

BAB III

METODE PENELITIAN

Metode penelitian yang digunakan dalam pengerjaan Tugas Akhir ini adalah studi literatur berupa data-data dari masing-masing komponen, perancangan perangkat keras dan pembuatan program untuk melakukan pengaturan temperatur serta kelembaban pada ruangan.

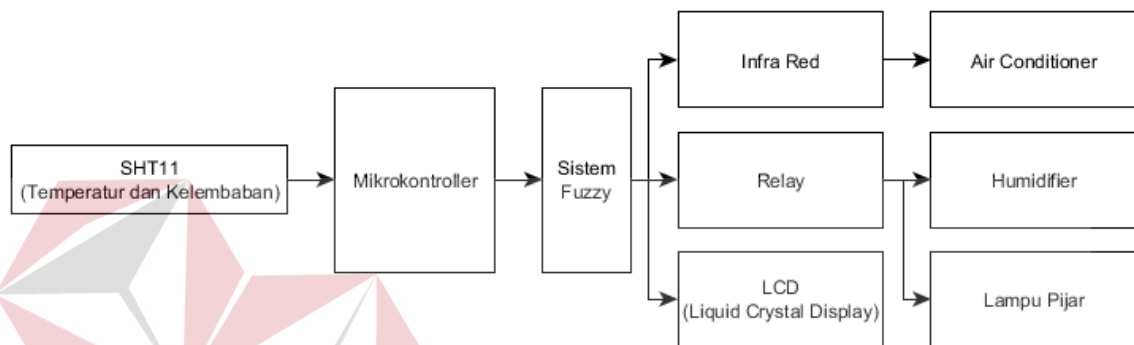
Pada sistem terdapat sebuah input yaitu sensor modul SHT11 sebagai pendeteksi temperatur dan kelembaban. Terdapat empat buah output yaitu satu buah AC, lima lampu, satu buah *humidifier*, dan satu buah LCD. Dengan AC sebagai alat yang mengatur temperatur serta membantu mengurangi kelembaban. Pada lampu dapat menghasilkan temperatur panas pada ruangan serta dapat mengurangi kelembaban, *humidifier* sebagai alat yang mengatur kelembaban jika pada ruangan sangat rendah nilai kelembabannya dan dengan kelembaban yang tinggi dapat menurunkan temperatur, dan LCD menampilkan nilai temperatur serta kelembaban serta hasil perhitungan. Dengan adanya *output* ini maka aktuator akan berkoordinasi sehingga temperatur serta kelembaban pada ruangan dapat terkontrol sesuai dengan input yang telah dideteksi oleh sensor.

Sistem bekerja didukung oleh mikrokontroler Arduino Mega 2560 sebagai unit pengendali. Nilai temperatur dan kelembaban yang dideteksi modul SHT11 akan diolah menggunakan logika *fuzzy* pada Arduino Mega 2560, setelah diproses maka Arduino Mega 2560 akan mengirimkan perintah pada komponen-komponen

yang akan mengeluarkan *output*-an sesuai *rule* yang telah ditetapkan pada program yang terdapat pada Arduino Mega 2560.

3.1 Perancangan Sistem

Secara umum gambar Blok Diagram pada rancangan perangkat keras.



Gambar 3.1 Blok Diagram Sistem

Tiap-tiap bagian dari diagram blok sistem pada Gambar 3.1 dapat dijelaskan sebagai berikut :

1. *Input* pada Mikrokontroler :

- a. Sensor Modul SHT11 : Sebagai pendeteksi nilai temperatur dan kelembaban pada disekitar sistem otomasi.

2. *Output* pada Mikrokontroler :

- a. *Air Conditioner* : Sebagai pendingin ruangan sistem otomasi saat temperatur tinggi dengan beberapa tingkatan *level* pada temperatur.
- b. *Humidifier* : Sebagai pengatur kelembaban pada sistem otomasi dengan metode *on* dan *off* sesuai dengan kondisi serta durasi waktu sampai dengan menyesuaikan tingkat kekurangan kelembaban yang telah ditentukan.

- c. Lampu Pijar : Sebagai menurunkan tingkat kelembaban dengan kontrol *on* dan *off* sesuai dengan kondisi ruang.
- d. LCD (*Liquid Crystal Display*) : Sebagai media pengeluaran berupa nilai yang didapat oleh sensor SHT11 dan nilai perhitungan dari sistem *fuzzy*.
- e. Infra Red : Sebagai pengganti *remote* untuk mengatur temperatur pada *air conditioner*
- f. Relay : Sebagai saklar elektrik yang berguna untuk mengaktifkan atau memutus aliran listrik.

Penjelasan pada blok diagram ini, nilai temperatur dan kelembaban yang dideteksi oleh sensor SHT11 akan dikirim ke mikrokontroller dan diolah oleh sistem *fuzzy*. Pada sistem *fuzzy* ini terdapat *rule-rule* yang telah ditentukan. Setelah selesai diolah pada sistem *fuzzy* maka mikrokontroller akan mengirimkan perintah kepada aktuator sesuai dengan *rule* yang telah ditetapkan pada program. Aktuator LCD, AC, *humidifier*, dan lampu pijar akan aktif sesuai dengan kondisi yang telah ditentukan. Aktuator akan berhenti ketika nilai yang dibaca oleh sensor SHT11 sesuai dengan *rule* pada sistem *fuzzy* karena sistem telah terpenuhi.

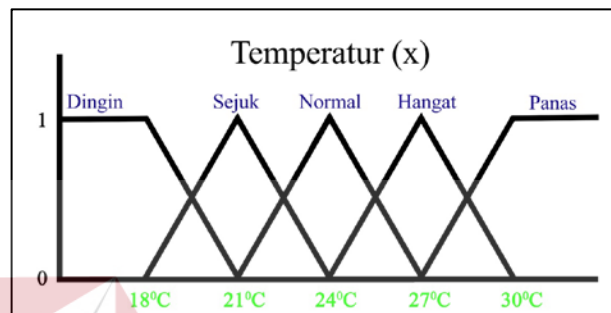
3.2 Penjelasan Sistem *Fuzzy* Tsukamoto

Pada penelitian ini metode *fuzzy* yang akan digunakan oleh penulis yaitu *Fuzzy* Tsukamoto. Penggunaan *Fuzzy* Tsukamoto ini untuk mengontrol seluruh aktuator secara *real time*. Berikut adalah *rule-rule* metode *Fuzzy* Tsukamoto yang telah diterapkan sebagai keluaran pada sistem yang telah dirancang.

1. *Input* pada *Fuzzy Tsukamoto* :

Terdapat dua *input* pada *fuzzy* Tsukamoto yaitu temperatur dan kelembaban, berikut adalah pembagian tingkat dari *input*-an temperatur serta kelembaban tertera pada Gambar 3.2 dan Gambar 3.3.

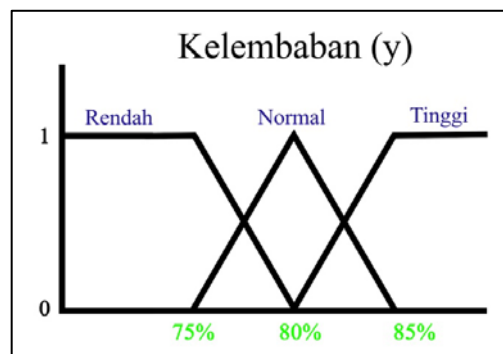
a. Temperatur (x) :



Gambar 3.2 Himpunan *Fuzzy* pada Temperatur

- Temperatur Dingin : [0 21]
- Temperatur Sejuk : [18 24]
- Temperatur Normal : [21 27]
- Temperatur Hangat : [24 30]
- Temperatur Panas : [27 40]

b. Kelembaban (y) :



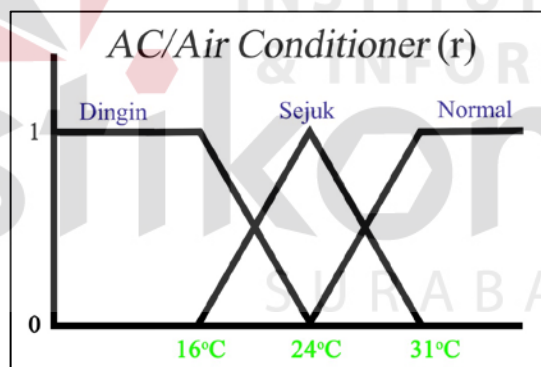
Gambar 3.3 Himpunan *Fuzzy* pada Kelembaban

- Kelembaban Rendah : [0 70]
- Kelembaban Normal : [70 80]
- Kelembaban Tinggi : [80 100]

2. Output pada Fuzzy Tsukamoto :

Pada fuzzy Tsukamoto ini terdapat tiga *output* yaitu *air conditioner*, lampu pijar, dan *humidifier*, berikut adalah pembagian tingkat dari *output*-an ketiga aktuator, yaitu pada Gambar 3.4 - Gambar 3.6 dan *rule-rule fuzzy* Tsukamoto yang telah ditentukan sebagai keluaran pada Tabel 3.1 - Tabel 3.3.

a. AC (*Air Conditioner*) (r)



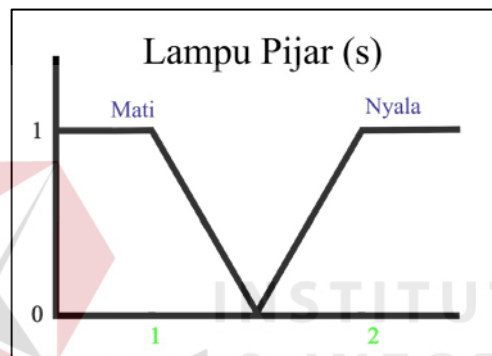
Gambar 3.4 Himpunan Fuzzy pada Output Air Conditioner (AC)

- AC Dingin : [0 24]
- AC Sejuk : [16 31]
- AC Normal : [24 40]

Tabel 3.1 Rule Fuzzy pada Output AC/Air Conditioner

| | Kelembaban Rendah | Kelembaban Normal | Kelembaban Basah |
|--------------------------|-------------------|-------------------|------------------|
| Temperatur Dingin | Sejuk | Sejuk | Normal |
| Temperatur Sejuk | Sejuk | Sejuk | Normal |
| Temperatur Normal | Sejuk | Normal | Normal |
| Temperatur Hangat | Dingin | Dingin | Sejuk |
| Temperatur Panas | Dingin | Dingin | Dingin |

b. Output Lampu Pijar (s)



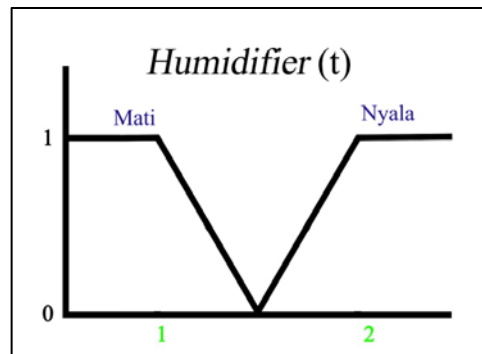
Gambar 3.5 Himpunan Fuzzy pada Output Lampu Pijar

- Lampu Pijar Mati : $[0 \quad 1.5]$
- Lampu Pijar Nyala : $[1.6 \quad 3]$

Tabel 3.2 Rule Fuzzy pada Output Lampu Pijar

| | Kelembaban Rendah | Kelembaban Normal | Kelembaban Basah |
|--------------------------|-------------------|-------------------|------------------|
| Temperatur Dingin | Nyala | Nyala | Nyala |
| Temperatur Sejuk | Nyala | Nyala | Nyala |
| Temperatur Normal | Mati | Mati | Nyala |
| Temperatur Hangat | Mati | Mati | Nyala |
| Temperatur Panas | Mati | Mati | Nyala |

c. *Humidifier* (t)



Gambar 3.6 Himpunan *Fuzzy* pada *Output Humidifier*

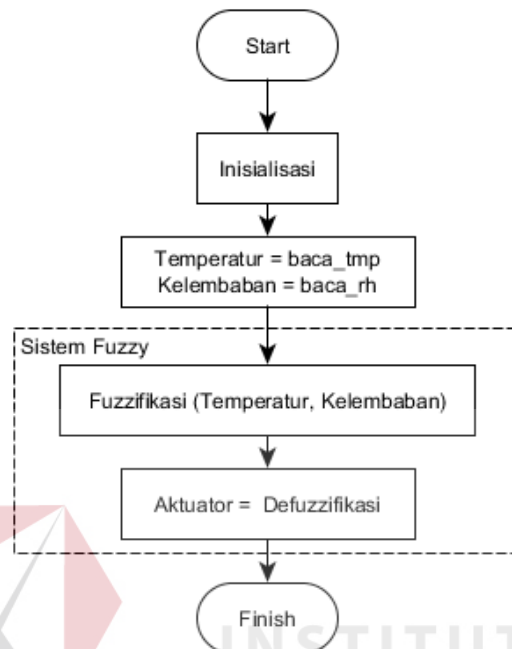
- *Humidifier* Mati : [0 2]
- *Humidifier* Nyala : [1 4]

Tabel 3.3 *Rule Fuzzy* pada *Output Humidifier*

| | Kelembaban Rendah | Kelembaban Normal | Kelembaban Basah |
|-------------------|-------------------|-------------------|------------------|
| Temperatur Dingin | Nyala | Mati | Mati |
| Temperatur Sejuk | Nyala | Mati | Mati |
| Temperatur Normal | Nyala | Mati | Mati |
| Temperatur Hangat | Nyala | Mati | Mati |
| Temperatur Panas | Nyala | Mati | Mati |

3. Flowchart Sistem Fuzzy

a. Kontrol Sistem Fuzzy

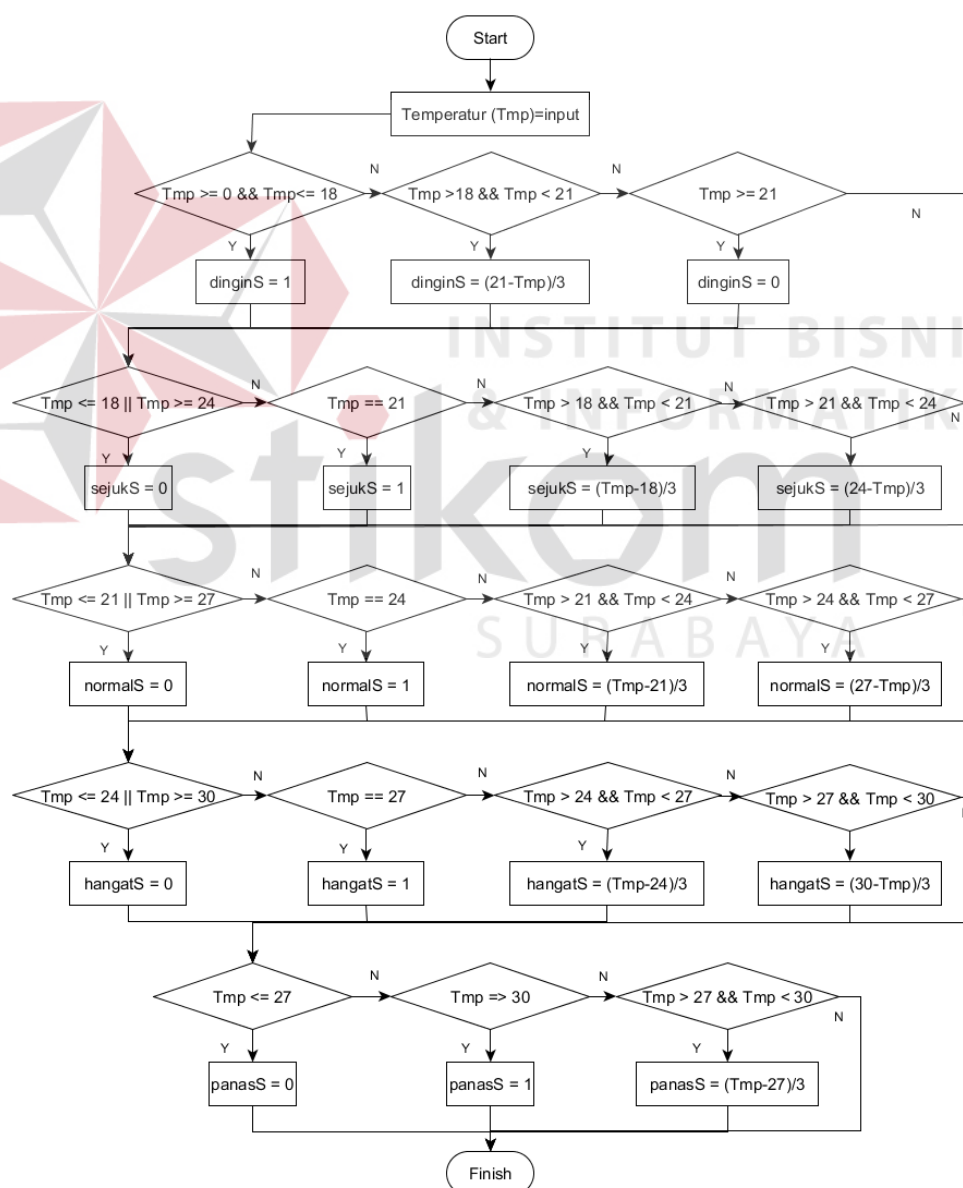


Gambar 3.7 Flowchart Kontrol Sistem Fuzzy

Pada Gambar 3.7 di atas pada *flowchart* kontrol sistem fuzzy hal pertama ialah inisialisai port pada modul, selanjutnya sensor SHT11 akan mendeteksi nilai temperatur serta kelembaban dan akan dikirimkan pada Arduino Mega 2560. Setelah menerima nilai temperatur serta kelembaban maka nilai tersebut dimasukkan kedalam variabel yang telah disediakan. Pada variabel itu nilai akan diolah menggunakan sistem fuzzy Tsukamoto dengan *rule* yang telah ditentukan. Dilakukan proses fuzzifikasi dan defuzzifikasi sehingga didapatkan nilai yang berfungsi sebagai keluaran yang menggerakkan aktuator LCD, AC, lampu pijar, dan *humidifier*. Ketika temperatur dan kelembaban sudah sesuai dengan nilai normal dan sesuai *rule* yang diharapkan maka, sistem akan berhenti karena telah terpenuhi.

b. Fuzzifikasi Temperatur

Pada proses fuzzyfikasi temperatur data yang akan diproses didapatkan dari pembacaan sensor SHT11, berupa nilai temperatur yang ada pada sekitar ruang lingkup sensor. Proses fuzzyfikasi berdasarkan *rule* yang telah ditentukan. Terdapat lima temperatur antara lain dingin, sejuk, normal, hangat, dan panas. Berikut *flowchart* fuzzifikasi temperatur pada Gambar 3.8.



Gambar 3.8 *Flowchart* Fuzzifikasi Temperatur

Pada kondisi pertama temperatur yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi dingin, jika suhu yang dibaca kurang dari sama dengan 18 derajat dan lebih dari sama dengan 0 derajat, maka *fuzzy* menganggap suhu pada sekitar sensor adalah dingin. Jika kondisi belum terpenuhi maka masuk kondisi kedua yaitu jika suhu yang dibaca antara lebih dari 18 derajat dan kurang dari 21 derajat, maka *fuzzy* akan mengolah suhu yang didapatkan dengan rumus turun pada *fuzzy*. Jika kondisi belum terpenuhi, akan di lanjutkan ke kondisi ketiga yaitu jika suhu lebih dari 21 derajat maka *fuzzy* menganggap temperatur sekitar sensor tidak termasuk kondisi dingin.

Pada kondisi kedua temperatur yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi sejuk, jika temperatur yang dibaca kurang dari sama dengan 18 derajat dan lebih dari sama dengan 24 derajat, maka *fuzzy* menganggap suhu pada sekitar sensor adalah tidak dalam kondisi sejuk. Jika kondisi belum terpenuhi maka masuk kondisi kedua yaitu jika temperatur yang dibaca sama dengan 21 maka *fuzzy* menganggap temperatur pada sekitar sensor termasuk dalam kondisi sejuk. Jika kondisi belum terpenuhi maka masuk kondisi ketiga yaitu jika temperatur yang dibaca antara lebih dari 18 derajat dan kurang dari 21 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus naik pada *fuzzy*. Jika kondisi belum terpenuhi maka masuk kondisi keempat yaitu jika temperatur yang dibaca antara lebih dari 21 derajat dan kurang dari 24 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus turun pada *fuzzy*.

Pada kondisi ketiga temperatur yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi normal, jika temperatur yang dibaca kurang dari sama dengan 21 derajat dan lebih dari sama dengan 27 derajat, maka *fuzzy* menganggap temperatur pada sekitar sensor adalah tidak dalam kondisi normal. Jika kondisi belum terpenuhi maka masuk pada kondisi kedua yaitu jika temperatur yang dibaca sama dengan 24 maka *fuzzy* menganggap temperatur pada sekitar sensor adalah normal. Jika kondisi belum terpenuhi maka masuk pada kondisi ketiga yaitu jika temperatur yang dibaca antara lebih dari 21 derajat dan kurang dari 24 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus naik pada *fuzzy*. Jika kondisi belum terpenuhi maka masuk kondisi keempat yaitu jika temperatur yang dibaca antara lebih dari 24 derajat dan kurang dari 27 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus turun pada *fuzzy*.

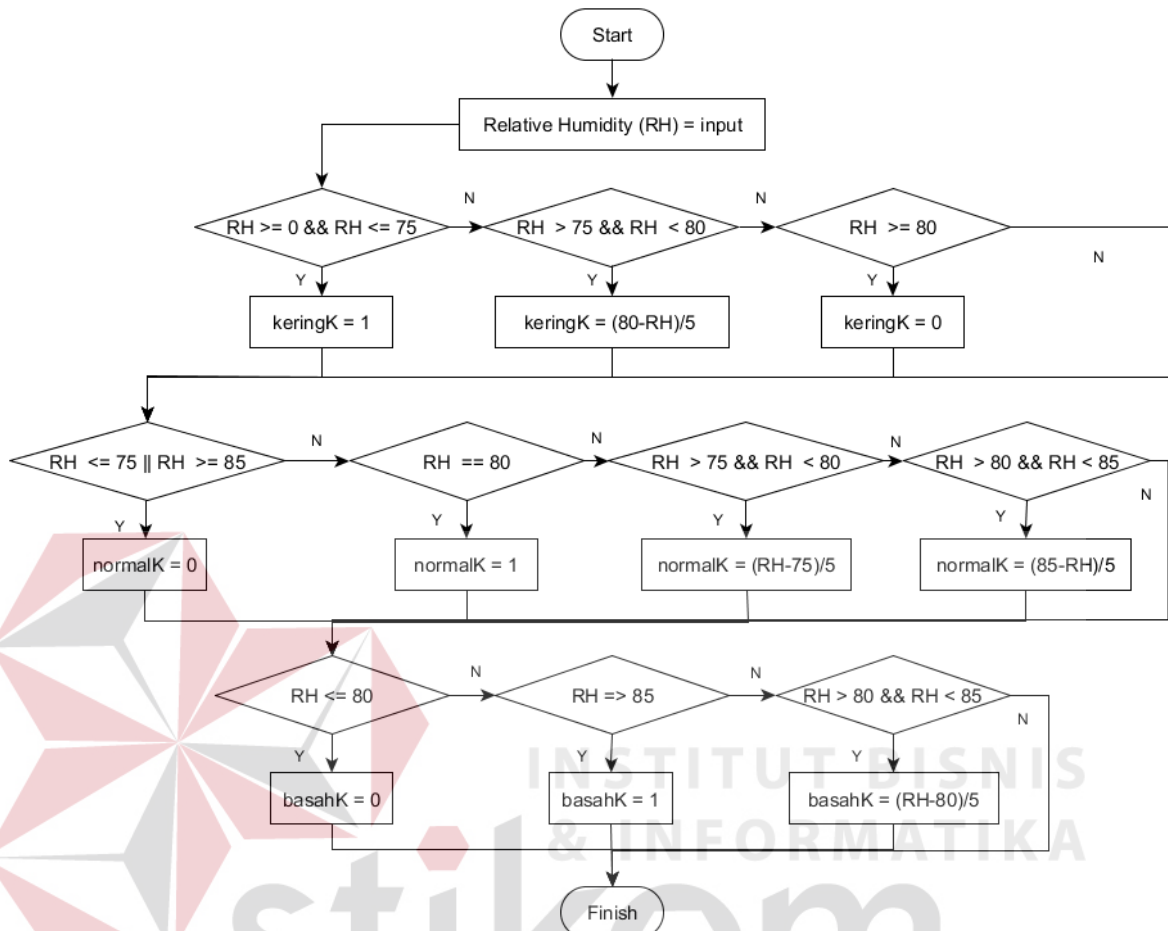
Pada kondisi keempat temperatur yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi hangat, jika temperatur yang dibaca kurang dari sama dengan 24 derajat dan lebih dari sama dengan 30 derajat, maka *fuzzy* menganggap temperatur pada sekitar sensor tidak dalam kondisi normal. Jika kondisi belum terpenuhi maka masuk kondisi kedua yaitu jika temperatur yang dibaca sama dengan 27 maka *fuzzy* menganggap temperatur pada sekitar sensor adalah kondisi normal. Jika kondisi belum terpenuhi maka masuk kondisi ketiga yaitu jika temperatur yang dibaca antara lebih dari 24 derajat dan kurang dari 27 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus naik pada *fuzzy*. Jika kondisi belum terpenuhi maka masuk kondisi keempat yaitu

jika temperatur yang dibaca antara lebih dari 27 derajat dan kurang dari 30 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus turun pada *fuzzy*.

Pada kondisi kelima temperatur yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi panas. Jika temperatur yang dibaca kurang dari 27 derajat maka *fuzzy* akan menganggap temperatur sekitar sensor tidak pada kondisi panas. Jika kondisi belum terpenuhi maka masuk kondisi kedua, jika temperatur yang dibaca lebih dari sama dengan 30 derajat maka *fuzzy* menganggap temperatur sekitar sensor dalam kondisi panas. Apabila kondisi belum terpenuhi maka akan masuk kondisi ketiga, jika temperatur yang dibaca lebih dari 27 derajat dan kurang dari 30 derajat, maka *fuzzy* akan mengolah temperatur yang didapatkan dengan rumus naik pada *fuzzy*.

c. Fuzzifikasi Kelembaban

Pada proses fuzzyfikasi kelembaban data yang akan diproses didapatkan pada pembacaan sensor SHT11, berupa nilai kelembaban yang ada pada sekitar sensor. Pemrosesan fuzzyfikasi berdasarkan *rule* yang telah ditentukan. Terdapat tiga kondisi kelembaban antara lain kering, normal, basah. Berikut *flowchart* fuzzifikasi kelembaban pada Gambar 3.9.



Gambar 3.9 Flowchart Fuzzifikasi Kelembaban

Pada kondisi pertama kelembaban yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi kering, jika kelembaban yang dibaca lebih dari sama dengan 0% dan kurang dari sama dengan 75%, maka *fuzzy* menganggap kelembaban pada sekitar sensor dalam kondisi kering. Jika kondisi belum terpenuhi maka masuk kondisi kedua yaitu jika kelembaban yang dibaca antara lebih dari 75% dan kurang dari 80%, maka *fuzzy* akan mengolah kelembaban yang didapatkan dengan rumus turun pada *fuzzy*. Jika kondisi belum terpenuhi, akan di

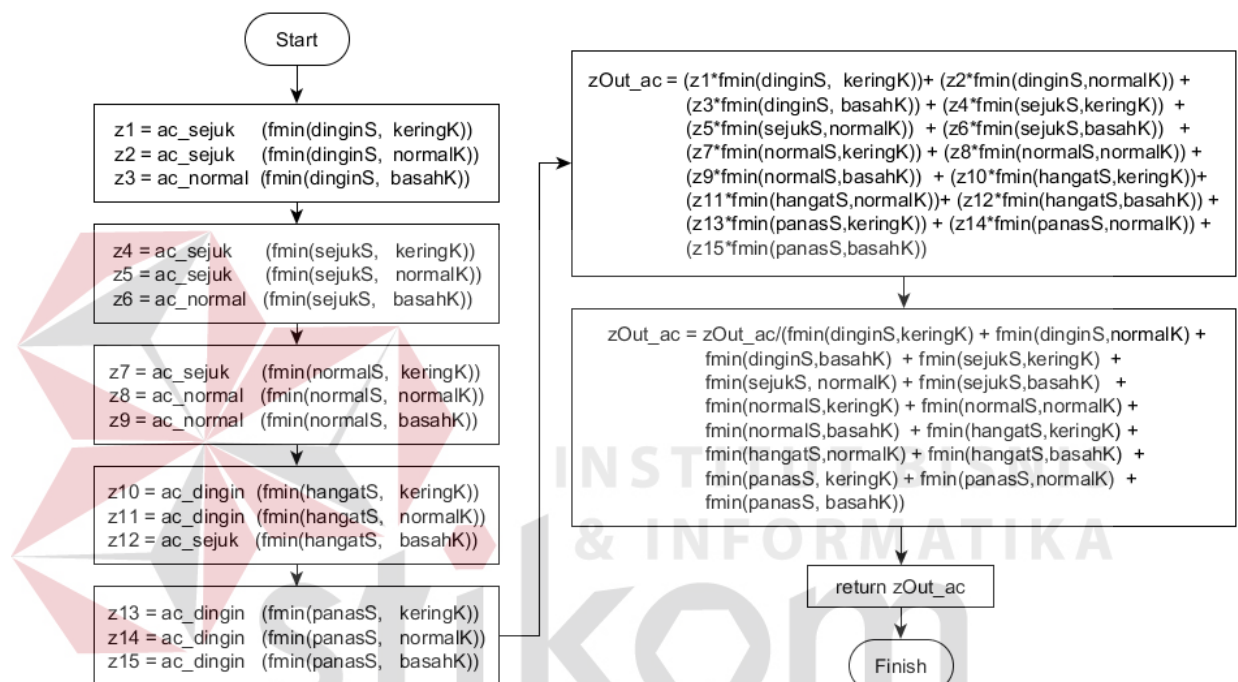
lanjutkan ke kondisi ketiga yaitu jika kelembaban lebih dari sama dengan 80% maka *fuzzy* menganggap kelembaban sekitar sensor tidak pada kondisi kering.

Pada kondisi kedua kelembaban yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi normal, jika kelembaban yang dibaca kurang dari sama dengan 75% dan lebih dari sama dengan 85%, maka *fuzzy* menganggap kelembaban pada sekitar sensor adalah tidak masuk kedalam kondisi normal. Jika kondisi belum terpenuhi maka masuk kondisi kedua yaitu jika kelembaban yang dibaca sama dengan 80% maka *fuzzy* menganggap kelembaban pada sekitar sensor adalah kondisi normal. Jika kondisi belum terpenuhi maka masuk kondisi ketiga yaitu jika kelembaban yang dibaca antara lebih dari 75% dan kurang dari 80%, maka *fuzzy* akan mengolah kelembaban yang didapatkan dengan rumus naik pada *fuzzy*. Jika kondisi belum terpenuhi maka masuk kondisi keempat yaitu jika kelembaban yang dibaca antara lebih dari 80% dan kurang dari 85%, maka *fuzzy* akan mengolah kelembaban yang didapatkan dengan rumus turun pada *fuzzy*.

Pada kondisi ketiga kelembaban yang dibaca akan masuk pada pemrosesan fuzzyfikasi kondisi basah. Jika kelembaban yang dibaca kurang dari 80% maka *fuzzy* akan menganggap kelembaban sekitar sensor tidak dalam kondisi basah. Jika kondisi belum terpenuhi maka masuk kondisi kedua, jika kelembaban yang dibaca lebih dari sama dengan 85% maka *fuzzy* menganggap kelembaban sekitar sensor adalah kondisi basah. Apabila kondisi belum terpenuhi maka akan masuk kondisi ketiga, jika kelembaban yang dibaca lebih dari 80% dan kurang dari 85%, maka *fuzzy* akan mengolah kelembaban yang didapatkan dengan rumus naik pada *fuzzy*.

d. Defuzzifikasi AC (*Air Conditioner*)

Data yang telah didapatkan dari proses fuzzyfikasi akan diproses di defuzzyfikasi. Pada proses ini berguna untuk mencari *Zout*, yang digunakan untuk pengolahan pemberian nilai temperatur *output* pada aktuatur AC.



Gambar 3.10 Flowchart Defuzzifikasi Air Conditioner

Pada Gambar 3.10 diatas proses defuzzyfikasi akan membandingkan nilai yang temperatur dan kelembaban dari proses fuzzyfikasi. Penetapan proses defuzzyfikasi ini telah ditetapkan oleh *rule* yang telah dibuat yaitu sebanyak 15 kondisi. Kemudian membandingkan nilai fuzzyfikasi terkecil antara setiap kondisi temperatur dan kondisi kelembaban, kemudian dikalikan nilai z1 sampai z15 dan dibagi dengan nilai pembanding yang terkecil antara temperatur dan kelembaban. Kemudian menghasilkan nilai *Zout* untuk AC yang digunakan untuk mengontrol

temperatur pada AC. Berikut ini adalah program mengenai defuzzifikasi pada aktuator *air conditioner*/AC :

```
float ac_dingin (float x)
{
    float z;
    z = -((x*8)-24);
    return z;
}

float ac_sejuk (float x)
{
    float z1, z2;

    z1 = (x*8)+16;
    z2 = -((x*7)-31);
    return fmax(z1,z2);
}

float ac_normal(float x)
{
    float z;
    z = (x*7)+24;
    return z;
}

float defuzzifikasi_ac()
{
    float z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13,
    z14, z15;
    float zOut_ac;

    z1 = ac_sejuk    (fmin(dinginS,  keringK));
    z2 = ac_sejuk    (fmin(dinginS,  normalK));
    z3 = ac_normal   (fmin(dinginS,  basahK));
    z4 = ac_sejuk    (fmin(sejukS,    keringK));
    z5 = ac_sejuk    (fmin(sejukS,    normalK));
    z6 = ac_normal   (fmin(sejukS,    basahK));
    z7 = ac_sejuk    (fmin(normalsS,  keringK));
    z8 = ac_normal   (fmin(normalsS,  normalK));
    z9 = ac_normal   (fmin(normalsS,  basahK));
    z10 = ac_dingin  (fmin(hangatS,   keringK));
    z11 = ac_dingin  (fmin(hangatS,   normalK));
    z12 = ac_sejuk   (fmin(hangatS,   basahK));
    z13 = ac_dingin  (fmin(panasS,    keringK));
    z14 = ac_dingin  (fmin(panasS,    normalK));
    z15 = ac_dingin  (fmin(panasS,    basahK));

    zOut_ac = (z1*fmin(dinginS,  keringK))+
    (z2*fmin(dinginS,normalK)) + (z3*fmin(dinginS, basahK)) +
    (z4*fmin(sejukS,keringK))  + (z5*fmin(sejukS,normalK)) +
    (z6*fmin(sejukS,basahK))   + (z7*fmin(normalsS,keringK)) +
    (z8*fmin(normalsS,normalK)) + (z9*fmin(normalsS,basahK)) +
    (z10*fmin(hangatS,keringK))+ (z11*fmin(hangatS,normalK))+
```



```

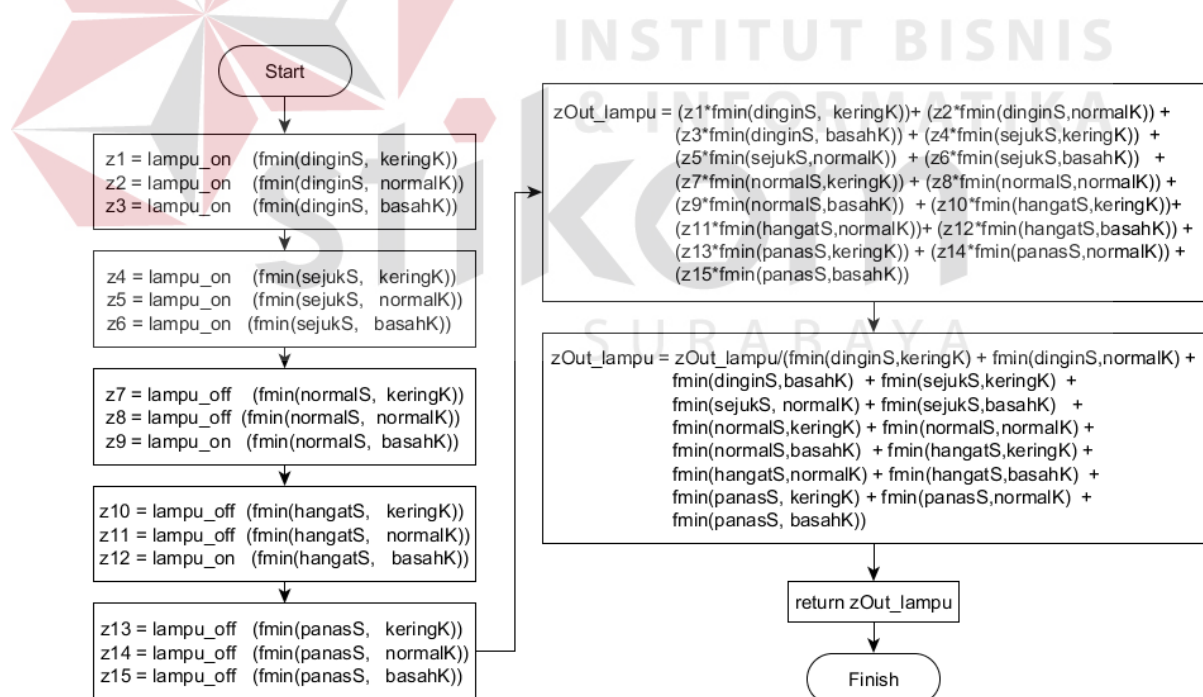
(z12*fmin(hangatS,basahK)) + (z13*fmin(panasS,keringK)) +
(z14*fmin(panasS,normalK)) + (z15*fmin(panasS,basahK));

zOut_ac = zOut_ac/(fmin(dinginS,keringK)+fmin(dinginS,normalK) +
fmin(dinginS,basahK) + fmin(sejukS,keringK) +
fmin(sejukS, normalK) + fmin(sejukS,basahK) +
fmin(normalS,keringK) + fmin(normalS,normalK) +
fmin(normalS,basahK) + fmin(hangatS,keringK) +
fmin(hangatS,normalK) + fmin(hangatS,basahK) +
fmin(panasS, keringK) + fmin(panasS,normalK) +
fmin(panasS, basahK));
Serial.print("A_Zout:");Serial.print(zOut_ac);Serial.println();
return zOut_ac;
}

```

e. Defuzzifikasi Lampu Pijar

Data yang telah didapatkan dari proses fuzzyfikasi akan diproses di defuzzifikasi. Pada proses ini berguna untuk mencari Zout, yang digunakan untuk pengolahan pemberian nilai lampu menyala atau mati.



Gambar 3.11 Flowchart Defuzzifikasi Lampu Pijar

Pada Gambar 3.11, proses defuzzifikasi lampu pijar akan membandingkan nilai temperatur dan kelembaban dari proses fuzzyfikasi. Penetapan proses

defuzzifikasi ini telah ditetapkan oleh *rule* yang telah dibuat yaitu sebanyak 15 kondisi. Kemudian membandingkan nilai fuzzyfikasi terkecil antara setiap kondisi temperatur dan kondisi kelembaban, kemudian dikalikan nilai z1 sampai z15 dan dibagi dengan nilai pembanding yang terkecil antara suhu dan kelembaban. Kemudian menghasilkan nilai *Zout* yang digunakan untuk mengontrol aktuator lampu pijar menyala atau matinya lampu. Berikut ini adalah program mengenai defuzzifikasi pada aktuator lampu pijar :

```
float lampu_off (float x)
{
    float z;
    z = -((x*0.5)-1.5);
    return z;
}

float lampu_on (float x)
{
    float z;
    z = (x*0.5)+1.5;
    return z;
}

float defuzzifikasi_lampu()
{
    float z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13,
    z14, z15;
    float zOut_lampu;

    z1 = lampu_on      (fmin(dinginS,   keringK));
    z2 = lampu_on      (fmin(dinginS,   normalK));
    z3 = lampu_on      (fmin(dinginS,   basahK));
    z4 = lampu_on      (fmin(sejukS,    keringK));
    z5 = lampu_on      (fmin(sejukS,    normalK));
    z6 = lampu_on      (fmin(sejukS,    basahK));
    z7 = lampu_off     (fmin(normalS,   keringK));
    z8 = lampu_off     (fmin(normalS,   normalK));
    z9 = lampu_on      (fmin(normalS,   basahK));
    z10 = lampu_off    (fmin(hangatS,   keringK));
    z11 = lampu_off    (fmin(hangatS,   normalK));
    z12 = lampu_on     (fmin(hangatS,   basahK));
    z13 = lampu_off    (fmin(panasS,    keringK));
    z14 = lampu_off    (fmin(panasS,    normalK));
    z15 = lampu_on     (fmin(panasS,    basahK));

    zOut_lampu = (z1*fmin(dinginS,   keringK))+
```

```

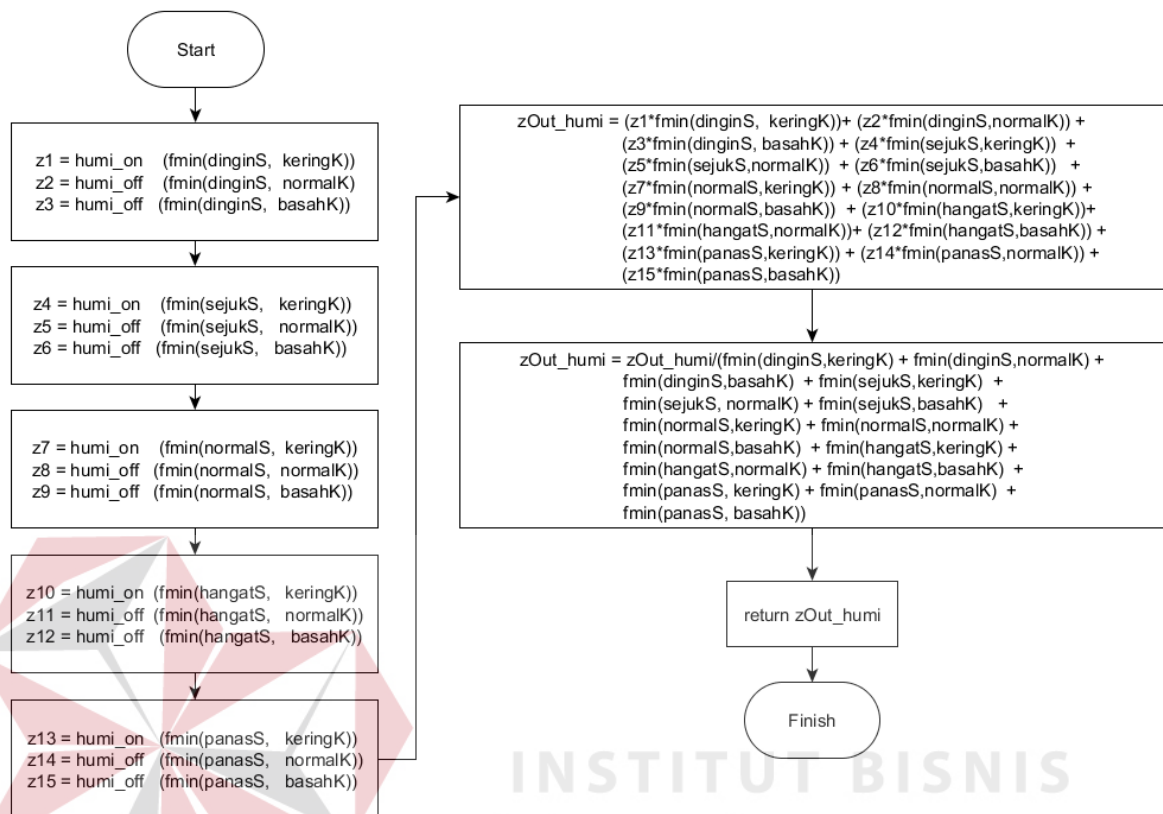
(z2*fmin(dinginS,normalK)) + (z3*fmin(dinginS, basahK)) +
(z4*fmin(sejukS,keringK)) + (z5*fmin(sejukS,normalK)) +
(z6*fmin(sejukS,basahK)) + (z7*fmin(normalS,keringK)) +
(z8*fmin(normalS,normalK)) + (z9*fmin(normalS,basahK)) +
(z10*fmin(hangatS,keringK))+ (z11*fmin(hangatS,normalK))+
(z12*fmin(hangatS,basahK)) + (z13*fmin(panasS,keringK)) +
(z14*fmin(panasS,normalK)) + (z15*fmin(panasS,basahK));

zOut_lampu = zOut_lampu/(fmin(dinginS,keringK) +
                          fmin(dinginS,normalK) +
                          fmin(dinginS,basahK) + fmin(sejukS,keringK) +
                          fmin(sejukS, normalK) + fmin(sejukS,basahK) +
                          fmin(normalS,keringK) + fmin(normalS,normalK) +
                          fmin(normalS,basahK) + fmin(hangatS,keringK) +
                          fmin(hangatS,normalK) + fmin(hangatS,basahK) +
                          fmin(panasS, keringK) + fmin(panasS,normalK) +
                          fmin(panasS, basahK));
Serial.print("L_Zout:");Serial.print(zOut_lampu);Serial.println();
return zOut_lampu;
}

```

f. Defuzzifikasi *Humidifier*

Data yang telah didapatkan dari proses fuzzyfikasi akan diproses di Defuzzyfikasi. Pada Gambar 3.12 proses defuzzifikasi *humidifier* ini berguna untuk mencari Zout, yang digunakan untuk pengolahan pemberian nilai aktuator *humidifier* menyala atau mati.



Gambar 3.12 Flowchart Defuzzifikasi Humidifier

Pada proses defuzzyfikasi akan membandingkan nilai yang temperatur dan kelembaban dari proses fuzzyfikasi. Penetapan proses defuzzyfikasi ini telah ditetapkan oleh *rule* yang telah dibuat yaitu sebanyak 15 kondisi. Kemudian membandingkan nilai fuzzyfikasi terkecil antara setiap kondisi temperatur dan kondisi kelembaban, kemudian dikalikan nilai z1 sampai z15 dan dibagi dengan nilai pembanding yang terkecil antara suhu dan kelembaban. Kemudian menghasilkan nilai *Zout* yang digunakan untuk mengontrol menyala atau matinya *Humidifier*. Berikut ini adalah program mengenai defuzzifikasi pada aktuator *humidifier* :

```
float humi_off (float x)
{
```

```

float z;
z = -((x*0.5)-1.5));
return z;
}

float humi_on (float x)
{
    float z;
    z = (x*0.5)+1.5;
    return z;
}

float defuzzifikasi_humi()
{
    float z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13,
    z14, z15;
    float zOut_humi;

    z1 = humi_on    (fmin(dinginS,  keringK));
    z2 = humi_off   (fmin(dinginS,  normalK));
    z3 = humi_off   (fmin(dinginS,  basahK));
    z4 = humi_on    (fmin(sejukS,    keringK));
    z5 = humi_off   (fmin(sejukS,    normalK));
    z6 = humi_off   (fmin(sejukS,    basahK));
    z7 = humi_on    (fmin(normalS,   keringK));
    z8 = humi_off   (fmin(normalS,   normalK));
    z9 = humi_off   (fmin(normalS,   basahK));
    z10 = humi_on   (fmin(hangatS,   keringK));
    z11 = humi_off  (fmin(hangatS,   normalK));
    z12 = humi_off  (fmin(hangatS,   basahK));
    z13 = humi_on   (fmin(panasS,    keringK));
    z14 = humi_off  (fmin(panasS,    normalK));
    z15 = humi_off  (fmin(panasS,    basahK));

    zOut_humi = (z1*fmin(dinginS,  keringK)) +
    (z2*fmin(dinginS,normalK)) + (z3*fmin(dinginS,basahK)) +
    (z4*fmin(sejukS,keringK))  + (z5*fmin(sejukS,normalK)) +
    (z6*fmin(sejukS,basahK))   + (z7*fmin(normalS,keringK)) +
    (z8*fmin(normalS,normalK)) + (z9*fmin(normalS,basahK)) +
    (z10*fmin(hangatS,keringK)) + (z11*fmin(hangatS,normalK)) +
    (z12*fmin(hangatS,basahK)) + (z13*fmin(panasS,keringK)) +
    (z14*fmin(panasS,normalK)) + (z15*fmin(panasS,basahK));
    zOut_humi = zOut_humi/(fmin(dinginS,keringK) +
    fmin(dinginS,normalK) +
    fmin(dinginS,basahK) + fmin(sejukS,keringK) +
    fmin(sejukS, normalK) + fmin(sejukS,basahK) +
    fmin(normalS,keringK) + fmin(normalS,normalK) +
    fmin(normalS,basahK) + fmin(hangatS,keringK) +
    fmin(hangatS,normalK) + fmin(hangatS,basahK) +
    fmin(panasS, keringK) + fmin(panasS,normalK) +
    fmin(panasS, basahK));
    Serial.print("H_Zout: ");Serial.print(zOut_humi);Serial.println();
    return zOut_humi;
}

```

3.3 Pengujian Sistem Keseluruhan

Pada sistem yang dirancang diuji kinerja dari beberapa komponen serta pengujian sistem keseluruhan dengan menggabungkan perangkat keras dan perangkat lunak.

3.3.1 Pengujian Arduino Mega2560

Pada pengujian Arduino Mega 2560, dilakukan dengan memasukkan program perintah sederhana kedalam Arduino dengan menggunakan *software* Arduino IDE. Arduino dan program yang baik dapat mengeksekusi dengan hasil yang baik.

Tujuan dilakukan pengujian ini apakah pada Arduino yang digunakan pada penelitian tidak mengalami kerusakan dan kegagalan pada saat mengeksekusi program. Sehingga pada saat Arduino digunakan dapat berjalan dengan baik dan lancar. Berikut alat yang dilakukan pada pengujian, antara lain :

- a. *PC (Personal Computer)/Laptop*
- b. Arduino Mega 2560
- c. Kabel USB
- d. *Software* Arduino IDE

Berikut ini langkah-langkah pada prosedur pengujian Arduino Mega 2560, sebagai berikut :

- a. Hidupkan *PC/laptop*.
- b. Sambungkan *PC/laptop* pada Arduino Mega 2560 dengan melalui kabel *usb*.
- c. Pada *PC/laptop* buka *software* Arduino IDE. Isi program perintah dalam bahasa C

pada Arduino IDE. Berikut contoh program pada Arduino IDE, sebagai berikut :

```
int a;
void setup()
{
    Serial.begin(9600);
    Serial.println("Tes Arduino");
}
void loop()
{
    for(a=5; a>=0; a--)
    {
        Serial.print("Angka ke : "); Serial.println(a);
        delay(1000);
    }
    while(1){};
}
```

d Setelah selesai mengetikkan program perintah maka pada tekan *icon* berbentuk centang *"Verify"* untuk mengecek apakah terdapat kesalahan pada perintah program yang telah dibuat. Program dicek dalam bahasa C. Selanjutnya *setting board* dengan memilih Arduino Mega 2560 pada kolom *"Tools"*, lalu *setting port* arduinoyang terdeteksi oleh komputer/PC. Berikut tekan *icon* berbentuk arah kekanan/ *"Upload"* untuk meng-*upload* program kedalam Arduino Mega 2560.

e. Apabila program telah berhasil di-*upload*, maka tekan *icon* *"Serial Monitor"* pada kanan atas. Maka akan tampil jendela yang berisikan hasil dari *serial* yang dicetak. Amati hasil yang dilakukan oleh Arduino Mega 2560 pada Gambar 4.1 dan Gambar 4.2.

3.3.2 Pengujian Sensor Modul SHT11

Sensor modul SHT11 digunakan sebagai pendeteksi temperatur dan kelembaban pada ruang lingkup sekitar sensor. Dengan sensor ini dapat menghasilkan nilai dan menampilkan pada jendela *serial monitor* atau dengan aktuator LCD. Pada pengujian ini Arduino Mega 2560 diberikan

program perintah yang dapat membaca sensor modul SHT11 agar dapat mengetahui nilai yang ada di ruang lingkup sekitar.

Pada pengujian ini bertujuan untuk melihat kinerja sensor modul SHT11 dapat mendeteksi nilai temperatur dan nilai kelembaban di sekitar ruang lingkup. Berikut alat yang dilakukan pada pengujian, antara lain :

- a. *PC (Personal Computer)/Laptop*
- b. Arduino Mega 2560
- c. Kabel USB
- d. Kabel *Jumper*
- e. Sensor Modul SHT11
- f. *Software* Arduino IDE

Berikut ini langkah-langkah pada prosedur pengujian sensor modul SHT11, sebagai berikut :

- a. Hubungkan antara sensor modul SHT11, pin data, *clock*, *power* dan *ground* sesuai dengan *direction*-nya pada Arduino Mega 2560 menggunakan kabel *jumper*.
- b. Hidupkan PC/laptop.
- c. Sambungkan PC/laptop pada Arduino Mega 2560 dengan melalui kabel *usb*.
- d. Pada PC/laptop buka *software* Arduino IDE. Isi program perintah dalam bahasa C pada Arduino IDE. Berikut program pada Arduino IDE :

```
#include <SHT1x.h>
#define dataPin 20
#define clockPin 21

SHT1x sht(dataPin, clockPin);

void setup()
{
  Serial.begin(9600);
  Serial.println("Arduino SHT11");
}
```



```

delay(200);
}

void loop()
{
float temp_c;
float humidity;
temp_c = sht.readTemperatureC();
humidity = sht.readHumidity();
Serial.print("Temperatur: ");
Serial.print(temp_c);
Serial.println(" celsius");
Serial.print("Kelembaban: ");
Serial.print(humidity);
Serial.println("%");
delay(1000);
}

```

- e. Setelah selesai tekan icon "Verify" pada *toolbars*, jika tidak ada kesalahan pada *syntax* maka lakukan *upload* pada program yang telah dibuat . Jika telah selesai maka tekan icon "Serial Monitor".
- f. Jendela *serial monitor* akan tampil dan hasil dari program yang telah di-*upload* akan seperti pada Gambar 4.3.
- g. Amati hasil nilai temperatur dan kelembaban yang terbaca oleh sensor yang ditampilkan oleh jendela *serial monitor*.

3.3.3 Pengujian Aktuator Air Conditioner (AC)

Pada pengujian *air conditioner* (AC) ini akan mengatur temperatur pada ruang lingkup otomasi yang akan di-*remote* oleh *infra red* dan temperatur yang keluar akan dideteksi oleh sensor SHT11.

Pengujian aktuator *air conditioner* ini bertujuan untuk melihat seberapa baik kinerja AC dan mengetahui keluaran sistem dalam menuju *setpoint* yang diinginkan sehingga keluaran yang dihasilkan sesuai dengan kebutuhan serta

keinginan pada penelitian. Berikut alat yang dilakukan pada pengujian, antara lain :

- a. *PC (Personal Computer)/Laptop*
- b. *Arduino Mega 2560*
- c. *Kabel USB*
- d. *Kabel Jumper*
- e. *Sensor Infra Red Transmitter*
- f. *Sensor SHT11*
- g. *Air Conditioner*
- h. *Software Arduino IDE*

Berikut ini langkah-langkah pada prosedur pengujian aktuator *air conditioner*, sebagai berikut :

- a. Hubungkan antara pin pada sensor *infra red transmitter* serta sensor SHT11 sesuai dengan *direction*-nya pada *Arduino Mega 2560* menggunakan kabel *jumper*.
- b. Hidupkan *PC/laptop*.
- c. Sambungkan *PC/laptop* pada *Arduino Mega 2560* dengan melalui kabel *usb*.
- d. Pada *PC/laptop* buka *software Arduino IDE*. Isi program perintah dalam bahasa C

pada *Arduino IDE*. Berikut program pada *Arduino IDE* :

```
#include <SHT1x.h>
#include <IRremote.h>
IRsend irsend;
#define dataPin 20
#define clockPin 21
SHT1x sht(dataPin, clockPin);
unsigned long wkt;
float suhusht;

void setup()
{
  Serial.begin(9600);
```

```

suhusht = sht.readTemperatureC();
Serial.print("suhu= ");
Serial.print(suhusht);
Serial.print("\t");
wkt = millis();
Serial.print("wkt= ");
Serial.println(wkt);
int khz = 38;
unsigned int irSignal[100] = {3450,3500, 900,2500, 900,900,
900,850, 900,2550,      900,850, 900,2550, 900,2500, 900,900,
900,2500, 900,900, 900,850,   900,2550, 900,850, 900,2550,
900,2550, 900,850, 900,850, 900,2550, 900,900, 850,2550, 900,900,
900,850, 900,850, 900,900, 900,850, 900,2550, 900,850, 900,2550,
900,850, 900,900, 900,850, 900,900, 3450,3450, 900,2550, 900,850,
900,900, 900,2500, 900,900, 900,2500, 900,2550, 900,900, 900,2500,
900,900, 900,850, 900,2550, 900,850, 900,2550, 900,2550, 900,850
}; // 96965050
irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]),
khz);
}
void loop()
{
    suhusht = sht.readTemperatureC();
    Serial.print("suhu= ");
    Serial.print(suhusht);
    Serial.print("\t");
    wkt = millis();
    Serial.print("wkt= ");
    Serial.println(wkt);
}

```

- e. Setelah selesai tekan icon "Verify" pada *toolbars*, jika tidak ada kesalahan pada *syntax* maka lakukan *upload* pada program yang telah dibuat . Jika telah selesai maka tekan icon "Serial Monitor".
- f. Jendela *serial monitor* akan tampil. Amati hasil nilai sensor SHT11 yang terbaca pada jendela *serial monitor*.

3.3.4 Pengujian Otomasi Sistem

Pengujian ini merupakan pengambilan data dan hasil pada otomasi sistem yang telah dirancang. Mengolah input serta diproses menggunakan metode fuzzy Tsukamoto sehingga menghasilkan sebuah output yang dapat mengatur temperatur serta kelembaban pada ruangan.

Tujuan dari pengujian ini sensor dapat mengontrol temperatur serta kelembaban dengan menggunakan metode logika fuzzy Tsukamoto pada otomasi sistem hidroponik yang dirancang. Dengan pengambilan data pada pagi, siang dan, sore hari bertujuan melihat perbedaan pada kinerja otomasi sistem. Berikut alat yang dilakukan pada pengujian, antara lain :

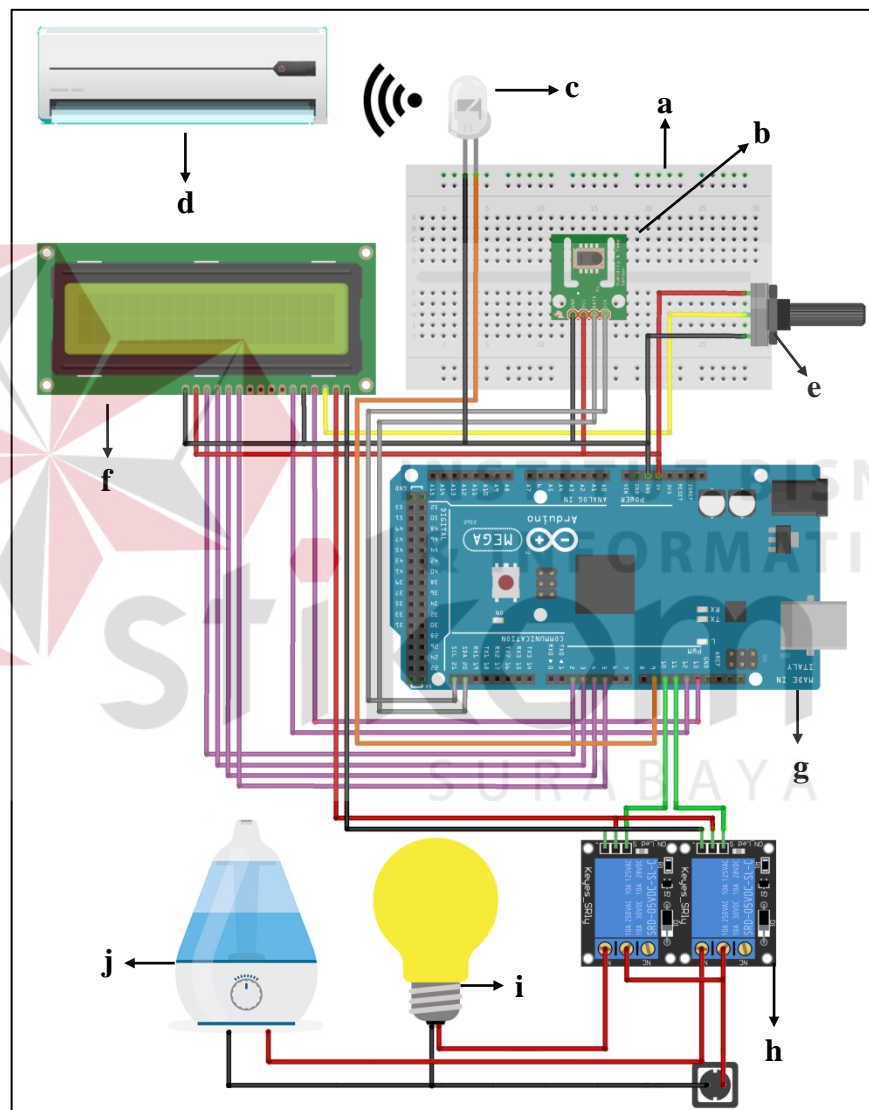
- a. *PC (Personal Computer)/Laptop*
- b. Arduino Mega 2560
- c. Kabel USB
- d. Kabel *Jumper*
- e. Sensor *Infra Red Transmitter*
- f. Sensor SHT11
- g. *Air Conditioner*
- h. Lampu Pijar
- i. *Humidifier*
- j. Modul *Relay*
- k. LCD
- l. *Software Arduino IDE*

Berikut ini langkah-langkah pada prosedur pengujian otomasi sistem, sebagai berikut :

- a. Hubungkan komponen antara Arduino Mega 2560 menggunakan kabel *jumper* sesuai dengan *direction* pin-pinnya.
- b. Hidupkan PC/laptop.
- c. Sambungkan PC/laptop pada Arduino Mega 2560 dengan melalui kabel *usb*.

- d Pada PC/laptop buka *software* Arduino IDE. Isi program perintah dalam bahasa C pada Arduino IDE dan *upload* program. Program terdapat pada LAMPIRAN 1.

3.4 Rangkaian pada Otomasi Sistem



Gambar 3.13 Rangkaian Otomasi Sistem

Keterangan pada Gambar 3.13, sebagai berikut :

a = Project Board

b = Sensor Modul SHT11

c = Sensor Infra Red

d = Air Conditioner (AC)

e = Variable Resistor

f = Liquid Crystal Display (LCD)

g = Arduino Mega 2560

h = Modul Relay

i = Lampu Pijar

j = Humidifier

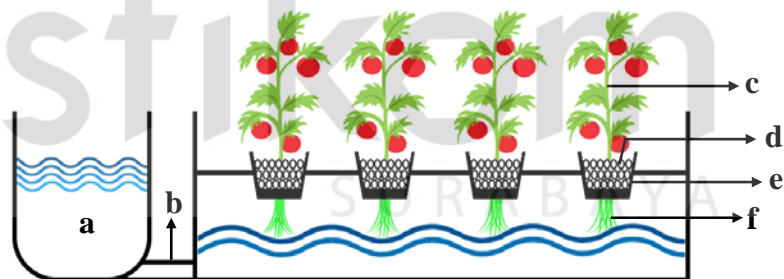
Pada Gambar 3.13 di atas adalah rangkaian sistem yang mana akan dipasang pada rancang bangun otomasi hidroponik. Sebagai pusat kendali menggunakan Arduino Mega 2560, sedangkan input terdapat modul sensor SHT11 serta terdapat hasil keluaran yang memerintah aktuator berupa LCD, sensor *infra red* yang akan *me-remote* AC, dan dua buah *relay* yang akan menghidupkan serta mematikan lampu pijar dan *humidifier*. Berikut ini pin-pin yang menghubungkan antar rangkaian pada Tabel 3.4.

Tabel 3.4 Rangkaian Pin pada Otomasi Sistem

| | LCD | Sensor SHT 11 | Relay 1 (Lampu Pijar) | Relay 2 (Humidifier) | Sensor Infra Red |
|-------------------|------------------------------|------------------------------|--------------------------|-------------------------|---------------------|
| Arduino Mega 2560 | Pin 13 = Pin 4 (RS) | Pin 20 (SDA) = Pin 1 (Data) | Pin 10 = Vin | Pin 11 = Vin | |
| | Ground = Pin 5 (R/W) | | | | |
| | Pin 12 = Pin 6 (Enable) | | | | |
| | Pin 5 = Pin 11 (D4) | Pin 21 (SCL) = Pin 3 (Clock) | GND = GND | GND = GND | |
| | Pin 4 = Pin 12 (D5) | | | | |
| | Pin 3 = Pin 13 (D6) | | | | |
| | Pin 2 = Pin 14 (D7) | GND = Pin 4 (GND) | 5V = VCC | 5V = VCC | |
| | GND = Pin 1 (GND) | | | | |
| | 5V = Pin 2 (VCC) | | | | |
| | 5V = Pin 15 (Backlight+) | 5V = Pin 8 (VCC) | | | |
| | Ground = Pin 16 (Backlight-) | | | | |
| | | | | | Pin 9 = Data |
| | | | | | GND = GND |
| Variabel Resistor | Out = Pin 3 (V0) | | | | |

3.5 Skema Sistem Hidroponik Wick

Berikut ini skema sistem hidroponik wick pada Gambar 3.14.



Gambar 3.14 Skema Sistem Hidroponik Wick

Keterangan pada Gambar 3.14 :

a = Tandon Air

b = Pipa

c = Tanaman Tomat Cherry

d = Rockwool/Sekam Bakar

e = Pot Tanaman

f = Sumbu/Kain *Flanel*

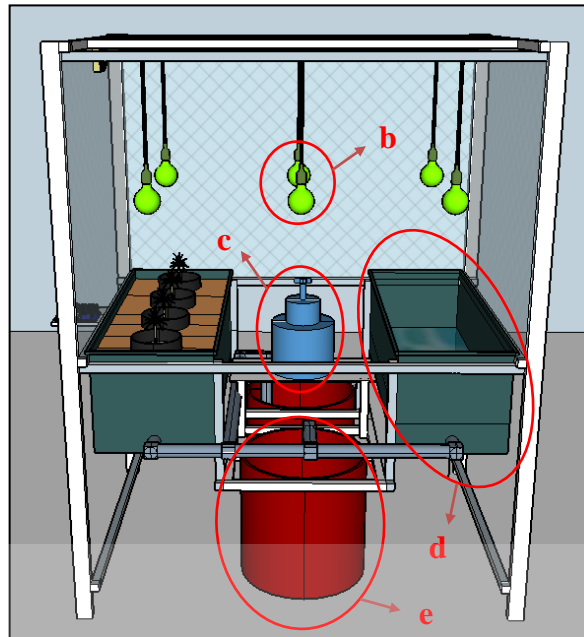
Pada Gambar 3.14 di atas adalah skema sistem hidroponik *wick* yang mana pada tandon akan mengalirkan air dengan campuran nutrisi mineral yang akan diserap oleh sumbu dan diterima oleh sekam bakar/*rockwool* lalu diserap oleh tanam tomat cherry.

3.6 Model Perancangan

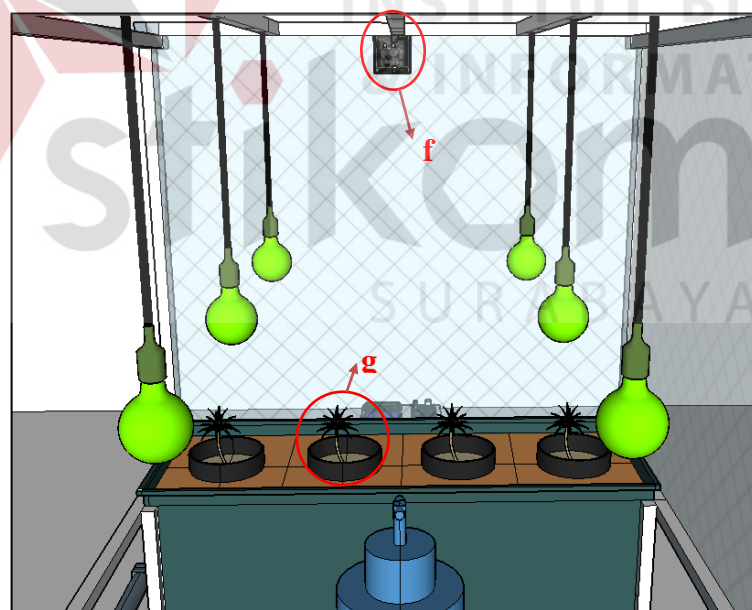
Berikut ini desain rancang otomasi sistem hidroponik pada Gambar 3.15 - Gambar 3.18.



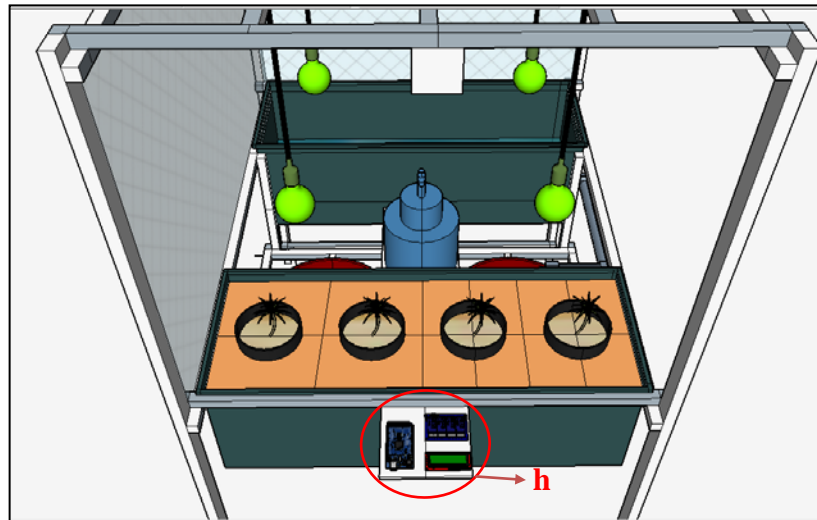
Gambar 3.15 Desain Otomasi Sistem Hidroponik Tampak Keseluruhan



Gambar 3.16 Desain Otomasi Sistem Hidroponik Tampak Depan



Gambar 3.17 Desain Otomasi Sistem Hidroponik Tampak Dalam

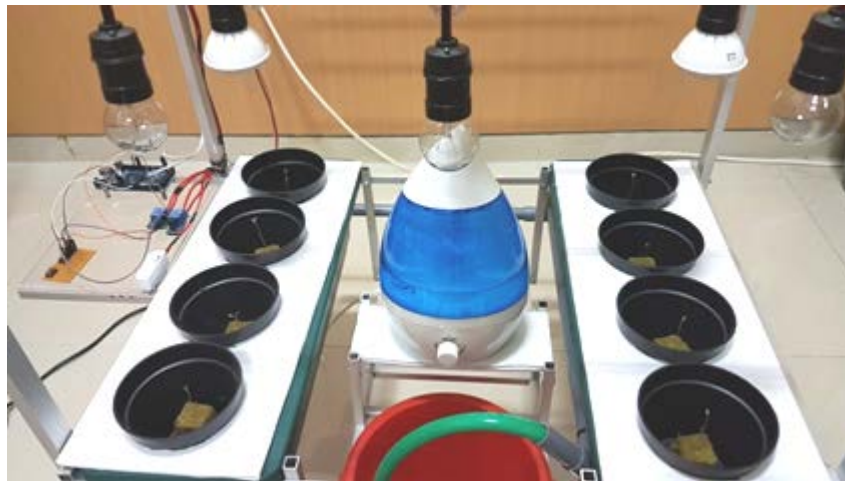


Gambar 3.18 Desain Otomasi Sistem Hidroponik Tampak Samping

Berikut ini hasil rancang otomasi sistem hidroponik pada Gambar 3.19 dan Gambar 3.20.



Gambar 3.19 Rancang Otomasi Sistem Hidroponik Tampak Keseluruhan



Gambar 3.20 Rancang Otomasi Sistem Hidroponik Tampak Dalam

Keterangan pada Gambar 3.15 sampai dengan Gambar 3.18 :

a = AC (*Air Conditioner*)

b = Lampu Pijar

c = *Humidifier*

d = Pot *Rectangular*

e = Timba dan Tangki air

f = Sensor Modul SHT11

g = Pot dan Benih Tumbuhan

h = Panel Kontrol

3.6.1 Ukuran Dimensi Rancang Bangun Hidroponik

Pada saat setelah pemasangan komponen telah dilakukan maka dihasilkan dimensi dari rancang bangun, berikut ukuran tersebut :

1. Panjang Rancang bangun : 72 cm
2. Lebar rancang bangun : 55 cm

3. Tinggi rancang bangun : 90 cm
4. Luas rancang bangun : $356,4 \text{ cm}^3$

3.6.2 Struktur Material Rancang Bangun Hidroponik

Dalam penelitian ini bahan material yang digunakan adalah, sebagai berikut :

A. Bagian rancang bangun hidroponik

1. Almini
2. Baut, Mur dan *Ring*
3. Tangki
4. Pot
5. *Rockwool*
6. Kain sumbu
7. Pipa *PVC*

B. Bagian Elektro

1. Arduino Mega 2560
2. Modul SHT11
3. AC (*Air Conditioner*)
4. Lampu Pijar
5. *Humidifier*
6. LCD
7. IR (*Infra Red*)



3.6.3 Rincian Harga Material dan BEP (*Break Event Point*)

Pada perancangan didapatkan rincian harga material dan BEP yang dibutuhkan untuk menunjang budidaya rancang bangun otomasi sistem hidroponik untuk tomat cherry, rincian tersebut dapat dilihat pada LAMPIRAN 2.

