

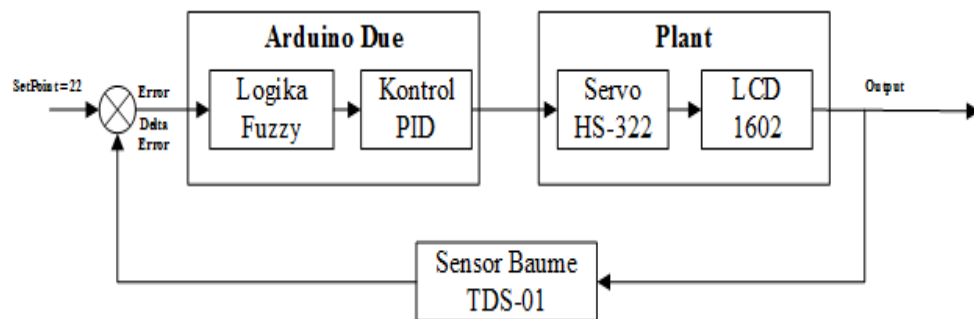
BAB III

PERENCANAAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat merupakan proses perencanaan desain yang dilakukan supaya tujuan dari penelitian dapat terpenuhi. Perancangan dan pembuatan alat pada penelitian ini berisi tentang rencana pembuatan dan pemodelan dari alat pengendali tingkat salinitas secara detail mulai dari perancangan mekanik hingga perancangan sistem kontrol.

Perancangan mekanik merupakan rencana bagaimana merakit perangkat mekanik yang berupa masukan, kontroler, dan keluaran. Sementara perancangan sistem kontrol merupakan rencana sistem pengendalian alat dengan merancang parameter – parameter pada sistem kontrol logika *fuzzy* dan PID supaya alat yang telah dirancang dalam perancangan mekanik dapat berfungsi dengan baik.

Penggambaran awal dari cara kerja seluruh sistem dapat digambarkan dalam diagram blok. Diagram blok secara umum perancangan alat pada penelitian ini dapat dilihat dalam gambar 3.1



Gambar 3.1 Diagram Blok Perancangan Alat

Dalam Gambar 3.1, dijelaskan tentang cara kerja dari alat pengendali tingkat salinitas dimana kerja sistem dimulai dari pemberian nilai setpoint yang berupa nilai salinitas optimal berdasarkan Tabel 2.1 sebesar 22, setelah itu akan dihitung nilai *error* dan *delta error* dengan menggunakan persamaan 2.2. Setelah didapat nilai *error* dan *delta error*, maka pengendalian sistem akan menggunakan logika *fuzzy* dan PID yang terdapat dalam arduino due, setelah itu akan dihasilkan nilai yang berupa pengontrolan sudut motor servo HS-322. Perubahan sudut servo akan

menyebabkan katup berubah posisi yang menyebabkan perubahan nilai pada salinitas dalam air. Nilai salinitas ini akan ditampilkan pada LCD 1602, kemudian nilai salinitas ini akan diumpun balikkan pada kontroler untuk dihitung nilai *error* dan delta *error* lagi sehingga nilai salinitas dapat bertahan pada nilai *setpoint* yang telah diberikan.

3.1 Pemodelan Sistem Alat

3.1.1 Pengumpulan Data Perancangan

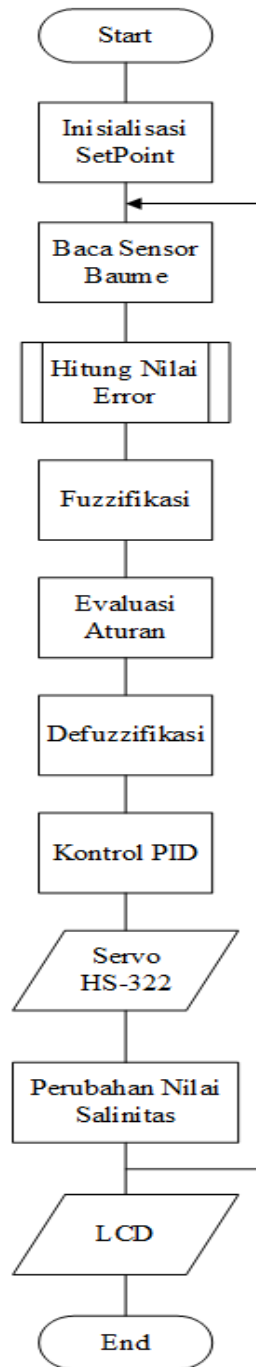
Dalam perancangan dan pembuatan alat pengendalian tingkat salinitas dalam proses penggaraman kulit sapi ini, tentunya membutuhkan data – data sebagai acuan dalam melakukan perancangan dan pembuatan mekanik. Data – data yang akan dijadikan acuan dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Data Perancangan Alat

Kapasitas Drum	250 L
Volume Air	200 L
Jenis Garam	Garam Industri
Volume Garam	20 Kg
Tingkat Salinitas	22 Be
Kapasitas Drum pada Alat	9 L
Volume Air pada Alat	3 L
Jenis Garam pada Alat	Garam Dapur
Kapasitas Wadah Garam pada Alat	500 G

3.1.2 Pemodelan Sistem Kerja Alat

Setelah merancang diagram blok untuk alat, maka dapat dilakukan pemodelan sistem alat untuk mengetahui aliran kerja dari alat tersebut. Pada tahap ini , akan dilakukan pemodelan bagaimana alur kerja secara detail dari sistem yang dapat dilihat dalam Gambar 3.2.



Gambar 3.2 Diagram Alir Pemodelan Sistem

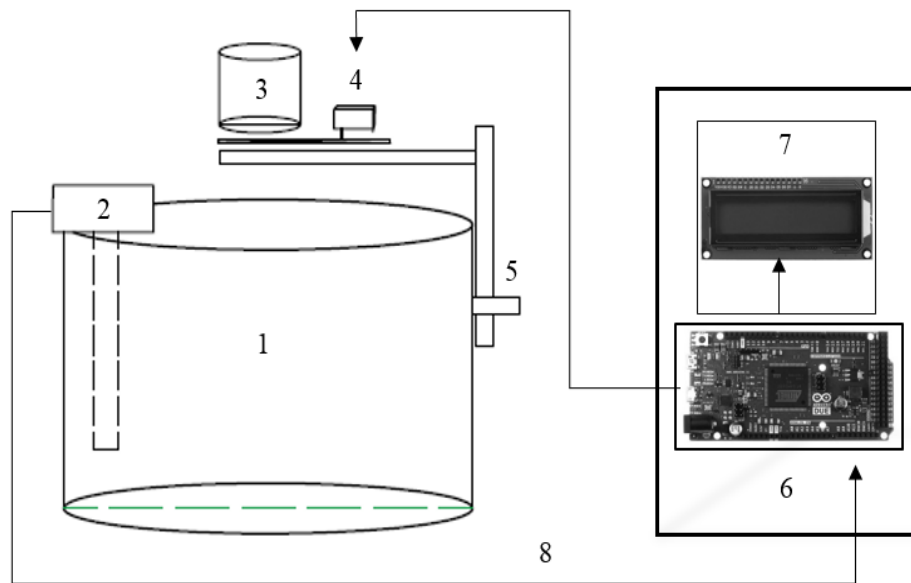
Gambar 3.2 menunjukkan cara kerja dari sistem, cara kerja berdasarkan diagram alir tersebut adalah ketika sistem dimulai, dilakukan inisialisasi sitem yang berupa set point sebesar 22, setelah itu sensor baume akan membaca tingkat salinitas, nilai dari sensor baume akan dihitung nilai *error* dan *delta error* berdasarkan persamaan 2.1. Hasil perhitungan nilai *error* dan *delta error* akan menjadi nilai masukan dari proses fuzzifikasi.

Proses fuzzifikasi akan menghasilkan nilai masukan *fuzzy* yang digunakan dalam proses inferensi *fuzzy* yang kemudian dievaluasi terhadap aturan *fuzzy* yang telah dibuat. Proses inferensi *fuzzy* akan menghasilkan nilai keluaran *fuzzy* yang akan diproses dalam proses defuzzifikasi. Setelah itu, proses defuzzifikasi akan menghasilkan nilai keluaran tegas berupa nilai desimal yang digunakan sebagai nilai sudut untuk mengendalikan nilai sudut dari motor servo. Setelah itu hasil nilai sudut tersebut akan melalui proses pengendalian menggunakan kontrol PID yang bertujuan agar pergerakan motor servo sesuai dengan nilai defuzzifikasi yang telah dihasilkan.

Perubahan sudut pada motor servo akan menyebabkan katup penutup wadah garam terbuka dan menyebabkan garam mengalir ke dalam air dan menyebabkan perubahan nilai salinitas. Perubahan nilai salinitas ini akan dibaca oleh sensor baume, dan selanjutnya akan diumpan balikkan pada kontroler untuk dihitung kembali nilai *error* dan delta *error* yang akan dikontrol kembali oleh logika *fuzzy* dan kontrol PID sehingga tingkat salinitas dalam air mampu bertahan sesuai nilai setpoint yang telah diberikan. Hasil pengendalian nilai tingkat salinitas dalam alat ini akan ditampilkan dalam LCD untuk memudahkan operator dalam mengetahui nilai tingkat salinitas yang ada dalam air pada proses penggaraman.

3.2 Perancangan Mekanik

Setelah perancangan diagram blok, pemodelan sistem, dan pengumpulan data telah selesai dilakukan, maka dapat dibuat rancangan mekanik yang nantinya akan dikontrol oleh sistem kontrol yang dirancang. Perancangan mekanik pada penelitian ini dapat dilihat dalam gambar 3.3.



Gambar 3.3 Perancangan Mekanik

Dari gambar 3.3, dapat dilihat bahwa dalam perancangan alat ini terdiri dari beberapa alat dan bahan. Beberapa alat dan bahan yang digunakan dalam alat ini adalah :

1. Galon Air 9 Liter
2. Sensor TDS-01
3. Botol 650 ml
4. Motor Servo HS-322 HD
5. Papan Kayu
6. Arduino Due
7. LCD 1602
8. Kabel *Jumper*

3.3 Perancangan Sistem Kontrol Logika *Fuzzy*

Pada tahap ini, akan dirancang kendali menggunakan logika *fuzzy* yang digunakan untuk menghasilkan parameter berupa nilai sudut dari motor servo yang nantinya akan mengendalikan seberapa besar debit garam yang dialirkan dari wadah ke dalam air dalam drum melalui katup yang dikendalikan oleh motor servo.

Pada perancangan kendali *fuzzy* ini, digunakan 2 masukan yaitu nilai *error* dari sensor baume dan nilai *delta error* dari sensor baume dengan keluaran yang berupa nilai sudut motor servo HS-322. Proses perancangan kendali logika *fuzzy* dalam penelitian ini sebagai berikut :

3.3.1 Fuzzifikasi

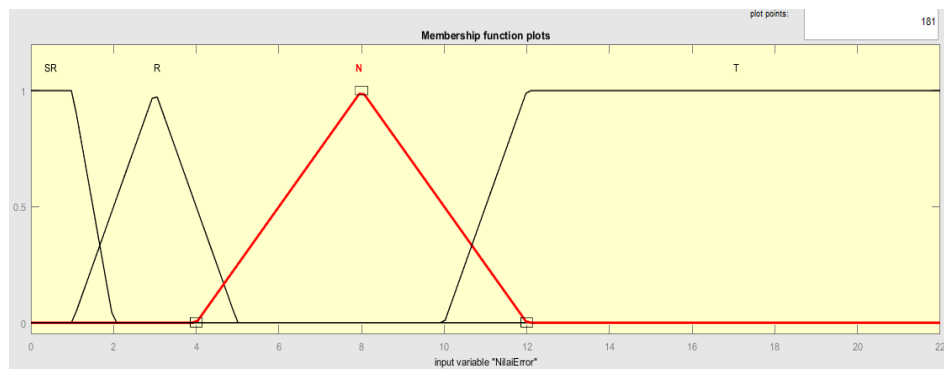
a. Nilai *Error* Sensor Baume

Pada penelitian ini, sensor baume TDS-01 digunakan untuk mendeteksi tingkat kepekatan garam dalam air. Pada perancangan logika *fuzzy* ini, keluaran dari sensor baume akan dibaca besar nilai *error* dengan menggunakan persamaan 2.12. Nilai *error* tersebut akan dikategorikan dalam empat buah atribut linguistik, yaitu sangat rendah, rendah, normal, dan tinggi. Pengelompokan nilai *error* dari sensor baume dapat dilihat dalam Tabel 3.2.

Tabel 3.2 Himpunan *Fuzzy* Nilai *Error* Sensor Baume

Atribut Linguistik	Nilai <i>Error</i> Sensor Baume
Sangat Rendah	0 s/d 2
Rendah	1 s/d 5
Normal	4 s/d 12
Tinggi	10 s/d 22

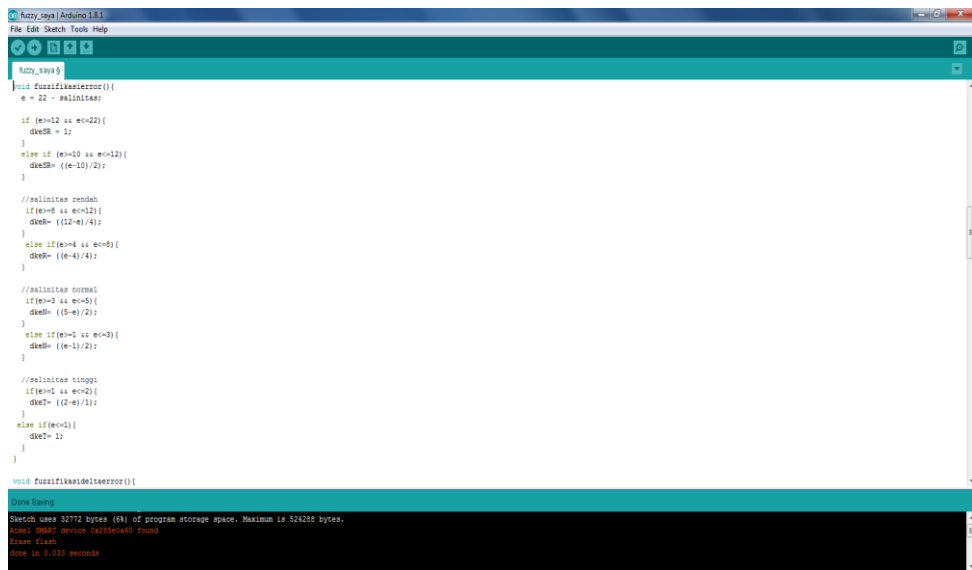
Pengelompokkan nilai *error* dari sensor baume dalam alat pengendalian tingkat salinitas ini diimplementasikan dengan menggunakan Matlab seperti yang terdapat dalam Gambar 3.4.



Gambar 3.4 Variabel Masukan Nilai *Error* Sensor Baume

Proses pembacaan nilai *error* dari sensor baume akan diimplementasikan ke dalam mikrokontroler Arduino Due dengan

menggunakan *software Arduino IDE 1.8.1* seperti yang ditunjukkan dalam Gambar 3.5.



Gambar 3.5 *Script* Program untuk Proses Fuzzifikasi Nilai Error

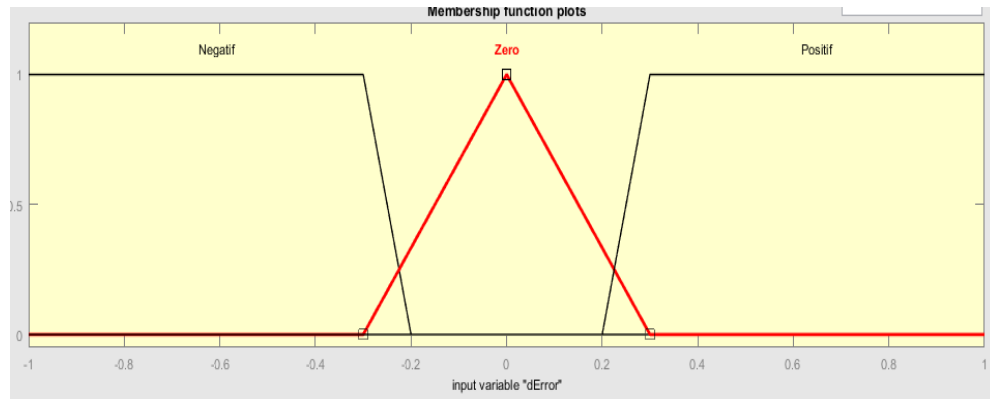
b. Nilai *Delta Error* Sensor Baume

Pada penelitian ini, selain menggunakan nilai *error* dari sensor baume, juga digunakan nilai *delta error* dari sensor baume sebagai variabel masukan logika *fuzzy*. Pada perancangan logika *fuzzy* ini, nilai *delta error* didapat dari persamaan 2.13. Nilai *delta error* tersebut dikategorikan dalam tiga buah atribut linguistik, yaitu negatif, zero, dan positif. Pengelompokan nilai *delta error* dari sensor baume dapat dilihat pada Tabel 3.3.

Tabel 3.3 Himpunan *Fuzzy* Nilai *Delta Error*

Atribut Linguistik	Nilai <i>Delta Error</i>
Negatif	-1 s/d -0,2
Zero	-0,3 s/d 0,3
Positif	0,2 s/d 1

Pengelompokkan nilai *delta error* dari sensor baume dalam alat pengendalian tingkat salinitas ini diimplementasikan kedalam mikrokontroler Arduino Due dengan menggunakan Matlab seperti yang terdapat dalam Gambar 3.6.



Gambar 3.6 Variabel Masukan *Delta Error* Sensor Baume

Proses pembacaan nilai *error* dari sensor baume akan diimplementasikan ke dalam mikrokontroler Arduino Due dengan menggunakan *software Arduino IDE 1.8.1* seperti yang ditunjukkan dalam Gambar 3.7.

```

fuzzy_jaya | Arduino 1.8.1
File Edit Sketch Tools Help
-----
fuzzy_jaya$
void FuzzifikasiDeltaError()
{
  de = e - ea;
  if (de == -1 || de == 1)
  {
    de = 0;
  }
  ea = e;
  //delta error negatif
  if (de == -1 || de < -0.3)
  {
    dide1 = 1;
  }
  else if (de > -0.3 || de < -0.2)
  {
    dide1 = ((-0.2 - de) / 0.1);
  }
  //delta error zero
  else if (de == -0.3 || de == 0)
  {
    dide2 = ((de + 0.3) / 0.3);
  }
  else if (de == 0 || de < 0.3)
  {
    dide2 = ((0.3 - de) / 0.3);
  }
  //delta error positif
  else if (de == 0.2 || de < 0.3)
  {
    dide3 = ((de - 0.2) / 0.1);
  }
  else if (de == 0.3 || de < 1)
  {
    dide3 = 1;
  }
}
}

-----
Sketch uses 32772 bytes (81% of program storage space, Maximum is 51200 bytes).
Sketch uses 10240 bytes (100% of dynamic memory, Maximum is 10240 bytes) free.
-----
Done
-----
Done in 0.033 seconds
-----
Arduino Due (Programming Port) on COM6

```

Gambar 3.7 Script Program Proses Fuzzifikasi Nilai *Delta Error*.

Setelah membuat pengelompokan nilai masukan dari logika *fuzzy*, langkah berikutnya adalah menghitung nilai derajat keanggotaan dari tiap – tiap nilai masukan hasil perhitungan nilai *error* dan *delta error*,

Pengelompokan nilai derajat keanggotaan dari nilai *error* dan *delta error* pada penelitian ini menggunakan representasi nilai segitiga. Maka dari itu, perhitungan nilai derajat keanggotaan dari masukan *error* dan *delta error* menggunakan metode representasi segitiga dari fuzzifikasi yang terdapat pada Persamaan 2.3.

Setelah didapatkan nilai derajat keanggotaan untuk tiap – tiap masukan *error* dan delta *error*, maka nilai derajat keanggotaan tersebut akan dievaluasi terhadap aturan *fuzzy* yang telah dibuat.

3.3.2 Evaluasi Aturan

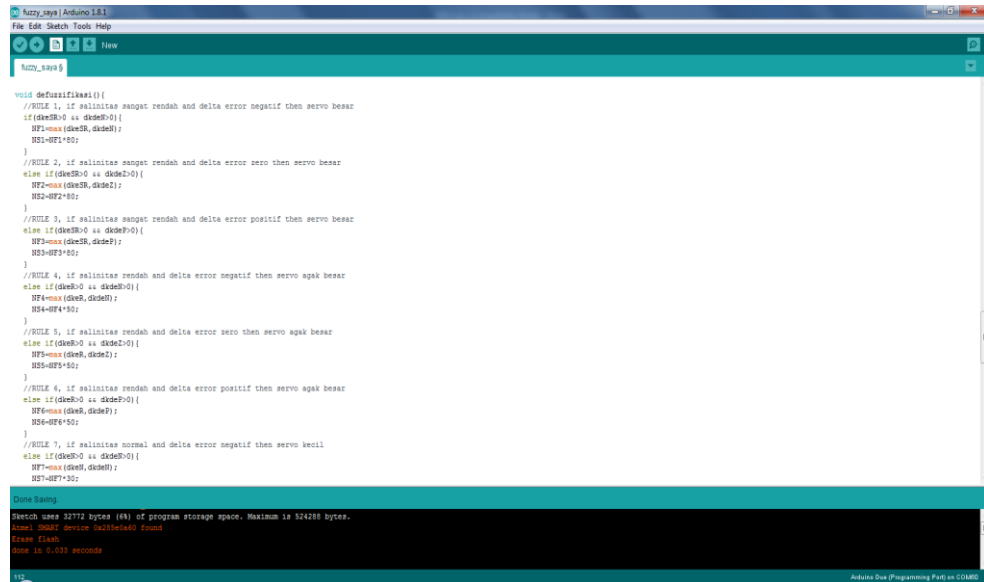
Evaluasi aturan adalah proses mengevaluasi derajat keanggotaan tiap-tiap nilai derajat keanggotaan dari himpunan *fuzzy* masukan yang telah dihitung dalam proses fuzzifikasi ke dalam basis aturan yang telah ditetapkan. Basis aturan yang dibuat berdasarkan karakteristik plant untuk memaksimalkan performa dari proses pengendalian agar hasil keluaran yang diinginkan dapat dicapai dengan baik. Basis aturan dari perancangan logika *fuzzy* dalam sistem kendali dari penelitian ini dibuat berdasarkan saran dari narasumber yang merupakan operator dari proses penggaraman serta ditambahkan dengan pemikiran logika berdasarkan karakteristik dari garam bila berada didalam air tanpa diencerkan seperti yang terdapat dalam Sub Bab 2.10.

Basis aturan yang akan digunakan dalam penelitian ini terdiri dari 12 aturan yang saling berkaitan antara nilai *error*, delta *error*, dan nilai keluaran sudut servo seperti ditunjukkan dalam Tabel 3.4.

Tabel 3.4 Perancangan Basis Aturan Logika *Fuzzy*

Nilai <i>Error</i>	Nilai Delta <i>Error</i>	Keluaran
Sangat Rendah	Negatif	Sangat Kecil
Sangat Rendah	Zero	Sangat Kecil
Sangat Rendah	Positif	Sangat Kecil
Rendah	Negatif	Kecil
Rendah	Zero	Kecil
Rendah	Positif	Kecil
Normal	Negatif	Sedang
Normal	Zero	Sedang
Normal	Positif	Sedang
Tinggi	Negatif	Besar
Tinggi	Zero	Besar
Tinggi	Positif	Besar

Perancangan basis aturan logika *fuzzy* dalam penelitian ini akan diimplementasikan ke dalam mikrokontroler Arduino Due dengan menggunakan *software Arduino IDE 1.8.1* seperti dalam Gambar 3.8.



```

void defuzzifikasi() {
  //RULE 1, if salinitas sangat rendah and delta error negatif then servo besar
  if (diseR<0 && diseD<0) {
    MF1=max(diseR,diseD);
    MS1=MF1*80;
  }
  //RULE 2, if salinitas sangat rendah and delta error zero then servo besar
  else if (diseR<0 && diseD==0) {
    MF2=max(diseR,diseD);
    MS2=MF2*80;
  }
  //RULE 3, if salinitas sangat rendah and delta error positif then servo besar
  else if (diseR<0 && diseD>0) {
    MF3=max(diseR,diseD);
    MS3=MF3*80;
  }
  //RULE 4, if salinitas rendah and delta error negatif then servo agak besar
  else if (diseR>0 && diseD<0) {
    MF4=max(diseR,diseD);
    MS4=MF4*50;
  }
  //RULE 5, if salinitas rendah and delta error zero then servo agak besar
  else if (diseR>0 && diseD==0) {
    MF5=max(diseR,diseD);
    MS5=MF5*50;
  }
  //RULE 6, if salinitas rendah and delta error positif then servo agak besar
  else if (diseR>0 && diseD>0) {
    MF6=max(diseR,diseD);
    MS6=MF6*50;
  }
  //RULE 7, if salinitas normal and delta error negatif then servo kecil
  else if (diseR>0 && diseD<0) {
    MF7=max(diseR,diseD);
    MS7=MF7*30;
  }
}

```

Done Burning
 Sketch uses 32772 bytes (68) of program storage space. Maximum is 524288 bytes.
 Avrami 30M32 device (23131640 found)
 Erase flash
 done in 0.031 seconds

Gambar 3.8 *Script* Program Perancangan Basis Aturan Logika *Fuzzy*

Nilai derajat keanggotaan dari masukan *error* dan *delta error* yang telah dihitung pada proses fuzzifikasi akan dievaluasi terhadap basis aturan yang sudah dibuat. Proses evaluasi aturan *fuzzy* ini ditentukan dengan metode *Maximum* yang menggunakan Persamaan 2.6.

Setelah melalui proses evaluasi terhadap basis aturan, maka akan dihasilkan nilai hasil operator *fuzzy* yang akan dijadikan nilai komposisi pada proses defuzzifikasi.

3.3.3 Defuzzifikasi

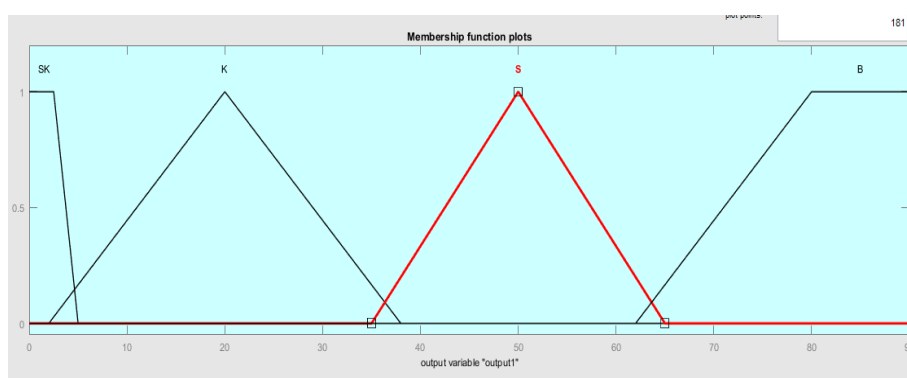
Defuzzifikasi adalah proses perubahan nilai keluaran *fuzzy* menjadi keluaran tegas. Dalam penelitian ini, keluaran *fuzzy* merupakan nilai sudut dari motor servo sebesar 0 – 90°.

Pada proses ini, fungsi keanggotaan dari keluaran *fuzzy* dirancang berbentuk segitiga yang dikelompokkan kedalam 4 nilai atribut linguistik yang terdiri dari sangat kecil, kecil, sedang dan besar. Nilai dari masing – masing atribut linguistik tersebut terdapat dalam Tabel 3.5.

Tabel 3.5 Himpunan *Fuzzy* Nilai Keluaran Servo

Atribut Linguistik	Nilai Sudut Motor Servo
Sangat Kecil	0 s/d 5
Kecil	2 s/d 38
Sedang	35 s/d 65
Besar	62 s/d 90

Masing – masing nilai atribut linguistik pada proses defuzzifikasi ini kemudian dirancang dalam *Matlab Simulink* dalam satu fungsi *fuzzy* yang sama dengan proses fuzzifikasi dan evaluasi aturan. Perancangan proses defuzzifikasi ditunjukkan dalam Gambar 3.9.



Gambar 3.9 Desain *Matlab Simulink* Keluaran Servo HS-322

Proses defuzzifikasi pada penelitian ini dilakukan dengan metode weighted average dengan menggunakan Persamaan 2.10. Proses ini akan diimplementasikan dengan menggunakan *software Arduino IDE 1.8.1* seperti yang ditunjukkan dalam Gambar 3.10.

```

fuzzy_servo | Arduino 1.8.1
File Edit Sketch Tools Help

fuzzy_servo
HS=HS*50;
}
//RULE 7, if salinitas normal and delta error negatif then servo kecil
else if (dkeR<0 && dkeD<0){
  HF7=max(dkeR,dkeD);
  HS7=HF7*30;
}
//RULE 8, if salinitas normal and delta error zero then servo kecil
else if (dkeR<0 && dkeD==0){
  HF8=max(dkeR,dkeD);
  HS8=HF8*30;
}
//RULE 9, if salinitas normal and delta error positif then servo kecil
else if (dkeR<0 && dkeD>0){
  HF9=max(dkeR,dkeD);
  HS9=HF9*30;
}
//RULE 10, if salinitas tinggi and delta error negatif then servo sangat kecil
else if (dkeT<0 && dkeD<0){
  HF10=max(dkeT,dkeD);
  HS10=HF10*2.5;
}
//RULE 11, if salinitas tinggi and delta error zero then servo sangat kecil
else if (dkeT<0 && dkeD==0){
  HF11=max(dkeT,dkeD);
  HS11=HF11*2.5;
}
//RULE 12, if salinitas tinggi and delta error positif then servo sangat kecil
else if (dkeT<0 && dkeD>0){
  HF12=max(dkeT,dkeD);
  HS12=HF12*2.5;
}
HF= (HF1+HF2+HF3+HF4+HF5+HF6+HF7+HF8+HF9+HF10+HF11+HF12);
HS=(HS1+HS2+HS3+HS4+HS5+HS6+HS7+HS8+HS9+HS10+HS11+HS12);
HS1a1Subat= (HS/HS);
}

Done Setup
Sketch uses 32774 bytes (4%) of program storage space. Maximum is 514288 bytes.
Ardu: SMART device 0x1100000 found
Erased flash
Done in 0.107 seconds
219
Arduino IDE (Programming Panel) on COM5

```

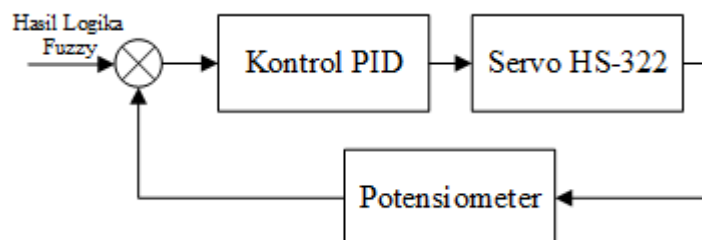
Gambar 3.10 Script Program untuk Proses Defuzzifikasi

3.4 Perancangan Sistem Kontrol PID

Kontrol PID pada penelitian ini digunakan untuk memperbaiki kelemahan motor servo HS-322 yang tidak bisa mencapai nilai sudut aktual sesuai dengan nilai masukan. Kontrol PID yang digunakan pada pembuatan alat pengendali tingkat salinitas ini adalah PID diskrit dengan masukan dari nilai masukan sudut yang dihasilkan dari proses pengendalian logika *fuzzy*.

Untuk menentukan nilai dari masing – masing parameter PID yaitu K_p , K_i dan K_d , digunakan metode *trial and error* dengan menggunakan dengan perancangan nilai awal $K_p=0,92$, $K_i=0,001$, $K_d=0,02$.

Motor Servo HS-322 akan digabungkan dengan potensiometer. Potensiometer tersebut akan dijadikan acuan pergerakan dari sudut motor servo dan akan diumpkan balikkan kepada kontroler untuk kemudian diolah dalam kendali PID agar pergerakan sudut motor servo dapat sesuai dengan masukan yang diberikan .Sistem kerja pengendalian PID dalam penelitian ditunjukkan dalam diagram blok yang terdapat dalam Gambar 3.11.



Gambar 3.11 Perancangan Kendali PID Servo HS-322

Dari Gambar 3.11, dapat dilihat bahwa potensiometer mengirimkan sinyal umpan balik kepada kontroler. Untuk mengimplementasikan potensiometer sebagai pembaca sudut motor servo dengan memanfaatkan nilai analog dari potensiometer yang disebabkan oleh perubahan sudut motor servo. Pengimplementasian potensiometer sebagai pembaca sudut motor servo yang kemudian dilanjutkan dengan proses penghitungan nilai parameter PID dilakukan dengan menggunakan *software Arduino IDE 1.8.1* yang ditunjukkan dalam Gambar 3.12.

```

Sketch name: Arduino IDE 1.8.1
File Edit Sketch Tools Help

Arduino IDE 1.8.1
Arduino IDE (Programming)

void setup() {
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);
}

void loop() {
  int P;
  int I;
  int D;
  float errorServo;
  float errorServoBelum;
  errorServo = nilaiSudut - nilaiSudutDesired;
  int Kp=0.5;
  int Ki=0.001;
  int Kd=0.02;
  P = Kp*errorServo;
  I = Ki*errorServo*errorServoBelum;
  D = Kd*errorServo - errorServoBelum;
  errorServoBelum = errorServo;
  float PID;
  PID = P + I + D;
}

Sketch Size:
Sketch uses 12772 bytes (4%) of program storage space. Maximum is 524288 bytes.
Serial 2000000000 device (usbToSerial) found
Upload Done
Done in 0.033 seconds

```

Gambar 3.12 *Script* Program Kontrol PID untuk Motor Servo

3.9 Perancangan Keluaran LCD

Pada perancangan alat pengendali tingkat kepekatan garam ini, digunakan LCD sebagai tampilan alat untuk memudahkan operator dalam mengetahui tingkat kepekatan garam yang ada di dalam air saat melakukan proses pengaraman.

LCD pada alat ini akan menampilkan nilai salinitas secara real time mulai dari saat alat mulai menyala hingga proses selesai. Untuk mengimplementasikan LCD yang dikontrol oleh Arduino Due, digunakan *software Arduino IDE 1.8.1*.

Script program untuk menampilkan nilai salinitas yang didapat dari sensor baume TDS-01 dapat dilihat dalam Gambar 3.13.

```

fuzzy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

fuzzy_saya
-----
setancreall1();
campilan();
baca();
fuzzifikasierror();
defuzzifikasi();
delay(1000);
}

void setancreall1() {
  average = analogRead(inputPin);
  if(average<120){
    salinitas = map(average, 120, 380, 0, 23);
  }
  else if(average<120){
  }
  delay(100);
}

void campilan(){
  loadServo(0,0);
  digitalWrite("Salinitas");
  loadServo(0,1);
  digitalWrite(salinitas);
  delay(1000);
}

void fuzzifikasierror(){
  e = 22 - salinitas;
  if (e<=11 || e>=23){
    dskR = 1;
  }
  else if (e<=11 || e>=23){
    dskB = (e-10)/2;
  }
}

Dino Saung
Sketch uses 32772 bytes (64%) of program storage space. Maximum is 514288 bytes.
Global variables use 1430 bytes of memory.
Done flash
Done in 0.133 seconds
-----

```

Gambar 3.13 Script Program Tampilan LCD

3.9 Perancangan Keseluruhan Sistem

Setelah semua perangkat dari sistem yang terdiri dari masukan sensor, sistem kendali logika *fuzzy* dan PID, keluaran servo dan LCD telah selesai dirancang, maka dapat dilakukan perancangan desain sistem secara menyeluruh dengan menggabungkan semua desain yang telah dibuat dalam ke dalam Arduino Due sebagai kontroler dengan menggunakan *software Arduino IDE 1.8.1*. Perancangan program secara keseluruhan dapat dilihat dalam Gambar 3.14 – Gambar 3.20.

```

fuzzy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

fuzzy_saya$
-----
**Alat Pengendali Tingkat Salinitas
*
* by RDT
*

//LIBRARY
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//I2C
LiquidCrystal_I2C lcd(0x27,16,2);

//servo
Servo myservo;
int pinpot = A1;
int motor;
int nilaiinputpot;

//Tampilan
float e; float de; float ea;
float dskR; float dskB; float dskG; float dskT;
float dskM0; float dskM1; float dskM2;
float NF1; float NF2; float NF3; float NF4; float NF5; float NF6; float NF7; float NF8; float NF9; float NF10; float NF11; float NF12; float NF13;
float NS1; float NS2; float NS3; float NS4; float NS5; float NS6; float NS7; float NS8; float NS9; float NS10; float NS11; float NS12; float NS13;
float nilaiSubst;

//sensor
const int numReadings = 15;
int readings[numReadings];
int readIndex = 0;
int total = 0;
int average = 0;
int inputPin = A0;
float salinitas;

Dino Saung
Sketch uses 32772 bytes (64%) of program storage space. Maximum is 514288 bytes.
Global variables use 1430 bytes of memory.
Done flash
Done in 0.133 seconds
-----

```

Gambar 3.14 Script Program Keseluruhan Bagian ke - 1

```

Arduy_saya | Arduino 1.8.1
File Edit Sketch Tools Help
Arduy_saya 9
void setup() {
  Serial.begin(9600);
  myServo.attach(2);
  pinMode(A1, INPUT);
  led.init();
  led.on();
  led.blinkLight();
}

void loop() {
  sensorAll();
  sampel();
  baca();
  fuzifikasierror();
  defuzifikasi();
  delay(2000);
}

void sensorAll() {
  average = analogRead(inputPin);
  if(average>120){
    salinitas = map(average, 120, 380, 0, 23);
  }
  else if(average<120){
  }
  delay(100);
}

void sampel() {
  led.setCursor(0,0);
  led.print("Salinitas ");
  led.setCursor(0,1);
  led.print(salinitas);
  delay(1000);
}

Data Dump
Sketch uses 32772 bytes (6%) of program storage space. Maximum is 512256 bytes.
Global Symbol device not declared found
Done flash
Done in 0.033 seconds
Arduino IDE (Programming Port) on COM6

```

Gambar 3.15 Script Program Keseluruhan Bagian ke - 2

```

Arduy_saya | Arduino 1.8.1
File Edit Sketch Tools Help
Arduy_saya 9
void defuzifikasierror() {
  e = 22 - salinitas;
  if (e<=12 || e<=-23){
    dkeR = 1;
  }
  else if (e>=18 || e<=-12){
    dkeB = (e-10)/2;
  }
  //salinitas rendah
  if(e>=8 || e<=-12){
    dkeI = (12-e)/4;
  }
  else if(e>=4 || e<=-8){
    dkeI = (e-4)/4;
  }
  //salinitas normal
  if(e>=3 || e<=-3){
    dkeI = (5-e)/2;
  }
  else if(e>=1 || e<=-3){
    dkeI = (e-1)/2;
  }
  //salinitas tinggi
  if(e>=5 || e<=-2){
    dkeI = (2-e)/1;
  }
  else if(e<=-1){
    dkeI = 1;
  }
}

Data Dump
Sketch uses 32772 bytes (6%) of program storage space. Maximum is 512256 bytes.
Global Symbol device not declared found
Done flash
Done in 0.033 seconds
Arduino IDE (Programming Port) on COM6

```

Gambar 3.16 Script Program Keseluruhan Bagian ke - 3

```

Arduy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

Arduy_saya
void fuzzifikasiDeltaerror()
{
  de = e - es;
  if (de<=1 && de>=1){
    de=0;
  }
  es = e;
  //delta error negatif
  if (de<=-1 && de<=-0.3)
  {
    ddeM= 1;
  }
  else if (de<=-0.3 && de<=-0.2)
  {
    ddeM= ((-0.2-de)/0.1);
  }
  //delta error zero
  else if (de<=-0.2 && de<=0)
  {
    ddeM= (de+0.3)/0.3;
  }
  else if (de<=0 && de<=0.3)
  {
    ddeM= ((0.3-de)/0.3);
  }
  //delta error positif
  else if (de>=0.2 && de>=0.3)
  {
    ddeM= ((de-0.2)/0.1);
  }
  else if (de>=0.3 && de<=1)
  {
    ddeM= 1;
  }
}
}

Dino Savoy
Sketch uses 32772 bytes (6%) of program storage space. Maximum is 514288 bytes.
Error: cannot open file: fuzzifikasi04.ino
Error: flash
Done in 0.033 seconds
Arduino IDE (Programming Frit on COM4)

```

Gambar 3.17 Script Program Keseluruhan Bagian ke - 4

```

Arduy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

Arduy_saya
void defuzzifikasi()
{
  //RULE 1, if salinitas sangat rendah and delta error negatif then servo besar
  if (dneR<0 && ddeM<0){
    NF1=max(dneR, ddeM);
    NS1=NF1*50;
  }
  //RULE 2, if salinitas sangat rendah and delta error zero then servo besar
  else if (dneR<0 && ddeM=0){
    NF2=max(dneR, ddeM);
    NS2=NF2*50;
  }
  //RULE 3, if salinitas sangat rendah and delta error positif then servo besar
  else if (dneR<0 && ddeM>0){
    NF3=max(dneR, ddeM);
    NS3=NF3*50;
  }
  //RULE 4, if salinitas rendah and delta error negatif then servo agak besar
  else if (dneR>0 && ddeM<0){
    NF4=max(dneR, ddeM);
    NS4=NF4*50;
  }
  //RULE 5, if salinitas rendah and delta error zero then servo agak besar
  else if (dneR>0 && ddeM=0){
    NF5=max(dneR, ddeM);
    NS5=NF5*50;
  }
  //RULE 6, if salinitas rendah and delta error positif then servo agak besar
  else if (dneR>0 && ddeM>0){
    NF6=max(dneR, ddeM);
    NS6=NF6*50;
  }
  //RULE 7, if salinitas normal and delta error negatif then servo kecil
  else if (dneR>0 && ddeM<0){
    NF7=max(dneR, ddeM);
    NS7=NF7*30;
  }
}
}

Dino Savoy
Sketch uses 32772 bytes (6%) of program storage space. Maximum is 514288 bytes.
Error: cannot open file: fuzzifikasi04.ino
Error: flash
Done in 0.033 seconds
Arduino IDE (Programming Frit on COM4)

```

Gambar 3.18 Script Program Keseluruhan Bagian ke - 5


```

Azzy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

%idy_saya %
//RULE 0, if salinitas normal and delta error zero then servo kecil
else if(dkeD==0 || dkeE==0){
  HFO=map(dkeD,dkeE);
  HFO=HFO*30;
}
//RULE 9, if salinitas normal and delta error positif then servo kecil
else if(dkeD==0 || dkeE>0){
  HFO=map(dkeD,dkeE);
  HFO=HFO*30;
}
//RULE 10, if salinitas tinggi and delta error negatif then servo sangat kecil
else if(dkeD>0 || dkeE==0){
  HFO=map(dkeD,dkeE);
  HFO=HFO*1.5;
}
//RULE 11, if salinitas tinggi and delta error zero then servo sangat kecil
else if(dkeD>0 || dkeE==0){
  HFO=map(dkeD,dkeE);
  HFO=HFO*1.5;
}
//RULE 12, if salinitas tinggi and delta error positif then servo sangat kecil
else if(dkeD>0 || dkeE>0){
  HFO=map(dkeD,dkeE);
  HFO=HFO*1.5;
}
}
}

HFT= (HFT1+HFT2+HFT3+HFT4+HFT5+HFT6+HFT7+HFT8+HFT9+HFT10+HFT11+HFT12);
HST= (HST1+HST2+HST3+HST4+HST5+HST6+HST7+HST8+HST9+HST10+HST11+HST12);
HlialSudut= (HST/HFT);

Serial.println("Salinitas :");
Serial.println(salinitas);
Serial.println("Nilai Error :");
Serial.println(e);
Serial.println("HFT :");
Serial.println(HFT);

Date Stamp
Sketch uses 32772 bytes (64%) of program storage space. Maximum is 524288 bytes.
Serial monitor device not detected found.
Erase flash
Done in 0.033 seconds
Arduino IDE (Programming Port on COM6)

```

Gambar 3.19 Script Program Keseluruhan Bagian ke - 6

```

Azzy_saya | Arduino 1.8.1
File Edit Sketch Tools Help

%idy_saya %
Serial.println(e);
Serial.println("HFT :");
Serial.println(HFT);
Serial.println("HST :");
Serial.println(HST);
Serial.println("NilaiSudut :");
Serial.println(NilaiSudut);
delay(1000);
}

void baca() {
  sudut = analogRead(A1);
  nilaiOutputpot = map(sudut, 100, 360, 0, 99);
}

void PID() {
  int P;
  int I;
  int D;
  float errorServo;
  float errorServoSebelum;
  errorServo= NilaiSudut - nilaiOutputpot;
  int Kp=0.92;
  int Ki=0.001;
  int Kd=0.02;
  P = Kp*errorServo;
  I = Ki*errorServo*errorServoSebelum;
  D = Kd*errorServo*errorServoSebelum;
  errorServoSebelum = errorServo;
  float PID;
  PID = P + I + D;
}

Date Stamp
Sketch uses 32772 bytes (64%) of program storage space. Maximum is 524288 bytes.
Serial monitor device not detected found.
Erase flash
Done in 0.033 seconds
Arduino IDE (Programming Port on COM6)

```

Gambar 3.20 Script Program Keseluruhan Bagian ke - 7