BAB IV
PEMBUATAN ALAT

Proses pembuatan alat *Coil winder* otomatis meliputi tahap realisasi perangkat keras dan perangkat lunak yang digunakan untuk mendukung sistem kerja alat secara keseluruhan. Tahapan ini dilakukan untuk memastikan seluruh komponen dapat bekerja sesuai dengan fungsi dan rancangan yang telah ditentukan.

Pembuatan alat dilakukan melalui beberapa tahapan, dimulai dari persiapan alat dan bahan, pembuatan desain mekanik, pencetakan komponen menggunakan printer 3D, perakitan sistem mekanik, pemasangan rangkaian elektronik, hingga proses pemrograman mikrokontroler. Seluruh tahapan dilakukan secara bertahap agar proses integrasi antar komponen dapat berjalan dengan baik dan menghasilkan sistem yang stabil.

Selain proses pembuatan, pada bab ini juga dijelaskan proses pengujian dasar terhadap sistem yang telah dirakit. Pengujian dilakukan untuk memastikan fungsi setiap bagian alat, seperti sistem penggerak motor, pengaturan jumlah lilitan, pengaturan diameter kawat, tampilan LCD, serta sistem pembatas gerakan menggunakan *limit switch* dapat bekerja sesuai dengan perancangan. Dengan dilakukannya tahapan pembuatan dan pengujian ini, diharapkan alat *Coil winder* otomatis dapat bekerja dengan baik dan mampu membantu proses pelilitan kawat secara lebih efektif dan presisi.

4.1 Alat dan Bahan

Pada tahap pembuatan alat *Coil winder* ini, dilakukan proses realisasi dari hasil perancangan yang telah dijelaskan pada Bab III. Proses ini melibatkan penggunaan berbagai alat kerja serta bahan atau komponen yang dibutuhkan untuk membangun sistem secara fisik. Alat digunakan untuk mendukung proses pengerjaan seperti penyolderan, perakitan, dan pengujian rangkaian, sedangkan bahan merupakan komponen utama yang dirakit menjadi satu kesatuan sistem sesuai dengan desain yang telah dirancang sebelumnya.

Pemilihan alat dan bahan disesuaikan dengan kebutuhan selama proses pembuatan, mulai dari tahap perakitan mekanik, pemasangan rangkaian elektronik, hingga pengujian sistem. Seluruh alat dan bahan yang digunakan dalam proses ini disajikan dalam bentuk tabel untuk memudahkan identifikasi serta memberikan gambaran umum mengenai komponen yang digunakan dalam pembuatan alat *Coil winder*.

Daftar komponen perangkat keras dan perangkat lunak yang digunakan dalam pengembangan sistem ini dapat dilihat pada Tabel 4.1 berikut:

Tabel 4- 1 Alat dan Bahan

Nama	Satuan	Jumlah
Arduino Nano	Unit	1
Motor Driver a4988	Unit	2
Motor Stepper Nema 17	Unit	2
LM2596 Buck Converter	Unit	1
Adaptor 12V 3A	Unit	1
Fan 25mm x 25mm	Unit	1
LCD I2C 16X2	Unit	1
***Push button* Switch**	Unit	6
Capasitor Elco 470uF	Unit	3
Power Switch	Unit	1
Limit switch	Unit	2
Jack Female DC	Unit	1
Kabel Jumper Male to Male	Unit	10
Kabel Jumper Female to Female	Unit	10
Kabel Jumper Male to Female	Unit	10
Kabel Serabut 22 AWG	m	1

PCB	cm^2	52
Filament PLA 2.0+	kg	1
Magnet Neodynium 3x2mm	Unit	8
THSL T8 Lead Screw 8mm	Cm	20
Flange Coupling 5mm	Unit	4
Linear Shaft Rod 10mm	Cm	20
Linear Shaft Rod 5mm	Cm	15
Bearing 5mm	Unit	1
Lock Collar As 5mm	Unit	1
Flexible Coupling 5mm to 8mm	Unit	1
SK10 Linear Shaft Holder	Unit	4
SCS10UU	Unit	1
KP008 Pillow Block Bearing	Unit	1
Braket L Shoket Motor Stepper Nema 17	Unit	2
T Nut T8 Leadsrew	Unit	1
Stick Glue	Unit	3
Alteco Glue	Mg	20
Solder 60W	Unit	1
Timah	m	25

4.2 Perancangan Perangkat Keras (Hardware)

Pada tahap ini dilakukan proses realisasi perangkat keras berdasarkan hasil perancangan yang telah dibuat sebelumnya. Perangkat keras pada alat *Coil winder* ini terdiri dari sistem mekanik dan sistem elektronik yang saling terhubung untuk mendukung proses pelilitan kawat secara otomatis. Proses pembuatan dilakukan mulai

dari pembuatan desain mekanik, pencetakan komponen menggunakan printer 3D, perakitan mekanik, hingga pemasangan komponen elektronik pada sistem.

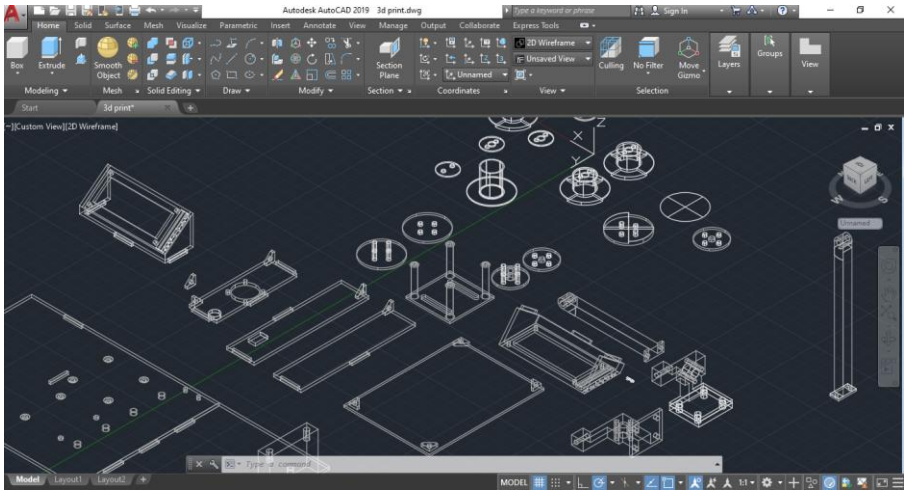
Perancangan perangkat keras dilakukan dengan mempertimbangkan kestabilan sistem, kemudahan perakitan, serta kesesuaian fungsi setiap komponen terhadap proses *winding*. Seluruh bagian dirancang agar dapat bekerja secara sinkron, mulai dari sistem pemutar koker, pengarah kawat, sistem kontrol, hingga sistem pembatas gerakan. Dengan adanya integrasi antara perangkat mekanik dan elektronik, alat diharapkan mampu melakukan proses pelilitan secara otomatis dan menghasilkan susunan lilitan yang rapi serta presisi.

4.2.1 Pembuatan 3D Desain dan Print Desain

Pada tahap ini dilakukan proses pembuatan desain mekanik alat *Coil winder* otomatis yang kemudian direalisasikan dalam bentuk fisik menggunakan teknologi pencetakan 3D. Pembuatan desain dilakukan berdasarkan hasil perancangan yang telah dijelaskan pada bab sebelumnya dengan mempertimbangkan kebutuhan sistem, seperti posisi motor, jalur pergerakan kawat, kestabilan rangka, serta kemudahan dalam proses perakitan alat.

Proses desain dilakukan menggunakan software AutoCad, di mana setiap komponen dirancang secara terpisah agar memudahkan proses modifikasi dan pencetakan. Komponen yang dibuat meliputi rangka utama, dudukan motor stepper, dudukan koker, pengarah kawat, dudukan *limit switch*, serta beberapa komponen pendukung lainnya. Desain setiap bagian disesuaikan dengan ukuran komponen asli agar seluruh bagian dapat terpasang dengan baik saat proses perakitan dilakukan.

Selain memperhatikan ukuran dan bentuk komponen, proses desain juga mempertimbangkan kekuatan struktur dan posisi antar komponen agar alat dapat bekerja secara stabil selama proses *winding* berlangsung. Penyesuaian jarak antar bagian dilakukan untuk memastikan pergerakan motor pengarah kawat dapat berjalan dengan lancar mengikuti jalur lead screw tanpa mengalami hambatan.



Gambar 4- 1 Desain 3D

Setelah proses desain selesai dilakukan, file desain kemudian diekspor ke format yang sesuai untuk proses pencetakan 3D. Proses pencetakan dilakukan menggunakan printer 3D dengan material yang memiliki kekuatan cukup untuk menopang beban motor dan komponen mekanik lainnya. Pada tahap ini dilakukan pengaturan parameter pencetakan seperti ketebalan layer, kepadatan infill, dan kecepatan pencetakan agar hasil cetakan memiliki tingkat presisi dan kekuatan yang baik. Hasil desain dapat dilihat di Gambar 4-1.

Gambar 4- 2 Hasil Sebagian 3D Print

Hasil cetakan yang telah selesai kemudian diperiksa kembali untuk memastikan tidak terdapat cacat pada permukaan maupun ketidaksesuaian ukuran. Jika ditemukan bagian yang kurang sesuai, dilakukan proses perbaikan desain atau pencetakan ulang hingga diperoleh hasil yang sesuai dengan kebutuhan sistem. Komponen yang telah selesai dicetak selanjutnya dipersiapkan untuk tahap perakitan mekanik alat. Hasil dari cetak 3D dapat dilihat di Gambar 4-2.

4.2.2 Perakitan Mekanik Alat

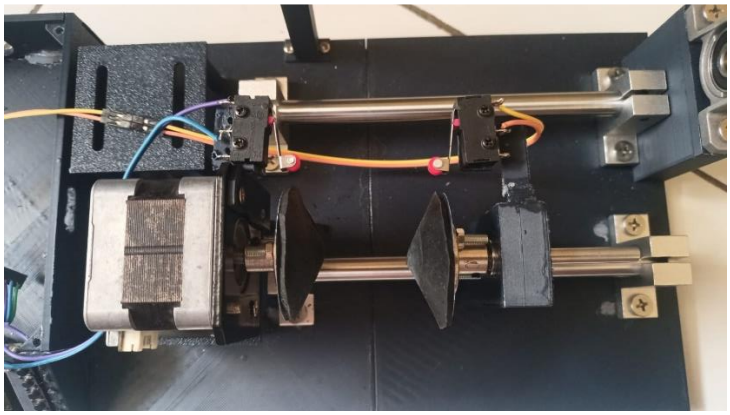
Pada tahap ini dilakukan proses perakitan seluruh komponen mekanik yang telah dicetak menggunakan printer 3D menjadi satu kesatuan sistem alat *Coil winder* otomatis. Proses perakitan dilakukan berdasarkan desain yang telah dibuat sebelumnya dengan memperhatikan posisi pemasangan setiap komponen agar sistem dapat bekerja dengan baik dan stabil selama proses *winding* berlangsung.

Perakitan diawali dengan pemasangan rangka utama yang berfungsi sebagai penopang seluruh sistem mekanik dan elektronik pada alat. Rangka dirancang untuk menjaga kestabilan posisi komponen serta memudahkan proses pemasangan bagian-bagian lainnya. Setelah rangka utama terpasang, dilakukan pemasangan dudukan motor stepper pemutar koker beserta *holder* koker yang digunakan sebagai tempat proses pelilitan kawat.

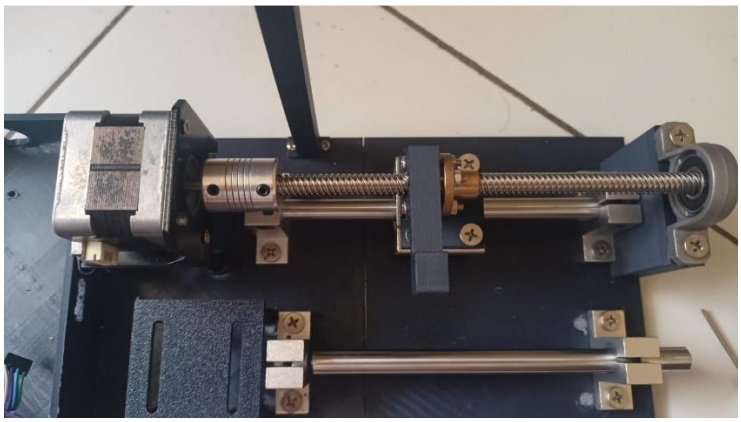
Gambar 4- 3 Gambar Kerangka 3D Print

Seperti pada Gambar 4-3, motor stepper pemutar koker dipasang pada dudukan yang telah disesuaikan dengan ukuran motor agar posisi poros motor tetap sejajar dengan holder koker. Pemasangan dilakukan dengan memperhatikan kekencangan dan kestabilan posisi motor untuk mengurangi getaran saat proses pelilitan berlangsung. *Holder* koker dipasang langsung pada poros motor sehingga putaran motor dapat diteruskan secara langsung ke bagian spool atau koker.

Selanjutnya dilakukan pemasangan sistem pengarah kawat yang terdiri dari motor stepper penggerak, *lead screw*, dan dudukan pengarah kawat. Sistem ini berfungsi untuk mengatur posisi kawat selama proses *winding* agar susunan lilitan dapat tersusun secara rapi dan merata. Dudukan pengarah kawat dirancang dapat bergerak mengikuti putaran lead screw secara horizontal ke arah kiri dan kanan.



Gambar 4- 4 Motor Pemutar Spool dan Penempatan Limit switch



Gambar 4- 5 Motor Pengatur Posisi Tembaga

Pada Gambar 4-4 dan Gambar 4-5, bagian pengarah kawat juga dipasang *limit switch* pada sisi kiri dan kanan sebagai pembatas gerakan sistem. *Limit switch* digunakan untuk mendeteksi batas maksimum pergerakan dudukan pengarah kawat sehingga sistem dapat mengubah arah gerakan secara otomatis ketika mencapai batas tertentu. Pemasangan *limit switch* dilakukan dengan menyesuaikan posisi akhir jalur pergerakan agar sistem homing dan pergantian arah dapat bekerja dengan baik.

Selain itu dilakukan penyesuaian posisi antar komponen mekanik seperti jarak antara pengarah kawat dan koker, posisi lead screw, serta kesejajaran jalur pergerakan agar tidak terjadi hambatan selama proses kerja alat berlangsung. Setelah seluruh komponen mekanik terpasang, dilakukan pengecekan terhadap kelancaran gerakan sistem secara manual untuk memastikan setiap bagian dapat bergerak dengan baik sesuai dengan fungsi yang dirancang.

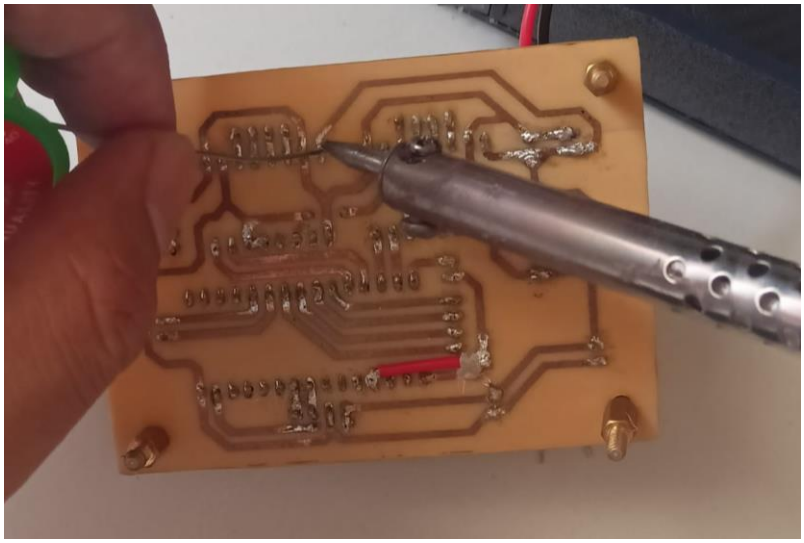
Hasil dari tahap ini berupa sistem mekanik alat *Coil winder* otomatis yang telah terpasang dan siap diintegrasikan dengan sistem elektronik serta perangkat lunak pada tahap selanjutnya.

4.2.3 Perakitan Sistem Elektronik

Pada tahap ini dilakukan proses perakitan dan pemasangan seluruh komponen elektronik yang digunakan pada alat *Coil winder* otomatis. Sistem elektronik dirancang untuk mengendalikan seluruh proses kerja alat, mulai dari pembacaan input pengguna, pengaturan parameter *winding*, pengendalian motor stepper, hingga menampilkan informasi sistem pada LCD. Seluruh komponen dihubungkan sesuai dengan rancangan sistem yang telah dibuat sebelumnya agar dapat bekerja secara terintegrasi.

Proses perakitan diawali dengan pemasangan Arduino Nano sebagai pusat pengendali utama sistem. Arduino Nano digunakan untuk menerima input dari *push button*, mengolah data parameter *winding*, serta mengendalikan driver motor stepper dan LCD. Penggunaan Arduino Nano dipilih karena memiliki ukuran yang ringkas sehingga memudahkan proses pemasangan pada PCB dan penempatan di dalam alat.

Selanjutnya dilakukan pemasangan dua buah driver motor stepper A4988 yang berfungsi sebagai pengendali motor stepper. Driver motor menerima sinyal STEP dan DIR dari Arduino Nano untuk mengatur arah putaran dan jumlah langkah motor. Driver pertama digunakan untuk mengendalikan motor pemutar koker, sedangkan driver kedua digunakan untuk mengendalikan motor pengarah kawat. Pada bagian driver juga dipasang kapasitor sebagai penstabil tegangan untuk mengurangi gangguan saat motor bekerja.



Gambar 4- 6 Pengerjaan PCB

Motor stepper kemudian dihubungkan ke driver sesuai dengan pasangan *Coil* masing-masing motor. Motor pemutar koker berfungsi untuk memutar spool atau koker selama proses *winding* berlangsung, sedangkan motor pengarah kawat digunakan untuk menggerakkan dudukan pengarah kawat secara horizontal mengikuti jalur lead screw. Kedua motor dirancang bekerja secara bersamaan agar susunan lilitan dapat tersusun secara rapi dan merata.

Selain sistem penggerak, dilakukan pemasangan *push button* yang digunakan sebagai media input pengguna. *Push button* terdiri dari tombol UP, DOWN, LEFT, RIGHT, ENTER, dan BACK yang digunakan untuk melakukan pengaturan jumlah lilitan, diameter kawat, navigasi menu, serta menjalankan proses *winding*. Seluruh *push*

button dihubungkan ke pin input Arduino Nano menggunakan konfigurasi internal pull-up agar pembacaan tombol menjadi lebih stabil. Semua komponen disatukan di PCB yang telah dibuat. Hasil PCB dapat dilihat pada Gambar 4-6.

Pada sistem tampilan digunakan LCD I2C 16x2 yang berfungsi untuk menampilkan menu pengaturan, jumlah lilitan, diameter *wire*, proses *winding*, serta status sistem lainnya. Penggunaan modul I2C bertujuan untuk mengurangi penggunaan pin pada Arduino Nano sehingga proses wiring menjadi lebih sederhana dan rapi. Hasil tampilan LCD dapat dilihat pada Gambar 4-7 berikut.

(a) (b)

Gambar 4- 7 Tampilan Antarmuka Pengguna (a) Setting Jumlah Lilitan, (b) Setting Diameter Kawat Tembaga

Untuk mendukung sistem pembatas gerakan, dipasang dua buah *limit switch* pada sisi kiri dan kanan jalur pergerakan pengarah kawat. *Limit switch* digunakan untuk mendeteksi batas gerakan maksimum sistem sekaligus mendukung proses homing otomatis saat alat dinyalakan maupun saat proses *winding* selesai.

Sumber daya utama sistem menggunakan adaptor 12V yang digunakan untuk mensuplai driver motor dan motor stepper. Selain itu digunakan buck converter untuk menurunkan tegangan menjadi 5V yang digunakan oleh Arduino Nano dan LCD. Seluruh ground pada sistem disatukan agar komunikasi antar komponen dapat berjalan dengan baik dan stabil.

Setelah seluruh komponen elektronik terpasang, dilakukan pengecekan terhadap jalur koneksi dan polaritas tegangan untuk memastikan seluruh sistem telah terhubung dengan benar sebelum proses integrasi perangkat lunak dilakukan.

4.3 Perancangan Perangkat Lunak

Pada tahap ini dilakukan proses perancangan perangkat lunak yang digunakan untuk mengendalikan seluruh sistem kerja pada alat *Coil winder* otomatis. Perangkat lunak berfungsi sebagai penghubung antara perangkat keras dengan sistem kontrol sehingga setiap komponen dapat bekerja sesuai dengan fungsi yang telah dirancang. Program dibuat menggunakan bahasa pemrograman Arduino (C/C++) dan diunggah ke Arduino Nano sebagai pusat pengendali sistem.

Perancangan perangkat lunak dilakukan dengan menyusun logika program agar mampu mengatur proses kerja alat secara otomatis mulai dari pembacaan input pengguna, pengolahan data, hingga pengendalian komponen *output*. Program dirancang untuk mengatur komunikasi antara Arduino Nano dengan berbagai komponen seperti LCD I2C, *push button*, motor stepper, driver motor A4988, dan *limit switch*.

Pada alat ini perangkat lunak memiliki beberapa fungsi utama, yaitu melakukan proses inisialisasi sistem, membaca parameter yang dimasukkan pengguna, mengendalikan pergerakan motor stepper, menjalankan proses homing, serta mengatur proses *winding* berdasarkan jumlah lilitan dan diameter kawat yang telah ditentukan. Selain itu, program juga dirancang agar mampu menampilkan informasi sistem secara real-time melalui LCD sehingga pengguna dapat memantau proses kerja alat dengan lebih mudah.

Subbab ini akan menjelaskan tahapan persiapan pemrograman, proses pembuatan program Arduino, cara kerja program pada setiap komponen yang digunakan, serta implementasi perangkat lunak terhadap sistem alat *Coil winder* otomatis secara keseluruhan.

4.3.1 Pembuatan Program Arduino

Pada tahap ini dilakukan proses pembuatan program yang digunakan untuk mengendalikan seluruh sistem pada alat *Coil winder* otomatis. Program dibuat menggunakan Arduino IDE dengan bahasa pemrograman Arduino (C/C++) dan diunggah ke Arduino Nano sebagai mikrokontroler utama sistem. Pembuatan program dilakukan berdasarkan alur kerja yang telah dirancang sebelumnya agar seluruh komponen dapat bekerja secara terintegrasi dan sesuai dengan fungsi yang diinginkan.

Arduino Nano berfungsi sebagai pusat pengendali sistem yang bertugas membaca *input* dari pengguna, mengolah data yang diterima, kemudian memberikan perintah kepada komponen *output*. *Input* pada sistem berasal dari *push button* dan *limit switch*, sedangkan *output* sistem terdiri dari LCD I2C dan driver motor stepper A4988 yang mengendalikan motor stepper.

Tahap awal pembuatan program dilakukan dengan menambahkan *library* yang dibutuhkan untuk mendukung komunikasi antar komponen. Pada sistem ini digunakan *library Wire*.h untuk komunikasi I2C dan LiquidCrystal_I2C.h untuk mengendalikan tampilan LCD. *Library* tersebut digunakan agar proses komunikasi dengan LCD menjadi lebih sederhana dan efisien.

Setelah proses pemanggilan *library* dilakukan, tahap berikutnya yaitu menentukan konfigurasi pin *input* dan *output* pada Arduino Nano. Konfigurasi pin dilakukan untuk mendefinisikan fungsi setiap pin yang digunakan pada sistem. Pin *output* digunakan untuk mengendalikan sinyal STEP dan DIR pada driver motor stepper, sedangkan pin *input* digunakan untuk membaca kondisi *push button* dan *limit switch*.

Tahap selanjutnya yaitu melakukan deklarasi variabel yang digunakan dalam program. Variabel digunakan untuk menyimpan parameter sistem seperti jumlah lilitan, jumlah lilitan yang sedang berjalan, diameter *wire*, posisi kursor, serta beberapa parameter lain yang digunakan selama proses *winding* berlangsung.

Setelah proses deklarasi selesai dilakukan, program kemudian dibagi menjadi beberapa fungsi untuk mempermudah pengelolaan program dan meningkatkan

keterbacaan kode. Fungsi-fungsi tersebut meliputi proses pembacaan tombol, pengaturan jumlah lilitan, pengaturan diameter *wire*, proses homing, pemeriksaan *limit switch*, dan proses *winding*.

Pada program juga dibuat sistem navigasi menu menggunakan *push button* agar pengguna dapat melakukan pengaturan parameter alat secara langsung melalui LCD. Sistem navigasi tersebut digunakan untuk mengatur jumlah lilitan dan diameter kawat sebelum proses *winding* dimulai.

Setelah seluruh fungsi program selesai dibuat, dilakukan proses integrasi antar fungsi agar seluruh bagian sistem dapat bekerja secara berurutan. Integrasi program dilakukan mulai dari proses inisialisasi sistem, pengaturan parameter, proses *winding*, hingga proses homing setelah proses *winding* selesai.

Untuk mempermudah pemahaman terhadap struktur program yang dibuat, potongan program utama ditampilkan pada Program 4-1 dan Program 4-2 berikut.

Program 4- 1 Inisialisasi Komponen Perangkat Keras

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

// Motor Stepper
#define STEP_KOKER   9
#define DIR_KOKER    8
#define STEP_COIL    11
#define DIR_COIL     10

// Sensor
#define LIMIT_LEFT   2
#define HALL_SENSOR  3

// Push Button
#define BTN_UP       4
#define BTN_DOWN     5
#define BTN_LEFT     6
#define BTN_RIGHT    7
#define BTN_ENTER    12
#define BTN_BACK     A0
```

Program 4- 2 Deklarasi Variabel dan Parameter Program

```
volatile long turnCounter = 0;
int digitTurn[4] = {0,0,0,0};
int cursorTurn = 0;
long targetTurn = 0;
float wireDiameter = 0.20;
float step_mm = 0.0035;
int directionGuide = HIGH;
int digitWire[4] = {0,2,0,0};
int cursorWire = 0;
```

Selanjutnya dilakukan penjelasan mengenai cara kerja program pada masing-masing komponen yang digunakan pada sistem *Coil winder* otomatis.

4.3.2 Cara Kerja Program

Pada tahap ini dijelaskan cara kerja program yang digunakan pada alat *Coil winder* otomatis berdasarkan fungsi-fungsi yang telah dibuat pada program Arduino. Setiap fungsi memiliki tugas dan peran yang berbeda dalam mengendalikan sistem sehingga seluruh komponen dapat bekerja secara terintegrasi sesuai dengan proses yang telah dirancang. Pembagian program ke dalam beberapa fungsi dilakukan untuk mempermudah proses pengelolaan program, meningkatkan keterbacaan kode, serta mempermudah proses perbaikan apabila terjadi kesalahan pada sistem.

Program pada alat ini dirancang agar dapat menjalankan proses secara berurutan mulai dari proses inisialisasi sistem, pembacaan *input* pengguna, pengaturan parameter pelilitan, proses *winding*, hingga proses homing setelah proses selesai dilakukan. Selain itu program juga dirancang agar mampu mengendalikan motor stepper, membaca kondisi *limit switch*, serta menampilkan informasi proses kerja alat melalui LCD secara real-time.

Pada subbab ini akan dijelaskan cara kerja setiap fungsi yang digunakan pada program Arduino beserta potongan kode yang berkaitan dengan fungsi tersebut. Penjelasan dilakukan untuk memberikan gambaran mengenai alur kerja program serta

hubungan antar fungsi dalam mengendalikan seluruh sistem pada alat *Coil winder* otomatis.

a. Fungsi Inisialisasi Sistem (setup())

Program 4- 3 Fungsi setup()

```
void setup()
{
    pinMode(STEP_KOKER,OUTPUT);
    pinMode(DIR_KOKER,OUTPUT);

    pinMode(STEP_COIL,OUTPUT);
    pinMode(DIR_COIL,OUTPUT);

    pinMode(LIMIT_LEFT,INPUT_PULLUP);
    pinMode(LIMIT_RIGHT,INPUT_PULLUP);

    pinMode(HALL_SENSOR,INPUT_PULLUP);

    attachInterrupt(
        digitalPinToInterrupt(HALL_SENSOR),
        hallISR,
        FALLING);

    lcd.init();
    lcd.backlight();

    homeCoil();
}
```

Program 4-3 diatas menunjukkan fungsi setup() yang dieksekusi satu kali ketika Arduino Nano pertama kali dinyalakan atau setelah proses *reset*. Fungsi ini bertujuan untuk melakukan inisialisasi seluruh perangkat keras yang digunakan sehingga sistem berada pada kondisi siap bekerja sebelum memasuki program utama (loop()).

Tahap pertama pada fungsi setup() adalah melakukan konfigurasi arah setiap pin masukan dan keluaran menggunakan fungsi pinMode(). Pin yang terhubung ke driver motor stepper dikonfigurasi sebagai OUTPUT karena digunakan untuk menghasilkan sinyal STEP dan DIR menuju driver A4988.

Sebaliknya, pin yang terhubung dengan limit switch, Hall Sensor, dan push button dikonfigurasi sebagai INPUT_PULLUP sehingga kondisi logika masukan tetap stabil tanpa memerlukan resistor pull-up eksternal pada rangkaian.

Selanjutnya dilakukan konfigurasi Hall Sensor A3144 menggunakan fungsi attachInterrupt(). Hall Sensor dihubungkan pada pin interrupt Arduino Nano sehingga setiap perubahan logika akibat magnet yang melewati sensor dapat langsung dideteksi oleh mikrokontroler. Ketika interrupt terjadi, Arduino akan menjalankan fungsi hallISR() yang bertugas menambah nilai penghitung putaran (turnCounter). Penggunaan interrupt dipilih karena mampu mendeteksi pulsa Hall Sensor secara cepat tanpa dipengaruhi oleh proses lain yang sedang dijalankan oleh program utama.

Setelah seluruh pin selesai dikonfigurasi, sistem melakukan inisialisasi LCD 16×2 melalui antarmuka I2C menggunakan fungsi lcd.init() dan lcd.backlight(). Proses ini bertujuan agar LCD siap digunakan sebagai media tampilan informasi selama proses pengoperasian mesin.

Tahap terakhir pada fungsi setup() adalah memanggil fungsi homeCoil(). Fungsi ini digunakan untuk mengembalikan posisi awal mekanisme wire guide menuju titik referensi (*home position*) dengan memanfaatkan limit switch kiri. Proses homing dilakukan setiap kali sistem dinyalakan agar posisi awal wire guide selalu sama sebelum proses penggulungan dimulai. Dengan demikian, proses winding dapat berlangsung secara konsisten dan mengurangi kemungkinan terjadinya kesalahan posisi pada awal penggulungan.

b. Fungsi Pembacaan Hall Sensor (hallISR())

Program 4- 4 Fungsi hallISR()

```
volatile long turnCounter = 0;
void hallISR()
{
    turnCounter++;
}
pinMode(HALL_SENSOR, INPUT_PULLUP);
```

```
attachInterrupt(
    digitalPinToInterrupt(HALL_SENSOR),
    hallISR,
    FALLING);
```

Program 4-4 diatas menunjukkan fungsi hallISR() yang merupakan Interrupt Service Routine (ISR) yang digunakan untuk membaca pulsa dari Hall Sensor A3144 selama proses penggulungan berlangsung. Berbeda dengan fungsi biasa yang dipanggil oleh program utama, fungsi ISR akan dijalankan secara otomatis oleh Arduino Nano ketika terjadi interrupt pada pin yang telah dikonfigurasi.

Pada penelitian ini Hall Sensor A3144 dihubungkan ke pin D3 Arduino Nano, yang merupakan salah satu pin external interrupt. Sensor bekerja dengan mendeteksi medan magnet dari magnet permanen yang dipasang pada poros spool. Setiap satu kali putaran spool, magnet akan melewati Hall Sensor sehingga menghasilkan perubahan logika dari HIGH menjadi LOW (falling edge). Perubahan logika tersebut akan memicu interrupt dan secara otomatis menjalankan fungsi hallISR().

Di dalam fungsi hallISR(), program hanya melakukan satu proses yaitu menambahkan nilai variabel turnCounter sebesar satu setiap kali interrupt terjadi. Variabel turnCounter dideklarasikan sebagai volatile karena nilainya dapat berubah sewaktu-waktu akibat proses interrupt, sehingga Arduino selalu membaca nilai terbaru dari memori tanpa melakukan optimasi oleh compiler.

Konsep pembacaan Hall Sensor pada penelitian ini menggunakan konfigurasi satu magnet untuk satu putaran spool, sehingga hubungan antara pembacaan sensor dan putaran spool dapat dinyatakan sebagai berikut.

$$1\ Pulsa\ Hall\ Sensor = 1\ Putaran\ Spool$$

Dengan konfigurasi tersebut, nilai turnCounter merepresentasikan jumlah putaran aktual spool selama proses coil winding. Data tersebut selanjutnya

digunakan oleh fungsi startWinding() untuk memantau proses penggulungan serta ditampilkan pada LCD sebagai informasi jumlah putaran aktual (ENC).

Penggunaan metode interrupt dipilih karena memiliki respon yang lebih cepat dibandingkan metode polling. Dengan interrupt, Arduino Nano tidak perlu melakukan pembacaan sensor secara terus-menerus pada program utama, sehingga kemungkinan kehilangan pulsa ketika spool berputar dapat dikurangi. Selain itu, metode ini memungkinkan proses pembacaan Hall Sensor berlangsung bersamaan dengan proses pengendalian motor stepper dan pembaruan tampilan LCD.

c. Fungsi Pengaturan Jumlah Lilitan (inputLilitan())

Program 4- 5 Fungsi tombol()

```
bool tombol(int pin)
{
    if(digitalRead(pin)==LOW)
    {
        delay(50);
        if(digitalRead(pin)==LOW)
        {
            while(digitalRead(pin)==LOW);
           delay(50);
            return true;
        }
    }
    return false;
}
```

Program 4-5 diatas menunjukkan fungsi tombol() yang digunakan untuk membaca kondisi seluruh push button yang terdapat pada mesin *coil winding*. Fungsi ini dipanggil oleh beberapa fungsi lain, seperti inputLilitan(), inputDiameter(), dan startWinding(), sehingga seluruh pembacaan tombol dilakukan menggunakan satu fungsi yang sama. Dengan cara ini, program menjadi lebih sederhana, mudah dipelihara, serta menghindari penulisan kode yang berulang.

Pada fungsi ini, Arduino Nano membaca kondisi logika setiap tombol menggunakan fungsi digitalRead(). Seluruh push button dikonfigurasi menggunakan mode INPUT_PULLUP, sehingga ketika tombol tidak ditekan kondisi pin bernilai HIGH, sedangkan ketika tombol ditekan kondisi pin berubah menjadi LOW. Oleh karena itu, fungsi terlebih dahulu memeriksa apakah nilai pembacaan pin sama dengan LOW sebagai indikasi bahwa tombol sedang ditekan.

Setelah tombol terdeteksi ditekan, program memberikan waktu tunda (*delay*) selama 50 ms sebagai proses *software debounce*. Debounce dilakukan untuk mengurangi pengaruh *contact bouncing* yang umum terjadi pada saklar mekanik, sehingga satu kali penekanan tombol tidak terbaca sebagai beberapa kali penekanan.

Selanjutnya program melakukan pembacaan ulang kondisi tombol untuk memastikan bahwa tombol benar-benar masih berada pada kondisi ditekan. Apabila kondisi tetap LOW, program akan menunggu hingga tombol dilepas menggunakan perulangan while(digitalRead(pin)==LOW);. Metode ini memastikan bahwa satu kali penekanan tombol hanya menghasilkan satu kali pembacaan (*single press*), sehingga navigasi menu menjadi lebih stabil dan mudah dikendalikan oleh pengguna.

Apabila seluruh proses pembacaan berhasil dilakukan, fungsi akan mengembalikan nilai true sebagai indikasi bahwa tombol telah ditekan. Sebaliknya, apabila tidak terdapat penekanan tombol, fungsi akan mengembalikan nilai **false** sehingga program utama dapat melanjutkan proses berikutnya tanpa melakukan perubahan.

Pada penelitian ini, fungsi tombol() digunakan untuk membaca enam buah push button yang memiliki fungsi berbeda pada sistem. Adapun fungsi masing-masing tombol adalah sebagai berikut.

Tabel 4- 2 Fungsi Push Button

Tombol	Fungsi
UP	Menambah nilai parameter yang sedang dipilih.
DOWN	Mengurangi nilai parameter yang sedang dipilih.
LEFT	Memindahkan posisi kursor ke kiri.
RIGHT	Memindahkan posisi kursor ke kanan.
ENTER	Menyimpan parameter atau menjalankan proses winding, serta digunakan untuk *pause* dan *resume* selama proses berlangsung.
BACK	Membatalkan proses pengaturan atau menghentikan proses winding.

d. Fungsi Pengaturan Jumlah Lilitan (inputLilitan())

Program 4- 6 Inisialisasi fungsi inputLilitan()

```
void inputLilitan()
{
    cursorTurn = 0;
    bool update = true;

    while(true)
    {
        if(update)
        {
            lcd.clear();

            lcd.setCursor(0,0);
            lcd.print("SET LILITAN");

            lcd.setCursor(0,1);

            for(int i=0;i<4;i++)
                lcd.print(digitTurn[i]);

            lcd.setCursor(cursorTurn,1);
            lcd.blink();

            update = false;
        }
```

Program 4- 7 Pembacaan tombol navigasi

```
if(tombol(BTN_UP))
{
    digitTurn[cursorTurn]++;

    if(digitTurn[cursorTurn]>9)
        digitTurn[cursorTurn]=0;
    update=true;
}
if(tombol(BTN_DOWN))
{
    digitTurn[cursorTurn]--;
    if(digitTurn[cursorTurn]<0)
        digitTurn[cursorTurn]=9;
    update=true;
}
```

Program 4- 8 Penyimpanan jumlah lilitan

```
if(tombol(BTN_ENTER))
{
    targetTurn =
    digitTurn[0]*1000 +
    digitTurn[1]*100 +
    digitTurn[2]*10 +
    digitTurn[3];
    lcd.noBlink();
    return;
}
```

Program 4-6, Program 4-7, dan Program 4-8 diatas menunjukkan fungsi inputLilitan() yang digunakan untuk memasukkan jumlah lilitan yang diinginkan sebelum proses coil winding dimulai. Nilai jumlah lilitan dimasukkan menggunakan enam buah push button sebagai media interaksi antara pengguna dan sistem. Parameter yang telah dimasukkan kemudian disimpan ke dalam variabel targetTurn sebagai target jumlah lilitan selama proses penggulungan.

Pada awal fungsi, sistem menginisialisasi posisi kursor pada digit pertama dan mengaktifkan variabel update. Variabel ini digunakan untuk mengontrol proses pembaruan tampilan LCD sehingga tampilan hanya diperbarui ketika

terjadi perubahan nilai atau perpindahan posisi kursor. Pendekatan ini membuat proses tampilan lebih efisien karena LCD tidak diperbarui secara terus-menerus.

Selanjutnya sistem menampilkan menu SET LILITAN pada LCD beserta empat digit angka yang dapat diubah oleh pengguna. Posisi digit yang sedang dipilih ditandai dengan fitur blink pada LCD sehingga pengguna dapat mengetahui digit yang sedang diedit.

Perubahan nilai setiap digit dilakukan menggunakan tombol UP dan DOWN. Tombol UP digunakan untuk menambah nilai digit, sedangkan tombol DOWN digunakan untuk mengurangi nilai digit. Nilai digit dibatasi antara 0 hingga 9, sehingga apabila nilai melebihi batas maksimum maka akan kembali ke angka 0, sedangkan apabila nilai kurang dari 0 maka akan kembali ke angka 9.

Perpindahan antar digit dilakukan menggunakan tombol LEFT dan RIGHT. Tombol tersebut berfungsi memindahkan posisi kursor sehingga pengguna dapat mengubah setiap digit secara independen. Dengan metode ini, jumlah lilitan dapat dimasukkan hingga empat digit, yaitu mulai dari 0000 sampai 9999 lilitan.

Setelah seluruh parameter selesai dimasukkan, pengguna menekan tombol ENTER. Program kemudian menggabungkan keempat digit menjadi sebuah bilangan bulat menggunakan operasi perkalian berdasarkan nilai tempat (ribuan, ratusan, puluhan, dan satuan). Nilai tersebut selanjutnya disimpan pada variabel targetTurn sebagai target jumlah lilitan yang akan digunakan oleh fungsi startWinding().

e. Fungsi Pengaturan Diameter Kawat (inputDiameter())

Program 4- 9 Inisialisasi fungsi inputDiameter()

```
void inputDiameter()
{
    cursorWire = 0;
    bool update = true;
    while(true)
```

```
    {
        if(update)
        {
            lcd.clear();

            lcd.setCursor(0,0);
            lcd.print("SET DIA KAWAT");
            lcd.setCursor(0,1);
            lcd.print(digitWire[0]);
            lcd.print(".");
            lcd.print(digitWire[1]);
            lcd.print(digitWire[2]);
            lcd.print(digitWire[3]);
```

Program 4- 10 Pembacaan tombol pengaturan diameter

```
      if(tombol(BTN_UP))
      {
          digitWire[cursorWire]++;
          if(cursorWire==0)
          {
              if(digitWire[0]>1)
                  digitWire[0]=0;
          }
          else
          {
              if(digitWire[cursorWire]>9)
                  digitWire[cursorWire]=0;
          }
          update=true;
}
```

Program 4- 11 Penyimpanan diameter kawat

```
if(tombol(BTN_ENTER))
{
    wireDiameter =
    (
        digitWire[0]*1000 +
        digitWire[1]*100 +
        digitWire[2]*10 +
        digitWire[3]
    ) / 1000.0;
    if(wireDiameter<0.10)
        wireDiameter=0.10;
    if(wireDiameter>1.00)
```

```
        wireDiameter=1.00;

    lcd.noBlink();
    return;
}
```

Program 4-9, Program 4-10, dan Program 4-11 diatas menunjukkan fungsi inputDiameter() yang digunakan untuk memasukkan nilai diameter kawat email yang akan digunakan pada proses penggulungan. Diameter kawat merupakan salah satu parameter penting karena digunakan untuk menentukan jarak perpindahan *wire guide* setiap satu kali putaran spool. Dengan pengaturan diameter yang sesuai, setiap lilitan kawat dapat tersusun secara rapi tanpa saling bertumpuk maupun meninggalkan celah yang terlalu besar.

Pada awal fungsi, sistem menginisialisasi posisi kursor pada digit pertama serta mengaktifkan variabel update. Variabel tersebut berfungsi untuk mengontrol proses pembaruan tampilan LCD sehingga tampilan hanya diperbarui ketika terjadi perubahan data atau perpindahan posisi kursor.

Selanjutnya LCD menampilkan menu SET DIA KAWAT beserta nilai diameter dalam format desimal. Nilai diameter disimpan dalam array digitWire[] yang terdiri atas empat digit, di mana digit pertama menunjukkan nilai sebelum tanda desimal, sedangkan tiga digit berikutnya menunjukkan nilai di belakang koma. Dengan format tersebut, pengguna dapat memasukkan diameter kawat mulai dari 0,100 mm hingga 1,000 mm.

Perubahan nilai setiap digit dilakukan menggunakan tombol UP dan DOWN, sedangkan perpindahan posisi digit dilakukan menggunakan tombol LEFT dan RIGHT. Untuk menjaga agar nilai diameter tetap berada pada rentang yang digunakan dalam penelitian, program memberikan batas minimum sebesar 0,10 mm dan batas maksimum sebesar 0,40 mm.

Setelah pengguna menekan tombol ENTER, seluruh digit digabungkan menjadi sebuah bilangan bertipe *float* dan disimpan pada variabel wireDiameter. Nilai tersebut selanjutnya digunakan sebagai parameter untuk menghitung

jumlah langkah motor *wire guide* pada setiap satu putaran spool selama proses winding berlangsung.

f. Fungsi Proses Winding (startWinding()))

Program 4- 12 Inisialisasi proses winding

```
void startWinding()
{
    bool paused = false;
    long stepGuide = wireDiameter / step_mm;
    resetCounter();
    digitalWrite(DIR_KOKER,HIGH);
    digitalWrite(DIR_COIL,directionGuide);
    lcd.clear();
    while(getTurnCounter() < targetTurn)
    {
```

Program 4- 13 Proses pause dan resume

```
if(tombol(BTN_ENTER))
{
    paused = !paused;
    lcd.clear();
    if(paused)
        lcd.print("PAUSED");
    else
    {
        lcd.print("RESUME");
        delay(500);
    }
}
while(paused)
{
    if(tombol(BTN_ENTER))
    {
        paused=false;
        lcd.clear();
        lcd.print("RESUME");
        delay(500);
    }
}
```

Program 4- 14 Proses winding

```
rotateSpool();
```

```
moveGuide(stepGuide);
updateLCD();
```

Program 4- 15 Penyelesaian proses winding

```
finishMessage();
```

Program 4-12, Program 4-13, Program 4-14, Program 4-15 diatas menunjukkan fungsi startWinding() yang merupakan fungsi utama yang mengendalikan seluruh proses penggulungan kawat (*coil winding*). Fungsi ini mulai dijalankan setelah pengguna selesai memasukkan parameter jumlah lilitan dan diameter kawat. Selama proses winding berlangsung, fungsi ini mengatur sinkronisasi antara motor stepper pemutar spool, motor *wire guide*, Hall Sensor, LCD, serta pembacaan tombol kontrol sehingga seluruh proses dapat berjalan secara otomatis.

Pada awal fungsi, program menginisialisasi variabel paused yang digunakan sebagai indikator kondisi jeda (*pause*) selama proses winding. Selain itu, dilakukan perhitungan nilai stepGuide berdasarkan diameter kawat (wireDiameter) dan nilai perpindahan linear setiap langkah motor (step_mm). Nilai tersebut menentukan jumlah langkah motor *wire guide* yang harus dilakukan setiap satu kali putaran spool agar jarak antar lilitan sesuai dengan diameter kawat yang digunakan.

Selanjutnya program mengatur arah putaran kedua motor stepper menggunakan pin DIR pada driver A4988. Setelah proses inisialisasi selesai, sistem memasuki perulangan while() yang akan terus dijalankan selama jumlah putaran aktual yang dibaca Hall Sensor (turnCounter) masih lebih kecil dibandingkan jumlah lilitan target (targetTurn).

Berbeda dengan program versi awal yang menghitung jumlah lilitan berdasarkan jumlah langkah motor (*step count*), pada program hasil revisi proses winding dipantau menggunakan Hall Sensor A3144. Setiap satu kali magnet melewati Hall Sensor, interrupt akan dijalankan sehingga nilai turnCounter

bertambah satu. Dengan demikian, jumlah putaran yang digunakan selama proses winding merupakan hasil pembacaan aktual sensor.

g. Fungsi Pemeriksaan Limit Switch (checkLimit())

Program 4- 16 Fungsi checkLimit()

```
void checkLimit()
{
    if(digitalRead(LIMIT_LEFT)==LOW)
    {
        directionGuide = HIGH;
        digitalWrite(DIR_COIL,directionGuide);
    }
    if(digitalRead(LIMIT_RIGHT)==LOW)
    {
        directionGuide = LOW;
        digitalWrite(DIR_COIL,directionGuide);
    }
}
```

Program 4-16 diatas menunjukkan fungsi checkLimit() yang digunakan untuk memantau kondisi kedua limit switch yang dipasang pada mekanisme *wire guide*. Fungsi ini bertugas mengubah arah pergerakan motor stepper *wire guide* ketika salah satu limit switch aktif, sehingga pergerakan mekanisme tetap berada di dalam batas lintasan yang telah ditentukan.

Pada sistem yang dirancang, digunakan dua buah limit switch, yaitu limit switch kiri (LIMIT_LEFT) dan limit switch kanan (LIMIT_RIGHT). Kedua sensor tersebut dipasang pada kedua ujung lintasan *wire guide* sebagai pembatas mekanis. Ketika *wire guide* mencapai salah satu ujung lintasan, limit switch akan aktif dan memberikan sinyal logika LOW kepada Arduino Nano.

Program membaca kondisi kedua limit switch menggunakan fungsi digitalRead(). Apabila limit switch kiri aktif, variabel directionGuide diubah menjadi HIGH sehingga arah putaran motor stepper berubah menuju sisi kanan. Sebaliknya, apabila limit switch kanan aktif, variabel directionGuide diubah menjadi LOW sehingga motor bergerak kembali menuju sisi kiri.

Perubahan arah tersebut dikirimkan ke driver A4988 melalui pin DIR_COIL menggunakan fungsi digitalWrite(). Dengan mekanisme ini, motor *wire guide* dapat bergerak bolak-balik secara otomatis mengikuti panjang spool tanpa memerlukan intervensi pengguna.

h. Fungsi Homing (homeCoil())

Program 4- 17 Fungsi homeCoil()

```
      void homeCoil()
      {
          lcd.clear();
          lcd.setCursor(0,0);
          lcd.print("HOMING...");

          digitalWrite(DIR_COIL,LOW);
          while(digitalRead(LIMIT_LEFT)==HIGH)
          {
              digitalWrite(STEP_COIL,HIGH);
              delayMicroseconds(1000);

              digitalWrite(STEP_COIL,LOW);
              delayMicroseconds(1000);
          }
          delay(300);
          directionGuide=HIGH;
}
```

Program 4-17 diatas menunjukkan fungsi homeCoil() yang digunakan untuk mengembalikan posisi awal (*home position*) mekanisme wire guide sebelum maupun sesudah proses *coil winding*. Proses homing bertujuan agar setiap siklus penggulungan selalu dimulai dari titik referensi yang sama, sehingga posisi awal wire guide tetap konsisten dan hasil penggulungan menjadi lebih seragam.

Pada sistem yang dirancang, posisi home ditentukan menggunakan limit switch kiri (LIMIT_LEFT)**.** Ketika fungsi homeCoil() dipanggil, Arduino Nano terlebih dahulu mengatur arah putaran motor stepper wire guide menuju sisi kiri dengan memberikan logika LOW pada pin DIR_COIL. Selanjutnya motor stepper akan bergerak secara terus-menerus hingga limit switch kiri aktif.

Selama limit switch belum aktif, program akan menghasilkan pulsa STEP secara berulang menggunakan perulangan while(). Ketika wire guide menyentuh limit switch kiri, kondisi logika limit switch berubah menjadi LOW, sehingga perulangan berhenti dan motor stepper dihentikan secara otomatis.

Setelah proses homing selesai, sistem memberikan waktu tunda selama 300 ms untuk memastikan motor benar-benar berhenti dan mekanisme telah stabil. Selanjutnya variabel directionGuide diatur menjadi HIGH sebagai arah awal wire guide sebelum proses winding dimulai.

i. Fungsi Penggerak Spool (rotateSpool())

Program 4- 18 Fungsi rotateSpool()

```
      void rotateSpool()
      {
          for(int i=0;i<200;i++)
          {
              digitalWrite(STEP_KOKER,HIGH);
              delayMicroseconds(1000);
              digitalWrite(STEP_KOKER,LOW);
              delayMicroseconds(1000);
          }
}
```

Program 4-18 diatas menunjukkan fungsi rotateSpool() yang digunakan untuk menggerakkan motor stepper yang berfungsi sebagai pemutar spool selama proses penggulungan kawat. Fungsi ini menghasilkan pulsa STEP menuju driver A4988 sehingga motor stepper melakukan satu putaran penuh sesuai jumlah langkah yang telah ditentukan.

Pada penelitian ini motor stepper yang digunakan memiliki resolusi 200 langkah per putaran (1,8°/step)**.** Oleh karena itu fungsi rotateSpool() menghasilkan 200 pulsa STEP, sehingga motor melakukan satu putaran penuh. Setiap pulsa dibentuk dengan mengubah kondisi pin STEP dari HIGH menjadi LOW yang dipisahkan oleh waktu tunda menggunakan fungsi delayMicroseconds().

Selama motor spool berputar, magnet yang dipasang pada poros spool akan melewati Hall Sensor A3144 sehingga menghasilkan satu pulsa interrupt. Pulsa tersebut kemudian dihitung sebagai satu putaran aktual spool.

j. Fungsi Penggerak Wire Guide (moveGuide())

Program 4- 19 Fungsi Wire Guide (moveGuide())

```
void moveGuide(long stepGuide)
{
    for(long i=0;i<stepGuide;i++)
    {
        checkLimit();
        digitalWrite(STEP_COIL,HIGH);
        delayMicroseconds(1000);
        digitalWrite(STEP_COIL,LOW);
        delayMicroseconds(1000);
    }
}
```

Program 4-19 diatas menunjukkan fungsi moveGuide() yang digunakan untuk menggerakkan motor stepper yang berfungsi sebagai pengarah kawat (*wire guide*). Besarnya perpindahan motor ditentukan oleh parameter stepGuide, yang sebelumnya telah dihitung berdasarkan diameter kawat dan nilai perpindahan linear setiap langkah motor.

Sebelum setiap langkah diberikan kepada motor stepper, fungsi checkLimit() dipanggil untuk memeriksa kondisi limit switch. Apabila salah satu limit switch aktif, arah putaran motor akan diubah sehingga wire guide bergerak kembali ke arah berlawanan. Dengan mekanisme tersebut, wire guide dapat bergerak bolak-balik secara otomatis mengikuti panjang spool tanpa melewati batas mekanis.

k. Fungsi Pembaruan LCD (updateLCD())

Program 4- 20 Fungsi (updateLCD())

```
      void updateLCD()
      {
          long counter = getTurnCounter();
          lcd.setCursor(0,0);
```

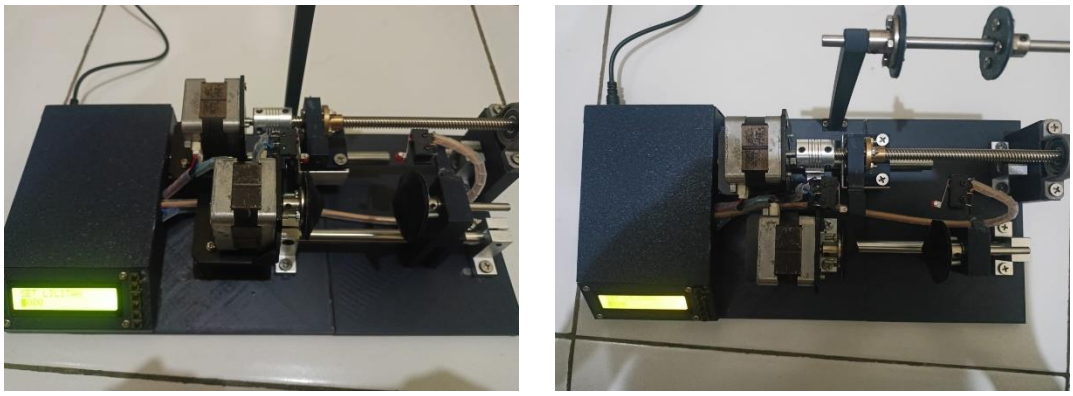
```
            lcd.print("SET:");
            lcd.print(targetTurn);
            lcd.setCursor(0,1);
            lcd.print("ENC:");
            lcd.print(counter);
}
```

Program 4-20 diatas menunjukkan fungsi updateLCD() yang digunakan untuk memperbarui informasi yang ditampilkan pada LCD selama proses penggulungan berlangsung. Informasi yang ditampilkan meliputi jumlah lilitan target (SET) dan jumlah putaran aktual hasil pembacaan Hall Sensor (ENC).

Nilai ENC diperoleh melalui fungsi getTurnCounter(), sehingga data yang ditampilkan pada LCD selalu merupakan hasil pembacaan terbaru dari Hall Sensor. Dengan demikian pengguna dapat memantau perkembangan proses winding secara real-time serta membandingkan jumlah putaran aktual dengan target lilitan yang telah ditentukan.

4.4 Hasil Akhir Alat

Setelah proses perancangan, perakitan mekanik, pemasangan komponen elektronika, serta pengembangan perangkat lunak selesai dilakukan, maka diperoleh sebuah mesin *Coil winder* otomatis yang dapat digunakan untuk melakukan proses penggulungan kawat tembaga email secara otomatis. Hasil akhir alat yang telah direalisasikan ditunjukkan pada Gambar 4-16.



(a) (b)

Gambar 4- 8 Hasil Akhir Alat Coil Winder Otomatis (a) Tampak Depan, (b) Tampak Atas

Alat yang telah direalisasikan terdiri dari beberapa bagian utama, yaitu rangka mekanik, sistem penggerak, sistem pengarah kawat, sistem kendali, serta antarmuka pengguna. Sistem penggerak menggunakan motor stepper NEMA 17 yang berfungsi untuk memutar spool selama proses *winding* berlangsung. Selain itu, motor stepper kedua digunakan untuk menggerakkan mekanisme pengarah kawat sehingga distribusi lilitan dapat dilakukan secara otomatis dan lebih seragam.

Sistem kendali alat menggunakan Arduino Nano sebagai pengendali utama yang bertugas mengatur seluruh proses *winding* berdasarkan parameter yang dimasukkan oleh pengguna. Parameter seperti jumlah lilitan dan diameter kawat dapat diatur melalui tombol *input* yang tersedia, kemudian informasi pengaturan dan proses *winding* ditampilkan pada LCD I2C.

Secara keseluruhan, alat *Coil winder* otomatis yang telah direalisasikan mampu melakukan proses penggulungan kawat secara otomatis sesuai parameter yang telah ditentukan. Hasil realisasi alat ini selanjutnya diuji untuk mengetahui kinerja, konsistensi, distribusi lilitan, serta tingkat ketelitian yang dihasilkan sebagaimana dibahas pada BAB V.