



**DESAIN KONTROL MESIN PERONTOK PADI MENGGUNAKAN  
MOTOR *BRUSHLESS DIRECT CURRENT* BERBASIS  
*NEURAL NETWORK***

**SKRIPSI**

Oleh

**Ahmad Sulhan  
NIM 121910201068**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER**

**2016**



**DESAIN KONTROL MESIN PERONTOK PADI MENGGUNAKAN  
MOTOR *BRUSHLESS DIRECT CURRENT* BERBASIS  
*NEURAL NETWORK***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Teknik Elektro (S1)  
dan mencapai gelar Sarjana Teknik

Oleh

**Ahmad Sulhan**  
**NIM 121910201068**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**

## PERSEMBAHAN

Segala puji bagi Allah, Tuhan yang Maha pengasih lagi Maha penyayang, serta sholawat dan salam semoga terlimpahkan kepada makhluk-Mu yang paling mulia, Nabi Muhammad S.A.W. dengan penuh rasa syukur saya persembahkan skripsi ini kepada:

1. Kedua orang tua, Bukarto dan Kiptiyah yang selalu memberi cinta dan kasih sayang yang begitu berarti hingga akhirnya saya dapat memperoleh gelar sarjana teknik (ST).
2. Adik tercinta Syafri dan Riki, keluarga besar ayahanda dan ibunda tercinta yang telah memberikan dukungan, motivasi dan senantiasa menjadi penghibur lara.
3. Bapak dan Ibu Guru saya sejak usia saya masih belia hingga dewasa yang mengajarkan ilmu hingga saya bisa menjadi seperti sekarang ini.
4. Teman-teman SATE UJ 12 yang selama ini selalu mendukung, membantu dan memberikan arahan.
5. Bangsa dan Negara Kesatuan Republik Indonesia serta almamater Fakultas Teknik Universitas Jember.

**MOTTO**

Maka sesungguhnya bersama kesulitan ada kemudahan.  
(Terjemahan Surat Al-Insyirah Ayat 5)<sup>1</sup>

Persahabatan manusia itu lebih mulia dibandingkan dengan manusia itu sendiri  
(Film 3 Idiot)



---

<sup>1</sup> Syarif, Al Jumanatusy. 2003. *Al-Qur'an dan Terjemahannya*. Bandung: CV Penerbit J.ART

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Sulhan

NIM : 121910201068

menyatakan dengan sesungguhnya bahwa karya tulis yang berjudul “Desain Kontrol Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* Berbasis *Neural Network*” adalah benar-benar hasil karya sendiri, kecuali dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 30 September 2016

Yang menyatakan,

Ahmad Sulhan  
NIM. 121910201068

**SKRIPSI**

**DESAIN KONTROL MESIN PERONTOK PADI MENGGUNAKAN  
MOTOR *BRUSHLESS DIRECT CURRENT* BERBASIS  
*NEURAL NETWORK***

Oleh

Ahmad Sulhan  
NIM 121910201058

Pembimbing

Dosen Pembimbing Utama : Prof. Dr. Ir. Bambang Sujanarko M.M.

Dosen Pembimbing Anggota : Dr. Bambang Sri Kaloko, S.T., M.T.

**PENGESAHAN**

Skripsi berjudul “Desain Kontrol Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* dengan Berbasis *Neural Network*” telah diuji dan disahkan pada:

Hari, tanggal : Jumat, 30 September 2016

Tempat : Fakultas Teknik Universitas Jember

Tim Penguji

Pembimbing Utama,

Pembimbing Anggota,

Prof. Dr. Ir. Bambang Sujanarko, M.M.  
NIP 19631201 199402 1 002

Dr. Bambang Sri Kaloko, S.T., M.T.  
NIP 19710402 200312 1 001

Penguji I

Penguji II

Ir. Widyono Hadi, MT.  
NIP 196104141989021001

H.R.B. Moch. Gozali, S.T., M.T.  
NIP 19690608 199903 1 002

Mengesahkan  
Dekan Fakultas Teknik,

Dr. Ir. Entin Hidayah, M.UM.  
NIP 19661215 199503 2 001

## Desain Kontrol Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* Berbasis *Neural Network*

Ahmad Sulhan

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

### ABSTRAK

Pada umumnya mesin perontok padi menggunakan motor bakar sebagai penggerak untuk merontokkan padi. Penelitian sebelumnya dilakukan dengan menggunakan motor BLDC sebagai penggerak perontok padi. Namun kontrol yang diberikan masih secara manual. Motor BLDC sebagai penggerak perontok padi harus memiliki kecepatan yang sesuai agar mendapatkan kualitas padi yang baik. Namun pada penelitian ini memiliki kekurangan yaitu ketika di beri beban maka kecepatan mengalami penurunan. Saat motor BLDC dihidupkan dan tidak ada beban padi, maka motor BLDC masih tetap menyala sehingga daya pada baterai terbuang sia-sia.

Pada penelitian ini dirancang sistem kontrol mesin perontok padi yang dapat mengatur kecepatan motor sesuai dengan beban padi yang diberikan. Prinsip kerja dari kontrol motor ini sebagai pengatur kecepatan motor BLDC ketika diberi beban yang berubah namun kecepatan motor BLDC konstan pada kecepatan 400 Rpm. Hal ini bertujuan agar kecepatan yang konstan pada 400 Rpm saat diberi beban padi yang berubah dan memberikan hasil panen padi yang bagus dan tidak hancur sehingga menghasilkan kualitas perontok padi yang baik. Beban yang diberikan pada penelitian ini adalah 0 Kg, 1 Kg, 2 Kg, 3 Kg dan 4 Kg.

Hasil penelitian ini dapat diketahui bahwa kontrol *neural network* dapat menjaga kecepatan motor BLDC tetap konstan pada kecepatan 400 Rpm. Mesin perontok padi tanpa kontrol diberi beban maka kecepatan motor BLDC mengalami penurunan kecepatan. Saat tanpa beban kecepatan motor BLDC 400 Rpm, dengan beban 1 kg kecepatan motor BLDC 375 Rpm, beban 2 kg kecepatan motor BLDC 369 Rpm, beban 3 kg kecepatan motor BLDC 334 Rpm dan dengan beban 4 kg kecepatan motor BLDC 304 Rpm. Sedangkan saat menggunakan kontrol *neural network* kecepatan motor BLDC konstan pada kecepatan 400 Rpm. Saat beban 0 kg membutuhkan waktu 7 detik untuk mencapai kecepatan konstan, dengan beban 1 kg membutuhkan waktu 7 detik untuk mencapai kecepatan konstan, beban 2 kg membutuhkan waktu 15 detik untuk mencapai kecepatan konstan, beban 3 kg membutuhkan waktu 5 detik untuk mencapai kecepatan konstan dan dengan beban 4 kg membutuhkan waktu 4 detik untuk mencapai kecepatan konstan.

**Kata Kunci:** Mesin Perontok Padi, BLDC dan *Neural Network*



## Design Paddy Thresher Machine Control Using Brushless Direct Current Motor Based on Neural Network

**Ahmad Sulhan**

Department of Electrical Engineering, Faculty of Engineering,  
University of Jember

### **ABSTRACT**

In general threshing machine using the internal combustion engine as the driving force to thresh rice. Previous research carried out by using the BLDC motor as the driving threshers. However, given still seacara control manual. BLDC motor as the driving threshers must have the appropriate speed in order to get a good grain quality. But in this study have the disadvantage that when a given load, the speed decreased. When the BLDC motor is turned on and no-load of rice, the BLDC motor is still on so that the battery power is wasted.

In this study designed threshing machine control system that can regulate motor speed according to the load of rice given. The working principle of this motor control as the BLDC motor speed control when given load change but BLDC motor speed constant at a speed of 400 rpm. It is intended that a constant speed at 400 rpm when given a load of rice that is changing and the rice harvest were nice and not destroyed so as to produce good quality rice thresher. Load given to this study is 0 kg, 1 kg, 2 kg, 3 kg and 4 kg.

The results of this study can be seen that control neural network can keep BLDC motor speed remains constant at a speed of 400 rpm. Threshing machine without control by the burden of the BLDC motor speed decrease speed. When no load BLDC motor speed of 400 rpm, with a load of 1 kg BLDC motor speed of 375 rpm, a load of 2 kg BLDC motor speed 369 rpm, 3 kg load speed of 334 rpm and a BLDC motor with a load of 4 kg BLDC motor speed 304 rpm. Meanwhile, when using neural network control BLDC motor constant speed at a speed of 400 rpm. When a load of 0 kg takes 7 seconds to reach constant speed, with a load of 1 kg takes 7 seconds to reach constant speed, load 2 kg takes 15 seconds to reach constant speed, the load 3 kg takes 5 seconds to reach constant speed and with a load of 4 kg takes four seconds to reach a constant speed.

**Keywords:** Rice Thresher machine, BLDC and Neural Network

## RINGKASAN

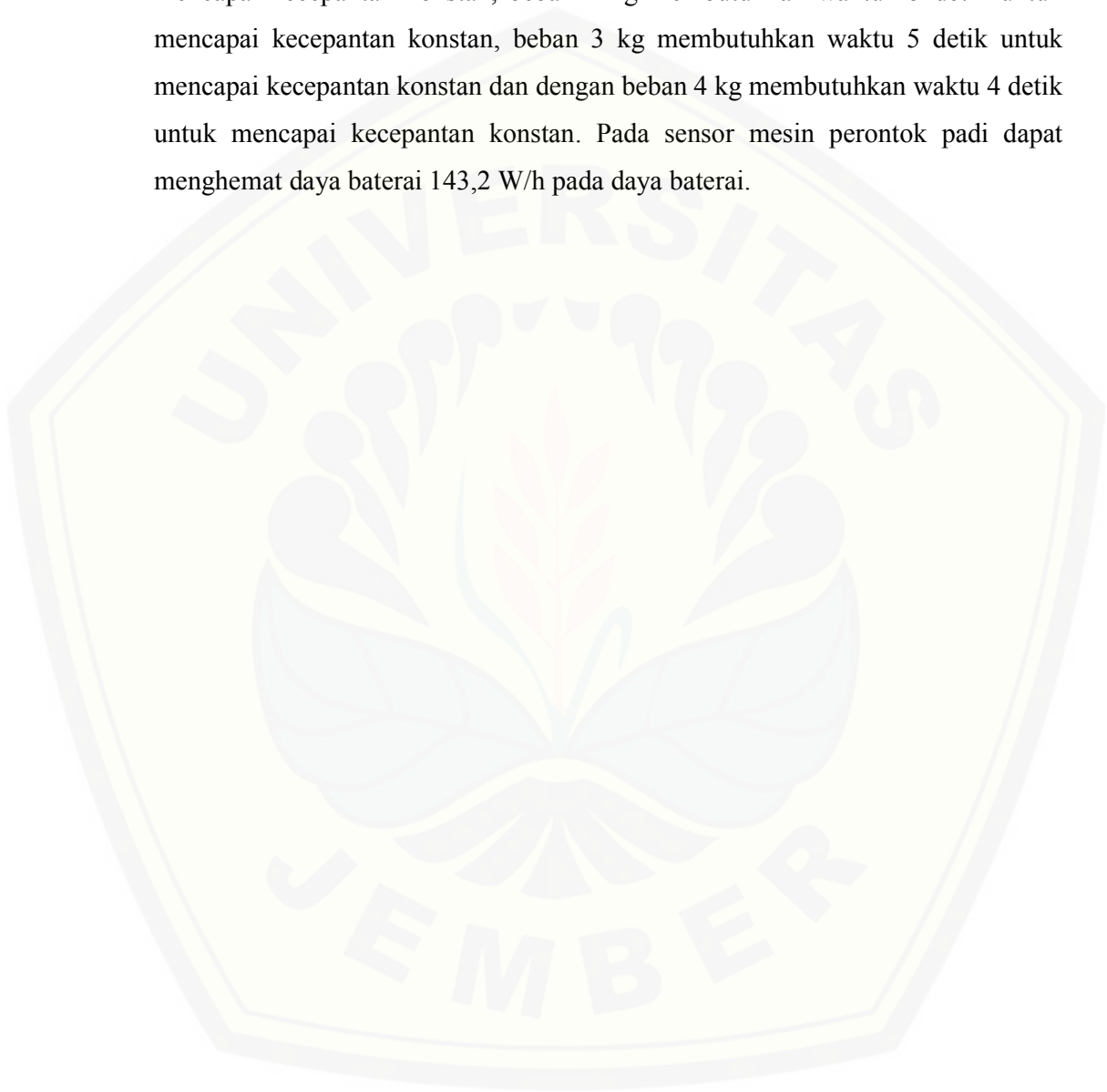
**Desain Kontrol Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* Berbasis *Neural Network* ; Ahmad Sulhan, 121910201068; 75 Lembar; Jurusan Teknik Elektro; Fakultas Teknik, Universitas Jember.**

Pada umumnya mesin perontok padi menggunakan motor bakar sebagai penggerak untuk merontokkan padi. Penelitian sebelumnya dilakukan dengan menggunakan motor BLDC sebagai penggerak perontok padi. Namun kontrol yang diberikan masih secara manual. Motor BLDC sebagai penggerak perontok padi harus memiliki kecepatan yang sesuai agar mendapatkan kualitas padi yang baik. Namun pada penelitian ini memiliki kekurangan yaitu ketika di beri beban maka kecepatan mengalami penurunan. Saat motor BLDC dihidupkan dan tidak ada beban padi, maka motor BLDC masih tetap menyala sehingga daya pada baterai terbuang sia-sia sehingga dibutuhkan sensor sebagai pendeteksi keberadaan padi sebagai *switch* pada motor sehingga lebih efisien dan hemat daya baterai.

Pada penelitian ini dirancang sistem kontrol mesin perontok padi yang dapat mengatur kecepatan motor sesuai dengan beban padi yang diberikan. Prinsip kerja dari kontrol motor ini sebagai pengatur kecepatan motor BLDC ketika diberi beban yang berubah namun kecepatan motor BLDC konstan pada kecepatan 400 Rpm. Hal ini bertujuan agar kecepatan yang konstan pada 400 Rpm saat diberi beban padi yang berubah dan memberikan hasil panen padi yang bagus dan tidak hancur sehingga menghasilkan kualitas perontok padi yang baik. Beban yang diberikan pada penelitian ini adalah 0 Kg, 1 Kg, 2 Kg, 3 Kg dan 4 Kg.

Hasil penelitian ini dapat diketahui bahwa kontrol *neural network* dapat menjaga kecepatan motor BLDC tetap konstan pada kecepatan 400 Rpm. Mesin perontok padi tanpa kontrol diberi beban maka kecepatan motor BLDC mengalami penurunan kecepatan. Saat tanpa beban kecepatan motor BLDC 400 Rpm, dengan beban 1 kg kecepatan motor BLDC 375 Rpm, beban 2 kg kecepatan motor BLDC 369 Rpm, beban 3 kg kecepatan motor BLDC 334 Rpm

dan dengan beban 4 kg kecepatan motor BLDC 304 Rpm. Sedangkan saat menggunakan kontrol *neural network* kecepatan motor BLDC konstan pada kecepatan 400 Rpm. Saat beban 0 kg membutuhkan waktu 7 detik untuk mencapai kecepatan konstan, dengan beban 1 kg membutuhkan waktu 7 detik untuk mencapai kecepatan konstan, beban 2 kg membutuhkan waktu 15 detik untuk mencapai kecepatan konstan, beban 3 kg membutuhkan waktu 5 detik untuk mencapai kecepatan konstan dan dengan beban 4 kg membutuhkan waktu 4 detik untuk mencapai kecepatan konstan. Pada sensor mesin perontok padi dapat menghemat daya baterai 143,2 W/h pada daya baterai.



## PRAKATA

Puji syukur kami panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga dapat terselesaikannya skripsi dengan judul “Desain Kontrol Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* dengan Berbasis *Neural Network*”, sebagai salah satu persyaratan akademis dalam rangka untuk melakukan penelitian dan menyelesaikan Program Studi Teknik Elektro (S1) di Fakultas Teknik Universitas Jember.

Dalam skripsi ini dijabarkan tentang desain kontrol mesin perontok padi menggunakan motor *brushless direct current* dengan berbasis *neural network* sebagai sistem kontrol mesin perontok padi yang dapat mengatur kecepatan motor sesuai dengan beban padi yang diberikan yang dapat menghasilkan sistem kendali motor BLDC yang handal untuk diaplikasikan pada mesin perontok padi yang memiliki putaran dengan kecepatan yang konstan sehingga dapat meningkatkan gabah yang berkualitas.

Pada kesempatan ini kami menyampaikan terima kasih dan penghargaan yang setinggi-tingginya penulis sampaikan kepada Prof. Dr. Ir. Bambang Sujanarko, M.M dan Dr. Bambang Sri Kaloko, S.T., M.T selaku dosen pembimbing yang telah memberikan petunjuk, koreksi serta saran hingga terwujudnya skripsi ini.

Terima kasih dan penghargaan penulis sampaikan kepada yang terhormat:

1. Dr. Ir. Entin Hidayah, M.UM, selaku Dekan Fakultas Teknik Universitas Jember
2. Dr. Bambang Sri Kaloko, S.T., M.T, selaku Ketua Jurusan Teknik Elektro.
3. Deddy Kurnia Setiawan, S.T., M.T, Ketua Program Studi S1 Teknik Elektro.
4. Mohammad Agung Prawira Negara, S.T, M.T, selaku Komisi Bimbingan S1 Teknik Elektro.
5. Kedua orang tua saya Bukarto dan Kiptiyah, terimakasih atas doa dan dukungan materil serta pengorbanan yang tidak ada batasnya.
6. Adik tercinta Syafrial Ukti Maulana dan Ahmad Maftuh Rifki Maulana yang selalu jadi penghibur dan penyemangat.

7. Teman-teman seangkatan SATE UJ 12 khususnya Cris Avian, Ardiansyah, Dony Kurniaan T, Mahfud, Yuda Herdiansyah dan Surya Ahmadi yang telah memberikan dukungan dan semangat serta membantu dalam proses pengerjaan skripsi.
8. Teman sejawat kontrakan Riau 60 Fabian Deni M, Febriyanto, M Riski E.F, Moh Nur Yuski, Akhmad Kusyairi, Dian Agus P, Ahmad Junaidi dan Yudan Aries Maulana yang selalu menghibur dan memberikan semangat.
9. Kelompok Kuliah Kerja Nyata 29 Universitas Jember Makyik, Neni, Rini, Dini dan Mas nico yang selalu memberikan motivasi dan dukungan semangat.
10. Adinda Fatimatus Zainiyah yang telah mendoakan yang terbaik dan senantiasa memberiku semangat.
11. Seluruh pihak yang telah membantu memberikan kontribusi dalam penyelesaian skripsi ini

Skripsi ini telah disusun dengan optimal, namun tidak menutup kemungkinan adanya kekurangan. Oleh karena itu, penulis dengan tangan terbuka menerima masukan yang membangun. Semoga tulisan ini berguna bagi semua pihak yang memanfaatkannya.

Jember, 30 September 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL .....	ii
HALAMAN PERSEMBAHAN .....	iii
HALAMAN MOTTO .....	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBINGAN.....	vi
HALAMAN PENGESAHAN.....	vii
ABSTRAK .....	viii
ABSTRACT .....	ix
RINGKASAN .....	x
PRAKATA.....	xii
DAFTAR ISI.....	xiv
DAFTAR TABEL .....	xvii
DAFTAR GAMBAR.....	xviii
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang.....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>3</b>
<b>1.3 Batasan Masalah .....</b>	<b>3</b>
<b>1.4 Tujuan.....</b>	<b>4</b>
<b>1.5 Manfaat.....</b>	<b>4</b>
<b>1.6 Sistematika Pembahasan.....</b>	<b>4</b>
<b>2.1 Mesin Perontok Padi .....</b>	<b>5</b>
<b>2.2 Motor Brushless Direct Current (BLDC).....</b>	<b>6</b>
2.2.1 Gambaran Umum Motor BLDC .....	6
2.2.2 Model Sistem Motor BLDC.....	8
2.2.3 Prinsip Kerja Motor BLDC.....	8
<b>2.3 Inverter Tiga Fasa .....</b>	<b>10</b>

2.4	Metode Six Step Commutation .....	12
2.5	Sensor Hall.....	13
2.6	Jaringan Syaraf Tiruan ( <i>Artificial Neural Network</i> ).....	17
2.7	Algoritma <i>Levenberg-Marquardt</i> .....	23
2.8	Mesin Perontok Padi dengan motor BLDC .....	25
2.8.1	Arduino Uno R3 .....	26
2.9	<i>Photodiode</i> .....	27
<b>BAB 3. METODOLOGI PENELITIAN.....</b>		<b>29</b>
3.1	Diagram Blok Sistem .....	29
3.2	Parameter <i>Levenber Marquadt Neural Network</i> .....	30
3.3	Diagram Alir Penelitian .....	33
<b>BAB 4 HASIL DAN PEMBAHASAN.....</b>		<b>35</b>
4.1	Pengujian Sensor Kecepatan .....	36
4.2	Data Awal Motor BLDC .....	37
4.2.1	Pengujian Mesin Perontok Padi Tanpa Kontrol pada Tanpa Beban .....	37
4.2.2	Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 1 Kg.....	39
4.2.3	Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 2 Kg.....	41
4.2.4	Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 3 Kg.....	42
4.2.5	Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 4 Kg.....	44
4.3	Kontrol <i>Neural Network</i> .....	45
4.4	Data Motor BLDC Setelah Menggunakan Kontrol <i>Neural Network</i> .....	54
4.4.1	Pengujian Mesin Perontok Padi dengan Kontrol Tanpa Beban .....	54
4.4.2	Pengujian Mesin Perontok Padi dengan Kontrol pada Beban 1 Kg.....	56

4.4.3 Pengujian Mesin Perontok Padi dengan Kontrol pada Beban 2 Kg.....	57
4.4.4 Pengujian Mesin Perontok Padi dengan Kontrol pada Beban 3 Kg.....	59
4.4.5 Pengujian Mesin Perontok Padi dengan Kontrol pada Beban 4 Kg.....	61
<b>4.5 Data Perbandingan Sebelum dan Sesudah Menggunakan Kontrol <i>Neural Network</i> .....</b>	<b>62</b>
4.5.1 Pengujian Mesin Perontok Padi sebelum dan sesudah menggunakan Kontrol Tanpa Beban.....	62
4.5.2 Pengujian Mesin Perontok Padi sebelum dan sesudah menggunakan Kontrol Pada Beban 1 Kg.....	64
4.5.3 Pengujian Mesin Perontok Padi sebelum dan sesudah menggunakan Kontrol Pada Beban 2 Kg.....	66
4.5.4 Pengujian Mesin Perontok Padi sebelum dan sesudah menggunakan Kontrol Pada Beban 3 Kg.....	68
4.5.5 Pengujian Mesin Perontok Padi sebelum dan sesudah menggunakan Kontrol Pada Beban 4 Kg.....	70
<b>4.6 Efisiensi baterai yang di hemat oleh mesin perontok padi .....</b>	<b>72</b>
<b>BAB 5 PENUTUP.....</b>	<b>73</b>
<b>5.2 Kesimpulan.....</b>	<b>73</b>
<b>5.3 Saran .....</b>	<b>73</b>
<b>DAFTAR PUSTAKA.....</b>	<b>75</b>
<b>LAMPIRAN.....</b>	<b>77</b>



DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel logika kebenaran sistem switching pada inverter BLDC.....	15
Tabel 2. 2 Penyederhanaan menggunakan Peta <i>Karnaugh</i> .....	17
Tabel 2. 3 Spesifikasi Arduino Uno R3 .....	27
Tabel 4. 1 Hasil Pembacaan Sensor Kecepatan dan <i>Tachometer</i> .....	37
Tabel 4. 2 Pengujian Kontrol Motor BLDC Tanpa Kontrol dengan Beban 0 Kg	38
Tabel 4.3 Pengujian Kontrol Motor BLDC Tanpa Kontrol dengan Beban 1 Kg .	40
Tabel 4. 4 Pengujian Kontrol Motor BLDC Tanpa Kontrol dengan Beban 2 Kg	41
Tabel 4. 5 Pengujian Kontrol Motor BLDC Tanpa Kontrol dengan Beban 3 Kg	43
Tabel 4. 6 Pengujian Kontrol Motor BLDC Tanpa Kontrol dengan Beban 4 Kg	44
Tabel 4. 7 Parameter <i>Weight</i> dan bias pada <i>Hidden Layer 1</i> .....	51
Tabel 4. 8 Parameter <i>Weight</i> dan Bias pada <i>Hidden Layer 2</i> .....	52
Tabel 4. 9 Pengujian Kontrol Motor BLDC dengan Kontrol Tanpa Beban .....	55
Tabel 4. 10 Pengujian Kontrol Motor BLDC dengan kontrol pada Beban 1 Kg..	56
Tabel 4. 12 Pengujian Kontrol Motor BLDC dengan kontrol pada Beban 2 Kg..	58
Tabel 4. 13 Pengujian Kontrol Motor BLDC dengan Kontrol pada Beban 3 Kg.	60
Tabel 4. 14 Pengujian Kontrol Motor BLDC dengan kontrol pada Beban 4 Kg..	61
Tabel 4. 15 Perbedaan Tanpa Kontrol dan dengan Menggunakan Kontrol tanpa Beban .....	63
Tabel 4. 16 Perbedaan Tanpa Kontrol dan dengan Menggunakan Kontrol pada Beban .....	65
Tabel 4. 17 Perbedaan Tanpa Kontrol dan dengan Menggunakan Kontrol pada Beban 2 kg .....	67
Tabel 4. 18 Perbedaan tanpa Kontrol dan dengan Menggunakan Kontrol pada Beban 3 Kg .....	69
Tabel 4. 19 Perbedaan tanpa Kontrol dan dengan Menggunakan Kontrol pada Beban 4 kg .....	71

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Mesin perontok padi yang konvensional.....	6
Gambar 2. 2 (a) Rotor (magnet permanen), (b) Stator.....	7
Gambar 2. 3 Rangkaian ekivalen dari motor BLDC.....	8
Gambar 2.4 Tegangan stator BLDC.....	9
Gambar 2. 5 Inverter 3 fasa.....	11
Gambar 2. 6 Mode konduksi 120° .....	12
Gambar 2. 7 Algoritma PWM six-step .....	12
Gambar 2.8 Gelombang trapezoid 3 Fasa .....	13
Gambar 2. 9 Algoritma PWM Six-Step 3 Fasa.....	13
Gambar 2.10 Penempatan sensor hall pada stator.....	14
Gambar 2. 11 Urutan switch Q1-Q6 .....	16
Gambar 2.12 Struktur jaringan syaraf Tiruan .....	19
Gambar 2. 13 (a) Modul kontroler arduino UNO-R3 (b) Pinout arduino UNO-R3 .....	26
Gambar 3. 1 Diagram Blok Sistem .....	29
Gambar 3. 2 Grafik Parameter <i>Regression Bacpropagation</i> .....	31
Gambar 3. 3 Diagram Blok Parameter <i>Hidden Layer</i> Pada <i>Neural Network</i> .....	31
Gambar 3. 4 Blok Sistem <i>Weight</i> dan <i>Bias</i> pada <i>Layer 1</i> .....	32
Gambar 3. 5 Blok Sistem <i>Weight</i> dan <i>Bias</i> Pada <i>Layer 2</i> .....	32
Gambar 3. 6 Tampilan Program Arduino Ketika <i>Weight</i> dan <i>Bias</i> di Masukkan. 32	
Gambar 3. 7 <i>Flowchart</i> Penelitian .....	33
Gambar 4. 1 Mesin Perontok Padi Menggunakan Motor BLDC Dengan Kontrol <i>Neural Network</i> .....	35
Gambar 4. 2 Grafik Kecepatan Motor BLDC Tanpa Kontrol dengan Beban 0 Kg .....	39
Gambar 4. 3 Grafik Kecepatan Motor BLDC Tanpa Kontrol dengan Beban 1 Kg .....	40

Gambar 4. 4 Grafik Kecepatan Motor BLDC Tanpa Kontrol dengan Beban 2 Kg .....	42
Gambar 4. 5 Grafik Kecepatan Motor BLDC Tanpa Kontrol dengan Beban 3 Kg .....	43
Gambar 4. 6 Grafik Kecepatan Motor BLDC Tanpa Kontrol dengan Beban 4 Kg .....	45
Gambar 4. 7 Hasil Training Neural Network Levenberg Marquadth.....	47
Gambar 4. 8 Grafik Performance Backpropagation.....	47
Gambar 4. 9 Grafik Training State Backpropagation .....	49
Gambar 4. 10 Grafik Regression Backpropagation .....	49
Gambar 4. 11 Struktur Jaringan yang Ada pada Weight Hidden Layer 1 .....	50
Gambar 4. 12 Struktur Jaringan yang Ada di Neural Network.....	51
Gambar 4. 13 Grafik Kecepatan Motor BLDC dengan Kontrol tanpa Beban.....	56
Gambar 4. 14 Grafik Kecepatan Motor BLDC dengan Kontrol pada Beban 1 Kg .....	57
Gambar 4. 15 Grafik Kecepatan Motor BLDC dengan Kontrol pada Beban 2 Kg .....	59
Gambar 4. 16 Grafik Kecepatan Motor BLDC dengan Kontrol pada Beban 3 Kg .....	60
Gambar 4. 17 Grafik Kecepatan Motor BLDC dengan Kontrol pada Beban 4 Kg .....	62
Gambar 4. 18 Grafik Kecepatan Motor BLDC tanpa kontrol dan dengan Kontrol Tanpa Beban .....	64
Gambar 4. 19 Grafik Kecepatan Motor BLDC tanpa kontrol dan dengan Kontrol Pada Beban 1 Kg.....	66
Gambar 4. 20 Grafik Kecepatan Motor BLDC tanpa kontrol dan dengan Kontrol Pada Beban 2 Kg.....	68
Gambar 4. 21 Grafik Kecepatan Motor BLDC tanpa kontrol dan dengan Kontrol Pada Beban 3 Kg.....	70
Gambar 4. 22 Grafik Kecepatan Motor BLDC tanpa kontrol dan dengan Menggunakan Kontrol pada Beban 4 Kg.....	72

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Sejak mulainya diterapkan teknologi kimia-biologi pada tahun 1960-an dibidang budidaya, padi Indonesia telah terjadi pergeseran lingkungan karena para petani sebagian besar menanam varietas padi unggul yang berumur pendek. Oleh karena itu, penanganan pasca panen padi menjadi masalah yang serius, mengingat panen gabah padi yang banyak dan waktu panen yang singkat. Dalam penanganan pasca panen padi, salah satu permasalahan yang dihadapi adalah masih kurangnya kesadaran dan pemahaman petani terhadap penanganan pasca panen yang baik sehingga mengakibatkan masih tingginya kehilangan hasil dan rendahnya mutu gabah (Ariani, 2013).

Penanganan masalah pasca panen merupakan hal yang serius oleh masyarakat petani karena dapat berdampak pada kualitas gabah yang di hasilkan oleh mesin perontok yang digunakan untuk merontokkan padi. Banyaknya bulir padi yang ikut terbang dengan malai disebabkan karena mesin perontok yang digunakan kurang efisien, sehingga di butuhkan mesin perontok yang cukup baik untuk menangani masalah penyusutan hasil gabah yang dirontok. Penggunaan mesin perontok selain untuk mempercepat proses perontokan juga dapat menekan susut hasil (Rachmat *et al.* 1993). Perontokan gabah dapat berlangsung lebih sempurna sehingga hampir semua gabah terlepas dari malainya dan kurang dari 1% bulir gabah yang terbang bersama jerami.

Pada mesin perontok padi dibutuhkan motor penggerak yang dapat memutar baling-baling dari perontok padi tersebut. Namun, motor yang digunakan harus mempunyai torsi yang besar dan efisiensi yang tinggi. Motor DC memiliki efisiensi yang tinggi dibandingkan dengan motor AC karena penggunaan tegangan DC pada rotor untuk menggerakkan motor tersebut. Namun, kerugian dari motor DC ini yaitu biaya perawatan yang tinggi karena adanya sikat yang ada pada motor DC. *Brush* pada motor DC ini cepat mengalami kerusakan. Hal ini terjadi

karena pada saat motor berputar, pada *brush* akan timbul *arching* akibat proses komutasi. Untuk memenuhi kebutuhan akan efisiensi, kecepatan, torsi yang tinggi dan dapat divariasikan serta biaya perawatan yang rendah, maka digunakan motor *brushless* DC motor. Biaya perawatan yang tinggi inilah yang menyebabkan kurang populernya motor DC dikalangan masyarakat. Oleh karena itu, ditemukannya *brushless DC motor* (BLDC) atau kita lebih mengenalnya motor DC tanpa sikat. Popularitas motor BLDC meningkat dari hari ke hari. Pada kenyataannya motor memiliki kepadatan energi yang tinggi, biaya pembuatan bahan magnet permanen seperti Samarium Cobalt (Sm-Co) dan Nd-Fe-B yang makin murah dan adanya kemajuan dalam inovasi desain. Motor BLDC paling sesuai untuk sistem yang membutuhkan pengontrolan posisi pada industri sampai ukuran menengah, karena sangat dinamis dan kerugian yang kecil serta rasio torsi/berat yang baik (Sujanarko, 2012).

Kelebihan dari motor BLDC dengan upaya untuk meningkatkan efisiensi maka dilakukan pengontrolan dengan menggunakan kecerdasan buatan (*artificial intelligence*). Salah satunya kecerdasan buatan yang akan saya gunakan adalah dengan menggunakan jaringan syaraf tiruan (JST) atau NN (*neural network*). *Neural network* adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir pada jaringan tersebut. Pemodelan dari *neural network* (NN) adalah pemodelan data statistik yang non-linier dengan memodelkan hubungan antara input dan output untuk menemukan pola-pola pada data.

Kekurangan dari penggunaan motor bakar pada mesin perontok yaitu polusi udara dan suara yang dihasilkan oleh motor bakar yang digunakan sebagai penggerak pada perontok padi. Suara mesin terpapar oleh suara bising terutama bagi petani yang mengoperasikan mesin-mesin yang menggunakan motor diesel dengan tingkat kebisingan mencapai 105 desibel/ dB(A) (Sucipta & Suriasih, 2004). Tingkat kebisingan sekian melebihi nilai ambang batas untuk kebisingan di tempat kerja yaitu 85 dB(A) untuk pekerja yang terpapar bising selama 8 jam sehari terus menerus (Surat Edaran Menteri Tenaga Kerja, Transmigrasi dan

Koperasi No. SE.01/MEN/1078). Pada penelitian sebelumnya sudah ada mesin perontok padi yang menggunakan motor konvensional (Ajav & B.A, 2005) dan mesin perontok padi yang menggunakan motor BLDC dengan menggunakan Kontrol otomatis (Sujanarko, Kuswardhani & Septiandrini, 2016). Berangkat dari permasalahan ini, penulis ingin membuat pengontrolan motor *Brushless DC* berbasis *neural network* pada mesin perontok untuk mengatur kecepatan konstan walaupun dengan penambahan beban.

## 1.2 Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah diantaranya:

- a. Bagaimana desain kontrol kecepatan motor BLDC yang di aplikasikan pada mesin perontok padi dengan NN (*neural network*) menggunakan arduino UNO R3?
- b. Bagaimana cara mengatur kinerja motor BLDC dengan mengatur kecepatan yang konstan saat beban berubah?

## 1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, maka batasan masalah pada skripsi ini diantaranya:

- a. Motor BLDC yang digunakan berdaya 500 Watt.
- b. Mikrokontroler yang digunakan arduino UNO R3.
- c. Tidak membahas tentang konstruksi dari mesin perontok padi.
- d. Tidak membahas secara mendetail tentang konstruksi motor BLDC.
- e. Kecepatan motor BLDC 400 Rpm.
- f. Rugi – rugi motor dan mesin perontok padi diabaikan.
- g. Beban yang diberikan yaitu 0 Kg, 1 Kg, 2 Kg, 3 Kg dan 4 Kg.

## 1.4 Tujuan

Penelitian yang diusulkan dalam skripsi ini memiliki beberapa tujuan diantaranya:

- a. Mendesain kontrol kecepatan motor BLDC yang di aplikasikan pada mesin perontok padi dengan NN (*neural network*) menggunakan arduino UNO R3.
- b. Mendapatkan sistem kontrol kecepatan yang konstan saat tanpa beban maupun berbeban.

## 1.5 Manfaat

Manfaat yang diharapkan setelah tujuan penelitian ini tercapai adalah dapat menghasilkan sistem kendali motor BLDC yang handal untuk diaplikasikan pada mesin perontok padi yang memiliki putaran dengan kecepatan yang konstan jika diberi beban gabah yang berubah

## 1.6 Sistematika Pembahasan

Secara garis besar penyusunan skripsi ini adalah sebagai berikut:

### BAB 1. PENDAHULUAN

Latar belakang, tujuan pembahasan, rumusan masalah, batasan masalah, dan sistematika pembahasan.

### BAB 2. TINJAUAN PUSTAKA

Berisi tentang tinjauan pustaka yang menguraikan pendapat-pendapat atau hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan, landasan teori merupakan penjabaran dari tinjauan pustaka.

### BAB 3. METODOLOGI PENELITIAN

Menjelaskan tentang metode kajian yang digunakan untuk menyelesaikan skripsi.

### BAB 4. HASIL DAN PEMBAHASAN

Berisi hasil pengujian dan analisa hasil pengujian.

### BAB 5. PENUTUP

Berisi tentang kesimpulan dan saran dari penulis.

## BAB 2. KAJIAN PUSTAKA

### 2.1 Mesin Perontok Padi

Mesin perontok padi adalah sistem manusia-mesin yang dilaksanakan dengan menggunakan bantuan peralatan kerja berupa *thresher*. Mesin ini digunakan untuk membantu pekerjaan petani dalam merontokkan padi untuk memperoleh gabah. Perontokan merupakan bagian integral dari proses penanganan pasca panen padi, karena padi yang telah layak dipanen dirontokkan untuk memisahkan bulir-bulir padi jeraminya. Prinsip kerja mesin perontok padi adalah dengan memukul bagian tangkai padi (jerami) sehingga bulir-bulir terlepas. Besarnya daya threshers (mesin perontok benih padi) yang diperlukan dalam proses perontokan padi dipengaruhi oleh ukuran, bentuk dan stuktur jaringan pada bulir-bulir yang akan dirontokkan. Efisiensi gabah yang terontok oleh mesin perontok padi dapat di hitung dengan menggunakan rumus berikut:

$$n_p = \frac{B_{GR}}{B_{GR} + B_{GTR}} \times 100\% \dots\dots\dots (1)$$

Keterangan:

$n_p$  = Efisiensi perontokan padi (%)

$B_{GR}$  = Bobot gabah terontok (kg)

$B_{GTR}$  = Bobot gabah tak terontok (kg)

Dengan adanya efisiensi dari perontokan padi ini kita bisa melihat dari gabah yang terontok dan tak terontok dengan menggunakan mesin perontok padi tersebut sehingga kita bisa memilih mesin perontok padi dengan efisiensi perontokan padi yang paling besar. Tapi dengan menggunakan mesin perontok padi yang pada umumnya menggunakan motor bakar. Mahalnya harga mesin *thresher* yang beredar di pasaran saat ini dipengaruhi oleh mahalanya harga enjin penggerak. Masih ada peluang untuk membuat desain dan merekayasa mesin perontok padi yaitu prototipe dengan kemampuan kinerja diantara pedal *thresher* dengan kapasitas kerja 120 kg/jam dan power *thresher* kapasitas kerja 600 kg/jam (Sulistiadji, Rosmeika & Gunanto, 2008)





Gambar 2.1 Mesin perontok padi yang konvensional  
(Sujanarko, Kuswardhani & Septiandrini, 2016)

Pada mesin perontok padi konvensional yang menggunakan motor bakar umumnya mahal biaya perawatan dan bahan bakar yang digunakan untuk operasional mesin perontok padi. Penggunaan motor bakar sebagai penggerak pada mesin perontok padi memiliki efisiensi yang rendah, polusi udara dan suara yang dihasilkan gaduh (Sujanarko, Kuswardhani & Septiandrini, 2016)

## 2.2 Motor Brushless Direct Current (BLDC)

### 2.2.1 Gambaran Umum Motor BLDC

Motor BLDC merupakan motor sinkron. Hal ini berarti bahwa medan magnet yang dihasilkan stator dan medan magnet yang dihasilkan rotor berputar pada frekuensi yang sama (Asif Ali *et al*, Tanpa Tahun). Putaran pada rotor disebabkan oleh medan magnet pada stator yang pada setiap saatnya hanya aktif dua fasa (hanya dua fasa yang ter-suply pada setiap saat sementara satu fasa lainnya tak tersuply).

Beberapa keuntungan *Brushless DC* motor dibandingkan dengan motor DC biasa, antara lain sebagai berikut.

- a. Lebih tahan lama, karena tidak memerlukan perawatan terhadap sikatnya.
- b. Memiliki tingkat efisiensi yang tinggi.
- c. Torsi awal dan kecepatan yang tinggi.

Secara umum motor BLDC terdiri dari dua bagian, yakni: *rotor* merupakan bagian yang bergerak yang terbuat dari permanen magnet dan *stator* merupakan bagian yang tidak bergerak yang terbuat dari kumparan 3 fasa. Wujud dari stator dan rotor BLDC dapat dilihat pada gambar 2.2 berikut.



(a)

(b)

Gambar 2. 2 (a) Rotor (magnet permanen), (b) Stator  
(<http://www.eskonwebsolutions.com>)

Dengan tidak adanya lilitan pada rotor maka tidak mengalami gaya sentrifugal dan karena gulungan lilitan ditutupi oleh sebuah rumah, sehingga dapat didinginkan oleh konduksi dan tidak memerlukan aliran udara di dalam motor untuk pendinginan. Hal ini berarti bahwa internal motor dapat sepenuhnya tertutup dan dilindungi dari kotoran atau benda asing lainnya. Walaupun sumber tegangan yang diperlukan untuk memutar motor adalah AC 3 fasa, motor ini tetap disebut dengan BLDC karena pada implementasinya BLDC menggunakan sumber DC sebagai sumber energi utama yang kemudian diubah menjadi tegangan AC dengan menggunakan inverter 3 fasa. Tujuan dari pemberian tegangan AC 3 fasa pada stator BLDC adalah menciptakan medan magnet putar stator untuk menarik magnet rotor (Kenjo, 2010).

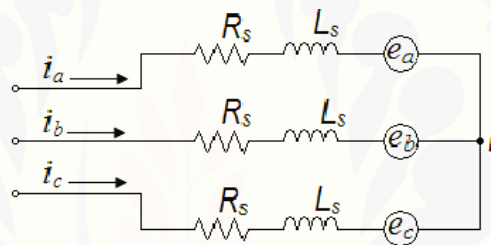
### 2.2.2 Model Sistem Motor BLDC

Motor BLDC menghasilkan sebuah *back-EMF* berupa *trapezoidal*. Resistansi stator pada semua lilitan diasumsikan sama, sedangkan induktansi diri dan induktansi mutual adalah konstan. Sehingga rumus tegangan tiga fasa dapat ditulis sebagai berikut.

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_s - M & 0 & 0 \\ 0 & L_s - M & 0 \\ 0 & 0 & L_s - M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (2)$$

$R_s$  adalah resistansi fasa,  $L_s$  adalah induktansi diri dan  $M$  adalah induktansi mutual. Variabel  $V_a$ ,  $V_b$  dan  $V_c$  adalah tegangan fasa lalu  $i_a$ ,  $i_b$ ,  $i_c$  adalah arus fasa, sedangkan  $e_a$ ,  $e_b$  dan  $e_c$  adalah tegangan fasa *back-EMF*.

Pada gambar 2.3 berikut merupakan rangkaian ekivalen dari motor BLDC.



Gambar 2. 3 Rangkaian ekivalen dari motor BLDC

Dalam kaitannya dengan interaksi arus pada lilitan stator dan medan magnet pada magnet rotor, torsi elektromagnetik dihasilkan di dalam motor BLDC. Torsi elektromagnetik secara matematis dapat ditulis sebagai berikut.

$$T_e = (e_a i_a + e_b i_b + e_c i_c) / \omega_m \quad (3)$$

(Salah, 2011)

### 2.2.3 Prinsip Kerja Motor BLDC

Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 fasa. Akibat arus yang melewati kumparan pada stator timbul medan magnet ( $B$ ) yang secara matematis dapat ditulis sebagai berikut.

$$B = \frac{\mu N I}{2i} \quad (4)$$

Dimana  $N$  merupakan jumlah lilitan,  $i$  merupakan arus,  $l$  merupakan panjang lilitan dan  $\mu$  merupakan permeabilitas bahan.

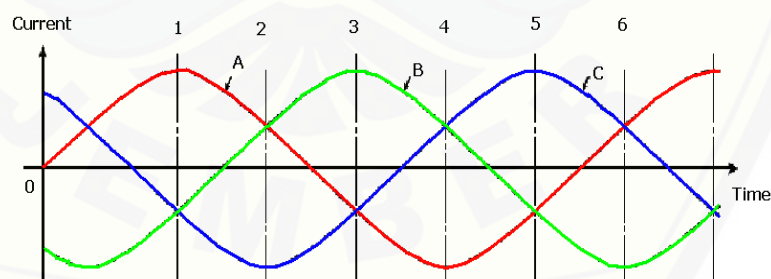
Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 fasa. Karena arus yang diberikan berupa arus AC 3 fasa, nilai medan magnet dan polarisasi setiap kumparan akan berubah-ubah setiap saat. Akibat yang ditimbulkan dari adanya perubahan polarisasi tersebut dan besar medan magnet tiap kumparan adalah terjadinya medan putar magnet dengan kecepatan  $N_s$ :

$$N_s = \frac{120f}{p} \quad (5)$$

Dimana  $f$  merupakan frekuensi tegangan input dinyatakan dalam Hz per satuan detik,  $p$  merupakan jumlah kutub (*pole*) pada rotor dan 120 didapat dalam 1 putaran ( $360^\circ$ ) per 3 fasa motor (Christian, 2013).

Motor BLDC berputar sebagai akibat dari interaksi magnet permanen pada rotor dengan medan magnet yang dihasilkan ketika tegangan DC terhubung pada satu kumparan stator. Untuk menjaga rotasi, orientasi dari medan magnet di stator harus diputar secara berurutan. Hal ini dilakukan dengan menghubungkan tegangan DC pada kumparan stator berikutnya agar rotor berputar (NEC Electronic Corporation, 2006).

Pada gambar 2.4 berikut ditunjukkan bentuk sinyal pada tegangan stator motor BLDC yang berupa sinyal tiga fasa.



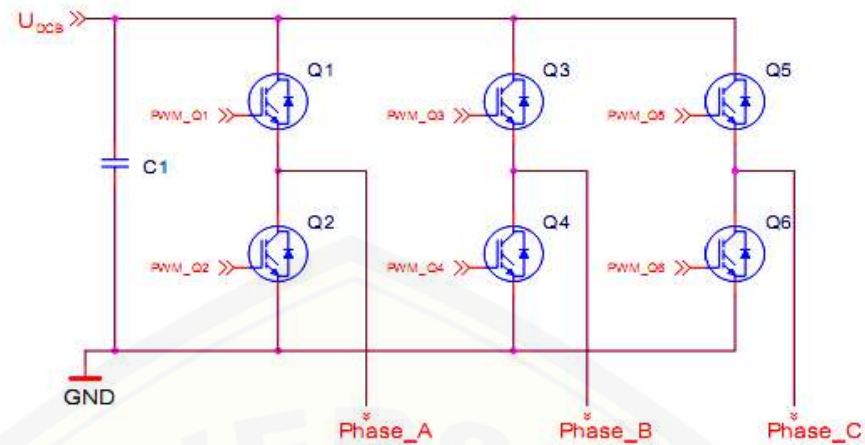
Gambar 2.4 Tegangan stator BLDC

Ketika stator A diberikan tegangan negative maka akan timbul medan magnet dengan polaritas negatif sedangkan B dan C yang diberikan tegangan

positif akan memiliki polaritas positif. Akibat adanya perbedaan polaritas antara medan magnet kumparan stator dan magnet rotor, sisi positif magnet rotor akan berputar mendekati medan magnet stator A, sedangkan sisi negatifnya akan berputar mengikuti medan magnet stator B dan C. Akibat tegangan yang digunakan berupa tegangan AC sinusoidal, medan magnet stator A, B, dan C akan berubah-ubah polaritasnya dan besarnya mengikuti perubahan tegangan sinusoidal AC. Ketika A dan B memiliki medan magnet negatif akibat mendapatkan tegangan negative dan C memiliki medan magnet positif akibat tegangan positif, magnet permanen rotor akan berputar menuju ke polaritas yang bersesuaian yakni bagian negative akan berputar menuju medan magnet stator C dan sebaliknya bagian positif akan berputar menuju medan magnet stator A dan B. Selanjutnya ketika B memiliki medan magnet negative dan A serta C memiliki medan magnet positif, bagian positif magnet permanen akan berputar menuju B dan bagian negative akan menuju A dari kumparan C. Karena tegangan AC sinusoidal yang digunakan berlangsung secara kontinu, proses perubahan polaritas tegangan pada stator ini akan terjadi secara terus menerus sehingga menciptakan medan putar magnet stator dan magnet permanen rotor akan berputar mengikuti medan putar magnet stator ini. Hal inilah yang menyebabkan rotor pada BLDC dapat berputar (Dharmawan, 2009).

### 2.3 Inverter Tiga Fasa

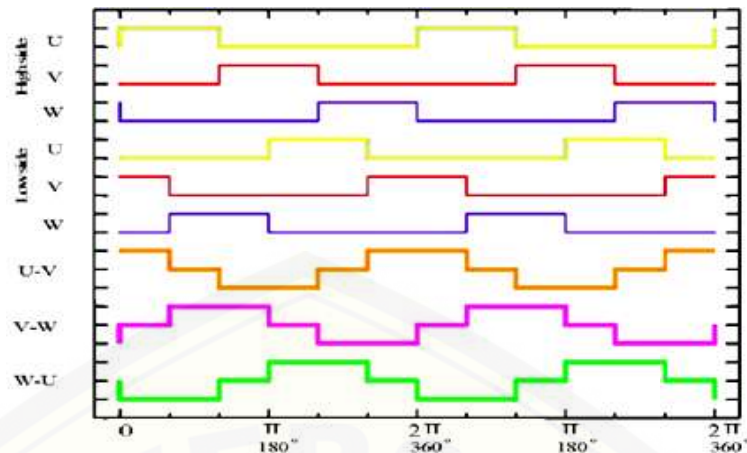
Salah satu fitur kunci dari motor DC brushless adalah adanya inverter yang digunakan untuk menggantikan sikat dan komutator pada motor DC konvensional sehingga bebas perawatan. Definisi secara umum dari inverter adalah peralatan elektronika daya yang berfungsi mengubah tegangan searah (DC) menjadi tegangan bolak-balik (AC). Untuk BLDC diperlukan inverter 3 fasa yang terdiri dari 6 komponen switching. Susunan dari keenam komponen *switching* inverter 3 fasa ditunjukkan pada gambar 2.5 berikut.



Gambar 2. 5 Inverter 3 fasa  
(freescala, 2005)

Terdapat 2 jenis mode operasi dari inverter jenis ini, yaitu mode konduksi  $120^\circ$  dan mode konduksi  $180^\circ$ . Agar dapat berputar, kumparan stator motor BLDC yang aktif tiap stepnya hanya dua buah dan satu buah kumparan lainnya off atau tidak teraliri listrik. Hal ini menyebabkan tiap stepnya hanya diperbolehkan dua buah *switch* saja yang aktif secara bergantian sehingga mode konduksi  $120^\circ$  yang digunakan. Inverter 3 fasa dengan mode konduksi  $120^\circ$  memungkinkan setiap komponen pensakelaran akan konduksi selama  $120^\circ$  dengan pasangan konduksi yang berbeda, seperti pada gambar 2.5,  $60^\circ$  pertama antara  $U-\bar{V}$ , dan  $60^\circ$  ke dua antara  $U-\bar{W}$ . Kemudian  $V-\bar{W}$ ,  $V-\bar{U}$ ,  $W-\bar{U}$ ,  $W-\bar{V}$ ,  $U-\bar{V}$  dan siklus berulang kembali.

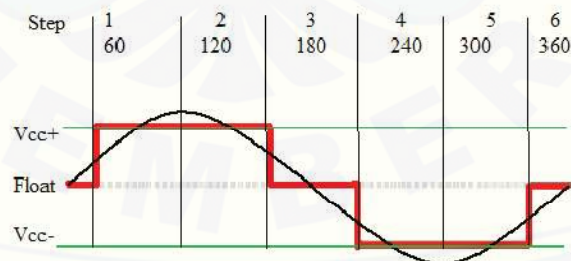
Pada gambar 2.6 berikut merupakan bentuk sinyal untuk tiap switching dan tegangan antar fase yang dihasilkan pada mode konduksi  $120^\circ$ .



Gambar 2. 6 Mode konduksi 120°

#### 2.4 Metode Six Step Commutation

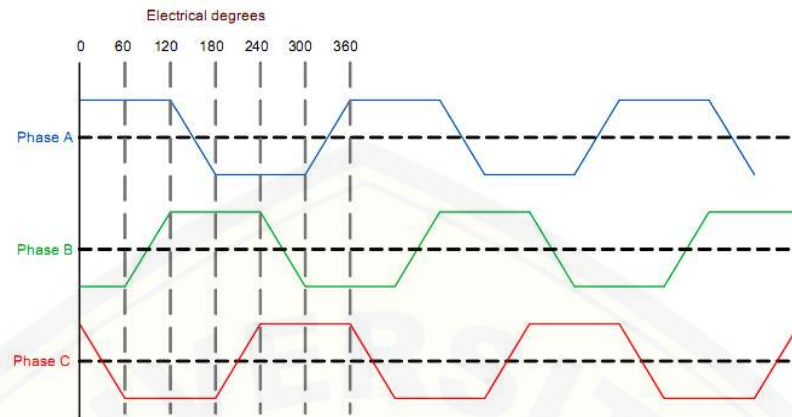
Metode *six-step* merupakan metode yang paling sering digunakan dalam pengendalian BLDC. Hal ini terjadi karena metode ini sederhana sehingga mudah diimplementasikan. Metode ini disebut metode *six-step* karena mampu menciptakan gelombang *trapezoid* atau *square* yang menyerupai gelombang sinusoidal, digunakan PWM *square* yang terdiri dari 6 bagian yakni 2 bagian positif, 2 bagian negatif, dan 2 bagian *floating*. Masing-masing bagian besarnya 60° gelombang sinusoidal. Kondisi *floating* pada algoritma ini adalah kondisi ketika gelombang sinusoidal berpotongan pada titik 0. Pada gambar 2.7 berikut ditampilkan suatu algoritma dari PWM *six step*.



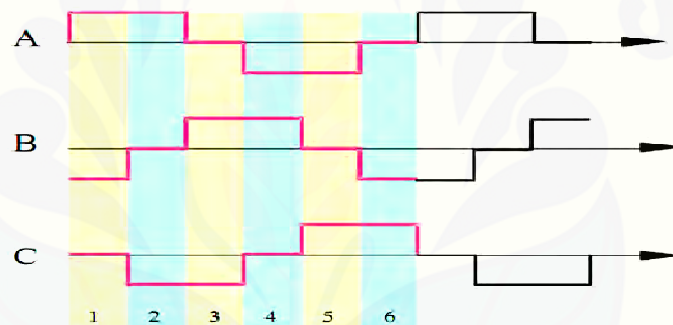
Gambar 2. 7 Algoritma PWM six-step

Untuk membentuk gelombang *trapezoid* atau gelombang *square* 3 fasa seperti yang ditunjukkan pada gambar 2.8, digunakan 3 buah algoritma *six-step* yang

masing-masing berbeda satu step (60 derajat) antara satu algoritma dengan algoritma lainnya seperti yang ditunjukkan pada gambar 2.9.



Gambar 2.8 Gelombang trapezoid 3 Fasa (Microsemi, tanpa tahun)



Gambar 2.9 Algoritma PWM Six-Step 3 Fasa (Shao, 2003)

## 2.5 Sensor Hall

Oleh karena tidak adanya brush pada motor BLDC, untuk menentukan *timing* komutasi yang tepat pada motor ini sehingga didapatkan torsi dan kecepatan yang konstan, diperlukan 3 buah sensor *Hall* dan atau *encoder*. Pada sensor *Hall*, *timing* komutasi ditentukan dengan cara mendeteksi medan magnet rotor dengan menggunakan 3 buah sensor hall untuk mendapatkan 6 kombinasi *timing* yang berbeda, sedangkan pada *encoder*, *timing* komutasi ditentukan dengan cara menghitung jumlah pola yang ada pada *encoder* (Dharmawan, 2009).



Sensor *Hall* dirancang untuk merasakan adanya objek magnetis dengan perubahan posisinya. Perubahan medan magnet yang terus menerus menyebabkan timbulnya pulsa yang kemudian dapat ditentukan frekuensinya, sensor jenis ini biasa digunakan sebagai pengukur kecepatan (Varsani Anis, 2003). Dalam penggunaannya pada motor BLDC, kapanpun kutub magnetik rotor (*North* atau *South*) mendekati sensor *hall*, maka akan menghasilkan sinyal *HIGH* atau *LOW*, yang mana dapat digunakan untuk mendeteksi posisi rotor.

Posisi dari letak penempatan ketiga sensor *hall* pada stator ditunjukkan pada gambar 2.10 berikut.



Gambar 2.10 Penempatan sensor hall pada stator  
(<http://electricisart.blogspot.com>)

Dengan menggunakan tiga sensor *hall*, sebuah kode 3 bit dapat diperoleh dengan nilai antara 1 sampai 6. Masing-masing nilai kode menunjukkan sebuah bagian yang mana posisi rotor sekarang berada. Dari masing-masing nilai kode ini memberikan kita informasi mengenai lilitan mana yang perlu dialiri listrik. Sehingga kita dapat dengan mudah merancang program untuk menentukan dua lilitan mana yang akan dialiri listrik sehingga mengakibatkan rotor berputar (D'Souza, 2004).

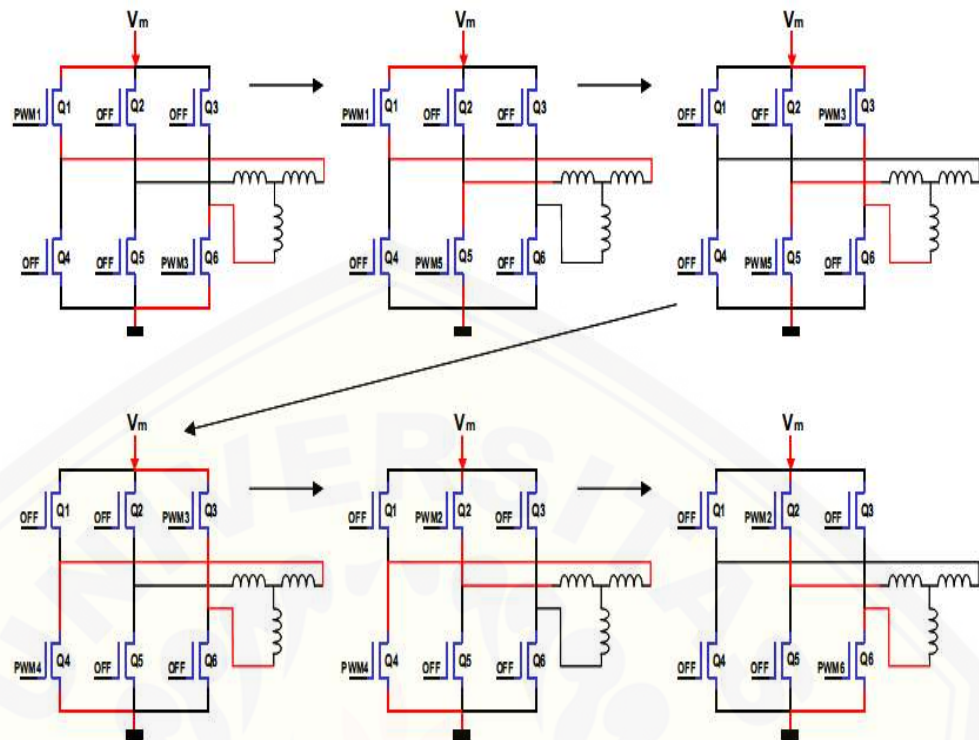
Pada tabel 2.1 berikut ini adalah daftar tabel logika kebenaran pada sistem *switching driver* BLDC yang ditambahkan input logika *direction* untuk menentukan arah putar motor. Pada saat *hall sensor* mendeteksi medan magnet maka di analogikan 1 sedangkan apabila *hall sensor* tidak mendeteksi adanya

medan magnet maka dianalogikan 0. Pada saat arah maju (*CW Direction*) maka di analogikan 1 sedangkan untuk arah mundur (*CCW Direction*) dianalogikan menjadi 0. Q1-Q6 adalah masing-masing *switch* pada bagian inverter BLDC.

Tabel 2. 1 Tabel logika kebenaran sistem switching pada inverter BLDC  
(Bambang Sujanarko, 2012)

Direction	Hall A	Hall B	Hall C	Q1	Q2	Q3	Q4	Q5	Q6
CW	1	1	0	1	0	0	0	1	0
	1	0	0	1	0	0	0	1	0
	1	0	1	1	0	0	1	0	0
	1	0	1	0	0	1	1	0	0
	1	1	1	0	0	1	1	0	1
	1	1	0	0	1	0	0	0	1
CCW	0	1	0	0	0	1	1	0	0
	0	1	1	0	0	1	0	1	0
	0	0	1	0	1	0	0	1	0
	0	0	1	1	1	0	0	0	1
	0	0	0	1	0	1	0	0	1
	0	1	0	1	0	1	0	1	0

Pada gambar 2.11 di bawah ini merupakan penampakan urutan nyala *switch inverter* 3 fasa dari nilai kode dari sensor *hall* dan *direction* CW yang menghasilkan 6 komutasi berbeda untuk membangkitkan medan magnet putar pada stator motor BLDC berdasarkan tabel 2.1 diatas.



Gambar 2. 11 Urutan switch Q1-Q6  
(NXP Semiconductor, 2007)

Untuk memudahkan dalam pemrograman, tabel 2.1 diatas dapat di sederhanakan menggunakan metode *Peta Karnaugh*. *Peta Karnaugh* adalah suatu peralatan grafis yang digunakan untuk mengkonversikan sebuah tabel kebenaran menjadi sebuah rangkaian logika yaitu dengan cara mengubah tabel kebenaran ke dalam *Peta Karnaugh* dan menggambarinya dalam jaringan AND-OR.

Pada tabel 2.2 berikut ditunjukkan cara penyederhanaan menggunakan metode *Peta Karnaugh*.

Tabel 2. 2 Penyederhanaan menggunakan Peta *Karnaugh*

<b>Q</b>	0	0	1	1
<b>1</b>	0	1	1	0
00	0	0	0	1
01	0	0	0	1
11	0	1	0	0
10	0	1	0	0

<b>Q</b>	0	0	1	1
<b>4</b>	0	1	1	0
00	0	1	0	0
01	0	1	0	0
11	0	0	0	1
10	0	0	0	1

<b>Q</b>	0	0	1	1
<b>2</b>	0	1	1	0
00	0	0	0	0
01	1	1	0	0
11	0	0	0	0
10	0	0	1	1

<b>Q</b>	0	0	1	1
<b>5</b>	0	1	1	0
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

<b>Q</b>	0	0	1	1
<b>3</b>	0	1	1	0
00	0	1	1	0
01	0	0	0	0
11	1	0	0	1
10	0	0	0	0

<b>Q</b>	0	0	1	1
<b>6</b>	0	1	1	0
00	0	0	0	0
01	1	0	0	1
11	0	0	0	0
10	0	1	1	0

Berdasarkan Tabel 2.2 diatas, maka didapatkan enam persamaan hasil penyederhanaan dengan *Peta Karnaugh* yaitu:

$$\begin{aligned}
 Q1 &= D' B A' + D B' A & Q4 &= D' B' A + D B A' \\
 Q2 &= D' C B' + D C' B & Q5 &= D' C' B + D C B' \\
 Q3 &= D' C' A + D C A' & Q6 &= D' C A' + D C' A
 \end{aligned}
 \tag{6}$$

(Bambang Sujanarko, 2012)

### 2.6 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan Syaraf Tiruan adalah sistem pemroses informasi dengan karakteristik dan performa yang mendekati syaraf biologis. Jaringan Syaraf Tiruan adalah generalisasi dari pemodelan syaraf biologi dengan asumsi asumsi antara lain:

- a. Pemrosesan informasi terletak pada sejumlah komponen yang dinamakan neuron.

- b. Sinyal merambat antara satu neuron ke neuron-neuron lainnya melalui jalur penghubung.
- c. Tiap jalur penghubung memiliki bobot dan mengalikan besar nilai sinyal yang masuk (jenis *neuron* tertentu).
- d. Tiap neuron menerapkan fungsi aktivasi (biasanya nonlinier) yang menjumlahkan semua masukan untuk menentukan sinyal keluarannya.

Selain memproses, jaringan syaraf tiruan juga memiliki kemampuan menyimpan informasi seperti definisi oleh (Widodo, 2013) bahwa jaringan syaraf adalah pemroses sederhana yang berjumlah banyak dan bekerja secara paralel dan terdistribusi. Jaringan ini memiliki kemampuan menyimpan pengetahuan dan memberikan saat dibutuhkan yang terdiri dari pengetahuan yang dimiliki sebagai hasil proses pembelajaran dan koneksi antar neuron yang berfungsi menyimpan pengetahuan itu. Oleh karena itu, jaringan syaraf tiruan bermaksud membuat sistem yang menyerupai syaraf tiruan biologis.

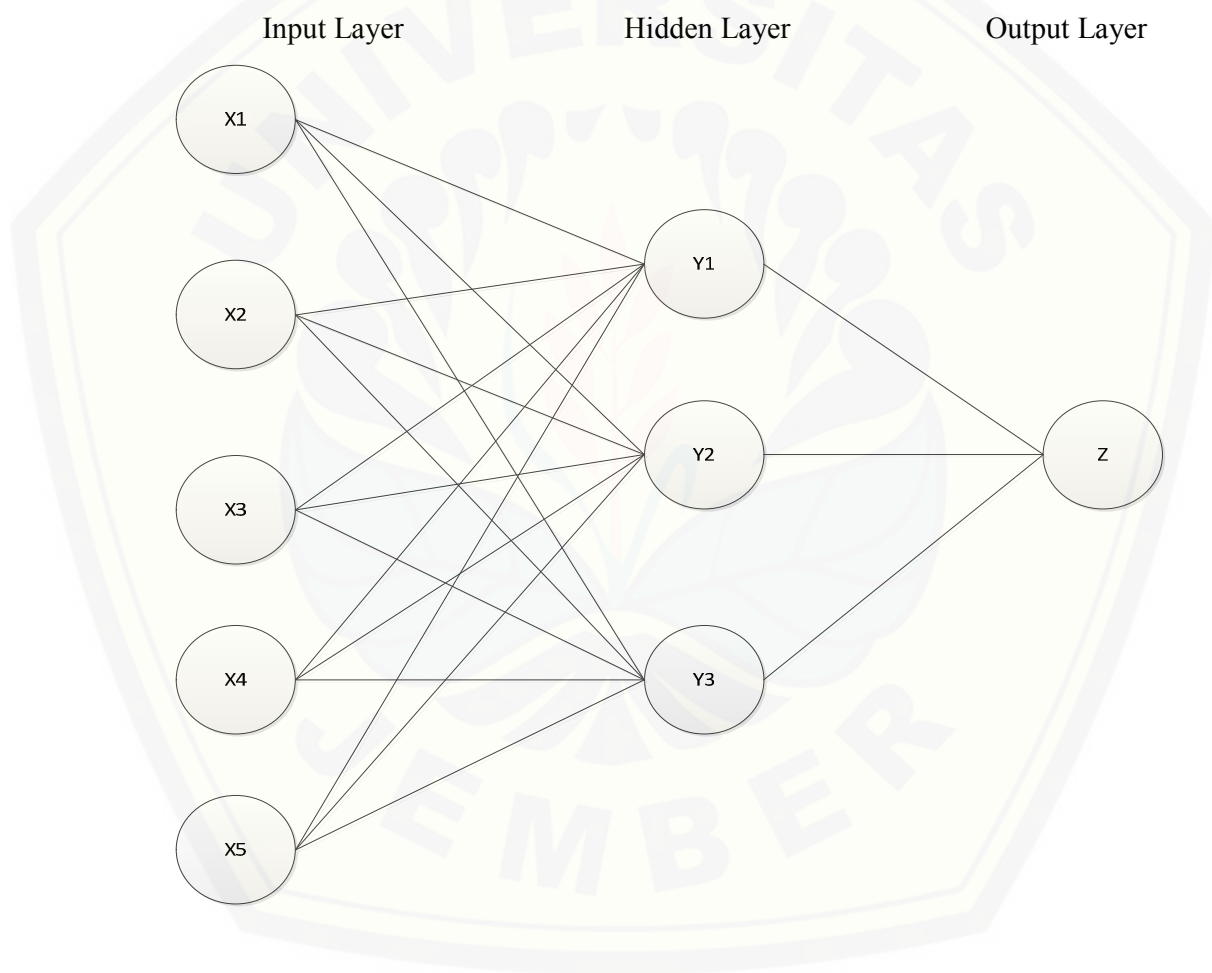
Sedangkan menurut Trisnawan (2013) DARPA *Neural Network Study* (1988, AFCEA International Press) jaringan syaraf tiruan adalah sebuah jaringan syaraf yang dibentuk dari sejumlah elemen pemroses sederhana yang bekerja secara paralel dimana fungsinya ditentukan oleh stuktur jaringan, kekuatan hubungan, dan pengolahan dilakukan pada komputasi elemen atau *nodes*. Jaringan syaraf tiruan memiliki kegunaan antara lain untuk perkiraan Fungsi, atau Analisis *Regresi*, termasuk prediksi *time series* dan modeling, klasifikasi, termasuk pengenalan pola dan pengenalan urutan, serta pengambil keputusan dalam pengurutan, pengolahan data, termasuk penyaringan, pengelompokan, kompresi, dan program dari robot yang bergerak secara mandiri tanpa di gerakan oleh manusia.

Pemrosesan informasi pada jaringan syaraf tiruan ini terjadi pada elemen sederhana (*neuron*), kemudian sinyal dikirimkan diantara *neuron-neuron* melalui penghubung (*dendrit* dan *akson*) tetapi penghubung antar elemen memiliki bobot yang akan menambah atau mengurangi sinyal, selanjutnya untuk menentukan output, setiap neuron memiliki fungsi aktivasi (biasanya non linier) yang

dikenakan pada semua *input* dan besar *output* akan dibandingkan dengan *threshold*.

Suatu model Jaringan Syaraf Tiruan (Irawan, 2012) ditentukan oleh:

- Pola antar neuron (arsitektur jaringan).
- Metode untuk menentukan dan mengubah bobot (disebut metode learning).
- Fungsi aktivasi.
- Jaringan Syaraf Tiruan disebut juga: *brain metaphor*, *computational neuroscience*, *parallel distributed processing*.



Gambar 2.12 Struktur jaringan syaraf Tiruan  
(Novikagianto, 2012)

Berdasarkan Gambar 2.18 adalah *backpropagation* dengan 3 (tiga) *layer*, yakni *input layer*, *hidden layer*, dan *output layer*. *Input layer* terdiri dari  $x_1$ ,  $x_2$ ,  $x_3$  yang merupakan data yang diinputkan, angka 1 (satu) merupakan bias pada *input layer*. Pada *hidden layer* terdapat  $z_1$ ,  $z_2$  yang merupakan penghubung antara sinyal *input* ke sinyal *output*. Angka 1 (satu) merupakan bias pada *hidden layer*. Pada *output layer* terdapat  $y_1$  yang merupakan *output* dari *backpropagation* itu sendiri.

a. Metode Pembelajaran

Cara berlangsungnya pembelajaran atau pelatihan jaringan syaraf tiruan dikelompokkan menjadi 3 (tiga) yaitu: (Jek Siang, 2005)

1) *Supervised learning* (pembelajaran terawasi)

Pada metode ini, setiap pola yang diberikan kedalam jaringan syaraf tiruan telah diketahui *output*nya. Selisih antara pola *output* yang dihasilkan dengan pola *output* yang dikehendaki yang disebut *target error*. *Target error* digunakan untuk mengoreksi bobot jaringan syaraf tiruan sehingga jaringan syaraf tiruan mampu menghasilkan *output* yang sedekat mungkin dengan pola yang telah diketahui oleh jaringan syaraf tiruan. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yakni: *Hebbian*, *Perceptron*, *ADALINE*, *Boltzman*, *Holpfield*, *Backpropagation*, *LVQ*.

2) *Unsupervised learning* (pembelajaran tak terawasi)

Pada metode ini, tidak memerlukan *target output*. Pada metode ini juga tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah menglompokkan unit-unit yang hampir sama dalam area tertentu. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yakni : Kohonen

3) *Hybrid learning* (pembelajaran hibrida)

Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*. Sebagian dari bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah *RBF*.

b. Fungsi Aktivasi

Jaringan syaraf tiruan mempunyai fungsi aktivasi yang digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya) (Jek Siang, 2005)

$$\text{Jika } net = \sum x_i w_i \dots\dots\dots(8)$$

maka fungsi aktivasinya adalah

$$f(net) = f(\sum x_i w_i) \dots\dots\dots(9)$$

Beberapa fungsi aktivasi yang sering digunakan adalah sebagai berikut:

1) Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \alpha \\ 0, & \text{jika } x < \alpha \end{cases} \dots\dots\dots(10)$$

Fungsi *threshold* merupakan fungsi *threshold* biner. Untuk kasus bilangan bipolar, maka angka 0 diganti dengan angka -1. Sehingga persamaan diubah menjadi :

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \alpha \\ -1, & \text{jika } x < \alpha \end{cases} \dots\dots\dots(11)$$

Adakalanya dalam jaringan syaraf tiruan ditambahkan suatu unit masukan yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah *input* yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi 0.

2) Fungsi sigmoid

$$f(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(12)$$

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk dideferensiasikan.

$$f(x) = f(x)(1 - f(x)) \dots\dots\dots(13)$$

3) Fungsi identitas

$$f(x) = x \dots\dots\dots(14)$$

Digunakan jika keluaran yang dihasilkan oleh jaringan syaraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range [0,1] atau [-1,1]).



c. Algoritma *Backpropagation*

Algoritma *Backpropagation* meliputi 3 (tiga) fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran.

Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

4.1 Fase I: Propagasi maju

Selama propagasi maju, sinyal masukan ( $x_i$ ) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit tersembunyi ( $z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $y_k$ ).

Berikutnya, keluaran jaringan ( $y_k$ ) dibandingkan dengan target yang harus dicapai ( $t_k$ ). Selisih antara  $t_k - y_k$  adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

4.2 Fase II: Propagasi mundur

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k$  ( $k = 1, 2, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

#### 4.3 Fase III: Perubahan bobot

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas  $\delta_k$  yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

#### 2.7 Algoritma *Levenberg-Marquardt*

Algoritma *Levenberg-Marquardt* merupakan pengembangan dari algoritma *error backpropagation*. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma *error backpropagation* dengan memanfaatkan teknik optimasi numerik standar yaitu menggunakan pendekatan matriks *Jacobian*.

Tujuan dari *Levenberg-Marquardt* adalah meminimalkan total *error*.

Berikut beberapa struktur algoritma *Levenberg-Marquardt*:

- a. Inisialisasi bobot dan bias dengan bilangan acak, *epoch* maksimum, minimal *goal* (*performance* yang dihitung dengan MSE)
- b. Menentukan parameter yang dibutuhkan, antara lain:
  - 1) Parameter *Levenberg-Marquardt* yang nilainya harus lebih besar dari nol
  - 2) Parameter faktor *input* dan bias yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *levenberg-Marquardt*.
- c. Menghitung maju (*feedforward*) pada *hidden* dan *otuput layer* seperti langkah-langkah pada algoritma *error backpropagation*.
- d. Menghitung nilai *Mean Square Error* (MSE)
- e. Menghitung error dan total *error* jaringan
  - 1) Rumus untuk *error* :

$$er = tr - yr \dots\dots\dots(15)$$

$r$  merupakan *input* ke- $r$

2) Rumus untuk menghitung total *error* :

$$e = [e_1 \ e_2 \ e_3 \ \dots \ e_N]^T \dots \dots \dots (16)$$

$e$  merupakan vektor kesalahan berukuran  $N \times 1$  yang terdiri dari  $e_r$

$r = 1, 2, 3, \dots, N$

f. Menghitung matriks *Jacobian*  $J(x)$

$x$  merupakan matriks yang berisi nilai bobot dan bias dari keseluruhan jaringan.

$$X = [v_{11}, v_{12}, \dots, v_{ij}; v_{01}, v_{02}, \dots; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02}, \dots, w_{0k}]$$

Matriks *Jacobian* berisi turunan pertama *error* jaringan terhadap bobot dan bias jaringan. Rumus untuk mencari *Jacobian* Matriks antara lain.

$$J = \left[ \frac{\partial e_r}{\partial w} \right] \dots \dots \dots (17)$$

g. Setelah didapatkan nilai  $J(x)$  maka dapat dihitung perubahan koreksi bobot dan biasnya dengan rumus berikut:

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * \text{Gradient}$$

Gradient :

$$J(x)^T e \dots \dots \dots (18)$$

h. Setelah didapatkan nilai  $\Delta x$  tahap selanjutnya adalah pengkoreksian bobot dengan rumus yang sama seperti pada algoritma *error backpropagation*.

i. Menghitung maju (*feedforward*) dengan bobot dan bias yang baru.

j. Menghitung MSE jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti

k. Jika epoch atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan:

- a. MSE naik
- b. MSE turun

Dimana:

- $e_r$  = *error* ke-  $r$
- $t_r$  = target ke- $r$
- $y_r$  = *output* ke- $r$

$r$  = input ke-r

$J(x)$  = matriks Jacobian

## 2.8 Mesin Perontok Padi dengan motor BLDC

Kemampuan motor BLDC yang lebih efisien dibandingkan motor DC lainnya jika di terapkan pada mesin perontok padi. Dengan adanya keunggulan dari motor BLDC ini bisa menekan biaya perawatan untuk mesin perontok padi yang konvensional pada umumnya. Motor BLDC digunakan pada mesin perontok padi diharapkan untuk meningkatkan efisiensi, mengurangi polusi udara dan suara serta mendapatkan kesenangan dalam memanen padi dan dapat mengurangi kerugian dan kerusakan selama memanen padi (Sujanarko, Kuswardhani & Septiandrini, 2016).

*Power Threshing Efficiency* (PTE) adalah total daya yang dikonsumsi dalam watt/jam untuk berat total bulir padi yang di panen oleh mesin perontok padi. Dapat di tuliskan kedalam persamaan sebagai berikut:

$$PTE = \frac{PW}{GT} \dots\dots\dots(19)$$

Keterangan:

PTE = *Power Threshing Efficiency* (Wh/ kg)

PW = *Power consumption* (Wh)

GT = *Total weight of grains* (Kg)

*The Threshing Efficiency* (TE) rasio total berat padi yang rontok dalam berat padi yang di panen oleh perontok padi ini. Hal ini dapat di tunjukkan oleh persamaan berikut:

$$TE = 100 - \left( \frac{GU}{GT} \times 100 \right) \dots\dots\dots(20)$$

Keterangan:

TE = *The Threshing Efficiency* (%)

GU = *Total weight of unthreshed grains* (Kg)

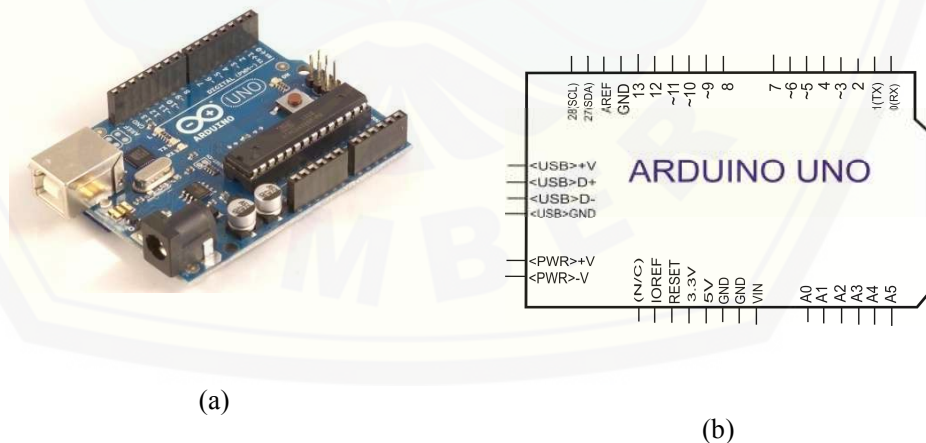
GT = *Total weight of grains* (Kg)

Pada penelitian sebelumnya telah dilakukan menggunakan motor BLDC namun menggunakan kontrol otomatis (Sujanarko, Kuswardhani & Septiandrini, 2016) namun kekurangan kontrol otomatis ini adalah motor BLDC tidak bisa bekerja secara maksimal sehingga di butuhkan kontrol yang lebih kompleks, sehingga pada penelitian ini menggunakan *neural network*.

### 2.8.1 Arduino Uno R3

Arduino uno adalah modul mikrokontroler berbasis ATmega 328. Modul ini memiliki empat belas pin *input-output* (14 I/O) di mana enam diantaranya dapat digunakan sebagai *output pulse width modulation* (PWM), enam *input* analog, sebuah osilator kristal 16 MHz, sebuah koneksi USB, tombol *reset* dan koneksi ke catu daya. Kontroler ini semua yang dibutuhkan untuk mendukung mikrokontroler.

“Uno” berarti satu dalam bahasa Itali dan hal ini untuk menandai versi pertama (Arduino 1.0) yang akan menjadi versi acuan bagi Arduino berikutnya. Uno adalah seri terakhir dari seri papan atau modul USB Arduino. Uno tidak lagi menggunakan *chip* FTDI USB-to-serial tetapi menggunakan Atmega 8U2 sebagai konverter USB ke serial. Gambar 2.20 (a) adalah modul Arduino uno R3 sedangkan Gambar 2.20 (b) menunjukkan skema dari modul Arduino uno R3 yang merupakan penyederhanaan dari gambar modul aslinya (Arduino Uno 2015).



Gambar 2. 13 (a) Modul kontroler arduino UNO-R3 (b) Pinout arduino UNO-R3  
(Sumber : [www.arduino.com](http://www.arduino.com))

Masing-masing pin I/O dapat digunakan sebagai *input* maupun *output* dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Masing-masing pin ini beroperasi pada tegangan 5 Volt dan dapat menerima arus 40 mA serta sudah memiliki *pull-up* resistor 20-50 k $\Omega$  yang secara *default* tidak tersambung (*disconnected*).

3Berikut ini adalah tabel spesifikasi dari arduino uno R3 dapat dilihat pada Tabel 2.2 dibawah ini.

Tabel 2. 3 Spesifikasi Arduino Uno R3

<i>Device</i>	Keterangan
<i>Microcontroller</i>	ATmega328
Tegangan Operasi	5 V
Tegangan <i>Input</i>	7-12 V
I/O <i>Digital</i>	14 (6 buah menyediakan PWM output)
<i>Input Analog</i>	6 buah
Arus DC per I/O pin	40 mA
Arus DC untuk pin 3.3	50 mA
<i>Flash Memory</i>	<i>Bootloader</i>
SRAM	2 Kb
EEPROM	1 Kb
Kecepatan <i>Clock</i>	16 MHz

Diterjemahkan dari: *datasheet* arduino uno R3

## 2.9 Photodiode

*Photodiode* digunakan sebagai komponen pendeteksi ada tidaknya cahaya maupun dapat digunakan untuk membentuk sebuah alat ukur akurat yang dapat mendeteksi intensitas cahaya dibawah 1 pW/cm<sup>2</sup> sampai intensitas diatas 10 mW/cm<sup>2</sup>. *Photo* dioda mempunyai resistansi yang rendah pada kondisi *forward* bias, kita dapat memanfaatkan photo dioda ini pada kondisi *reverse* bias dimana resistansi dari photo dioda akan turun seiring dengan intensitas cahaya yang masuk. Komponen ini mempunyai sensitivitas yang lebih baik jika dibandingkan dengan diodapeka cahaya. Halini disebabkan karena *electron* yang ditimbulkan oleh foton cahaya pada junction ini diinjeksikan di bagian base dan diperkuat di bagian kolektornya. Namun demikian, waktu respons dari transistor foto secara umum akan lebih lambat dari pada dioda peka cahaya.

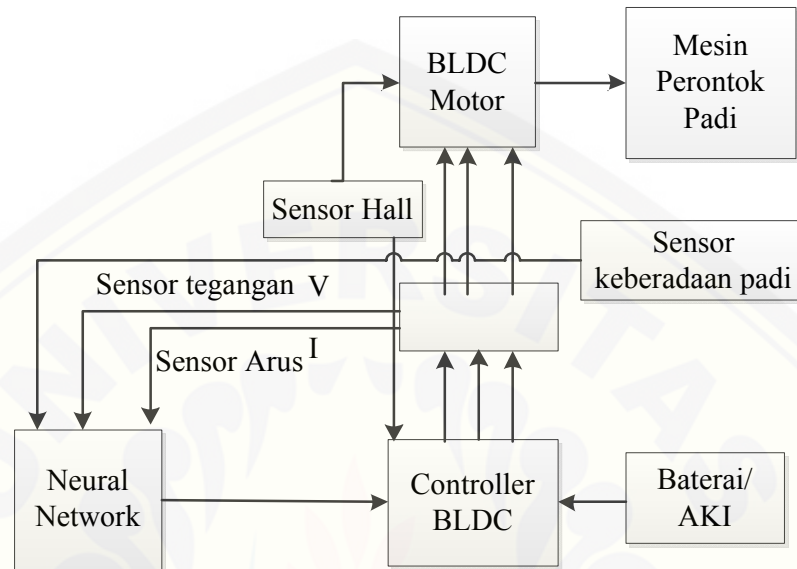
Jika photo dioda tidak terkena cahaya, maka tidak ada arus yang mengalir ke rangkaian pembanding, jika *photo* dioda terkena cahaya maka photodiode akan

bersifat sebagai tegangan, sehingga Vcc dan photo dioda tersusun seri, akibatnya terdapat arus yang mengalir ke rangkaian pembanding.



### BAB 3. METODOLOGI PENELITIAN

#### 3.1 Diagram Blok Sistem



Gambar 3. 1 Diagram Blok Sistem

Perancangan diagram blok sistem pada gambar 3.1 adalah rancangan secara sederhana dari penelitian yang dilakukan. Prinsip kerja blok diagram diatas adalah motor BLDC yang di supply oleh Aki/baterai berupa tegangan DC yang masuk melalui iverter dan keluaran yang dihasilkan berupa tegangan AC yang akan menggerakkan motor BLDC. Adanya sensor akan mendeteksi keberadaan padi, fungsi dari sensor ini adalah sebagai *switch* untuk menghidupkan motor BLDC sehingga daya yang terpakai oleh sistem ini lebih hemat energy dan lebih efisien. Arus dan tegangan yang akan dipakai untuk mengatur kecepatan motor BLDC jika diberi beban yang lebih sehingga kecepatan yang diberikan oleh motor BLDC konstan. Pengaturan kecepatan pada sisteim ini menggunakan *neural network* didapat dari hasil pengambilan data sebagai data input untuk perancangan sistem *levenberg marquadt neural network*. Dengan mendapatkan hasil pemodelan tersebut selanjutnya membuat simulasi untuk control kecepatan pada mesin perontok padi.



### 3.2 Parameter *Levenberg Marquadt Neural Network*

Pada pembuatan sistem *levenberg marquadt neural network* membutuhkan beberapa parameter untuk merancang sistem yang ingin kita buat. Berikut ini rancangan parameternya:

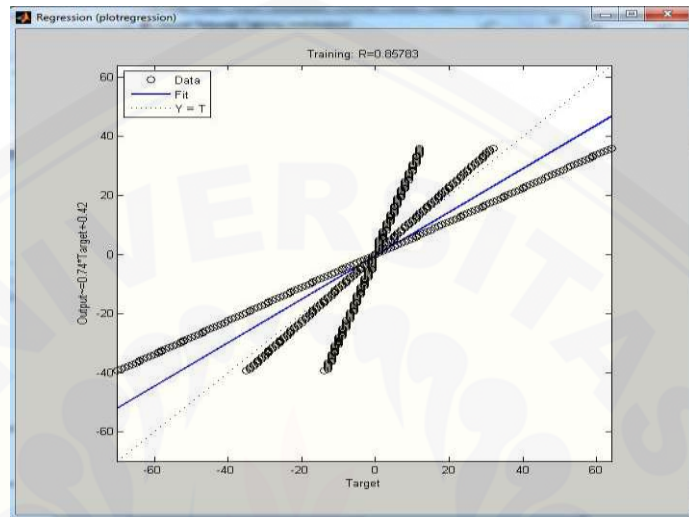
1. Fungsi transfer: 'tansig' untuk input, 'purelin' untuk output
2. Epoch (iterasi maksimum): digunakan 1000 epoch
3. Goal (error toleransi): untuk mendapatkan hasil yang baik digunakan *error* toleransi mendekati nol yaitu  $10^{-6}$ .
4. *Hidden Layer*: digunakan 1 buah hidden layer
5. Train: pelatihan jaringan pada *neural network* menggunakan *levenberg marquadt* (trainlm)

Program *neural network* bisa di masukkan jika sudah mendapatkan parameter pada layer 1 dan layer 2. Untuk layer 1 dimana kita harus mengetahui nilai setiap parameter dari setiap neuron yang dihasilkan oleh program *neural network*. Untuk mendapatkan program *neural network* agar bisa dimasukkan ke arduino adalah sebagai berikut :

1. Membuat program *neural network* dengan *backpropogation* dengan metode *levenberg marquadth*. Program matlabnya sebagai berikut :

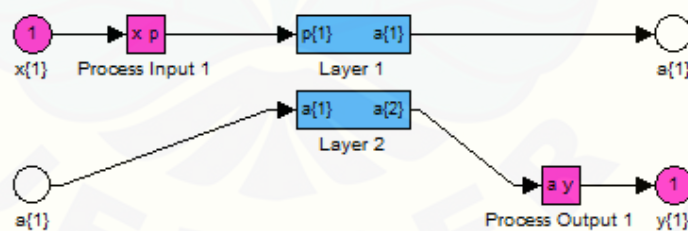
```
P= inputT;
T= targetT;
Net = newff(minmax(P), [10
1], {'tansig', 'purelin'}, 'trainlm');
net.trainParam.goal = 0;
net.trainParam.epochs = 1000;
net.trainParam.lr = 0.1;
net.trainParam.show = 10;
[net,tr] = train(net,P,T);
Y= sim(net,P);
gensim (net);
```

2. Setelah itu kita bisa menjalankan program *neural network* yang dihasilkan.
3. Kita terus mengulang training program *neural network* sampai *regression* yang dihasil mendekati 1. Untuk tampilan *regression* yang dihasilkan sebagai berikut:



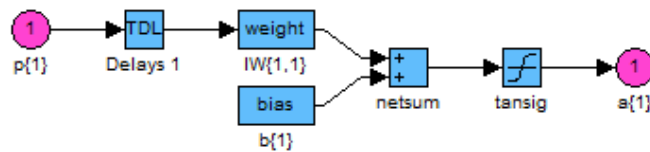
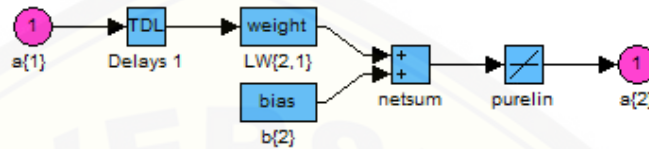
Gambar 3. 2 Grafik Parameter *Regression Backpropagation*

4. Mengambil data yang ada dalam pada *hidden layer* pada setiap *layer 1* dan 2.



Gambar 3. 3 Diagram Blok Parameter *Hidden Layer* Pada *Neural Network*

5. Data yang diambil merupakan data *weight* dan *bias* yang ada pada setiap *layer 1* dan 2.

Gambar 3. 4 Blok Sistem *Weight* dan *Bias* pada *Layer 1*Gambar 3. 5 Blok Sistem *Weight* dan *Bias* Pada *Layer 2*

6. Parameter weight dan bias inilah yang akan di masukkan ke dalam program arduino.

```

SulhanNNOK
#include <MatrixMath.h>
#include <Math.h>

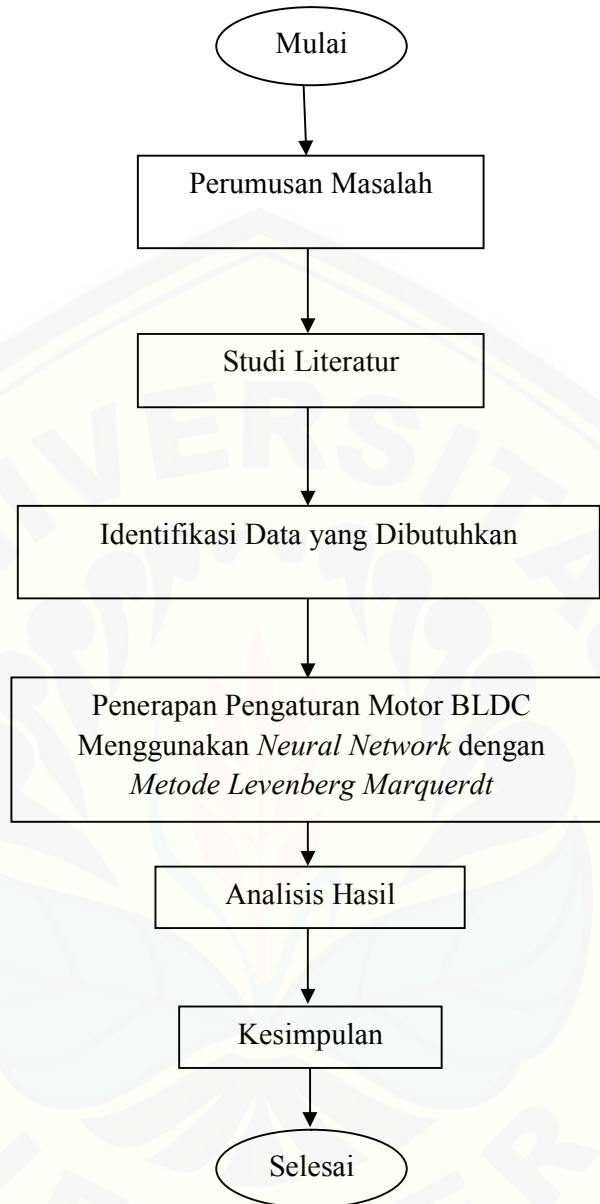
volatile byte half_revolutions;
unsigned int rpm;
unsigned long timeold;
float avgrpm;
int a = 0;

float WeightA[11][2] = {{ -0.01958263820343812 , -0.0039040153984162581},
float BiasA[10][1] = {{0.53812521345230269} , { -3.4094308404168481} ,
float WeightB[10][1] = {{ -26.182735118998938} , {4.050379317016052} ,
float BiasB = -5.5332020480082589; //BiasB
float Layer1[10][1]; //Layer1
float Layer2 = 0; //Layer2
float Input[1][2] = {0, 0}; //Matrix Input
int PWMVal = 0;

```

Gambar 3. 6 Tampilan Program Arduino Ketika *Weight* dan *Bias* di Masukkan

### 3.3 Diagram Alir Penelitian



Gambar 3. 7 Flowchart Penelitian

Diagram alur penelitian pada gambar 3.2 merupakan perancangan secara keseluruhan dari penelitian yang akan dilakukan. Mulai dari menyiapkan data berupa input data berupa arus dan tegangan. kemudian data diolah untuk membuat sistem *neural network* yang akan dimasukkan ke parameter pelatihan sehingga kita bisa mendapatkan pola pembelajaran pada jaringan neuron yang ada pada

*neural network*. Pelatihan akan terus di ulang sehingga kita bisa mendapatkan hasil yang baik. Jika data sudah baik maka akan dimasukkan ke simulasi *simulink* pada matlab.



## BAB 5 PENUTUP

### 5.2 Kesimpulan

Berdasarkan pengujian yang telah dilakukan dari penelitian dengan judul “Desain Mesin Perontok Padi Menggunakan Motor *Brushless Direct Current* Dengan Kontrol Berbasis *Neural Network* ” maka dapat ditarik beberapa kesimpulan sebagai berikut.

1. Pengaturan kontrol kecepatan motor BLDC pada mesin perontok padi dengan *neural network* menggunakan arduino UNO R3 lebih stabil dan konstan mencapai kecepatan 400 Rpm saat di beri beban dibandingkan dengan tanpa kontrol pada mesin perontok padi ketika diberi beban 1 kg konstan pada kecepatan 375 Rpm, jika diberi beban 2 kg konstan pada kecepatan 369 Rpm, jika diberi beban 3 kg konstan pada kecepatan 334 Rpm, jika diberi beban 4 kg konstan pada kecepatan 304 Rpm.
2. Pada pengujian mesin perontok padi menggunakan motor BLDC dengan kontrol *neural network* untuk mencapai kecepatan stabil yaitu : tanpa beban membutuhkan waktu 7 detik, pada beban 1 kg membutuhkan waktu 7 detik, pada beban 2 kg membutuhkan waktu 15 detik, pada beban 3 kg membutuhkan waktu 5 detik dan pada beban 4 kg membutuhkan waktu 4 detik, hal ini disebabkan karena perbedaan nilai *weight* pada setiap *hidden layer* berubah jika beban yang diberikan berubah.

### 5.3 Saran

Berdasarkan hasil penelitian dan pengujian alat yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk lebih menyempurnakan hasil penelitian ini dan untuk dikembangkan lebih lanjut antara lain :

1. Penggunaan mesin perontok padi yang kapasitas beban yang diberikan lebih besar sehingga beban dapat melebihi kapasitas 4 Kg.

2. Pengujian dan *training neural network* bisa dilakukan lebih dari 1000 kali sehingga bisa mendapatkan *regression* yang lebih baik untuk mendapatkan hasil yang lebih optimal dalam mengontrol kecepatan motor BLDC.
3. Menggunakan arduino lebih dari satu sehingga sensor dalam memberikan *feedback* pada kontrol bisa lebih cepat dan kecepatan motor lebih stabil dan sempurna.



DAFTAR PUSTAKA

- Ajav, E., & B.A, A. 2005. Performance Evaluation Of An Okra Thresher. *Jurnal Cigr Ejournal*. 7, pp: 1-8.
- Ariani, N.A. 2013. Modifikasi Mesin Perontok Padi (*Power Thresher*) Tipe *Throw-in*. *Skripsi Fakultas Teknologi Pertanian Universitas Jember*.
- Dharmawan, Abe. 2009. *Pengendali Motor DC Brushless dengan Metode PWM Sinusoidal Menggunakan ATmega 16*. Universitas Indonesia.
- D'Souza, Stan. 2004. *Sensored BLDC Motor Control Using dsPIC30F2010*. Microchip Technology.
- Freescale Semiconductor, 2005. *BLDC Motor with Speed Closed Loop and DC-Bus Break Controller, Driven by eTPU on MPC5554*.
- Makridakis, S., Wheelright, S.C., & McGee, V. E. (1999). *Metode dan Aplikasi Peramalan*, (U.S. Andriyanto dan A. Basith, terj.). Jakarta: Erlangga.
- Microsemi. *Speed Control of Brushless DC Motors-Sinusoidal Commutation with Hall Sensors*.
- Novikagianto. 2012 *Desain ANN untuk Identifikasi Sistem*. <https://novikagianto.wordpress.com/kategori/materi/artificial-neural-network/>. [17 maret 2016]
- NXP Semiconductor. 2007. *Brushless DC motor control using the LPC2141*.
- Rachmat, R., Setyono, A., & Tharir, R. 1993. *Evaluasi Sistem Pemanenan Berek Menggunakan Beberapa Mesin Perontok, Agrimex*. 4 & 5 (1). pp 1-7
- Shao, Jianwen. 2003. *Direct Back EMF Detection Method for Sensorless Brushless DC (BLDC) Motor Drives*. Virginia Polytechnic Institute and the State University.
- Sucipta, N., & Suriasih, K. 2004. Modifikasi Mesin Perontok Padi Dengan Pendekatan Ergonomi Dapat Meningkatkan Hasil Perontokkan. *Jurnal Seminar Nasional Sosialisasi dan Komunikasi Hasil-Hasil Penelitian dan Pengkajian Program Studi Teknologi Pertanian Universitas Udayana*, pp: 1-28



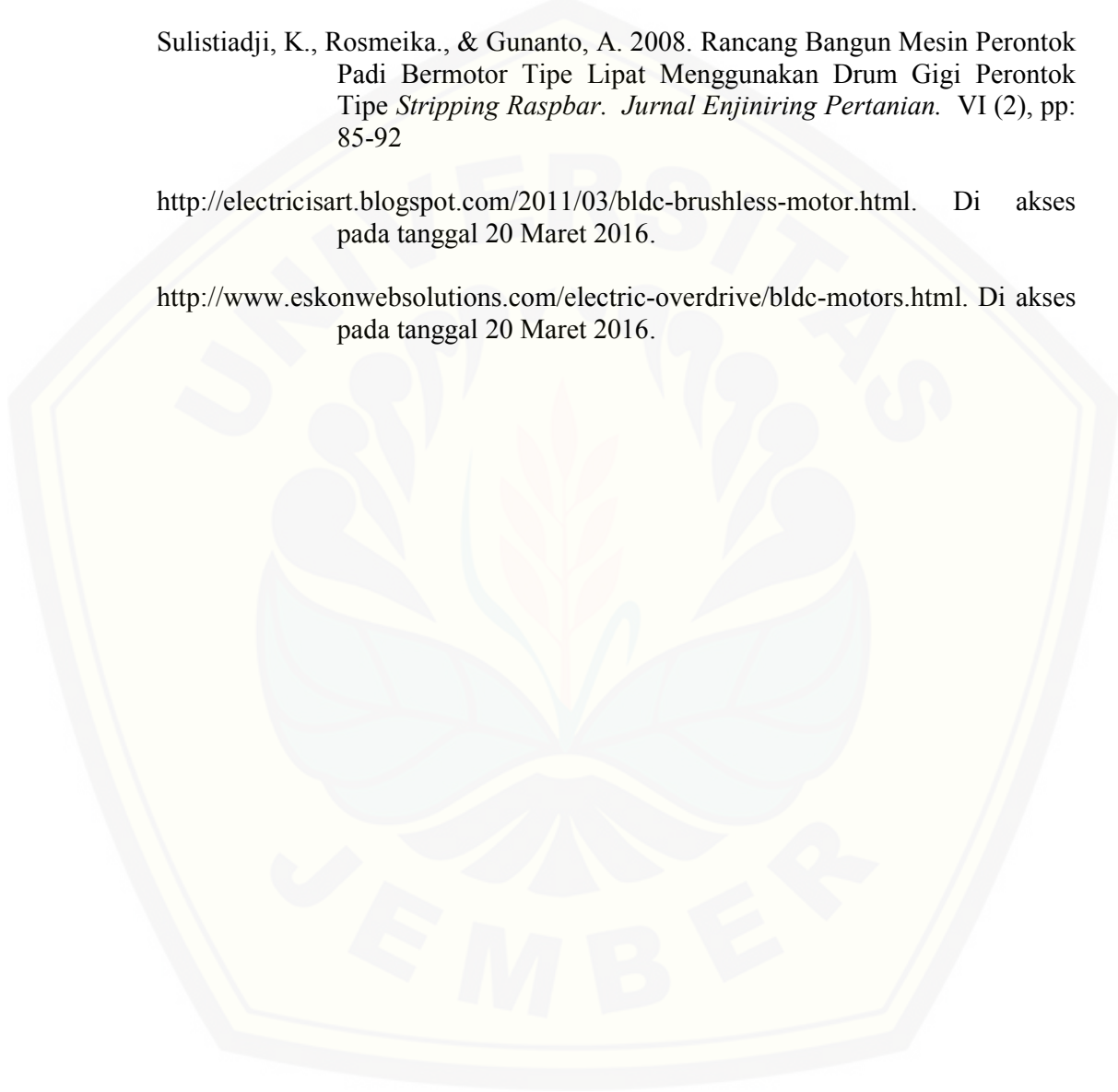
Sujanarko, B., Kuswardhani, N., & Andrini, V.S. 2016. Performance Improvement of Paddy Threshing Unit Using Brushless Direct Current Motor. *Jurnal International Journal of Engineering Research*. 5 (1), pp: 59-61

Sujanarko, Bambang. 2012. *BLDC Controller Using Digital Logic For Electric Vehicle*. University Of Jember. Seminar Retii, Yogyakarta.

Sulistiadji, K., Rosmeika., & Gunanto, A. 2008. Rancang Bangun Mesin Perontok Padi Bermotor Tipe Lipat Menggunakan Drum Gigi Perontok Tipe *Stripping Raspbar*. *Jurnal Enjiniring Pertanian*. VI (2), pp: 85-92

<http://electricisart.blogspot.com/2011/03/bldc-brushless-motor.html>. Di akses pada tanggal 20 Maret 2016.

<http://www.eskonwebsolutions.com/electric-overdrive/bldc-motors.html>. Di akses pada tanggal 20 Maret 2016.



## LAMPIRAN

### LAMPIRAN A. Listing Program Arduino UNO R3

```
#include <MatrixMath.h>
#include <Math.h>
volatile byte half_revolutions;
unsigned int rpm;
unsigned long timeold;
float avgrpm;
int a = 0;

float WeightA[11][2] = {{ -0.01958263820343812 , -0.0039040153984162581},
{0.052848518846518117 , 0.010576341056076001}, {0.48187412140960534 ,
0.094760767841874802}, {0.064758900504074554 , 0.43390418143190512}, { -
0.021360440564142054 , -0.0043824269184829668}, {0.60766623361981098 ,
0.2285738041019241}, {0.23574845539849812 , 0.25353096870323055}, {1.3452340232278051 ,
0.00087819694272425535}, {0.04513755941874778 , 0.0085498094956017179},
{1.6696140151983334 , -0.0066131370512092051}}; //BebanA
float BiasA[10][1] = {{0.53812521345230269} , {-3.4094308404168481} , {-21.313737280600055}
, {-2.9926297545029739} , {-1.4103399745562102} , {9.5613544897124534} ,
{27.857763123592136} , {34.690947768655519} , {0.97285063842755581} ,
{66.09009328676251}}; //BiasA
float WeightB[10][1] = {{ -26.182735118998938} , {4.050379317016052} ,
{0.030616301583532488} , {0.011200679113240516}, {-23.14512917781698},
{0.013899523401586584}, {-0.026333709336066342}, {0.016524744056740297},
{2.5367850905188223}, {0.024714012345111336}}; //BebanB
float BiasB = -5.5332020480082589; //BiasB
float Layer1[10][1]; //Layer1
float Layer2 = 0; //Layer2
float Input[1][2] = {0, 0}; //Matrix Input
int PWMVal = 0;

void setup() {
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  analogWrite(9, 100);
  delay(2000);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
```

```
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
attachInterrupt(0, rpm_fun, RISING);
half_revolutions = 0;
rpm = 0;
timeold = 0;
avgrpm = 0;
Serial.println("READY");

}

void loop() {

if (half_revolutions >= 1) {
rpm = 30 * 1000 / (millis() - timeold) * half_revolutions;
timeold = millis();
half_revolutions = 0;
avgrpm = avgrpm + rpm;
a++;
}

if (a > 3) {
avgrpm = avgrpm / 3;
avgrpm = (0.3615 * avgrpm) - 1.7716;
Serial.print(avgrpm, DEC);
Serial.println("");
nn();
avgrpm = 0;
a = 0;
}
}

void rpm_fun()
{
half_revolutions++;
}

void nn() {
Input[0][0] = 400 - avgrpm;

//Serial.print("");
```

```
//Serial.print(Input[0][0]);

Input[0][1] = Input[0][0] - Input[0][1];
//Serial.print("");
//Serial.print(Input[0][1]);
//Serial.print("");

Input[0][1] = Input[0][0];

for (int wA1 = 0; wA1 <= 9; wA1++) {
  Layer1[wA1][0] = tanh((Input[0][0] * WeightA[wA1][0]) + (Input[0][1] * WeightA[wA1][1]) +
BiasA[wA1][0]);
}

for (int wB1 = 0; wB1 <= 9; wB1++) {
  Layer2 = Layer2 + (Layer1[wB1][0] * WeightB[wB1][0]);
}

Layer2 = Layer2 + BiasB;

if (PWMVal < 255 && Layer2 > 0 || PWMVal > 0 && Layer2 < 0)
  PWMVal = PWMVal + Layer2;

if (PWMVal < 0)
  PWMVal = 0;

if (PWMVal > 255)
  PWMVal = 255;

//Serial.print(Layer2);
//Serial.print("");
//Serial.println(PWMVal);
analogWrite(9, PWMVal);
Layer2 = 0;
}
```

## LAMPIRAN B. Listing Program Monitoring VB

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports
Imports Mitov.PlotLab
Imports Excel = Microsoft.Office.Interop.Excel

Public Class frmDisplay
    Dim comOpen As Boolean      'Keeps track of the port status. True =
Open; False = Closed
    Dim readbuffer As String   'Buffer of whats read from the serial port
    Dim time As Integer = 0
    Dim plotScopeA As Double
    Dim timeSet As Integer
    Dim excelready As Integer
    Dim strFileName As String

    Dim APP As Excel.Application
    Dim worksheet As Excel.Worksheet
    Dim workbook As Excel.Workbook
    Dim y As Integer = 2
    Dim jarakt As Double = 0
    Dim jarakn As Double
    Dim speedt As Double = 0
    Dim speedn As Double
    Dim va As Double

    Private Sub frmDisplay_Load(ByVal sender As Object, ByVal e As
EventArgs) Handles MyBase.Load

        'Get all connected serial ports
        Dim comPorts As String() = System.IO.Ports.SerialPort.GetPortNames

        If comPorts.Count < 1 Then
            'If there are not ports connected, show an error and close the
program.
            MsgBox("There are no com ports available! Closing program.")
            Me.Close()
        Else
            cmbPort.Items.AddRange(comPorts)
            cmbPort.Text = comPorts(0)
        End If

        Scope1.XAxis.AxisLabel.Text = "Sampling"
        Scope1.YAxis.AxisLabel.Text = "Amplitude"

    End Sub

    Private Sub frmDisplay_FormClosing(ByVal sender As Object, ByVal e As
FormClosingEventArgs) Handles Me.FormClosing
        'Gracefully disconnect before form closes
        If excelready = 1 Then
            workbook.Save()
            workbook.Close()
        End If
    End Sub
End Class
```

```
End If

DoDisconnect()
End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As System.Object,
ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
    If comOpen Then

        Try
            'Send data to a new thread to update the ph display
            readbuffer = SerialPort1.ReadLine()
            Me.Invoke(New EventHandler(AddressOf updateTemp))
        Catch ex As Exception
            'Otherwise show error. Will display when disconnecting.
            'MsgBox(ex.Message)
        End Try
    End If
End Sub

Public Sub updateTemp(ByVal sender As Object, ByVal e As
System.EventArgs)
    'Update ph display as it comes in
    Dim kode As Integer = 0
    Dim read As String
    Dim aryTextFile() As String

    read = readbuffer.Replace(vbCr, "").Replace(vbLf, "")
    aryTextFile = read.Split("|")

    Speed.Text = aryTextFile(0).Replace(".", ",")
    Speed.Text = FormatNumber(Speed.Text, 2)
    plotScopeA = Speed.Text

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    time = time + 1

    Scope1.Channels(0).Data.AddXYPoint(time, Speed.Text)
    Scope1.Channels(1).Data.AddXYPoint(time, 400)

    worksheet.Cells(time, 1).Value = TimeString
    worksheet.Cells(time, 2).Value = Speed.Text
    workbook.Save()

End Sub

Public Sub DoDisconnect()
    'Graceful disconnect if port is open
    If comOpen Then
        SerialPort1.DiscardInBuffer()
        SerialPort1.Close()
    End If
End Sub
```

```
'Reset our flag and controls
comOpen = False
btnDisconnect.Enabled = False
btnConnect.Enabled = True
cmbBaud.Enabled = True
cmbPort.Enabled = True

End If
End Sub

Public Sub DoConnect()
'Setup the serial port connection
With SerialPort1()
.PortName = cmbPort.Text           'Selected Port
.BaudRate = CInt(cmbBaud.Text)     'Baud Rate. 9600 is default.
.Parity = IO.Ports.Parity.None
.DataBits = 8
.StopBits = IO.Ports.StopBits.One
.Handshake = IO.Ports.Handshake.None
.RtsEnable = False
.ReceivedBytesThreshold = 1
.NewLine = vbCrLf
.ReadTimeout = 10000
End With

'Try to open the selected port...
Try
SerialPort1.Open()
comOpen = SerialPort1.IsOpen
Catch ex As Exception
comOpen = False
MsgBox("Error Open: " & ex.Message)
End Try

btnDisconnect.Enabled = True
btnConnect.Enabled = False
cmbBaud.Enabled = False
cmbPort.Enabled = False
End Sub

Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConnect.Click
'Conect to serial port

If namafile.Text = "" Then
MsgBox.Show("Data Excel Belum Terload", "Connecting
Cancelled", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

If namafile.Text <> "" Then
excelready = 1
End If

If excelready = 1 Then
DoConnect()
Timer1.Enabled = True
```

```
End If

End Sub

Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisconnect.Click
    'Disconnect the serial port
    Timer1.Enabled = False

    If excelready = 1 Then
        workbook.Save()
    End If

    If excelready <> 1 Then
        MessageBox.Show("Tidak Ada Aktivitas", "Disconnect Cancelled",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If

    DoDisconnect()
End Sub

Private Sub Ok_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Ok.Click
    timeSet = TimeBox.Text
    TPlot.Text = TimeBox.Text
    Timer1.Interval = timeSet
End Sub

Private Sub LoadExcel_Click(sender As Object, e As EventArgs) Handles LoadExcel.Click
    OpenFileDialog1.Filter = "Excel (*.xlsx)|*.xlsx"
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        namafile.Text = OpenFileDialog1.FileName
    End If

    If namafile.Text <> "" Then
        APP = New Excel.Application
        workbook = APP.Workbooks.Open(namafile.Text)
        worksheet = workbook.Worksheets("sheet1")
    End If

End Sub
End Class
```



**LAMPIRAN C. Gambar Hardware**

C1. Gambar keseluruhan sistem mesin perontok padi



C2. Gambar motor BLDC



C3. Sensor kecepatan motor



C4. Pengujian Mesin Perontok Padi dan Pengambilan Data



C5. Pengujian Motor BLDC dan Proses Kalibrasi Sensor



C6. Proses monitoring dan pengambilan data dengan *visual basic*



**LAMPIRAN D. Data Hasil Pengujian**

D1. Pengujian Mesin Perontok Padi Tanpa Kontrol pada Tanpa Beban

Waktu	RPM	22:48:51	400.00	22:49:11	398.00
22:48:32	0	22:48:52	400.00	22:49:12	398.00
22:48:33	398	22:48:53	400.00	22:49:13	398.00
22:48:34	404	22:48:54	400.00	22:49:14	398.00
22:48:35	404	22:48:55	400.00	22:49:15	398.00
22:48:36	401	22:48:56	400.00	22:49:16	398.00
22:48:37	401	22:48:57	400.00	22:49:17	398.00
22:48:38	401	22:48:58	400.00	22:49:18	398.00
22:48:39	401	22:48:59	400.00	22:49:19	398.00
22:48:40	401	22:48:59	400.00	22:49:20	398.00
22:48:41	401	22:49:00	400.00	22:49:21	398.00
22:48:42	401	22:49:02	398.00	22:49:22	398.00
22:48:43	401	22:49:02	398.00	22:49:23	398.00
22:48:44	401	22:49:03	398.00	22:49:24	398.00
22:48:45	401.00	22:49:04	398.00	22:49:25	398.00
22:48:45	401.00	22:49:05	398.00	22:49:26	398.00
22:48:46	401.00	22:49:06	398.00	22:49:27	398.00
22:48:47	400.00	22:49:07	398.00	22:49:28	398.00
22:48:48	400.00	22:49:08	398.00	0:50:12	403.00
22:48:49	400.00	22:49:09	398.00		
22:48:50	400.00	22:49:10	398.00		

## D2. Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 1 Kg

Waktu	Kecepatan
23:02:31	0
23:02:32	0
23:02:33	57
23:02:34	57
23:02:35	57
23:02:36	57
23:02:37	57
23:02:38	57
23:02:39	57
23:02:40	57
23:02:41	57
23:02:42	57
23:02:43	57
23:02:44	57
23:02:45	57
23:02:46	57
23:02:47	57
23:02:48	375
23:02:49	375
23:02:50	375
23:02:51	375
23:02:52	375
23:02:53	375
23:02:54	375
23:02:55	375
23:02:56	375
23:02:57	375
23:02:58	375
23:02:59	375
23:03:00	375
23:03:01	375
23:03:02	375
23:03:03	375
23:03:04	375
23:03:05	375
23:03:06	375

23:03:07	375
23:03:08	375
23:03:09	375
23:03:10	375
23:03:11	375
23:03:12	375
23:03:13	375
23:03:14	375
23:03:15	375
23:03:16	375
23:03:17	375
23:03:18	387
23:03:19	387
23:03:20	387
23:03:21	387
23:03:22	387
23:03:23	387
23:03:24	387
23:03:25	387
23:03:26	387
23:03:27	387
23:03:28	387
23:03:29	387
23:03:30	387
23:03:31	387
23:03:32	387
23:03:33	387
23:03:34	387
23:03:35	387
23:03:36	387
23:03:37	387
23:03:38	387
23:03:39	387
23:03:40	387
23:03:41	387
23:03:42	387
23:03:43	387

23:03:44	387
23:03:45	387
23:03:46	387
23:03:47	387
23:03:48	387
23:03:49	387
23:03:50	387
23:03:51	387
23:03:52	387
23:03:53	387
23:03:54	387
23:03:55	387
23:03:56	387
23:03:57	387
23:03:58	387
23:03:59	387
23:04:00	387
23:04:01	387
23:04:02	387
23:04:03	387
23:04:03	387
23:04:04	387
23:04:06	387
23:04:07	387
23:04:08	387
23:04:08	387
23:04:09	387
23:04:10	387
23:04:11	387
23:04:12	387
23:04:13	387
23:04:14	387
23:04:15	387
23:04:16	381
23:04:17	381
23:04:18	381
23:04:19	381

23:04:20	381
23:04:21	381
23:04:22	381
23:04:23	381
23:04:24	381
23:04:25	381
23:04:26	381
23:04:27	381
23:04:28	381
23:04:29	381
23:04:30	381
23:04:31	381
23:04:32	381
23:04:33	381
23:04:34	381
23:04:35	381
23:04:36	381
23:04:37	381
23:04:38	381
23:04:39	381
23:04:40	381
23:04:41	381
23:04:42	381
23:04:43	381
23:04:44	381
23:04:45	381
23:04:46	381
23:04:47	381
23:04:48	381
23:04:49	381
23:04:50	381
23:04:51	381
23:04:52	381
23:04:53	381
23:04:54	381
23:04:55	381
23:04:56	381
23:04:57	381
23:04:58	381
23:04:59	381
23:05:00	381

23:05:01	381
23:05:01	381
23:05:02	381
23:05:03	381
23:05:04	381
23:05:05	381
23:05:06	381
23:05:07	381
23:05:08	381
23:05:09	381
23:05:10	381
23:05:11	381
23:05:12	381
23:05:13	381
23:05:14	381
23:05:15	381
23:05:16	381
23:05:17	381
23:05:18	381
23:05:19	381
23:05:20	381
23:05:21	381
23:05:22	381
23:05:23	381
23:05:24	381
23:05:25	381
23:05:26	381
23:05:27	381
23:05:28	381
23:05:29	381
23:05:30	381
23:05:31	381
23:05:32	381
23:05:33	381
23:05:34	381
23:05:35	381
23:05:36	381
23:05:37	381
23:05:38	381
23:05:39	381
23:05:40	381

23:05:41	381
23:05:42	381
23:05:42	381
23:05:43	381
23:05:44	381
23:05:45	381
23:05:46	381
23:05:47	381
23:05:48	381
23:05:49	381
23:05:50	381
23:05:51	381
23:05:52	381
23:05:53	381
23:05:54	381
23:05:55	381
23:05:56	381
23:05:57	381
23:05:58	381
23:05:59	381
23:06:00	381
23:06:01	381
23:06:02	381
23:06:03	381
23:06:04	381
23:06:05	381
23:06:06	381
23:06:07	381
23:06:08	381
23:06:09	381
23:06:09	381
23:06:10	381
23:06:11	381
23:06:12	381
23:06:13	381
23:06:14	381
23:06:15	381
23:06:16	381
23:06:17	381
23:06:18	381
23:06:19	381

23:06:20	381
23:06:21	381
23:06:21	381
23:06:22	381
23:06:23	381
23:06:24	381
23:06:25	381
23:06:26	381
23:06:27	381
23:06:28	381
23:06:29	381
23:06:30	381
23:06:31	381
23:06:32	381
23:06:33	381
23:06:34	381
23:06:35	381
23:06:36	381
23:06:37	381
23:06:38	381
23:06:39	381
23:06:40	381
23:06:41	381
23:06:42	381
23:06:43	381
23:06:44	381
23:06:45	381
23:06:46	381
23:06:47	381
23:06:48	381
23:06:49	381
23:06:50	381
23:06:51	381
23:06:52	381
23:06:53	381
23:06:54	381
23:06:55	381
23:06:56	381
23:06:57	381
23:06:58	381
23:06:59	381

23:07:00	381
23:07:01	381
23:07:02	381
23:07:03	381
23:07:04	381
23:07:05	381
23:07:06	381
23:07:07	381
23:07:08	381
23:07:09	381
23:07:10	381
23:07:10	381
23:07:11	381
23:07:12	381
23:07:13	381
23:07:14	381
23:07:15	381
23:07:16	381
23:07:18	381
23:07:18	381
23:07:19	381
23:07:20	381
23:07:21	381
23:07:22	381
23:07:23	381
23:07:24	381
23:07:25	381
23:07:26	381
23:07:27	381
23:07:28	381
23:07:29	381
23:07:30	381
23:07:31	381
23:07:32	381
23:07:33	381
23:07:34	381
23:07:35	381
23:07:36	381
23:07:37	381
23:07:38	381
23:07:39	381

23:07:40	381
23:07:41	381
23:07:42	381
23:07:43	381
23:07:44	381
23:07:45	381
23:07:46	381
23:07:47	381
23:07:48	381
23:07:49	381
23:07:50	381
23:07:51	381
23:07:52	381
23:07:53	381
23:07:54	381
23:07:55	381
23:07:56	381
23:07:57	381
23:07:58	381
23:07:58	381
23:07:59	381
23:08:00	381
23:08:01	381
23:08:02	381
23:08:03	381
23:08:04	381
23:08:05	381
23:08:06	381
23:08:07	381
23:08:08	381
23:08:09	381
23:08:10	381
23:08:11	381
23:08:12	381
23:08:13	381
23:08:14	381
23:08:15	381
23:08:16	381
23:08:17	381
23:08:18	381
23:08:19	381

23:08:20	381
23:08:21	381
23:08:22	381
23:08:23	381
23:08:24	381
23:08:25	381
23:08:26	381
23:08:27	381
23:08:28	381
23:08:29	381
23:08:30	381
23:08:31	381
23:08:32	381
23:08:33	381
23:08:34	381
23:08:35	381
23:08:36	381
23:08:37	381
23:08:38	381
23:08:39	381
23:08:40	381
23:08:41	381
23:08:42	381
23:08:43	381
23:08:44	381
23:08:45	381
23:08:46	381
23:08:47	381
23:08:48	381
23:08:49	381
23:08:50	381
23:08:51	381
23:08:52	381
23:08:53	381
23:08:54	381
23:08:55	381
23:08:56	381
23:08:57	381
23:08:58	381
23:08:59	381
23:09:00	381

23:09:01	381
23:09:02	381
23:09:03	381
23:09:04	381
23:09:05	381
23:09:05	381
23:09:06	381
23:09:07	381
23:09:08	381
23:09:09	381
23:09:10	381
23:09:11	381
23:09:12	381
23:09:13	381
23:09:14	381
23:09:15	381
23:09:16	381
23:09:17	381
23:09:18	381
23:09:19	381
23:09:20	381
23:09:21	381
23:09:22	381
23:09:23	381
23:09:24	381
23:09:25	381
23:09:26	381
23:09:27	381
23:09:28	381
23:09:29	381
23:09:30	381
23:09:31	381
23:09:32	381
23:09:33	381
23:09:34	381
23:09:35	381
23:09:36	381
23:09:37	381
23:09:38	381
23:09:39	381
23:09:39	381

23:09:40	381
23:09:41	381
23:09:42	381
23:09:43	381
23:09:44	381
23:09:45	381
23:09:46	381
23:09:47	381
23:09:48	381
23:09:49	381
23:09:50	381
23:09:51	381
23:09:52	381
23:09:53	381
23:09:54	381
23:09:55	381
23:09:56	381
23:09:57	381
23:09:58	381
23:09:59	381
23:10:00	381
23:10:01	381
23:10:02	381
23:10:03	381
23:10:04	381
23:10:05	381
23:10:06	381
23:10:07	381
23:10:08	381
23:10:09	381
23:10:10	381
23:10:11	381
23:10:12	381
23:10:13	381
23:10:13	381
23:10:14	381
23:10:15	381
23:10:16	381
23:10:17	381
23:10:18	381
23:10:19	381

23:10:20	381
23:10:21	381
23:10:22	381
23:10:23	381
23:10:24	381
23:10:25	381
23:10:26	381
23:10:27	381
23:10:28	381
23:10:29	381
23:10:30	381
23:10:31	381
23:10:32	381
23:10:33	381
23:10:34	381
23:10:35	381
23:10:36	381
23:10:37	381
23:10:38	381
23:10:39	381
23:10:40	381
23:10:41	381
23:10:42	381
23:10:43	381
23:10:44	381
23:10:45	381
23:10:46	381
23:10:47	381
23:10:48	381
23:10:49	381
23:10:50	381
23:10:51	381
23:10:52	381
23:10:53	381
23:10:54	381
23:10:55	381
23:10:56	381
23:10:57	381
23:10:58	381
23:10:59	381
23:11:00	381

23:11:01	381
23:11:02	381
23:11:03	381
23:11:04	381
23:11:05	381
23:11:06	381
23:11:07	381
23:11:08	381
23:11:09	381
23:11:10	381
23:11:11	381
23:11:11	381
23:11:12	381
23:11:14	381
23:11:14	381
23:11:16	381
23:11:17	381
23:11:17	381
23:11:18	381
23:11:19	381
23:11:20	381
23:11:21	381
23:11:22	381
23:11:23	381
23:11:24	381
23:11:25	381
23:11:26	381
23:11:27	381
23:11:28	381
23:11:29	381
23:11:30	381
23:11:31	381
23:11:32	381
23:11:33	381
23:11:34	381
23:11:35	381
23:11:36	381
23:11:37	381
23:11:38	381
23:11:39	381
23:11:40	381

23:11:41	381
23:11:42	381
23:11:43	381
23:11:44	381
23:11:45	381
23:11:46	381
23:11:47	381
23:11:48	381
23:11:49	381
23:11:50	381
23:11:51	381
23:11:52	381
23:11:53	381
23:11:54	381
23:11:55	381
23:11:56	381
23:11:57	381
23:11:58	381
23:11:59	381
23:12:00	381
23:12:01	381
23:12:02	381
23:12:03	381
23:12:04	381
23:12:05	381
23:12:06	381
23:12:07	381
23:12:08	381
23:12:09	381
23:12:10	381
23:12:11	381
23:12:12	381
23:12:13	381
23:12:14	381
23:12:15	381
23:12:16	381
23:12:17	381
23:12:17	381
23:12:19	381
23:12:20	381
23:12:21	381



23:12:22	381
23:12:23	381
23:12:24	381
23:12:25	381
23:12:26	381
23:12:27	381
23:12:28	381
23:12:29	381
23:12:30	381

23:12:31	381
23:12:32	381
23:12:33	381
23:12:34	381
23:12:35	381
23:12:36	381
23:12:37	381
23:12:38	381
23:12:39	381

23:12:40	381
23:12:41	381
23:12:42	381
23:12:43	381
23:12:44	381
23:12:45	381
23:12:46	381
23:12:47	381
23:12:48	381

### D3. Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 2 Kg

Waktu	kecepatan
23:07:16	0
23:07:17	0
23:07:18	172
23:07:18	172
23:07:19	158
23:07:19	158
23:07:20	144
23:07:20	144
23:07:21	128
23:07:22	128
23:07:22	111
23:07:23	111
23:07:23	93
23:07:24	93
23:07:24	93
23:07:25	72
23:07:26	72
23:07:26	72
23:07:27	45
23:07:27	45
23:07:28	45
23:07:29	45
23:07:29	151
23:07:30	151
23:07:30	151
23:07:31	151
23:07:31	369

23:07:32	369
23:07:32	369
23:07:33	369
23:07:33	369
23:07:34	369
23:07:34	369
23:07:35	369
23:07:35	369
23:07:36	369
23:07:37	369
23:07:37	369
23:07:38	369
23:07:38	369
23:07:39	369
23:07:39	369
23:07:40	369
23:07:40	369
23:07:41	369
23:07:42	375
23:07:42	375
23:07:43	375
23:07:43	375
23:07:44	375
23:07:44	375
23:07:45	375
23:07:45	369
23:07:46	369
23:07:47	369

23:07:47	369
23:07:48	369
23:07:48	369
23:07:49	369
23:07:49	369
23:07:50	369
23:07:50	369
23:07:51	369
23:07:52	369
23:07:52	369
23:07:53	369
23:07:53	369
23:07:54	369
23:07:54	369
23:07:55	369
23:07:55	369
23:07:56	369
23:07:57	369
23:07:57	369
23:07:58	369
23:07:58	369
23:07:59	369
23:07:59	369
23:08:00	369
23:08:00	369
23:08:01	369
23:08:01	369
23:08:02	369

23:08:02	369
23:08:03	369
23:08:03	369
23:08:04	369
23:08:05	369
23:08:05	369
23:08:06	369
23:08:06	369
23:08:07	369
23:08:07	369
23:08:08	369
23:08:08	369
23:08:09	369
23:08:10	369
23:08:10	369
23:08:11	369
23:08:11	369
23:08:12	369
23:08:12	369
23:08:13	369
23:08:13	369
23:08:14	369
23:08:14	369
23:08:15	369
23:08:15	369
23:08:16	369
23:08:17	369

23:08:18	369
23:08:18	369
23:08:19	369
23:08:19	369
23:08:20	369
23:08:20	369
23:08:21	369
23:08:22	369
23:08:22	369
23:08:23	369
23:08:24	369
23:08:24	369
23:08:25	369
23:08:25	369
23:08:26	369
23:08:26	369
23:08:27	369
23:08:27	369
23:08:28	369
23:08:29	369
23:08:29	369
23:08:30	369
23:08:30	369
23:08:31	369
23:08:32	369
23:08:32	369
23:08:33	369

23:08:34	369
23:08:34	369
23:08:35	369
23:08:35	369
23:08:36	369
23:08:36	369
23:08:37	370
23:08:37	369
23:08:38	369
23:08:38	369
23:08:39	369
23:08:40	369
23:08:40	366
23:08:41	369
23:08:42	369
23:08:42	369
23:08:43	356
23:08:43	369
23:08:44	369
23:08:45	369
23:08:45	369
23:08:46	369
23:08:46	369
23:08:47	369
23:08:47	369
23:08:48	369

D4. Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 3 Kg

Waktu	Kecepatan
23:55:58	0
23:55:59	0
23:56:00	0
23:56:01	0
23:56:01	93
23:56:02	93
23:56:03	72
23:56:04	72
23:56:05	72

23:56:06	45
23:56:07	45
23:56:08	45
23:56:09	45
23:56:10	190
23:56:11	190
23:56:12	151
23:56:13	151
23:56:14	334
23:56:15	335

23:56:16	335
23:56:17	335
23:56:18	335
23:56:19	335
23:56:20	335
23:56:21	335
23:56:22	335
23:56:23	335
23:56:25	325
23:56:26	334

23:56:27	334
23:56:28	334
23:56:30	334
23:56:31	334
23:56:32	334
23:56:33	334
23:56:34	334
23:56:35	334
23:56:36	334
23:56:37	334
23:56:38	334
23:56:39	334
23:56:40	334
23:56:41	334
23:56:43	334
23:56:44	334
23:56:45	334
23:56:46	334

23:56:47	334
23:56:48	334
23:56:49	334
23:54:23	334
23:54:24	334
23:54:25	334
23:54:26	334
23:54:27	334
23:54:28	334
23:54:29	334
23:54:30	334
23:54:31	334
23:54:32	334
23:54:33	334
23:54:34	334
23:54:35	334
23:54:36	334
23:54:37	334

23:54:38	334
23:54:39	334
23:54:40	334
23:54:41	334
23:54:41	334
23:54:42	334
23:54:43	334
23:54:44	334
23:54:45	334
23:54:46	334
23:54:47	334
23:54:48	334
23:54:49	334
23:54:50	334
23:54:51	334
23:54:52	334
23:54:53	334

## D5. Pengujian Mesin Perontok Padi Tanpa Kontrol pada Beban 4 Kg

Waktu	Kecepatan
23:07:17	0
23:07:18	0
23:07:18	0
23:07:19	0
23:07:19	0
23:07:20	0
23:07:20	0
23:07:21	0
23:07:22	0
23:07:22	0
23:07:23	0
23:07:23	93
23:07:24	93
23:07:24	93
23:07:25	72
23:07:26	72
23:07:26	72
23:07:27	185

23:07:27	172
23:07:28	172
23:07:29	158
23:07:29	158
23:07:30	144
23:07:30	144
23:07:31	128
23:07:31	128
23:07:32	111
23:07:32	111
23:07:33	310
23:07:33	309
23:07:34	309
23:07:34	309
23:07:35	302
23:07:35	302
23:07:36	302
23:07:37	302
23:07:37	304

23:07:38	304
23:07:38	304
23:07:39	304
23:07:39	304
23:07:40	304
23:07:40	304
23:07:41	304
23:07:42	304
23:07:42	304
23:07:43	304
23:07:43	304
23:07:44	304
23:07:44	304
23:07:45	304
23:07:45	304
23:07:46	304
23:07:47	301
23:07:47	301
23:07:48	301

23:07:48	301
23:07:49	301
23:07:49	301
23:07:50	301
23:07:50	301
23:07:51	301
23:07:52	301
23:07:52	301
23:07:53	301
23:07:53	301
23:07:54	301
23:07:54	301
23:07:55	301
23:07:55	301
23:07:56	301
23:07:57	301
23:07:57	301
23:07:58	301
23:07:58	301
23:07:59	301
23:07:59	301
23:08:00	301
23:08:00	301
23:08:01	301
23:08:01	301
23:08:02	301
23:08:02	301
23:08:03	301
23:08:03	301
23:08:04	301
23:08:05	301
23:08:05	301
23:08:06	301
23:08:06	301
23:08:07	301
23:08:07	301

23:08:08	301
23:08:08	301
23:08:09	301
23:08:10	301
23:08:10	301
23:08:11	301
23:08:11	301
23:08:12	301
23:08:12	301
23:08:13	301
23:08:13	301
23:08:14	301
23:08:14	301
23:08:15	301
23:08:15	301
23:08:16	301
23:08:17	301
23:08:18	301
23:08:18	301
23:08:19	301
23:08:19	301
23:08:20	301
23:08:20	301
23:08:21	301
23:08:22	301
23:08:22	301
23:08:23	301
23:08:24	301
23:08:24	301
23:08:25	301
23:08:25	301
23:08:26	301
23:08:26	301
23:08:27	301
23:08:27	301
23:08:28	301

23:08:29	301
23:08:29	301
23:08:30	301
23:08:30	301
23:08:31	301
23:08:32	301
23:08:32	301
23:08:33	301
23:08:34	301
23:08:34	301
23:08:35	301
23:08:35	301
23:08:36	301
23:08:36	301
23:08:37	301
23:08:37	301
23:08:38	301
23:08:38	301
23:08:39	301
23:08:40	301
23:08:40	301
23:08:41	301
23:08:42	301
23:08:42	301
23:08:43	301
23:08:43	301
23:08:44	301
23:08:45	301
23:08:45	301
23:08:46	301
23:08:46	301
23:08:47	301
23:08:47	301
23:08:48	301

## D6. Pengujian Mesin Perontok Padi Dengan Kontrol Tanpa Beban

Waktu	RPM
23:07:25	0
23:07:26	0
23:07:26	0
23:07:27	0
23:07:27	45
23:07:28	45
23:07:29	45
23:07:29	151
23:07:30	151
23:07:30	151
23:07:31	151
23:07:31	369
23:07:32	369
23:07:32	369
23:07:32	369
23:07:33	369
23:07:33	401
23:07:34	401
23:07:34	401
23:07:35	401
23:07:35	401
23:07:36	399
23:07:37	399
23:07:37	399
23:07:38	399
23:07:38	399
23:07:39	390
23:07:39	390
23:07:40	390
23:07:40	390
23:07:41	390
23:07:42	399
23:07:42	399
23:07:43	399
23:07:43	399
23:07:44	399
23:07:44	399
23:07:45	395
23:07:45	395

23:07:46	395
23:07:47	395
23:07:47	395
23:07:48	395
23:07:48	396
23:07:49	396
23:07:49	396
23:07:50	396
23:07:50	396
23:07:51	396
23:07:52	396
23:07:52	398
23:07:53	398
23:07:53	398
23:07:54	398
23:07:54	398
23:07:55	398
23:07:55	398
23:07:56	398
23:07:57	400
23:07:57	400
23:07:58	400
23:07:58	400
23:07:59	400
23:07:59	400
23:08:00	400
23:08:00	400
23:08:01	395
23:08:01	395
23:08:02	395
23:08:02	395
23:08:03	395
23:08:03	395
23:08:04	395
23:08:05	395
23:08:05	395
23:08:06	401
23:08:06	401
23:08:07	401

23:08:07	401
23:08:08	401
23:08:08	401
23:08:09	401
23:08:10	401
23:08:10	401
23:08:11	398
23:08:11	398
23:08:12	398
23:08:12	398
23:08:13	398
23:08:13	398
23:08:14	398
23:08:14	398
23:08:15	398
23:08:15	398
23:08:16	404
23:08:17	404
23:08:18	404
23:08:18	404
23:08:19	404
23:08:19	404
23:08:20	404
23:08:20	404
23:08:21	404
23:08:22	404
23:08:22	404
23:08:23	401
23:08:24	401
23:08:24	401
23:08:25	401
23:08:25	401
23:08:26	401
23:08:26	401
23:08:27	401
23:08:27	401
23:08:28	401
23:08:29	401
23:08:29	401

23:08:30	401
23:08:30	397
23:08:31	397
23:08:32	399
23:08:32	403
23:08:33	402
23:08:34	387
23:08:34	398
23:08:35	401
23:08:35	382
23:08:36	389
23:08:36	389

23:08:37	403
23:08:37	403
23:08:38	409
23:08:38	409
23:08:39	412
23:08:40	412
23:08:40	399
23:08:41	399
23:08:42	399
23:08:42	399
23:08:43	399
23:08:43	395

23:08:44	395
23:08:45	395
23:08:45	416
23:08:46	416
23:08:46	416
23:08:47	416
23:08:47	382
23:08:48	409

D7. Pengujian Mesin Perontok Padi Dengan Kontrol Beban 1Kg

Waktu	RPM
23:47:51	0
23:47:51	0
23:47:52	0
23:47:52	0
23:47:53	1
23:47:54	1
23:47:54	1
23:47:55	1
23:47:55	1
23:47:56	337
23:47:56	337
23:47:57	337
23:47:57	337
23:47:58	337
23:47:59	390
23:47:59	390
23:48:00	390
23:48:00	390
23:48:01	390
23:48:02	390
23:48:02	388
23:48:03	388
23:48:03	388
23:48:04	388

23:48:04	388
23:48:05	388
23:48:05	388
23:48:06	399
23:48:07	399
23:48:07	399
23:48:08	399
23:48:08	399
23:48:09	399
23:48:09	399
23:48:10	399
23:48:10	399
23:48:11	399
23:48:12	399
23:48:12	399
23:48:13	399
23:48:13	399
23:48:14	399
23:48:14	399
23:48:15	390
23:48:15	390
23:48:16	390
23:48:16	390
23:48:17	390
23:48:18	390

23:48:18	390
23:48:19	390
23:48:19	390
23:48:20	401
23:48:20	401
23:48:21	401
23:48:21	401
23:48:22	401
23:48:23	401
23:48:23	401
23:48:24	401
23:48:24	401
23:48:25	401
23:48:25	393
23:48:26	393
23:48:26	393
23:48:27	393
23:48:28	393
23:48:28	393
23:48:29	393
23:48:29	393
23:48:30	393
23:48:30	393
23:48:31	393
23:48:31	398

23:48:32	398
23:48:32	398
23:48:33	398
23:48:34	398
23:48:34	398
23:48:35	398
23:48:35	398
23:48:36	398
23:48:36	398
23:48:37	398
23:48:37	398
23:48:38	395
23:48:39	395
23:48:39	395
23:48:40	395
23:48:40	395
23:48:41	395
23:48:42	395
23:48:42	395
23:48:43	395
23:48:43	395
23:48:44	395
23:48:44	395
23:48:45	395
23:48:46	400
23:48:46	400
23:48:47	400
23:48:47	400
23:48:48	400
23:48:49	400

23:48:49	400
23:48:50	400
23:48:50	400
23:48:51	400
23:48:51	400
23:48:52	400
23:48:53	400
23:48:53	400
23:48:54	398
23:48:54	398
23:48:55	398
23:48:55	398
23:48:56	398
23:48:57	398
23:48:57	398
23:48:58	398
23:48:59	398
23:48:59	398
23:49:00	398
23:49:00	398
23:49:01	398
23:49:01	398
23:49:02	398
23:49:02	398
23:49:03	398
23:49:04	398
23:49:04	398
23:49:05	398
23:49:05	384
23:49:06	381

23:49:07	383
23:49:07	383
23:49:08	406
23:49:09	406
23:49:09	372
23:49:10	372
23:49:10	372
23:49:11	406
23:49:12	406
23:49:12	406
23:49:13	416
23:49:13	416
23:49:14	416
23:49:14	409
23:49:15	409
23:49:15	409
23:49:16	412
23:49:17	412
23:49:17	412
23:49:18	412
23:49:18	400
23:49:19	400
23:49:20	400
23:49:20	397
23:49:21	404
23:49:21	402
23:49:22	399
23:49:23	395

D8. Pengujian Mesin Perontok Padi Dengan Kontrol Beban 2 Kg

Waktu	RPM
0:14:51	0
0:14:51	0
0:14:52	0
0:14:53	134
0:14:53	134
0:14:54	134

0:14:54	134
0:14:55	214
0:14:55	214
0:14:56	214
0:14:56	214
0:14:57	214
0:14:58	248

0:14:58	248
0:14:59	248
0:14:59	248
0:15:00	248
0:15:00	240
0:15:01	240
0:15:01	240

0:15:02	240
0:15:03	240
0:15:03	240
0:15:04	343
0:15:04	343
0:15:05	343
0:15:05	343
0:15:06	343
0:15:07	343
0:15:07	393
0:15:08	393
0:15:08	393
0:15:09	393
0:15:09	393
0:15:10	393
0:15:11	393
0:15:11	402
0:15:12	402
0:15:13	402
0:15:13	402
0:15:14	402
0:15:14	402
0:15:15	402
0:15:15	402
0:15:16	403
0:15:16	403
0:15:17	403
0:15:18	403
0:15:18	403
0:15:19	403
0:15:19	403
0:15:20	403
0:15:20	403
0:15:21	395
0:15:21	395
0:15:22	395
0:15:23	395
0:15:23	395
0:15:24	395

0:15:24	395
0:15:25	395
0:15:25	395
0:15:26	395
0:15:26	401
0:15:27	401
0:15:28	401
0:15:28	401
0:15:29	401
0:15:29	401
0:15:30	401
0:15:30	401
0:15:31	401
0:15:32	401
0:15:32	401
0:15:33	398
0:15:33	398
0:15:34	398
0:15:35	398
0:15:35	398
0:15:36	398
0:15:36	398
0:15:37	398
0:15:37	398
0:15:38	398
0:15:38	398
0:15:39	398
0:15:40	398
0:15:40	405
0:15:41	405
0:15:41	405
0:15:42	405
0:15:42	405
0:15:43	405
0:15:44	405
0:15:44	405
0:15:45	405
0:15:45	405
0:15:46	405

0:15:46	405
0:15:47	405
0:15:48	405
0:15:48	404
0:15:49	404
0:15:49	404
0:15:50	404
0:15:50	404
0:15:51	404
0:15:51	404
0:15:52	404
0:15:53	404
0:15:54	404
0:15:54	404
0:15:55	404
0:15:55	404
0:15:56	404
0:15:56	404
0:15:57	404
0:15:57	401
0:15:58	401
0:15:58	401
0:15:59	401
0:16:00	401
0:16:00	401
0:16:01	401
0:16:01	401
0:16:02	401
0:16:02	402
0:16:03	404
0:16:04	401
0:16:04	401
0:16:05	404
0:16:06	404
0:16:06	401
0:16:07	401
0:16:08	401
0:16:08	401



## D9. Pengujian Mesin Perontok Padi Dengan Kontrol Beban 3 Kg

Waktu	RPM
0:51:41	0
0:51:41	0
0:51:42	0
0:51:42	230
0:51:43	230
0:51:43	377
0:51:44	377
0:51:44	382
0:51:45	382
0:51:45	384
0:51:46	384
0:51:47	398
0:51:47	398
0:51:48	398
0:51:48	379
0:51:49	379
0:51:49	379
0:51:50	390
0:51:51	390
0:51:51	390
0:51:52	382
0:51:52	382
0:51:53	382
0:51:53	382
0:51:54	388
0:51:54	388
0:51:55	388
0:51:56	388
0:51:56	385
0:51:57	385
0:51:57	385
0:51:58	385
0:51:59	387
0:51:59	387
0:52:00	387
0:52:00	387
0:52:01	387
0:52:01	387

0:52:02	387
0:52:03	387
0:52:03	387
0:52:04	387
0:52:04	387
0:52:05	392
0:52:05	392
0:52:06	392
0:52:07	392
0:52:07	392
0:52:08	392
0:52:08	392
0:52:09	392
0:52:09	392
0:52:10	392
0:52:11	392
0:52:11	392
0:52:12	392
0:52:12	392
0:52:13	398
0:52:13	398
0:52:14	398
0:52:14	398
0:52:15	398
0:52:15	398
0:52:16	398
0:52:16	398
0:52:17	404
0:52:17	404
0:52:18	404
0:52:19	404
0:52:19	404
0:52:20	404
0:52:20	404
0:52:21	404
0:52:22	404
0:52:22	401
0:52:23	401
0:52:23	406

0:52:24	385
0:52:25	385
0:52:25	413
0:52:26	413
0:52:27	409
0:52:27	409
0:52:28	409
0:52:28	403
0:52:29	403
0:52:30	403
0:52:30	395
0:52:31	395
0:52:31	395
0:52:32	412
0:52:32	412
0:52:33	412
0:52:33	412
0:52:34	412
0:52:35	412
0:52:35	412
0:52:36	412
0:52:36	407
0:52:37	407
0:52:37	407
0:52:38	407
0:52:38	410
0:52:39	410
0:52:40	410
0:52:40	410
0:52:41	410
0:52:41	402
0:52:42	402
0:52:42	402
0:52:43	402
0:52:44	402
0:52:44	416
0:52:45	416
0:52:45	416
0:52:46	416

D10. Pengujian Mesin Perontok Padi Dengan Kontrol Beban 4 Kg

Waktu	RPM
1:41:21	0
1:41:21	0
1:41:22	0
1:41:23	155
1:41:23	155
1:41:24	107
1:41:24	107
1:41:25	240
1:41:25	240
1:41:26	301
1:41:26	301

1:41:27	309
1:41:28	309
1:41:28	309
1:41:29	384
1:41:30	384
1:41:31	384
1:41:32	384
1:41:33	400
1:41:33	400
1:41:34	400
1:41:34	400
1:41:35	400

1:41:36	400
1:41:36	400
1:41:37	396
1:41:37	396
1:41:38	396
1:41:39	396
1:41:39	396
1:41:40	396
1:41:40	396
1:41:41	396
1:41:41	405

D11. Data Pengujian sebelum dan sesudah menggunakan Kontrol Tanpa Beban

Sebelum	Sesudah
0	0
398	0
404	0
404	0
401	45
401	45
401	45
401	151
401	151
401	151
401	151
401	369
401	369
401	369
401	369
401	401
400	401
400	401
400	401
400	401
400	399
400	399

400	399
400	399
400	399
400	390
400	390
400	390
400	390
400	390
400	390
400	399
398	399
398	399
398	399
398	399
398	399
398	395
398	395
398	395
398	395
398	395
398	395
398	395
398	395
398	396
398	396
398	396

398	396
398	396
398	396
398	396
398	398
398	398
398	398
398	398
398	398
398	398
398	398
398	398
398	400
398	400
398	400
398	400
398	400
398	400
398	400
398	395
398	395
398	395









334	398
334	398
334	398
334	404
334	404
334	404
334	404

334	404
334	404
334	404
334	404
334	404
334	401
334	401

334	406
334	385
334	385
334	413
334	413
334	409

D15. Data Pengujian sebelum dan sesudah menggunakan Kontrol pada Beban 4 Kg

Sebelum	Sesudah
0	0
0	0
0	0
0	155
0	155
0	107
0	107
0	240
0	240
0	301
0	301

93	309
93	309
93	309
72	384
72	384
72	384
185	384
172	400
172	400
158	400
158	400
144	400

144	400
128	400
128	396
111	396
111	396
310	396
309	396
309	396
309	396
302	396
302	405