

**RANCANG BANGUN SISTEM *INTERNET OF THINGS*  
UNTUK MONITORING KINERJA MOTOR DC BERBASIS  
DATA VIBRASI**

**SKRIPSI**

oleh:  
**DIMAS ADI WIBISANA**  
**195090807111024**



**DEPARTEMEN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN  
ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2024**

*Halaman ini sengaja dikosongkan*

**RANCANG BANGUN SISTEM *INTERNET OF THINGS*  
UNTUK MONITORING KINERJA MOTOR DC BERBASIS  
DATA VIBRASI**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains  
dalam bidang fisika

oleh:  
**DIMAS ADI WIBISANA**  
**195090807111024**



**DEPARTEMEN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN  
ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2024**

*Halaman ini sengaja dikosongkan*

## LEMBAR PENGESAHAN

### RANCANG BANGUN SISTEM *INTERNET OF THINGS* UNTUK MONITORING KINERJA MOTOR DC BERBASIS DATA VIBRASI

oleh:

Dimas Adi Wibisana

195090807111024

Setelah dipertahankan di depan Majelis Penguji pada tanggal 24  
Juni 2024 dan dinyatakan memenuhi syarat untuk memperoleh  
gelar Sarjana Sains dalam bidang instrumentasi

Pembimbing I



Prof. Dr. Eng. Agus Naba, S.Si., M.T.  
NIP. 197208061995121001

Pembimbing II



Prof. Dr. -Ing. Setyawan P. Sakti, M. Eng  
NIP. 196508251990021001

Mengetahui,

Ketua Departemen Fisika  
Fakultas MIPA Universitas Brawijaya



*Halaman ini sengaja dikosongkan*

## **LEMBAR PERNYATAAN**

Saya yang bertanda tangan di bawah ini :

Nama : Dimas Adi Wibisana

NIM : 195090807111024

Departemen : Fisika / Instrumentasi

Penulis Skripsi berjudul :

### **RANCANG BANGUN SISTEM *INTERNET OF THINGS* UNTUK MONITORING KINERJA MOTOR DC BERBASIS DATA VIBRASI**

Dengan ini menyatakan :

1. Skripsi ini adalah benar-benar karya saya sendiri bukan hasil plagiat dari karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka, semata-mata digunakan sebagai acuan atau referensi.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

**Malang, 24 Juni 2024**

**Yang menyatakan,**



**DIMAS ADI WIBISANA**  
**195090807111024**

*Halaman ini sengaja dikosongkan*

# **RANCANG BANGUN SISTEM *INTERNET OF THINGS* UNTUK MONITORING KINERJA MOTOR DC BERBASIS DATA VIBRASI**

## **ABSTRAK**

Pengukuran kinerja motor DC tidak bisa diketahui secara langsung dan mudah, diperlukan sebuah pengukuran yang mampu mengetahui kinerjanya. Salah satu parameter kerusakan dapat dilihat dari getaran atau vibrasi yang terjadi pada motor DC. Dalam penelitian ini dilakukan analisa kinerja pada motor DC melalui analisis getaran dengan melihat puncak magnitude pada suatu frekuensi dominan pada grafik spektrum vibrasi. Setelah didapatkan data getaran atau vibrasi, proses monitoring atau pengukuran menggunakan *internet of things* dan mengetahui hasil pengujian sistem monitoring kinerja motor DC. Data yang telah didapatkan dikirimkan menggunakan protokol komunikasi MQTT untuk menghubungkan perangkat dengan internet di NodeRed. Selanjutnya data disimpan pada sebuah database MySql dan ditampilkan pada tampilan visualisasi dashboard Grafana. Hasil penelitian menunjukkan sistem *monitoring* ini mampu menampilkan dan mengirimkan data vibrasi untuk membandingkan kinerja motor DC yang baik dan rusak dengan analisis spektrum vibrasi.

**Kata kunci:** Monitoring, Kerusakan, Spektrum Vibrasi, *Internet of Things*, Motor DC, MQTT, NodeRed, Grafana.

*Halaman ini sengaja dikosongkan*

# **DESIGN OF AN INTERNET OF THINGS SYSTEM FOR MONITORING DC MOTOR PERFORMANCE BASED ON VIBRATION DATA**

## **ABSTRACT**

DC motor performance measurements cannot be known directly and easily, a measurement is needed that is able to determine its performance. One of the damage parameters can be seen from the vibration or vibration that occurs in the DC motor. In this study, the performance of the DC motor was analyzed through vibration analysis by looking at the peak magnitude at a dominant frequency on the vibration spectrum graph. After obtaining vibration or vibration data, the monitoring or measurement process uses the internet of things and knows the test results of the DC motor performance monitoring system. The data that has been obtained is sent using the MQTT communication protocol to connect the device with the internet on NodeRed. Furthermore, the data is stored in a MySQL database and displayed on the Grafana dashboard visualization display. The results showed that this monitoring system is able to display and send vibration data to compare the performance of good and damaged DC motors with vibration spectrum analysis.

**Keyword:** Monitoring, Fault, Vibration Spectrum, *Internet of Things*, Motor DC, MQTT, NodeRed, Grafana.

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Segala puji bagi dan syukur bagi Allah SWT atas rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan Tugas Akhir ini dengan judul “**RANCANG BANGUN SISTEM INTERNET OF THINGS UNTUK MONITORING KINERJA MOTOR DC BERBASIS DATA VIBRASI**”. Selama pembuatan laporan Skripsi penulis mendapatkan banyak bimbingan, bantuan, dorongan, dan masukan dari berbagai pihak yang terkait. Skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Sains (S.Si) dibidang Instrumentasi, yang dalam pembuatannya tidak lepas dari bantuan dan bimbingan dari berbagai pihak. Penulis ingin mengucapkan syukur dan terima kasih sebanyak-banyaknya kepada :

1. Allah SWT yang telah memberikan rahmat dan anugrah-Nya serta bantuannya sehingga penulis dapat menyelesaikan perkuliahan dengan baik dan dapat berakhir dengan baik.
2. Kedua orang tua yang telah memberikan do'a dan dukungan untuk penulis sehingga dapat melaksanakannya dengan lancar.
3. Prof. Dr. Eng. Agus Naba, S.Si., M.T. selaku Ketua Program Studi Instrumentasi Universitas Brawijaya dan selaku dosen pembimbing pertama yang selalu memberikan bimbingan dan saran dalam penyusunan Skripsi.
4. Prof Dr.-Ing. Setyawan P. Sakti, M. Eng. Selaku dosen pembimbing kedua yang telah memberikan bimbingannya, saran, arahan, kritik kepada penulis selama penyusunan Skripsi.
5. Seluruh dosen Departemen Fisika dan Program Studi Instrumentasi Universitas Brawijaya yang telah memberikan pengetahuan untuk penulis sebagai bekal penyusunan Skripsi.
6. Serta seluruh teman dan pihak yang tidak bisa disebutkan satu persatu selama penyusunan Tugas Akhir dan penulisan laporan.

Dalam penulisan laporan ini, penulis menyadari bahwa laporan ini jauh dari kata sempurna dikarenakan kekurangannya pengetahuan selama berjalannya penyusunan Skripsi. Oleh karena itu, penulis mengharapkan kemakluman terhadap laporan yang sudah dibuat, serta

penulis mengaharapkan kritik dan masukan untuk laporan Skripsi ini agar dapat membantu pembaca untuk memperoleh informasi yang baik dan bermanfaat. Sehingga, dapat mampu menciptakan laporan yang lebih baik.

Semoga Allah SWT memberikan balasan berlipat atas bantuan dan dukungan yang telah diberikan. Dan semoga skripsi ini bermanfaat khususnya bagi penulis, dan pembaca.

Malang, Mei 2024



Dimas Adi Wibisana  
195090807111024

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
LEMBAR PERNYATAAN .....	vii
ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xvi
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxiii
DAFTAR LAMPIRAN .....	xxv
BAB I PENDAHULUAN .....	1
1.1.    Latar Belakang.....	1
1.2.    Rumusan Masalah.....	2
1.3.    Batasan Masalah .....	3
1.4.    Tujuan Penelitian.....	3
1.5.    Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1.    Motor Listrik.....	5
2.1.1. Motor Listrik DC ( <i>Dirrect Current</i> ) .....	6
2.2.    Vibrasi .....	7
2.2.1. Parameter Vibrasi .....	11
2.3.    Analisis Vibrasi .....	13
2.4.    Sinyal Vibrasi .....	13
2.4.1. Domain Waktu.....	14
2.4.2. Fast Fourier Transform .....	14
2.4.3. Domain Frekuensi.....	15
2.5.    Internet of Things .....	16
2.6.    Message Queing Telemetry Transport.....	17
2.7.    NodeRed .....	19
2.8.    PHP & MySql.....	20
2.9.    Grafana .....	21
2.10.    STM32F401CCU6.....	22
2.11.    ESP32 .....	23
2.12.    Sensor .....	24
2.12.1.    MPU6050.....	25
BAB III METODOLOGI .....	29

3.1.	Tempat dan Waktu Penelitian .....	29
3.2.	Alat dan Bahan.....	29
3.3.	Prosedur Penelitian .....	30
3.3.1.	Studi Literatur .....	31
3.3.2.	Perancangan Sistem .....	32
3.3.3.	Perancangan Sistem Perangkat Keras .....	33
3.3.4.	Perancangan Sistem Perangkat Lunak .....	35
3.3.5.	Pengujian.....	47
3.3.6.	Pengambilan Data dan Analisis Data.....	49
	BAB IV HASIL DAN PEMBAHASAN .....	53
4.1.	Desain Alat.....	53
4.1.1.	Hasil Desain Perangkat Keras.....	53
4.1.2.	Hasil Desain Program .....	55
4.2.	Pengujian Pengiriman Data Vibrasi .....	64
4.3.	Pengujian Sistem Internet of Things .....	66
4.3.1.	Pengujian Konfigurasi NodeRed.....	66
4.3.2.	Pengujian Data Database PhpMysql .....	69
4.3.3.	Pengujian Dashboard Grafana .....	70
4.4.	Pengujian Pengukuran Data .....	71
4.4.1.	Beban 0 Gram .....	72
4.4.2.	Beban 20 Gram .....	82
4.4.3.	Beban 40 Gram .....	92
4.4.4.	Beban 60 Gram .....	102
4.4.5.	Beban 80 Gram .....	112
4.4.6.	Beban 100 Gram .....	122
4.5.	Pembahasan.....	132
	BAB V PENUTUP .....	137
5.1.	Kesimpulan .....	137
5.2.	Saran .....	137
	DAFTAR PUSTAKA .....	140

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1. Bagian motor listrik.....	5
Gambar 2.2. Bentuk fisik motor DC .....	6
Gambar 2.3. Motor DC double bearing 555 .....	7
Gambar 2.4. Gerakan pegas-massa terhadap waktu.....	8
Gambar 2.5. Bentuk gelombang dalam gerak harmonik sederhana...	9
Gambar 2.6. Sinyal perpindahan, kecepatan, percepatan.....	11
Gambar 2.7. Data getaran terhadap waktu .....	14
Gambar 2.8. Grafik data domain frekuensi.....	16
Gambar 2.9. Internet of things .....	17
Gambar 2.10. Message queing telemetry transport dengan mqtt sebagai pusat protokol, publish pengirim dan subscribe penerima ...	19
Gambar 2.11. Tampilan halaman utama NodeRed .....	20
Gambar 2.12. Tampilan utama PHPMyadmin.....	21
Gambar 2.13. Tampilan dashboard Grafana .....	22
Gambar 2.14. STM32F401CCU6 .....	23
Gambar 2.15. ESP32.....	24
Gambar 2.16. MPU6050 .....	26
Gambar 3.1. Diagram alir .....	31
Gambar 3.2. Diagram blok sistem .....	32
Gambar 3.3. Skematik rangkaian alat .....	34
Gambar 3.4. Bentuk papan sirkuit rangkaian.....	35
Gambar 3.5. <i>Flowchart</i> program ekstensi.....	37
Gambar 3.6. <i>Flowchart</i> program STM32F401CCU6 .....	38
Gambar 3.7. <i>Flowchart</i> program ESP32.....	39
Gambar 3.8. Tampilan flow NodeRed .....	40
Gambar 3.9. Halaman download NodeRed .....	41
Gambar 3.10. Tampilan command NodeRed.....	41
Gambar 3.11. Tampilan halaman NodeRed.....	42
Gambar 3.12. Halaman download database .....	43
Gambar 3.13. File <i>control panel</i> web php .....	43
Gambar 3.14. Tampilan web PhpMysql .....	44
Gambar 3.15. Basis database PhpMysql.....	44
Gambar 3.16. Grafana <i>download</i> .....	45
Gambar 3.17. Tampilan web grafana.....	45
Gambar 3.18. Tampilan <i>new dashboard</i> .....	46
Gambar 3.19. Tampilan pengaturan <i>dashboard</i> .....	46

Gambar 3.20. Tampilan pengaturan panel.....	47
Gambar 4.1. Rangkain PCB .....	54
Gambar 4.2. <i>Casing</i> rangkaian .....	54
Gambar 4.3. Pengiriman data vibrasi motor baru .....	65
Gambar 4.4. Pengiriman data vibrasi motor rusak .....	65
Gambar 4.5. <i>Flow</i> pada NodeRed.....	67
Gambar 4.6. Pengaturan message queing telemetry transport.....	68
Gambar 4.7. Node database Mysql.....	69
Gambar 4.8. <i>Database</i> Mysql.....	70
Gambar 4.9. Hasil data masuk <i>database</i> .....	70
Gambar 4.10. Hasil data pada <i>dashboard</i> .....	71
Gambar 4.11. Data vibrasi motor baru 0 gram .....	72
Gambar 4.12. Percepatan total vibrasi motor baru 0 gram .....	73
Gambar 4.13. Spektrum vibrasi motor DC baru 0 gram.....	75
Gambar 4.14. Data vibrasi motor rusak 0 gram.....	76
Gambar 4.15. Percepatan total vibrasi motor rusak 0 gram.....	77
Gambar 4.16. Spektrum vibrasi motor DC rusak 0 gram .....	79
Gambar 4.17. Perbandingan spektrum vibrasi.....	81
Gambar 4.18. Data vibrasi motor baru 20 gram .....	82
Gambar 4.19. Percepatan total vibrasi motor baru 20 gram .....	83
Gambar 4.20. Spektrum vibrasi motor DC baru 20 gram.....	85
Gambar 4.21. Data vibrasi motor rusak 20 gram.....	86
Gambar 4.22. Percepatan total vibrasi motor rusak 20 gram.....	87
Gambar 4.23. Spektrum vibrasi motor DC rusak 20 gram .....	89
Gambar 4.24. Perbandingan spektrum vibrasi.....	91
Gambar 4.25. Data vibrasi motor baru 40 gram .....	92
Gambar 4.26. Percepatan total vibrasi motor baru 40 gram .....	93
Gambar 4.27. Spektrum vibrasi motor DC baru 40 gram.....	95
Gambar 4.28. Data vibrasi motor rusak 40 gram.....	96
Gambar 4.29. Percepatan total vibrasi motor rusak 40 gram.....	97
Gambar 4.30. Spektrum vibrasi motor DC rusak 40 gram .....	99
Gambar 4.31. Perbandingan spektrum vibrasi.....	101
Gambar 4.32. Data vibrasi motor baru 60 gram .....	102
Gambar 4.33. Percepatan total vibrasi motor baru 60 gram .....	103
Gambar 4.34. Spektrum vibrasi motor DC baru 60 gram.....	105
Gambar 4.35. Data vibrasi motor rusak 60 gram.....	106
Gambar 4.36. Percepatan total vibrasi motor rusak 60 gram.....	107
Gambar 4.37. spektrum vibrasi motor DC rusak 60 gram.....	109

Gambar 4.38. Perbandingan spektrum vibrasi .....	111
Gambar 4.39. Data vibrasi motor baru 80 gram .....	112
Gambar 4.40. Percepatan total vibrasi motor baru 80 gram .....	113
Gambar 4.41. Spektrum vibrasi motor DC baru 80 gram .....	115
Gambar 4.42. Data vibrasi motor rusak 80 gram .....	116
Gambar 4.43. Percepatan total vibrasi motor rusak 80 gram.....	117
Gambar 4.44. Spektrum vibrasi motor DC rusak 80 gram .....	119
Gambar 4.45. Perbandingan spektrum vibrasi .....	121
Gambar 4.46. Data vibrasi motor baru 100 gram.....	122
Gambar 4.47. Percepatan total vibrasi motor baru 100 gram .....	123
Gambar 4.48. Spektrum vibrasi motor DC baru 100 gram .....	125
Gambar 4.49. Data vibrasi motor rusak 100 gram .....	126
Gambar 4.50. Percepatan total vibrasi motor rusak 100 gram.....	127
Gambar 4.51. Spektrum vibrasi motor DC rusak 100 gram .....	129
Gambar 4.52. Perbandingan spektrum vibrasi .....	131
Gambar 4.53. Perbandingan magnudio motor DC.....	133
Gambar 4.54. Perbandingan Frekuensi motor DC .....	134

*Halaman ini sengaja dikosongkan*

## **DAFTAR TABEL**

Tabel 2.1. Spesifikasi Motor DC 555 Double Bearing .....	7
Tabel 2.2. Speisifikasi STM32F401CCU6 .....	23
Tabel 2.3. Spesifikasi ESP32 .....	24
Tabel 2.4. Spesifikasi MPU6050 .....	26
Tabel 3.1. Daftar alat dan bahan perangkat keras .....	29
Tabel 3.2. Daftar alat dan bahan perangkat lunak.....	29
Tabel 4.1. Tabel perbandingan data motor baru dan motor rusak..	132

*Halaman ini sengaja dikosongkan*

## **DAFTAR LAMPIRAN**

Lampiran 1 Data Penelitian .....	145
Lampiran 2 Foto Dokumentasi .....	152

*Halaman ini sengaja dikosongkan*

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Dalam dunia industri saat ini peralatan-peralatan dituntut semakin efisien untuk menunjang proses produksinya. Kebanyakan peralatan industry dipasok oleh benda bergerak yang memiliki gerakan yang linier maupun berputar dengan ditenagai penggerak utama yang menghasilkan tenaga mekanik dengan energi yang berasal dari listrik, uap, panas, gas, dan lain-lain. Contoh benda bergerak yang sering dipakai di industry salah satunya adalah motor listrik. Motor listrik merupakan sebuah instrument yang mampu mengubah energi listrik menjadi energi mekanik yang membutuhkan catu daya DC maupun sumber listrik AC. Motor listrik belakangan ini dikenal menjadi salah satunya peralatan yang paling penting dan penggunaanya meningkat dengan sangat cepat. Untuk mencapai hasil yang memuaskan, peralatan industry perlu dilakukannya pengontrolan dan pengawasan secara bertahap yang bisa di control secara jarak jauh. Kegiatan tersebut lebih praktis, tepat, dan berdaya guna tinggi dalam pengontrolan dan pengawasan peralatan industri (Gurocak, 2015).

Dalam penggunaannya motor DC sering mengalami permasalahan yang mengganggu dan menghambat proses operasionalnya. Adapun gangguan yang terjadi pada motor DC dapat di deteksi melalui vibrasi. Vibrasi merupakan parameter yang paling umum di zaman modern pada mesin berputar. Indikasi vibrasi akan mengukur kondisi dari mesin berputar, yang menandakan terjadinya gangguan dan kerusakan. Vibrasi dapat menunjukkan peringatan dan informasi secara bertahap yang mampu berpotensi merusak mesin, sehingga dapat membantu sebuah keputusan untuk melakukan perbaikan atau pemeriksaan pada mesin (Adams, 2000).

Umumnya getaran yang berasal dari suatu mesin atau benda bergerak merupakan sesuatu hal yang tidak disukai. Selain gangguan getaran, faktor yang mempengaruhi kinerja suatu motor yaitu kestabilan putaran motor. Kestabilan putaran motor dapat membantu untuk mengetahui kualitas motor dengan putaran yang stabil maka memiliki kualitas yang baik. Jika putaran tidak stabil maka motor tersebut memiliki kualitas yang kurang baik. Meskipun motor tersebut dapat berputar, putaran motor yang tidak stabil pada porosnya maka motor tersebut tidak dapat melakukan pekerjaannya dengan

maksimal. Sehingga, diperlukan pemantauan kestabilan motor untuk dapat melihat kinerja dari motor yang diukur (Adams, 2000).

Analisis vibrasi digunakan untuk menentukan kondisi yang terjadi pada sebuah mesin berputar yang dapat mendiagnosa terjadinya kerusakan atau kegagalan pada mesin. Keuntungan utamanya adalah analisis getaran dapat mengidentifikasi masalah yang terjadi sebelum menjadi terlalu serius dan menyebabkan waktu berhenti yang tidak terjadwal. Pemantauan getaran secara rutin dapat mendeteksi terjadinya ketidakseimbangan, ketidak lurusan poros, mendeteksi bantalan yang membburuk atau rusak, kelonggaran mekanis, dan roda gigi yang aus atau rusak. Semua mesin yang berputar akan menghasilkan getaran yang merupakan fungsi dari dinamika mesin sendiri. Mengukur magnitude getaran pada frekuensi tertentu dapat memberikan informasi penting tentang keakuratan kerusakan yang terjadi. Pengukuran getaran adalah metode yang efektif dan tidak mengganggu untuk memantau kondisi mesin selama start-up, shutdown, dan operasi normal. Analisis getaran dapat digunakan untuk meningkatkan performa peralatan secara signifikan yang mencakup penyelaran dan penyeimbangan yang lebih tepat, perbaikan dengan kualitas yang baik, dan mampu menurunkan tingkat getaran pada mesin(Scheffer & Girdhar, 2004).

Pemanfaatan *internet of things* (IoT) dirasa sangat memadai dalam era moderenisasi sekarang ini untuk pemantauan berbagai macam alat elektronik yang dapat dilakukan hanya dengan menggunakan perangkat elektronik dan jaringan internet sebagai jembatan penghubung antara peralatan dan system control yang digunakan. IoT merupakan salah satu teknologi yang dapat dikembangkan dan mempermudah dalam berbagai hal bagi manusia. Dari uraian permasalahan di atas, penulis akan membuat sebuah rangkaian elektronika yang mampu mendeteksi kinerja pada motor listrik dengan menganalisis dari vibrasi atau getaran yang dihasilkan oleh motor listrik dengan menggunakan sensor accelerometer dan pemantauan secara online, sehingga dapat mengetahui dan perbedaan dari motor listrik yang berbeda (Serpanos & Wolf, 2017).

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan terdapat beberapa masalah, yaitu sebagai berikut :

1. Bagaimana membangun sistem komunikasi *Internet of Things* untuk monitoring kinerja motor DC ?
2. Bagaimana mengetahui kondisi motor DC melalui spektrum vibrasi menggunakan mikrokontroler STM32F401CCU6 dan ESP32 ?

### **1.3. Batasan Masalah**

Penelitian ini diupayakan untuk membatasi ruang lingkup masalah dari pembahasan yang terlalu luas atau lebar yang bisa mengakibatkan penulis tidak terfokus dari materi yang terdapat pada latar belakang. Sehingga, penulis mengupayakan agar hanya terfokus pada pembahasan sebagai berikut :

1. Penggunaan mikrokontroler STM32F401CCU6 dan ESP32, serta jaringan internet sebagai monitoring kinerja motor DC.
2. Penerapan spektrum vibrasi sebagai indikator kinerja yang dihasilkan oleh motor Dc.
3. Perbandingan antara motor yang tidak layak dan masih layak dipakai.

### **1.4. Tujuan Penelitian**

Dengan mengacu berdasarkan latar belakang serta rumusan masalah penelitian yang ada maka tujuan penulisan penelitian Tugas Akhir ini adalah :

1. Untuk merancang dan membangun sebuah desain sistem *Internet of Things* untuk monitoring kinerja motor DC.
2. Untuk mengetahui kondisi motor DC melalui spektrum vibrasi menggunakan mikrokontroler STM32F401CCU6 dan ESP32.

### **1.5. Manfaat Penelitian**

Penelitian ini bermanfaat untuk memberikan kemudahan dalam melakukan pemantauan kinerja motor DC dengan melalui jaringan internet. Mengetahui tentang perbedaan kinerja motor DC melalui metode vibrasi berdasarkan data vibrasi. Mampu melakukan prediksi tentang terjadinya kerusakan sehingga dapat meminimalisir biaya dan waktu yang dikeluarkan.

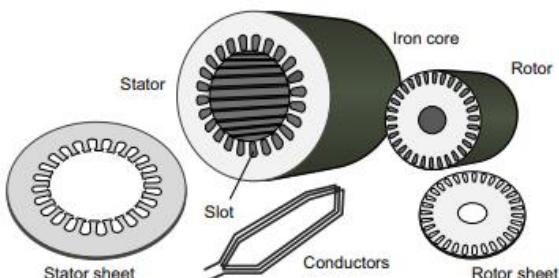
*Halaman ini sengaja dikosongkan*

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Motor Listrik

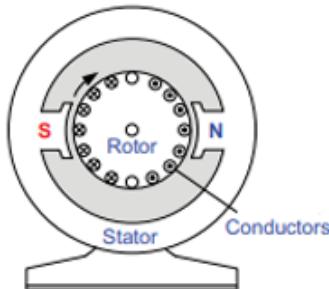
Motor listrik adalah sebuah perangkat yang berfungsi untuk mengubah energi listrik menjadi energi mekanik. Pada dasarnya motor listrik dibagi menjadi 2 jenis yaitu, motor listrik arus DC dan motor listrik arus AC. Motor listrik pertama kali diciptakan oleh Michael Faraday atas penemuan elektromagnetik induksi yang secara bersamaan telah mengoperasikan motor dengan sumber arus DC pertama kali sebagai purwarupa sudut torsi. Setelah berkembangnya motor DC, Nikola Tesla mengembangkan motor listrik dengan arus AC yang termasuk synchronous motor dan motor induksi. Gambar 2.1. merupakan sebuah motor listrik memiliki 2 bagian utama, yaitu bagian yang diam disebut *stator* dan bagian yang berputar disebut *rotor* serta terdapat air gap di antara *stator* dan *rotor* yang berfungsi untuk *rotor* dapat berputar. *Stator* dan bagian *rotor* masing-masing memiliki sirkuit listrik dan magnetik. *Stator* dan *rotor* dibangun dengan inti besi di mana fluks magnetik yang diciptakan oleh arus kumparan yang akan mengalir bersamaan dengan kumparan konduktor arus yang akan membuat sebuah arus listrik dan membuat inti besi menjadi medan magnet sehingga, membentuk sebuah elektromagnetik di dalam *stator* dan *rotor*. Medan magnet yang dihasilkan akan membentuk sebuah gaya untuk menimbulkan gerakan torsi pada *rotor* dan menyebabkan *rotor* berputar (Kim, 2017).



Gambar 2.1. Bagian motor listrik

### 2.1.1. Motor Listrik DC (*Direct Current*)

Gambar 2.2 adalah motor yang akan bekerja jika di suplai dengan tegangan searah atau tegangan DC. *Rotor* berputar dengan menggunakan gaya yang dihasilkan pada konduktor pembawa arus yang ditempatkan di medan magnet yang dibuat oleh *stator*. Terdapat 2 medan magnet stasioner pada motor DC, pertama adalah medan magnet di *stator* yang dihasilkan oleh kumparan medan arus. Satu medan magnet lainnya adalah medan magnet yang berada di *rotor* berasal dari arus di dalam konduktor *rotor*. Dengan demikian, dua medan magnet stasioner akan berinteraksi satu sama lain menghasilkan torsi yang akan membuat *rotor* akan berputar secara terus menerus. Medan magnet pada rotor akan diam meskipun rotasi rotor. Hal ini disebabkan oleh *brush* dan komutator, di mana distribusi arus di dalam konduktor *rotor* selalu dibuat sama termasuk saat *rotor* berputar. Sehingga, medan magnet *rotor* tidak akan berputar bersamaan dengan *rotor* (Kim, 2017).



**Gambar 2.2.** Bentuk fisik motor DC

Pada penelitian ini penulis menggunakan motor DC 555 motor double bearing 12-24 V yang dapat dilihat pada Gambar 2.3.



**Gambar 2.3.** Motor DC double bearing 555

Spesifikasi dari motor DC 555 double bearing dapat dilihat pada tabel 2.1.

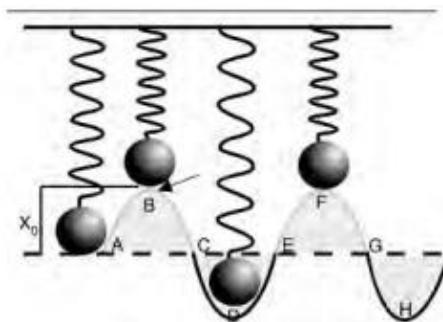
**Tabel 2.1.** Spesifikasi Motor DC 555 Double Bearing

Keterangan	Nilai
Panjang	76 mm
Diameter	35,8 mm
Kecepatan	12 V – 6000 rpm / 24 V – 12000 rpm
Arus	12 V – 210 mA / 24 V – 260 mA
Diameter Tangkai	3,17 mm
Jarak Lubang Depan	25 mm

## 2.2. Vibrasi

Suatu gerakan yang terjadi secara berulang-ulang akibat diberikannya gaya setelah interval waktu atau titik kesetimbangan secara gerakan maju mundur, atas bawah dari posisi diam disebut vibrasi (getaran). Vibrasi secara umum, mencakup energi potensial dari pegas atau elastisitas, energi kinetic dari massa atau inersia, dan energi yang hilang secara bertahap atau teredam. Vibrasi pada suatu sistem akan melakukan perpindahan energi potensialnya ke energi kinetik dan energi kinetik ke energi potensial secara bergantian. Jika sistem mengalami peredaman, beberapa energi dihamburkan dalam

setiap siklus getaran dan harus diganti dengan sumber eksternal jika keadaan getaran stabil yang harus dipertahankan. Sebagian besar peralatan-peralatan listrik tidak menginginkan vibrasi terjadi karena dapat meningkatkan lonjakan tegangan dan mengurangi energi untuk kinerja peralatan itu sendiri. Vibrasi dapat dilihat melalui perubahan secara periodic dari suatu besaran yang dapat dilihat pergerakan massa sehubungan dengan waktu, seperti ditunjukkan pada Gambar 2.4. sebuah pegas dilekatkan pada benda diam dan bawahnya dilekatkan sebuah beban (Scheffer & Girdhar, 2004).



**Gambar 2.4.** Gerakan pegas-massa terhadap waktu

Gerakan pada massa dari posisi netralnya menuju batas atas gerakan, kembali lagi melalui posisi netralnya kemudian batas bawah gerakan dan kembali ke posisi netralnya, sehingga mewakili satu siklus gerak. Satu siklus gerak ini berisi semua informasi yang diperlukan untuk mengukur getaran pada sistem ini. Gerakan massa yang bergerak secara berkelanjutan hanya akan mengulangi siklus yang sama dengan gerakan awal. Gerakan ini dapat disebut periodic dan harmonic, serta hubungan antara perpindahan massa dengan waktu dapat dinyatakan dengan persamaan sinusoidal sebagai berikut :

$$X = X_0 \sin \omega t \quad (2.1)$$

Persamaan 2.1 merupakan perpindahan massa pada saat waktu tertentu,  $t$  merupakan waktu (detik),  $X_0$  merupakan perpindahan massa maksimum, dan  $\omega$  merupakan perpindahan sudut dengan rumus :

$$\omega = 2\pi f \quad (2.2)$$

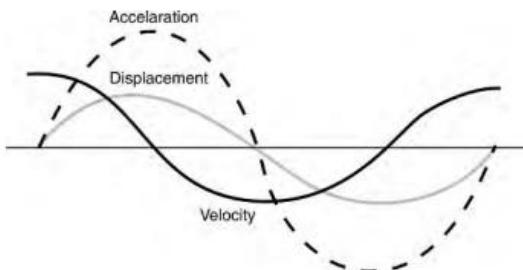
$f$  yang merupakan frekuensi merupakan jumlah getaran dalam satu detik. Pada persamaan 2.1  $X_0$  dapat diartikan sebagai amplitudo atau simpangan dan persamaan tersebut menyatakan perpindahan massa dengan waktu, maka untuk kecepatan dapat dituliskan dengan :

$$v = \frac{dX}{dt} = X_0 \omega \cos \omega t \quad (2.3)$$

Untuk percepatan dapat dituliskan sebagai berikut :

$$a = \frac{d(v)}{dt} = -X_0 \omega^2 \sin \omega t \quad (2.4)$$

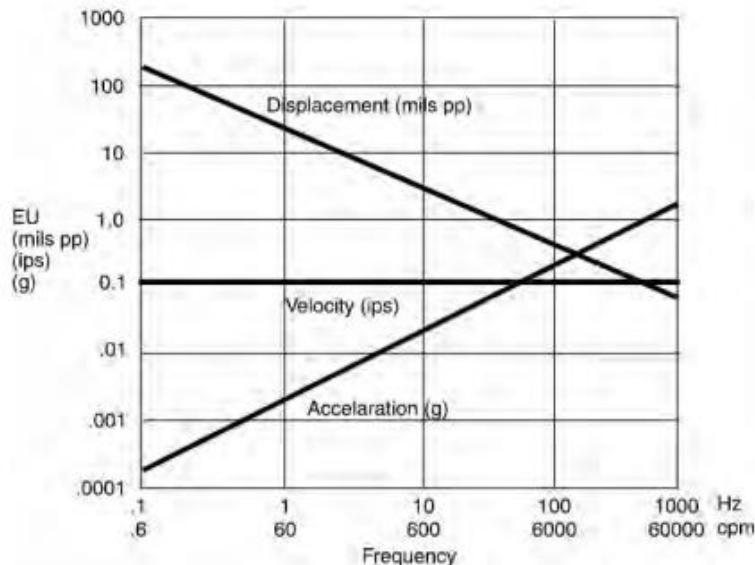
Dari persamaan perpindahan, kecepatan, dan percepatan dapat ditunjukkan hubungan antara persamaan tersebut yang membentuk hubungan antara masing-masing simpangan dalam gerak harmonik sederhana, dimana grafik perpindahan menampilkan grafik kurva sinus, kecepatan menampilkan grafik kurva kosinus, dan percepatan menampilkan grafik kurva sinus ditunjukkan pada Gambar 2.5.



**Gambar 2.5.** Bentuk gelombang dalam gerak harmonik sederhana

Karakteristik sinyal perpindahan, kecepatan, dan percepatan getaran diukur untuk menentukan tingkat keparahan getaran yang disebut sebagai amplitudo getaran bentuk perbedaan dapat dilihat pada Gambar 2.6. Dalam hal operasi mesin, amplitudo getaran merupakan indikator pertama yang menunjukkan seberapa baik atau buruk kondisi mesin. Hubungan antara percepatan, kecepatan, dan perpindahan dalam hal amplitudo getaran dan kinerja mesin dialkukannya kembali pengukuran dan analisis data yang harus digunakan. Getaran di bawah 10 Hz akan menghasilkan getaran yang

sangat kecil dalam hal percepatan, getaran sedang dalam hal kecepatan, dan getaran yang relatif besar dalam hal perpindahan. Oleh karena itu, perpindahan digunakan dalam rentang ini. Dalam rentang frekuensi tinggi, nilai percepatan memberikan nilai yang lebih besar dibandingkan kecepatan atau perpindahan. Oleh karena itu, untuk frekuensi di atas 1000 Hz, unit pengukuran yang disarankan untuk getaran adalah percepatan. Jika amplitudo yang diterima antara 10 Hz dan 1000 Hz, kecepatan memberikan indicator yang baik tentang tingkat keparahan getaran. Kecepatan terlihat konstan pada rentang frekuensi 10-1000 Hz karena hubungan matematis antara percepatan, kecepatan, dan perpindahan dalam domain frekuensi tersebut. Pada rentang frekuensi tersebut, perpindahan memiliki amplitudo yang relatif besar dibandingkan percepatan dan kecepatan. Dalam domain frekuensi, amplitudo perpindahan menurun secara signifikan dengan peningkatan frekuensi, sementara amplitudo percepatan dan kecepatan tetap relatif konstan atau hanya sedikit berubah. Hal ini disebabkan oleh sifat karakteristik respons terhadap perubahan dari sistem mekanis. Pada frekuensi rendah di bawah 10 Hz, sistem mekanis cenderung menghasilkan perpindahan yang lebih besar, sementara percepatan dan kecepatan masih cukup rendah. Di sisi lain, pada frekuensi tinggi di atas 1000 Hz perpindahan relatif kecil, tetapi percepatan dan kecepatan meningkat secara signifikan (Rao, 2012).



**Gambar 2.6.** Sinyal perpindahan, kecepatan, percepatan

### 2.2.1. Parameter Vibrasi

Penggambaran gelombang tidak lengkap dari sebuah parameter untuk membentuk gelombang itu sendiri yang meliputi beberapa parameter sebagai berikut :

#### a. Frekuensi

Frekuensi didefinisikan sebagai jumlah getaran yang terjadi dalam satuan waktu dengan satuannya Hertz (Hz). Dalam pengukuran satuan pada frekuensi menggunakan Revolutin Per Minute (RPM).

#### b. Amplitudo

Amplitudo dapat didefinisikan sebagai besaran simpangan terjauh dari titik kesetimbangan atau ketinggian puncak gelombang di atas garis referensi.

#### c. Panjang Gelombang

Panjang gelombang adalah jarak satu gelombang yang dapat dilihat dari satu siklus ke titik yang sesuai pada siklus berikutnya di garis referensi dan huruf Yunani  $\lambda$  (lambda) biasanya digunakan untuk menandakan panjang gelombang.

d. Bentuk Gelombang

Bentuk gelombang adalah representasi visual (grafik) dari nilai sesaat dari gerakan yang diplot terhadap waktu yang di bentuk oleh gelombang sinus dan kosinus.

e. *Peak to Peak*

Peak to Peak adalah selisih antara nilai puncak positif dan nilai puncak negatif dari suatu sinyal atau getaran. Besaran ini dihubungkan dengan besarnya tegangan maksimum yang terjadi atau dihubungkan dengan Lembah.

f. Root mean square (RMS)

RMS adalah nilai dari kuadrat dari setiap titik data dalam gelombang yang diambil rata-rata dari amplitudo yang dihasilkan oleh getaran mesin.

g. *Displacement*

Displacement dapat diartikan sebagai perubahan suatu posisi dari suatu objek yang melewati garis referensinya dengan satuan biasanya diukur dalam mikron (1/1000 milimeter). Dalam satuan imperial diukur dalam mils (mili inci-1/1000 inci). Pengukuran dalam getaran yang sering digunakan dalam displacement adalah peak-to-peak yaitu total jarak yang ditempuh oleh bagian yang bergetar, dari satu batas gerakan atas maksimum ke batas gerakan bawah maksimum lainnya.

h. Kecepatan

Kecepatan dapat didefinisikan sebagai turunan pertama dari displacement terhadap waktu. Saat massa bergerak akan menimbulkan sebuah kecepatan dengan maksimum melewati posisi netralnya. Kecepatan maksimum ini disebut sebagai puncak kecepatan getaran. Dalam pengukuran getaran mesin, disarankan menggunakan kecepatan root mean square (rms) sebagai unit standar pengukurannya. Hal ini digunakan untuk mendapatkan kriteria yang akan menentukan nilai efektif untuk berbagai fungsi kecepatan. Kecepatan-rms cenderung memberikan kandungan energi dalam sinyal getaran, sedangkan puncak kecepatan berkorelasi lebih baik dengan intensitas getaran.

i. Percepatan

Percepatan atau akselerasi merupakan perubahan kecepatan dalam satuan waktu. Sebagai contoh pada benda pegas bermassa, percepatan massa berada pada batas maksimum gerakan di mana

kecepatan massa adalah nol. Saat kecepatan mendekati nilai maksimum, akselerasi turun ke nol dan akan terus naik ke nilai maksimumnya pada batas gerakan maksimum lainnya. Percepatan atau akselerasi biasanya dinyatakan dalam  $g$ , yang merupakan percepatan yang dihasilkan oleh gaya gravitasi di permukaan bumi (Rao, 2012).

### **2.3. Analisis Vibrasi**

Analisis vibrasi digunakan untuk menentukan kondisi yang terjadi pada sebuah mesin berputar yang dapat mendiagnosa terjadinya kerusakan atau kegagalan pada mesin. Keuntungan utamanya adalah analisis getaran dapat mengidentifikasi masalah yang terjadi sebelum menjadi terlalu serius dan menyebabkan waktu berhenti yang tidak terjadwal. Dengan melakukan pemantauan getaran secara rutin dapat mendeteksi terjadinya ketidakseimbangan, ketidak lurusan poros, mendeteksi bantalan yang memburuk atau rusak, kelonggaran mekanis, dan roda gigi yang aus atau rusak. Semua mesin yang berputar akan menghasilkan getaran yang merupakan fungsi dari dinamika mesin sendiri. Mengukur magnitude getaran pada frekuensi tertentu dapat memberikan informasi penting tentang keakuratan kerusakan yang terjadi. Pengukuran getaran adalah metode yang efektif dan tidak mengganggu untuk memantau kondisi mesin selama start-up, shutdown, dan operasi normal. Analisis getaran dapat digunakan untuk meningkatkan performa peralatan secara signifikan yang mencakup penyelaran dan penyeimbangan yang lebih tepat, perbaikan dengan kualitas yang baik, dan mampu menurunkan tingkat getaran pada mesin. Analisis getaran biasanya digunakan terutama pada peralatan berputar dengan menggunakan sensor atau tranducer yang dipasang pada badan mesin maupun sekitar mesin, serta sebuah pemantauan vibrasi sebagai pengolah data hasil dari pengukuran untuk dapat di analisa atau mendiagnosis kerusakan yang terjadi (Scheffer & Girdhar, 2004).

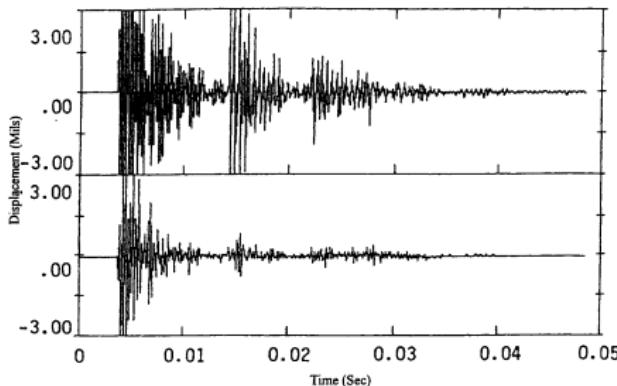
### **2.4. Sinyal Vibrasi**

Sinyal-sinyal yang telah diperoleh dari transducers pada pengukuran vibrasi merupakan sebuah kumpulan respon yang diberikan pada mesin-mesin tersebut. Data-data yang kompleks tersebut perlu dilakukan proses analisis untuk penguraian sinyal yang kompleks ini. Dua gelombang sinus dengan frekuensi berbeda akan

bersifat periodik tetapi bukan harmonik dari kedua frekuensi tersebut. Secara sederhana, sumber getaran dengan frekuensi yang berbeda dapat bergabung dan menghasilkan frekuensi komposit yang tidak sama dengan kedua sumber tersebut. Cara untuk menganalisis data getaran adalah domain waktu dan domain frekuensi (Mobley, 1999).

#### 2.4.1. Domain Waktu

Data getaran keseluruhan dapat ditampilkan pada perangkat perekam sinyal. Metode-metode ini menampilkan tingkat getaran relatif dalam kaitannya dengan waktu yang telah berlalu. Dengan menyesuaikan periode waktu dapat dengan cermat memisahkan masing-masing sumber unik yang membentuk tingkat getaran keseluruhan. Bentuk plot data getaran terhadap waktu sebagai domain waktu dapat ditunjukkan pada Gambar 2.7. (Mobley, 1999).



Gambar 2.7. Data getaran terhadap waktu

#### 2.4.2. Fast Fourier Transform

*Fast Fourier Transform* (FFT) adalah algoritma yang digunakan untuk melakukan transformasi Fourier diskrit pada suatu sinyal dengan cepat. Transformasi Fourier adalah teknik matematika yang memungkinkan kita untuk mengonversi suatu sinyal dari domain waktu ke domain frekuensi. Dalam domain waktu, sinyal direpresentasikan sebagai fungsi waktu. Sedangkan dalam domain frekuensi, sinyal direpresentasikan sebagai kombinasi dari beberapa gelombang sinusoidal dengan frekuensi yang berbeda. Dalam transformasi Fourier, sinyal dibagi menjadi beberapa komponen

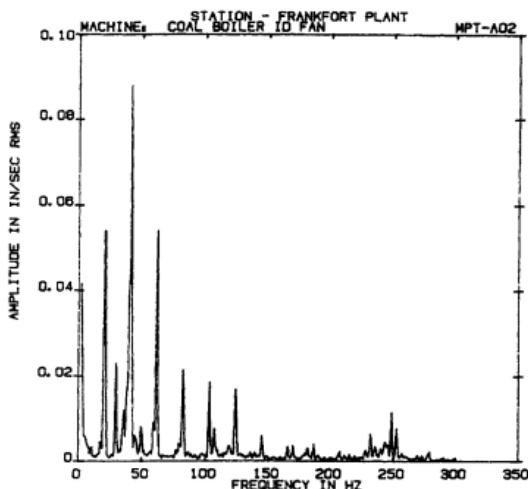
frekuensi. FFT memungkinkan untuk menghitung spektrum frekuensi dari sinyal secara cepat dengan memanfaatkan aljabar bilangan kompleks. Sebagai hasilnya, FFT membuat komputasi transformasi Fourier menjadi lebih efisien dibandingkan dengan metode sebelumnya seperti *Discrete Fourier Transform* (DFT). Pada dasarnya, FFT bekerja dengan mengambil sinyal waktu diskrit dan mengubahnya menjadi sinyal frekuensi diskrit. Sinyal waktu ini kemudian dipecah menjadi subbagian yang lebih kecil, dihitung transformasi Fourier diskrit dari setiap subbagian, dan kemudian dikombinasikan kembali menjadi transformasi Fourier dari seluruh sinyal (Rahman, 2011).

Fast Fourier Transform ini merupakan representasi dari Transformasi Fourier Diskrit (DFT) pada sinyal input  $x[n]$ .

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \quad (2.5)$$

#### 2.4.3. Domain Frekuensi

Domain frekuensi didapatkan dengan mengkonversikan data domain waktu menggunakan model matematika transformasi Fourier. frekuensi ini didapatkan dari beberapa kelipatan dari kecepatan gerakan mesin yang dinyatakan dalam putaran per menit (rpm) atau siklus per menit (cpm). Bentuk plot domain frekuensi dapat ditunjukkan pada Gambar 2.8. Sumbu Y pada diagram domain frekuensi menyatakan amplitudo dari variabel respon dapat menjadi perpindahan per satuan waktu yang terkait dengan frekuensi tertentu, sedangkan sumbu X menyatakan frekuensi dalam satuan (Hz) atau (rpm) yang dinormalisasi dengan kecepatan, perubahan kecepatan tidak akan mempengaruhi plot diagram (Mobley, 1999).



**Gambar 2.8.** Grafik data domain frekuensi

## 2.5. Internet of Things

*Internet of things* merupakan sebuah konsep di mana suatu benda atau objek ditanamkan teknologi-teknologi seperti system informasi, jaringan sensor dan *embedded system* dengan tujuan untuk berkomunikasi, mengendalikan, menghubungkan, dan bertukar data melalui beberapa perangkat dengan menggunakan jaringan internet sebagai jembatan penghubung. *IoT* memiliki hubungan yang erat dengan istilah *machine-to-machine* atau M2M. Seluruh alat yang memiliki kemampuan komunikasi M2M ini sering disebut dengan perangkat cerdas atau *smart device*. Perangkat cerdas ini diharapkan dapat membantu kinerja manusia dalam menyelesaikan berbagai urusan atau masalah yang ada. Salah satu contoh penggunaanya pada system industry menggunakan berbagai macam sensor untuk memantau proses industry, kualitas produk, dan keadaan peralatan. Misalnya pada motor listrik, termasuk sebuah permasalahan yang menggunakan sensor untuk memprediksi kegagalan atau kerusakan pada motor yang akan mendatang (Serpanos & Wolf, 2017).

*Internet of Things* dalam era digitalisasi industri, dapat disebutkan *Industrial Internet of Things* (IIoT) yang telah menjadi komponen penting dalam meningkatkan efisiensi dan keamanan dalam pemantauan proses industri. Penerapan teknologi IIoT, seperti

yang diilustrasikan dalam sistem pemantauan proses industri menggunakan modul Wi-Fi, menggabungkan sensor canggih dan teknologi komputasi awan untuk memantau berbagai parameter industri secara real-time dari jarak jauh. Konektivitas yang memprediksi peningkatan eksponensial dalam kekuatan pemrosesan dan pengurangan biaya perangkat elektronik yang berperan penting dalam memfasilitasi komunikasi antara perangkat di lingkungan industri. Integrasi sensor dan aktuator memungkinkan pemantauan dan kontrol yang lebih akurat dan responsif terhadap kondisi lingkungan membuka jalan bagi operasi yang lebih aman dan efisien. IIoT tidak hanya mengurangi kebutuhan untuk pengawasan manual tetapi juga memberikan kemampuan pemadaman instan dalam kasus kegagalan yang sangat penting untuk mencegah kecelakaan dan kerugian dalam industri. Secara keseluruhan, IIoT telah muncul sebagai katalis untuk inovasi dan optimalisasi dalam sektor industri, memperkuat fondasi untuk Revolusi Industri 4.0 (Karmakar et al., 2019).



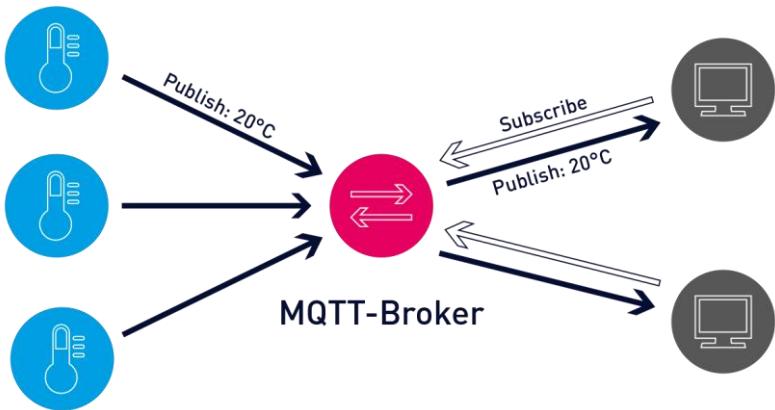
**Gambar 2.9.** Internet of things

## 2.6. Message Queuing Telemetry Transport

MQTT atau (*Message Queuing Telemetry Transport*) merupakan sebuah protocol pengiriman data menggunakan model *publisher* atau *subscriber* yang lebih cocok digunakan untuk IoT karena lebih sederhana, menghemat bandwidth dan mudah dirancang. Dengan hal ini, MQTT dapat digunakan dalam komunikasi M2M atau machine-to-machine dan *Internet of Things*. Protokol MQTT berjalan dengan

menggunakan TCP atau IP dua arah protocol yang membantu dalam pemrosesan data, pengontrolan, dan berbagi informasi. Protokol MQTT ini mampu membuat protokol pengiriman data dengan bandwidth jaringan yang terbatas serta lebih struktural dan lengkap. Pembuatan MQTT protokol ini bertujuan agar mengefisiensi bandwith, mengirim pesan secepat mungkin, meminimalisir encoding atau mengubah bentuk data agar dapat dibaca oleh sistem lain dan decoding atau mengubah kembali data yang telah di-encode menjadi bentuk aslinya yang membuat protokol dapat bekerja dengan daya dan media penyimpanan yang minim (Diono et al., 2021).

MQTT dinilai penting dalam menjalankan system IoT, dikarenakan protocol yang ringan dan efisien serta protocol MQTT memudahkan dalam enskripsi data dan autentikasi perangkat. Gambar 2.10 menunjukan MQTT menggunakan arsitektur *publish* dan *subscribe* yang tergantung pada topik yang merupakan salah satu komponen pembentuk pesan pada broker MQTT. *Publisher* MQTT yang bertanggung jawab untuk mengirimkan pesan ke broker MQTT. *publisher* menentukan topik yang sesuai dengan pesan yang dikirim. Setelah pesan dikirim, broker akan meneruskannya kepada *subscriber* yang telah terhubung ke topik yang sama. Pengirim dapat mengirim pesan tunggal atau serangkaian pesan ke broker. *Subscriber* adalah klien MQTT terhubung ke broker untuk menerima pesan dari topik tertentu. *Subscriber* menyampaikan permintaan langganan ke broker, dan broker akan mengirimkan pesan yang diterbitkan ke topik tersebut kepada *subscribe* yang telah terhubung. *Subscribe* dapat terhubung ke satu atau lebih topik, dan mereka akan menerima pesan yang diterbitkan ke topik yang terhubung (Hunkeler et al., 2008).



**Gambar 2.10.** Message queing telemetry transport dengan mqtt sebagai pusat protokol, publish pengirim dan subscribe penerima

## 2.7. NodeRed

NodeRed adalah perangkat lunak *open source* bagi pengembang untuk menghubungkan perangkat keras dengan *internet of things* dan layanan online lainnya berbasis JavaScript dan berdasarkan pada Node.js. NodeRed berfungsi sebagai basis pemrograman berbasis proses yang menciptakan "aliran data" dari sensor ke *cloud* dengan menghubungkan perangkat keras dan perangkat lunak. Hal ini memudahkan proses penulisan dan pemrosesan data, memungkinkan transfer data yang telah diproses ke sistem tingkat tinggi, seperti server SQL, sistem manajemen perusahaan, pengumpul data sentral, atau layanan berbasis *cloud* atau menampilkannya secara langsung. Gambar 2.11 merupakan tampilan awal NodeRed yang menyediakan antarmuka *Dashboard* yang memungkinkan pengguna membuat antarmuka yang menarik tanpa memerlukan pengetahuan khusus tentang HTML atau CSS. NodeRed menunjukkan keunggulan signifikan dalam bidang visualisasi data dan protocol komunikasi MQTT dibandingkan dengan solusi industri tradisional. Dengan teknologi ini, antarmuka dapat diperluas dengan mudah sesuai kebutuhan untuk memantau data sensor yang dianggap penting. Penggunaan NodeRed memungkinkan prototipe cepat, mempertahankan daya saing industri meskipun terjadi perkembangan teknologi yang cepat. NodeRed muncul sebagai alat yang sangat

berguna dan fleksibel dalam lingkungan industri, terutama untuk aplikasi yang terkait dengan IIoT dan Industri 4.0. Kemampuannya yang cepat dalam mengembangkan prototipe dengan fungsionalitas visualisasi data yang kuat dan integrasi mudah dengan protocol komunikasi MQTT (Ferencz & Domokos, 2020).

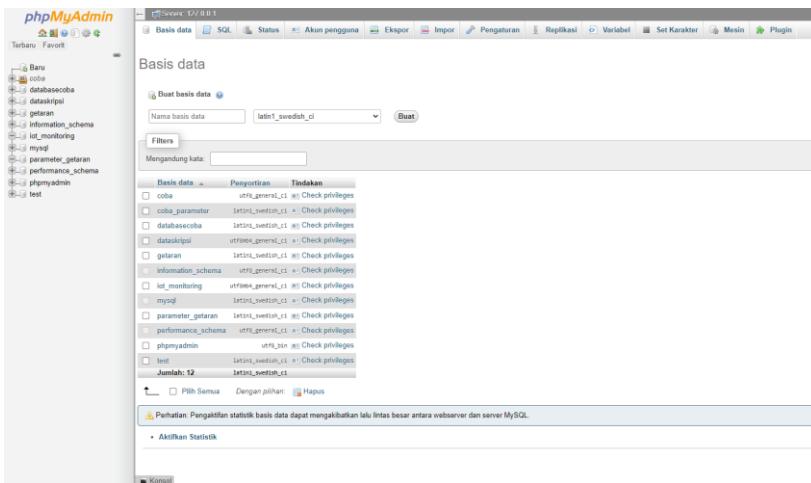


Gambar 2.11. Tampilan halaman utama NodeRed

## 2.8. PHP & MySql

PHP dan MySql adalah kolaborasi antara bahasa pemrograman dan database berbasis web dengan kemampuan dalam memproses data dinamis yang dapat dikerjakan secara mudah dan otomatis. Database Mysql merupakan kumpulan informasi atau data yang terintegrasi dengan baik yang tersimpan di dalam komputer. Untuk mengolah database diperlukan sebuah sistem perangkat lunak yaitu Database Management System atau DBMS yang merupakan suatu sistem perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol dan mengakses database secara praktis dan efisien. Pada Mysql, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Untuk mengelola database Mysql, DBMS dapat menggunakan menggunakan phpMyAdmin, pengguna dapat melakukan berbagai tugas administrasi database seperti membuat, mengedit, dan menghapus tabel, menjalankan interaksi SQL, mengelola pengguna dan hak akses, mengimpor dan mengekspor data. PhpMyAdmin menyediakan antarmuka pengguna

grafis yang mudah digunakan dan mudah diapahami, yang memungkinkan pengguna untuk berinteraksi dengan database MySQL tanpa perlu menulis kode SQL secara langsung. Dengan phpMyAdmin, pengguna dapat memvisualisasikan struktur database, menjalankan query SQL, mengedit data, dan melakukan tugas administrasi lainnya melalui antarmuka web yang mudah diakses melalui *browser* dengan tampilan diberikan pada Gambar 2.12 (Solichin, 2005).

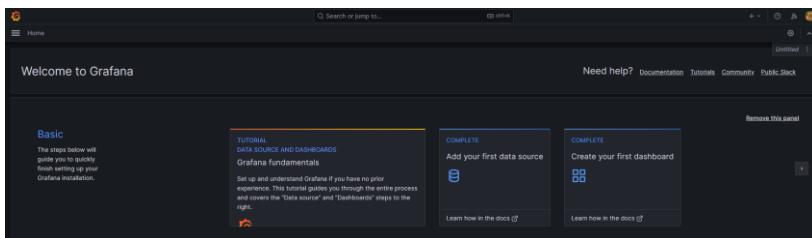


**Gambar 2.12.** Tampilan utama PHPMyadmin

## 2.9. Grafana

Grafana merupakan *platform open source* untuk visualisasi data dan pemantauan, tetapi membutuhkan sebuah sumber data atau data source sebagai masukan dalam menampilkan data yaitu data yang berasal dari berbagai macam database. Grafana dirancang untuk membuat dan berbagi dashboard yang menampilkan data waktu nyata dari berbagai sumber seperti database, server, dan layanan cloud. Mendukung berbagai sumber data, seperti MySql, InfluxDB, Prometheus, Graphite, Elastic search dan mencakup fitur seperti peringatan, anotasi, dan plugin untuk visualisasi data. Pada umumnya, grafana digunakan dalam pemantauan sistem, IoT, dan pemantauan jaringan. Grafana memiliki dashboard antarmuka utama di dalam

grafana yang memungkinkan pengguna untuk memvisualisasikan, menganalisis, dan memahami berbagai data. Gambar 2.13 terdiri dari berbagai panel yang menarik visual dapat menampilkan data dalam bentuk grafik, tabel, peta, dan banyak format visual lainnya. Setiap panel dapat dikonfigurasi secara independen untuk menampilkan data spesifik dari sumber data yang terhubung. Pengguna dapat menyesuaikan tata letak dashboard, menambah atau menghapus panel, dan mengubah ukuran panel untuk menciptakan tampilan yang diinginkan (Noprianto et al., 2023).



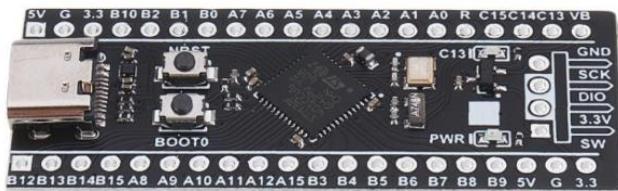
**Gambar 2.13.** Tampilan dashboard Grafana

## 2.10. STM32F401CCU6

Pada dasarnya, penelitian ini membutuhkan sebuah mikrokontroler sebagai pengendalian dan memproses data ataupun sinyal yang masuk. Salah satu mikrokontroler tersebut adalah STM32 yang termasuk ke dalam keluarga arsitektur Cortex-M. Semua mikrokontroler STM32 memiliki inti Cortex-M, ditambah beberapa fitur ST yang khas seperti akselerator ARTTM. Mikrokontroler ini terdiri dari inti prosesor, RAM statis, memori flash, antarmuka debugging, dan berbagai periferal lainnya. Beberapa MCU menyediakan jenis memori tambahan EEPROM, CCM, dan seluruh fitur perangkat yang menargetkan aplikasi berdaya rendah yang terus berkembang. STM32F401CCU6 dapat diprogram menggunakan berbagai bahasa pemrograman, seperti C, C++, dan assembler. STM32F401CCU6 *Blackpill* salah satu keluarga dari STM32 buatan Cortex-M. Gambar 2.14 bentuk fisik dan spesifikasi dari mikrokontroler STM32F401CCU6 dapat dilihat pada tabel 2.2 (Noviello, 2015).

**Tabel 2.2.** Speisifikasi STM32F401CCU6

Keterangan	Nilai
Core	ARM Cortex M4
Maximum Clock Frequency	84 MHz
Tegangan Masuk Minimum	1,7 V
Tegangan Masuk Maksimum	3,6 V
Data Bus Width	32 bit
I/O	36 pin
Data RAM	64 kB



**Gambar 2.14.** STM32F401CCU6

## 2.11. ESP32

ESP32 merupakan salah satu mikronkontroler yang dikembangkan oleh perusahaan Espressif yang populer dan efektif dalam mengembangkan berbagai aplikasi IoT dengan kemampuan fitur Wifi dan Bluetooth didalamnya. ESP32 dilengkapi dengan prosessor dual core Tensilicia Xtensa LX6 yang memberikan daya dalam pemrosesan yang tinggi, sehingga mampu melaksanakan berbagai tugas yang kompleks secara efisien. Kemampuan dalam aplikasi IoT membutuhkan konektivitas Wifi, supaya terhubung dengan jaringan internet yang terintegrasi untuk menyederhanakan koneksi dan komunikasi dengan perangkat atau jaringan lainnya. Hal ini mendukung berbagai protokol Wi-Fi, seperti 802.11 b/g/n yang dapat beroperasi di frekuensi 2,4 GHz dan mendukung kecepatan transfer data hingga 600 Mbps. ESP32 menyediakan banyak pin GPIO yang memfasilitasi koneksi dan pengendalian perangkat dan sensor eksternal. Pin-pin ini mendukung berbagai antarmuka, termasuk SPI, I2C, UART, dan PWM. ESP32 dapat diprogram menggunakan

berbagai kerangka pengembangan dan bahasa pemrograman. Bahasa pemrograman yang paling umum digunakan adalah C++ atau C#, dan dapat diprogram menggunakan Arduino IDE atau PlatformIO. Spesifikasi dari mikrokontroler ESP32 dapat dilihat pada tabel 2.3 (Michon et al., 2020).

**Tabel 2.3.** Spesifikasi ESP32

Keterangan	Nilai
Core	Xtensa LX6 24 MHz
Flash Memory	4 MB
Resolusi	16-bit
Konektor	USB microSD
Wifi	2,4 GHz
I/O	40 Pin
Bluetooth	4,2 BLE



**Gambar 2.15.** ESP32

## 2.12. Sensor

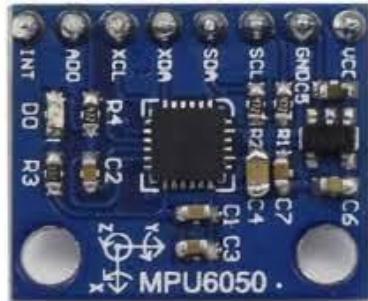
Sensor merupakan sebuah perangkat yang mampu mendeteksi dan mengukur sebuah nilai kuantitas fisis yang dapat diubah ke dalam bentuk sinyal yang dapat diterima oleh prangkat lainnya. Terdapat istilah lainnya dinamakan transducer merupakan perangkat yang mengubah energi dari satu bentuk ke bentuk lainnya.

Tujuan sensor adalah untuk mendeteksi dan mengukur apakah efisiensi dari angka yang diketahui. Transduser, sebaliknya, dimaksudkan untuk mengubah energi dan efisiensinya.

### 2.12.1. MPU6050

Sensor Accelerometer yang digunakan dalam penelitian ini berjenis MPU6050. Accelerometer MPU6050 merupakan suatu sensor yang dapat digunakan untuk mengukur gerakan serta perubahan orientasi dan posisi yang berguna dalam aplikasi seperti deteksi guncangan, pengukuran kecepatan dan arah gerakan, serta percepatan, salah satunya percepatan akselerasi. Accelerometer MPU6050 memiliki rentang pengukuran dari sebesar  $\pm 2g$  hingga  $\pm 16g$  dengan  $1g$  merupakan satu satuan percepatan rata-rata gravitasi bumi yaitu sebesar  $9,8 \text{ m/s}^2$ . Sensor ini dapat mengimplementasikan keluaran dalam 3 aksis (x, y, dan z). Sensor ini dapat digunakan untuk mengukur akselerasi statis gravitasi pada aplikasi *tilt sensing* yang sama baiknya dalam mengukur akselerasi dinamis dari *shock, motion, dan vibration*. Accelerometer MPU6050 memiliki keluaran analog yang berupa tegangan. Accelerometer MPU6050 dapat beroperasi pada tegangan  $2V$  hingga  $3,4V$ . MPU6050 memiliki regulator tegangan yang memungkinkan untuk menerima tegangan masukan yang lebih tinggi, seperti  $5V$  dan menurunkannya ke  $3.4V$  yang aman untuk chip itu sendiri Spesifikasi dari sensor MPU6050 dapat dilihat pada tabel 2.4.

Sensor MPU6050 bekerja berdasarkan pergerakan dari MEMS (*Micro Electro Mechanic Systems*) yang memiliki *micro-mechanic* diatas silicon atau pegas. Jika terjadi pergeseran maka mekanik akan bergerak dan mengubah gerakan menjadi kapasitif dan menimbulkan percepatan yang berbanding lurus. Perubahan dari kapasitif dari sensor yang akan di proses sebagai nilai keluaran tegangan analog atau digital. (MPU-6000 and MPU-6050 Product Specification Revision 3.4 MPU-6000/MPU-6050 Product Specification, 2013).



**Gambar 2.16.** MPU6050

**Tabel 2.4.** Spesifikasi MPU6050

Keterangan	Nilai
Rentang Akselerometer	$\pm 2\text{g}$ , $\pm 4\text{g}$ , $\pm 8\text{g}$ , $\pm 16\text{g}$
Rentang Giroskop	$\pm 250^\circ/\text{s}$ , $\pm 500^\circ/\text{s}$ , $\pm 1000^\circ/\text{s}$ , $\pm 2000^\circ/\text{s}$
Resolusi	16-bit
Output Data Rate	0,10 – 3200 Hz
Tegangan	2V – 3,4V
Komunikasi	I2C

*Halaman ini sengaja dikosongkan*

*Halaman ini sengaja dikosongkan*

## BAB III

### METODOLOGI

#### 3.1. Tempat dan Waktu Penelitian

Penelitian “Rancang Bangun Sistem *Internet of Things* Untuk Monitoring Kinerja Motor DC Berbasis Data Vibrasi” dilaksanakan pada bulan Februari hingga Mei 2024 dengan kegiatan bertempatan di Laboratorium Teknologi Sensor, Gedung Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya, Malang, Jawa Timur, Indonesia.

#### 3.2. Alat dan Bahan

Penelitian ini menggunakan berbagai macam alat dan bahan yang dibagi menjadi dua jenis perangkat keras dan perangkat lunak guna berlangsungnya kegiatan penelitian di antara lain :

**Tabel 3.1.** Daftar alat dan bahan perangkat keras

<b>Nama Alat dan Bahan</b>	<b>Keterangan</b>
Motor DC 555 Double Bearing	Objek yang akan diukur
STM32F401CCU6	Mikrokontroler sebagai otak dari seluruh proses yang dibuat
ESP32	Mikrokontroler yang dilengkapi dengan fitur Wi-Fi
MPU6050	Sensor pendekripsi getaran
Laptop	Perangkat untuk melakukan proses koding dari sistem

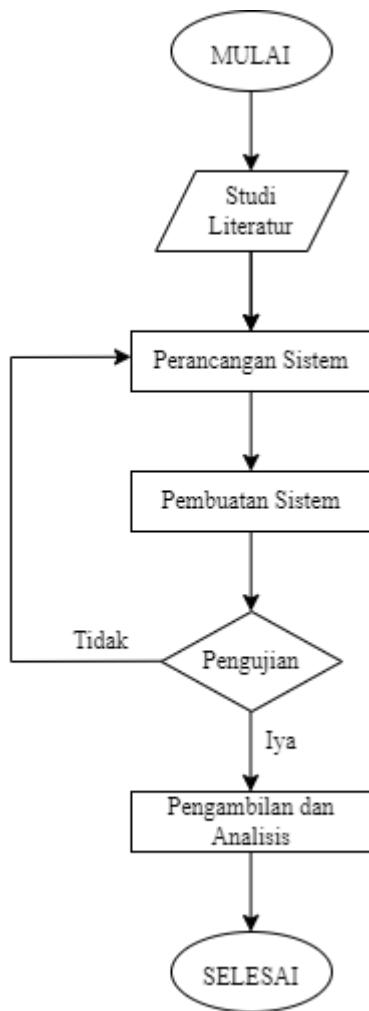
**Tabel 3.2.** Daftar alat dan bahan perangkat lunak

<b>Nama Alat dan Bahan</b>	<b>Keterangan</b>
Arduino IDE	Perangkat lunak yang digunakan untuk melakukan pengkodingan program
STM32 Cube Programmer	Perangkat lunak yang digunakan untuk konfigurasi mikrokontroler STM32

Matlab	Digunakan untuk melakukan pemrosesan perhitungan
Xampp	Perangkat penunjang dalam membuat database mysql
Php Mysql	Digunakan untuk membentuk database
Microsoft Office	Perangkat lunak yang digunakan untuk penunjang dalam pencatatan laporan
NodeRed	Perangkat lunak yang menghubungkan perangkat keras dengan jaringan internet
Grafana Dashboard	Sebuah <i>platform</i> penampil data

### 3.3. Prosedur Penelitian

Proses penelitian dilakukan melalui beberapa tahap yang dilakukan secara terencana, sehingga penelitian dapat dilakukan secara baik dan benar serta mendapatkan hasil yang baik pula. Pada Gambar 3.1 menunjukkan perancangan penelitian secara keseluruhan yang diharapkan agar dapat dilakukan secara terstruktur dengan baik dan benar. Hal ini dilakukan untuk mencari permasalahan yang dapat untuk mencari solusi yang terbaik.



**Gambar 3.1.** Diagram alir

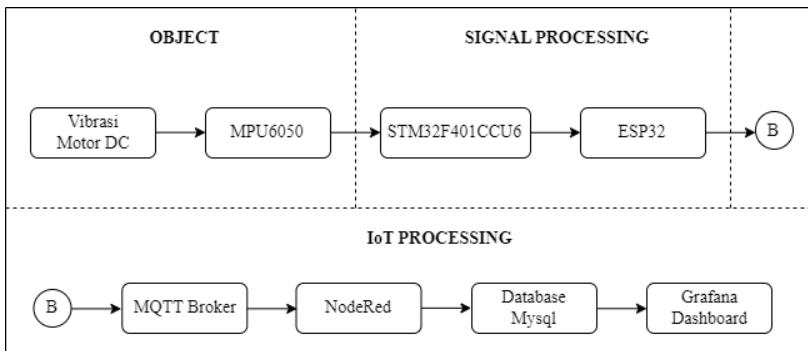
### 3.3.1. Studi Literatur

Pada tahap awal dilakukan dengan pengamatan terhadap masalah yang terjadi pada permasalahan. Untuk mendukung permasalahan tersebut, perlu dilakukan studi literatur dimulai dari mempelajari dan mencari pustaka-pustaka yang berkaitan dengan vibrasi dan penyebab-penyebab kerusakannya, motor dc,

STM32F401CCU6, ESP32, dan data vibrasi dari sensor accelerometer MPU6050. Alur studi literatur dapat dilihat pada Gambar 3.1 diperoleh dari berbagai sumber, diantaranya catatan maupun buku dan beberapa sumber lain seperti buku, jurnal ilmiah, dan beberapa penelitian terdahulu.

### 3.3.2. Perancangan Sistem

Gambar 3.2 merupakan salah satu tahapan selanjutnya yang meliputi desain system penelitian. Perancangan system untuk memenuhi kebutuhan perancangan untuk memberikan gambaran yang jelas dan rancang bangun dari beberapa komponen yang terpisah ke dalam satu kesatuan yang lengkap. Dalam tahapan ini dibagi menjadi perancangan perangkat keras dan perangkat lunak.



**Gambar 3.2.** Diagram blok sistem

#### 1. Object

Objek mencakup pengambilan data getaran dari Motor Dc. Data berasal dari sensor yang telah terkumpul dan tersimpan ke dalam data excel yang merupakan data getaran atau vibrasi yang dihasilkan oleh motor saat beroperasi. Data ini sangat penting untuk monitoring kondisi motor DC.

#### 2. Signal Processing

Data yang diperoleh dari motor Dc kemudian dikirim ke mikrokontroler STM32F401CCU6. Mikrokontroler ini bertugas untuk memproses data mentah vibrasi yang konversi data ke format yang lebih berguna untuk analisis lebih lanjut. Proses ini mengubah data vibrasi menjadi sinyal digital yang dapat diinterpretasikan oleh sistem. Data yang sudah diproses

oleh STM32F401CCU6 kemudian dikirimkan ke ESP32 melalui komunikasi serial yang merupakan modul Wi-Fi. ESP32 berperan sebagai gateway yang mengirim data tersebut ke dalam jaringan internet atau jaringan lokal.

### 3. IoT Processing

Data dikirim melalui protokol MQTT yang merupakan standar untuk komunikasi *IoT*. Protokol ini dirasa sangat efisien karena meminimalisir bandwidth dan cocok untuk komunikasi antar perangkat *IoT*. MQTT Broker menerima data ini dan bertindak sebagai server yang mengelola pesan-pesan yang dikirim oleh perangkat *IoT*. Broker bertugas untuk memastikan data tersebut dapat diteruskan ke NodeRed.

NodeRed berfungsi untuk menghubungkan perangkat keras dengan jaringan internet berbasis pemrograman flow. NodeRed digunakan untuk mengatur aliran data dari MQTT Broker yang kemudian mengirimkan, serta menyimpan ke dalam database Mysql.

Data yang telah diproses kemudian disimpan dalam database Mysql. Database ini mengatur penyimpanan data dan memudahkan pengambilan data untuk analisis lebih lanjut atau visualisasi.

Data yang disimpan dapat divisualisasikan menggunakan Grafana Dashboard yang merupakan platform open source untuk monitoring dan analisis. Grafana memungkinkan untuk dapat membuat dashboard yang dapat menampilkan data dalam berbagai format visual seperti grafik, tabel, atau visualisasi lainnya untuk monitoring secara real time.

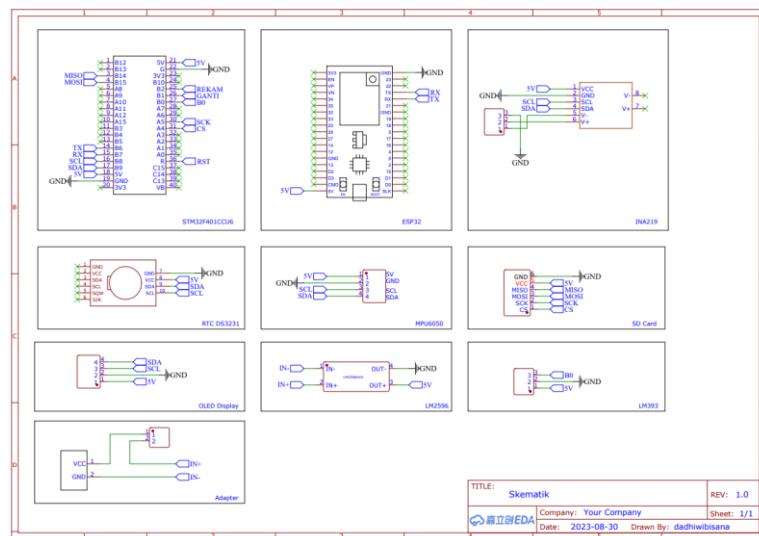
#### 3.3.3. Perancangan Sistem Perangkat Keras

Pada system terdapat data vibrasi motor DC yang merupakan objek yang akan dibandingkan kinerjanya yang disimpan dalam bentuk excel. Data vibrasi motor Dc akan berjalan sesuai dengan jenis baru dan rusak, sehingga analisa vibrasi getaran dapat dilakukan dan dihubungkan dengan perangkat keras.

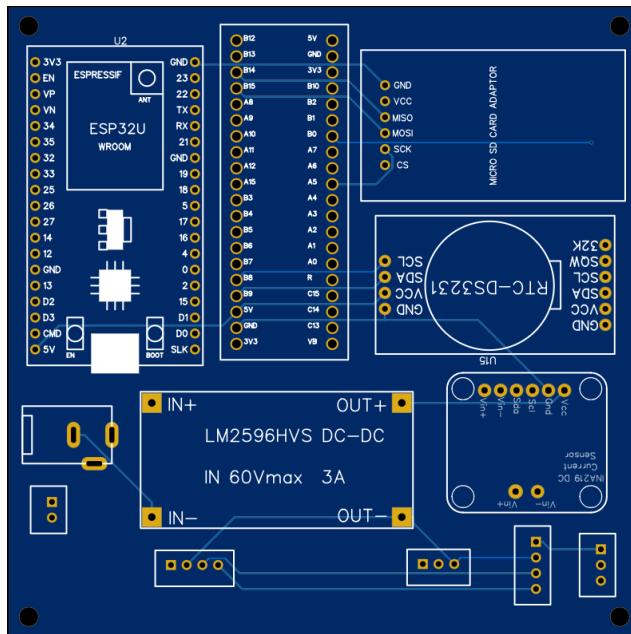
Perancangan *signal processing* merupakan pemrosesan sinyal-sinyal masukan yang akan diproses dan disimpan berupa data yang dapat diterima oleh pengamat dan diteruskan proses selanjutnya. *Signal processing* terdiri dari berbagai macam perangkat keras, meliputi, data vibrasi sensor accelerometer (MPU6050),

mikrokontroller (STM32F401CCU6 dan ESP32). Perancangan meliputi pemasangan masing-masing komponen dengan kabel yang dimana mikrokontroler STM32F401CCU6 sebagai pusat penghubungan kabel. Perancangan perangkat lunak dilakukan dengan menggunakan *software* Visual Code Studio untuk melakukan kode pemrograman python, Arduino IDE untuk melakukan kode pemrograman dengan basis Bahasa C dan STM32CubeProgrammer untuk verifikasi koneksi mikrokontroler.

Sinyal-sinyal yang sudah diproses berbagai perangkat keras dan pengolahan *signal processing* akan dilanjutkan ke dalam system *internet of things processing* yang bertujuan untuk mengirimkan sinyal-sinyal atau data ke jaringan Wi-Fi. STM32F401CCU6 tidak bisa langsung mengirim data menuju internet sehingga data ini harus dikirimkan terlebih dahulu ke ESP32 menggunakan “SoftwareSerial” data dikirimkan menuju platform internet ke ESP32 dengan komunikasi UART (*Universal Asynchronous Receiver-Transmitter*). Pada STM32F401CCU6, data akan diproses dan kemudian dilanjutkan ke ESP32 untuk dihubungkannya perangkat dengan *Wi-Fi*.



Gambar 3.3. Skematik rangkaian alat

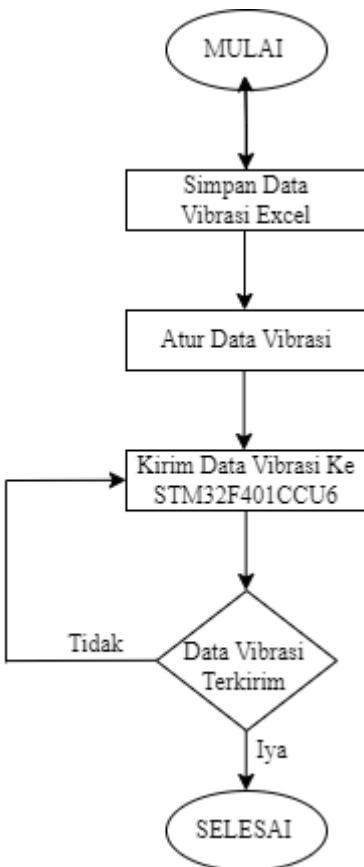


**Gambar 3.4.** Bentuk papan sirkuit rangkaian

### **3.3.4. Perancangan Sistem Perangkat Lunak**

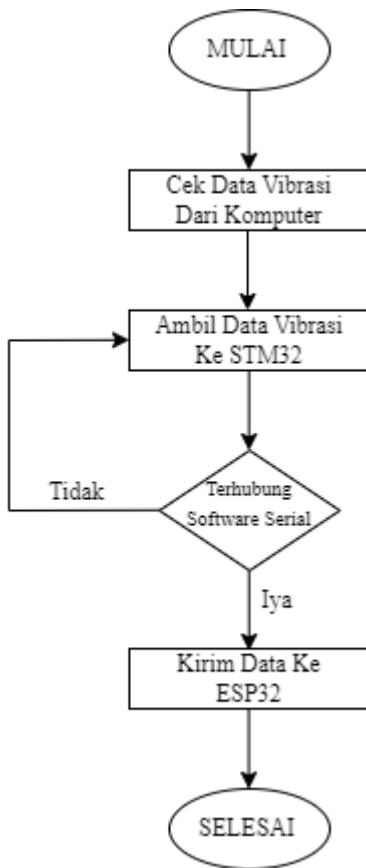
Perancangan program dilakukan dengan menggunakan beberapa *software*, terdiri dari Visual Studio Code, Arduino IDE, Xampp, NodeRed, dan Grafana. *Software* Visual Studio Code digunakan untuk melakukan pemrograman dari data vibrasi berupa excel yang dikirimkan ke mikrokontroler. Arduino IDE digunakan untuk *upload* program pada STM32F401CCU6 dan ESP32. Matlab digunakan untuk melakukan proses perhitungan data vibrasi. NodeRed digunakan untuk penghubung antara alat dengan jaringan internet melalui protocol MQTT. Xampp digunakan sebagai database dari data untuk dapat ditampilkan secara *online* melalui *software* Grafana. Data vibrasi yang telah dilakukan proses perhitungan dengan Matlab akan dikirimkan ke mikrokontroler STM32F401CCU6 dengan menjalankan ekstensi program *python* dengan *command prompt* untuk dilihat hasil dari kinerja motor DC. Data tersebut akan diteruskan dengan komunikasi serial ke ESP32 untuk terhubung dengan jaringan internet. ESP32 akan terhubung dengan *Wi-Fi*.

dengan program yang sudah disesuaikan. Pengkoneksian perangkat-perangkat ke Wi-Fi akan menggunakan melalui broker MQTT dengan protocol komunikasi MQTT (*Message Queing Telemetry Transport*) yang menentukan bagaimana data di transfer antara perangkat keras dengan sistem jaringan internet untuk dapat berinteraksi secara padu dalam lingkungan yang terkoneksi dalam jaringan pribadi atau perusahaan. Setelah perangkat keras terhubung dengan protocol MQTT, data akan dihubungkan dengan NodeRed. NodeRed berfungsi untuk menghubungkan perangkat keras dengan jaringan internet berbasis pemrograman flow. NodeRed digunakan untuk mengatur aliran data dari MQTT Broker yang kemudian mengirimkan, serta menyimpan ke dalam database MySql. Di dalam NodeRed akan menggunakan node MQTT untuk berlangganan topik di mana data dikirimkan dan dapat memproses atau mentransformasi data sesuai kebutuhan. Kemudian, data akan disimpan ke dalam database MySql menggunakan node khusus untuk MySql di NodeRed untuk terhubung dengan database Mysql dan mengirimkan data. Kemudian data yang terimpan di dalam databse Mysql akan ditampilkan ke platform web dashboard grafana yang merupakan platform analisis dan visualisasi yang digunakan untuk membuat dashboard yang interaktif dan informatif. Grafana dapat melakukan konfigurasi sumber data dengan database Mysql. Setelah terhubung dengan Mysql, dapat membuat berbagai jenis grafik dan panel untuk menampilkan data yang telah disimpan.



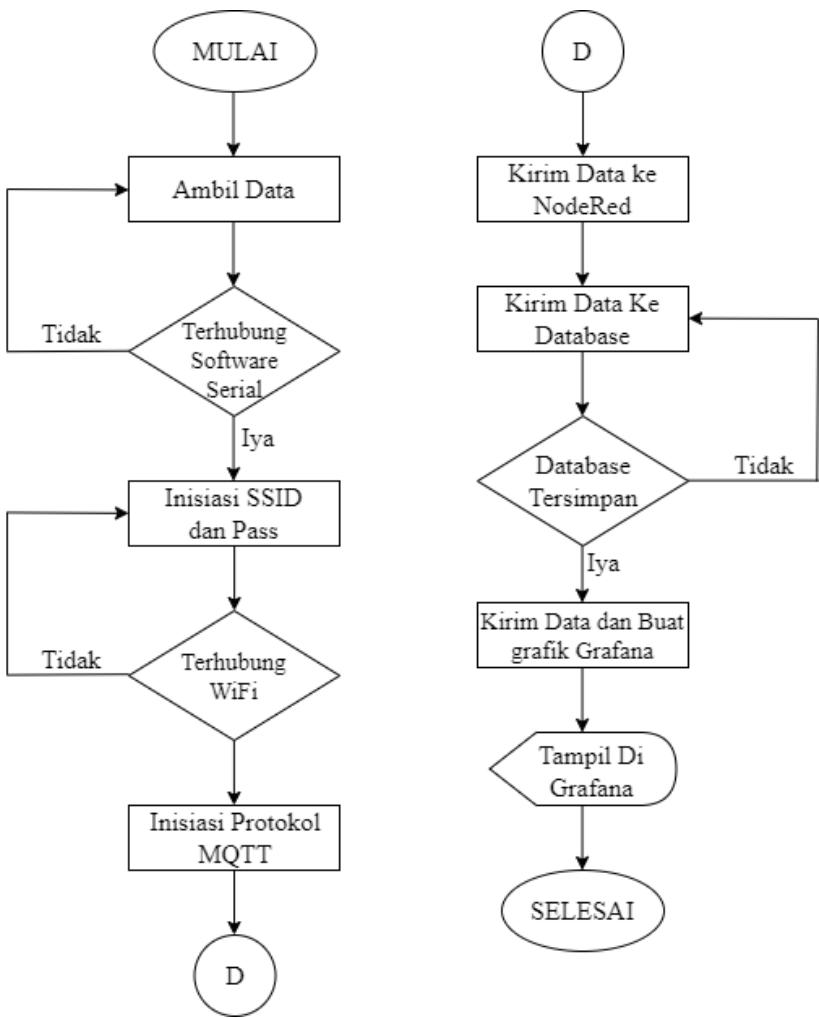
**Gambar 3.5.** Flowchart program ekstensi

Gambar 3.5 menunjukan awal program melakukan penyimpanan data vibrasi dalam file excel, kemudian atur data yang akan dikirimkan pada baris excel apakah sudah sesuai dengan yang diinginkan. Selanjutnya, program ekstensi akan mengirimkan data vibrasi dari file excel kondisi motor Dc. Setelah itu, STM32F401CCU6 akan menerima data vibrasi.



**Gambar 3.6.** *Flowchart* program STM32F401CCU6

Gambar 3.6 melakukan pengecekan data terkirim dari komputer yang diinisialisasikan terlebih dahulu, kemudian mengambil data vibrasi motor DC apakah sudah sesuai dengan yang dikirimkan. Selanjutnya, STM32F401CCU6 akan mengambil data vibrasi dari kondisi motor DC. Setelah itu, STM32F401CCU6 akan terhubung dengan SoftwareSerial, jika tidak terhubung STM32F401CCU6 akan terus mengambil data dari sensor, jika terhubung data akan diteruskan ke ESP32.



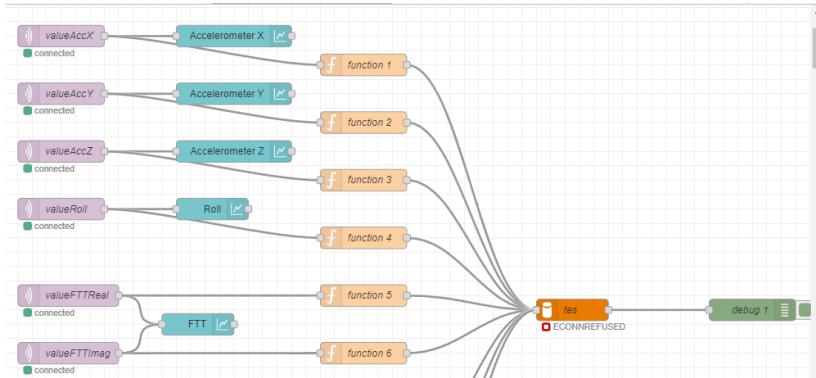
**Gambar 3.7.** Flowchart program ESP32

Gambar 3.7 merupakan awal program ESP32 akan mengambil data yang dikirimkan STM232F401CCU6 dengan softwareserial, jika tidak terhubung terdapat kesalahan input program atau pemasangan *wiring* kedua perangkat. Kemudian, SSID dan *password* Wi-Fi diinisialisasi terlebih dahulu agar dapat terhubung dengan jaringan internet, jika tidak terhubung terdapat kesalahan dalam memasukkan

SSID dan *password*. Setelah terhubung dengan Wi-Fi, maka akan dilakukan insiasi protocol komunikasi MQTT untuk mengirimkan data ke jaringan internet. Protokol MQTT akan mengirimkan data ke NodeRed untuk menghubungkan perangkat dengan antarmuka internet lainnya. Data akan dirikim untuk disimpan secara *online* pada database, jika pada database tidak terdapat data yang masuk, maka akan data yang dikirim mengalami kesalahan alamat database dari NodeRed. Jika data berhasil disimpan di database, data akan dipanggil atau diambil oleh Grafana untuk dilakukan penampilan data sesuai dengan kebutuhan yang diinginkan.

### 3.3.4.1. Konfigurasi NodeRed

Penggunaan NodeRed dilakukan dengan sederhana dengan merangkai sebuah node sehingga terbentuk sebuah alur atau aliran tanpa menggunakan sebuah kode program. NodeRed memanfaatkan publish ataupun subscribe data dari protokol MQTT sudah tersedia dan yang perlu dilakukan adalah melakukan konfigurasi alamat server broker dan topik. Ketika akan mengirimkan atau menyimpan data pada database, dibutuhkan beberapa konfigurasi seperti alamat server database, nama database, serta table database. Berikut merupakan rancangan flow untuk menyimpan data ke database.



Gambar 3.8. Tampilan flow NodeRed

Untuk dapat mengkonfigurasi NodeRed dapat dilakukan dengan cara sebagai berikut:

- Buka *browser* pada computer, dan membuka alamat

<https://nodered.org/> untuk mendownload NodeRed.



**Gambar 3.9.** Halaman download NodeRed

- Kemudian install NodeRed, kemudian buka *command prompt* dan tuliskan node-red.

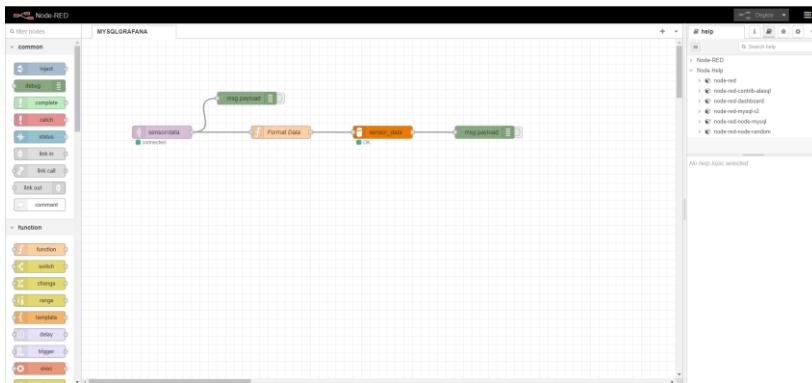
```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dadhi\node-red
21 Nov 02:50:03 - [info]

Welcome to Node-RED
=====
21 Nov 02:50:03 - [info] Node-RED version: v3.0.2
21 Nov 02:50:03 - [info] Node.js version: v20.1.0
21 Nov 02:50:03 - [info] Windows_NT 10.0.22621 x64 LE
21 Nov 02:50:05 - [info] Loading palette nodes
21 Nov 02:50:06 - [info] Dashboard version 3.4.0 started at /ui
21 Nov 02:50:06 - [info] Settings file : C:\Users\dadhi\.node-red\settings.js
21 Nov 02:50:06 - [info] Context store : 'default' [module=memory]
21 Nov 02:50:06 - [info] User directory : \Users\dadhi\.node-red
21 Nov 02:50:06 - [warn] Projects disabled : editorTheme.projects.enabled=false
21 Nov 02:50:06 - [info] Flows file : \Users\dadhi\.node-red\flows.json
21 Nov 02:50:06 - [info] Server now running at http://127.0.0.1:1880/
21 Nov 02:50:06 - [warn]
```

**Gambar 3.10.** Tampilan command NodeRed

- Selanjutnya buka alamat server localhost yang tertera pada command NodeRed pada web, sehingga tampilan halaman NodeRed seperti berikut:



**Gambar 3.11.** Tampilan halaman NodeRed

### 3.3.4.2. Pembuatan Database PhpMysql

Penggunaan database PHP menyediakan fungsi untuk terhubung ke server Mysql membuat queries, mengambil data, dan memanipulasi database. Skrip PHP dapat mengirim perintah SQL ke MySQL untuk melakukan operasi seperti INSERT, SELECT, UPDATE, dan DELETE. Database dan tabel dapat dibuat secara manual melalui Mysql *Command Line* atau menggunakan phpMyAdmin. PHP dan MySQL merupakan kombinasi yang populer karena mudah digunakan, fleksibel, dan cukup kuat untuk sebagian besar kebutuhan pengembangan web. Untuk dapat membuat database phpMysql dapat dilakukan dengan cara sebagai berikut:

- Buka *browser* pada computer, dan membuka alamat <https://www.apachefriends.org/download.html> untuk mendownload web database php.

Apache Friends   Download   Hosting   Community   About   Search   Search   EN

## Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

 XAMPP for Windows 8.0.28, 8.1.17 & 8.2.4

Version	Checksum	Size		
8.0.28 / PHP 8.0.28	What's included?	md5 sha1	<a href="#">Download (64 bit)</a>	144 Mb
8.1.17 / PHP 8.1.17	What's included?	md5 sha1	<a href="#">Download (64 bit)</a>	146 Mb
8.2.4 / PHP 8.2.4	What's included?	md5 sha1	<a href="#">Download (64 bit)</a>	149 Mb

Requirements: [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

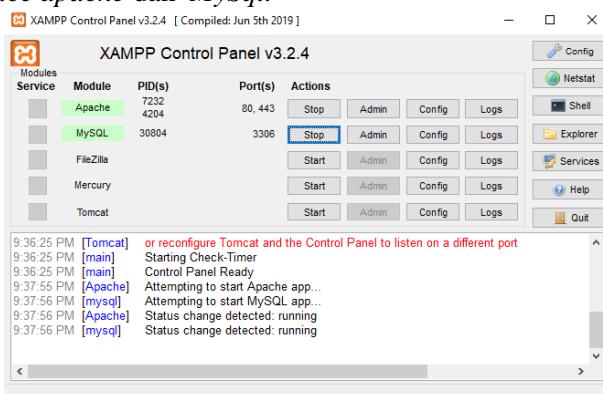
### Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs

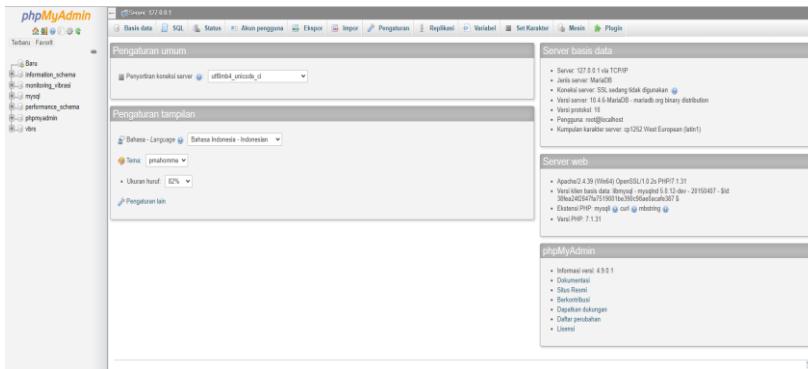
### Gambar 3.12. Halaman download database

- Kemudian install hasil *download* dan jalankan file web database php dengan nama *Xampp Control Panel*, Setelah terbuka *start service apache* dan *MySql*.



### Gambar 3.13. File control panel web php

- Kemudian buka kembali *browser* pada computer dan jalankan alamat web <http://localhost/phpmyadmin/index.php?lang=id> untuk membuka web php.



**Gambar 3.14.** Tampilan web PhpMysql

- Terakhir klik pada pilihan beranda baru untuk membuat basis data dan sesuaikan dengan kebutuhan dalam pembuatan basis database phpmysql.

Basis data	Penyortiran	Tindakan
information_schema	utf8_general_ci	<a href="#">Check privileges</a>
monitoring_vibrasi	latin1_swedish_ci	<a href="#">Check privileges</a>
mysql	latin1_swedish_ci	<a href="#">Check privileges</a>
performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
phpmyadmin	utf8_bin	<a href="#">Check privileges</a>
vbris	utf8mb4_general_ci	<a href="#">Check privileges</a>
Jumlah: 6	latin1_swedish_ci	

Buat basis data [?](#)

Nama basis data: latin1\_swedish\_ci [Buat](#)

Pilih Semua [Dengan pilihan:](#) [Hapus](#)

**Gambar 3.15.** Basis database PhpMysql

### 3.3.4.3. Dashboard Grafana

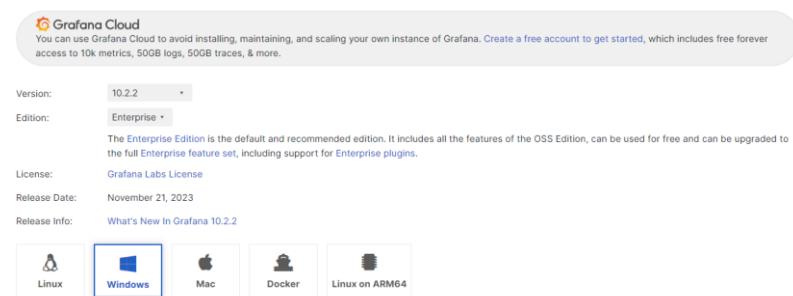
Untuk membuat sebuah dashboard grafana, terdapat langkah-langkah yang perlu dilakukan yaitu menambahkan *data source* pada grafana. *Data source* yang dimaksud adalah database pada web php

dan mysql yang akan diambil datanya kemudian ditampilkan pada sebuah dashboard.

- Langkah awal dalam membuat dashboard Grafana, yaitu mendownload Grafana di Alamat <https://grafana.com/grafana/download?platform=windows>

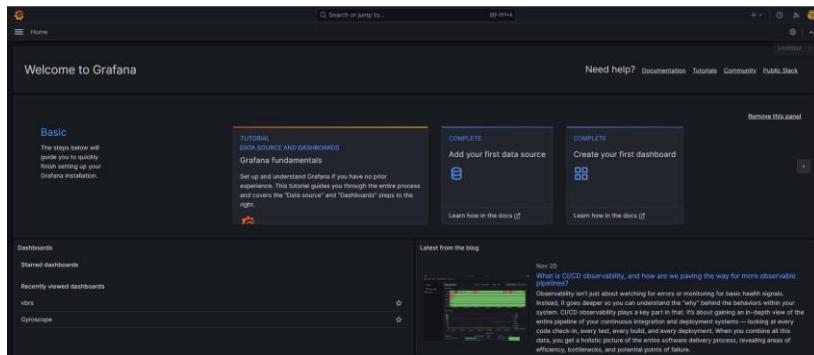
## Download Grafana

Nightly Builds



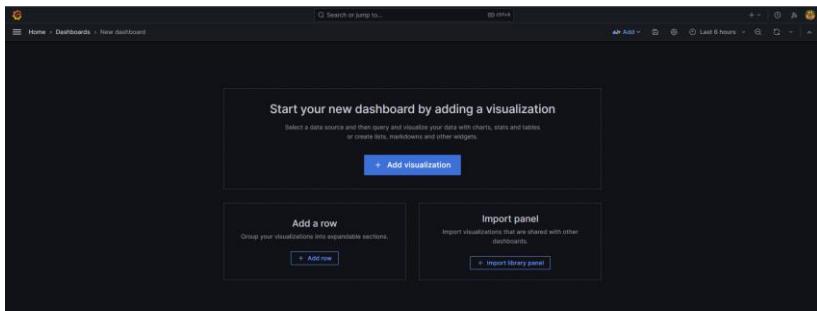
**Gambar 3.16.** Grafana *download*

- Setelah selesai mendownload, install file grafana pada computer dan buka *browser* pada computer dengan alamat <http://localhost:3000/?orgId=1>



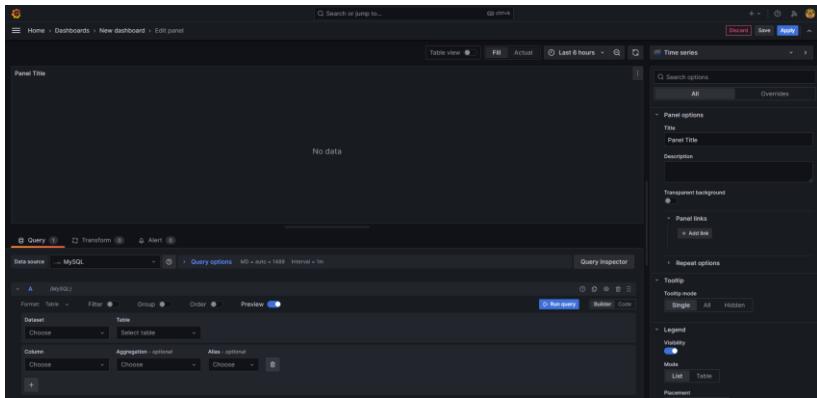
**Gambar 3.17.** Tampilan web grafana

- Buat halaman *dashboard* baru dengan memilih pilihan *dashboard* pada menu beranda, dan klik *add visualization*



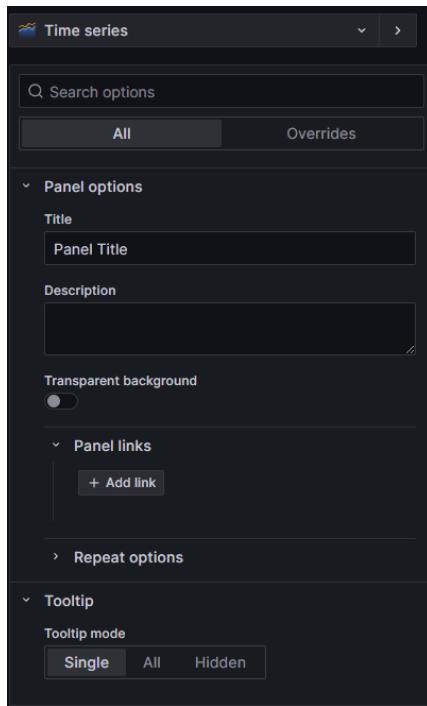
Gambar 3.18. Tampilan *new dashboard*

- Kemudian pilih *data source* Mysql, atur *dataset* sesuai dengan basis database dan tabel database php untuk memilih data apa yang ditampilkan, serta *column* berupa data keseluruhan atau sebagian.



Gambar 3.19. Tampilan pengaturan *dashboard*

- Pada bagian kanan dapat diatur jenis grafik, tabel, variable maksimum dan minimum, dan warna.



**Gambar 3.20.** Tampilan pengaturan panel

### 3.3.5. Pengujian

Setelah seluruh perangkat atau komponen terpasang dengan baik, tahap selanjutnya adalah pengujian. Pengujian dilakukan untuk mengetahui apakah bagian-bagian yang telah dirancang beroperasi sesuai dengan semestinya atau tidak. Bagian yang di uji dalam penelitian adalah kemampuan program dalam memproses data vibrasi untuk diubah menjadi spektrum vibrasi, serta program dalam pengiriman, pengambilan dan pemrosesan oleh ekstensi program dan STM32F401CCU6 sebagai mikrkontroler dan kesuaian program dalam melakukan perintah, serta kemampuan ESP32 dalam kemampuan mengirimkan data ke dalam jaringan internet Wi-Fi atau ke dalam system *Internet of Things*. Ketidaksesuaian yang terjadi pada pengujian terjadi karena berbagai faktor dari perangkat maupun program, sehingga diperlukannya kembali proses pembuatan program.

Data viabiasi merupakan data nilai sumbu X, Y, dan Z yang berasal dari sensor accelerometer MPU6050 yang dipasangkan di motor Dc baru dan motor Dc rusak sebelumnya. Data tersebut akan disimpan dalam excel untuk mempermudah proses perhitungan untuk mendapatkan sinyal spektrum vibrasi. Data sumbu X, Y, dan Z merupakan percepatan yang dihasilkan oleh motor Dc, untuk mendapatkan hasil spektrum vibrasi perlu dilakukannya perhitungan percepatan total dari sumbu tersebut. Percepatan total dari sumbu-sumbu X, Y, dan Z dapat ditentukan menggunakan rumus perhitungan vector, percepatan total dapat dihitung menggunakan rumus:

$$a_{total} = \sqrt{x^2 + y^2 + z^2} \quad (3.1)$$

Dimana :

- $a_{total}$  = Percepatan total
- X = Mendeteksi percepatan mengarah kiri ke kanan
- Y = Mendeteksi percepatan mengarah atas ke bawah
- Z = Mendeteksi percepatan mengarah maju ke depan

Setelah didapatkan nilai dari percepatan total, data vibrasi akan disimpan dalam excel yang kemudian dilakukan perubahan ke spektrum vibrasi dengan *software* Matlab dengan beberapa program perhitungan *Fast Fourier Transform* dengan program sebagai berikut:

```
Mb = xlsread('0g','DATA','B2:B3003');
Fsb = 100;
Nb = length(Mb);
Y_Mb = fft(Mb);
dBMB = 20*log10(abs(Y_Mb));
frequencies_Mb = (0:Nb-1)*Fsb/Nb;
magnitude_Mb = (dBMB)/Nb;
```

Baris pertama akan membaca data dari file Excel dengan nama file "DATA", dari sheet "BARU", dan dari sel B2 sampai B3003 dan kemudian disimpan dalam matriks Mb. Variabel Fsb menentukan frekuensi sampling menggunakan 100 Hz. Variabel Nb menentukan panjang data, yaitu jumlah data yang diinginkan. Baris selanjutnya melakukan FFT pada data Mb akan menghasilkan koefisien spektrum frekuensi kompleks yang disimpan dalam matriks Y\_Mb. Untuk mengubah hasil FFT ke dalam satuan desibel (dB) memungkinkan

untuk memperoleh bentuk spektrum yang mudah dipahami dalam spektrum sinyal yang luas menggunakan rumus  $20*\log_{10}(FFT)$ . Membuat vektor frekuensi yang sesuai dengan hasil FFT pada rentang frekuensi yang dihasilkan berasal dari 0 Hz hingga ( $F_{sb} - \Delta_f$ ), dimana  $\Delta_f$  adalah resolusi frekuensi. Setelah itu, menghitung spektrum dengan membagi hasil konversi ke dB dengan panjang data. Dengan melakukan proses perhitungan FFT, selanjutnya data spektrum akan dikirimkan ke mikrokontroler dan jaringan Wi-Fi untuk dapat memonitoring kinerja motor Dc.

### 3.3.6. Pengambilan Data dan Analisis Data

Sistem monitoring dilakukan untuk mendeteksi kinerja motor Dc melalui data vibrasi yang didapatkan dari data sensor. Pengambilan data dilakukan jika seluruh system perangkat keras maupun perangkat lunak telah berhasil dan tidak ada kesalahan. Analisis data dilakukan untuk menentukan kesesuaian rumusan masalah dan tujuan penelitian dengan hasil penelitian. Data vibrasi, untuk melihat kinerja dari sebuah motor Dc yang bekerja untuk membandingkan motor Dc yang masih layak untuk dipakai dengan yang perlu dilakukannya pemeriksaan.

Pengambilan data dilakukan dengan menyiapkan pengambilan data vibrasi sumbu X,Y, dan Z dengan sensor MPU6050 dan dilakukan pemrosesan dengan mikrokontroler, kemudian dilakukan perhitungan data vibrasi di *software Matlab*. Setelah data spektrum vibrasi sesuai, ekstensi *python* akan dijalankan untuk mengirimkan ke mikrokontroler agar di dikirimkan ke jaringan Wi-Fi untuk dilakukan monitoring kinerja motor Dc dengan mengamati nilai spektrum vibrasi.

Pada tahap analisis, penelitian mencakup perbandingan antara data spektrum vibrasi motor Dc baru dan motor Dc rusak. Nilai tersebut sebelumnya sudah dilakukan pengubahan satuan dan diubah menjadi grafik FFT yang merupakan alat penting dalam pemrosesan sinyal digital, yang mengubah sinyal dari domain waktu menjadi domain frekuensi atau spektrum vibrasi. Nilai dari data spektrum vibrasi tersebut, akan dilihat dari puncak nilai magnitude spektrum serta karakteristik bentuk gelombang spektrum untuk dapat mengetahui perbedaan diantara dua jenis motor Dc, sehingga mengetahui mana jenis motor Dc baru dan motor Dc rusak, serta

melakukan monitoring kinerja motor Dc secara *online* dengan jaringan Wi-Fi.

*Halaman ini sengaja dikosongkan*

*Halaman ini sengaja dikosongkan*

## **BAB IV**

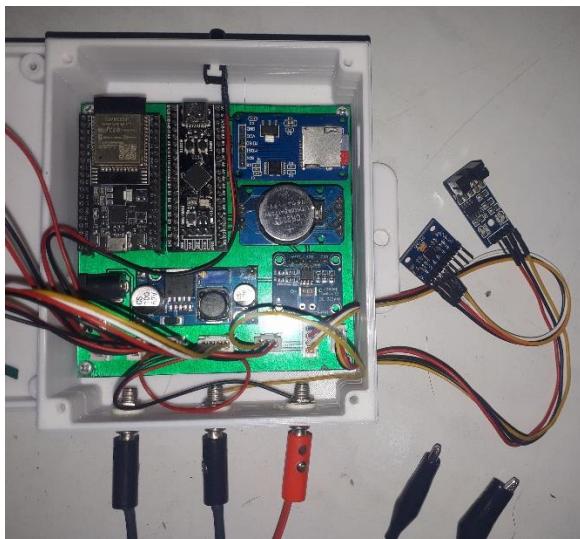
### **HASIL DAN PEMBAHASAN**

#### **4.1. Desain Alat**

Desain alat yang digunakan terdiri dari 2 bagian, perangkat keras dan perangkat lunak. Desain perangkat keras tersusun dari data vibrasi dengan mikrokontroler. Perangkat keras akan disatukan menjadi suatu system dengan menggunakan kabel penghubung. Setelah perangkat keras menjadi satu kesatuan, untuk dapat menjalankan seluruh system akan digunakannya perangkat lunak untuk dapat membaca nilai-nilai data yang akan diproses untuk ditampilkan hasilnya.

##### **4.1.1. Hasil Desain Perangkat Keras**

Desain perangkat keras menggunakan sebuah kbael penghubung untuk menjadikan beberapa bagian alat menjadi sebuah satu system yang saling terhubung. Sensor MPU6050 akan menjadi sumber nilai-nilai data yang diinginkan. Mikrokontroler STM32F401CCU6 akan menjadi pusat dalam pemrosesan data. Sedangkan, mikrokontroler ESP32 akan bertugas untuk menghubungkan system ke dalam jaringan nirkabel Wi-Fi agar dapat saling berkomunikasi di jaringan *Internet of Things*.



**Gambar 4.1.** Rangkain PCB



**Gambar 4.2.** Casing rangkaian

Keseluruhan rangkaian PCB disatukan kedalam casing berbahan PLA untuk melindungi perangkat PCB dan gangguan dari luar. Perangkat keras terhubung dengan sumber tegangan melalui port jack DC adaptor dan kemudian ditampilkan melalui Oled display.

#### 4.1.2. Hasil Desain Program

Hasil desain program dilakukan dengan *software* Visual Studio Code dan Arduino IDE dalam Bahasa *python* untuk program ekstensi dan Bahasa C untuk STM32F401CCU6 dan ESP32. Pertama merupakan program skrip *python* untuk membaca data vibrasi dari *file* Excel berupa data Frekuensi, X, Y, Z dan data FFT, kemudian mengirimkan data vibrasi tersebut melalui komunikasi serial ke dalam STM32F401CCU6 yang dibuat untuk dapat memproses dan mengolah sinyal atau data vibrasi motor Dc yang sebelumnya berasal dari sensor yang datanya telah di ambil. Kemudian, data akan diterima oleh ESP32 yg berasal dari STM32F401CCU6 untuk dapa memproses data untuk dikirimkan ke dalam system *Internet of Things* untuk dilakukan monitoring kinerja motor Dc.

Pada awal program STM32F401CCU6 diawali dengan menyisipkan Pustaka yang dibutuhkan dalam kode program untuk berkomunikasi antar perangkat. Serta, Menginisialisasi objek SoftwareSerial untuk komunikasi serial dengan pin yang sudah ditetapkan.

```
#include <SoftwareSerial.h>
SoftwareSerial esp32(PB6, PB7);
double X,Y,Z;
int newdatacom;
```

Selanjutnya program akan menginisialisasi berupa void setup. Void setup digunakan untuk menginisialisasi komunikasi serial melalui port serial STM32F401CCU6. Fungsi ini mengatur kecepatan data untuk komunikasi serial. Kecepatan data ini adalah jumlah bit per detik yang akan ditransmisikan melalui port serial.

```
void setup() {
Serial.begin(115200);
Esp32.begin(115200);
```

Selanjutnya setelah program void setup sudah berjalan dengan semestinya, akan dilanjutkan dengan program void loop. Program ini dirancang untuk membaca dan memproses data untuk mengirimkan data melalui port serial ke ESP32 melalui komunikasi serial SoftwareSerial.

```
void loop() {  
    int16_t ax, ay, az;  
    mpu.getAcceleration(&ax, &ay, &az);  
    Wire.beginTransmission(0x68);  
    Wire.write(0x3B);  
    Wire.endTransmission(false);  
    Wire.requestFrom(0x68, 14, true);  
    AcX = Wire.read() << 8 | Wire.read();  
    AcY = Wire.read() << 8 | Wire.read();  
    AcZ = Wire.read() << 8 | Wire.read();  
    float accX = ((float)AcX) / 2048.0;  
    float accY = ((float)AcY) / 2048.0;  
    float accZ = ((float)AcZ) / 2048.0;  
}
```

Pada awal program ESP32 diawali dengan menyisipkan kode program *library* untuk menghubungkan ESP32 dengan internet melalui jaringan Wi-Fi, menghubung ESP32 dengan broker MQTT, komunikasi SoftwareSerial.

```
#include <WiFi.h>  
#include <PubSubClient.h>  
SoftwareSerial stm32(3, 1);
```

Selanjutnya program akan mendeklarasikan sebuah array karakter untuk nama SSID jaringan Wi-Fi dan password jaringan Wi-Fi. Setelah terkoneksi dengan jaringan Wi-Fi terdapat pengelolaan koneksi jaringan. Kemudian mendeklarasikan protocol MQTT untuk mengirim dan menerima pesan melalui protocol komunikasi MQTT.

Terdapat alamat server pada protocol MQTT dan port untuk membuka alamat web.

```
const char* ssid = ".....";
const char* password = ".....";
const char* mqtt_server = "test.mosquitto.org";
const int mqtt_port = 1883;
const char* mqtt_topic = "sensor/data";
```

Selanjutnya program akan menginisialisasi berupa void setup. Void setup digunakan untuk inisialisasi fungsi yang digunakan untuk menginisialisasi komunikasi serial melalui port serial ESP32. Fungsi ini mengatur kecepatan data untuk komunikasi serial. Kecepatan data ini adalah jumlah bit per detik yang akan ditransmisikan melalui port serial. ESP32 melakukan perintah menghubungkan ke jaringan Wi-Fi dan broker MQTT.

```
void setup() {
    Serial.begin(115200);
    stm32.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}
```

Selanjutnya setelah program void setup sudah berjalan dengan semestinya, akan dilanjutkan dengan program void loop. Program ini dirancang untuk membaca dan memproses data yang diterima melalui port serial. Program digunakan untuk mengumpulkan data dari STM32F401CCU6 melalui komunikasi serial SoftwareSerial. Data akan dibaca dan dikumpulkan menjadi string, kemudian diproses dan dipisahkan menjadi elemen-elemen yang lebih kecil untuk digunakan lebih lanjut dalam program. Program void loop digunakan untuk melakukan pekerjaan secara terus-menerus untuk mendapatkan data tanpa berhenti dengan mengirimkan pesan client data vibrasi ke MQTT broker.

```

void loop() {
    while (stm32.available() > 0) {
        delay(10);
        char c = stm32.read();
        data += c;
    }
    if (data.length() > 0) {
        data.toCharArray(input, sizeof(input));
        splitString(input, delimiter, output,
&outputSize);
        for (int i = 0; i < outputSize; i++) {
            switch (i) {
                case 0:
                    value_accX = output[0];
                    break;
                case 1:
                    value_accY = output[1];
                    break;
                case 2:
                    value_accZ = output[2];
                    break;
            }
            Serial.println(data);
            if (!client.connected()) {
                reconnect();
            }
            client.loop();
            if(newdatacom ==1){
                client.publish(mqtt_vx, String(x).c_str());
                client.publish(mqtt_vy, String(y).c_str());
                client.publish(mqtt_vz, String(z).c_str());
                newdatacom =0;
            }
        }
    }
}

```

Selanjutnya program berjalan pada perangkat yang terhubung ke broker MQTT dan melakukan beberapa tugas terkait koneksi dan pengiriman data. Pengecekan kondisi apakah klien MQTT (*client*) terhubung ke broker MQTT atau tidak. Jika tidak terhubung, fungsi

reconnect() akan dipanggil. Hal ini bertujuan untuk memastikan koneksi tetap aktif dan stabil. Serta, mengirimkan pesan ke broker MQTT dan menandakan bahwa data baru telah dikirim.

```
if (!client.connected()) {  
    reconnect();  
}  
client.loop();  
if(newdatacom ==1){  
    Frek.toCharArray(sendValue, sizeof(sendValue));  
    client.publish(mqtt_vfrek,  
    String(Frek).c_str(),sendValue);  
    memset(sendValue, 0, sizeof(sendValue));  
    newdatacom =0;  
}
```

Fungsi setup\_wifi() bertujuan untuk membuat perangkat terhubung ke jaringan WiFi yang ditentukan pada saat booting, dan akan mencoba kembali koneksi jika gagal dalam waktu yang ditentukan dengan penundaan 10 milidetik yang dimaksudkan untuk memberi waktu bagi sistem untuk inisialisasi sebelum memulai koneksi Wi-Fi. Kemudian, mengatur mode Wi-Fi perangkat sebagai mode station dan memulai proses koneksi Wi-Fi ke dalam jaringan internet dengan nama SSID dan kata sandi yang telah ditentukan sebelumnya. Jika koneksi tidak berhasil perangkat akan mengulang kembali menggunakan perintah ESP.restart()

```
void setup_wifi() {  
    delay(10);  
    Serial.println();  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
    WiFi.mode(WIFI_STA);  
    WiFi.begin(ssid, password);  
    unsigned long startAttemptTime = millis();
```

```
while (WiFi.status() != WL_CONNECTED && millis() -  
startAttemptTime < 3000) {  
    delay(500);  
    Serial.print(".");  
}  
if (WiFi.status() != WL_CONNECTED) {  
    Serial.println("Connection failed. Restarting...");  
    ESP.restart();  
}
```

Salah satu bagian yang ada di dalam void loop adalah reconnect yang bertujuan untuk memastikan bahwa papan mikrokontroler tetap terhubung ke broker MQTT. Terdapat fungsi *loop while* yang berjalan selama MQTT tidak terhubung ke broker MQTT. Jika kehilangan koneksi, kode di dalam loop akan berusaha menghubungkan kembali. Fungsi *reconnect()* ini sangat penting dalam program, karena memastikan bahwa perangkat tetap terhubung ke broker MQTT dan dapat terus menerima atau mengirim pesan, bahkan jika terjadi masalah sementara pada jaringan atau broker.

```
void reconnect() {  
    while (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
        String clientId = "ESP8266Client-";  
        clientId += String(random(0xffff), HEX);  
  
        if (client.connect(clientId.c_str())) {  
            Serial.println("connected");  
        } else {  
            Serial.print("failed, rc=");  
            Serial.print(client.state());  
            Serial.println(" try again in 5 seconds");  
            delay(5000);  
        }  
    }  
}
```

Program skrip *python* ini menggunakan modul ‘pandas’ untuk manipulasi data Excel, ‘serial’ untuk komunikasi serial, ‘time’ untuk waktu antar pengiriman data, dan ‘sys’ untuk mengakses argumen baris perintah yang diberikan kepada skrip. Perintah ‘sys’ memeriksa apakah jumlah argumen baris perintah yang diberikan tepat 5 argumen berupa nama skrip, path file Excel, baris awal, baris akhir, dan detik pengiriman data.

```
import pandas as pd
import serial
import time
import sys
if len(sys.argv) != 5:
    print("Usage: python data.py <path_to_excel_file>
<start_row> <end_row> <delay_seconds>")
    sys.exit(1)
```

Program akan membuka dan membaca *file* Excel yang diberikan melalui argumen baris perintah. Jika *file* tidak ditemukan, akan muncul pesan kesalahan, dan program akan keluar. Program mengatur koneksi serial dengan perangkat yang terhubung melalui *port* ‘COM15’ dengan baud rate 115200. Program kemudian membaca *file* Excel dan mengulangi melalui baris-barisnya. Jika file tidak ditemukan, maka akan muncul pesan kesalahan dicetak dan skrip keluar dengan status keluar. Untuk setiap baris yang ditentukan oleh argumen baris awal dan akhir, program memformat sebuah string dengan data dari kolom tertentu dalam baris tersebut. Format string mengikuti format “Data {row[‘FFT’]} \r\n” yang merupakan data dari file Excel sesuai dengan row yang tersimpan, kemudian dikirimkan melalui koneksi serial ke perangkat terhubung. Setelah pengiriman setiap baris data, program akan menunggu selama periode yang ditentukan oleh argumen detik pengiriman. Setelah seluruh data Excel terkirim, program akan menutup koneksi.

```
excel_file = sys.argv[1]
start_row = int(sys.argv[2])
end_row = int(sys.argv[3])
delay_seconds = float(sys.argv[4])
```

```

try:
    pd.read_excel(excel_file)
except FileNotFoundError:
    print(f"Error: The file '{excel_file}' does not exist.")
    sys.exit(1)
ser = serial.Serial('COM15', 115200)
df = pd.read_excel(excel_file)
for index, row in df.iterrows():
    if start_row <= index + 1 <= end_row:
        row_number = index + 1
        data_to_send = f"DATAS,{row['Z']}]\r\n"
        print(f"Sending {row_number}:
{data_to_send}")
        ser.write(data_to_send.encode())
        time.sleep(delay_seconds)
ser.close()

```

Setelah program ekstensi berjalan pada computer, selanjutnya merupakan program ekstensi mikrokontroler untuk menerima data vibrasi yang telah tersimpan dalam bentuk excel. Program penerimaan data dari ekstensi computer ke dalam mikrokontroler menggunakan file header dengan mendefinisikab serial untuk berkomunikasi dengan computer.

Fungsi void serialEventRun() dalam kode program menggunakan fungsi serial untuk berkomunikasi dengan komputer ke mikrokontroler. Fungsi ini dirancang untuk memeriksa apakah ada data yang tersedia di port serial dan mengambil data yang tersedia di port serial dan menyimpannya dalam array dataSerialcntr. Ketika data tersedia di port serial, fungsi ini akan dipanggil secara otomatis.

```

void serialEventRun() {
    if (Sercntr.available()) {
        dataSerialcntr[count_serial_cntr] = SercntrRX();
        count_serial_cntr++;
        if (dataSerialcntr[count_serial_cntr - 1] ==
'\r')
    {

```

```

        slais_r_cntr = count_serial_cntr - 1;
    }
    if (dataSerialcntr[count_serial_cntr - 1] ==
'\n')
    {
        slais_n_cntr = count_serial_cntr - 1;
    }
    if (slais_n_cntr - slais_r_cntr == 1)
    {
        olah_data_cntr();
    }
}

```

Fungsi void olah\_data\_cntr() dalam kode tersebut bertugas untuk mengolah data yang diterima dari port serial. Fungsi ini menggunakan sebuah loop for untuk memindai setiap karakter dalam array dataSerialcntr yang berisi data serial yang diterima. Program memisahkan data ketika fungsi menemukan karakter koma (',') menandakan pemisahan antara data. Setelah data terpisah, fungsi mengecek apakah data tersebut sesuai dengan format yang diharapkan. Jika sesuai, data-data numerik yang terpisah diubah ke tipe data yang sesuai menggunakan fungsi toFloat() dan toDouble() dan memproses data yang diterima agar dapat digunakan dalam aplikasi yang sesuai.

```

void olah_data_cntr()
{
    for (int i = 0; i < count_serial_cntr; i++)
    {
        if (dataSerialcntr[i] == ',')
        {
            dataSerialTempcntr[count_parse_serial_temp_cntr
] = '\0';
            dataSerialParsecntr[count_parse_serial_total_cn
tr] = String(dataSerialTempcntr);
            count_parse_serial_temp_cntr = 0;
        }
    }
}

```

```

        count_parse_serial_total_cntr++;
        continue;
    }
    dataSerialTempcntr[count_parse_serial_temp_cntr]
= dataSerialcntr[i];
    count_parse_serial_temp_cntr++;
}
dataSerialTempcntr[count_parse_serial_temp_cntr -
2] = '\0';
dataSerialParsecntr[count_parse_serial_total_cntr]
= String(dataSerialTempcntr);
count_parse_serial_temp_cntr = 0;
count_parse_serial_total_cntr++;
if (dataSerialParsecntr[0].equals("DATAS"))
{
    FFT = dataSerialParsecntr[1].toFloat();
    newdatacom =1;
    debug();
}
count_parse_serial_total_cntr = 0;
count_serial_cntr = 0;
slais_n_cntr = 0;
slais_r_cntr = 0;
}

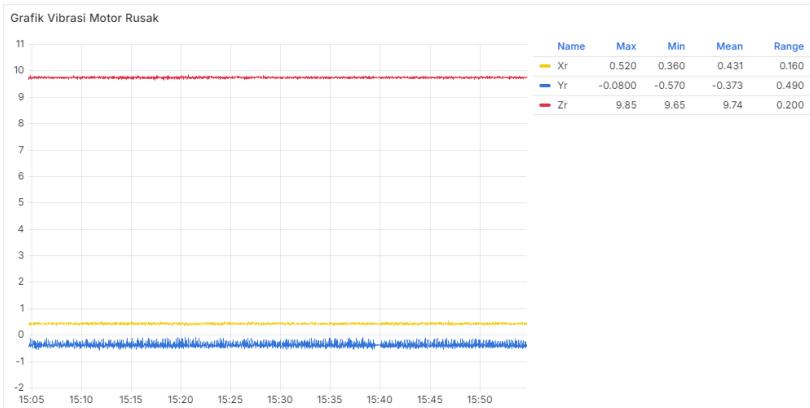
```

## 4.2. Pengujian Pengiriman Data Vibrasi

Pengujian dilakukan untuk mengetahui kondisi dari motor DC serta membandingkan kondisi motor DC yang rusak maupun kondisi motor DC yang baik. Data-data tersebut digunakan untuk mengidentifikasi kinerja yang terdapat beberapa faktor yang dapat dijadikan sebagai acuan untuk mengetahui seberapa besar tingkat vibrasi yang dihasilkan secara baik ke dalam system *Internet of Things*.



**Gambar 4.3.** Pengiriman data vibrasi motor baru



**Gambar 4.4.** Pengiriman data vibrasi motor rusak

Pengiriman data vibrasi dilakukan dengan menjalankan program arduino di STM32F401CCU6 dan ESP32, dimana program menjalankan program membaca sensor MPU6050 dan mengirimkan data ke dalam jaringan internet dan data vibrasi akan diunduh dengan format .xlsx atau Excel. Setelah data vibrasi dikirimkan oleh program-prgram yang telah dibuat, data vibrasi akan diterima oleh system IoT dengan informasi data telah sukses terkirim yang ditunjukkan pada Gambar 4.1 dan Gambar 4.2. Selanjutnya, program *python* untuk mengirimkan data vibrasi yang telah disimpan dalam excel untuk

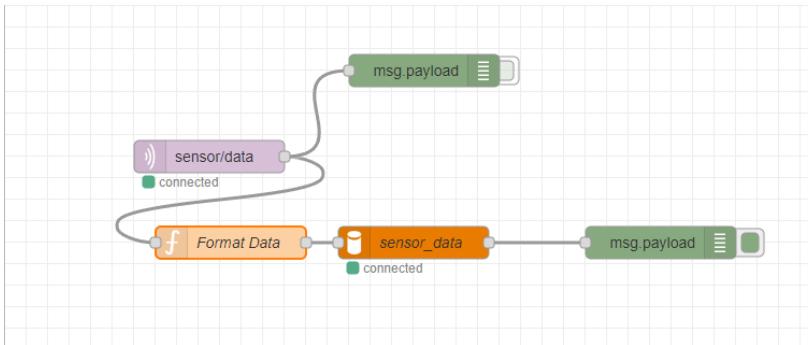
meneruskan data vibrasi ke mikrokontroler. Pada saat menjalankan program ekstensi, hal pertama yang dilakukan adalah membuka *Command Prompt* dengan mengarahkan lokasi file ekstensi *python*. Kemudian melakukan perintah program dengan mengetik *python datavibrasi.py 0g.xlsx 2 10 1.0* yang mewakili nama program ekstensi dan nama file excel data vibrasi yang dimulai pada baris 2 dan berakhir pada baris 10 dengan kecepatan pengiriman 1 detik setiap data vibrasi.

Development board yang digunakan berbasis mikrokontroler ESP32 dengan bahasa pemrograman C. Data yang dikirimkan adalah data vibrasi, dimana data tersebut telah terkirim dari computer file excel ke mikrokontroler STM32F401CCU6, yang dimana data telah diterima oleh STM32F401CCU6 akan dikirimkan melalui komunikasi serial dengan menghubungkan pin RX dan TX STM32F401CCU6 ke RX dan TX ESP32 secara terbalik. Penggunaan data vibrasi tersebut dengan alasan sensor telah mendapatkan data yang disimpan di excel. Pengiriman data dari ESP32 dilakukan setiap 1 detik sekali menggunakan protocol komunikasi MQTT. Pemilihan penggunaan protokol MQTT ini karena dikenal sebagai protokol yang ringan dan aman dikarenakan sebelum terkoneksi diwajibkan untuk memasukkan topik pada program, khususnya ketika digunakan untuk perangkat mikrokontroler. Data yang akan dipublish atau dikirimkan ke *message broker* menggunakan topik “*sensor/data*”. Topik ini adalah sebuah kata kunci yang digunakan oleh *client* lain agar dapat menerima data ketika melakukan *subscribe*.

### 4.3. Pengujian Sistem Internet of Things

#### 4.3.1. Pengujian Konfigurasi NodeRed

Konfigurasi NodeRed untuk dapat melakukan *publish* dan *subscribe*. Ketika ingin melakukan pengiriman data protocol komunikasi MQTT yang sudah tersedia dan yang perlu dilakukan adalah melakukan konfigurasi alamat server broker dan topiknya apa. Hal yang sama ketika akan mengirimkan atau menyimpan data pada database MySql, dibutuhkan beberapa konfigurasi seperti alamat server pada MySql, nama database, serta tabel yang akan dikirim data. Gambar 4.3 merupakan rancangan *flow* untuk mengirimkan dan menyimpan data ke *database* MySql.



**Gambar 4.5.** Flow pada NodeRed

Pada Gambar 4.3 terdapat beberapa node yaitu Data Sensor In dengan topik “*sensor/data*”, plot nilai input dari parsing data vibrasi berlabel “*Format Data*”, node *database* MySql, serta msg.payload. untuk mengubah data format data dalam gambar terdapat fungsi untuk memasukkan data ke dalam table *database* sesuai dengan jenis data. Data Sensor In ini sebuah node Broker MQTT dengan nama “*sensor/data*” untuk dapat melakukan *subscribe* data yang di-*publish* oleh ESP32. *Function* sebuah node fungsi untuk melakukan parsing data JSON ke dalam format yang dibutuhkan oleh *database* MySql. Node *database* MySql merupakan node yang membantu dalam menyimpan data ke dalam *database*. Node msg.payload merupakan node debug yang membantu dalam testing atau ujicoba sebuah flow karena datanya akan ditampilkan pada jendela *debugging* NodeRed untuk diteruskan. Pada node *Format Data*, dalam melakukan parsing data perlu menambahkan perintah dengan menggunakan pemrograman JavaScript.

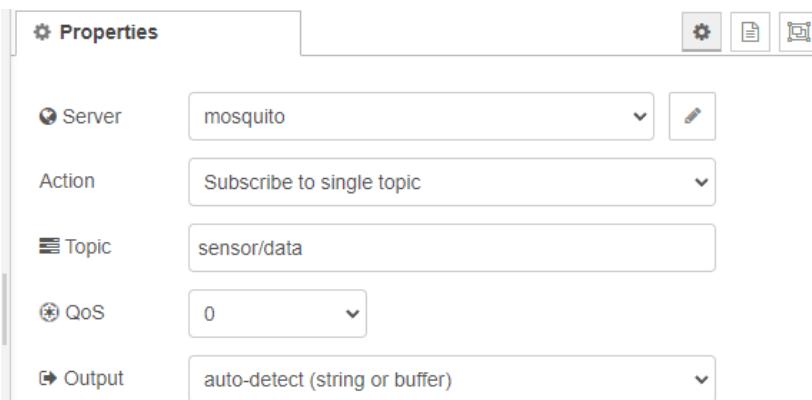
Program prasing data dengan mengubah data je dalam format JSON untuk dapat dilakukan proses selanjutnya. Kemudian, sebuah query SQL dengan menggunakan timestamp dan data sensor yang telah di parsing sebelumnya. Setelah menambahkan query SQL dan data yang terkait, query akan dikirim ke sistem basis data untuk dieksekusi atau diproses lebih lanjut.

```
var data = JSON.parse(msg.payload);
var date = new Date();

date.setHours(date.getHours() + 0);

var currentTimestamp = date.toISOString().slice(0, 19).replace('T', ' ');

var query = "INSERT INTO sensor_data (timestamp, X, Y, Z) VALUES (?, ?, ?, ?)";
msg.topic = query;
msg.payload = [currentTimestamp, data.x, data.y, data.z];
return msg;
```



**Gambar 4.6.** Pengaturan message queing telemetry transport

**Properties**

 Host	127.0.0.1
 Port	3306
 User	root
 Password	
 Database	sensor_data
 Timezone	±hh:mm
 Charset	
 Name	Name

**Gambar 4.7.** Node database Mysql

#### 4.3.2. Pengujian Data Database PhpMysql

Database MySQL membutuhkan sebuah pemrograman server yang digunakan untuk mengembangkan situs web dinamis. Kode PHP dijalankan di server, menghasilkan HTML yang kemudian dikirim ke browser web. PHP dan MySQL sering digunakan bersama untuk membangun aplikasi web dinamis. PHP bertindak sebagai jembatan antara *frontend* situs web dan basis data MySQL di *backend*. Dengan PHP, pengembang dapat membuat skrip yang melakukan kueri terhadap basis data MySQL, mengambil informasi, dan kemudian menampilkan informasi tersebut di halaman web dalam format yang dapat dibaca pengguna. Penghubungan *database* MySQL dengan computer dapat menjalankan *software* Xampp untuk mengaktifkannya. Tabel database dapat ditunjukan pada Gambar 4.6

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** sensor\_data
- Table:** sensor\_data
- Table Structure View:** Shows columns: id (int(11)), Frek (float), Xb (float), Yb (float), Zb (float), Fb (float), Fr (float). Primary key is id.
- Action Bar:** Includes buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers, and a toolbar with icons for Change, Drop, More, Primary, Unique, Index, Spatial, and Fulltext.
- Status Bar:** Shows "With selected:" and various edit and search options.

**Gambar 4.8. Database Mysql**

Gambar 4.7 *database* terdapat table yang menunjukan nama-nama dari data yang akan disimpan di dalam server. Data tersebut merupakan sebuah data yang dikirimkan melalui NodeRed, yang sebelumnya dikirimkan ke dalam alamat “data”. Data-data yang telah masuk ke dalam *database* akan ditampilkan sementara di dalam tabel yang sudah disesuaikan penamaannya.

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** sensor\_data
- Table:** sensor\_data
- Data View:** Shows 15 rows of data with columns: id, Frek, Xb, Yb, Zb, Fb, Fr.
- Action Bar:** Includes buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers, and a toolbar with icons for Copy, Delete, Edit, and More.
- Status Bar:** Shows "Number of rows: 25" and various search and sort options.

	id	Frek	Xb	Yb	Zb	Fb	Fr
[checkbox]	1	0.0333111	0.43	-0.36	9.59	0.00524246	0.00531321
[checkbox]	2	0.0666222	0.39	-0.38	9.58	0.00529689	0.00415359
[checkbox]	3	0.0999334	0.41	-0.37	9.56	0.00710042	0.00462298
[checkbox]	4	0.133244	0.37	-0.38	9.59	0.0037473	0.00434831
[checkbox]	5	0.166556	0.39	-0.38	9.57	0.00257463	0.00487925
[checkbox]	6	0.199867	0.38	-0.39	9.56	0.00467819	0.00250194
[checkbox]	7	0.233178	0.37	-0.37	9.57	0.0044325	0.00337814
[checkbox]	8	0.266489	0.39	-0.4	9.59	0.00169857	0.00491096
[checkbox]	9	0.2998	0.4	-0.37	9.6	0.00493765	0.00401342
[checkbox]	10	0.333111	0.4	-0.38	9.58	0.00616455	0.00120084
[checkbox]	11	0.366422	0.39	-0.36	9.59	0.0035722	0.00366741
[checkbox]	12	0.399734	0.43	-0.37	9.58	0.0018038	0.0029765
[checkbox]	13	0.433045	0.41	-0.36	9.58	0.00488116	0.00337644
[checkbox]	14	0.466356	0.37	-0.4	9.59	0.00464382	0.00348208
[checkbox]	15	0.499667	0.42	-0.36	9.57	0.00525721	0.00315188

**Gambar 4.9. Hasil data masuk database**

#### 4.3.3. Pengujian Dashboard Grafana

Pembuatan hasil *dashboard* memerlukan sebuah *datasource* pada Grafana. *Datasource* yang dimaksud disini merupakan sebuah *database* yang yang akan diambil datanya kemudian ditampilkan pada sebuah *dashboard*. *Dashboard* dilakukan dengan menambahkan

panel serta jenis visualisasi yang kita inginkan, visualisasi yang ditawarkan bisa berupa Graph, Gauge, Heatmap, bar gauge, dan yang lain. Panel ini nantinya yang akan menampilkan data pada dashboard, pada panel juga terdapat query inspector untuk mengambil data pada data source yang sebelumnya telah dikonfigurasi.



**Gambar 4.10.** Hasil data pada *dashboard*

Gambar 4.8. menampilkan hasil *dashboard* yang masing-masing menampilkan data spektrum vibrasi. Dimana data disesuaikan dengan data yang diterima oleh mikrokontroler dan dimasukan ke dalam *datasource*. Data yang ditampilkan pada *dashboard* bisa difilter berdasarkan waktu misalkan 5, 15, 30 menit terakhir atau bahkan jam dan hari. Agar dashboard terlihat perjalanan datanya, dapat dilakukan dengan mengubah waktu *refresh* dengan pilihan setiap 5 detik sampai dengan *refresh* per hari.

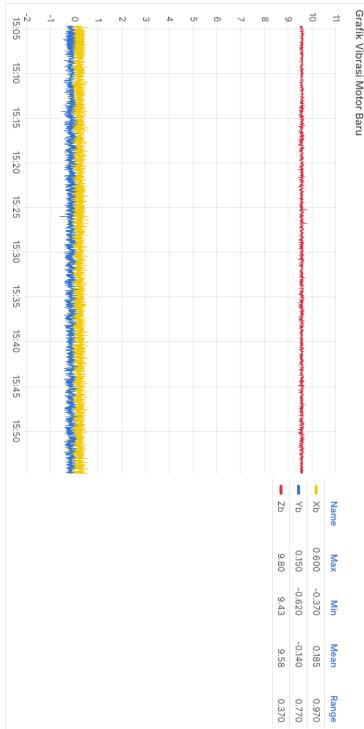
#### 4.4. Pengujian Pengukuran Data

Pengujian pengukuran data dilakukan dengan mengambil nilai vibrasi motor DC dengan sensor untuk mendapatkan nilai sumbu X, Y, dan Z dan dilakukan perhitungan total, serta memperhatikan perbedaan pada grafik spektrum vibrasi dari data vibrasi yang ditampilkan pada sistem *Internet of Things*. Data spektrum vibrasi berasal dari dua kondisi motor DC yang baru dan rusak yang masing-masing diberikan beban sebesar 0 gram, 20 gram, 40 gram, 60 gram,

80 gram, dan 100 gram. Spektrum vibrasi ditampilkan melalui *dashboard* Grafana yang akan dilakukan analisis data vibrasi dengan memperhatikan puncak magnitude dan frekuensi, serta karakteristik bentuk spektrum.

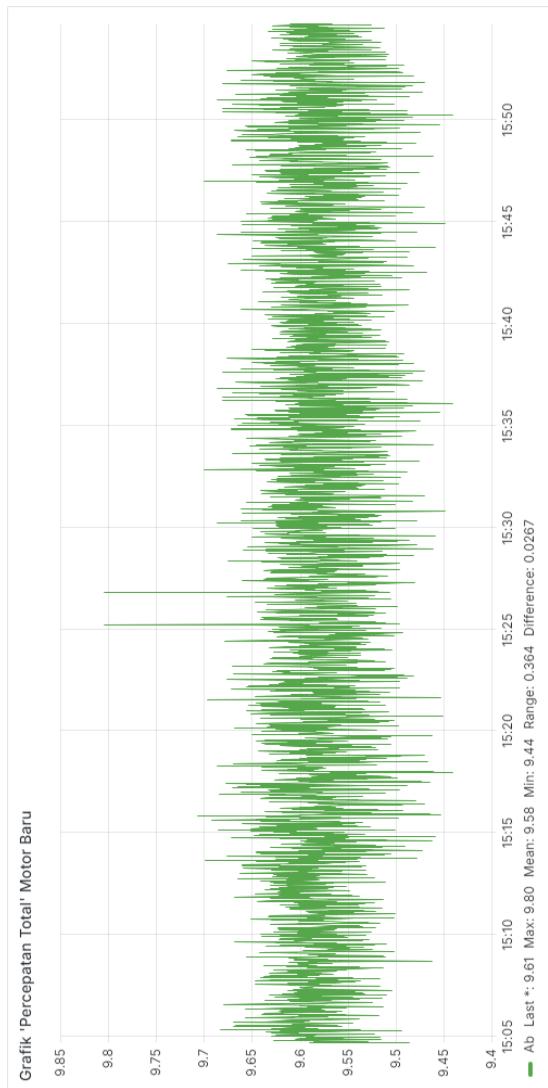
#### 4.4.1. Beban 0 Gram

Gambar 4.11 menunjukkan hasil data vibrasi motor baru dengan beban 0 gram. Dimana nilai maksimum sumbu X sebesar 0,6 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,15 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,8 ( $\text{m/s}^2$ ). Nilai minimum sumbu X -0,370 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,62 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,43 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,185 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,140 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,58 ( $\text{m/s}^2$ ).



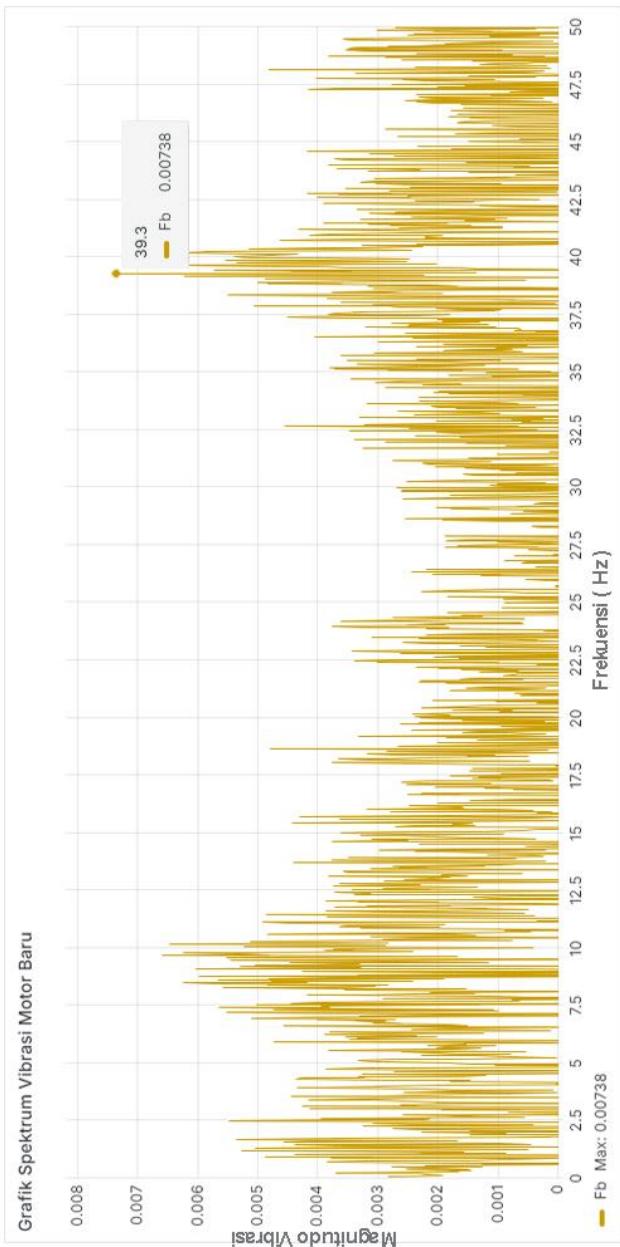
**Gambar 4.11.** Data vibrasi motor baru 0 gram

Gambar 4.12 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



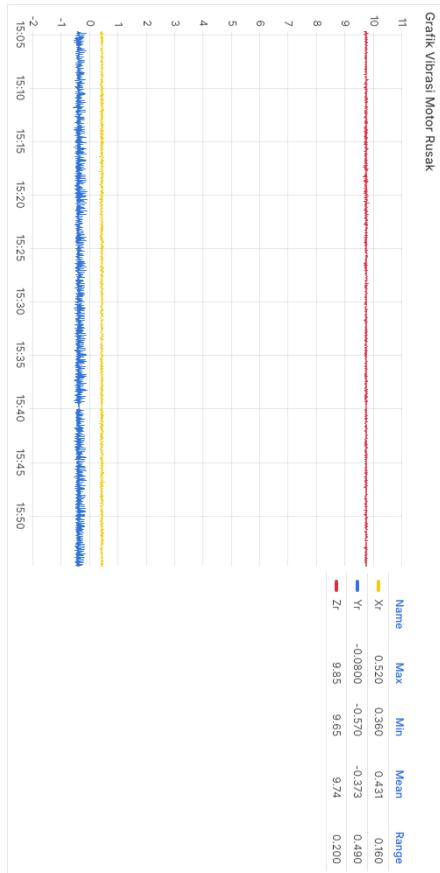
**Gambar 4.12.** Percepatan total vibrasi motor baru 0 gram

Grafik spektrum vibrasi pada Gambar 4.13 motor DC baru dengan beban 0 gram ditandai dengan puncak magnitude tertinggi sekitar 0,00738 m/s<sup>2</sup> pada frekuensi sekitar 39,3 Hz. Spektrum terbentuk lebih konsisten dengan puncak terdistribusi cukup sama pada seluruh rentang frekuensi. Terdapat beberapa puncak yang tinggi yang menunjukkan variasi magnitude vibrasi yang lebih rendah di rentang frekuensi. Gambar 4.13 grafik sumbu x menunjukan magnitudo vibrasi dan sumbu y menunjukan magnitude frekuensi.



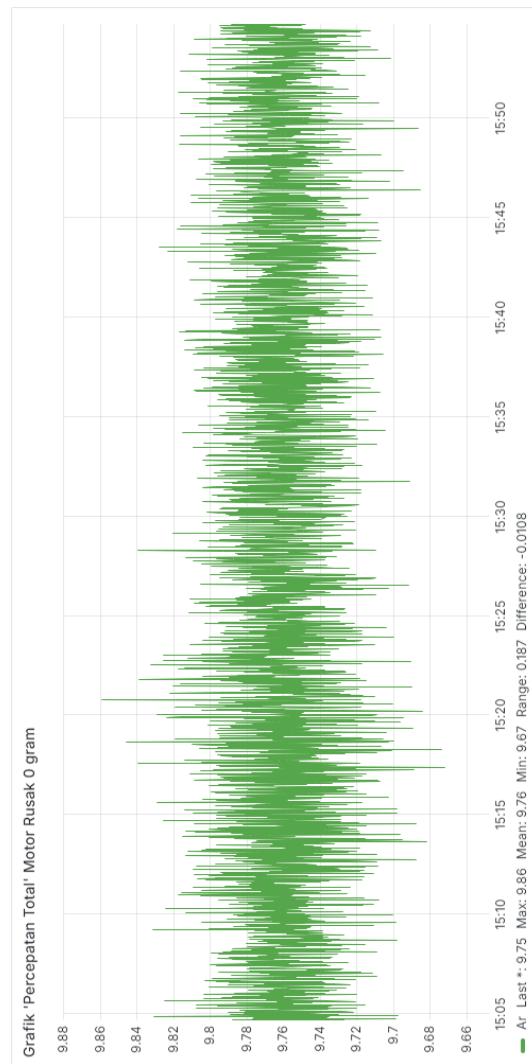
Gambar 4.13. Spektrum vibrasi motor DC baru 0 gram

Gambar 4.14 menunjukkan hasil data vibrasi motor rusak dengan beban 0 gram. Dimana nilai maksimum sumbu X sebesar 0,520 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,80 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,85 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar 0,360 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,570 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,65 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,431 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,373 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,74 ( $\text{m/s}^2$ ).



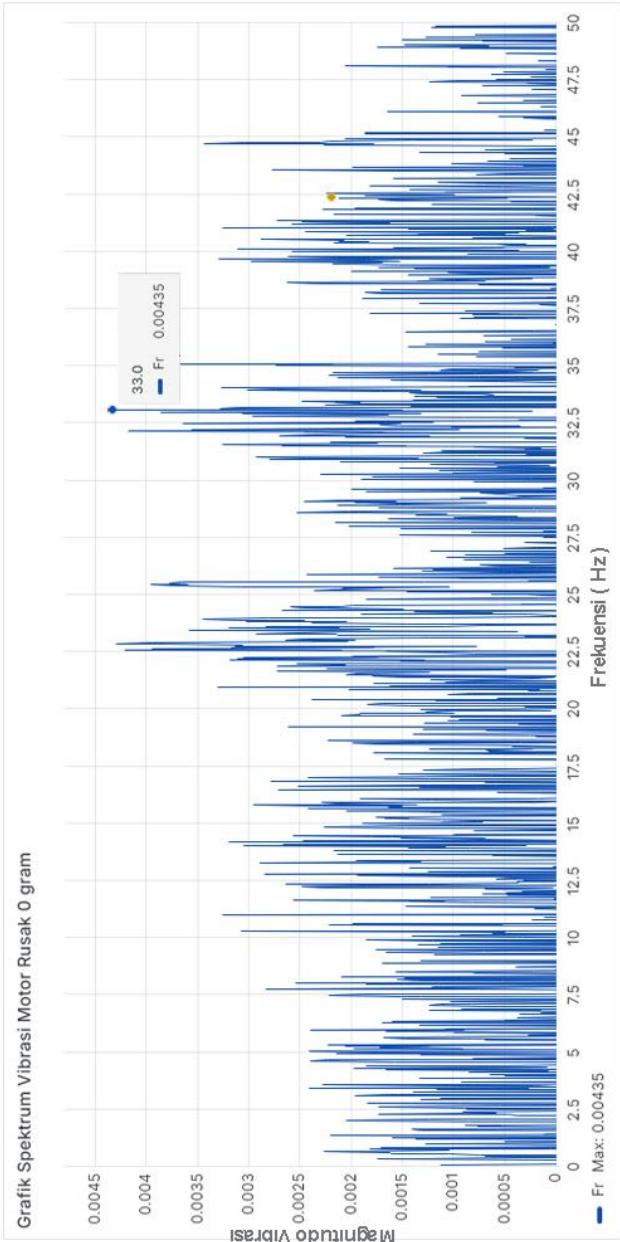
**Gambar 4.14.** Data vibrasi motor rusak 0 gram

Gambar 4.15 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



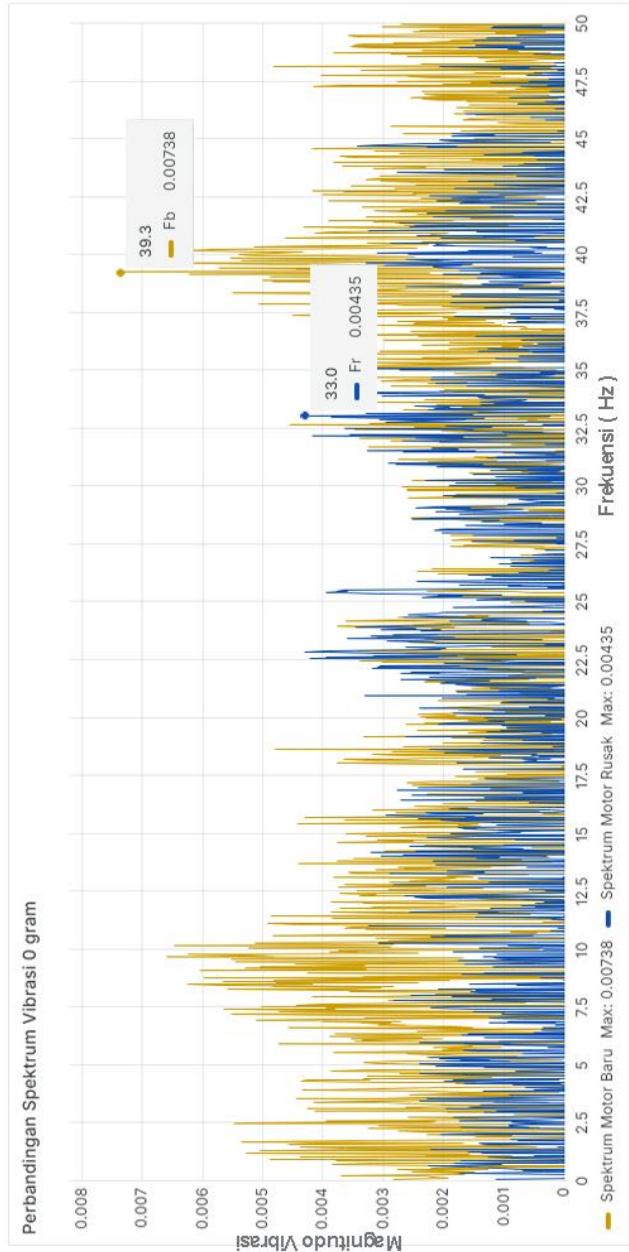
**Gambar 4.15.** Percepatan total vibrasi motor rusak 0 gram

Grafik spektrum vibrasi pada Gambar 4.16 motor DC rusak dengan beban 0 gram ditandai dengan puncak magnitude tertinggi sekitar 0,00435 m/s<sup>2</sup> pada frekuensi sekitar 33,0 Hz. Spektrum terbentuk lebih tidak beratur dengan banyak puncak terdistribusi cukup sama pada seluruh rentang frekuensi. Magnitude puncak tertinggi lebih rendah dibandingkan dengan motor baru, yang bisa menunjukkan bahwa secara keseluruhan vibrasi lebih lemah atau bahwa puncak vibrasi utama telah bergeser karena kerusakan. Terdapat beberapa puncak yang tidak teratur yang menunjukkan adanya kerusakan yang mempengaruhi kinerja dan menimbulkan getaran tambahan pada frekuensi yang berbeda-beda. Gambar 4.14 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



**Gambar 4.16.** Spektrum vibrasi motor DC rusak 0 gram

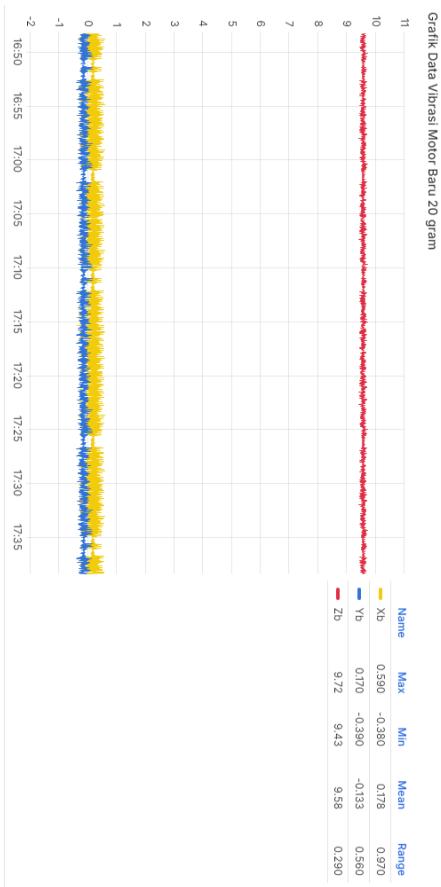
Motor baru memiliki magnitude lebih tinggi yang dapat menunjukkan kinerja yang lebih kuat pada titik frekuensi. Motor rusak memiliki magnitude lebih rendah, yang mungkin mengindikasikan kehilangan efisiensi atau kemampuan motor untuk beroperasi pada kinerja yang sama seperti motor baru. Frekuensi puncak pada motor baru lebih tinggi dibandingkan dengan motor rusak, yang dapat menunjukkan bahwa kerusakan telah menyebabkan perubahan pada karakteristik dinamis dari motor, seperti pergeseran frekuensi alami atau resonansinya. Motor baru menunjukkan spektrum yang lebih konsisten dan teratur. Sedangkan, motor rusak menunjukkan tingkat kekacauan yang lebih tinggi dalam spektrumnya yang menunjukkan ketidakstabilan yang mungkin disebabkan oleh komponen motor yang rusak, yang mengganggu kinerja motor. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.17.



Gambar 4.17. Perbandingan spektrum vibrasi

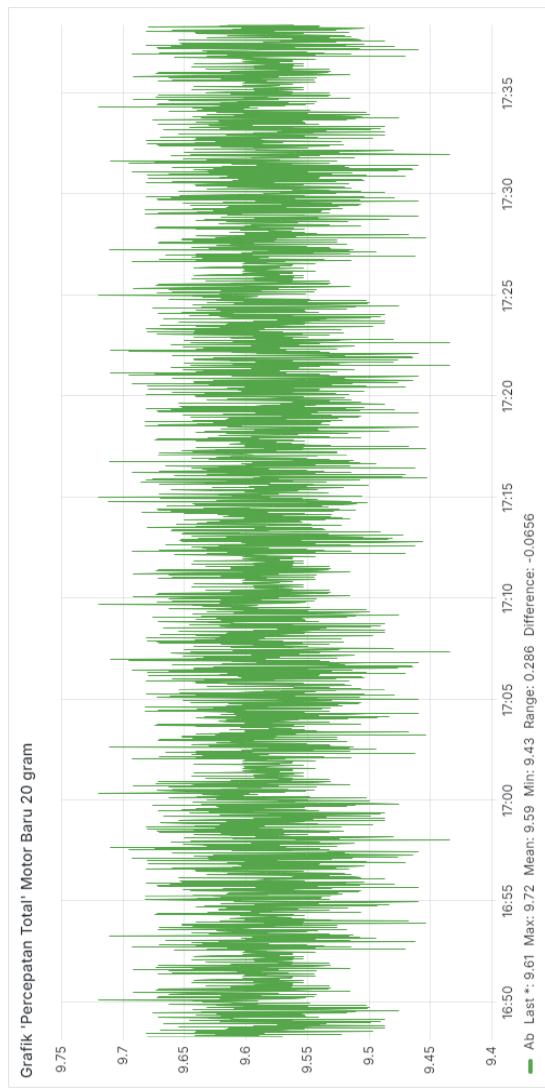
#### 4.4.2. Beban 20 Gram

Gambar 4.18 menunjukan hasil data vibrasi motor baru dengan beban 20 gram. Dimana nilai maksimum sumbu X sebesar 0,590 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,170 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,72 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -0,380 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,390 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,43 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,178 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,133 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,58 ( $\text{m/s}^2$ ).



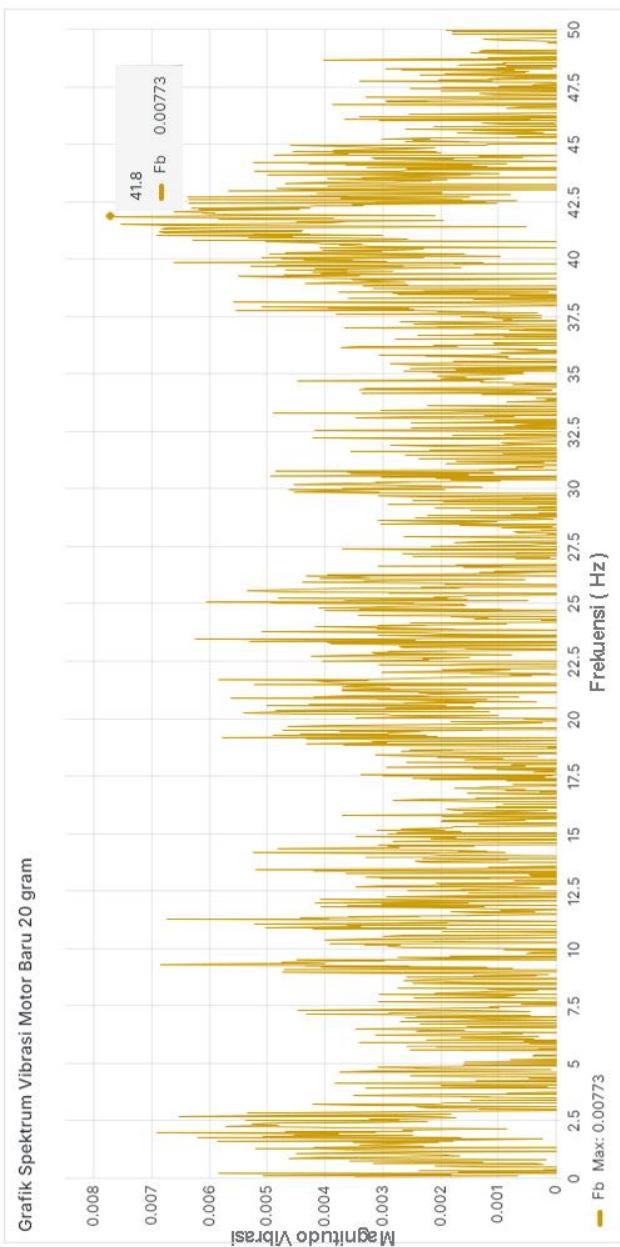
**Gambar 4.18.** Data vibrasi motor baru 20 gram

Gambar 4.19 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



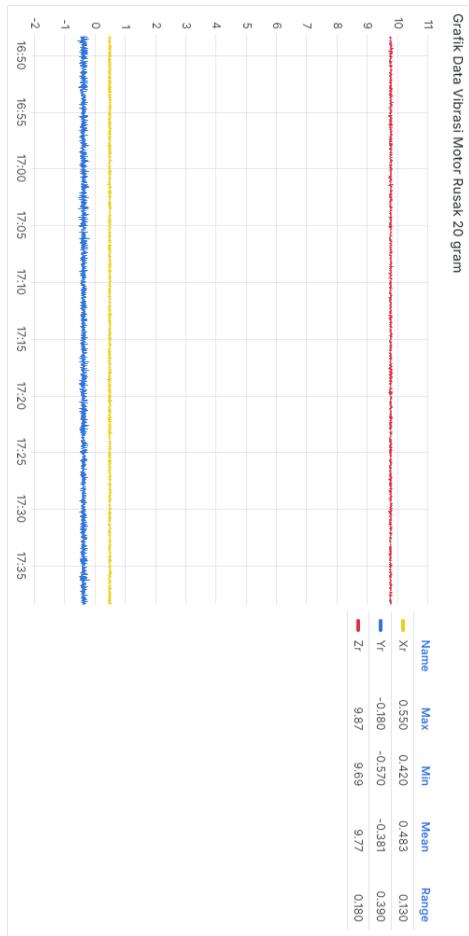
**Gambar 4.19.** Percepatan total vibrasi motor baru 20 gram

Grafik spektrum vibrasi pada Gambar 4.20 motor DC baru dengan beban 20 gram ditandai dengan puncak magnitude tertinggi sekitar 0,00773 m/s<sup>2</sup> pada frekuensi sekitar 41,8 Hz. Spektrum terbentuk lebih konsisten dengan puncak dominan yang jelas bahwa motor baru beroperasi dengan lebih stabil dan seimbang yang mengindikasikan karakteristik kinerja motor pada seluruh rentang frekuensi. Gambar 4.20 grafik sumbu x menunjukan magnitudo vibrasi dan sumbu y menunjukan magnitude frekuensi.



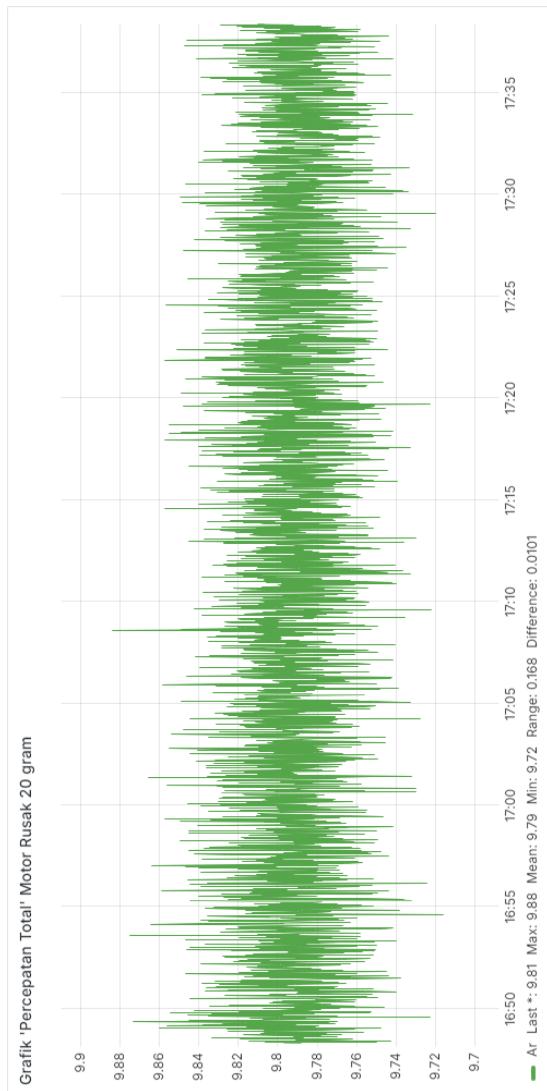
**Gambar 4.20.** Spektrum vibrasi motor DC baru 20 gram

Gambar 4.21 menunjukkan hasil data vibrasi motor rusak dengan beban 20 gram. Dimana nilai maksimum sumbu X sebesar 0,550 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,180 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,87 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar 0,420 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,570 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,69 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,483 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,381 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,77 ( $\text{m/s}^2$ ).



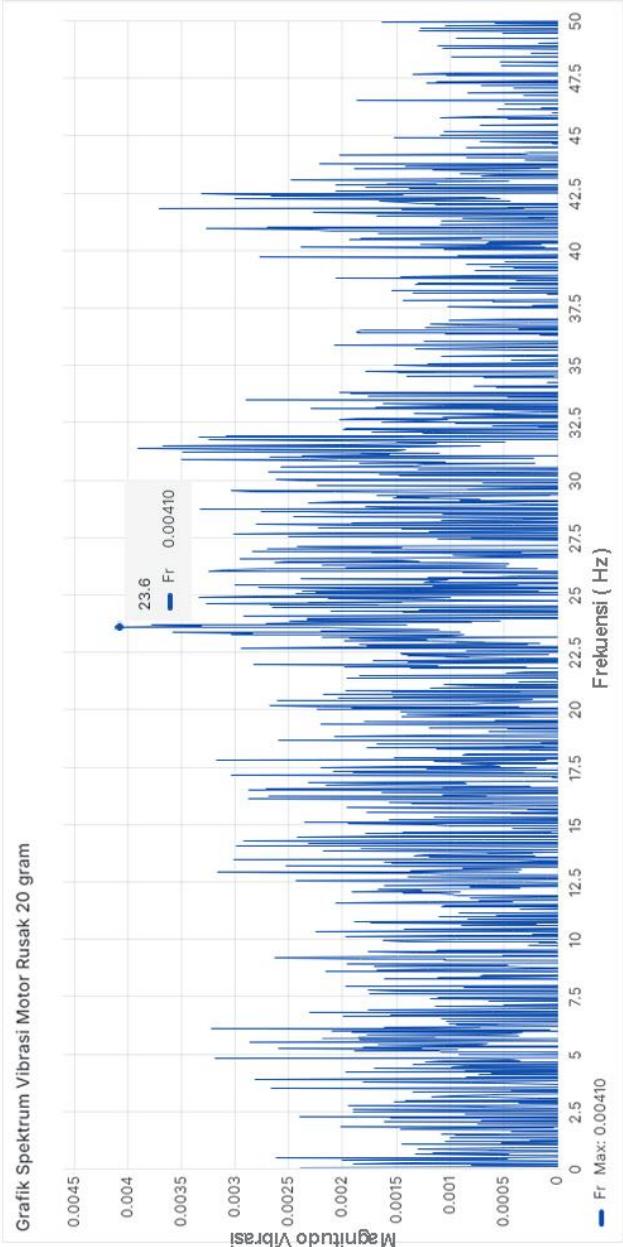
**Gambar 4.21.** Data vibrasi motor rusak 20 gram

Gambar 4.22 menunjukan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



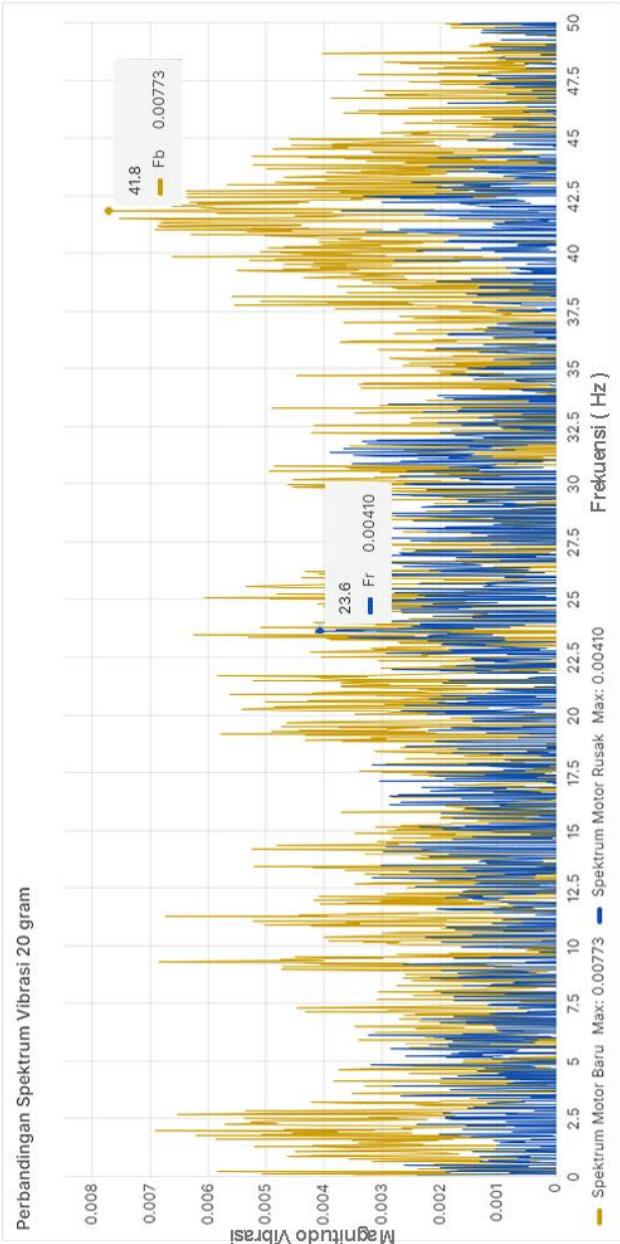
**Gambar 4.22.** Percepatan total vibrasi motor rusak 20 gram

Grafik spektrum vibrasi pada Gambar 4.23 motor DC rusak dengan beban 20 gram ditandai dengan puncak magnitude tertinggi sekitar 0,00410 m/s<sup>2</sup> pada frekuensi sekitar 23,6 Hz. Spektrum terbentuk lebih tidak beratur dengan puncak terdistribusi cukup sama pada seluruh rentang frekuensi. Magnitude puncak tertinggi lebih rendah dibandingkan dengan motor baru, yang bisa menunjukkan bahwa secara keseluruhan vibrasi lebih lemah atau bahwa puncak vibrasi utama telah bergeser karena kerusakan. Puncak magnitude pada frekuensi yang tidak teratur yang menunjukkan adanya kerusakan yang mempengaruhi kinerja dan menimbulkan getaran tambahan pada frekuensi yang berbeda-beda. Gambar 4.23 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitudo frekuensi.



**Gambar 4.23.** Spektrum vibrasi motor DC rusak 20 gram

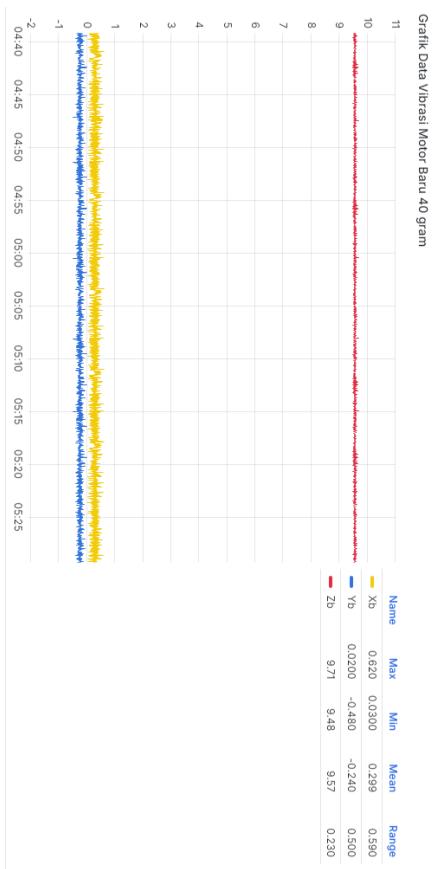
Spektrum motor DC baru menunjukkan satu frekuensi dengan magnitude yang sangat tinggi, yang mungkin menunjukkan karakteristik operasi kinerja normal. Sedangkan, motor rusak menunjukkan magnitude yang lebih rendah, yang bisa menandakan penurunan kinerja atau efisiensi. Motor baru tampak memiliki distribusi frekuensi yang lebih terfokus sementara motor rusak menunjukkan distribusi yang lebih luas, menandakan kemungkinan adanya kerusakan yang menyebabkan lebih banyak frekuensi terlibat dalam proses vibrasi. Perbedaan spektrum yang cukup signifikan dalam vibrasi yang dihasilkan oleh motor baru terlihat lebih konsisten dan terfokus menunjukkan bahwa beban 20 gram pada rotor tidak berpengaruh signifikan terhadap operasi normal. Sedangkan, motor yang rusak motor rusak menunjukkan karakteristik yang lebih beragam dan magnitude yang lebih rendah, menunjukkan kerusakan atau ketidakseimbangan yang mempengaruhi operasinya. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.24.



Gambar 4.24. Perbandingan spektrum vibrasi

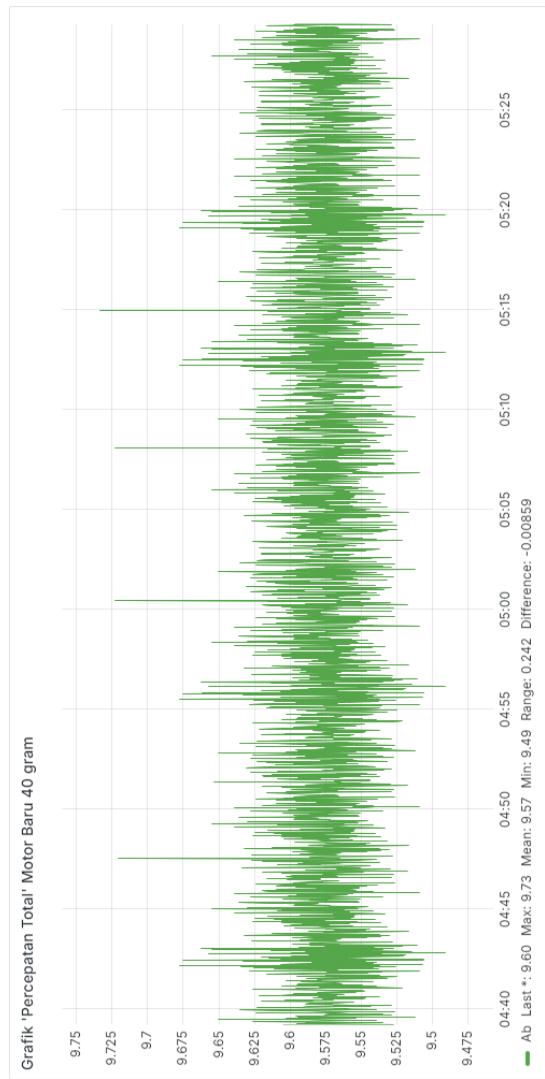
#### 4.4.3. Beban 40 Gram

Gambar 4.25 menunjukan hasil data vibrasi motor baru dengan beban 40 gram. Dimana nilai maksimum sumbu X sebesar 0,620 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,0200 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,71 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar 0,0300 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,480 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,48 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,299 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,240 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,57 ( $\text{m/s}^2$ ).



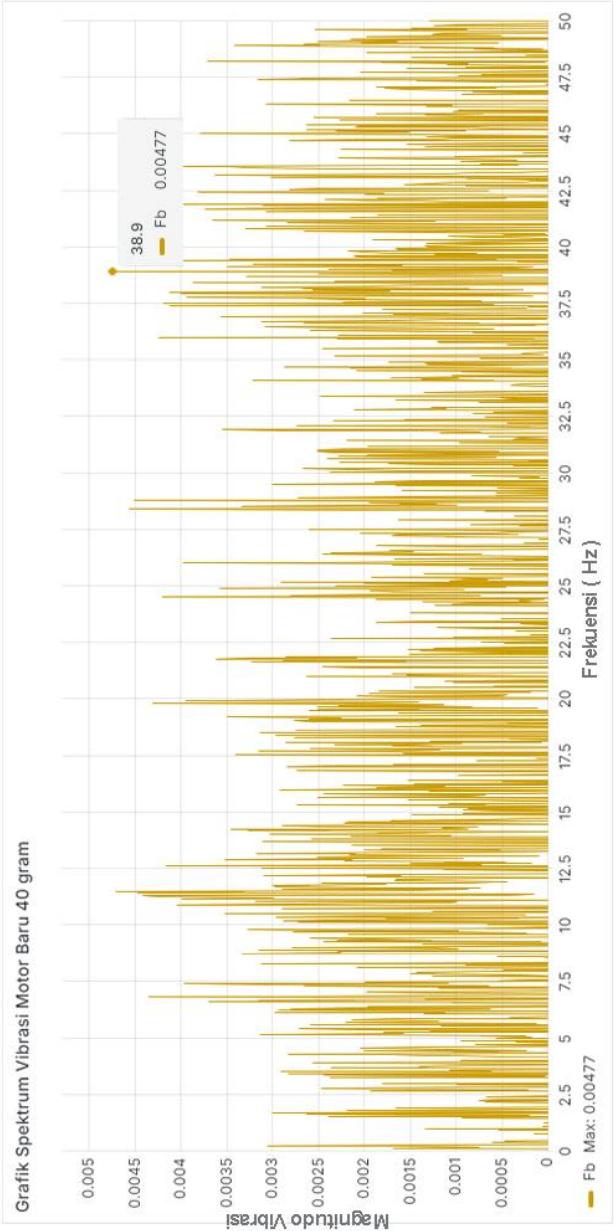
**Gambar 4.25.** Data vibrasi motor baru 40 gram

Gambar 4.26 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



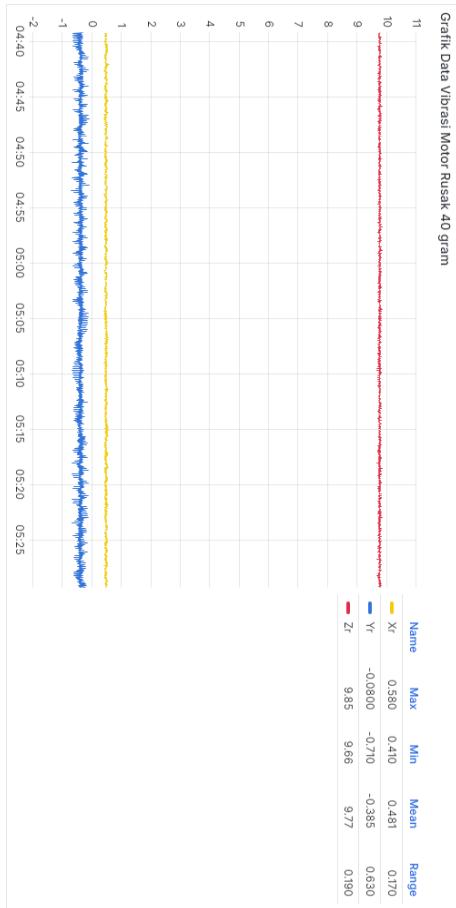
**Gambar 4.26.** Percepatan total vibrasi motor baru 40 gram

Grafik spektrum vibrasi pada Gambar 4.75 motor DC rusak dengan beban 40 gram ditandai dengan puncak amagnitude tertinggi sekitar  $0,00477 \text{ m/s}^2$  pada frekuensi sekitar 38,9 Hz. Puncak amplitude terjadi pada frekeunsi yang lebih rendah dibandingkan dengan pengujian dengan beban 20 gram. Spektrum terbentuk lebih konsisten dengan puncak dominan yang jelas bahwa motor baru beroperasi dengan lebih stabil dan seimbang yang mengindikasikan karakteristik kinerja motor pada seluruh rentang frekuensi. Gambar 4.27 grafik sumbu x menunjukan magnitudo vibrasi dan sumbu y menunjukan magnitude frekuensi.



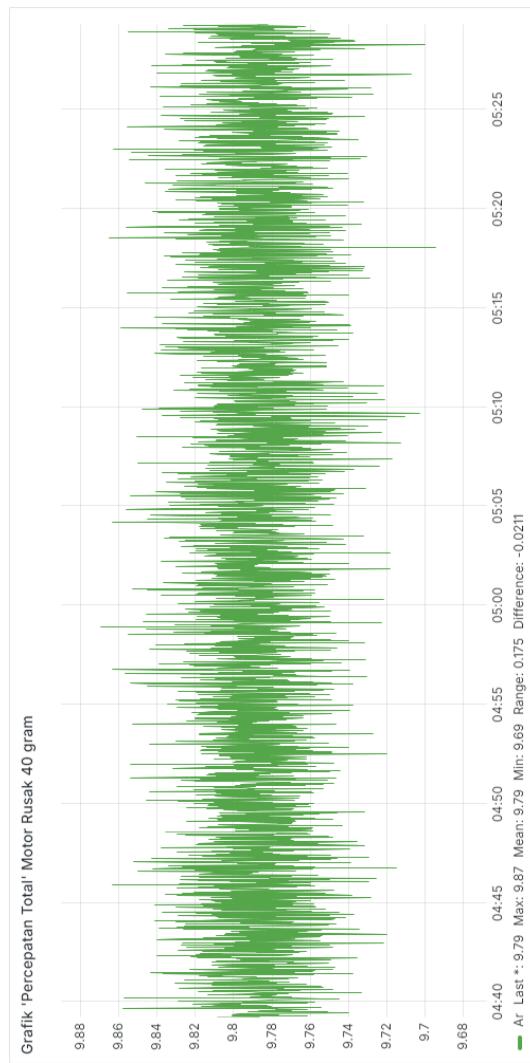
**Gambar 4.27.** Spektrum vibrasi motor DC baru 40 gram

Gambar 4.28 menunjukkan hasil data vibrasi motor rusak dengan beban 40 gram. Dimana nilai maksimum sumbu X sebesar 0,580 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,0800 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,85 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar 0,410 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,710 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,66 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,481 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,385 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,77 ( $\text{m/s}^2$ ).



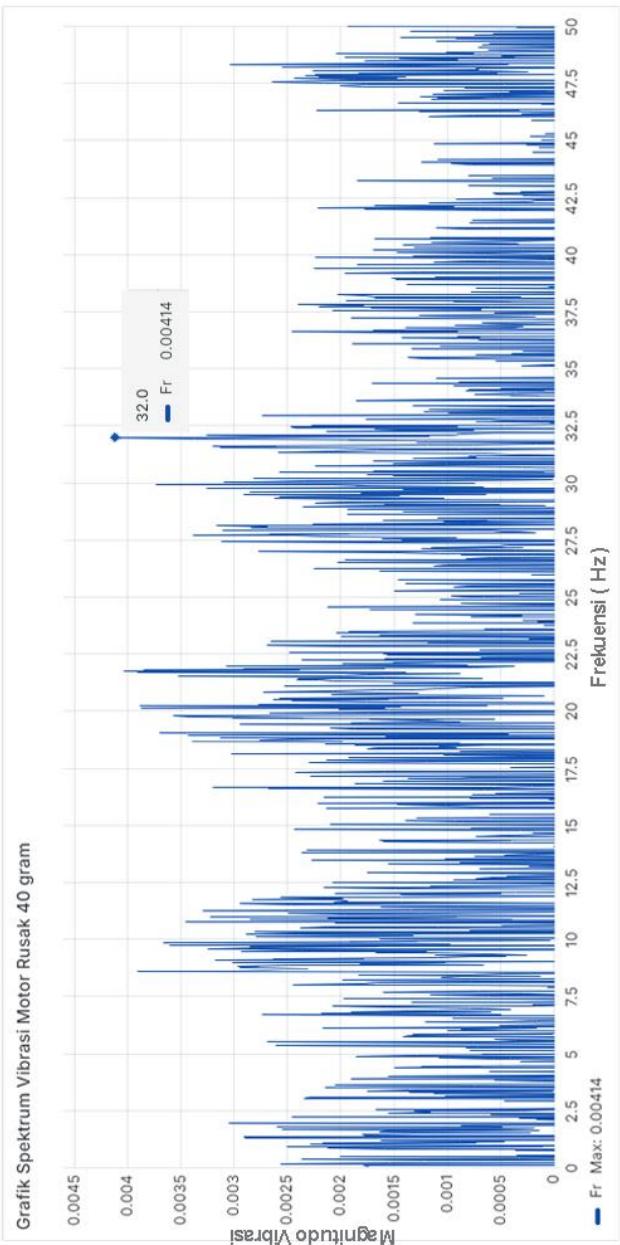
**Gambar 4.28.** Data vibrasi motor rusak 40 gram

Gambar 4.29 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



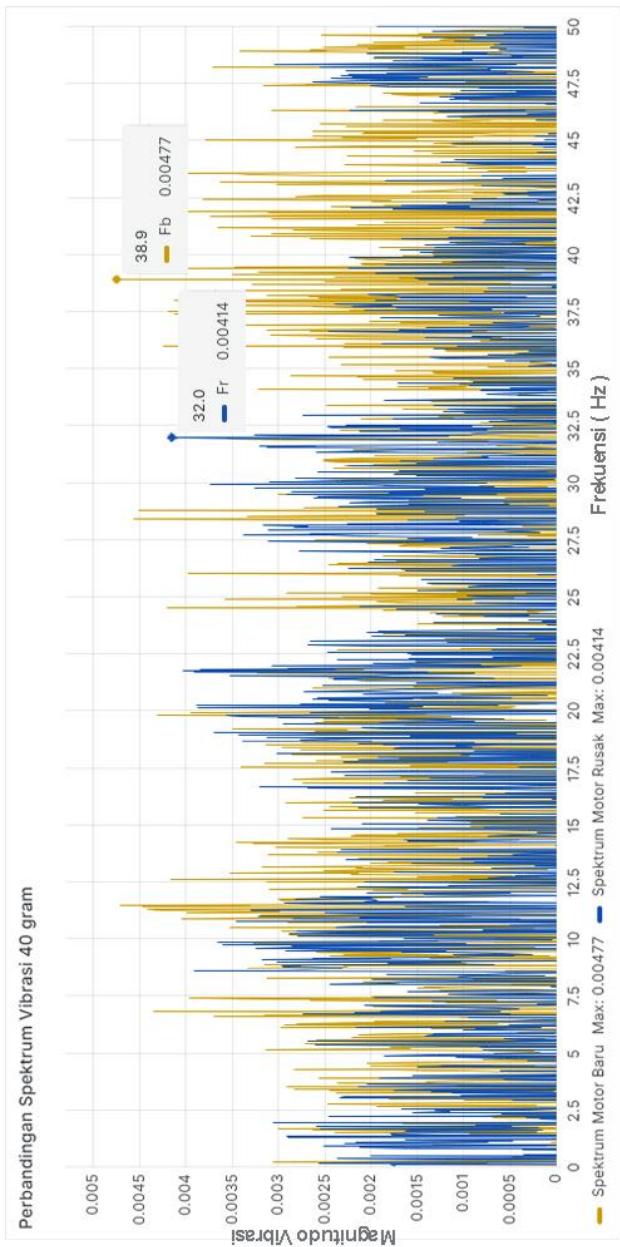
**Gambar 4.29.** Percepatan total vibrasi motor rusak 40 gram

Grafik spektrum vibrasi pada Gambar 4.30 motor DC rusak dengan beban 40 gram ditandai dengan puncak magnitude tertinggi sekitar  $0,00414 \text{ m/s}^2$  pada frekuensi sekitar 32,0 Hz. Spektrum terbentuk lebih tidak beratur dengan puncak terdistribusi dengan bentuk tidak sama pada seluruh rentang frekuensi. Puncak magnitude pada frekuensi yang tidak teratur yang menunjukkan adanya kerusakan yang mempengaruhi kinerja dan menimbulkan getaran tambahan pada frekuensi yang berbeda-beda. Gambar 4.30 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



Gambar 4.30. Spektrum vibrasi motor DC rusak 40 gram

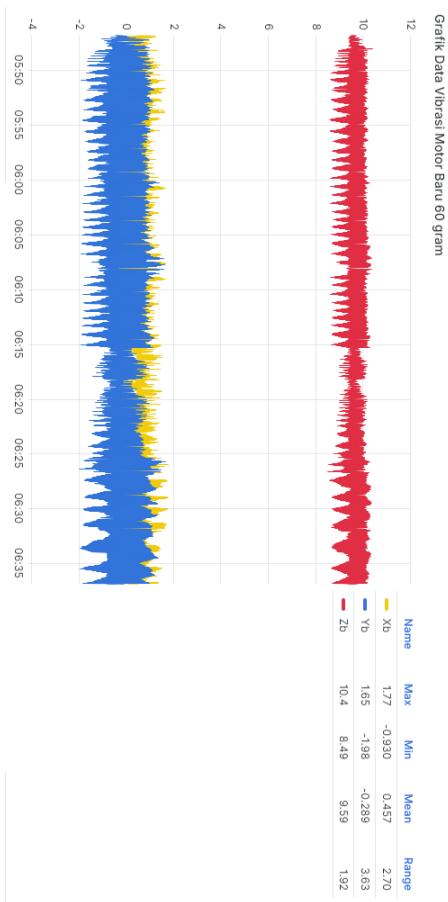
Motor baru memiliki magnitude puncak yang lebih tinggi dibandingkan dengan motor rusak, meskipun puncak tersebut terjadi pada frekuensi yang lebih rendah. Terjadi perubahan frekuensi dominan di kedua motor dari pengujian sebelumnya, yang bisa disebabkan oleh penambahan beban pada motor. Kedua motor menunjukkan pola distribusi frekuensi yang lebih kompleks dengan peningkatan beban, namun motor baru masih menunjukkan satu puncak yang lebih dominan. Penambahan beban pada motor mempengaruhi frekuensi dan magnitude vibrasi kedua motor, baik yang baru maupun yang rusak. Pada motor baru menunjukkan karakteristik vibrasi yang lebih stabil dengan satu frekuensi dominan yang lebih menonjol. Sedangkan, motor rusak menunjukkan pola yang lebih seragam dan kurang terdefinisi dengan baik yang bisa mengindikasikan kerusakan lebih lanjut atau karakteristik kinerja yang kurang optimal, meskipun mengalami peningkatan beban. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.31.



**Gambar 4.31.** Perbandingan spektrum vibrasi

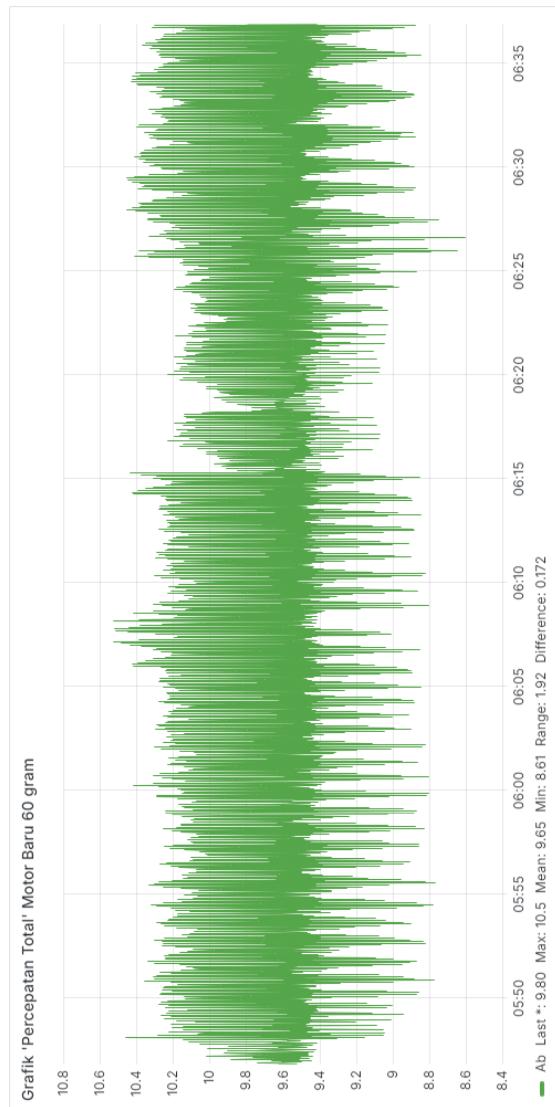
#### 4.4.4. Beban 60 Gram

Gambar 4.32 menunjukkan hasil data vibrasi motor baru dengan beban 60 gram. Dimana nilai maksimum sumbu X sebesar 1,77 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 1,65 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 10,4 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -0,930 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -1,98 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 8,49 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,457 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,289 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,59 ( $\text{m/s}^2$ ).



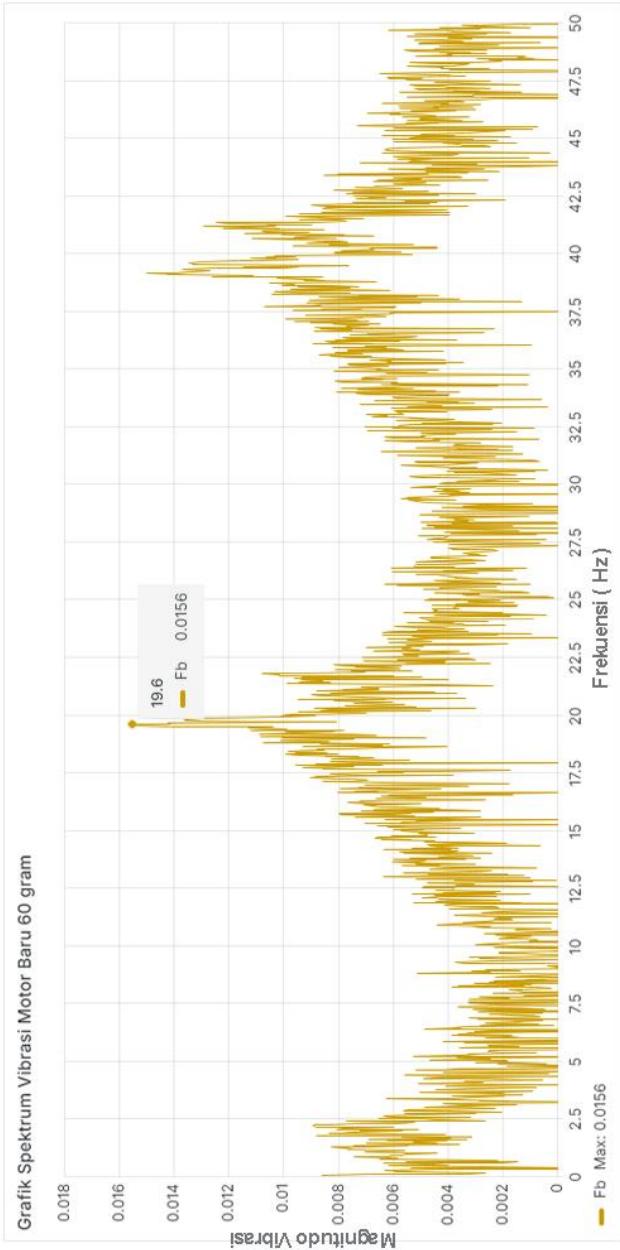
**Gambar 4.32.** Data vibrasi motor baru 60 gram

Gambar 4.33 menunjukan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



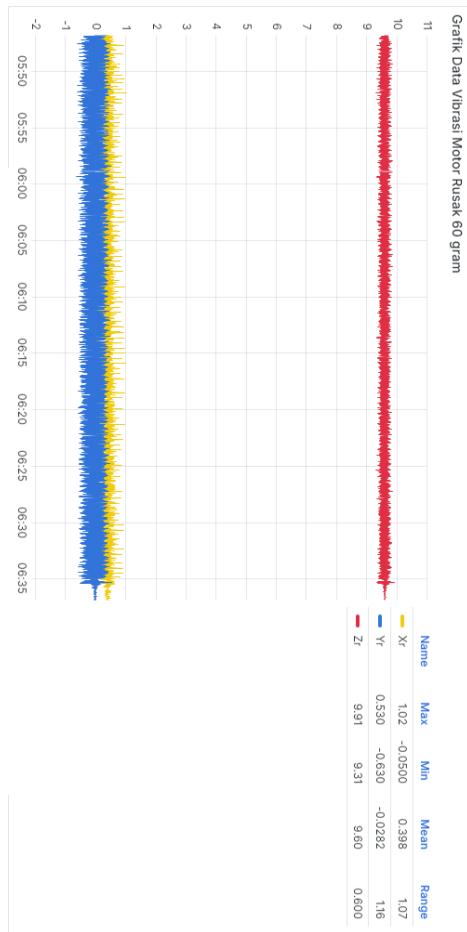
**Gambar 4.33.** Percepatan total vibrasi motor baru 60 gram

Grafik spektrum vibrasi pada Gambar 4.34 motor DC baru dengan beban 60 gram ditandai dengan puncak magnitude tertinggi sekitar 0,0156 m/s<sup>2</sup> pada frekuensi sekitar 19,6 Hz. Spektrum terbentuk lebih konsisten dengan puncak terdistribusi cukup sama pada seluruh rentang frekuensi. Spektrum terbentuk lebih konsisten dengan puncak dominan yang jelas bahwa motor baru beroperasi dengan lebih stabil dan seimbang yang mengindikasikan karakteristik kinerja motor pada seluruh rentang frekuensi. Gambar 4.34 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



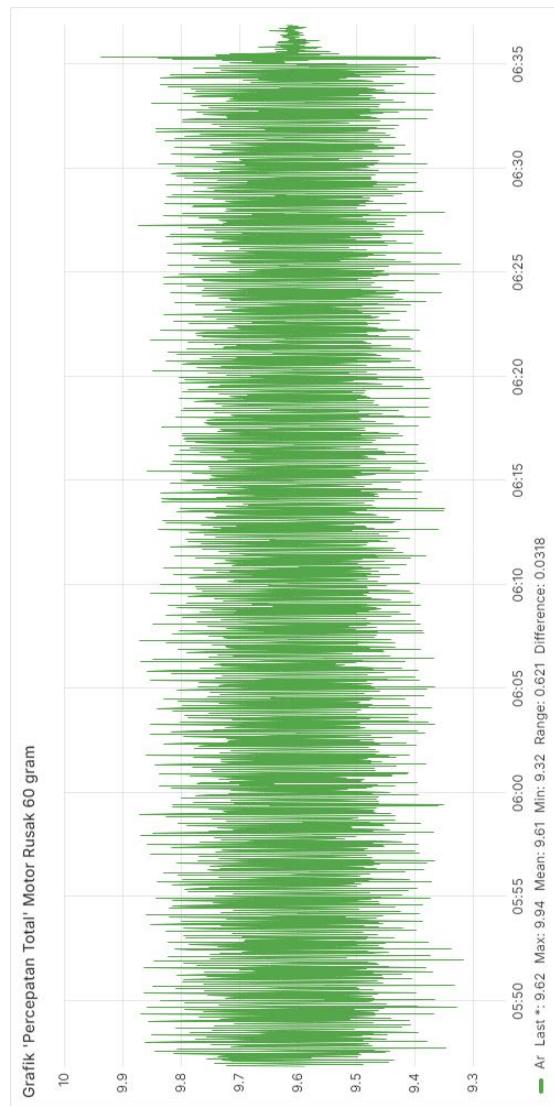
Gambar 4.34. Spektrum vibrasi motor DC baru 60 gram

Gambar 4.35 menunjukkan hasil data vibrasi motor rusak dengan beban 60 gram. Dimana nilai maksimum sumbu X sebesar 1,02 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,530 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,91 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -0,0500 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,630 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,31 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,396 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,0282 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,60 ( $\text{m/s}^2$ ).



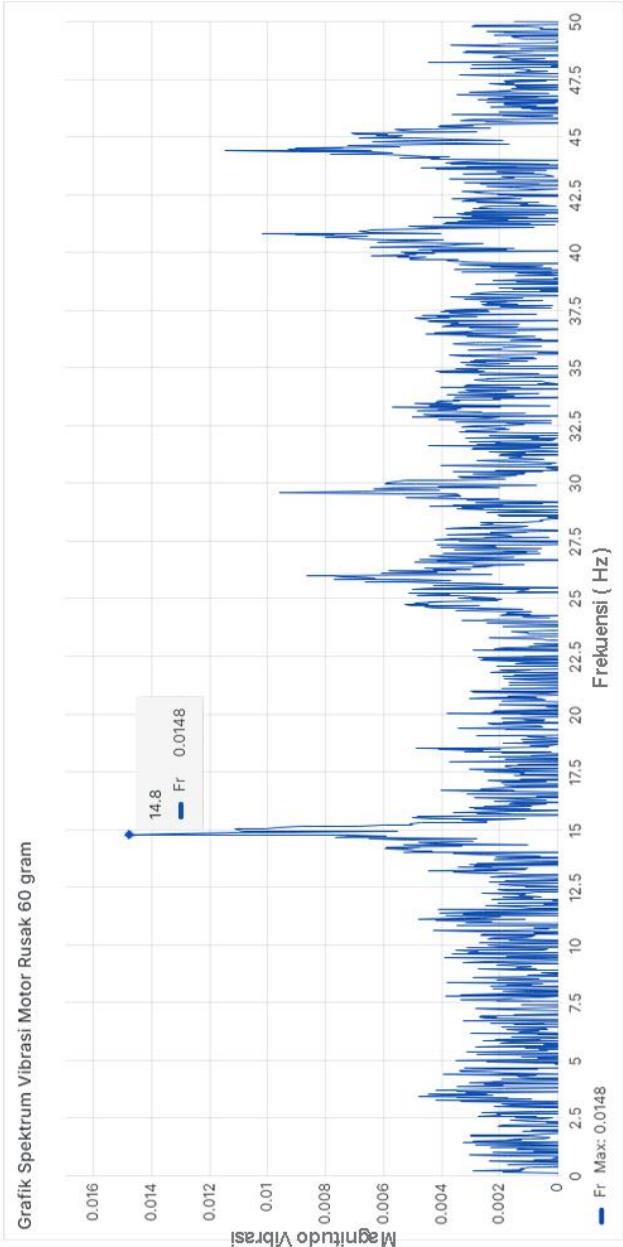
**Gambar 4.35.** Data vibrasi motor rusak 60 gram

Gambar 4.36 menunjukan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



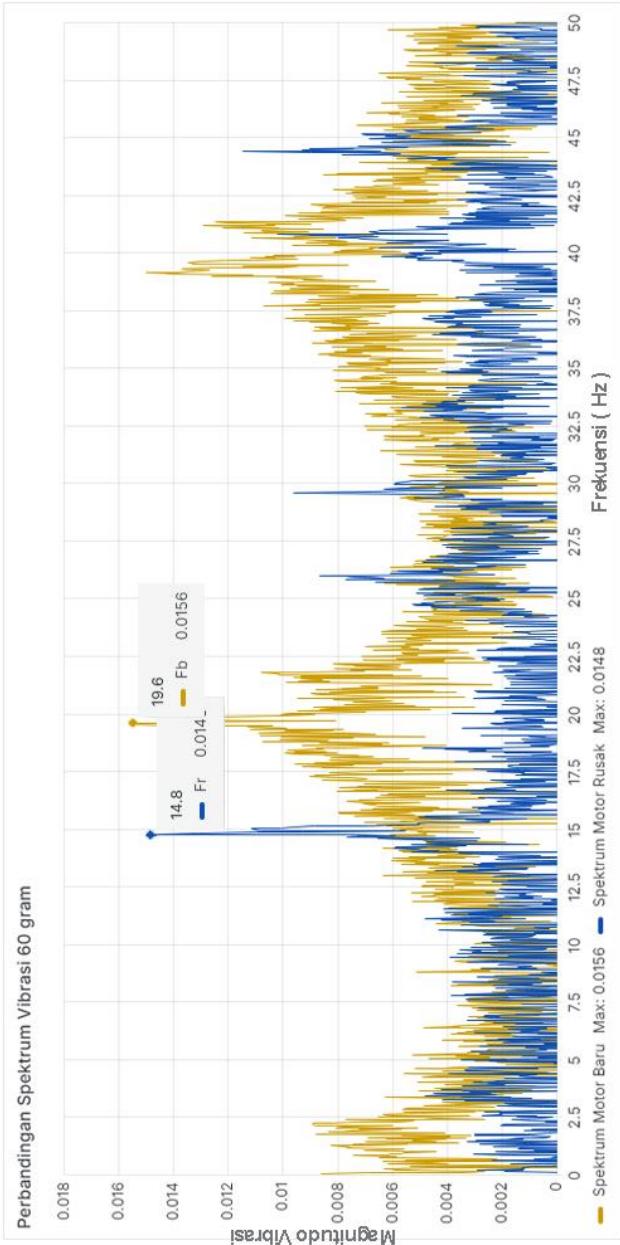
**Gambar 4.36.** Percepatan total vibrasi motor rusak 60 gram

Grafik spektrum vibrasi pada Gambar 4.37 motor DC rusak dengan beban 60 gram ditandai dengan puncak magnitude tertinggi sekitar  $0,0148 \text{ m/s}^2$  pada frekuensi sekitar 14,8 Hz. Menandakan adanya kerusakan atau masalah mekanis yang berbeda dibandingkan dengan motor baru. Spektrum menunjukkan beberapa puncak, namun secara keseluruhan magnitude cenderung lebih rendah dan lebih tersebar daripada motor baru. Gambar 4.37 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



Gambar 4.37. spektrum vibrasi motor DC rusak 60 gram

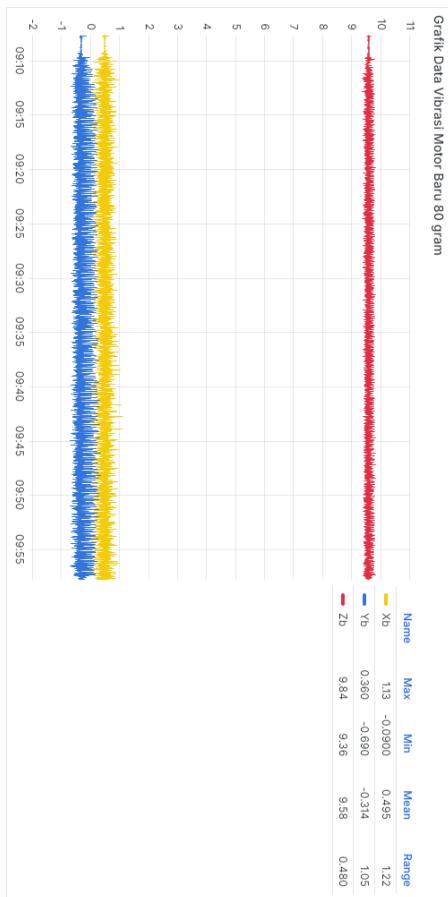
Puncak magnitude motor rusak hampir mendekati motor baru penyebaran spektrum vibrasi pada motor rusak lebih luas yang menunjukkan potensi kerusakan mekanis atau masalah ketidakseimbangan yang lebih beragam. Motor baru menunjukkan puncak pada frekuensi yang lebih tinggi dibandingkan motor rusak, yang mungkin mengindikasikan perbedaan dalam respon terhadap beban atau kondisi mekanis yang lebih baik. Motor baru menunjukkan penyebaran frekuensi yang lebih terfokus pada puncak tertentu, sementara motor rusak memiliki frekuensi yang lebih lebar dengan banyak puncak yang kurang dominan, mengindikasikan kemungkinan berbagai komponen mekanis yang mengalami kerusakan atau keausan. Secara keseluruhan, motor baru tampak memiliki operasi yang lebih stabil dengan satu frekuensi dominan yang jelas menandakan kinerja mekanis yang baik meskipun dengan beban tambahan. Sedangkan, motor rusak menunjukkan spektrum vibrasi yang lebih kompleks dan bervariatif yang mengindikasikan berbagai masalah mekanis yang mungkin terjadi pada motor tersebut. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.38.



Gambar 4.38. Perbandingan spektrum vibrasi

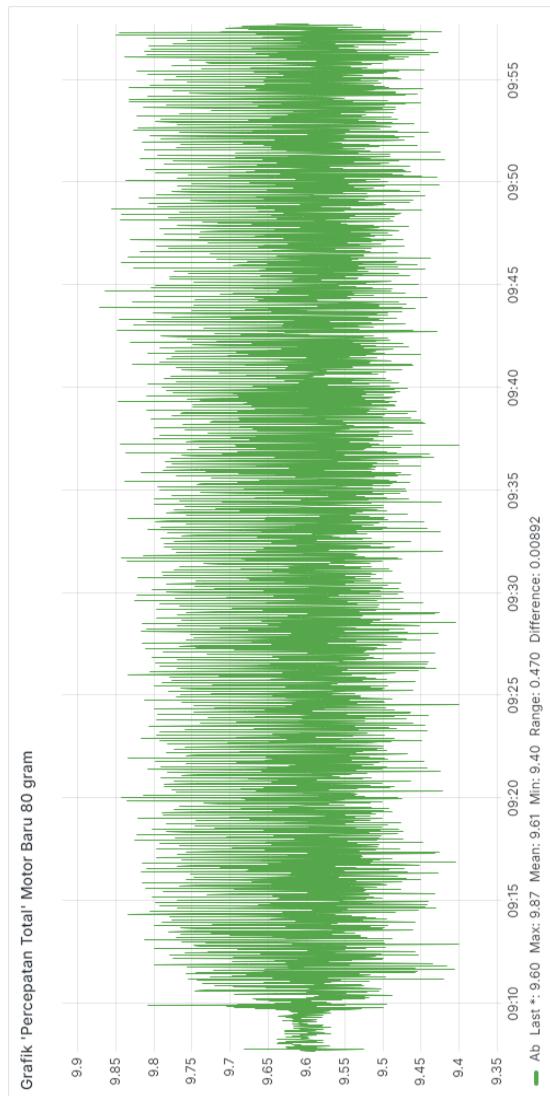
#### 4.4.5. Beban 80 Gram

Gambar 4.39 menunjukkan hasil data vibrasi motor baru dengan beban 80 gram. Dimana nilai maksimum sumbu X sebesar 1,13 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,360 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,84 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -0,0900 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,690 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,36 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,495 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,314 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,58 ( $\text{m/s}^2$ ).



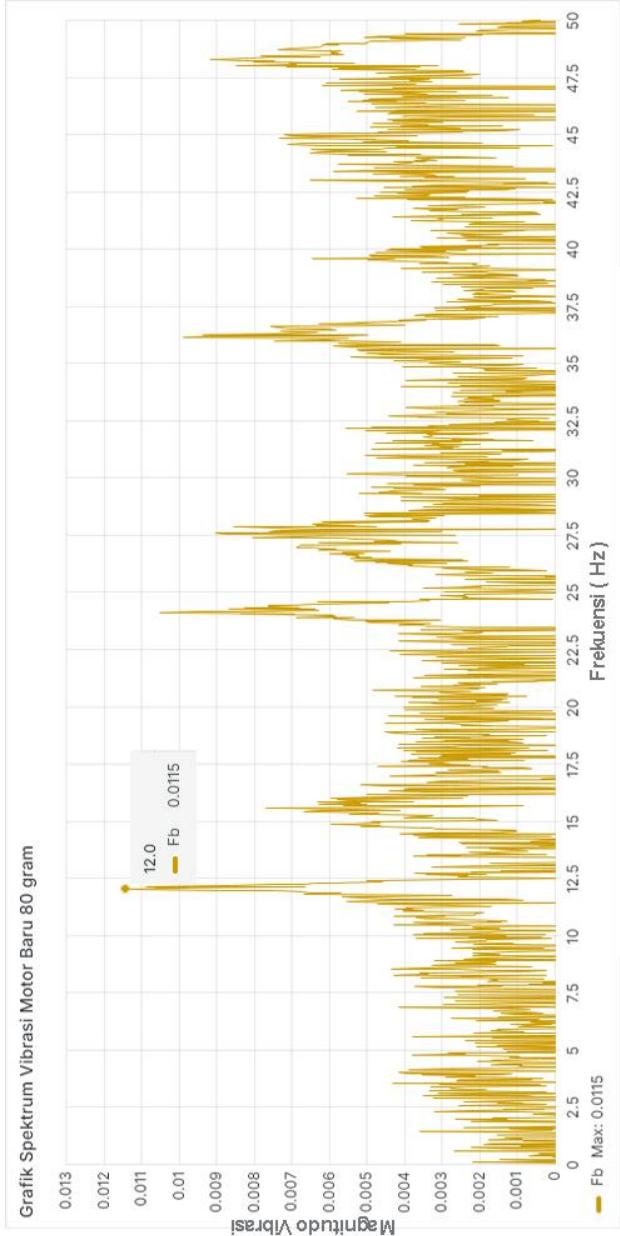
**Gambar 4.39.** Data vibrasi motor baru 80 gram

Gambar 4.40 menunjukan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



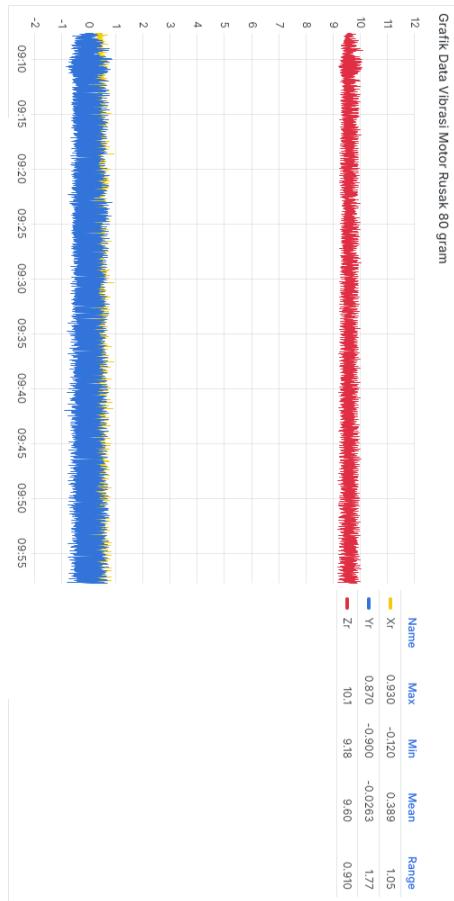
**Gambar 4.40.** Percepatan total vibrasi motor baru 80 gram

Grafik spektrum vibrasi pada Gambar 4.41 motor DC baru dengan beban 80 gram ditandai dengan puncak magnitudetertinggi sekitar 0,0115 m/s<sup>2</sup> pada frekuensi sekitar 12,0 Hz. Puncak magnitudee terjadi pada frekeunsi yang lebih rendah dibandingkan dengan pengujian dengan beban lainnya kecuali beban 40 gram. Bentuk spketrum pada motor baru tampak memiliki puncak yang lebih terfokus yang dapat menunjukkan kondisi operasional yang lebih konsisten atau seimbang pada frekuensi tertentu. Terdapat beberapa puncak lainnya tetapi tidak sebesar puncak utama pada 12,0 Hz. Gambar 4.41 grafik sumbu x menunjukan magnitudo vibrasi dan sumbu y menunjukan magnitude frekuensi.



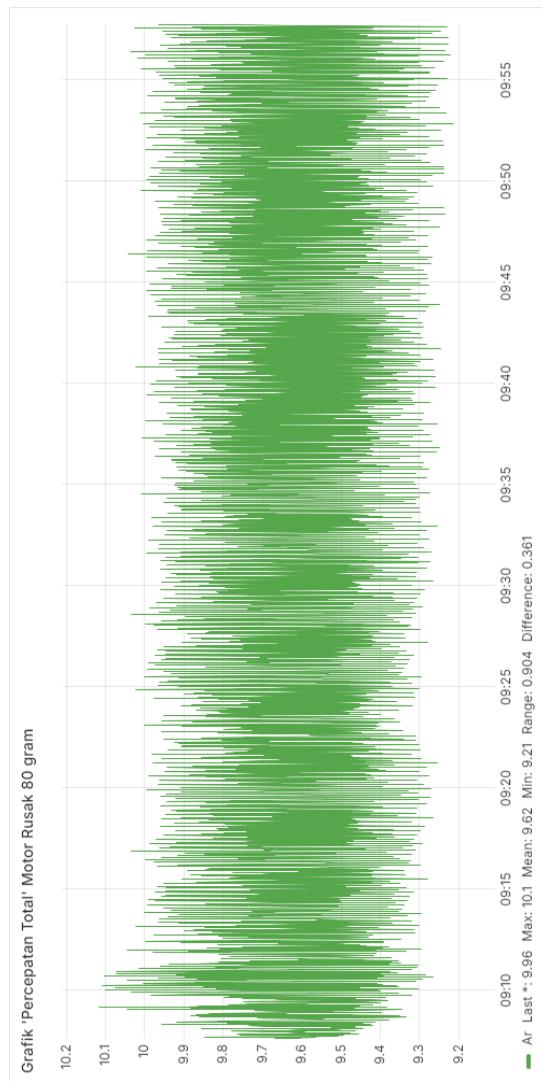
**Gambar 4.41.** Spektrum vibrasi motor DC baru 80 gram

Gambar 4.42 menunjukkan hasil data vibrasi motor rusak dengan beban 80 gram. Dimana nilai maksimum sumbu X sebesar 0,930 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 0,870 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 10,1 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -0,120 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,900 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,18 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,389 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,0263 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,18 ( $\text{m/s}^2$ ).



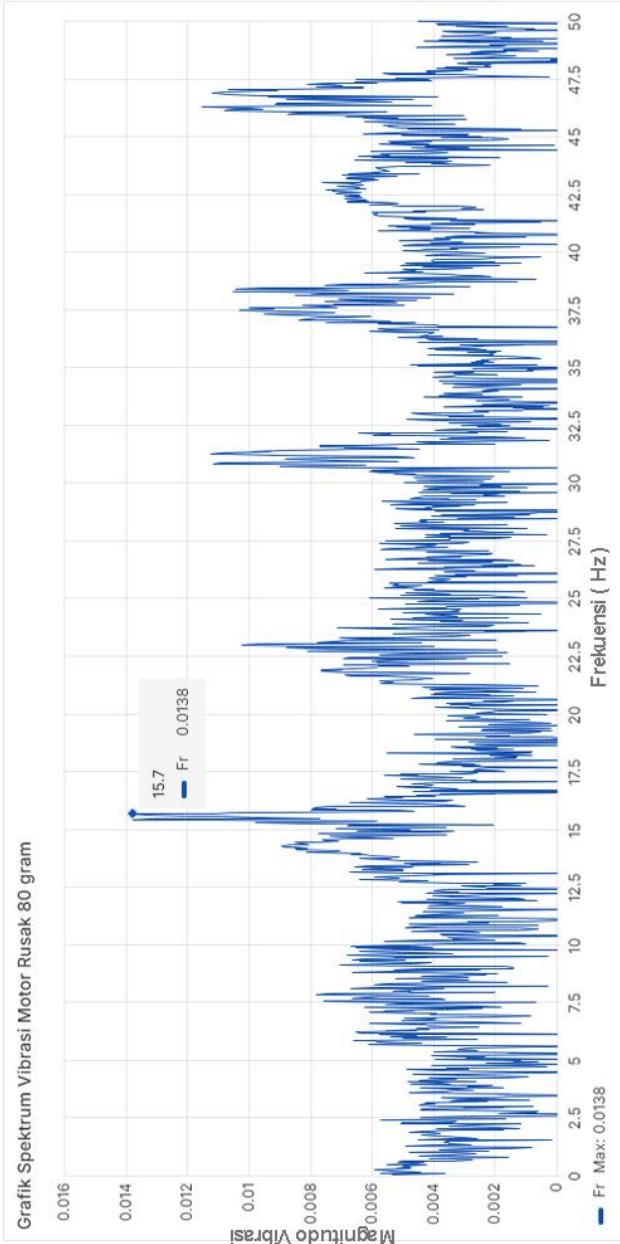
**Gambar 4.42.** Data vibrasi motor rusak 80 gram

Gambar 4.43 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



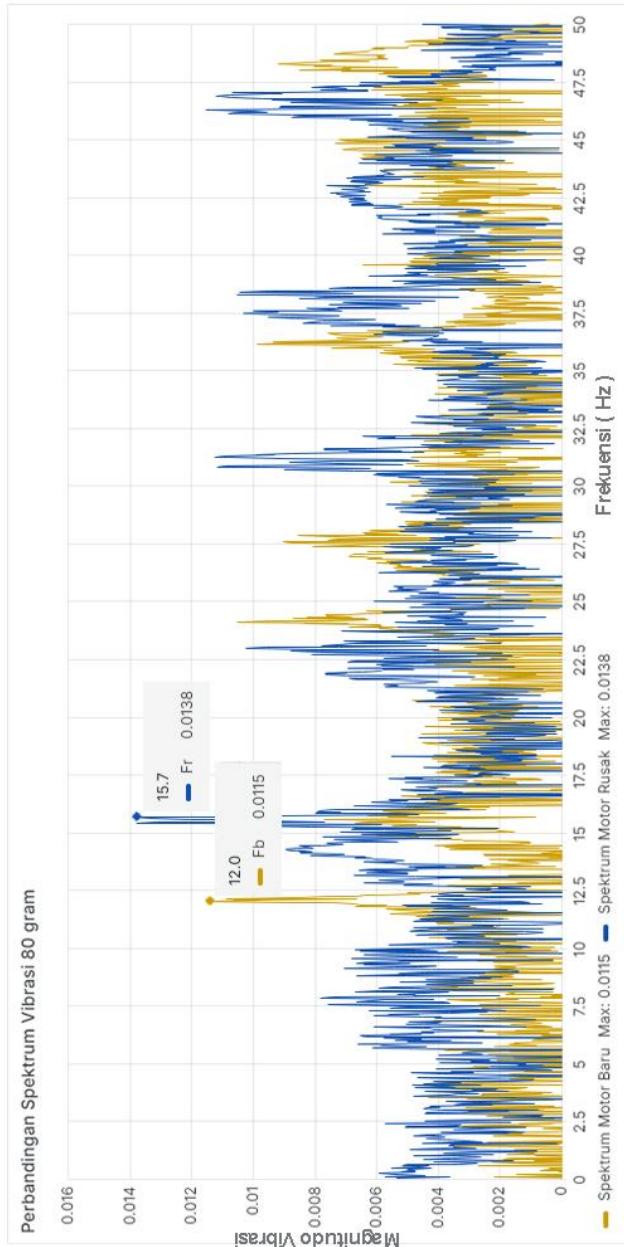
**Gambar 4.43.** Percepatan total vibrasi motor rusak 80 gram

Grafik spektrum vibrasi pada Gambar 4.44 motor DC rusak dengan beban 80 gram ditandai dengan puncak magnitude tertinggi sekitar 0,0138 m/s<sup>2</sup> pada frekuensi sekitar 15,7 Hz. Spektrum menunjukkan sebaran vibrasi yang lebih beragam dengan banyak puncak magnitude yang lebih kecil, menunjukkan bahwa ada lebih banyak komponen yang berkontribusi terhadap vibrasi motor yang rusak. Gambar 4.44 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitudo frekuensi.



**Gambar 4.44.** Spektrum vibrasi motor DC rusak 80 gram

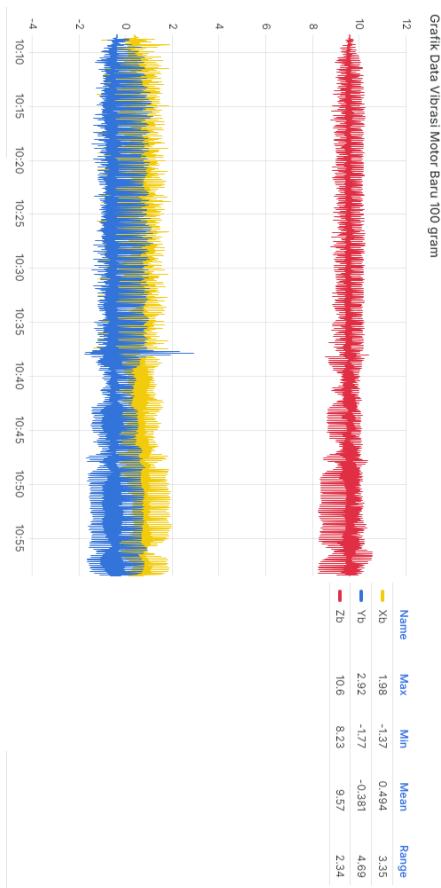
Motor baru menunjukkan puncak magnitudo yang lebih tinggi pada frekuensi yang lebih rendah dibandingkan dengan motor rusak. Ini bisa menunjukkan adanya respon yang lebih besar terhadap beban dinamis pada frekuensi tersebut atau kondisi operasi yang lebih stabil. Terdapat pergeseran dalam frekuensi puncak magnitudo antara motor baru dan motor rusak yang menunjukkan perubahan dalam kondisi dinamis atau mungkin kerusakan internal motor. Motor baru menunjukkan puncak yang lebih dominan pada frekuensi tertentu, sedangkan motor rusak memiliki sebaran puncak yang lebih merata dan lebih luas, menunjukkan potensi kerusakan mekanis yang lebih kompleks dan bervariatif. Motor baru dan motor rusak memiliki perbedaan yang signifikan dalam karakteristik vibrasi ketika diberi beban yang sama. Motor rusak menunjukkan sebaran vibrasi yang lebih kompleks dan yang mengindikasikan berbagai masalah mekanis yang mungkin terjadi pada motor tersebut. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.45.



**Gambar 4.45.** Perbandingan spektrum vibrasi

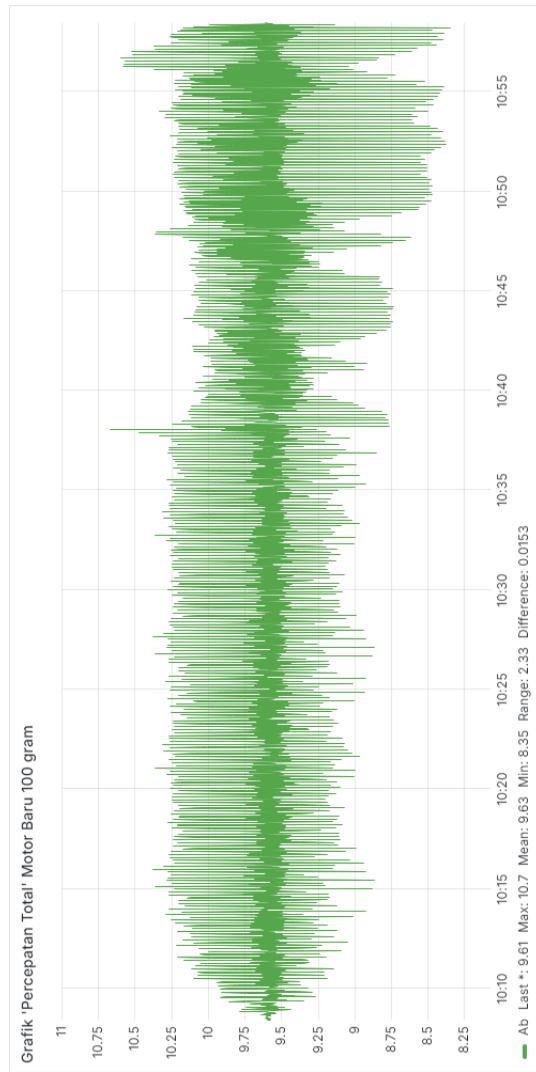
#### 4.4.6. Beban 100 Gram

Gambar 4.46 menunjukkan hasil data vibrasi motor baru dengan beban 100 gram. Dimana nilai maksimum sumbu X sebesar 1,98 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 2,92 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 10,6 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -1,37 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -1,77 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 8,23 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,494 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,381 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,57 ( $\text{m/s}^2$ ).



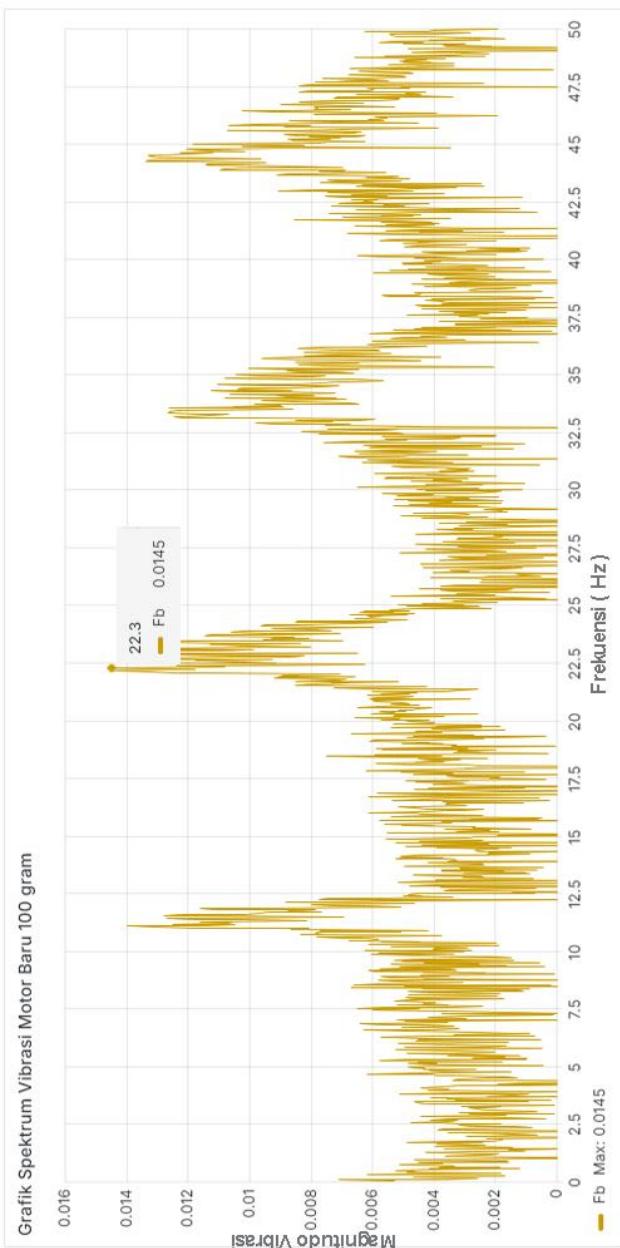
**Gambar 4.46.** Data vibrasi motor baru 100 gram

Gambar 4.47 menunjukan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



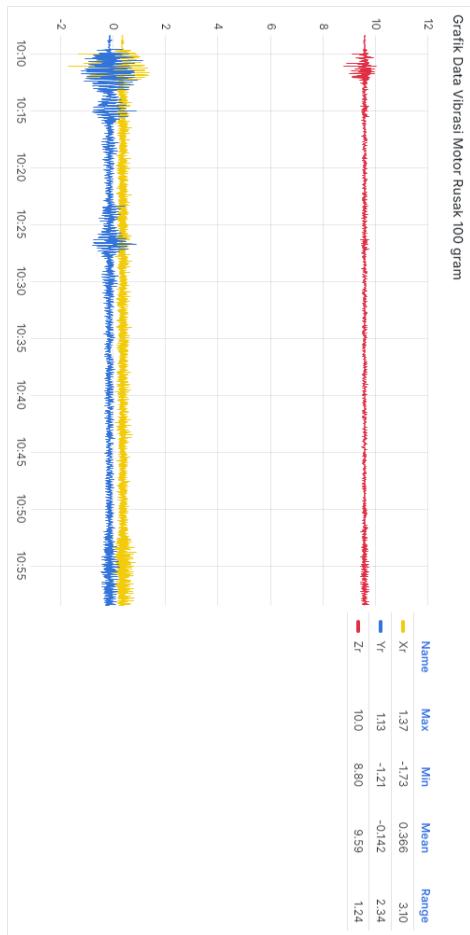
**Gambar 4.47.** Percepatan total vibrasi motor baru 100 gram

Grafik spektrum vibrasi pada Gambar 4.48 motor DC baru dengan beban 100 gram ditandai dengan puncak magnitude tertinggi sekitar 0,0145 m/s<sup>2</sup> pada frekuensi sekitar 22,3 Hz. Hal ini menunjukkan bahwa motor baru memiliki respon vibrasi terbesar pada frekuensi ini dengan beban 100 gram. Spektrum menunjukkan sebaran vibrasi yang cukup merata di sekitar puncak yang ditentukan yang dapat menunjukkan kondisi operasional yang lebih konsisten atau seimbang pada frekuensi tertentu. Gambar 4.48 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



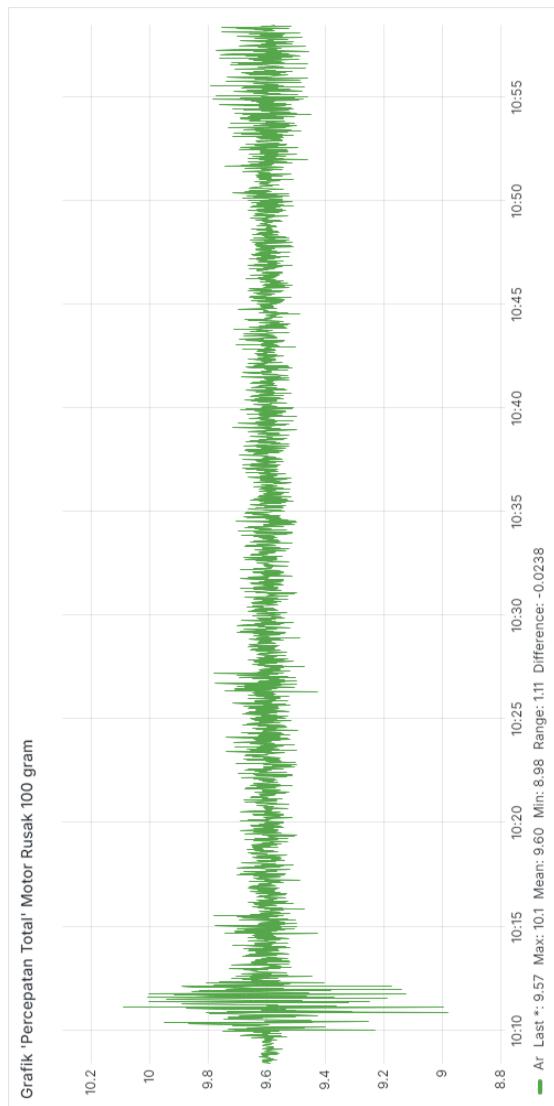
Gambar 4.48. Spektrum vibrasi motor DC baru 100 gram

Gambar 4.49 menunjukkan hasil data vibrasi motor rusak dengan beban 100 gram. Dimana nilai maksimum sumbu X sebesar 1,37 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar 1,13 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 10,0 ( $\text{m/s}^2$ ). Nilai minimum sumbu X sebesar -1,73 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -1,21 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 8,80 ( $\text{m/s}^2$ ). Dengan rata-rata nilai sumbu X sebesar 0,366 ( $\text{m/s}^2$ ), kemudian sumbu Y sebesar -0,142 ( $\text{m/s}^2$ ) dan sumbu Z sebesar 9,59 ( $\text{m/s}^2$ ).



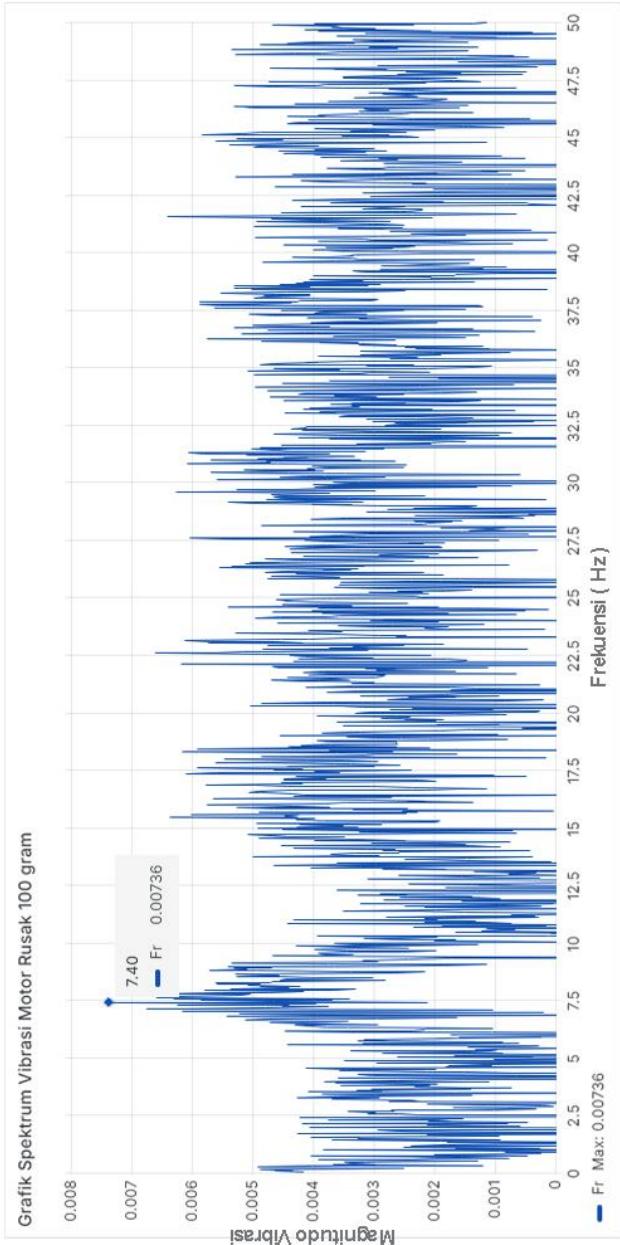
**Gambar 4.49.** Data vibrasi motor rusak 100 gram

Gambar 4.50 menunjukkan hasil dari perhitungan percepatan total sumbu-sumbu data vibrasi yang akan disimpan dan dilakukan perhitungan untuk mengubah menjadi spektrum vibrasi.



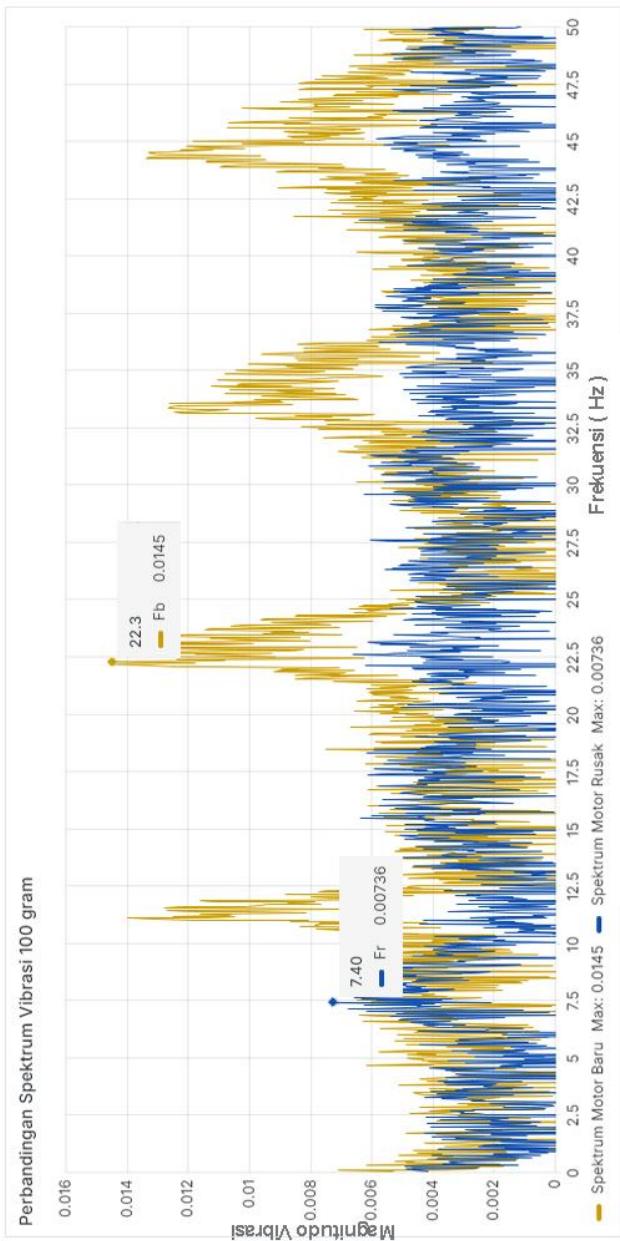
**Gambar 4.50.** Percepatan total vibrasi motor rusak 100 gram

Grafik spektrum vibrasi pada Gambar 4.51 motor DC rusak dengan beban 100 gram ditandai dengan puncak magnitude tertinggi sekitar 0,00736 m/s<sup>2</sup> pada frekuensi sekitar 7,4 Hz. Magnitude ini lebih rendah dibandingkan dengan motor baru yang dapat menunjukkan bahwa motor rusak menghasilkan vibrasi yang lebih rendah pada frekuensi yang berbeda. Puncak pada 7,4 Hz mungkin mengindikasikan adanya masalah mekanis yang berbeda yang mempengaruhi frekuensi resonansi motor atau mengubah karakteristik operasionalnya. Spekturm motor rusak menunjukkan sebaran puncak yang lebih seragam yang menandakan adanya gangguan mekanis pada kondisi motor. Gambar 4.51 grafik sumbu x menunjukkan magnitudo vibrasi dan sumbu y menunjukkan magnitude frekuensi.



Gambar 4.51. Spektrum vibrasi motor DC rusak 100 gram

Motor baru menunjukkan puncak magnitudo yang lebih tinggi, menandakan respons vibrasi yang lebih besar pada frekuensi tertentu, sedangkan motor rusak memiliki magnitudo yang lebih rendah. Puncak frekuensi pada kedua motor sangat berbeda, yang menunjukkan perbedaan dalam cara motor menanggapi beban. Motor baru memiliki distribusi yang lebih terfokus pada satu frekuensi, sedangkan motor rusak menunjukkan variasi yang lebih luas dengan banyak puncak yang menunjukkan potensi kerusakan atau gangguan mekanis yang lebih beragam. Beban tambahan memiliki dampak yang signifikan pada kedua motor, namun motor yang rusak mungkin memiliki masalah yang lebih parah yang mempengaruhi kinerjanya pada berbagai frekuensi, tidak hanya pada frekuensi dominan. Sedangkan, motor baru mempertahankan karakteristik spektrum vibrasi yang lebih konsisten meskipun dengan beban yang ditingkatkan. Bentuk perbandingan spektrum vibrasi motor baru dan motor rusak ditunjukkan pada Gambar 4.52.



Gambar 4.52. Perbandingan spektrum vibrasi

#### 4.5. Pembahasan

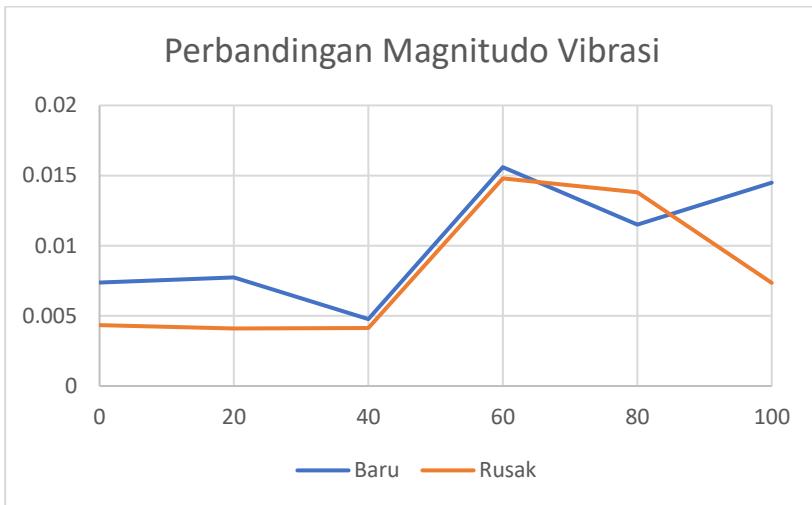
Data yang digunakan berasal dari dua motor DC yang diuji, Motor DC Baru dan Motor DC Rusak. Analisis vibrasi digunakan untuk mengevaluasi kondisi mesin berputar. Frekuensi vibrasi dan magnitudenya adalah dua parameter penting yang biasanya diukur. Magnitude vibrasi, biasanya diukur dalam meter per detik kuadrat ( $m/s^2$ ), mengindikasikan besar gerakan vibrasi yang dihasilkan oleh mesin, sedangkan frekuensi yang diukur dalam Hertz (Hz), mengidentifikasi berapa kali mesin bergetar dalam satu detik.

**Tabel 4.1.** Tabel perbandingan data motor baru dan motor rusak

Beban ( Gram )	Motor DC Baru		Motor DC Rusak	
	Frekuensi	Magnitudo	Frekuensi	Magnitudo
0	39,3	0,00738	33,0	0,00435
20	41,8	0,00773	23,6	0,00410
40	38,9	0,00477	32,0	0,00414
60	19,6	0,0156	14,8	0,0148
80	12,0	0,0115	15,7	0,0138
100	22,3	0,0145	7,4	0,00736

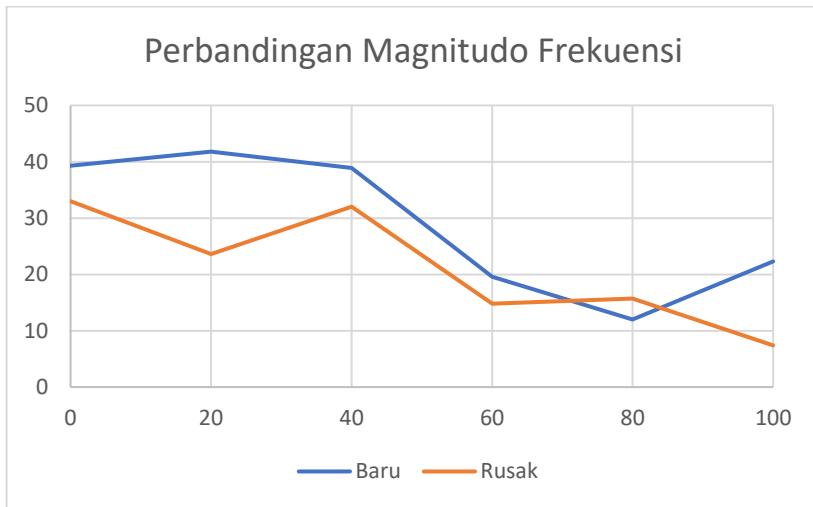
Analisis vibrasi pada tabel menunjukkan bahwa ada perbedaan signifikan antara motor DC baru dan motor DC rusak dalam hal frekuensi dan magnitudo pada berbagai beban. Magnitudo motor DC baru berkisar antara 0,00738 hingga 0,0156. Magnitudo tertinggi terjadi pada beban 60 gram (0,0156), menunjukkan puncak vibrasi yang signifikan pada beban ini. Magnitudo motor DC rusak berkisar antara 0,00435 hingga 0,0148. Magnitudo tertinggi terjadi pada beban 60 gram (0,0148), yang menunjukkan vibrasi yang lebih rendah dibandingkan motor baru pada beban tertinggi.

Frekuensi motor DC baru berkisar antara 12,0 Hz hingga 41,8 Hz. Terlihat bahwa frekuensi meningkat secara signifikan pada beban 20 gram (41,8 Hz) dan 100 gram (22,3 Hz), dengan nilai frekuensi terendah pada beban 80 gram (12,0 Hz). Frekuensi motor DC rusak berkisar antara 7,4 Hz hingga 33,0 Hz. Frekuensi tertinggi terjadi pada beban 40 gram (32,0 Hz) dan terendah pada beban 100 gram (7,4 Hz).



**Gambar 4.53.** Perbandingan magnitudo vibrasi motor DC

Gambar 4.51 menjelaskan magnitudo vibrasi motor Dc baru meningkat secara signifikan pada beban 60 gram di kedua kondisi menunjukkan bahwa titik ini mungkin merupakan beban kritis dimana terjadi perubahan signifikan dalam perilaku getaran, tetapi kemudian menurun dan relatif stabil pada beban yang lebih tinggi. Peningkatan tajam ini mungkin menunjukkan resonansi dan bisa disebabkan oleh faktor seperti ketidakstabilan dalam suplai daya, kondisi lingkungan, atau kesalahan pembacaan sensor atau kondisi khusus lain pada beban tertentu. Fluktuasi yang lebih besar pada kondisi motor baru dapat mengindikasikan bahwa komponen yang masih motor baru lebih responsif terhadap perubahan beban dibandingkan komponen yang motor rusak. Magnitudo vibrasi motor rusak menunjukkan magnitudo vibrasi yang lebih rendah pada beban ringan namun mengalami peningkatan yang lebih terkontrol setelah beban kritis. Magnitudo motor baru menunjukkan magnitudo vibrasi yang lebih tinggi pada beban kritis dan lebih bervariasi, hal ini menunjukkan bahwa motor baru memiliki elastisitas atau respons yang lebih besar terhadap beban yang diterapkan.



**Gambar 4.54.** Perbandingan Frekuensi motor DC

Gambar 4.52 motor baru memiliki fluktuasi magnitude frekuensi bertambahnya beban, tetapi secara umum magnitude frekuensi lebih tinggi daripada motor rusak. Ini mengindikasikan bahwa motor baru mungkin memiliki ketahanan yang lebih baik terhadap peningkatan beban. Pada motor rusak menunjukkan penurunan yang signifikan dalam frekuensi saat beban meningkat, khususnya pada beban 100 gram di mana frekuensi turun drastis ke 7,5 Hz. Ini menandakan kemungkinan adanya masalah dalam motor, seperti kehilangan performa atau masalah pada kondisi fisik yang mengakibatkan kinerja yang tidak efisien.

Motor baru memiliki kinerja yang lebih stabil dan dapat diandalkan daripada motor rusak, yang menunjukkan tanda-tanda kerusakan yang signifikan dan kehilangan efisiensi mekanis. Untuk memprediksi kinerja masa depan, pemantauan terus-menerus pada frekuensi dan magnitude vibrasi sangat penting, terutama di beban yang lebih tinggi yang menunjukkan tanda-tanda kerusakan yang lebih besar pada motor rusak. Jika magnitude terus meningkat dengan penambahan beban dan disertai dengan penurunan frekuensi yang tidak biasa, ini bisa mengindikasikan bahwa motor mengalami kerusakan yang memburuk.

*Halaman ini sengaja dikosongkan*

*Halaman ini sengaja dikosongkan*

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah dilakukannya percobaan di atas didapatkan kesimpulan sebagai berikut :

1. Telah dibagun sebuah system monitoring *Internet of Things* yang bekerja secara baik dan efisien. Berdasarkan pengujian data yang dikirimkan berupa data vibrasi sesuai dengan yang diinginkan dan dapat ditampilkan sesuai dengan data yang didapatkan cukup akurat.
2. Data vibrasi motor baru memiliki kinerja yang lebih stabil dan dapat diandalkan daripada motor rusak, yang menunjukkan tanda-tanda kerusakan yang signifikan dan kehilangan efisiensi mekanis dengan bentuk grafik puncak magnitude yang tidak beraturan.
3. Sistem *internet of things* ini dapat ditampilkan sesuai dengan informasi yang dibutuhkan dan dapat terihat langsung ketika data mengalami perubahan. Kombinasi berbagai perangkat menjadi sebuah solusi dalam monitoring suatu system berupa kondisi, data, dan pengukuran yang ditampilkan serta melakukan analisis data secara cepat. Sistem ini diharapkan menjadi sistem yang efektif dan dapat membantu dalam memantau kinerja motor Dc secara langsung dan mendeteksi kerusakan dengan cepat. Sistem ini dapat membantu dalam mencegah kerusakan yang lebih parah dan meningkatkan efisiensi operasi.

#### 5.2. Saran

Saran pada penelitian ini adalah sistem monitoring *Internet of Things* perlu dibuat yang lebih efisien dengan berbagai perangkat yang ada. Serta disarankan untuk meningkatkan teknik analisis data vibrasi yang lebih efektif. Sehingga dapat mencegah kerusakan suatu mesin dengan waktu yang tepat, cepat, dan mampu mengurangi atau mengetahui biaya dan waktu sebelum mesin benar-benar rusak.



*Halaman ini sengaja dikosongkan*

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- Adams, M. L. (2000). *Rotating machinery vibration: from analysis to troubleshooting*. CRC Press.
- Diono, M., Azwar, H., & Khabzli, W. (2021). Sistem Monitoring Jaringan Sensor Node Berbasis Protokol MQTT. *Jurnal Elementer*, 7(2), 120–126.
- Ferencz, K., & Domokos, J. (2020). *Using Node-RED platform in an industrial environment*.
- Gurocak, H. (2015). *Industrial motion control: motor selection, drives, controller tuning, applications*. John Wiley & Sons.
- Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008). MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, 791–798. <https://doi.org/10.1109/COMSWA.2008.4554519>
- Karmakar, A., Dey, N., Baral, T., Chowdhury, M., & Rehan, M. (2019). *Industrial Internet of Things: A Review*. <https://doi.org/10.1109/OPTRONIX.2019.8862436>
- Kim, S.-H. (2017). *Electric motor control: DC, AC, and BLDC motors*. Elsevier.
- Michon, R., Overholt, D., Letz, S., Orlarey, Y., Fober, D., Dumitrescu, C., Faust, al A., & Letz, S. (2020). *A Faust Architecture for the ESP32 Microcontroller*.
- Mobley, R. K. (1999). *Vibration fundamentals*. Elsevier.
- MPU-6000 and MPU-6050 Product Specification Revision 3.4*
- MPU-6000/MPU-6050 Product Specification*. (2013).
- Noprianto, N., Wijayaningrum, V. N., & Wakhidah, R. (2023). Monitoring Development Board based on InfluxDB and Grafana. *Telematika*, 20, 81. <https://doi.org/10.31315/telematika.v20i1.7643>
- Noviello, C. (2015). *Mastering STM32 A step-by-step guide to the most complete ARM Cortex-M platform, using a free and powerful development environment based on Eclipse and GCC*. <http://leanpub.com/mastering-stm32>
- Rahman, M. (2011). *Applications of Fourier transforms to generalized functions*. WIT press.

- Rao, S. S. (2012). *Mechanical vibrations fifth edition in SI units*.  
PEARSON, USA.
- Scheffer, C., & Girdhar, P. (2004). *Practical machinery vibration analysis and predictive maintenance*. Elsevier.
- Serpanos, D., & Wolf, M. (2017). *Internet-of-things (IoT) systems: architectures, algorithms, methodologies*. Springer.
- Solichin, A. (2005). *Pemrograman Web dengan PHP dan MySQL*.  
Achmad Solichin.  
<https://books.google.co.id/books?id=kcD4BQAAQBAJ>

*Halaman ini sengaja dikosongkan*

*Halaman ini sengaja dikosongkan*

## Lampiran 1 Data Penelitian

Data Motor Baru

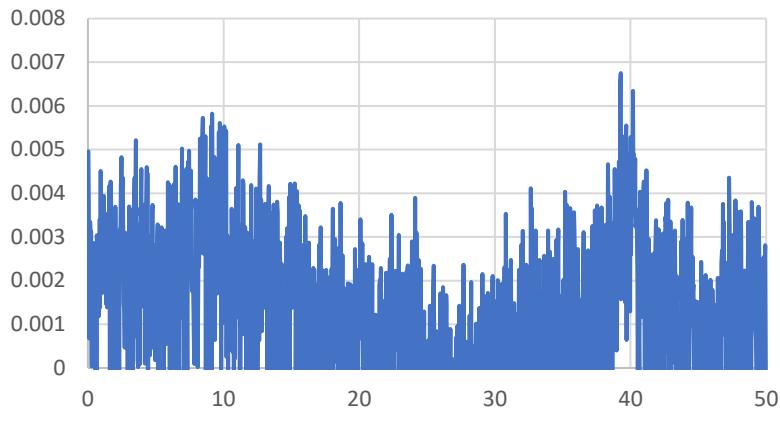
0 Gram	20 Gram	40 Gram	60 Gram	80 Gram	100 Gram
0,0297059	0,02970694	0,02970377	0,0297253	0,0297134	0,029720112
0,0049585	-0,0021032	-0,0020851	0,0033196	0,00372237	0,005224257
0,0036745	-0,0004708	-0,0001481	0,00860414	-0,000971	0,005296885
0,0006815	0,00221673	-0,0007622	0,0068542	0,00218523	0,007100423
0,0006971	0,00113411	0,00318854	0,00333685	-0,0040717	0,003747297
0,003342	0,00606811	0,00048315	0,00262552	0,00033637	0,002574628
0,0029856	0,00358055	-0,0001714	0,00592799	0,00146776	0,00467819
0,0031216	0,0023384	0,00080968	0,00531464	-0,001435	0,004432495
3,456E-05	0,00194744	0,00103444	0,00631055	-0,0002253	0,001698568
0,0025745	0,00049668	-0,0001021	0,00442853	-0,002444	0,004937645
0,0017953	0,00213772	0,00055041	-0,0010526	7,7063E-05	0,006184551
0,0002677	-0,0026759	0,00120306	0,00532845	-0,001283	0,003572202

Data Motor Rusak

0 Gram	20 Gram	40 Gram	60 Gram	80 Gram	100 Gram
0,0297594	0,02976946	0,02976845	0,02971543	0,02971758	0,029711985
0,0002987	-0,0006889	0,00132101	0,00105359	0,00479305	0,005313206
0,0011278	-0,0014126	0,00058246	-0,0020687	0,00535961	0,004153593
-0,000347	-0,0035552	0,00231499	-0,0002494	0,00362876	0,004622977
0,0009187	0,00018664	-0,0026936	-0,0005172	0,00571129	0,004348315
-0,000659	-0,002469	-0,0001561	0,00189379	0,00443694	0,004879255
-0,001193	-0,001485	0,00072924	0,00123416	0,00539697	0,002501938
-0,002168	-0,000406	0,00283578	0,00292209	0,00535253	0,003378138
-3,78E-05	-0,0041167	-0,0042503	0,00116935	0,00594005	0,004910963
0,0017382	0,00158131	-0,0030669	0,00118717	0,00476025	0,004013418
0,0013655	-0,0029892	0,00082703	0,00127968	0,00518209	0,001200839
0,0003215	8,9946E-05	0,00101875	0,0008834	0,00495577	0,003667413

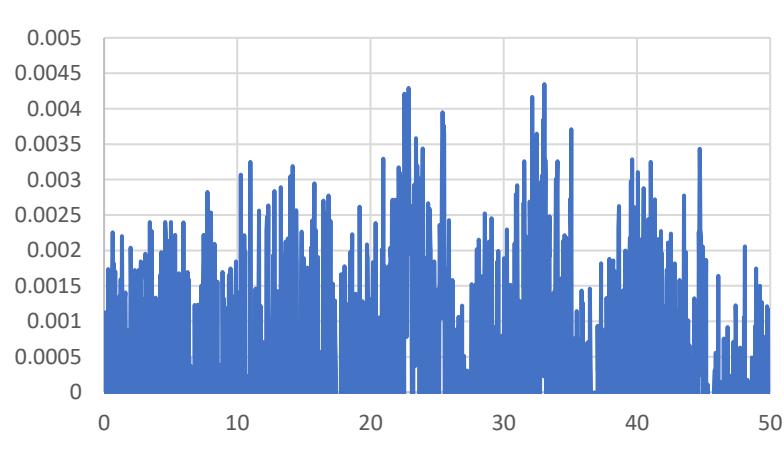
### 1. Data 0 Gram Motor Baru

0 Gram Motor Baru

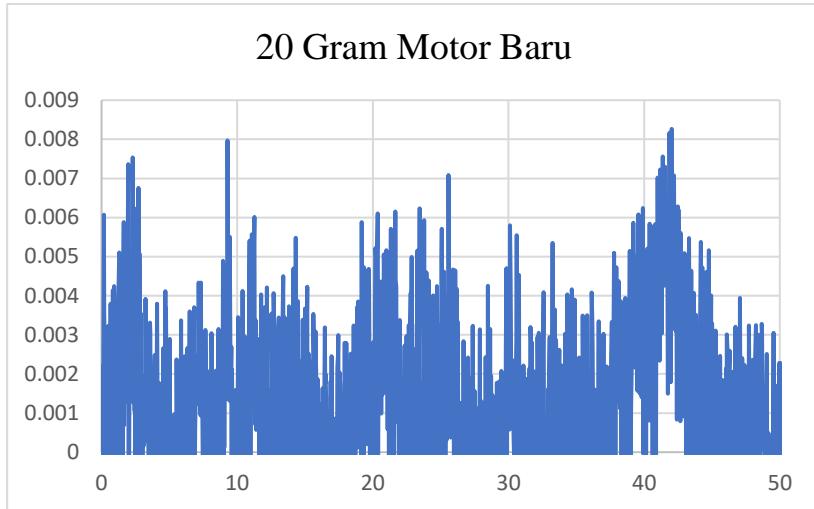


### 2. Data 0 Gram Motor Rusak

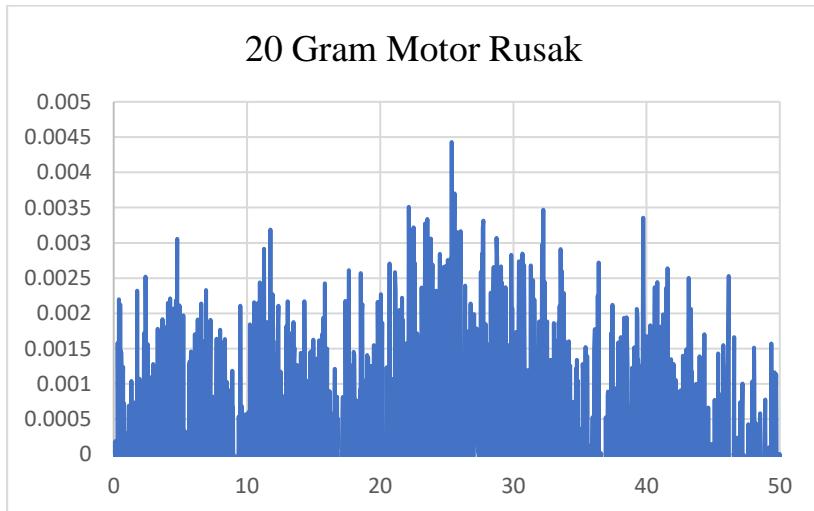
0 Gram Motor Rusak



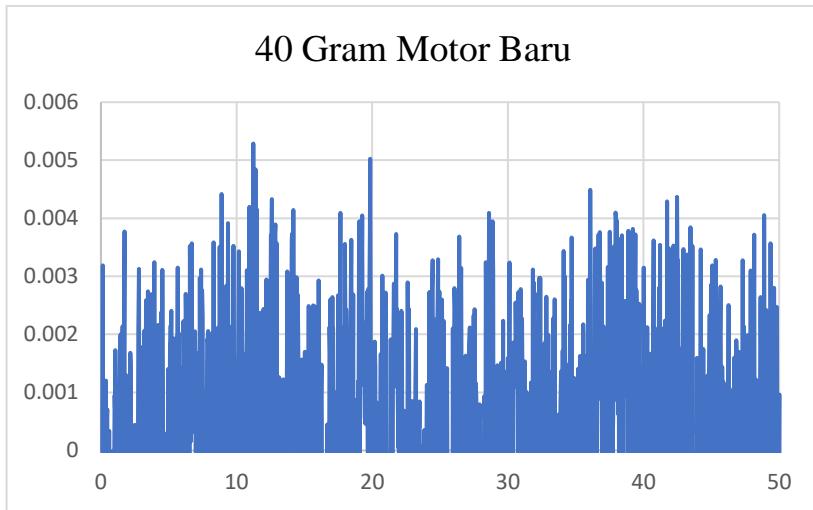
### 3. Data 20 Gram Motor Baru



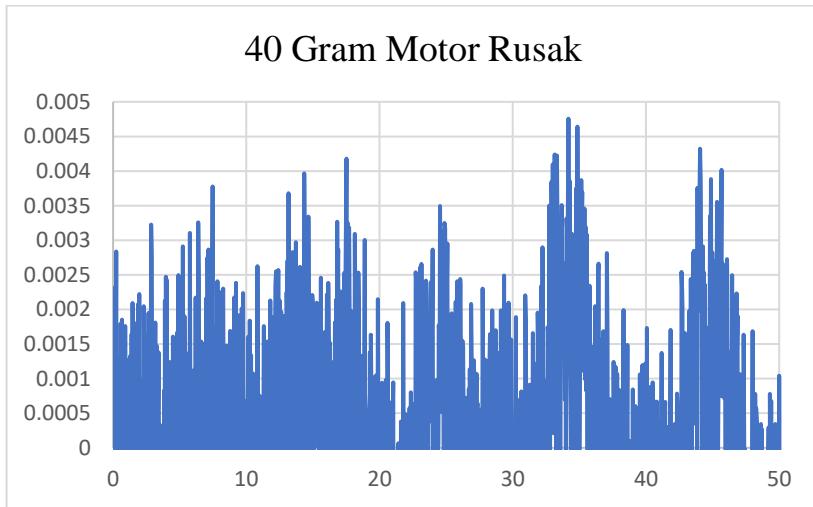
### 4. Data 20 Gram Motor Rusak



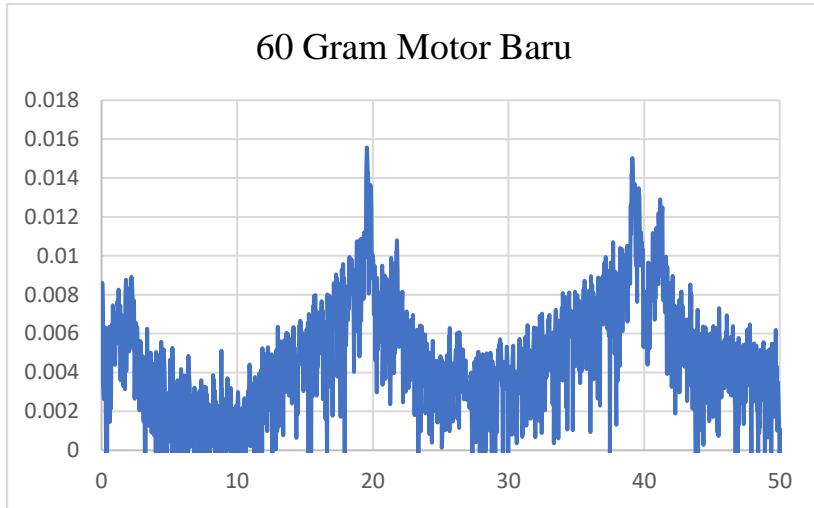
## 5. Data 40 Gram Motor Baru



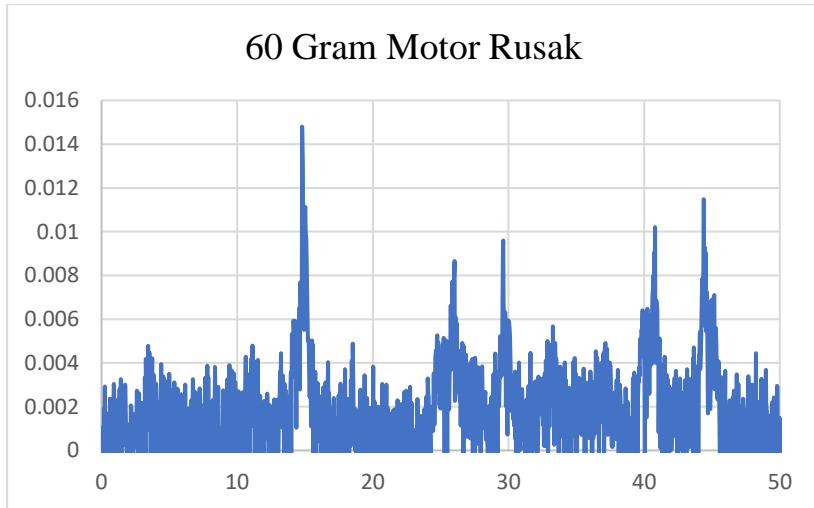
## 6. Data 40 Gram Motor Rusak



7. Data 60 Gram Motor Baru

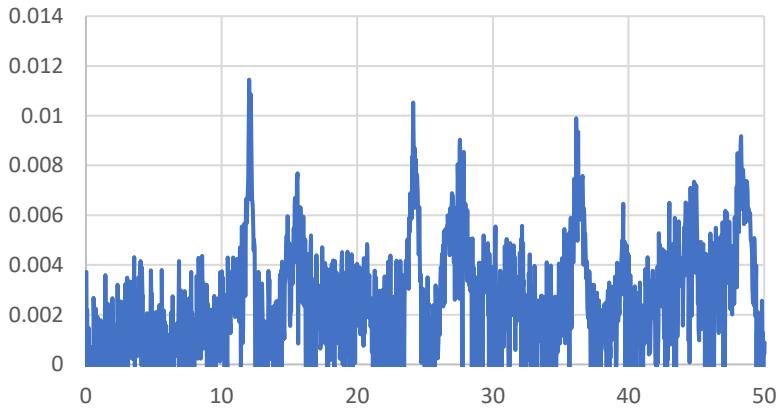


8. Data 60 Gram Motor Rusak



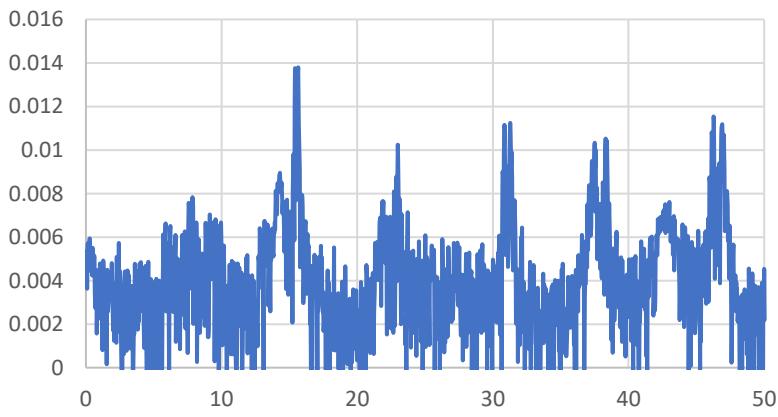
9. Data 80 Gram Motor Baru

80 Gram Motor Baru

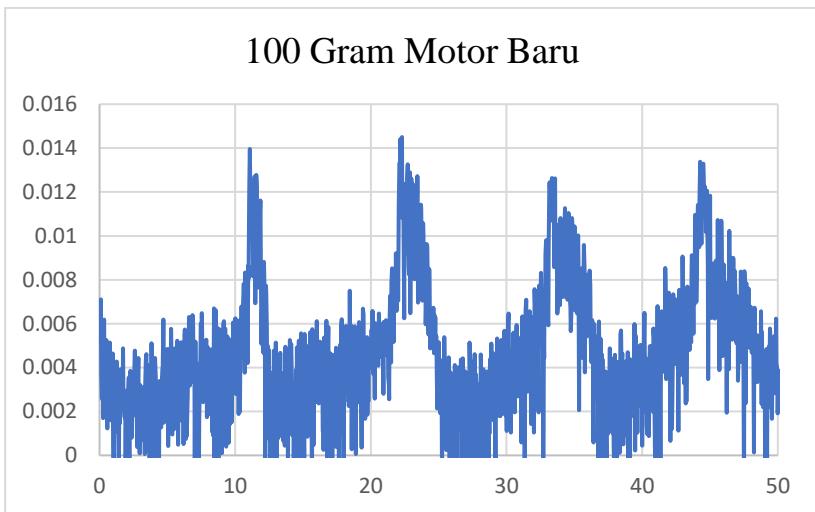


10. Data 80 Gram Motor Rusak

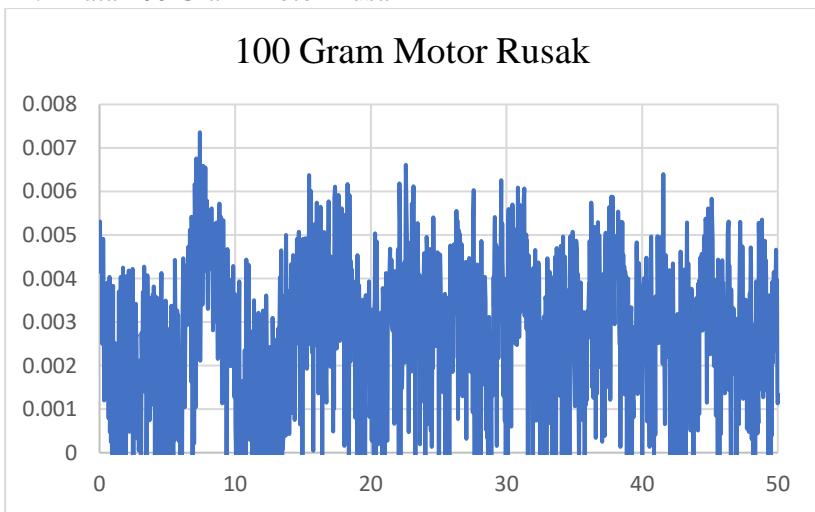
80 Gram Motor Rusak



11. Data 100 Gram Motor Baru



12. Data 100 Gram Motor Rusak



## Lampiran 2 Foto Dokumentasi

