

# Rancang Bangun Sistem *Fall Detection* dengan Analisa Percepatan sudut rotasi (*Quaternion*) secara Real-time menggunakan Mikrokontroler Arduino

M. Satrio Ramadhana Yudhono, Waskitho Wibisono, dan Henning Titi Ciptaningtyas  
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
*e-mail:* waswib@if.its.ac.id

**Abstrak**— Deteksi jatuh merupakan salah satu tantangan utama dalam domain kesehatan publik, terutama untuk orang yang berusia lanjut yang menurunnya kebugaran fisik mereka, serta penderita stroke dan yang terindikasi mempunyai dampak stroke, maka diperlukan pendeteksian untuk mengurangi dampak negatif dari jatuh. Penelitian ini mengembangkan sistem deteksi jatuh berdasarkan perangkat wearable dengan mikrokontroler arduino. Sistem ini memonitor gerakan tubuh manusia, memonitor gerakan jatuh yang tidak biasa dari aktivitas normal sehari-hari dengan menggunakan analisa nilai quaternion secara real-time yang digunakan sebagai feature dalam metode K Nearest Neighbors maupun Decision tree. Percobaan dilakukan dengan meletakkan perangkat pada bagian dada pengguna. Hasil percobaan pada fall detection bahwa dengan menggunakan metode Decision tree didapatkan akurasi sebesar 90%, dan akurasi sebesar 65% dengan menggunakan metode kNN. Aplikasi dapat mengirimkan notifikasi SMS kepada kerabat beserta lokasi pasien ketika aplikasi mendeteksi aktivitas fisik pengguna dalam kondisi terjatuh dengan menggunakan GPS/GPRS/GSM Shield v3.0.

**Kata Kunci**— Wearable, Mikrokontroler, Quaternion, Real-time, K-Nearest Neighbors, Decision tree, GPS/GPRS/GSM Shield, GPS v3.0.

## I. PENDAHULUAN

INSIDEN jatuh merupakan masalah dan sering terjadi di lingkungan sekitar, pada umumnya terjadi pada usia lanjut namun tidak menutup kemungkinan insiden terjatuh terjadi pada siapa pun dan tidak mengenal usia. Pada sebagian kalangan hal itu dianggap biasa dan tidak perlu dikhawatirkan, namun untuk kalangan tertentu yaitu pasien penderita stroke ataupun seseorang yang mempunyai indikasi Stroke ini sangat berbahaya, selain itu yang sedang mengalami gangguan kesehatan atau penuaan umur juga diperlukan perhatian khusus. menurut data dari *World Health Organization* (WHO) pada tahun 2012 menyebutkan sejumlah 420.000 terjadi insiden terjatuh fatal yang menyebabkan kematian pada setiap tahunnya dan terjadi insiden terjatuh dengan kategori berat sejumlah 37,3 sehingga dibutuhkan perhatian medis di setiap tahun [1]. Selain itu seseorang yang berumur 65 tahun atau lebih paling tidak pernah mengalami insiden terjatuh minimal satu kali selama kurun waktu 1 tahun terakhir, dan 2/3 orang tersebut mengalami insiden terjatuh kembali dalam periode waktu 2 tahun [2]. Dampak yang dapat terjadi selain kematian antara lain memar, dislokasi sendi patah tulang dan trauma. Kebanyakan usia lanjut

yang terjatuh yang tidak mampu untuk bangkit sendiri tergeletak dilantai dalam waktu lebih dari 1 jam meninggal dalam waktu 6 bulan setelah insiden tersebut [3].

Tingginya angka kematian yang terjadi akibat insiden terjatuh akibat dari beberapa faktor antara lain: tidak adanya pertolongan pertama karena posisi korban yang jauh dari keramaian dan tidak diketahui keberadaannya, serta korban yang sulit untuk menghubungi keluarganya dan lainnya.

Berdasarkan latar belakang tersebut penulis menawarkan sebuah solusi untuk membuat rancang bangun *fall detection system* menggunakan mikrokontroler Arduino dengan menggunakan metode analisis pergerakan sudut rotasi menggunakan nilai quaternion serta menggunakan SIM908 GSM/GPRS/GPS Shield. Sensor *accelerometer* akan mengambil data-data percepatan linier secara *real-time*. pendeteksian percepatan tersebut berdasarkan pada 3 sumbu yaitu sumbu  $x$ , sumbu  $y$  dan sumbu  $z$  [4], dan nilai quaternion akan menghasilkan sudut pergerakan rotasi saat terjadi transisi gerakan dengan menggunakan sensor orientasi dapat menentukan apakah perubahan gerakan yang terjadi merupakan perubahan yang mengindikasikan gerakan posisi jatuh. selain itu diperlukan Shield GPRS dan GPS untuk melanjutkan data ke server dan mengirim notifikasi berupa SMS ke kerabat korban beserta lokasi tempat di mana korban tergeletak.

Pada makalah ini membahas mengenai implementasi *fall detection system* menggunakan analisa nilai percepatan sudut (quaternion) dan berdasarkan nilai perubahan percepatan menurut sumbu  $x, y, z$ . Dari nilai-nilai tersebut digunakanlah metode *decision tree* dan metode *kNN* untuk pendeteksian aktivitas jatuh, dan kemudian hasil dari kedua tersebut dibandingkan, metode mana yang memiliki nilai akurasi yang lebih baik.

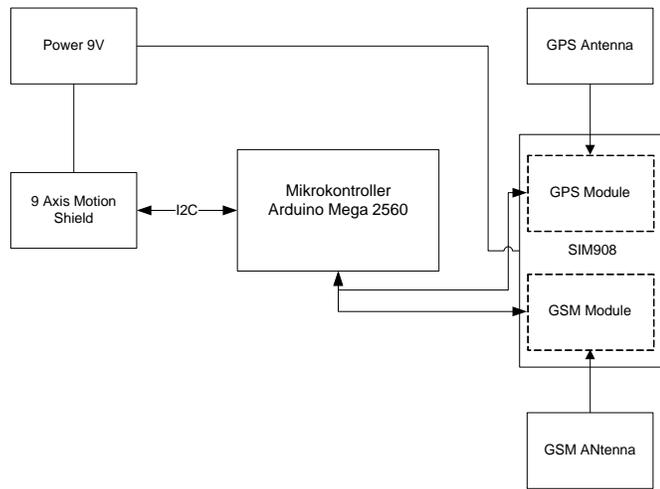
## II. URAIAN PENELITIAN

### A. Perangkat Wearable

Dalam Penelitian ini perangkat *wearable* yang digunakan dalam *fall detection* menggunakan sensor orientasi 9 axis motion Shield Arduino yang membaca data-data percepatan secara *real-time* yaitu data percepatan akselerasi pada sumbu  $x, y$  dan  $z$  dan data-data nilai quaternion  $q (q_0, q_1, q_2, q_3)$ . Kemudian untuk mendapatkan data lokasi saat insiden terjatuh berupa titik koordinat *latitude* dan *longitude* serta pengiriman

data tersebut ke *website monitoring* dan pengiriman peringatan melalui SMS kepada kerabat menggunakan GPS/GPRS/GSM *Shield V3.0*.

Kemudian kedua *Shield* tersebut di hubungkan dengan mikrokontroler Arduino Mega 2560, Arduino sebagai pusat dalam pembacaan sensor, apabila terjadi pergerakan yang mengindikasikan gerakan jatuh, Arduino juga akan mengirimkan perintah untuk mengirimkan pemberitahuan berupa SMS. Semua perangkat tersebut dihubungkan dan menggunakan *power* berupa baterai sebesar 9V, sehingga rangkaian perangkat *wearable* dalam *fall detection system* ini seperti pada Gambar 1.



Gambar 1 Rangkaian Perangkat *Wearable Fall Detection*

**B. Activity Recognition**

*Activity recognition* merupakan teknik yang digunakan untuk proses pendeteksian aktivitas fisik *user*. Tahap-tahap *activity recognition* adalah data *collection*, *feature extraction*, dan data *interpretation*. Data *collection* merupakan proses pengambilan data dengan menggunakan *device* berupa sensor. Data ini diharapkan dapat digunakan untuk dilakukan proses identifikasi yang bertujuan untuk menggambarkan sebuah entitas tertentu. *Feature extraction* merupakan proses pengolahan data sedemikian hingga agar data tersebut tidak terlalu konvergen atau terlalu divergen untuk dilakukan proses pendeteksian. Data *interpretation* merupakan proses pendeteksian aktivitas fisik *user* dengan menggunakan algoritma *fall detection*.

**C. Algoritma Fall Detection**

Dalam penelitian ini digunakan dua metode dalam pendeteksian aktivitas fisik *user* pada saat terjatuh, pertama menggunakan Decision Tree dan kedua menggunakan metode K-Nearest Neighbors, kedua metode tersebut sama-sama menggunakan nilai perubahan percepatan akselerasi, nilai skalar alpha ( $\alpha$ ) dan nilai *quaternion* untuk menentukan sudut postur *user*, apakah postur *user* termasuk dalam kondisi berbaring atau tidak [5].

Terdapat beberapa langkah untuk mendeteksi aktivitas fisik *user* pada saat terjatuh dengan menggunakan 2 metode tersebut. Namun sebelumnya dilakukan penghitungan nilai skalar alpha ( $\alpha$ ) dari sensor *accelerometer* dan penghitungan nilai skalar theta ( $\theta$ ) sebagai nilai yang menunjukkan sudut postur *user* dari

nilai *quaternion* q. Adapun rumus dalam pencarian nilai  $\alpha$ ,  $\omega$ , dan  $\theta$  adalah seperti berikut.

- Penghitungan perubahan percepatan  
 Penghitungan perubahan percepatan dimaksudkan untuk mencari nilai sebelum terjadinya insiden terjatuh dan sesudah terjadinya insiden, perubahan tersebut dapat dilihat pada Gambar 2. Untuk mengetahui adanya perubahan percepatan yang dilakukan ketika terjadi transisi gerakan, dapat dilakukan dengan menghitung perubahan nilai percepatan pada masing-masing sumbu X, Y, Z untuk menghitung perubahan percepatan pada sumbu X dapat dilakukan dengan cara sebagai berikut:

$$d(x_i, x_{i+1}) = (x_{i+1} - x_i). \tag{1}$$

$x_{i+1}$  = percepatan sumbu x data ke-i+1  
 $x_i$  = percepatan sumbu x data ke-i

- Penghitungan nilai alpha ( $\alpha$ )  
 Penghitungan nilai *alpha* ( $\alpha$ ) didapat dari kuadrat dari penjumlahan sumbu x, y, dan z dari sensor *accelerometer* seperti Persamaan 2.1 [5].

$$|\alpha| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{2}$$

$a_x^2$  = kuadrat nilai *alpha* pada sumbu x  
 $a_y^2$  = kuadrat nilai *alpha* pada sumbu y  
 $a_z^2$  = kuadrat nilai *alpha* pada sumbu z

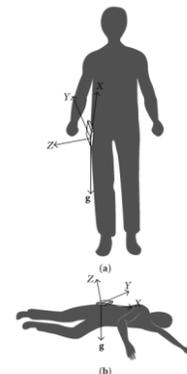
Nilai *alpha* ( $\alpha$ ) digunakan pada algoritma *fall detection* untuk mendeteksi aktivitas tersebut termasuk aktivitas statis atau dinamis.

- Penghitungan nilai theta ( $\theta$ )  
 Penghitungan nilai *theta* ( $\theta$ ) didapat dari *arctan* dari nilai *quaternion* seperti Persamaan 3 [5].

$$\theta = (2 \cdot \arctan \left( \frac{\sqrt{q_1^2 + q_2^2 + q_3^2}}{q_0} \right) * \frac{180}{3.14}) \tag{3}$$

$q_0$  = komponen nilai  $q_0$  pada *quaternion*  
 $q_1$  = komponen nilai  $q_1$  pada *quaternion*  
 $q_2$  = komponen nilai  $q_2$  pada *quaternion*  
 $q_3$  = komponen nilai  $q_3$  pada *quaternion*

Nilai *theta* ( $\theta$ ) digunakan pada algoritma *fall detection* untuk mendeteksi postur *user*, apakah postur *user* termasuk dalam kondisi berbaring atau tidak [5].



Gambar 2 Perubahan Koordinat Sumbu x,y,z dan g (a) Sebelum Jatuh (b) Sesudah Jatuh [5]

Setelah nilai  $\alpha$ , dan  $\theta$  didapatkan maka akan dilanjutkan dengan proses pendeteksian jatuh dengan masing-masing metode baik menggunakan *decision tree* maupun  $kNN$ , berikut langkah-langkah yang dilakukan dalam mendeteksian menggunakan 2 metode tersebut:

#### 1) Pendeteksian Jatuh Menggunakan Nilai Treshlold

Pada algoritma ini dilakukan pembagian dua aktivitas, yaitu statis dan dinamis. algoritma ini dibagi menjadi 3 langkah, yaitu analisis terjadinya transisi gerakan, analisis aktivitas, dan analisis postur. Secara umum Perancangan pendeteksian jatuh adalah sebagai berikut:

1. Pertama, sensor accelerometer mencatat pergerakan fisik user berdasarkan sumbu  $x, y, z$ , dan merekam nilai *quaternion*  $q_0, q_1, q_2, q_3$  setiap 200 ms, untuk mendapatkan nilai-nilai tersebut digunakan library Adafruit\_BNO055.
2. Untuk mereduksi adanya noise digunakan low pass filter pada nilai  $x, y, z$ .
3. Menghitung nilai perubahan pada masing-masing sumbu  $dX, dY, dZ$  dan nilai  $\theta$ .
4. Kemudian menghitung nilai  $\alpha$  yaitu percepatan  $\alpha$  dari akar dari pertambahan nilai sensor accelerometer
5. Kemudian mengecek apakah nilai  $|\alpha|$  kurang dari nilai  $\alpha$  *threshold*, jika iya maka transisi pergerakan *user* dianggap statis. Sebaliknya jika tidak, maka terjadi transisi pergerakan dinamis.
6. Kemudian mengecek apakah nilai  $|dX|$  dan  $|dZ|$  melebihi nilai  $|dX|$  dan  $|dZ|$  *threshold*. Jika tidak maka dipastikan *user* mengalami transisi gerakan dinamis namun bukan transisi jatuh, namun jika nilai tersebut melebihi dari nilai *threshold*, maka *user* berpotensi mengalami insiden terjatuh.
7. Kemudian mengecek apakah sudut dari postur *user* adalah mendekati 0, jika iya maka *user* mengalami insiden terjatuh. Untuk mengetahui postur *user*, maka dilakukan perhitungan sudut  $\theta$ . Sudut  $\theta$  adalah sudut antara perangkat bergerak yang berada pada daerah saku kantong terhadap pusat gravitasi.

#### 2) Pendeteksian Jatuh Menggunakan Metode K-Nearest Neighbors

Dalam mendeteksi *user* mengalami insiden terjatuh dengan menggunakan metode  $kNN$  dilakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain, pengklasifikasian pada titik *query instance* yakni pada sistem ini ialah data-data perubahan percepatan pengguna akan digolongkan berdasarkan sejumlah  $K$  objek yang paling dekat dengan titik *query*, jika mayoritas pada  $K$  objek tersebut pada suatu *class* misalnya dalam hal ini ialah *class* jatuh, maka *query instance* tersebut tergolong kedalam *class* jatuh. Pada  $kNN$  jarak antara satu data dengan data yang lain dapat dihitung. Nilai jarak inilah yang digunakan sebagai kedekatan atau kemiripan antara data uji dengan data training. Salah satu yang diperhatikan pada  $kNN$  adalah pemilihan nilai  $K$  yang tepat. Pada nilai  $K$  yang besar bisa menyebabkan distorsi data yang besar, hal itu karena setiap tetangga mempunyai bobot yang sama terhadap data uji. Sedangkan  $K$  yang terlalu kecil bisa menyebabkan terlalu sensitif terhadap noise.

Langkah yang digunakan dalam pendeteksian jatuh dengan menggunakan metode  $kNN$  ialah mula-mula sistem menginisialisasi *data set* nilai  $dX, q_0$  dan  $q_2$  serta

menginisialisasi *data set* target, kemudian menentukan nilai  $K$  pada sistem ini nilai  $K$  adalah 5, berdasarkan hasil uji coba  $K$  pada 5 memiliki tingkat akurasi yang lebih baik. Kemudian pada setiap pembacaan sensor nilai *accelerometer*  $dX, q_0$  dan  $q_2$  dilakukan penghitungan jarak dengan nilai *dataset*  $dX, q_0$  dan  $q_2$  dengan *eculidean distance*. Setelah itu pada nilai jarak dilakukan sortir dari nilai terkecil hingga terbesar sekaligus memasangkan dengan *dataset* target. Setelah itu dilakukan penghitungan dengan  $K$  terdekat apakah termasuk pada target “Jatuh Terdeteksi” atau “Tidak Terjatuh”. Pada *class* target mayoritas merupakan prediksi akhir dari proses metode ini, sehingga dapat disimpulkan bahwa prediksi dari metode  $k-NN$  sesuai dengan pergerakan fisik *user*.

#### D. Global Positioning System (GPS)

Data GPS berupa koordinat titik *latitude* dan *longitude* yang dikirimkan melalui pemberitahuan SMS maupun yang dikirimkan ke *website monitoring* ialah data yang diambil dari GPS/GPRS/GSM *Shield*, namun data dari *Shield* tersebut masih berupa data *raw* dan perlu dilakukan konversi terlebih dahulu agar sesuai dengan standar dan data posisi tersebut valid [6].

#### E. Pengiriman Data Realtime Menggunakan HTTP GET

Pada artikel ini digunakan metode HTTP GET dalam pengiriman data-data lokasi terjadinya insiden terjatuh dari *Shield* ke *database* yang kemudian data-data tersebut ditampilkan dalam sebuah *web*. Dalam artikel ini digunakan *framework* CodeIgniter dengan model MVC (*Model, View, Controller*) untuk membangun *web* dinamis dengan menggunakan PHP. *Model* akan berhubungan langsung dengan *database* untuk memanipulasi data, *view* akan menangani presentasi logika berupa *file* HTML, dan *controller* akan mengatur hubungan antara bagian *model* dan bagian *view*.

#### F. Pengiriman Pemberitahuan Melalui SMS

Sistem ini akan mengirimkan SMS secara otomatis kepada kerabat korban apabila *users* mengalami insiden terjatuh. Data yang dikirimkan berupa koordinat *longitude* dan *latitude* beserta dengan alamat *url* dengan paramater koordinat sehingga dapat dilihat dengan menggunakan *Google Maps*.

Pengiriman pemberitahuan ini menggunakan perintah dari AT Command yang dilakukan pada GSM/GPS *Shield* V3.0 SIM908 [7]. Perintah AT Command yang digunakan dalam sistem ini sebagai berikut :

- AT+CMGF: memberikan spesifikasi format input dan output SMS. 0 untuk Mode PDU, sedangkan 1 untuk mode teks. Respon yang diberikan berupa “OK”.
- AT+CMGS=\” <number>”: untuk memasukkan nomor tujuan dalam pengiriman SMS, diawali dengan kode internasional, untuk Indonesia menggunakan kode 62.

Format yang digunakan dalam mengirimkan pemberitahuan tersebut ialah: Fall alarm from ederly! latitude: <koordinat> longitude: <koordinat> link:<alamat url *Google Maps*>.

### III. HASIL DAN DISKUSI

#### A. Akurasi Deteksi

Uji coba akurasi deteksi dilakukan untuk mengetahui tingkat akurasi algoritma *fall detection* ditinjau berdasarkan

pendeteksian jatuh dengan menggunakan metode *decision tree* maupun *kNN*. Aktivitas fisik *user* yang akan diuji coba adalah aktivitas fisik *user* pada saat terjatuh dan aktivitas fisik *user* pada saat tidak terjatuh. Uji coba ini menggunakan perhitungan *true positive*, *true negative*, *false positive*, dan *false negative*.

Uji coba ini dilakukan sebanyak 20 kali pada setiap aktivitas fisik *user*. Adapun jenis uji coba kasus berdasarkan perhitungan *true positive*, *true negative*, *false positive*, dan *false negative* adalah sebagai berikut:

• **True Positive**

Uji coba dikatakan *true positive* ketika aktivitas jatuh terdeteksi oleh sistem sebagai aktivitas jatuh.

• **True Negative**

Uji coba dikatakan *true negative* ketika aktivitas tidak jatuh terdeteksi oleh sistem sebagai aktivitas tidak jatuh.

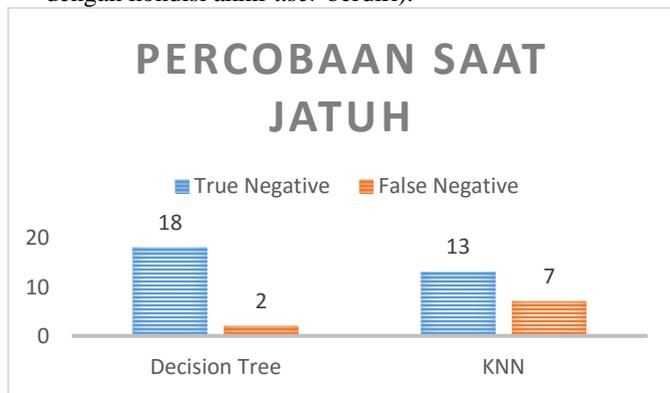
• **False Positive**

Uji coba dikatakan *false positive* ketika aktivitas tidak jatuh namun terdeteksi oleh sistem sebagai aktivitas jatuh.

• **False Negative**

Uji coba dikatakan *false negative* ketika aktivitas jatuh namun terdeteksi oleh sistem sebagai aktivitas tidak jatuh. Adapun aktivitas fisik *user* yang dilakukan uji coba adalah sebagai berikut:

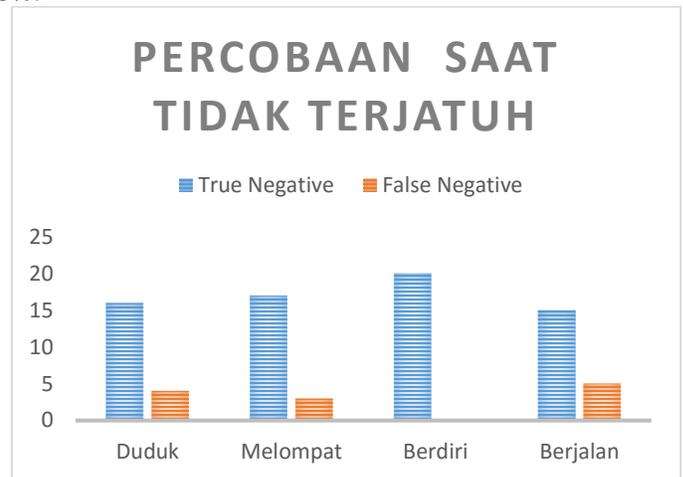
- Aktivitas jatuh (aktivitas *user* dikatakan jatuh ketika *user* dalam keadaan statis menuju keadaan transisi dinamis, yaitu dari keadaan duduk, berdiri, atau berjalan menuju keadaan berbaring atau tergeletak).
- Aktivitas duduk (aktivitas *user* dikatakan duduk secara cepat ketika *user* dalam keadaan berdiri atau berjalan kemudian duduk secara tiba-tiba).
- Aktivitas berjalan (aktivitas *user* dikatakan berjalan ketika *user* dalam keadaan berjalan dari awal hingga akhir).
- Aktivitas melompat (*user* dikatakan melompat ketika *user* dalam keadaan berdiri atau berjalan kemudian melompat dengan kondisi akhir *user* berdiri).



Gambar 3 Grafik Perbandingan Aktivitas Fisik User pada Saat Terjatuh Berdasarkan Metode yang Digunakan

Gambar 1 merupakan grafik perbandingan pendeteksian aktivitas fisik *user* pada saat jatuh antara pendeteksian dengan menggunakan metode *decision tree* dan pendeteksian dengan menggunakan metode *kNN*. Pada proses pendeteksian aktivitas fisik *user* pada saat terjatuh dengan menggunakan perhitungan *true positive* dengan menggunakan metode *decision tree* didapatkan tingkat akurasi sebesar 90%. Sedangkan jika pendeteksian menggunakan metode *kNN* didapatkan tingkat akurasi sebesar 65%.

Pada proses pendeteksian aktivitas fisik *user* pada saat terjatuh dengan perhitungan *false negative* dengan metode *decision tree*, maka akan didapatkan nilai 10%. Sedangkan pendeteksian aktivitas fisik *user* pada saat terjatuh dengan perhitungan *false negative* dengan metode *kNN* adalah sebesar 35%.



Gambar 4 Grafik Perbandingan Aktivitas Fisik User pada Kondisi Tidak Terjatuh dengan menggunakan Metode Decision Tree

Tabel 1 Uji Coba Perhitungan Akurasi Aktivitas Fisik User pada Saat Tidak Terjatuh dengan Parameter True Negative dan False Positive dengan menggunakan Decision Tee

No	Aktivitas	Jumlah Percobaan	True Negative (TN)	False Positive (FP)	% TN	% FP
1	Duduk secara cepat	20	16	4	80	20
2	Melompat	20	17	3	85	15
3	Berdiri	20	20	0	100	0
4	Berjalan	20	15	5	75	25
Jumlah					340	60

Adapun rata-rata nilai *true negative* (TN) dan *false positive* (FP) dari Tabel 5.2 adalah seperti berikut:

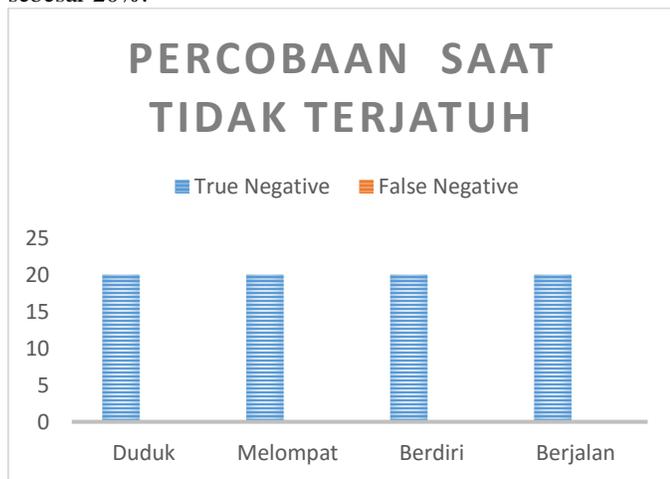
$$\begin{aligned} \text{Rata-rata \% TN} &= \frac{\text{jumlah nilai persentase TN}}{\text{jumlah aktivitas}} \times 100\% \\ &= \frac{(80+85+100+75)}{4} \times 100\% \\ &= \frac{340}{4} \times 100\% \\ &= 85\% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata \% FP} &= \frac{\text{jumlah nilai persentase FP}}{(\text{jumlah aktivitas} \times 100)} \\ &= \frac{60}{40} \\ &= 15\% \end{aligned}$$

$$\begin{aligned} \text{Spesifitas} &= \frac{\text{jumlah nilai persentase TN}}{\text{jumlah nilai persentase (FP+TN)}} \times 100\% \\ &= \frac{85}{(85+15)} \times 100\% \\ &= 85\% \end{aligned}$$

Gambar 4 merupakan grafik uji coba aktivitas fisik *user* pada saat tidak terjatuh dengan menggunakan metode *decision tree* masing-masing uji coba sebanyak 20 kali. Berdasarkan

informasi tersebut dan dari Tabel 1 menunjukkan bahwa untuk masing-masing aktivitas memiliki nilai persentase *true negative* dan *false negative* yang berbeda-beda. Empat aktivitas yang terdiri atas aktivitas duduk secara cepat, aktivitas berjalan, aktivitas melompat, dan aktivitas berdiri memiliki nilai persentase *true negative* sebesar 85% dan nilai *false positive* sebesar 15%. Nilai *true negative* sebesar 100% pada aktivitas berdiri menandakan bahwa aplikasi mendeteksi aktivitas berdiri sebagai aktivitas fisik user dengan kondisi tidak jatuh secara akurat. Pada Aktivitas berjalan yang memiliki nilai *true negative* sebesar 75% dan nilai *false positive* sebesar 25% sebagai presentasi terkecil dalam pendeteksian aktivitas tidak terjatuh. Aktivitas melompat *true negative* sebesar 85% dan nilai *false positive* sebesar 15%. Hal tersebut menandakan bahwa aplikasi mendeteksi aktivitas melompat sebagai aktivitas fisik user dalam kondisi tidak jatuh sebanyak 17 kali dari 20 percobaan yang ada dan mendeteksi sebagai keadaan tidak terjatuh sebagai aktivitas duduk sebanyak 3 kali dari 20 percobaan yang ada. Pada aktivitas duduk, memiliki nilai persentase *true negative* sebesar 80% dan nilai *false positive* sebesar 20%.



Gambar 5 Grafik Perbandingan Aktivitas Fisik User pada Kondisi Tidak Terjatuh dengan menggunakan Metode kNN

Tabel 2 Uji Coba Perhitungan Akurasi Aktivitas Fisik User pada Saat Tidak Terjatuh dengan Parameter True Negative dan False Positive dengan menggunakan KNN

No	Aktivitas	Jumlah Percobaan	True Negative (TN)	False Positive (FP)	% TN	% FP
1	Duduk secara cepat	20	18	2	90	10
2	Melompat	20	20	0	100	0
3	Berdiri	20	20	0	100	0
4	Berjalan	20	20	0	100	0
Jumlah					400	0

Adapun rata-rata nilai *true negative* (TN) dan *false positive* (FP) dari Tabel 5.2 adalah seperti berikut:

$$\begin{aligned} \text{Rata-rata \% TN} &= \frac{\text{jumlah nilai persentase TN}}{\text{jumlah aktivitas}} \times 100\% \\ &= \frac{(90+100+100+100)}{4} \times 100\% \\ &= \frac{390}{4} \times 100\% \end{aligned}$$

$$\begin{aligned} &= 97,5\% \\ \text{Rata-rata \% FP} &= \frac{\text{jumlah nilai persentase FP}}{(\text{jumlah aktivitas} \times 100)} \\ &= \frac{10}{400} \\ &= 2,5\% \end{aligned}$$

$$\begin{aligned} \text{Spesifitas} &= \frac{\text{jumlah nilai persentase TN}}{\text{jumlah nilai persentase (FP+TN)}} \times 100\% \\ &= \frac{97,5}{(90+10)} \times 100\% \\ &= 97,5\% \end{aligned}$$

Gambar 5 merupakan grafik uji coba aktivitas fisik user pada saat tidak terjatuh dengan menggunakan metode kNN masing-masing uji coba sebanyak 20 kali. Berdasarkan informasi tersebut dan dari informasi Tabel 2 menunjukkan bahwa untuk aktivitas melompat, berdiri dan berjalan memiliki nilai persentase *true negative* dan *false negative* yang sama yaitu untuk presentasi *true negative* 100%, Hal tersebut menandakan bahwa ketiga aktivitas dalam kondisi tidak terjatuh tersebut benar-benar terdeteksi tidak jatuh sebanyak 20 kali dari 20 percobaan yang dilakukan. Sedangkan untuk aktivitas duduk memiliki persentase *true negative* sebesar 90%, dari 20 kali hasil uji coba, sebanyak 2 kali percobaan aktivitas duduk yang terdeteksi sebagai aktivitas jatuh.

#### IV. KESIMPULAN/RINGKASAN

Dari penelitian ini, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Nilai quaternion ( $q0, q1, q2, q3$ ) dapat digunakan untuk mendeteksi seseorang dalam insiden jatuh dengan menggunakan metode *Decision Tree* dan metode *k-NN*.
2. Dengan menggunakan *Decision Tree* akurasi dalam pendeteksian orang jatuh sebesar 90%, sedangkan dengan menggunakan metode *k-NN* hanya 65%.
3. Sensor orientasi *motion Shield* yang digunakan dapat mendeteksi seseorang dalam keadaan terjatuh, menggunakan analisis sudut rotasi sebelum dan sesudah terjadinya insiden terjatuh.
4. Sensor *accelerometer* mendeteksi percepatan pada sumbu x,y,z. Pada keadaan normal nilai percepatan berkisar antara 9.8 m/s<sup>2</sup>
5. Pengiriman data lokasi jatuh melalui SMS kepada kerabat korban dan pengiriman data ke server membutuhkan waktu paling lama 5 detik.
6. Tingkat akurasi sistem *fall detection* untuk mendeteksi gerakan selain jatuh dengan akurasi 85 % dengan menggunakan metode *decision tree* dan 95% dengan metode kNN.

#### UCAPAN TERIMA KASIH

Penulis D.N. mengucapkan terima kasih kepada Allah SWT, kedua orangtua dan keluarga penulis, dosen pembimbing, kepala jurusan Teknik Informatika, segenap dosen dan karyawan, kerabat-kerabat dekat, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan artikel ini.

## DAFTAR PUSTAKA

- [1] Baraff L.J., Della Penna R., Williams N., Sanders A., Practice guideline for the ED management of falls in community-dwelling elderly persons, 1997, p. 480–492.
- [2] N. Carroll, P. Slattum dan F. Cox, The cost of falls among the community-dwelling elderly, PubMed, 2005.
- [3] W. DEIDRE , U. S. L. NAYAK dan B. ISAACS, How dangerous are falls in old people at home?, PubMed.
- [4] K. B. Firdaus, “APLIKASI ACCELEROMETER SEBAGAI KENDALI (JOYSTICK),” Surabaya.
- [5] H. Z. Fallin Wu, “Development of a Wearable-Sensor-Based Fall Detection System,” *International Journal of Telemedicine and Applications*, no. Hindawi Publishing Corporation, p. 11, 2014.
- [6] “GPS/GPRS/GSM Module V3.0 (SKU:TEL0051),” [Online]. Available: [http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM\\_Module\\_V3.0\\_\(SKU:TEL0051\)](http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)). [Diakses 17 Juni 2016].
- [7] “SIMCOM,” [Online]. Available: [http://www.simcom.ee/documents/gsm-gprs/sim900/SIM900\\_AT%20Command%20Manual\\_V1.09.pdf](http://www.simcom.ee/documents/gsm-gprs/sim900/SIM900_AT%20Command%20Manual_V1.09.pdf). [Diakses 13 April 2016].