

RANCANG BANGUN SISTEM KONTROL KECEPATAN DAN KETEPATAN PENGHENTIAN PROSES PENGELASAN PADA PROSES PENGELASAN PIPA

Candra Dwi Cahyono

Fakultas Teknik, Program Studi Teknik Mesin
Universitas Muria Kudus
Email: candradwicahyono.cdc@gmail.com

Qomaruddin

Fakultas Teknik, Program Studi Teknik Mesin
Universitas Muria Kudus
Email: qomaruddin@umk.ac.id

Hera Setiawan

Fakultas Teknik, Program Studi Teknik Mesin
Universitas Muria Kudus
Email: hera.setiawan@umk.ac.id

Akhmad Zidni Hudaya

Fakultas Teknik, Program Studi Teknik Mesin
Universitas Muria Kudus
Email: akhmad.zidni@umk.ac.id

ABSTRAK

Proses pengelasan melingkar banyak dijumpai pada proses manufaktur di dunia industri seperti proses pengelasan pada rangka berbentuk pipa, tabung tekan, pipa distribusi dan lain-lain. Pada proses pengelasan melingkar secara manual sering kali ditemukan cacat las karena adanya kesulitan pengelasan pada bidang yang melingkar. Untuk mengatasi permasalahan ini, perlu dibuat alat bantu pemutar benda kerja pada proses pengelasan dan dapat diatur kecepatannya. Tujuan dari penelitian ini adalah melakukan rancang bangun sistem kontrol urutan proses, kecepatan dan ketepatan penghentian proses pengelasan pada proses pengelasan pipa. Metodologi penelitian ini dimulai dengan kajian pustaka, analisa kebutuhan sistem kontrol, desain sistem kontrol, pembuatan dan pengujian alat. Sebagai hasil, peralatan bantu proses pengelasan pada benda kerja yang melingkar telah berhasil dibuat. Beberapa parameter proses manufaktur dapat ditampilkan dalam LCD. Kecepatan pengelasan benda kerja pada tampilan LCD dibandingkan dengan pengujian dengan *tacho meter* mempunyai persentase kesalahan berturut-turut berkisar 1,5 ~ 1,7% dan 1,75 ~ 2% pada *range* pengujian antara 860 ~ 400 rpm.

Kata kunci : kecepatan pengelasan, pengelasan pipa, sistem kontrol

ABSTRACT

The Circular welding processes are often encountered in various manufacturing processes in the industrial production such as welding processes on the circular pipes frame, compressed tubes, distribution pipes and others. The manual circular welding process are often found welding defects, due to welding difficulties in the circular plane. To overcome this problem, it is necessary to make a workpiece turning tool in the validation process and the speed can be adjusted. The objective of research is to design and make control system of process sequence, speed and accuracy of the welding process termination in the pipe welding process. The research methodology is preseded by initial literature review, analysis of control system requirements,

control system design, manufacturing and testing of tools. The result was welding aids on circular workpieces have been made successfully. Some manufacturing process parameters was displayed on the LCD. The welding speed of the workpiece on the LCD display compared to testing with a tacho meter has a percentage of errors ranging from 1.5 ~ 1.7% and 1.75 ~ 2% in the test range between 860 ~ 400 rpm.

Keywords: *welding speed, pipe welding, control system.*

1. PENDAHULUAN

Proses pengelasan melingkar banyak dijumpai pada proses manufaktur di dunia industri seperti proses pengelasan pada rangka berbentuk pipa, tabung tekan, pipa distribusi dan lain-lain. Proses pengelasan ini dapat dilakukan dengan dua cara yaitu benda kerja diam dan elektroda las bergerak melingkar, atau benda kerja berputar dan electrode las diam. Cara pertama banyak dijumpai pada pengelasan manual dimana operator las melakukan proses pengelasan pada benda *circular* yang diam. Pada proses ini sering kali ditemukan cacat las karena adanya kesulitan pengelasan pada bidang yang melingkar. Sebaliknya, cara kedua sangat efektif untuk mengatasi kesulitan pengelasan pada bidang melingkar ini, namun cara kedua ini membutuhkan alat bantu pemutar benda kerja yang dapat diatur kecepatannya.

Pengelasan dengan las *Gas Metal Arc Welding* (GMAW) sering dipakai pada jenis pengelasan melingkar. Hal ini disebabkan karena pada jenis pengelasan GMAW, pelindung lasan berupa gas sehingga pasca pengelasan tidak perlu adanya proses pembersihan terak dan hasil pengelasan tidak terdapat retakan [1]. Untuk menjaga kualitas hasil lasan, pengaturan urutan proses, pengatur kecepatan, pengatur *welding holder* dan ketepatan penghentian proses pengelasan sangat diperlukan.

Untuk mendesain suatu proses manufaktur agar beroperasi dengan baik maka perlu dipahami beberapa kaidah pengaturan. Beberapa jenis pengaturan dapat dilakukan dengan berbagai cara dan alat yang tersedia di lapangan antara lain mikro komputer, mikro kontroler, PLC, arduino dan lain-lain. Para peneliti sebelumnya yang telah melakukan penelitian pengaturan urutan proses manufaktur antara lain Silvia[2], Kushartanto dkk [3] Huda dkk [4] dan Kabib[5]. Pada penelitian ini, pengaturan dengan arduino dipakai untuk mengatur urutan proses, penekanan *holder* las, dan akurasi penghentian proses pengelasan sehingga kualitas pengelasan dapat dijaga.

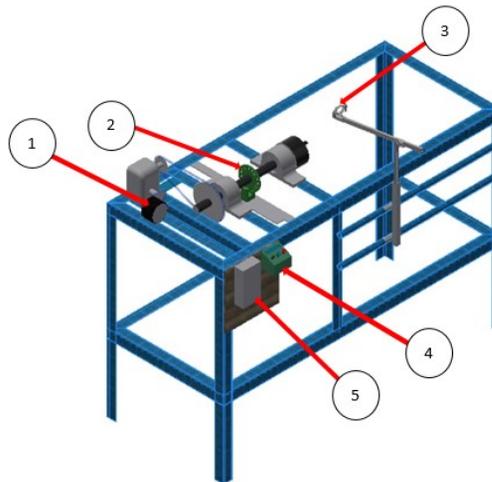
Penelitian ini bertujuan melakukan rancang bangun sistem kontrol urutan proses, kecepatan dan ketepatan penghentian proses pengelasan pada proses pengelasan pipa. Peralatan yang di rancang dan di buat sangat membantu dalam proses pengelasan pipa untuk mendapatkan kualitas pengelasan yang baik.

2. METODOLOGI PENELITIAN

Metodologi penelitian ini dimulai dengan kajian pustaka, analisa kebutuhan sistem kontrol, desain sistem kontrol dan pembuatan dan pengujian alat. Desain sistem kontrol meliputi penyusunan diagram block sistem, perancangan instalasi LCD, perancangan instalasi motor, perancangan instalasi servo dan perancangan instalasi sensor. Konsep desain dalam merancang sistem kontrol alat bantu pengelasan pipa ini meliputi beberapa mekanisme gerakan. Pengaturan putaran kecepatan motor bertujuan untuk mendapatkan gerakan berputar benda kerja sesuai kecepatan pengelasan yang diinginkan. Rangkaian *rotary encoder* dan tampilan LCD dipakai untuk pembacaan kecepatan putar dan ketepatan penghentian proses pengelasan secara otomatis. Sedangkan motor servo dipakai untuk menekan *holder* pengelasan. Rancangan desain alat bantu pengelasan dapat dilihat pada Gambar 1. Pada Gambar 1 terlihat bahwa, peralatan yang didesain terdiri dari: 1). Motor DC, 2). Rangkaian *rotary encoder*, 3). Motor servo, 4). *Panel box*, dan 5). *Power supply*.

Prinsip kerja dari komponen-komponen tersebut adalah sebagai berikut :

- a. Motor DC dipakai sebagai penggerak utama mesin, dan dihubungkan dengan gerbox dengan ratio 60:1. Putaran poros keluaran gearbox selanjutnya ditransmisikan dengan menggunakan gear dan rantai ke poros pengecam benda kerja dengan ratio 4:1.
- b. Rangkaian *rotary encoder* yang terdiri dari piringan berlubang, sensor cahaya dan lampu LED dipakai sebagai sensor pembacaan kecepatan pengelasan dan sistem berhenti otomatis pada mesin.
- c. Motor servo dipakai untuk menekan *holder* las secara otomatis.
- d. *Panel box* berisi beberapa komponen seperti: PWM *speed control (dimmer)*, LCD dan tombol tekan. PWM *speed control (dimmer)* berfungsi untuk mengatur kecepatan motor, sedangkan LCD berfungsi untuk menampilkan mode-mode yang telah dibuat.
- e. *Power supply* berfungsi sebagai tegangan utama untuk mengoperasikan mesin input *power supply* dengan tegangan 220 AC dan outputnya 12 V DC.

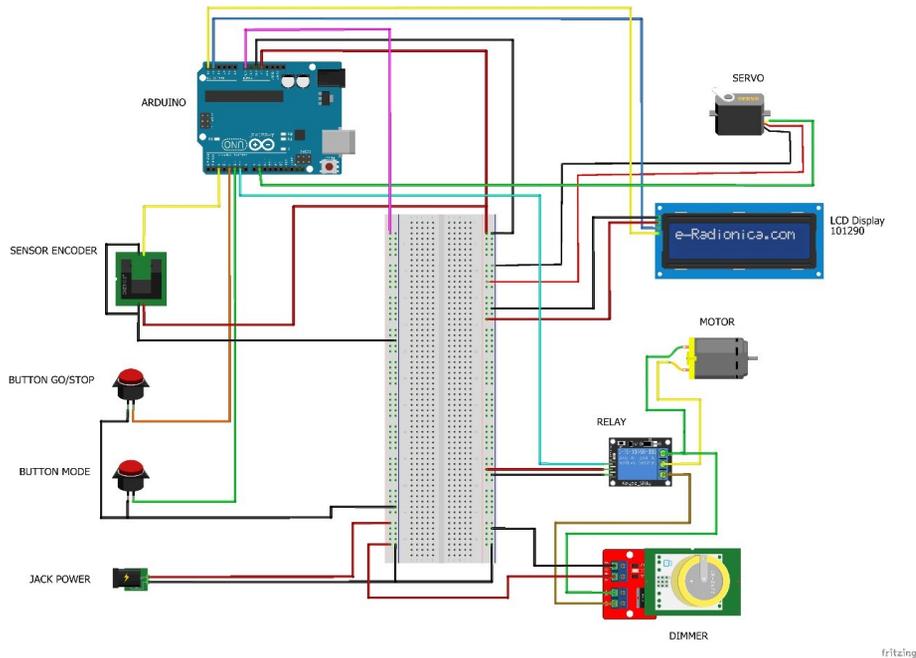


Gambar 1. Konsep desain sistem kontrol

3. HASIL DAN PEMBAHASAN

3.1 Rangkaian *Hardware* Sistem Kontrol

Hasil rancangan rangkaian *hardware* sistem kontrol dapat dilihat pada Gambar 2. Rangkaian *hardware* sistem kontrol berupa *layout* komponen dan diagram pengkabelan rangkaian kontrol. Pembuatan *hardware* pada sistem kontrol alat bantu pengelasan pipa ini menggunakan arduino sebagai kontroler, rangkaian *rotary encoder*, LCD, *power supply*, *relay*, *dimmer*, motor DC dan motor servo.



Gambar 2. Rangkaian *Hardware* Sistem Kontrol

Penjelasan dari rangkaian yang dapat dilihat pada Gambar 2 adalah sebagai berikut:

- Arduino uno berfungsi sebagai komponen kontroler pengolah data dari komponen komponen lainnya.
- Jack power (power supply)* sebagai sumber tegangan utama sebesar 12 vdc.
- Dimmer berfungsi untuk mengatur kecepatan putar motor dc, *dimmer* terhubung ke relay dan motor dc.
- Relay berfungsi sebagai saklar elektronik untuk menghentikan arus ke motor dc untuk sistem berhenti otomatis, relay terhubung ke pin 6 digital Arduino serta VCC relay ke pin 5v *Arduino* dan GND relay ke GND Arduino untuk outputnya menggunakan tegangan 12 vdc dari *power supply* pin NC dan pin GND pada relay dihubungkan ke motor DC.
- Motor DC berfungsi sebagai penggerak utama dari mesin.
- Motor servo berfungsi untuk menekan *holder* las secara otomatis, pin GND motor servo terhubung ke pin GND Arduino, pin vcc motor servo terhubung ke pin 5v *Arduino*, dan pin sinyal motor servo terhubung ke pin digital 9 *Arduino*.
- Sensor *encoder* berfungsi untuk membaca jumlah lubang pada piringan yang diletakkan pada poros mesin yang berfungsi untuk menghentikan mesin secara otomatis, pin vcc sensor terhubung ke 5v *Arduino*, pin GND sensor terhubung ke pin GND *Arduino*, dan pin sinyal sensor terhubung ke pin digital 2 *Arduino*.
- Button mode terdapat beberapa mode yang bisa dijalankan melalui button ini yaitu mode *setting speed* berfungsi mengatur kecepatan motor DC, mode *seting count* untuk menentukan jumlah lubang pada piringan yang akan digunakan dalam pengelasan, *seting* diameter pipa untuk memasukan diameter pipa yang akan di las.
- Button GO* berfungsi untuk melakukan proses pengelasan ketika button go ditekan motor servo akan menekan holder las selang 2 detik motor dc hidup memutar pengecam dan menggerakkan pipa setelah pengelasan selesai mesin akan berhenti secara otomatis dan motor servo melepaskan tang las sehingga proses pengelasan berhenti.
- LCD berfungsi untuk menampilkan berbagai mode yang telah di atur dalam Arduino, pin vcc LCD dihubungkan dengan pin 5V *Arduino* , pin GND LCD dihubungkan ke pin GND

Arduino, pin sda LCD dihubungkan ke pin sda *Arduino*, pin scl LCD dihubungkan ke pin scl *Arduino*.

3.1.1 Proses perakitan *hardware* ke mesin

Dalam merakit *hardware* sistem kontrol akan disesuaikan dengan desain instalasi *hardware* yang dapat dilihat pada Gambar 2 adapun langkah-langkahnya sebagai berikut:

- a. Memasang sensor *rotary encoder* ke poros mesin
- b. Memasang motor servo di pemegang holder.
- c. Memasang motor DC ke rangka mesin.
- d. Memasang kabel penghubung motor ke relay dan *dimmer*.
- e. Memasang kabel sesuai dengan instalasi *hardware* yang ada pada gambar 2.
- f. Memasang kabel penghubung dari motor servo, sensor *rotary encoder* dan relay ke *Arduino*.
- g. Memasang kabel *power supply* ke relay dan *dimmer*.
- h. Menghubungkan relay ke *Arduino*.
- i. Setelah instalasi *hardware* terpasang selanjutnya mengupload program ke dalam *Arduino* untuk mengontrol semua komponen.

3.2 Perancangan dan pembuatan Software Sistem Kontrol

Proses pembuatan program sistem kontrol menggunakan *software Arduino IDE*, *Arduino* menggunakan pemrograman dengan Bahasa C, setiap program *Arduino* (biasa disebut sketch) mempunyai dua buah fungsi yang harus ada dalam setiap program yaitu:

- a. *Void setup () {}*

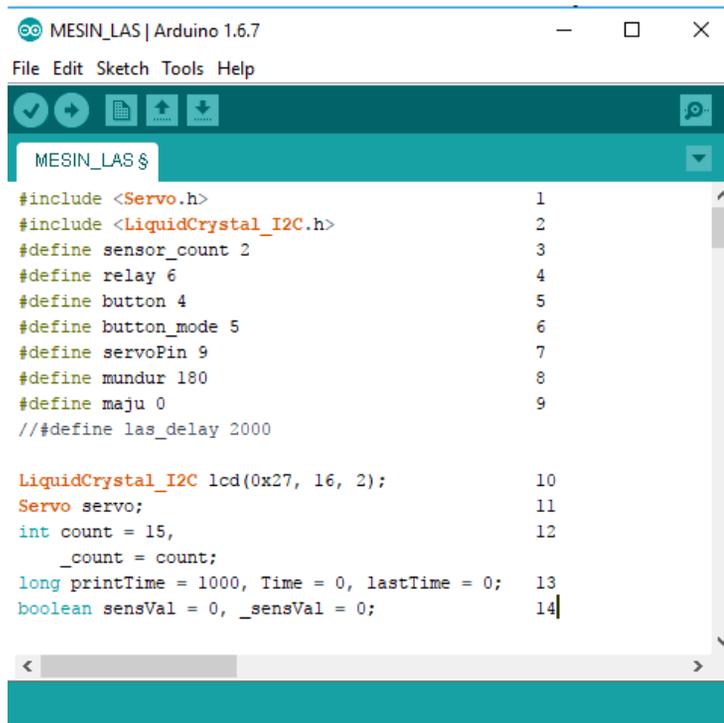
Semua kode dalam kurung kurawal hanya akan dijalankan selama satu kali ketika program *Arduino* dijalankan untuk pertama kalinya.

- b. *Void loop () {}* Fungsi ini akan dijalankan setelah fungsi *void setup* selesai, setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampe sumber tegangan di matikan.

Dalam pembuatan *software* sistem kontrol terdapat beberapa mode untuk membantu proses pengelasan untuk berganti mode menggunakan button terdapat dua buton yaitu *button go/stop* dan *button mode* adapun mode yang ada dalam pemrograman yaitu:

- a. Mode *setting speed* dalam mode ini berfungsi untuk mengatur kecepatan putar motor dengan cara menekan button dan mengarahkannya ke mode *setting speed* lalu menekan tombol go, setelah itu putar *dimmer* untuk mengatur putaran sesuai kebutuhan pengelasan.
- b. Mode *setting count* mode ini berfungsi untuk menentukan jumlah lubang yang akan dibaca oleh sensor untuk proses berhenti otomatis, untuk *seting count* tekan *button mode* pilih *setting count* untuk menambah dan mengurangi jumlah *count* gunakan *button go* seting sesuai kebutuhan pengelasan.
- c. Mode *setting pipa* mode ini untuk memasukkan diameter pipa yang akan di las, dengan cara menekan tombol mode pilih *setting pipa* lalu tekan button go untuk memilih diameter pipa sesuai dengan pipa yang akan di las.
- d. Mode *ready mode* ini untuk proses pengelasan setelah semua mode kita *setting* sesuai kebutuhan untuk menjalankan mode *ready* tekan button mode pilih mode *ready* lalu tekan button go maka servo akan menekan holder lalu motor akan berputar menggerakkan pipa setelah pengelasan selesai servo akan melepas *holder* dan motor akan berhenti berputar.

Adapun detail penulisan program sistem kontrol menggunakan *software Arduino IDE* dan penjelasan tiap baris adalah sebagaimana di tunjukkan pada gambar 2 sampai 9 berikut :



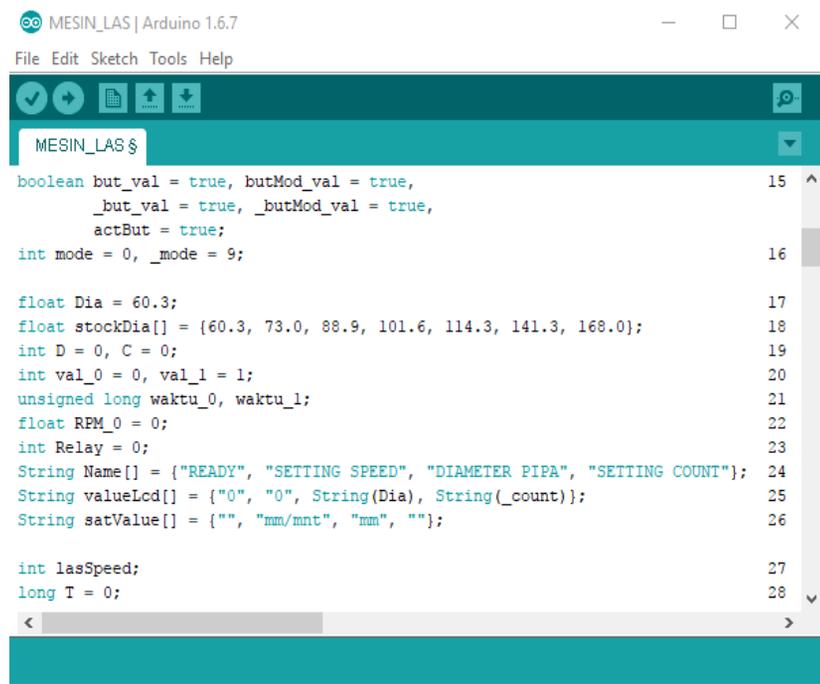
```
MESIN_LAS $
#include <Servo.h> 1
#include <LiquidCrystal_I2C.h> 2
#define sensor_count 2 3
#define relay 6 4
#define button 4 5
#define button_mode 5 6
#define servoPin 9 7
#define mundur 180 8
#define maju 0 9
//#define las_delay 2000

LiquidCrystal_I2C lcd(0x27, 16, 2); 10
Servo servo; 11
int count = 15, 12
    _count = count;
long printTime = 1000, Time = 0, lastTime = 0; 13
boolean sensVal = 0, _sensVal = 0; 14
```

Gambar 3. Deklarasi program

Keterangan Gambar 3:

1. Memasukkan *library* servo.
2. Memasukkan library lcd dengan i2c.
3. Deklarasi sensor ke pin 2.
4. Deklarasi relay ke pin 6.
5. Deklarasi *button* ke pin 4 (untuk *button go/stop*).
6. Deklarasi *button* mode ke pin 5 (untuk *button mode*).
7. Deklarasi servo pin 9.
8. Variabel untuk derajat servo untuk mundur.
9. Variabel untuk derajat servo untuk maju.
10. Membuat variabel lcd(alamat i2c, jumlah baris lcd, jumlah kolom lcd).
11. Membuat variabel servo.
12. Deklarasi variabel count dan *_count* type integreter.
13. Deklarasi variabel time untuk waktu update lcd type data long.
14. Deklarsi variabel untuk sensor count type data Boolean.



```
MESIN_LAS $
boolean but_val = true, butMod_val = true,
    _but_val = true, _butMod_val = true,
    actBut = true;
int mode = 0, _mode = 9;

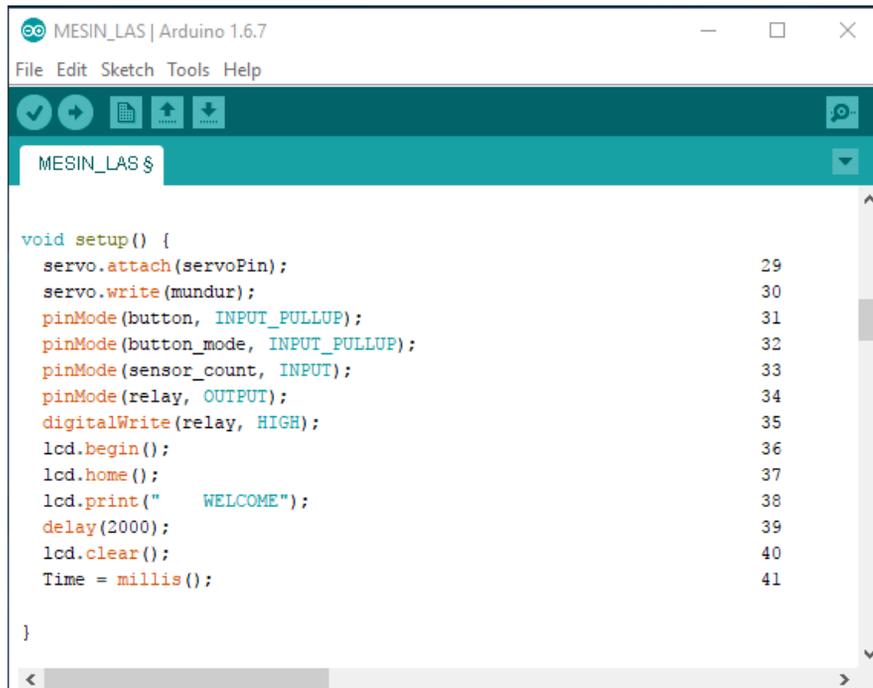
float Dia = 60.3;
float stockDia[] = {60.3, 73.0, 88.9, 101.6, 114.3, 141.3, 168.0};
int D = 0, C = 0;
int val_0 = 0, val_1 = 1;
unsigned long waktu_0, waktu_1;
float RPM_0 = 0;
int Relay = 0;
String Name[] = {"READY", "SETTING SPEED", "DIAMETER PIPA", "SETTING COUNT"};
String valueLcd[] = {"0", "0", String(Dia), String(_count)};
String satValue[] = {"", "mm/mnt", "mm", ""};

int lasSpeed;
long T = 0;
```

Gambar 4. Deklarasi Mode pengelasan

Keterangan Gambar 4:

15. Deklarasi variabel untuk button.
16. Mode 0: ready.
17. Deklarasi variabel diameter pipa.
18. Daftar stok diameter pipa.
19. Variabel untuk nomor diameter pipa dan nomor count.
20. Variabel sensor speed.
21. Variabel untuk waktu *speed*.
22. Deklarasi rpm.
23. Deklarasi untuk nilai relai 0 = nyala, 1 = mati.
24. Daftar variabel nama mode.
25. Daftar nilai mode.
26. Daftar satuan mode.
27. Variabel *speed* pengelasan.
28. Deklarasi T untuk selang waktu.



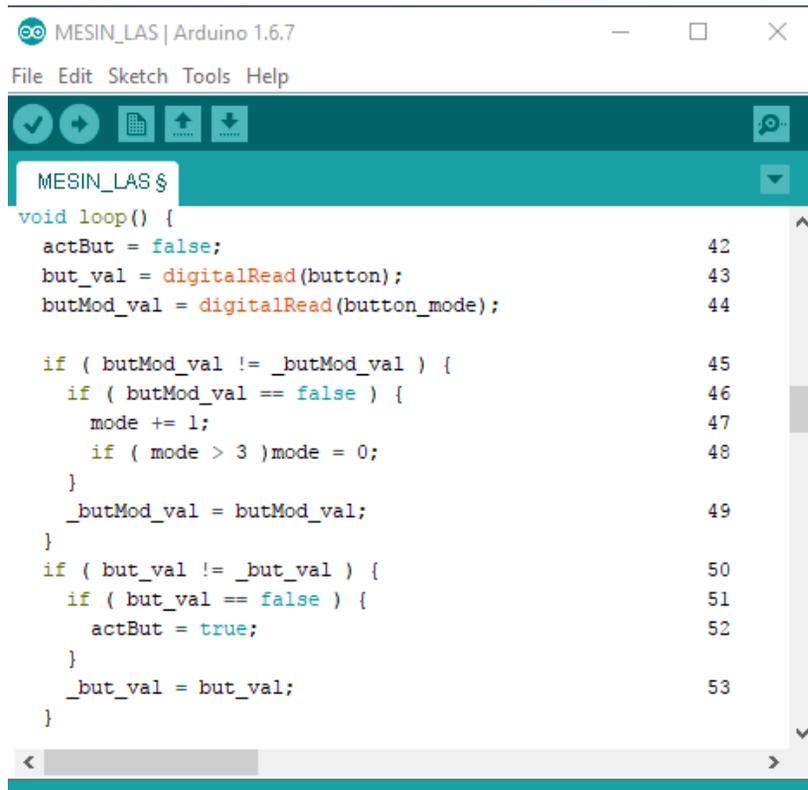
```
MESIN_LAS | Arduino 1.6.7
File Edit Sketch Tools Help
MESIN_LAS $

void setup() {
  servo.attach(servoPin);           29
  servo.write(mundur);              30
  pinMode(button, INPUT_PULLUP);    31
  pinMode(button_mode, INPUT_PULLUP); 32
  pinMode(sensor_count, INPUT);      33
  pinMode(relay, OUTPUT);           34
  digitalWrite(relay, HIGH);        35
  lcd.begin();                      36
  lcd.home();                       37
  lcd.print("  WELCOME");           38
  delay(2000);                      39
  lcd.clear();                      40
  Time = millis();                  41
}
```

Gambar 5. program void setup

Keterangan Gambar 5:

29. Mengatur pin servo ke *variable servo Pin*.
30. Memundurkan servo agar las off pada awal dihidupkan.
31. Mengatur pin input.
32. Mengatur pin input.
33. Mengatur pin input untuk sensor.
34. Mengatur pin output untuk *relay*.
35. Mengatur relay HIGH(mati) untuk pertama kali dihidupkan.
36. Mulai mengaktifkan lcd.
37. Set cursor lcd ke *home* (baris 0, kolom 0) mulai tulis dari posisi ini.
38. Menampilkan teks *WELCOME* di tengah.
39. Tunggu 2 detik.
40. Menghapus karakter di lcd
41. Menyamakan *variabel Time = millis()* (waktu berjalan arduino).



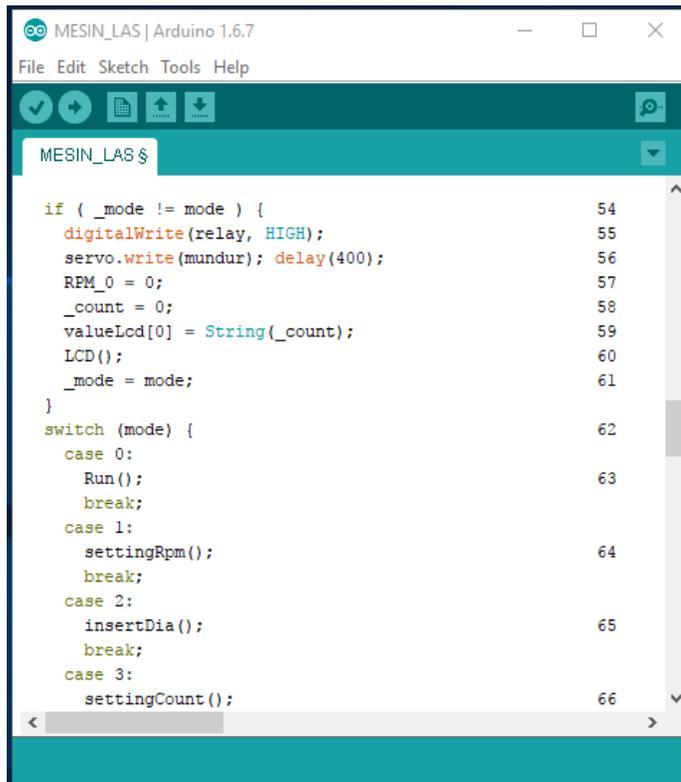
```
MESIN_LAS $
void loop() {
  actBut = false;           42
  but_val = digitalRead(button); 43
  butMod_val = digitalRead(button_mode); 44

  if ( butMod_val != _butMod_val ) { 45
    if ( butMod_val == false ) { 46
      mode += 1; 47
      if ( mode > 3 ) mode = 0; 48
    }
    _butMod_val = butMod_val; 49
  }
  if ( but_val != _but_val ) { 50
    if ( but_val == false ) { 51
      actBut = true; 52
    }
    _but_val = but_val; 53
  }
}
```

Gambar 6. program void loop

Keterangan Gambar 6:

42. Mengecek jika mode di ubah > mode tadi tidak sama dengan mode sekarang.
43. Mengubah nilai but_val = hasil baca sensor *buton go/stop*.
44. Mengubah nilai but_val = hasil baca sensor *buton mode*.
45. Membandingkan hasil baca sensor button mode yang sekarang dengan yang sebelum nya.
46. Jika *button* ditekan atau di lepas.
47. Pindah mode.
48. Mengembalikan ke mode 0 jika mode = 4 (mode 4 tidak ada).
49. Menyamakan value button mode tadi dengan sekarang.
50. Membandingkan hasil baca sensor button go/sto yang sekarang dengan yang sebelum nya.
51. Jika button ditekan atau di lepas.
52. Mengubah *value actBut* = true > tanda jika tombol go/stop ditekan.
53. Menyamakan *value button go/stop* tadi dengan sekarang.



```
MESIN_LAS | Arduino 1.6.7
File Edit Sketch Tools Help
MESIN_LAS $
54  if ( _mode != mode ) {
55      digitalWrite(relay, HIGH);
56      servo.write(mundur); delay(400);
57      RPM_0 = 0;
58      _count = 0;
59      valueLcd[0] = String(_count);
60      LCD();
61      _mode = mode;
62  }
63  switch (mode) {
64      case 0:
65          Run();
66          break;
67      case 1:
68          settingRpm();
69          break;
70      case 2:
71          insertDia();
72          break;
73      case 3:
74          settingCount();
```

Gambar 7. Pengecekan mode program

Keterangan Gambar 7 :

54. Mengecek jika mode di ubah > mode tadi tidak sama dengan mode sekarang.
55. Untuk menghindari eror karena mode di ubah ketika mesin jalan maka: mematikan relay motor.
56. Mematikan las.
57. Mengeset rpm ke 0.
58. Mengembalikan count yang berjalan ke 0.
59. Update data valueLcd ke 0 = _count
60. Mengupdate lcd > note: progam lompat ke void LCD.
61. Mengubah nilai mode tadi sama denga ke mode sekarang.
62. Mengecek mode yg sedang di pakai.
63. Apabila mode = 0 > maka progam pergi ke void RUN.
64. Apabila mode = 1 > maka progam pergi ke void settingRpm.
65. Apabila mode = 2 > maka progam pergi ke void insertDia.
66. Apabila mode = 3 > maka progam pergi ke void settingCount.



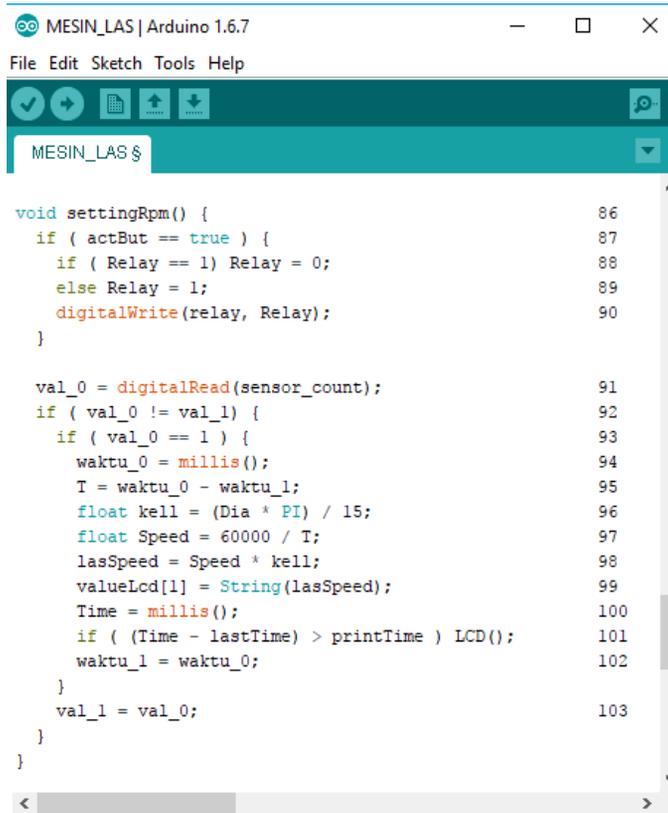
```
void Run() {
  if ( actBut == true ) {
    _count = count;
    valueLcd[mode] = String(_count);
    LCD();
    servo.write(maju);
    delay(300);
    digitalWrite( relay, LOW);
  }
  if ( _count < 1 ) {
    digitalWrite( relay, HIGH);
    servo.write(mundur);
    delay(300);
    return;
  }

  sensVal = digitalRead(sensor_count);
  if ( sensVal != _sensVal ) {
    if ( sensVal == true ) _count -- 1 ;
    valueLcd[mode] = String(_count);
    LCD();
  }
  _sensVal = sensVal;
}
```

Gambar 8. Program void run

Keterangan Gambar 8:

67. Blok kode *ready*.
68. Jika tombol go/stop di tekan saat mode ready.
69. Menyamakan nilai *_count* dengan *count* yg di set.
70. Update *valueLcd* ke 0 sama dengan *count* > artinya start putaran.
71. Update LCD.
72. Menggerakkan servo untuk maju.
73. Menunggu servo sampai di tujuan.
74. Menghidpkan relay motor.
75. Cek jika *count* tercapai atau sudah 1 putaran as.
76. Mematikan relay motor.
77. Mundurkan servo.
78. Menunggu servo sampai di tujuan.
79. Progam kembal ke *void loop*.
80. Membaca sensor dan menaruh nilai di *variabe sensVal*.
81. Cek hasil baca sensor tida sama dengan sebelumnya *true>>false false>>true*.
82. Jika sensor *true* atau *detected* maka *count* di kurang 1.
83. Update *valueLcd* ke 0 dengan *count* (untuk print lcd).
84. Update lcd.
85. Mengubah *value sensor* tadi sama dengan sekarang.



```
MESIN_LAS | Arduino 1.6.7
File Edit Sketch Tools Help
MESIN_LAS $

void settingRpm() {                                86
  if ( actBut == true ) {                          87
    if ( Relay == 1) Relay = 0;                    88
    else Relay = 1;                                89
    digitalWrite(relay, Relay);                    90
  }

  val_0 = digitalRead(sensor_count);               91
  if ( val_0 != val_1) {                           92
    if ( val_0 == 1 ) {                             93
      waktu_0 = millis();                           94
      T = waktu_0 - waktu_1;                         95
      float kell = (Dia * PI) / 15;                 96
      float Speed = 60000 / T;                      97
      lasSpeed = Speed * kell;                      98
      valueLcd[1] = String(lasSpeed);               99
      Time = millis();                              100
      if ( (Time - lastTime) > printTime ) LCD();  101
      waktu_1 = waktu_0;                            102
    }
    val_1 = val_0;                                  103
  }
}
```

Gambar 9. Program setting rpm

Keterangan Gambar 9:

86. Blok kode setting rpm.
87. Cek jika tombol go/stop di tekan.
88. Jika value Relay sebelumnya 0 maka valuenya di ubah menjadi 1
89. Jika tidak value di ubah ke 0
90. Mematikan atau menhidupkan relay sesuai value Relay jika sebelumnya hidup maka di matikan.
91. Membaca sensor dan menaru value di val_0.
92. Jika hasil sensor tidak sama dengan sebelumnya; note: perubahan detected dan tidak detected.
93. Jika sensor detected.
94. Menyamakan value variabel waktu_0 sama dengan waktu millis Arduino.
95. Menghitung jarak waktu detected sensor tadi dengan sekarang.
96. Menghitung jarak tempuh sisi pipa dalam 1/15 putaran.
97. Menghitung RPM poros.
98. Menghitung kecepatan tempuh sisi pipa mm/menit.
99. Update valueLcd ke 1 dengan hasil speed las.
100. Menyamakan variabel Time dengan waktu milis Arduino.
101. Cek jika sudah waktu update lcd;
102. Menyamakan waktu pembacaan sensor tadi dan sekarang, untuk digunakan pembacaan waktu selanjutnya.
103. Menyamakan value sensor tadi dengan sekarang.

3.3 Hasil Pengujian

Pengujian dilakukan bertujuan untuk membandingkan pembacaan kecepatan pengelasan (Kecepatan putar benda kerja) dengan *tachometer* digital dan perhitungan secara manual. Pengujian dilakukan dengan variasi tiga kecepatan yang berbeda dan tiap kecepatan di uji sebanyak tiga kali.

3.3.1 Tabel Perbandingan Kecepatan pada LCD dan *Tachometer*

Perbandingan kecepatan pada pembacaan LCD dan *tachometer* (kontak) dapat dilihat dalam pada Tabel 1. Persentase kesalahan pembacaan kecepatan translasi pengelasan berkisar antara 1,58 s/d 1,7%.

Tabel 1. perbandingan speed LCD dengan *Tacho meter* (Kontak)

NO	Kecepatan Pembacaan LCD (mm/min)	Kecepatan <i>tachometer</i> (mm/menit)				Rata-rata	Penyimpangan	Persentase kesalahan
		Pengujian pertama	Pengujian kedua	Pengujian ketiga				
1	860	840	850	850	846,6	13,4	1,58 %	
2	620	610	600	620	610	10	1,6 %	
3	400	400	390	390	393,3	6,7	1,7 %	

3.3.2 Tabel Perbandingan Speed LCD Dengan Perhitungan Manual

Perbandingan pembacaan kecepatan LCD dengan perhitungan manual dapat dilihat dalam Table 2. Persentase kesalahan pembacaan kecepatan translasi pengelasan dengan perhitungan manual berkisar antara 1,75 s/d 2%

Tabel 2. Perbandingan speed LCD dengan Perhitungan manual

NO	Kecepatan Pembacaan LCD (mm/min)	Perhitungan manual (mm/menit)				Rata-rata	Penyimpangan	Persentase kesalahan
		Pengujian pertama	Pengujian kedua	Pengujian ketiga				
1.	860	843,6	847,4	844,8	845,2	14,8	1,75 %	
2.	620	613	606	607,7	608,9	11,1	1,8 %	
3.	400	395	390	391	392	8	2 %	

4. KESIMPULAN

Sistem pengaturan urutan proses, kecepatan pengelasan dan ketepatan penghentian pada alat bantu pengelasan pipa telah berhasil dirancang, dibuat dan diuji. Beberapa parameter proses manufaktur ditampilkan dalam LCD. Kecepatan pengelasan benda kerja pada tampilan LCD dibandingkan dengan pengujian dengan *tacho meter* mempunyai persentase kesalahan berturut-turut berkisar 1,5 ~ 1,7% dan 1,75 ~ 2% pada range pengujian antara 860 ~ 400 rpm.

5. DAFTAR PUSTAKA

- [1] Sopian & F.B. Susetyo, 2017, " Pengaruh Besar Sudut Kampuh Terhadap Kekuatan Tarik Hasil Pengelasan GMAW', *Jurnal Kajian Teknik Mesin*, vol. 2, no. 2, pp. 99-105.
- [2] Silvia, 2014, "Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino Dan Android", *Electrans*, 2014, vol. 13, no. 1, pp. 1–10.
- [3] P. Kushartanto, M. Kabib dan R.Winarso, 2019, "Sistem Kontrol Gerak Dan Perhitungan Produk Pada Mesin Pres dan Pemotong Kantong Plastik", *Jurnal CRANKSHAFT*, Vol. 2 No.1,
- [4] S. Huda, M. Kabib dan R.Winarso, 2017," *Desain Automatic Line Plastic Packing of Cake Berbasis Mikrokontroler Atmega 328*", *Prosiding SNATIF Ke -4*.
- [5] M. Kabib, 2008," *Analisa dan Pengembangan Sistem Kontrol pada Mesin Automatic Feeding untuk Proses U Bending*", Thesis, Program studi Teknik Mesin, Institut Teknologi Sepuluh Nopember.