

BAB II

LANDASAN

TEORI

2.1 Tinjauan Penelitian Sebelumnya

Berikut beberapa ulasan penelitian yang terkait dan menjadi inspirasi pada penelitian ini:

Alfaiz, M., & Wismarini, S.Kom., M.Cs, T. (2019). Dalam penelitiannya yang berjudul "PENGUNAAN METODE FUZZY SAW (SIMPLE ADDITIVE WEIGHTING) DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN BONUS KARYAWAN (STUDI KASUS : CV. NUGRAHA CITRA DIRGAHAYU SEMARANG)". Penulis dalam penelitian ini menjelaskan bahwa Sistem Pendukung Keputusan (SPK) ini dirancang untuk membantu departemen personalia dalam membuat keputusan yang lebih mudah, lebih efisien, dan lebih objektif terkait bonus karyawan, menurut penulis studi. SPK ini menggunakan teknik Fuzzy SAW (Simple Additive Weighting) dan Fuzzy Multi Attribute Decision Making (Fuzzy MADM) untuk mencapai hal ini.

Charolina, Yanthi (2016) dalam penelitiannya yang berjudul "SISTEM PENDUKUNG KEPUTUSAN UNTUK MENENTUKAN PEMBERIAN BONUS TAHUNAN MENGGUNAKAN METODE FUZZY LOGIC TIPE MAMDANI (Studi Kasus Pada Karyawan PT. Sunhope Indonesia Di Jakarta)". Hasil penelitian ini, yang memperhitungkan faktor-faktor seperti ketekunan, masa kerja, pendidikan, dan kehadiran, merupakan penentu bonus tahunan karyawan, menurut penelitian

tersebut. Fuzzy Toolbox Matlab digunakan untuk membangun model sistem. Proses seleksi pemberian bonus PT. Sunhope Indonesia dikaji secara menyeluruh untuk melakukan eksperimen pemodelan sistem. Untuk memperkirakan pembayaran bonus tahunan karyawan, penelitian ini berhasil mengembangkan sistem pemodelan inferensi fuzzy berdasarkan teknik Mamdani.

Murdianto, Handri (2016) dalam penelitiannya yang berjudul "SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KARYAWAN TERBAIK PER TRIWULAN PT.CAHAYA FAJAR KALTIM PLTU EMBALUT TANJUNG BATU MENGGUNAKAN METODE SIMPLE ADDITIVE WEIGHTING". Penulis menjelaskan bahwa Untuk mempercepat proses, penulis mengusulkan sistem pendukung keputusan (SPK) untuk menghitung semua faktor yang memengaruhi keputusan. Pembobotan aditif sederhana digunakan untuk menilai atribut spesifik perusahaan seperti pengetahuan, inisiatif, produktivitas, komunikasi, kerja sama tim, akuntabilitas, dan kehadiran. Sistem akhir menggunakan pendekatan Pembobotan Aditif Sederhana (SAW) untuk menentukan nilai dalam memilih dan merekomendasikan karyawan terbaik setiap kuartal.

2.2 Clustering

Pengelompokan, terkadang disebut klasifikasi, adalah metode pengorganisasian data menurut kesamaan yang dicatat sebelumnya, menurut Metisen dan Sari (2015). Kumpulan titik data yang berbeda satu sama lain namun memiliki beberapa karakteristik disebut kluster. Jika suatu objek memiliki karakteristik tertentu dibandingkan objek lain dalam kluster, objek tersebut akan dimasukkan ke dalam satu atau lebih kluster..

2.3.1 Karakteristik Clustering

Berbagai teknik pengelompokan telah diciptakan oleh para ahli. Setiap strategi memiliki karakteristik, manfaat, dan kekurangan yang unik. Pengelompokan mencakup berbagai kualitas unik, termasuk struktur data, kekompakan, dan keanggotaan klaster.

Berdasarkan strukturnya, algoritma pengelompokan dapat dibagi menjadi dua kategori: partisional dan hierarkis. Dengan mempertimbangkan setiap item sebagai kelompok yang berbeda, menggabungkan dua atau lebih kelompok yang lebih kecil menjadi satu kelompok yang lebih besar, dan seterusnya, pengelompokan hierarkis menyatukan data. Hanya ada satu teknik dalam kategori pengelompokan hierarkis. Setelah membagi dan mengelompokkan suatu koleksi ke dalam kelompok-kelompok yang tidak tumpang tindih, setiap titik data dimasukkan ke dalam salah satu kelompok tersebut. K-Means adalah salah satu teknik partisi dan pengelompokan.

Bergantung pada keanggotaan grup, teknik pengelompokan dapat diklasifikasikan sebagai eksklusif atau tumpang tindih. Suatu pendekatan dianggap eksklusif ketika suatu data sepenuhnya termasuk dalam satu grup tetapi tidak termasuk dalam grup lain. Teknik pengelompokan seperti K-Means dan DBSCAN, serta pendekatan tumpang tindih seperti logika fuzzy, dapat digunakan untuk mengklasifikasikan data dalam hal ini.

Klasifikasi strategi pengelompokan sebagai parsial atau lengkap bergantung pada tingkat kekompakannya. Informasi yang dapat dikelompokkan ke dalam satu klaster disebut sebagai data kompak. Data kompak dianggap tidak umum meskipun hanya sebagian kecil data yang tidak sesuai dengan kelompok mayoritas. Kami menyebut data tak terduga ini sebagai "noise".

2.3 Teorema *K-Means*

2.3.1 Clustering *K-Means*

Data dikategorikan berdasarkan fitur numerik menggunakan algoritma pengelompokan berbasis jarak *K-Means*. Data dibagi menjadi k bagian berbeda menggunakan teknik pengelompokan *K-Means*. Teknik *K-Means* dapat mengklasifikasikan kumpulan data besar dengan cepat dan efektif, termasuk outlier. Teknik *K-Means* memungkinkan satu titik data tetap berada di satu kluster sebelum berpindah ke kluster berikutnya, meskipun setiap titik data harus berada di klusternya sendiri.

Untuk mengklasifikasikan data yang masuk, kami menggunakan pendekatan pengelompokan data *K-Means* non-hierarkis. Data dengan fitur serupa (kemiripan intra-kelas yang tinggi) dan data dengan fitur berbeda (kemiripan antar-kelas yang rendah) digabungkan dalam pendekatan ini.

Pengelompokan dimulai dengan menentukan apakah data, X_{ij} ($i=1, \dots, n$; $j=1, \dots, m$), terkluster. Di sini, n adalah jumlah data yang akan dikluster, dan m adalah jumlah variabel. Pada awal setiap siklus, pusat setiap kluster (C_{kj}) dipilih secara acak. Jarak antara setiap titik data dan pusat kluster kemudian dihitung. Dengan menggunakan pendekatan Euclidean, perangkat lunak berikut menghitung jarak antara titik data ke- i (x_i) dan pusat kluster ke- k (d_{ik}).

$$d_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - c_{kj})^2}$$

Jika jarak suatu titik data dari pusat kluster adalah yang terpendek dari semua jarak ke pusat kluster lainnya, titik tersebut dialokasikan ke kluster ke-k. Hal ini dapat dihitung menggunakan prosedur yang diuraikan di bawah ini. Pengelompokan data merupakan tahap selanjutnya.

$$d_{dd} = \sqrt{\sum_{d=1}^d (x_{dd} - y_{dd})^2}$$

Rumus persamaan dan rata-rata data kluster dapat digunakan untuk menentukan pusat kluster baru.

$$x_{kk} = \frac{\sum_{i=1}^p x_{ki}}{p}$$

Dimana $x_{ki} \in$ cluster ke - k

p = banyaknya anggota cluster ke k

2.3.2 Tujuan Clustering K-Means

Pengelompokan data meningkatkan pemahaman dan penggunaan. Jika tujuannya adalah pemahaman, kategori-kategori harus mencerminkan struktur dasar teks. Oleh karena itu, pengelompokan seringkali merupakan tahap pertama, diikuti oleh tugas-tugas sederhana seperti memberi label pada setiap kelompok untuk data pelatihan klasifikasi, meringkas (rata-rata dan SD), dan sebagainya. Sebaliknya,

pengelompokan memilih kelompok data prototipe yang paling representatif untuk mengabstraksikan semua item data dalam suatu koleksi. M.A.W.K. Murti (2017).

Contoh – contoh tujuan pengelompokan untuk pemahaman adalah sebagai berikut:

a. **Biologi**

Sebagaimana diketahui seluruh masyarakat, hewan di alam terbagi menjadi kingdom, filum, kelas, ordo, famili, genera, dan spesies. Kingdom merupakan tingkatan tertinggi, sedangkan spesies merupakan tingkatan terendah. Setiap hewan memiliki julukan yang unik untuk spesiesnya. Genus yang sama dapat berisi dua spesies organisme yang berbeda. Satu famili dapat berisi beberapa spesies dari berbagai genera. Hal ini juga berlaku untuk kingdom, filum, ordo, dan kelas. Semua makhluk diklasifikasikan sebagai hewan pada tingkat kingdom. Cara lain untuk mengkategorikan secara biologis adalah dengan mengelompokkan gen-gen dengan fungsi yang serupa.

b. ***Information retrieval***

Internet terdiri dari miliaran halaman. Mesin pencari menghasilkan ratusan hasil saat Anda mengetikkan sebuah kueri. Hasil mesin pencari dapat dibagi menjadi beberapa kelompok yang lebih kecil, masing-masing berisi halaman web dengan atribut yang serupa, dengan menggunakan algoritma pengelompokan. Kata "film" dapat digunakan untuk mengurutkan hasil situs web berdasarkan "genre", "bintang", "bioskop", dan kriteria lainnya. Subkategori kemudian ditambahkan ke setiap kategori, memberikan pengguna dasar yang mudah untuk menampilkan hasil pencarian.

c. **Klimatologi**

Memantau perubahan di laut dan atmosfer sangat penting untuk memahami cuaca Bumi. Analisis kluster dapat membantu menemukan perbedaan tekanan udara di lokasi kutub dan laut yang secara signifikan memengaruhi cuaca daratan.

d. **Bisnis**

Bisnis seringkali memiliki banyak informasi tentang klien mereka saat ini dan calon klien. Anda dapat menggunakan pengelompokan untuk mengelompokkan klien Anda untuk tujuan pemasaran dan riset.

Contoh – contoh tujuan pengelompokan untuk penggunaan adalah sebagai berikut:

a. **Peringkasan (*Summarization*)**

Regresi dan PCA adalah dua teknik analisis data yang memiliki kompleksitas komputasi dan/atau temporal $O(m^2)$. Biaya peringkasan meningkat seiring dengan jumlah data. Untuk membuat prototipe yang secara akurat menangkap keseluruhan dataset, teknik pengelompokan data dapat diterapkan. Rata-rata semua data dalam setiap kategori dapat digunakan untuk merepresentasikan satu set data. Akibatnya, kompleksitas dan waktu pemrosesan data telah berkurang.

b. **Kompresi**

Jika indeks prototipe untuk setiap grup terkini dan datanya sama atau serupa, maka data dapat dikompresi. Untuk setiap grup, indeks prototipe digunakan untuk merepresentasikan objek. Kuantisasi vektor adalah nama yang diberikan untuk metode kompresi ini.

c. **Pencarian tetangga terdekat secara efisien**

Seiring bertambahnya jumlah data, pendekatan K-NN untuk menentukan tetangga terdekat membutuhkan sumber daya komputasi yang semakin besar. Di sisi

lain, terlalu banyak informasi yang dikumpulkan mengenai tetangga terdekat. Pengelompokan memungkinkan kita membuat prototipe yang hanya mewakili satu kelas. Menemukan tetangga terdekat mungkin lebih mudah jika prototipe terdekat digunakan. Dengan cara ini, waktu perhitungan dapat dikurangi secara signifikan. Pendekatan tetangga terdekat dapat menghasilkan hasil yang kurang representatif dan bias yang tidak disengaja karena berfokus pada pengelompokan data, alih-alih titik data individual.

2.3.3 Langkah *Clustering K-Means*

Metode pengelompokan K-Means terdiri dari langkah-langkah berikut: Aulia S. (2020).

1. Tentukan jumlah kluster (k) yang perlu dibentuk.
2. Ada beberapa pendekatan untuk menginisialisasi k pusat kluster. Namun, strategi acak adalah yang paling sering digunakan. Pusat kluster dialokasikan secara acak ke dalam nilai awal.
3. Untuk setiap item atau bagian data, pilih kluster dengan korelasi tertinggi. Jarak dua objek menunjukkan seberapa dekat keduanya. Demikian pula, jarak dari pusat kluster menentukan seberapa dekat data dengan kluster tersebut. Tujuan dari tahap ini adalah menemukan jarak antara setiap titik data dan setiap pusat kluster. Jarak antara dua titik data menentukan kluster mana yang mereka ikuti. Teori jarak Euclidean dapat digunakan untuk menentukan jarak antara setiap pusat kluster dan setiap titik data.
4. Dengan menggunakan keanggotaan kluster saat ini, hitung ulang pusat kluster. Rata-rata semua informasi atau objek dalam kluster tertentu diwakili oleh pusat kluster. Anda juga dapat menggunakan

median kluster jika diinginkan. Oleh karena itu, angka-angka tertentu lebih penting daripada rata-rata.

5. Pusat kluster yang baru seharusnya menampung semuanya. Proses pengelompokan selesai ketika pusat kluster berhenti bergerak. Langkah 3 dapat diselesaikan tanpa memengaruhi inti kluster.

2.3.4 Jenis Data Dalam Set Data

Titik-titik data yang diperoleh digabungkan untuk membentuk set data. Rekaman, titik, vektor, pola, peristiwa, kasus, sampel, observasi, dan entitas adalah sebutan lain untuk item data. Kumpulan atribut yang secara kolektif menyampaikan karakteristik fundamental suatu entitas, seperti massanya atau waktu terjadinya suatu peristiwa, disebut objek data. Variabel, karakteristik, bidang, fitur, dan dimensi adalah sebutan lain untuk atribut.

Properti atau fitur suatu item data yang nilainya bervariasi seiring waktu dan antar objek disebut atribut. Bobot dapat berubah seiring waktu. Meskipun warna kulit (hitam, putih, kuning, coklat muda, atau sawo matang) mungkin memiliki makna simbolis, bobot dapat dipastikan.

Titik-titik data individual digabungkan untuk membentuk set data. Rekaman, titik, vektor, pola, peristiwa, kasus, sampel, observasi, dan entitas adalah sebutan lain untuk item data. Seperangkat properti yang merepresentasikan karakteristik fundamental suatu item, seperti massanya atau waktu terjadinya suatu peristiwa, disebut objek data. Variabel, karakteristik, bidang, fitur, dan dimensi adalah sebutan lain untuk atribut.

2.4 Metodologi Fuzzy

Proses penciptaan perangkat cerdas menggunakan "komputasi lunak" semakin maju. Sistem cerdas terkadang dapat mengungguli manusia karena kemampuannya untuk belajar dan beradaptasi dengan perubahan keadaan. Komputasi lunak didasarkan pada empat pilar utama: komputasi evolusioner, penalaran probabilistik, sistem fuzzy, dan jaringan saraf tiruan. Penalaran sistem fuzzy biasanya melibatkan lima langkah:

1. Memasukkan input fuzzy.
2. Mengaplikasikan operator fuzzy.
3. Mengaplikasikan metode implikasi.
4. Komposisi semua output.
5. Defuzifikasi.

Teknik yang tepat untuk mengubah ruang masukan menjadi ruang keluaran adalah logika fuzzy. Logika fuzzy dapat diterapkan pada permasalahan dengan sistem yang terlalu kompleks. Mengendalikan satu keluaran yang dihasilkan dari sekelompok masukan yang terpisah merupakan tujuan sistem tradisional. Kemandirian ini membuat kontrol menjadi lebih sulit karena setiap fungsi perlu dihitung setiap kali masukan baru diberikan. Fungsi keanggotaan dan aturan baru memungkinkan sistem fuzzy, yang dibangun berdasarkan konsep logika fuzzy, untuk menerima lebih banyak masukan.

Dalam hal penalaran aproksimasi, sistem fuzzy biasanya paling efektif, terutama dalam skenario yang sulit dikarakterisasi menggunakan model konvensional. Jika data masukan dan atribut sistem tidak akurat atau tidak presisi, pembuatan model matematika bisa jadi lebih sulit.

Salah satu keunggulan utama sistem fuzzy dibandingkan sistem konvensional adalah banyaknya aturan yang digunakan. Pengendali dapat membuat keputusan lebih mudah dalam sistem fuzzy dengan menggabungkan sejumlah besar nilai menjadi satu nilai derajat keanggotaan. Kemampuan sistem fuzzy untuk berpikir serupa dengan manusia merupakan manfaat lainnya. Umumnya, sistem fuzzy merespons masukan yang kualitatif, ambigu, atau tidak presisi. Penggunaan logika fuzzy dibenarkan karena sejumlah alasan.

1. Terdapat banyak pilihan yang tersedia dengan logika fuzzy.
2. Argumen yang ambigu dapat diterima olehnya.
3. Logika fuzzy merupakan konsep yang mudah dipahami. Penalaran fuzzy didasarkan pada matematika yang sederhana dan mudah dipahami.

4. Sistem nonlinier yang sangat kompleks dapat disimulasikan menggunakan logika fuzzy.
5. Karena logika fuzzy dapat langsung diterapkan dan dibangun berdasarkan pengetahuan pakar, logika fuzzy tidak memerlukan pelatihan.
6. Sistem kendali konvensional dapat ditingkatkan dengan logika fuzzy.
7. Bahasa alami berfungsi sebagai fondasi bagi logika fuzzy. Kusumadwi, Sri (2002:3).

L.A. Zadeh, seorang profesor di Berkeley, merancang sistem fuzzy pada tahun 1965. Sistem fuzzy adalah estimator numerik yang terstruktur secara dinamis. Strategi ini dapat menghasilkan sistem cerdas dalam kondisi yang tidak umum. Sistem ini menggunakan logika fuzzy untuk menghitung suatu fungsi. Logika fuzzy melibatkan berbagai langkah, termasuk penentuan himpunan fuzzy, teknik inferensi fuzzy, dan aturan IF-THEN.

Tiga metode, Tsukamoto, Sugeno, dan Mamdani, dapat digunakan untuk menggambarkan hasil logika fuzzy. Teknik Tsukamoto merepresentasikan setiap konsekuensi sebagai himpunan fuzzy dengan fungsi keanggotaan monoton. Predikat menghasilkan himpunan terpisah, z , yang berisi keluaran inferensi untuk setiap aturan. Kesimpulan akhir ditentukan dengan mengambil rata-rata terbobot. Kusumadewi Sri (2002, hlm. 108).

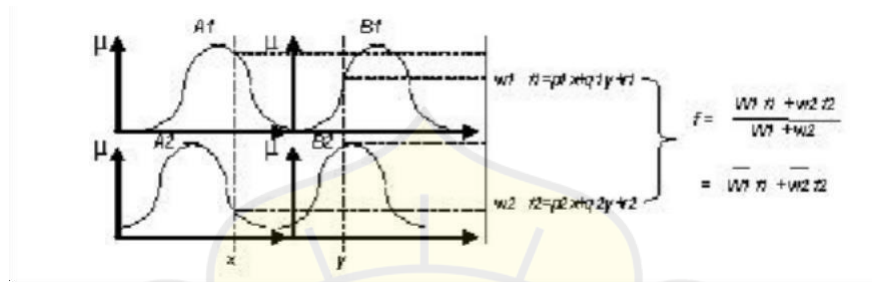
Mirip dengan metode Mamdani, proses Sugeno menghasilkan persamaan linear atau konstanta sebagai keluaran (konsekuensi), bukan himpunan fuzzy. Terdapat dua model dalam pendekatan Sugeno: orde nol dan orde pertama. Struktur umum model fuzzy Sugeno orde nol adalah sebagai berikut:

IF (x1 is A1) o (x2 is A2) o o (xn is An) THEN z = k

Salah satu bentuk model fuzzy Sugeno orde satu adalah:

IF (x_1 is A_1) *o* (x_2 is A_2) *o* *o* (x_n is A_n) *THEN* $z = p_1.x_1 + \dots p_n.x_n + q$

Untuk melakukan defuzifikasi metode Sugeno, nilai rata-rata dicari.



Gambar 2.16

Gambar 2.16 Model fuzzy sugeno orde 1

Berbeda dengan pendekatan komposisi aturan yang menggunakan metode MAX, metodologi Mamdani menggunakan metode MIN untuk menerapkan fungsi implikasi. Teknik MAX-MIN adalah nama lain untuk metodologi Mamdani. Hasilnya harus presisi dan diskrit karena keluaran inferensi berupa bilangan bulat fuzzy. Proses ini disebut defuzzifikasi. Produk akhir merupakan hasil dari beberapa proses:

1. Menciptakan himpunan fuzzy

Variabel input dan output dalam metode Mamdani dibagi menjadi satu atau lebih himpunan fuzzy.

2. Memanfaatkan fungsi implikasi

Fungsi implikasi metode Mamdani adalah Min.

3. Komposisi Aturan

Alih-alih menggunakan penalaran berulang, inferensi dalam sistem multi-aturan dicapai dengan mengumpulkan dan membandingkan aturan. OR probabilistik, OR aditif, dan OR maksimal semuanya dapat digunakan untuk mengidentifikasi sistem fuzzy.

- a. Metode Max (Maximum)

Pendekatan ini menghasilkan solusi himpunan fuzzy dengan memilih nilai aturan terbesar, mengubah domain fuzzy, dan menerapkannya pada keluaran menggunakan operator OR(union). Setiap konsep akan diperiksa, dan hasilnya akan diagregasi untuk menghasilkan himpunan fuzzy. Ini adalah metode pengucapan informal.

$$\mu_{sf}[x_i] \leftarrow \max (\mu_{sf}[x_i] , \mu_{kf}[x_i])$$

dengan :

$\mu_{sf}[xi]$ =nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen fuzzy aturan ke-i

b. Metode Additive (Sum)

Pendekatan ini menghasilkan solusi himpunan fuzzy dengan menggabungkan semua keluaran area fuzzy menjadi suatu penjumlahan terbatas. Biasanya, solusi ini ditulis sebagai:

$$\mu_{sf}[xi] \leftarrow \max (1, \mu_{sf}[xi] + \mu_{kf}[xi])$$

$\mu_{sf}[xi]$ =nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen fuzzy aturan ke-i

c. Metode Probabilistik OR

Untuk mendapatkan solusi himpunan fuzzy, pendekatan ini mengalikan setiap keluaran wilayah fuzzy dua kali. Biasanya ditulis sebagai:

$$\mu_{sf}[xi] \leftarrow \max (\mu_{sf}[xi] + \mu_{kf}[xi]) - (\mu_{sf}[xi] * \mu_{kf}[xi])$$

$\mu_{sf}[xi]$ =nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen fuzzy aturan ke-i

4. Penegasan /Defuzzifikasi

Himpunan fuzzy yang terdiri dari aturan-aturan fuzzy digunakan sebagai masukan dalam proses defuzzifikasi, dan bilangan keluaran berada dalam domain himpunan fuzzy tersebut. Oleh karena itu, keluaran harus presisi dan jelas mengingat himpunan fuzzy berada dalam rentang yang ditentukan. Defuzzifikasi menggunakan sejumlah teknik.

a. Metode Centroid.

Langkah ini menemukan titik pusat zona fuzzy untuk menghitung nilai renyah.

b. Metode Bisektor.

Dengan memilih nilai dalam domain fuzzy yang berkorelasi dengan nilai keanggotaan, misalnya, jumlah total nilai keanggotaan di wilayah fuzzy, metode ini menghasilkan respons yang tidak ambigu.

c. Metode Means of Maximum (MOM).

Pendekatan ini mencapai tujuan tertentu dengan menentukan nilai rata-rata domain dengan nilai keanggotaan tertinggi.

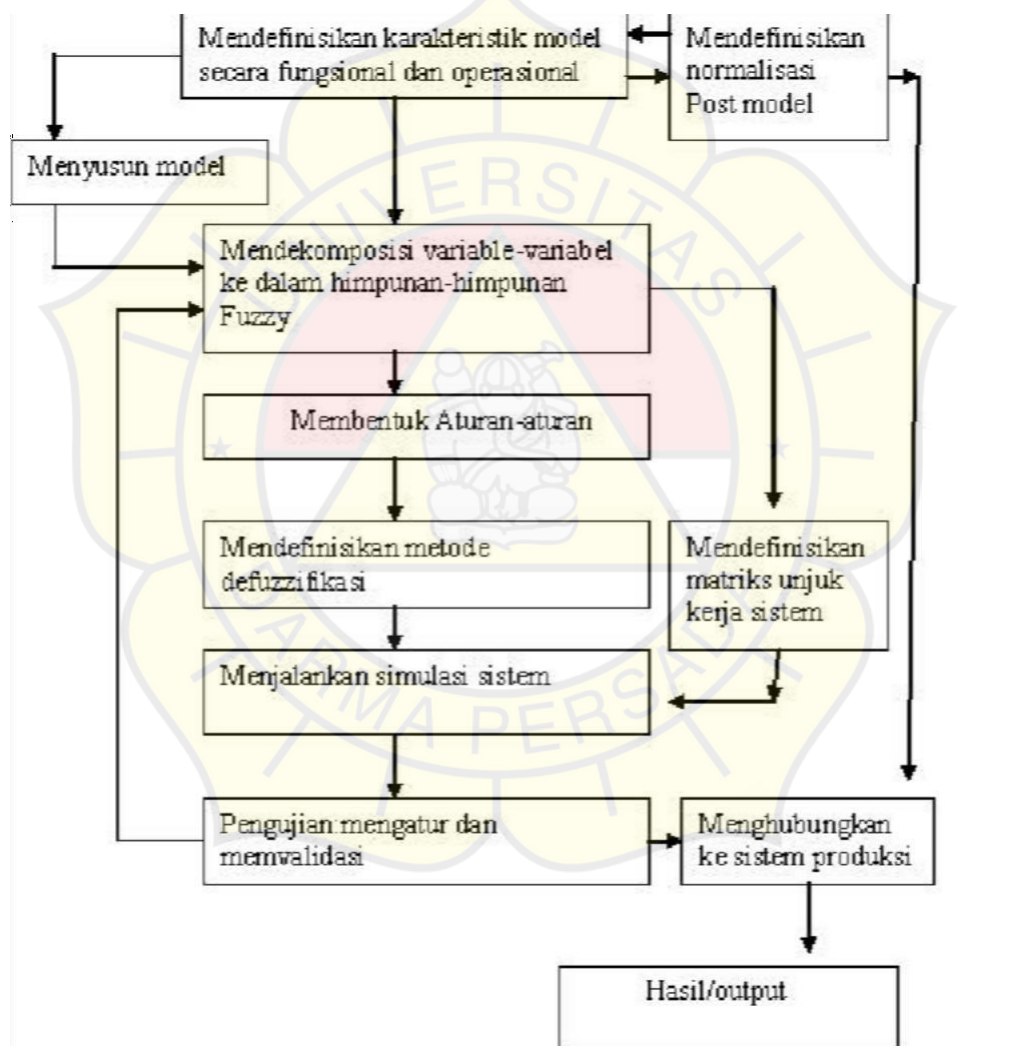
d. Metode Largest of Maximum (LOM)

Pendekatan ini memilih nilai terbesar dari domain dengan nilai keanggotaan tertinggi untuk mencapai hasil tertentu.

e. Metode Smallest of Maksimum (SOM).

Memilih nilai terendah dari domain dengan nilai keanggotaan tertinggi menghasilkan jawaban yang jelas.

Langkah Umum Pengembangan Model Fuzzy



Gambar 2.17

2.5 Pemrograman Aplikasi

2.5.1 HTML

Bahasa Markup Hiperteks disebut sebagai HTML. Bahasa komputer daring yang dikenal sebagai HTML menentukan bagaimana materi pada halaman web seharusnya disusun dan ditampilkan oleh peramban web. HTML mudah digunakan karena ditulis dalam bahasa Inggris yang sederhana. HTML saat ini memungkinkan teks, tabel, tautan, gambar, audio, dan video (Solichin, 2016).

2.5.2 CSS

Hal ini tercantum dalam buku "PHP Melengkapi Situs Web 30 Juta Rupiah!" karya YM Kusuma Ardhana yang terbit tahun 2015. Banyak komponen situs web dikelola oleh Cascading Style Sheet (CSS), yang memberikan tampilan yang lebih terpadu dan terorganisir. Halaman web HTML dan XHTML seringkali ditata dengan CSS.

Spasi teks dan paragraf, batas, mouseover, hyperlink, margin (atas, bawah, kiri, dan kanan), tabel, gambar, dan elemen lainnya semuanya dipengaruhi oleh CSS. Tampilan dokumen ditentukan oleh CSS, atau bahasa style sheet. Kita dapat menampilkan halaman yang sama dalam berbagai format berkat CSS.

2.5.3 JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi yang dinamis, menurut Wikipedia. Peramban web yang paling banyak digunakan yang mendukung JavaScript adalah Google Chrome, Mozilla Firefox, Opera, Netscape, dan Internet Explorer (IE). Anda dapat menambahkan kode JavaScript ke halaman web Anda menggunakan elemen SCRIPT. Seperti HTML dan CSS, JavaScript merupakan teknologi web yang esensial. Karena memungkinkan pembuatan halaman web interaktif, JavaScript merupakan bagian penting dari aplikasi daring.

Mesin JavaScript, yang sebelumnya hanya tersedia di sisi klien peramban web, kini dapat diakses sebagai lingkungan runtime. Dengan demikian, JavaScript dapat digunakan di sisi server dalam basis data, server internet, program desktop, aplikasi seluler, dan aplikasi non-web seperti pengolah kata dan pembaca PDF.

2.5.4 PHP

Akronim untuk Hypertext Preprocessor adalah PHP. PHP adalah skrip HTML, menurut Anhar (2010). PHP adalah salah satu bahasa pemrograman yang dapat digunakan di server. Akibatnya, server menjalankan PHP dan mengembalikannya ke peramban (Jubilee Enterprise 2017).

Skrip PHP dapat diubah menjadi skrip web menggunakan PHP Hypertext Preprocessor, yang sering dikenal sebagai PHP, sebuah bahasa pemrograman sisi server. PHP menawarkan tampilan sisi klien yang estetis. Rasmus Lerdoff menciptakan Form Interface, atau FI, pada tahun 1995, dan PHP merupakan salah satu penerapannya.



Berbeda dengan HTML, kode PHP dikelola oleh server, alih-alih dikirim langsung ke permintaan klien. Kode HTML dan PHP digabungkan. Berbeda dengan HTML, program PHP membutuhkan tag pembuka (<?php) dan tag penutup (?>) di akhir.

2.5.5 Bootstrap

Menurut Anhar (2018) dalam bukunya "SEO & Security Tricks for Codeigniter", Bootstrap adalah alat untuk mengembangkan tata letak halaman situs web yang dapat membantu pengembang dan desainer situs web bekerja lebih efisien. Situs web yang dibangun dengan teknologi ini, sesuai namanya, memiliki tata letak halaman yang mirip dengan Twitter, tetapi desainer dapat menyesuaikannya sesuai spesifikasi mereka.

Dengan menggabungkan berkas HTML siap pakai dengan teknik CSS (kelas), Twitter Bootstrap memudahkan pembuatan tata letak halaman situs web, tabel, tombol, formulir, navigasi, dan fitur lainnya. Komponen lain juga dapat dibuat dengan JavaScript.

2.6 Basis Data dan Mysql

2.6.1 Basis Data

Menurut buku "Structured and Object-Oriented Software Engineering" karya Rosa A. S. dan M. Shalahuddin (2016), sistem basis data adalah sistem terkomputerisasi yang menyimpan data atau informasi yang telah diproses dan menyediakannya saat dibutuhkan. Secara sederhana, basis data adalah lokasi tempat informasi dapat disimpan dan diakses dengan mudah.

2.6.2 DBMS

Dalam buku "Structured and Object-Oriented Software Engineering" yang

diterbitkan pada tahun 2016, Sistem manajemen basis data (DBMS) adalah sistem aplikasi untuk mengatur, menyimpan, dan menyajikan data, menurut Rosa A. S. dan M. Shalahuddin. Perangkat lunak yang memenuhi persyaratan ini mencakup sistem manajemen basis data (DBMS).

1. Menawarkan instrumen untuk mengelola akses data.
2. Mampu menjaga integritas data.
3. Mampu mengelola aksesibilitas data.
4. Mampu menangani pencadangan data.

2.6.3 Mysql

Menurut Mukhamad Masrur, 2016 dalam buku “Pemrograman Web Dinamis menggunakan Java Server Pages dengan Database Relasional Mysql”. Sistem manajemen basis data relasional sumber terbuka disebut MySQL. Arsitektur basis data dikelola dengan menggunakan tabel-tabel tertaut. Karena MySQL bersifat sumber terbuka, siapa pun dapat menggunakan dan mendistribusikannya tanpa izin dari pengembang aslinya. MySQL kompatibel dengan berbagai sistem operasi, termasuk Windows, Linux, Unix, dan Sun OS.

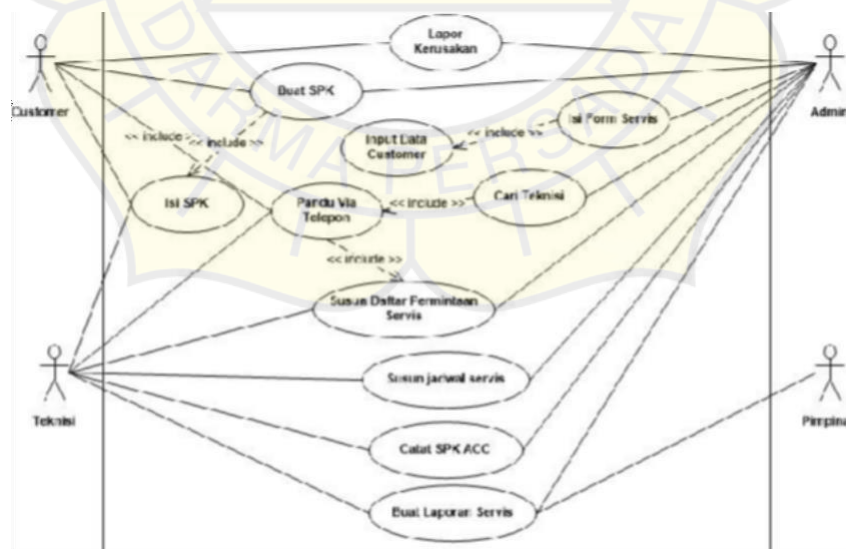
2.7 UML

Dalam bukunya "Object-Oriented Software Engineering", Verdi Yasin (2015) mengklaim bahwa UML (Unified Modeling Language) menyediakan penggambaran visual yang komprehensif dari sebuah model sistem. Sistem paling sering digunakan untuk meniru sistem komputer, namun juga dapat merepresentasikan informasi dan fungsi. Menggunakan kode yang ditulis dalam bahasa pemrograman untuk mendeskripsikan sistem berorientasi objek kepada orang lain dikenal sebagai

pemodelan objek, dan hal ini mungkin sulit. Ketidakpastian dan salah tafsir yang terjadi saat membaca kode pemrograman pemodelan objek merupakan akar permasalahannya.

2.7.1 Use Case Diagram

Berdasarkan buku "Object-Oriented Software Engineering" karya Verdi Yasin yang terbit tahun 2015. Diagram use case merupakan penggambaran beberapa atau semua aktor dan use case yang bertujuan untuk memastikan bagaimana mereka berinteraksi dengan sistem. Diagram use case memprioritaskan "apa" di atas "bagaimana" untuk menggambarkan fungsionalitas sistem yang diharapkan. Interaksi aktor dengan sistem ditunjukkan dengan use case. Use case mencakup hal-hal seperti administrasi pengguna dan autentikasi sistem. Dengan demikian, diagram use case dapat membantu analisis kebutuhan sistem. Istilah-istilah seperti aktor, use case, dan hubungan use case semuanya termasuk dalam diagram use case.


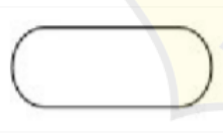




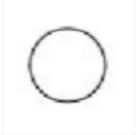
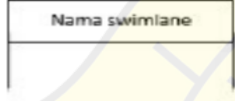


Gambar 2.1 Contoh Use Case Diagram menurut Verdi Yasin (2015)

2.7.2 Activity Diagram

Diagram aktivitas menunjukkan tindakan suatu sistem atau proses bisnis, serta menu aplikasi perangkat lunak. Penting untuk diingat bahwa diagram aktivitas menunjukkan kapabilitas sistem, bukan tindakan pemain. Struktur tampilan sistem atau antarmuka pengguna, di mana setiap aktivitas memiliki desain antarmukanya sendiri, dan pengembangan proses bisnis, di mana setiap rangkaian tindakan mencerminkan proses bisnis sistem tertentu, keduanya umumnya dijelaskan menggunakan diagram aktivitas. M. Shalahudin dan Rosa A. S. (2016)

Tabel 2.1. Activity Diagram Sumber: (Rosa A.S & M Shalahudin, 2016)


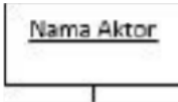
Simbol	Fungsi	Keterangan
	Status Akhir	Sebuah diagram aktivitas menunjukkan tingkat aktivitas sistem saat ini.
	Aktivitas	Dalam kebanyakan kasus, kata "kerja" muncul di awal setiap aktivitas yang dilakukan sistem.
	Percabangan / <i>Decicion</i>	Asosiasi percabangan di mana ada banyak aktivitas yang dapat dilakukan.


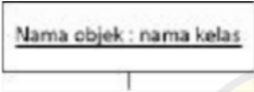
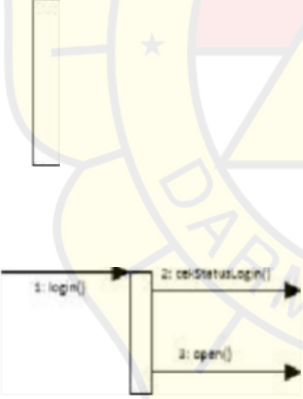


	Penggabungan / <i>Join</i>	Dalam kemitraan merger, beberapa kegiatan digabungkan menjadi satu.
	Status Akhir	Keadaan akhir diagram aktivitas menggambarkan kinerja sistem.
	<i>Swimlane</i>	Memisahkan entitas organisasi yang bertanggung jawab atas peristiwa tersebut.
	<i>Fork</i>	Digunakan untuk menunjukkan tindakan yang sedang dieksekusi secara bersamaan.
	<i>Join</i>	Digunakan untuk mewakili prosedur yang digabungkan.

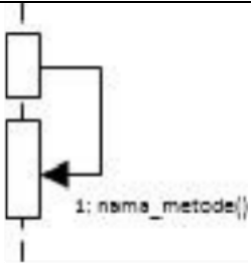

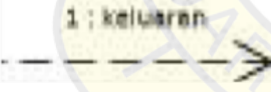
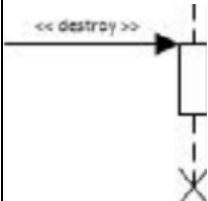
2.7.3 Sequence Diagram

Dengan mendokumentasikan siklus hidup suatu objek dan pesan yang dikirim dan diterimanya dari objek lain, suatu urutan menggambarkan bagaimana suatu objek harus bertindak dalam kasus penggunaan tertentu. Memahami objek dalam kasus penggunaan dan metode yang dapat diakses oleh kelas yang membuat objek tersebut merupakan prasyarat untuk membuat diagram sekuens. Untuk memahami keadaan use case, diagram sekuens juga diperlukan. Rosa A. S. dan M. Shalahudin (2016).

Tabel 2.2 *Sequence Diagram* Sumber: (Rosa A.S & M Shalahudin, 2016)

No	Simbol	Keterangan
1.	<p>Aktor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>Sebuah kata di awal nama aktor biasanya menekankan bahwa aktor tersebut belum tentu manusia, meskipun tandanya menunjukkan seseorang. Hal ini disebabkan oleh fakta bahwa sistem, orang, dan proses lain berinteraksi dengan sistem informasi yang sedang dibangun.</p>

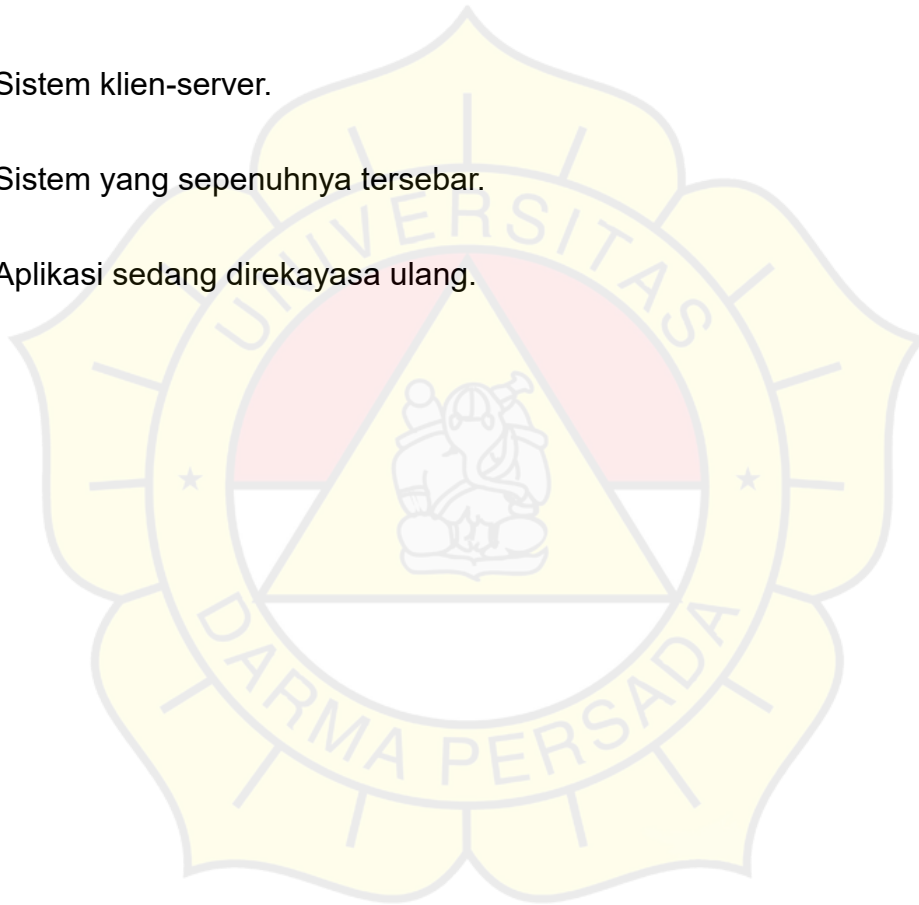
2.	<p>Garis hidup / <i>lifeline</i></p> 	Menunjukkan kehidupan sesuatu
3.	<p>Objek</p> 	Menunjukkan item yang berinteraksi dengan pesan
4.	<p>Waktu aktif</p> 	Ketika objek diatur agar aktif dan interaktif, semua hal yang terkait dengan periode aktif tersebut—seperti tahapan—terjadi di dalamnya. Akibatnya, baik open() maupun checkStatusLogin() dipanggil oleh fungsi login(). Tidak ada waktu luang bagi aktor.
5.	<p>Pesan tipe create</p> 	Panah menunjukkan ke arah objek yang baru dibuat saat satu hal berubah menjadi hal lain.
6.	<p>Pesan tipe call</p> 	Karena harus ada dalam diagram kelas berdasarkan kelas objek yang berinteraksi, panah menunjuk ke item yang berisi operasi atau metode yang sedang dieksekusi. Hal ini menunjukkan bahwa suatu objek memanggil operasi atau metode yang ada pada objek tersebut atau objek lain.

		
7.	<p>Pesan tipe send</p> 	<p>Panah menunjukkan item tempat data, masukan, atau informasi ditransfer.</p>
8.	<p>Pesan tipe return</p> 	<p>Panah yang menunjuk ke objek yang menerima pengembalian menunjukkan bahwa objek yang telah menjalankan operasi atau metode mengembalikan objek tertentu.</p>
9.	<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

2.8 Deployment Diagram


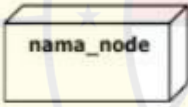


Diagram penerapan menunjukkan bagaimana komponen dikonfigurasi saat program dijalankan. Dengan diagram penerapan, hal-hal berikut dapat diilustrasikan:

- Desain perangkat keras, node, dan perangkat ditentukan oleh sistem tertanam.
- Sistem klien-server.
- Sistem yang sepenuhnya tersebar.
- Aplikasi sedang direkayasa ulang.



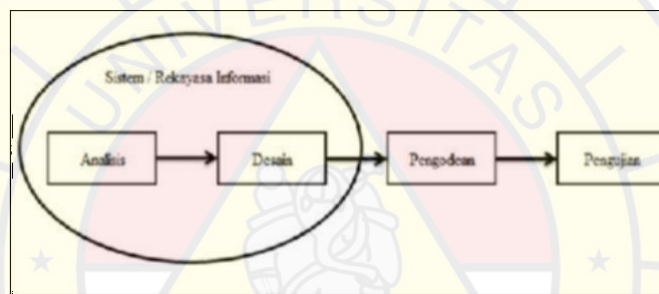
Untuk deployment diagram, simbol-simbol berikut digunakan:

Tabel 2.5 Simbol-simbol Deployment Diagram (Rosa A.S dan M. Shalahuddin, 2016)

Simbol	Deskripsi
<p>Package</p> 	<p>Paket terdiri dari satu atau lebih node.</p>
<p>Node</p> 	<p>Umumnya digunakan untuk menggambarkan perangkat lunak atau peralatan yang tidak dikembangkan secara internal. Komponen yang ditambahkan ke node harus sesuai dengan komponen yang ada di diagram komponen agar desain tetap koheren.</p>
<p>Kebergantungan / dependency</p> 	<p>Ketergantungan antara node: Arah panah menunjukkan node yang digunakan.</p>
<p>Link</p> 	<p>Relasi antar node.</p>

2.9 Metodologi Pengembangan Sistem (Waterfall)

Buku "Structured and Object-Oriented Software Engineering" karya A.S. Rosa dan Shalahuddin M. yang terbit tahun 2016 membahas paradigma SDLC waterfall, yang terkadang disebut sebagai model linier sekuensial atau siklus hidup klasik. Pengembangan perangkat lunak dapat dilakukan secara logis maupun berurutan menggunakan paradigma waterfall, dimulai dari analisis, desain, pengodean, pengujian, dan dukungan. Gambar di bawah ini menunjukkan model waterfall:



Gambar 2.2 Ilustrasi Model Waterfall

1. Analisis kebutuhan perangkat lunak

Untuk mengidentifikasi kebutuhan perangkat lunak dan membantu pelanggan mengidentifikasi jenis perangkat lunak yang mereka butuhkan, proses pengumpulan persyaratan dilakukan dengan cermat. Mendokumentasikan persyaratan perangkat lunak menjadi semakin penting.

2. Desain

Proses pengumpulan persyaratan yang lengkap digunakan untuk memeriksa kebutuhan perangkat lunak dan membantu klien mengidentifikasi jenis perangkat lunak yang dibutuhkan. Mendokumentasikan kebutuhan perangkat lunak menjadi

semakin penting.

3. Pembuatan Kode Program

Desain tersebut harus diubah menjadi program perangkat lunak. Program komputer yang sesuai dengan desain yang dibuat pada tahap desain akan diproduksi pada langkah ini.

4. Pengujian

Aspek logis dan fungsional program menjadi fokus utama pengujian, yang menjamin bahwa setiap komponen telah menjalani pengujian ekstensif. Hal ini mengurangi kesalahan dan memastikan bahwa produk akhir memenuhi persyaratan yang diperlukan.

5. Pendukung (support) atau pemeliharaan (maintenance)

Perangkat lunak dapat berubah setelah tersedia bagi pelanggan. Perubahan mungkin diperlukan ketika program harus beradaptasi dengan lingkungan baru atau ketika bug ditemukan selama pengujian tetapi tidak diperbaiki..