

KONTROL 3D *PRINTER* BERBASIS ARDUINO

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh:

Muhammad Boby Kusnadi	NPM	1051714
Reza Pahlevi	NPM	1051722

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2021**

LEMBAR PENGESAHAN

KONTROL 3D *PRINTER* BERBASIS ARDUINO

Oleh:

Muhammad Bobby Kusnadi NPM 1051714

Reza Pahlevi NPM 1051722

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma IV Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



I Made Andik Setiawan, M.Eng, Ph.D

Pembimbing 2



Charlotha, M.Tr.T

Penguji 1



Indra Dwisaputra, M.T

Penguji 2



Muhammad Iqbal Nugraha, M.Eng

Penguji 3



Linda Fujiyanti, MTI

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini:

Nama Mahasiswa 1: Muhammad Bobby Kusnadi NPM: 1051714

Nama Mahasiswa 2: Reza Pahlevi NPM: 1051722

Dengan Judul : Kontrol 3D *Printer* Berbasis Arduino

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 15 Februari 2021

Nama Mahasiswa

Tanda Tangan

1. Muhammad Bobby Kusnadi



2. Reza Pahlevi



ABSTRAK

Saat ini dunia industri terus mengalami perkembangan, terutama industri yang bergerak di bidang manufaktur. Pengembangan produk oleh perusahaan manufaktur merupakan sebuah keharusan untuk memenuhi kebutuhan konsumen. Salah satu cara yang paling efisien untuk pengembangan produk tersebut adalah menggunakan 3D printer. 3D printer yang beredar di pasaran menggunakan papan kontrol RAMPS 1.4 dengan marlin firmware yang bersifat open source, sedangkan 3D printer yang dibuat pada Proyek akhir ini menggunakan Arduino sebagai sistem kontrolnya. 3D printer yang dibuat pada proyek akhir ini memiliki kemampuan untuk mencetak garis lurus dengan aplikasi Matlab sebagai aplikasi untuk memberi perintah untuk mencetak. Untuk mendapatkan hasil yang maksimal perlu dilakukan persiapan dan pengujian untuk mendapatkan parameter yang sesuai. Berdasarkan pengujian dari alat yang telah dibuat, 3D printer dapat berfungsi cukup dengan parameter seperti suhu nozzle pada kisaran 150°C dan jarak antara ujung nozzle dan alas cetak sebesar 0,25 mm. 3D printer yang telah dibuat memiliki tingkat akurasi sebesar $\pm 9,4$ mm dengan resolusi sebesar 0,16 mm dan reprodusibilitas sebesar 0,2 mm.

Kata kunci: 3D printer, Arduino, Matlab, Parameter

ABSTRACT

Currently the industrial world continues to experience development, especially industries engaged in manufacturing. Product development by manufacturing companies is a must to meet consumer needs. One of the most efficient ways to develop these products is to use a 3D printer. 3D printers on the market use RAMPS 1.4 control board with marlin firmware which is open source, while 3D printers made in this final project use Arduino as the control system. The 3D printer made in this final project has the ability to print straight lines with the Matlab application as an application to give orders to print. To get maximum results, it is necessary to prepare and test to obtain the appropriate parameters. Based on the testing of the tools that have been made, the 3D printer can function sufficiently with parameters such as nozzle temperature in the range of 150 °C and the distance between the nozzle tip and the print bed of 0.25 mm. The 3D printer that has been made has an accuracy rate of ± 9.4 mm with a resolution of 0.16 mm and a repeatability of 0.2 mm.

Keywords: 3D printer, Arduino, Matlab, Parameters

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas berkah rahmat dan hidayah-Nya, penulis bisa menyelesaikan laporan Proyek Akhir ini dengan baik. Laporan Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma IV (D-IV) di Politeknik Manufaktur Negeri Bangka Belitung.

Penulis mencoba untuk menerapkan ilmu pengetahuan yang telah didapatkan selama kurang lebih 4 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan alat dan makalah Proyek Akhir ini.

Penulis menyadari bahwa selesainya Proyek Akhir ini tidak luput dari bantuan banyak pihak yang telah membantu dan memberi dukungan dalam pembuatan alat maupun dalam menyelesaikan laporan Proyek Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada:

1. Kedua Orang tua serta keluarga yang besar selalu senantiasa memberikan kasih sayang, doa, dukungan moril maupun materi dan juga semangat.
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung dan pembimbing I yang telah meluangkan banyak waktu, tenaga, pikiran, materi dan yang paling penting motivasi di dalam memberikan pengarahan dalam penulisan karya tulis Proyek Akhir ini.
3. Ibu Charlotha, M.Tr.T selaku pembimbing II yang telah banyak saran-saran dan solusi dari masalah-masalah yang penulis hadapi selama proses penyusunan karya tulis Proyek Akhir ini.
4. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
5. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung terutama teman-teman satu kelas penulis yang telah banyak membantu selama menyelesaikan Proyek Akhir.

6. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan Proyek Akhir ini masih jauh dari sempurna dikarenakan penulis adalah manusia biasa yang tidak luput dari kesalahan, karena yang benar hanya datang dari ALLAH SWT. Akhir kata penulis mohon maaf atas kekurangan dalam penulisan karya tulis Proyek Akhir ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca. Besar harapan penulis semoga Proyek Akhir ini berguna untuk menambah wawasan dan wacana bagi rekan-rekan mahasiswa.

Sungailiat, 15 Februari 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
BAB II DASAR TEORI	3
2.1 <i>3D Printer</i>	3
2.2 Cara Kerja <i>3D Printer</i>	3
2.3 <i>Fused Deposition Modeling (FDM)</i>	4
2.4 <i>Arduino Mega 2560</i>	4
2.5 <i>Motor Stepper</i>	5
2.6 <i>Stepper Driver A4988</i>	6
2.7 <i>Thermistor NTC 100K</i>	7
2.8 <i>Cartridge Heater 40W</i>	8
2.9 <i>Modul Relay 5V</i>	9
2.10 <i>Extruder</i>	10
2.11 <i>Matlab</i>	11
BAB III METODE PELAKSANAAN	12
3.1 Tahap Pertama	14

3.2	Tahap Kedua.....	14
3.3	Tahap Ketiga.....	15
3.4	Tahap Keempat.....	16
3.5	Tahap Kelima	16
3.6	Tahap Keenam.....	16
BAB IV PEMBAHASAN.....		17
4.1	Bagian Mekanik.....	17
4.2	Pembuatan <i>Hardware</i> Elektrik Kontrol 3D <i>Printer</i> Berbasis Arduino	18
4.3	Pengujian Motor <i>Stepper</i> Nema 17 Dengan <i>Driver</i> A4988.....	19
4.4	Pembuatan Aplikasi Dengan Matlab	24
4.5	Pengujian Keseluruhan Kontrol 3D <i>Printer</i> Berbasis Arduino	27
BAB V KESIMPULAN DAN SARAN.....		31
5.1	Kesimpulan.....	31
5.2	Saran	31
DAFTAR PUSTAKA		32

DAFTAR GAMBAR

Gambar 2.1 Arduino Mega 2560 [6].....	4
Gambar 2.2 Motor <i>stepper</i> Nema 17 [7].....	5
Gambar 2. 3 <i>Driver</i> A4988 [8].....	7
Gambar 2.4 NTC 100K [9]	8
Gambar 2.5 <i>Cartridge heater</i> 40W [10]	9
Gambar 2.6 Modul <i>Relay</i> 5V [11].....	9
Gambar 2.7 Proses kerja <i>extruder</i>	10
Gambar 2.8 Logo Matlab	11
Gambar 3.1 Blok diagram alir metode pelaksanaan	13
Gambar 3.2 Blok diagram 3D printer berbasis Arduino secara umum.....	14
Gambar 3.3 Blok diagram <i>hardware</i> elektrik	15
Gambar 4.1 Rangka mekanik.....	17
Gambar 4.2 <i>Timing belt</i>	17
Gambar 4.3 <i>Timing belt pulley</i>	18
Gambar 4.4 <i>Lead screw</i>	18
Gambar 4.5 Rangkaian <i>hardware</i> elektrik.....	19
Gambar 4.6 <i>Timing belt</i> GT2	19
Gambar 4.7 <i>Timing belt pulley</i>	20
Gambar 4.8 Rangkaian <i>hardware</i> Nema 17 dan <i>driver</i> A4988	21
Gambar 4.9 Rangkaian skematik motor <i>stepper</i> Nema 17	22
Gambar 4.10 Pengujian motor <i>stepper</i> dengan <i>driver</i> A4988	24
Gambar 4.11 Desain awal aplikasi <i>input</i> nilai x dan y.....	25
Gambar 4.12 Tampilan pengujian GUI Matlab	26
Gambar 4.13 Hasil cetakan dari parameter suhu 150°C dan jarak 0,25 mm	28

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Mega 2560 [6].....	5
Tabel 2.2 Spesifikasi motor <i>stepper</i> Nema 17 [6].....	6
Tabel 2.3 Spesifikasi <i>driver</i> A4988 [8].....	7
Tabel 2.4 Spesifikasi NTC 100K [9]	8
Tabel 2.5 Spesifikasi <i>cartridge heater</i> [10]	9
Tabel 2.6 Spesifikasi <i>relay</i> 5V [11]	10
Tabel 4.1 Pemasangan pin tiap komponen.....	21
Tabel 4.2 Hasil pengujian motor <i>stepper</i> dengan <i>driver</i> A4988.....	24
Tabel 4.3 Pengujian dengan parameter yang pertama	27
Tabel 4.4 Pengujian dengan parameter yang kedua.....	28
Tabel 4.5 Uji coba cetak garis 100 mm.....	29

DAFTAR LAMPIRAN

- Lampiran 1 : Daftar Riwayat Hidup (Perorangan)
- Lampiran 2 : Program Arduino
- Lampiran 3 : Program Matlab

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini dunia industri terus mengalami perkembangan, terutama industri yang bergerak di bidang manufaktur. Dalam industri manufaktur perencanaan suatu produk menjadi bagian yang sangat penting mengingat begitu ketatnya persaingan dan cepatnya inovasi baru yang dikeluarkan oleh produsen untuk mendapatkan pasar penjualan. [1]

Pengembangan produk oleh perusahaan manufaktur merupakan sebuah keharusan untuk memenuhi kebutuhan konsumen. Beberapa perusahaan manufaktur melakukan pengembangan produk, yaitu proses dimana konsep produk harus diterjemahkan dari gambar teknik menjadi produk fisik. Pembuatan produk fisik model pertama atau *prototype* dinamakan *prototyping*. *Prototyping* sangat penting karena merupakan makna terakhir dalam verifikasi bentuk, kesesuaian, dan fungsi produk. *Rapid Prototyping* atau *Layered Manufacturing* adalah proses fabrikasi suatu produk dengan *layer by layer*, atau penambahan *raw material* berturut-turut pada layer hingga terbentuk produk yang sesuai dengan model [2]. Untuk pembuatan *prototype* tersebut, salah satu cara yang paling efisien adalah menggunakan *3 dimensions printer* (*3D Printer*). Penggunaan *3D printer* dapat memangkas biaya produksi dan waktu pengerjaannya akan lebih cepat dibandingkan dengan cara yang konvensional.

Pada Proyek Akhir ini kontrol *3D printer* menggunakan Arduino Mega 2560, sedangkan pada *3D printer* yang beredar di pasaran menggunakan papan kontrol RAMPS 1.4 dengan marlin firmware yang bersifat *open source*. Dengan menggunakan RAMPS 1.4 *3D printer* mampu mencetak bentuk 3 dimensi dengan format STL. Karena marlin firmware bersifat *open source*, maka dari itu *3D printer* yang akan dibuat pada proyek akhir ini memiliki

kemampuan untuk mencetak garis lurus dengan aplikasi Matlab sebagai aplikasi untuk memberi perintah untuk mencetak.

1.2 Rumusan Masalah

Berdasarkan berbagai hal yang telah dikemukakan di atas, maka rumusan masalah yang akan dikaji dalam Proyek Akhir ini adalah:

1. Bagaimana mengatur pergerakan sumbu x, y dan z sesuai dengan ukuran cetak yang diinginkan.
2. Bagaimana membuat pergerakan lebih dari satu sumbu secara bersamaan.

1.3 Batasan Masalah

Adapun batasan masalah dalam proyek akhir ini adalah :

1. Hanya dapat mencetak garis lurus pada sumbu x dan y.
2. Aplikasi untuk *input* nilai sumbu x dan sumbu y garis yang ingin dicetak menggunakan Matlab.
3. Untuk dimensi cetak pada sumbu z belum bisa dilakukan.

1.4 Tujuan

Tujuan Proyek Akhir ini adalah merancang Sistem Kontrol 3D *Printer* Berbasis Arduino dengan spesifikasi:

1. Dapat menjadi media pembelajaran untuk mahasiswa.
2. Sumbu x, y dan z dapat bergerak ke titik yang sesuai dengan nilai yang telah di *input* pada aplikasi Matlab.
3. Dapat mencetak garis lurus dengan bahan *filamen* jenis PLA.

BAB II

DASAR TEORI

2.1 3D Printer

3D *printer* adalah konstruksi objek tiga dimensi dari model *computer aided design* (CAD) atau model 3D digital. Pencetakan 3D sendiri merujuk pada berbagai proses di mana bahan disimpan, digabungkan atau dipadatkan dengan computer sebagai pengendali untuk membuat objek tiga dimensi, dengan bahan yang ditambahkan bersama-sama seperti molekul cair atau butiran bubuk yang menyatu bersama, biasanya lapis demi lapis.

Pada 1980-an, teknik pencetakan 3D dianggap hanya cocok untuk produksi *prototype* fungsional atau estetika, dan istilah yang lebih tepat untuk itu pada saat itu adalah pembuatan *prototype* cepat. Memasuki tahun 2019, presisi, pengulangan dan jangkauan material pencetakan 3D telah meningkat ke titik di mana beberapa proses pencetakan 3D dianggap layak sebagai teknologi produksi industri. Keunggulan utama pencetakan 3D adalah kemampuan untuk menghasilkan bentuk atau geometri yang sangat kompleks yang tidak mungkin dibuat dengan tangan, termasuk bagian berlubang atau bagian dengan struktur rangka internal untuk mengurangi berat. Pemodelan deposisi gabungan atau *fused deposition modeling* (FDM) adalah proses pencetakan 3D yang paling umum digunakan pada tahun 2020. [3]

2.2 Cara Kerja 3D Printer

Cara kerja 3D *printing* sama saja dengan *printer* inkjet konvensional dimana *printer* mencetak lapisan-lapisan cetakan warna untuk membuat sebuah objek terlihat seperti seharusnya. Hanya saja pada 3D *printer* yang digunakan bukanlah tinta tetapi *plastic molten wax* atau *filamen* sehingga menjadi sebuah objek yang diinginkan.

Berbeda dengan teknologi seperti pada mesin CNC (*Computer Numeric Control*) yaitu *subtractive manufacturing*, sedangkan 3D *printer* menggunakan

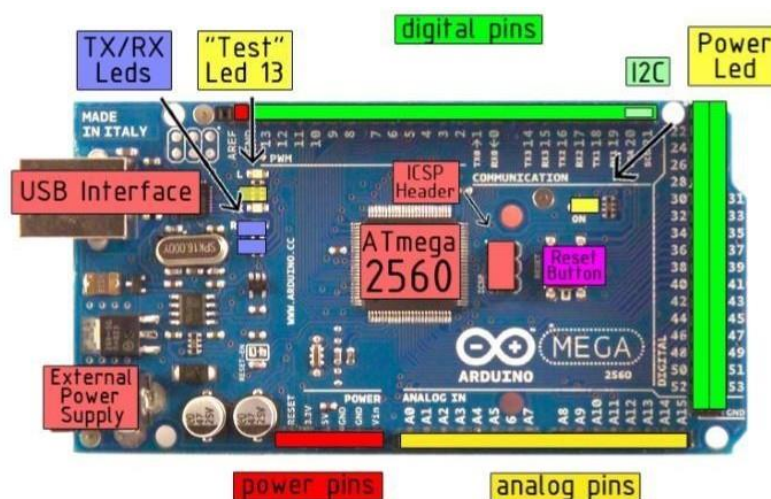
teknologi *additive manufacturing* yaitu objek terbangun dengan membentuk lapis demi lapis material, bukan membuang material seperti pada laser *cutting/milling machine*. [4]

2.3 Fused Deposition Modeling (FDM)

Jenis 3D *printer* FDM bekerja membentuk objek dengan cara melelehkan material yang kemudian ditempatkan lapis demi lapis sehingga membentuk sebuah objek yang di inginkan. Material yang digunakan dalam FDM adalah Jenis *Thermoplastic* dalam bentuk *filamen*. 3D *printer* yang beredar kebanyakan menggunakan metode FDM dan ini merupakan teknologi yang banyak digunakan pada jenis 3D *printer*. [5]

2.4 Arduino Mega 2560

Arduino Mega merupakan salah satu *microcontroller* yang berbasis Atmel AVR ATmega 2560. Spesifikasi Arduino Mega 2560 dapat dilihat pada Tabel 2.1. Terdapat 54 pin digital *input/output*, 14 pin pada Arduino Mega 2560 dapat digunakan sebagai PWM *output*, 16 pin analog *input*, 4 pin UART perangkat keras serial *port*, 16 MHz kristal osilator, satu *port* USB *connection*, *power jack*, ICSP *header*, dan tombol *reset*. [6]



Gambar 2.1 Arduino Mega 2560 [6]

Tabel 2.1 Spesifikasi Arduino Mega 2560 [6]

<i>Microcontroller</i>	ATMega 2560
Tegangan kerja	5 V
Tegangan <i>input</i> (disarankan)	7-12 V
Tegangan <i>input</i> (limit)	6-20 V
Pin digital I/O	54 (15 diantaranya merupakan <i>output</i> PWM)
Pin <i>output</i> analog	16
Arus DC tiap pin I/O	20 mA
Arus DC untuk pin 3.3 V	50 mA
Memori <i>flash</i>	256 KB dimana 8 KB digunakan oleh <i>bootloader</i>
SRAM	8 KB
EEPROM	4 KB
Kecepatan <i>clock</i>	16 MHz
Panjang	101.52 mm
Lebar	53.3 mm
Berat	37

2.5 Motor Stepper

Motor *stepper* adalah motor digital yang dikontrol oleh pulsa-pulsa digital. Motor *stepper* yang paling sederhana terdiri atas sebuah rotor dan sebuah stator yang dililit kumparan sehingga membentuk magnet listrik. [6]

Prinsip kerja sebuah motor *stepper* sama seperti sebuah magnet yang kedua ujungnya memiliki kutub berbeda. Jika pada ujung stator (elektromagnet) terdapat kutub yang sama dengan salah satu ujung dari rotor (magnet permanen), magnet akan saling tolak menolak, akibatnya rotor akan berputar. Arah perputarannya dapat searah jarum jam atau berlawanan dengan arah jarum jam. [6]



Gambar 2.2 Motor *stepper* Nema 17 [7]

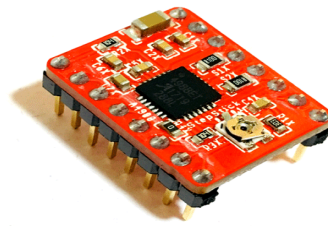
Nema 17 dipilih sebagai penggerak sumbu X, Y dan Z dengan spesifikasi pada Tabel 2.2.

Tabel 2.2 Spesifikasi motor *stepper* Nema 17 [6]

<i>Size</i>	Nema 17
<i>Step angle</i>	1.8° <i>full step</i>
<i>Voltage & Current</i>	12V at 400 mA
<i>Resistance per Phase</i>	30 <i>ohms</i>
<i>Inductance per Phase</i>	23 mH
<i>Holding Torque</i>	2000 g-cm
<i>Detent Torque</i>	220 g-cm max
<i>Weight</i>	0.24 kg (0.5 lbs.)
<i>Max continuous power</i>	5 W
<i>Rotor Inertia</i>	22 g-cm ²
<i>Bearings</i>	Ball
<i>Leads</i>	18 in. 26 AWG UL 1007
<i>Insulation resistance: >100 MΩ at 500</i>	VDC
<i>Dielectric strength</i>	500V 50Hz/minute
<i>Mounting hole space diagonal</i>	1.73 in.
<i>Mounting screws</i>	M3
<i>Shaft diameter</i>	5 mm
<i>Motor footprint</i>	1.7 in. × 1.7 in.
<i>Motor height</i>	1.5 in.
<i>Ambient temperature</i>	-10°C to +55°C

2.6 Stepper Driver A4988

Stepper driver merupakan perantara antara motor step dengan *control board* seperti Arduino. *Stepper driver* bekerja dengan menyederhanakan sinyal untuk mengendalikan motor *stepper* sehingga pembuatan program dapat menjadi lebih mudah. Pergerakan motor dapat lebih presisi karena *Stepper driver* A4988 memiliki fitur *microstep*.



Gambar 2. 3 *Driver A4988* [8]

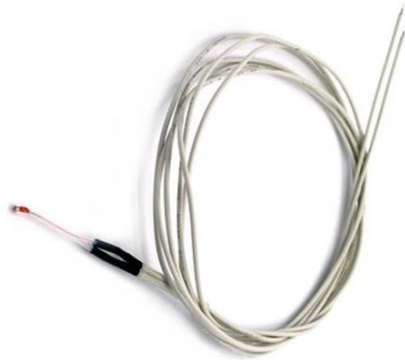
Tabel 2.3 Spesifikasi *driver A4988* [8]

<i>Driver A4988</i>	Spesifikasi
<i>Maximum operating voltage</i>	35V
<i>Minimum operating voltage</i>	8V
<i>Maximum current per phase</i>	2A
<i>Microstep resolution</i>	<i>Full step, 1/2 step, 1/4 step, 1/8 dan 1/16 step</i>
<i>Reverse voltage protection</i>	<i>No</i>
<i>Dimensions</i>	15.5× 20.5 mm (0.6" × 0.8")

2.7 Thermistor NTC 100K

Salah satu sensor temperatur yang sering digunakan adalah Thermistor. Thermistor adalah salah satu jenis *Resistor* yang nilai resistansi atau nilai hambatannya dipengaruhi oleh Temperatur. Thermistor merupakan singkatan dari “*Thermal Resistor*” yang artinya adalah *Resistor* (tahanan) yang berkaitan dengan *Thermal* (panas). Thermistor terdiri dari 2 jenis, yaitu Thermistor NTC (*Negative Temperature Coefficient*) dan Thermistor PTC (*Positive Temperature Coefficient*).

Seperti namanya, Nilai Resistansi Thermistor NTC akan turun jika temperatur di sekitar Thermistor NTC tersebut tinggi (berbanding terbalik / Negatif). Sedangkan untuk Thermistor PTC, semakin tinggi temperatur disekitarnya, semakin tinggi pula nilai resistansinya (berbanding lurus / Positif). [6]



Gambar 2.4 NTC 100K [9]

Tabel 2.4 Spesifikasi NTC 100K [9]

Thermistor	Nilai
<i>Standard resistance</i>	1.5K, 2K, 5K, 10K, 20K, 30K, 47K, 50K, 100K, 200K, 500K, etc
<i>Resistance tolerance</i>	±1%, ±2%, ±3%
<i>B-value R25/50</i>	3435K, 3600K, 3950K, 3990K, 4100K, 4200K, etc
<i>B-value accuracy</i>	±1%
<i>Operating temperature</i>	-40~+300°C
<i>Power dissipation coefficient</i>	≥ 5 mW/°C (in static air)
<i>Maximum rated power</i>	45mW
<i>Thermal time constant</i>	≤ 7S (in static air)
<i>Temperature coefficient of resistance</i>	-2~-5%/°C
<i>It is recommended to use</i>	100K, B25/50 = 3950K±1%

2.8 Cartridge Heater 40W

Elemen pemanas yang digunakan pada *extruder* pada proyek akhir ini menggunakan *cartridge heater* dengan daya 40 watt. Alasan menggunakan *cartridge heater* karena memiliki realibilitas yang lebih baik dibandingkan dengan elemen pemanas yang lainnya serta kemampuan untuk memanaskannya lebih cepat.



Gambar 2.5 Cartridge heater 40W [10]

Tabel 2.5 Spesifikasi cartridge heater [10]

Cartridge heater 40W	Spesifikasi
<i>Materials</i>	<i>Stainless steel</i>
<i>Core</i>	<i>Ceramic</i>
<i>Voltage</i>	12 Volt
<i>Power</i>	40 Watt
<i>Diameter</i> Ø6mm	Ø6 mm
<i>Length</i> 20mm	20 mm
<i>Wire Length</i>	100 cm

2.9 Modul Relay 5V

Relay adalah sebuah saklar elektrik dengan arus listrik sebagai pengendalinya, terdapat 2 bagian utama pada *relay* yaitu koil dan rangkaian saklar. Ketika koil dialiri arus listrik maka akan menimbulkan medan magnet yang akan menarik kemudian melepas plat yang ada pada rangkaian saklar dan akan menghubungkan atau memutusk arus listriknya.



Gambar 2.6 Modul Relay 5V [11]

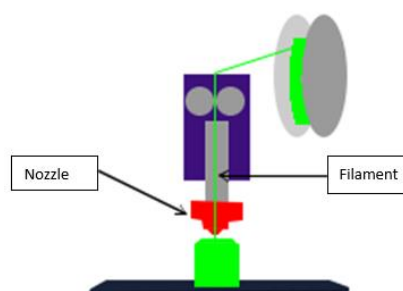
Tabel 2.6 Spesifikasi *relay* 5V [11]

Modul <i>Relay</i> 5V	Spesifikasi
<i>Relay board</i>	1 <i>channel</i>
<i>Opertaing voltage</i>	5 volt
<i>Maximum current</i>	20 mA
<i>Relay contact current capacity at AC 250 V</i>	10 A
<i>Relay contact current capacity at AC 250 V</i>	10 A
<i>contact</i>	NC and NO

2.10 *Extruder*

Gambar 2.7 menggambarkan proses penarikan filamen plastik pada *extruder* dilakukan dengan cara menghimpit filamen tersebut dengan 2 silinder *roller*. Satu silinder merupakan silinder tetap sedangkan yang satunya merupakan silinder kendali yang tersambung dengan motor *stepper*.

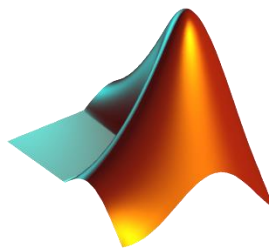
Setelah filamen ditarik, filamen akan menuju *cool end*. *Cool end* adalah silinder berlubang dengan lubang yang presisi sesuai dengan diameter filamen yang berfungsi untuk mencegah filamen macet saat proses ekstruksi. Pada bagian luar *cool end* terdapat pendingin untuk mencegah panas dari *hot end* menyebar ke *cool end*. Proses berikutnya, filamen masuk ke *hot end* yang memiliki diameter dalam yang semakin mengecil. Diameter dalam pada ujung *hot-end* menunjukkan ukuran keluaran filamen nantinya. [6]



Gambar 2.7 Proses kerja *extruder*

2.11 Matlab

Matlab atau *matrices laboratory* adalah salah satu aplikasi perangkat lunak interaktif yang digunakan untuk komputasi numerik dan visualisasi data. Dengan Matlab, pengguna dapat melakukan analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Dengan menggunakan bahasa pemrograman tingkat tinggi, Matlab mudah dioperasikan oleh penggunanya. Matlab sering dijumpai pada lingkungan akademi maupun industri.



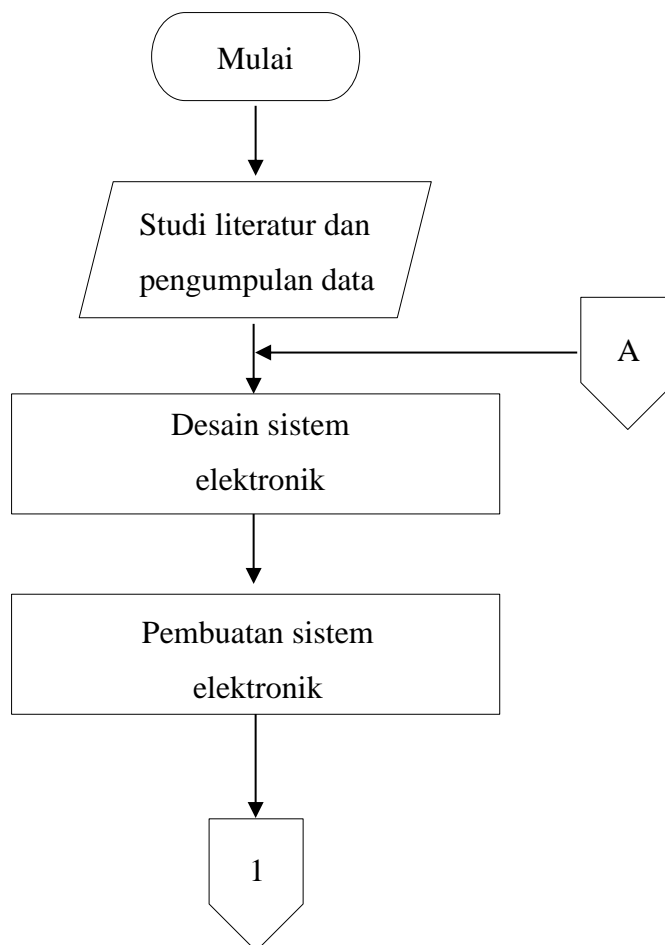
Gambar 2.8 Logo Matlab

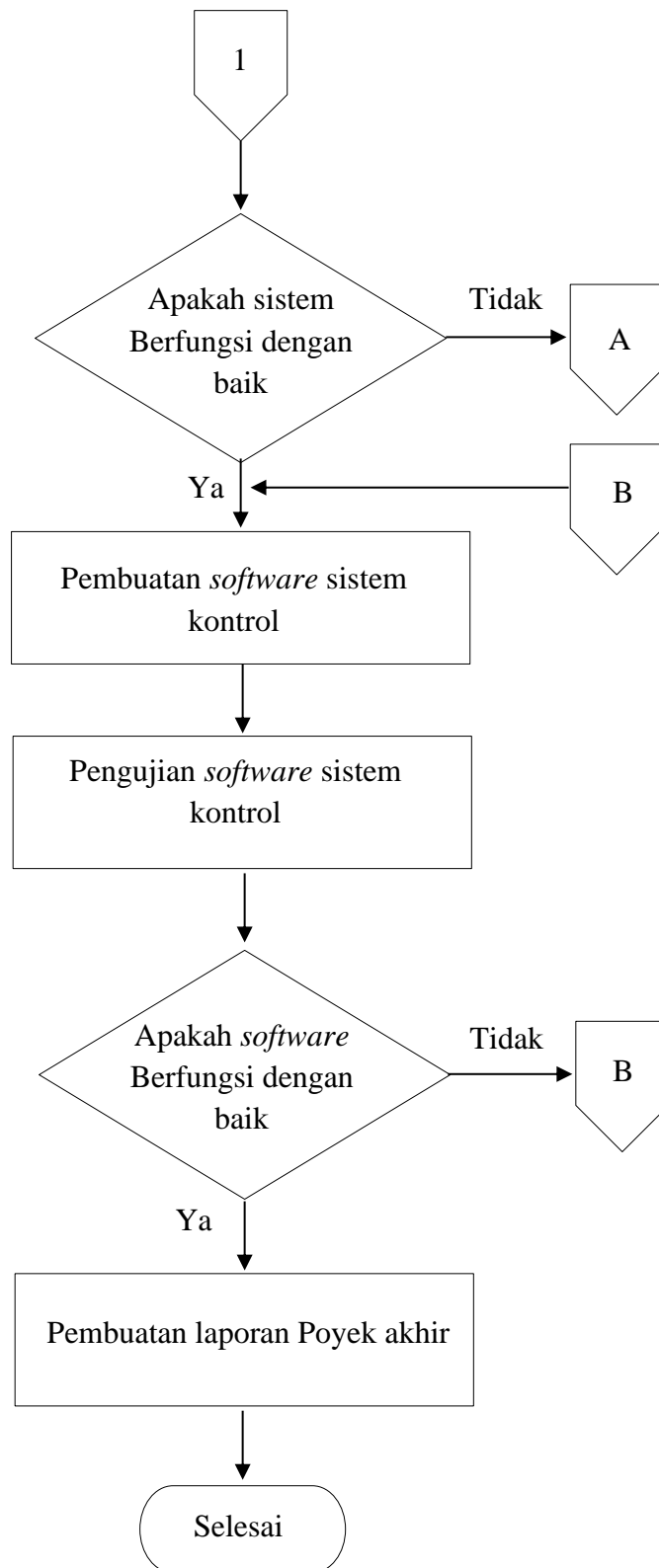
BAB III

METODE PELAKSANAAN

Dalam proses pembuatan Proyek Akhir yang berjudul *Kontrol 3D Printer Berbasis Arduino* terdiri dari beberapa tahapan yaitu sebagai berikut:

1. Studi literatur dan pengumpulan data
2. Desain sistem elektronik
3. Pembuatan sistem elektronik
4. Desain dan pembuatan *software*
5. Pengujian
6. Pembuatan laporan Proyek Akhir





Gambar 3.1 Blok diagram alir metode pelaksanaan

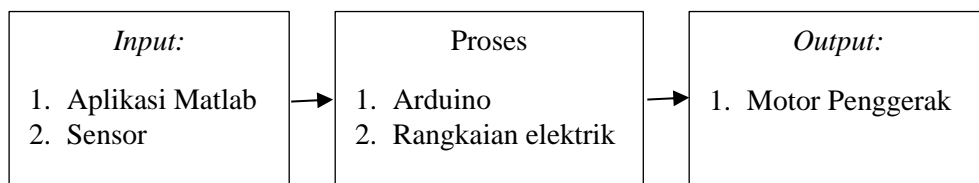
Penjelasan lebih lanjut dari tahapan di atas dapat dilihat pada langkah-langkah berikut ini:

3.1 Tahap Pertama

Melakukan studi literatur serta pengumpulan data yang berkaitan dengan 3D *printer*. Studi literatur serta pengumpulan data ini bertujuan untuk mengetahui secara garis besar sistem dari 3D *printer*, serta untuk mengatasi permasalahan yang terjadi. Data yang telah terkumpul dijadikan acuan untuk tahapan proses pembuatan kontrol 3D *printer* selanjutnya.

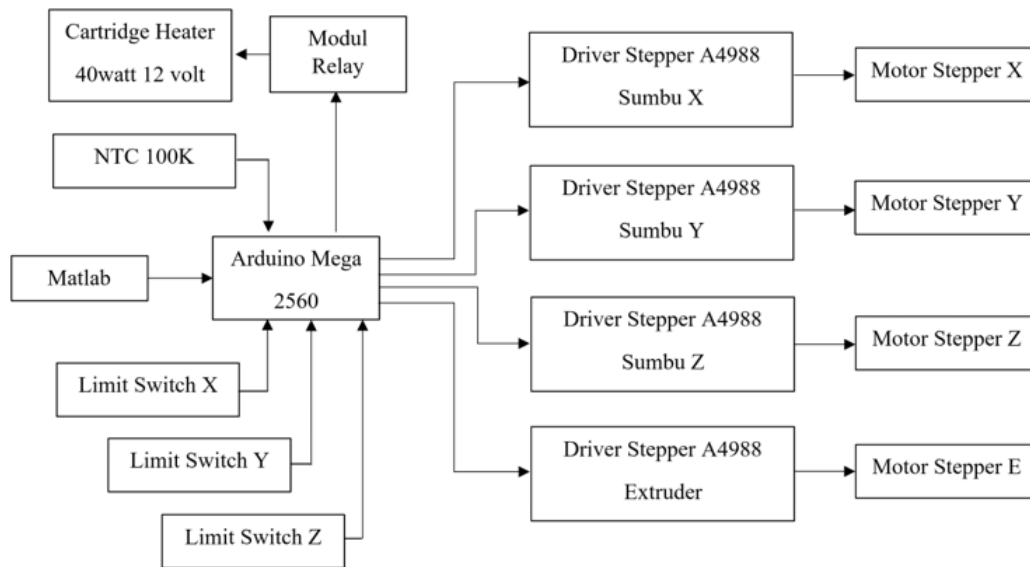
3.2 Tahap Kedua

Mendesain Proyek Akhir Kontrol 3D *Printer* Berbasis Arduino baik dari *hardware* maupun *software*. Tujuannya agar mempermudah dalam pembuatan tugas akhir ini. Perancangan *hardware* ini berupa perancangan elektrik yang dibutuhkan dalam 3D *printer* itu sendiri. Untuk perancangan *software* sistem kontrolnya menggunakan aplikasi Arduino dan aplikasi Matlab. Berikut ini adalah blok diagram 3D *printer* secara umum yang akan dibuat:



Gambar 3.2 Blok diagram 3D printer berbasis Arduino secara umum

Blok diagram *hardware* sangat penting dalam pembuatan Proyek Akhir. Sebelum merakit *hardware* elektrik, lebih baik merancang blok diagram *hardware* terlebih dahulu. Blok Diagram *hardware* untuk Proyek Akhir yang berjudul Kontrol 3D *Printer* Berbasis Arduino yaitu dapat dilihat pada gambar 3.3.



Gambar 3.3 Blok diagram *hardware* elektrik

Pada gambar 3.3 yaitu merupakan blok diagram kontrol 3D *printer* berbasis Arduino. Untuk perintah *input* nilai yang ingin dicetak menggunakan aplikasi Matlab, *input* yang dimaksud berupa nilai sumbu X dan sumbu Y. Kemudian Arduino akan memproses seluruh kontrol yang telah dibuat sehingga semua sistem bekerja sesuai dengan kontrol yang telah dibuat. Mulai dari motor *stepper* bergerak sesuai dengan nilai *input* yang telah diperintahkan, kemudian pemanas akan mulai bekerja untuk memanaskan *nozzle* sehingga proses pencetakan dapat berjalan dan akan menghasilkan *output* yang diinginkan.

3.3 Tahap Ketiga

Pembuatan *hardware* dan *software*, untuk konstruksi mekanik dalam Proyek Akhir ini penulis tidak membuat langsung konstruksi mekaniknya karena konstruksi mekaniknya membeli konstruksi yang sudah siap pakai.

Sedangkan proses pembuatan sistem kontrol meliputi:

1. Pemrograman aplikasi perintah menjalankan 3D *printer* dengan Matlab
2. Pemrograman motor *stepper* sumbu X, Y dan Z serta motor *stepper extruder*
3. Pemrograman pada sistem pemanas di 3D *printer*

3.4 Tahap Keempat

Setelah proses pembuatan *hardware* elektrik dan *software* selesai, langkah selanjutnya adalah proses uji coba alat. Proses ini bertujuan untuk mengetahui kerja alat apakah bekerja secara optimal dan berfungsi sesuai dengan fungsi yang diinginkan. Uji coba ini terdapat dua tahapan:

1. Uji coba *software*

Uji coba koneksi antara Arduino dengan *driver* A4988 dan motor *stepper* Nema 17.

2. Uji coba keseluruhan

Uji coba secara keseluruhan dilakukan untuk mengetahui hasil pengujian dari keseluruhan alat yang telah dibuat.

3.5 Tahap Kelima

Pada tahap ini merupakan tahap analisis data pada hasil uji coba yang telah dilakukan yang bertujuan untuk melihat kekurangan terhadap alat yang telah dibuat baik dari segi konstruksi, rangkaian kontrol serta program yang dibuat.

3.6 Tahap Keenam

Pembuatan laporan Proyek Akhir merupakan tahap terakhir dalam pembuatan Proyek Akhir. Setelah proyek akhir selesai, penulis membuat laporan Proyek Akhir yang bertujuan untuk merangkum temuan-temuan dan alat yang telah dibuat dalam Proyek Akhir.

Laporan Proyek Akhir terdiri dari latar belakang, tujuan, rumusan masalah, batasan masalah, landasan teori, metode pelaksanaan, pembahasan serta kesimpulan dan saran.

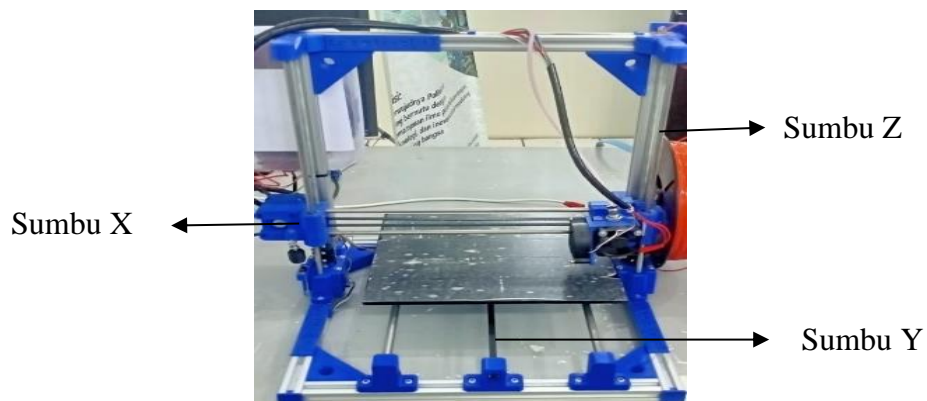
BAB IV

PEMBAHASAN

Pada bab ini akan dibahas mengenai bagian mekanik dan proses pembuatan Kontrol 3D *Printer* Berbasis Arduino yang terdiri dari perancangan sistem kontrol dan pembuatan *software*.

4.1 Bagian Mekanik

Rangka mekanik pada alat yang digunakan pada Proyek Akhir ini tidak dibuat langsung, tetapi dengan membeli rangka yang sudah siap pakai. Pada rangka mekanik terdapat beberapa komponen yang berfungsi untuk menggerakkan sumbu x, y dan z, komponen tersebut terdiri dari *timing belt*, *timing belt pulley* dan *lead screw*.



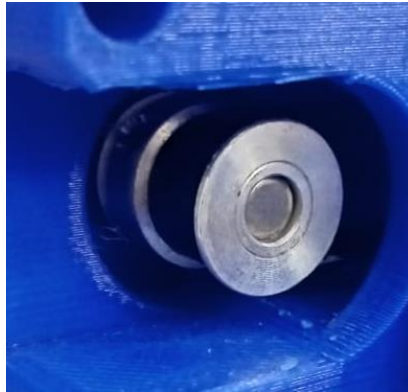
Gambar 4.1 Rangka mekanik

Timing belt adalah sabuk dari bahan karet yang kuat dan memiliki permukaan bergerigi yang digunakan untuk menggerakkan sumbu x dan sumbu y.



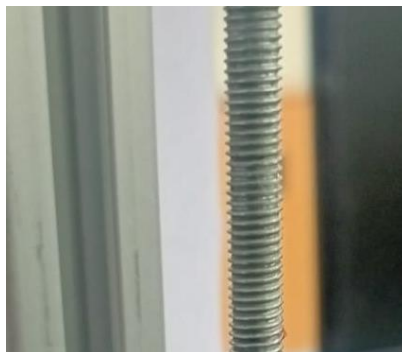
Gambar 4.2 *Timing belt*

Timing belt pulley berbentuk seperti roda bergerigi yang penggunaannya selalu dihubungkan dengan *Timing belt* dan biasanya digunakan untuk memindahkan suatu barang dari satu titik ke titik lainnya.



Gambar 4.3 *Timing belt pulley*

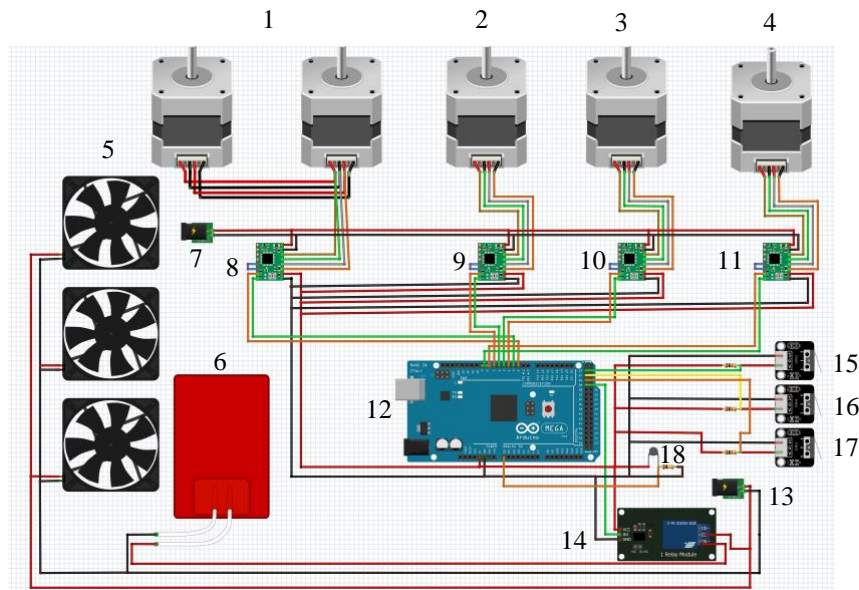
Lead screw adalah poros yang memiliki ulir sebagai pengubah gerakan dengan cara memanfaatkan gaya tekan dari perputaran ulir menjadi gerakan linier sama seperti cara kerja mur dan baut.



Gambar 4.4 *Lead screw*

4.2 Pembuatan *Hardware* Elektrik Kontrol 3D *Printer* Berbasis Arduino

Proses pembuatan *hardware* elektrik dimulai dengan perancangan rangkaian elektrik, mulai dari menentukan komponen yang diperlukan sampai ke pembelian komponen. Komponen yang digunakan pada Proyek Akhir yang berjudul kontrol 3D *printer* berbasis Arduino ini yaitu Arduino, motor *stepper*, *driver stepper*, thermistor, modul *relay*, *cartridge heater*, kipas dan *limit switch*.

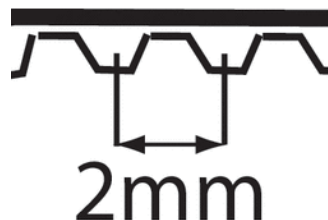


Gambar 4.5 Rangkaian *hardware* elektrik

- | | | | |
|--------------------------|---------------------|------------------------|--------------------|
| 1. Motor z sumbu | 6. Cartridge heater | 11. Driver extruder | 16. Limit switch z |
| 2. Motor x sumbu | 7. DC 12 volt | 12. Arduino Mega | 17. Limit switch y |
| 3. Motor y sumbu | 8. Driver sumbu z | 13. DC 5 volt | 18. Thermistor |
| 4. Motor <i>extruder</i> | 9. Driver sumbu x | 14. Modul <i>relay</i> | |
| 5. Kipas | 10. Driver sumbu y | 15. Limit switch x | |

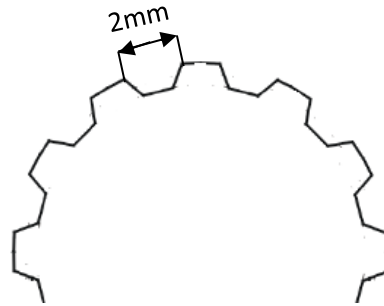
4.3 Pengujian Motor *Stepper* Nema 17 Dengan *Driver* A4988

Untuk mendapatkan rumus perhitungan untuk mencetak garis lurus pada Proyek Akhir ini kita harus mengetahui spesifikasi dari *timing belt* dan *timing belt pulley*.



Gambar 4.6 *Timing belt* GT2

Belt yang digunakan pada Proyek Akhir ini adalah tipe GT2 dimana jarak antara *pitch* yaitu 2 milimeter.



Gambar 4.7 Timing belt pulley

Tipe *timing belt pulley* yang digunakan pada Proyek Akhir ini adalah GT2 dengan jumlah gigi 32 dimana jarak antara *pitch* yaitu 2 milimeter. Perhitungan untuk mengkonversi jumlah *step* motor menjadi jarak (linier) dengan satuan milimeter.

$$\begin{aligned} \text{Jarak dari satu putaran penuh} &= \text{jumlah gigi } \textit{timing belt pulley} \times \text{jarak antar } \textit{pitch} \\ &= 16 * 2 \text{ mm} \\ &= 32 \text{ mm} \end{aligned}$$

Jadi untuk panjang satu putaran penuh motor *stepper* yaitu 32 mm.

$$\begin{aligned} \text{Satu milimeter} &= \frac{\text{langkah/putaran penuh}}{\text{jarak dari satu putaran penuh}} \\ &= \frac{200 \text{ step}}{32 \text{ mm}} \\ &= 6,25 \text{ step} \end{aligned}$$

Jadi untuk mencetak sepanjang satu milimeter diperlukan 6,25 *step*.

$$\begin{aligned} \text{Satu } \textit{step} &= \frac{\text{jarak dari satu putaran penuh}}{\text{langkah/putaran penuh}} \\ &= \frac{32 \text{ mm}}{200 \text{ step}} \\ &= 0,16 \text{ mm} \end{aligned}$$

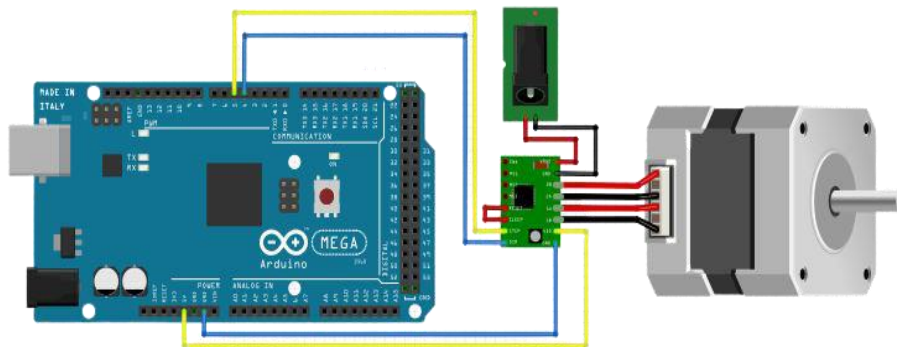
Jadi satu *step* motor *stepper* akan mencetak garis sepanjang 0,16 mm.

Jika ingin mencetak garis dengan panjang 100 mm, maka kita dapat menghitung berapa jumlah *step* yang diperlukan untuk mencetak garis tersebut dengan rumus sebagai berikut.

$$\begin{aligned} \text{Jumlah } \textit{step} &= \text{Panjang yang diinginkan (mm)} \times \textit{step/mm} \\ &= 100 \text{ mm} * 6,25 \\ &= 625 \text{ step} \end{aligned}$$

Jadi *step* motor *stepper* yang diperlukan untuk mencetak garis sepanjang 100 mm yaitu 625 *step*.

Pengujian pergerakan motor *stepper* Nema 17 dengan *driver* A4988 dengan cara memasukkan jumlah *step* yang diinginkan kemudian motor *stepper* akan berputar sesuai *step* yang diinginkan. Cara menguji motor *stepper* Nema 17 dengan *driver* A4988 yaitu sebagai berikut:



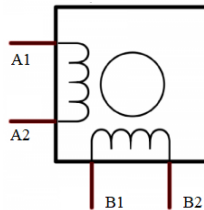
Gambar 4.8 Rangkaian *hardware* Nema 17 dan *driver* A4988

Tabel 4.1 Pemasangan pin tiap komponen

<i>Power Supply</i>	<i>Driver</i> A4988	Arduino	Nema 17
8-35 Volt	VMOT	-	-
<i>Ground</i>	GND	-	-
-	VDD	5 Volt	-
-	GND	GND	-
-	DIR	4	-
-	<i>Step</i>	5	-
-	<i>Reset ke Sleep</i>	-	-
-	1B	-	1B
-	2B	-	2B
-	1A	-	1A
-	2B	-	2B

Nema 17 memiliki 4 pin yaitu pin 1B, 2A, 1A dan 2B. Pin 2B dihubungkan ke pin 2B pada *driver* A4988, pin 2A dihubungkan ke pin 2A pada *driver* A4988,

pin 1A dihubungkan ke pin 1A pada *driver* A4988 dan pin 1B dihubungkan ke pin 1B pada *driver* A4988.



Gambar 4.9 Rangkaian skematik motor *stepper* Nema 17

Sedangkan *driver* A4988 memiliki 16 pin yaitu pin *enable*, MS1, MS2, MS3, *reset*, *sleep*, *step*, DIR, VMOT, GND, 2B, 2A, 1A, 1B, VDD dan GND. Pin VMOT dihubungkan ke *power supply* 8-35 volt 2 *ampere* dan pin GND dihubungkan ke negatif *power supply*. Pin VDD dihubungkan ke *logic power supply* 3-5.5 volt dan pin GND dihubungkan ke negatif *logic power supply*. Pin DIR dihubungkan ke pin 4 Arduino sedangkan pin *step* dihubungkan ke pin 5 Arduino dan pin *reset* dihubungkan langsung dengan pin *sleep*.

Pengujian motor *stepper* dilakukan dengan cara memasang motor *stepper* pada rangka sumbu x 3D *printer* yang sudah terpasang *timing belt* dan *body hot end*. Pada *body hot end* di pasang dengan pena sehingga pada saat motor *stepper* berputar maka *timing belt* akan bergerak dan pena akan membuat garis pada *bed* 3D *printer*.

Berikut adalah program Arduino dari pengujian motor *stepper* dengan *driver* A4988:

```
void loop() {
  Serial.println("Masukkan jumlah step cw ?");
  while (Serial.available() == 0) {}
  j_cw = Serial.parseInt();
  Serial.print("step cw = ");
  i_cw=step_lama_cw + j_cw;
  Serial.println(i_cw);
  delay(500);
  step_lama_cw = i_cw;
}
```

Program
untuk *input*
jumlah *step*
yang
diinginkan.

```

Serial.println("Masukkan jumlah step ccw ?");
while (Serial.available() == 1) {
j_ccw = Serial.parseInt();
Serial.print("step ccw = ");
i_ccw=step_lama_ccw + j_ccw;
Serial.println(i_ccw);
delay(500);
step_lama_ccw = i_ccw;
digitalWrite(DirPin, HIGH);
for(i_cw = 0; i_cw < j_cw; i_cw++)
{
digitalWrite(StepPin, HIGH);
delayMicroseconds(1000);
digitalWrite(StepPin, LOW);
delayMicroseconds(1000);
}
delay(1000);

digitalWrite(DirPin, LOW);
for(i_ccw = 0; i_ccw < j_ccw; i_ccw++)
{
digitalWrite(StepPin, HIGH);
delayMicroseconds(1000);
digitalWrite(StepPin, LOW);
delayMicroseconds(1000);
}
delay(1000);
}

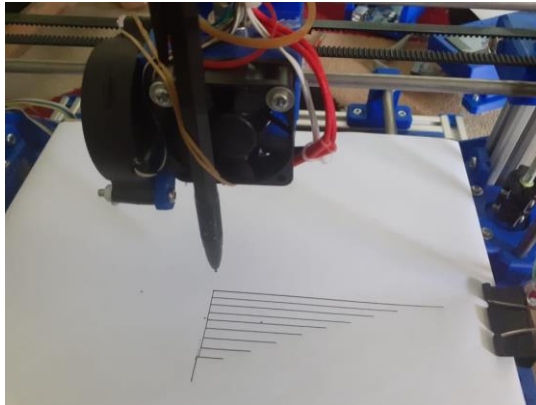
```

Step lama akan dijumlahkan dengan *step* yang baru.

Program untuk menggerakkan motor searah jarum jam.

Program untuk menggerakkan motor berlawanan arah jarum jam.

Jalankan program Arduino yang telah dibuat, masukkan jumlah *step* yang kita inginkan pada serial monitor Arduino kemudian tekan *enter*. Motor *stepper* akan berputar dan menjalankan *timing belt* dan pena akan menarik garis pada *bed 3D printer*.



Gambar 4.10 Pengujian motor *stepper* dengan *driver* A4988

Tabel 4.2 Hasil pengujian motor *stepper* dengan *driver* A4988

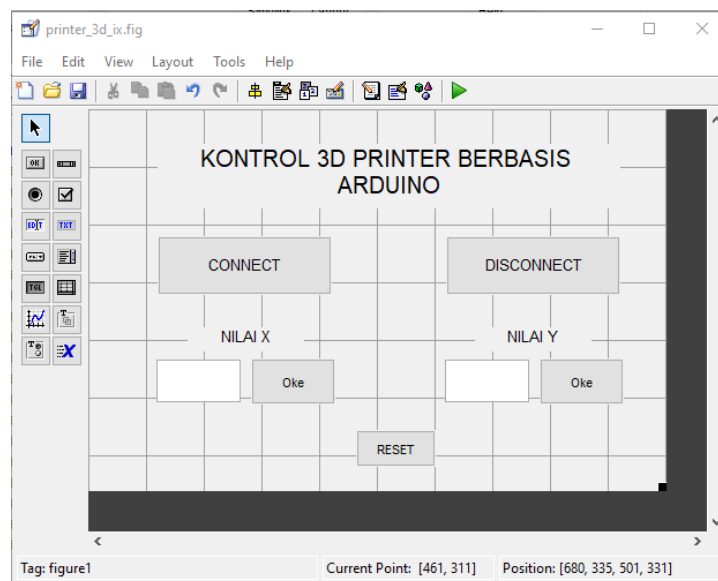
<i>Step</i>	Hasil Pengukuran (mm)	Hasil Perhitungan (mm)
62	10	10
125	20	20
187	30	30
250	40	40
312	50	50
375	60	60
473	70	70
500	80	80
562	90	90
625	100	100

Pada tabel 4.2 diatas merupakan hasil pengujian motor *stepper* dengan *driver* A4988. Dari data pengujian menggunakan pena dapat disimpulkan bahwa hasil perhitungan sama dengan hasil garis yang terbentuk.

4.4 Pembuatan Aplikasi Dengan Matlab

Dalam membuat Proyek Akhir ini menggunakan fungsi GUI, untuk mencoba interaksi antara Arduino dan Matlab. Pertama mendesain tampilan aplikasi yang akan dibuat, untuk menghubungkan dan memutuskan koneksi antara Arduino dan Matlab dibuatkanlah tombol *connect* dan *disconnect*. Kemudian

dibuatkan kolom untuk memasukkan nilai cetak sumbu x dan sumbu y serta tombol oke di setiap kolom nilai x dan nilai y yang berfungsi sebagai tombol *enter*. Untuk memudahkan pengguna membedakan kolom nilai x dan y maka dibuatkan keterangan di atas kolom nilai x dan y. Agar pengguna mudah dalam memasukkan nilai cetak x dan y yang baru dibuatkanlah tombol *reset* yang berguna untuk menghapus nilai x dan nilai y yang lama sehingga didapatkan desain seperti gambar 4.7 dibawah ini.



Gambar 4.11 Desain awal aplikasi *input* nilai x dan y

Program Matlab untuk *Graphical User Interface* (GUI) sebagai berikut.

```
global s;
s = serial('COM3','Baudrate',9600);
fopen(s);
set(handles.disconnect,'Enable','on');
set(handles.connect,'Enable','off');
```

Untuk menghubungkan arduino dengan matlab, serta mengaktifkan fungsi tombol *connect*.

```
global s;
fclose(s);
instrreset;
set(handles.disconnect,'Enable','off');
set(handles.connect,'Enable','on');
```

Untuk mengaktifkan fungsi tombol *disconnect*.

```

global s;
a=get(handles.x,'String');
fprintf(s,a)
pause(0.001)

```

} Menyimpan dan mengirim data nilai x melalui serial komunikasi.

```

global s;
b=get(handles.y,'String');
fprintf(s,b)
pause(0.001)

```

} Menyimpan dan mengirim data nilai y melalui serial komunikasi.

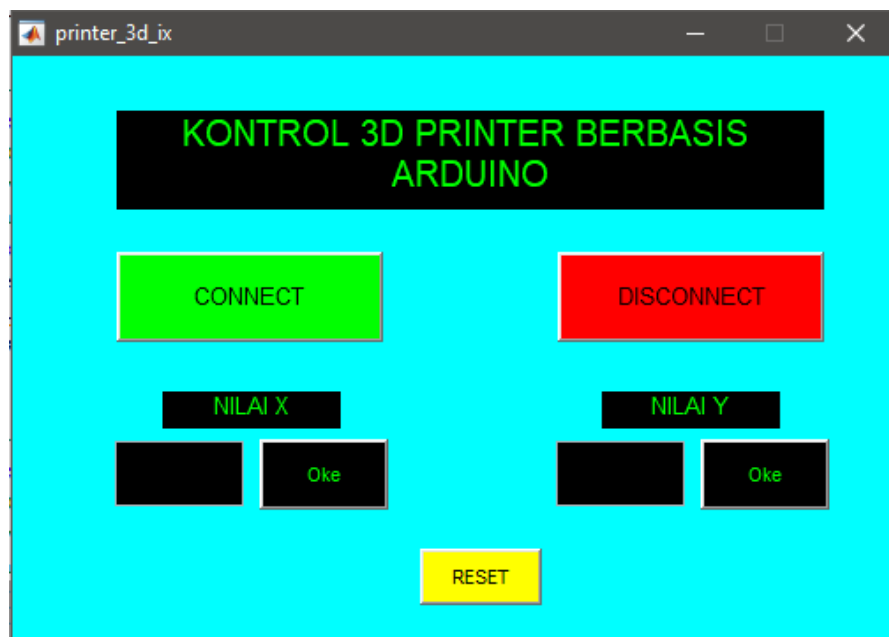
```

set(handles.x,'String','0')
set(handles.y,'String','0')

```

} Untuk *reset* nilai x dan y.

Dari program pengujian pada *Graphical User Interface (GUI)* Matlab tampilan dapat dilihat pada Gambar 4.12.



Gambar 4.12 Tampilan pengujian GUI Matlab

Terdapat beberapa menu yang disediakan seperti tombol *connect* dan *disconnect* yang berfungsi untuk menghubungkan dan memutuskan koneksi antara Arduino dan Matlab. Serta menu untuk mengatur nilai X dan nilai Y yang

diinginkan, dan terakhir tombol *reset* yang berfungsi untuk mengembalikan nilai X dan nilai Y menjadi nol.

4.5 Pengujian Keseluruhan Kontrol 3D *Printer* Berbasis Arduino

Pengujian dilakukan dalam bentuk kalibrasi parameter dengan mencetak. Hasil yang sudah tercetak kemudian diukur dan dilihat secara visual tampilan luarnya. Jika hasil pengukuran maupun hasil secara visual tidak sesuai maka, parameter diubah sampai didapat hasil yang sesuai. Berikut pengujian yang dilakukan beserta hasil pengujiannya.

Tabel 4.3 Pengujian dengan parameter yang pertama

Parameter yang diatur	Hasil
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 100 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 1 mm 	Filamen yang keluar kurang baik dan terlalu cepat mengeras, filamen tidak menempel pada alas cetak.
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 125 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 1 mm 	Filamen yang keluar kurang baik dan terlalu cepat mengeras, filamen tidak menempel pada alas cetak.
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 150 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 1 mm 	Filamen yang keluar sudah sesuai dengan yang diinginkan, filamen tidak menempel pada alas cetak.

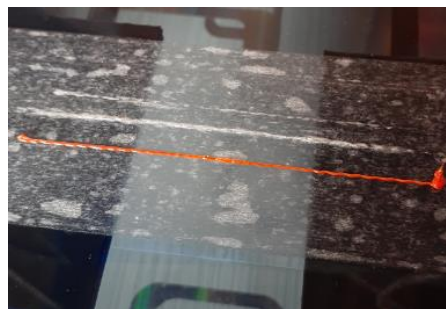
Parameter suhu pada *nozzle* sangat berpengaruh pada hasil cetakan, suhu yang terlalu rendah akan membuat filamen cepat mengeras sehingga hasil cetakan tidak menempel pada alas cetak. Sedangkan apabila suhu terlalu tinggi akan berpengaruh pada *hot end* yang berfungsi sebagai penyerap panas *nozzle*, kemampuan *hot end* akan berkurang apabila suhu *nozzle* terlalu tinggi sehingga *nozzle* akan mengalami *over heat*.

Parameter jarak antara ujung *nozzle* ke alas cetak juga berpengaruh pada kemampuan menempelnya. Secara ideal diameter keluaran plastik dari *nozzle* sama dengan diameter ujung *nozzle*. Jenis alas cetak yang digunakan juga

berpengaruh pada kemampuan menempelnya filamen. Alas cetak *heat bed* biasanya paling banyak digunakan karena *heat bed* dapat menghantarkan panas. Beda dengan alas cetak dari jenis akrilik ataupun kaca yang tidak dapat menghantarkan panas.

Tabel 4.4 Pengujian dengan parameter yang kedua

Parameter yang diatur	Hasil
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 150 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 0,80 mm 	Filamen tidak menempel dengan baik pada alas cetak.
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 150 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 0,50 mm 	Filamen tidak menempel dengan baik pada alas cetak.
<ul style="list-style-type: none"> • Suhu <i>nozzle</i> = 150 °C • Jarak antara ujung <i>nozzle</i> dan alas cetak = 0,25 mm 	Filamen menempel pada alas cetak sesuai dengan hasil yang diinginkan.



Gambar 4.13 Hasil cetakan dari parameter suhu 150°C dan jarak 0,25 mm

Setelah dilakukan pengujian dengan parameter kedua, suhu pada *nozzle* yang baik ada pada kisaran 150°C sedangkan untuk jarak antara alas cetak dengan ujung *nozzle* yang baik ada pada jarak 0,25 mm. Apabila jarak lebih kecil dari 0,25 filamen sulit untuk keluar dan dapat menyebabkan *nozzle* menjadi buntu karena adanya gumpalan filamen yang menumpuk dalam *nozzle*. Pada Proyek

Akhir kali ini menggunakan alas cetak dari kaca dimana filamen sulit untuk menempel jika tidak dibantu dengan perekat seperti lem kertas atau *double tape*. Kemudian dilakukan pengecekan apakah alat yang dibuat memiliki akurasi, resolusi dan reproductibilitas yang baik atau tidak. Cara yang digunakan untuk mengecek akurasi, resolusi dan reproductibilitas yaitu dengan mencetak banyak garis dengan ukuran yang sama, perhitungan menggunakan hasil pencetakan sumbu x.

Tabel 4.5 Uji coba cetak garis 100 mm

Cetak garis 100 mm	Hasil
Garis 1	90,40 mm
Garis 2	90,40 mm
Garis 3	90,40 mm
Garis 4	90,60 mm
Garis 5	90,50 mm

Akurasi adalah selisih nilai target terhadap nilai percobaan terjauh. Dari 5 percobaan yang terjauh adalah percobaan 1,2 dan 3 yaitu 90,40 mm sehingga:

$$\begin{aligned} \text{Akurasi} &= 100 - 90,60 \\ &= 9,4 \end{aligned}$$

Jadi tingkat akurasi dari alat yang telah dibuat adalah sebesar $\pm 9,4$ mm.

Resolusi adalah kemampuan terkecil dari pergerakan alat yang telah dibuat. Berdasarkan spesifikasi motor *stepper* dapat dilihat satu putaran penuh sumbu x bergerak sejauh 32 mm. *Step* motor *stepper* untuk satu putaran penuh yaitu 200 *step* sehingga:

$$\begin{aligned} \text{Resolusi} &= \frac{32}{200} \\ &= 0,16 \end{aligned}$$

Jadi resolusi dari alat yang telah dibuat adalah sebesar 0,16 mm.

Repitabilitas yaitu selisih terbesar dari percobaan yang dilakukan, sehingga:

$$\begin{aligned}\text{Repitabilitas} &= 90,60 - 90,40 \\ &= 0,2 \text{ mm}\end{aligned}$$

Dari hasil pengujian yang telah dilakukan, didapat bahwa alat yang telah dibuat dapat menjadi media pembelajaran bagi mahasiswa. Tetapi alat tersebut memiliki akurasi yang rendah yaitu sebesar $\pm 9,4$ mm. Hal ini dikarenakan motor *stepper* yang digunakan memiliki torsi yang kecil sehingga proses penarikan filamen dari *extruder* menjadi tersendat. Akibatnya membuat filamen yang tercetak tidak memenuhi nilai yang diinginkan.

BAB V

KESIMPULAN DAN SARAN

Setelah dilakukannya pembuatan dan pengujian sampai selesainya alat ini dibuat, kesimpulan dan saran yang dapat diambil yaitu:

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari pembuatan Proyek Akhir ini adalah:

1. Alat ini dapat berfungsi baik dengan parameter suhu *nozzle* pada kisaran 150°C dan jarak antara ujung *nozzle* dan alas cetak adalah 0,25 mm.
2. Hasil cetak pada alat ini memiliki tingkat akurasi sebesar $\pm 9,4$ mm dengan resolusi sebesar 0,16 mm dan repetabilitas sebesar 0,2 mm.
3. Penggunaan alas cetak dari kaca kurang efisien karena perlu penambahan perekat sebelum mencetak agar filamen menempel.

5.2 Saran

Jika Proyek Akhir ini akan dikembangkan saran yang diberikan yaitu:

1. Mengganti *limit switch* menjadi sensor *proximity*, dikarenakan pegas pada *limit switch* tidak tahan apabila sering tertekan.
2. Memasang kapasitor secara paralel pada vin +12 volt motor *stepper* untuk meminimalisir tegangan transien yang muncul pada saat *switching*.
3. Aplikasi *input* nilai cetak yang semula hanya 2 kolom *input* x dan y ditambah menjadi 4 kolom *input* menjadi x1, y1, x2 dan y2 agar 3D *printer* dapat mencetak dalam bentuk kotak.
4. Alas cetak sebaiknya diganti dengan *heated bed* karena dapat menghantarkan panas sehingga tidak memerlukan perekat tambahan agar filamen menempel.

DAFTAR PUSTAKA

- [1] D. Sumantri, "Gandjar Kiswanto, dkk, Pengembangan Mesin Rapid Prototyping Berbasis Fdm (Fused Deposition Modeling) Untuk Produk Berkontur Dan Prismatic, 2010, Seminar Nasional Tahunan Teknik Mesin (SNTTM) ke-9 Palembang.," pp. 1-2, 2012.
- [2] A. S. B. R. K. A. S. S. H. M. Gandjar Kiswanto, "PENGEMBANGAN MESIN RAPID PROTOTYPING BERBASIS FDM (FUSED DEPOSITION MODELING) UNTUK PRODUK BERKONTUR DAN PRISMATIK," p. 123, 2010.
- [3] J. K. McMahon, "Pencetakan 3D," [Online]. Available: https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/3D_printing&hl=id&sl=en&tl=id&client=srp&prev=search. [Accessed 17 November 2020].
- [4] P. D. GROUP, "Mesin 3D Printer," PT. DEPRINTZ GROUP, [Online]. Available: <https://jualmesinprinter3d.com/prinsip-dasar-dan-cara-kerja-mesin3dprinter/#:~:text=Prinsip%20utama%20untuk%20pencetakan%203D,menampilkan%20data%20dalam%203%20dimensi..> [Accessed 17 November 2020].
- [5] I. Bagus, "Pengertian 3D Printer FDM (Fused Deposition Modeling)," 2018.[Online].Available:<https://www.idekubagus.com/2018/02/pengertian-3d-printer-fdm-fused.html>. [Accessed 17 November 2020].
- [6] L. Aditya, "PROTOTIPE 3D PRINTER BERBASIS MIKROKONTROLER ARDUINO MEGA 2560," pp. 9-21, 2019.
- [7] "NEMA 17 Stepper Motor," Components 101, 19 Agustus 2019. [Online]. Available: <https://components101.com/motors/Nema17-stepper->

motor. [Accessed 11 Januari 2021].

- [8] "A4988 *Stepper* Motor Driver Module," Components 101, 22 Agustus 2019. [Online]. Available: <https://components101.com/modules/a4988-stepper-motor-driver-module>. [Accessed 11 Januari 2021].
- [9] "Reprap Hotend Thermistor NTC 3950 100K with 1M Cable," Makeralot, 04 januari 2016. [Online]. Available: <https://www.makeralot.com/reprap-hotend-thermistor-ntc-3950-100k-with-1m-cable-p176/>. [Accessed 11 Januari 2021].
- [10] "12V 40W Ceramic Heater Cartridge," Hand On Tech, [Online]. Available: <https://handsontec.com/index.php/product/12v-40w-ceramic-heater-cartridge/>. [Accessed 11 Januari 2021].
- [11] "1 Channel 5V Relay Module with Optocoupler," Electronicscomp, [Online]. Available: <https://www.electronicscomp.com/1-channel-5v-relay-module>. [Accessed 11 Januari 2021].

LAMPIRAN

LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Muhammad Bobby Kusnadi
Tempat, Tanggal lahir : Pangkalpinang, 26 Mei 1999
Alamat rumah : JL. LetkolSaleh Ode Mo.18
Pangkalpinang
Telp : -
Hp : 082177576708
Email : muhammadbobby1996@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 60 Pangkalpinang (2005 - 2011)
SMP Negeri 4 Pangkalpinang (2011 - 2014)
SMK Negeri 2 Pangkalpinang (2014 - 2017)

3. Pendidikan Non Formal

-

Sungailiat, 15 Februari 2021



Muhammad Bobby Kusnadi

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Reza Pahlevi
Tempat, Tanggal lahir : Balunijuk, 01 Oktober 1999
Alamat rumah : Gg. Bukit 3 RT. 04 Blok A No.
81 Desa Balunijuk
Telp : -
Hp : 081299691275
Email : rpahlevi1324@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 15 Balunijuk (2005 - 2011)
SMP Negeri 2 Merawang (2011 - 2014)
SMK Negeri 2 Pangkalpinang (2014 - 2017)

3. Pendidikan Non Formal

-

Sungailiat, 15 Februari 2021



Reza Pahlevi

LAMPIRAN 2

PROGRAM ARDUINO

```
#define DirPinZA 3
#define StepPinZA 2
#define DirPinYA 5
#define StepPinYA 4
#define DirPinXA 6
#define StepPinXA 7
#define DirPinXT 9
#define StepPinXT 8
#define Relay 28
#define ThermistorPin A0

int j_xa = 0;
int j_ya = 0;

int stepTommXA = 0;
int stepTommYA = 0;

int step_program = 0;

int Vo;

float R1 = 10000;

float logR2, R2, T, Tc;

float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 =
2.019202697e-07;

void setup()
{
  pinMode(StepPinXA, OUTPUT);
  pinMode(DirPinXA, OUTPUT);
  pinMode(StepPinZA, OUTPUT);
  pinMode(DirPinZA, OUTPUT);
  pinMode(StepPinYA, OUTPUT);
  pinMode(DirPinYA, OUTPUT);
  pinMode(StepPinXT, OUTPUT);
  pinMode(DirPinXT, OUTPUT);
}
```

```
pinMode(22, INPUT_PULLUP);
pinMode(24, INPUT_PULLUP);
pinMode(26, INPUT_PULLUP);
pinMode(28, OUTPUT);
Serial.begin(9600);
}

void loop()
{
  while (step_program == 0)
  {
    input();
  }
  while (step_program == 1)
  {
    start_XA_ZA();
  }
  while (step_program == 2)
  {
    pemanas();
    if(Tc>140)
    delay(15000);
    {
      cetak_X();
      delay(1000);
      cetak_Y();
    }
    reset_ZA();
  }
  while (step_program == 3)
```

```

    {
        reset_XA_ZA();
    }
}

void input()
{
    Serial.println("Masukkan Nilai X Sumbu (mm) - Panjang Maksimal
200 mm");

    while (Serial.available() == 0) {}

    j_xa = Serial.parseInt();
    stepTOmmXA = j_xa * 6.25;
    Serial.print("X Sumbu (mm)= ");
    Serial.println(stepTOmmXA);

    delay(500);

    Serial.println("Masukkan Nilai Y Sumbu (mm) - Panjang Maksimal
130 mm");

    while (Serial.available() == 1) {}

    j_ya = Serial.parseInt();
    stepTOmmYA = j_ya * 6.25;
    Serial.print("Y Sumbu (mm)= ");
    Serial.println(stepTOmmYA);

    delay(500);

    step_program=1;
}

void start_XA_ZA()
{
    digitalWrite(DirPinXA, LOW);
    digitalWrite(DirPinZA, LOW);
    for(int x = 0; x < 20000; x++)
    {

```

```

if(x < 1370)
{
    digitalWrite (StepPinXA, HIGH);
    digitalWrite (StepPinZA, HIGH);
    delayMicroseconds (1000);
    digitalWrite (StepPinXA, LOW);
    digitalWrite (StepPinZA, LOW);
    delayMicroseconds (1000);
}
else
{
    digitalWrite (StepPinZA, HIGH);
    delayMicroseconds (1000);
    digitalWrite (StepPinZA, LOW);
    delayMicroseconds (1000);
}
int LM_ZA = digitalRead(26);
if (LM_ZA == HIGH)
{
    digitalWrite (StepPinZA, LOW);
    digitalWrite (StepPinZA, LOW);
    delayMicroseconds (1000);
    x=20000;
    step_program=2;
}
}

void reset_XA_ZA()
{

```

```

digitalWrite (DirPinXA, HIGH);
digitalWrite (DirPinZA, HIGH);
for (int x = 0; x < 5000; x++)
{
    if (x < 4000)
    {
        digitalWrite (StepPinXA, HIGH);
        digitalWrite (StepPinZA, HIGH);
        delayMicroseconds (1000);
        digitalWrite (StepPinXA, LOW);
        digitalWrite (StepPinZA, LOW);
        delayMicroseconds (1000);
    }
    int LM_XA = digitalRead (22);
    if (LM_XA == HIGH)
    {
        digitalWrite (StepPinXA, LOW);
        digitalWrite (StepPinXA, LOW);
        digitalWrite (StepPinZA, HIGH);
        delayMicroseconds (1000);
        digitalWrite (StepPinZA, LOW);
        delayMicroseconds (1000);
        if (x < 4000)
        {
            x = 4000;
        }
    }
}

digitalWrite (StepPinXA, LOW);
digitalWrite (StepPinXA, LOW);

```



```
        delayMicroseconds(1000);
        digitalWrite(StepPinZA, LOW);
        digitalWrite(StepPinZA, LOW);
        delayMicroseconds(1000);
    }
```

```
void reset_ZA()
```

```
{
    digitalWrite(DirPinZA, HIGH);
    for(int x = 0; x < 2000; x++) {
        digitalWrite(StepPinZA, HIGH);
        delayMicroseconds(1000);
        digitalWrite(StepPinZA, LOW);
        delayMicroseconds(1000);
        step_program=3;
    }
}
```

```
void cetak_X()
```

```
{
    digitalWrite(DirPinXA, HIGH);
    digitalWrite(DirPinXT, HIGH);
    for(int x = 0; x < stepTommXA; x++)
    {
        digitalWrite(StepPinXA, HIGH);
        digitalWrite(StepPinXT, HIGH);
        delayMicroseconds(1500);
        digitalWrite(StepPinXA, LOW);
        digitalWrite(StepPinXT, LOW);
        delayMicroseconds(1500);
    }
}
```

```

    }
}

void cetak_Y()
{
    digitalWrite(DirPinYA, HIGH);
    digitalWrite(DirPinXT, HIGH);
    for(int x = 0; x < stepTommYA; x++)
    {
        digitalWrite(StepPinYA, HIGH);
        digitalWrite(StepPinXT, HIGH);
        delayMicroseconds(2000);
        digitalWrite(StepPinYA, LOW);
        digitalWrite(StepPinXT, LOW);
        delayMicroseconds(2000);
    }
    digitalWrite(DirPinXT, LOW);
    for(int x = 0; x < 100; x++)
    {
        digitalWrite(StepPinXT, HIGH);
        delayMicroseconds(1500);
        digitalWrite(StepPinXT, LOW);
        delayMicroseconds(1500);
    }
}

void pemanas ()
{
    Vo = analogRead(ThermistorPin);
    R2 = R1 * (1023.0 / (float)Vo - 1.0);
}

```

```
logR2 = log(R2);
T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
Tc = T - 273.15;

Serial.print("Temperature: ");
Serial.print(Tc);
Serial.println(" C");
delay(1000);
int RELAY = digitalRead(28);
if (Tc < 90)
{
    digitalWrite(RELAY, LOW);
    Serial.print("Relay ON");
}
else if (Tc > 150)
{
    digitalWrite(RELAY, HIGH);
    Serial.print("Relay OFF");
}
delay(1000);
}
```

LAMPIRAN 3

PROGRAM MATLAB

```

function varargout = printer_3d_ix(varargin)
% PRINTER_3D_IX MATLAB code for printer_3d_ix.fig
%     PRINTER_3D_IX, by itself, creates a new PRINTER_3D_IX or
raises the existing
%     singleton*.
%
%     H = PRINTER_3D_IX returns the handle to a new PRINTER_3D_IX
or the handle to
%     the existing singleton*.
%
%     PRINTER_3D_IX('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in PRINTER_3D_IX.M with the given
input arguments.
%
%     PRINTER_3D_IX('Property','Value',...) creates a new
PRINTER_3D_IX or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before printer_3d_ix_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to printer_3d_ix_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help printer_3d_ix

% Last Modified by GUIDE v2.5 10-Feb-2021 21:36:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @printer_3d_ix_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @printer_3d_ix_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before printer_3d_ix is made visible.
function printer_3d_ix_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to printer_3d_ix (see
VARARGIN)

% Choose default command line output for printer_3d_ix
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes printer_3d_ix wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = printer_3d_ix_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in connect.
function connect_Callback(hObject, eventdata, handles)
% hObject    handle to connect (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s;
s = serial('COM3','Baudrate',9600);
fopen(s);
set(handles.disconnect,'Enable','on');
set(handles.connect,'Enable','off');

% --- Executes on button press in disconnect.
function disconnect_Callback(hObject, eventdata, handles)
% hObject    handle to disconnect (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

global s;
fclose(s);
instrreset;
set(handles.disconnect,'Enable','off');
set(handles.connect,'Enable','on');

function x_Callback(hObject, eventdata, handles)
% hObject    handle to x (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of x as text
%        str2double(get(hObject,'String')) returns contents of x
as a double

% --- Executes during object creation, after setting all
properties.
function x_CreateFcn(hObject, eventdata, handles)
% hObject    handle to x (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in okex.
function okex_Callback(hObject, eventdata, handles)
% hObject    handle to okex (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s;
a=get(handles.x,'String');
fprintf(s,a)
pause(0.001)

function y_Callback(hObject, eventdata, handles)
% hObject    handle to y (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of y as text

```

```

%          str2double(get(hObject,'String')) returns contents of y
as a double

% --- Executes during object creation, after setting all
properties.
function y_CreateFcn(hObject, eventdata, handles)
% hObject    handle to y (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in okey.
function okey_Callback(hObject, eventdata, handles)
% hObject    handle to okey (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s;
b=get(handles.y,'String');
fprintf(s,b)
pause(0.001)

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.x,'String','0')
set(handles.y,'String','0')

```