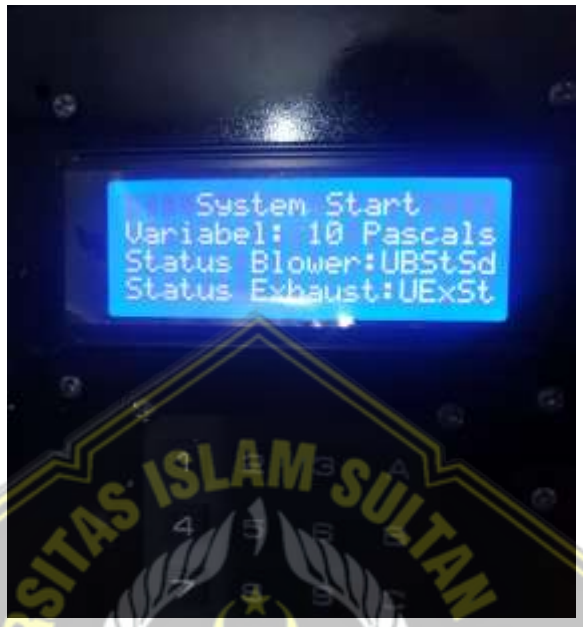
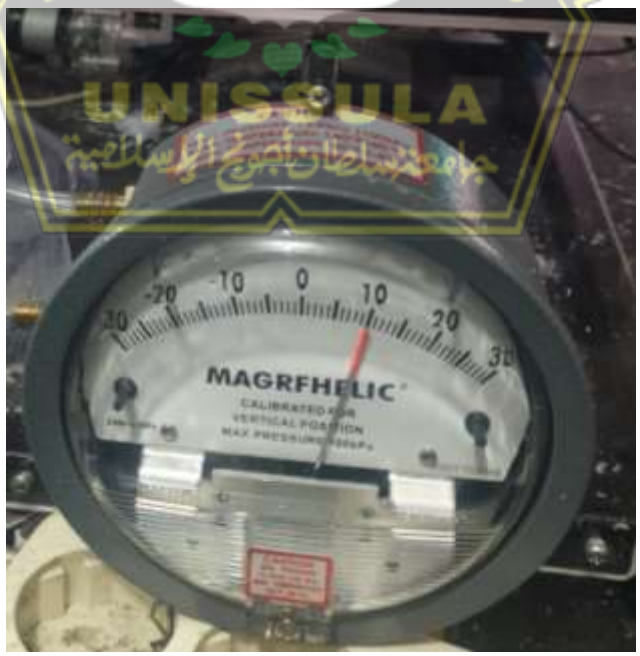
LAMPIRAN

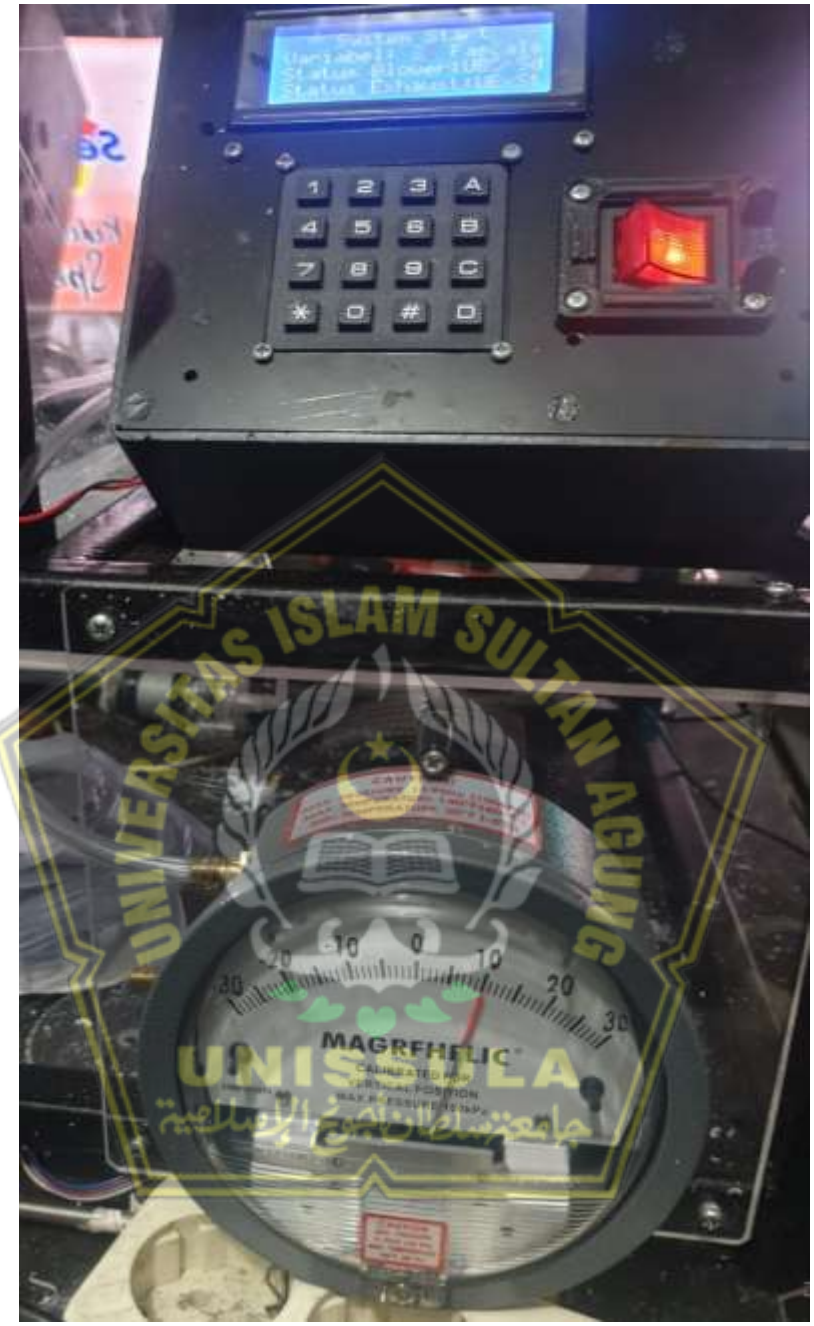
Gambar ketika posisi pengukuran sesuai dengan target atau set point.



Gambar alat ukur tekanan terkalibrasi



Gambar tampilan pengukuran sistem dan pengukuran alat ukur terkalibrasi



Program

Program Fuzzy:

```
#include <LiquidCrystal.h>
#include <Q2HX711.h>
float error_1,error,dError;
float anggotaErrorNB,anggotadErrorNB,anggotapwmNB;
float anggotaErrorNS,anggotadErrorNS,anggotapwmNS;
float anggotaErrorNK,anggotadErrorNK,anggotapwmNK;
float anggotaErrorTS,anggotadErrorTS,anggotapwmTS;
float anggotaErrorPK,anggotadErrorPK,anggotapwmPK;
float anggotaErrorPS,anggotadErrorPS,anggotapwmPS;
float anggotaErrorPB,anggotadErrorPB,anggotapwmPB;
float rule1,rule2,rule3,rule4,rule5,rule6,rule7,rule8,rule9,rule10;
float rule11a,rule11b,rule12a,rule12b,rule13a,rule13b,rule14a,rule14b,
rule15a,rule15b;
float rule16a,rule16b,rule17a,rule17b,rule18a,rule18b,rule19a,rule19b,
rule20a,rule20b;
float rule21a,rule21b,rule22a,rule22b,rule23a,rule23b;
float rule24,rule25,rule26,rule27,rule28,rule29,rule30,rule31,rule32,rule33,
rule34,rule35;
float rule36,rule37,rule38,rule39,rule40,rule41,rule42,rule43,rule44,rule45,
rule46,rule47,rule48,rule49;

float rule1x,rule2x,rule3x,rule4x,rule5x,rule6x,rule7x,rule8x,rule9x,rule10x;
float rule11x,rule12x,rule13x,rule14x,rule15x,rule16x,rule17x,rule18x,
rule19x,rule20x;
float rule21x,rule22x,rule23x;
float rule24x,rule25x,rule26x,rule27x;
float rule28ax,rule28bx,rule29ax,rule29bx,rule30ax,rule30bx,rule31ax,rule31bx;
float rule32ax,rule32bx,rule33ax,rule33bx,rule34ax,rule34bx,rule35ax,rule35bx,
rule36ax,rule36bx,rule37ax,rule37bx,rule38ax,rule38bx,rule39ax,rule39bx;
```

```
float rule40x,rule41x,rule42x,rule43x,rule44x,rule45x,rule46x,rule47x,rule48x,
rule49x;
int def;
int defx;
float tekanan=0;
const int hx711_data_pin = A5;
const int hx711_clock_pin = A4;
const int hx711_data_pin1 = 7;
const int hx711_clock_pin1 = 4 ;
Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); //s1
Q2HX711 hx7112(hx711_data_pin1, hx711_clock_pin1); //s2
const int rs = 13, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int exhaust=5;
int blower=6;
float vs;
float p1;
float p2;
float calp1;
float calp2;
float cal;

void error_tekanan_NB(){
if(error<=7){anggotaErrorNB = 1;} else if(error > 7 && error <8){anggotaErrorNB
= (8 - error)/1;} else if (error >= 0) {anggotaErrorNB = 0;}
}
void error_tekanan_NS(){
if (error <=7){ anggotaErrorNS = 0;} else if (error > 7 && error <8 ){
anggotaErrorNS = (error-7)/1;}
else if (error >= 8 && error < 9){ anggotaErrorNS = (9 - error)/1;} else if (error >=
9) {anggotaErrorNS = 0;}
}
```

```
void error_tekanan_NK(){
if (error <=8){ anggotaErrorNK = 0;} else if (error > 8 && error <9 ){ anggotaErrorNK = (error-8)/1;}
else if (error >= 9 && error < 10){ anggotaErrorNK = (10 - error)/1;} else if (error >= 10) {anggotaErrorNK = 0;}
}

void error_tekanan_TS(){
if (error <=9){ anggotaErrorTS = 0;} else if (error > 9 && error <10 ){ anggotaErrorTS = (error-9)/1;}
else if (error >= 10 && error < 11){ anggotaErrorTS = (11 - error)/1;} else if (error >= 11) {anggotaErrorTS = 0;}
}
void error_tekanan_PK(){
if (error <=10){ anggotaErrorPK = 0;} else if (error > 10 && error <11 ){ anggotaErrorPK = (error-10)/1;}
else if (error >= 11 && error < 12){ anggotaErrorPK = (12 - error)/1;} else if (error >= 12) {anggotaErrorPK = 0;}
}
void error_tekanan_PS(){
if (error <=10){ anggotaErrorPS = 0;} else if (error > 11 && error <12 ){ anggotaErrorPS = (error-11)/1;}
else if (error >= 12 && error < 13){ anggotaErrorPS = (13 - error)/1;} else if (error >= 13) {anggotaErrorPS = 0;}
}
void error_tekanan_PB(){
if (error >= 13){ anggotaErrorPB = 1;}
else if (error < 13 && error > 12){ anggotaErrorPB = (error - 12)/1;}
else if (error < 12 ) {anggotaErrorPB = 0;}
}
```

```
//=====================
void dError_tekanan_NB(){
if (error <=7){ anggotadErrorNB = 1;} else if (dError > 7 && dError < 8){
anggotadErrorNB = (8 - dError)/1;} else if (dError >= 0) {anggotadErrorNB = 0;}
}
void dError_tekanan_NS(){
if (dError <=7){ anggotadErrorNS = 0;} else if (dError > 7 && dError <8 ){
anggotadErrorNS = (dError-7)/1;}
else if (dError >= 8 && dError < 9){ anggotadErrorNS = (9 - dError)/1;} else if
(dError >= 9) {anggotadErrorNS = 0;}
}
void dError_tekanan_NK(){
if (dError <=8){ anggotadErrorNK = 0;} else if (dError > 8 && dError <9 ){
anggotadErrorNK = (dError-8)/1;}
else if (dError >= 9 && dError < 10){ anggotadErrorNK = (10 - dError)/1;} else if
(dError >= 10) {anggotadErrorNK = 0;}
}
void dError_tekanan_TS(){
if (dError <=9){ anggotadErrorTS = 0;} else if (dError > 9 && dError <10 ){
anggotadErrorTS = (dError-9)/1;}
else if (dError >= 10 && dError < 11){ anggotadErrorTS = (11 - dError)/1;} else if
(dError >= 11) {anggotadErrorTS = 0;}
}

void dError_tekanan_PK(){
if (dError <=10){ anggotadErrorPK = 0;} else if (dError > 10 && dError <11 ){
anggotadErrorPK = (dError-10)/1;}
else if (dError >= 11 && dError < 12){ anggotadErrorPK = (12 - dError)/1;} else
if (dError >= 12) {anggotadErrorPK = 0;}
}
void dError_tekanan_PS(){
```

```
if (dError <=10){ anggotadErrorPS = 0;} else if (dError > 11 && dError <12 ){
anggotadErrorPS = (dError-11)/1;}
else if (dError >= 12 && dError < 13){ anggotadErrorPS = (13 - dError)/1;} else if
(dError >= 13) {anggotadErrorPS = 0;}
}
void dError_tekanan_PB(){
if (dError >= 13){ anggotadErrorPB = 1;}
else if (dError < 13 && dError > 12){ anggotadErrorPB = (dError - 12)/1;}
else if (dError < 12 ){anggotadErrorPB = 0;}
}

void fuzifikasi() {
error_tekanan_NB();
error_tekanan_NS();
error_tekanan_NK();
error_tekanan_TS();
error_tekanan_PK();
error_tekanan_PS();
error_tekanan_PB();

dError_tekanan_NB();
dError_tekanan_NS();
dError_tekanan_NK();
dError_tekanan_TS();
dError_tekanan_PK();
dError_tekanan_PS();
dError_tekanan_PB();
}

float Min (float a, float b) {
if (a < b) {
return a;
```

```
}
else if (b < a) {
return b;
}
else return a;
}

void rule() {
fuzifikasi();
//NB Cepat
float x1 = Min (anggotaErrorNB,anggotadErrorNB);
rule1 = 150 + (100 * x1);
rule1x=150-(100*x1);
float x2 = Min (anggotaErrorNB,anggotadErrorNS);
rule2 = 150 + (100 * x2);
rule2x=150-(100*x2);
float x3 = Min (anggotaErrorNB,anggotadErrorNK);
rule3 = 150 + (100 * x3);
rule3x=150-(100*x3);
float x4 = Min (anggotaErrorNB,anggotadErrorTS);
rule4 = 150 + (100 * x4);
rule4x=150-(100*x4);
float x5 = Min (anggotaErrorNB,anggotadErrorPK);
rule5 = 150 + (100 * x5);
rule5x=150-(100*x5);
float x6 = Min (anggotaErrorNS,anggotadErrorNB);
rule6 = 150 + (100 * x6);
rule6x=150-(100*x6);
float x7 = Min (anggotaErrorNS,anggotadErrorNS);
rule7 = 150 + (100 * x7);
rule7x=150-(100*x7);
float x8 = Min (anggotaErrorNS,anggotadErrorNK);
```

```
rule8 = 150 + (100 * x8);
rule8x=150-(100*x8);
float x9 = Min (anggotaErrorNK,anggotadErrorNB);
rule9 = 150 + (100 * x9);
rule9x=150-(100*x9);
float x10 = Min (anggotaErrorNK,anggotadErrorNS);
rule10 = 150 + (100 * x10);
rule10x=150-(100*x10);


//NS Sedang
float x11=Min(anggotaErrorNB,anggotadErrorPS);
rule11a =100+(50*x11);
rule11b =200-(50*x11);
rule11x=150-(100*x11);
float x12=Min(anggotaErrorNB,anggotadErrorPB);
rule12a =100+(50*x12);
rule12b =200-(50*x12);
rule12x=150-(100*x12);
float x13=Min(anggotaErrorNS,anggotadErrorTS);
rule13a =100+(50*x13);
rule13b =200-(50*x13);
rule13x=150-(100*x13);
float x14=Min(anggotaErrorNS,anggotadErrorPK);
rule14a =100+(50*x14);
rule14b =200-(50*x14);
rule14x=150-(100*x14);
float x15=Min(anggotaErrorNS,anggotadErrorPS);
rule15a =100+(50*x15);
rule15b =200-(50*x15);
rule15x=150-(100*x15);
float x16=Min(anggotaErrorNK,anggotadErrorNK);
```

```
rule16a =100+(50*x16);
rule16b =200-(50*x16);
rule16x=150-(100*x16);
float x17=Min(anggotaErrorTS,anggotadErrorNB);
rule17a =100+(50*x17);
rule17b =200-(50*x17);
rule17x=150-(100*x17);
float x18=Min(anggotaErrorTS,anggotadErrorNS);
rule18a =100+(50*x18);
rule18b =200-(50*x18);
rule18x=150-(100*x18);


//NK Pelan
float x19 = Min (anggotaErrorNS,anggotadErrorPB);
rule19a = 50 + (50 * x19) ;
rule19b = 150 - (50 * x19) ;
rule19x=150-(100*x19);
float x20 = Min (anggotaErrorNK,anggotadErrorTS);
rule20a = 50 + (50 * x20) ;
rule20b = 150 - (50 * x20) ;
rule20x=150-(100*x20);
float x21 = Min (anggotaErrorNK,anggotadErrorPK);
rule21a = 50 + (50 * x21) ;
rule21b = 150 - (50 * x21) ;
rule21x=150-(100*x21);
float x22 = Min (anggotaErrorNK,anggotadErrorPS);
rule22a = 50 + (50 * x22) ;
rule22b = 150 - (50 * x22) ;
rule22x=150-(100*x22);
float x23 = Min (anggotaErrorTS,anggotadErrorNK);
rule23a = 50 + (50 * x23) ;
```

```
rule23b = 150 - (50 * x23) ;
rule23x=150-(100*x23);


//TS STOP
float x24 = Min (anggotaErrorNK,anggotadErrorPB);
rule24 = 150 - (100 * x24);
rule24x=150-(100*x24);
float x25 = Min (anggotaErrorTS,anggotadErrorTS);
rule25 = 150 - (100 * x25);
rule25x=150-(100*x25);
float x26 = Min (anggotaErrorPK,anggotadErrorNB);
rule26 = 150 - (100  * x26);
rule26x=150-(100*x26);
float x27 = Min (anggotaErrorPK,anggotadErrorNS);
rule27 = 150 - (100  * x27);
rule27x=150-(100*x27);


//PK
float x28 = Min (anggotaErrorTS,anggotadErrorPK);
rule28 = 150 - (100  * x28);
rule28ax = 50 + (50 * x28) ;
rule28bx = 150 - (50 * x28) ;
float x29 = Min (anggotaErrorTS,anggotadErrorPS);
rule29 = 150 - (100  * x29);
rule29ax = 50 + (50 * x29) ;
rule29bx = 150 - (50 * x29) ;
float x30 = Min (anggotaErrorPK,anggotadErrorNK);
rule30 = 150 - (100  * x30);
rule30ax = 50 + (50 * x30) ;
rule30bx = 150 - (50 * x30) ;
```

```
float x31 = Min (anggotaErrorPK,anggotadErrorTS);
rule31 = 150 - (100  * x31);
rule31ax = 50 + (50 * x31) ;
rule31bx = 150 - (50 * x31) ;
float x32 = Min (anggotaErrorPS,anggotadErrorNB);
rule32 = 150 - (100  * x32);
rule32ax = 50 + (50 * x32) ;
rule32bx = 150 - (50 * x32) ;
float x33 = Min (anggotaErrorPS,anggotadErrorNS);
rule33 = 150 - (100  * x33);
rule33ax = 50 + (50 * x33) ;
rule33bx = 150 - (50 * x33) ;


//PS
float x34 = Min(anggotaErrorTS,anggotadErrorPB);
rule34 = 150 - (100  * x34);
rule34ax =100+(50*x34);
rule34bx =200-(50*x34);
float x35 = Min(anggotaErrorPK,anggotadErrorPK);
rule35 = 150 - (100  * x35);
rule35ax =100+(50*x35);
rule35bx =200-(50*x35);
float x36 = Min(anggotaErrorPS,anggotadErrorNK);
rule36 = 150 - (100  * x36);
rule36ax =100+(50*x36);
rule36bx =200-(50*x36);
float x37 = Min(anggotaErrorPS,anggotadErrorTS);
rule37 = 150 - (100  * x37);
rule37ax =100+(50*x37);
rule37bx =200-(50*x37);
float x38 = Min(anggotaErrorPB,anggotadErrorNB);
```

```
rule38 = 150 - (100 * x38);
rule38ax =100+(50*x38);
rule38bx =200-(50*x38);
float x39 = Min(anggotaErrorPB,anggotadErrorNS);
rule39 = 150 - (100  * x39);
rule39ax =100+(50*x39);
rule39bx =200-(50*x39);


//PB
float x40 = Min(anggotaErrorPK,anggotadErrorPS);
rule40 =150 - (100  * x40);
rule40x = 150 + (100 * x40);
float x41 = Min(anggotaErrorPK,anggotadErrorPB);
rule41 = 150 - (100  * x41);
rule41x = 150 + (100 * x41);
float x42 = Min(anggotaErrorPS,anggotadErrorPK);
rule42 = 150 - (100  * x42);
rule42x = 150 + (100 * x42);
float x43 = Min(anggotaErrorPS,anggotadErrorPS);
rule43 = 150 - (100  * x43);
rule43x = 150 + (100 * x43);
float x44 = Min(anggotaErrorPS,anggotadErrorPB);
rule44 = 150 - (100  * x44);
rule44x = 150 + (100 * x44);
float x45 = Min(anggotaErrorPB,anggotadErrorNK);
rule45 = 150 - (100  * x45);
rule45x = 150 + (100 * x45);
float x46 = Min(anggotaErrorPB,anggotadErrorTS);
rule46 = 150 - (100  * x46);
rule46x = 150 + (100 * x46);
float x47 = Min(anggotaErrorPB,anggotadErrorPK);
```

```
rule47 = 150 - (100  * x47);
rule47x = 150 + (100 * x47);
float x48 = Min(anggotaErrorPB,anggotadErrorPS);
rule48 = 150 - (100  * x48);
rule48x = 150 + (100 * x8);
float x49 = Min(anggotaErrorPB,anggotadErrorPB);
rule49 = 150 - (100  * x49);
rule49x = 150 + (100 * x49);

def = ((x1 * rule1) + (x2 * rule2) +(x3 * rule3)+ (x4 * rule4) +(x5 * rule5)+(x6 *
rule6) + (x7 * rule7) +(x8 * rule8)+ (x9 * rule9) +(x10 * rule10) + (x11 * rule11a)
+(x11 * rule11b)+ (x12 * rule12a) +(x12 * rule12b)+ (x13 * rule13a)+ (x13 *
rule13b)+ (x14 * rule14a)+ (x14 * rule14b)+ (x15 * rule15a)+ (x15 * rule15b)+
(x16 * rule16a)+ (x16 * rule16b)+ (x17 * rule17a)+ (x17 * rule17b)+ (x18 *
rule18a)+ (x18 * rule18b)+ (x19 * rule19a)+ (x19 * rule19b)+ (x20 * rule20a)+
(x20 * rule20b)+ (x21 * rule21a)+ (x21 * rule21b)+ (x22 * rule22a)+ (x22 *
rule22b)+ (x23 * rule23a)+(x24 * rule24) + (x25 * rule25) +(x26 * rule26)+ (x27
* rule27) +(x28 * rule28)+(x29 * rule29) + (x30 * rule30) +(x31 * rule31)+ (x32 *
rule32) +(x33 * rule33)+(x34 * rule34) + (x35 * rule35) +(x36 * rule36)+ (x37 *
rule37) +(x38 * rule38)+(x39 * rule39) + (x40 * rule40) +(x41 * rule41)+ (x42 *
rule42) +(x43 * rule43)+(x44 * rule44) + (x45 * rule45) +(x46 * rule46)+ (x47 *
rule47) +(x48 * rule48)+(x49 * rule49)) / (x1 + x2+x3 + x4+x5 + x6 + x7 + x8+x9
+ x10 +x11+ x11+x12 + x12+x13 + x13+x14 + x14 + x15 + x15+x16 + x16 +x17+
x17+x18 + x18+x19 + x19+x20 + x20 + x21 + x21+x22 + x22 +x23+ x23+x24 +
x25+x26 + x27+x28 + x29 + x30 + x31+x32 + x33 +x34+ x35+x36 + x37 +x38+
x39+ x40+x41 + x42 +x43+ x44+x45 + x46 +x47+ x48+x49);

defx = ((x1 * rule1x) + (x2 * rule2x) +(x3 * rule3x)+ (x4 * rule4x) +(x5 *
rule5x)+(x6 * rule6x) + (x7 * rule7x) +(x8 * rule8x)+ (x9 * rule9x) +(x10 *
rule10x) + (x11 * rule11x) +(x12 * rule12x)+ (x13 * rule13x) +(x14 * rule14x)+
(x15 * rule15x)+ (x16 * rule16x)+ (x17 * rule17x)+ (x18 * rule18x)+ (x19 *
rule19x)+ (x20 * rule20x)+ (x21 * rule21x)+ (x22 * rule22x)+ (x23 * rule23x)+
```

```
(x24 * rule24x)+ (x25 * rule25x)+ (x26 * rule26x)+ (x27 * rule27x)+ (x28 * rule28ax)+ (x28 * rule28bx)+ (x29 * rule29ax)+ (x29 * rule29bx)+ (x30 * rule30ax)+ (x30 * rule30bx)+ (x31 * rule31ax)+(x31 * rule31bx) + (x32 * rule32ax) +(x32 * rule32bx)+ (x33 * rule33ax) +(x33 * rule33bx)+(x34 * rule34ax) + (x34 * rule34bx) +(x35 * rule35ax)+ (x35 * rule35bx)+(x36 * rule36ax) + (x36 * rule36bx) +(x37 * rule37ax)+ (x37 * rule37bx) +(x38 * rule38ax)+(x38 * rule38bx) + (x39 * rule39ax) +(x39 * rule39bx)+ (x40 * rule40x) +(x41 * rule41x)+ (x42 * rule42x) + (x43 * rule43x) +(x44 * rule44x)+ (x45 * rule45x) +(x46 * rule46x)+(x47 * rule47x)+(x48 * rule48x)+(x49 * rule49x)) / (x1 + x2+x3 + x4+x5 + x6 + x7 + x8+x9 + x10 +x11+ x12+x13 + x14+x15 + x16+x17 + x18 + x19 + x20+x21 + x22 +x23+ x24+x25 + x26+x27 + x28+x28 + x29 + x29 + x30+x30 + x31 +x31+ x32+x32 + x33+x33 + x34+x34 + x35 + x35 + x36+x36 + x37 +x37+ x38+x38 + x39 +x39+ x40+ x41+x42 + x43 +x44+ x45+x46 + x47 +x48+ x49)-
50;
}

void runFuzzy() {
error_1 = error;
bacaTekanan();
error = tekanan;
dError = error_1;
rule();
analogWrite(blower, def);
analogWrite(exhaust, defx);
delay(100);
}

void setup()
{
 pinMode(exhaust,OUTPUT);
 pinMode(blower,OUTPUT);
 lcd.begin(20, 4);
```

```
//Serial.begin(9600);
calib();
}
void loop()
{
 main:
  digitalWrite(blower,LOW);
  digitalWrite(exhaust,LOW);
 start1();
}

void calib(){
for (int x=0; x<100; x++){
calp1=hx711.read()/100.0;
calp2=hx7112.read()/100.0;
cal=calp2-calp1;
lcd.setCursor(0,1);
lcd.print(" Sensor Calibration ");
lcd.setCursor(0,3);
lcd.print("       WAIT!!       ");
delay(10);
}
lcd.clear();
}

void start1(){
xx:
lcd.setCursor(0,0);
lcd.print("    Fuzzy  Start    ");
lcd.setCursor(0,1);
lcd.print("Tekanan:");
p1=hx711.read()/100.0+cal;
```

```
p2=hx7112.read()/100.0;
vs=p1-p2;
vs=vs/10+2;
if (vs<10){
lcd.setCursor(9,1);
lcd.print(vs);
lcd.setCursor(12,1);
lcd.print("  ");
lcd.setCursor(14,1);
lcd.print("Pa");
}
if (vs>9){
lcd.setCursor(9,1);
lcd.print(vs);
 lcd.setCursor(13,1);
lcd.print(" ");
lcd.setCursor(14,1);
lcd.print("Pa");
}
runFuzzy();
if (def>=100){
lcd.setCursor(0,2);
lcd.print("Blower :");
lcd.setCursor(9,2);
lcd.print(def);
}
if (def<100){
lcd.setCursor(0,2);
lcd.print("Blower :");
lcd.setCursor(9,2);
lcd.print(def);
lcd.setCursor(11,2);
```

```
lcd.print(" ");
}
if (defx>=100){
lcd.setCursor(0,2);
lcd.print("Exhaust:");
lcd.setCursor(9,2);
lcd.print(defx);
}
if (defx<100){
lcd.setCursor(0,3);
lcd.print("Exhaust:");
lcd.setCursor(9,3);
lcd.print(defx);
lcd.setCursor(11,3);
lcd.print(" ");
}
goto xx;
}

void bacaTekanan() {
//float f=analogRead(A0); //Nilai sensor
tekanan =vs;//map(f,0,1023,0*100,20*100)/100.0;
delay(100);
}
```

Program PID:

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Q2HX711.h>

const byte hx711_data_pin = A5;
const byte hx711_clock_pin = A4;
```

```
const byte hx711_data_pin1 = 7;
const byte hx711_clock_pin1 = 6 ;

Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); //s1
Q2HX711 hx7112(hx711_data_pin1, hx711_clock_pin1); //s2

const int rs = 13, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {A3, A2, A1, A0};
byte colPins[COLS] = {2, 3, 4, 5};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
long p1;
long p2;
long u;
long vs;
void setup() {
lcd.begin(20, 4);
lcd.setCursor(0,0);
lcd.print("   Sistem Kendali   ");
lcd.setCursor(0,1);
lcd.print("  Tekanan Posistif  ");
```

```
lcd.setCursor(0,2);
lcd.print("  Ruang  Operasi  ");
lcd.setCursor(0,3);
lcd.print("--------------------");
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("   Susilo Ari W   ");
lcd.setCursor(0,2);
lcd.print(" (Loading Sistem) ");
lcd.setCursor(0,3);
lcd.print("=");
delay(1000);
lcd.setCursor(0,3);
lcd.print("===");
delay(1000);
lcd.setCursor(0,3);
lcd.print("======");
delay(500);
lcd.setCursor(0,3);
lcd.print("========");
delay(500);
lcd.setCursor(0,3);
lcd.print("============");
delay(300);
lcd.setCursor(0,3);
lcd.print("===============");
delay(300);
lcd.setCursor(0,3);
lcd.print("==================");
delay(100);
lcd.setCursor(0,3);
```

```
lcd.print("====================");
delay(100);
lcd.clear();
}

void loop() {
  main:
  monitorx();
  goto main;
}

void monitorx(){
 char customKey = customKeypad.getKey();
 if (customKey=='A'){
 lcd.clear();
 start1();
  }
 p1=hx711.read()/100.0+1500;
 p2=hx7112.read()/100.0;
lcd.setCursor(0,0);
lcd.print("   monitor Sistem   ");
lcd.setCursor(0,1);
lcd.print("Kendali: PID");
lcd.setCursor(0,2);
lcd.print("S1:");
lcd.setCursor(3,2);
lcd.print(p1);
lcd.setCursor(10,2);
lcd.print("S2:");
lcd.setCursor(14,2);
lcd.print(p2);
lcd.setCursor(0,3);
```

```
lcd.print("D Start, A Menu");
delay(200);
}

void start1(){
 xx:
  char customKey = customKeypad.getKey();
  if (customKey=='D'){
 lcd.clear();
 loop();
  }
p1=hx711.read()/100.0+1500;
p2=hx7112.read()/100.0;
vs=p1-p2;
vs=vs/10-15;
if (vs<10){
lcd.setCursor(10,1);
lcd.print(vs);
 lcd.setCursor(11,1);
lcd.print(" ");
lcd.setCursor(13,1);
lcd.print("Pascals");
}
if (vs>=10){
lcd.setCursor(10,1);
lcd.print(vs);
 lcd.setCursor(12,1);
lcd.print(" ");
lcd.setCursor(13,1);
lcd.print("Pascals");
}
if(vs>10){
```

```
lcd.setCursor(0,2);
lcd.print("Status Blower: OFF");
}
if(vs<=10){
lcd.setCursor(0,2);
lcd.print("Status Blower: ON ");
}
if(vs>11){
lcd.setCursor(0,3);
lcd.print("Status Exhaust: ON ");
}
if(vs<=10){
lcd.setCursor(0,3);
lcd.print("Status Exhaust: OFF ");
}
lcd.setCursor(0,0);
lcd.print("    System Start    ");
lcd.setCursor(0,1);
lcd.print("Variabel:");
lcd.setCursor(0,2);
lcd.print("Status Blower:");
lcd.setCursor(0,3);
lcd.print("Status Exhaust:");
delay(200);
goto xx;
}
```