



TUGAS AKHIR - TE 145561

**PERANCANGAN CLIENT SERVER PADA SISTEM ANTARMUKA
SERTA PROSES KALIBRASI SUMBU Y MESIN CNC PORTABLE**

Prayoga Riono P.
NRP 2214030016

Dosen Pembimbing
Ir. Djoko Suprajitno Rahardjo, MT.
Andri Ashfahani, ST., MT., M.Sc.

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



FINAL PROJECT - TE 145561

***CLIENT SERVER DESIGNING IN INTERFACE SYSTEM AND
CALIBRATION AXIS Y MACHINE CNC PORTABLE***

Prayoga Riono P.
NRP 2214030016

Advisor
Ir. Djoko Suprajitno Rahardjo, MT.
Andri Ashfahani, ST., MT., 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 kami menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul "**PERANCANGAN CLIENT SERVER PADA SISTEM ANTARMUKA SERTA PROSES KALIBRASI SUMBU Y MESIN CNC PORTABLE**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan ya pihak lain yang kami akui sebagai karya sendiri.

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

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

Surabaya, 19 Juli 2017



Prayoga Riono P.
NRP 2214030016

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN CLIENT SERVER PADA SISTEM ANTARMUKA
SERTA PROSES KALIBRASI SUMBU Y 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

Dosen Pembimbing 1

Dosen Pembimbing 2

Ir. Djoko Suprajitno, Rahardjo, MT., Andri Ashfahani, ST., MT., M.Sc.
NIP. 195506221987011001 NIP. 2200201405003

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN CLIENT SERVER PADA SISTEM ANTARMUKA SERTA PROSES KALIBRASI SUMBU Y MESIN CNC PORTABLE

Nama : Prayoga Riono P.
Pembimbing : Ir. Djoko Suprajitno Rahardjo, MT.
Andri Ashfahani, ST., MT., M.Sc.

ABSTRAK

Dalam pemrograman web, dikenal istilah Client-side dan Server-side *Scripting*. Dua istilah tersebut digunakan untuk mengelompokkan beberapa bahasa pemrograman web berdasarkan pihak yang mana berfungsi untuk pengolahan data di web. Untuk penjelasan Client-side sendiri adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer penggunaan, dan Server side sendiri adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer. Sebagai bagian dari konteks data, Client Server mengatur interface yang fungsinya sebagai tempat menjalankan aplikasi berbasis data. Pada tugas akhir ini telah dibuat suatu client dan server yang di integrasikan menjadi satu untuk mengendalikan mesin CNC, yang mana nantinya CNC Portable dapat digunakan *engraving* atau dengan istilah lain untuk menggambar suatu pola secara langsung dengan pemanggilan kode G-Code dari *software* Grbl dengan antarmuka Client Server. Pada tugas akhir ini akan dilakukan pengujian pergerakan mesin CNC terhadap sumbu Y dengan proses kalibrasi yang didapat sebuah data rata-rata error sebesar 0,12 mm dan nilai step baru sebesar 16,61 steps/mm. Jadi apabila diberikan sebuah perintah dengan target 9 mm, mesin mampu menirukan pergerakan dengan perpindahan sebesar 8,9 mm.

Kata Kunci : CNC, G-Code, *software* Grbl, *interface* CAD/CAM.

-----Halaman ini sengaja dikosongkan-----

CLIENT SERVER DESIGNING IN INTERFACE SYSTEM AND CALIBRATION AXIS Y MACHINE CNC PORTABLE

Nama : Prayoga Riono P.
Pembimbing : Ir. Djoko Suprajitno Rahardjo, MT.
Andri Ashfahani, ST., MT., M.Sc.

ABSTRACT

In web programming, known term Client-side and Server-side Scripting. The two terms are used to group some web programming languages based on the party which serves for data processing on the web. For explanation Client-side itself is a web programming language that data processing is done by computer usage, and Server side itself is a web programming language that data processing is done by computer. As part of the data context, Client Server manages interfaces that function as a place to run data-driven applications. In this final project has created a client and server that integrate into one to control the CNC machine, which later CNC Portable can be used engraving or with another term to draw a pattern directly by calling the code G-Code from Grbl software with Client interface Server. In this final project will be testing the movement of CNC machine to the Y axis with the calibration process obtained sebebaut average data error of 0.12 mm and a new step value of 16.61 steps / mm. So when given a command with a target of 9 mm, the machine is able to mimic the movement with a displacement of 8.9 mm.

Keywords: CNC, G-Code, Grbl software, CAD/CAM interface

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

PERANCANGAN CLIENT SERVER PADA SISTEM ANTARMUKA SERTA PROSES KALIBRASI SUMBU Y MESIN CNC PORTABLE

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Djoko Suprajitno Rahardjo, M.T dan Bapak Andri Ashfahani ST.,MT.,M.Sc. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
BAB II TEORI PENUNJANG.....	5
2.1 CNC (<i>Computer Numerical Control</i>).....	5
2.1.1 <i>Cartesian Coordinate System</i>	6
2.1.2 Jenis-jenis Mesin CNC	7
2.1.3 Cara Kerja Mesin CNC	8
2.1.4 CNC Portable	8
2.2 Driver A4988	8
2.3 Arduino	9
2.4 Sistem Program	10
2.4.1 Sistem Inkremental.....	10
2.4.2 Sistem Absolute.....	11
2.4.3 Interpolasi.....	11
2.4.4 Continuous Path (<i>Contouring</i>)	11
2.4.5 <i>Point to Point</i>	12
2.5 <i>G-Code</i>	12
2.6 <i>Software Grbl</i>	12

2.7 Pemrograman Web	13
2.7.1 Javascript	13
2.8 Node Server	15
2.9 Sublime Text 3.....	16
BAB III PERANCANGAN SISTEM.....	17
3.1 Konfigurasi Keseluruhan Sistem CNC Portable	17
3.1.1 Perancangan Blok Sistem Kerja.....	17
3.1.2 Perancangan Proses Pengerjaan	18
3.2 Perancangan Sistem Mekanik CNC Portable terhadap Sumbu Y20	
3.2.1 Perancangan Mekanik Sumbu Y.....	20
3.3 Pembuatan <i>Box Controller</i>	21
3.4 Konfigurasi Elektrik pada CNC Portable.....	23
3.4.1 Pembuatan Rangkaian <i>Shield Driver</i> A4988	23
3.4.2 Pembuatan Rangkaian <i>Shield</i> Arduino.....	24
3.4.3 Koneksi Driver pada Arduino	26
3.4.4 Instalasi Arduino IDL pada Raspberry Pi	28
3.5 Perancangan <i>Software</i> Grbl.....	28
3.5.1 Proses Kompilasi Grbl	28
3.5.2 Konfigurasi Grbl	29
3.5.2.1 Pengaturan Grbl.....	30
3.5.2.2 Penjelasan Pengaturan Perbagian pada Grbl.....	31
3.6 Perancangan Sistem <i>Client</i> dan Server Side pada <i>Interface</i>	32
3.6.1 Perancangan Program Javascript sebagai <i>Client Side</i>	32
3.6.2 Perancangan Sistem Server pada Antarmuka.....	35
BAB IV HASIL PERANCANGAN DAN PENGUJIAN.....	39
4.1 Hasil Perancangan Tampilan Antarmuka	39
4.2 Pengujian Perpindahan Posisi Sumbu Y.....	41
4.2.1 Perpindahan Sumbu Y sebelum Kalibrasi	41
4.2.2 Perpindahan Sumbu Y Setelah Kalibrasi	43
4.3 Pengujian <i>Limit Switch</i>	44
4.4 Pengujian Kode G-Code Terhadap Sumbu Y.....	45
4.4.1 Sistem Inkremental	46
4.4.2 Sistem Absolute	46
4.5 Pengujian G-Code dengan CAD dan CAM	47
4.5.1 Persegi Mode Engrave	47
4.5.2 Persegi Mode Pocket.....	48
4.5.3 Segienam Mode Engrave	49

4.5.4 Segienam Mode Pocket	51
BAB V PENUTUP.....	53
DAFTAR PUSTAKA	55
LAMPIRAN A.....	57
A.1.G-Code	57
LAMPIRAN B	59
B.1 Datasheet Raspberry Pi 3 Model B	59
B.2 Datasheet Driver Stepper A4988	61
LAMPIRAN C	67
C.1 Listing Program Javascript untuk Antarmuka	67
C.2 Listing Program Node Server pada Tampilan Antarmuka.....	73
LAMPIRAN D.....	77
D.1 Dokumentasi Mesin CNC Portable	77
D.2 Dokumentasi Pengujian Kode G-Code.....	78
D.3 Hasil Rangkain Shield Driver Motor dan Shield Arduino	79
DAFTAR RIWAYAT HIDUP	81

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Mesin CNC di Industri [2]	5
Gambar 2.2	<i>Zero Point</i> [3].....	7
Gambar 2.3	Rangkaian Driver A4988 [4].....	9
Gambar 2.4	Sistem Program pada Mesin CNC [3].....	10
Gambar 2.5	Sistem Inkremental [3].....	10
Gambar 2.6	Sistem Absolute [3]	11
Gambar 2.7	Gerak Interpolasi Melingkar [6]	11
Gambar 2.8	Tiga Bagian Terpenting dalam Pemrograman Web [8] ..	13
Gambar 2.9	Salah Satu fungsi Javascript [9]	14
Gambar 2.10	Perbedaan Server Tradisional dengan Node Js [11].....	15
Gambar 2.11	Tampilan Sublime Text.....	16
Gambar 3.1	Perancangan Keseluruhan Sistem Kerja CNC Portable..	17
Gambar 3.2	Proses Kerja dari Keseluruhan Sistem	19
Gambar 3.3	Perancangan Mesin CNC Portable.....	20
Gambar 3.4	Perancangan Desain Sumbu <i>Y</i>	21
Gambar 3.5	Tampak Dalam <i>Box Controller</i>	22
Gambar 3.6	Tampak Luar <i>Box Controller</i>	22
Gambar 3.7	Blok Diagram Arduino dengan Motor Stepper	23
Gambar 3.8	Skematik Rangkaian Shield Driver A4988	24
Gambar 3.9	Skematik Rangkaian <i>Shield</i> Arduino	25
Gambar 3.10	Hasil <i>Routing</i> dari Skematik <i>Shield</i> Arduino	25
Gambar 3.11	Konfigurasi Pin Grbl pada Arduino Uno	26
Gambar 3.12	Koneksi Driver pada Arduino untuk Grbl.....	27
Gambar 3.13	Tampilan Pemilihan Port	33
Gambar 3.14	Tampilan Tombol Kendali	34
Gambar 3.15	Tampilan Upload Kode G-Code	35
Gambar 3.16	Proses Pembuatan Konfigurasi	36
Gambar 3.17	Diagram Blok Client Server pada Tampilan Antarmuka WEB	37
Gambar 4.1	Tampilan Antarmuka Otomatis.....	39
Gambar 4.2	<i>Listing</i> Program Kolom Pengerjaan	40
Gambar 4.3	<i>Listing</i> Program Kolom <i>Input</i>	40
Gambar 4.4	Posisi Awal Mesin CNC	42
Gambar 4.5	Jarak Sebenarnya dengan <i>Target</i> 10 mm	42

Gambar 4.6	Setelah Proses Kalibrasi	44
Gambar 4.7	Gambar Persegi	47
Gambar 4.8	Gambar Persegi Mode Engrave	48
Gambar 4.9	Konversi Gambar Persegi Mode Pocket	48
Gambar 4.10	Hasil Gambar Persegi Mode Pocket	49
Gambar 4.11	Pembuatan Gambar Segienam pada Inkscape	49
Gambar 4.12	Konversi Gambar Segienam Mode Engrave pada Jscut .	50
Gambar 4.13	Hasil Gambar Segienam Mode Engrave	50
Gambar 4.14	Konversi Gambar Segienam Mode Pocket pada Jscut ...	51
Gambar 4.15	Hasil Gambar Segienam Mode Pocket	51

DAFTAR TABEL

	HALAMAN
Tabel 2.1	Nilai Logika pada Driver A49889
Tabel 3.1	Konfigurasi Grbl29
Tabel 3.2	Kode dalam Pengaturan Grbl.....30
Tabel 4.1	Pengujian Sumbu <i>Y</i> Sebelum Kalibrasi.....41
Tabel 4.2	Perhitungan Mencari Nilai <i>New Step</i>43
Tabel 4.3	Perpindahan Sumbu <i>Y</i> Setelah Kalibrasi.....43
Tabel 4.4	Pengukuran Nilai <i>Error Limit Switch</i> Sumbu <i>Y</i>45
Tabel 4.5	Pengukuran Perpindahan Sistem Inkremental.....46
Tabel 4.6	Pengukuran Perpindahan Sistem Absolute46

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman web, dikenal istilah Client-side dan Server-side *Scripting*. Dua istilah tersebut digunakan untuk mengelompokkan beberapa bahasa pemrograman web berdasarkan pihak yang mana berfungsi untuk pengolahan data di web. Untuk penjelasan Client-side sendiri adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer penggunaan, dan server side sendiri adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer [1]. Sedangkan kalau digabung Client Server adalah sebuah pembagian kerja didalam mengakses server sebuah jaringan tertentu antara Client dan Server. Sebagai bagian dari konteks data, Client Server mengatur interface yang fungsinya sebagai tempat menjalankan aplikasi berbasis data. Pada tugas akhir ini telah dibuat suatu Client dan Server yang di integrasikan menjadi satu untuk mengendalikan mesin CNC, yang mana nantinya CNC Portable dapat digunakan *engraving* atau dengan istilah lain untuk menggambar suatu pola secara langsung dengan pemanggilan kode *G-Code* dari *software* GBRL dengan antarmuka Client Server, dan antarmuka web yang berjalan *offline* jadi tetap berjalan meskipun tidak ada koneksi dengan internet Adapun beberapa keuntungan penggunaan mesin perkakas CNC yaitu: produktivitas tinggi, ketelitian pengerjaan tinggi, kualitas produk yang seragam yang dimana ini menggunakan GBRL *controller*. GBRL sendiri merupakan *software* yang digunakan untuk mengontrol gerakan dari sebuah mesin CNC, dan biasanya GBRL mempunyai kode-kode standart. Kode standart ini dinamakan dengan *G-Code*, sedangkan proses pengiriman kode dari Kontroller ke mesin CNC dinamakan dengan *G-Code Sender*. Pada dasarnya *G-Code* dibuat *open source*, dimana ini dapat dikembangkan dalam ratusan proyek termasuk proyek dengan pemotong laser, penulis tangan otomatis, pengebor lubang, pelukis gravity dan mesin drawing eksentrik. Itu dikarenakan kesederhanaan dalam hal kinerja dan pembiayaan *hardware* yang hemat, serta GBRL telah tumbuh menjadi sebuah *open source* yang fenomenal pada masa kini.

1.2 Rumusan Masalah

Antarmuka yang dibuat dari internet pasti tidak bisa berjalan kalau tanpa koneksi, jadi apabila sebuah mesin dikendalikan dengan antarmuka web pasti tidak dapat dioperasikan oleh pengguna. Dan mesin CNC pada umumnya mempunyai tampilan antarmuka yang sulit untuk dioperasikan sehingga sebagai seorang pengguna pemula akan merasa kesulitan jika mengoperasikan sebuah mesin CNC. Tempat pengoperasian melalui komputer juga memiliki beberapa kekurangan diantaranya, kurang fleksibel apabila digunakan pada era modern ini serta biaya yang digunakan untuk membuat tampilan antarmuka mesin CNC mahal.

1.3 Batasan Masalah

Untuk menghindari kesalah pahaman dan meluasnya bahasan mengenai alat yang akan dibuat, maka penulis membatasi atau memfokuskan masalah yang berkaitan dengan mesin CNC Portable, yaitu sebagai berikut:

1. Penulis lebih memfokuskan pada pengujian aksis Y
2. Pada tampilan antar muka web berfokus pada pemrograman JavaScript.
3. Antarmuka web ini hanya bisa memsukkan file berupa *G-Code*.
4. Serta dalam pembuatan Server hanya dibuat sebatas *offline*, dengan Node Server.js.

1.4 Tujuan

Dengan tampilan antarmuka yang diprogram menggunakan pemrograman web, didapatkan suatu pengoperasian yang lebih *user-friendly* serta dapat dikembangkan lagi sesuai dengan kebutuhan pengguna. Pada desain yang dibuat, penggunaan tombol pada tampilan antarmuka digantikan dengan layar sentuh untuk meminimalkan kemungkinan terjadinya kerusakan serta menambah kesan yang lebih ringkas jika dibandingkan dengan menggunakan tombol-tombol bersifat mekanik.

1.5 Metodologi Penelitian

Perancangan Client Server pada sistem antarmuka, terbagi menjadi empat tahapan, yaitu studi literatur, perancangan sistem, uji coba dan hasil pengujian, serta pembuatan laporan.

Pada tahap studi literatur ini akan dilakukan pencarian data, bahan, dan literatur. Dimana literatur diperoleh dari *Design of Control Unit for CNC Machine Tool using Arduino based Embedded System*. Dari studi literatur tersebut didapatkan bagaimana perancangan mesin CNC menggunakan Arduino dan pembuatan Client dan Server untuk sistem antarmukanya. Akhirnya dari beberapa artikel yang didapatkan, memunculkan sebuah ide membuat mesin CNC Portable yang bisa dikontrol secara manual maupun dengan interface CAD/CAM.

Pada tahap perancangan sistem akan dilakukan sebuah perancangan mengenai garis besar kerja sistem yang dibuat, perancangan mekanik, perancangan *hardware* atau konfigurasi elektrik pada mesin CNC Portable, perancangan sistem antarmuka ini di bangun menggunakan pemrograman JavaScript dan Node Server untuk sistem Client dan Servernya. Serta menggunakan arduino sebagai controller dalam pembacaan kode *G-Code*.

Dengan proses uji coba dan analisa terhadap rancangan sistem diharapkan akan membantu dan mempermudah mahasiswa apabila rancangan sistem mengalami kesalahan atau kegagalan. Beberapa pengujian yang dilakukan pada rancangan sistem yaitu:

- a. Pengujian antarmuka mesin CNC terhadap sumbu Y.
- b. Pengujian kalibrasi mesin CNC terhadap sumbu Y.
- c. Pengujian sensor limit switch terhadap sumbu Y.
- d. Pengujian antarmuka untuk mendapatkan beberapa gambar CAD.

Setelah pembuatan alat semuanya terselesaikan dengan baik dan benar, mahasiswa dituntut untuk membuat sebuah buku laporan sebagai bukti dan hasil yang telah dicapai selama ini pada saat pengerjaan alat hingga selesai. Dengan disusunnya buku laporan ini diharapkan dapat memberikan wawasan dan ilmu pengetahuan yang cukup luas kepada mahasiswa lain membacanya, sehingga dapat dijadikan acuan dan motivasi untuk mengembangkan kreativitas untuk membuat karyanya sendiri.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari CNC Portable, Driver A4988, Arduino, Sistem Program, G-Code, Grbl, Pemrograman Web, Node Server.

Bab III Perancangan Sistem

Bab ini membahas konfigurasi keseluruhan sistem mesin CNC Portable, perancangan mekanik sumbu *Y* CNC Portable, Pembuatan Box Elektrik, Konfigurasi Elektrik Pada CNC Portable, Perancangan *software* Grbl, dan Perancangan Sistem Client dan Server pada Sistem Antarmuka.

Bab IV Hasil Perancangan dan Pengujian

Bab ini memuat hasil perancangan serta pengujian \ dari hasil pengujian

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat digunakan oleh masyarakat sekitar yang memiliki industri skala kecil seperti pengerajin atau pengukir kayu untuk dapat memudahkan pekerjaan mereka serta dapat menjadi referensi untuk penelitian selanjutnya.

BAB II

TEORI PENUNJANG

Pada bab teori penunjang ini akan membahas mengenai materi-materi yang akan digunakan sebagai bahan pengerjaan tugas akhir mulai dari penjelasan mesin CNC, *software-software* yang digunakan dalam pembuatan tugas akhir ini sebagai berikut:

2.1 CNC (*Computer Numerical Control*)

CNC adalah mesin yang dipergunakan untuk pengontrolan otomatis dalam dunia industri. Mesin ini berfungsi untuk mengontrol kinerja mesin-mesin lain yang dipergunakan. NC/CNC (*Numerical Control/Computer Numerical Control*) merupakan istilah yang digunakan untuk menunjukkan bahwa suatu peralatan manufaktur; misalnya bubut, milling, dll; dikontrol secara numerik berbasis komputer yang mampu membaca instruksi kode N, G, F, T, dan lain-lain, dimana kode-kode tersebut akan menginstruksikan ke mesin CNC agar bekerja sesuai dengan program benda kerja yang akan dibuat mengoperasikannya [1]. Dengan mesin CNC, ketelitian suatu produk dapat dijamin hingga 1/1000 mm lebih, pengerjaan produk masal dengan hasil yang sama persis dan waktu permesinan yang cepat. Untuk mesin CNC yang sering digunakan dalam dunia industri bisa di lihat pada Gambar 2.1 bawah ini:



Gambar 2.1 Mesin CNC di Industri [2]

Fungsi CNC dalam hal ini lebih banyak menggantikan pekerjaan operator dalam mesin perkakas konvensional. Misalnya pekerjaan setting tool atau mengatur gerakan pahat sampai pada posisi siap memotong, gerakan pemotongan dan gerakan kembali keposisi awal,

dan lain-lain. Demikian pula dengan pengaturan kondisi pemotongan (kecepatan potong, kecepatan makan dan kedalaman pemotongan) serta fungsi pengaturan yang lain seperti penggantian pahat, perubahan transmisi daya (jumlah putaran poros utama), dan arah putaran poros utama, pengekleman, pengaturan cairan pendingin dan sebagainya.

Mesin perkakas CNC dilengkapi dengan berbagai alat potong yang dapat membuat benda kerja secara presisi dan dapat melakukan interpolasi yang diarahkan secara numerik (berdasarkan angka). Parameter sistem operasi CNC dapat diubah melalui program perangkat lunak (*software load* program) yang sesuai. CNC telah banyak dipergunakan dalam industri logam. Dalam kondisi ini, CNC dipergunakan untuk mengontrol sistem mekanis mesin-mesin perkakas dan pemotong logam. Jadi seberapa tebal dan panjangnya potongan logam yang dihasilkan oleh mesin pemotong logam, dapat diatur oleh mesin CNC. [2]

Tuntutan konsumen yang menghendaki kualitas benda kerja yang presisi, berkualitas sama baiknya, dalam waktu singkat dan dalam jumlah yang banyak, akan lebih mudah dikerjakan dengan mesin perkakas CNC (*Computer Numerically Controlled*), yaitu mesin yang dapat bekerja melalui pemrograman yang dilakukan dan dikendalikan melalui komputer. Mesin CNC dapat bekerja secara otomatis atau semi otomatis setelah diprogram terlebih dahulu melalui komputer yang ada. Program yang dimaksud merupakan program membuat benda kerja yang telah direncanakan atau dirancang sebelumnya. Sebelum benda kerja tersebut dieksekusi atau dikerjakan oleh mesin CNC, sebaiknya program tersebut di cek berulang-ulang agar program benar-benar telah sesuai dengan bentuk benda kerja yang diinginkan, serta benar-benar dapat dikerjakan oleh mesin CNC. Pengecekan tersebut dapat melalui layar monitor yang terdapat pada mesin atau bila tidak ada fasilitas cheking melalui monitor dapat pula melalui plotter yang dipasang pada tempat dudukan pahat/palsu frais. Setelah program benar-benar telah berjalan seperti rencana, baru kemudian dilaksanakan/dieksekusi oleh mesin CNC.

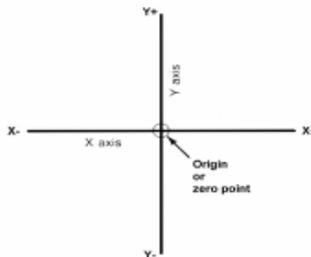
2.1.1 Cartesian Coordinate System

Hampir segala sesuatu yang dapat diproduksi pada alat mesin konvensional dapat diproduksi pada kontrol numerik komputer alat mesin, dengan banyak keuntungan. Gerakan alat mesin yang digunakan dalam memproduksi suatu produk terdiri dari dua jenis dasar: (gerakan

garis lurus) *point to point* dan *continuous path* (gerakan berkelanjutan) gerakan). [3]

Cartesian, atau persegi panjang merupakan sistem koordinat yang dirancang oleh matematikawan dan filsuf Perancis Rene Decartes. Dengan sistem ini, setiap titik tertentu dapat dijelaskan dalam matematika bahwa titik lain sepanjang tiga sumbu merupakan garis tegak lurus. Konsep ini sangat cocok untuk peralatan mesin CNC karena konstruksi umumnya didasarkan pada tiga sumbu gerak (X , Y , Z) ditambah sumbu rotasi. Pada mesin ini sumbu X adalah gerakan horizontal (kanan atau kiri), sumbu Y adalah gerakan menyilang (menuju atau menjauh dari kolom), dan sumbu Z gerakan vertikal dari atas maupun ke bawah. Mesin CNC sangat bergantung pada penggunaan koordinat kartesius karena programmer dapat menemukan setiap titik pada suatu pekerjaan dengan tepat maupun akurat.

Ketika suatu titik poin berada pada bidang kerja, dua berpotongan lurus dengan garis, satu vertikal dan satu horisontal maka garis-garis ini harus berada di sudut kanan satu sama lain. Dan titik di mana akan menyeberang disebut titik asal, atau titik nol. Keterangan ini bisa di lihat pada Gambar 2.2 bawah ini:



Gambar 2.2 Zero Point [3]

2.1.2 Jenis-jenis Mesin CNC

Di industri menengah dan besar, akan banyak dijumpai penggunaan mesin CNC dalam mendukung proses produksi. Secara garis besar, mesin CNC dibagi dalam 2 macam yaitu mesin bubut dan mesin *milling*.

Mesin Bubut adalah suatu Mesin perkakas yang digunakan untuk memotong benda yang diputar. Bubut sendiri merupakan suatu proses pemakanan benda kerja yang sayatannya dilakukan dengan cara memutar benda kerja kemudian dikenakan pada pahat yang digerakkan secara translasi sejajar dengan sumbu putar dari benda kerja. Gerakan

putar dari benda kerja disebut gerak potong relatif dan gerakan translasi dari pahat disebut gerak umpan. [1]

Mesin frais adalah sejenis mesin perkakas untuk mengerjakan peralatan mesin dari logam dengan gerakan utama alat potongnya berputar.

2.1.3 Cara Kerja Mesin CNC

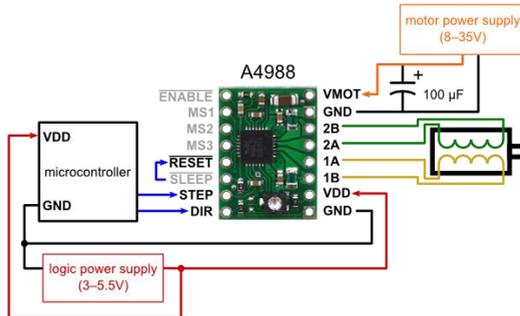
Prinsip kerja dari mesin CNC *milling* yaitu gerakan pemotongan terjadi saat alat potong berputar yang diikuti dengan gerakan pemakanan dan gerakan pengikat benda kerja. Gerakan berputar disebut juga dengan gerakan utama yaitu gerakan berputar alat potong sambil memotong benda kerja. Gerakan pemakanan adalah gerakan alat potong sepanjang daerah pemotongan. Gerakan pemakanan berbentuk lurus dan melingkar. Gerakan pengikatan adalah gerakan menekan benda kerja dan alat potong yang memungkinkan sisi potong dapat dengan mudah memotong benda kerja.

2.1.4 CNC Portable

CNC Portable merupakan sebuah mesin CNC yang nantinya akan mudah di bawah kemana-mana. Mesin CNC Portable dibuat dengan tujuan dapat digunakan oleh sekelompok wirausaha pengerajin kayu yang bisa membantu pekerjaan mereka serta dapat digunakan sebagai media pembelajaran untuk mahasiswa yang melakukan penelitian ataupun pengadaan seminar

2.2 Driver A4988

A4988 adalah driver motor *microstepping* yang dirancang untuk mengoperasikan motor stepper bipolar secara *mode full-step, half-step, quarter-step, eighth-step, and sixteenth-step*, dengan kapasitas *drive output* hingga 35 V dan ± 2 A. Karena dapat melakukan 1/16 step maka ketelitian tiap step makin bertambah. Driver ini memiliki internal sirkuit *protection* meliputi *thermal shutdown, undervoltage lockout* (UVLO) dan *crossover-current protection*. Sehingga menambah keamanan pada driver motor stepper [4]. A4988 Stepper Motor Driver Module memiliki kemampuan untuk beroperasi dalam *slow* atau *mixed decay mode* seperti yang ditunjukkan pada Gambar 2.3 bawah ini:



Gambar 2.3 Rangkaian Driver A4988 [4]

Bentuk asli IC A4988 dapat dilihat pada gambar. IC A4988 memiliki 8 buah kaki yang terdapat pada sisi kanan dan sisi kirinya. Pada Tabel 2.1 dijelaskan mengenai bagaimana pengaturan *microstep resolution* pada sebuah driver motor A4988.

Tabel 2.1 Nilai Logika pada Driver A4988

MS1	MS2	MS3	<i>Microstep Resolution</i>
<i>Low</i>	<i>Low</i>	<i>Low</i>	<i>Full Step</i>
<i>High</i>	<i>Low</i>	<i>Low</i>	<i>Half Step</i>
<i>Low</i>	<i>High</i>	<i>Low</i>	<i>Quarter Step</i>
<i>High</i>	<i>High</i>	<i>Low</i>	<i>Eighth Step</i>
<i>High</i>	<i>High</i>	<i>High</i>	<i>Sixteenth Step</i>

2.3 Arduino

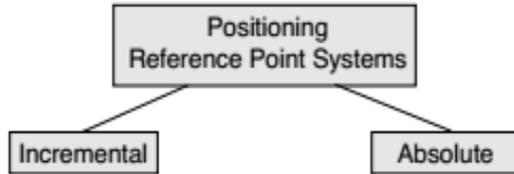
Arduino merupakan platform open source baik secara *hardware* dan *software*. Arduino terdiri dari mikrokontroler ATmega328 dengan menggunakan Kristal osilator 16 MHz. Catu daya yang dibutuhkan untuk mencatu sistem minimum arduino cukup dengan tegangan 5 VDC. Port Arduino seri ATmega terdiri dari 20 pin yang meliputi 14 pin I/O digital dengan 6 pin dapat berfungsi sebagai output PWM (*Pulse Width Modulation*) dan 6 pin sebagai I/O *analog*. Kelebihan Arduino adalah tidak membutuhkan flash programmer external karena di dalam chip mikrokontroler arduino telah diisi dengan *bootloader* yang membuat proses *upload* menjadi lebih sederhana. [5]

Untuk koneksi terhadap computer dapat menggunakan RS232 to TTL *converter* atau menggunakan chip USB ke serial converter. Arduino Uno dilengkapi dengan static random-access memory (SRAM) berukuran 2KB untuk menyimpan data, flash memory berukuran 32KB,

dan *Erasable Programmable Read-Only Memory* (EEPROM) untuk menyimpan program.

2.4 Sistem Program

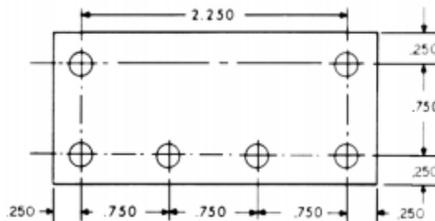
Pemrograman adalah suatu urutan perintah yang disusun secara rinci tiap blok per blok untuk memberikan masukan mesin perkakas CNC tentang apa yang harus dikerjakan. Ada dua jenis mode pemrograman, sistem incremental dan sistem absolut yang digunakan pada CNC [3]. Kedua sistem memiliki aplikasi dalam pemrograman CNC, dan tidak ada sistem yang benar atau salah tiap waktunya. Sebagian besar kontrol pada peralatan mesin ini adalah mampu menangani baik inkremental atau pemrograman absolut. Untuk menyusun pemrograman pada mesin CNC diperlukan seperti pada Gambar 2.4 bawah ini:



Gambar 2.4 Sistem Program pada Mesin CNC [3]

2.4.1 Sistem Inkremental

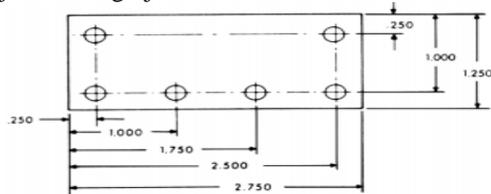
Pada sistem ini titik awal penempatan yang digunakan sebagai acuan adalah selalu berpindah sesuai dengan titik actual yang dinyatakan terakhir. Untuk mesin bubut maupun mesin frais diberlakukan cara yang sama. Setiap kali suatu gerakan pada proses pengerjaan benda kerja berakhir, maka titik akhir dari gerakan alat potong itu dianggap sebagai titik awal gerakan sesuai pada Gambar 2.5 bawah ini:



Gambar 2.5 Sistem Inkremental [3]

2.4.2 Sistem Absolute

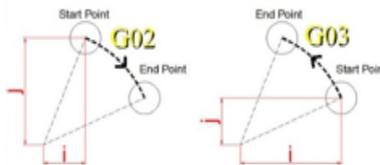
Pada sistem ini titik awal penempatan alat potong yang digunakan sebagai acuan adalah menetapkan titik referensi yang berlaku tetap selama proses operasi mesin berlangsung. Untuk mesin bubut, titik referensinya diletakkan pada sumbu (pusat) benda kerja yang akan dikerjakan pada bagian ujung. Sedangkan pada mesin frais, titik referensinya diletakkan pada pertemuan antara dua sisi pada benda kerja yang akan dikerjakan. Pengerjaan absolute sesuai Gambar 2.6 bawah ini:



Gambar 2.6 Sistem Absolute [3]

2.4.3 Interpolasi

Metode interpolasi adalah metode yang berpindah dari satu titik ke titik berikutnya yang disebut interpolasi. Ada lima metode interpolasi: linier, melingkar, heliks, parabola, dan kubik. Semua kontrol contouring menyediakan interpolasi linier, dan sebagian besar kontrol mampu melakukan gerakan linier dan interpolasi melingkar. Heliks, parabola, dan interpolasi kubik yang digunakan oleh sebuah industri yang memproduksi bagian-bagian yang memiliki bentuk kompleks, seperti pengerjaan badan mobil. [6] Pengerjaan interpolasi sesuai Gambar 2.7 bawah ini:



Gambar 2.7 Gerak Interpolasi Melingkar [6]

2.4.4 Continous Path (Contouring)

Contouring, atau pergerakan mesin yang terus menerus, melibatkan kerja seperti yang diproduksi menggunakan mesin bubut atau penggilingan mesin, di mana alat pemotong akan berhubungan langsung dengan benda kerja karena pergerakan dari program untuk

menunju ke arah depan. Positioning adalah kemampuan untuk melakukan gerakan kontrol pada dua atau lebih mesin terhadap sumbu tertentu secara bersamaan untuk menjaga kecepatan *cutter* agar tetap konstan.

2.4.5 Point to Point

Point-to-point positioning digunakan apabila diperlukan untuk pengerjaan yang memerlukan akurasi yang akurat pada pengerjaan motor spindle. Point-to-point positioning adalah proses positioning dari satu koordinat (XY) posisi atau pada satu titik ke titik lainnya, melakukan operasi permesinan, dan pola ini akan dikerjakan sampai selesai.

2.5 G-Code

G-Code adalah bahasa yang digunakan untuk mengontrol sebuah mesin CNC . *G-code* adalah salah satu jenis pemrograman CNC yang digunakan oleh para programmer CNC, jenis lainnya adalah program COM. *G-code* biasanya disebut *cycle codes* karena fungsinya yang mengacu pada pergerakan sumbu X,Y, dan Z dari mesin CNC. *G-Code* dikelompokkan kedalam beberapa kelompok seperti kelompok 01 yang mencakup kode G00,G01,G02,G03 yang mempengaruhi pergerakan dari bidang cetak dan nozzle. Kelompok 03 yang mencakup absolute programing atau incremental programming. Kode G00 memposisikan nozzle di atas bidang cetak dari satu titik ke titik yang lain dengan cepat. Selama proses pergerakan yang cepat, baik sumbu X atau Y bisa digerakkan secara terpisah atau bersamaan. Tingkat pergerakan cepat bervariasi dalam setiap mesin, dengan rentang dari 200 sampai 800 in/min atau 5 sampai 20m/min. [2]

2.6 Software Grbl

Grbl adalah sebuah *software open source* gratis yang digunakan untuk mengontrol sebuah mesin CNC yang dapat di unggah ke *library* Arduino. Pada dasarnya Grbl adalah sebuah hex file yang dapat di unggah ke Arduino agar Arduino dapat membaca perintah dalam g-code [7]. Untuk mengirimkan g-code ke arduino digunakan grbl controller grbl *controller* adalah sebuah *software* yang digunakan untuk mengirimkan g-code ke sebuah mesin CNC, seperti 3D printer. *Software* ini memudahkan pengguna dalam proses pemrograman sebuah mesin

CNC.Pengguna bisa memberikan perintah secara langsung atau pengguna juga bisa mengunggah satu file dalam bentuk *notepad* yang berisi kode-kode *G-code* yang sebelumnya telah dibuat.

2.7 Pemrograman Web

Pemrograman adalah kumpulan instruksi atau perintah tertulis yang dibuat oleh manusia secara logis untuk memerintahkan komputer agar melakukan langkah atau proses tertentu dalam menyelesaikan suatu masalah. Pemrograman biasanya menghasilkan sebuah perangkat lunak baru yang dapat dijalankan dengan mudah oleh orang lain tanpa harus mengetahui tahapan-tahapan detail dalam melakukan tugas tersebut. Sedangkan kata web, dapat diartikan sebagai halaman atau media informasi yang dapat diakses dengan perangkat lunak browser melalui jaringan komputer atau internet. Pemrograman web adalah proses membuat aplikasi komputer yang dapat digunakan/ditampilkan dengan bantuan browser. Tiga program dasar web dapat dilihat pada Gambar 2.8 bawah ini:



Gambar 2.8 Tiga Bagian Terpenting dalam Pemrograman Web [8]

2.7.1 Javascript

JavaScript adalah bahasa pemrograman web yang bersifat Client-Side *Programming Language*. Client-Side *Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh client. Aplikasi Client yang dimaksud merujuk kepada web browser seperti Google Chrome dan Mozilla Firefox. Bahasa pemrograman Client-Side berbeda dengan bahasa pemrograman Server-Side seperti PHP, dimana untuk Server Side seluruh kode program dijalankan di sisi Server. Untuk menjalankan JavaScript, kita hanya membutuhkan aplikasi text editor dan web browser. JavaScript memiliki fitur: *high-level programming language*, Client-Side, loosely typed dan berorientasi objek. JavaScript pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat

tanpa harus menunggu pemrosesan di web server. Sebelum JavaScript, setiap interaksi dari user harus diproses oleh web server. [9]

Dalam perkembangannya, JavaScript mengalami permasalahan yang sama seperti kode pemrograman web yang bersifat client side seperti CSS, yakni bergantung kepada implementasi web browser. Kode JavaScript yang kita buat, bisa saja tidak bekerja di Internet Explorer, karena web browser tersebut tidak mendukungnya. Sehingga programmer harus bekerja extra untuk membuat kode program agar bisa “mengakali” dukungan dari web browser. Karena hal tersebut, JavaScript pada awalnya termasuk bahasa pemrograman yang rumit, karena harus membuat beberapa kode program untuk berbagai web browser. Namun, beberapa tahun belakangan ini, JavaScript kembali bersinar berkat kemudahan yang ditawarkan oleh komunitas programmer yang membuat *library* JavaScript seperti jQuery. *Library* ini memudahkan kita membuat program JavaScript untuk semua web browser, dan membuat fitur-fitur canggih yang sebelumnya membutuhkan ribuan baris kode program menjadi sederhana. Contoh hasil program JavaScript bisa dilihat pada Gambar 2.9 bawah ini:

The image shows a registration form on a green background. It contains a text input field for 'Username' with a red border and a red error message 'Username is required' below it. Below the error message is a text input field containing 'newuser@gmail.com'. Underneath that are two password input fields, each with a red border and a red error message 'Password is required'. At the bottom of the form is a blue button labeled 'Register'.

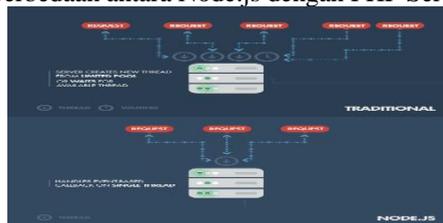
Gambar 2.9 Salah Satu Fungsi JavaScript [9]

Salah satu fungsi JavaScript yaitu dalam pemrosesan form adalah untuk menambah aspek kenyamanan (*user friendly*) dan membuat form lebih interaktif. Dengan JavaScript kita bisa memeriksa isian form apakah sudah sesuai atau belum. Jika ada yang kurang, tampilkan pesan kesalahan agar user bisa memperbaiki bagian yang salah. Tanpa JavaScript, proses validasi ini dilakukan di Server menggunakan PHP. Jika ditemukan kesalahan, Server akan mengirim kembali form ke web browser untuk diperbaiki. Setelah diputuskan, form disubmit kembali, jika masih ditemukan kesalahan, form akan dikembalikan lagi, demikian seterusnya hingga form dinyatakan valid dan bisa diproses lebih lanjut oleh PHP. [10]

Untuk Server yang sibuk dan jaringan yang cukup lambat, proses pengiriman dari web browser ke web Server dan sebaliknya bisa memakan waktu yang lama. Bayangkan jika anda mengisi sebuah form register, kemudian men-klik tombol submit, menunggu sekitar 2 menit, ternyata tanggal lahir belum diisi. Setelah diisi, klik lagi tombol submit dan menunggu lagi selama 2 menit untuk mengetahui bahwa password harus diisi minimal 6 karakter. Menggunakan JavaScript, validasi seperti ini bisa dilakukan di web browser sebelum dikirim ke web Server.

2.8 Node Server

Node.js adalah sebuah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis WEB. Node.js dieksekusi sebagai aplikasi Server. Platform ini menggunakan bahasa pemrograman javascript dan menggunakan teknik non-blokir untuk mempercepat proses. Teknik non-blokir adalah sebuah metode penyelesaian sebuah fungsi. atau pada dasarnya node.js adalah sebuah platform yang menggunakan bahasa JavaScript untuk dieksekusi sebagai Server. Untuk kelebihan Node js, kita bisa menggunakan JavaScript yang dieksekusi sebagai Server-Side, kelebihan lain adalah teknik non-blokir yang ada di Node.js, teknik non-blokir yaitu teknik dimana Node.js akan melakukan eksekusi secara independen. Pada Gambar 2.10 memperlihatkan perbedaan antara Node.js dengan PHP Server.



Gambar 2.10 Perbedaan Server Tradisional dengan Node Js [11]

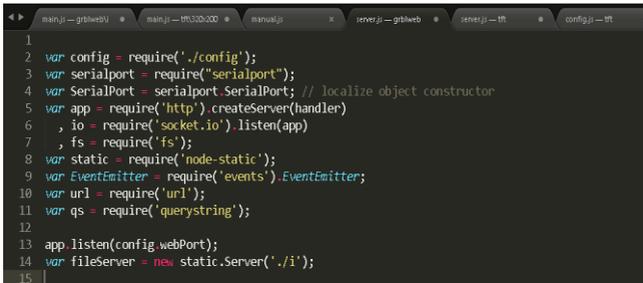
Node.js akan mengeksekusi sebuah operasi tanpa harus menunggu operasi sebelumnya selesai dieksekusi, sehingga menghasilkan aplikasi web yang lebih cepat dan efisien. Node.js dengan teknik non-blokir dapat diilustrasikan sebagai sebuah restoran, saat kita datang ke sebuah restoran, pelayan akan mencatat pesanan kita lalu menyerahkan pesanan ke petugas memasak, setelah itu pelayan akan mencatat pesanan orang lain lalu menyerahkan pesanan tersebut ke petugas memasak tanpa harus

menunggu pesanan kita selesai dikerjakan dan begitu seterusnya hingga semua pesanan diantar ke pelanggan. [11]

Dibanding dengan teknologi wev server tradisional dimana setiap request dibuat masing-masing satu thread, dan jumlah koneksi terbatas dengan jumlah RAM yang dimiliki server tersebut karena setiap thread mengambil space RAM. Node.js beroperasi dengan satu thread saja dan menggunakan asikronus dan non-blocking I/O memungkinkan jumlah konkurensi yang tinggi. [12]

2.9 Sublime Text 3

Sublime Text Editor adalah editor teks untuk berbagai bahasa pemrograman termasuk pemrograman PHP. Sublime Text Editor merupakan editor text lintas-platform dengan Python application programming interface (API). Sublime Text Editor juga mendukung banyak bahasa pemrograman dan bahasa markup, dan fungsinya dapat ditambah dengan plugin, dan Sublime Text Editor tanpa lisensi perangkat lunak. Tampilan software sublime text bisa lihat Gambar 2.11 di bawah ini:



```
1
2 var config = require('./config');
3 var serialport = require("serialport");
4 var SerialPort = serialport.SerialPort; // localize object constructor
5 var app = require('http').createServer(handler)
6   , io = require('socket.io').listen(app)
7   , fs = require('fs');
8 var static = require('node-static');
9 var EventEmitter = require('events').EventEmitter;
10 var url = require('url');
11 var qs = require('querystring');
12
13 app.listen(config.webPort);
14 var fileServer = new static.Server('./i');
15
```

Gambar 2.11 Tampilan Sublime Text

Sublime Text mendukung operation system seperti Linux, Mac Os X, dan juga windows. Sangat Banyak fitur yang tersedia pada Sublime Text Editor diantaranya minimap, membuka script secara side by side, bracket highlight sehingga tidak bingung mencari pasangannya, kode snippets, drag and drop direktori ke sidebar terasa mirip dengan TextMate untuk Mac OS.

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

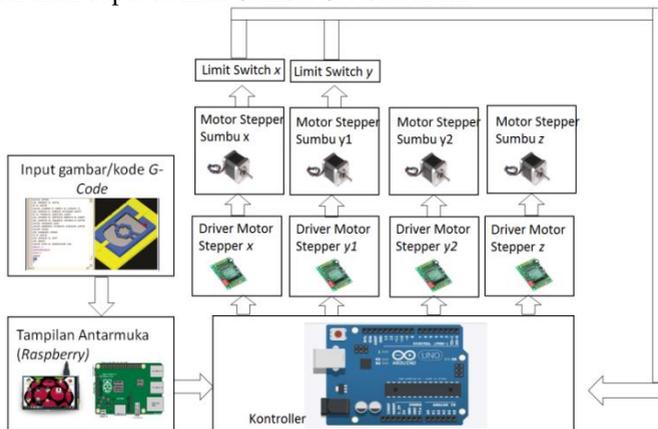
Pada bab perencanaan dan pembuatan sistem akan dibahas mengenai mekanisme pembuatan CNC Portable secara hardware maupun software, diantaranya meliputi, konfigurasi sistem CNC Portable secara keseluruhan, Konfigurasi sistem mekanik CNC Portable terhadap sumbu Y, Pembuatan Box Kontroller, Konfigurasi Elektrik pada CNC Portable, Perancangan software grbl pada CNC Portable, dan Perancangan antarmuka berbasis WEB.

3.1 Konfigurasi Keseluruhan Sistem CNC Portable

Keseluruhan dari sistem mesin CNC Portable ini terdiri dari dua sub bab yang akan dibahas yaitu perancangan blok sistem kerja keseluruhan serta perancangan diagram alir.

3.1.1 Perancangan Blok Sistem Kerja

Dalam merealisasikan mesin CNC Portable diperlukan sebuah konsep atau sistem konfigurasi yang benar agar mesin CNC Portable berjalan dengan semestinya, yaitu dapat menggerakkan empat motor stepper dengan keadaan yang sinkron dan melakukan proses pengeboran seperti halnya yang dilakukan mesin CNC *milling*. Skema kerja alat keseluruhan dapat di lihat Gambar 3.1 bawah ini:

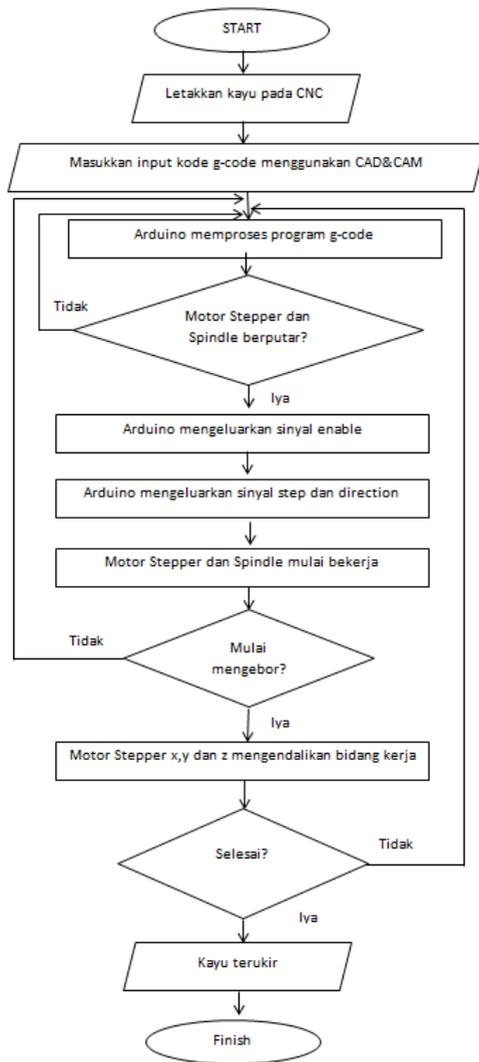


Gambar 3.1 Perancangan Keseluruhan Sistem Kerja CNC Portable

Dari Gambar 3.1 dijelaskan, langkah awal memulai proses operasi mesin dilakukan dengan memberikan masukan kode *G-Code* secara manual maupun secara gambar CAD dan CAM, dimasukkan pada tampilan antarmuka, yang terdapat pada *operating system* raspberry, dimana pengoperasian sistem dilakukan dengan tampilan *touchscreen* LCD pada Raspberry pi. Selanjutnya masukan dari *G-Code* akan diproses ke dalam program Grbl pada arduino, proses ini berfungsi untuk mengendalikan setiap motor stepper melalui driver motor A4988. Limit switch dipasang sebagai penanda batas akhir pada sumbu *x* dan *y* agar menonaktifkan otomatisasi pada mesin CNC Portable.

3.1.2 Perancangan Proses Pengerjaan

Proses pengerjaan dimulai dari Objek benda yang ingin diukir diletakkan pada mesin CNC, Pada antarmuka masukkan kode *G-Code* atau bisa menggunakan gambar yang sudah didesain terlebih dahulu dengan *software* CAD dan CAM, Arduino akan memproses kode *G-Code* yang dimasukkan, Motor Stepper dan motor spindle akan berputar, jika tidak berputar maka arduino akan memproses kembali kode *G-Code* yang dimasukkan. Jika motor stepper dan motor spindle berputar maka arduino akan mengeluarkan sinyal *enable*, Arduino mengeluarkan sinyal *step* dan *direction* untuk mengontrol motor stepper yang menggerakkan motor spindle dan bidang cetak pada setiap sumbu, Motor stepper dan motor spindle mulai bekerja. Jika tidak melakukan proses *milling* maka arduino akan kembali memproses kode *G-Code* yang dimasukkan. Jika proses *milling* dilakukan ketiga motor stepper sumbu X, Y, dan Z akan mengendalikan bidang kerja, proses *milling* akan berjalan sesuai dengan input yang diberikan, jika belum berjalan sesuai dengan input yang diberikan maka kembali arduino memproses kode *G-Code* yang dimasukkan, Setelah proses *milling* berjalan sesuai dengan input yang diberikan maka akan terlihat hasil dari ukiran kayu tersebut. Proses pengerjaan mesin CNC keseluruhan dapat dilihat pada Gambar 3.2



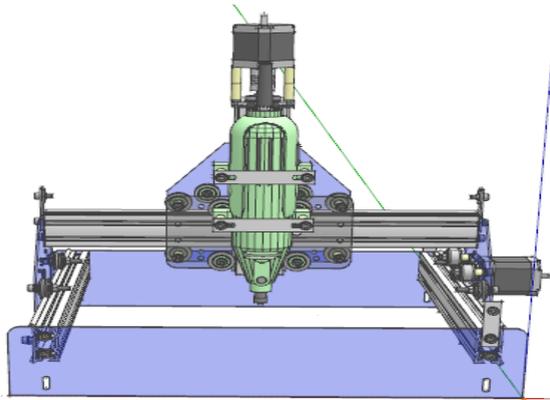
Gambar 3.2 Proses Kerja dari Keseluruhan Sistem

3.2 Perancangan Sistem Mekanik CNC Portable terhadap Sumbu Y

Untuk konfigurasi sistem mekanik dari CNC Portable ini tentu diperlukan konsep mekanik yang baik, agar CNC Portable dapat bekerja dengan semestinya, tidak mudah bergetar saat melakukan proses pengeboran (*milling processes*). Pada buku ini akan dibahas perancangan mekanik terhadap sumbu Y, dimana ini berperan penting pada bidang kerja. Karena menggunakan dua motor stepper untuk menggerakannya, dan pasti gerakan dari dua motor ini harus sinkron.

3.2.1 Perancangan Mekanik Sumbu Y

Agar mesin CNC dapat bekerja dengan baik maka perancangannya dibuat dengan menggunakan *software* sketchup seperti pada Gambar 3.3. Mesin CNC Portable ini memiliki 3 bagian utama yaitu sistem X, Y, dan Z.



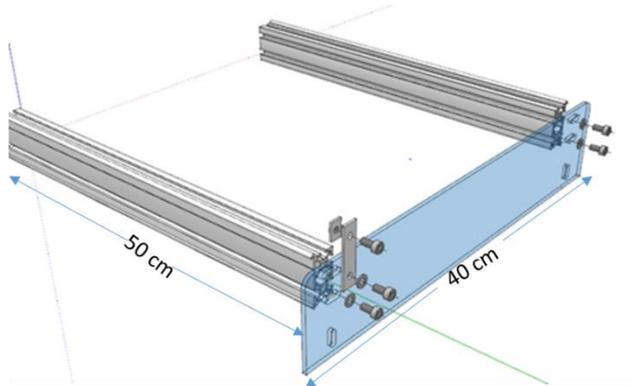
Gambar 3.3 Perancangan Mesin CNC Portable

Dari gambar di atas merupakan gambar design dari mesin CNC Portable 3 axis menggunakan empat motor stepper, dengan menggunakan media belt. Belt sendiri mempunyai fungsi untuk memindahkan tenaga melalui kontak antara belt dengan pulley penggerak dan pulley yang digerakkan. Belt digerakkan oleh gaya gesek penggerak, kemampuan belt untuk memindahkan tenaga tergantung pada kriteria berikut ini.

- a. Tegangan belt terhadap pulley
- b. Gesekan antara belt dan pulley

- c. Sudut kontak antara belt dan pulley
- d. Kecepatan belt

Maka dari itu pada pemasangan mekanik sumbu Y ini harus begitu cermat dikarenakan kalau tidak ada keserasian dalam pemasangan belt, maka gerakan pada sumbu Y akan tidak seimbang dan berjalan kasar karena tegangan belt yang tidak sama. Skema desain sumbu Y dapat di lihat Gambar 3.4 bawah ini:



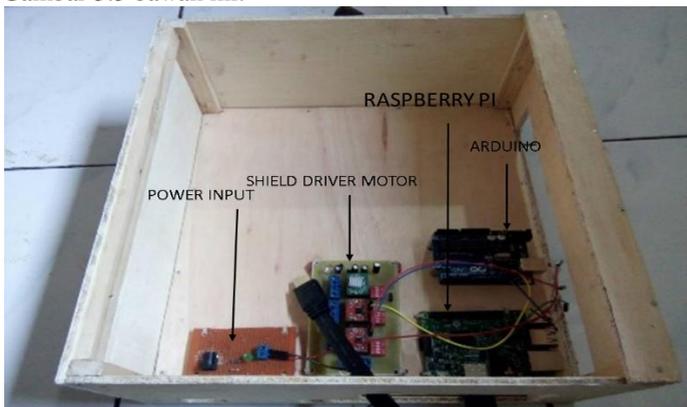
Gambar 3.4 Perancangan Desain Sumbu Y

Sistem Y seperti pada Gambar 3.4 merupakan bagian yang bergerak berdasarkan sumbu Y yaitu bergerak ke arah vertikal. Sistem Y ini terdapat pada bagian penampang mesin yang merupakan tempat peletakan bidang kerja yang dilubangi maupun dibentuk dengan kode g-code yang dilah secara konvensional maupun secara CAD/CAM.

3.3 Pembuatan *Box Controller*

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 A4988, arduino, Raspberry, cooling fan serta Touchscreen LCD. Pembuatan Box Controller sendiri hanya membutuhkan gergaji kayu untuk membentuk setiap potongan kayu menjadi box yang ringan dan mudah untuk di

bawah. Untuk detail tampilan dari *box controller* sendiri dapat dilihat pada Gambar 3.5 bawah ini:



Gambar 3.5 Tampak Dalam *Box Controller*

Gambar di atas merupakan bagian dalam dari box controller dengan beberapa rangkaian dan controller, ada beberapa lubang sebagai media keluarnya kabel-kabel dari rangkaian seperti kabel motor stepper, kabel koneksi antar komponen controller dengan rangkaian driver motor dan sebagainya, tampilan luar box seperti Gambar 3.6 bawah ini:



Gambar 3.6 Tampak Luar *Box Controller*

Gambar di atas merupakan tampak luar box controller terlihat tampilan lcd touchscreen ukuran 5 inchi, serta *coolant fan* yang dipasang

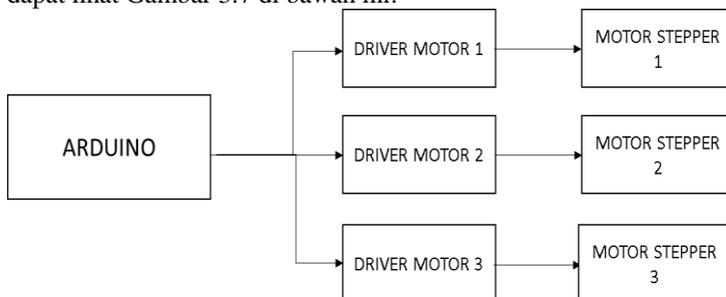
pojok kanan atas, dimana mempunyai fungsi untuk pendingin mesin bagian dalam.

3.4 Konfigurasi Elektrik pada CNC Portable

Pada subbab ini akan membahas mengenai konfigurasi elektriknya, antara lain Pembuatan rangkaian shield driver a4988, instalasi Arduino pada raspberry pi, koneksi driver dengan arduino, dan pembuatan rangkaian shield arduino uno. Penjelasan ini akan mengupas skematik elektrik yang tersusun dalam box controller. Bagaimana pemasangan driver pada arduino dan lain sebagainya.

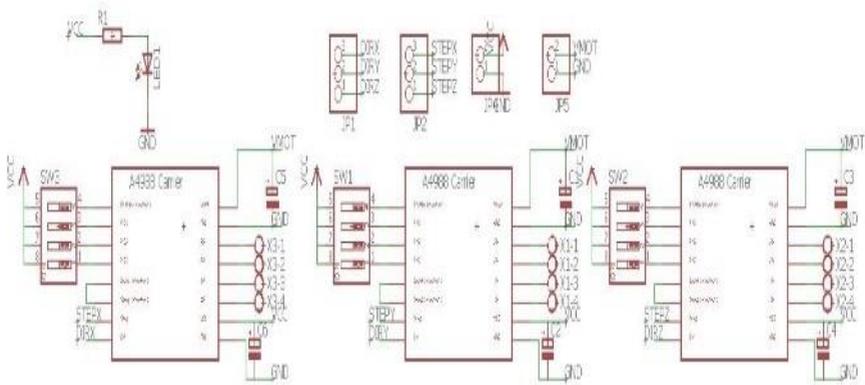
3.4.1 Pembuatan Rangkaian *Shield Driver A4988*

Telah dijelaskan pada bab dua mengenai driver A4988, pada kali ini untuk dapat menggerakkan motor stepper dibutuhkan IC A4988. Satu buah IC A4988 hanya dapat mengendalikan satu buah motor stepper. Oleh karena itu pada driver motor ini terdapat 4 buah IC A4988 untuk dapat menggerakkan 4 buah motor stepper. Untuk skema blok diagram dapat lihat Gambar 3.7 di bawah ini:



Gambar 3.7 Blok Diagram Arduino dengan Motor Stepper

Pada gambar di atas merupakan skema blok diagram mengenai rangkaian motor yang akan dibuat, dimana proses pertama dari arduino memberikan suatu perintah kepada driver untuk mengendalikan motor stepper. Pada gambar di atas terlihat tanpa adanya sebuah *feedback*. Dan membuat sistem ini berjalan dengan keadaan *open-loop*. Pada tahap selanjutnya yaitu pembuatan skematik rangkaian menggunakan software eagle. Dan untuk skematik rangkaian dapat di lihat pada Gambar 3.8 :



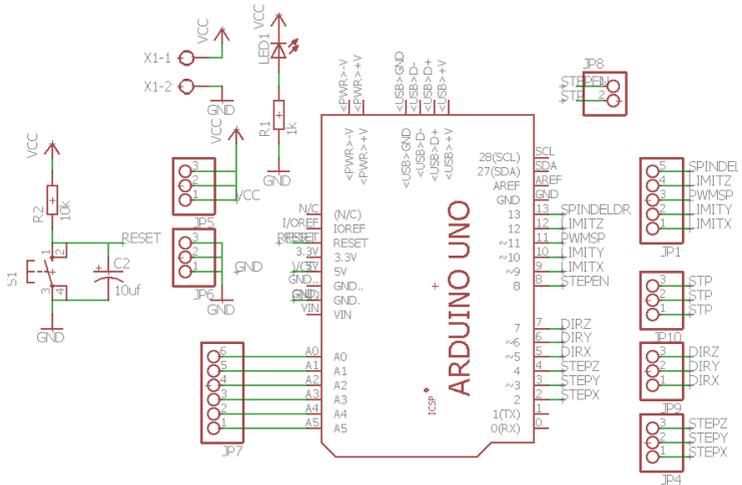
Gambar 3.8 Skematik Rangkaian Shield Driver A4988

Dari skematik rangkaian di atas yaitu, 3 buah driver motor yang gabung menjadi satu rangkaian shield, untuk meringkan 3 driver menjadi satu bagian. Setiap driver akan dihubungkan pada dipswitch untuk mengatur microstepping pada motor stepper menjadi mode full step, half step maupun mode lain sebagainya yang sudah disediakan pada driver tersebut. Dan pada skematik rangkaian dua sumber dc yang mempunyai ukuran tegangan yang berbeda dimana tegangan pertama sebagai sumber tegangan driver yang hanya berkapasitas lima volt, dan tegangan satunya mempunyai besaran 12 volt sebagai tegangan motor stepper, tetapi untuk groundingnya dijadikan satu. Serta setiap driver akan memiliki empat terminal block sebagai tempat kabel motor stepper disambungkan. Hail perancangan *shield* driver bisa lihat lampiran D3.

3.4.2 Pembuatan Rangkaian *Shield* Arduino

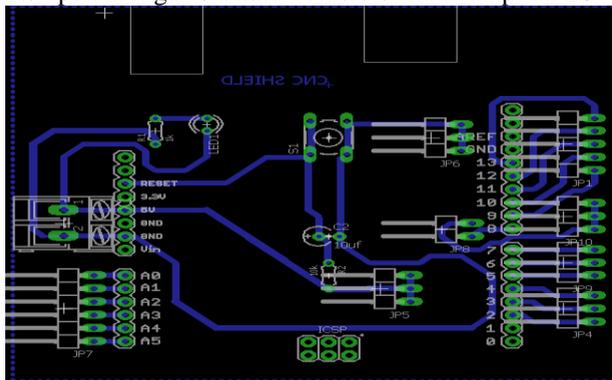
Pembuatan rangkaian shield ini bertujuan untuk mempermudah memasang kabel koneksi dari kabel driver ke arduino sesuai aturan dari konfigurasi grbl, jadi tinggal memasangkan sesuai pin yang tertera pada rangkaian shield grbl, pembuatan shield dibuat dengan seefien mungkin untuk menimalkan bentuk board dari arduinonya. Kalau semisal board

yang dibuat terlalu besar maka terlihat tidak seimbang dengan board arduinonya, rangkaian shield grbl bisa dilihat Gambar 3.9 bawah ini:



Gambar 3.9 Skematik Rangkaian *Shield* Arduino

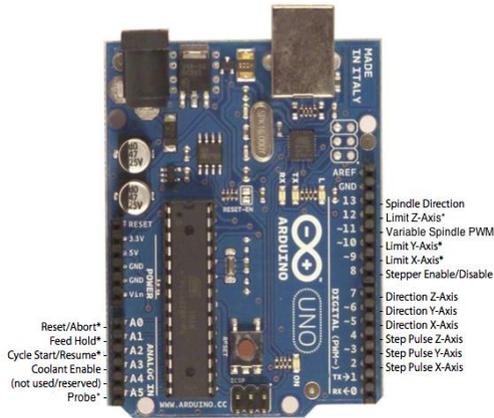
Gambar di atas merupakan skematik rangkaiannya, setelah itu dibuat rangkaian dalam bentuk board untuk mudah dalam proses penyablonannya, dari hasil routernya bisa lihat Gambar 3.10 untuk lebih jelasnya. Hasil perancangan *shield* arduino bisa lihat lampiran D3



Gambar 3.10 Hasil *Routing* dari Skematik *Shield* Arduino

3.4.3 Koneksi Driver pada Arduino

Konfigurasi pin mempunyai peran untuk mengetahui pemasangan driver terhadap pin pada arduino. Awal mula harus melihat pin konfigurasi grbl pada arduino. Dan langkah pertama, untuk menghubungkan motor stepper ke Grbl, harus memerlukan beberapa driver motor stepper untuk menyalakan steppers dan menghubungkan masukan driver A4988 ke pin kontrol Arduino. Ada sejumlah driver yang bisa melakukan ini, tersedia pre-built, partially pre-built, atau completely DIY. Ada beberapa contoh di bagian bawah halaman. Driver stepper perlu untuk dibagi beberapa pin pengaktif stepper (D8) ke pin masing-masing, sedangkan pin pin direction dan step (D2-D7) perlu dihubungkan ke pin masing-masing pada driver. Memastikan semua driver dan Arduino berbagi prinsip yang sama (kalau bisa sesuai dengan spesifikasi motor stepper yang digunakan). Konfigurasi dari pin arduino untuk grbl bisa dilihat gambar 3.11:



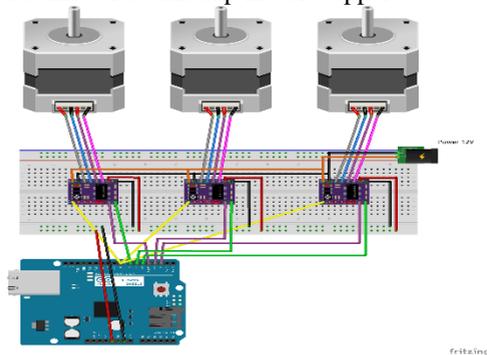
* - Indicates input pins. Held high with internal pull-up resistors.

Gambar 3.11 Konfigurasi Pin Grbl pada Arduino Uno

Setelah pemasangan driver sesuai dengan konfigurasi yang di atas, bisa melakukan persiapan untuk mengaktifkan grbl. Ini bertujuan untuk mengetahui pemasangan driver pada arduino sudah terpasang dengan baik maupun belum. Untuk pengecekan bisa mencoba secara langsung terhadap tampilan antarmuka dari grbl. Dari gambar di atas juga arduino bisa difungsikan sebagai fungsi limit dimana entar sebagai pembatas saat mesin cnc bekerja, dengan cara menghubungkan saklar batas

normal-terbuka ke masing-masing pin batas (D9-D11). Homing dan hard limits menggunakan switch yang sama. Pin batas ini sudah terpasang dengan konsep resistor pull-up internal, jadi yang harus dilakukan adalah memasangnya ke ground. Jadi saat ini dengan menutup saklar, saklar akan menarik batas pin ke ground. Pada perancangan ini menggunakan saklar hard limit pada kedua ujung perjalanan sumbu, maka hanya dikaitkan dua saklar batas sejajar dengan pin dan batas sumbu. Pastikan pemasangan sakelar sebelum mencoba harus melakukan aktivasi siklus homing. Yang dimaksud dengan siklus homing adalah sebuah siklus untuk menentukan posisi 0 dari bidang kerja mesin CNC.

Jika memiliki keinginan atau kebutuhan untuk mengontrol spindle atau coolant, Grbl sudah menyediakan pin output ini (D12, D13, A3) tinggi atau rendah, tergantung dari perintah kode G yang dikirimkan ke Grbl. Dengan v0.9 dan variable spindle PWM diaktifkan, pin D11 akan menampilkan kisaran tegangan dari 0V sampai 5V tergantung pada perintah spindle speed G-code. 0V menunjukkan spindle off dalam kasus ini. Pada Gambar 3.12 di bawah ini merupakan konfigurasi driver a4988 terhadap arduino serta terhadap motor stepper.



Gambar 3.12 Koneksi Driver pada Arduino untuk Grbl

Karena pada pin-pin ini tergantung pada bagaimana cara menggunakannya, maka pada pembuatan tugas akhir ini kami hanya mengaktifkan beberapa pin yang digunakan antara lain pin untuk tiga sumbu yaitu sumbu x, y, dan z. Serta pengaktifkan fungsi homing dan fungsi hard limit sebagai pembatas bidang kerja mesin cnc. Penyedia software ini juga memperbolehkan untuk dapat meng-hack file sumber kontrol spindle dan coolant dengan mudah mengubah cara kerjanya dan

kemudian mengkompilasi dan mengunggah Grbl yang dimodifikasi oleh pengguna melalui Arduino IDE. Karena pada dasarnya software grbl ini merupakan software yang open source sama halnya dengan arduino.

3.4.4 Instalasi Arduino IDL pada Raspberry Pi

Proses instalasi program arduino pada raspberry pi mempunyai cara tersendiri, karena sistem OS pada komputer mini menggunakan sistem linux raspbian maka proses instalasi menggunakan terminal untuk mendownload arduino idle dan juga harus terhubung dengan internet, langkah-langkah instalasinya sebagai berikut. Sebelum melakukan instalasi Arduino IDE, perlu melakukan update. Boot PI dan ketik atau salin kode berikut ke terminal LX: dengan mengetikkan Sudo apt-get install arduino. Selanjutnya melakukan instalasi arduino IDE menggunakan perintah, Sudo apt-get install arduino. Menunggu beberapa waktu untuk menyelesaikan proses instalasi ini, setelah selesai maka cek pada bilah menu, pada bagian Elektronika, untuk pengecekan dengan meng-upload program blink led pada arduino. Apakah program arduino sudah berjalan dengan benar atau masih ada kekurangan yang lain.

3.5 Perancangan Software Grbl

Pada bab dua sudah dijelaskan mengenai penjelasan grbl, dan bab kali ini akan menjelaskan secara detail perancangannya dan penggunaan grbl sebagai program dasar yang menjalankan mesin CNC Portable. Pada subbab ini akan membahas beberapa bagian antara lain: Proses kompilasi grbl, konfigurasi grbl, melakukan flashsing grbl, penggunaan grbl, dan interface grbl yang terbagi lagi menjadi subbab bagian.

3.5.1 Proses Kompilasi Grbl

Proses kompilasi atau biasa disebut dengan proses instalasi pada media lain, Arduino memiliki semua yang dibutuhkan untuk mengkompilasi grbl yang disertakan dalam paket perangkat lunak. Metode ini mengkompilasi kode sumber Grbl dan secara otomatis menguploadnya ke Arduino. Tidak bisa langsung flash file .hex pra-dikompilasi melalui antarmuka IDE.

1. Download source code grbl (v0.9j).
 - a. Klik tombol Download ZIP di halaman rumah Grbl.
 - b. Unzip download dan Anda akan memiliki folder bernama grbl-master.
2. Meluncurkan Arduino IDE
 - a. Pastikan Anda menggunakan versi terbaru dari Arduino IDE!

3. Memuat Grbl ke Arduino IDE Perpustakaan
 - a. Klik menu drop-down Sketch, arahkan ke Sertakan Perpustakaan dan pilih Add ZIP Perpustakaan.
 - b. Jika tidak sengaja memilih file .zip atau folder yang salah, maka diperlukan untuk menghapus file library dari Arduinonya, menghapus kesalahan, dan re-do Langkah 3.
4. Buka contoh grbl Upload Arduino
 - a. Klik File menu, kemudian arahkan ke Examples-> Grbl, dan pilih GrblUpload.
5. Mengkompilasi dan meng-upload Grbl untuk Arduino.
 - e. Menghubungkan Arduino Uno ke komputer.
 - f. Pastikan board sudah diatur ke Arduino Uno di Tool-> menu dan port serial yang dipilih benar di Tool-> Serial Port.
 - g. Klik Upload, dan Grbl harus mengkompilasi dan flash ke Arduino yang dituju (Flashing dengan programmer juga bekerja dengan menggunakan Upload Menggunakan perintah menu Programmer.)

3.5.2 Konfigurasi Grbl

Ketik \$ dan menekan tombol enter agar Grbl menampilkan pesan bantuan. Selanjutnya lihat pada tampilan antarmuka, akan muncul konfigurasi-konfigurasi tertentu untuk mengatur grbl arduino, seperti terlihat pada Tabel 3.1 di bawah ini.

Tabel 3.1 Konfigurasi Grbl

KODE	FUNGSI
\$\$	Setelan Grbl
\$#	# parameter
~	Cycle start
!	Feed hold
?	Current status
Ctrl-x	Reset grbl

Perintah '\$' adalah perintah sistem Grbl yang digunakan untuk menampilkan pengaturan, melihat atau mengubah keadaan dan mode Grbl, dan memulai siklus homing. Empat perintah '\$' terakhir adalah perintah kontrol realtime yang dapat dikirim kapan saja, tidak peduli apa yang dilakukan Grbl. Ini segera mengubah perilaku Grbl yang sedang

berjalan atau segera mencetak laporan data realtime penting seperti posisi saat ini (alias DRO).

3.5.2.1 Pengaturan Grbl

Untuk melihat \$\$ pengaturan , ketik dan tekan enter setelah terhubung ke Grbl. Grbl harus merespon dengan daftar pengaturan sistem saat ini, seperti yang ditunjukkan pada contoh di bawah ini. Semua pengaturan ini terus berlanjut dan disimpan di EEPROM, jadi jika ingin mematikannya, ini akan dimuat kembali seperti saat menghidupkan Arduino dengan keadaan semula. Pada Tabel 3.2 bisa di lihat kode-kode dalam pengaturan Grbl.

Tabel 3.2 Kode dalam Pengaturan Grbl

Kode	Fungsi
\$0	Step Pulsa, mikrodetik
\$1	Step idle delay, mikrodetik
\$21	Hard Limit
\$100, \$101, dan \$102	Step /mm

The `$x=val` perintah menyimpan atau mengubah pengaturan Grbl, yang dapat dilakukan secara manual dengan mengirimkan perintah ini ketika terhubung ke Grbl melalui program terminal serial, tetapi kebanyakan Grbl GUI akan melakukan ini hanya sebagai fitur *user-friendly*. Jika semuanya berjalan baik, Grbl akan merespon dengan 'ok' dan setting ini disimpan di EEPROM dan akan dipertahankan seterusnya atau disaat ingin mengubahnya sesuai dengan keinginan pengguna. Pengguna juga dapat memeriksa apakah Grbl telah menerima dan menyimpan pengaturan ini dengan benar dengan mengetikkan \$\$ untuk melihat pengaturan sistem lagi.

3.5.2.2 Penjelasan Pengaturan Perbagian pada Grbl

a. \$0- Langkah Pulsa, mikrodetik

Driver stepper telah diberi sebuah nilai dengan panjang pulsa langkah minimum tertentu. Memeriksa bagian data atau mencoba beberapa nilai yang ingin dimasukkan ke tampilan antarmuka. Dan jika menginginkan pulsa terpendek yang dapat dikenali oleh driver stepper dengan baik, maka beri nilai kecil sesuai dengan yang di inginkan pengguna. Jika pulsa terlalu panjang, maka akan mengalami masalah saat menjalankan sistem dengan tingkat tertentu dan denyut step yang sangat tinggi, karena setiap langkah pulsa bisa mulai saling tumpang tindih. Pada grbl sendiri telah merekomendasikan sesuatu sekitar 10 mikrodetik, yang merupakan nilai default-nya.

b. \$1 - Langkah Idle Delay, msec

Setiap saat steppers menyelesaikan sebuah gerakan dan berhenti, Grbl akan menunda penonaktifan steppers dengan nilai tertentu. Atau, pengguna selalu dapat menjaga agar sumbu dapat diaktifkan (bertenaga sehingga dapat menahan posisi) dengan menetapkan nilai yang telah ditetapkan secara maksimal sekitar 255 milidetik. Dan apabila hanya untuk mengulanginya, pengguna dapat menjaga agar semua sumbu selalu diaktifkan dengan pengaturan \$1=255.

c. \$21 - Hard Limits

Kerja hard limit pada dasarnya sama dengan soft limit, tapi penggunaan hanya sebatas *physical switches* saja. Pada dasarnya pengguna perlu memasang beberapa switch (mekanis, magnetik, atau optik) di mana limit switch ini diletakan pada bagian akhir pergerakan setiap sumbu, atau di mana pun yang merasa mungkin ada masalah jika program pengguna bergerak terlalu jauh ke tempat yang seharusnya tidak diinginkan, maka perlu pemasangan limit switch. Saat saklar dipicu, akan segera menghentikan semua gerakan, mematikan pendingin dan spindle (jika terhubung), dan masuk ke mode alarm, yang memaksa pengguna untuk memeriksa mesin dan mengatur ulang semuanya.

Untuk menggunakan *hard limits* dengan Grbl, pin batas dipasang dengan menggabungkan resistor *pull-up* internal, jadi yang harus dilakukan hanyalah memasang kabel pada saklar yang biasanya terbuka dengan pin dan ground dan memungkinkan *hard limits* \$21=1. (\$21= 0 Nonaktifkan dengan). Jadi pengguna Grbl lainnya sangat menganjurkan untuk melakukan tindakan pencegahan gangguan listrik. Jika pengguna menginginkan batas kedua pada ujung pergerakan satu sumbu, cukup menggunakan dua saklar secara paralel dengan pin dan ground, jadi jika

salah satu dari mereka melakukan perjalanan, pergerakan yang menuju limit switches akan memicu batas dari mesin cncnya.

d. \$ 100, \$ 101 dan \$ 102 - [X, Y, Z] steps / mm

Grbl perlu tahu seberapa jauh setiap langkah akan membawa alat ini dalam kenyataan. Untuk menghitung langkah / mm untuk poros mesin cnc, maka perlu tahu:

- a. Harus melakukan pengamatan terhadap pergerakan revolusi motor stepper pengguna. Dan ini tergantung pada roda gigi sabuk pengaman atau pitch sekrup timbal yang terdapat pada motor stepper.
- b. Step penuh per revolusi steppers pada umumnya sekitar 200 step/mm.
- c. Sedangkan step mikro per langkah pada setiap driver (biasanya 1, 2, 4, 8, atau 16). Dan ada tip khusus untuk menentukan nilai microstep yang tinggi (misalnya, 16) dapat mengurangi torsi motor stepper pengguna, jadi gunakan yang terendah yang dapat memberi resolusi sumbu yang diinginkan dan properti yang mudah digunakan. Langkah-langkah / mm kemudian bisa dihitung seperti ini: $\text{steps_per_mm} = (\text{steps per revolusi} * \text{microsteps}) / \text{mm per rev}$.

3.6 Perancangan Sistem *Client* dan *Server Side* pada *Interface*

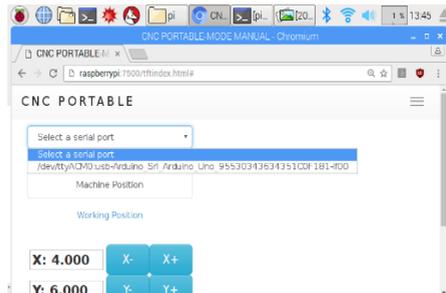
Antarmuka (*Interface*) merupakan mekanisme komunikasi antara pengguna (*user*) dengan sistem. Antarmuka (*Interface*) dapat menerima informasi dari pengguna (*user*) dan memberikan informasi kepada pengguna (*user*) untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi. Pada perancangan pertama yaitu pembuatan program mengenai client side dimana client adalah merupakan salah satu jenis bahasa pemrograman web yang proses pengolahannya dilakukan di sisi client. Proses penerjemahan atau pengolahannya dilakukan oleh web browser sebagai client-nya, didalam web browser sudah terdapat library yang mampu menerjemahkan semua perintah dihalaman web yang menggunakan client side scripting. Sedangkan *Server Side* sendiri adalah merupakan bahasa pemrograman web yang pengolahannya dilakukan dalam server, maksudnya ialah web server yang sudah telah terintegrasi oleh web engine

3.6.1 Perancangan Program Javascript sebagai *Client Side*

JavaScript merupakan bagian penting dari pemrograman web, karena javascript dapat berjalan di sisi Client. Misalnya, jam

ditampilkan pada halaman yang update sendiri untuk menunjukkan waktu saat ini pada komputer pengguna. Dan pada HTML yang di buat ini berfungsi untuk menekan button yang dibuat agar dapat menjalankan perintah dari tampilan antar muka dan dapat memberikan aksi pemilihan untuk kecepatan feed rate berupa rpm pada kecepatan motor spindle. Dan pada gambar di bawah merupakan tampilan antarmuka mesin CNC, dimana banyak program client side yang menggunakan JavaScript antara lain pemilihan serial port, program tombol reset, program tombol kendali setiap sumbu mesin CNC, serta tombol upload *G-Code*.

Perancangan pertama diawali dengan pembuatan program untuk pemilihan port, dimana ini mempunyai fungsi untuk memastikan serial port sudah terkoneksi dengan antarmuka atau belum. Pemilihan port akan terjadi apabila panjang data sama dengan 1, maka pemilihan port akan bernilai benar dimana akan berindikasi port sudah terkoneksi dengan sistem antarmuka. Sekilas tampilan pemilihan port bisa dilihat Gambar 3.13 di bawah ini:



Gambar 3.13 Tampilan Pemilihan Port

Dan untuk program mengenai pemilihan port bisa dilihat pada sebuah lampiran dengan halaman sekian, penulisan program JavaScript dilakukan menggunakan *software* sublime text editor. Sublime text editor sendiri adalah editor teks untuk berbagai bahasa pemrograman termasuk pemrograman javascript. Sublime Text Editor merupakan editor text lintas-platform dengan Python application programming interface (API). Sublime Text Editor juga mendukung banyak bahasa pemrograman dan bahasa markup, dan fungsinya dapat ditambah dengan plugin, dan Sublime Text Editor tanpa lisensi perangkat lunak. Perancangan selanjutnya mengenai program tombol pada tampilan sistem antarmuka, terdapat banyak tombol termasuk tombol reset,

tombol kendali, maupun tombol yang digunakan untuk upload gambar. Yang mana gambar tersebut akan dikerjakan oleh mesin cnc.

Tombol reset ini kan berfungsi untuk mereset tampilan antarmuka, apabila sistem alarm di aktifkan maka antarmuka tidak akan bisa dikendalikan. Untuk pengendaliannya harus dilakukan tombol reset dan menuliskan perintah \$X agar sistem kunci terbuka, isyarat ini seperti tertera pada subbab sebelumnya yaitu mengenai konfigurasi Grbl. Untuk perancangan tombol reset sebenarnya tidak jauh berbeda dengan kodingan sebelumnya, hanya fungsi change di ubah dengan fungsi click yang mengisyaratkan bahwa tombol reset dalam keadaan press (tertekan), dan logika akan bernilai satu dan dikirimkan ke Grbl agar dikendalikan sesuai prosedur. Tampilan tombol kendali bisa di lihat pada Gambar 3.14 di bawah ini:



Gambar 3.14 Tampilan Tombol Kendali

Perancangan tombol kendali ini sama halnya dengan kendali manual pada tampilan antarmuka manual, pada kendali antarmuka otomatis juga perancangannya sama, hanya memasukan kode *G-Code* pada perintah programnya. Untuk perintah programnya menggunakan kode G91, dimana kode ini ada perintah incremental. Jadi pemindahan dari sumbu ke sumbu itu naik satu tingkat, untuk pembacaannya dilakukan dua mode yaitu mode positif dan negatif. Mode ini mempunyai peran sebagai menjalankan fungsi direct pada motor stepper yaitu *clockwise* dan *counter clockwise*. Untuk pengerjaannya atau pengaktifannya menggunakan perintah case switch, perintah ini akan memanggil case berapa pada grbl. Berhubung untuk kode g code pada grbl yang yang membuat perintah tombol X berjalan untuk arah sebaliknya pada case 65, maka program javascriptnya juga dibuat case 65, dan diberi fungsi click. Seperti pada tombol reset yang artinya tombol ini akan bekerja apabila di klik oleh pengguna. Dan untuk

selanjutnya juga sama diberikan case berupa sesuai kode g code yang tertera pada software grbl. Fungsi tambahanya menggunakan preventDefault, yang artinya tombol kendali akan difungsikan secara default pada tampilan antarmuka.

Untuk perancangan upload file g code, menambahkan fungsi input pada program javascriptnya, input mempunyai fungsi untuk mengambile file pada komputer untuk mendapatkan file yang dituju. File yang dapat di upload harus berekstensi gcode maupun .nc karena juga menggunakan fungsi openGcodeFromText();. Pada Gambar 3.15 bawah ini terlihat tempat yang digunakan untuk meng-upload kode g-code:



Gambar 3.15 Tampilan Upload Kode G-Code

Proses upload kode g code bisa di awali dengan mengetikkan kode pada tempat yang disediakan, sedangkan kalau proses upload gambar bisa menekan tombol Upload GCODE dan akan tertuju pada library komputer dan pilih gcode yang di inginkan, maka pada kolom kotak tersebut akan di isi dengan hasil konversi gambar yang sudah di ubah menjadi kode g code. Dan langkah terakhir yaitu dengan menekan tombol Send to CNC (shift+enter), maka mesin cnc akan berjalan dengan semestinya sesuai gambar yang dikirimkan.

3.6.2 Perancangan Sistem Server pada Antarmuka

Node.js adalah framework untuk membuat scalable network applications yang menggunakan model event-driven. Framework ini memungkinkan bekerja dengan kehandalan tinggi dan dengan I/O blocking. I/O blocking ini bertujuan untuk menghindari proses waiting yang lama dan mengurangi beban CPU. Dengan menggunakan Node.JS, dapat membuat web server dengan mengandalkan modul-modul yang sudah tersedia secara default di NodeJS. Dan perancangan kali ini akan

web server yang diperuntukan untuk website NodeJS yang akan dibangun. Sebenarnya dalam pembuatan website NodeJS, setiap kode sumber dapat dipisahkan kedalam beberapa file Javascript.

Tahap pertama dalam pembuatan server harus melakukan proses instalasi pada os raspbian. Klik pada ikon jaringan di kanan atas layar, ikon jaringan akan terlihat berbeda dari ikon sebelumnya. Tekan ikon tersebut, serta pilih nama wifi sesuai jaringan yang ada. Karena sebelum melakukan install nodejs harus download terlebih dahulu. Proses download dilakukan melalui terminal pada raspberry.

Membuka website nodejs dengan menuliskan perintah `$ Curl -sL https://deb.nodesource.com/setup_7.x | Sudo -E bash -` itu ke terminal, selanjutnya melakukan install nodejs dengan memberikan perintah `$ Sudo apt install nodejs` pada terminal maka melakukan proses instalasi, dan menunggu sampai proses selesai.

Setelah proses instalasi selesai, maka dapat membuat program pada text editor, untuk data program bisa dilihat pada lampiran dengan halaman sekian, program ini diberi nama server.js. Pada program server ini digunakan untuk menyimpan data base seperti file-file dari grbl seperti perintah program maupun kode-kode *G-Code*. Jadi seperti media penyimpanannya. Tanpa adanya server penyimpanan maka antarmuka tidak dapat berjalan dengan semestinya. Setelah program server selesai, maka langkah selanjutnya membuat konfigurasi port dan localhost, agar dapat bisa dibuka pada web browser dan dapat dijalankan. Proses pembuatan konfigurasi dengan melihat Gambar 3.16 dibawah ini:

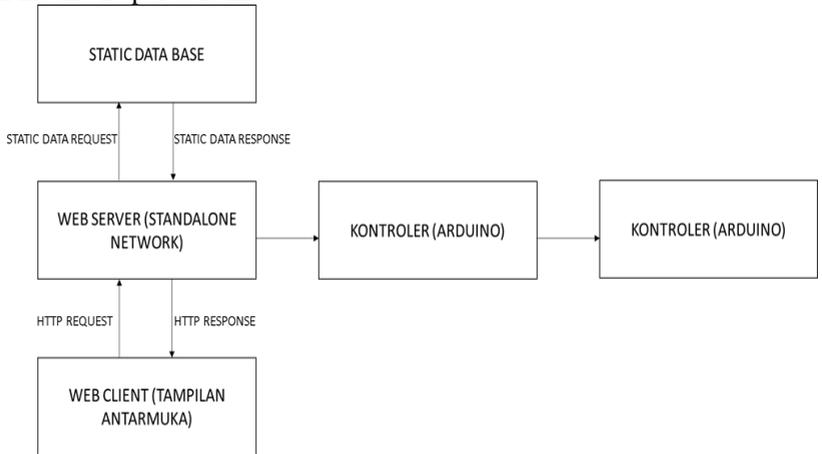


```
main.js | server.js
1 |
2 | var config = {};
3 |
4 | config.webPort = 80;
5 |
6 | config.serialBaudRate = 115200;
7 |
8 | config.useTTYAMA0 = 0;
9 |
10 | module.exports = config;
11 |
```

Gambar 3.16 Proses Pembuatan Konfigurasi

Pada config.webport bisa diisi dengan alamat port yang diinginkan, misalnya 8000, jadi nilai 80 bisa diganti dengan nilai 8000. Sedangkan pada setingan config.serialBaudRate bisa diganti sesuai dengan setingan pada komunikasi serial dengan arduino, tapi berhubung menggunakan grbl v0.9 maka serialnya harus 115200. Kalau tidak melakukan

penggantian tersebut maka serial akan tidak terkoneksi dan pada tampilan cmd raspberry akan mengalami masalah. Pada Gambar 3.17 di bawah merupakan blok diagram mengenai cara kerja server pada sebuah sistem mesin cnc portable ini:



Gambar 3.17 Diagram Blok Client Server pada Tampilan Antarmuka WEB

-----Halaman ini sengaja dikosongkan-----

BAB IV

HASIL PERANCANGAN DAN PENGUJIAN

Bab ini dibagi menjadi lima bagian. Pada bagian pertama dijelaskan mengenai hasil perancangan JavaScript dan Node Server pada antarmuka web. Pada hasil pengujian kedua yaitu perpindahan posisi sumbu Y. Pengujian ini dilakukan untuk mengetahui perpindahan motor stepper terhadap sumbu Y, apakah sudah sesuai dengan nilai yang dimasukkan dari tampilan antarmuka. Bagian ketiga adalah pengujian sensor limit switch sebagai sensor penanda dari batas akhir koordinat sumbu y. Dengan adanya sensor limit bisa mencegah terjadinya gerakan yang melebihi batas akhir sumbu Y. Pengujian bagian keempat adalah pengujian kode G-Code untuk menguji pembuatan suatu bidang dari kode G-Code sesuai dengan program konvensional. Data pengujian yang di ambil hanya pemrograman secara incremental dan pemrograman secara absolute. Pengujian data yang terakhir adalah pengujian kode G-Code dengan sistem CAD/CAM, pengujian ini diperlukan untuk menguji seberapa presisi suatu pola dari pengerjaan mesin CNC.

4.1 Hasil Perancangan Tampilan Antarmuka

Hasil perancangan menggunakan JavaScript mempunyai fungsi sebagai pembuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di web server. Hasil perancangan ini di dapat dengan tampilan kolom bar, yang mana nilainya dapat berubah bersamaan dengan perpindahan yang terjadi pada setiap sumbu mesin CNC. Tampilan tersebut dapat dilihat Gambar 4.1 di bawah ini:



Gambar 4.1 Tampilan Antarmuka otomatis

Dari gambar di atas merupakan hasil dari pemrograman JavaScript, dimana pada bagian nomor satu mempunyai fungsi untuk menampilkan nilai yang terukur dari pergerakan sumbu dari mesin CNC. Perpindahan nilai itu dikarenakan sebuah aksi dari program JavaScript untuk memberikan perintah dan membuat kolom nilai berubah sesuai dengan pergerakan setiap sumbu mesin CNC. Pada Gambar 4.2 memperlihatkan sebagian listing program mengenai pemberian nilai pada bagian nomor satu tersebut:

```
// memegang file g-code
if (window.FileReader) {

    var reader = new FileReader ();
    function dragEvent (ev) {
        ev.stopPropagation ();
        ev.preventDefault ();
        if (ev.type == 'drop') {
            reader.onloadend = function (ev) {
                document.getElementById('command').value = this.result;
                openGCodeFromText();
            };
            reader.readAsText (ev.dataTransfer.files[0]);
        }
    }
}
```

Gambar 4.2 Listing Program Kolom Pengerjaan

Pada bagian nomor dua, dari gambar 4.1 merupakan bagian penting dalam pemuatan alarm untuk mengetahui nilai kode G. Jadi dengan mengetahui kode G-Code Pengguna awal tidak akan mengalami kesulitan saat mengoperasikan sebuah mesin CNC, seperti pada Gambar 4.3 pembuatan listing mengenai alarm bawah ini:

```
var fileInput = document.getElementById('fileInput');
fileInput.addEventListener('change', function(e) {
    reader.onloadend = function (ev) {
        document.getElementById('command').value = this.result;
        openGCodeFromText();
    };
    reader.readAsText (fileInput.files[0]);
});
```

Gambar 4.3 Listing Program Kolom Input

4.2 Pengujian Perpindahan Posisi Sumbu Y

Pengujian motor stepper dilakukan untuk mengetahui seberapa presisi perpindahan motor stepper dari setiap sumbu pada mesin CNC. Bentuk fisik mesin CNC dapat dilihat Gambar pada lampiran D1. Pengujian ini dilakukan secara manual dengan menggunakan penggaris sebagai media pengukuran dan mode yang digunakan dari pengujian motor stepper ini menggunakan mode *quarter step*. Selanjutnya mengaktifkan panel HMI dan menekan tombol yang tersedia pada panel HMI.

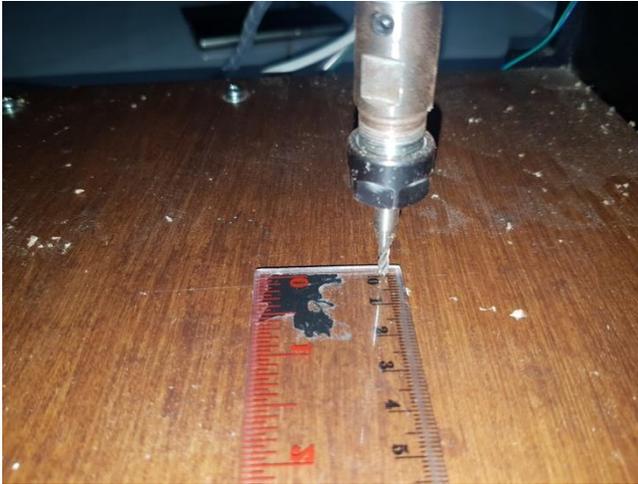
4.2.1 Perpindahan Sumbu Y sebelum Kalibrasi

Langkah-langkah untuk menguji perpindahan sumbu Y adalah dengan memberikan kode G-Code pada tampilan antarmuka dari parameter 1mm sampai dengan 10 mm, kemudian pada bidang kerja mesin CNC diberikan sebuah penggaris 20cm yang diletakkan tepat dibawah mata pahat mesin CNC. Penguji hanya mencocokkan apakah jarak yang diberikan pada tampilan antarmuka sudah sesuai dengan jarak sebenarnya. Dan pada Tabel 4.1 pengujian sumbu Y sebelum kalibrasi

Tabel 4.1 Pengujian Sumbu Y Sebelum Kalibrasi

<i>Current Step</i>	<i>Target</i>	<i>Actual</i>	<i>Error</i>
250 steps/mm	1 mm	16 mm	15 mm
250 steps/mm	2 mm	34 mm	32 mm
250 steps/mm	3 mm	50 mm	47 mm
250 steps/mm	4 mm	66 mm	62 mm
250 steps/mm	5 mm	84 mm	79 mm
250 steps/mm	6 mm	100 mm	94 mm
250 steps/mm	7 mm	117 mm	110 mm
250 steps/mm	8 mm	134 mm	126 mm
250 steps/mm	9 mm	150 mm	141 mm
250 steps/mm	10 mm	168 mm	158 mm
Rata-rata Nilai <i>Error</i>			86,4 mm

Dari tabel di atas terlihat nilai error yang begitu besar dari semestinya, dengan demikian perlu adanya proses kalibrasi untuk mengurangi nilai error. Dan mendapatkan nilai step yang diinginkan dengan menggunakan rumus yang telah ditentukan. Pada Gambar 4.4 merupakan posisi awal saat memulai kalibrasi.



Gambar 4.4 Posisi Awal Mesin CNC

Pada Gambar 4.3 merupakan titik awal dari mesin CNC atau posisi sumbu y berada pada jarak 0 mm. Ketika diberikan nilai sebesar 10 mm jarak sebenarnya tidak tepat 10 mm atau berada pada posisi 168mm, seperti pada Gambar 4.5 berikut ini.



Gambar 4.5 Jarak Sebenarnya dengan *Target* 10 mm

4.2.2 Perpindahan Sumbu Y Setelah Kalibrasi

Dari semua data yang terkumpul terlihat nilai *error* yang masih besar. Untuk mengurangi nilai error dilakukan kegiatan kalibrasi pada sumbu Y, dengan melakukan suatu perhitungan dengan rumus yang tertera di halaman berikutnya:

$c = \frac{a}{b}$
$c \times d = e$

Dimana:

a = jarak yang diinginkan (mm)

b =jarak sebenarnya (mm)

c =hasil kalkulasi (mm)

d =*current step* (steps/mm)

e =*new steps* (steps/mm)

Dari rumus di atas yang dimaksud dengan jarak yang di inginkan adalah jarak yang ingin dituju semisal jarak 10 cm, berarti sebuah jarak awal yang akan sebagai patokan. Sedangkan jarak sebenarnya merupakan jarak hasil dari pergerakan dari sebuah sumbu yang dijalankan melalu perintah sebelumnya. Hasil kalkulasi sendiri merupakan perbandingan dari jarak yang di inginkan dengan jarak sebenarnya. *Current step* sendiri merupakan nilai patokan step yang digunakan untuk mencari nilai step yang baru. Dan yang terakhir adalah *new step* merupakan step yang dicari untuk memperoleh hasil kalibrasi yang tepat dan sesuai hasil pengukuran yang sebenarnya.

Pada Tabel 4.1 diambil salah satu data untuk mencari nilai *new step* dengan menggunakan rumus perhitungan di atas. Hasil perhitungannya dapat dilihat pada Tabel 4.2 berikut ini.

Tabel 4.2 Perhitungan Mencari Nilai *New Step*

<i>Target</i>	<i>Actual</i>	<i>Calculated</i>	<i>Current Step</i>	<i>New Step</i>
1 mm	16 mm	0.063	250	15.63

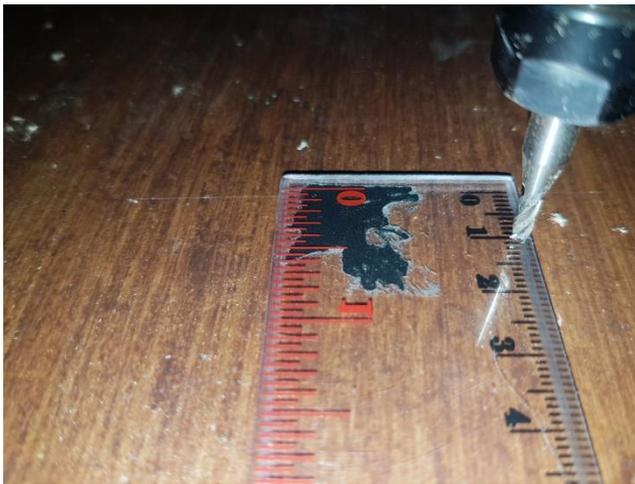
Tabel 4.3 Perpindahan Sumbu y Setelah Kalibrasi

<i>New step</i>	<i>Target</i>	<i>Actual</i>	<i>Error</i>
16,63 steps/mm	1 mm	1 mm	0 mm
16,63 steps/mm	2 mm	2,1 mm	0,1 mm
16,63 steps/mm	3 mm	3,5 mm	0,5 mm
16,63 steps/mm	4 mm	3,9 mm	0,1 mm

16,63 steps/mm	5 mm	5,1 mm	0,1 mm
16,63 steps/mm	6 mm	5,9 mm	0,1 mm
16,63 steps/mm	7 mm	6,9 mm	0,1 mm
16,63 steps/mm	8 mm	8,1 mm	0,1 mm
16,63 steps/mm	9 mm	8,9 mm	0,1 mm
16,63 steps/mm	10 mm	10 mm	0 mm
Rata-rata Nilai <i>Error</i>			0,12 mm

Dari data pengujian di atas setelah pengambilan data terkalibrasi, maka kesalahan error yang besar dapat teratasi dan hasil pengukuran atau perpindahan posisi hampir sesuai dengan yang diinginkan, serta mendapatkan nilai step/mm yang baru.

Diambil salah satu contoh pada nilai *target* 10 mm yang awalnya menjadi 168mm setelah dilakukan proses kalibrasi menjadi tepat 10 mm seperti pada Gambar 4.6 di bawah ini.



Gambar 4.6 Setelah Proses Kalibrasi

4.3 Pengujian *Limit Switch*

Pada pengujian *limit switch* dilakukan untuk mengetahui ketepatan dari komponen *limit switch* bekerja. Pengujian dilakukan dengan menggerakkan motor stepper pada jarak yang diinginkan, namun sebelum motor stepper menempuh jarak tersebut *limit switch* diaktifkan.

Dalam menghitung nilai *error* dapat dilihat pada perhitungan rumus di bawah ini.

$$Error = |Limit\ Switch\ ON\ (mm) - Jarak\ Sebenarnya\ (mm)|$$

Pada pengujian *Limit Switch* langkah-langkah yang harus diperhatikan pertama kali adalah mengaktifkan kode *limit switch* pada *software* Grbl dengan mengetikkan \$\$ untuk mengetahui konfigurasi dari *hard limit switch* itu sendiri. Setelah mengetahui kode dari *hard limit switch* maka perlu diaktifkan dengan mengetikkan \$21=1. Maka pada tampilan antarmuka akan ada pemberitahuan bahwa kode *hard limit switch* telah diaktifkan. Sehingga apabila saklar limit switch mengenai mesin CNC maka yang terjadi akan ada pemberitahuan bahwa sistem mesin CNC tidak dapat dioperasikan kecuali dengan menekan tombol reset pada *navigation bar* dan mengetikkan \$x pada tampilan antarmuka. Pada Tabel 4.4 merupakan hasil pengukuran limit switch:

Tabel 4.4 Pengukuran Nilai *Error Limit Switch* Sumbu *Y*

Limit Switch ON (mm)	Jarak yang sebenarnya (mm)	<i>Error</i>
8 mm	7.8 mm	0.2 mm
19 mm	19.5 mm	0.5 mm
30 mm	29.8 mm	0.2 mm
42 mm	42.8 mm	0.8 mm
58 mm	56.2 mm	1.8 mm
60 mm	59.5 mm	0.5 mm
78 mm	75.8 mm	2.2 mm
86 mm	85 mm	1 mm
94 mm	91 mm	3 mm
100 mm	100 mm	0 mm

4.4 Pengujian Kode G-Code Terhadap Sumbu *Y*

Pengujian kode G-Code ini menerapkan 2 sistem yaitu sistem inkremental dan sistem absolute. Tujuan dari pengujian data ini yaitu dapat mengetahui perbedaan dari sistem inkremental dan sistem absolute.

Pengambilan data dapat dimulai dari menyalakan sistem antarmuka dan menghubungkan sumbu *Y* ke kontroler serta

menghubungkan power supply dari setiap komponen. Selanjutnya penggaris diletakkan pada bidang kerja mesin CNC.

4.4.1 Sistem Inkremental

Pengujian data pada sistem incremental bertujuan untuk mengetahui bagaimana cara transisi perpindahan posisida titik awal perpindhan menuju ke titik tujuan. Tetapi pengujian data ini stelah sumbu y sudah terkalibrasi sebelumnya untuk mendapatkan hasil yang akurat, data pengujian dapat dilihat pada Tabel 4.5 halaman berikutnya :

Tabel 4.5 Pengukuran Perpindahan Sistem Inkremental

Titik Awal (mm)	Kode G-Code	Perpindahan (mm)	Titik Akhir (mm)
0 mm	G91 Y50	50 mm	50 mm
50 mm	G91 Y20	20 mm	70 mm
70 mm	G91 Y30	30 mm	100 mm
100 mm	G91 Y50	50 mm	150 mm
150 mm	G91 Y100	50 mm	250 mm
250 mm	G91 Y-100	-100 mm	150 mm
150 mm	G91 Y-50	-50 mm	100 mm
100 mm	G91 Y-30	-30 mm	70 mm
70 mm	G91 Y-20	-20 mm	50 mm
50 mm	G91 Y-50	-50 mm	0 mm

4.4.2 Sistem Absolute

Pengujian data pada sistem Absolute bertujuan untuk mengetahui bagaimana cara transisi perpindahan posisida titik awal perpindhan menuju ke titik tujuan, data pengujian dapat dilihat pada Tabel 4.6 di bawah ini:

Tabel 4.6 Pengukuran Perpindahan Sistem Absolute

Titik Awal (mm)	Kode G-Code	Perpindahan (mm)	Titik Akhir (mm)
0 mm	G90 Y50	50 mm	50 mm
50 mm	G90 Y70	20 mm	70 mm
70 mm	G90 Y100	30 mm	100 mm
100 mm	G90 Y150	50 mm	150 mm
150 mm	G90 Y200	50 mm	200 mm

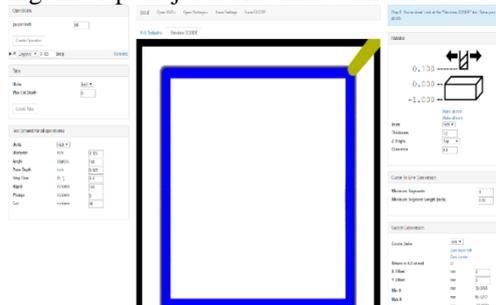
200 mm	G90 Y150	50 mm	150 mm
150 mm	G90 Y180	30 mm	180 mm
180 mm	G90 Y100	-80 mm	100 mm
180 mm	G90 Y70	-110 mm	70 mm
70 mm	G90 Y0	-70 mm	0 mm

4.5 Pengujian G-Code dengan CAD dan CAM

Pengujian G-Code dengan CAD/CAM berupa file gambar berformat .Gcode yang telah didesain dan di konversi terlebih dahulu pada *software* CAD/CAM. Pengujian ini dilakukan guna mengetahui apakah gambar yang telah didesain dapat diterjemahkan dengan mesin CNC Portable ini. Terdapat beberapa pengujian data gambar diantaranya gambar lingkaran dengan mode engrave, gambar lingkaran dengan mode pocket, gambar persegi dengan mode engrave, gambar persegi dengan mode pocket.

4.5.1 Persegi Mode Engrave

Pengujian dilakukan dengan membuat desain bangun datar berbentuk persegi pada *software* Inkscape. Langkah-langkah dalam pembuatan dengan membuat gambar persegi menggunakan *software* corel drawing maupun inkscape. Selanjutnya gambar di simpan dengan format svg, kemudian mengupload ke jscut dan di koversi menjadi kode *G-Code*. Hasil gambar pada jscut bisa lihat Gambar 4.7 bawah ini:



Gambar 4.7 Gambar Persegi

Hanya saja pada pengambilan data ini menggunakan bangun datar berbentuk persegi, sehingga pada Inkscape dipilih *tool* dengan bentuk Persegi. Diameter dari Persegi yang diinginkan sebesar 50 mm. Hasil

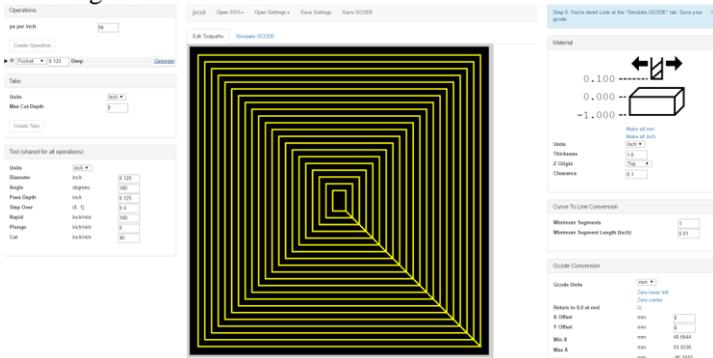
dari eksekusi gambar persegi dapat dilihat pada Gambar 4.8 Hasil uji yang didapatkan bahwa lingkaran terbentuk dengan diameter sebesar 50mm sesuai dengan desain yang telah dibuat:



Gambar 4.8 Gambar Persegi Mode Engrave

4.5.2 Persegi Mode Pocket

Pada pengujian kedua pada persegi akan dilakukan pengujian dengan model pocket, dimana mode pocket itu sendiri mode pengujian yang akan dilakukan secara pengeboran secara merata jadi prosesnya akan lebih lama daripada proses engrave, pengujian pada bangun persegi di bentuk dengan ukuran diameter.



Gambar 4.9 Konversi Gambar Persegi Mode Pocket

Pada Gambar 4.9 di atas merupakan hasil dari jsut pada bentuk persegi, terlihat garis kuning yang merata pada bagian dalam persegi, yang merupakan lintasan dari proses pembentukan dari model pocket,

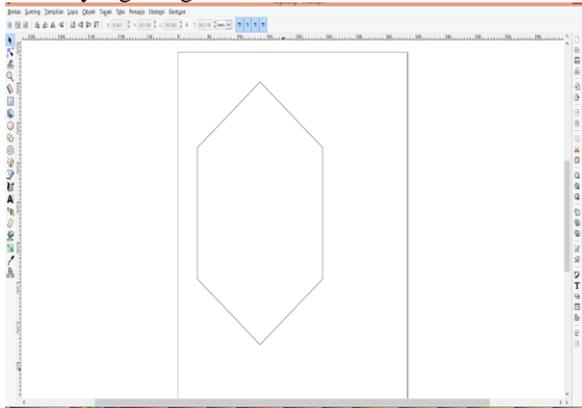
dan proses pengerjaan juga. Hasil pengerjaan dapat dilihat pada Gambar 4.10 di bawah ini:



Gambar 4.10 Hasil Gambar Persegi Mode Pocket

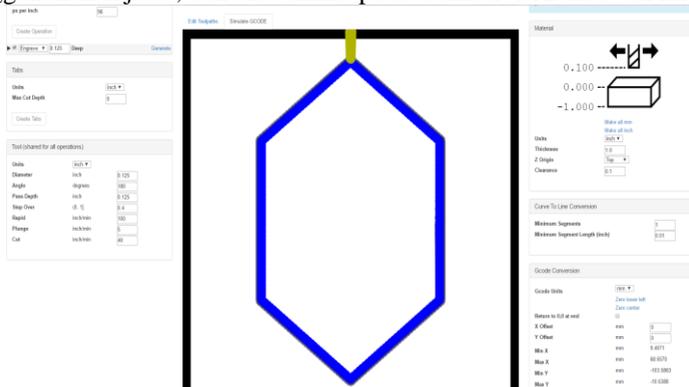
4.5.3 Segienam Mode Engrave

Pengujian dilakukan dengan membuat desain bangun datar berbentuk lingkaran pada *software* Inkscape. Hanya saja pada pengambilan data ini menggunakan bangun datar berbentuk lingkaran, sehingga pada Inkscape dipilih tool dengan bentuk Persegi. Diameter dari Segi Enam yang diinginkan sebesar 50 mm.



Gambar 4.11 Pembuatan Gambar Segienam pada Inkscape

Dari Gambar 4.11 merupakan gambar pembuatan yang dilakukan dari software inkscape, setelah proses pembuatan selesai dari inkscape. Maka proses selanjutnya akan dilanjutkan ke tahap konversi menggunakan jscut, dan bisa dilihat pada Gambar 4.12 dibawah ini:



Gambar 4.12 Konversi Gambar Segienam Mode Engrave pada Jscut

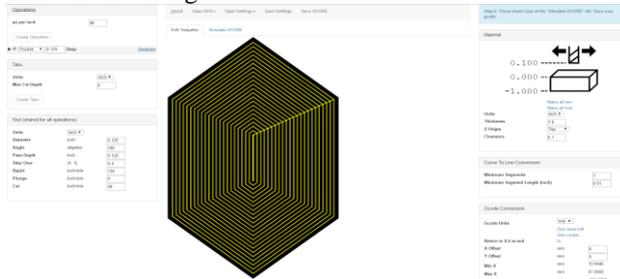
Pada gambar di atas merupakan hasil dari jscut pada bentuk persegi, terlihat garis kuning yang merata pada bagian dalam persegi, yang merupakan lintasan dari proses pembentukan dari model pocket, dan proses pengerjaan juga. Hasil pengerjaan dapat dilihat pada Gambar 4.13 di bawah ini:



Gambar 4.13 Hasil Gambar Segienam Mode Engrave

4.5.4 Segienam Mode Pocket

Pada pengujian kedua pada pergienam, akan dilakukan pengujian dengan model pocket, dimana mode pocket itu sendiri mode pengujian yang akan dilakukan secara pengeboran secara merata jadi prosesnya akan lebih lama daripada proses engrave, pengujian pada bangun persegienam di bentuk dengan ukuran diameter



Gambar 4.14 Konversi Gambar Segienam Mode Pocket pada Jscut

Pada Gambar 4.14 di atas merupakan hasil dari jscut pada bentuk persegienam, terlihat garis kuning yang merata pada bagian dalam persegienam, yang merupakan lintasan dari proses pembentukan dari model pocket, dan proses pengerjaan juga. Hasil pengerjaan dapat dilihat pada Gambar 4.15 di bawah ini:



Gambar 4.15 Hasil Gambar Segienam Mode Pocket

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari hasil pengujian ini, didapatkan sebuah sistem antarmuka yang dapat menjalankan mesin CNC dengan baik, serta dapat memproses gambar yang di *upload* ke sistem antarmuka. Data-data pengujian dari gambar juga bisa membuktikan bahwa kerja antara sistem client yang sebagai pengguna dapat menjalankannya dengan baik. Dan server dari atarmuka yang masih berjalan secara offline juga berjalan tanpa ada kendala, serta pengujian pergerakan mesin CNC terhadap sumbu Y dengan proses kalibrasi yang didapat sebuah data rata-rata error sebesar 0,12 mm dan nilai step baru sebesar 16,61 steps/mm. Jadi apabila diberikan sebuah perintah dengan target 9 mm, mesin mampu menurunkan pergerakan dengan perpindahan sebesar 8,9 mm.

Saran untuk penelitian selanjutnya, sistem antarmuka bisa di tambahkna aplikasi konversi gambar menjadi file *g-code* jadi untuk melakukan konversi tidak perlu bantuan konversi dari web lain maupun aplikasi lain, sehingga pengguna tidak perlu membuka aplikasi lainnya, selain aplikasi antarmuka yang dibuat. Pada saran selanjutnya diharapkan web bisa berjalan secara online jadi bisa dikendalikan melalui handphone maupun pc yang sudah terhubung dengan server.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Fatimah. Nisvi, "Perancangan Interpreter G-Code untuk Gerak Linier pada Mesin CNC Milling", Fakultas Teknik Industri- Institut Teknologi Sepuluh Nopember, 2016.
- [2] Krar, Steve. Computer Numerical Control Programming Basics. Desember 2016. Industrial Press.,pdf..
- [3] Smid, Peter.2005. CNC Programming Techniques:An Insider's Guide to Effective Methods and Applications. New York:Madison Avenue
- [4] Polulu. A4988 Stepper Motor Driver Carrier, 10 Januari 2011, 10.56[dikutip 12 September 2016]. Tersedia dari:<https://www.pololu.com/product/1182>
- [5] It-Jurnal. Pengertian dan Kelebihan Arduino, 21 Agustus 2016, 13.43[dikutip 10 September 2016]. Tersedia dari:<https://www.it-jurnal.com/pengertian-dan-kelebihan-arduino/>
- [6] Desain-Manufaktur-Blog. Interpolasi Melingkar, 16 Oktober 2015, 18.34[dikutip 12 Desember 2016]. Tersedia dari:<http://desainmanufaktur.bayuwiro.net/index.php/2015/10/16/interpolasi-melingkar/>
- [7] Mercubuana. BAB II Landasan Teori, 11 April 2014, 17.12[dikutip 05 Januari 2017]. Tersedia dari:http://digilib.mercubuana.ac.id/manager/n@file_skripsi/Isi2801571026013.pdf
- [8] Desrizal. JavaScript Guide. Juni 2016. www.codingwear.com,.pdf
- [9] Duckett, Jon. 2011. JavaScript & JQuery: Interactive front-end web development. John Wiley and Sons.Inc
- [10]Teixeira, Pedro. 2013. Professional Node.js: Building JavaScript Based Scalable Software.Indianapolis:JohnWiley & Sons,Inc.
- [11]Brown, Ethan. 2014. Web Development With Node & Express :Leveraging The Javascripts Stack. Gravenstein Highway Nort:O'REILLY Media.

[12]nodeid. Mengenal Node.js Node.js Indonesia, 1 Desember 2016, 11.34[dikutip 18 September 2016]. Tersedia dari:<https://nodeid.wordpress.com/2014/12/01/mengenal-node-js/>

LAMPIRAN A

A.1. G-Code

- G00 Pengeposisian bebas
- G01 Interpolasi Lurus (gerak pemakanan)
- G02 Interpolasi melingkar searah jarum jam (CW)
- G03 Interpolasi melingkar berlawanan arah jarum jam (CCW)
- G04 Program berhenti pada waktu tertentu
- G10 Data program dapat di-input
- G15 Pembatalan perintah koordinat polar
- G16 Perintah koordinat polar
- G17 Interpolasi helical
- G20 Konversi satuan inci (british)
- G21 Konversi satuan mm (metric)
- G28 Pengembalian posisi referensi
- G31 perintah skip (melangkahi)
- G33 Pembuatan ulir (Threading cutting)
- G40 Cancel kompensasi cutter
- G43, G44 Kompensasi panjang tool positif(G43), Negatif (G44)
- G45 Menaikkan offset tool
- G46 Menurunkan offset tool
- G47 Menaikkan ganda offset tool
- G48 Menurunkan ganda offset tool
- G49 Pembatalan kompensasi panjang tool
- G52 Penyatuan system koordinat local
- G54, G55, G56, G57 Sistem koordinat workpiece
- G60 Pengeposisian arah tunggal
- G63 Pengerjaan Tapping (ulir dalam)
- G64 Pengerjaan pemotongan
- G74 Menghitung putaran Tapping
- G76 Pengerjaan proses canned cycle
- G80 Pembatalan pengerjaan siklus
- G81 Pengoperasian eksernal atau putaran drilling
- G82 Keliling counter boring
- G83 Peck drilling cycle
- G84 Pengerjaan keliling Tapping
- G85 Pengerjaan keliling boring
- G90 Perintah system koordinat absolute

G91 Perintah system koordinat incremental
G94 Penentuan asutan pemakanan dalam (inchi/menit)
G95 Penentuan asutan pemakanan dalam (inchi/putaran)
G96 Kecepatan potong permukaan konstan
G98 Kembali ke titik initial di sebuah siklus
G99 Kembali ke titik R di sebuah siklus

LAMPIRAN B

B.1 Datasheet Raspberry Pi 3 Model B

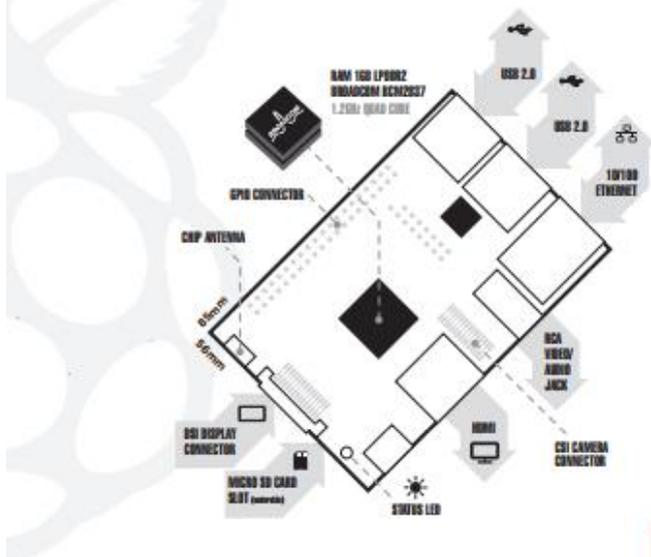


Raspberry Pi 3 Model B

Product Name Raspberry Pi 3

Product Description The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. While maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

RS Part Number 896-8660



www.rs-components.com/raspberrypi



Raspberry Pi

Raspberry Pi 3 Model B

Specifications

Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A

Connectors:

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	Push/pull Micro SDIO

Key Benefits

- Low cost
- 10x faster processing
- Consistent board format
- Added connectivity

Key Applications

- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming

B.2 Datasheet Driver Stepper A4988



A4988

DMOS Microstepping Driver with Translator And Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $1/2$, $1/4$, $1/8$, and $1/16$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Description

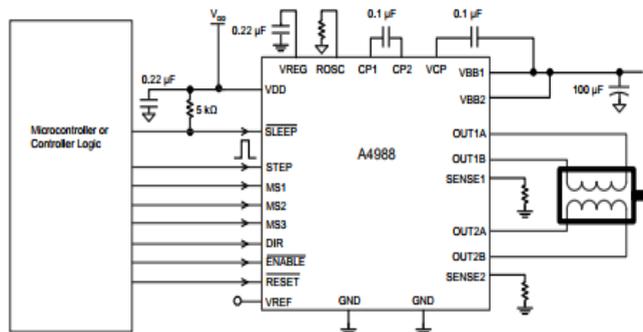
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ± 2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page...

Typical Application Diagram



A4988

DMOS Microstepping Driver with Translator And Overcurrent Protection

Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix -T), with 100% matte tin plated leadframes.

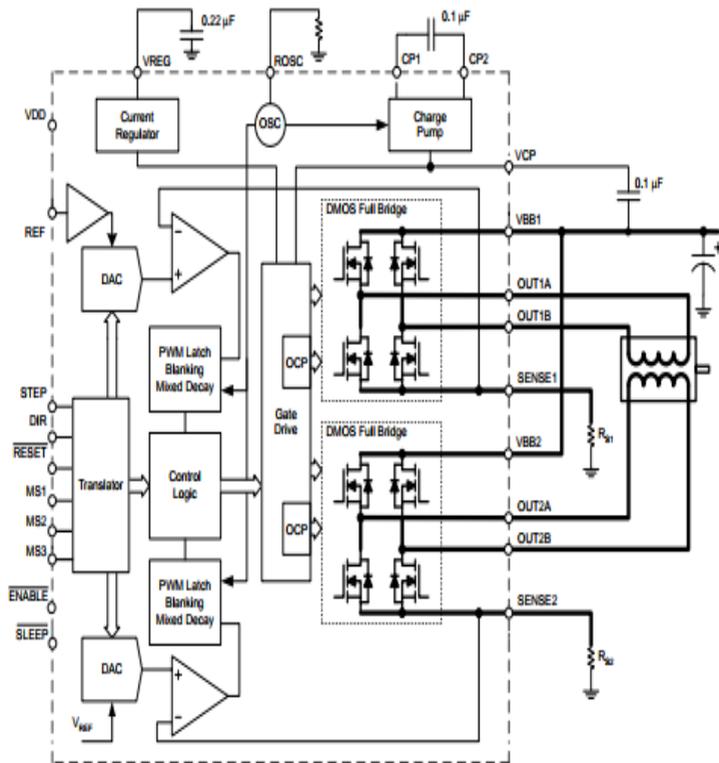
Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BH}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		-0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		-0.3 to 5.5	V
Motor Outputs Voltage			-2.0 to 37	V
Sense Voltage	V_{SENSE}		-0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	-20 to 85	°C
Maximum Junction	$T_J(\max)$		150	°C
Storage Temperature	T_{STG}		-55 to 150	°C

Functional Block Diagram



ELECTRICAL CHARACTERISTICS¹ at $T_A = 25^\circ\text{C}$, $V_{BB} = 35\text{ V}$ (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. ²	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	-	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	-	5.5	V
Output On Resistance	$R_{DS(ON)}$	Source Driver, $I_{OUT} = -1.5\text{ A}$	-	320	430	m Ω
		Sink Driver, $I_{OUT} = 1.5\text{ A}$	-	320	430	m Ω
Body Diode Forward Voltage	V_F	Source Diode, $I_F = -1.5\text{ A}$	-	-	1.2	V
		Sink Diode, $I_F = 1.5\text{ A}$	-	-	1.2	V
Motor Supply Current	I_{BB}	$f_{PWM} < 50\text{ kHz}$	-	-	4	mA
		Operating, outputs disabled	-	-	2	mA
Logic Supply Current	I_{DD}	$f_{PWM} < 50\text{ kHz}$	-	-	8	mA
		Outputs off	-	-	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		$V_{DD} \times 0.7$	-	-	V
	$V_{IN(0)}$		-	-	$V_{DD} \times 0.3$	V
Logic Input Current	$I_{IN(1)}$	$V_{IN} = V_{DD} \times 0.7$	-20	<1.0	20	μA
	$I_{IN(0)}$	$V_{IN} = V_{DD} \times 0.3$	-20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	-	100	-	k Ω
	R_{MS2}	MS2 pin	-	50	-	k Ω
	R_{MS3}	MS3 pin	-	100	-	k Ω
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	μs
		$R_{OSC} = 25\text{ k}\Omega$	23	30	37	μs
Reference Input Voltage Range	V_{REF}		0	-	4	V
Reference Input Current	I_{REF}		-3	0	3	μA
Current Trip-Level Error ³	err_1	$V_{REF} = 2\text{ V}$, % $_{TRIMMAX} = 38.27\%$	-	-	± 15	%
		$V_{REF} = 2\text{ V}$, % $_{TRIMMAX} = 70.71\%$	-	-	± 5	%
		$V_{REF} = 2\text{ V}$, % $_{TRIMMAX} = 100.00\%$	-	-	± 5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold ⁴	I_{OCPST}		2.1	-	-	A
Thermal Shutdown Temperature	T_{TSD}		-	165	-	$^\circ\text{C}$
Thermal Shutdown Hysteresis	T_{TSDHYS}		-	15	-	$^\circ\text{C}$
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLHYS}$		-	90	-	mV

¹For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

²Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

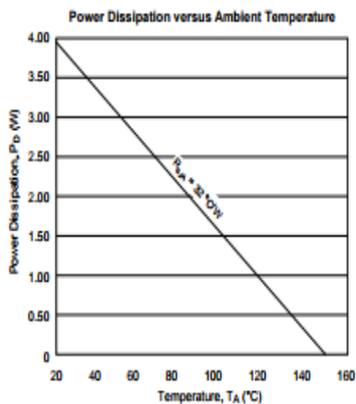
³ $V_{ERR} = [(V_{REF}/8) - V_{SENSE}] / (V_{REF}/8)$.

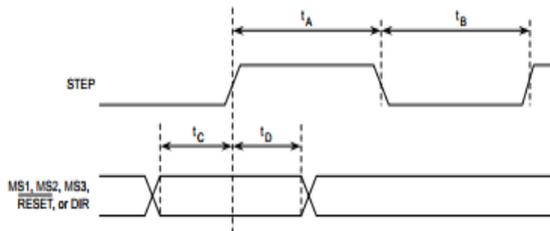
⁴Overcurrent protection (OCP) is tested at $T_A = 25^\circ\text{C}$ in a restricted range and guaranteed by characterization.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Test Conditions*	Value	Units
Package Thermal Resistance	$R_{\theta JA}$	Four-layer PCB, based on JEDEC standard	32	$^{\circ}\text{C/W}$

*Additional thermal information available on Allegro Web site.





Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	μs
STEP minimum, LOW pulse width	t_B	1	μs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns

Figure 1: Logic Interface Timing Diagram

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

LAMPIRAN C

C.1 Listing Program Javascript untuk Antarmuka

```
$(document).ready(function() {

    $( window ).resize(function() {
        // when header resizes, move ui down
        $('.table_layout').css('margin_top',$('.navbar-collapse').height()-34);
    });

    var socket = io.connect("");

    socket.on('serverError', function (data) {
        alert(data);
    });

    socket.on('gcodeFromJscut', function (data) {
        $('#command').val(data.val);
        openGCodeFromText();
        alert('new data from jscut');
    });

    socket.on('ports', function (data) {
        //console.log('ports event',data);
        $('#choosePort').html('<option    val="no">Select a
serial port</option>');
        for (var i=0; i<data.length; i++) {
            $('#choosePort').append('<option
value="'+i+'">'+data[i].comName+':'+data[i].pnpId+'</option>');
        }
        if (data.length == 1) {
            $('#choosePort').val('0');
            $('#choosePort').change();
        }
    });

    socket.on('qStatus', function (data) {
```

```

$('#qStatus').html(data.currentLength+'/'+data.currentMax);
});

socket.on('machineStatus', function (data) {
    $('#mStatus').html(data.status);
    $('#mX').html('X: '+data.mpos[0]);
    $('#mY').html('Y: '+data.mpos[1]);
    $('#mZ').html('Z: '+data.mpos[2]);
    $('#wX').html('X: '+data.wpos[0]);
    $('#wY').html('Y: '+data.wpos[1]);
    $('#wZ').html('Z: '+data.wpos[2]);
    //console.log(data);
});

socket.on('serialRead', function (data) {
    $('#console').append(data.line);
    $('#console').scrollTop($("#console")[0].scrollHeight -
$("#console").height());
});

$('#choosePort').on('change', function() {
    // pilih port
    socket.emit('usePort', $('#choosePort').val());
    $('#mStatus').html('Port Selected');
})

$('#sendReset').on('click', function() {
    socket.emit('doReset', 1);
});

$('#sendGrblHelp').on('click', function() {
    socket.

('gcodeLine', { line: '$' });
});

$('#sendGrblSettings').on('click', function() {
    socket.emit('gcodeLine', { line: '$$' });
});

```

```

});

$('#pause').on('click', function() {
    if ($('#pause').html() == 'Pause') {
        // pause queue on server
        socket.emit('pause', 1);
        $('#pause').html('Unpause');
        $('#clearQ').removeClass('disabled');
    } else {
        socket.emit('pause', 0);
        $('#pause').html('Pause');
        $('#clearQ').addClass('disabled');
    }
});

$('#clearQ').on('click', function() {

    socket.emit('clearQ', 1);
    $('#pause').click();

});

$('#mpC').on('click', function() {
    $('#mpA').addClass('active');
    $('#wpA').removeClass('active');
    $('#mPosition').show();
    $('#wPosition').hide();

});

$('#wpC').on('click', function() {
    $('#wpA').addClass('active');
    $('#mpA').removeClass('active');
    $('#wPosition').show();
    $('#mPosition').hide();

});

$('#sendZero').on('click', function() {
    socket.emit('gcodeLine', { line: 'G92 X0 Y0 Z0' });
});

```

```

    $('#sendCommand').on('click', function() {
        socket.emit('gcodeLine', { line: $('#command').val()
    });
        $('#command').val("");
    });

    //
    $('#command').keydown(function (e) {
        if (e.shiftKey) {
            var keyCode = e.keyCode || e.which;
            if (keyCode == 13) {
                //
                $('#sendCommand').click();
                //
                e.preventDefault();
            }
        }
    });

    $('#xM').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ ' X-'+$('#jogSize').val()+'\nG90'});
    });
    $('#xP').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ ' X'+$('#jogSize').val()+'\nG90'});
    });
    $('#yP').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ ' Y'+$('#jogSize').val()+'\nG90'});
    });
    $('#yM').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ ' Y-'+$('#jogSize').val()+'\nG90'});
    });
    $('#zP').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ ' Z'+$('#jogSize').val()+'\nG90'});
    });

```

```

    });
    $('#zM').on('click', function() {
        socket.emit('gcodeLine', { line: 'G91\nG1
F'+$('#jogSpeed').val()+ 'Z-'+$('#jogSize').val()+'\nG90'});
    });

// WASD and up/down keys
$(document).keydown(function (e) {
    var keyCode = e.keyCode || e.which;

    if ($('#command').is(':focus')) {
        //
        return;
    }
    switch (keyCode) {
    case 65:
        // a tombol X-
        e.preventDefault();
        $('#xM').click();
        break;
    case 68:
        // d tombol X+
        e.preventDefault();
        $('#xP').click();
        break;
    case 87:
        // w tombol Y+
        e.preventDefault();
        $('#yP').click();
        break;
    case 83:
        // s tombol Y-
        e.preventDefault();
        $('#yM').click();
        break;
    case 38:
        // Z+ naik
        e.preventDefault();
        $('#zP').click();

```

```

        break;
    case 40:
        // Z- menurun
        e.preventDefault();
        $('#zM').click();
        break;
    }
});
// upload gcode
if (window.FileReader) {

    var reader = new FileReader ();

    // drag and tempelkan
    function dragEvent (ev) {
        ev.stopPropagation ();
        ev.preventDefault ();
        if (ev.type == 'drop') {
            reader.onloadend = function (ev) {

                document.getElementById('command').value = this.result;
                openGCodeFromText();
            };
            reader.readAsText
(ev.dataTransfer.files[0]);
        }
    }

    document.getElementById('command').addEventListener
('dragenter', dragEvent, false);

    document.getElementById('command').addEventListener
('dragover', dragEvent, false);

    document.getElementById('command').addEventListener
('drop', dragEvent, false);
    // tombol
    var fileInput = document.getElementById('fileInput');
    fileInput.addEventListener('change', function(e) {

```

```

        reader.onloadend = function (ev) {

document.getElementById('command').value = this.result;
            openGCodeFromText();
        };
        reader.readAsText (fileInput.files[0]);
    });

} else {
    alert('command preventDefault');
}

});

```

C.2 Listing Program Node Server pada Tampilan Antarmuka

```

var config = require('./config');
var serialport = require("serialport");
var SerialPort = serialport.SerialPort; // localize object constructor
var app = require('http').createServer(handler)
    , io = require('socket.io').listen(app)
    , fs = require('fs');
var static = require('node-static');
var EventEmitter = require('events').EventEmitter;
var url = require('url');
var qs = require('querystring');

app.listen(config.webPort);
var fileServer = new static.Server('./i');

function handler (req, res) {

    //console.log(req.url);

    if (req.url.indexOf('/api/uploadGcode') == 0 && req.method ==
'POST') {

        // this is a gcode upload, probably from jscut
        console.log('new data from jscut');
        var b = "";
        req.on('data', function (data) {

```

```

        b += data;
        if (b.length > 1e6) {
            req.connection.destroy();
        }
    });
    req.on('end', function() {
        var post = qs.parse(b);
        //console.log(post);
        io.sockets.emit('gcodeFromJscut',
{'val':post.val});
        res.writeHead(200,          {"Content-Type":
"application/json"});
        res.end(JSON.stringify({'data':'ok'}));
    });
    } else {
        fileServer.serve(req, res, function (err, result) {
            if (err) console.log('fileServer error: ',err);
        });
    }
}

function ConvChar( str ) {
    c = {'<':'&lt;','>':'&gt;','&':'&amp;','"':'&quot;','"':'&#039;',
        '#':'&#035;'};
    return str.replace(/[<&>"#]/g, function(s) { return c[s]; });
}

var sp = [];
var allPorts = [];

serialport.list(function (err, ports) {

    // if on rPi - http://www.hobbytronics.co.uk/raspberry-pi-serial-
port
    if (fs.existsSync('/dev/ttyAMA0') && config.usettyAMA0 ==
1) {
        ports.push({comName:'/dev/ttyAMA0',manufacturer:
undefined,pnpId: 'raspberrypi__GPIO'});
    }
}

```

```
        console.log('adding /dev/ttyAMA0 because it is
enabled in config.js, you may need to enable it in the os -
http://www.hobbytronics.co.uk/raspberry-pi-serial-port');
    }
```

```
allPorts = ports;
```

```
for (var i=0; i<ports.length; i++) {
!function outer(i){
```

```
    sp[i] = {};
    sp[i].port = ports[i].comName;
    sp[i].q = [];
    sp[i].qCurrentMax = 0;
    sp[i].lastSerialWrite = [];
    sp[i].lastSerialReadLine = "";
    // 1 means clear to send, 0 means waiting for response
    sp[i].handle = new SerialPort(ports[i].comName, {
        parser: serialport.parsers.readline("\n"),
        baudrate: config.serialBaudRate
    });
    sp[i].sockets = [];
```

```
    sp[i].handle.on("open", function() {
```

```
        console.log('connected to '+sp[i].port+' at
'+config.serialBaudRate);
```

```
        // line from serial port
        sp[i].handle.on("data", function (data) {
            serialData(data, i);
        });
```

```
        // loop for status ?
        setInterval(function() {
            // console.log('writing ? to serial');
            sp[i].handle.write('?');
        }, 1000);
```

```
    });  
    }  
    }  
});
```

LAMPIRAN D

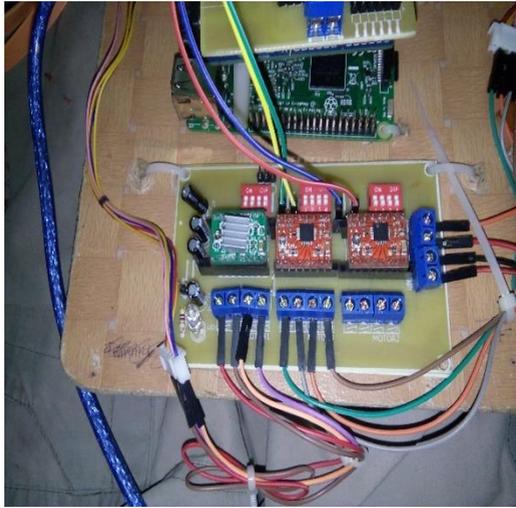
D.1 Dokumentasi Mesin CNC Portable



D.2 Dokumentasi Pengujian Kode G-Code



D.3 Hasil Rangkain Shield Driver Motor dan Shield Arduino



-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Prayoga Riono Putra
TTL : Sidoarjo, 29 Mei 1996
Agama : Islam
Alamat : Ds. Janti Kec. Tulangan
Telp/HP : 08991109479
Email : yogarionoputra@gmail.com

RIWAYAT PENDIDIKAN

1. 2002-2008 : SDN Janti II
2. 2008-2011 : SMP Negeri 1 Tulangan
3. 2011-2014 : SMA Negeri 1 Krebung
4. 2014-2017 : D3 Teknik Elektro Otomasi, Program Studi Komputer Kontrol – Fakultas Vokasi – Institut Teknologi Sepuluh Nopember

PENGALAMAN KERJA

1. Kerja Praktek di PT PJB PLN GRESIK

PENGALAMAN ORGANISASI

1. Staff BPU JMMI 2015-2016
2. Staff KWU LDJ SALAF 2015-2016
3. ASISTEN COMPUTER LABORATORY 2016-2017
4. Pengurus Harian Bendahara 1 LDJ SALAF 2016-2017

-----Halaman ini sengaja dikosongkan-----