

LAMPIRAN

```
% Deklarasi range setiap masukan
range_LokasiGangguan = [0 90];
range_BesarGangguan = [0 80];
range_StatusRelay = [0 100];

% Deklarasi Fungsi Keanggotaan dalam bentuk array (4 angka)
MF_lokasiInternal = [0 0 30 39];
MF_lokasiEksternal = [30 40 80 90];

MF_gangguanKecil = [0 0 20 30];
MF_gangguanBesar = [20 30 80 80];

MF_statusTidakBekerja = [0 0 50 60];
MF_statusBekerja = [50 60 100 100];

MF_kesimpulanTidakHandal = [0 0 50 60];
MF_kesimpulanHandal = [51 60 100 100];

% Input/Masukan Sistem Inferensi Fuzzy
inputlokasiGangguan = input(['Nilai lokasi
gangguan(',int2str(range_LokasiGangguan(1)),'-',
int2str(range_LokasiGangguan(2)),'): ']);
inputbesarGangguan = input(['Besar
gangguan(',int2str(range_BesarGangguan(1)),'-',
int2str(range_BesarGangguan(2)),'): ']);
inputstatusRelay = input(['Status relay(',
int2str(range_StatusRelay(1)),'-',int2str(range_StatusRelay(2)),
'): ']);

% Mencari nilai fungsi keanggotaan dari setiap himpunan
masukan/input
nilaiMF_lokasiInternal = trapmf(inputlokasiGangguan,
MF_lokasiInternal);
nilaiMF_lokasiEksternal = trapmf(inputlokasiGangguan,
MF_lokasiEksternal);

nilaiMF_gangguanKecil = trapmf(inputbesarGangguan,
MF_gangguanKecil);
nilaiMF_gangguanBesar = trapmf(inputbesarGangguan,
MF_gangguanBesar);

nilaiMF_statusTidakBekerja = trapmf(inputstatusRelay,
MF_statusTidakBekerja);
nilaiMF_statusBekerja = trapmf(inputstatusRelay, MF_statusBekerja);

% Mencari nilai fungsi keanggotaan setiap rule
% Rule 1 :
% Lokasi Eksternal, Gangguan Kecil, Relay tidak bekerja,
Kesimpulan Baik
% Alpha 1
nilaiMF_Rule(1) = min([nilaiMF_lokasiEksternal,
nilaiMF_gangguanKecil, nilaiMF_statusTidakBekerja]);
```

```

% Z1
centerKesimpulan_Rule(1) = hitung_centerKesimpulan(nilaiMF_Rule(1),
MF_kesimpulanHandal);

% Rule 2 :
% Lokasi Eksternal, Gangguan Kecil, Relay bekerja, Kesimpulan
Tidak Baik
% Alpha 2
nilaiMF_Rule(2) = min([nilaiMF_lokasiEksternal,
nilaiMF_gangguanKecil, nilaiMF_statusBekerja]);
% Z2
centerKesimpulan_Rule(2) = hitung_centerKesimpulan(nilaiMF_Rule(2),
MF_kesimpulanTidakHandal);

% Rule 3 :
% Lokasi Eksternal, Gangguan Besar, Relay tidak bekerja,
Kesimpulan Baik
% Alpha 3
nilaiMF_Rule(3) = min([nilaiMF_lokasiEksternal,
nilaiMF_gangguanBesar, nilaiMF_statusTidakBekerja]);
% Z3
centerKesimpulan_Rule(3) = hitung_centerKesimpulan(nilaiMF_Rule(3),
MF_kesimpulanHandal);

% Rule 4 :
% Lokasi Eksternal, Gangguan Besar, Relay Bekerja, Kesimpulan
Tidak Baik
% Alpha 4
nilaiMF_Rule(4) = min([nilaiMF_lokasiEksternal,
nilaiMF_gangguanBesar, nilaiMF_statusBekerja]);
% Z4
centerKesimpulan_Rule(4) = hitung_centerKesimpulan(nilaiMF_Rule(4),
MF_kesimpulanTidakHandal);

% Rule 5 :
% Lokasi Internal, Gangguan Kecil, Relay Bekerja, Kesimpulan Tidak
Baik
% Alpha 5
nilaiMF_Rule(5) = min([nilaiMF_lokasiInternal,
nilaiMF_gangguanKecil, nilaiMF_statusBekerja]);
% Z5
centerKesimpulan_Rule(5) = hitung_centerKesimpulan(nilaiMF_Rule(5),
MF_kesimpulanTidakHandal);

% Rule 6 :
% Lokasi Internal, Gangguan Besar, Relay Bekerja, Kesimpulan Baik
% Alpha 6
nilaiMF_Rule(6) = min([nilaiMF_lokasiInternal,
nilaiMF_gangguanBesar, nilaiMF_statusBekerja]);
% Z6
centerKesimpulan_Rule(6) = hitung_centerKesimpulan(nilaiMF_Rule(6),
MF_kesimpulanHandal);

% Rule 7 :
% Lokasi Internal, Gangguan Besar, Relay Tidak Bekerja, Kesimpulan

```

```

Tidak Baik
% Alpha 7
nilaiMF_Rule(7) = min([nilaiMF_lokasiInternal,
nilaiMF_gangguanBesar, nilaiMF_statusTidakBekerja]);
% Z7
centerKesimpulan_Rule(7) = hitung_centerKesimpulan(nilaiMF_Rule(7),
MF_kesimpulanTidakHandal);

% Rule 8 :
% Lokasi Internal, Gangguan Kecil, Relay Tidak Bekerja, Kesimpulan
Baik
% Alpha 8
nilaiMF_Rule(8) = min([nilaiMF_lokasiInternal,
nilaiMF_gangguanKecil, nilaiMF_statusTidakBekerja]);
% Z8
centerKesimpulan_Rule(8) = hitung_centerKesimpulan(nilaiMF_Rule(8),
MF_kesimpulanHandal);

% Menghitung nilai numerator/pembilang untuk rumus defuzzifikasi
center
% average
num = 0;
for i=1:length(centerKesimpulan_Rule)
    num = num + nilaiMF_Rule(i)*centerKesimpulan_Rule(i);
end
% Menghitung nilai denominator/penyebut untuk rumus defuzzifikasi
center
% average
den = sum(nilaiMF_Rule);

%Menghitung hasil defuzzifikasi
Kesimpulan = num/den;

% Menyimpulkan hasil akhir
nilaiMF_KesimpulanHandal = trapmf(Kesimpulan, MF_kesimpulanHandal);
nilaiMF_KesimpulanTidakHandal = trapmf(Kesimpulan,
MF_kesimpulanTidakHandal);

if nilaiMF_KesimpulanHandal > nilaiMF_KesimpulanTidakHandal
    disp(['Status relay HANDAL dengan nilai kehandalan : ',
num2str(nilaiMF_KesimpulanHandal)])
else
    disp(['Status relay TIDAK HANDAL dengan nilai
ketidakehandalan : ', num2str(nilaiMF_KesimpulanTidakHandal)])
end

function center = hitung_centerKesimpulan(x, y)
    a = y(1);
    m = y(2);
    n = y(3);
    b = y(4);
    if a == m
        crisp = x*(b-n)+b;
        centerPersegi = a+((crisp-a)/2);
        luasPersegi = (crisp-a)*x;
    end
end

```

```

        centerSegitiga = ((b-crisp)/3)+crisp;
        luasSegitiga = (b-crisp)*x/2;

elseif b == n
    crisp = -(x*(m-a)-m);
    centerPersegi = crisp+((b-crisp)/2);
    luasPersegi = (b-crisp)*x;
    centerSegitiga = (2*(crisp-a)/3)+a;
    luasSegitiga = (crisp-a)*x/2;
end
if (luasPersegi+luasSegitiga) == 0
    center = 0;
else
    center =
        ((centerPersegi*luasPersegi)+(centerSegitiga*luasSegitiga))/(luasP
        ersegi+luasSegitiga);
end

end

```