

**KLASIFIKASI DATA GETARAN MOTOR BEAT POP
DENGAN METODE *SUPPORT VECTOR MACHINE* (SVM)
MENGUNAKAN BAHASA *PYTHON***

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh :

**Nama : Awi Hamzah Rambe
No. Mahasiswa : 16525098
NIRM : 2016060869**

**PROGRAM STUDI TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2022

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa karya tulis ilmiah yang saya buat ini merupakan karya sendiri, bukan hasil *plagiarsm* dari karya tulis yang dibuat oleh orang lain. Semua referensi dan kutipan yang saya gunakan pada karya tulis ini saya cantumkan berupa sitasi dan sumber pustakanya. Apabila dikemudian hari karya tulis saya ini dianggap melakukan pelanggaran hak kekayaan intelektual, maka saya bersedia untuk menerima sanksi ataupun hukuman yang berlaku.

Yogyakarta, 05 Oktober 2022



Awi Hamzah Rambe

LEMBAR PENGESAHAN DOSEN PEMBIMBING

**KLASIFIKASI DATA GETARAN MOTOR BEAT POP
DENGAN METODE *SUPPORT VECTOR MACHINE* (SVM)
MENGUNAKAN BAHASA *PYTHON***

TUGAS AKHIR

Disusun Oleh :

Nama : Awi Hamzah Rambe
No. Mahasiswa : 16525098
NIRM : 2016060869

Yogyakarta, 15 September 2022

Dosen Pembimbing I,



Mohammad Faizun, S.T., M.Eng., Ph.D.

NIP. 115250101

Dosen Pembimbing II,



Ir. Donny Suryawan, S.T., M.Eng., IPP.

NIP. 175250404

LEMBAR PENGESAHAN DOSEN PENGUJI

**KLASIFIKASI DATA GETARAN MOTOR BEAT POP
DENGAN METODE *SUPPORT VECTOR MACHINE (SVM)*
MENGUNAKAN BAHASA PYTHON**

TUGAS AKHIR

Disusun Oleh :

Nama : Awi Hamzah Rambe
No. Mahasiswa : 16525098
NIRM : 2016060869

Tim Penguji

Ir. Donny Suryawan, S.T., M.Eng., IPP.

Ketua


Tanggal : 5 Oktober 2022

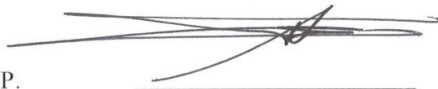
Yustiasih Purwaningrum, S.T., M.T.

Anggota I


Tanggal : 3 Oktober 2022

Ir. Faisal Arif Nurgesang, S.T., M.Sc. IPP.

Anggota II


Tanggal : 4 Oktober 2022

Mengetahui

Ketua Program Studi Teknik Mesin



Dr. Ir. Muhammad Khafid, S.T., M.T., IPP

iv

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT atas nikmat, karunia-Nya dan berkah-Nya yang tak pernah putus. Shalawat beriringkan salam yang tak lupa disanjung tinggikan kepada junjungan kita nabi besar Muhammad SWA yang kita harapkan syafa'atnya di hari akhir nanti.

Skripsi ini saya persembahkan untuk :

Kedua orang tua saya yang sangat saya banggakan dan syukuri atas kehadiran mereka di dalam hidup saya.

Ayah dan Ibu yang tidak pernah berhenti memberikan cinta dan kasih sayang yang tulus, yang tidak pernah putus dan lupa menyebut nama saya dalam setiap doa-doa yang mereka panjatkan serta yang selalu memberikan masukan dan mendukung semua cita-cita dan impian saya.

Ayah dan Ibu, Terima kasih sudah rela berjuang dan mendidik saya untuk menjadi anak yang lebih baik. Saya bersemangat dan mampu tetap berjuang itu semua karena doa-doa terbaik dari Ayah dan Ibu.

Keluarga saya yang tercinta, abang, kaka, adik yang selalu ada untuk menemani saya baik dalam saat senang maupun susah serta yang selalu memberikan motivasi dan dukungan agar saya tetap semangat dan berjuang. Saya sangat bersyukur dan bangga dikelilingi keluarga yang saling peduli, saling menyayangi dan saling mendukung.

Saya juga sangat bersyukur atas diri saya sendiri. Terima kasih atas kepercayaan dan kemampuan dalam melewati rintangan demi rintangan dan berjuang tanpa mengeluh. Terima kasih juga sudah menghargai kemampuan dirimu sendiri.

HALAMAN MOTTO

“Hidup yang baik adalah hidup yang di inspirasi oleh cinta dan dipandu oleh ilmu pengetahuan” (**Bertrand Russell**)

“Hanya ada dua pilihan untuk memenangkan kehidupan : Keberanian atau Keikhlasan. Jika tidak berani, ikhlaslah menerimanya. Jika tidak ikhlas, beranilah mengubahnya” (**Lenang Manggala**)

“Orang cerdas ialah orang yang menyiapkan dirinya dan beramal untuk hari kematian, sedangkan orang yang lemah ialah orang yang mengikutkan dirinya kepada hawa nafsunyadan berkhayal kosong pada Allah” (**H.R At-Tirmidzi**)

“Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu. Allah mengetahui, sedang kamu tidak mengetahui” (**Q.S. Al Baqarah: 216**)

“Barang siapa yang menunjuki kepada kebaikan, maka ia akan mendapat pahala seperti pahala orang yang mengerjakannya” (**HR. Muslim**)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu 'alaikum Warahmatullahi Wabarakaatuh

Alhamdulillahirobbil 'Alamin, segala puji syukur penulis panjatkan ke hadirat Allah SWT atas nikmat, hidayah dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan judul “Klasifikasi Data Getaran Motor Beat Pop Dengan Metode *Support Vector Machine* (SVM) Menggunakan Bahasa *Python*”. Shalawat serta salam senantiasa tercurahkan kepada Nabi Besar Muhammad SAW beserta keluarga, para sahabat dan para pengikutnya. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh Gelar Sarjana Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulis sangat menyadari bahwa dalam penulisan skripsi ini tidak terlepas dari berbagai hambatan dan rintangan yang dialami, namun berkat bantuan dari berbagai pihak, baik itu bantuan berupa bimbingan, motivasi, dukungan, fasilitas, kerjasama dan bantuan lainnya. Sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Maka pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada :

1. Allah SWT, karena dengan limpahan nikmat, hidayah dan karunia-Nya, penulis bisa melewati rintangan dan hambatan dalam penyusunan skripsi ini hingga selesai.
2. Nabi Muhammad SAW yang menjadi suri teladan bagi ummat muslim
3. Kepada kedua orang tua, Daman Rambe sebagai ayah dan Air Batun Siregar sebagai ibu, yang selalu berjuang, mendukung dan mendo'akan penulis agar menjadi orang yang bermanfaat dan sukses.
4. Bapak Fathul Wahid, S.T., M.Sc., Ph.D, selaku Rektor Universitas Islam Indonesia
5. Bapak Dr. Ir. Muhammad Khafid, S.T., M.T., IPP, selaku Ketua Prodi Teknik Mesin Universitas Islam Indonesia

6. Bapak Faizun, S.T., M.Eng., Ph.D, selaku Dosen pembimbing I pada Tugas Akhir ini
7. Bapak Ir. Donny Suryawan, S.T.,M.Eng.,IPP, selaku Dosen pembimbing II pada Tugas Akhir ini
8. Seluruh Dosen Teknik Mesin Universitas Islam Indonesia yang telah ikhlas dalam memberikan ilmu yang bermanfaat di dalam perkuliahan
9. Kepada saudara saya tercinta. Bang Karya, Kak Efrida, Kak Ika dan Musbar. Terima kasih telah menjadi keluarga yang saling mendukung serta menjadi sumber semangat dan kebahagiaan.
10. Seluruh teman-teman Jurusan Teknik Mesin, khususnya angkatan 2016 Universitas Islam Indonesia yang telah berjuang bersama-sama selama proses perkuliahan dan memberikan banyak masukan maupun berbagai ilmu pengetahuan.
11. Kepada semua pihak *internal* maupun *external* kampus Universitas Islam Indonesia yang penulis tidak dapat sebutkan satu per satu. Terima kasih atas segala bantuan dalam kebaikan yang telah diberikan kepada penulis.

Penulis berharap semoga skripsi ini dapat memberikan manfaat serta sumber ilmu pengetahuan, khususnya untuk penulis sendiri dan pembaca. Penulis juga sangat menyadari bahwa didalam skripsi ini masih terdapat kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun. Kritik dan saran dapat disampaikan dengan mengirim pesan email ke : 16525098@uii.ac.id sebagai bahan untuk evaluasi demi kesempurnaan penyusunan skripsi selanjutnya

Wassalamu'alaikum Warahmatullahi Wabarakaatuh.

Yogyakarta, 28 Juni 2022



Awi Hamzah Rambe

ABSTRAK

Getaran yang ditimbulkan oleh medan jalan sangat mempengaruhi faktor kenyamanan dan keselamatan pengendara dalam berkendara. Selain mempengaruhi keselamatan dan kenyamanan juga berpengaruh terhadap tubuh pengendara. Getaran tidak dapat diukur oleh *panca indera* manusia sehingga untuk dapat memahami hal tersebut salah satu cara yang dapat dilakukan adalah dengan merancang sebuah alat pengukur getaran sepeda motor saat dikendarai. Pada penelitian ini, alat yang digunakan yaitu rangkaian mekatronika dari sensor *Accelerometer ADXL345* dan *Arduino Uno*. Setelah data getaran diperoleh dari hasil pengukuran, selanjutnya akan dilakukan pengolahan data dengan metode *Machine Learning* yaitu algoritma Klasifikasi *Support Vector Machine* (SVM). Tujuan dari penelitian ini adalah membuat *Machine Learning Model* dengan algoritma Klasifikasi *Support Vector Machine* (SVM) dengan menggunakan konsep *Polynomial Kernel* dan menerapkan metode *Cross Validation*. Dari hasil implementasi klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) menggunakan konsep *polynomial kernel* yang diterapkan pada data hasil pengujian didapatkan nilai akurasi *Machine Learning Model* sebesar 78%, dihitung dari 9 *testing dataset* berhasil diklasifikasi secara benar dengan kelas aslinya sebanyak 7 data.

Kata Kunci : *Machine Learning, Support Vector Machine, Supervised Learning, Polynomial Kernel , Cross Validation, Getaran, Sensor ADXL345, Kenyamanan Kendaraan*

ABSTRACT

The vibration caused by the road terrain greatly affects the comfort and safety of the driver in driving. In addition to affecting safety and comfort, it also affects the rider's body. Vibration cannot be measured by the physiological senses of humans, so to be able to understand this, one way that can be done is to design a motorcycle vibration measuring device when riding. In this study, the tool used is a mechatronic circuit from the Accelerometer ADX345 sensor and Arduino Uno. After the vibration data is obtained from the measurement results, then the data will be processed using the Machine Learning method, namely the Support Vector Machine (SVM) Classification Algorithm. The purpose of this research is to create a Machine Learning Model with the Support Vector Machine (SVM) Classification Algorithm using the polynomial kernel concept and applying Cross Validation. From the results of the classification implementation using the Support Vector Machine (SVM) algorithm using the polynomial kernel concept which is applied to the data, the Machine Learning Model accuracy value is 78%. Counting from 9 testing datasets were successfully classified correctly with the original class of 7 data.

Keywords: *Machine Learning, Support Vector Machine, Supervised Learning, Polynomial Kernel, Cross Validation, Vibration, ADXL345 Sensor, Vehicle Comfort*

DAFTAR ISI

Pernyataan Keaslian	ii
Lembar Pengesahan Dosen Pembimbing	iii
Lembar Pengesahan Dosen Penguji	iv
Halaman Persembahan	v
Halaman Motto	vi
Kata Pengantar.....	vii
Abstrak	ix
<i>Abstract</i>	x
Daftar Isi.....	xi
Daftar Tabel.....	xiv
Daftar Gambar	xv
daftar Notasi	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian atau Perancangan	5
1.5 Manfaat Penelitian atau Perancangan.....	5
1.6 Sistematika Penulisan	5
Bab 1 : Pendahuluan	5
Bab 2 : Tinjauan Pustaka	5
Bab 3 : Metode Penelitian	6
Bab 4 : Hasil dan Pembahasan	6
Bab 5 : Penutup	6
BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Dasar Teori	7
2.2.1 Data Mining.....	7
2.2.2 <i>Machine Learning</i>	8
2.2.3 Analisis Regresi Logistik	11

2.2.4	Klasifikasi.....	13
2.2.5	<i>Support Vector Machine (SVM)</i>	13
2.2.6	<i>Multiclass Classification</i>	17
2.2.7	Validasi Silang (<i>Cross Validation</i>).....	20
2.2.8	Pengukuran Kinerja Atau Akurasi <i>Machine Learning Model</i>	20
2.2.9	<i>Python</i>	21
2.2.10	<i>Anaconda</i>	22
2.2.11	<i>Jupyter Notebook</i>	23
2.2.12	Getaran.....	24
2.2.13	Sensor <i>Accelerometer ADXL345</i>	25
2.2.14	Mikrokontroler	30
2.2.15	<i>Arduino Uno</i>	30
2.2.16	<i>Arduino IDE 1.8.20</i>	34
2.2.17	Modul <i>SD-Card</i>	35
2.3	Statistika Deskriptif Data.....	37
BAB 3 METODE PENELITIAN		38
3.1	Langkah Penelitian	38
3.1.1	Studi Literatur.....	39
3.1.2	Perancangan Perangkat Keras	39
3.1.3	Perancangan Perangkat Lunak.....	39
3.1.4	Pengujian Alat atau Perekaman Getaran	39
3.1.5	Pelabelan Data	39
3.1.6	Statistika Deskriptif Data	40
3.1.7	<i>Data Preparation</i>	40
3.1.8	Metode Klasifikasi <i>Support Vector Machine (SVM)</i>	40
3.2	Peralatan dan Bahan.....	41
3.2.1	Perangkat Lunak	41
3.2.2	Perangkat Keras	41
3.3	Perancangan Sistem	43
3.3.1	Desain Sistem	43
3.3.2	Perancangan Perangkat Keras	44
3.3.3	Perancangan Perangkat Lunak.....	47

3.3.4	Kalibrasi Sensor <i>Accelerometer ADXL345</i>	48
3.3.5	Pengujian Alat atau Perekaman Getaran	49
3.3.6	<i>Setup</i> Penempatan Perangkat Keras	49
3.3.7	Lokasi dan Objek Pengujian Alat.....	52
3.4	Data.....	53
3.4.1	Pelabelan Data	55
3.4.2	Statistika Deskriptif	58
3.5	<i>Data Preparation</i>	58
BAB 4 HASIL DAN PEMBAHASAN		60
4.1	Hasil Perancangan.....	60
4.2	Hasil Pengujian	61
4.2.1	Hasil Pengujian Objek A	62
4.2.2	Hasil Pengujian Objek B	65
4.2.3	Hasil Pengujian Objek C	68
4.3	Analisis Dan Pembahasan.....	71
4.4	Metode Klasifikasi <i>Support Vector Machine (SVM)</i>	71
4.4.1	<i>Polynomial Kernel SVM</i>	72
4.5	Ketepatan Klasifikasi Pada <i>Machine Learning Model SVM</i>	73
4.5.1	<i>Confusion Matrix Training Dataset</i>	74
4.5.2	<i>Confusion Matrix Testing Dataset</i>	75
4.6	<i>Accuracy</i>	75
BAB 5 PENUTUP		77
5.1	Kesimpulan	77
5.2	Saran atau Penelitian Selanjutnya.....	77
DAFTAR PUSTAKA		78
LAMPIRAN		83

DAFTAR TABEL

Tabel 2- 1 Fungsi <i>Kernel SVM</i>	16
Tabel 2- 2 <i>Confusion Matrix</i> Tiga Kelas	21
Tabel 2- 3 Deskripsi Pin <i>Accelerometer ADXL345</i>	26
Tabel 2 - 4 Spesifikasi Data Teknik <i>Arduino Uno R3</i>	32
Tabel 2- 5 Deskripsi Pin <i>Arduino UNO</i>	32
Tabel 2- 6 Deskripsi Pin <i>SD-Card Model SPI</i>	36
Tabel 3- 1 Software Perangkat Lunak dan Versi	41
Tabel 3- 2 Peralatan dan Bahan	41
Tabel 3- 3 Komponen Perancangan dan Fungsinya	44
Tabel 3- 4 Parameter Pengujian Alat atau Perekaman Getaran	49
Tabel 3- 5 Perekaman Getaran	53
Tabel 3- 6 Pelabelan Data	56
Tabel 3- 7 Variabel Data Input (x) (<i>Independent Variable</i>)	58
Tabel 3- 8 Variabel Data Output (y) (<i>Dependent Variable</i>)	59
Tabel 4- 1 Parameter Model SVM <i>Polynomial Kernel</i> Metode <i>Grid Search Cross Validation</i>	72
Tabel 4- 2 Parameter <i>Machine Learning Model SVM Polynomial Kernel</i>	73
Tabel 4- 3 <i>Confusion Matrix Training Dataset</i>	74
Tabel 4- 4 <i>Confusion Matrix Testing Dataset</i>	75
Tabel 4- 5 <i>Accuracy</i> Klasifikasi <i>Training Dataset</i> dan <i>Testing Dataset</i>	76

DAFTAR GAMBAR

Gambar 2- 1: Skema <i>Artificial Intelligence</i> dan <i>Machine Learning</i>	9
Gambar 2- 2: <i>Support Vector Classifier</i>	14
Gambar 2- 3 <i>Transformasi Input Space Ke Feature Space</i>	15
Gambar 2- 4 Metode <i>One vs Rest (Ovr)</i>	18
Gambar 2- 5 Metode <i>One vs One (Ovo)</i>	19
Gambar 2- 6 Logo <i>Python</i>	22
Gambar 2- 7 Logo <i>Anaconda</i>	23
Gambar 2- 8 Logo <i>Jupyter Notebook</i>	23
Gambar 2- 9 Diagram Pin <i>Accelerometer ADXL345</i>	26
Gambar 2- 10 Sensor <i>ADXL345</i>	28
Gambar 2- 11 Sumbu Sensor <i>Accelerometer ADXL345</i>	29
Gambar 2- 12 Respon output <i>ADXL345</i> Sehubungan Dengan Orientasi Gravitasi	29
Gambar 2- 13 <i>Arduino Uno R3</i>	31
Gambar 2- 14 Halaman <i>Arduino IDE</i> Versi 1.8.20	34
Gambar 2- 15 Modul Micro <i>SD-Card</i>	35
Gambar 2- 16 Diagram Koneksi <i>SD-Card</i> Model SPI.....	36
Gambar 3- 1 Alur Penelitian.....	38
Gambar 3- 2 Alur Desain Sistem.....	43
Gambar 3- 3 <i>Schematic</i> Perangkat Keras	45
Gambar 3- 4 Diagram alir program dengan <i>Arduino UNO</i>	47
Gambar 3- 5 Kalibrasi <i>Sensor Accelerometer ADXL345</i>	48
Gambar 3- 6 Posisi <i>Switch</i>	50
Gambar 3- 7 <i>Box Arduino Uno</i> dan <i>Powerbank</i>	50
Gambar 3- 8 Posisi <i>Arduino Uno</i> dan <i>Power Bank</i>	51
Gambar 3- 9 Posisi Sensor <i>ADXL345</i>	52
Gambar 3- 10 Objek A	52
Gambar 3- 11 Objek B	52
Gambar 3- 12 Objek C	53
Gambar 4- 1 Rangkaian Perangkat Keras.....	60

Gambar 4- 2 Grafik Data Hasil Pengujian Objek A Axis X 62
Gambar 4- 3 Grafik Data Hasil Pengujian Objek A Axis Y 63
Gambar 4- 4 Grafik Data Hasil Pengujian Objek A Axis Z..... 64



DAFTAR NOTASI

$C = cost$

$\gamma = gamma$

$d = degree$

$cv = cross\ validation$



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan jaman, transportasi menjadi suatu kebutuhan yang sangat penting dalam kehidupan sehari-hari. Salah satu alat transportasi yang sering dijumpai adalah sepeda motor dan sesuai dengan fungsinya yaitu sepeda motor dapat digunakan sebagai alat transportasi dalam sehari-hari. Sepeda motor banyak sekali digunakan khususnya di Indonesia. Selain pengoperasinya yang mudah harganya juga relatif lebih murah dibandingkan dengan transportasi lainnya seperti mobil (SUHANDOKO, S., 2014).

Getaran merupakan gerakan bolak-balik dalam suatu interval waktu tertentu. Getaran berhubungan dengan gerakan osilasi atau sebuah variasi periodik. Salah satu contoh getaran yang sering dirasakan dalam kehidupan sehari-hari yaitu getaran pada sepeda motor saat dikendarai. Dimana getaran yang terjadi disebabkan oleh getaran mesin maupun permukaan jalan yang tidak rata.

Getaran yang terjadi pada sebuah motor dapat menentukan keamanan maupun keselamatan pengendara. Efek dari sebuah getaran yang berlebihan akan menyebabkan suatu komponen akan lebih cepat aus atau rusak, selain itu getaran yang berlebihan juga akan mengurangi kenyamanan pengguna khususnya pada saat mengendarai sepeda motor. Dengan demikian, salah satu faktor kenyamanan pengendara sepeda motor tergantung dari seberapa besar getaran yang terjadi (SUHANDOKO, S., 2014).

Berdasarkan data dari Asosiasi Industri Sepeda Motor Indonesia (AISI) sampai dengan bulan juli 2022 tercatat total penjualan sepeda mencapai 2.573.079 selama 7 bulan terakhir. Dimana 5,58 % penjualan sepeda motor kategori *sport*, 7,43% penjualan sepeda motor untuk kategori *underbone* dan 86,99% penjualan sepeda motor untuk kategori *scootic* (jenis motor *matic*). Salah satunya jenis motor *matic* terlaris adalah sepeda motor *matic* Honda Beat Series dengan kapasitas mesin 110cc, sehingga sepeda motor *matic* yang digunakan

pada penelitian ini adalah sepeda motor beat pada generasi ketiga yaitu sepeda motor beat pop dengan kapasitas mesin 110 cc.

Tujuan dari penelitian ini yaitu mengimplentasikan metode Klasifikasi *Support Vector Machine* (SVM). Sehingga dibutuhkan sebuah data untuk bahan sampel klasifikasi. Data yang digunakan yaitu data getaran dari sepeda motor matic beat pop.

Klasifikasi adalah teknik yang digunakan untuk menentukan atau memperkirakan kelas suatu objek berdasarkan atribut data yang ada. Klasifikasi ini dapat diaplikasikan pada sejumlah sektor termasuk: medis, perbankan, industri dan bahkan komersial. Di beberapa bidang, klasifikasi digunakan sebagai alat pendukung pembuat keputusan secara cepat pada masalah yang kompleks dan *big data*.

Pada pemodelan klasifikasi dibutuhkan model terbaik untuk dapat digunakan memprediksi suatu data agar sesuai dengan kelas target. Nilai *accuracy* yang rendah dapat dijadikan sebagai salah satu pertimbangan dalam pemilihan model terbaik. Kasus klasifikasi dapat diselesaikan dengan menggunakan metode statistika dan *Machine Learning*. Dalam menyelesaikan kasus klasifikasi, ada beberapa metode yang dapat digunakan yaitu *Naïve Bayes*, *Decision Tree based Methods*, *Rule-based Methods*, *Support Vector Machine*, *Neural Network* dan *K-Neavrest Neighbor* (KNN). Salah satu metode dari *machine learning* yang digunakan untuk menyelesaikan kasus klasifikasi adalah *Support Vector Machine* (SVM).

Metode Support Vector Machine (SVM) merupakan metode dengan kelas *Supervised Learning* yang dapat digunakan untuk pengklasifikasian maupun regresi. Metode SVM ini digunakan untuk mencari pembatas (*hyperplane*) yang paling optimal antara dua set data dari dua kelas yang berbeda dalam ruang *vektor*. Pada SVM juga terdapat proses *training* data yang digunakan untuk mencari posisi paling optimal dari *hyperplane* di ruang *vektor*. Pada dasarnya SVM merupakan *linear classifier*, namun SVM juga dapat digunakan dalam permasalahan non-linear dengan memasukkan fungsi kernel di dalamnya. Pada SVM *non-linear*, data diproyeksikan ke ruang vektor baru berdimensi tinggi hingga data tersebut dapat dipisahkan secara linear.

(Nugroho, A. S. dkk, 2003) menjelaskan bahwa SVM adalah metode yang cocok untuk memecahkan masalah berdimensi tinggi dengan sampel data yang terbatas, pendekatan prinsip SVM untuk *Structural Risk Minimization* akan menghasilkan kesalahan generalisasi yang lebih kecil.

Dalam kasus klasifikasi, metode *Support Vector Machine* (SVM) memiliki kelebihan yaitu waktu komputasi yang lebih cepat dibandingkan dengan metode yang lainnya (Octaviani, P. A.dkk, 2014). Penelitian-penelitian tentang SVM sudah banyak dilakukan, diantaranya adalah penelitian yang dilakukan oleh Rustam, el at (2003) dalam (Aprilla, S, 2018) yaitu klasifikasi data aroma metode SVM lebih baik daripada metode *K-Nearest Neighbor* (KNN) dengan tingkat *accuracy* SVM yang diperoleh yaitu 100%. Kemudian pada penelitian yang dilakukan oleh Rachman dan Purnami (2012) dalam jurnal Gaussian (Octaviani, P. A.dkk, 2014) yang melakukan penelitian mengenai klasifikasi tingkat keganasan *breast cancer* dengan menggunakan metode regresi logistik dan SVM yang akhirnya diperoleh hasil bahwa tingkat *accuracy* menggunakan metode SVM lebih tinggi, yaitu sebesar 98,11% dengan menggunakan kernel *polynomial*. Selanjutnya pada penelitian yang dilakukan oleh (Kesumawati, 2018) yang berjudul “Perbandingan Metode *Support Vector Machine* (SVM) Linear, *Radial Basis Function* (RBF), dan *Polynomial Kernel* dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII” didapatkan hasil akurasi terbaik yaitu fungsi *Polynomial Kernel* dengan nilai ketepatan klasifikasi sebesar 95,45%.

Berdasarkan uraian diatas, pada penelitian ini penulis tertarik untuk menerapkan metode klasifikasi *Support Vector Machine* (SVM) dengan menggunakan konsep fungsi *Polynomial Kernel* dan menerapkan metode *Grid Search Cross Validation* untuk mendapatkan parameter terbaik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, maka diperoleh beberapa rumusan masalah sebagai berikut :

1. Bagaimana cara merancang alat untuk mengukur nilai getaran yang terjadi pada sepeda motor beat pop menggunakan sensor *Accelerometer ADXL345*?

2. Bagaimana mengimplementasikan metode klasifikasi *Support Vector Machine* (SVM) dengan menggunakan konsep fungsi *Polynomial Kernel* dan menerapkan metode *Grid Search Cross Validation* pada data getaran motor beat pop?
3. Bagaimana tingkat akurasi yang dihasilkan pada metode klasifikasi *Support Vector Machine* (SVM) dengan menggunakan konsep fungsi *Polynomial Kernel* dan menerapkan metode *Grid Search Cross Validation*?

1.3 Batasan Masalah

Berikut adalah batasan-batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Perangkat lunak yang digunakan dalam penelitian adalah software arduino dan *Microsoft Excel*.
2. Sensor yang digunakan pada penelitian ini adalah sensor *Accelerometer ADXL345* tiga Axis (X, Y, Z).
3. Objek perekaman data getaran pada penelitian ini terdiri dari tiga objek jalan dan kecepatan sepeda motor beat pop *low speed* (*maximal* 10 km/jam), *medium speed* (>10 km/jam/*maximal* 20 km/jam) dan *high speed* (>20 km/jam/*maximal* 30 km/jam).
4. Data yang digunakan pada penelitian ini adalah data getaran yang diperoleh dari getaran motor beat pop saat dikendarai
5. Label kelas data di isi secara manual dengan metode “*Self Assesment*” berdasarkan kualitas getaran yang terjadi saat mengendarai sepeda motor beat pop. Label data pada penelitian terdiri dari tiga kelas data yaitu kelas 0 sebagai kelas ‘Rendah’, kelas 1 sebagai kelas ‘Sedang’ dan kelas 2 sebagai kelas ‘Tinggi’.
6. Fungsi *kernel* yang digunakan pada penelitian ini untuk metode klasifikasi adalah *Polynomial Kernel*
7. Implementasi Klasifikasi SVM hanya sampai pada pengukuran kinerja *Model Machine Learning* yang telah dibuat berdasarkan nilai *accuracy* model.
8. Bahasa pemrograman yang digunakan adalah *Python 3.0*

1.4 Tujuan Penelitian atau Perancangan

Berdasarkan rumusah masalah diatas, maka penelitian ini dilakukan dengan tujuan sebagai berikut.

1. Merancang alat untuk mengukur nilai getaran yang terjadi pada sepeda motor beat pop saat dikendarai menggunakan sensor *Accelerometer ADXL345*
2. Mengimplementasikan metode klasifikasi *Support Vector Machine (SVM)* dengan menggunakan konsep *Polynomial Kernel* dan menerapkan *Cross Validation* pada data getaran motor beat pop.
3. Menguji tingkat akurasi model klasifikasi dengan konsep *polynomia kernel* dan menerapkan metode *Cross Validation*.

1.5 Manfaat Penelitian atau Perancangan

Adapun manfaat dari penelitian ini adalah agar dapat menambah wawasan baru terkait proses-proses untuk membentuk sebuah *Machine Learning Model* klasifikasi dengan algoritma SVM menggunakan konsep *Polynomial Kernel* dan menerapkan metode *Cross Validation* dengan bahasa pemrograman *Python*.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini disusun terdiri dari beberapa bab, yaitu :

Bab 1 : Pendahuluan

Pada bab ini dijelaskan terkait latar belakang masalah yang diangkat, solusi yang ditawarkan, tujuan dan manfaat dan metode yang digunakan.

Bab 2 : Tinjauan Pustaka

Pada bab tinjauan pustaka dijelaskan terkait tentang dasar teori yang berhubungan dengan dengan penelitian sebagai dasar yang kuat untuk menjawab sementara pada rumusan masalah

Bab 3 : Metode Penelitian

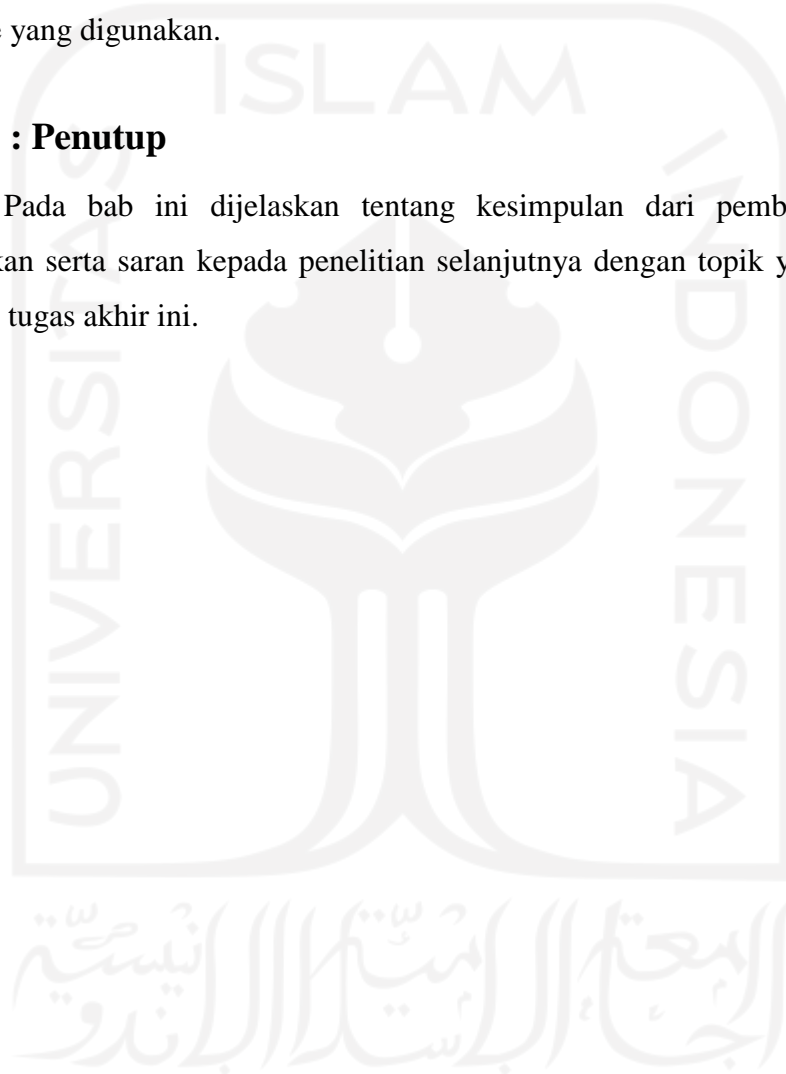
Pada bab ini membahas tentang alir penelitian yang akan digunakan, seperti langkah-langkah yang dilakukan dalam penelitian ini.

Bab 4 : Hasil dan Pembahasan

Pada bab ke empat ini akan dijelaskan terkait dengan hasil analisa dengan metode yang digunakan.

Bab 5 : Penutup

Pada bab ini dijelaskan tentang kesimpulan dari pembahasan yang dilakukan serta saran kepada penelitian selanjutnya dengan topik yang berkaitan dengan tugas akhir ini.



BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam kasus klasifikasi, metode *Support Vector Machine* (SVM) memiliki kelebihan yaitu waktu komputasi yang lebih cepat dibandingkan dengan metode yang lainnya (Octaviani, P. A.dkk, 2014). Penelitian-penelitian tentang SVM sudah banyak dilakukan, diantaranya adalah penelitian yang dilakukan oleh Rustam, et al (2003) dalam (Aprilla, S, 2018) yaitu klasifikasi data aroma metode SVM lebih baik daripada metode *K-Nearest Neighbor* (KNN) dengan tingkat *accuracy* SVM yang diperoleh yaitu 100%.

Kemudian pada penelitian yang dilakukan oleh Rachman dan Purnami (2012) dalam jurnal Gaussian (Octaviani, P. A.dkk, 2014) yang melakukan penelitian mengenai klasifikasi tingkat keganasan *breast cancer* dengan menggunakan metode regresi logistik dan SVM yang akhirnya diperoleh hasil bahwa tingkat *accuracy* menggunakan metode SVM lebih tinggi, yaitu sebesar 98,11% dengan menggunakan kernel *polynomial*. Selanjutnya pada penelitian yang dilakukan oleh (Kesumawati, 2018) yang berjudul “Perbandingan Metode Support Vector Machine (SVM) Linear, Radial Basis Function (RBF), dan Polinomial Kernel dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII” didapatkan hasil akurasi terbaik yaitu fungsi *Polynomial Kernel* dengan nilai ketepatan klasifikasi sebesar 95,45%.

2.2 Dasar Teori

2.2.1 Data Mining

Data mining adalah proses yang dilakukan supaya memperoleh informasi dari data yang dapat bermanfaat untuk pengambilan keputusan. Terdapat beberapa bidang ilmu yang termasuk dalam *data mining* diantaranya adalah *Machine Learning* dan statistika (Hermawati, 2013).

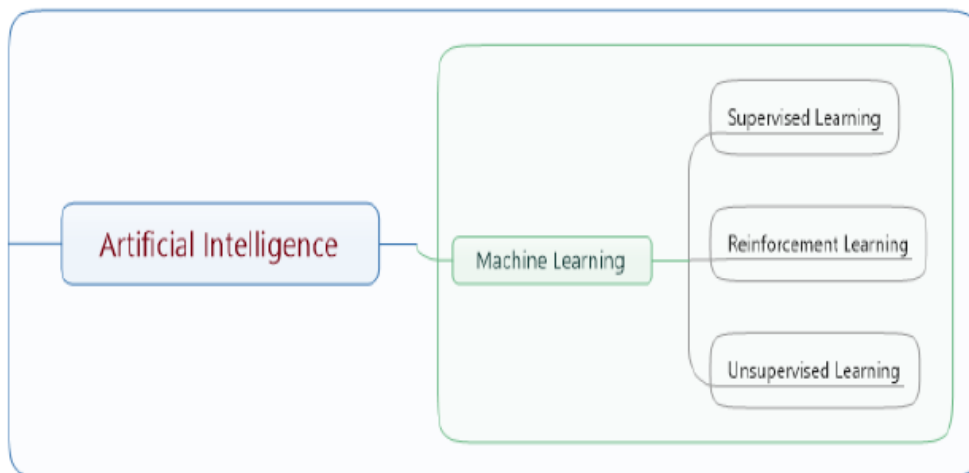
Data mining dibagi menjadi *Supervised Learning* dan *Unsupervised Learning*. *Supervised Learning* adalah pendekatan yang menggunakan data pelatihan untuk mengelompokkan data ke dalam data yang ada. Sedangkan *Unsupervised Learning* merupakan pendekatan yang tidak menggunakan data latih yang bertujuan untuk mengelompokkan data menjadi *n* kelompok. Salah satu cara untuk mencari informasi atau mengekstrak pengetahuan dengan *data mining* adalah melalui metode klasifikasi (Aprilla, S, 2018).

2.2.2 Machine Learning

Machine Learning merupakan sebuah ilmu pembelajaran yang memfokuskan pada algoritma komputer yang mampu memperbaiki dirinya sendiri secara otomatis melalui pengalaman. Metode ini dibedakan menjadi dua metode yaitu *Supervised Learning* (pembelajaran terawasi) dan *Unsupervised Learning* (pembelajaran tidak terawasi) (Ayungtyas, 2017).

Masih banyak lagi para pendapat ilmuwan terkait dengan definisi *Machine Learning*. Seperti dalam jurnal yang ditulis oleh (Roihan Ahmad,dkk, 2020), *Machine Learning* dapat didefinisikan sebagai aplikasi *computer* dan *algoritma* matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang (Goldberg, D.E., & Holland, J.H., 1988). Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (*training*) dan pengujian (*testing*) (Huang, G.B,dkk, 2006).

Dalam pembelajaran *Machine Learning* terdapat menjadi tiga jenis metode khusus atau tiga algoritma yaitu : *Supervised Learning*, *Unsupervised Learning*, *Reinforcement Learning* (Somvanshi & Chavan, 2016).



Gambar 2- 1: Skema *Artificial Intelligence* dan *Machine Learning*
(Somvanshi & Chavan, 2016)

Masing-masing algoritma digunakan untuk menyelesaikan masalah-masalah tertentu. Algoritma *Supervised Learning* digunakan untuk menuntaskan masalah yang berkaitan dengan prediksi (*regression problem*) dan klasifikasi (*classification problem*). Algoritma *Unsupervised Learning* digunakan untuk menuntaskan masalah yang berkaitan dengan klasifikasi (*classification problem*) dan deteksi anomali (*unsupervised anomaly detection*). Algoritma *Reinforcement Learning* digunakan untuk menuntaskan masalah yang berkaitan dengan masalah yang tidak dapat diatasi oleh algoritma *Supervised Learning* dan *Unsupervised Learning* (Amr, T., 2020).

2.2.2.1 *Supervised Learning*

Supervised Learning adalah suatu algoritma pembelajaran yang memiliki data *input* dan *output* yang telah diketahui sebelumnya. Sehingga tujuan dari metode *Supervised Learning* adalah untuk membangun suatu model yang menghasilkan *output* (label) yang sesuai dengan data *input*. Dari buku yang ditulis oleh (Scholkopf, B & Smola, A.J., 2002) terdapat tiga metode yang termasuk metode *supervised learning* yaitu metode *Rule-based Methods*, *Support Vector Machine* (SVM) dan *Decision Tree based Methods*. Secara umum metode *Supervised Learning* digunakan untuk menuntaskan masalah seperti prediksi

(*regression problem*) dan masalah klasifikasi (*classification problem*) (Amr, T., 2020). Masalah regresi (*regression problem*) adalah ketika variabel outputnya nilai riil, seperti dollar atau berat. Sedangkan Masalah klasifikasi (*classification problem*) adalah ketika variabel output berbentuk kategori seperti, seperti merah atau biru atau penyakit dan tidak ada penyakit (Brownlee, 2016).

Metode *Supervised Learning* beroperasi dengan melatih beberapa model atau beberapa kumpulan data yang berlabel untuk menghasilkan output data baru yang diprediksi dari sampel data (Thupae, R.dkk, 2018). Sehingga algoritma *Supervised Learning* ini didasarkan pada sekumpulan data yang memiliki label. Kumpulan sampel data digunakan untuk meringkas karakteristik dalam setiap jenis aplikasi sehingga membentuk sebuah model integritas (Amei, W.dkk, 2011).

Unsupervised Learning adalah metode pembelajaran yang tidak memerlukan goal output, sehingga *unsupervised learning* bertujuan untuk mengelompokkan data berdasarkan kemiripan variabel seperti *Clustering* (Scholkopf, B & Smola, A.J., 2002).

2.2.2.2 Unsupervised Learning

Dalam metode pembelajaran *Unsupervised Learning* mengacu pada proses pengelompokan data ke dalam *cluster* menggunakan metode atau algoritma otomatis pada data yang belum diklasifikasikan atau dikategorikan (Alloghani, M.dkk, 2020). Dalam situasi ini algoritma harus “mempelajari” hubungan atau fitur yang mendasarinya dari data yang tersedia dan kelompok kasus dengan fitur atau karakteristik yang sama (Alloghani, M.dkk, 2020). *Unsupervised Learning* melibatkan pengenalan pola tanpa keterlibatan atribut target. Artinya, semua variable yang digunakan dalam analisis adalah digunakan sebagai input dan karena pendekatan tersebut, teknik ini cocok untuk teknik *clustering* dan *association mining* (Alloghani, M.dkk, 2020).

2.2.2.3 Reinforcement Learning

Reinforcement Learning adalah suatu algoritma yang merupakan bagian dari *artificial intelligence* yang melatih sebuah algoritma dengan *sistem trial and error*. Reinforcement Learning berinteraksi dengan sekitarnya dan mengamati

konsekuensi akibat tindakannya sebagai tanggapan atas penghargaan maupun hukuman yang diterima. Informasi yang didapatkan dari interaksi dengan sekitarnya akan digunakan untuk mengupdate wawasannya (Andreasnus, J. & Kurniawan, A, 2017).

2.2.3 Analisis Regresi Logistik

Regresi logistik adalah suatu metode yang bisa digunakan untuk mengetahui hubungan antara variabel respon yang mempunyai karakteristik *dichotomous* (dua kategori) atau *polychotomous* (lebih dari dua kategori) dengan satu atau lebih variabel prediktor ber-skala kategori atau kontinu (Sahitayakti, R. P. & Fithriasari, K., 2016). Analisis regresi logistik merupakan salah satu jenis analisis regresi dimana variabel responnya bersifat kategori atau numerik. Jika variabel responnya terdiri dari dua kategori maka disebut dengan regresi logistik biner. Regresi logistik biner yaitu variabel responnya terdiri dari dua kategori yaitu bernilai satu ketika kejadian sukses dan bernilai nol ketika kejadian gagal. Sedangkan jika variabel responnya lebih dari dua kategori dan kategori tersebut merupakan variabel tingkatan maka disebut dengan regresi logistik ordinal. Model peluang antara variabel prediktor $X_{1i}, X_{2i}, \dots, X_{pi}$ dengan variabel responnya (π) adalah sebagai berikut (Ayungtyas, 2017) :

$$\pi(x_i) = \frac{\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})} \quad (2.1)$$

Dimana :

$\hat{\pi}(x_i)$: Peluang terjadinya kategori variabel respon

X_{ji} : Variabel prediktor ke-j

p : Banyaknya variabel prediktor

β_0 : Intersep

$\beta_1, \beta_2, \dots, \beta_p$: Koefisien regresi untuk setiap variabel prediktor

i : 1, 2, ..., n

Untuk memudahkan *interpretasi* dan pendugaan parameter, maka peluang pada persamaan diatas dibuat transformasi logit. Sehingga didapatkan fungsi logit sebagai berikut (Ayungtyas, 2017) :

$$g(x_i) = \text{logit } \pi(x_i) = \ln \left(\frac{\pi(x_i)}{1-\pi(x_i)} \right), \quad (2.2)$$

Jika,
$$\pi(x_i) = \frac{\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})} \quad (2.3)$$

$$\begin{aligned} g(x_i) &= \ln \left(\frac{\pi(x_i)}{1-\pi(x_i)} \right) \\ &= \ln \left(\frac{\frac{\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}}{1 - \frac{\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})}} \right) \\ &= \ln(\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi})) \\ &= \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi} \\ g(x_i) &= \beta_0 + \sum_{j=1}^p \beta_j X_{ji} \end{aligned}$$

2.2.3.1 Klasifikasi Pada Regresi Logistik

Regresi logistik adalah salah satu metode statistika klasik yang digunakan untuk menuntaskan masalah dalam klasifikasi. Pada pendekatan dalam memakai regresi logistik untuk memprediksi kelas dilakukan dengan cara menghitung probabilitas. Klasifikasi yang berasal dari variabel respon biner dilakukan dengan cara menentukan nilai titik potong. Dalam penelitian yang dilakukan oleh (Ayungtyas, 2017), titik potong yang dapat digunakan sebesar 0.5. Klasifikasi berdasarkan melalui pendekatan analisis regresi logistik menggunakan model peluang dengan ketentuan berikut :

$$\text{ketegori} \begin{cases} 0, & \pi(x) < 0.5 \\ 1, & \pi(x) \geq 0.5 \end{cases}$$

Apabila peluang yang dihasilkan dari model peluang bernilai lebih kecil (<) dari 0.5, maka hasil prediksi termasuk kategori 0. Sebaliknya, jika peluang model yang dihasilkan dari model bernilai lebih besar (>) atau sama dengan (=) 0.5, maka hasil prediksi termasuk kategori 1 (Ayungtyas, 2017).

Pada kasus klasifikasi multilevel (variabel responnya lebih dari dua) dilakukan dengan cara menghitung peluang pada setiap kategori. Sehingga menentukan nilai parameter sangat penting dalam model regresi logistik karena berhubungan dengan probabilitas yang diperoleh (Bishop.dkk, 2006). Penentuan suatu prediksi kategori atau kelas berdasarkan pada besarnya nilai peluang. Untuk

penentuan kategori didasarkan pada nilai terbesar peluang tiap kategori. Jika kategori A memiliki nilai peluang paling besar diantara dua kategori lainnya, maka prediksi kelas adalah kategori A, begitu juga untuk penentuan kategori selanjutnya (Ayungtyas, 2017).

2.2.4 Klasifikasi

Klasifikasi merupakan suatu teknik yang digunakan untuk mengetahui ataupun memperkirakan kelas dari suatu obyek berdasarkan atribut atau label yang ada. Dalam hal ini, atribut merupakan suatu sifat atau karakteristik dari suatu obyek yang nilainya bermacam-macam. Misalnya warna, kesukaan dan lain sebagainya. Sedangkan variabel adalah karakteristik suatu obyek yang nilainya unik. Misalnya tinggi badan, berat badan dan lainnya. Atribut memiliki arti yang lebih luas dibandingkan variabel, sehingga apabila suatu metode dalam pengolahan data berdasarkan variabel maka hasilnya dapat menandakan atributnya (Prasetyo, E., 2012).

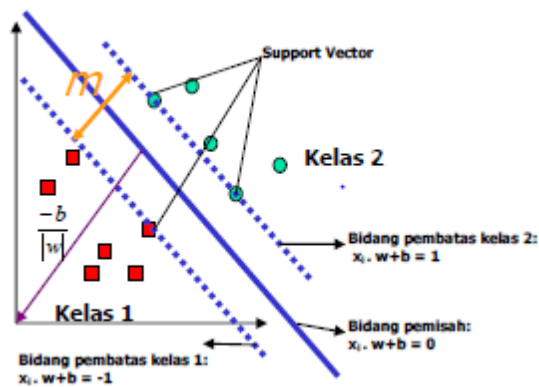
2.2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah suatu teknik pembelajaran yang memakai fungsi linier dalam sebuah ruang kerja yang berdimensi tinggi (*feature space*) dan dilatih dengan algoritma berdasarkan pada pembelajaran optimasi (Cristianini, N. & Shawe-Taylor, J., 2000). Ide dasar dari SVM adalah menemukan *hyperplane* terbaik untuk memisahkan dua kelas data input. Namun, karena data sering kali tidak dapat dipisahkan secara linier, SVM membuat gagasan baru tentang konsep *kernel* agar SVM dapat digunakan pada masalah non-linier pada ruang kerja berdimensi tinggi dengan mencari *hyperplane* yang memaksimalkan margin (jarak) antar kelas data (Boswell, D., 2002).

Dari jurnal yang ditulis oleh (Hastie, T.dkk, 2009) SVM adalah suatu teknik untuk melakukan prediksi dalam kasus klasifikasi dan regresi. Teknik ini akan bekerja untuk menemukan fungsi *hyperplane* ter-optimal yang mampu memisahkan data set dari sekumpulan set data dari dua kelas yang berbeda.

Fungsi *hyperlane* dapat dinotasikan sebagai berikut (Ayungtyas, 2017) :

$$f(x) = \mathbf{w}'\mathbf{x} + b \quad (2.4)$$



Gambar 2- 2: *Support Vector Classifier*
(Sembiring, K., 2007)

Berdasarkan gambar diatas, pada kasus terpisah, garis linier menunjukkan batas pemisah dan garis putus-putus menunjukkan margin maksimal dengan lebar $= \frac{2}{\|w\|}$. Dalam SVM terdapat dua *hyperplane* pendukung yaitu *hyperplane* yang menunjukkan kelas +1 dan *hyperplane* yang menunjukkan -1. Secara spesifik margin dapat dihitung dengan persamaan berikut (Santosa, 2007) :

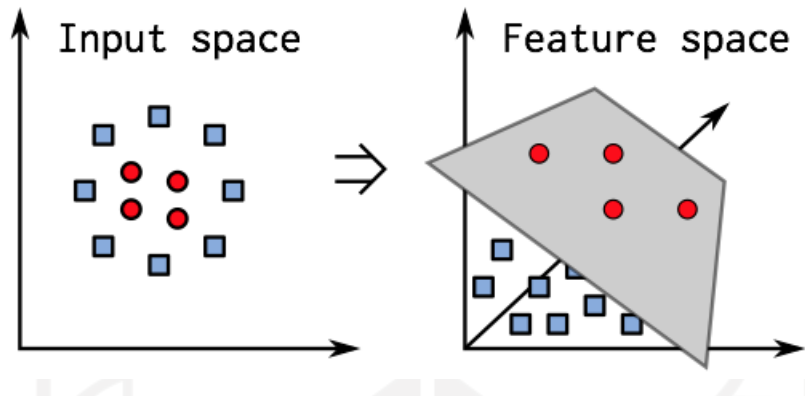
$$\begin{aligned} w'x &= b + 1 \\ w'x &= b - 1 \end{aligned} \quad (2.5)$$

Hyperplane terbaik adalah *hyperplane* yang berada di tengah-tengah antara dua set obyek dari dua kelas (Santosa, 2007). *Hyperplane* yang dimaksud bisa dalam bentuk berupa line pada *two dimension* serta berbentuk berupa *flat plane* pada *multiple line*. SVM adalah salah satu jenis dari machine learning yang dapat melakukan pelatihan dengan menggunakan *training dataset* sehingga akan membentuk suatu machine learning model yang dapat membuat prediksi ataupun kalasifikasi dari data baru.

2.2.5.1 Kernel *Support Vector Machine* (SVM)

Permasalahan di dalam dunia nyata data tidak selalu dapat dipisahkan secara linier membuat SVM kurang mampu dalam menemukan *hyperplane* terbaik, sehingga akan berpengaruh terhadap kualitas hasil klasifikasi SVM. Dengan begitu, kernel ini berfungsi sebagai salah satu trik untuk men-tranformasi data kedalam dimensi yang lebih tinggi (*feature space*). Data akan dipetakan oleh

kernel dari ruang dimensi yang rendah ke ruang dimensi yang lebih tinggi. Sehingga dengan begitu data yang awalnya tidak dapat dipisahkan secara linier akan dapat dipisahkan secara linier dengan trik kernel.



Gambar 2- 3 Transformasi Input Space Ke Feature Space

(Dima, 2016)

Cara kerja dari kernel ini yaitu data akan disimpan berupa bentuk kernel dengan mengukur kesamaan dan ketidaksamaan objek dalam data. Sampai saat ini konsep kernel ini dapat digunakan untuk berbagai objek data yang kontinu maupun data diskrit dengan melewati urutan data dan grafik. Sehingga SVM dengan metode berbasis kernel merupakan metode yang paling terkenal karena jangkauan kelas semakin luas dengan bantuan konsep kernel (Kesumawati, 2018).

Menurut teori Mercer fungsi kernel yang awalnya merupakan produk dot antara $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ dapat diganti dengan fungsi kernel $K(\vec{x}_i, \vec{x}_j)$. Tujuannya untuk membuat lebih mudah dipahami. Kernel secara implisit mendefinisikan transformasi yang disebut *kernel-trick* yang dibangun dengan persamaan berikut :

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \quad (2.6)$$

Dengan adanya fungsi kernel, di SVM jadi sangat lebih mudah untuk mendefinisikan *Support Vector*. Fungsi kernel yang biasa digunakan sebagai pada SVM adalah sebagai berikut :

Tabel 2- 1 Fungsi *Kernel* SVM

No	Kernel	Fungsi
1	Linier	$K(x_i, x) = x_i^T x$
2	Polinomial	$K(x_i, x) = (\gamma x_i^T x + r)^2. \gamma > 0$
3	<i>Radial Basis Function</i> (RBF)	$K(x_i, x) = \exp(-\gamma \ x_i - x\ ^2). \gamma > 0$
4	Sigmoid	$K(x_i, x) = \tanh(\gamma x_i^T x + r)$

1. *Linear Kernel*

Linear kernel adalah fungsi *kernel* yang paling sederhana. *Linear kernel* digunakan ketika data yang akan di analisis dapat dipisahkan secara linier. *Linear kernel* sangat rekomendasi digunakan ketika banyak fitur, karena pemetaan ke ruang dimensi yang lebih tinggi tidak benar-benar meningkatkan kinerja atau kualitas model seperti dalam klasifikasi teks. Dalam klasifikasi teks, jumlah *instance* (dokumen) dan jumlah objek (kata) adalah sama (Kesumawati, 2018).

2. *Polynomial Kernel*

Polynomial kernel adalah fungsi *kernel* yang digunakan ketika data tidak dapat dipisahkan secara linier. Ideal dari *polynomial kernel* untuk masalah dimana semua set data pelatihan dinormalisasi (Kesumawati, 2018). *Polynomial kernel* juga memiliki parameter yang khusus yang tidak dimiliki oleh *kernel* lainnya yaitu parameter *degree* : *int. default = 3*. Fungsi parameter *degree* secara default =3 yaitu untuk mencari nilai optimal pada setiap dataset. Semakin besar nilai *degree* maka akurasi dari model yang dihasilkan akan *fluktuatif* dan kurang stabil.

3. *Radial Basis Function* (RBF) *Kernel*

Radial Basis Function (RBF) *Kernel* adalah fungsi *kernel* yang biasa digunakan dalam analisis ketika data tidak dapat dipisahkan secara linier. Secara default semua *kernel* SVM memiliki parameter *Cost* (C) dan *Gamma*. Parameter C akan berperan sebagai optimasi SVM untuk menghindari kesalahan klasifikasi pada setiap sampel data pelatihan (*training*). Parameter *Gamma* berperan untuk menentukan pengaruh sampel dataset pelatihan

dengan nilai rendah berarti “jauh” dan nilai tertinggi berarti “dekat”. Nilai γ yang rendah, poin yang jauh dan pembagian logis diperhitungkan dalam perhitungan garis pembagian (hyperplane). Ketika γ tinggi, berarti titik-titik yang terletak di sekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan (Kesumawati, 2018).

2.2.6 Multiclass Classification

Ketika memecahkan masalah klasifikasi yang hanya memiliki dua label kelas, maka menjadi mudah untuk memfilter data, menerapkan algoritma klasifikasi apapun, melatih model dengan data yang difilter, dan memprediksi hasilnya. Tetapi ketika data lebih dari dua instance kelas dalam data rangkaian masukan, maka mungkin akan menjadi rumit untuk menganalisis data, melatih model, dan memprediksi hasil yang relatif akurat. Untuk menangani beberapa instance kelas ini, kami menggunakan klasifikasi multi-kelas.

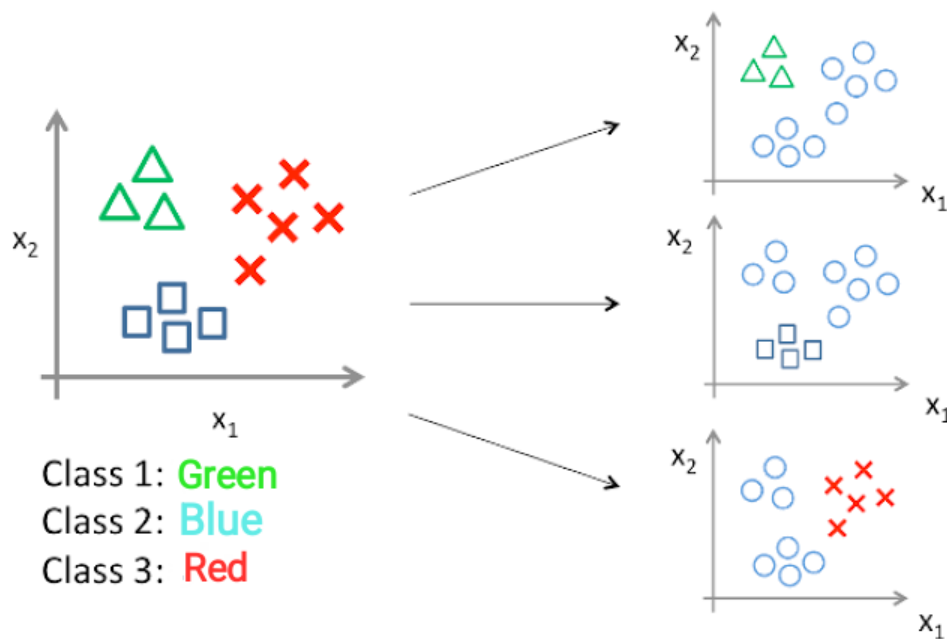
Klasifikasi multi-kelas adalah teknik klasifikasi yang memungkinkan kita untuk mengkategorikan data uji ke dalam beberapa label kelas yang ada dalam data terlatih sebagai prediksi model.

Saat ini ada dua jenis teknik klasifikasi untuk data multi-kelas yaitu :

1. *One vs Rest (Ovr)*
2. *One vs One (Ovo)*

2.2.6.1 One vs Rest (Ovr)

Dalam teknik klasifikasi *One vs Rest (Ovr)* atau *One vs All*, untuk dataset *instance* kelas-N harus membuat model pengklasifikasi biner-N. Jumlah label kelas yang ada dalam dataset dan jumlah pengklasifikasi *biner* yang dihasilkan harus sama.



Gambar 2- 4 Metode *One vs Rest (Ovr)*

(Band, 2020)

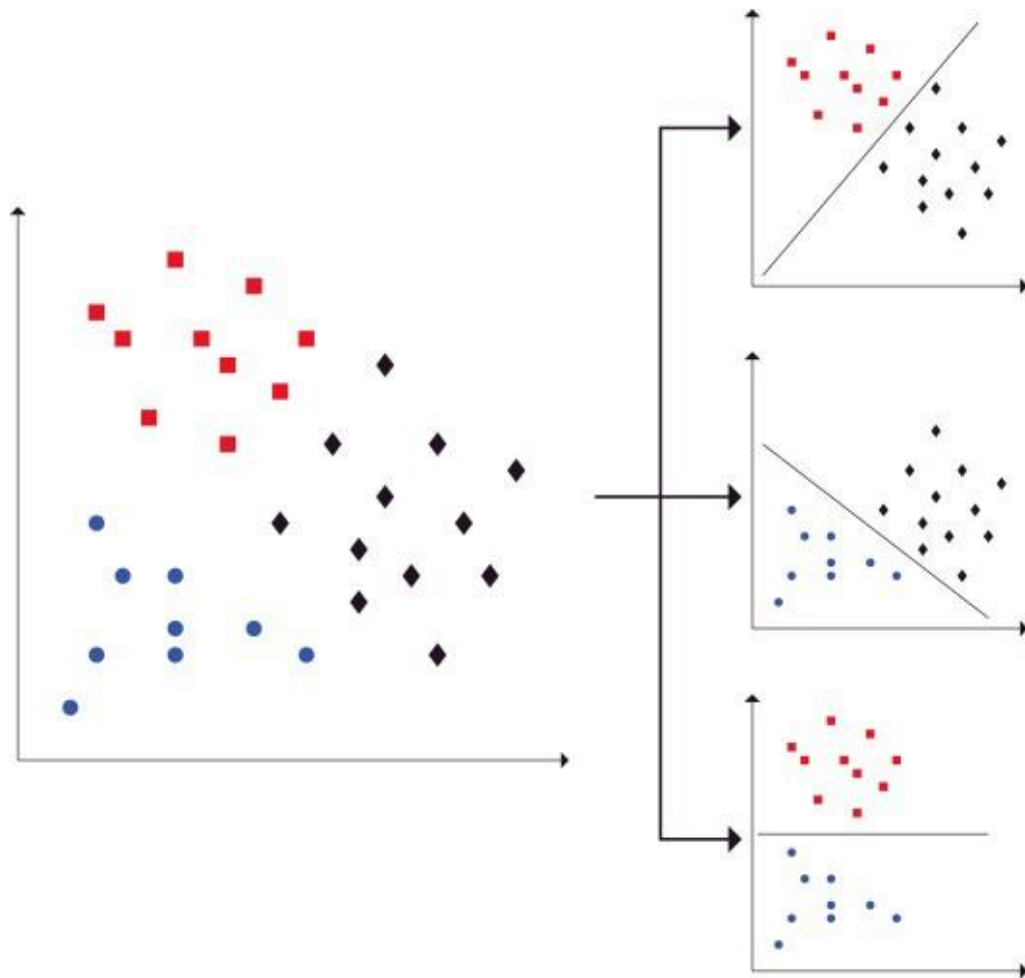
Dapat dilihat pada gambar 2- 4 diatas, data memiliki tiga kelas, misalnya kelas 1 tipe untuk hijau, tipe 2 untuk biru dan tipe 3 untuk merah. Seperti penjelasan sebelumnya bahwa teknik “Ovr” akan menghasilkan jumlah pengklasifikasian yang sama dengan label atau kelas yang ada pada dataset.

1. Klasifikasi 1 : [*Green*] vs [*Red, Blue*]
2. Klasifikasi 2 : [*Blue*] vs [*Green, Red*]
3. Klasifikasi 3 : [*Red*] vs [*Blue, Green*]

Namun dalam pembuatan *machine learning model* untuk dataset multi-kelas sudah sangat mudah digunakan karena secara *default* metode svm sudah menggunakan parameter “Ovr” ketika dataset yang di input multi-kelas.

2.2.6.2 *One vs One (Ovo)*

Dalam klasifikasi *One vs One (Ovo)*, untuk dataset *instance* kelas-N, harus membuat model pengklasifikasi *biner* $N*(N-1)/2$. Dengan menggunakan pendekatan klasifikasi ini, dataset primer akan dibagi menjadi satu dataset untuk setiap kelas yang berlawanan dengan setiap kelas lainnya.



Gambar 2- 5 Metode *One vs One (Ovo)*

(Band, 2020)

Dari plot data diatas, dapat diketahui bahwa data masalah klasifikasi yang memiliki tiga jenis kelas yaitu : Hijau, Biru dan Merah ($N=3$). Maka dalam metode *One vs One (Ovo)*, klasifikasi akan dibagi menjadi $N*(N-1)/2 = 3$ masalah klasifikasi biner :

1. Klasifikasi 1 : *Green vs Blue*
2. Klasifikasi 2 : *Green vs Red*
3. Klasifikasi 3 : *Blue vs Red*

Dalam metode *One vs One (Ovo)* setiap pengklasifikasian biner memprediksi satu label kelas. Ketika dimasukkan data uji ke pengklasifikasian, maka model dengan jumlah mayoritas disimpulkan sebagai hasil.

Berdasarkan penjelasan diatas, pada penelitian ini dataset berjenis multi-kelas dan algoritma klasifikasi menggunakan metode multi-kelas dengan teknik *One vs Rest (Ovr)*.

2.2.7 Validasi Silang (*Cross Validation*)

Cross Validation adalah metode pengambilan sampel dan evaluasi hasil klasifikasi. Selama pelatihan pada SVM perlu untuk memilih parameter penggali yang optimal dan nilai C. Pemilihan biasanya dilakukan dengan memperkirakan generalisasi. Metode yang umum digunakan untuk memperkirakan generalisasi adalah validasi silang (*Cross Validation*). Biasanya metode *cross validation* dapat digunakan untuk menentukan fungsi kernel. Sehingga dengan melakukan tahap *cross validation* bisa menjadi acuan untuk meningkatkan kualitas atau akurasi dari *machine learning model* (Ayungtyas, 2017).

Metode *cross validation* ini yaitu dengan membagi data menjadi K bagian yang teridentifikasi dan dengan ukuran/banyak (Aprilla, S, 2018). Proses dilakukan sebanyak k kali dan akan dicoba dengan beberapa nilai parameter dan nilai parameter terbaik ditentukan melalui *cross validation*. Pencarian nilai parameter dapat dilakukan dengan fungsi *grid search* yaitu mencari kombinasi secara *one by one*, kemudian pemilihan kombinasi tersebut didasarkan dari kombinasi yang akan memberkan akurasi klasifikasi terbesar.

Menurut Hastie al (2008) didalam tulisan (Aprilla, S, 2018), untuk mengukur tingkat kesalahan yang dapat terjadi dapat digunakan K sebanyak 5 s/d 10. Karena dengan K tersebut sudah cukup untuk membedakan data latih yang dimiliki.

2.2.8 Pengukuran Kinerja Atau Akurasi *Machine Learning Model*

Pengukuran kinerja model dibutuhkan untuk dapat mengetahui sistem yang dibuat dapat berjalan dengan baik sesuai dengan tujuannya. Nilai *accuracy* suatu sistem dapat dijadikan sebagai acuan dari keberhasilan *Machine Learning Model* yang dibuat. Semakin besar nilai *accuracy* nya maka kualitas suatu model juga semakin akurat (Aprilla, S, 2018).

Pengukuran kinerja klasifikasi pada data asli dan data model klasifikasi dilakukan menggunakan tabel silang atau yang biasa dikenal didalam SVM adalah *Confusion Matrix* yang berisi informasi tentang kelas data asli yang direpresentasikan pada deretan matriks dan kelas data yg diprediksi oleh suatu algoritma direpresentasikan pada kolom klasifikasi (Prasetyo, E., 2012). Berikut adalah contoh *Confusion Matrix* untuk tiga kelas :

Tabel 2- 2 *Confusion Matrix* Tiga Kelas

F_{gh}		Kelas Prediksi/Klasifikasi (h)		
		Kelas 1	Kelas 2	Kelas 3
Kelas Asli/Actual (g)	Kelas 1	F_{11}	F_{12}	F_{13}
	Kelas 2	F_{21}	F_{22}	F_{23}
	Kelas 3	F_{31}	F_{32}	F_{33}

Menurut (Prasetyo, E., 2012), ketepatan dalam klasifikasi dapat dilihat dari akurasi klasifikasi. Akurasi klasifikasi akan menunjukkan *performance* model klasifikasi secara keseluruhan. Semakin tinggi akurasi klasifikasi berarti semakin baik pula *performance* model klasifikasi.

$$\text{Akurasi} = \frac{F_{11} + F_{22} + F_{33}}{F_{12} + F_{13} + F_{21} + F_{23} + F_{31} + F_{32} + F_{11} + F_{22} + F_{33}}$$

$$\text{Error Klasifikasi} = \frac{F_{12} + F_{13} + F_{21} + F_{23} + F_{31} + F_{32}}{F_{12} + F_{13} + F_{21} + F_{23} + F_{31} + F_{32} + F_{11} + F_{22} + F_{33}}$$

Semua klasifikasi berusaha agar membentuk model dengan nilai *accuracy* yang tinggi dan *error* yang rendah

2.2.9 Python

Python adalah bahasa pemrograman komputer berbasis *interpreter* yang mudah dipelajari. *Python* dapat dikatakan sebagai bahasa pemrograman yang tinggi karena dapat melakukan eksekusi multi-guna secara bersamaan (*interpretatif*) dengan menggunakan metode *Object Oriented Programming* dan menggunakan semantik dinamis agar dapat memberikan tingkat keterbacaan

syntax. *Python* sebagai bahasa pemrograman yang tinggi dapat dipelajari dan dipahami dengan mudah karena sudah dilengkapi dengan memori secara otomatis. *Python* sebagai salah satu bahasa pemrograman yang sangat banyak digunakan untuk mengerjakan proyek yang berkaitan dengan *data science* dikarenakan *Python* menyediakan banyak *library* yang mudah digunakan untuk mendukung berbagai tugas seperti manipulasi *dataframe*, visualisasi, sampai proses pemodelan *machine learning*.



Gambar 2- 6 Logo *Python*
(*Foundation*, 2002 - 2022)

Python sudah digunakan di berbagai bidang pengembangan seperti pengembangan untuk website dan internet, penelitian ilmiah dan *numerik*, *data science* dan *big data*, media pembelajaran pemrograman, *graphical user interface* (GUI), pengembangan *software* serta untuk aplikasi bisnis. Sehingga *python* sebagai bahasa pemrograman yang tinggi terpopuler menduduki peringkat nomor lima yang paling banyak digunakan (Trisno, 2016).

2.2.10 Anaconda

Anaconda adalah sebuah platform bahasa pemrograman. *Python* yang bersifat *open-source*. Tujuan untuk menyederhanakan dalam manajemen paket *library* pemrograman *Python*. *Anaconda* memiliki *package installer* yang cukup handal dan memiliki *package* atau *library* yang sangat lengkap dan juga terupdate. Sehingga *anaconda* sangat direkomendasikan ketika menggunakan *python* sebagai bahasa pemrograman untuk *Machine Learning*.



Gambar 2- 7 Logo Anaconda
(Inc., 2022)

Aplikasi yang disediakan dalam distribusi anaconda terdiri dari :

1. *JupyterLab*
2. *Jupyter Notebook*
3. *QT Console*
4. *Spyder*
5. *VS Code*
6. *Glueviz*
7. *Orange 3*

2.2.11 *Jupyter Notebook*

Jupyter notebook adalah aplikasi web gratis yang dapat digunakan dalam membuat dan membagikan dokumen yang berisi kode, hasil perhitungan, visualisasi dan teks.



Gambar 2- 8 Logo *Jupyter Notebook*
(Notebook, 2022)

Jupyter adalah singkatan dari tiga bahasa pemrograman yaitu *Julia* (Ju), *Python* (Py) dan *R*. *Jupyter notebook* memiliki fungsi sebagai lembar kerja untuk membuat narasi komputasi yang menjelaskan tentang makna dari data

didalamnya. *Jupyter* dapat mempermudah para praktisi *data scientist* karena kemudahannya dalam menulis berbagai teks dan kode. Sehingga dapat mempermudah kolaborasi antara *data scientist*, *data researches* atau *data engineers* lainnya.

2.2.12 Getaran

Getaran merupakan gerakan bolak-balik dalam suatu interval waktu tertentu. Semua benda yang mempunyai massa dan elastisitas mampu bergerak, jadi kebanyakan mesin dan struktur rekayasa *engineering* mengalami gerakan sampai derajat tertentu dan rancangannya memerlukan pertimbangan sifat osilasinya (HUTAPEA, 2022).

Getaran sangat berhubungan dengan gerak osilasi beda dan gaya yang berhubungan dengan gerak tersebut. Getaran dapat terjadi karena beberapa hal, diantaranya adalah :

1. Adanya Massa
2. Adanya Kekakuan
3. Adanya gaya yang berasal dari dalam mesin dan gaya yang berasal dari luar

Salah satu contoh getaran yang sering dirasakan dalam kehidupan sehari-hari yaitu getaran pada sebuah motor. Dimana getaran yang terjadi di sebabkan oleh getaran mesin maupun permukaan jalan yang tidak rata.

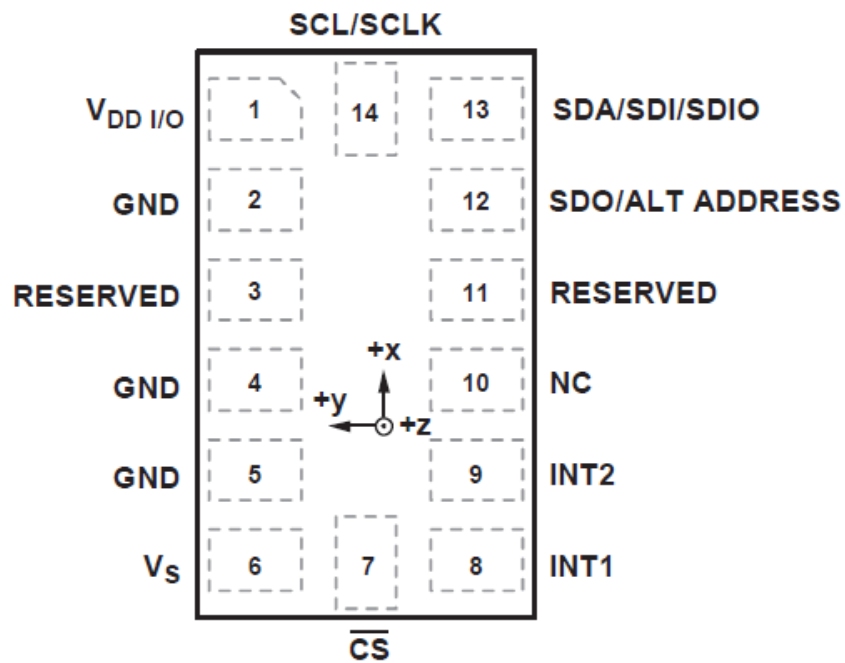
Getaran yang terjadi pada sebuah motor dapat menentukan keamanan maupun keselamatan pengendara. Efek dari sebuah getaran yang berlebihan akan menyebabkan suatu komponen akan lebih cepat aus atau rusak, selain itu getaran yang berlebihan juga akan mengurangi kenyamanan pengguna khususnya pada saat mengendarai sepeda motor. Dengan demikian, salah satu faktor kenyamanan pengendara sepeda motor tergantung dari seberapa besar getaran yang terjadi (SUHANDOKO, S., 2014). Semakin besar getaran yang terjadi pada sepeda motor, maka tingkat kenyamanan juga akan semakin berkurang.

2.2.13 Sensor Accelerometer ADXL345

Sensor *Accelerometer ADXL345* merupakan salah satu sensor percepatan yang menangkap respon berupa getaran serta kemiringan. *Accelerometer ADXL345* adalah modul sensor gera/akselerasi dengan tiga sumbu (*triple axis acceleration sensor module*) yang memiliki resolusi 13-bit ($2^{13} = 8194$ tingkat presisi) yang mampu mendeteksi hingga jangkauan 16g ($16 \times 9,81 \text{ m/s}^2$). Pengaplikasiannya mencakup untuk deteksi kemiringan dengan melihat perubahan gaya statik (*static gravity acceleration on tilt sensing application*) dan percepatan dinamik (*dynamic acceleration*) yang timbul diakibatkan gerakan atau berupa tumbukan (Adistya, 2014).

Pada penelitian yang dilakukan oleh Nayyar & Puri, 2016 dalam jurnal yang ditulis oleh (Liandana, M., dkk, 2019) menjelaskan modul sensor *ADXL345* memiliki tiga sumbu yaitu x, y dan z. Modul sensor *ADXL345* memiliki ukuran yang tipis, daya yang diperlukan rendah dan memiliki resolusi 13 bit. Resolusi yang dimiliki mampu mengukur perubahan sudut hingga dibawah 1 derajat (Liandana, M., dkk, 2019). Sensor *ADXL345* dapat digunakan dalam aplikasi perangkat seluler seperti *hadset seluler, smartphone*, perangkat game, perangkat penunjuk, perangkat navigasi pribadi, perlindungan *hard drive*, instrumentasi medis dan industri.

Sensor Accelerometer *ADXL345* dapat mengukur orientasi sudut dan lintasan objek. Sensor dapat melakukannya dalam tiga sumbu X, Y dan Z. Itu juga dapat mengukur kekuatan akselerasi dinamis dan statis. Gaya percepatan dinamis disebabkan oleh gerak dan getaran dan percepatan statis disebabkan oleh gravitasi. Rentang sensitivitas sensor akselerometer *ADXL345* adalah dari $+2g$ hingga $+16g$. (g kira-kira 9,8) Misalnya, ketika sensor ini ditempatkan pada permukaan datar, nilainya adalah $+1g$ pada arah sumbu z dan nol pada arah sumbu x dan y.



Gambar 2- 9 Diagram Pin Accelerometer ADXL345
(Agnihotri, 2016)

Sensor Accelerometer ADXL345 memiliki deskripsi atau fungsi pin yang berbeda-beda sebagaimana ditunjukkan pada tabel 2- 3.

Tabel 2- 3 Deskripsi Pin Accelerometer ADXL345

No	Nama Pin	Deskripsi Pin
1	V _{DD I/O}	Supply Voltage Antarmuka Digital
2	GND	Ground (-)
3	RESERVED	Reserved; harus terhubung ke VS atau dibiarkan terbuka
4	GND	Ground (-)
5	GND	Ground (-)
6	VS	Supply Voltage
7	CS	Untuk komunikasi I2C yang berfungsi sebagai Chip Select
8	INT1	Interupsi 1 Keluaran
9	INT2	Interupsi 2 Keluaran

No	Nama Pin	Deskripsi Pin
10	NC	Tidak terhubung secara internal
11	<i>RESERVED</i>	<i>Reserved</i> ; harus terhubung dengan Ground atau dibiarkan terbuka
12	SDO/ALT ADDRESS	<i>Serial data output (SPI 4-wire)/Alternate I2C Address Select (I2C)</i>
13	SDA/SDI/SDIO	<i>Serial data (I2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input Output (SPI 3-Wire)</i>
14	SCL/SCLK	<i>Serial Communications. SCL (I2C)/SCLK(SPI)</i>

1. Rentang Pengukuran Sensor ADXL345 (*Measurement range accelerometer ADXL345*)

Sensor *ADXL345* dapat mengukur akselerasi dalam tiga sumbu menggunakan rentang +/-2g yang dapat dipilih pengguna, +/-4g, +/-8g, dan +/-16g. Semakin tinggi rentang pengukuran, semakin tinggi akselerasi yang dapat dirasakan oleh akselerometer. Dengan rentang pengukuran +/-2g yang dipilih, *ADXL345* dapat mengukur akselerasi hingga 19,6 m/s² (2 * 9,8 m/s) di kedua arah di sepanjang setiap sumbu. Dengan rentang pengukuran +/-16g yang dipilih, *ADXL345* dapat mengukur akselerasi hingga 153,6 m/s² (16 * 9,8 m/s) di kedua arah di sepanjang setiap sumbu.

2. Resolusi Keluaran (*Output Resolution*)

Sensor *ADXL345* mendukung resolusi keluaran 10-bit untuk rentang pengukuran +/- 2g, 11-bit untuk +/- 4g, 12-bit untuk +/- 8g dan 13-bit untuk +/- 16g. Resolusi *default* adalah 10-bit untuk semua rentang pengukuran.

3. Sensitivitas (*Sensitivity*)

Resolusi *default* 10-bit, *ADXL345* memiliki sensitivitas tipikal 3,9 mg/LSB untuk rentang pengukuran *default* (yaitu +/- 2g), 7,8 mg/LSB untuk +/- 4g, 15,6 mg/LSB untuk +/- 8g dan 31,2 mg/LSB untuk rentang pengukuran +/- 16g. Artinya *ADXL345* dengan resolusi *default* 10-bit yang dipilih dapat mendeteksi perubahan akselerasi minimal 3,822 cm/s² (3,9 * 9,8/1000 * 100) untuk +/- 2g, 7,64 cm/s² untuk +/- 4g, 15,28 cm/s² untuk +/- 8g, dan 30,57 cm/s² untuk rentang +/- 16g.

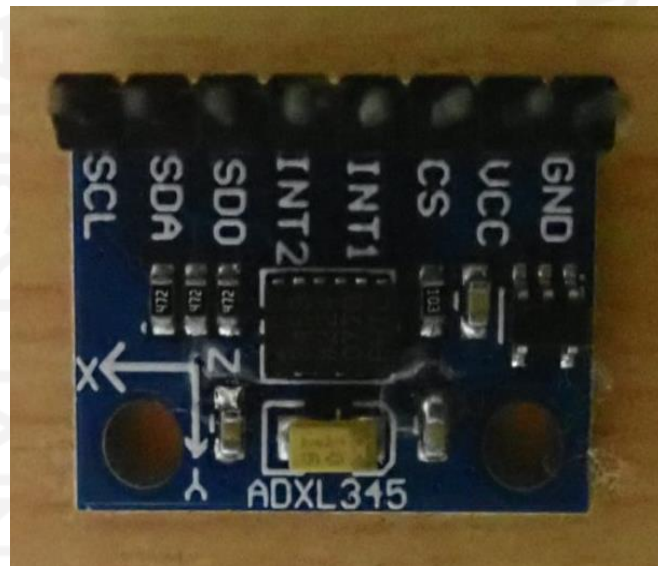
4. Output data rate dan bandwidth (*Output data rate and bandwidth*)

Kecepatan data keluaran sensor *ADXL345* dapat berkisar dari 0,1 Hz (sekali dalam 10 detik) hingga 3200 Hz (3200 kali per detik)

5. Tegangan dan arus operasi (*Operating voltage and current*)

Sensor *ADXL345* membutuhkan tegangan operasi 2.5V yang dapat berkisar dari 2.0V hingga 3.6V dan mengkonsumsi sekitar 30 uA untuk kecepatan transfer data kurang dari 10 Hz dan sekitar 140 uA untuk kecepatan transfer data di atas 100 Hz.

Gambar sensor *Accelerometer ADXL345* dapat dilihat pada gambar 2- 10.

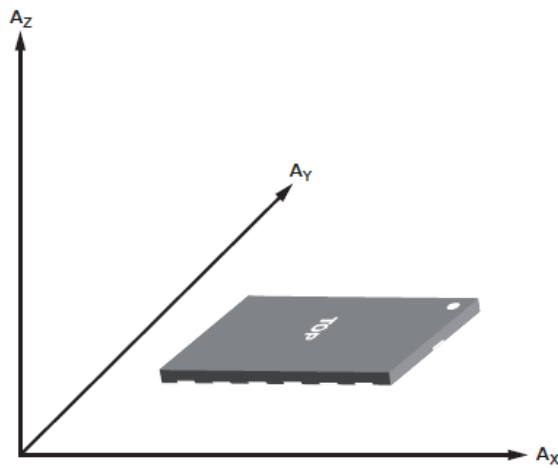


Gambar 2- 10 Sensor *ADXL345*

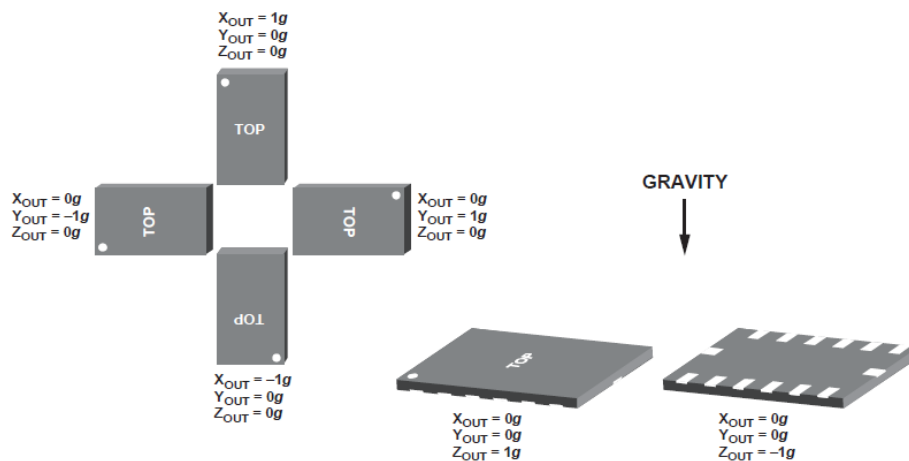
(Agnihotri, 2016)

Sensor *Accelerometer ADXL345* mungkin tersedia sebagai modul di mana semua atau beberapa pin mungkin tersedia untuk antarmuka dengan sirkuit. Sensor yang digunakan pada penelitian ini yaitu sebagaimana ditunjukkan pada gambar 2-10, yaitu hanya pin yang diperlukan untuk antarmuka dengan sirkuit (I2C, SPI, Interupsi, dan catu daya).

Sumbu akselerasi ditunjukkan pada modul yaitu pada gambar 2- 10 atau seperti yang ditunjukkan pada gambar 2- 11



Gambar 2- 11 Sumbu Sensor *Accelerometer ADXL345*
(Agnihotri, 2016)



Gambar 2- 12 Respon output *ADXL345* Sehubungan Dengan Orientasi Gravitasi
(Agnihotri, 2016)

Sensor *Accelerometer ADXL345* peka terhadap akselerasi statis (akselerasi karena gravitasi) maupun akselerasi dinamis (akselerasi akibat gerakan atau guncangan). Sehingga sensor memiliki respon keluaran, seperti yang ditunjukkan pada gambar 2- 12, sehubungan dengan orientasinya terhadap gravitasi.

2.2.14 Mikrokontroler

Mikrokontroler merupakan sebuah chip yang berperan sebagai pengontrol dalam rangkaian elektronik dan secara umum dapat menyimpan kode program dan juga didalamnya terdapat sistem *microprocessor* yang berperan sebagai sistem pengendalian. Secara umum, mikrokontroler terdiri dari CPU (*Central Processing Unit*), I/O tertentu dan uni-unit pendukung lainnya seperti *Analog-to Digital Converter* (ADC) yang sudah terintegrasi didalam mikrokontroler itu sendiri. Mikrokontroler bekerja sebagai alat yang bertugas untuk menjalankan intruksi-intruksi. Sehingga hal yang terpenting dalam mikrokontroler ini adalah kode program yang didalamnya karena mikrokontroler dapat menjalankan intruksi yang berupa kode program yang diberikan (Adistya, 2014).

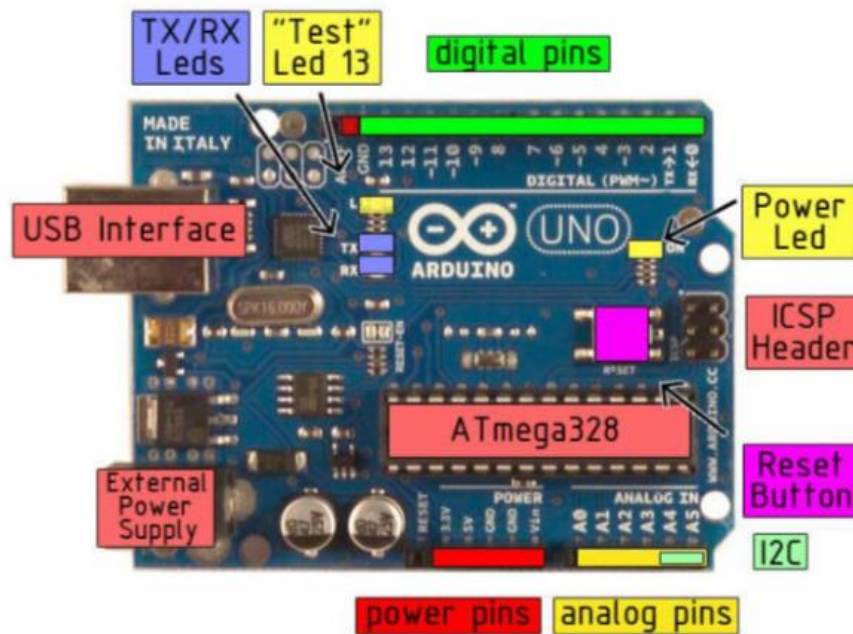
Pada penelitian yang dilakukan oleh (Adistya, 2014), perangkat elektronik yang menggunakan mikrokontroler dalam sistemnya terdapat beberapa kelebihan, diantaranya adalah :

1. Tidak membutuhkan daya yang besar
2. Ukurannya yang lebih kecil
3. Memiliki kemampuan yang tinggi dan mudah berinteraksi dengan komponen-komponen elektronik lainnya
4. Biaya produksi yang lebih rendah dikarenakan tidak memerlukan komponen yang banyak
5. Pembuatan bisa dilakukan secara cepat sehingga tidak memakan banyak waktu
6. Terdapat fasilitas tambahan dalam upaya pengembangan memori dan I/O untuk kebutuhan sistem.

2.2.15 Arduino Uno

Arduino Uno adalah sebuah board mikrokontroler yang didasarkan pada *Atmega328*. *Arduino Uno* memiliki 14 pin digital *input/output* (6 diantaranya dapat digunakan sebagai keluaran PWM), 6 masukan analog, sebuah isolator 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP *header*, dan sebuah tombol reset. *Arduino Uno* mampu *men-support* mikrokontroler dan dapat dikoneksikan dengan komputer menggunakan kabel USB (Mochtiarsa, 2014).

Arduino Uno berbeda dengan semua jenis arduino sebelumnya, dimana dalam hal koneksi USB to serial menggunakan fitur IC *Atmega8U2*, sementara pada board sebelumnya menggunakan chip *FDTI driver* USB to serial. *Arduino Uno R3* merupakan seri terakhir dan terbaru dari seri *arduino* USB.



Gambar 2- 13 *Arduino Uno R3*

(Arga, 2020)

Arduino Uno diaktifkan dengan menggunakan koneksi kabel USB atau dengan *power supply* eksternal secara otomatis. Untuk eksternal atau Non USB dapat menggunakan adaptor berupa arus AC ke DC atau dari baterai. Adaptor bisa dihubungkan dengan cara menghubungkan plug pusat positif 2,1 mm kedalam board colokan listrik. Sedangkan jika menggunakan baterai dapat dihubungkan dengan header pin GND dan Vin dari konektor power supply. *Board arduino* dapat beroperasi dengan besar daya antara 6-20 volt dan rentang daya yang dianjurkan yaitu sebesar 7 volt sampai dengan 12 volt (Mochtiarsa, 2014).

Adapun fitur atau spesifikasi data yang dimiliki oleh *board Arduino Uno R3* dapat ditunjukkan pada tabel 2- 4.

Tabel 2 - 4 Spesifikasi Data Teknik *Arduino Uno R3*

Mikrokontroler	<i>Atmega328</i>
Operasi Daya	5V
Input Tegangan (Rekomendasi)	7-12V
Input Tegangan Batas	6-20V
Digital I/O Pin	14 (6 sebagai output PWM)
Analog Input Pin	6
Arus DC setiap I/O Pin	40mA
Arus DC untuk 3,3V pin	50mA
<i>Flash</i> Memori	32 KB
SRAM	2 KB
EEPROM	1 KB
<i>Clock Speed</i>	16Mhz
Panjang	68.6 mm
Lebar	53.4 mm
Berat	25 g

Pada *Arduino Uno* terdapat juga beberapa pin yang memiliki fungsi khusus sebagaimana ditunjukkan pada tabel 2- 5.

Tabel 2- 5 Deskripsi Pin *Arduino UNO*

Pin	Keterangan
0 (RX/Receiver) dan 1 (TX/Transmitter)	Pin RX berfungsi sebagai pin untuk menerima (RX) dan pengirim (TX) TTL serial data. Pin ini tersambung dengan pin yang korespondensing dari USB ke TTL Chip
Eksternal Interupsi (Pin 2 dan 3)	Pin ini berfungsi sebagai konfigurator untuk <i>trigger</i> sebuah interupsi pada <i>value low, riding</i> dan <i>faling edge</i> atau

Pin	Keterangan
	nilai value yang berubah-ubah.
PWM : Pin 3,5,6,9,10 dan 11	Mensupport output 8 bit PWM dengan fungsi <i>analogWrite()</i> .
SPI : 10 (SS), 11(MOSI), 12 (MISO), 13 (SCK)	Pin ini berfungsi untuk mendukung komunikasi SPI dengan menggunakan <i>library SPI</i>
LED : 13	Terdapat LED bawaan (<i>built in</i>) yang berfungsi sebagai indicator dan terhubung dengan pin digital 13. Ketika nilai value pada pin bernilai <i>High</i> , maka LED akan menyala (on) dan sebaliknya jika nilai value <i>Low</i> , maka LED akan mati (off).
12C/A4(SDO) dan A5 (SCL)	Berfungsi sebagai pendukung komunikasi 12C
Vin	Tegangan yang berfungsi untuk masukkan kepada <i>board arduino</i> ketika menggunakan daya <i>external</i> .
GND	Berfungsi sebagai <i>ground</i> (-)
5V	Catu daya yang digunakan untuk daya mikrokontroler dan komponen lainnya
3v3	Berfungsi sebagai pasokan 3,3 volt dihasilkan oleh <i>regulator onboard</i>
Reset	Berfungsi untuk mereset mikrokontroler
IOfref	Referensi tegangan untuk input analog. Digunakan untuk <i>analogReference()</i>

Arduino menggunakan bahasa C sebagai bahasa untuk pemrogramannya, dimana setiap program *Arduino* (*sketch*) harus mempunyai dua buah fungsi, yaitu

:

1. `void setup() { }`

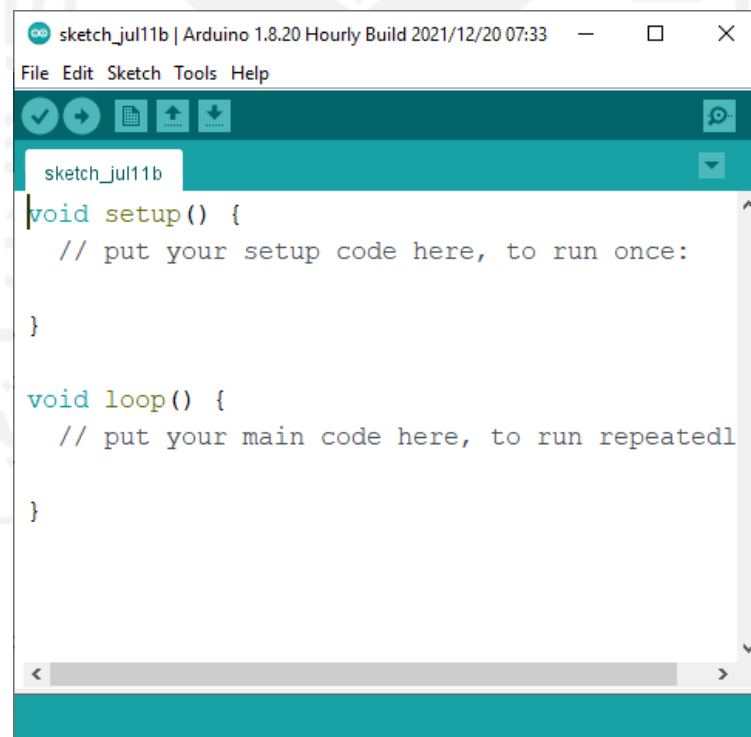
Semua kode berada didalam kurung kurawal { } akan dijalankan hanya satu kali ketika program Arduino dijalankan pada pertama kalinya.

2. `void loop () { }`

Fungsi `void loop` ini akan dijalankan setelah fungsi `void setup` selesai. Setelah dijalankan satu kali gungsi ini akan dijalankan ladi sampai seterusnya hingga power supply dilepaskan.

2.2.16 Arduino IDE 1.8.20

Arduino UNO mempunyai kelebihan tersendiri yaitu memiliki lingkungan khusus untuk permrograman yang disebut dengan *Arduino IDE*. Penggunaan IDE-nya pun tidak terlalu sulit karena kesederhanaan untuk programnya (Adistya, 2014). *Arduino IDE* juga terus dikembangkan dengan menghadirkan versi terbaru tentunya dengan fitur atau *library* yang lebih lengkap. Berikut adalah tampilan *Arduino IDE* versi 1.8.20



Gambar 2- 14 Halaman *Arduino IDE* Versi 1.8.20

Pada penelitian yang dilakukan oleh (Adistya, 2014), menjelaskan *Arduino IDE* adalah sebuah *software compiler* untuk pengembangan perangkat mikrokontroler *Arduino*. *Arduino IDE* dirancang untuk pengguna dengan berbagai keunggulan agar mudah untuk dipelajari bagi siapa yang membutuhkannya.

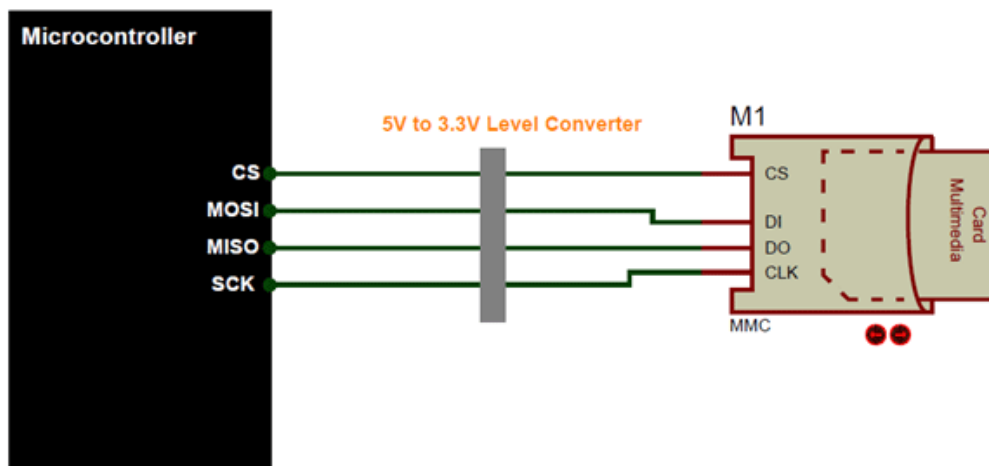
2.2.17 Modul *SD-Card*

Modul *Sd Card* sangat dibutuhkan untuk proyek yang memerlukan pencatatan data. Modul *SD-Card* yang dapat dihubungkan dengan *Arduino* atau rangkaian elektronik berbasis mikrokontroler, sehingga perangkat elektronik dapat mengakses data pada *SD-Card* (Pindrayana, K & dkk, 2018). *SD-Card* dapat bekerja dalam dua mode operasi, satu menggunakan perintah mode *SD* dan yang lainnya adalah mode *SPI*. Model modul *SD-Card* juga berbeda-beda tergantung dari pemasok produk *SD-Card* itu sendiri, namun prinsip kerja yang mendasarinya tetap sama. Modul yang digunakan pada penelitian ini untuk penyimpanan data yaitu menggunakan *data logger* dengan modul kartu micro *SD* dengan mode operasi *SPI* sebagaimana ditunjukkan pada gambar 2- 15 dan 2- 16.



Gambar 2- 15 Modul Micro *SD-Card*

(Components101, 2021)



Gambar 2- 16 Diagram Koneksi *SD-Card* Model SPI
(Components101, 2021)

Mode SPI hanya membutuhkan empat kabel dan sangat didukung oleh sebagian besar mikrokontroler. Namun *SD-Card* beroperasi dengan tegangan 3.3V dan semua pinnya berbicara hanya dengan 3.3V, disisi lain mikrokontroler mungkin bekerja dengan +5V dalam kasus tersebut pemindah level logika dua arah (seperti 74HC245) yang dapat mengubah sinyal 5V ke 3.3V. Berikut beberapa pin yang tersedia pada modul *SD-Card* mode SPI dan keterangannya :

Tabel 2- 6 Deskripsi Pin *SD-Card* Model SPI

Nama Pin	Type	Keterangan
CS	I (input)	Jalur kontrol untuk memilih pin (<i>chip select</i>)
DI	I (input)	Data In
DO	O/PP (<i>push-pull</i>)	Data Out
CLK/SCLK	I (input)	Clock

Pada penelitian ini getaran sebagai output dari alat pendeteksi getaran akan tersimpan di dalam kartu memori (*Sd-Card*) dalam bentuk file excel dengan format .csv. *Modul SD-Card* .

2.3 Statistika Deskriptif Data

Statistika adalah statistika yang digunakan untuk menggambarkan atau menganalisis data hasil penelitian, akan tetapi tidak untuk mengambil kesimpulan yang lebih luas terhadap ciri-ciri populasi (generalisasi/inferensi) (Nalim, N. & Salafudin, S.).

Statistika deskriptif adalah metode yang digunakan untuk pengumpulan dan penyajian data sehingga dalam penyajiannya dapat memberikan informasi yang dapat memudahkan pembaca untuk memahami data secara lebih mudah.

Statistika dalam pengertian sebagai ilmu dibedakan menjadi dua yaitu :

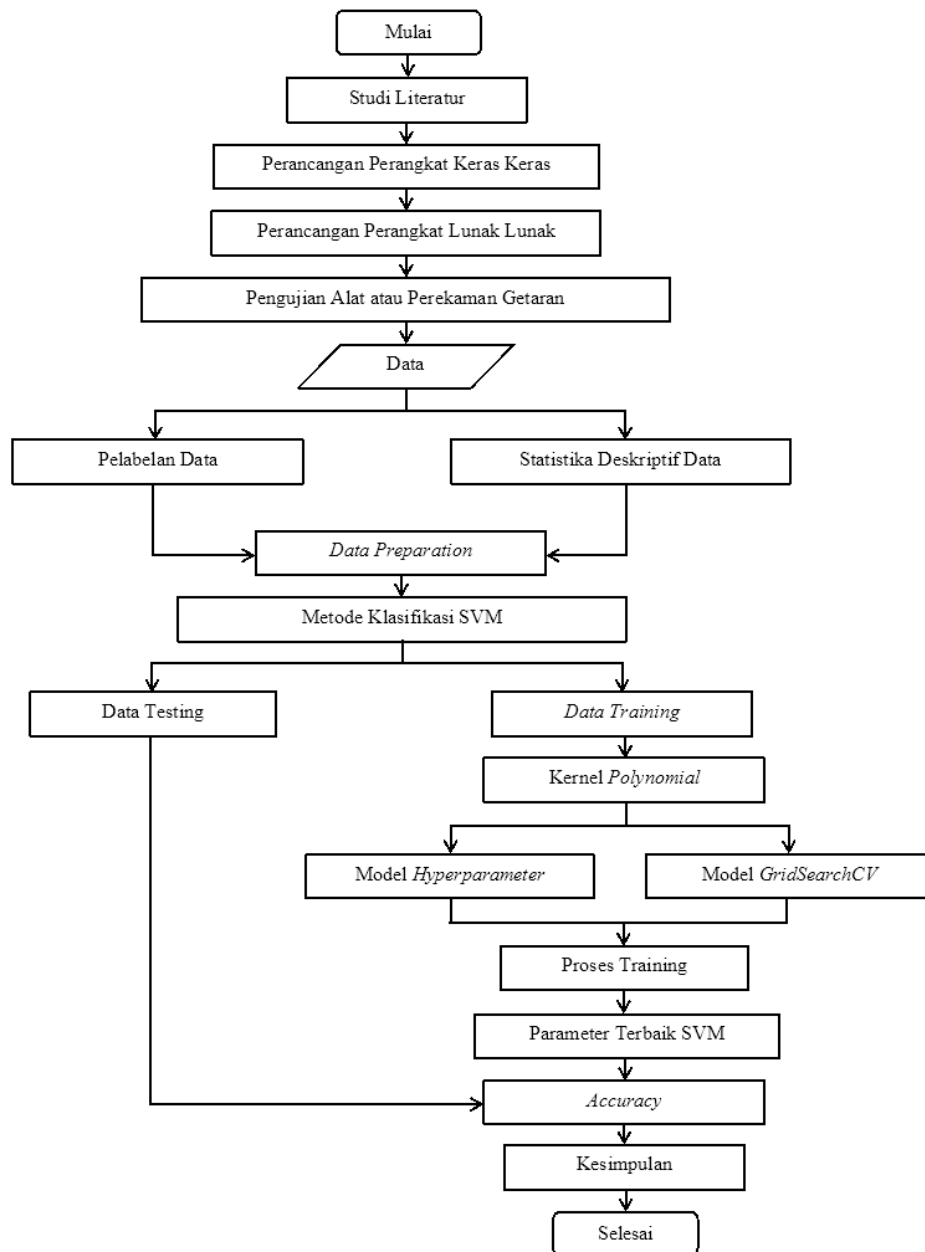
1. Statistika deskriptif mempunyai tujuan untuk mendeskripsikan atau memberikan sebuah gambaran dari objek yang diteliti. Dalam statistika deskriptif dikemukakan dengan cara penyajian data dalam bentuk tabel maupun diagram, penentuan rata-rata (*mean*), standar deviasi, modus, median, varian, simpangan baku dan nilai-nilai lainnya.
2. Statistika *inferensial* (induktif) mempunyai tujuan untuk penarikan kesimpulan. Sebelum penarikan kesimpulan dilakukan suatu dugaan yang dapat diperoleh dari *statistic* deskriptif.

BAB 3

METODE PENELITIAN

3.1 Langkah Penelitian

Langkah atau tahapan dalam penelitian ini adalah tahapan-tahapan yang dilakukan peneliti, hal ini dapat ditunjukkan pada flowchart gambar 3- 1.



Gambar 3- 1 Alur Penelitian

3.1.1 Studi Literatur

Studi literature adalah tahap pengumpulan informasi dari sumber yang dapat dipercaya dan relevan dengan penelitian yang dilakukan. Informasi yang didapatkan akan menjadi dasar untuk mencapai tujuan dari penelitian ini. Studi literatur dapat berupa jurnal, artikel, buku dan lain-lain sesuai dengan peraturan dari prodi Teknik Mesin Universitas Islam Indonesia.

Studi literature pada penelitian ini yaitu terkait pemrograman arduino, sensor *ADXL345* serta studi pustaka terkait Python, *Machine Learning*, Klasifikasi, dan *Support Vector Machine (SVM)*.

3.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras pada penelitian ini merupakan tahap perangkaian perangkat arduino dan sensor *Accelerometer ADXL345* untuk dapat merekam getaran motor beat pop saat dikendarai.

3.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini yaitu memprogram perangkat keras agar dapat merekam getaran dan menyimpan didalam memori *sd-card* dalam bentuk file dengan format *.csv* serta membuat program analisis dengan metode *Support Vector Machine (SVM)* klasifikasi.

3.1.4 Pengujian Alat atau Perekaman Getaran

Pengumpulan alat atau perekaman getaran merupakan tahap dimana peneliti akan melakukan pengumpulan data dengan cara menguji alat perangkat keras yang telah dirancang, sehingga menghasilkan data getaran yang akan di analisa dengan metode Klasifikasi *Support Vector Machine (SVM)*.

3.1.5 Pelabelan Data

Berdasarkan buku yang ditulis oleh (Scholkopf, B & Smola, A.J., 2002), algoritma SVM merupakan salah satu algoritma yang termasuk dalam kategori *Supervised Learning*, yang artinya data yang digunakan untuk dipelajari oleh

mesin merupakan data yang sudah memiliki label. Sehingga dalam proses penentuan keputusan, mesin akan mengkategorikan data *testing* ke dalam label yang sesuai dengan karakteristik data yang dipelajarinya.

Pelabelan data adalah tahap dimana peneliti akan melakukan pelabelan pada semua kumpulan data getaran yang telah didapatkan. Tujuan pada pelabelan data ini adalah untuk memberikan nilai kualitas getaran. Label data pada penelitian ini terdiri dari tiga label atau tiga kelas data yaitu kelas ‘Rendah’, kelas ‘Sedang’ dan kelas ‘Tinggi’. Metode pelabelan pada penelitian ini dilakukan secara manual terhadap semua data berdasarkan kualitas getaran yang terjadi saat mengendarai sepeda motor beat pop.

3.1.6 Statistika Deskriptif Data

Pada tahap ini, semua data hasil pengujian alat atau perekaman getaran akan dicari nilai statistika deskriptif untuk memudahkan dalam penyajian data hasil pengujian. Nilai statistika deskriptif yang digunakan pada penelitian ini yaitu nilai *mean*, *standar deviasi*, *median*, nilai minimal dan nilai maksimal.

3.1.7 Data Preparation

Data preparation merupakan tahap dimana peneliti akan melakukan pemisahan data yang akan dibagi menjadi dua bagian melalui variabel data yaitu data *input* (*independent variable*) atau x dan data *output* (*dependent variable*) atau y .

3.1.8 Metode Klasifikasi Support Vector Machine (SVM)

Metode implementasi model klasifikasi yang digunakan pada penelitian ini adalah menggunakan metode klasifikasi *Support Vector Machine* (SVM) dengan bahasa pemrograman Python. Dalam melakukan analisis dengan menggunakan SVM akan digunakan dengan fungsi *polynomial kernel* dengan menerapkan *Cross Validation* melalui library *GridSearchCV* kemudian akan didapatkan hasil akurasi dari parameter terbaik.

3.2 Peralatan dan Bahan

Peralatan dan bahan yang digunakan pada penelitian ini terdiri dari beberapa perangkat lunak dan perangkat yang diantaranya adalah sebagai berikut :

3.2.1 Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini terdiri dari beberapa *software* dan versi sebagaimana ditunjukkan pada tabel 3- 1.


Tabel 3- 1 Software Perangkat Lunak dan Versi

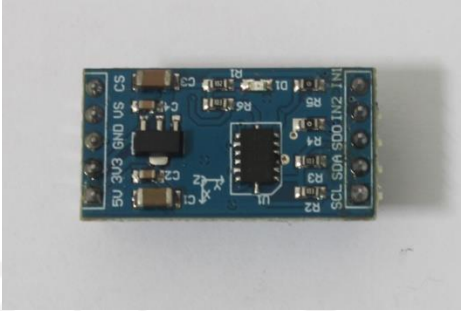

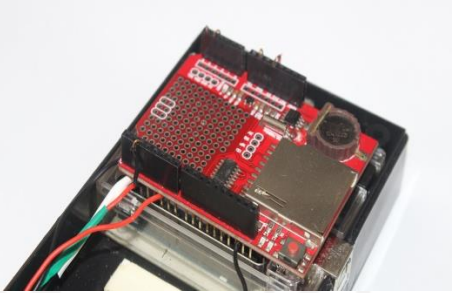

No	Software	Versi
1	Python	3.10.0
2	Anaconda	4.12.0
3	Jupyter Notebook	1.0.0
4	Arduino IDE	1.8.20

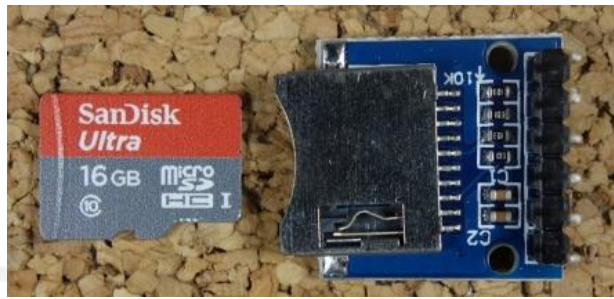
3.2.2 Perangkat Keras

Komponen perangkat keras pada penelitian ini terdiri dari beberapa komponen yaitu :

Tabel 3- 2 Peralatan dan Bahan

No	Nama Bahan	Gambar
1	Sepeda Motor Beat Pop	

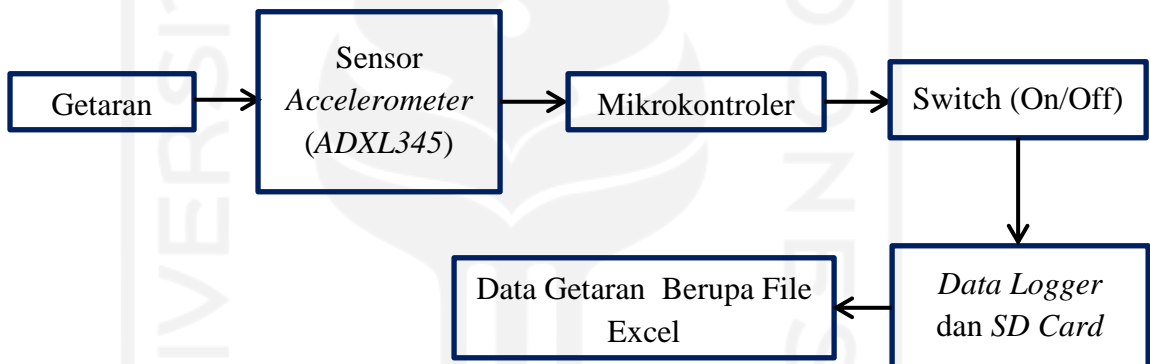
No	Nama Bahan	Gambar
2	<p>Sensor Accelerometer ADXL345</p>	
3	<p>Arduino Uno</p>	
4	<p>Data Logger Shield V.1.0 SD Card Reader Writer + RTC Module</p>	
5	<p>Power Bank 10.000 mAh + Kabel USB Arduino Uno</p>	

No	Nama Bahan	Gambar
6	Modul <i>Micro SD-Card</i>	

3.3 Perancangan Sistem

3.3.1 Desain Sistem

Desain sistem pada penelitian ini yaitu berupa sistem rancang bangun alat dengan alur sebagai berikut :



Gambar 3- 2 Alur Desain Sistem

Dapat dilihat dari gambar 3- 2 diatas, bahwa input dari pembacaan sensor adalah getaran motor beat pop yang dideteksi oleh sensor *Accelerometer ADXL345*. Setelah sensor *Accelerometer ADXL345* mendeteksi adanya getaran berupa nilai eksak akan diteruskan ke mikrokontroler. Mikrokontroler sebagai pusat utama yang mengatur sistem. Sebelum mikrokontroler meneruskan data getaran ke *data logger*, switch berfungsi sebagai intruksi 'if'. Jika switch On maka data getaran yang terdeteksi akan diteruskan ke *data logger* dan *sd card* untuk tersimpan dalam bentuk file dengan format .csv

3.3.2 Perancangan Perangkat Keras

Perancangan untuk implementasi sistem perekam getaran, maka dirancang suatu sistem perangkat keras yang mampu merekam getaran dan menyimpan data getaran tersebut ke dalam bentuk file excel dengan format .csv.

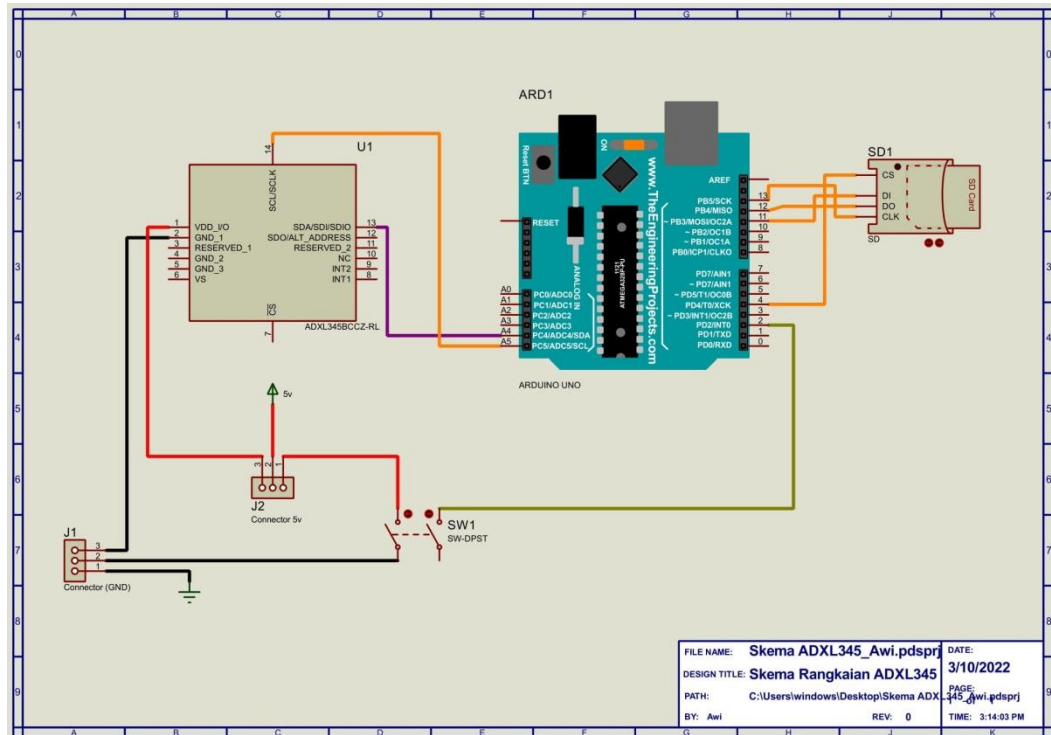
Data getaran dideteksi dengan menggunakan sensor *Accelerometer ADXL345*. Getaran yang terdeteksi akan tersimpan dalam bentuk file didalam modul *SD-Card*. Kemudian data getaran tersebut akan digunakan untuk implementasi klasifikasi data getaran dengan metode *Support Vector Machine (SVM)* menggunakan bahasa pemrograman *Python*.

Untuk perancangan perangkat keras perekam getaran pada penelitian ini dibutuhkan beberapa komponen khusus sebagaimana ditunjukkan pada tabel 3- 3.

Tabel 3- 3 Komponen Perancangan dan Fungsinya

Nama Komponen Perangkat Keras	Fungsi
<i>Arduino Uno R3 Atmega328P</i>	Komponen ini berfungsi sebagai pengendali sekaligus untuk memproses kode program yang telah ditulis
Sensor <i>ADXL345</i>	Komponen sensor ini berfungsi untuk mendeteksi ataupun mengukur getaran
Kabel Jumper	Komponen kabel jumper ini digunakan untuk menkoneksi antar perangkat
<i>Powerbank</i>	<i>Powerbank</i> pada perancangan perangkat keras penelitian ini digunakan sebagai sumber arus atau <i>power supply</i>
<i>Data Logger Shield</i>	Komponen ini berfungsi sebagai pencatatan data menggunakan transduser
Modul Micro <i>SD-Card</i>	Berfungsi untuk menyimpan data yang telah dideteksi oleh sensor <i>ADXL345</i>
<i>Switch (2 kaki)</i>	Berfungsi untuk menghubungkan dan memutuskan rekaman data yang tersimpan ke modul <i>sd-card</i>

Komponen pada tabel 3- 3 akan dirancang menjadi sebuah rangkaian perangkat keras yang dapat mendeteksi getaran dan menyimpannya dalam bentuk file didalam modul *Micro SD-Card*. Rangkaian rancangan perangkat keras pada penelitian ini dapat digambarkan dengan *software proteus versi 8.0* sebagaimana ditunjukkan pada gambar 3- 3.



Gambar 3- 3 Schematic Perangkat Keras

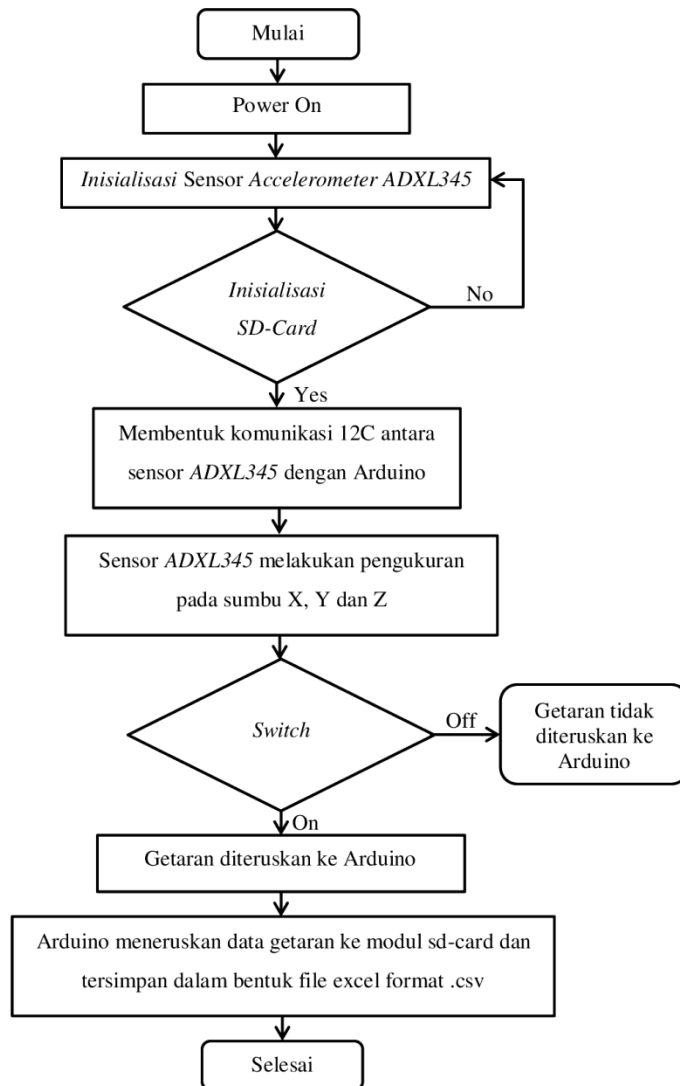
Pada gambar 3- 3 diatas dapat dilihat bahwa Arduino UNO harus terhubung dengan beberapa pin komponen lainnya, sebagaimana dapat dijelaskan sebagai berikut :

1. Pin VDD_I/O pada sensor *Accelerometer ADXL345* terhubung dengan pin 5V *Arduino UNO* kemudian terhubung ke salah satu kaki *switch (-)*. *Swicth* berfungsi untuk memutuskan ataupun menghubungkan *power supply* . Pin ini berfungsi untuk tempat *power supply* dari semua komponen
2. Pin *ground (GND)* pada sensor *Accelerometer ADXL345* terhubung dengan pin *ground (GND)* *Arduino UNO* kemudian dihubungkan dengan

salah satu kaki *switch* (-). Pin *ground* (GND) berfungsi sebagai tempat terminal *negative* (*ground*). Pin 2 pada *Arduino* UNO dihubungkan ke salah satu kaki *switch* (+) yang berfungsi sebagai eksternal interupsi.

3. Pin *analog* A5 (SCL) pada *Arduino* UNO terhubung ke pin 14 (SCL/SCLK) pada sensor *Accelerometer ADXL345*. Pin ini berfungsi sebagai serial komunikasi 12C
4. Pin digital (4) pada *Arduino* UNO dihubungkan ke pin CS pada *SD-Card* model SPI
5. Pin digital (11) *Arduino* UNO dihubungkan ke pin DI *SD-Card* mode SPI.
6. Pin digital (12) *Arduino* UNO dihubungkan ke pin DO *SD-Card* mode SPI.
7. Kemudian, pin digital (13) *Arduino* UNO dihubungkan ke pin CLK/SCLK *SD-Card* mode SPI.

Alir program *Arduino* UNO pada penelitian ini dapat ditunjukkan pada gambar 3-4.

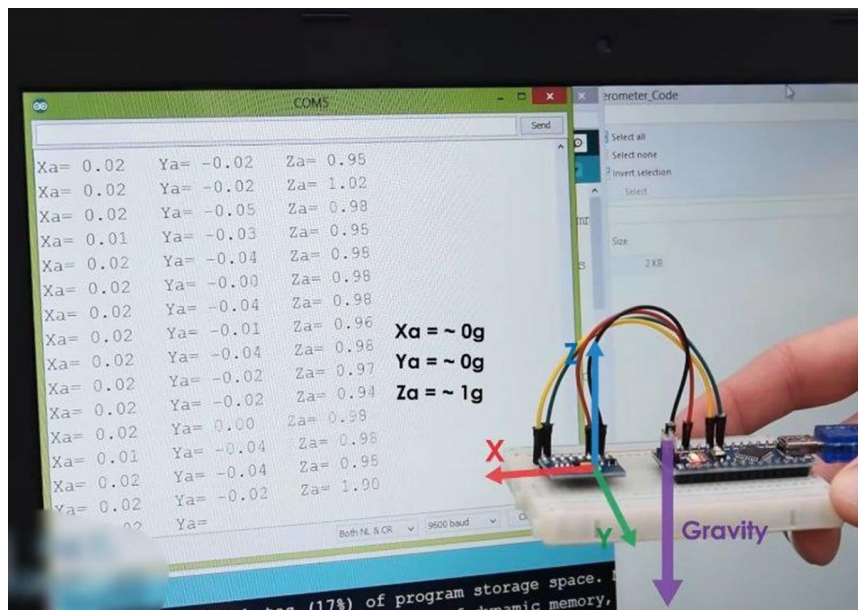


Gambar 3- 4 Diagram alir program dengan *Arduino* UNO

3.3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini dilakukan untuk membuat program berupa coding untuk mengoperasikan alat atau perangkat keras yang telah dirancang agar dapat mendeteksi dan merekam getaran pada saat motor beat pop dikendarai. Pada penelitian ini, untuk merancang perangkat lunak digunakan dengan bantuan *Software Arduino* UNO versi 1.8.20. Program perangkat lunak dapat dilihat pada lampiran 1.

3.3.4 Kalibrasi Sensor *Accelerometer ADXL345*



Gambar 3- 5 Kalibrasi Sensor *Accelerometer ADXL345*
(Dejan, 2020)

Sebelum melakukan pengujian alat atau perekaman data, maka akan dilakukan kalibrasi pada komponen sensor *Accelerometer ADXL345*. Cara kerja sensor *Accelerometer ADXL345* adalah sensor percepatan 3 sumbu yang dapat mengukur gaya akselerasi statis dan dinamis. Gaya gravitasi bumi merupakan contoh khas gaya statis, sedangkan gaya dinamis dapat disebabkan oleh getaran, gerakan dan sebagainya. Satuan ukuran untuk percepatan adalah meter per sekon kuadrat (m/s^2). Namun, sensor akselerometer menyatakan pengukuran dalam "g" atau gravitasi. Satu "g" adalah nilai gaya gravitasi bumi yang sama dengan 9,8 meter per detik kuadrat. Kalibrasi ini digunakan untuk meningkatkan akurasi sensor dan bekerja bersama dengan kerangka teori berdasarkan prinsip. Proses kalibrasi menggunakan metode sensor *Accelerometer ADXL345* yang diposisikan secara datar dan akan terlihat hasilnya di *serial monitor* dengan sumbu Z mengarah ke atas (berlawanan dengan gaya gravitasi), sehingga output sumbu Z dari sensor akan menjadi 1g ($\sim 1g$). Disisi lain, output X dan Y akan menjadi nol (~ 0), karena gaya gravitasi tegak lurus terhadap sumbu X dan Y.

3.3.5 Pengujian Alat atau Perekaman Getaran

Pengujian alat atau perekaman getaran dilakukan untuk menguji perangkat keras dan perangkat lunak apakah sudah bisa mendeteksi getaran yang terjadi dan menyimpannya dalam sebuah file. Adapun parameter pengujian alat atau perekaman getaran ditunjukkan pada tabel 3- 4.

Tabel 3- 4 Parameter Pengujian Alat atau Perekaman Getaran

No	Parameter Pengujian	Keterangan
1	Kecepatan	<i>low speed</i> (maximal 10 km/jam), <i>medium speed</i> (>10 km/jam/ <i>maximal</i> 20 km/jam) dan <i>high speed</i> (>20 km/jam/ <i>maximal</i> 30 km/jam).
2	Motor	Honda beat pop
3	Profil motor yang diuji	Bagian profil behel (pegangan belakang)

3.3.6 Setup Penempatan Perangkat Keras

Setup penempatan komponen perangkat keras untuk pengujian alat terdiri dari penempatan komponen *switch*, penempatan perangkat *Arduino* Uno, *power bank* dan penempatan perangkat sensor *ADXL345*. Supaya semua komponen dapat berfungsi dengan baik, maka antar komponen dihubungkan menggunakan kabel jumper. Penempatan *switch* diletakan dibagian dekat kaca *speedometer* motor seperti yang ditunjukkan pada gambar 3- 6.



Gambar 3- 6 Posisi *Switch*

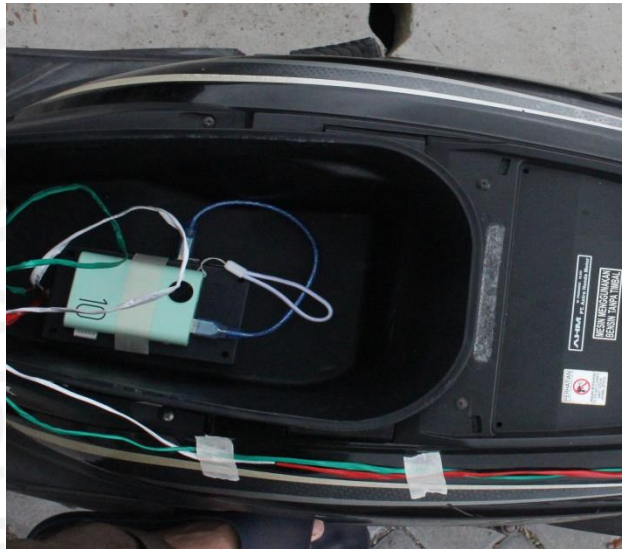
Mekanisme penempatan *switch* dilakukan dengan beberapa tahap yaitu *switch* dimasukkan ke dalam box dan direkatkan dengan lem lilin, kemudian box akan ditempelkan ke bagian sepeda motor menggunakan *double tape*. Pemilihan penempatan *switch* ini agar mempermudah pengemudi dalam memutus dan menghubungkan (*on/off*) perekaman data getaran. Saat posisi *switch on*, maka getaran yang dideteksi akan tersimpan ke dalam *SD-Card* dan sebaliknya ketika *switch off*, maka getaran yang terdeteksi tidak tersimpan ke dalam *SD-Card*.

Selanjutnya penempatan komponen *Arduino Uno* dan *power bank* sebagai sumber arus diletakkan di dalam bagasi sepeda motor dengan mekanisme penempatannya, komponen dimasukkan ke dalam sebuah box sebagaimana ditunjukkan pada gambar 3- 7.



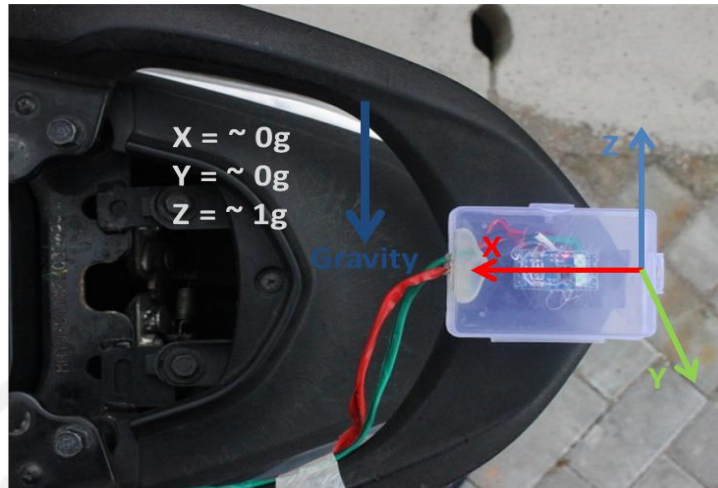
Gambar 3- 7 *Box Arduino Uno* dan *Powerbank*

Kemudian *box Arduino Uno* dan *powerbank* seperti yang ditunjukkan pada gambar 3- 7 akan ditempelkan didalam bagasi motor menggunakan *double tape* agar tidak bergerak saat sepeda motor dijalankan. Peletakan posisi perangkat *Arduino Uno* dan *powerbank* ditunjukkan pada gambar 3- 8.



Gambar 3- 8 Posisi *Arduino Uno* dan *Power Bank*

Pada penelitian ini, perekaman pada getaran motor beat pop dilakukan pada saat motor dikendarai dengan komposisi muatan terdiri dari pengemudi dan penumpang. Supaya pembacaan sensor getaran *ADXL345* semakin akurat dan getaran tidak teredam, maka peletakan posisi sensor *ADXL345* diletakkan dibagian behel (pegangan belakang) sepeda motor beat pop dengan mekanisme sensor *ADXL345* dimasukkan ke dalam box dan direkatkan dengan lem lilin, kemudian box akan ditempelkan ke bagian sepeda motor menggunakan *double tape*, peletakan sensor *ADXL345* ditunjukkan seperti pada seperti pada gambar 3- 9.



Gambar 3- 9 Posisi Sensor ADXL345

3.3.7 Lokasi dan Objek Pengujian Alat

Objek pengujian alat atau perekaman getaran dilakukan pada tiga objek kondisi permukaan jalan yang berbeda untuk mendapatkan nilai getaran yang bervariasi. Karena besarnya getaran yang terjadi pada sebuah motor saat dikendarai salah satunya disebabkan oleh kondisi permukaan jalan. Objek A yaitu permukaan jalan *paving block* berlokasi di Desa Jomblangan, Kec. Banguntapan, Objek B yaitu permukaan jalan perlintasan kereta api di Desa Tegal Tanda, Kec. Banguntapan dan Objek C yaitu permukaan jalan aspal tanpa lubang (halus) di daerah Jl. Laksda Adisucipto, Caturtunggal, Kec. Depok.



Gambar 3- 10 Objek A



Gambar 3- 11 Objek B



Gambar 3- 12 Objek C

3.4 Data

Data yang digunakan pada penelitian ini yaitu data getaran yang didapatkan dari hasil pengujian alat atau perekaman data getaran pada tiga objek permukaan jalan yang berbeda dan menggunakan kecepatan sepeda motor *low speed* (maximal 10 km/jam), *medium speed* (>10 km/jam/maximal 20 km/jam) dan *high speed* (>20 km/jam/maximal 30 km/jam). Setiap objek dilakukan sebanyak 15 kali pengujian, kecepatan *low speed* digunakan untuk pengujian ke 1 sampai dengan pengujian ke 5, kecepatan *medium speed* digunakan untuk pengujian ke 6 sampai dengan ke 10 dan kecepatan *high speed* digunakan untuk pengujian ke 11 sampai dengan pengujian ke 15, sehingga total pengujian pada semua objek yaitu sebanyak 45 kali pengujian. Hal ini dapat ditunjukkan pada tabel 3- 5.

Tabel 3- 5 Perekaman Getaran

Objek	Kecepatan Sepeda Motor	Perekaman	Banyaknya perekaman
A	<i>Low Speed (Maximal 10 km/jam)</i>	A1	15 kali
		A2	
		A3	

Objek	Kecepatan Sepeda Motor	Perekaman	Banyaknya perekaman
		A4	15 kali
		A5	
A	<i>Medium Speed</i> (>10 km/jam atau <i>maximal</i> 20 km/jam)	A6	
		A7	
		A8	
		A9	
		A10	
A	<i>High Speed</i> (>20 km/jam atau <i>maximal</i> 30 km/jam)	A11	
		A12	
		A13	
		A14	
		A15	
B	<i>Low Speed (Maximal 10</i> km/jam)	B1	
		B2	
		B3	
		B4	
		B5	
B	<i>Medium Speed</i> (>10 km/jam atau <i>maximal</i> 20 km/jam)	B6	
		B7	
		B8	
		B9	
		B10	
B	<i>High Speed</i> (>20 km/jam atau <i>maximal</i> 30 km/jam)	B11	
		B12	
		B13	
		B14	
		B15	
C	<i>Low Speed (Maximal 10</i> km/jam)	C1	15 kali
		C2	

Objek	Kecepatan Sepeda Motor	Perekaman	Banyaknya perekaman	
		C3		
		C4		
		C5		
C	<i>Medium Speed</i> (>10 km/jam atau <i>maximal</i> 20 km/jam	C6		
		C7		
		C8		
		C9		
		C10		
C	<i>High Speed</i> (>20 km/jam atau <i>maximal</i> 30 km/jam	C11		
		C12		
		C13		
		C14		
		C15		
Total jumlah banyaknya perekaman getaran				45 kali

Dari tabel 3- 5 dapat dilihat total jumlah pengujian alat atau perekaman getaran dari semua objek dan kecepatan yaitu sebanyak 45 kali perekaman. Artinya ada 45 file data getaran yang sudah tersimpan dalam bentuk file excel dengan format .csv yang akan dijadikan sebagai sampe data untuk diolah dengan metode Klasifikasi *Support Vector Machine* (SVM). File hasil perekaman getaran terdiri dari variabel tanggal, waktu, nilai getaran axis x, nilai getaran axis y dan nilai getaran axis z.

3.4.1 Pelabelan Data

Salah satu syarat pemilihan metode pengolahan data menggunakan jenis *machine learning supervised* yaitu label atau output data sudah diketahui. Sedangkan data yang dihasilkan pada pengujian alat yang dilakukan label datanya belum diketahui, sehingga perlu dilakukan pelabelan pada semua data yang dihasilkan pada pengujian alat.

Pelabelan data ini bertujuan untuk membantu model memahami, memproses dan belajar secara *komprehensif* dari pola data input. Seperti yang sudah dijelaskan pada bab sebelumnya yaitu pada bagian rumusan masalah, kelas data pada penelitian ini di isi secara manual dengan metode “*Self Assesment*” berdasarkan kualitas kenyamanan getaran yang terjadi saat mengendarai sepeda motor beat pop. Jadi, setiap perekaman data yang dilakukan akan diberikan label berdasarkan *self assesment* untuk menentukan kualitas dari suatu getaran yang terjadi pada saat sepeda motor beat pop dikendarai. Hasil pelabelan data dapat ditunjukkan pada tabel 3- 6.

Tabel 3- 6 Pelabelan Data

Objek	Kecepatan Sepeda Motor	Perekaman	Kelas/Label
A	<i>Low Speed (Maximal 10 km/jam)</i>	A1	2 = Tinggi
		A2	2 = Tinggi
		A3	2 = Tinggi
		A4	2 = Tinggi
		A5	2 = Tinggi
A	<i>Medium Speed (>10 km/jam atau maximal 20 km/ja)</i>	A6	1 = Sedang
		A7	1 = Sedang
		A8	1 = Sedang
		A9	1 = Sedang
		A10	1 = Sedang
A	<i>High Speed (>20 km/jam atau maximal 30 km/jam)</i>	A11	0 = Rendah
		A12	0 = Rendah
		A13	0 = Rendah
		A14	0 = Rendah
		A15	0 = Rendah
B	<i>Low Speed (Maximal 10 km/jam)</i>	B1	2 = Tinggi
		B2	2 = Tinggi
		B3	2 = Tinggi
		B4	2 = Tinggi

Objek	Kecepatan Sepeda Motor	Perekaman	Kelas/Label
		B5	2 = Tinggi
B	<i>Medium Speed</i> (>10 km/jam atau <i>maximal</i> 20 km/ja	B6	1 = Sedang
		B7	1 = Sedang
		B8	1 = Sedang
		B9	1 = Sedang
		B10	1 = Sedang
B	<i>High Speed</i> (>20 km/jam atau <i>maximal</i> 30 km/jam	B11	0 = Rendah
		B12	0 = Rendah
		B13	0 = Rendah
		B14	0 = Rendah
		B15	0 = Rendah
C	<i>Low Speed (Maximal 10</i> km/jam)	C1	2 = Tinggi
		C2	2 = Tinggi
		C3	2 = Tinggi
		C4	2 = Tinggi
		C5	2 = Tinggi
C	<i>Medium Speed</i> (>10 km/jam atau <i>maximal</i> 20 km/ja	C6	1 = Sedang
		C7	1 = Sedang
		C8	1 = Sedang
		C9	1 = Sedang
		C10	1 = Sedang
C	<i>High Speed</i> (>20 km/jam atau <i>maximal</i> 30 km/jam	C11	0 = Rendah
		C12	0 = Rendah
		C13	0 = Rendah
		C14	0 = Rendah
		C15	0 = Rendah

3.4.2 Statistika Deskriptif

Hasil nilai statistika deskriptif pada data semua pengujian dapat dilihat pada lampiran 2, terdiri dari nilai *mean*, nilai *standar deviasi*, nilai *median*, nilai minimal dan nilai *maximal*.

3.5 Data Preparation

Pada tahap ini dilakukan pembersihan variabel data sebelum di input ke *machine learning*. Adapun variabel data yang dibersihkan atau dihapus yaitu variabel 'Run', 'Count' dan 'Speed'. Sehingga variabel data input (*independent variable*) atau x terdiri dari lima belas variabel dan variabel data output (*dependent variable*) atau (y) terdiri dari satu variabel yaitu variabel Respon. Hal ini dapat ditunjukkan pada tabel 3- 7 dan tabel 3- 8.

Tabel 3- 7 Variabel Data Input (x) (*Independent Variable*)

Variabel	Nama Variabel Data
Data Input (x) (<i>Independent Variable</i>)	X_mean
	Y_mean
	Z_mean
	X_std
	Y_std
	Z_std
	X_min
	Y_min
	Z_min
	X_max
	Y_max
	Z_max
	X_median
	Y_median
	Z_median

Tabel 3- 8 Variabel Data Output (y) (*Dependent Variable*)

Variabel	Nama Variabel Data	Kelas/Label
Data Output (y) (<i>Dependent Variable</i>)	Respon	2 = Tinggi
		1 = Sedang
		0 = Rendah



BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan

Penelitian ini diawali dengan proses perancangan perangkat keras yang dapat merekam data getaran yang terjadi pada sepeda motor beat pop saat dikendarai. Dalam perangkat keras yang dirancang terdapat sensor *Accelerometer ADXL345*, *Arduino UNO R3*, *SD-Card*, *Switch* yang dihubungkan dengan kabel jumper dan power bank dihubungkan ke *Arduino UNO R3* melalui kabel USB. Berikut adalah penampakan dari perangkat keras yang dirancang :



Gambar 4- 1 Rangkaian Perangkat Keras

Perangkat keras yang dirancang dapat beroperasi jika *Arduino UNO R3* dihubungkan ke power bank menggunakan kabel USB. Karena power supply untuk *Arduino UNO R3* pada perancangan ini diberikan oleh *powerbank* dan akan aktif melakukan perekaman getaran apabila *switch* pada rangkaian di aktifkan (on).

Proses kerja dari perangkat keras ini adalah sebagai berikut :

1. Mikrokontroler *Arduino* UNO R3 yang telah terhubung dengan power bank dan telah diisi dengan program yang akan memberikan intruksi kepada sensor *Accelerometer ADXL345* untuk melakukan pengukuran atau pendeteksian getaran.
2. Sensor *Accelerometer ADXL345* akan mendeteksi data akselerasi dari getaran yang dideteksi berdasarkan vibrasi yang terjadi pada sepeda motor beat pop saat dikendarai. Data pengukuran yang diterima adalah nilai akselerasi pada sumbu x, y dan z.
3. Data yang diterima oleh sensor *Accelerometer* kemudian akan dikirim ke mikrokontroler *Arduino* UNO R3 dan kemudian akan dikirim ke *SD-Card* agar tersimpan dalam sebuah file excel dengan format .csv.
4. Selanjutnya data yang telah tersimpan di *SD-Card* akan dipindahkan ke windows untuk diolah dengan bahasa *Python* menggunakan metode Klasifikasi *Support Vector Machine* (SVM) untuk membentuk sebuah model *Machine Learning*.

4.2 Hasil Pengujian

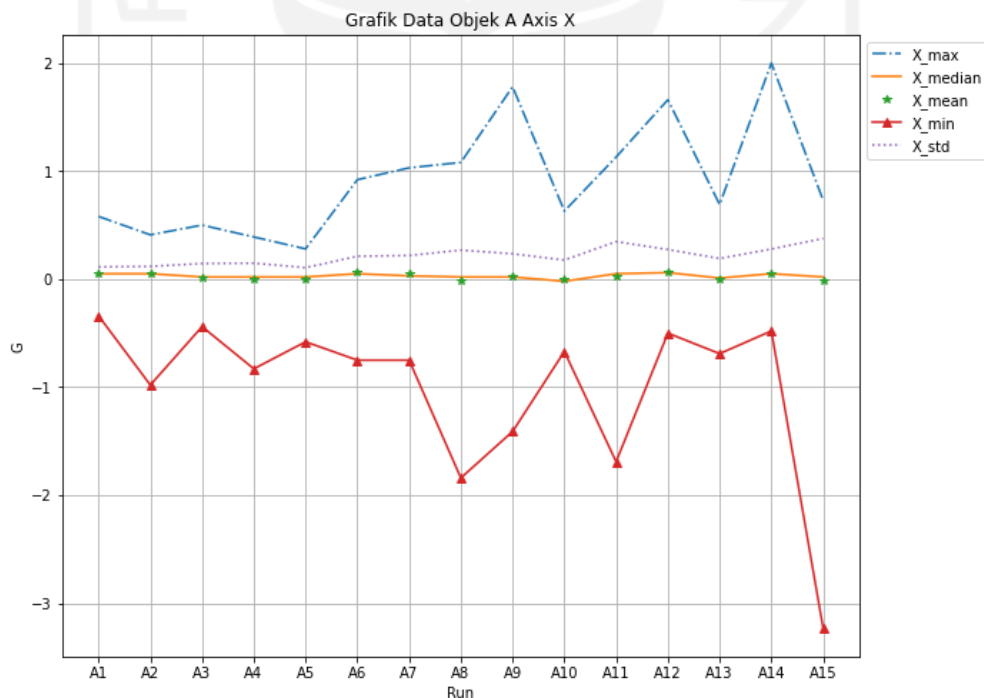
Berdasarkan pengujian alat didapatkan nilai-nilai getaran berdasarkan objek dan kecepatan sepeda motor yang divisualisasikan dengan salah satu *type* grafik dari *librray matplotlib* yaitu grafik bar. Nilai getaran yang sudah diolah dengan analisis statistika deskriptif dapat dilihat pada sumbu Y sedangkan pada sumbu X adalah variabel nilai data getaran. Seperti yang sudah dijelaskan pada bab sebelumnya yaitu pengujian alat dilakukan pada tiga objek jalan dengan permukaan yang berbeda dan menggunakan kecepatan sepeda motor *low speed* (*maximal* 10 km/jam), *medium speed* (>10 km/jam/*maximal* 20 km/jam) dan *high speed* (>20 km/jam/*maximal* 30 km/jam). Setiap objek dilakukan sebanyak 15 kali pengujian, kecepatan *low speed* digunakan untuk pengujian ke 1 sampai dengan pengujian ke 5 (A1 – A5), kecepatan *medium speed* digunakan untuk pengujian ke 6 sampai dengan ke 10 (A6 – A10) dan kecepatan *high speed* digunakan untuk pengujian ke 11 sampai dengan pengujian ke 15 (A11 – A15), sehingga total pengujian pada semua objek yaitu sebanyak 45 kali pengujian.

Hasil pengujian dapat ditunjukkan dengan visualisasi data berdasarkan nilai *mean*, nilai *standar deviasi*, nilai *median*, nilai minimal dan nilai *maximal* pada semua axis dan semua pengujian, data untuk visualisasi data hasil pengujian dapat dilihat pada lampiran 2.

4.2.1 Hasil Pengujian Objek A

Pengujian dilakukan dengan kecepatan sepeda motor 10 km/jam (*low speed*), 20 km/jam (*medium speed*) dan 30 km/jam (*high speed*). Kecepatan *low speed* digunakan untuk pengujian ke 1 sampai dengan pengujian ke 5 (A1 - A5), kecepatan *medium speed* digunakan untuk pengujian ke 6 sampai dengan pengujian ke 10 (A6 - A10) dan kecepatan *high speed* digunakan untuk pengujian ke 11 sampai dengan pengujian 15 (A11 - A15), sehingga total pengujian pada objek A yaitu 15 kali pengujian.

a) Hasil Pengujian Objek A Axis X

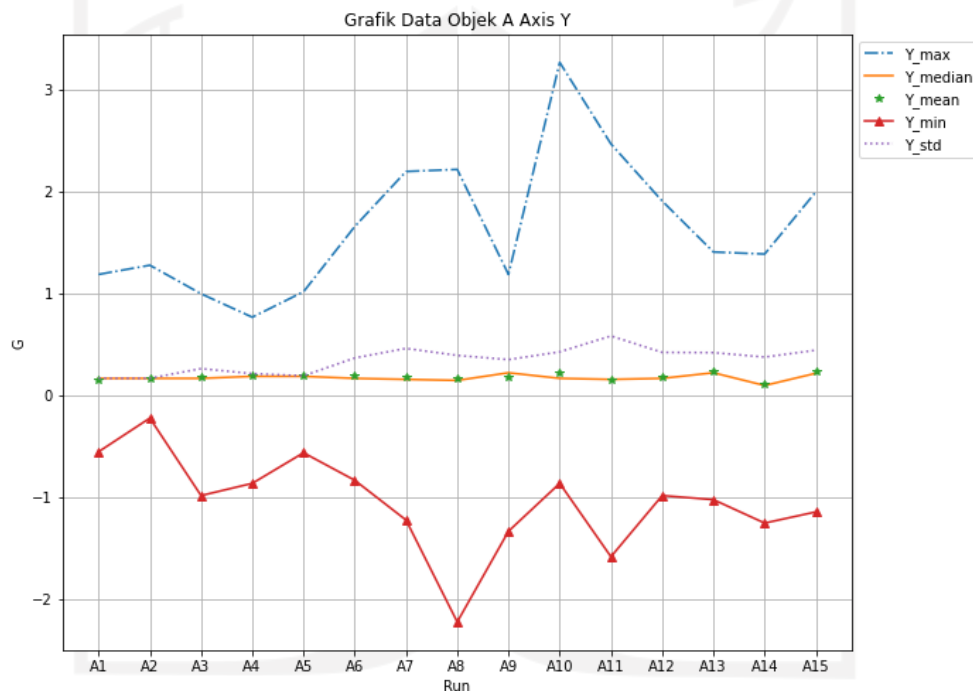


Gambar 4- 2 Grafik Data Hasil Pengujian Objek A Axis X

Gambar 4- 2 adalah grafik hasil pengujian alat pada objek A berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai

maximal axis X. Berdasarkan grafik 4- 2, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis X dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,064029 G terjadi pada pengujian ke 6 (A6) atau pada saat kecepatan sepeda motor *medium speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis X berdasarkan *standar deviasi* yaitu 0,375973 G. Kemudian nilai maximal terbesar yang terjadi pada axis X objek A yaitu 2 G, sedangkan nilai minimal terbesarnya yaitu -3,23 G.

b) Hasil Pengujian Objek A Axis Y

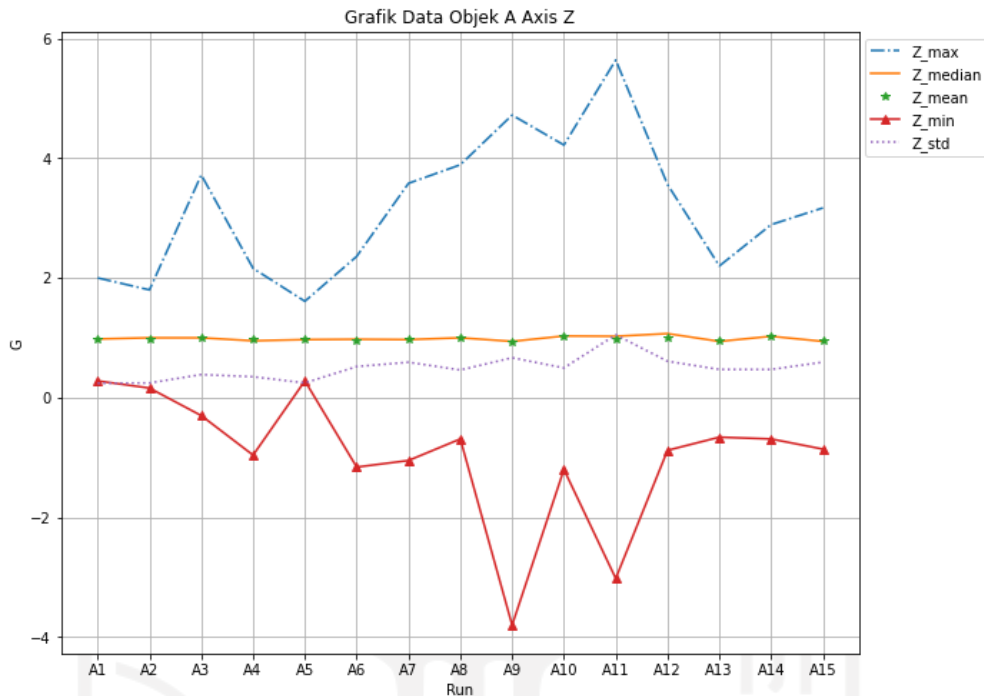


Gambar 4- 3 Grafik Data Hasil Pengujian Objek A Axis Y

Gambar 4- 3 adalah grafik hasil pengujian alat pada objek A berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal axis Y*. Berdasarkan grafik 4- 3, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis Y dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,239058 G terjadi pada pengujian ke 13 (A13) atau pada saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar

yang terjadi pada axis Y berdasarkan *standar deviasi* yaitu 0,586059 G. Kemudian nilai *maximal* terbesar yang terjadi pada axis Y objek A yaitu 3,27 G, sedangkan nilai minimal terbesar nya yaitu -2,22 G.

c) Hasil Pengujian Objek A Axis Z



Gambar 4- 4 Grafik Data Hasil Pengujian Objek A Axis Z

Gambar 4- 4 adalah grafik hasil pengujian alat pada objek A berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis Z. Berdasarkan grafik 4- 4, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis Z dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 1,02623 G terjadi pada pengujian ke 11 (A11) atau pada saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis Z berdasarkan *standar deviasi* yaitu 1,049778 G. Kemudian nilai *maximal* terbesar yang terjadi pada axis Z objek A yaitu 5,64 G, sedangkan nilai minimal terbesar nya yaitu -3,8 G.

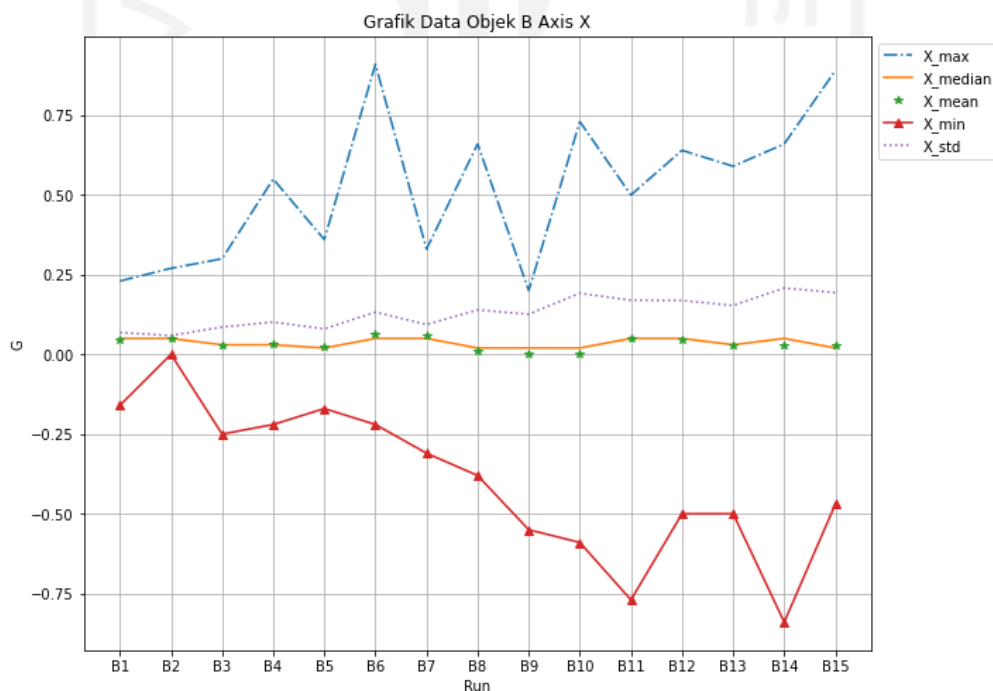
Berdasarkan grafik data hasil pengujian pada semua axis (X, Y dan Z) dapat diketahui bahwa pada axis X dan Y nilai *standar deviasi* nya selalu lebih

besar daripada nilai *mean* dan *median* nya, kecuali pada axis Z nilai standar deviasi nya lebih kecil daripada nilai *mean* dan *median* nya. Kemudian dari grafik hasil pengujian berdasarkan nilai maximal, dapat diketahui bahwa getaran yang terjadi pada axis Z relatif lebih besar daripada getaran yang terjadi pada axis X dan Y.

4.2.2 Hasil Pengujian Objek B

Pada pengujian Objek B kecepatan sepeda motor sama dengan kecepatan yang digunakan pada pengujian objek A yaitu *low speed*, *medium speed*, dan *high speed*. Kecepatan *low speed* digunakan untuk pengujian ke 1 sampai dengan pengujian ke 5 (B1 - B5), kecepatan *medium speed* digunakan untuk pengujian ke 6 sampai dengan pengujian ke 10 (B6 - B10) dan kecepatan *high speed* digunakan untuk pengujian ke 11 sampai dengan pengujian 15 (B11 - B15), sehingga total pengujian pada objek B yaitu 15 kali pengujian atau sama dengan total pengujian yang dilakukan pada objek B.

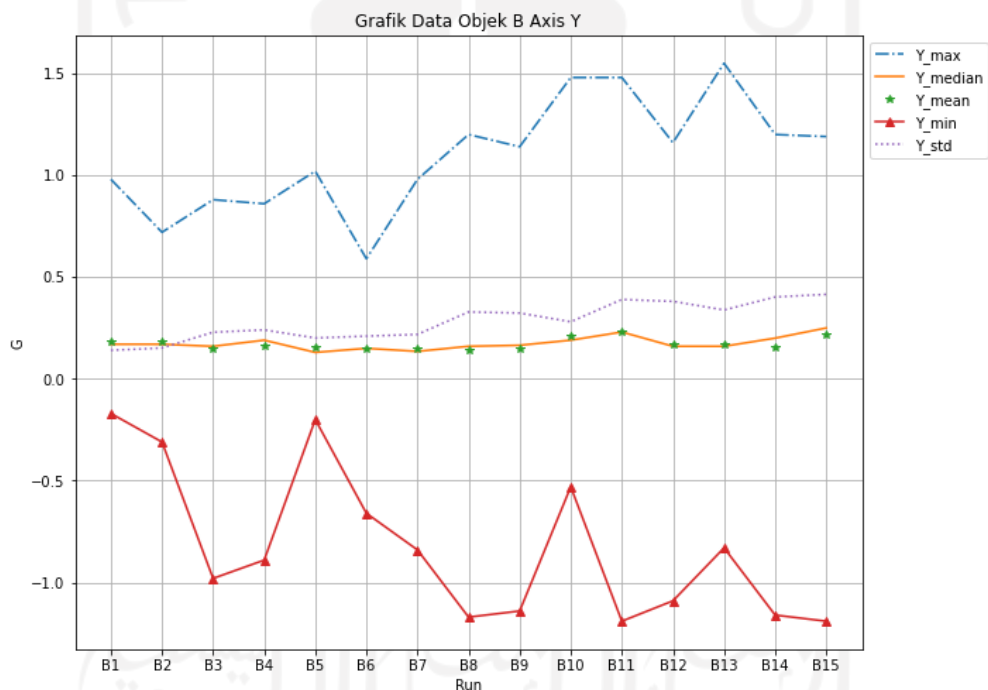
a) Hasil Pengujian Objek B Axis X



Gambar 4- 5 Grafik Data Hasil Pengujian Objek B Axis X

Berdasarkan grafik data nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* pada axis Z hasil pengujian objek B sebagaimana ditunjukkan pada gambar 4- 5, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis X dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,064603 G terjadi pada pengujian ke 6 (B6) atau pada saat kecepatan sepeda motor *medium speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis X berdasarkan *standar deviasi* yaitu 0,208282 G. Kemudian nilai maximal terbesar yang terjadi pada axis X objek B yaitu 0,91 G, sedangkan nilai minimal terbesar nya yaitu -0,84 G.

b) Hasil Pengujian Objek B Axis Y

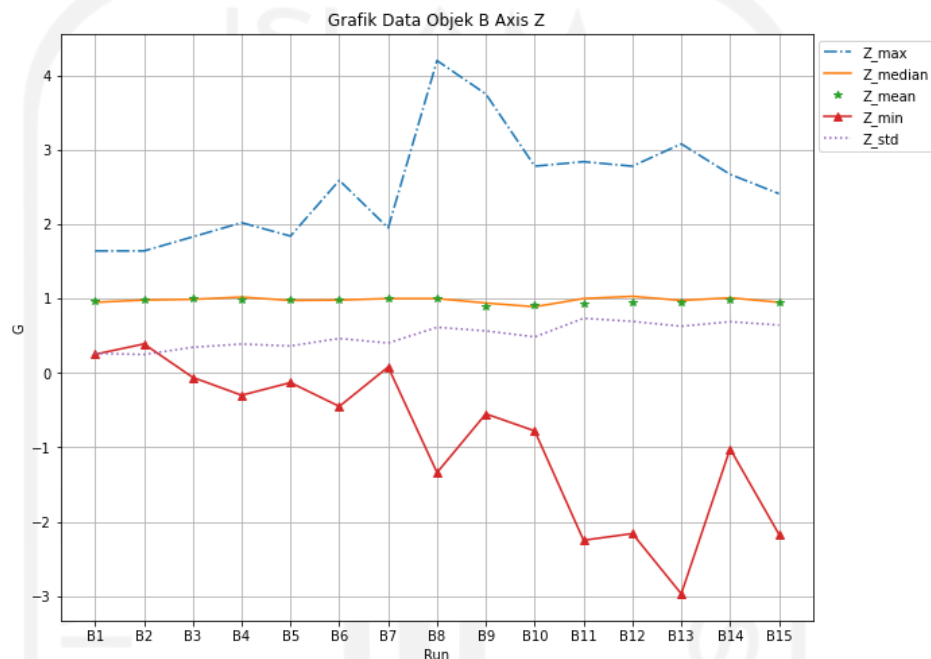


Gambar 4- 6 Grafik Data Hasil Pengujian Objek B Axis Y

Gambar 4- 6 adalah grafik hasil pengujian alat pada objek B berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis Y. Berdasarkan grafik 4- 6, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis Y dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,229655 G terjadi pada pengujian ke 11 (B11) atau pada

saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis Y berdasarkan *standar deviasi* yaitu 0,41498 G. Kemudian nilai *maximal* terbesar yang terjadi pada axis Y objek B yaitu 1,55 G, sedangkan nilai minimal terbesarnya yaitu -1,19 G.

c) Hasil Pengujian Objek B Axis Z



Gambar 4- 7 Grafik Data Hasil Pengujian Objek B Axis Z

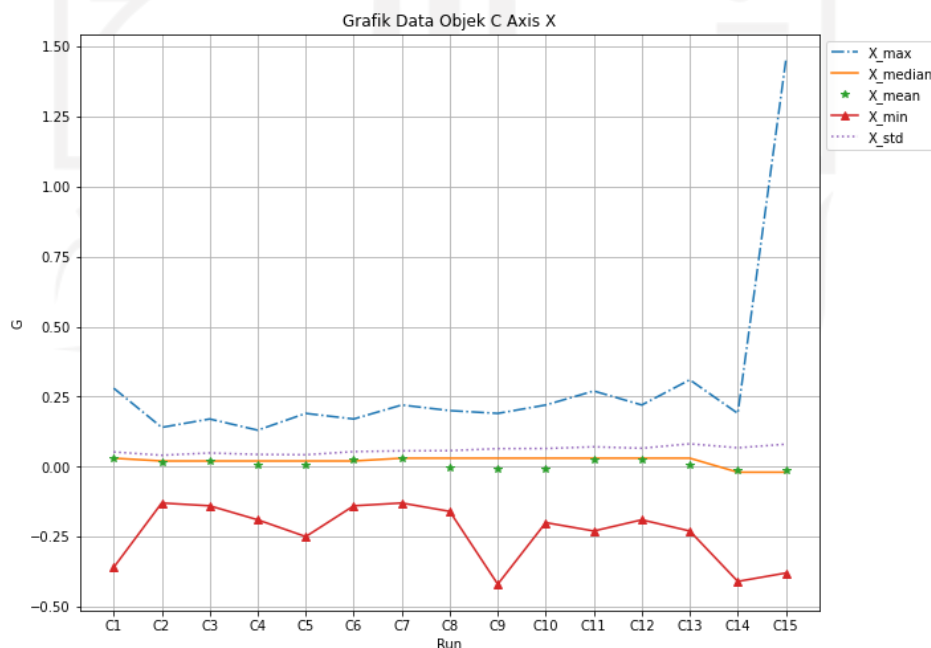
Gambar 4- 7 adalah adalah grafik hasil pengujian alat pada objek B berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis Z. Berdasarkan grafik 4- 7, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis Z dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 1,002143 G terjadi pada pengujian ke 7 (B7) atau pada saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis Z berdasarkan *standar deviasi* yaitu 0,734514 G. Kemudian nilai *maximal* terbesar yang terjadi pada axis Z objek B yaitu 4,2 G, sedangkan nilai minimal terbesarnya yaitu -2,97 G.

Berdasarkan grafik data hasil pengujian pada objek B semua axis (X, Y dan Z) dapat diketahui bahwa pada axis X dan Y nilai *standar deviasi* nya selalu lebih besar daripada nilai *mean* dan *median* nya, kecuali pada axis Z nilai standar deviasi nya lebih kecil daripada nilai *mean* dan *median* nya. Kemudian dari grafik hasil pengujian berdasarkan nilai *maximal*, dapat diketahui bahwa getaran yang terjadi pada axis Z relatif lebih besar daripada getaran yang terjadi pada axis X dan Y.

4.2.3 Hasil Pengujian Objek C

Pada pengujian Objek C kecepatan sepeda motor sama dengan kecepatan yang digunakan pada pengujian objek A dan objek B yaitu *low speed*, *medium speed*, dan *high speed*. Kecepatan *low speed* digunakan untuk pengujian ke 1 sampai dengan pengujian ke 5 (C1 - C5), kecepatan *medium speed* digunakan untuk pengujian ke 6 sampai dengan pengujian ke 10 (C6 - C10) dan kecepatan *high speed* digunakan untuk pengujian ke 11 sampai dengan pengujian 15 (C11 - C15), sehingga total pengujian pada objek C yaitu 15 kali pengujian atau sama dengan total pengujian yang dilakukan pada objek A dan objek B.

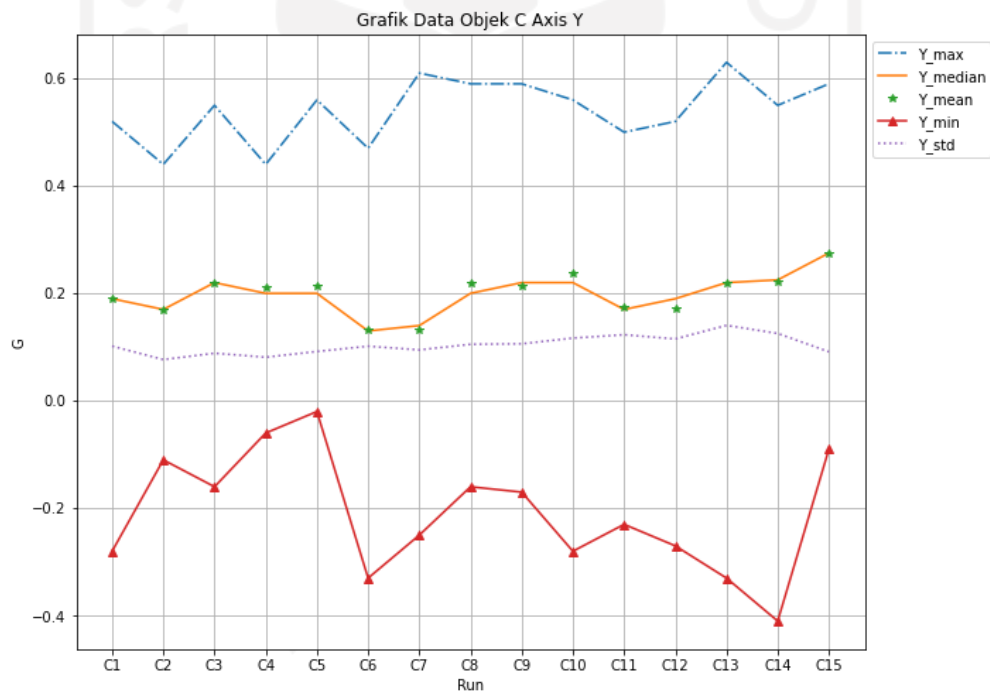
a) Hasil Pengujian Objek C Axis X



Gambar 4- 8 Grafik Data Hasil Pengujian Objek C Axis X

Gambar 4- 8 adalah adalah grafik hasil pengujian alat pada objek C berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis X. Berdasarkan grafik 4- 8, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis X dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,027746 G terjadi pada pengujian ke 7 (C7) atau pada saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis X berdasarkan *standar deviasi* yaitu 0,081357 G. Kemudian nilai maximal terbesar yang terjadi pada axis X objek Z yaitu 1,45 G terjadi pada pengujian ke 15 (C15), sedangkan nilai minimal terbesar nya yaitu – 0,42 G dan terjadi pada pengujian ke 9 (C9).

b) Hasil Pengujian Objek C Axis Y

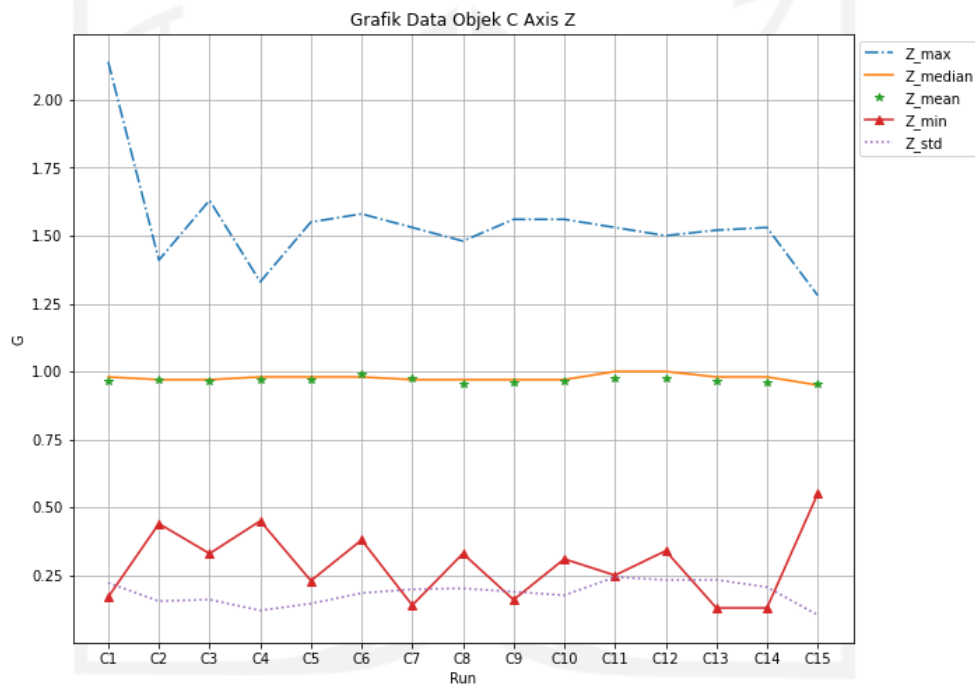


Gambar 4- 9 Grafik Data Hasil Pengujian Objek C Axis Y

Gambar 4- 9 adalah grafik hasil pengujian alat pada objek C berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis Y. Berdasarkan grafik 4- 9, dapat diketahui nilai getaran terbesar

berdasarkan nilai *mean* pada axis Y dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,272787 G terjadi pada pengujian ke 15 (C15) atau pada saat kecepatan sepeda motor *high speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai getaran terbesar yang terjadi pada axis Y berdasarkan *standar deviasi* yaitu 0,140412 G, namun nilainya selalu lebih rendah dibandingkan dengan nilai *mean* dan *median*. Kemudian nilai *maximal* terbesar yang terjadi pada axis Y objek C yaitu 0,63 G, sedangkan nilai minimal terbesarnya yaitu -0,41 G.

c) Hasil Pengujian Objek C Axis Z



Gambar 4- 10 Grafik Data Hasil Pengujian Objek C Axis Z

Gambar 4- 10 adalah adalah grafik hasil pengujian alat pada objek C berdasarkan nilai *mean*, nilai *median*, nilai *standar deviasi* (std), nilai minimal dan nilai *maximal* axis Z. Berdasarkan grafik 4- 10, dapat diketahui nilai getaran terbesar berdasarkan nilai *mean* pada axis Z dari pengujian 1 sampai dengan pengujian ke 15 dapat diketahui yaitu 0,989638 G terjadi pada pengujian ke 6 (C6) atau pada saat kecepatan sepeda motor *medium speed*, dan nilai *median* yang diperoleh juga tidak terlalu jauh beda dengan nilai *mean*. Kemudian nilai

getaran terbesar yang terjadi pada axis Z berdasarkan *standar deviasi* yaitu 0,243375 G. Selanjutnya nilai *maximal* terbesar yang terjadi pada axis Z objek C yaitu 2,14 G, sedangkan nilai minimal terbesar nya yaitu -0,13 G.

Berdasarkan grafik data hasil pengujian pada objek C semua axis (X, Y dan Z) dapat diketahui bahwa pada axis Y dan Z nilai *standar deviasi* nya selalu lebih kecil daripada nilai *mean* dan *median* pada axis X. Kemudian dari grafik hasil pengujian berdasarkan nilai *maximal*, dapat diketahui bahwa getaran yang terjadi pada axis Z relatif lebih besar daripada getaran yang terjadi pada axis X dan Y.

4.3 Analisis Dan Pembahasan

Prosedur analisis Klasifikasi *Support Vector Machine* (SVM) pada penelitian ini penentuan data input dan data output, metode klasifikasi *Support Vector Machine* (SVM), mengukur kinerja ketepatan klasifikasi model *machine learning* dan analisis *accuracy machine learning model*.

4.4 Metode Klasifikasi *Support Vector Machine* (SVM)

Metode analisis yang digunakan pada penelitian ini yaitu dengan menggunakan metode Klasifikasi *Support Vector Machine* (SVM). Dalam penerapan metode SVM data input dan data output akan dipisahkan menjadi *training dataset* dan *testing dataset*. *Training dataset* digunakan sebagai bahan latihan *machine learning model* dan *testing dataset* digunakan untuk menguji performa *machine learning model*. Sehingga data akan terbagi menjadi empat bagian (*dataset splitting*) yaitu data input *training* (x_{train}), data input *testing* (x_{test}), data output *training* (y_{train}) dan data output *testing* (y_{test}).

Training dataset diambil 80% dari keseluruhan total 45 data yaitu 36 data, sedangkan untuk *testing dataset* merupakan sisa dari data yang akan diuji akurasi nya yaitu 9 data. Pengambilan *training dataset* maupun *testing dataset* dilakukan secara acak dengan bantuan *software Python library train_test_split* dari modul *sklearn.model_selection*. Seperti yang sudah dijelaskan pada metode penelitian pada bab sebelumnya, akan digunakan fungsi SVM untuk mencari nilai

accuracy terbaik pada penerapan *polynomial kernel* dengan menggunakan metode *Grid Search Cross Validation*. Metode *Grid Search Cross Validation* adalah metode pemilihan model secara kombinasi dan *hyperparameter tuning* dengan cara menguji satu persatu kombinasi dan melakukan validasi untuk setiap kombinasi secara otomatis. Tujuannya adalah menentukan kombinasi yang menghasilkan satu performa model terbaik yang dapat dipilih untuk dijadikan parameter model klasifikasi.

4.4.1 *Polynomial Kernel SVM*

Parameter dari fungsi *Polynomial Kernel* terdiri dari *C* atau *Cost*, *Degree* (*d*) dan *Gamma* (γ). Pengoptimalan parameter fungsi kernel dapat dilakukan dengan menerapkan metode *Grid Search Cross Validation*. Tabel 4- 1 merupakan penentuan parameter terbaik untuk fungsi *Polynomial Kernel* dengan metode *Grid Search Cross Validation*.

Tabel 4- 1 Parameter Model SVM *Polynomial Kernel* Metode *Grid Search Cross Validation*

Parameter					
SVM - Type	SVM - Kernel	C (cost)	γ (gamma)	d (degree)	cv (Cross Validation)
C-Classification	Polynomial	C = 1	$\gamma = 1$	d = 3	cv = 5
		C = 5	$\gamma = 5$		
		C = 6	$\gamma = 6$		
		C = 7	$\gamma = 7$		
		C = 8	$\gamma = 8$		
		C = 9	$\gamma = 9$		
		C = 10	$\gamma = 10$		
		C = 15	$\gamma = 15$		
		C = 20	$\gamma = 20$		
		C = 25	$\gamma = 25$		

Parameter					
SVM - Type	SVM - Kernel	C (cost)	γ (gamma)	d (degree)	cv (Cross Validation)
		C = 26	$\gamma = 26$		
		C = 27	$\gamma = 27$		
		C = 28	$\gamma = 28$		
		C = 29	$\gamma = 29$		
		C = 30	$\gamma = 30$		

Pada tabel 4- 1 dapat dilihat kombinasi parameter *cost* dan *gamma* yang digunakan dalam analisis klasifikasi penelitian ini. Dengan metode *Grid Search Cross Validation* akan didapatkan satu parameter terbaik dari beberapa kombinasi parameter yang di input. Parameter terbaik yang didapatkan yaitu parameter C (*cost*) = 15, dan γ (*gamma*) = 1. Sehingga akan diperoleh parameter model *Support Vector Machine* (SVM) sebagai berikut :

Tabel 4- 2 Parameter *Machine Learning Model SVM Polynomial Kernel*

Parameter <i>Support Vector Machine</i> (SVM)	
SVM - Type	C - Classification
SVM - Kernel	Polynomial
C (<i>cost</i>)	15
γ (<i>gamma</i>)	1
d (<i>degree</i>)	3
cv (<i>cross validation</i>)	5

4.5 Ketepatan Klasifikasi Pada *Machine Learning Model SVM*

Berdasarkan parameter seperti yang ditunjukkan pada tabel 4- 2 dapat dibuat *counfusion matrix* yang bertujuan untuk mengukur kinerja atau ketepatan klasifikasi dari *machine learning model* dalam melakukan klasifikasi data yang di

input. *Confusion matrix* pada penelitian ini akan membentuk *confusion matrix* 3 x 3 karena data memiliki 3 kelas yaitu kelas 0 (Rendah), kelas 1 (Sedang) dan kelas 2 (Tinggi).

4.5.1 *Confusion Matrix Training Dataset*

Berikut adalah *confusion matrix* antara kelas prediksi dengan kelas aktual pada *training dataset*.

Tabel 4- 3 *Confusion Matrix Training Dataset*

Aktual	Kelas	Prediksi		
		Kelas		
		0	1	2
0	0	10	2	0
1	1	0	10	2
2	2	0	0	12

Pada tabel 4- 3 terdapat total 36 prediksi yang diperoleh dengan menganalisis *training dataset* untuk melihat kesesuaian model dalam mengklasifikasikan data dengan kelas sebenarnya. Dari total 36 *training dataset*, terdapat 10 data yang tepat klasifikasinya pada kelas 0 yaitu kelas kenyamanan Rendah dan terdapat 2 kelas data klasifikasi pada kelas 0 yang salah klasifikasi. Kemudian terdapat sebanyak 10 data yang tepat klasifikasinya pada kelas 1 yaitu kelas kenyamanan Sedang dan terdapat 2 data klasifikasi pada kelas 1 model salah dalam mengklasifikasi. Selanjutnya terdapat sebanyak 12 data juga yang tepat klasifikasinya pada kelas 2 yaitu kelas kenyamanan Tinggi dan tidak terdapat kelas data salah dalam mengklasifikasi. Sehingga, dari total 36 *training dataset* terdapat total 4 data model salah dalam mengklasifikasikan dan terdapat 32 data model secara tepat mengklasifikasikan kelas data dengan yang sebenarnya.

4.5.2 Confusion Matrix Testing Dataset

Model yang telah diperoleh dari *training dataset* akan digunakan untuk klasifikasi pada data baru yaitu *testing dataset*. Dalam menguji kinerja model pada *testing dataset* dapat dilakukan dengan pembuatan *confusion matrix* seperti yang dilakukan pada *confusion matrix training dataset*. *Confusion matrix* antara kelas prediksi dengan kelas aktual dari *testing dataset* dapat dilihat pada tabel 4-4.

Tabel 4- 4 *Confusion Matrix Testing Dataset*

Aktual	Kelas	Prediksi		
		Kelas		
		0	1	2
0	0	3	0	2
1	1	1	1	1
2	2	0	0	3

Pada tabel 4- 4 dapat dilihat dari total *testing dataset* yang digunakan yaitu 9 data, terdapat 3 klasifikasi data secara tepat pada kelas 0 dan terdapat pula 2 data klasifikasi yang salah yaitu kelas kenyamanan Rendah. Pada kelas 1 yaitu kelas kenyamanan Sedang terdapat 1 data diklasifikasikan secara tepat dan terdapat 2 data model salah dalam pengklasifikasian. Kemudian, pada kelas 2 yaitu kelas kenyamanan Tinggi terdapat 3 data secara tepat diklasifikasi dan tidak terdapat data yang salah diklasifikasikan oleh model. Sehingga, dari total 9 *testing dataset* terdapat 7 data secara tepat diklasifikasikan oleh model dan terdapat 2 data model salah dalam mengklasifikasikan.

4.6 Accuracy

Berdasarkan penelitian yang dilakukan oleh (Aprilla, S, 2018) menjelaskan bahwa nilai *accuracy* suatu sistem dapat dijadikan sebagai acuan dari keberhasilan *machine learning model* yang dibuat. Semakin besar nilai *accuracy* nya maka kualitas suatu model juga semakin akurat. Berdasarkan uraian tersebut, setelah diperoleh *confusion matrix* pada *training dataset* yaitu pada

tabel 4- 3 dan *testing dataset* pada tabel 4- 4, selanjutnya dapat dilakukan dengan mencari nilai *accuracy*.

Accuracy dari *training dataset* dan *testing dataset* dapat dilihat melalui *classification report* pada *machine learning model* yang telah dibuat dan dapat dirangkum kedalam tabel 4- 5 :

Tabel 4- 5 *Accuracy* Klasifikasi *Training Dataset* dan *Testing Dataset*

<i>Dataset</i>	<i>Accuracy</i>
<i>Training Dataset</i>	0,89 atau 89%
<i>Testing Dataset</i>	0,78 atau 78%

Dalam konsep *machine learning model*, model Klasifikasi SVM dilatih menggunakan *training dataset* dan untuk melihat *performance* nya akan dievaluasi dengan *testing dataset*. Seperti yang sudah dijelaskan pada bab sebelumnya analisis penelitian ini menggunakan metode *Support Vector Machine* (SVM) dengan fungsi *Polynomial Kernel*, didapatkan hasil bahwa model SVM mengklasifikasi data secara tepat dengan kelas yang sebenarnya yaitu 7 data dari total 9 *testing dataset* seperti yang ditunjukkan pada *confusion matrix* di tabel 4- 4 dengan nilai *accuracy* sebesar 78%.

Berdasarkan nilai akurasi *testing dataset* yaitu 78% dan ada 2 data dimana model salah dalam mengklasifikasikan, hal ini disebabkan karena beberapa hal, diantaranya adalah ukuran dataset terlalu sedikit. Selain itu kesalahan klasifikasi juga dapat disebabkan karena perbedaan rentang nilai dari variabel terlalu jauh, sehingga dalam algoritma SVM nilai variabel yang lebih tinggi akan lebih berpengaruh dibandingkan dengan nilai variabel lainnya. Untuk mengatasi hal tersebut salah satu metode yang bisa dilakukan terhadap variabel data yang memiliki rentang nilai yang terlalu jauh yaitu *feature scaling*. *Feature scaling* adalah proses membuat beberapa variabel data memiliki rentang atau skala nilai yang sama. Secara umum metode yang dapat dilakukan dalam metode *feature scaling* terdiri dari *simple feature scaling method*, *min-max method*, dan *Z-score method*.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengujian dan hasil analisis yang dilakukan pada penelitian ini, maka dapat ditarik beberapa kesimpulan, diantaranya adalah sebagai berikut :

1. Berhasil merancang alat yang mampu mengukur nilai getaran pada profil behel belakang sepeda motor bet pop dengan output nilai *eksak* dari sensor *Accelerometer ADXL345* tiga axis yang berupa data nilai getaran.
2. Didapatkan parameter terbaik pada metode Klasifikasi *Support Vector Machine* (SVM) dengan fungsi *Polynomial Kernel* yaitu C (*cost*) = 15, γ (*gamma*) = 1, d (*degree*) = 3 dan cv (*cross validation*) = 5
3. Berdasarkan analisis uji ketepatan Klasifikasi *Support Vector Machine* (SVM) menggunakan fungsi *Polynomial Kernel* dengan *training dataset* sebanyak 36 data dan *testing dataset* sebanyak 9 data, diperoleh hasil *accuracy machine learning model* sebesar 78% untuk *testing dataset* dan 89% untuk *training dataset*.

5.2 Saran atau Penelitian Selanjutnya

1. Pada penelitian ini, implementasi Klasifikasi *Support Vector Machine* (SVM) hanya menggunakan fungsi *Polynomial Kernel*. Sehingga pada penelitian selanjutnya bisa melakukan implementasi Klasifikasi menggunakan fungsi kernel lainnya seperti *Linear kernel*, *Radial Basic Function* (RBF) *kernel*, dan *Sigmoid kernel*.
2. Saran selanjutnya yaitu *training dataset* dan *testing dataset* sebaiknya lebih diperbanyak lagi dan melakukan *feature scalling* terlebih dahulu terhadap variabel data yang memiliki rentang nilai yang terlalu jauh, kemudian dapat mencoba dengan kombinasi parameter yang lebih banyak lagi.

DAFTAR PUSTAKA

- Adisty, I. S. (2014). Pengembangan sistem monitoring vibrasi pada kipas pendingin menggunakan accelerometer ADXL345 dengan metode FFT berbasis labview.
- Agnihotri, N. (2016, Juny 12). *engineersgarage.com*. Retrieved Juni 02, 2022, from How to use the ADXL345 accelerometer sensor:
<https://www.engineersgarage.com/adxl345-accelerometer-sensor-how-to-use/>
- Alloghani, M.dkk. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and unsupervised learning for data science*, 3-21.
- Amei, W.dkk. (2011). A survey of application-level protocol identification based on machine learning. In 2011 International Conference on Information Management, Innovation Management and Industrial Engineering. *IEEE*, 201-204.
- Amr, T. (2020). *Hands-On Machine Learning with scikit-learn and Scientific Python Toolkits: A practical guide to implementing supervised and unsupervised machine learning algorithms in Python*. Packt Publishing Ltd.
- Andreasnus, J., & Kurniawan, A. (2017). Sejarah, Teori Dasar dan Penerapan Reinforcement Learning: Sebuah Tinjauan Pustaka. *Jurnal Telematika*, 12(2), 113-118.
- Aprilla, S. (2018). Klasifikasi Penyakit Skizofrenia dan Episode Depresi Pada Gangguan Kejiwaan Dengan Menggunakan Metode Support Vector Machine (SVM). (*Doctoral dissertation, Universitas Brawijaya*), 13.
- Arga. (2020, Juli 10). *Pengertian Arduino Uno dan Spesifikasinya*. Retrieved Juni 02, 2022, from pintarelektro.com:
<https://pintarelektro.com/pengertian-arduino-uno/>
- Ayungtyas, D. A. (2017). Klasifikasi Menggunakan Metode Regresi Logistik Dan Support Vector Machine. (*Doctoral dissertation, Universitas Brawijaya*), 5-6.

- Band, A. (2020, Mei 10). *Multi Class Classification One vs All & One vs One*. Retrieved Juni 01, 2022, from towardsdatascience.com: <https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>
- Bishop.dkk. (2006). *Pattern recognition and machine learning (Vol. 4, No. 4, p. 738)*. New York: Springer.
- Boswell, D. (2002). Introduction to support vector machines. *Departement of Computer Science and Engineering University of California San Diego*.
- Brownlee, J. (2016). Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch.
- Components101. (2021, July). *Micro SD Card Adapter Module*. Retrieved Juni 03, 2022, from components101.com: <https://components101.com/modules/micro-sd-card-module-pinout-features-datasheet-alternatives>
- Cortes,dkk. (1995). "Support-vector networks". *Machine learning* 20.3. 237-297.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.
- Dejan. (2020, October 06). *How To Track Orientation with Arduino and ADXL345 Accelerometer*. Retrieved October 02, 2022, from howtomechatronics.com: <https://howtomechatronics.com/tutorials/arduino/how-to-track-orientation-with-arduino-and-adxl345-accelerometer/>
- Dima, C. (2016, September 19). *Basics Of Support Vector Machines*. Dipetik Mei 15, 2022, dari cristiandima.com: <https://www.cristiandima.com/basics-of-support-vector-machines>
- Foundation, P. S. (2002 - 2022). *The Python Logo*. Dipetik Juni 01, 2022, dari python.org: <https://www.python.org/community/logos/>
- Goldberg, D.E., & Holland, J.H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2), 95-99.
- HAKIM, & Ayu Aulia. (2009.2017). *Pemodelan dan analisis pengaruh perubahan parameter variable orifice sistem suspensi hidrolis terhadap*

- gaya redam yang dihasilkan dan respon dinamis penumpang pada sepeda motor honda beat.* Institut Teknologi Sepuluh Nopember.: PhD Thesis.
- Hastie, T.dkk. (2009). Overview of supervised learning. *In The elements of statistical learning*, 9-41.
- Hermawati, F. A. (2013). *Data Mining*. Yogyakarta: CV.Andi Offset.
- Huang, G.B,dkk. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), 489-501.
- HUTAPEA, D. (2022). ANALISA KARAKTERISTIK GETARAN UNTUK MESIN PEMERAS TEBU DENGAN PENGGERAK MOTOR BENSIN DAN ELEKTROMOTOR BERDASARKAN TIME DOMINE ARAH HORIZONTAL, VERTIKAL, DAN LONGITUDINAL TUGAS AKHIR.
- Inc., A. (2022). *Who is Anaconda?* Retrieved Juni 01, 2022, from anaconda.com: <https://www.anaconda.com/about-us>
- Kesumawati, A. (2018). Perbandingan Metode Support Vector Machine (SVM) Linear, Radial Basis Function (RBF), dan Polinomial Kernel dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII.
- Liandana, M., dkk. (2019). Wearable Device untuk Merekam Data Akselerasi Aktivitas Fisik Menggunakan Sensor Accelerometer. *J. Explor. STMIK Mataram*–, Volume, 9(1).
- Mahmud, M.dkk. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE transactions on neural networks and learning systems*, 29(6), 2063-2079.
- Mochtiarsa, Y. (2014). Rancangan kendali lampu menggunakan mikrokontroller ATmega328 berbasis sensor getar. *Jurnal Informatika SIMANTIK*, 1(1), 40-44.
- Nalim, N., & Salafudin, S. (n.d.). *Statistika Deskriptif*. 2012.
- Notebook, J. (2022). *Who Is Jupyter Notebook*. Retrieved Juni 01, 2022, from [jupyter.org: https://jupyter.org/about](https://jupyter.org/about)
- Nugroho, A. S. dkk. (2003). *Support Vector Machine*. roceeding Indones. Sci. Meeiting Cent Japan.

- Octaviani, P. A.dkk. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang. *Jurnal Gaussian*, 3(4), 811-820.
- Pindrayana, K, & dkk. (2018). Prototipe Pemandu Parkir Mobil Dengan Output Suara Manusia Menggunakan Mikrokontroler Arduino Uno. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(2).
- Prasetyo, E. (2012). *Data mining konsep dan aplikasi menggunakan matlab*. Yogyakarta: Yogyakarta: Andi, 1.
- Qiang, Q., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 1143-1146.
- Rao, & Singiresu S. (2011). *Mechanical Vibrations*. Miami : Pearson Education, Inc., Fifth Edition.
- Rao, S. (2011). *Mechanical Vibrations Fifth*.
- Roihan Ahmad,dkk. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. 75-82.
- Roihan, A., Sunarya, P.A., & Rafika, A.S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang. *IJCIT (Indonesian J. Comput. Inf. Technol., 5(1)*, 75-82.
- Rokhman, & Taufiqur. (2016). Analisis Getaran Pada Footrest Sepeda Motor Tipe Matic dan Non-Matic. *Jurnal Ilmiah Teknik Mesin 4 (2)*, 31-40.
- Sahitayakti, R. P., & Fithriasari, K. (2016). Klasifikasi Kesejahteraan Rumah Tangga di Provinsi Papua dengan Metode Regresi Logistik dan Support Vector Machine. *Jurnal Sains dan Seni ITS*, 4(2).
- Santosa, B. (2007). *Data mining teknik pemanfaatan data untuk keperluan bisnis*. Yogyakarta: Graha Ilmu, 978(979) 756.
- Scholkopf, B & Smola, A.J. (2002). *Learning With Kernels*. London: The MIT Press Cambridge Massachussets.
- Sembiring, K. (2007). *Penerapan teknik support vector machine untuk pendeteksian intrusi pada jaringan*. Institut Teknologi Bandung.
- Somvanshi, M., & Chavan, P. (2016). A review of machine learning techniques using decision tree and support vector machine. *2016 International*

Conference on Computing Communication Control and Automation (ICCUBEA), 1-7.

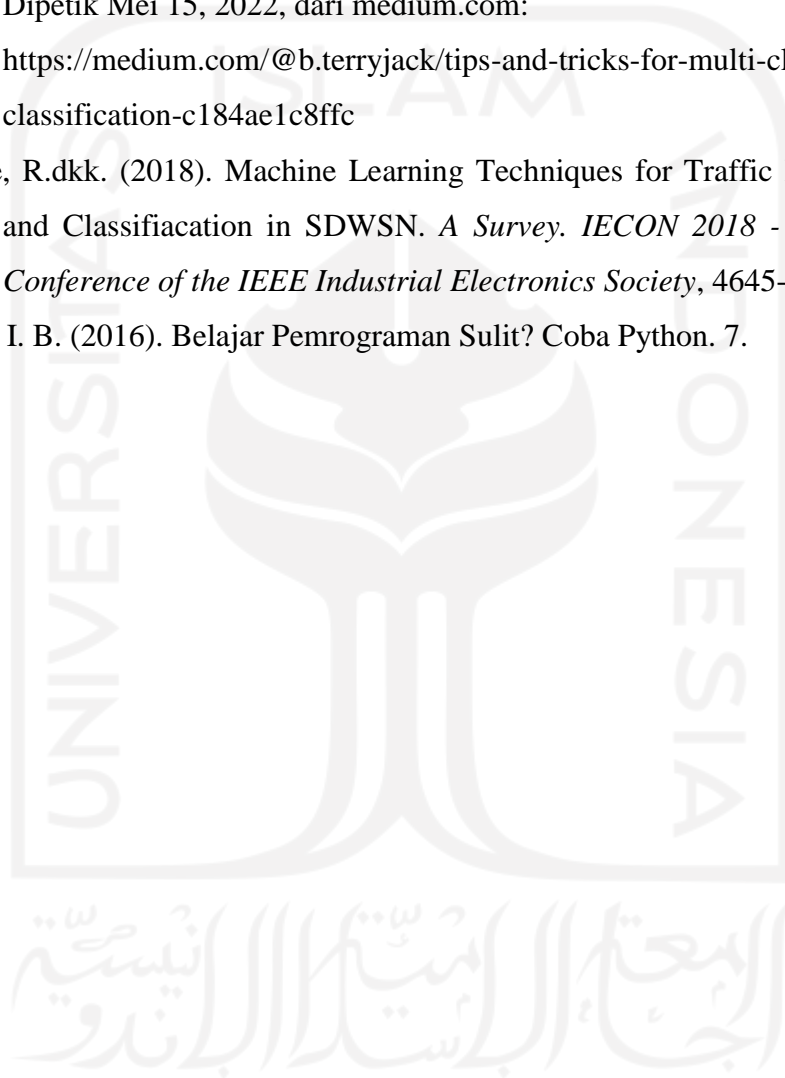
SUHANDOKO, S. (2014). *Analisis Getaran Pada Sistem Suspensi Kendaraan Roda Dua (Yamaha Jupiter Z 2004) Menggunakan Software Matlab 6.5*. Diss: Universitas Muhammadiyah Surakarta.

Terry-Jack, M. (2019, April 28). *Tips and Tricks for Multi-Class Classification*. Dipetik Mei 15, 2022, dari medium.com:

<https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc>

Thupae, R.dkk. (2018). Machine Learning Techniques for Traffic Identification and Classification in SDWSN. *A Survey. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 4645-4650.

Trisno, I. B. (2016). Belajar Pemrograman Sulit? Coba Python. 7.



LAMPIRAN

Lampiran 1 : List program arduino untuk perangkat keras

```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"

const int chipSelect = 10;

int jam, menit, detik;
String waktu, tanggal;
RTC_DS1307 rtc;
char
daysOfTheWeek[7][21]={"Sunday", "Monday", "Tuesday", "Wednesday",
", "Thursday", "Friday", "Saturday"};
File myFile;

int ADXL345 = 0x53; // The ADXL345 sensor I2C address
float X_out, Y_out, Z_out; // Outputs

int sw = 2;

void setup() {
  // put your setup code here, to run once:

  Wire.begin();
  pinMode(sw, INPUT);
  Serial.begin(9600); // Initiate serial communication for
printing the results on the Serial monitor
  while (!Serial) {; // wait for serial port to connect.
Needed for native USB port only
}

  Serial.print("Initializing SD card...");
```

```

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  // don't do anything more:
  return;
}
Serial.println("card initialized.");
if (SD.exists("datavib.csv")) {
  Serial.println("datavib.txt exists.");
} else {
  Serial.println("datavib.txt doesn't exist.");
}
Serial.println("Creating example.txt...");
myFile = SD.open("datavib.csv", FILE_WRITE);
myFile.close();

if (SD.exists("datavib.csv")) {
  Serial.println("datavib.txt exists.");
} else {
  Serial.println("datavib.txt doesn't exist.");
}

#ifdef ESP8266
while (!Serial); // wait for serial port to connect.
Needed for native USB
#endif

if (!rtc.begin()) {
  Serial.println("Couldn't find RTC");
}
Serial.flush();
abort();
}

```

اجتاز آزمون در این زمینه

```

    if (! rtc.isrunning()) {
        Serial.println("RTC is NOT running, let's set the time!
");
        // When time needs to be set on a new device, or after
a power loss, the
        // following line sets the RTC to the date & time this
sketch was compiled
        rtc.adjust(DateTime(F(_DATE), F(TIME_)));
        // This line sets the RTC with an explicit date & time,
for example to set
        // January 21, 2014 at 3am you would call:
        // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
    }

    // When time needs to be re-set on a previously
configured device, the
    // following line sets the RTC to the date & time this
sketch was compiled
    // rtc.adjust(DateTime(F(_DATE), F(TIME_)));
    // This line sets the RTC with an explicit date & time,
for example to set
    // January 21, 2014 at 3am you would call:
    //rtc.adjust(DateTime(2021, 10, 28, 10, 06, 0));

    // Set ADXL345 in measuring mode
    Wire.beginTransmission(ADXL345); // Start communicating
with the device
    Wire.write(0x2D); // Access/ talk to POWER_CTL Register -
0x2D

```

```

// Enable measurement
Wire.write(8); // (8dec -> 0000 1000 binary) Bit D3 High
for measuring enable
Wire.endTransmission();
delay(10);

//8 G resolution SET
Wire.beginTransaction(ADXL345); // Start communicating
with the device
Wire.write(0x31); // Access/ talk to Register
0x31—DATA_FORMAT
// Enable measurement
Wire.write(2); // (2dec -> 0000 0010 binary) Bit D1 High
for +/-8g enable
Wire.endTransmission();
delay(10);

// This code goes in the SETUP section
// Off-set Calibration
//X-axis
Wire.beginTransaction(ADXL345);
Wire.write(0x1E); // X-axis offset register
Wire.write(-6);
Wire.endTransmission();
delay(10);
//Y-axis
Wire.beginTransaction(ADXL345);
Wire.write(0x1F); // Y-axis offset register
Wire.write(-4);
Wire.endTransmission();
delay(10);

```

اجتازہ سہولت دہانہ

```

//Z-axis
Wire.beginTransaction(ADXL345);
Wire.write(0x20); // Z-axis offset register
Wire.write(3);
Wire.endTransmission();
delay(10);
}

void loop() {
  // put your main code here, to run repeatedly:
  int kond = digitalRead (sw);
  //Serial.println (kond);
  //delay (1000);
  if (kond == 1) {

    DateTime now = rtc.now();
    String dataString = "";
    String tahun = String(now.year(), DEC);
    String('/');
    String bulan = String(now.month(), DEC);
    String('/');
    String hari = String(now.day(), DEC);
    String("(");
    String(daysOfTheWeek[now.dayOfTheWeek()]);
    String(")");
    String jam = String(now.hour(), DEC);
    String(':');
    String menit = String(now.minute(), DEC);
    String(':');
    String detik = String(now.second(), DEC);
    String waktu = String("(" + tahun + "/" + bulan + "/" +
hari + ")" + "," + jam + ":" + menit + ":" + detik);

```

اجتازہ امتحان


```

//Serial.println(waktu);
//Serial.println(now.second(),DEC);
// === Read accelerometer data === //
Wire.beginTransmission(ADXL345);
Wire.write(0x32); // Start with register 0x32
(ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(ADXL345, 6, true); // Read 6 registers
total, each axis value is stored in 2 registers
X_out = ( Wire.read() | Wire.read() << 8); // X-axis value
X_out = X_out/64; //For a range of +-2g, we need to
divide the raw values by 256, according to the datasheet
Y_out = ( Wire.read() | Wire.read() << 8); // Y-axis value
Y_out = Y_out/64;
Z_out = ( Wire.read() | Wire.read() << 8); // Z-axis value
Z_out = Z_out/64;

// read three axis and append to the string:
// make a string for assembling the data to log:String
dataString = "";

// open the file. note that only one file can be open
at a time,
// so you have to close this one before opening another.
myFile = SD.open("datavib.csv", FILE_WRITE);

// if the file is available, write to it:
if (myFile) {
myFile.println(waktu + "," + X_out + "," + Y_out + ",

```

```

" + Z_out);
    myFile.close();
    // print to the serial port too:
    Serial.println(waktu + "-->" + ", Getaran: " + X_out
+ ", " + Y_out + ", " + Z_out);

}
// if the file isn't open, pop up an error:
else {
    Serial.println("error opening...");
}

delay(10);
}
else {
    Serial.println ("nyalakan saklar");
}
}

```

اجتازہ سہولت کارڈ

Lampiran 2 : Data Getaran Hasil Pengujian Objek A, B dan C

Count	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min	X_max	Y_max	Z_max	X_median	Y_median	Z_median	Speed	Respon
388	0.049639	0.160644	0.974665	0.112848	0.171425	0.240139	-0.34	-0.55	0.28	0.58	1.19	2	0.05	0.17	0.98	Low	Tinggi
317	0.053344	0.172997	0.981293	0.117765	0.171034	0.245584	-0.98	-0.22	0.16	0.41	1.28	1.8	0.05	0.17	1	Low	Tinggi
275	0.008873	0.181309	0.996764	0.144489	0.26565	0.385278	-0.44	-0.98	-0.3	0.5	1	3.72	0.02	0.17	1	Low	Tinggi
258	-0.007325581	0.191472868	0.974612403	0.147851733	0.216358618	0.349326262	-0.83	-0.86	-0.96	0.39	0.77	2.16	0.02	0.19	0.95	Low	Tinggi
296	0.003445946	0.198074324	0.967736486	0.105997844	0.195557459	0.246156788	-0.58	-0.56	0.28	0.28	1.02	1.61	0.02	0.19	0.97	Low	Tinggi
206	0.064029126	0.189805825	0.948252427	0.211537413	0.370083657	0.519509012	-0.75	-0.83	-1.16	0.92	1.66	2.36	0.05	0.17	0.975	Medium	Sedang
203	0.051625616	0.183596059	0.981871921	0.218439147	0.464044184	0.591500394	-0.75	-1.22	-1.05	1.03	2.2	3.58	0.03	0.16	0.97	Medium	Sedang
162	-0.018209877	0.162962963	1.005185185	0.268325621	0.395087676	0.461617763	-1.84	-2.22	-0.69	1.08	2.22	3.89	0.02	0.15	1	Medium	Sedang
182	0.019395604	0.183296703	0.916978022	0.23454474	0.354132837	0.666623544	-1.41	-1.33	-3.8	1.78	1.19	4.72	0.02	0.225	0.94	Medium	Sedang
167	-0.001017964	0.22760479	1.013173653	0.176056654	0.429896611	0.496429529	-0.67	-0.86	-1.2	0.63	3.27	4.22	-0.02	0.17	1.03	Medium	Sedang
122	0.028360656	0.149590164	0.977213115	0.347006944	0.586058973	1.04977783	-1.69	-1.58	-3.02	1.13	2.47	5.64	0.05	0.16	1.025	High	Rendah
122	0.063852459	0.188442623	1.003688525	0.273925992	0.425007575	0.607849754	-0.5	-0.98	-0.88	1.66	1.91	3.56	0.06	0.17	1.07	High	Rendah
138	-0.00260896	0.239057971	0.944637681	0.190578468	0.422619374	0.471308412	-0.69	-1.02	-0.66	0.69	1.41	2.2	0.01	0.225	0.94	High	Rendah
122	0.055327869	0.118852459	1.026229508	0.278238807	0.378693266	0.470837706	-0.48	-1.25	-0.69	2	1.39	2.89	0.05	0.1	1.025	High	Rendah
129	-0.015813953	0.232248062	0.945736434	0.375973108	0.446407956	0.593333769	-3.23	-1.14	-0.86	0.73	2	3.17	0.02	0.22	0.94	High	Rendah
159	0.046100629	0.184779874	0.960943396	0.068601878	0.140369478	0.264306584	-0.16	-0.17	0.25	0.23	0.98	1.64	0.05	0.17	0.95	Low	Tinggi
171	0.051988304	0.184736842	0.979005848	0.059087372	0.15175567	0.24767544	0	-0.31	0.39	0.27	0.72	1.64	0.05	0.17	0.98	Low	Tinggi
120	0.026833333	0.144666667	0.9975	0.085690309	0.229247704	0.345056931	-0.25	-0.98	-0.06	0.3	0.88	1.83	0.03	0.16	0.99	Low	Tinggi
105	0.032666667	0.165047619	0.977619048	0.101415621	0.240452793	0.387865272	-0.22	-0.89	-0.3	0.55	0.86	2.02	0.03	0.19	1.02	Low	Tinggi
114	0.024912281	0.156052632	0.98877193	0.079847705	0.200819641	0.36262841	-0.17	-0.2	-0.13	0.36	1.02	1.84	0.02	0.13	0.975	Low	Tinggi
126	0.064603175	0.146111111	0.991507937	0.132512042	0.209822676	0.463693118	-0.22	-0.66	-0.45	0.91	0.59	2.59	0.05	0.15	0.98	Medium	Sedang
126	0.059126984	0.147301587	1.002142857	0.093603588	0.218439603	0.401522317	-0.31	-0.84	0.08	0.33	0.98	1.95	0.05	0.135	1	Medium	Sedang
108	0.010740741	0.141226296	0.996296296	0.139509848	0.328625266	0.61359153	-0.38	-1.17	-1.34	0.66	1.2	4.2	0.02	0.16	1	Medium	Sedang
104	-0.000576923	0.150961538	0.884807692	0.126059216	0.323216598	0.565096402	-0.55	-1.14	-0.55	0.2	1.14	3.75	0.02	0.165	0.94	Medium	Sedang
75	0.000666667	0.207866667	0.9192	0.191786292	0.280339048	0.48390238	-0.59	-0.53	-0.78	0.73	1.48	2.78	0.02	0.19	0.89	Medium	Sedang
87	0.048735632	0.229655172	0.929310345	0.169690584	0.389700088	0.734514225	-0.77	-1.19	-2.25	0.5	1.48	2.84	0.05	0.23	1	High	Rendah
97	0.044329897	0.171649485	0.951340206	0.16895455	0.38073033	0.692776162	-0.5	-1.09	-2.16	0.64	1.16	2.78	0.05	0.16	1.03	High	Rendah
148	0.029256757	0.166081081	0.940067568	0.153313033	0.338155531	0.627339156	-0.5	-0.83	-2.97	0.59	1.55	3.08	0.03	0.16	0.975	High	Rendah
82	0.029634146	0.155	0.988292683	0.208281891	0.402084845	0.689160887	-0.84	-1.16	-1.02	0.66	1.2	2.67	0.05	0.2	1.01	High	Rendah
94	0.025531915	0.216276596	0.947978723	0.193427584	0.414980202	0.643153487	-0.47	-1.19	-2.17	0.89	1.19	2.41	0.02	0.25	0.95	High	Rendah
730	0.027410959	0.190191781	0.965575342	0.052299742	0.101768809	0.222112344	-0.36	-0.28	0.17	0.28	0.52	2.14	0.03	0.19	0.98	Low	Tinggi
703	0.014807966	0.169573257	0.969487909	0.040494611	0.076713772	0.154861466	-0.13	-0.11	0.44	0.14	0.44	1.41	0.02	0.17	0.97	Low	Tinggi
672	0.018065476	0.217589286	0.965535714	0.048652255	0.088566112	0.160781471	-0.14	-0.16	0.33	0.17	0.55	1.63	0.02	0.22	0.97	Low	Tinggi
692	0.005433526	0.21216763	0.97099711	0.0434005	0.081115135	0.12070231	-0.19	-0.06	0.45	0.13	0.44	1.33	0.02	0.2	0.98	Low	Tinggi
680	0.005147059	0.213235294	0.9685	0.042220846	0.091782242	0.146012639	-0.25	-0.02	0.23	0.19	0.56	1.55	0.02	0.2	0.98	Low	Tinggi
497	0.023018109	0.133179074	0.989637827	0.053548908	0.1016217	0.184499841	-0.14	-0.33	0.38	0.17	0.47	1.58	0.02	0.13	0.98	Medium	Sedang
519	0.027745665	0.131194605	0.974836224	0.05621701	0.094767133	0.198543738	-0.13	-0.25	0.14	0.22	0.61	1.53	0.03	0.14	0.97	Medium	Sedang
512	-0.002949219	0.217949219	0.956132813	0.057003565	0.105161759	0.201757168	-0.16	-0.16	0.33	0.2	0.59	1.48	0.03	0.2	0.97	Medium	Sedang
521	-0.007159309	0.214702495	0.957927063	0.064085216	0.106100269	0.189502188	-0.42	-0.17	0.16	0.19	0.59	1.56	0.03	0.22	0.97	Medium	Sedang
560	-0.007213439	0.237826087	0.962193676	0.06461113	0.116712465	0.176567184	-0.2	-0.28	0.31	0.22	0.56	1.56	0.03	0.22	0.97	Medium	Sedang
384	0.026953125	0.175182292	0.974973958	0.070617107	0.122982807	0.243374877	-0.23	-0.23	0.25	0.27	0.5	1.53	0.03	0.17	1	High	Rendah
397	0.026725441	0.171486146	0.972619647	0.065619485	0.115095735	0.232888513	-0.19	-0.27	0.34	0.22	0.52	1.5	0.03	0.19	1	High	Rendah
412	0.005137845	0.219197995	0.964235589	0.081357321	0.140411665	0.233469257	-0.23	-0.33	0.13	0.31	0.63	1.52	0.03	0.22	0.98	High	Rendah
612	-0.013446602	0.221820388	0.960946602	0.067136528	0.125379567	0.206374868	-0.41	-0.41	0.13	0.19	0.55	1.53	-0.02	0.225	0.98	High	Rendah
610	-0.011213115	0.272786885	0.951737705	0.079984628	0.091233675	0.104293369	-0.38	-0.09	0.55	1.45	0.59	1.28	-0.02	0.275	0.95	High	Rendah



Lampiran 3 : Script Python – Jupyter Notebook SVM Klasifikasi Polynomial Kernel

Machine Learning Model SVM

1. Import Data

Pada tahap ini, agar dapat memahami permasalahan sesuai berdasarkan kondisi dataset, maka data harus dimuat terlebih

```
In [94]: # Import standard Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import ListedColormap
```

```
In [95]: datavibration= pd.read_csv('datavibration1.csv')
datavibration.head()
```

```
Out[95]:
```

	Run	Count	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min
0	A1	388	0.049639	0.160644	0.974665	0.112848	0.171425	0.240139	-0.34	-0.55	0.21
1	A2	317	0.053344	0.172997	0.981293	0.117765	0.171034	0.245584	-0.98	-0.22	0.11
2	A3	275	0.008873	0.181309	0.996764	0.144489	0.265650	0.385278	-0.44	-0.98	-0.31
3	A4	258	-0.007326	0.191473	0.974612	0.147852	0.216359	0.349326	-0.83	-0.86	-0.91
4	A5	296	0.003446	0.198074	0.967736	0.105998	0.195557	0.246157	-0.58	-0.56	0.21

Berdasarkan kondisi dataset diatas. Jenis machine learning pada project ini merupakan Supervised Learning, karena data memiliki label/kelas. Label dalam data_vibration ini adalah data yang dimuat pada kolom 'Respon' atau sebagai target. Selain itu, data yang dimiliki merupakan structured data karena data tersebut berwujud tabel.

Atau dalam istilah machine learning juga sering disebut :

Independent Variable yaitu Kolom 'Respon' dan kolom lainnya disebut sebagai Dependent Variable

Sedangkan permasalahan pada data ini merupakan classification, dimana akan dibuat machine learning model untuk memprediksi Respon yang terjadi menurut data yang diberikan.



In [96]: `#Cek apakah ada data yang kosong`

```
datavibration.isna().sum()
```

```
Out[96]: Run          0
Count         0
X_mean        0
Y_mean        0
Z_mean        0
X_std         0
Y_std         0
Z_std         0
X_min         0
Y_min         0
Z_min         0
X_max         0
Y_max         0
Z_max         0
X_median      0
Y_median      0
Z_median      0
Speed         0
Respon        0
dtype: int64
```

Dengan bantuan fungsi `isna()` dan `sum()` dapat diketahui bahwa dalam dataset semua kolom tidak ada nilai yang kosong. Dari tampilan diatas, dapat melihat bahwa data tidak ada yang missing

Tampilan dataframe diatas adalah data yang akan di berikan ke machine learning sebagai data train dan data test

In []:

2. Menghapus Kolom

Pada kasus data ini, ada beberapa kolom yang harus dibuang atau didelete. karena tidak berpengaruh terhadap pembuatan machine learning model yang akan dibuat. Pada kasus data ini kolom 'Count', 'Run', 'Speed' akan di buang/delete.

Untuk membuang/delete kolom dapat di lakukan dengan cara :



```
In [97]: datavibration.drop(['Count', 'Run', 'Speed'], axis = 1, inplace = True)
datavibration.head()
```

```
Out[97]:
```

	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min	X_max	Y_max
0	0.049639	0.160644	0.974665	0.112848	0.171425	0.240139	-0.34	-0.55	0.28	0.58	0.58
1	0.053344	0.172997	0.981293	0.117765	0.171034	0.245584	-0.98	-0.22	0.16	0.41	0.41
2	0.008873	0.181309	0.996764	0.144489	0.265650	0.385278	-0.44	-0.98	-0.30	0.50	0.50
3	-0.007326	0.191473	0.974612	0.147852	0.216359	0.349326	-0.83	-0.86	-0.96	0.39	0.39
4	0.003446	0.198074	0.967736	0.105998	0.195557	0.246157	-0.58	-0.56	0.28	0.28	0.28

3. Mengubah Data Object Menjadi Numerik

Setelah berhasil memuat data ke dalam program, proses selanjutnya ialah memproses data dan mengubahnya menjadi numerik atau angka. Namun, sebelumnya harus mengetahui tipe data dalam dataframe dulu.

```
In [98]: #Menampilkan informasi dari dataframe data
datavibration.info()
```

```
<<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  -
0   X_mean      45 non-null    float64
1   Y_mean      45 non-null    float64
2   Z_mean      45 non-null    float64
3   X_std       45 non-null    float64
4   Y_std       45 non-null    float64
5   Z_std       45 non-null    float64
6   X_min       45 non-null    float64
7   Y_min       45 non-null    float64
8   Z_min       45 non-null    float64
9   X_max       45 non-null    float64
10  Y_max       45 non-null    float64
11  Z_max       45 non-null    float64
12  X_median    45 non-null    float64
13  Y_median    45 non-null    float64
14  Z_median    45 non-null    float64
15  Respon     45 non-null    object
dtypes: float64(15), object(1)
memory usage: 5.8+ KB
```

Pada kolom Dtype terdapat type data yang berbeda beda, yaitu float64 dan object. Data dengan tipe object ini merupakan categorical data, sehingga setelah ini harus dilakukan preprocessing

untuk mengubah data menjadi numerik atau angka. Karena untuk pengaplikasian SVM hanya dapat membaca data yang berupa angka.

Pada datavibration1 terdapat satu type data object yaitu pada kolom Respon, sehingga akan ubah menjadi type data numerik

Caranya, dapat menggunakan encoder dari module Sklearn. Encoder digunakan untuk mengubah data kategori menjadi angka.

```

In [99]: # Import LabelEncoder dari module sklearn
from sklearn.preprocessing import LabelEncoder

# Menyalin / copy dataframe agar dataframe awal tetap utuh
datavibration_int = datavibration.copy()

# Membuat objek/instance yang bernama encoder
encoder = LabelEncoder()

# Membuat list dari nama kolom data kategori
categorical_data = ['Respon']

# Mengubah setiap data kategori menjadi numerik dengan encoder
for kolom in categorical_data:
    datavibration_int[kolom] = encoder.fit_transform(datavibration[kolom])

# Sekarang data sudah berupa angka sepenuhnya, dapat dilihat pada tampilan dibawah
datavibration_int

```

Out[99]:

	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min	X_max
0	0.049639	0.160644	0.974665	0.112848	0.171425	0.240139	-0.34	-0.55	0.28	0.58
1	0.053344	0.172997	0.981293	0.117765	0.171034	0.245584	-0.98	-0.22	0.16	0.41
2	0.008873	0.181309	0.996764	0.144489	0.265650	0.385278	-0.44	-0.98	-0.30	0.50
3	-0.007326	0.191473	0.974612	0.147852	0.216359	0.349326	-0.83	-0.86	-0.96	0.39
4	0.003446	0.198074	0.967736	0.105998	0.195557	0.246157	-0.58	-0.56	0.28	0.28
5	0.064029	0.189806	0.948252	0.211537	0.370084	0.519509	-0.75	-0.83	-1.16	0.92
6	0.051626	0.183596	0.981872	0.218439	0.464044	0.591500	-0.75	-1.22	-1.05	1.03
7	-0.018210	0.162963	1.005185	0.268326	0.395088	0.461618	-1.84	-2.22	-0.69	1.08
8	0.019396	0.183297	0.916978	0.234545	0.354133	0.666624	-1.41	-1.33	-3.80	1.78
9	-0.001018	0.227605	1.013174	0.176057	0.429897	0.496430	-0.67	-0.86	-1.20	0.63
10	0.028361	0.149590	0.977213	0.347007	0.586059	1.049778	-1.69	-1.58	-3.02	1.13
11	0.063852	0.188443	1.003689	0.273926	0.425008	0.607850	-0.50	-0.98	-0.88	1.66
12	-0.002609	0.239058	0.944638	0.190578	0.422619	0.471308	-0.69	-1.02	-0.66	0.69
13	0.055328	0.118852	1.026230	0.278239	0.378693	0.470838	-0.48	-1.25	-0.69	2.00
14	-0.015814	0.232248	0.945736	0.375973	0.446408	0.593334	-3.23	-1.14	-0.86	0.73
15	0.046101	0.184780	0.960943	0.068602	0.140369	0.264307	-0.16	-0.17	0.25	0.23
16	0.051988	0.184737	0.979006	0.059087	0.151756	0.247675	0.00	-0.31	0.39	0.27
17	0.026833	0.144667	0.997500	0.085690	0.229248	0.345057	-0.25	-0.98	-0.06	0.30
18	0.032667	0.165048	0.977619	0.101416	0.240453	0.387865	-0.22	-0.89	-0.30	0.55
19	0.024912	0.156053	0.988772	0.079848	0.200820	0.362628	-0.17	-0.20	-0.13	0.36
20	0.064603	0.146111	0.991508	0.132512	0.209823	0.463693	-0.22	-0.66	-0.45	0.91
21	0.059127	0.147302	1.002143	0.093604	0.218440	0.401522	-0.31	-0.84	0.08	0.33
22	0.010741	0.141296	0.996296	0.139510	0.328625	0.613592	-0.38	-1.17	-1.34	0.66

	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min	X_max
23	-0.000577	0.150962	0.884808	0.126059	0.323217	0.565096	-0.55	-1.14	-0.55	0.20
24	0.000667	0.207867	0.919200	0.191786	0.280339	0.483902	-0.59	-0.53	-0.78	0.73
25	0.048736	0.229655	0.929310	0.169691	0.389700	0.734514	-0.77	-1.19	-2.25	0.50
26	0.044330	0.171649	0.951340	0.168955	0.380730	0.692776	-0.50	-1.09	-2.16	0.64
27	0.029257	0.166081	0.940068	0.153313	0.338156	0.627339	-0.50	-0.83	-2.97	0.59
28	0.029634	0.155000	0.988293	0.208282	0.402085	0.689161	-0.84	-1.16	-1.02	0.66
29	0.025532	0.216277	0.947979	0.193428	0.414980	0.643153	-0.47	-1.19	-2.17	0.89
30	0.027411	0.190192	0.965575	0.052300	0.101769	0.222112	-0.36	-0.28	0.17	0.28
31	0.014808	0.169573	0.969488	0.040495	0.076714	0.154861	-0.13	-0.11	0.44	0.14
32	0.018065	0.217589	0.965536	0.048652	0.088566	0.160781	-0.14	-0.16	0.33	0.17
33	0.005434	0.212168	0.970997	0.043401	0.081115	0.120702	-0.19	-0.06	0.45	0.13
34	0.005147	0.213235	0.968500	0.042221	0.091782	0.146013	-0.25	-0.02	0.23	0.19
35	0.023018	0.133179	0.989638	0.053549	0.101622	0.184500	-0.14	-0.33	0.38	0.17
36	0.027746	0.131195	0.974836	0.056217	0.094767	0.198544	-0.13	-0.25	0.14	0.22
37	-0.002949	0.217949	0.956133	0.057004	0.105162	0.201757	-0.16	-0.16	0.33	0.20
38	-0.007159	0.214702	0.957927	0.064085	0.106100	0.189502	-0.42	-0.17	0.16	0.19
39	-0.007213	0.237826	0.962194	0.064611	0.116712	0.176567	-0.20	-0.28	0.31	0.22
40	0.026953	0.175182	0.974974	0.070617	0.122983	0.243375	-0.23	-0.23	0.25	0.27
41	0.026725	0.171486	0.972620	0.065619	0.115096	0.232889	-0.19	-0.27	0.34	0.22
42	0.005138	0.219198	0.964236	0.081357	0.140412	0.233469	-0.23	-0.33	0.13	0.31
43	-0.013447	0.221820	0.960947	0.067137	0.125380	0.206375	-0.41	-0.41	0.13	0.19
44	-0.011213	0.272787	0.951738	0.079985	0.091234	0.104293	-0.38	-0.09	0.55	1.45




```
In [100]: datavibration_int.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   X_mean      45 non-null    float64
1   Y_mean      45 non-null    float64
2   Z_mean      45 non-null    float64
3   X_std       45 non-null    float64
4   Y_std       45 non-null    float64
5   Z_std       45 non-null    float64
6   X_min       45 non-null    float64
7   Y_min       45 non-null    float64
8   Z_min       45 non-null    float64
9   X_max       45 non-null    float64
10  Y_max       45 non-null    float64
11  Z_max       45 non-null    float64
12  X_median    45 non-null    float64
13  Y_median    45 non-null    float64
14  Z_median    45 non-null    float64
15  Respon      45 non-null    int32
dtypes: float64(15), int32(1)
memory usage: 5.6 KB
```

```
In [101]: #Menampilkan kolom yang telah dirubah dari type data object menjadi numerik
for kolom in categorical_data:
    print(kolom,datavibration_int[kolom].unique())
```

```
Respon [2 1 0]
```

Dapat dilihat dataframe dari datavibration semua sudah berubah menjadi data Numerik, karena untuk klasifikasi SVM type data harus numerik. Pada Kolom Target atau 'Respon' yang sebelumnya bertipe kategori, sekarang sudah berubah jadi numerik. Terdapat 3 kelas tingkat kenyamanan yaitu 0 1 dan 2. Dimana : 0 = Rendah, 1 = Sedang, 2 = Tinggi

Ini adalah apa yang terjadi saat menggunakan LabelEncoder().

```
In [ ]:
```

4. Memisahkan Data

Pada tahap ini, dataframe akan dibagi menjadi dua, yaitu x dan y.

x merupakan kolom feature atau data (selain kolom 'Respon').

y merupakan kolom "Respon" sebagai data target klasifikasi

```
In [102]: # Memisahkan dataframe awal menjadi x (feature) dan y (target)
x= datavibration_int.drop('Respon',axis=1)
y = datavibration_int['Respon']
```

```
In [103]: #Menampilkan Lima data teratas data variabel x atau sebagai 'feature'
```

```
x.head()
```

```
Out[103]:
```

	X_mean	Y_mean	Z_mean	X_std	Y_std	Z_std	X_min	Y_min	Z_min	X_max	Y_max
0	0.049639	0.160644	0.974665	0.112848	0.171425	0.240139	-0.34	-0.55	0.28	0.58	0.97
1	0.053344	0.172997	0.981293	0.117765	0.171034	0.245584	-0.98	-0.22	0.16	0.41	0.97
2	0.008873	0.181309	0.996764	0.144489	0.265650	0.385278	-0.44	-0.98	-0.30	0.50	0.97
3	-0.007326	0.191473	0.974612	0.147852	0.216359	0.349326	-0.83	-0.86	-0.96	0.39	0.97
4	0.003446	0.198074	0.967736	0.105998	0.195557	0.246157	-0.58	-0.56	0.28	0.28	0.97

```
In [104]: #Menampilkan Lima data teratas data variabel y atau sebagai 'target'
```

```
y.head()
```

```
Out[104]:
```

```
0    2
1    2
2    2
3    2
4    2
Name: Respon, dtype: int32
```

```
In [ ]:
```

Setelah data dibagi menjadi dua bagian yaitu data (x), label(y), maka data di pisahkan lagi menjadi dua, data latihan dan tes.

```
In [105]: # Kita dapat memisahkan data menjadi data Latihan dan data tes dengan train_test_split
from sklearn.model_selection import train_test_split
```

```
In [106]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.20,stratify=y)
```

```
In [107]: print(x_train.shape,y_train.shape)
```

```
(36, 15) (36,)
```

```
In [108]: print(x_test.shape,y_test.shape)
```

```
(9, 15) (9,)
```

Sekarang data telah terpisah menjadi empat. Dimana data terdapat 36 data untuk latihan dan 9 data untuk test. Dimana jumlah 9 ini berasal dari argumen `test_size=0.20` atau 20% dari total 45 data

Data latihan digunakan sebagai bahan latihan model, sementara data tes digunakan untuk menguji performa model sebelum digunakan untuk bekerja. Sehingga sekarang memiliki empat data yaitu data latihan (`x_train`), data test (`x_test`), label/kelas latihan (`y_train`) dan label/kelas test

(y_test).

In []:

5. Membuat Machine Learning Model

Untuk menyelesaikan masalah klasifikasi atau classification maka kita dapat memilih beberapa pilihan. Dokumentasi dari sklearn https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html (https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html)

```
In [109]: # Import Linear SVC model dari sklearn
from sklearn.svm import SVC

# Menggunakan GridSearchCV untuk menemukan model dengan parameter terbaik
from sklearn.model_selection import GridSearchCV
```

```
In [110]: # Membuat objek/instance dengan nama model

model = SVC(gamma='scale')

SVC(gamma='scale', kernel='poly', degree=3, random_state=42).get_params()

# SVC Model Hyperparameter
param_grid = {'C': [1, 5, 6, 7, 8, 9, 10, 15, 20, 25, 26, 27, 28, 29, 30],
              'gamma': [1, 5, 6, 7, 8, 9, 10, 15, 20, 25, 26, 27, 28, 29, 30]}

# Membuat model terbaik dari semua kemungkinan kombinasi param_grid
best_model = GridSearchCV(SVC(), param_grid, cv=5, refit=True)
```

In []:

6. Melatih Machine Learning Model

```
In [111]: # Melatih model terbaik
best_model.fit(x_train, y_train)
```

```
Out[111]: GridSearchCV(cv=5, estimator=SVC(),
                      param_grid={'C': [1, 5, 6, 7, 8, 9, 10, 15, 20, 25, 26, 27, 28, 29, 30],
                                   'gamma': [1, 5, 6, 7, 8, 9, 10, 15, 20, 25, 26, 27, 28, 29, 30]})
```

```
In [112]: # Model dengan parameter terbaik
best_model.best_estimator_
```

```
Out[112]: SVC(C=15, gamma=1)
```

```
In [113]: # Membuat prediksi terhadap data tes
prediction = best_model.predict(x_test)
prediction
```

```
Out[113]: array([2, 0, 2, 0, 2, 0, 1, 2, 0])
```

```
In [ ]:
```

7. Evaluasi Model /Analys Model

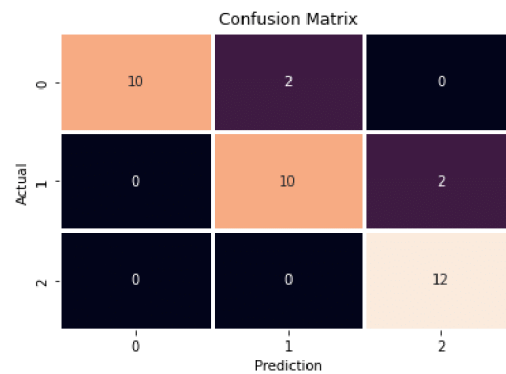
```
In [114]: # Import confusion matrix dari sklearn
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

7.1 Report Data Train

```
In [115]: def display_conf(y_train,prediction):
sns.heatmap(confusion_matrix(y_train,prediction),annot=True,linewidths=3,cbar=
plt.title('Confusion Matrix')
plt.ylabel('Actual')
plt.xlabel('Prediction')
plt.show()
```

```
In [116]: # Membuat klasifikasi dengan model yang telah ditingkatkan
prediction = best_model.predict(x_train)

# Menampilkan confusion matrix pada model yang telah ditingkankan
display_conf(y_train,prediction)
```



```
In [117]: print('Classification Report :')
print(classification_report(y_train,prediction))
```

```
Classification Report :
              precision    recall  f1-score   support

     0           1.00      0.83     0.91         12
     1           0.83      0.83     0.83         12
     2           0.86      1.00     0.92         12

 accuracy                   0.89         36
 macro avg                 0.90      0.89     0.89         36
 weighted avg             0.90      0.89     0.89         36
```

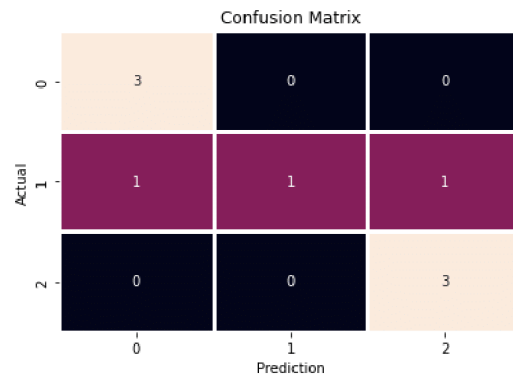
In []:

7.2 Resport Data Test

```
In [118]: def display_conf(y_test,prediction):
sns.heatmap(confusion_matrix(y_test,prediction),annot=True,linewidths=3,cbar=
plt.title('Confusion Matrix')
plt.ylabel('Actual')
plt.xlabel('Prediction')
plt.show()
```

```
In [119]: # Membuat klasifikasi dengan model yang telah ditingkatkan
prediction = best_model.predict(x_test)

# Menampilkan confusion matrix pada model yang telah ditingkankan
display_conf(y_test,prediction)
```



```
In [120]: print('Classification Report :')
print(classification_report(y_test,prediction))
```

```
Classification Report :
              precision    recall  f1-score   support

     0           0.75         1.00         0.86         3
     1           1.00         0.33         0.50         3
     2           0.75         1.00         0.86         3

 accuracy          0.78
 macro avg          0.83
 weighted avg       0.83
```

In []:

Awil Hamzah Rambe | 16525098 | Teknik Mesin | Fakultas Teknologi Industri | Universitas Islam Indonesia

In []:

