

BAB III

METODE PENELITIAN

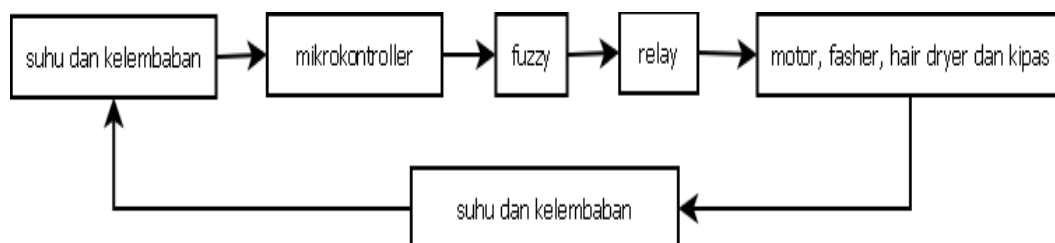
Perancangan perangkat keras dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari internet, dan konsep-konsep teoretis dari buku-buku penunjang. Setelah literatur terkumpul barulah dilakukan perancangan perangkat keras yang dilanjutkan dengan perancangan dan pembuatan perangkat lunak yaitu program sederhana untuk melakukan pengujian pada tiap bagian perangkat keras yang telah dibuat dan program untuk menjalankan perangkat keras secara utuh. Perancangan perangkat keras dan lunak dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari *internet*, dan kosep-konsep teoritis dari buku-buku penunjang.

Dari data-data yang diperoleh maka dilakukan perancangan rangkaian perangkat keras dan perangkat lunak. Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dirancang, kemudian melakukan kolaborasi perangkat keras dengan kerja perangkat lunak yang telah selesai dirancang.

Pada bab ini akan dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (*hardware*) maupun perangkat lunak (*software*). Dari kedua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem yang baik.

Perancangan ini diperlukan sebelum proses pembuatan sistem tersebut, perancangan ini berguna agar pengerjaan tahapan selanjutnya berjalan dengan lancar. Tahapan-tahapan meliputi tahap pembuatan perangkat keras, perangkat lunak dan menggabungkan keduanya.

3.1 Perancangan Perangkat Keras



Gambar 3.1 Blok Diagram Sistem Keseluruhan

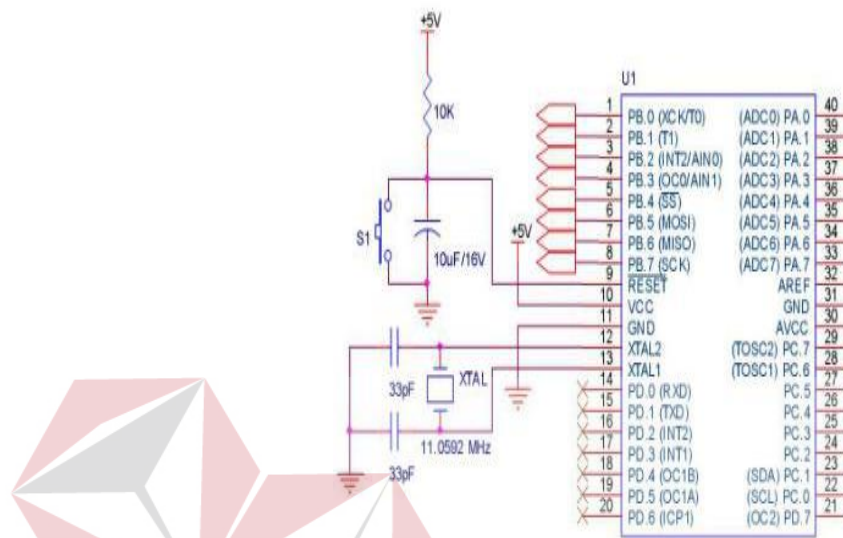
Dalam sistem diatas mikrokontroller sebagai pemroses akan mendapatkan input dari sensor suhu dan kelembaban. Sensor suhu dan kelembaban akan memberikan data tentang suhu dan kelembaban pada tempat tinggal cacing (*Lumbricus Rubellus*). Setelah itu data yang diberikan oleh sensor suhu dan kelembaban pada mikrokontroller kemudian diolah. Hasil dari pengolahan data tersebut kemudian digunakan untuk menggerakkan motor dan fan.

3.1.1 Rangkaian Mikrokontroller

pada proyek akhir ini dibuat piranti pengendali menggunakan *mikrokontroller* keluaran AVR, yaitu Atmega32. Untuk mengaktifkan atau menjalankan *mikrokontroller* ini diperlukan rangkaian minimum sistem. Rangkaian minimum sistem tersebut terdiri rangkaian *reset* dan *rangkaian osilator*.

Untuk menjalankan *mikrokontroller* ini dibutuhkan sebuah rangkaian agar *mikrokontroller* tersebut dapat bekerja dengan baik. Rangkaian *mikrokontroller* terdiri dari rangkaian *reset* dan *rangkaian osilator*.

Pada Pin VCC diberi masukan tegangan operasi berkisar antara 4.5 Volt sampai dengan 5,5 Volt. Pin RST mendapatkan *input* dari *manual reset*. Rangkaian minimum sistem dapat terlihat pada gambar 3.2



Gambar 3.2 Rangkaian minimum sistem Atmega32

Berdasarkan Gambar diatas. *Pin* VCC diberi tegangan operasi yang besaran 5 volt. *Port A*, digunakan sebagai inputan dari sensor suhu dan kelembaban. *Port B*, digunakan untuk menampilkan suhu dan kelembaban menggunakan media LCD. *Port D*, digunakan sebbagai *port* yang terhubung pada motor DC 12 volt. Dalam gambar 5.3 diatas, *pin* XTAL1 dan XTAL2 dihubungkan dengan komponen XTAL sebesar 8,0000 MHz. Pemilihan *frekuensi osilasi* dari XTAL tersebut berdasarkan penggunaan *microkontroller* agar setiap *clock mikrokontroller* berlangsung setiap 1us. Berikut adalah perhitungan besar *clock cycle* yang dipakai:

$$1 \text{ siklus mesin} = 8 \text{ clock}$$

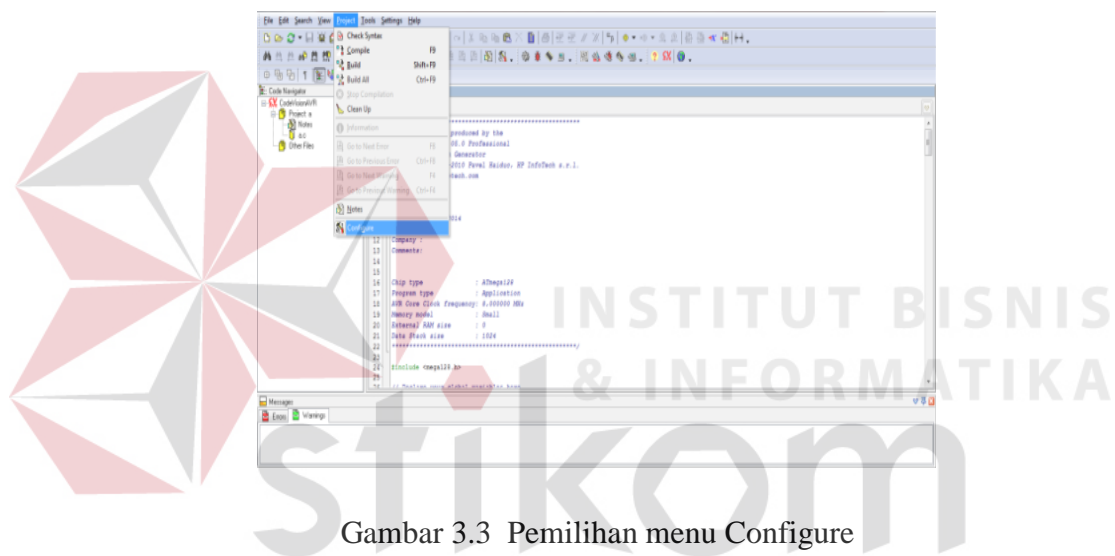
$$T(s) = 1 / F \dots\dots\dots (3.1)$$

$$= \frac{1}{1/8 \times \text{Frekuensi Kristal}}$$

$$\begin{aligned}
 &= \frac{1}{1/8 \times 8 \text{ MHz}} \\
 &= 1 \times 10^{-6} \\
 &= 1 \mu\text{s}
 \end{aligned}$$

Untuk melakukan proses *download* program dari komputer kedalam memori program internal mikrokontroller, dapat dilakukan sebagai berikut :

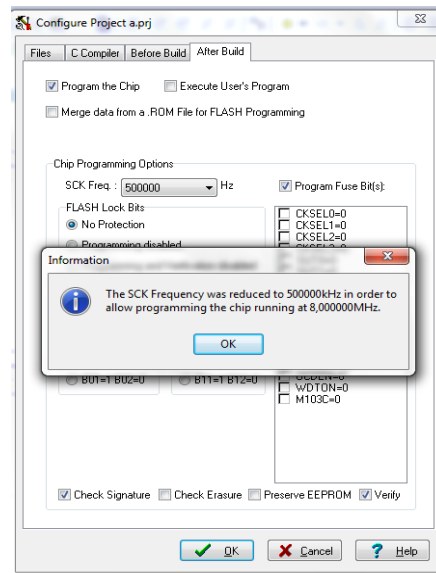
1. Pilih menu bar *Tool*, pilih *Configure*.



Gambar 3.3 Pemilihan menu Configure

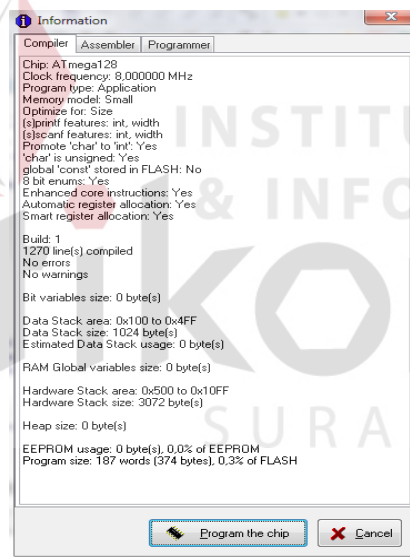
2. Pilih bagian *After Build*, centang *Program the Chip*, kemudian tekan OK.

Beikut tampilan dari program *Chip signature* pada pada *CodeVision AVR* yang akan digunakan untuk menuliskan program dan melakukan percobaan terhadap *Minimum system*.



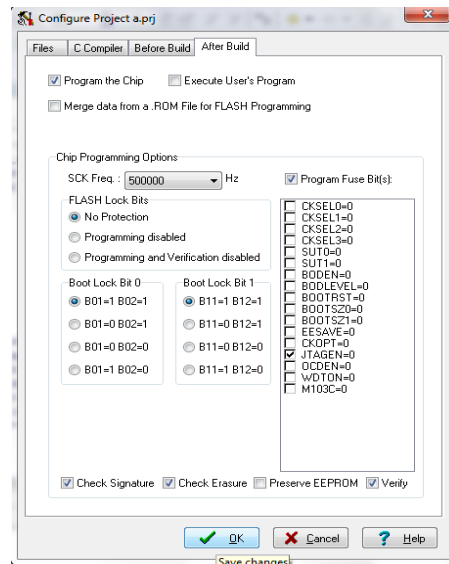
Gambar 3.4 Tampilan program *Chip signature*

3. Kemudian centang *Check Erasure*, lalu tekan OK.



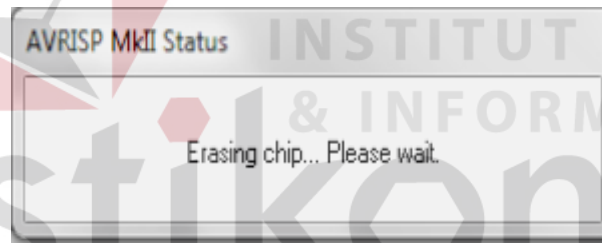
Gambar 3.5 Proses centang Check Erasure

4. Kemudian “Run” program AVR, tekan Ctrl-F9, setelah muncul tampilan dibawah tekan **Program the chip**.



Gambar 3.6 Proses Running Program

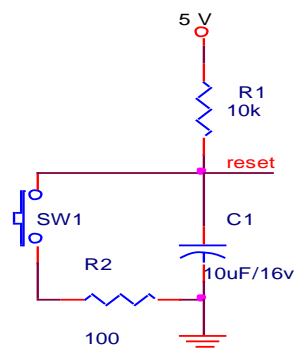
5. Terakhir *download* program, setelah proses *download* berhasil dapat dikatakan *Minimum system* dapat bekerja dengan baik.



Gambar 3.7 Proses Download Program AVR ke Mikro

A. Rangkaian Reset

Pin reset pada *microcontroller* adalah pin (kaki) 1. Reset dapat dilakukan secara manual atau otomatis saat power dihidupkan (Power reset ON).

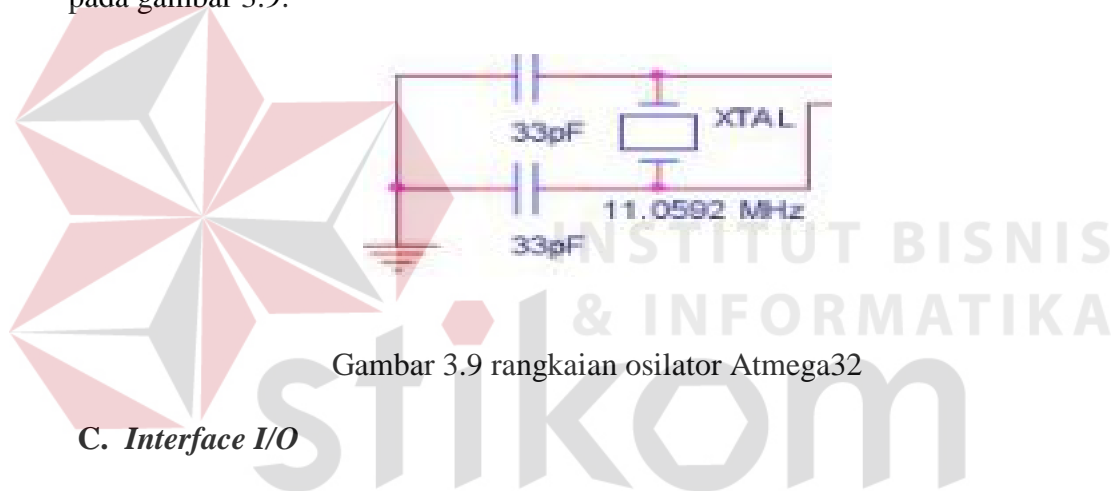


Gambar 3.8 Rangkaian Reset

Reset terjadi dengan adanya logika 1 selama minimal 2 *machine cycle* yang diterima pin reset dan akan bernilai *low*. Pada saat reset bernilai *low*, *microcontroller* akan melakukan reset program yang ada di dalam *microcontroller* dan mengakhiri semua aktivitas pada *microcontroller*.

B. Rangkaian osilator

Pin XTAL1 dan XTAL2 merupakan pin osilator bagi mikrokontroler Atmega32. Pin XtAL1 berfungsi sebagai input dan XTAL2 sebagai output osilator. Osilator ini bisa berasal dari kristal. Rangkaian osilator ini dapat dilihat pada gambar 3.9.



Gambar 3.9 rangkaian osilator Atmega32

C. Interface I/O

Rangkaian *I/O* dari *mikrokontroler* mempunyai kontrol direksi yang tiap bitnya dapat dikonfigurasi secara individual, maka dalam perancangan *I/O* yang digunakan ada yang berupa operasi *port* ada pula yang dikonfigurasi tiap bit yang ada pada masing-masing *port* yang terdapat pada *mikrokontroler*.

1. Port A

Port A digunakan untuk pengambilan data ADC (*Analog to Digital Converter*).

2. Port B

Port B digunakan untuk LCD (*Liquid Crystal Display*) agar dapat mengetahui suhu dan kelembaban yang ditangkap oleh LM35 dan *Soil Moisture*.

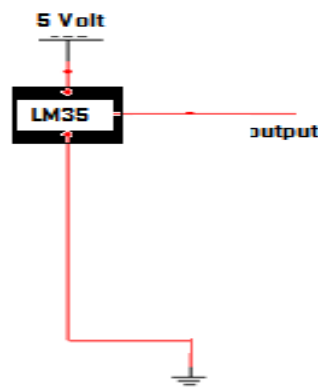
3. Port D

Port D digunakan untuk menggerakkan aktuator, seperti : motor, kipas, *headreyer* dan *fasher*.

3.1.2 Rangkaian LM35

LM35 digunakan untuk mengetahui suhu. Sensor LM35 akan membaca ACD terus menerus dan mengirimkan data ke *microkontroller* yang bertujuan agar suhu yang dihasilkan diolah agar sesuai dengan yang diinginkan. Pada LM35 terdapat 3 pin yang digunakan untuk jalur power 5 volt, ground dan data. Pin data dapat langsung dihubungkan dengan mikrokontroller port A.0 tanpa tambahan komponen apapun. LM35 akan menangkap data dari luar yang berupa suhu dan kemudian dikirimkan ke mikrokontroller untuk diolah, data tersebut dikirimkan oleh LM35 secara terus-menerus. Setelah itu mikrokontroller hanya mengukur suhu tersebut dan mengkonversikannya dalam bentuk suhu.

Pada gambar 3.10 ini adalah rangkaian sensor LM35 pada mikrokontroller.



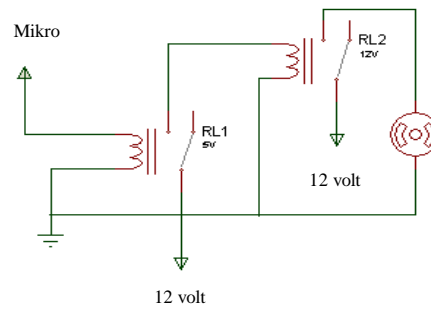
Gambar 3.10 Rangkaian LM35

3.1.3 Rangkaian Relay Motor DC

Motor DC memiliki dua kabel yang terhubung, kabel pertama untuk *power supply* dengan besaran 12 volt, kabel kedua digunakan untuk *ground*. Motor ini digunakan untuk memutar wadah tempat penyimpanan cacing yang tujuannya saat mengalami penyiraman air pada kondisi suhu tinggi dan kelembaban kering, agar grajen yang berada didalam wadah penyimpanan cacing tersebut mengalami perubahan suhu dan kelembaban yang merata. Pergerakan motor ini tidak tentu karena pergerakan motor tersebut diatur oleh perhitungan dengan menggunakan metode *fuzzy logic*.

Motor yang digunakan adalah motor gear box yang membutuhkan tegangan 12 volt dengan ampere 50 ampere. Daya yang digunakan untuk menggerakkan motor tersebut menggunakan aki mobil yang tegangan dan amperenya sesuai dengan yang dibutuhkan oleh motor tersebut.

Pergerakan motor tersebut diatur dengan menggunakan dua *relay* yang pertama outputan yang dihasilkan oleh *mikrokontroler* memicu *relay* 5 volt agar *relay* yang posisi awalnya berada pada posisi NO (*Normali Open*) akan terbuka menjadi NC (*Normali close*) saat *relay* pada posisi NC (*normali Close*) *relay* tersebut menghantarkan tegangan 12 volt yang berfungsi untuk memicu *relay* yang kedua. Sistem kerja *relay* kedua hampir sama dengan *relay* pertama bedanya *relay* kedua digunakan untuk langsung menggerakkan motor DC tersebut. Gambar rangkaian *relay* motor ditunjukkan pada gambar 3.11.



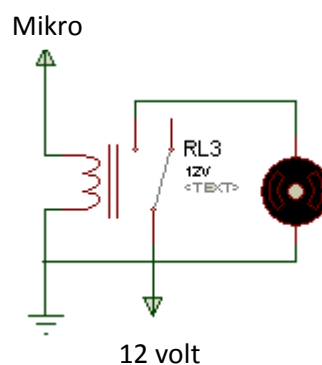
3.11 Rangkaian *relay* motor DC

3.1.4 Rangkaian Relay Kipas

Kipas digunakan untuk mendinginkan suhu dan mengeringkan grajen yang berada pada wadah cacing tersebut yang bertujuan agar suhu dan kelembaban sesuai dengan keadaan yang dibutuhkan oleh cacing tersebut.

Cacing jenis *lunbricus rubellus* membutuhkan suhu yang dengan dan kelembaban yang lebab, maka dari itu saat kelembaban mengalami kondisi yang basah kipas akan hidup dan mengeringkan agar kelembaban yang berada pada wadah cacing tersebut berubah menjadi lembab.

Selain itu, kipas ini akan menyala apabila suhu dalam wadah cacing tersebut melebihi dari suhu yang dibutuhkan oleh cacing tersebut agar cacing dapat bertahan hidup. Pergerakan kipas juga diatur oleh relay, tapi pada rangkaian kipas ini hanya menggunakan satu *relay*.



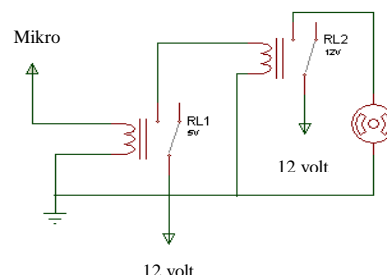
3.12 Rangkaian *relay* Kipas

3.1.5 Rangkaian Relay *Hair Dryer*

Hair dryer membutuhkan tegangan 220 volt AC, maka dari itu digunakan relay yang dapat dialirin tegangan sebesar yang dibutuhkan oleh *Hair dryer* tersebut. Untuk rangkaian ini menggunakan dua relay, sistem kerjanya hampir sama dengan rangkaian *relay* motor. *Hair dryer* ini digunakan untuk memanaskan suhu pada wadah cacing pada saat kondisi suhu didalam wadah cacing tersebut terlalu dingin.

Suhu yang dibutuhkan oleh cacing sekitar 15-25°C, apabila data yang dikirim oleh sensor LM35 tersebut kurang dari 15°C maka *Hair dryer* akan menyalah untuk menghangatkan suhu dalam wadah cacing tersebut sampai suhu benar-benar sesuai dengan suhu yang dibutuhkan oleh cacing tersebut.

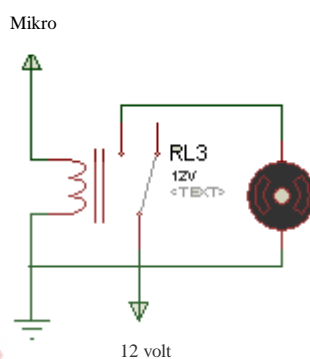
Nyala atau matinya *Hair dryer* diatur oleh dua *relay*, *relay* yang pertama outputan yang dihasilkan oleh *microkonroller* memicu relay 5 volt agar relay yang posisi awalnya berada pada posisi NO (*Normali Open*) akan terbuka menjadi NC (*Normali close*) saat *relay* pada posisi NC (*normali Close*) relay tersebut menghantarkan tegangan 12 volt yang berfungsi untuk memicu relay yang kedua. Sistem kerja *relay* kedua hampir sama dengan relay pertama bedanya *relay* kedua digunakan untuk langsung menggerakkan *headreyer* tersebut. Gambar rangkaian *relay Hair dryer* ditunjukkan pada gambar 3.13.



3.13 Rangkaian *relay Hair Dryer*

3.1.6 Rangkaian Relay *Fasher*

fasher digunakan untuk menyiram grajen, proses penyiraman ini terjadi apabila kelembaban yang dihasilkan oleh *sensor moisture* dibawah 30% maka dari itu untuk meningkatkan kelembaban untuk menghasilkan kelembaban yang lembab maka *fasher* akan menyalah selama sensor kelembaban tidak menghasilkan data lebih dari 50%. Pada gambar 3.14 adalah gambar rangkaian dari *relay fasher*.



3.14 Rangkaian *relay Fasher*

3.2 Peletakan Komponen Elektronik

Rangkaian elektronika mulai dari rangkaian *microkontroller*, rangkaian *relay motor*, rangkaian *relay fasher*, rangkaian *relay motor*, rangkaian *relay Hair dreyer* dan modul *soil moisture*, terletak pada bagian samping atau bagian terluar dari wadah tempat penyimpanan cacing. Sedangkan sensor *soil moisture* dan LM35 terdapat didalam wadah cacing. Berikut ini adalah gambar keseluruhan :



3.15 Tampilan Alat Keseluruhan

Berikut arsitektur detail dari gambar 3.15 :

Ukuran dimensi

Ukuran Alat : 80 cm x 60 cm x 60

Struktur material :

1. Balok diameter 60 cm x 40 cm
2. Besi ukuran 80 cm x 60 cm x 60 cm
3. Mur dan baut
4. Aluminium
5. Pelekat
6. Kipas dengan diameter 140 mm
7. Selang
8. *Fasher, Hair Dreyer* dan Motor

Sensor kelembaban dan sensor LM35



3.16 Peletakan Sensor kelembaban dan LM35

Sensor kelembaban dan sensor LM35 diletakkan pada bagian dalam wadah cacing menempel dengan gajen kayu yang digunakan sebagai medianya untuk mengetahui kelembaban dan suhu dalam wadah tersebut.



Gambar 3.17 Peletakan Motor

Motor diletakkan separuh bagian dalam dan separuh bagian luar. Pada bagian luar tersebut untuk menggerakkan wadah saat proses pengadukan dihubungkan dengan pully.



3.18 Peletakan Kipas

Kipas diletakkan pada bagian dalam dimana terdapat delapan kipas, empat kipas untuk mengeluarkan udara yang berada dalam ruangan dan yang empat untuk mengambil udara dari luar ruangan. Kipas tersebut berada pada bagian depan dan belakang.



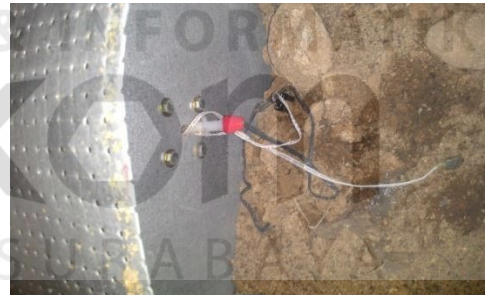
3.19 Peletakan Aki dan Botol *fasher*

Aki dan botor *fasher* terletak pada bagian belakang dari alat ini, karena perangkat elektroniknya hampir semuanya terletak pada bagian belakang.



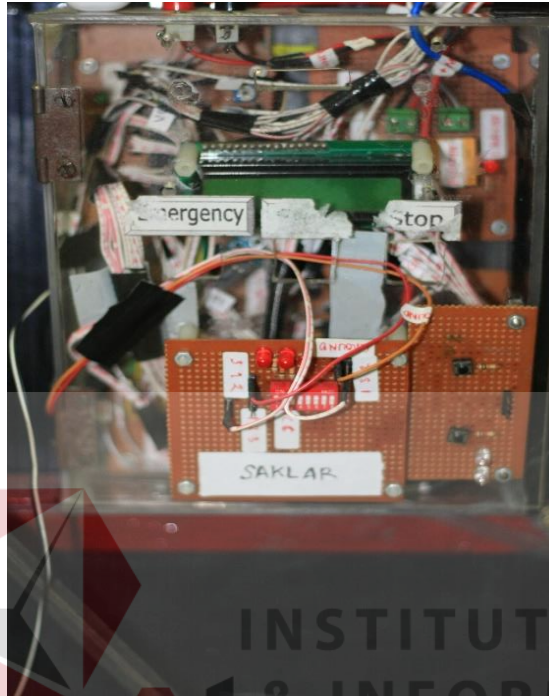
3.20 Peletakan *Hair dreyer*

Hair Dreyer diletakkan pada bagian samping pada wadah cacing. *Hair Dreyer* ini langsung menghadap wadah cacing agar saat *Hair Dreyer* menyala panasnya akan menyeluruh.



Gambar 3. 21 Peletakan *fasher*

Fasher ini diletakkan 2 pada bagian depan wadah, 1 pada bagian belakang wadah dan 1 pada bagian dalam wadah yang mana tujuannya agar saat penyiram kelembaban akan merata.



3.22 Peletakkan komponen elektronika

Peletakkan komponen elektronika terletak pada bagian samping.

Komponen elektronika ini mencakup rangkaian *relay* motor, *fasher*, *Hair Dreyer* dan kipas serta display LCD.



3.23 Peletakan Wadah Cacing

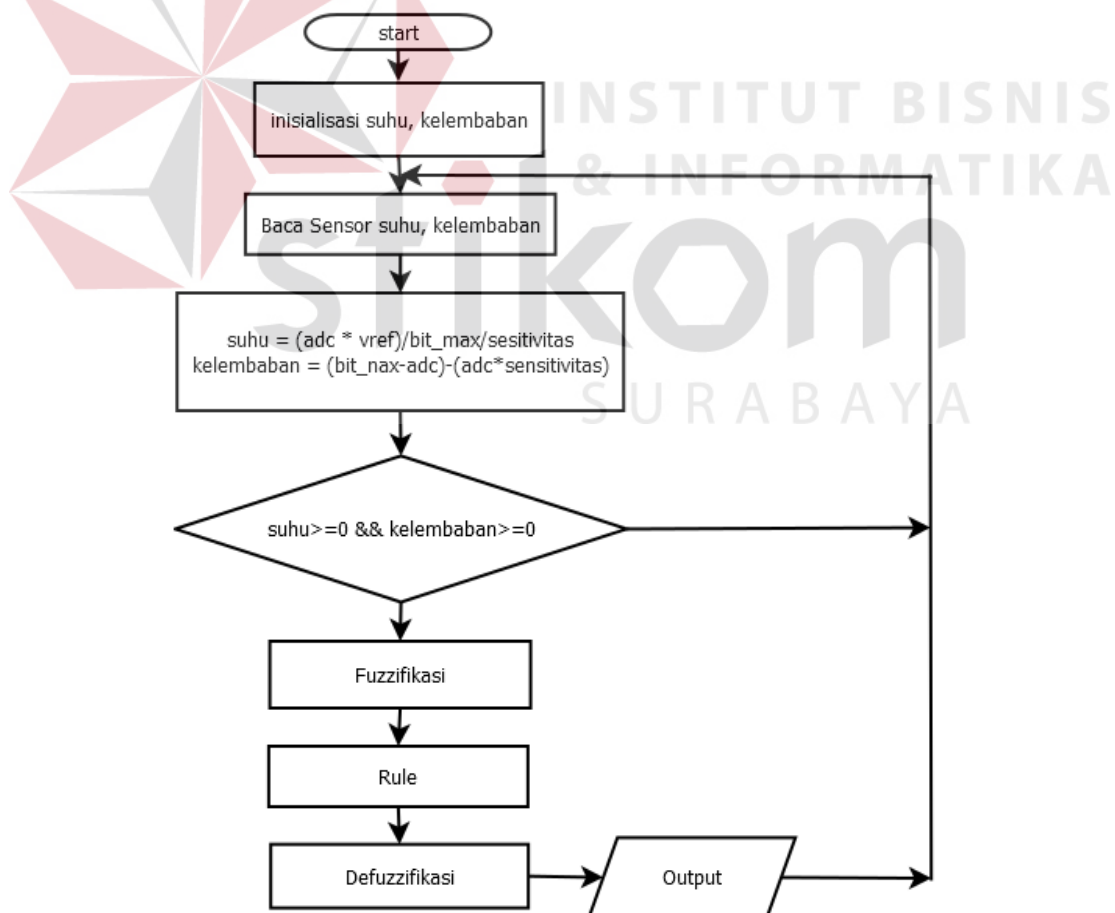
Peletakan wadah cacing berada pada bagian dalam kotak dan berada pada posisi tengah pada kotak tersebut.

3.3 Perancangan Perangkat Lunak

Perangkat lunak bertujuan untuk mengirim dan menerima input output dari sensor kelembaban dan suhu. Minimum sistem memperoleh data dari LM35 dan *soil moisture* yang berupa ADC yang kemudian diolah menjadi kelembaban dan celcius yang bertujuan agar dapat menggerakkan aktuator.

Perancangan perangkat lunak terbagi dalam beberapa program antara lain :

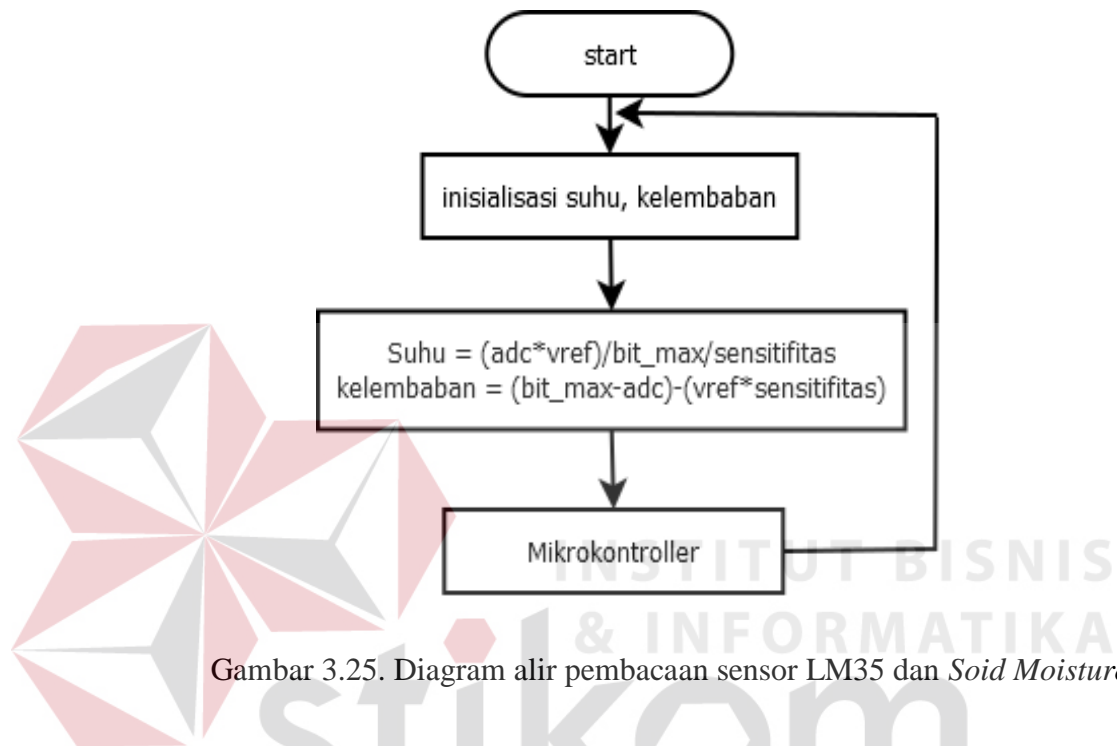
Program LM35, Program *soil moisture*, dan algoritma *fuzzy*. Diagram alir perangkat lunak secara umum dapat dilihat pada gambar 3.24.



Gambar 3.24 Diagram alir Program secara umum pada mikrokontroller

3.3.1 Program Membaca Suhu dan Kelembaban

Diagram alir untuk mengetahui suhu dan kelembaban berdasarkan pembacaan sensor LM35 dan Sensor *Soil Moisture* terdapat pada gambar 3.25.



Gambar 3.25. Diagram alir pembacaan sensor LM35 dan *Soild Moisture*

Pada gambar 3.25, pembacaan LM35 dan *Soil Moisture* dimulai dengan inisialisasi sensor LM35 dan sensor *Soil Moisture*. Jika data LM35 dan data *soil moisture* berhasil dibaca maka data keduanya akan diproses oleh *microkontroller* dan dimasukkan kedalam algoritma *fuzzy* setelah itu ditampilkan ke LCD. Berikut ini potongan program pembacaan LM35 dan *soil moisture* serta menampilkan ke *display* LCD :

```

hasil = read_adc(0x00);
hasil = ((float)hasil*2500);
hasil = hasil/255;
hasil = hasil/10;

```

```

a = read_adc (0x01);
a = ((float) 255-a);
a = a/2,5;

lcd_gotoxy(0,0);
lcd_putsf ("Suhu : ");
ftoa(hasil,1,temp);
lcd_gotoxy(7,0);
lcd_puts(temp);
lcd_gotoxy(0,1);
lcd_putsf ("lembab : ");
itoa(a,temp1);
lcd_gotoxy(8,1);
lcd_puts(temp1);

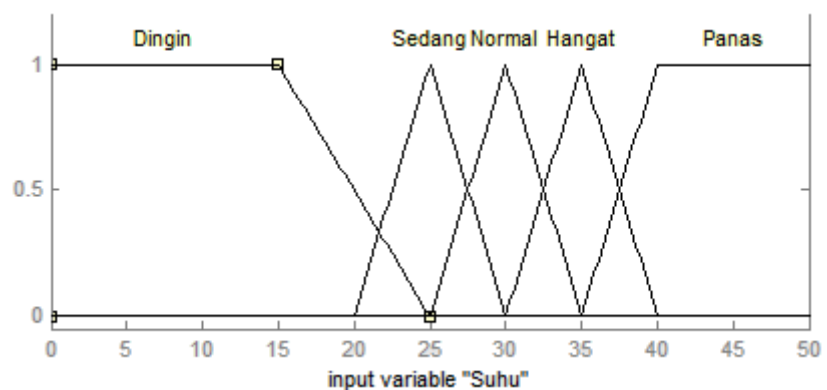
```

3.3.2 Fuzzy

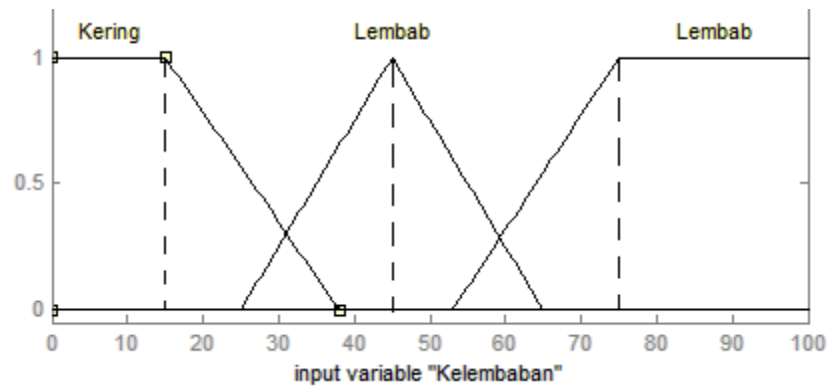
Untuk menentukan pergerakan aktuator terdapat tiga proses yaitu :

1. Fuzzifikasi

Pada proses *fuzzifikasi* ini terjadi pengambilan keputusan dengan cara mengubah masukkan *crisp* (bentuk tegas) menjadi *fuzzy* (variabel linguistik). Pengubahan *crisp* menjadi *fuzzy* dapat dilihat pada gambar 3.26.



Gambar 3.26. Pemetaan Keanggotaan Suhu



Gambar 3.27 Pemetaan Keanggotaan Kelembaban

Realisasi proses memperoleh nilai derajat keanggotaan dari masing-masing fungsi keanggotaana adalah sebagai berikut :

```
unsigned char fuzzifikasi_suhu_dingin()
{
    suhu_dingin=0;
    if (hasil<=15 && hasil>0)
    { suhu_dingin = 1; }
    else if (hasil>=15 && hasil<20)
    { suhu_dingin=(hasil-15)/5; }
    else if (hasil>=25)
    { suhu_dingin = 0; }
    return suhu_dingin;
};
```

```
unsigned char fuzzifikasi_suhu_sejuk()
{
    suhu_sejuk=0;
    if (hasil<=20 || hasil>=30)
    { suhu_sejuk = 0; }
    else if (hasil>20 && hasil<25)
    { suhu_sejuk=(hasil-20)/5; }
    else if (hasil>25 && hasil<30)
    { suhu_sejuk=(hasil-25)/5; }
    else if (hasil==25)
    { suhu_sejuk=1; };
};
```

```

return suhu_sejuk;
};

```

2. Rule Set

Rule set adalah proses evaluasi derajat keanggotaan tiap-tiap fungsi himpunan *fuzzy* masukan kedalam basis aturan yang telah ditetapkan. Tujuan evaluasi ini adalah menentukan derajat keanggotaan dari nilai fuzzy.

Sebelum melakukan evaluasi aturan terlebih dahulu ditetapkan oleh basis aturan. Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin secara lengkap jumlah kombinasi yang mungkin dari dua himpunan *fuzzy* masukan terdiri lima himpunan *fuzzy* dan tiga himpunan *fuzzy* fungsi keanggotaan sehingga jumlah aturan adalah lima belas aturan. Lima belas aturan kendali *fuzzy* dapat dilihat pada tabel 3.1.

Tabel 3.1. Penentuan *Rule Set*

kelembaban	suhu					
		D	S	N	H	P
	K	MFH	MF	MF	MF	MFK
	L	DM	DM	K	K	K
	B	KH	KH	KH	K	K

Dari tabel diatas dapat diartikan sebagai berikut :

IF kering AND dingin THEN motor, *fasher*, *hairdryer*.

IF kering AND sedang THEN motor, *fasher*.

IF kering AND normal THEN motor, *fasher*.

IF kering AND hangat THEN motor, *fasher*.

IF kering AND panas THEN motor, *fasher*, kipas.

IF lembab AND dingin THEN diam.

IF lembab AND sedang THEN diam.

IF lembab AND normal THEN kipas.

IF lembab AND hangat THEN kipas.

IF lembab AND panas THEN kipas.

IF basah AND dingin THEN kipas, *hairdryer*.

IF basah AND sedang THEN kipas, *hairdryer*.

IF basah AND normal THEN kipas, *hairdryer*.

IF basah AND hangat THEN kipas.

IF basah AND panas THEN kipas.

Keterangan :

D : Dingin

S : Sedang

N : Normal

H : Hangat

P : Panas

K : Kering

L : Lembab

B : Basah

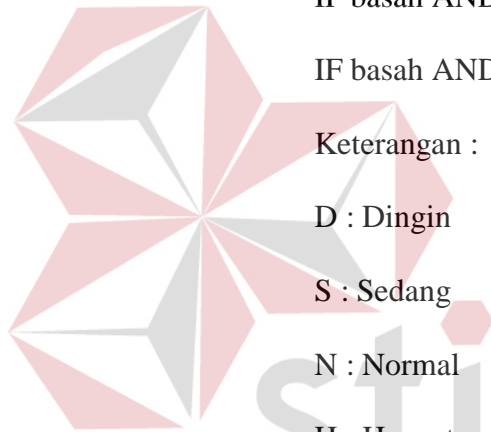
DM : Diam

KH : Kipas dan *Hair dryer*

K : Kipas

MFK : Motor, *fasher*, Kipas

MF : Motor dan *Fasher*



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA

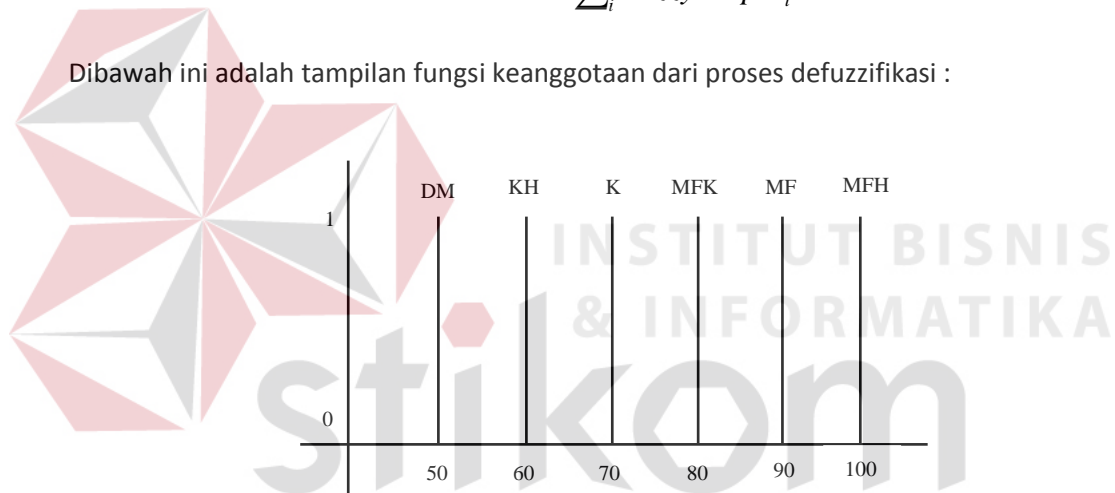
MFH : Motor, Fasher dan Hair dryer

3. Defuzzifikasi

Defuzzifikasi adalah mengubah himpunan *fuzzy* keluaran menjadi keluaran tegas (*crisp*). Pengubahan ini diperlkan karena konstata kendali *fuzzy* hanya menal nilai tegas sebagai variabel kontrol. Realisasi proses pengambilan keputusan metode sugeno menjadi bentuk *crisp output* dapat menggunakan rumus sebagai berikut :

$$Crisp_out = \frac{\sum_i FuzzyOutput_i \times (PosisiSingletonXaxis_i)}{\sum_i FuzzyOutput_i}$$

Dibawah ini adalah tampilan fungsi keanggotaan dari proses defuzzifikasi :



Gambar 3.28. fungsi keanggotaan output

Nilai z yang dihasilkandefuzzifikasi berfungsi untuk menentukan suatu nilai *crisp* output. suatu nilai fuzzy output yang berasal dari evaluasi rule diambil kemudian dimasukkan ke dalam suatu fungsi keanggotaan keluaran (*membership function output*). Besar nilai fuzzy output dinyatakan sebagai *degree of membership function output*. Nilai-nilai tersebut dimasukkan ke dalam suatu rumus untuk mendapatkan hasil akhir yang disebut *crisp output*.