinya. Konsep dasar dari suatu sistem pakar sebagai berikut:

1. Keahlian

Adalah suatu pengetahuan khusus yang diperoleh dari latihan, belajar dan pengetahuan. Pengetahuan dapat berupa fakta, teori, aturan, strategi global untuk memecahkan masalah.

2. Ahli (*Expert*)

Melibatkan kegiatan mengenali dan memformulasikan permasalahan,

memecahkan masalah secara cepat dan tepat, menerangkan pemecahannya, belajar dari pengalaman, merestrukturisasi pengetahuan, memecahkan aturan serta menentukan relevansi. Mentransfer keahlian (*Transferring Expertise*) Adalah proses pentransferan keahlian dari seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan pakar. Pengetahuan tersebut ditempatkan ke dalam sebuah komponen yang dinamakan basis pengetahuan.

3. Menyimpulkan aturan (*Inferencing Rule*)

Merupakan kemampuan komputer yang telah diprogram. Penyimpulan ini dilakukan oleh mesin inferensi yang meliputi prosedur tentang penyelesaian masalah.

4. Peraturan (*Rule*)

Diperlukan karena mayoritas dari sistem pakar bersifat *rule-based systems*, yang berarti pengetahuan disimpan dalam bentuk peraturan.

5. Kemampuan menjelaskan (*Explanation Capability*)

Adalah karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan tertentu dianjurkan atau tidak dianjurkan.

2.1.5. Teknik Inferensi *Forward Chaining*

Pada sistem pakar berbasis *rule*, domain pengetahuan dipresentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data dipresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian ini.

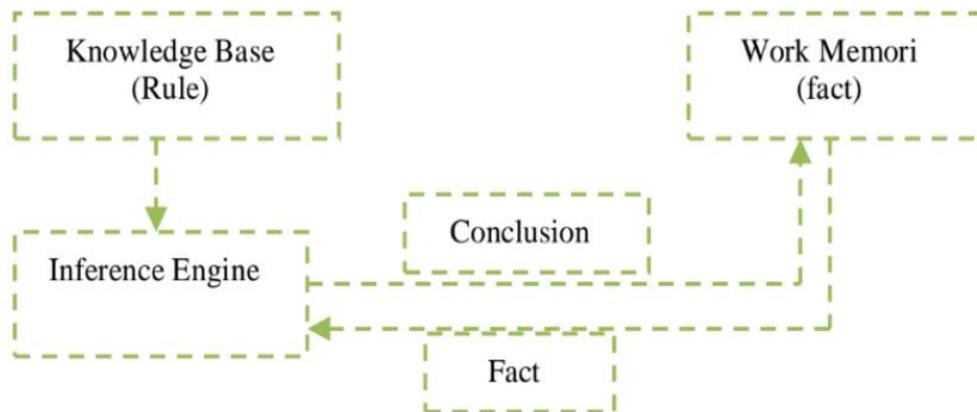
A. Forward Chaining

Metode *forward chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan (Hayadi, 2018).

Forward chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*. Maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam *database*. Setiap *rule* hanya boleh dieksekusi sekali saja (Hayadi, 2018).

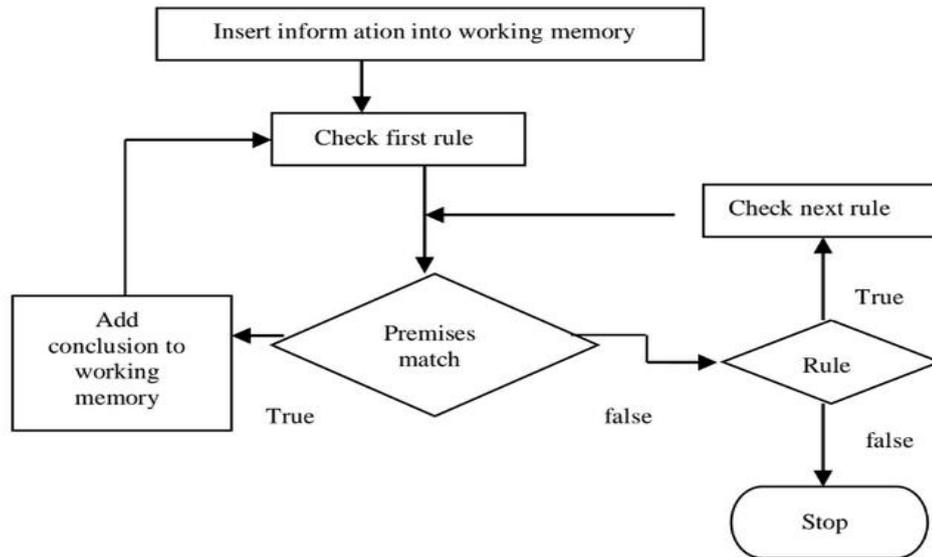
Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan dari seorang pakar dalam bidang tertentu dengan pandangan untuk memecahkan masalah atau memberikan nasihat. Pakar manusia (*human expert*) adalah seseorang yang mempunyai penguasaan yang mendalam terhadap suatu masalah. Berdasarkan pengalamannya, pakar manusia mengembangkan kemampuannya dalam memecahkan masalah secara lebih efisien dan efektif. Sistem pakar juga harus dapat menjelaskan alasan dari setiap langkah dalam mencapai suatu tujuan (*goal*) dan menjawab pertanyaan tentang solusi yang dicapainya, seperti halnya seseorang pakar manusia (Hayadi, 2018).

Sistem pakar *forward chaining* berbasis aturan dapat dimodelkan seperti gambar berikut ini:



GAMBAR: 2.1. Model *Forward Chaining* Berbasis Aturan (Hayadi, 2018)

Operasi dari sistem *forward chaining* dimulai dengan memasukan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Operasi tersebut dapat digambarkan seperti gambar.



GAMBAR: 2.2. Operasi Sistem *Forward Chaining* (Hayadi, 2018)

Langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu:

1. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.

2. Pendefinisian Data Input

Sistem *forward chaining* memerlukan data awal untuk memulai inferensi.

3. Pendefinisian Struktur Pengendalian Data

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

4. Penulisan Kode Awal

Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.

5. Pengujian Sistem

Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

6. Perancangan Antarmuka

Antarmuka adalah satu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

7. Pengembangan Sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan *prototype* sistem.

8. Evaluasi Sistem

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

2.1.6. Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah.

Ada dua bentuk pendekatan basis pengetahuan:

1. Penalaran berbasis aturan (*rule-based reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

2. Penalaran berbasis kasus (*case-based reasoning*)

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu bentuk ini juga digunakan bila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

2.2. Diagnosis

2.2.1. Pengertian Diagnosis

Diagnosis sering digunakan oleh dokter untuk menyebutkan suatu penyakit yang diderita seorang pasien atau keadaan yang dapat menyebabkan seorang pasien memerlukan atau mencari atau menerima asuhan medis guna memperoleh pelayanan pengobatan, pencegahan memburuknya masalah kesehatan atau juga

untuk peningkatan kesehatan. Sedangkan diagnosis utama adalah penyakit, cacat, luka atau keadaan sakit yang utama dari pasien yang dirawat dirumah sakit.

2.2.2. Batasan Diagnosis

Batasan diagnosis utama adalah :

- a) Diagnosis yang ditentukan setelah cermat dikaji.
- b) Menjadi alasan untuk dirawat
- c) Menjadi fakta arahan atau pengobatan

2.2.3. Macam-macam Diagnosis

Macam – macam Diagnosis Menurut **WHO**.

a. *Principal* Diagnosis

Adalah diagnosis yang ditegakkan pasien setelah dikaji yang terutama bertanggung jawab menyebabkan *admission* pasien.

b. *Other* Diagnosis

Adalah diagnosis *selin principal* diagnosis yang menggambarkan suatu kondisi dimana pasien mendapatkan pengobatan atau dimana dokter mempertimbangkan kebutuhan-kebutuhan untuk memasukkannya dalam pemeriksaan lebih lanjut.

c. *Complication*

Suatu diagnosis tambahan yang menggambarkan suatu kondisi yang muncul setelah dimulainya observasi dan perawatan di rumah sakit yang mempengaruhi

perjalanan pasien atau asuhan medis yang dibutuhkan oleh pasien.

2.3. *Coronavirus Disease 2019 (COVID-19)*

2.3.1. COVID-19

Penyakit *coronavirus disease 2019 (COVID-19)* adalah penyakit menular yang disebabkan oleh *coronavirus* yang baru ditemukan. Sebagian besar orang yang terinfeksi virus **COVID-19** akan mengalami penyakit pernapasan ringan hingga sedang dan sembuh tanpa memerlukan perawatan khusus. Orang yang lebih tua, dan mereka yang memiliki masalah medis mendasar seperti penyakit kardiovaskular, diabetes, penyakit pernapasan kronis, dan kanker lebih mungkin mengembangkan penyakit serius.

Cara terbaik untuk mencegah dan memperlambat penularan diberitahukan dengan baik tentang virus **COVID-19**, penyakit yang disebabkan dan bagaimana penyebarannya. Lindungi diri Anda dan orang lain dari infeksi dengan mencuci tangan atau sering menggunakan alkohol berbasis gosok (*hand sanitizer*) dan tidak menyentuh wajah anda.

Virus **COVID-19** menyebar terutama melalui tetesan air liur atau keluar dari hidung ketika orang yang terinfeksi batuk atau bersin, jadi penting bagi Anda juga berlatih etiket pernapasan (misalnya, dengan batuk pada siku yang tertekuk).

Saat ini, tidak ada vaksin atau perawatan khusus untuk **COVID-19**. Namun, ada banyak uji klinis yang sedang berlangsung mengevaluasi perawatan potensial.

WHO akan terus memberikan informasi terbaru segera setelah temuan klinis tersedia.

2.3.2. Sejarah COVID-19

Diawal tahun 2020, dunia digemparkan dengan merebaknya virus baru yaitu *Coronavirus* jenis baru (**SARS-CoV-2**) dan penyakitnya disebut *Coronavirus Disease 2019 (COVID-19)*. Diketahui, asal mula virus ini berasal dari Wuhan, Tiongkok. Ditemukan pada akhir Desember tahun 2019. Sampai saat ini sudah dipastikan terdapat 65 negara yang telah terjangkit virus satu ini menurut (Data **WHO**, 1 Maret 2020).

Pada awalnya data epimiologi menunjukkan 66% pasien berkaitan atau terpajan dengan satu pasar *seafood* atau *live market* di Wuhan, Provinsi Hubei, Tiongkok (Huang, et.al., 2020). Sampel isolat dari pasien diteliti dengan hasil menunjukkan adanya infeksi *Coronavirus*, jenis *Betacoronavirus* tipe baru, diberi nama *2019 novel Coronavirus (2019-nCoV)*. Pada tanggal 11 Februari 2020, **WHO** (*World Health Organization*) memberi nama virus baru tersebut *Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2)* dan nama penyakitnya sebagai *Coronavirus Disease 2019 (COVID-19)* (**WHO**, 2020). Pada mulanya transmisi virus ini belum dapat ditentukan apakah dapat melalui manusia-manusia. Jumlah kasus terus bertambah seiring dengan waktu. Kejadian luar biasa oleh *Coronavirus* bukanlah kejadian yang pertama kali. Tahun 2002 *Severe Acute Respiratory Syndrome (SARS)* disebabkan oleh *SARS-Coronavirus (SARS-CoV)*

dan penyakit *Middle East Respiratory Syndrome* (**MERS**) tahun 2012 disebabkan oleh *MERS-Coronavirus* (**MERS-CoV**) dengan total akumulatif kasus sekitar 10.000 (1000-an kasus **MERS** dan 8000-an kasus **SARS**). Mortalitas akibat **SARS** sekitar 10% sedangkan **MERS** lebih tinggi yaitu sekitar 40%.

2.3.3. Gejala Yang Timbul Saat Terkena COVID-19

Menurut **WHO** (*World Health Organization*), gejala umum yang dapat dirasakan jika terjangkit virus ini adalah demam, rasa lelah, dan batuk kering. Beberapa pasien mungkin mengalami rasa nyeri dan sakit, hidung tersumbat, pilek, sakit tenggorokan atau diare, gejala-gejala yang dialami biasanya bersifat ringan dan muncul secara bertahap. Beberapa orang yang terinfeksi tidak menunjukkan gejala apa pun dan tetap merasa sehat. Sebagian besar (sekitar 80%) orang yang terinfeksi berhasil pulih tanpa perlu perawatan khusus. Sekitar 1 dari 6 orang yang terjangkit **COVID-19** menderita sakit parah dan kesulitan bernapas. Orang-orang lanjut usia (lansia) dan orang-orang dengan kondisi medis yang sudah ada sebelumnya seperti tekanan darah tinggi, gangguan jantung atau diabetes, punya kemungkinan lebih besar mengalami sakit lebih serius. Mereka yang mengalami demam, batuk dan kesulitan bernapas sebaiknya mencari pertolongan medis.

COVID-19 mempengaruhi orang yang berbeda dengan cara yang berbeda. Kebanyakan orang yang terinfeksi akan mengembangkan penyakit ringan hingga sedang dan sembuh tanpa dirawat di rumah sakit.

Gejala paling umum:

- Demam.
- Batuk kering.
- Kelelahan.
- Kehilangan rasa atau bau.

Gejala yang kurang umum:

- Sakit dan nyeri.
- Sakit tenggorokan.
- Diare.
- Konjungtivitis.
- Sakit kepala.
- Ruam pada kulit, atau perubahan warna jari tangan atau kaki.

Gejala serius:

- Kesulitan bernafas atau sesak nafas.
- Nyeri atau tekanan pada dada.
- Kehilangan bicara atau bergerak.

Cari pertolongan medis segera jika anda memiliki gejala serius. Selalu menelepon sebelum mengunjungi dokter atau fasilitas kesehatan anda. Orang dengan gejala ringan yang dinyatakan sehat harus mengelola gejalanya di rumah. Rata-rata dibutuhkan 5-6 hari dari saat seseorang terinfeksi virus untuk menunjukkan gejala, namun dapat memakan waktu hingga 14 hari

2.3.4. Cara Mencegah Terkena COVID-19

Menurut **WHO**, Untuk mencegah infeksi dan memperlambat transmisi **COVID-19**, lakukan hal sebagai berikut:

- Cuci tangan anda secara teratur dengan sabun dan air, atau bersihkan dengan *hand sanitizer*.
- Jaga jarak minimal 1 meter antara anda dan orang lain yang terkena batuk atau bersin (*Physical Distancing/Sosial Distancing*).
- Hindari menyentuh wajah anda, jika tangan kotor.
- Tutupi mulut dan hidung anda dengan menggunakan masker (sehat atau terkena bersin dan pilek).
- Tetap di rumah jika anda merasa tidak sehat.
- Jangan merokok dan aktivitas lain yang melemahkan paru-paru.
- Berlatih menjaga jarak dengan menghindari perjalanan yang tidak perlu dan menjauh dari kerumunan banyak orang.

2.3.5 Perkembangan Penanganan COVID-19 di Indonesia

Perkembangan penanganan pandemi **COVID-19** di Indonesia, sejalan dengan perkembangan pandemi **COVID-19** di tingkat global atau tingkat dunia. Juru Bicara Satgas Penanganan **COVID-19** Prof Wiku Adisasmito menyajikan data perkembangan penanganan **COVID-19** tingkat dunia sebagaimana disampaikan World Health Organization (**WHO**).

Yang menyatakan bahwa kasus positif **COVID-19** tingkat dunia menurun sebesar 17% pada minggu lalu, dan dibarengi penurunan kematian sebesar 10%. "Ini sejalan dengan perkembangan penanganan **COVID-19** di Indonesia pada beberapa minggu terakhir," jelasnya dalam keterangan pers perkembangan penanganan **COVID-19** di Graha BNPB, Selasa (16/2/2022) yang disiarkan kanal *YouTube* BNPB Indonesia.

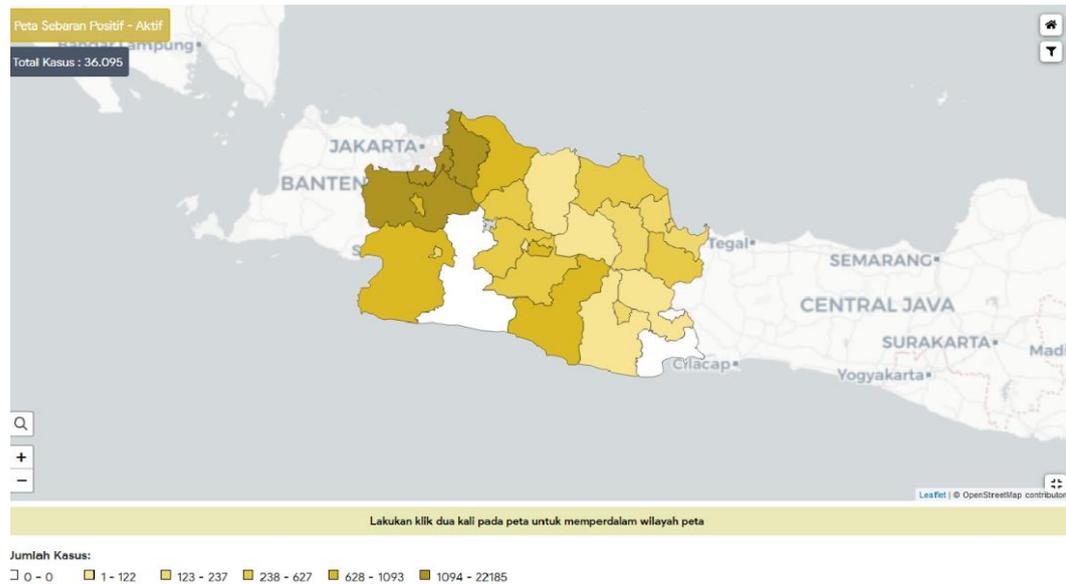
Perkembangan di Indonesia dalam beberapa minggu terakhir juga menunjukkan penurunan kasus positif dan juga penurunan tingkat keterisian tempat tidur di rumah sakit rujukan **COVID-19**. Dan adanya dampak dari kebijakan yang diambil pemerintah yaitu Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) tingkat kabupaten/kota Pulau Jawa - Bali dalam 4 minggu terakhir, serta dilanjutkan PPKM tingkat desa dan kelurahan (mikro).

Diharapkan kebijakan tersebut dapat berdampak positif pada perkembangan penanganan **COVID-19**. Perkembangan positif ini harus terus ditingkatkan. Karena jika perkembangan kasus di tingkat global dan di Indonesia terus secara konsisten mengalami penurunan, maka penanganan **COVID-19** di Indonesia akan berhasil dengan tidak adanya penambahan kasus sama sekali.

"Hal ini membutuhkan kerjasama yang baik dari masyarakat terutama dalam mematuhi Protokol Kesehatan secara disiplin, serta komitmen pemerintah dalam berupaya meningkatkan kualitas testing, tracing dan treatment," pesan Wiku (covid19.go.id, 24 Februari 2021).

2.3.5.1 Data Penyebaran COVID-19 di Jawa Barat

Berikut ini adalah peta penyebaran **COVID-19** di Jawa Barat.



GAMBAR: 2.3. Peta Penyebaran **COVID-19** di Jawa Barat

(pikobar.jabarprov.go.id, 2021)

Pada sumber gambar dapat diketahui penyebarannya, dimana dari jumlah kasus bisa dilihat dengan warna dari tipis hingga tebal. Dimana warna tipis jumlah kasusnya sedikit dan warna yang paling tebal jumlah kasusnya paling banyak.

Berikut adalah gambar kasus **COVID-19** di kota dan kabupaten di Jawa Barat, di *update* pada tanggal 24 Februari 2021.

https://pikobar.jabarprov.go.id/distribution-case

Izinkan notifikasi mengirim pesan [Allow](#)

Pusat Informasi & Koordinasi COVID-19
Provinsi Jawa Barat

Home Data Berita FAQ Kontak Donasi Bantuan Sosial [Download App](#)

Tabel Kasus Covid-19 di Jawa Barat

Data yang ditampilkan berdasarkan: Kota/Kabupaten di Jawa Barat

[Terkonfirmasi](#) [Suspek](#) [Kontak Erat](#) [Probable](#)

Nama Kota/Kabupaten	Terkonfirmasi			Isolasi/ Dalam Perawatan			Selesai Isolasi/ Sembuh			Meninggal		
	23 Februari 2021	H-7	Total	23 Februari 2021	H-7	Total	23 Februari 2021	H-7	Total	23 Februari 2021	H-7	Total
TOTAL KESELURUHAN	4.334	26.563	202.510	2.724	9.440	36.095	1.610	17.123	164.309	0	0	2.106
KOTA/KAB BELUM TERIDENTIFIKASI	4.334	22.178	24.467	4.334	22.178	22.185	0	0	2.282	0	0	0
KAB. BOGOR	0	0	9.590	-731	-1.814	1.782	731	1.814	7.640	0	0	168
KAB. SUKABUMI	0	0	4.006	-3	-170	932	3	170	3.040	0	0	34
KAB. CIANJUR	0	0	1.610	-47	-1.132	0	47	1.132	1.608	0	0	2
KAB. BANDUNG	0	29	8.373	0	-1.061	261	0	1.090	8.074	0	0	38

GAMBAR: 2.4. Jumlah Kasus COVID-19 di Jawa Barat

(pikobar.jabarprov.go.id, 2021)

Izinkan notifikasi mengirim pesan [Allow](#)

Pusat Informasi & Koordinasi COVID-19
Provinsi Jawa Barat

Home Data Berita FAQ Kontak Donasi Bantuan Sosial [Download App](#)

Nama Kota/Kabupaten	Terkonfirmasi			Isolasi/ Dalam Perawatan			Selesai Isolasi/ Sembuh			Meninggal		
	23 Februari	H-7	Total	23 Februari	H-7	Total	23 Februari	H-7	Total	23 Februari	H-7	Total
TOTAL KESELURUHAN	4.334	26.563	202.510	2.724	9.440	36.095	1.610	17.123	164.309	0	0	2.106
KAB. GARUT	0	60	6.952	-24	-852	1.077	24	912	5.738	0	0	137
KAB. TASIKMALAYA	0	0	1.845	-15	-59	61	15	59	1.724	0	0	60
KAB. CIAMIS	0	0	2.176	0	-165	105	0	165	2.016	0	0	55
KAB. KUNINGAN	0	98	3.369	-127	-551	354	127	649	2.999	0	0	16
KAB. CIREBON	0	115	5.397	-9	-34	164	9	149	5.061	0	0	172
KAB. MAJALENGKA	0	0	1.689	-48	-50	125	48	50	1.525	0	0	39
KAB. SUMEDANG	0	0	1.638	0	-8	8	0	8	1.593	0	0	37
KAB. INDRAMAYU	0	56	3.284	-26	-608	341	26	664	2.900	0	0	43
KAB. SUBANG	0	0	1.832	0	-190	50	0	190	1.768	0	0	14
KAB. PURWAKARTA	0	340	3.306	-18	-295	512	18	635	2.735	0	0	59
KAB. KARAWANG	0	98	11.052	-104	-366	790	104	464	10.025	0	0	237
KAB. BEKASI	0	677	15.683	-9	-284	1.686	9	961	13.937	0	0	60

GAMBAR: 2.5. Jumlah Kasus COVID-19 di Jawa Barat

(pikobar.jabarprov.go.id, 2021)

Izinkan notifikasi mengirim pesan [Allow](#)

Pusat Informasi & Koordinasi COVID-19
Provinsi Jawa Barat

Home Data Berita FAQ Kontak Donasi Bantuan Sosial [Download App](#)

Nama Kota/Kabupaten	Terkonfirmasi			Isolasi/ Dalam Perawatan			Selesai Isolasi/ Sembuh			Meninggal		
	23 Februari	H-7	Total	23 Februari	H-7	Total	23 Februari	H-7	Total	23 Februari	H-7	Total
TOTAL KESELURUHAN	4.334	26.563	202.510	2.724	9.440	36.095	1.610	17.123	164.309	0	0	2.106
KAB. PURWAKARTA	0	340	3.306	-18	-295	512	18	635	2.735	0	0	59
KAB. KARAWANG	0	98	11.052	-104	-366	790	104	464	10.025	0	0	237
KAB. BEKASI	0	677	15.683	-9	-284	1.686	9	961	13.937	0	0	60
KAB. BANDUNG BARAT	0	0	3.517	0	-354	620	0	354	2.873	0	0	24
KAB. PANGANDARAN	0	0	812	-97	-447	0	97	447	811	0	0	1
KOTA BOGOR	0	449	9.924	-2	-880	637	2	1.329	9.167	0	0	120
KOTA SUKABUMI	0	127	2.444	-3	41	199	3	86	2.203	0	0	42
KOTA BANDUNG	0	0	11.518	-133	-1.022	721	133	1.022	10.708	0	0	89
KOTA CIREBON	0	0	2.805	0	-1.050	180	0	1.050	2.589	0	0	36
KOTA BEKASI	0	1.641	28.911	0	879	2.021	0	762	26.695	0	0	195
KOTA DEPOK	0	695	28.636	-197	-1.691	1.155	197	2.386	27.159	0	0	322
KOTA CIMAHI	0	0	3.774	0	-241	1	0	241	3.719	0	0	54

GAMBAR: 2.6. Jumlah Kasus **COVID-19** di Jawa Barat

(pikobar.jabarprov.go.id, 2021)

Pada sumber gambar jumlah kasus **COVID-19** di Jawa Barat termasuk yang terkonfirmasi, isolasi/dalam perawatan, setelah isolasi/sembuh, dan meninggal dunia. Dengan menampilkan data-data yang bermacam-macam dan pertumbuhannya sangat fruktatif atau berubah-ubah dengan sebagaimana kondisi yang sedang dialami di wilayahnya masing-masing.

2.4. Aplikasi berbasis WEB

2.4.1. Pengertian Aplikasi berbasis WEB

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** yang lainnya disebut *hyperlink*.

Domain adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet, misalnya *lintau.com*, *yahoo.com*, *google.com*, *ephi.web.id* dan lain-lain. Untuk mendapat sebuah domain kita harus melakukan *register* pada *register-register* yang ditentukan.

Istilah lain yang sering ditemui sehubungan dengan *website* adalah *homepage*. *Homepage* adalah halaman awal sebuah *domain*. Misalnya, Anda membuka *website* *www.lintau.com*, halaman pertama yang muncul disebut dengan *homepage*, jika anda meng-klik menu-menu yang ada dan meloncat ke lokasi lainnya, disebut *web page*, sedangkan keseluruhan isi/*content* domain disebut *website*.

2.4.2. Jenis-jenis WEB

Seiringan dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokan jenis **WEB**, lebih diarahkan berdasarkan pada fungsi, sifat dan bahasa pemrograman yang digunakan.

Jenis-jenis **WEB** berdasarkan sifatnya adalah:

1. *Website* dinamis, merupakan *website* yang menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Misalnya *website* berita, seperti, *www.kompas.com*, *www.detik.com*, *www.polindg.ac.id*, dan lain-lain.
2. *Website* statis, merupakan *website* yang *content*nya sangat jarang diubah.

Misalnya, web profile organisasi, dan lain-lain.

2.4.3. Pengertian PHP (*Hypertext Preprocessor*)

PHP sering dipakai para *programmer* untuk membuat situs **WEB** yang bersifat dinamis karena gratis dan berguna dalam merancang aplikasi **WEB**. **PHP** singkatan dari “*Hypertext Preprocessor*” yaitu bahasa pemrograman **WEB Server Side** yang bersifat *open source*. **PHP** merupakan *script* yang terintegrasi dengan **HTML** dan berada *server (server side HTML embedded scripting)*. **PHP** adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script PHP* dieksekusi pada *server* dimana *script* tersebut dijalankan.

2.4.4. Pengertian HTML (*Hypertext Markup Language*)

HTML (*Hyper Text Markup Language*) merupakan bahasa pemrograman **WEB** yang memiliki sintak atau aturan tertentu dalam menuliskan *script* atau kode-kode, sehingga *browser* dapat menampilkan informasi dengan membaca kode-kode **HTML**. **HTML** (*Hyper Text Markup Language*) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada **WEB browser**. Tag-tag **HTML** selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana x tag **HTML** itu seperti b, i, u, dan lain-lain.

2.4.5. Pengertian MySQL

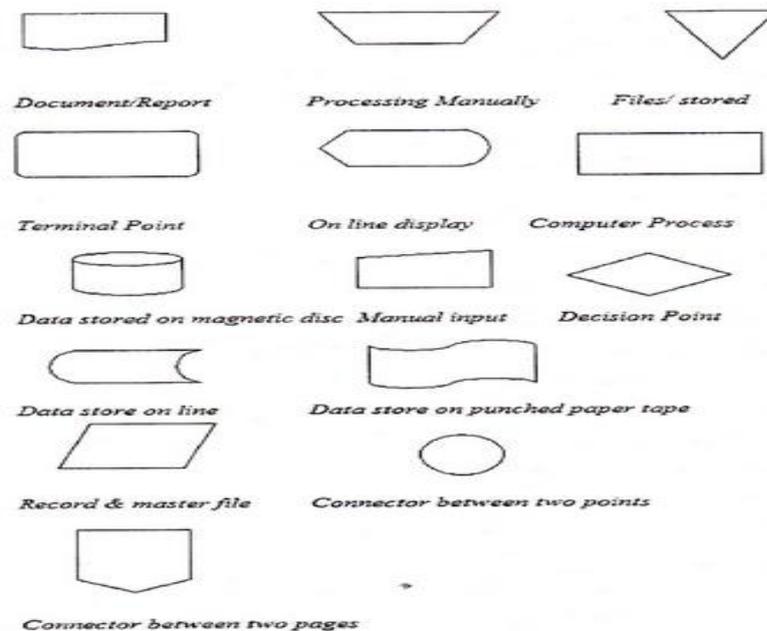
Sebuah *website* yang dinamis membutuhkan tempat penyimpanan data agar pengunjung dapat memberikan komentar, saran, dan masukan atas *website* yang dibuat. Tempat penyimpanan data berupa informasi dalam sebuah tabel disebut dengan *database*. Program yang digunakan untuk mengolah *database* adalah **MySQL** yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur, dan menampilkan data. **MySQL** (*My Structure Query Language*) adalah salah satu *Database Management System* (**DBMS**) dari sekian banyak **DBMS** seperti *Oracle*, **MS SQL**, *Postagre SQL*, dan lainnya. **MySQL** berfungsi untuk mengolah *database* menggunakan bahasa **SQL**. **MySQL** bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman **PHP** juga sangat mendukung atau support dengan database **MySQL**.

2.5. Flowchart

Flowchart atau bagan alir, awal mulanya memang berkembang dari industri komputer yaitu untuk menggambarkan urutan proses penyelesaian masalah. Namun seperti kata pepatah lama bahwa “satu gambar adalah sejuta kata” maka suatu *flowchart* dapat dengan mudah menjelaskan suatu urutan proses yang relatif rumit bila diuraikan dalam kata-kata. Dengan visualisasi maka adanya *bottleneck* (penumpukan) atau ketidak-efisienan dari suatu proses dapat terdeteksi untuk dilakukan perbaikan. Oleh karena itu *flowchart* juga diterima di kalangan lain

(manufaktur, sains, militer, manajemen, dan sebagainya). *Flowchart* sangat berguna khususnya untuk menjelaskan urutan proses yang pelaksanaannya mempunyai banyak *option* pilihan atau percabangan. *Flowchart* dalam pemrograman komputer kadangkala diibaratkan sebagai cetak-biru suatu rencana bangunan yang akan dikerjakan para insinyur. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. Jelasnya dengan *flowchart* maka mengkomunikasikan jalannya program ke orang lain (bukan program) akan lebih mudah. Jadi pembuatan *flowchart* adalah mutlak khususnya untuk melengkapi cara dokumentasi atau *listing* program komputer yang relatif rumit, sehingga dokumentasi tersebut suatu saat nanti apabila diperlukan kembali dapat dengan mudah dipahami.

Simbol-simbol *flowchart* adalah sebagai berikut:



GAMBAR: 2.7. Simbol-simbol Pada Diagram *Flowchart* (Irwansyah, dkk, 2004)

2.6. UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. **UML** memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep **UML** ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. **UML** bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya (Muslihudin, dan Oktafianto, 2016). Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi *error* yang terjadi? Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan **UML**.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisis dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam bidang investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, *retail*, *sales* dan *supplier*.

2.6.1. Use case Diagram

Use case diagram adalah suatu urutan interaksi yang saling berkaitan antara sistem dan aktor. *Use case* dijalankan melalui cara menggambarkan tipe interaksi antara *user* suatu program (sistem) dengan sistemnya sendiri. *Use case* melalui sebuah cerita yang mana sebuah sistem itu dipakai. *Use case* juga dipakai untuk membentuk perilaku (*behaviour*) sistem yang akan dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang sudah ada. *Use Case* diagram ini memaparkan proses kegiatan dan proses bisnis yang dilakukan oleh pengguna (aktor). Pada diagram *Use Case* ini, setiap proses digambarkan kedalam sebuah *Use Case* berbentuk oval dengan relasi yang ada didalamnya.

2.6.1.1. Tujuan Use case Diagram

Berikut ini tujuan dari *Use case*, antara lain:

1. Memetakan kebutuhan sistem.
2. Merepresentasikan interaksi pengguna terhadap sistem.
3. Untuk mengetahui kebutuhan diluar sistem.

2.6.1.2. Fungsi Use case Diagram

Adapun fungsi dari *Use case* diagram, sebagai berikut.

1. Dapat menggambarkan urutan aktivitas proses yang ada pada suatu sistem.

2. Dapat menggambarkan proses bisnis dan juga urutan aktivitas yang ada dalam sebuah proses.

2.6.1.3. Manfaat *Use case* Diagram

Sedangkan manfaat *Use case* sendiri adalah:

1. Untuk memudahkan hubungan dengan menggunakan *domain expert* dan juga *end user*.
2. Adanya *Interface* yang harus dimiliki oleh sebuah sistem.
3. Memberikan kepastian pemahaman yang pas, tentang *requirement* atau juga kebutuhan sebuah sistem.
4. Dapat digunakan untuk mengidentifikasi, siapa yang sedang berinteraksi dengan sistem, dan juga apa yang harus dilakukan untuk sistem tersebut.
5. Biasanya digunakan untuk verifikasi.

2.6.1.4. Karakteristik *Use case* Diagram

Adapun ciri-ciri dari *Use case*, antara lain:

1. *Use case* merupakan interaksi atau dialog antara sistem dan pengguna (aktor), termasuk peralihan pesan dan tindakan yang dilakukan oleh suatu sistem.
2. *Use case* diprakarsai oleh pengguna dan mungkin melibatkan peran aktor yang lain. *Use case* harus menyediakan nilai minimal kepada satu pengguna.

3. *Use case* bisa mempunyai perluasan yang menjelaskan tindakan khusus dalam interaksi atau *Use case* lain mungkin disisipkan.
4. *Use case class* mempunyai objek yang disebut skenario. Skenario menyatakan urutan pesan dan reaksi tunggal.

2.6.1.5. Simbol *Use case* Diagram

Berikut ini simbol/notasi/komponen dan penjelasan dari *Use case* diagram yang ada didalam tabel ini.

TABEL: 2.1. Simbol-simbol *Use case* Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Merupakan peran orang, sistem yang lain, atau alat ketika berhubungan dengan <i>Use case</i> .
	<i>Use Case</i>	Abstraksi dari penghubung antara aktor dengan <i>Use case</i> .
	<i>Association</i>	Abstraksi dari penghubung antara aktor dengan <i>Use case</i> .
	Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>Use case</i> .
	<i>Include</i>	Menunjukkan bahwa suatu <i>Use case</i> seluruhnya merupakan fungsionalitas dari <i>Use case</i> lainnya.
	<i>Extend</i>	Menunjukkan bahwa suatu <i>Use case</i> merupakan tambahan fungsional dari <i>Use case</i> lainnya jika suatu kondisi terpenuhi.

2.6.2. Activity Diagram

Activity diagram ialah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir. *Activity* diagram juga dapat menjelaskan metode paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram adalah *state* diagram khusus, yang mana *state* ini berfungsi sebagai *action* dan sebagian besar transisi *ditrigger* oleh akhir *state* sebelumnya (*internal processing*).

2.6.2.1. Tujuan Activity Diagram

Inilah tujuan dari *activity* diagram, antara lain:

1. Menjelaskan proses bisnis dan urutan aktivitas dalam sebuah proses.
2. Dipakai pada bisnis modeling untuk memperlihatkan urutan aktifitas proses bisnis.
3. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram (DFD)* pada perancangan terstruktur.
4. Bermanfaat apabila anda membuat diagram ini terlebih dahulu dalam memodelkan suatu metode agar mudah memahami proses secara keseluruhan.
5. *Activity* diagram dibuat berdasarkan beberapa *Use case* pada *Use case* diagram.

2.6.2.2. Simbol Activity Diagram

Inilah simbol atau komponen dari *activity* diagram, antara lain:

TABEL: 2.2. Simbol-simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/ <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/ <i>Join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.6.3. *Class Diagram*

Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. *Class diagram* merupakan alur jalannya *database* pada sebuah sistem. *Class diagram* merupakan penjelasan proses *database* dalam suatu program. Dalam sebuah laporan sistem maka *class diagram* ini wajib ada.

2.6.3.1. Fungsi dan Manfaat *Class Diagram*

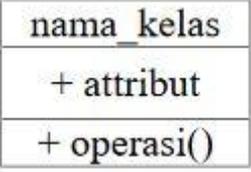
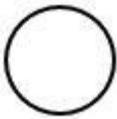
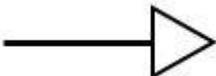
Adapun fungsi dan manfaat dari *class diagram* adalah sebagai berikut.

1. Menjelaskan suatu model data untuk program informasi, tidak peduli apakah model data tersebut sederhana maupun kompleks.
2. Dengan menguasai *class diagram* maka akan meningkatkan pemahaman mengenai gambaran umum skema dari suatu program.
3. Mampu menyatakan secara visual akan kebutuhan spesifik suatu informasi serta dapat berbagi informasi tersebut ke seluruh bisnis.
4. Dengan *Class diagram* dapat dibuat bagan secara terperinci dan jelas, dengan cara memperhatikan kode spesifik apa saja yang dibutuhkan oleh program. Hal ini mampu mengimplementasikan ke struktur yang dijelaskan.
5. *Class diagram* mampu memberikan penjelasan implementasi-independen dari suatu jenis program yang digunakan, kemudian dilewatkan diantara berbagai komponennya.

2.6.3.2. Simbol *Class Diagram*

Berikut ini simbol-simbol dari *class diagram*.

TABEL: 2.3. Simbol-simbol *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Kelas pada struktur sistem.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Association</i>	Relasi antar <i>class</i> dengan arti umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Directed Association</i>	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Generalisasi</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	<i>Dependenc</i>	Relasi antarkelas dengan makna kebergantungan antarkelas
	<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

2.6.4. *Sequence Diagram*

Sequence diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence* diagram digunakan untuk menjelaskan perilaku pada sebuah

skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi. *Sequence* diagram berkaitan erat dengan *Use Case* diagram, yang mana 1 *Use Case* akan menjadi 1 *Sequence* diagram.

2.6.4.1. Tujuan *Sequence* Diagram

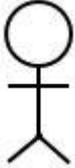
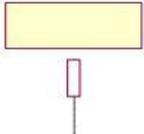
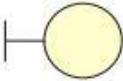
Berikut ini tujuan dari *sequence* digram adalah:

1. Menghubungkan *requirement* kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model *design*.
2. Merupakan diagram yang paling relevan untuk menguraikan model deskripsi *use case* menjadi spesifikasi *design*.
3. Analisa dan Desain, memfokuskan pada identifikasi metode didalam sebuah sistem.
4. Menganalisa, mendesain dan memfokuskan pada identifikasi sebuah metode yang digunakan sistem.
5. *Sequence* diagram ini dipakai untuk menjelaskan dan memodelkan *Use case*.
6. Berfungsi untuk memodelkan sebuah logika dari sebuah *method* operasi, *function* ataupun prosedur.
7. Berfungsi untuk memodelkan logika dari *service*.

2.6.4.2. Simbol *Sequence* Diagram

Adapun simbol-simbol atau komponen *sequence* digram sebagai berikut.

TABEL: 2.4. Simbol-simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem
	<i>Lifeline</i>	Menghubungkan objek selama sequence (<i>message</i> dikirim atau diterima dan aktifasinya).
	<i>General</i>	Merepresentasikan entitas tunggal dalam <i>sequence diagram</i> .
	<i>Boundary</i>	Berupa tepi dari sistem, seperti <i>user interface</i> atau suatu alat yang berinteraksi dengan sistem yang lain.
	<i>Control</i>	Elemen mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.
	Entitas	Elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model objek.
	<i>Activation</i>	Suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah sequence yang menunjukkan kapan sebuah objek mengirim atau menerima objek.
	<i>Message</i>	Berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan.

Lanjutan TABEL: 2.4. Simbol-simbol Class Diagram

Simbol	Nama	Keterangan
	<i>Message Entry</i>	Berfungsi untuk menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
	<i>Message to Self</i>	Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
	<i>Message Return</i>	Menggambarkan hasil dari pengiriman message dan digambarkan dengan arah dari kanan ke kiri.

2.6.5. Component Diagram

Diagram komponen atau *Component* diagram dibuat untuk menggambarkan struktur dan ketergantungan antara kumpulan komponen dalam sebuah sistem. Pengertian komponen dalam **UML** merupakan hal-hal fisik dari program yang akan digambarkan (modelkan) dan ada ketika sistem dijalankan.

2.6.5.1. Fungsi Component Diagram

Adapun fungsi/kegunaan dari *Component* diagram antara lain.

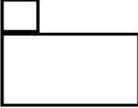
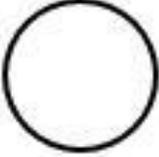
1. Sebagai *source code* perangkat lunak (*software*).
2. Komponen *executable* yang dilepas dan dipakai oleh pengguna.

3. Bagian dari basis data secara fisik.
4. Sistem yang harus bekerja sama dengan sistem yang lain.
5. Sebagai *framework* sistem, dimana dalam hal ini memiliki fungsi untuk mempermudah pengelolaan dan perawatan dari sistem.

2.6.5.2. Simbol *Component Diagram*

Berikut ini adalah simbol-simbol dari *Component* diagram.

TABEL: 2.5. Simbol-simbol *Component Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	<i>Package</i> merupakan sebuah simbol yang dipakai untuk tempat komponen.
	<i>Component</i>	<i>Component system</i> adalah simbol yang menjelaskan perangkat keras atau objek dalam sistem tersebut.
	<i>Dependency</i>	Simbol yang menjelaskan sebuah keterkaitan antara komponen, satu komponen dengan yang lain. Arah panah dalam simbol tersebut diarahkan pada komponen yang dipakai.
	<i>Interface</i>	Hal ini mirip dengan bahasa pemrograman berorientasi objek (PBO), dimana simbol ini dipakai untuk antar muka dengan fungsi supaya tidak langsung mengakses objek.

Lanjutan TABEL: 2.5. Simbol-simbol *Component Diagram*

Simbol	Nama	Keterangan
	<i>Link</i>	Simbol <i>link</i> ini dipakai untuk mengarahkan relasi antar komponen, jika suatu komponen memiliki relasi

2.6.6. *Deployment Diagram*

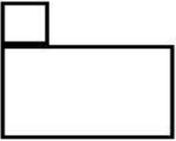
Deployment diagram adalah sebuah bahasa (UML) yang dipakai untuk menggambarkan, menspesifikasikan, dan mendokumentasikan metode yang terjadi pada suatu sistem software berorientasi objek yang akan dibangun. *Deployment* diagram mengarahkan konfigurasi komponen dalam metode eksekusi pada sistem.

2.6.6.1. Tujuan *Deployment Diagram*

Diagram *deployment* mewakili pandangan penyebaran sistem sehingga hanya akan ada satu diagram *deployment* untuk satu sistem. Diagram *deployment* terdiri dari *node*, *node* adalah *hardware* fisik yang dipakai untuk menyebarkan aplikasi. Diagram *deployment* banyak dipakai oleh *system enginer* (SE). Setiap *node* pada diagram *deployment* mewakili satu unit komputasi sistem yang dalam banyak hal adalah bagian dari perangkat keras. Diagram *deployment* umumnya memiliki *node* dan hubungan kebergantungan. Memungkinkan juga dalam diagram *deployment* terdapat komponen.

2.6.6.2. Simbol *Deployment Diagram*

TABEL: 2.6. Simbol-simbol *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	<i>Package</i> adalah sebuah bungkus dari satu atau lebih <i>node</i> .
	<i>Node</i>	<i>Node</i> biasanya mengacu pada <i>hardware</i> , perangkat lunak yang tidak dibuat sendiri (<i>software</i>). Jika didalam <i>node</i> disertai komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	<i>Dependency</i>	<i>Dependency</i> adalah kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
	<i>Link</i>	<i>Link</i> merupakan relasi antar <i>node</i> .

2.6.7. *Communication Diagram*

Communication diagram adalah suatu diagram yang menggambarkan struktur interaksi yang terdapat disekitar objek (seperti halnya *Sequence diagram*) dan hubungannya terhadap yang lainnya. *Communication diagram* lebih mengedepankan peran setiap objek dan bukan pada waktu pengiriman pesan (*message*). *Communication diagram* mirip dengan *Sequence diagram*. Yang membedakan *Communication diagram* dari yang lain adalah diagram yang mengarahkan objek-objek dan asosiasi dengan objek-objek yang lain dalam sistem bersamaan dengan interaksinya. Asosiasi ini tidaklah dijelaskan dalam *sequence diagram*.

2.6.7.1. Tujuan *Communication Diagram*

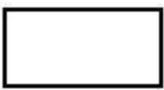
Adapun tujuan dari *Communication diagram* adalah :

1. Model pesan melalui antar peran atau objek yang mengirimkan operasi penggunaan dan fungsionalitas kasus.
2. Mekanisme model dalam desain arsitektur sistem.
3. Tangkap interaksi yang mengarahkan pesan yang dilewati antar peran dan objek dalam skenario kolaborasi.
4. Model skenario alternatif dalam kasus pemakaian atau operasi yang melibatkan kolaborasi dari berbagai interaksi dan objek.
5. Dukung pengidentifikasian objek (sebab kelas), atributnya (parameter pesan), dan operasi (pesan) yang ikut serta dalam *Use case*.

2.6.7.2. Simbol *Communication Diagram*

Berikut ini simbol-simbol *Communication diagram* antara lain.

TABEL: 2.7. Simbol-simbol *Communication Diagram*

Simbol	Nama	Keterangan
	Objek	Objek adalah <i>instance</i> dari sebuah kelas yang tersusun secara horizontal. Digambarkan sebagai sebuah kelas (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma (;).

Lanjutan TABEL: 2.7. Simbol-simbol *Communication Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> juga dapat berhubungan dengan objek , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbolnya sama pada <i>Actor Use Case</i> diagram.
	<i>Message</i>	<i>Message</i> digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek, seperti yang sudah digambarkan dan dijelaskan di atas.

2.6.8. *Statechart Diagram*

Statechart diagram adalah diagram yang menggambarkan perubahan dan transisi keadaan (dari satu status ke status lainnya) sebuah objek pada program sebagai akibat dari stimulan yang diterima. *Statechart* diagram berfungsi untuk memodelkan metode/*behavior (lifecycle)* sebuah kelas atau objek menunjukkan urutan kejadian sesaat (status) yang dilewati sebuah objek, transisi dari sebuah status ke status lainnya. *Statechart* diagram, bahkan seluruh *state* yang membolehkan objek-objek dalam kelas dapat dipunyai dan kejadian yang memicu status berubah. Perubahan dalam suatu status juga disebut transisi (*transition*). Suatu transisi juga dapat memiliki sebuah aksi yang disalurkan pada status, lebih spesifik apa yang harus dilakukan dalam hubungannya dengan transisi status. Pada diagram ini, perilaku program (sistem) ditunjukkan. Sebuah status ialah kondisi selama hidup objek atau interaksi selama memenuhi suatu kondisi, menjalankan suatu aksi, atau menunggu suatu kejadian.

2.6.8.1. Simbol *Statechart* Diagram

Berikut ini simbol-simbol dari *Statechart* diagram adalah:

TABEL: 2.8. Simbol-simbol *Statechart* Diagram

Simbol	Nama	Keterangan
	<i>State</i>	Digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisi saat itu.
	Titik awal (<i>start</i>)	Digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram <i>flowchart</i> .
	Titik Akhir(<i>end</i>)	Digunakan untuk menjelaskan/ menggambarkan akhir (<i>end</i>) dari kejadian dalam suatu diagram <i>statechart</i> .
[<i>Guard</i>]	<i>Guard</i>	Merupakan syarat transisi yang bersangkutan.
	<i>Point</i>	Digunakan untuk menggambarkan/ menjelaskan apakah akan masuk (<i>entry point</i>) ke dalam status atau keluar (<i>exit point</i>).
<i>event</i>	<i>Event</i>	Digunakan untuk menjelaskan kondisi yang menyebabkan sesuatu pada status.

2.7. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram adalah suatu teknik dokumentasi yang digunakan untuk menyajikan relasi antar entitas dalam sebuah sistem (Suprpto, 2021). **ERD** menyediakan cara untuk mendeskripsikan perancangan basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. **ERD** berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. **ERD** adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. **ERD** berbeda dengan **DFD** (*Data Flow Diagram*) yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan **ERD** merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data.

Pada model *Data Relation* hubungan antara file direlasikan dengan *relation key* yang merupakan kunci utama dari masing-masing file, adapun komponen utama dari *Entity Relationship Diagram* adalah:

1. *Entitas*

Kumpulan dari objek antara objek yang satu dengan objek yang lain dapat dibedakan.

2. *Relationship*

Hubungan yang terjadi antara satu *entity* atau lebih. *Entity Relationship* adalah relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam. Yaitu, *One to One* (1 : 1), *One to Many* (1 : M, M : 1), *Many to*

Many (M : M).

3. *Atribut*

Kumpulan elemen-elemen data yang membentuk suatu *entity* yang menyediakan penjelasan *detail* dalam *entity*.

2.7.1 Simbol ERD (*Entity Relationship Diagram*)

Berikut adalah simbol-simbol dari ERD (*Entity Relationship Diagram*)

TABEL: 2.9. Simbol-simbol *Entity Relationship Diagram*

Simbol	Nama	Penjelasan
	<i>Entity</i>	Digunakan untuk menggambarkan objek yang dapat diidentifikasi dalam lingkungan pemakai.
	<i>Attribute</i>	Menggambarkan elemen-elemen dari satu <i>entity</i> yang menggambarkan <i>entity</i> .
	<i>Relationship</i>	<i>Entity</i> dapat berhubungan satu sama lain. Hubungan ini disebut <i>relationship</i> .
	<i>Link</i>	Digunakan untuk menghubungkan <i>entity</i> dengan relasi dan <i>entity</i> dengan <i>atribut</i>

2.7.2 Derajat Relasi dan Kardinalitas

Cardinality Ration Constraint yang terjadi diantara dua himpunan dapat berupa:

1. Relasi satu ke satu (*One to One*)

Berarti setiap entitas pada suatu himpunan berhubungan paling banyak dengan satu entitas pada himpunan entitas lainnya, begitupun sebaliknya.

2. Relasi satu ke banyak (*One to Many*)

Berarti setiap entitas pada suatu himpunan berhubungan dengan banyak entitas pada himpunan entitas lainnya, tetapi tidak sebaliknya.

3. Relasi banyak ke satu (*Many to One*)

Berarti setiap entitas pada suatu himpunan berhubungan dengan paling banyak satu entitas pada himpunan lainnya, tetapi tidak sebaliknya.

4. Relasi banyak ke banyak (*Many to Many*)

Berarti setiap entitas pada suatu himpunan dapat berhubungan dengan banyak entitas pada himpunan entitas lainnya, dan begitupun sebaliknya.

2.8. Pengembangan Perangkat Lunak

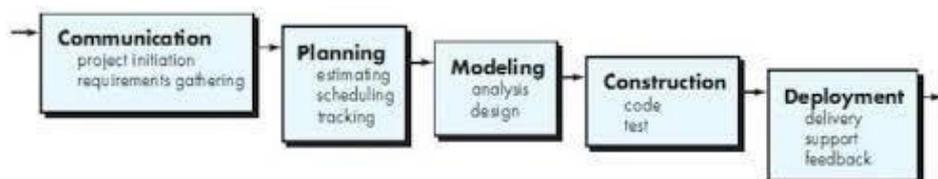
2.8.1. Metode Pengembangan Perangkat Lunak

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.

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model generik pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman, 2015).

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



GAMBAR: 2.8. Waterfall Pressman (Pressman, 2015)

1. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, sertamembantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

3. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang

sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17).

2.9. *Testing*

2.9.1. *Pengertian Testing*

Testing merupakan proses eksekusi program ataupun sistem secara *intens* untuk menemukan *error* (Romeo, 2003).

2.9.2. *Behavioral Test (Black Box Test)*

Black Box Testing yaitu pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kitanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya (*interface*), fungsionalitasnya. Tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (Astuti, 2018).

BAB III

PEMBAHASAN

3.1. *Communication*

3.1.1. Teknik Pengumpulan Data

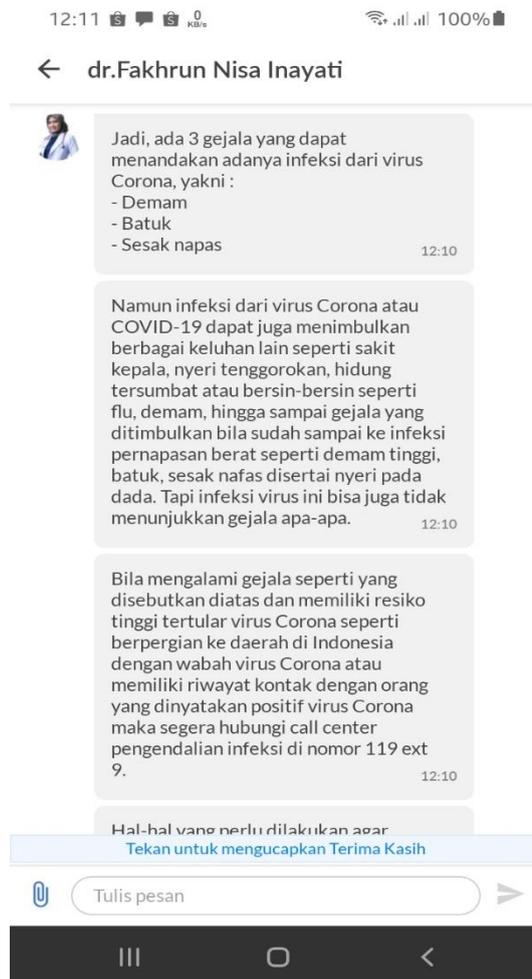
Adapun teknik pengumpulan data yang dilakukan penulis adalah :

3.1.1.1. Studi Lapangan

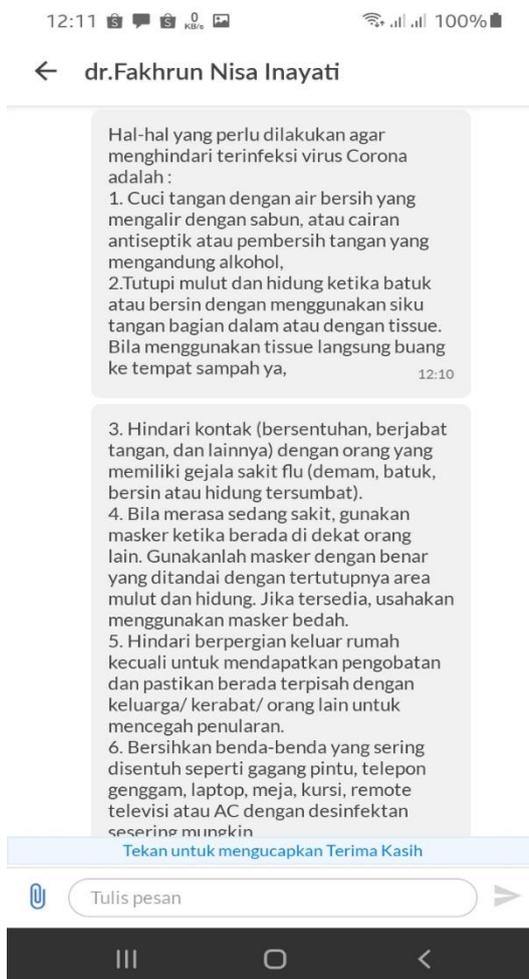
Studi lapangan meliputi:

- 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.

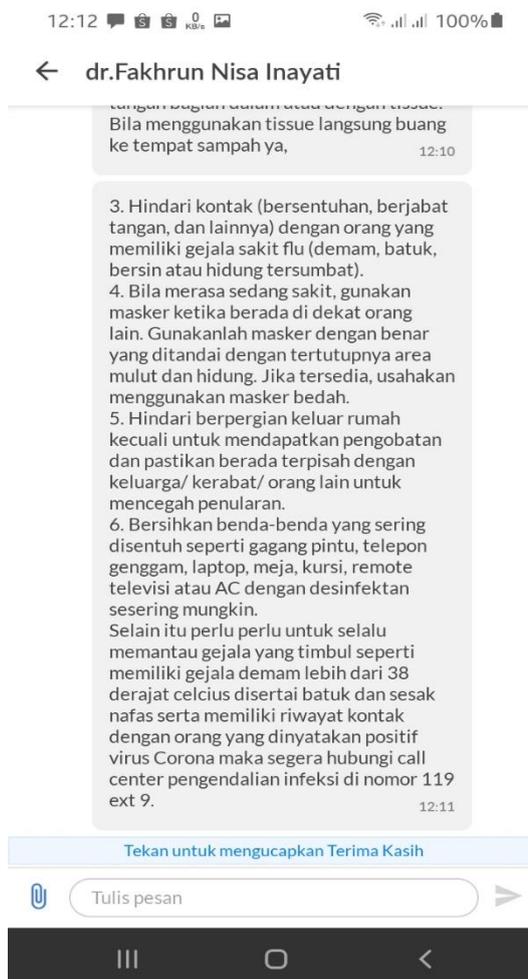
Berikut ini adalah hasil wawancara secara tidak langung dengan dr. Fakhrun Nisa Inayati melalui aplikasi kesehatan *online* tentang masalah yang berkaitan dengan penelitian ini yaitu tentang penyakit **COVID-19**.



GAMBAR: 3.1. Hasil Wawancara Secara Tidak Langsung dengan Pakar Kesehatan



GAMBAR: 3.2. Hasil Wawancara Secara Tidak Langsung dengan Pakar Kesehatan



GAMBAR: 3.3. Hasil Wawancara Secara Tidak Langsung dengan Pakar Kesehatan

3.1.1.2. Studi Literatur

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

TABEL: 3.1. Referensi Penelitian

No	Literatur	Pembahasan
1	Agus Rizal Ardy Hariandy Hamid “ <i>Social responsibility of medical journal: a concern for COVID-19 pandemic</i> ” <i>Medical Journal of Indonesia</i> Vol 29 No.1–3, 2020	Penelitian ini dibuat untuk kepedulian terhadap penyakit baru yaitu <i>coronavirus disease</i> 2019 (COVID-19) di Indonesia.
2	Jasper Fuk-Woo Chan*, Shuofeng Yuan*, Kin-Hang Kok*, Kelvin Kai-Wang To*, Hin Chu*, Jin Yang, Fanfan Xing, Jieling Liu, Cyril Chik-Yan Yip, Rosana Wing-Shan Poon, Hoi-Wah Tsoi, Simon Kam-Fai Lo, Kwok-Hung Chan, Vincent Kwok-Man Poon, Wan-Mui Chan, Jonathan Daniel Ip, Jian-Piao Cai, Vincent Chi-Chung Cheng, Honglin Chen, Christopher Kim-Ming Hui, Kwok-Yung Yuen “ <i>A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster</i> ” <i>Lancet</i> Vol 395 No.514-23, 2020	Penelitian ini dibuat untuk mengetahui bahwa sekelompok keluarga terkena pneumonia yang terkait dengan virus baru yaitu <i>coronavirus disease</i> 2019 (COVID-19) yang bisa menunjukkan penularan dari satu orang ke lainnya dengan studi kluster keluarga.
3	Anthony S. Fauci, M.D., H. Clifford Lane, M.D., and Robert R. Redfield, M.D. “ COVID-19 Navigating the Uncharted ” <i>The New England Journal of Medicine</i> Vol 382 No. 1268-1269, 2020	Penelitian ini dibuat untuk mengetahui kemunculan virus baru-baru ini yang dinamakan dengan <i>coronavirus disease</i> 2019 (COVID-19).
4	Thirumalaisamy P. Velavan, Christian G. Meyer “ The COVID-19 Epidemic ” <i>Tropical Medicine International Health</i> Vol 25 No. 278-280, 2020	Penelitian ini dibuat untuk mengetahui bahwa virus baru yaitu <i>coronavirus disease</i> 2019 (COVID-19) telah menyebar secara cepat ke seluruh penjuru dunia.

Lanjutan TABEL: 3.1. Referensi Penelitian

No	Literatur	Pembahasan
5	Angelo Carfi, MD., Roberto Bernabei, MD., Francesco Landi, MD, PhD., “ <i>Persistent Symptoms in Patients After Acute COVID-19</i> ” <i>JAMA</i> Vol 324 No. 603-605, 2020	Penelitian ini dibuat untuk mengetahui gejala yang ditimbulkan jika terkena <i>coronavirus disease</i> 2019 (COVID-19) akut.

3.1.2. Analisis Sistem

Analisis ini bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem yang akan dibangun. Bisa dikatakan analisis sistem merupakan penguraian dari suatu sistem untuk mengidentifikasi masalah-masalah dalam sistem yang akan dibuat atau sedang berjalan. Sehingga akan dapat diusulkan kebutuhan-kebutuhan untuk memperbaikinya.

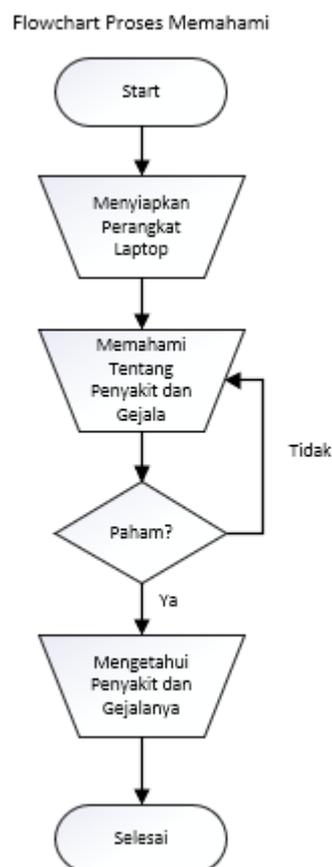
3.1.3. Analisis Proses

Analisis terhadap sistem yang sedang berjalan bertujuan untuk mengetahui bagaimana cara kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan sebagai landasan usulan perancangan. Tahap analisis diperlukan untuk mengetahui bagaimana proses mengetahui tentang penyakit tersebut beserta gejala yang ditimbulkan. Tahap perancangan ini, penulis menggambar sistem yang berjalan dalam bentuk *flowmap*.

Berdasarkan hasil analisis yang dilakukan dengan menggunakan metode observasi, proses pengenalan penyakit baru ini dilakukan sebagai berikut :

Pengguna dapat mengetahui informasi tentang penyakit baru ini dengan lengkap beserta gejalanya, dengan tampilan yang sangat simpel. Metode ini dipercaya dapat memudahkan pengguna dalam mengetahui terkena atau tidak terkenanya tentang penyakit baru ini dengan diagnosis gejala karena metode yang berhubungan langsung dengan kehidupan nyata.

Adapun penggambaran *flowchart* berdasarkan prosedur diatas digambarkan dalam gambar 3.4. berikut :



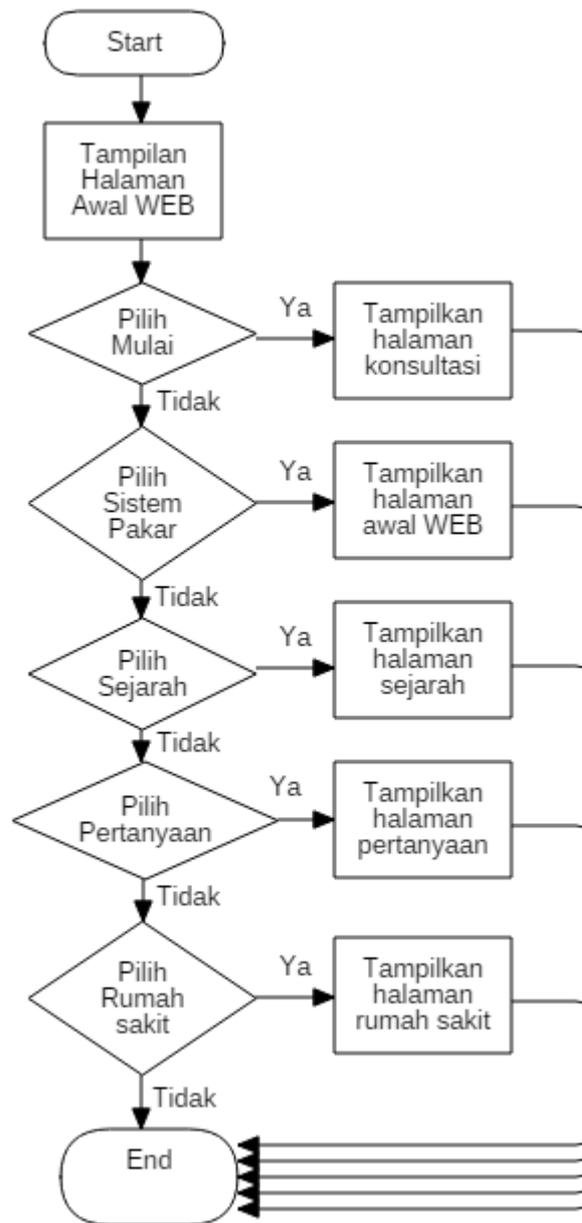
GAMBAR: 3.4. *Flowchart* Proses Memahami Yang Sedang Berjalan

3.1.4. Analisis Masalah

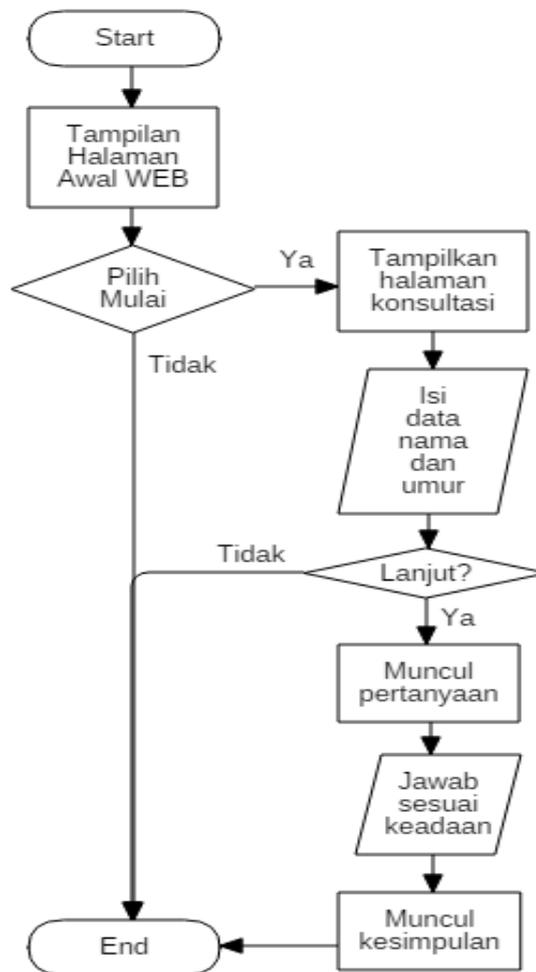
Analisis masalah merupakan langkah dimana langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang sedang berjalan. Adapun identifikasi masalah pada penelitian ini adalah bagaimana cara meningkatkan kembali ketertarikan dan menganalisis secara tepat mengenai terindikasi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

3.1.5. Analisis Sistem yang Diusulkan

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



GAMBAR: 3.5. Flowchart Media Untuk Memahami Yang Diusulkan



GAMBAR 3.6. *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.

Adapun sistem yang akan dibangun ini adalah sebuah sistem **WEB**, dimana aplikasi ini dibuat sebagai sarana menunjukkan arah pada pengguna dengan cara mengisi pertanyaan seputar gejala yang ditimbulkan atau dirasakan oleh pengguna,

dan mendapat kesimpulan di akhir pertanyaan dengan terkena penyakit tersebut atau tidak terkena. Sistem ini yang diusulkan dalam bentuk *flowchart*

Berikut ini adalah kebutuhan yang harus dipenuhi yang terdiri dari :

3.1.5.1. Analisis Pengguna

Agar penggunaan aplikasi dapat berjalan secara optimal dan sesuai dengan yang diharapkan, maka dibutuhkan kemampuan untuk menjalankan aplikasi. Dalam hal ini, kemampuan tersebut terutama adalah dapat mengoperasikan laptop dengan sistem operasi *Windows* atau yang lainnya.

3.1.5.2. Analisis Perangkat

Agar aplikasi dapat dijalankan secara optimal, ada beberapa hal yang harus diperhatikan. Diantaranya adalah perangkat keras (*hardware*) dan perangkat lunak (*software*) harus sesuai spesifikasi yang di rekomendasikan.

1. Perangkat Keras (*Hardware*)

Analisis perangkat dari segi perangkat keras (*Hardware*) merupakan perangkat yang mendukung fitur aplikasi bagi pengguna (*User*) sehingga aplikasi dapat berjalan secara optimal.

a. Pengguna (*User*)

Perangkat yang digunakan untuk fitur aplikasi bagi pengguna (*user*) adalah laptop minimal spesifikasi sebagai berikut :

1. *Processor* : **Dual Core 2 Ghz**
2. **RAM : 2 GB**
3. *Harddisk* : **80 GB**

b. Pengembang (*Developer*)

Perangkat keras minimum komputer yang dibutuhkan dalam pembuatan aplikasi sistem pakar ini adalah sebagai berikut :

1. *Processor* : **Quad Core 3.60 Ghz**
2. **RAM : 8 GB**
3. *Harddisk* : **1 TB**
4. **VGA : Radeon R7 2 GB**
5. Monitor : Resolusi layar 1366 x 768 keatas

2. Perangkat Lunak (*Software*)

Adapun analisis perangkat lunak (*Software*) merupakan perangkat yang mendukung fitur aplikasi bagi pengguna (*User*) dan pengembang sehingga aplikasi dapat berjalan secara optimal tidak terjadi kendala.

a. Pengguna (User)

Perangkat lunak yang digunakan untuk fitur aplikasi bagi pengguna (*user*) yaitu minimal sistem operasi *Windows 7*, atau yang lainnya keatas.

b. Pengembang (Developer)

Perangkat lunak minimum komputer yang dibutuhkan dalam pembuatan aplikasi ini adalah sebagai berikut :

1. Sistem operasi : *Windows 10 Pro*
2. Perangkat Lunak Pendukung :
 - a) *Notepad++ V. 7.8.7*
 - b) *Sublime Text V. 3.2.1 build 3207*
 - c) *Xampp V. 7.3.9-0*
 - d) *Mozilla FireFox/Google Chrome/Microsoft Edge* terbaru

3.2. Planning

Dalam tahap ini penelitian memfokuskan pada penjadwalan pengerjaan penelitian. Pada penelitian ini terdapat beberapa proses yang harus dilakukan dari tahap *communication* hingga *implementation* dan *testing* maka dari itu diperlukan penjadwalan yang tepat agar penelitian ini dapat selesai pada waktunya. Berikut penjadwalan penelitian berdasarkan aktifitas yang dilakukan dengan skala waktu. Tabel 3.2. dibawah ini akan menjelaskan penjadwalan penelitian.

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

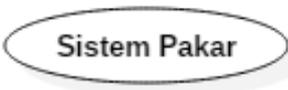
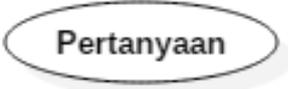
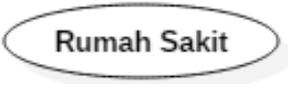
Dalam perancangan sistem perangkat lunak, penulis menggunakan model berorientasi objek dengan menggunakan notasi **UML** (*Unified Modelling Language*), diantaranya *Use case* Diagram, *Activity* Diagram, *Class* Diagram, *Statechart* Diagram, *Sequence* Diagram, *Component* Diagram, *Deployment* Diagram.

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 3.3.

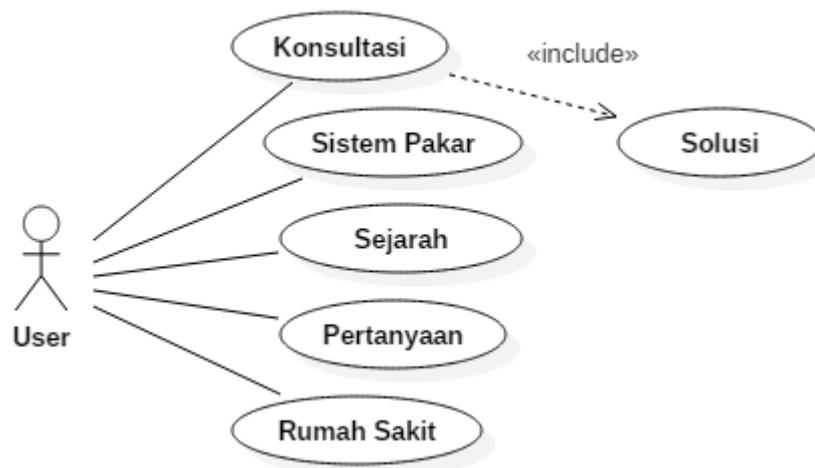
TABEL: 3.3. Deskripsi Perancangan *Use case* Diagram

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 WEB .
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.

Lanjutan TABEL: 3.3. Deskripsi Perancangan *Use case Diagram*

No	Notasi	Deskripsi
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 3.7. berikut ini :



GAMBAR: 3.7. *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: 3.4. Skenario *Use case* Diagram

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

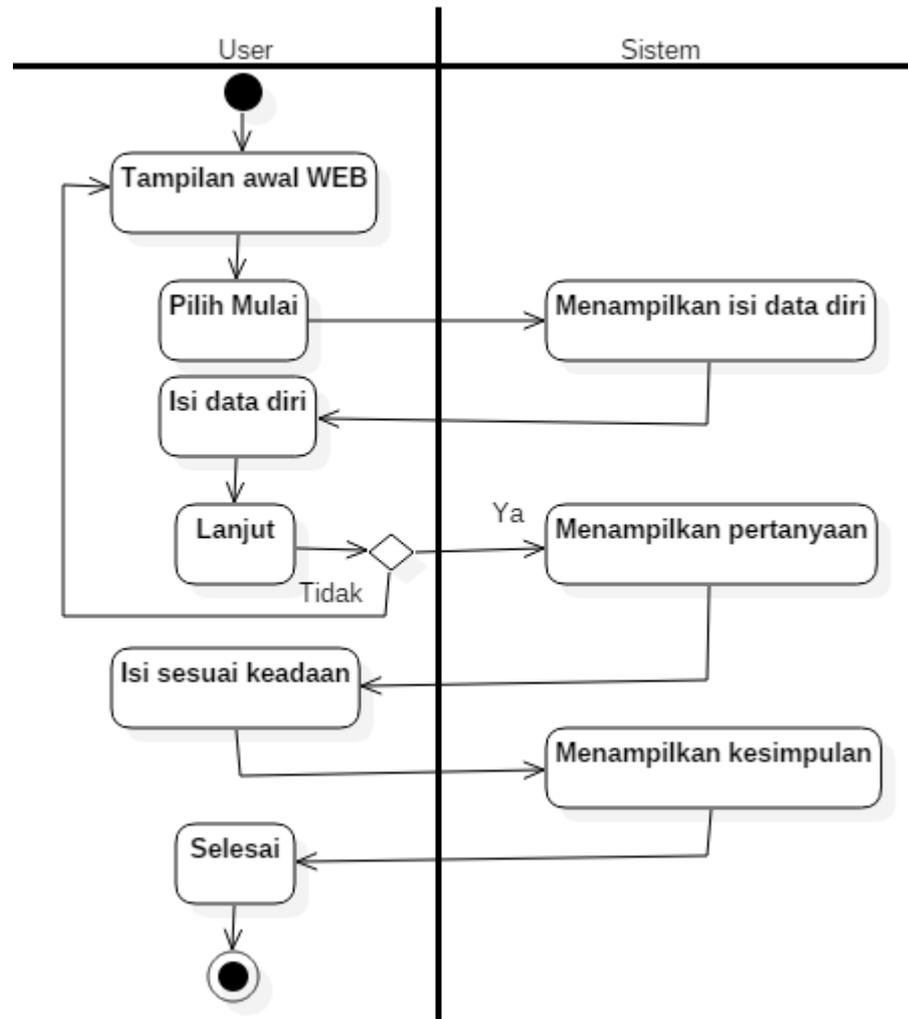
Lanjutan TABEL: 3.4. Skenario *Use case Diagram*

<i>Scenario</i>	<i>Step</i>	<i>Actor Activity</i>	<i>Respon System</i>
	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.

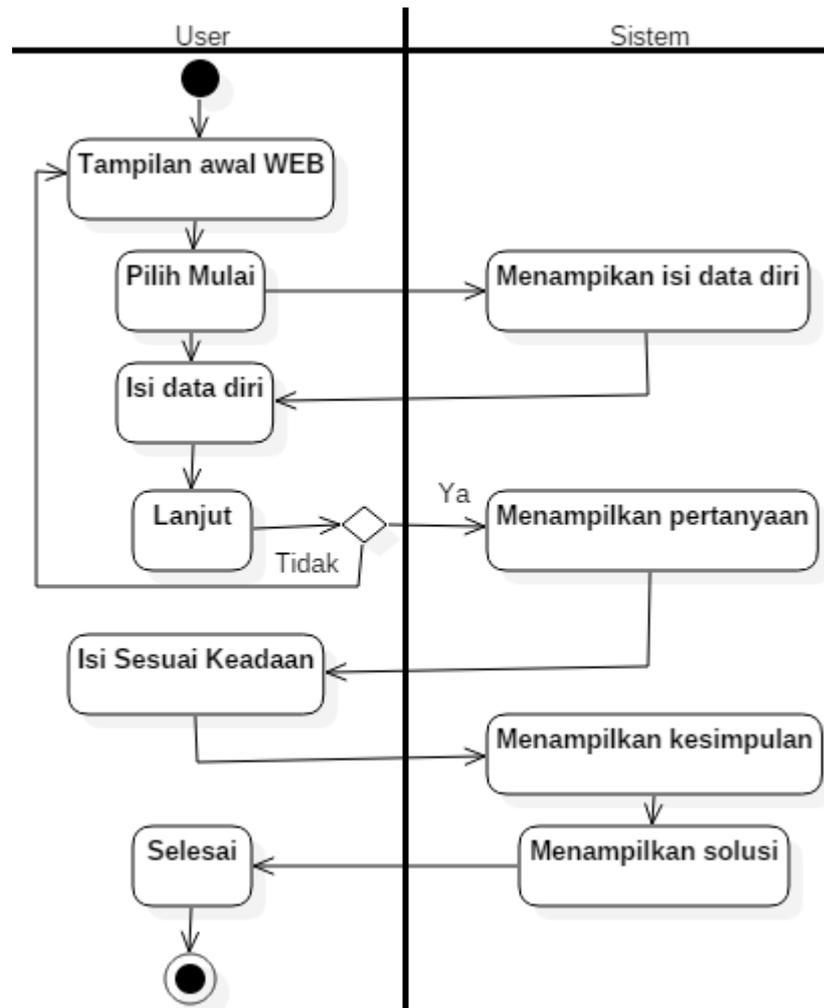
Activity diagram pada gambar 3.8. menggambarkan proses dalam menampilkan pertanyaan hingga proses selesai.



GAMBAR: 3.8. Activity Diagram Menampilkan Pertanyaan Diagnosa

2. *Activity* diagram identifikasi menampilkan solusi.

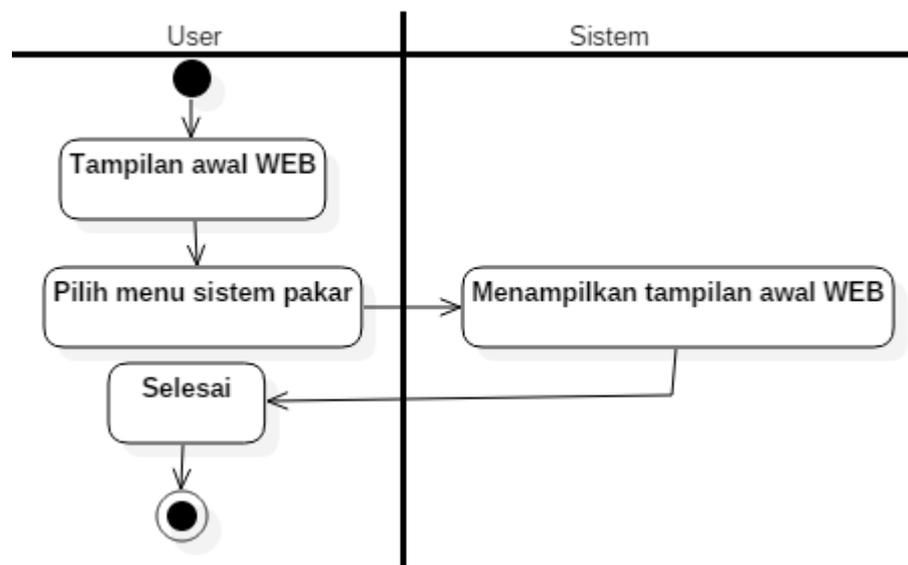
Activity diagram pada gambar 3.9. menggambarkan proses dalam menampilkan pertanyaan hingga proses solusi.



GAMBAR: 3.9. *Activity* Diagram Menampilkan Solusi

3. Activity diagram Menampilkan halaman awal WEB

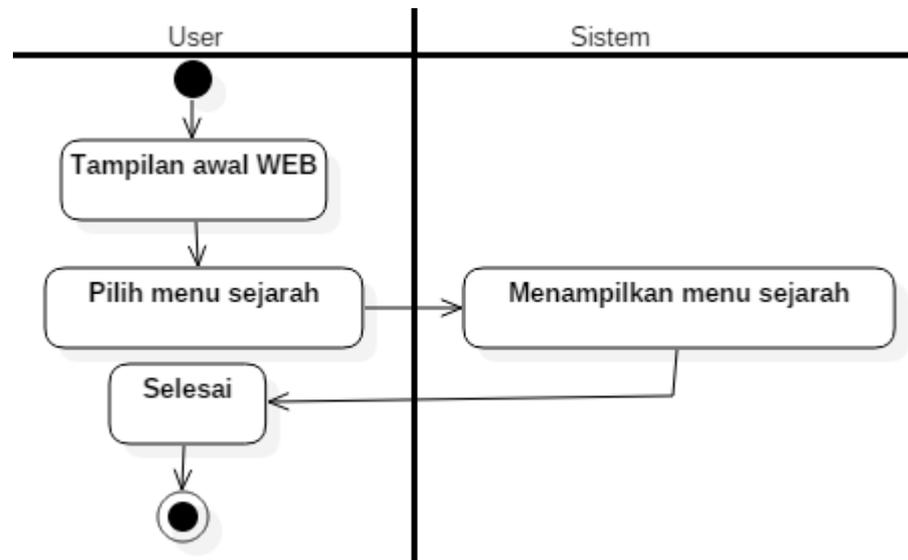
Activity Diagram pada gambar 3.10. menggambarkan proses dari menampilkan rumah sakit rujukan COVID-19.



GAMBAR: 3.10. Activity Diagram Menampilkan Tampilan Awal WEB

4. Activity diagram menampilkan sejarah penyakit COVID-19

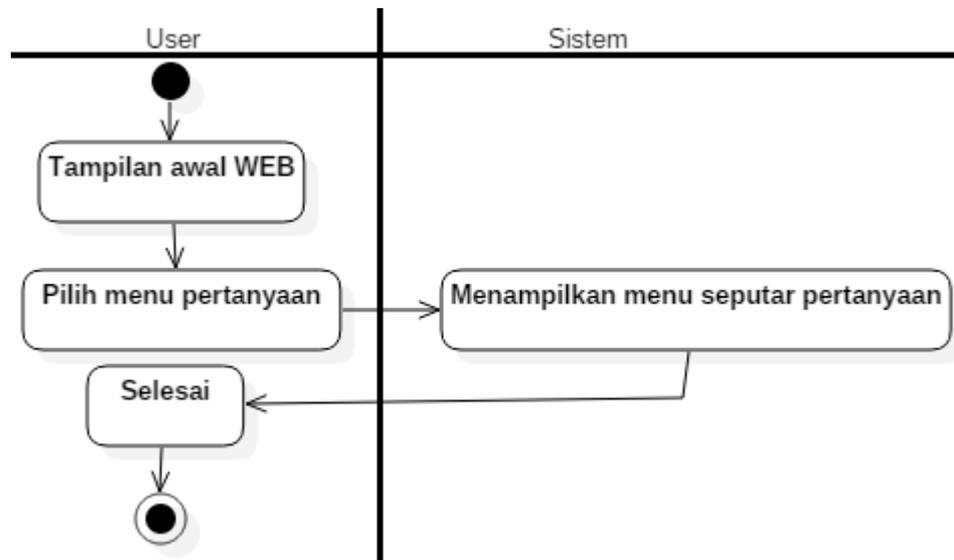
Activity diagram pada gambar 3.11. menggambarkan proses dari menampilkan info sejarah penyakit COVID-19.



GAMBAR: 3.11. Activity Diagram Info Sejarah Penyakit **COVID-19**

4. Activity diagram menampilkan seputar pertanyaan mengenai **COVID-19**

Activity diagram pada gambar 3.12. menggambarkan proses dari menampilkan seputar pertanyaan mengenai **COVID-19**.

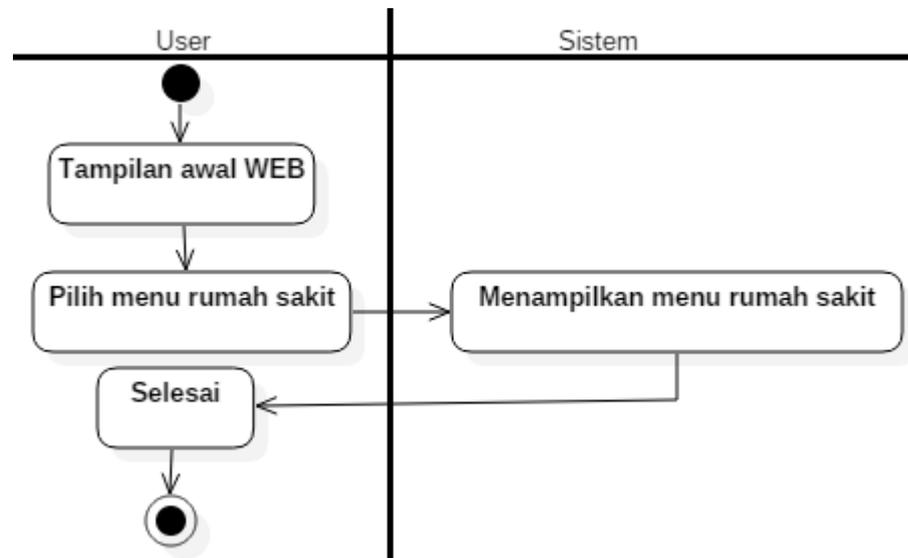


GAMBAR: 3.12. Activity Diagram Menampilkan Seputar Pertanyaan COVID-

19

5. *Activity* diagram menampilkan rumah sakit rujukan

Activity diagram pada gambar 3.13 menggambarkan proses dari menampilkan rumah sakit rujukan **COVID-19**.

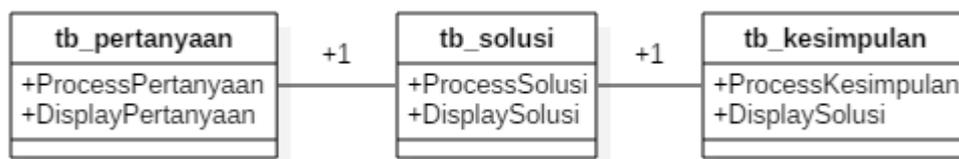


GAMBAR: 3.13. Activity Diagram Menampilkan Rumah Sakit Rujukan

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 3.14.



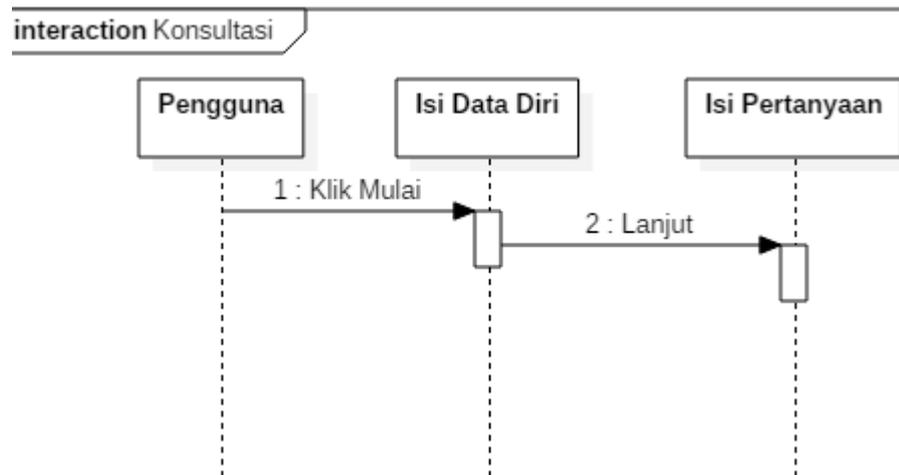
GAMBAR: 3.14. Class Diagram Aplikasi Sistem Pakar **COVID-19**

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 3.15. sampai gambar 3.19. :

1. *Sequence Diagram* Pertanyaan Diagnosa.

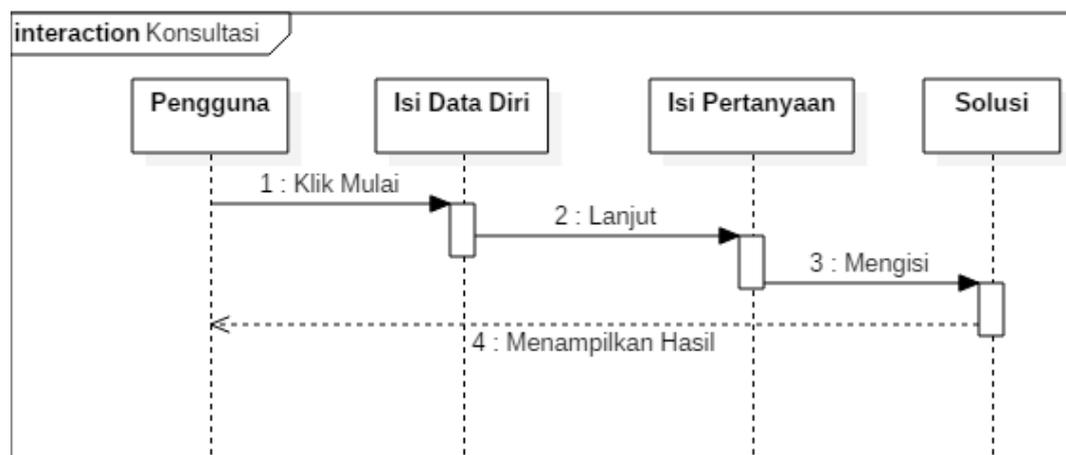
Sequence diagram pada gambar 3.15. menggambarkan urutan pesan yang ada dalam *Use case* diagram identifikasi pertanyaan diagnosa.



GAMBAR: 3.15. *Sequence* Diagram Pertanyaan Diagnosa

2. *Sequence* Diagram Menampilkan Solusi.

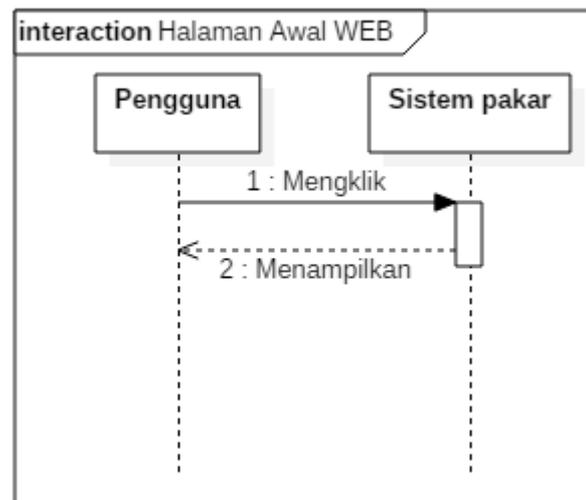
Sequence diagram pada gambar 3.16. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan solusi.



GAMBAR: 3.16. *Sequence* Diagram Solusi

3. *Sequence* Diagram Menampilkan Menu Tampilan Awal.

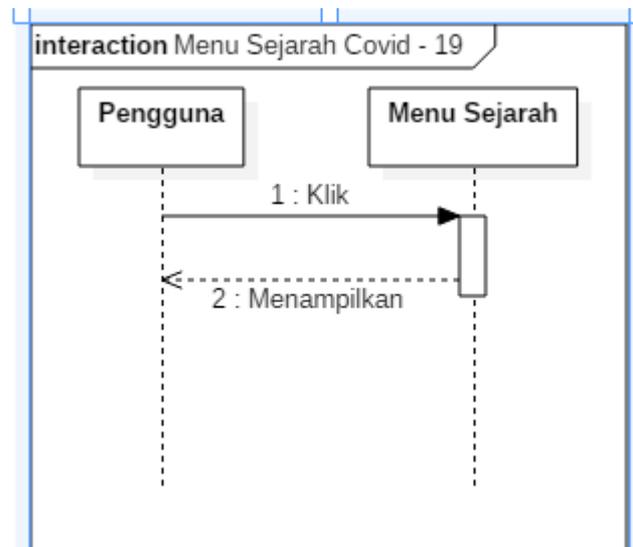
Sequence diagram pada gambar 3.17. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan menu tampilan awal **WEB**.



GAMBAR: 3.17. *Sequence* Diagram Menampilkan Menu Tampilan Awal **WEB**

4. *Sequence* Diagram Menampilkan Menu Sejarah.

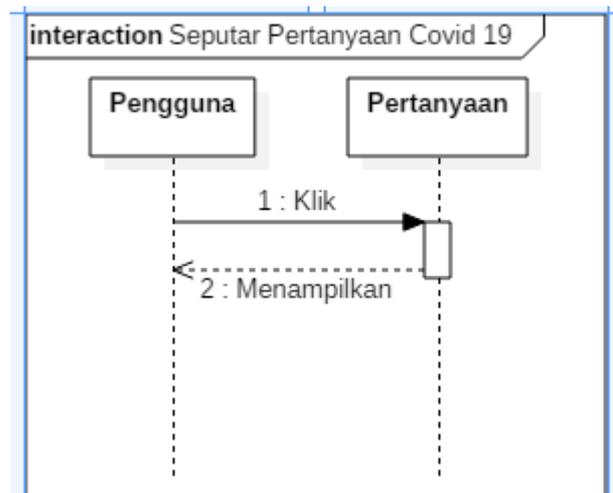
Sequence diagram pada gambar 3.18. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan menu sejarah.



GAMBAR: 3.18. *Sequence* Diagram Menampilkan Menu Sejarah COVID-19

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

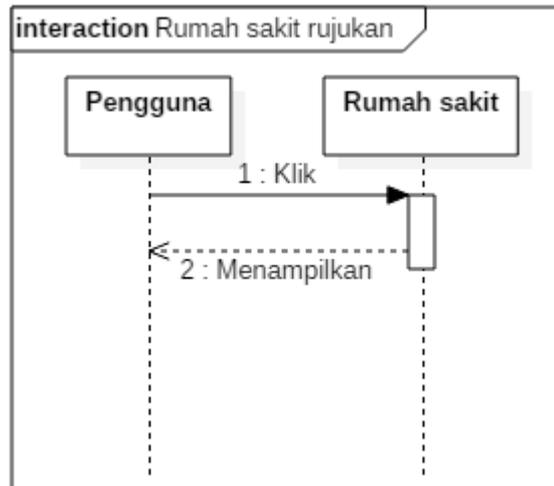
Sequence Diagram pada gambar 3.19. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan seputar pertanyaan.



GAMBAR: 3.19. *Sequence* Diagram Menampilkan Menu Seputar Pertanyaan
COVID-19

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

Sequence diagram pada gambar 3.20. menggambarkan urutan pesan yang ada dalam *Use case* diagram menampilkan rumah sakit rujukan.



GAMBAR: 3.20. *Sequence* Diagram Menampilkan Rumah Sakit Rujukan

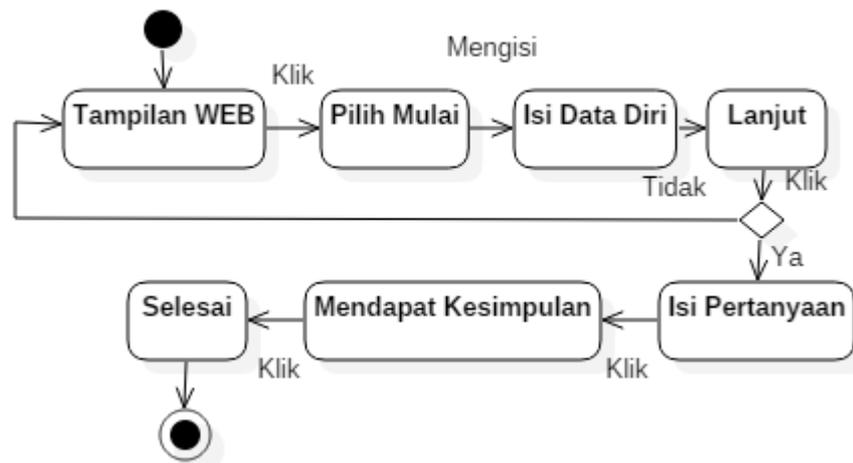
COVID-19

3.3.2.5. *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 3.20. sampai gambar 3.24. :

1. *Statechart* Diagram Pertanyaan Diagnosa

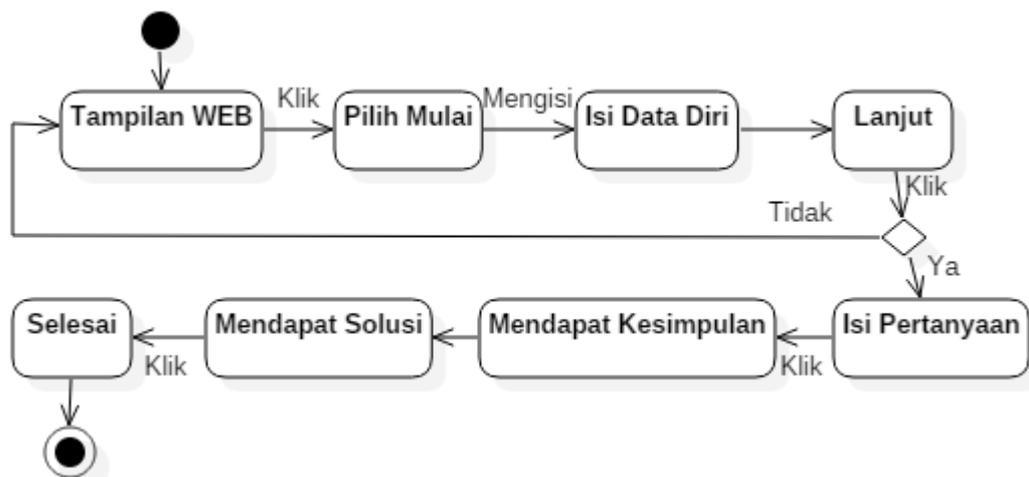
Statechart diagram pertanyaan diagnosa menggambarkan dari tampilan awal ke proses pertanyaan diagnosa untuk tercapainya kesimpulan terkena atau tidak terkena terhadap penyakit **COVID-19**.



GAMBAR: 3.21. Statechart Diagram Pertanyaan Diagnosa

2. Statechart Diagram Solusi

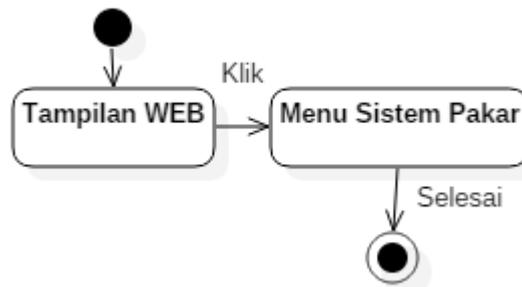
Statechart diagram menu solusi menggambarkan saat proses dilakukannya pilihan untuk mendapatkan solusi.



GAMBAR: 3.22. Statechart Diagram Solusi

3. *Statechart* Diagram Menu Awal **WEB**

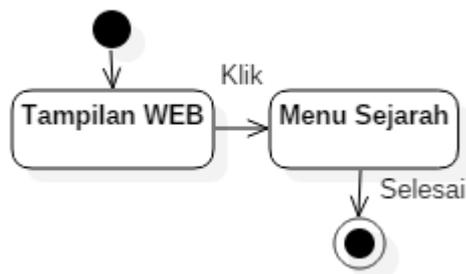
Statechart diagram menu awal **WEB** menggambarkan saat proses dilakukannya pilihan untuk menampilkan tentang tampilan awal **WEB**.



GAMBAR: 3.23. *Statechart* Diagram Menu Awal **WEB**

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

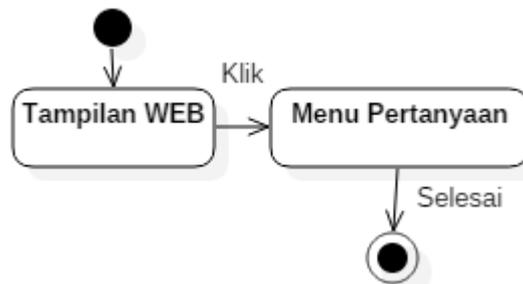
Statechart diagram menu sejarah menggambarkan saat proses dilakukannya pilihan untuk menampilkan tentang sejarah **COVID-19**.



GAMBAR: 3.24. *Statechart* Diagram Menu Sejarah **COVID-19**

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

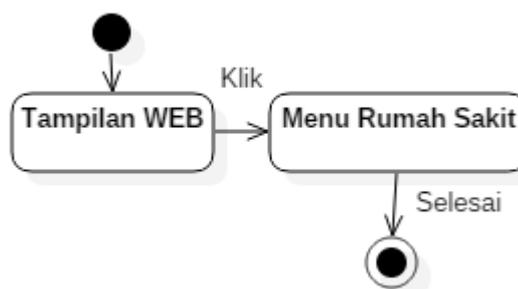
Statechart diagram menampilkan menu pertanyaan seputar **COVID-19**.



GAMBAR: 3.25. *Statechart* Diagram Menampilkan Pertanyaan Seputar
COVID-19

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

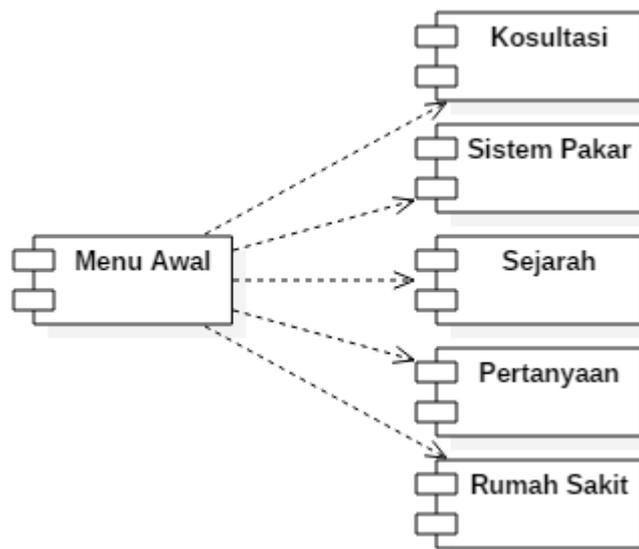
Statechart diagram menampilkan menu rumah sakit rujukan **COVID-19**.



GAMBAR: 3.26. *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 3.27.

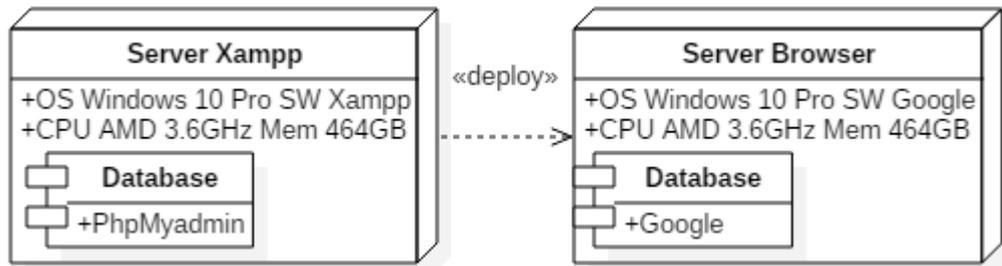


GAMBAR: 3.27. Component Diagram Aplikasi Sistem Pakar **COVID-19**

3.3.2.7. Deployment Diagram

Deployment diagram pada aplikasi sistem pakar **COVID-19** ini menjelaskan hubungan antar *user*, aplikasi. *User* menggunakan *device* berupa

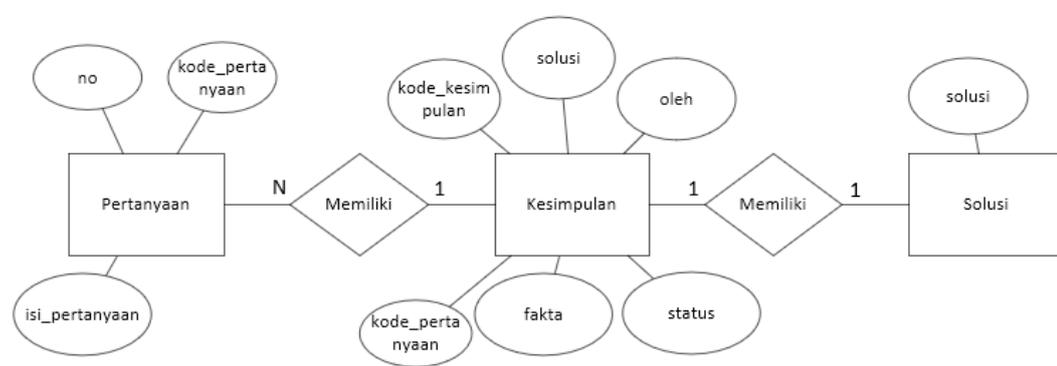
perangkat laptop dalam menjalankan aplikasi ini. Model dari *Deployment* diagram aplikasi sistem pakar **COVID-19** yang diusulkan pada gambar 3.28.



GAMBAR: 3.28. *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 3.29.



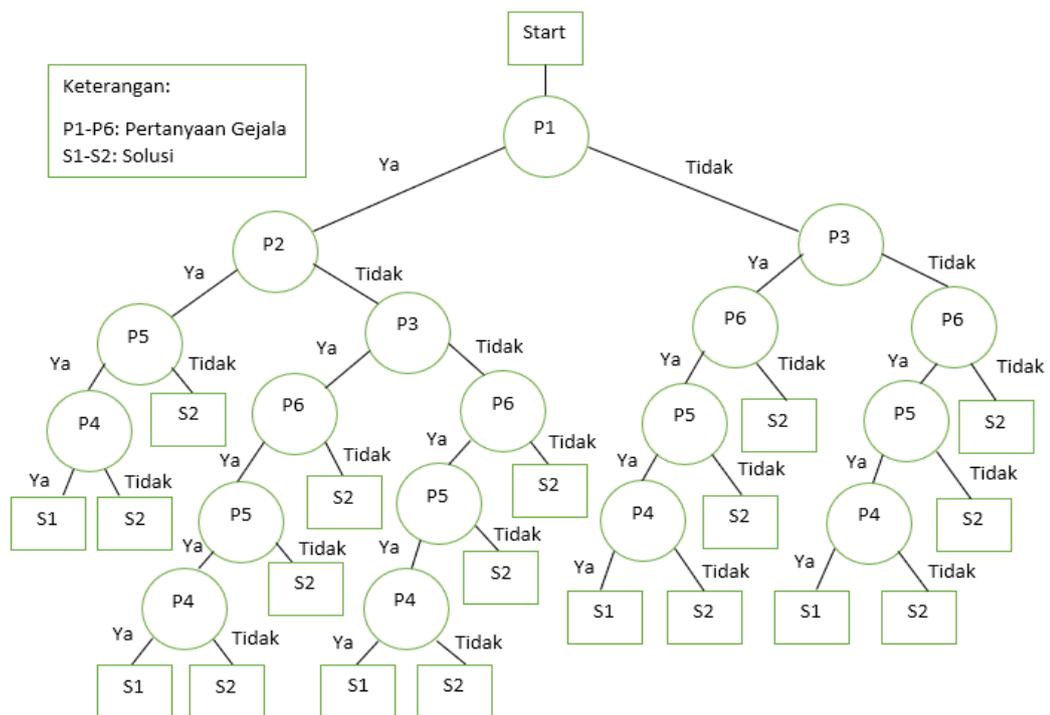
GAMBAR: 3.29. *Entity Relationship* Diagram Aplikasi Sistem Pakar **COVID-19**

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 ditemukannya 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: 3.30. 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

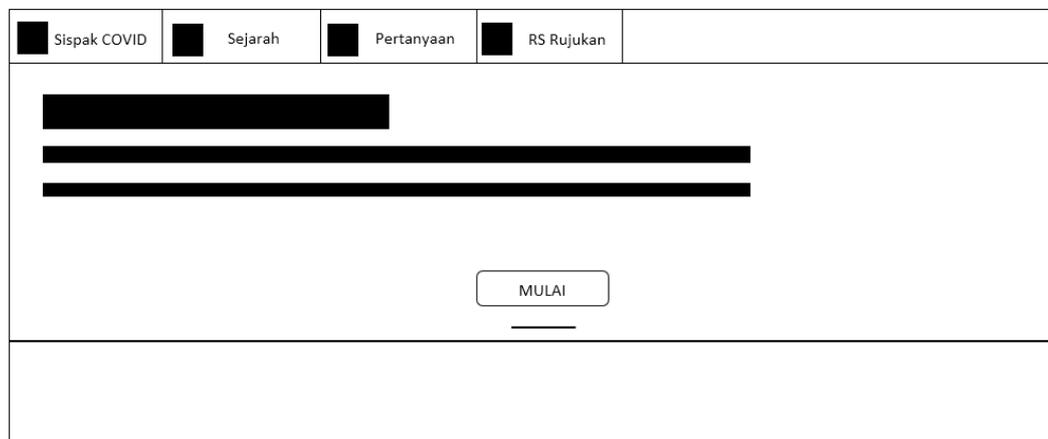
*Rule 6**IF* Demam*AND* Tidak Kehilangan Rasa/Bau*AND* Sesak Nafas*AND* Pilek*AND* Tidak Kelelahan*THEN* Anda Tidak Terindikasi*Rule 7**IF* Demam*AND* Tidak Kehilangan Rasa/Bau*AND* Sesak Nafas*AND* Tidak Pilek*THEN* Anda Tidak Terindikasi*Rule 8**IF* Demam*AND* Tidak Kehilangan Rasa/Bau*AND* Tidak Batuk*AND* 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 12**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Kelelahan*AND* Sesak Nafas*THEN* Anda Terindikasi*Rule 16**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Kelelahan*AND* Sesak Nafas*THEN* Anda Terindikasi*Rule 13**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Kelelahan*AND* Tidak Sesak Nafas*THEN* Anda Tidak Terindikasi*Rule 17**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Kelelahan*AND* Tidak Sesak Nafas*THEN* Anda Tidak Terindikasi*Rule 14**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Tidak Kelelahan*THEN* Anda Tidak Terindikasi*Rule 18**IF* Tidak Demam*AND* Batuk*AND* Pilek*AND* Tidak Kelelahan*THEN* Anda Tidak Terindikasi*Rule 15**IF* Tidak Demam*AND* Batuk*AND* Tidak Pilek*THEN* Anda Tidak Terindikasi*Rule 19**IF* Tidak Demam*AND* Batuk*AND* Tidak Pilek*THEN* Anda Tidak Terindikasi

3.3.6. Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan dari antarmuka yang digunakan sebagai perantara *user* dengan perangkat yang dikembangkan. Perangkat antarmuka dari aplikasi sistem pakar **COVID-19** ini dapat dilihat pada gambar berikut ini :

1. Menu Utama (Main Menu).



GAMBAR: 3.31. Perancangan Tampilan Main Menu Aplikasi

2. Isi Data

The image shows a web application interface with a navigation bar at the top containing four menu items: "Sispak COVID", "Sejarah", "Pertanyaan", and "RS Rujukan". Each menu item has a small black square icon to its left. Below the navigation bar is a large white area. A modal dialog box is open in the center, featuring a title bar with a close button (an 'x' icon). The dialog contains three input fields: "Isi Nama", "Umur", and "Lanjut >".

GAMBAR: 3.32. Perancangan Tampilan Isi Data

3. Pertanyaan Diagnosa

The image shows a web application interface with a navigation bar at the top containing four menu items: "Sispak COVID", "Sejarah", "Pertanyaan", and "RS Rujukan". Each menu item has a small black square icon to its left. Below the navigation bar is a large white area. A form is displayed in the center, containing the text "Nama, Umur" and "Pertanyaan" stacked vertically. Below the text are two buttons: "YA" and "TIDAK".

GAMBAR: 3.33. Perancangan Tampilan Pertanyaan Diagnosa

4. Diagnosa

<input type="checkbox"/> Sispak COVID	<input type="checkbox"/> Sejarah	<input type="checkbox"/> Pertanyaan	<input type="checkbox"/> RS Rujukan	
Hasil Diagnosa				
Biodata				
Terdiagnosa				
<input type="button" value="Akhir"/>				

GAMBAR: 3.34. Perancangan Tampilan Hasil Diagnosa

5. Menu Tampilan Sejarah

<input type="checkbox"/> Sispak COVID	<input type="checkbox"/> Sejarah	<input type="checkbox"/> Pertanyaan	<input type="checkbox"/> RS Rujukan	
<div style="display: flex; justify-content: space-between;"> <div style="background-color: black; width: 80%; height: 15px;"></div> Judul </div> <div style="margin-top: 10px;"> ISI <div style="background-color: black; width: 80%; height: 15px; margin: 5px 0;"></div> <div style="background-color: black; width: 80%; height: 15px; margin: 5px 0;"></div> <div style="background-color: black; width: 80%; height: 15px; margin: 5px 0;"></div> <div style="background-color: black; width: 80%; height: 15px; margin: 5px 0;"></div> <div style="background-color: black; width: 80%; height: 15px; margin: 5px 0;"></div> </div>				

GAMBAR: 3.35. Perancangan Tampilan Sejarah Tentang COVID-19

6. Menu Tampilan Seputar Pertanyaan

<input type="checkbox"/> Sispak COVID	<input type="checkbox"/> Sejarah	<input checked="" type="checkbox"/> Pertanyaan	<input type="checkbox"/> RS Rujukan	
[Redacted]		Judul		
[Redacted]		Penjelasan		
Pertanyaan (Klik, Jawaban)				
Pertanyaan (Klik, Jawaban)				
Pertanyaan (Klik, Jawaban)				
Pertanyaan (Klik, Jawaban)				
Pertanyaan (Klik, Jawaban)				
Pertanyaan (Klik, Jawaban)				

GAMBAR: 3.36. Perancangan Tampilan Seputar Pertanyaan COVID-19

7. Menu Tampilan Rumah Sakit Rujukan

<input type="checkbox"/> Sispak COVID	<input type="checkbox"/> Sejarah	<input checked="" type="checkbox"/> Pertanyaan	<input type="checkbox"/> RS Rujukan	
[Redacted]		Judul		
[Redacted]		Penjelasan		
Pulau (Klik, Daftar per Kota)				
Pulau (Klik, Daftar per Kota)				
Pulau (Klik, Daftar per Kota)				
Pulau (Klik, Daftar per Kota)				
Pulau (Klik, Daftar per Kota)				

GAMBAR: 3.37. Perancangan Tampilan Rumah Sakit Rujukan COVID-19

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1. Construction (Code & Test)

Dalam tahap ini penelitian berfokus pada pengkodean menggunakan Bahasa **PHP**, setelah itu dilakukan pengujian hasil menggunakan metode *black box testing*. *Black box testing* menguji spesifikasi suatu fungsi atau modul, apakah berjalan sesuai dengan semestinya dan sesuai yang diharapkan atau tidak.

4.1.1. Implementasi

Setelah aplikasi dianalisis dan dirancang secara rinci, maka selanjutnya menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Berikut ini adalah tahap implementasi aplikasi sistem pakar penyakit **COVID-19**.

4.1.1.1. Implementasi Hardware & Software

Dalam tahap ini peneliti memerlukan spesifikasi *hardware* dan *software* untuk mengoperasikan *website*. Untuk lebih jelasnya akan dijelaskan dibawah ini :

A. Hardware

Hardware yang digunakan sebagai berikut :

1. *Processor AMD Quad Core FX-7600P Up To 3.60 Ghz*
2. *Memory 8 GB DDR 4 2133 MHz*
3. *Harddisk 1 TB SATA 5400 RPM*

B. Software

1. *Sistem Operasi Windows 10 Pro*
2. *Notepad++ V. 7.8.7*
3. *Xampp V. 7.3.9-0*
4. *Mozilla FireFox/Google Chrome/Microsoft Edge terbaru*

4.1.1.2. Implementasi Antarmuka

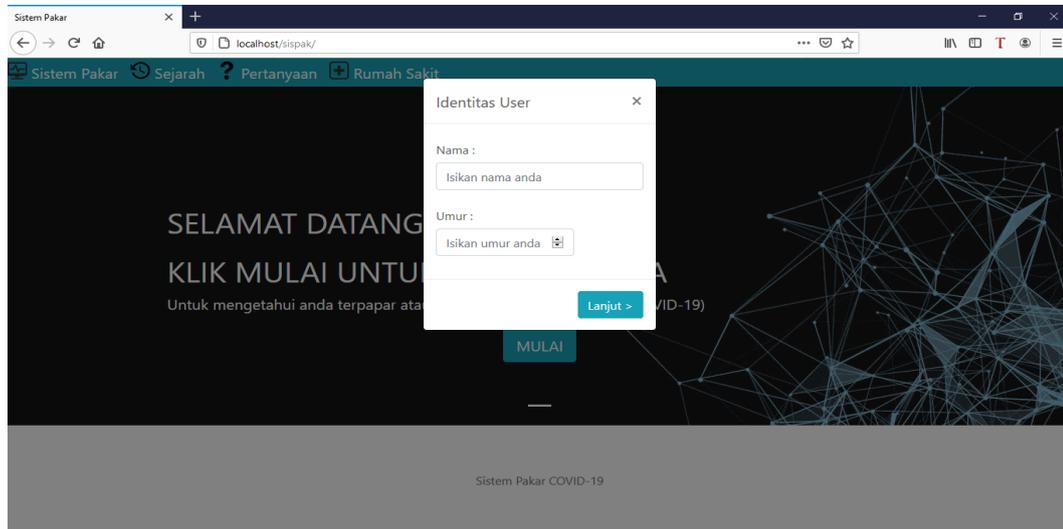
Pada gambar ini menampilkan tampilan awal **WEB**.



Sistem Pakar COVID-19

GAMBAR: 4.1. Tampilan Awal **WEB**

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



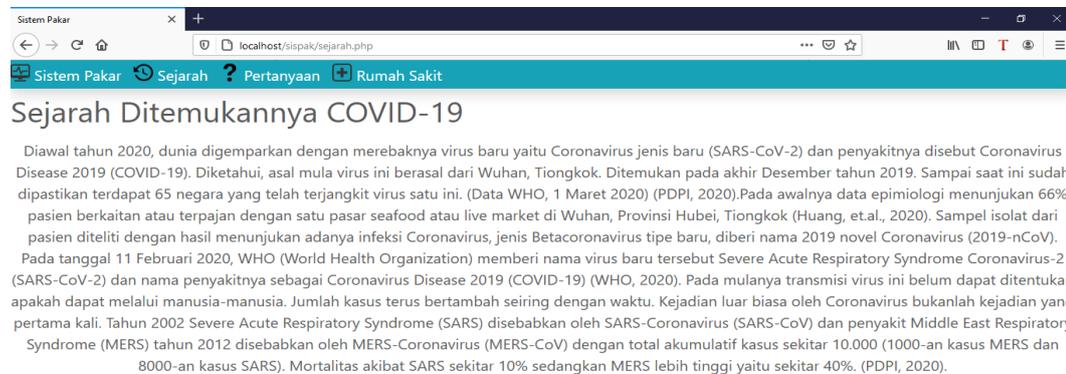
GAMBAR: 4.2. Tampilan Untuk Memulai Diagnosa

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



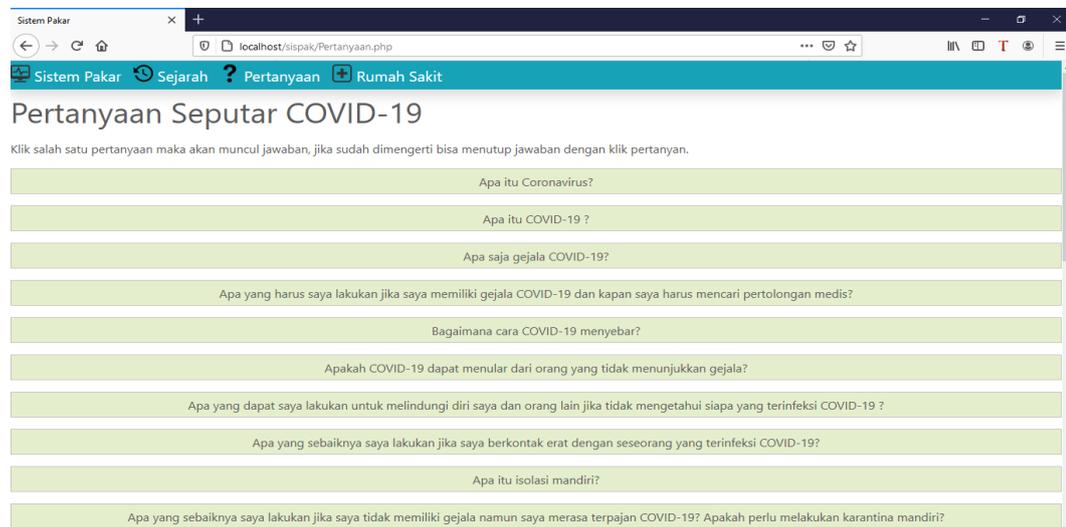
GAMBAR: 4.3. Tampilan Pertanyaan Diagnosa

Pada gambar berikut ini tampilan jika mengklik menu sejarah.



GAMBAR: 4.4. Tampilan Menu Sejarah

Pada gambar berikut ini tampilan jika mengklik menu seputar Pertanyaan.



GAMBAR: 4.5. Tampilan Menu Pertanyaan

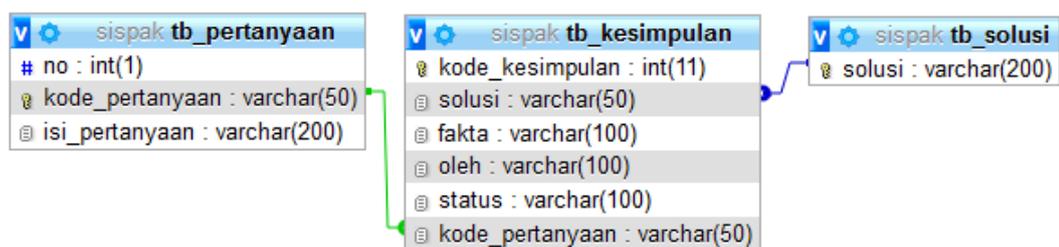
Pada gambar berikut ini tampilan jika mengklik menu pilihan Rumah Sakit rujukan.



GAMBAR: 4.6. Tampilan Menu Rumah Sakit

4.1.1.3. Implementasi *Database*

Berikut adalah gambar dari relasi tabel dan struktur tabel.



GAMBAR: 4.7. Relasi Tabel Aplikasi Sistem Pakar COVID-19

sispak tb_pertanyaan	
#	no : int(1)
	kode_pertanyaan : varchar(50)
	isi_pertanyaan : varchar(200)

GAMBAR: 4.8. Tabel Pertanyaan Aplikasi Sistem Pakar COVID-19

sispak tb_kesimpulan	
	kode_kesimpulan : int(11)
	solusi : varchar(50)
	fakta : varchar(100)
	oleh : varchar(100)
	status : varchar(100)
	kode_pertanyaan : varchar(50)

GAMBAR: 4.9. Tabel Kesimpulan Aplikasi Sistem Pakar COVID-19

sispak tb_solusi	
	solusi : varchar(200)

GAMBAR: 4.10. Tabel Solusi Aplikasi Sistem Pakar COVID-19

4.1.1.4. *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

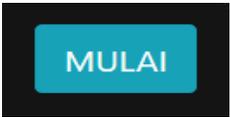
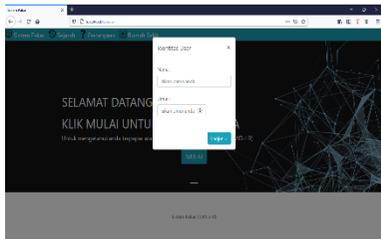
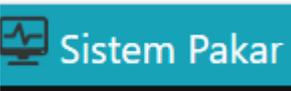
Black box testing juga dikenal sebagai *behavioral testing*, metode ini merupakan pengujian terhadap fungsionalitas atau kegunaan sebuah aplikasi.

TABEL: 4.1. Rencana Pengujian

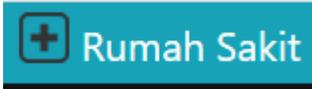
No	Kelas Uji	Butir Pengujian	Jenis Pengujian
1	Menu/Button	Memilih <i>button</i> Mulai	<i>Black Box</i>
		Memilih <i>button</i> Sistem Pakar	<i>Black Box</i>
		Memilih <i>button</i> Sejarah	<i>Black Box</i>
		Memilih <i>button</i> Pertanyaan	<i>Black Box</i>
		Memilih <i>button</i> Rumah Sakit	<i>Black 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:

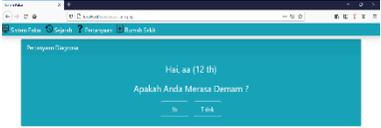
TABEL: 4.2. Tabel Hasil Pengujian *Black box*

No	Item Uji	Skenario Uji	<i>Input</i>	Proses	Yang Diharapkan	<i>Output</i>	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 WEB	Menampilkan awal WEB		Berhasil

Lanjutan dari TABEL: 4.2. 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

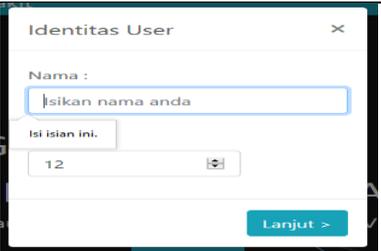
Lanjutan dari TABEL: 4.2. Tabel Hasil Pengujian *Black box*

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 WEB		Berhasil
8		Memilih <i>button</i> Ya atau Tidak		Pilihan jawaban sesuai dengan keadaan	Menampilkan pertanyaan selanjutnya hingga selesai		Berhasil

Lanjutan dari TABEL: 4.2. 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 WEB		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: 4.2. 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

BAB V

PENUTUP

5.1. 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.

5.2. Saran

Adapun saran yang disampaikan sebagai pengembangan aplikasi sistem pakar penyakit **COVID-19** adalah sebagai berikut :

1. Aplikasi bisa berjalan di semua perangkat *mobile phone* berbasis *Android* atau *iOS*.
2. Aplikasi bisa digunakan secara *online* ataupun *offline* di perangkat *mobile phone*.

DAFTAR PUSTAKA

- Akil, Ibnu. (2017). Analisa Efektifitas Metode Forward Chaining dan Backward Chaining Pada Sistem Pakar. Jurnal Pilar Nusa Mandiri [Online].13(1), Hal. 35-41. Tersedia di: <http://ejournal.nusamandiri.ac.id/index.php/pilar/article/view/12>. (16 Mei 2020).
- Anhar. (2010). Panduan Menguasai PHP & MySQL Secara Otodidak, Mediakita. Jakarta (12 Januari 2021).
- Arhami, M., (2005). Konsep Dasar Sistem Pakar, Andi. Yogyakarta (12 Januari 2021).
- Arifin, Jaenal. (2016). Sistem Pakar Diagnosa Penyakit Gigi dan Mulut Manusia Menggunakan Knowledge Base System dan Certainty Factor. [Online]. 10(2), 50-64. Tersedia di: <https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/86>. (25 Februari 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 (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).
- Irwansyah, E., Moniaga, J.V., (2004). *Pengantar Teknologi Informasi*, Deepublish. Yogyakarta (02 Februari 2021).
- Muslihudin, M., Oktafianto. (2016): *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, C.V Andi Offset. Yogyakarta (30 Mei 2020).

Pikobar jabarprov. (2021). Sebaran Kasus Covid-19 di Jawa Barat <https://pikobar.jabarprov.go.id/distribution-case>. (24 Februari 2021).

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).

RDV., Ayu, D., Ernawati. (2012). Tinjauan Penulisan Diagnosis Utama dan Ketepatan Kode ICD-10 Pada Pasien Umum di RSUD Kota Semarang Triwulan I Tahun 2012. [Online]. Tersedia di: https://eprints.dinus.ac.id/7832/1/jurnal_12674.pdf (12 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 (12 Februari 2021).

Satuan Tugas Penanganan COVID-19. (2021). Perkembangan Penanganan Covid-19 di Indonesia Sejalan Dengan Tingkat Dunia <https://covid19.go.id/p/berita/perkembangan-penanganan-covid-19-di-indonesia-sejalan-dengan-tingkat-dunia>. (24 Februari 2021).

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

Suprpto, Untung. (2021). Pemodelan Perangkat Lunak SMK/MAK Kelas XI, Gramedia Widiasarana indonesia. Jakarta (17 Februari 2021).

Velavan, T. P., & Meyer, C. G. (2020). The COVID-19 epidemic. Tropical medicine & international health, 25(3), 278
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7169770/>. (6 Januari 2021)

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. (27 Mei 2020).

WHO. (2020). Questions and answer for public <https://www.who.int/indonesia/news/novel-coronavirus/qa-for-public>. (9 Mei 2020).

Yuhefizar, Mooduto, HA. Hidayat, Rahmat. (2009). Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla Edisi Revisi, PT. Elex Media Komputindo. Jakarta (16 Mei 2020).

Yunita. (2014). Penerapan Metode Forward Chaining Untuk Deteksi Kerusakan Pada Laptop. [Online]. XI(1), 1-10. Tersedia di :
<http://ejournal.nusamandiri.ac.id/index.php/techno/article/view/165> (5 Mei 2020).

LAMPIRAN-LAMPIRAN

```
<?php

//pertanyaan keputusan

function answer($kode){

    if($kode=='P1'){

        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P2'>Ya</a>";

        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P3'>Tidak</a>";

    }

    if($kode=='P2'){

        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P5'>Ya</a>";

        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P3'>Tidak</a>";

    }

    if($kode=='P3'){
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P6'>Ya</a>";
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P6'>Tidak</a>";
```

```
    }
```

```
    if($kode=='P4'){
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='solusi.php?kode=s1'>Ya</a>";
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='solusi.php?kode=s2'>Tidak</a>";
```

```
    }
```

```
    if($kode=='P5'){
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P4'>Ya</a>";
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='solusi.php?kode=s2'>Tidak</a>";
```

```
    }
```

```
    if($kode=='P6'){
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='question.php?kode=P5'>Ya</a>";
```

```
        echo "<a class='btn col-sm-1 mrg btn-lg btn-outline-light'
href='solusi.php?kode=s2'>Tidak</a>";

    }

}

//kesimpulan

function kesimpulan($diagnosa){

    include 'koneksi.php';

    $sql = "SELECT * from tb_kesimpulan WHERE kode_kesimpulan='$diagnosa'
AND status='setuju'";

    $data = mysqli_query($connect,$sql);

    while ($row = mysqli_fetch_assoc($data)) {

        echo '<p> Keterangan : <br>'. $row['fakta'].'</p>';

    }

}

//solusi

function solusi($kode){

    if ($kode=='s1') {

        $diagnosa = "1";

        kesimpulan($diagnosa);

    }

}
```

```
}  
  
if ($kode=='s2') {  
  
    $diagnosa = "2";  
  
    kesimpulan($diagnosa);  
  
}  
  
}  
  
?>
```

```
<!DOCTYPE html>

<html lang="id">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

  <meta name="description" content="website sistem pakar">

  <meta name="author" content="mr k">

  <link rel="icon" href="image/icon.png">

  <title>Sistem Pakar</title>

  <!-- Bootstrap core CSS -->

  <link rel="stylesheet"
href="https://getbootstrap.com/docs/4.1/dist/css/bootstrap.min.css"
crossorigin="anonymous">

  <!-- Custom styles for this template -->

  <link href="css/style.css" rel="stylesheet">

</head>

<body>

  <?php include ('navbar.php'); ?>
```

```
<main class="batas-atas">

<div class="card text-white bg-info mb-3">

<h5 class="card-header">Diagnosa</h5>

<div class="card-body text-left ukuran-20">

<form method="post" action="solusi.php" enctype="multipart/form-data"
role="form">

<?php

include ('koneksi.php');

//$kode='m01';

session_start();

echo "<p>Nama : ".$_SESSION['nama']."</p>";

echo "<p>Umur : ".$_SESSION['umur']."</p>";

        if(isset($_GET['kode'])){

            $kode=$_GET['kode'];

        }

?>

<?php
```

```

        include "fungsi.php";

        solusi($kode);

        ?>

<?php

$sql = "SELECT * from tb_solusi WHERE kode_solusi='$kode'";

$data = mysqli_query($connect,$sql);

$row = mysqli_fetch_assoc($data);

if ($row['kode_solusi']=="x-1" || $row['isi_solusi']=="x-2" ||
$row['isi_solusi']=="x-3" || $row['isi_solusi']=="x-4" || $row['isi_solusi']=="x-5")
{

        echo "<center><p><strong style='color:red'>SISTEM TIDAK
MENEMUKAN JAWABAN !</strong></p></center><hr>";

        ?>

<?php

}

else{

        echo "<p>Diagnosa : <strong
style='color:red'>".$row['isi_solusi']."</strong></p>";

}

```

```

        ?>

</form>

<br>

<div class="text-center">

    <a style="margin-bottom: 10px;" href="hapus-session.php" class="btn
btn-outline-light col-sm-2">Akhir</a>

</div>

</div>

</div>

</main>

<!-- Bootstrap core JavaScript
===== -->

<!-- Placed at the end of the document so the pages load faster -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jiz
o" crossorigin="anonymous"></script>

<script
src="https://getbootstrap.com/docs/4.1/dist/js/bootstrap.min.js"></script>

```

```
</body>
```

```
</html>
```