

BAB II

KAJIAN PUSTAKA

2.1 Teori-teori terkait bahasan penelitian (*Relevant Theories*)

2.1.1 Konsep Dasar Data dan Informasi

Istilah data berasal dari bahasa Latin "datum" yang bermakna "sesuatu yang diberikan", sementara kata data sendiri merupakan bentuk jamaknya. Menurut (Janssen et al., 2020), data merupakan aset strategis yang harus dikelola melalui tata kelola data (data governance) agar dapat menghasilkan informasi yang berkualitas, mendukung pengambilan keputusan, serta memberikan nilai bagi organisasi. Dalam era Big Data, data tidak lagi dipandang sebagai sekumpulan fakta mentah semata, tetapi sebagai sumber daya yang harus dikelola sepanjang siklus hidupnya, mulai dari pengumpulan, penyimpanan, pengolahan, hingga pemanfaatannya secara bertanggung jawab. Oleh karena itu, pengelolaan data yang efektif menjadi faktor penting dalam mendukung terciptanya organisasi yang adaptif, inovatif, dan mampu memanfaatkan teknologi informasi secara optimal. Bentuknya bisa berupa satu kesatuan (*datum*) atau kumpulan dari berbagai datum yang membentuk suatu kelompok data. Data juga dapat dipahami sebagai sekumpulan fakta yang diproses untuk menghasilkan informasi sebagai keluaran. Supaya data dapat digunakan secara maksimal, data perlu memiliki kredibilitas (*reliable*), ketepatan, kecepatan penyajian, dan kesesuaian dengan kebutuhan. Dalam pengambilan keputusan, data memiliki fungsi vital sebagai fondasi untuk menetapkan strategi dan kebijakan (Prehanto, 2020).

Informasi merupakan produk dari pengolahan data menggunakan metode tertentu sehingga memiliki arti dan kegunaan bagi yang menerimanya. McLeod dalam Yakub (2012) mendefinisikan informasi sebagai data yang sudah diproses hingga menjadi lebih bermanfaat dan bermakna bagi pengguna dalam menunjang kegiatan sistem informasi. Jadi, informasi bukan hanya sekedar himpunan data, melainkan telah melalui

tahapan yang membuatnya relevan dan aplikatif. Informasi juga dapat didefinisikan sebagai data yang telah mengalami pemrosesan, interpretasi, atau klasifikasi sehingga bisa dijadikan landasan dalam pengambilan keputusan (Ikram & Muthmainnah, 2019). Informasi yang bermutu tinggi dapat membantu pengguna dalam memahami situasi yang berlangsung dan menentukan langkah yang sesuai. Sumber primer informasi adalah data yang merepresentasikan peristiwa aktual pada periode tertentu. Data tersebut harus melalui rangkaian yang disebut siklus pengolahan data (*data processing life cycle*), mencakup tahap pengumpulan, pengolahan, penyimpanan, dan penyajian data, agar menghasilkan informasi yang tepat dan berguna.

Menurut (Ati, 2014), pengetahuan adalah produk dari pemahaman manusia tentang realitas yang terbentuk lewat proses pengolahan dan interpretasi informasi yang didapat. Pengetahuan memiliki sifat dinamis dan dapat berubah seiring bertambahnya informasi serta pengalaman. Karena itu, meski menerima informasi yang identik, pengetahuan yang terbentuk pada tiap individu bisa berbeda, bergantung pada cara mereka memahami dan menginterpretasikannya.

Data memiliki fungsi yang sangat krusial dalam proses pengambilan keputusan karena menyediakan informasi yang objektif dan berdasarkan fakta. Data menjadi fondasi utama bagi manajemen atau pembuat keputusan untuk memahami kondisi riil yang dihadapi dan menentukan tindakan yang paling tepat (Iraza et al., 2023). Fungsi data dalam pengambilan keputusan meliputi hal-hal berikut:

- 1) Data menyediakan basis yang kokoh bagi pembuat keputusan sehingga keputusan tidak semata-mata bergantung pada intuisi atau dugaan, namun pada fakta yang akuntabel.
- 2) Lewat analisis data, organisasi mampu mengidentifikasi masalah yang ada serta menemukan peluang-peluang baru yang sebelumnya tidak terdeteksi.

- 3) Data memfasilitasi organisasi dalam melakukan proyeksi dan peramalan terhadap situasi mendatang, sehingga keputusan yang diambil lebih proaktif dan bersifat strategis.
- 4) Dengan menggunakan data, organisasi dapat mengevaluasi alur kerja, menekan pemborosan, serta meningkatkan efisiensi dan efektivitas dalam operasional.
- 5) Data dipakai untuk mengukur dan mengevaluasi performa organisasi, baik dari aspek finansial, operasional, maupun capaian target yang telah ditetapkan.
- 6) Keputusan yang ditopang oleh data yang akurat dan relevan mampu menekan risiko kesalahan serta dampak negatif yang mungkin muncul dari keputusan yang kurang tepat.
- 7) Pemanfaatan data secara optimal membantu organisasi dalam merancang strategi yang lebih baik, sehingga dapat berkompetisi secara efektif dengan kompetitor.

2.1.2 Konsep Dasar Sistem Pendukung Keputusan

Secara umum Sistem adalah suatu kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian dalam suatu tujuan utama. Menurut Gordon B. Davis dalam bukunya menyatakan, Sistem bisa berupa abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling bergantung. Sedangkan Sistem yang bersifat fisis adalah serangkaian unsur yang bekerjasama untuk mencapai suatu tujuan (Arjuna et al., 2021).

Sistem Pendukung Keputusan (SPK) merupakan sistem berbasis komputer yang mendukung proses pengambilan keputusan dengan mengolah data dan model untuk menghasilkan alternatif solusi dari suatu permasalahan, baik yang bersifat terstruktur maupun tidak terstruktur (Sopandy et al., 2025). Sistem Pendukung Keputusan (SPK) juga berfungsi untuk memberikan dukungan dalam proses pengambilan keputusan di suatu organisasi, instansi ataupun Perusahaan (Cahyono & Wibisono, 2024).

Terdapat beberapa karakteristik dan kemampuan yang menjadi ciri khas dari Sistem Pendukung Keputusan. Karakteristik tersebut membedakan SPK dengan sistem informasi lainnya dan menunjukkan fungsi serta perannya dalam mendukung proses pengambilan keputusan di berbagai tingkatan organisasi. Selain karakteristik, SPK juga harus memiliki sejumlah kemampuan tertentu agar dapat berfungsi secara optimal dalam membantu pengguna menyelesaikan permasalahan yang kompleks dan semi-terstruktur :

1. Karakteristik dan Kemampuan Sistem Pendukung Keputusan.

Sistem pendukung keputusan memiliki beberapa karakteristik antara lain (Indahini et al., 2023) :

- 1) Dapat mendukung berbagai aplikasi terkait.
- 2) Dapat digunakan berulang kali dan bersifat permanen.
- 3) Memiliki dua bagian utama, yaitu model dan data.
- 4) Dapat melakukan analisis "bagaimana jika" dan analisis pencarian tujuan.
- 5) Menggunakan berbagai jenis model kuantitatif.

Selain itu sistem pendukung keputusan harus memiliki kemampuan seperti (Indahini et al., 2023) :

- 1) Mampu menangani masalah secara semi-terstruktur saat mengambil keputusan manajemen.
- 2) Mampu mengambil keputusan baik di tingkat kelompok maupun individu.
- 3) Mampu mengambil keputusan yang saling terkait dan berurutan.
- 4) Memiliki kemampuan beradaptasi kapan saja dan fleksibel.
- 5) Mudah berinteraksi dengan sistem.
- 6) Mudah dikembangkan dan digunakan oleh pengguna.

2. Komponen Sistem Pendukung Keputusan.

Sistem pendukung keputusan memiliki komponen sebagai berikut (Ramadhani, 2020) :

1) Manajemen Data.

Ini mencakup basis data yang menyimpan informasi terkait berbagai situasi, dan dikelola oleh perangkat lunak Sistem Manajemen Basis Data (DBMS).

2) Manajemen Model.

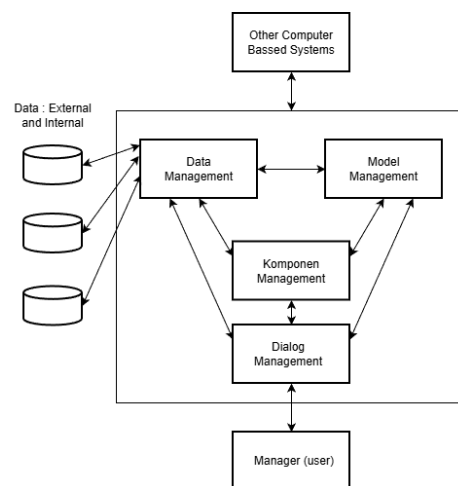
Ini melibatkan penggunaan berbagai model kualitatif seperti model statistik, keuangan, dan ilmu manajemen, yang membantu memberikan kemampuan analitis pada sistem dan manajemen perangkat lunak.

3) Komunikasi.

Dalam subsistem pengguna, pengguna dapat berkomunikasi dan memberikan perintah kepada sistem pendukung keputusan.

4) Manajemen Pengetahuan.

Ini adalah subsistem opsional yang mendukung subsistem lain dan bertindak sebagai komponen mandiri.



Gambar 2.1 Komponen SPK, Sumber: (Ramadhani, 2020)

2.1.3 Kerangka atau Metodologi SPK

Kerangka atau metodologi Sistem Pendukung Keputusan (SPK) merupakan rangkaian tahapan sistematis yang digunakan untuk membangun dan mengimplementasikan sistem yang mampu membantu pengambil

keputusan dalam memilih alternatif terbaik berdasarkan data dan kriteria tertentu. Metodologi SPK dirancang agar proses pengambilan keputusan dapat dilakukan secara terstruktur, objektif, dan rasional, terutama dalam menghadapi permasalahan yang bersifat semi-terstruktur maupun tidak terstruktur (Turban et al., 2020). Adapun tahapan sebagai berikut :

1. Tahapan Pengambilan Keputusan.

Ada beberapa tahapan dalam pengambilan keputusan antara lain (Maulana et al., 2024):

1) Tahap Pemahaman

Tahap ini merupakan tahap awal untuk penelusuran terhadap data yang akan diproses dalam rangka mengidentifikasi masalah.

2) Tahap Perancangan.

Merupakan proses pengembangan dan pencarian sebuah solusi yang bisa diambil sebagai representasi kejadian nyata lalu penyembunyian, sehingga diperlukan proses pemeriksaan untuk mengetahui seberapa akurat model dalam masalah yang diteliti.

3) Tahap Pemilihan.

Memilih dari berbagai alternatif solusi yang telah ditampilkan pada tahap sebelumnya, hal ini bertujuan agar dapat menentukan kriteria berdasarkan tujuan yang ingin dicapai.

4) Tahap Implementasi.

Perencanaan perencanaan yang telah dibuat serta pelaksanaan tindakan alternatif yang telah dibuat sebelumnya.

5) Tahap Evaluasi

Implementasi Adalah tahapan keputusan harus selalu dimonitor. meangevaluasi apakah tepat dalam implementasi nya

2.1.4 Calon Petani Calon Lokasi (CPCL)

2.1.4.1 Pengertian Calon Petani Calon Lokasi (CPCL)

Calon Petani Calon Lokasi (CPCL) adalah langkah pertama dalam pelaksanaan program bantuan atau pengembangan pertanian

yang dilakukan oleh pemerintah daerah melalui Dinas Ketahanan Pangan dan Pertanian. CPCL berfungsi sebagai dasar untuk menentukan kelompok petani mana yang akan menerima bantuan dan lahan mana yang sesuai untuk kegiatan pertanian sesuai dengan pedoman teknis yang ada.

Tujuan CPCL adalah untuk memastikan bahwa bantuan pemerintah sampai kepada orang dan tempat yang tepat. Proses ini melibatkan pengumpulan data, verifikasi, dan validasi petani dan lahan potensial oleh petugas penyuluh pertanian dan instansi terkait, sehingga program berjalan efektif dan berkelanjutan. Dalam pelaksanaannya, CPCL mencakup beberapa kriteria penting, seperti status dan tingkat aktivitas petani atau kelompok petani, kepemilikan atau penguasaan lahan, kesesuaian lahan untuk tanaman yang direncanakan, dan kesiapan petani untuk menerima dan mengelola bantuan pertanian.

Dengan adanya CPCL, Dinas Pertanian dapat mengurangi kesalahan dalam penyaluran bantuan dan meningkatkan produktivitas serta kesejahteraan petani. Dinas Ketahanan Pangan dan Pertanian Kabupaten Kuningan, Calon Petani Calon Lokasi (CPCL) merupakan konsep yang digunakan dalam perencanaan dan pelaksanaan program pembangunan pertanian untuk memastikan ketepatan sasaran penerima program dan kesesuaian lokasi kegiatan. Beberapa ahli dan literatur pertanian menjelaskan konsep CPCL atau prinsip yang sejalan dengannya sebagai berikut:

1) Menurut Kementrian Pertanian 2025 (Pertanian, 2025)

CPCL merupakan proses pendataan, seleksi, dan penetapan calon petani atau kelompok tani beserta calon lokasi lahan yang layak sebagai penerima program bantuan pertanian berdasarkan kriteria teknis, administratif, dan kesesuaian agroekosistem.

2) Menurut (Yusianto et al., 2021)

Dalam perencanaan pembangunan pertanian, penentuan sasaran program harus dilakukan melalui pendekatan sistematis yang mempertimbangkan karakteristik petani, kondisi lahan, serta kesesuaian komoditas, sehingga keputusan yang diambil dapat bersifat objektif dan tepat sasaran. Konsep ini sejalan dengan prinsip CPCL sebagai dasar pengambilan keputusan program pertanian.

3) Menurut (Everest et al., 2021)

Pemilihan lokasi pertanian harus mempertimbangkan berbagai kriteria seperti kesesuaian lahan, faktor lingkungan, dan aksesibilitas. Penggunaan pendekatan multi-kriteria sangat penting untuk menentukan lokasi yang paling layak bagi kegiatan pertanian, yang secara konseptual sejalan dengan penetapan calon lokasi dalam CPCL.

Berdasarkan berbagai pendapat tersebut, dapat disimpulkan bahwa *Calon Petani Calon Lokasi* (CPCL) merupakan suatu mekanisme perencanaan dalam pembangunan pertanian yang bertujuan untuk menentukan petani dan lokasi yang paling layak dan diprioritaskan melalui proses pendataan, seleksi, dan verifikasi berdasarkan kriteria teknis, sosial ekonomi, dan kesesuaian lahan. Penerapan CPCL diharapkan mampu meningkatkan ketepatan sasaran program pertanian, efisiensi penggunaan anggaran, serta keberlanjutan pembangunan pertanian, termasuk di Kabupaten Kuningan.

2.1.4.2 Dasar Hukum

- 1) Undang-Undang Nomor 22 Tahun 2019 tentang Sistem Budi Daya Pertanian Berkelanjutan
- 2) Undang-Undang Nomor 41 Tahun 2009 tentang Perlindungan Lahan Pertanian Pangan Berkelanjutan

- 3) Undang-Undang Nomor 16 Tahun 2006 tentang Sistem Penyuluhan Pertanian, Perikanan dan Kehutanan
- 4) Peraturan Presiden Republik Indonesia Nomor 192 Tahun 2024 Tentang Kementerian Pertanian
- 5) Peraturan Menteri Pertanian Republik Indonesia Nomor 29 Tahun 2025 Tentang Pedoman Umum Bantuan Pemerintah Lingkup Kementerian Pertanian
- 6) 6b) Peraturan Gubernur Jawa Barat Nomor 20 Tahun 2023 Tentang Rencana Aksi Perlindungan Dan Pemberdayaan Petani Daerah Provinsi Jawa Barat Tahun 2022-2026
- 7) Peraturan Daerah Kabupaten Kuningan Nomor 7 Tahun 2015 Tentang Perlindungan Lahan Pertanian Pangan Berkelanjutan
- 8) Peraturan Bupati Kuningan Nomor 169 Tahun 2021 Tentang Kedudukan, Susunan Organisasi, Tugas Pokok, Fungsi Dan Uraian Tugas, Serta Tata Kerja Dinas Ketahanan Pangan Dan Pertanian Kabupaten Kuningan

2.1.4.3 Peran Pemerintah Kabupaten Kuningan

Pemerintah Daerah Kabupaten Kuningan telah menyelenggarakan pelaksanaan Calon Petani Potensial dan Calon Lokasi Potensial (CPCL) sebagai bagian dari perencanaan dan pelaksanaan program pengembangan pertanian daerah yang di atur melalui peraturan daerah Peraturan Bupati Kuningan Nomor 169 Tahun 2021; Peraturan Daerah Kabupaten Kuningan Nomor 7 Tahun 2015. CPCL berfungsi sebagai dasar untuk menentukan penerima bantuan pertanian dan mengidentifikasi lokasi kegiatan untuk memastikan program tersebut tepat sasaran dan sesuai dengan kondisi setempat. Pelaksanaan CPCL di Kabupaten Kuningan mengikuti kebijakan dan pedoman teknis dari Kementerian Pertanian Republik Indonesia, yang kemudian dilaksanakan

oleh Dinas Pertanian Kabupaten Kuningan sesuai dengan kebutuhan dan karakteristik daerah.

Selama pelaksanaannya, CPCL disiapkan melalui proses pengumpulan data, verifikasi, dan validasi petani potensial dan lokasi oleh petugas penyuluh pertanian dan satuan pemerintah daerah terkait. Untuk memastikan konsistensi dengan program pembangunan daerah, Dinas Pertanian Kabupaten Kuningan berkolaborasi dengan Dinas Perencanaan, Penelitian, dan Pengembangan Daerah (*Bappeda*) Kabupaten Kuningan dalam perencanaan, pengendalian, dan evaluasi kegiatan pertanian berdasarkan CPCL. Melalui mekanisme ini, CPCL berfungsi sebagai alat perencanaan yang mendukung efektivitas program pertanian, meningkatkan produktivitas, dan memperbaiki kesejahteraan petani di Kabupaten Kuningan.

2.1.4.4 Peran Tim Calon Petani Calon Lokasi (CPCL)

Tim Calon Petani Calon Lokasi (CPCL) Kabupaten Kuningan adalah sebuah tim yang bertugas menyusun, memverifikasi, dan menentukan daftar calon petani serta calon lokasi yang akan menjadi sasaran dari program pembangunan pertanian di kabupaten tersebut Peraturan Bupati Kuningan Nomor 169 Tahun 2021. Tim CPCL sangat penting karena membantu memastikan program pertanian berjalan tepat sasaran, efektif, dan sesuai dengan rencana Pembangunan daerah.

Dalam melakukan tugasnya, Tim CPCL dibantu oleh Dinas Pertanian Kabupaten Kuningan serta bekerja sama dengan Badan Perencanaan Pembangunan, Penelitian, dan Pengembangan Daerah (*Bappeda*) Kabupaten Kuningan, para penyuluh pertanian, dan instansi terkait lainnya. Tujuan utamanya adalah menyelaraskan data calon petani dan calon lokasi dengan program pertanian daerah sesuai dengan arahan

dan kebijakan dari Kementerian Pertanian Republik Indonesia.

Tugas utama Tim CPCL Kabupaten Kuningan mencakup:

1. Koordinasi dalam perencanaan CPCL, yang meliputi:
 - 1) Menyampaikan informasi kepada perangkat daerah dan penyuluh pertanian untuk mengidentifikasi dan mengumpulkan data calon petani dan calon lokasi yang layak sebagai sasaran program pertanian;
 - 2) Membuat jadwal dan agenda untuk pendataan, verifikasi, serta validasi CPCL;
 - 3) Mengumpulkan informasi terkini tentang calon petani dan calon lokasi sesuai dengan kondisi di lapangan;
 - 4) Mengumpulkan dan menyusun daftar prioritas CPCL sesuai dengan jenis program pertanian yang akan dilaksanakan;
 - 5) Menyelaraskan program pembangunan pertanian daerah dengan data CPCL yang telah disusun;
 - 6) Melakukan verifikasi dan validasi terhadap usulan calon petani dan calon lokasi;
 - 7) Menyusun dan memberikan laporan awal daftar CPCL sebagai dasar untuk menentukan penerima program pertanian;
 - 8) Menetapkan CPCL sebagai dasar pelaksanaan kegiatan dan pemberian bantuan pertanian.
2. Koordinasi dalam pelaksanaan CPCL, yaitu membantu menjalankan program pertanian sesuai dengan CPCL yang telah ditetapkan, termasuk memberikan bantuan kepada petani atau kelompok tani yang menerima program.
3. Koordinasi dengan Pemerintah Daerah, khususnya dalam perencanaan program, penyelesaian pelaksanaan, pengawasan, serta pelaporan kegiatan pertanian yang berbasis CPCL.

4. Evaluasi dan pelaporan CPCL, yang mencakup:
 - 1) Mengevaluasi pelaksanaan program pertanian berdasarkan kesesuaian antara CPCL dan keadaan di lapangan;
 - 2) Melaporkan ke pimpinan daerah secara rutin atau saat diperlukan, yang mencakup:
 - a) Hasil pelaksanaan program pertanian berdasarkan CPCL;
 - b) Capaian kinerja dari pelaksanaan CPCL;
 - c) Masalah yang ditemukan di lapangan serta langkah-langkah penyelesaiannya; dan
 - d) Rencana tindak lanjut untuk meningkatkan pelaksanaan CPCL di masa depan.

2.1.4.5 Mekanisme Perencanaan Program Calon Petani Calon

Lokasi (CPCL)

Dalam peraturan penyelenggaraan kegiatan CPCL di Kabupaten Kuningan terdapat mekanisme perencanaan program CPCL sebagai berikut :

- 1) Tim CPCL membuat daftar usulan CPCL yang berpotensi menjadi sasaran program pertanian, berdasarkan saran dari penyuluh pertanian, kelompok tani, dan pihak daerah terkait.
- 2) Usulan CPCL diperiksa dan dioperasikan oleh Tim CPCL agar sesuai dengan aturan teknis dan kondisi di lapangan terkait calon petani dan lokasi yang diusulkan.
- 3) Penentuan prioritas CPCL dilakukan dengan memperhatikan prioritas pembangunan pertanian di daerah, manfaat dan dampak dari program, kesetaraan dalam pemberian bantuan, serta keinginan untuk kegiatan pertanian.
- 4) Usulan CPCL juga bisa datang dari inisiatif kelompok tani atau masyarakat, yang kemudian dikirim ke Tim CPCL untuk

diperiksa dan pelaksanaannya sesuai dengan tahapan yang sudah ditentukan.

2.1.5 Fuzzy Logic

Logika *fuzzy* pertama kali diperkenalkan pada tahun 1965 oleh Prof. Lotfi A. Zadeh, seorang peneliti di bidang ilmu komputer dari *University of California, Berkeley*. Prof. Zadeh berpendapat bahwa logika biner yang hanya mengenal nilai benar dan salah belum sepenuhnya mampu merepresentasikan cara berpikir manusia. Oleh karena itu, dikembangkanlah logika *fuzzy* yang dapat menggambarkan berbagai kondisi secara lebih fleksibel serta mendekati pola penalaran manusia (Rumfot, Lenussa, et al., 2024).

Perbedaan utama antara logika tegas dan logika *fuzzy* terletak pada konsep keanggotaan suatu elemen dalam himpunan. Pada logika tegas, suatu elemen hanya memiliki dua kemungkinan nilai, yaitu bernilai 1 jika termasuk dalam himpunan (benar) dan bernilai 0 jika tidak termasuk dalam himpunan (salah). Sementara itu, pada logika *fuzzy*, derajat keanggotaan suatu elemen dinyatakan dalam rentang nilai antara 0 hingga 1. Salah satu alasan penggunaan logika *fuzzy* adalah karena konsepnya mudah dipahami, mengingat pendekatannya menggunakan bahasa alami yang sesuai dengan cara manusia dalam menilai suatu kondisi (Rumfot, Lenussa, et al., 2024).

2.1.5.1 Konsep Dasar Himpunan Logika Fuzzy

Menurut (Efriska et al., 2024). Jika X adalah sebuah koleksi obyek-obyek yang dinotasikan dengan x , maka himpunan *Fuzzy A* dalam X adalah sebuah himpunan pasangan berurutan $\tilde{A} = \{x, \mu_{\tilde{A}}(x) | x \in X\}$. Notasi $\mu_{\tilde{A}}(x)$ disebut fungsi keanggotaan atau derajat keanggotaan x dalam A yang memetakan X ke ruang keanggotaan M yang terletak pada rentang $[0, 1]$, bila M hanya memuat dua titik 0 dan 1

2.1.5.2 Penalaran Metode Sugeno

Menurut (Rumfot, Lesnussa, et al., 2024). Penalaran dengan metode *Fuzzy Sugeno* pada prinsipnya hampir sama dengan metode *Fuzzy Mamdani*. Perbedaan utama terletak pada bentuk output (konsekuen) sistem. Pada metode *Sugeno*, output tidak berupa himpunan *fuzzy*, melainkan berupa nilai tegas (*crisp*) yang dinyatakan dalam bentuk konstanta atau persamaan linear.

Metode *Takagi-Sugeno-Kang* (TSK) ini diperkenalkan oleh *Michio Sugeno*, yang mengusulkan penggunaan *singleton* sebagai fungsi keanggotaan pada bagian konsekuen. *Singleton* merupakan himpunan *fuzzy* yang memiliki nilai keanggotaan tertentu pada satu titik dan bernilai nol di luar titik tersebut. Berdasarkan bentuk konsekuennya, metode *Fuzzy Sugeno* dibedakan menjadi dua model, yaitu sebagai berikut (Efriska et al., 2024).

1) Definisi Variabel dan Domains (*Crisp*)

Dalam penelitian berbasis *Sistem Inferensi Fuzzy* (FIS), setiap variabel input harus didefinisikan secara eksplisit bersama dengan domain nilai yang digunakan sebagai dasar fuzzifikasi. Menurut (Rifai & Fitriyadi, 2023), dalam aplikasi *Fuzzy Sugeno*, variabel input diberi skala nilai numerik yang sesuai dengan karakteristik data dan domain masalah, sehingga fungsi keanggotaan *fuzzy* dapat dibangun secara tepat berdasarkan domain tersebut.

Pemodelan variabel yang tepat mencakup dua komponen utama:

- a) Domain nilai rentang atau interval nilai numerik variabel, dan
- b) Fuzifikasi himpunan membagi domain menjadi sejumlah himpunan *fuzzy* linguistik, seperti rendah, sedang, dan tinggi.

2) Model *Fuzzy Sugeno* Orde-Nol

Model *Fuzzy Sugeno* Orde-Nol menggunakan konsekuen berupa konstanta. Secara umum, model ini dirumuskan sebagai berikut:

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = k$

dengan keterangan:

- A_i adalah himpunan fuzzy ke- i sebagai anteseden,
- k adalah konstanta tegas (*crisp*) sebagai konsekuen,
- $\circ$ adalah operator fuzzy (AND atau OR).

3) Model *Fuzzy Sugeno* Orde-Satu

Model *Fuzzy Sugeno* Orde-Satu menggunakan konsekuen berupa persamaan linear dari variabel input. Secara umum dirumuskan sebagai berikut:

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN

$$z = p_1x_1 + p_2x_2 + \dots + p_Nx_N + q$$

dengan keterangan:

- A_i adalah himpunan fuzzy ke- i sebagai anteseden,
- p_i adalah konstanta ke- i ,
- q adalah konstanta,
- z adalah output sistem fuzzy.

4) Tahapan Implementasi Metode *Fuzzy Sugeno*

Berdasarkan model tersebut, terdapat beberapa tahapan utama dalam implementasi metode *Fuzzy Sugeno*, yaitu sebagai berikut.

a. Pembentukan Himpunan Fuzzy

Pada tahap ini, variabel input sistem fuzzy yang masih berupa nilai tegas (*crisp*) ditransformasikan ke dalam himpunan fuzzy. Proses ini bertujuan untuk menentukan derajat keanggotaan setiap nilai input terhadap himpunan fuzzy yang telah ditentukan. Dengan demikian, nilai-nilai tegas dapat

digunakan untuk menghitung tingkat kebenaran premis pada setiap aturan dalam basis pengetahuan.

b. Aplikasi Fungsi Implikasi

Setiap aturan (*rule*) pada basis pengetahuan *fuzzy* direpresentasikan dalam bentuk relasi *fuzzy*. Bentuk umum aturan *fuzzy* adalah sebagai berikut:

IF x is A THEN y is B

Keterangan :

- x dan y adalah variabel input dan output,
- A dan B adalah himpunan *fuzzy*.

Aturan tersebut dapat diperluas dengan beberapa variabel input menggunakan operator *fuzzy*:

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_N \text{ is } A_N)$ THEN y is B

Pada metode *Fuzzy Sugeno*, fungsi implikasi yang digunakan adalah fungsi minimum (MIN), yaitu dengan mengambil nilai derajat keanggotaan terkecil dari seluruh anteseden sebagai nilai *fire strength* (α).

c. Defuzzifikasi

Defuzzifikasi merupakan proses untuk menghasilkan output akhir berupa nilai tegas (*crisp*). Pada metode *Fuzzy Sugeno*, proses defuzzifikasi dilakukan menggunakan metode *Weighted Average* (WA).

Rumus defuzzifikasi *Fuzzy Sugeno* adalah sebagai berikut:

$$WA = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n}$$

Keterangan :

- WA = nilai output akhir,
- α_n = nilai predikat (*fire strength*) aturan ke- n ,

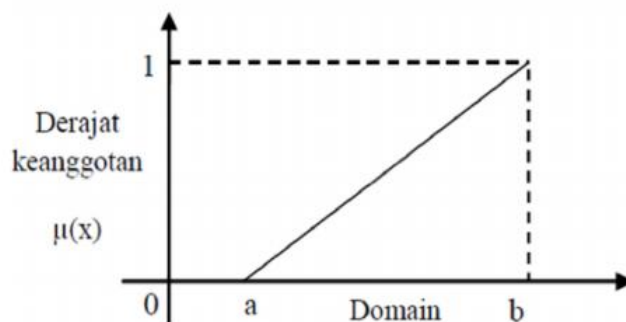
- z_n = nilai output (konstanta atau persamaan linear) pada aturan ke- n .

2.1.5.3 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) merupakan suatu kurva yang menggambarkan hubungan antara nilai input dengan derajat keanggotaannya dalam suatu himpunan *fuzzy*. Fungsi ini berperan sebagai alat pemetaan nilai input ke dalam nilai keanggotaan pada interval tertentu. Salah satu pendekatan yang umum digunakan untuk menentukan nilai keanggotaan adalah melalui pendekatan fungsi matematis (Wardani et al., 2017). Terdapat beberapa bentuk fungsi keanggotaan yang sering digunakan, antara lain representasi linier, kurva segitiga, kurva trapesium, kurva bahu, dan kurva-S, dengan penjelasan sebagai berikut:

1) Representasi Linear

Pada representasi linier, pemetaan nilai input terhadap derajat keanggotaannya digambarkan dalam bentuk garis lurus. Bentuk ini merupakan representasi yang paling sederhana dan sering digunakan untuk mendekati suatu konsep yang tidak memiliki batasan yang tegas. Representasi linier memiliki dua kondisi, yaitu linier naik dan linier turun.



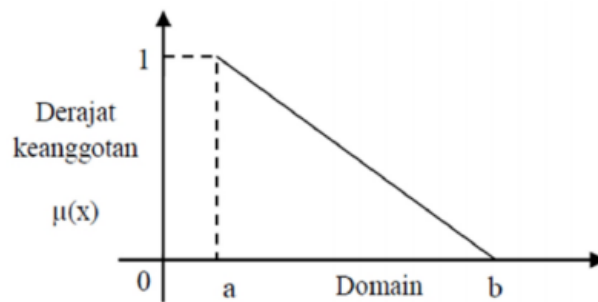
Gambar 2 2 Representasi Linear naik

Fungsi keanggotaan representasi linear naik yang dapat dilihat pada persamaan:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b \end{cases}$$

Keterangan:

- a = nilai minimum
- b = nilai maksimum



Gambar 2 3 Representasi Linear Turun

Fungsi keanggotaan representasi linear rumus yang dapat dilihat pada persamaan:

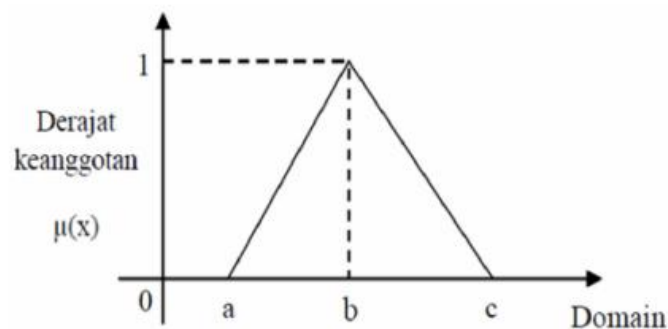
$$\mu(x) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

Keterangan :

- x : nilai input
- a : batas bawah, yaitu nilai input dengan derajat keanggotaan maksimum
- b : batas atas, yaitu nilai input dengan derajat keanggotaan minimum
- $\mu(x)$: derajat keanggotaan *fuzzy*

2) Representasi Kurva Segitiga

Representasi kurva segitiga pada dasarnya merupakan kombinasi dari dua fungsi linier, yaitu linier naik dan linier turun, yang membentuk puncak di satu titik tertentu.



Gambar 2 4 Representasi Kurva Segitiga

Fungsi keanggotaan representasi kurva segitiga yang dapat dilihat pada persamaan:

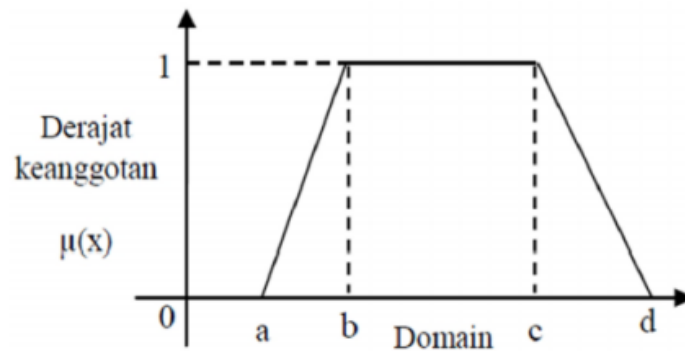
$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \end{cases}$$

Keterangan :

- a= batas bawah
- b= titik puncak ($\mu = 1$)
- c= batas atas
- x= nilai input

3) Representasi Kurva Trapezium

Kurva trapesium memiliki bentuk yang mirip dengan kurva segitiga, namun terdapat rentang nilai tertentu yang memiliki derajat keanggotaan sama dengan 1, sehingga bagian puncaknya berbentuk datar.



Gambar 2 5 Representasi Kurva Trapesium

Fungsi keanggotaan representasi kurva trapesium yang dapat dilihat pada persamaan:

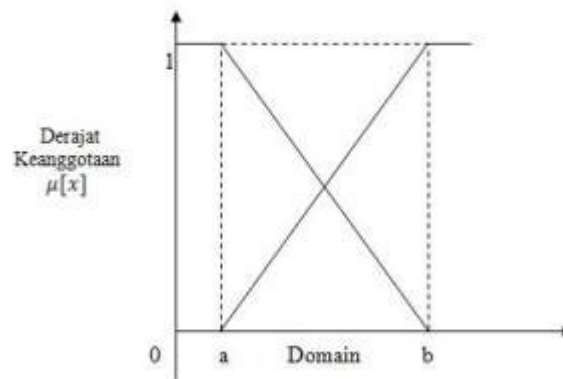
$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x < d \end{cases}$$

Keterangan :

- a = batas bawah
- b = awal nilai keanggotaan maksimum
- c = akhir nilai keanggotaan maksimum
- d = batas atas
- x = nilai input

4) Representasi Kurva Bahu

Representasi kurva bahu digunakan untuk menggambarkan daerah tengah dari suatu variabel. Bentuknya menyerupai segitiga, dengan sisi kiri dan kanan yang menunjukkan kenaikan dan penurunan nilai keanggotaan.



Gambar 2 6 Representasi Kurva Bahu

Fungsi keanggotaan representasi kurva bahu yang dapat dilihat pada persamaan:

Digunakan untuk himpunan fuzzy dengan nilai rendah.

$$\mu(x) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

Keterangan:

- Fungsi ini digunakan untuk himpunan fuzzy dengan nilai rendah
- Nilai keanggotaan paling tinggi (1) terjadi saat $x \leq a$
- Nilai keanggotaan menurun secara linier pada interval $a < x < b$
- Nilai keanggotaan menjadi 0 saat $x \geq b$

Makna Parameter:

- x : nilai input (crisp)
- a : batas bawah dimana keanggotaan mulai berkurang
- b : batas atas dimana keanggotaan menjadi nol
- Digunakan untuk himpunan fuzzy dengan nilai tinggi.

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b \end{cases}$$

Keterangan:

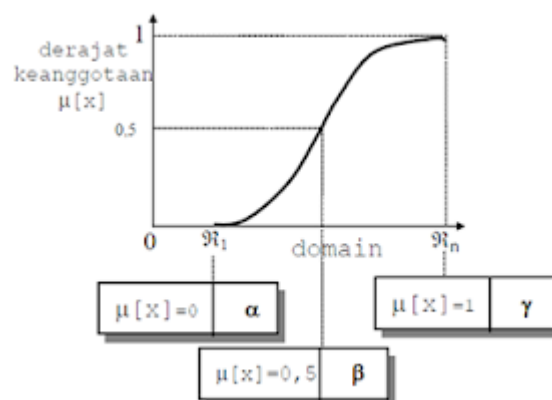
- Fungsi ini digunakan untuk himpunan fuzzy dengan nilai tinggi
- Nilai keanggotaan 0 saat $x \leq a$
- Nilai keanggotaan meningkat secara linier pada interval $a < x < b$
- Nilai keanggotaan maksimum (1) saat $x \geq b$

Makna Parameter:

- x : nilai input (crisp)
- a : batas bawah mulai naiknya keanggotaan
- b : batas atas saat keanggotaan penuh

5) Representasi Kurva – S

Kurva-S atau kurva sigmoid menggambarkan proses pertumbuhan dan penyusutan secara tidak linier. Pada kurva-S pertumbuhan, nilai keanggotaan meningkat secara bertahap dari sisi kiri dengan derajat keanggotaan mendekati 0 hingga sisi kanan dengan derajat keanggotaan bernilai 1.



Gambar 2 7 Representasi Kurva – S

Fungsi keanggotaan representasi kurva – S yang dapat dilihat pada persamaan:

$$\mu(x) = \frac{1}{1 + e^{-a(x-c)}}$$

Keterangan :

- $\mu(x)$: derajat keanggotaan
- x : nilai input
- a : parameter kemiringan kurva
- c : titik tengah (nilai ambang)

2.1.6 Skala Prioritas

Skala merupakan perbandingan antar kategori dimana setiap kategori diberi bobot nilai yang berbeda. Skala terbagi menjadi 4 yaitu *nominal*, *ordinal*, *interval* dan *ratio*. Penelitian ini menggunakan skala *interval*. Skala *interval* merupakan skala pengukuran yang memiliki jarak atau *interval* yang sama antar tingkatan nilai sehingga dapat menunjukkan perbedaan yang terukur antar data. Skala *interval* digunakan untuk mengelompokkan dan membandingkan objek berdasarkan rentang nilai tertentu, namun tidak memiliki nilai nol mutlak. Dengan demikian, skala *interval* memungkinkan data diurutkan sekaligus menunjukkan besarnya perbedaan antar tingkatan nilai (Stevens, 1946). Sedangkan prioritas adalah yang didahulukan dan diutamakan daripada yang lain (KBBI, 2021). Sehingga dapat disimpulkan Skala prioritas adalah urutan angka penilaian berdasarkan tingkatan dari tinggi ke rendah yang didahulukan atau diutamakan dari pada yang lain.

2.1.7 Dinas Ketahanan Pangan dan Pertanian Kabupaten Kuningan

Dinas Ketahanan Pangan dan Pertanian Kabupaten Kuningan adalah instansi pemerintah daerah yang bertanggung jawab mengelola urusan pertanian dan dipimpin oleh seorang Kepala Dinas yang melapor kepada Bupati melalui Wakil Bupati. Menurut Diskatan Kuningan 2025 Dinas Ketahanan Pangan dan Pertanian merupakan salah satu institusi yang mempunyai peran penting dan akses besar dalam mensukseskan

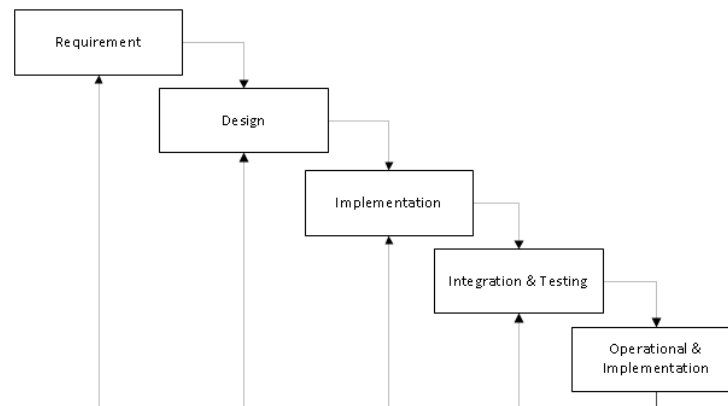
pembangunan pertanian yang dalam pelaksanaannya tidak terlepas dari kerjasama/ koordinasi baik lintas sektoral maupun vertikal dengan dinas, instansi, badan lembaga (*Dibale*) terkait serta memperhatikan aspirasi masyarakat khususnya para Kelompok Tani.

Berdasarkan Peraturan Daerah Kabupaten Kuningan Nomor 6 Tahun 2024 tentang Perubahan Kedua Atas Peraturan Daerah Kabupaten Kuningan Nomor 5 Tahun 2016 tentang Pembentukan dan Susunan Perangkat Daerah Kabupaten Kuningan Tugas, fungsi, dan struktur organisasi Dinas Pertanian diatur lebih lanjut oleh Peraturan Bupati Kuningan Nomor 57 Tahun 2019 tentang status, struktur organisasi, tugas dan fungsi utama, serta tata cara kerja aparat pemerintah daerah, yang berfungsi sebagai dasar operasional untuk melaksanakan kegiatan pertanian di daerah tersebut. Penelitian ini dilakukan di Dinas Pertanian Kabupaten Kuningan, yang memiliki peran strategis dalam perencanaan dan pelaksanaan program pertanian berdasarkan CPCL. Melalui peran ini, Dinas Pertanian Kabupaten Kuningan berkontribusi dalam meningkatkan efektivitas program pengembangan pertanian dan meningkatkan kesejahteraan petani di Kabupaten Kuningan.

2.1.8 Metode Pengembangan Sistem

System Development Life Cycle (SDLC) digunakan untuk mengembangkan sistem informasi manajemen perjalanan dinas secara digital yang menggantikan proses manual berbasis Excel. Tahapan SDLC seperti analisis kebutuhan, desain sistem, dan implementasi dilakukan secara terstruktur sehingga menghasilkan sistem yang efektif dalam meningkatkan efisiensi, akurasi data, dan transparansi informasi real-time. Penelitian ini menunjukkan bahwa penggunaan SDLC mampu mengurangi penggunaan dokumen fisik dan mempercepat alur kerja administrasi (Septiani et al., 2025).

Metode pengembangan sistem yang digunakan untuk penelitian ini adalah metode waterfall. Tahapan Metode Waterfall (Bakri & Nasution, 2024):



Gambar 2.8 Tahapan Metode *Waterfall* (Pressman, Roger S. *Software Engineering: A Practitioner's Approach.*)

1) *Requirement*

Pada tahap ini adalah tahapan pencarian seluruh informasi mengenai kebutuhan *software* seperti kegunaan *software* yang diinginkan oleh Dinas Ketahanan Pangan Kuningan dan batasan *software*. Pada penelitian ini, informasi diperoleh dari hasil wawancara, Observasi, ataupun diskusi. Kemudian informasi dianalisis sehingga mendapatkan data-data yang lengkap mengenai kebutuhan pengguna akan *software* yang akan dikembangkan.

2) *Design*

Tahap desain ini dilakukan sebelum proses coding dimulai. Tujuannya adalah untuk memberikan gambaran lengkap tentang apa yang harus dikerjakan dan juga bagaimana tampilan dari sebuah sistem yang diinginkan sehingga membantu menspesifikan kebutuhan *hardware* dan sistem, juga mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan.

3) *Implementation*

Pada tahap ini yaitu mulai proses penulisan code. Pembuatan *software* akan dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

4) *Integration & Testing*

Pada tahap keempat ini akan dilakukan penggabungan fitur-fitur yang sudah dibuat sebelumnya. Setelah itu akan dilakukan pengujian yang bertujuan untuk mengetahui apakah *software* sudah sesuai desain yang diinginkan dan apakah masih ada kesalahan atau tidak.

5) *Operation & Maintenance*

Pada tahapan ini, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan (Bakri & Nasution, 2024).

2.1.9 Pemodelan dan Perancangan Sistem

2.1.9.1 Rich Picture

Menurut Bayu Waseso et al (2021). *Soft System Methodology* (SSM). Dalam pendekatannya terbagi menjadi beberapa tahapan. Tahap pertama adalah pemetaan situasi permasalahan yang dianggap bermasalah, menggambarkan masalah dengan cara yang konkret atau lebih abstrak. Kemudian, tahap kedua melibatkan pengungkapan situasi permasalahan melalui *Rich Picture*, menggunakan gambaran yang kaya akan detail dan konteks. Sementara itu, tahap ketiga mencakup pembuatan root definition dari sistem aktivitas yang berkaitan dengan tujuan yang relevan, dengan merumuskan definisi yang terkait dengan aspek-aspek kunci dari situasi masalah. Tahap keempat melibatkan pembuatan model konseptual dari sistem yang disebutkan dalam root definition dengan menyusun representasi visual yang menggambarkan permasalahan secara terstruktur. Tahap kelima adalah membandingkan model dengan kondisi aktual, mengevaluasi model hasil rancangan dengan keadaan yang ada di lapangan. Tahap

keenam mengedepankan keinginan secara sistematis dan memastikan kelayakan secara budaya dengan mengidentifikasi rekomendasi tindakan untuk perbaikan dan mengintegrasikan model konseptual yang diusulkan. Terakhir, tahap ketujuh melibatkan pelaksanaan tindakan perbaikan terhadap situasi permasalahan, dengan melakukan langkah-langkah yang telah direncanakan untuk mengatasi masalah yang ada (Alfathsyah et al., 2024).

2.1.9.2 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) merupakan sebuah metode pemodelan visual yang digunakan untuk membuat sebuah sistem berbasis objek. Beberapa diagram UML yang umum digunakan adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *State Machine Diagram*, dan *Component Diagram* (Noneng Marthiawati et al., 2024).

UML memiliki beberapa macam kelebihan, contohnya seperti memberikan bahasa pemodelan visual bagi *developer* dari berbagai macam bahasa pemrograman dan proses rekayasa secara umum. UML juga menggabungkan berbagai elemen terbaik dalam pemodelan, memberikan ilustrasi yang jelas bagi pengembang sistem, serta memungkinkan pemodelan sistem berorientasi objek. Perangkat ini juga memudahkan developer memahami struktur sistem dan berfungsi sebagai *blueprint* yang menyediakan informasi lebih mendalam untuk membantu pengembangan hingga tahap penulisan kode program (Hidayati et al., 2023). Adapun beberapa macam jenis UML adalah sebagai berikut:

1. Use Case Diagram

Use Case Diagram merupakan salah satu jenis diagram di dalam *Unified Modeling Language (UML)* yang berguna untuk menggambarkan fungsi sebuah sistem dari pandangan pengguna. Diagram ini dapat membantu kita dalam perancangan interaksi antara sistem dan pengguna. *Use Case Diagram* membantu

developer memahami kebutuhan fungsional sistem dan bagaimana pengguna akan berinteraksi dengan sistem yang sudah dibuat tersebut. Elemen utama yang ada pada *Use Case Diagram* adalah sebagai berikut:

- 1) *Actor* (Aktor): Aktor adalah entitas yang berinteraksi dengan sistem. Aktor tersebut bisa berupa pengguna, sistem lain, atau perangkat eksternal yang berkomunikasi dengan sistem lain. Aktor bisa berupa *Primary Actor* (yang membutuhkan layanan dari sistem) atau bisa menjadi *Secondary Actor* (yang berperan sebagai pendukung sistem untuk memberikan layanan tersebut).
- 2) *Use Case* (Kasus Pengguna): Kasus pengguna merupakan sebuah fungsi atau aktivitas dari suatu sistem yang memberikan nilai pada aktor tertentu. Setiap kasus pengguna menggambarkan sebuah layanan atau proses yang dilakukan oleh suatu sistem. *Use case* berfokus pada bagaimana sebuah sistem dapat merespons tindakan yang dilakukan oleh aktor dan bagaimana aktor dapat berinteraksi dengan sistem tersebut.

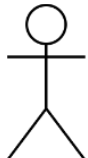
Contoh dari *Use Case* misalnya dalam sistem pemesanan tiket pesawat. Aktor sebagai “pelanggan” dan *use case* seperti “Membeli Tiket”. Alurnya pelanggan melakukan pencarian tiket, menentukan jadwal pemberangkatan, memilih kelas atau fasilitas, memasukkan data pribadi dan melakukan pembayaran untuk membeli tiket. Berikut adalah fungsi dari *use case diagram* yaitu:

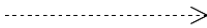
- 1) Membantu dalam menggambarkan berbagai fungsi yang harus ada di dalam sistem yang dibuat sesuai dengan kebutuhan dari pengguna.
- 2) Dapat membantu kita untuk menggambarkan siapa saja yang dapat berinteraksi dengan sistem dan apa saja fungsi atau layanan yang mereka inginkan pada sistem tersebut.

- 3) *Use case* diagram membantu pengembang dengan memberikan gambaran visual yang sederhana dan mudah untuk dipahami.
- 4) Dengan diberikannya gambaran alur interaksi antara aktor dengan sistem, diagram ini dapat mempermudah pengembang untuk membuat sebuah sistem yang dapat memenuhi kebutuhan pengguna dengan lebih efektif.

Use case diagram kerap digunakan pada tahap awal pengembangan perangkat lunak dan diagram tersebut dapat membantu dalam komunikasi antara tim pengembang, para pengguna, dan pemangku kepentingan

Tabel 2. 1 *Use Case Diagram* (Sumber : <https://app.diagrams.net/>)

Gambar	Nama	Fungsi
	<i>Package</i>	Untuk menambahkan paket baru di dalam diagram.
	<i>Actor</i>	Untuk menambahkan aktor di dalam diagram.
	<i>Use case</i>	Untuk menambahkan <i>use case</i> di dalam diagram.
	<i>Unidirectional Association</i>	Menunjukkan relasi atau hubungan antara aktor dengan <i>use case</i> .

Gambar	Nama	Fungsi
	<i>Dependencies or Instantiates</i>	Menunjukkan ketergantungan antar item di dalam diagram.
	<i>Generalization</i>	Menunjukkan hubungan atau relasi lanjut antar <i>use case</i> atau menunjukkan struktur pewarisan antar aktor.

2. Activity Diagram

Activity Diagram merupakan jenis diagram yang terdapat pada *Unified Modeling Language* (UML) yang berguna untuk menggambarkan atau mendeskripsikan alur proses di dalam sistem. Diagram ini menggambarkan berbagai macam langkah-langkah yang dilakukan pada suatu proses atau aktivitas yang lebih kompleks. Diagram ini mirip seperti *flowchart*, namun lebih terstruktur dan lebih fokus pada alur eksekusi aktivitas, alur keputusan, dan pemrosesan bersamaan (*concurrent processing*). Elemen-elemen utama yang ada pada *activity* diagram yaitu:

- 1) *Activity* (Aktivitas): Aktivitas adalah langkah atau tindakan yang dilakukan di dalam suatu proses. Aktivitas dapat mewakili operasi yang sedang dilakukan oleh objek lainnya.
- 2) *Control Flow* (Alur Kontrol): Menunjukkan jalur yang diikuti oleh alur eksekusi antar aktivitas. Alur kontrol dapat digambarkan dengan garis putus atau putus-putus yang menunjukkan urutannya.
- 3) *Initian Node* (Titik Awal): Tanda ini menandakan titik awal dari sebuah proses. Biasanya berupa lingkaran berisi titik hitam.

- 4) *Final Code* (Titik Akhir): Tanda ini menunjukkan akhir dari sebuah aktivitas atau proses. Digambarkan seperti lingkaran dengan batas tebal atau titik hitam dengan lingkaran.
- 5) *Decision Node* (Node Keputusan): Menjelaskan titik di mana alur proses atau aktivitas harus memilih satu keputusan berdasarkan kondisi tertentu. Biasanya digambarkan dengan belah ketupat.
- 6) *Fork Node* (Node Pemisah): Menjelaskan tentang pemisahan alur menjadi dua atau lebih jalur keputusan secara paralel. Ditandai dengan gambar berbentuk garis vertikal atau horizontal yang menghubungkan beberapa jalur dan kemudian kembali bersatu.
- 7) *Join Node* (Node Penyatuan): Tanda ini menjelaskan adanya penyatuan jalur yang berjalan paralel kembali menyatu. Biasanya digambarkan dengan garis vertikal atau horizontal yang menghubungkan beberapa jalur kembali menjadi satu.
- 8) *Object Node* (Node Objek): Dijelaskan sebagai objek yang terlibat di dalam aktivitas mewakili entitas yang menerima data selama aktivitas itu berjalan.

Adapun fungsi dan tujuan dari *activity* diagram adalah sebagai berikut:

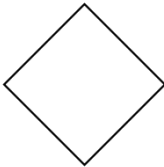
- 1) Berguna untuk memberikan gambaran langkah-langkah dalam proses bisnis atau alur kerja yang lebih kompleks.
- 2) Dengan menggunakan *activity* diagram, pengembang dapat menggambarkan bagaimana aktivitas atau pekerjaan yang akan dilakukan di dalam sistem dan mengidentifikasi alur eksekusinya.
- 3) Dengan menggunakan *activity* diagram, pengembang dapat menggambarkan bagaimana aktivitas atau pekerjaan yang akan dilakukan di dalam sistem dan mengidentifikasi alur eksekusinya.

- 4) Membantu pengembang untuk mendesain atau merancang sistem aplikasi dengan memasukkan langkah-langkah yang perlu dilakukan untuk mencapai tujuan sistem tertentu.
- 5) Membantu semua pemangku kepentingan, termasuk pengguna dan pengembang untuk dapat memahami cara kerja dari sebuah sistem dengan menggambarkan alur dengan cara yang mudah dipahami.

Activity diagram sangat berguna untuk mendeskripsikan atau mendokumentasikan alur kerja adalah sebuah sistem, dengan berfokus pada aktivitas yang sedang dilakukan, alur kontrol antar aktivitas, serta proses yang berjalan secara paralel. Diagram ini sangat membantu baik untuk pengembang, pengguna, maupun pemangku kepentingan agar dapat memahami lebih jelas tentang bagaimana cara sistem bekerja dan bagaimana proses dalam sistem saling berinteraksi.

Tabel 2. 2 Activity Diagram (Sumber : <https://app.diagrams.net/>)

Gambar	Nama	Fungsi
	<i>State</i>	Untuk menambahkan state pada suatu objek
	<i>Activity</i>	Untuk menambahkan aktivitas baru di dalam diagram
	<i>Start State</i>	Menggambarkan aliran kerja berawal
	<i>End State</i>	Menggambarkan aliran kerja berakhir

Gambar	Nama	Fungsi
	<i>State Transition</i>	Untuk menambahkan transisi dari suatu kegiatan ke kegiatan yang lainnya
	<i>Decisions Points</i>	Untuk menambahkan titik keputusan pada aliran kerja

3. Class Diagram

Class Diagram merupakan salah satu jenis diagram yang ada pada *Unified Modeling Language* (UML) yang memiliki fungsi untuk menggambarkan struktur statis dari sistem perangkat lunak dengan menampilkan beberapa kelas yang ada pada sistem dan hubungan antar kelas tersebut. *Class diagram* tidak mendeskripsikan perilaku sistem secara dinamis, tetapi lebih kepada struktur data dan hubungan antar komponen sistem. Elemen utama yang ada pada *class diagram*:

- 1) *Class* (Kelas): Kelas merupakan entitas utama pada diagram ini, menggambarkan tipe objek yang ada pada sistem tersebut. Kelas biasanya berisi atribut dan metode.
- 2) *Attributes* (Atribut): Atribut ialah properti atau data yang dimiliki oleh kelas. Atribut bisa berbentuk variabel atau parameter yang mendeskripsikan atau menggambarkan suatu objek. Contohnya nama, alamat, jabatan, dan lain-lain.
- 3) *Methods* (Metode): Metode merupakan suatu operasi atau fungsi yang dapat digunakan pada objek dari kelas tersebut. Metode menggambarkan tindakan yang dapat dilakukan oleh

objek pada sebuah sistem. Contohnya `hitung_harga()`, `edit_data()`, `update_data()`, dan lain-lain.

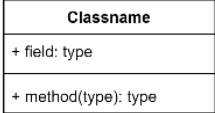
- 4) *Relationship* (Hubungan): Merujuk pada hubungan antar kelas, seperti hubungan agregasi, komposisi, asosiasi, pewarisan (*inheritance*), dan ketergantungan.
- 5) *Interface* (Antarmuka): Berisi sekumpulan metode yang harus diaplikasikan oleh kelas yang menggunakan *interface* tersebut.

Fungsi dari *class diagram* adalah sebagai berikut:

- 1) Memberikan gambaran tentang bagaimana struktur objek dalam sistem dibuat dan bagaimana sekumpulan objek tersebut bisa saling terhubung.
- 2) Membantu dalam membuat struktur kode yang lebih sistematis dengan mengidentifikasi bagaimana kode-kode tersebut akan berinteraksi satu sama lain.
- 3) *Class diagram* sangat penting dalam fase merancang sistem untuk menggambarkan bagaimana kumpulan komponen yang ada dalam sistem bisa saling berinteraksi.
- 4) Diagram ini membantu para pengembang untuk menganalisis entitas utama di dalam sebuah sistem, serta relasi antar entitas seperti agregasi, asosiasi, dan pewarisan.

Class diagram adalah alat yang sangat penting dalam membuat perencanaan perangkat lunak yang memberikan gambaran struktur statis dari sistem dengan memperlihatkan kelas-kelas dan hubungan antar kelas. Dengan adanya diagram ini dapat membantu pengembang dalam merancang, membuat, dan memahami struktur sistem, serta memudahkan komunikasi antara tim pengembang dengan pemangku kepentingan.

Tabel 2. 3 *Class Diagram* (Sumber : <https://app.diagrams.net/>)

Gambar	Nama	Fungsi
	<i>Class</i>	Untuk menambahkan kelas baru di dalam diagram
	<i>Interface</i>	Untuk menambahkan interface di dalam diagram
	<i>Association</i>	Menunjukkan relasi atau hubungan asosiasi
	<i>Association Class</i>	Menyatukan kelas asosiasi pada suatu relasi
	<i>Generalization</i>	Menunjukkan hubungan generalisasi
	<i>Aggregation</i>	Menunjukkan hubungan agregasi

4. Sequence Diagram

Sequence Diagram merupakan salah satu jenis diagram yang terdapat dalam *Unified Modeling Language* (UML) yang biasanya digunakan untuk memberi gambaran urutan atau alur jalannya interaksi antar objek di dalam sistem berdasarkan waktu. Diagram ini memperlihatkan bagaimana objek-objek dalam sistem bisa saling berkomunikasi melalui pesan yang dikirimkan, maka dari itu sequence diagram dapat membantu dalam

memahami bagaimana skenario tertentu bisa berjalan sampai akhir. Elemen-elemen utama yang ada pada diagram ini adalah:

- 1) Objek atau aktor: sebuah entitas yang tergabung di dalam interaksi. Biasanya objek digambarkan di bagian atas diagram dengan bentuk kotak dan di dalam kotak tersebut menampilkan nama dari objek atau aktor.
- 2) *Lifeline* (Garis Kehidupan): Merupakan garis vertikal yang berada di bawah setiap objek atau aktor yang memperlihatkan keberadaan objek tersebut dalam alur waktu.
- 3) *Message* (Pesan): Menggambarkan bentuk komunikasi antar objek berupa panah. Pesan tersebut bisa digunakan untuk permintaan, panggilan metode, atau balasan.
- 4) *Self-Message* (Pesan Diri Sendiri): Menggambarkan sebuah objek yang memanggil metode pada dirinya sendiri.
- 5) *Timeline* (Garis Waktu) : Menunjukkan urutan atau alur waktu dari interaksi antar objek, dimulai dari bagian atas hingga bawah.
- 6) *Loop*: Blok yang memperlihatkan aktivitas yang berulang atau pengulangan di dalam interaksi antar objek.
- 7) *Condition* (Kondisi): Menggambarkan kondisi tertentu yang harus terpenuhi untuk mengirim pesan atau meneruskan ke langkah selanjutnya.
- 8) *Alternative* (Alternatif): Blok yang menggambarkan percabangan di dalam alur interaksi, seperti kondisi *if-else*. Hal ini dapat mempermudah alur untuk memilih antara beberapa jalur yang berbeda.

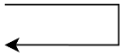
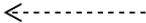
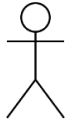
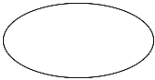
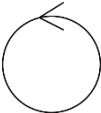
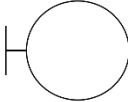
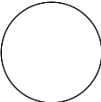
Berikut adalah fungsi dari *sequence diagram*:

- 1) *Sequence diagram* menunjukkan urutan pesan dalam alur waktu, dengan begitu diagram ini dapat membantu memahami bagaimana objek berinteraksi satu dengan yang lainnya secara berurutan.
- 2) Diagram ini mendistribusikan skenario atau kasus penggunaan secara lebih rinci dan membantu menggambarkan interaksi antar objek di dalam proses tertentu.
- 3) Diagram ini dapat membantu dengan menggunakannya sebagai referensi bagi para pengembang untuk menuliskan kode-kode karena diagram ini memberikan sebuah gambaran alur sebuah proses yang dibutuhkan dalam sistem.
- 4) Membantu dalam menguasai peran masing-masing objek di dalam skenario tertentu dan bagaimana objek tersebut berintegrasi.

Sequence diagram merupakan alat yang sangat berguna untuk merancang sistem. Diagram ini menggambarkan bagaimana sekumpulan objek dalam sistem bisa saling berinteraksi berdasarkan urutan waktu.

Tabel 2. 4 *Sequence Diagram* (Sumber : <https://app.diagrams.net/>)

Gambar	Nama	Fungsi
	<i>Object</i>	Untuk menambahkan objek baru di dalam diagram
	<i>Object Message</i>	Menunjukkan pesan antara dua objek

Gambar	Nama	Fungsi
	<i>Message to Self</i>	Menunjukkan pesan yang tertuju pada diri sendiri
	<i>Return Message</i>	Menunjukkan pengembalian dari sebuah pemanggilan prosedur
	<i>Destruction Marker</i>	Menampilkan pada saat objek tertentu dihancurkan
	<i>Actor</i>	Menampilkan aktor di dalam diagram kelas
	<i>Use Case</i>	Menampilkan use case di dalam diagram kelas
	<i>Control</i>	Menampilkan unsur kendali dalam diagram
	<i>Boundary</i>	Untuk menambahkan kelas batasan di dalam diagram
	<i>Entity</i>	Untuk menambahkan kelas entitas di dalam diagram

2.1.10 Konsep Basis Data

Basis data merupakan kumpulan data yang saling berhubungan dan disimpan secara terstruktur dalam suatu sistem penyimpanan sehingga dapat dikelola, dikelola, dan diakses kembali secara efisien. Dalam konsep dasar data relasional, data direpresentasikan dalam bentuk tabel yang terdiri atas baris (*record*) dan kolom (*field*). Setiap tabel mewakili satu entitas tertentu, sedangkan kolom diyakini sebagai atribut yang menggambarkan karakteristik dari entitas tersebut. Hubungan antar tabel terbentuk melalui relasi, yang umumnya menggunakan kunci utama (*primary key*) dan kunci tamu (*foreign key*), sehingga memungkinkan keterkaitan data antar entitas secara logistik dan terintegrasi (Putra & Novembrianto, 2021).

Pengelolaan basis data dilakukan menggunakan *Database Management System* (DBMS), yaitu perangkat lunak yang berfungsi untuk mengelola penyimpanan, pemrosesan, pengamanan, serta pengendalian akses data. DBMS menyediakan fasilitas untuk melakukan operasi manipulasi data, seperti penyimpanan, pembaruan, penghapusan, dan pengambilan data sesuai kebutuhan sistem (Tumini & Fitria, 2021).

1) Relasional

Relasional merupakan konsep yang menggambarkan hubungan antar tabel dalam basis data. Hubungan ini dapat berupa *one-to-one*, *one-to-many*, atau *many-to-many*, yang digunakan untuk merepresentasikan keterkaitan antar entitas dalam sistem. Relasi dibangun menggunakan kunci utama dan kunci tamu agar integritas data tetap terjaga serta mendukung proses normalisasi data (Suryadi, 2019).

Adapun Normalisasi menurut (Nurhayati et al., 2025):

a) Bentuk Tidak Normal / Bentuk Tidak Normal (UF)

Data awal disusun dalam bentuk *Unnormalized Form* yang menggabungkan seluruh informasi ke dalam satu baris. Pada tahap ini jika ditemukan pelanggaran terhadap prinsip normalisasi seperti

adanya nilai *multivalued*, sehingga diperlukan proses normalisasi lebih lanjut.

b) Bentuk Normal Kesatu / Bentuk Normal Pertama (1NF)

Pada tahap 1NF, data *multivalued* dipecah menjadi nilai *atom* (tunggal) dan setiap entitas dipisahkan ke dalam tabel tersendiri. Kunci primer ditetapkan untuk setiap tabel guna menjamin keunikan data dan mendukung hubungan antar entitas. Struktur data mulai terorganisir dan bebas dari umum.

c) Bentuk Normal Kedua / Bentuk Normal Kedua (2NF)

Tahap ini menghilangkan ketergantungan parsial, yaitu ketergantungan atribut non-primer terhadap Sebagian dari kunci komposit. Tabel-tabel yang memiliki kunci komposit, disusun ulang agar setiap atribut bergantung sepenuhnya pada kunci utama. Kunci asing mulai diterapkan untuk membangun relasi antar entitas secara konsisten.

d) Bentuk Normal Ketiga / Third Normal Form (3NF)

Tahap akhir normalisasi bertujuan menghilangkan ketergantungan transitif, yaitu ketergantungan atribut non- kunci terhadap atribut non-kunci lainnya. Hasil akhir tahap ini berupa struktur basis data yang efisien, terstruktur dan siap diterapkan

2) Field

Field merupakan kolom dalam sebuah tabel yang berfungsi untuk menyimpan atribut atau karakteristik tertentu dari suatu entitas. Setiap *field* memiliki nama, tipe data, dan domain nilai tertentu yang membatasi jenis data yang dapat disimpan. Penggunaan *field* yang tepat membantu meningkatkan konsistensi data dan mempermudah proses pengolahan data dalam sistem basis data (Tumini & Fitria, 2021).

3) Entitas

Entitas adalah objek nyata atau konseptual yang datanya disimpan dalam basis data dan dapat dibedakan secara unik. Setiap entitas direpresentasikan dalam bentuk tabel dan memiliki sekumpulan

atribut (*field*) yang menjelaskan karakteristiknya. Dalam sistem informasi, entitas digunakan untuk memodelkan objek utama yang terlibat dalam proses bisnis, seperti pengguna, data pendaftaran, atau hasil penilaian (Suryadi, 2019).

4) Kunci

a) *Primary key*

Primary key adalah atribut atau kombinasi atribut yang digunakan untuk mengidentifikasi setiap record dalam suatu tabel secara unik. Nilai *primary key* tidak boleh sama dan tidak boleh bernilai kosong (*NULL*). *Primary key* memastikan bahwa setiap data dapat dibedakan secara jelas dalam tabel (Putra & Novembrianto, 2021).

b) *Foreign Key*

Foreign key adalah atribut dalam suatu tabel yang merujuk pada *primary key* di tabel lain. *Foreign key* digunakan untuk membangun relasi antar tabel dan menjaga konsistensi data antar entitas. Dengan adanya *foreign key*, integritas referensial dapat dipertahankan sehingga data yang saling berhubungan tetap valid (Suryadi, 2019).

2.1.11 Konsep Pengujian Sistem

Pengujian sistem merupakan tahapan pengujian perangkat lunak yang bertujuan untuk memastikan bahwa setiap unit atau modul sistem telah berfungsi sesuai dengan kebutuhan yang ditetapkan. Pengujian ini dilakukan sebelum sistem diintegrasikan secara keseluruhan, sehingga kesalahan dapat diidentifikasi dan diperbaiki sejak dini. Menurut Pressman dan Maxim (2020), pengujian perangkat lunak merupakan aktivitas penting dalam siklus pengembangan sistem untuk menjamin kualitas, kerahasiaan, dan kesesuaian fungsi sistem dengan spesifikasi yang telah dirancang (Pressman & Maxim, 2020).

2.1.11.1 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang fokus pada pengujian fungsional sistem tanpa memperhatikan struktur internal atau kode program. Pengujian ini dilakukan dengan mengamati kesesuaian antara data masukan (input) dan keluaran (output) berdasarkan spesifikasi kebutuhan sistem. Metode ini banyak digunakan untuk menguji sistem berbasis web karena efektif dalam menilai fungsi aplikasi dari sudut pandang pengguna (Giansyah et al., 2023). Beberapa teknik yang digunakan dalam Black Box Testing antara lain:

1) Partisi Ekuivalensi

Equivalence Partitioning merupakan teknik pengujian dengan cara membagi data masukan ke dalam beberapa kelas ekuivalen yang memiliki karakteristik serupa. Setiap kelas diuji menggunakan satu atau beberapa data perwakilan untuk memastikan bahwa sistem dapat menangani seluruh variasi input dengan benar. Teknik ini bertujuan untuk mengurangi jumlah kasus uji tanpa mengurangi efektivitas pengujian (Abi Mayu Abdi, 2024).

2) Grafik Sebab-Akibat

Grafik Sebab-Akibat digunakan untuk mengidentifikasi hubungan sebab dan akibat antara kondisi input dan output yang dihasilkan sistem. Teknik ini membantu penguji dalam menguji kombinasi kondisi input yang kompleks secara sistematis sehingga kesalahan logika fungsional dapat terdeteksi dengan lebih baik (Abi Mayu Abdi, 2024).

3) Pengujian Fitur

Pengujian Fitur dilakukan untuk memastikan bahwa seluruh fitur yang tersedia dalam sistem telah berfungsi sesuai dengan kebutuhan pengguna. Setiap fitur utama diuji secara terpisah

untuk memastikan tidak terjadi kegagalan fungsi maupun ketidaksesuaian spesifikasi sistem (Pirdaus & Hidayana, 2024).

2.1.11.2 White Box Testing

White Box Testing merupakan metode pengujian perangkat lunak yang dilakukan dengan memperhatikan struktur internal sistem, termasuk logika program, alur kontrol, dan kode sumber. Pengujian ini bertujuan untuk memastikan bahwa setiap jalur logika dalam program telah berjalan sesuai dengan perancangan sistem dan bebas dari kesalahan logika (Giansyah et al., 2023).

White Box Testing umumnya dilakukan oleh pengembang sistem dan sangat penting dalam memastikan keakuratan algoritma serta efisiensi proses komputasi, terutama pada sistem yang menggunakan metode perhitungan seperti *Fuzzy Logic*. Beberapa teknik yang digunakan dalam White Box Testing antara lain:

1) *Decision (Branch) Coverage*

Decision Coverage bertujuan untuk memastikan bahwa setiap percabangan logika dalam program telah dieksekusi minimal satu kali selama proses pengujian. Teknik ini membantu mengidentifikasi kesalahan pada struktur keputusan program (Raychan Zen & Nuryasin, 2025).

2) *Condition Coverage*

Condition Coverage memastikan bahwa setiap kondisi logika dalam suatu keputusan telah diuji baik dalam kondisi benar (true) maupun salah (false). Teknik ini penting untuk mengurangi kesalahan logika yang dapat mempengaruhi hasil pemrosesan sistem (Nurfauziah & Jamaliyah, 2023).

3) *Path Analysis*

Path Analysis dilakukan untuk menguji seluruh kemungkinan jalur eksekusi program berdasarkan diagram alir dan desain sistem. Teknik ini memastikan bahwa alur program

sesuai dengan rancangan yang telah ditentukan (Prasetya & Nuryasin, 2024).

4) *Execution Time Testing*

Execution Time Testing digunakan untuk mengukur waktu eksekusi sistem, mulai dari proses input hingga menghasilkan output. Pengujian ini penting untuk memastikan bahwa sistem berjalan secara efisien dan responsif, terutama pada aplikasi berbasis web (Giansyah et al., 2023).

4) *Algorithm Analysis*

Algorithm Analysis dilakukan untuk mengevaluasi ketepatan dan efisiensi algoritma yang digunakan dalam sistem, termasuk algoritma *Fuzzy Logic* dalam proses pengambilan keputusan. Pengujian ini bertujuan memastikan bahwa perhitungan algoritma menghasilkan keluaran yang sesuai dengan perhitungan teoritis (Raychan Zen & Nuryasin, 2025).

2.1.11.3 UAT (*User Acceptance Testing*)

Menurut (Wendland, 2021) *User Acceptance Testing* adalah tahapan pengujian perangkat lunak yang bertujuan untuk memverifikasi bahwa sistem telah memenuhi kebutuhan serta harapan pengguna, sehingga perangkat lunak tersebut dinyatakan layak untuk diterima dan dioperasikan oleh pengguna akhir. Dan menurut (Blokdyk, 2021) *User Acceptance Testing* (UAT) merupakan suatu proses pengujian atau evaluasi yang bertujuan untuk memastikan bahwa sistem yang dikembangkan telah sesuai dengan kebutuhan bisnis yang ditetapkan serta memenuhi kriteria penerimaan dari sisi pengguna akhir sebelum sistem diterapkan secara penuh. Penerapan metode User Acceptance Test (UAT) menunjukkan bahwa tingkat penerimaan pengguna terhadap suatu sistem dapat menjadi salah satu indikator dalam menilai keberhasilan implementasi teknologi informasi. Dengan formula umum (Hermansah et al., 2025):

1. Menghitung Jumlah Bobot

$$\text{Jumlah Bobot} = (SS \times 5) + (S \times 4) + (N \times 3) + (TS \times 2) + (STS \times 1) \quad (1)$$

2. Menghitung Mean

$$\bar{X} = \frac{\Sigma X}{N} \quad (2)$$

Keterangan:

- $\bar{X}$ = Mean
- ΣX = Jumlah Bobot
- N = Jumlah Responden (30)

3. Menghitung Persentase

$$\text{Persentase} = \frac{\text{Mean}}{5} \times 100\% \quad (3)$$

4. Hasil Akhir UAT

Rumus:

$$NA = \frac{PV_A + PV_B + PV_C + PV_D}{4} \quad (4)$$

Keterangan:

NA = Nilai Akhir UAT

Metode ini digunakan untuk mengukur sejauh mana sistem yang dikembangkan telah sesuai dengan kebutuhan, harapan, dan kriteria penerimaan pengguna akhir (Yakub et al., 2024). Dengan melibatkan pengguna secara langsung dalam proses pengujian, UAT dapat memberikan gambaran mengenai kualitas, kinerja, serta kesesuaian fitur sistem. Selain itu, hasil pengujian melalui UAT dapat menjadi dasar evaluasi untuk melakukan penyempurnaan dan peningkatan terhadap sistem agar lebih optimal dalam mendukung kebutuhan pengguna (Hermansah et al., 2025).

2.1.12 Teknologi Pendukung Sistem (Opsional dan Kontekstual)

Tools perangkat lunak adalah seperangkat aplikasi atau perangkat lunak pendukung yang digunakan dalam proses perancangan, pengembangan, pengujian, dan implementasi sistem. *Tools* ini berfungsi untuk membantu pengembang dalam meningkatkan efisiensi, akurasi,

serta kualitas sistem yang dibangun (Firmansyah et al., 2024). Dalam penelitian ini, tools perangkat lunak digunakan untuk mendukung berbagai aktivitas, mulai dari pemodelan sistem, pengolahan data, hingga pengujian aplikasi, sehingga proses pengembangan sistem dapat dilakukan secara lebih terstruktur dan sistematis.

1. *Website*

Website merupakan media penyajian informasi yang disampaikan melalui jaringan internet, sehingga dapat diakses dari berbagai lokasi selama terhubung dengan koneksi internet. Sementara itu, pengembangan sistem informasi adalah serangkaian kegiatan atau tahapan yang dilakukan untuk merancang, membangun, dan menyempurnakan suatu sistem agar dapat berfungsi sesuai dengan kebutuhan pengguna (Rahmi et al., 2023).

2. HTML

Hypertext Markup Language (HTML) merupakan bahasa pemrograman yang digunakan untuk mengatur cara penyajian informasi di internet serta memungkinkan pengguna berpindah dari satu halaman ke halaman lain melalui tautan. HTML dikembangkan oleh Tim Berners-Lee saat bekerja di CERN dan pertama kali dipopulerkan melalui peramban Mosaic. Sejak awal tahun 1990-an, HTML mengalami perkembangan yang sangat pesat, di mana setiap versi pembaruannya menghadirkan kemampuan dan fitur yang lebih baik dibandingkan versi sebelumnya (Rina Noviana, 2022).

3. CSS

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman yang berfungsi untuk mengatur serta memperindah tampilan atau tata letak halaman web agar terlihat lebih menarik dan elegan. CSS bersifat mandiri dan dapat ditanamkan langsung ke dalam kode HTML atau digunakan sebagai berkas terpisah yang dijadikan acuan dalam pengaturan gaya tampilan (Rina Noviana, 2022).

Dengan CSS, pengembang dapat mengatur berbagai aspek visual seperti warna teks, jenis huruf, jarak antarparagraf, ukuran kolom, hingga latar belakang halaman. Selain itu, CSS juga memungkinkan perancangan layout, penyesuaian tampilan pada berbagai jenis perangkat, serta penerapan beragam efek visual pada website. Meskipun relatif mudah dipelajari, CSS memiliki kemampuan yang kuat dalam mengendalikan tampilan dokumen HTML, baik untuk desain sederhana maupun kompleks. Oleh karena itu, CSS hampir selalu digunakan dalam pengembangan website dan dikombinasikan dengan HTML maupun PHP (Rina Noviana, 2022).

4. PHP

Hypertext Preprocessor (PHP). Merupakan bahasa skrip yang terintegrasi dengan HTML dan dijalankan pada sisi server (*server-side HTML embedded scripting*). PHP digunakan untuk membangun halaman website yang bersifat dinamis, sehingga konten yang ditampilkan dapat berubah sesuai dengan proses, data, atau interaksi pengguna (Sitanggang et al., 2022).

Kelebihan PHP menurut (Sitanggang et al., 2022) adalah sebagai berikut:

- i. Bersifat multiplatform, dapat berjalan di berbagai sistem operasi seperti Linux, Unix, Macintosh, dan Windows.
- ii. Merupakan bahasa *open source* sehingga dapat digunakan secara gratis.
- iii. Didukung oleh banyak web server seperti Apache, IIS, Lighttpd, Nginx, dan Xitami dengan konfigurasi yang relatif mudah.
- iv. Mudah dikembangkan karena didukung oleh banyak komunitas dan developer.
- v. Relatif mudah dipahami karena memiliki dokumentasi dan referensi yang sangat banyak.

- vi. Tersedia banyak aplikasi dan program PHP gratis serta siap pakai, seperti WordPress dan Prestashop.
- vii. Mendukung berbagai jenis *database*, seperti MySQL, Oracle, dan MS-SQL.

5. *Laragon*

Laragon merupakan solusi pengembangan *web* yang menyediakan lingkungan server lokal yang lengkap, ringan, fleksibel, dan portabel sehingga memudahkan pengembang dalam membangun dan mengelola aplikasi berbasis *web* (Yurita Fanani et al., 2023). Perangkat lunak ini telah terintegrasi dengan berbagai komponen utama seperti *Apache*, *MySQL* atau *MariaDB*, serta *PHP*, sehingga pengembang tidak perlu melakukan instalasi secara terpisah. Dengan konfigurasi yang sederhana dan kecepatan eksekusi yang tinggi, *Laragon* berperan penting dalam mendukung proses pengujian aplikasi berbasis *web* secara lokal sebelum dipublikasikan ke server utama. Selain itu, *Laragon* menyediakan lingkungan pengembangan yang stabil dan efisien sehingga dapat mempercepat proses pembangunan sistem informasi (Ni'mah & Bhakti, 2025).

6. *MySQL*

MySQL merupakan sistem manajemen basis data SQL yang bersifat *open source* dan termasuk yang paling populer saat ini. MySQL mendukung berbagai fitur seperti *multithreaded*, *multi-user*, serta berfungsi sebagai *Database Management System* (DBMS). Basis data ini dirancang untuk menghasilkan kinerja yang cepat, andal, dan mudah digunakan. Selain itu, MySQL merupakan database multiuser yang menggunakan bahasa *Structured Query Language* (SQL) dalam pengelolaan datanya (Putra & Novembrianto, 2021; Tumini & Fitria, 2021).

7. *Bootstrap*

Bootstrap adalah sebuah framework CSS yang paling banyak diminati oleh para pengembang website. Dengan memakai bootstrap kita dengan mudah dapat mendesign tampilan website yang responsif. Responsif maksudnya, lebar halaman website akan disesuaikan secara otomatis berdasarkan perangkat yang digunakan untuk mengaksesnya, baik itu ketika diakses menggunakan PC, Laptop, Tablet ataupun Smartphone maka website akan menyesuaikan dengan lebar perangkat yang di gunakan pengunjung (Anis et al., 2022; Khairina et al., 2022).

8. *Laravel*

Laravel merupakan framework aplikasi web modern yang bersifat *open source* dan banyak dimanfaatkan dalam pengembangan aplikasi web secara cepat dan efisien. Laravel dilengkapi dengan berbagai fitur modern yang mendukung pengembang dalam membangun aplikasi, serta memiliki keunggulan pada penulisan kode yang lebih ringkas, mudah dipahami, dan bersifat ekspresif. Dengan kelebihan tersebut, framework Laravel sangat sesuai digunakan dalam pengembangan sistem informasi, termasuk sistem informasi penggajian pada perusahaan (Herdiyatomoko, 2022; Melyani & Aji, 2023).

9. *Visual Studio Code*

Visual Studio Code merupakan perangkat lunak editor kode sumber yang ringan namun memiliki kemampuan yang sangat kuat dan berjalan pada lingkungan desktop. Aplikasi ini digunakan untuk membantu proses penulisan dan pengelolaan kode program secara efektif (Dewi et al., 2021). Visual Studio Code mendukung berbagai bahasa pemrograman, seperti JavaScript, HTML, CSS, PHP, Python, C++, dan lainnya. Selain itu, Visual Studio Code dapat dijalankan pada berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Visual Studio Code juga dilengkapi dengan fitur

Live Share yang memungkinkan beberapa pengembang bekerja secara bersamaan pada satu proyek yang sama meskipun berada di lokasi yang berbeda (Nanda Syarif et al., 2023).

10. *Figma*

Figma merupakan salah satu *tools* atau aplikasi desain yang dapat digunakan pada sistem operasi Windows untuk membuat prototipe aplikasi serta berbagai kebutuhan desain lainnya. *Figma* umumnya dimanfaatkan oleh profesional di bidang UI/UX, *web design*, dan bidang sejenis. UI dan UX merupakan singkatan dari *User Interface* dan *User Experience*, yaitu konsep yang berkaitan dengan tampilan visual dan pengalaman pengguna dalam sebuah aplikasi atau media pemasaran digital berbasis website, yang berperan penting dalam meningkatkan citra dan nilai merek suatu bisnis atau perusahaan (Sari et al., 2022; Suparman et al., 2023).

11. *Draw.io*

Draw.io merupakan sebuah layanan berbasis website yang dirancang khusus untuk membuat diagram UML secara daring. Aplikasi ini memiliki tampilan yang responsif serta terintegrasi dengan layanan penyimpanan Google Drive, sehingga memudahkan pengguna dalam menyimpan dan mengelola file. Dengan kemudahan tersebut, *Draw.io* menjadi salah satu alternatif yang efektif dalam pembuatan diagram UML karena dapat mempercepat proses perancangan (Hadisman et al., 2024; Noneng Marthiawati et al., 2024).

2.2 Penelitian Sebelumnya (*Previous Work*)

Berdasarkan penelitian-penelitian terdahulu yang relevan dengan penelitian yang dilakukan oleh penyusunan saat ini.

Ditemukan alternatif solusi penyelesaian permasalahan dengan menggunakan acuan sebagai berikut :

Tabel 2. 5 Penelitian Sebelumnya

No	Judul, Penulis, Tahun	Masalah Penelitian	Tujuan Penelitian	Metodologi Penelitian / Pemecahan Masalah	Jumlah Data / Sampel	Hasil Penelitian / Temuan Utama	Limitasi / Keterbatasan	Persamaan & Perbedaan dengan Penelitian Saat Ini
1	Sistem Pendukung Keputusan untuk pemilihan bibit semangka <i>Metode Fuzzy Tsukamoto</i> (Sihombing, 2020)	Proses penentuan bibit masih manual dan subjektif	Menghasilkan sistem yang membantu penentuan bibit secara objektif	Fuzzy Tsukamoto, SPK berbasis desktop	5 data bibit	Sistem mampu memberikan rekomendasi ranking bibit (kualitas) berdasarkan kriteria yang ditetapkan	Sistem belum berbasis web dan belum terintegrasi antar pengguna	Persamaan: menggunakan logika Fuzzy Perbedaan: penelitian ini berbasis web dan fokus pada CPCL
2	Penerapan Metode Fuzzy Multi Criteria Decision	Kesulitan menentukan daerah kategori miskin karena banyak kriteria	menentukan daerah yang berada dalam kategori miskin. Alternatif	FMCDM, Fuzzy	dari tahun 2017 sampai tahun 2021.	Metode FMCDM efektif dalam pemeringkatan	Tidak dikembangkan dalam	Persamaan: pendekatan multi-kriteria dan fuzzy. Perbedaan: fokus

No	Judul, Penulis, Tahun	Masalah Penelitian	Tujuan Penelitian	Metodologi Penelitian / Pemecahan Masalah	Jumlah Data / Sampel	Hasil Penelitian / Temuan Utama	Limitasi / Keterbatasan	Persamaan & Perbedaan dengan Penelitian Saat Ini
	Making pada Interpretasi Hasil Penentuan Kemiskinan Provinsi Jawa Timur Hakimatul (Maulidiyah & Turmudi, 2023)				Data tersebut merupakan data 38 kabupaten/kota Provinsi Jawa Timur yang diperoleh dari website Badan Pusat Statistika (BPS).		bentuk sistem web	wilayah, bukan CPCL
3	Sistem Pendukung Keputusan dalam Penentuan Beasiswa menggunakan	Proses seleksi penerima beasiswa tidak terintegrasi	Mengembangkan SPK untuk penentuan beasiswa secara objektif	Fuzzy Logic	4 data	Hasil perhitungan menentukan kelayakan penerima	Tidak menggunakan metode fuzzy, hanya logic	Persamaan: SPK Perbedaan: metode fuzzy tidak digunakan

No	Judul, Penulis, Tahun	Masalah Penelitian	Tujuan Penelitian	Metodologi Penelitian / Pemecahan Masalah	Jumlah Data / Sampel	Hasil Penelitian / Temuan Utama	Limitasi / Keterbatasan	Persamaan & Perbedaan dengan Penelitian Saat Ini
	n Metode Fuzzy Logic (Studi Kasus Universitas Peradaban) (Arjuna et al., 2021)					bantuan beasiswa		
4	Penerapan Metode Fuzzy Mamdani dan Metode Fuzzy Sugeno dalam Penentuan Jumlah Produksi (Studi Kasus: PJ Menara Kudus) (Muflihunna, K., &	Penentuan jumlah produksi jenang masih dilakukan secara manual sehingga berpotensi menimbulkan ketidaksesuaian antara permintaan, persediaan, dan bahan baku	Menerapkan metode Fuzzy Mamdani dan Fuzzy Sugeno serta membandingkan tingkat keakuratan kedua metode dalam menentukan jumlah produksi berdasarkan nilai MAPE	Logika fuzzy Mamdani dan Sugeno dengan tahapan fuzzyfikasi, pembentukan aturan (27 aturan), komposisi aturan, defuzzifikasi; evaluasi akurasi menggunakan MAPE dengan	Data bulanan Januari 2020–Maret 2021 (15 periode) untuk dua jenis produk: jenang kombinasi dan jenang wijen sehingga	Metode Fuzzy Sugeno menghasilkan nilai MAPE lebih kecil dibandingkan Fuzzy Mamdani. MAPE Sugeno sebesar 7,1% (jenang kombinasi) dan 8,2% (jenang wijen), sehingga lebih akurat	Variabel input terbatas pada permintaan, persediaan, dan bahan baku; objek penelitian hanya pada satu industri (PJ Menara Kudus); belum menambahkan variabel eksternal lain	Persamaan: Sama-sama menggunakan metode fuzzy (Mamdani/Sugeno) dan MAPE sebagai evaluasi akurasi. Perbedaan: Penelitian ini fokus pada penentuan jumlah produksi jenang, sedangkan penelitian saat ini diterapkan pada

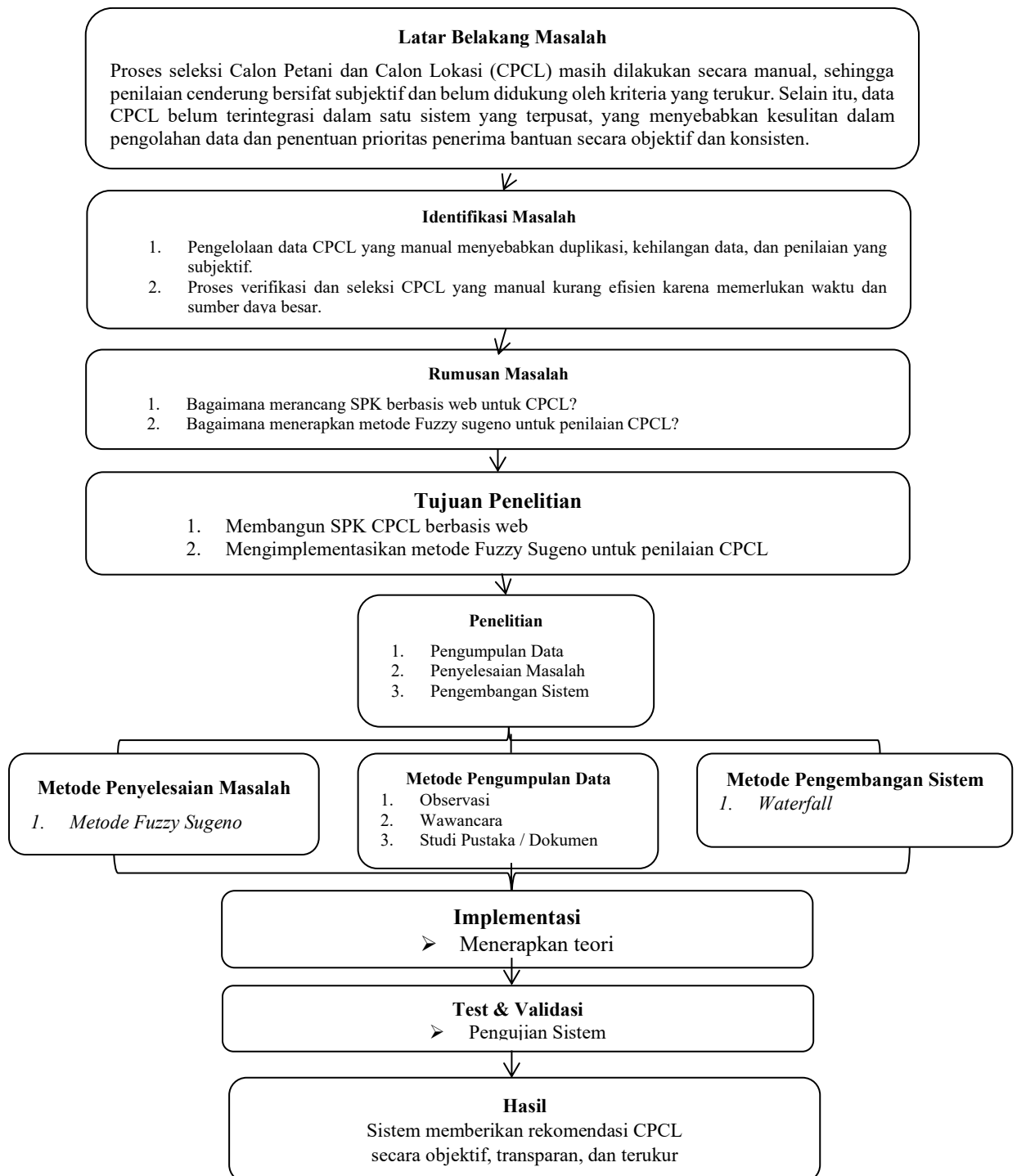
No	Judul, Penulis, Tahun	Masalah Penelitian	Tujuan Penelitian	Metodologi Penelitian / Pemecahan Masalah	Jumlah Data / Sampel	Hasil Penelitian / Temuan Utama	Limitasi / Keterbatasan	Persamaan & Perbedaan dengan Penelitian Saat Ini
	Mashuri (2022)			bantuan Matlab R2018b	lebih akurat			objek dan konteks sistem yang berbeda
5	Penerapan Logika Fuzzy Metode Sugeno Untuk Menentukan Jumlah Produksi Roti Ganda Berdasarkan Data Sedia Dan Jumlah Minta (Studi Kasus : Pabrik Roti Ganda Siantar) (efriska et.al 2024)	Menentukan jumlah produksi Roti.	Mengembangkan SPK untuk peramalan dalam menentukan jumlah produksi	Fuzzy Sugeno dengan fuzzifikasi, inferensi MIN, defuzzifikasi weighted average	1 bulan perimntaa n dan persediaan roti ganda	Metode Fuzzy Sugeno efektif dalam menilai kelayakan dengan output numerik yang presisi	Belum terintegrasi dalam sistem berbasis web	Persamaan: Metode Fuzzy Sugeno, konteks kelompok tani, kriteria kelayakan. Perbedaan: Penelitian ini mengembangkan sistem berbasis web dengan fokus CPCL dan skala prioritas
6	Sistem pendukung keputusan	Banyaknya pendaftar dan kriteria	Membangun Sistem Pendukung	Metode Fuzzy Sugeno Orde 0	Data mahasiswa AMIK	Metode fuzzy Sugeno dapat digunakan	Belum terintegrasi	Persamaan dengan penelitian ini adalah sama-sama

No	Judul, Penulis, Tahun	Masalah Penelitian	Tujuan Penelitian	Metodologi Penelitian / Pemecahan Masalah	Jumlah Data / Sampel	Hasil Penelitian / Temuan Utama	Limitasi / Keterbatasan	Persamaan & Perbedaan dengan Penelitian Saat Ini
	Dalam menentukan penerima beasiswa Dengan fuzzy sugeno Sundari Retno Andani, S.T, M.Kom (1), Subastian Wibowo, S.Kom (2)	beasiswa membuat diperlukan Sistem Pendukung Keputusan (SPK). Dengan metode fuzzy Sugeno, sistem ini membantu merekomendasi kan penerima beasiswa secara lebih cepat dan meningkatkan kualitas keputusan.	Keputusan penentuan penerima beasiswa untuk meningkatkan kualitas pemilihan berdasarkan kriteria		Tunas Bangsa Pematangs iantar yang merupaka n pemohon beasiswa (10 Data)	sebagai sistem pendukung keputusan untuk menentukan penerima beasiswa berdasarkan hasil analisis dan pengujian data, dengan nilai output yang ditentukan secara manual dalam rentang 0–100 (Girona, 2010).	dalam sistem berbasis web	menggunakan metode Fuzzy Sugeno sebagai sistem pendukung keputusan untuk menentukan kelayakan berdasarkan kriteria yang telah ditetapkan. Selain itu, keduanya bertujuan membantu proses seleksi agar lebih objektif dan efisien dalam menentukan prioritas penerima.

Berdasarkan hasil telaah terhadap penelitian-penelitian sebelumnya, dapat diketahui bahwa metode logika fuzzy telah banyak diterapkan dalam berbagai kasus sistem pendukung keputusan, seperti pemilihan bibit, penentuan tingkat kemiskinan, seleksi penerima beasiswa, hingga penentuan jumlah produksi. Namun demikian, sebagian besar penelitian tersebut masih memiliki beberapa keterbatasan, di antaranya belum dikembangkan dalam bentuk sistem berbasis web, jumlah data penelitian yang relatif terbatas, serta belum adanya penerapan pada konteks penentuan Calon Petani dan Calon Lokasi (CPCL). Selain itu, beberapa penelitian hanya berfokus pada aspek perhitungan dan pemeringkatan tanpa memperhatikan standarisasi kriteria yang digunakan dalam proses pengambilan keputusan. sebaiknya, dalam proses penentuan CPCL diperlukan mekanisme evaluasi yang objektif, transparan, dan mampu mengakomodasi berbagai kriteria yang telah ditetapkan oleh instansi terkait. Keterbatasan tersebut menunjukkan bahwa masih terdapat peluang untuk mengembangkan sistem pendukung keputusan yang lebih adaptif, terintegrasi, dan sesuai dengan kebutuhan penentuan CPCL.

2.3 Kerangka Teoritis (*Theoretical Framework*)

Kerangka teoritis berdasarkan pemikiran dari penelitian yang akan dilakukan oleh penulis adalah sebagaimana digambarkan di bawah ini :



Gambar 2 9 Kerangka Teoritis.