

**APLIKASI DETEKSI JATUH PADA PONSEL PINTAR
ANDROID (VOICE ALERT) MENGGUNAKAN SENSOR
AKSELEROMETER**



Disusun Oleh:

N a m a : Anggita Wisnu Pradan Putra

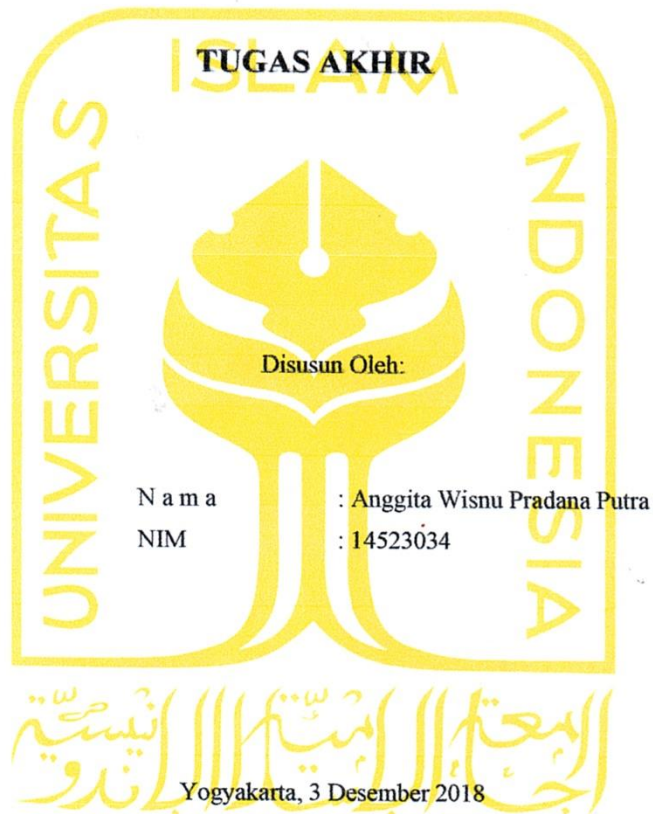
NIM : 14523034

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**APLIKASI DETEKSI JATUH PADA PONSEL PINTAR
ANDROID (VOICE ALERT) MENGGUNAKAN SENSOR
AKSELEROMETER**



Pembimbing,

(Syarif Hidayat S.kom., MIT.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**APLIKASI DETEKSI JATUH PADA PONSEL PINTAR
ANDROID (VOICE ALERT) MENGGUNAKAN SENSOR
AKSELEROMETER
TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 3 Desember 2018

Tim Penguji

Syarif Hidayat, S.Kom., M.IT.

Anggota 1

Elyza Gustri Wahyuni, S.T., M.Cs.

Anggota 2

Hari Setiaji, S.Kom., M.Eng.

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Drs. H. Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Anggita Wisnu Pradana Putra

NIM : 14523034

Tugas akhir dengan judul:

**APLIKASI DETEKSI JATUH PADA PONSEL PINTAR
ANDROID (VOICE ALERT) MENGGUNAKAN SENSOR
AKSELEROMETER**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 3 Desember 2018


(Anggita Wisnu Pradana Putra)

HALAMAN PERSEMBAHAN

Perjuangan merupakan pengalaman Berharga
Yang dapat menjadikan kita
Manusia yang berkualitas.
Tugas akhir ini kupersembahkan
Untuk kedua orang tuaku tercinta
Serta kakak dan adikku
Yang selalu mendukung serta nasihatnya
Yang menjadi jembatan perjalanan hidupku.

HALAMAN MOTO

“Kegagalan hanya terjadi bila kita menyerah”

(Lessing)

“Kebanggaan kita yang terbesar adalah bukan tidak pernah gagal, tetapi bangkit kembali
setiap kali kita jatuh”

(Confusius)

“Manusia tidak merancang untuk gagal, mereka gagal untuk merancang”

(William J. Siegel)

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga penulis dapat menyelesaikan skripsi dengan judul “Aplikasi Deteksi Jatuh Pada Ponsel Pintar Android (Voice Alert) Menggunakan Sensor Akselerometer”. Tidak lupa sholawat dan salam kita haturkan kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan para sahabatnya.

Laporan tugas akhir ini disusun sebagai syarat kelulusan yang harus ditempuh untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia dan sebagai salah satu penerapan ilmu yang didapat selama perkuliahan.

Di dalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada :

1. Allah SWT, yang telah memberikan keselamatan kemudahan dan petunjuk dalam menyelesaikan laporan tugas akhir dengan baik.
2. Dr. Raden Teduh Dirgahayu, S.T., M.Sc, selaku Kepala Jurusan Teknik Informatika.
3. Kedua orang tua penulis Bapak Sigit Yogyanto dan Ibu Istiyatun, terimakasih telah memberikan semua hal, baik materi dan doa yang tidak pernah putus kepada penulis.
4. Kakak penulis Galih Nur Kuncoro Jati dan adik penulis Arum Sari Tria Anggraeni yang selalu memberi kebahagiaan dan memberikan keceriaan.
5. Syarif Hidayat, S.Kom., MIT, selaku dosen pembimbing tugas akhir yang telah memberikan waktu dan membantu penulis untuk menyelesaikan tugas akhir.
6. Seluruh Dosen Teknik Informatika Universitas Islam Indonesia yang telah memberikan ilmu yang bermanfaat.
7. Teman-teman Teknik Informatika angkatan 2014 untuk semua bantuan dan dukungannya.
8. Serta ucapan terima kasih kepada semua pihak yang tidak bisa penulis sebutkan namanya satu persatu.

Penulis menyadari kalau dalam penyusunan laporan tugas akhir ini masih jauh dari kata sempurna. Oleh sebab itu dengan hati yang terbuka, Penulis mengharapkan kritik serta saran yang membangun guna kesempurnaan laporan tugas akhir ini. Semoga laporan tugas akhir ini bermanfaat bagi kita semua. Amin.

Yogyakarta, 4 April 2018

(Anggita Wisnu Pradana Putra)

SARI

Jatuh adalah kejadian yang dapat dialami oleh siapa saja, baik anak-anak, dewasa, maupun orang yang sudah lanjut usia. Pada rentang umur lanjut usia, resiko terjatuh akan semakin besar karena berkurangnya kemampuan fisik. Maka perlu dibuat sebuah aplikasi deteksi jatuh maka dengan segera kita dapat memberikan pertolongan yang lebih cepat sehingga dapat mengurangi resiko-resiko buruk yang terjadi. Deteksi jatuh dapat dibuat dengan *smartphone* android yang kita punya. Dengan memanfaatkan sensor akselerometer pada *smartphone* android kita dapat menghitung besarnya akselerasi yang selanjutnya diproses untuk menghasilkan sebuah data yang akan digunakan sebagai hitungan untuk mengukur kejadian jatuh. Pada penelitian ini jatuh dideteksi dari data yang didapat pada saat akselerasi. Dalam implementasinya, sistem harus bisa menerima masukan data agar dapat memberikan sebuah alert, tentunya sistem harus mempunyai *threshold* yang telah ditentukan. Sehingga apabila data pada sistem telah melewati *threshold* maka secara otomatis akan memberikan sebuah *alert*.

Kata kunci: Smartphone, android, threshold, dan deteksi jatuh.

GLOSARIUM

Threshold	Ambang batas dalam sebuah perhitungan.
Android	Sistem operasi berbasis LINUX yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan computer tablet.
Akselerometer	Sensor yang digunakan oleh sistem untuk mendeteksi orientasi suatu perangkat berdasarkan gerak ke segala arah atau dengan menggoyangkan yang memungkinkan fitur untuk bertindak.
Smartphone	Telepon genggam yang memiliki kemampuan dengan penggunaan dan fungsi yang menyerupai komputer.
Sensor	Jenis transduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, kimia menjadi tegangan dan arus listrik.

DAFTAR ISI

JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING....	Error! Bookmark not defined.
HALAMAN PENGESAHAN DOSEN PENGUJI	Error! Bookmark not defined.
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .	Error! Bookmark not defined.
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah	2
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penelitian	4
BAB II LANDASAN TEORI	6
2.1 Sensor.....	6
2.2 Sensor <i>Accelerometer</i>	6
2.3 Gerak Jatuh	8
2.4 Android	10
2.5 Java.....	13
2.6 <i>Decision Tree</i>	14
2.7 <i>Ensemble dan Bootstrap Aggregating</i>	14
2.8 <i>Random Forest</i>	15
2.9 Fitur	16
2.10 <i>Data Preprocessing</i>	16
2.11 <i>Feature Extraction</i>	17
2.12 <i>Feature Selection</i>	18
2.13 <i>Classification</i>	18
BAB III METODOLOGI	20
3.1 Analisis Kebutuhan	20
3.1.1 Analisis Kebutuhan Fungsi.....	20
3.1.2 Analisis Kebutuhan Masukan.....	20
3.1.3 Analisis Kebutuhan Keluaran	20
3.1.4 Analisis Kebutuhan Perangkat Lunak	21
3.1.5 Analisis Kebutuhan Perangkat Keras	21
3.2 Metode Perancangan	21

3.2.1 Metode Perancangan Aplikasi	21
3.2.2 Use Case Diagram.....	22
3.2.3 Activity Diagram.....	25
3.2.4 Perancangan Antarmuka	28
3.2.5 Perancangan Skema Sistem	29
3.2.6 Perancangan Data.....	30
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	34
4.1 Implementasi	34
4.1.1 Implementasi Keluaran Suara	34
4.1.2 Implementasi Aplikasi Android	37
4.1.3 Implementasi Perangkat Keseluruhan.....	38
4.2 Implementasi	39
4.2.1 Pengujian Aplikasi Android.....	39
4.3 Analisis Penelitian.....	50
4.3.1 Kelebihan Aplikasi.....	50
4.3.2 Kekurangan Aplikasi.....	50
BAB V KESIMPULAN DAN SARAN.....	51
5.1 Kesimpulan	51
5.2 Saran.....	51
DAFTAR PUSTAKA	52
LAMPIRAN	55

DAFTAR TABEL

Tabel 2.1 Fitur-fitur Android.....	11
Tabel 2.2 <i>Feature Extraction</i>	18
Tabel 3.1 Keterangan use case diagram.....	22
Tabel 3.2 Tabel identifikasi <i>use case diagram</i>	24
Tabel 3.3 Keterangan <i>activity diagram</i>	26
Tabel 4.1 Akurasi aplikasi terhadap gerakan jatuh.....	47

DAFTAR GAMBAR

Gambar 2.1 Koordinat sensor akselerometer.....	7
Gambar 2.2 Gerakan jatuh bebas.....	9
Gambar 2.3 Konsep dasar <i>Decision Tree</i>	14
Gambar 2.4 Algoritma <i>Random Forest</i>	15
Gambar 2.5 Ilustrasi data <i>preprocessing</i>	17
Gambar 2.6 Klasifikasi data <i>random forest</i>	19
Gambar 3.1 <i>Use Case Diagram</i> Sistem Deteksi Jatuh.....	23
Gambar 3.2 <i>Activity diagram</i> deteksi jatuh.....	28
Gambar 3.3 Rancangan Halaman Utama.....	29
Gambar 3.4 Skema Aplikasi Deteksi Jatuh.....	30
Gambar 3.5 Alur perancangan data deteksi jatuh.....	30
Gambar 3.6 Skema <i>10 fold cross validation</i>	31
Gambar 3.7 <i>Tree</i> pertama aplikasi deteksi jatuh.....	32
Gambar 3.8 <i>Tree</i> kedua aplikasi deteksi jatuh.....	32
Gambar 3.9 <i>Tree</i> ketiga aplikasi deteksi jatuh.....	33
Gambar 4.1 Kode program <i>Text to Speech</i>	35
Gambar 4.2 Kode program penyimpanan data.....	36
Gambar 4.3 Kode program fitur rata-rata.....	36
Gambar 4.4 Kode program standar deviasi.....	36
Gambar 4.5 Kode program entropi.....	37
Gambar 4.6 Halaman utama aplikasi.....	38
Gambar 4.7 Penamaan file di Aplikasi.....	39
Gambar 4.8 Penyimpanan file coba di folder AccFolder.....	40
Gambar 4.9 Penyimpanan file coba di folder Rata.....	40
Gambar 4.10 Penyimpanan file coba di folder Entropi.....	41
Gambar 4.11 Penyimpanan file coba di folder Deviasi.....	41
Gambar 4.12 Perubahan nilai sumbu sensor.....	46
Gambar 4.13 Perubahan nilai sumbu sensor setelah aksi.....	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jatuh merupakan masalah serius yang harus dihadapi seiring bertambahnya usia dan menurunnya kemampuan fisik manusia. Jatuh dapat menimpa siapa saja tak terkecuali anak-anak, remaja, orang dewasa apalagi lanjut usia. Setiap orang pasti pernah mengalami jatuh meskipun hanya sekali. Jatuh tidak dapat diprediksi tetapi dapat dideteksi. Dengan memanfaatkan teknologi saat ini, kita dapat membuat sebuah sistem yang dapat digunakan untuk mendeteksi kejadian jatuh. Dengan adanya sistem deteksi jatuh tersebut, tentunya dapat mengurangi resiko-resiko yang akan terjadi setelah jatuh karena kita dapat melakukan pertolongan lebih cepat terutama bagi lanjut usia. Bagi lanjut usia, tentunya pengawasan harus lebih ditingkatkan karena lanjut usia lebih besar mengalami kejadian jatuh. Pengawasan dapat dilakukan oleh keluarga maupun saudara tentunya. Tetapi tidak mungkin jika pengawasan dilakukan harus selalu dekat dengan lanjut usia. Oleh karena itu dengan kemajuan teknologi saat ini kita dapat melakukan pengawasan secara tidak langsung dengan memanfaatkan teknologi, yaitu dengan perangkat *smartphone* android. Dengan kemampuan komputasi yang relatif cepat dan kebanyakan orang memilikinya menjadikan *smartphone* salah satu cara untuk membuat sistem deteksi jatuh.

Perkembangan ilmu pengetahuan dan teknologi (IPTEK) saat ini begitu cepat, baik teknologi perangkat lunak maupun teknologi perangkat keras berlomba-lomba membuat sesuatu yang dapat digunakan untuk memenuhi kebutuhan manusia. Tidak heran lagi saat ini banyak sarana dan prasarana yang menggunakan teknologi tepat guna untuk mempermudah manusia dalam menjalankan kegiatannya. Karena teknologi tersebut mudah diakses dan digunakan dimana saja dan kapan saja. Salah satu perkembangan yang sangat pesat yaitu dunia aplikasi *smartphone* android. Banyak yang berlomba-lomba menciptakan aplikasi yang bertujuan untuk memudahkan pekerjaan manusia. Tetapi aplikasi-aplikasi yang dibuat kebanyakan adalah aplikasi game dan multimedia. Masih sedikit aplikasi dengan biaya murah yang dapat membantu manusia untuk mendeteksi suatu kejadian jatuh. Hal ini dapat kita lihat pada toko aplikasi *Playstore* yang kebanyakan aplikasi deteksi jatuh masih berbayar dan dengan harga yang mahal. Saat ini *smartphone* merupakan kebutuhan manusia. Banyak hal yang dapat dilakukan dengan menggunakan *smartphone*. Hanya menekan tombol pada

smartphone, banyak hal yang dapat diselesaikan. Karena telah menjadi bagian dari kebutuhan manusia, apabila *smartphone* tersebut terjatuh dan kemudian hilang, mungkin saja pekerjaan akan terganggu. Dari sekian banyaknya kejadian tersebut, maka dibuat sebuah aplikasi yang dapat memberitahu atau memberikan peringatan apabila *smartphone* android terjatuh dengan memanfaatkan sensor akselerometer yang terdapat pada *smartphone* android.

Sensor akselerometer adalah sebuah sensor yang digunakan untuk mengukur percepatan linear, mengukur dan pendeteksi getaran (vibrasi), dan mengukur percepatan akibat gravitasi (inklinasi). Sensor akselerometer mengukur percepatan akibat gerakan benda yang melekat padanya.(Iman Fahruzi, 2014). Sensor akselerometer ini dalam penggunaannya masih memerlukan banyak inovasi sehingga dapat menghasilkan banyak kegunaan. Dalam pemanfaatannya sensor akselerometer hanya digunakan untuk bermain game saja. Tetapi kenyataannya sensor ini dapat berfungsi lebih penting dan bermanfaat. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Dengan kelebihanannya itu android cepat terkenal di kalangan pengguna mobile phone. Android merupakan "Platform mobile pertama yang lengkap, Terbuka, dan Bebas". (Agus Joko, 2014). Dengan adanya platform terbuka yang siapa saja dapat mempergunakan, maka banyak yang berlomba-lomba membuat aplikasi android dengan memanfaatkan sensor akselerometer pada *smartphone*. Sensor ini dapat dimanfaatkan sebagai deteksi jatuh. Ketika ada pergerakan yang abnormal (jatuh), kita dapat memperoleh data dari hasil yang ditangkap oleh sensor, maka data tersebut dihitung dan dibandingkan dengan threshold yang ada. (Rakhman dkk., 2015). Sehingga dengan adanya sensor akselerometer dapat dibuat aplikasi deteksi jatuh pada *smartphone* android agar dapat diketahui apabila terjadi kejadian jatuh.

Salah satu contoh penerapan aplikasi deteksi jatuh ini adalah pada manula. Sebagian besar manula yang terjatuh tidak memiliki kemampuan untuk bangkit kembali tanpa mendapatkan bantuan dari orang lain. Ketidakmampuan untuk bangun kembali setelah jatuh dapat diakibatkan oleh cedera yang dialami. Apabila tidak ada orang yang membantu bangun untuk jangka waktu yang lama kemungkinan akan terjadi hal yang lebih fatal seperti dehidrasi, luka karena tekanan, atau cedera otot. Dengan adanya aplikasi deteksi jatuh ini, manula yang terjatuh dapat segera diketahui dan diberikan pertolongan dengan cepat.

1.2 Batasan Masalah

Sehubungan dengan keterbatasan yang dimiliki, baik dari segi waktu serta pemikiran maka penelitian ini dibatasi. Adapun ruang lingkup masalah yang terdiri dari :

- a. Algoritma yang digunakan adalah *decision tree*.
- b. Menggunakan Rapid miner sebagai *software* deteksi akurasi jatuh.
- c. Aplikasi harus berjalan secara aktif agar aplikasi dapat bekerja .

1.3 Rumusan Masalah

Berdasarkan penjelasan di atas, maka merumuskan masalah-masalah sebagai berikut :

- a. Masih sedikitnya aplikasi deteksi jatuh dengan biaya murah di pada android yang dapat mendeteksi apabila terjadi kejadian jatuh dengan mengeluarkan suara
- b. Membuat aplikasi deteksi jatuh dengan menggunakan algoritma *decision tree* dan *threshold*.

1.4 Tujuan Penelitian

Berdasarkan pada rumusan masalah yang telah dikemukakan, maka dapat dirumuskan tujuan dilaksanakannya penelitian ini adalah :

- a. Membuat sebuah sistem berbasis android yang dapat memberikan *alert* apabila terjadi kejadian jatuh.

1.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah sebagai berikut :

- a. Bagi penulis
 1. Dapat mengimplementasikan pemanfaatan sensor akselerometer pada ponsel pintar android.
 2. Dapat melakukan pengembangan aplikasi android yang sedang terkenal saat ini.
- b. Bagi Pengguna
 1. Menjadi tahu apabila terjadi kejadian jatuh karena mengeluarkan *alert* berupa suara dengan bahasa Indonesia jika akselerasi sensor akselerometer telah melewati *threshold*.

1.6 Metodologi Penelitian

Untuk menjawab rumusan masalah diatas, penelitian ini akan dilakukan dengan beberapa tahapan yaitu:

1. Melakukan pencarian data
Pada tahap ini dilakukan pencarian semua data yang dibutuhkan untuk membuat sistem deteksi jatuh. Data bisa didapat dari wawancara dan studi pustaka.

2. Perancangan sistem aplikasi android

Pada tahap ini dilakukan perancangan dan pembuatan sistem aplikasi android agar dapat mencatat data dari sensor akselerometer.

3. Menentukan threshold

Menentukan threshold dari data yang didapatkan pada sensor akselerometer dengan pengujian yang akan digunakan sebagai nilai minimal jatuh.

4. Membangun sistem deteksi jatuh

Pada tahap ini dilakukan pembuatan aplikasi deteksi jatuh ponsel pintar dengan menggunakan android studio berdasarkan data threshold yang sudah didapatkan melalui pengujian.

5. Pengujian

Sistem yang telah jadi akan diujikan untuk melihat apakah sistem deteksi jatuh sudah berjalan dengan baik.

1.7 Sistematika Penelitian

Untuk mempermudah pembacaan serta dapat memberikan gambaran secara menyeluruh terhadap masalah yang akan dibahas, maka sistematika penulisan laporan tugas akhir ini dibagi dalam lima bab. Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Pendahuluan membahas permasalahan umum tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan penelitian sistem deteksi jatuh berbasis android.

BAB II LANDASAN TEORI

Landasan teori ini berisi tentang teori-teori yang berhubungan dengan penelitian aplikasi sistem deteksi jatuh berbasis android, meliputi konsep deteksi jatuh, sensor akselerometer, sistem Android, dan *platform* yang digunakan dalam pembuatan sistem ini.

BAB III METODOLOGI

Metodologi memuat uraian tentang kebutuhan perangkat keras dan perangkat lunak serta kebutuhan masukan, keluaran, ataupun antarmuka sistem deteksi jatuh.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini membahas tentang sistem deteksi jatuh yang telah dibuat dan penjabaran dari BAB METODOLOGI dalam bentuk implementasi sistem. Implementasi yang dimaksud

meliputi implementasi perangkat lunak, implementasi perangkat keras dan implementasi antarmuka. Bab ini juga membahas pengujian dan analisis aplikasi sistem deteksi jatuh berbasis Android yang telah dibangun.

BAB V KESIMPULAN DAN SARAN

Dalam bab ini memuat kesimpulan dari penelitian aplikasi sistem deteksi jatuh berbasis android dan merupakan rangkuman dari analisis kinerja yang akan menemukan beberapa saran untuk dilaksanakan lebih lanjut guna pengembangan penelitian ini.

BAB II

LANDASAN TEORI

2.1 Sensor

Sensor adalah perangkat canggih yang sering digunakan untuk mendeteksi dan merespon sinyal listrik atau optik. Sensor mengubah parameter fisik misalnya suhu, kelembaban, tekanan darah, kecepatan, dan lain-lain. Parameter fisik ini diubah menjadi sinyal yang dapat diukur secara elektrik. Sensor memungkinkan pendeteksian, analisa, dan merekam fenomena fisis yang sulit dicerna dengan mengubahnya menjadi sinyal-sinyal yang mudah dimengerti. Sinyal yang dihasilkan oleh sensor setara dengan kuantitas yang akan diukur misalnya suhu. Dengan suhu sekian maka sensor akan memberikan data yang sesuai dengan panas yang akan diukur.

Sensor yang baik mempunyai kriteria kriteria seperti berikut ini.

- a. *High sensitivity* : sensitivitas menunjukkan berapa banyak output perangkat berubah dengan perubahan unit input (kuantitas yang akan diukur).
- b. *Linearity* : output harus berubah secara linear dengan input.
- c. *High Resolution* : perubahan terkecil dalam input yang bisa dideteksi oleh sensor
- d. *Less noise and disturbance* : menghasilkan suara yang tidak mengganggu bahkan tidak bersuara
- e. *Less power consumption* : konsumsi daya yang sedikit

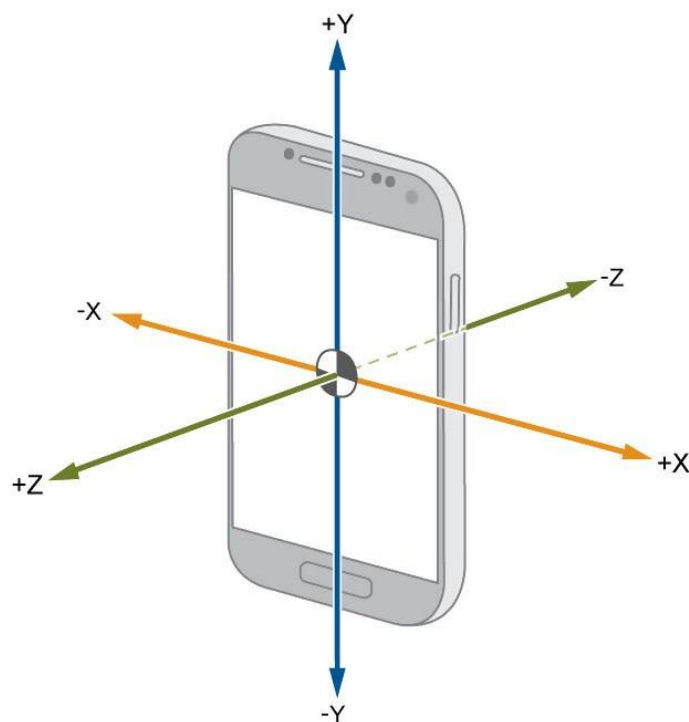
Ukuran dari sensor juga penting, sensor dengan ukuran yang lebih kecil banyak dicari karena lebih simple dalam penggunaannya, kepadatan sensor yang lebih tinggi, dan biaya yang lebih murah. Saat ini telah dibuat sensor yang lebih efektif. Sensor dibuat dengan penginderaan listrik yang sudah menjadi satu dengan sensor. Produk ini dinamakan MEMS. Contoh produk yang menggunakan MEMS saat ini adalah *inkjet printers*.(Liu, n.d.)

2.2 Sensor Accelerometer

Sensor *accelerometer* mengukur percepatan dari sensor dan apapun yang berhubungan dengan sensor seperti getaran. Sensor ini memiliki banyak penggunaan dan aplikasi. Penggunaannya mulai dari bidang transportasi, deteksi dini, dan keamanan. Penggunaan yang paling umum digunakan adalah untuk mengeluarkan *air bags* pada mobil jika terjadi kecelakaan. Jika usaha yang diterima melebihi threshold maka sensor akan bekerja secara otomatis.

Sensor *accelerometer* berfungsi untuk mengukur akselerasi dengan tepat. Akselerasi tepat yang diukur dengan *accelerometer* belum tentu memiliki ketepatan koordinat (laju perubahan velositas). Sebaliknya, *accelerometer* melihat akselerasi terkait dengan fenomena berat yang dialami oleh massa uji pada kerangka acuan perangkat *accelerometer*. Sebagai contoh, *accelerometer* di permukaan bumi akan mengukur akselerasi $g=9,81 \text{ m/s}^2$ lurus ke atas karena beratnya. Sebaliknya, *accelerometer* jatuh bebas atau di luar angkasa akan mengukur nol. Jenis akselerasi yang bisa diukur oleh *accelerometer* adalah akselerasi gaya gravitasi.(Riantana, 2015)

Pada ponsel pintar, sensor akselerometer memiliki tiga sumbu koordinat yaitu x,y, dan z. Sumbu x adalah sumbu dengan posisi horizontal dari ponsel pintar. Sumbu y adalah sumbu dengan posisi vertikal dari ponsel pintar. Sedangkan untuk sumbu z merupakan sumbu yang mengarah keluar dari layar ponsel pintar. Dalam sistem ini, koordinat pada layar belakang ponsel pintar memiliki nilai negatif z. Untuk lebih jelas nya ada pada gambar 2.1



Gambar 2.1 Koordinat sensor akselerometer

Accelerometer dibagi menjadi beberapa tipe karena setiap jenis sensor berbeda-beda sesuai dengan spesifikasi yang dikeluarkan oleh perusahaannya. Tipe-tipenya yaitu sebagai berikut ini.(Sinaga, 2013)

a. *Capacitive*

Lempengan metal pada sensor memproduksi sejumlah kapasitansi, kapasitansi akan mempengaruhi percepatan.

b. *Piezoelectric*

Kristal *piezoelectric* yang terdapat pada *accelerometer* jenis ini mengeluarkan tegangan yang selanjutnya dikonversi menjadi percepatan.

c. *Piezoresistive*

Lempengan yang secara resistan akan berubah sesuai dengan perubahan percepatan.

d. *Hall Effect*

Percepatan yang dirubah menjadi sinyal elektrik dengan cara mengukur setiap perubahan pergerakan yang terjadi pada daerah yang terinduksi magnet.

e. *Magnetoresistive*

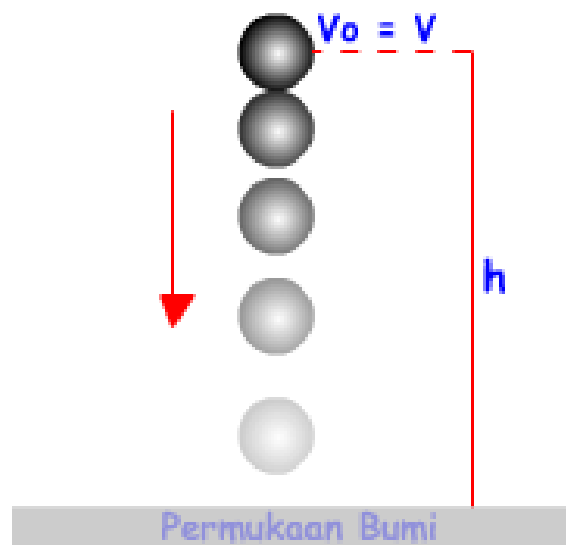
Perubahan percepatan diketahui berdasarkan resistivitas material karena adanya daerah yang terinduksi magnet.

f. *Heat Transfer*

Percepatan dapat diketahui dari lokasi sebuah benda yang dipanaskan dan diukur ketika terjadi percepatan dengan sensor temperatur.

2.3 Gerak Jatuh

Sebuah benda yang jatuh tentu akan menuju ke pusat bumi atau ke bawah. Hal ini dikarenakan adanya gaya gravitasi bumi yang menarik semua benda kearah bawah yang diberi notasi g . Percepatan yang dihasilkan dari gravitasi bumi adalah konstan. Benda yang dilemparkan vertical ke atas akan mengalami perlambatan tetap sebesar g . Gerak jatuh bebas ada yang diberi percepatan awal dengan notasi v_0 ataupun tidak. Jelas untuk melemparkan sebuah benda ke atas dibutuhkan kecepatan awal. Tetapi apabila benda jatuh ke bawah tidak harus memerlukan kecepatan awal misalnya saja benda jatuh karena dilepaskan dan memerlukan kecepatan awal apabila benda jatuh karena dibanting.(Sinaga & Situmorang, n.d.) Agar lebih jelas tentang gerakan jatuh bebas dapat dilihat pada gambar 2.2



Gambar 2.2 Gerakan jatuh bebas

Jatuh pada manusia sering terjadi pada lanjut usia karena dipengaruhi oleh beberapa faktor misalnya saja usia, kemampuan, dan lainnya. Tetapi faktor jatuh pada lansia dapat digolongkan dalam 2 bagian. (Iis Cholilah, 2009)

a. Faktor Intrinsik

Faktor intrinsik adalah faktor yang dapat disebabkan oleh beberapa proses seperti penyakit misalnya stroke, Parkinson, maupun depresi yang menyebabkan lansia dapat mengalami kejadian jatuh. Gangguan penglihatan juga dapat menyebabkan jatuh karena lanjut usia tidak fokus untuk melihat benda di sekitarnya.

b. Faktor Ekstrinsik

Faktor ekstrinsik adalah faktor yang lebih banyak dipengaruhi oleh benda atau alat yang ada di lingkungan. Alat-alat perlengkapan rumah tangga atau benda yang sudah tua usianya, benda yang tergeletak begitu saja, lantai yang licin dan tidak rata, penerangan yang kurang dan lainnya.

Pencegahan dapat dilakukan dengan berdasarkan resiko yang dapat menyebabkan jatuh pada lanjut usia seperti penyakit yang sedang dialami, pengobatan yang dijalani, atau dengan memperbaiki dari factor lingkungannya. Di bawah ini beberapa metode pencegahan jatuh :

a. Memperbaiki lingkungan

Untuk melakukan pencegahan dapat dilakukan dengan memperbaiki lingkungannya, artinya meletakkan barang sesuai dengan tempatnya dan menata barang yang sering dibutuhkan berada dekat dengan jangkauan. Selanjutnya

melakukan perbaikan prasarana seperti memasang lantai yang tidak licin, menggunakan karpet anti slip, atau menggunakan penerangan yang cukup.

b. Memperbaiki kebiasaan

Kebiasaan yang salah dapat meningkatkan resiko jatuh. Hal ini tentunya akan sangat membahayakan, oleh karena itu perlu untuk merubah kebiasaan tersebut. Misalnya, jangan mengangkat berat yang berat atau tidak sesuai dengan kemampuannya, mengambil barang dengan benar dari lantai, dan menghindari kegiatan fisik yang berlebihan.

c. Latihan fisik

Latihan fisik diperlukan tentunya untuk mengurangi resiko jatuh. Karena dengan latihan diharapkan dapat meningkatkan kekuatan pada otot sehingga lebih kuat. Latihan juga dapat meningkatkan keseimbangan tubuh dan koordinasi anggota badan.

d. Memperhatikan konsumsi obat

Obat juga dapat mempengaruhi kejadian jatuh, oleh karena itu perlu mengatur konsumsi obat dan memperhatikan efek samping obat tersebut. Jika bisa, menghentikan konsumsi obat yang sudah tidak terlalu diperlukan.

e. Alat bantu jalan

Menggunakan alat bantu jalan dapat mengurangi resiko jatuh, tetapi tetap harus berhati-hati dalam penggunaannya. Pilih alat bantu jalan yang memudahkan dan kuat. Alat bantu jalan dapat membantu meningkatkan keseimbangan, namun pengguna harus membungkuk oleh karena itu harus ada rekomendasi dari pengguna terlebih dahulu sebelum menggunakan alat bantu jalan.

2.4 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan computer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007.(Jubilee, 2015)

Android dapat dioperasikan dengan menggesek, mengetuk, mencubit pada perangkat yang telah dipasang sistem operasi android ini. Android adalah system operasi yang *open source* sehingga siapa saja dapat mengembangkan dan memodifikasi secara bebas sistem

operasi ini. Agar lebih mengetahui tentang fitur android dapat dilihat pada tabel 2.1 (Jubilee, 2015)

Tabel 2.1 Fitur-fitur Android

FITUR	DESKRIPSI
antarmuka	Layar system operasi android menampilkan antarmuka yang bagus dan intuitif
konektifitas	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-fi, LTE, NFC dan Wimax
penyimpanan	SQLite, database relasional ringan yang digunakan untuk penyimpanan data
media	H.263, H.264, MPEG-4, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg vorbis, WAV, JPEG, PNG, GIF, BMP
messaging	SMS dan MMS
Web Browser	Berdasarkan layout engine Webkit, dengan Chrome's V8 JavaScript engine yang mendukung HTML5 dan CSS3
multi touch	Android memiliki dukungan multi touch (lebih dari satu sentuhan layar)
Multi tasking	Pengguna dapat berpindah dari satu task ke task yang lainnya dan beberapa aplikasi dapat berjalan pada waktu yang bersamaan
Ukuran widgets	Widgets dapat diatur ukurannya sesuai keinginan pengguna, baik memperbesar ukuran untuk menampilkan lebih banyak konten, atau memperkecil ukuran untuk menyediakan lebih banyak ruang
Multi language	Mendukung teks dalam berbagai bahasa
GCM	Google Cloud Massaging (GCM) adalah layanan yang memungkinkan developer untuk mengirim data pesan pendek kepada

	pengguna tanpa menggunakan solusi sync proprietary
Wifi direct	Teknologi yang memungkinkan aplikasi untuk menemukan dan memasang secara langsung melalui koneksi peer-to-peer high-bandwidth
Android beam	Teknologi berbasis NFC populer yang memungkinkan pengguna untuk berbag konten secara langsung hanya dengan menyentuhkan dua perangkat android

Pada sistem operasi android memiliki empat komponen aplikasi. Keempat komponen tersebut adalah sebagai berikut.(Alicia, 2013)

a. Activity

Activity akan menyajikan sebuah *user interface* sehingga dapat melakukan interaksi dengan android. *Activity* merupakan komponen yang paling penting dalam sistem operasi android atau aplikasi karena menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya akan ada *activity* yang pertama kali tampil jika aplikasi dijalankan.

b. Service

Service adalah komponen yang berjalan di *background/backend* melakukan operasi (*long-running operation*) atau melakukan pekerjaan untuk *remote process*. Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen ini dalam proses tidak terlihat, memperbarui sumber data dan menampilkan pemberitahuan. *Service* digunakan untuk melakukan pengolahan data yang perlu diproses.

c. Content Provider

Komponen ini berguna untuk membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system*, seperti *database SQLite*. Konten ini juga menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

d. Broadcast Receiver

Komponen ini berfungsi untuk menerima dan menyampaikan pemberitahuan, yang sebagian besar berasal dari system. Misalnya zona waktu yang berubah, baterai lemah, gambar telah selesai diambil oleh kamera, atau perubahan referensi Bahasa yang digunakan.

Android memungkinkan penggunaannya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti *Google play*, *Amazon Appstore*, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. (Rahadi, 2014)

Aplikasi Android dikembangkan dengan Bahasa pemrograman java yang sudah sangat terkenal saat ini dan dengan pengembangan perangkat lunak Android atau *software development kit* (SDK). Android SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan Bahasa pemrograman *java*. (Sinsuw & Najooan, 2013) Beberapa fitur android yang dianggap penting adalah mesin virtual Dalvik yang dioptimalkan untuk perangkat mobile, grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl*, kemudian SQLite yang digunakan untuk penyimpanan basis data (*database*). SDK ini terdiri dari seperangkat perangkat pengembangan, termasuk *debugger*, emulator *handset* berbasis QEMU, dokumentasi, kode sampel, dan tutorial. (Rahadi, 2014)

2.5 Java

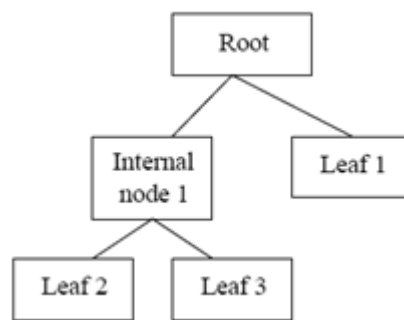
Java merupakan suatu teknologi perangkat lunak yang di dalamnya mencakup Bahasa pemrograman. Bahasa pemrograman *java*, pertama kali diaplikasikan oleh Sun Microsystems sebagai Bahasa yang mendukung program *utilities*, permainan, dan aplikasi bisnis. Hal tersebut dikarenakan *java* merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjelaskan suatu program. (Alicia, 2013) Seperti diketahui *java* adalah Bahasa pemrograman yang kode programnya dikompilasi dan diinterpretasi. (Prasetyo, 2007) Kompilasi program java dilakukan menggunakan *tool command-line* yang bernama *javac*. Atau biasa disebut *compiler java*.

Bahasa pemrograman *java* tentunya memiliki tipe data. Tipe data dalam java dikelompokkan dalam dua jenis, yaitu tipe data primitif dan kelas. Tipe primitif/dasar adalah tipe-tipe bawaan, seperti *Boolean*, *char*, *byte*, *short*, *int*, *long*, *float*, dan *double*. Sedangkan tipe data kelas meliputi *interface* dan *array*.

Java sendiri merupakan OOP (*object oriented programming*) (Alicia, 2013), maksudnya adalah semua aspek yang terdapat pada *java* adalah objek sehingga sangat memudahkan untuk mendesain, membuat, dan mengembangkan program *java* dengan cepat.

2.6 Decision Tree

Decision Tree atau yang lebih sering dikenal dengan pohon keputusan merupakan sebuah algoritma yang digunakan untuk memudahkan dalam mengambil keputusan. *Decision Tree* adalah pohon dimana setiap cabangnya menunjukkan pilihan diantara sejumlah alternatif pilihan yang ada, dan setiap daunnya menunjukkan keputusan yang dipilih. (Setiawati, Taufik, Jumadi, & Zulfikar, 2016) *Decision Tree* ini bertujuan untuk pengambilan sebuah keputusan. Kelebihan algoritma ini adalah dapat melakukan pengambilan sebuah keputusan yang sebelumnya sangat kompleks menjadi lebih mudah dan simpel. Algoritma *Decision Tree* dimulai dengan sebuah *root* yang digunakan untuk mengambil sebuah aksi atau tindakan. Selanjutnya diikuti daun atau *leaf* di bawahnya yang setiap cabangnya menunjukkan kemungkinan skenario dari tindakan. Untuk lebih jelasnya algoritma *Decision Tree* dapat dilihat pada gambar 2.3



Gambar 2.3 Konsep dasar *Decision Tree*

Untuk membentuk sebuah tree, C4.5 menggunakan kriteria split yang telah dimodifikasi sebelumnya yang dinamakan *Gain Ratio* oleh Mitchael (1997). Split atribut merupakan proses yang utama dalam pembentukan sebuah tree. Adapun tahapan-tahapan untuk membentuk sebuah tree adalah, pertama menghitung nilai *entropy*, selanjutnya adalah menghitung nilai *Gain Ratio* masing-masing atribut dan untuk nilai yang paling tinggi dari *Gain Ratio* dipilih sebagai akar/*root*, sedangkan untuk nilai yang lebih rendah dipilih menjadi cabangnya. Selanjutnya menghitung lagi *Gain Ratio* dengan tidak mengikutsertakan nilai *Gain* yang telah menjadi root. Langkah terakhir adalah mengulangi langkah menghitung *Gain* sampai didapatkan nilai *Gain* sebesar 0.

2.7 Ensemble dan Bootstrap Aggregating

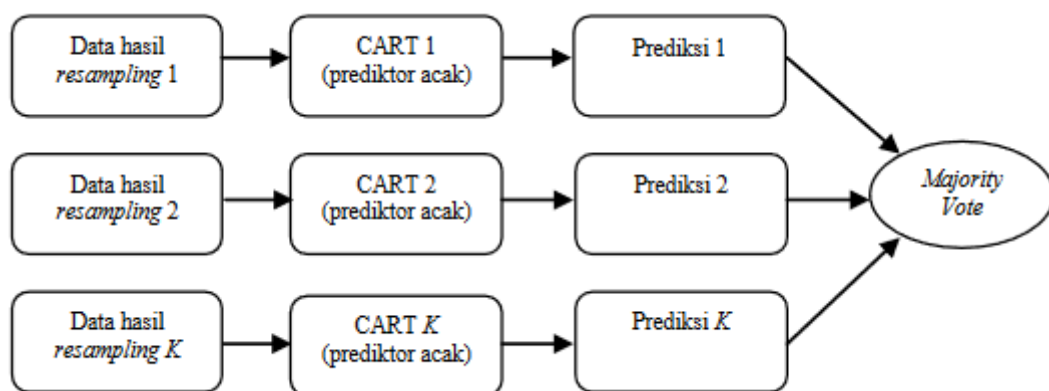
Metode *Ensemble* merupakan metode dengan ide melakukan kombinasi dari banyak *classifier* tunggal dimana hasil prediksi masing-masing *classifier* digabungkan menjadi prediksi akhir melalui proses voting mayoritas. (Muttaqin, Otok, & Hakim, n.d.) Lebih mudahnya adalah metode yang menggabungkan beberapa *tree* menjadi satu *tree* sehingga

lebih simpel. Ada dua metode *Ensemble* yang sebelumnya lebih populer yaitu *bootstrap aggregating* dan *boosting*, sedangkan metode yang dikembangkan adalah *Random Forest*.

Bagging (Bootstrap Aggregating) awalnya dikembangkan oleh Breiman pada tahun 1996 sedangkan *Boosting* dikembangkan oleh Freund dan Schepire. Jika dibandingkan, metode *bagging* lebih akurat jika dibandingkan dengan metode *boosting* walaupun metode *bagging* tidak selalu lebih akurat. Metode *bagging* dapat membantu mengurangi ketidakstabilan pada klasifikasi. Metode *bootstrap aggregating* ini baik digunakan pada regresi dan klasifikasi. Adapun langkah-langkah untuk membuat *tree* dengan metode *bagging* adalah, pertama Tarik contoh acak berukuran n dari data training, kemudian menyusun pohon yang terbaik berdasarkan data tersebut, dan langkah yang terakhir adalah mengulangi langkah pertama dan kedua sehingga didapatkan pohon sebanyak k buah pohon yang acak. Untuk mendapatkan satu buah *tree* dilakukan penggabungan dari beberapa *tree* yang telah dibuat sebelumnya menggunakan metode *majority vote* yang digunakan untuk kasus klasifikasi dan rata-rata untuk kasus regresi. (Sartono & Syafitri, n.d.)

2.8 Random Forest

Metode *Random Forest* banyak sekali digunakan untuk klasifikasi. Metode ini adalah pengembangan dari metode *classification and regression tree (CART)*. *Random Forest* bekerja dengan menerapkan metode *bootstrap aggregating* yang sudah dijelaskan sebelumnya. Pada *Random Forest* proses pengacakan untuk membentuk pohon klasifikasi tidak hanya dilakukan untuk data sampel saja melainkan juga pada pengambilan variabel prediktor sehingga akan menghasilkan kumpulan pohon yang berbeda-beda. (Dewi, Syafitri, & Mulyadi, 2011) Untuk lebih jelasnya algoritma *Random Forest* dapat dilihat pada gambar 2.4



Gambar 2.4 Algoritma *Random Forest*

Metode *Random Forest* dilakukan dengan cara seperti berikut ini.(Dewi dkk., 2011) :

1. Melakukan penarikan contoh acak berukuran n . Tahap ini merupakan tahap *bootstrap*
2. Dengan menggunakan contoh *bootstrap*, pohon dibangun sampai mencapai ukuran maksimum. Tahap ini dinamakan *random feature selection*
3. Langkah terakhir adalah mengulangi langkah 1 dan 2 sebanyak k kali, sehingga terbentuk sebuah hutan yang terdiri atas k pohon.

Pada metode *Random Forest* untuk menentukan hasil prediksi akhir dengan menggunakan *majority vote* atau suara yang paling banyak dari beberapa pohon yang telah dibuat sebelumnya.

2.9 Fitur

Dalam mendeteksi jatuh tentunya terdapat fitur-fitur yang digunakan untuk melakukan sebuah pengukuran untuk menentukan *threshold*. Fitur yang dimaksud adalah rumus-rumus yang digunakan untuk melakukan penentuan jatuh. Sistem tidak dapat menentukan jatuh hanya dengan memanfaatkan sumbu x, y , dan z saja pada sensor akselerometer. Tentunya disini fitur akan sangat bermanfaat dan penting agar aplikasi jatuh dapat lebih akurat. Terdapat beberapa fitur yang digunakan untuk melakukan perhitungan jatuh yaitu rata-rata, standar deviasi, dan entropi.

Fitur rata-rata, salah satu fitur yang digunakan untuk membuat sebuah *tree* pada perhitungan aplikasi deteksi jatuh ini. Fitur ini dibuat berdasarkan rata-rata per 10 data yang dihasilkan oleh sensor akselerometer. Artinya setiap 10 data yang dihasilkan oleh pergerakan sensor akselerometer terdapat sebuah perhitungan rata-rata.

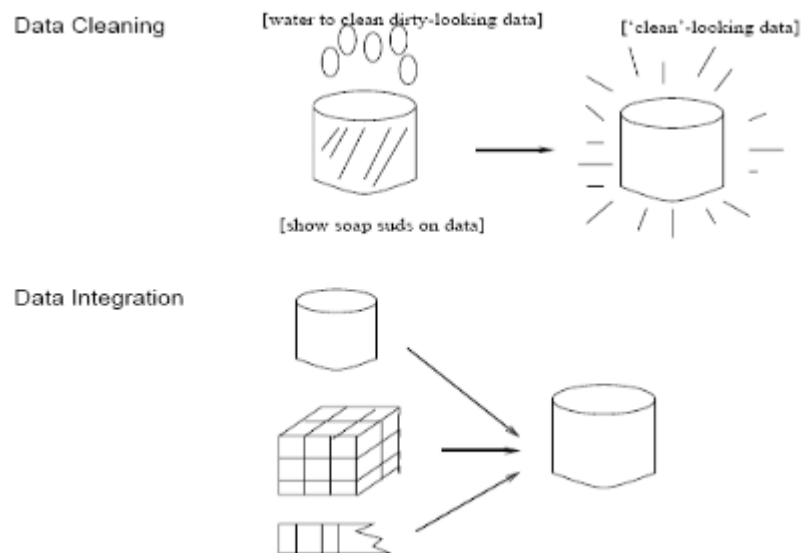
Selanjutnya adalah fitur standar deviasi, fitur ini digunakan untuk melihat banyaknya sebaran data yang dihasilkan oleh sensor akselerometer yang kemudian diolah menjadi salah satu komponen untuk menghasilkan *tree* .

Fitur yang terakhir digunakan adalah entropi, entropi digunakan untuk mengukur ketidakpastian data yang dihasilkan dari sensor akselerometer. Entropi berperan untuk menentukan suatu percabangan dari sebuah *decision tree*. Nilai pada entropi juga bertingkat, maksudnya apabila nilai entropi pada suatu atribut rendah maka, atribut tersebut yang akan menjadi akar/*root* pada sebuah *decision tree*.

2.10 Data Preprocessing

Data *preprocessing* merupakan salah satu tahapan yang harus dilakukan sebelum melakukan penghitungan sebuah data. *Preprocessing* dilakukan agar data yang akan dihitung

menjadi bersih. Bersih artinya banyak data yang dihasilkan masih belum baik. Penyebab data belum baik sehingga tidak bisa diolah adalah adanya kekurangan nilai atribut, banyak penyimpangan data yang dihasilkan, dan ketidakcocokan data yang digunakan. Untuk lebih jelasnya dapat dilihat pada gambar 2.5 berikut ini.



Gambar 2.5 Ilustrasi data *preprocessing*

2.11 Feature Extraction

Ekstraksi fitur adalah proses yang dilakukan dalam tahapan untuk mendapatkan suatu data atau atribut yang berpengaruh terhadap sebuah pemodelan dan analisa data. Untuk melakukan ekstraksi fitur terdapat beberapa cara, secara garis besar terdapat 2 cara yaitu *ranking selection* dan *subset selection*. *Ranking selection* tentunya memberikan *ranking* pada fitur yang dianggap sangat berpengaruh dalam membuat pemodelan data dan tidak menggunakan fitur yang dianggap tidak berpengaruh. *Ranking Ranking* yang tertinggi tentunya merupakan fitur yang paling berpengaruh dan *ranking* di bawahnya merupakan fitur yang kurang berpengaruh. Selanjutnya *subset selection*, yaitu dengan melakukan pencarian set dari fitur yang dianggap sebagai optimal fitur. Terdapat beberapa metode yang dapat digunakan untuk melakukan *subset selection* yaitu dengan metode *wrapper*, *filter*, dan *embedded*.

Dalam aplikasi jatuh ini, fitur-fitur yang digunakan ada 3 yaitu rata-rata, standar deviasi, dan entropi. Masing-masing dari fitur tersebut menghasilkan data yang kemudian diolah dan dijadikan suatu model untuk melakukan perhitungan penentuan jatuh. Agar lebih jelas fitur-fiturnya dapat dilihat pada tabel 2.2 berikut ini.

Tabel 2.2 *Feature Extraction*

Timestamp	Feature	X	Y	Z	Activity
1537409259	Rata-rata	0.625	9.489	2.127	walking
1537409261	Standar deviasi	4.243	63.631	14.527	walking
1537409263	Entropi	0.065	9.493	1.152	walking

2.12 *Feature Selection*

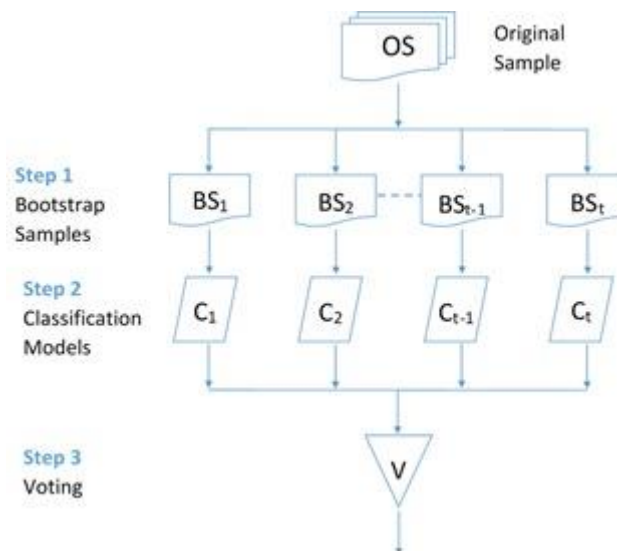
Feature selection dan *feature extraction* sangat berkaitan erat. Seleksi fitur digunakan untuk menentukan ekstraksi fitur seperti bagian yang sudah dijelaskan sebelumnya. Memilih fitur didasarkan pada bobot atau ranking yang paling tinggi yang kemudian digunakan untuk membuat pemodelan. Skema yang digunakan pada aplikasi deteksi jatuh ini adalah dengan *random forest*, maka setiap node akan dikunjungi. Karena setiap node pohon dikunjungi, maka semua pohon akan digunakan. Untuk menentukan *feature selection* terdapat 2 parameter yang dapat digunakan yaitu, *criterion* dan *normalize weights*. Parameter *criterion* menentukan kriteria yang akan dijadikan pembobotan atribut. Kriteria tersebut contohnya *information gain*, *gain ratio*, dan akurasi. Sedangkan untuk parameter *normalize weights* didasarkan pada nilai bobot, jika bobot disetel menjadi true, maka semua bobot harus dinormalisasi dari rentang 0 sampai 1.

2.13 *Classification*

Classification atau penggolongan pada dasarnya adalah mengelompokkan pohon-pohon keputusan yang sudah dibuat sebelumnya. Selanjutnya *decision tree* dipilih agar mendapatkan suatu hasil prediksi akhir. Dengan menggunakan *classification random forest* dapat menyelesaikan masalah tugas klasifikasi dan regresi. Metode *random forest* digunakan untuk mencapai akurasi yang lebih baik dari metode lain dengan memanfaatkan sekelompok klasifikasi dan juga akan didapatkan hasil stabilitas yang lebih baik. Data dengan banyak suara atau sumber dapat mengarah ke klasifikasi yang tidak stabil.

Pada metode *random forest* terdapat 2 metode yang sering digunakan adalah pilihan acak fitur dan metode bagging (*bootstrap aggregating*). Tujuan dari kedua metode tersebut

adalah sama yaitu untuk mengurangi varian prediksi. Berikut ini adalah gambaran cara kerja klasifikasi data dengan metode *random forest* yang dapat dilihat pada gambar 2.6 berikut ini.



Gambar 2.6 Klasifikasi data *random forest*

Pada bagian paling atas dapat diasumsikan bahwa jumlah *record* data adalah M . Selanjutnya *bagging* bekerja dengan sampel M secara acak. Kemudian melakukan *bagging* untuk membuat data *training* dari beberapa pohon. Mengingat karena fitur yang digunakan adalah N , maka dipilih n variabel secara acak dari sub-sampel yang dibuat sebelumnya. Langkah selanjutnya adalah membuat pohon dengan menggunakan data dari sub-sampel dengan membagi *node* dengan *split* yang terbaik, dan membuat pohon tanpa pemangkasan. Langkah yang terakhir adalah melakukan klasifikasi pohon dengan melihat dari *majority vote* atau suara terbanyak yang akan dijadikan hasil dari *random forest*.

BAB III

METODOLOGI

3.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan di mana dilakukan proses pengumpulan data dan informasi yang akan digunakan dalam pembuatan sistem aplikasi yang akan dibuat. Metode pengumpulan data yang dilakukan adalah dengan metode studi pustaka yaitu melakukan pencarian data melalui buku, jurnal, paper, atau internet. Studi pustaka dilakukan dengan mencari literatur dari internet yang akan digunakan untuk membuat sistem deteksi jatuh, mencari informasi perangkat apa saja yang dibutuhkan dalam pembuatan aplikasi agar dapat berjalan seperti yang diharapkan.

3.1.1 Analisis Kebutuhan Fungsi

Analisis kebutuhan fungsi adalah tahapan dimana dilakukan penetapan fungsi yang ada pada aplikasi yang nantinya dapat dilakukan oleh aplikasi. Aplikasi deteksi jatuh ini nantinya akan memiliki fungsi seperti berikut :

- a. Memberikan *alert* apabila terjadi kejadian jatuh.
- b. Menyimpan data dari sensor akselerometer sehingga dapat dilakukan pengembangan kedepannya.

3.1.2 Analisis Kebutuhan Masukan

Pada tahap analisis kebutuhan masukan dilakukan tahapan untuk menentukan masukan yang dibutuhkan untuk membuat sistem deteksi jatuh ini.

- a. Data dari sensor akselerometer yang nantinya akan digunakan untuk menentukan atau membuat *threshold* yang digunakan sebagai ukuran kapan *alert* akan berbunyi apabila terjadi kejadian terjatuh

3.1.3 Analisis Kebutuhan Keluaran

Pada penelitian ini akan dibuat sebuah sistem yang memberikan *alert* kepada pengguna ponsel pintar. Kebutuhan keluaran dalam penelitian ini adalah sebagai berikut.

- a. Suara yang keluar dari ponsel pintar
- b. Data yang didapat dari pergerakan sensor akselerometer

3.1.4 Analisis Kebutuhan Perangkat Lunak

Penelitian ini menggunakan perangkat lunak dalam pembuatannya dan untuk penunjang kinerja sistem. Perangkat lunak yang digunakan untuk pembuatan sistem ini adalah Android Studio.

3.1.5 Analisis Kebutuhan Perangkat Keras

Penelitian ini menggunakan perangkat keras yang digunakan untuk pengujian dan pemasangan aplikasi deteksi jatuh. Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

- a. Ponsel Pintar Android
- b. Kabel usb type c
- c. Laptop
- d. Sensor akselerometer

3.2 Metode Perancangan

Metode perancangan merupakan suatu cara yang digunakan untuk menjelaskan perancangan dari suatu penelitian. Tahapan ini merupakan tahapan penting sebelum melanjutkan ke tahap pengerjaan penelitian. Pada tahap ini akan didapatkan metode yang digunakan untuk penelitian, sehingga dapat mengevaluasi dan identifikasi apabila terdapat kendala dalam pengembangan aplikasi.

3.2.1 Metode Perancangan Aplikasi

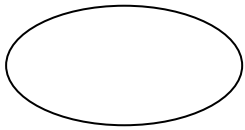
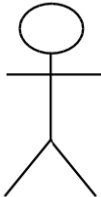


Aplikasi deteksi jatuh pada ponsel pintar ini memiliki keluaran berupa suara. Aplikasi ini juga dapat menyimpan data yang dihasilkan dari pergerakan sensor akselerometer pada ponsel pintar di memori ponsel pintar. *Alert* suara akan keluar apabila terjadi kejadian terjatuh dan telah melebihi *threshold* yang telah ditentukan. *Alert* suara ini sangat dipengaruhi oleh posisi jatuh dan kecepatan jatuh, jika posisi jatuh dan kecepatan jatuh tidak memenuhi syarat dan *threshold* yang telah ditentukan maka suara *alert* dari ponsel pintar tidak akan berbunyi. Karena ponsel mengeluarkan suara saat terjadi kejadian jatuh, agar lebih jelas untuk mendengarkan suara yang keluar maka dapat menggunakan *bluetooth* agar suara lebih terdengar jelas. Sedangkan untuk data akselerometer yang akan disimpan, sebelumnya pengguna harus memasukkan nama file pada halaman utama aplikasi. Pengguna dapat memasukkan nama apa saja karena nama tersebut hanya digunakan untuk nama file hasil dari pergerakan sensor akselerometer yang disimpan di dalam memori ponsel pintar sehingga

tidak mempengaruhi jalannya program dan selanjutnya menekan tombol mulai untuk memulai aplikasi. Aplikasi ini dapat berjalan apabila aplikasi di *minimize* sehingga pengguna tidak perlu selalu membuka aplikasi ini. Selanjutnya untuk menghentikan aplikasi pengguna dapat menekan tombol stop pada halaman utama aplikasi dan secara otomatis data pergerakan dari sensor akselerometer akan tersimpan di dalam memori ponsel pintar. Pengguna dapat juga menonaktifkan fitur penyimpanan sensor aplikasi jika memori ponsel terbatas, yaitu dengan mematikan penyimpanan aplikasi melalui perizinan aplikasi yang ada di sistem ponsel pintar android.

3.2.2 Use Case Diagram

Use Case Diagram adalah sebuah diagram yang menjelaskan dan menggambarkan fungsi suatu sistem. *Use Case Diagram* seringkali digunakan untuk menggambarkan bagaimana suatu sistem bekerja, sehingga pembuatan *Use Case Diagram* lebih kepada fungsi suatu sistem, bukan berdasarkan pada alur proses sistem. Adapun symbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada tabel 3.1 (Hendini, 2016)

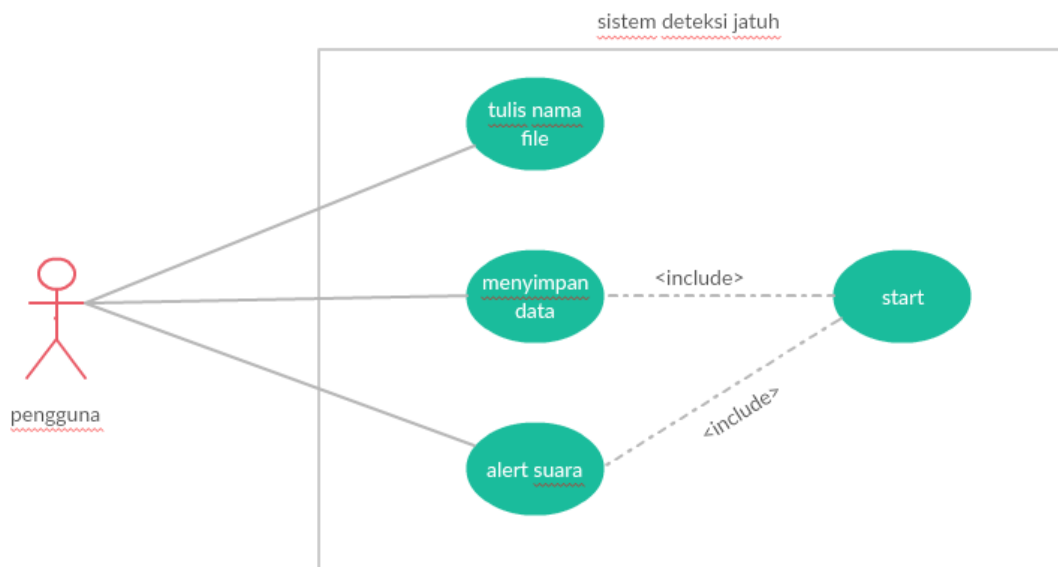
Tabel 3.1 Keterangan *use case diagram*

Gambar	Keterangan
	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan antar unit dengan actor, yang dinyatakan dengan menggunakan kata kerja
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara

	pasif dengan sistem
-----	Include, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain
----->	Extend, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

a. Use Case Diagram Sistem Deteksi Jatuh

Pada sistem deteksi jatuh hanya ada satu aktor yang berinteraksi dengan sistem yaitu pengguna ponsel pintar. Pengguna memiliki semua akses untuk menjalankan aplikasi dan menggunakan fungsi yang ada. Pengguna dapat melakukan penyimpanan data dan menghentikan aplikasi deteksi jatuh. *Use Case Diagram* sistem deteksi jatuh dapat dilihat pada Gambar 3.1



Gambar 3.1 *Use Case Diagram* Sistem Deteksi Jatuh

Pada Gambar 3.1 terdapat 1 aktor yaitu pengguna ponsel pintar dan 4 *case* yaitu menulis nama file, menyimpan data, dan mengeluarkan alert berupa suara. Dari gambar di atas dapat dilihat interaksi antara pengguna dengan sistem.

b. Identifikasi Use Case Diagram

Terdapat empat case yang telah dipetakan oleh kebutuhan fungsionalitas perangkat lunak sistem deteksi jatuh. Keterangan *use case diagram* dari sistem terdapat pada tabel 3.2

Tabel 3.2 Tabel identifikasi *use case diagram*

Kode	Nama & Deskripsi
SDJ01	<p><u>Tulis Nama File</u></p> <p>Deskripsi :Pengguna bisa menuliskan nama file yang akan digunakan untuk menyimpan data. Nama file dapat ditulis bebas berdasarkan kebutuhan dan tidak berpengaruh terhadap jalannya aplikasi</p>
SDJ02	<p><u>Menyimpan data</u></p> <p>Deskripsi : Pengguna dapat melakukan penyimpanan data pada aplikasi deteksi jatuh di ponsel pintar android. Proses penyimpanan data berlangsung secara otomatis pada saat sensor membaca pergerakan yang ada pada ponsel pintar dan langsung akan menyimpannya ke dalam memori ponsel.</p>
SDJ03	<p><u>Alert suara</u></p> <p>Deskripsi : Pengguna dapat mendengarkan <i>alert</i> berupa suara saat terjadi kejadian terjatuh. Agar suara lebih jelas pengguna dapat menggunakan <i>bluetooth</i> untuk menghubungkan suara dengan perangkat audio.</p>
SDJ04	<p><u>Start</u></p> <p>Deskripsi : Pengguna harus menekan tombol start dan mengisi file untuk menjalankan aplikasi</p>

c. Skenario *use case diagram*

1. SDJ01 Tulis nama file

Aktor : pengguna

Prerequisite : start

- a. Sistem menampilkan *user interface* awal dari sistem
- b. Pengguna mengisikan nama file

2. SDJ02 Menyimpan data

Aktor : pengguna

Prerequisite : start

- a. Sistem menyimpan data hasil akselerometer pada folder AccFolder, entropi, Rata, Deviasi
- b. Data disimpan dalam file .txt

3. SDJ03 Alert suara

Aktor : pengguna

Prerequisite : start

- a. Sistem akan mengeluarkan suara dari ponsel pintar apabila terjadi kejadian jatuh

4. SDJ04 Start

Aktor : pengguna

Prerequisite : -




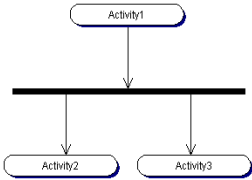
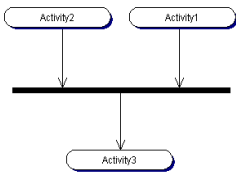
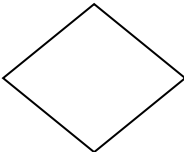
- a. Menampilkan halaman awal dari aplikasi
- b. Sistem memberikan respon apabila nama file belum diisi
- c. Pengguna menekan *start* dan aplikasi akan berjalan

3.2.3 Activity Diagram

Activity diagram adalah representasi grafis dari semua proses alur kerja sistem. Diagram ini memiliki pilihan tindakan, aktivitas, hasil, dan perulangan dari aktivitas tersebut. *Activity diagram* dapat digunakan untuk menjelaskan proses bisnis atau alur kerja yang ada pada sebuah sistem secara urut. *Activity diagram* dibutuhkan untuk menjelaskan semua fungsi

yang ada pada suatu aplikasi. Simbol-simbol pada *activity diagram* dapat dilihat pada tabel 3.3 (Hendini, 2016)

Tabel 3.3 Keterangan *activity diagram*

Gambar	Keterangan
	<p><i>Start point</i>, diletakkan di pojok kiri atas dan merupakan awal aktivitas</p>
	<p><i>End point</i>, akhir dari aktivitas</p>
	<p><i>Activities</i>, menggambarkan proses atau aktivitas bisnis</p>
	<p><i>Fork</i>/percabangan, menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan menjadi satu</p>
	<p><i>Join</i>/penggabungan, digunakan untuk menunjukkan adanya dekomposisi</p>
	<p><i>Decision</i>, menggambarkan pilihan untuk pengambilan keputusan. <i>True</i> atau <i>False</i></p>

Activity diagram menjelaskan langkah aktivitas dari setiap *case* atau pekerjaan yang terdapat pada *use case diagram* untuk menjabarkan semua fungsi-fungsi yang ada pada suatu sistem. *Activity diagram* juga menjelaskan alur kerja atau proses bisnis suatu sistem, mulai dari awal hingga aplikasi tersebut telah selesai berjalan atau bekerja. Pada *use case diagram* yang telah dijelaskan sebelumnya, terdapat beberapa *case* yang dapat dilakukan oleh pengguna aplikasi. Berikut ini penjabaran dari *use case diagram* di atas menggunakan *activity diagram*.

a. Tulis nama file

Pertama kali yang dilakukan adalah pengguna membuka aplikasi deteksi jatuh, setelah terbuka terdapat form untuk menuliskan nama *file* yang akan disimpan. Form ini harus diisi apabila tidak diisi atau kosong aplikasi deteksi jatuh tidak akan bisa berjalan.

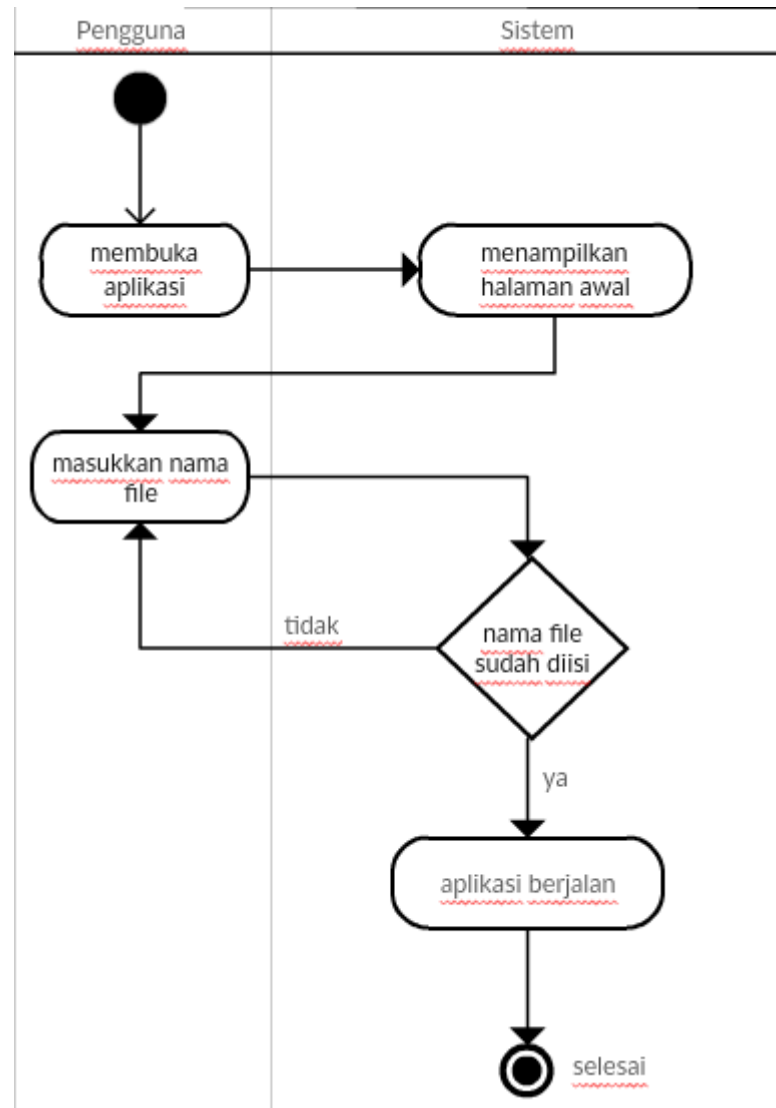
b. Menyimpan data

Setelah mengisi form nama file selanjutnya pengguna menekan tombol start untuk memulai program. Pada saat program berjalan, secara otomatis data yang dihasilkan dari sensor akselerometer akan disimpan ke dalam memori dalam format .txt. Untuk melihat hasil datanya, pengguna dapat melihat di dalam AccFolder yang telah dibuat di dalam aplikasi.

c. Alert suara

Pada *case* ini apabila ponsel pintar terjatuh dan nilai jatuh melebihi *threshold* yang telah ditentukan maka secara otomatis ponsel pintar akan mengeluarkan *alert* bahwa telah terjadi kejadian terjatuh. *Alert* ini tidak dapat dikontrol oleh pengguna karena secara otomatis akan keluar dengan sendirinya.

Agar lebih jelas mengenai aktivitas-aktivitas yang terjadi pada aplikasi deteksi jatuh yang telah dibuat, dapat dilihat pada *activity diagram* yang telah dibuat pada Gambar 3.2 berikut ini.

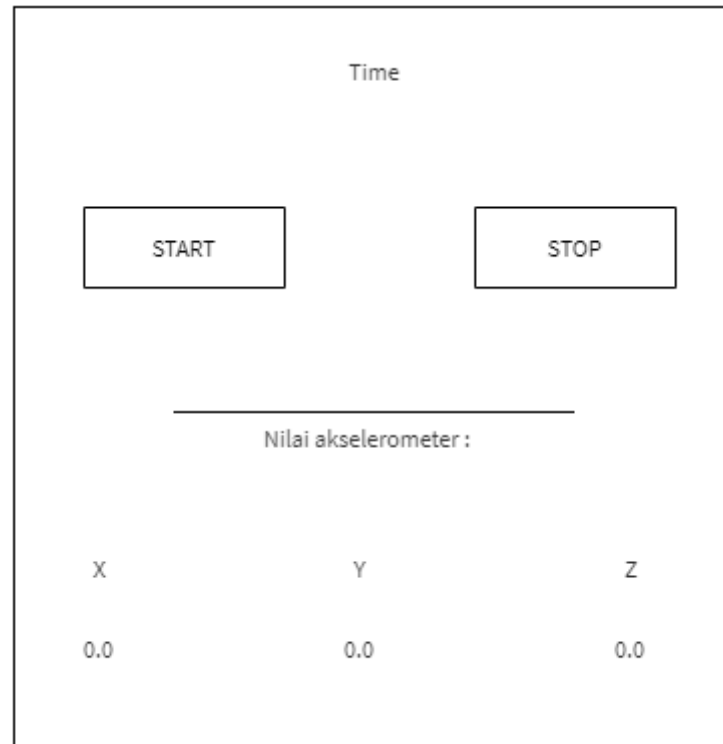


Gambar 3.2 Activity diagram deteksi jatuh

3.2.4 Perancangan Antarmuka

a. Halaman utama aplikasi

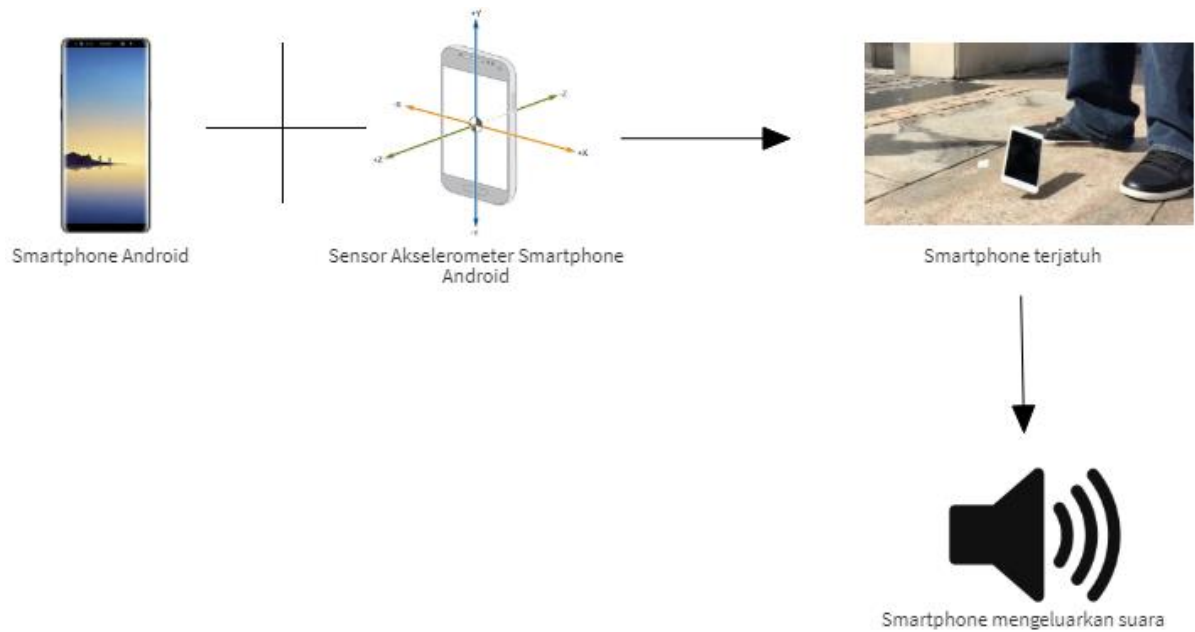
Aplikasi deteksi jatuh ini hanya memiliki satu halaman saja, yaitu halaman utama aplikasi. Karena aplikasi ini lebih mengutamakan fungsinya daripada *user interface*. Aplikasi ini dapat juga menyimpan data hasil dari akselerometer sehingga akan diberikan form untuk menamai file dan menyimpannya di dalam memori ponsel secara otomatis. Rancangan antar muka halaman utama dapat dilihat pada Gambar 3.3 berikut ini.



Gambar 3.3 Rancangan Halaman Utama

3.2.5 Perancangan Skema Sistem

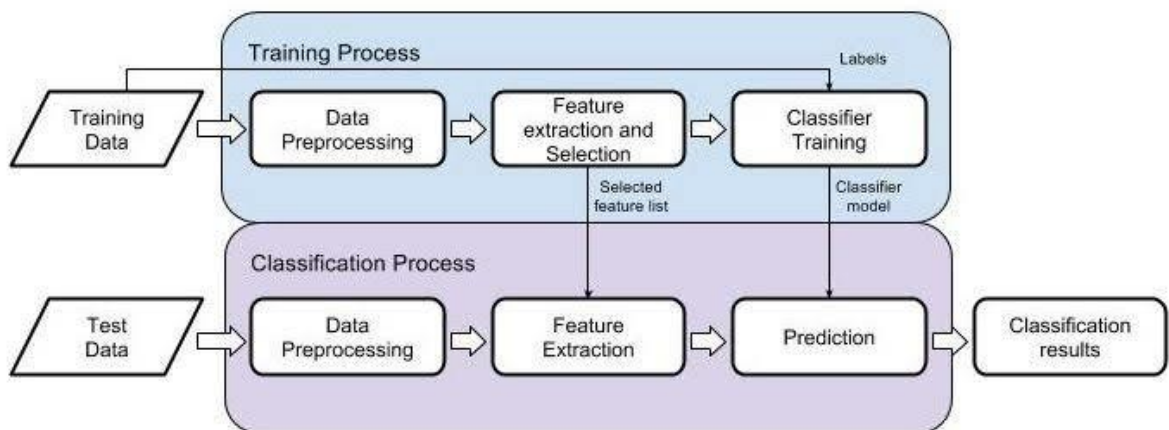
Pada penelitian deteksi jatuh menggunakan ponsel pintar android ini memanfaatkan sensor akselerometer yang terdapat di dalam ponsel android dengan sistem operasi android itu sendiri. Penelitian ini dilakukan agar pengguna ponsel pintar dapat mengetahui apabila ponsel mereka terjatuh. Sehingga kejadian ponsel terjatuh kemudian hilang dapat diminimalkan dengan menggunakan aplikasi ini. Dengan adanya *output* berupa suara, tentunya pengguna akan lebih cepat tanggap apabila ponsel terjatuh. Penjelasan mengenai skema aplikasi deteksi jatuh dapat dilihat pada Gambar 3.4 berikut ini.



Gambar 3.4 Skema Aplikasi Deteksi Jatuh

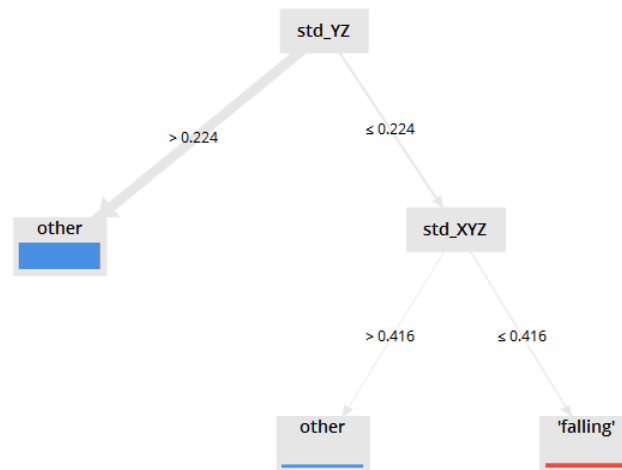
3.2.6 Perancangan Data

Ada beberapa macam data yang digunakan untuk membuat aplikasi deteksi jatuh ini. Data-data tersebut didapatkan dari berbagai macam sumber. Agar dapat lebih jelas mengetahui alurnya, dapat dilihat pada gambar 3.5 berikut ini.



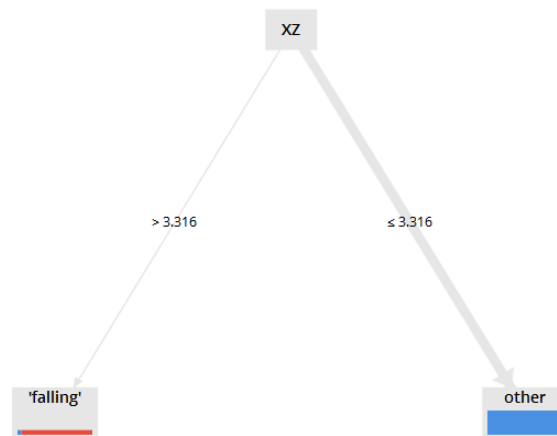
Gambar 3.5 Alur perancangan data deteksi jatuh

Training data, digunakan oleh algoritma untuk membentuk sebuah model klasifikasi. Setelah didapatkan data training, selanjutnya dilakukan normalisasi data sesuai dengan aktivitasnya. Selanjutnya model ini akan diproses yang merupakan representasi dari pengetahuan dan digunakan untuk membuat prediksi baru yang belum pernah ada. Setelah didapatkan data dari proses tersebut, kemudian digunakan *data set* untuk mengukur sejauh mana klasifikasi sudah dapat dilakukan dengan benar.



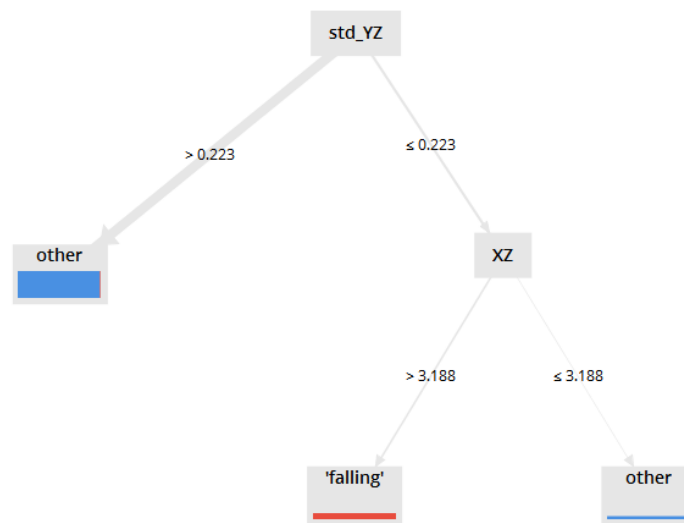
Gambar 3.7 *Tree* pertama aplikasi deteksi jatuh

Pada *tree* yang pertama terdapat 3 level yang dihasilkan dari pemrosesan data. Yang menjadi *root* adalah *std_YZ* yang diikuti *leaf* di bawahnya. Selanjutnya terdapat *tree* lain yang dihasilkan dari pemrosesan data dengan rapid miner yang dapat dilihat pada gambar 3.8 berikut ini.



Gambar 3.8 *Tree* kedua aplikasi deteksi jatuh

Tree diatas hanya terdapat 2 level saja, yang menjadi *root* adalah sumbu *XZ* dan diikuti oleh *leaf* di bawahnya. Kemudian dihasilkan *tree* kembali dari pemrosesan data yang dapat dilihat pada gambar 3.9 berikut ini.



Gambar 3.9 *Tree* ketiga aplikasi deteksi jatuh

Pada *tree* ketiga dihasilkan 3 level dengan *root* adalah `std_YZ` dan diikuti *leaf* di bawahnya. Setelah didapatkan beberapa *tree* seperti gambar diatas, selanjutnya diimplementasikan ke dalam baris kode untuk membuat *threshold* aplikasi deteksi jatuh.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi adalah tahap dimana dilakukan sesuatu apa yang telah dirancang sebelumnya. Implementasi akan menunjukkan perancangan yang telah dilakukan sebelumnya dapat berjalan baik dan benar sebagaimana fungsinya. Implementasi ini hanya ada satu bagian yaitu hanya implementasi perangkat lunak saja. Dalam implementasi perangkat lunak ini terdiri dari implementasi keluaran suara dan penyimpanan data sensor akselerometer.

4.1.1 Implementasi Keluaran Suara

Implementasi keluaran suara berperan penting dalam pembuatan aplikasi ini karena suara sebagai keluaran utama dari hasil aplikasi deteksi jatuh ini. Dengan adanya keluaran berupa suara, pengguna ponsel pintar akan dimudahkan apabila terjadi kejadian jatuh.

Pada proses keluaran suara dibutuhkan proses yang selanjutnya akan dieksekusi jika syarat-syarat yang dibutuhkan apabila ponsel terjatuh telah terpenuhi. Keluaran suara ini sebelumnya telah diatur sehingga hanya ada satu suara saja dan tidak bisa berkomunikasi dua arah. Peringatan suara hanya akan memberikan *alert* saja. Keluaran suara menggunakan Bahasa Indonesia yang sudah diatur dalam pembuatan aplikasi deteksi jatuh sebelumnya. Ketika terjadi kejadian jatuh, sistem akan mendeteksi besarnya *threshold* yang terjadi dan akan diolah oleh sistem yang selanjutnya akan dihasilkan keluaran berupa suara.

Pada implementasi keluaran suara ini digunakan salah satu fitur yang dimiliki oleh Google yakni *Google Text to Speech*. *Google Text to Speech* dibutuhkan untuk menghasilkan suara yang akan dikeluarkan oleh aplikasi deteksi jatuh ini. Ketika aplikasi mendeteksi adanya kejadian jatuh dan nilai sudah melebihi *threshold* maka *Google Text to Speech* akan memproses dan mengeluarkan suara dari ponsel pintar. Sistem menggunakan *Text to Speech* untuk mengubah respon sistem yang awalnya berupa tulisan menjadi sebuah suara. Layanan yang diimplementasikan dalam keluaran suara adalah sebagai berikut :

a. Google Text to Speech

Google Text to Speech merupakan sistem pengubah teks ke suara yang dapat dengan mudah digunakan dan terintegrasi dengan aplikasi android dengan mudah. Secara umum pengertian *Text To Speech* adalah sebuah sistem yang mengubah suatu teks menjadi ucapan. (Rieke, Tolle, & Setyawati, 2016) Pada

prinsipnya *tts* dibagi menjadi 2 sub sistem, yaitu *converter* teks ke fonem dan *converter* fonem ke ucapan. Bagian *converter* teks ke fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode bunyi yang biasanya direpresentasikan dengan kode fonem.(Rieke dkk., 2016) Pada implementasi aplikasi deteksi jatuh menggunakan sensor akselerometer, *Text to Speech* digunakan oleh sistem untuk memberikan respon berupa keluaran suara atau memberikan sebuah *alert* kepada pengguna ponsel pintar apabila terjadi kejadian jatuh. Implementasi *Text to Speech* dapat dilihat pada gambar 4.1 berikut ini.

```
public void onSensorChanged(SensorEvent event) {

    Long tsLong = System.currentTimeMillis()/1000;
    ts = tsLong.toString();

    // clean current values
    displayCleanValues();
    // display the current x,y,z accelerometer values
    displayCurrentValues();

    // get the change of the x,y,z values of the accelerometer
    deltaX = abs(lastX - event.values[0]);
    deltaY = abs(lastY - event.values[1]);
    deltaZ = abs(lastZ - event.values[2]);

    String data=i+" \t\t\t "+ts+" x = "+deltaX+" y = "+deltaY+" z =
    "+deltaZ+"\t" + deltaA;
    // if the change is below 2, it is just plain noise

    if ((deltaA >=20)) {
        tts.speak("ponsel anda jatuh", TextToSpeech.QUEUE_FLUSH, null);
    }
}
```

Gambar 4.1 Kode program *Text to Speech*

b. Penyimpanan Data

Untuk menyimpan data yang dihasilkan dari sensor akselerometer digunakan penyimpanan yang terdapat pada ponsel pintar itu sendiri. Penyimpanan data akan dilakukan apabila aplikasi deteksi jatuh telah selesai dijalankan. Penyimpanan data terletak di folder yang diberi nama AccFolder, rata, entropi, deviasi dan memiliki format data .txt, dimana folder tersebut telah dibuat di dalam sistem deteksi jatuh sebelumnya. Data tersebut yang digunakan untuk mengumpulkan data-data sehingga dapat diolah dan dijadikan *threshold* dalam pembuatan aplikasi ini. Kode

program yang menampilkan implementasi penyimpanan data dapat dilihat pada gambar 4.2.

```
public void writeToFile(String data,String nama)
{
    String path =
        Environment.getExternalStorageDirectory() + File.separator
+ "AccFolder";
    // Create the folder.
    File folder = new File(path);
    if (!folder.exists()) {
        folder.mkdirs();
    }
}
```

Gambar 4.2 Kode program penyimpanan data

c. Fitur Rata-rata

Fitur ini digunakan di perhitungan untuk menentukan *threshold* pada sistem deteksi jatuh. Fitur ini menghitung rata-rata tiap 10 data yang dihasilkan dari pergerakan sensor akselerometer. Kode program yang menampilkan implementasi nilai rata-rata dapat dilihat pada gambar 4.3 berikut.

```
String data2 = datax/10 + "\t" + datay/10 + "\t" + dataz/10 + "\t" +
dataa/10 + "\t" + datab/10 + "\t" + datac/10 + "\t" + datad/10;
```

Gambar 4.3 Kode program fitur rata-rata

d. Fitur Standar Deviasi

Dalam menentukan deteksi jatuh, digunakan juga standar deviasi dalam perhitungannya. Fitur ini juga digunakan dalam perhitungan sistem deteksi jatuh. Kode program yang menampilkan implementasi nilai standar deviasi dapat dilihat pada gambar 4.4 berikut ini.

```
if (i%10==0) {
    Double deviasix = 0.0;
    Double deviasiy = 0.0;
    Double deviasiz = 0.0;

    writeToFile(data2,namaFile, "Rata");
    for (int x = 0;x<10;x++) {
        deviasix = Math.sqrt(((Math.pow(datakex[i] -
(dataax/10),2)) + (Math.pow(datakex[i] - (dataax/10),2)) +
(Math.pow(datakex[i] - (dataax/10),2)) + (Math.pow(datakex[i] -
(dataax/10),2)) + (Math.pow(datakex[i] - (dataax/10),2)) +
(Math.pow(datakex[i] - (dataax/10),2)) + (Math.pow(datakex[i] -
(dataax/10),2) )/10);
```

Gambar 4.4 Kode program standar deviasi

e. Fitur Entropi

Fitur entropi ini juga digunakan di perhitungan sistem deteksi jatuh. Fitur ini akan digabungkan dengan fitur yang lain dan digunakan untuk membentuk pohon keputusan. Kode program yang menampilkan fitur entropi dapat dilihat pada gambar 4.5 berikut.

```

Double entropix = 0.0;
    Double entropiy = 0.0;
    Double entropiz = 0.0;
    Double entropia = 0.0;
    Double entropib = 0.0;
    Double entropic = 0.0;
    Double entropid = 0.0;
if (i>=1){
    entropix+=(datakex[i]-datakex[i-1]);
    entropiy+=(datakey[i]-datakey[i-1]);
    entropiz+=(datakez[i]-datakez[i-1]);
    entropia+=(datakea[i]-datakea[i-1]);
    entropib+=(datakeb[i]-datakeb[i-1]);
    entropic+=(datakec[i]-datakec[i-1]);
    entropid+=(dataked[i]-dataked[i-1]);
}

```

Gambar 4.5 Kode program entropi

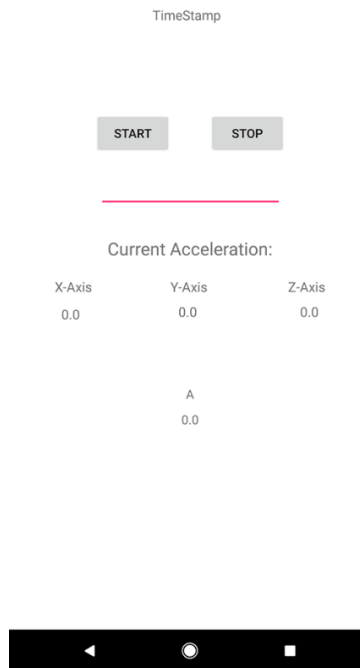
4.1.2 Implementasi Aplikasi Android

Implementasi pada aplikasi Android merupakan tahapan dimana diterapkannya rancangan antar muka pada aplikasi yang telah dibuat sebelumnya. Aplikasi Android memiliki fungsi dan fitur yang berbeda-beda tergantung pada tujuan pembuatan aplikasi itu sendiri. Aplikasi Android ini dibuat dengan menggunakan *platform* Android Studio dan diterapkan di Android Pie 9.0 dan dapat digunakan untuk versi sistem operasi Android dengan versi di bawahnya. Berikut adalah hasil implementasi antar muka halaman utama di aplikasi deteksi jatuh.

a. Halaman Utama

Dalam aplikasi ini hanya ada satu halaman yaitu halaman utama aplikasi saja, Karena sistem ini adalah sistem yang memberikan peringatan kepada pengguna ponsel pintar apabila terjadi kejadian jatuh. Pada halaman utama aplikasi pengguna

dapat langsung menjalankan aplikasi dan menghentikan aplikasi. Pada halaman utama ini juga dapat dilihat perubahan nilai sensor akselerometer apabila terjadi gerakan di ponsel pintar. Perubahan nilai tersebut adalah nilai sumbu x,y, dan sumbu z tergantung pada gerakan yang terjadi di ponsel pintar. Berikut ini adalah halaman utama aplikasi yang dapat dilihat pada gambar 4.6.



Gambar 4.6 Halaman utama aplikasi

4.1.3 Implementasi Perangkat Keseluruhan

Setiap baris kode yang telah dibuat akan menjadi satu aplikasi dengan menggunakan *Text to Speech* yang digabungkan dengan Bahasa *java* sehingga terciptalah sebuah aplikasi Android yang digunakan sebagai peringatan atau deteksi jatuh dan dapat berjalan sebagaimana fungsinya. Pengguna aplikasi deteksi jatuh tidak dapat melakukan kontrol suara karena suara pada aplikasi ini akan keluar secara otomatis tergantung pada kondisi ponsel. Agar aplikasi dapat berjalan maka harus terjadi gerakan yang nilainya melebihi ambang batas atau *threshold*. Setelah aplikasi membaca *threshold* dan melebihi nilai yang sudah diatur, maka aplikasi deteksi jatuh akan mengeluarkan keluaran berupa suara sehingga dapat mengetahui bahwa terjadi kejadian jatuh.

4.2 Implementasi

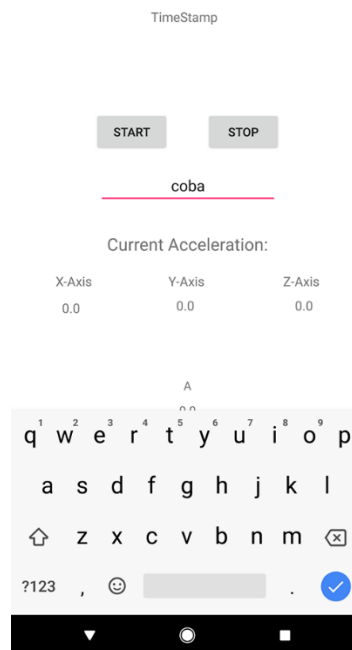
Tahap pengujian penelitian dilakukan apakah aplikasi sudah dapat berjalan lancar dan benar. Dengan dilakukan pengujian ini dapat diketahui kekurangan dalam aplikasi ini. Pengujian dilakukan dengan cara :

- a. Melakukan uji penamaan file dan penyimpanan data.
- b. Melakukan uji gerakan di aplikasi android dengan ponsel pintar.
- c. Melakukan uji gerakan jatuh pada ponsel

4.2.1 Pengujian Aplikasi Android

- a. Pengujian penamaan *file* dan *folder*

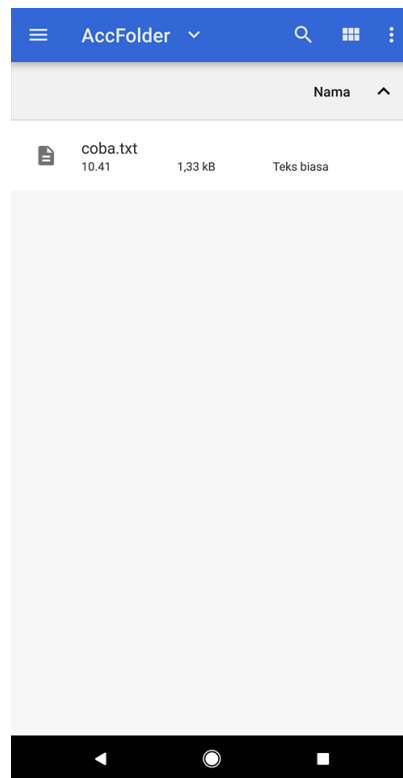
Sebelum menggunakan aplikasi deteksi jatuh, pengguna harus melakukan pengisian nama file pada halaman utama. Nama file yang telah diisi oleh pengguna nantinya akan disimpan ke dalam folder secara otomatis apabila aplikasi dihentikan. Pengisian nama file dapat dilihat pada gambar 4.7.



Gambar 4.7 Penamaan file di Aplikasi

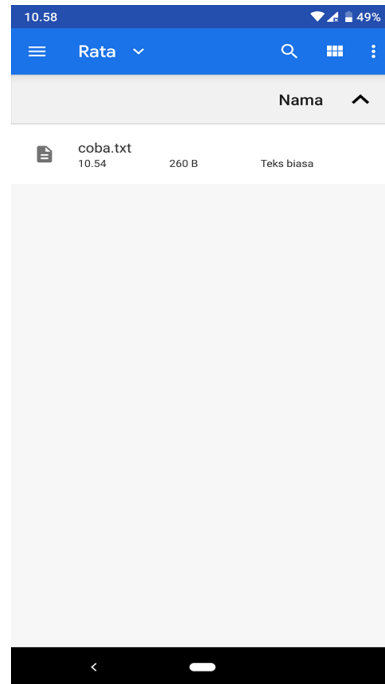
Setelah pengguna menuliskan nama file, maka aplikasi android deteksi jatuh baru dapat digunakan. Apabila nama file kosong maka akan keluar peringatan bahwa nama file tidak boleh kosong dan harus wajib diisi. Kemudian untuk menjalankan aplikasi pengguna dapat menekan tombol start pada aplikasi tersebut. Untuk menyimpan file dapat dilakukan dengan cara menghentikan aplikasi dengan menekan tombol stop.

Maka file sudah tersimpan di dalam ponsel. Hasilnya dapat dilihat pada gambar 4.8 berikut ini.



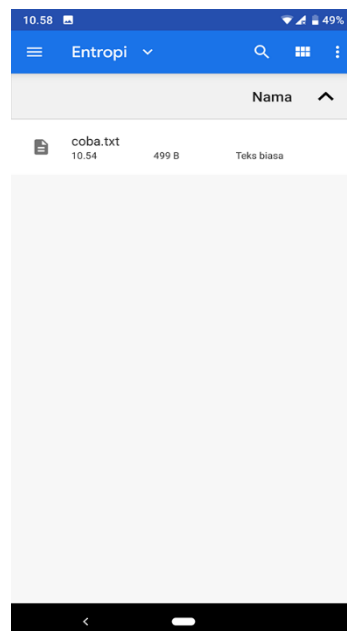
Gambar 4.8 Penyimpanan file coba di folder AccFolder

Selain disimpan di folder AccFolder file hasil pergerakan sensor akselerometer juga disimpan dalam folder Rata. Folder rata berisi hasil rata-rata dari pergerakan sensor akselerometer. Kemudian untuk penyimpanan di folder Rata dapat dilihat pada gambar 4.9 berikut ini.



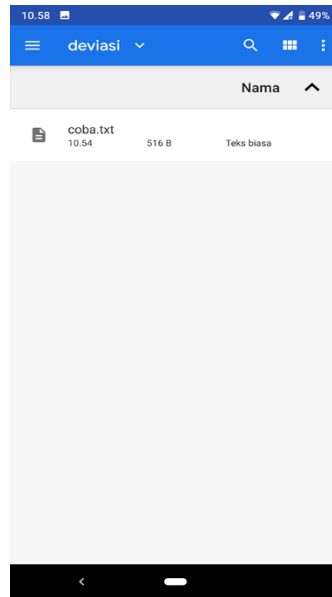
Gambar 4.9 Penyimpanan file coba di folder Rata

Pergerakan sensor akselerometer juga disimpan di folder entropi yang dapat dilihat pada gambar 4.10 berikut ini.



Gambar 4.10 Penyimpanan file coba di folder Entropi

Hasil pergerakan sensor juga disimpan di folder deviasi yang dapat dilihat pada gambar 4.11 berikut ini.



Gambar 4.11 Penyimpanan file coba di folder Deviasi

Isi dari file adalah nilai x,y, dan z dari gerakan sensor akselerometer yang disimpan dengan format .txt yang dapat dilihat oleh pengguna aplikasi.

Isi file dapat dilihat di bawah ini.

Isi file coba.txt pada folder ACC

```

3 1534823652 x =0.6241913 y =1.7742157 z =9.742355 A=9.68725349358751 B=1.8969414864314993 C=9.517721812672294 D=9.669557555687536
4 1534823653 x =0.6870575 y =1.8597412 z =9.546387 A=9.922244856541491 B=1.8808126631944482 C=9.762331085111384 D=9.902592221364905
5 1534823653 x =0.5659332 y =1.8748322 z =9.640442 A=9.75008646608795 B=1.9825956522125532 C=9.571078572655503 D=9.725849073358784
6 1534823653 x =0.661911 y =1.9244232 z =9.594254 A=9.837346782727181 B=1.958386102601214 C=9.657038881736932 D=9.82105441086704
7 1534823653 x =0.625885 y =1.8764954 z =9.616394 A=9.807712820730723 B=2.035075180577019 C=9.617058983198977 D=9.785351560550813
8 1534823653 x =0.6447449 y =1.8884735 z =9.640686 A=9.817739072122524 B=1.9781220205564796 C=9.636740521737668 D=9.797768584023315
9 1534823654 x =0.6715851 y =1.8875275 z =9.57901 A=9.845041922910598 B=1.9955019437493224 C=9.66222130298269 D=9.823907332290828
10 1534823654 x =0.6432953 y =1.8808289 z =9.626862 A=9.786276991082534 B=2.003443614359528 C=9.602523519720082 D=9.763205950387787
1 1534823654 x =0.63990784 y =1.81781 z =9.617584 A=9.829944436119032 B=1.9877993256315698 C=9.648330749662493 D=9.808872208313423
2 1534823654 x =0.72164917 y =1.8032074 z =9.610916 A=9.808763314942139 B=1.9271521341225188 C=9.638848980452424 D=9.787868028057385
3 1534823654 x =0.78933716 y =1.7902832 z =9.520187 A=9.805204558701888 B=1.9422498132924346 C=9.63797074292633 D=9.778612457638337

```

Pada folder acc ini menyimpan semua pergerakan yang dilakukan pada sensor akselerometer. Semua hasil akan tercatat dan langsung otomatis akan disimpan pada folder ini. Data-data yang tersimpan yaitu *timestamp*, sumbu x, y, dan z yang merupakan sumbu dari pergerakan yang dihasilkan oleh sensor akselerometer yang bergerak secara vertical, horizontal, dan bergerak secara acak, ada juga data A,B,C,D yang

merupakan rumus yang dibuat untuk menghitung besarnya nilai akselerasi yang akan digunakan untuk membuat *tree* yang selanjutnya dibuat *threshold*.

Isi file coba.txt pada folder Deviasi.

4.672279507919221 12.688025071085201 64.23112149749339 58.2895280278099912.039176619835743 57.18633070827698
58.12875413187254

5.64183396529987 68.654187787240625 64.72486913018716 64.90269896172411 10.93502660629516 64.13389245144108
64.71438790030153

11.512700676587116 8.724266649879539 64.15254418939834 65.85977185096372 14.081117442727374 65.36387945990116
64.55767372556217

11.942488550782718 24.752491966686385 57.139931158923375 65.3608985010969 27.855145062940956 59.66630332034429
63.574902035153535

Pada folder deviasi data yang disimpan akan langsung dihitung dengan rumus standar deviasi. Hasilnya akan otomatis terhitung dan folder deviasi akan menghitung data yang ada di folder acc. Tiap 10 data pada folder acc akan menghasilkan 1 data standar deviasi. Nilai yang dihitung adalah sumbu x,y, dan z serta rumus yang sudah dibuat sebelumnya.

Isi file coba.txt pada folder Entropi.

0.6715850830078125 1.8875274658203125 9.579010009765625 9.845039367675781 1.9955015182495117 9.662223815917969
9.823905944824219

1.715423583984375 0.579925537109375 10.551742553710938 9.782562255859375 1.4907512664794922 9.73046875
9.720733642578125

1.28021240234375 1.759124755859375 9.634735107421875 9.960090637207031 2.380615234375 9.8153076171875
9.818305969238281

1.3302459716796875 4.3846282958984375 8.545806884765625 9.325859069824219 6.29149055480957 7.0079345703125
9.233062744140625

Folder entropi menyimpan data yang dihasilkan oleh sensor akselerometer dan langsung akan menghitung menggunakan rumus entropi yang sudah dibuat sebelumnya. Sama seperti folder deviasi, folder entropi akan menghitung tiap 10 data dari folder acc yang akan menghasilkan 1 data entropi. Nilai yang dihitung adalah sumbu x,y, dan z serta rumus-rumus yang sudah dibuat sebelumnya.

Isi file coba.txt pada folder Rata.

0.6919708 1.8939911 9.599749 8.828399 1.8214457 8.661474 8.804483

0.7878464 1.202562 9.655373 9.808841.6534061 9.704993 9.768461

1.6217865 1.386389.553833 9.843867 2.1117563 9.760466 9.661688

1.7374862 3.609221 8.640951 9.757543 4.208368.884844 9.51017

Pada folder rata akan dihasilkan nilai rata-rata dari pergerakan sensor akselerometer. Tiap 10 data yang dihasilkan terdapat 1 data rata-rata yang akan langsung disimpan pada folder rata. Nilai yang dihitung yaitu sumbu x,y, dan z.

b. Uji gerakan di aplikasi

Untuk mendapatkan nilai dari file yang telah dilihat di atas, maka perlu adanya gerakan pada ponsel pintar sehingga nilai sensor akselerometer dapat berubah dan disimpan ke dalam folder. Gerakan diperlukan untuk mengubah nilai x,y dan z pada ponsel. Jika ponsel tidak digerakkan maka nilainya juga tetap akan tersimpan tetapi nilainya konstan yaitu 9,8 m/s sesuai dengan besarnya gravitasi. Uji gerakan ini juga berfungsi apakah sensor akselerometer sudah bekerja dengan baik dan sesuai keinginan atau belum. Pengujian ini juga harus sesuai dengan sumbu gerakan, misalnya saja jika ponsel digerakkan ke sumbu x, maka nilai dari sumbu x yang harus berubah. Tidak boleh terbalik antar sumbu karena jika terbalik akan menyusahakan dalam perhitungan. Untuk melihat perubahan nilai sumbu di aplikasi dapat dilihat pada gambar 4.12 berikut ini.

TimeStamp
1532491347

START STOP

coba

Current Acceleration:

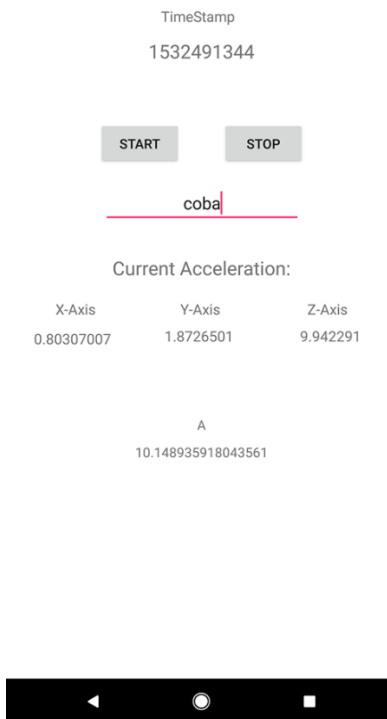
X-Axis	Y-Axis	Z-Axis
5.6362457	0.20503235	13.084061

A
14.247875389860763



Gambar 4.12 Perubahan nilai sumbu sensor

Setelah ponsel digerakkan maka akan terjadi perubahan nilai sumbu sensor akselerometer yang dapat dilihat pada gambar 4.13 berikut ini.



Gambar 4.13 Perubahan nilai sumbu sensor setelah aksi

c. Uji gerakan jatuh

Pada aplikasi deteksi jatuh ini dilakukan pengujian dengan menjatuhkan badan ke arah depan, belakang, dan gerakan biasa agar dapat dilihat tingkat keberhasilan aplikasi ini. Uji gerakan jatuh ini dilakukan beberapa kali dengan posisi jatuh dan gerakan yang berbeda-beda. Uji gerakan jatuh ini dilakukan untuk menguji apakah nilai *threshold* yang digunakan sudah sesuai dengan kondisi atau belum. Total pengujian yang digunakan untuk melakukan pengujian akurasi ini dilakukan sebanyak 3 orang dengan masing-masing pengujian untuk setiap kategori sebanyak 10 kali. Hasil pengujian terhadap respon ponsel ketika terjadi kejadian jatuh dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4.1 Akurasi aplikasi terhadap gerakan jatuh

NO	KATEGORI	ALARM		THRESHOLD	
		YA	TIDAK	YA	TIDAK
1	Jatuh ke arah depan	24	6	$x \geq 0.847$ $y \geq 13.63$ $z \geq 4.419$	$x < 0.847$ $y < 13.637$ $z < 4.419$
2	Jatuh ke arah belakang	21	9	$x \geq 0.896$	$x < 0.896$

				$y \geq 13.667$ $z \geq 4.154$	$y < 13.667$ $z < 4.154$
3	Berjalan	0	30	-	-
4	Berlari	13	17	$x \geq 0.505$ $y \geq 13.56$ $z \geq 0.582$	$x < 0.505$ $y < 13.56$ $z < 0.582$
5	Naik tangga	0	30	-	-
6	Turun tangga	0	30	-	-
7	Tidur	0	30	-	-
8	Melompat	24	6	$x \geq 3.629$ $y \geq 13.29$ $z \geq 2.002$	$x < 3.629$ $y < 13.29$ $z < 2.002$

Setelah didapatkan hasil pengujian dengan kategori-kategori di atas, kemudian menghitung nilai akurasi dari aplikasi deteksi jatuh. Yakni dengan menggunakan kategori rumus *true positive (TP)*, *true negative(TN)*, *false positive(FP)*, dan *false negative(FN)*.

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN}$$

Penjelasan :

TP : terjadi kejadian jatuh dan ponsel melaporkan jatuh

TN : tidak terjadi kejadian jatuh dan ponsel melaporkan tidak jatuh

FP : tidak terjadi kejadian jatuh dan ponsel melaporkan jatuh

FN : terjadi kejadian jatuh dan ponsel melaporkan tidak jatuh

Hasilnya adalah :

TP sebanyak 45

TN sebanyak 143

FP sebanyak 37

FN sebanyak 15

Setelah dihitung menggunakan rumus akurasi, maka besarnya akurasi yang didapatkan dari aplikasi deteksi jatuh sebesar 78,3%.

4.3 Analisis Penelitian

Tahap analisi penelitian merupakan tahapan dimana sebuah aplikasi dinilai kelebihan maupun kekurangannya. Semua kekurangan dan kelebihan akan digunakan sebagai bahan evaluasi untuk menilai keberhasilan dari aplikasi yang telah dibuat. Kekurangan yang didapatkan juga diharapkan dapat menjadi pertimbangan untuk mengembangkan aplikasi yang lebih baik lagi. Pada aplikasi deteksi jatuh ini didapatkan hasil analisis berupa akurasi jatuh sistem. Akurasi adalah ketepatan sistem membaca kejadian jatuh yang terjadi dengan nilai akurasi sebesar 78,3% dengan berbagai macam posisi jatuh yaitu jatuh ke depan, belakang, dan gerakan biasa. Adapun Kelebihan dan kekurangan dari aplikasi deteksi jatuh ini adalah :

4.3.1 Kelebihan Aplikasi

- a. Aplikasi dapat dapat mendeteksi jatuh dengan berbagai macam posisi jatuh.
- b. Aplikasi menyimpan hasil sensor akselerometer di dalam memori ponsel.
- c. Dapat berjalan tanpa aplikasi dibuka (aplikasi di minimize) .

4.3.2 Kekurangan Aplikasi

- a. Aplikasi berjalan tanpa internet.
- b. Dapat mengurangi penyimpanan memori ponsel karena menyimpan *file log* sensor akselerometer.
- c. Aplikasi tidak dapat berkomunikasi secara dua arah.
- d. *Output* aplikasi hanya berupa Bahasa Indonesia.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan pada aplikasi deteksi jatuh, maka didapatkan kesimpulan sebagai berikut ini.

- a. Aplikasi deteksi jatuh dapat memberikan *alert* kepada pengguna ponsel pintar apabila terjadi kejadian jatuh.
- b. Aplikasi dapat menyimpan data dari sensor akselerometer dan log pada saat terjadi kejadian jatuh.
- c. Aplikasi dapat berjalan dengan berbagai kondisi posisi jatuh.

5.2 Saran

Setelah dilakukan pengujian dengan melibatkan 3 orang dan dilakukan pengujian sebanyak 10 kali tiap orang dengan aktivitas seperti pada tabel uji didapatkan bahwa masih terdapat nilai dari percepatan yang masuk ke dalam nilai *threshold* sehingga seringkali walaupun tidak terjadi kejadian jatuh, aplikasi akan tetap memberikan peringatan jatuh. Pada aplikasi ini masih terdapat banyak kekurangan seperti yang telah dijelaskan sebelumnya, maka diharapkan pada penelitian selanjutnya dapat dilakukan perbaikan pada pengembangan aplikasi deteksi jatuh ini. Saran untuk pengembang sistem yang akan dikembangkan selanjutnya adalah sebagai berikut :

- a. Sistem operasi Android adalah sistem operasi yang perkembangannya sangat cepat sehingga pengembang dapat melakukan modifikasi sesuai dengan kebutuhan.
- b. Memperbaiki output untuk sumbu x, y, dan z pada aplikasi deteksi jatuh ini menjadi lebih baik lagi.
- c. Menambahkan fitur pada aplikasi deteksi jatuh seperti mengirimkan email atau sms jika terjadi kejadian jatuh.
- d. Membuat penyimpanan dengan file yang lebih tertata agar mudah diolah dan dimasukkan ke dalam satu folder.

DAFTAR PUSTAKA

- 10 Fold-Cross Validation. (n.d.). Retrieved September 13, 2018, from <https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/>
- 150 Rahasia Pemrograman Java*. (n.d.). Elex Media Komputindo.
- 2013-2-00424-MTIF Bab2001.pdf. (n.d.). Retrieved from <http://library.binus.ac.id/eColls/eThesiscoll/Bab2/2013-2-00424-MTIF%20Bab2001.pdf>
- Data_Preprocessing.pdf. (n.d.). Retrieved from http://iasri.res.in/ebook/win_school_aa/notes/Data_Preprocessing.pdf
- Dewi, N. K., Syafitri, U. D., & Mulyadi, S. Y. (2011). PENERAPAN METODE RANDOM FOREST DALAM DRIVER ANALYSIS. *FORUM STATISTIKA DAN KOMPUTASI*, 16(1). Retrieved from <http://jurnal.ipb.ac.id/index.php/statistika/article/view/5443>
- Enterprise, J. (2015). *Mengenal Dasar-Dasar Pemrograman Android*. Elex Media Komputindo.
- Hendini, A. (2016). PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK). *Jurnal Khatulistiwa Informatika*, 4(2). Retrieved from <http://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/1262>
- Iman Fahruzi, & Emilio Santos Abdullah. (2014). Integrasi Sensor Multifungsi Accelerometer untuk Mendeteksi Kekuatan Benturan, 3.
- Liu, C. (n.d.). Progress in MEMS and Micro Systems Research, 10.
- Muttaqin, M. J., Otok, B. W., & Hakim, J. A. R. (n.d.). Metode Ensemble pada CART untuk Perbaikan Klasifikasi Kemiskinan di Kabupaten Jombang, 6.

- PENGAJIAN DAN PENCEGAHAN JATUH PADA LANSIA. (2009, September 19). Retrieved August 13, 2018, from <https://stikeskabmalang.wordpress.com/2009/09/19/pengkajian-dan-pencegahan-jatuh-pada-lansia/>
- Rahadi, D. R. (2014). PENGUKURAN USABILITY SISTEM MENGGUNAKAN USE QUESTIONNAIRE PADA APLIKASI ANDROID. *Jurnal Sistem Informasi*, 6(1). Retrieved from <https://ejournal.unsri.ac.id/index.php/jsi/article/view/772>
- Rakhman, A. Z., Ir. Lukito Edi Nugroho, M. S., & Widyawan, S. T. (2015). *SISTEM MONITORING KEJADIAN JATUH MENGGUNAKAN SMARTPHONE*. Universitas Gadjah Mada. Retrieved from http://etd.repository.ugm.ac.id/index.php?mod=penelitian_detail&sub=PenelitianDetail&act=view&typ=html&buku_id=83451&obyek_id=4
- Riantana, R. (2015). Aplikasi Sensor Accelerometer pada Handphone Android sebagai Pencatat Getaran Gempabumi secara Online. *Jurnal Fisika Dan Aplikasinya*, 11(3), 114. <https://doi.org/10.12962/j24604682.v11i3.1071>
- Sartono, B., & Syafitri, U. D. (n.d.). METODE POHON GABUNGAN: SOLUSI PILIHAN UNTUK MENGATASI KELEMAHAN POHON REGRESI DAN KLASIFIKASI TUNGGAL, 7.
- Setiawati, D., Taufik, I., Jumadi, J., & Zulfikar, W. B. (2016). Klasifikasi Terjemahan Ayat Al-Quran Tentang Ilmu Sains Menggunakan Algoritma Decision Tree Berbasis Mobile. *Jurnal Online Informatika*, 1(1), 24–27.
- Sinaga, T. E. (2013). SISTEM PENGAMANAN HANDPHONE MENGGUNAKAN SENSOR ACCELEROMETER UNTUK MENDETEKSI GERAKAN JATUH BEBAS. *Saintia Fisika*, 6(1). Retrieved from <https://jurnal.usu.ac.id/index.php/sfisika/article/view/5611>

- Sinaga, T. E., & Situmorang, M. (n.d.). SISTEM PENGAMANAN HANDPHONE MENGGUNAKAN SENSOR ACCELEROMETER UNTUK MENDETEKSI GERAKAN JATUH BEBAS, 8.
- Sinsuw, A., & Najoan, X. (2013). Prototipe Aplikasi Sistem Informasi Akademik Pada Perangkat Android. *Jurnal Teknik Elektro Dan Komputer*, 2(5), 21–30.
- W, R. A., Tolle, H., & Setyawati, O. (2016). Pengembangan Aplikasi Text-to-Speech Bahasa Indonesia Menggunakan Metode Finite State Automata Berbasis Android. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 5(1). Retrieved from <http://ejnteti.jteti.ugm.ac.id/index.php/JNTETI/article/view/179>

LAMPIRAN