BAB II

TINJAUAN PUSTAKA

2.1 Virus Covid-19

Covid-19 (*Corona Virus Desease 2019*) adalah jenis penyakit baru yang disebabkan oleh virus dari golongan *coronavirus*, yaitu SARS-CoV-2. Kasus pertama penyakit ini terjadi di kota Wuhan, China, pada akhir Desember 2019. Setelah itu, Covid-19 menular antarmanusia dengan sangat cepat dan menyebar ke puluhan negara, hanya dalam beberapa bulan saja.

Menurut data yang dirilis Gugus Tugas Percepatan Penanganan COVID-19 Republik Indonesia, jumlah kasus terkonfirmasi positif hingga 19 April 2021 adalah 1.609.300 orang dengan jumlah kematian 43.567 orang. Dari kedua angka ini dapat disimpulkan bahwa *case fatality rate* atau tingkat kematian yang disebabkan oleh COVID-19 di Indonesia adalah sekitar 2,7%. *Case fatality rate* adalah presentase jumlah kematian dari seluruh jumlah kasus positif COVID-19 yang sudah terkonfirmasi dan dilaporkan.

COVID-19 dapat menginfeksi siapa saja, tetapi efeknya akan lebih berbahaya atau bahkan fatal bila menyerang orang lanjut usia, ibu hamil, perokok, penderita penyakit tertentu, dan orang yang daya tahan tubuhnya lemah, seperti penderita kanker. Karena mudah menular, penyakit ini juga berisiko tinggi menginfeksi para tenaga medis yang merawat pasien COVID-19. Oleh karena itu, tenaga medis dan orang yang melakukan kontak dengan pasien COVID-19 perlu menggunakan alat pelindung diri (APD).

Selain itu, pemerintah bersama perusahaan farmasi dan berbagai institusi kesehatan kini juga tengah mengembangkan dan meneliti vaksin COVID-19. Setelah melalui uji klinis dan dinyatakan efektif dan aman diberikan pada manusia, pembuatan vaksin COVID-19 akan diteruskan agar dapat diberikan kepada masyarakat (Dzahabi Yunas, Triayudi, and Sholihati 2021).

2.2 Sistem Pakar

Sistem pakar merupakan aplikasi berbasis komputer yang digunakan untuk menyelesaikan sebuah masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud adalah seseorang yang memiliki keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sistem pakar adalah sistem perangkat lunak komputer yang menggunakan ilmu, fakta, dan teknik berpikir dalam pengambilan keputusan (Fatkhurrozi and Kurniawan 2020).

2.3 Certainty Factor

Sistem pakar adalah sistem yang berusaha untuk mengadopsi pengetahuan manusia ke dalam sebuah komputer, agar dapat digunakan oleh orang yang tidak memiliki keahlian dan mendapatkan hasil penyelesaian selayaknya seorang pakar. Sistem pakar akan mempertanyakan fakta-fakta yang akan menunjukkan gejala penyakit tertentu dan dapat memberikan penjelasan atas hasil konsultasi yang telah dilakukan. Seorang dokter memungkinkan mengungkapkan informasi berupa pernyataan yang tidak pasti seperti misalnya mungkin, kemungkinan besar dan hampir pasti. Salah satu metode yang dapat digunakan untuk mengatasi ketidak pastian adalah metode *certainty factor* (CF). Pada *certainty factor* setiap *rule* memiliki nilai keyakinannya sendiri, dan tidak hanya premis-premisnya saja yang memiliki nilai keyakinan (Dzahabi Yunas, Triayudi, and Sholihati 2021).

Certainty Factor didefinisikan sebagai berikut :

$$CF(H,E) = MB(H,E) - MD(H,E)$$

CF(H,E): *certainty factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak, sedangkan nilai 1 menunjukkan kepercayaan mutlak.

MB(H,E): ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

MD(H,E): ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

2.4 Android

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (*touchscreen*) yang pertama kal dikembangkan oleh perusahaan *startup* di California bernama Android, Inc. Android dipelopori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Pada tahun 2005, Google membeli Android dan mengambil alih proses pengembangannya hingga saat ini. Google merilis versi beta Android SDK (*System Development Kit*) pada November 2007.

Sejak pertama kali diperkenalkan pada tahun 2007, Android telah mengalami perkembangan yang begitu pesat, ditandai dengan banyaknya versi sistem operasi Android ini. Tabel 2.1 menunjukkan beberapa versi yang ada beserta tanggal rilisnya (Soepomo 2014).

Tabel 2. 1 Versi-versi Android

VERSI ANDROID	API LEVEL	TANGGAL RILIS	CODENAME
1.0	1	23 September 2008	
1.1	2	9 Februari 2009	
1.5	3	30 April 2009	Cupcake
1.6	4	15 September 2009	Donut
2.0/2.0.1/2.1	5/6/7	26 Oktober 2009	Eclair

2.2-2.2.3	8	20 Mei 2010	Froyo
2.3-2.3.2/2.3.3-2.3-7	9/10	6 Desember 2010	Gingerbread
3.0/3.1/3.2-3.26	11/12/13	22 Februari 2011	Honeycomb
4.0-4.0.2/4.0.3-4.0.4	14/15	19 Oktober 2011	Ice Cream Sandwich
4.1-4.1.2/4.2-4.2.2/4.3-4.3.1	16/17/18	27 Juni 2012	Jelly Bean
4.4-4.4.4/4.4 W	19/20	31 Oktober 2013	Kitkat
5.0-5.0.2/5.1-5.1.1	21/22	12 November 2014	Lolipop
6.0-6.0.1	23	5 Oktober 2015	Marshmallow

2.5 Java

Sejarah Java dimulai pada tahun 1991, sebuah team kecil yang berisikan *engineer* dari Sun Microsystem mempelopori proyek Java ini. Tim ini disebut *Green Team* dan dipimpin oleh James Gosling. Pada awal perkembangan, bahasa yang dikembangan oleh Green Team disebut dengan nama "*Greentalk"*, kemudian diganti namanya menjadi "Oak" ketika pohon oak yang tiba-tiba muncul di halaman kantor Green Team. Beberapa waktu kemudian diganti menjadi "Java" setelah diketahui bahwa Oak sudah menjadi hak cipta merek dari Oak Technologies.

Dan pada tahun 1995, Sun Microsystem kemudian merilis Java untuk pertama kalinya. Dan pada 13 november 2006, Sun Microsystem merilis Java gratis dan *open source* dengan lisensi GNU *General Public License* (GPL). Pada tahun 2010, Sun Microsystem dibeli oleh Oracle dan menjadikan Java dikembangkan dan dipelihara dibawah kendail Oracle (Mallu 2015).

2.6 Android Studio

Pada 16 Mei 2013, Google meluncurkan Android Studio sebagai lingkungan pengembangan terintegrasi (*Integrated Development Development*) baru untuk Android. Android Studio dikembangkan oleh team pengembang dari Rusia yaitu *Jetbrains*. Versi stabil dari Android Studio pertama kali dirilis pada Desember 2014 dan menjadi IDE resmi yang digunakan dalam pengembangan Android (Ilhami 2017).

2.7 MySQL

MySQL adalah sebah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database managementsystem*) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *GeneralPublic License* (GPL) (Ryanda et al. 2015).

MySQL adalah salah satu perangkat lunak *Database Management System* (DBMS) yang sering digunakan saat ini, yang didistribusikan secara gratis di bawah lisensi GPL (*GNU General Public License*). Sehingga setiap orang mudah untuk mendapatkan dan bebas untuk menggunakan MySQL.

2.8 Basis Data

Menurut *Chou*, basis data adalah kumpulan informasi yang bermanfaat yang diorganisasikann ke dalam tata cara yang khusus. Menurut *Gabbri* dan *Schwab*, basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data.Sedangkan menurut *Date*, basis data dapat dianggap sebagai tempat untuk sekumpulan berkas dan terkomputerisasi yang tujuan utamanya adalah melakukan pemeliharaan terhadap informasi dan membuat informasi tersebut tersedia saat dibutuhkan.Secara konsep, basis data

adalah kumpulan data yang membentuk suatu berkas atau *file* yang saling berhubungan dengan atacara yang tertentu untuk membentuk data baru atau informasi (Mulyani and Dale 2018).
Sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data. Basis data yang kompleks dan disertai dengan teknik pendokumentasian dan prosedur manipulasinya akan membentuk sistem manajemen basis data. Beberapa komponen-komponen sistem basis data adalah:

1. Perangkat keras (Hardware), sebagai pendukung operasi pengolahan data.
2. Sistem operasi.
3. Basis data.
4. Sistem pengelola basis data.
5. Aplikasi atau perangkat lunak lain yang bersifat opsional.
6. Pemakai/pengguna.

Pengguna basis data meliputi:

1. *Database Administrator*, yaitu pengguna yang memiliki kewenangan sebagai pusat pengendali seluruh sistem baik basis data maupun programprogram yang mengaksesnya, menentukan pola struktur basis data, memodifikasi, membagi tugas pengolahan dan memberikan otoritas tertentu.
2. *Application Programmer*, yaitu *programmer* aplikasi yang berinteraksi dengan sistem melalui pemanggilan*Data Manipulation Language* (DML) yang dimasukkan ke dalam program yang ditulis.
3. *Sophisticated Users,* yaitu pengguna yang berinteraksi dengan sistem tanpa harus menuliskan sendiri programnya, tetapi diganti dengan melakukan permintaan dalam bentuk bahasa *query*.
4. *Specialized Users*, yaitu pengguna yang menuliskan program aplikasi basis data khusus yang tidak sesuai dengan *framework* pemrosesan data tradisional.

5. *Naive Users*, yaitu kebanyakan pengguna yang berinteraksi dengan siste dengan cara memanggil salah satu program aplikasi yang telah disediakan.

2.9 Use Case Diagram

Use case adalah teknik untuk merekam persyaratan fungsional sebuah *system Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar, *Use case* digunakan untuk mengetahui fungsi atau proses apa saja yang ada di dalam aplikasi dan siapa saja yang berhak menggunakan fungsi-fungsi atau proses-proses itu.

Dari pengertian di atas dapat disimpulkan usecase adalah sebuah diagram yang digunakan untuk mendeskripsikan fungsi atau merancang sistem aplikasi yang akan dibuat (Makiolor, Sinsuw, and B.N. Najoan 2017). Berikut adalah simbol-simbol yang ada pada *use case* diagram dapat di lihat pada table 2.1

Tabel 2. 2 Simbol-simbol Use Case Diagram

(Rosa A.S dan M.Shalahuddin : 2011)

Nama Simbol	Simbol	Deskripsi
Use Case	Nama	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor / *actor*		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan

	Nama *actor*	dibuat itusendiri,jadiwalaupunsimbol dari aktor adalah gambarorang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawali frase nama actor
Asosiasi / *association*		Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Ekstensi / *extend*	<<extend>>	Case tambahan memiliki nama depan yangsama dengan use case yang ditambahkan. Arah panah mengarah pada use case yang ditambahkan
Generalisasi/*generalization*		Hubungan generalisasi dengan spesialisasi (umum – khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.

		Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)
Menggunakan / *include / uses*	<<include>> ----------> <<uses>> ——▷	Fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai include di *use case* : • Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahandijalankan • Include berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan

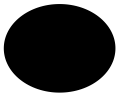
		dijalankan Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan

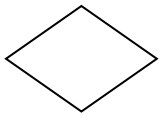
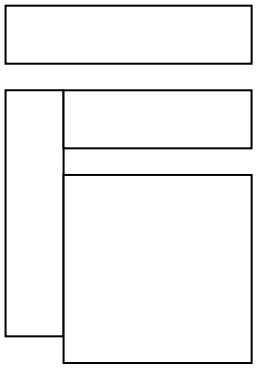
2.10 Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Aprilia, Fauziah, and Komala Sari 2021). Berikut adalah simbol-simbol yang ada pada *activity* diagram :dapat di lihat pada table 2.2

Tabel 2. 3 Simbol-simbol Use Case Diagram

(Rosa A.S dan M.Shalahuddin : 2014)

Nama	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitasmemilikisebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan

		kata kerja.
Percabangan/ *Decision*		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ *Join*		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir		Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane		Memisahkan organisasi bisnis yang bertanggunga jawab terhadap aktivitas yang terjadi.

2.11 Flowchart

Diagram flowchartmerupakan metode untuk menggambarkan tahap-tahap pemecahan masalah dengan mempresentasikan simbol-simbol tertentu yang

mudah dimengerti, mudah digunakan dan standar.Tujuan utama penggunaan flowchartadalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan simbol-simbol yang standar (Fahindra and Al Amin 2021).

Flowchartterbagi menjadi 5 jenis yaitu :

1.FlowchartSistem (SystemFlowchart)Merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan didalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.

2.Flowchart Paperwork / Flowchart Dokumen(Document Flochart)Flowchart Paperworkmenelusuri alur dari data yang ditulis melalui sistem. Flowchart Paperwork juga sering disebut FlowchartDokumen.

3.Flowchart Skematik (Schematic Flowchart)FlowchartSkematik merupakan bagan yang mirip dengan FlowchartSistem yang menggambarkan suatu sistem atau prosedur. Perbedaannya bukan hanya menggunakan simbol-simbol flochartstandar, tetapi juga menggunakan gambar-gamabar komputer, peripheral, from-from atau peralatan lain yang digunakan dalam sistem.

4.FlowchartProgram (Program Flochart)FlowchartProgram merupakan bagan yang menjelaskan keterangan lebih rinci tentang langkah-langkah dari proses program. Flowchart Program juga dihasilkan dari FlowchartSistem.

5.FlowchartProses (Proses Flowchart).Flowchart Proses merupakan bagan alir yang banyak digunakan di teknik industri.Flowchartjuga digunakan untuk memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedutr atau sistem.

Adapun dalam penelitian ini penulis memilih menggunakan flowchartprogram karena flowchart program dapat menjelaskan keterangan lebih rinci tentang langkah-langkah dari proses program.dapat di lihat pada table 2.2

Tabel 2. 4 Simbol-Simbol FlowchartProgram

No	Simbol	Keterangan
1.		Input/Output digunakan untuk mewakili data I/O
2.		Proses, Digunakan untuk mewakili suatu proses
3.		Garis alir, Menunjukkan alur dari suatu proses
4.		Keputusan, digunakan untuk suatu seleksi kondisi di dalam program
5.		Penghubung, menunjukkan penghubung ke halaman yang sama atau halaman lain
6.		Proses terdefinisi, menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain
7.		Persiapan, digunakan untuk memberi nilai awal suatu besaran
8.		Terminal, emnunjukkan awal dan akhir dari suatu Proses

2.12 Penelitian Terkait

Penelitian tentang *Certainty Factor* telah dilakukan oleh peneliti sebelumnya, diantaranya adalah Fajar Januriawan (2018) menunjukkan bahwa aplikasi sistem pakar yang berjalan dengan metode *Certainty Factor* untuk mendiagnosa penyakit pada ayam dimana user dapat memilih nilai sesuai dengan interpretasi *Certainty Factor* yang telah diberikan dan di proses bersama pakar.