A screen shot of a cell phone

AI-generated content may be incorrect.

3223600019

Muhammad Bimo Fachrizky

Logika Fuzzy

Senin, 19 Mei 2025

**Praktikum Sistem Cerdas**

**Praktikum 11**

Logika Fuzzy

1. Tujuan Pembelajaran

* Mahasiswa dapat memahami dan menjelaskan konsep Logika Fuzzy
* Mahasiswa dapat menjelaskan model Logika Fuzzy
* Mahasiswa dapat membuat aplikasi Logika Fuzzy

Software yang di perlukan

* Microsoft Visual C++
* PyCharm

1. Langkah percobaan
2. Logika Fuzzy.cpp

#include <conio.h>

#include <cstdio>

float MFx(float a, float b, float z) {

float hasil\_MFx;

if ((z >= a) && (z < b)) hasil\_MFx = (b - z) / (b - a);

if (z <= a) hasil\_MFx = 1;

if (z >= b) hasil\_MFx = 0;

return hasil\_MFx;

}

float MFy(float a, float b, float z) {

float hasil\_MFy;

if ((z >= a) && (z < b)) hasil\_MFy = (z - a) / (b - a);

if (z <= a) hasil\_MFy = 0;

if (z >= b) hasil\_MFy = 1;

return hasil\_MFy;

}

float Min(float a, float b) {

float hasil\_min;

if (a < b)

hasil\_min = a;

else

hasil\_min = b;

return hasil\_min;

}

int main() {

float u1x, u2x, u3x, u1y, u2y, u3y;

float minR1, minR2, minR3, minR4;

float proR1, proR2, proR3, proR4;

// Fuzzyfication

float permintaan = 4000;

float persediaan = 300;

printf("Persoalan:\n");

printf("Permintaan: %.0f\n", permintaan);

printf("Persediaan: %.0f\n", persediaan);

printf("Berapa jumlah produksi?\n");

u1x = MFx(1000, 5000, permintaan); // permintaan turun

printf("Fungsi keanggotaan permintaan turun: %.2f\n", u1x);

u1y = MFy(1000, 5000, permintaan); // permintaan naik

printf("Fungsi keanggotaan permintaan naik: %.2f\n", u1y);

u2x = MFx(100, 600, persediaan); // persediaan sedikit

printf("Fungsi keanggotaan persediaan sedikit: %.2f\n", u2x);

u2y = MFy(100, 600, persediaan); // persediaan banyak

printf("Fungsi keanggotaan persediaan banyak: %.2f\n", u2y);

// Rule 1: IF Permintaan TURUN And Persediaan BANYAK THEN Produksi BERKURANG

minR1 = Min(u1x, u2y);

printf("Minimum Permintaan TURUN And Persediaan BANYAK: %.2f\n", minR1);

proR1 = 7000 - minR1 \* 5000;

printf("Produksi berkurang: %.0f\n", proR1);

// Rule 2: IF Permintaan TURUN And Persediaan SEDIKIT THEN Produksi BERKURANG

minR2 = Min(u1x, u2x);

printf("Minimum Permintaan TURUN And Persediaan SEDIKIT: %.2f\n", minR2);

proR2 = 7000 - minR2 \* 5000;

printf("Produksi berkurang: %.0f\n", proR2);

// Rule 3: IF Permintaan NAIK And Persediaan BANYAK THEN Produksi BERTAMBAH

minR3 = Min(u1y, u2y);

printf("Minimum Permintaan NAIK And Persediaan BANYAK: %.2f\n", minR3);

proR3 = 2000 + minR3 \* 5000;

printf("Produksi bertambah: %.0f\n", proR3);

// Rule 4: IF Permintaan NAIK And Persediaan SEDIKIT THEN Produksi BERTAMBAH

minR4 = Min(u1y, u2x);

printf("Minimum Permintaan NAIK And Persediaan SEDIKIT: %.2f\n", minR4);

proR4 = 2000 + minR4 \* 5000;

printf("Produksi bertambah: %.0f\n", proR4);

// Defuzzifikasi (Metode Center of Area)

float pembilang = minR1 \* proR1 + minR2 \* proR2 + minR3 \* proR3 + minR4 \* proR4;

float penyebut = minR1 + minR2 + minR3 + minR4;

float hasil\_COA = pembilang / penyebut;

printf("Produksi: %.0f\n", hasil\_COA);

}

1. Tugas

Buat program untuk suatu fuzzy interference system untuk menentukan level pengereman kendaraan berdasarkan 2 variable input, jarak antara kendaraan yang bersangkutan dengan kendaraan di depannya, dan kecepatan sesaat kendaraan. Variable jarak dibagi menjadi empat kategori fuzzy dengan 4 linguistic variable pada input membership function (IMF), short, medium, long, very long. Sementara itu variable kecepatan dibagi menjadi 3 himpunan fuzzy dengan 3 linguistic terms, low, moderat, high. Sementara level pengereman dibagi menjadi 4 tingkat dengan single-tone outputmembership function (OMF), no brake, soft, moderat, hard, seperti yang diberikan pada Fungsi Keanggotaan, Fuzzy rules dari FIS tersebut seperti diberika pada Fuzzy Rule Set.

#include <stdio.h>

#include <math.h>

#include <string.h>

float shortMF\_dis(float x) {

if (x <= 2) return 1;

else if (x > 2 && x < 6) return (6 - x) / 4.0;

else return 0;

}

float mediumMF\_dis(float x) {

if (x >= 2 && x < 6) return (x - 2) / 4.0;

else if (x >= 6 && x <= 10) return (10 - x) / 4.0;

else return 0;

}

float longMF\_dis(float x) {

if (x >= 6 && x < 10) return (x - 6) / 4.0;

else if (x >= 10 && x <= 14) return (14 - x) / 4.0;

else return 0;

}

float veryLongMF\_dis(float x) {

if (x >= 10 && x < 14) return (x - 10) / 4.0;

else if (x >= 14) return 1;

else return 0;

}

float lowMF\_sp(float y) {

if (y <= 20) return 1;

else if (y > 20 && y < 40) return (40 - y) / 20.0;

else return 0;

}

float mediumMF\_sp(float y) {

if (y >= 20 && y < 40) return (y - 20) / 20.0;

else if (y >= 40 && y <= 60) return (60 - y) / 20.0;

else return 0;

}

float highMF\_sp(float y) {

if (y >= 40 && y < 60) return (y - 40) / 20.0;

else if (y >= 60) return 1;

else return 0;

}

float minf(float a, float b) {

return (a < b) ? a : b;

}

int main() {

const float B0 = 0, B1 = 25, B2 = 50, B3 = 100;

float testData[][2] = {

{1,75}, {2,80}, {4,10}, {5,40}, {16,10}, {12,40}, {13,35},

{50,50}, {8.9,48.5}, {30,25.8}, {11,22.5}, {9.5,44}

};

int numTests = sizeof(testData) / sizeof(testData[0]);

for (int t = 0; t < numTests; ++t) {

float d = testData[t][0];

float v = testData[t][1];

float dx\_s = shortMF\_dis(d);

float dx\_m = mediumMF\_dis(d);

float dx\_l = longMF\_dis(d);

float dx\_v = veryLongMF\_dis(d);

float dy\_l = lowMF\_sp(v);

float dy\_m = mediumMF\_sp(v);

float dy\_h = highMF\_sp(v);

float a[12], z[12];

a[0] = minf(dy\_l, dx\_s); z[0] = B2;

a[1] = minf(dy\_l, dx\_m); z[1] = B1;

a[2] = minf(dy\_l, dx\_l); z[2] = B0;

a[3] = minf(dy\_l, dx\_v); z[3] = B0;

a[4] = minf(dy\_m, dx\_s); z[4] = B3;

a[5] = minf(dy\_m, dx\_m); z[5] = B2;

a[6] = minf(dy\_m, dx\_l); z[6] = B1;

a[7] = minf(dy\_m, dx\_v); z[7] = B0;

a[8] = minf(dy\_h, dx\_s); z[8] = B3;

a[9] = minf(dy\_h, dx\_m); z[9] = B3;

a[10]= minf(dy\_h, dx\_l); z[10]= B2;

a[11]= minf(dy\_h, dx\_v); z[11]= B1;

float sum\_a = 0, sum\_az = 0;

for (int i = 0; i < 12; ++i) {

sum\_a += a[i];

sum\_az += a[i] \* z[i];

}

float out = (sum\_a > 1e-6) ? sum\_az / sum\_a : B0;

const char \*label;

if (out < B1) label = "TANPA REM";

else if (out < B2) label = "RINGAN";

else if (out < B3) label = "SEDANG";

else label = "KERAS";

printf("Data uji ke-%d: Jarak = %.2f m, Kecepatan = %.2f km/jam\n", t+1, d, v);

printf(">> Tingkat Pengereman: %.2f (%s)\n\n", out, label);

}

return 0;

}

1. Hasil Percobaan
2. Logika Fuzzy.cpp

A screenshot of a computer

AI-generated content may be incorrect.

1. Tugas

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

1. Analisa

Praktikum ini mencakup beberapa program logika fuzzy. Program pertama yaitu sistem logika fuzzy yang digunakan untuk mentukan jumlah produksi berdasarkan dua input yatitu permintaan dan persediaan. Program memiliki fungsi keangggotaan MFx (a, b, z) untuk fungsi turun, dan MFy (a, b, z) untuk fungsi naik. MFx dipakai untuk mendefinisikan “permintaan turun” atau “persediaan sedikit”. MFy dipakai utnuk mendefinisikan “permintaan naik” atau “persediaan banyak”. Input yang digunakan program adalah float permintaan = 4000 dan float persediaan = 300. Kemudian di ubah menjadi derajat keanggotaan dengan menggunakan fungsi MFx dan MFy. Seterlah nilai keanggotaan diperoleh, program masuk ke tahap infersi dengan empat aturan fuzzy. Setiap aturan menggabungkan dua kondisi dengan operator AD menggunakan fungsi minimum. Kemudian defuzzifikasi menggunakan metode center of area (COA), dengan mencari rata-rata ketimbang dari semua hasil infers berdasarkan niai derajat minimum. Hasilnya berupa nilai produksi yang disarankan dalam bentuk angka pasti. Program ini mencerminkan prinsip dasar sistem berbasis logika fuzzy yang meniru cara berfikir manusia dalam mengambil Keputusan dengan mempertimbangkan ketidakpastian dan tingkat kepercayaan.

Program kedua adalah implementasi sistem logika fuzzy untuk menentukan tingkat pengereman berdasarkan dua input yaitu jarak kendaraan dengan objek dan kecepatan kendaraan. Sistem ini menggunakan fuzzy untuk mengolah data numerik menjadi Keputusan linguistic seperti “tanpa rem”, “ringan”, “sedang”, dan “keras”. Program ini akan mendefinisika fungsi keanggotaan untuk variable input jarak (dis) dan kecepatan (sp). Jarak diklasifikasikan menjadi long, medium, short, dan very long, sedangkan kecepatan dikalsifikasikan menjadi low, medium, dan high. Setelah itu program membaca data uji yang berisi pasangan nilai jarak dan kecepatan. Untuk setiap pasangan dihitung semua kemungkinan kombinasi antara kondisi jarak dan kecepatan. Setiap aturan menghasilkan nilai aktivasi dan Keputusan output. Kemudian dilakukan defuzzifikasi dengan metode rata-rata terbobot untuk menghitung nilai crisp dari Keputusan pengereman. Program ini sama seperti program pertama yaitu meniru cara manusia mengambil Keputusan dari kondisi yang tidak pasti.

1. Kesimpulan

Dari analisis dua program logika fuzzy dalam praktikum ini, dapat disimpulkan bahwa logika fuzzy merupakan pendekatan yang efektif dalam pengambilan keputusan pada kondisi yang tidak pasti dan bersifat samar. Kedua program menunjukkan penerapan prinsip dasar sistem fuzzy yang meniru cara berpikir manusia, yaitu dengan mempertimbangkan berbagai kondisi melalui derajat keanggotaan dan menggabungkannya menggunakan aturan-aturan logis untuk menghasilkan keputusan yang realistis. Program pertama fokus pada penentuan jumlah produksi berdasarkan input permintaan dan persediaan. Program ini menggunakan fungsi keanggotaan naik dan turun, serta metode inferensi dengan operator minimum dan defuzzifikasi menggunakan metode Center of Area (COA). Sedangkan program kedua menerapkan logika fuzzy untuk menentukan tingkat pengereman berdasarkan input jarak dan kecepatan kendaraan. Kedua implementasi ini memperlihatkan bahwa logika fuzzy dapat digunakan secara fleksibel dalam berbagai konteks, baik untuk sistem kontrol produksi maupun sistem pendukung keselamatan kendaraan.