



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

**MENGATUR KECEPATAN MOTOR DC *SPINDLE*
BERBASIS PWM (*Pulse Width Modulation*) pada
MESIN CNC *PORTABLE***

Fahmi Addinul Haq
NRP. 2214 030 043

Dosen Pembimbing

1. Ir. Djoko Suprajitno Rahardjo, M.T
2. Andri Ashfahani, ST., M.Sc

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

Spindle DC Motor Speed Control Based PWM (Pulse Width Modulation) in Portable CNC machine

Fahmi Addinul Haq
NRP. 2214 030 043

Advisor

1. Ir. Djoko Suprajitno Raharjo, M.T
2. Andri Ashfahani, ST., M.Sc

***COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**MENGATUR KECEPATAN MOTOR DC SPINDLE BERBASIS PWM (Pulse Width Modulation) pada MESIN CNC PORTABLE**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017



Fahmi Addinul Haq
NRP. 2214 030 043

(halaman ini sengaja dikosongkan)

**MENGATUR KECEPATAN MOTOR DC SPINDLE BERBASIS
PWM (PULSE WIDTH MODULATION) PADA MESIN CNC
PORTABLE**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Ahli Madya
Pada

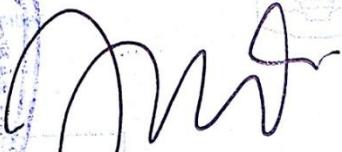
Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,

Dosen Pembimbing II,


Ir. Djoko Suprajitno Raharjo, M.T
NIP. 195506221987011001


Andri Ashfahani, ST., M.Sc
NIP. 2200201405003

**SURABAYA
JUNI, 2017**

(halaman ini sengaja dikosongkan)

MENGATUR KECEPATAN MOTOR DC *SPINDLE* BERBASIS PWM (*Pulse Width Modulation*) pada MESIN CNC *PORTABLE*

Nama : Fahmi Addinul Haq
NRP : 2214 030 043

Pembimbing I : Ir. Djoko Suprajitno Rahadjo, M.T
NIP : 195506221987011001

Pembimbing II : Andri Ashfahani, ST., M.Sc
NIP : 2200201405003

ABSTRAK

Mesin CNC merupakan mesin yang memiliki produktifitas tinggi dan ketelitian pengerjaan tinggi. Keuntungan menggunakan mesin CNC adalah waktu produksi lebih cepat, kapasitas produksi lebih besar, dan biaya pembuatan produk lebih murah.. Pada prakteknya, saat melakukan pengeboran tingkat kedalaman pahat selalu berubah-ubah sesuai dengan proses yang dilakukan. Hal tersebut sangat mempengaruhi kecepatan putar motor DC *spindle* yang mengakibatkan tingkat kepresisian menjadi berkurang. Oleh karena itu, perlu dijaga kecepatan putar dari motor bor *spindle* saat melakukan pengeboran.

Untuk meningkatkan kinerja dari motor DC spindle agar tetap stabil, dibuat lah sistem kendali kecepatan motor DC berbasis PWM pada mikrokontroller Arduino. PWM (*Pulse Width Modulation*) yaitu dengan mengatur durasi waktu tunda dari pulsa yg diumpun ke rangkaian *driver* motor. Penentuan parameter dari nilai PWM dilakukan sesuai nilai *setpoint* yang dimasukkan. Respon dari kecepatan motor DC *spindle* dapat dilihat pada *LabVIEW*.

Pada tugas akhir ini telah dibuat fungsi PWM pada arduino diprogram otomatis untuk mencapai *setpoint*. Hasil pengujian menunjukkan kecepatan maksimal motor DC *spindle* 12.170 rpm menggunakan sensor *optocoupler*. Kecepatan motor DC *spindel* pada *range* 1000-5000 rpm membutuhkan *risetime* rata-rata 24 detik untuk mencapai *setpoint*.

Kata kunci : CNC, Motor DC, arduino, PWM

(halaman ini sengaja dikosongkan)

Spindle DC Motor Speed Control Based PWM (Pulse Width Modulation) in Portable CNC machine

Nama : Fahmi Addinul Haq
Register Number : 2214 030 043

Supervisor I : Ir Ir. Djoko Suprajitno Rahadjo, M.T
ID Number : 195506221987011001

Supervisor II : Andri Ashfahani, ST., M.Sc
ID Number : 2200201405003

ABSTRACT

CNC machine is a machine that has high productivity and high accuracy of workmanship. The advantage of using CNC machines is faster production time, larger production capacity, and cheaper product-making costs. In practice, when drilling the depth of the cutting tool is always changing according to the process. This greatly affects the rotational speed of the DC spindle motor resulting in reduced precision. Therefore, it is necessary to maintain the rotational speed of the spindle drill motor during drilling.

To improve the performance of the DC spindle motor to remain stable, a PWM-based DC motor speed control system was developed on the Arduino microcontroller. PWM (Pulse Width Modulation) is by adjusting the duration of time delay of the pulse fed to the motor driver circuit. The parameter determination of the PWM value is performed according to the setpoint value entered. Response of DC spindle motor speed can be seen in LabVIEW.

In this final project has been made PWM function in automatic programmed arduino to reach setpoint. Test results showed the maximum speed of 12,170 rpm DC spindle motor using optocoupler sensor. DC spindle motor speeds in the range of 1000-5000 rpm require averaging 24 seconds to reach the setpoint.

Keywords : CNC, Motor DC, arduino, PWM

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat menyusun dan menyelesaikan laporan Tugas Akhir dengan judul:

MENGATUR KECEPATAN MOTOR DC *SPINDLE* BERBASIS PWM (*Pulse Width Modulation*) pada MESIN CNC *PORTABLE*

Penulis ingin berterima kasih kepada beberapa pihak yang telah membantu dalam penyusunan dan penyelesaian dari Tugas Akhir ini, diantaranya:

1. Kedua orang tua, Ayahanda Suwito dan Muntianah serta kakak dan adek yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Wanita istimewa yang pernah hadir dalam kehidupan penulis Rizkyah Talha Assegaff sebagai teman, sahabat, kekasih, guru, murid yang selalu menemani penulis saat suka dan duka dan tiada henti untuk tetap menyemangati
3. Bapak Djoko Suprajitno Rahardjo, M.T. selaku Dosen Pembimbing I dan Bapak Andri Asfahani, ST, M.Sc. selaku Dosen Pembimbing II atas bantuan dan bimbingan hingga Tugas Akhir ini terselesaikan.
4. Para dosen penguji Tugas Akhir yang telah memberi masukan dan saran kepada penulis dalam penyelesaian Tugas Akhir.
5. Semua teman yang ada di Teknik D3 Teknik Elektro angkatan 2014 ANDROMEDA, khususnya anggota kelompok Tugas Akhir Rico, atas bantuan dan kerjasama yang telah diberikan.

Harapan besar penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan D3 Teknik Elektro pada khususnya. Penulis juga mengharapkan kritik dan saran atas buku Tugas Akhir ini karena penulis menyadari bahwa dalam penyusunan Tugas Akhir ini banyak terdapat kekurangan.

Surabaya, 19 Juli 2017

Penulis

(halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB 1	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Laporan.....	3
1.7. Relevansi.....	4
BAB 2	5
2.1. <i>Computer Numerical Control (CNC)</i>	5
2.2. Motor DC	6
2.3. Komponen Penyusun Mesin CNC <i>Milling</i>	8
2.3.1 Komponen Mekanik	9
2.3.2 Komponen Elektrik	12
2.4. PWM (Pulse Width Modulation).....	18
2.5. GRBL/CAD.....	20
2.6. <i>Software</i> Arduino IDE.....	20
2.7. LabVIEW(<i>Laboratory Virtual Instrumentation Engineering</i>)..	21
BAB 3	25
3.1. Gambaran Umum Sistem	25
3.2. Perancangan Perangkat Keras	25
3.2.1 Perancangan Mekanik.....	26
3.3.2 Kerangka Mesin CNC <i>Portable</i>	27
3.3.3 <i>Power Supply Switching</i>	31
3.3.4 <i>Wiring Driver MACH3 AL75</i> dengan Arduino	31
3.3.5 <i>Box</i> Kontrol.....	32
3.3.6 <i>Wiring Keypad 4x4</i> dengan Arduino	33

3.3.7	<i>Wiring LCD 16x2 dengan Arduino</i>	33
3.3.8	<i>Wiring Sensor Optocoupler</i>	34
3.3	Perancangan Perangkat Lunak.....	35
3.3.1	Program Sensor Kecepatan.....	35
3.3.2	Program <i>Keypad</i>	36
3.4.3	Program Mengatur PWM pada Arduino	38
3.4.4	Diagram Blok <i>LabVIEW</i>	40
BAB 4	45
4.1	Lingkungan Uji Coba	45
4.2	Pengujian Perangkat Keras	45
4.2.1	Mengukur Kecepatan Motor DC <i>Spindle</i> dengan <i>Optocoupler</i>	45
4.2.2	Mengukur Kecepatan Motor DC <i>Spindle</i> dengan <i>Tachometer</i>	47
4.2.3	Perbandingan Kecepatan Sensor <i>Optocoupler</i> dengan <i>Tachometer</i>	49
4.2.4	Pengujian <i>Output</i> PWM Mikrokontroler Arduino pada <i>Driver Motor</i>	51
4.2.4	Pengujian Menampilkan Karakter pada LCD 16x2	52
4.3	Pengujian Perangkat Lunak	52
4.3.1	Pengujian <i>Setpoint</i> 1000 RPM.....	53
4.3.2	Pengujian <i>Setpoint</i> 2000 RPM.....	54
4.3.3	Pengujian <i>Setpoint</i> 3000 RPM.....	55
4.3.4	Pengujian <i>Setpoint</i> 4000 RPM.....	56
4.3.5	Pengujian <i>Setpoint</i> 5000 RPM.....	58
BAB 5	61
5.1	Kesimpulan.....	61
5.2	Saran	61
DAFTAR PUSTAKA	63
LAMPIRAN A	65
•	Dokumentasi hasil pembuatan alat	65
LAMPIRAN B	67
•	Program Sensor Kecepatan.....	67
LAMPIRAN C	69
•	Program <i>Keypad</i>	69
LAMPIRAN D	73
•	Program Mengatur Kecepatan Motor DC.....	73
RIWAYAT HIDUP PENULIS	77

DAFTAR GAMBAR

Gambar 2.1 Mesin CNC <i>Milling</i> ^[5]	6
Gambar 2.2 Motor DC ^[2]	7
Gambar 2.3 Mekanisme Kerja Motor DC ^[2]	8
Gambar 2.4 Motor <i>Stepper</i> ^[9]	9
Gambar 2.5 Blok Diagram Kontrol Motor DC	10
Gambar 2.6 Motor DC <i>Spindle</i>	11
Gambar 2.7 Arduino Uno	12
Gambar 2.8 <i>Driver</i> Motor DC <i>Spindle</i>	14
Gambar 2.9 <i>Power Supply</i> 48V 10A.	15
Gambar 2.10 <i>Keypad</i> 4x4 ^[3]	15
Gambar 2.11 Sensor <i>Optocoupler</i> ^[12]	16
Gambar 2.12 LCD 16x2 ^[4]	17
Gambar 2.13 I ² C (<i>Inter Integrated Circuit</i>) ^[4]	18
Gambar 2.14 <i>Duty Cycle</i>	19
Gambar 2.15 Arduino IDE ^[1]	21
Gambar 2.16 Tampilan <i>Front Panel LabVIEW</i>	22
Gambar 2.17 Tampilan <i>Block Diagram</i>	23
Gambar 3.1 Diagram Alur Perancangan <i>System</i>	25
Gambar 3.2 Piringan 4 Lubang	26
Gambar 3.3 Pemasangan Piringan Pada Sensor <i>Optocoupler</i>	26
Gambar 3.4 Penggabungan Sensor <i>Optocoupler</i> dengan Motor DC ...	27
Gambar 3.5 Perancangan Kerangka Meja CNC <i>Portable</i>	27
Gambar 3.6 Perancangan Motor <i>Stepper</i> pada Sumbu Y Kanan	28
Gambar 3.7 Perancangan Motor <i>Stepper</i> pada Sumbu Y Kiri	28
Gambar 3.8 Meja Kerja pada Mesin CNC <i>Portable</i> Menggunakan Alas Kayu.	29
Gambar 3.9 Perancangan Letak Motor <i>Stepper</i> pada Sumbu X	30
Gambar 3.10 Perancangan Mesin CNC <i>Portable</i>	30
Gambar 3.11 <i>Power Supply Switching</i>	31
Gambar 3.12 <i>Wiring Driver</i> Motor MACH3 AL75	32
Gambar 3.13 Perancangan <i>Box</i> Kontrol	32
Gambar 3.14 <i>Wiring Keypad</i> dengan Arduino	33
Gambar 3.16 <i>Wiring Optocoupler</i> dengan Arduino	34
Gambar 3.17 <i>Flowchart</i> Pembacaan Sensor	35
Gambar 3.18 Program Tes Sensor <i>Optocoupler</i>	36
Gambar 3.19 Rangkaian <i>Scanning Keypad</i>	36
Gambar 3.20 <i>Flowchart Scanning Keypad</i>	37

Gambar 3.21 Program <i>Scanning Keypad</i>	38
Gambar 3.22 <i>Flowchart</i> Program Mengatur PWM pada Arduino	39
Gambar 3.23 Program Mengatur PWM pada Arduino.....	40
Gambar 3.24 Diagram Blok Sistem pada <i>LabVIEW</i>	40
Gambar 3.27 Inisialisasi Data Sensor pada <i>LabVIEW</i>	42
Gambar 3.28 Inisialisasi Data <i>Keypad</i> pada <i>LabVIEW</i>	43
Gambar 3.29 <i>Structures Case</i> Program LabVIEW	43
Gambar 4.1 Pengujian Sensor Optocoupler	46
Gambar 4.2 Grafik Pembacaan Sensor <i>Optocoupler</i>	47
Gambar 4.4 Pengujian Kecepatan Motor dengan <i>Tachometer</i>	49
Gambar 4.5 Perbandingan Pengukuran Kecepatan <i>Optocoupler</i> dengan <i>Tachometer</i>	49
Gambar 4.6 Pengujian <i>Output</i> PWM Mikrokontroler pada <i>Driver</i> Motor	51
Gambar 4.7 Menampilkan Karakter Angka pada LCD 16x2	52
Gambar 4.8 Tampilan <i>Front Panel LabVIEW</i> Kontrol Motor DC	52
Gambar 4.9 Hasil Pengujian <i>Rise Time</i> 1000 RPM	53
Gambar 4.10 Hasil pengujian <i>Rise Time</i> 2000 RPM.....	54
Gambar 4.11 Pengujian <i>Rise Time</i> Tepat pada <i>Setpoint</i> 2000 RPM	54
Gambar 4.12 Hasil Pengujian <i>Rise Time</i> 3000 RPM	55
Gambar 4.13 Pengujian <i>Rise Time</i> Tepat pada <i>Setpoint</i> 3000 RPM	56
Gambar 4.14 Hasil Pengujian <i>Rise Time</i> 4000 RPM	56
Gambar 4.15 Pengujian <i>Rise Time</i> Tepat pada <i>Setpoint</i> 4000 RPM	57
Gambar 4.16 Hasil Pengujian <i>Rise Time</i> 5000 RPM	58
Gambar 4.17 Pengujian <i>Rise Time</i> Tepat pada <i>Setpoint</i> 5000 RPM	58

DAFTAR TABEL

Tabel 2.1. <i>Name Plate</i> Motor DC <i>Spindle</i>	11
Tabel 2.2. Spesifikasi Arduino UNO.....	13
Tabel 2.4. Spesifikasi pin LCD 16x2 ^[4]	17
Tabel 4.1 Hasil Pengujian Sensor <i>Optocoupler</i>	46
Tabel 4.2 Hasil Pengujian Alat Ukur Kecepatan <i>Tachometer</i>	48
Tabel 4.3 Perbandingan Selisih Kecepatan pada <i>Tachometer</i> dan <i>Optocoupler</i>	50
Tabel 4.4 Hasil Pengujian Nilai <i>Digital PWM</i>	51
Tabel 4.5 Hasil Pengujian <i>Rise Time</i> 1000-5000 rpm.	59

(halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang pembuatan Tugas Akhir, perumusan masalah, batasan masalah, tujuan, sistematika penulisan dan relevansi.

1.1. Latar Belakang

Perkembangan zaman saat ini berpengaruh terhadap penemuan teknologi yang semakin maju. Terlebih di bidang industri, penggunaan komputer dan peralatan mekanik terbukti lebih efisien daripada bekerja secara manual menggunakan tenaga manusia. Perpaduan teknologi komputer dan peralatan mekanik ini menghasilkan suatu alat yang dinamakan CNC (*Computer Numerically Controlled*). Mesin CNC ini mampu bekerja secara otomatis sesuai yang kita inginkan sehingga hasil yang didapatkan lebih presisi daripada bekerja secara manual. Pemanfaatan teknologi ini berdampak terhadap pekerjaan manusia. Hal tersebut juga berimbas pula pada industri-industri kecil, menengah dan keatas, khususnya yang masih menggunakan peralatan konvensional atau bahkan masih menggunakan peralatan tradisional dan manual. Pemahaman teknologi secara mendasar dan mendalam dilakukan melalui pelaksanaan program yang tepat untuk memproduksi barang dan jasa sesuai dengan industrinya. Salah satunya adalah jenis mesin CNC milling.

Mesin CNC yang terdapat di industri biasanya berukuran besar karena objek kerjanya berukuran besar pula. Penggunaan mesin CNC tidak semuanya efisien diterapkan di industri. Untuk skala industri menengah dan kecil, mesin tersebut tidaklah efisien dikarenakan biayanya produksi yang mahal dan desain yang terlalu besar sehingga membutuhkan tempat yang luas. Persoalan ini dapat diatasi dengan membuat mesin CNC yang berukuran kecil sehingga dapat dibawa kemana-mana dan dapat dioperasikan kapanpun dan dimanapun kita menginginkannya.

Pada tugas akhir ini yang menjadi permasalahannya adalah pada saat melakukan pengeboran atau proses *face milling*. Dimana motor DC *spindle* mengenai benda kerja saat pertama kali. Pada kenyataannya saat melakukan proses pengeboran, tingkat kedalaman pahat akan selalu berubah-ubah sesuai dengan proses kerja yang dilakukan sehingga mempengaruhi kecepatan motor *spindle* dan mengakibatkan tingkat presisi hasil kerja menjadi berkurang. Proses pengeboran benda kerja

memerlukan kecepatan potong yang stabil agar hasil kerja memiliki tingkat presisi yang tinggi.

Agar dapat mewujudkan kebutuhan tersebut, dirancang sebuah mesin CNC *portable* jenis *milling* dengan motor *spindle* yang dapat diatur kecepatannya melalui PWM (*Pulse Width Modulation*) yang ada pada mikrokontroler. Metode ini merupakan salah satu cara paling mudah untuk membangkitkan sebuah tegangan analog dari sebuah nilai digital.

1.2. Perumusan Masalah

Tidak semua industri menggunakan mesin CNC untuk memproduksi sebuah produk, terutama pada industri kecil dan menengah. Karena terbatasnya lahan untuk menempatkan mesin CNC.

Mesin CNC *portable* memiliki tingkat kepresisian yang tinggi. Agar dapat menghasilkan benda yang halus, kecepatan motor DC *spindle* saat melakukan pemotongan harus dijaga konstan. Kecepatan motor DC *spindle* dapat berkurang dikarenakan naiknya torsi beban. Oleh karena itu diperlukan pengatur kecepatan motor DC *spindle* menggunakan PWM berbasis mikrokontroler arduino sehingga dihasilkan pengatur otomatis gerak motor DC *spindle* untuk memudahkan dalam mengontrol kecepatan.

1.3. Batasan Masalah

Mesin CNC *portable* ini menggunakan 3 Axis, yaitu X,Y,Z dan motor *spindle* untuk memotong dan mengebor benda kerja. Dimana mesin CNC *portable* ini berukuran 40x35 cm yang diharapkan bisa melakukan pemotongan objek berukuran 30 x 20 cm. Pada sumbu Z dipasang sebuah motor DC *spindle*. Tipe dari motor *spindle* menggunakan jenis motor DC. Untuk menjaga kestabilan dari motor *spindle* saat melakukan pemotongan dan pengeboran terhadap benda kerja maka dirancang sistem kendali kecepatan motor DC dengan metode PWM (*Pulse Width Modulation*) dari mikrokontroler arduino. Sensor yang digunakan merupakan sensor *optocoupler* yang akan membaca kecepatan putar dari motor DC *spindle*. Dengan adanya batasan masalah ini diharapkan hasil akhir dari Tugas Akhir ini dapat tercapai.

1.4. Tujuan Penelitian

Agar mesin CNC ini cocok diterapkan terutama pada industri kecil dan menengah, maka dibuat bentuk mesin CNC *portable* dimana fungsinya sama dengan mesin CNC yang lain. Mesin *portable* ini mudah dipindah tempat dan dibawa kemana-mana.

Mesin CNC *portable* ini dapat mengontrol atau menjaga kestabilan kecepatan motor *spindle* saat melakukan proses pemotongan dan pengeboran benda kerja.

1.5. Metodologi Penelitian

Dalam melakukan perancangan alat, terbagi menjadi empat tahap yang meliputi studi literatur, perancangan sistem, uji coba dan hasil pengujian, serta penyusunan laporan.

Pada tahap studi literatur, dilakukan pencarian literatur baik dari buku maupun kumpulan makalah dan jurnal yang mengarah pada topik tugas akhir, kegiatan tersebut dilakukan untuk mencari informasi dan spesifikasi tentang data-data mesin CNC *portable* yang akan dikontrol motor *spindle* nya.

Selanjutnya pada perancangan sistem, dibuat program melalui *software LabVIEW* dan Arduino untuk mengetahui perhitungan kecepatan motor *spindle*.

Setelah melalui tahap perancangan sistem, dilakukan uji coba. Pada proses ini, dilakukan uji coba panel untuk mengatur kecepatan motor *spindle* saat melakukan pengeboran terhadap benda kerja.

Tahap terakhir yaitu, penyusunan laporan. Dalam tahap ini, luaran yang diharapkan berupa sebuah laporan yang meliputi semua proses pengerjaan yang dilakukan dalam proses perancangan mengatur kecepatan dari motor *spindle*.

1.6. Sistematika Laporan

Sistematika penulisan pada laporan Tugas Akhir ini terdiri atas 5 bab, seperti yang dapat dilihat pada uraian berikut ini :

BAB 1 : PENDAHULUAN

Pada bab ini, akan dijelaskan mengenai latar belakang serta perumusan dan batasan masalah pada Tugas Akhir ini. Selain itu, akan dijabarkan pula tujuan dari Tugas Akhir ini beserta metodologi yang digunakan. Terakhir, akan dijelaskan pula mengenai sistematika penulisan dan relevansi Tugas Akhir ini.

BAB 2 : TINJAUAN PUSTAKA

Dasar pemikiran dan pengetahuan dari sistem yang akan dirancang seperti teori CNC, motor DC, PWM dan *hardware* dan *software* yang digunakan akan dibahas pada Bab ini.

BAB 3 : PERANCANGAN SISTEM

Bab ini membahas tentang perancangan sistem yang akan dibuat, perancangan perangkat keras (*hardware*), perancangan perangkat lunak (*software*)

BAB 4 : PENGUJIAN DAN ANALISA

Pada bab ini, akan dijabarkan mengenai hasil simulasi desain sistem pada *software* beserta analisisnya.

BAB 5 : PENUTUP

Bab terakhir ini akan menjelaskan tentang penarikan kesimpulan pelaksanaan Tugas Akhir serta kritik dan saran untuk penelitian selanjutnya.

1.7. Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat menjadi referensi untuk penelitian dengan fokus pada pengaturan kecepatan motor *spindle* pada mesin CNC *portable*. Selain itu penulis berharap penelitian ini dapat menjadi perbandingan metode pengaturan kecepatan motor DC spindle pada mesin CNC *portable* diwaktu yang akan datang.

BAB 2 TINJAUAN PUSTAKA

Pada Bab ini akan dibahas mengenai materi dasar dalam penyusunan Tugas Akhir. Beberapa hal yang dibahas meliputi tinjauan pustaka mengenai CNC, motor DC, PWM dan *hardware* maupun *software* yang digunakan.

2.1. *Computer Numerical Control* (CNC)

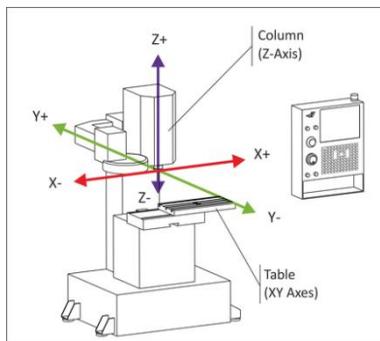
Adanya mesin CNC berawal dari berkembangnya sistem *Numerically Controlled* (NC) pada akhir tahun 1940-an dan awal tahun 1952-an yang ditemukan oleh John C. Parsons dengan bekerja sama dengan Perusahaan *Servomechanism Massachusetts Institute of Technology MIT*.^[7]

CNC sendiri merupakan mesin bantu berbasis mekatronika yang diaplikasikan untuk mengatur berbagai macam mesin, seperti mesin perkakas, fabrikasi, perkayuan, ukir, dan lain-lain. Penggunaan mesin ini dalam dunia industri makin pesat perkembangannya, karena ketidakmampuan mesin perkakas manual dalam menghasilkan produk massal dengan hasil konsisten, kuat, berkualitas, dan akurat. Pada dunia industri mesin ini banyak diaplikasikan pada berbagai sektor, mulai dari produksi bagianmesin, kelistrikan, hingga produksi mainan dan perabot rumah tangga. CNC ini merupakan mesin perkakas yang dilengkapi dengan sistem mekanik dan kontrol berbasis komputer yang mampu membaca instruksi kode angka, huruf dan simbol sesuai dengan standar ISO.^[11] Mesin ini dibedakan menjadi 2 jenis berdasarkan jumlah sumbu atau aksis yang dimiliki, yaitu jenis *Lathe turning* yang memiliki 2 aksis yaitu X dan Y, serta jenis *Milling* yang mempunyai 3 aksis yaitu X, Y dan Z. Dimana X dan Y berfungsi untuk membentuk kontur horizontal, vertikal sedangkan Z menentukan kedalaman potongan mata pahat terhadap benda kerja. Dapat dilihat pada Gambar 2.1^[5] Pergerakan pada aksis X, Y dilakukan oleh motor stepper dan Z dilakukan oleh penggerak spindle. Menurut cara kerjanya, kedua mesin CNC tersebut dibedakan berdasarkan cara kerja dari bor spindle. Untuk mesin CNC jenis lathe (bubut) motor bor spindle diam benda kerja di sumbu X dan Y bergerak. Sedangkan untuk jenis mesin CNC *milling* motor bor spindle dapat bergerak pada sumbu Z dan benda kerja bergerak pada sumbu X dan Y.

Hal yang harus dilakukan untuk mengoperasikan mesin CNC *milling* adalah dengan memberi data masukan berupa program. Berbagai

jenis program yang dapat digunakan untuk menjalankan mesin, namun terdapat salah satu bahasa pemrograman yang sering digunakan karena dianggap mudah bagi operator. *G-Code* adalah bahasa pemrograman tersebut, dimana berkaitan erat dengan grafik dan vektor yang memanfaatkan komputer sebagai alat bantu penghubung antara mesin dan perangkat lunak. Mesin ini akan mengikuti perintah berdasarkan gerak alur dari vektor yang dituliskan dalam program. Selain itu, bahasa ini juga memiliki kode-kode yang memiliki fungsi perintah masing-masing. Hal ini membuat mesin dapat bekerja dengan beragam fungsi

Berdasarkan sistem operasinya mesin *milling* dapat bekerja dengan 2 cara yaitu secara konvensional atau kendali manual dan pemrograman. Cara konvensional adalah metode mesin yang dioperasikan secara manual, menggunakan tombol kendali tangan. Sedangkan metode pemrograman dilakukan pada mesin dengan pergerakan meja dan pemotong yang dikendalikan oleh suatu program (menggunakan *G-Code*). Pembuatan program bisa dilakukan secara langsung pada sebuah PC/panel yang terkoneksi dengan mesin, atau membuat program diluar mesin dengan menggunakan *software* seperti *MasterCAM*, *GRBL*, maupun *MACH3*



Gambar 2.1 Mesin CNC *Milling*^[5]

2.2. Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Dapat dilihat pada Gambar 2.2 Motor DC memiliki 3 komponen utama untuk dapat berputar sebagai berikut :

1. Kutub medan.

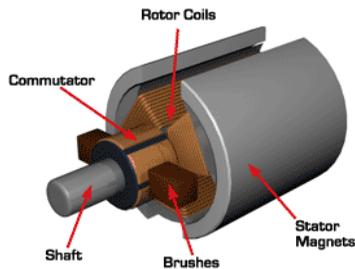
Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Digambarkan bahwa interaksi dua kutub magnet ini akan menyebabkan perputaran pada motor DC. Kutub medan pada motor DC yang stasioner akan menggerakkan bearing pada ruang diantara kutub medan. Garis magnetik energi melintasi diantara kutub utara ke selatan. Untuk motor yang lebih besar terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya luar sebagai penyedia struktur medan.

2. *Current* Elektromagnet atau Dinamo.

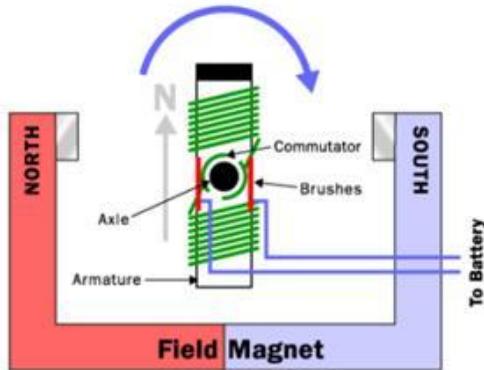
Arus yang mengalir pada dinamo motor DC akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi.

3. Commutator. Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk transmisi arus antara dinamo dan sumber daya.

Prinsip kerja motor DC secara sederhana adalah motor DC bekerja bila mendapatkan tegangan searah yang cukup pada kedua kutub nya. Kerja motor ini didasarkan pada gaya elektromagnetik, tegangan yang masuk pada motor DC ini akan menimbulkan induksi elektromagnetik sehingga menyebabkan motor berputar. Untuk meningkatkan kecepatan motor DC perlu juga meningkatkan tegangan yang yang diberikan. Secara umum, jika arah polaritas tegangan berubah maka putaran motor DC juga berubah.



Gambar 2.2 Motor DC^[2]



Gambar 2.3 Mekanisme Kerja Motor DC^[2]

Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak yang disebut komutator. Gambar 2.3. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor VTM.^[2] Metode lain yang biasa digunakan untuk mengendalikan kecepatan motor DC adalah dengan teknik modulasi lebar pulsa atau *Pulse Width Modulation (PWM)*.

2.3 Komponen Penyusun Mesin CNC Milling

Secara garis besar, bagian utama pada mesin CNC *milling* dibagi menjadi 2, yang pertama bagian mekanik, lalu bagian elektrik:

2.3.1 Komponen Mekanik

Komponen mekanik merupakan bagian pada mesin yang bergerak, terdiri dari:

a. Meja mesin

Pada mesin CNC jenis *milling* ini, dapat bergerak dalam 2 sumbu yaitu sumbu X, Y. Fungsi dari meja mesin pada CNC adalah untuk tempat kedudukan benda kerja yang akan dibor. Untuk masing-masing sumbu meja ini dilengkapi dengan motor stepper, *ball screw* dan biasanya pada mesin CNC *milling* ini perlu diberi pelumas pada *ball screw* yang bertujuan menjaga pergerakan meja agar lebih halus.

b. Motor *Stepper*^[9]

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkannya diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Dapat dilihat pada Gambar 2.4.

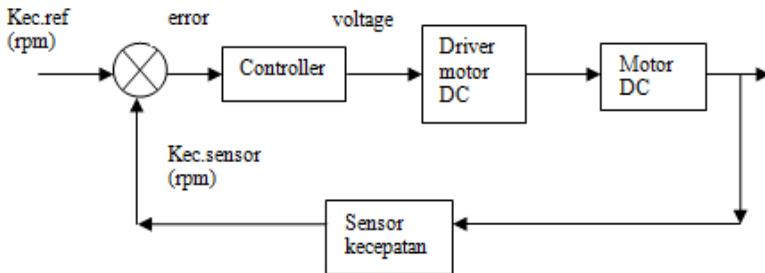
Prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor *stepper* mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor *stepper* adalah banyaknya fasa, besarnya nilai derajat per step, besarnya volt tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan.



Gambar 2.4 Motor *Stepper*^[9]

c. Motor DC *Spindle*

Motor *Spindle* merupakan alat yang digunakan untuk melakukan pemotongan maupun pengeboran pada benda kerja. Motor ini ditempatkan pada bagian axis Z dimana pergerakan naik turunnya axis Z digerakkan oleh motor stepper. Terdapat dua jenis motor yang bisa digunakan sebagai motor *spindle* yaitu motor DC atau motor AC. Masing-masing dari jenis motor tersebut baik motor DC maupun AC memiliki kelebihan dan kekurangan tersendiri. Motor AC memiliki kecepatan yang tinggi dan torsiya juga besar namun untuk pengaturan kecepatannya agak susah. Untuk motor DC kecepatannya tidak secepat motor AC namun untuk jenis *high speed* motor DC kuat untuk melakukan pemotongan maupun pengeboran dan lebih gampang diatur kecepatannya. Dapat dilihat pada Gambar 2.6.



Gambar 2.5 Blok Diagram Kontrol Motor DC.

Pada Gambar 2.5 merupakan diagram blok dari kontrol motor DC. Nilai kecepatan referensi menjadi acuan setpoint. Ketika masuk ke *controller*, tegangan akan aktif yang kemudian dikirimkan ke driver motor DC guna untuk menggerakkan motor DC. Ketika motor DC berputar terdapat sebuah sensor kecepatan untuk mengetahui kecepatan motor. Data kecepatan tersebut dikirimkan ke pembanding (*sum*) untuk dibandingkan dengan nilai kecepatan referensinya. Apabila belum mencapai nilai referensi maka *controller* akan mengatur otomatis yang menimbulkan kecepatan motor bertambah atau berkurang.

Motor DC banyak digunakan di berbagai bidang mulai dari peralatan industri sampai peralatan rumah tangga. Dengan adanya perkembangan teknologi elektronik sehingga memungkinkan dibuat perangkat pengendali dengan ukuran yang kecil akan tetapi memiliki kemampuan komputasi, kecepatan dan keandalan serta efisiensi daya yang tinggi.

Salah satu sistem kendali kecepatan motor dc adalah mengontrol kecepatan motor DC jarak jauh.^[10] Namun karena pengendalian tersebut menghasilkan efisiensi daya yang rendah serta kelebihan tegangan yang digunakan untuk menggerakkan motor di buang ke transistor. Untuk mengatasi permasalahan tersebut, dibuatlah sistem kendali kecepatan motor dc berbasis PWM. Dimana efisiensi daya dapat ditingkatkan karena tidak ada pembuangan daya ke transistor. Transistor bekerja dengan mode *on* atau *off* yang diatur periodenya secara PWM.



Gambar 2.6 Motor DC *Spindle*

Motor DC *spindle* yang dipakai dalam tugas akhir ini adalah di produksi secara custom yang menggunakan sikat. Berikut spesifikasinya pada Tabel 2.1.

Tabel 2.1. Name Plate Motor DC *Spindle*

<i>Spindle</i> Motor	<i>Brushless</i> DC Motor
Tegangan	12-48 V DC
Daya	300 Watt
Kecepatan	3000-12000r/min (12 V-3000 putaran 24 V-6000 putaran 36 V-9000 putaran 48 V-12000 putaran)
Torsi	400mN.m
Resistansi Insulalasi	>2 megaohms

Kekuatan <i>Dielectric</i>	400V
Diameter	52mm
Panjang <i>collet</i>	35mm
Diameter <i>collet</i>	16mm
Panjang Motor	175mm (termasuk penjepit motor dan motor)
<i>Spindle radial runout</i>	Rentang 0.01-0.04

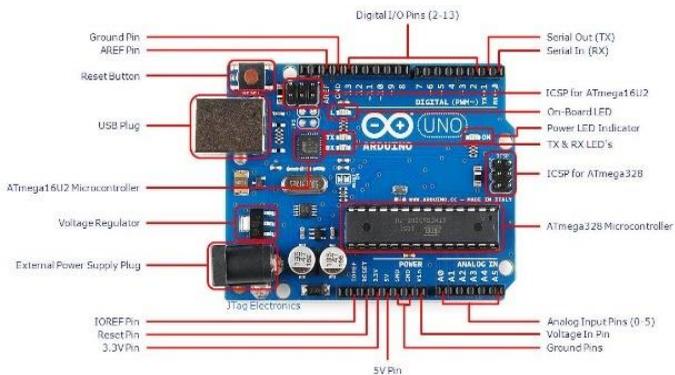
d. *Ballscrew* dan *fandbell*

Aktuator linier mekanik yang menerjemahkan gerak rotasi ke gerakan linier dengan sedikit gesekan. Sebuah poros berulir menyediakan jalur untuk bantalan bola yang bertindak sebagai sekrup presisi. Serta mampu menerapkan atau menahan beban dorong tinggi, mereka dapat melakukannya dengan gesekan internal minimum. Mereka dibuat untuk menutup toleransi dan karena itu cocok untuk digunakan dalam situasi di mana presisi tinggi diperlukan.

2.3.2 Komponen Elektrik

Komponen elektrik merupakan bagian yang berfungsi memberikan tenaga ke komponen mekanik supaya bergerak sesuai perintah *controller*, yaitu:

a. Arduino UNO^[1]



Gambar 2.7 Arduino Uno

Arduino UNO adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Dapat dilihat pada Gambar 2.7. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino.^[4]

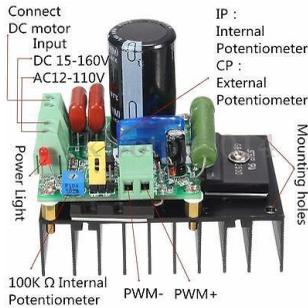
Arduino memakai IC mikro ATmega328 yang mempunyai 14 pin masukan dan keluaran digital (termasuk 6 diantaranya dapat berfungsi sebagai keluaran *Pulse Width Modulation*), 6 masukan analog, , dan sebuah tombol reset. Arduino Uno beroperasi pada tegangan 5 Volt. Lebih lengkap nya dapat dilihat pada Tabel 2.2.^[1]

Tabel 2.2. Spesifikasi Arduino UNO

Mikrokontroler	ATMega 328
Tegangan <i>input (recommended)</i>	7V-12V
Tegangan <i>input (limit)</i>	6V-20V
Pin digital I/O	14 (6 diantaranya pin PWM)
Pin analog <i>input</i>	6
Arus DC per pin I/O	40 mA
Arus DC untuk pin 3.3V	150 mA
<i>Flash Memory</i>	32 KB dengan 0.5 KB digunakan untuk <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Kecepatan Pewaktu	16 Hz

b. *Driver Motor*^[7]

Driver Motor merupakan alat elektronik yang terdiri dari berbagai IC dan komponen yang menjadi satu yang berfungsi untuk menggerakkan motor DC. Dapat dilihat pada Gambar 2.8. Alat ini menerima catu daya dari sumber luar sesuai spesifikasi dari *driver* motor tersebut.



Gambar 2.8 Driver Motor DC Spindle

Driver motor spindle ini dapat menggerakkan motor DC yang memiliki spesifikasi daya sampai dengan 400 watt. Sumber tegangan dari *driver* ini menggunakan power supply 48V 10A. Terdapat pula pin motor pada *driver* motor yang disambungkan dengan kabel motor dan pin pwm yang tersambung dengan kontroler. Berikut spesifikasi dari *driver spindle* motor yang digunakan pada tugas akhir ini dapat dilihat pada Tabel 2.3.

Tabel 2.3. Spesifikasi driver AL75

Input DC	5 – 11 V DC
Input AC	12 – 110 V AC
Duty cycle PWM input	0% - 93%

c. *Power Supply*^[8]

Power supply adalah perangkat keras berupa kotak yang isinya merupakan kabel-kabel untuk menyalurkan tegangan ke dalam perangkat keras lainnya. Perangkat keras ini biasanya terpasang di bagian belakang (di dalam) casing komputer. *Input power supply* berupa arus bolak-balik (AC) sehingga *power supply* harus mengubah tegangan AC menjadi DC (arus searah). Besarnya listrik yang mampu ditangani *power supply* ditentukan oleh dayanya dan dihitung dengan satuan *Watt*. *Power supply* berfungsi sebagai penyuplai tegangan listrik langsung kepada komponen-komponen yang berada di dalam casing komputer. Dapat dilihat pada Gambar 2.9. *Power Supply* juga berfungsi untuk mengubah tegangan AC menjadi DC, karena perangkat keras komputer hanya dapat beroperasi dengan arus DC. *Power supply* yang digunakan

pada tugas akhir ini memakai daya 480 watt dengan spesifikasi 48V 10 A yang mempunyai 3 *channel* pada masing V+ dan V-.



Gambar 2.9 Power Supply 48V 10A.

d. *Keypad* 4x4

Keypad adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (*Human Machine Interface*). *Keypad* yang dapat digunakan untuk berkomunikasi antara manusia dengan *mikrokontroler*. Matrix *keypad* 4x4 memiliki konstruksi atau susunan yang simple dan hemat dalam penggunaan port mikrokontroler. Dapat dilihat pada Gambar 2.10.



Gambar 2.10 Keypad 4x4^[3]

Konfigurasi *keypad* dengan susunan bentuk *matrix* ini bertujuan untuk penghematan *port mikrokontroler* karena jumlah *key* (tombol) yang dibutuhkan banyak pada suatu sistem dengan *mikrokontroler*.

e. Sensor *Optocoupler*

Sensor Kecepatan digunakan untuk mendeteksi kecepatan putar dari motor DC spindle. *Optocoupler* merupakan gabungan dari LED infra merah dengan fototransistor yang terbungkus menjadi satu chips. Dapat dilihat pada Gambar 2.11. Cahaya infra merah termasuk dalam gelombang elektromagnetik.^[12] Pada *optocoupler* yang bertugas sebagai penerima cahaya infra merah adalah fototransistor. Fototransistor merupakan komponen elektronika yang berfungsi sebagai detektor cahaya infra merah. Detektor cahaya ini mengubah efek cahaya menjadi sinyal listrik, oleh sebab itu fototransistor termasuk dalam golongan detektor optik. Sensor *optocoupler* terdiri dari dua bagian yaitu *transmitter* dan *receiver*. Pada bagian *transmitter* terdapat komponen IR Led yang memancarkan cahaya infra merah. Kemudian cahaya infra merah tersebut diterima untuk *receiver*. Prinsip kerja dari sensor ini yaitu jika antara *phototransistor* dan LED terhalang maka *phototransistor* tersebut akan off sehingga output dari kolektor akan berlogika *high*.



Gambar 2.11 Sensor *Optocoupler*^[12]

f. LCD (*Liquid Cristal Display*)^[4]

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada

di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. Spesifikasi PIN LCD ini dapat dilihat pada Tabel 2.4.



Gambar 2.12 LCD 16x2^[4]

LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. Dapat dilihat pada Gambar 2.12.

Tabel 2.4. Spesifikasi pin LCD 16x2^[4]

Kaki pin	Nama	Keterangan
1	GND	<i>Ground</i>
2	VCC	+5V
3	VEE	<i>Contras</i>
4	RS	<i>Register select</i>
5	RW	<i>Read/write</i>
6	E	<i>Enable</i>
7-14	D0-D7	Data 0-7
15	A	Anoda (back light)
16	K	Katoda (back light)

g. I2C (*Inter Integrated Circuit*)^[4]

Inter Integrated Circuit atau sering disebut I²C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Dapat dilihat pada Gambar 2.13. Sistem I²C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I²C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai transfer data pada I²C Bus dengan membentuk sinyal *Start*, mengakhiri transfer data dengan membentuk sinyal *Stop*, dan

membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master*.

Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”.

Dalam melakukan transfer data pada I2C *Bus*, kita harus mengikuti tata cara yang telah ditetapkan yaitu pertama transfer data hanya dapat dilakukan ketika *Bus* tidak dalam keadaan sibuk. Kedua selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal *start* atau sinyal *stop*.



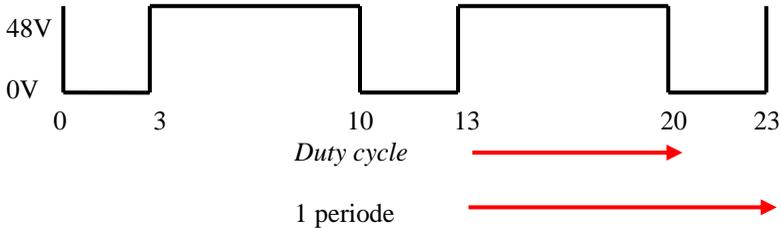
Gambar 2.13 I²C (*Inter Integrated Circuit*)^[4]

2.4 PWM (Pulse Width Modulation)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam 1 periode, untuk mendapatkan tegangan rata-rata yang berbeda. Teknik ini menggunakan dengan cara merubah-ubah besarnya *duty cycle* pulsa. Besarnya amplitudo dan frekuensi pulsa adalah tetap, sedangkan besarnya *duty cycle* berubah-ubah sesuai dengan kecepatan yang diinginkan, semakin besar *duty cycle* maka semakin cepat pula kecepatan motor, dan sebaliknya semakin kecil *duty cycle* maka semakin pelan pula kecepatan motor.^[10] Modulasi lebar pulsa (PWM) dicapai/diperoleh dengan bantuan sebuah gelombang kotak yang mana

siklus kerja (*duty cycle*) gelombang dapat diubah-ubah untuk mendapatkan sebuah tegangan keluaran yang bervariasi yang merupakan nilai rata-rata dari gelombang tersebut.

1 periode = terdiri dari sebuah bukit dan lembah.
Duty cycle = lamanya pulsa *high (on)* selama 1 periode



Gambar 2.14 *Duty Cycle*.

Dengan merubah nilai *duty cycle* pada *driver* motor, dengan begitu tegangan yang dialirkan pada motor dapat diatur. Untuk mencari nilai dari *duty cycle* gelombang pada Gambar 2.14 adalah :

$Duty\ cycle = (\text{interval\ pulsa\ high\ dalam\ 1\ periode/periode\ gelombang}) * 100\%$

$Duty\ cycle = \{(20-13)/(23-13)\} * 100\%$

$Duty\ cycle = \{7/10\} * 100\% = 70\%$

Duty Cycle merupakan representasi dari kondisi logika *high* dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan range 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi *high* terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan *high* sama dengan keadaan *low* maka sinyal mempunyai *duty cycle* sebesar 50%. Semakin tinggi frekuensi kerja PWM, maka akan semakin baik motor bekerja. Jika frekuensi kecil, maka akan terdengar dengungan saat motor dijalankan dengan *duty cycle* kecil, jika frekuensi semakin tinggi suara tersebut akan semakin mengecil. Dengan mengatur besarnya *duty cycle* pulsa kotak yang dikirimkan, kita dapat mengatur banyaknya logika *high* yang diberikan pada motor, dengan kata lain mengatur lamanya waktu motor untuk berputar dalam satu periode pulsa. Jika lamanya waktu motor untuk berputar dalam satu periode pulsa ini berubah maka kecepatan putaran motor juga akan

berubah, sesuai dengan *duty cycle* atau waktu motor untuk berputar dalam satu periode pulsa.

2.5 GRBL/CAD

GRBL/CAD merupakan sebuah *software* untuk membuat desain sebuah objek yang mampu dikembangkan oleh pengguna untuk melakukan analisis. Dengan *software* ini, pengguna mampu membuat sketsa ide serta bereksperimen dengan berbagai macam desain berbeda untuk membuat model 3D. Pengguna *software* ini beragam, dari berbagai macam kalangan mulai dari pelajar, teknisi, desainer dan kalangan profesional untuk keperluan mendesain *part* kompleks, *assembly* dan menggambar. Mendesain dengan menggunakan Solidworks mempunyai banyak keuntungan, salah satunya lebih menghemat waktu, tenaga dan biaya.

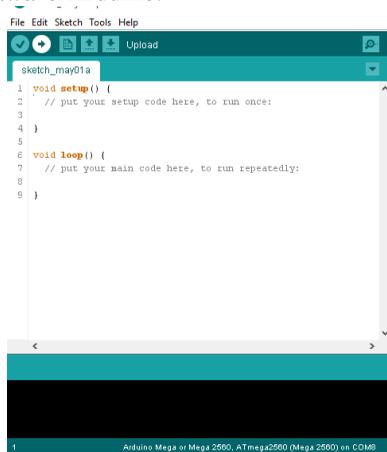
2.6 Software Arduino IDE

IDE merupakan kependekan dari *Integrated Development Environment* atau sebuah *software* yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam *memory microcontroller*. *Processing* adalah bahasa pemrograman yang digunakan untuk menulis program di dalam Arduino, bahasa pemrograman tingkat tinggi yang sangat mirip dengan C++ dan *Java*, sehingga pengguna yang sudah terbiasa dengan kedua bahasa tersebut dan tidak akan menemui kesulitan dengan *Processing*. Bahasa pemrograman *Processing* sangat memudahkan dan mempercepat pembuatan sebuah program karena bahasa ini sangat mudah dipelajari dan diaplikasikan dibandingkan bahasa pemrograman tingkat rendah seperti *Assembler* yang umum digunakan *pada platform* lain namun cukup sulit menjadi kode biner dan meng-upload ke dalam *memory microcontroller*. Dapat dilihat pada Gambar 2.15.

Berikut beberapa fungsi untuk menjalankan *software* arduino :

1.  *Verify*
Befungsi berfungsi untuk mengecek program yang ditulis apakah ada yang salah atau *error*.
2.  *Upload*
Befungsi untuk memuat atau mentransfer program yang dibuat di *software* Arduino ke *hardware* Arduino.

3.  *Create new project.*
Berfungsi untuk memulai sebuah proyek program.
4.  *Open*
Berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan *software* Arduino.
5.  *Save*
Berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi
6.  *Menu Serial Monitor*
Berfungsi mengirim atau menampilkan serial komunikasi data saat dikirim dari *hardware* Arduino.



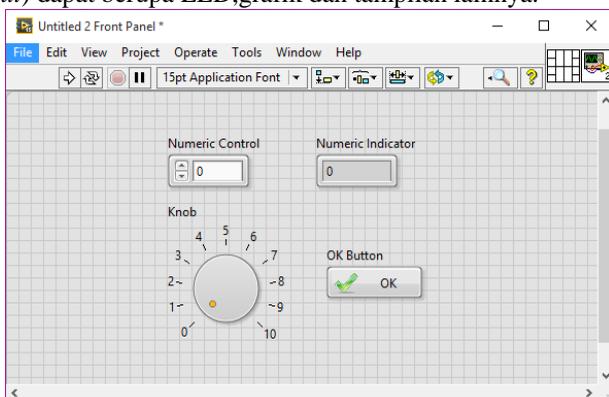
Gambar 2.15 Arduino IDE^[1]

2.7 LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*)

LabVIEW adalah suatu bahasa pemrograman berbasis grafis yang menggunakan *icon* sebagai ganti bentuk teks untuk menciptakan aplikasi. Berlawanan dengan bahasa pemrograman berbasis *text*, di mana instruksi menentukan pelaksanaan program, *LabVIEW* menggunakan pemrograman *data flow*, yang mana alur data menentukan pelaksanaan (*execution*). Tampilan pada *LabVIEW* menirukan *instrument* secara *virtual*. Dalam *LabVIEW*, membangun antarmuka

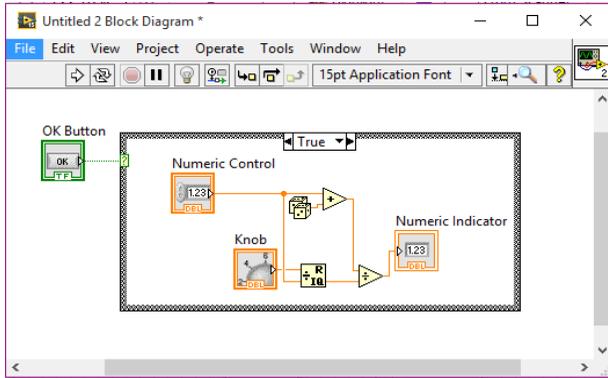
pemakai dengan satu set peralatan (*tools*) dan objek-objek. Antarmuka pemakai dikenal sebagai panel depan (*front Panel*). Selanjutnya menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan objek panel muka dengan *diagram blok* berisi kode ini. Program *LabVIEW* disebut sebagai *virtual instruments* atau VIs sebab operasi dan penampilannya meniru instrumen secara fisik, seperti *multimeter* dan *osiloskop*.^[14] *LabVIEW* terdiri dari 3 komponen utama, yaitu front panel, blok diagram, dan icon and connector panel.

Front panel merupakan interface antara pengguna (*user*) dengan program. Dapat dilihat pada Gambar 2.16 Didalam front panel terdapat kontrol (*input*) dan indikator (*output*). Kontrol pada *front panel* dapat berupa knop, tombol, dial dan lainnya. Sedangkan untuk indikator (*output*) dapat berupa LED,grafik dan tampilan lainnya.



Gambar 2.16 Tampilan *Front Panel LabVIEW*

Setelah membangun front panel, selanjutnya menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan obyek-obyek *front panel*. Blok diagram berisi *source code* program untuk tampilan pada *front panel*. Dapat dilihat pada Gambar 2.17 Blok diagram yang sering digunakan ada pada *tab programming* Terakhir *icon and connector pane*, berfungsi untuk membangun sebuah sub VI, atau program didalam program.



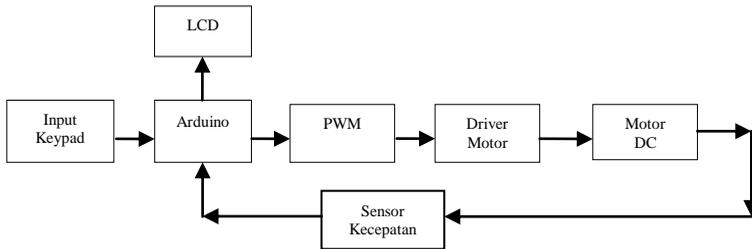
Gambar 2.17 Tampilan *Block Diagram*

(halaman ini sengaja dikosongkan)

BAB 3 PERANCANGAN SISTEM

Pada Bab ini dibahas mengenai perancangan sistem pengaturan kecepatan motor DC *spindle*, yaitu perancangan perangkat keras, perangkat lunak.

3.1 Gambaran Umum Sistem



Gambar 3.1 Diagram Alur Perancangan *System*.

Alur perancangan motor *spindle* pada mesin CNC *Portable* yang saya buat dapat dilihat pada Gambar 3.1. menggunakan *software* arduino dibuat sebuah program untuk mengendalikan kecepatan motor DC *spindle*. Pada bagian awal menginputkan nilai PWM menggunakan *keypad* yang bisa dimonitor melalui LCD, untuk menggerakkan motor *spindle* dari kontroler Arduino Uno. Nilai PWM berfungsi untuk mengatur tegangan yang masuk ke motor DC *spindle* dan tegangan ini mempresentasikan kecepatan motor pada saat itu. Agar dapat menggerakkan motor DC *spindle* perlu alat penggerak motor yang dinamakan *driver* Motor AL75. Untuk melihat kecepatan pada motor DC *spindle* diperlukan sensor *optocoupler* yang nantinya akan mengirim sebuah sinyal pulsa ke controller arduino Uno.

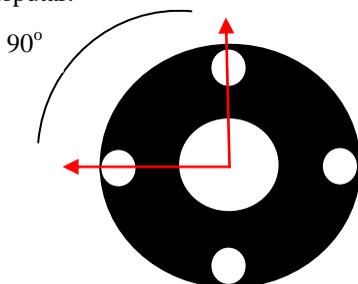
3.2 Perancangan Perangkat Keras

Tahap pertama dalam perancangan sistem adalah perancangan *hardware* yang meliputi perancangan mekanik, perancangan kotak box, perancangan pengkabelan komponen pada *box*. *Hardware* dari sistem terdiri dari beberapa komponen. Komponen - komponen tersebut dirancang dan disesuaikan berdasarkan kebutuhan yang diperlukan. Komponen yang diperlukan diantaranya adalah *power supply* 48V 10A,

modul sensor *Optocoupler*, mikrokontroler arduino, motor DC, keypad, LCD, driver motor DC.

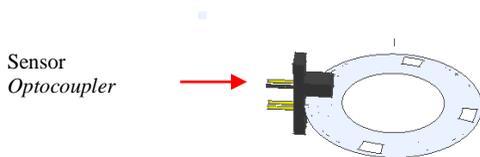
3.2.1 Perancangan Mekanik

Untuk pembacaan sensor *optocoupler*, ujung atas motor DC *spindle* diberi sebuah piringan untuk pembacaan sensor *optocoupler*. Piringan ini terbuat dari bahan akrilik dengan diameter 5 cm. Terdapat 4 lubang. Dapat dilihat pada Gambar 3.2. Penggunaan 4 lubang disini untuk mempermudah pembacaan sensor pada motor DC dengan kecepatan tinggi. Apabila terdapat lebih dari 4 lubang, frekuensi pembacaan pada sensor *optocoupler* tidak bisa stabil, dikarenakan kecepatan putar tidak seimbang dengan pembacaan lubang saat motor DC *spindle* berputar.



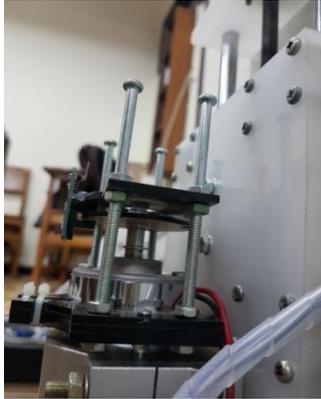
Gambar 3.2 Piringan 4 Lubang

Perbedaan jarak disetiap lubang kecil pada piringan adalah 90°.



Gambar 3.3 Pemasangan Piringan Pada Sensor *Optocoupler*

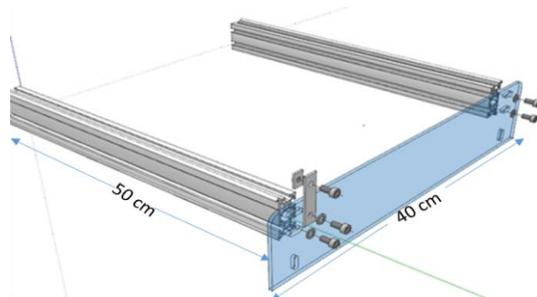
Setelah pemasangan piringan pada sensor *optocoupler* dipasang, dapat dilihat pada Gambar 3.3. kemudian menggabungkan motor DC *spindle* dengan piringan yang sudah terpasang sensor. Posisi penempatan sensor harus berada di tengah karena menyesuaikan posisi piringan. Dapat dilihat pada Gambar 3.4.



Gambar 3.4 Penggabungan Sensor *Optocoupler* dengan Motor DC

3.3.2 Kerangka Mesin CNC *Portable*

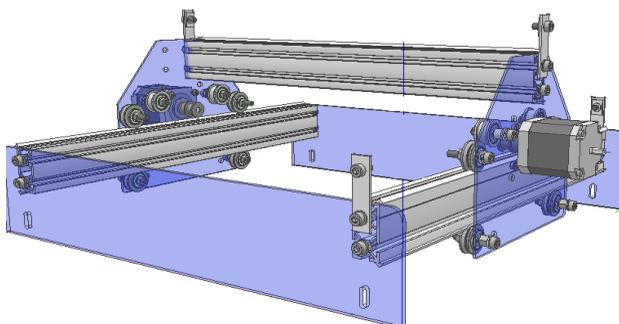
Dalam perakitan mesin CNC *portable* terdapat berbagai komponen penggerak mesin, seperti motor *stepper*, *ballscrew*, *belt*, dan motor DC *spindle*. Peletakan komponen tersebut didalam kerangka mesin CNC perlu sesuai ukuran dan tempat yang ditentukan. Gambar 3.5



Gambar 3.5 Perancangan Kerangka Meja CNC *Portable*

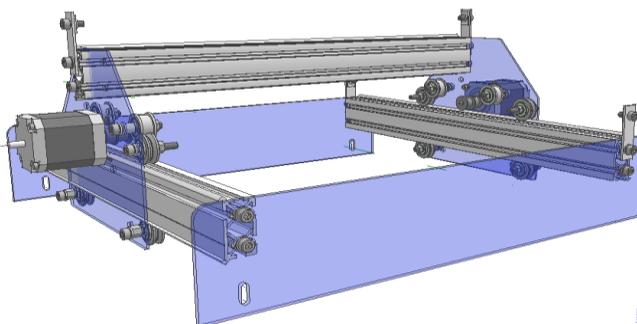
Selain peletakan komponen pada kerangka mesin CNC *portable*, perlu juga memperhatikan bahan pembuat mesin. Bahan yang digunakan harus kuat dan kokoh agar saat mesin bekerja tidak terdapat getaran berlebihan yang. Hal tersebut agar hasil kerja mesin CNC *portable* bisa presisi dan rapi sehingga sesuai dengan gambaran yang kita inginkan.

Pergerakan pada setiap sumbu X,Y, dan Z ini dilakukan oleh motor *stepper*. Motor *Stepper* tersebut dipasang *gear* dan terdapat 2 *gear* tambahan untuk dipasang *belt* agar tiap sumbu dapat bergerak dengan baik.



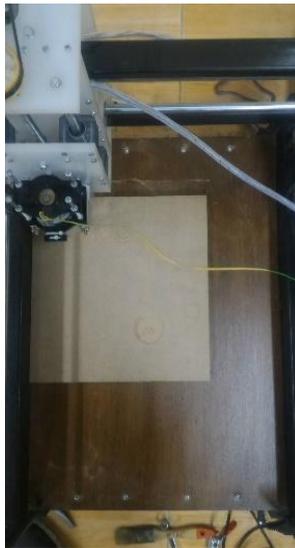
Gambar 3.6 Perancangan Motor *Stepper* pada Sumbu Y Kanan

Pada sumbu Y terdapat 2 motor *stepper* kanan dan kiri Gambar 3.6 dan Gambar 3.7 untuk memperkuat dan menyeimbangkan pergerakan pada sumbu ini, karena ukuran meja kerja mesin CNC *portable* ini cukup besar. Sehingga membutuhkan dorongan gerak yang kuat.



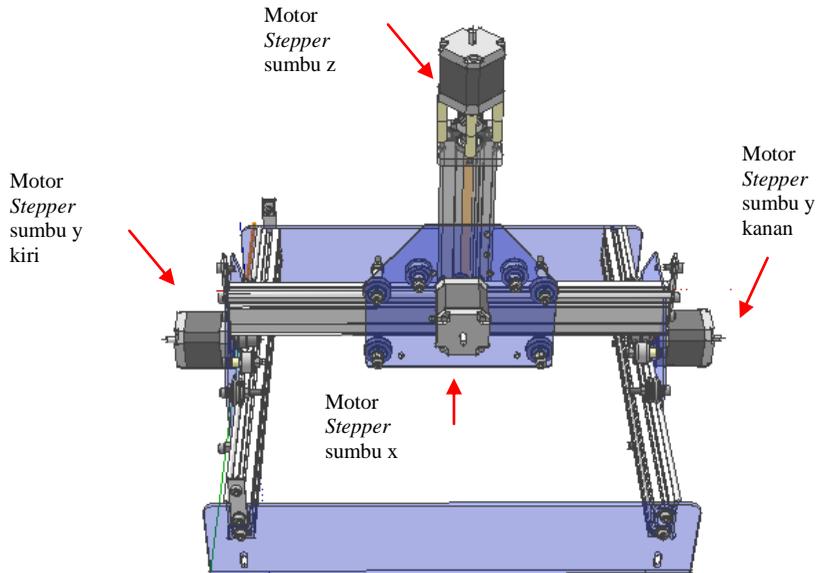
Gambar 3.7 Perancangan Motor *Stepper* pada Sumbu Y Kiri

Pada bagian samping kanan dan kiri terpasang motor *stepper* dengan diberikan *gear* dan *fanbelt* untuk menggerakkan mesin CNC *portable*. Tidak hanya bagian samping kanan dan kiri, tapi juga bagian belakang dan atas yang juga terpasang motor *stepper* dengan *gear* dan *fanbelt*. Pada bagian meja kerja mesin CNC *portable* menggunakan bahan kayu dan di sisi kanan kiri dipasang bahan metal untuk menyangga mesin CNC *portable* sehingga tidak menimbulkan getaran berlebih yang bisa mengganggu kinerja mesin CNC *portable*. Gambar 3.8 Di bagian meja kerja tersebut merupakan tempat untuk benda kerja yang akan dikerjakan oleh mesin CNC *portable*.

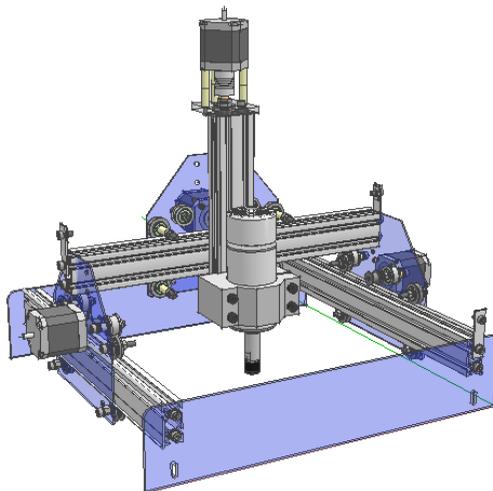


Gambar 3.8 Meja Kerja pada Mesin CNC *Portable* Menggunakan Alas Kayu.

Pada bagian sumbu Z terdapat *holder* untuk menopang motor DC spindle yang menggunakan *ballscrew* untuk. Dapat dilihat pada Gambar 3.9. Penggunaan *ballscrew* ini agar sumbu Z dapat bergerak naik turun dengan baik. Untuk menjaga pergerakannya agar tetap baik, maka diperlukan pelumasan pada *ballscrew*. Karena *ballscrew* sendiri terbuat dari poros berulir yang rentan berkarat.



Gambar 3.9 Perancangan Letak Motor *Stepper* pada Sumbu X

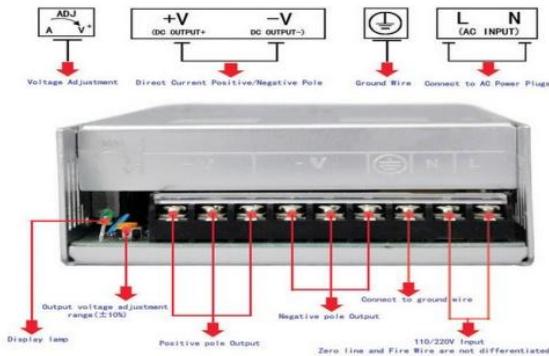


Gambar 3.10 Perancangan Mesin CNC *Portable*

Holder pada sumbu Z sebagai penopang motor *stepper* dan motor *spindle* dibuat dengan menggunakan bahan akrilik. Akrilik tersebut memiliki ketebalan 5mm agar akrilik tersebut dapat menopang dengan kuat komponen yang terpasang pada *holder* sumbu Z seperti motor *stepper*, motor DC *spindle*, dan *ballscrew*. Hasil pembuatan alat dapat dilihat pada lembar lampiran A.

3.3.3 Power Supply Switching

Sumber tegangan yang di pakai dalam perancangan ini menggunakan *power supply switching* dengan model ZL-400W-48. *Output* dari *power supply* ini 47-49 VDC. Untuk mengubah tegangan menjadi 48VDC hanya memutar *adjustable* tegangan.



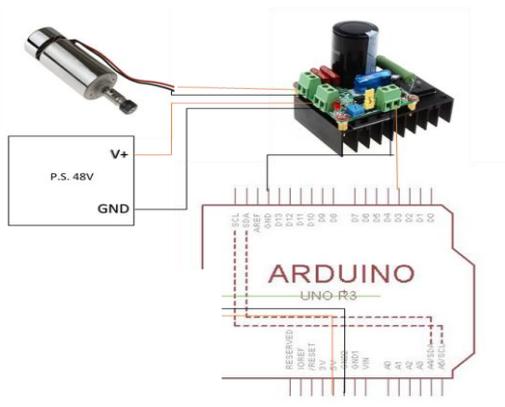
Gambar 3.11 *Power Supply Switching*

Port L dan *N* pada *power supply switching* digunakan untuk kabel *power* (kabel AC) 110 – 220 VAC. Dapat dilihat pada Gambar 3.11. Terdapat sebuah led untuk indikator *power supply* mendapatkan sumber dari tegangan AC.

3.3.4 Wiring Driver MACH3 AL75 dengan Arduino

Agar dapat menggerakkan motor DC *spindle* perlu alat penggerak motor DC *spindle* yang dinamakan *Driver* Motor. Dapat dilihat pada Gambar 3.12. Indikator led menandakan driver tersebut sudah mendapatkan sumber, kaki PWM+ masuk ke pin arduino D3 dan

kaki PWM- masuk ke GND arduino. *Output* dari *driver* ini dihubungkan ke motor DC *spindle*.



Gambar 3.12 wiring *Driver* Motor MACH3 AL75

3.3.5 *Box* Kontrol

Pada pembuatan *box controller* dibuat dengan menggunakan material kayu lapis sebagai bahan utama. Pemilihan bahan kayu lapis dikarenakan sifatnya yang mudah dibentuk serta lebih murah jika dibandingkan dengan bahan akrilik. *Box Controller* ini memiliki fungsi sebagai wadah bagi komponen-komponen lain yang berperan dalam menjalankan mesin CNC *portable* ini. Komponen-komponen dalam *box controller* tersebut diantaranya: rangkaian *driver* motor, arduino, *cooling* LCD.

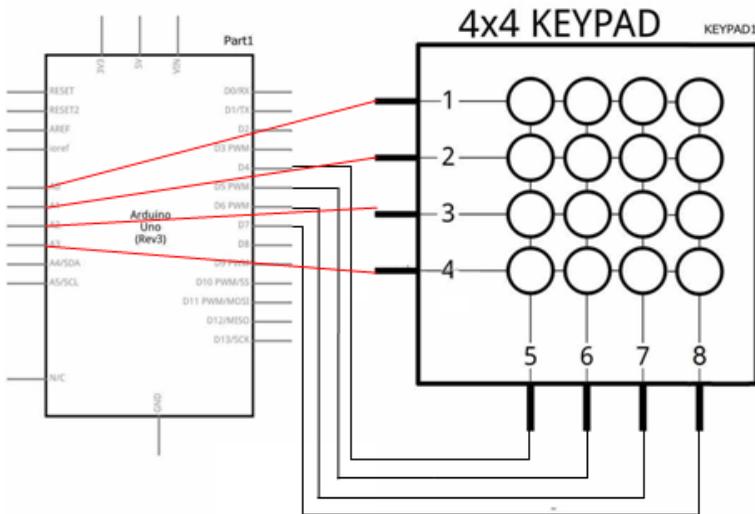


Gambar 3.13 Perancangan *Box* Kontrol

Pembuatan *box* controller sendiri hanya membutuhkan gergaji kayu untuk membentuk setiap potongan kayu menjadi *box* Gambar 3.13 yang ringan dan mudah untuk di bawa.

3.3.6 Wiring Keypad 4x4 dengan Arduino

Pada tugas akhir ini penginputan nilai RPM menggunakan *keypad* 4x4 yang tersambung ke Arduino. Pin yang tersedia pada *keypad* adalah pin 1-8. Dimana pin 1-4 untuk baris dan pin 5-8 untuk kolom. Kaki *pinout* 4 baris *keypad* tersambung dengan pin *digital* A0, A1, A2 dan A3 sedangkan kaki *pinout* 4 kolom *keypad* tersambung dengan pin *digital* D7, D6, D5 dan D4. Untuk mengetahui kaki *pinout* keypad 4x4. Dapat dilihat pada Gambar 3.14

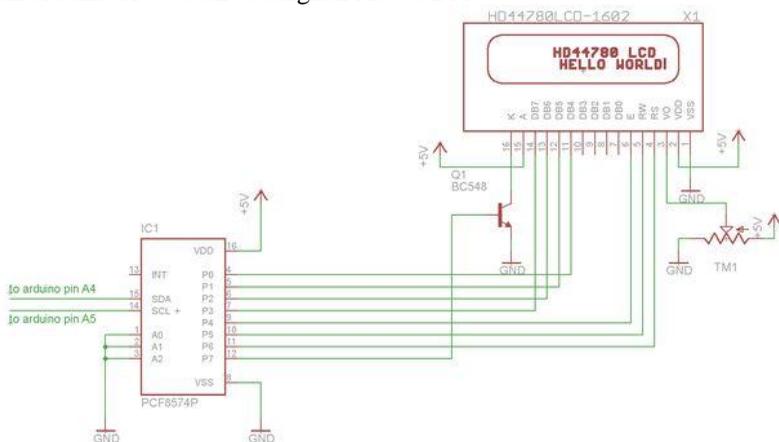


Gambar 3.14 Wiring Keypad dengan Arduino

3.3.7 Wiring LCD 16x2 dengan Arduino

Untuk mengetahui suatu nilai dari input PWM (*Pulse Width Modulation*) dan kecepatan motor terdapat sebuah interface LCD 16x2. Komponen ini berfungsi untuk sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. LCD mempunyai 16 kaki pin yang mempunyai fungsi masing-masing. Dapat dilihat pada Gambar 3.15.

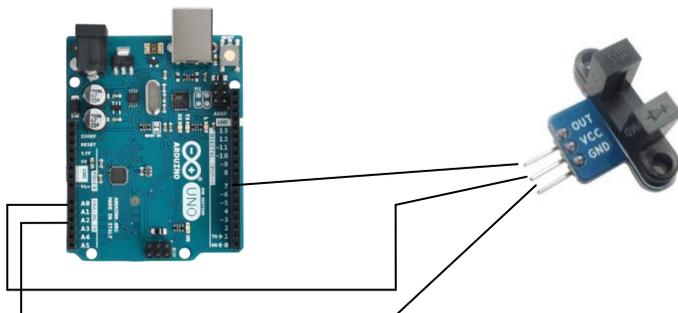
Untuk menghemat pin pada arduino, membutuhkan komponen I2C. 16 kaki pin pada LCD di hubungkan ke I2C kemudian output dari I2C adalah dataSLC dan SDA. Pin *output* tersebut dihubungkan pada pin arduino A4 = SCL sedangkan A5 = SDA.



Gambar 3.15 Wiring LCD 16x2 dengan I2C

3.3.8 Wiring Sensor Optocoupler

Sensor untuk mengukur kecepatan dari motor DC *spindle* yang digunakan dalam perancangan dan pembuatan Tugas Akhir ini adalah sensor *optocoupler*. Mempunyai 3 pin diantaranya VCC terhubung ke 5V dan GND terhubung pada GND arduino. Sedangkan pin data dihubungkan ke pin arduino D2. Dapat dilihat pada Gambar 3.16.



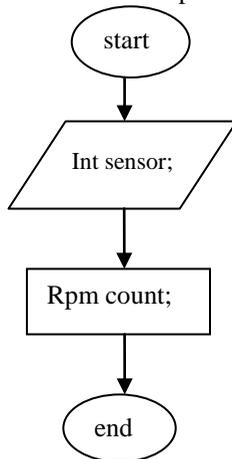
Gambar 3.16 Wiring Optocoupler dengan Arduino

3.3 Perancangan Perangkat Lunak

Pada perancangan tugas akhir ini perangkat lunak yang di gunakan adalah Arduino *Software* (IDE) dan LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*).

3.3.1 Program Sensor Kecepatan

Software ini menggunakan *software* Arduino. Yang nantinya sensor tersebut terbaca pada serial monitor arduino IDE.



Gambar 3.17 Flowchart Pembacaan Sensor

Fungsi *falling* pada program ini berfungsi untuk pembacaan pulsa setiap ada *interrupt* dari *low* ke *high*. Lebih lengkap nya program dapat dilihat pada lembar lampiran A.

```
File Edit Sketch Tools Help
sensor_rpm_1
// read RPM
int half_revolutions = 0;
int rpm = 0;
unsigned long lastmillis = 0;
void setup(){
  Serial.begin(9600);
  attachInterrupt(0, rpm_fan, FALLING);
}
void loop(){
  if (millis() - lastmillis == 1000){ //Uptade every one second, th
    detachInterrupt(0); //Disable interrupt when calculating
```

```

rpm = half_revolutions * 15; // Convert frequency to RPM, note: t
Serial.print("RPM =\t"); //print the word "RPM" and tab.
Serial.println(rpm); // print the rpm value.
Serial.print("\t Hz=\t"); //print the word "Hz".
Serial.println(half_revolutions); //print revolutions per second
half_revolutions = 0; // Restart the RPM counter
lastmillis = millis(); // Uptade lastmillis
attachInterrupt(0, rpm_fan, FALLING); //enable interrupt

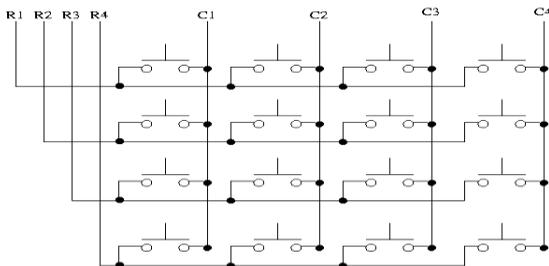
```

Gambar 3.18 Program Tes Sensor *Optocoupler*.

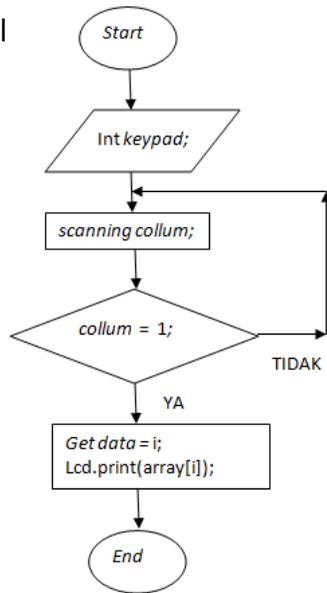
Pada Gambar 3.17 Arduino menerima sebuah sinyal pulsa yang didapat dari pembacaan lubang pada piringan, kemudian di counter dengan dengan fungsi “rpm count”. Kecepatan dapat dibaca pada jumlah pulsa counter yang terbaca pada arduino dikali 600 untuk dijadikan per menit karena satuan dari RPM adalah *Revolution Per Minute* setelah itu dibagi 4 karena pada piringan terdapat 4 lubang. Hasil perhitungan tadi ditampilkan pada LCD 16x2. Terdapat sebuah fungsi *lastmillis* kegunaan nya sebagai sistem pewaktuan *update counter* pada sensor kecepatan. Lebih lengkap nya program Gambar 3.18 dapat dilihat pada lembar lampiran A.

3.3.2 Program Keypad

Program ini menggunakan *software* arduino. Hasil dari program ini adalah memunculkan angka pada LCD 16x2. Metode yang digunakan merupakan metode *scanning keypad*.



Gambar 3.19 Rangkaian *Scanning Keypad*



Gambar 3.20 Flowchart Scanning Keypad

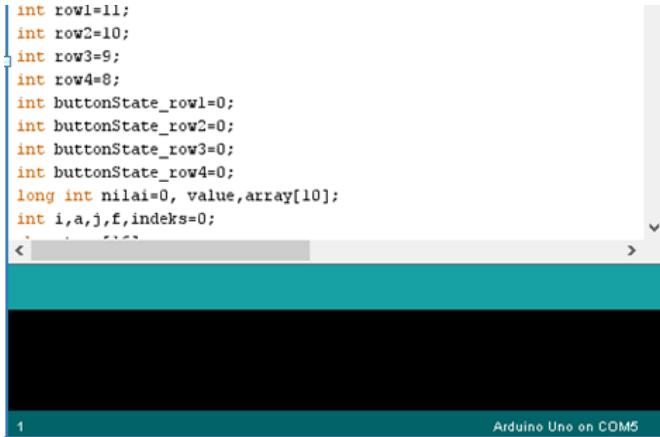
Metode *scanning* ini memberi *logic low* dan *high* antara kolom dan baris pada *keypad*. Dapat dilihat pada Gambar 3.19 dan Gambar 3.20. Dalam pembuatan program ini diberi logika *high* untuk baris dan *low* untuk kolom. Ketika salah satu baris mendapat sinyal *high* maka data tersebut disimpan pada variable *i*. Apabila belum mendapatkan sinyal *high* maka kembali pada proses *scanning*. Data yang tersimpan pada variable *i* akan ditampilkan pada LCD 16x2.

```

File Edit Sketch Tools Help
serialkeypad
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 ,2,1,0,4,5,6,7,3, POSITIVE);

int pwm=3;
int col1=7;
int col2=6;
int col3=5;
int col4=4;
  
```

```
int row1=11;
int row2=10;
int row3=9;
int row4=8;
int buttonState_row1=0;
int buttonState_row2=0;
int buttonState_row3=0;
int buttonState_row4=0;
long int nilai=0, value,array[10];
int i,a,j,f,indeks=0;
```



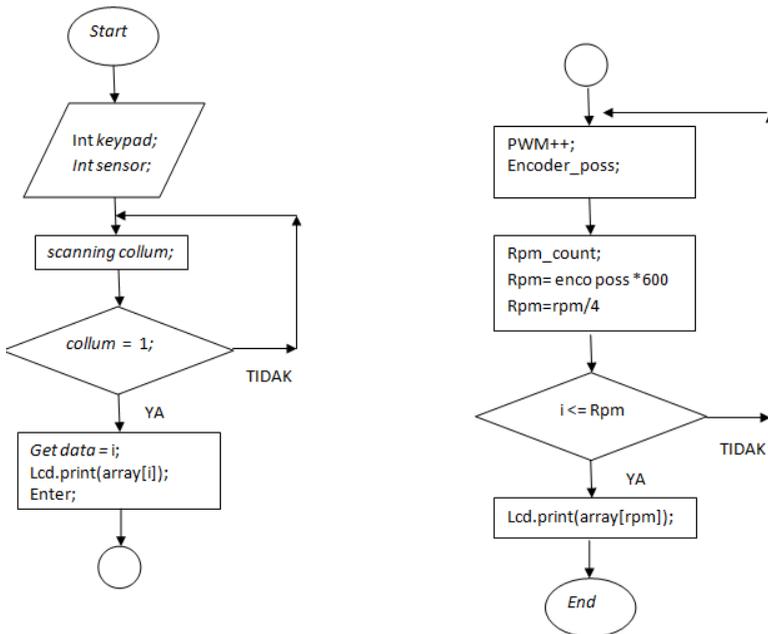
Gambar 3.21 Program *Scanning Keypad*.

Lebih lengkap nya program Gambar 3.21 dapat dilihat pada lembar lampiran C.

3.4.3 Program Mengatur PWM pada Arduino

Program ini menggunakan software arduino. Hasil dari pembuatan program ini adalah menggerakkan motor DC *spindle* sesuai dengan *setpoint*. Dapat dilihat pada Gambar 3.22, terdapat inisialisasi pin yang digunakan oleh *keypad* dan sensor *optocoupler* ke arduino. Sama seperti program sebelumnya, metode memasukkan nilai *setpoint* melalui *keypad* yaitu dengan metode *scanning*. Apabila sudah mendapatkan suatu nilai, maka nilai tersebut disimpan pada *variable* “i”. selanjutnya adalah tekan enter pada *keypad*. Maka secara otomatis fungsi “*pwmwrite*” dieksekusi. Fungsi inilah yang mengakibatkan motor DC *spindle* bergerak melalui nilai *digital* PWM pada arduino. Rentang nilai PWM pada *mikrokontroller* ini 0-255. Pada program ini dibuat nilai *digital* PWM mengkonter +1. Ketika motor DC *spindle* mulai berputar sensor *optocoupler* akan menerima sebuah sinyal yang nantinya akan di konversi ke RPM. Gambar 3.23 Sinyal tersebut berasal dari piringan 4 lubang. Jika nilai “i” (*setpoint*) sama dengan RPM, maka nilai *digital* PWM akan berhenti mengkonter. Namun jika tidak nilai *digital* PWM akan terus mengkonter sampai sama dengan nilai “i”. Pembacaan sensor yang sudah mencapai nilai “i” akan ditampilkan pada LCD 16x2.

Lebih lengkap nya program dapat dilihat pada lembar lampiran D.



Gambar 3.22 Flowchart Program Mengatur PWM pada Arduino

```

File Edit Sketch Tools Help
[Icons]
luluslabview
}

void loop() {
  lcd.setCursor(0,0);
  lcd.print("Target:");
  lcd.setCursor(0,1);
  lcd.print("Status:");
  keypad();
  value=nilai;
  while (indeks==1) {
    pwm_out++;
    delay(100);
  }
}
  
```

```

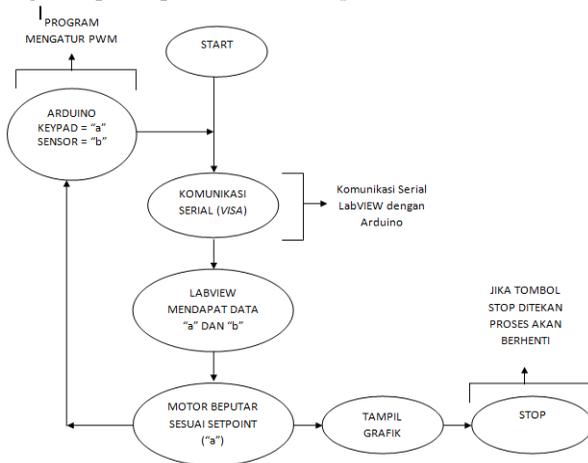
//lcd.setCursor(13,1);
//lcd.print(pwm_out);
analogWrite(pwm,pwm_out);
for(i=0;i<100;i++){
    rpm = encoder_pos * 60;
    rpm=rpm/8;
    delay(10);
}

```

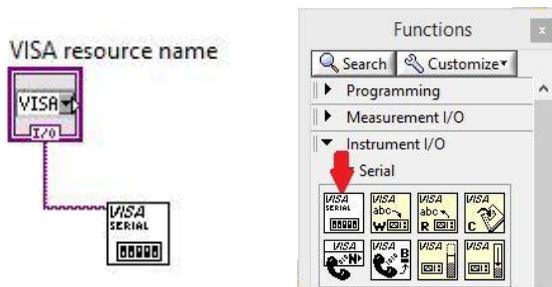
Gambar 3.23 Program Mengatur PWM pada Arduino.

3.4.4 Diagram Blok LabVIEW

Software ini menggunakan Labview. Hasil dari pembuatan program ini adalah menampilkan gelombang *setpoint* dan menampilkan gelombang kecepatan pada motor DC *spindle*. Gambar 3.24

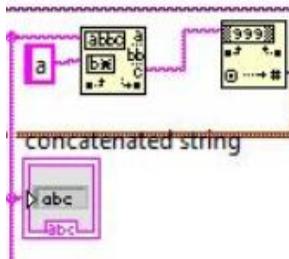


Gambar 3.24 Diagram Blok Sistem pada LabVIEW



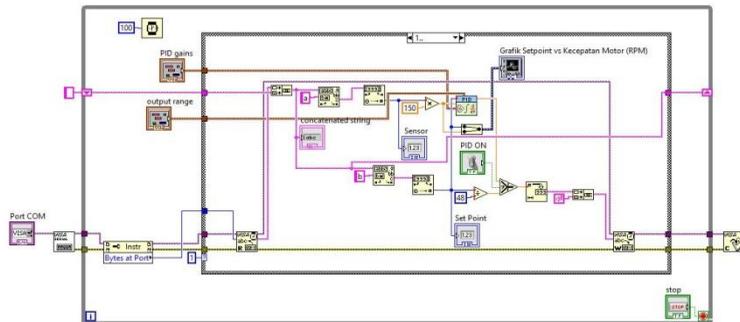
Gambar 3.25 VISA Resources Names

Pembuatan program untuk menjalankan motor DC *Spindle* pada *LabVIEW* dimulai dari memasukkan komponen komunikasi serial pada *Diagram Block LabVIEW* yang nantinya tersambung dengan program *Arduino*. Komunikasi serial yang terdapat pada *LabVIEW* dinamakan VISA (*Virtual Instrument Software Architecture*) Gambar 3.25 yaitu standard untuk memprogram, mengkonfigurasi, dan memecahkan masalah sistem instrumentasi yang terdiri dari antarmuka, GPIB, VXI, PXI, Serial, Ethernet, dan /atau USB. VISA menyediakan fitur antarmuka antara *hardware* dengan aplikasi *LabVIEW*. Pada *Diagram Block* yang saya buat, diawal program terdapat *VISA resources name* untuk inialisasi port COM yang tersambung di komputer kita. *VISA resources name* ini dapat dimasukkan pada program *LabVIEW* melalui palette control pada bagian menu Silver I/O. Kemudian *VISA resources name* dihubungkan dengan *VISA Configure Serial Port* dimana fungsi komponen ini untuk mengatur komunikasi serial antara *hardware* dengan *LabVIEW*. Pada *VISA Configure Serial Port* terdapat beberapa pin pengaturan komunikasi serial diantaranya pengaturan *baud rate*, data bits, *parity,error*, *VISA resources name out*, *error out*, dan yang lainnya. *VISA Configure Serial Port* terdapat dalam *Function Palette* pada sub menu Serial. Komunikasi serial antara *hardware* dengan *LabVIEW* harus sama agar dapat menjalankan *hardware*nya dengan program *LabVIEW*. Dalam pemrograman di *Arduino*, untuk dapat menjalankan motor DC *Spindle* menggunakan komunikasi serial dengan *baud rate* bernilai 9600 maka dari itu dalam program *LabVIEW* diatur nilai *baud ratenya* bernilai 9600. Sedangkan data bits yang digunakan merupakan data bits *default* yaitu bernilai 8.



Gambar 3.28 Inisialisasi Data Keypad pada LabVIEW

Data dikirim melalui arduino dengan fungsi “*serial.println*”. berikut blok diagram yang telah dibuat. Gambar 3.29. *Structure Case* ini dapat diambil pada bagian *Programming* kemudian *Structures* di *Function Palette*. Pada *Structures Case* ini terdapat berbagai komponen dan fungsi untuk mengolah masukan yang diberikan agar hasil keluarannya sesuai dengan program yang diberikan Arduino. Lalu komunikasi serial yang telah diatur dihubungkan dengan *VISA Read* agar programnya dapat terbaca pada LabVIEW.



Gambar 3.29 *Structures Case* Program LabVIEW

(halaman ini sengaja dikosongkan)

BAB 4

PENGUJIAN DAN ANALISA

Pada Bab ini dibahas tentang pengujian dan analisa sistem mengatur kecepatan motor DC dengan PWM yang telah dibuat.

4.1 Lingkungan Uji Coba

- a. Perangkat Keras (*Hardware*) :
 1. Arduino Uno
 2. LCD 16x2
 3. Sensor *Optocoupler*
 4. Keypad 4x4
 5. PowerSupply 48V 10A
 6. *Driver MACH AL75*
 7. Motor DC
 8. *Tachometer*
- b. Perangkat Lunak (*Software*) :
 1. *Software* Arduino IDE
 2. *LabVIEW*

Hasil proses yang telah dilakukan dan dihasilkan dalam tugas akhir ini meliputi :

4.2 Pengujian Perangkat Keras

Pengujian perangkat keras bertujuan untuk mengetahui perangkat keras yang dirancang telah berfungsi baik, dan mengetahui performa dari perangkat tersebut. Pengujian tersebut meliputi, pengujian sensor *optocoupler*, kecepatan motor dengan *tachometer*, dan pengujian output PWM.

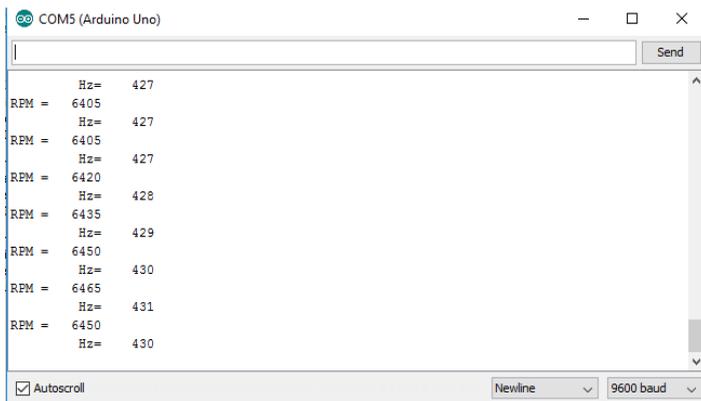
4.2.1 Mengukur Kecepatan Motor DC *Spindle* dengan *Optocoupler*

Pengujian ini dilakukan untuk mengetahui seberapa sensitif sensor membaca piringan dengan metode lubang. Logika sensor ini adalah :

- 0 = Saat celah sensor terhalang
- 1 = Saat celah sensor tanpa-halangan

Perhatikan dulu sistem sensor kecepatannya ada piringan dengan 4 lubang. Artinya jika motor berputar 1 putaran penuh (1 rotasi), lubang terbaca oleh sensor dan menimbulkan sinyal 4 pulsa. Pulsa tersebut dibaca oleh *mikrokontroler* menggunakan *counter*. Kecepatan putar

sering disebut dengan rpm (*rotation per minute*). Jadi pembacaan jumlah putaran dibua dalam 1 menit. Jumlah *counter* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Pengujian Sensor Optocoupler

Hasil pengujian sensor *optocoupler* tersebut disajikan dalam tabel hasil pengujian pada Tabel 4.1.

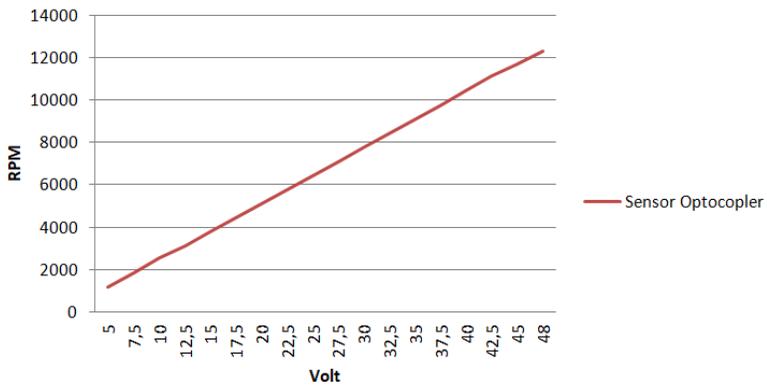
Tabel 4.1 Hasil Pengujian Sensor *Optocoupler*

Tegangan input (Volt)	Kecepatan (rpm)
5 V	1194 rpm
7.5 V	1838 rpm
10 V	2576 rpm
12.5 V	3120 rpm
15V	3760 rpm
17.5 V	4440 rpm
20 V	5100 rpm
22.5 V	5760 rpm
25 V	6420 rpm
27.5 V	7080 rpm
30 V	7740 rpm
32.5 V	8395 rpm
35 V	9050 rpm
37.5 V	9701 rpm

40 V	10390 rpm
42.5 V	11197 rpm
45 V	11700 rpm
48 V	12170 rpm

Berdasarkan Tabel 4.1 dapat dilihat bahwa semakin besar tegangan yang diberikan maka kecepatan motor juga bertambah. Gambar 4.2. Spesifikasi motor *spindle* adalah 48V = 12.000 RPM. Pada data tidak sesuai dengan spesifikasi motor DC spindle, dimana tegangan pada 48V sensor membaca 12.170 RPM. Selisih 170 dari spesifikasi nya.

Pembacaan Sensor *Optocoupler*



Gambar 4.2 Grafik Pembacaan Sensor *Optocoupler*

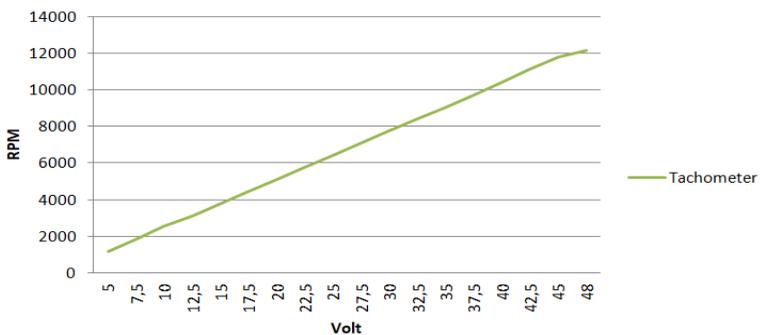
4.2.2 Mengukur Kecepatan Motor DC *Spindle* dengan *Tachometer*

Pengujian kecepatan motor ini adalah untuk mengetahui pembacaan nilai kecepatan yang diukur oleh sensor optocoupler dengan hasil pengukuran kecepatan motor menggunakan alat ukur kecepatan, sehingga dapat dilakukan kalibrasi yang menghasilkan persamaan konversi. Alat ukur kecepatan yang digunakan adalah *Laser Tachometer DT-2234C⁺* Berikut adalah hasil pengujian kecepatan motor pada data hasil pengujiannya ditampilkan pada Tabel 4.2. Dapat dilihat bahwa semakin besar tegangan yang diberikan maka kecepatan motor juga bertambah. Gambar 4.3. Diambil salah satu sampel dari pengukuran ini yaitu pada tegangan 25 V. Gambar 4.4

Tabel 4.2 Hasil Pengujian Alat Ukur Kecepatan *Tachometer*

Tegangan input (Volt)	Kecepatan (rpm)
5 V	1200 rpm
7.5 V	1860 rpm
10 V	2580 rpm
12.5 V	3125 rpm
15V	3780 rpm
17.5 V	4437 rpm
20 V	5106 rpm
22.5 V	5761 rpm
25 V	6431 rpm
27.5 V	7083 rpm
30 V	7742 rpm
32.5 V	8400 rpm
35 V	9061 rpm
37.5 V	9720 rpm
40 V	10440 rpm
42.5 V	11100 rpm
45 V	11781 rpm
48 V	12155 rpm

Alat Ukur Tachometer



Gambar 4.3 Pengujian *Optocoupler* dengan *Tachometer*



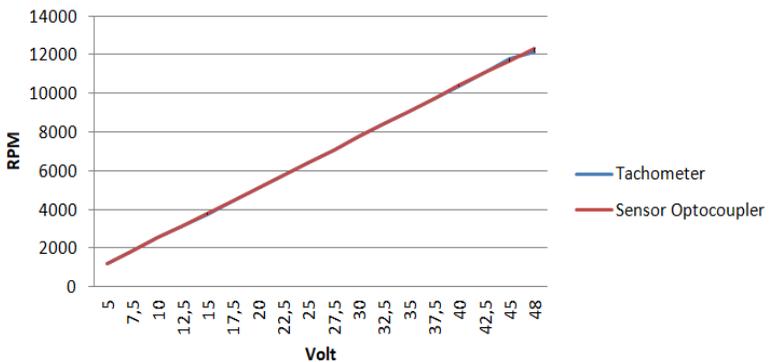
Gambar 4.4 Pengujian Kecepatan Motor dengan *Tachometer*

Spesifikasi dari motor DC *spindle* ini adalah 48V = 12.000 rpm. Sama seperti pembacaan sensor *optocoupler* pada saat diberikan tegangan 48V kecepatan yang terukur adalah 12.155 rpm melebihi dari spesifikasi motor. Ini dikarenakan produk *powersupply* yang digunakan memiliki perbedaan frekuensi.

4.2.3 Perbandingan Kecepatan Sensor *Optocoupler* dengan *Tachometer*

Telah didapat data kecepatan antara sensor *optocoupler* dengan alat ukur *tachometer*. Pengujian kedua nya bertujuan untuk mencari nilai ketelitian pada sensor *optocoupler*.

Perbandingan *Optocoupler* dengan Alat Ukur *Tachometer*



Gambar 4.5 Perbandingan Pengukuran Kecepatan *Optocoupler* dengan *Tachometer*

Tabel 4.3 Perbandingan Selisih Kecepatan pada *Tachometer* dan *Optocoupler*

Tegangan (Volt) <i>input</i>	Kecepatan (rpm) <i>Tachometer</i>	Kecepatan (rpm) <i>Sensor Optocoupler</i>	Selisih
5 V	1200 rpm	1194 rpm	6 rpm
7.5 V	1860 rpm	1838 rpm	22 rpm
10 V	2580 rpm	2576 rpm	4 rpm
12.5 V	3125 rpm	3120 rpm	5 rpm
15V	3780 rpm	3760 rpm	20 rpm
17.5 V	4437 rpm	4440 rpm	3 rpm
20 V	5106 rpm	5100 rpm	6 rpm
22.5 V	5761 rpm	5760 rpm	1 rpm
25 V	6431 rpm	6420 rpm	11 rpm
27.5 V	7083 rpm	7080 rpm	3 rpm
30 V	7742 rpm	7740 rpm	2 rpm
32.5 V	8400 rpm	8395 rpm	5 rpm
35 V	9061 rpm	9050 rpm	11 rpm
37.5 V	9720 rpm	9711 rpm	9 rpm
40 V	10440 rpm	10390 rpm	10 rpm
42.5 V	11100 rpm	11197 rpm	3 rpm
45 V	11781 rpm	11700 rpm	81 rpm
48 V	12170 rpm	12155 rpm	15 rpm
Rata-rata			12,05 rpm

Dari perbandingan data Gambar 4.5 dan Tabel 4.3 dapat dilihat bahwa ketelitian dari sensor *optocoupler* hampir mendekati alat ukur *tachometer* hanya selisih rata-rata 12,05 rpm. Hal ini terjadi karena ketelitian sistem pembacaan pada sensor *optocoupler* dibuat fungsi “lastmilis” yaitu fungsi yang mengupdate data pulsa permilisekon pada arduino. Sama halnya dengan ketelitian pembacaan alat ukur *tachometer*. Menggunakan penurunan waktu yang diambil untuk setiap pilihan celah yang dilewati cahaya laser.

4.2.4 Pengujian *Output* PWM Mikrokontroler Arduino pada *Driver* Motor



Gambar 4.6 Pengujian *Output* PWM Mikrokontroler pada *Driver* Motor

Pengujian ini dilakukan untuk mengetahui *duty cycle* yang keluar pada PWM mikrokontroler arduino. Nilai PWM di inputkan melalui *keypad* 4x4. Kemudian tegangan di ukur pada kaki pin *output driver* motor DC positif dan negatif menggunakan alat ukur *AVO meter*.

Tabel 4.4 Hasil Pengujian Nilai *Digital* PWM

Nilai <i>Digital</i> PWM	Tegangan (V_{out})	<i>Duty Cycle</i>
35	4,95 V	10%
63	9,41 V	20%
78	14,47 V	30%
110	20,60 V	40%
147	24,56 V	50%
165	28,81 V	60%
187	33,73 V	70%
205	38,45 V	80%
225	43,26 V	90%
255	47,54 V	100%

Berikut adalah hasil dari pengukuran tegangan nilai *digital* PWM pada arduino. *Dutycycle* 40% dapat dilihat pada *osiloskop* dengan tegangan output 20,60 volt pada alat ukur tegangan *AVO* meter. Gambar 4.6. Dari tabel 4.4 diambil data *digital* PWM dari *mikrokontroler* arduino dengan nilai maksimal 255. Dengan *input* tegangan 48V. Pada data nilai *digital* PWM 255 berarti tegangan yang dikeluarkan maksimum, tetapi pada Tabel 4.2 tegangan yg dikeluarkan hanya 47,54

selisih 0,46 V. Ini dikarenakan frekuensi PWM pada arduino terlalu besar dengan frekuensi driver motor AL75. *Duty cycle* dapat dilihat sinyal yang dihasilkan oleh *oscilloscope* atau dapat dihitung dengan persamaan $V_{out} = Duty\ Cycle \times V_{in}$

4.2.4 Pengujian Menampilkan Karakter pada LCD 16x2

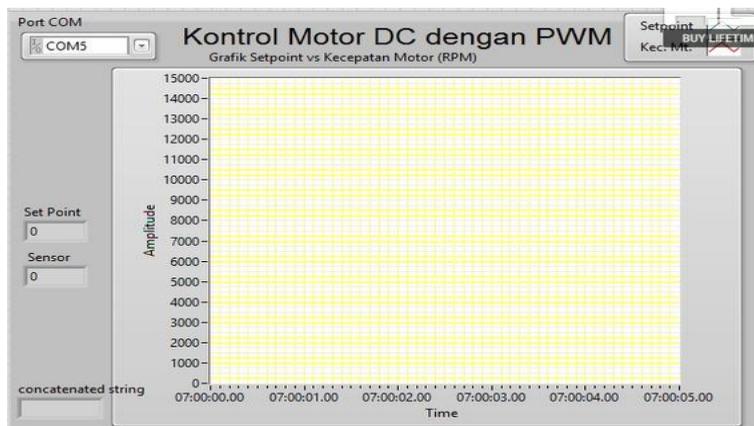
Pengujian ini dilakukan untuk mengetahui LCD 16x2 bisa berfungsi dengan baik atau tidak. Gambar 4.7. Angka dimasukkan melalui *keypad*. Metode yang dilakukan pada pengujian ini dilakukan dengan metode scanning pada *keypad*. Metode ini mengeksekusi pada baris dan kolom. *Keypad* ditekan kemudian dikirim ke arduino untuk ditampilkan pada LCD 16x2.



Gambar 4.7 Menampilkan Karakter Angka pada LCD 16x2

4.3 Pengujian Perangkat Lunak

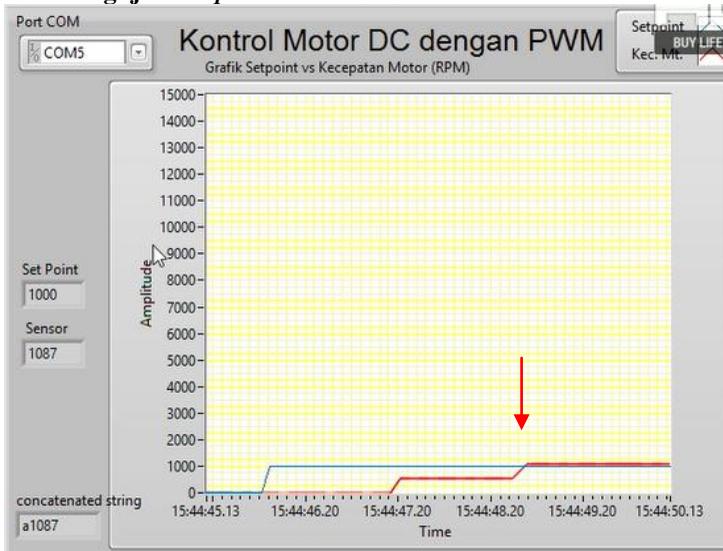
Pengujian ini dilakukan pada software Labview. Pada tugas akhir ini Labview digunakan untuk mengetahui *rise time* sebuah gelombang respon kecepatan motor DC spindle untuk mencapai *setpoint*.



Gambar 4.8 Tampilan *Front Panel* LabVIEW Kontrol Motor DC

Terdapat 2 gelombang pada Gambar 4.8 gelombang berwarna biru untuk setpoint, sedangkan gelombang berwarna merah untuk sensor. Pada sumbu X dibuat parameter waktu (*time*) dan sumbu Y dibuat parameter RPM (*speed*). *PortCOM* digunakan untuk komunikasi serial antara Arduino dan LabVIEW.

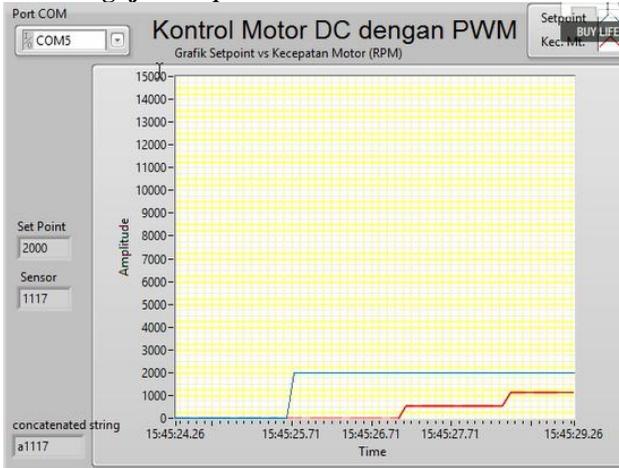
4.3.1 Pengujian *Setpoint* 1000 RPM



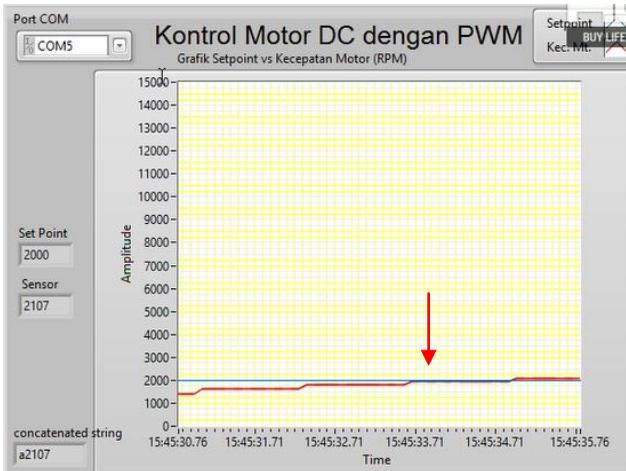
Gambar 4.9 Hasil Pengujian *Rise Time* 1000 RPM

Respon kecepatan motor DC bergerak pada detik ke-47. Garis berwarna biru merupakan data dari *keypad* yang disimpan pada *variable* “a”. Garis berwarna merah merupakan pembacaan sensor yang disimpan pada *variabel* b. Penyimpanan ini digunakan untuk pengiriman data pada *LabVIEW*. Sehingga muncul grafik. Untuk mencapai *setpoint* 1000 rpm, waktu yang dibutuhkan 1 detik. Terlihat pada Gambar 4.9 respon kecepatan tepat berada pada *setpoint* pada detik ke-48.

4.3.2 Pengujian *Setpoint* 2000 RPM



Gambar 4.10 Hasil pengujian *Rise Time* 2000 RPM

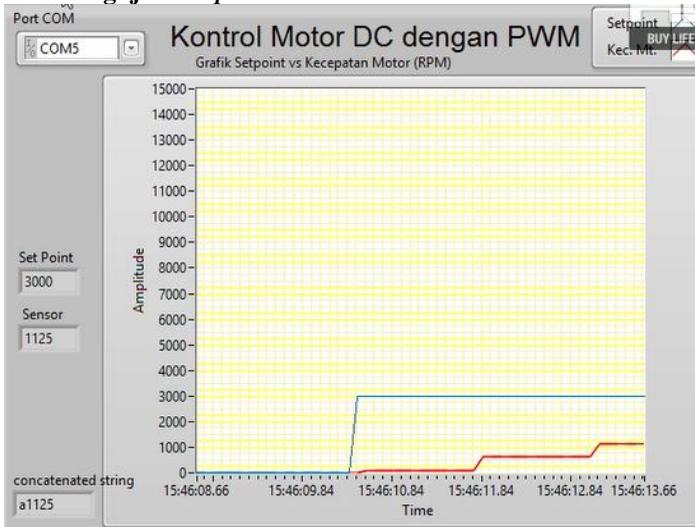


Gambar 4.11 Pengujian *Rise Time* Tepat pada *Setpoint* 2000 RPM

Respon kecepatan motor DC pada Gambar 4.10 bergerak pada detik ke 26. Garis berwarna biru merupakan data dari *keypad* yang disimpan pada *variable* "a". Garis berwarna merah merupakan

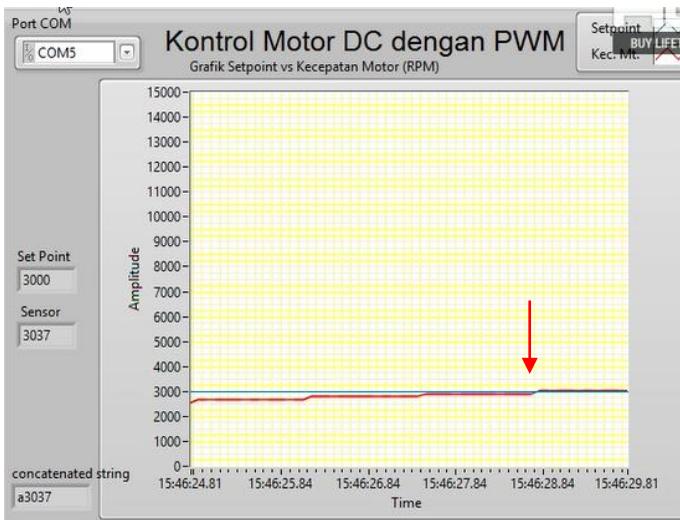
pembacaan sensor yang disimpan pada *variabel* b. Penyimpanan ini digunakan untuk pengiriman data pada *LabVIEW*. Sehingga muncul grafik. Untuk mencapai *setpoint* 2000 rpm waktu yang dibutuhkan 8 detik. Terlihat pada Gambar 4.11 respon kecepatan tepat berada pada *setpoint* pada detik ke-33.

4.3.3 Pengujian *Setpoint* 3000 RPM



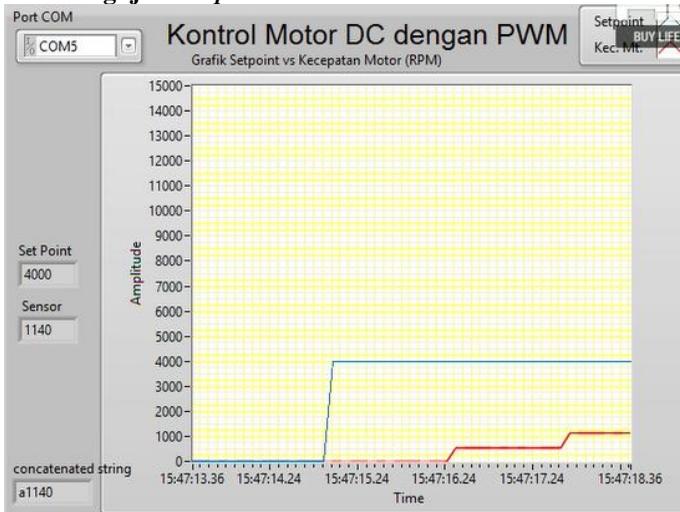
Gambar 4.12 Hasil Pengujian *Rise Time* 3000 RPM

Respon kecepatan motor DC pada Gambar 4.12 bergerak pada detik ke-10. Garis berwarna biru merupakan data dari *keypad* yang disimpan pada *variable* “a”. Garis berwarna merah merupakan pembacaan sensor yang disimpan pada *variabel* b. Penyimpanan ini digunakan untuk pengiriman data pada *LabVIEW*. Untuk mencapai *setpoint* 3000 rpm waktu yang dibutuhkan 18 detik. Terlihat pada Gambar 4.13 tanda anak panah menunjukkan respon kecepatan tepat berada pada *setpoint* pada detik ke-28.

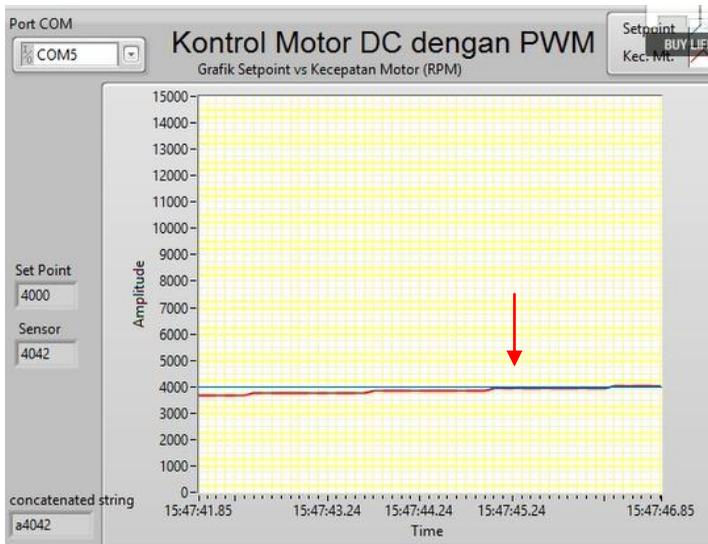


Gambar 4.13 Pengujian *Rise Time* Tepat pada *Setpoint* 3000 RPM

4.3.4 Pengujian *Setpoint* 4000 RPM



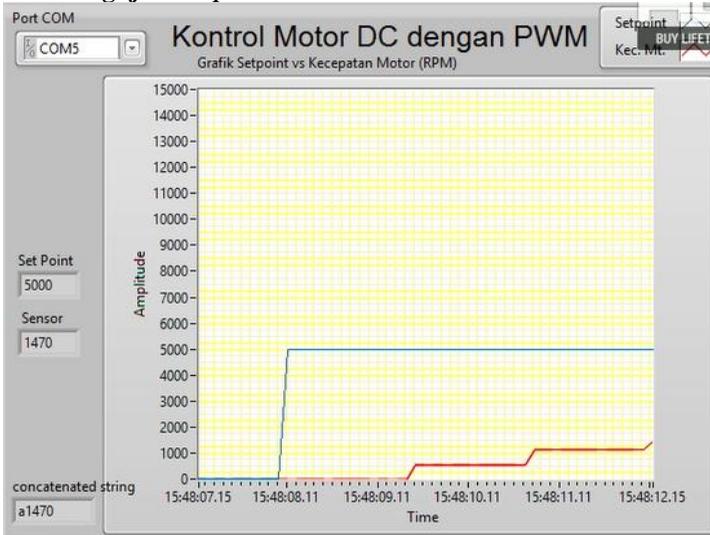
Gambar 4.14 Hasil Pengujian *Rise Time* 4000 RPM



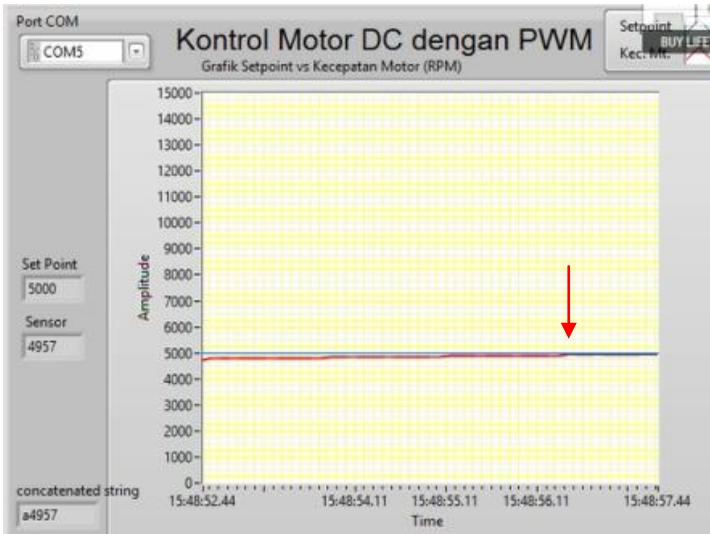
Gambar 4.15 Pengujian *Rise Time* Tepat pada *Setpoint* 4000 RPM

Respon kecepatan motor DC pada Gambar 4.14 bergerak pada detik ke-16. Garis berwarna biru merupakan data dari *keypad* yang disimpan pada *variable* “a”. Garis berwarna merah merupakan pembacaan sensor yang disimpan pada *variabel* b. Penyimpanan ini digunakan untuk pengiriman data pada *LabVIEW*. Untuk mencapai *setpoint* 4000 rpm waktu yang dibutuhkan 29 detik. Terlihat pada Gambar 4.15 tanda anak panah menunjukkan respon kecepatan tepat berada pada *setpoint* pada detik ke-45.

4.3.5 Pengujian *Setpoint* 5000 RPM



Gambar 4.16 Hasil Pengujian *Rise Time* 5000 RPM



Gambar 4.17 Pengujian *Rise Time* Tepat pada *Setpoint* 5000 RPM

Respon kecepatan motor DC pada Gambar 4.16 bergerak pada detik ke-9. Garis berwarna biru merupakan data dari *keypad* yang disimpan pada *variable* "a". Garis berwarna merah merupakan pembacaan sensor yang disimpan pada *variabel* b. Penyimpanan ini digunakan untuk pengiriman data pada *LabVIEW*. Untuk mencapai *setpoint* 5000 rpm waktu yang dibutuhkan 48 detik. Terlihat pada Gambar 4.17 tanda anak panah menunjukkan respon kecepatan tepat berada pada *setpoint* pada detik ke-56.

Tabel 4.5 Hasil Pengujian Rata-Rata *Rise Time* 1000-5000 rpm.

NO	<i>Range Set Point</i>	Rentang waktu mencapai <i>setpoint</i>
1	1000 rpm	1 detik
2	2000 rpm	8 detik
3	3000 rpm	18 detik
4	4000 rpm	29 detik
5	5000 rpm	48 detik
Rata – rata		24 detik

Pada Tabel 4.5 terdapat rata-rata waktu untuk mencapai *setpoint* pada *range* 1000-5000 rpm yaitu 24 detik.

(halaman ini sengaja dikosongkan)

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian dan analisa terhadap mengatur kecepatan motor DC *spindle* pada CNC *portable* dapat diambil beberapa kesimpulan sebagai berikut :

1. Sensor *optocoupler* dapat digunakan dalam pembacaan kecepatan motor yang memiliki spesifikasi kurang lebih 12.000 rpm. Dapat dibandingkan dengan alat ukur kecepatan *tachometer*. Seperti pada Tabel 4.1 dan Tabel 4.2 pada tegangan maksimal 48V sensor membaca 12.170 rpm sedangkan pada alat ukur *tachometer* 12.155 rpm. Dapat dilihat bahwa antara sensor *optocoupler* dan *tachometer* sama-sama membaca kecepatan pada *range* 12.000 rpm.
2. Rata-rata selisih kecepatan antara sensor *optocoupler* dan alat ukur *tachometer* adalah 12,05 rpm.
3. Mengatur kecepatan menggunakan metode PWM dapat diterapkan pada motor DC *spindle*, akan tetapi jika ingin nilai *setpoint* antara 1000-5000 rpm akan membutuhkan *risetime* rata-rata 24 detik untuk mencapai *setpoint*.

5.2 Saran

Dari hasil perancangan tugas akhir ini masih kurang sempurna sehingga ada beberapa yang harus diperbaiki agar hasil tugas akhir ini mendekati sempurna yaitu:

1. Mengatur kecepatan motor DC *spindle* lebih baik menggunakan kontroller lain. Karena pada saat mencapai *setpoint* nilai *risetime* tidak membutuhkan waktu yang lama.
2. Penggunaan *driver* motor DC *spindle* harus sesuai dengan frekuensi arduino, agar metode PWM ini dapat sesuai dengan pengukuran dan perhitungan.

(halaman ini sengaja dikosongkan)

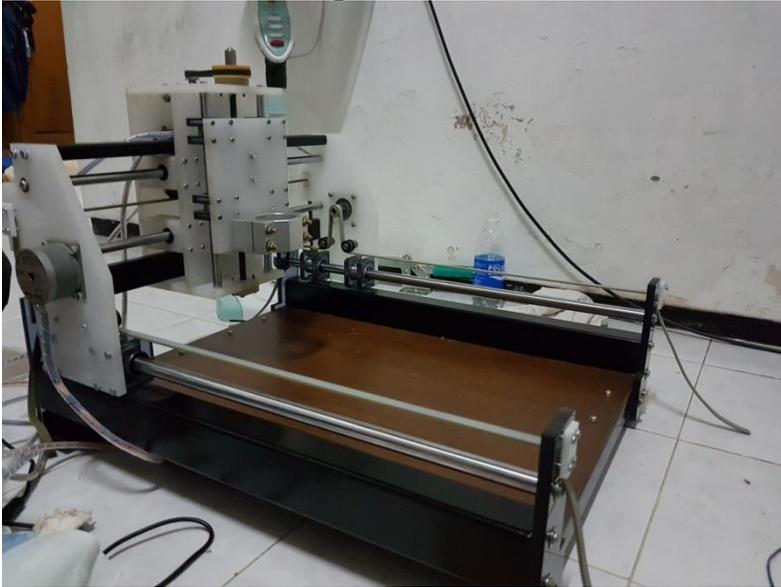
DAFTAR PUSTAKA

- [1] Arduino, <https://www.arduino.cc> (Diakses tanggal 5 Maret 2017).
- [2] Elektronika Dasar, Prinsip Kerja Motor DC, <https://elektronika-dasar.web.id/prinsip-kerja-motor-dc/> (diakses pada tanggal 5 Maret 2017).
- [3] Elektronika Dasar, Matrix Keypad 4x4, <https://elektronika-dasar.web.id/matrix-keypad-4x4> (diakses pada tanggal 5 Maret 2017)
- [4] Elektronika Dasar, LCD Liquid Cristal Display, <https://elektronika-dasar.web.id/lcd-liquid-cristal-display> (diakses pada tanggal 5 Maret 2017)^[4]
- [5] Hafidhillisan, Rizal, 2016, PERANCANGAN CNC (COMPUTER NUMERICAL CONTROL) PORTABLE PADA MESIN PLOTTER BERBASIS MIKROKONTROLER, Tugas Akhir Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.
- [6] It-Jurnal. Pengertian dan Kelebihan, <https://www.it-jurnal.com/pengertian-dan-kelebihan-arduino/> (Diakses tanggal 5 Maret 2017).
- [7] Kief, Hans B. And Waters, T. Frederick. "Computer Numerical Control". California: Glence,1992
- [8] Mochtar Wijaya, ST,2001, "Dasar dasar Mesin Listrik", Djambatan, Jakarta.
- [9] Partner 3D, Motor Stepper; Pengertian, Cara Kerja dan Jenis-jenisnya, www.partner3d.com/motor-stepper-pengertian-cara-kerja-dan-jenis-jenisnya/ (diakses pada tanggal 5 Maret 2017)
- [10] Syaiful, Muhammad, 2007, SISTEM PENGENDALI MOTOR DC DENGAN PWM, Tugas Akhir Teknik Elektro, Universitas Mercu Buana.
- [11] Widarto. 2008. Teknik Pemesinan untuk Sekolah Menengah Kejuruan, Departmen Pendidikan Nasional, Direktorat Pembinaan SMK, Jakarta.
- [12] Yudistiro, Ardi 2011, PENERAPAN SENSOR OPTOCOUPLER PADA ALAT PENGUKUR KECEPATAN, Tugas Akhir Teknik Fisika, Universitas Negeri Malang.
- [13] GRBL. <https://github.com/grbl/grbl/wiki> (diakses pada tanggal 26 Februari 2017).

- [14] Bitter, Rick, Taqi Mohiuddin, dan Matt Nawrocki. LabVIEW : Advanced Programming Techniques. New York : CRC Press. 2006.

LAMPIRAN A

- Dokumentasi hasil pembuatan alat



(halaman ini sengaja dikosongkan)

LAMPIRAN B

- **Program Sensor Kecepatan**

```
// read RPM
int half_revolutions = 0;
int rpm = 0;
unsigned long lastmillis = 0;
void setup(){
  Serial.begin(9600);
  attachInterrupt(0, rpm_fan, FALLING);
}
void loop(){
  if (millis() - lastmillis == 1000){ //Uptade every one second, this will be
  equal to reading frecueny (Hz).
  detachInterrupt(0); //Disable interrupt when calculating
  rpm = half_revolutions * 15; // Convert frecueny to RPM, note: this
  works for one interruption per full rotation. For two interrups per full
  rotation use half_revolutions * 30.
  Serial.print("RPM =\t"); //print the word "RPM" and tab.
  Serial.println(rpm); // print the rpm value.
  Serial.print("\t Hz=\t"); //print the word "Hz".
  Serial.println(half_revolutions); //print revolutions per second or Hz.
  And print new line or enter.
  half_revolutions = 0; // Restart the RPM counter
  lastmillis = millis(); // Uptade lasmillis
  attachInterrupt(0, rpm_fan, FALLING); //enable interrupt
  }
}
// this code will be executed every time the interrupt 0 (pin2) gets low.
void rpm_fan(){
  half_revolutions++;
}
```

(halaman ini sengaja dikosongkan)

LAMPIRAN C

- **Program Keypad**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 ,2,1,0,4,5,6,7,3, POSITIVE);

int col1=7;
int col2=6;
int col3=5;
int col4=4;
int row1=11;
int row2=10;
int row3=9;
int row4=8;
int buttonState_row1=0;
int buttonState_row2=0;
int buttonState_row3=0;
int buttonState_row4=0;
long int nilai=0, value,array[10];
int i,a,j,indeks=0;
char temp[16];

void read_button(){
    buttonState_row1 = digitalRead(row1);
    buttonState_row2 = digitalRead(row2);
    buttonState_row3 = digitalRead(row3);
    buttonState_row4 = digitalRead(row4);
}
void simpan_variabel(){
    if (i==1){
        nilai=array[i];
        //itoa(array[i],temp);
        lcd.setCursor(i+6,0);
        lcd.print(array[i]);
        //Serial.print(array[i]);
        delay(500);
        //lcd.setCursor(i+6,0);
        //lcd.print("*");
    }
}
```

```

    //Serial.print("*");
}
if (i>=2 && i<=8){
    nilai=(nilai*10)+array[i];
    //itoa(array[i],temp);
    lcd.setCursor(i+6,0);
    lcd.print(array[i]);
    delay(500);
    //Serial.print(array[i]);
    //delay(500);
    //lcd.setCursor(i+5,0);
    //lcd.print("*");
    //Serial.print("*");
}
}
void hapus(){
    //c++;
    nilai=(nilai-array[i])/10;
    //itoa(nilai,temp);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Target:");
    /*for(a=0;a<i;a++){
        lcd.setCursor(a,1);
        lcd.print("*");
    } */
    lcd.setCursor(7,0);
    lcd.print(nilai);
}
void enter(){
    indeks=1;
    lcd.setCursor(0,1);
    lcd.print("OK");
    Serial.print("b");
    Serial.println(nilai);

    //lcd.clear();
}
void keypad(){

```

```

//lcd.setCursor(0,0);
read_button();
digitalWrite(col1,LOW);
digitalWrite(col2,HIGH);
digitalWrite(col3,HIGH);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){delay(50);} //1 //i++;
array[i]=1; simpan_dlm_1variabel(); delay_ms(300);
else if(buttonState_row2==LOW){i++; array[i]=3; simpan_variabel();
delay(50);} //4
else if(buttonState_row3==LOW){i++; array[i]=2; simpan_variabel();
delay(50);} //7
else if(buttonState_row4==LOW){i++; array[i]=1; simpan_variabel();
delay(50);} //A

read_button();
digitalWrite(col1,HIGH);
digitalWrite(col2,LOW);
digitalWrite(col3,HIGH);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){i--; hapus(); delay(50);} //1
//i++; array[i]=1; simpan_dlm_1variabel(); delay_ms(300);
else if(buttonState_row2==LOW){ delay(50);} //4
else if(buttonState_row3==LOW){ i++; array[i]=0;
simpan_variabel(); delay(50);} //7
else if(buttonState_row4==LOW){enter(); delay(50);}
//A

read_button();
digitalWrite(col1,HIGH);
digitalWrite(col2,HIGH);
digitalWrite(col3,LOW);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){ delay(50);} //1 //i++;
array[i]=1; simpan_dlm_1variabel(); delay_ms(300);

```

```

    else if(buttonState_row2==LOW){i++; array[i]=9; simpan_variabel();
delay(50);} //4
    else if(buttonState_row3==LOW){i++; array[i]=8; simpan_variabel();
delay(50);} //7
    else if(buttonState_row4==LOW){i++; array[i]=7; simpan_variabel();
delay(50);} //*/ //A

    read_button();
    digitalWrite(col1,HIGH);
    digitalWrite(col2,HIGH);
    digitalWrite(col3,HIGH);
    digitalWrite(col4,LOW);
    delay(50);
    if(buttonState_row1==LOW){delay(50);} //1 //i++;
array[i]=1; simpan_dlm_1variabel(); delay_ms(300);
    else if(buttonState_row2==LOW){i++; array[i]=6; simpan_variabel();
delay(50);} //4
    else if(buttonState_row3==LOW){i++; array[i]=5; simpan_variabel();
delay(50);} //7
    else if(buttonState_row4==LOW){i++; array[i]=4; simpan_variabel();
delay(50);} //*/ //A
}

void setup(){
    Serial.begin(9600);
    pinMode(col1,OUTPUT);
    pinMode(col2,OUTPUT);
    pinMode(col3,OUTPUT);
    pinMode(col4,OUTPUT);
    pinMode(row1,INPUT_PULLUP);
    pinMode(row2,INPUT_PULLUP);
    pinMode(row3,INPUT_PULLUP);
    pinMode(row4,INPUT_PULLUP);
    lcd.begin(16,2);
}

void loop(){
    lcd.setCursor(0,0);
    lcd.print("Target:");
    keypad();}

```

LAMPIRAN D

- **Program Mengatur Kecepatan Motor DC**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 ,2,1,0,4,5,6,7,3, POSITIVE);

int col1=7;
int col2=6;
int col3=5;
int col4=4;
int row1=11;
int row2=10;
int row3=9;
int row4=8;
int buttonState_row1=0;
int buttonState_row2=0;
int buttonState_row3=0;
int buttonState_row4=0;
long int nilai=0, value,array[10];
int i,a,j,indeks=0;
char temp[16];

void read_button(){
  buttonState_row1 = digitalRead(row1);
  buttonState_row2 = digitalRead(row2);
  buttonState_row3 = digitalRead(row3);
  buttonState_row4 = digitalRead(row4);
}
void simpan_variabel(){
  if (i==1){
    nilai=array[i];
    //itoa(array[i],temp);
    lcd.setCursor(i+6,0);
    lcd.print(array[i]);
    //Serial.print(array[i]);
    delay(500);
    //lcd.setCursor(i+6,0);
    //lcd.print("*");
  }
}
```

```

    //Serial.print("*");
}
if (i>=2 && i<=8){
    nilai=(nilai*10)+array[i];
    //itoa(array[i],temp);
    lcd.setCursor(i+6,0);
    lcd.print(array[i]);
    delay(500);
    //Serial.print(array[i]);
    //delay(500);
    //lcd.setCursor(i+5,0);
    //lcd.print("*");
    //Serial.print("*");
}
}
}
void hapus(){
    //c++;
    nilai=(nilai-array[i])/10;
    //itoa(nilai,temp);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Target:");
    /*for(a=0;a<i;a++){
        lcd.setCursor(a,1);
        lcd.print("*");
    } */
    lcd.setCursor(7,0);
    lcd.print(nilai);
}
void enter(){
    indeks=1;
    lcd.setCursor(0,1);
    lcd.print("OK");
    Serial.print("b");
    Serial.println(nilai);
    //lcd.clear();
}
void keypad(){
    //lcd.setCursor(0,0);

```

```

read_button();
digitalWrite(col1,LOW);
digitalWrite(col2,HIGH);
digitalWrite(col3,HIGH);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){delay(50);} //1 //i++;
array[i]=1; simpan_dlm_1 variabel(); delay_ms(300);
else if(buttonState_row2==LOW){i++; array[i]=3; simpan_variabel();
delay(50);} //4
else if(buttonState_row3==LOW){i++; array[i]=2; simpan_variabel();
delay(50);} //7
else if(buttonState_row4==LOW){i++; array[i]=1; simpan_variabel();
delay(50);} //A //A
read_button();
digitalWrite(col1,HIGH);
digitalWrite(col2,LOW);
digitalWrite(col3,HIGH);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){i--; hapus(); delay(50);} //1
//i++; array[i]=1; simpan_dlm_1 variabel(); delay_ms(300);
else if(buttonState_row2==LOW){ delay(50);} //4
else if(buttonState_row3==LOW){ i++; array[i]=0;
simpan_variabel(); delay(50);} //7
else if(buttonState_row4==LOW){enter(); delay(50);}
//A //A
read_button();
digitalWrite(col1,HIGH);
digitalWrite(col2,HIGH);
digitalWrite(col3,LOW);
digitalWrite(col4,HIGH);
delay(50);
if(buttonState_row1==LOW){ delay(50);} //1 //i++;
array[i]=1; simpan_dlm_1 variabel(); delay_ms(300);
else if(buttonState_row2==LOW){i++; array[i]=9; simpan_variabel();
delay(50);} //4
else if(buttonState_row3==LOW){i++; array[i]=8; simpan_variabel();
delay(50);} //7

```

```

    else if(buttonState_row4==LOW){i++; array[i]=7; simpan_variabel();
delay(50);}          /*          //A

    read_button();
    digitalWrite(col1,HIGH);
    digitalWrite(col2,HIGH);
    digitalWrite(col3,HIGH);
    digitalWrite(col4,LOW);
    delay(50);
    if(buttonState_row1==LOW){delay(50);}          //1 //i++;
array[i]=1; simpan_dlm_1variabel(); delay_ms(300);
    else if(buttonState_row2==LOW){i++; array[i]=6; simpan_variabel();
delay(50);}          //4
    else if(buttonState_row3==LOW){i++; array[i]=5; simpan_variabel();
delay(50);}          //7
    else if(buttonState_row4==LOW){i++; array[i]=4; simpan_variabel();
delay(50);}          /*          //A
}

void setup(){
    Serial.begin(9600);
    pinMode(col1,OUTPUT);
    pinMode(col2,OUTPUT);
    pinMode(col3,OUTPUT);
    pinMode(col4,OUTPUT);
    pinMode(row1,INPUT_PULLUP);
    pinMode(row2,INPUT_PULLUP);
    pinMode(row3,INPUT_PULLUP);
    pinMode(row4,INPUT_PULLUP);
    lcd.begin(16,2);
}

void loop(){
    lcd.setCursor(0,0);
    lcd.print("Target:");
    keypad();
}

```

RIWAYAT HIDUP PENULIS



Nama : Fahmi Addinul Haq
TTL : Surabaya, 14 April 1995
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Jalan Gubeng Klingsingan 3
no 8 Surabaya
Telp/HP : 085335405505
E-mail : fahmi14041995@gmail.com

RIWAYAT PENDIDIKAN

- 2001 – 2007 : SDN Gubeng 3 Surabaya
- 2007 – 2010 : SMP Muhammadiyah 9 Surabaya
- 2010 – 2013 : SMA IPIEMS Surabaya
- 2014 – 2017 : D3 Teknik Elektro Otomasi, Program Studi Teknik Elektro Komputer Kontrol- Fakultas Vokasi Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

- Kerja Praktek di PT Telkom MSC V Jatim (Juni - Juli 2015).

PENGALAMAN ORGANISASI

- Staff Departemen Dalam Negeri Tahun 2014/2015.
- Seketaris 2 Lembaga Dakwah Jurusan Salman AL-Farisi 2016/2017

