

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah suatu jaringan kerja dari prosedur - prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu.

II.1.1 Karakteristik Sistem

Agar sistem itu dikatakan sistem yang baik harus memiliki karakteristik yaitu :

1. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen sistem terdiri dari komponen berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang

harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan sistem (*input*)

Masukan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran. Contoh dalam sistem komputer, program adalah *maintenance input* sedangkan data adalah *signal input*.

6. Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

II.2. Landasan Teori

II.2.1. Sistem Pakar

Sistem pakar merupakan cabang dari Artificial Intelligence (AI) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Istilah sistem pakar berasal dari istilah *knowledge based expert system*. Istilah ini muncul karena memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Berikut beberapa definisi dari Sistem Pakar, antara lain:

1. Sistem pakar merupakan kecerdasan buatan dalam ilmu komputer seiring dengan bidang ilmu pengetahuan. Umumnya sistem pakar berupaya untuk mempelajari pemahaman manusia dengan komputer dalam menuntaskan permasalahan yang kerap dilakukan oleh pakar. Ahli merupakan orang yang memiliki kemampuan di bidang tertentu, yakni ahli yang memiliki *knowledge* ataupun keahlian spesial dalam bidang yang dimilikinya (Sembiring, 2021).
2. Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia. Pengetahuan tersebut dimasukan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia (Nugroho, 2018).
3. Sistem pakar (*Expert System*) merupakan salah satu cabang dari Kecerdasan Buatan atau *Artificial Intelligence* (AI). Sistem Pakar berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan

masalah seperti yang biasa dilakukan oleh para ahli (Pangesti, 2020).

Dapat disimpulkan bahwa Sistem Pakar adalah sebuah sistem komputer yang dapat meniru keahlian seorang pakar dengan pemanfaatan basis pengetahuan (*knowledge base*) yang dapat digunakan oleh orang awam dalam pengambilan keputusan yang biasanya hanya dapat dilakukan oleh seorang pakar.

II.2.2. Karakteristik dan Kelebihan Sistem Pakar

Karakteristik sistem pakar yang membedakan dari sistem lainnya adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Bekerja berdasarkan kaidah/*rule* tertentu.
3. Mudah dimodifikasi.
4. Basis pengetahuan dan mekanisme inferensi terpisah.
5. Keluarnya bersifat anjuran.

Kelebihan sistem pakar adalah sebagai berikut:

1. Sistem Pakar dapat bertindak sebagai konsultan, instruktur, atau rekan.
2. Meningkatkan *availability* atau kepakaran tersedia pada semua perangkat komputer.
3. Mengurangi bahaya.
4. Permanen.
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan.

6. Database yang cerdas, sistem pakar dapat digunakan untuk mengakses database secara cerdas.

II.2.3. Komponen Sistem Pakar

Sebuah program Sistem Pakar terdiri atas beberapa komponen yang mutlak. Komponen-komponen ini berinteraksi dan bekerja sama untuk mencapai tujuan dari sistem pakar. Adapun komponen sistem pakar sebagai berikut:

- a. Basis Pengetahuan (*Knowledge Base*)

Basis Pengetahuan merupakan inti program sistem pakar karena merupakan representasi pengetahuan (*Knowledge Representation*) dari seorang pakar.

- b. Basis Data

Basis Data adalah bagian yang mengandung semua fakta, baik fakta awal pada saat sistem mulai beroperasi maupun fakta yang didapatkan pada saat pengambilan kesimpulan yang dilakukan.

- c. Mesin Inferensi

Mesin Inferensi adalah bagian yang mengandung mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya sistem ini akan mencari jawaban atau kesimpulan yang terbaik.

- d. Antar muka pemakai (*user interface*)

Pada bagian ini terjadi dialog atau penghubung antara program dan pemakai dimana program mengajukan pertanyaan berbentuk ya atau tidak (*yes or no question*) atau berbentuk menu pilihan. Melalui jawaban dari pemakai maka sistem pakar akan mengambil kesimpulan berupa informasi atau anjuran

sesuai dengan sifat dari sistem pakar.

II.2.4. Virus Covid-19

Pada bulan Desember, 2019, wabah *pneumonia* yang disebabkan oleh virus corona terjadi di Wuhan, provinsi Hubei, dan telah menyebar dengan cepat ke seluruh China hingga ke seluruh dunia tak terkecuali Indonesia. Wabah ini diberi nama *Corona Virus Disease (Covid-19)* yang disebabkan oleh *Server Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2)*.

Corona Virus Disease (Covid-19) merupakan keluarga besar virus yang dapat menyebabkan penyakit pada manusia. Beberapa corona virus pada manusia diketahui menyebabkan infeksi pernafasan mulai dari flu biasa hingga penyakit yang lebih parah seperti *Middle East Respiratory Syndrome (MERS)* dan *Server Acute Respiratory Syndrome (SARS)* dan corona virus yang terbaru adalah yang menyebabkan *Covid-19*.

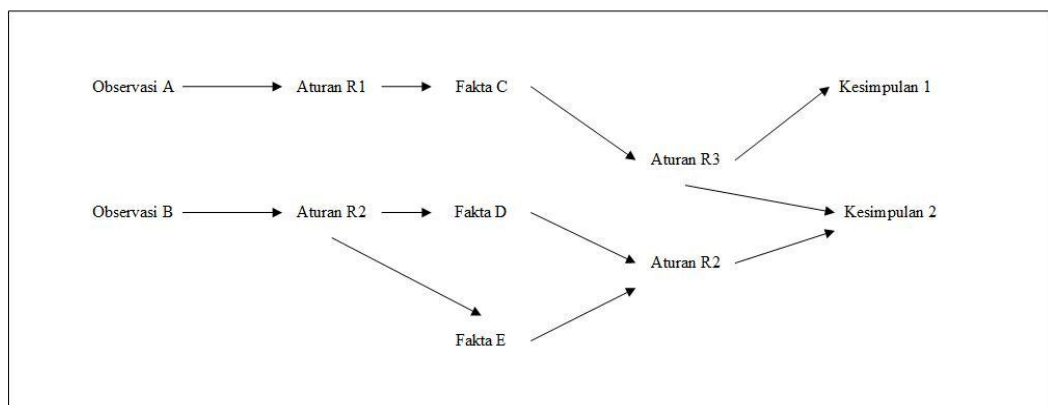
Kebanyakan virus corona menyebar seperti virus lain pada umumnya, seperti melalui percikan air liur pengidap (batuk dan bersin), dan menyentuh tangan atau wajah orang yang terinfeksi. Khusus *Covid-19*, masa inkubasi belum diketahui secara pasti. Namun, rata-rata gejala yang timbul setelah 2-14 hari setelah virus pertama masuk ke dalam tubuh.

II.2.5. Forward Chaining

Forward Chaining merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (*conclusion*) dari fakta tersebut. *Forward Chaining* bisa dikatakan sebagai strategi *inference* yang bermula dari sejumlah

fakta yang diketahui (Pangesti, 2021).

Penelitian ini menggunakan metode *Forward Chaining* dalam Sistem Pakar untuk menggambarkan alur kerja yang akan penulis lakukan untuk menyelesaikan penelitian. Proses akan dilakukan pengecekan terhadap setiap *rule*. Apabila memenuhi, maka *rule* akan dieksekusi untuk menghasilkan fakta baru yang mungkin digunakan oleh *rule* lain. Proses pengecekan *rule* disebut *rule interpretation*. Sistem akan menganalisa permasalahan dengan mencari fakta yang cocok dengan bagian *IF* dari aturan *IF-THEN*. Proses *Forward Chaining* dapat dilihat pada gambar III.1



Gambar II.1. Proses Forward Chaining

Dimulai dari premis-premis atas informasi masukan (*if*) dahulu kemudian menuju kesimpulan atau derived-information (*then*) atau dapat diuraikan sebagai berikut:

1. Informasi masukan (*if*)
2. Kesimpulan (*then*)

Diawali dengan informasi yang ada dan kesimpulan akan diperoleh. Informasi masukan mungkin berupa data, pengujian hasil dan pengamatan.

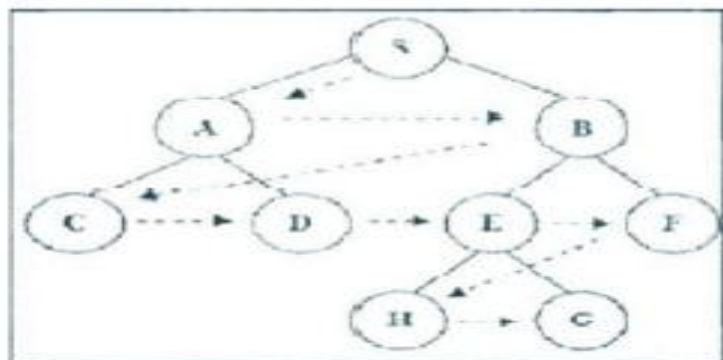
Sementara kesimpulan berupa tujuan, hipotesa, penjelasan, atau diagnosis. Secara umum untuk mempresentasikan pengetahuan dalam metode *Forward Chaining* dapat dibentuk dengan aturan (rule) **IF...THEN (Jika...maka)**.

II.2.6. Metode Pencarian

Untuk mendapatkan hasil yang diinginkan oleh *user*, sistem pakar menggunakan metode pencarian. Adapun metode pencarian sebagai berikut:

a. *Breadth First Search*

Pada metode *Breadth First Search* ini adalah semua *node* pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level $n+1$. Pencarian dimulai dari *node* akar terus ke level 1 dari kiri ke kanan, kemudian berpindah ke level berikutnya dari kiri ke kanan hingga solusi ditemukan.

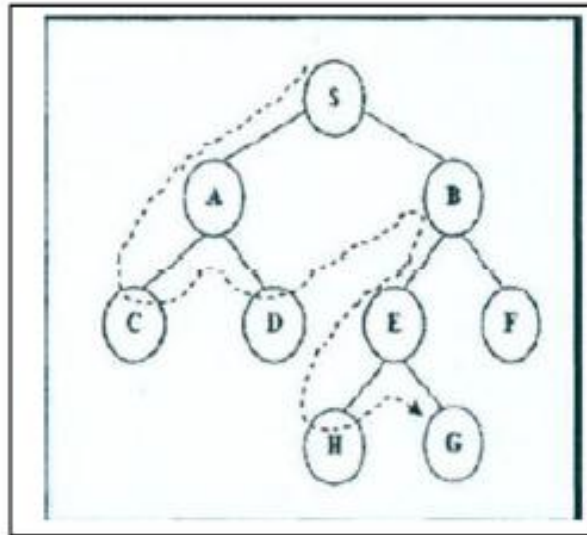


Gambar II.2. Metode *Breadth First Search*

b. *Depth First Search*

Pada metode *Depth First Search*, proses pencarian dilakukan pada semua anaknya sebelum dilakukan pencarian ke *node-node* yang selevel. Pencarian dimulai dari *node* akar ke level yang lebih tinggi. Proses diulangi terus hingga

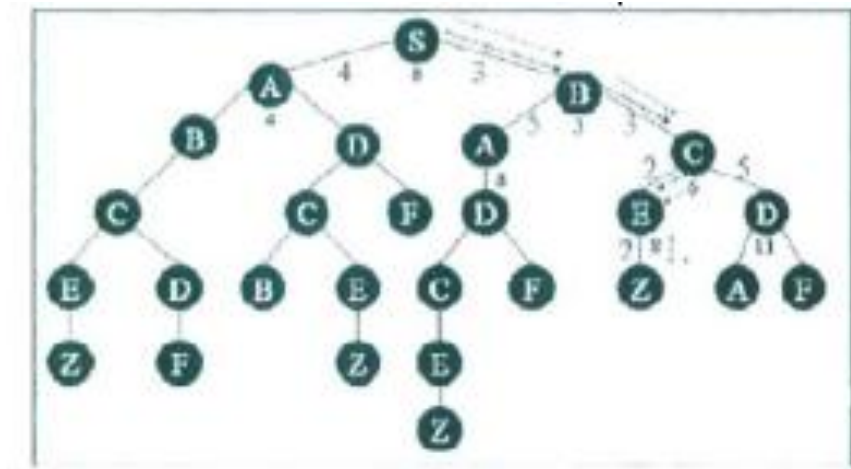
ditemukan solusi.



Gambar II.3. Metode *Dept First Search*

c. Metode Pencarian Terbaik Pertama (*Best FirstSearch*)

Merupakan kombinasi dari metode *depth first search* dan metode *breadth first search* dengan mengambil kelebihan dari kedua metode tersebut. Pada setiap langkah proses pencarian terbaik pertama, kita memilih *node-node* dengan menerapkan fungsi heuristik yang memadai pada setiap *node* atau simpul yang kita pilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Pada *best first search*, pencarian diperbolehkan mengunjungi node di lebih rendah dan jika ternyata node di level lebih tinggi memiliki nilai heuristik lebih buruk.



Gambar II.4. Metode *Best First Search*

II.2.7. Probabilitas

Instrumen penting sebagai salah satu penyelesaian masalah dalam *Artificial Intelligence* (AI) adalah probabilitas (peluang). Probabilitas merupakan peluang suatu kejadian, yaitu: suatu ukuran tentang kemungkinan suatu peristiwa (*event*) akan terjadi di masa mendatang. Probabilitas dinyatakan dengan nilai 0 sampai 1 atau dalam prosentase. Teori probabilitas diperkenalkan oleh Pascal dan Fermat pada tahun 1654 (Parrat, 1961). Banyak aplikasi probabilitas telah ditunjukkan dalam ilmu pengetahuan, teknik, bisnis, ekonomi dan bidang lainnya. Sifat probabilitas:

$$0 \leq P(A) \leq 1$$

Simbol P digunakan untuk melambangkan nilai probabilitas dari suatu kejadian. Dengan demikian $P(A)$ menyatakan probabilitas bahwa kejadian A akan terjadi dalam observasi atau percobaan tunggal. Nilai probabilitas terkecil adalah 0 (ini menyatakan suatu kejadian tidak mungkin terjadi) dan tertinggi adalah 1 (ini

menyatakan suatu kejadian yang pasti terjadi).

Ada tiga pendekatan konsep untuk mendefinisikan probabilitas dan menentukan nilai-nilai probabilitas, adalah sebagai berikut:

a. Pendekatan Klasik

Pendekatan klasik didasarkan pada banyaknya kemungkinan-kemungkinan yang dapat terjadi pada suatu kejadian. Setiap peristiwa memiliki kemungkinan yang sama untuk terjadi. Rumus probabilitas: banyaknya *outcome* memenuhi A, banyaknya *outcomes* keseluruhan.

$$P(A) = \frac{\text{banyaknya outcome memenuhi A}}{\text{banyaknya outcomes keseluruhan}}$$

b. Pendekatan Relatif

Dengan pendekatan ini nilai probabilitas ditentukan atas dasar proporsi dari kemungkinan yang terjadi dalam suatu observasi atau percobaan. Tidak ada asumsi awal tentang kesamaan kesempatan, karena penentuan nilai-nilai probabilitas didasarkan pada hasil observasi dan pengumpulan data. Konsep frekuensi *relative*, yaitu jika eksperimen dilakukan n kali, dan kejadian A teramati f kali, maka rumusnya:

$$P(A) = \frac{f}{n}$$

c. Pendekatan Subjektif

Dalam penentuan nilai probabilitas adalah tepat apabila hanya ada satu kemungkinan kejadian yang terjadi dalam satu kejadian. Dengan pendekatan ini, nilai probabilitas dari suatu kejadian ditentukan berdasarkan tingkat kepercayaan yang bersifat individual dengan berlandaskan pada semua petunjuk yang

dimilikinya.

II.2.8. Kelebihan dan Kekurangan Forward Chaining

Kelebihan dari metode *Forward Chaining* adalah sebagai berikut:

1. Metode akan bekerja dengan baik ketika problem bermula dari mengumpulkan atau menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
2. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data.

Adapun kekurangan dari metode *Forward Chaining* adalah sebagai berikut:

1. Kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem bisa saja menyatakan pertanyaan yang tidak berhubungan. Walaupun jawaban dari pertanyaan tersebut penting, hal ini akan membingungkan user untuk menjawab pada subjek yang tidak berhubungan.

II.2.9. Studi Kasus

Peningkatan kasus *Corona Virus* berkembang cepat di masyarakat. Dari informasi yang di dapat pada kantor Dinas Kesehatan Provinsi Sumatera Utara berdasarkan laporan gugus tugas penanganan *Covid-19* bidang SURVEILANS data per Kabupaten/Kota di provinsi Sumatera Utara yang di *update* tanggal 16 November 2021, dari 33 Kabupaten/Kota serta domisili luar sumatera dan informasi domisili belum jelas, jumlah kasus terkonfirmasi sebanyak 256. 497 kasus dengan total sembuh mencapai 102. 941 kasus dan 2889 kasus meninggal.

Kasus *Covid-19* masih dapat terus bertambah seiring dengan timbulnya varian baru yang menyebabkan kepanikan di masyarakat. Dengan data yang berhasil penulis peroleh baik dari kantor Dinas Kesehatan Provinsi Sumatera Utara dan beberapa jurnal ilmiah terkait *Corona Virus* berupa data gejala, data penyakit, dan data cara penanggulangan, penulis akan membangun sebuah sistem yang dapat memberikan pengetahuan kepada masyarakat agar lebih bijak dalam menghadapi *Corona Virus*. Dengan penggunaan metode *Forward Chaining* dalam pembuatan sistem pakar dengan teknik pelacakan ke depan, penulis melakukan pengumpulan data-data yang diperlukan, yang nantinya data tersebut akan menjadi informasi yang ada dengan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan.

II.3. Database

Basis data dapat didefinisikan dalam sejumlah sudut pandang, antara lain:

1. Menurut Sitinjak (2020), “Basis data atau *database* adalah sekumpulan tabel-tabel yang berisikan sekumpulan data yang fakta sebagai sumber informasi yang disimpan dalam media penyimpanan secara digital dan dapat diperiksa menggunakan suatu program komputer yang berguna untuk memudahkan aktivitas untuk memperoleh informasi.
2. Menurut Jayanti & Sumarni (2018), “Basis Data atau *Database* merupakan data yang terintegrasi, yang diorganisasikan untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi.”

Berdasarkan pendapat diatas, dapat disimpulkan bahwa *Database* adalah

sebuah struktur data atau kumpulan informasi yang telah disusun dan disimpan dalam media penyimpanan secara digital untuk dapat memenuhi kebutuhan pemakai di dalam suatu organisasi.

II.4. PHP (*Hypertext Preprocessor*)

PHP merupakan sebuah bahasa untuk membuat halaman *web* yang interaktif yang menyatu dengan HTML yang di jalankan di sisi server. *Server side scripting* digunakan untuk pembangun halaman *web* yang berisikan sintaks dan perintah-perintah yang diberikan dan sepenuhnya dijalankan di server kemudian disertakan pada dokumen HTML. Bahasa ini sangat memungkinkan aplikasi *web* menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server (Syam & Arie, 2018) . Sedangkan Nugroho (Suryadi & Zulaikhah, 2019) hanya menyimpulkan bahwa PHP adalah bahasa program yang berbentuk skrip yang diletakan di dalam *server web*.

Dapat disimpulkan dari kutipan diatas, bahwa *hypertext preprocessor* (PHP) merupakan bahasa pemrograman untuk membuat atau mengembangkan aplikasi berbasis *web* dan bersifat *open soure* dan ditanamkan ke dalam *script* HTML.

II.5. MySQL

MySQL menjadi populer di zaman sekarang ini karena diantaranya terdapat beberapa fitur pendukung yang sangat membantu dalam membuat sebuah *website*. Fitur-fitur yang mendukung didalam MySQL ini diantaranya adalah *multithreaded*, *multi-user*, dan *SQL databasemanagement system* (DBMS).

Database ini sengaja dibuat memang digunakan untuk keperluan sistem *database* yang cepat, handal, dan mudah digunakan sehingga sangat membantu sekali para pembuat *website* baik yang masih pemula maupun yang sudah terbiasa dalam membuat *webiste* (Hidayat & Faisal, 2019). Faizal dan Irnawati juga menjelaskan bahwa MySQL ini sebagai perangkat lunak pada sebuah sistem manajemen berbasis data SQL (*database management system*) atau DBMS yang memiliki sifat *multi thread* dan *multi user*, dengan jumlahnya berkisar kurang lebih enam juta instalasi di seluruh dunia (Suryadi & Zulaikhah, 2019).

Dari definisi diatas, dapat disimpulkan bahwa sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*), database ini *multithread*, *multi user*.

II.6. Unified Modeling Language (UML)

UML menggambarkan perancangan awal dari sistem yang akan dibangun. UML memiliki banyak jenis permodelan. Berikut ini definisi menurut para ahli :

1. *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan permodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada

metodologi berorientasi objek (Sukamto dan Shalahuddin, 2018).

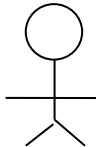
2. UML ini merupakan sebuah standarisasi bahasa pemodelan muncul pada perkembangan teknik pemrograman berorientasi objek untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek (Hariyanto & Prasetyo, 2019).

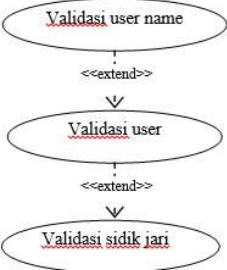
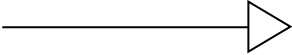
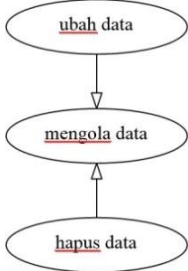
Adapun jenis-jenis UML yang digunakan dalam perancangan sistem, antara lain:

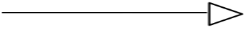
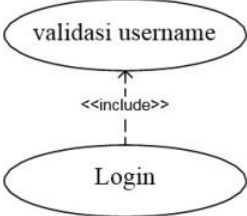
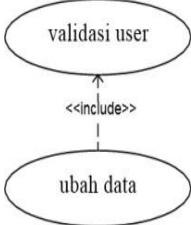
a. *Use Case Diagram*

Merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. Usecase mendeskripsikan sebuah interaksi antara satu atau lebih. Use case digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol diagram use case yang dapat dilihat pada tabel

II.1

No	Simbol	Deskripsi
1.	<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	<i>Actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

3.	Assosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal  Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> induknya.
5.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya(umum)
6.	Menggunakan / <i>include</i> / <i>uses</i> <<include>>  <<uses>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : - <i>Include</i> berarti <i>use case</i> yang

		ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :  <pre> graph BT Login([Login]) -- "<<include>>" --> validasi_username([validasi username]) </pre> - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :  <pre> graph BT ubah_data([ubah data]) -- "<<include>>" --> validasi_user([validasi user]) </pre> Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	---	---

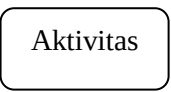
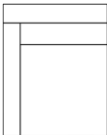
Tabel II.1 Simbol-simbol dalam Use Case Diagram

Sumber : (Sukamto dan Shalahuddin, 2018)

b. *Activity Diagram*

Sukamto dan Shalahuddin (2018), “diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas

yang dapat dilakukan oleh sistem”. Keterangan simbol- simbol yang ada pada diagram aktivitas dapat dilihat pada tabel II.2

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan katakerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

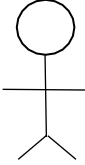
Tabel II.2 Simbol-simbol dalam Activity Diagram

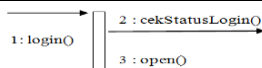
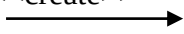
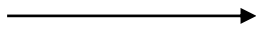
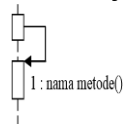
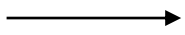
Sumber : (Sukamto dan Shalahuddin, 2018

c. *Sequence Diagram*

Sukamto dan Shalahuddin (2018), “diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat

dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak”. Keterangan simbol-simbol yang ada pada sequence diagram dapat dilihat pada tabel II.3.

No	Simbol	Deskripsi
1.	Aktor  Atau <u>Nama aktor</u> Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	Garis hidup/<i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	Objek <u>Nama objek : nama kelas</u>	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalkan:

		 Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif
5.	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirim data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

Tabel II.3 Simbol-simbol dalam Sequence Diagram

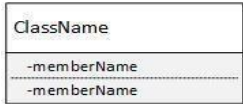
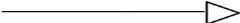
Sumber : (Sukamto dan Shalahuddin, 2018)

d. *Class Diagram*

Sukamto dan Shalahuddin (2018), “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variabel-variabel yang dimiliki suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Keterangan simbol class diagram dapat dilihat pada tabel II.4

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/<i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/<i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/<i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)
6.	Kebergantungan/<i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/<i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Tabel II.4 Simbol-simbol dalam Class Diagram

Sumber : (Sukamto dan Shalahuddin, 2018)