

**ROBOT PEMETAAN JARAK TANAM
BERBASIS *SMARTPHONE***

PROYEK AKHIR

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



Disusun Oleh :

Noviansyah NIRM : 0031619

Rafirda Aini NIRM : 0031620

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2019

LEMBAR PENGESAHAN

**ROBOT PEMETAAN JARAK TANAM
BERBASIS *SMARTPHONE***

Oleh :

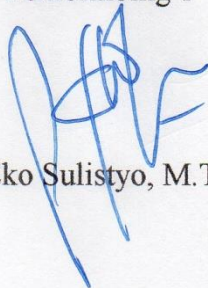
Noviansyah NIRM : 0031619

Rafirda Aini NIRM : 0031620

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

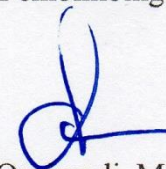
Menyetujui,

Pembimbing 1



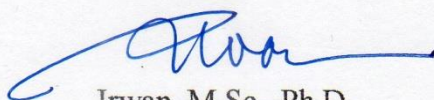
Eko Sulistyono, M.T

Pembimbing 2



Ocsirendi, M.T

Penguji 1



Irwan, M.Sc., Ph.D

Penguji 2



Nofriyani, M.T

Penguji 3



Zanu Saputra, M.T

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Mahasiswa 1 :Noviansyah NIRM : 0031619

Mahasiswa 2 : Rafirda Aini NIRM : 0031620

Dengan judul : Robot Pemetaan Jarak Tanam Berbasis *Smartphone*

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

Sungailiat, 21 Agustus 2019

Nama Mahasiswa

1. Noviansyah
2. Rafirda Aini

Tanda Tangan



ABSTRAK

Kualitas pertumbuhan tanaman salah satunya dipengaruhi oleh kerapatan jarak tanam antar tanaman melalui proses pemetaan jarak tanam. Proses pemetaan jarak tanam tersebut masih dilakukan dengan cara manual yaitu mengukur jarak tanam menggunakan meteran dan menggunakan peralatan fisik lainnya untuk menandai titik tanamnya, sehingga untuk menentukan jarak antar titik tanam dan untuk menandai titik tanam tergantung dari tenaga manusia. Untuk mengatasi masalah tersebut dibuatlah sistem robot yang dapat melakukan pemetaan jarak tanam yang tergantung pada keinginan pengguna dengan memasukkan data berupa luas lahan dan jarak tanam melalui smartphone. Metodologi yang digunakan pada proyek akhir ini adalah membuat pemetaan jarak tanam dengan menggunakan robot. Proses pemetaan dilakukan dengan cara memasukkan data berupa luas lahan dan jarak tanam dari smartphone kemudian data tersebut dikirim dan diolah oleh mikrokontroler untuk mengendalikan robot sesuai dengan data yang di masukkan. Hasil yang didapat dari pengujian menunjukkan waktu pengiriman data dari smartphone ke modul wifi yang akan diolah mikrokontroler kurang dari 15 detik dengan jangkauan koneksi 100 m. Rata-rata persentase error pengujian proses pemetaan dan pengeboran jarak tanam yang dilakukan robot dengan nilai rata-rata error koordinat X 2,66%, koordinat Y 3% dan jarak tanam 4%. Robot dapat melakukan pemetaan jarak tanam dengan waktu selama 85 menit 5 detik dan pengisian baterai selama 2 jam menggunakan spesifikasi baterai 12 V 9 Ah.

Kata kunci : *Jarak tanam, Robot, Smartphone*

ABSTRACT

The quality of plant growth is influenced by the density of plant spacing between plants through the process of mapping plant spacing. The process of mapping plant spacing is still done manually by measuring the spacing using a meter and using other physical equipment to mark the planting points, so as to determine the spacing between planting points and to mark the planting points depending on human labor. To overcome these problems, a robot system was created that can map planting distances depending on the user's wishes by entering data in the form of land area and planting distance through a smartphone. The methodology used in this final project is to make a mapping of planting distances using robots. The mapping process is done by entering data in the form of land area and planting distance from the smartphone then the data is sent and processed by a microcontroller to control the robot in accordance with the data entered. The results obtained from the test show the time of sending data from the smartphone to the wifi module which will be processed by a microcontroller in less than 15 seconds with a connection range of 100 m. The average percentage of error testing the process of mapping and drilling distance planted by robots with an average value of X coordinate errors 2.66%, Y coordinates 3% and planting distance 4%. The robot can map the planting distance with a time of 85 minutes 5 seconds and charging the battery for 2 hours using the 12 V 9 Ah battery specifications.

Keywords: Plant distance, Robot, Smartphone

KATA PENGANTAR

Puji dan Syukur penulis panjatkan kehadirat Allah SWT karena atas rahmat dan ridha-Nya sehingga penulis dapat menyelesaikan karya tulis proyek akhir dengan judul “Robot Pemetaan Jarak Tanam Berbasis *Smartphone*” di Politeknik Manufaktur Negeri Bangka Belitung dapat terselesaikan dengan baik.

Karya tulis proyek akhir ini disusun untuk memenuhi salah satu persyaratan akademik dan kewajiban mahasiswa pada semester VI (enam) untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Laporan ini dapat terselesaikan atas bantuan dan bimbingan dari semua pihak. Untuk itu penulis mengucapkan terimakasih kepada semua pihak yang ikut membantu dalam penyelesaian laporan ini, terutama kepada :

1. Bapak Eko Sulistyono, M.T, selaku pembimbing I sekaligus Kepala Prodi D3 Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung yang telah memberi saran dan solusi dari masalah yang penulis hadapi selama proses penyusunan proyek akhir ini.
2. Bapak Ocsirendi, M.T selaku pembimbing II yang telah memberi saran dan solusi dari masalah penulis hadapi selama proses penyusunan proyek akhir ini.
3. Bapak Aan Febriansyah, M.T selaku Kepala Jurusan Teknik Elektronika Dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung
5. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
6. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan Proyek Akhir.
7. Kedua orang tua dan keluarga yang telah mendoakan dan selalu memberikan motivasi serta dukungan.
8. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan proyek akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari sepenuhnya bahwa karya tulis ini masih jauh dari sempurna. Untuk itu kritik dan saran yang sifatnya membangun demi kesempurnaan laporan ini sangat penulis harapkan. Mudah-mudahan karya tulis ini dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya.

Sungailiat, 21 Agustus 2019

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DARTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I <u>P</u> ENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir	3
BAB II <u>D</u> ASAR TEORI.....	4
2.1 Pengertian Robot	4
2.1.1 Robot Penanam Benih Jagung	5
2.1.2 Robot Tanam Benih Langsung (TABELA).....	6
2.1.3 Robot Penyiram Tanaman Otomatis.....	6
2.1.4 EcoRobotix	7
2.2 Penanaman.....	8
2.2.1 Pembuatan Peta Areal Kerja	8
2.2.2 Penataan Areal	9
2.2.3 Penentuan Jarak Tanam	9
2.2.4 Penentuan Arah Larikan	10
2.2.5 Pembersihan Lahan Dari Gulma.....	10

2.2.6	Pemasangan Ajir	10
2.2.7	Pembuatan Lubang Tanam	10
BAB III METODE PELAKSANAAN		11
3.1	Survei, Pengumpulan Data dan Pengolahan Data	12
3.2	Perancangan <i>Hardware</i> Mekanik Robot	13
3.3	Pembuatan <i>Hardware</i> Mekanik Robot	13
3.4	Perakitan <i>Hardware</i> Mekanik Robot.....	13
3.5	Implementasi <i>Hardware</i> Elektrik Robot	14
3.6	Pengujian <i>Hardware</i> Elektrik Robot	14
3.7	Perancangan <i>Software</i> Robot.....	14
3.8	Pembuatan <i>Software</i> Robot	14
3.9	Pengujian <i>Software</i> Robot	15
3.10	Perakitan <i>Hardware</i> Mekanik Dan Elektrik Robot	15
3.11	Pengujian Sistem Robot Pemetaan Jarak Tanam.....	15
3.12	Pembuatan Laporan Proyek Akhir.....	15
BAB IV PEMBAHASAN.....		16
4.1	Perancangan <i>Hardware</i> Robot Pemetaan Jarak Tanam.....	16
4.1.1	Perancangan <i>Hardware</i> Mekanik Robot Pemetaan Jarak Tanam ...	16
4.1.2	Pembuatan <i>Hardware</i> Mekanik Robot Pemetaan Jarak Tanam	18
4.1.3	Perakitan <i>Hardware</i> Mekanik Robot Pemetaan Jarak Tanam.....	20
4.2	Implementasi <i>Hardware</i> Elektrik Robot Pemetaan Jarak Tanam	21
4.2.1	Diagram Blok.....	21
4.2.2	<i>Buck Converter</i>	22
4.2.3	Arduino Uno	28
4.2.4	Sensor Tegangan DC	30
4.2.5	Sensor <i>Rotary Encoder</i> KY-040	34
4.2.6	Sensor kompas GY-271	37
4.2.7	NodeMCU ESP8266.....	40
4.2.8	<i>Driver</i> IBT-02	42
4.2.9	Modul <i>Relay 2-Channel</i>	44
4.3	Perancangan <i>Software</i> Robot Pemetaan Jarak Tanam.....	46

4.3.1	Aplikasi <i>Blynk</i>	46
4.3.2	Pemrograman	49
4.4	Perakitan <i>Hardware</i> Mekanik Dan Elektrik Robot	50
4.5	Pengujian Sistem Robot Pemetaan Jarak Tanam.....	51
4.5.1	Pengujian Memasukkan Data Melalui <i>Smartphone</i>	51
4.5.2	Pengujian Robot Pemetaan Jarak Tanam.....	53
4.5.3	Pengujian Pola Jalan Robot Pemetaan Jarak Tanam	66
4.5.4	Pengujian Waktu Pengerjaan Pemetaan Jarak Tanam	70
4.5.5	Pengujian Pemakaian dan Pengisian Baterai	71
BAB V_PENUTUP.....		74
5.1	Kesimpulan	74
5.2	Saran	74
DAFTAR PUSTAKA		xvi
LAMPIRAN		

DAFTAR TABEL

Tabel	Halaman
Tabel 2. 1 Jarak Tanam Pada Tanaman	9
Tabel 4. 1 Hasil Pengujian <i>Buck Converter Step Down</i> LM2596.....	25
Tabel 4. 2 Hasil Pengujian <i>Breadboard Power Supply</i> MB102	26
Tabel 4. 3 Hasil Pengujian Regulator IC 7810	27
Tabel 4. 4 Hasil Pengujian Pin <i>Output</i> Arduino Uno.....	30
Tabel 4. 5 Hasil Pengujian Sensor Tegangan DC	33
Tabel 4. 6 Hasil Pengujian Sensor <i>Rotary Encoder</i> KY-040.....	36
Tabel 4. 7 Hasil Pengujian Sensor Kompas GY-271 HMC5883L	39
Tabel 4. 8 Hasil Pengujian LED Pada NodeMCU ESP8266	41
Tabel 4. 9 Hasil Pengujian <i>Driver</i> IBT-02.....	43
Tabel 4. 10 Hasil Pengujian <i>Relay 2-Channel</i>	45
Tabel 4. 11 Pengujian Jangkauan NodeMCU ESP8266 Terhadap <i>Smartphone</i> ...	53
Tabel 4. 12 Hasil Pengujian 1 Koordinat Pengeboran (a).....	56
Tabel 4. 13 Hasil Pengujian 2 Koordinat Pengeboran (a).....	57
Tabel 4. 14 Hasil Pengujian 1 Jarak Tanam (a)	57
Tabel 4. 15 Hasil Pengujian 2 Jarak Tanam (a)	58
Tabel 4. 16 Hasil Pengujian 1 Koordinat Pengeboran (b)	59
Tabel 4. 17 Hasil Pengujian 2 Koordinat Pengeboran (b)	60
Tabel 4. 18 Hasil Pengujian 1 Jarak Tanam (b)	60
Tabel 4. 19 Hasil Pengujian 2 Jarak Tanam (b)	61
Tabel 4. 20 Hasil Pengujian 1 Koordinat Pengeboran (c).....	62
Tabel 4. 21 Hasil Pengujian 2 Koordinat Pengeboran (c).....	62
Tabel 4. 22 Hasil Pengujian 1 Jarak Tanam (c)	63
Tabel 4. 23 Hasil Pengujian 2 Jarak Tanam (c)	64
Tabel 4. 24 Hasil Pengujian Pola Jalan Robot Pemetaan Jarak Tanam	68
Tabel 4. 25 Hasil Pengujian Waktu Pemetaan Jarak Tanam	71
Tabel 4. 26 Hasil Pemakaian Arus dan Tegangan Robot Pemetaan Jarak Tanam	72

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2. 1 Robot Penanam Benih Jagung.....	5
Gambar 2. 2 3D Robot Tanam Benih Langsung.....	6
Gambar 2. 3 Robot Penyiraman Tanaman Otomatis	7
Gambar 2. 4 EcoRobotix.....	8
Gambar 3. 1 <i>Flowchart</i> Metode Pelaksanaan Proyek Akhir.....	12
Gambar 4. 1 Perancangan Robot Pemetaan Jarak Tanam.....	17
Gambar 4. 2 Dudukan dan Poros Roda.....	18
Gambar 4. 3 Kaki Robot	19
Gambar 4. 4 <i>Base</i> Robot	19
Gambar 4. 5 Sistem Pengeboran	20
Gambar 4. 6 <i>Hardware</i> Mekanik Robot Pemetaan Jarak Tanam	21
Gambar 4. 7 Diagram Blok Sistem Kontrol Robot Pemetaan Jarak Tanam	21
Gambar 4. 8 Modul <i>Buck Converter Step Down</i> LM2596.....	23
Gambar 4. 9 Modul <i>Breadboard Power Supply</i> MB102	23
Gambar 4. 10 Skematik Regulator IC 7810	24
Gambar 4. 11 Hasil Pembuatan Regulator IC 7810.....	24
Gambar 4. 12 Blok Pengujian <i>Buck Converter</i>	24
Gambar 4. 13 Pengujian <i>Buck Converter Step Down</i> LM2596	25
Gambar 4. 14 Pengujian <i>Breadboard Power Supply</i> MB102.....	26
Gambar 4. 15 Pengujian Regulator IC 7810	27
Gambar 4. 16 Arduino Uno.....	28
Gambar 4. 17 Skematik Pengujian Arduino Uno.....	29
Gambar 4. 18 Sensor Tegangan DC.....	31
Gambar 4. 19 Rangkaian Pengujian Sensor Tegangan DC.....	31
Gambar 4. 20 Hasil Pengujian 1 Sensor Tegangan DC Dengan Alat Ukur.....	32
Gambar 4. 21 Hasil Pengujian 2 Sensor Tegangan DC Dengan Alat Ukur.....	32
Gambar 4. 22 Hasil Pengujian 3 Sensor Tegangan DC Dengan Alat Ukur.....	33

Gambar 4. 23 Sensor <i>Rotary Encoder</i> KY-040	34
Gambar 4. 24 Sensor <i>Rotary Encoder</i> KY-040	35
Gambar 4. 25 Blok Diagram Pengujian Sensor <i>Rotary Encoder</i> KY-040.....	35
Gambar 4. 26 Pembacaan Sensor <i>Rotary Encoder</i> KY-040	36
Gambar 4. 27 Sensor Kompas GY-271 HMC5883L.....	37
Gambar 4. 28 Titik Pengujian Sensor Kompas GY-271 HMC5883L.....	38
Gambar 4. 29 Pengujian Pembacaan Sensor Kompas GY-271 HMC5883L.....	39
Gambar 4. 30 Modul NodeMCU ESP8266.....	40
Gambar 4. 31 Blok Diagram Pengujian NodeMCU ESP8266	40
Gambar 4. 32 Pengujian NodeMCU ESP8266	41
Gambar 4. 33 Modul <i>Driver</i> IBT-02.....	42
Gambar 4. 34 Blok Diagram Pengujian <i>Driver</i> IBT-02.....	42
Gambar 4. 35 Pengujian <i>Driver</i> IBT-02 Terhadap Motor DC <i>Power Window</i> ...	43
Gambar 4. 36 Modul <i>Relay 2-Channel</i>	44
Gambar 4. 37 Diagram Blok Pengujian <i>Relay 2-Channel</i>	44
Gambar 4. 38 Pengujian <i>Relay 2-Channel</i> Terhadap Motor DC	45
Gambar 4. 39 <i>Layout</i> Aplikasi <i>Blynk</i>	46
Gambar 4. 40 Hasil Pembuatan <i>Layout</i> Pada Aplikasi <i>Blynk</i>	47
Gambar 4. 41 Pengujian Aplikasi <i>Blynk</i>	48
Gambar 4. 42 Pengujian Data Yang Diterima Arduino Dari Aplikasi <i>Blynk</i>	48
Gambar 4. 43 <i>Flowchart</i> Arduino Uno	49
Gambar 4. 44 Hasil Perakitan Robot Pemetaan Jarak Tanam	51
Gambar 4. 45 Pengujian <i>Input</i> Nilai Sudut.....	52
Gambar 4. 46 Pengujian <i>Input</i> Luas Lahan Dan Jarak Tanam	52
Gambar 4. 47 Pengujian Pengeboran Titik Tanam	54
Gambar 4. 48 Pengujian Pengukuran Jarak Tanam	54
Gambar 4. 49 Simulasi Pengujian Jumlah Tanaman Secara Manual.....	55
Gambar 4. 50 Pengujian Jumlah Tanaman Manual (a).....	56
Gambar 4. 51 Pengujian 2 Jumlah Titik Tanam Manual (b).....	59
Gambar 4. 52 Pengujian 3 Jumlah Titik Tanam Manual (c).....	61
Gambar 4. 53 Simulasi Pengujian Pola Jalan Robot Pertama.....	67

Gambar 4. 54 Simulasi Pengujian Pola Jalan Robot Kedua	67
Gambar 4. 55 Pengujian Waktu Pola Jalan Robot	68
Gambar 4. 56 Pengujian Waktu Menggunakan Robot.....	70
Gambar 4. 57 Pengujian Waktu Menggunakan Cara Manual.....	71
Gambar 4. 58 Pengujian Pengukuran Arus Pada Motor DC <i>Power Window</i>	72

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : Desain Robot Pemetaan Jarak Tanam Berbasis *Smartphone*

Lampiran 3 : Program Keseluruhan

Lampiran 4 : Standar Operasional Prosedur (SOP) Robot Pemetaan Jarak Tanam
Berbasis *Smartphone*

Lampiran 5 : *Datasheet* Komponen Elektrik

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penanaman adalah kegiatan memindahkan bibit dari tempat penyemaian ke lahan penanaman untuk di dapatkan hasil produk berkualitas dari tanaman yang di budidayakan. Proses pemindahan bibit tidak boleh di lakukan dengan sembarangan, perlu adanya metode agar tanaman dapat hidup di media dan lingkungannya yang baru.

Persiapan yang harus dilakukan sebelum proses menanam selain persiapan bibit adalah membuat peta lahan dengan memperhitungkan kerapatan jarak tanam. Peningkatan pertumbuhan tanaman dapat diupayakan melalui pengaturan kerapatan jarak tanam sampai mencapai populasi optimal. Kegiatan tersebut kelihatannya mudah dilakukan, namun kesalahan kecil dalam kerapatan jarak tanam berpengaruh terhadap kualitas pertumbuhan tanaman selanjutnya.

Pengaturan kerapatan tanaman bertujuan meminimalkan kompetisi antar populasi agar tanaman dapat memanfaatkan lingkungan secara optimal. Jarak tanam yang terlalu rapat akan menghambat pertumbuhan tanaman karena jumlah populasi tanaman yang berlebihan dapat menurunkan hasil dan terjadi persaingan terhadap unsur hara, air, radiasi matahari, dan ruang. Sedangkan jarak tanam yang terlalu lebar dapat mengurangi jumlah tanaman yang ditanam sehingga memperbaiki pertumbuhan individu tanaman. Namun, hal ini menyebabkan kurangnya pemanfaatan sinar matahari yang jatuh ke permukaan tanah dan unsur hara akan hilang karena terjadi proses penguapan serta memberikan peluang bagi gulma untuk tumbuh disekitar tanaman [1].

Di Bangka Belitung, cara pemetaan jarak tanam tersebut masih dilakukan secara manual atau tradisional yaitu dengan mengukur jarak tanam menggunakan meteran dan menggunakan peralatan fisik lainnya untuk menandai titik tanamnya. Dari hasil wawancara yang telah dilakukan ke beberapa petani lada di Desa Kimak Kecamatan Merawang bahwa para petani melakukan proses pemetaan

jarak tanam masih menggunakan cara tradisional dengan menggunakan alat ukur baik berupa meteran maupun tali serta menandai lubang tanam dengan menggunakan kayu yang telah diruncingkan terlebih dahulu. Secara tradisional metode ini masih di rasakan efektif oleh sebagian petani, akan tetapi cara seperti ini juga mempunyai beberapa faktor kekurangan yaitu menentukan jarak tanam yang kurang optimal dan menentukan berapa jumlah tanaman yang dapat ditanam dilahan serta masalah waktu yang dibutuhkan untuk melakukan pemetaan jarak tanam tergantung dari tenaga manusia.

Berdasarkan uraian diatas perlu sebuah sistem kontrol menggunakan robot yang dapat melakukan pemetaan jarak tanam dengan berbasis *smartphone*. Proses sistem kontrol pemetaan jarak tanam meliputi pemberian tanda atau titik tanam pada lahan yang akan ditanami dengan memasukkan data melalui *smartphone* oleh pengguna. Pengguna juga bisa melakukan perhitungan terhadap jumlah tanam sesuai luas lahan yang akan dipetakan. Dengan sistem kontrol robot pemetaan tersebut diharapkan dapat mempermudah pekerjaan manusia dalam pembuatan jarak dan titik tanam serta menentukan jumlah tanaman yang dapat ditanam.

1.2 Rumusan Masalah

Rumusan masalah pada tugas proyek akhir ini adalah:

1. Bagaimana cara mendesain dan membuat konstruksi robot pemetaan jarak tanam.
2. Bagaimana cara membuat sistem kontrol robot pemetaan jarak tanam.
3. Bagaimana cara membuat aplikasi untuk *smartphone* yang bisa memasukkan data luas lahan yang dapat ditanami, jarak tanam dan menampilkan daya yang digunakan.

1.3 Batasan Masalah

Batasan masalah pada tugas proyek akhir ini adalah:

1. Luas lahan yang digunakan hanya berdimensi persegi dengan kondisi lahan yang sudah dibersihkan dan permukaan tanah datar serta tidak gembur.

2. Jarak tanam berdimensi persegi.
3. Komunikasi *smartphone* dengan mikrokontroller tergantung dari jaringan *smartphone* sendiri.
4. Jangkauan memasukkan data dari *smartphone* dengan robot pemetaan jarak tanam berjarak maksimal 100 m.

1.4 Tujuan Proyek Akhir

Tujuan dalam proyek akhir ini adalah membuat robot yang dapat melakukan pemetaan jarak tanam yang tergantung pada keinginan pengguna dengan memasukkan data berupa luas lahan dan jarak tanam melalui *smartphone*.

BAB II

DASAR TEORI

2.1 Pengertian Robot

Istilah robot berasal dari bahasa *cheko* “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah bosan. Pada kamus *webster* pengertian robot adalah “*An automatic device that performs function ordinarily ascribed to human beings* (sebuah alat otomatis yang melakukan fungsi berdasarkan kebutuhan manusia)” [2]. Dari penjelasan tersebut disimpulkan bahwa robot adalah sistem mekanik untuk melakukan tugas fisik yang dapat dikontrol manusia ataupun diprogram yang telah dibuat pada sistem kontrolnya sesuai dengan kebutuhan manusia.

Pada dasarnya robot yang telah dibuat berfungsi untuk mempermudah dan membantu pekerjaan manusia. Berdasarkan rancangan dan kegunaannya, ada robot yang dirancang dalam bentuk sederhana sehingga fungsinya hanya bisa melakukan kegiatan yang sederhana, mudah dan berulang-ulang, ataupun robot yang dirancang secara khusus untuk melakukan pekerjaan yang sulit sehingga dapat berperilaku dengan kompleks dan bersifat otomatis mampu mengontrol dirinya sendiri hingga batasan tertentu. Secara umum robot dapat diklasifikasikan berdasarkan penggunaan aktuator, salah satunya adalah robot *mobile*. Robot *mobile* adalah robot yang dapat bergerak atau melakukan perpindahan posisi dari satu titik ke titik lainnya. Secara umum robot *mobile* memiliki dua jenis yaitu robot beroda dan robot berkaki. Robot beroda adalah jenis robot yang dapat berpindah tempat atau posisi dengan alat gerak utamanya menggunakan roda. Kelebihan dari robot beroda yaitu dapat bergerak atau berpindah posisi dengan cepat pada medan yang datar. Kekurangan robot beroda yaitu tidak dapat bergerak dengan cepat pada medan yang tidak rata seperti bebatuan dan permukaan tanah yang bergelombang. Jumlah dan jenis roda pada robot beroda sangat bervariasi tergantung dengan medan yang dilewati. Sedangkan robot berkaki hampir sama dengan robot beroda hanya saja alat penggerak utamanya adalah berupa kaki.

Kemajuan teknologi saat ini sangat berkembang pesat yaitu menciptakan sebuah inovasi yang menguntungkan bagi manusia salah satunya bidang pertanian. Dibidang pertanian sangat membutuhkan teknologi robot yang dapat membantu pekerjaan para petani. Berikut ada beberapa contoh robot pertanian, yaitu:

2.1.1 Robot Penanam Benih Jagung

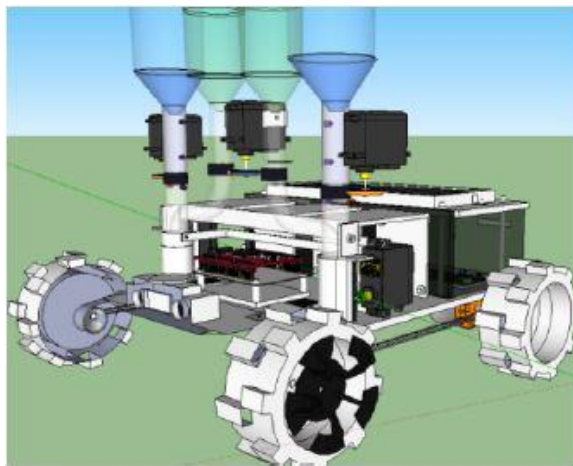
Robot penanam benih jagung memiliki sistem penanaman dengan menggunakan pengurutan dalam menanam benih sehingga petani hanya mengontrol alat ini ketika memulai titik awal penanaman dan ketika sampai titik akhir ujung lahan, begitu seterusnya. Robot ini dibuat untuk memudahkan petani jagung dalam penanaman benih jagung. Sistem kontrol robot ini menggunakan Arduino Uno sebagai *chip* utama dalam pengolah data dan sensor *photodiode* sebagai sensor pengikut garis (*line follower*) dan pendeteksi lahan yang akan ditanami. Sistem *line follower* ini mengendalikan robot untuk bergerak atau berjalan mengikuti garis sesuai dengan masukan intensitas cahaya sensor *photodiode*. Keluaran dari sistem kontrol robot berupa motor DC sebagai penggerak jalan robot, *LCD* sebagai penampil teks dan menu untuk mengisi perintah pada EPROM [3]. Bentuk robot penanam benih jagung dapat dilihat pada gambar 2.1 dibawah ini.



Gambar 2. 1 Robot Penanam Benih Jagung [3]

2.1.2 Robot Tanam Benih Langsung (TABELA)

Robot tanam benih langsung (TABELA) digunakan untuk menanam benih secara otomatis dengan jarak tanam antar benih yang diatur sesuai kebutuhan. Sistem kontrol robot ini menggunakan beberapa sensor seperti sensor infrared (SHARP GP12), sensor kompas HMC5883L, *optocoupler* dan *encoder* serta mikrokontroler *basic stamp* BS2P40 sebagai pengolah data. Sensor kompas digunakan sebagai penentu arah jalan robot, sensor infrared digunakan agar robot tidak berbenturan dengan penghalang yang berada di sekitar robot, *optocoupler* dan *encoder* sebagai penentu jarak tanam antar benih melalui pengaturan RPM pada motor DC. Sensor-sensor tersebut akan mengirim data ke mikrokontroler *basic stamp* BS2P40 dan data diolah untuk mengendalikan jalan robot baik berjalan lurus, berhenti dan berbelok dengan otomatis [4]. Adapun bentuk 3D dari robot tanam benih langsung dapat dilihat pada gambar 2.2.



Gambar 2. 2 3D Robot Tanam Benih Langsung [4]

2.1.3 Robot Penyiram Tanaman Otomatis

Pada sistem penyiraman tanaman otomatis menggunakan Arduino Uno sebagai pusat pengendali dan pengolah data dari sistem penyiraman otomatis, pemancar sinyal NRF24L01 sebagai pemancar dan penerima data kelembaban tanah yang dibaca dan diproses oleh sensor kelembaban tanah YL-69. Semua data kelembaban yang terbaca diolah Arduino Uno untuk menggerakkan motor DC

yang berfungsi mengendalikan jalan robot selama proses penyiraman yang dilakukan. Proses penyiraman yang dilakukan robot dimulai dari robot pada posisi awal dan bergerak secara bertahap sampai robot berada di tiap pot. Robot bergerak mengikuti garis menggunakan sensor *line follower* dan menuju ke tiap pot yang akan disiram dan proses penyiraman dilakukan sesuai data kelembaban tanah yang dikirim ke Arduino Uno [5]. Adapun bentuk robot penyiraman tanaman otomatis dapat dilihat pada gambar 2.3 dibawah ini.



Gambar 2. 3 Robot Penyiraman Tanaman Otomatis [5]

2.1.4 EcoRobotix

Robot EcoRobotix berfungsi untuk meminimalisir kegagalan panen melalui penyemprotan herbisida pada gulma yang terdeteksi. EcoRobotix ini memanfaatkan tenaga surya untuk sistem operasi sehingga mampu bekerja selama 12 jam secara terus menerus tanpa dikendalikan oleh operator manusia. Sistem jalan EcoRobotix menggunakan sensor kamera dan GPS sebagai sistem penglihatan yang digunakan untuk mengikuti baris tanaman, mendeteksi keberadaan gulma pada lahan pertanian dan memposisikan robot ke jalur yang dilewati. Setelah keberadaan gulma terdeteksi maka dua lengan robot akan bergerak untuk menyemprot herbisida dengan dosis kecil pada gulma tersebut. EcoRobotix sepenuhnya dapat dikontrol melalui aplikasi *smartphone* dengan

komunikasi menggunakan jaringan *smartphone* itu sendiri [6]. Adapun bentuk fisik dari ecoRobotix dapat dilihat pada gambar 2.4 seperti berikut.



Gambar 2. 4 EcoRobotix [6]

2.2 Penanaman

Penanaman adalah kegiatan memindahkan bibit dari tempat penyemaian ke lahan penanaman untuk di dapatkan hasil produk berkualitas dari tanaman yang di budidayakan. Ada beberapa hal yang harus diperhatikan sebelum proses penanaman, yaitu meliputi [7]:

1. Pembuatan peta areal kerja
2. Penataan areal
3. Penentuan jarak tanam
4. Penentuan arah larikan
5. Pembersihan lahan dari gulma
6. Pemasangan ajir
7. Pembuatan lubang tanam

2.2.1 Pembuatan Peta Areal Kerja

Kegiatan ini bertujuan untuk membuat peta areal kerja sesuai dengan luas lahan yang digunakan, jarak tanam antar tanaman yang diterapkan sehingga mengetahui jumlah kebutuhan bibit tanaman yang akan ditanam. Selain itu aspek seperti jenis tanaman yang akan ditanam dan pola tanam yang digunakan perlu diperhatikan untuk mempermudah proses pengerjaan ke lapangannya.

2.2.2 Penataan Areal

Kegiatan penataan areal tanam bertujuan untuk membuat batas-batas bagian lahan yang digunakan untuk mengetahui lokasi dan luas lahan yang digunakan melalui kegiatan pengukuran dan pemasangan patok serta anak patok pada lahan yang digunakan. Kegiatan ini menggunakan alat bantu seperti meteran untuk mengukur batas-batas lahan dan kayu/bambu sebagai penanda patokan.

2.2.3 Penentuan Jarak Tanam

Jarak tanam adalah suatu jarak antara tanaman satu dengan tanaman yang lainnya. Jarak tanam pada tanaman memiliki banyak variasi tergantung seberapa besar tubuh tanaman yang ditanam. Jarak tanam menentukan efisiensi pemanfaatan ruang tumbuh, mempermudah tindakan budidaya lainnya, tingkat dan jenis teknologi yang digunakan yang dapat ditentukan oleh jenis tanaman, kesuburan tanah, kelembaban tanah dan kemiringan lahan. Pengaturan jarak tanam sangat mendukung pertumbuhan tanaman dan produksi. Jarak tanam yang tidak tepat akan menimbulkan pengaruh negatif dan beberapa kerugian. Jarak tanam yang terlalu rapat menyebabkan pertumbuhan dahan terhambat sehingga mahkota pohon yang tidak rimbun. Jarak tanam yang terlalu rapat juga menyebabkan terjadinya persaingan terhadap unsur hara dan cahaya matahari tidak dapat diterima dengan baik oleh tanaman sehingga proses fotosintesis terhambat dan produksi buah tidak maksimal. Sedangkan jarak tanam yang terlalu lebar akan mengurangi jumlah populasi tanaman pada lahan yang digunakan [1]. Berikut contoh jarak tanam beberapa tanaman yang dapat dilihat pada tabel 2.1 dibawah ini.

Tabel 2. 1 Jarak Tanam Pada Tanaman

No.	Jenis Tanaman	Jarak Tanam
1	Kelapa Sawit	9 m x 9 m
2	Pohon Jati	3 m x 3 m
3	Mangga	10 m x 10 m / 12 m x 12 m
4	Lada	2,5 m x 2,5 m / 3 m x 3 m
5	Kakao	2 m x 2 m / 3 m x 3 m

2.2.4 Penentuan Arah Larikan

Penentuan arah larikan pada penyiapan lahan secara manual dilakukan sebelum kegiatan pembersihan lahan karena arah larikan berfungsi untuk membuat jalur tanaman pada saat akan dibersihkan berdasarkan kondisi lahan seperti topografi atau garis kontur. Penandaan arah larikan menggunakan alat bantu tali yang dipasang pada setiap jalur tanam dan ditarik lurus sampai ke ujung lahan. Untuk mempermudah pekerjaan, penentuan arah larikan dimulai dari batas kawasan areal tanaman, jalan hutan atau batas blok/petak. Selain itu penetapan arah larikan harus disesuaikan dengan pola dan jarak tanam yang akan diterapkan.

2.2.5 Pembersihan Lahan Dari Gulma

Kegiatan pembersihan ini bertujuan untuk menghentikan persaingan antara tanaman pertanian yang ditanam dengan gulma. Hal yang dilakukan adalah dengan melakukan pengolahan tanah dengan cara memotong perakaran gulma menggunakan alat bantu seperti parang, cangkul atau mencabut langsung gulma bersama akarnya dengan tangan. Cara umum lainnya dapat dilakukan dengan penyemprotan gulma menggunakan herbisida.

2.2.6 Pemasangan Ajir

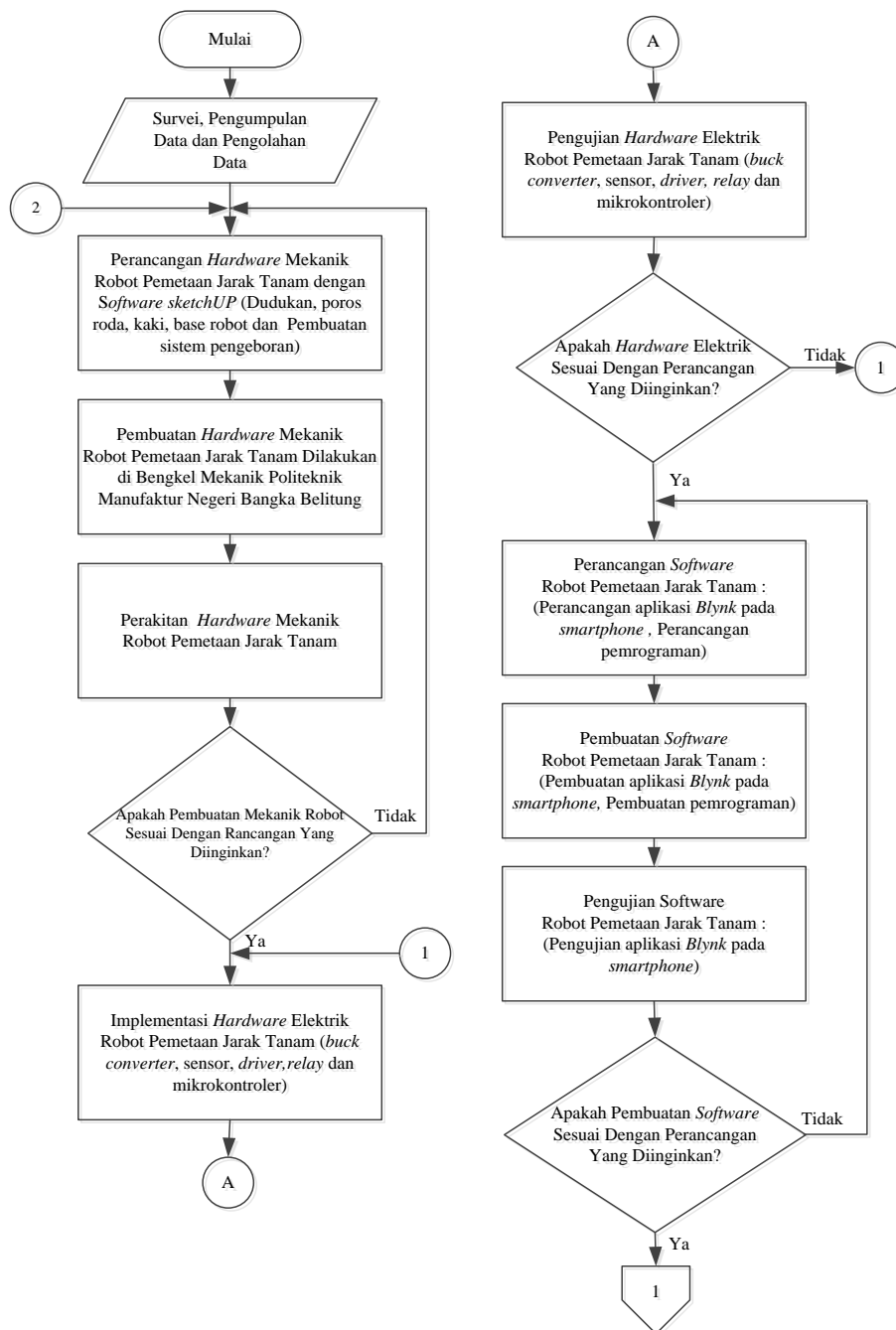
Pemasangan ajir dilakukan setelah pembersihan lahan dengan cara menarik tali dari arah larikan pertama ke arah sejajar dan mengikuti jarak tanam yang telah ditetapkan pada peta areal yang dibuat. Pemasangan ajir bertujuan sebagai penanda titik tanam bahwa tempat tersebut akan dibuat lubang tanam sesuai dengan jarak tanam yang ditentukan.

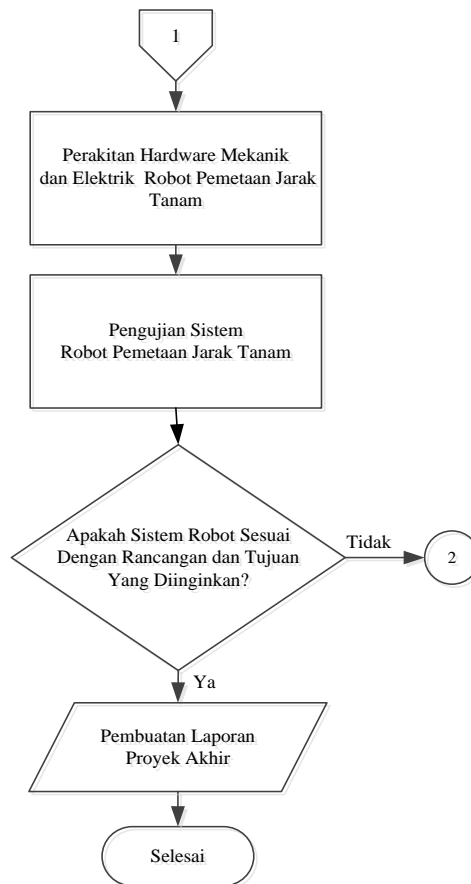
2.2.7 Pembuatan Lubang Tanam

Lubang tanam adalah tempat yang ditentukan sebagai penanaman bibit. Pembuatan lubang tanam dilakukan dengan mencangkul bagian yang sudah ditandai melalui pemasangan ajir sebelumnya. Untuk besar, kecil serta dalamnya lubang tanam yang dibuat tergantung pada jenis tanaman yang akan ditanam.

BAB III METODE PELAKSANAAN

Untuk mempermudah proses pembuatan proyek akhir, dibuat beberapa tahapan dalam bentuk *flowchart* yang dapat dilihat pada gambar 3.1 berikut ini:





Gambar 3. 1 *Flowchart* Metode Pelaksanaan Proyek Akhir

3.1 Survei, Pengumpulan Data dan Pengolahan Data

Survei, pengumpulan data dalam proyek akhir ini dilakukan dengan mencari informasi tentang data-data sumber referensi yang diperlukan untuk mengetahui dan mengatasi permasalahan yang terjadi. Pada tahap ini meliputi beberapa aspek sumber data, yaitu data primer dan data sekunder. Data primer diperoleh dari survei kelapangan dan wawancara serta konsultasi dengan dosen pembimbing. Hasil wawancara dan survei yang dilakukan di Desa Kimak Kecamatan Merawang dengan menanyakan ke beberapa petani lada bahwa proses pemetaan jarak tanam masih menggunakan cara manual yaitu menggunakan meteran untuk mengukur jarak tanam dan menandai lubang tanam dengan kayu runcing ataupun menggunakan tali sudah diukur panjangnya sesuai dengan jarak tanam tanaman yang masing-masing ujungnya diikat pada kayu runcing.

Sedangkan data sekunder diperoleh dari referensi buku, jurnal, makalah dan internet tentang proses pemetaan jarak tanam. Salah satu jurnal yang dijadikan referensi adalah “Rancang Bangun Robot Penanam Benih Jagung Otomatis Berbasis Mikrokontroller”.

Setelah data-data dikumpulkan selanjutnya data tersebut diolah menjadi referensi dan acuan dalam pembuatan proyek akhir. Dari proses pengolahan data yang dilakukan akan dirancang sebuah alat yang dapat melakukan pemetaan jarak tanam sehingga mengatasi permasalahan yang ada dalam proses pemetaan jarak tanam sekarang ini.

3.2 Perancangan *Hardware* Mekanik Robot

Perancangan *hardware* mekanik robot pemetaan jarak tanam dilakukan dengan *software SolidWorks* 2011 yang meliputi perancangan dudukan dan poros roda, *base*, kaki serta sistem pengeboran robot pemetaan jarak tanam yang akan dibuat dan penentuan dimensi/ukuran serta bahan material pada setiap bagian sistem mekaniknya.

3.3 Pembuatan *Hardware* Mekanik Robot

Pembuatan *hardware* mekanik dilakukan di bengkel mekanik Politeknik Manufaktur Negeri Bangka Belitung, dengan membuat konstruksi pada robot meliputi pembuatan pada bagian-bagian sistem mekanik dari robot pemetaan jarak tanam. Pembuatan *hardware* mekanik robot pemetaan jarak tanam yang akan dibuat meliputi dudukan dan poros roda, kaki, *base*, dan sistem pengeboran robot pemetaan jarak tanam.

3.4 Perakitan *Hardware* Mekanik Robot

Perakitan *hardware* mekanik robot pemetaan jarak tanam dilakukan dengan cara merakit pada masing-masing dari setiap bagian robot pemetaan jarak tanam yang telah dibuat meliputi dudukan dan poros roda, *base*, kaki depan dan belakang serta sistem pengeboran robot pemetaan jarak tanam.

3.5 Implementasi *Hardware* Elektrik Robot

Implementasi *hardware* elektrik robot pemetaan jarak tanam dilakukan dengan menentukan komponen elektrik yang akan digunakan seperti *buck converter*, sensor-sensor, *driver*, *relay* dan mikrokontroler. Komponen elektrik dilakukan dengan pembelian komponen yang telah jadi dan siap dipakai sehingga tanpa harus membuat.

3.6 Pengujian *Hardware* Elektrik Robot

Pengujian dilakukan dengan menguji komponen elektrik untuk mengetahui apakah komponen yang dipakai bekerja sesuai dengan fungsi yang diinginkan dan dibutuhkan. Uji coba *hardware* pada bagian elektrik diantaranya:

1. Uji coba *buck converter step down* dan *breadboard power supply*
2. Uji coba Arduino Uno
3. Uji coba sensor tegangan DC, sensor *rotary encoder* dan sensor kompas
4. Uji coba NodeMCU ESP8266
5. Uji coba *driver*
6. Uji coba *relay*

3.7 Perancangan *Software* Robot

Perancangan *software* dilakukan dengan membuat perancangan berupa aplikasi pada *smartphone* dan perancangan pemrograman. Perancangan aplikasi pada *smartphone* yaitu dengan menggunakan aplikasi *blynk* untuk mengontrol sistem robot pemetaan jarak tanam dan perancangan pemrograman dilakukan sebagai pengolahan data sesuai dengan yang diinginkan.

3.8 Pembuatan *Software* Robot

Pembuatan *software* yang akan dilakukan meliputi :

1. Pembuatan aplikasi pada *smartphone* dengan menggunakan aplikasi *blynk*.
2. Pembuatan program aplikasi komunikasi antara *smartphone* dengan sistem kontrol.

3.9 Pengujian *Software* Robot

Pengujian dilakukan dengan menguji aplikasi *blynk* pada *smartphone* yang telah dibuat dengan cara memasukkan data.

3.10 Perakitan *Hardware* Mekanik Dan Elektrik Robot

Perakitan *hardware* mekanik dan elektrik robot pemetaan jarak tanam dilakukan dengan cara merakit secara keseluruhan pada masing-masing dari setiap bagian *hardware* mekanik dan elektrik robot pemetaan jarak tanam yang telah dibuat menjadi satu kesatuan.

3.11 Pengujian Sistem Robot Pemetaan Jarak Tanam

Pengujian sistem robot pemetaan jarak tanam dilakukan secara keseluruhan untuk mengetahui hasil akhir secara keseluruhan pada sistem yang dibuat, diantaranya:

1. Uji coba memasukkan dan penerimaan data robot pemetaan jarak tanam melalui *smartphone*.
2. Uji coba proses pemetaan jarak tanam oleh robot berupa koordinat pengeboran titik tanam dan keakurasian jarak tanam.
3. Uji coba *database*.

3.12 Pembuatan Laporan Proyek Akhir

Pada tahap ini, semua kegiatan dalam pembuatan proyek akhir dilaporkan dalam bentuk makalah proyek akhir. Makalah ini dibuat berdasarkan format yang sudah ditentukan, berisikan mulai dari latar belakang, landasan teori, konsep perancangan *hardware* mekanik, *hardware* elektrik, dan *software* aplikasi untuk *smartphone* serta pembuatan alat, pengujian alat, dan analisa hasil pengujian sistem alat berupa tingkat keberhasilan secara keseluruhan.

BAB IV

PEMBAHASAN

Pada bab ini akan membahas bagaimana membuat robot pemetaan jarak tanam berbasis *smartphone*. Sebelum melakukan perancangan dan pembuatan pada robot pemetaan jarak tanam berbasis *smartphone* ini, ada beberapa hal yang harus diperhatikan yaitu, perancangan *hardware* mekanik, implementasi *hardware* elektrik, pembuatan *hardware* mekanik dan pengujian *hardware* elektrik pada robot pemetaan jarak tanam. Perancangan, pembuatan dan pengujian *software* robot pemetaan jarak tanam, pengujian sistem robot pemetaan jarak tanam secara keseluruhan serta pembuatan laporan.

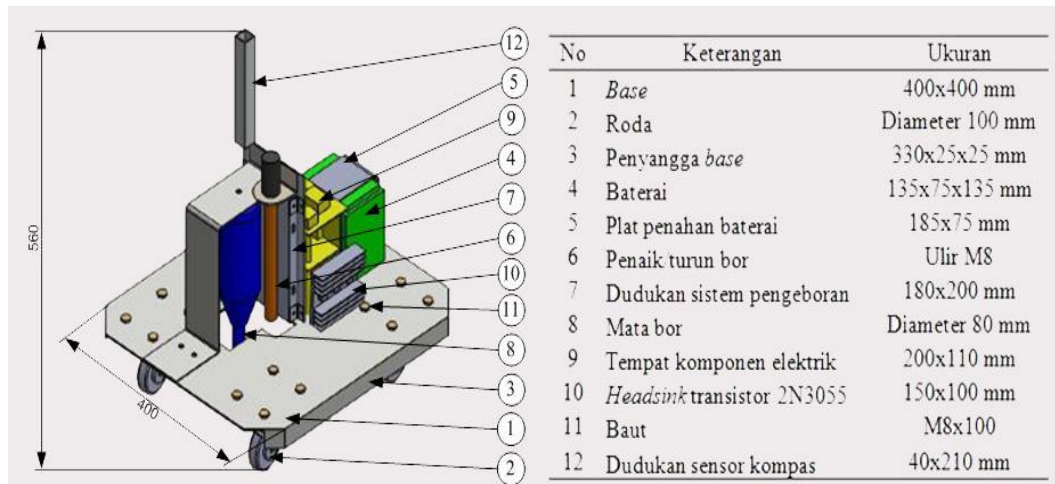
4.1 Perancangan *Hardware* Robot Pemetaan Jarak Tanam

Perancangan *hardware* robot pemetaan jarak tanam ini bertujuan untuk menentukan deskripsi kerja dari alat yang akan digunakan dan dibuat, seperti membuat perancangan *hardware* mekanik berupa konstruksi pada bagian keseluruhan robot seperti pemilihan bahan, menentukan ukuran sehingga dapat meminimalisir kesalahan dalam proses pembuatan alat agar alat yang dihasilkan sesuai dengan apa yang telah direncanakan. Sedangkan untuk perancangan *hardware* elektrik dengan menentukan komponen elektrik yang akan digunakan dalam proses pembuatan proyek akhir seperti *buck converter*, sensor-sensor, *driver* dan mikrokontroler.

4.1.1 Perancangan *Hardware* Mekanik Robot Pemetaan Jarak Tanam

Perancangan *hardware* mekanik robot pemetaan jarak tanam adalah suatu proses perancangan bagaimana membuat konstruksi yang akan dibuat dan dari *hardware* mekanik robot pemetaan jarak tanam. Untuk membuat perancangan desain *hardware* mekanik ini dilakukan melalui *PC (Personal Computer)* menggunakan *software SolidWorks 2011*. Adapun hasil perancangan robot

pemetaan jarak tanam menggunakan *software SolidWorks 2011* pada gambar 4.1 berikut.



Gambar 4. 1 Perancangan Robot Pemetaan Jarak Tanam

Perancangan konstruksi dirancang dengan menentukan ukuran dan bahan. Adapun ukuran/dimensi dari robot pemetaan jarak tanam yang akan dibuat yaitu dengan ukuran/dimensi 400x400x560 mm. Desain robot pemetaan jarak tanam yang dibuat dapat dilihat lebih jelas pada lampiran 2. Untuk perancangan bahan yang akan digunakan adalah plat aluminium, plat *hollow* dan baja ST-37. Plat aluminium ini berguna untuk penempatan berupa komponen elektrik, plat *hollow* digunakan untuk kaki robot dan baja ST-37 untuk pembuatan poros roda. Pada tahap pembuatan perancangan *hardware* mekanik dibuat secara bertahap dimulai dari perancangan pada masing-masing bagian robot pemetaan jarak tanam. Tahap yang perlu dilakukan adalah menentukan bentuk *base*, dudukan dan poros roda, kaki robot serta sistem pengeboran yang akan digunakan dalam pembuatan *hardware* mekanik robot. Berdasarkan penempatan komponen elektrik dan gerak robot pemetaan jarak tanam maka ditentukan bentuk *base* yang sesuai adalah berbentuk persegi. Setelah masing-masing bagian dari robot pemetaan jarak tanam selesai dibuat selanjutnya adalah tahap perakitan masing-masing dari setiap bagian robot pemetaan jarak tanam menjadi satu kesatuan.

4.1.2 Pembuatan Hardware Mekanik Robot Pemetaan Jarak Tanam

Proses pembuatan *hardware* mekanik robot pemetaan jarak tanam dilakukan di bengkel mekanik Politeknik Manufaktur Negeri Bangka Belitung, berikut beberapa tahapan yang akan dibuat meliputi:

A. Pembuatan Dudukan dan Poros Roda

Proses pembuatan dudukan diperlukan sebagai tempat dudukan kaki dan roda robot, untuk bahan yang akan digunakan dalam pembuatan dudukan ini adalah bahan yang terbuat dari bahan yang kuat agar bisa menahan beban yang berlebih, sehingga bahan yang dipilih untuk dudukan adalah plat *hollow* 25 mm dengan ukuran panjang 165 mm dan lebar 130 mm.

Proses pembuatan poros roda diperlukan untuk menggerakkan roda robot pemetaan jarak tanam dan sebagai tempat penghubung dengan motor DC *power window* serta untuk penempatan sensor *rotary encoder* KY-040. Untuk pemilihan bahan yang dipilih adalah bahan yang kuat dan mudah dilas, sehingga mempermudah untuk menghubungkan poros dengan motor DC *power window* KY-040. Bahan yang digunakan dalam pembuatan poros roda ini adalah baja ST-37 dengan ukuran panjang 120 mm dengan diameter 100 mm. Berikut gambar 4.2 hasil pembuatan dudukan dan poros roda yang telah terpasang dengan roda dan motor DC *power window*.



Gambar 4. 2 Dudukan dan Poros Roda

B. Pembuatan Kaki Robot

Pembuatan kaki roda dilakukan untuk penghubung antara dudukan roda dan *base* pada robot pemetaan jarak tanam, pemilihan material untuk kaki robot belakang sama dengan bahan untuk dudukan roda yaitu plat *hollow* 25 mm. Kaki robot terdiri dari satu pasang kaki depan dan kaki belakang dengan ukuran kaki depan menggunakan baut ke *base* robot 50 mm sedangkan kaki belakang berukuran 25x85 mm. Berikut tampak gambar 4.3 hasil pembuatan kaki robot.



Gambar 4. 3 Kaki Robot

C. Pembuatan *Base* Robot

Proses pembuatan *base* yaitu dengan pemilihan bahan material yang akan digunakan sesuai dengan perancangan. *Base* robot dibuat dengan bahan plat aluminium dengan ketebalan 3 mm dan berbentuk persegi dengan ukuran setiap sisinya 400 mm dan proses pembuatannya dilakukan dengan memotong plat aluminium tersebut berbentuk persegi dengan ukuran 400x400 mm. Berikut gambar 4.4 Pembuatan *base* robot pemetaan jarak tanam yang telah dibuat.



Gambar 4. 4 *Base* Robot

D. Pembuatan Sistem Pengeboran

Sistem pengeboran robot pemetaan jarak tanam bertujuan untuk menandai koordinat titik tanam yang diinginkan. Pemilihan bahan material yang akan digunakan untuk dudukan sistem pengeboran yaitu memakai plat aluminium adapun ukuran untuk dudukan penempatan sistem pengeboran yaitu 180 x 200 mm. Pembuatan dudukan untuk pengeborannya dilakukan dengan membeli bor tangan bekas yang dijual dipasaran. Dudukan pada sistem pengeboran berfungsi untuk penempatan dudukan motor DC 1 untuk pengeboran dan motor DC 2 yang berfungsi untuk menggerakkan ulir untuk menaik dan menurunkan bor. Adapun hasil pembuatan sistem pengeboran dapat dilihat pada gambar 4.5 berikut.

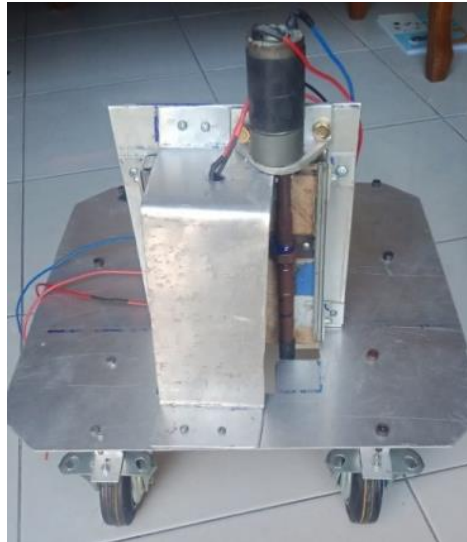


Gambar 4. 5 Sistem Pengeboran

4.1.3 Perakitan *Hardware* Mekanik Robot Pemetaan Jarak Tanam

Setelah pembuatan *hardware* mekanik robot pemetaan jarak tanam selesai dibuat selanjutnya akan dilakukan perakitan atau pemasangan konstruksi pada bagian dari masing-masing *hardware* mekanik seperti dudukan dan poros roda, base, kaki serta sistem pengeboran menjadi satu kesatuan yang utuh. Perakitan pada bagian dari masing-masing *hardware* mekanik robot pemetaan jarak tanam dilakukan dengan pemasangan baut dan mur yang berguna untuk merakit seluruh bagian dari masing-masing *hardware* mekanik robot pemetaan jarak tanam.

Berikut adalah gambar 4.6 hasil dari perakitan *hardware* mekanik robot pemetaan jarak tanam.

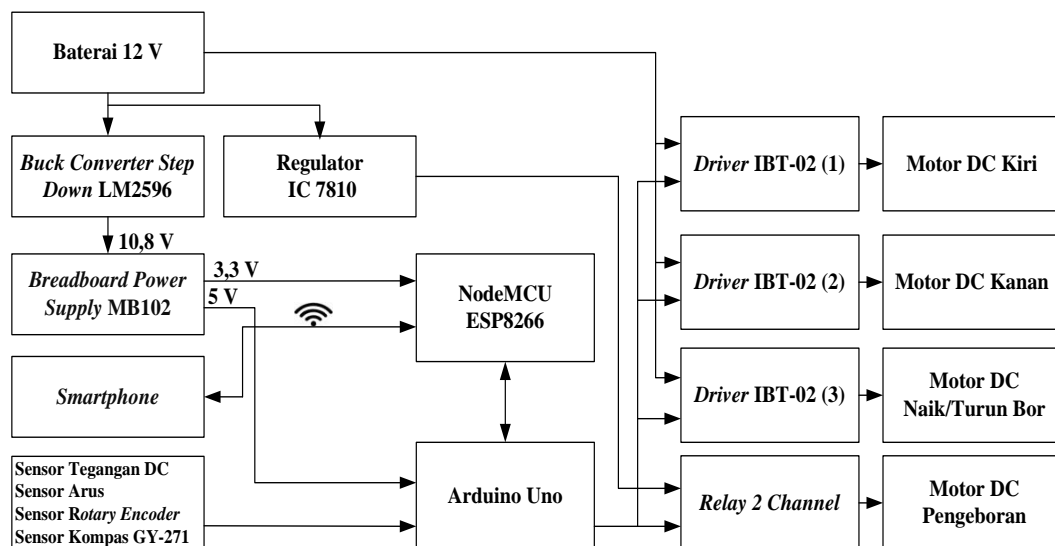


Gambar 4. 6 *Hardware* Mekanik Robot Pemetaan Jarak Tanam

4.2 Implementasi *Hardware* Elektrik Robot Pemetaan Jarak Tanam

4.2.1 Diagram Blok

Diagram blok sistem kontrol dari robot pemetaan jarak tanam berbasis *smartphone* ditunjukkan pada gambar 4.7 berikut.



Gambar 4. 7 Diagram Blok Sistem Kontrol Robot Pemetaan Jarak Tanam

Dari gambar 4.7 diagram blok sistem kontrol robot pemetaan jarak tanam diatas, baterai 12 V digunakan sebagai *supply* tegangan untuk mikrokontroller Arduino Uno, NodeMCU ESP8266, sensor-sensor, *driver* IBT-02 dan *relay*. *Buck converter step down* LM2596 berfungsi untuk menurunkan tegangan dari baterai 12 V menjadi 10,8 V sebagai *supply* tegangan *breadboard power supply* MB102. *Buck converter* tipe MB102 menurunkan tegangan dari 10,8 V menjadi 3,3 V sebagai *supply* tegangan NodeMCU ESP8266 dan 5 V sebagai *supply* tegangan Arduino Uno. Regulator IC 7810 berfungsi sebagai *supply* tegangan *relay* untuk menggerakkan motor DC sistem pengeboran. *Smartphone* digunakan untuk *input* data berupa nilai sudut, luas lahan dan jarak tanam, proses pengiriman data dari *smartphone* ke Arduino Uno menggunakan modul *wifi* NodeMCU ESP8266, setelah proses pengiriman data selesai, data diolah oleh Arduino Uno dan dikirim ke *smartphone* melalui modul *wifi* NodeMCU ESP8266 untuk mengetahui nilai dari jumlah tanaman, titik tanam dan daya yang digunakan. Selanjutnya sensor tegangan DC dan sensor arus digunakan untuk mengukur *supply* tegangan *input* dan mengukur arus dari rangkaian sistem elektrik serta sensor *rotary encoder* KY-040 sebagai pengatur jarak tempuh robot pemetaan jarak tanam. Sensor kompas GY-271 digunakan sebagai arah untuk jalan robot pemetaan jarak tanam. Setelah semua data yang telah diproses oleh Arduino Uno maka *output* Arduino Uno akan mengontrol *driver* IBT-02 (1) dan *driver* IBT-02 (2) berguna untuk mengontrol motor DC kiri dan motor DC kanan untuk membuat robot berjalan serta *relay* dan *driver* IBT-02 (3) untuk naik/turun bor.

4.2.2 Buck Converter

A. Implementasi Buck Converter

Buck coverter yang akan digunakan dalam proyek akhir ini adalah *buck converter* tipe *step down* LM2596 dan *power supply* MB102. *Buck converter* tipe *step down* LM2596 ini sebagai penurun tegangan dari *supply* baterai 12 V menjadi 10,8 V. *Buck converter* tipe *breadboard power supply* MB102 ini akan menurunkan tegangan *output* dari 10,8 V menjadi 5 V dan 3,3 V masing-masing untuk *supply* tegangan Arduino Uno dan NodeMCU ESP8266. Regulator IC 7810

digunakan sebagai penurun tegangan dari baterai 12 V untuk motor DC pengeboran tegangan *output* yang diinginkan sebesar 10 V. Modul *buck converter step down* LM2596 dan *breadboard power supply* MB102 dilakukan dengan pembelian modul yang telah jadi dan siap dipakai. Sedangkan regulator IC 7810 dilakukan dengan pembuatan berdasarkan perancangan dari rangkaian regulator IC 7810 dengan membeli satu buah IC 7810 dan tiga buah transistor tipe 2N3055 yang kemudian dirangkai sesuai dengan perancangan. Berikut adalah pemasangan modul *buck converter step down* LM2596 pada *board* sistem kontrol ditunjukkan pada gambar 4.8 seperti berikut.



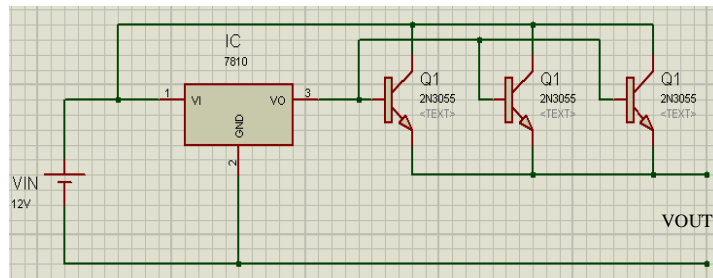
Gambar 4. 8 Modul *Buck Converter Step Down* LM2596

Adapun bentuk fisik dari modul *breadboard power supply* MB102 yang telah terpasang pada papan tempat komponen elektrik, ditunjukkan pada gambar 4.9 seperti berikut.



Gambar 4. 9 Modul *Breadboard Power Supply* MB102

Perancangan regulator IC 7810 dapat dilihat pada gambar 4.10 sebagai berikut.



Gambar 4. 10 Skematik Regulator IC 7810

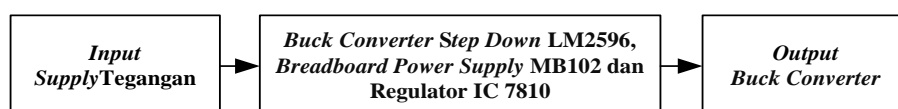
Berikut adalah hasil pembuatan rangkaian regulator IC 7810 sesuai dengan perancangan, yang dapat dilihat pada gambar 4.11.



Gambar 4. 11 Hasil Pembuatan Regulator IC 7810

B. Pengujian *Buck Converter*

Pengujian *buck converter* dilakukan dengan mengukur *output* dari *buck converter step down* LM2596, *breadboard power supply* MB102 dan regulator IC 7810 menggunakan multimeter, hal ini berfungsi untuk mengetahui apakah hasil sesuai dengan yang diinginkan atau tidak. Berikut ini gambar 4.12 blok pengujian *buck converter step down* LM2596 *breadboard power supply* MB102 dan regulator IC 7810.



Gambar 4. 12 Blok Pengujian *Buck Converter*

Gambar 4.13 dibawah ini menunjukkan hasil pengujian *buck converter step down* LM2596.



a) Pengujian Pertama



b) Pengujian Kedua

Gambar 4. 13 Pengujian *Buck Converter Step Down* LM2596

Berdasarkan gambar 4.13 didapatkan hasil pengujian *buck converter step down* LM2596 terhadap alat ukur dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4. 1 Hasil Pengujian *Buck Converter Step Down* LM2596

No.	Vin (V)	Vout diinginkan (V)	Vout terukur (V)	Error (%)
1	12	10,8	10,78	0,18
2	12	10,8	10,79	0,09
3	12	10,8	10,8	0
Rata-rata error				0,09

Setelah melakukan pengujian dilakukan perbandingan antara tegangan yang diinginkan dengan tegangan yang terukur melalui rumus (4.1) dibawah ini.

$$error = \left| \frac{Tegangan\ diinginkan - Tegangan\ terukur}{Tegangan\ diinginkan} \right| \cdot 100\% \dots\dots\dots (4.1)$$

$$error = \left| \frac{10,8 - 10,78}{10,8} \right| \cdot 100\% = 0,18\%$$

$$error = \left| \frac{10,8 - 10,8}{10,8} \right| \cdot 100\% = 0\%$$

Gambar 4.14 dibawah ini menunjukkan hasil pengujian *breadboard power supply* MB102.



a) Pengujian MB102 5 V b) Pengujian MB102 3,3 V

Gambar 4. 14 Pengujian *Breadboard Power Supply* MB102

Berdasarkan gambar 4.14 didapatkan hasil pengujian *breadboard power supply* MB102 terhadap alat ukur dapat dilihat pada tabel 4.2 dibawah ini.

Tabel 4. 2 Hasil Pengujian *Breadboard Power Supply* MB102

No.	Vin (V)	Vout diinginkan (V)	Vout terukur (V)	Error (%)
1	10,8	5	4,93	1,4
2	10,8	5	4,93	1,4
3	10,8	3,3	3,24	1,8
4	10,8	3,3	3,24	1,8
Rata-rata <i>error</i>				1,6

Setelah melakukan pengujian dilakukan perbandingan antara tegangan yang diinginkan dengan tegangan yang terukur menggunakan rumus (4.1).

$$error = \left| \frac{5 - 4,93}{5} \right| \cdot 100\% = 1,4\%$$

$$error = \left| \frac{3,3 - 3,24}{3,3} \right| \cdot 100\% = 1,8\%$$

Gambar 4.15 dibawah ini menunjukkan hasil pengujian rangkaian regulator IC 7810 yang telah dibuat.



Gambar 4. 15 Pengujian Regulator IC 7810

Berdasarkan gambar 4.15 didapatkan hasil pengujian rangkaian regulator IC 7810 terhadap alat ukur dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4. 3 Hasil Pengujian Regulator IC 7810

No.	V _{in}	V _{out} diinginkan (V)	V _{out} terukur (V)	Error (%)
1	12	10	9,59	4,1
2	12	10	9,59	4,1
Rata-rata <i>error</i>				4,1

Setelah melakukan pengujian dilakukan perbandingan antara tegangan yang diinginkan dengan tegangan yang terukur menggunakan rumus (4.1).

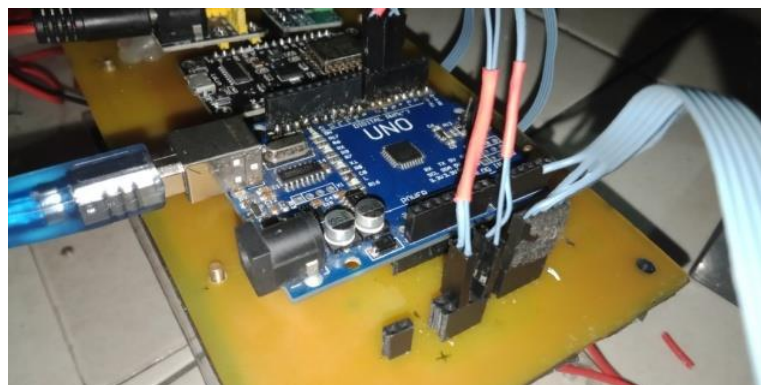
$$error = \left| \frac{10 - 9,59}{10} \right| \cdot 100\% = 4,1\%$$

Dari tabel 4.1, 4.2 dan 4.3 hasil pengujian *buck converter step down* LM2596 didapat hasil rata-rata presentase *error* yaitu sebesar 0,09% *breadboard power supply* MB102 hasil presentase *error* yaitu sebesar 1,6% sedangkan untuk dan regulator IC 7810 sebesar 4,1%. Maka dari hasil pengujian *buck converter* dapat disimpulkan bahwa *buck converter step down* LM2596, *breadboard power supply* MB102 dan regulator IC 7810 masih bisa dipakai dan masih bekerja dengan baik, sesuai yang diinginkan untuk pengerjaan dalam proyek akhir.

4.2.3 Arduino Uno

A. Implementasi Arduino Uno

Arduino Uno berfungsi untuk membuat sistem kontrol dalam mengendalikan komponen elektrik yang digunakan dalam proyek akhir seperti sensor tegangan DC, sensor arus, sensor *rotary encoder* KY-040, NodeMCU ESP8266, *driver* IBT-02 dan *relay*, sehingga memudahkan dalam melakukan pemrograman pada Arduino Uno. Arduino Uno memiliki 14 pin *input/output* yang berfungsi sebagai *input/output* yang dapat diatur dalam program. Pada proyek akhir ini dilakukan pembelian modul Arduino Uno yang sudah jadi dan siap dipakai. Berikut adalah gambar 4.16 Arduino Uno yang telah dibeli.

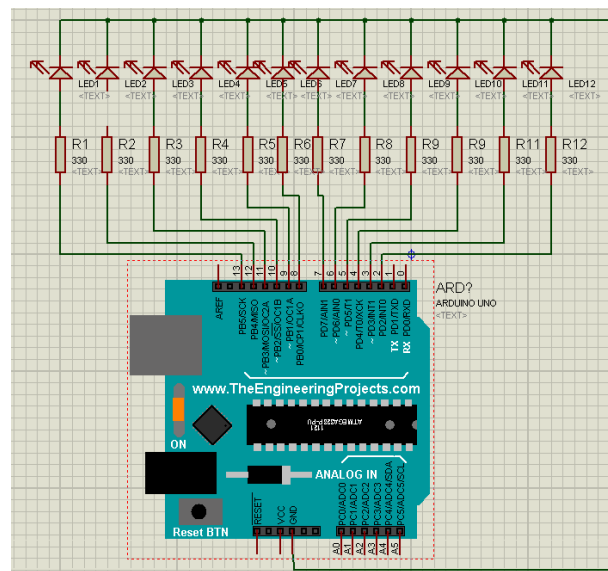


Gambar 4. 16 Arduino Uno

B. Pengujian Arduino Uno

Pengujian Arduino Uno dilakukan dengan membuat program pada komputer/laptop menggunakan *software* Arduino IDE, yaitu dengan pengujian

PORT Arduino Uno pada COM yang ada di *device manager* PC. Pengecekan pin Arduino Uno yang akan dilakukan adalah dengan memberi *input* untuk melihat *output* dari pin Arduino Uno dengan cara dihubungkan ke LED, hal ini bertujuan untuk pengecekan pin Arduino Uno apakah sesuai dengan perancangan. Berikut adalah gambar 4.17 skematik pengujian Arduino Uno.



Gambar 4. 17 Skematik Pengujian Arduino Uno

Berikut adalah *list* program untuk pengujian pin-pin *output* pada Arduino Uno berdasarkan rangkaian pengujian yang terdapat pada gambar 4.17 diatas, dapat dilihat list program Arduino Uno dibawah ini.

```
//Arduino Uno
void loop()
{
  //LED Menyala
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);
  delay(1000);
  //LED Mati
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
  delay(1000);
}
```


Setelah melakukan pengujian terhadap pin *output* Arduino Uno didapatkan hasil pengujian dapat dilihat pada tabel 4.4 dibawah ini.

Tabel 4. 4 Hasil Pengujian Pin *Output* Arduino Uno

No.	<i>Input</i> (V)	Pin <i>output</i> yang diuji	Hasil pengujian LED	Keterangan
1	5	Pin 2	1	Hidup
2		Pin 3	1	Hidup
3		Pin 4	1	Hidup
4		Pin 5	1	Hidup
5		Pin 6	1	Hidup
6		Pin 7	1	Hidup
7		Pin 8	1	Hidup
8		Pin 9	0	Mati
9		Pin 10	1	Hidup
10		Pin 11	1	Hidup
11		Pin 12	1	Hidup
12		Pin 13	1	Hidup

Dari tabel 4.4 hasil pengujian pin Arduino Uno dapat dianalisa bahwa modul Arduino Uno yang akan digunakan sebagai sistem kontrol dapat berfungsi dengan baik dan sesuai dengan perancangan yang diinginkan kecuali pin 9 yang rusak atau tidak bisa digunakan, tetapi modul Arduino Uno masih tetap bisa digunakan. Berdasarkan perancangan penggunaan pin-pin yang dipakai Arduino Uno sebagai sistem kontrol dari komponen-komponen elektrik pada proyek akhir, maka Arduino Uno bisa digunakan dikarenakan jumlah pin-pin yang dirancang cukup untuk koneksi ke komponen-komponen elektrik.

4.2.4 Sensor Tegangan DC

A. Implementasi Sensor Tegangan DC

Sensor tegangan DC berfungsi untuk pembacaan nilai tegangan dari rangkaian sistem kontrol robot pemetaan jarak tanam dengan mengukur nilai tegangan ukurnya melalui pengukuran tegangan *input* yang diberikan sebesar 12 VDC dengan tegangan maksimal sensor yang terbaca 0-25 VDC. Pada proyek

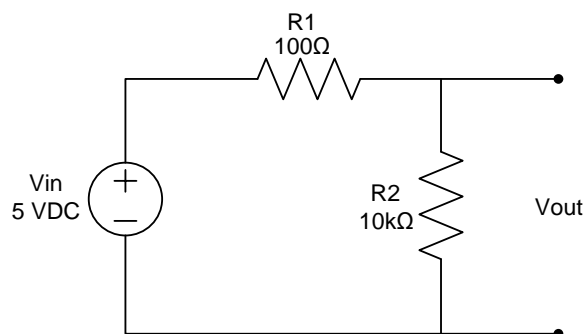
akhir ini dilakukan dengan pembelian sensor tegangan DC yang sudah jadi dan siap dipakai. Adapun pemasangan modul sensor tegangan DC ditunjukkan pada gambar 4.18 berikut.



Gambar 4. 18 Sensor Tegangan DC

B. Pengujian Sensor Tegangan DC

Pengujian berfungsi untuk mengetahui nilai tegangan *input* yang terbaca oleh sensor tegangan DC, pengujian dilakukan dengan menghubungkan pin VCC ke terminal positif *power supply* dengan tegangan *input* 5 V, GND dihubungkan ke terminal negatif *power supply*, S dihubungkan ke input analog dari Arduino Uno pin A1, “+” dihubungkan ke 5 V Arduino Uno dan “-“ dihubungkan ke GND dari Arduino Uno. Kemudian dilakukan dengan membuat program menampilkan nilai tegangan *input* yang terbaca oleh sensor tegangan DC pada *software* Arduino IDE. Adapun gambar 4.19 rangkaian pengujian sensor tegangan DC, rangkaian pengujian digunakan untuk membandingkan nilai dari sensor tegangan DC dengan perhitungan.



Gambar 4. 19 Rangkaian Pengujian Sensor Tegangan DC

Berikut adalah *list* program untuk pengujian pembacaan nilai dari sensor tegangan DC.

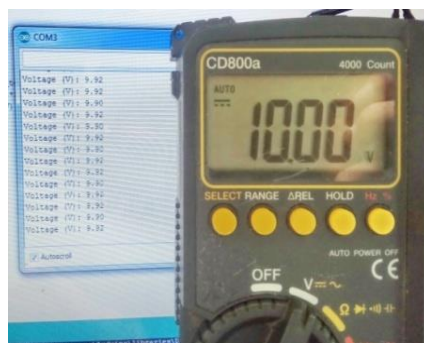
```
void loop()  
{  
  Volt1=analogRead(sensor_tegangan); //pembacaan bentuk desimal  
  tegangan=((Volt1*0.0049)*5); //rumus sensor dalam volt  
  Serial.print("Voltage (V): ");  
  Serial.println(tegangan); //menampilkan tegangan diserial monitor  
  delay(500);  
}
```

Gambar 4.20 dibawah ini menunjukkan hasil pengujian 1 sensor tegangan DC terhadap alat ukur dengan mengukur *input* tegangan sebesar 5V.



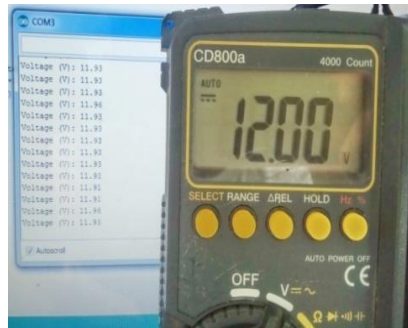
Gambar 4. 20 Hasil Pengujian 1 Sensor Tegangan DC Dengan Alat Ukur

Gambar 4.21 dibawah ini menunjukkan hasil pengujian 2 sensor tegangan DC terhadap alat ukur dengan mengukur *input* tegangan sebesar 10 V.



Gambar 4. 21 Hasil Pengujian 2 Sensor Tegangan DC Dengan Alat Ukur

Gambar 4.22 dibawah ini menunjukkan hasil pengujian 3 sensor tegangan DC terhadap alat ukur dengan mengukur *input* tegangan sebesar 12 V.



Gambar 4. 22 Hasil Pengujian 3 Sensor Tegangan DC Dengan Alat Ukur

Setelah melakukan pengujian terhadap sensor tegangan DC terhadap alat ukur didapatkan hasil pengujian pada tabel 4.5 dibawah ini.

Tabel 4. 5 Hasil Pengujian Sensor Tegangan DC

No.	Resistansi (Ω)		Vin (Volt)	Vout (Volt)			Error (%)
	R1	R2		Sensor	Multimeter	Perhitungan	
1	100	10.000	5	4,95	5,02	4,95	1,3
2	100	10.000	10	9,92	10	9,90	0,8
3	100	10.000	12	11,96	12	11,88	0,33
Rata-rata <i>error</i> (%)							0,81

Adapun hasil perhitungan secara teori dari sensor tegangan DC dapat dirumuskan sebagai berikut:

$$V_{out} = \frac{R_2}{R_1+R_2} \cdot V_{in} \dots\dots\dots (4.2)$$

1. Perhitungan Secara Teori

$$V_{out} = \frac{R_2}{R_1+R_2} \cdot V_{in}$$

$$V_{out} = \frac{10.000\Omega}{100\Omega + 10.000\Omega} \cdot 5$$

$$V_{out} = 4,95 \text{ Volt}$$

2. Perhitungan Persentase *Error*

$$error = \left| \frac{5,02 - 4,95}{5,02} \right| \cdot 100\% = 1,3\%$$

$$error = \left| \frac{10 - 9,92}{10} \right| \cdot 100\% = 0,8\%$$

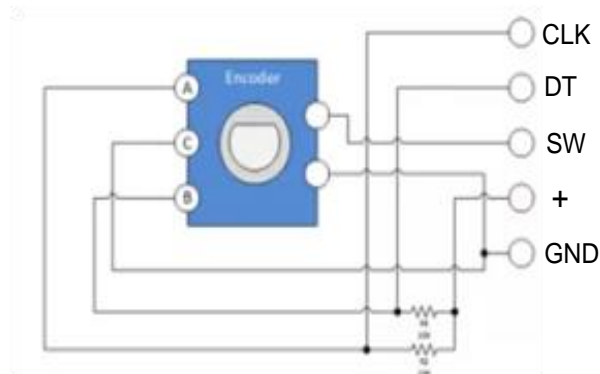
$$error = \left| \frac{12 - 11,96}{12} \right| \cdot 100\% = 0,33\%$$

Berdasarkan dari tabel 4.5 pengujian sensor tegangan DC dapat dilihat presentase *error* yang didapatkan bahwa presentase kesalahan sebesar 1,3% dan presentase kesalahan terkecil 0,33% dengan presentase rata-rata *error* sebesar 0,81%. Dari data presentase *error* yang di dapatkan dapat di simpulkan bahwa sensor tegangan DC masih dapat bekerja dengan baik.

4.2.5 Sensor Rotary Encoder KY-040

A. Implementasi Sensor Rotary Encoder KY-040

Sensor *rotary encoder* KY-040 berfungsi sebagai pengatur jarak jalan robot pemetaan jarak tanam. Pengaturan jarak jalan robot pemetaan jarak tanam dilakukan dengan cara menentukan nilai *encoder* pada sensor dengan cara menyesuaikan diameter pada roda robot pemetaan jarak tanam sehingga bisa menentukan jarak tempuh pada robot. Pada proyek akhir ini dilakukan pembelian modul sensor *rotary encoder* KY-040, sehingga dapat langsung digunakan tanpa harus membuat modul *rotary encoder* KY-040. Adapun skematik sensor *rotary encoder* KY-040 pada poros roda roboditunjukkan pada gambar 4.23 berikut.



Gambar 4. 23 Sensor *Rotary Encoder* KY-040 [8]

Adapun pemasangan sensor *rotary encoder* KY-040 pada poros robot pemetaan jarak tanam dapat dilihat pada gambar 4.24 berikut.



Gambar 4. 24 Sensor *Rotary Encoder* KY-040

B. Pengujian Sensor *Rotary Encoder* KY-040

Pengujian sensor *rotary encoder* KY-040 dilakukan dengan membuat program menggunakan *software* Arduino IDE. Pengujian sensor *rotary encoder* KY-040 hanya menguji nilai *encoder* yang terbaca pada sensor dengan cara sensor *rotary encoder* KY-040 di *couple* keporos roda, selanjutnya nilai akan ditampilkn di *serial monitor* pada *software* Arduino IDE. Adapun Gambar 4.25 blok diagram pengujian sensor *rotary encoder* KY-040 seperti berikut.



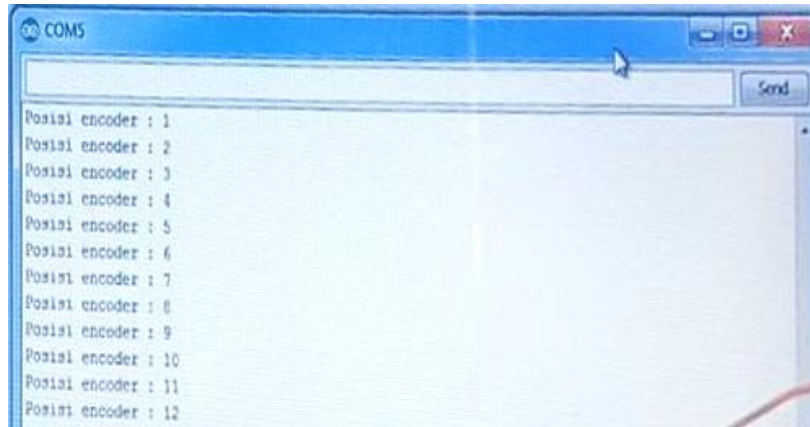
Gambar 4. 25 Blok Diagram Pengujian Sensor *Rotary Encoder* KY-040

Berikut adalah *list* program Arduino IDE untuk pengujian sensor *rotary encoder* KY-040.

```
void loop()
{
  awal = digitalRead(pinCLK);
  if (awal != akhir)
  {
    if (digitalRead(pinDT) != awal) {encoder ++;nilai = true;}
    else {nilai = false; encoder++;}
    Serial.print("Encoder Position: ");
    Serial.println(encoder); //Menampilkan nilai encoder
  }
  akhir = awal;}

```

Dibawah ini adalah hasil pembacaan dari sensor *rotary encoder* KY-040 dapat dilihat pada gambar 4.26 seperti berikut.



Gambar 4. 26 Pembacaan Sensor *Rotary Encoder* KY-040

Berikut adalah hasil pengujian dari sensor *rotary encoder* KY-040 dapat dilihat pada tabel 4.6 seperti berikut.

Tabel 4. 6 Hasil Pengujian Sensor *Rotary Encoder* KY-040

Jumlah step	Nilai <i>encoder</i>	Jarak (cm)
1	2	2,3
2	4	4,6
3	6	6,9
4	8	9,2
5	10	11,5
6	12	13,8
7	14	16,1
8	16	18,4
9	18	20,7
10	20	23
11	22	25,3

Setelah melakukan pengujian dapat dilihat pada tabel 4.6 bahwa nilai *encoder* yang terbaca berupa nilai positif. Hal ini dikarenakan apabila knop pada sensor *rotary encoder* KY-040 di putar searah jarum jam maupun berlawanan

arah jarum jam maka akan bernilai positif. Pada proyek akhir ini dikarenakan pemasangan sensor *rotary encoder* di *couple* dengan poros roda maka pembacaan nilai sensor *rotary encoder* tergantung dari putaran roda itu sendiri. Roda yang dipakai berdiameter 10 cm, maka nilai sensor *rotary encoder* untuk satu putaran roda adalah 36 (18 step) dengan *error* pembacaan nilai sensor *rotary encoder* sampai 2 step.

4.2.6 Sensor kompas GY-271

A. Implementasi Sensor kompas GY-271

Sensor kompas GY-271 berfungsi untuk menentukan nilai sudut dan nilai sudut tersebut digunakan sebagai arah untuk jalan robot pemetaan jarak tanam. Dalam proyek akhir ini sensor kompas yang digunakan adalah tipe sensor kompas GY-271 HMC5883L hal ini dikarenakan sensor mendeteksi medan magnet bumi sebagai acuan pendeteksi arah sudut. Berdasarkan hal tersebut, dilakukan pembelian sensor kompas GY-271 HMC5883L yang sudah jadi sehingga dapat langsung digunakan tanpa harus membuat modul sensor kompas GY-271 HMC5883L. Berikut adalah pemasangan modul kompas GY-271 dapat dilihat pada gambar 4.27.

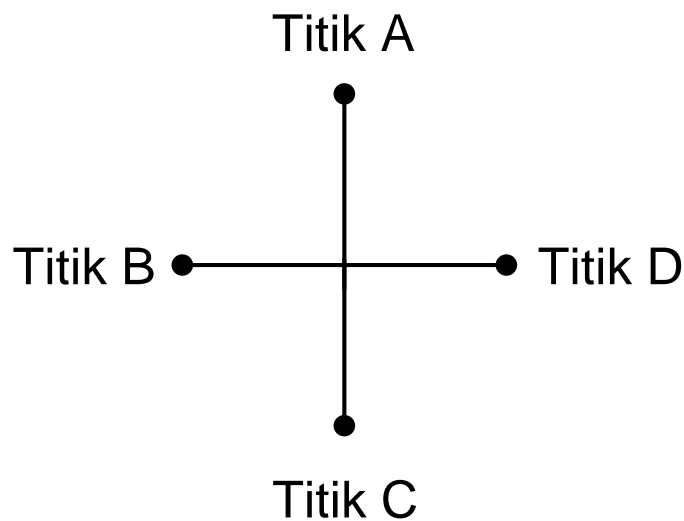


Gambar 4. 27 Sensor Kompas GY-271 HMC5883L

B. Pengujian Sensor Kompas GY-271 HMC5883L

Pengujian sensor kompas GY-271 HMC5883L dilakukan dengan membuat *list* program pada Arduino IDE, pengujian ini dilakukan untuk

mengetahui pembacaan nilai sudut dari sensor kompas GY-271 HMC5883L dengan menentukan titik pengujiannya. Adapun gambar 4.28 titik pengujian sensor kompas GY-271 HMC5883L sebagai berikut.

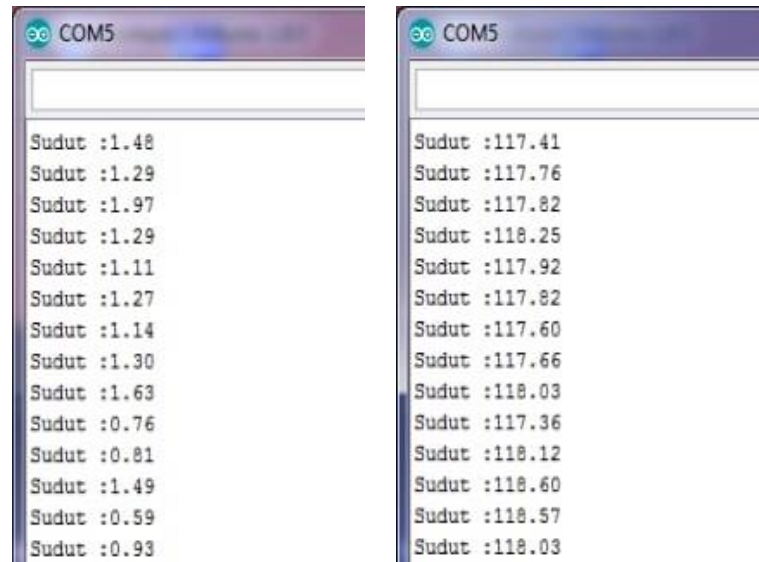


Gambar 4. 28 Titik Pengujian Sensor Kompas GY-271 HMC5883L

Adapun *list* dari program dari sensor kompas GY-271 HMC5883L sebagai berikut.

```
void loop()
{
    int x,y,z;
    compass.read(&x,&y,&z); // tiga aksis x,y,z
    // menghitung heading pada sensor dengan nilai sumbu x dan y
    // 'atan2()' untuk memeriksa formula pada kuadran
    float heading = atan2(y, x);
    float declinationAngle = 0.0404;
    heading += declinationAngle;
    if(heading < 0)
    heading += 2*PI;
    if(heading > 2*PI)
    heading -= 2*PI;
    float headingDegrees = heading * 180/M_PI; //radian ke derajat
    Serial.println(headingDegrees); //menampilkan nilai sudut
    delay(100);
}
```


Berikut adalah hasil pengujian titik C dan titik D pembacaan sensor kompas GY-271 pada *serial monitor* dapat dilihat pada gambar 4.29 sebagai berikut.



Gambar 4. 29 Pengujian Pembacaan Sensor Kompas GY-271 HMC5883L

Berikut adalah hasil dari pengujian sensor kompas GY-271 HMC5883L dapat dilihat pada tabel 4.7 sebagai berikut.

Tabel 4. 7 Hasil Pengujian Sensor Kompas GY-271 HMC5883L

No.	Titik pengujian sensor			
	Titik A	Titik B	Titik C	Titik D
1	254°	190°	118°	0°
2	256°	192°	117°	1°

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa nilai titik pengujian dari sensor kompas GY-271 HMC5883L memiliki selisih nilai sudut yang tidak jauh berbeda dapat dilihat pada tabel 4.7, selisih nilai ini dikarenakan sensor kompas yang sensitif terhadap adanya pengaruh logam disekitar sensor dan pengaruh medan magnet pada tempat tertentu. Untuk itu pemasangan sensor kompas harus berada jauh dengan benda-benda seperti plat, besi dan baterai, walaupun memiliki selisih nilai sudut yang sedikit berbeda sensor kompas ini masih bisa digunakan dalam proyek akhir.

4.2.7 NodeMCU ESP8266

A. Implementasi NodeMCU ESP8266

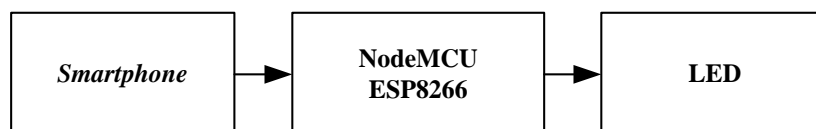
NodeMCU ESP8266 berfungsi sebagai modul *wifi* yang menerima dan mengirim data serta sebagai koneksi antara aplikasi *blynk* pada *smartphone* sehingga *user* bisa mengontrol sistem dari *smartphone* menggunakan modul NodeMCU ESP8266 ini. Pada proyek akhir ini dilakukan pembelian modul NodeMCU ESP8266 yang telah jadi dan mudah didapatkan di pasaran. Adapun gambar 4.30 merupakan bentuk fisik dari modul NodeMCU ESP8266 sebagai berikut.



Gambar 4. 30 Modul NodeMCU ESP8266

B. Pengujian NodeMCU ESP8266

Pengujian dilakukan untuk mengetahui koneksi modul NodeMCU ESP8266 dengan membuat program menghidupkan LED melalui aplikasi *blynk* menggunakan *software* Arduino IDE yang telah dilengkapi *library* ESP8266. Setelah program di *download*, selanjutnya membuat *switch* pada aplikasi *blynk* untuk menghidupkan LED yang telah dirangkai pada *board* NodeMCU ESP8266. Jika LED hidup maka *smartphone* dengan NodeMCU ESP8266 bisa menerima dan mengirim data. Adapun gambar 4.31 blok diagram pengujian NodeMCU ESP8266 sebagai berikut.



Gambar 4. 31 Blok Diagram Pengujian NodeMCU ESP8266

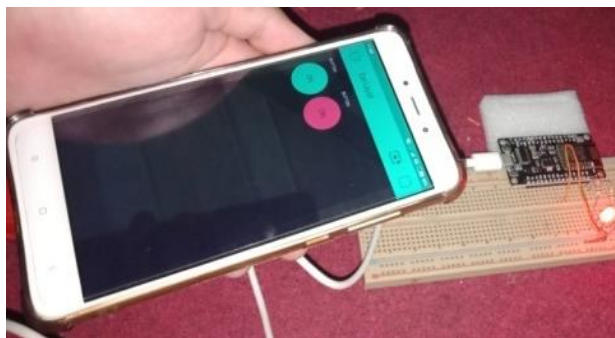
Adapun *list* program untuk pengujian NodeMCU ESP8266 adalah sebagai berikut.

```

/* Program NodeMCU terhubung ke smartphone dengan BLYNK */
void loop()
{
  digitalWrite(led1, HIGH); //kondisi led hidup
  digitalWrite(led1, LOW); //kondisi led mati
  blynk .run();
}

```

Pada gambar 4.32 merupakan pengujian NodeMCU ESP8266 dengan aplikasi *blynk* pada *smartphone*.



Gambar 4. 32 Pengujian NodeMCU ESP8266

Adapun hasil pengujian dari NodeMCU ESP8266 terdapat pada tabel 4.8 dibawah ini.

Tabel 4. 8 Hasil Pengujian LED Pada NodeMCU ESP8266

<i>Switch (Smartphone)</i>	Pin yang digunakan	Kondisi LED
On	D5	hidup
Off	D5	mati

Dari pengujian yang telah dilakukan dapat dilihat dari tabel 4.8 hasil pengujian NodeMCU ESP8266 bahwa, NodeMCU ESP8266 dapat menerima data yang dikirim oleh aplikasi *blynk* begitupula sebaliknya, apabila *switch* pada aplikasi *blynk* ditekan maka LED hidup. Jadi dapat disimpulkan bahwa modul NodeMCU ESP8266 dalam keadaan baik dan dapat digunakan untuk mengontrol robot pemetaan jarak tanam.

4.2.8 Driver IBT-02

A. Implementasi Driver IBT-02

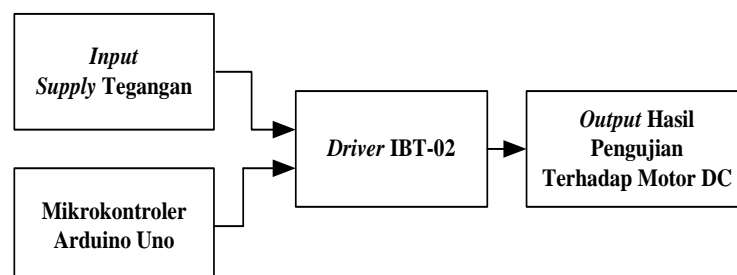
Driver IBT-02 berfungsi mengatur arah putaran dan kecepatan motor DC sebagai penggerak motor DC *power window* untuk mengendalikan roda bagian belakang robot pemetaan jarak tanam berputar sehingga robot bisa berjalan, serta motor DC sebagai pengendali penurun/penaik bor pada sistem pengeboran. Pada proyek akhir ini dilakukan pembelian driver IBT-02 yang telah jadi dan mudah didapat dipasaran sehingga tanpa harus membuat modul driver IBT-02. Adapun gambar 4.33 driver IBT-02 sebagai berikut.



Gambar 4. 33 Modul Driver IBT-02

B. Pengujian Driver IBT-02

Pengujian modul driver IBT-02 dilakukan dengan membuat program untuk mengatur arah putaran dan kecepatan motor DC hal ini berfungsi untuk mengetahui pergerakan motor DC yang dikendalikan dengan modul driver IBT-02. Adapun pembuatan program driver IBT-02 menggunakan software Arduino IDE. Adapun blok diagram pengujian driver IBT-02 dapat dilihat pada gambar 4.34 sebagai berikut.



Gambar 4. 34 Blok Diagram Pengujian Driver IBT-02

Adapun *list* program untuk pengujian modul *driver* IBT-02 sebagai berikut.

```
void loop()
{
  analogWrite(R_M2, 150); //arah putaran berlawanan dengan pwm 150
  analogWrite(L_M2, 0);
  delay(3000);
  analogWrite(R_M2, 0);
  analogWrite(L_M2, 150); //arah putaran searah dengan pwm 150
  delay(3000);
}
```

Pengujian *driver* IBT-02 terhadap motor DC *power window* dapat dilihat pada gambar 4.35 sebagai berikut.



Gambar 4. 35 Pengujian *Driver* IBT-02 Terhadap Motor DC *Power Window*

Adapun hasil pengujian *driver* IBT-02 terhadap motor DC *power window* ditunjukkan pada tabel 4.9 dibawah ini.

Tabel 4. 9 Hasil Pengujian *Driver* IBT-02

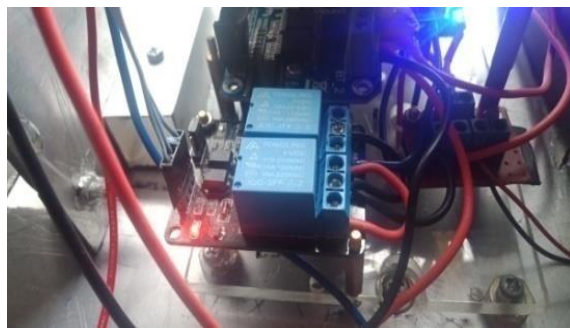
No.	Pengujian terhadap pin <i>driver</i> IBT-02	Pengujian terhadap putaran motor
1	RPWM	berlawanan arah jarum jam
2	LPWM	berputar searah jarum jam

Dari tabel 4.9 hasil pengujian *driver* IBT-02 pin RPWM dan LPWM bekerja sesuai dengan perancangan yang diinginkan yaitu motor akan berputar berlawanan arah dan berputar searah jarum jam. Jadi bisa disimpulkan bahwa *driver* IBT-02 mengatur arah putaran dan kecepatan motor sesuai diinginkan.

4.2.9 Modul *Relay 2-Channel*

A. Implementasi Modul *Relay 2-Channel*

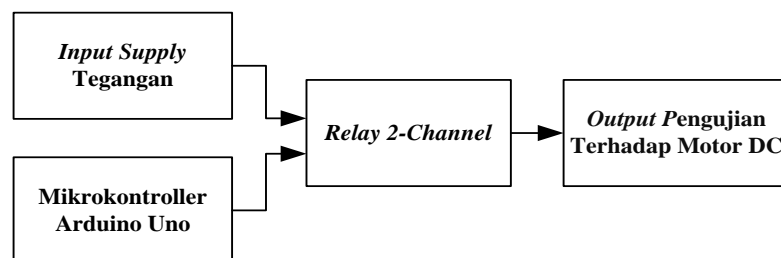
Relay 2-channel berfungsi menggerakkan motor DC untuk sistem pengeboran dalam menandai titik tanam pada tanah. Pada proyek akhir ini dilakukan pembelian modul *relay 2-channel* yang telah jadi dipasaran. Berikut adalah gambar hasil pembelian modul *relay 2-channel* yang dapat dilihat pada gambar 4.36 dibawah ini.



Gambar 4. 36 Modul *Relay 2-Channel*

B. Pengujian Modul *Relay 2-Channel*

Pengujian dilakukan dengan membuat membuat program modul *relay 2-channel* menggunakan *software* Arduino IDE untuk mengendalikan motor DC sebagai sistem pengeboran yang berfungsi dalam menandai titik tanam pada tanah. Berikut diagram blok pengujian *relay 2-channel* dapat dilihat pada gambar 4.37.



Gambar 4. 37 Diagram Blok Pengujian *Relay 2-Channel*

Adapun *list* program untuk pengujian *relay 2-channel* menggunakan *software* Arduino IDE adalah sebagai berikut.

```

void loop()
{
  digitalWrite(Relay1, LOW); //penurun bor dijalankan
  digitalWrite(Relay2, HIGH); delay(3000);
  digitalWrite(Relay1, HIGH); // penaik bor dijalankan
  digitalWrite(Relay2, LOW); delay(3000);
}

```

Berikut merupakan gambar 4.38 pengujian *relay 2-channel* terhadap motor DC dapat dilihat sebagai berikut.



Gambar 4. 38 Pengujian *Relay 2-Channel* Terhadap Motor DC

Berikut merupakan tabel 4.10 hasil pengujian *relay 2-channel* terhadap motor DC dapat dilihat sebagai berikut.

Tabel 4. 10 Hasil Pengujian *Relay 2-Channel*

No.	<i>Channel relay</i>	Logika	Pengujian terhadap motor DC	Keterangan
1	1	Low	motor DC berputar	searah
2	2	Low	motor DC berputar	berlawanan arah
3	1 & 2	High	motor DC berhenti	berhenti

Berdasarkan hasil pengujian, sebagaimana terdapat pada tabel 4.10 dari data yang didapat bahwa modul *relay 2-channel* berfungsi dengan baik, apabila *input* dalam kondisi low maka *relay* akan aktif dan motor DC akan berputar sehingga menyebabkan bor turun/naik. Jadi dapat disimpulkan bahwa *relay 2-channel* bekerja dengan baik dan dapat dipakai dalam pembuatan proyek akhir.

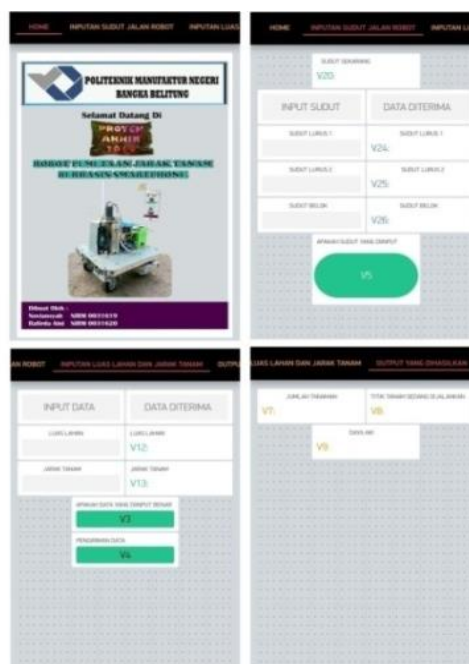
4.3 Perancangan *Software* Robot Pemetaan Jarak Tanam

Perancangan *software* robot pemetaan jarak tanam merupakan tahapan penting dalam hubungan antara pengguna dengan sistem *hardware* untuk memudahkan pengerjaan pembuatan sistem *software*. Perancangan *software* meliputi perancangan berupa aplikasi *blynk* pada *smartphone* dan pemrograman mikrokontroler pada Arduino Uno dan NodeMCU ESP8266.

4.3.1 Aplikasi *Blynk*

A. Perancangan Aplikasi *Blynk* Pada *Smartphone*

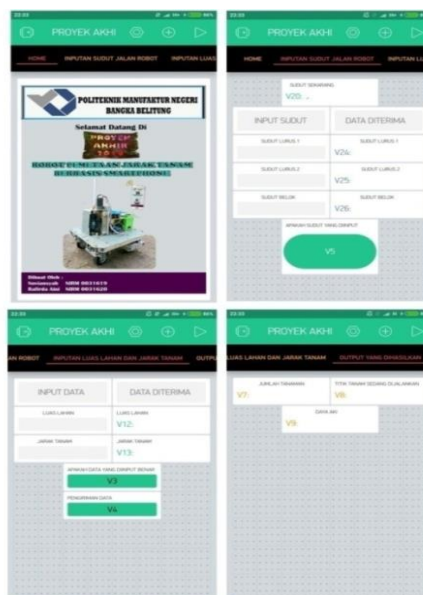
Perancangan aplikasi *blynk* pada *smartphone* bertujuan untuk mengontrol sistem robot pemetaan jarak tanam. Aplikasi *blynk* digunakan sebagai sistem komunikasi dengan sistem kontrol pada robot pemetaan jarak tanam, sistem memasukkan data berupa nilai sudut, luas lahan, jarak tanam serta menampilkan nilai jumlah tanaman, titik tanam yang dikerjakan dan pemakaian daya baterai yang digunakan. Aplikasi *blynk* dilakukan dengan cara mengunduh dan menginstal aplikasi *blynk* dari *playstore* pada *smartphone*. Berikut tampak pada gambar 4.39 perancangan *layout* tampilan aplikasi *blynk* pada *smartphone*.



Gambar 4. 39 *Layout* Aplikasi *Blynk*

B. Pembuatan Aplikasi *Blynk* Pada *Smartphone*

Setelah perancangan *layout* aplikasi *blynk* telah selesai, selanjutnya dilakukan pembuatan berupa *layout* aplikasi *blynk* pada *smartphone* yang sesuai dengan perancangan, dalam pembuatan aplikasi *blynk* ini ada beberapa jenis kategori komponen yang digunakan untuk membuat *layout* yaitu menggunakan *controller* untuk mengirimkan data atau perintah ke *hardware*, *display* untuk menampilkan data dari *hardware* ke *smartphone* dan *interface* digunakan dalam pengaturan tampilan pada aplikasi *blynk* berupa menu ataupun tab. Berikut adalah gambar 4.40 hasil pembuatan *layout* aplikasi *blynk* pada *smartphone* sesuai dengan perancangan yang telah dibuat.

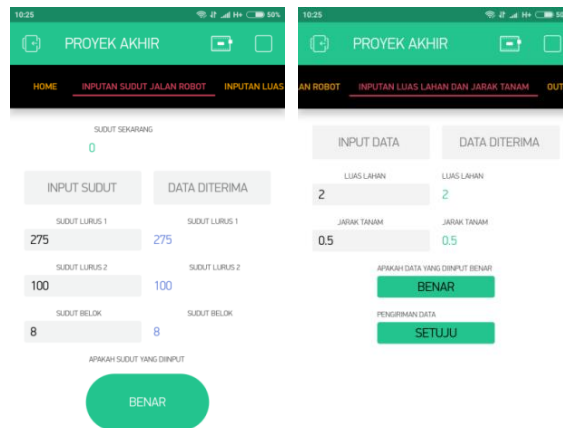


Gambar 4. 40 Hasil Pembuatan *Layout* Pada Aplikasi *Blynk*

C. Pengujian Aplikasi *Blynk* Pada *Smartphone*

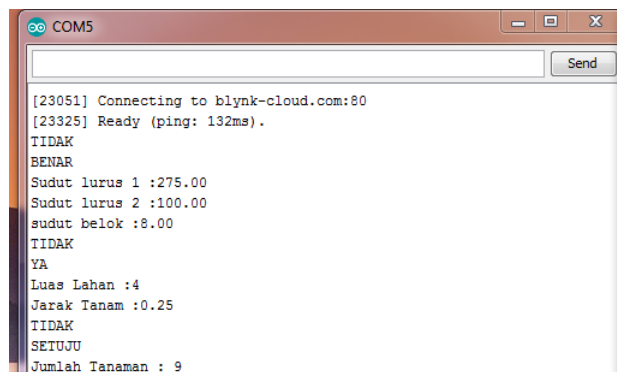
Pengujian aplikasi *blynk* dilakukan dengan memasukkan data berupa nilai sudut, luas lahan dan jarak tanam pada *smartphone*, setelah data di masukkan akan tampil nilai data yang diterima oleh NodeMCU ESP8266 berupa nilai sudut, luas lahan dan jarak tanam hal ini difungsikan untuk apakah NodeMCU ESP8266 telah menerima data dari aplikasi *blynk* pada *smartphone*, apakah data yang dimasukkan pada aplikasi *blynk* berhasil dan sesuai dengan perancangan atau

tidak. Berikut adalah hasil pengujian memasukkan nilai sudut, luas lahan dan jarak tanam terhadap aplikasi *blynk* pada *smartphone* yang ditampilkan tampak pada gambar 4.41 berikut.



Gambar 4. 41 Pengujian Aplikasi *Blynk*

Berikut adalah hasil pengujian terhadap aplikasi *blynk* pada *smartphone* yang ditampilkan pada *serial monitor* tampak pada gambar 4.42 berikut.



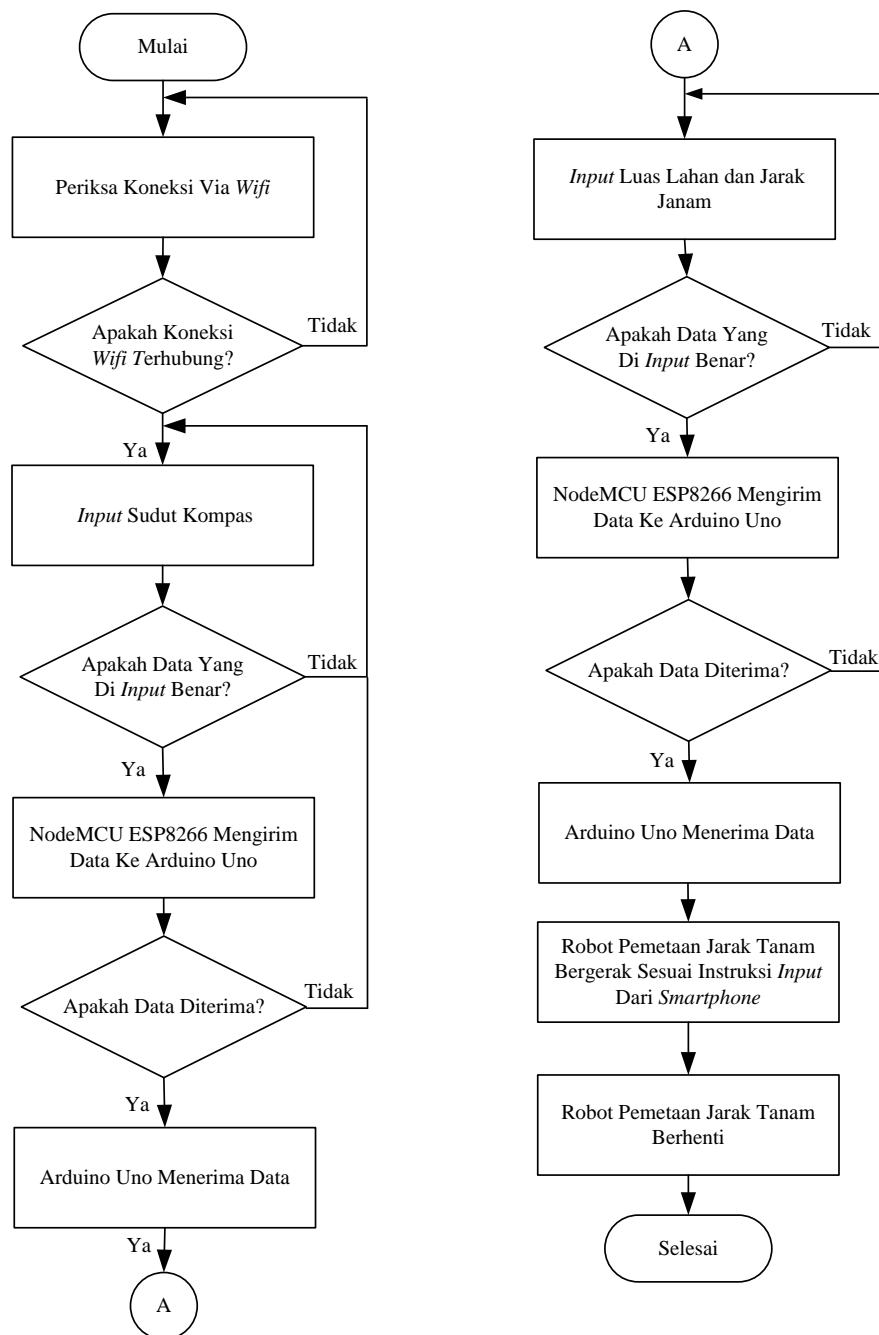
Gambar 4. 42 Pengujian Data Yang Diterima Arduino Dari Aplikasi *Blynk*

Setelah melakukan pengujian aplikasi *blynk* pada *smartphone* dapat disimpulkan bahwa data berupa nilai sudut, luas lahan dan jarak tanam aplikasi *blynk* pada *smartphone* sesuai yang diharapkan dapat dilihat pada gambar 4.41 pengujian *input output* data aplikasi *blynk* dan gambar 4.42 pengujian data yang diterima Arduino dari aplikasi *blynk* , bahwa nilai data yang di *input* dan *output* pada aplikasi *blynk* sama dengan tampilan di *serial monitor*.

4.3.2 Pemrograman

A. Perancangan Program Arduino Uno

Perancangan program Arduino Uno adalah perancangan untuk sistem kontrol robot pemetaan jarak tanam yang akan dibuat. Berikut gambar 4.43 *flowchart* program Arduino Uno.



Gambar 4. 43 *Flowchart* Arduino Uno

Perancangan program berdasarkan *flowchart* Arduino Uno, meliputi pengkoneksian dengan jaringan *wifi*, perancangan pemrograman mengirim dan menerima data, pembacaan sensor tegangan DC, sensor arus, *rotary encoder* KY-040 dan sensor kompas GY-271 serta perancangan pemrograman secara keseluruhan.

B. Pembuatan Program Arduino Uno

Pembuatan program Arduino Uno untuk robot pemetaan jarak tanam dilakukan menggunakan *software* Arduino IDE. Pembuatan pemrograman robot pemetaan jarak tanam dibagi menjadi beberapa tahapan sesuai dengan perancangan yaitu meliputi:

1. Pemrograman NodeMCU ESP8266 untuk mengirim dan menerima data dari aplikasi *blynk* pada *smartphone*.
2. Pemrograman sensor tegangan DC untuk mengukur *supply* tegangan *input*.
3. Pemrograman sensor *rotary encoder* KY-040 untuk mengatur jarak tempuh robot.
4. Pemrograman sensor kompas GY-271 HMC5883L untuk menentukan arah jalan robot.
5. Pemrograman *driver* IBT-2 untuk mengatur arah putaran dan kecepatan motor DC.
6. Pemrograman *relay 2-channel* untuk mengatur arah putaran motor DC.
7. Pemrograman keseluruhan robot pemetaan jarak tanam.

4.4 Perakitan *Hardware* Mekanik Dan Elektrik Robot

Perakitan *hardware* mekanik dan elektrik bertujuan untuk menggabungkan keseluruhan dari *hardware* yang sesuai dengan perancangan yang telah dibuat. Perakitan *hardware* robot pemetaan jarak tanam dilakukan dengan cara merakit secara keseluruhan pada masing-masing dari setiap bagian *hardware* mekanik dan *hardware* elektrik robot pemetaan jarak tanam yang telah dibuat menjadi satu kesatuan.

Berikut gambar 4.44 hasil perakitan *hardware* mekanik dan elektrik robot pemetaan jarak tanam.



Gambar 4. 44 Hasil Perakitan Robot Pemetaan Jarak Tanam

4.5 Pengujian Sistem Robot Pemetaan Jarak Tanam

Setelah selesai pada tahapan perancangan, pembuatan, perakitan, pengujian *hardware* mekanik elektrik dan *software* pada robot pemetaan jarak tanam, maka tahap selanjutnya adalah tahap pengujian sistem robot pemetaan jarak tanam yang akan dilakukan pengujian secara keseluruhan terhadap robot. Pengujian terhadap robot pemetaan jarak tanam bertujuan untuk mengetahui kinerja dari robot apakah robot sudah mampu bekerja sesuai dengan tujuan yang diharapkan dan dicapai, yaitu robot yang bisa melakukan pemetaan jarak tanam yang sesuai dengan keinginan pengguna dengan memasukkan data melalui *smartphone*. Adapun pengujian dari sistem robot pemetaan jarak tanam, yaitu sebagai berikut:

4.5.1 Pengujian Memasukkan Data Melalui *Smartphone*

Pengujian memasukkan data dari *smartphone* bertujuan untuk mengetahui apakah data yang diinginkan sesuai dengan yang diinginkan atau tidak dan bertujuan untuk mengetahui jangkauan maksimal yang dapat dijangkau oleh NodeMCU ESP8266 terhadap *smartphone*, hal ini berguna dalam

mengontrol robot pemetaan jarak tanam untuk pergerakan robot sesuai dengan *input* data dari *smartphone* yaitu berupa luas lahan dan jarak tanam.

Pengujian memasukkan data dari *smartphone* dilakukan menggunakan robot dengan cara *input* data berupa luas lahan dan jarak tanam yang di *input* melalui aplikasi *blynk* pada *smartphone* untuk menjalankan robot sehingga mengetahui hasil dari pemetaan jarak tanam dan jumlah titik tanam yang dikerjakan oleh robot pemetaan jarak tanam serta menampilkan pemakaian daya baterai dan jumlah titik tanam yang sedang dikerjakan. Sedangkan untuk pengujian jangkauan NodeMCU ESP8266 terhadap *smartphone* dilakukan dengan cara menjauhkan perangkat sistem kontrol robot pemetaan jarak tanam dari *smartphone*. Adapun hasil pengujian memasukkan data nilai sudut dari *smartphone* dapat dilihat pada gambar 4.45 sebagai berikut.



Gambar 4. 45 Pengujian *Input* Nilai Sudut

Berikut ini hasil pengujian memasukkan data berupa luas lahan dan jarak tanam pada *smartphone* dapat dilihat pada gambar 4.46 sebagai berikut.



Gambar 4. 46 Pengujian *Input* Luas Lahan Dan Jarak Tanam

Berikut adalah hasil pengujian jangkauan NodeMCU ESP8266 terhadap *smartphone*, dapat dilihat pada tabel 4.11 berikut ini.

Tabel 4. 11 Pengujian Jangkauan NodeMCU ESP8266 Terhadap *Smartphone*

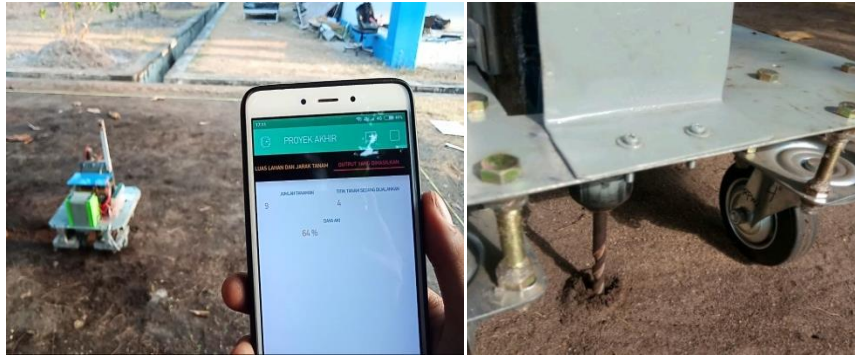
Jarak jangkauan (m)	Keterangan
1	Terhubung
5	Terhubung
10	Terhubung
20	Terhubung
50	Terhubung
70	Terhubung
100	Terhubung

Berdasarkan pengujian memasukkan data dan jangkauan koneksi *smartphone* dengan mikrokontroler sesuai dengan yang diinginkan. Pada pengujian memasukkan data melalui *smartphone*, data yang diterima oleh mikrokontroler sesuai dengan data yang diinput dari *smartphone* dan waktu pengiriman data yang di *input* dari *smartphone* ke mikrokontroler adalah 1-15 detik, serta waktu penerimaan data yang dikirim dari mikrokontroler ke *smartphone* adalah 1-15 detik. Pada pengujian jangkauan yang dilakukan didapat hasil seperti tabel 4.11 bahwa semua jarak yang diuji koneksi antara *smartphone* dengan mikrokontroler tetap terhubung karena koneksi *smartphone* dengan mikrokontroler tergantung dengan jaringan *smartphone* itu sendiri.

4.5.2 Pengujian Robot Pemetaan Jarak Tanam

Pengujian ini bertujuan untuk mengetahui hasil dari sistem robot yang telah dibuat apakah sesuai dengan keinginan atau tidak. Pengujian dilakukan untuk mengetahui hasil dari pemetaan jarak tanam dan jumlah tanaman dengan perbandingan menggunakan robot maupun manual. Pengujian manual yaitu menentukan pemetaan jarak tanam dan jumlah tanaman menggunakan alat ukur berupa meteran. Untuk pengujian ini nilai luas lahan dan jarak tanam telah ditentukan terlebih dahulu. Berikut hasil pengujian pemetaan jarak tanam dan jumlah tanaman.

Berikut adalah gambar 4.47 hasil pengujian pengeboran titik tanam yang sedang dikerjakan oleh robot pemetaan jarak tanam.



Gambar 4. 47 Pengujian Pengeboran Titik Tanam

Dibawah ini tampak pada gambar 4.48 merupakan pengujian pengukuran jarak tanam dengan meteran.



Gambar 4. 48 Pengujian Pengukuran Jarak Tanam

Berikut adalah rumus perhitungan jumlah tanaman pada sistem robot yang digunakan serta keterangan untuk simulasi pengujian jumlah tanaman secara manual, yaitu sebagai berikut:

1. Rumus Perhitungan Jumlah Tanaman

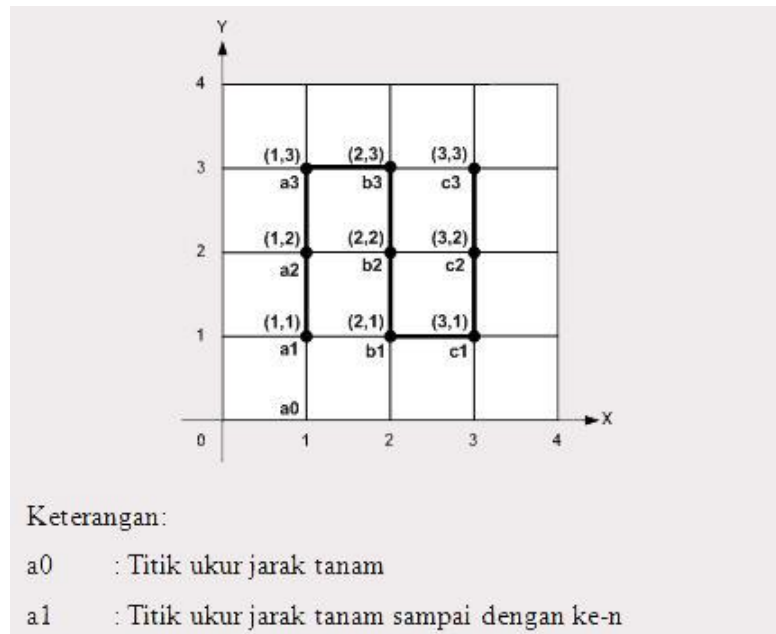
Adapun rumus perhitungan yang digunakan dalam sistem robot pemetaan jarak tanam adalah sebagai berikut.

$$\text{Luas lahan tanam} = \text{Luas lahan} - (2 \times \text{jarak tanam}) \dots\dots\dots (4.3)$$

$$\text{Jumlah tanaman} = 2 + \frac{(\text{Luas lahan tanam})^2}{\text{jarak tanam}^2} + \left(\left(\frac{\text{Luas lahan tanam}}{\text{jarak tanam}} \times 2 \right) - 1 \right) \dots\dots (4.4)$$

2. Simulasi Pengujian Jumlah Tanaman Manual

Berikut adalah simulasi perhitungan jumlah tanam secara manual dapat dilihat pada gambar 4.49 sebagai berikut.



Gambar 4. 49 Simulasi Pengujian Jumlah Tanaman Secara Manual

a) Luas lahan 2 m x 2 m, jarak tanam 0,5 m x 0,5 m

$$\text{Luas lahan tanam} = 2 - (2 \times 0,5)$$

$$\text{Luas lahan tanam} = 2 - 1$$

$$\text{Luas lahan tanam} = 1$$

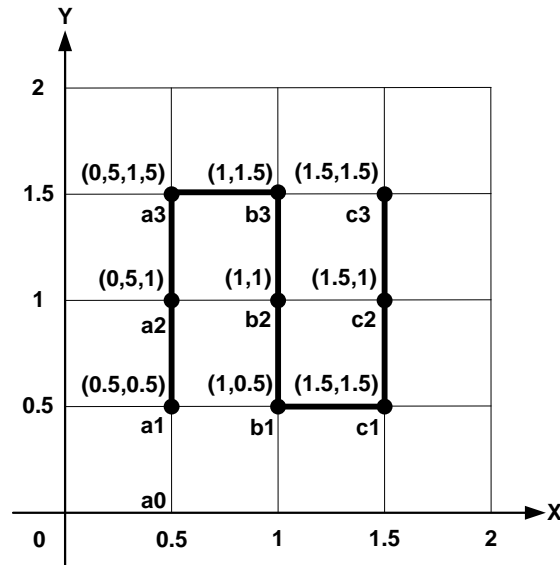
$$\text{Jumlah Tanaman} = 2 + \frac{(1)^2}{0,5^2} + \left(\left(\frac{1}{0,5} \times 2 \right) - 1 \right)$$

$$\text{Jumlah Tanaman} = 2 + \frac{1}{0,25} + \left(\left(\frac{1}{0,5} \times 2 \right) - 1 \right)$$

$$\text{Jumlah Tanaman} = 2 + 4 + ((4) - 1)$$

$$\text{Jumlah Tanaman} = 2 + 4 + 3 = 9$$

Adapun pengujian jumlah tanaman manual (a) dapat disimulasi tampak pada gambar 4.50 sebagai berikut.



Gambar 4. 50 Pengujian Jumlah Tanaman Manual (a)

Adapun hasil pengujian pertama koordinat pengeboran (a) dapat dilihat pada tabel 4.12 sebagai berikut.

Tabel 4. 12 Hasil Pengujian 1 Koordinat Pengeboran (a)

No	Koordinat pengeboran	Posisi robot		Koordinat error (%)	
	(X,Y)	X	Y	X	Y
1	(0, 0)	-	-	-	-
2	(0.5, 0.5)	0,5	0,5	0%	0%
3	(0.5, 1)	0.47	1.02	6%	2%
4	(0.5, 1.5)	0.47	1.54	6%	2,6%
5	(1, 1.5)	0.99	1.5	1%	0%
6	(1, 1)	0.99	1.04	1%	4%
7	(1, 0.5)	1.01	0.5	1%	0%
8	(1.5, 0.5)	1.46	0.56	2,6%	12%
9	(1.5, 1)	1.46	0.99	2,6%	1%
10	(1.5, 1.5)	1.46	1.5	2.6%	0%
Rata-rata error				3%	2%

Adapun hasil pengujian kedua koordinat pengeboran (a) dapat dilihat pada tabel 4.13 sebagai berikut.

Tabel 4. 13 Hasil Pengujian 2 Koordinat Pengeboran (a)

No	Koordinat pengeboran	Posisi robot		Koordinat <i>error</i> (%)	
	(X,Y)	X	Y	X	Y
1	(0, 0)	-	-	-	-
2	(0.5, 0.5)	0.5	0.5	0%	0%
3	(0.5, 1)	0.47	1.04	6%	4%
4	(0.5, 1.5)	0.5	1.55	0%	3,3%
5	(1, 1.5)	1.01	1.55	1%	3,3%
6	(1, 1)	1.04	1.05	4%	5%
7	(1, 0.5)	1.06	0.55	6%	10%
8	(1.5, 0.5)	1.54	0.57	2,6%	14%
9	(1.5, 1)	1.52	1.07	1,3%	7%
10	(1.5, 1.5)	1.48	1.57	4,6%	4,6%
Rata-rata <i>error</i>				3%	6%

Adapun hasil pengujian pertama jarak tanam (a) dapat dilihat pada tabel 4.14 sebagai berikut.

Tabel 4. 14 Hasil Pengujian 1 Jarak Tanam (a)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	<i>Error</i> (%)
1		a0-a1	0,5	0%
2		a1-a2	0,51	2%
3		a2-a3	0,49	2%
4		a3-b3	0,53	6%
5		b3-b2	0,47	6%
6	0,5	b2-b1	0,54	8%
7		b1-b0	0,52	4%
8		b1-c1	0,5	0%
9		c0-c1	0,56	12%
10		c1-c2	0,46	8%
11		c2-c3	0,5	0%
Rata-rata <i>error</i>				4%

Berikut merupakan hasil pengujian kedua jarak tanam (a) dapat dilihat pada tabel 4.15 sebagai berikut.

Tabel 4. 15 Hasil Pengujian 2 Jarak Tanam (a)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	Error (%)
1		a0-a1	0,5	0%
2		a1-a2	0,51	2%
3		a2-a3	0,51	2%
4		a3-b3	0,5	0%
5		b3-b2	0,48	4%
6	0,5	b2-b1	0,49	2%
7		b1-b0	0,57	14%
8		b1-c1	0,48	4%
9		c0-c1	0,57	14%
10		c1-c2	0,51	2%
11		c2-c3	0,49	2%
Rata-rata error				4%

b) Luas lahan 4 m x 4 m, jarak tanam 1 m x 1 m

$$\text{Luas lahan tanam} = \text{Luas lahan} - (2 \times \text{jarak tanam})$$

$$\text{Luas lahan tanam} = 4 - (2 \times 1)$$

$$\text{Luas lahan tanam} = 4 - 2$$

$$\text{Luas lahan tanam} = 2$$

$$\text{Jumlah Tanaman} = 2 + \frac{(2)^2}{1^2} + \left(\left(\frac{2}{1} \times 2 \right) - 1 \right)$$

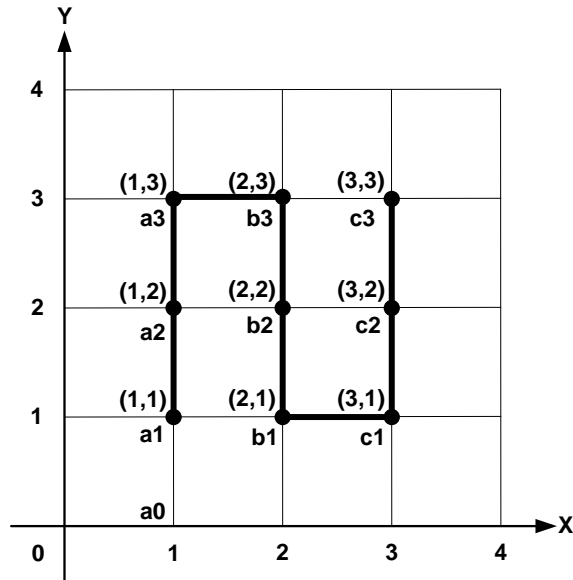
$$\text{Jumlah Tanaman} = 2 + \frac{4}{1} + \left(\left(\frac{4}{1} \right) - 1 \right)$$

$$\text{Jumlah Tanaman} = 2 + \frac{4}{1} + (4 - 1)$$

$$\text{Jumlah Tanaman} = 2 + 4 + 3$$

$$\text{Jumlah Tanaman} = 9$$

Adapun pengujian jumlah tanaman manual (b) dapat disimulasi tampak pada gambar 4.51 sebagai berikut.



Gambar 4. 51 Pengujian 2 Jumlah Titik Tanam Manual (b)

Adapun hasil pengujian pertama koordinat pengeboran (b) dapat dilihat pada tabel 4.16 sebagai berikut.

Tabel 4. 16 Hasil Pengujian 1 Koordinat Pengeboran (b)

No	Koordinat pengeboran	Posisi robot		Koordinat error (%)	
	(X,Y)	X	Y	X	Y
1	(0,0)	-	-	-	-
2	(1,1)	1	1	0%	0%
3	(1,2)	1.02	2.07	2%	3.5%
4	(1,3)	1.03	3.07	3%	2,3%
5	(2,3)	2.07	3.07	2,5%	2,3%
6	(2,2)	2.12	2.10	6%	5%
7	(2,1)	2.18	1.12	9%	12%
8	(3,1)	3.10	1.15	3,3%	15%
9	(3,2)	3.10	2.18	3,3%	9%
10	(3,3)	3.10	3.15	3,3%	5%
Rata-rata error				4%	6%

Adapun hasil pengujian kedua koordinat pengeboran (b) dapat dilihat pada tabel 4.17 sebagai berikut.

Tabel 4. 17 Hasil Pengujian 2 Koordinat Pengeboran (b)

No	Koordinat pengeboran	Posisi robot		Koordinat error (%)	
	(X,Y)	X	Y	X	Y
1	(0, 0)	-	-	-	-
2	(1,1)	1	1	0%	0%
3	(1,2)	1.01	1.97	1%	1,5%
4	(1,3)	0.99	2.97	1%	1%
5	(2,3)	1.99	2.92	0,5%	2,6%
6	(2,2)	2	1.97	0%	1,5%
7	(2,1)	2.05	1	2,5%	0%
8	(3,1)	2.95	1.02	1,6%	2%
9	(3,2)	2.97	1.98	1%	1%
10	(3,3)	2.98	2.95	0,6%	1,6%
Rata-rata error				1%	1%

Adapun hasil pengujian pertama jarak tanam (b) dapat dilihat pada tabel 4.18 sebagai berikut.

Tabel 4. 18 Hasil Pengujian 1 Jarak Tanam (b)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	Error (%)
1		a0-a1	1	0%
2		a1-a2	1,07	7%
3		a2-a3	1,05	5%
4		a3-b3	1,04	4%
5		b3-b2	1,02	2%
6	1	b2-b1	1,01	1%
7		b1-b0	1,18	18%
8		b1-c1	0,96	4%
9		c0-c1	1,15	15%
10		c1-c2	1,07	7%
11		c2-c3	1,04	4%
Rata-rata error				6%

Adapun hasil pengujian kedua jarak tanam (b) dapat dilihat pada tabel 4.19 sebagai berikut.

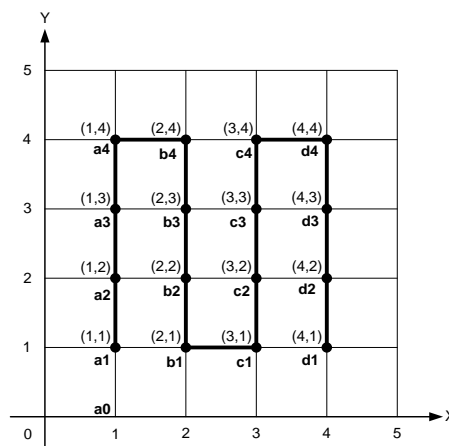
Tabel 4. 19 Hasil Pengujian 2 Jarak Tanam (b)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	Error (%)
1		a0-a1	1	1%
2		a1-a2	0,96	4%
3		a2-a3	1,02	2%
4		a3-b3	0,98	2%
5		b3-b2	0,92	8%
6	1	b2-b1	1.05	5%
7		b1-b0	1,01	1%
8		b1-c1	0,92	8%
9		c0-c1	0,99	1%
10		c1-c2	0,97	3%
11		c2-c3	1,01	1%
Rata-rata <i>error</i>				3%

c) Luas lahan 5 m x 5 m, jarak tanam 1 m x 1 m

$$Jumlah\ Tanaman = 2 + \frac{(3)^2}{1^2} + \left(\left(\frac{3}{1} \times 2 \right) - 1 \right) = 16$$

Adapun pengujian jumlah tanaman manual (c) dapat disimulasi tampak pada gambar 4.52 sebagai berikut.



Gambar 4. 52 Pengujian 3 Jumlah Titik Tanam Manual (c)

Adapun hasil pengujian pertama koordinat pengeboran (c) dapat dilihat pada tabel 4.20 sebagai berikut.

Tabel 4. 20 Hasil Pengujian 1 Koordinat Pengeboran (c)

No	Koordinat pengeboran	Posisi robot		Koordinat error (%)	
	(X, Y)	X	Y	X	Y
1	(0, 0)	-	-	-	-
2	(1, 1)	1	1	0%	0%
3	(1, 2)	0.97	2	3%	0%
4	(1, 3)	0.96	2.98	4%	0,6%
5	(1, 4)	1	3.99	0%	0,25%
6	(2, 4)	2.01	4	0,5%	0%
7	(2, 3)	2.02	3.03	1%	1%
8	(2, 2)	2.10	2.01	5%	0,5%
9	(2, 1)	2.17	1.05	8,5%	5%
10	(3, 1)	3.07	1.04	2,3%	4%
11	(3, 2)	3.08	2.13	2,6%	6,5%
12	(3, 3)	3.02	3.1	0,6%	3,3%
13	(3, 4)	3.13	4.05	4,3%	1,25%
14	(4, 4)	3.93	4.03	1,75%	0,75%
15	(4, 3)	3.9	3.04	2,5%	1,3%
16	(4, 2)	3.93	2.05	1,75%	2,5%
17	(4, 1)	4.02	1.05	5%	5%
Rata-rata error				3%	2%

Adapun hasil pengujian pertama koordinat pengeboran (c) dapat dilihat pada tabel 4.21 sebagai berikut.

Tabel 4. 21 Hasil Pengujian 2 Koordinat Pengeboran (c)

No	Koordinat pengeboran	Posisi robot		Koordinat error (%)	
	(X, Y)	X	Y	X	Y
1	(0, 0)	-	-	-	-

2	(1, 1)	1	1	0%	0%
3	(1, 2)	0.95	2	5%	0%
4	(1, 3)	0.95	3	5%	0%
5	(1, 4)	1	4	0%	0%
6	(2, 4)	2.01	4	0,5%	0%
7	(2, 3)	2.01	3.04	0,5%	1,3%
8	(2, 2)	2.06	1.98	3%	1%
9	(2, 1)	2.15	1.06	7,5%	6%
10	(3, 1)	3	1	0%	0%
11	(3, 2)	3.05	2.08	1,6%	4%
12	(3, 3)	3.12	3.06	4%	2%
13	(3, 4)	3.08	4	2,6%	0%
14	(4, 4)	4.01	3.95	0,25%	1,25%
15	(4, 3)	4.01	2.95	0,25%	1,6%
16	(4, 2)	4.02	1.99	0,5%	0,5%
17	(4, 1)	4.05	1.02	1,25%	2%
Rata-rata <i>error</i>				2%	1%

Adapun hasil pengujian pertama jarak tanam(c) dapat dilihat pada tabel 4.22 sebagai berikut.

Tabel 4. 22 Hasil Pengujian 1 Jarak Tanam (c)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	<i>Error</i> (%)
1		a0-a1	1	0%
2		a1-a2	1,03	3%
3		a2-a3	0,97	3%
4		a3-a4	1	0%
5		a4-b4	0,99	1%
6	1	b4-b3	0,97	3%
7		b3-b2	1,03	3%
8		b2-b1	0,96	4%
9		b1-c1	0,9	10%
10		c1-c2	1,02	2%
11		c2-c3	0,94	6%

12	c3-c4	0,97	3%
13	c4-d4	0,85	15%
14	d4-d3	1	0%
15	d3-d3	1	0%
16	d2-d1	1	0%
17	b1-b0	1,05	5%
18	c1-c0	1,1	10%
19	d1-d0	1,07	7%
Rata-rata <i>error</i>			4%

Adapun hasil pengujian kedua jarak tanam (c) dapat dilihat pada tabel 4.23 sebagai berikut.

Tabel 4. 23 Hasil Pengujian 2 Jarak Tanam (c)

No	Jarak tanam diinginkan (meter)	Titik tanam	Pengukuran jarak tanam (meter)	<i>Error</i> (%)
1		a0-a1	1	0%
2		a1-a2	1	0%
3		a2-a3	1	0%
4		a3-a4	1	0%
5		a4-b4	1,01	1%
6		b4-b3	0,99	1%
7		b3-b2	0,97	3%
8		b2-b1	1,04	4%
9		b1-c1	0,86	14%
10	1	c1-c2	1,08	8%
11		c2-c3	1	0%
12		c3-c4	0,96	4%
13		c4-d4	0,85	15%
14		d4-d3	1	0%
15		d3-d3	1	0%
16		d2-d1	1	0%
17		b1-b0	1,06	6%
18		c1-c0	1	0%
19		d1-d0	1	0%
Rata-rata <i>error</i>				3%

Berdasarkan pengujian pemetaan jarak tanam yang telah dilakukan berupa perbandingan jumlah titik tanam melalui perhitungan rumus dengan perhitungan manual, koordinat pengeboran dan jarak tanam dengan melakukan 3 kali pengujian yang masing-masing pengujian dilakukan 2 kali. Pada pengujian koordinat pengeboran dilihat pada tabel 4.12 dan tabel 4.13 untuk luas lahan 2 m x 2 m, jarak tanam 0,5 m x 0,5 m bahwa hasil pengerjaan koordinat titik pengeboran robot tidak ada yang benar-benar sesuai dengan koordinat titik pengeboran yang diinginkan. Meskipun begitu hasil yang didapat dari 2 kali pengujian yang dilakukan dengan persentase rata-rata *error* sebesar 3% untuk koordinat X dan 2% untuk koordinat Y untuk pengujian pertama koordinat titik pengeboran. Pada pengujian jarak tanam dapat dilihat pada tabel 4.14 dan tabel 4.15 bahwa keakuratan jarak antar titik tanam belum akurat. Untuk keakuratan jarak antar titik tanam yang dilakukan setelah dilakukan 2 kali pengujian dengan persentase rata-rata *error* sebesar 4% untuk pengujian jarak tanam pertama. Berdasarkan hasil pengujian pemetaan jarak tanam yang meliputi koordinat pengeboran dan jarak tanam, diperoleh nilai rata-rata persentase *error* dengan rumus (4.5) sebagai berikut:

$$error = \left| \frac{\text{jumlah total setiap data}}{\text{jumlah data}} \right| \cdot 100\% \dots\dots\dots (4.5)$$

a) Persentase *error* koordinat pengeboran terhadap (X)

$$error = \left| \frac{3\% + 3\% + 4\% + 1\% + 3\% + 2\%}{6} \right| \cdot 100\% = 2,66\%$$

b) Persentase *error* koordinat pengeboran terhadap (Y)

$$error = \left| \frac{2\% + 6\% + 6\% + 1\% + 2\% + 2\%}{6} \right| \cdot 100\% = 3\%$$

c) Persentase *error* jarak tanam

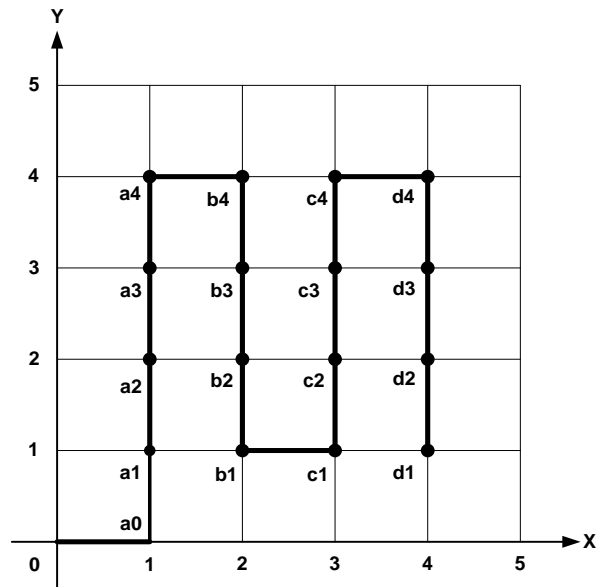
$$error = \left| \frac{4\% + 4\% + 6\% + 3\% + 4\% + 3\%}{6} \right| \cdot 100\% = 4\%$$

Setelah dilakukan analisis terdapat beberapa penyebab koordinat pengeboran dan jarak tanam tidak akurat yang dilakukan oleh robot adalah putaran roda, pembacaan sensor *rotary encoder* dan pembacaan sensor kompas kurang baik. Pengaruh putaran roda adalah ketika robot bergerak melewati permukaan tanah yang kurang padat atau gembur yang menyebabkan putaran roda menjadi kurang lancar atau cengkeraman roda terhadap permukaan tanah menjadi kurang atau yang dikenal dengan roda selip. Kemudian putaran roda yang selip tersebut menyebabkan pergerakan jalan robot ke titik koordinat yang diinginkan selanjutnya menjadi kurang baik. Hal ini dikarenakan putaran roda dengan sensor *rotary* sebagai pengatur jarak jalan robot saling terhubung yaitu pemasangan sensor *rotary encoder* yang di *couple* terhadap poros roda. Oleh karena itu putaran roda tergantung dari nilai sensor *rotary encoder* yang diinginkan artinya jika roda berputar maka sensor *rotary* akan ikut berputar dan membaca nilai dari sensor *rotary encoder* sehingga jalan robot akan berhenti sesuai dengan nilai sensor *rotary encoder* yang sudah diatur pada bagian memasukkan jarak tanam yang dilakukan. Jika putaran roda terjadi selip maka pembacaan sensor *rotary encoder* akan dilanjutkan meskipun robot tidak bergerak selama roda masih berputar hingga nilai sensor *rotary encoder* terpenuhi untuk keadaan roda berhenti. Kemudian saat robot sedang bergerak baik jalan lurus, jalan belok kanan/kiri maupun saat proses pengeboran pasti terjadi getaran pada tiang penyangga yang merupakan tempat pemasangan sensor kompas. Getaran-getaran tersebut menyebabkan pembacaan sudut pada sensor kompas menjadi kurang baik. Pada saat diam saja pembacaan sudut sensor kompas tidak akurat hingga 3 derajat *error* nya, apalagi ditambah dengan adanya getaran-getaran tersebut pastinya akan menyebabkan pembacaan sudut sensor kompas lebih besar lagi derajat *error* nya.

4.5.3 Pengujian Pola Jalan Robot Pemetaan Jarak Tanam

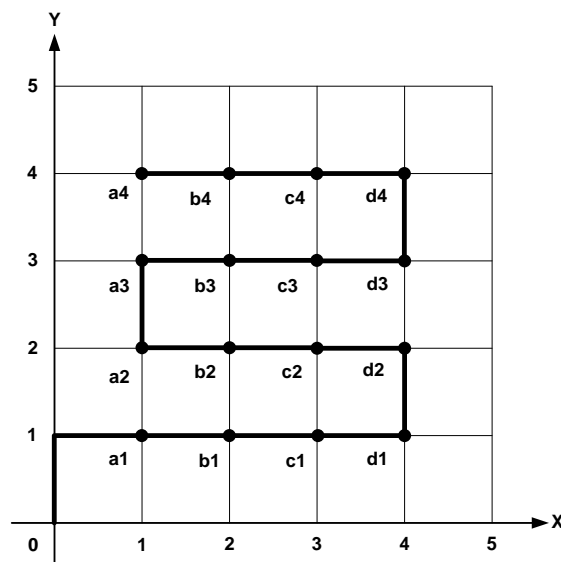
Pengujian ini bertujuan untuk mengetahui tingkat keefisienan pola jalan robot ditentukan dengan membandingkan waktu pengerjaan robot pola jalan pertama dan kedua. Pengujian pola jalan dengan pergerakan robot yang dilakukan

sesuai dengan gambar simulasi yang telah dibuat. Berikut adalah gambar 4.53 untuk simulasi pola pertama jalan robot pemetaan jarak tanam sebagai berikut.



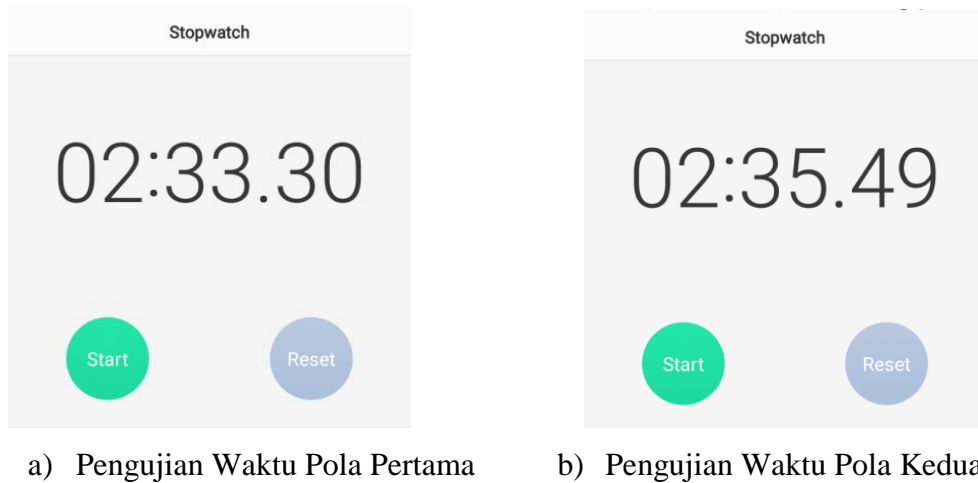
Gambar 4. 53 Simulasi Pengujian Pola Jalan Robot Pertama

Berikut adalah gambar 4.54 untuk simulasi pola jalan kedua robot pemetaan jarak tanam.



Gambar 4. 54 Simulasi Pengujian Pola Jalan Robot Kedua

Adapun hasil pengujian waktu pola jalan robot pertama pemetaan jarak tanam dengan luas lahan 2 m x 2 m, jarak tanam 0,5 m x 0,5 m terdapat pada gambar 4.55 dibawah ini.



Gambar 4. 55 Pengujian Waktu Pola Jalan Robot

Adapun hasil pengujian pola jalan robot pemetaan jarak tanam terdapat pada tabel 4.24 dibawah ini.

Tabel 4. 24 Hasil Pengujian Pola Jalan Robot Pemetaan Jarak Tanam

Luas lahan (m)	Jarak tanam (m)	Waktu pengerjaan robot (s)		Perhitungan waktu (s)	Error (%)
		Pertama	Kedua		
2 x 2	0,5 x 0,5	153	155	151,9	0,7%
4 x 4	1 x 1	166	171	162,7	2%
5 x 5	1 x 1	301	303	269,6	0,6%
Rata-rata <i>error</i>					1,1%

Adapun perhitungan untuk membandingkan waktu pengerjaan robot adalah sebagai berikut :

$$Waktu\ total = (jumlah\ titik\ tanam \cdot (waktu\ jalan\ per\ titik + waktu\ bor\ per\ titik)) + (jumlah\ belok \cdot waktu\ belok) \dots\dots\dots (4.6)$$

Keterangan :

Waktu total = waktu keseluruhan pengerjaan robot (s)

Jumlah titik tanam = jumlah tanaman

Waktu jalan per titik = waktu jalan robot antar titik (s)

Waktu bor per titik = waktu pengeboran untuk satu titik (s)

Jumlah belok = $\sqrt{\text{jumlah titik tanam}} + (\sqrt{\text{jumlah titik tanam}} - 2)$

Waktu belok = waktu untuk setiap satu belokan (s)

1. Luas lahan 2 m x 2 m, jarak tanam 0,5 m x 0,5 m

$$\begin{aligned} \text{Waktu total} &= (\text{jumlah titik tanam} \\ &\quad \cdot (\text{waktu jalan per titik} + \text{waktu bor per titik})) \\ &\quad + (\text{jumlah belok} \cdot \text{waktu belok}) \end{aligned}$$

$$\text{Waktu total} = (9 \cdot (1,2 \text{ s} + 15,5 \text{ s})) + (4 \cdot 0,4 \text{ s})$$

$$\text{Waktu total} = 151,9 \text{ s}$$

$$\text{error waktu} = \left| \frac{151,9 - 153}{153} \right| \cdot 100\% = 0,7 \%$$

2. Luas lahan 4 m x 4 m, jarak tanam 1 m x 1 m

$$\begin{aligned} \text{Waktu total} &= (\text{jumlah titik tanam} \\ &\quad \cdot (\text{waktu jalan per titik} + \text{waktu bor per titik})) \\ &\quad + (\text{jumlah belok} \cdot \text{waktu belok}) \end{aligned}$$

$$\text{Waktu total} = (9 \cdot (2,4 \text{ s} + 15,5 \text{ s})) + (4 \cdot 0,4 \text{ s})$$

$$\text{Waktu total} = 162,7 \text{ s}$$

$$\text{error waktu} = \left| \frac{162,7 - 166}{166} \right| \cdot 100\% = 2 \%$$

3. Luas lahan 5 m x 5 m, jarak tanam 1 m x 1 m

$$\text{Waktu total} = (16 \cdot (2,4 \text{ s} + 15,5 \text{ s})) + (6 \cdot 0,4 \text{ s})$$

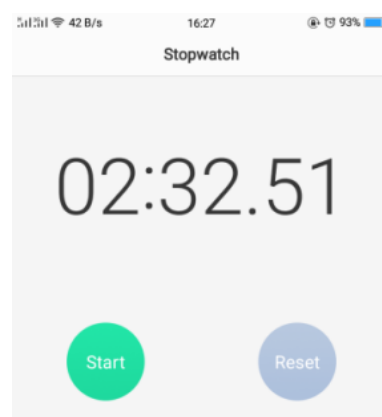
$$\text{Waktu total} = 269,6 \text{ s}$$

$$\text{error waktu} = \left| \frac{269,6 - 301}{301} \right| \cdot 100\% = 0,6 \%$$

Pada pengujian pola jalan robot dilakukan dengan dua pola jalan robot pemetaan jarak tanam yang berbeda yaitu pola pertama pada gambar 4.57 dan pola kedua pada gambar 4.58. Perbandingan waktu pengerjaan robot pemetaan jarak tanam antara pola pertama dan kedua didapatkan tidak jauh berbeda yang dapat dilihat pada tabel 4.24 diatas. Berdasarkan perbandingan waktu pengerjaan antara dua pola tersebut maka dapat disimpulkan pergerakan robot pemetaan jarak tanam di kedua pola yang diuji sama efektifnya sehingga dalam proyek akhir ini hanya menggunakan pola pertama untuk proses pemetaan jarak tanam, selisih waktu pengerjaan robot pemetaan jarak tanam antara pola pertama dan kedua dengan tiga kali pengujian adalah 2 detik, 5 detik dan 2 detik.

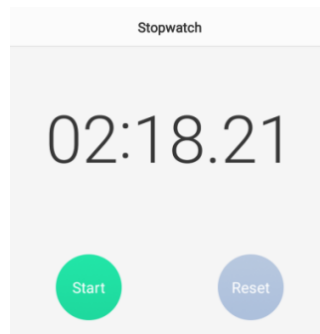
4.5.4 Pengujian Waktu Pengerjaan Pemetaan Jarak Tanam

Pengujian bertujuan untuk mengetahui lama waktu robot beroperasi selama melakukan pemetaan jarak tanam, baik pada luas lahan berukuran 2 m x 2m, 4 m x 4 m dan 5 mx 5 m. Pengujian dilakukan dengan cara membandingkan waktu pengerjaan terhadap robot pemetaan jarak tanam dan dengan cara manual. Sehingga bisa membandingkan waktu pengerjaan pemetaan jarak tanam lebih efektif dengan menggunakan robot pemetaan jarak tanam atau dengan cara manual. Berikut adalah hasil pengujian waktu pengerjaan pemetaan jarak tanam dengan menggunakan robot pemetaan jarak tanam diketahui luas lahan 2 m x 2 m jarak tanam 0,5 m x 0,5 m dapat dilihat pada gambar 4.56 dibawah ini.



Gambar 4. 56 Pengujian Waktu Menggunakan Robot

Berikut adalah hasil pengujian waktu pengerjaan pemetaan jarak tanam dengan menggunakan cara manual diketahui luas lahan 2 m x 2 m jarak tanam 0,5 m x 0,5 m dapat dilihat pada gambar 4.57 dibawah ini.



Gambar 4. 57 Pengujian Waktu Menggunakan Cara Manual

Adapun hasil pengujian waktu yang dilakukan menggunakan robot dan menggunakan cara manual, dapat dilihat pada tabel 4.25 sebagai berikut.

Tabel 4. 25 Hasil Pengujian Waktu Pemetaan Jarak Tanam

Luas lahan (m)	Jarak tanam (m)	Waktu pengerjaan (detik)	
		Robot	Manual
2 x 2	0,5 x 0,5	139	130
4 x 4	1 x 1	147	138
5 x 5	1 x 1	301	255

Setelah melakukan pengujian waktu antara menggunakan robot dan dengan cara manual dapat dilihat pada tabel 4.25 bahwa selisih waktu pengerjaan pemetaan jarak tanam menggunakan robot dan cara manual dengan tiga kali pengujian adalah 9 detik, 9 detik dan 46 detik. Semakin luas lahan dan banyak titik tanam yang dikerjakan maka waktu pengerjaan pemetaan jarak tanam semakin lama.

4.5.5 Pengujian Pemakaian dan Pengisian Baterai

Pengujian pemakaian baterai bertujuan untuk mengetahui berapa lama pemakaian baterai pada saat robot pemetaan jarak tanam sedang beroperasi dan

bertujuan untuk mengetahui berapa lama pengisian baterai. Adapun kapasitas baterai yang digunakan dalam proyek akhir yaitu 12 V 9 Ah. Pengujian pemakaian baterai dilakukan dengan cara pengukuran berupa pemakaian arus dan tegangan terhadap robot pemetaan jarak tanam sehingga bisa mengetahui daya total dari robot. Sedangkan untuk pengujian pengisian baterai dilakukan dengan cara menghitung lama waktu baterai pada saat di *charger*. Berikut adalah salah satu hasil dari pengujian pemakaian arus dan tegangan dapat dilihat pada gambar 4. 58 berikut.



Gambar 4. 58 Pengujian Pengukuran Arus Pada Motor DC *Power Window*

Adapun hasil pengukuran arus dan tegangan robot pemetaan jarak tanam dapat dilihat pada tabel 4.26 sebagai berikut.

Tabel 4. 26 Hasil Pemakaian Arus dan Tegangan Robot Pemetaan Jarak Tanam

Komponen	Arus (A)	Tegangan (V)	Daya (W)
Motor <i>power window</i> 1 dan 2	2,88	12	34,56
Motor pengeboran	2,65	9,59	25,41
Motor penurun/penaik Bor	0,33	12	3,96
Sistem pengeboran naik	0,45	12	5,4
Arduino Uno dengan sensor tegangan DC, sensor <i>rotary encoder</i> KY-040, sensor kompas GY-271	1,04	4,93	5,13
NodeMCU ESP8266	0,1	3,24	0,324
Jumlah Daya Total Pemakaian Sistem Kontrol Robot			75,784

Setelah melakukan pengujian terhadap pemakaian arus dan tegangan didapatkan daya total keseluruhan adalah 75,784 W

1. Waktu Pemakaian Baterai

Diketahui : Kapasitas baterai = 12 V 9 AH

Daya total Pemakaian = 75,784 W

$$\begin{aligned} \text{Daya Baterai} &= V \cdot I \\ &= 12 V \times 9 Ah = 108 Wh \end{aligned}$$

$$\text{Waktu pemakaian} = \frac{108 Wh}{75,784 W} = 1,425 h \text{ (85 menit 5 detik)}$$

2. Waktu Pengisian Baterai

Adapun rumus waktu pengisian baterai adalah sebagai berikut :

$$\text{Waktu pengisian} = \frac{\text{Arus Baterai}}{\text{Arus Charger}} \dots\dots\dots (4.7)$$

Diketahui : Kapasitas baterai = 9 Ah

Arus pengisian baterai = 4,5 A

$$\text{Waktu pengisian} = \frac{9 Ah}{4,5 A} = 2 h \text{ (2 jam)}$$

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa lama waktu pemakaian baterai secara perhitungan adalah selama 85 menit 5 detik dan waktu pengisian baterai selama 2 jam. Dengan catatan bahwa penggunaan konsumsi tegangan baterai dalam kondisi penuh/*full*.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian sistem robot pemetaan jarak tanam dapat disimpulkan bahwa :

1. Robot dapat melakukan pemetaan jarak tanam tergantung pada keinginan pengguna dengan memasukkan data berupa luas lahan dan jarak tanam melalui *smartphone* dengan waktu pengiriman data dari *smartphone* ke modul *wifi* yang diolah mikrokontroller adalah kurang dari 15 detik dan jangkauan koneksi antara *smartphone* dengan modul *wifi* berjarak maksimal 100 m.
2. Proses pemetaan dan pengeboran jarak tanam yang dilakukan oleh robot belum akurat dengan nilai rata-rata *error* koordinat X 2,66%, koordinat Y 3% dan jarak tanam 4%, hal ini disebabkan oleh pengaruh putaran roda yang selip, ketidakpresisian pembacaan sensor *rotary encoder* dan sensor kompas.
3. Robot dapat melakukan pemetaan jarak tanam dengan waktu selama 85 menit 5 detik dan pengisian baterai selama 2 jam menggunakan spesifikasi baterai 12 V 9 Ah.

5.2 Saran

Untuk pengembangan lebih lanjut ada beberapa saran guna pembuatan sistem robot pemetaan jarak tanam yang lebih baik, meliputi :

1. Untuk memperkecil *error* putaran roda karena selip pada proses pemetaan jarak tanam yang dilakukan robot, perlu memilih penggunaan jenis roda depan dan belakang yang sesuai agar bisa bergerak lancar pada kondisi tanah yang berbeda-beda.
2. Untuk memperkecil *error* koordinat titik pengeboran (X,Y) dan jarak tanam yang dilakukan robot, pemasangan konstruksi sistem pengeboran harus berada di titik tengah (*center*) *base* robot dan penggunaan sensor posisi (*rotary encoder*) serta sensor kompas yang lebih presisi pembacaannya.

3. Untuk pemakaian daya yang lebih lama pada robot pemetaan jarak tanam menggunakan baterai dengan kapasitas yang lebih besar.

DAFTAR PUSTAKA

- [1] B. T. R. Erawati dan A. Hipi, "Pengaruh Jarak Tanam terhadap Pertumbuhan dan Hasil Beberapa Varietas Jagung Hibrida di Kawasan Pengembangan Jagung Kabupaten Sumbawa," *Jurnal Pertanian*, p. 609, 2016.
- [2] A. Saefullah, D. Immaniar dan A. R. Juliansah, "Sistem Kontrol Robot Pemindah Barang Menggunakan Aplikasi Android Berbasis Arduino Uno," *REDAKSI Creative Communtion And innovative Technology*, vol. 08, no. 02, Januari 2015.
- [3] R. M. A, "Rancang Bangun Robot Penanam Benih Jagung," Makasar, 2018.
- [4] M. F. Iqbal, "Digital library - Perpustakaan Pusat Unikom - Knowledge Center," 20 januari 2014. [Online]. Available: <https://elib.unikom.ac.id>. [Diakses 31 Agustus 2019].
- [5] T. F. Rajagukguk, C. V. Poekoel dan D. M. Putro, "Implementasi WSN Pada Robot Penyiram Tanaman Otomatis," *Teknik Elektro dan Komputer*, vol. VII, no. 1, 2018.
- [6] Aurélien Demaurex, "ecorobotix," [Online]. Available: <https://www.ecorobotix.com>. [Diakses 4 September 2019].
- [7] B. W. S. R, "Modul Pelatihan Penanaman Pohon", Jambi: ITTO Training Proceedings, 2006.
- [8] H. Bench, "A Place ForTthe Electronic Hobbyist," 16 Juni 2016. [Online]. Available: <http://henrysbench.capnfatz.com>. [Diakses 27 April 2019].

LAMPIRAN 1
Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Noviansyah
Tempat & Tanggal Lahir : Kimak, 22 November 1998
Alamat Rumah : Jl. Pemuda Desa Kimak, Kec. Merawang
Jenis Kelamin : Laki-laki
Agama : Islam
No.Handphone : 0822-7967-3911
Email : noviansyahbaru22@gmail.com

2. Riwayat Pendidikan

SDN 18 Kimak : 2004-2010
SMPN 3 Pemali : 2010-2013
SMAN 1 Pemali : 2013-2016

3. Pengalaman Kerja

Praktek Kerja Lapangan di PT. PLN (Persero) Unit Pelaksanaan Pembangkitan Bukit Asam periode 03 September 2019 – 28 Desember 2019

4. Pengetahuan Bahasa

Bahasa yang dikuasai : Bahasa Indonesia

5. Hobi

- Olahraga
- Traveling

Sungailiat, 21 Agustus 2019

Noviansyah

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Rafirda Aini
Tempat & Tanggal Lahir : Kelumbi, 26 Juni 1998
Alamat Rumah : Jl. Raya Desa Buyan Kelumbi, Kec. Tempilang
Jenis Kelamin : Perempuan
Agama : Islam
No.Handphone : 0812-7343-1427
Email : rafirda26@gmail.com

2. Riwayat Pendidikan

SDN 13 Tempilang : 2004-2010
SMPN 2 Puding Besar : 2010-2013
SMKN 1 Tempilang : 2013-2016

3. Pengalaman Kerja

Praktek Kerja Lapangan di PT. Keihin Indonesia periode 13 Agustus 2019 – 13 Desember 2019

4. Pengetahuan Bahasa

Bahasa yang dikuasai : Bahasa Indonesia

5. Hobi

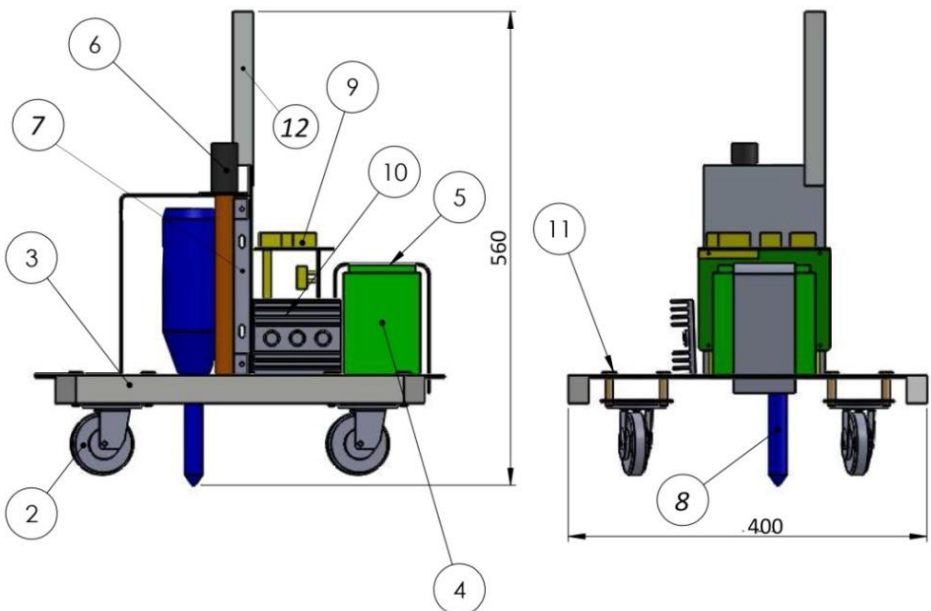
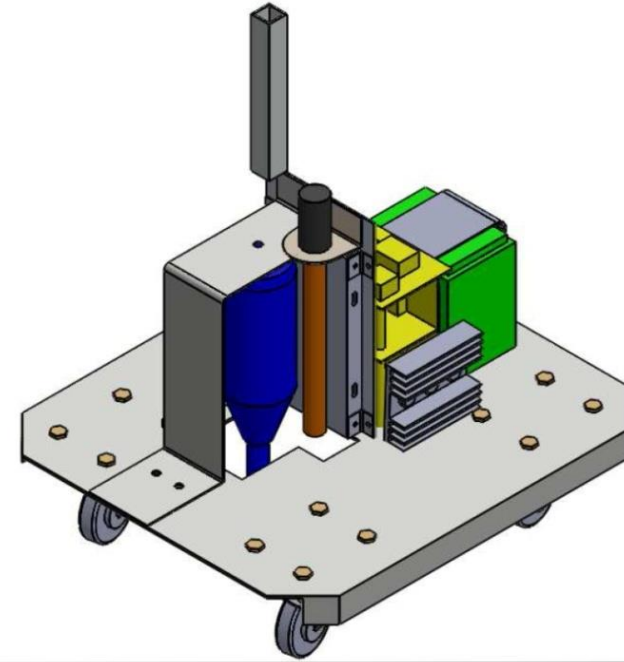
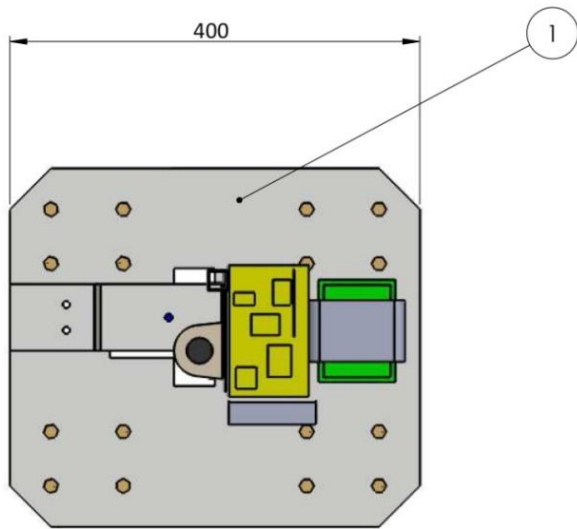
- Olahraga
- Membaca novel

Sungailiat, 21 Agustus 2019

Rafirda Aini

LAMPIRAN 2

Desain Robot Pemetaan Jarak
Tanam Berbasis *Smartphone*



1	Dudukan Sensor Kompas	12		40x210	
16	Baut	11	As Baja	M8x100	std
1	Headsink Transistor 2N3055	10		150x100	std
1	Tempat Komponen Elektrik	9	Acrylic	200x110	
1	Mata Bor	8		O80	std
1	Dudukan Sistem Pengeboran	7	Steel	180x200	std
1	Penaik/Turun Bor	6	ST 37	M8	
1	Plat Penahan Baterai	5	Alumunium	185x75	
1	Baterai	4		135x75x135	std
2	Penyangga Base	3	Alumunium	330x25x25	
4	Roda	2		O100	std
1	Base	1	Alumunium	400x400	
Jumlah	Nama Bagian	No. Bag.	Bahan	Ukuran	Ket.

III	II	I	Perubahan	Pengganti Dari Diganti Dengan	
<p style="text-align: center;">Robot Pemetaan Jarak Tanam Berdasarkan Smartphone</p>				Skala	Digambar
				1 : 1	Diperiksa
					Dilihat

POLITEKNIK MANUFaktur NEGERI BABEL

001/TA/2019

LAMPIRAN 3
Program Keseluruhan

```

                /*PROGRAM ARDUINO*/
//INISIALIASI KOMUNIKASI
#include <SoftwareSerial.h>
SoftwareSerial arduino_serial(10, 11);

//INISIALIASI MOTOR DC
#define Relay1 2 //pengebor
#define Relay2 3
#define R_M1 5 //motor kanan berlawanan arah
#define L_M1 4 //motor kanan searah
#define R_M2 7 //motor kiri berlawanan arah
#define L_M2 6 //motor kiri searah
#define R_M3 8 //penurun bor
#define L_M3 12 //penaik bor

//INISIALIASI Sensor Kompas
#include <Wire.h>
#include <QMC5883L.h>
QMC5883L compass;
int x, y, z;
int sudut_sekarang;
uint16_t sudut_diinginkan;
int16_t range_sudut;

//INISIALIASI ROTARY ENCODER MOTOR KANAN
const int pinCLK_kanan = 13; // kaki CLK ke pin 13
const int pinDT_kanan;
int encoder; // hitungan nilai sensor
int akhir_kanan;
int awal_kanan;
boolean nilai_kanan;

//INSIALIASI UNTUK KONDISI JALAN ROBOT
int aman = 100;
int kondisi = 100;
int jalan = 0;
short pwm_maju = 150;
short pwm_belok = 140;
short pwm_kurang = 110;
int p;

//INISIALIASI SENSOR TEGANGAN
const int sensor_tegangan = A1;
int Volt1;
float tegangan;
float persen;

```

```

/////INISIALISASI TERIMA DATA DARI NODEMCU
float r, s, t;
float s1, s2, s3;
int v;
/*-----FUNGSI-FUNGSI SEMUA KOMPONEN-----*/
//-----FUNGSI MOTOR-----//
void robot_maju()
{
    analogWrite(R_M1, 0); analogWrite(L_M1, 255); //240
    analogWrite(R_M2, 0); analogWrite(L_M2, 255); //145
}
void percepat()
{
    pwm_maju = pwm_maju + 5;
    if (pwm_maju > 255)
    {
        pwm_maju = 255;
    }
    analogWrite(R_M1, 0); analogWrite(L_M1, pwm_maju);
    analogWrite(R_M2, 0); analogWrite(L_M2, pwm_maju);
}
void robot_kanan()
{
    pwm_belok = pwm_belok + 2;
    if (pwm_belok > 170)
    {
        pwm_belok = 170;
    }
    analogWrite(R_M1, 0); analogWrite(L_M1, 0);
    analogWrite(R_M2, 0); analogWrite(L_M2, pwm_belok);
}
void robot_kiri()
{
    pwm_belok = pwm_belok + 2;
    if (pwm_belok > 170)
    {
        pwm_belok = 170;
    }
    analogWrite(R_M1, 0); analogWrite(L_M1, pwm_belok);
    analogWrite(R_M2, 0); analogWrite(L_M2, 0);
}
void robot_mundur()
{
    analogWrite(R_M1, 255); analogWrite(L_M1, 0);
    analogWrite(R_M2, 255); analogWrite(L_M2, 0);
}
void robot_kanan11()

```

```

{
  analogWrite(R_M1, 0); analogWrite(L_M1, 0);
  analogWrite(R_M2, 0); analogWrite(L_M2, 140);
}
void robot_kirill()
{
  analogWrite(R_M1, 0); analogWrite(L_M1, 140);
  analogWrite(R_M2, 0); analogWrite(L_M2, 0);
}
void robot_berhenti()
{
  analogWrite(R_M1, 0); analogWrite(L_M1, 0);
  analogWrite(R_M2, 0); analogWrite(L_M2, 0);
}
void bor_turun()
{
  analogWrite(R_M3, 255); analogWrite(L_M3, 0);
  digitalWrite(Relay1, LOW); digitalWrite(Relay2,
HIGH);
}
void bor_naik()
{
  analogWrite(R_M3, 0); analogWrite(L_M3, 255);
  digitalWrite(Relay1, HIGH); digitalWrite(Relay2, HIGH);
}
void bor_berhenti()
{
  analogWrite(R_M3, 0); analogWrite(L_M3, 0);
  digitalWrite(Relay1, HIGH); digitalWrite(Relay2, HIGH);
}
void motor_init() //Inisialisai pin motor di void Setup
{
  pinMode(R_M1, OUTPUT); pinMode(L_M1, OUTPUT);
  pinMode(R_M2, OUTPUT); pinMode(L_M2, OUTPUT);
  pinMode(R_M3, OUTPUT); pinMode(L_M3, OUTPUT);
  pinMode(Relay1, OUTPUT); pinMode(Relay2, OUTPUT);

  robot_berhenti();
  bor_berhenti();
  delay(100);
}
//-----SENSOR KOMPAS-----//
void baca_kompas()
{
  compass.read(&x, &y, &z);
  // menghitung heading pada sensor nilai sumbu x dan y
  // 'atan2()' untuk memeriksa formula pada kuadran

```

```

float heading = atan2(y, x);
float declinationAngle = 0.0404;
heading += declinationAngle;
if (heading < 0)
    heading += 2 * PI;
if (heading > 2 * PI)
    heading -= 2 * PI;
sudut_sekarang = heading * 180 / M_PI; //konversi
//radius ke derajat
// Serial.print("Sudut: ");
// Serial.println(sudut_sekarang);
// delay(10);
}
//-----ROTARY ENCODER-----//
void rotary()
{
    awal_kanan = digitalRead(pinCLK_kanan);
    if (awal_kanan != akhir_kanan)
    {
        if (digitalRead(pinDT_kanan) != awal_kanan)
        {
            nilai_kanan = true; encoder++;
        }
        else
        {
            nilai_kanan = false; encoder++;
        }
        Serial.println(encoder); //Menampilkan nilai encoder
    }
    akhir_kanan = awal_kanan;
}
void re_init() //Inisialisasi pin Rotary di Void Setup
{
    pinMode (pinCLK_kanan, INPUT_PULLUP);
    pinMode (pinDT_kanan, INPUT_PULLUP);
    akhir_kanan = digitalRead(pinCLK_kanan);
    attachInterrupt (digitalPinToInterrupt (pinCLK_kanan),
rotary, RISING);
}
void se_tegangan()
{
    Volt1 = analogRead(sensor_tegangan);
    tegangan = ((Volt1 - 3) * 0.0247);
    if ((Volt1 - 4) <= 0 || tegangan <= 0)
    {
        persen = 0;
    }
}

```



```

else
{
    persen = map(Volt1, 450, 570, 35, 100);
    if (persen <= 0)
    {
        persen = 0;
    }
    else {persen = persen;}
}
Serial.print("aa "); Serial.println(persen);
Serial.print("Voltage (V): ");
Serial.println(tegangan);
delay(1000);
}

void setup()
{
    Wire.begin();
    Serial.begin(9600);
    while (!Serial) {
        ; // tunggu port serial untuk terhubung
    }
    arduino_serial.begin(9600);

    compass.init();
    motor_init();
    re_init();
    aman = 102;
    kondisi = 102;
    jalan = 0;
    encoder = -1;
}

void loop()
{
    //-----Fungsi mengirim data-----//
    baca_kompas();
    arduino_serial.print(sudut_sekarang);
    arduino_serial.print('j');

    se_tegangan();
    arduino_serial.print(persen);
    arduino_serial.print('i');
    //-----//

    int a; //banyak jumlah titik tanam
    int j = 115; //nilai encoder per 1 meter

```

```

int k = 0;
int l = 0;
int m = 0;
uint16_t ss1;
uint16_t ss2;
uint16_t ssbelok;

//-----Fungsi menerima data-----//
while (arduino_serial.available() > 0)
{
    r = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'k')
    {
        Serial.print("L"); Serial.println(r);
    }
    s = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'l')
    {
        j = j-2;
        k = 3;
        l = 5;
    }
    t = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'm')
    {
        a = t;
//        Serial.print("JMM"); Serial.println(a);
    }
    s1 = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'o')
    {
        ss1 = s1;
//        Serial.print("S1"); Serial.println(s1);
    }
    s2 = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'p')
    {
        ss2 = s2;
//        Serial.print("S2"); Serial.println(s2);
    }
    s3 = arduino_serial.parseFloat();
    if (arduino_serial.read() == 'q')
    {
        ssbelok = s3;
//        Serial.print("SB"); Serial.println(s3);
    }
}

```

```

//-----//

int b = sqrt(a); //banyak jumlah pola
int c = s * j; //jarak tanam yang dijalankan
int f = c - 37; //untuk belok kanan
int g = 35; //kejalur dari belok kanan
int h = c - 50 - 1; //untuk belok kiri
int i = 42; //kejalur dari belok kiri

uint16_t sudut_jalur1 = ss1; //sudut jalur 1
uint16_t sudut_jalur2 = ss2; //sudut jalur 1
uint16_t sudut_belok = ssbelok; //sudut belok

while (jalan != a)
{
  arduino_serial.print(jalan);
  arduino_serial.print('h');

  //robot mengebor titik awal
  if (aman == 102 && kondisi == 102)
  {
    bor_turun(); delay(8000);
    bor_naik(); delay(7500);
    bor_berhenti(); delay(100);
    encoder = -1;
    jalan++;
    aman = 0;
    kondisi = 0;
  }

  //robot maju kedepan jalur 1
  if (aman >= 0 && aman <= (b - 2) && kondisi == 0)
  {
    rotary();
    baca_kompas();
    sudut_diinginkan = sudut_jalur1;
    if (sudut_sekarang < sudut_diinginkan)
    {
      range_sudut=sudut_diinginkan - sudut_sekarang;
      if (range_sudut > 4 && range_sudut < 180)
      {
        robot_kanan11();
      }
      if (range_sudut > 180)
      {
        robot_kiri11();
      }
    }
  }
}

```

```

    }
}
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 4 && range_sudut < 180)
    {
        robot_kirill();
    }
    if (range_sudut > 180)
    {
        robot_kananl1();
    }
}
if (range_sudut >= 0 && range_sudut <= 3)
{
    percepat();
    if (encoder >= c && encoder <= (c + 10))
    {
        robot_berhenti();
        bor_turun(); delay(8000);
        bor_naik(); delay(7500);
        bor_berhenti(); delay(100);
        encoder = -1;
        aman++;
        jalan++;
    }
}
}

//robot maju sedikit untuk belok kanan1
if (aman == (b - 1) && kondisi == 0)
{
    robot_maju(); delay(165);
    robot_berhenti(); delay(50);
    aman++;
    kondisi = 1;
}

if (aman == b && kondisi == 1) //robot belok kanan1
{
    baca_kompas();
    sudut_diinginkan = sudut_belok;
    if (sudut_sekarang < sudut_diinginkan)
    {
        range_sudut=sudut_diinginkan - sudut_sekarang;
        if (range_sudut > 6 && range_sudut < 180)

```

```

    {
        robot_kanan();
    }
    if (range_sudut > 180)
    {
        robot_kanan();
    }
}
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 6 && range_sudut < 180)
    {
        robot_kanan();
    }
    if (range_sudut > 180)
    {
        robot_kanan();
    }
}
if (range_sudut >= 0 && range_sudut <= 6)
{
    robot_berhenti();
    encoder = -1;
    aman++;
    kondisi = 1;
}
}

//robot maju mengebor pindah jalur
if (aman == (b + 1) && kondisi == 1)
{
    rotary();
    baca_kompas();
    sudut_diinginkan = sudut_belok;
    if (sudut_sekarang < sudut_diinginkan)
    {
        range_sudut=sudut_diinginkan - sudut_sekarang;
        if (range_sudut > 4 && range_sudut < 180)
        {
            robot_kanan11();
        }
        if (range_sudut > 180)
        {
            robot_kiri11();
        }
    }
}
}

```

```

else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 4 && range_sudut < 180)
    {
        robot_kirill1();
    }
    if (range_sudut > 180)
    {
        robot_kanan11();
    }
}
if (range_sudut >= 0 && range_sudut <= 3)
{
    percepat();
    if (encoder >= f && encoder <= (f + 10))
    {
        robot_berhenti();
        bor_turun(); delay(8000);
        bor_naik(); delay(7500);
        bor_berhenti(); delay(100);
        encoder = -1;
        aman++;
        jalan++;
        kondisi = 2;
    }
}
}

//robot maju sedikit untuk belok kanan2
if (aman == (b + 2) && kondisi == 2)
{
    robot_maju(); delay(160);
    robot_berhenti(); delay(50);
    aman++;
    kondisi = 2;
}

//robot belok kanan2
if (aman == (b + 3) && kondisi == 2)
{
    baca_kompas();
    sudut_diinginkan = sudut_jalur2;
    if (sudut_sekarang < sudut_diinginkan)
    {
        range_sudut=sudut_diinginkan - sudut_sekarang;
        if (range_sudut > 6 && range_sudut < 180)

```

```

        {
            robot_kanan();
        }
        if (range_sudut > 180)
        {
            robot_kanan();
        }
    }
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 6 && range_sudut < 180)
    {
        robot_kanan();
    }
    if (range_sudut > 180)
    {
        robot_kanan();
    }
}
if (range_sudut >= 0 && range_sudut <= 5)
{
    robot_berhenti();
    kondisi = 200;
}
}

if (aman == (b + 3) && kondisi == 200)
{
    robot_berhenti();
    encoder = g;
    aman++;
    kondisi = 3;
}

//robot maju kedepan jalur 2
if (aman >= (b+4) && aman <= (b+b+2) && kondisi==3)
{
    rotary();
    baca_kompas();
    sudut_diinginkan = sudut_jalur2;
    if (sudut_sekarang < sudut_diinginkan)
    {
        range_sudut=sudut_diinginkan - sudut_sekarang;
        if (range_sudut > 4 && range_sudut < 180)
        {
            robot_kanan11();
        }
    }
}

```

```

    }
    if (range_sudut > 180)
    {
        robot_kirill();
    }
}
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 4 && range_sudut < 180)
    {
        robot_kirill();
    }
    if (range_sudut > 180)
    {
        robot_kananll();
    }
}
if (range_sudut >= 0 && range_sudut <= 3)
{
    percepat();
    if (encoder >= c && encoder <= (c + 10))
    {
        robot_berhenti();
        bor_turun(); delay(8000);
        bor_naik(); delay(7500);
        bor_berhenti(); delay(100);
        encoder = -1;
        aman++;
        jalan++;
    }
}
}

//robot maju sedikit untuk belok kiril
if (aman == (b + b + 3) && kondisi == 3)
{
    robot_maju(); delay(110);
    robot_berhenti(); delay(50);
    aman++;
    kondisi = 4;
}

//robot belok kiril
if (aman == (b + b + 4) && kondisi == 4)
{
    baca_kompas();
}

```



```

sudut_diinginkan = sudut_belok; //-
if (sudut_sekarang < sudut_diinginkan)
{
    range_sudut=sudut_diinginkan - sudut_sekarang;
    if (range_sudut > 5 && range_sudut < 180)
    {
        robot_kiri();
    }
    if (range_sudut > 180)
    {
        robot_kiri();
    }
}
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 5 && range_sudut < 180)
    {
        robot_kiri();
    }
    if (range_sudut > 180)
    {
        robot_kiri();
    }
}
if (range_sudut >= 0 && range_sudut <= 5)
{
    robot_berhenti();
    encoder = -10;
    aman++;
    kondisi = 5;
}
}

//Robot maju mengebor pindah jalur 3
if (aman == (b + b + 5) && kondisi == 5)
{
    rotary();
    baca_kompas();
    sudut_diinginkan = sudut_belok;
    if (sudut_sekarang < sudut_diinginkan)
    {
        range_sudut=sudut_diinginkan - sudut_sekarang;
        if (range_sudut > 4 && range_sudut < 180)
        {
            robot_kanan11();
        }
    }
}

```

```

        if (range_sudut > 180)
        {
            robot_kiri11();
        }
    }
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 4 && range_sudut < 180)
    {
        robot_kiri11();
    }
    if (range_sudut > 180)
    {
        robot_kanan11();
    }
}
if (range_sudut >= 0 && range_sudut <= 3)
{
    percepat();
    if (encoder >= h && encoder <= (h + 10))
    {
        robot_berhenti();
        bor_turun(); delay(8000);
        bor_naik(); delay(7500);
        bor_berhenti(); delay(100);
        encoder = -1;
        aman++;
        jalan++;
        kondisi = 6;
    }
}
}

//robot maju sedikit untuk belok kiri2
if (aman == (b + b + 6) && kondisi == 6)
{
    robot_maju(); delay(80);
    robot_berhenti(); delay(50);
    aman++;
    kondisi = 7;
}

//robot belok kiri2
if (aman == (b + b + 7) && kondisi == 7)
{
    baca_kompas();
}

```

```

sudut_diinginkan = sudut_jalur1;
if (sudut_sekarang < sudut_diinginkan)
{
    range_sudut=sudut_diinginkan - sudut_sekarang;
    if (range_sudut > 5 && range_sudut < 180)
    {
        robot_kiri();
    }
    if (range_sudut > 180)
    {
        robot_kiri();
    }
}
else if (sudut_sekarang > sudut_diinginkan)
{
    range_sudut=sudut_sekarang - sudut_diinginkan;
    if (range_sudut > 5 && range_sudut < 180)
    {
        robot_kiri();
    }
    if (range_sudut > 180)
    {
        robot_kiri();
    }
}
if (range_sudut >= 0 && range_sudut <= 5)
{
    robot_berhenti();
    kondisi = 300;
}
}

if (aman == (b + b + 7) && kondisi == 300)
{
    robot_berhenti();
    encoder = i;
    aman++;
    kondisi = 8;
}

//robot maju kedepan jalur 3 (sama jalur 1)
if (aman >= (b+b+8)&&aman <= (b+b+b+6)&&kondisi==8)
{
    rotary();
    baca_kompas();
    sudut_diinginkan = sudut_jalur1;
    if (sudut_sekarang < sudut_diinginkan)

```

```

{
  range_sudut=sudut_diinginkan - sudut_sekarang;
  if (range_sudut > 4 && range_sudut < 180)
  {
    robot_kanan11();
  }
  if (range_sudut > 180)
  {
    robot_kirill1();
  }
}
else if (sudut_sekarang > sudut_diinginkan)
{
  range_sudut=sudut_sekarang - sudut_diinginkan;
  if (range_sudut > 4 && range_sudut < 180)
  {
    robot_kirill1();
  }
  if (range_sudut > 180)
  {
    robot_kanan11();
  }
}
if (range_sudut >= 0 && range_sudut <= 3)
{
  percepat();
  if (encoder >= c && encoder <= (c + 10))
  {
    robot_berhenti();
    bor_turun(); delay(8000);
    bor_naik(); delay(7500);
    bor_berhenti(); delay(100);
    encoder = -1;
    aman++;
    jalan++;
  }
}
}

if (aman == (b + b + b + 7) && kondisi == 8)
{
  robot_berhenti();
  aman = (b - 1);
  kondisi = 0;
  encoder = -1;
}

```

```
//robot berhenti ketika proses pengerjaan selesai
if (jalan==a)
{
    robot_berhenti();
    bor_berhenti();
}

//memastikan robot benar-benar berhenti
while (jalan == (a + 1))
{
    robot_berhenti();
    bor_berhenti();
    aman = 400;
    kondisi = 400;
}
}
```

```

/* PROGRAM NODEMCU ESP8266 */
//INISIALISASI KOMUNIKASI
#include <SoftwareSerial.h>
SoftwareSerial nodemcu_serial(D6, D5);

//INISIALISASI BLYNK
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
BlynkTimer timer;

//KONEKSI WIFI KE SMARTPHONE
char auth[]="de2818372d0d4f089cb5749eebbba712"; //token
email
const char ssid[] = "robot"; //nama wifi
const char pass[] = "proyekakhir2019"; //password wifi

//INISIALISASI TERIMA DAN KIRIM DATA
float jarak_tanam, jarak, jarak1;
float daya, aki=0, aki1=0;
float sudut1, sudut2, belok, kompas, sudut;
float sudutl1, sudutl2, belok1;
int luas_lahan, jumlah_tanaman, lahan, lahan1;
int pertama, kedua, ketiga;
int titik_tanam=0, terima=0;

//----- FUNGSI-FUNGSI -----
---//
//Fungsi Menginput Data LUAS LAHAN dan JARAK TANAM
BLYNK_WRITE(V1) //LUAS LAHAN
{
  lahan1 = param.asInt();
  Serial.println(lahan1);
}
BLYNK_WRITE(V2) //JARAK TANAM
{
  jarak1 = param.asFloat();
  Serial.println(jarak1);
}

//Fungsi Menginput SUDUT
BLYNK_WRITE(V21) //jalur 1
{
  sudutl1 = param.asInt();
  Serial.println(sudutl1);
}

```

```

}
BLYNK_WRITE(V22) //Jalur 2
{
  sudut12 = param.asFloat();
  Serial.println(sudut12);
}
BLYNK_WRITE(V23) //belok
{
  belok1 = param.asFloat();
  Serial.println(belok1);
}

//Fungsi tombol ON/OFF
BLYNK_WRITE(V3) //Input Luas Lahan && Jarak Tanam
BENAR/TIDAK
{
  if (lahan1 < jarak1)
  {
    lahan1 = 0;
    jarak1 = 0;
  }
  int pin_V1 = param.asInt();
  if (pin_V1 == HIGH)
  {
    lahan = lahan1;
    jarak = jarak1;
    Serial.println("YA");
    luas_lahan = lahan * lahan;
    // Serial.print("Luas Lahan :");
    // Serial.println(luas_lahan);
    jarak_tanam = jarak * jarak;
    // Serial.print("Jarak Tanam :");
    // Serial.println(jarak_tanam);
  }
  else {
    Serial.println("TIDAK");
  }
}
BLYNK_WRITE(V4) //Mengirim Jumlah Tanaman ke Arduino
{
  int pin_V2 = param.asInt();
  if (pin_V2 == HIGH)
  {
    Serial.println("SETUJU");
    pertama = lahan - (2 * jarak); //lahan
    kedua = (pertama * pertama) / jarak_tanam;
    ketiga = (pertama / jarak) * 2;
  }
}

```

```

        jumlah_tanaman = 2 + kedua + (ketiga - 1);
        Serial.print("Jumlah Tanaman : ");
        Serial.println(jumlah_tanaman);
    }
    else {
        Serial.println("TIDAK");
    }
}
BLYNK_WRITE(V5) //Inputan sudut BENAR/TIDAK
{
    int pin_V5 = param.asInt();
    if (pin_V5 == HIGH)
    {
        sudut1 = sudut11;
        sudut2 = sudut12;
        belok = belok1;
        // Serial.print("S1 :"); Serial.println(sudut1);
        // Serial.print("S2 :"); Serial.println(sudut2);
        // Serial.print("B1 :"); Serial.println(belok);
    }
    else {
        Serial.println("TIDAK");
    }
}

void setup()
{
    Serial.begin(9600);
    //Serial.flush();
    while (!Serial) {
        ; //tunggu port serial untuk terhubung
    }
    nodemcu_serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
}

void loop()
{
    Blynk.run();
    timer.run();
    //-----Menampilkan Data di Smartphone-----
    //
    Blynk.virtualWrite(V7, jumlah_tanaman);
    Blynk.virtualWrite(V13, lahan1);
    Blynk.virtualWrite(V13, jarak1);
    Blynk.virtualWrite(V24, sudut1);
    Blynk.virtualWrite(V25, sudut2);
}

```



```

    Blynk.virtualWrite(V26, belok);
//-----
//

//-----Menerima Data di NodeMCU-----
//
while (nodemcu_serial.available() > 0)
{
    terima = nodemcu_serial.parseFloat();
    if (nodemcu_serial.read() == 'h')
    {
        Serial.println(terima);
    }
    daya = nodemcu_serial.parseFloat();
    if (nodemcu_serial.read() == 'i')
    {
        Serial.print("AA"); Serial.println(daya);
        aki = daya;
    }
    kompas = nodemcu_serial.parseFloat();
    if (nodemcu_serial.read() == 'j')
    {
        Serial.println(kompas);
    }
    titik_tanam = terima;
    akil = aki;
    sudut = kompas;
    Blynk.virtualWrite(V8, titik_tanam); //Tampil Titik
Tanam
    Blynk.virtualWrite(V9, akil); //Menampilkan daya
aki
    Blynk.virtualWrite(V20, sudut); //Menampilkan sudut
kompas
}

//-----Mengirim Data ke Arduino-----
//
    nodemcu_serial.print(lahan); //luas lahan
    nodemcu_serial.print('k');

    nodemcu_serial.print(jarak); // jarak tanam
    nodemcu_serial.print('l');

    nodemcu_serial.print(jumlah_tanaman); // jumlah
tanaman
    nodemcu_serial.print('m');

```

```
nodemcu_serial.print(sudut1); //nilai sudut jalur1
nodemcu_serial.print('o');

nodemcu_serial.print(sudut2); //nilai sudut jalur2
nodemcu_serial.print('p');

nodemcu_serial.print(belok); //nilai sudut belok
nodemcu_serial.print('q');
//-----//
}
```

LAMPIRAN 4

Standar Operasional Prosedur (SOP)

Robot Pemetaan Jarak

Tanam Berbasis *Smartphone*

Standar Operasional Prosedur (SOP)
Robot Pemetaan Jarak Tanam Berbasis *Smartphone*

1. Hidupkan mikrokontroler dan motor DC dengan masing-masing *switch* yang sudah terpasang pada robot.
2. Buka aplikasi *blynk* pada *smartphone*.
3. Koneksikan aplikasi *blynk* ke *hotspot wifi smartphone*.
4. Setelah jaringan *hotspot wifi* terhubung, selanjutnya *user* menginput data nilai sudut, luas lahan dan jarak tanam yang diinginkan.
5. Setelah setuju, *user* mengklik tombol ON.
6. Lalu *smartphone* akan mengirimkan data ke modul *wifi*, selanjutnya modul *wifi* akan mengirim data untuk diolah mikrokontroler.
7. Jika di mikrokontroler data belum diterima maka tekan tombol RESET.
8. Jika tombol RESET sudah ditekan dan data belum diterima mikrokontroler maka periksa ulang koneksi jaringan pada *smartphone* dan ulangi penginputan data.
9. Selanjutnya mikrokontroler akan menjalankan perintah untuk menggerakkan robot sesuai data yang didapat dari *smartphone*.
10. Kemudian robot secara otomatis akan bergerak sesuai dengan pola jalan yang telah dibuat dan robot akan berhenti ketika semua titik tanam sudah dikerjakan.
11. *User* dapat memonitor berapa persen nilai *supply* tegangan *input* dan jumlah titik tanam yang sedang dikerjakan.

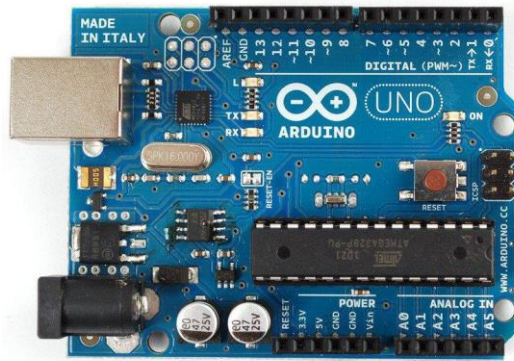
LAMPIRAN 5

Datasheet Komponen Elektrik

Arduino Uno

A. Description

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino. Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.



B. Specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
PWM Digital I/O Pins	6
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Frequency (Clock Speed)	16 Hz
LED_BUILTIN	13

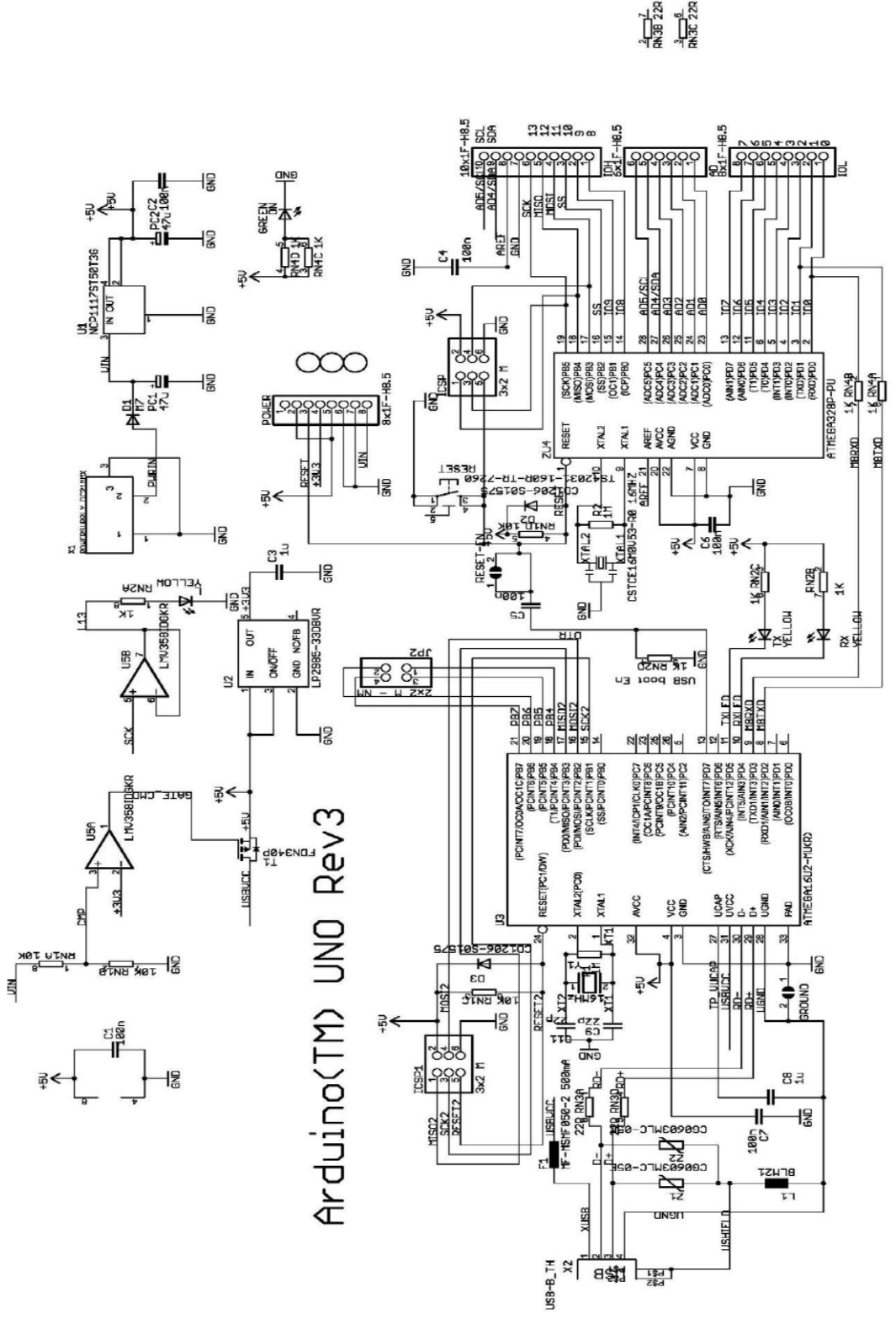
Length	68.6 mm
Width	58.4 mm
Weight	25

C. Pins Configuration

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	<ul style="list-style-type: none"> • Vin: Input voltage to Arduino when using an external power source. • 5V: Regulated power supply used to power microcontroller and other components on the board. • 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. • GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4(SDA), A5(SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input

voltage.

D. Schematic



10kΩ 1% 0603
RN33 22R
RN33 22R

NodeMCU ESP8266-12E

A. Description

NodeMCU is an open source IoT platform. Which includes firmware which runs on the ESP8266 Wi-Fi Module from Espressif Systems, and hardware which is based on the ESP-12 module. The term “NodeMCU” by default refers to the firmware rather than the dev kits. NodeMCU firmware was developed so that AT commands can be replaced with Lua scripting making the life of developers easier. So it would be redundant to use AT commands again in NodeMCU.

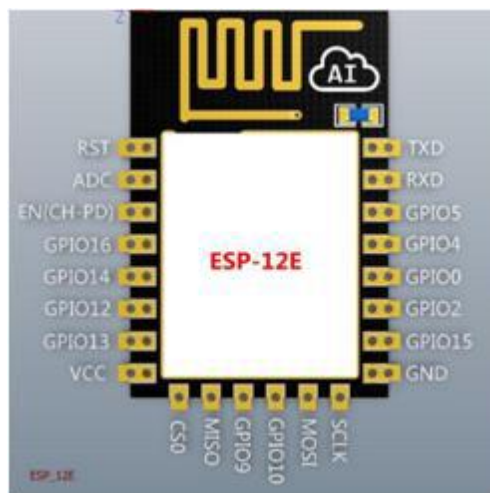


B. Specifications

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	Tx Power	802.11 b: +20 dBm 802.11 g: +17 dBm 802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps) 802.11 g: -75 dbm (54 Mbps) 802.11 n: -72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control

		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C ~ 125°C
	Storage Temperature Range	-40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Software Development	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

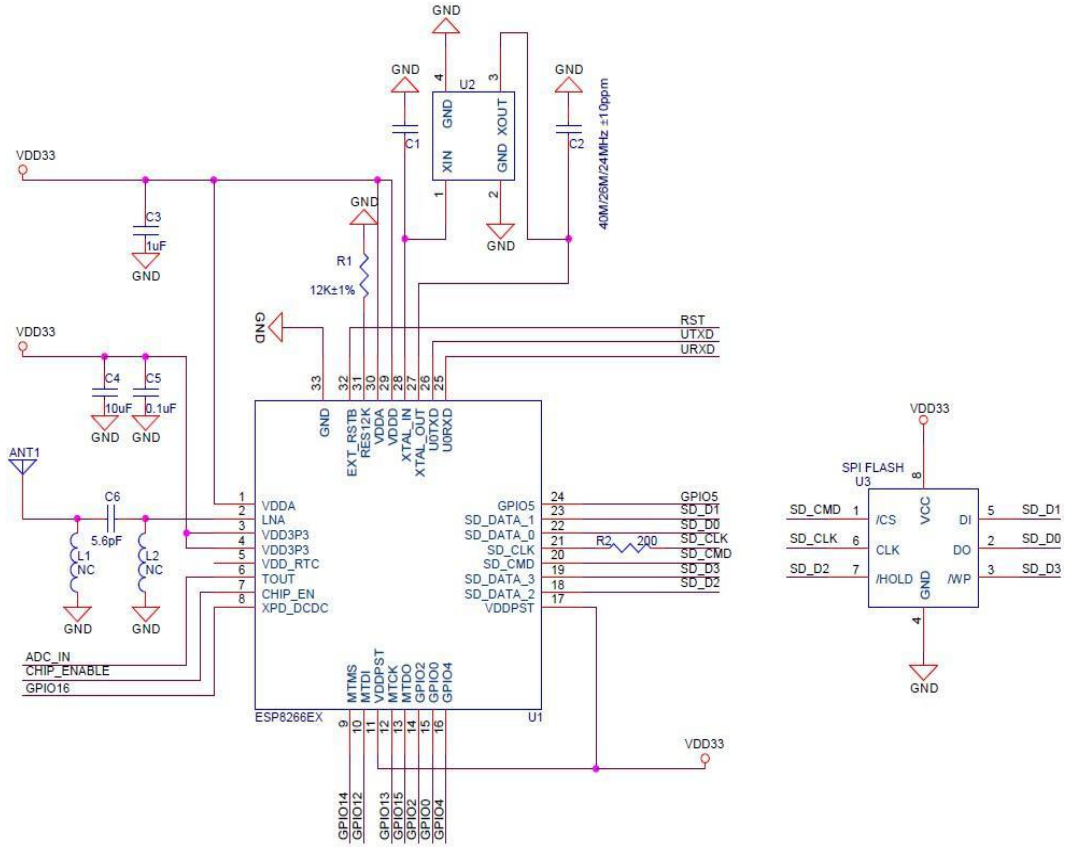
C. Pins Configuration



No	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion result.Input voltage range 0-1v,scope:0-1024
3	EN	Chip enable pin.Active high
4	IO16	GPIO16; can be used to wake up the chipset from deep sleep mode.
5	IO14	GPIO14; HSPI_CLK
6	IO12	GPIO12; HSPI_MISO
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input
11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	Main output slave input
14	SCLK	Clock
15	GND	GND
16	IO15	GPIO15; MTDO; HSPICS; UART0_RTS
17	IO2	GPIO2; UART1_TXD
18	IO0	GPIO0
18	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD; GPIO3
22	TXD	UART0_TXD; GPIO1

PIN Mode			
Mode	GPIO15	GPIO0	GPIO2
UART	Low	Low	High
Flash Boot	Low	High	High

D. Schematic



IC LM78XX

A. Description

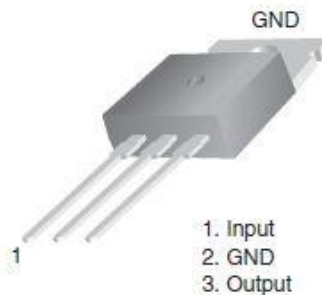
The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



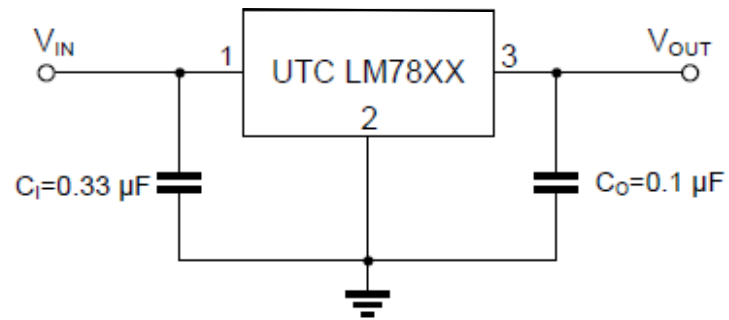
B. Specifications LM7810

Regulator Type	Standard
Output Polarity	Positive
Output Voltage Type	Fixed
Package Type	Other, TO-220, 3 PIN
Life Cycle Stage	Active
Output Voltage	9.5 to 10.5 volts
Vin	12.5 to 25 volts

C. Pins Configuration LM7810



D. Schematic



- Note 1: To specify an output voltage, substitute voltage value for "XX".
- 2: Bypass capacitors are recommended for optimum stability and transient response and should be located as close as possible to the regulators.

MB102 Breadboard 3.3V/5V Power Supply

A. Description

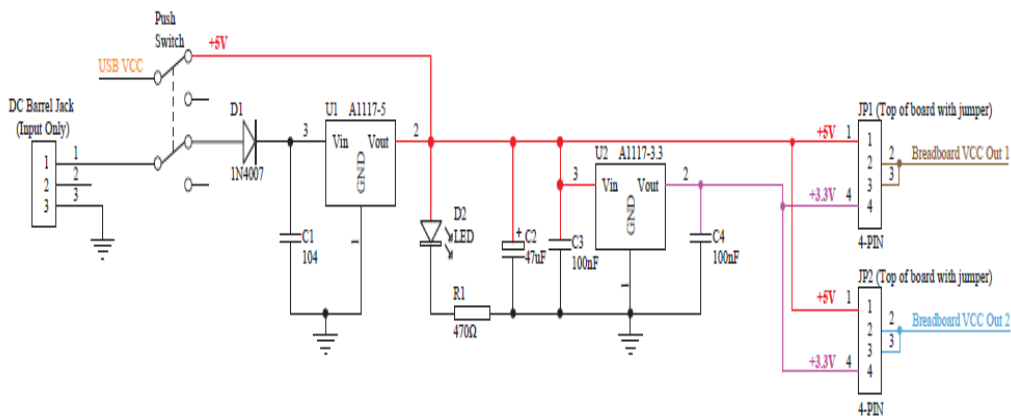
A 3.3V and 5V Breadboard Power Supply Module with series diode, polarity reversal protection. The module can take 6.5V to 12V input and can produce 3.3V and +5V. This is a must have power supply module for experimenters who have to test/prototype electronic circuits on breadboard or perforated/veroboards.



B. Specifications

Input voltage	6.5-12 V (DC) or 5V USB power supply
Output voltage	3.3V and 5V can switch over
Maximum output current	<700 mA
External Input voltage ON/OFF switch	
Independent control of upper and Lower Bread Board Power Rails. Can switch over to 0V, 3.3V, 5V using jumpers on any rail	
USB device connector onboard for power output to external device	
Size	5.3x3.5 cm

C. Schematic



LM2596 DC-DC Adjustable PSU Module

A. Description

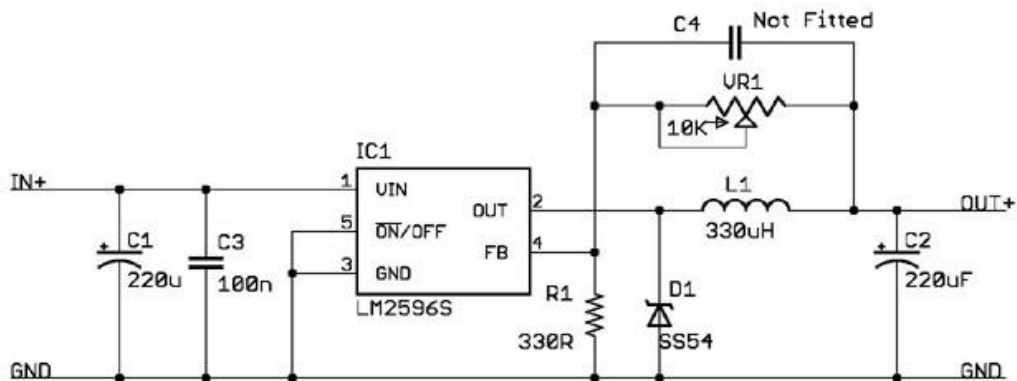
LM2596 DC to DC step down regulator, adjustable +1.23 to 35vdc output, 2A. Ideal for battery operated projects requiring a regulated powersupply.



B. Specifications

Regulator Type:	Step Down (Non Isolated input to Output)
Input Voltage:	+4 to 40vdc
Output Voltage:	+1.23 to 35vdc
Output Current:	2A rated, (3A maximum with heatsink)
Efficiency:	Up to 92% (when output voltage is set high)
Switching Frequency:	150kHz
Dropout Voltage:	2vdc minimum
Protection:	Short circuit current limiting
Load Regulation:	+/- 0.5%
Voltage Regulation:	+/- 2.5%
Temperature:	-40 to +85 deg C (output power less than 10Watts)
Board Size:	43.6mm L x 21mm W x 14mm H

C. Schematic



Voltage Detection Sensor Module 25V

A. Description

The Voltage Detection Sensor Module 25V allows you to use the analog input of a microcontroller to monitor voltages much higher than it capable of sensing. This module is based on the principle of resistive voltage divider design, can make the red terminal connector input voltage to 5 times smaller. Arduino analog input voltages up to 5 v, the voltage detection module input voltage not greater than $5V \times 5 = 25V$ (if using 3.3V systems, input voltage not greater than $3.3V \times 5 = 16.5V$). Arduino AVR chips have 10-bit AD, so this module simulates a resolution of 0.00489V (5V/1023), so the minimum voltage of input voltage detection module is $0.00489V \times 5 = 0.02445V$.



B. Specifications

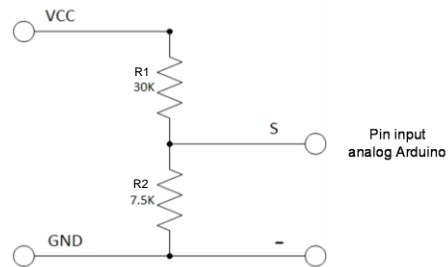
Input Voltage (V)	0 to 25
Voltage Detection Range (V)	0.02445 to 25
Analog Voltage Resolution (V)	0.00489
Length (mm)	28
Width (mm)	14
Height (mm)	13
Weight (gm)	4
Shipment Weight	0.095 kg
Shipment Dimensions	4 x3x2 cm

C. Pins Configuration



VCC	Positive terminal of the voltage to be measured (0-25V)
GND	Negative terminal of the voltage to be measured
S	Analog Input of Arduino
+	VCC of Arduino
-	GND of Arduino

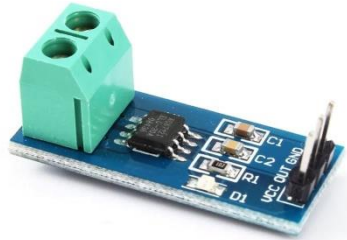
D. Schematic



ACS712 Current Sensor

A. Description

ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection. The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer.



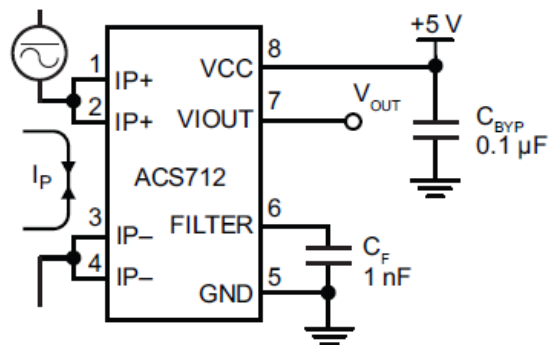
B. Specifications

	5A Module	20A Module	30A Module
Supply Voltage (VCC)	5Vdc Nominal	5Vdc Nominal	5Vdc Nominal
Measurement Range	-5 to +5 Amps	-20 to +20 Amps	-30 to +30 Amps
Voltage at 0A	VCC/2 (nominally 2.5Vdc)	VCC/2 (nominally 2.5Vdc)	VCC/2 (nominally 2.5VDC)
Scale Factor	185 mV per Amp	100 mV per Amp	66 mV per Amp
Chip	ACS712ELC-05A	ACS712ELC-10A	ACS712ELC-30A

C. Pins Configuration

Pin Number	Pin Name	Pin Description
1 & 2	IP+	+ve terminals for sensing current
3 & 4	IP-	-ve terminals for sensing current
5	GND	Signal Ground
6	FILTER	External Capacitor (to set the bandwidth)
7	VIOUT	Analog Output
8	VCC	Power Supply

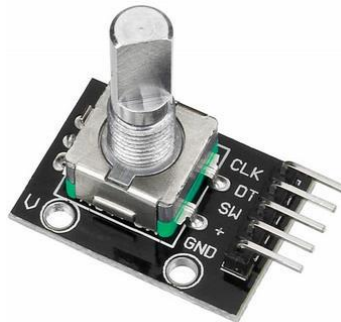
D. Schematic



Rotary Encoder

A. Description

The KY-040 rotary encoder is a rotary input device (as in knob) that provides an indication of how much the knob has been rotated AND what direction it is rotating in. It's a great device for stepper and servo motor control. You could also use it to control devices like digital potentiometers.



B. Specifications

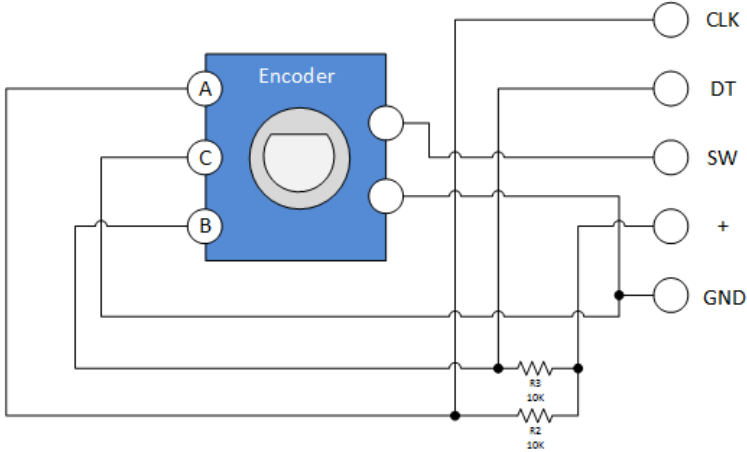
Operating voltage	5V
Pulses/360° Rotation	20
Output	2-bit gray code
Mechanical Angle	360° continuous
Dimensions	(30 x 18 x 30) mm
With built in push button switch (push to operate)	

C. Pins Configuration



CLK	Enable pin A
DT	Enable pin B
SW	No pushbutton switch
+	5 V supply
GND	Ground

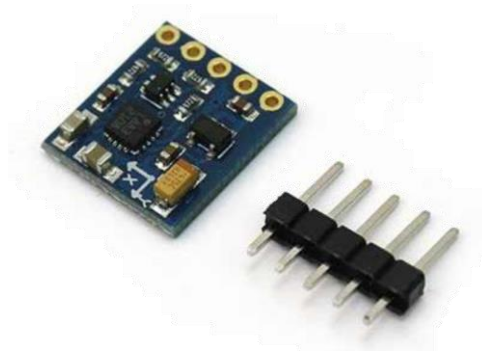
D. Schematic



GY-271 ELECTRONIC COMPASS

A. Description

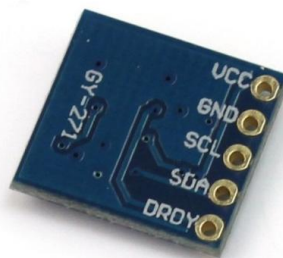
The Compass Module is designed for low-field magnetic sensing with a digital interface and perfect to give precise heading information. This compact sensor fits into small projects such as UAVs and robot navigation systems. The sensor converts any magnetic field to a differential voltage output on 3 axes. This voltage shift is the raw digital output value, which can then be used to calculate headings or sense magnetic fields coming from different directions.



B. Specifications

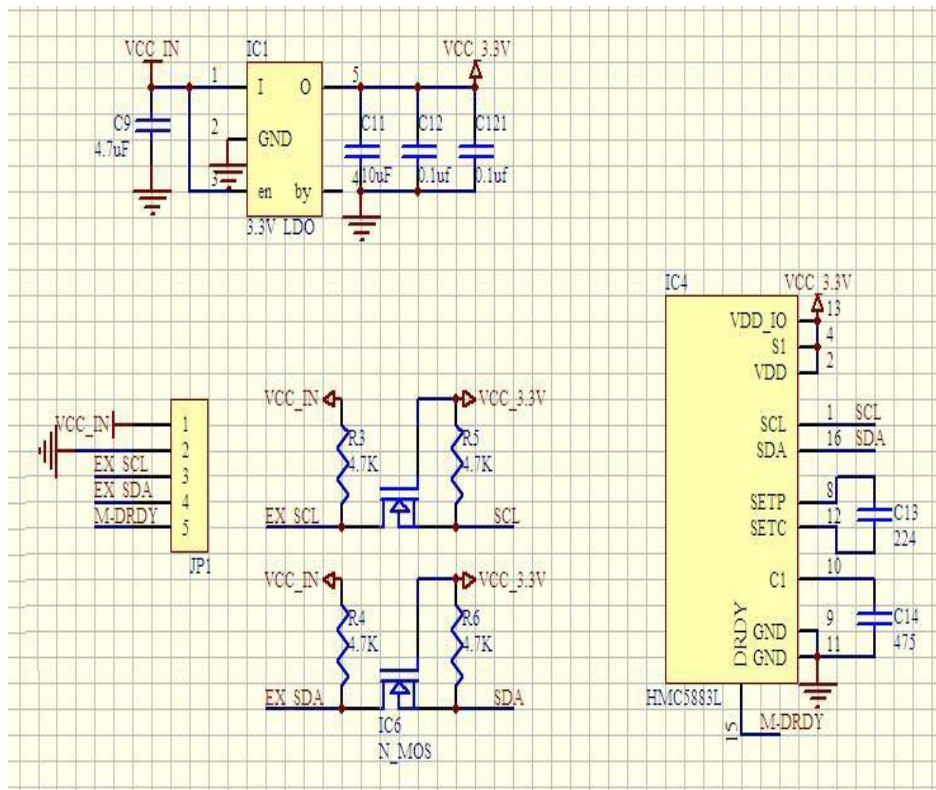
Power	3 – 5 VDC
Chip	HMC5883L
Communication	I2C protocol
Measuring range	± 1.3 -8 Gauss
Dimensions	14.8 x 13.5 x 3.5 mm

C. Pins Configuration



VCC	3V-5V DC
GND	Ground
SCL	Analog input (A5)
SDA	Analog input (A4)
DRDY	Not connected

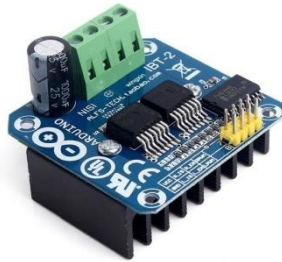
D. Schematic



IBT-2 HIGH CURRENT 30A DC MOTOR DRIVER

A. Description

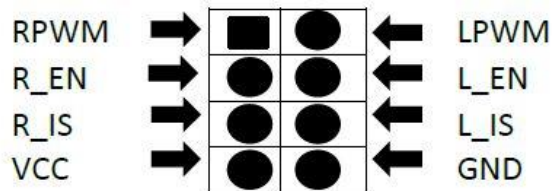
This driver uses two high current half bridge Infineon BTS 7960 chip for motor drive applications. Interfacing to a microcontroller is made easy using this driver which features current sensing, slew rate adjustment and protection against overn temperature, overvoltage, undern voltage, Overn current and short circuit. This small size driver provides a cost optimized solution for protected high current PWM motor drives.



B. Specifications

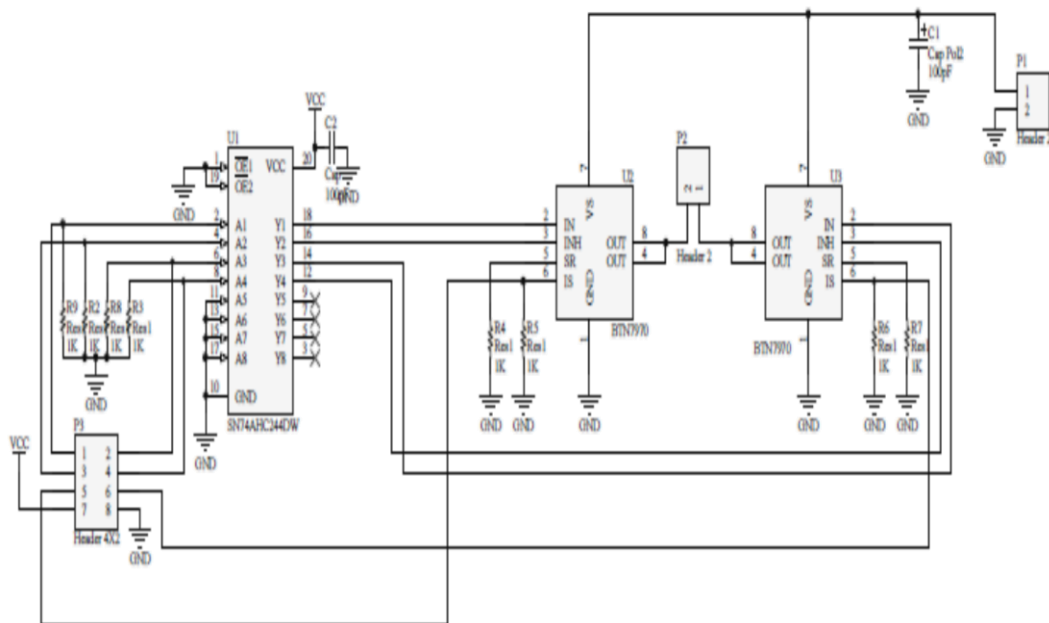
Input voltage	5 – 27 VDC
Maximum Current	43A
Input level	3.3V – 5 VDC
Control motor speed	PWM up to 25 kHz
Current limitation level	30 A Current sense capability
Size	4*5*1.2 cm
Weight	66 gm

C. Pins Configuration



RPWM	Input PWM Forward Level , Active High
LPWM	Input PWM Reverse Level , Active High
R_EN	Input Enable Forward Driver, Active High
L_EN	Input Enable Reverse Driver, Active High
R_IS	Forward Drive, Side current alarm output
L_IS	Reverse Drive, Side current alarm output
VCC	+5 V Power Supply Mikrokontroler
GND	Gnd Power Supply Mikrokontroler
B+, B-	Motor power Input
M+, M-	Motor power Output

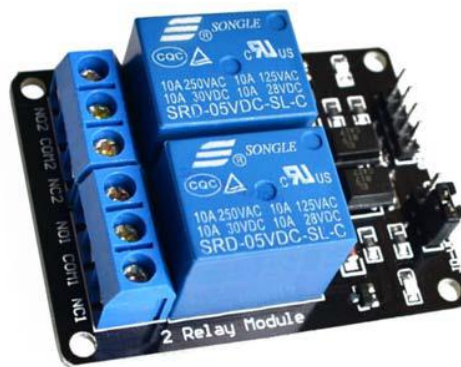
D. Schematic



2 Channel 5V Optical Isolated Relay Module

A. Description

This is a LOW Level 5V 2-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.



B. Specifications

Relay Maximum output	DC 30V/10A, AC 250V/10A.
2 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.	
Standard interface	Microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
That can be controlled directly by Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.	
Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.	
PCB Size	45.8 x 32.4 mm

C. Pins Configuration



VCC	5 VDC
GND	Ground
IN1	High/Low output
IN2	High/Low output
COM	5 VDC

D. Schematic

