BAB II
TINJAUAN PUSTAKA

2.1 Sistem Pakar

Sistem pakar atau *Expert System*, juga dikenal sebagai sistem basis pengetahuan, adalah aplikasi komputer yang dirancang untuk membuat keputusan atau memecahkan masalah di bidang tertentu. Sistem ini bekerja dengan mengidentifikasi pengetahuan analitis dan metode yang sebelumnya ditentukan oleh ahli sesuai dengan disiplin ilmu. Sistem pakar tidak menggantikan peran pakar itu sendiri, tetapi dapat digunakan sebagai asisten yang handal dan sangat berpengalaman (Hayadi, 2018). Sistem pakar merupakan program komputer yang berisi pengetahuan dari satu atau lebih pakar dalam bidang tertentu. Sistem ini bekerja dengan pengetahuan dan metode analisis yang telah didefinisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya (Zufria & Santoso, 2021).

2.1.1 Pengertian Sistem Pakar Menurut Beberapa Ahli

Sistem pakar merupakan bidang *Artificial Intelligent* (AI) yang cukup tua karena sistem ini mulai dikembangkan pada tahun 1960. Sistem pakar merupakan salah satu bidang kecerdasan buatan yang cukup diminati untuk aplikasi di berbagai bidang baik ilmu pengetahuan maupun bisnis, dan telah terbukti sangat berguna untuk pengambilan keputusan dan berbagai aplikasi yang sangat luas. Sistem pakar adalah suatu sistem komputer yang dirancang agar dapat dilakukan penalaran seperti layaknya seorang pakar pada suatu bidang keahlian tertentu menurut shelly, 1990; setiawan, 1993; margianti, 1995 (Hayadi, 2018).

2.1.2 Kelebihan dan Kekurangan Sistem Pakar

1. Kelebihan Sistem Pakar

   Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan kelebihan yang diberikandiantaranya:

a. Sistem pakar dapat memberikan pengetahuan kepada masyarakat umum dan bertindak seperti pakar/expert.
b. Semua informasi yang diterima memungkinkan sistem pakar untuk terus bekerja.
c. Sistem pakar bekerja lebih cepat, yang meningkatkan produktivitas.
d. Sistem pakar selalu aktif (tidak pernah lelah) konsisten memberikan jawaban, dan memperhatikan konsekuensi input pengguna.
e. Sistem pakar dapat menjangkau jarak yang jauh dan luas. Dengan menggunakan sistem pakar, pengguna seolah-olah berbicara langsung dengan seorang pakar. Meskipun si pakar telah tiada.
f. Sistem pakar memiliki kemampuan untuk memecahkan masalah yang rumit dan rumit yang hanya dapat dikuasai oleh para pakar.

2. Kekurangan Sistem Pakar

Selain kelebihan, ada juga beberapa kekurangan yang ada pada Sistem Pakar, diantaranya:

a. Sistem pakar hanya dapat memproses input pengetahuan ke dalam sistem, dan hasilnya pasti sesuai dengan alur penalaran input. Karena bersifat dinamis dan berubah dari waktu ke waktu, basis pengetahuan harus selalu diperbaharui.
b. Sistem pakar hanya menangani kepastian berupa saran atau rekomendasi, bukan keputusan.
c. Basis pengetahuan memiliki format terbatas dan berisi aturan yang ditulis dalam bentuk pernyataan *if–then*(Pratiwi, 2019).

2.1.3 Ciri-ciri Sistem Pakar

Sistem pakar memiliki ciri-ciri sebagai berikut:

a. Terbatas pada area keahlian tertentu.
b. Dapat memberikan alasan data tidak aman.
c. Dapat menjelaskan berbagai alasan dengan cara yang mudah dipahami.
d. Berdasarkan kaidah atau rule tertentu.

e. Dirancang untuk dapat dikembangkan secara bertahap.
f. Keluarannya bersifat anjuran(Kasman Rukun, 2018).

2.1.4 Klasifikasi Sistem Pakar

Klasifikasi sistem pakar berdasarkan kegunaannya yaitu:

1. Diagnosis:
   a. Digunakan untuk menyarankan obatuntuk orang sakit, kerusakan mesin,dan kerusakan rangkaian elektronik.
   b. Menemukan apa kerusakan/masalah yang terjadi.
   c. Menggunakan pohon keputusan (*decision tree*) sebagai representasi pengetahuannya.
2. Pengajaran:
   a. Digunakan untuk pengajaran, mulai dari SD sampai dengan PT.
   b. Membuat diagnosis apa penyebab kekurangannya dari siswa, kemudian memberikan cara untuk memperbaikinya.
3. Prediksi:
   a. Untuk peramalan cuaca.
   b. Penentuan masa tanam(Hayadi, 2018).

2.1.5 Konsep Dasar Sistem Pakar

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

a. Keahlian

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

b. Ahli (*Expert*)

   Melibatkan kegiatan mengidentifikasi dan meyusun permasalahan, memecahkan masalah dengan cara cepat dan tepat,menerangkan pemecahannya, belajar dari pengalaman dan lain sebagainya.

c. Mentransfer Keahlian

Adalah proses pentransferan keahlian dari seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan pakar.

d. Menyimpulkan Aturan

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

e. Peraturan (*Rule*)

Diperlukan karena kebanyakan dari sistem pakar bersifat rule based sistem, yang berarti pengetahuan disimpan dalam bentuk peraturan.

f. Kemampuan menjelaskan

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

2.1.6 Komponen Sistem Pakar

Sebuah program sistem pakar terdiri atas beberapa komponen yang mutlak harus ada. Komponen itu adalah sebagai berikut:

a. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan inti program sistem pakar karena basis pengetahuan ini merupakan representasi pengetahuan 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 sedang dilakukan.

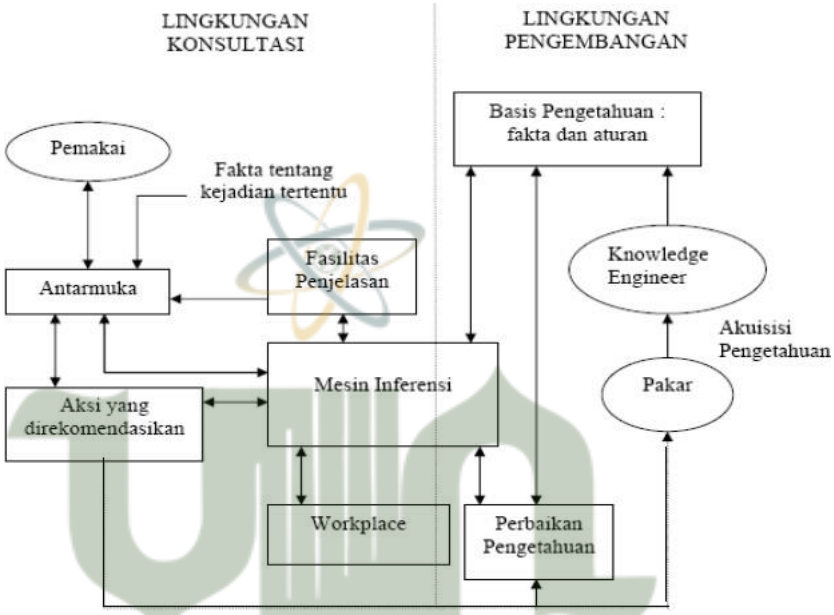
c. Mesin Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian dan strategi penalaran. Strategi pengendalian fungsinya untuk panduan arah dalam melakukan proses penalaran. Ada tiga teknik pengendalian

yang digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

d. Antar Muka Pemakai (*User Interface*)

Antar muka pemakai adalah bagian penghubung antara program sistem pakar dengan pemakainya. Bagian ini terjadi dialog antara program dengan pemakai (Hayadi, 2018).



Gambar 2. 1 Komponen Sistem Pakar

(Sumber: Dahria, 2021)

2.2 Periodontitis

Penyakit periodontitis adalah salah satu penyakit gigi dan mulut yang sering dialami oleh perokok. Periodontitis merupakan suatu proses penyakit *inflamasi* yang mempengaruhi struktur penyangga gigi (*ligamen periodontal*), sementum dan tulang *alveolar*. Secara bertahap, periodontitis ini dapat menyebabkan kerusakan pada tulang *alveolar* dan *ligamen peridontal* yang kemudian menyebabkan hilangnya gigi. Kasus periodontitis dikaitkan dengan buruknya kebersihan mulut yang berpengaruh terhadap tekstur bakteri *gingiva*. Periodontitis biasanya berkembang dari *gingivitis* yang sudah terjadi, walaupun tidak semua *gingivitis* berkembang menjadi periodontitis (Vinay Kumar et al., 2019).

2.2.1 Gigi dan Mulut

Gigi merupakan organ pengunyah yang terdiri dari gigi-gigi pada rahang bawah, lidah dan saluran-saluran penghasil air ludah. Gigi secara tidak langsung mempengaruhi kesehatan seseorang, karena fungsi utama gigi adalah untuk merobek dan mengunyah makanan dalam sistem pencernaan. Di dalam mulut, makanan dipecah oleh gigi dan dilumasi oleh air liur sebelum masuk ke lambung. Mulut adalah tempat yang ideal bagi bakteri untuk tumbuh karena suhu dan kelembabannya (Arfajsyah et al., 2018).

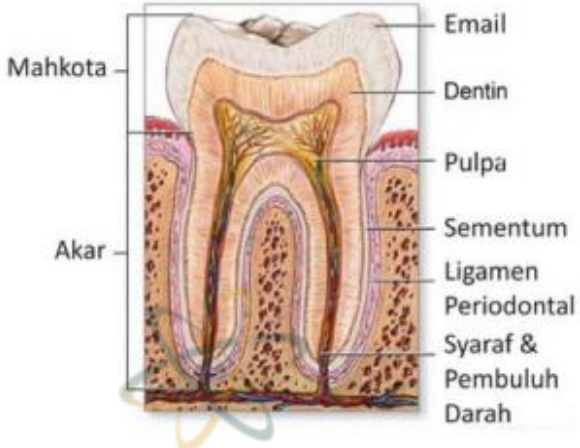
2.2.2 Anatomi Gigi

Anatomi gigi dibagi menjadi dua bagian dasar. Mahkota adalah bagian putih gigi yang dapat dilihat. Bagian kedua adalah akar gigi yang tersembunyi. Akar gigi memanjang di bawah garis gusi dan membantu mengikat gigi ke tulang. Gigi memiliki berbagai jenis jaringan dan masing-masing memiliki tujuan tertentu. Anatomi gigi tersebut yaitu:

a. *Enamel* adalah bagian luar gigi yang paling keras dan putih dari gigi. *Enamel* melindungi bagian-bagian dalam gigi dari rangsangan panas dan dingin, yang sebagian besar terbuat dari fosfat dan kalsium.
b. *Dentin* adalah lapisan di bawah *enamel*. Ini adalah jaringan keras yang mengandung tabung kecil. Saat *enamel* rusak, suhu dingin atau panas dapat masuk gigi melalui jalur ini dan menyebabkan sensitifitas gigi atau rasa sakit.
c. *Cementum* adalah lapisan jaringan ikat yang melekatkan akar gigi ke gusi dan tulang rahang. Untuk menjaga jaringan lunak tetap sehat, kita harus merawat gusi dengan baik. *Cementum* berwarna kuning pucat dan biasanya ditutupi oleh gusi dan tulang. Jika tidak merawat gigi dengan baik, gusi bisa menjadi sakit dan surut, yang dapat menyebabkan akumulasi bakteri berbahaya dan plak di *cementum*.
d. *Pulpa* adalah bagian dalam anatomi gigi yang lebih lembut, dapat ditemukan di pusat dan inti gigi serta berisi pembuluh darah, saraf, dan jaringan lunak lainnya. Bagian ini berguna untuk memberikan sinyal ke gigi dan nutrisi.
e. *Periodontal ligamentum* adalah serabut-serabut yang menyelubungi akar gigi

yang melekat pada tulang alveolar dan *cementum*. Gunanya untuk membantu menahan gigi dengan kuat melawan rahang.

f. Gusi adalah jaringan lunak yang menutupi dan melindungi akar gigi. Gusi tidak menempel pada gigi (Jusuf Kristianto, 2022).



Gambar 2. 2 Anatomi Gigi

(Sumber: Astuti & Mulawarman, 2020)

2.3 Rokok

Rokok adalah tabung kertas yang panjangnya sekitar 70 hingga 120 mm dan memiliki diameter sekitar 10 mm. Rokok terbuat dari daun tembakau kering cincang. Tembakau adalah kombinasi bahan kimia, rokok yang dibakar mengeluarkan 4000 bahan kimia. Rokok menyebabkan pembakaran yang tidak sempurna dan dapat menumpuk di dalam tubuh saat dihisap. Secara umum, komponen rokok dapat dibagi menjadi dua kelompok utama yaitu komponen gas (92%) dan komponen padat atau partikel (8%). Asap tembakau terdiri dari Karbon monoksida, Karbon dioksida, Hidrogen sianida, Amonia, Nitrogen oksida, dan senyawa Hidrokarbon. Timah hitam (Pb) juga merupakan bahan yang sangat berbahaya dalam rokok. Rokok mengandung hingga 0,5 µg partikel ini. Ambang batas timbal dalam tubuh adalah 20 mg perhari. Efek merokok juga dipengaruhi oleh jumlah rokok yang dihisap, durasi merokok, jenis rokok yang dihisap, bahkan berhubungan dengan dalamnya hisapan rokok yang dilakukan.

Bahan kimia dalam rokok dapat bervariasi dalam jumlah. Kekuatan suatu produk rokok tergantung dari jenis dan merek produk tersebut. Diketahui bahwa kandungan yang paling banyak ditemukan pada rokok dan paling berbahaya bagi kesehatan adalah Nikotin, Tar, dan Karbon Monoksida (Mega et al., 2019).

2.4 Efek Merokok pada Kesehatan Gigi dan Mulut

Dari penelusuran literatur, ditemukan bahwa merokok memiliki efek negatif pada kondisi sistemik, serta lingkungan lokal rongga mulut. Kanker paru-paru, penyakit *kardiovaskular*, tumor laring dan *esofagus*, adalah penyakit sistemik yang berhubungan dengan kebiasaan merokok, efek lokal merokok pada gigi dan rongga mulut termasuk *gingivitis*, periodontitis, karies akar, kehilangan tulang *alveolar*, kehilangan gigi, dan berhubungan dengan munculnya lesi khas pada jaringan lunak di rongga mulut.

Merokok menyebabkan periodontitis dengan menyebabkan plak menumpuk pada gigi dan *gingiva*. Mengendapnya tar pada gigi dapat menyebabkan masalah pada penampilan dan juga menyebabkan permukaan gigi menjadi kasar, sehingga memudahkan pembentukan plak. Kondisi kebersihan mulut yang buruk dapat menyebabkan *gingivitis*.

Gingivitis yang tidak diobati dapat berkembang menjadi periodontitis sebagai akibat dari *invasi* bakteri kronis pada plak di bawah garis gusi. Peningkatan *vaskularisasi*, diikuti oleh akumulasi sel *inflamasi* kronis, menyebabkan hilangnya kolagen pada jaringan ikat *gingiva* yang terbuka. Kehilangan perlekatan gigi pada *gingiva* (gusi) dapat menyebabkan *resesi gingiva* (gusi turun), yang dapat menyebabkan kerusakan gigi.

Merokok merupakan faktor risiko kanker mulut, yang dapat meningkatkan kemungkinan terkena penyakit ini sekitar 2-4 kali lipat. Iritasi kronis tar karsinogenik menyebabkan perubahan awal pada struktur dasar epitel mukosa mulut, seperti *atrofi*, *deskuamasi*, dan *keratosis*, dan bahkan dapat menyebabkan *papila filiformis* membesar, yang pada gilirannya menyebabkan perkembangan rasa sakit. Pembakaran rokok coklat meninggalkan banyak abu di lidah perokok, sehingga sulit untuk merasakan rasa manis, asin dan pahit.

2.5 Metode *Fuzzy Logic Sugeno*

Metode *Fuzzy Logic Sugeno* merupakan pendekatan sistematis pembangkitan aturan *fuzzy* dari himpunan data masukan-masukan yang diberikan. Metode *Fuzzy Sugeno* memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai *THEN*. Pada perubahan ini, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values)* di dalam bagian aturan *Fuzzy IF THEN* (Sari Mutiara Siregar, 2021).

Fuzzy Sugeno adalah salah satu metode yang digunakan untuk memperoleh hasil suatu diagnosis pada sistem pakar. Pada sistem diagnosis *fuzzy* peranan operator/manusia lebih dominan. Data yang dikirim akan dilaksanakan oleh operator ke dalam sistem. Operator dapat meminta atau bertanya mengenai informasi dari sistem diagnosis berupa hasil pendapat atau prosedur detail hasil diagnosis oleh sistem. Tahapan metode *Fuzzy Sugeno* adalah pembentukan himpunan *fuzzy* mengaplikasikan fungsi implikasi (aturan). Komposisi aturan, didapat dari kumpulan data hubungan antar aturan. Penegasan (*Defuzzifikasi*), input dari *defuzzifikasi* adalah konstanta atau persamaan linear (Ricardo, 2021).

Berikut ini tahapan-tahapan yang digunakan dalam metode *Fuzzy Logic Sugeno*, yaitu:

1. Pembentukan himpunan *fuzzy*

   Tahapan ini yaitu tahapan untuk mengubah variabel numerik (variabel non *fuzzy*) berupa bobot nilai, batas interval minimum dan maksimum dari gejala yang dipilih menjadi variabel linguistik (variabel *fuzzy*) dengan rumus *fuzzifikasi* sehingga didapatkan nilai *fuzzy*.

$$\mu[x,a,b,c] = \begin{cases} 0 & x \leq a \text{ atau } x \geq c \\ ; \frac{(x-a)}{(b-a)} & ; a \uparrow \leq x\ b \\ ; \frac{(c-x)}{(c-b)} & ; b \uparrow \leq x\ c \end{cases} \quad (1)$$

Keterangan:

X : Bobot nilai dari pakar

a : Batas nilai minimum pada setiap gejala

b : Nilai tengah dari batas minimum dan maksimum

c : Batas nilai maksimum pada setiap gejala

$$b = \frac{\sum asampaib}{n} \quad (2)$$

2. Aplikasi fungsi implikasi

Tahapan ini menghitung nilai *fuzzifikasi* dari gejala yang digunakan dengan rumus sebagai berikut:

$$F = \frac{(X-a)}{(b-a)} \quad (3)$$

3. Defuzzifikasi

Tahapan ini merupakan tahapan akhir dari logika *fuzzy* dimana setelah dilakukan *fuzzifikasi* pada tiap gejala yang dipilih, kemudian dari gejala-gejala tersebut diproses berdasarkan aturan dari fungsi implikasi yang telah dibuat sehingga didapatkan hasil diagnosa. Rumus umum untuk *defuzzifikasi* metode *Fuzzy Logic Sugeno* adalah sebagai berikut:

$$WA = \frac{a1z1+a2z2+a3z3+\cdots+anzn}{a1+a2+a3+\cdots an} \quad (4)$$

Keterangan :

WA = (*Weighted Average*) Nilai rata-rata

a_n= Nilai predikat aturan ke-n

z_n= Indeks nilai input (konstanta) ke-n

2.6 Penerapan Metode *Fuzzy Logic Sugeno*

Berikut ini merupakan penerapan metode *fuzzy sugeno*, pada penelitian ini dilakukan untuk tahap-tahap perhitungan dalam menyelesaikan masalah pada proses mendiagnosis penyakit *Iskemia*. Untuk tahap ini dihasilkan perhitungan yang dimana perhitungan tersebut akurat untuk hasil diagnosis penyakit *Iskemia*. Untuk menerapkan metode *fuzzy sugeno* pada sistem, ada beberapa variabel yang

diperlukan, yaitu bobot nilai dari setiap gejala, batas nilai, minimum setiap gejala, batas nilai maksimum setiap gejala, dan aturan yang menunjukkan gejala-gejala yang dimiliki oleh penyakit. Berikut basis pengetahuan gejala dari penyakit Iskemia adalah sebagai berikut:

Tabel 2. 1 Basis Pengetahuan Gejala *Iskemia*

(Sumber: Sari Mutiara, 2021)

Kode	Gejala	Kategori	Bobot	Interval
KG1	Nyeri dada seperti tertekan	Ringan	0,15	$0,0 \leq a \leq 0,4$
KG2	Mual dan muntah	Ringan	0,15	$0,0 \leq a \leq 0,4$
KG3	Nyeri pada leher, rahang, bahu, atau lengan	Sedang	0,5	$0,3 \leq a \leq 0,7$
KG4	Tubuh terasa lemas	Sedang	0,5	$0,3 \leq a \leq 0,7$
KG5	Mengeluarkankeringat yang banyak	Sedang	0,5	$0,3 \leq a \leq 0,7$
KG6	Sesak napas, terutama saat melakukan aktivitas fisik	Parah	0,8	$0,6 \leq a \leq 1$
KG7	Detak jantung menjadi lebih cepat	Parah	0,8	$0,6 \leq a \leq 1$

Berikut basis pengetahuan berdasarkan asumsi dari pakar dan penerapan dari logika *fuzzy* maka *range* interval dibagi menjadi 3 kategori, yaitu ringan dengan kisaran ($0,0 \leq a \leq 0,4$), sedang dengan kisaran ($0,3 \leq a \leq 0,7$), dan parah dengan kisaran ($0,6 \leq a \leq 1$). Untuk contoh kasus dalam penelitian ini yaitu seorang pasien mengalami gejala mual dan muntah, tubuh terasa lemas, mengeluarkan keringat yang banyak, dan sesak napas, terutama saat melakukan aktivitas fisik.

Berikut penyelesaian diagnosis penyakit *iskemia* menggunakan *fuzzy logic sugeno*:

1. Menghitung Proses *Fuzzifikasi*

   a. Kategori Ringan Dengan Interval 0,0≤ a ≤ 0,4

   $b=\frac{0,0+0,1+0,2+0,3+0,4}{5}=\frac{1}{5}=0,2$

   b. Kategori Sedang Dengan Interval 0,3 ≤ a ≤ 0,7

   $b=\frac{0,3+0,4+0,5+0,6+0,7}{5}=\frac{1}{5}=0,5$

   c. Kategori Parah Dengan Interval 0,6 ≤ a ≤ 1

   $b=\frac{0,6+0,7+0,8+0,9+1}{5}=\frac{4}{5}=0,8$

2. Menghitung Nilai *Fuzzifikasi*

   a. Menghitung F (G02)

   $F=\frac{0,15-0,0}{0,2-0.0}=0,75$

   b. Menghitung F (G04,G05)

   $F=\frac{0,5-0,3}{0,5-0,3}=1$

   C. Menghitung F (G06)

   $F=\frac{0,8-0,6}{0,8-0,6}=1$

3. Menghitung Proses *Defuzzifikasi*

   WA = (FG02 *BNG02)+(FG04 *BNG04)+(FG05 * BNG05)+(FG06 *BNG06)/FG02+ FG04 + FG05 +FG06

   = ((0,75*0,15)+(1*0,5)+(1*0,5)+(1*0,8))/0,75+1+1+1

   = 0,1125+0,5+0,5+0,8/3,75

   = 1,9125/3,75

   = 0,51

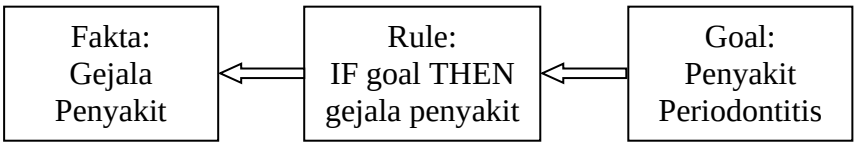
Tingkat keparahan penyakit = 0,51*100% = 51%

Berdasarkan perhitungan menggunakan metode *fuzzy logic sugeno* yang telah dilakukan terkait dengan gejala-gejala penyakit *Iskemia* yang dialami oleh pasien terhadap G2, G4, G5, dan G6 dapat disimpulkan bahwa hasil *defuzzifikasi* hasil diagnosis *Iskemia* pada tubuh pasien adalah 51% (Sari Mutiara Siregar, 2021).

2.7 *Backward Chaining* (Runut Balik)

Metode *Backward Chaining* adalah metode pelacakan ke belakang yang memulai penalarannya dari kesimpulan (*goal*), sesuai dengan fakta atau pernyataan yang dimulai dari sisi kanan. Dengan kata lain, penalaran yang dimulai dari hipotesis harus mencari basis pengetahuan untuk fakta-fakta. Teknik pencarian yang mengambil fakta yang diketahui dan mencocokkannya dengan *IF* dari aturan *IF-THEN* (Zufria & Santoso, 2021).

Backward Chaining merupakan metode penalaran kebalikan dari *forward chaining*. Proses ini mirip dengan pengujian hipotesis dalam pemecahan masalah manusia. Misalnya, seorang dokter mungkin mencurigai seorang pasien memiliki masalah dan mencoba membuktikannya dengan mencari gejala-gejala tertentu. *Backward Chaining* mencari *rule* inferensi hingga menemukan aturan dengan klausa *THEN* yang cocok dengan tujuan yang diinginkan. Jika kalusa *IF* aturan inferensi ini ternyata salah, maka akan ditambahkan ke daftar tujuan (agar tujuan dikonfirmasi, itu juga harus menyediakan data yang mengkonfirmasi aturan baru ini). Metode ini mencari aturan yang dapat memperoleh informasi yang diinginkan untuk tujuan dan mencoba untuk memenuhi beberapa aturan tersebut. *Backward Chaining* ini mengadopsi model pencarian *Depth-First* (Kiswanto et al., 2022). Secara sederhana gambar alur *Backward Chaining* penyakit periodontitis pada penelitian ini dapat dilihat pada gambar 2.3.



Gambar 2. 3*Backward Chaining* Penyakit Periodontitis

(Sumber: Kiswanto et al., 2022)

Berikut adalah contoh kasus sederhana untuk perhitungan manual *backward chaining*. Perhitungan algoritma *backward chaining* untuk diagnosis penyakit dan hama tanaman padi, yang menjadi tujuan utama ialah penyakit dan hama yang di alami petani. Fakta tentang aturan gejala-gejala yang ditimbulkan oleh penyakit dan hama tanaman padi sesuai dengan pengetahuan pakar. *Rule-rule* yang digunakan sesuai dengan tabel 2.2.

Tabel 2. 2 Aturan Algoritma *Backward Chaining*

(Sumber: Hidayatus dkk, 2022)

Aturan	**Daftar Rule *IF-Then***
Rule 1	*IF* (V,D) *Then* U
Rule 2	*IF* (W, A, B) *Then* U
Rule 3	*IF* C,W
Rule 4	*IF* L, M
Fakta	A, B, C dan D bernilai benar

Keterangan :

Fakta-fakta yaitu A,B,C,D merupakan gejala-gejala. Sedangkan U,V,W,L,M dan X adalah basis pengetahuan yang terdapat dalam algoritma *backward chaining* sesuai dengan pengetahuan pakar. Rumus perhitungan presentasi algoritma *backward chaining* menggunakan probabilitas.

$$\mathrm{P(X)} = \frac{n!}{(n-X)!\,X!} . P^{x} . q^{n-x}$$

Keterangan :

P(X) : Peluang terjadinya kejadian X

X : Banyaknya kejadian X

n : Jumlah objek yang bisa dipilih

p : Peluang sukses

n : Peluang gagal, yaitu (1-p) atau jika p dalam persentase maka q=(100%-p)

Hasil algoritma *backward chaining* adalah tahapan yang dilakukan dalam implementasi sebuah algoritma untuk menganalisis keakuratan hasil sesuai dengan yang diinginkan pakar. Tahapan ini berfungsi untuk mengamati eksekusi melalui data uji dan memeriksa kebenaran fungsional. Hasil algoritma *backward chaining* dapat dilihat pada tabel 2.3.

Tabel 2. 3 Hasil Algoritma *Backward Chaining*

(Sumber: Hidayatus dkk, 2022)

No	Skenario Pengujian (Pilih masukan)	Pakar	Hasil	Kesimpulan
1.	A,B,C,D	*Blast*	*Blast*	Sesuai
2.	A,B,C,D,E	Hawar daun	*Blast*	Tidak Sesuai
3.	A,B,C,D,E F,G	*Blast* Hawar daun	*Blast* Hawar daun	Sesuai
4.	A,B,C	*Blast*	*Blast*	Sesuai
5.	E,F,G,H	Hawar daun	Hawar daun	Sesuai
...	...	...	...	...
45.	W,X,Y,Z	Putih palsu	Putih palsu	Sesuai

Keterangan :

A-Z : Data gejala penyakit dan hama tanaman padi sesuai dengan keterangan tabel 2.2. Berdasarkan hasil algoritma *backward chaining* yang dilakukan pada tabel 2.3 yaitu 45 data pengujian. Pengujian dilakukan terhadap pakar dinas pertanian, diperoleh rumus sesuai dengan aturan yang diberikan oleh dosen pembimbing. Maka diperoleh rumus sesuai dengan perhitungan.

$$\text{Nilai Akurasi} = \frac{Semua\ data\ sesuai}{Semua\ data} \text{ x } 100\%$$

$$\text{Nilai Akurasi} = \frac{37\ Data}{45\ Data} \text{ x } 100\%$$

$$\text{Nilai Akurasi} = 82{,}21\%$$

Kesimpulan dari perhitungan rumus tersebut adalah 45 data pengujian mendapatkan hasil sesuai dengan pengetahuan pakar sebanyak 37 data sesuai dari 45 data pengujian. Pengujian akurasi menghasilkan akurasi pengujian sebesar 82,21% (Hidayatus Sholikhin, Deden Wahiddin, 2022).

2.8 *Java*

Java merupakan salah satu dari sekian banyak bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi, termasuk telepon gengam. Bahasa

pemrograman ini pertama kali dikembangkan ketika James Gosling masih di *Sun Microsystem*. Bahasa pemrograman ini merupakan pengembangan C++, *Java* saat ini merupakan bahasa pemrograman yang paling banyak digunakan dan banyak digunakan dalam pengembangan berbagai jenis perangkat lunak aplikasi atau aplikasi berbasis web (Harumy, T.H.F., Julham Sitorus, 2018).

Java adalah bahasa pemrograman yang bertujuan untuk menyelesaikan pemrograman berorientasi objek. *Java* menyediakan banyak ektensi yang dapat mendukung pengembangan aplikasi melalui tampilan *Graphical User Intercase* (GUI), mengembangkan aplikasi *client/server* yang tidak hanya mencakup jaringan lokal namun lebih jaringan yang lebih luas lagi. Program *Java* dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM), yang dapat berjalan pada *platform* yang berbeda-beda. *Java interpreter* dapat dijalankan langsung dari *command prompt*; atau *applet viewer* atau *web browser* (untuk *applet*). Langkah pertama dalam membuat program berbasis *Java* adalah menulis kode program dalam editor teks. Editor teks dapat menyertakan *notepad*, vi, emacs, dan sebagainya. Kode program yang dibuat kemudian disimpan dalam sebuah berkas berekstensi *java*. Setelah membuat dan menyimpan kode program, gunakan *compiler Java* untuk mengkompilasi file yang berisi kode program. File *bytecode* yang dikompilasi menyertakan ekstensi kelas. *Interpreter Java* akan mengubah *bytecode* yang terdapat dalam file menjadi kode mesin pada *platform* dan jenis komputer yang digunakan (Harefa, 2022).

2.9 *Database*

Dalam membuat aplikasi, penulis atau *programmer* menggunakan *database* sebagai dasar untuk mengolah data atau mengelola file. Basis data umumnya merupakan salah satu aspek terpenting dari sistem informasi di mana basis data digunakan sebagai penyimpanan data untuk pemrosesan lebih lanjut. Basis data adalah kumpulan data yang disusun secara sistematis pada komputer menggunakan perangkat lunak untuk menghasilkan informasi. Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah untuk memelihara data atau informasi yang diproses dan untuk membuat informasi tersebut tersedia saat

dibutuhkan. Adapun pengertian *database* yaitu memberikan batasan bahwa satu atau lebih *database* adalah kumpulan informasi atau data yang sistematis yang dapat diperiksa oleh program komputer untuk mengambil informasi dari *database* (Vol et al., 2022). Prinsip dasar-dasar pemrograman *database* yang disebut CRUD, yaitu:

a. *Create* adalah membuat data yang dimasukkan ke dalam sistem *database* oleh sistem atau oleh pengguna.
b. *Read* adalah data yang diinput dari sistem *database* untuk diproses dan ditampilkan di aplikasi sesuai dengan perintah *user*.
c. *Update* adalah data yang disimpan dalam sistem *database* dapat diubah atas perintah *user*.
d. *Delete* adalah menghapus data yang disimpan dalam sistem *database* yang dapat dihapus dengan perintah *user*.

2.10 MySQL

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Siapa pun bebas menggunakan MySQL, tetapi tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial. MySQL sebenarnya berasal dari SQL (*Structured Query Language*), yang telah lama menjadi salah satu konsep utama dalam *database.* SQL adalah konsep manipulasi *database* yang mengkhususkan diri dalam memilih atau menyeleksi dan memasukkan data, sehingga memudahkan untuk melakukan manipulasi data secara otomatis. Keandalan suatu sistem *database* (DBMS) ditunjukkan oleh bagaimana pengoptimal menangani perintah SQL yang dibuat oleh pengguna dan program aplikasi. Sebagai *database server*, MySQL adalah keunggulan dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user,* kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari *Postagre* SQL dan lima kali lebih cepat dibandingkan *Interbase*. Kemampuan yang cukup menakjubkan dari *software* gratis (Larno & Sahrun, 2019). Terdapat tiga jenis perintah dasar SQL yaitu:

1. Data *Definition Language* (DDL) adalah jenis pernyataan SQL yang

berhubungan dengan struktur tabel dan pembuatan *database.* Diantaranya adalah *create* untuk membuat *database* dan tabel, *drop* untuk menghapus *database* dan tabel, *alter* untuk mengubah struktur tabel, dan *rename* untuk perubahan nama tabel.

2. Data *Manipulation Language* (DML) adalah jenis pernyataan SQL yang berhubungan dengan data yang terdapat dalam sebuah tabel. Diantaranya adalah *select* untuk menampilkan data. *Insert* untuk menambah data, *update* untuk mengubah data yang ada, dan *delete* untuk menghapus data.
3. Data *Control Language* (DCL) adalah jenis pernyataan SQL yang terkait dengan manajemen hak akses dan penggunaan *user* yang dapat mengakses *database* maupun tabel diantaranya adalah *grant* dan *revoke*.

2.11*Android*

Android adalah sistem operasi yang digunakan pada perangkat *mobile*, dan saat ini menjadi sangat terkenal dan populer. *Android* adalah *platform* pemrograman yang dibuat *Google* untuk ponsel cerdas dan perangkat seluler lainnya. *Android* dapat berjalan di perangkat yang dibuat oleh produsen *smartphone* yang berbeda. *Android* tidak hanya menyediakan paket pengembangan untuk aplikasi *android,* tetapi juga pasar untuk mendistribusikan aplikasi yang sudah jadi. Dengan potensi penuh yang ditawarkan *android*, dapat dikatakan bahwa *android* secara keseluruhan sedang menciptakan ekosistemnya sendiri.

Android pertama kali dikembangkan oleh sebuah perusahaan kecil bernama *Android* Inc. Di Silicon Valley. Pada tahun 2005, sistem operasi diakuisisi oleh *Google*, menjadikannya "*Open Source*" dan gratis untuk digunakan siapa saja, termasuk menggunakan kode sumber yang digunakan untuk mengembangkan sistem operasi (Herlinah, 2019).

Gambar 2. 4 Logo *Android*

(Sumber:Syafrial dan Ichsan, 2020)

Fitur-fitur *Android* adalah:

1. *Framework* aplikasi, itu dapat didaur ulang dan diganti.
2. *Browser* terintegrasi berbasis *engine Open Source* WebKit ini juga digunakan pada *browser* Iphone dan Nokia S6ov3.
3. Rancangan *handset*. *Platform* ini memenuhi kebutuhan VGA (Video *Graphics* Adapter), pustaka grafis 2D dan 3D yang lebih besar berdasarkan spesifikasi OpenGL ES 1.0 dan tata letak *smartphone* yang tradisional.
4. *Multi-touch*. *Android* memiliki dukungan *multi-touch* bawaan dan tersedia di ponsel yang lebih baru seperti HTC Hero.
5. Dukungan *hardware* tambahan. *Android* mendukung kamera, layar sentuh, GPS (Global *Positioning System*), pengukur kecepatan, magnetometer, akselerasi 2D bit blits (dengan orientasi *hardware*, *scaling*, konversi format piksel) dan akselerasi grafis 3D (Ira Puspita Sari, 2021).

2.11.1 Kelebihan *Android*

Android saat ini menjadi salah satu sistem operasi yang paling populer. Sistem operasi *Android* merupakan sistem operasi yang memiliki banyak keunggulan. Berikut beberapa keunggulan sistem operasi *Android*:

1. Sistem operasi *Open source*

   Kelebihan pertama dari *Android* adalah sebuah sistem operasi yang sifatnya *open source*. Hal ini dikarenakan *Android* merupakan sistem operasi yang

berbasis Linux dan memiliki sistem *open source*, yang memudahkan untuk mengembangkan sistem operasi.

2. Dapat dikustomisasi dan dimodifikasi

   Sistem operasi *Android* dapat dimodifikasi. Tidak hanya developer profesional yang dapat mengembangkan sistem operasi *Android* untuk *smartphone*, tetapi juga para penghobi operasi *Android* ini. Mulai dari kustomisasi ROM, hingga modifikasi *overclock.*

3. Dapat dibeli dengan harga murah

   Dulu harga *smartphone* sangat mahal sehingga hanya orang-orang tertentu saja yang bisa membelinya. Berkat lahirnya sistem operasi *Android*, masyarakat bisa membeli *smartphone* dengan harga yang sangat murah.

4. Dapat dijalankan pada banyak pilihan spesifikasi pada perangkat keras

   Android merupakan sistem operasi yang sangat fleksibel. *Android* dapat disematkan pada perangkat keras dengan spesifikasi apa pun. Dari spesifikasi perangkat keras rendah hingga spesifikasi perangkat keras tinggi.

5. Dukungan aplikasi yang sangat banyak

   Salah satu hal yang membuat *Android* begitu populer di kalangan pengguna adalah mendukung banyak aplikasi.

6. Dapat diaplikasikan pada peralatan elektronik

   Android dapat diterapkan ke berbagai perangkat seperti *smartphone*, tablet, mini PC, jam tangan, dan lain-lain.

7. Dikembangkan oleh Google

   Android dibuat dan dikembangkan oleh salah satu raksasa teknologi dunia, yaitu Google, Inc. Pengembangan sistem operasi *Android* ini sudah terjamin, karena sering dilakukan pembaruan oleh pihak google (Syafrial et al., 2020).

2.11.2 Kekurangan *Android*

Meskipun banyak yang memiliki kelebihan, *Android* juga memiliki

beberapa kelemahan dan kekurangan. Berikut ini adalah beberapa kelemahan dan kekurangan dari sistem operasi *Android*:

1. Proses kerja sistem yang berat dan menghabiskan banyak RAM

   Proses kerja sistem yang cukup berat menyebabkan banyak *memory*, baik RAM maupun ROM yang terpakai. Jika menggunakan *smartphone* dengan jumlah RAM dan ROM kecil, proses kerja akan terhambat.
2. Apabila dijalankan dengan spesifikasi *hardware smartphone* yang rendah, menjadi kurang responsif

   Spesifikasi *smartphone* yang buruk dapat sedikit memperlambat sistem operasi Android dan membuat kurang responsif. Ini mengacu pada jumlah RAM, kapasitas ROM dan kecepatan prosesor yang digunakan di *smartphone.*
3. Iklan yang mengganggu

   Aplikasi *Android* gratis untuk digunakan tetapi aplikasi yang digunakan akan memunculkan iklan yang cukup mengganggu.
4. Penggunaan baterai yang cepat habis

   Baterai *smartphone Android* sangat boros dibandingkan dengan sistem operasi lain karena melalui banyak proses yang memakan energi baterai dengan cepat (Syafrial et al., 2020).

2.12 *Android Studio*

Android Studio merupakan *software* terbaru yang dikeluarkan secara resmi oleh pihak *Android* yang dikhususkan untuk membuat aplikasi *Android*. Untuk bahasa pemrograman yang digunakan sama seperti pada *software* ECLIPSE yaitu XML dan *Java*. *Android Studio* juga mempunyai banyak fitur terbaru yang diberikan untuk memudahkan pembuat *software*, terutama pada tampilan yang lebih menarik dari pada ECLIPSE. Kebanyakan dari pengguna ECLIPSE juga sudah mulai berpindah ke *Android Studio*. Karena diragukan suatu saat tidak mendapatkan dukungan dari *Android* sendiri (Nasution & Kom, 2019).

Android Studio merupakan lingkungan Pengembangan Perangkat Lunak Terpadu – *Integrated Development Environment* (IDE) untuk mengembangkan

aplikasi *Android* berdasarkan IntelliJ IDEA. *Android Studio* bukan hanya editor kode dan alat pengembang IntelliJ yang andal. Beberapa fitur yang disediakan oleh *Android Studio* adalah:

1. Sistem versi berbasis *Gradle* yang fleksibel.
2. *Emulator* yang cepat dan kaya fitur.
3. Lingkungan pengembangan terintegrasi untuk semua perangkat *Android*.
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. Kode Templat dan Integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat dan kerangka kerja pengujian yang luas (Herlinah & Musliadi KH, 2019).

Setelah program dibuat di *Android Studio*, lalu program sudah dapat diuji coba. Diperlukan *emulator Android* untuk mencoba program ini. Atau kita dapat menjalankannya langsung dari *smartphone Android* masing-masing. Jika kita menggunakan *emulatorAndroid*, maka dibutuhkan kapasitas RAM minimal 4GB, dan dianjurkan 8GB.

Android Virtual Device (AVD) adalah *emulator* yang disertakan dalam Android Studio dan disediakan oleh Android *Software Development Kit* (SDK). Selanjutnya, ada beberapa *emulator* pihak ketiga lainnya, misalnya *Genymotion*, *Genymotion* adalah *emulator Android* berbasis *Virtualbox*. Dibandingkan dengan di atas, perangkat *emulator* bisa jadi bahan pertimbangan yaitu menggunakan *handphone* kita sendiri sebagai alat untuk membuat aplikasi *Android*. Hal ini membuat RAM laptop lebih hemat karena tidak melakukan aktivitas yang melelahkan, sehingga memudahkan laptop untuk dijalankan saat *coding*.

Perangkat yang dibutuhkan untuk simulasi ini adalah *handphone* Android dan kabel USB. *Handphone* tersebut nantinya akan terhubung ke laptop melalui kabel USB. Sebelum menjalankan program, terlebih dahulu memastikan bahwa *driver handphone* telah diinstal di *handphone* kita atau belum. Jika sudah maka kita bisa langsung menghubungkan *handphone* ke laptop, jika memang belum kita bisa

download melalui internet (Siregar, 2021).

2.13 *Flowchart*

Diagram alur atau biasa disebut dengan *flowchart* merupakan notasi yang menggambarkan suatu aliran atau fase proses. Dalam hal ini, alur proses disimbolkan (dinotasikan) dengan bentuk-bentuk simbol gambar. Notasi ini biasanya mengacu pada fase-fase seperti permulaan proses, pemrosesan berulang, proses pengambilan keputusan, masukan, keluaran dan lain sebagainya.

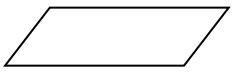
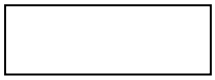
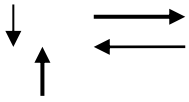
Flowchart adalah representasi secara diagram yang menggambarkan serangkaian operasi yang terkait dengan pemecahan masalah. *Flowchart* biasanya dibuat lebih awal sebelum pemrograman. *Flowchart* juga merupakan fungsi komunikasi antara programmer dan pebisnis. Selain itu *flowchart* juga membantu memahami logika program, terutama logika yang panjang dan kompleks. Dengan adanya *flowchart* program dibuat dengan mudah (Suyanto, 2018).

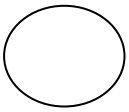
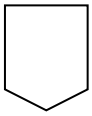
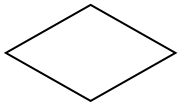
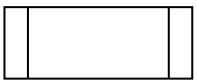
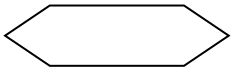
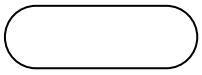
Tujuan utama pembuatan *flowchart* adalah untuk:

1. Mendeskripsikan tahap pemecahan masalah.
2. Secara sederhana, terurai, rapi dan jelas.
3. Gunakan simbol standar.

Tabel 2. 4 Simbol-simbol *Flowchart*

(Sumber: Jogiyanto, 2007)

No	Simbol	Nama	Keterangan
1		Input/Output	Digunakan untuk mewakili data input/output.
2		Proses	Digunakan untuk mewakili suatu proses.
3		Garis Alir	Digunakan untuk menunjukkan arus dari proses.

4		Penghubung	Digunakan untuk menunjukkan sambungan dari bagian alir yang terputus di halaman yang masih sama.
5		Penghubung	Digunakan untuk menunjukkan sambungan dari bagian alir yang terputus di halaman lainnya.
6		Keputusan	Digunakan untukpenyeleksian kondisi di dalam program.
7		Proses Terdefenisi	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
8		Persiapan	Digunakan untuk memberi nilai awal suatu besaran.
9		Titik Terminal	Digunakan untuk menunjukkan awal dan akhir dari suatu proses.

2.14 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah bahasa pemodelan pengembangan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek (Sudaria et al., 2021). UML adalah "bahasa" standar industri untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. Diagram UML yang umum digunakan adalah: *use case diagram, activity diagram, sequence diagram, class diagram* (Amaliyah et al., 2021).

Adapun tujuan UML (*Unified Modelling Language*) diantaranya adalah:

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek terbaik yang terdapat dalam pemodelan (Herianto, 2018).

Beberapa hal terpenting dalam pengembangan UML, UML dijelaskan

dalam diagram, antara lain:

1. *Use Case* Diagram

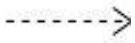
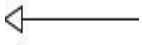
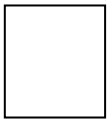
Use Case diagram merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya (Larno & Sahrun, 2019).

Adapun pengertian dari simbol-simbol yang terdapat pada *use case* diagram adalah sebagai berikut :

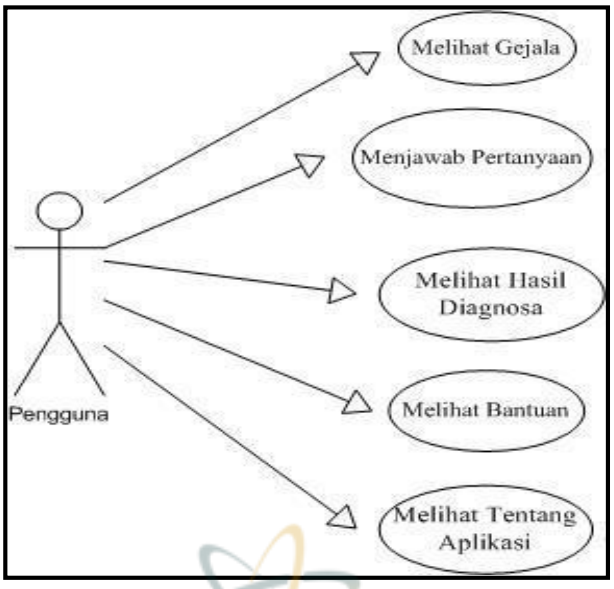
Tabel 2. 5 Simbol *Use Case* Diagram

(Sumber: Ismail et al., 2021)

Simbol	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent).*
	*Generalization*	Hubungan dimana objek anak (*descendent)* berbagi perilaku
Simbol	**Nama**	**Keterangan**
		ada di atasnya objek induk (*ancestor).*

	*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit.*
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Contoh dari penggunaan *use case* diagram adalah sebagai berikut :



Gambar 2. 5 Contoh *Use Case* Diagram

(Sumber: Pallangan et al., 2017)

2. *Activity* Diagram

Activity diagram atau diagram aktivitas merupakan suatu yang menggambarkan konsep aliran data dan proses bisnis dan urutan aktivitas dalam sistem. Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Pada tahapan pemodelan bisnis ini, diagram aktivitas bisa digunakan untuk menunjukkan aliran kerja bisnis (*bussiness flow*). Bisa juga digunakan untuk menggambarkan aliran kejadian (*flow of events*) dalam *use case*. *Activity* diagram digunakan untuk memodelkan perilaku didalam sebuah bisnis. *Activity* diagram memiliki notasi untuk memodelkan aktivitas yang berlangsung secara bersamaan, pararel, dan juga proses mengambil keputusan yang kompleks (Kantor & Sekejati, 2021).

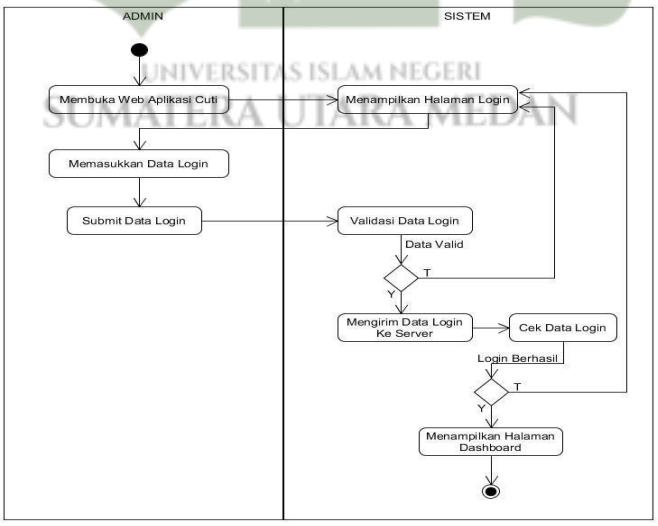
Berikut merupakan simbol yang digunakan dalam membangun *activity* diagram adalah sebagai berikut :

Tabel 2. 6 Simbol *Activity* Diagram

(Sumber: Ismail et al., 2021)

Simbol	Nama	Keterangan
	*Actifity*	Menunjukkan bagaimana setiap kelas antarmuka berinteraksi.
	*Action*	Keadaan sistem yang mencerminkan eksekusi tindakan.
	*Initial Node*	Bagaimana objek terbentuk atau dimulai.
	*Actifity Final Node*	Bagaimana objek dibuat dan dihancurkan.
	*Fork Node*	Aliran bergabung menjadi beberapa aliran pada titik waktu tertentu.

Di bawah ini merupakan contoh dari *activity* diagram :



Gambar 2. 6 Contoh *Activity* Diagram

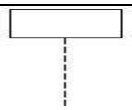
(Sumber: Khumaidi & Muljadi, 2020)

3. *Sequence* Diagram

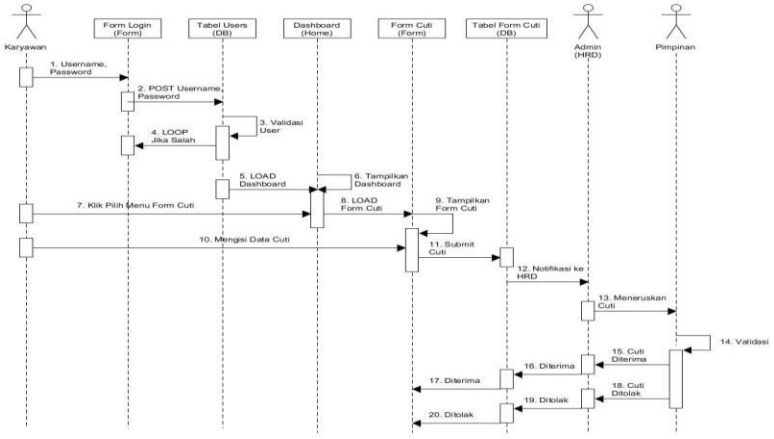
Sequence diagram digunakan untuk menggambarkan interaksi antara objek di dalam dan di sekitar sistem dalam hal pesan yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek terkait) (T. Bayu Kurniawan, 2020).

Tabel 2. 7 Simbol-simbol Pada *Sequence* Diagram

(Sumber: Ismail et al., 2021)

Simbol	Nama	Keterangan
	*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Di bawah ini merupakan contoh dari *sequence* diagram:



Gambar 2. 7 Contoh *Sequence* Diagram

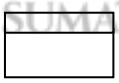
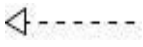
(Sumber: Khumaidi & Muljadi, 2020)

4. *Class* Diagram

Class diagram adalah diagram yang sering ditemui pada pemodelan berbasis UML. *Class* diagram digunakan untuk menunjukkan interaksi antar *class* di dalam sistem (Mur et al., 2019). *Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan (Dan et al., 2022). Simbol-simbol yang digunakan dalam *Class* diagram yaitu:

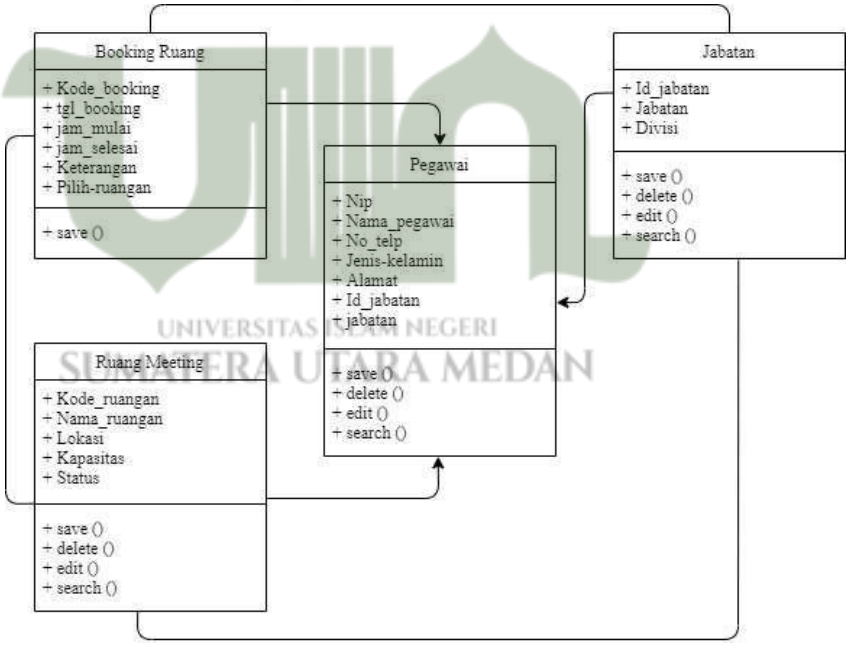
Tabel 2. 8 Simbol *Class* Diagram

(Sumber: Dan et al., 2022)

Simbol	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (*descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor)*.
	*Nory Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Realization*	Operasiyang benar-benardilakukanoleh suatuobjek.
	Dependency	Hubungan dimana perubahan

Simbol	Nama	Keterangan
		yang terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
———	*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Dibawah ini merupakan contoh dari *class* diagram:



Gambar 2. 8 Contoh *Class* Diagram

(Sumber: Vol et al., 2022)

2.15 Penelitian Terdahulu

Ada beberapa penelitian terdahulu yang terkait dengan penelitian yang dilakukan oleh penulis untuk dijadikan sebagai bahan referensi, diantaranya terdapat di dalam tabel berikut:

Tabel 2. 9 Penelitian Terdahulu

No	Nama Peneliti/ Identitas Jurnal	Judul	Kelebihan	Kekurangan
1.	Sarma Tampubolon RESOLUSI, Vol. 2, No. 3, Januari 2022; ISSN: 2745-7966	Sistem Pakar Diagnosa Penyakit Skistosomiasis Menggunakan Kombinasi *Forward Chaining* Dan *Fuzzy Logic Takagi Sugeno Kang*	Prosedur diagnosa penyakit Skistosomiasis pada penelitian ini dilakukan berdasarkan ketentuan dari gejala yang diadopsi pada tubuh penderita, penerapan metode *fuzzy logic* dapat menghasilkan keluaran diagnosa penyakit Tonsilitis secara akurat.	Penerapan metode *fuzzy logic*dapat menghasilkan keluaran diagnosis penyakit Tonsilitis secara akurat.
2.	Hendrikus Daely dan Dito Putro Utomo KOMIK, Vol. 4, No. 1, Oktober	Sistem Pakar Diagnosa Hepatomegali Menerapkan	Sistem pakar pada penelitian ini bekerja untuk menemukan solusi diagnosa	Penyelesaian masalah ketidakpastian diagnosa penyakit

	2020; ISSN: 2597-4645	Metode *Fuzzy Logic Sugeno*	yang akurat dari permasalahan ketidakpastiannya diagnosa awal terhadap penyakit Hepatomegali.	Hepatomegali yang dialami oleh masyarakat awam sehingga harus membutuhkan adanya bantuan seorang ahli yang disebut pakar diagnosa penyakit Hepatomegali.
3.	Humaidillah Kurniadi Wardana, Immatul Ummah dan Lina Arifah Fitriyah Jurnal Fisika Flux, Vol. 19, No. 2, Juni 2022; ISSN: 2541-1713	Sistem Pakar *Fuzzy* dengan Metode *Sugeno* Untuk Diagnosa Penyakit Diabetes Mellitus	Sistem pakar ini membantu membuat suatu keputusan dengan cepat dalam mendiagnosa penyakit diabetes mellitus dan memiliki tingkat keakuratan sistem ini sebesar 68%.	Penelitian ini hanya mendapatkan variabel inputan berupa glukosa sewaktu dari data pasien.
4.	Ilka Zufria, Heri Santoso dan Darsih J-SAKTI, Vol. 5, No. 1, Maret	Sistem Pakar Menggunakan Metode *Backward Chaining* Untuk	Sistem pakar ini dapat digunakan untuk membantu para petani kedelai dalam	Sistem pakar ini dapat menampilkan hasil diagnosis yang disertai

	2021; ISSN: 2548-9771	Mengantisipasi Permasalahan Tanaman Kacang Kedelai Berbasis Web	upaya mengetahui hama dan penyakit pada tanaman serta bagaimana penanggulangan dan pencegahan serangan hama dan penyakit kedelai.	dengan solusi pencegahan dan penanggulangan dari gejala-gejala yang diderita.
5.	Nico Alvio Maiyedra JURSIMA, Vol. 6, No. 2, Desember 2018; ISSN: 2338-1523	Perancangan Sistem Pakar Mendiagnosa Penyakit Kulit Pada Anak dengan Menggunakan Metode *Backward Chaining*	Sistem pakar ini dapat memberikan informasi tentang gejala penyakit kulit pada anak kepada pasien dan dalam penelitian ini diperoleh 8 jenis gangguan, 27 gejala dan 28 aturan (*rule*).	Dengan pembangunan aplikasi sistem pakar untuk penyakit kulit pada anak akan membantu pasien dalam mendiagnosis penyakit kulit pada anak.