

**RANCANG BANGUN ALAT PENDETEKSI KECELAKAAN MOBIL
MENGUNAKAN SENSOR AKSELEROMETER DAN SENSOR 801S
VIBRATION**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
ADNAN MAHFUZHON
NIM: 135150301111122



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

RANCANG BANGUN ALAT PENDETEKSI KECELAKAAN MOBIL MENGGUNAKAN SENSOR AKSELEROMETER DAN SENSOR 801S VIBRATION

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Adnan Mahfuzhon
NIM: 135150301111122

Skripsi ini telah diuji dan dinyatakan lulus pada
01 AGUSTUS 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Tibyani, S.T, M.T
NIP: 19691101 199512 1 002

Gembong Edhi Setyawan, S.T, M.T
NIK: 201208 761201 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Agustus 2018



Adnan Mahfuzhon

NIM: 135150301111122



KATA PENGANTAR

Puja dan puji syukur penulis panjatkan kepada Allah SWT karna berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian dan Laporan Skripsi untuk memperoleh gelar Sarjana Komputer yang berjudul **Rancang Bangun Alat Pendeteksi Kecelakaan Menggunakan Sensor Akselerometer Dan Sensor 801s Vibration**.

Dalam pelaksanaan dan penyusunan laporan skripsi ini, tidak sedikit hambatan yang penulis hadapi. Namun penulis menyadari bahwa kelancaran dalam penyusunan laporan ini tidak lain berkat bantuan, dorongan dan bimbingan dari berbagai pihak, sehingga kendala-kendala yang penulis hadapi dapat teratasi. Penghargaan dan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Bapak Nuswanto, Ibu Dami Asih selaku orang tua penulis beserta adik kandung penulis yaitu Wafa Nazhofa, Alifah Az Zahra, dan Iffat Muhammad Nabil. Yang tiada henti memberikan semangat dan dorongan baik berupa doa maupun materi selama penulis melakukan penelitian.
2. Bapak Tibyani , S.T, M.T dan Bapak Gembong Edhi Setyawan, S.T, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dengan mengarahkan penulis sehingga dapat menyelesaikan penelitian.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika
4. Bapak Sabriansyah Rizqika Akbar, S.T., M.Eng. selaku Ketua Program Studi Teknik Komputer.
5. Teman-Teman kontrakan : Noor, Cholilulloh, Delta, Kresana, Bagus, Indra, Fajar, Willy, Zamroni, Alfian, Rizki, Adi, Anang, Tegar, Yogi, Billy. Yang selalu memberikan semangat.
6. Teman-teman teknik komputer angkatan 2013 yang telah banyak memberi bantuan dan dorongannya selama penulis menempuh studi di fakultas teknik komputer universitas Brawijaya dan selama penyelesaian penelitian ini.

Akhir kata penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan yang perlu disempurnakan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga skripsi ini bermanfaat bagi semua pihak yang membacanya.

Malang, 01 Agustus 2018

Penulis

Adnanmahfuzho@gmail.com

ABSTRAK

Peningkatan infrastruktur, terutama jalan dan jalan tol, akan membuat industri jasa transportasi semakin menjanjikan. Salah satunya adalah bisnis jasa sewa mobil yaitu meningkat hingga 70%. Ada sejumlah alasan, kenapa jasa sewa mobil lebih menjadi pilihan dari pada membeli atau memiliki mobil sendiri, terutama bagi kalangan dunia usaha. Selain menghemat anggaran, menyewa kendaraan bermotor juga menghilangkan sejumlah kerepotan, seperti soal pemeliharaan, perpanjangan surat kendaraan bermotor, bahkan menghilangkan risiko kehilangan kendaraan bermotor. Berdasarkan data yang di dapat dari meningkatnya usaha jasa sewa mobil perlunya upaya menjaga kenyamanan antara pemberi jasa dengan pelanggan, hal yang perlu diperhatikan ketika pemberi jasa harus menjaga asetnya berupa mobil mereka yang sedang disewa. Dan pelanggan harus menjaga mobil yang sedang mereka sewa. Dari permasalahan tersebut diperlukan adanya penelitian yang terkait dengan notifikasi kecelakaan untuk menjaga aset perusahaan dan kenyamanan pelanggan. Proses pengamilan data uji dengan cara manual untuk di jadikan data masukan untuk perhitungan naïve bayes. Penyesuaian data di tetapkan dengan mengacu pada data akselerometer jika data lebih dari 4g maka masuk dalam kondisi kecelakaan. Pengambilan data dari kondisi berjalan di dapat persentase kesesuaian sebesar 90%, lalu pada saat pengambilan dalam kondisi berhenti mendadak sebesar 78%, dan saat kondisi kecelakaan sebesar 98%. Dari hasil data uji dari hasil klasifikasi menggunakan metode naïve bayes didapat dengan pengambilan data uji pada masing masing 16 kali pengujian pada setiap skenario, pada kondisi berjalan didapat nilai keakurasian sebesar 98,7 %, kondisi berhenti mendadak didapat nilai keakurasian sebesar 87,5 %, dan kondisi kecelakaan didapat nilai keakurasian sebesar 98,7 %.

Kata Kunci : Pendeteksi kecelakaan, *Naïve Bayes*, MMA7361, 801s vibration

ABSTRACT

The improvement of infrastructure, especially road and freeway, will make the transportation service industry more promising. One of them is car rental service business, increases up to 70%. There are several reasons why car rental service becoming the first choice than buying or owing a car, specifically for business world. Besides saving the budget, renting a motorbike also erasing several difficulties, for instances, maintenance, extension for important letters, even decreasing losing the motorbike. Based on the data obtained from the increasing of car rental service business, it is needed to keep the comfort between the service provider with the customer. The important thing to be noticed, when the service provider must maintain the rental cars. Also, the customer must take care of the rented car. From the problem, it is need to conduct a research related to accidental notification for keeping the cars and customer's comfort. The process of collecting the test data manually to be made the input for calculating naïve bayes. The data adjustment is set by referring to accelerometer data, if the data is more than 4 grams so it is categorized in accident condition. Collecting the data in the running condition obtained the value of accuracy up to 90%, then when collecting in the sudden stop condition obtained up to 78%, and in accident condition obtained up to 98%. The data test results from the classification results using naïve bayes, they are obtained by collecting the test data in each one for 16-time tests in every scenarios. In running condition, it is obtained the value of accuracy up to 98,7%, in the sudden stop condition obtained the value of accuracy up to 87,5% and in the accident condition obtained the value of accuracy up to 98,7%.

Keyword : *Accident Detection, Naïve Bayes, MMA7361, 801s vibration*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka.....	4
2.2 Dasar Teori.....	6
2.2.1 Naive Bayes.....	6
2.2.2 Mikrokontroler.....	8
2.2.3 Akselerometer.....	9
2.2.4 Sensor Getaran.....	10
2.2.5 SMS (<i>Short Message Service</i>).....	11
2.2.6 Arus Listrik DC	11
2.2.7 Bluetooth	12
BAB 3 METODOLOGI	13
3.1 Studi Literatur	14
3.2 Rekayasa Kebutuhan.....	14
3.2.1 Kebutuhan Perangkat Keras	15
3.2.2 Kebutuhan Perangkat Lunak.....	15
3.3 Perancangan dan Implementasi.	15

3.3.1 Perancangan Perangkat Keras	15
3.3.2 Perancangan Perangkat Lunak.....	16
3.4 Pengujian dan Analisis.....	17
3.5 Kesimpulan	17
BAB 4 REKAYASA KEBUTUHAN.....	18
4.1 Kebutuhan Sistem	18
4.1.1 Kebutuhan Perangkat Keras	18
4.1.2 Kebutuhan Perangkat Lunak.....	24
4.2 Kebutuhan Fungsional.....	25
4.3 Kebutuhan Non Fungsional	26
4.3.1 Asumsi kebutuhan sistem.....	26
4.3.2 Batasan Disain Sistem.....	26
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	27
5.1 Perancangan Sistem.....	27
5.1.1 Perancangan Perangkat Keras	27
5.1.2 Perancangan Prototipe Alat Pendeteksi Tabrakan	31
5.1.3 Perancangan Perangkat Lunak.....	32
5.2 Implementasi Sistem.....	41
5.2.2 Implementasi perangkat keras	41
5.2.1 Implementasi prototipe alat pendeteksi kecelakaan	42
5.2.3 Implementasi perangkat lunak	42
BAB 6 PENGUJIAN DAN ANALISIS.....	52
6.1 Pengujian Sistem Deteksi Tabrakan Pada Setiap Kondisi	52
6.1.1 Tujuan Pengujian Pada Setiap kondisi.....	52
6.1.2 Analisis Dan Pengujian Mobil Remote Control Berjalan Kedepan.....	52
6.1.3 Analisis dan Pengujian Mobil Remote Control Berhenti Mendadak 54	
6.1.4 Analisis dan Pengujian Mobil Remote Control Menabrak	56
6.2 Pengujian Sistem Deteksi Tabrakan Dengan Klasifikasi Naïve Bayes ...	58
6.2.1 Tujuan Pengujian.....	58
6.2.2 Pengujian Mobil Remote Control Berjalan Kedepan Menggunakan Naïve Bayes	58



6.2.3 Pengujian Mobil Remote Control Berhenti Mendadak Menggunakan Naïve Bayes	59
6.2.4 Pengujian Mobil Remote Control Kecelakaan Menggunakan Naïve Bayes	60
6.3 Pengujian Pengiriman Notifikasi Menggunakan <i>SIM 800I</i>	61
BAB 7 PENUTUP	62
7.1 Kesimpulan	62
7.2 Saran.....	62
DAFTAR PUSTAKA	63
LAMPIRAN A DATA LATIH MOBIL BERJALAN	64
LAMPIRAN B DATA LATIH MOBIL BERJALAN	65
LAMPIRAN D GRAFIK AMPLITUDO FITUR MOBIL BERJALAN	66
LAMPIRAN E DATA LATIH MOBIL BERHENTI MENDADAK	67
LAMPIRAN F DATA LATIH MOBIL BERHENTI MENDADAK	68
LAMPIRAN G GRAFIK AMPLITUDO FITUR MOBIL BERHENTI MENDADAK	69
LAMPIRAN H DATA LATIH MOBIL KECELAKAAN	70
LAMPIRAN I DATA LATIH MOBIL KECELAKAAN	71
LAMPIRAN J GRAFIK AMPLITUDO FITUR MOBIL KECELAKAAN	72



DAFTAR TABEL

Tabel 5. 1 Keterangan koneksi pin <i>aklselerometer MMA7361</i> dengan Arduino nano..	28
Tabel 5. 2 Keterangan koneksi pin <i>801s vibration</i> dengan Arduino nano	29
Tabel 5. 3 Keterangan koneksi pin <i>SIM 800I</i> dengan Arduino nano	30
Tabel 5. 4 Keterangan koneksi pin <i>Bluetooth</i> dengan Arduino nano	31
Tabel 5. 5 <i>Mean</i> tiap kondisi	37
Tabel 5. 6 standar deviasi tiap kondisi.....	38
Tabel 5. 7 nilai <i>gaussian</i> tiap kondisi.....	39
Tabel 5. 8 Kode program inialisasi <i>Liblary</i> sistem pendeteksi tabrakan	42
Tabel 5. 9 Kode program Inialisasi variable pembacaan sensor.....	43
Tabel 5. 10 Kode program pengambilan data.....	44
Tabel 5. 11 Kode program inialisasi variable pembacaan sensor	45
Tabel 5. 12 Kode program Inialisasi sensor akselesometer MMA7361	45
Tabel 5. 13 Kode program inialisasi sensor getar <i>801s</i>	46
Tabel 5. 14 Kode program untuk mengecek kondisi	46
Tabel 5. 15 Kode program inialisasi variable <i>Naïve Bayes</i>	47
Tabel 5. 16 Kode program fungsi nBayes()	47
Tabel 5. 17 Kode program Inialisasi variable pembacaan sensor	48
Tabel 5. 18 Kode program perhitungan fungsi probPrior()	49
Tabel 5. 19 Kode program perhitungan fungsi mean dan standar deviasi.....	49
Tabel 5. 20 Kode program perhitungan fungsi <i>gaussian</i>	50
Tabel 5. 21 Kode program perhitungan fungsi probPosterior	50
Tabel 5. 22 Kode program perhitungan fungsi kesimpulan.....	51
Tabel 6. 1 Kesesuaian Pengambilan Data Percepatan Mobil Berjalan	54
Tabel 6. 2 Kesesuaian Pengambilan Data Percepatan Mobil Berhenti Mendadak.....	55
Tabel 6. 3 Kesesuaian Pengambilan Data Percepatan Mobil Kecelakaan	57
Tabel 6. 4 Pengujian Saat Mobil Berjalan	59
Tabel 6. 5 Pengujian Saat Mobil Berhenti mendadak	60
Tabel 6. 6 Pengujian Saat Mobil Kecelakaan.....	61
Tabel 0. 1 Data Latih Mobil Berjalan Data 1- Data 5	64
Tabel 0. 2 Data Latih Mobil Berjalan Data 6 - Data 10.....	65
Tabel 0. 3 Data Latih Mobil Berhenti Mendadak Data 1- Data 7	67

Tabel 0. 4 Data Latih Mobil Berhenti Mendadak Data 8- Data 1468
Tabel 0. 5 Data Latih Mobil Berhenti Kecelakaan Data 1- Data 870
Tabel 0. 6 Data Latih Mobil Berhenti Kecelakaan Data 9- Data 1671



DAFTAR GAMBAR

Gambar 2. 1 Blok Diagram Sistem Deteksi Kecelakaan Menggunakan Sensor Akselerometer	4
Gambar 2. 2 Blok Diagram Sistem Deteksi.....	5
Gambar 2. 3 Blok Diagram Sistem pengiriman pesan	6
Gambar 3. 1 Diagram Alir Penelitian	13
Gambar 3. 2 Blok diagram Sistem	16
Gambar 3. 3 Diagram Alir Cara Kerja Sistem.....	16
Gambar 4. 1 Diagram Rekayasa Kebutuhan.....	18
Gambar 4. 2 Arduino Nano	19
Gambar 4. 3 Arduino Nano Skematik	19
Gambar 4. 4 MMA 7361.....	21
Gambar 4. 5 MMA 7361 Skematik.....	21
Gambar 4. 6 Chipset LM393 dan Skematik.....	22
Gambar 4. 7 SIM 800I.....	23
Gambar 4. 8 SIM 800I Skematik	23
Gambar 4. 9 Bluetooth HC-05	24
Gambar 4. 10 Skematik Bluetooth HC-05	24
Gambar 5. 1 Diagram Perancangan dan Implementasi	27
Gambar 5. 2 Rancangan Akselerometer Pada Arduino	28
Gambar 5. 3 Rancangan Sensor 801s Pada Arduino	29
Gambar 5. 4 Rancangan SIM 800I Dengan Arduino	29
Gambar 5. 5 Rancangan <i>Bluetooth</i> Dengan Arduino.....	30
Gambar 5. 6 Ukuran Mobil <i>Remote Control</i>	31
Gambar 5. 7 Letak Sensor	32
Gambar 5. 8 Diagram alir perancangan pengambilan data latihan	33
Gambar 5. 9 Diagram alir perancangan pengambilan data Uji.....	34
Gambar 5. 10 Diagram alir fungsi <i>prior()</i>	35
Gambar 5. 11 Diagram alir fungsi <i>Gaussian()</i>	35
Gambar 5. 12 Diagram alir fungsi <i>posterior()</i>	36
Gambar 5. 13 Diagram alir perancangan pengambilan data Uji.....	40
Gambar 5. 14 Implementasi rangkaian sensor MMA 7361 dan 801s dengan Arduino nano ...	41
Gambar 5. 15 Implementasi prototipe alat pendeteksi kecelakaan	42



Gambar 6. 1 Amplitudo sensor getar dari percobaan mobil berjalan 53
 Gambar 6. 2 Amplitudo sensor akselerometer dari percobaan mobil berjalan .. 53
 Gambar 6. 3 Amplitudo sensor getar dari percobaan mobil mengerem 54
 Gambar 6.4 Amplitudo sensor akselerometer dari percobaan mobil berhenti mendadak 55
 Gambar 6. 5 Amplitudo sensor getar dari percobaan mobil menabrak..... 56
 Gambar 6. 6 Amplitudo sensor getar dari percobaan mobil menabrak..... 57
 Gambar 0. 1 Grafik Amplitudo Sensor Getaran dan Akselerometer..... 66
 Gambar 0. 2 Grafik Amplitudo Sensor Getaran dan Akselerometer..... 69
 Gambar 0. 3 Grafik Amplitudo Sensor Getaran dan Akselerometer..... 72



BAB 1 PENDAHULUAN

1.1 Latar belakang

Perkembangan diberbagai bidang saat ini sangatlah pesat khususnya dibidang transportasi tidak hanya jumlahnya saja yang bertambah dari tahun ke tahun, menurut badan pusat statistika pertumbuhan transportasi di Indonesia mengalami peningkatan yang sangat pesat, dalam kurun waktu dari 2001 hingga 2013 rata-rata pertumbuhan transportasi darat sebesar 10,34%. Tersadar terhadap peningkatan kendaraan yang sangat pesat pemerintah Indonesia menargetkan pembangunan jalan tol yang direncanakan pada tahun 2015 hingga 2019 yang ditargetkan 1000 km, berdasarkan data badan pengatur jalan tol hingga awal tahun 2017 telah dioperasikan 984 km jalan tol diseluruh tanah air, dengan adanya penambahan jalan tol baru sepanjang 1.851 km maka jalan tol yang beroperasi di Indonesia hingga akhir tahun 2018 mencapai 2.835 km (BPJT, 2018).

Adanya perbaikan dan peningkatan infrastruktur, terutama jalan dan jalan tol, akan membuat industri jasa transportasi semakin menjanjikan. Salah satunya adalah bisnis jasa sewa mobil. Ada sejumlah alasan, kenapa jasa sewa mobil lebih menjadi pilihan dari pada membeli atau memiliki mobil sendiri, terutama bagi kalangan dunia usaha. Selain menghemat anggaran, menyewa kendaraan bermotor juga menghilangkan sejumlah kerepotan, seperti soal pemeliharaan, perpanjangan surat kendaraan bermotor, bahkan menghilangkan risiko kehilangan kendaraan bermotor. Dengan menyewa, maka kebutuhan transportasi dan mobilitas bisnis tetap terjaga. bisnis jasa penyewaan mobil meningkat hingga 70% berdasarkan data organisasi pengusaha rent car daerah (ASPERDA) jumlah kendaraan yang dimiliki anggotanya sekitar 10.000 unit, meningkatnya jasa penyewaan ini mencakup penyewaan mobil berbasis aplikasi atau penyewaan mobil konvensional (Kristianto, 2018).

Berdasarkan data yang didapat dari meningkatnya usaha jasa sewa mobil perlunya upaya menjaga kenyamanan antara pemberi jasa dengan pelanggan, hal yang perlu diperhatikan ketika pemberi jasa harus menjaga asetnya berupa mobil, ketika mobil sedang disewa. Dan pelanggan harus menjaga mobil yang sedang mereka sewa. Dari data yang diambil dari tempat penyewaan mobil yaitu *Rent Car Tama* yang bertempat di cileungsi, jawa barat. Perusahaan tersebut memiliki aset mobil yang tidak sedikit, perusahaan tersebut bisa mendapatkan pelanggan setidaknya dalam 1 bulan permintaan sewa mobil 200 hingga 800 pelanggan data tersebut mencakup data peminjaman jangka panjang maupun jangka pendek, perusahaan memiliki kendala untuk mengidentifikasi kerusakan pada asetnya (Hidayat, 2018).

Penelitian yang dilakukan sudaryatmo (2016) tentang pendeteksian tabrakan pada kendaraan mobil dengan menggunakan sensor akselerometer dengan sensitivitas paling tinggi $\pm 8g$ (*gravity*). yang ditempatkan dalam mobil mainan, pendeteksian tabrakan berdasarkan percepatan kendaran yang dibaca oleh sensor akselerometer dengan percepatan yang dideteksi melalui sensor akselerometer yaitu tabrakan ringan dengan nilai percepatan $4g$ (*gravity*), tabrakan sedang dengan percepatan $20g$ (*gravity*), dan untuk kecelakaan berat sebesar $40g$ (*gravity*). Hasil penelitian didapat pada pendeteksian tabrakan dari depan 80% dan pendeteksian dari arah samping

sebesar 80% dan mengirim data status tabrakan melalui modul wireless. Dari hasil pengujian sudaryatmo (2016), setelah melakukan pengujian di dapat data yang tidak akurat dikarenakan pengambilan data hanya menggunakan sensor akselerometer saja ketika kondisi pengereman dianggap kecelakaan.

Berdasarkan kajian dari beberapa penelitian diatas penulis mengusulkan sistem pendeteksian kecelakaan menggunakan sensor akselerometer MMA 3761 untuk mengetahui percepatan dari kendaraan dan menggunakan sensor getar 801s untuk menjadikan *trigger* untuk memulai perhitungan dengan metode *naive bayes*. Dengan input nilai X,Y dari sensor Akselerometer dan nilai yang didapat dari sensor getaran.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang diatas maka dapat dirumuskan permasalahan pada skripsi ini sebagai berikut:

1. Bagaimana hasil pengambil data uji pada sensor getar 801s dan sensor akselerometer MMA 7361 untuk data latih pada setiap kondisi, yaitu kondisi berjalan, berhenti mendadak, dan kecelakaan ?
2. Bagaimana hasil sistem deteksi tabrakan menggunakan sensor akselerometer dan sensor 801s vibration menggunakan metode *naive bayes* ?
3. Bagaimana Hasil dan tingkat keberhasilan dalam pengiriman SMS menggunakan modul GSM SIM 800I ?

1.3 Tujuan

Tujuan dari pembuatan skripsi ini adalah untuk melakukan pembacaan dan pengolahan nilai sensor sesuai dengan kebutuhan mendapatkan hasil *sensing*. Agar dapat mendeteksi kecelakaan dan dapat menyimpulkan kecelakaan tersebut, yang dimaksudkan alat pendeteksi ini dapat memberi informasi kepada perusahaan jasa peminjaman mobil agar menjadi bukti kepada pelanggannya. Pembacaan meliputi sensor getarandan akselerometer ketika alat dioperasikan dan hasil pembacaan diolah menggunakan metode *naive bayes*.

1.4 Manfaat

Manfaat yang dapat diambil dari penelitian ini, yaitu :

1. Manfaat yang didapat langsung oleh Penulis, adalah penerapan ilmu yang didapat selama jenjang perkuliahaan.
2. Manfaat yang didapat langsung oleh masyarakat untuk memberi notifikasi kecelakaan, Khususnya peusahaan pemberi jasa pinjaman mobil / *rental* mobil

1.5 Batasan Masalah

Agar pembahasan tidak melebar dari latar belakang dan terfokus, maka diberikan batasan masalah yaitu:

1. Elemen pemrosesannya pada node sensor berupa Arduino nano.
2. Perancang berfokus pada hasil *sensing* berdasarkan nilai percepatan dan getaran pada kendaraan.
3. Alat deteksi ditempatkan pada mobil *remote control*.

4. Pengambilan data latih diambil sesuai dengan skenario, berupa *output* dari sensor getaran dan percepatan.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini diuraikan dan digambarkan sebagai berikut:

BAB 1 Pendahuluan

Bab ini berisikan latar belakang masalah, tujuan dan manfaat penulisan, permasalahan, batasan masalah dan juga sistematika penyusunan Penulisan.

BAB 2 Landasan Kepustakaan

Bab ini berisi tentang dasar teori yang berkaitan dengan tema dan judul penulisan yang diambil, dan menjadi acuan dasar pengerjaan skripsi ini.

BAB 3 Metodologi

Bab ini dijelaskan pembahasan metodologi yang digunakan pada penelitian.

BAB 4 Rekayasa Kebutuhan

Bab ini membahas tentang semua kebutuhan dari sistem, seperti kebutuhan user, kebutuhan fungsional dan kebutuhan non fungsional pada sistem.

BAB 5 Perancangan dan Implementasi

Bab ini menjelaskan tentang perancangan dan implementasi dari sistem.

BAB 6 Pengujian

Bab ini menjelaskan tentang pengujian dan analisis sistem yang telah dibuat.

BAB 7 Perancangan dan Implementasi

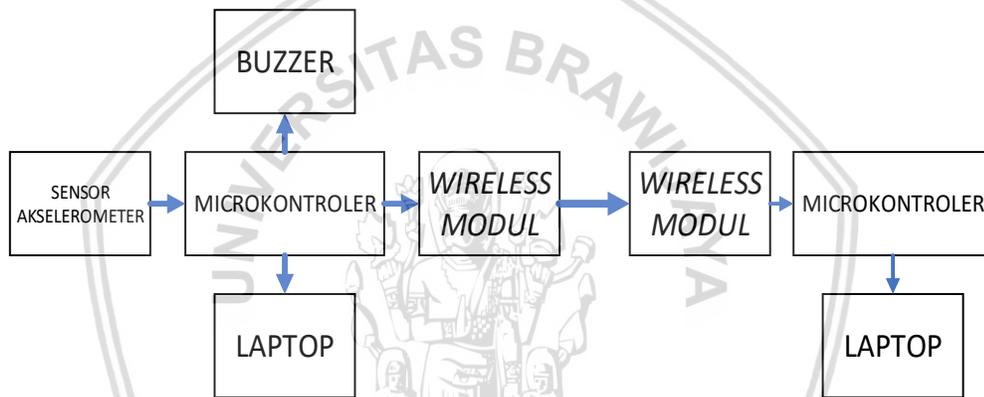
Bab ini menjelaskan tentang saran dan kesimpulan dari hasil pembuatan sistem.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tinjauan pustaka dan dasar teori yang digunakan. Tinjauan pustaka membahas tentang penelitian penelitian yang ada dan diusulkan. Dasar teori yang diperlukan untuk membahas penelitian yang akan dilakukan.

2.1 Kajian Pustaka

Pada penelitian pertama yang ditulis sudaryatmo (2016) untuk di jadikan kajian yaitu penelitian yang membahas tentang perancangan sistem deteksi kecelakaan pada mobil menggunakan sensor akselerometer. Dalam penelitian ini menggunakan mikrokontroler berupa Arduino uno, sensor menggunakan modul akselerometer MMA7455, dan data pengirimannya menggunakan nRF24L01 yang merupakan modul *wireless*. Berikut adalah gambaran sistem dari penelitian yang ditunjukkan pada gambar 2.1



Gambar 2. 1 Blok Diagram Sistem Deteksi Kecelakaan Menggunakan Sensor Akselerometer

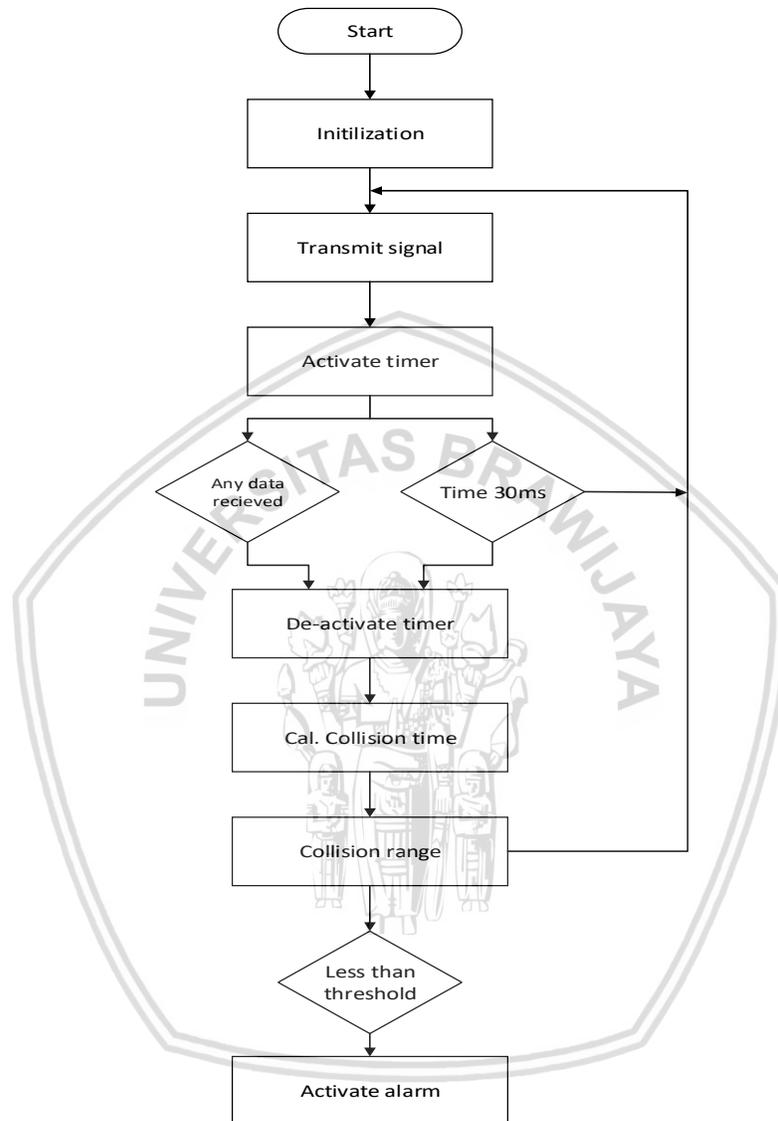
Sumber : (Sudaryatmo, 2016)

Pada gambar 2.1 menjelaskan bahwa sistem deteksi tabrakan yang diteliti oleh sudaryatmo (2016) akselerometer sebagai pembaca kondisi dari sistem saat kondisi membaca perubahan percepatan lebih dari 4g (*g-force*) mikrokontroler mengaktifkan buzzer dari sistem deteksi, data yang hasil perubahan dilihat menggunakan *serial monitor* dan notifikasi akan dikirim menggunakan *wireless module* untuk diproses mikrokontroler kedua untuk menerima notifikasi.

Penelitian diatas mengalami kendala ketika sensor akselerometer tidak bisa membedakan antara pengereman dengan kecelakaan sehingga nilai yang dihasilkan tidak sesuai untuk dijadikan notifikasi kecelakaan, saran dari penelitian (Sudaryatmo, 2016) untuk menambahkan sensor untuk pembacaan tabahan untuk memproses hasil kejadian.

Pada penelitian kedua yang dilakukan oleh ditulis oleh (Rashida Nazir, 2014), pada jurnal ini membahas tentang deteksi kecelakaan lalu lintas secara otomatis yang menggunakan sensor getaran dan fokus pada pengiriman notifikasi melalui notifikasi

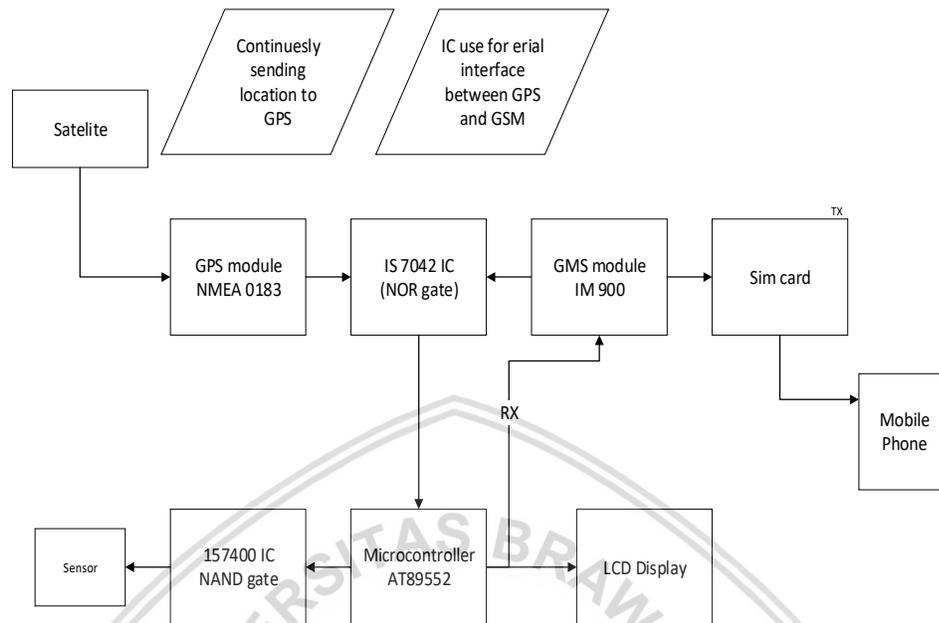
yang dikirimkan melalui *SIM Card*. Sistem ini dapat memberikan informasi lokasi kecelakaan, berikut adalah gambaran sistem deteksi kecelakaan pada gambar 2.2 dan sistem pengiriman pesan pada gambar 2.3.



Gambar 2. 2 Blok Diagram Sistem Deteksi

Sumber : (Rashida Nazir, 2014)

Pada kerja sistem deteksi pada penelitian Rashida Nazir (2014) yaitu menghubungkan sensor getar dengan mikroontroler 7400IC, ketika tabrakan kendaraan terjadi, getaran sensor akan memberikan nilai pembacaan yang sangat besar dan akan memberikan interupsi ke mikrokontroler. Ketika sistem dijalankan menginisialisasi lalu mengirimkan sinyal ke *timer*, yang akan berulang 30ms lalu setelah itu sistem menonaktifkan *timer*, jika ada getaran yang diterima maka alarm aktif.



Gambar 2. 3 Blok Diagram Sistem pengiriman pesan

Sumber : (Rashida Nazir, 2014)

Pada Gambar 2.3 kerja sistem deteksi untuk pengiriman pesan yang ditulis Rashida Nazir (2014) Ketika mikrokontroler menerima intruksi maka modem GSM akan mengirimkan sms pada penerima yang telah disimpan pada mikriokontroler, setelah berhasilmengirimkan pesan akan ditampilkan ke lcd.

Berdasarkan penelitian yang telah dijelaskan diatas, penelitian pertama merupakan dasar landasan dari penelitian ini karna penelitian ini mengembangkan dari penelitian dengan menambahkan fitur berupa sensor getar yang diterapkan pada penelitian kedua, dimana pada penelitian ini diterapkan menggunakan metode naïve bayes untuk menyimpulkan kondisi yang telah diproses dari sensor akselerometer yang diharapkan nilai yang dihasilkan lebih akurat.

2.2 Dasar Teori

Pada dasar terori ini akan membahas dasar-dasar yang digunakan dalam pembuatan sistem berupa teori tentang

2.2.1 Naive Bayes

Algoritma *Naive Bayes* adalah sebuah klasifikasi yang didasarkan pada aturan *Bayes* dan sekumpulan asumsi independensi kondisional. Independensi yang dimaksud disini adalah tidak adanya ketergantungan antara tiap fitur dalam setiap kelas objek yang diklasifikasikan. Berikut merupakan rumus dari teorema *Bayes* (Astuti, 2016):

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.1)$$

Keterangan dari Pesamaan (2.1) yaitu : (Astuti, 2016)

- $P(y|x)$: Peluang posterior (probabilitas kondisional) dari suatu hipotesis kelas y akan terjadi setelah diberikan data x .
- $P(x|y)$: Peluang likelihood dari sebuah data x terjadi akan mempengaruhi hipotesis kelas y .
- $P(y)$: Peluang prior (awal) hipotesis kelas y terjadi tanpa memperhatikan data yang diberikan.
- $P(x)$: Peluang evidence x terjadi tanpa memperhatikan hipotesis kelas lainnya, yakni jumlah total dari semua peluang likelihood yang dikalikan dengan peluang prior.

Hipotesis dalam teorema Bayes merupakan label kelas yang menjadi target dalam sebuah klasifikasi, sedangkan evidence adalah fitur yang menjadi masukan dalam klasifikasi. Naive Bayes dilambangkan dengan $P(X|Y)$, dimana X adalah masukan yang berupa fitur-fitur dan Y adalah kelas dalam sebuah klasifikasi. Notasi $P(X|Y)$ berarti peluang kelas Y didapatkan setelah fitur-fitur X diamati, notasi ini merupakan peluang likelihood dan $P(Y)$ merupakan notasi dari peluang prior. Berikut ini adalah persamaan untuk rumus Naive Bayes (Astuti, 2016):

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|y)}{P(x)} \quad (2.2)$$

Keterangan dari Pesamaan (2.2) yakni :

- $P(Y|X)$: Peluang posterior (probabilitas kondisional) dari suatu kelas Y akan terjadi setelah mengamati fitur-fitur X .
- $\prod_{i=1}^q P(X_i|y)$: Peluang likelihood dari masing-masing fitur X terjadi akan mempengaruhi kelas Y .
- $P(Y)$: Peluang prior (awal) hipotesis kelas y terjadi tanpa memperhatikan fitur yang diberikan.
- $P(X)$: Peluang evidence X terjadi tanpa memperhatikan kelas/evidence, yakni jumlah total dari semua peluang likelihood yang dikalikan dengan peluang prior.

Dalam perhitungan klasifikasi untuk setiap kelas Y yang berbeda akan mempunyai nilai $P(X)$ yang sama, sehingga dalam penentuan klasifikasi Naive Bayes selanjutnya ditentukan dari nilai peluang terbesar antara tiap kelas Y dari hasil perhitungan $\prod_{i=1}^q P(X_i|y)$ (Astuti, 2016).



Beberapa permasalahan yang ada untuk menentukan nilai peluang dari suatu kondisi yang mudah adalah dengan menghitung peluang dari data diskrit. Namun dalam kenyataannya tidak semua data tersaji dalam bentuk diskrit, tetapi ada yang berbentuk data kontinu. Untuk itu dalam melakukan proses klasifikasi terhadap data kontinu dengan Naive Bayes terdapat 2 cara yakni (Astuti, 2016):

1. Melakukan proses perubahan data kontinu menjadi data diskrit (diskritisasi) terhadap setiap fitur yang akan diestimasi.
2. Menganggap setiap fitur sesuai dengan data latih menggunakan fungsi univariate normal (Gaussian) distribution yang ditunjukkan pada Persamaan (2.3), dimana parameter utama dari fungsi Gaussian ini adalah mean (μ) dan varian (σ^2).

$$P(X = x_i | Y = Y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.3)$$

Parameter μ_{ij} bisa didapatkan dari mean pada sampel X_i (\bar{x}) dari semua data latih yang menjadi milik kelas y_i , sedangkan σ_{ij}^2 dapat dipekirakan dari varian sampel (s^2) dari data latih.

Adapun fungsi untuk mencari nilai Mean dapat dilihat pada Persamaan (2.4) berikut.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Perhitungan mean dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sample tersebut. Dimana \bar{x} merupakan rata-rata hitung, x_i merupakan nilai sample ke-i, dan n adalah jumlah sample.

Adapun fungsi untuk mencari nilai standar deviasi dapat dilihat pada Persamaan (2.5) berikut.

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

Untuk menghitung standar deviasi yaitu dengan mengurangi setiap nilai data dengan rata-rata kelompok data tersebut, selanjutnya semua hasil dijumlahkan kemudian dibagi dengan jumlah data secara keseluruhan dikurangi 1, dan terakhir hasilnya di akarkan. Dimana s merupakan standar deviasi (simpangan baku), x_i merupakan nilai x ke i, \bar{x} merupakan rata-rata, n adalah ukuran sampel.

2.2.2 Mikrokontroler

adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat



menyimpan program did umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi didalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas. Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program did MCS51 ialah mikrokomputer CMOS 8 bit dengan 4 KB Flash PEROM (Programmable and Erasable Only Memory) yang dapat dihapus dan ditulisi sebanyak 1000 kali. Mikrokontroler ini diproduksi dengan menggunakan teknologi high density non-volatile memory. Flash PEROM on-chip tersebut memungkinkan memori program untuk diprogram ulang dalam sistem (in-system programming) atau dengan menggunakan programmer non-volatile memory konvensional. Kombinasi CPU 8 bit serba guna dan Flash PEROM, menjadikan mikrokontroler MCS51 menjadi microcomputer handal yang fleksibel.

2.2.3 Akselerometer

Akselerometer adalah sensor yang digunakan untuk mengukur percepatan suatu objek. Accelerometer dapat mengukur percepatan dinamis dan percepatan statis. Pengukuran dinamis adalah pengukuran percepatan pada objek bergerak, sedangkan pengukuran statis adalah pengukuran terhadap gravitasi bumi. Berikut ini adalah susunan perangkat pada akselerometer.

a. *Microelectromechanical (MEMS)*

Microelectromechanical (MEMS) adalah sebuah miniature perangkat atau susunan perangkat yang menggabungkan komponen elektrik, sistem MEMS dapat mengkonversikan energi listrik menjadi energi mekanis, demikian sebaliknya. Teknologi MEMS memungkinkan untuk improvement dalam hal stability, sensitivity serta kualitas dari perangkat .

b. *MEMS Accelerometer*

adalah perangkat elektromekanis yang akan mengukur kekuatan percepatan. Kekuatan ini mungkin statis, seperti gaya konstan gravitasi yang menarik kaki Anda, atau kecepatan ini bisa menjadi dinamis disebabkan oleh pergerakan atau getaran. Percepatan yang diukur dengan accelerometer belum tentu percepatan koordinat (laju perubahan kecepatan). Sebaliknya, accelerometer melihat percepatan terkait dengan fenomena berat yang dialami.

prinsip kerja dari accelerometer adalah prinsip percepatan (acceleration). Sebuah per dengan beban dan dilepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu lalu berhenti. Bila ada sesuatu yang menggoncangkannya maka beban akan berayun kembali.

Pengukuran kapasitansi inilah yang umumnya menjadi hasil pengukuran chip. Agar sensor bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing-masing. Akselerometer saat ini banyak memiliki fungsi di antaranya yaitu :

- a. Pada Kamera video juga memanfaatkan accelerometer untuk menstabilkan tangkapan gambar, untuk bisa meminimalisir blur saat menangkap gambar.
- b. Untuk kontrol antarmuka pengguna, accelerometer sering digunakan untuk menyajikan pandangan landscape atau potret layar perangkat, berdasarkan cara perangkat yang sedang digunakan. Misalnya mengganti tampilan layar dari vertical (portrait) menjadi horizontal (landscape).
- c. Pada accelerometer ponsel ada yang namanya fungsi shake control, dengan fungsi ini maka dengan hanya menggoyangkan ponsel kita bisa mengubah fitur, misalnya mengubah lagu yang sedang di putar ke lagu selanjutnya atau lagu sebelumnya.
- d. Penggunaan lain dari accelerometer adalah sebagai pedometer, atau penghitung langkah. Dipakai di banyak gadget sebagai monitor banyaknya langkah saat berolah raga.

2.2.4 Sensor Getaran

Sensor getaran adalah suatu alat yang berfungsi untuk mendeteksi adanya getaran dan akan diubah dalam ke dalam sinyal listrik. Macam macam sensor dibagi dua yaitu:

2.2.4.1 Kontak

Sensor ini disebut juga *cassing measurement*. Sensor yang digunakan adalah sensor seismic transducer, yaitu sensor yang digunakan untuk mengukur kecepatan dan percepatan. Untuk mengukur kecepatan menggunakan velocity probe dan velomitor probe, sedangkan untuk mengukur percepatan menggunakan sensor acceleration probe.

A. Velocity probe

Ujung sensor ini akan bersentuhan langsung dengan benda yang akan diukur fibrasinya, sensor ini berfungsi untuk mengukur getaran dari suatu alat atau mesin menggunakan kecepatan sebagai parameternya. Adapun kontruksi dari sensor getar ini yaitu: massa, kumparan, pegas, magnet permanen, dan damper konektor, dan *casing*.

Prinsip kerja velocity probe sesuai dengan hukum fisika yaitu apabila suatu konduktor/kumparan yang dikelilingi oleh medan magnet kemudian konduktor bergerak terhadap medan magnet atau medan magnet bergerak terhadap konduktor maka akan menimbulkan suatu tegangan induksi pada konduktor. Apabila transducer ini ditempatkan pada bagian mesin yang bergetar, maka transducer inipun akan ikut bergetar, sehingga kumparan yang ada di dalamnya akan bergerak relatif terhadap medan magnet sehingga akan menghasilkan tegangan listrik pada ujung kawat kumparannya. Dengan mengolah sinyal listrik dan transdusernya, maka getaran dapat diukur.

B. Acceleration Probe

Termasuk sensor kontak yang berfungsi untuk mengukur getaran dengan mengukur kecepatan dari mesin tersebut. Prinsip kerja dari Acceleration Probe adalah Getaran mekanis yang menimbulkan gaya akan mengenai bahan piezoelectric tersebut sehingga bahan piezoelectric tersebut menghasilkan

muatan listrik. Tetapi arus listrik yang dihasilkan oleh piezoelectric ini sangat kecil, sehingga diperlukan alat lain agar menghasilkan muatan listrik yang standard. Karena muatan listrik yang ditimbulkan oleh piezoelectric sangat kecil maka didalamnya dipasang rangkaian elektronik/amplifier yang dapat membangkitkan muatan agar muatan listrik yang dihasilkan oleh bahan piezoelectric menjadi lebih besar. Besar muatan listrik yang dihasilkan oleh bahan piezo electric sebesar picocoulombs per g. Sedangkan besarnya sinyal yang dihasilkan setelah didalamnya dipasang penguat, mempunyai sensitivitas 50 mv per g.

2.2.4.2 Non Kontak

Sensor non-kontak biasanya disebut Shaft Relative Measurement. Sensor yang digunakan adalah proximity probe (Eddy current probe). Untuk proxymity probe, yang diukur adalah perpindahannya. Untuk sensor non-kontak, probe dan mesin atau media tidak bersentuhan langsung. Untuk menggunakan sensor proximity probe ada beberapa syarat yang harus terpenuhi agar dapat menghasilkan pengukuran yang presisi, diantaranya adalah :

- Roundness (kelingaran) dari mesin yang akan diukur harus bagus untuk menghasilkan bacaan yang bagus pula
- Run out

2.2.5 SMS (Short Message Service)

SMS adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada perangkatlain. Teks ersebut bisa terdiri dari kata-kata atau nomor kompikasi alpha *numeric*. SMS di ciptakan sebagai standar pesan oleh *European Telekomunication Standar Institute*. Panjang setiap pesan dalam SMS maksimal terdiri dari 160 karakter jika menggunakan alphabet latin dan 70 karakter jika menggunakan alphabet non-latin.

Teknologi SMS memiliki beberapakeunggulan yaitu pesan yang akan di kirim akan selalu di usahakan untuk dikirim ketujuan jika suatu nomor ujuan tidak aktif atau di luar jangkauan, makapesan akan di simpan di SMS center.

2.2.6 Arus Listrik DC

Arus listrik DC (Direct current) merupakan arus listrik searah. Pada awalnya aliran arus pada listrik DC dikatakan mengalir dari ujung positif menuju ujung negatif. Semakin kesini pengamatan-pengamatan yang dilakukan oleh para ahli menunjukkan bahwa pada arus searah merupakan arus yang alirannya dari negatif (elektron) menuju kutub positif. Nah aliran-aliran ini menyebabkan timbulnya lubang-lubang bermuatan positif yang terlihat mengalir dari positif ke negatif.

Listrik DC (direct current) biasanya digunakan oleh perangkat lektronika. Meskipun ada sebagian beban selain perangkat elektronika yang menggunakan arus DC (contohnya; Motor listrik DC) namun kebanyakan arus DC digunakan untuk keperluan beban elektronika. Beberapa beban elektronika yang menggunakan arus listrik DC diantaranya: Lampu LED (Light Emiting Diode), Komputer, Laptop, TV, Radio,

dan masih banyak lagi. Selain itu listrik DC juga sering disimpan dalam suatu baterai, contohnya saja baterai yang digunakan untuk menghidupkan jam dinding, mainan mobil-mobilan dan masih banyak lagi. Intinya kebanyakan perangkat yang menggunakan listrik DC merupakan beban perangkat elektronika.

2.2.7 Bluetooth

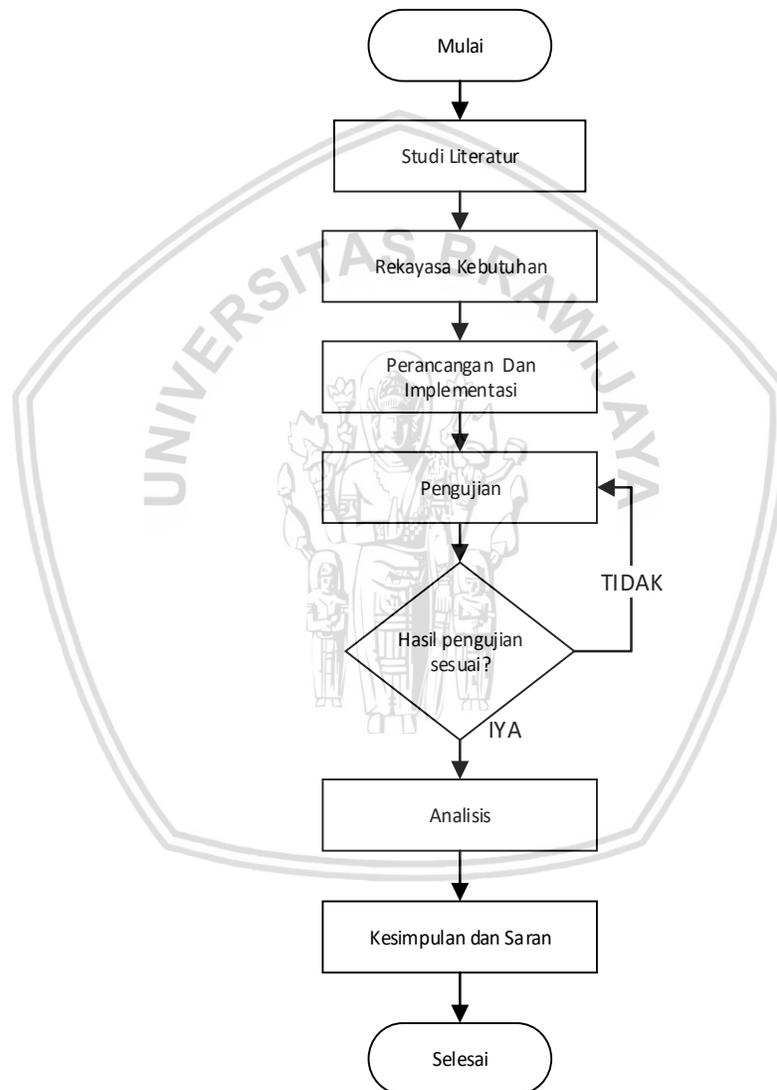
Bluetooth adalah sebuah teknologi komunikasi wireless atau tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz (antara 2.402 GHz s/d 2.480 GHz) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan juga suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas.

Pada dasarnya teknologi bluetooth ini diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan media kabel dalam melakukan pertukaran data atau informasi, tetapi juga mampu menawarkan fitur yang bagus atau baik untuk teknologi mobile wireless atau tanpa kabel, dengan biaya yang relatif rendah, konsumsi daya rendah, interoperability yang sangat menjanjikan, mudah dalam pengoperasiannya dan juga mampu menyediakan berbagai macam layanan.



BAB 3 METODOLOGI

Pada bab ini akan di jelaskan tentang langkah – langkah yang akan di tempuh dalam penyusunan skripsi dimana meliputi studi kasus, studi literatur, rekayasa kebutuhan, perancangan dan impleentasi, pengujian dan analisis, kesimpulan dan saran, alur metode penelitian yang di lakukan secara umum dapat dilihat dari diagram alir pada gambar 3.1.



Gambar 3. 1 Diagram Alir Penelitian

3.1 Studi Literatur

Pada bagian ini dibahas mengenai teori yang mendukung penelitian rancang bangun alat pendeteksi tingkat keparahan kecelakaan mobil menggunakan metode *naive bayes*. Teori-teori yang digunakan dalam tahapan studi literature ini berasal dari buku, Jurnal, *website* resmi, dan artikel. Berikut merupakan dasar teori yang digunakan sebagai bahan studi:

1. Transportasi Darat.
Mempelajari terkait pengertian macam macam transportasi, jumlah volume kendaraan yang setiap tahun meningkat, apa yang menyebabkan volume kendaraan meningkat.
2. Kecelakaan
Mempelajari penyebab kecelakaan, melihat data survey tingkat kecelakaan lalulintas, khususnya mobil.
3. Investigasi Kecelakaan
Mempelajari bagaimana penanganan kecelakaan di lakukan, data apa saja yang mendukung penyelesaian investigasi khususnya, penanganaaan kecelakaan mobil.
4. Arduino Nano
Mempelajari Komfigurasi penggunaan Arduino Nano sebagai otak utama dalam pengolah sistem baik segi konfigurasi pin secara perangkat keras maupunn dari segi penggunaan algoritma dan logika yang sesuai dengan hasil yang diinginkan.
5. MMA 7361
Mempelajari tentang spesifikasi sensor dan bagaimana cara konfigurasi pin sensor untuk membaca nilai percepatan benda yang bergerak.
6. Vibrator 801s
Mempelajari tentang spesifikasi sensor dan bagaimana cara mengkonfigurasi pin sensor untuk membaca besar kecilnya suatu hentakan atau benturan kontak langsung.
7. SIM 800I
Mempelajari tentang spesifikasi sensor dan bagaimana cara mengkonfigurasi in sensor untuk mengirimkan hasil pembacaan melalui SMS
8. Bluetooth HC-05
Mempelajari tentang spesifikasi sensor dan bagaimana cara mengkonfigurasi pin sensor untuk mengirimkan hasil pembacaan ke serial monitor yang ada di *handphone*.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan diperlukan untuk menganalisis apa saja yang dibutuhkan oleh sistem pada penelitian yang dilakukan, sehingga dapat sistem dapat melakukan sesuai dengan yang diharapkan. Berikut beberapa kebutuhan fungsional yang dibutuhkan oleh sistem dalam penelitian ini :

1. Sensor MMA 7361 dapat membaca tingkat perubahan percepatan dalam satuan *g-force*.

2. Sensor Vibrator 801s dapat membaca kontak langsung sehingga bisa membedakan tabrakan atau berhenti mendadak.
3. Data dari sensor dapat diolah untuk mengklasifikasikan hasilnya menjadi jenis kecelakaan ringan, sedang atau berat berdasarkan metode *Naive Bayes*.
4. Sistem menggunakan SIM 800I dan Bluetooth HC-05 untuk notifikasi yang di kirim ke *Handphone*.

3.2.1 Kebutuhan Perangkat Keras

Berikut beberapa kebutuhan perangkat keras yang digunakan untuk membuat sistem pada penelitian ini :

1. Mikrokontroler Arduino Nano
2. Sensor akselereometer MMA 7361
3. Sensor *Vibrator 801s*
4. SIM 800I
5. Bluetooth HC-05
6. Mobil *Remote control*
7. *Push Button*
8. Kabel jumper
9. *Handphone*
10. Laptop

3.2.2 Kebutuhan Perangkat Lunak

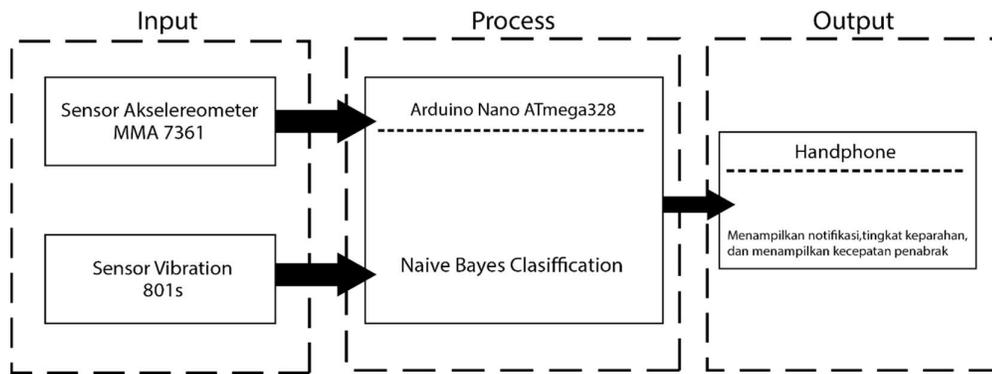
Pada bagian ini menganalisis terkait kebutuhan perangkat lunak apa saja yang dibutuhkan oleh sistem pada penelitian ini adalah Arduino IDE 1.6.6 serta arduino library.

3.3 Perancangan dan Implementasi.

Perancangan dan implementasi dalam pembuatan sistem ini meliputi dua bagian yaitu perancangan sistem dan implementasi sistem secara garis besar sistem dibangun dari perancangan perangkat keras dan perancangan perangkat lunak.

3.3.1 Perancangan Perangkat Keras

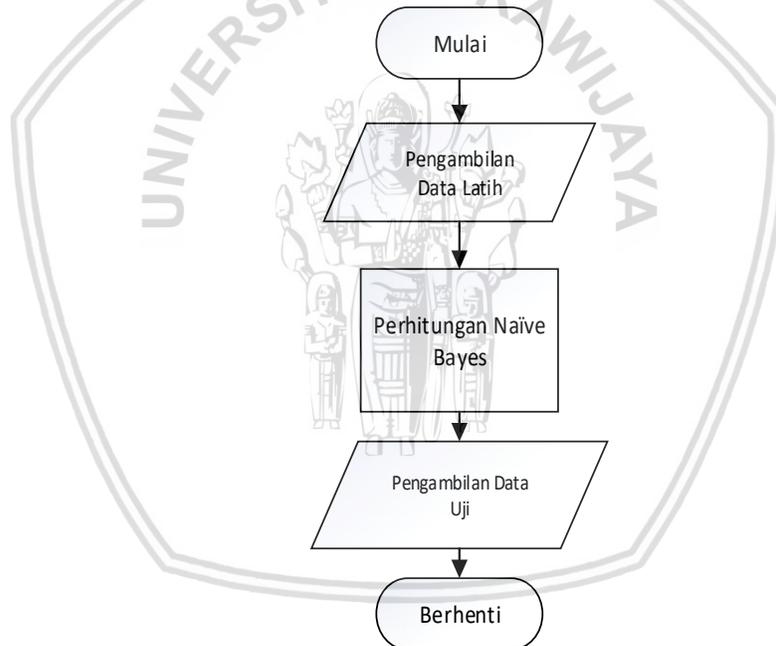
Pada perancangan perangkat keras ini yaitu berupa blok diagram dari sistem yang akan dibuat. Telihat dari gambar 3.2 terdapat tiga bagian utama yakni input, Process, dan Output. Pada bagian input terdapat dua buah sensor yang berfungsi untuk membaca perubahan percepatan dan membaca getaran akibat tabrakan, lalu pada bagian Process terdapat Arduino Nano yang berfungsi sebagai mikrokontroler pengeolah data dari sensor untuk dilakukan klasifikasi menggunakan metode *naive bayes*, dan pada bagian Output akan menampilkan.



Gambar 3. 2 Blok diagram Sistem

3.3.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak berupa logika untuk algoritma yang digunakan sesuai dengan tujuan utama dari pembuatan sistem yakni pembuatan program pengklasifikasian dengan metode Naive Bayes.



Gambar 3. 3 Diagram Alir Cara Kerja Sistem

Terlihat dari Gambar 3.3 diatas bahwa sistem akan menjalankan proses pembacaan sensor berupa akselerometer dan sensor getaran lalu nilai akan di proses untuk pengkalsifikasian menggunakan metode naïve bayes dan hasil akan di tampilkan. Berikut penjelasan lebih terperinci :

1. Pengambilan data latih di ambil 3 kondisi yaitu berjalan, berhenti mendadak dan kecelakaan.
2. Perhitungan naïve bayes mengambil sample perhitungan untuk dihitung manual.

3. Pengujian dilakukan ketika tahapan pengambilan data telah dan perhitungan sudah berjalan dengan baik.

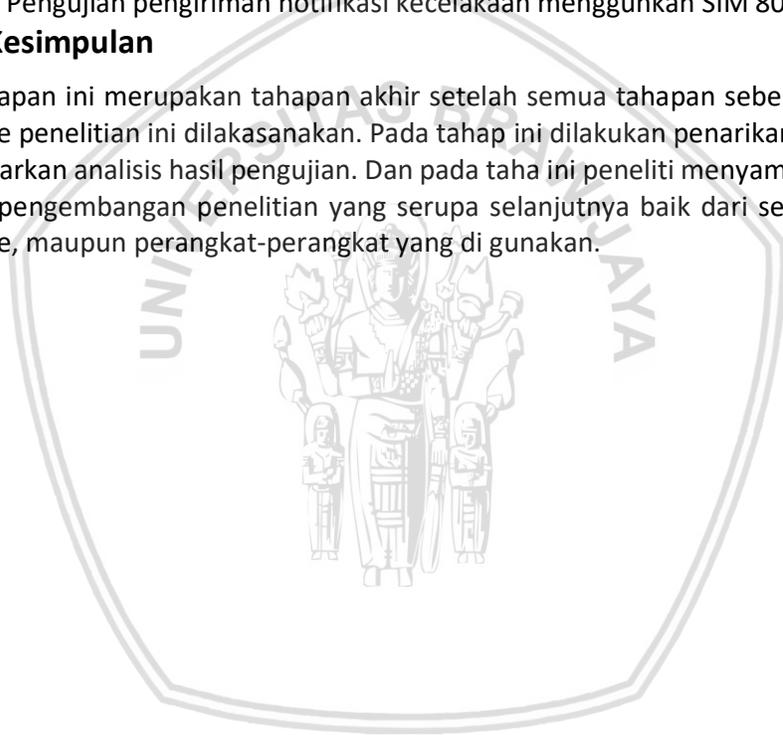
3.4 Pengujian dan Analisis

Pada tahap ini akan dilakukan pengujian dan analisis hasil pengujian yang mana akan dilaksanakan sesuai dengan parameter perancangan sistem. Ada beberapa skenario yang akan dilakukan untuk pengujian sistem, yaitu :

1. Pengujian sensor sistem deteksi tabrakan pada setiap kondisi yaitu kondisi berjalan, berhenti mendadak, dan kecelakaan yang di uji mengacu pada hasil penelitian sudaryatmo (2016) dimana sistem dinyatakan tabrakan ketika nilai $4g$ (*g-force*) dimana hasil akan di masukkan kedalam data latih.
2. Pengujian sensor sistem deteksi tabrakan dengan klasifikasi naïve bayes pada setiap kondisi yaitu kondisi berjalan, berhenti mendadak, dan kecelakaan.
3. Pengujian pengiriman notifikasi kecelakaan menggunakan SIM 800l.

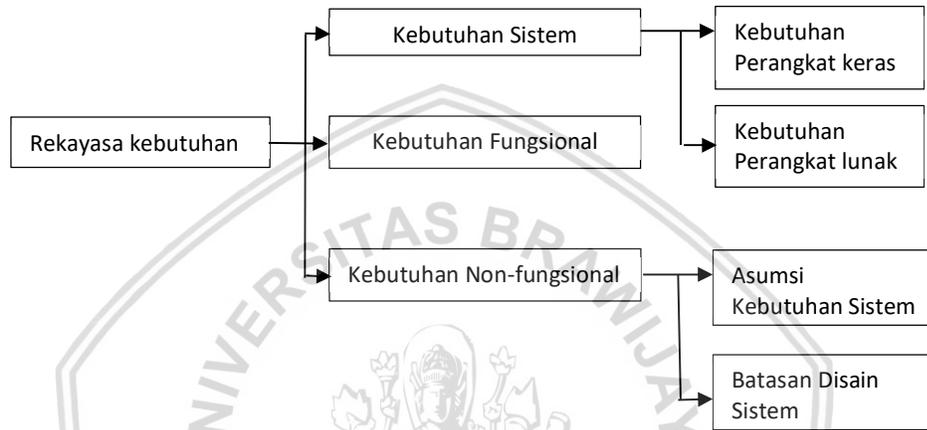
3.5 Kesimpulan

Tahapan ini merupakan tahapan akhir setelah semua tahapan sebelumnya pada metode penelitian ini dilaksanakan. Pada tahap ini dilakukan penarikan kesimpulan berdasarkan analisis hasil pengujian. Dan pada tahap ini peneliti menyampaikan saran untuk pengembangan penelitian yang serupa selanjutnya baik dari segi penulisan, metode, maupun perangkat-perangkat yang di gunakan.



BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan menjelaskan secara terperinci terkait rekayasa kebutuhan yang dibutuhkan untuk merancang sistem deteksi tabrakan menggunakan akselerometer dan sensor getar, pada bab ini terdapat beberapa bagian dari bab yaitu kebutuhan sistem, kebutuhan fungsional, dan kebutuhan non-fungsional. Pada gambar 4.1 menjelaskan diagram rekayasa kebutuhan.



Gambar 4. 1 Diagram Rekayasa Kebutuhan

4.1 Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengkaji semua kebutuhan yang diperlukan untuk sistem pendeteksi tingkat keparahan kecelakaan ini. Dalam melakukan analisis kebutuhan sistem ini memerlukan penjabaran yakni kebutuhan fungsional dan kebutuhan non fungsional, dimana kebutuhan non fungsional dibagi menjadi dua terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak.

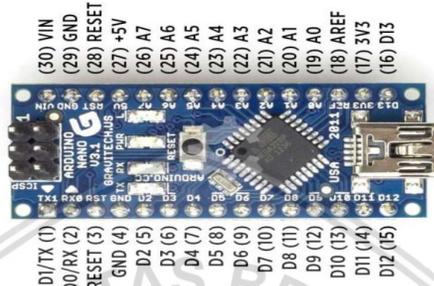
4.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras untuk membangun sistem ini akan dijelaskan antara lain sebagai berikut:

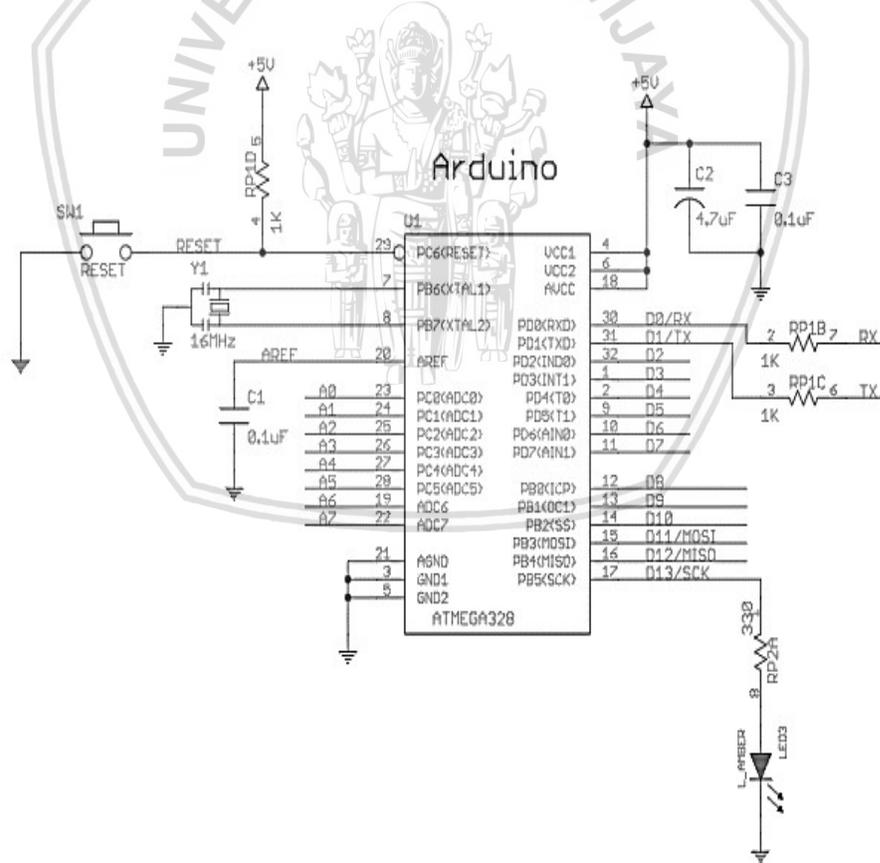
4.1.1.1 Arduino nano

Untuk Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. (Banzi, 2014)

Arduino merupakan sebuah platform dari physical computing yang bersifat open source. Pertama-tama perlu dipahami bahwa kata “platform” di sini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan Integrated Development Environment (IDE) yang canggih. IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory microcontroller. Berikut adalah gambar Arduino Nano yang tertera pada Gambar 4.2 (Banzi, 2014).



Gambar 4. 2 Arduino Nano
 Sumber : www.Arduino.cc, 2014



Gambar 4. 3 Arduino Nano Skematik
 Sumber : www.Arduino.cc, 2014

Penjelasan konfigurasi pin Arduino Nano yang memiliki 30 pin yang terlampir pada Gambar 2.4 sebagai berikut:

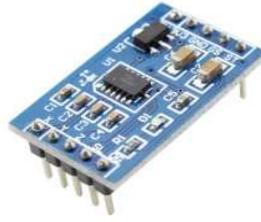
1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital.
2. GND merupakan pin ground untuk catu daya digital.
3. AREF merupakan Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
4. RESET merupakan Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino
5. Serial RX (0) merupakan pin yang berfungsi sebagai penerima TTL data serial.
6. Serial TX (1) merupakan pin yang berfungsi sebagai pengirim TT data serial.
7. External Interrupt (Interupsi Eksternal) merupakan pin yang dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
8. Output PWM 8-Bit merupakan pin yang berfungsi untuk `analogWrite()`.
9. SPI merupakan pin yang berfungsi sebagai pendukung komunikasi.
10. LED merupakan pin yang berfungsi sebagai pin yang diset bernilai HIGH, maka LED akan menyala, ketika pin diset bernilai LOW maka LED padam. LED Tersedia secara built-in pada papan Arduino Nano.
11. Input Analog (A0-A7) merupakan pin yang berfungsi sebagai pin yang dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi `analogReference`.

Sumber daya Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi HIGH (Banzi, 2014).

4.1.1.2 Sensor MMA 7361

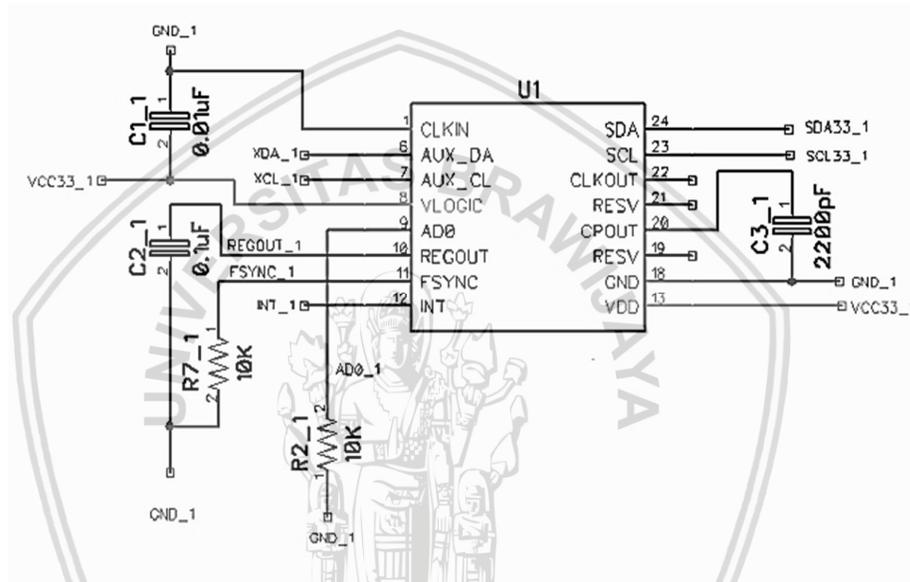
Sensor MMA 7361 adalah sensor akselerometer, prinsip kerja dari Accelerometer yaitu prinsip percepatan (acceleration). Sebuah perbandingan dengan beban dan dilepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu akan berhenti. Bila ada sesuatu yang menggoncangkannya maka beban akan berayun kembali.

Pengukuran kapasitansi inilah yang umumnya menjadi hasil pengukuran chip. Agar sensor bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing-masing. berikut adalah gambar akselerometer pada gambar 4.4 dan skematik akselerometer gambar 4.5.



Gambar 4. 4 MMA 7361

Sumber : www.invensense.com,2016



Gambar 4. 5 MMA 7361 Skematik

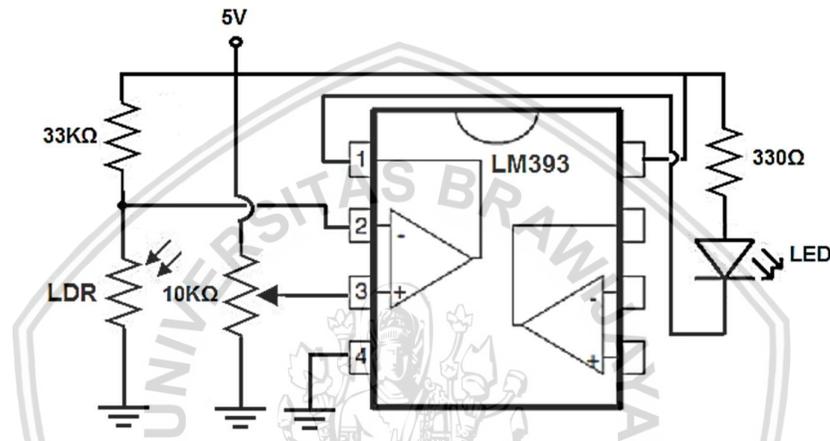
Sumber : www.invensense.com,2016

4.1.1.3 Sensor vibration 801s

Modul ini dapat mendeteksi getaran atau guncangan yang dapat diatur tingkat getaran yang diinginkan melalui trimpot yang tersedia di modul. Menggunakan 801s sebagai trigger yang mengeluarkan output analog sehingga microcontroller dapat mengenali jenis getaran yang terjadi berikut adalah gambar sensor 801s yang terdapat pada **Gambar 4.6** adalah chipset LM393 .

Spesifikasi dari sensor sebagai berikut :

- Tegangan 3.3-5v DC
- Menggunakan chipset LM393
- Output analog



Gambar 4. 6 Chipset LM393 dan Skematik
 Sumber : www.elecrow.com, 2016

4.1.1.4 SIM 800I

Modul GSM Sim800I ini berfungsi sebagai pengiriman data yang terkonfigurasi dengan GSM/GPRS. Alat ini bekerja di frekuensi 850MHz, di dalam modul ini terdapat 12 pin yang dapat berfungsi sebagai berikut:

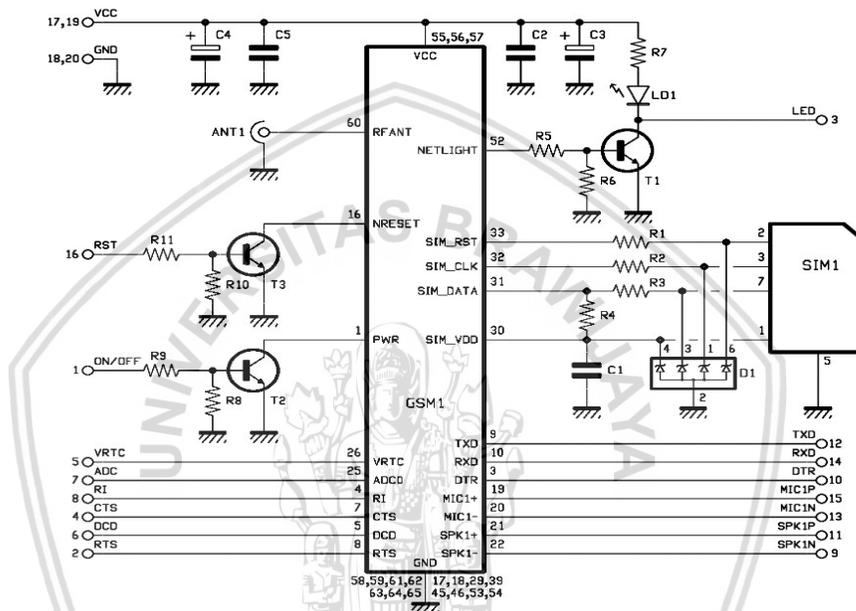
- NET = Antena
- VCC = +3,7v - 4,2v
- RST = Reset
- RXD = RX data serial
- TXD = TX data serial
- GND = Ground
- Ring
- DTR
- MICP = Microphone+
- MICN = Microphone-
- SPKP = Speaker+
- SPKN = Speaker-

Modul GSM Sim800I hanya dapat di operasikan dengan tegangan +3,7v - 4,2v jika lebih atau kurang dari itu modul ini tidak bisa di gunakan, pada gambar 4.7 adalah gambar dari modul SIM 800I dan sekematiknya pada gambar 4.8.



Gambar 4. 7 SIM 800L

Sumber : www.belajarduino.com,2014



Gambar 4. 8 SIM 800L Skematik

Sumber : www.belajarduino.com,2014

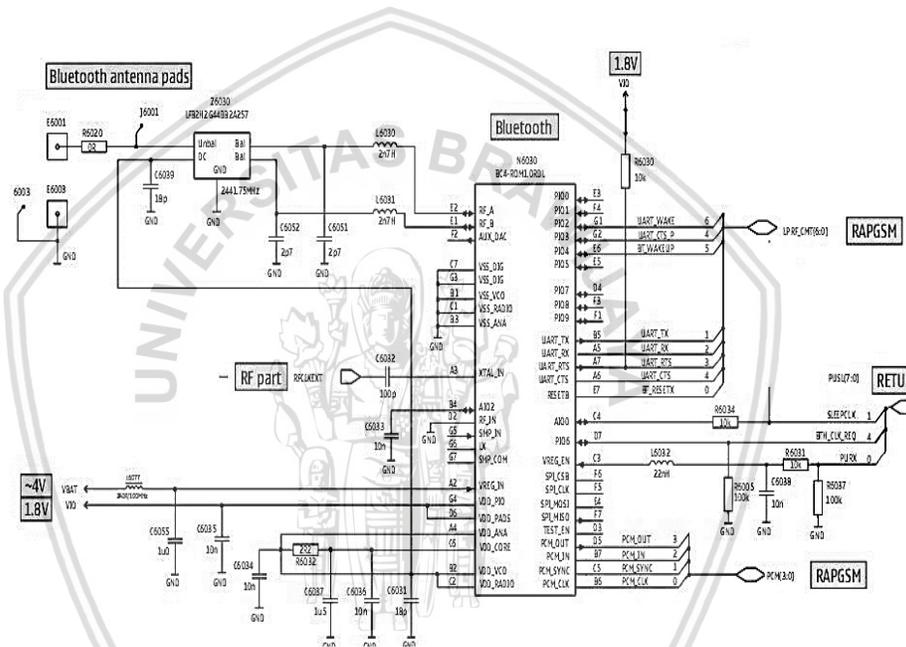
4.1.1.5 Bluetooth HC-05

Bluetooth Module HC-05 merupakan module komunikasi nirkabel pada frekuensi 2.4GHz dengan pilihan koneksi bisa sebagai slave, ataupun sebagai master. Sangat mudah digunakan dengan mikrokontroler untuk membuat aplikasi wireless. Interface yang digunakan adalah serial RXD, TXD, VCC dan GND. Built in LED sebagai indikator koneksi bluetooth.

Tegangan input antara 3.6 ~ 6V, jangan menghubungkan dengan sumber daya lebih dari 7V. Arus saat unpaired sekitar 30mA, dan saat paired (terhubung) sebesar 10mA. 4 pin interface 3.3V dapat langsung dihubungkan ke berbagai macam mikrokontroler (khusus Arduino, 8051, 8535, AVR, PIC, ARM, MSP430, etc.). Jarak efektif jangkauan sebesar 10 meter, meskipun dapat mencapai lebih dari 10 meter, namun kualitas koneksi makin berkurang, berikut merupakan gambar yang di tunjukkan pada gambar 4.9 dan gambar 4.10 merupakan skematik HC-05.



Gambar 4. 9 Bluetooth HC-05
 Sumber : www.Geraicerdas.com,2017



Gambar 4. 10 Skematik Bluetooth HC-05
 Sumber : www.Geraicerdas.com,2016

4.1.2 Kebutuhan Perangkat Lunak

Kebutuhan antarmuka perangkat lunak merupakan perangkat lunak yang dibutuhkan untuk membangun sistem. Kebutuhan antarmuka perangkat keras akan dijelaskan.

1. Windows 7 Ultimate

Perangkat lunak yang berfungsi sebagai sistem operasi pada perangkat PC atau Laptop yang dipakai dalam penelitian untuk melakukan perancangan sistem yaitu untuk menjalankan Arduino IDE. Juga terdapat beberapa sistem operasi yang didukung diantaranya windows 7, Windows 8, Windows 10 dan juga Linux. Pada penelitian ini peneliti menggunakan Windows 7 sebagai sistem operasi

pada perangkat laptop yang digunakan karena dapat menjalankan aplikasi Arduino IDE tanpa adanya kendala.

2. Arduino IDE versi 1.8.4

Merupakan perangkat lunak yang berfungsi sebagai editor untuk menuliskan kode program, *compile* serta *upload* program ke arduino MEGA pada sistem. Bahasa yang digunakan adalah bahasa khusus arduino IDE yang mirip dengan bahasa C. Juga dilengkapi dengan berbagai fungsi dan juga *library* yang mendukung dalam perancangan sistem diantaranya *library* kamera, servo, dan spi. Arduino IDE juga dapat digunakan sebagai *serial monitoring* untuk melakukan beberapa pengujian pada sistem yang sedang berjalan pada mikrokontroler yang terhubung langsung ke PC. Arduino IDE dapat dioperasikan pada sistem operasi Windows 7, Windows 8, Windows 10 dan juga Linux.

3. *Liblary*

Arduino IDE terdapat *liblary* untuk mempermudah dalam memprogram . contohnya "math.h" *liblary* yang digunakan untuk melakukan perhitungan matematika yang cukup rumit.

4.2 Kebutuhan Fungsional

Berikut ini adalah kebutuh fungsional yang haarus mampu dilakukan oleh sistem:

- 1. Sensor akselerometer dapat membaca perubahan percepatan yang terdeteksi.**
Sensor akselerometer MMA 7361 untuk membaca dan mengakuisisi nilai percepatan dari mobil yang dideteksi tertabrak. Nilai percepatan di dapat dari nilai pembacaan perubahan gerakan yang tiba tiba dari nilai X dan Y dalam satuan *g-force*. Sehingga nilai tersebut dapat di proses untuk mendapatkan nilai dari percepatannya.
- 2. Sensor *vibration* dapat membaca jika terjadi kontak langsung pada mobil yang tertabrak.**
Sensor *vibration* 801s bertugas untuk membaca dan mengakuisisi nilai getaran yang di dihasilkan untuk mengetahui apakah mobil dalam posisi tertabrak atau berhenti mendadak. dan juga sensor ini dapat mengetahui apakah mobil di tabrak dari depan, belakang, kanan, ataupun kiri.
- 3. Data dari sensor dapat diolah untuk mengklasifikasikan hasilnya menjadi jenis tabrakan, dan kecepatan penabrak yang berbeda berdasarkan hasil data yang menggunakan sensor akselerometer saja dan dibandingkan dengan akselerometer dan sensor getar.**
Pada fungsionalitas ini bertugas untuk melakukan prose pendataan yang didapat dari sensor MMA 7361 dan sensor 801s
- 4. Sistem dapat menampilkan hasil dari tabrakan menggunakan *SIM 800I* yang memberikan notifikasi melalui SMS.**
Pada fungsi ini bertugas melakukan proses pengiriman notifikasi ke *handphone* menggunakan SMS

5. **Sistem dapat menampilkan hasil dari tabrakan menggunakan *Bluetooth* yang di asumsikan memberikan notifikasi.**

Pada fungsi ini karna penelitian ini tidak berfokus pada pengiriman data dan hanya berfokus untuk mengklasifikasikan sebuah data tabrakan.

4.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari asumsi kebutuhan dan batasan disain sistem.

4.3.1 Asumsi kebutuhan sistem

1. Alat yang di buat bekerja ketika dalam posisi dinyalakan akan memberikan notifikasi kecelakaan ketika pembacaan akselerometer melebihi 4g.
2. Mikrokontroler yang di gunakan adalah *Arduino Nano*.
3. *Arduino Nano* berfungsi mengatur sistem antar *sensing* dan *actuator*.
4. Yang di hasilkan alat ini berupa data kecelakaan atau tidaknya, dan dikirim menggunakan modul SIM 800I dan HC 05
5. *Prototype* menggunakan mobil *remote control*.

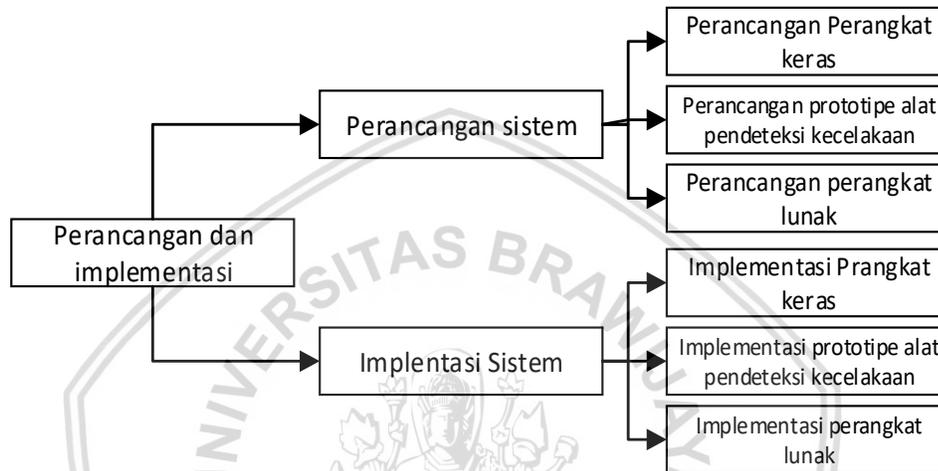
4.3.2 Batasan Disain Sistem

Dalam pembuatan sistem deteksi tabrakan terdapat beberapa batasan sehingga lingkup pembahasan, perancangan, maupun mengimplementasikan tidak meluas. Adapun batasan-batasan desain sistem ini adalah sebagai berikut:

1. Sistem melakukan klasifikasi 3 jenis kondisi yaitu kondisi berjalan, kondisi berhenti mendadak dan kondisi kecelakaan
2. Sistem menunjukan jika ada kecelakaan dengan data yang diambil dari mobil *Remote Control*.
3. Sistem melakukan test dengan hanya melakukan maju dan mundur pada Mobil *Remote Control*.
4. Sistem tidak membedakan seberapa parah kecelakaan yang terjadi hanya memberikan hasil ada tidaknya kecelakaan.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan di jelaskan perancangan dan implementasi dari sistem yang akan di buat untuk penelitian yang akan di lakukan yaitu “Rancang Bangun Pendeteksi Kecelakaan Pada Mobil Menggunakan Sensor Akselerometer Dan Sensor *Vibration* 801s”. Dalam perancangan ini sendiri menjelaskan proses perancangan dan implementasi dari sisi perangkat keras dan perangkat lunak. Pada gambar 5.1 menjelaskan diagram Perancangan dan Implementasi.



Gambar 5. 1 Diagram Perancangan dan Implementasi

5.1 Perancangan Sistem

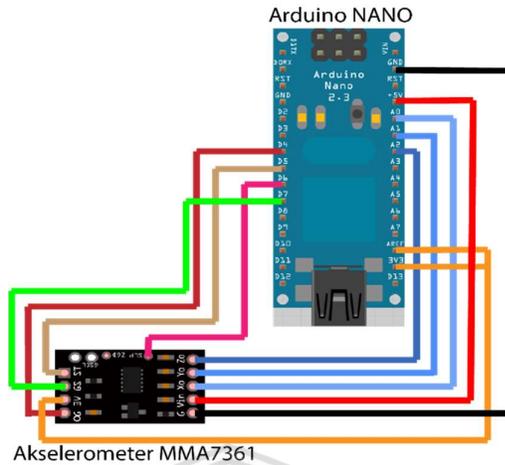
Perangkat sistem yang akan dirancang di jabarkan dalam sub bab yang di mulai dari perancangan Prototipe alat, perancangan perangkat keras dan perancangan perangkat lunak.

5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras ini di sesuaikan dengan analisis kebutuhan perangkat keras yang berada pada spesifikasi masing-masing dan sesuai dengan yang di diharapkan. Pada perancangan perangkat keras ini merupakan penjabaran secara terperinci antara pin pin yang di gunakan tiap komponen, dimana komponen *vibration 801s* dan akselerometer *MMA7361* adalah komponen utama yang di hubungkan dengan Arduino nano, serta komponen lainya berupa Sim 800I dan *Bluetooth HC-05* yaitu komponen *output*. rancangan perangkat keras ditunjukan pada subbab berikut.

5.1.1.1 Perancangan Rangkaian akselerometer MMA 7361

Pada diagram skematik diatas terdapat beberapa koneksi koneksi antar akselerometer dengan pin dengan aduino. Konektor antara lain menggunakan analog input dan digital input, dimana analog input di gunakan untuk pembacaan perubahan percepatan yang di dihasilkan sensor dan sensor digital untuk mengirimsignal *HIGH* dan *LOW*.



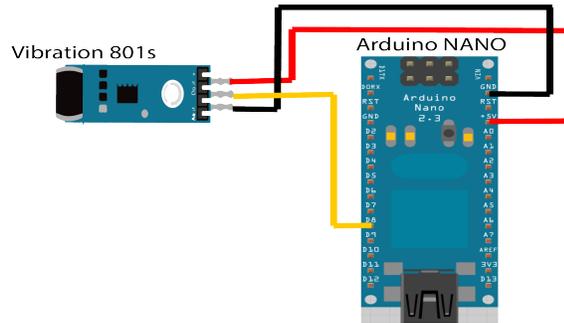
Gambar 5. 2 Rancangan Akselerometer Pada Arduino

Pada Tabel 5.2 akan menunjukan pin antar sensor MMA7361 dengan Arduino nano secara terperinci. Sistem ini menggunakan sensor akselerometer MMA7361 yang memiliki 10 pin diantaranya X, Y, dan Z yaitu nilai *axis analog output* di hubungkan dengan pin Arduino nano A0, A1, A2. SLP(*sleep*) diamana nilai ini menjadi *Input HIGH* dan *LOW* untuk kondisi *sleep* dan *normal* di hubungkan dengna in D6 pada arduino nano. 0G untuk mengetahui titik pusat sensor. GS untuk mengetahui tingkat pembacaan besar kecilnya perubahan gerakan yang di hubungkan dengan pin 7 di Arduino nano. pin ST(*self test*) diamana nilai ini menjadi *Input HIGH* dan *LOW* untuk kondisi *normal* dan *test* di hubungkan langsung pada pin 4 arduino nano , dan pin 5v,3v3, dan GND.

Tabel 5. 1 Keterangan koneksi pin akselerometer MMA7361 dengan Arduino nano

MMA7361	Pin Arduino nano
X	A0
Y	A1
Z	A2
SL	6
ST	5
0G	4
GS	7
5v	5v
3v3	3V3
	Aref
GND	GND

5.1.1.2 Perancangan Rangkaian Vibration 801S



Gambar 5. 3 Rancangan Sensor 801s Pada Arduino

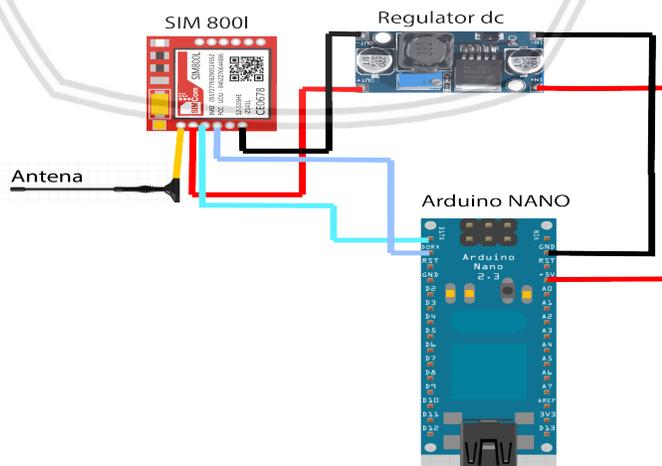
Pada gambar diatas merupakan hasil rancangan menghubungkan pin dari sensor getaran 801S dengan ppin Arduino nano, yang di hasilkan dari sensor akan di masukkan kedalam signal digital Arduino nano.

Pada Tabel 5.3 akan menunjukan pin antar sensor 801s vibration dengan Arduino nano secara terperinci. Di sensor 801s vibration ini menggunakan 3 pin yaitu 5v yang di hubungan dengan 5v di Arduino nano begitupun pin GND di sensor 801s vibration dihubungan GND di Arduino nano, pada sensor nilai *output* dihubungan dengan pin 8 di Arduino nano, berikut adalah tabel terperinci.

Tabel 5. 2 Keterangan koneksi pin 801s vibration dengan Arduino nano

801s	Pin Arduino nano
5v	5v
GND	GND
Dout	8

5.1.1.3 Perancangan Rangkaian SIM 800I



Gambar 5. 4 Rancangan SIM 800I Dengan Arduino

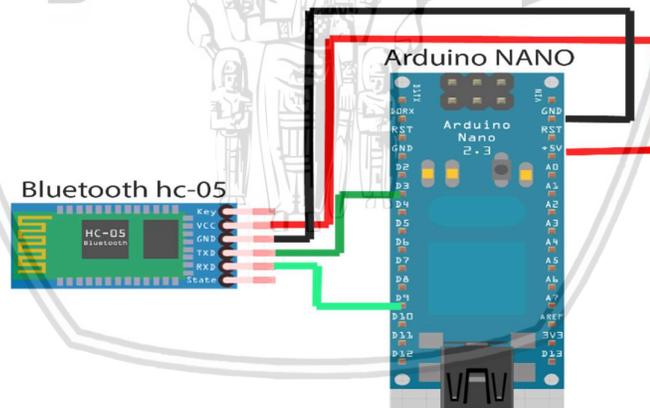
Pada gambar perancangan 5.4 diatas terdapat beberapa koneksi antar pin dengan aduino antara *SIM 800I* dengan Arduino nano yaitu RX, TX, Vcc dan ground, dimana tegangan yang dapat diterapkan module *SIM 800I* kisaran 3,8 volt hingga 4,2 volt, jika tegangan tidak didalam range tersebut module tidak dapat bekerja, untuk mensiasati tegangan yang masuk ke modul menggunakan regulator dc . Rangkaian *SIM 800I* dengan Arduino akan dijelaskan koneksi antar pin pada Tabel 5.1.

Tabel 5. 3 Keterangan koneksi pin *SIM 800I* dengan Arduino nano

Pin <i>SIM 800I</i>	Pin Arduino nano
VCC	5V
GND	GND
RX	2
TX	3

Bedasarkan tabel 5.3 terlihat dari 4 pin yang di sambungkan dari modul *SIM800I* ke arduino nano yaitu VCC, GND, RX, dan TX. Untuk VCC di hubungkan dengan pin 5V di Arduino nano. Sedangkan untuk GND dihubungkan ke pin GND di Arduino nano. Pin TX di *SIM 800I* untuk mentransmisikan data yang di yang di sambungkan dengan pin 3 di Arduino nano, sedangkan pin RX di modul *SIM 800I* untuk menerima data di hubungkan dengan in 2 di Arduino nano.

5.1.1.4 Perancangan Rangkaian *Bluetooth*



Gambar 5. 5 Rancangan *Bluetooth* Dengan Arduino

Gambar 5.5 menunjukkan hubungan pin Bluetooth dengan Arduino nano dimana pin output RX dan TX di terapkan pada pin digital Arduino nano.

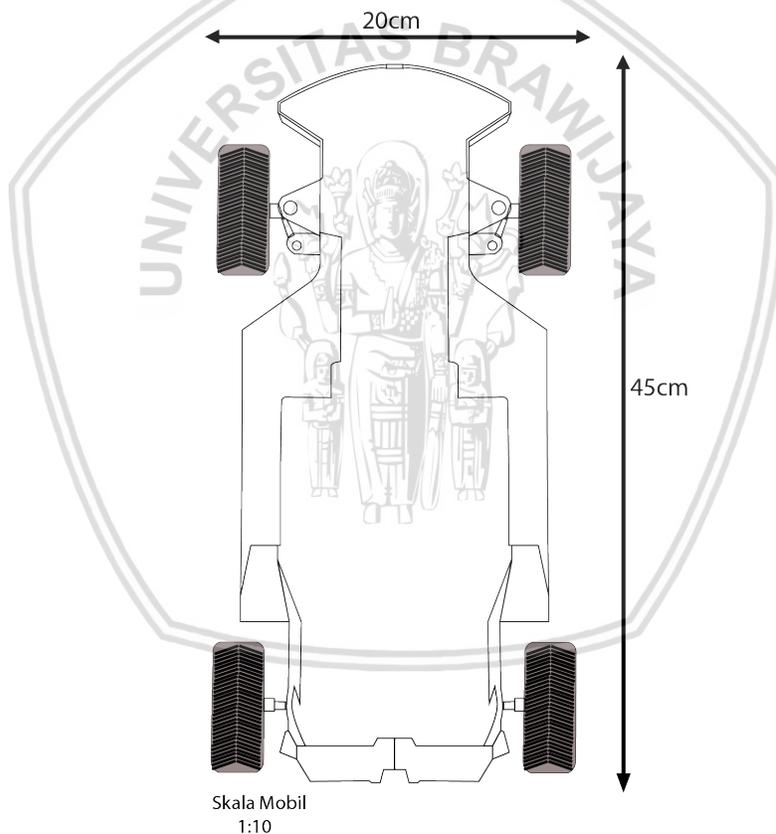
Pada Tabel 5.4 akan menunjukan pin antar *Bluetooth* dengan Arduino nano secara terperinci. modul *Bluetooth* terdapan 4 pin dimana dua pin untuk menerima dan mengirim data yaitu pin TX dan RX yang di hubungan cecara berurutan dengan pin Arduino nano yaitu pin 3 dan pin 9. Untuk dua pin yaitu 5v dan GND yang saling berhungan di pin Arduino nano.

Tabel 5. 4 Keterangan koneksi pin *Bluetooth* dengan *Arduino nano*

<i>Bluetooth</i>	Pin <i>Arduino nano</i>
5v	5v
GND	GND
tx	10
rx	9

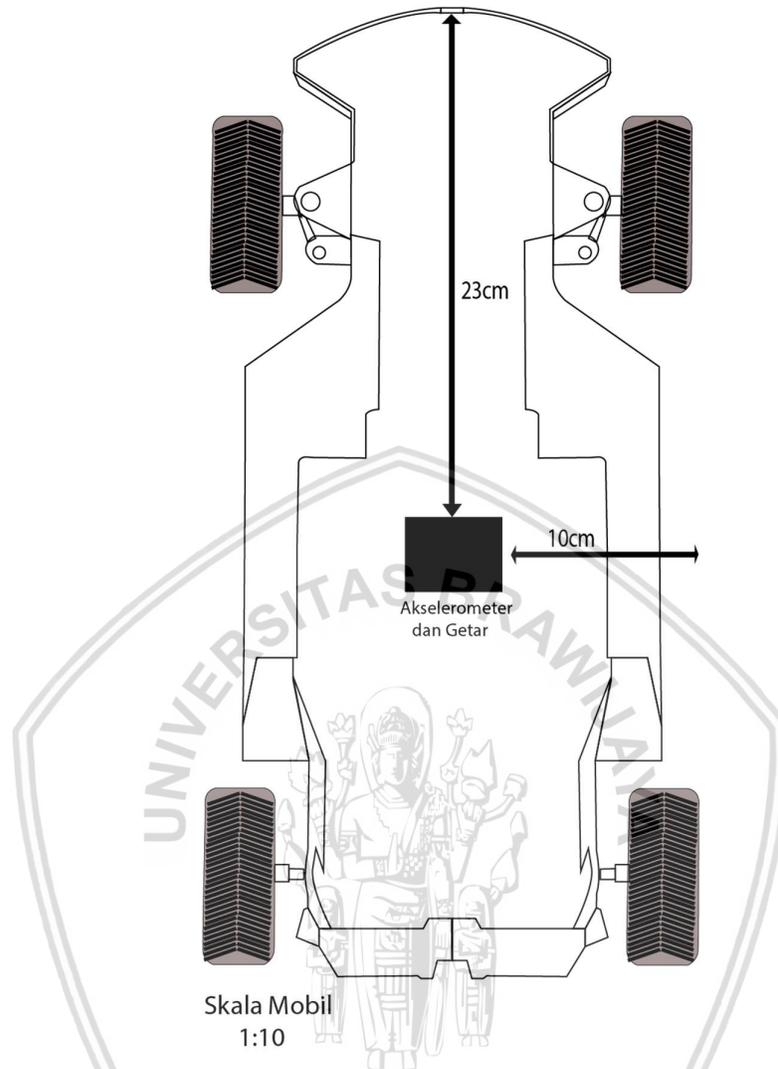
5.1.2 Perancangan Prototipe Alat Pendeteksi Tabrakan

Perangkat pendeteksi tabrakan ini menggunakan mobil remote control dimana memiliki kecepatan maksimal adalah 20km/h yang tertera pada buku manual, skala mobil yaitu 1:10, untuk penjelasan lebih lanjut tertera pada gambar 5.6.



Gambar 5. 6 Ukuran Mobil *Remote Control*

Perangkat pendeteksi tabrakan ini di tempatkan dalam mobil penempatan ada di titik tengah dari mobil *remote control*, penempatan di tengah mobil bertujuan untuk memberikan tingkat kurasi dari sensor akselerometer menjadi lebih baik, karna nilai pengambilan dari nilai X dan Y mobil *remote control* tersebut. Pada **gambar 5.7** adalah penjelasan penempatan sensor.



Gambar 5. 7 Letak Sensor

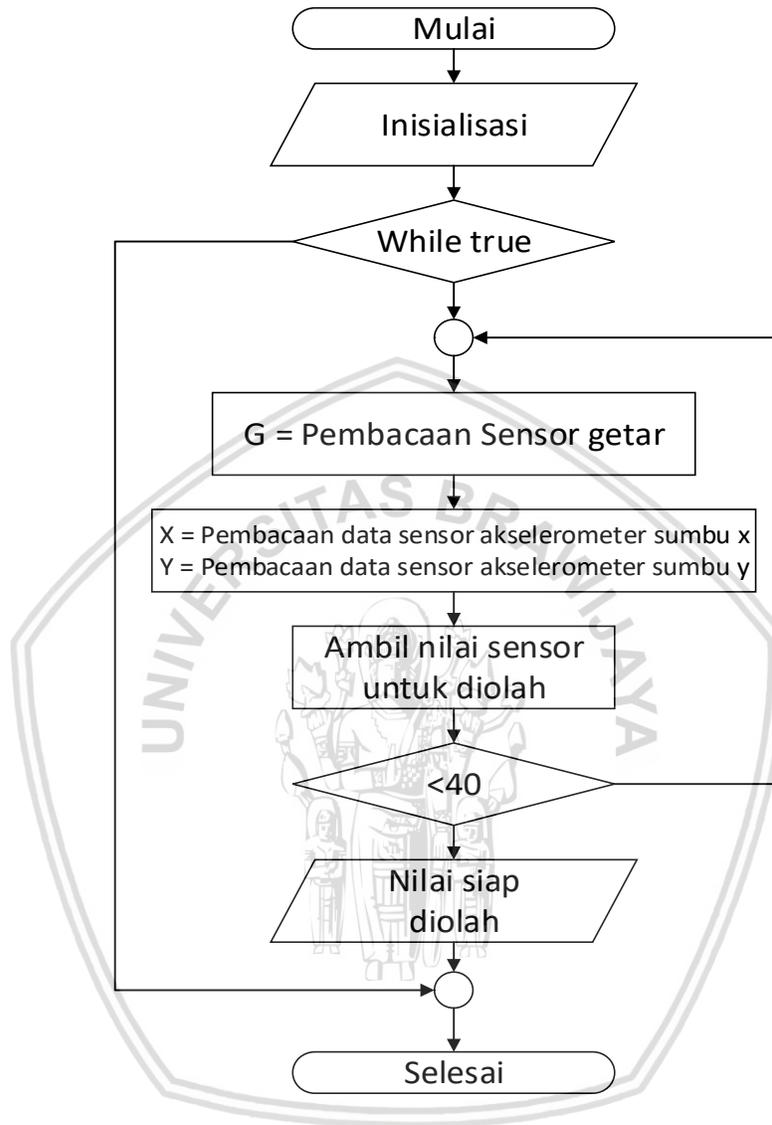
Pada gambar di atas di jelaskan penempatan akselerometer dan sensor getar di hitunga dari depan mobil sejauh 23cm dan di hitunga dari kanan kiri mobil 10cm penempatan itu bertujuan mencari posisi tengah mobil yang paling mempunyai nilai imbang, karna jika ditempatkan paling depan maka nilai yang di dapat semakin besar begitu sebaliknya.

Perangkat pendeteksi tabrakan ini terdiri dari Arduino nano, sensor akselerometer, vibration 801s, SIM 800I, *stepdown dc to dc*, dan *Bluetooth*. Penempatan kedua sensor berdampingan di maksudkan agar sensor getar membaca getaran di titik yang sama dari sensor akselerometer.

5.1.3 Perancangan Perangkat Lunak

Didalam subab ini menjelaskan tetang dua pembahasan , yaitu perancangan perangkat lunak pada mikrokontroler untuk engambilan data latih, dan data uji. Lalu perancangan klasifikasi naïve bayes dimana dilakukanya perhitungan dari setiap tahapan yaitu perhitungan prior, perhitungan gaussian dan perhitungan posterior.

5.1.3.1 Perancangan Pengambilan Data Latih

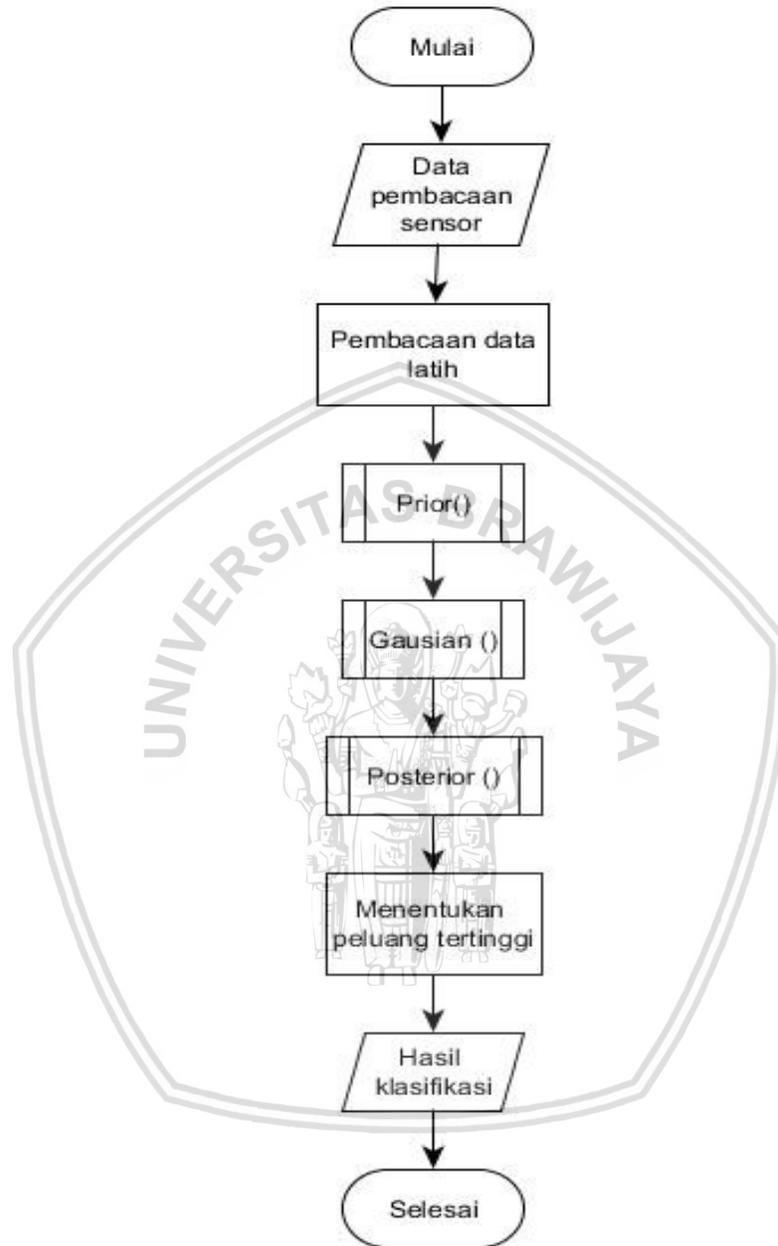


Gambar 5. 8 Diagram alir perancangan pengambilan data latih

Proses pengambilan data sensor dari dua sensor yaitu sensor akselerometer dan sensor getaran di jelaskan pada Gambar 5.8, pengambilan data yang di maksudkan untuk menentukan nilai klasifikasi pada metode naïve bayes.

Tahapan pengambilan data latih yaitu awal semua sensor harus di inisialisasi sesuai perancangan yang telah di buat terlebih dahulu, lalu sensor getar dan akselero apakan bisa membaca hasil dari perubahan perubahan, jika bisa ambil data dari nilai sensor, pada tahap ini hasil data latih akan disimpan dan di ambil nilai amplitude pada setiap kondisi yaitu kondisi berjalan, berhenti mendadak dan tabrakan.

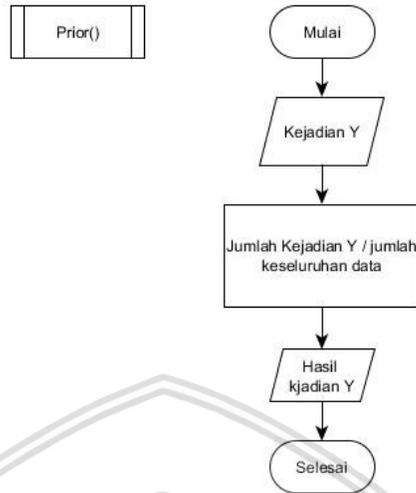
5.1.3.2 Perancangan Klasifikasi Naïve Bayes



Gambar 5. 9 Diagram alir perancangan pengambilan data Uji

Proses klasifikasi menggunakan metode naïve bayes dijabarkan pada Gambar 5.9 proses pertama pembacaan sensor yang dimana nilai pembacaan dapat mempengaruhi klasifikasi nilai akhirnya, data klasifikasi di pengaruhi oleh data latih yang telah dibuat. Tahap perhitungan dimulai dengan menentukan nilai fungsi dari data prior(), nilai fungsi Gaussian(), dan menentukan nilai fungsi Posterior(), adapun penjelasan dari fungsi fungsi tersebut sebagai berikut.

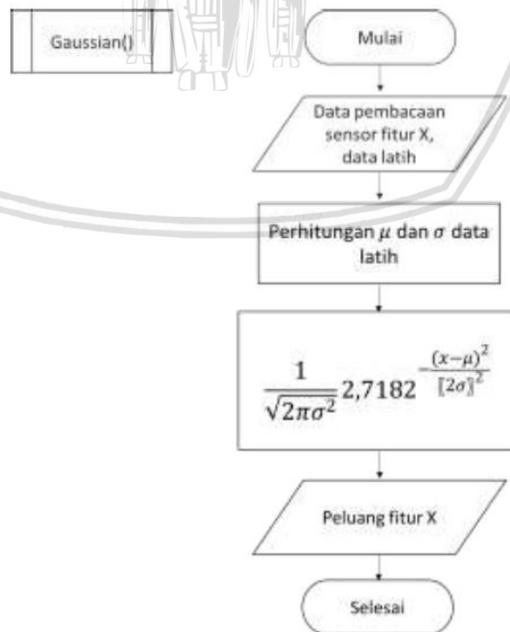
A. Tahap perhitungan nilai prior.



Gambar 5. 10 Diagram alir fungsi prior()

Tahap pertama pada fungsi prior yang dijelaskan pada Gambar 5.10 yaitu menentukan nilai klasifikasi keadaan dengan metode naïve bayes adalah menghitung nilai prior dari masing masing kelas keadaan kendaraan. Nilai prior adalah nilai peluang terjadi suatu kelas dengan cara membagi banyak data dalam satu kelas di dalam sistem ini di bagi menjadi tiga kelas yaitu mobil berjalan, mobil berhenti mendadak dan mobil kecelakaan. Data perhitungan nilai prior di masukan menjadi data latih.

B. Tahap perhitungan nilai gaussian.

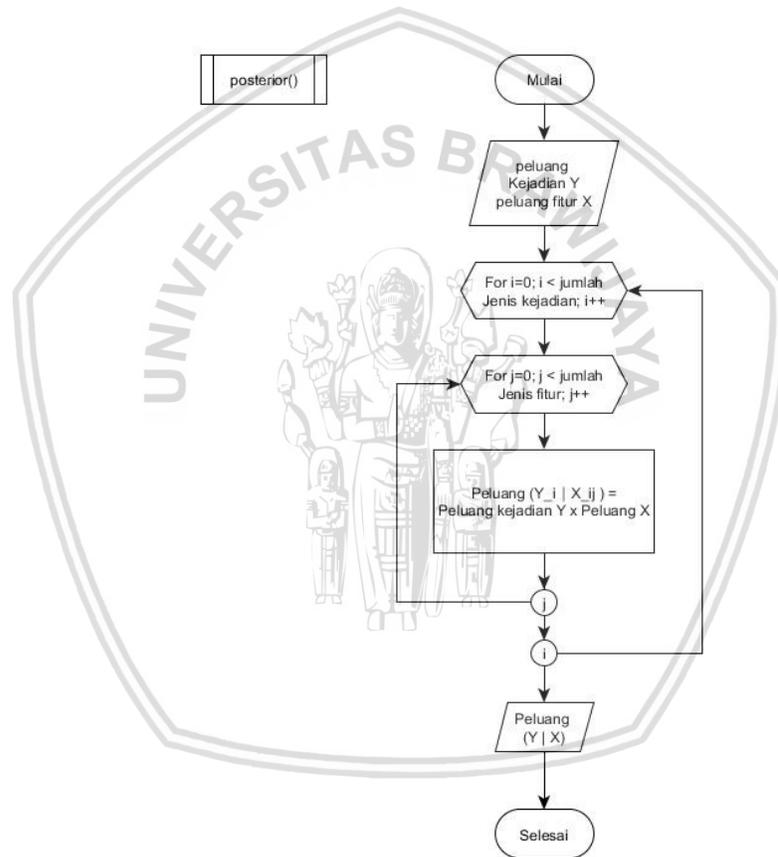


Gambar 5. 11 Diagram alir fungsi Gaussian()



Tahap kedua pada fungsi gaussian dijelaskan pada Gambar 5.11 yaitu ada tahap ini menentukan nilai dari masing masing fitur pada sistem ini menggunakan 3 fitur yaitu 2 fitur dari pembacaan sensor MMA7361 akselerometer diambil nilai dari sumbu X dan nilai dari sumbu Y. serta 1 fitur dari nilai sensor getar 801s. Namun sebelum dapat menentukan nilai peluang dari masing-masing fitur, terlebih dahulu harus dilakukan perhitungan mean dan standar deviasi dari data latih menggunakan Persamaan (2.4) dan Persamaan (2.5). Pada sistem ini, data latih yang berupa nilai mean dan standar deviasi disimpan pada pemrograman mikrokontroler untuk mepermudah dalam mengakses nilai dari data latih saat sistem dijalankan.

C. Tahap perhitungan nilai posterior.



Gambar 5. 12 Diagram alir fungsi posterior()

Pada Gambar 5.12 menjelaskan tahapan dimana untuk menentukan besar peluang dari masing-masing kelas yang akan di hitung ketika terjadi ketika adanya masukan dari setiap fitur. Sistem ini membaca keadaan mobil yang sedang berjalan dengan pembacaan fitur X, fitur Y, dan fitur G. dengan proses melakukan perkalian antar fungsi prior() dengan fungsi gaussian(). Tahapan ini adalah tahapan terakhir dari pengklasifikasian naïve bayes. dengan cara membandingkan nilai tertinggi posterior.



5.1.3.3 Perhitungan Manual Naïve Bayes

Contoh perhitungan manual untuk melakukan klasifikasi mobil kecelakaan atau tidak dengan data uji yang mempunyai fitur $X=4,69$, $Y=2,96$, dan $G=8326$ berdasarkan data latih yang terdapat pada lampiran adalah sebagai berikut:

1. Menghitung peluang prior dari masing-masing jenis kondisi

$$P_{Berjalan} = \frac{\text{Jumlah data kondisi mobil bejalan}}{\text{Jumlah seluruh data}} = \frac{10}{40} = 0.25$$

$$P_{Rem} = \frac{\text{Jumlah data kondisi Berhenti mendadak}}{\text{Jumlah seluruh data}} = \frac{14}{40} = 0.35$$

$$P_{Kecelakaan} = \frac{\text{Jumlah data kondisi mobil kecelakaan}}{\text{Jumlah seluruh data}} = \frac{16}{40} = 0.4$$

2. Menghitung nilai *mean* dan standar deviasi dari masing-masing jenis kondisi

A. Perhitungan Mean

$$\begin{aligned} \bar{x}_{X(\text{berjalan})} &= \frac{\sum_{i=1}^n Y_i}{\text{Jumlah seluruh data kondisi berjalan}} \\ &= \frac{2.69 + 2.22 + 4.56 + 2.13 + 2.68 + \dots + 3.29}{10} \\ &= 2.842 \end{aligned}$$

$$\begin{aligned} \bar{x}_{Y(\text{berjalan})} &= \frac{\sum_{i=1}^n G_i}{\text{Jumlah seluruh data kondisi berjalan}} \\ &= \frac{2.28 + 3.16 + 5.72 + 3.37 + 2.99 + \dots + 2.62}{10} \\ &= 2.884 \end{aligned}$$

$$\begin{aligned} \bar{x}_{G(\text{berjalan})} &= \frac{\sum_{i=1}^n X_i}{\text{Jumlah seluruh data kondisi berjalan}} \\ &= \frac{110009 + 13757 + 9035 + 4527 + \dots + 5083}{10} \\ &= 9107.3 \end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari mean di kondisi berhenti mendadak dan kondisi kecelakaan. Hasil perhitungan mean keseluruhan kondisi ditunjukkan pada Tabel 5.5

Tabel 5. 5 Mean tiap kondisi

	X	Y	G
Berjalan	2.842	2.884	9107.3
Rem	3.154	3.140	9929.4
Kecelakaan	11.28188	8.33625	16015.44

B. Perhitungan Standar Deviasi

$$\begin{aligned} \sigma_{x(berjalan)} &= \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{x}_x(berjalan))^2}{\text{jumlah data kondisi berjalan}}} \\ &= \sqrt{\frac{(2.69 - 2.84)^2 + (2.22 - 2.84)^2 + (4.56 - 2.84)^2 + \dots + (3.29 - 2.84)^2}{10 - 1}} \\ &= 0.757228866034276 \\ \sigma_{y(berjalan)} &= \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{x}_y(berjalan))^2}{\text{jumlah data kondisi berjalan}}} \\ &= \sqrt{\frac{(2.28 - 2.88)^2 + (3.16 - 2.88)^2 + (5.72 - 2.88)^2 + \dots + (2.62 - 2.88)^2}{10 - 1}} \\ &= 1.18333990608503 \\ \sigma_{g(berjalan)} &= \sqrt{\frac{\sum_{i=1}^n (g_i - \bar{x}_g(berjalan))^2}{\text{jumlah data kondisi berjalan}}} \\ &= \sqrt{\frac{(110009 - 9107.3)^2 + (13757 - 9107.3)^2 + \dots + (5083 - 2.88)^2}{10 - 1}} \\ &= 5632.69615824528 \end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari standar deviasi di kondisi berhenti mendadak dan kondisi kecelakaan. Hasil perhitungan standar deviasi keseluruhan kondisi ditunjukkan pada Tabel 5.6

Tabel 5. 6 standar deviasi tiap kondisi

	X	Y	G
Berjalan	0.7572	1.1833	5632.6961
Rem	1.009287	1.575444	4748.5516
Kecelakaan	9.40358	6.739873	11220.3321

C. Perhitungan Nilai Gausisan

Menghitung nilai gaussian dari masing masing fitur yang di ujikan yaitu di masukan nilai fitur yang masing masing memiliki nilai (X=4.69, Y=2.96, dan G=8326).

$$\begin{aligned} P(X=4.69 | berjalan) &= \frac{1}{\sqrt{2\pi\sigma_{X(berjalan)}^2}} e^{-\frac{(4.69 - \mu_{X(berjalan)})^2}{2\sigma_{X(berjalan)}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times 0.7572^2}} e^{-\frac{(4.69 - 2.842)^2}{2(0.7572)^2}} \\ &= 0.0266438 \end{aligned}$$



$$\begin{aligned}
 P(X=4.69|rem) &= \frac{1}{\sqrt{2\pi \sigma_{X(rem)}^2}} e^{-\frac{(4.69-\mu_{X(rem)})^2}{2\sigma_{X(rem)}^2}} \\
 &= \frac{1}{\sqrt{2 \times 3,14 \times 1.009287^2}} e^{-\frac{(4.69-3.154)^2}{2(1.009287)^2}} \\
 &= 0.123439
 \end{aligned}$$

$$\begin{aligned}
 P(X=4.69|kecelakaan) &= \frac{1}{\sqrt{2\pi \sigma_{X(kecelakaan)}^2}} e^{-\frac{(4.69-\mu_{X(kecelakaan)})^2}{2\sigma_{X(kecelakaan)}^2}} \\
 &= \frac{1}{\sqrt{2 \times 3,14 \times 9.40358^2}} e^{-\frac{(4.69-11.28188)^2}{2(9.40358)^2}} \\
 &= 0.0331957
 \end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari nilai gaussian di kondisi berhenti mendadak dan kondisi kecelakaan. Hasil perhitungan standar deviasi keseluruhan kondisi ditunjukkan pada Tabel 5.7

Tabel 5.7 nilai gaussian tiap kondisi

	X	Y	G
Berjalan	0.0266438	0.336534	0.0000701
Rem	0.123439	0.251642	0.0000793
Kecelakaan	0.0331957	0.0431043	0.0000281

D. Perhitungan Nilai Posterior.

Menghitung nilai peluang *posterior* tiap kelas dari contoh data yang telah di hitung di atas.

$P(\text{berjalan} | X=4,69, Y=2,96, G=8326)$

$$\begin{aligned}
 &= P_{\text{berjalan}} \times P(X = 4,69|\text{berjalan}) \times P(Y = 2,96|\text{berjalan}) \times P(G = 8326|\text{berjalan}) \\
 &= 0.25 \times 0.0266438 \times 0.336534 \times 0.0000701 \\
 &= 0.000000157
 \end{aligned}$$

$P(\text{rem} | X=4,69, Y=2,96, G=8326)$

$$\begin{aligned}
 &= P_{\text{rem}} \times P(X = 4,69|\text{rem}) \times P(Y = 2,96|\text{rem}) \times P(G = 8326|\text{rem}) \\
 &= 0.35 \times 0.123439 \times 0.251642 \times 0.0000793 \\
 &= 0.000000862
 \end{aligned}$$



$$P(\text{kecelakaan} | X=4,69, Y=2,96, G=8326)$$

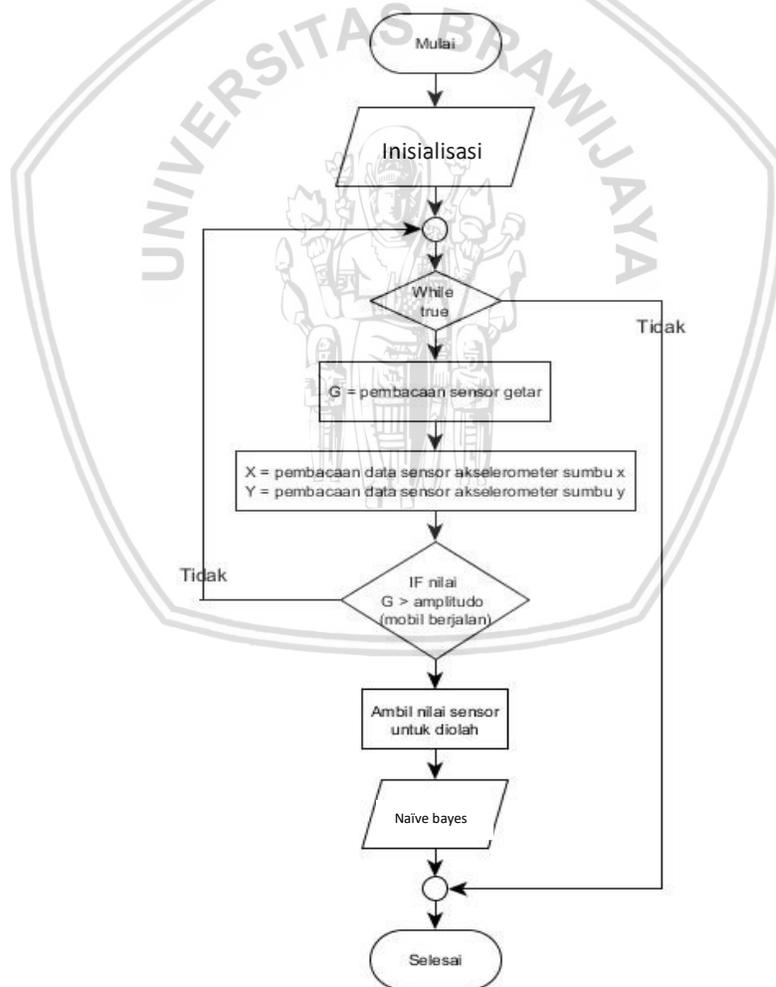
$$= P_{\text{kecelakaan}} \times P(X = 4,69 | \text{kecelakaan}) \times P(Y = 2,96 | \text{kecelakaan}) \times P(G = 8326 | \text{kecelakaan})$$

$$= 0.4 \times 0.0331957 \times 0.0431043 \times 0.0000281$$

$$= 0.000000016$$

Berdasarkan perhitungan yang dilakukan untuk mencari nilai peluang posterior, di dapat dari perkalian antar peluang prior dan nilai *gaussian* dari tiap tiap fitur yang di ujikan, nilai data yang di ujikan yaitu X=4.69, Y=2.96, dan G=8326 di simpulkan dari data tersebut mempunyai posterior tertinggi pada kondisi rem mendadak dan dengan demikian hasil perhitungan di atas menunjukan jenis keadaan rem mendadak.

5.1.3.4 Perancangan Pengambilan data uji



Gambar 5. 13 Diagram alir perancangan pengambilan data Uji

Pada Gambar 5.13 menjelaskan pengambilan nilai data uji dilakukan setelah pengambilan data latih, pengambilan data uji menggunakan nilai atau amplitude tertinggi dari sensor getar. Yang akan di jelaskan sebagai berikut:

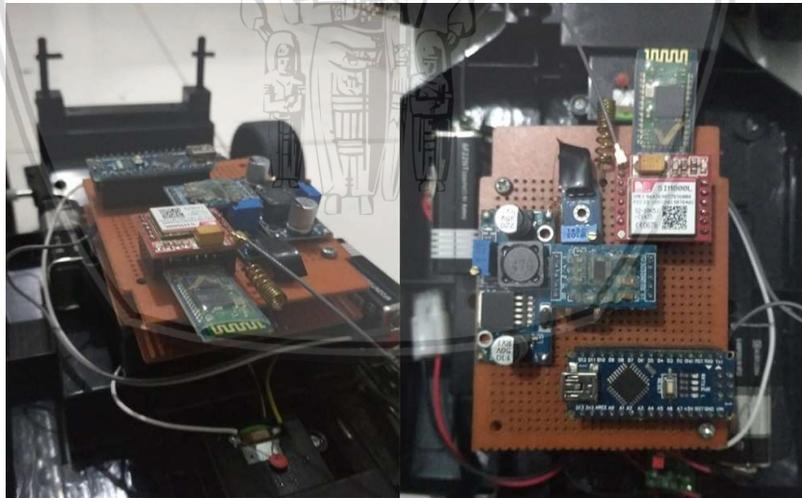
1. Inialisasi setiap sensor untuk mengetahui apakah sensor berjalan dengan baik.
2. Jika iya maka sensor akan membaca kondisi dari sensor yang ada.
3. Baca sensor getaran.
4. Baca sensor akselerometer.
5. Jika nilai getaran lebih dari amplitude berjalan maka nilai akan di masukan untuk di hitung.
6. Pengambilan nilai sensor untuk di hitung.
7. Perhitungan nilai klasifikasi naïve bayes.

5.2 Implementasi Sistem

Implementasi sistem adalah tahapan dimana merealisasikan sistem berdasarkan perancangan yang telah dilakukan sebelumnya. subab ini menjelaskan secara terperinci terkait implementasi prototipe. Implementasi perangkat keras dan implementasi perangkat lunak.

5.2.1 Implementasi perangkat keras

Implementasi perangkat keras mencakup beberapa komponen elektronika antara lain Arduino nano, sensor akselerometer, sensor 801s, sim 800l, dan regulator dc yang di hubungan menggunakan PCB .



Gambar 5. 14 Implementasi rangkaian sensor MMA 7361 dan 801s dengan Arduino nano

Pada gambar 5.14 menunjukkan hasil rangkain implementasi dimana sensor MMA7361 dan sensor 801s di rangkai sesuai perancangan dan di masukkan kedalam mobil remote kontrol diletakkan pada titik tengah mobil. Dihubungkan melalui kabel dan dirangkai di atas PCB. Dan untuk modul SIM800l di hubungan dengan regulator DC dikarenakan SIM 800l membutuhkan tegangan antara 3.6v-4.5v.

5.2.2 Implementasi prototipe alat pendeteksi kecelakaan

Implementasi prototipe alat pendeteksi kecelakaan yang mengacu pada penggunaan mobil remote control. Dengan skala 1:10 dan dengan kecepatan maksimal 20 km/jam yang akan di letakkan rangkaian pendeteksi kecelakaan. Hasil implemementasi prototipe beserta peletakan komponen elektronik yang digunakan di tunjukan pada Gambar 5.15 sebagai berikut.



Gambar 5. 15 Implementasi prototipe alat pendeteksi kecelakaan

5.2.3 Imlementasi perangkat lunak

Implementasi perangkat lunak menjelaskan realisasi kode program menggunakan Bahasa C yang di terapkan pada software Arduino IDE 1.8.0 dengan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam proses pemrograman Arduino nano terdapat beberapa *library* yang di gunakan untuk mempermudah implementasi pada fungsi tertentu, di antaranya yaitu *library* “SoftwareSerial.h” digunakan untuk memberikan status serial pada pindigital Arduino. “AcceleroMMA7361.h” di gunakan untuk mengambil nilai status perubahan nilai dari sensor akselerometer, “stdio.h” diugnakan untuk standar oprasi input-output dan *library* “math.h” digunakan untuk perhitungan matematika yang rumit dan kompleks.

Tabel 5. 8 Kode program inialisasi *Library* sistem pendeteksi tabrakan

Baris	Kode Program
1	#include <SoftwareSerial.h>
2	#include <AcceleroMMA7361.h>
3	#include <stdio.h>
4	#include <math.h>

5.2.3.1 Implementasi Kode Pengambilan Data latihan

Implementasi ini dilakukan untuk mendapatkan nilai data latihan, untuk pengambilan data suatu sensor perlu dilakukan inisialisasi variable dan konfigurasi pin berdasarkan perancangan perangkat keras dan perangkat lunak yang di terangkan pada subbab sebelumnya, Table 5.9 pada baris ke-1 di tetapkan nilai konstanta untuk pengambilan nilai sensor accelerometer untuk kalibrasi nilai sensor, baris ke-3 dan ke-4 adalah variable untuk menaplikasikan nilai dari sensor kaselerometer, untuk baris ke5 sampai baris ke-10 adalah variable perhitungan nilai noise awal di pambnaan sensor akselerometer. Pada baris ke-11 hingga baris ke-17 variable untuk sensor getar.

Tabel 5. 9 Kode program Inisialisasi variable pembacaan sensor

Baris	Kode Program
1	#define samples 10
2	AcceleroMMA7361 accelero;
3	float x;
4	float y;
5	float xsample = 0;
6	float ysample = 0;
7	float value1;
8	float value2;
9	float xvalue;
10	float yvalue;
11	int EP1 = 8;
12	long TP_initia() {
13	delay(10);
14	long measurementa = pulseIn (EP1, HIGH);
15	return measurementa ;
16	}
17	}

Implementasi kode program untuk pengambilan data latihan untuk mendapatkan nilai acuan untuk dimasukan kedalam standar metode naïve bayes, yang ditunjukan pada Table 5.10 pada baris ke-2 hingga baris ke-4 merupan variable yang mengambil nilai dari sensor akselerometer dan sensor getar, baris ke-6 dan ke-7 variable yang mendapatkan nilai akselerometer dari pengurangan data noise dikurang data sensor, untuk pembagian dengan 32 di tetapkan oleh data *sheeat MMA 7361* untuk pembacaan sensor di range 6g. baris ke-9 hingga baris ke-15 merubah nilai yang di dapat dari sensor akselerometer menjadi nilai positif karna membutuhkan data tertinggi dari pengambilan data latihnya. Dan baris ke-16 hingga baris ke-27 perintah untuk melihat hasil pembacaan, dari sensor getar yaitu (meausurmena) dan sensor akselerometer (xvalue) untuk pembacaan percepatan pada sumbu X serta (yvalue) untuk pembacaan percepatan pada sumbu Y.

Tabel 5. 10 Kode program pengambilan data

Baris	Kode Program
1	<code>void loop() {</code>
2	<code> int measurementa = TP_init();</code>
3	<code> value1 = accelero.getXAccel();</code>
4	<code> value2 = accelero.getYAccel();</code>
5	
6	<code> xvalue = (xsample - value1) / 32;</code>
7	<code> yvalue = (ysample - value2) / 32;</code>
8	
9	<code> if (xvalue < 0) {</code>
10	<code> xvalue = -(xvalue);</code>
11	<code> } else {}</code>
12	
13	<code> if (yvalue < 0) {</code>
14	<code> yvalue = -(yvalue);</code>
15	<code> } else {}</code>
16	<code> Serial.print(" ");</code>
17	<code> Serial.print(measurementa);</code>
18	<code> Serial.print(" ");</code>
19	<code> Serial.print(" ");</code>
20	<code> Serial.print(" ");</code>
21	<code> Serial.print(xvalue);</code>
22	<code> Serial.print(" ");</code>
23	<code> Serial.print(" ");</code>
24	<code> Serial.print(" ");</code>
25	<code> Serial.print(yvalue);</code>
26	<code> Serial.print(" ");</code>
27	<code> Serial.println(" ");</code>
	<code>}</code>

5.2.3.1 Implementasi Kode Program Perancangan Deteksi Tabrakan

Implementasi kode program perancangan dibutuhkan inialisasi variabel dan konfigurasi pin sensor berdasarkan perancangan perangkat keras dan perancangan perangkat lunak yang telah jelaskan pada subbab sebelumnya. Tabel 5.11 menunjukkan pada baris ke-1 hingga baris ke-10 variable untuk mendapatkan nilai dari sensor akselerometer, baris ke-12 dan baris ke-13 variable untuk mencari nilai terbesar dari sensor akselerometer. Pada baris ke-15 inialisasi variable di pin digital 8 Arduino untuk sensor getar. Baris ke-17 untuk inialisasi pin *Bluetooth* pada pin-10 dan pin-9, baris ke-18 untuk inialisasi pin *SIM800L* pada pin-2 dan pin-3. Dan baris ke-21 variable array masukan untuk data pembacaan sensor yang nantinya akan di olah menggunakan metode *naïve bayes*.



Tabel 5. 11 Kode program inialisasi variable pembacaan sensor

Baris	Kode Program
1	<code>float x;</code>
2	<code>float y;</code>
3	<code>float xsample = 0;</code>
4	<code>float ysample = 0;</code>
5	
6	<code>float value1;</code>
7	<code>float value2;</code>
8	
9	<code>float xvalue;</code>
10	<code>float yvalue;</code>
11	
12	<code>float x_max = 0;</code>
13	<code>float y_max = 0;</code>
14	
15	<code>int EP1 = 8; //getar pin</code>
16	
17	<code>SoftwareSerial bluetooth(10, 9); // TX RX</code>
18	<code>SoftwareSerial SIM800L(2, 3); // RX TX</code>
19	
20	
21	<code>float masukan [3];</code>

Implementasi kode program terhadap sensor akselerometer di deklasikan pada fungsi void setup(), yang di tunjukan pada Tabel 5.12. Pada baris ke-5 merupakan inialisasi pin dari sensor akselerometer yang di rincikan pada Tabel 5.1 , pada baris ke-6 merupakan penetapan inialisasi input tegangan akselerometer sebesar 3.3v, baris ke-7 inialisasi sensitivitas ke low/ rendah. Dan baris ke-12 hingga baris ke-19 perulangan nilai yang didapat dari sneosr akselerometer sebanyak sepuluh kali lalu di bagi sepuluh untk mendapatkan nilai standar untuk mendapatkan nilai kalibrasi dan nilai yang didapat cukup akurat.

Tabel 5. 12 Kode program Inialisasi sensor akselesometer MMA7361

Baris	Kode Program
1	<code>void setup() {</code>
2	
3	<code>.....</code>
4	
5	<code>accelero.begin(6, 5, 4, 7, A0, A1, A2);</code>
6	<code>accelero.setARefVoltage(3.3);</code>
7	<code>accelero.setSensitivity(LOW);</code>
8	<code>accelero.calibrate();</code>
9	
10	<code>.....</code>
11	
12	<code>for (int i = 0; i < samples; i++)</code>
13	<code>{</code>

```

14     xsample += accelero.getXAccel();
15     ysample += accelero.getYAccel();
16     }
17
18     xsample /= samples;
19     ysample /= samples;
20
21     Serial.println(xsample);
22     Serial.println(ysample);

```

Untuk implementasi pembacaan sensor getar di dirincikan pada table 5.13, pada baris ke-1 variable TP_inita() fungsi yang di buat untuk nilai sensor getar. Baris ke-2 delay program, baris ke-3 variable measurmena mendapatkan nilai dari pin sensor getar, dan baris ke-4 pengambalian nilai measurmena.

Tabel 5. 13 Kode program inialisasi sensor getar 801s

Baris	Kode Program
1	long TP_inita() {
2	delay(10);
3	long measurementa = pulseIn (EP1, HIGH);
4	return measurementa ;
5	}

Pada table 5.14 menunjukan implementasi proses pengambilan data sensor untuk di olah dengan adanya *trigger* nilai tertinggi sensor getar yag telah di uji coba, tertera pada baris ke-4 hingga baris ke-16, nilai di simpan pada array masukkan(). Sedangkan apabila tidak ada triger maka program terus berulang.

Tabel 5. 14 Kode program untuk mengecek kondisi

Baris	Kode Program
1	int measurementa = TP_inita();
2	value1 = accelero.getXAccel();
3	value2 = accelero.getYAccel();
4	if (measurementa > 5000) {
5	masukkan[0] = measurementa;
6	masukkan[1] = x_max;
7	masukkan[2] = y_max;
8	Serial.print("Prediksi keadaan =");
9	nBayes(masukkan);
10	if(x_max y_max>4){
11	Serial.print("Prediksi tanpa metode");
12	Serial.print(x_max);
13	Serial.println("kecelakaan");
14	}else{}
15	delay(1000);
16	} else {}
17	}



5.2.3.2 Implementasi Kode Program *Naïve Bayes*

Tahap implementasi *Naïve Bayes* di maksudkan untuk merealisasikan program untuk pengambilan keputusan jenis kondisi dari nilai akselerasi dan tingkat besar getaran yang akan di masukkan ke kondisi kecelakaan atau tidak terdeteksi kecelaakaan.

Tabel 5. 15 Kode program inialisasi variable *Naïve Bayes*

Baris	Kode Program
1	float gaussian[3][3]; //array hasil hitung gaussian
2	float hasil[3]; //array hasil hitung possterior
3	float hasilMeanStd[3][3];
4	float tertinggi;
5	int index = 0;
6	int gauske = 0;

Berdasarkan Tabel 5.15 menunjukan proses inialisasi variable dan tipe data yang digunakan untuk memprogram metode *naïve bayes*. Kode diatas merupakan variable global yang di fungsikan untuk variable gaussian, mean, hasil, nilai tertinggi, index (penanda), dan gauske, sehingga fungsi diatas dapat digunakan pada seluruh program deteksi kecelakaan.

Tabel 5. 16 Kode program fungsi nBayes()

Baris	Kode Program
1	void nBayes (float x[3]) {
2	probPrior();
3	meanStd(ada_berjalan, latih_berjalan , 0);
4	meanStd(ada_rem, latih_rem, 1);
5	meanStd(ada_kecelakaan, latih_kecelakaan, 2);
6	gaussian(x, ada_k_berjalan);
7	gaussian(x, ada_k_rem);
8	gaussian(x, ada_k_kecelakaan);
9	probPosterior(pro_ber, 0);
10	probPosterior(pro_rem, 1);
11	probPosterior(pro_kec, 2);
12	kesimpulan();
13	gauske = 0;
14	}

Berdasarkan Tabel 5.16 menunjukan implementasi tahap eksekusi fungsi nBayes() yang di baris ke-2 pemanggilan fungsi probPrior(), baris ke-3 hingga baris ke-5 memanggil fungsi meanStd () dengan nilai bayak data, data latih dan nilai kondisi untuk di proses, baris ke-6 hingga baris ke-8 memanggil fungsi gaussian() dengan nilai pembacaan sensor, dan nilai rata-rata untuk di proses di dalamnya, pada baris ke-9 sampai baris ke-11 adalah pemanggilan fungsi probPosterior() dengan nilai bayak kejadian yanf di kalikan dengan perhitungan Gaussian, baris ke-12 pemanggilan fungsi kesimpulan(), dan baris ke-13 pengembalian nilai gauske ke nilai 0.

Tabel 5. 17 Kode program Inisialisasi variable pembacaan sensor

Baris	Kode Program
1	<code>float ada_berjalan = 10;</code>
2	<code>float ada_rem = 14;</code>
3	<code>float ada_kecelakaan = 16;</code>
4	<code>float pro_ber, pro_rem, pro_kec;</code>
5	<code>float ada_k_berjalan [3][3]; //array hasil hitung mean</code>
6	<code>float ada_k_rem [3][3]; //array hasil hitung mean</code>
7	<code>float ada_k_kecelakaan [3][3]; //array hasil hitung mean</code>
8	<code>float hasilMeanStd[3][3]; //array perhitungan mean</code>
9	<code>float s [3] = {0}, jData[3];</code>
10	<code>float latih_berjalan [3][16] = {</code>
11	<code>{11009, 13757, 9035, 4527, 5922, 22359, 7971, 8248, 3162,</code>
12	<code>5086},</code>
13	<code>{2.69, 2.22, 4.56, 2.13, 2.68, 2.39, 3.49, 2.79, 2.18,</code>
14	<code>3.29},</code>
15	<code>{2.28, 3.16, 5.72, 3.37, 2.99, 2.33, 3.1, 2.04, 1.23, 2.62}</code>
16	<code>};</code>
17	<code>float latih_rem [3][16] = {</code>
18	<code>{11883, 16340, 12194, 8326, 6315, 10578, 12093, 2851, 6157,</code>
19	<code>9038, 15367, 8048, 5792, 14029},</code>
20	<code>{3.22, 2.88, 3.09, 4.69, 4.59, 2.53, 3.13, 2.22, 4.25, 2.88,</code>
21	<code>3.53, 2.72, 2.09, 2.34},</code>
22	<code>{3.91, 3.18, 3.1, 2.96, 2.35, 3.54, 3.06, 3.12, 2.71, 2.43,</code>
23	<code>1.49, 7.19, 2.16, 2.76}</code>
24	<code>};</code>
25	<code>float latih_kecelakaan [3][16] = {</code>
26	<code>{12871, 28585, 9771, 18907, 8345, 6814, 5561, 25874, 7571,</code>
27	<code>5323, 17114, 11188, 27012, 28193, 17884, 25234},</code>
28	<code>{2.94, 25.12, 16.72, 14.03, 10.84, 6.84, 6.04, 9.79, 21.47,</code>
29	<code>7.06, 25.07, 5.13, 4.53, 9.52, 11.18, 4.23},</code>
30	<code>{7.46, 3.6, 3.96, 21.79, 6.94, 10.9, 4.02, 5.55, 15.54,</code>
31	<code>8.33,</code>
32	<code>8.83, 5.24, 5.52, 15.88, 5.42, 4.4}</code>
33	<code>};</code>

Berdasarkan Tabel 5.17 di atas merupakan proses inisialisasi variabel beserta tipe data yang di gunakan pada perhitungan metode *Naive bayes*. Inisialisasi variable menggunakan variable bebas agar mempermudah dalam pengimplementasian di keseluruhan sistem. Pada baris ke-1 hingga baris ke-3 merupakan inisialisasi dari tiap kelas untuk menghitung nilai prior, pada baris ke-4 adalah inisialisasi variable untuk menghitung nilai prior, pada baris ke-5 sampai baris ke-8 merupakan variabel array dua dimensi hasil hitung mean dari masing masing fitur, pada baris ke-9 digunakan untuk menghitung proses perhitungan standar deviasi, dan ada baris ke-10 sampai baris ke-24 merupan data latih dengan array 2 dimensi.

Tabel 5. 18 Kode program perhitungan fungsi probPrior()

Baris	Kode Program
1	<code>void probPrior() {</code>
2	<code> pro_ber =ada_berjalan/(ada_berjalan+ada_rem+ada_kecelakaan);</code>
3	<code> pro_rem =ada_rem / (ada_berjalan + ada_rem + ada_kecelakaan);</code>
4	<code> pro_kec=ada_kecelakaan/(ada_berjalan+ada_rem+ ada_kecelakaan);</code>
5	<code>}</code>

Tabel 5.18 di atas menunjukkan implementasi program untuk menghitung nilai peluang *prior*. Pada fungsi probPrior pada baris ke-2 yaitu menghitung *prior* “berjalan”, baris ke-3 yaitu menghitung *prior* “rem”, dan pada baris ke-4 menghitung *prior* “kecelakaan”, dengan cara menjumlah seluruh data pada setiap kondisi lalu di bagi dengan salah satu kondisi.

Tabel 5. 19 Kode program perhitungan fungsi mean dan standar deviasi

Baris	Kode Program
1	<code>void meanStd(int y, float d_Latih[3][16], int z) {</code>
2	<code> for (int j = 0; j < 3; j++) {</code>
3	<code> for (int i = 0; i < y; i++) {</code>
4	<code> jData[j] = jData[j] + d_Latih[j][i];</code>
5	<code> }</code>
6	<code> hasilMeanStd[0][j] = jData[j] / y;</code>
7	<code> }</code>
8	<code> for (int j = 0; j < 3; j++) {</code>
9	<code> for (int i = 0; i < y; i++) {</code>
10	<code> s[j] = s[j] + (pow(d_Latih[j][i] - hasilMeanStd[0][j], 2));</code>
11	<code> }</code>
12	<code> hasilMeanStd[1][j] = sqrt(s[j] / (y - 1));</code>
13	<code> if (z == 0) {</code>
14	<code> ada_k_berjalan [0][j] = hasilMeanStd[0][j];</code>
15	<code> ada_k_berjalan [1][j] = hasilMeanStd[1][j];</code>
16	<code> ada_k_berjalan [2][j] = hasilMeanStd[2][j];</code>
17	<code> } if (z == 1) {</code>
18	<code> ada_k_rem [0][j] = hasilMeanStd[0][j];</code>
19	<code> ada_k_rem [1][j] = hasilMeanStd[1][j];</code>
20	<code> ada_k_rem [2][j] = hasilMeanStd[2][j];</code>
21	<code> } else {</code>
22	<code> ada_k_kecelakaan [0][j] = hasilMeanStd[0][j];</code>
23	<code> ada_k_kecelakaan [1][j] = hasilMeanStd[1][j];</code>
24	<code> ada_k_kecelakaan [2][j] = hasilMeanStd[2][j];</code>
25	<code> }</code>
26	<code> jData[j] = {0};</code>
27	<code> s [j] = 0;</code>
28	<code> }</code>
29	<code>}</code>

Tabel 5.19 diatas menunjukkan fungsi meanStd() yang didalamnya menunjukkan implementasikan perhitungan mean dan standar deviasi, pada baris ke 2 hingga baris ke-7 menghitung mean dari setiap kondisi, dan baris ke-8 hingga baris ke-28 perhitungan standar deviasi dan disimpan pada array ada_k_berjalan, ada_k_rem, dan ada_k_kecelakaan.

Tabel 5. 20 Kode program perhitungan fungsi gaussian

Baris	Kode Program
1	void gaussian(float data_uji[3], float data_latih[2][3]) {
2	double a, b, c, d, k, l, z;
3	//mencari nilai gaussian setiap fitur
4	for (int i = 0; i < 3; i++) {
5	a = 2 * 3.14 * (pow(data_latih[1][i], 2)); //s.dev
6	b = (pow((data_uji[i] - data_latih[0][i]), 2)); //mean
7	k = (2 * pow(data_latih[1][i], 2)); //s.dev
8	l = - (b / k);
9	c = pow(2.718282, l);
10	d = 1 / sqrt(a);
11	gaussian[gauske][i] = c * d;
12	}
13	gauske++;
14	}

Tabel 5.20 diatas menunjukkan fungsi gaussian() yang sebelumnya dijelaskan pada diagram alir Gambar 5.10, dimana baris ke-5 hingga baris ke-13 adalah perhitungan yang penerapannya sama dengan diagram alir tersebut.

Tabel 5. 21 Kode program perhitungan fungsi probPosterior

Baris	Kode Program
1	void probPosterior(float prior, int i) {
2	for (int j = 0; j < 3; j++) {
3	if (j == 0) {
4	hasil[i] = (gaussian[i][j] * 1000); //dikali 1000
5	} else {
6	hasil[i] = hasil[i] * gaussian[i][j] * 1000;
7	}
8	}
9	hasil[i] = hasil[i] * prior;
10	}

Tabel 5.21 diatas menjelaskan fungsi probPosterior pada baris ke-2 hingga baris ke-9 merupakan perhitungan nilai gaussian seriap fitur di kalikan dengan nilai peluang prior sehingga di dapam peluang posterior, lalu di masukkan kedalam array hasil.

Tabel 5. 22 Kode program perhitungan fungsi kesimpulan

Baris	Kode Program
1	<code>void kesimpulan() {</code>
2	<code> if (hasil[0] > hasil[1] && hasil[0] > hasil[2]) {</code>
3	<code> Serial.print("berjalan = "); //Berjalan\</code>
4	<code> Serial.println(hasil[1]);</code>
5	<code> } else if (hasil[1] > hasil[0] && hasil[1] > hasil[2]) {</code>
6	<code> Serial.print("rem"); //rem</code>
7	<code> Serial.println(hasil[2]);</code>
8	<code> } else if (hasil[2] > hasil[0] && hasil[2] > hasil[1]) {</code>
9	<code> Serial.print("tabrak"); //tabrak</code>
10	<code> Serial.println(hasil[3]);</code>
11	<code> bluetooth.println("Terjadi Kecelakaan");</code>
12	<code> SIM800L.write("AT+CMGS=\""082xxxxxxx\""\r\n");</code>
13	<code> delay(500);</code>
14	<code> SIM800L.write("Mobil teridentifikasi kecelakaan");</code>
15	<code> delay(500);</code>
16	<code> SIM800L.write((char)26);</code>
17	<code> delay(5000);</code>
18	<code> Serial.println("SMS Selesai Dikirim");</code>
19	<code> }</code>
20	<code>}</code>

Tabel 5.22 diatas merupakan fungsi mencari nilai peluang posterior terbesar dari masing masing fitur, nilai yang di bandingkan didapat dari array hasil []. Baris ke-2 hingga baris ke-4 kondisi dimana hasil menunjukkan nilai peluang posterior kejadian "berjalan" lebih besar, Baris ke-5 hingga baris ke-7 kondisi dimana hasil menunjukkan nilai peluang posterior kejadian "rem" lebih besar, dan Baris ke-8 hingga baris ke-20 kondisi dimana hasil menunjukkan nilai peluang posterior kejadian "kecelakaan" lebih besar, pada baris ke-11 adalah pengiriman hasil melalui module *Bluetooth*, dan pada baris ke-12 hingga baris ke-18 pengiriman hasil melalui module *GSM SIM800L*.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan menjelaskan mengenai pengujian dan analisis dari pengujian yang di lakukan sesuai dengan sistem yang telah di buat. Tujuan dilakukannya pengujian untuk mengetahui apakah kebutuhan yang di perlukan sistem dapat terpenuhi. Pengujian sistem pendeteksi tabrakan ini meliputi pengujian deteksi kecelakaan menggunakan mobil remote *control* dari sensor akselerometer dan sensor 801s vibration yang telah dirangkai menggunakan menggunakan metode naïve bayes

6.1 Pengujian Sistem Deteksi Tabrakan Pada Setiap Kondisi

Pengujian sistem deteksi tabrakan pada setiap kondisi merupakan tahapan mendapatkan nilai acuan dari setiap kondisinya, terdapat beberapa langkah pengujiannya yaitu:

1. Menghubungkan sistem deteksi tabrakan pada mobil *remote control* dengan laptop melalui kabel serial.
2. Dilakukan pengujian dengan 3 skenario untuk mendapatkan nilai pembacaan dari sensor dari setiap kondisi. Yang di jelaskan sebagai berikut :
 - a. Pengujian mobil *remote control* dalam kondisi berjalan kedepan tanpa mengerem, dilakukan sebanyak 10 kali pengambilan data.
 - b. Pengujian mobil *remote control* dalam kondisi berjalan lalu mengerem, dilakukan sebanyak 14 kali pengambilan data.
 - c. Pengujian mobil *remote control* dalam kondisi di tabrakkan kea rah tembok, dilakukan sebanyak 16 kali pengamilan data.

Total pengambilan data dari setiap kondisi yaitu 40 kali percobaan. Semua percobaan di lakukan dengan jarak percobaan dari titik "A" ke titik "B" sejauh 3 meter.

6.1.1 Tujuan Pengujian Pada Setiap kondisi

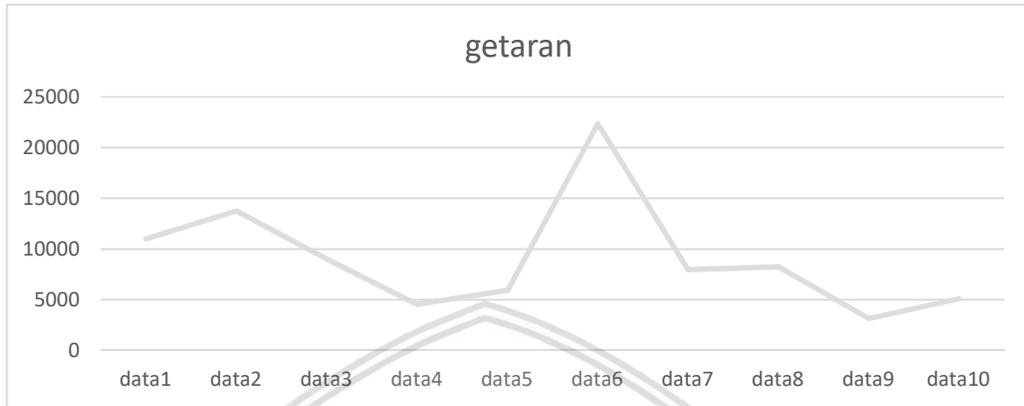
Tujuan dilakukannya pengujian ini untuk mengetahui apakah nilai yang di baca oleh sensor dapat di gunakan sebagai nilai data latih yang di mana nilai tersebut dimasukkan kedalam program naïve bayes, acuan yang di gunakan sesuai dengan skenario dan sesuai dengan penelitian terdahulu dimana nilai yang di dapat sensor akselerometer pada saat kejadian kecelakaan adalah $>4g$.

6.1.2 Analisis Dan Pengujian Mobil Remote Control Berjalan Kedepan

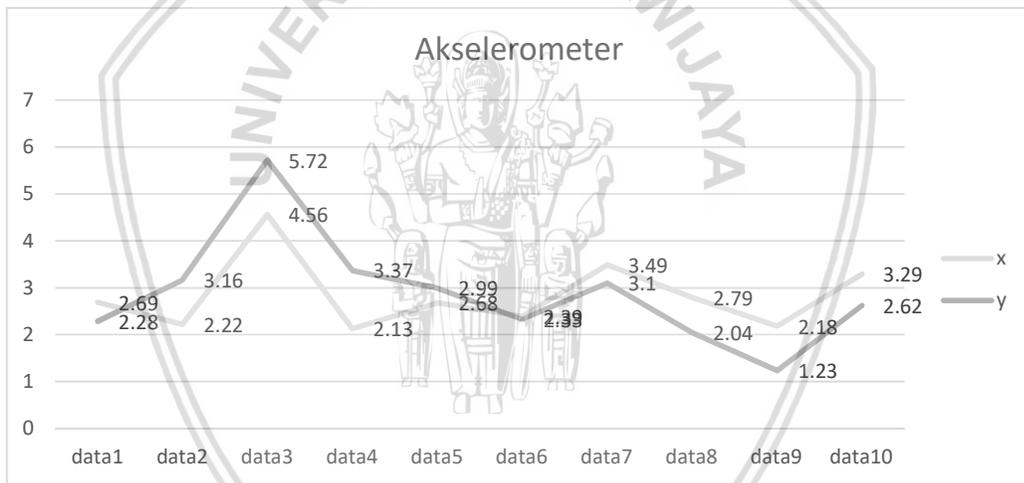
Pengujian mobil *remote control* berjalan kedepan dilakukan sebanyak 10 kali dari setiap kejadian saat mobil berjalan sejauh 3 meter dan di ambil nilai amplitudonya. Data lengkap pengujian dilampirkan pada "Lampiran A Data Latih Mobil Berjalan" dan "Lampiran B Data Latih Mobil Berjalan".

6.1.2.1 Hasil Pengujian Saat Mobil berjalan

Hasil pengujian didapat data dari perubahan data sensor akselerometer dan sensor getar yang ditampilkan pada Gambar 6.1 dan Gambar 6.2.



Gambar 6. 1 Amplitudo sensor getar dari percobaan mobil berjalan



Gambar 6. 2 Amplitudo sensor akselerometer dari percobaan mobil berjalan

Gambar di atas menunjukkan grafik dari sensor 801s pada Gambar 6.1 dapat dilihat nilai perubahan nilai sensor getaran pada sisi vertical adalah besaran nilai sensor getar yang di dapat dan sisi horizontal adalah data sampel dari pembacaan sensor getar, nilai sensor getar saat minimum yaitu sebesar 5000, nilai saat maksimum sebesar 20000 dan mean sebesar 9107.

Gambar 6.2 menunjukkan grafik sensor MMA7361 perubahan nilai sensor pada sisi vertical adalah pembacaan percepatan dan sisi horizontal adalah data sampel dari pembacaan percepatan, untuk nilai maksimum pada sumbu X sebesar 5.72 dan sumbu Y sebesar 4.56. untuk rata-rata dari masing masing sumbu ,untuk X yaitu 2,8g dan Y adalah 2.8g

6.1.2.2 Analisis Pengujian Saat Mobil berjalan

Pada Tabel 6.1 akan menjelaskan keterkaitan data dengan penelitian sebelumnya. Dimana nilai pembacaan sensor akselerometer $<4g$ termasuk dalam kondisi tidak terjadi kecelakaan.

Tabel 6. 1 Kesesuaian Pengambilan Data Percepatan Mobil Berjalan

	X (g-force)	Y (g-force)	Kesesuaian Nilai ($x < 4g$)	Kesesuaian Nilai ($y < 4g$)
data1	2.69	2.28	Sesuai	Sesuai
data2	2.22	3.16	Sesuai	Sesuai
data3	4.56	5.72	Tidak Sesuai	Tidak Sesuai
data4	2.13	3.37	Sesuai	Sesuai
data5	2.68	2.99	Sesuai	Sesuai
data6	2.39	2.33	Sesuai	Sesuai
data7	3.49	3.1	Sesuai	Sesuai
data8	2.79	2.04	Sesuai	Sesuai
data9	2.18	1.23	Sesuai	Sesuai
data10	3.29	2.62	Sesuai	Sesuai

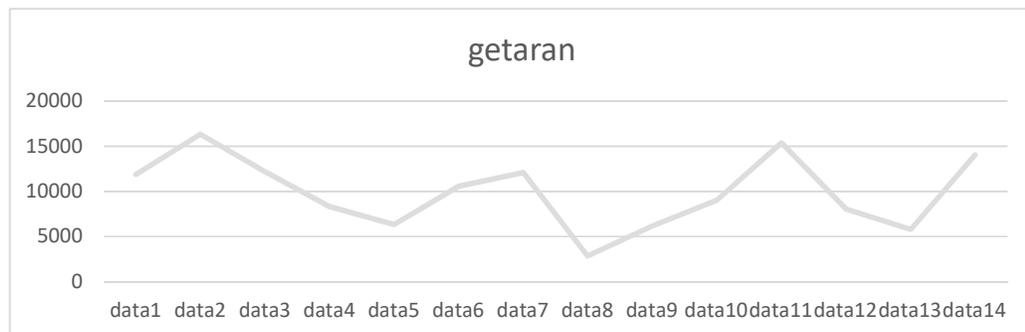
Berdasarkan hasil pengujian pada saat mobil berjalan didapat nilai akselerometer dari 10 data yang di ambil 9 data menunjukkan nilai di bawah $4g$ yang menunjukkan bahwa mobil tidak terjadi kecelakaan. Adapun persentase kesesuaian data sebesar 90%. Maka nilai dapat di terapkan kedalam data latih.

6.1.3 Analisis dan Pengujian Mobil Remote Control Berhenti Mendadak

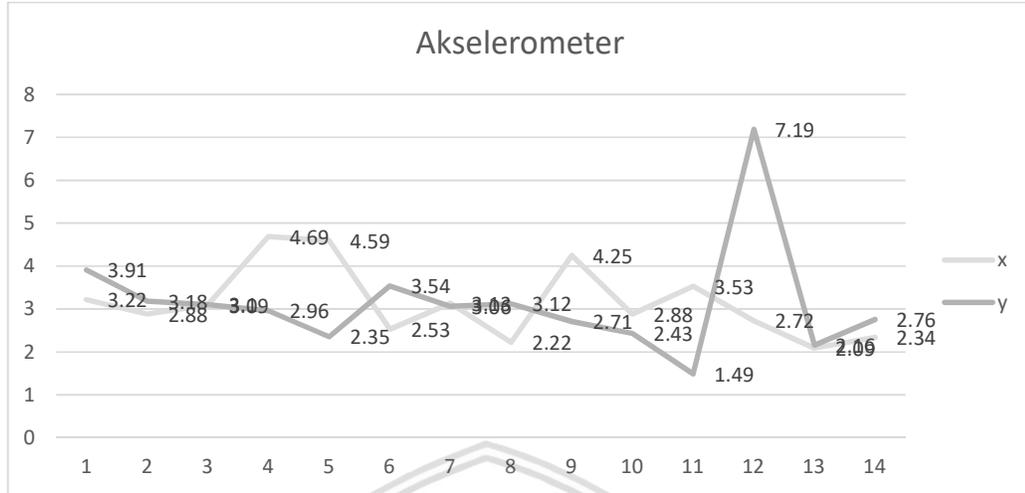
Pengujian mobil *remote control* berjalan lalu mengerem dilakukan sebanyak 14 kali dari setiap kejadian saat mobil berjalan sejauh 3 meter dan di ambil nilai amplitudonya. Data lengkap pengujian dilampirkan pada “Lampiran C Data Latih Mobil Berhenti Mendadak” dan “Lampiran D Data Latih Mobil Berhenti Mendadak”.

6.1.3.1 Hasil Pengujian Saat Mobil Berhenti Mendadak

Hasil pengujian didapat data dari perubahan data sensor akselerometer dan sensor getar yang ditampilkan pada Gambar 6.3 dan Gambar 6.4.



Gambar 6. 3 Amplitudo sensor getar dari percobaan mobil mengerem



Gambar 6. 4 Amplitudo sensor akselerometer dari percobaan mobil berhenti mendadak

Gambar di atas menunjukkan grafik dari sensor 801s pada Gambar 6.3 dapat dilihat nilai perubahan nilai sensor getaran pada sisi vertical adalah besaran nilai sensor getar yang di dapat dan sisi horizontal adalah data sampel dari pembacaan sensor getar, nilai sensor getar saat minimum yaitu sebesar 4000, nilai saat maksimum sebesar 16000 dan mean sebesar 9929.

Gambar 6.4 menunjukkan grafik sensor MMA7361 perubahan nilai sensor pada sisi vertical adalah pembacaan percepatan dan sisi horizontal adalah data sempel dari pembacaan percepatan, untuk nilai maksimum pada sumbu X sebesar 7,19 dan sumbu Y sebesar 4,59. untuk rata-rata dari masing masing sumbu ,untuk X yaitu 3,15g dan Y adalah 3,14g.

6.1.3.2 Analisis Pengujian Saat Mobil Berhenti Mendadak

Pada Tabel 6.2 akan menjelaskan keterkaitan data dengan penelitian sebelumnya. Dimana nilai pembacaan sensor akselerometer <4g termasuk dalam kondisi tidak terjadi kecelakaan, karna kondisi berhenti mendadak termasuk kejadian bukan kecelakaan.

Tabel 6. 2 Kesesuaian Pengambilan Data Percepatan Mobil Berhenti Mendadak

	X (g-force)	Y (g-force)	Kesesuaian nilai(x<4g)	Kesesuaian nilai(y<4g)
data1	3.22	3.91	Sesuai	Sesuai
data2	2.88	3.18	Sesuai	Sesuai
data3	3.09	3.1	Sesuai	Sesuai
data4	4.69	2.96	Tidak Sesuai	Sesuai
data5	4.59	2.35	Tidak Sesuai	Sesuai
data6	2.53	3.54	Sesuai	Sesuai
data7	3.13	3.06	Sesuai	Sesuai
data8	2.22	3.12	Sesuai	Sesuai

data9	4.25	2.71	Tidak Sesuai	Sesuai
data10	2.88	2.43	Sesuai	Sesuai
data11	3.53	1.49	Sesuai	Sesuai
data12	2.72	7.19	Sesuai	Sesuai
data13	2.09	2.16	Sesuai	Sesuai
data14	2.34	2.76	Sesuai	Sesuai

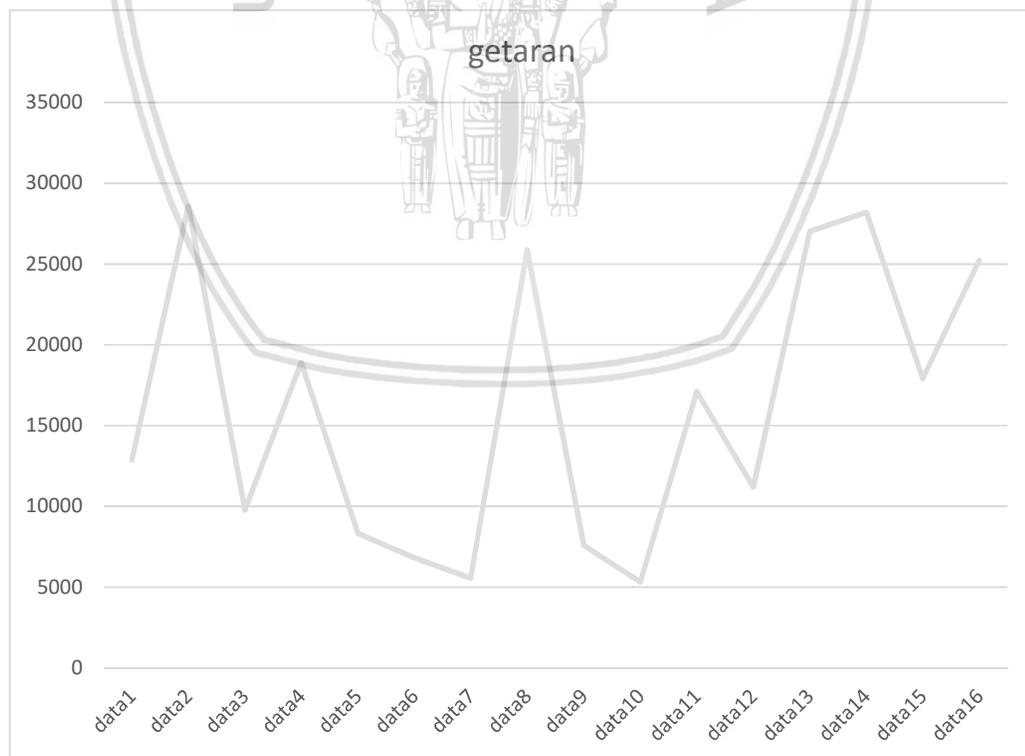
Berdasarkan pengujian yang diambil dari kejadian mobil berhenti mendadak didapat nilai akselerometer nilai $>4g$ ada 3 data yang tidak sesuai dengan penelitian terdahulu. Persentasi kesesuaian pada sumbu X sebesar 78% dan persentase kesesuaian pada sumbu Y sebesar 100%, Maka nilai dapat di terapkan kedalam data latih.

6.1.4 Analisis dan Pengujian Mobil Remote Control Menabrak

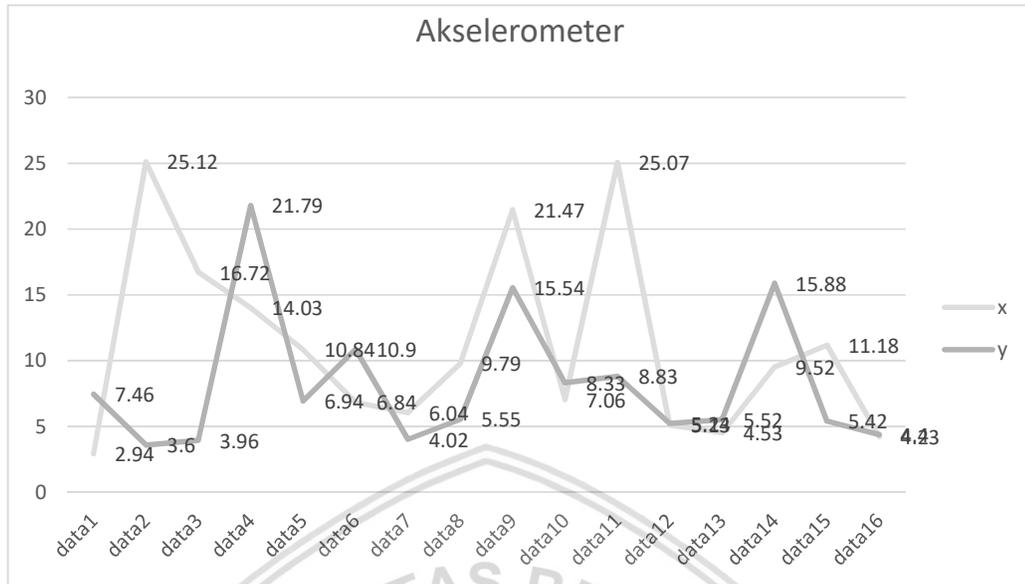
Pengujian mobil *remote control* berjalan lalu mengerem dilakukan sebanyak 16 kali dari setiap kejadian saat mobil berjalan sejauh 3 meter dan di ambil nilai amplitudonya. Data lengkap pengujian dilampirkan pada “Lampiran E Data Latih Mobil Kecelakaan” dan “Lampiran F Data Latih Mobil Kecelakaan”.

6.1.4.1 Hasil Pengujian Saat Mobil Menabrak

Hasil pengujian didapat data dari perubahan data sensor akselerometer dan sensor getar yang ditampilkan pada Gambar 6.5 dan Gambar 6.6.



Gambar 6. 5 Amplitudo sensor getar dari percobaan mobil menabrak



Gambar 6. 6 Amplitudo sensor getar dari percobaan mobil menabrak

Gambar di atas menunjukkan grafik dari sensor 801s pada Gambar 6.5 dapat dilihat nilai perubahan nilai sensor getaran pada sisi vertical adalah besaran nilai sensor getar yang di dapat dan sisi horizontal adalah data sampel dari pembacaan sensor getar, nilai sensor getar saat minimum yaitu sebesar 5000, nilai saat maksimum sebesar 29000 dan mean sebesar 16015.

Gambar 6.4 menunjukkan grafik sensor MMA7361 perubahan nilai sensor pada sisi vertical adalah pembacaan percepatan dan sisi horizontal adalah data sampel dari pembacaan percepatan, untuk nilai maksimum pada sumbu X sebesar 25,12 dan sumbu Y sebesar 21,79. untuk rata-rata dari masing masing sumbu ,untuk X yaitu 11.28g dan Y adalah 8,33g.

6.1.4.1 Analisis Pengujian Saat Mobil Menabrak

Pada Tabel 6.3 akan menjelaskan keterkaitan data dengan penelitian sebelumnya. Dimana nilai pembacaan sensor akselerometer harus lebih dari 4g. karna kejadian termasuk kecelakaan.

Tabel 6. 3 Kesesuaian Pengambilan Data Percepatan Mobil Kecelakaan

	X (g-force)	Y (g-force)	Kesesuaian nilai(x<4g)	Kesesuaian nilai(y<4g)
data1	2.94	7.46	Tidak Sesuai	Sesuai
data2	25.12	3.6	Sesuai	Sesuai
data3	16.72	3.96	Sesuai	Sesuai
data4	14.03	21.79	Sesuai	Sesuai
data5	10.84	6.94	Sesuai	Sesuai
data6	6.84	10.9	Sesuai	Sesuai
data7	6.04	4.02	Sesuai	Sesuai
data8	9.79	5.55	Sesuai	Sesuai

data9	21.47	15.54	Sesuai	Sesuai
data10	7.06	8.33	Sesuai	Sesuai
data11	25.07	8.83	Sesuai	Sesuai
data12	5.13	5.24	Sesuai	Sesuai
data13	4.53	5.52	Sesuai	Sesuai
data14	9.52	15.88	Sesuai	Sesuai
data15	11.18	5.42	Sesuai	Sesuai
data16	4.23	4.4	Sesuai	Sesuai

Berdasarkan pengujian didapat nilai kesesuaian terhadap scenario yang telah dibuat terhadap penelitian terdahulu pada sumbu X sebesar 98% dan pada sumbu Y sebesar 100%.

6.2 Pengujian Sistem Deteksi Tabrakan Dengan Klasifikasi Naïve Bayes

Sistem pendeteksi tabrakan berdasarkan sensor akselerometer dan sensor getar menggunakan metode naïve bayes mempunyai tujuan utama untuk mendapat klasifikasi berjalan, berhenti mendadak, dan menabrak. Oleh karna itu perlu di ketahui tingkat keakuratan pada sistem dalam melakukan klasifikasi. Pengujian ini terdapat beberapa langkah pengujian yaitu :

1. Menghubungkan sistem deteksi tabrakan pada mobil *remote control* dengan laptop melalui kabel serial.
2. Pengujian dilakukan sebanyak 15 kali , pada setiap kejadian berjalan, kejadian berhenti mendadak, dan kejadian menabrak.
3. Sistem akan membaca dan menghitung nilai klasifikasi ketika sensor getar mamberi nilai >5000.
4. Hasil klasifikasi menghasilkan prediksi dari presiksi nilai sensor dan menampilkan prediksi.
5. Penentuan hasil akurasi klasifikasi sistem akan di bandingkan dengan pengamatan secara langsung.

6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini di maksudkan untuk mengetahui keakurasian sistem melakukan klasifikasi menggunakan metode naïve bayes.

6.2.2 Pengujian Mobil Remote Control Berjalan Kedepan Menggunakan Naïve Bayes

Pada Tabel 6.4 akan menjelaskan keterkaitan kejadian dengan metode naïve bayes, dimana data yang di hasilkan metode apakah sama dengan kejadian berjalan ketika diuji.

Tabel 6. 4 Pengujian Saat Mobil Berjalan

Percobaan	Data Input			Hasil Sistem	Keseuaian
	Getaran (A)	X (g)	Y (g)		
Percobaan 1	19971	2.39	1.33	B	S
Percobaan 2	7607	1.92	0.99	B	S
Percobaan 3	7782	1.55	2.68	B	S
Percobaan 4	6604	0.89	0.52	B	S
Percobaan 5	7221	1.58	1.67	B	S
Percobaan 6	5017	1.95	1.26	B	S
Percobaan 7	5147	1	0.74	B	S
Percobaan 8	5688	1.94	0.52	B	S
Percobaan 9	5017	1.84	0.39	B	S
Percobaan 10	1774	2	3.2	B	S
Percobaan 11	2590	2.3	1.8	B	S
Percobaan 12	5571	1.56	0.71	B	S
Percobaan 13	2434	3.3	3.1	B	S
Percobaan 14	7603	2.75	3.95	R	TS
Percobaan 15	6613	1.53	0.96	B	S
Percobaan 16	5142	1.59	0.74	B	S

Keterangan Table :

B = Berjalan

R = Rem

S = Sesuai

TS = Tidak Sesuai

Berdasarkan pengujian klasifikasi keadaan kondisi mobil menggunakan metode naïve bayes yang di tunjukan pada tabel 6.4, menunjukan bahwa dari 16 data dari kedaan berjalan terdapat 1 data yang tidak sesuai dengan keadaan yaitu terbaca keadaan berhenti mendadak. Berikut proses perhitungan akurasi adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total Data} - \text{Data Tidak Sesuai}}{\text{Total data}} \times 100\% \\
 &= \frac{16-1}{16} \times 100\% \\
 &= 93,7 \%
 \end{aligned}$$

Berdasarkan perhitungan didapat nilai sebesar 93,7 % dari pengujian sistem dalam kondisi berjalan, dengan menggunakan metode naïve bayes.

6.2.3 Pengujian Mobil Remote Control Berhenti Mendadak Menggunakan Naïve Bayes

Pada Tabel 6.5 akan menjelaskan keterkaitan kejadian dengan metode naïve bayes, dimana data yang di hasilkan metode apakah sama dengan kejadian berhenti mendadak ketika diuji.

Tabel 6. 5 Pengujian Saat Mobil Berhenti mendadak

Percobaan	Data Input			Hasil Sistem	Keseuaian
	Getaran (A)	X (g)	Y (g)		
Percobaan 1	13743	2.31	2.27	R	S
Percobaan 2	9161	2.47	3.55	R	S
Percobaan 3	12642	1.97	4.2	R	S
Percobaan 4	10057	2.22	0.61	R	S
Percobaan 5	7106	0.66	0.36	R	S
Percobaan 6	8319	0.78	2.14	R	S
Percobaan 7	12114	3.69	4.61	R	S
Percobaan 8	11432	1.07	0.7	R	S
Percobaan 9	8368	3.23	2.17	R	S
Percobaan 10	9253	1.06	2.52	R	S
Percobaan 11	9934	0.43	1.89	R	S
Percobaan 12	13222	2.26	2.45	R	S
Percobaan 13	6321	1.43	0.61	B	TS
Percobaan 14	8514	1.23	1.61	R	S
Percobaan 15	10063	2.03	2.39	R	S
Percobaan 16	7969	0.56	1.02	B	TS

Keterangan Table :

B = Berjalan

R = Rem

S = Sesuai

TS = Tidak Sesuai

Berdasarkan pengujian klasifikasi keadaan kondisi mobil menggunakan metode naïve bayes yang di tunjukan pada tabel 6.5, menunjukan bahwa dari 16 data dari keadaan berjalan terdapat 2 data yang tidak sesuai dengan keadaan yaitu terbaca keadaan Berjalan. Berikut proses perhitungan akurasi adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total Data} - \text{Dat Tidak Sesuai}}{\text{Total data}} \times 100\% \\
 &= \frac{16-2}{16} \times 100\% \\
 &= 87,5 \%
 \end{aligned}$$

Berdasarkan perhitungan didapat nilai sebesar 87,5% dari pengujian sistem dalam kondisi Berhenti Mendadak, dengan menggunakan metode naïve bayes.

6.2.4 Pengujian Mobil Remote Control Kecelakaan Menggunakan Naïve Bayes

Pada Tabel 6.6 akan menjelaskan keterkaitan kejadian dengan metode naïve bayes, dimana data yang di dihasilkan metode apakah sama dengan kejadian kecelakaan ketika diuji.

Tabel 6. 6 Pengujian Saat Mobil Kecelakaan

Percobaan	Data Input			Hasil Sistem	Keseuaian
	Getaran (A)	X (g)	Y (g)		
Percobaan 1	17546	4.04	6.05	T	S
Percobaan 2	28835	7.21	1.74	T	S
Percobaan 3	5253	0.46	0.15	T	S
Percobaan 4	7688	0.07	0.12	B	TS
Percobaan 5	14418	0.24	4.12	T	S
Percobaan 6	15186	10.65	12.35	T	S
Percobaan 7	32286	16.49	2.92	T	S
Percobaan 8	6558	12.12	3.48	T	S
Percobaan 9	8306	0.07	0.02	T	S
Percobaan 10	5822	0.05	0.04	T	S
Percobaan 11	18043	5.97	0.48	T	S
Percobaan 12	7039	0.41	0.52	T	S
Percobaan 13	23704	5.49	3.87	T	S
Percobaan 14	8467	0.04	0.04	T	S
Percobaan 15	24520	14.17	6.96	T	S
Percobaan 16	20888	7.11	2.38	T	S

Keterangan Table :

T = Tabrakan

B = Berjalan

S = Sesuai

TS = Tidak Sesuai

Berdasarkan pengujian klasifikasi keadaan kondisi mobil menggunakan metode naïve bayes yang di tunjukan pada tabel 6.6, menunjukan bahwa dari 16 data dari keadaan berjalan terdapat 1 data yang tidak sesuai dengan keadaan yaitu terbaca keadaan Berjalan. Berikut proses perhitungan akurasi adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total Data} - \text{Data Tidak Sesuai}}{\text{Total data}} \times 100\% \\
 &= \frac{16-1}{16} \times 100\% \\
 &= 93,7 \%
 \end{aligned}$$

Berdasarkan perhitungan didapat nilai sebesar 93,7% dari pengujian sistem dalam kondisi Kecelakaam, dengan menggunakan metode naïve bayes.

6.3 Pengujian Pengiriman Notifikasi Menggunakan SIM 800I

Berdasarkan pengujian saat pengujian Kecelakaan mobil *Remote Control* dari 16 kali percobaan yang di terangkan pada tabel 6.6 pengiriman notifikasi melalui SMS menggunakan module SIM 800I akan dijalankan ketika mobil terindentikasi kecelakaan makan dari Hasil pengiriman yang berhasil adalah 15 kali berhasil terkirim dan 1 kali gagal.

BAB 7 PENUTUP

Pada bab ini memuat hasil penarikan kesimpulan dari beberapa tahap yang telah dilakukan sesuai dengan rumusan masalah yang ada, dan pada bab ini terdapat saran yang diharapkan dapat digunakan untuk mengembangkan penelitian selanjutnya yang serupa dengan penelitian ini.

7.1 Kesimpulan

Kesimpulan yang dapat ditarik dari beberapa hasil analisis dan pengujian yang telah dilakukan pada penelitian ini sesuai dengan rumusan masalah adalah sebagai berikut:

1. Pada implementasi sistem dilakukan proses pengambilan data uji dengan cara manual untuk di jadikan data masukan untuk perhitungan naïve bayes. Penyesuaian data di tetapkan dengan mengacu pada data akselerometer jika data lebih dari 4g maka masuk dalam kondisi kecelakaan. Pengambilan data dari kondisi berjalan di dapat persentase kesesuaian sebesar 90%, lalu pada saat pengambilan dalam kondisi berhenti mendadak sebesar 78%, dan saat kondisi kecelakaan sebesar 98%.
2. Hasil data uji dari hasil klasifikasi menggunakan metode naïve bayes didapat dengan pengambilan data uji pada masing masing 16 kali pengujian pada setiap skenario, pada kondisi berjalan didapat nilai keakurasian sebesar 98,7 %, kondisi berhenti mendadak didapat nilai keakurasian sebesar 87,5 %, dan kondisi kecelakaan didapat nilai keakurasian sebesar 98,7 %.
3. Hasil data uji pengiriman menggunakan module SIM 800I saat mobil remote kontrol yang diambil sebanyak 16 kali teridentifikasi kecelakaan nilai keberhasilan pengiriman pesan sebesar 98,7%.

7.2 Saran

Terdapat beberapa saran yang kedepannya dapat di jadikan untuk acuan pada pengembangan penelitian yang serupa dengan penelitian ini:

1. Dalam klasifikasi dapat di tambahkan kondisi mobil apakah tertabrak dari depan atau samping.
2. Menambahkan parameter baru untuk klasifikasi pendeteksian kecelakaan.
3. Menambahkan perintah cek kondisi mobil saat ini ketika dibutuhkan.

DAFTAR PUSTAKA

- Astuti, E. H., 2016. *Sistem Pendukung Keputusan Deteksi Dini Penyakit Stroke Menggunakan Metode Naive Bayes*, Malang: Universitas Brawijaya.
- Athavan, K. & Jagadeeshwaran, S., 2012. AUTOMATIC AMBULANCE RESCUE SYSTEM. *International Journal of Advanced Technology & Engineering Research*, 2(2), pp. 89-91.
- Banzi, M., 2014. *Arduino Nano Schematic*. [Online] Available at: www.arduino.cc/en/Main/ArduinoBoardNano [Accessed 10 march 2017].
- BPJT, 2018. *Badan Pengatur Jalan Tol Kementerian PUPR Republik Indonesia*. [Online] Available at: <http://bpjt.pu.go.id/konten/progress/beroperasi> [Accessed 25 February 2018].
- Hidayat, A., 2018. *Supervisor* [Interview] (12 March 2018).
- Invensense, 2013. *MPU-6000/MPU-6050 Product Specification*. 3.4 ed. Sunnyvale: Invensense.
- invensense, 2015. *Accelerometer MPU-6050 Schematic*. [Online] Available at: www.invensense.com [Accessed 08 May 2016].
- Keselamatan, T. K. N., 2017. *Komite Nasional Keselamatan Transportasi*. [Online] Available at: knkt.dephub.go.id/webknkt/knkt.php [Accessed 15 November 2017].
- Kristianto, F., 2018. *Bisnis Indonesia*. [Online] Available at: <http://industri.bisnis.com/read/20180225/12/742925/bisnis-rental-mobil-diprediksi-terdongkrak-70-persen> [Accessed 14 Maret 2018].
- Prabha, C. & Sunitha, R., 2014. Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem. *International Journal of Advanced Research in Electrical*, 3(7), p. 10725.
- Rashida Nazir, A. T. S. M. S. R., 2014. Accident Prevention and Reporting System Using GSM (SIM 900D) and GPS (NMEA 0183). *Scientific Research Publishing*, Volume 7, pp. 286-293.
- Sofia, Y., 1999. *Studi Identifikasi Kecelakaan Lalu Lintas Di Kotip Depok*. - ed. Depok: Universitas Indonesia.
- Sudaryatmo, D., 2016. *Sistem deteksi tabrtakan pada mobil menggunakan sensor akselerometer*. - ed. malang: universitas brawijaya.
- Weiner, S. J., 2010. *Feasibility of a 802.11 VANET Based Car Accident Alert System*. s.l.:Northeastern University.