Modifikasi Robot *Gripper Manipulator* Dengan Motor *Stepper* Sebagai Media Pembelajaran

PROYEK AKHIR

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

Disusun Oleh :

INDARINI	NIRM : 0031612
SATRIA HUTOMO JATY	NIRM : 0031623

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

2019

LEMBAR PENGESAHAN

Modifikasi Robot *Gripper Manipulator* Dengan Motor *Stepper* Sebagai Media Pembelajaran

Oleh :

Indarini	NIRM : 0031612
Satria Hutomo Jaty	NIRM : 0031623

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

Pembimbing 1	Pembimbing 2
I Made Andik Setiawan, Ph.D.	Surojo, M.T.

Penguji 1	Penguji 2	Penguji 3
Zanu Saputra, M.T.	M. Iqbal Nugraha, M.Eng.	Yudhi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Indarini NIRM : 0031612
Nama Mahasiswa 2 : Satria Hutomo Jaty NIRM : 0031623

Dengan Judul : Modifikasi Robot *Gripper Manipulator* Dengan Motor *Stepper* Sebagai Media Pembelajaran

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

Sungailiat, 2019

Nama Mahasiswa	Tanda Tangan
1. Indarini	.....
2. Satria Hutomo Jaty	.....

ABSTRAK

Motor stepper adalah salah satu jenis motor yang sangat akurat dalam pengaturan posisi dengan cara mengatur pulsa sinyal yang diberikan padanya sehingga derajat perputaran motor dapat diatur. Oleh karena itu motor stepper semakin banyak digunakan dalam kehidupan sehari-hari, misalnya pada mesin fotokopi dan printer. Penggunaan yang semakin meningkat mengakibatkan perlunya kendali posisi dan kecepatan pada motor stepper untuk membuat penggunaan motor stepper semakin fleksibel. Proyek akhir ini terdiri dari dua bagian, yaitu bagian kendali kecepatan dan bagian kendali posisi yang sudah dapat diatur oleh operator. Pada kendali posisi, motor stepper akan berputar kemudian berhenti pada posisi yang dimasukan oleh operator melalui 2 cara yaitu secara manual menggunakan push button dan melalui serial monitor. Kendali posisi dan kecepatan motor stepper diimplementasikan dengan menggunakan mikrokontroler jenis arduino yang diprogram dengan menggunakan sofware arduino itu sendiri. Sebagai penggerak lengan robot digunakan 2 motor stepper pada masing-masing joint, sehingga lengan robot memiliki 2 derajat kebebasan yang dapat bergerak dalam ruang 2 dimensi dan mengggunakan alat cekam sebagai akuatornya.

Kata kunci : media pembelajaran, motor stepper, posisi, robot.

ABSTRACT

A stepper motor is one type of motor that is accurate in position positioning by adjusting the signal pulses given to it so that the motor rotation rate can be adjusted. Therefore stepper motors are increasingly used in everyday life, for example on copiers and printers. The increasing use results in the need for position and speed control on the stepper motor to make the use of stepper motors more flexible. This final project consists of two parts, namely the speed control section and the position control section that can be arranged by the operator. In position control, the stepper motor will rotate then stop at the position entered by the operator through two ways, namely manually using the push button and through the serial monitor. Control of the position and speed of the stepper motor is implemented using an Arduino microcontroller that is programmed using Arduino software itself. As a robotic arm drive, 2 stepper motors are used at each joint, so that the robotic arm has 2 degrees of freedom that can move in 2-dimensional space and use a concave device as its amplifier.

Keywords: instructional media, position, robot, stepper motor.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan laporan Proyek Akhir ini dengan baik. Adapun laporan Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III (D-III) di Politeknik Manufaktur Negeri Bangka Belitung. Dalam Proyek Akhir ini penulis membuat sebuah Modifikasi Robot *Gripper* Manipulator Dengan Motor *Stepper* Sebagai Media Pembelajaran.

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

1. Orang tua serta keluarga yang selalu memberikan kasih sayang, doa serta dukungan.
2. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Aan Febriansyah, M.T, selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulistyo, M.T, selaku Kepala Prodi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak I Made Andik Setiawan, Ph.D, selaku pembimbing I yang telah meluangkan banyak waktu memberi saran-saran dan solusi dari masalah-masalah yang penulis hadapi selama proses penyusunan Makalah Proyek Akhir ini dan Bapak Surojo, M.T selaku pembimbing II yang telah meluangkan banyak tenaga, dan pikiran di dalam memberikan pengarahan dalam penulisan makalah Proyek Akhir ini.
6. Seluruh staf pengajar dan karyawan di Politeknik Manufaktur Negeri Bangka Belitung.

7. Rekan-rekan mahasiswa tingkat akhir Politeknik Manufaktur Negeri Bangka Belitung.
8. Teman-teman yang telah ikut mendukung dan memberikan bantuan serta masukan dalam pembuatan Proyek Akhir ini.
9. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam laporan Proyek Akhir ini masih jauh dari sempurna, untuk itu penulis sangat mengharapkan semua jenis saran, kritik dan masukan yang bersifat membangun dalam rangka perbaikan laporan ini. Demikian laporan ini dibuat dan semoga laporan ini dapat bermanfaat dan menambah wawasan bagi pembaca. Akhir kata penulis mengucapkan terima kasih.

Sungailiat, 2019

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Semakin berkembangnya dunia teknologi khususnya dalam bidang robotika, terdapat berbagai jenis robot yang bermanfaat dalam kehidupan manusia. Berbagai bentuk robot juga dikembangkan sesuai dengan aplikasi yang dibutuhkannya seperti pengambilan, peletakan barang dan sebagainya. Perkembangan dunia robotika juga banyak merambah ke berbagai bidang salah satunya dunia pendidikan terutama dalam pendidikan perguruan tinggi. Di perguruan tinggi khususnya di Politeknik manufaktur Negeri Bangka Belitung (Polman Babel), telah mempelajari tentang teknologi robotika seperti *cartesian, manipulator, degree of freedom (DOF)* dan sebagainya. Di Polman Babel sendiri telah dibuat teknologi robotika sebelumnya yaitu "*Prototype manipulator* dengan *2 degrees of freedom (DOF)* dan *gripper"*. Teknologi tersebut dibuat dengan tujuan untuk membuat suatu robot yang kontrolnya dapat bergerak pada posisi yang diinginkan sesuai instruksi koordinat sumbu x dan sumbu y yang diberikan dengan menggunakan *keypad* .

Namun robot tersebut belum dilengkapi dengan pengaturan letak titik koordinat posisi benda kerja yang dibawa dan diletakkan dan kecepatan putaran motor *stepper* masih perlu diatur lagi sehingga robot tersebut masih kurang efisien sebagai media pembelajaran. Maka dari itu, dibuatlah "Modifikasi Robot *Gripper Manipulato*r Dengan Motor *Stepper* Sebagai Media Pembelajaran" dengan fungsi yang lebih efisien yaitu tanpa perlu mengatur lagi kecepatan putaran motor *stepper* dan pengontrolan robot *gripper manipulator* dengan 2 cara, secara manual menggunakan *push button* untuk mengendalikan pergerakan motor *stepper* sumbu x dan sumbu y serta *gripper*, menginputkan letak titik koordinat posisi benda melalui *PC*/Laptop melalui *serial monitor arduino.* Pengontrolan

dengan cara tersebut ditujukkan agar mahasiswa dapat mengetahui berbagai macam kontrol yang digunakan dalam mengendalikan robot *manipulator*.

1.2 Rumusan Masalah

Rumusan masalah yang dibuat pada proyek akhir ini adalah :

1. Bagaimana cara membuat robot yang dapat mengambil, membawa dan meletakkan suatu barang.
2. Bagaimana membuat suatu kontrol robot yang dapat bergerak menuju posisi yang diinginkan.

1.3 Batasan Masalah

1. Manipulator robot hanya terdiri dari 2 *degrees of freedom* (DOF) yaitu X dan Y.
2. Ukuran benda kerja yaitu memiliki tinggi 8,7cm dengan diameter 4cm. Bahan yang digunakan yaitu PVC.
3. Hanya disediakan 4 titik lokasi untuk mengambil dan meletakan barang.
4. Pergerakan sumbu X maksimum sebesar 180^0.

1.4 Tujuan

Tujuan dari proyek akhir ini adalah :

1. Untuk membuat suatu robot dan kontrolnya yang dapat bergerak pada posisi yang diinginkan sesuai dengan instruksi yang diberikan.
2. Untuk membuat robot yang dapat berpindah posisi lengan, mengambil dan meletakkan barang sesuai dengan posisi titik benda kerja yang diinginkan.
3. Sebagai modul pembelajaran motor *stepper* di Politeknik Manufaktur Negeri Bangka Belitung.

BAB II
DASAR TEORI

2.1 Robot Manipulator

Robot lengan atau yang lebih dikenal dengan Manipulator Robot adalah salah satu jenis robot yang paling banyak digunakan oleh masyarakat industri. Robot manipulator industri ini sering disebut juga sebagai lengan robot, dengan link dan sendi. Sendi atau *Joint* adalah koneksi antar link yang dapat menentukan pergerakan, terdapat sendi *rotary* (atau *revolute*). Sementara sendi robot memiliki derajat lebih sedikit kebeban atau disebut derajat kebebasan atau *Degree of Freedom* (DOF), derajat kebebasan pada robot dapat diartikan sebagai jumlah gerakan independen yang dapat dibuat oleh suatu objek terhadap sistem koordinat yang menyebabkan perubahan posisi atau orientasi. Dalam menentukan derajat kebebasan yang dimiliki oleh sebuah robot, tidak dapat dilakukan hanya dengan menghitung jumlah persendian *(joint)* yang dimiliki oleh robot, karena tidak semua gerakan independen yang dibuat oleh persendian dapat dikategorikan sebagai derajat kebebasan. Terdapat enam gerakan independen yang dapat dibuat oleh suatu objek yang disebut sebagai derajat kebebasan yaitu tiga gerakan translasi T1, T2, T3 sepanjang aksis OX, OY dan OZ, dan tiga gerakan rotasional R1, R2, R3 pada aksis OX, OY dan OZ. Sedangkan link merupakan bagian-bagian kerangka yang kaku yang dihubungkan secara bersamaan sehingga membentuk suatu rangkaian kinematic [1].

Teknologi robot manipulator yang berbentuk hampir mirip dengan lengan manusia memiliki fungsi yang sama yaitu bisa membantu pekerjaan manusia seperti memindahkan dan meletakan barang. Namun walaupun demikian, gerakan lengan robot berbeda dari gerakan lengan manusia. Karena manipulator robot dapat bergerak bebas walaupun dari suatu sudut tertentu dengan kebebasan yang lebih dibandingkan manusia. Misalnya, siku dari robot dapat menekuk ke atas atau bawah sedangkan seseorang hanya bisa

menekuk siku mereka dalam satu arah selebihnya hanya pada posisi lengan lurus [1].

Secara umum robot manipulator terdiri dari Mekanikal tangan *(Mechanical Arm)* dimana merupakan pembentukan utama konstruksi pada lengan robot, dimana pembentukannya disesuaikan dengan kebutuhan dari lengan robot dan pengendali lengan robot tersebut. Dan *End Effector* yaitu suatu komponen dari lengan robot yang berfungsi mencengkram suatu objek tertentu untuk dipegang atau dipindahkan [1].

Gambar 2.1 Robot Manipulator [2]

Terdapat beberapa istilah dasar tentang robot diantaranya adalah [3]:

1. Robot: Merupakan peralatan yang dapat diprogram ulang, yang memiliki banyak sekali fungsi didesain untuk memindahkan material, suku cadang *(part)*, atau peralatan khusus lainnya.
2. Kontroler: Merupakan suatu peralatan yang berfungsi sebagai pengendali dari gerakan robot yang membentuk suatu sistem kontrol yang akan menentukan input dan output suatu robot.
3. Manipulator: Merupakan suatu lengan yang bisa bergerak memutar, melipat, ataupun menjangkau suatu objek. Gerakan ini disebut juga dengan gerakan

derajat kebebasan robot atau jumlah sumbu yang dimiliki robot. Manipulator sendiri mempunyai beberapa segmen dan sambungan *(joint)* .

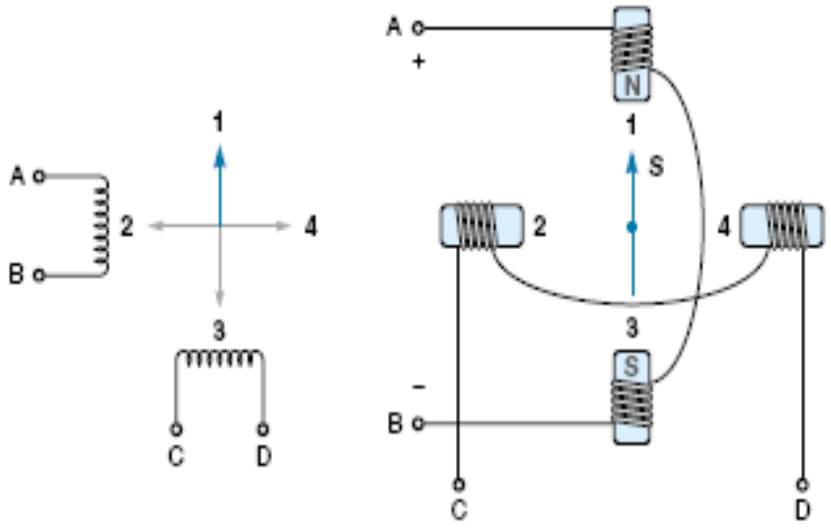
4. *Joint*: Merupakan sambungan yang menghubungkan antara lengan *(arm)* dan lengan lainnya sehingga dipisahkan oleh sumbu *(axis).*
5. *Gripper:* Merupakan sebuah efektor yang berfungsi untuk menggenggam dan menahan suatu objek. Objek yang bisa dipindahkan oleh suatu robot dapat berupa kertas, botol, bahan mentah dan peralatan-peralatan lainnya.

2.2 Motor *Stepper*

Motor *stepper* adalah salah satu jenis motor yang dikendalikan dengan pulsa-pulsa digital. Prinsip kerja motor *stepper* adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit dimana motor *stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor *stepper* tersebut. Keunggulan dari motor *stepper* adalah gerakannya yang dapat dikendalikan dengan mudah dan tingkat kepresisiannya yang tinggi, karena keunggulannya motor ini biasa dipakai sebagai penggerak pada *flopy disk driver* atau printer. Untuk menggerakkan motor *stepper* tidak sama dengan menggerakkan motor DC. Namun memerlukan sebuah rangkaian pengontrol dengan menggunakan prinsip kerja pulsa [4].

Setiap kali mengirim pulsa ke pengontrol elektronik, maka motor akan bergerak ‘selangkah’, yaitu satu putaran sudut kecil. Motor *stepper* dapat berputar atau berotasi dengan sudut step yang bisa bervariasi tergantung motor yang digunakan. Ukuran step (step size) dapat berada pada range 0,90 sampai 900. Misalnya sudut step 7,50°, 15°, 30° dan seterusnya tergantung aplikasi atau kebutuhan yang diinginkan. Posisi putarannya pun relatif eksak dan stabil. Dengan adanya variasi sudut step tersebut akan lebih memudahkan untuk menggunakan sinyal digital tanpa perlu menggunakan rangkaian *closed-loop feedback* untuk memonitor posisinya. Dengan alasan inilah maka motor *stepper* banyak digunakan sebagai actuator yang menerapkan rangkaian digital sebagai interfacing kepiranti yang berbasis mikroprosesor/mikrokontroler [4].

Ada dua jenis motor *stepper* yang banyak dipakai, yakni *bipolar stepper motor* dan *unipolar* stepper motor. *Bipolar stepper motor* bekerja dengan menggunakan satu lilitan penuh pada setiap bagian *stator* untuk melakukan gerak satu langkah. Ciri-ciri dari *bipolar stepper motor*, adalah terdapat empat kabel sebagai media transfer arus [4].

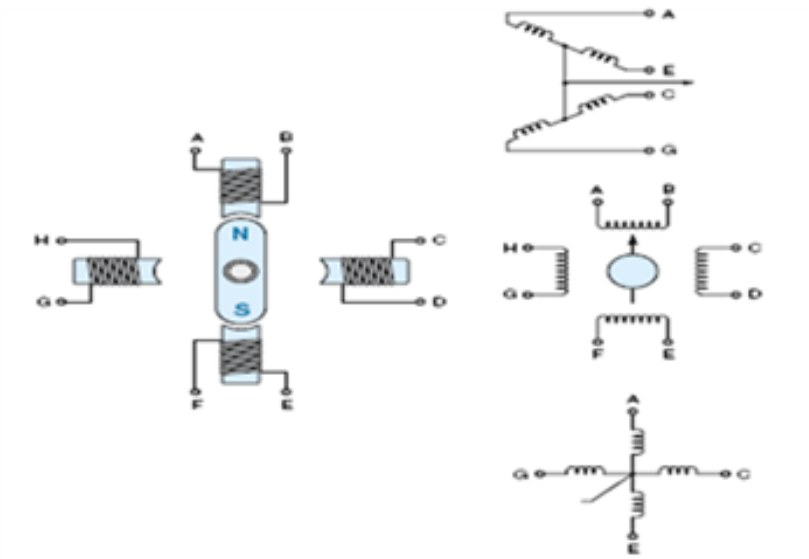


(a). Symbol (b). Wiring diagram

Gambar 2.2 Motor *stepper* dua-phase (bipolar) [4]

Penggunaan motor *stepper* jenis *bipolar* memerlukan rangkaian yang agak lebih rumit untuk mengatur agar motor ini dapat berputar dalam dua arah. Untuk menggerakkan motor *stepper* jenis ini biasanya diperlukan sebuah *driver motor* yang dikenal dengan nama H *bridge.* Rangkaian ini akan mengontrol setiap kumparan secara terpisah (*independent*) termasuk polaritas untuk setiap kumparan [2].

Sedangkan *unipolar stepper motor*, bekerja dengan menggunakan setengah dari lilitan pada setiap statornya. Sehingga kabel yang disediakan pada *unipolar stepper motor* biasanya lebih dari empat. Dimana terdapat sebuah *common*, yang dapat diberikan ke ground atau VCC [2].



(a). Wiring diagram (b). Symbol

Gambar 2.3 Motor *stepper* empat-*phase (unipolar)* [4]

BAB III
METODE PELAKSANAAN

3.1 Parameter-Parameter Penting

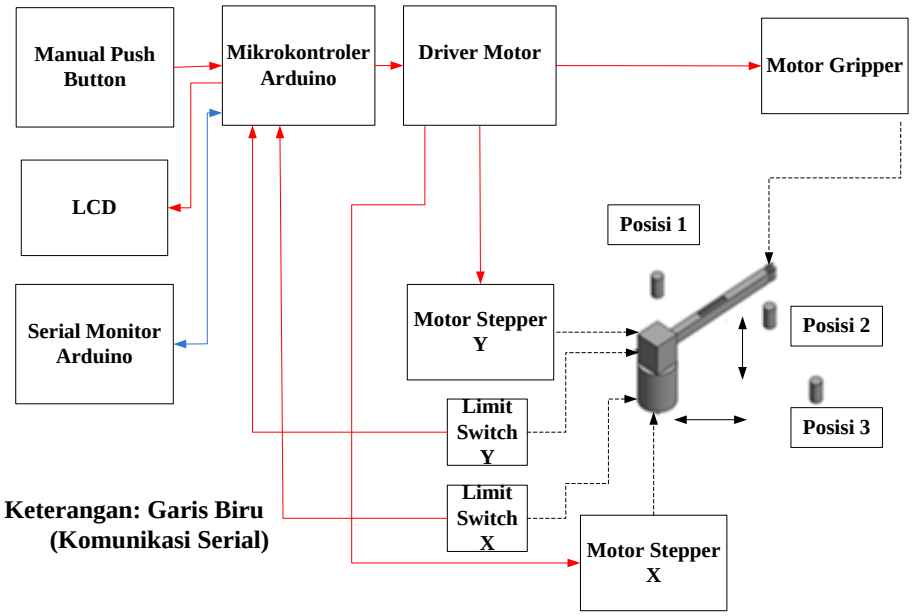
Parameter-parameter yang paling penting untuk dilakukan dalam menyelesaikan proyek akhir ini adalah sebagai berikut :

- Robot bisa bergerak untuk mengambil maupun meletakan barang.
- Robot bisa digerakan secara manual.
- Robot dapat bergerak secara otomatis.

Sehingga untuk menyelesaikan parameter yang disebutkan diatas maka desain keseluruhan dari proyek akhir ini dapat digambarkan dalam blok diagram sebagai pada Gambar 3.1.

Blok diagram keseluruhan terdiri dari mikrokontroler, lengan manipulator dan gripper. Mikrokontroler yang digunakan adalah jenis Arduino yang berfungsi sebagai pengontrol dari program push button dan melalui serial monitor. Lengan manipulator dan gripper digunakan sebagai keluaran. Lengan manipulator digerakan oleh motor stepper pada sumbu x dan sumbu y, sedangkan untuk gripper (cekam) menggunakan motor dc sebagai penggeraknya dan untuk tampilan hasil program digunakan LCD.

Prinsip kerja alat ini adalah lengan manipulator akan bergerak berdasarkan inputan dari push button untuk mengendalikan pergerakan motor stepper sumbu x dan sumbu y serta gripper, menginputkan letak titik koordinat posisi benda melalui PC/Laptop melalui serial monitor arduino.



Gambar 3.1 Blok Diagram Sistem Secara Keseluruhan

3.2 Alat-Alat dan Komponen

Adapun alat-alat atau komponen yang digunakan diantaranya:

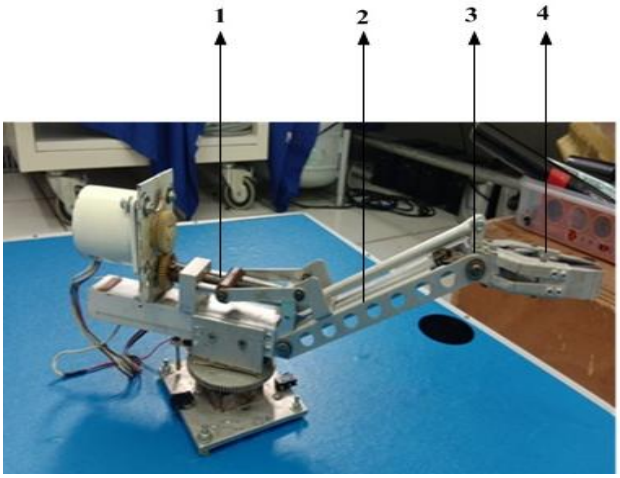
Tabel 3.1 Alat-alat dan Komponen

No	Nama Komponen	Spesifikasi
1	Motor stepper X dan motor stepper Y	a. 200 steps/rev atau 1,80/step b. Tegangan kerja 12 Vdc
2	Motor dc	a. Tegangan kerja 12 Vdc
3	Driver motor L298N	a. Tegangan minimal masukan power 5V-35V b. Arus untuk masukan antara 0-36mA c. Daya maksimal yaitu 25W
4	LCD	a. Data I/O pins 7-14 b. Vcc 15 c. Ground 16
5	Arduino Mega	a. Tegangan Input(rekomendasi) 7-12V b. Tegangan Input(limit) 6-20V c. Arus DC per Pin I/O 20 mA d. Arus DC untuk pin 3.3 V 50 mA e. Pin Digital I/O 54(of which 15 provide PWM output) f. Pin Analog Input 16

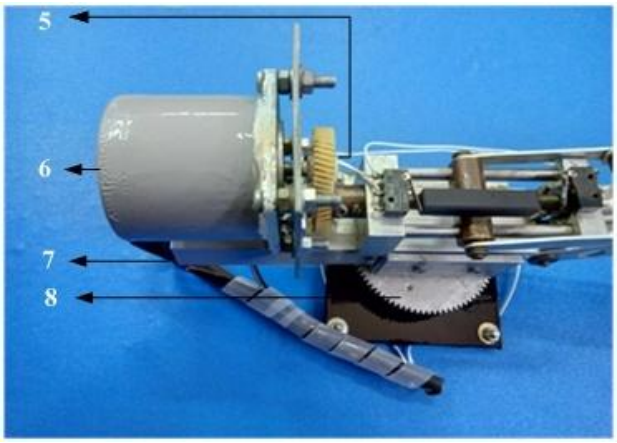
BAB IV
PEMBAHASAN

4.1 Manipulator Robot

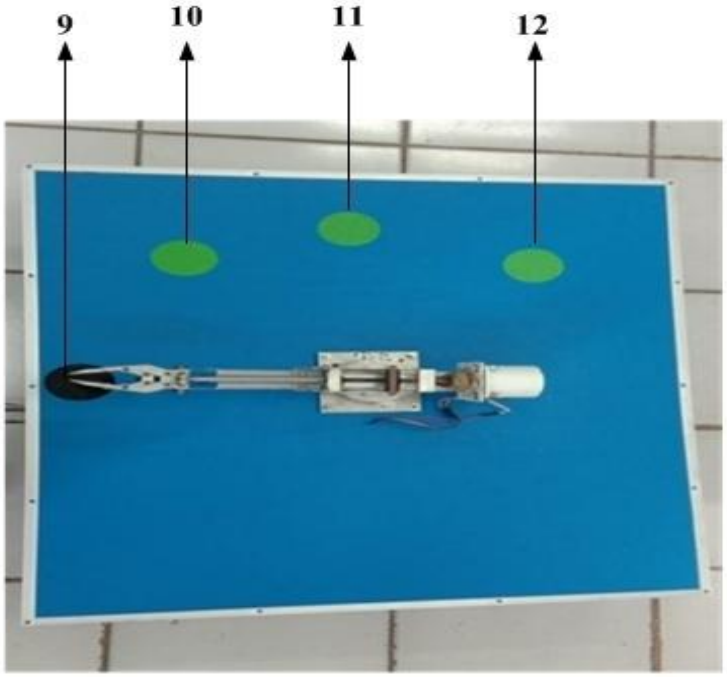
Manipulator dirancang memilik 2 bagian, yaitu lengan robot dan gripper. Lengan robot terdiri dari pergerakan sumbu x dan pergerakan sumbu y. Pergerakan lengan robot ini terdiri dari gerak angkat pada sumbu y dan gerak putar pada sumbu x.



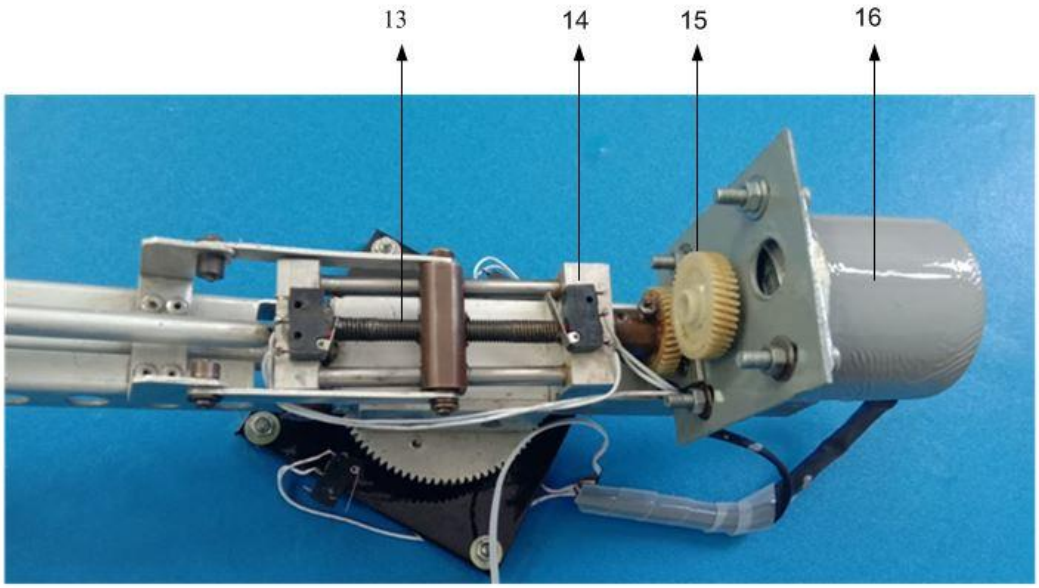
Gambar 4.1 Lengan Manipulator



Gambar 4.2 Tampak Samping lengan Manipulator



Gambar 4.3 Tampak Atas Robot Manipulator



Gambar 4.4 Bagian Robot Manipulator

Keterangan

1. Ulir Motor Y
2. Poros
3. *Joint*
4. *Gripper*
5. Roda Gigi
6. Motor Stepper Sumbu Y
7. Motor Dc *Gripper*
8. Motor Stepper Sumbu X
9. Posisi Reset
10. Posisi 1
11. Posisi 2
12. Posisi 3
13. Ulir
14. Limit Switch
15. Roda gigi
16. Motor Stepper

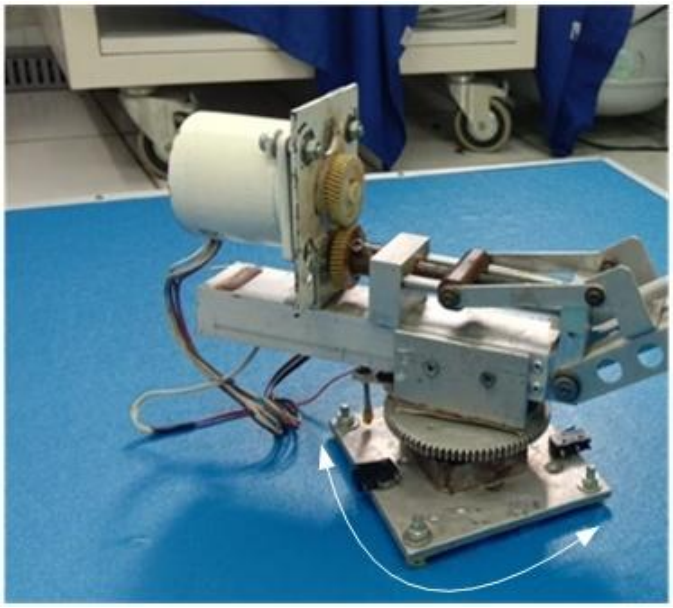
4.2 Sumbu X

Pada pergerakan sumbu x menggunakan sistem putar, pergerakan sumbu x diletakkan sejajar dengan meja kerja dan dapat berputar kurang lebih 180º, dimana poros motor stepper langsung terhubung dengan robot. Untuk menghindari tekanan beban langsung pada motor stepper maka diberi bearing pada sambungan poros motor dan lengan. Untuk mengetahui letak minimum dan maksimum sumbu x maka diberi limit switch sebagai sensor. Motor stepper yang digunakan untuk menggerakkan sumbu x ini mempunyai 200 step per putaran, sehingga setiap step motor stepper akan menggerakkan lengan sebesar 1,8º.

Berdasarkan hasil pengukuran driver motor sumbu X didapat tegangan sebesar 12V.

Tabel 4.1 Hasil Percobaan Sumbu X

Sumbu X	
Posisi	Titik Absolut
Start-1	8
Start-2	14
Start-3	21
1-2	6
1-3	13
2-3	7

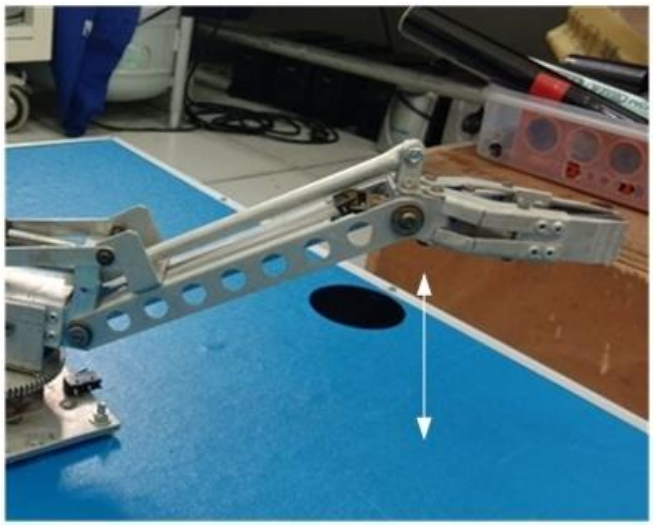


Gambar 4.5 Pergerakan Sumbu X Pada Lengan Manipulator

4.3 Sumbu Y

Untuk sumbu y menggunakan sistem angkat yang dihubungkan dengan gear dan ulir yang dihubungkan dengan motor stepper. Sebagai penghubung antara *elbow* dan *pitch* menggunakan poros. Apabila motor berputar maka efektor naik atau turun tergantung dengan arah putaran motor.

Pitch/kisar ulir yang digunakan menggunakan baut M8 yaitu 1,25 mm. Panjang baut yang digunakan 36 mm. Roda gigi yang digunakan terpasang pada motor memiliki jumlah gigi 45 dan sedangkan pada ulir memiliki jumlah gigi sebanyak 38.



Gambar 4.6 Pergerakan Sumbu Y Pada Lengan Manipulator

Perhitungan pergerakan motor sumbu y

1 kisar ullir = 1,25mm (4.1)

1 kisar ulir $= \frac{38}{45}$ putaran motor (4.2)

1 kisar ulir $= \frac{38}{45} \times 200$ step = 168 step (4.3)

1,25mm =168 step (4.4)

1 mm $= \frac{168 \text{ step}}{1{,}25 \text{ mm}}$ = 134,4 step (4.5)

= 135 step/ mm (4.6)

Jadi untuk pergerakan 1 mm ulir dibutuhkan 135 step motor y.

Untuk percobaan sumbu Y, panjang lintasan ulir yang dibutuhkan untuk menurunkan lengan manipulator ke benda kerja dan menaikkannya sebesar 11mm

atau dalam jumlah steps yang dibutuhkan sebesar 1485 *steps.* Sesuai dengan perhitungan diatas untuk menempuh jarak sebesar 11mm maka perhitungannya adalah :

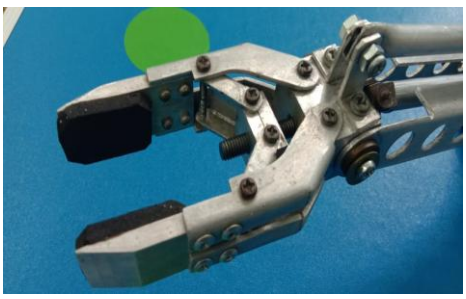
11mm x 135 step = 1485 *step.* (4.7)

Berdasarkan hasil pengukuran driver motor sumbu Y didapat tegangan sebesar 12V.

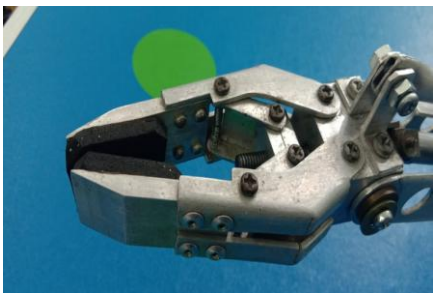
4.4 Gripper (cekam)

Pada bagian pencekam menggunakan sistem ulir yang disambung dengan poros engsel kardan. Poros engsel kardan ini memiliki kemampuan untuk berputar pada kondisi poros yang miring, sedangkan kemampuan gerakan memanjang harus dicapai melalui hubungan antar poros. Hubungan antar poros ini dapat menggeser ulir gripper dengan transmisi momen putar sehingga gripper dapat melakukan gerakan membuka dan menutup.

Mekanikal Gripper didesain untuk menggenggam dan menahan objek dengan memberikan inputan pada motor dc tersebut. Diameter objek benda yang dapat dicekam kurang lebih 30mm. Mekanikal gripper biasanya menggunakan *finger/*jari mekanik yang disebut dengan jaws. Sumber tenaga yang diberikan pada gripper ini bias elektrik.



(a). Gripper open



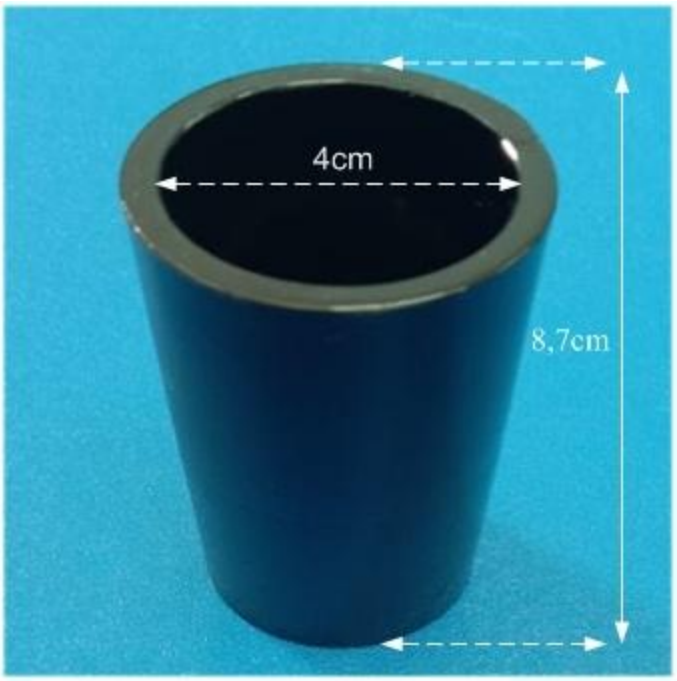
(b). Gripper Close

Gambar 4.7 Gripper Pada Lengan Manipulator

Berdasarkan hasil pengukuran driver motor sumbu X didapat tegangan sebesar 12V.

4.5 Benda Kerja

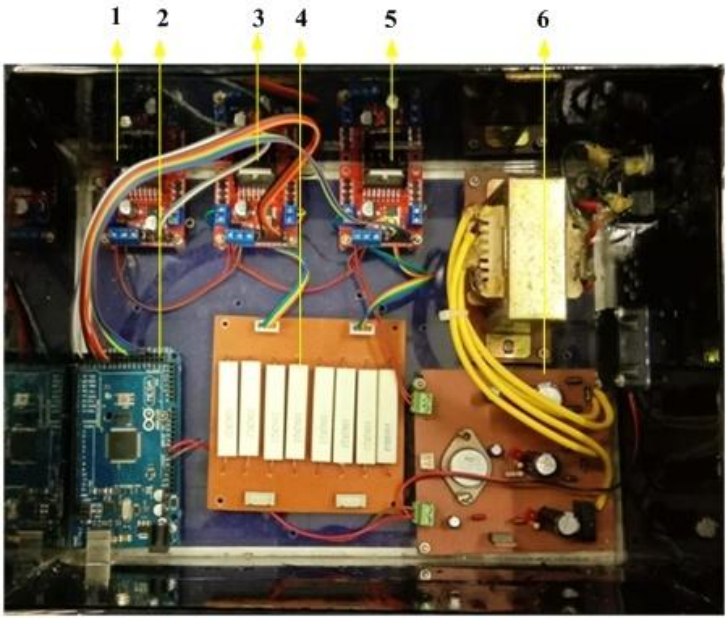
Benda kerja yang digunakan berbahan dasar PVC, bahan dasar PVC dipilih karena memiliki berat yang ringan sehingga memudahkan lengan robot untuk bisa mengangkat benda kerja tersebut. Ukuran benda kerja yaitu memiliki tinggi 8,7cm dengan diameter 4cm.



Gambar 4.8 Benda Kerja

4.6 Desain Layout Kontrol Manual Robot Manipulator

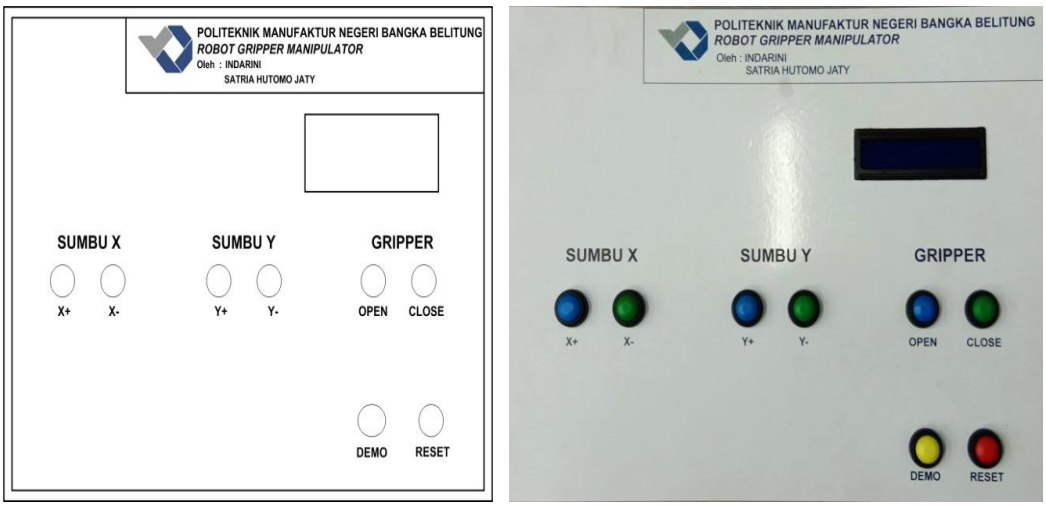
Untuk membuat suatu kontrol robot manipulator dibutuhkan beberapa komponen yang diperlukan seperti driver Motor L298N, Arduino Mega, Resistor, Power Supply, dan LCD.



Gambar 4.9 Kontrol Robot Manipulator

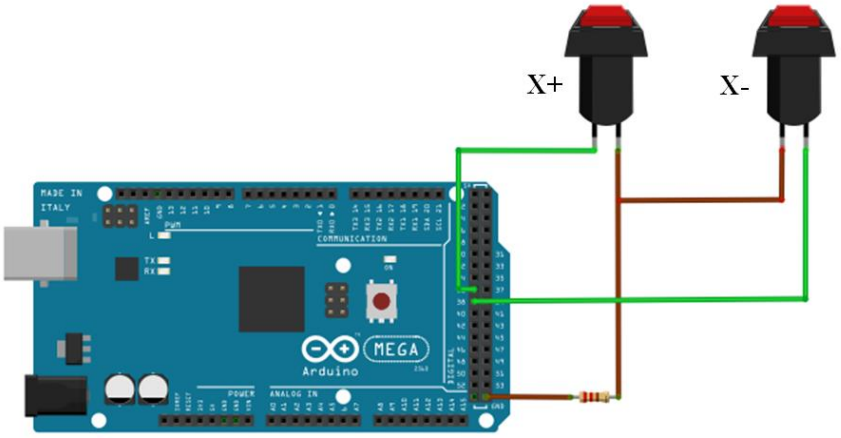
Keterangan

1. Motor dc untuk gripper
2. Arduino Mega
3. Motor Sumbu Y
4. Resistor untuk kebeban motor
5. Motor Sumbu X
6. Power Supply



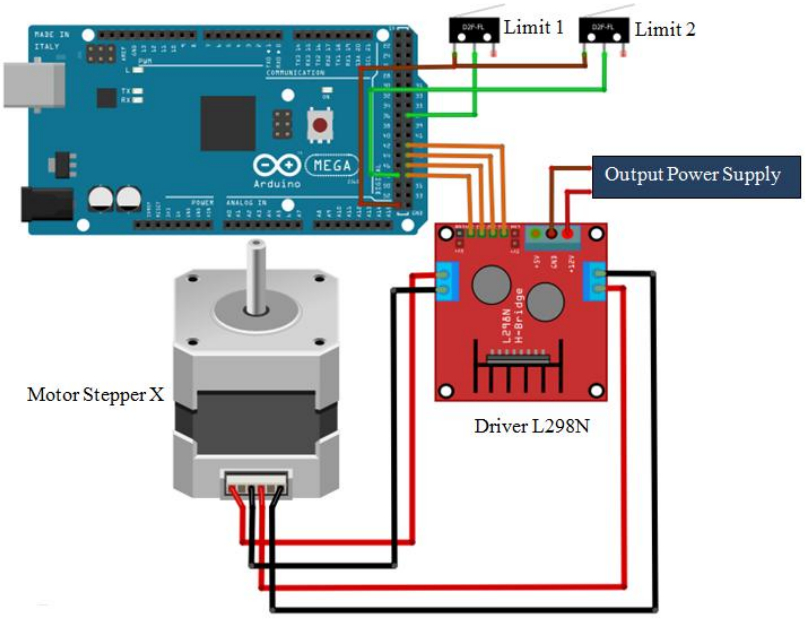
(a). Desain Kontrol Manual Push Button (b). Kontrol Manual Push Button

Gambar 4.10 Kontrol Robot Manipulator Manual Push Button



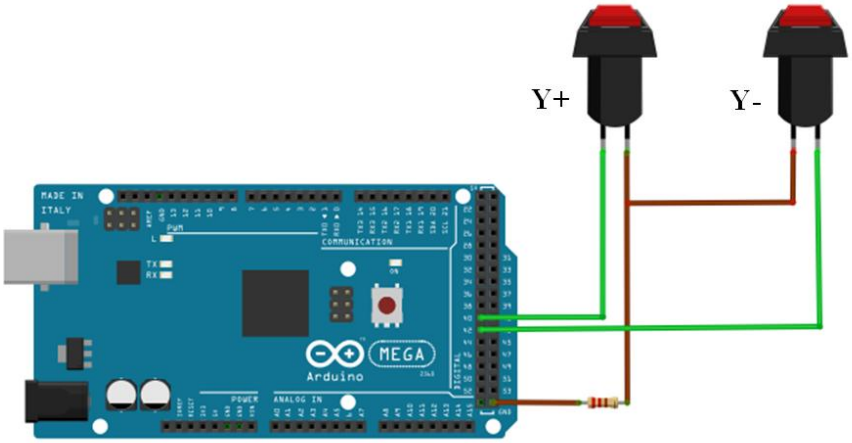
Gambar 4.11 Skematik Kontrol Sumbu X

Pin yang digunakan pada tombol X+ adalah pin 36 sedangkan pada tombol X- adalah pin 38 dan pin untuk resistor menggunakan pin ground arduino.



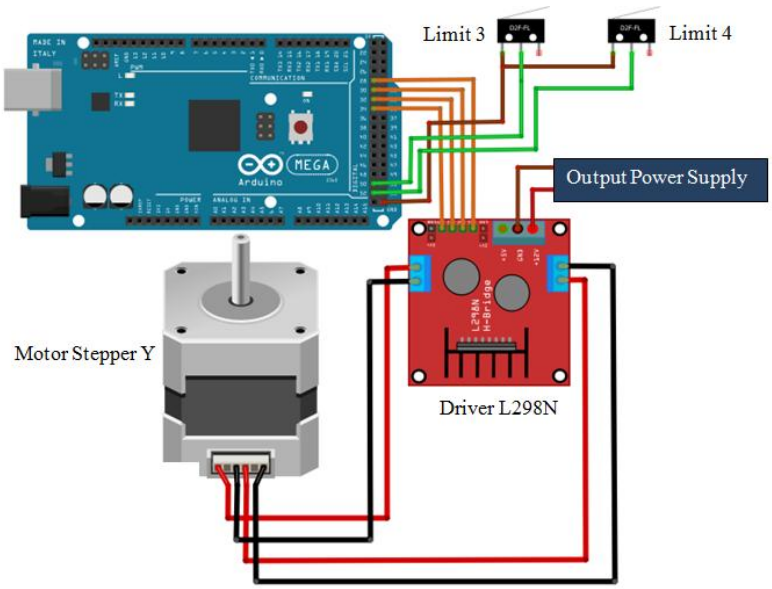
Gambar 4.12 Skematik Driver Motor Sumbu X

Pin yang digunakan adalah IN 1 : 43 , IN2 : 45 , IN3 : 47 , IN4 : 49 Limit switch 1 menggunakan pin 37 sedangkan Limit switch 2 menggunakan pin 48.



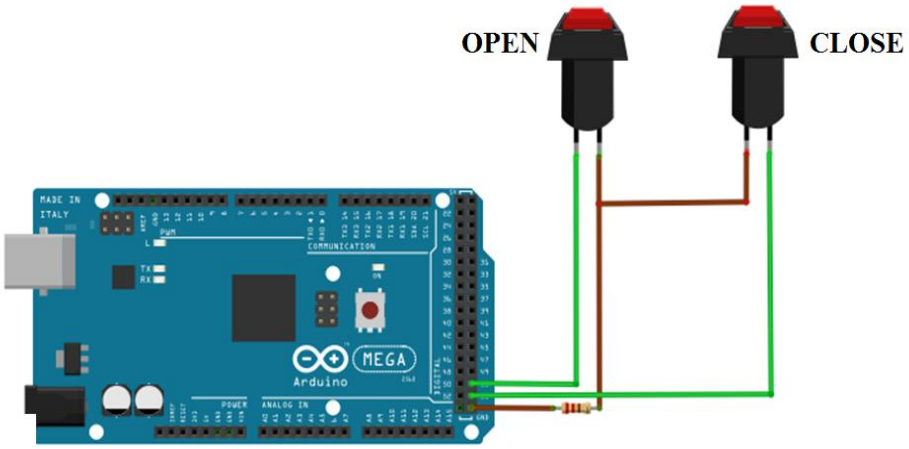
Gambar 4.13 Skematik Kontrol Sumbu Y

Pin yang digunakan pada tombol Y+ adalah pin 40 sedangkan pada tombol Y- adalah pin 42.



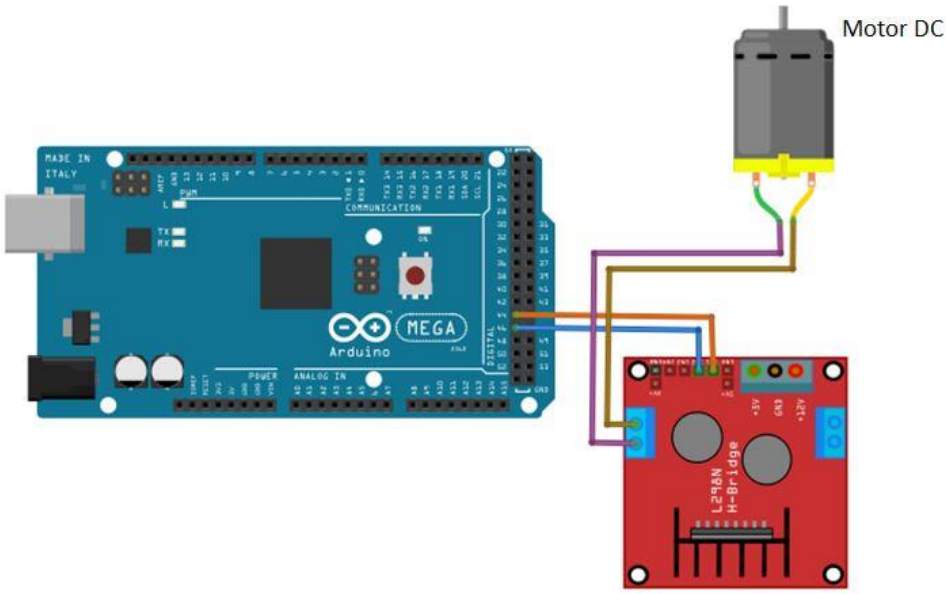
Gambar 4.14 Skematik Driver Motor Sumbu Y

Pin yang digunakan adalah IN 1 : 28 , IN2 : 30 , IN3 : 32 , IN4 : 34 Limit switch 3 menggunakan pin 50 sedangkan Limit switch 4 menggunakan pin 52.



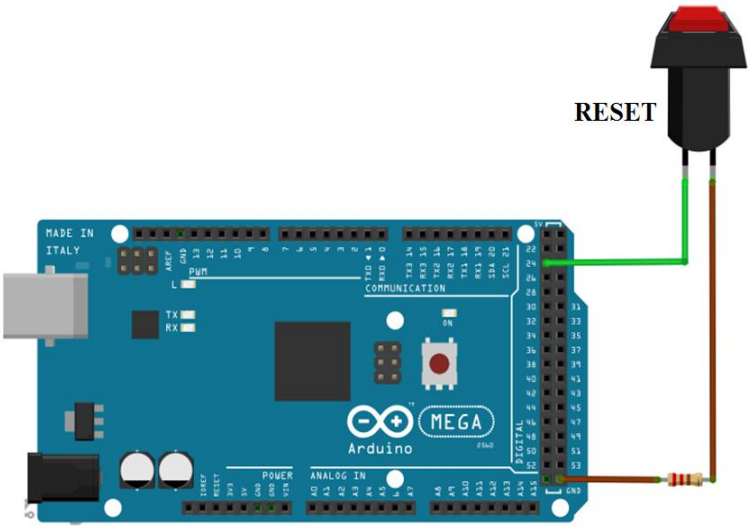
Gambar 4.15 Skematik Kontrol Gripper

Pin yang digunakan pada Gripper open adalah pin 51 sedangkan pada Gripper close adalah pin 53.



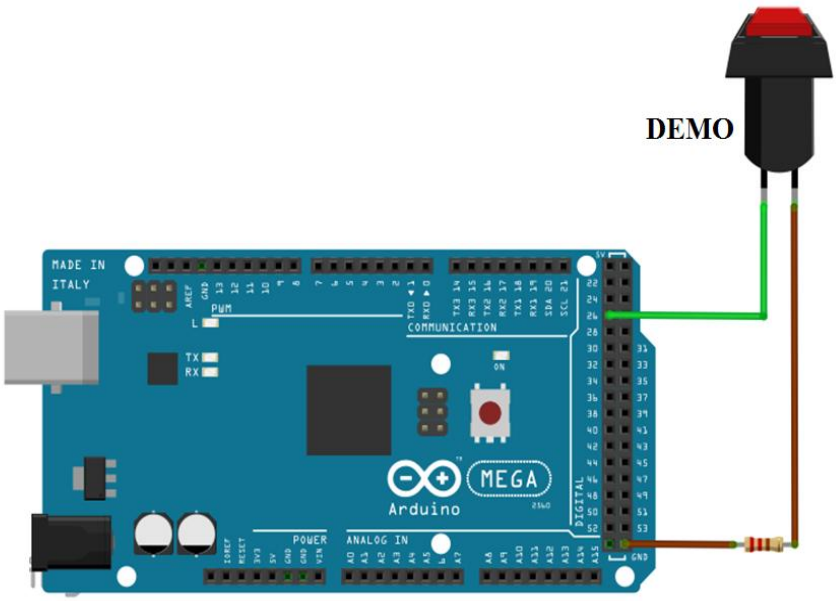
Gambar 4.16 Skematik Driver Motor Dc Gripper

Pin yang digunakan pada IN 1 adalah pin 44, sedangkan pada IN 2 menggunakan pin 46.



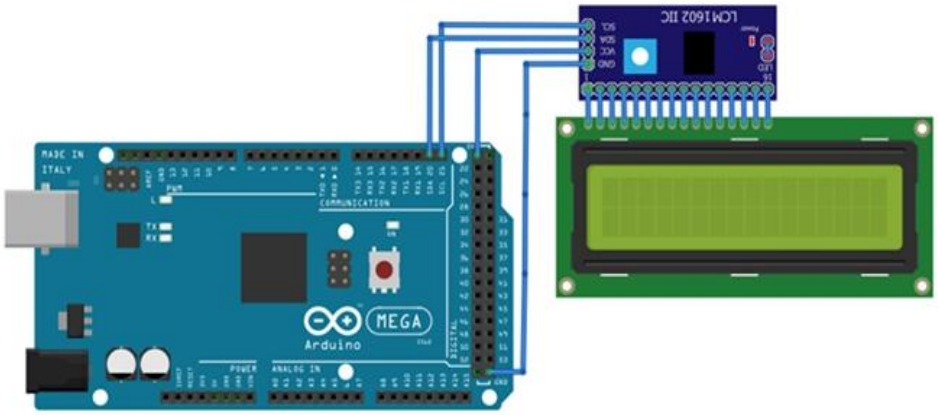
Gambar 4.17 Skematik Kontrol Reset

Pin yang digunakan pada Reset adalah pin 24.



Gambar 4.18 Skematik Kontrol Demo

Pin yang digunakan untuk Demo adalah pin 46.



Gambar 4.19 Skematik LCD

Pin yang digunakan adalah pin VCC, pin ground, pin 20/SDA dan 21/SCL.

4.7 Power Supply

Power supply atau catu daya adalah komponen yang dapat menyediakan dan mendistribusikan tegangan pada rangkaian elektronika seperti sumber catu daya dan baterai.

Power supply yang dipakai untuk proyek akhir ini adalah:

1. Power supply 5VDC, digunakan untuk supply rangkaian *Pull Up* Eksternal.
2. Power supply 9VDC, digunakan untuk supply sistem minimum Supply sistem minimum menggunakan 9VDC karena pada rangkaian sistem minimum terdapat IC reguler 5 V.
3. Power supply 15VDC, digunakan untuk *supply driver* motor stepper dan motor dc.

Pengujian rangkaian power supply

Secara perhitungan

1. Untuk power supply 15 VDC :

   $Vrms = ½ \sqrt{2}\ Vp$ (4.8)

   $15 = ½ \sqrt{2}\ Vp$

Vp = Vm = 21,216

Vdc = 0,636 × Vm = 0,636 × 21,216 = 13,49 Vdc. (4.9)

2. Untuk power supply 9 VDC :

Vrms = ½ √2 Vp (4.10)

9 = ½ √2 Vp

Vp = Vm = 12,7296

Vdc = 0,636 × Vm = 0,636 × 12,7296 = 8,096 Vdc. (4.11)

3. Untuk power supply 5 VDC :

Vrms = ½ √2 Vp (4.12)

5 = ½ √2 Vp

Vp = Vm = 7,0721

Vdc = 0,636 × Vm = 0,636 × 7,0721 = 4,497 Vdc. (4.13)

Karena Vdc lebih besar dari tegangan IC 7805 sebesar 5 Volt, maka tegangan yang keluar adalah 5 volt.

Tabel 4.2 Hasil Pengujian *Power Supply*

NO	Hasil Perhitungan	Hasil Pengukuran	% error
1	4,497 V	4,8 V	6,7%
2	8,096 V	8,57 V	5,80%
3	13,49 V	14,29 V	5,90%

Persentase error (% error) dari pengukuran rangkaian catu daya diatas adalah :

$$\%\text{error} = \frac{pengukuran\ \ -sebenarnya}{sebenarnya} \times 100\% \quad (4.14)$$

Untuk tegangan 5 volt:

$$\%\text{error} = \frac{4{,}8volt - 4{,}497volt}{4{,}497volt} \times 100\%$$

$$= 6{,}7\%$$

Untuk tegangan 9 volt:

$$\%\text{error} = \frac{8{,}57volt - 8{,}096volt}{8{,}096volt} \times 100\%$$

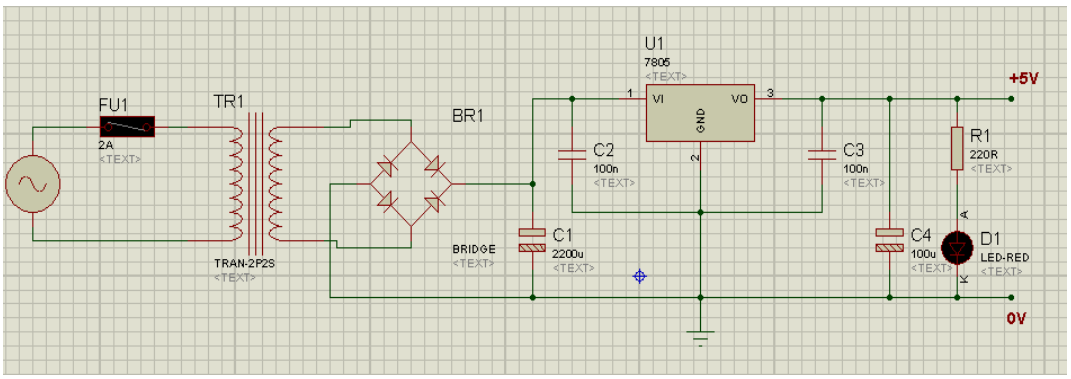
$$= 5{,}8\%$$

Untuk tegangan 15 volt:

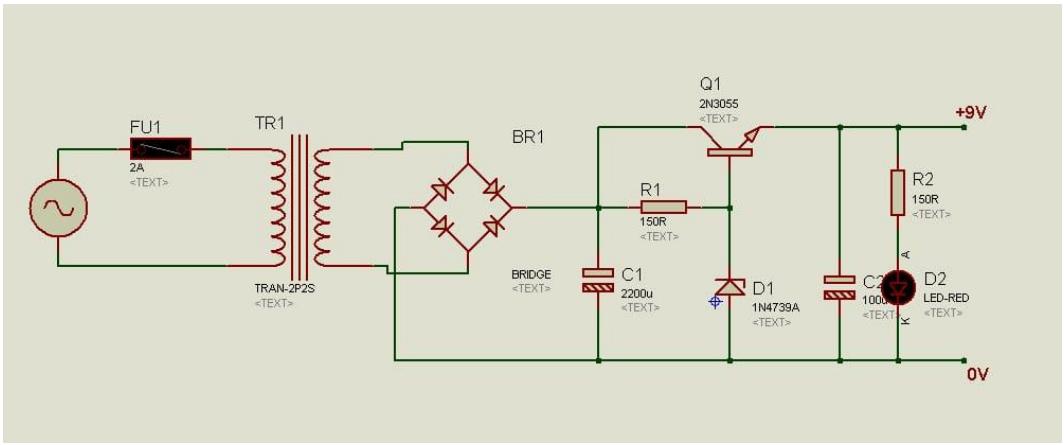
$$\%\text{error} = \frac{14{,}29volt - 13{,}49volt}{13{,}49volt} \times 100\%$$

$$= 5{,}9\%$$

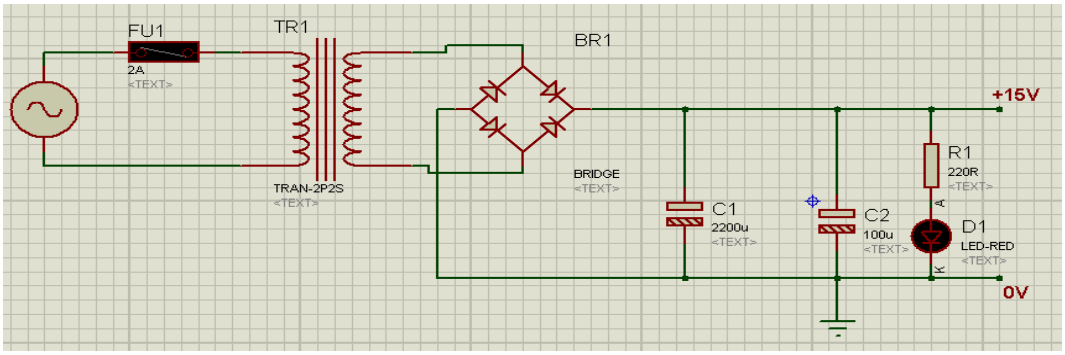
Dari perhitungan % error diatas, kesalahan tegangan output untuk +5volt sebesar 6,7% sedangkan untuk +9v adalah 5,8% dan untuk tegangan +15volt sebesar 5,9%. Hal ini masih dapat diabaikan, kerena error tidak begitu besar sehingga tegangan yang dijadikan sebagai input mikrokontroler, rangkaian *pull up* serta rangkaian driver ini masih dapat digunakan dan bisa mengaktifkan rangkaian serta tidak menyebabkan kerusakan pada komponen.



Gambar 4.20 Skematik Power Supply 5V



Gambar 4.21 Skematik Power Supply 9V



Gambar 4.22 Skematik Power Supply 15V

4.8 Program

Ada beberapa program yang disediakan diantaranya adalah program manual dengan menggunakan push button, dan program dengan serial monitor.

4.8.1 Stepper CW (clock wise) / CCW (counter clock wise)

Program stepper cw/ccw berisi perintah untuk menggerakkan motor stepper sesuai dengan keinginan pengguna. Motor stepper tersebut dapat bergerak searah jarum jam (*clock wise*) atau bergerak berlawanan arah jarum jam (*counter clock wise*). Berikut adalah *list* programnya:

```
void motory_cw(){
  digitalWrite(28,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);                    } Kumparan 1 (P1) motor stepper sumbu y aktif
  digitalWrite(28,LOW);
  digitalWrite(30,HIGH);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);                    } Kumparan 2 (P2) motor stepper sumbu y aktif
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,HIGH);
  digitalWrite(34,LOW);
  delay(Ty);                    } Kumparan 3 (P3) motor stepper sumbu y aktif
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,HIGH);
  delay(Ty);                    } Kumparan 4 (P4) motor stepper sumbu y aktif
```

```
}
void motory_ccw() {
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,HIGH);      Kumparan 4 (P4) motor stepper sumbu y aktif
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,HIGH);      Kumparan 3 (P3) motor stepper sumbu y aktif
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,HIGH);
  digitalWrite(32,LOW);       Kumparan 2 (P2) motor stepper sumbu y aktif
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);       Kumparan 1 (P1) motor stepper sumbu y aktif
  digitalWrite(34,LOW);
  delay(Ty);
}
```

4.8.2 Gripper Open dan Close

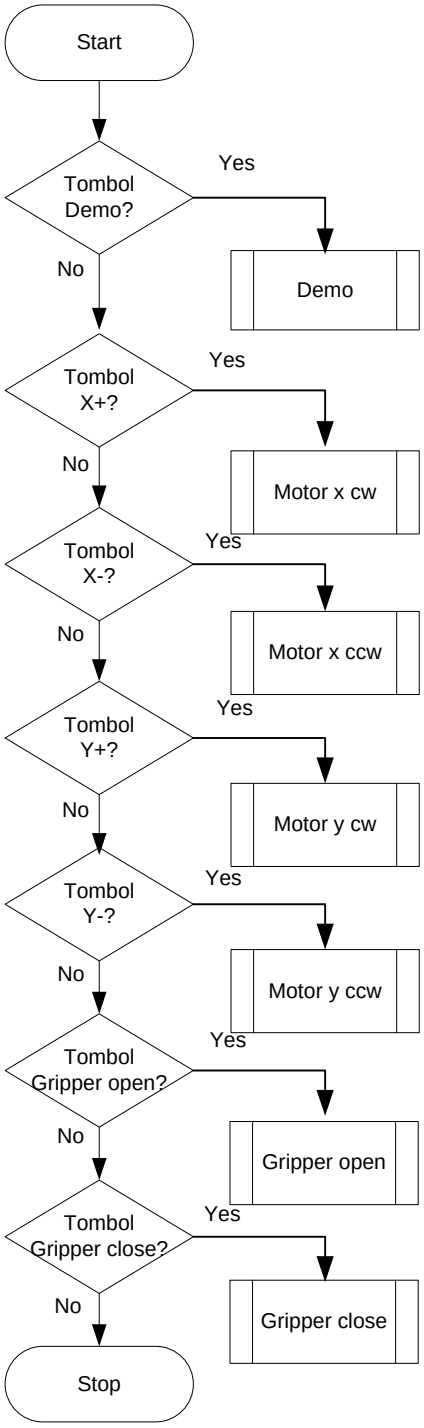
Terdapat 2 pilihan yaitu gripper open dan griper close, program pada gripper open berisi perintah untuk membuat gripper pada robot bisa membuka dan mencengkam benda kerja, sedangkan gripper close adalah program yang berisi perintah untuk membuat gripper pada robot bisa menutup atau dengan kata lain bisa melepaskan benda kerja yang telah disediakan.

```
int tombol_gripbuka=digitalRead(gripbuka);
int tombol_griptutup=digitalRead(griptutup);   Konfigurasi tombol gripper buka dan tutup
if (tombol_gripbuka==LOW && tombol_griptutup==HIGH)
{
  digitalWrite(23,LOW);
  digitalWrite(25,HIGH);      Maka motor dc bergerak berlawanan arah jarum jam sekaligus
}                             membuka gripper
else if (tombol_gripbuka==HIGH && tombol_griptutup==LOW)
{
  digitalWrite(23,HIGH);
  digitalWrite(25,LOW);       Maka motor dc bergerak searah jarum jam sekaligus menutup
}                             gripper
else
{
 digitalWrite(23,LOW);
  digitalWrite(25,LOW);       Jika tidak memenuhi kondisi di atas maka motor dc akan
}                             berhenti
```

Keterangan program

If (tombol_gripbuka==LOW&&tombol_griptutup==HIGH) adalah jika tombol gripper buka berlogika LOW dan tombol gripper tutup berlogika HIGH. Sedangkan *else if (tombol_gripbuka==HIGH&&tombol_griptutup==LOW)* adalah jika tidak memenuhi kondisi dan persyaratan diatas maka tombol gripper buka berlogika HIGH dan tombol gripper tutup berlogika LOW.

4.8.3 Flowchart Program Manual



Gambar 4.23 Flowchart Program Manual

Flowchart program manual menggunakan push button, terdapat beberapa tombol yang digunakan yaitu tombol demo yang digunakan untuk menjalankan

robot secara otomatis, tombol reset digunakan untuk mengembalikan kondisi robot keposisi awal atau posisi 0 titik benda kerja, tombol sumbu x+ dan tombol sumbu x- yang digunakan untuk menjalankan robot bergerak ke kiri dan ke kanan, tombol sumbu y+ dan tombol sumbu y- digunakan untuk menggerakkan robot ke atas dan ke bawah, sedangkan tombol gripper open dan gripper close digunakan untuk mencengkram dan melepaskan benda kerja.

```
////==============================MANUAL SUMBU X
int Tombolx_kanan = digitalRead(tombolx1);   ┐  Konfigurasi tombol x+ dan tombol x-
 int Tombolx_kiri = digitalRead(tombolx2);   ┘
 int limit2= digitalRead(lm2);               ┐  Konfigurasi Limit Switch sumbu x maksimum dan minimum
 int limit1=digitalRead(lm1);                ┘
 if(limit1==HIGH&&Tombolx_kanan==LOW&&Tombolx_kiri==HIGH )
{
        motorx_cw();                         ┐
        lcd.clear();                         │  Maka motor stepper sumbu x akan
        lcd.setCursor(9,0);                  │  bergerak searah jarum jam
        lcd.print(x);                        ┘
}
 elseif(limit2==HIGH&&Tombolx_kanan==HIGH&&Tombolx_kiri==LOW)
 {
        motorx_ccw();                        ┐
         lcd.clear();                        │  Maka motor stepper sumbu x akan
         lcd.setCursor(9,0);                 │  bergerak berlawanan arah jarum
         lcd.print(x);                       ┘  jam
 }
 else                                        ┐
 {                                           │  Jika tidak memenuhi kondisi di atas
        stopx();                             │  maka motor stepper sumbu x akan
 }                                           ┘  berhenti
//==============================MANUAL SUMBU Y
int Tomboly_naik=digitalRead(tomboly1);      ┐  Konfigurasi tombol y+ dan tombol
int Tomboly_turun=digitalRead(tomboly2);     ┘  y-
int limit4=digitalRead(lm4);                 ┐  Konfigurasi Limit Switch sumbu y maksimum dan minimum
 int limit3= digitalRead(lm3);               ┘
if(limit3==HIGH&&Tomboly_naik==LOW&&Tomboly_turun==HIGH )
{
         motory_cw();                        ┐
        lcd.clear();                         │  Maka motor stepper sumbu y akan
        lcd.setCursor(9,1);                  │  bergerak searah jarum jam
         lcd.print(y);                       ┘
}
 elseif(limit4==HIGH&&Tomboly_naik==HIGH&&Tomboly_turun==LOW)
 {
        motory_ccw();                        ┐
        lcd.clear();                         │  Maka motor stepper sumbu y akan
        lcd.setCursor(9,1);                  │  bergerak berlawanan arah jarum
        lcd.print(y);                        ┘
 }
 else                                        ┐
 {                                           │  Jika tidak memenuhi kondisi di atas
                                             │  maka motor stepper sumbu y akan
                                             ┘  berhenti
```

```
        stopy();
 }
//============================GRIPPER
int tombol_gripbuka=digitalRead(gripbuka);
int tombol_griptutup=digitalRead(griptutup);
```
Konfigurasi tombol gripper buka dan gripper tutup

```
if (tombol_gripbuka==LOW && tombol_griptutup==HIGH)
{
 digitalWrite(22,LOW);
 digitalWrite(24,HIGH);
```
Maka motor dc bergerak berlawanan arah jarum jam sekaligus membuka gripper

```
}
else if (tombol_gripbuka==HIGH && tombol_griptutup==LOW)
{
 digitalWrite(22,HIGH);
 digitalWrite(24,LOW);
```
Maka motor dc bergerak searah jarum jam sekaligus menutup gripper

```
}
else
{
 digitalWrite(22,LOW);
 digitalWrite(24,LOW);
```
Jika tidak memenuhi kondisi di atas maka motor dc akan berhenti

```
}
```

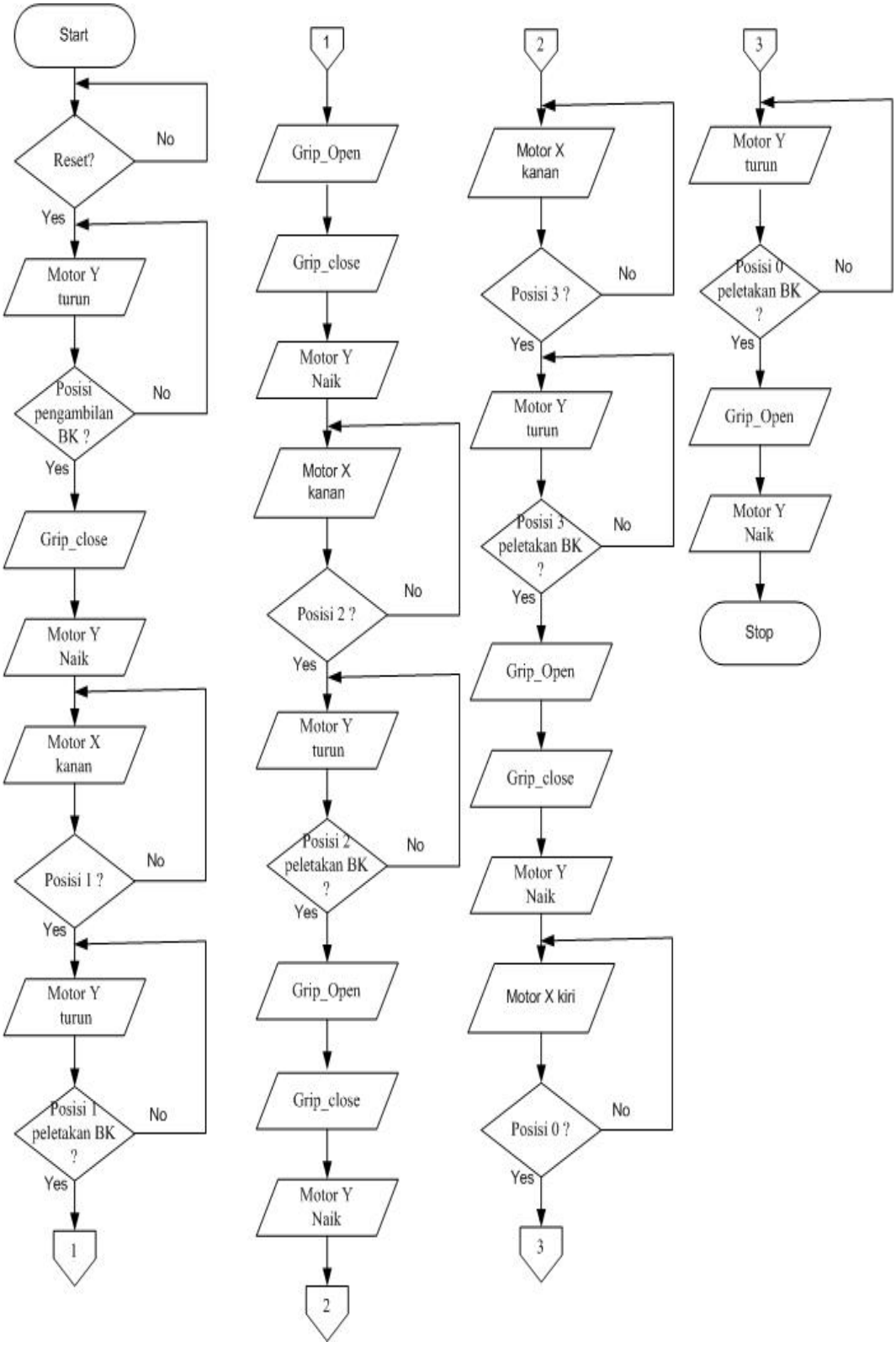
Keterangan program:

if(limit1==HIGH&&Tombolx_kanan==LOW&&Tombolx_kiri==HIGH) adalah jika limit switch minimum berlogika high dan tombol x+ berlogika low dan tombol x- berlogika high maka akan menjalankan intruksi program selanjutnya.

else(limit2==HIGH&&Tombolx_kanan==HIGH&&Tombolx_kiri==LO W) adalah program jika tidak memenuhi kondisi/pernyataan diatas maka limit switch maksimum berlogika high dan tombol x+ berlogika high dan tombol x- berlogika low.

4.8.4 Flowchart Program Demo

Flowchart program demo digunakan untuk menggerakkan motor stepper secara otomatis.



Gambar 4.24 Flowchart Program Demo

```
void demo()
{
  putary_ccw();
  delay(Tdemo);      Pergerakan motor stepper sumbu y berlawanan arah jarum jam
  grip_close();
  delay(Tdemo);      Pergerakan motor dc gripper close/tutup
  putary_cw();
  delay(Tdemo);      Pergerakan motor stepper sumbu y searah arah jarum jam
  pos1x_ccw();
  delay(Tdemo);      Pergerakan motor stepper sumbu x ke posisi 1
  putary_ccw();
  delay(Tdemo);
  grip_open();
  delay(Tdemo);      Pergerakan motor dc gripper open/buka
  grip_close();
  delay(Tdemo);
  putary_cw();
  delay(Tdemo);
  pos2x_ccw();
  delay(Tdemo);      Pergerakan motor stepper sumbu x ke posisi 2
  putary_ccw();
  delay(Tdemo);
  grip_open();
  delay(Tdemo);
  grip_close();
  delay(Tdemo);
  putary_cw();
  delay(Tdemo);
  pos3x_ccw();
  delay(Tdemo);      Pergerakan motor stepper sumbu x ke posisi 3
  putary_ccw();
  delay(Tdemo);
  grip_open();
  delay(Tdemo);
  grip_close();
  delay(Tdemo);
  putary_cw();
```

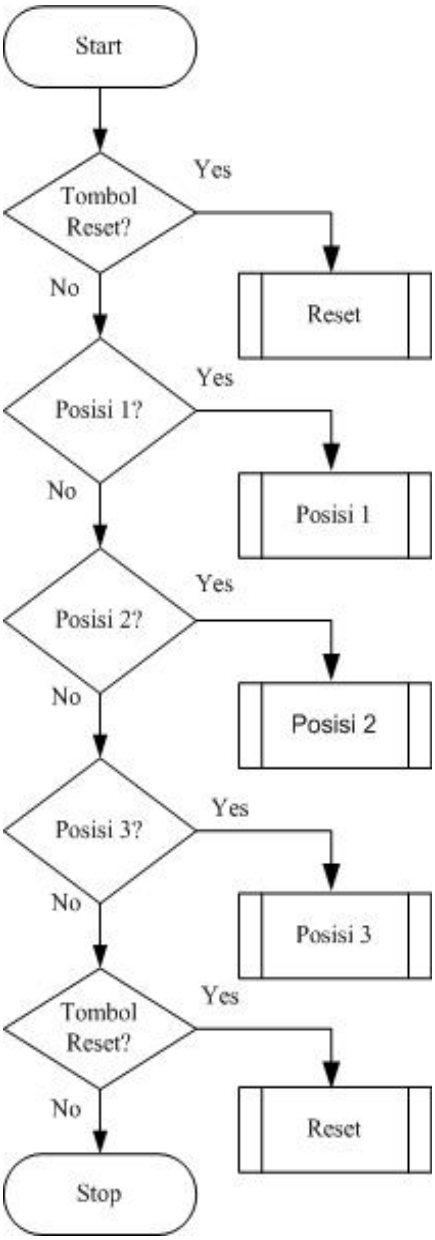
```
 delay(Tdemo);
 awal();
 delay(Tdemo);      ] Pergerakan motor stepper sumbu x ke posisi 0/posisi awal
 putary_ccw();
 delay(Tdemo);
 grip_open();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 demonstrasi=1;     ] Untuk menghentikan seluruh proses demostrasi robot
 delay(100);        ] manipulator
}
```

4.8.5 Reset

Tombol reset digunakan untuk mengembalikan kondisi robot keposisi awal atau posisi 0 titik benda kerja.

```
//=============================================RESET
void RESET()
{
  while(pindah==0)  ] Pengulangan program pada variabel pindah jika nilai pindah sama
                    ] dengan 0
 {
   int limit3= digitalRead(lm3);  ]
   int limit1=digitalRead(lm1);   ] Konfigurasi Limit Switch sumbu y minimum dan  x minimum
   motory_cw();
   if(limit3==LOW)  ] Jika Limit switch sumbu y minimum berlogika low
   {
          stopy();
         motorx_cw();
   }
   if(limit1==LOW)  ] Jika Limit switch sumbu y minimum berlogika low
   {
          stopx();
         grip_open();
          break;
   }
    }
}
```

4.8.6 Flowchart Proram Serial Monitor



Gambar 4.25 Flowchart Program Serial Monitor

Flowchart Program Serial Monitor digunakan untuk menjalankan robot sesuai dengan nilai titik koordinat input yang dimasukan, diawali dengan kondisi reset kemudian memasukan nilai titik koordinat sesuai dengan posisi yang diinginkan misalnya bisa memasukan titik koordinat untuk ke posisi 1, posisi 2 atau keposisi 3 secara berurutan atau secara acak.

```
//==SERIAL INPUT
if (Serial.available())
  {
    datax=Serial.readStringUntil('\n');
    Serial.print("Masukkan Posisi :  ");
    inputx=datax.toInt();
    Serial.println(inputx);
//==POSISI 0
    if(inputx==0)
    {
            Serial.println("Posisi 0");
             posisi=0;
             if(posisi<x)
     {
       int jumlah_siklus1=x-posisi;
       for(int i=0;i<jumlah_siklus1;i++)
       {
            motorx_cw();
            delay(Tx);
            Serial.println(x);
       }
       delay(1000);
       stopx();
     }
    }
//==POSISI 1
else if(inputx==1)
    {
     posisi=8;
      if( posisi>x)
      {
            Serial.println(posisi-x);
            int jumlah_siklus=posisi-x;
            for(int i=0;i<jumlah_siklus;i++)
       {
            motorx_ccw();
             delay(Tx);
            Serial.print(i);
            Serial.print("\t");
            Serial.println(x);
       }
       delay(1000);
       stopx();
      }
      if(posisi<x)
      {
        int jumlah_siklus1=x-posisi;
       for(int i=0;i<jumlah_siklus1;i++)
       {
             motorx_cw();
            delay(Tx);
            Serial.println(x);
       }
            delay(1000);
            stopx();
```

Program untuk konversi tipe data dari string ke integer.

Program untuk pergerakan ke posisi 0.

Program untuk pergerakan ke posisi 1.

```
    }
   }
//==POSISI 2
   else if(inputx==2)
   {
    posisi=14;
     if( posisi>x)
     {
       Serial.println(posisi-x);
      int jumlah_siklus=posisi-x;
       for(int i=0;i<jumlah_siklus;i++)
       {
            motorx_ccw();
           delay(Tx);
           Serial.print(i);
           Serial.print("\t");
           Serial.println(x);
       }
           delay(1000);
           stopx();
     }
     if(posisi<x)
     {
       int jumlah_siklus1=x-posisi;
       for(int i=0;i<jumlah_siklus1;i++)
       {
            motorx_cw();
           delay(Tx);
           Serial.println(x);
       }
           delay(1000);
           stopx();
     }
   }
//==POSISI 3
   else if(inputx==3)
   {
     posisi=21;
     if( posisi>x)
     {
       Serial.println(posisi-x);
       int jumlah_siklus=posisi-x;
       for(int i=0;i<jumlah_siklus;i++)
       {
           motorx_ccw();
           delay(Tx);
           Serial.print(i);
           Serial.print("\t");
           Serial.println(x);
       }
            delay(1000);
            stopx();
     }
     if(posisi<x)
     {
```

Program untuk pergerakan ke posisi 2.

Program untuk pergerakan ke posisi 3.

```
      int jumlah_siklus1=x-posisi;
     for(int i=0;i<jumlah_siklus1;i++)
     {
          motorx_cw();
          delay(Tx);
          Serial.println(x);
     }
          delay(1000);
           stopx();
    }
   }
//POSISI JIKA NILAI YANG DIINPUT SALAH
   else
   {
          Serial.println("Masukkan Ulang");
   }
  }
 }
```

Program untuk pergerakan ke posisi 3.

Program untuk pergerakan ke posisi sembarang.

4.8.7 LCD

Program yang berisi perintah untuk menampilkan jumlah siklus pergerakan motor X dan motor Y.

```
void setup() {
 lcd.init();
 lcd.init();
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Nilai X = ");
 lcd.setCursor(0,1);
 lcd.print("Nilai Y = ");
 lcd.setCursor(9,1);
 lcd.print(y1);
 lcd.setCursor(9,0);
 lcd.print(x1);
 Serial.begin(9600);
 pinMode(35,OUTPUT);
 pinMode(37,OUTPUT);
 pinMode(39,OUTPUT);
 pinMode(41,OUTPUT);
 pinMode(27,OUTPUT);
 pinMode(29,OUTPUT);
 pinMode(31,OUTPUT);
 pinMode(33,OUTPUT);
 pinMode(23,OUTPUT);
 pinMode(25,OUTPUT);
 pinMode(tombolx1,INPUT_PULLUP);
 pinMode(tombolx2,INPUT_PULLUP);
 pinMode(tomboly1,INPUT_PULLUP);
 pinMode(tomboly2,INPUT_PULLUP);
 pinMode(demns,INPUT_PULLUP);
```

Inisialisasi program LCD

Menghapus semua karakter dilayar LCD

Menampilkan nilai/karakter di LCD

Posisi karakter yang ditampilkan

Pin 35, 37, 39, 41, 27, 29, 31, 33, 23, 25 sebagai output

Tombol x+ dan x- sebagai input pull up

Tombol y+ dan y- sebagai input pull up

Tombol demonstrasi sebagai input pull up

```
  pinMode(lm1,INPUT_PULLUP);
  pinMode(lm2,INPUT_PULLUP);
  pinMode(lm3,INPUT_PULLUP);
  pinMode(lm4,INPUT_PULLUP);
  pinMode(gripbuka,INPUT_PULLUP);
  pinMode(griptutup,INPUT_PULLUP);
  pinMode(Rset,INPUT_PULLUP);
}
```

Limit switch 1, 2, 3, 4 sebagai input pull up

Gripper buka dan tutup sebagai input

Tombol reset sebagai input

Gambar 4.26 Tampilan hasil program LCD ketika direset

Gambar 4.27 Posisi X=1 dan Y pada posisi turun

Gambar 4.28 Posisi X=2 dan Y pada posisi naik

BAB V
PENUTUP

Setelah dilakukan pengujian alat, maka diperoleh beberapa kesimpulan dan saran yang dapat diharapkan berguna untuk perbendaharaan ilmu dan teknologi serta bagi kelanjutan dalam penyempurnaan proyek akhir.

5.1 Kesimpulan

Berdasarkan hasil studi dan analisa terhadap pembuatan proyek akhir, maka dapat diambil kesimpulan antara lain:

1. Pergerakan maksimal lengan manipulator pada sumbu x adalah 189 step sedangkan pada sumbu y adalah 36mm, pergerakan maksimal dan minimal lengan manipulator dibatasi oleh limit switch.

5.2 Saran

Pada proyek akhir ini hanya dapat digunakan untuk mengambil dan memindahkan barang, diharapkan dalam pengembangan proyek akhir dimasa mendatang dapat membuat robot yang membuat pergerakan robot yang lebih bervariasi.

DAFTAR PUSTAKA

[1] E. Purnama and T. A. Saputra, Prototype Robot Manipulator Dengan 2 Degress Of Freedom (DOF) Dan Gripper, Sungailiat, Bangka Belitung, 2010.

[2] "MOTOR STEPPER: TEKNOLOGI, METODA DAN RANGKAIAN KONTROL," [Online]. Available: jurnal.unikom.ac.id. [Accessed Senin April 2019].

[3] "pengertian-robot-manipulator," [Online]. Available: https://didinlubis.wordpress.com. [Accessed Rabu April 2019].

[4] "pengertian-motor-stepper," [Online]. Available: https://www.idekubagus.com. [Accessed Selasa April 2019].

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap	: Indarini
Tempat & Tanggal Lahir	: Simpang Pesak, 16 Desember 1997
Alamat	: Simpang Pesak, Belitung Timur
	Hp.081995645442
	Email : Indarini1713@gmail.com
Status	: Mahasiswi

2. Riwayat Pendidikan

SD Negeri 1 Simpang Pesak	Lulus 2010
SMP Negeri 1 Simpang Pesak	Lulus 2013
SMA Negeri 1 Simpang Pesak	Lulus 2016

3. Pengalaman Kerja

Praktik kerja lapangan di PT.Festo	Tahun 2018

4. Pengetahuan Bahasa : Bahasa Indonesia dan Bahasa inggris

5. Hobi : Olahraga

Sungailiat, Juli 2019

Indarini

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap	: Satria Hutomo Jaty
Tempat & Tanggal Lahir	: Pangkal Pinang, 10 Februari 1999
Alamat	: Jalan Baong No.6A Gabek 2
	Hp.082146582938
	Email : satriajaty10@gmail.com
Status	: Mahasiswa

2. Riwayat Pendidikan

SD Depati Amir Pangkal Pinang	Lulus 2010
SMP Negeri 2 Pangkal Pinang	Lulus 2013
SMK Negeri 2 Pangkal Pinang	Lulus 2016

3. Pengalaman Kerja

Praktik kerja lapangan di PT.Toso Industri Indonesia	Tahun 2018

4. Pengetahuan Bahasa inggris : Bahasa Indonesia, Bahasa Jepang dan Bahasa

5. Hobi : Nonton Film

Sungailiat, Juli 2019

Satria Hutomo Jaty

LAMPIRAN 2

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
int posisi=0;
int Tx=25,Ty=3;
int Tdemo=500;
int demonstrasi=0;
int a=0;
int lompat=0,pindah=0;
int ulangi=0;
int inputx=0;
String datax,datay;
long int x=0,y=0;
const int tombolx1=36;
const int tombolx2=38;
const int tomboly1=40;
const int tomboly2=42;
const int gripbuka=51;
const int griptutup=53;
const int demns=26;
const int Rset=44;
const int lm1=46;
const int lm2=48;
const int lm3=50;
const int lm4=52;
void setup() {
  lcd.backlight();
  lcd.init();
  lcd.init();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Nilai X = ");
  lcd.setCursor(0,1);
  lcd.print("Nilai Y = ");
  lcd.setCursor(9,1);
  lcd.print(y);
  lcd.setCursor(9,0);
  lcd.print(x);
  Serial.begin(9600);
pinMode(43,OUTPUT);
pinMode(45,OUTPUT);
pinMode(47,OUTPUT);
pinMode(49,OUTPUT);
pinMode(28,OUTPUT);
pinMode(30,OUTPUT);
pinMode(32,OUTPUT);
pinMode(34,OUTPUT);
pinMode(22,OUTPUT);
pinMode(24,OUTPUT);
pinMode(tombolx1,INPUT_PULLUP);
```

```
  pinMode(tombolx2,INPUT_PULLUP);
  pinMode(tomboly1,INPUT_PULLUP);
  pinMode(tomboly2,INPUT_PULLUP);
  pinMode(demns,INPUT_PULLUP);
  pinMode(lm1,INPUT_PULLUP);
  pinMode(lm2,INPUT_PULLUP);
  pinMode(lm3,INPUT_PULLUP);
  pinMode(lm4,INPUT_PULLUP);
  pinMode(gripbuka,INPUT_PULLUP);
  pinMode(griptutup,INPUT_PULLUP);
  pinMode(Rset,INPUT_PULLUP);
}

void motorx_cw() {
  digitalWrite(43,HIGH);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,HIGH);
  digitalWrite(45,HIGH);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,HIGH);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,HIGH);
  digitalWrite(47,HIGH);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,HIGH);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,HIGH);
  digitalWrite(49,HIGH);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,HIGH);
  delay(Tx);
  digitalWrite(43,HIGH);
```

```
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,HIGH);
  delay(Tx);
  x=x-1;

}
void motorx_ccw(){
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,HIGH);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,HIGH);
  digitalWrite(49,HIGH);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,HIGH);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,HIGH);
  digitalWrite(47,HIGH);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,LOW);
  digitalWrite(45,HIGH);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,HIGH);
  digitalWrite(45,HIGH);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,HIGH);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
  delay(Tx);
  digitalWrite(43,HIGH);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,HIGH);
  delay(Tx);
  x=x+1;
```

```
}

void motory_cw(){
 digitalWrite(28,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,HIGH);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,HIGH);
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,HIGH);
  delay(Ty);
  y=y+1;

}

void motory_ccw() {
 digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,HIGH);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,HIGH);
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,LOW);
  digitalWrite(30,HIGH);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);
  digitalWrite(28,HIGH);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
  delay(Ty);
    y=y-1;
}
```

```
void stopx(){
  digitalWrite(43,LOW);
  digitalWrite(45,LOW);
  digitalWrite(47,LOW);
  digitalWrite(49,LOW);
}

void stopy(){
  digitalWrite(28,LOW);
  digitalWrite(30,LOW);
  digitalWrite(32,LOW);
  digitalWrite(34,LOW);
}
//================================POSISI DEMO

void pos1x_ccw(){
  for (int i=0;i<=8;i++)
    {
          motorx_ccw();
          delay(Tx);
    }
          delay(1000);
         stopx();
}

void pos2x_ccw()
{
  for (int j=0;j<=6;j++)
    {
         motorx_ccw();
         delay(Tx);
    }
         delay(1000);
         stopx();
}
void pos3x_ccw()
{
for (int k=0;k<=6;k++)
    {
          motorx_ccw();
         delay(Tx);
    }
         delay(1000);
          stopx();
}
void awal()
{
for (int m=0;m<=20;m++)
    {
```

```
            motorx_cw();
            delay(20);
    }
             delay(1000);
            stopx();
}

void putary_ccw()
{
 for (int i=0;i<=350;i++)
    {
            motory_ccw();
            delay(Ty);
    }
            delay(1000);
             stopy();
}
void putary_cw()
{
  for (int i=0;i<=350;i++)
    {
            motory_cw();
            delay(Ty);
    }
            delay(1000);
            stopy();
}
//=====================================GERAK GRIPPER

void grip_close()
{
  digitalWrite(44,HIGH);
  digitalWrite(46,LOW);
  for(int i=0;i<=4000;i++)
  {
             delay(1);
  }
  digitalWrite(44,LOW);
  digitalWrite(46,LOW);
}

void grip_open(){
  digitalWrite(44,LOW);
  digitalWrite(46,HIGH);
  for(int z=0;z<=4000;z++)
  {
            delay(1);
  }
  digitalWrite(44,LOW);
  digitalWrite(46,LOW);
```

```
}

//================================================RESET

void RESET()
{
  while(pindah==0)
 {
   int limit3= digitalRead(lm3);
   int limit1=digitalRead(lm1);
   motory_cw();
   if(limit3==LOW)
    {
          stopy();
           motorx_cw();
    }
   if(limit1==LOW)
    {
          stopx();
          grip_open();
          break;
    }
  }
}
//================================== DEMO
void demo()
{
 putary_ccw();
 delay(Tdemo);
 grip_close();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 pos1x_ccw();
 delay(Tdemo);
 putary_ccw();
 delay(Tdemo);
 grip_open();
 delay(Tdemo);
 grip_close();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 pos2x_ccw();
 delay(Tdemo);
 putary_ccw();
 delay(Tdemo);
 grip_open();
 delay(Tdemo);
```

```
 grip_close();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 pos3x_ccw();
 delay(Tdemo);
 putary_ccw();
 delay(Tdemo);
 grip_open();
 delay(Tdemo);
 grip_close();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 awal();
 delay(Tdemo);
 putary_ccw();
 delay(Tdemo);
 grip_open();
 delay(Tdemo);
 putary_cw();
 delay(Tdemo);
 demonstrasi=1;
 delay(100);
}

void loop()
{
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Nilai X = ");
  lcd.setCursor(0,1);
  lcd.print("Nilai Y = ");
  lcd.setCursor(9,1);
  lcd.print(y);
  lcd.setCursor(9,0);
  lcd.print(x);

//=====================================================
==========Demonstrasi
  int sequent = digitalRead(demns);

 while(sequent==LOW && demonstrasi==0)
 {
          demo();
          demonstrasi=1;
           break;
 }
 //================================================RESET
 int Reset=digitalRead(Rset);
```

```
if(Reset==LOW)
{
  RESET();
}

////=====================================MANUAL SUMBU X
 int Tombolx_kanan = digitalRead(tombolx1);
 int Tombolx_kiri = digitalRead(tombolx2);
 int limit2= digitalRead(lm2);
 int limit1=digitalRead(lm1);
   if ( limit1==HIGH && Tombolx_kanan == LOW && Tombolx_kiri==HIGH )
{
            motorx_cw();
           lcd.clear();
           lcd.setCursor(9,0);
           lcd.print(x);
}
   else if ( limit2==HIGH && Tombolx_kanan == HIGH && Tombolx_kiri==LOW)
 {
            motorx_ccw();
           lcd.clear();
           lcd.setCursor(9,0);
           lcd.print(x);
 }
   else
 {
           stopx();
 }
//=====================================MANUAL SUMBU Y
int Tomboly_naik=digitalRead(tomboly1);
int Tomboly_turun=digitalRead(tomboly2);

 int limit4=digitalRead(lm4);
 int limit3= digitalRead(lm3);
 if ( limit3==HIGH && Tomboly_naik == LOW && Tomboly_turun==HIGH )
    {
            motory_cw();
            lcd.clear();
            lcd.setCursor(9,1);
           lcd.print(y);
    }
else if ( limit4==HIGH && Tomboly_naik == HIGH && Tomboly_turun==LOW)
   {
  motory_ccw();
  lcd.clear();
  lcd.setCursor(9,1);
  lcd.print(y);
   }
 else
  {
```

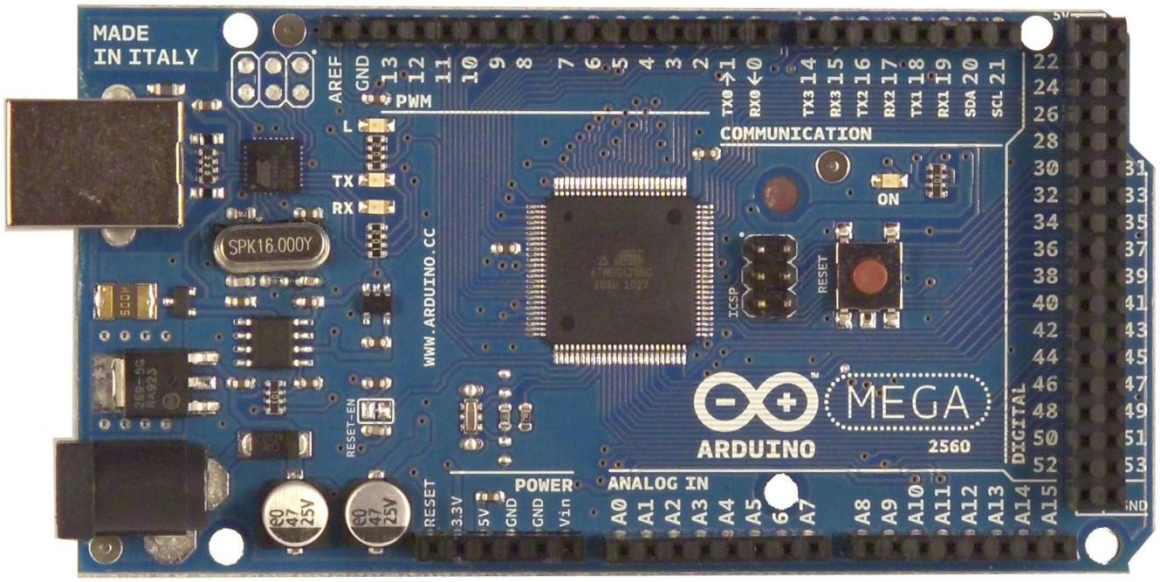
```
 stopy();
 }

//============================================== GRIPPER
int tombol_gripbuka=digitalRead(gripbuka);
int tombol_griptutup=digitalRead(griptutup);
if (tombol_gripbuka==LOW && tombol_griptutup==HIGH)
 {
        digitalWrite(22,LOW);
        digitalWrite(24,HIGH);
 }
else if (tombol_gripbuka==HIGH && tombol_griptutup==LOW)
 {
        digitalWrite(22,HIGH);
         digitalWrite(24,LOW);
 }
else
{
        digitalWrite(22,LOW);
         digitalWrite(24,LOW);
}
//=========================================SERIAL INPUT
if (Serial.available())
  {
         datax=Serial.readStringUntil('\n');
        Serial.print("Masukkan Posisi :  ");
         inputx=datax.toInt();
        Serial.println(inputx);
//================================================POSISI 0
   If (inputx==0)
    {
      Serial.println("Posisi 0");
      posisi=0;
     if (posisi<x)
    {
      int jumlah_siklus1=x-posisi;
      for(int i=0;i<jumlah_siklus1;i++)
      {
       motorx_cw();
       delay(Tx);
       Serial.println(x);
      }
      delay(1000);
      stopx();
    }
   }
//=======================================POSISI 1
   else if(inputx==1)
   {
```

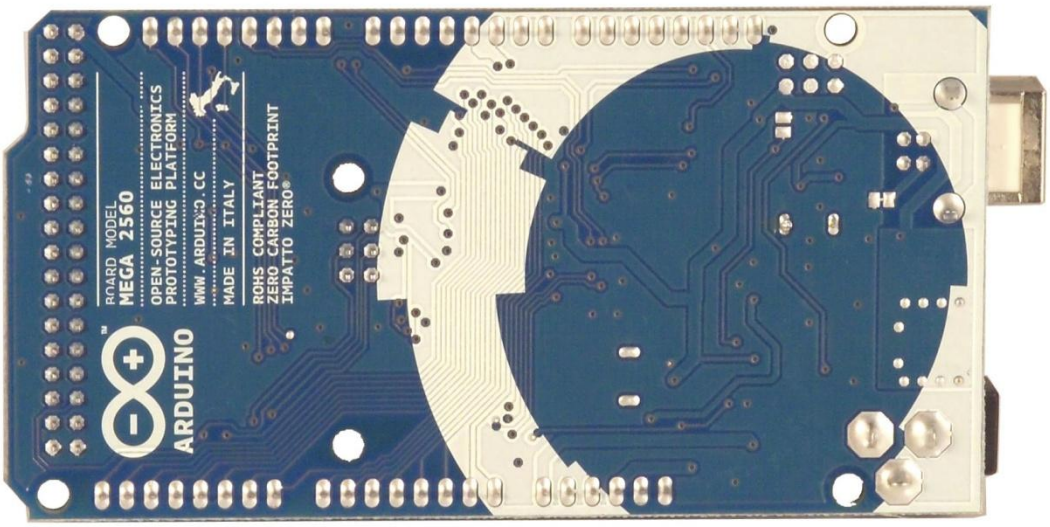
```
    posisi=8;
     if( posisi>x)
     {
       Serial.println(posisi-x);
       int jumlah_siklus=posisi-x;
       for(int i=0;i<jumlah_siklus;i++)
       {
         motorx_ccw();
         delay(Tx);
         Serial.print(i);
         Serial.print("\t");
         Serial.println(x);
       }
       delay(1000);
       stopx();
     }
     if(posisi<x)
     {
        int jumlah_siklus1=x-posisi;
       for(int i=0;i<jumlah_siklus1;i++)
       {
         motorx_cw();
         delay(Tx);
         Serial.println(x);
       }
       delay(1000);
       stopx();
     }
    }
//=========================================POSISI 2
   else if(inputx==2)
   {
    posisi=14;
     if( posisi>x)
     {
       Serial.println(posisi-x);
       int jumlah_siklus=posisi-x;
       for(int i=0;i<jumlah_siklus;i++)
       {
         motorx_ccw();
         delay(Tx);
         Serial.print(i);
         Serial.print("\t");
         Serial.println(x);
       }
       delay(1000);
       stopx();
     }
     if(posisi<x)
     {
```

```
      int jumlah_siklus1=x-posisi;
      for(int i=0;i<jumlah_siklus1;i++)
      {
       motorx_cw();
       delay(Tx);
       Serial.println(x);
      }
      delay(1000);
      stopx();
     }
    }
//==================================POSISI 3
    else if(inputx==3)
    {
     posisi=21;
     if( posisi>x)
     {
      Serial.println(posisi-x);
      int jumlah_siklus=posisi-x;
      for(int i=0;i<jumlah_siklus;i++)
      {
       motorx_ccw();
       delay(Tx);
       Serial.print(i);
       Serial.print("\t");
       Serial.println(x);
      }
      delay(1000);
      stopx();
     }
     if(posisi<x)
     {
       int jumlah_siklus1=x-posisi;
      for(int i=0;i<jumlah_siklus1;i++)
      {
       motorx_cw();
       delay(Tx);
       Serial.println(x);
      }
      delay(1000);
      stopx();
     }
    }
//=======POSISI JIKA NILAI YANG DIINPUT SALAH
    else
    {
     Serial.println("Masukkan Ulang");
    }
  }
 }
```

LAMPIRAN 3

Arduino Mega 2560
Datasheet





Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet).
It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog
inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a
power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-
to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic: [arduino-mega2560-schematic.pdf]

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) orbattery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

it through this pin.

- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode() , digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the

- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH.

value, the LED is on, when the pin is LOW, it's off.

- **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins.The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the


documentation on the Wiring website for details. For SPI communication, use the SPI library

Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials.
The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. Itcommunicates using the original STK500 protocol (reference, C header files).You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.
If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively,with the USB connector and power jack extending beyond the former dimension. Threescrew holes allow the board to be attached to a surface or case. Note that the distancebetween digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I^2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*