

Nama : Nusalendra Putra Restu Bumi
NBI : 1462000017
Kelas : Robotika (A)

1. Sistem kendali pada perangkat embedded atau robotika adalah serangkaian komponen dan algoritma yang bertanggung jawab untuk mengatur dan mengendalikan perilaku perangkat tersebut. Tujuan utama dari sistem kendali ini adalah untuk memungkinkan perangkat atau robot beroperasi secara otomatis atau semi-otomatis dalam berbagai tugas atau lingkungan. Beberapa komponen penting dalam sistem kendali pada perangkat embedded atau robotika meliputi:

- Sensor: Sensor digunakan untuk mengumpulkan informasi tentang lingkungan sekitar atau kondisi internal perangkat. Contoh sensor yang umum digunakan adalah sensor jarak, sensor suhu, sensor kecepatan, sensor tekanan, sensor cahaya, dan sebagainya. Data yang diperoleh dari sensor digunakan sebagai masukan untuk sistem kendali.
- Aktuator: Aktuator bertanggung jawab untuk mengubah sinyal kendali menjadi aksi fisik. Contohnya termasuk motor, aktuator hidrolik, aktuator pneumatik, dan sebagainya. Aktuator merespons perintah dari sistem kendali untuk menggerakkan atau mengubah kondisi fisik perangkat.
- Mikrokontroler atau Mikroprosesor: Mikrokontroler atau mikroprosesor adalah otak dari sistem kendali yang melakukan pemrosesan data dan mengambil keputusan berdasarkan program yang dijalankan. Ini melibatkan pemantauan sensor, pengolahan data, dan mengirim sinyal kendali ke aktuator.
- Algoritma Kontrol: Algoritma kontrol merujuk pada serangkaian aturan, rumus matematika, atau model yang digunakan untuk mengendalikan perilaku perangkat. Jenis algoritma kontrol yang digunakan dapat bervariasi tergantung pada aplikasi dan kebutuhan khusus. Beberapa contoh algoritma kontrol yang umum digunakan termasuk kontrol PID (Proporsional-Integral-Derivatif), kontrol adaptif, kontrol fuzzy, dan sebagainya.
- Antarmuka Pengguna: Sistem kendali sering memiliki antarmuka pengguna yang memungkinkan pengguna berinteraksi dengan perangkat atau robot. Antarmuka ini

dapat berupa tampilan visual, seperti layar sentuh atau tampilan grafis, atau dapat berupa perangkat input seperti tombol atau joystick.

Selain itu, dalam sistem kendali robotika yang lebih kompleks, ada konsep seperti perencanaan gerakan, persepsi lingkungan, dan navigasi yang terlibat untuk menghadapi tantangan yang lebih kompleks.

Pengembangan sistem kendali pada perangkat embedded atau robotika melibatkan berbagai disiplin ilmu, seperti teknik kendali, elektronika, ilmu komputer, dan mekatronika. Tujuan utama dari pengembangan sistem kendali adalah untuk mencapai performa yang baik, stabilitas, keandalan, dan keselamatan operasi perangkat atau robotika.

2. Berikut adalah beberapa jenis sistem kendali yang umum digunakan:

- Kontrol Proporsional (Proportional Control): Pada kontrol proporsional, output atau tindakan kontrol proporsional terhadap perbedaan antara nilai target dan nilai saat ini. Semakin besar perbedaan, semakin besar pula aksi kontrol yang dihasilkan. Namun, kontrol proporsional cenderung memiliki overshoot dan tidak stabil saat mencapai nilai target.
- Kontrol Integral (Integral Control): Kontrol integral bertujuan untuk mengurangi kesalahan akumulasi antara nilai target dan nilai saat ini. Ini dicapai dengan mengintegrasikan perbedaan seiring waktu. Kontrol integral dapat membantu menghilangkan steady-state error dalam sistem dan meningkatkan presisi.
- Kontrol Derivatif (Derivative Control): Kontrol derivatif berfokus pada laju perubahan kesalahan. Dengan menganalisis laju perubahan, kontrol derivatif membantu dalam mengantisipasi perubahan arah atau kecepatan yang cepat. Ini membantu dalam mengurangi overshoot dan meningkatkan stabilitas sistem. Namun, kontrol derivatif rentan terhadap noise dan tidak stabil jika digunakan sendiri.
- Kontrol PID (Proportional-Integral-Derivative Control): Kontrol PID adalah kombinasi dari tiga jenis kontrol di atas. Kontrol PID mencoba menggabungkan keuntungan dari setiap jenis kontrol untuk mencapai kinerja yang lebih baik. Komponen proporsional memberikan aksi kontrol sebanding dengan kesalahan saat ini, komponen integral menghilangkan steady-state error, dan komponen derivatif membantu dalam mengatasi perubahan cepat.

- Kontrol Fuzzy (Fuzzy Control): Kontrol fuzzy adalah jenis kontrol yang berbasis pada logika fuzzy, yang menggabungkan aspek dari logika boolean dan analisis statistik. Dalam kontrol fuzzy, variabel input dan output dinyatakan dalam bentuk himpunan fuzzy dengan fungsi keanggotaan yang mencerminkan tingkat keanggotaan suatu nilai dalam himpunan tersebut. Kontrol fuzzy memungkinkan penanganan informasi yang ambigu dan tidak pasti, serta dapat digunakan untuk mengendalikan sistem yang kompleks dan non-linear.
- Kontrol Adaptif (Adaptive Control): Kontrol adaptif adalah jenis kontrol yang dapat menyesuaikan diri terhadap perubahan dalam sistem atau lingkungan. Kontrol adaptif menggunakan algoritma yang mampu mengubah parameter atau struktur kontrol sesuai dengan kondisi yang berubah untuk mencapai kinerja yang optimal. Ini berguna dalam situasi di mana parameter sistem berubah seiring waktu atau ketika sistem menghadapi variasi yang signifikan.

Setiap jenis sistem kendali ini memiliki kelebihan dan kelemahan tertentu, dan pilihan jenis kontrol yang tepat tergantung pada karakteristik sistem yang dikendalikan serta kebutuhan spesifik aplikasi.

3. Mencari nilai $\mu_{Kering}(40)$, $\mu_{Normal}(40)$ dan $\mu_{Basah}(40)$

- $\mu_{Kering}(40) = \frac{45-40}{45-20} = \frac{5}{25} = \frac{1}{5} = 0.2$
- $\mu_{Normal}(40) = \frac{40-25}{45-25} = \frac{15}{20} = \frac{3}{4} = 0.75$
- $\mu_{Basah}(40) = 0$

4. Mencari

nilai

$\mu_{Dingin}(37)$, $\mu_{Sejuk}(37)$, $\mu_{Normal}(37)$, $\mu_{Hangat}(37)$, $\mu_{Panas}(37)$

- $\mu_{Dingin}(37) = 0$
- $\mu_{Sejuk}(37) = 0$
- $\mu_{Normal}(37) = 0$
- $\mu_{Hangat}(37) = \frac{40-37}{40-36} = \frac{3}{4} = 0.75$

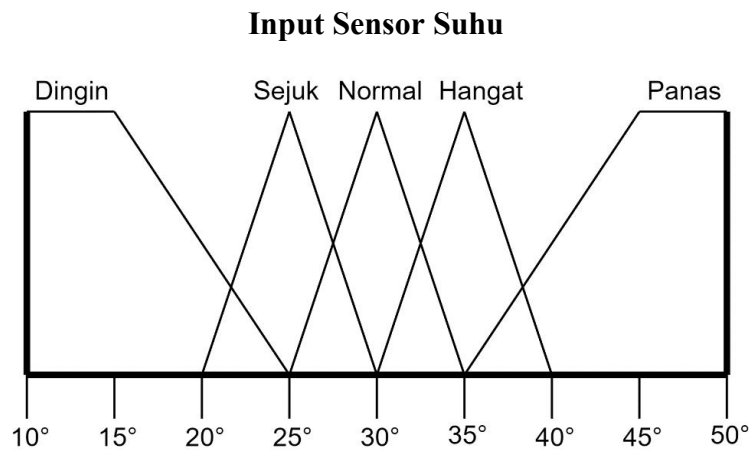
- $\mu_{Panas}(37) = \frac{37-35}{40-35} = \frac{2}{5} = 0.4$

5. Rancangan sistem kendali Fuzzy logic

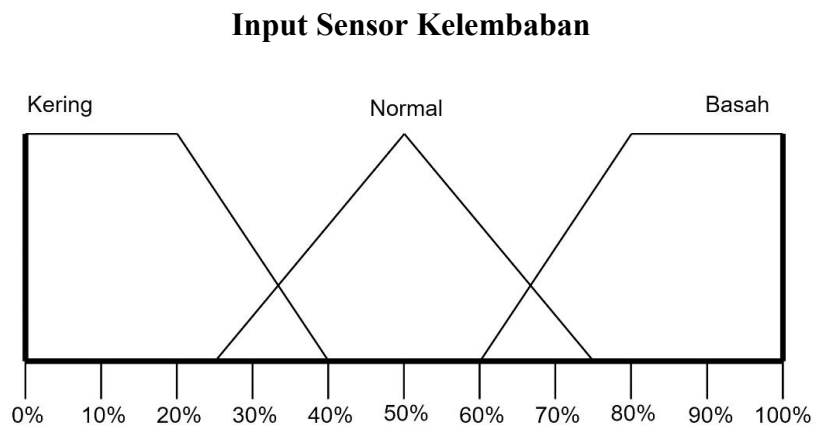
- Sistem Kendali : Sistem pengaturan jumlah air untuk penyiraman otomatis
- Parameter Input 1 : Sensor Suhu
- Parameter Input 2 : Sensor Kelembaban
- Parameter Output : Pompa untuk kontrol debit air

6. Membership Function

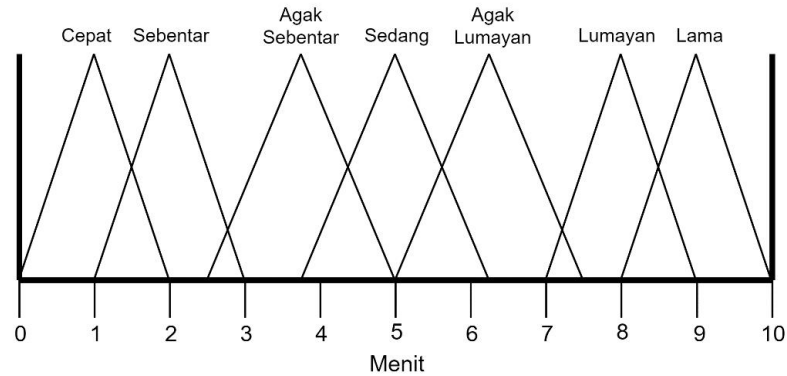
- Gambar Membership Function Parameter Input 1 :



- Gambar Membership Function Parameter Input 2 :



- Gambar Membership Function Parameter Output :



7. Basis Aturan / Rule

a) Tabel

Kelembaban/Suhu	Dingin	Sejuk	Normal	Hangat	Panas
Basah	Cepat	Sebentar	Agak Sebentar	Sedang	Agak Lumayan
Normal	Sebentar	Agak Sebentar	Sedang	Agak Lumayan	Lumayan
Kering	Agak Sebentar	Sedang	Agak Lumayan	Lumayan	Lama

b) IF THEN

- 1) IF Suhu = Dingin & Kelembaban = Basah THEN Kecepatan Air = Cepat
- 2) IF Suhu = Dingin & Kelembaban = Normal THEN Kecepatan Air = Sebentar
- 3) IF Suhu = Dingin & Kelembaban = Kering THEN Kecepatan Air = Agak Sebentar
- 4) IF Suhu = Sejuk & Kelembaban = Basah THEN Kecepatan Air = Sebentar
- 5) IF Suhu = Sejuk & Kelembaban = Normal THEN Kecepatan Air = Agak Sebentar
- 6) IF Suhu = Sejuk & Kelembaban = Kering THEN Kecepatan Air = Sedang
- 7) IF Suhu = Normal & Kelembaban = Basah THEN Kecepatan Air = Agak Sebentar
- 8) IF Suhu = Normal & Kelembaban = Normal THEN Kecepatan Air = Sedang
- 9) IF Suhu = Normal & Kelembaban = Kering THEN Kecepatan Air = Agak Lumayan

- 10) IF Suhu = Hangat & Kelembaban = Basah THEN Kecepatan Air = Sedang
- 11) IF Suhu = Hangat & Kelembaban = Normal THEN Kecepatan Air = Agak Lumayan
- 12) IF Suhu = Hangat & Kelembaban = Kering THEN Kecepatan Air = Lumayan
- 13) IF Suhu = Panas & Kelembaban = Basah THEN Kecepatan Air = Agak Lumayan
- 14) IF Suhu = Panas & Kelembaban = Normal THEN Kecepatan Air = Lumayan
- 15) IF Suhu = Panas & Kelembaban = Kering THEN Kecepatan Air = Lama

8. Input sensor suhu = 27 derajat celcius, input sensor kelembaban = 35%

- Derajat keanggotaan sensor suhu

a) $\mu_{Dingin}(27) = 0$

b) $\mu_{Sejuk}(27) = \frac{30-27}{30-25} = \frac{3}{5} = 0.6$

c) $\mu_{Normal}(27) = \frac{27-25}{30-25} = \frac{2}{5} = 0.4$

d) $\mu_{Hangat}(27) = 0$

e) $\mu_{Panas}(27) = 0$

- Derajat keanggotaan sensor kelembaban

a) $\mu_{Kering}(35) = \frac{40-35}{40-20} = \frac{5}{20} = \frac{1}{4} = 0.25$

b) $\mu_{Normal}(35) = \frac{35-25}{50-25} = \frac{10}{25} = \frac{2}{5} = 0.4$

c) $\mu_{Basah}(35) = 0$

- Dari hasil 2 input, didapatkan nilai keanggotaan tertinggi ada pada μ_{Sejuk} pada sensor suhu dan μ_{Normal} pada sensor kelembaban. Sehingga dapat diambil basis aturan/rule pada no 4, yaitu : **IF Suhu = Sejuk & Kelembaban = Normal THEN Kecepatan Air = Agak Seentar**
- Sehingga dari hasil input sensor suhu 28 dan sensor kelembaban 35, output kecepatan air yang didapat adalah “Agak Seentar”

9. Kode program menghitung Membership Function dan Basis aturan / rule

```

#include <stdio.h>

// Fungsi untuk menghitung derajat keanggotaan suhu
float hitungDerajatKeanggotaanSuhu(float suhu, int range_min, int range_max) {
    if (suhu <= range_min || suhu >= range_max) {
        return 0.0;
    } else if (suhu < range_max - 5) {
        return (suhu - range_min) / (range_max - range_min - 5);
    } else {
        return 1.0;
    }
}

// Fungsi untuk menghitung derajat keanggotaan kelembapan
float hitungDerajatKeanggotaanKelembapan(float kelembapan, int range_min, int range_max) {
    if (kelembapan <= range_min || kelembapan >= range_max) {
        return 0.0;
    } else if (kelembapan < range_max - 5) {
        return (kelembapan - range_min) / (range_max - range_min - 5);
    } else {
        return 1.0;
    }
}

// Fungsi untuk menghitung kecepatan air berdasarkan basis aturan
float hitungKecepatanAir(float derajatKeanggotaanSuhu, float derajatKeanggotaanKelembapan) {
    if (derajatKeanggotaanSuhu == 0.0 || derajatKeanggotaanKelembapan == 0.0) {
        return 0.0;
    } else if (derajatKeanggotaanSuhu <= 1.0 && derajatKeanggotaanKelembapan <= 1.0) {
        if (derajatKeanggotaanSuhu <= 0.5 && derajatKeanggotaanKelembapan <= 0.5) {
            return 0.0;
        } else if (derajatKeanggotaanSuhu <= 0.5 && derajatKeanggotaanKelembapan <= 0.75) {
            return derajatKeanggotaanKelembapan - 0.5;
        } else if (derajatKeanggotaanSuhu <= 0.5 && derajatKeanggotaanKelembapan <= 1.0) {
            return derajatKeanggotaanKelembapan;
        } else if (derajatKeanggotaanSuhu <= 0.75 && derajatKeanggotaanKelembapan <= 0.5) {
            return derajatKeanggotaanSuhu;
        } else if (derajatKeanggotaanSuhu <= 0.75 && derajatKeanggotaanKelembapan <= 0.75) {
            return (derajatKeanggotaanSuhu + derajatKeanggotaanKelembapan) / 2;
        } else if (derajatKeanggotaanSuhu <= 0.75 && derajatKeanggotaanKelembapan <= 1.0) {
            return derajatKeanggotaanSuhu;
        } else if (derajatKeanggotaanSuhu <= 1.0 && derajatKeanggotaanKelembapan <= 0.5) {
            return derajatKeanggotaanKelembapan;
        } else if (derajatKeanggotaanSuhu <= 1.0 && derajatKeanggotaanKelembapan <= 0.75) {
            return derajatKeanggotaanSuhu;
        } else {
            return (derajatKeanggotaanSuhu + derajatKeanggotaanKelembapan) / 2;
        }
    } else {
        return 0.0;
    }
}

int main() {
    float suhu, kelembapan;

    printf("Masukkan suhu (dalam derajat Celsius): ");
    scanf("%f", &suhu);

    printf("Masukkan kelembapan (dalam persen): ");
    scanf("%f", &kelembapan);

    float derajatKeanggotaanSuhu[5];
    derajatKeanggotaanSuhu[0] = hitungDerajatKeanggotaanSuhu(suhu, 10, 25); // Dingin
    derajatKeanggotaanSuhu[1] = hitungDerajatKeanggotaanSuhu(suhu, 20, 30); // Sejuk
    derajatKeanggotaanSuhu[2] = hitungDerajatKeanggotaanSuhu(suhu, 25, 35); // Normal
    derajatKeanggotaanSuhu[3] = hitungDerajatKeanggotaanSuhu(suhu, 30, 40); // Hangat
    derajatKeanggotaanSuhu[4] = hitungDerajatKeanggotaanSuhu(suhu, 35, 50); // Panas

    float derajatKeanggotaanKelembapan[3];
    derajatKeanggotaanKelembapan[0] = hitungDerajatKeanggotaanKelembapan(kelembapan, 0, 40); // Kering
    derajatKeanggotaanKelembapan[1] = hitungDerajatKeanggotaanKelembapan(kelembapan, 25, 75); // Normal
    derajatKeanggotaanKelembapan[2] = hitungDerajatKeanggotaanKelembapan(kelembapan, 60, 100); // Basah

    float kecepatanAir[7];
    kecepatanAir[0] = hitungKecepatanAir(derajatKeanggotaanSuhu[0], derajatKeanggotaanKelembapan[2]); //
    Cepat
    kecepatanAir[1] = hitungKecepatanAir(derajatKeanggotaanSuhu[0], derajatKeanggotaanKelembapan[1]); //
    Sebarang
    kecepatanAir[2] = hitungKecepatanAir(derajatKeanggotaanSuhu[0], derajatKeanggotaanKelembapan[0]); //
    Agak Sebarang
    kecepatanAir[3] = hitungKecepatanAir(derajatKeanggotaanSuhu[1], derajatKeanggotaanKelembapan[2]); //
    Sebarang
    kecepatanAir[4] = hitungKecepatanAir(derajatKeanggotaanSuhu[1], derajatKeanggotaanKelembapan[1]); //
    Agak Sebarang
    kecepatanAir[5] = hitungKecepatanAir(derajatKeanggotaanSuhu[1], derajatKeanggotaanKelembapan[0]); //
    Sedang
    kecepatanAir[6] = hitungKecepatanAir(derajatKeanggotaanSuhu[2], derajatKeanggotaanKelembapan[2]); //
    Agak Sebarang
    // Lanjutkan perhitungan kecepatanAir sesuai dengan basis aturan yang diberikan

    // Cetak hasil kecepatan air
    printf("Hasil kecepatan air:\n");
    for (int i = 0; i < 7; i++) {
        printf("Kecepatan Air %d: %f\n", i+1, kecepatanAir[i]);
    }

    return 0;
}

```