



TUGAS AKHIR - TE 1415199

**SISTEM PENDETEKSI KEPATAHAN MATA BOR PADA
MESIN CETAK PCB BERDASARKAN ANALISA GETARAN
SPINDLE MENGGUNAKAN TEENSY BOARD**

Putra Trimardian Asri
NRP 0711134000016

Dosen Pembimbing
1. Dr. Muhammad Rivai, ST., MT.
2. Ir. Tasripan, MT

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 1415199

**TOOL BREAKAGE DETECTION SYSTEM IN PCB PRINTER
BASED ON SPINDLE VIBRATION ANALYSIS USING
TEENSY BOARD**

Putra Trimardian Asri
NRP 0711134000016

Advisor

1. Dr. Muhammad Rivai, ST., MT.
2. Ir. Tasripan, MT

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institute Technology of Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa sebagian maupun keseluruhan isi tugas akhir saya yang berjudul “Sistem Pendeteksi Kepatahan Mata Bor pada Mesin Cetak PCB berdasarkan Analisa Getaran Spindle menggunakan Teensy Board” merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya mandiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2018

Putra Trimardian Asri
NRP. 0711134000016

Halaman ini sengaja dikosongkan

**Sistem Pendeteksi Kepatahan Mata Bor pada Mesin
Cetak PCB berdasarkan Analisa Getaran Spindle
menggunakan Teensy Board**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I,



Dr. Muhammad Rivai, ST., MT.

NIP. 196904261994031003

Dosen Pembimbing II,



Ir. Tasripan, MT.

NIP. 196204181990031004



Halaman ini sengaja dikosongkan

SISTEM PENDETEKSI KEPATAHAN MATA BOR PADA MESIN CETAK PCB BERDASARKAN ANALISA GETARAN SPINDLE MENGGUNAKAN TEENSY BOARD

Nama : Putra Trimardian Asri
Pembimbing : 1. Dr. Muhammad Rivai, ST., MT.
2. Ir. Tasripan, MT.

ABSTRAK

Pemanfaatan *Computer Numerical Control* (CNC) adalah salah satu penerapan teknologi industri yang membuat hasil produksi lebih presisi dan akurat. CNC juga dapat diterapkan pada proses pencetakan *Printed Circuit Board* (PCB) untuk menggantikan proses pelarutan secara kimiawi yang tidak ramah lingkungan. Akan tetapi pencetak PCB berbasis CNC memiliki risiko yakni patahnya mata bor. Pada penelitian ini telah dibuat suatu sistem pendeteksian kepatahan mata bor berdasarkan analisa getaran. Sensor yang digunakan untuk dapat mendeteksi getaran adalah MEMS *accelerometer* yang mempunyai kemampuan pengukuran sampai 3,6g. Untuk mengenali kepatahan mata bor perlu dilakukan pengolahan sinyal dengan menggunakan *Fast Fourier Transform*. Lebar spektrum frekuensi yang digunakan adalah 0-1000Hz. Pola spektrum frekuensi antara 0-1000Hz digunakan sebagai *input Artificial Neural Network* dengan satu *hidden layer* untuk mengenali kepatahan mata bor. Pemrosesan *Fast Fourier Transform* dan *Artificial Neural Network* dilakukan pada *Teensy 3.2 development board*. Hasil eksperimen dengan kecepatan putaran *spindle* 30000RPM menunjukkan bahwa *Artificial Neural Network* dapat mendeteksi kepatahan mata bor dengan tingkat keberhasilan 80%. Penggunaan jenis PCB yang lebih keras meningkatkan keberhasilan menjadi 91.67%. Diharapkan dengan implementasi sistem ini, penggunaan CNC sebagai pencetak PCB dapat lebih efisien pada konsumsi daya dan waktu.

Kata kunci: *artificial neural network, fast fourier transform, kepatahan mata bor, pencetak PCB*

Halaman ini sengaja dikosongkan

TOOL BREAKAGE DETECTION SYSTEM IN PCB PRINTER BASED ON SPINDLE VIBRATION ANALYSIS USING TEENSY BOARD

Name : Putra Trimardian Asri

Advisor : 1. Dr. Muhammad Rivai, ST., MT.

2. Ir. Tasripan, MT.

ABSTRACT

Utilization of Computer Numerical Control (CNC) is one application of industrial technology that makes the product more precise and accurate. CNC can also be used to fabricate Printed Circuit Board (PCB) as a substitute for chemical dissolution process which is not environmentally friendly. However, CNC-based PCB printers are prone to tool breakage. In this research, tool breakage detection system based on vibration analysis has been created. The sensor used to detect vibration is a MEMS accelerometer that has a measurement capability up to 3,6g. To recognize the tool breakage, it is necessary to do signal processing by implementing Fast Fourier Transform. The width of the frequency spectrum pattern between 0-1000Hz is used as an Artificial Neural Network input with one hidden layer to recognize the tool breakage. Fast Fourier Transform and Artificial Neural Network processing are performed on Teensy 3.2 development board. Experimental results with 30000RPM spindle speeds indicate that Artificial Neural Network can detect tool breakage with 80% success rate. The use on harder PCB types increases the success rate to 91.67%. Implementation of the system on the CNC as PCB printer will improve efficiency in power consumption and time.

Keywords: artificial neural network, fast Fourier transformation, PCB printer, tool breakage

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala rahmat-Nya karena tugas akhir ini dapat berjalan lancar dan dapat selesai tepat pada waktunya. Tidak lupa juga penulis mengucapkan banyak terima kasih atas bantuan dari berbagai pihak, maka penulis mengucapkan terima kasih kepada :

1. Dr. Muhammad Rivai, ST., MT. dan Ir. Tasripan, MT. selaku dosen pembimbing serta yang senantiasa membimbing, mengarahkan, dan memotivasi penulis dalam mengerjakan penelitian ini.
2. Dr. Ir. Hendra Kusuma, M.Eng.Sc., Ir. Haris Pirngadi, MT., Dr. Eng. Ir. Totok Mujiono M.Ikom., serta Dr. Astria Nur Irfansyah, ST., M.Eng. selaku dosen penguji dalam memberikan perbaikan dan masukan untuk penulis menyempurnakan penelitian ini.
3. Dr. Eng. Ardyono Priyadi, ST., M.Eng selaku Kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
4. Kedua orang tua penulis, Asan Hadi dan Sri Suharti yang terus memberi doa dan dorongan untuk penulis menyelesaikan penelitian ini.
5. Rekan-rekan Laboratorium Elektronika Industri B.402 yang memberikan bantuan dan semangat dalam mengerjakan penelitian ini.

Terlepas dari semua itu, penulis sadar bahwa masih ada kekurangan pada berbagai aspek dan masih banyak hal yang perlu diperbaiki. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran dari pembaca agar dapat memperbaiki Tugas Akhir ini. Semoga makalah ini dapat memberikan manfaat bagi pembaca dan selanjutnya dapat dikembangkan kembali pada penelitian yang lebih mendalam.

Surabaya, Januari 2018
Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Metodologi Penelitian	2
1.5 Sistematika Penulisan	3
1.6 Relevansi	4
BAB II DASAR TEORI	5
2.1 Pencetak PCB	5
2.1.1 CNC <i>Spindle</i>	6
2.1.2 Mata bor	6
2.2 Getaran	7
2.3 <i>Accelerometer</i>	8
2.3.1 Sensor <i>Accelerometer</i> MEMS	8
2.3.2 Sensor ADXL335	10
2.4 <i>Fourier Transform</i>	12
2.4.1 <i>Discrete Fourier Transform</i>	12
2.4.2 <i>Fast Fourier Transform</i>	12
2.5 <i>Artificial Neural Network</i>	14
2.6 <i>Teensy USB Development Board</i>	17
2.7 <i>Solid State Relay</i>	20
BAB III PERANCANGAN SISTEM	23
3.1 Deskripsi Sistem	23
3.2 Perancangan <i>Fast Fourier Transform</i>	24
3.3 Perancangan <i>Artificial Neural Network</i>	26
3.3.1 Pengambilan data untuk pembelajaran ANN	27
3.3.2 Pembelajaran ANN pada MATLAB	28
3.3.3 Penerapan ANN untuk pengenalan pola getaran	28
BAB IV PENGUJIAN ALAT	31
4.1 Pengujian Hardware	31
4.1.1 Pengujian ADC <i>Teensy 3.2</i>	31

4.1.2 Pengujian Sensor <i>Accelerometer</i>	32
4.2 Pengujian Software.....	34
4.2.1 Pengujian FFT	34
4.2.1.1 Pengujian dengan <i>Input Function Generator</i>	34
4.2.2.2 Pengujian dengan <i>Input Sensor</i>	36
4.2.2 Pengujian ANN Secara <i>Offline</i>	41
4.2.2.1 Pengambilan Data untuk Proses <i>Learning</i>	41
4.2.2.2 Proses <i>Learning</i>	43
4.2.2.3 Proses Pengujian secara <i>Offline</i> pada MATLAB.....	43
4.2.2.4 Proses Pengujian secara <i>Offline</i> pada <i>Teensy</i> 3.2.....	44
4.3 Pengujian Sistem Secara Keseluruhan.....	45
4.3.1 Pengujian Sistem dengan Satu Kecepatan Putaran <i>Spindle</i>	46
4.3.2 Pengujian Sistem dengan Berbagai Kecepatan Putaran <i>Spindle</i>	47
4.3.3 Pengujian Sistem dengan Perbedaan Jenis Papan Tembaga (PCB).....	48
BAB V PENUTUP	51
5.1 Kesimpulan	51
5.2 Saran	51
DAFTAR PUSTAKA	53
LAMPIRAN	55
BIODATA PENULIS	93

DAFTAR GAMBAR

Gambar 2.1	Pencetak PCB	5
Gambar 2.2	Spindle CNC.....	6
Gambar 2.3	<i>V-shape CNC engraving bits</i>	7
Gambar 2.4	Ilustrasi pegas	8
Gambar 2.5	Struktur <i>Accelerometer</i> MEMS	9
Gambar 2.6	Modul ADXL335	11
Gambar 2.7	<i>Functional Block Diagram</i> ADXL	11
Gambar 2.8	Alur Transformasi <i>Fourier</i> [7].....	12
Gambar 2.9	Sinyal getaran yang terukur osiloskop.....	13
Gambar 2.10	Transformasi sinyal dalam domain waktu ke domain frekuensi	14
Gambar 2.11	<i>Artificial Neural Network</i> dengan 3 <i>layer</i>	15
Gambar 2.12	Struktur <i>Artificial Neural Network</i>	16
Gambar 2.13	<i>Pinout</i> dari <i>Teensy 3.2</i> [10].....	19
Gambar 2.14	Diagram skematik dari <i>Teensy 3.2</i> [9].....	20
Gambar 2.15	<i>Solid State Relay</i>	21
Gambar 2.16	Skematik <i>Solid State Relay</i>	21
Gambar 3.1	Diagram blok sistem perangkat keras	23
Gambar 3.2	Diagram blok sistem pemrosesan pada <i>Teensy 3.2</i>	24
Gambar 3.3	Sinyal getaran pada domain waktu	24
Gambar 3.4	Sinyal getaran pada domain frekuensi	26
Gambar 3.5	Diagram blok proses pengambilan data untuk pembelajaran ANN	27
Gambar 3.6	Arsitektur <i>Artificial Neural Network</i>	29
Gambar 4.1	Diagram koneksi pengujian ADC.....	31
Gambar 4.2	Diagram koneksi pengujian sensor <i>accelerometer</i> ADXL335	32
Gambar 4.3	Hasil pengujian sensor <i>accelerometer</i> pada permukaan rata dan tanpa getaran.....	32
Gambar 4.4	Hasil pengujian sensor <i>accelerometer</i> saat diberi getaran <i>spindle</i>	33
Gambar 4.5	Pengukuran frekuensi sinyal getaran	33
Gambar 4.6	Skematik pengujian FFT.....	34
Gambar 4.7	Pengujian FFT dengan <i>input</i> sinyal sinusoidal 500Hz (domain waktu)	34
Gambar 4.8	Pengujian FFT dengan <i>input</i> sinyal sinusoidal 500Hz (domain frekuensi)	35

Gambar 4.9	Pengujian FFT dengan <i>input</i> sinyal kotak 500Hz (domain waktu)	35
Gambar 4.10	Pengujian FFT dengan <i>input</i> sinyal kotak 500Hz (domain frekuensi)	36
Gambar 4.11	Pemasangan ADXL335 dan <i>Teensy 3.2</i> pada <i>spindle</i>	36
Gambar 4.12	Pengujian FFT dengan sensor pada <i>X-axis</i> tanpa menggores (domain waktu)	37
Gambar 4.13	Pengujian FFT dengan sensor pada <i>X-axis</i> tanpa menggores (domain frekuensi)	37
Gambar 4.14	Pengujian FFT dengan sensor pada <i>X-axis</i> dengan menggores (domain waktu).....	38
Gambar 4.15	Pengujian FFT dengan sensor pada <i>X-axis</i> dengan menggores (domain frekuensi).....	38
Gambar 4.16	Pengujian FFT dengan sensor pada <i>Y-axis</i> tanpa menggores (domain waktu)	38
Gambar 4.17	Pengujian FFT dengan sensor pada <i>Y-axis</i> tanpa menggores (domain frekuensi)	39
Gambar 4.18	Pengujian FFT dengan sensor pada <i>Y-axis</i> dengan menggores (domain waktu).....	39
Gambar 4.19	Pengujian FFT dengan sensor pada <i>Y-axis</i> dengan menggores (domain frekuensi).....	39
Gambar 4.20	Pengujian FFT dengan sensor pada <i>Z-axis</i> tanpa menggores (domain waktu)	40
Gambar 4.21	Pengujian FFT dengan sensor pada <i>Z-axis</i> tanpa menggores (domain frekuensi)	40
Gambar 4.22	Pengujian FFT dengan sensor pada <i>Z-axis</i> dengan menggores (domain waktu).....	40
Gambar 4.23	Pengujian FFT dengan sensor pada <i>Z-axis</i> dengan menggores (domain frekuensi).....	41
Gambar 4.24	Proses pengambilan data untuk <i>learning</i>	42
Gambar 4.25	<i>Software</i> untuk menampung data serial dan menyimpannya	42
Gambar 4.26	Hasil <i>learning</i> ANN di MATLAB	43
Gambar 4.27	Grafik pengujian ANN secara <i>offline</i> pada MATLAB	44
Gambar 4.28	Grafik pengujian ANN secara <i>offline</i> pada	

	<i>Teensy 3.2</i>	45
Gambar 4.29	Sistem secara keseluruhan	46
Gambar 4.30	Grafik pengujian sistem dengan satu kecepatan <i>spindle</i>	47
Gambar 4.31	Grafik pengujian sistem dengan banyak kecepatan spindle.....	48
Gambar 4.32	Grafik pengujian sistem dengan material PCB Fiber.....	49

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Pola target dan nilai target dari Artificial Neural Network 28

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi dalam bidang industri membawa pengaruh positif pada proses produksi. Pemanfaatan *Computer Numerical Control* (CNC) adalah salah satu bentuk penerapan teknologi industri yang membuat hasil produksi lebih presisi dan akurat. CNC juga dapat diterapkan pada proses pencetakan *Printed Circuit Board* (PCB) menggantikan proses pelarutan secara kimiawi yang tidak ramah lingkungan. Namun pengoperasian mesin CNC dalam kurun waktu tertentu akan menimbulkan perubahan dalam mesin tersebut karena pengaruh faktor eksternal [1]. Sehingga dalam bidang industri pemantauan proses produksi dan kesalahan yang timbul adalah hal yang wajib untuk dilakukan.

Menggunakan CNC untuk memproses material keras dapat menimbulkan kerusakan pada mata bor. Suatu bentuk kerusakan dalam proses produksi adalah hambatan. Untuk sebuah industri, hambatan tersebut menyebabkan kerugian finansial dan waktu. Tidak terdapatnya pendeteksi kepatahan mata bor yang bekerja secara realtime pada mesin CNC yang ada di pasaran membuat kerugian produksi semakin tinggi karena energi tetap diserap namun alat tidak bekerja sebagaimana mestinya sehingga energi tersebut akan terbuang percuma.

Pendeteksian kerusakan mata bor sebenarnya telah dipelajari dengan berbagai bentuk dan cara. Berdasar dari pengetahuan yang didapat dari penelitian pemantauan kondisi mata bor secara konvensional, pendeteksian tak langsung seperti daya, gaya pemotongan, getaran, dan sinyal emisi akustik adalah kandidat untuk dapat mendeteksi kondisi mata bor. Dengan mempertimbangkan murahnya harga MEMS *accelerometer*, sistem pendeteksi getaran pada *spindle* menawarkan keuntungan dari mudahnya pemasangan dan biaya yang efektif [2].

Masalah di atas melatarbelakangi penulis untuk melakukan penelitian ini. Diharapkan dari penelitian ini akan didapatkan suatu bentuk pendeteksi kepatahan mata bor yang efektif, murah, dan mudah

dipasang. Dengan begitu, dapat meningkatkan efisiensi, akurasi dan stabilitas mesin pencetak PCB.

1.2. Perumusan Masalah

Dari judul tugas akhir tersebut, didapatkan rumusan masalah, yaitu:

1. Bagaimana mendeteksi kepatahan mata bor?
2. Metode apa yang digunakan untuk mengenali kepatahan mata bor?
3. Bagaimana mengenali kepatahan mata bor secara otomatis?
4. Bagaimana cara mengimplementasikan metode ini agar lebih ringkas?

1.3. Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah :

1. Penggunaan sensor *accelerometer* untuk mengukur getaran pada *spindle* agar dapat mendeteksi kepatahan mata bor.
2. Metode yang digunakan adalah *Fast Fourier Transform*. untuk mengenali kepatahan mata bor.
3. Penggunaan algoritma *Artificial Neural Network* untuk mengenali kepatahan mata bor secara otomatis.
4. Penggunaan *Teensy USB Developement Board* untuk mengimplementasikan metode ini agar lebih ringkas.

1.4. Metodologi Penelitian

Pada penelitian tugas akhir ini, digunakan metodologi penelitian sebagai berikut :

1. Studi literatur
Tahap ini meliputi pengumpulan dasar teori yang dapat menjadi acuan tugas akhir. Dasar teori akan diambil dari buku – buku, jurnal yang telah dipublikasikan, *proceeding*, dan artikel dari internet.
2. Perancangan Sistem *Hardware*
Tahapan ini akan meliputi perancangan alat yang nantinya digunakan untuk menunjang penelitian ini. Secara khusus perancangan *hardware* sebagai berikut :
 - a. Perakitan *printer* PCB sebagai subjek penelitian
 - b. Pengintegrasian sensor *accelerometer* dengan *Teensy Board* untuk dapat membaca getaran pada *spindle printer* PCB.

- c. Pemasangan alat pada *spindle*.
3. Perancangan Sistem *Software*
Pada tahapan ini akan dilakukan perancangan dan pembuatan *software / firmware* pengatur (*controller*). *Software / firmware* ini yang akan memproses sinyal dari sensor *accelerometer* untuk dapat digunakan sebagai penentu rusak tidaknya mata bor yang dipakai.
4. Pengujian Sistem
Pengujian sistem dilakukan dengan tahap-tahap sebagai berikut:
 - a. Pengujian performasi sensor
Pengujian ini dilakukan dengan membaca nilai pengukuran oleh sensor menggunakan sumber getaran yang terukur.
 - b. Pengujian FFT
Pengujian ini dilakukan untuk mengukur keberhasilan algoritma *software* dalam mengubah gelombang sinyal dari domain waktu ke domain frekuensi
 - c. Pengujian ANN
Pengujian ini dilakukan untuk mengukur keberhasilan algoritma *software* dalam mengenali bentuk getaran apakah mata bor yang digunakan patah atau tidak.
 - d. Pengujian keseluruhan dan pengambilan data
Pengujian ini dilakukan dengan menjalankan sistem secara keseluruhan sekaligus untuk mendapatkan data yang akan ditulis pada buku Tugas Akhir.
5. Penulisan Laporan Tugas Akhir
Setelah dilakukan segala percobaan dan pengambilan data, maka akan dilakukan penulisan laporan tugas akhir yang final serta penulisan POMITS.

1.5. Sistematika Penulisan

Laporan tugas akhir ini terdiri atas lima bab, yakni :

1. Bab 1 : Pendahuluan
Bagian pendahuluan memaparkan mengenai latar belakang, rumusan masalah, tujuan penelitian, metode penelitian, sistematika penulisan, serta relevansi penelitian.
2. Bab 2 : Dasar Teori
Bagian ini memaparkan mengenai teori serta penelitian terdahulu yang menunjang penelitian ini, seperti teori pendeteksian kepatahan mata bor serta metodenya.

3. Bab 3 : Perancangan Alat
Bagian perancangan alat menjelaskan mengenai perancangan untuk merealisasikan sistem.
4. Bab 4 : Pengujian Alat
Bagian ini memaparkan metode pengambilan data pada penelitian serta hasil analisa pengujian alat. Pengambilan data dilakukan secara langsung pada alat.
5. Bab 5 : Penutup
Bagian penutup memaparkan kesimpulan yang didapatkan dalam penelitian serta saran dalam pengembangan hasil penelitian.

1.6. Relevansi

Gagasan tugas akhir ini diharapkan dapat membantu penelitian dalam bidang pendeteksian kepatahan mata bor pada mesin cetak PCB dengan memanfaatkan getaran pada *spindle* sebagai objek ukurnya. Tugas akhir ini selanjutnya dapat dikembangkan lagi untuk diserap metodenya untuk mendeteksi kerusakan lain pada mesin lain.

BAB II

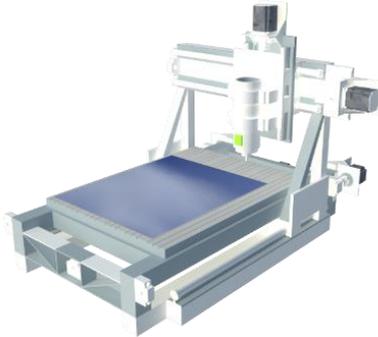
DASAR TEORI

Pada bab ini akan dibahas teori-teori yang menunjang penelitian ini, yakni mengenai pencetak PCB, sensor yang digunakan, metode pendeteksian, serta pengendali yang digunakan.

2.1. Pencetak PCB

Pencetak PCB adalah suatu peranti untuk mencetak jalur-jalur sirkuit pada media berlapis tembaga untuk selanjutnya dapat dipasang komponen. Dengan perkembangan teknologi, saat ini pencetakan PCB dapat dilakukan dengan mesin *Computer Numerical Control* (CNC) yang lebih ramah lingkungan. Jalur-jalur ini akan digores dan lubang untuk komponen akan dibor menggunakan mesin CNC [3]. Cara lain untuk melakukan proses pencetakan PCB adalah dengan proses kimia, yakni dengan FeCl_3 pada proses *etching* untuk membuat PCB [4]. Akan tetapi menurut hasil penelitian terbaru, penggunaan FeCl_3 sebagai zat kimia pelarut dapat menyebabkan risiko yang tinggi terhadap kesehatan [5].

Penggunaan CNC sebagai pencetak PCB akan mengurangi kontak FeCl_3 dengan manusia sehingga lebih aman. CNC adalah sebuah alat yang digunakan dalam proses produksi dan menggunakan komputer untuk mengontrol. CNC dapat digunakan sebagai mesin bubut, *milling*, *routing*, penggiling, dan pengebor. Dalam pencetak PCB fungsi yang digunakan adalah sebagai *router* dan pengebor. *Router* digunakan sebagai pembuat jalur PCB dan pengebor untuk mengebor lubang komponen PCB.



Gambar 2.1. Pencetak PCB



Gambar 2.2. *Spindle* CNC

2.1.1. CNC *Spindle*

Dalam CNC ada satu bagian yang berputar dan dengan kombinasi mata bor, bagian ini yang berperan untuk mengebor, *routing*, *milling*, dan sebagainya. Bagian tersebut adalah *spindle*. Secara visual, spindle ditunjukkan pada gambar 2.2.

Spindle sebenarnya merupakan sebuah motor dengan sumber AC maupun DC. Dalam penelitian ini, jenis *spindle* yang digunakan merupakan motor sinkron AC. *Spindle* ini yang nantinya dianalisa getarannya. Dari getaran *spindle* diharapkan dapat diketahui jika mata bor yang digunakan patah. Berikut merupakan spesifikasi *spindle* yang digunakan pada penelitian ini,

Tegangan kerja	: 220V AC
Daya Maksimum	: 150 W
Kecepatan putaran	: 10000 – 32000 rpm (<i>adjustable</i>)
Diameter mata bor	: 3.2mm (maksimum)

2.1.2. Mata Bor

Mata bor umum digunakan dalam penggunaan CNC untuk menggerus (*engrave*) material yang dalam penelitian ini PCB. Mata bor adalah sebuah alat untuk membentuk suatu lubang atau goresan dengan prinsip kerja mengurangi / mengikis bagian material yang dikenai.

Mata bor yang digunakan pada CNC biasa disebut dengan CNC bits atau CNC tool. CNC bits memiliki banyak varian bergantung pada tujuan penggunaannya. Untuk mencetak PCB, CNC bits atau mata bor yang tepat adalah yang berjenis *V-shape CNC engraving bits*. Jenis CNC bits ini dibagi dalam beberapa macam berdasar diameter dan sudut bagian



Gambar 2.3. *V-shape CNC engraving bits*

ujungnya. Gambar 2.3. menunjukkan ilustrasi *V-shape CNC engraving bits*.

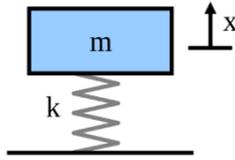
Spesifikasi CNC bits yang digunakan adalah sebagai berikut,

Tipe	: Flat bottom engraving bits
Bentuk	: V-shape
Derajat Kemiringan	: 30°
Material	: Tungsten steel
Diameter <i>shank</i>	: 3.175mm
Cutting edge diameter	: 0.2mm
Panjang total	: about 32mm
Material yang sesuai	: Nylon, Resin, ABS, Acrylic, PVC, MDF, hardwood and other three-dimensional sculpture, surface machining, ect.
Peruntukan	: 3D relief sculpturing, cutting, engraving, slotting, milling,ect.

2.2. Getaran

Getaran adalah perulangan gerakan pada sistem mekanik terhadap titik keseimbangan [6]. Getaran sendiri dapat dibagi dua bagian besar yaitu getaran yang periodik maupun acak.

Pada getaran yang periodik, maka getaran akan berulang pada interval waktu tertentu secara konstan. Sebaliknya pada getaran acak, getaran terjadi tidak konsisten dan berubah-ubah sepanjang waktu.



Gambar 2.4. Ilustrasi pegas

Getaran juga dapat diklasifikasikan menjadi getaran yang diinginkan dan tidak. Contoh getaran yang diinginkan yaitu yang terjadi pada *speaker*. Untuk getaran yang tidak diinginkan seperti pada peralatan mekanik, dapat mengindikasikan suatu kondisi tertentu. Prinsip dasar getaran dapat digambarkan sebagai sebuah gerak bebas dari per (*spring*) yang diilustrasikan pada gambar 2.4.

Persamaan untuk gerak bebas sebuah pegas adalah $F_s = -k \cdot x$. Di mana persamaan x dapat dituliskan sebagai persamaan 2.1.

$$X(t) = A \sin(2\pi \cdot fn \cdot t) \quad (2.1)$$

$$X(t) = A \sin(\omega t) \quad (2.2)$$

$$fn = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (2.3)$$

Persamaan 2.1 merupakan persamaan perpindahan getaran secara umum. Jika persamaan tersebut diturunkan 1 kali maka akan didapat persamaan kecepatan getaran. Dan jika diturunkan 2 kali maka akan didapat persamaan percepatan getaran.

$$v = \frac{dx}{dt} = \omega A \cos(\omega t) \quad (2.4)$$

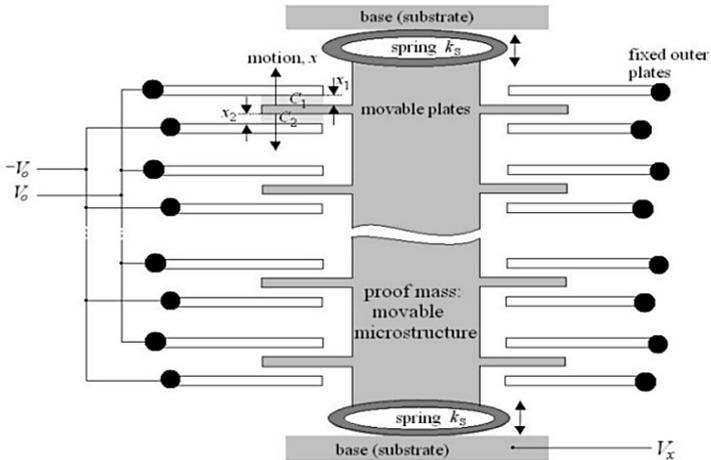
$$a = \frac{d^2x}{dt^2} = -\omega^2 A \sin(\omega t) \quad (2.5)$$

2.3. Accelerometer

2.3.1. Sensor Accelerometer MEMS

MEMS (*Micro-electro Mechanical System*) adalah merupakan sebuah sensor mekanik yang dikemas ke dalam bentuk *Integrated Circuit* (IC). Karena merupakan sebuah *microelectronic* maka komponen utama penyusunannya adalah silikon dan dalam ukuran mikron. Dan bisa digunakan untuk mengukur percepatan, posisi, atau kejutan. Karena material yang digunakan adalah jenis silikon, maka MEMS dapat dibuat dalam ukuran hingga mikron.

Pada sensor percepatan (*accelerometer*) konvensional yang ada menggunakan prinsip *piezoelectric* yang bentuk fisiknya lebih besar,



Gambar 2.5. Struktur *Accelerometer* MEMS

aplikasinya meletakkan sensor ke obyek yang diukur getarannya sedikit lebih susah dan memerlukan daya yang cukup besar. Dengan beberapa kelebihan tersebut diharapkan MEMS dapat menggantikan sensor getaran yang lebih baik.

MEMS untuk percepatan memiliki struktur kapasitansi untuk mendeteksi adanya perubahan percepatan (*acceleration*). Di mana kapasitansi akan berubah-ubah sesuai dengan pergerakan dari getaran sebagai perubahan percepatan.

Di mana ϵ merupakan permitifitas bahan yang pada MEMS digunakan udara sehingga suhu cukup memengaruhi besarnya permitifitas bahan. a adalah luas permukaan plat dan d adalah jarak antar plat. Gambar 2.5. menunjukkan pemodelan struktur *accelerometer* MEMS.

Jika percepatan sama dengan nol (tidak ada percepatan), maka $X_1 = X_2 = X_3$ dan nilai $C_1 = C_2$. Dan di mana nilai X_1 dan X_2 dihasilkan dari pergerakan dari pelat pembentuk kapasitansi pada MEMS yang akan berubah-ubah jika adanya percepatan yang timbul pada MEMS. Jika ada percepatan yang muncul maka nilai dari C_1 dan C_2 akan menjadi :

$$C_1 = \epsilon \frac{1}{x_1+x} \text{ dan } C_2 = \epsilon \frac{1}{x_1-x} \quad (2.6)$$

Sehingga perbedaan kapasitansinya menjadi :

$$\Delta C = C_1 - C_2 = 2\epsilon \frac{x}{x_2 - x_1^2} \quad (2.7)$$

Sehingga akan didapat persamaan aljabar menjadi :

$$\Delta C x^2 - 2\epsilon x - \Delta C x_1^2 = 0 \quad (2.8)$$

Untuk pergerakan percepatan yang terlalu kecil, maka Cx_2 dapat diabaikan. Sehingga persamaan di atas menjadi :

$$x \approx -\frac{x_1^2}{2\epsilon} \Delta C \quad (2.9)$$

Sehingga dapat dikatakan bahwa nilai pergerakan x akan mendekati nilai perbedaan kapasitansi C . Menurut hukum pegas, bahwa gaya F_s proporsional dengan pergerakan x dan sesuai dengan persamaan pegas menjadi :

$$F_s = K \cdot x \quad (2.10)$$

Berdasar hukum Newton kedua maka dapat dituliskan

$$m \cdot a = m \frac{Kx_1^2}{dt^2} = k \cdot x \quad (2.11)$$

Sehingga :

$$a = -\frac{k}{m} x \quad (2.12)$$

$$a = -\frac{Kx_1^2}{2m\epsilon} \Delta C \quad (2.13)$$

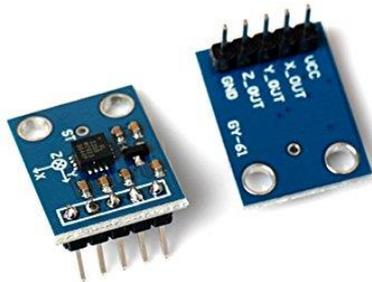
Di sini dapat disimpulkan bahwa percepatan (*acceleration*) dapat dipengaruhi oleh besarnya perbedaan kapasitansi dari dalam MEMS.

2.3.1. Sensor ADXL335

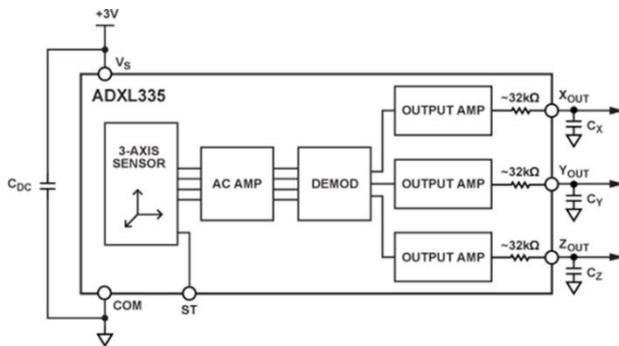
Di dunia industri, sensor getaran pada umumnya yang digunakan ada dua tipe yaitu tipe *proximity* dan tipe *accelerometer*. Tipe *proximity* digunakan untuk mengukur pergeseran benda-benda yang berputar misalnya pergeseran *shaft*. Tipe *proximity* ini tidak bersentuhan langsung dengan obyek yang diukur. Sementara tipe *accelerometer* ini digunakan untuk mengukur vibrasi benda-benda yang statis seperti mengukur vibrasi pada rumah *bearing*. Tipe *accelerometer* ini bersentuhan langsung dengan obyek yang diukur.

Sensor yang digunakan dalam penelitian ini adalah sensor *accelerometer* ADXL335. ADXL335 ini memiliki jangkauan pengukuran pada $\pm 3,6g$ (gravitasi). Sinyal keluaran tegangan analog yang dihasilkan oleh sensor ini berbanding lurus dengan percepatan. *Accelerometer* dapat mengukur percepatan statis gravitasi dalam aplikasi deteksi kemiringan dan juga mengukur percepatan dinamis yang dihasilkan dari gerakan, kejutan/guncangan atau getaran.

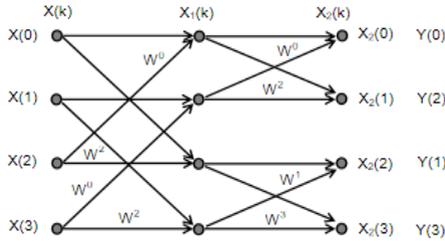
Accelerometer ADXL335 adalah sejenis *Micro-Electro Mechanical Sensor (MEMS)*. Dari namanya ada hubungan dengan benda mekanik dan sensor elektronik di dalamnya. Disebut mikro karena ukuran komponen dalam *chips* seukuran 0,001 – 0,1 mm. Prinsip kerja yang digunakan adalah prinsip percepatan (*acceleration*). Sebuah per dengan beban dan dilepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu akan berhenti. Bila ada sesuatu yang mengguncangnya maka beban akan berayun kembali. Konsep ini diterapkan pada sebuah *chip* kecil, di mana per digantikan dengan plat mini yang ujungnya diberi penyangga dan di tengahnya diberi beban. Lain pelat dipasang dalam posisi tetap berhadapan. Saat ada gaya dari luar maka pelat beban akan bergerak mendekat pelat tetap, bila pelat sebagai kapasitor maka besaran kapasitansi *chip MEMS*. Agar sensor ini bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing-masing.



Gambar 2.6. Modul ADXL335



Gambar 2.7. *Functional Block Diagram* ADXL335



Gambar 2.8. Alur Transformasi *Fourier* [7]

2.4. *Fourier Transform*

2.4.1. *Discrete Fourier Transform*

Discrete Fourier Transform atau Transformasi *Fourier* Diskrit adalah suatu bentuk transformasi *fourier* terhadap sinyal diskrit atau digital. Secara umum transformasi *fourier* adalah suatu bentuk persamaan matematis untuk mengubah domain waktu X menjadi domain atau spektrum frekuensi Y , yaitu menguraikan sinyal menjadi komponen sinusoidal penyusunnya [7] [8]. Persamaan Transformasi *Fourier* Diskrit ditujukan pada persamaan 2.14.

$$Y(n) = \sum_{k=0}^{N-1} X(k) \cdot W^{nk} \quad (2.14)$$

Dengan $W = e^{-j2\pi/N}$, $j = \sqrt{-1}$, dan $N = 2^\gamma$

N merupakan jumlah data sinyal yang dicuplik, γ merupakan bilangan bulat positif, sedangkan k dan n masing-masing adalah indeks pada domain waktu dan frekuensi ditunjukkan pada persamaan 2.15 dan 2.16.

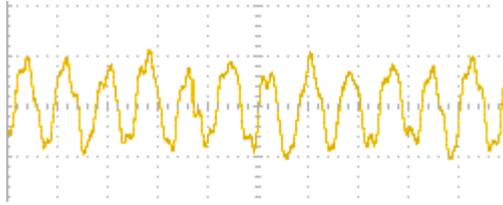
$$k = k_0 + 2k_1 + 4k_2 + \dots + 2^{\gamma-2}k_{\gamma-2} + 2^{\gamma-1}k_{\gamma-1} \quad (2.15)$$

$$n = n_0 + 2n_1 + 4n_2 + \dots + 2^{\gamma-2}n_{\gamma-2} + 2^{\gamma-1}n_{\gamma-1} \quad (2.16)$$

Gambar 2.7. menunjukkan aliran penyelesaian empat buah titik sampel ($N=4$) pada domain waktu ditransformasikan menjadi empat buah titik pada domain frekuensi dengan hanya dua informasi frekuensi ($N/2$) yang diperoleh.

2.4.2. *Fast Fourier Transform*

Analisa *getaran* yang digunakan pada penelitian ini adalah menggunakan metode *Fast Fourier Transform* (FFT). *Fast Fourier Transform* adalah suatu algoritma untuk menerapkan transformasi *fourier*



Gambar 2.9. Sinyal getaran yang terukur osiloskop

diskrit secara efisien. Sinyal vibrasi yang diukur oleh *transducer* masih sebagai fungsi waktu dengan satuan amplitudo tertentu. Gambar 2.7 adalah contoh hasil pengukuran fungsi waktu vibrasi.

Agar dapat dilakukan analisa terhadap getaran, maka perlu dilakukan transformasi dari domain waktu ke domain frekuensi. Metode untuk melakukan transformasi tersebut adalah menggunakan FFT.

Persamaan dasar *Fast Fourier Transform* ditunjukkan pada persamaan 2.17 sampai persamaan 2.20.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.17)$$

Dengan n, k, N adalah bilangan bulat

$$k = 0, 1, 2, 3, \dots, N - 1$$

$$j = \sqrt{-1}$$

$$W_N = e^{-j2\pi/N} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N} \quad (2.18)$$

Karena $x(n) = x_r(n) + jx_i(n)$ bisa bernilai kompleks, maka

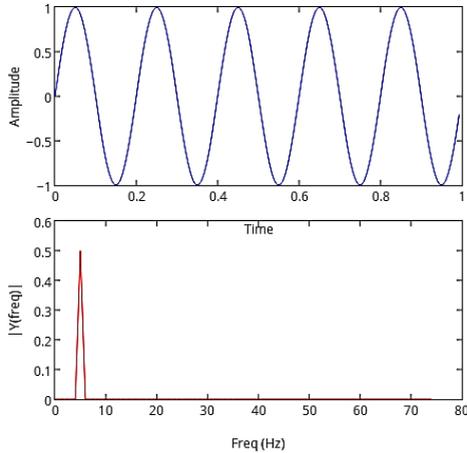
$X(k) = X_R(k) + jX_I(k)$ yaitu :

$$X_R(k) = \sum_{n=0}^{N-1} \left[X_r(n) \cos 2\pi \frac{k}{N} n + X_i(n) \sin 2\pi \frac{k}{N} n \right] \quad (2.19)$$

$$X_I(k) = \sum_{n=0}^{N-1} \left[X_r(n) \sin 2\pi \frac{k}{N} n + X_i(n) \cos 2\pi \frac{k}{N} n \right] \quad (2.20)$$

Dari persamaan tersebut dapat dilihat memerlukan $2N^2$ evaluasi fungsi trigonometri ditambah $4N^2$ perkalian real ditambah $4N(N-1)$ penjumlahan real dan sejumlah indexing dan addressing operators. Hal tersebut sering disebut dengan $O(N^2)$. Persamaan ini merupakan bentuk transformasi *fourier* yang siap dihitung secara langsung dari bentuk sinyal $x(t)$.

Gambar 2.10. adalah contoh mengubah sinyal sinus 5Hz dengan amplitudo 1 Vpp dari domain waktu ke domain frekuensi. Gambar sinyal sinus dalam domain waktu ditunjukkan gambar bagian atas dan gambar sinyal dalam domain frekuensi ditunjukkan pada gambar bagian bawah.



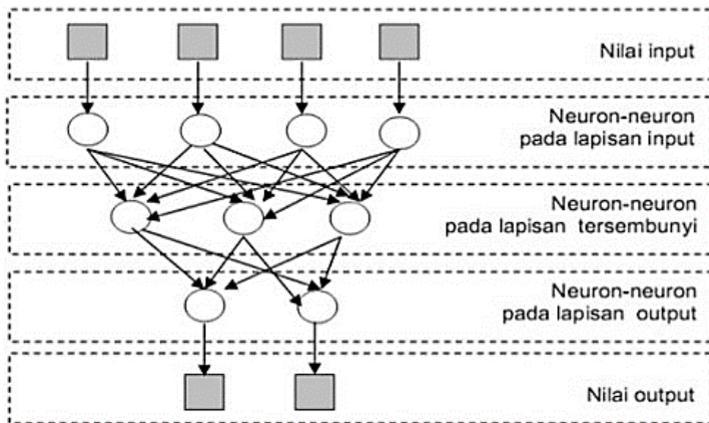
Gambar 2.10. Transformasi sinyal dalam domain waktu ke domain frekuensi

Gambar tersebut menunjukkan bahwa sinyal *input* yang diumpankan merupakan sinyal sinus murni karena spektrum frekuensi yang muncul hanya ada pada satu titik.

2.5. *Artificial Neural Network*

Artificial Neural Network (ANN) adalah bentuk pemodelan komputer dari sistem saraf manusia. Metode ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

Setiap pola-pola informasi *input* dan *output* yang diberikan ke dalam ANN diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron *layers*. Lapisan-lapisan penyusun ANN tersebut dapat dibagi menjadi 3, yaitu:

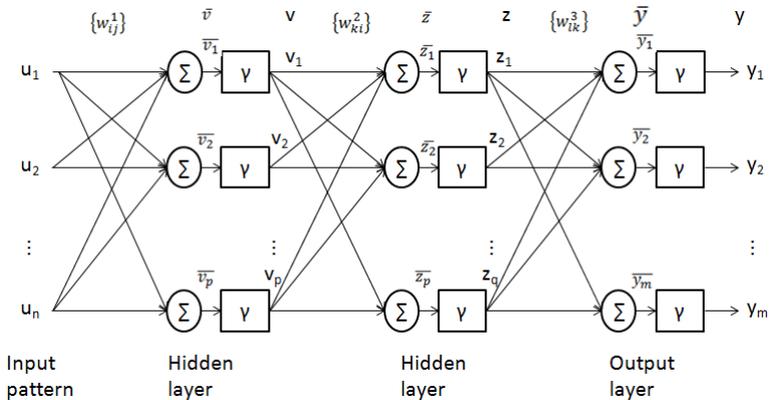


Gambar 2.11. *Artificial Neural Network* dengan 3 layer

1. Lapisan *input*
Unit-unit di dalam lapisan *input* disebut unit-unit *input*. Unit-unit *input* tersebut menerima pola *input* data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi
Unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di mana *output*-nya tidak dapat secara langsung diamati.
3. Lapisan *output*
Unit-unit di dalam lapisan tersembunyi disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi ANN terhadap suatu permasalahan.

Metode algoritma yang digunakan dalam penelitian ini adalah algoritma *backpropagation*. Propagasi balik atau *backpropagation* merupakan suatu teknik pembelajaran/pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

Dalam jaringan propagasi balik, setiap unit yang berbeda di lapisan *input* terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan *output*. Jaringan ini terdiri dari banyak lapisan



Gambar 2.12. Struktur *Artificial Neural Network*

(*multilayer network*). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan ke unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respons sebagai keluaran ANN. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) ke lapisan tersembunyi lalu kembali ke lapisan masukan. Tahap pelatihan ini merupakan langkah untuk melatih suatu ANN dengan melakukan perubahan bobot. Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai. Fase ini disebut fase pengujian.

Aturan pelatihan jaringan propagasi balik terdiri dari 2 tahapan, *feedforward* dan *backward propagation*. Jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini diasosiasikan dengan sebuah *vector feature* yang disebut vektor *input* dan diasosiasikan pula dengan vektor *output* yang menjadi target pelatihannya. Keluaran dari jaringan berupa sebuah vektor *output* aktual. Selanjutnya dilakukan perbandingan antara *output* aktual yang dihasilkan dengan *output* target dengan cara melakukan operasi pengurangan pada kedua *output* tersebut. Hasil dari pengurangan tersebut merupakan nilai *error*. *Error* dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikan kembali. Setiap perubahan bobot yang terjadi dapat mengurangi *error*. Siklus setiap perubahan bobot dilakukan pada setiap set pelatihan hingga

tercapainya jumlah bobot yang dikehendaki atau hingga sebuah nilai ambang yang ditetapkan terlampaui

Algoritma pelatihan jaringan propagasi balik terdiri dari 3 tahapan yaitu:

1. Tahap umpan maju (*feedforward*)
2. Tahap umpan mundur (*backpropagation*)
3. Tahap memperbarui bobot.

Algoritma *feedforward* ditunjukkan pada persamaan 2.21.

$$y(k) = \sum_{i=0}^m w(k, i) * f(i) \quad (2.21)$$

Dengan aktivasi *log sigmoid*, sehingga

$$output = \frac{1}{1 + \exp(-y(k))} \quad (2.22)$$

Algoritma *backpropagation* yang digunakan ditunjukkan pada persamaan 2.23.

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} x_{out,i}^{(s-1)} \quad (2.23)$$

dengan $\delta_j^{(s)} = (d_{qh} - x_{out,i}^{(s)})g(v_j^{(s)})$ untuk layer terluar

$$\delta_j^{(s)} = (\sum_{h=0}^{s+1} \delta_h^{(s+1)} w_{hj}^{(s+1)})g(v_j^{(s)}) \quad (2.24)$$

Algoritma untuk memperbarui nilai *Weight* dan *Bias* ditunjukkan pada persamaan 2.25 sampai persamaan 2.28.

$$\Delta w^m(k) = \gamma \Delta w^m(k-1) - (1-\gamma)\alpha s^m (a^{m-1})^T \quad (2.25)$$

$$\Delta b^m(k) = \gamma \Delta b^m(k-1) - (1-\gamma)\alpha s^m \quad (2.26)$$

$$w^m(k+1) = w^m(k) + \Delta w^m(k) \quad (2.27)$$

$$b^m(k+1) = b^m(k) + \Delta b^m(k) \quad (2.28)$$

2.6. Teensy USB Development Board 3.2

Teensy 3.2 adalah *development board* berbentuk kecil dan dapat dipasang pada *breadboard* yang di desain oleh Paul Stoffregen dan PJRC. Teensy 3.2 membawa 32-bit ARM Cortex-M4 yang murah kepada penghobi, siswa, dan *engineer* menggunakan versi adaptasi dari Arduino IDE (Teensyduino) atau diprogram secara langsung di bahasa C. Selain itu, Teensy 3.2 juga telah disertai dengan *bootloader* sehingga dapat diprogram langsung melalui koneksi USB dan tidak membutuhkan *external programmer* lagi [9].

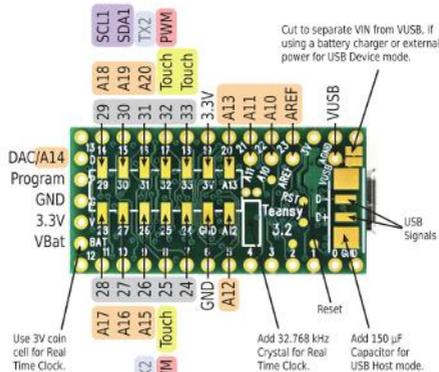
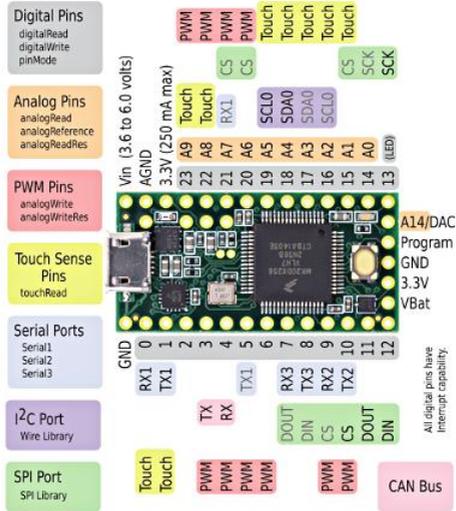
Prosesor Teensy 3.2 memiliki akses USB untuk mengemulasi berbagai peranti USB yang dibutuhkan seperti USB-MIDI dan USB HID.

Prosesor 32-bit membawa fitur lainnya seperti *multiple channel Direct Memory Access*, beberapa ADC beresolusi tinggi dan bahkan antarmuka I2S audio digital. *Teensy 3.2* juga memiliki 4 interval *timer* dan satu buah *delay timer*. Semua pin pada *Teensy 3.2* juga memiliki kemampuan untuk *interrupt* dan mentoleransi *input 5V*. Regulator tegangan pada *Teensy 3.2* juga mampu memberikan tegangan 3.3V dengan arus sampai dengan 250mA.

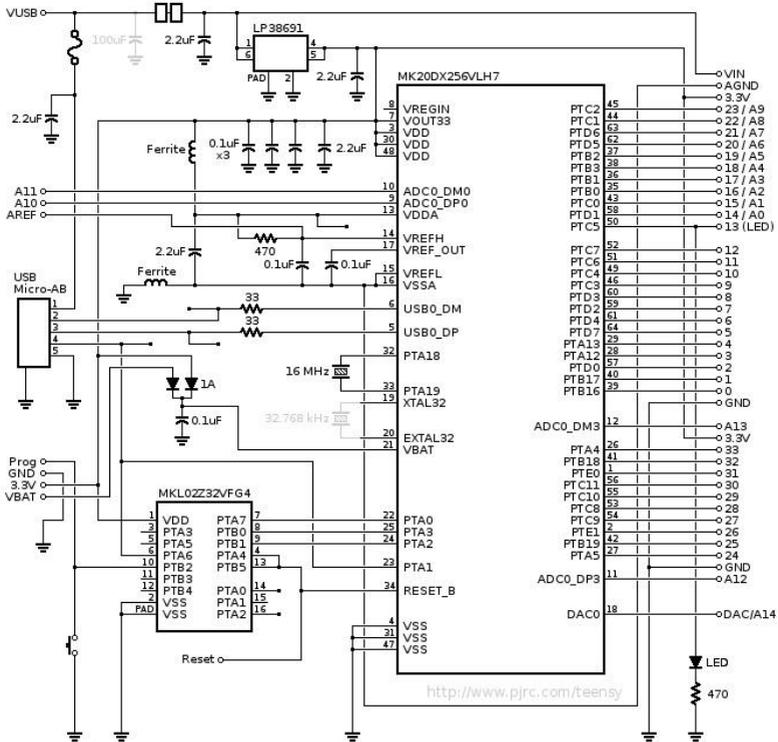
Pemilihan *Teensy 3.2* berdasar pada kebutuhan pemrosesan sinyal yang akan memakan *resources* yang besar namun tetap memperhatikan keringkasan. Hal ini bertujuan agar mudah diimplementasikan pada mesin CNC. Selain itu, ADC yang beresolusi tinggi juga dapat dimanfaatkan untuk dapat mengukur sinyal kecil dari sensor. Pinout *Teensy* ditunjukkan pada gambar 2.11 untuk memudahkan instalasi. Pada gambar ini juga dapat diketahui pin mana yang dapat digunakan sebagai analog *input*, serial, PWM, dan sebagainya. Pada gambar 2.12 ditunjukkan skematik internal pada *Teensy 3.2 Development Board*. Skematik ini berfungsi untuk menelusuri koneksi antar komponen jika diperlukan *troubleshooting*.

Berikut adalah spesifikasi teknis dari *Teensy 3.2* :

- 32 bit ARM Cortex-M4 72MHz CPU (M4 = DSP extensions)
- 256K Flash Memory, 64K RAM, 2K EEPROM
- 21 buah High Resolution Analog Inputs (12 bits usable, 16 bit hardware)
- 34 buah Digital I/O Pins (21 shared with analog)
- 12 buah PWM outputs
- 1 buah 12-bit DAC output
- 8 Timers for intervals/delays, separate from PWM
- USB with dedicated DMA memory transfers
- CAN bus
- 3 buah UARTs (serial ports)
- SPI, I2C, I2S, IR modulator
- I2S (for high quality audio interface)
- Real Time Clock (with user-added 32.768 crystal and battery)
- 16 general purpose DMA channels (separate from USB)
- Touch Sensor Inputs



Gambar 2.13. Pinout dari Teensy 3.2 [10]



Gambar 2.14. Diagram skematik dari Teensy 3.2 [9]

2.7. Solid State Relay

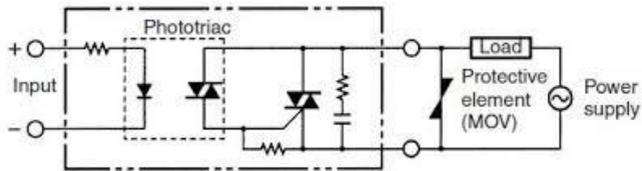
Berbeda dengan *relay* konvensional yang menggunakan elektromagnetik untuk pensaklaran, *Solid State Relay* (SSR) melakukan pensaklaran dengan *Field Effect Transistor* yang dikendalikan oleh *photodiode* [11]. Sedangkan *Light Emitting Diodes* (LED) menyediakan cahaya yang dibutuhkan. Sehingga antara pengendali dan beban dipisahkan oleh komponen optik. Kelebihan SSR adalah tidak adanya bagian yang bergerak, berumur panjang, dan dapat diandalkan.

Dalam penelitian ini, *Solid State Relay* digunakan sebagai aktuatur untuk bertindak sebagai saklar *spindle* jika terdeteksi adanya kepatahan mata bor. SSR dipilih karena lebih aman digunakan dan memiliki respons

yang cepat mengingat SSR tidak menggunakan *coil* atau lilitan elektromagnetis. Wujud sesungguhnya dari SSR ditunjukkan pada gambar 2.13 dan skematik rangkaiannya ditunjukkan pada gambar 2.14.



Gambar 2.15. *Solid State Relay*



Gambar 2.16. Skematik *Solid State Relay* [12]

Halaman ini sengaja dikosongkan

BAB III

PERANCANGAN SISTEM

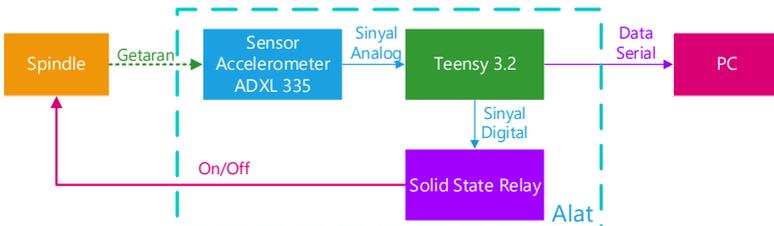
Pada penelitian ini, metode penelitian dibagi menjadi 2 bagian besar yaitu sistem identifikasi getaran saat melakukan pengukuran *on-line* dan sistem pembelajaran *artificial neural network*.

3.1. Deskripsi Sistem

Perancangan sistem diperlukan agar penelitian ini dapat direalisasikan. Diagram blok sistem dari perangkat keras dan perangkat lunak diperlukan agar mempermudah proses implementasi pembuatan sistem.

Gambar 3.1 menunjukkan diagram penghubungan komponen perangkat keras. Perancangan perangkat keras terdiri dari tiga bagian yaitu pembacaan sensor, pengendalian *spindle*, dan pengiriman data ke PC. Keseluruhan sistem dikendalikan dan diproses oleh *Teensy 3.2*. Proses pembacaan data dilakukan oleh sensor *accelerometer* ADXL335 untuk membaca getaran dari *spindle*. Pengendalian *spindle* dilakukan oleh *Solid State Relay* untuk menyalakan atau mematikan *spindle*. PC dalam penelitian ini bertugas sebagai antarmuka untuk menampilkan nilai-nilai hasil pengolahan pada *Teensy 3.2* melalui komunikasi serial.

Berdasarkan gambar 3.1, alur kerja dari sistem adalah dengan mengukur nilai getaran pada *spindle* menggunakan sensor *accelerometer* ADXL335. Pemasangan sensor *accelerometer* ADXL335 adalah menempel permukaan *spindle* dan dekat dengan mata bor untuk dapat merasakan getaran *spindle* dengan baik. Kemudian sinyal analog dari sensor diolah pada *Teensy 3.2* untuk menentukan apakah *spindle* perlu



Gambar 3.1. Diagram blok sistem perangkat keras



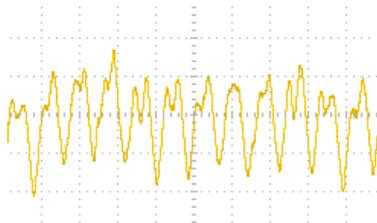
Gambar 3.2. Diagram blok sistem pemrosesan pada *Teensy 3.2*

dimatikan atau tidak yang dikontrol menggunakan SSR. Terakhir, nilai pengolahan pada *Teensy 3.2* ditampilkan juga pada PC.

Sistem pemrosesan yang terjadi pada *Teensy 3.2* ditampilkan pada gambar 3.2. Data pembacaan analog dari sensor *accelerometer* diubah ke nilai bit digital pada *Analog to Digital Converter (ADC)*. ADC internal *Teensy 3.2* memiliki resolusi 12-bit sehingga perubahan tegangan yang kecil dapat dibaca dengan baik. Konversi ini bertujuan untuk memudahkan pengolahan nilai lebih lanjut secara digital. bit-bit konversi nilai tegangan sensor diolah untuk dilakukan transformasi *fourier* yang dikerjakan oleh modul DSP internal *Teensy 3.2*. Pemrosesan pada DSP memungkinkan beban kerja pada prosesor *Teensy 3.2* berkurang karena tidak perlu lagi melakukan komputasi persamaan *Fast Fourier Transform* yang rumit secara manual. Hasil transformasi *fourier* yang berupa nilai bit digital selanjutnya akan diproses menggunakan *Artificial Neural Network* untuk mengidentifikasi pola.

3.2. Perancangan *Fast Fourier Transform*

Sinyal getaran seperti pada gambar 3.3 yang berada pada domain waktu sangat sulit untuk dianalisa. Untuk dapat melakukan analisa sinyal getaran dengan lebih mudah baik secara visual maupun secara otomatis dengan *Artificial Neural Network*, *Fast Fourier Transform* digunakan. *Fast Fourier Transform* akan memecah sinyal menjadi sinyal-sinyal



Gambar 3.3. Sinyal getaran pada domain waktu

sinusoidal penyusunnya yang akan menciptakan spektrum-spektrum frekuensi.

Selanjutnya dengan FFT, sinyal getaran pada domain waktu dikonversi ke domain frekuensi dengan persamaan 3.1.

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{N-1} x[n] \left\{ \cos\left(\frac{2\pi}{N}nk\right) - j \sin\left(\frac{2\pi}{N}nk\right) \right\}$$

untuk $k = 0, 1, 2, \dots, N - 1$ (3.1)

Fast Fourier Transform adalah bentuk implementasi dari *Discrete Fourier Transform*. Menjalankan operasi DFT membutuhkan kerumitan waktu operasi N^2 dan pada FFT dikurangi menjadi bentuk $N \log N$. *Fast Fourier Transform* menyelesaikan pengurangan kerumitan ini dengan dua bentuk pendekatan.

Pendekatan pertama menggunakan *decimation in time approach*. Pada pendekatan ini, cuplikan sejumlah N titik dibagi menjadi dua urutan, disebut dengan urutan ganjil dan genap. Setelah selesai, sebuah properti ditemukan dengan fungsi rekursif. Kemudian penghitungan $N/2$ pertama harus dilakukan sedangkan penghitungan $N/2$ kedua diselesaikan oleh fungsi rekursif untuk memperbarui sisa cuplikan.

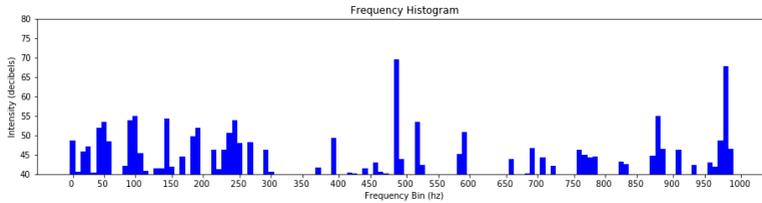
Pendekatan kedua menggunakan *decimation in frequency approach*. Pendekatan ini mirip dengan pendekatan sebelumnya, tetapi pembagian urutan menjadi dua bagian tidak berdasar urutan ganjil atau genapnya cuplikan (*sample*). Pembagian urutan dibentuk oleh $N/2$ pertama dan $N/2$ terakhir.

Semua penghitungan untuk memproses *Fast Fourier Transform* akan dikerjakan oleh DSP *Teensy 3.2* sehingga pemrosesannya akan lebih cepat dan meningkatkan efisiensi kinerja *Teensy 3.2*. Pada program FFT, sinyal pembacaan analog dari sensor *accelerometer* akan diumpangkan secara langsung dan akan menghasilkan *array* data berupa magnitudo spektrum frekuensi.

Hal yang perlu diperhatikan adalah, pemrosesan FFT pada *Teensy 3.2* ini sangat dipengaruhi oleh besarnya *sampling rate* dan *FFT size* (banyaknya spektrum FFT yang dihasilkan). Hubungan antara *sampling rate* dan *FFT size* adalah ditunjukkan pada persamaan 3.2.

$$\text{Freq per Magnitude} = \frac{\text{Sampling rate}}{\text{FFT size}} \quad (3.2)$$

Semakin tinggi *sampling rate* maka frekuensi yang diwakili setiap titik magnitudo akan semakin besar atau banyak menjadikan ketelitian hasil menjadi menurun. Masalah tersebut dapat diatasi dengan menambah *FFT size* sehingga frekuensi yang diwakili setiap magnitudonya akan



Gambar 3.4. Sinyal getaran pada domain frekuensi

tetap sedikit. Tetapi cara tersebut memiliki kerugian yaitu penurunan kinerja *Teensy 3.2*. Oleh sebab itu, ditetapkan aturan untuk pemrosesan pada *Teensy 3.2* dibatasi sampai maksimum 256 titik untuk *FFT size*. Jumlah *FFT size* tersebut merupakan nilai optimal untuk ketelitian yang baik dengan performa yang baik pula.

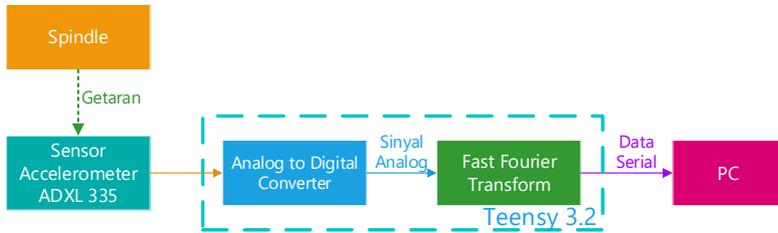
Hal lain yang perlu diperhatikan adalah jumlah *sampling rate* harus memenuhi aturan *Nyquist* yakni nilai *sampling* harus dua kali dari frekuensi sinyal yang disampling. Sehingga misal frekuensi yang hendak disampling maksimum pada 1000Hz, *sampling rate*-nya harus 2000Hz. Jika syarat ini tidak terpenuhi, maka yang nampak hanya frekuensi harmonisanya yang menyebabkan kesalahan perhitungan.

Hasil dari konversi sinyal getaran pada domain waktu (gambar 3.3) ke domain frekuensi ditunjukkan pada gambar 3.4. Hasil pengolahan FFT dari *Teensy 3.2* berupa nilai-nilai magnitudo dikirimkan ke PC yang selanjutnya ditampilkan pada *software Spectrogram* untuk mendapat visualisasi nilai magnitudo.

3.3. Perancangan Artificial Neural Network

Hasil yang didapat pada proses transformasi *fourier* selanjutnya perlu dikenali secara otomatis. Kerumitan spektrum frekuensi tidak bisa dikenali dengan hanya melakukan seleksi kondisi sederhana. Perlu algoritma *Artificial Neural Network* untuk menyelesaikannya.

Proses identifikasi pola oleh *Artificial Neural Network* dibagi menjadi dua tahap yakni tahap pembelajaran / *learning* dan tahap pengenalan pola getaran. Pada tahap pembelajaran ANN dilakukan pembaruan nilai bobot dan bias yang digunakan untuk proses identifikasi. Tahap ini dilakukan menggunakan *software MATLAB* di PC.



Gambar 3.5. Diagram blok proses pengambilan data untuk pembelajaran ANN

Tahap pembelajaran ANN akan didahului oleh proses pengambilan data. Proses pengambilan data dilakukan dengan menghubungkan perangkat seperti pada gambar 3.5.

3.3.1. Pengambilan data untuk pembelajaran ANN

Pada tahap ini dilakukan pengambilan data untuk setiap pola target. Ada tiga pola yang perlu dikenali oleh ANN. Masing-masing pola dilakukan pada tujuh tingkat kecepatan yang berbeda. Semakin banyak data yang ada, semakin tinggi pula akurasi dari algoritma ANN untuk mengenali pola.

Pengambilan data ini dilakukan pada mesin CNC pencetak PCB dengan media cetak PCB polos. CNC digerakkan melalui perintah PC dengan gerakan yang identik.

Nilai *output* dari *Teensy 3.2* yang diterima PC melalui komunikasi serial adalah besaran magnitudo masing-masing rentang frekuensi dalam bentuk bit. Untuk mengubah ke bentuk natural dari besaran yang terbaca oleh *Teensy 3.2* yaitu tegangan maka dihitung dengan persamaan 3.3. Sedangkan untuk normalisasi data dilakukan menggunakan persamaan 3.4.

$$\text{Output in voltage} = \text{Output in bit} \times \text{Volt per bit} \quad (3.3)$$

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.4)$$

3.3.1.1. Kondisi Spindle Tidak Berputar dan Tidak Menggores

Pada kondisi ini, *spindle* yang tidak berputar digerakkan dengan jalur tertentu yang juga akan diterapkan pada kondisi yang lain.

3.3.1.2. Kondisi Spindle Berputar dan Tidak Menggores

Pada kondisi ini, *spindle* dijalankan dengan mengatur kecepatan dari kecepatan paling rendah yang dalam hal ini dituliskan dengan angka 1

sampai kecepatan tertinggi yakni dituliskan dengan kata “max”. Setiap kecepatan diambil datanya secara terpisah. Menggerakkan *spindle* pada kondisi ini juga mengikuti jalur yang sama dengan kondisi sebelumnya.

3.3.1.3. Kondisi Spindle Berputar dan Menggores

Kondisi terakhir ini dilakukan dengan menjalankan kondisi sebelumnya tetapi dengan ujung mata bor menggores permukaan PCB polos. Gerakan *spindle* juga pada jalur yang sama dengan kondisi sebelumnya untuk menjaga konsistensi.

3.3.2. Pembelajaran ANN pada MATLAB

Setelah proses pengambilan data selesai, nilai yang terekam dikumpulkan dalam satu *file* berformat *spreadsheet* (excel) beserta target *output*-nya. *File* ini yang selanjutnya diakses melalui MATLAB untuk melakukan proses *learning* atau pembelajaran ANN guna mendapat nilai bobot (*weight*) dan *bias*. *Weight* dan *bias* ini selanjutnya diterapkan pada *Teensy 3.2* untuk mengenali pola getaran dan menentukan kepatahan mata bor. Proses pembelajaran ini menerapkan *backpropagation*.

Pada penelitian ini arsitektur ANN yang digunakan seperti yang digambarkan pada gambar 3.6. *Output* dari FFT akan menjadi *input* dari ANN. Pembelajaran ANN akan mengikuti nilai Target yang dituliskan pada tabel 3.1. Sedangkan untuk nilai *error* minimal yang diset adalah 10^{-5} dengan harapan bisa mendapatkan *output* yang akurat.

Setelah proses pembelajaran selesai perlu diverifikasi keberhasilan algoritma ANN beserta bobot dan biasanya dengan memasukkan data *learning*-nya sebagai *input* ANN dan diplot grafiknya dibandingkan dengan target yang telah diset. Semakin dekat antara grafik ANN dengan target maka semakin kecil *error*.

3.3.3. Penerapan ANN untuk pengenalan pola getaran

Nilai *Weight* dan *Bias* yang didapat pada proses pembelajaran *Artificial Neural Network* akan ditanamkan pada program *Teensy 3.2*. Hasil transformasi *fourier* berupa nilai-nilai magnitudo diubah ke bentuk

Tabel 3.1 Pola target dan nilai target dari *Artificial Neural Network*

Pola Target	Nilai Target
Spindle tidak berputar	0
Spindle berputar dan tidak menggerus	1
Spindle berputar dan menggerus	2

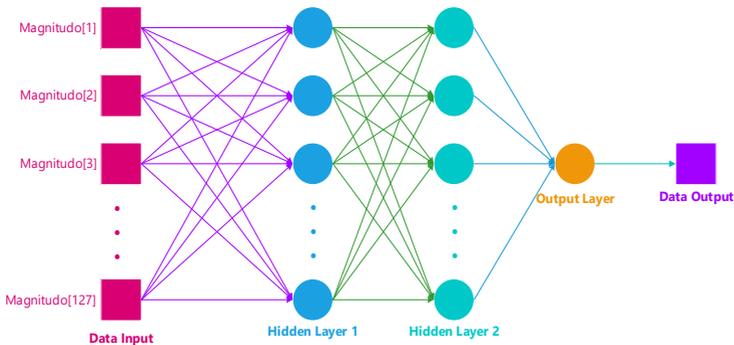
aslinya yaitu tegangan kemudian dinormalisasi. Hasil normalisasi inilah yang menjadi *input* dari ANN. Selanjutnya dengan ANN akan diketahui kondisi *spindle*.

Pada bagian ini ANN akan menggunakan model *Feed Forward* dengan 2 *hidden layer* yang masing-masing menggunakan aktivasi log sigmoid. Sedangkan untuk *layer output* akan menggunakan aktivasi linier. Persamaan untuk log sigmoid ditunjukkan oleh persamaan 3.5 dan untuk persamaan *Feed forward* ditunjukkan pada persamaan 3.6.

Gambaran arsitektur proses *Feed Forward Artificial Neural Network* yang diterapkan pada identifikasi pola getaran ini ditunjukkan pada gambar 3.6. Data dari normalisasi magnitudo output FFT menjadi data *input* pada *hidden layer 1*. Selanjutnya data diolah dengan persamaan *feed forward* dan aktivasi log sigmoid. *Output* dari *hidden layer 1* kemudian menjadi *input hidden layer 2*. Operasi penghitungan pada bagian ini sama dengan *layer* sebelumnya. Hasil dari *hidden layer 2* diteruskan ke *output layer* untuk dihitung secara linier dan didapat data *output*.

$$f(n) = 1 - \frac{1}{\exp(input)} \quad (3.5)$$

$$y(k) = \sum_{i=0}^m w(k, i) * f(i) \quad (3.6)$$



Gambar 3.6. Arsitektur *Artificial Neural Network*

Halaman ini sengaja dikosongkan

BAB IV

PENGUJIAN ALAT

Pada bab ini akan dilakukan pembahasan mengenai hasil-hasil pengujian sensor, ADC, FFT, dan sistem secara keseluruhan. Secara rinci, pengujian yang dilakukan adalah sebagai berikut,

- a. Pengujian *hardware*
 - Pengujian ADC *Teensy 3.2*
 - Pengujian Sensor *Accelerometer*
- b. Pengujian *software*
 - Pengujian FFT
 - Pengujian ANN secara *offline*
- c. Pengujian sistem secara keseluruhan
 - Pengujian dengan satu kecepatan putaran spindle
 - Pengujian dengan banyak kecepatan putaran spindle
 - Pengujian dengan variasi jenis PCB

4.1. Pengujian *Hardware*

4.1.1. Pengujian ADC *Teensy 3.2*

Pada pengujian ADC *Teensy 3.2*, pin 17 pada *Teensy 3.2* diberi tegangan DC sebesar 1Volt. Resolusi ADC *Teensy 3.2* diset pada 12bit. Hasil pembacaan dikirimkan via serial ke PC. Secara umum ditunjukkan pada diagram gambar 4.1.

Secara teoritis 1Volt DC akan menghasilkan

$$\frac{V_{input}}{V_{ref}} \times ADC \text{ Resolution} = \frac{1}{3.3} \times 2^{12} = 1241,2121 \approx 1241 \text{ bit}$$

dan menurut hasil pengujian didapatkan nilai 1254. sehingga dapat disimpulkan bahwa adanya kesesuaian antara perhitungan teoritis dan kondisi sesungguhnya dengan sedikit *margin error*.



Gambar 4.1. Diagram koneksi pengujian ADC



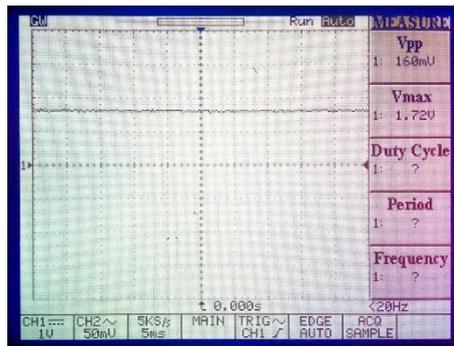
Gambar 4.2. Diagram koneksi pengujian sensor *accelerometer* ADXL335

4.1.2. Pengujian Sensor *Accelerometer*

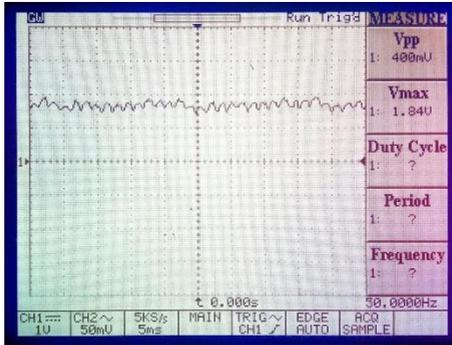
Pengujian sensor *accelerometer* dilakukan dengan menyambungkan ADXL335 dengan *power supply* 5VDC dan osiloskop sesuai dengan diagram koneksi pada gambar 4.2.

Proses pengujian ini dilakukan dengan menempatkan sensor pada permukaan yang rata dan tidak diberi getaran. Kemudian diamati sinyal *output*-nya. Selanjutnya sensor *accelerometer* dipasang pada sumber getaran yakni *spindle* yang menyala kemudian diamati sinyal *output*-nya juga.

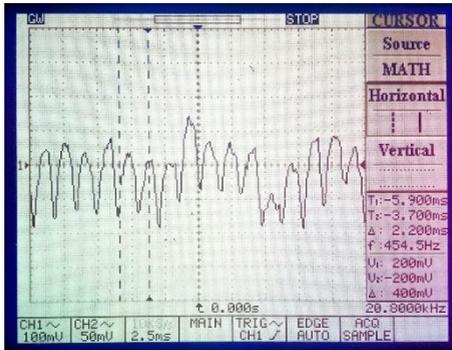
Dari pengujian ini didapat bahwa saat sensor *accelerometer* ditempatkan pada permukaan rata dan tidak diberi getaran hasil sinyal *output*-nya juga diam seperti yang ditunjukkan pada gambar 4.3. Berbeda kondisi saat diberikan getaran, hasil *output* yang terukur osiloskop nampak adanya sinyal dengan frekuensi dasar yang sama dengan frekuensi putaran *spindle* yakni berkisar pada 450Hz seperti pada gambar 4.4 yang diperjelas pada gambar 4.5.



Gambar 4.3. Hasil pengujian sensor *accelerometer* pada permukaan rata dan tanpa getaran



Gambar 4.4. Hasil pengujian sensor *accelerometer* saat diberi getaran *spindle*



Gambar 4.5. Pengukuran frekuensi sinyal getaran

Dari pengujian ini juga didapat kondisi bahwa sensor *accelerometer* ADXL335 bekerja pada tegangan positif atau dapat dikatakan tidak ada sinyal dengan tegangan negatif sehingga dapat disambungkan dengan *Teensy 3.2* tanpa memerlukan rangkaian tambahan. Selain itu tegangan *output* ADXL335 berayun pada tegangan referensi 1.7V. Sehingga dipastikan aman dan sesuai untuk *input* ADC *Teensy 3.2*. yang memiliki tegangan referensi pada tegangan 3.3V.

4.2. Pengujian Software

4.2.1. Pengujian FFT

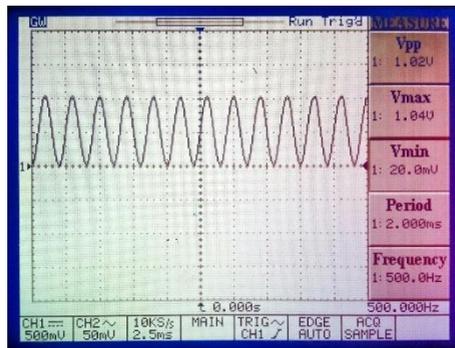
4.2.1.1. Pengujian dengan input function generator

Pengujian FFT dilaksanakan dengan menjalankan program FFT pada *Teensy 3.2*. Program yang digunakan untuk pengujian ini dapat tersambung dengan sebuah *tool software* pada PC untuk keperluan analisa yang bernama *spectrogram*. Tujuan pengujian ini adalah untuk memverifikasi algoritma FFT yang digunakan. Pelaksanaan pengujian ini dengan menyusun *hardware* sesuai skematik pada gambar 4.6.

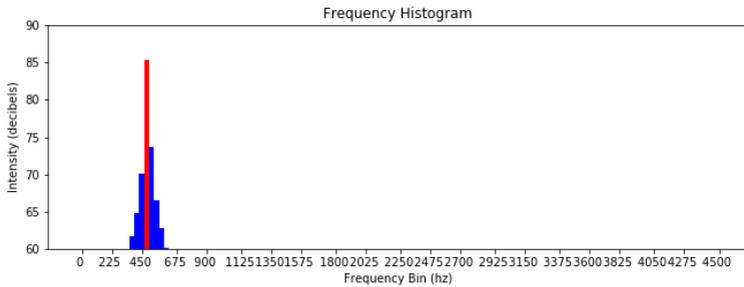
Teensy 3.2 dihubungkan ke *function generator* untuk mendapat sinyal *input* dan juga dihubungkan ke PC untuk menganalisa *output* dari FFT. Pengujian ini menerapkan parameter untuk FFT size sebesar 256, dan *sample rate* 9000Hz. Frekuensi yang diset pada *function generator* adalah 500Hz dan amplitudo 1Vpp dengan bentuk gelombang yang diubah-ubah. Pengujian pertama adalah gelombang sinusoidal. Hasil pengujian tersebut ditunjukkan pada gambar 4.4 sampai gambar 4.5.



Gambar 4.6. Skematik pengujian FFT



Gambar 4.7. Pengujian FFT dengan *input* sinyal sinusoidal 500Hz (domain waktu)

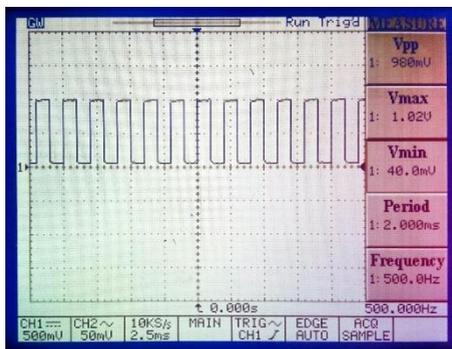


Gambar 4.8. Pengujian FFT dengan *input* sinyal sinusoidal 500Hz (domain frekuensi)

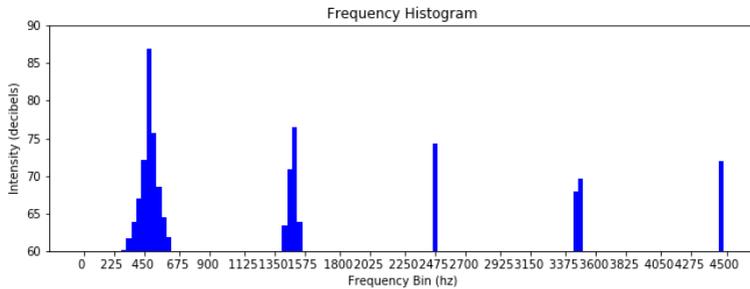
Hasil tersebut menunjukkan bahwa sinyal sinus murni hanya memiliki 1 spektrum frekuensi. Spektrum lain di sekitar spektrum dominan merupakan derau dari sumber sinyal. Di sisi lain juga memverifikasi kebenaran algoritma FFT yang digunakan karena letak spektrum frekuensinya menunjukkan frekuensi 500Hz.

Pengujian selanjutnya adalah dengan sinyal persegi dengan frekuensi yang sama dan dihasilkan bentuk spektrum yang ditunjukkan pada gambar 4.9 sampai gambar 4.10.

Hasil tersebut menunjukkan bahwa sinyal persegi memiliki banyak frekuensi penyusun. Namun tetap memiliki frekuensi dasar pada 500Hz terlihat dari spektrum yang paling dominan. Kemunculan spektrum lain adalah secara periodik.



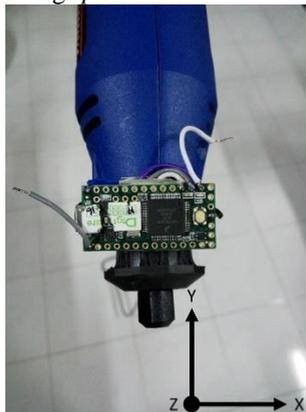
Gambar 4.9. Pengujian FFT dengan *input* sinyal kotak 500Hz (domain waktu)



Gambar 4.10. Pengujian FFT dengan *input* sinyal kotak 500Hz (domain frekuensi)

4.2.1.2. Pengujian dengan *input* sensor

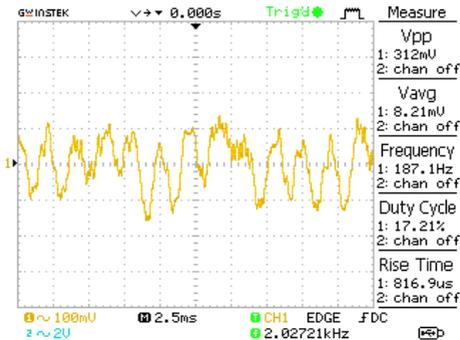
Pengujian ini dilakukan untuk mengetahui performa FFT jika diberi *input* sinyal dari sensor *accelerometer* ADXL335. ADXL335 memiliki 3 *axis* dan masing-masing *axis* memiliki satu pin *output*. Pengujian ini dilakukan pada masing-masing *axis* untuk mengetahui *axis* terbaik dalam hal pengukuran getaran *spindle*. ADXL335 dan *Teensy* 3.2 dipasang pada *spindle* dengan posisi ADXL335 menempel pada permukaan *spindle* seperti ditunjukkan pada gambar 4.11. Sumbu X memiliki arah tegak lurus dengan bidang *spindle*, sumbu Y sejajar dengan bidang *spindle*, dan sumbu Z menembus bidang *spindle*.



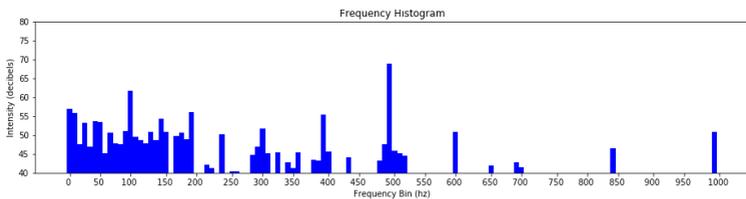
Gambar 4.11. Pemasangan ADXL335 dan *Teensy* 3.2 pada *spindle*

Kemudian *spindle* tersebut dinyalakan pada kecepatan 4 dan digerakkan dengan alur tertentu di atas permukaan PCB dengan dan tanpa menggosok. Pengujian ini dilakukan untuk masing-masing *axis* serta diulangi 10 kali setiap *axis*. Dari pengujian tersebut didapat data yang ditunjukkan oleh gambar 4.12 hingga gambar 4.23.

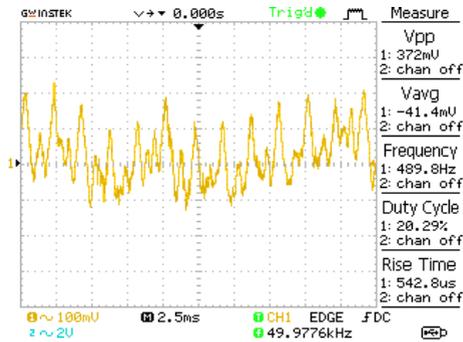
Gambar 4.12 sampai 4.15 merupakan gambar pengujian pada sumbu X dengan dan tanpa menggosok serta masing-masing sinyal pada domain waktu dan frekuensi. Jika antara menggosok dan tidak menggosok dibandingkan secara visual, maka pada domain waktu gelombang sinyal tampak lebih rapat atau bisa disebut memiliki frekuensi penyusun dominan yang lebih tinggi pada kondisi menggosok. Jika ditinjau pada spektrum frekuensinya akan nampak dengan jelas frekuensi penyusunnya. Dalam hal ini pada kondisi menggosok spektrum penyusunnya lebih banyak dan frekuensi dominannya berada pada dekat 1000Hz sedangkan pada kondisi tidak menggosok pada 500Hz.



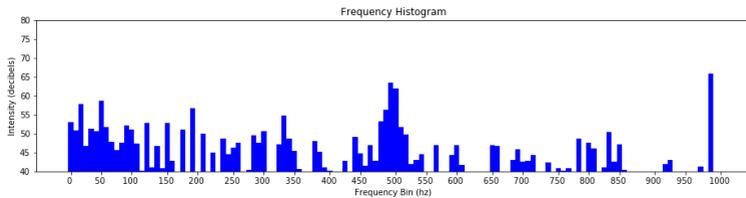
Gambar 4.12. Pengujian FFT dengan sensor pada X-axis tanpa menggosok (domain waktu)



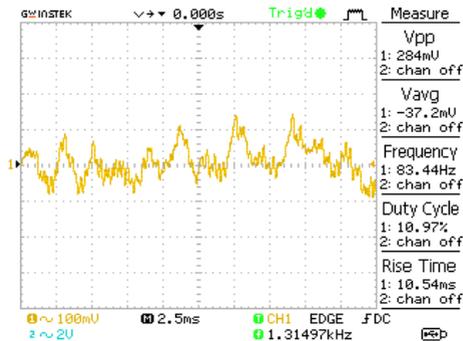
Gambar 4.13. Pengujian FFT dengan sensor pada X-axis tanpa menggosok (domain frekuensi)



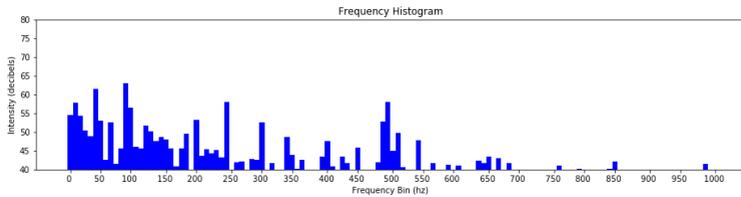
Gambar 4.14. Pengujian FFT dengan sensor pada X-axis dengan menggores (domain waktu)



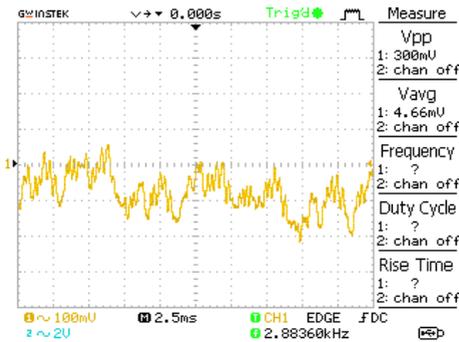
Gambar 4.15. Pengujian FFT dengan sensor pada X-axis dengan menggores (domain frekuensi)



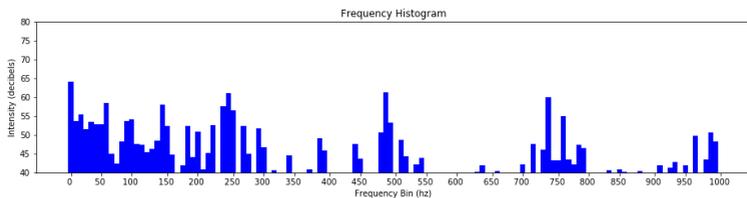
Gambar 4.16. Pengujian FFT dengan sensor pada Y-axis tanpa menggores (domain waktu)



Gambar 4.17. Pengujian FFT dengan sensor pada Y-axis tanpa menggores (domain frekuensi)

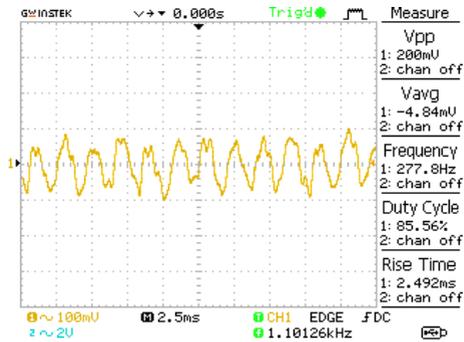


Gambar 4.18. Pengujian FFT dengan sensor pada Y-axis dengan menggores (domain waktu)

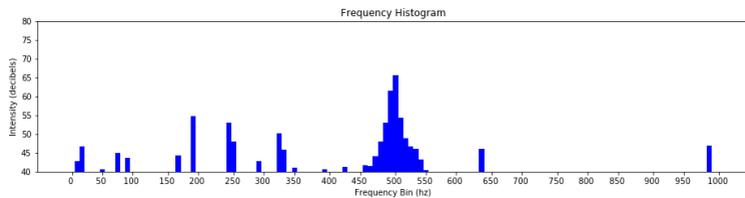


Gambar 4.19. Pengujian FFT dengan sensor pada Y-axis dengan menggores (domain frekuensi)

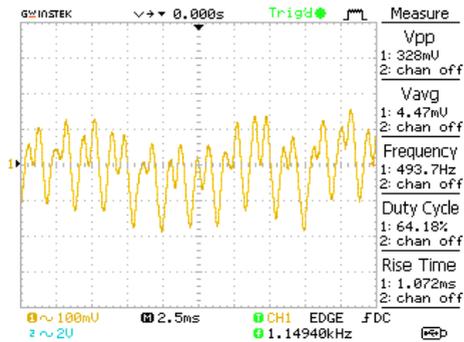
Gambar 4.16 sampai 4.19 merupakan hasil pengujian pada sumbu Y. Dapat diketahui pada sumbu Y hasil pembacaan sensor terhadap getaran tidak terlalu baik karena magnitudo yang rendah dan tidak nampak spektrum dominan. Hal ini disebabkan oleh arah getaran yang tidak sesuai dengan sumbu Y.



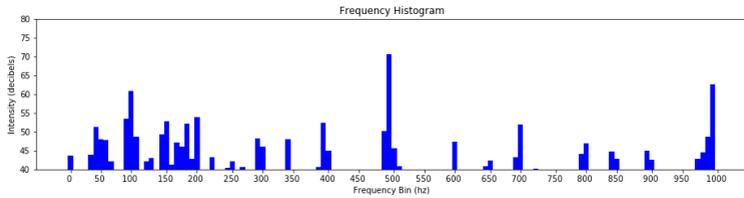
Gambar 4.20. Pengujian FFT dengan sensor pada Z-axis tanpa mengores (domain waktu)



Gambar 4.21. Pengujian FFT dengan sensor pada Z-axis tanpa mengores (domain frekuensi)



Gambar 4.22. Pengujian FFT dengan sensor pada Z-axis dengan mengores (domain waktu)



Gambar 4.23. Pengujian FFT dengan sensor pada Z-axis dengan menggores (domain frekuensi)

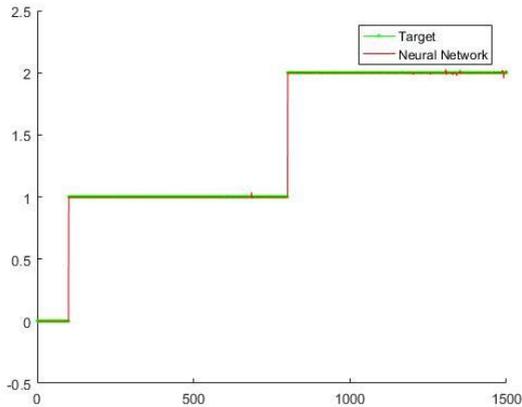
Gambar 4.20 hingga 4.23 adalah hasil pengujian pada sumbu Z. Hasil ini menunjukkan pembacaan sensor yang baik ditinjau dari magnitudo untuk spektrum dominan cukup tinggi dan adanya perbedaan yang berarti antara kondisi menggerus atau tidak. Jika dibandingkan dengan sumbu X, sumbu Z memiliki hasil yang lebih rapi.

Dari data pengujian di atas dapat diperhatikan bahwa perbedaan pola spektrum frekuensi antara tanpa menggores dan dengan menggores di antara ketiga *axis* tersebut paling kentara pada Z-axis atau sumbu Z. Sehingga dapat disimpulkan bahwa dengan konfigurasi pemasangan sensor *accelerometer* seperti itu *axis* terbaik adalah Z-axis. Hal tersebut berhubungan dengan arah getaran pada *spindle* yang bergerak secara sentrifugal.

4.2.2. Pengujian ANN secara *offline*

4.2.2.1. Pengambilan data untuk *learning*

Pengambilan data pada tahap ini adalah untuk keperluan *learning* atau pembelajaran pada *Artificial Neural Network*. Pada tahap ini pemrosesan FFT berada di *Teensy 3.2* yang selanjutnya data *output*-nya dikirim via serial ke PC untuk ditampung. Data-data tersebut kemudian diolah untuk diubah ke bentuk tegangan dan dinormalisasi. Hasil akhirnya kemudian diumpankan ke MATLAB untuk dipelajari. *Output* dari MATLAB pada tahap ini adalah nilai *weight* dan bias. Nilai-nilai tersebut tersimpan pada satu *file* variabel MATLAB dan siap untuk diuji secara *offline*. Sumber data untuk pengujian ini adalah sisa data pada proses pengambilan data yang dikurangi sejumlah data untuk proses pembelajaran. Sehingga total didapat 1500 data untuk semua kondisi dan 35 data untuk pengujian *offline*. Proses pengambilan data ditunjukkan pada gambar 4.24.



Gambar 4.26. Hasil *learning* ANN di MATLAB

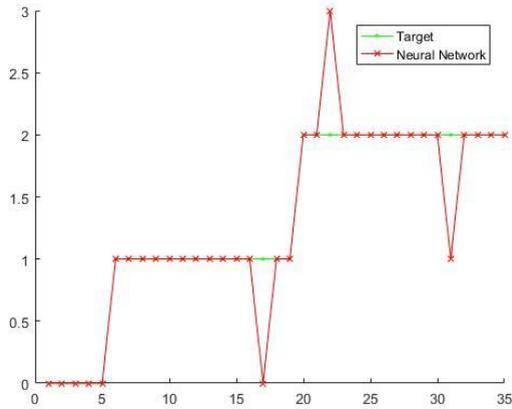
4.2.2.2. Proses *learning*

Proses *learning* atau pembelajaran ANN dilakukan di MATLAB untuk mempercepat proses. Proses ini melibatkan *Training tool* MATLAB untuk ANN. Pada proses *learning* ini nilai *error* diset dengan nilai *error* terbesarnya adalah 10^{-5} . Hasil verifikasi yang diperoleh pada tahap ini ditunjukkan pada gambar 4.26.

Dari hasil pada gambar 4.26 dapat diketahui bahwa selisih antara target dan hasil pembelajaran *Artificial Neural Network* sangat kecil dengan *error* 6.25×10^{-6} sehingga hasil *Artificial Neural Network* sudah sangat mendekati nilai target.

4.2.2.3. Proses pengujian secara *offline* pada MATLAB

Proses selanjutnya yaitu menguji kebenaran dari proses pembelajaran sebelumnya. Nilai pengambilan data yang tidak diikuti dalam proses pembelajaran diambil secara acak 35 data. Proses ini dilakukan dengan menjalankan algoritma *feedforward* dengan nilai bobot dan bias dari hasil pembelajaran sebelumnya pada program MATLAB. Dengan membandingkan antara hasil ANN dan target, grafik selanjutnya di-plot dan ditunjukkan pada gambar 4.27.

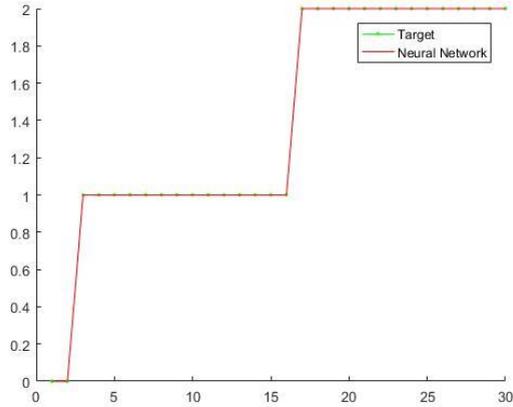


Gambar 4.27. Grafik pengujian ANN secara *offline* pada MATLAB

Dari gambar 4.27 dapat dilihat adanya perbedaan antara target dan hasil *Artificial Neural Network* hanya terjadi pada 3 titik dari 35 data uji. Sehingga hasil pengujian ANN secara *offline* diperoleh keberhasilan $\frac{35-3}{35} \times 100\% \approx 91,4\%$. Hasil ini menjadi acuan untuk menerapkan nilai *weight* dan *bias* pada *Teensy 3.2* agar dapat menjalankan pengujian secara keseluruhan dengan tingkat keberhasilan yang tinggi.

4.2.2.4. Proses pengujian secara *offline* pada *Teensy 3.2*

Proses pengujian secara *offline* pada *Teensy 3.2* bertujuan untuk memverifikasi algoritma *Artificial Neural Network* yang diterapkan pada *Teensy 3.2*. Proses ini dilakukan dengan menerapkan algoritma *feedforward* dan nilai *Weight* serta *Bias* pada *Teensy 3.2*. Untuk mendapat nilai yang pasti, *input* ANN yang digunakan adalah nilai Training yang dipilih secara acak. Data acak tersebut dipilih 30 data. Antara hasil pemrosesan ANN dan target yang ditetapkan dikirimkan ke PC untuk ditampung dan selanjutnya di-plot menggunakan MATLAB. Hasil pengujian ini ditunjukkan pada grafik yang disajikan pada gambar 4.28.

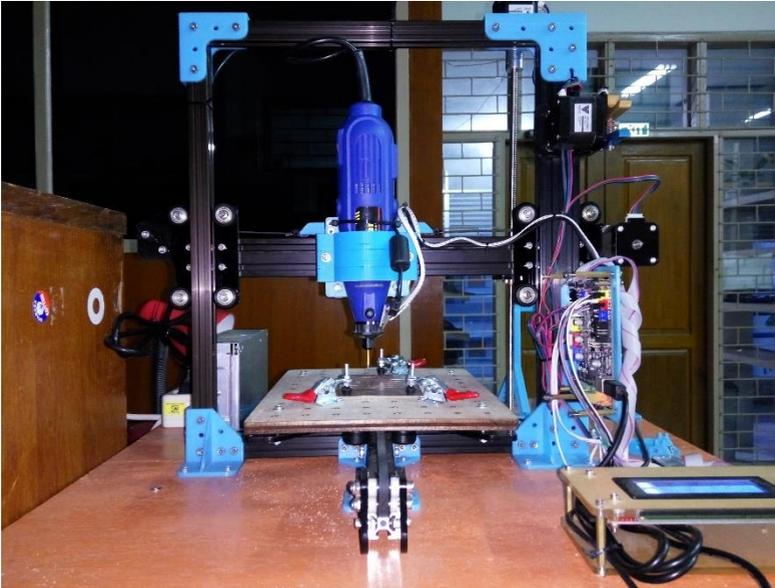


Gambar 4.28. Grafik pengujian ANN secara *offline* pada *Teensy 3.2*

Gambar 4.28 menunjukkan hasil pengujian ANN secara offline yang dilakukan pada *Teensy 3.2*. Berdasar grafik tersebut dapat disimpulkan bahwa algoritma yang diterapkan pada *Teensy 3.2* sudah benar dan siap untuk memproses *input* dari FFT secara langsung untuk pengujian sistem secara keseluruhan.

4.3. Pengujian sistem secara keseluruhan

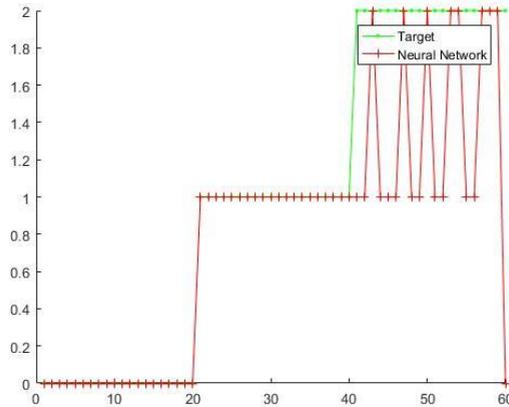
Pengujian sistem secara keseluruhan dilakukan dengan menerapkan algoritma ANN pada *Teensy 3.2*. Pada pengujian ini, seluruh pemrosesan sinyal sampai bisa dikenali menjadi patah atau tidaknya mata bor dikerjakan pada *Teensy 3.2*. PC dalam bagian ini hanya sebagai monitor *output* serta sebagai pengendali jalannya *printer PCB*. Pengujian ini menerapkan beberapa variasi kondisi. Ilustrasi sistem secara keseluruhan ditampilkan pada gambar 4.29. Sensor *accelerometer* dipasang menempel pada *spindle* dan *Teensy 3.2* dipasang menumpuk pada sensor *accelerometer*. Tujuan pemasangan seperti ini adalah agar sensor dapat lebih peka terhadap getaran *spindle* dan penyaluran sinyal ke *Teensy 3.2* tidak terganggu atau terdistorsi mengingat sinyal yang dikirimkan merupakan sinyal analog yang rentan terhadap derau.



Gambar 4.29. Sistem secara keseluruhan

4.3.1. Pengujian sistem dengan satu kecepatan putaran *spindle*

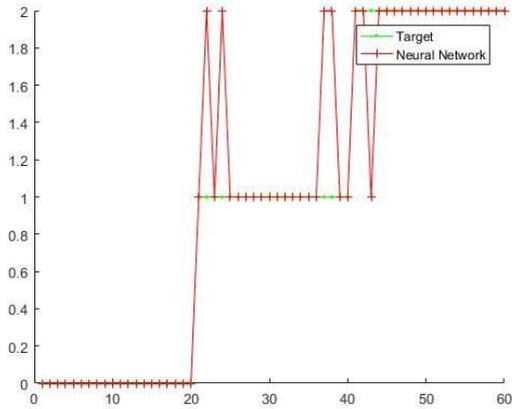
Pengujian sistem dengan satu kecepatan putaran *spindle* dijalankan dengan melakukan *learning* ANN untuk satu kecepatan tertentu. Selanjutnya diimplementasikan pada *Teensy 3.2* untuk diukur tingkat keberhasilannya. Hasil dari pengujian ini ditunjukkan pada grafik gambar 4.30. Dari grafik tersebut diketahui bahwa dari pengambilan data sebanyak 20 data masing-masing kondisi, kesalahan pendeteksian yang terjadi adalah sebanyak 12 titik. sehingga *error* yang terjadi sebesar 20% atau tingkat keberhasilannya 80%. *Error* yang timbul hanya berada pada kondisi 2 yaitu kondisi saat *spindle* berputar dan menggores. Hal ini diduga disebabkan oleh keberagaman kondisi saat proses *learning* / pembelajaran ANN.



Gambar 4.30. Grafik pengujian sistem dengan satu kecepatan *spindle*

4.3.2. Pengujian sistem dengan berbagai kecepatan putaran *spindle*

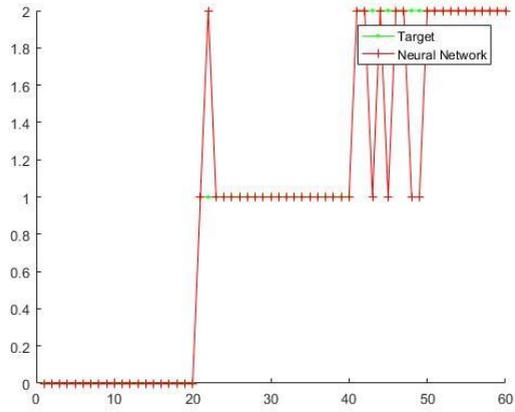
Pengujian sistem dengan berbagai kecepatan putaran *spindle* dilakukan dengan melakukan proses pembelajaran ANN menggunakan data FFT pada tujuh kecepatan putaran *spindle*. Masing-masing kecepatan dijalankan dua kondisi yaitu menggores dan tidak menggores dengan gerakan mesin CNC yang sama. Hasil bobot dan bias yang didapat kemudian diimplementasikan pada *Teensy 3.2*. Algoritma *Feedforward* untuk ANN diterapkan juga. Hasil penghitungan ANN dikirimkan melalui koneksi serial untuk selanjutnya ditampilkan pada PC menggunakan *software*. Data-data tersebut kemudian di-plot menggunakan MATLAB untuk dapat menyajikan data yang lebih baik. Hasil pengujian ditampilkan pada gambar 4.31. dari grafik tersebut diketahui bahwa pengujian ini memiliki keberhasilan 91,67%. Hal ini diperoleh karena variasi pembelajaran lebih banyak dibandingkan dengan satu kecepatan.



Gambar 4.31. Grafik pengujian sistem dengan banyak kecepatan *spindle*

4.3.3. Pengujian sistem dengan perbedaan jenis papan tembaga (PCB)

Pengujian sistem dengan menerapkan 2 jenis papan tembaga ini dilakukan untuk melihat karakter getaran terhadap jenis papan tembaga yang dicetak. Kedua jenis papan tembaga dibelajarkan pada ANN dengan satu kecepatan putaran *spindle*. Hasil bobot dan bias dari proses *learning Artificial Neural Network* kemudian diterapkan pada *Teensy 3.2*. Hasil pengenalan ANN dikirimkan ke PC melalui serial dan ditampilkan pada *software*. Selanjutnya di-plot menggunakan MATLAB. Hasil pengujian ini ditampilkan pada gambar 4.32. Tingkat keberhasilan pengujian ini adalah 91,67%. Keberhasilan yang tinggi pada pengujian ini diperoleh dari tingkat kekerasan pada papan PCB jenis fiber lebih keras daripada PCB biasa sehingga perbedaan getaran antara menggerus dan tidak dapat lebih dirasakan sensor *accelerometer*.



Gambar 4. 32. Grafik pengujian sistem dengan material PCB Fiber

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1. Kesimpulan

Pada penelitian ini telah dibuat suatu sistem pendeteksian kepatahan mata bor berdasarkan analisa getaran. Sensor yang digunakan untuk dapat mendeteksi getaran adalah MEMS *accelerometer* yang mempunyai kemampuan pengukuran sampai 3,6g. Pendeteksian kepatahan mata bor pada pencetak PCB dapat diamati menggunakan spektrum getaran. Untuk dapat mengenali kepatahan mata bor, perlu dilakukan pengolahan sinyal dengan menggunakan *Fast Fourier Transform*. Lebar spektrum frekuensi yang digunakan adalah 0-1000Hz. Pola spektrum frekuensi antara 0-1000Hz digunakan sebagai *input Artificial Neural Network* dengan 1 *hidden layer* untuk dapat mengenali kepatahan mata bor. Pemrosesan *Fast Fourier Transform* dan *Artificial Neural Network* dilakukan pada *Teensy 3.2*. Pola getaran saat menggores adalah munculnya frekuensi harmonik salah satunya pada frekuensi 1000Hz. Tingkat keberhasilan ANN pada pengujian *offline* sebesar 91,4% dan pada pengujian *online* sebesar 80%. Dengan memperbanyak data pembelajaran ANN, keberhasilan pengujian dapat meningkat menjadi 91,67%. Peningkatan keberhasilan juga dapat dicapai pada jenis PCB fiber dengan keberhasilan 91,67%. Dengan menggunakan sistem pendeteksian ini, kepatahan mata bor dapat diketahui lebih awal sehingga dapat meningkatkan efisiensi penggunaan pencetak PCB berbasis CNC.

5.2. Saran

Pada penelitian selanjutnya sebaiknya menggunakan mesin CNC yang standar industri agar lebih dekat dengan penerapan di lapangan. Pemilihan *spindle* juga penting untuk kestabilan putaran yang juga berpengaruh pada getaran yang ditimbulkan.

Halaman ini sengaja dikosongkan

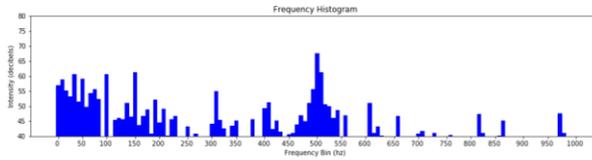
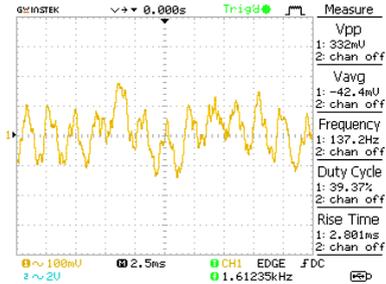
DAFTAR PUSTAKA

- [1] H. S. Cho, J.-h. Han, S.-y. Chi dan K.-H. Yoo, "A tool breakage detection system using load signals of spindle motors in CNC machines," *Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 160-163, 2016.
- [2] C.-R. Huang, M.-C. Lu, C.-E. Lu dan Y.-W. Hsu, "Study of spindle vibration signals for tool breakage monitoring in micro-drilling," *2011 9th World Congress on Intelligent Control and Automation*, vol. 8, pp. 1130-1134, 2011.
- [3] M. Ali, A. ELShaikh dan S. Babiker, "Controlling the CNC machine using microcontroller to manufacture PCB," *2016 Conference of Basic Sciences and Engineering Studies (SGCAC)*, vol. 3, pp. 116-120, 2016.
- [4] A. T. Purnomo, F. I. Hariadi dan A. Sasongko, "Development of Interface and Coordination for Control Module CNC PCB Milling Machine," *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 246-251, 2015.
- [5] D. Christian dan F. Hariadi, *Speed Control of Spindle Motor in The Designing Control Module of Computer Numerical Control (CNC) PCB Milling Machines*, Bandung: Teknik Elektro, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2014.
- [6] Muhammad Rivai, Tasripan dan Sumandri, "Pengenalan Pola Sinyal Suara Kerusakan Motor Listrik menggunakan Neural Network," *JAVA Journal of Electrical and Electronics Engineering*, vol. 5, pp. 51-55, 2007.

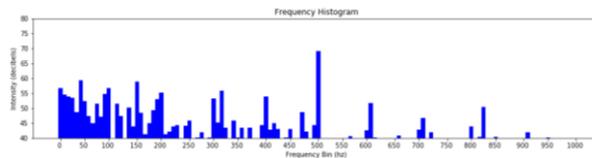
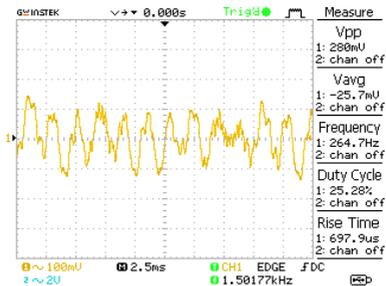
- [7] Muhammad Rivai, Tasripan dan Totok Mujiono, "Peningkatan Taraf Identifikasi Jenis Gas di Udara Terbuka menggunakan Transformasi Fourier dan Principal Component Analysis," *KURSOR*, vol. 6, pp. 103-110, 2011.
- [8] Muhammad Rivai, Totok Mujiono dan Djoko Purwanto, "Pengaruh Variasi Suhu Periodik Terhadap Selektivitas Gas Semikonduktor," *Industri*, vol. 8, no. 1, pp. 59-64, 2009.
- [9] Anonim, "Adafruit," Adafruit, [Online]. Available: <https://www.adafruit.com/product/2756>. [Diakses 23 Oktober 2016].
- [10] Anonim, "PJRC," PJRC, 22 Januari 2014. [Online]. Available: <https://www.pjrc.com/teensy/teensy31.html>. [Diakses 23 Oktober 2016].
- [11] A. Barkana, G. Cook dan E. S. McVey, "A Solid-State Relay," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, Vol. 1 dari 2 IECI-20, no. 2, pp. 97-99, 1973.
- [12] Anonim, "SeekIC," SeekIC.com Corp, 8 12 2011. [Online]. Available: http://xldz.en.seekic.com/product/integrated_circuits_ics/G3MC_202PL_VD_12VDC.html. [Diakses 18 Desember 2017].

LAMPIRAN

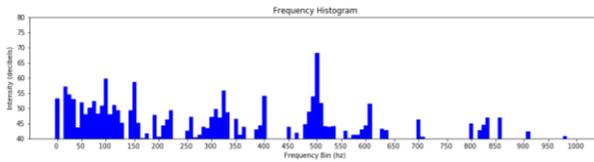
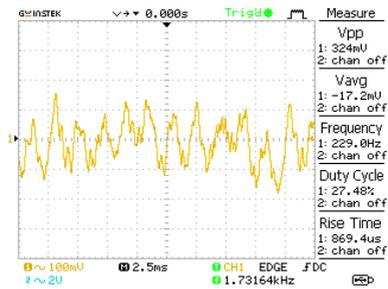
Pengujian sensor tanpa menggores pada sumbu X – Data 1



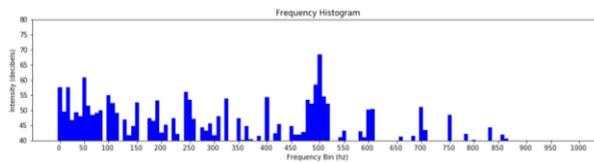
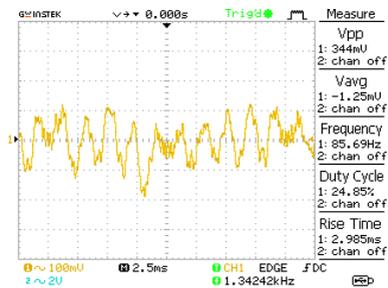
Pengujian sensor tanpa menggores pada sumbu X – Data 2



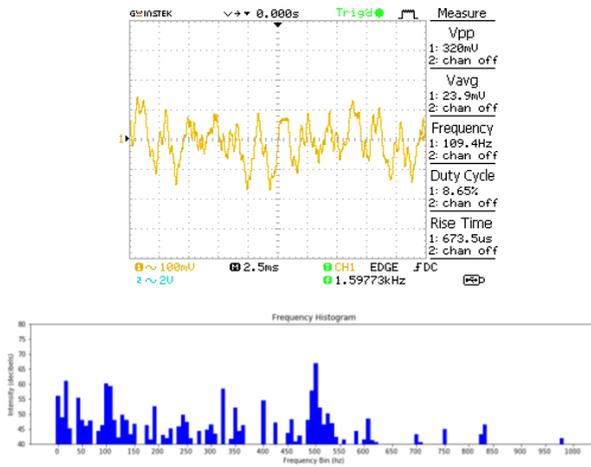
Pengujian sensor tanpa menggores pada sumbu X – Data 3



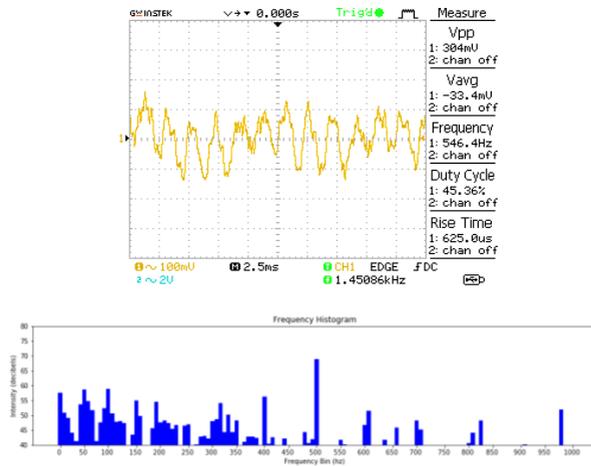
Pengujian sensor tanpa menggores pada sumbu X – Data 4



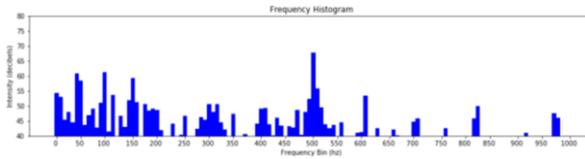
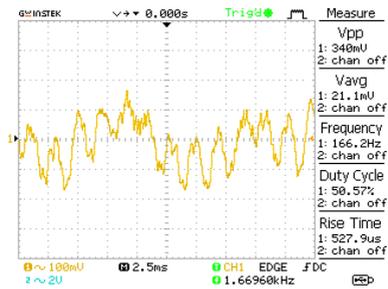
Pengujian sensor tanpa menggores pada sumbu X – Data 5



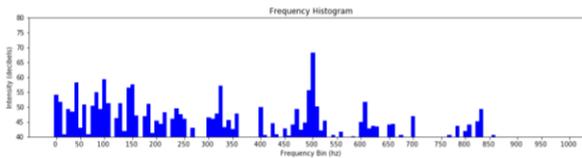
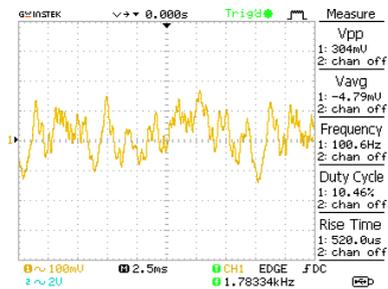
Pengujian sensor tanpa menggores pada sumbu X – Data 6



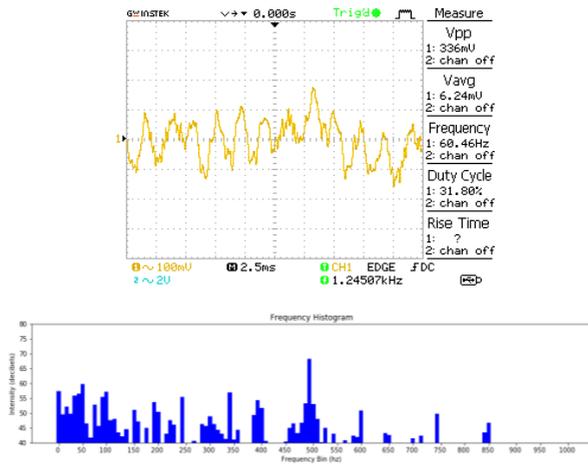
Pengujian sensor tanpa menggores pada sumbu X – Data 7



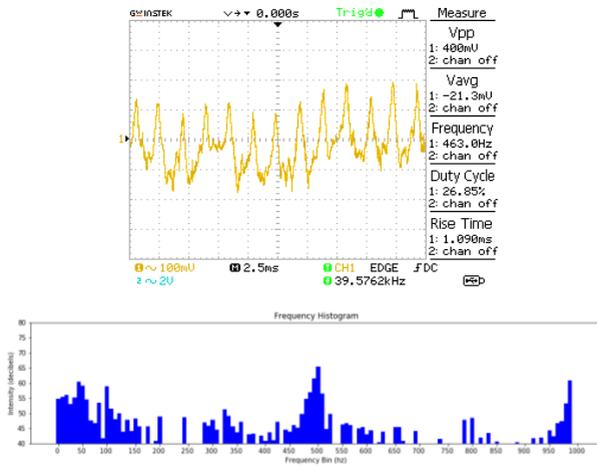
Pengujian sensor tanpa menggores pada sumbu X – Data 8



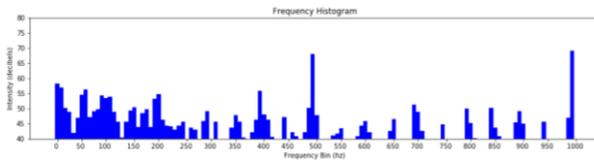
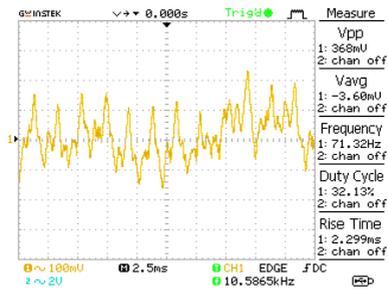
Pengujian sensor tanpa menggores pada sumbu X – Data 9



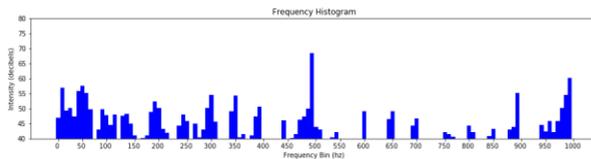
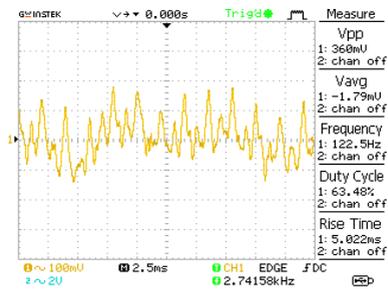
Pengujian sensor dengan menggores pada sumbu X – Data 1



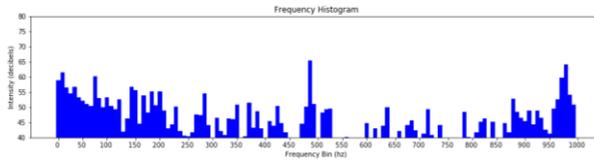
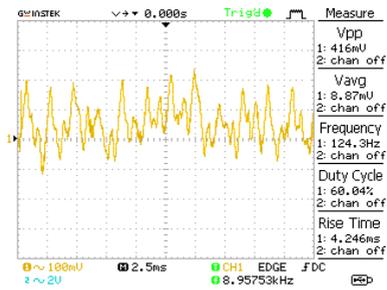
Pengujian sensor dengan menggores pada sumbu X – Data 2



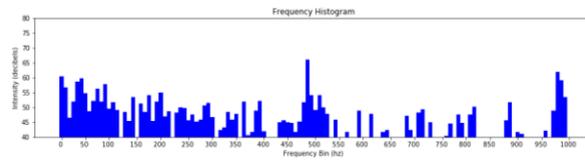
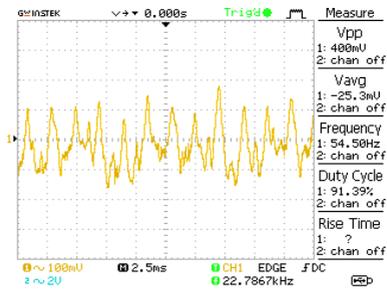
Pengujian sensor dengan menggores pada sumbu X – Data 3



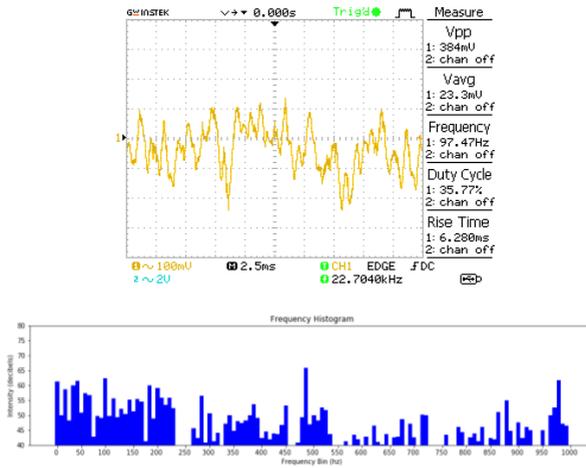
Pengujian sensor dengan menggores pada sumbu X – Data 4



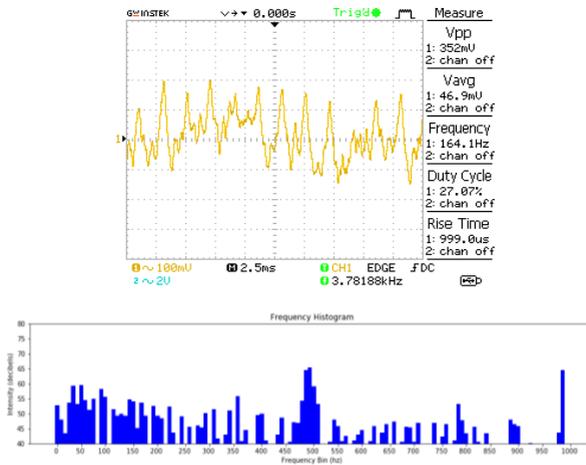
Pengujian sensor dengan menggores pada sumbu X – Data 5



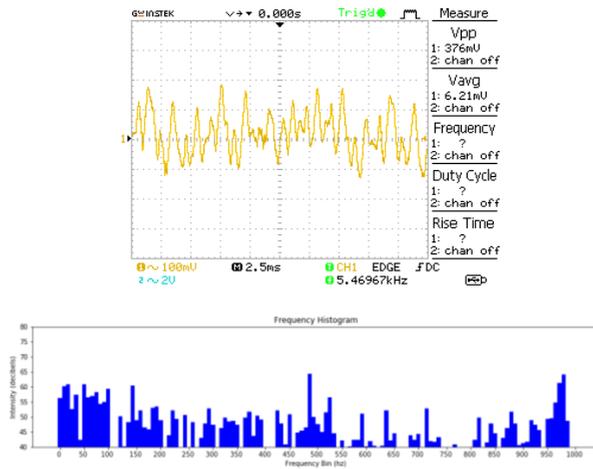
Pengujian sensor dengan menggores pada sumbu X – Data 6



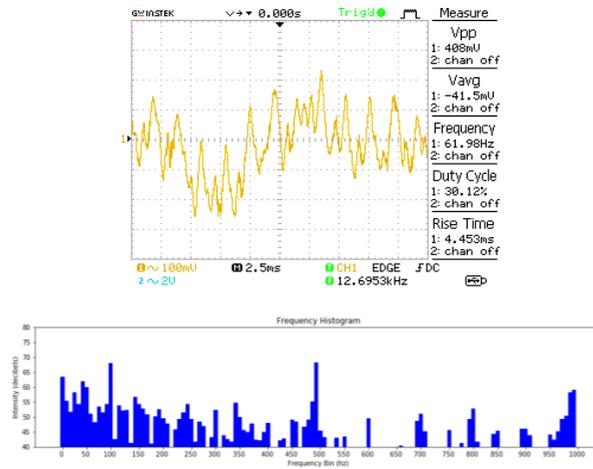
Pengujian sensor dengan menggores pada sumbu X – Data 7



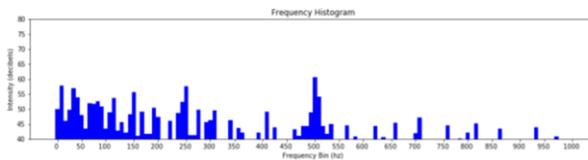
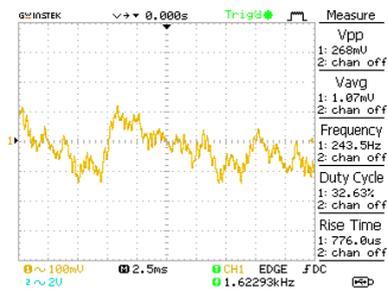
Pengujian sensor dengan menggores pada sumbu X – Data 8



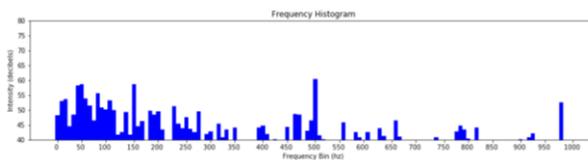
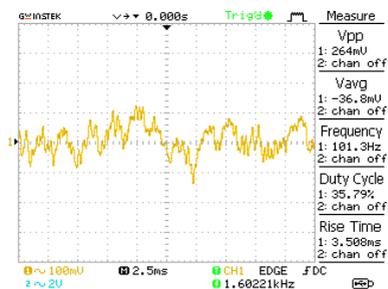
Pengujian sensor dengan menggores pada sumbu X – Data 9



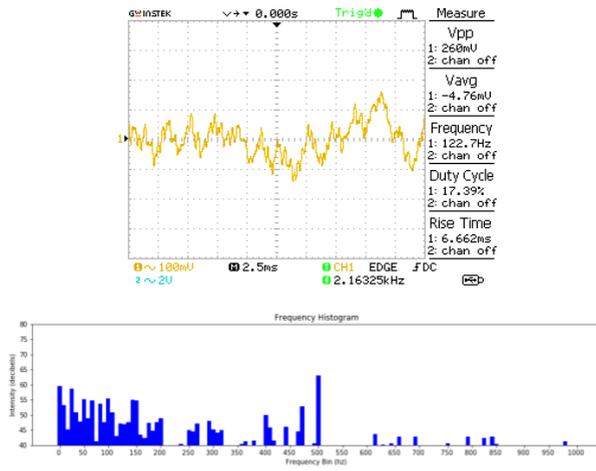
Pengujian sensor tanpa menggores pada sumbu Y – Data 1



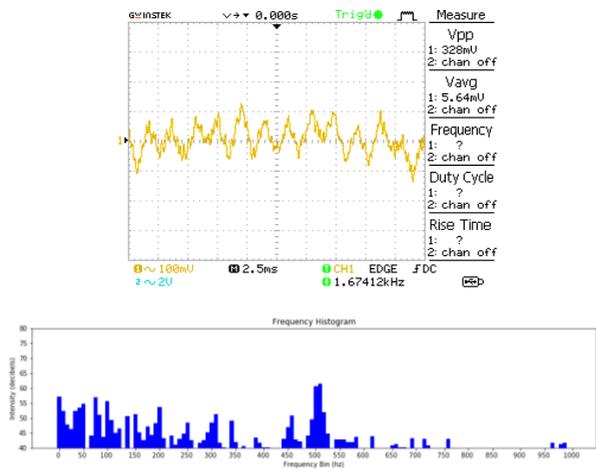
Pengujian sensor tanpa menggores pada sumbu Y – Data 2



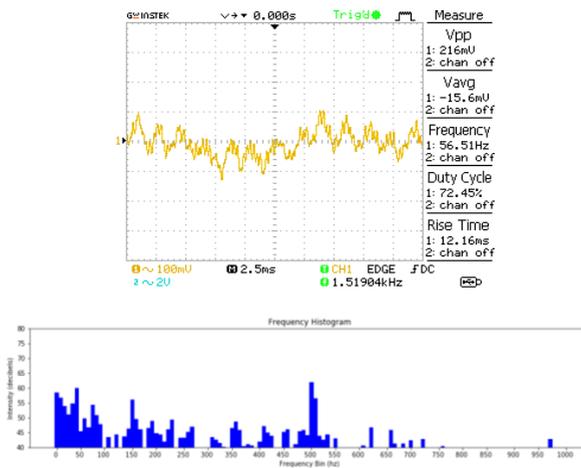
Pengujian sensor tanpa mengores pada sumbu Y – Data 3



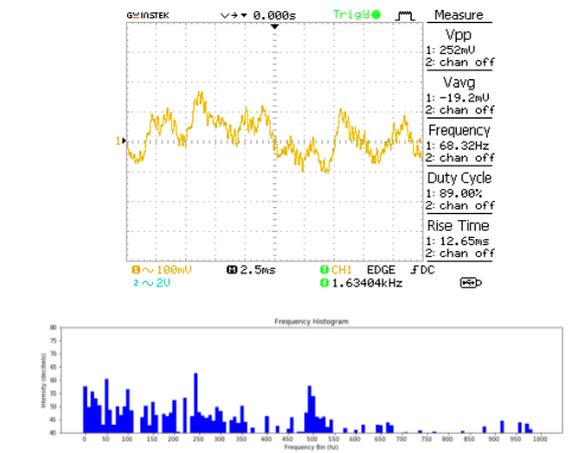
Pengujian sensor tanpa mengores pada sumbu Y – Data 4



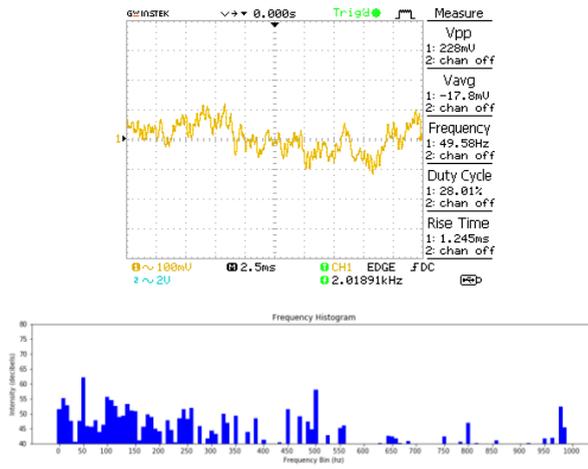
Pengujian sensor tanpa menggores pada sumbu Y – Data 5



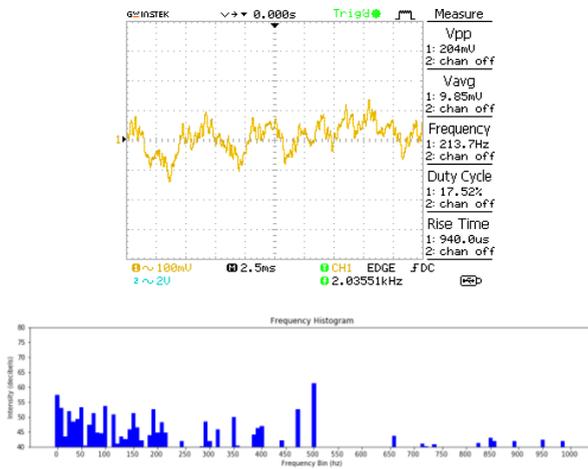
Pengujian sensor tanpa menggores pada sumbu Y – Data 6



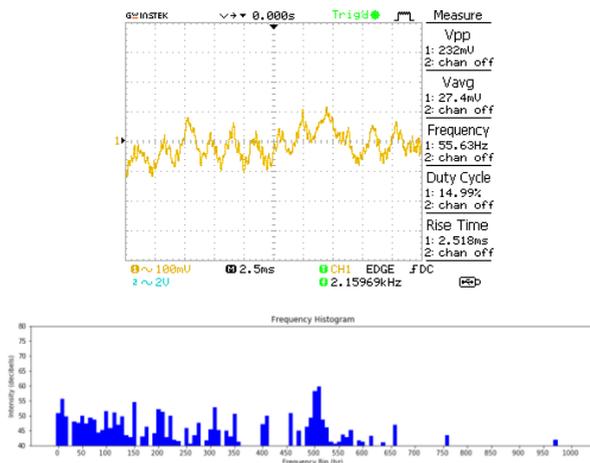
Pengujian sensor tanpa menggores pada sumbu Y – Data 7



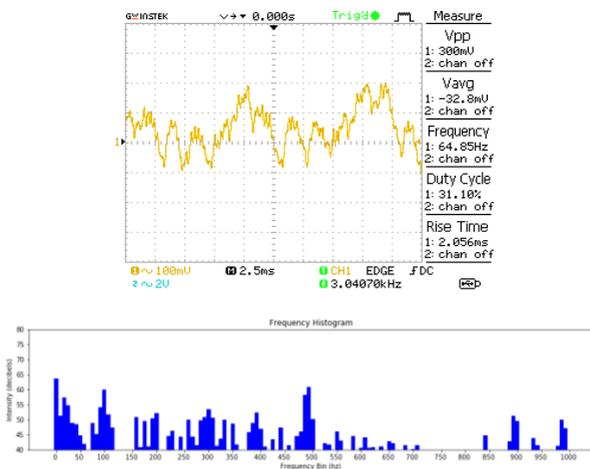
Pengujian sensor tanpa menggores pada sumbu Y – Data 8



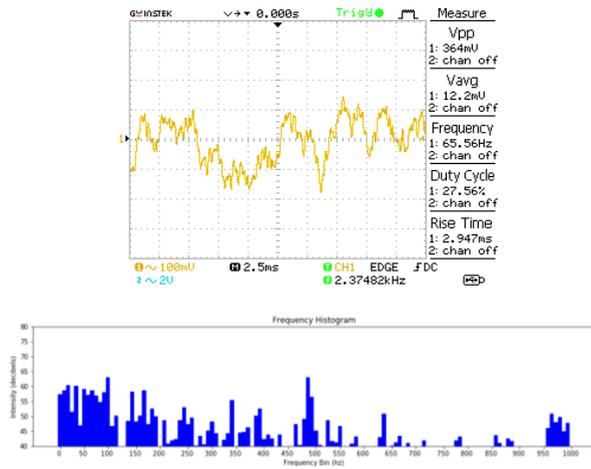
Pengujian sensor tanpa menggores pada sumbu Y – Data 9



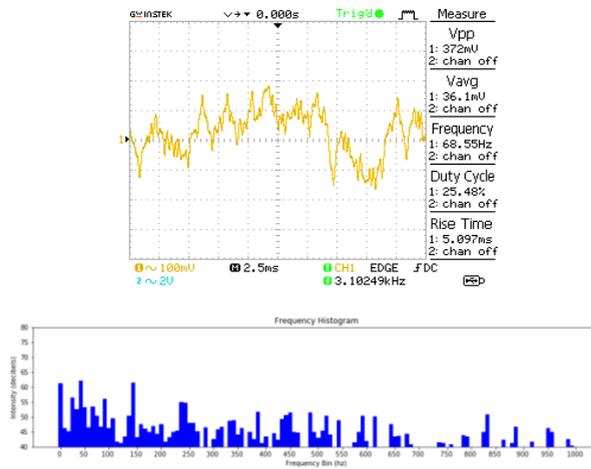
Pengujian sensor dengan menggores pada sumbu Y – Data 1



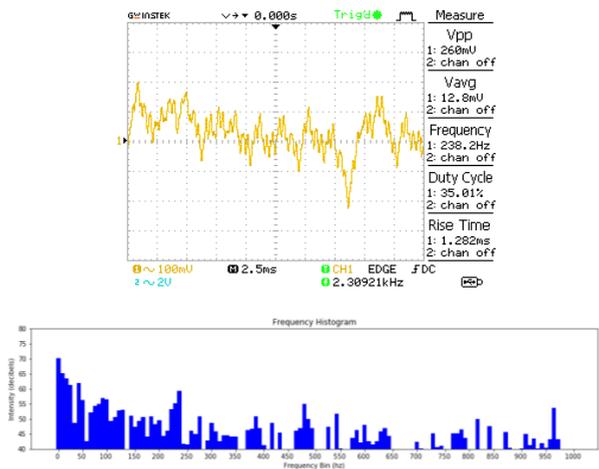
Pengujian sensor dengan menggores pada sumbu Y – Data 2



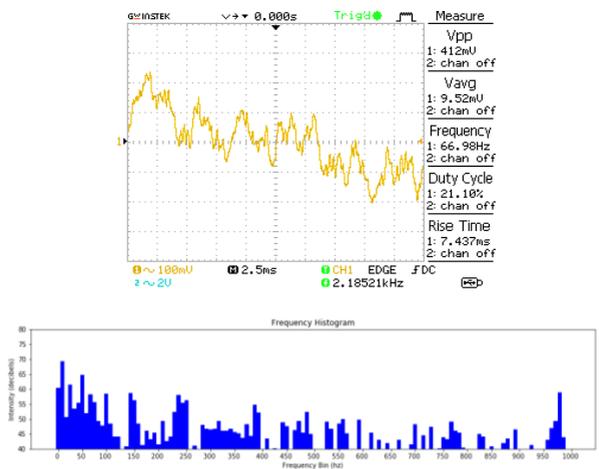
Pengujian sensor dengan menggores pada sumbu Y – Data 3



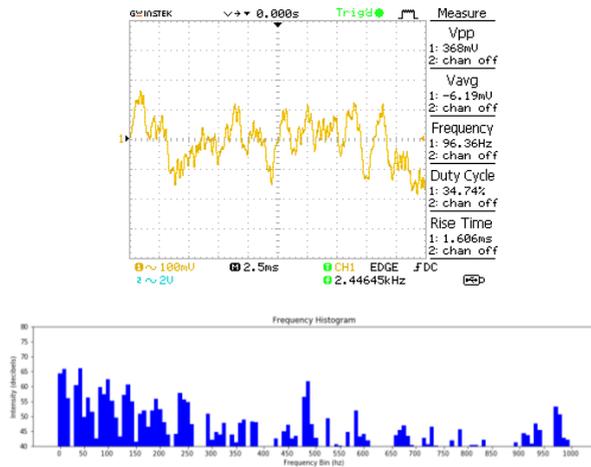
Pengujian sensor dengan menggores pada sumbu Y – Data 4



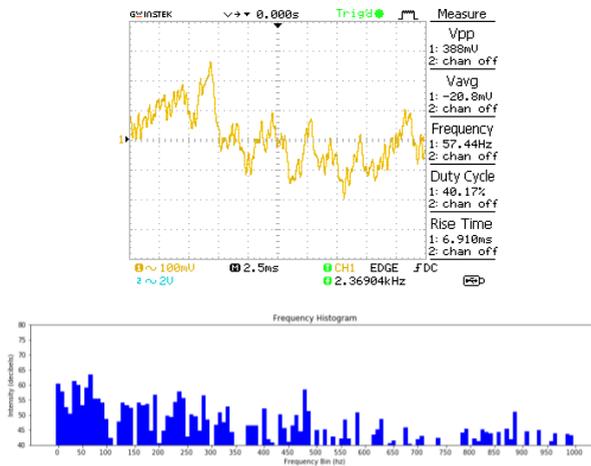
Pengujian sensor dengan menggores pada sumbu Y – Data 5



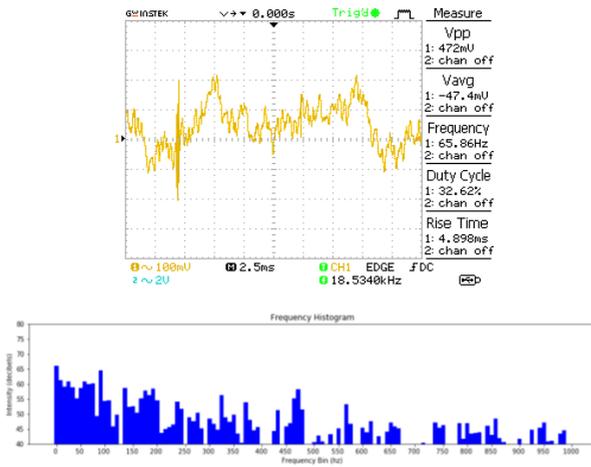
Pengujian sensor dengan menggores pada sumbu Y – Data 6



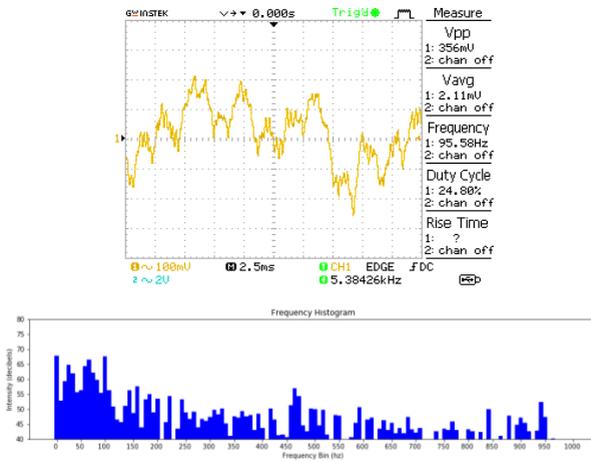
Pengujian sensor dengan menggores pada sumbu Y – Data 7



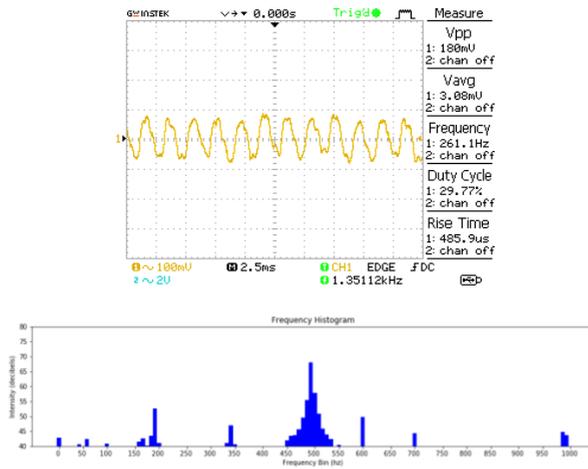
Pengujian sensor dengan menggores pada sumbu Y – Data 8



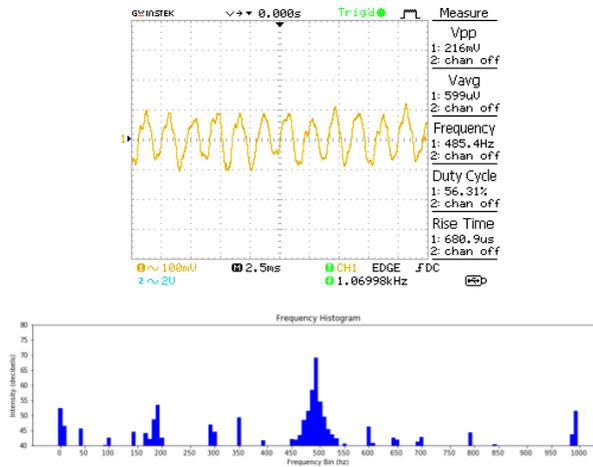
Pengujian sensor dengan menggores pada sumbu Y – Data 9



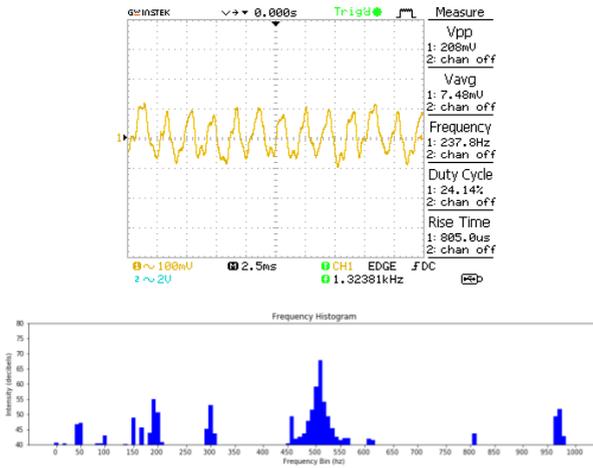
Pengujian sensor tanpa menggores pada sumbu Z – Data 1



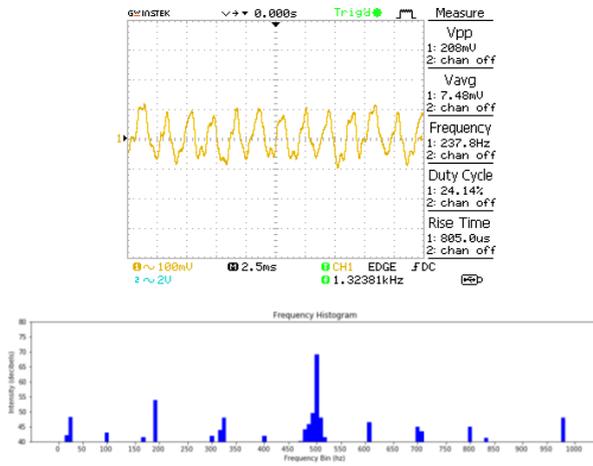
Pengujian sensor tanpa menggores pada sumbu Z – Data 2



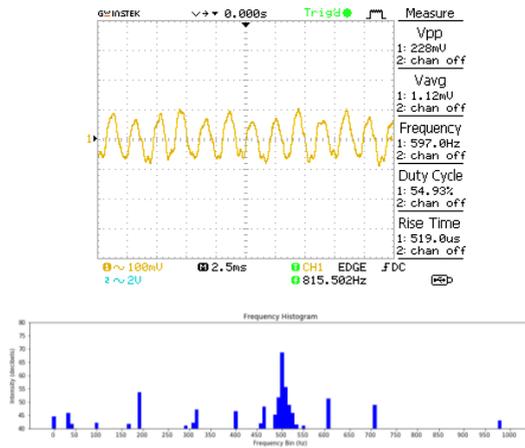
Pengujian sensor tanpa menggores pada sumbu Z – Data 3



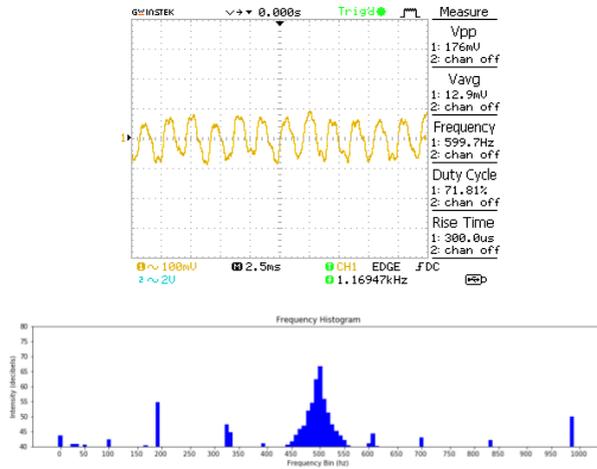
Pengujian sensor tanpa menggores pada sumbu Z – Data 4



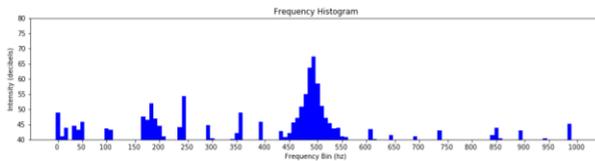
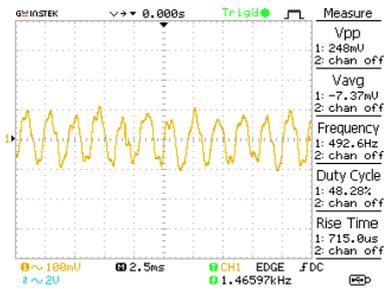
Pengujian sensor tanpa menggores pada sumbu Z – Data 5



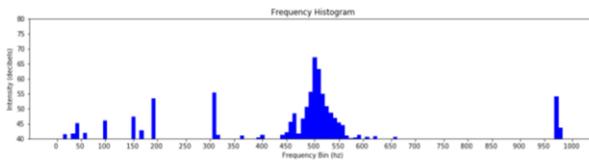
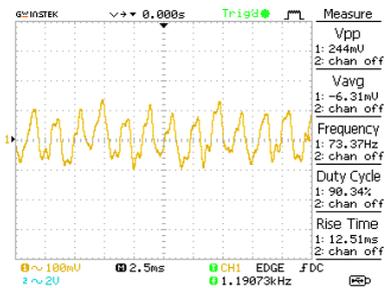
Pengujian sensor tanpa menggores pada sumbu Z – Data 6



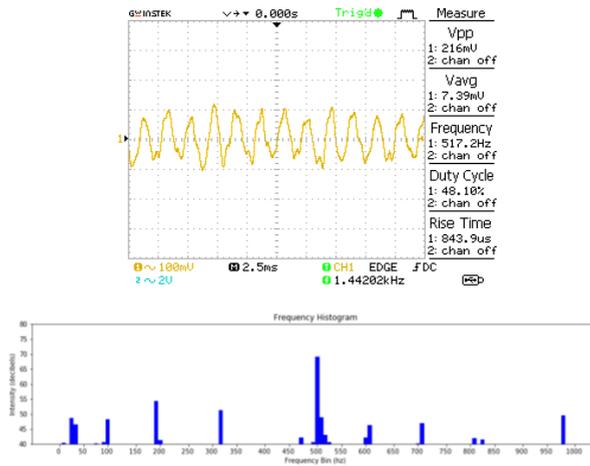
Pengujian sensor tanpa menggores pada sumbu Z – Data 7



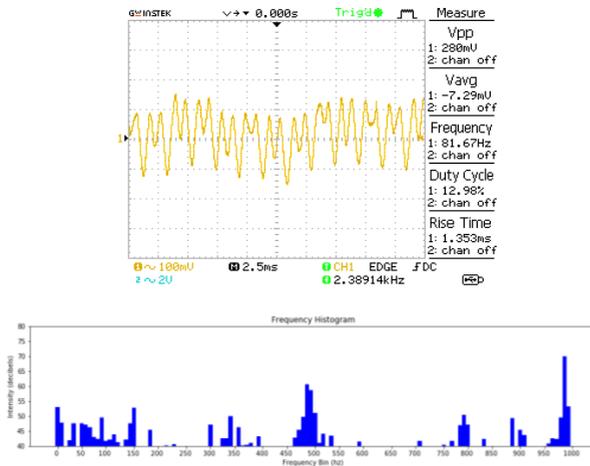
Pengujian sensor tanpa menggores pada sumbu Z – Data 8



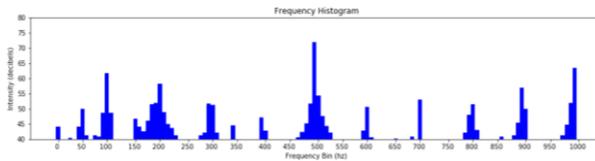
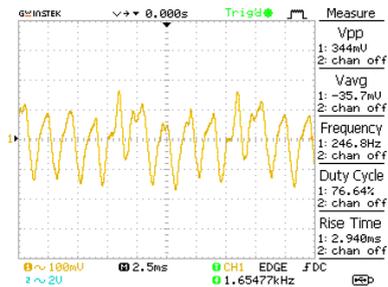
Pengujian sensor tanpa menggores pada sumbu Z – Data 9



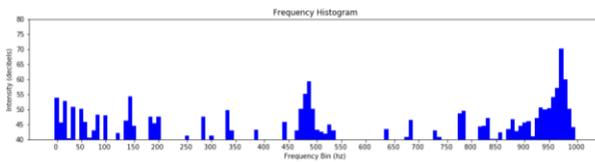
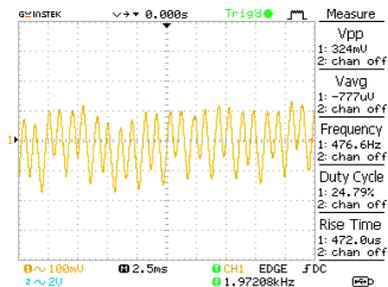
Pengujian sensor dengan menggores pada sumbu Z – Data 1



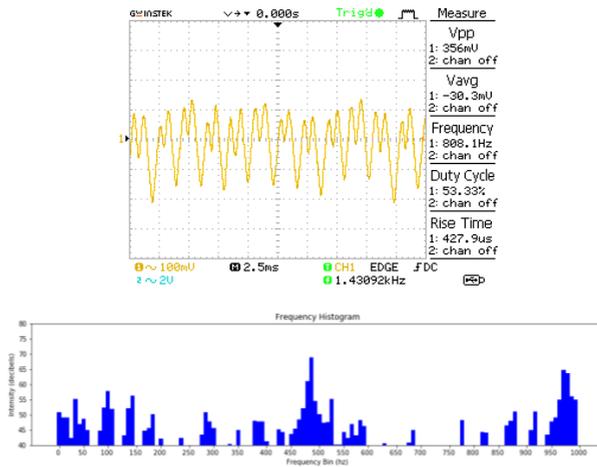
Pengujian sensor dengan menggores pada sumbu Z – Data 2



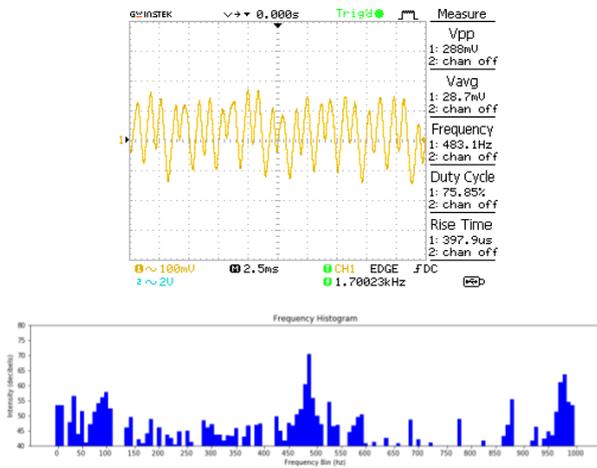
Pengujian sensor dengan menggores pada sumbu Z – Data 3



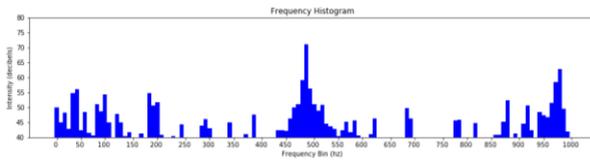
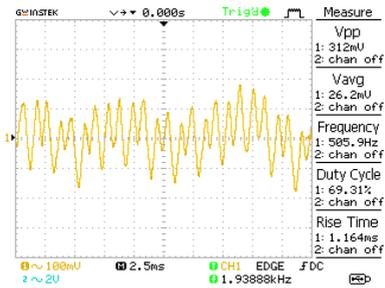
Pengujian sensor dengan menggores pada sumbu Z – Data 4



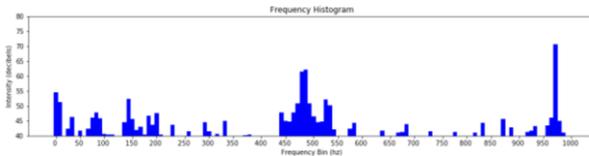
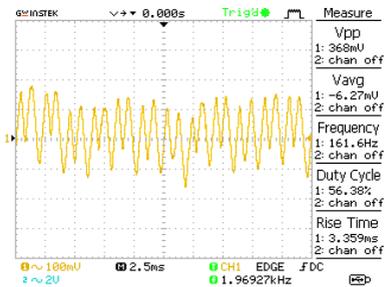
Pengujian sensor dengan menggores pada sumbu Z – Data 5



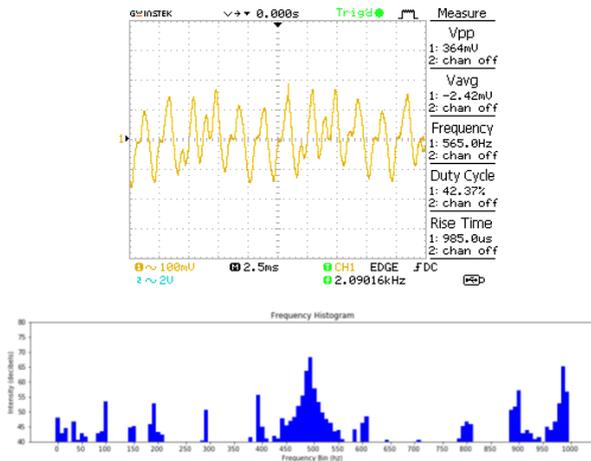
Pengujian sensor dengan menggores pada sumbu Z – Data 6



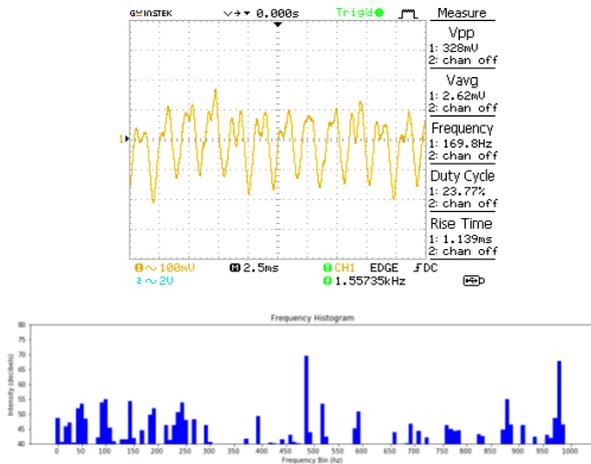
Pengujian sensor dengan menggores pada sumbu Z – Data 7



Pengujian sensor dengan menggores pada sumbu Z – Data 8



Pengujian sensor dengan menggores pada sumbu Z – Data 9



Program MATLAB untuk proses *learning* ANN dengan 2 *Hidden Layer*

```
clear all
% KUMPULAN DATA
data = xlsread('DataLearning.xlsx');

% DATA TRAINING Input
Data_training=data(1:300,2:128)'; %row, column

% DATA TARGET
Data_target=data(1:300,1)';

% membangun jaringan syaraf
net = newff(minmax(Data_training),[10 10 1],
{'logsig','logsig','purelin'});

% Bobot awal neuron pada layer pertama / input
BobotAwal_Layer_Pertama_Input=net.IW{1,1};
% Bias awal neuron pada layer pertama / input
BiasAwal_Layer_Pertama=net.b{1};
% Bobot awal neuron pada layer kedua
BobotAwal_Layer_Kedua=net.LW{2,1};
% Bias awal neuron pada layer kedua
BiasAwal_Layer_Kedua=net.b{2};
% Bobot awal neuron pada layer tiga
BobotAwal_Layer_Ketiga=net.LW{3,2};
% Bias awal neuron pada layer tiga
BiasAwal_Layer_Ketiga=net.b{3};

%set max epoch, goal, learning rate, show step
net.trainParam.epochs = 100000;
net.trainParam.goal = 0.00001;

% melakukan pembelajaran
net = train(net,Data_training,Data_target);

%melakukan simulasi
an = sim(net,Data_training);
```

```

% Bobot akhir neuron pada layer pertama / input
Bobotakhir_Layer_Pertama_Input=net.IW{1,1};
% Bias akhir neuron pada layer pertama / input
Biasakhir_Layer_Pertama=net.b{1};
% Bobot akhir neuron pada layer kedua
Bobotakhir_Layer_Kedua=net.LW{2,1};
% Bias akhir neuron pada layer kedua
Biasakhir_Layer_Kedua=net.b{2};
% Bobot akhir neuron pada layer tiga
Bobotakhir_Layer_ketiga=net.LW{3,2};
% Bias akhir neuron pada layer tiga
Biasakhir_Layer_ketiga=net.b{3};

% Menggambar hasil training dengan data aktual
figure(2)
hold on
plot(Data_target','g.-')
hold off
hold on
plot(an','r-')
hold off
legend('Target','Neural Network')
save DataTraining.mat

```

Program MATLAB untuk pengujian ANN secara offline pada MATLAB

```
load DataTraining.mat
data = xlsread('DataTest.xlsx');
Mags = data(1:35,2:128)'; %row, column
Target = data(1:35,1)'; %row, column
Node = zeros(1,35);
ErrorRate= 0;
for k = 1:1:35
    Synaps = zeros(1,10);
    Telden = zeros(1,10);
    Cell = zeros(1,10);
    Akson = zeros(1,10);
    for i = 1:1:10
        for j = 1:1:127
            Synaps(i) = Synaps(i) +
(Bobotakhir_Layer_Pertama_Input(i,j) .*
Mags(j,k));
        end
        Synaps(i) = Synaps(i) +
Biasakhir_Layer_Pertama(i,1);
    end

    for m = 1:1:10
        Telden(m) = 1./(1 + exp(-Synaps(m)));
    end

    for i = 1:1:10
        for j = 1:1:10
            Cell(i) = Cell(i) +
(Bobotakhir_Layer_Kedua(i,j) .* Telden(j));
        end
        Cell(i) = Cell(i) +
Biasakhir_Layer_Kedua(i,1);
    end

    for m = 1:1:10
        Akson(m) = 1./(1 + exp(-Cell(m)));
    end
end
```

```

    for i = 1:1:10
        Node(k) = Node(k) +
(Bobotakhir_Layer_ketiga(1,i) .* Akson(i));
    end
    Node(k) = Node(k) + Biasakhir_Layer_ketiga;

    if Node(k) > 0.5 && Node(k) <= 1.5
        Node(k) = 1;
    elseif Node(k) > 1.5 && Node(k) <= 2.5
        Node(k) = 2;
    elseif Node(k) > -0.5 && Node(k) <= 0.5
        Node(k) = 0;
    else
        Node(k) = 3;
    end

    if Node(k) ~= Target(k)
        ErrorRate = ErrorRate + 1;
    end
end

figure(1)
hold on
plot(Target','g.-')
hold off
hold on
plot(Node','rx-')
hold off
legend('Target','Neural Network')
ErrorRate = ErrorRate / 60 * 100

```

Program Arduino untuk menjalankan algoritma Fast Fourier Transform

```
// Audio Spectrum Display
// Copyright 2013 Tony DiCola
// (tony@tonydicola.com)
// Improved by Putra Asri 2017

#define ARM_MATH_CM4
#include <arm_math.h>

////////////////////////////////////
// CONFIGURATION
// These values can be changed to alter the
// behavior of the spectrum display.
////////////////////////////////////

int SAMPLE_RATE_HZ = 2000; // Sample rate of
the audio in hertz.
const int FFT_SIZE = 256; // Size of the FFT.
Realistically can only be at most 256 without
running out of memory for buffers and other
state.
const int AUDIO_INPUT_PIN = 16; // Input ADC pin
for audio data.
const int ANALOG_READ_RESOLUTION = 12; // Bits
of resolution for the ADC.
const int ANALOG_READ_AVERAGING = 16; // Number
of samples to average with each ADC reading.
const int POWER_LED_PIN = 13; // Output pin for
power LED (pin 13 to use Teensy 3.0's onboard
LED).

////////////////////////////////////
// INTERNAL STATE
// These shouldn't be modified unless you know
// what you're doing.
////////////////////////////////////

IntervalTimer samplingTimer;
```

```

float samples[FFT_SIZE*2];
float magnitudes[FFT_SIZE];
int sampleCounter = 0;

////////////////////////////////////
// MAIN SKETCH FUNCTIONS
////////////////////////////////////

void setup() {
    // Set up serial port.
    Serial.begin(115200);

    // Set up ADC and signal input.
    pinMode(AUDIO_INPUT_PIN, INPUT);
    analogReadResolution(ANALOG_READ_RESOLUTION);
    analogReadAveraging(ANALOG_READ_AVERAGING);

    // Turn on the power indicator LED.
    pinMode(POWER_LED_PIN, OUTPUT);
    digitalWrite(POWER_LED_PIN, HIGH);

    // Begin sampling signal
    samplingBegin();
}

void loop() {
    // Calculate FFT if a full sample is
    // available.
    if (samplingIsDone()) {
        // Run FFT on sample data.
        arm_cfft_radix4_instance_f32 fft_inst;
        arm_cfft_radix4_init_f32(&fft_inst,
FFT_SIZE, 0, 1);
        arm_cfft_radix4_f32(&fft_inst, samples);
        // Calculate magnitude of complex numbers
        // output by the FFT.
        arm_cmplx_mag_f32(samples, magnitudes,
FFT_SIZE);
    }
}

```

```

    for (int i = 1; i < FFT_SIZE / 2; ++i) {
        float mags = 20 * log10(magnitudes[i]) -
40;
        if(mags < 0) mags = 0;
        Serial.print(mags);
        if(i < FFT_SIZE / 2 - 1) Serial.print(";");
        else Serial.println("");
    }

    // Restart signal sampling.
    samplingBegin();
}

delay(500);
}

////////////////////////////////////
// UTILITY FUNCTIONS
////////////////////////////////////

// Compute the average magnitude of a target
frequency window vs. all other frequencies.
void windowMean(float* magnitudes, int lowBin,
int highBin, float* windowMean, float*
otherMean) {
    *windowMean = 0;
    *otherMean = 0;
    // Notice the first magnitude bin is skipped
because it represents the
    // average power of the signal.
    for (int i = 1; i < FFT_SIZE/2; ++i) {
        if (i >= lowBin && i <= highBin) {
            *windowMean += magnitudes[i];
        }
        else {
            *otherMean += magnitudes[i];
        }
    }
}
}

```

```

        *windowMean /= (highBin - lowBin) + 1;
        *otherMean /= (FFT_SIZE / 2 - (highBin -
lowBin));
    }

// Convert a frequency to the appropriate FFT
bin it will fall within.
int frequencyToBin(float frequency) {
    float binFrequency = float(SAMPLE_RATE_HZ) /
float(FFT_SIZE);
    return int(frequency / binFrequency);
}

//////////////////////////////////////
// SAMPLING FUNCTIONS
//////////////////////////////////////

void samplingCallback() {
    // Read from the ADC and store the sample data
    samples[sampleCounter] =
(float32_t)analogRead(AUDIO_INPUT_PIN);
    // Complex FFT functions require a coefficient
for the imaginary part of the input.
    // Since we only have real data, set this
coefficient to zero.
    samples[sampleCounter+1] = 0.0;
    // Update sample buffer position and stop
after the buffer is filled
    sampleCounter += 2;
    if (sampleCounter >= FFT_SIZE*2) {
        samplingTimer.end();
    }
}

void samplingBegin() {
    // Reset sample buffer position and start
callback at necessary rate.
    sampleCounter = 0;

```

```
    samplingTimer.begin(samplingCallback,  
1000000/SAMPLE_RATE_HZ);  
}  
  
boolean samplingIsDone() {  
    return sampleCounter >= FFT_SIZE*2;  
}
```

Program Arduino untuk menjalankan Artificial Neural Network

```
////////////////////////////////////  
// ANN FUNCTIONS  
////////////////////////////////////  
  
float ANN_Process() {  
    // Reset all Variable to zero  
    FlushVar();  
    // First Iteration  
    for(int i = 0; i < HiddenLayer1; i++) {  
        for(int j = 0; j < 127; j++)    Synaps[i] +=  
Wi[j + (i*127)] * mags[j];  
        Synaps[i] += Bi[i];  
    }  
    // Log Sigmoid 1 Activation  
    for(int i = 0; i < HiddenLayer1; i++)  
Telden[i] = 1/(1 + exp(-Synaps[i]));  
  
    //Second Iteration  
    for(int i = 0; i < HiddenLayer2; i++) {  
        for(int j = 0; j < HiddenLayer1; j++)  
Cell[i] += Wh[j + (i*10)] * Telden[j];  
        Cell[i] += Bh[i];  
    }  
    //Log Sigmoid 2 Activation  
    for(int i = 0; i < HiddenLayer2; i++)  
Akson[i] = 1/(1 + exp(-Cell[i]));  
  
    //Final Iteration  
    for(int i = 0; i < 8; i++)    Node += Wo[i] *  
Akson[i];  
    Node += Bo;  
  
    return Node;  
}
```

```
// Reset all Variable to zero
void FlushVar() {
    memset(Synaps, 0, sizeof(Synaps));
    memset(Telden, 0, sizeof(Telden));
    memset(Cell, 0, sizeof(Cell));
    memset(Akson, 0, sizeof(Akson));
    Node = 0.0;
}
```

BIODATA PENULIS



Penulis adalah mahasiswa Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember yang termuda di angkatannya. Penulis lahir di sebuah BKIA di Kabupaten Jombang pada tanggal 3 Maret 1997. Penulis menghabiskan waktu pra-kuliah di Jombang dengan menduduki TK Pra Bhakti, SDN Mojokrapak 1, SMP Negeri 1 Jombang, dan SMA Negeri 3 Jombang. Penulis merupakan saudara tertua dari 3 bersaudara.