



**Pengembangan Sistem Kontrol dan *Monitoring*  
menggunakan Visual Studio Berbasis *C#* pada  
Mesin Conveyor *Wire Harness***

**Tugas Akhir**

**Oleh:  
Herdi Gunanta Sirait (4212001027)**

**Program Studi Teknik Mekatronika  
Jurusan Teknik Elektro  
Politeknik Negeri Batam  
2022**

## Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Sebutkan Judul Tugas Akhir Anda" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 26 Juli 2024



---

Herdi Gunanta Sirait  
NIM: 4212001027

## Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Terapan Teknik (S.Tr.T)  
di  
Politeknik Negeri Batam

Oleh:  
Herdi Gunanta Siralt (4212001027)

Tanggal Sidang: 16 Juli 2024

Disetujui oleh :



1. Daniel Sutopo Pamungkas, S.T.,M.T.,Ph.D  
NIK:109006



1. Diono, S.Tr. T., M.Sc  
NIK:120243



2. Fadli Firdaus  
NIK: 12227



2. Adlian Jefiza S.Pd.,M.T.  
NIK:119220

# **Pengembangan Sistem Kontrol dan *Monitoring* menggunakan Visual Studio Berbasis C# pada Mesin Konveyor *Wire Harness***

## **Abstrak**

Dapat dilihat perancangan Konveyor *Wire Harness* sangat diperlukan rangkaian kabel yang berfungsi untuk mengalirkan arus listrik dan sinyal pada kendaraan otomotif, *Wire Harness* bertanggung jawab mendistribusikan sumber daya listrik yang bertugas untuk menghubungkan arus dari satu sistem ke sistem lainnya. Sebagai perusahaan yang memproduksi kabel harness untuk mobil, penggunaan Konveyor *Wire Harness* sangat diperlukan. Namun, saat ini *software* yang digunakan untuk mengendalikan Konveyor *Wire Harness* di Perusahaan masih dapat dikembangkan dengan menggunakan *design interface*, pemrograman C#, dan *monitoring* antar beberapa *device*. Pada proses perancangan sistem monitoring dan kontrol pada mesin Konveyor *Wire Harness* dibutuhkannya proses Mengamati dan menganalisa data dari pengujian sistem *monitoring* pada mesin Konveyor *Wire Harness* untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana dan mengidentifikasi masalah yang timbul langsung dapat diatasi secara tepat dan mencapai proses tujuan.

**Kata kunci:** Konveyor, Software, Monitoring

# ***Development of a Control and Monitoring System using C#-Based Visual Studio on a Wire Harness Conveyor Machine***

## ***Abstract***

*It can be seen that the design of the Wire Harness conveyor is very necessary for a series of cables that function to flow electric Current and signals in automotive vehicles, Wire Harnesses are responsible for distributing electrical resources in charge of connecting the Current from one system to another. As a company that produces cable harnesses for cars, the use of Wire Harness conveyors is indispensable. However, Currently the software used to control the Wire Harness conveyor in the company can still be developed using interface design, C# programming, and monitoring between multiple devices. There is a process of designing a monitoring and control system on the Wire Harness Conveyor machine, it is necessary to observe and analyze the data from testing the monitoring software on the Wire Harness Conveyor machine to assess whether the activities carried out are in accordance with the plan and identify problems that arise can be resolved immediately and achieve the goal process.*

***Keywords:*** Conveyor, Software, Monitoring

# Kata Pengantar

Puji Syukur Kehadirat Tuhan Yang Maha Esa atas berkat dan kasih Karunia-Nya serta pertolongan-Nya penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Sistem Kontrol dan *Monitoring* menggunakan Visual Studio Berbasis C# pada Mesin Konveyor *Wire Harness*” yang merupakan syarat dalam menyelesaikan studi untuk menempuh gelar sarjana di jurusan Teknik Elektro, program studi D-IV Teknik Mekatronika, Politeknik Negeri Batam.

Dalam penulisan skripsi ini, penulis menyadari banyak hambatan dan rintangan yang dihadapi . Tetapi, berkat kesabaran dan ketekunan yang disertai dengan doa, bantuan, dukungan, petunjuk, masukan, saran, dan dorongan dari berbagai pihak, sehingga Puji Tuhan skripsi ini dapat diselesaikan dengan baik. Untuk itu, pada kesempatan ini penulis ingin menyampaikan banyak terima kasih kepada semua pihak yang telah mendukung saya dalam proses penyelesaian tugas akhir ini, yaitu:

1. Tuhan Yang Maha Esa atas semua kasih, nikmat dan karunia-Nya.
2. Kepada kedua orang tua penulis, Ayahanda tercinta Herbert Sirait dan Ibunda tersayang Dahliana Tarigan yang tiada henti-hentinya selalu mendoakan penulis, memeberikan dukungan baik secara moril maupun material dan begitu banyak pengorbanan serta kerja keras yang tidak dapat penulis ungkapkan dengan kata-kata, yang orang tua penulis lalui demi keberhasilan dan proses menyelesaikan Tugas Akhir penulis. Kiranya Tuhan Yesus selalu menyertai dan memberkati dalam setiap Langkah Kehidupan-Nya.
3. Kepada adik penulis, yaitu Theresia Septiani Sirait dan Nathan Wifen Sirait yang selalu mendoakan, memberikan semangat dan dukungannya dalam menyelesaikan skripsi ini serta yang selalu memotivasi penulis.
4. Kepada semua keluarga yang tidak sempat penulis sebutkan satu persatu namanya yang telah memberikan dukungan dan doa, serta memberikan semangat dan membantu penulis selama penyusunan skripsi ini.
5. Kepada pemilik NIM 4523091045 yang telah memberikan dukungan dan doa, serta memberikan semangat dan membantu penulis selama penyusunan skripsi ini.
6. Bapak Uuf Brajawidagda., Ph.D. selaku Direktur Politeknik Negeri Batam
7. Bapak Dr. Budi Sugandi, S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Batam.

8. Bapak Indra Hardian Mulyadi, S.T., M. Eng. selaku Ketua Program Studi Teknik Mekatronika Politeknik Negeri Batam sekaligus dosen penguji penulis.
9. Bapak Sumantri Kurniawan Risandriya, S.T., M.T. selaku dosen wali penulis.
10. Bapak Diono, S.Tr.T., M.Sc. selaku dosen pembimbing penulis yang telah menyediakan waktu untuk mengarahkan penulis dalam menyelesaikan tugas akhir ini.
11. Bapak Adlian Jefiza, S.Pd., M.T. selaku dosen wali sekaligus dosen pembimbing penulis yang telah bersedia meluangkan waktu untuk mengarahkan penulis dalam proses penyelesaian tugas akhir ini.
12. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah tugas akhir.
13. Bapak Daniel Sutopo Pamungkas, S.T., M.T., Ph.D selaku dosen penguji penulis.
14. Bapak Fadhli Firdaus selaku dosen penguji penulis.
15. Seluruh staf pengajar, laboran, dan administrasi di jurusan Teknik Elektro Politeknik Negeri Batam.
16. Teman-teman seperjuanganku, Yoel Cristnadi Ompusunggu, Andreas Prasetyo Dowo Runon, Ryo Rizky Nugroho, dan Albertus Marsel Pirindia Situmorang, yang sama-sama berjuang dari awal semester hingga sampai saat ini yang selalu membantu dan menemani penulis dalam melakukan penelitian serta berjuang bersama-sama dalam menyusun skripsi masing-masing.
17. Seluruh rekan - rekan Teknik Mekatronika angkatan 2020 yang telah membantu dan mendukung penulis.

Penulis menyadari bahwa dalam penulisan skripsi ini masih banyak kekurangan dan kekeliruan serta masih jauh dari kata kesempurnaan dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan dan bahkan kritik yang sifatnya membangun dari berbagai pihak. Semoga skripsi ini dapat bermanfaat bagi semua pihak. Demikian ucapan terima kasih penulis sampaikan. Tuhan Yesus memberkati kita semua.

**“Serahkanlah perbuatanmu kepada TUHAN, maka terlaksanalah segala rencanamu.”** (Bdk. Amsal 16 : 3)

Batam, 26 Juli 2024



Herdi Gunanta Sirait

# Daftar Isi

Pernyataan Keaslian Tugas Akhir .....	<b>Error! Bookmark not defined.</b>
Lembar Pengesahan .....	<b>Error! Bookmark not defined.</b>
Abstrak.....	iii
<i>Abstract</i> .....	iv
Kata Pengantar .....	v
Daftar Isi.....	vii
Daftar Gambar.....	xi
Daftar Tabel.....	xii
Bab 1. Pendahuluan.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Tujuan.....	2
1.4. Manfaat.....	2
1.5. Batasan.....	2
Bab 2. Tinjauan Pustaka .....	3
2.1. Penelitian Terdahulu .....	3
2.2. Dasar Teori.....	4
2.2.1. Sistem <i>Monitoring</i> .....	4
2.2.2. Sistem Kontrol .....	5
2.2.3. Mesin <i>Conveyor Wire Harness</i> .....	5
2.2.4. Visual Studio 2010 .....	6
2.2.5. Pemrograman <i>C#</i> .....	7
2.2.6. XAMPP .....	7
2.2.7. phpMyAdmin.....	8
Bab 3. Metodologi Penelitian.....	9
3.1. Perancangan.....	9
3.1.1. Perancangan <i>Design</i> .....	10
3.1.2. Perancangan Program .....	10

3.1.3. Perancangan Software .....	11
3.1.4. Perancangan Sistem Kerja.....	12
3.1.5. Perancangan Pembuatan <i>Database</i> .....	14
3.2. Alat dan Bahan .....	15
3.3. Pengujian.....	15
3.3.1. Pembacaan data dengan menggunakan <i>interface</i> .....	16
3.3.2. Pengambilan data rpm Motor Dengan <i>Tachometer</i> .....	17
3.3.3. Perbandingan <i>Current</i> pada Interface dan Multimeter .....	18
Bab 4. Hasil dan Pembahasan .....	26
4.1. Data Hasil Penelitian .....	26
4.4.1. Hasil Pembacaan data motor pada <i>Interface</i> .....	26
4.4.2. Hasil Pengambilan Data Motor Dengan <i>Tachometer</i> .....	29
4.4.3. Hasil Perbandingan <i>Current</i> pada <i>Interface</i> dan Multimeter .....	32
4.2. Pembahasan.....	35
4.2.1. Rancangan Monitoring dan Kontrol Mesin <i>Konveyor Wire Harness</i> Menggunakan <i>Visual Studio</i> Berbasis C#.....	35
4.2.2. Pembacaan Data Motor Pada <i>Interface</i> .....	36
4.4.3. Pengambilan Data Motor Dengan <i>Tachometer</i> .....	37
4.2.4. Perbandingan <i>Current</i> pada <i>Interface</i> dan Multimeter .....	37
Bab 5. Kesimpulan dan Saran.....	39
5.1. Kesimpulan.....	39
5.2. Saran.....	39
Daftar Pustaka .....	41
Biodata.....	43
Lampiran.....	44
Lampiran 1 : Design Interface.....	44
Tampilan <i>Login</i> .....	44
Tampilan Menu <i>Home</i> .....	45
Menu <i>PLC</i> .....	45
Menu Data <i>Product</i> .....	46

Menu <i>Product OK</i> .....	46
Menu <i>Product Reject</i> .....	47
Monitoring .....	47
Lampiran 2 : Bahasa Pemograman C# .....	48
C# Tampilan <i>Login</i> .....	48
C# Tampilan Menu <i>Home</i> .....	49
C# Menu PLC.....	50
C# Menu Data Product .....	54
C# Product OK.....	57
C# Product Reject.....	61
C# Monitoring .....	65
Lampiran 3 : <i>Software to Database</i> .....	68
XAMPP Control Panel .....	68
Localhost phpMyAdmin .....	68
Lampiran 4 : Pembacaan data motor pada <i>Interface</i> .....	69
<i>Frequency 5 Hz</i> .....	69
<i>Frequency 10 Hz</i> .....	69
<i>Frequency 15 Hz</i> .....	70
<i>Frequency 20 Hz</i> .....	71
<i>Frequency 25 Hz</i> .....	71
<i>Frequency 30 Hz</i> .....	72
Lampiran 5 : Pengambilan data rpm motor pada <i>tachometer</i> .....	73
Frequency 5 Hz.....	73
Frequency 10 Hz.....	73
Frequency 15 Hz .....	74
Frequency 20 Hz.....	76
Frequency 25 Hz.....	76
Frequency 30 Hz.....	77
Lampiran 6 : Perbandingan Current pada Interface dan Multimeter.....	78

<i>Frequency 5 Hz</i> .....	78
<i>Frequency 10 Hz</i> .....	78
<i>Frequency 15 Hz</i> .....	79
<i>Frequency 20 Hz</i> .....	80
<i>Frequency 25 Hz</i> .....	80
<i>Frequency 30 Hz</i> .....	81

## Daftar Gambar

Gambar 1. Tampilan Monitoring .....	5
Gambar 2. Tampilan Sistem Kontrol .....	5
Gambar 3. Mesin Konveyor Wire Harness.....	6
Gambar 4. Visual Studio 2010.....	7
Gambar 5. Tampilan Pemograman C#.....	7
Gambar 6. XAMPP.....	8
Gambar 7. <i>phpMyAdmin</i> .....	8
Gambar 8. Flowchart Perancangan Software .....	9
Gambar 9. Perancangan <i>Design</i> .....	10
Gambar 10. Perancangan Program.....	11
Gambar 11. Pengujian Software Tampilan Login.....	12
Gambar 12 <i>Flowchart</i> Sistem Kerja.....	13
Gambar 13. <i>Flowchart DataBase</i> .....	14
Gambar 14. Pembacaan data dengan menggunakan <i>interface</i> .....	16
Gambar 15. Pengambilan data rpm Motor Dengan <i>Tachometer</i> .....	17
Gambar 16. <i>Current</i> pada <i>interface</i> .....	18
Gambar 17. <i>Current</i> pada Multimeter.....	18
Gambar 18. <i>Current</i> pada <i>interface</i> .....	19
Gambar 19. <i>Current</i> pada Multimeter.....	19
Gambar 20. <i>Current</i> pada <i>interface</i> .....	20
Gambar 21. <i>Current</i> pada Multimeter.....	21
Gambar 22. <i>Current</i> pada <i>interface</i> .....	22
Gambar 23. <i>Current</i> pada Multimeter.....	22
Gambar 24. <i>Current</i> pada <i>interface</i> .....	23
Gambar 25. <i>Current</i> pada Multimeter.....	23
Gambar 26. <i>Current</i> pada <i>interface</i> .....	24
Gambar 27. <i>Current</i> pada Multimeter.....	25

## Daftar Tabel

Tabel 1. Alat dan Bahan.....	15
Tabel 2. Pembacaan data motor 5 Hz .....	26
Tabel 3. Pembacaan data motor 10 Hz .....	26
Tabel 4. Pembacaan data motor 15 Hz .....	27
Tabel 5. Pembacaan data motor 20 Hz .....	27
Tabel 6. Pembacaan data motor 25 Hz .....	28
Tabel 7. Pembacaan data motor 30 Hz .....	28
Tabel 8. Pengambilan Data Motor 5 Hz .....	29
Tabel 9. Pengambilan Data Motor 10 Hz .....	29
Tabel 10. Pengambilan Data Motor 15 Hz .....	30
Tabel 11. Pengambilan Data Motor 20 Hz .....	30
Tabel 12. Pengambilan Data Motor 25 Hz .....	31
Tabel 13. Pengambilan Data Motor 30 Hz .....	31
Tabel 14. Perbandingan <i>Current</i> 5 Hz.....	32
Tabel 15. Perbandingan <i>Current</i> 10 Hz.....	32
Tabel 16. Perbandingan <i>Current</i> 15 Hz.....	33
Tabel 17. Perbandingan <i>Current</i> 20 Hz.....	33
Tabel 18. Perbandingan <i>Current</i> 25 Hz.....	34
Tabel 19. Perbandingan <i>Current</i> 30 Hz.....	34

# Bab 1. Pendahuluan

## 1.1. Latar Belakang

Konveyor merupakan sebuah mesin yang sangat dibutuhkan dalam industri, terutama untuk kemudahan transportasi dan distribusi barang serta bahan. Mengingat Konveyor merupakan penggerak utama dalam proses industri, otomatisasi dalam pengoperasian Konveyor sangatlah diperlukan untuk peningkatan efisiensi dan produktivitas perusahaan [1].

*Wire Harness* merupakan kumpulan dari rangkaian kabel yang berfungsi untuk mengalirkan arus listrik dan sinyal pada kendaraan otomotif [2], *Wire Harness* bertanggung jawab mendistribusikan sumber daya listrik [3] yang bertugas untuk menghubungkan arus dari satu sistem ke sistem lainnya [4].

Salah satu perusahaan yang menggunakan Konveyor *Wire Harness* sebagai penggerak utama untuk menunjang produktivitas perusahaan adalah PT. Sumitomo *Wiring System*. Sebagai perusahaan yang memproduksi kabel harness untuk mobil, penggunaan Konveyor *Wire Harness* sangat diperlukan. Namun, saat ini *software* yang digunakan untuk mengendalikan Konveyor *Wire Harness* di PT tersebut masih dapat dikembangkan.

Salah satu masalah yang dihadapi PT Sumitomo *Wiring System* adalah *software* yang digunakan untuk memonitor dan mengontrol mesin di lane produksi masih sangat tua dan hanya menampilkan tiga bagian: data produk, total produk, dan produk jadi.

*Software* yang digunakan masih dapat ditingkatkan terutama pada tampilan menu input dan *monitoring*. Mulai dari design *interface*, pemrograman *C#*, dan *monitoring* antar beberapa *device*.

Dari permasalahan yang ada diatas maka saya mengangkat judul "Pengembangan *Software* Kontrol dan *Monitoring* menggunakan *Visual Studio* Berbasis *C#* pada Mesin Konveyor *Wire Harness*"

## 1.2. Rumusan Masalah

Dalam Pengembangan *Software* Kontrol dan *Monitoring* menggunakan *Visual Studio* Berbasis *C#* pada Mesin Konveyor *Wire Harness*, terdapat beberapa permasalahan yang perlu dipecahkan, antara lain:

1. Bagaimana merancang *monitoring* dan kontrol pada mesin Konveyor *Wire Harness* menggunakan *visual studio* berbasis *C#*?
2. Bagaimana merakitulasi hasil uji coba *Software monitoring* pada mesin Konveyor *Wire Harness* menggunakan *visual studio* berbasis *C#*?

### 1.3. Tujuan

Tujuan dari Pengembangan *Software* Kontrol dan *Monitoring* menggunakan Visual Studio Berbasis *C#* pada Mesin Konveyor *Wire Harness* adalah:

1. Merancang sistem *monitoring* dan kontrol pada mesin Konveyor *Wire Harness*
2. Mengamati dan menganalisa data dari pengujian *Software monitoring* pada mesin Konveyor *Wire Harness*

### 1.4. Manfaat

Manfaat dari Pengembangan Kontrol dan *Monitoring* menggunakan Visual Studio Berbasis *C#* pada Mesin Konveyor *Wire Harness* adalah

1. Memudahkan operator dalam mengolah data untuk pelaporan di *lane* produksi
2. Memudahkan pengguna dalam memantau hasil dan proses produksi setiap *lane* melalui komputer

### 1.5. Batasan

Dalam Pengembangan Kontrol dan *Monitoring* menggunakan *Visual Studio* Berbasis *C#* pada Mesin Konveyor *Wire Harness*, terdapat beberapa batasan yang perlu diperhatikan. Beberapa batasan tersebut adalah:

1. Perancangan sistem *monitoring* dan kontrol pada mesin Konveyor *Wire Harness* menggunakan *visual studio* berbasis *C#*
2. Hasil pengujian dari *Software monitoring* dengan menggunakan tampilan *visual studio* berbasis *C#*

## Bab 2. Tinjauan Pustaka

### 2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Anantha dkk pada tahun 2021 yang berjudul “Mempelajari Proses Produksi *Wire Harness* Dan Studi Kasus Serta Perawatan Pada Mesin *Cheker* KCM Di CV.HIJ” menunjukkan informasi tentang proses produksi *wire harness* dan studi kasus perawatan mesin *cheker* KCM di CV. HIJ. Mesin *cheker* KCM adalah alat yang digunakan untuk pengecekan arus listrik dan sinyal, yang merupakan tahapan yang sangat penting dalam proses pembuatan *wire harness*. Jurnal ini mencakup pengembangan sistem kontrol dan monitoring menggunakan *Visual Studio* Berbasis C#, sebagai alat untuk mengelola dan mengontrol proses produksi *wire harness*. [5]

Penelitian yang dilakukan oleh Mohammad Dwi Cahyo dkk pada tahun 2018 yang berjudul “Rancang Bangun Sistem Proteksi Dan Monitoring Energi Listrik Berbasis Mikrokontroler Menggunakan *Visual Studio* Uji Coba Diakses Di PT. Pancawana Indonesia” menunjukkan sistem proteksi dan monitoring energi listrik digunakan untuk melindungi dan memantau penggunaan energi listrik oleh suatu beban. Pemantauan energi listrik sangat penting dalam industri untuk mengontrol penggunaan energi dan menghitung biaya yang terkait. Alat ini menggunakan berbagai komponen seperti *transformator stepdown* untuk mengukur tegangan sumber, sensor arus CT untuk mengukur arus beban, dan mikrokontroler Atmega 16 untuk mengolah data dan menampilkan hasilnya pada LCD. Data juga dapat ditransmisikan ke komputer *server* melalui komunikasi RS-485. Sistem ini dapat bekerja sesuai yang diharapkan dan memberikan perlindungan yang telah ditentukan. Pengujian modul tergantung pada pasokan daya yang baik dan sistem kelistrikan yang handal, sementara aplikasi *monitoring* tergantung pada PC atau Laptop yang berfungsi dengan baik. [6]

Penelitian yang dilakukan oleh Sumardi Sadi pada tahun 2012 yang berjudul “Sistem Pengendali Konveyor *Belt* Pada PT. XYZ Tangerang” menunjukkan hasil Sistem kendali *conveyor belt* di PT. XYZ digunakan untuk memindahkan produk dari bagian produksi ke gudang penyimpanan, meningkatkan efektivitas dan efisiensi perusahaan. Melalui pengamatan dan studi terhadap sistem kontrolnya, ditemukan bahwa *conveyor belt* tersebut mampu membawa sekitar 600 dus produk dalam satu hari pada kecepatan inverter 28-32 Rpm, dibandingkan dengan pembawaan manual yang membutuhkan waktu dan tenaga yang lebih besar. Dalam pembuatan *conveyor belt*, penting untuk memperhatikan penempatan sensor, menyesuaikan kekuatan motor dengan *gearbox*, dan memastikan penggunaan inverter disesuaikan dengan kondisi motor dan *gearbox*nya. Dengan demikian, penggunaan konveyor dapat meningkatkan efisiensi proses produksi dan distribusi produk. [7]

Penelitian yang dilakukan oleh Nabila dkk pada tahun 2022 yang berjudul "*Control and Monitoring System Process Handling Production on SMI 4.0 Machines using PLC Controller Wirelessly Based on Human Machine Interface*" Penelitian ini menunjukkan bahwa kolaborasi antara HMI dan mesin produksi dapat bekerja otomatis sesuai dengan instruksi yang diinginkan, meningkatkan kinerja efektif dan mengatasi beberapa masalah yang terkait dengan proses produksi. Data yang dihasilkan oleh sistem ini juga memberikan panduan untuk menentukan target output yang dibutuhkan, mempercepat analisis *troubleshoot*, dan memastikan keberhasilan dalam proses produksi secara keseluruhan.[8]

Penelitian yang saya lakukan yang berjudul "Pengembangan Sistem Kontrol dan *Monitoring* menggunakan *Visual Studio* Berbasis *C#* pada Mesin Konveyor *Wire Harness*", hasil penelitiannya adalah Membantu sasaran sistem yang sebelumnya, merancang *monitoring* dan kontrol untuk menampilkan hasil output yang saat ini digunakan untuk mencapai tujuan yang sesuai. Metode pengembangan yang saya gunakan adalah pengembangan Sistem kontrol dan *monitoring* berbasis *C#*.

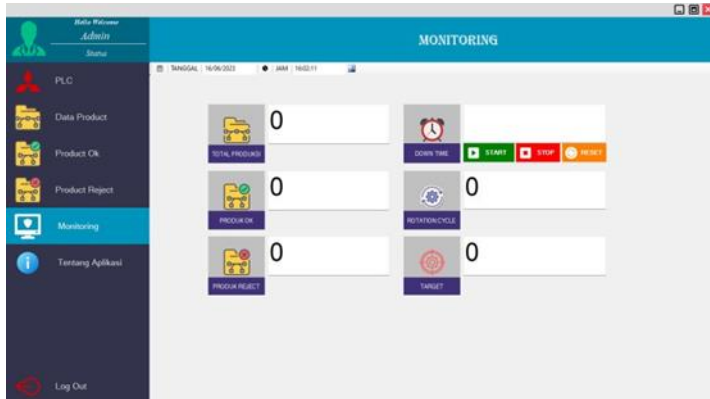
Pembedaannya pada penelitian saat ini dengan sebelumnya adalah Penelitian saat ini membangun sistem kontrol dan monitoring yang berbasis *C#*, Sedangkan Penelitian sebelumnya hanya dapat menampilkan hasil *output* dari suatu mesin. Persamaan pada penelitian sebelumnya merupakan penelitian yang di lakukan sama-sama membangun sistem *monitoring* tetapi hanya dapat menampilkan 2 *output* saja.

Sedangkan sistem yang saya kembangkan saat ini terdapat beberapa menu lainnya yaitu menu *PLC*, data produk, produk ok, produk *reject*, dan *monitoring* yang digunakan sebagai sistem kontrol dan monitoring yang dikembangkan dengan berbasis *C#* dan juga dikembangkan dengan *phpMyAdmin* untuk mengolah data dengan metode monitoring data dan kontrol untuk menampilkan pembacaan data produk dan kontrol motor

## **2.2. Dasar Teori**

### **2.2.1. Sistem *Monitoring***

*Monitoring* adalah merupakan sebuah kegiatan untuk menjamin akan tercapainya semua tujuan organisasi dan manajemen [9]. *Monitoring* juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan [10].



**Gambar 1. Tampilan Monitoring**

### 2.2.2. Sistem Kontrol

Sistem kendali atau sistem kontrol adalah suatu sistem yang menghasilkan nilai tertentu sebagai keluarannya melalui pengendalian ataupun perubahan ketentuan dari masukan sistem [11].



**Gambar 2. Tampilan Sistem Kontrol**

### 2.2.3. Mesin Conveyor Wire Harness

Konveyor merupakan sebuah mesin yang sangat dibutuhkan dalam industri, terutama untuk kemudahan transportasi dan distribusi barang serta bahan. Mengingat Konveyor merupakan penggerak utama dalam proses industri,

otomatisasi dalam pengoperasian Konveyor sangatlah diperlukan untuk peningkatan efisiensi dan produktivitas perusahaan [1].

*Wire Harness* merupakan kumpulan dari rangkaian kabel yang berfungsi untuk mengalirkan arus listrik dan sinyal pada kendaraan otomotif [2], *Wire Harness* bertanggung jawab mendistribusikan sumber daya listrik [3] yang bertugas untuk menghubungkan arus dari satu sistem ke sistem lainnya [4]

Jadi kesimpulannya, Konveyor *Wire Harness* adalah mesin yang digunakan untuk membantu proses produksi kabel



**Gambar 3. Mesin Konveyor *Wire Harness***

**<https://thumbs.dreamstime.com/b/production-room-stands-manufacturing-electric-wiring-cars-modern-plant-232310364.jpg>**

#### **2.2.4. Visual Studio 2010**

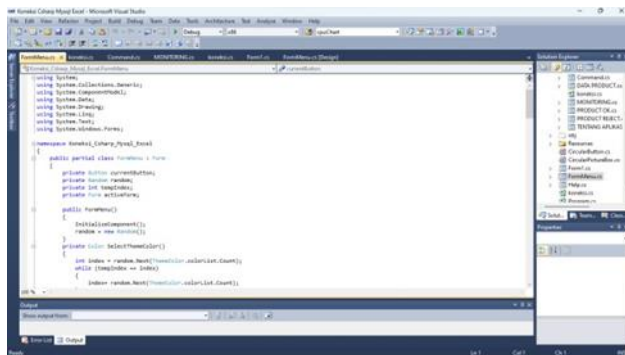
*Visual Studio 2010* pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah- perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas- tugas tertentu. *Visual Studio 2010* (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis *windows*. [12]



Gambar 4. Visual Studio 2010

### 2.2.5. Pemrograman C#

C# adalah bahasa pemrograman berbasis objek yang dibuat Microsoft yang menggabungkan kekuatan bahasa pemrograman C++ dan bahasa pemrograman Visual Basic. C# merupakan bahasa pemrograman yang berbasis C++ dan mengandung fitur-fitur yang serupa dengan bahasa pemrograman Java [13].



Gambar 5. Tampilan Pemrograman C#

### 2.2.6. XAMPP

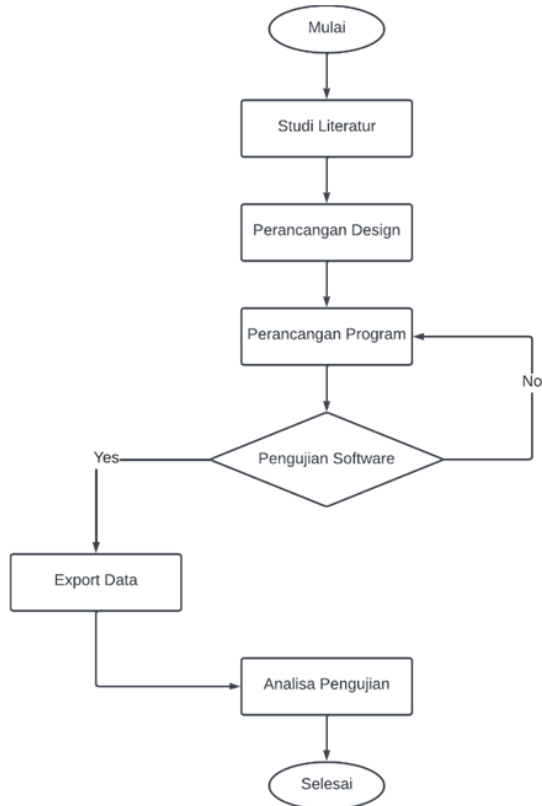
Menurut Bunafit Nugroho (2011:1), "XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh komunitas *Open Source*, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program



## Bab 3. Metodologi Penelitian

### 3.1. Perancangan

Tahapan perancangan pengembangan *Software* Kontrol dan *Monitoring* Menggunakan *Visual Studio* Berbasis *C#* pada Mesin Konveyor *Wire Harness* dengan beberapa tahapan yang bisa kita lihat pada gambar dibawah ini



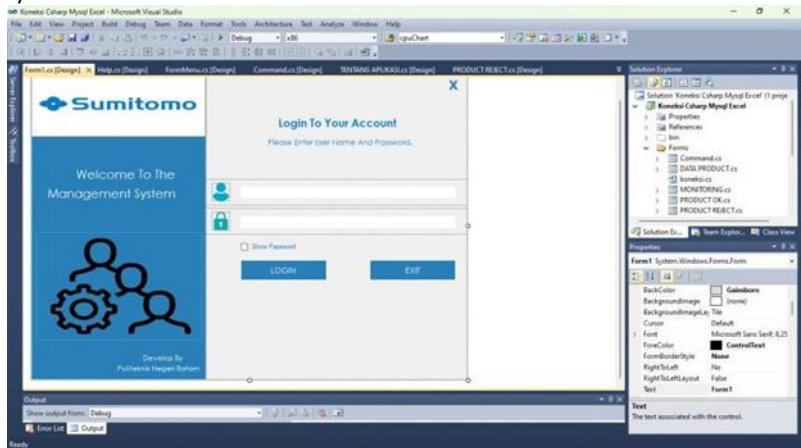
**Gambar 8. Flowchart Perancangan Software**

yang dilakukan Pada gambar 8 adalah melakukan studi literatur, lalu melakukan perancangan *design*, Setelah perancangan *design* akan dllanjutkan dengan perancangan porgram.Ketika selesai melakukan perancangan program, pengguna melakukan penguian *software* yang telah dibuat. Jika penguian tidak

berhasil maka melakukan perancangan program kembali, tetapi jika pengujian berhasil maka melanjutkan pengiriman data *monitoring* ke microsoft excel, setelah pengiriman data berhasil tahap akhir dilakukan analisa pengujian.

### 3.1.1. Perancangan *Design*

Berikut adalah perancangan *design* kontrol dan *monitoring* pada mesin koveyor *wire harness*

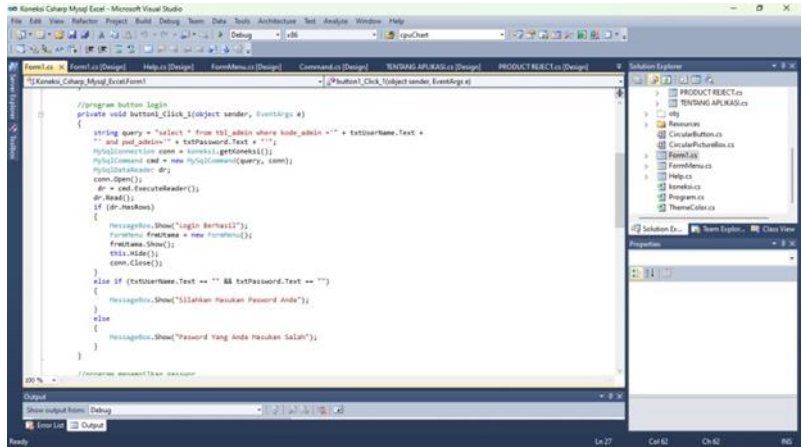


Gambar 9. Perancangan *Design*

Alasan saya menggunakan design *interface* dengan *visual studio* yaitu karena dengan menggunakan *visual studio* mempunyai kontrol *library* yang lengkap untuk membuat *interface*, *visual studio* juga mendukung sinkronisasi *database* dan dapat diintegrasikan oleh *microsoft* lain seperti *exel* dan lainnya

### 3.1.2. Perancangan Program

Berikut adalah perancangan program kontrol dan *monitoring* pada mesin konveyor *wire harness*

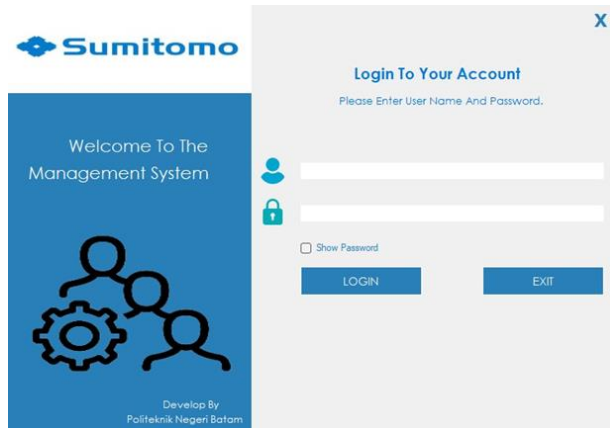


Gambar 10. Perancangan Program

Alasan saya menggunakan bahasa pemrograman *C#* yaitu lebih *fleksibel*, efisien dan lebih sederhana dalam menghilangkan hal hal yg rumit juga mempunyai bahasa pemrograman yang *modern*.

### 3.1.3. Perancangan *Software*

Berikut adalah pengujian *software* pada mesin konveyor *wire harness* berhasil dibuat dan dapat di gunakan

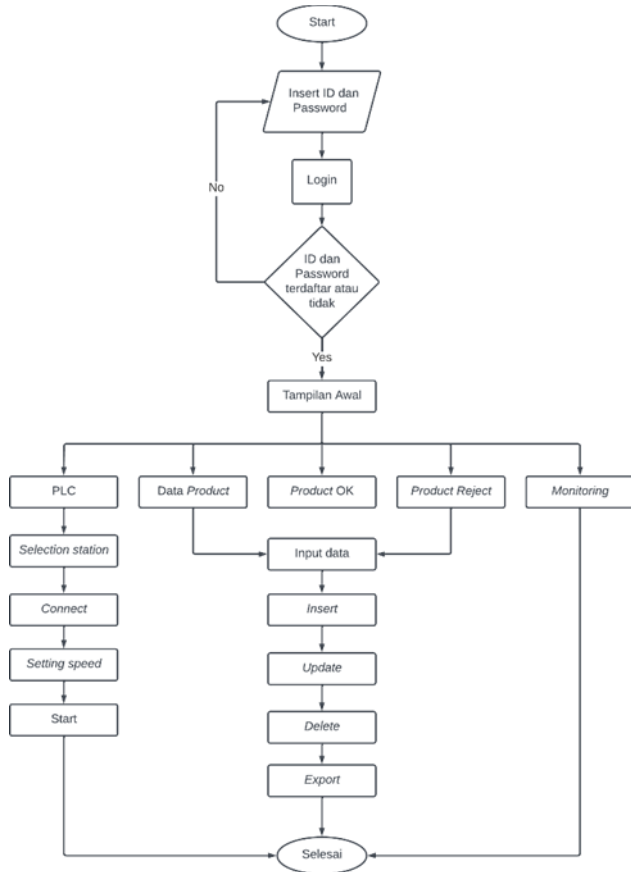


**Gambar 11. Pengujian Software Tampilan Login**

Alasan saya menggunakan tampilan *software* karena dari PT.Sumitomo *Wiring System* masih menggunakan *software* yang kelihatan lampau dan saya disini mengembangkan menjadi lebih modern supaya lebih layak pakai dengan jangka waktu yang panjang

#### **3.1.4. Perancangan Sistem Kerja**

Proses *flowchart* adalah suatu metode yang digunakan untuk menyajikan dalam bentuk bagan alur atau sekuensi aktual serangkaian peristiwa dalam proses atau layanan yang diberikan. Untuk melakukan koneksi dari perangkat computer ke PLC dan dihubungkan motor 3 fasa, berikut adalah alur *flowchart*nya:



**Gambar 12 Flowchart Sistem Kerja**

Hal pertama yang dilakukan pada gambar 12 adalah memasukkan id dan *password* lalu tekan tombol login jika tidak berhasil login kembali memasukkan ulang id dan *password*, tetapi jika berhasil kita langsung masuk ke aplikasi dengan tampilan *home*. Dari tampilan menu *home* ada beberapa menu yaitu *PLC*, *Data Product*, *Product Ok*, *Product Reject*, dan *Monitoring*.

Untuk di menu *PLC* kita dapat memilih *station 1* atau *2*, untuk *station 1* digunakan untuk melakukan simulasi dan *station 2* untuk menghubungkan kepanel dan mengontrol motor. Setelah memilih *station* lalu tekan tombol

### 3.1.5. Perancangan Pembuatan *Database*

Untuk bagian pemograman sistem *database* dimulai dengan login ke *Software* yang sudah dibuat. Berikut adalah alur *flowchart database* yang telah dibuat untuk menampilkan hasil *database* :



Gambar 13. *Flowchart DataBase*

pada gambar 13 yang dilakukan adalah memasukkan id dan *password* lalu tekan tombol login jika tidak berhasil login kembali memasukkan ulang id dan *password*, tetapi jika berhasil kita langsung masuk ke aplikasi dengan tampilan *home*. Dari tampilan menu *home* pengguna bisa langsung ke menu Data *Product*, *Product Ok*, *Product Reject*. Pada menu Data *Product*, *Product Ok*, *Product Reject* dimana mempunyai proses yang sama dengan menginput data lalu tekan insert untuk menampilkan data ditabel. Jika ingin merubah data pengguna dapat menekan tombol *update*, dan jika data yang dimasukkan salah bisa menekan tombol delete, setelah data lengkap dapat kita *export* ke excel untuk laporan *output* dari setiap *lane*.

### 3.2. Alat dan Bahan

Pada tahap ini merupakan rincian *Software* dan bahan yang akan digunakan pada perancangan Pengembangan *Software* Kontrol dan *Monitoring* menggunakan Visual Studio Berbasis *C#* pada Mesin Konveyor *Wire Harness*. Dalam rinciannya juga terdapat estimasi biaya pada *Software* dan bahan yang akan digunakan

**Tabel 1. Alat dan Bahan**

No.	Alat	Jumlah	Total (Rp.)	Keterangan
1	Monitor PC	1 pc	0	MF
2	<i>Keyboard</i>	1 pc	0	MF
3	<i>Mouse</i>	1 pc	0	MF
4	<i>Tachometer</i>	1 pc	0	Sewa/Pinjam

### 3.3. Pengujian

Berikut merupakan hasil pengujian yang sudah dilakukan setelah semua proses perancangan sistem kontrol dan monitoring telah dibuat, maka akan dilakukan pengujian untuk mengetahui apakah kontrol dan *monitoring* menggunakan visual studio berbasis *C#* berfungsi.

### 3.3.1. Pembacaan data dengan menggunakan *interface*

Berikut adalah pembacaan data motor pada tampilan *interface* yang telah dibuat pada *frequency* 5 Hz - 30 Hz

00:01:00	00:01:00	00:01:00
<b>Read Data Motor</b> Frequency : 5 Current : 0,47 Voltage : 3,33 Power : 1,5651 RPM : 150	<b>Read Data Motor</b> Frequency : 10 Current : 0,44 Voltage : 5,21 Power : 2,2924 RPM : 300	<b>Read Data Motor</b> Frequency : 15 Current : 0,4 Voltage : 6,98 Power : 2,792 RPM : 450
00:01:00	00:01:00	00:01:00
<b>Read Data Motor</b> Frequency : 20 Current : 0,39 Voltage : 8,86 Power : 3,4554 RPM : 600	<b>Read Data Motor</b> Frequency : 25 Current : 0,38 Voltage : 10,68 Power : 4,0584 RPM : 750	<b>Read Data Motor</b> Frequency : 30 Current : 0,37 Voltage : 12,64 Power : 4,6768 RPM : 900

**Gambar 14.** Pembacaan data dengan menggunakan *interface*

### 3.3.2. Pengambilan data rpm Motor Dengan *Tachometer*

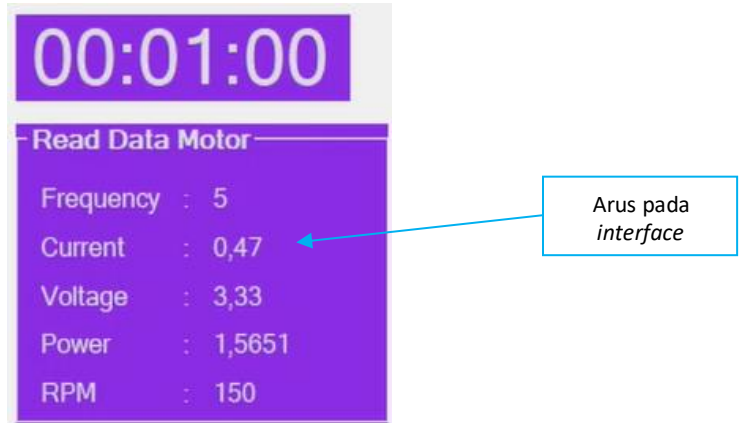
Berikut adalah pengambilan data rpm motor dengan *Tachometer* dengan frequency 5 Hz – 30 Hz



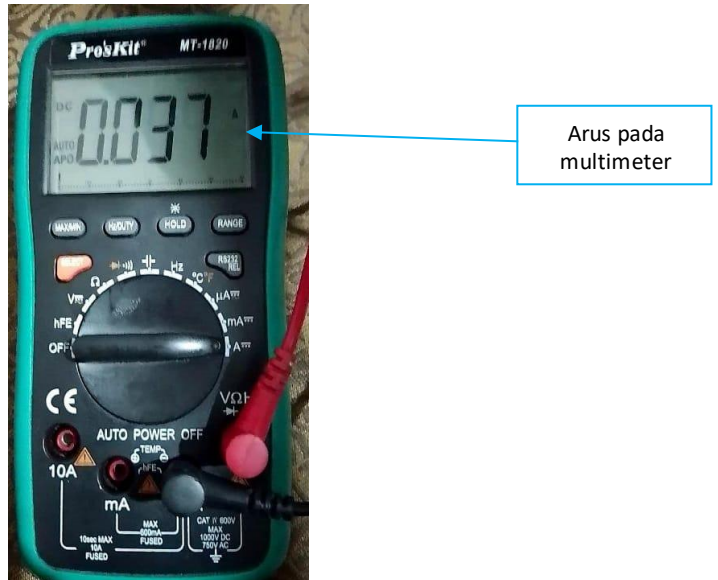
Gambar 15. Pengambilan data rpm Motor Dengan *Tachometer*

### 3.3.3. Perbandingan *Current* pada *Interface* dan Multimeter

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 5 Hz



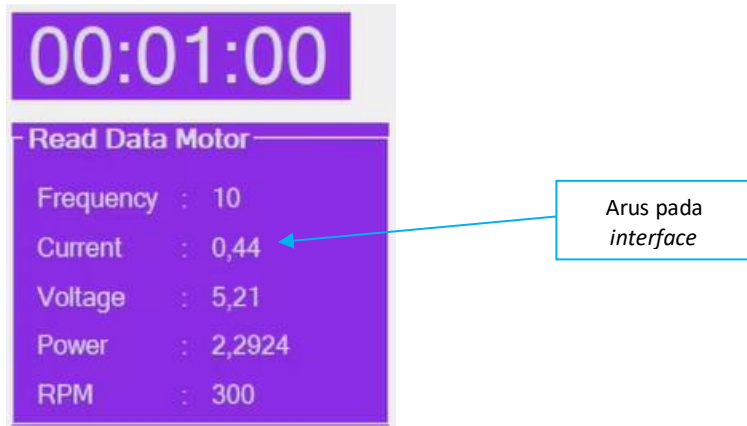
Gambar 16. *Current* pada *interface*



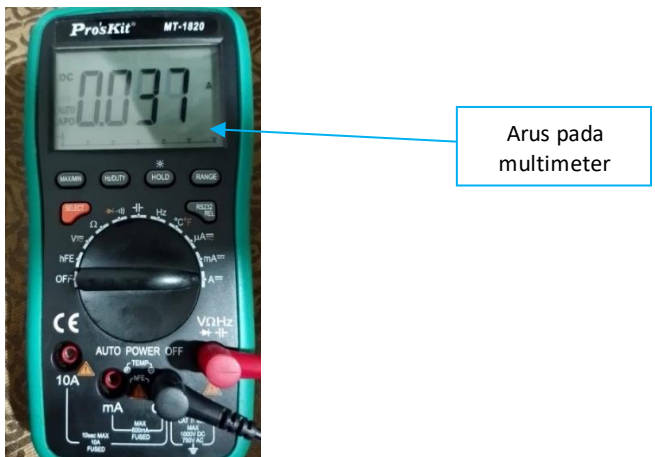
Gambar 17. *Current* pada Multimeter

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 5 Hz ialah 0,10 A

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 10 Hz



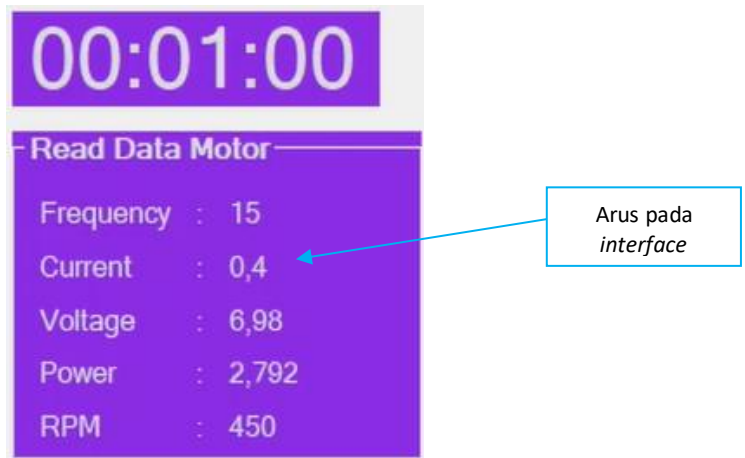
Gambar 18. *Current* pada *interface*



Gambar 19. *Current* pada Multimeter

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 10 Hz ialah 0,07 A

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 15 Hz



**Gambar 20. *Current* pada *interface***

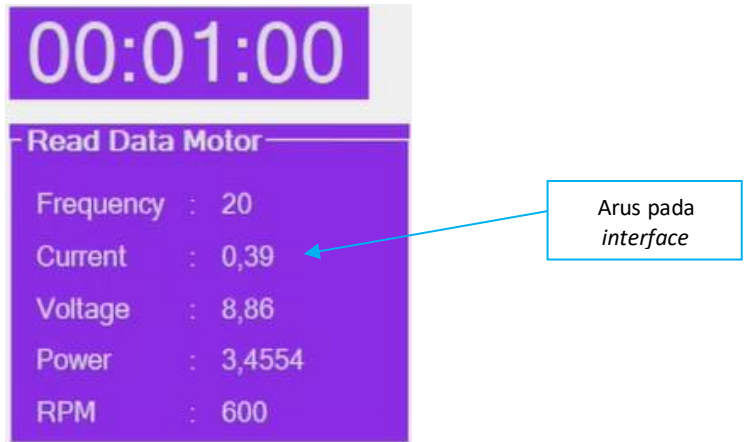


Arus pada multimeter

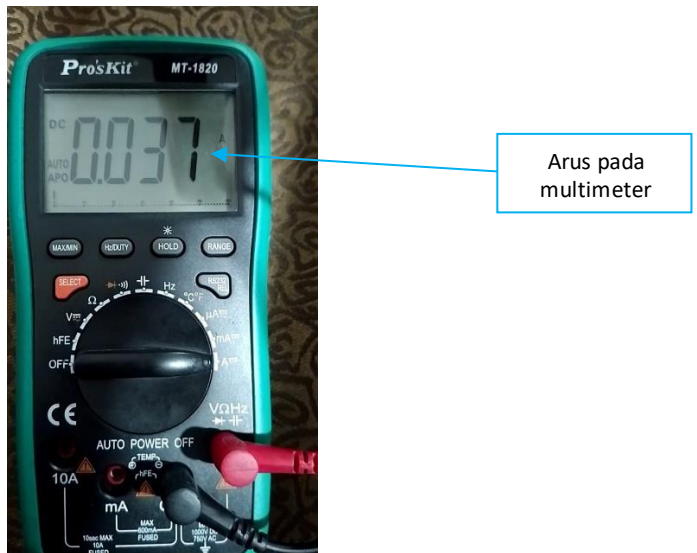
**Gambar 21. Current pada Multimeter**

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 15 Hz ialah 0,03 A

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 20 Hz



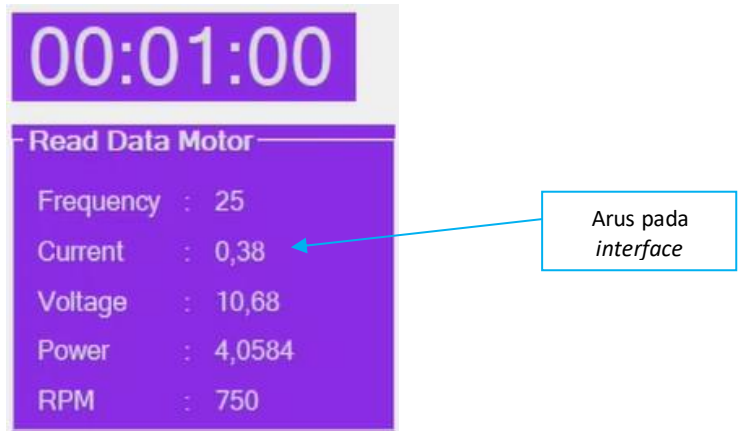
Gambar 22. *Current* pada *interface*



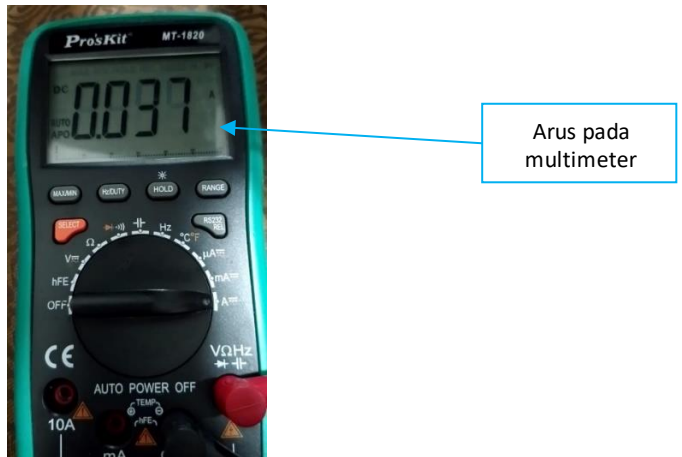
Gambar 23. *Current* pada Multimeter

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 20 Hz ialah 0,02 A

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 25 Hz



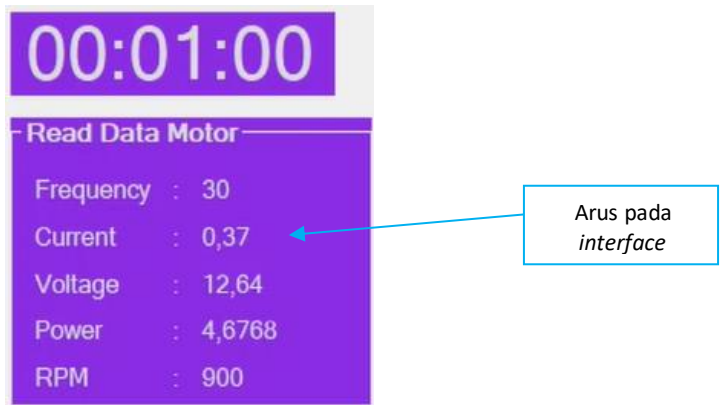
Gambar 24. *Current* pada interface



Gambar 25. *Current* pada Multimeter

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 25 Hz ialah 0,01 A

Berikut adalah perbandingan *Current* pada *interface* dan multimeter dengan *frequency* 30 Hz



**Gambar 26. *Current* pada *interface***



Arus pada multimeter

**Gambar 27. Current pada Multimeter**

Pada gambar diatas merupakan perbandingan *Current* pada *nterface* dan multimeter dimana tidak memiliki perbedan selisih data pada saat pengukuran dengan frekuensi 30 Hz

## Bab 4. Hasil dan Pembahasan

### 4.1. Data Hasil Penelitian

#### 4.4.1. Hasil Pembacaan data motor pada *Interface*

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *frequency* 5 Hz.

**Tabel 2. Pembacaan data motor 5 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
1	5hz	0,47	3,33	1,5651	150
2	5hz	0,47	3,32	1,5604	150
3	5hz	0,47	3,34	1,5698	150
4	5hz	0,47	3,32	1,5604	150
5	5hz	0,47	3,33	1,5651	150

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *frequency* 10 Hz.

**Tabel 3. Pembacaan data motor 10 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
1	10hz	0,44	5,21	2,2924	300
2	10hz	0,44	5,23	2,2489	300
3	10hz	0,44	5,24	2,3056	300
4	10hz	0,44	5,22	2,2968	300
5	10hz	0,44	5,23	2,2489	300

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *Frequency* 15 Hz.

**Tabel 4. Pembacaan data motor 15 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
1	15hz	0,4	6,99	2,796	450
2	15hz	0,4	7,02	2,808	450
3	15hz	0,4	6,98	2,792	450
4	15hz	0,4	7	2,8	450
5	15hz	0,4	7,01	2,804	450

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *frequency* 20 Hz.

**Tabel 5. Pembacaan data motor 20 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>Voltage</i>	<i>power</i>	<i>rpm</i>
1	20hz	0,39	8,84	3,3592	600
2	20hz	0,39	8,83	3,4437	600
3	20hz	0,39	8,86	3,3668	600
4	20hz	0,39	8,82	3,4398	600
5	20hz	0,39	8,8	3,432	600

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *frequency* 25 Hz.

**Tabel 6. Pembacaan data motor 25 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
1	25hz	0,38	10,65	4,047	750
2	25hz	0,38	10,7	4,066	750
3	25hz	0,38	10,71	4,0698	750
4	25hz	0,38	10,68	3,9516	750
5	25hz	0,38	10,66	4,0508	750

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

Berikut adalah hasil pembacaan data motor dari *Inetrface* dengan *frequency* 30 Hz.

**Tabel 7. Pembacaan data motor 30 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
1	30hz	0,37	12,64	4,6768	900
2	30hz	0,37	12,62	4,6694	900
3	30hz	0,37	12,66	4,6842	900
4	30hz	0,37	12,61	4,6657	900
5	30hz	0,37	12,59	4,6583	900

Pada tabel diatas merupakan hasil dari pembacaan data motor yang ditampilkan melalui *interface*, dimana pada *interface* tersebut menampilkan pembacaan *Frequency*, *Current*, *voltage*, *power* dan *rpm*.

#### 4.4.2. Hasil Pengambilan Data Motor Dengan *Tachometer*

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 5 Hz.

**Tabel 8. Pengambilan Data Motor 5 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan <i>Tachometer</i>
1	148,5
2	148,7
3	148,5
4	148,4
5	148,8

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* hanya mengukur *rpm* dengan *frequency* 5 Hz.

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 10 Hz.

**Tabel 9. Pengambilan Data Motor 10 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan <i>Tachometer</i>
1	298,6
2	298,8
3	298,7
4	298,8
5	298,5

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* hanya mengukur *rpm* dengan *frequency* 10 Hz.

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 15 Hz.

**Tabel 10. Pengambilan Data Motor 15 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
1	447,7
2	448,4
3	448,3
4	448,2
5	447,6

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* mengukur *rpm* dengan *frequency* 15 Hz.

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 20 Hz.

**Tabel 11. Pengambilan Data Motor 20 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
1	597,7
2	597,8
3	597,5
4	597,2
5	597,3

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* hanya mengukur *rpm* dengan *frequency* 15 Hz.

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 25 Hz.

**Tabel 12. Pengambilan Data Motor 25 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
1	746,5
2	747,3
3	746,8
4	747,4
5	746,9

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* hanya mengukur *rpm* dengan *frequency* 25 Hz.

Berikut adalah hasil pembacaan data motor dari *Tachometer* dengan *Frequency* 30 Hz.

**Tabel 13. Pengambilan Data Motor 30 Hz**

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
1	896,5
2	896,3
3	897,1
4	897,2
5	896,6

Pada tabel diatas merupakan hasil dari pengambilan data motor yang diukur dengan *tachometer*, dimana pada *tachometer* hanya mengukur *rpm* dengan *frequency* 30 Hz.

#### 4.4.3. Hasil Perbandingan Current pada *Interface* dan Multimeter

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 5 Hz

**Tabel 14. Perbandingan *Current* 5 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,47	0,37
2	0,47	0,37
3	0,47	0,37
4	0,47	0,37
5	0,47	0,37

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 10 Hz

**Tabel 15. Perbandingan *Current* 10 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,44	0,37
2	0,44	0,37
3	0,44	0,37
4	0,44	0,37
5	0,44	0,37

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 15 Hz

**Tabel 16. Perbandingan *Current* 15 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,4	0,37
2	0,4	0,37
3	0,4	0,37
4	0,4	0,37
5	0,4	0,37

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 20 Hz

**Tabel 17. Perbandingan *Current* 20 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,39	0,37
2	0,39	0,37
3	0,39	0,37
4	0,39	0,37
5	0,39	0,37

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 25 Hz

**Tabel 18. Perbandingan *Current* 25 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,38	0,37
2	0,38	0,37
3	0,38	0,37
4	0,38	0,37
5	0,38	0,37

Berikut adalah hasil Perbandingan *Current* pada *Interface* dan Multimeter dengan *frequency* 30 Hz

**Tabel 19. Perbandingan *Current* 30 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
1	0,37	0,37
2	0,37	0,37
3	0,37	0,37
4	0,37	0,37
5	0,37	0,37

## 4.2. Pembahasan

### 4.2.1. Rancangan Monitoring dan Kontrol Mesin *Konveyor Wire Harness* Menggunakan *Visual Studio* Berbasis C#

Sistem otomatisasi dan kontrol yang efektif sangat penting untuk meningkatkan efisiensi dan produktivitas. Mesin *konveyor Wire Harness* adalah salah satu komponen penting dalam proses produksi yang memerlukan monitoring dan kontrol yang tepat. Dalam proyek ini, telah dirancang sebuah aplikasi monitoring dan kontrol untuk mesin *konveyor Wire Harness* menggunakan *Visual Studio* berbasis C#. Aplikasi monitoring dan kontrol yang telah dirancang memiliki beberapa fitur utama yang ditampilkan dalam dua gambar antarmuka berikut:

#### 1. Antarmuka *PLC Command*

Antarmuka ini berfungsi sebagai panel kontrol utama untuk mesin *konveyor Wire Harness*. Fitur-fitur yang disediakan meliputi:

- *Tombol Start, Stop, dan Emergency*  
Tombol-tombol ini memberikan kontrol dasar untuk memulai, menghentikan, dan melakukan tindakan darurat pada mesin konveyor.
- *Status Koneksi dan Pengaturan PLC*  
Menampilkan status koneksi antara aplikasi dan PLC, jenis CPU, *mode PLC (RUN/STOP/PAUSE)*, serta pengaturan kecepatan.
- *Timer dan Data Motor*  
Menampilkan waktu operasi mesin dan data motor yang meliputi frekuensi, arus, tegangan, daya, dan RPM. Fitur ini memberikan informasi real-time tentang kinerja mesin.

#### 2. Antarmuka Monitoring

Antarmuka ini dirancang untuk memberikan informasi monitoring yang komprehensif tentang operasi mesin konveyor. Fitur-fitur utama meliputi:

- *Total Produksi, Produk OK, dan Produk Reject*  
Menampilkan jumlah total produksi, produk yang memenuhi standar kualitas (OK), dan produk yang ditolak (*Reject*).
- *Down Time, Rotation Cycle, dan Target*  
Menampilkan waktu henti mesin, jumlah siklus rotasi, dan target produksi. Pengguna dapat memulai, menghentikan, dan mereset data monitoring melalui tombol yang disediakan.

Dari Aplikasi monitoring dan kontrol yang dirancang dengan *Visual Studio* berbasis C# berhasil menjawab rumusan masalah, yaitu bagaimana merancang monitoring dan kontrol pada mesin *Konveyor Wire Harness* menggunakan *Visual Studio* berbasis C# dengan cara

- a) **Desain Antarmuka Pengguna (UI)**  
Antarmuka pengguna yang dirancang memberikan kemudahan penggunaan dan informasi yang jelas. Pengguna dapat dengan mudah mengontrol mesin konveyor melalui tombol *start*, *stop*, dan *emergency*. Selain itu, status koneksi dan pengaturan PLC memberikan transparansi tentang kondisi operasional mesin.
- b) **Pengumpulan dan Pengolahan Data**  
Aplikasi ini mampu mengumpulkan data dari sensor yang terhubung ke mesin konveyor melalui PLC. Data tersebut diolah dan ditampilkan dalam format yang mudah dipahami. Penggunaan *Visual Studio* berbasis C# memungkinkan integrasi yang mulus dengan perangkat keras dan protokol komunikasi industri, memastikan bahwa data real-time dapat diakses dan dianalisis dengan cepat.
- c) **Implementasi dan Pengujian**  
Proses implementasi melibatkan pengkodean dan integrasi perangkat keras yang efisien. Pengujian fungsional memastikan bahwa setiap fitur dalam aplikasi bekerja sesuai dengan spesifikasi. Pengujian kinerja memastikan bahwa antarmuka responsif dan dapat menampilkan data tanpa penundaan signifikan. Pengujian integrasi memastikan bahwa aplikasi dapat mengumpulkan dan mengontrol data dari perangkat keras dengan akurat.

Perancangan aplikasi monitoring dan kontrol pada mesin *konveyor Wire Harness* menggunakan *Visual Studio* berbasis C# telah berhasil memenuhi tujuan yang ditetapkan. Aplikasi ini menawarkan antarmuka yang intuitif, pengolahan data yang efisien, dan integrasi yang kuat dengan perangkat keras. Dengan adanya aplikasi ini, pengguna dapat memantau dan mengontrol mesin konveyor secara efektif, meningkatkan efisiensi dan produktivitas dalam proses produksi. Aplikasi ini memberikan solusi yang nyata terhadap rumusan masalah yang ada, yaitu bagaimana merancang monitoring dan kontrol pada mesin *Konveyor Wire Harness* menggunakan *Visual Studio* berbasis C#. Implementasi dan pengujian yang cermat memastikan bahwa sistem berjalan sesuai dengan harapan dan memberikan manfaat maksimal bagi pengguna. Dengan sistem ini, perusahaan dapat meningkatkan kualitas produk, mengurangi waktu henti, dan mencapai target produksi dengan lebih efektif.

#### **4.2.2. Pembacaan Data Motor Pada *Interface***

Pada pembacaan data motor dengan berbagai frekuensi yang dilakukan dengan menggunakan GUI pada mesin *conveyor wire harness*, ditemukan bahwa motor menunjukkan respons yang konsisten dan stabil terhadap perubahan frekuensi. Pengukuran dilakukan pada frekuensi 5Hz, 10Hz, 15Hz, 20Hz, 25Hz, dan 30Hz. Hasil pembacaan menunjukkan bahwa parameter-parameter motor seperti arus, tegangan, daya, dan putaran per menit (rpm) memiliki nilai yang

stabil pada setiap frekuensi yang diuji. Meskipun terdapat fluktuasi kecil dalam beberapa pembacaan, namun nilai-nilai tersebut tetap berada dalam rentang yang konsisten. Hal ini menunjukkan bahwa motor beroperasi dalam batas yang aman dan efisien pada rentang *frequency* yang diamati. Dari pembacaan data yang dilakukan, dapat disimpulkan bahwa motor pada mesin *conveyor wire harness* menunjukkan respons yang stabil dan *linier* terhadap peningkatan frekuensi. Kenaikan frekuensi menyebabkan peningkatan yang *proporsional* dalam arus, tegangan, daya, dan rpm. Hal ini menandakan bahwa mesin beroperasi dalam batas yang aman dan efisien pada berbagai kondisi *frequency* yang diuji. Data-data ini dapat digunakan dalam pengembangan sistem kontrol dan monitoring untuk memastikan kinerja optimal dari mesin *conveyor wire harness*, dengan memantau parameter-parameter motor secara *real-time* untuk mendeteksi potensi masalah dan mengambil tindakan pencegahan yang sesuai.

#### 4.4.3. Pengambilan Data Motor Dengan Tachometer

Pada analisis data pembacaan motor menggunakan *tachometer* pada setiap frekuensi yang dilakukan pada mesin *conveyor wire harness*, terlihat bahwa pembacaan RPM menunjukkan respons yang konsisten dan teratur dari motor. Pada setiap frekuensi yang diuji, pembacaan RPM menunjukkan nilai yang stabil dengan variasi yang relatif kecil. Meskipun terdapat beberapa nilai yang mungkin terlalu tinggi atau rendah dari rata-rata, namun secara keseluruhan, data pembacaan tersebut tetap berada dalam rentang yang dapat diterima. Pada beberapa frekuensi, terdapat variasi yang lebih signifikan, namun hal ini mungkin disebabkan oleh faktor-faktor seperti ketidakakuratan alat pengukur atau fluktuasi sementara dalam kinerja motor. Dari hasil analisis data pembacaan motor menggunakan tachometer pada setiap frekuensi, dapat disimpulkan bahwa motor pada mesin *conveyor wire harness* menunjukkan respons yang stabil dan teratur terhadap perubahan frekuensi. Meskipun terdapat beberapa variasi dalam pembacaan RPM, namun secara umum, motor menunjukkan kinerja yang baik dan merespons dengan baik terhadap perubahan frekuensi. Data-data ini sangat penting dalam pemantauan kinerja motor pada mesin *conveyor wire harness*, karena memungkinkan untuk memantau kecepatan putar motor secara *real-time* dan mendeteksi perubahan yang tidak diinginkan atau potensi masalah dengan cepat. Dengan demikian, dapat diambil tindakan yang tepat untuk menjaga kinerja optimal dari mesin tersebut.

#### 4.2.4. Perbandingan *Current* pada *Interface* dan Multimeter

Proses ini dilakukan untuk mengetahui perbandingan antara *interface* dengan multimeter, dimana Pengukuran arus (*current*) merupakan aspek penting dalam berbagai sistem, termasuk sistem kontrol dan monitoring pada Mesin Conveyor

*Wire Harness*. Dua cara umum untuk mengukur arus adalah melalui *Interface* dan Multimeter. Masing-masing cara memiliki kelebihan dan kekurangan, serta pertimbangan khusus saat digunakan pada frekuensi tertentu.

Pengukuran *current* dapat memengaruhi kinerja dan akurasi *Interface* dan Multimeter dimana jika pada frekuensi tinggi, *interface* dengan *hardware* dan *software* yang kurang optimal mungkin mengalami lag atau penurunan akurasi sedangkan Multimeter digital umumnya memiliki *bandwidth* terbatas untuk mengukur arus pada frekuensi yang diinginkan, sehingga harus melakukan pengujian dan kalibrasi secara berkala untuk memastikan akurasi pengukuran serta memperhatikan faktor-faktor lain seperti noise dan interferensi elektromagnetik pada frekuensi tinggi dalam melakukan pengukuran *current* pada *interface* dan multimeter.

## Bab 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Berdasarkan analisis yang telah dilakukan terhadap data pembacaan motor pada berbagai frekuensi menggunakan *tachometer*, serta data lainnya yang diperoleh dari pengembangan sistem, terdapat beberapa kesimpulan yang dapat diambil.

Pertama, penggunaan *Visual Studio* Berbasis C# sebagai platform pengembangan sistem kontrol dan monitoring telah terbukti efektif dalam memantau kinerja mesin *conveyor wire harness*. Dengan fitur-fitur yang disediakan, seperti pembuatan antarmuka grafis pengguna (GUI), pemrograman berbasis *event*, dan kemampuan untuk berinteraksi dengan perangkat keras melalui berbagai protokol komunikasi, *Visual Studio* memberikan fleksibilitas dan kontrol yang diperlukan dalam pengembangan sistem tersebut.

Kedua, analisis data pembacaan motor pada setiap frekuensi menggunakan *tachometer* memberikan gambaran yang jelas tentang respons motor terhadap perubahan frekuensi. Meskipun terdapat variasi dalam pembacaan RPM, namun secara umum, motor menunjukkan kinerja yang stabil dan merespons dengan baik terhadap perubahan frekuensi. Data ini sangat penting dalam pemantauan kinerja motor dan pengambilan keputusan untuk menjaga kinerja optimal dari mesin *conveyor wire harness*.

Selanjutnya, pengembangan sistem kontrol dan monitoring tidak hanya memberikan informasi tentang kinerja mesin secara *real-time*, tetapi juga memungkinkan untuk deteksi dini terhadap potensi masalah atau kegagalan yang dapat terjadi. Dengan memantau parameter-parameter kritis seperti arus, tegangan, daya, dan putaran per menit (*rpm*) secara terus-menerus, sistem dapat memberikan peringatan dini kepada operator atau teknisi jika ada indikasi adanya masalah. Hal ini dapat mengurangi risiko kerusakan mesin yang lebih besar dan mengurangi waktu henti produksi.

Selain itu, pengembangan sistem kontrol dan monitoring juga memungkinkan untuk analisis performa mesin secara menyeluruh. Dengan mengumpulkan dan menganalisis data kinerja mesin dari waktu ke waktu, dapat diidentifikasi tren kinerja, pola kegagalan yang mungkin muncul, dan potensi untuk meningkatkan efisiensi dan produktivitas mesin. Informasi ini dapat digunakan untuk perencanaan pemeliharaan yang lebih efisien, penjadwalan produksi yang lebih baik, dan pengoptimalan operasi mesin secara keseluruhan.

### 5.2. Saran

Berdasarkan hasil analisis dan kesimpulan yang telah disampaikan, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut yaitu:

1. Disarankan untuk mengembangkan fitur tambahan dalam sistem kontrol dan *monitoring* yang dapat meningkatkan fungsionalitas dan utilitas sistem.
2. Coba untuk melakukan validasi dan verifikasi lebih lanjut terhadap sistem yang dikembangkan, baik melalui pengujian simulasi maupun pengujian di lapangan.
3. Optimasi Performa Sistem Untuk meningkatkan performa sistem, disarankan untuk melakukan optimasi lebih lanjut terhadap kode program, penggunaan sumber daya komputasi, dan algoritma yang digunakan.
4. Agar sistem dapat digunakan secara efektif oleh operator dan teknisi, disarankan untuk menyediakan pelatihan yang memadai tentang penggunaan sistem kontrol dan *monitoring*.

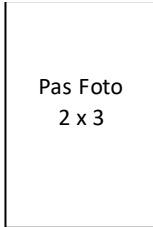
## Daftar Pustaka

- [1] D. James, "Perancangan Sistem Konveyor". Jakarta: FT UI, 2008.
- [2] K. E. Kamal, et al., "Implementation of Six Sigma Methodologies in Automotive Wiring Harnesses Manufacturing Companies. 'ABC' Plant Case Study," *International Journal of Scientific & Engineering Research*, vol. 9, no. 8, pp. 1-8, Aug. 2018.
- [3] M. S. Ab-Rahman and F. N. Khameneh, "Cost-Effective Wire harness Model by Using Polymer Optical Fiber," *Journal of Computer Science*, vol. 9, no. 7, pp. 935-942, 2013.
- [4] B. S. Kim, et al., "Large-Deflection Analysis of Automotive Vehicle's Door Wiring Harness System Using Finite Element Method," *International Scholarly and Scientific Research & Innovation*, vol. 2, no. 11, pp. 1258-1262, 2008.
- [5] A. Malden, Kardiman, Marno, N. Burhan, "Mempelajari Proses Produksi Wire Harness dan Studi Kasus Serta Perawatan pada Mesin Cheker KCM di CV. HII," *TRAKSI: Majalah Ilmiah Teknik Mesin*, vol. 21, no. 1, pp. 135-148, 2021.
- [6] M. D. Cahyo, A. Ubaidillah, A. F. Ibadillah, "Rancang Bangun Sistem Proteksi dan Monitoring Energi Listrik Berbasis Mikrokontroler Menggunakan Visual Studio Uji Coba Diakses di PT. Pancawana Indonesia," *CYCLOTRON*, vol. 1, no. 2, pp. 1-6, Jul. 2018.
- [7] S. Sadi, "Sistem Pengendali Konveyor Belt pada PT. XYZ Tangerang," *Jurnal Teknik*, vol. 1, no. 2, pp. 35-50, Dec. 2012.
- [8] N. Y. Putri, R. Mukhaiya, "Control and Monitoring System Process Handling Production on SMI 4.0 Machines using PLC Controller Wirelessly Based on Human Machine Interface," *JTEV (Jurnal Teknik Elektro dan Vokasional)*, vol. 8, no. 1, pp. 158-168, 2022.
- [9] T. H. Handoko, "Manajemen Personalia & Sumberdaya Manusia", 2012.
- [10] T. Sutabri, "Analisis Sistem Informasi", 2012.
- [11] "Sistem kendali." Internet: [https://p2k.stekom.ac.id/ensiklopedia/Sistem\\_kendali](https://p2k.stekom.ac.id/ensiklopedia/Sistem_kendali), accessed Jun. 19, 2023.
- [12] "Pengertian Microsoft Visual Studio 2010," didityudiatma. Internet: <https://didityudiatma.wordpress.com/2016/03/21/pengertian-microsoft-visual-studio-2010/>, Mar. 21, 2016 [accessed Jun. 19, 2023].
- [13] R. Sheldon, "C# (C-Sharp)," *TechTarget*, Dec. 01, 2022. Accessed: Jun. 14, 2023. [Online]. Available: <https://www.techtarget.com/whatis/definition/C-Sharp>.
- [14] I. R. Dewi and R. Malfiany, "Perancangan Sistem Informasi Administrasi Pembayaran Pada Sdit Lampu Iman Karawang Berbasis Visual Basic 6.0"

*Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 12, no. 2, pp. 4–12, Jul. 2017.

- [15] B. Tujni and M. Megawaty, "Pelatihan Pembuatan Web Dengan Php Pada SMP Negeri 27 Palembang," *Jurnal Abdimas Mandiri*, vol. 1, no. 1, Jan. 2018.

## Biodata

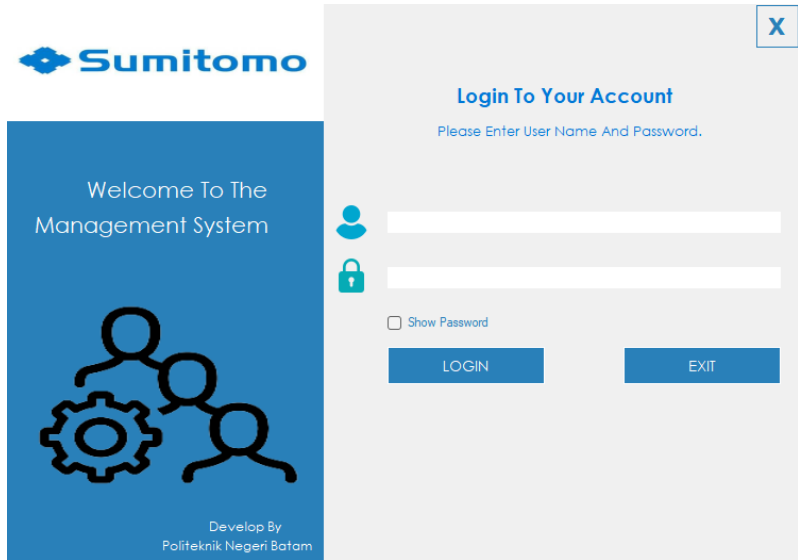


Nama : Herdi Gunanta Sirait  
TTL : Batam, 04 Agustus 2002  
Agama : Kristen Protestan  
Alamat : Kavling Baru Sagulung Sumber Mulia B5  
No 71  
Email : nantagurky@gmail.com  
Riwayat Pendidikan SMA/SMK : SMK Negeri 5 Batam  
SMP : SMP Negeri 35 Batam

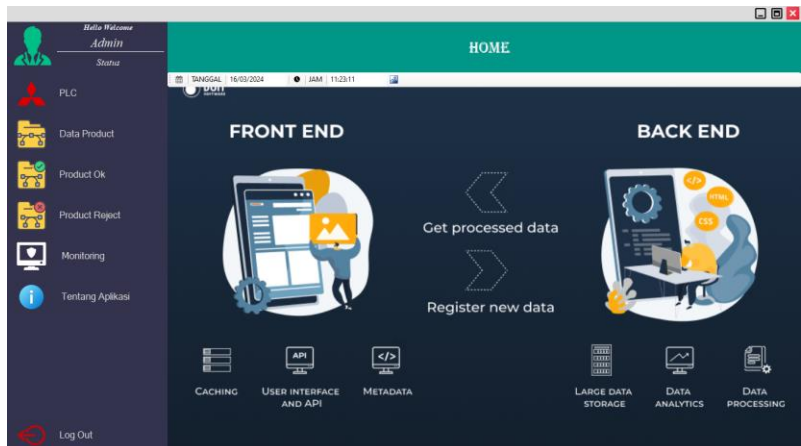
# Lampiran

## Lampiran 1 : Design Interface

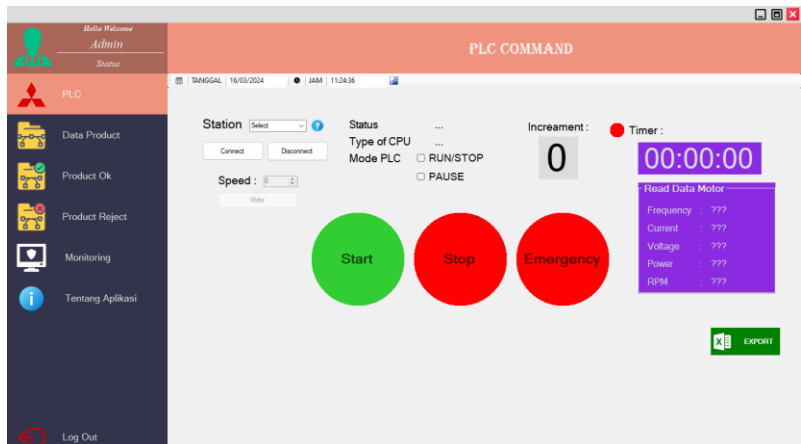
### Tampilan *Login*



## Tampilan Menu Home



## Menu PLC



## Menu Data Product

The screenshot shows the 'DATA PRODUCT' menu. The sidebar on the left contains the following items: PLC, Data Product (highlighted), Product Ok, Product Reject, Monitoring, and Tentang Aplikasi. The main content area has a header 'DATA PRODUCT' and a status bar showing 'TANGGAL | 16/03/2024' and 'JAM | 11:25:21'. Below the header is a search form with the following fields: CARI ID, ID, LANE, TANGGAL (with a value of 16/03/2024), TOTAL (with a value of 0), and TARGET. To the right of the form is a table with the following structure:

ID	LANE	TANGGAL
----	------	---------

At the bottom of the table are four buttons: INSERT, UPDATE, DELETE, and EXPORT.

## Menu Product OK

The screenshot shows the 'PRODUCT\_OK' menu. The sidebar on the left contains the following items: PLC, Data Product, Product Ok (highlighted), Product Reject, Monitoring, and Tentang Aplikasi. The main content area has a header 'PRODUCT\_OK' and a status bar showing 'TANGGAL | 16/03/2024' and 'JAM | 11:25:37'. Below the header is a search form with the following fields: CARI ID, ID, LANE, TANGGAL (with a value of 16/03/2024), TOTAL (with a value of 0), and TARGET. A 'TARGET:' label is visible above the search form. To the right of the form is a table with the following structure:

ID	LANE	TANGGAL
----	------	---------

At the bottom of the table are four buttons: INSERT, UPDATE, DELETE, and EXPORT.

## Menu Product Reject

The screenshot displays the 'PRODUCT\_REJECT' menu. On the left sidebar, the 'Product Reject' option is highlighted. The main area features a search form with the following fields:

- CARF ID:
- ID:
- LANE:
- TANGGAL: 16/03/2024
- TOTAL: 0

Below the search form is a table with the following structure:

ID	LANE	TANGGAL
[Empty Table]		

At the bottom of the table area, there are four action buttons: INSERT (blue), UPDATE (orange), DELETE (red), and EXPORT (green).

## Monitoring

The screenshot displays the 'MONITORING' menu. The dashboard contains the following data cards:

- TOTAL PRODUKS:** 0
- DOWN TIME:** [Clock icon] with control buttons: START (green), STOP (red), RESET (orange)
- PRODUK OK:** 0
- ROTATOR/CYCLE:** 0
- PRODUK REJECT:** 0
- TARGET:** 0

## Lampiran 2 : Bahasa Pemrograman C#

### C# Tampilan *Login*

```
//program button login
private void button1_Click_1(object sender, EventArgs e)
{
    string query = "select * from tbl_admin where kode_admin='" + txtUserName.Text +
    "' and pwd_admin='" + txtPassword.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    dr.Read();
    if (dr.HasRows)
    {
        MessageBox.Show("Login Berhasil");
        FormMenu frmUtama = new FormMenu();
        frmUtama.Show();
        this.Hide();
        conn.Close();
    }
    else if (txtUserName.Text == "" && txtPassword.Text == "")
    {
        MessageBox.Show("Silahkan Masukan Pasword Anda");
    }
    else
    {
        MessageBox.Show("Pasword Yang Anda Masukan Salah");
    }
}

//program menampilkan passwor
private void checkBox1_CheckedChanged_1(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        txtPassword.UseSystemPasswordChar = false;
    }
    else
    {
        txtPassword.UseSystemPasswordChar = true;
    }
}
}
```

```

//program button exit
private void button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

//program button close (X)
private void btnClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}

```

### C# Tampilan Menu *Home*

```

public partial class FormMenu : Form
{
    private Button currentButton;
    private Random random;
    private int tempIndex;
    private Form activeForm;

    public FormMenu()
    {
        InitializeComponent();
        random = new Random();
    }
    private Color SelectThemeColor()
    {
        int index = random.Next(ThemeColor.colorList.Count);
        while (tempIndex == index)
        {
            index= random.Next(ThemeColor.colorList.Count);
        }
        tempIndex = index;
        string color = ThemeColor.colorList[index];
        return ColorTranslator.FromHtml(color);
    }
}

```

## C# Menu PLC

```
public ActUtilType plc = new ActUtilType();
int mode = 0;

private void Command_Load(object sender, EventArgs e)
{
    //cannot push command button
    circularButtonStart.Enabled = false;
    circularButtonStop.Enabled = false;
    circularButtonEmg.Enabled = false;
    dataGridView1.Visible = false;
    btnWrite.Enabled = false;
    SpeedValue.Enabled = false;
    //Start
    circularButtonStart.Text = "Start";
    //Stop
    circularButtonStop.Text = "Stop";
    //Emergency
    circularButtonEmg.Text = "Emergency";
    t2.Interval = 750;
    t2.Tick += new EventHandler(t_tick2);
    //Increment
    NBIncrement.Text = "0";
    //timer
    t = new System.Timers.Timer();
    t.Interval = 1000;
    t.Elapsed += OnTimeEvent;
    waktu = new System.Timers.Timer();
    waktu.Interval = 1000;
    waktu.Elapsed += OnTimeEvent1;
    //menampilkan gridview
    tampilkandata();
}
```

```

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    // run or stop PLC
    if (checkBox1.Checked == true) { mode = 1; } else { mode = 0; }
    plc.SetCpuStatus(mode);
}

private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
    // pause PLC = 2
    if (checkBox2.Checked == true) { mode = 2; } else { mode = 0; }
    plc.SetCpuStatus(mode);
}

//timer
private void OnTimeEvent(object sender, System.Timers.ElapsedEventArgs e)
{
    Invoke(new Action() =>
    {
        s += 1;
        if (s == 60)
        {
            s = 0;
            m += 1;
        }
        if (m == 60)
        {
            m = 0;
            h += 1;
        }
        timerValue.Text = string.Format("{0}:{1}:{2}", h.ToString().PadLeft(2, '0'), m.ToString().PadLeft(2, '0'), s.ToString().PadLeft(2, '0'));
    });
}

private void OnTimeEvent1(object sender, System.Timers.ElapsedEventArgs e)
{
    Invoke(new Action() => { s1 += 1; secondT.Text = string.Format("{0}", s1.ToString().PadLeft(1, '0')); });
}

// Speed Motor Command
private void btnWrite_Click(object sender, EventArgs e)
{
    plc.SetDevice(d20.Text, Convert.ToInt32(SpeedValue.Text));
}

// Motor Command
private void circularButtonStart_Click(object sender, EventArgs e)
{
    circularButtonStart.Text = "Start";
    circularButtonStart.BackColor = Color.LimeGreen;
    //value 1
    plc.SetDevice(x0.Text, Convert.ToInt16(value0.Text));
    //connect data motor
    timer1.Enabled = true;
    timer1.Interval = 1000;
    //timer start
    t.Start();
    waktu.Start();
    lampTimer.BackColor = Color.LimeGreen;
}

//button stop
private void circularButtonStop_Click(object sender, EventArgs e)
{
    circularButtonStop.Text = "Stop";
    circularButtonStop.BackColor = Color.Red;
    //value 1
    plc.SetDevice(x1.Text, Convert.ToInt16(value0.Text));
    //disconnect data motor
    timer1.Enabled = false;
    //timer stop
    t.Stop();
    waktu.Stop();
    lampTimer.BackColor = Color.Red;
}
}

```

```

//Button emergency
private void circularButtonEmg_Click(object sender, EventArgs e)
{
    if (on2)
    {
        circularButtonEmg.Text = "Emergency";
        circularButtonEmg.BackColor = Color.Yellow;
        t2.Start();
        on2 = false;
        //cannot click other button
        circularButtonStart.Enabled = false;
        circularButtonStop.Enabled = false;
        //write value 1
        plc.SetDevice(x2.Text, Convert.ToInt16(value1.Text));
        //disconnect data motor
        timer1.Enabled = false;
        //timer stop
        t.Stop();
        waktu.Stop();
        lampTimer.BackColor = Color.Red;
    }
    else
    {
        circularButtonEmg.Text = "Emergency";
        t2.Stop();
        circularButtonEmg.BackColor = Color.Red;
        on2 = true;
        //can click other button
        circularButtonStart.Enabled = true;
        circularButtonStop.Enabled = true;
        //back to value 0
        plc.SetDevice(x2.Text, Convert.ToInt16(value0.Text));
    }
}
}

```

```

//read data motor
private void timer1_Tick_1(object sender, EventArgs e)
{
    int value0, value1, value2;

    //data frequency
    plc.GetDevice(d100.Text, out value0);
    valueF.Text = value0.ToString();
    int nilai0 = Convert.ToInt32(valueF.Text);
    double jumlah0 = nilai0 * 0.01;
    DataF.Text = (Convert.ToString(jumlah0));

    //data current
    plc.GetDevice(d101.Text, out value1);
    valueC.Text = value1.ToString();
    int nilai1 = Convert.ToInt32(valueC.Text);
    double jumlah1 = nilai1 * 0.01;
    DataC.Text = (Convert.ToString(jumlah1));

    //data voltage
    plc.GetDevice(d102.Text, out value2);
    valueV.Text = value2.ToString();
    int nilai2 = Convert.ToInt32(valueV.Text);
    double jumlah2 = nilai2 * 0.01;
    DataV.Text = (Convert.ToString(jumlah2));

    //data power
    double jumlah3 = jumlah2 * jumlah1;
    DataP.Text = (Convert.ToString(jumlah3));

    //data rpm
    double jumlah4 = (jumlah0 * 120) / 4;
    DataR.Text = (Convert.ToString(jumlah4));

    //pembacaan data
    string connection = "server=localhost;user id=root;database=monitoring";
    string query = "INSERT INTO pembacaan(frequency,current,voltage)VALUES('" + this.DataF.Text + "','" + this.DataC.Text + "','" + this.DataV.Text +
    MySqlConnection conn = new MySqlConnection(connection);
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    conn.Close();
}

```

### C# Menu Data Product

```
public DataProduct()
{
    InitializeComponent();
}

//program fungsi menampilkan data
public void tampilkandata()

SqlConnection conn = koneksi.getKoneksi();
SqlCommand command = conn.CreateCommand();
command.CommandText = "SELECT * FROM product";

conn.Open();
DataSet ds = new DataSet();
MySqlDataAdapter da = new MySqlDataAdapter(command);
da.Fill(ds, "data");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "data";
txt1.Text = "";
txt2.Text = "";
txt4.Text = dataGridView1.RowCount.ToString();
conn.Close();
```

```

//program menampilkan data dan tanggal secara realtime
private void DataProduct_Load(object sender, EventArgs e)
{
    txt3.Text = DateTime.Now.ToString("dd/MM/yyyy");
    tampilkandata();
}

//program insert data
private void btninsert_Click(object sender, EventArgs e)
{
    string query = "select * from product where ID='" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    dr.Read();
    if (dr.HasRows)
    {
        MessageBox.Show(" data sudah ada");
    }
    else
    {
        insert();
    }
}

```

```

//program update data
private void btnupdate_Click(object sender, EventArgs e)
{
    string query = "UPDATE product SET LANE='" + this.txt2.Text + "',TANGGAL='"
    + this.txt3.Text +
    "'WHERE ID='" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Record has been updated successfully");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}

//program button delete
private void btndelete_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM product WHERE ID='" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    dr.Read();
    MessageBox.Show("Data has been succesfully deleted!!");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}

//program button export
private void btnexport_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
        xcelApp.Application.Workbooks.Add(Type.Missing);
        for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
        {
            xcelApp.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
        }

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                xcelApp.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
        xcelApp.Columns.AutoFit();
        xcelApp.Visible = true;
    }
}

```

```

//program mencari data berdasarkan ID
private void txt6_TextChanged(object sender, EventArgs e)
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "select * from product where ID like '%" + txt6.Text + "%'";
    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "data";
}

```

### C# Product OK

```

public PRODUCT_OK()
{
    InitializeComponent();
}
// program fungsi menampilkan data
public void tampilkandata()
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "SELECT * FROM itemok";

    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "data";
    txt1.Text = "";
    txt2.Text = "";
    txt4.Text = dataGridView1.RowCount.ToString();
    conn.Close();
}

```

```

public void insert()
{
    string query = "INSERT INTO itemok(ID,LANE,TANGGAL)VALUES('" + this.txt1.Text
+ "','" + this.txt2.Text + "','" + this.txt3.Text + "')";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Successfully saved");
    //rumus target
    tampilkandata();
    total = txt4.Text;
    int nilai1 = Convert.ToInt32(txt4.Text);
    int nilai = Convert.ToInt32(txttarget.Text);
    target = nilai - nilai1;
    conn.Close();
}
//program load untuk menampilkan data, tanggal, target
private void PRODUCT_OK_Load(object sender, EventArgs e)
{
    txt3.Text = DateTime.Now.ToString("dd/MM/yyyy");
    txttarget.Text = DataProduct.target;
    tampilkandata();
}
//program button insert
private void btninsert_Click(object sender, EventArgs e)
{
    string query = "select * from itemok where ID =" + txt1.Text + """;
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    dr.Read();
    if (dr.HasRows)
    {
        MessageBox.Show(" data sudah ada");
    }
    else
    {
        insert();
    }
}
}

```

```

//program button update
private void btnupdate_Click(object sender, EventArgs e)
{
    string query = "UPDATE itemok SET LANE='" + this.txt2.Text + "',TANGGAL='"
    + this.txt3.Text + "'WHERE ID='" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Record has been updated successfully");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}

//program button delete
private void btndelete_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM itemok WHERE ID='" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Data has been succesfully deleted!!");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}

//program menyalin data dari datagridview ke textbox
private void dataGridView1_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)
{
    try
    {
        DataGridViewRow row = this.dataGridView1.Rows[e.RowIndex];
        txt1.Text = row.Cells["ID"].Value.ToString();
        txt2.Text = row.Cells["LANE"].Value.ToString();
        txt3.Text = row.Cells["TANGGAL"].Value.ToString();
    }
    catch (Exception X)
    {
        MessageBox.Show(X.ToString());
    }
}

```

```

//program button export
private void btnexport_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
        xcelApp.Application.Workbooks.Add(Type.Missing);
        for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
        {
            xcelApp.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
        }

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                xcelApp.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
        xcelApp.Columns.AutoFit();
        xcelApp.Visible = true;
    }
}

//program mencari data berdasarkan ID
private void txt5_TextChanged(object sender, EventArgs e)
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "select * from itemok where ID like '%" + txt5.Text + "%'";
    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "data";
}
}

```

## C# Product Reject

```
public PRODUCT_REJECT()
{
    InitializeComponent();
}
//program fungsi menampilkan data
public void tampilkandata()
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "SELECT * FROM itemreject";
    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "data";
    txt1.Text = "";
    txt2.Text = "";
    txt4.Text = dataGridView1.RowCount.ToString();
    conn.Close();
}
private void insert()
{
    string query = "INSERT INTO itemreject(ID,LANE,TANGGAL)VALUES('" + this.txt1.Text
    + "','" + this.txt2.Text + "','" + this.txt3.Text + "')";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Successfully saved");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}
//program menyalin data pada datagridview ke textbox
private void dataGridView1_CellMouseClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        DataGridViewRow row = this.dataGridView1.Rows[e.RowIndex];
        txt1.Text = row.Cells["ID"].Value.ToString();
        txt2.Text = row.Cells["LANE"].Value.ToString();
        txt3.Text = row.Cells["TANGGAL"].Value.ToString();
    }
    catch (Exception X)
    {
        MessageBox.Show(X.ToString());
    }
}
```

```

//program button export
private void btnexport_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
        xcelApp.Application.Workbooks.Add(Type.Missing);
        for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
        {
            xcelApp.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
        }

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                xcelApp.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
        xcelApp.Columns.AutoFit();
        xcelApp.Visible = true;
    }
}

//program mencari data berdasarkan id
private void txt5_TextChanged(object sender, EventArgs e)
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "select * from itemreject where ID like '%" + txt5.Text + "%'";
    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "data";
}

//program button insert
private void btninsert_Click(object sender, EventArgs e)
{
    string query = "select * from itemreject where ID =" + txt1.Text + " ";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    dr.Read();
    if (dr.HasRows)
    {
        MessageBox.Show(" data sudah ada");
    }
    else
    {
        insert();
    }
}
}

```

```

//program button update
private void btnupdate_Click_1(object sender, EventArgs e)
{
    string query = "UPDATE itemreject SET LANE=" + this.txt2.Text + ",TANGGAL="
+ this.txt3.Text + "'WHERE ID=" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Record has been updated successfully");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}
//program button delete
private void btndelete_Click_1(object sender, EventArgs e)
{
    string query = "DELETE FROM itemreject WHERE ID=" + this.txt1.Text + "'";
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader dr;
    conn.Open();
    dr = cmd.ExecuteReader();
    MessageBox.Show("Data has been succesfully deleted!!");
    tampilkandata();
    total = txt4.Text;
    conn.Close();
}
//program button export
private void btnexport_Click_1(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        Microsoft.Office.Interop.Excel.Application xcelApp = new Microsoft.Office.Interop.Excel.Application();
        xcelApp.Application.Workbooks.Add(Type.Missing);
        for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
        {
            xcelApp.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
        }

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                xcelApp.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
        xcelApp.Columns.AutoFit();
        xcelApp.Visible = true;
    }
}

```

```
//program menampilkan data dan tanggal secara realtime
private void PRODUCT_REJECT_Load(object sender, EventArgs e)
{
    txt3.Text = DateTime.Now.ToString("dd/MM/yyyy");
    tampilkandata();
}
```

## C# Monitoring

```
public partial class MONITORING : Form
{
    public Stopwatch stopw = null;
    public MONITORING()
    {
        InitializeComponent();
    }

    //program fungsi menampilkan monitoringg pada DATA PRODUCT
    public void tampilkandata()
    {
        MySqlConnection conn = koneksi.getKoneksi();
        MySqlCommand command = conn.CreateCommand();
        command.CommandText = "SELECT * FROM product";

        conn.Open();
        DataSet ds = new DataSet();
        MySqlDataAdapter da = new MySqlDataAdapter(command);
        da.Fill(ds, "data");
        dataGridView1.DataSource = ds;
        dataGridView1.DataMember = "data";
        textBox1.Text = dataGridView1.RowCount.ToString();
        conn.Close();
    }
}
```

```

// program fungsi menampilkan PRODUCT REJECT
public void tampilkandata1()
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "SELECT * FROM itemreject";

    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView2.DataSource = ds;
    dataGridView2.DataMember = "data";
    textBox22.Text = dataGridView2.RowCount.ToString();
    conn.Close();
}

// program fungsi menampilkan data
public void tampilkandata2()
{
    MySqlConnection conn = koneksi.getKoneksi();
    MySqlCommand command = conn.CreateCommand();
    command.CommandText = "SELECT * FROM itemok";

    conn.Open();
    DataSet ds = new DataSet();
    MySqlDataAdapter da = new MySqlDataAdapter(command);
    da.Fill(ds, "data");
    dataGridView3.DataSource = ds;
    dataGridView3.DataMember = "data";
    textBox23.Text = dataGridView3.RowCount.ToString();
    conn.Close();
}

```

```
//program monitoring data berdasarkan nilai yang masuk
private void MONITORING_Load(object sender, EventArgs e)
{
    tampilkandata(); //monitoring DATA PRODUCT
    tampilkandata1();// monitoring PRODUCT REJECT
    tampilkandata2();
    dataGridView1.Visible = false;
    dataGridView2.Visible = false;
    dataGridView3.Visible = false;
    txt5.Text = (Convert.ToString(DataProduct.rotation));
    txt6.Text = (Convert.ToString(PRODUCT_OK.target));
}
```

## Lampiran 3 : Software to Database

### XAMPP Control Panel

XAMPP Control Panel v3.3.0 [ Compiled: Apr 6th 2021 ]

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	14484 13268	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	3260	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

10:48:57 [mysql] This may be due to a blocked port, missing dependencies,  
10:48:57 [mysql] improper privileges, a crash, or a shutdown by another method.  
10:48:57 [mysql] Press the Logs button to view error logs and check  
10:48:57 [mysql] the Windows Event Viewer for more clues  
10:48:57 [mysql] If you need more help, copy and post this  
10:48:57 [mysql] entire log window on the forums  
10:48:59 [mysql] Attempting to start MySQL app...  
10:48:59 [mysql] Status change detected: running

### Localhost phpMyAdmin

phpMyAdmin 4.9.2-0.1

Struktur SQL Cari Kueri Ekspor Import Operasi Hak Akses Rutine Event Trigger Desain

Mengandung kata:

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
<input type="checkbox"/> itemok	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> itemreject	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> pembacaan	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	27,547	InnoDB	utf8mb4_general_ci	1.5 MB	-
<input type="checkbox"/> product	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> sensor	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> tbl_admin	<input type="checkbox"/> Jelajahi <input type="checkbox"/> Struktur <input type="checkbox"/> Cari <input type="checkbox"/> Tambahkan <input type="checkbox"/> Kosongkan <input type="checkbox"/> Hapus	1	InnoDB	utf8mb4_general_ci	16.0 KB	-

6 tabel Jumlah 27,516 InnoDB utf8mb4\_general\_ci 1.6 MB

Cetak Kamus data

Create new table

Nama tabel:  Jumlah kolom:  Buat

## Lampiran 4 : Pembacaan data motor pada *Interface*

### *Frequency 5 Hz*

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
6	5hz	0,47	3,32	1,5604	150
7	5hz	0,47	3,32	1,5604	150
8	5hz	0,47	3,33	1,5651	150
9	5hz	0,47	3,32	1,5604	150
10	5hz	0,47	3,32	1,5604	150
11	5hz	0,47	3,32	1,5604	150
12	5hz	0,47	3,34	1,5698	150
13	5hz	0,47	3,32	1,5604	150
14	5hz	0,47	3,32	1,5604	150
15	5hz	0,47	3,33	1,5651	150
16	5hz	0,47	3,32	1,5604	150
17	5hz	0,47	3,32	1,5604	150
18	5hz	0,47	3,33	1,5651	150
19	5hz	0,47	3,32	1,5604	150
20	5hz	0,47	3,33	1,5651	150

### *Frequency 10 Hz*

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
6	10hz	0,44	5,22	2,2968	300
7	10hz	0,44	5,24	2,3056	300
8	10hz	0,44	5,23	2,2489	300
9	10hz	0,44	5,21	2,2924	300
10	10hz	0,44	5,22	2,2968	300
11	10hz	0,44	5,24	2,3056	300

12	10hz	0,44	5,22	2,2968	300
13	10hz	0,44	5,23	2,2489	300
14	10hz	0,44	5,24	2,3056	300
15	10hz	0,44	5,21	2,2924	300
16	10hz	0,44	5,22	2,2968	300
17	10hz	0,44	5,23	2,2489	300
18	10hz	0,44	5,24	2,3056	300
19	10hz	0,44	5,21	2,2924	300
20	10hz	0,44	5,23	2,2489	300

### **Frequency 15 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
6	15hz	0,4	6,99	2,796	450
7	15hz	0,4	7,02	2,808	450
8	15hz	0,4	6,98	2,792	450
9	15hz	0,4	7	2,8	450
10	15hz	0,4	7,01	2,804	450
11	15hz	0,4	6,99	2,796	450
12	15hz	0,4	7,02	2,808	450
13	15hz	0,4	6,98	2,792	450
14	15hz	0,4	7	2,8	450
15	15hz	0,4	7,01	2,804	450
16	15hz	0,4	6,99	2,796	450
17	15hz	0,4	7,02	2,808	450
18	15hz	0,4	6,98	2,792	450
19	15hz	0,4	7	2,8	450
20	15hz	0,4	7,01	2,804	450

### Frequency 20 Hz

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>Voltage</i>	<i>power</i>	<i>rpm</i>
6	20hz	0,39	8,84	3,3592	600
7	20hz	0,39	8,83	3,4437	600
8	20hz	0,39	8,86	3,3668	600
9	20hz	0,39	8,82	3,4398	600
10	20hz	0,39	8,8	3,432	600
11	20hz	0,39	8,84	3,3592	600
12	20hz	0,39	8,83	3,4437	600
13	20hz	0,39	8,86	3,3668	600
14	20hz	0,39	8,82	3,4398	600
15	20hz	0,39	8,8	3,432	600
16	20hz	0,39	8,84	3,3592	600
17	20hz	0,39	8,83	3,4437	600
18	20hz	0,39	8,86	3,3668	600
19	20hz	0,39	8,82	3,4398	600
20	20hz	0,39	8,8	3,432	600

### Frequency 25 Hz

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
6	25hz	0,38	10,65	4,047	750
7	25hz	0,38	10,7	4,066	750
8	25hz	0,38	10,71	4,0698	750
9	25hz	0,38	10,68	3,9516	750
10	25hz	0,38	10,66	4,0508	750
11	25hz	0,38	10,65	4,047	750
12	25hz	0,38	10,7	4,066	750

13	25hz	0,38	10,71	4,0698	750
14	25hz	0,38	10,68	3,9516	750
15	25hz	0,38	10,66	4,0508	750
16	25hz	0,38	10,65	4,047	750
17	25hz	0,38	10,7	4,066	750
18	25hz	0,38	10,71	4,0698	750
19	25hz	0,38	10,68	3,9516	750
20	25hz	0,38	10,66	4,0508	750

### **Frequency 30 Hz**

Pengujian	Data				
	<i>frequency</i>	<i>Current</i>	<i>voltage</i>	<i>power</i>	<i>rpm</i>
6	30hz	0,37	12,64	4,6768	900
7	30hz	0,37	12,62	4,6694	900
8	30hz	0,37	12,66	4,6842	900
9	30hz	0,37	12,61	4,6657	900
10	30hz	0,37	12,59	4,6583	900
11	30hz	0,37	12,64	4,6768	900
12	30hz	0,37	12,62	4,6694	900
13	30hz	0,37	12,66	4,6842	900
14	30hz	0,37	12,61	4,6657	900
15	30hz	0,37	12,59	4,6583	900
16	30hz	0,37	12,64	4,6768	900
17	30hz	0,37	12,62	4,6694	900
18	30hz	0,37	12,66	4,6842	900
19	30hz	0,37	12,61	4,6657	900
20	30hz	0,37	12,59	4,6583	900

## Lampiran 5 : Pengambilan data rpm motor pada *tachometer*

### Frequency 5 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	148,5
7	148,5
8	148,8
9	148,5
10	148,6
11	148,8
12	148,5
13	148,8
14	148,5
15	148,7
16	148,8
17	148,5
18	148,7
19	148,8
20	148,5

### Frequency 10 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	298,6
7	398,7

8	298,5
9	298,6
10	298,5
11	298,8
12	298,6
13	398,7
14	298,5
15	298,6
16	298,5
17	398,7
18	298,5
19	298,6
20	298,5

### Frequency 15 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	447,9
7	447,5
8	448,3
9	447,6
10	447,7
11	448,4
12	447,6
13	448,3
14	447,6

15	447,8
16	448,2
17	447,8
18	448,3
19	448,2
20	447,5

### Frequency 20 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	597,6
7	597,7
8	597,2
9	597,3
10	597,4
11	597,7
12	597,4
13	597,3
14	597,1
15	597,8
16	597,9
17	597,2
18	597,5
19	597,6
20	597,7

### Frequency 25 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	747,4
7	747,5
8	746,9

9	747,3
10	747,1
11	746,7
12	746,8
13	747,2
14	747,7
15	747,5
16	757,6
17	746,9
18	746,8
19	747
20	746,9

### Frequency 30 Hz

Pengujian	Data
	Pembacaan RPM Menggunakan Tachometer
6	896,4
7	897
8	896,9
9	896,5
10	897,1
11	896,3
12	896,9
13	897,2
14	896,8
15	896,6
16	897,1

17	897,3
18	897,5
19	897,1
20	896,8

## Lampiran 6 : Perbandingan Current pada Interface dan Multimeter

### *Frequency 5 Hz*

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
6	0,47	0,37
7	0,47	0,37
8	0,47	0,37
9	0,47	0,37
10	0,47	0,37
11	0,47	0,37
12	0,47	0,37
13	0,47	0,37
14	0,47	0,37
15	0,47	0,37
16	0,47	0,37
17	0,47	0,37
18	0,47	0,37
19	0,47	0,37
20	0,47	0,37

### *Frequency 10 Hz*

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter

6	0,44	0,37
7	0,44	0,37
8	0,44	0,37
9	0,44	0,37
10	0,44	0,37
11	0,44	0,37
12	0,44	0,37
13	0,44	0,37
14	0,44	0,37
15	0,44	0,37
16	0,44	0,37
17	0,44	0,37
18	0,44	0,37
19	0,44	0,37
20	0,44	0,37

**Frequency 15 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
6	0,4	0,37
7	0,4	0,37
8	0,4	0,37
9	0,4	0,37
10	0,4	0,37
11	0,4	0,37
12	0,4	0,37
13	0,4	0,37
14	0,4	0,37
15	0,4	0,37

16	0,4	0,37
17	0,4	0,37
18	0,4	0,37
19	0,4	0,37
20	0,4	0,37

**Frequency 20 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
6	0,39	0,37
7	0,39	0,37
8	0,39	0,37
9	0,39	0,37
10	0,39	0,37
11	0,39	0,37
12	0,39	0,37
13	0,39	0,37
14	0,39	0,37
15	0,39	0,37
16	0,39	0,37
17	0,39	0,37
18	0,39	0,37
19	0,39	0,37
20	0,39	0,37

**Frequency 25 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
6	0,38	0,37

7	0,38	0,37
8	0,38	0,37
9	0,38	0,37
10	0,38	0,37
11	0,38	0,37
12	0,38	0,37
13	0,38	0,37
14	0,38	0,37
15	0,38	0,37
16	0,38	0,37
17	0,38	0,37
18	0,38	0,37
19	0,38	0,37
20	0,38	0,37

**Frequency 30 Hz**

Pengujian	Perbandingan <i>Current</i>	
	GUI	Multimeter
6	0,37	0,37
7	0,37	0,37
8	0,37	0,37
9	0,37	0,37
10	0,37	0,37
11	0,37	0,37
12	0,37	0,37
13	0,37	0,37
14	0,37	0,37
15	0,37	0,37
16	0,37	0,37

17	0,37	0,37
18	0,37	0,37
19	0,37	0,37
20	0,37	0,37