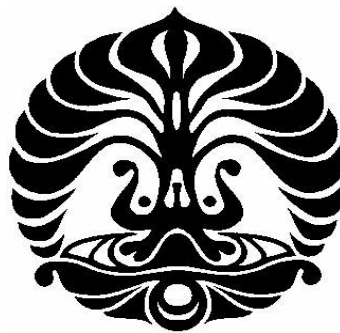


PERANGKAT LUNAK SISTEM PENEREMAN

Tugas Akhir

Diajukan Untuk Melengkapi Persyaratan Kurikulum Program

Diploma III Fisika Instrumentasi Elektronika



Agustina Wulandari

2304210413

PROGRAM DIII FISIKA INSTRUMENTASI ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

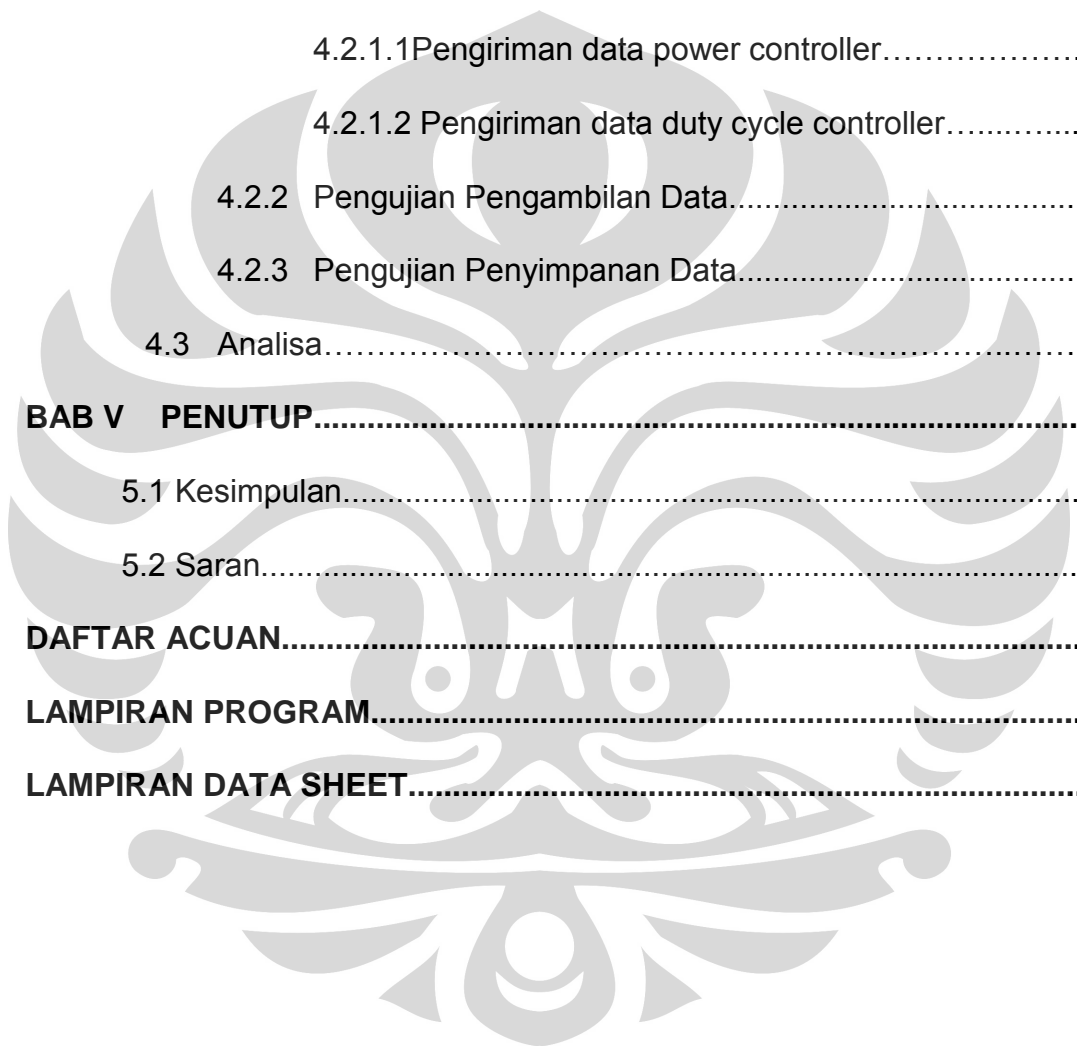
DEPOK

2007

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
KATA PENGANTAR	iii
DAFTAR ISI	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Pembatasan Masalah	3
1.4 Metode Penelitian	3
1.5 Sistematika Penulisan.....	5
BAB II TEORI DASAR	7
2.1 Gaya Gesek.....	7
2.1.1 Gaya Gesek Statik.....	8
2.1.2 Gaya Gesek Kinetik.....	9
2.2 Motor DC.....	11
2.2.1 Prinsip kerja motor dc.....	11
2.2.2 Cara pengendalian motor dc.....	15
2.3 Rem.....	17
2.4 Sensor Shaf Encoder.....	18
2.5 Roda Gigi (<i>Gear</i>).....	20

2.5.1	Macam-Macam Roda Gigi.....	21
2.6	Gerak Melingkar.....	22
2.8.1	Gerak melingkar beraturan.....	24
2.8.2	Gerak melingkar berubah beraturan.....	25
2.8.3	Gerak berubah beraturan.....	25
2.7	Momen Inersia.....	27
2.8	Konfigurasi Port Parallel.....	28
2.9	Pengenalan LabVIEW.....	32
BAB III	PERANCANGAN DAN CARA KERJA PROGRAM SISTEM	
	PENGREMAN.....	38
3.1	Rangkaian <i>Interfacing</i>	38
3.2	Flow Chart Global Modul Sistem Pengereman.....	40
3.2.1	Flow Chart pada LabVIEW.....	42
3.3	Pemrosesan Data.....	43
3.3.1	Menentukan Nilai Frekuensi Counter.....	46
3.4	Program Komunikasi Parallel.....	48
3.4.1	Pengiriman dan Pengambilan Data	48
3.4.2	Pengiriman data power motor.....	49
3.4.3	Pengiriman data duty cycle rem.....	50
3.4.4	Pengambilan data RPM.....	51
3.5	Penyimpanan data ke Microsoft Excel.....	53



BAB IV PENGUJIAN PROGRAM DAN ANALISA.....	55
4.1 Pengujian Rangkaian Interfacing.....	55
4.2 Pengujian Program.....	56
4.2.1 Pengujian Pengiriman Data.....	57
4.2.1.1 Pengiriman data power controller.....	57
4.2.1.2 Pengiriman data duty cycle controller.....	59
4.2.2 Pengujian Pengambilan Data.....	60
4.2.3 Pengujian Penyimpanan Data.....	80
4.3 Analisa.....	81
BAB V PENUTUP.....	82
5.1 Kesimpulan.....	82
5.2 Saran.....	83
DAFTAR ACUAN.....	84
LAMPIRAN PROGRAM.....	85
LAMPIRAN DATA SHEET.....	87

ABSTRAK

Telah dibuat sebuah Perangkat Lunak Sistem Pengereman menggunakan menggunakan program *LabVIEW* untuk mengatur atau sebagai kendali Modul Pengereman. Program ini yang akan menentukan nilai *Power Controller*, menentukan mode pengereman yaitu konvensional atau termodulasi dengan mengatur *Duty Cycle Controller* dan dari tiap mode pengereman tersebut dan kondisi penerimaan data. Cara program ini mengatur modul tersebut adalah dengan memanfaatkan koneksi parallel yang tersedia pada komputer *PC* sebagai jalur untuk mengirimkan dan menerima sinyal ke modul tersebut. Program ini juga akan mengambil serta mengolah data yang diterima dari modul tersebut lalu menampilkan data olahan tersebut dalam bentuk grafik Kecepatan (Km/Jam) dan menyimpannya dalam bentuk file Microsoft Excel agar dapat dianalisa hasilnya.

Kata kunci : Sistem Pengereman, LabVIEW, Power Controller, Duty Cycle Controller

BAB I

PENDAHULUAN

Pada Bab I akan dibahas mengenai Pendahuluan yaitu tentang latar belakang dibuatnya Tugas Akhir ini. Selain itu penulis juga akan membahas tujuan penulisan, pembatasan masalah, metode penelitian yang disertai skema penelitian, dan sistematika penulisan yaitu digunakan dalam penulisan ini dengan membagi menjadi beberapa bab dan masing-masing terdiri dari sub bab.

1.1 LATAR BELAKANG

Otomotif adalah ilmu yang mempelajari tentang alat-alat transportasi darat yang menggunakan mesin, terutama mobil dan sepeda motor. Otomotif mulai berkembang sebagai cabang ilmu seiring dengan diciptakannya mesin mobil. Perkembangan dalam bidang otomotif semakin lama semakin berkembang dengan pesat. Apabila melihat jumlah kecelakaan yang terjadi maka setiap perusahaan yang bergerak di bidang otomotif menitik-beratkan dalam hal keselamatan. Oleh karena itu, dibuat sistem pengereman agar keselamatan yang telah diprioritaskan perusahaan tersebut dapat terwujud. Para pengguna kendaraan bermotor mengatakan bahwa sistem pengereman secara on-off-on-off (termodulasi) lebih efektif dibandingkan dengan sistem pengereman langsung off (konvensional). Dengan alasan tersebut maka timbul pemikiran untuk mencoba membuktikan yang

mana yang lebih baik antara pengaruh dari rem konvensional dan rem termodulasi, sehingga pada tugas akhir ini akan dilakukan penelitian mengenai sistem pengereman.

Pada Tugas Akhir ini membandingkan sistem pengereman menggunakan 2 buah metode yaitu metode dengan sistem pengereman yang langsung mati (konvensional) dan metode sistem pengereman yang on-off-on-off (termodulasi). Dengan membandingkan dua metode tersebut dapat dibuktikan seberapa efektifnya sistem pengereman yang digunakan.

Modul sistem pengereman ini dibuat secara mekanik dengan rangkaian elektronik yang dapat menggerakkan sistem dan dapat mengendalikan kecepatan benda uji pada sistem pengereman. Penelitian dilakukan dengan cara menghubungkan komputer *PC* sebagai pengendali utama ke sistem elektronika dan microcontroller yang akan memproses data yang dikirim dari komputer sebagai informasi untuk keperluan pengaturan modul pengereman yang akan dijalankan pada sistem mekaniknya. Dan program pada komputer *PC* ini juga akan menangani data yang diambil dari sistem elektronika pada modul pengereman ini yang selanjutnya akan diproses dan dianalisa oleh program.

1.2 TUJUAN PENELITIAN

1. Membuat modul sistem pengereman dan membandingkan antara metode konvensional (langsung off) dan metode termodulasi (on-off-on-off).

2. Menerapkan dan mengembangkan eksperimen sistem pengereman dengan menggunakan software LabVIEW agar dapat di akses melalui teknologi komputer untuk melakukan pengaturan pada modul pengereman sehingga dapat dianalisa berdasarkan ilmu yang diperoleh selama kuliah.

I.3 PEMBATASAN MASALAH

Dalam pembuatan Tugas Akhir ini terdiri dari proses pembuatan Hardware dan software. Pada pembuatan hardware meliputi sistem mekanik pada system pengereman. Sedangkan pada pembuatan software meliputi program yang digunakan untuk mempermudah user dalam menjalankan mekanik yang di kontrol dari komputer. Dan penulis akan membahas mengenai cara kerja program dalam mengatur modul sistem pengereman.

I.4 METODOLOGI PENELITIAN

Metode penelitian yang akan dilakukan terdiri dari beberapa tahap antara lain:

1. Studi Literatur

Studi literatur digunakan penulis untuk memperoleh teori-teori dasar sebagai sumber penulisan Tugas Akhir. Informasi dan pustaka yang berkaitan dengan masalah ini diperoleh dari literatur, penjelasan yang diberikan dosen pembimbing, rekan-rekan mahasiswa, internet, dan buku-buku yang berhubungan dengan Tugas Akhir penulis.

2. Perancangan dan pembuatan Program

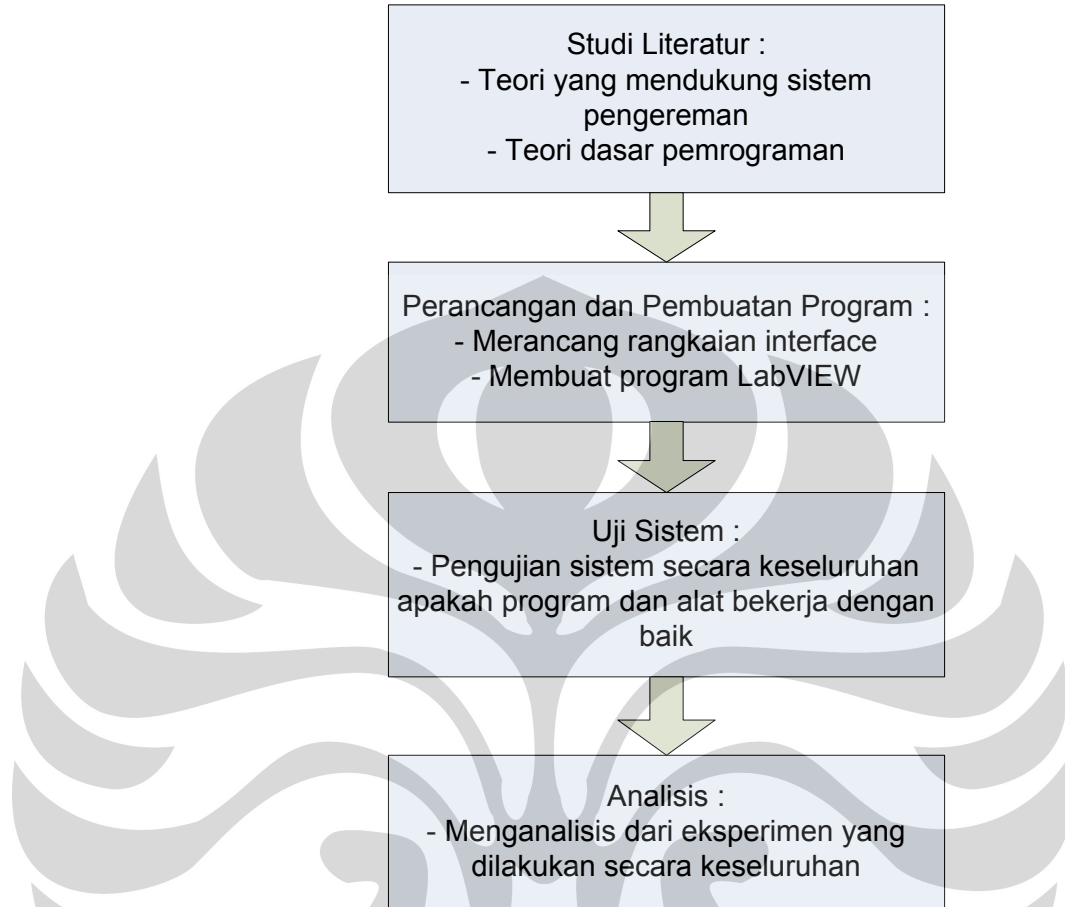
Perancangan alat merupakan tahap dimana penulis membuat rangkaian utama dari tugas akhir ini, khususnya rangkaian interface untuk mendukung komunikasi parallel yang dihubungkan ke program pada komputer.

3. Uji Sistem

Uji sistem ini berkaitan dengan pengujian alat dan program pada saat yang bersamaan serta pengambilan data.

4. Metode Analisis

Metode ini merupakan pengamatan terhadap data yang diperoleh dari pengujian alat serta pengambilan data. Setelah itu dilakukan penganalisaan sehingga dapat ditarik kesimpulan dan saran-saran untuk pengembangan lebih lanjut. Berikut ini adalah diagram langkah-langkah yang akan dilakukan dalam penelitian ini:



Gambar 1.1 Skema Penelitian

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan Tugas Akhir ini terdiri dari bab-bab yang memuat beberapa sub-bab. Untuk memudahkan pembacaan dan pemahaman maka Tugas Akhir ini dibagi menjadi beberapa bab yaitu :

BAB I Pendahuluan

Pendahuluan berisi latar belakang permasalahan, tujuan penulisan, pembatasan masalah, metode penulisan dan sistematika penulisan dari Tugas Akhir ini.

BAB II Teori Dasar

Berisi teori yang mendasari penelitian, yaitu prinsip dasar kerja program sebagai pengendali utama modul pengereman. Dan cara kerja modul pengereman.

BAB III Perancangan dan Cara Kerja Program Sistem Pengereman

Berisi tentang penjelasan perancangan program atau perangkat lunak (*software*) dan cara kerja dari program yang telah dibuat.

BAB IV Pengujian Program dan Analisa

Berisi tentang hasil dan analisa yang telah diperoleh dalam perancangan dan pengujian terhadap program baik kesalahan maupun kendala yang didapat.

BAB V Penutup

Berisi kesimpulan dari keseluruhan perancangan sistem hingga hasil penelitian yang didapat dan saran yang mungkin dapat digunakan untuk memperbaiki, menambahkan, ataupun memodifikasi alat yang sudah ada menjadi lebih baik lagi.

BAB II

TEORI DASAR

Pada Bab II akan dibahas mengenai Teori dasar yang akan dipakai dalam pembuatan Tugas Akhir. Teori dasar yang akan dipakai yaitu gaya gesek, motor dc, rem, sensor shaft encoder, gear, gaya gerak melingkar, momen inersia, serta program LabVIEW yang digunakan pada Personal Computer.

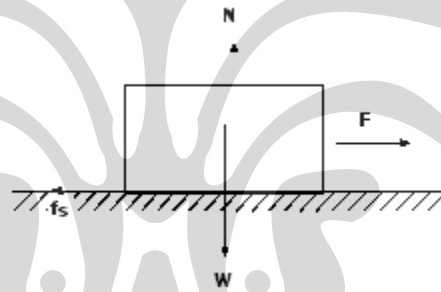
2.1 GAYA GESEK

Gaya gesekan adalah gaya yang ditimbulkan apabila dua permukaan benda saling bersentuhan. Hal ini terjadi karena adanya kekasaran dari permukaan benda yang bersentuhan. Gaya gesekan ditentukan oleh dua faktor yaitu massa benda dan koefisien gesekan benda. Pada kendaraan bermotor, gesekan antara ban dan jalan amat dibutuhkan bahkan gaya gesekan yang sebesar-besarnya antara ban dan jalan. Hal ini dimaksudkan agar ban tidak mengalami slip, supaya pengemudi dan penumpangnya merasa aman. Tetapi jika gesekan antara ban dan jalan nilainya kecil, umumnya disebabkan ban yang gundul (aus), maka hal ini dapat membahayakan pengemudi dan penumpangnya. Gaya gesekan dapat dibedakan menjadi dua yaitu gaya gesekan statik dan gaya gesek kinetik.

2.1.1 GAYA GESEKAN STATIK

Gaya gesekan statik yaitu gesekan yang bekerja pada benda tidak bergerak (diam). Gaya gesek statik pada kendaraan bermotor terjadi pada saat roda bergerak mengikuti kecepatan mobil, seolah-olah jalan dan roda diam. Pada gaya ini roda seolah-olah menggelinding terhadap jalan, maksudnya kecepatan roda dan kecepatan jalan sama, sehingga gaya gesek yang terjadi kecil.

Gaya gesekan statik dapat dirumuskan sebagai berikut:



Gambar 2-1. Gaya Gesekan Statik

$$f_s = \mu_s \cdot N \quad (2-1)$$

Karena $N = m \cdot g$, maka gaya gesekan statik dapat diuraikan menjadi :

$$f_s = \mu_s \cdot mg \quad (2-2)$$

Keterangan :

f_s = gaya gesekan statik (N)

μ_s = koefisien gesekan statik

N = gaya normal (N)

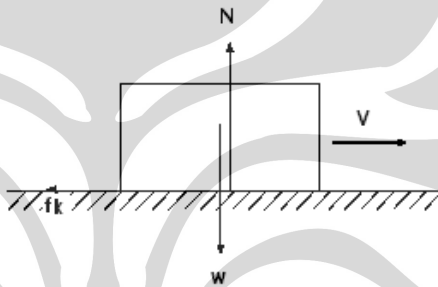
m = massa benda (kg)

g = percepatan gravitasi bumi (ms^{-2})

$g = 10 \text{ ms}^{-2}$ atau $9,8 \cdot \text{ms}^{-2}$

2.2.2 GAYA GESEK KINETIK

Gaya gesekan kinetik yaitu gesekan yang bekerja pada benda bergerak. Gaya gesekan statik dapat dirumuskan sebagai berikut.



Gambar 2-2. Gaya Gesekan Kinetik

$$f_k = \mu_k \cdot N \quad (2-3)$$

Karena $N = m \cdot g$, maka gaya gesekan kinetik dapat diuraikan menjadi :

$$f_k = \mu_k \cdot mg \quad (2-4)$$

Keterangan:

f_k = gaya gesekan kinetik (N)

N = gaya normal (N)

μ_k = koefisien gesekan kinetik

m = massa benda (kg)

g = percepatan gravitasi bumi (ms^{-2})

$g = 10 \text{ ms}^{-2}$ atau $9,8 \text{ ms}^{-2}$

Keuntungan akibat gaya gesekan dalam kehidupan sehari-hari adalah rem pada kendaraan. Prinsip kerja dari rem yaitu menahan atau menghentikan lajunya kendaraan. Sehingga kendaraan dapat mengurangi lajunya bahkan berhenti pada tempat yang diinginkan. Bila seorang pengendara mobil misalnya, menginjak pedal rem. Maka pada saat yang bersamaan kampas rem bergesekan dengan roda untuk menahan atau menghentikan gerak rotasi (putaran) roda. Gaya gesekan yang timbul antara kampas rem dan roda sangat penting

Dari prinsip kerja rem yang telah dijelaskan diatas dapat disimpulkan bahwa pada peristiwa pengereman terjadi Gaya Gesekan Kinetik dimana gesekan bekerja pada benda bergerak . Hukum gaya gesekan yaitu gaya yang timbul antara dua permukaan yang saling bersinggungan dan arahnya selalu berlawanan dengan kecenderungan arah gaya gerak benda. Pada modul pengereman ini hukum gaya gesekan yang digunakan adalah Gaya Gesekan Kinetik (f_k) seperti yang telah dijelaskan pada persamaan (2-3).

Pada tugas akhir ini, jika roda di rem, roda seakan-akan tergelincir. Maksudnya terdapat perbedaan kecepatan antara roda terhadap jalan, sehingga mengakibatkan gaya gesek kinetik yang besar.

2.2 MOTOR DC

Komponen utama yang diperlukan pada sistem pengereman adalah *actuator*. *Actuator* adalah komponen pertama untuk melakukan gerakan, mengubah energi elektrik menjadi gerakan mekanik. Jenis *actuator* yang digunakan yaitu motor listrik. Motor listrik dikelompokkan menjadi motor DC dan motor AC. Pada pembuatan tugas akhir ini menggunakan motor DC.

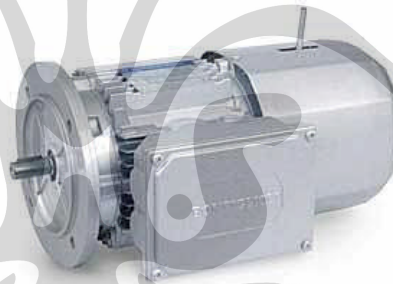
2.2.1 Prinsip Kerja Motor DC

Motor DC adalah motor yang memerlukan suplai tegangan searah pada kumparan jangkar dan kumparan medan untuk diubah menjadi energi mekanik. Pada motor DC, kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konverter energi baik energi listrik menjadi energi mekanik (motor) maupun sebaliknya dari energi mekanik menjadi energi listrik (generator) berlangsung melalui medium medan magnet. Energi yang akan diubah dari suatu sistem ke sistem yang lain, sementara akan tersimpan pada medium medan magnet untuk kemudian dilepaskan menjadi energi system lainnya. Dengan demikian, medan magnet disini selain berfungsi sebagai tempat penyimpanan energi juga sekaligus proses perubahan energi, dimana proses perubahan energi pada motor arus searah dapat digambarkan pada gambar 2.3.



Gambar 2-3. Proses Konversi Energi Pada Motor DC

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan *torque* yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.



Gambar 2-4. Motor DC

Gambar 2.4 memperlihatkan sebuah motor DC yang memiliki lima komponen utama:

- **Rotor**, yaitu bagian yang berputar pada motor berupa kumparan kawat.
- **Stator**, yaitu bagian pada motor berupa magnet.

- **Commutator**, Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.
- **Sikat**, yaitu sepasang batang garfit yang menempel pada *comutator* tetapi tidak berputar
- **Kutub medan**, Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
- **Dinamo**, Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.

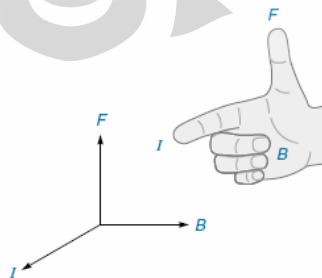
Motor DC memiliki prinsip kerja yaitu suatu penghantar yang berarus listrik dan ditempatkan dalam suatu medan magnet maka penghantar tersebut akan mengalami gaya. Prinsip kerja motor membutuhkan :

1. Adanya garis-garis gaya medan magnet (fluks), antara kutub yang berada di stator.
2. Penghantar yang berarus listrik yang ditempatkan dalam medan magnet tersebut.
3. Pada penghantar akan timbul gaya.

Gaya yang dihasilkan motor dc tergantung pada :

- a. Kekuatan pada medan magnet.
- b. Besarnya arus yang mengalir pada penghantar.
- c. Panjang kawat penghantar yang berada dalam medan magnet.

Apabila panjang kumparan rotor L dialiri arus listrik sebesar I dan terletak diantara kutub magnet utara dan selatan dengan kerapatan fluks sebesar B , maka kumparan rotor tersebut mendapat gaya F sebesar :



Gambar 2-5. Gaya Medan Magnet

$$F = B \cdot I \cdot L$$

(2-5)

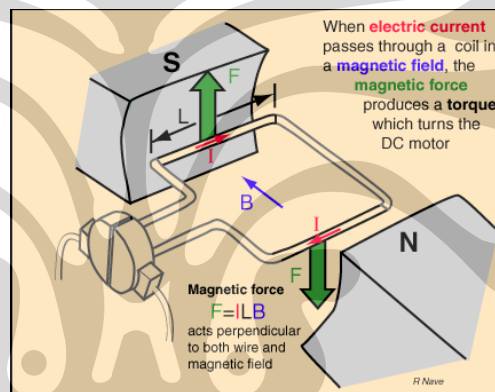
Keterangan :

F = Gaya Lorentz (Newton)

B = Kerapatan Fliks Magnet (Weber / m²)

I = Arus Listrik (Ampere)

L = Panjang sisi kumparan rotor (m)



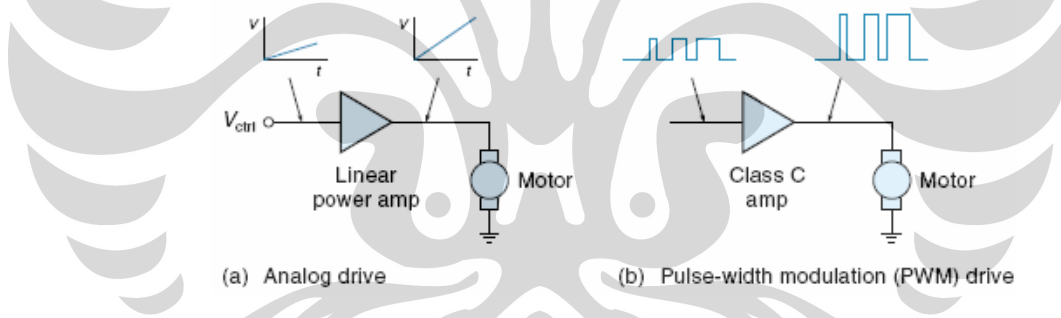
Gambar 2-6. Prinsip Motor DC

2.1.2 Cara Pengendalian Motor DC

Terdapat 2 cara untuk mengontrol kecepatan dari motor dc. Kontrol kecepatan merupakan sesuatu yang tidak akurasi dikarenakan motor yang dari energi listrik diubah menjadi torsi dan tidak diubah menjadi kecepatan. Kecepatan yang presisi ditentukan oleh torsi motor dan beban mekanik. Untuk mengendalikan motor diperlukan rangkaian interface yang dapat mengubah sinyal motor level

rendah dari controller menjadi sebuah sinyal yang cukup besar untuk menggerakkan motor. Dan cara ini disebut analog drive. Pada metode ini, sebuah penguatan power amplifier adalah linier dari controller dan tegangan analog diberikan ke motor.

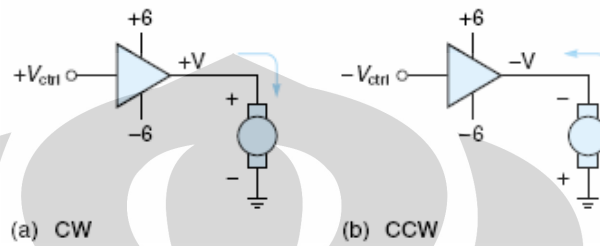
Teknik yang lain untuk mengontrol sebuah motor dc adalah pulse-width modulation (PWM). Dalam sistem ini, power di suplai ke motor dalam bentuk pulsa dc pada tegangan tertentu. Lebar pulsa bervariasi untuk mengontrol kecepatan motor. Pulsa yang lebih lebar, tegangan dc rata-rata yang lebih tinggi diperbolehkan untuk motor. Frekuensi dari pulsa yang cukup tinggi dapat menginduktansi rata-rata motor sehingga dapat menggerakkan motor secara baik.



Gambar 2-7. Cara Pengendalian Motor

Untuk mengubah arah rotasi dari PM motor, polaritas dari tegangan yang digunakan adalah berlawanan. Satu cara untuk dapat melakukannya sehingga sebuah motor driver mampu mengeluarkan tegangan positif dan tegangan negatif. Ketika tegangannya positif dan ground maka motor akan bergerak searah jarum jam (CW). Ketika tegangannya negatif dan ground maka polaritas tegangan pada

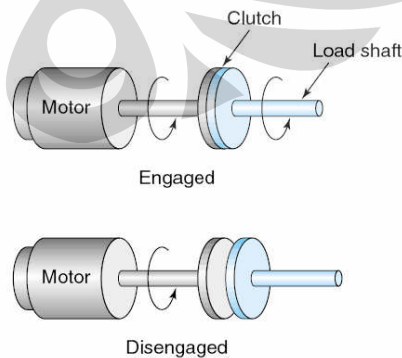
terminal motor berlawanan sehingga motor bergerak berlawanan arah jarum jam (CCW).



Gambar 2-8. Arah Putaran Motor DC

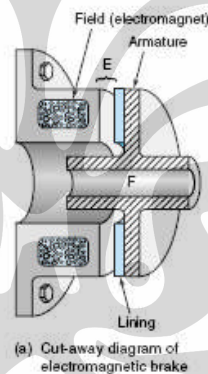
2.3 Rem

Pada mekanik, rem yang dipakai yaitu jenis rem *Clutch*. Rem clutch terdiri dari dua kondisi piringan yang saling berlawanan. Piringan yang dihubungkan ke motor dan ada pula yang dihubungkan ke beban. Ketika kedua piringan di tekan maka clutch akan menempel atau terhubung dan akan mentransmisikan torsi, pada saat itu akan terhubung arus listrik sehingga terjadi pengereman.



Gambar 2-9 Kondisi pada clutch

Pada rem elektromagnetic, terdapat dua bagian penting, yaitu field elektromagnetik dan armatur. Seperti pada clutch, field coil adalah elektromagnet dalam rumah logam. E adalah bentuk dari disk armatur. Armatur pada dasarnya merupakan sebuah logam berat dengan sambungan dan dapat berputar bebas tanpa rem. Ketika shaft direm, maka akan menyebabkan perpindahan dan mengunci F. Ketika power di-ON-kan ke elektromagnet, field magnetik akan melewati E ke armatur dan menarik field magnetik. Sehingga dapat digunakan bersama-sama dengan field stationary dan armatur untuk memberhentikan shaft.



Gambar 2-10 komponen pada rem elektromagnetik

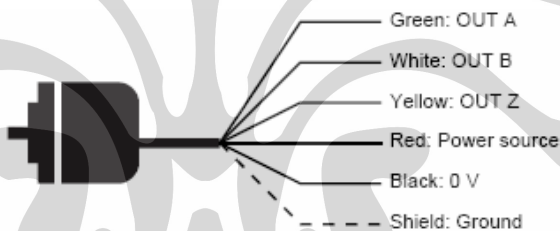
2.4 Sensor Shaf Encoder

Sensor yang dipakai dalam pengerjaan Tugas Akhir ini yaitu sensor shaf encoder. Sensor ini akan bekerja apabila motor on dan menggerakkan roda, pada roda diberikan beban, ketika beban berputar sensor akan berputar juga.

Rotasi shaft di konversi menjadi pulsa elektronik, encoder digunakan sebagai monitor elektronik untuk melihat posisi rotasi shaft. Pulsa output encoder dapat dihitung untuk menentukan kecepatan sehingga dapat dilihat tingkat akurasi dan fleksibilitas ketika di control.



Gambar 2-11 Shaft Encoder



Gambar 2-12 Diagram Pengkabelan

Tabel-1 Sifat Electric

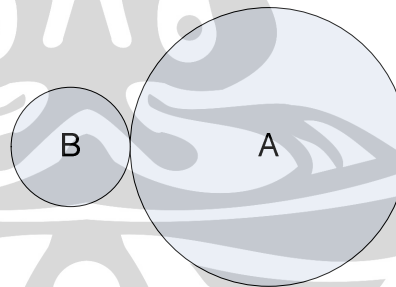
Pulse per Revolution	Up to 1024 P/R
Supply Voltage	5-30Vdc TTL & 5Vdc Line Driver
Output Format	Dual Square Wave in Quadrature with Index
Output Option	TTL Output, Line Driver
Current Requirement	TTL: 60mA Max. No Load Line Driver: 130mA Max. No Load
Frequency Response	200 KHz
Duty Ratio	50±25%
Phase Shift	25±12.5%
Index Signal Width	100±50%
Output Level	(H) ≥2.5V (L) ≤0.5V
Rise and Fall Time	≤1µs (When cable length is 1m)
Insulation Resistance	≥50MΩ

Tabel-2 Sifat Mekanik

Shaft Diameter	8 mm (0.315")
Shaft Loading	Axial: 60N (14 lbs) Radial: 100N (23 lbs)
Shaft Scraper	Standard (Detachable, Repairable)
Starting Torque	1.4 oz. in (0.003N•m) at 20°C
Shaft Angular Acceleration	1000 rad/Sec ²
Moment of Inertia	2.0x10 ³ Kg • m ²
Mech. Perm. Speed	6000 rpm Max.
Weight	0.5 lb

2.5 Roda Gigi (Gear)

Pada pembuatan mekanik dibutuhkan roda gigi. Roda gigi merupakan komponen transmisi mekanik, yang dasarnya adalah dua buah silinder yang menggelinding satu pada yang lainnya tanpa gesekan (slip). Kecepatan tangensialnya sama, tetapi kecepatan sudutnya bisa tidak sama.



Gambar 2-13. Roda Gesek

Jika roda A diputar, maka roda A akan menggerakkan roda B dalam arah yang berlawanan seperti pada gambar 2-13. Berputarnya roda B dikarenakan adanya gaya tangensial yang dipindahkan ke roda A dengan tidak melebihi

tahanan gesek antara kedua roda. Apabila gaya tangensial lebih besar dari tahanan gesek, akan terjadi slip. Alat yang menggunakan cara kerja seperti ini untuk memindahkan daya disebut roda gesek. Cara ini cukup efektif baik untuk meneruskan daya kecil dengan putaran yang tidak terlalu tinggi.

Untuk mentransmisikan daya besar dan rpm yang tepat, tidak dapat dilakukan dengan roda gesek. Untuk itu kedua roda tersebut harus dibuat bergigi pada sekelilingnya, sehingga penerusan daya dilakukan oleh gigi-gigi kedua roda yang saling terkait. Roda semacam ini, yang bias berbentuk silinder dan kerucut disebut roda gigi.

Selain cara transmisi di atas, ada pula cara lain untuk meneruskan daya, yaitu dengan sabuk (belt) atau rantai. Namun demikian transmisi roda gigi mempunyai beberapa keunggulan dibandingkan dengan sabuk dan rantai, karena lebih ringkas, putaran lebih besar, serta daya lebih besar.

2.5.1 Macam-Macam Roda Gigi

Gear atau yang biasa disebut roda gigi terdapat empat macam bentuk yaitu:

1. Gear lurus (spur gear)

Bentuk gigi ini lurus dan paralel dengan sumbu roda gigi

2. Gear miring (helical gear)

Bentuk gigi ini menyilang miring terhadap sumbu roda gigi

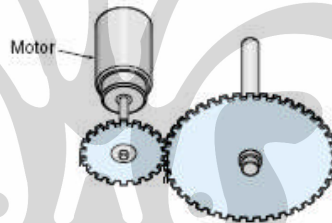
3. Gear panah (double helical / herring bone gear)

Bentuk gigi berupa panah atau miring dengan kemiringan berlawanan

4. Gear melengkung/bengkok (curved/spherical gear)

Bentuk gigi melengkung mengikuti pola tertentu (lingkaran/ellips)

Pada pembuatan Tugas Akhir ini digunakan Spur Gear, yaitu roda gigi yang berjalan pada dua bidang silinder yang saling bersinggungan, yang satu menggelinding pada yang lainnya dengan sumbu tetap sejajar. Dimana power akan dikirim oleh salah satu gear, kemudian mendorong berlawanan dengan gigi dari gear yang lain. Ketika 2 gear dengan diameter yang berbeda digabungkan, maka rotasi keduanya berada pada kecepatan yang berbeda.

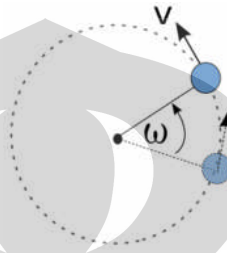


Gambar 2-14. Gear

2.6 Gerak Melingkar

Gerak Melingkar adalah gerak suatu benda yang membentuk lintasan berupa lingkaran mengelilingi suatu titik tetap. Agar suatu benda dapat bergerak melingkar ia membutuhkan adanya gaya yang selalu *membelokkan*-nya menuju pusat lintasan lingkaran. Gaya ini dinamakan gaya sentripetal. Suatu gerak melingkar beraturan dapat dikatakan sebagai suatu gerak dipercepat beraturan,

mengingat perlu adanya suatu percepatan yang besarnya tetap dengan arah yang berubah, yang selalu mengubah arah gerak benda agar menempuh lintasan berbentuk lingkaran.



Gambar-2.15 Gaya gerak melingkar

Besaran-besaran yang mendeskripsikan suatu gerak melingkar adalah θ , ω dan α atau berturut-turut berarti sudut, kecepatan sudut dan percepatan sudut. Besaran-besaran ini bila dianalogikan dengan gerak linier setara dengan posisi, kecepatan dan percepatan atau dilambangkan berturut-turut dengan r , dan a .

Besaran gerak lurus dan melingkar			
Gerak lurus		Gerak melingkar	
Besaran	Satuan (SI)	Besaran	Satuan (SI)
posisi r	<u>m</u>	sudut θ	<u>rad</u>
kecepatan v	<u>m/s</u>	kecepatan sudut ω	<u>rad/s</u>
percepatan a	<u>m/s²</u>	percepatan sudut	<u>rad/s²</u>

		α	
-	-	periode T	<u>s</u>
-	-	radius R	<u>m</u>

Tabel-3 Besaran gerak lurus melingkar

Gerak melingkar dapat dibedakan menjadi dua jenis, atas keseragaman kecepatan sudutnya ω , yaitu:

- gerak melingkar beraturan, dan
- gerak melingkar berubah beraturan.

2.8.1 Gerak melingkar beraturan

Gerak Melingkar Beraturan (GMB) adalah gerak melingkar dengan besar kecepatan sudut ω tetap. Besar Kecepatan sudut diperoleh dengan membagi kecepatan tangensial v_T dengan jari-jari lintasan R

$$\omega = \frac{v_T}{R} \quad (2-6)$$

Arah kecepatan linier v dalam GMB selalu menyinggung lintasan, yang berarti arahnya sama dengan arah kecepatan tangensial v_T . Tetapnya nilai kecepatan v_T

akibat konsekuensi dari tetapnya nilai ω . Selain itu terdapat pula percepatan radial a_R yang besarnya tetap dengan arah yang berubah. Percepatan ini disebut sebagai percepatan sentripetal, di mana arahnya selalu menunjuk ke pusat lingkaran.

2.8.2 Gerak melingkar berubah beraturan

Gerak Melingkar Berubah Beraturan (GMBB) adalah gerak melingkar dengan percepatan sudut α tetap. Dalam gerak ini terdapat percepatan tangensial a_T (yang dalam hal ini sama dengan percepatan linier) yang menyinggung lintasan lingkaran (berhimpit dengan arah kecepatan tangensial v_T).

$$\alpha = \frac{a_T}{R} \quad (2-7)$$

2.8.3 Gerak berubah beraturan

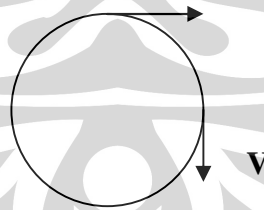
Gerak melingkar dapat dipandang sebagai gerak berubah beraturan. Bedakan dengan gerak lurus berubah beraturan (GLBB). Konsep kecepatan yang berubah kadang hanya dipahami dalam perubahan besarnya, dalam gerak melingkar beraturan (GMB) besarnya kecepatan adalah tetap, akan tetapi

arahnya yang berubah dengan beraturan, bandingkan dengan GLBB yang arahnya tetap akan tetapi besarnya kecepatan yang berubah beraturan.

Tabel-4 Perbedaan GLBB dan GMB

Gerak berubah beraturan		
Kecepatan	GLBB	GMB
Besar	berubah	Tetap
Arah	tetap	berubah

Silinder dan roda yang digunakan dalam mekanik modul pengereman ini terumus dalam hukum gaya melingkar yaitu:



Gambar 2-16. Gaya Melingkar

Adapun Ciri-cirinya adalah :

- Besar kecepatan sudut adalah konstan
- Besar kecepatan Linier (laju) konstan, tetapi arahnya berubah –

kecepatan V berubah.

Kecepatan linear :

$$V = \frac{2\Pi R}{T} \text{ m/s} \quad \text{atau} \quad V = \omega \cdot R \quad (2-8)$$

Kecepatan sudut:

$$\omega = \frac{2\Pi}{T} \quad \text{atau} \quad \omega = 2\Pi f \text{ rad/s} \quad (2-9)$$

Karena silinder dan roda saling terhubung maka : $\omega_1 \cdot R_1 = \omega_2 \cdot R_2$ (2-10)

Keterangan :

V = Kecepatan linier (m/s)

ω = kecepatan sudut (rad/s)

R = Panjang jari-jari roda (m)

T = Periode

f = frekuensi (Hz)

2.7 Momen Inersia

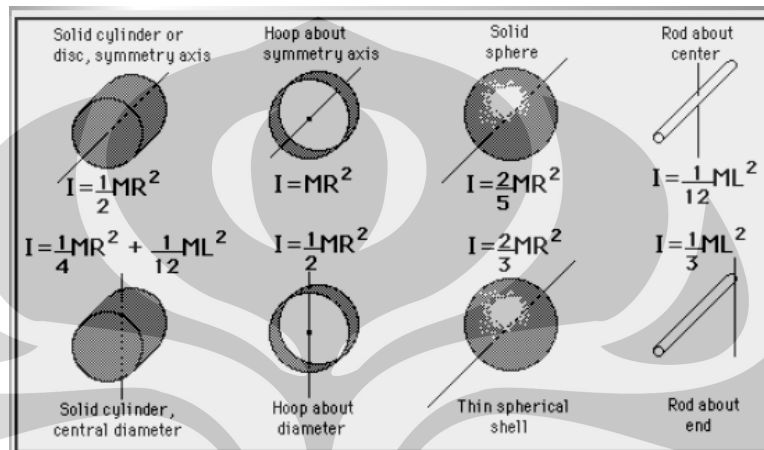
Momen inersia adalah kecenderungan benda untuk melakukan gerak rotasi terhadap sumbu. Dalam gerak rotasi setiap benda mempunyai kecenderungan untuk mempertahankan posisi atau keadaannya. Kecenderungan ini dinamakan inersia rotasi, sedangkan untuk menyatakan besar kecenderungan dinamakan momen inersia. Momen inersia tergantung pada bentuk benda, massa (m) dan letak sumbu putar (r) dan dinotasikan dengan I , satuannya $\text{kg} \cdot \text{m}^2$

Momen inersia dapat dibedakan 3 jenis dan Perumusan momen inersia :

- Benda titik = $m \cdot r^2$ (2.11)

- Kelompok benda titik = $\sum m_i \cdot r_i^2$ (2.12)

- Benda kontinyu = $r \int m_0^2 \cdot dm$ (2.13)



Gambar 2-17. Momen inersia benda-benda uniform dengan berbagai bentuk

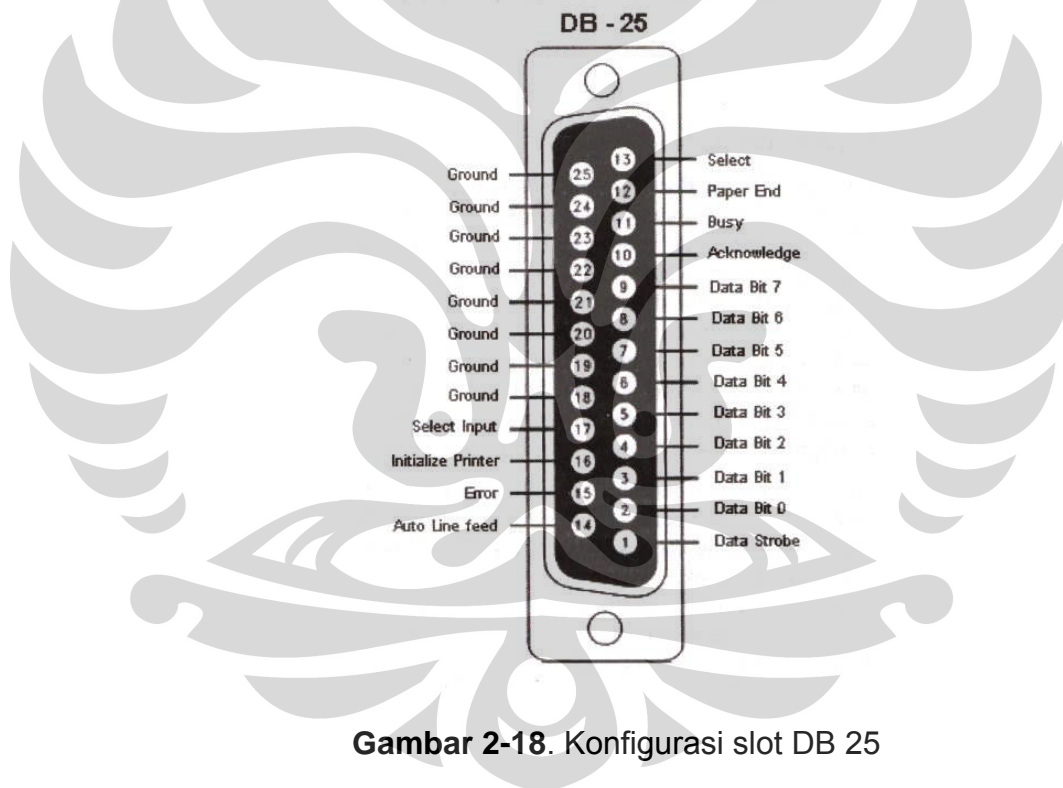
Pada mekanik, diberikan beban pada roda sebesar 5 kilogram, hal tersebut dimaksudkan agar roda memiliki momen inersia. Ketika beban mendekati as atau poros, yang terjadi yaitu roda berputar kencang seperti tidak ada beban. Maka dengan adanya hal tersebut, beban diletakkan menjauh dari poros atau as-nya. Dapat dilihat, jika beban diletakkan semakin menjauh dari poros maka momen inersia yang didapat semakin besar.

2.8 Konfigurasi Port Parallel

Port Paralel atau Port Printer sebenarnya terdiri dari tiga bagian masing-masing diberi nama sesuai dengan tugasnya dalam melaksanakan pencetakan

pada printer. Tiga bagian tersebut adalah *Data Port (DP)*, *Printer Control (PC)*, dan *Printer Status (PS)*. DP digunakan untuk mengirim data yang harus dicetak oleh printer, PC digunakan untuk mengirimkan kode-kode kontrol dari computer ke printer, dan PS digunakan untuk mengirimkan kode-kode status printer ke computer. Paralel port dapat mengirimkan 8 bit data sekaligus dalam satu waktu. Paralel port ini menggunakan konektor DB25.

DB25 memiliki 25 buah pin dengan gambaran sebagai berikut :



Gambar 2-18. Konfigurasi slot DB 25

DP, PC, dan PS sebenarnya adalah port-port 8 bit, namun hanya DP yang benar-benar 8 bit. Untuk PC dan PS, hanya beberapa bit saja yang dipakai yang berarti hanya beberapa bit saja dari port-port ini yang dapat dimanfaatkan untuk

keperluan interfacing. Port PC adalah port baca/tulis (*read/write*), PS adalah port baca saja (*read only*), sedangkan Port DP port baca/tulis juga. Akan tetapi, kemampuan ini hanya dimiliki oleh Enhanced Parallel Port (EPP), sedangkan port parallel standard hanya memiliki kemampuan tulis saja. Pada EPP, pengaturan arah jalur data DP dilakukan lewat bit 5 PC. Jika bit 5 PC bernilai 0, maka jalur data dwi-arah DP menjadi output dari port parallel, sebaliknya jika bit 5 PC bernilai 1, maka jalur data dwi-arah DP menjadi input dari port parallel. Selengkapnya, dibawah ini merupakan konfigurasi slot DB-25 female yang terdapat pada belakang computer dapat dilihat pada gambar 2.12.

Konfigurasi dari DP, PC, PS dapat dilihat pada tabel 1. Pin-pin dengan keterangan komplemen akan berlogika tinggi pada keadaan awal.

Tabel 5. Konfigurasi pin dan nama sinyal konektor parallel standard DB-25

Nomor Pin	Nama Sinyal	Direction	Register	Complemen
1	Strobe	In/Out	Control Bit 0	Ya
2	Data 0	out	Data Bit 0	
3	Data 1	out	Data Bit 1	
4	Data 2	out	Data Bit 2	
5	Data 3	out	Data Bit 3	
6	Data 4	out	Data Bit 4	
7	Data 5	out	Data Bit 5	
8	Data 6	out	Data Bit 6	

9	Data 7	out	Data Bit 7	
10	Ack	in	Status bit 6	
11	Busy	in	Status bit 7	Ya
12	Paper-Out/Paper	in	Status bit 5	
13	Select	in	Status bit 4	
14	Auto-Line Feed	In/Out	Contol Bit 1	Ya
15	Error/ Fault	in	Status bit 5	
16	Initialize	In/Out	Contol Bit 2	
17	Select	In/Out	Contol Bit 3	Ya
18-25	Ground	Gnd		

Untuk dapat menggunakan port parallel, kita harus mengetahui alamatnya. *Base address* LPT1 biasanya adalah 888 (378h) dan LPT2 biasanya 632 (278h). Alamat tersebut adalah alamat yang umumnya digunakan, tergantung dari jenis komputer.

Setelah kita mengetahui alamat dari port parallel, maka kita dapat menentukan alamat DP, PC, dan PS. Alamat DP adalah *base address* dari port parallel tersebut, alamat PS adalah *base address* + 1, dan alamat PC adalah *base address* +2. Tabel 2 adalah tabel alamat masing-masing port yang umumnya digunakan.

Tabel 6. Pengaksesan Port Paralel Pada LabVIEW

Nama Port	Alamat Register
LPT1 DP	378h / 888
LPT1 PS	379 / 889
LPT1 PC	37Ah / 890

Tentunya modul pengereman ini tidak dikendalikan secara manual melainkan sudah terkomputerisasi. Seperti disebutkan di awal tadi, perlu sebuah program yang mampu mengendalikan modul pengereman secara digital. Khusus dalam penelitian ini diperlukan sebuah komputer yang mampu mengatur modul pengereman. Pada komputer *PC* yang digunakan untuk simulasi pada modul ini tidak secara khusus terdapat *software* atau program yang mampu melakukan pengaturan modul ini. Maka perlu dibuat sebuah program atau *software* yang khusus dirancang untuk keperluan pengaturan modul pengereman ini.

Adapun pembuatan *software* ini adalah menggunakan *LabVIEW*, sebuah *software* pembuat program pada komputer *PC*.

2.8 Pengenalan LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) adalah sebuah perangkat lunak (*software*) yang dapat dihubungkan dengan perangkat keras (*hardware*) yang terhubung dengan PC (Personal Computer) dimana user ketika berinteraksi dengan komputer sama dengan menjalankan instrument elektronik [1]. LabVIEW merupakan sebuah pengembangan aplikasi program yang berbeda dengan bahasa pemrograman lain yang seperti Assembler, C, C++, dll. LabVIEW dalam merancang sebuah program menggunakan bahasa pemrograman grafik seperti halnya flowchart yang disebut dengan *block diagram*, yang mengurangi atau menghilangkan cukup banyak sintaks. Hal ini sangat berbeda dengan pemrograman yang lain dimana membutuhkan banyak perintah atau

sintaks untuk membuat sebuah program. Selain itu bahasa pemrograman lainnya juga dapat dimasukkan ke dalam LabVIEW yang dapat disinkronisasi oleh LabVIEW itu sendiri.

Program LabVIEW disebut juga Virtual Instrumentation (VI) karena tampilan dan cara pengoperasian program yang dihasilkan mirip dengan instrument sesungguhnya [2]. LabVIEW mempunyai daftar perpustakaan (library) contoh program aplikasi seperti data akuisisi, GPIB (General Purpose Interface Bus) dan kontrol instrument serial, analisis data, presentasi data, dan penyimpanan data. Selain mempunyai library, LabVIEW juga mempunyai berbagai macam contoh yang terdapat dalam LabVIEW itu sendiri misalnya konversi temperature, menghitung volume suatu bangunan, membuat database,dll. Dalam menjalankan program dapat juga user melihat aliran data yang terjadi ketika program sedang berjalan yang dapat dilihat pada bagian block diagram.

Pada LabVIEW terdapat beberapa bagian terpenting yaitu:

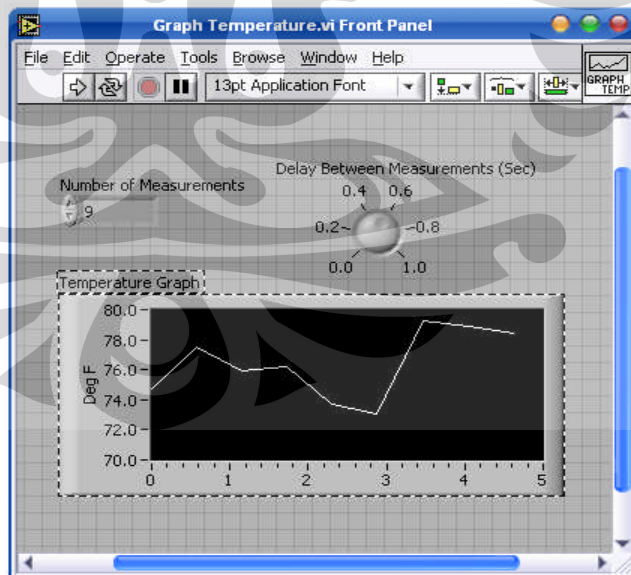
1. Toolbar
2. Front Panel
3. Block Diagram
4. Icon/Connector
5. Hirarki
6. Palette

i. Toolbar

Toolbar merupakan tombol yang berfungsi menjalankan dan mengatur program yang akan dijalankan sampai menghentikan jalannya program.

ii. Front Panel

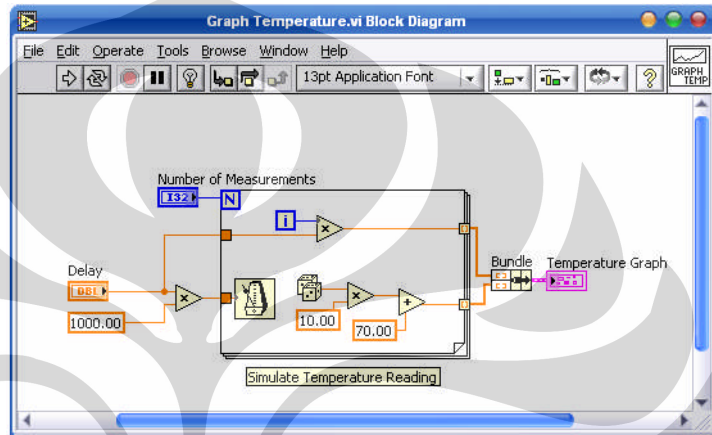
Front Panel merupakan bagian dimana user berinteraksi dengan VI ketika program dijalankan. Selain itu pada *front panel* terdapat bagian kontrol dan indicator. Kontrol berfungsi mengatur parameter dan input yang diinginkan sedangkan indicator berfungsi sebagai output yang mempunyai tampilan baik itu data, grafik, maupun gambar. Untuk mengontrol front panel digunakan mouse dan keyboard untuk memasukkan data yang diinginkan oleh klien.



Gambar 2-18. Contoh tampilan pada front panel

iii. Block Diagram

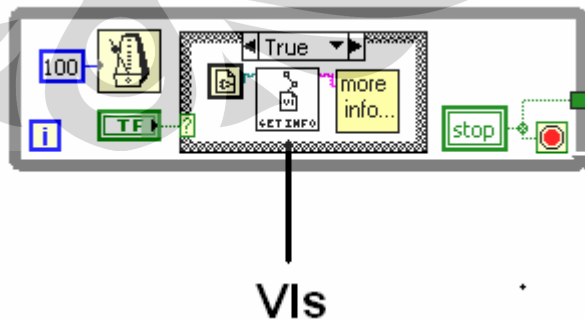
Block Diagram merupakan bagian dimana user menghubungkan komponen sehingga program dapat berjalan. Pada bagian ini user juga dapat melihat aliran program berjalan.



Gambar 2-19. Contoh tampilan pada block diagram

iv. Icon/ Connector

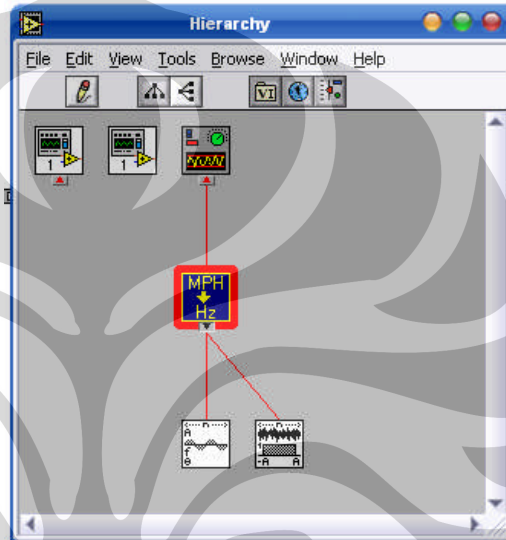
Icon/Connector merupakan bagian yang menghubungkan antar VI bila dalam suatu program digunakan lebih dari satu VI.



Gambar 2-20. Tampilan Icon/Connector

v. **Hirarki**

Hirarki merupakan struktur yang dimiliki oleh VI dimana icon yang terdapat dalam program akan diurutkan berdasarkan aliran data program yang bergerak.

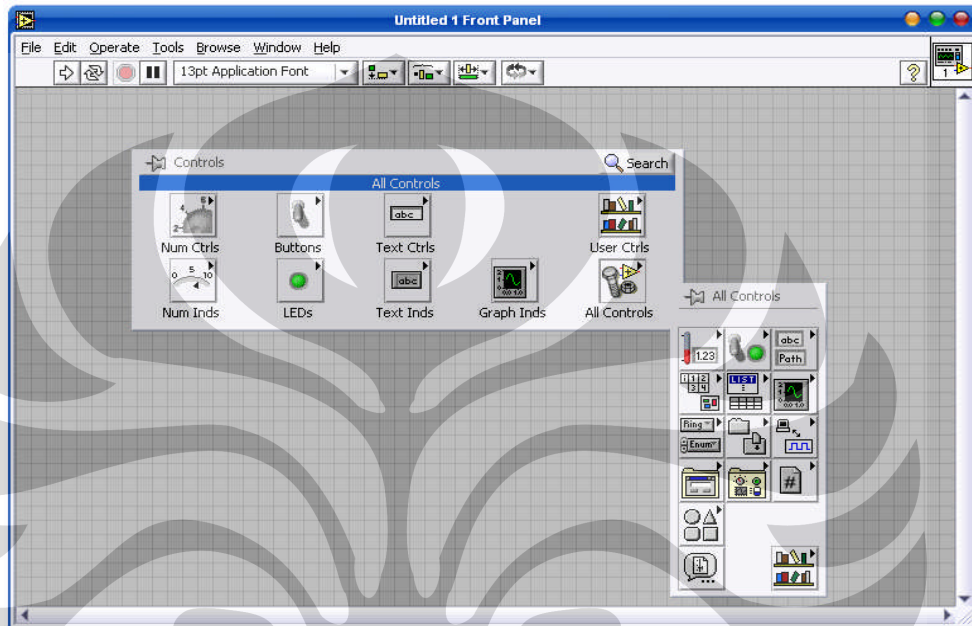


Gambar 2-21 Contoh Tampilan Hirarki dari program Simulated Frekuensi

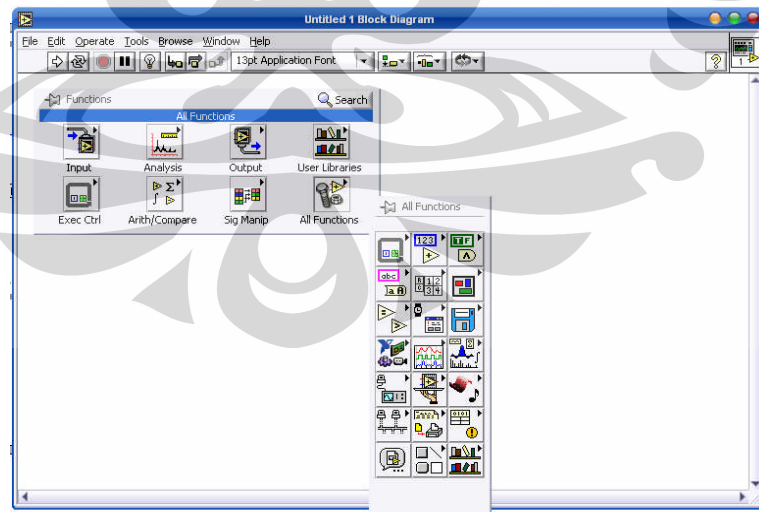
vi. **Palette**

Palette merupakan salah satu alat bantu dalam merancang sebuah VIs. *Palette* terdapat 2 buah tempat yaitu *front panel* dan block diagram dimana kegunaannya berbeda satu sama lain. Pada *front panel*, *palette* yang tersedia berfungsi sebagai kontrol dan indikator yang terdiri dari tampilan data, tampilan grafik, tombol, table, dll. Sedangkan pada block diagram, *palette* yang tersedia berfungsi sebagai fungsi atau proses pengolahan data misalnya grafik sinus, formula, analisis sinyal, dll. Sedangkan pada block diagram, *palette* yang tersedia berfungsi sebagai

fungsi atau proses pengolahan data misalnya grafik sinus, formula, analisis sinyal, dll. Berikut tampilan pallete pada front panel dan block diagram:



Gambar 2-22. Tampilan Pallete pada front panel.



Gambar 2-23. Tampilan Pallete pada Block Diagram

BAB III

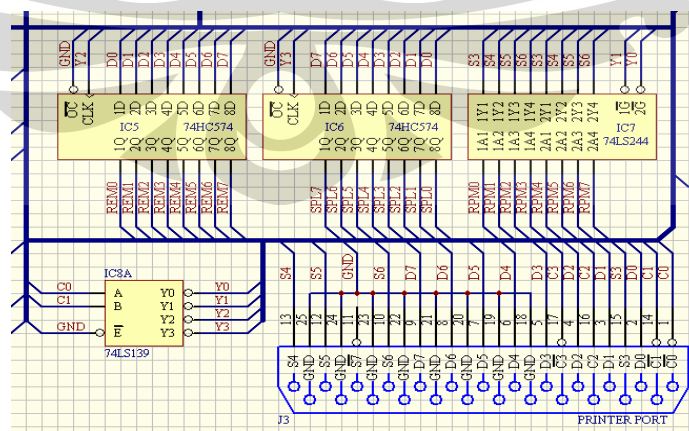
PERANCANGAN DAN CARA KERJA PROGRAM SISTEM

PENGEREMAN

Dalam perancangannya, program ini dilengkapi dengan rangkaian *interfacing*. Selain itu juga akan dibahas tentang flow chart yang dipakai dan penjelasan program labview dalam pembuatan Tugas Akhir ini.

3.1 Rangkaian *Interfacing*

Rangkaian *Interfacing* ini diperlukan untuk komunikasi data dengan hardware yang digunakan untuk mengirim data dan menerima data dari microcontroller. Adapun rangkaian *interfacing* yang digunakan adalah sebagai berikut:



Gambar 3-1 Rangkaian *interfacing*

Rangkaian *interfacing* disini berfungsi untuk menyatukan dua protocol yang berbeda, antara PC dengan microcontroller yang digunakan. Jalur komunikasi yang digunakan yaitu port parallel seperti pada table 5, bahwa port printer memiliki 8 pin yang berfungsi sebagai data output dan 4 pin yang berfungsi sebagai input yang bersifat normal. Normal yang dimaksudkan yaitu data yang dikirim dari printer port sama dengan kondisi data yang sampai ke microcontroller lalu terdisplay pada mekanik. Pada pembuatan rangkaian interfacing ini penyusun memanfaatkan 8 jalur data output dan 4 jalur input normal tersebut sebagai jalur pengiriman dan penerimaan data dari microcontroller yang digunakan.

Pada rangkaian ini terdapat dua jalur, yaitu jalur pengiriman dan penerimaan data. Untuk jalur pengiriman data, digunakan IC 74HC574. Seperti yang terlihat pada gambar skematik diatas bahwa lajur D (D0-D7) terhubung dengan port parallel. Delapan bit data yang dikirim ke microcontroller yaitu REM dan SPL (Power motor).

Jalur penerimaan data dari microcontrollercontroller, digunakan IC 74LS244. IC ini dapat membagi 8 bit data menjadi 4 bit data yang dikirim secara bertahap. Terdapat pin 1G yang berfungsi menerima 4 bit low dapat dikontrol melalui PC. Sama halnya dengan 2G yang berfungsi mengalirkan 4 bit high berikutnya.

Tabel 7. Truth table IC 74 LS139

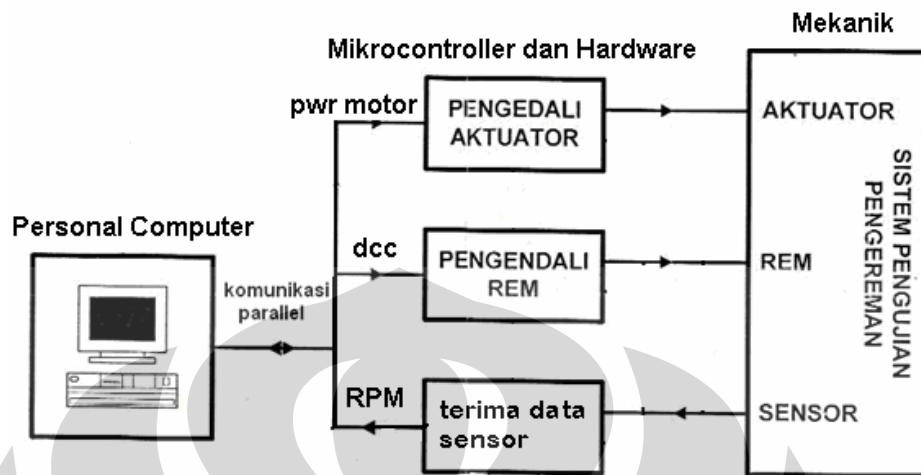
C0	C1	Y0	Y1	Y2	Y3
0	0	0	1	1	1
1	0	1	0	1	1
0	1	1	1	0	1
1	1	1	1	1	0

Pada table diatas merupakan truth table dari IC 74LS139, dimana IC ini merupakan aktif low. Printer Port mengeluarkan data Printer Control C0 dan C1 sebagai output dari printer port. Y0, Y1, Y2, Y3 merupakan output dari IC 74LS 139. Jika C0 dan C1 diberi input 0 maka Y0 aktif dan akan segera mengirim data RPM 4 bit high ke printer port.

Lalu jika C0 dan C1 diberi input 1 dan 0 maka Y1 akan aktif, akan mengirimkan data RPM 4 bit low ke printer port. Apabila C0 dan C1 diberi input 0 dan 1 maka Y2 akan aktif dan akan mengirimkan data 8 bit ke microcontroller melalui printer port. Dan jika C0 dan C1 bernilai 1 maka Y3 aktif dan akan mengirimkan data SPL (pwr motor) berupa 8 bit ke microcontroller.

3.2 Flow Chart Global Modul Sistem Pengereman

Pada pembuatan tugas akhir ini tentu saja terdapat flow chart yang digunakan sebagai patokan dalam pengerjaanya. *Flow chart* global untuk modul pengereman ini adalah seperti di bawah ini:

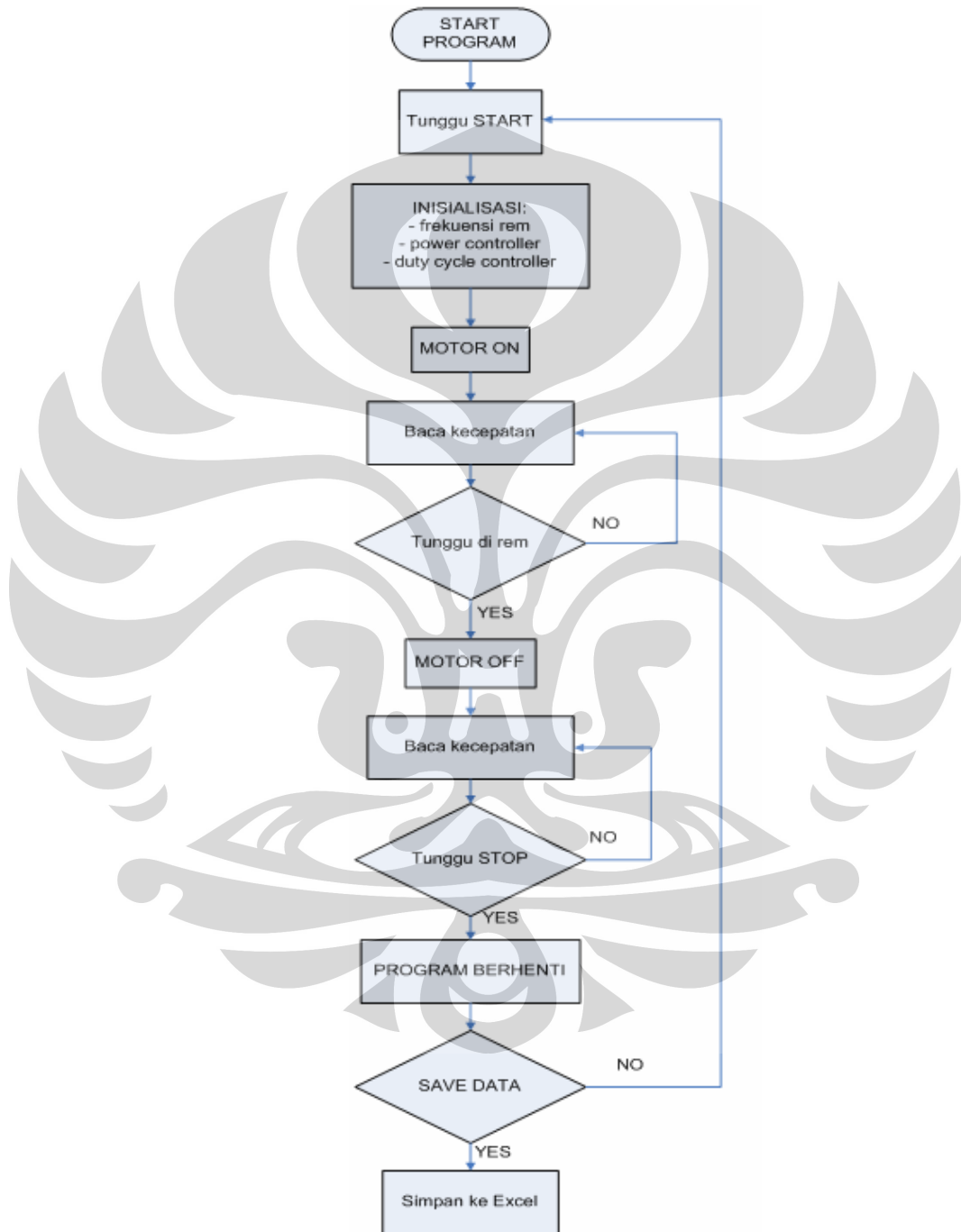


Gambar 3-2 Flow chart global sistem pengereman

Pada PC akan dikontrol pengendali actuator (Power Controller) dan pengendali rem (Duty Cycle Controller) dimana data yang diinginkan akan dikirim ke microcontroller. Dari microcontroller akan disampaikan ke mekanik, Pada mekanik terdapat sensor yang berfungsi untuk menghitung kecepatan roda yang akan disampaikan ke terima data sensor. Sehingga pada komputer akan dilihat grafik RPM yaitu (Kecepatan vs Waktu).

Dari flow chart diatas dapat dijelaskan bahwa labview mengirimkan data 8 bit ke microcontroller yaitu data power motor dan data rem atau duty cycle controller (dcc) melalui komunikasi paralel. Microcontroller mengirimkan data 16 bit berupa rpm ke labview, nilai rpm di dapat dari sensor shaf encoder yang menghitung kecepatan roda ketika roda berputar. Ketika roda berputar, shaf encoder akan ikut berputar juga.

3.2.1 Flow Chart pada LabVIEW



Gambar 3-3 Flow chart yang digunakan pada Labview

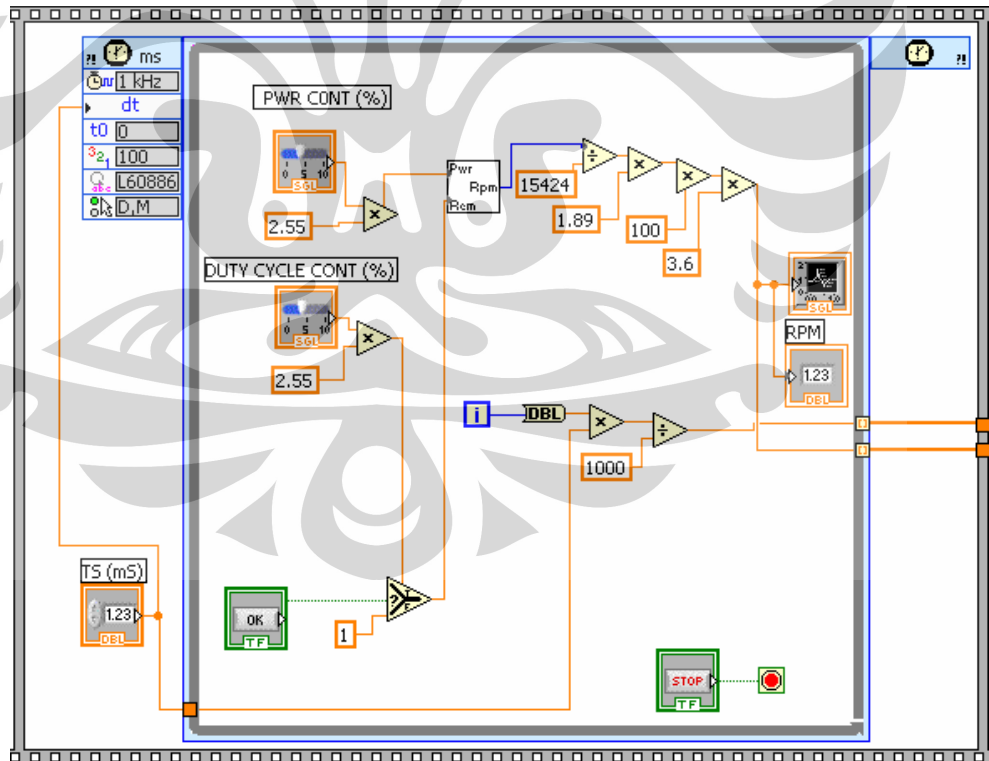
Pada saat program di start, tunggu tekan tombol START. Lalu akan menginisialisasi frekuensi rem, power controller, duty cycle controller. Adapun frekuensi rem yang disediakan yaitu 1, 2, 5,10 Hz. Menentukan power controller dan duty cycle. Setelah itu pada mekanik motor akan aktif, dan sensor akan mulai menghitung kecepatan motor. Pada LabVIEW akan dibaca kecepatan motor dan terdisplay pada grafik. Setelah mencapai kecepatan yang stabil maka tunggu untuk di rem (BRAKE). Bila tidak di rem (UNBRAKE) maka Labview akan terus membaca kecepatan motor. Jika di rem, motor akan off lalu program akan membaca kecepatan setelah di rem. Lalu apabila tombol STOP ditekan maka program akan berhenti. Jika tombol STOP tidak ditekan maka LabVIEW akan terus membaca kecepatan motor. Setelah program berhenti maka akan Save data. Bila data di save akan disimpan ke excel. Jika tidak di save

3.3 Pemrosesan Data

Data yang dibutuhkan oleh program ini adalah data *Power Controller* dan *Duty Cycle Controller*. *Power Controller* yaitu kecepatan yang diinginkan dari motor, sedangkan *Duty Cycle Controller* merupakan data yang diambil dari rem. Jika data dari rem 100% maka pengereman secara konvensional (langsung off) dan apabila data dari *Duty Cycle Controller* kurang dari 100% maka dilakukan pengereman secara termodulasi (on-off-on-off).

Ketika tombol start di tekan maka program ini dimulai, lalu akan mengirim frekuensi rem ke microcontrollercontrollercontroller. Pada frame selanjutnya

terdapat timed loop, time sampling yang menggunakan numeric control sebagai icon dihubungkan ke t pada timed loop. Time Sampling yaitu waktu yang digunakan setiap pengambilan data, adapun time sampling yang digunakan disini yaitu 10 ms. Selanjutnya Untuk mengatur Power motor dan duty cycle rem menggunakan fungsi control pointer slide. Data yang diatur pada kedua pointer slide tersebut dalam persen, lalu agar menghasilkan data 8 bit, dikalikan dengan 2,55. Pada power motor setelah dikalikan 2,55, hasil datanya akan langsung dikirim ke port parallel melalui sub vi LPT rem. Proses pengiriman data akan dijelaskan dibawah.



Gambar 3-4 Proses Pengolahan data

Pada duty cycle rem setelah dikalikan dengan 2,55, hasil datanya akan masuk ke fungsi icon *select*. Data akan masuk ke true, dimana jika brake ditekan maka data akan dikirim melalui sub vi ke port parallel. Jika tidak tekan (UNBRAKE) maka data yang akan dikirim melalui komunikasi parallel yaitu 1.

Selanjutnya data 8 bit berupa nilai rpm akan dikirim dari microcontroller. Proses penerimaan data tersebut telah dijelaskan sebelumnya pada proses penerimaan data. Nilai RPM pada sub vi yang diterima yaitu berupa bilangan desimal. Selanjutnya data yang diterima dikonversi dengan rumus agar dihasilkan nilai kecepatan (Km/ Jam) sebagai berikut:

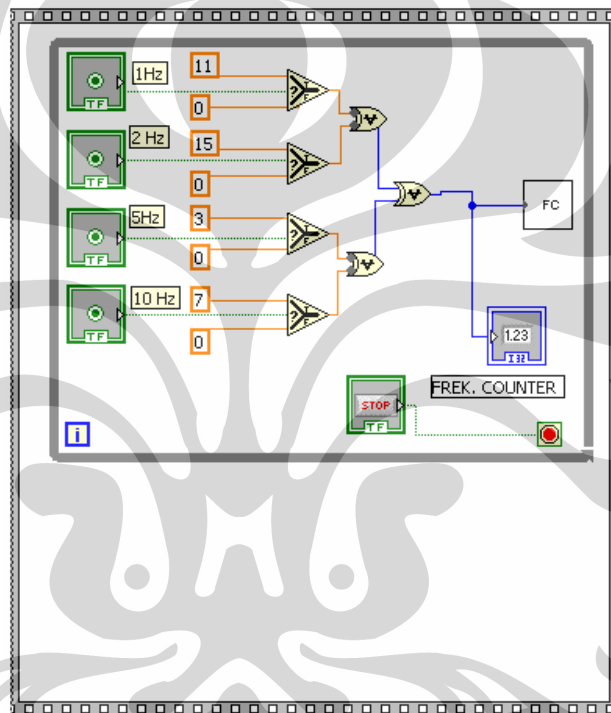
$$= \frac{\text{Nilai RPM yang dibaca}}{\text{Banyaknya pulsa sensor tiap 1 putaran}} \times \text{Keliling Roda (m)} \times \text{Time Sampling (ms)} \times 3,6 \text{ (Km/Jam)}$$

Nilai RPM yang dibaca yaitu berupa bilangan desimal dibagi dengan banyaknya pulsa sensor tiap satu putaran. Data pulsa sensor yang tiap 1 putaran didapatkan dari teman tim saya. Keliling roda yang didapat yaitu 1,89 meter. Sedangkan time sampling yang digunakan yaitu 10 ms. 10 ms dikalikan dengan 1000 untuk dikonvert menjadi Km. Sehingga dapat dikalikan langsung dengan 3,6 agar menjadi satuan Km/Jam.

Pada akhirnya setelah dikonversi akan ditampilkan pada grafik dan numerik indikator.

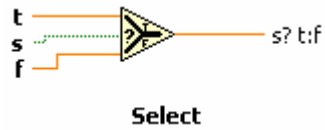
3.3.1 Menentukan Nilai Frekuensi Rem

Pada proses ini, setelah program di run, maka langsung memilih nilai frekuensi rem yang akan digunakan. Frekuensi rem yang disediakan yaitu 1Hz, 2Hz, 5Hz, dan 10 Hz.



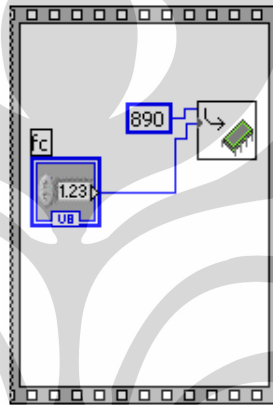
Gambar 3-5 Block Diagram Frekuensi rem

Pada block diagram diatas menggunakan fungsi *select*, dimana jika radio button 2 Hz ditekan maka terjadi kondisi true sehingga nilai 15 dikirim ke sub VI FC untuk dikirimkan ke alamat 890.



Gambar 3-6 Fungsi Select

Ketika radio button 5 Hz ditekan maka nilai 3 dikirimkan ke alamat 890 melalui sub vi FC. Dibawah ini merupakan front panel pada sub vi FC.



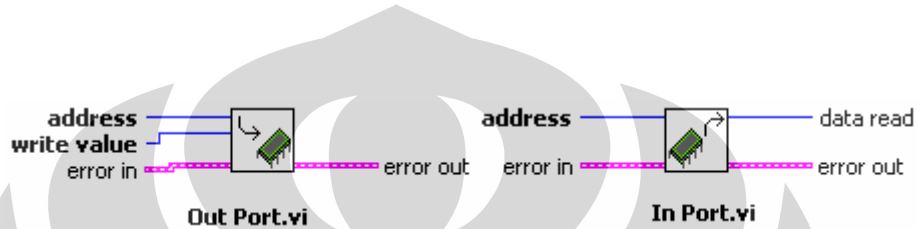
Gambar 3-7 Block diagram sub vi FC

Tabel 8. Data yang dikirim frekuensi rem ke microcontroller.

Frekuensi Counter	$\overline{C3}$	C2	$\overline{C1}$	$\overline{C0}$	Data yang dikirim ke alamat 890
1 Hz	1	0	1	1	11
2 Hz	1	1	1	1	15
5 Hz	0	0	1	1	3
10 Hz	0	1	1	1	7

3.4 Program Komunikasi Parallel

Guna mengakses suatu komunikasi parallel dapat digunakan instruksi out port dan in port. Instruksi Inport digunakan untuk mengambil data sedangkan instruksi outport digunakan untuk mengirimkan data ke microcontroller.



Gambar 3-8. icon outport dan import

Data yang dikirim memiliki alamat masing- masing sesuai dengan pin pada port printer sesuai dengan table 9. Alamat pada port printer terhubung dengan hardware dengan fungsi yang telah ditentukan.

Tabel 9. adres pada printer port dan fungsinya pada program

Alamat	Koneksi	Fungsi
888	D0-D7	Pengiriman Data Power motor dan duty cycle rem
889	S3-S6	Pembacaan Data RPM
890	C0-C3	Pengaturan Enable

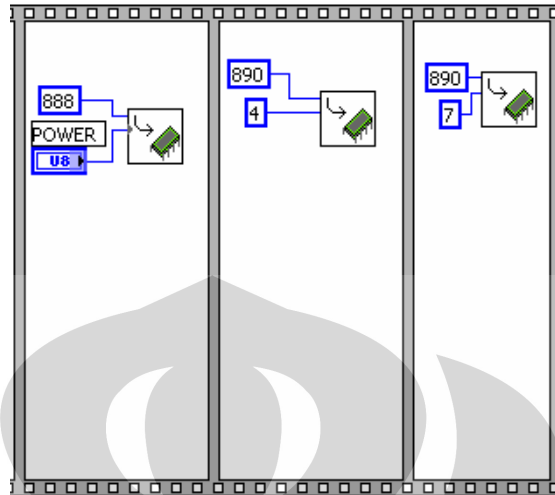
3.4.1 Pengiriman dan Pengambilan Data dari Microcontroller

Setelah data diproses sedemikian rupa maka data akan dikirim melalui port parallel. Data yang telah diproses sedemikian rupa ini akan diterima oleh setiap bagian dalam modul pengereman: Pengendali aktuator, pengendali rem. Masing-

masing bagian akan memproses data ini sesuai yang dibutuhkannya. Seperti terlihat pada flow chart keseluruhan modul, data akan diterima oleh pengendali aktuator yang akan menggerakkan motor penggerak silinder. Pada saat silinder bergerak, sensor pada modul pengereman terus menerus memberi informasi tentang kecepatan putar roda pada pengendali aktuator. Jika kecepatan putar roda sudah mencapai kecepatan yang ditentukan maka pengendali aktuator akan berhenti memutar motor penggerak dan pada saat yang bersamaan pengendali rem aktif. Pengendali rem ini akan mengaktifkan fungsi rem-nya sesuai mode pengereman yang ditentukan : Termodulasi (Duty Cycle = 100%) atau Konvensional (Duty Cycle < 100%). Dan pada saat yang bersamaan juga sensor akan memberi informasi tentang pengurangan kecepatan putar silinder pada pengendali rem dan terima data sensor. Informasi dari sensor pada pengendali rem digunakan untuk mengetahui apakah silinder sudah berhenti atau belum.

3.4.2 Pengiriman data power motor

Pada program LabVIEW terdapat subVI yaitu LPT rem. Sub VI ini berfungsi untuk mengirimkan data 8 bit ke microcontroller melalui komunikasi parallel. Data pertama yang dikirimkan yaitu power motor. Data power motor dikirimkan ke alamat register 888 melalui fungsi out port. Dimana port 888 berfungsi sebagai Data Port pada komunikasi parallel.



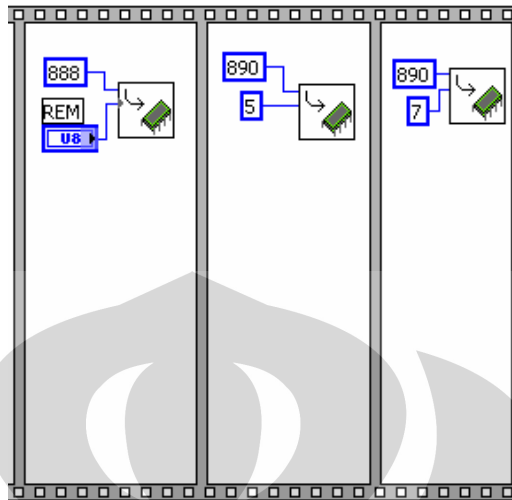
Gambar 3-9. sub vi pengiriman data 8 bit power motor ke microcontroller

Write value yaitu nilai yang akan dikirim ke address. Write value yang dipakai yaitu data 8 bit power motor yang akan dikirim ke address 888.

Pada pengiriman data power juga dikirimkan nilai 4 (Nilai C2 kondisi high/ INT1 = 1) ke address 890. Dan setelah itu dikirim lagi nilai 7 (open nibble high) ke alamat register 890. Alamat register 890 merupakan Printer Control.

3.4.3 Pengiriman data duty cycle rem

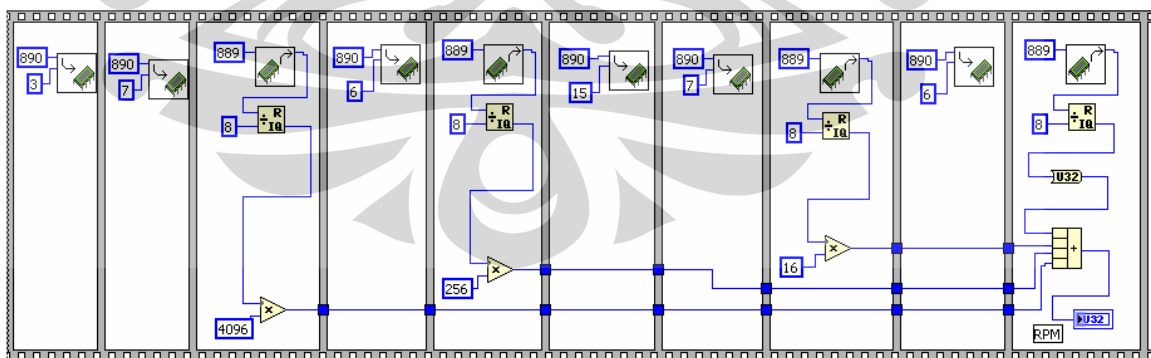
Pada pengiriman duty cycle rem tidak jauh berbeda dengan pengiriman data power motor. Data 8 bit dari rem dikirim ke alamat register 888 yaitu port Data Printer. Setelah itu nilai 5 (Nilai C2 dan $\overline{C0}$ dalam kondisi high) dikirimkan ke address 890, dimana alamat register tersebut merupakan Printer Control. Setelah itu dikirimkan kembali nilai 7 (open nibble high yaitu $\overline{C0}$, C1, $\overline{C2}$ bernilai = 1).



Gambar 3-10. sub vi pengiriman data 8 bit duty cycle rem ke microcontroller

3.4.4 Pengambilan data RPM

Microcontroller akan mengirimkan data berupa 16 bit ke labview melalui komunikasi parallel. Data yang diterima tidak langsung 16 bit tetapi di pecah menjadi 4 bit sehingga terjadi pengambilan data 4 bit sebanyak 4 kali.



Gambar 3-10. sub vi penerimaan data 18 bit RPM

Pertama-tama nilai 3 dikirimkan ke alamat 890. Dimasukkan nilai 3 karena untuk mengaktifkan interrupt 1 maka akan mengambil nilai high byte. Setelah itu dikirim nilai 7 (open high nibble) seperti yang terlihat pada table dibawah ini. Lalu akan diterima 4 data high pertama sehingga dikalikan dengan nilai 4096. Selanjutnya akan di buka nibble low ($\overline{C1}$, C2 aktif) dengan mengirim nilai 6 ke alamat register 890, Maka 4 bit low akan di dapat. Sampai disini data yang diterima yaitu high byte.

Tabel 9. 4 bit yang dikirimkan ke microcontrollercontrollercontroller dari printer port

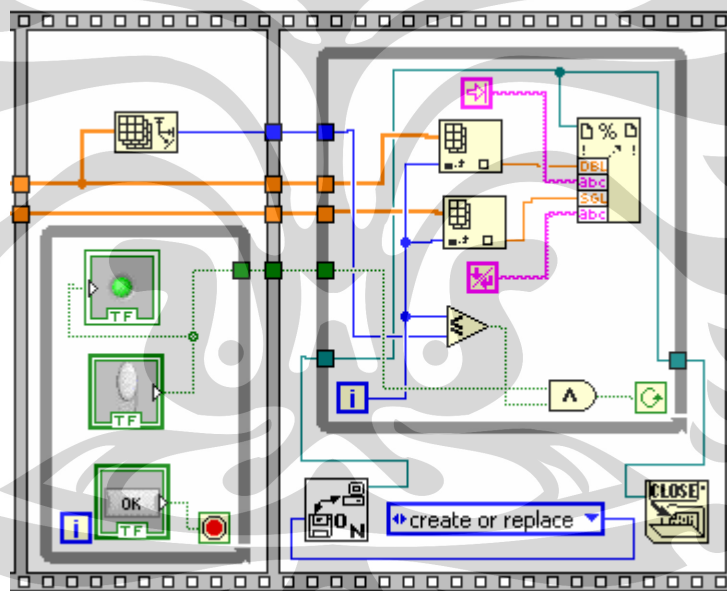
$\overline{C3}$	C2	$\overline{C1}$	$\overline{C0}$	KETERANGAN
INT 0	INT 1			
0	1	1	1	Open nibble high
0	1	1	0	Open nibble low
0	1	0	1	Masukkan data rem
0	1	0	0	Masukkan power motor
0	0	1	1	INT 1 aktif, nilai high byte
1	1	1	1	INT 0 aktif, nilai low byte

Untuk mendapatkan nilai low byte, labview akan menginterrupt C2 sehingga interrupt 0 aktif. Maka nilai 15 dikirim ke alamat register 890. Selanjutnya dikirim nilai 7 ke alamat 890 karena untuk membuka nibble high yaitu mengaktifkan $\overline{C1}$, C2 dan $\overline{C0}$. Pada alamat 889 diterima 4 bit high lalu dikalikan dengan 16. Setelah

itu nilai 6 dikirim ke alamat 890 untuk membuka nibble low. Dan pada akhirnya semua data dijumlahkan maka akan diterima data 16 bit.

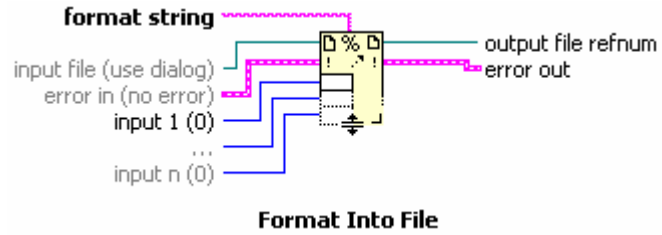
3.5 Penyimpanan data ke Microsoft Excel

Penyimpanan data disini yaitu proses penyimpanan data hasil pengolahan program sistem pengereman ke dalam file excel (.xls). Adapun data yang disimpan ke excel yaitu data RPM (Km/Jam) dan waktu (sekon) yang didapat dari hasil kali antara iterasi dengan Time Sampling (TS).



Gambar 3-11. program simpan data ke excel

Adapun data-data hasil pengolahan sistem pengereman dimasukkan kedalam dimensi array sedangkan index array terhubung ke iterasi. Sedangkan element array yang dihasilkan dimasukkan ke dalam format file excel (.xls).



Gambar 3-12. Icon format into file

Fungsi diatas berfungsi untuk memformat input data berupa string, numeric, boolean untuk dimasukkan ke file.

BAB IV

PENGUJIAN PROGRAM DAN ANALISA

Pengujian program dilakukan agar dapat diketahui apakah program yang dibuat sudah berfungsi seperti yang diharapkan atau belum. Pengujian ini terdiri dari pengujian rangkaian interfacing dan pengujian program. Pengujian program yang akan dilakukan yaitu: Pengujian pengiriman data, pengujian penerimaan data, penyimpanan data ke excel. Dan tidak lupa analisa setelah dilakukan pengujian.

4.1 Pengujian Rangkaian Interfacing

Pada pengujian rangkain interfacing, menggunakan komunikasi parallel. Pengujian menggunakan rangkaian utama dimana terdapat microcontroller dan driver motor. Untuk menguji apakah rangkaian interfacing sudah terhubung ke microcontroller menggunakan logic probe. Pada logic probe terdapat indikasi *pulse* yang menandakan terdapat pulsa dan terdapat pula indikasi bit low (led merah) dan bit high untuk menandakan bit high (led hijau).

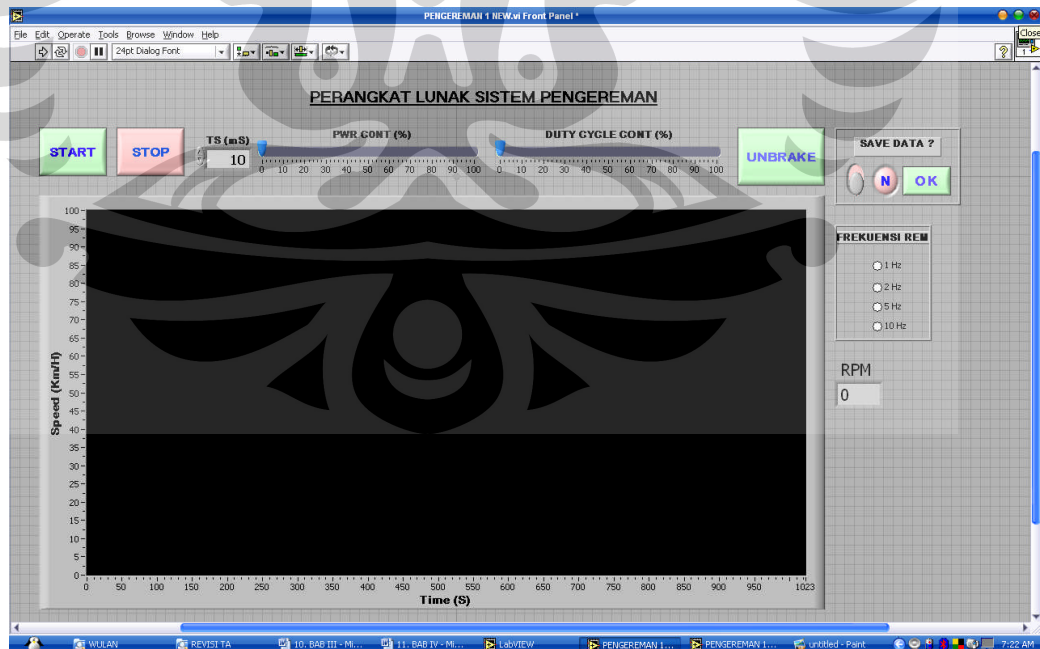
LabVIEW akan mengirimkan data power motor dan duty cycle rem masing-masing 8 bit. Pada pengiriman data power motor akan diterima microcontroller (ATMEL S52) pada kaki 32-39. Ketika power motor di set menjadi 50%, maka pada kaki 32 pada atmel S52 menunjukkan bit high pada logic probe. Lalu ketika

power motor di set kurang dari 50% maka kaki 32 pada atmel S52 menunjukkan bit low logic probe.

Sedangkan data duty cycle rem akan diterima pada ATMEL S52 di kaki 21-28. Ketika duty cycle rem di set pada LabVIEW sebesar 50% maka pada kaki 28 akan terdeteksi high bit pada logic probe. Dari pengujian diatas dapat dikatakan bahwa pengiriman data dari LabVIEW ke microcontroller dapat dikatakan berhasil tersampaikan.

4.2 Pengujian Program

Pada pengujian perangkat ini adalah semua fungsi utama program ini akan diujicobakan untuk mengetahui apakah ada kesalahan atau tidak. Berikut adalah "Front Panel" Program Pengaturan Modul Pengereman:



Gambar 4-1. Front Panel program system pengereman

Pengujian pada program ini adalah pengujian secara keseluruhan yang bertujuan untuk mengetahui ada kesalahan pemrograman atau tidak. Kesalahan atau *error* yang terjadi pada awal-awal pengujian *program* ini adalah seringnya pada kesalahan pada konversi data yang diterima yaitu 16 bit di konversi menjadi nilai Km/Jam sehingga akan memengaruhi pemrosesan data selanjutnya. Namun demikian, pengujian *program* telah bisa dikatakan berhasil.

4.3.1 Pengujian Pengiriman Data

Pada proses pengujian pengiriman data power motor dan rem tidak menemukan hambatan. Pengujian secara manual yang dilakukan yaitu ketika rangkaian utama (microcontroller) dihubungkan pada komunikasi parallel menggunakan indikasi logic probe.

Data power motor dikirim ke P0 pada microcontroller, jadi bila menggunakan logic probe akan terdeteksi pulsa pada P0 yaitu dikaki 32-39. Berarti data power motor telah di kirim 8 bit. Sama halnya bila mengirim data rem (Duty Cycle Controller) ke microcontroller, dimana pada microcontroller akan diterima di P2 yaitu dikaki 21-28. Jika di tes menggunakan logic probe pada P2 akan terdapat pulsa yang menandakan terkirimnya data 8 bit.

4.3.1.1 Pengiriman data power controller

Pada front panel dapat diatur nilai power controller yaitu berupa motor dc. Nilai power controller dapat diatur dari 0% sampai 10%. Sedangkan data dalam

persen dikali 2,55 pada pada program utama lalu dikirim berupa bilangan hexsa ke microcontroller. Dibawah ini merupakan data yang diterima pada port 0 pada microcontrollercontrollercontroller pada saat pengiriman nilai power controller:

Tabel 10. 8 bit data power motor yang dikirimkan ke microcontroller dari printer

port

Power Controller (%)	Data yang dikirim (Nilai decimal)	Data yang terima microcontroller (Nilai heksa)	Data yang terima (nilai biner)							
			PORT 1							
			7	6	5	4	3	2	1	0
			128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0	0	0
10	25	19	0	0	0	1	1	0	0	1
20	51	33	0	0	1	1	0	0	1	1
30	76,5	4C	0	1	0	0	1	1	0	0
40	102	66	0	1	1	0	0	1	1	0
50	127	7F	0	1	1	1	1	1	1	1
60	153	99	1	0	0	1	1	0	0	1
70	178	B2	1	0	1	1	0	0	1	0
80	204	CC	1	1	0	0	1	1	0	0
90	229	E5	1	1	1	0	0	1	0	1
100	255	FF	1	1	1	1	1	1	1	1

Setelah diperoleh data diatas, komunikasi pengiriman power controller berhasil karena data yang disampaikan dari PC sama dengan data yang diterima pada microcontroller.

4.3.1.2 Pengiriman data duty cycle controller

Pada pengiriman data duty cycle tidak jauh berbeda dengan pengiriman nilai power controller. Perbedaannya terdapat kondisi BRAKE (rem) dan UNBRAKE (tidak di rem).

Tabel 11. Kondisi data yang dikirim jika di rem atau tidak di rem

UNBRAKE (tidak di rem)	Data yang dikirim = 1	
BRAKE (di rem)	Data yang dikirim = 0% sampai 100%	<ul style="list-style-type: none">• Konvensional: 100%• Termodulasi: < 100%

Jika di BRAKE data yang dikirim yaitu nilai duty cycle controller (0-100%). Data yang diterima di port 2 pada microcontroller yaitu:

Tabel 12. Kondisi data 8 bit yang dikirim jika di rem (brake)

Power Controller (%)	Data yang dikirim (Nilai decimal)	Data yang terima microcontroller (Nilai heksa)	Data yang terima (nilai biner)							
			PORT 2							
			7	6	5	4	3	2	1	0
			128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0	0	0
10	25	19	0	0	0	1	1	0	0	1
20	51	33	0	0	1	1	0	0	1	1
30	76,5	4C	0	1	0	0	1	1	0	0
40	102	66	0	1	1	0	0	1	1	0
50	127	7F	0	1	1	1	1	1	1	1
60	153	99	1	0	0	1	1	0	0	1
70	178	B2	1	0	1	1	0	0	1	0
80	204	CC	1	1	0	0	1	1	0	0
90	229	E5	1	1	1	0	0	1	0	1
100	255	FF	1	1	1	1	1	1	1	1

Jadi, bila terjadi pengereman secara konvensional maka data yang diterima di microcontroller yaitu FF. Port 2.0 – Port 2.7 berkondisi high semua.

4.2.1 Pengujian Pengambilan Data

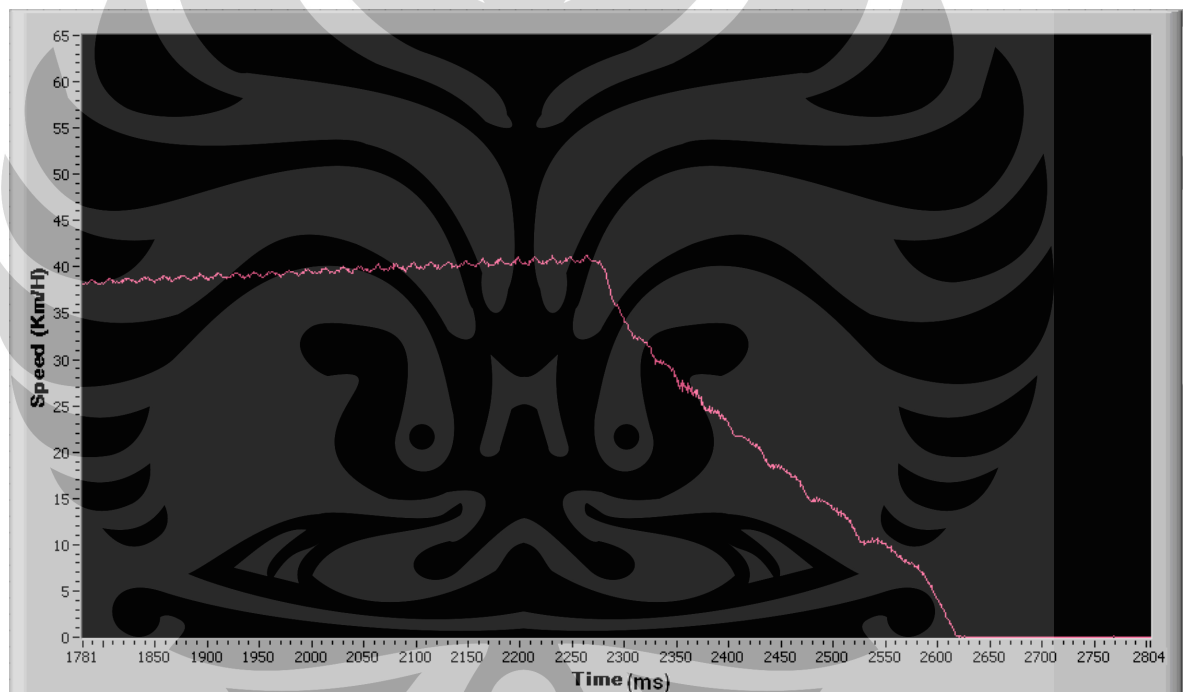
Pada pengujian pengambilan data, penulis mengambil data dari grafik yang ditunjukkan pada setiap percobaan. Pada pengambilan data power controller di

tentukan, lalu tunggu sampai diperoleh kecepatan roda stabil. Setelah itu dilakukan proses pengereman. Dapat dilihat grafik yang didapat pada setiap percobaan sebagai berikut:

GRAFIK 1

Power Controller = 100%

Duty Cycle Controller = 100%



Gambar 4-2 Grafik Konvensional power controller = 100%

Waktu pengereman = waktu pada saat roda berhenti – waktu mulai rem

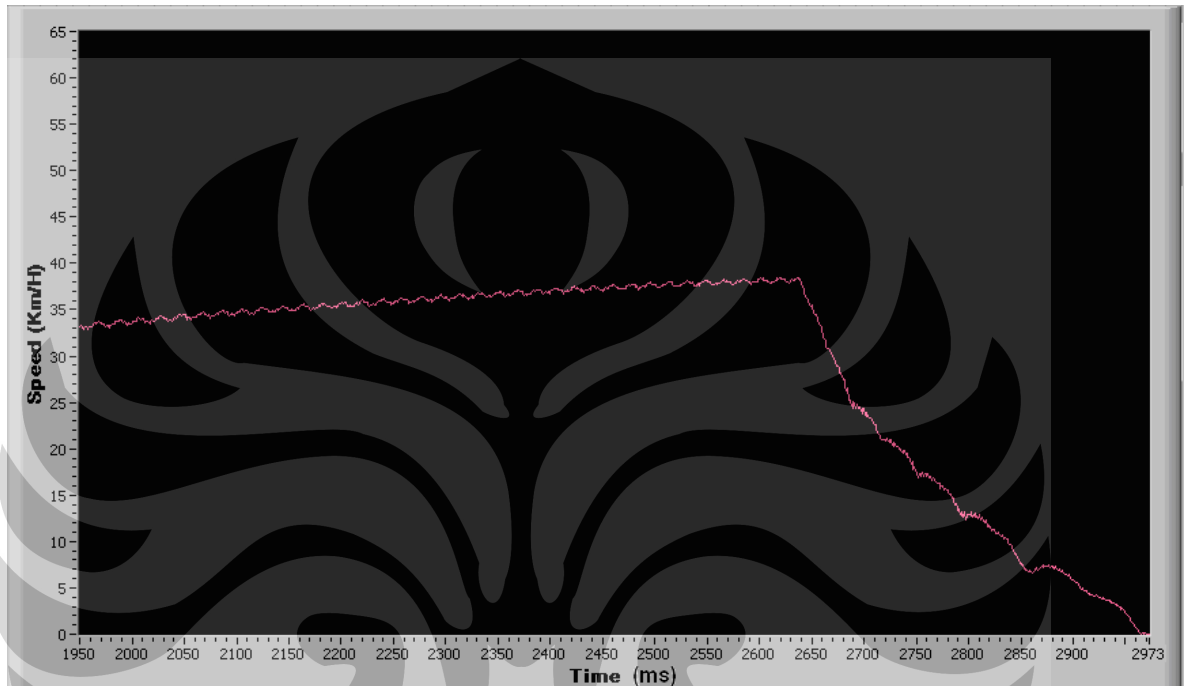
$$= (2620 - 2260) \times 10 \text{ ms}$$

$$= 3,6 \text{ sekon}$$

GRAFIK 2

Power Controller = 90%

Duty Cycle Controller = 100%



Gambar 4-3 Grafik Konvensional power controller = 90%

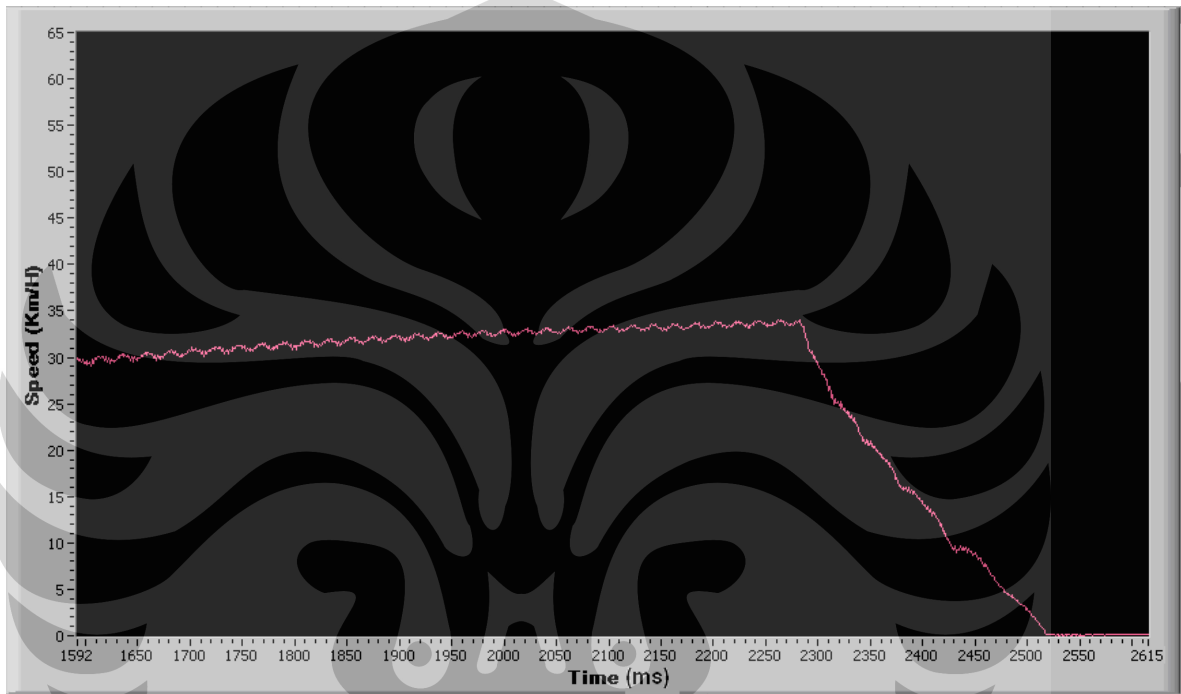
Waktu pengereman = $(2973 - 2640) \times 10 \text{ ms}$

= 3,3 sekon

GRAFIK 3

Power Controller = 80%

Duty Cycle Controller = 100%



Gambar 4-4 Grafik Konvensional power controller = 80%

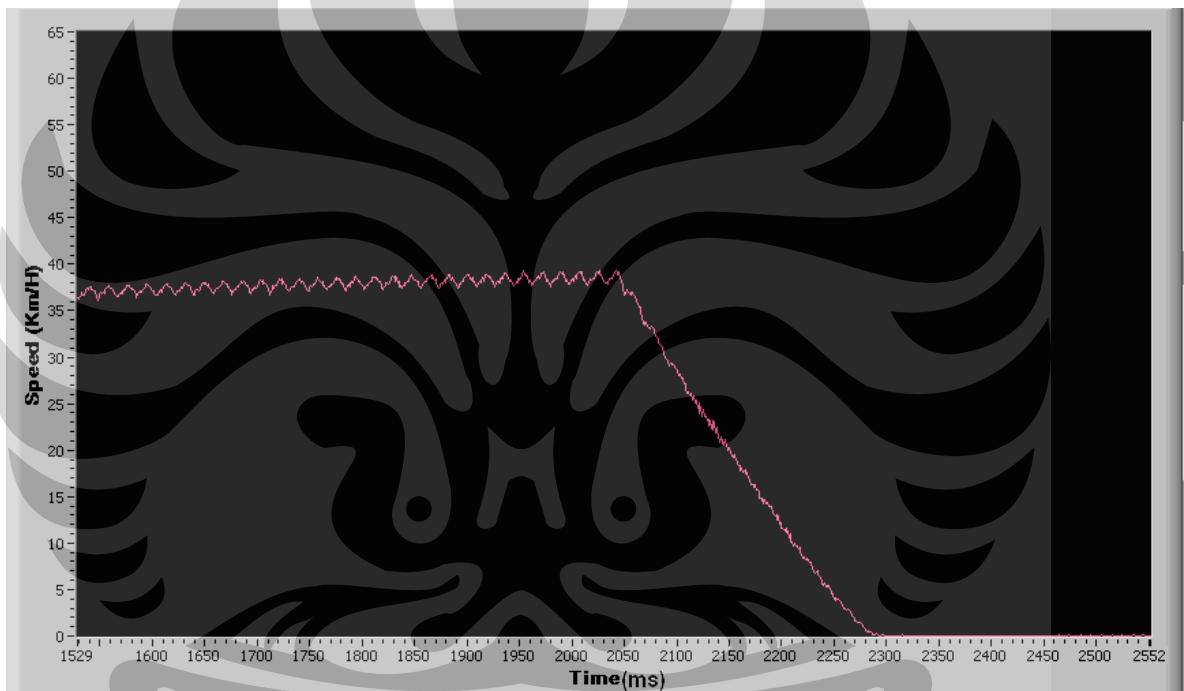
$$\begin{aligned}\text{Waktu pengereman} &= (2520 - 2280) \times 10 \text{ ms} \\ &= 2,9 \text{ sekon}\end{aligned}$$

GRAFIK 4

Power Controller = 90%

Duty Cycle Controller = 50%

Frekuensi Rem = 2 Hz



Gambar 4-5 Grafik termodulasi power controller = 90%, duty cycle = 50%,
frekuensi rem = 2 Hz

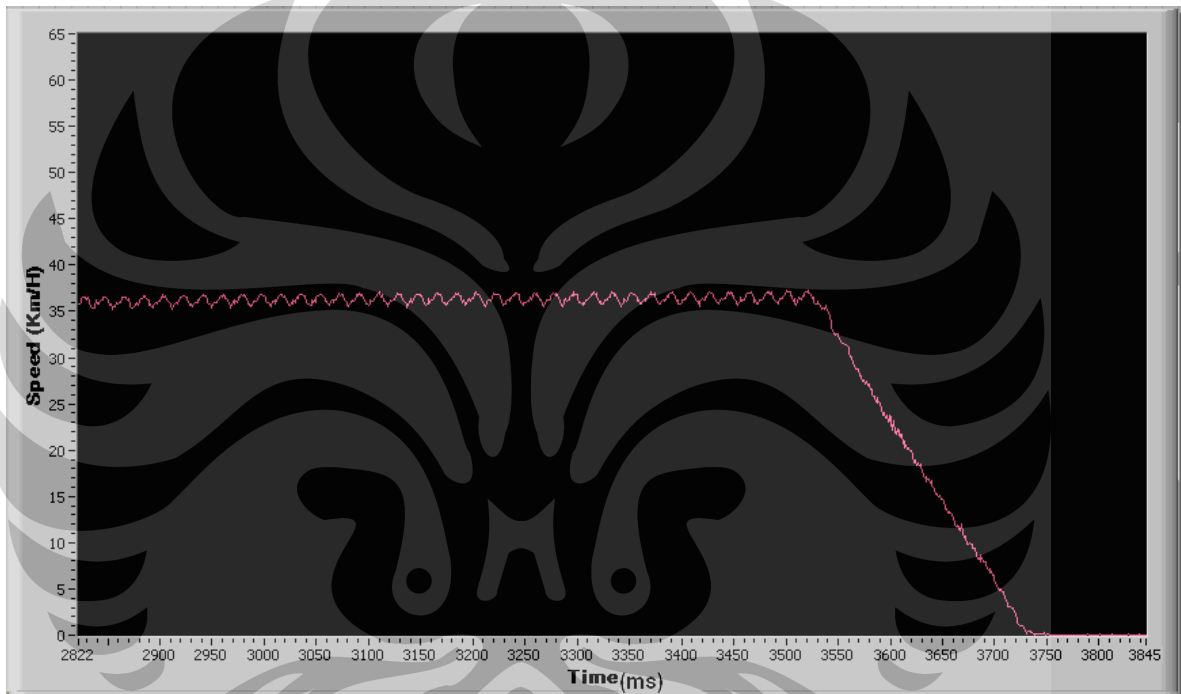
$$\begin{aligned}\text{Waktu pengereman} &= (2300 - 2050) \times 10 \text{ ms} \\ &= 2,5 \text{ sekon}\end{aligned}$$

GRAFIK 5

Power Controller = 90%

Duty Cycle Controller = 50%

Frekuensi Rem = 5 Hz



Gambar 4-6 Grafik termodulasi power controller = 90%, duty cycle = 50%,
frekuensi rem = 5 Hz

Waktu pengereman = $(3740 - 3510) \times 10 \text{ ms}$

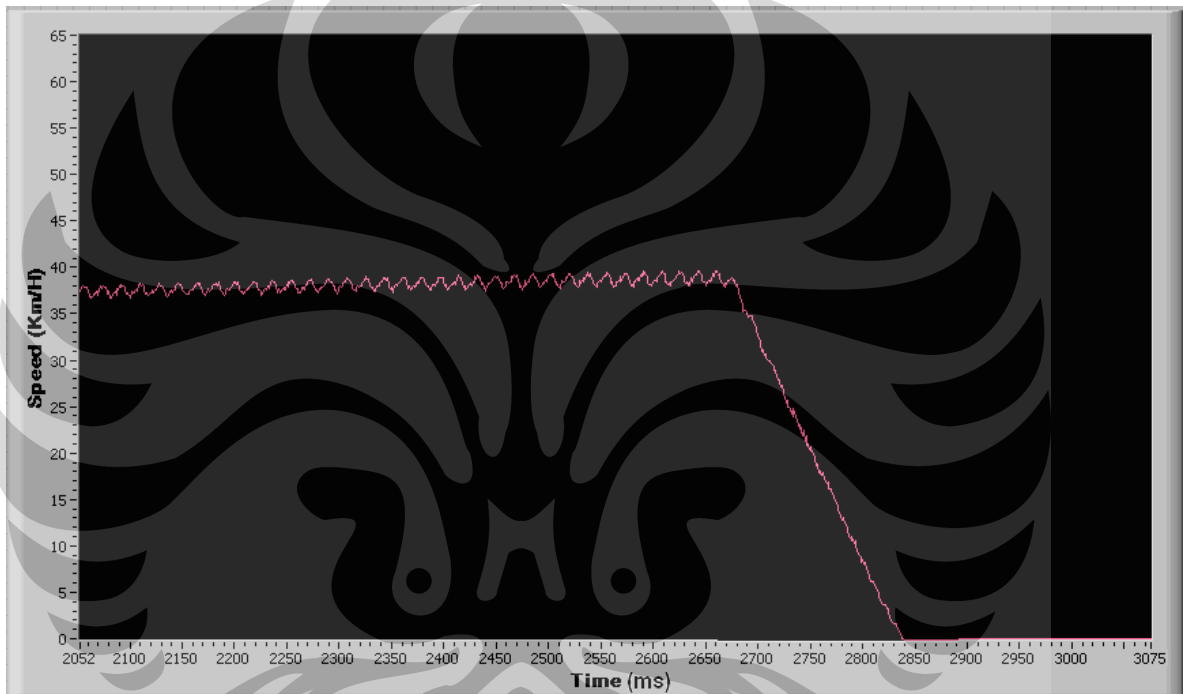
= 2,3 sekon

GRAFIK 6

Power Controller = 90%

Duty Cycle Controller = 60%

Frekuensi Rem = 2 Hz



Gambar 4-7 Grafik termodulasi power controller = 90%, duty cycle = 60%,
frekuensi rem = 2 Hz

Waktu pengereman = $(2850 - 2660) \times 10 \text{ ms}$

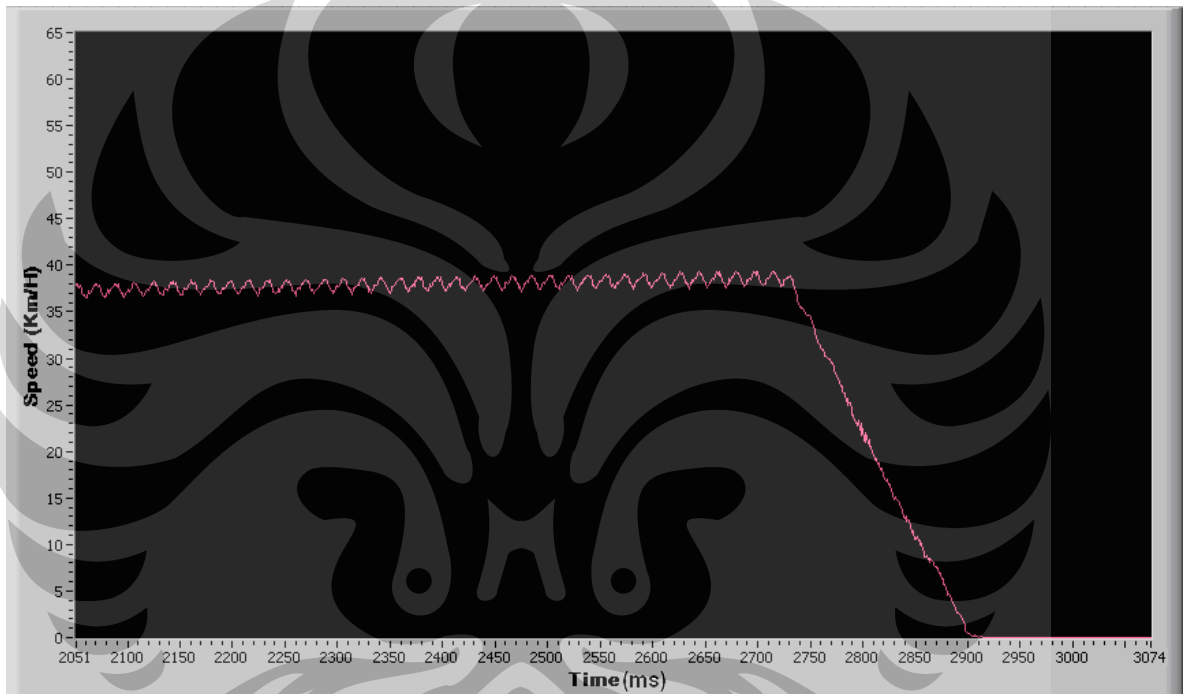
= 1,9 sekon

GRAFIK 7

Power Controller = 90%

Duty Cycle Controller = 60%

Frekuensi Rem = 5 Hz



Gambar 4-8 Grafik termodulasi power controller = 90%, duty cycle = 60%,
frekuensi rem = 5 Hz

Waktu pengereman = $(2900 - 2720) \times 10 \text{ ms}$

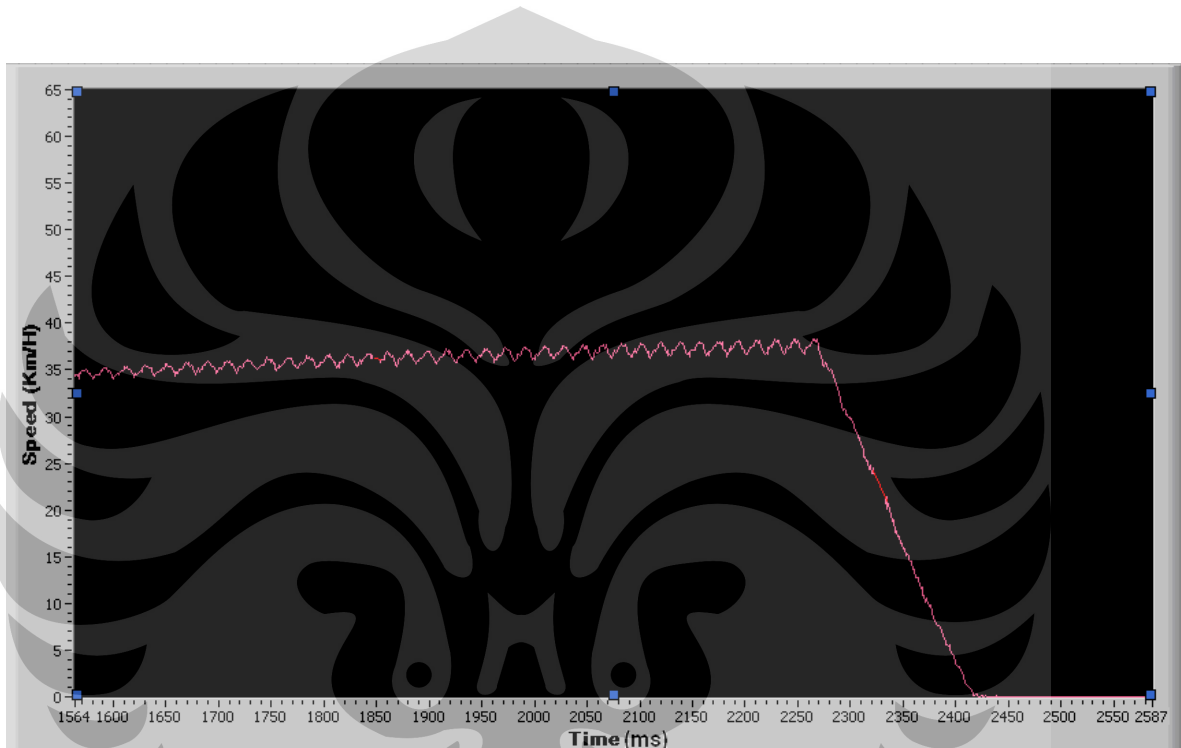
= 1,8 sekon

GRAFIK 8

Power Controller = 90%

Duty Cycle Controller = 70%

Frekuensi Rem = 2 Hz



Gambar 4-9 Grafik termodulasi power controller = 90%, duty cycle = 70%,
frekuensi rem = 2 Hz

Waktu pengereman = $(2420 - 2260) \times 10 \text{ ms}$

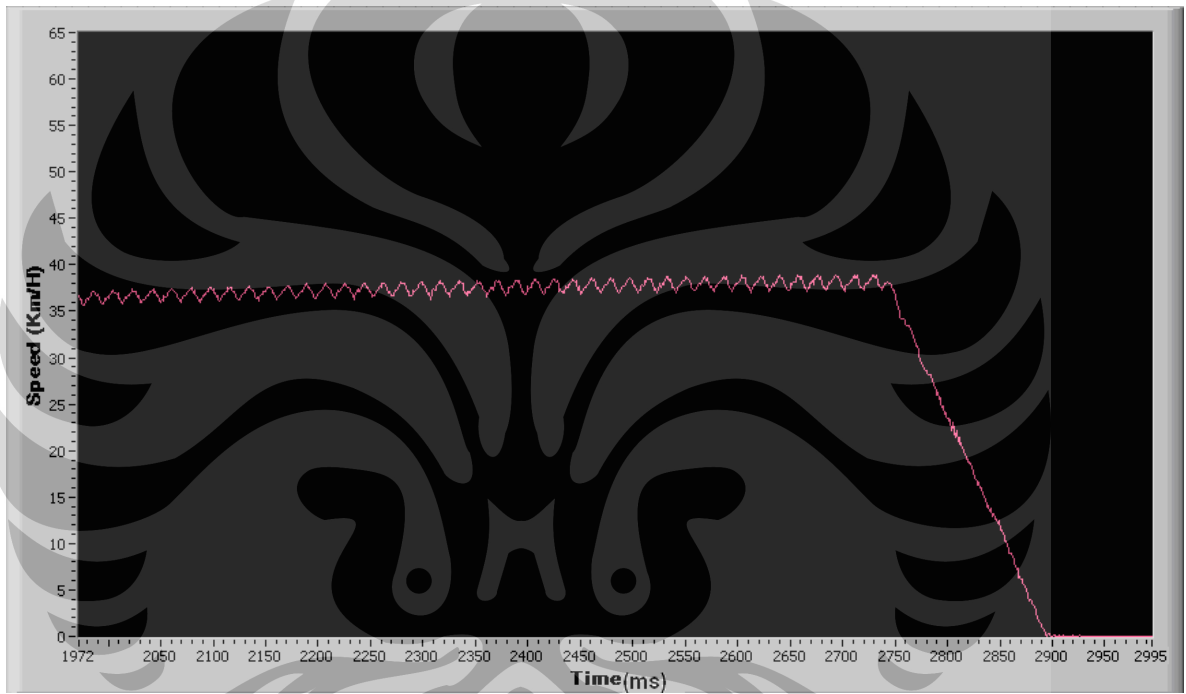
= 1,6 sekon

GRAFIK 9

Power Controller = 90%

Duty Cycle Controller = 70%

Frekuensi Rem = 5 Hz



Gambar 4-10 Grafik termodulasi power controller = 90%, duty cycle = 70%,
frekuensi rem = 5 Hz

Waktu pengereman = $(2890 - 2740) \times 10 \text{ ms}$

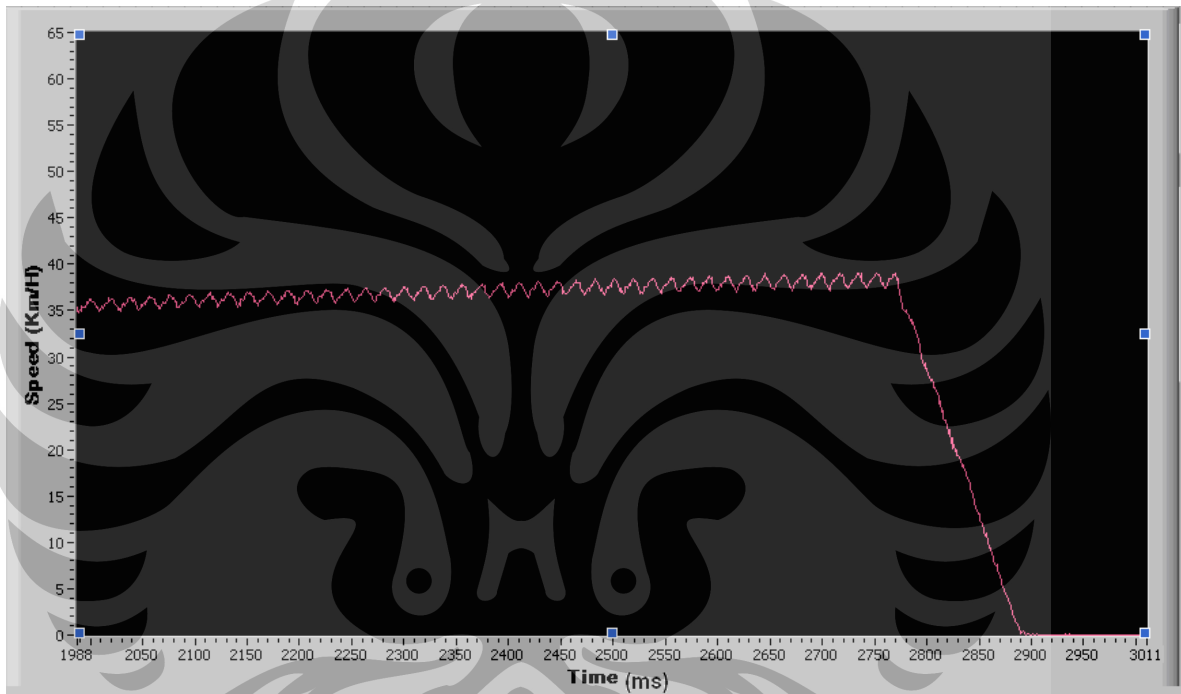
= 1,5 sekon

GRAFIK 10

Power Controller = 90%

Duty Cycle Controller = 80%

Frekuensi Rem = 2 Hz



Gambar 4-11 Grafik termodulasi power controller = 90%, duty cycle = 80%,
frekuensi rem = 2 Hz

Waktu pengereman = $(2890 - 2760) \times 10 \text{ ms}$

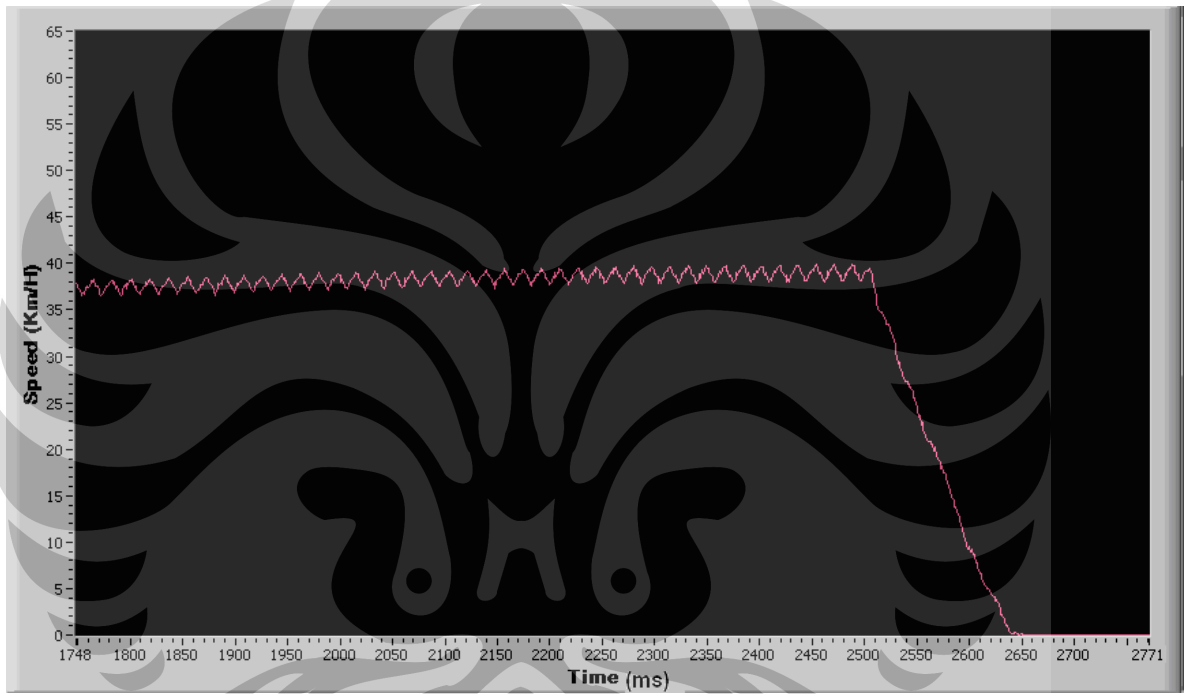
= 1,3 sekon

GRAFIK 11

Power Controller = 90%

Duty Cycle Controller = 80%

Frekuensi Rem = 5 Hz



Gambar 4-12 Grafik termodulasi power controller = 90%, duty cycle = 80%,
frekuensi rem = 5Hz

Waktu pengereman = $(2650 - 2530) \times 10 \text{ ms}$

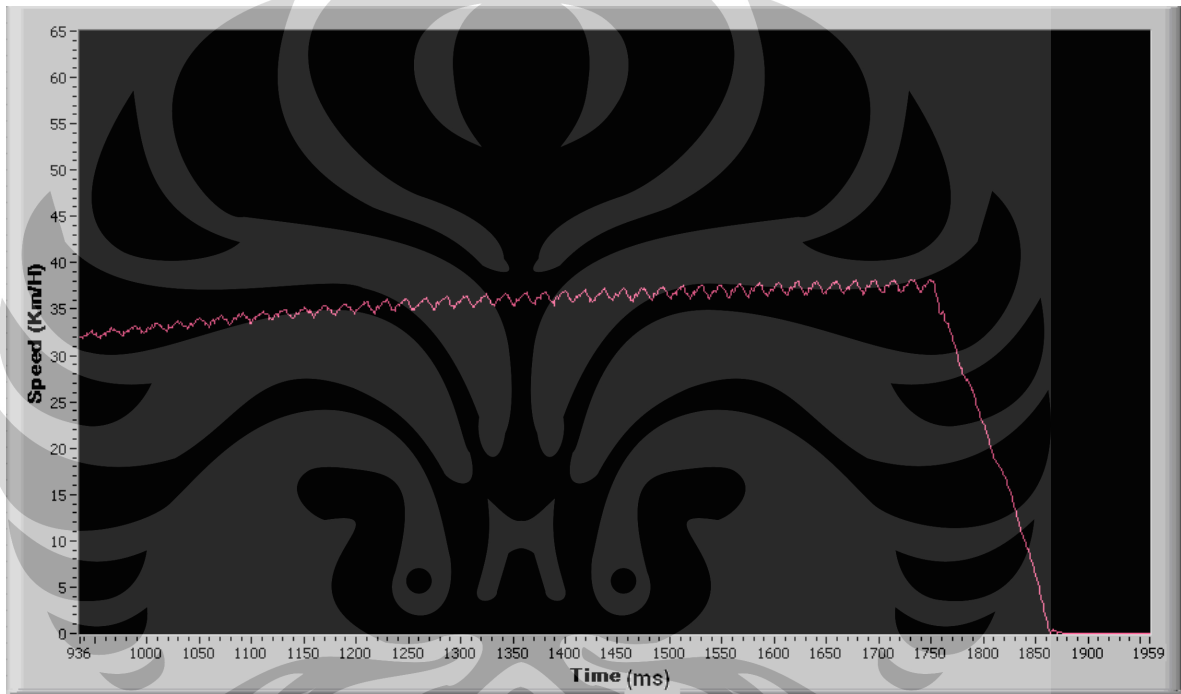
= 1,2 sekon

GRAFIK 12

Power Controller = 90%

Duty Cycle Controller = 90%

Frekuensi Rem = 2 Hz



Gambar 4-13 Grafik termodulasi power controller = 90%, duty cycle = 90%,
frekuensi rem = 2Hz

Waktu pengereman = $(1860 - 1750) \times 10 \text{ ms}$

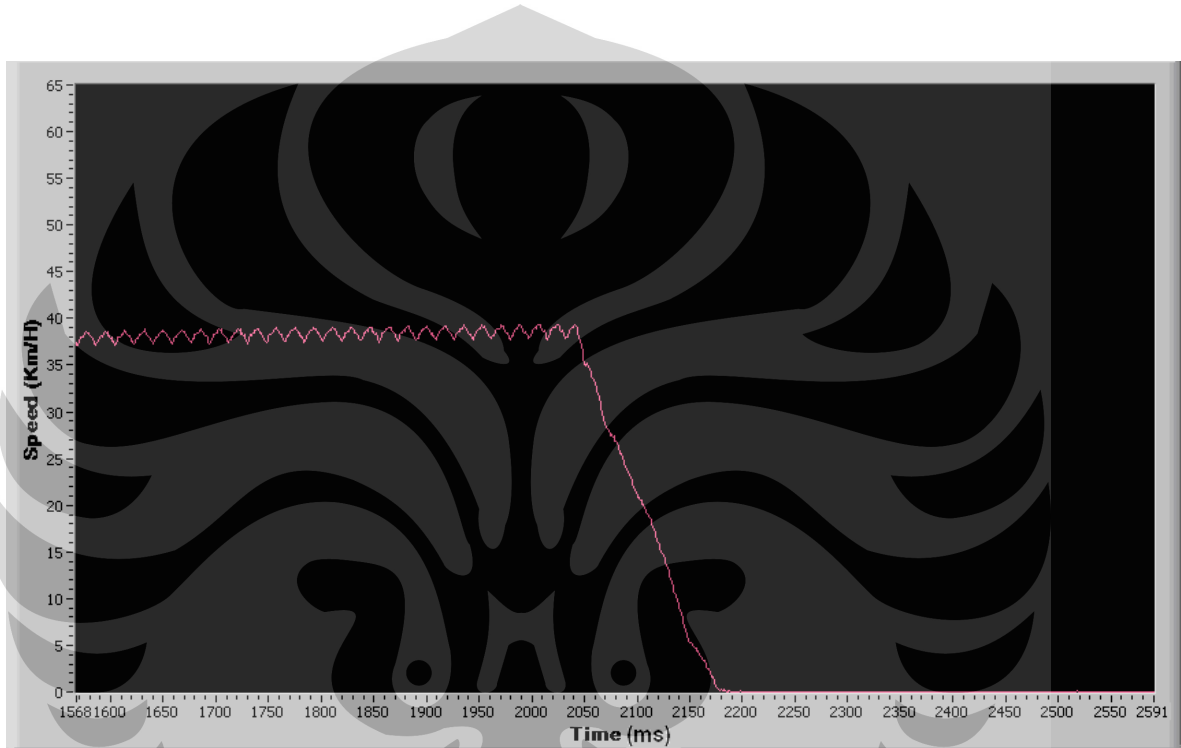
= 1,1 sekon

GRAFIK 13

Power Controller = 90%

Duty Cycle Controller = 90%

Frekuensi Rem = 5 Hz



Gambar 4-14 Grafik termodulasi power controller = 90%, duty cycle = 90%,
frekuensi rem = 5 Hz

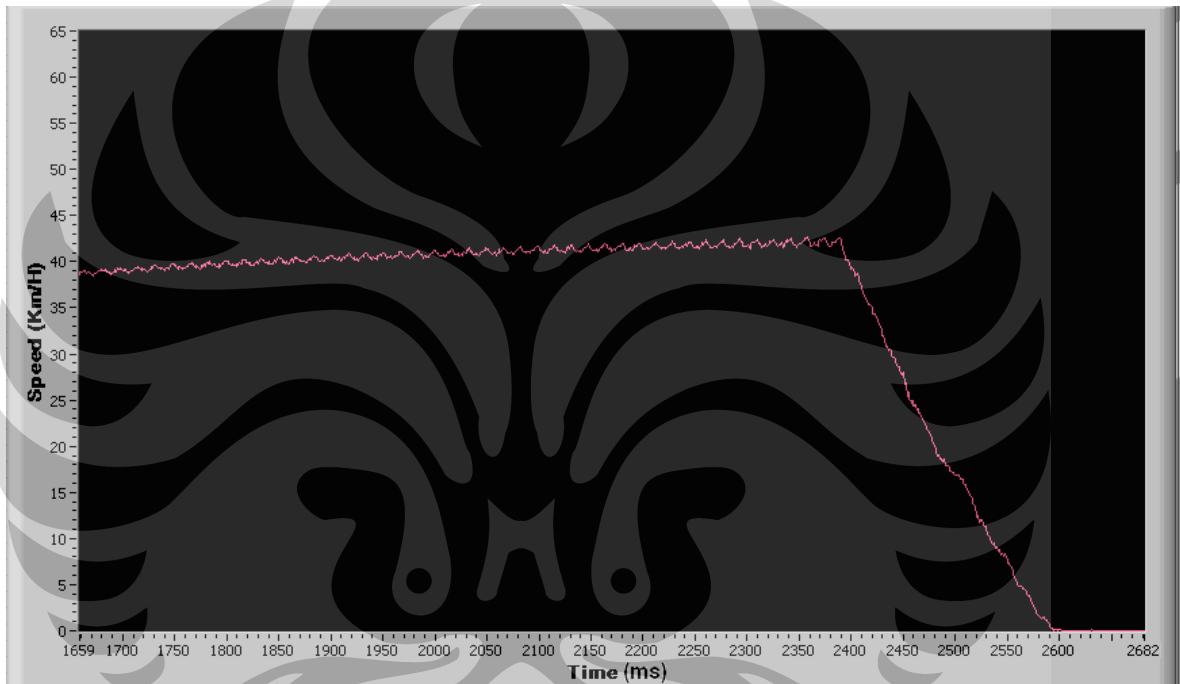
$$\begin{aligned}\text{Waktu pengereman} &= (2180 - 2080) \times 10 \text{ ms} \\ &= 1 \text{ sekon}\end{aligned}$$

GRAFIK 14

Power Controller = 100%

Duty Cycle Controller = 80%

Frekuensi rem = 2 Hz



Gambar 4-15 Grafik termodulasi power controller = 100%, duty cycle = 80%,
frekuensi rem = 2 Hz

Waktu pengereman = $(2590 - 2380) \times 10 \text{ ms}$

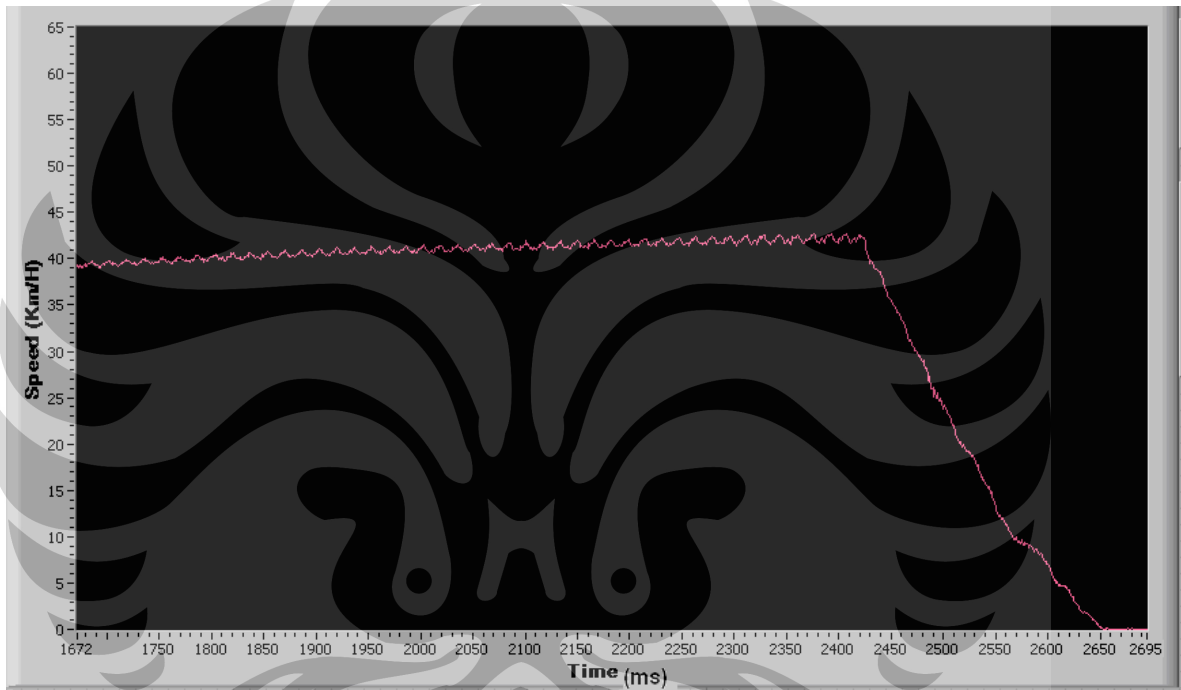
= 2,1 sekon

GRAFIK 15

Power Controller = 100%

Duty Cycle Controller = 80%

Frekuensi Rem = 5 Hz



Gambar 4-16 Grafik termodulasi power controller = 100%, duty cycle = 80%,
frekuensi rem = 5 Hz

Waktu pengereman = $(2650 - 2410) \times 10 \text{ ms}$

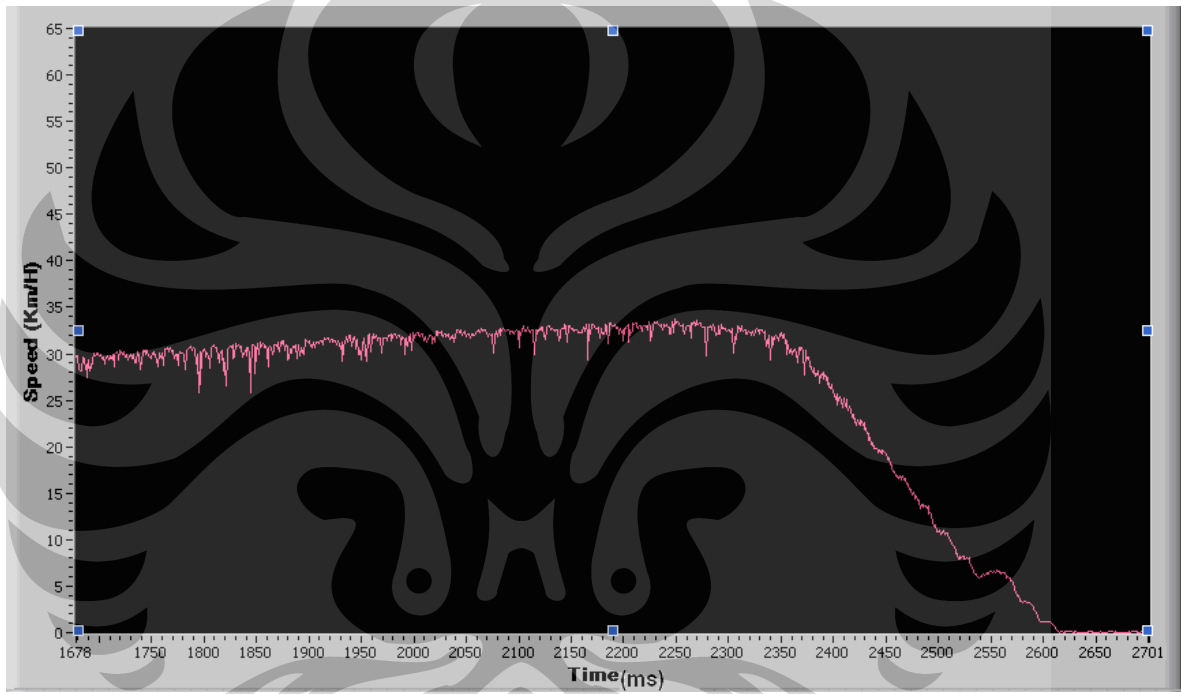
= 2,4 sekon

GRAFIK 16

Power Controller = 80%

Duty Cycle Controller = 60%

Frekuensi rem = 2 Hz



Gambar 4-17 Grafik termodulasi power controller = 80%, duty cycle = 60%,
frekuensi rem = 2 Hz

Waktu pengereman = $(2620 - 2350) \times 10 \text{ ms}$

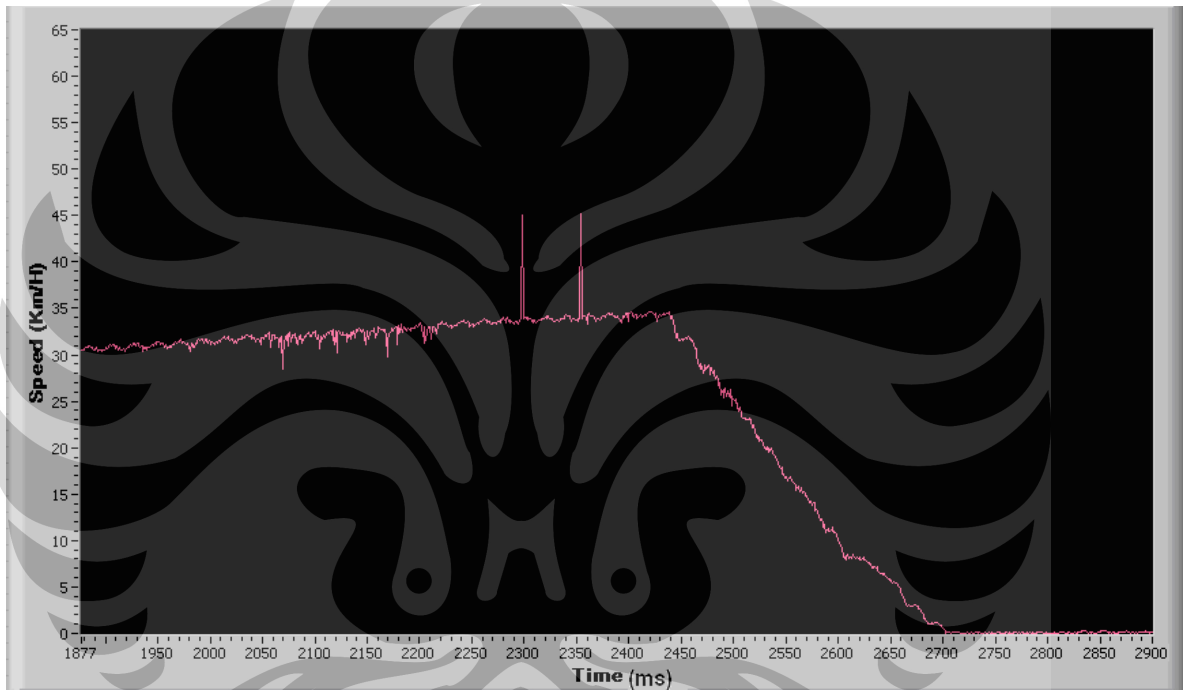
= 2,7 sekon

GRAFIK 17

Power Controller = 80%

Duty Cycle Controller = 60%

Frekuensi rem = 5 Hz



Gambar 4-17 Grafik termodulasi power controller = 80%, duty cycle = 60%,
frekuensi rem = 5 Hz

Waktu pengereman = $(2710 - 2430) \times 10 \text{ ms}$

= 2,8 sekon

Data yang diambil pada penelitian ini adalah:

- **Pengereman secara Konvensional**

Tabel 13. Data pengereman secara konvensional

No. Grafik (lampiran)	Pwr Controller	Duty Cycle Controller	Kecepatan (Km/Jam)	Waktu Pengereman (Sekon)
Grafik 1	100%	100%	41	3,6
Grafik 2	90%	100%	36	3,3
Grafik 3	80%	100%	32	2,9

- **Pengereman secara termodulasi**

1. Nilai duty cycle controller yang berbeda:

Tabel 14. Data pengereman secara termodulasi dengan perbedaan nilai duty cycle controller

No. Grafik (lampiran)	Pwr Controller	Duty Cycle Controller	Frekuensi Rem (Hz)	Kecepatan (Km/Jam)	Waktu Pengereman (Sekon)
Grafik 4	90%	50%	2	36	2,5
Grafik 5	90%	50%	5	36	2,3
Grafik 6	90%	60%	2	36	1,9

Grafik 7	90%	60%	5	36	1,8
Grafik 8	90%	70%	2	36	1,6
Grafik 9	90%	70%	5	36	1,5
Grafik 10	90%	80%	2	36	1,3
Grafik 11	90%	80%	5	36	1,2
Grafik 12	90%	90%	2	36	1,1
Grafik 13	90%	90%	5	36	1

2. power controller dan duty cycle controller yang berbeda

Tabel 15. Data pengereman secara termodulasi dengan perbedaan nilai duty cycle controller dan power controller

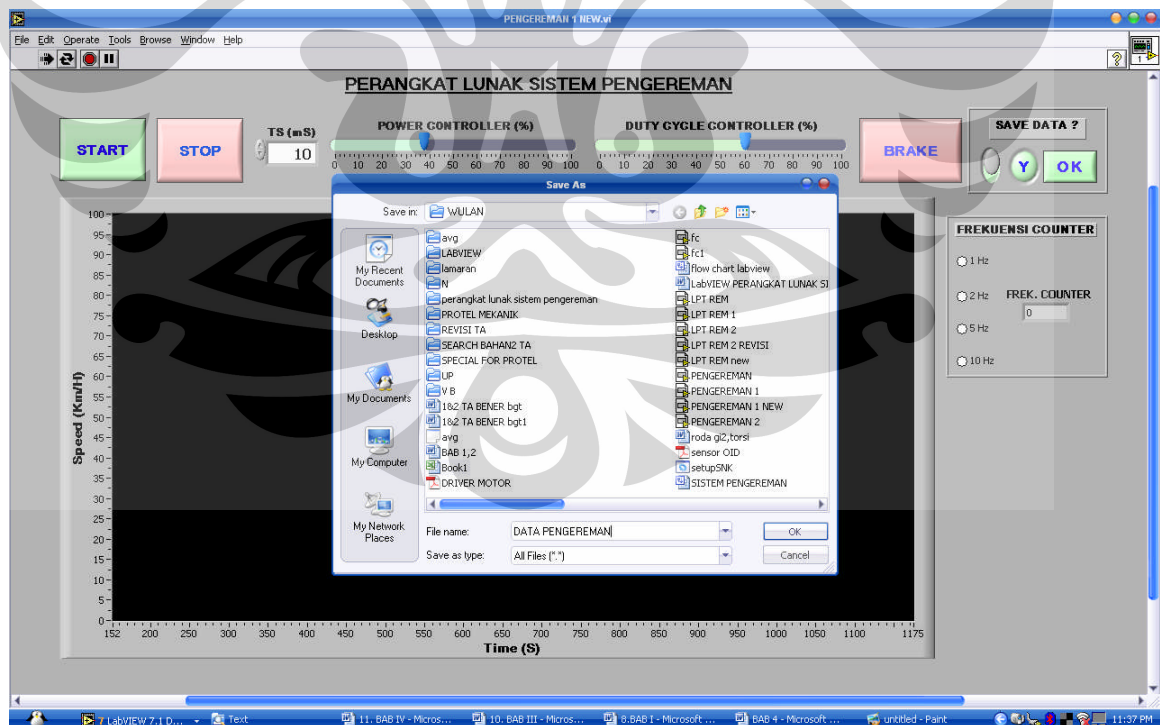
No. Grafik (lampiran)	Pwr Controller	Duty Cycle Controller	Frekuensi Rem (Hz)	Kecepatan (Km/Jam)	Waktu Pengereman (Sekon)
Grafik 14	100%	80%	2	41	2,1
Grafik 15	100%	80%	5	41	2,4
Grafik 16	80%	60%	2	32	2,7
Grafik 17	80%	60%	5	32	2,8

4.3.2 Pengujian Penyimpanan Data

Proses penyimpanan data ke excel dapat dikatakan berhasil. Terbukti dapat ditampilkan data RPM (Km/Jam) VS Waktu (Sekon). Setelah grafik ditampilkan user dapat menyimpan data yang didapat ke Microsoft Excel. Pada Front Panel dibawah ini terdapat pilihan save data?. User dapat menyimpannya atau tidak.

Jika user ingin menyimpan data, slide switch dapat ditekan lalu klik OK. Maka akan tampil SAVE AS. User dapat menyimpan langsung data dengan format excel. Contohnya : Data Pengereman.xls. Setelah itu user dapat melihat data di excel.

Apabila user tidak ingin menyimpan data maka slide switch ditekan, pada indicator led akan terindikasikan huruf N, lalu user dapat menekan tombol OK.



Gambar 4-3. tampilan jika ingin save data.

4.4 Analisa

Menurut pendapat banyak orang sistem pengereman termodulasi memiliki waktu pengereman yang lebih cepat dari pada sistem pengereman secara konvensional. Setelah dilakukan penelitian, didapatkan data dan dilakukan penelitian bahwa waktu pengereman secara termodulasi memang lebih cepat dibandingkan dengan konvensional.

Data yang diperoleh dari penelitian ini sesuai dengan teori yang telah dikemukakan sebelumnya pada bab 2. Dimana ketika terjadi pengereman konvensional roda akan langsung terkunci karena ada gaya gesekan kinetis antara jalan dan ban. Pengereman ini tidak maksimal karena besarnya gaya gesek kinetik lebih kecil dibandingkan gaya statik maksimal, sehingga mengakibatkan waktu pengereman yang lama.

Sedangkan pada proses pengereman termodulasi mengakibatkan roda tidak terkunci, maka akan terjadi gaya gesek kinetis antara kampas rem dengan bidang geseknya dan menimbulkan gaya reaksi yang sebanding dengan jalan terhadap roda yang menghambat laju kendaraan. Gaya reaksi semakin besar dan gaya kampas rem semakin besar pula. Batas maksimum besarnya gaya reaksi tersebut ditentukan dengan rumus $F = \mu_s \times N$. Pada kondisi ini terjadi pengereman maksimal pada frekuensi yang sesuai. Tingginya frekuensi dapat menyebabkan besarnya gaya gesek yang dapat memperpendek waktu pengereman. Dengan terjadinya pengereman yang maksimal maka waktu pengereman akan lebih cepat pula.

BAB V

PENUTUP

Bab ini berisi kesimpulan dari keseluruhan perancangan sistem hingga hasil penelitian yang didapat dan saran yang mungkin dapat digunakan untuk memperbaiki, menambahkan, ataupun memodifikasi alat yang sudah ada menjadi lebih baik lagi.

5.1 Kesimpulan

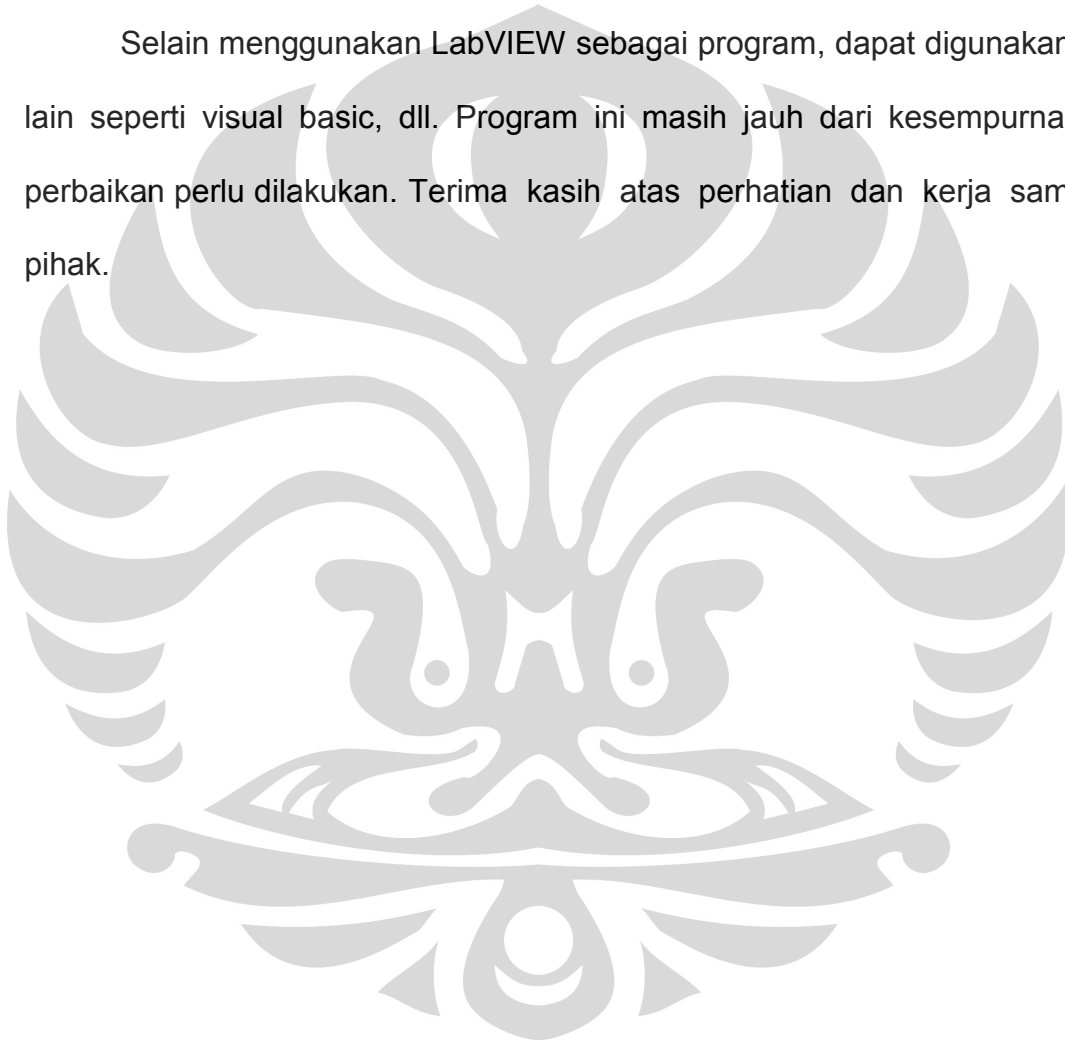
Setelah melakukan perancangan dan pengujian perangkat lunak sistem pengereman dapat disimpulkan:

1. Perangkat Lunak Sistem Pengereman berhasil digunakan untuk membandingkan sistem pengereman secara konvensional dan termodulasi.
2. Perangkat Lunak Sistem Pengereman ini dapat menganalisa sistem pengereman dan ditampilkan melalui grafik lalu dapat disimpan dalam bentuk file Microsoft Excel.
3. Setelah dilakukan perbandingan antara kedua sistem pengereman dapat disimpulkan waktu pengereman secara termodulasi lebih cepat dibandingkan sistem pengereman secara konvensional.

4. Waktu pengereman pada sistem pengereman termodulasi sangat dipengaruhi oleh besarnya frekuensi rem dan duty cycle controller.

V.2 Saran

Selain menggunakan LabVIEW sebagai program, dapat digunakan software lain seperti visual basic, dll. Program ini masih jauh dari kesempurnaan. Maka perbaikan perlu dilakukan. Terima kasih atas perhatian dan kerja sama semua pihak.



DAFTAR ACUAN

- 
- [1] Welling, Luke dan Thomson, Laura. *PHP and MySQL Web Development*. Indianapolis: Sams Publishing. 2003
- [2] Wellss, Lisa.K. *The LabVIEW Student Edition User's Guide*. NewJersey: Prentice Hall, 1994.
- [3] Tipler, Paul. A. *Fisika untuk Sains dan Teknik Jilid Kedua*. Jakarta: Erlangga. 1998.
- [4] Malvino, Albert Paul. *Electronic Principles Sixth Edition*. New York: McGraw-Hill.1999.
- [5] http://wikipedia/gerak_melingkar diakses pada tanggal 25 Mei 2007 pukul 17.00 WIB.