

# Praktikum Sistem Cerdas



<b>NRP</b>	: 3223600019
<b>Nama</b>	: Muhammad Bimo Fachrizky
<b>Materi</b>	: Logika Fuzzy
<b>Tanggal</b>	: Senin, 19 Mei 2025

# Praktikum 11

## Logika Fuzzy

### I. 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

### II. Langkah percobaan

#### 1. Logika Fuzzy.cpp

```
#include <conio.h>
#include <stdio.h>

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);
}

```

## 2. 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;
}

```

### III. Hasil Percobaan

#### 1. Logika Fuzzy.cpp

```

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\Praktikum 11>fuzzy
Persoalan:
Permintaan: 4000
Persediaan: 300
Berapa jumlah produksi?
Fungsi keanggotaan permintaan turun: 0.25
Fungsi keanggotaan permintaan naik: 0.75
Fungsi keanggotaan persediaan sedikit: 0.60
Fungsi keanggotaan persediaan banyak: 0.40
Minimum Permintaan TURUN And Persediaan BANYAK: 0.25
Produksi berkurang: 5750
Minimum Permintaan TURUN And Persediaan SEDIKIT: 0.25
Produksi berkurang: 5750
Minimum Permintaan NAIK And Persediaan BANYAK: 0.40
Produksi bertambah: 4000
Minimum Permintaan NAIK And Persediaan SEDIKIT: 0.60
Produksi bertambah: 5000
Produksi: 4983

```

## 2. Tugas

```
Data uji ke-1: Jarak = 1.00 m, Kecepatan = 75.00 km/jam
>> Tingkat Pengereman: 100.00 (KERAS)

Data uji ke-2: Jarak = 2.00 m, Kecepatan = 80.00 km/jam
>> Tingkat Pengereman: 100.00 (KERAS)

Data uji ke-3: Jarak = 4.00 m, Kecepatan = 10.00 km/jam
>> Tingkat Pengereman: 37.50 (RINGAN)

Data uji ke-4: Jarak = 5.00 m, Kecepatan = 40.00 km/jam
>> Tingkat Pengereman: 62.50 (SEDANG)

Data uji ke-5: Jarak = 16.00 m, Kecepatan = 10.00 km/jam
>> Tingkat Pengereman: 0.00 (TANPA REM)

Data uji ke-6: Jarak = 12.00 m, Kecepatan = 40.00 km/jam
>> Tingkat Pengereman: 12.50 (TANPA REM)

Data uji ke-7: Jarak = 13.00 m, Kecepatan = 35.00 km/jam
>> Tingkat Pengereman: 4.17 (TANPA REM)

Data uji ke-8: Jarak = 50.00 m, Kecepatan = 50.00 km/jam
>> Tingkat Pengereman: 12.50 (TANPA REM)

Data uji ke-9: Jarak = 8.90 m, Kecepatan = 48.50 km/jam
>> Tingkat Pengereman: 49.60 (RINGAN)

Data uji ke-10: Jarak = 30.00 m, Kecepatan = 25.80 km/jam
>> Tingkat Pengereman: 0.00 (TANPA REM)

Data uji ke-11: Jarak = 11.00 m, Kecepatan = 22.50 km/jam
>> Tingkat Pengereman: 2.50 (TANPA REM)

Data uji ke-12: Jarak = 9.50 m, Kecepatan = 44.00 km/jam
>> Tingkat Pengereman: 39.00 (RINGAN)
```

## IV. Analisa

Praktikum ini mencakup beberapa program logika fuzzy. Program pertama yaitu sistem logika fuzzy yang digunakan untuk menentukan jumlah produksi berdasarkan dua input yaitu permintaan dan persediaan. Program memiliki fungsi keanggotaan  $MF_x(a, b, z)$  untuk fungsi turun, dan  $MF_y(a, b, z)$  untuk fungsi naik.  $MF_x$  dipakai untuk mendefinisikan “permintaan turun” atau “persediaan sedikit”.  $MF_y$  dipakai untuk 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  $MF_x$  dan  $MF_y$ . Setelah nilai keanggotaan diperoleh, program masuk ke tahap inferensi 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 inferensi berdasarkan nilai 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.

## V. 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.