

Sistem Deteksi Orang Jatuh Dengan Menggunakan Sensor Kamera Kinect Dengan Metode AdaBoost

Satria Perwira*¹, Muhammad Idham Ananta Timur², Agus Harjoko³

¹Prodi Elektronika dan Instrumentasi, DIKE, FMIPA, UGM, Yogyakarta, Indonesia

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta, Indonesia

e-mail: *¹satria.perwira@mail.ugm.ac.id, ²idham@ugm.ac.id, ³aharjoko@ugm.ac.id

Abstrak

Kasus jatuhnya orang tua berumur 65 tahun keatas atau lansia sangat membahayakan kesehatan mereka karena dapat mengakibatkan patah tulang pinggul, cedera kepala, bahkan kematian. Pertolongan harus diberikan dengan cepat apabila terjadi jatuh sehingga diperlukan sistem yang dapat mendeteksi jatuh dengan otomatis dan unobtrusive. Ada tiga pendekatan dalam pembuatan sistem deteksi jatuh yaitu pendekatan wearable, ambience, dan berbasis vision. Pendekatan wearable memiliki kekurangan yaitu sifatnya yang obtrusive sedangkan pendekatan ambience memiliki kecenderungan untuk memiliki nilai false positive yang tinggi. Pendekatan berbasis vision dipilih karena bersifat unobtrusive dan memiliki nilai false positive yang rendah. Penelitian ini menggunakan kamera Kinect karena kemampuannya dalam mengekstraksi data skeletal.

Metode yang digunakan dalam sistem deteksi jatuh yang dibuat yaitu metode AdaBoost dan metode thresholding nilai kecepatan titik sendi. Metode thresholding digunakan sebagai perbandingan dengan metode AdaBoost. Kedua metode menggunakan data skeletal dari subyek yang terekam oleh kamera Kinect. Metode AdaBoost mencocokkan data skeletal dengan model yang sudah dibuat sedangkan metode thresholding membandingkan nilai kecepatan titik sendi dengan nilai threshold. Pengujian dilakukan dengan menggunakan data latih, data uji, dan data real-time. Nilai rata-rata akurasi yang diperoleh dari pengujian sistem yaitu dengan metode AdaBoost sebesar 91,75% dan dengan metode thresholding sebesar 68,22%

Kata kunci—Deteksi jatuh, Kinect, AdaBoost

Abstract

Fall cases of elderly people aged 65 or above put their health at risk because it could lead to hip bone fracture, concussion, even death. Immediate help is needed if fall happened which is why an automatic and unobtrusive fall detection system is needed. There are three approaches in fall detection system; wearable, ambience, and vision-based. Wearable approach has the drawback of its obtrusive nature while ambience approach is prone to high false positive value. Vision-based approach is chosen because its unobtrusive nature and low false positive value. This study uses Kinect camera because of its ability on extracting skeletal data.

The methods that are used in the fall detection system are AdaBoost method and joint velocity thresholding method. Thresholding method is used as a comparison to AdaBoost method. Both methods use skeletal data from the subject recorded by the Kinect camera. AdaBoost method compares the skeletal data with model that was made before while thresholding method compares the joint velocity value with the threshold value. System test is done using training data, test data, and real-time data. The average accuracy obtained from the system test with AdaBoost method is 91.75% and with thresholding method is 68.22%.

Keywords—Fall detection, Kinect, AdaBoost.

1. PENDAHULUAN

Kasus jatuhnya orang tua berumur 65 tahun keatas, atau lansia, seringkali menjadi penyebab utama dari patah tulang panggul, bahkan beresiko menyebabkan cedera kepala atau bahkan menjadi penyebab kematian [1]. Terjatuhnya orang tua membuat pihak keluarga khawatir akan kesehatan mereka sehingga biasanya harus ada seseorang yang selalu mengawasi orang tua yang bersangkutan, bisa dari pihak keluarga sendiri atau dengan menggunakan jasa orang lain sebagai pengasuh. Namun tentu saja pengasuh tidak bisa mengawasi terus menerus karena pasti ada kesibukan lain. Oleh karena itu perlu dibuat sistem yang bisa mengawasi orang tua sehingga apabila terjatuh bisa terdeteksi dengan cepat dan bisa memberikan pemberitahuan pada pengasuh atau pihak keluarga.

Sistem deteksi orang jatuh memiliki beberapa pendekatan seperti pendekatan wearable dan ambience. Pendekatan wearable menggunakan perangkat yang dilengkapi sensor yang dapat digunakan [2]. Pendekatan wearable digunakan langsung oleh subyek yang dipantau sehingga jika subyek terjatuh dapat langsung terdeteksi. Namun kekurangan dari pendekatan wearable adalah perangkat harus digunakan secara terus menerus agar bisa mendeteksi jatuh sehingga bisa saja memberikan perasaan tidak nyaman. Pendekatan ambience menggunakan parameter berupa kondisi lingkungan untuk mendeteksi jatuh seperti getaran lantai [3]. Ketika seseorang terjatuh tentu saja akan menyebabkan lantai yang ditempatinya bergetar sehingga sistem bisa mengetahui apabila ada orang jatuh. Namun kekurangan dari pendekatan ambience yaitu kecenderungannya untuk memiliki nilai false positive yang tinggi.

Pembuatan sistem deteksi jatuh dengan pendekatan wearable dan ambience menunjukkan sulitnya untuk membuat sistem deteksi jatuh yang otomatis, bersifat unobtrusive, dan tidak memiliki kecenderungan untuk memiliki nilai false positive yang tinggi. Namun hal ini dapat diatasi dengan menggunakan pendekatan lain dalam mendeteksi jatuh. Pendekatan lain yang dimaksud yaitu pendekatan berbasis vision. Pendekatan berbasis vision melakukan deteksi jatuh dengan menggunakan bantuan kamera. Penggunaan kamera dapat mengatasi kekurangan dari sistem deteksi orang jatuh dengan pendekatan wearable dan ambience karena sifatnya yang unobtrusive dan cenderung memiliki nilai false positive yang rendah.

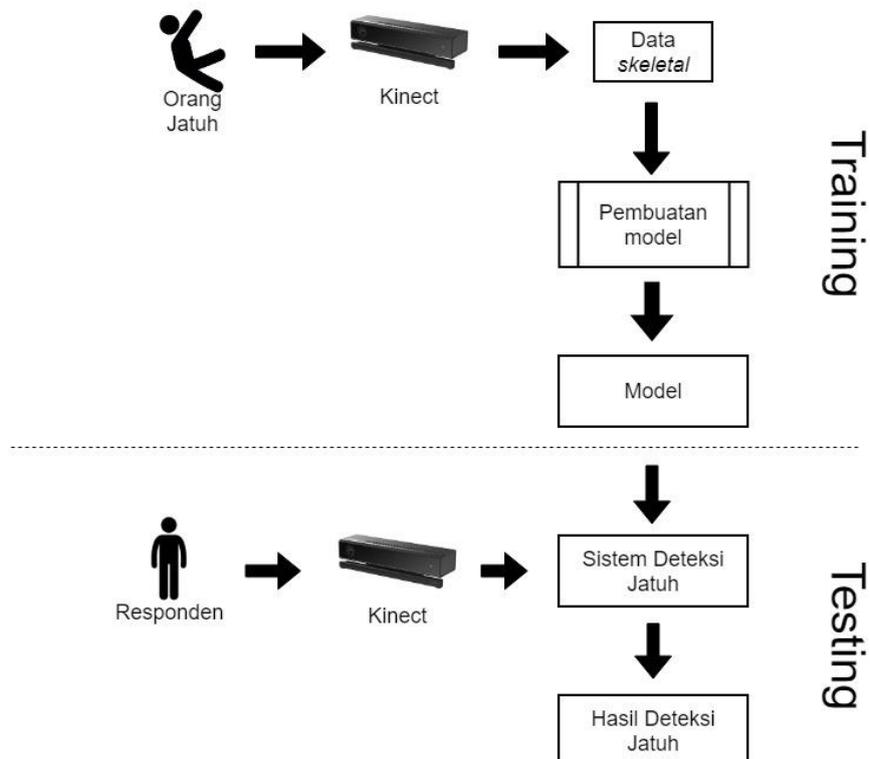
Penggunaan kamera RGB sebagai alat deteksi jatuh juga masih memiliki kekurangan seperti ketidakmampuannya dalam merekam dalam kondisi minim pencahayaan dan kekhawatiran akan privacy invasion, kondisi dimana seseorang merasa ranah privasinya terlanggar karena berada di bawah pengawasan konstan. Hal ini dapat diatasi dengan penggunaan kamera RGB dan depth. Kamera depth dapat melakukan perekaman dalam kondisi minim pencahayaan karena menggunakan sensor inframerah. Kamera RGB dan depth yang digunakan yaitu kamera Kinect karena kemampuannya dalam mendeteksi data *skeletal* dari subyek yang terekam oleh Kinect [4].

Penelitian untuk mendeteksi orang jatuh dengan menggunakan Kinect sudah banyak dilakukan sebelumnya dengan berbagai macam metode, seperti menghitung perubahan kecepatan koordinat sumbu x, sumbu y, dan sumbu z dari objek, perhitungan dari bounding box pada objek, perhitungan jarak dan sudut dari kerangka manusia terhadap lantai, dan lain lain. Penelitian [5] menggunakan perhitungan perubahan kecepatan posisi dari rangka yang sudah terdeteksi. Posisi dari rangka direpresentasikan pada sumbu x sebagai lebar, sumbu y sebagai tinggi, dan sumbu z sebagai kedalaman. Perhitungan posisi dilakukan dengan mengurangi nilai terbesar dengan nilai terkecil dari setiap sumbu. Kemudian untuk menghitung kecepatan dilakukan perbandingan nilai dari setiap posisi saat ini dengan nilai dari setiap posisi pada frame sebelumnya. Apabila hasilnya melebihi dari threshold yang sudah ditentukan, maka berarti ada yang terjatuh.

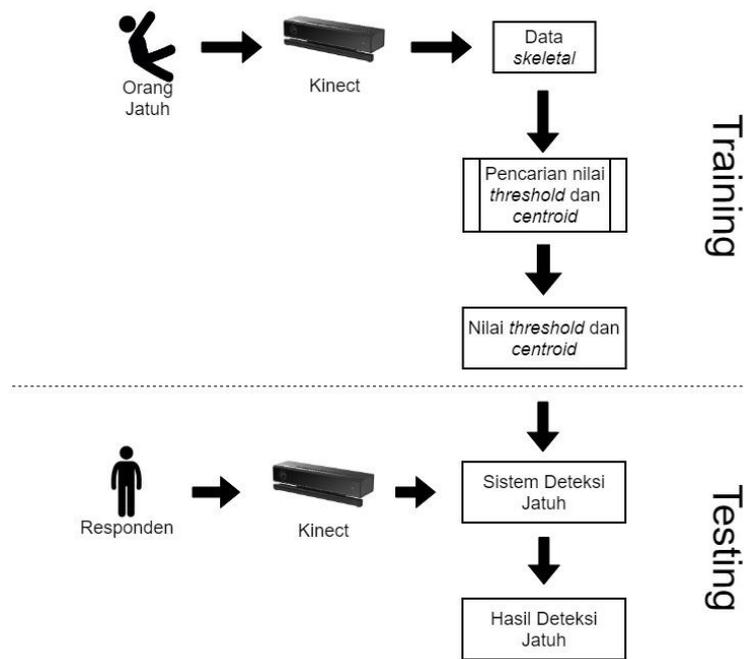
Selain dari menghitung perubahan kecepatan dari setiap sumbu, ada juga penelitian yang melakukan perhitungan perubahan kecepatan dari titik-titik rangka pada rangka manusia [6]. Titik rangka utama yang digunakan dalam penelitian ini adalah titik rangka yang merepresentasikan kepala. Titik rangka lain yang digunakan adalah bahu, pinggul, dan lutut untuk menentukan posisi dari subyek. Apabila perubahan kecepatan pada kepala melebihi threshold yang sudah ditentukan, maka akan dilakukan pengecekan untuk menentukan posisi subyek dengan melihat posisi dari bahu dan pinggul terhadap lutut. Apabila posisi menunjukkan bahwa subyek berada di lantai, maka menandakan bahwa terjadi jatuh

2. METODE PENELITIAN

Sistem deteksi yang dilakukan pada penelitian ini merupakan sistem deteksi jatuh. Metode yang digunakan dalam deteksi jatuh yaitu metode AdaBoost dan metode *thresholding* nilai kecepatan titik sendi. Sistem deteksi jatuh dengan metode AdaBoost memanfaatkan *tools* yang terdapat pada Kinect SDK yaitu *Visual Gesture Builder (VGB)* sedangkan sistem deteksi jatuh dengan metode *thresholding* nilai kecepatan titik sendi tidak membutuhkan *tools* tambahan hanya perlu menggunakan kamera *depth* pada Kinect. Diagram blok proses perancangan hingga pengujian sistem deteksi jatuh dengan metode AdaBoost dapat dilihat pada Gambar 1 dan untuk sistem deteksi jatuh dengan metode *thresholding* dapat dilihat pada Gambar 2.



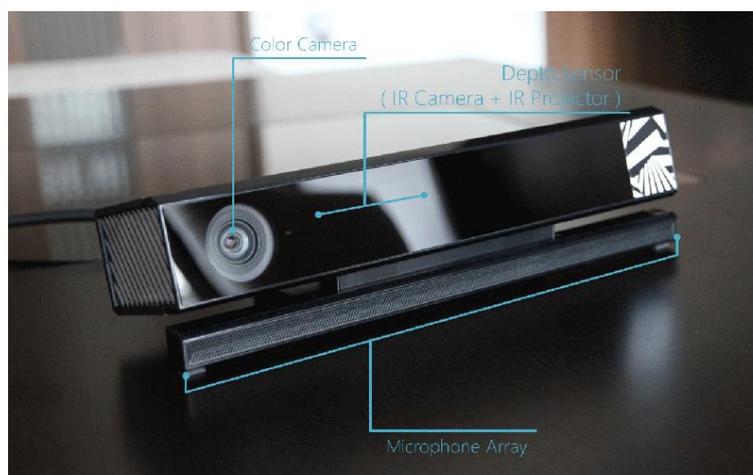
Gambar 1 Diagram blok sistem deteksi jatuh dengan metode AdaBoost



Gambar 2 Diagram blok sistem deteksi jatuh dengan metode *thresholding*

2.1 Kamera RGB dan *depth*

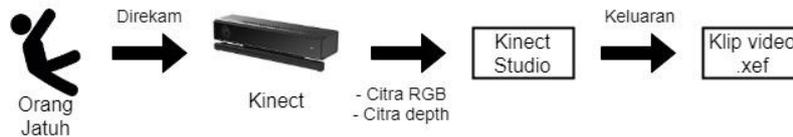
Kamera RGB dan *depth* atau kamera RGB-d adalah kamera yang memiliki sensor RGB dan sensor *depth*. Kamera RGB-d memiliki keunggulan dibandingkan kamera RGB biasa karena dengan menggunakan sensor *depth*, karena RGB-d dapat melakukan akuisisi data dalam keadaan minim pencahayaan. Salah satu contoh kamera RGB-d adalah kamera Kinect. Kinect adalah piranti buatan Microsoft yang merupakan perangkat pelengkap konsol permainan yang dibuat oleh Microsoft. Seiring perkembangan zaman, Kinect mulai digunakan sebagai alat akuisisi data karena memiliki fungsionalitas yang baik seperti kemampuannya dalam membaca gerakan obyek yang tertangkap dalam jangkauan sensornya. Sensor yang terdapat pada Kinect yaitu sensor RGB, sensor *depth*, dan mikrofon *multi-array* [7]. Spesifikasi Kinect dapat dilihat pada Gambar 3



Gambar 3 Spesifikasi Kinect

2.2 Akuisisi Data

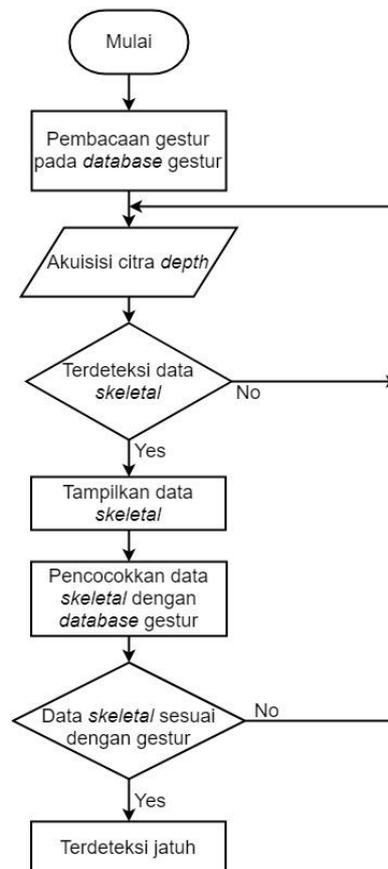
Data yang digunakan pada penelitian ini terbagi menjadi data latih, data uji, dan data *-real-time*. Akuisisi data dilakukan dengan menggunakan aplikasi perekam video dari Kinect yaitu Kinect Studio. Kinect Studio digunakan karena kemampuannya dalam merekam data *skeletal* dari subyek yang terekam oleh kamera Kinect. Keluaran dari proses perekaman yaitu klip video dengan format .xef. Proses akuisisi data dapat dilihat pada Gambar 3



Gambar 4 Diagram Blok Akuisisi Data

2.3 Metode AdaBoost

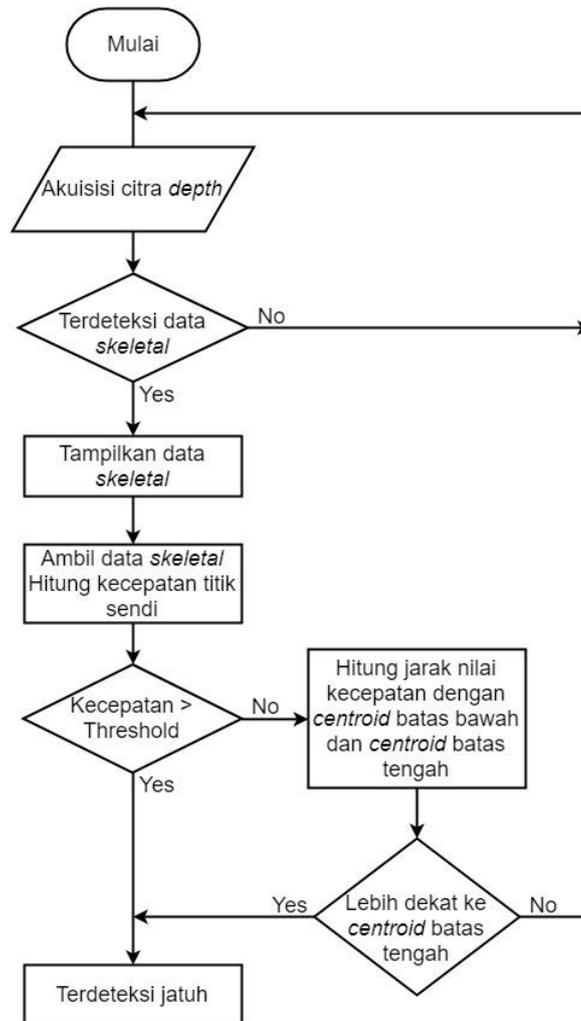
AdaBoost adalah sebuah algoritma yang menggabungkan beberapa *weak classifier* untuk mendapatkan hasil yang akurat [8]. AdaBoost digunakan oleh Kinect sebagai salah satu metode pengenalan gestur secara diskrit [9]. AdaBoost digunakan sebagai metode deteksi jatuh karena jatuh termasuk gerakan diskrit. Sistem deteksi orang jatuh dengan metode AdaBoost dilakukan dengan mencocokkan data *skeletal* dengan model yang tersimpan dalam *database* gestur. Pembuatan model dilakukan dengan menggunakan *tools* dari Kinect yaitu *Visual Gesture Builder* (VGB) dengan menggunakan metode AdaBoost. Proses pendeteksian jatuh dengan metode AdaBoost dapat dilihat pada Gambar 4.



Gambar 5 Diagram alir proses deteksi jatuh dengan metode AdaBoost

2.4 Metode *thresholding* nilai kecepatan titik sendi

Metode *thresholding* nilai kecepatan titik sendi digunakan karena ketika seseorang terjatuh nilai kecepatan pada titik sendi tertentu akan meningkat secara signifikan. Titik sendi yang digunakan pada penelitian ini yaitu titik sendi bagian kepala dan leher. Titik sendi kepala dan leher dipilih karena ketika seseorang terjatuh kepala dan leher merupakan bagian tubuh yang mengalami perubahan posisi relatif cepat. Diagram alir dari sistem deteksi jatuh dengan metode *thresholding* nilai kecepatan titik sendi dapat dilihat pada Gambar 5



Gambar 6 Diagram alir proses deteksi jatuh dengan metode *thresholding*

Nilai kecepatan didapatkan dari menghitung jarak posisi pada frame t dengan frame $t-1$ dan dibagi dengan $1/30$. Nilai $1/30$ digunakan karena kamera Kinect yang digunakan memiliki *framerate* 30fps. Nilai jarak dihitung dengan menggunakan jarak *euclidean* [10]. Nilai kecepatan digunakan untuk mendeteksi jatuh dengan membandingkan nilai kecepatan titik sendi dengan nilai *threshold*. Apabila nilai kecepatan titik sendi lebih besar dari nilai *threshold* maka akan dikategorikan sebagai jatuh. Namun apabila nilai kecepatan titik sendi lebih kecil dari *threshold* maka akan dilakukan pengecekan lebih lanjut dengan menggunakan metode K-Means. Nilai kecepatan titik sendi akan dihitung jaraknya terhadap *centroid* yang digunakan. Apabila lebih dekat dengan *centroid* batas bawah maka akan dikategorikan sebagai tidak jatuh sedangkan jika lebih dekat dengan *centroid* batas tengah maka akan dikategorikan sebagai jatuh.

3. HASIL DAN PEMBAHASAN

Hasil yang sudah didapat dari pengujian sistem yaitu berupa nilai akurasi, presisi, sensitivitas, spesifisitas, dan F-Score. Setiap nilai yang didapat menunjukkan karakteristik tertentu dari sistem yang telah dibuat. Nilai akurasi menunjukkan seberapa baik sistem dalam membedakan aktivitas jatuh dengan non-jatuh. Nilai presisi menunjukkan seberapa baik sistem mendeteksi jatuh dengan benar. Nilai sensitivitas menunjukkan seberapa baik sistem dalam membedakan jatuh dengan non-jatuh. Nilai spesifisitas menunjukkan seberapa baik sistem dalam membedakan non-jatuh dengan jatuh. Nilai F-Score menunjukkan seberapa baik sistem dalam mendeteksi jatuh. Hasil dari pengujian sistem deteksi jatuh dengan metode AdaBoost tersaji pada Tabel 1.

Tabel 1 Hasil Pengujian Sistem Deteksi Jatuh dengan Metode AdaBoost

Nilai	Metode AdaBoost			
	Data Latih	Data Uji	Data <i>real-time</i>	Rata-rata
Akurasi	94,26%	91,67%	89,33%	91,75%
Presisi	91,30%	87,80%	75,76%	84,95%
Sensitivitas	100%	100%	100%	100%
Spesifisitas	85,71%	79,17%	84%	82,96%
F-Score	95,46%	93,51%	86,21%	91,73%

Terlihat pada Tabel 1 hasil dari pengujian sistem deteksi jatuh dengan metode AdaBoost, baik pengujian dengan data latih, data uji, maupun data *real-time* dapat dibidang sangat baik dalam mendeteksi jatuh. Sistem salah mendeteksi aktivitas non-jatuh sebagai jatuh terjadi pada aktivitas berbaring. Dari segi gerakan, jatuh dengan berbaring memang sedikit mirip hanya berbeda kecepatan dalam melakukannya. Sistem dapat bekerja dengan baik karena dalam mendeteksi jatuh dengan metode AdaBoost ada banyak fitur yang digunakan. Data *skeletal* dicek dengan menggunakan fitur-fitur yang dipilih pada saat proses training menggunakan aplikasi Visual Gesture Builder (VGB). Fitur-fitur yang digunakan dapat dilihat pada Gambar 6

```

Top 10 contributing weak classifiers:
VelocityY( Head ) using inferred joints, fValue < -1.600003, alpha = 1.383987
VelocityZ( SpineShoulder ) rejecting inferred joints, fValue < -0.300003, alpha = 0.527709
VelocityY( ShoulderRight ) rejecting inferred joints, fValue < -0.200003, alpha = 0.371246
VelocityY^2( ShoulderLeft ) rejecting inferred joints, fValue >= 0.200000, alpha = 0.361591
VelocityY( Head ) using inferred joints, fValue >= -1.600003, alpha = 0.361136
DiffMuscleForceZ( Head, SpineMid ) using inferred joints, fValue >= 0.600000, alpha = 0.359060
DiffMuscleForceY( Head, SpineBase ) using inferred joints, fValue >= 0.800000, alpha = 0.343504
DiffPositionZ( Head, SpineBase ) rejecting inferred joints, fValue >= 0.000000, alpha = 0.335176
VelocityY( ShoulderLeft ) rejecting inferred joints, fValue < -0.100003, alpha = 0.286092
Angles( SpineMid, Head, Neck ) using inferred joints, fValue < 32.000000, alpha = 0.276911

```

Gambar 7 Fitur-fitur yang digunakan dalam proses pembuatan model dengan metode AdaBoost

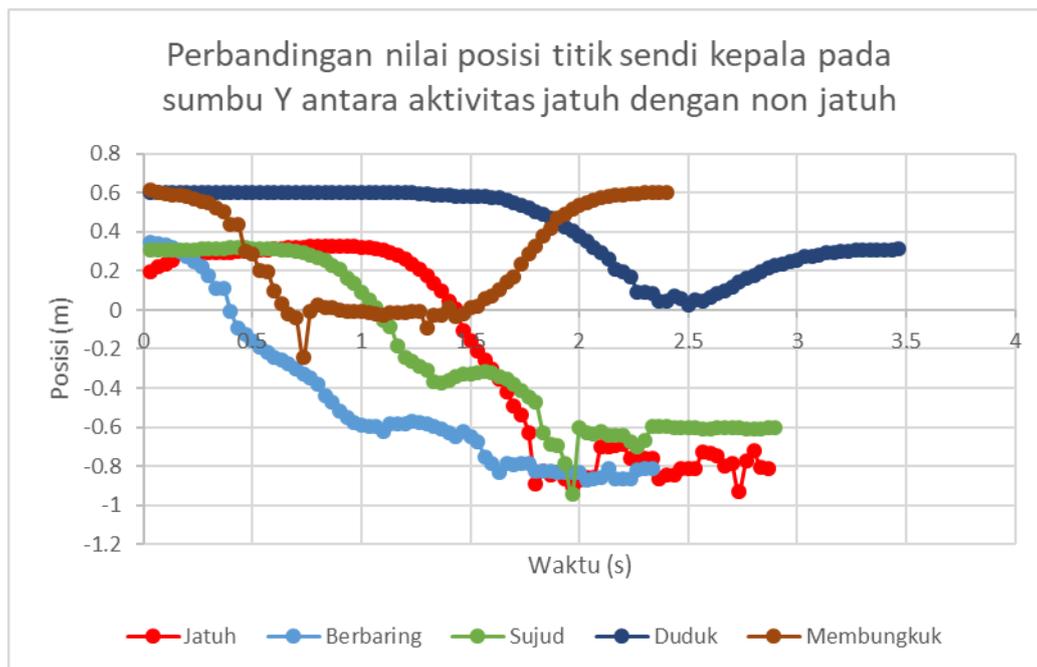
Berdasarkan Gambar 6 terlihat bahwa metode AdaBoost juga melakukan perhitungan nilai kecepatan titik sendi sebagai salah satu fitur untuk mendeteksi jatuh. Namun proses perhitungan nilai kecepatan titik sendi dan pemilihan titik sendi yang digunakan tidak diketahui karena program VGB yang digunakan untuk proses *training* bersifat *closed source*.

Setelah dilakukan pengujian sistem deteksi jatuh dengan metode AdaBoost, dilakukan pengujian sistem dilakukan sistem deteksi jatuh dengan metode *thresholding* nilai kecepatan titik sendi. Hasil dari pengujian sistem deteksi jatuh dengan metode AdaBoost tersaji pada Tabel 2.

Tabel 2 Hasil Pengujian Sistem Deteksi Jatuh dengan Metode *Thresholding*

Nilai	Metode <i>thresholding</i>			
	Data Latih	Data Uji	Data <i>real-time</i>	Rata-rata
Akurasi	75%	65%	64,67%	68,22%
Presisi	71,30%	67,44%	48,48%	62,41%
Sensitivitas	97,61%	80,56%	96%	91,39%
Spesifisitas	41,07%	41,67%	49%	43,91%
F-Score	82,41%	73,42%	64,43%	73,42

Terlihat pada Tabel 2 performa sistem deteksi jatuh dengan metode *thresholding* sudah cukup baik. Namun kekurangan dari sistem deteksi jatuh dengan metode *thresholding* adalah tingginya nilai *false positive*. Ini disebabkan karena sistem hanya menghitung nilai kecepatan titik sendi dan membandingkannya dengan nilai *threshold* tanpa mengetahui gerakan apa yang dilakukan sehingga meskipun gerakan yang dilakukan termasuk aktivitas non-jatuh namun kecepatannya melebihi nilai *threshold* atau pada perhitungan jarak terhadap *centroid* lebih dekat ke C_t , sistem akan mengklasifikasikan aktivitas sebagai jatuh. Nilai *false positive* untuk aktivitas non-jatuh terbesar terjadi pada gerakan sujud dan berbaring karena memang perubahan posisi pada titik sendi kepala dan leher pada waktu yang relatif singkat cukup signifikan jika dibandingkan dengan gerakan non-jatuh lainnya seperti duduk atau membungkuk. Gambar 7 menunjukkan kemiripan antara gerakan berbaring dan sujud dengan jatuh



Gambar 8 Grafik perbandingan perubahan posisi titik sendi kepala pada sumbu Y antara aktivitas jatuh dengan non jatuh

Terlihat pada Gambar 7 kemiripan perubahan posisi pada aktivitas jatuh dengan aktivitas berbaring dan sujud. Aktivitas membungkuk juga memiliki sedikit kemiripan jika dibandingkan dengan aktivitas jatuh namun tidak semirip aktivitas berbaring atau sujud, sehingga sistem akan terkadang mendeteksi membungkuk sebagai jatuh. Namun untuk aktivitas duduk di kursi sangat terlihat perbedaannya jika dibandingkan dengan ketiga gerakan lainnya sehingga sistem selalu mendeteksi duduk sebagai gerakan non jatuh.

Perbandingan antara hasil pengujian sistem deteksi jatuh dengan metode AdaBoost, dengan sistem deteksi jatuh dengan metode *thresholding* dilakukan untuk mengetahui perbedaan performa antara kedua metode. Perbandingan nilai rata-rata hasil antara metode AdaBoost dengan metode *thresholding* dapat dilihat pada Tabel 3

Tabel 3 Perbandingan rata-rata hasil antara metode AdaBoost dengan metode *thresholding*

Nilai	Rata-rata hasil	
	Metode AdaBoost	Metode <i>thresholding</i>
Akurasi	91,75%	68,22%
Presisi	84,95%	62,41%
Sensitivitas	100%	91,39%
Spesifisitas	82,96%	43,91%
F-Score	91,75%	73,42

Terlihat pada Tabel 3 rata-rata hasil pengujian sistem deteksi jatuh dengan metode AdaBoost jauh lebih baik dibandingkan hasil pengujian sistem deteksi jatuh dengan metode *thresholding*. Hal ini didukung dengan nilai sensitivitas dan spesifisitas yang tinggi pada pengujian sistem deteksi jatuh dengan metode AdaBoost. Nilai sensitivitas dan spesifisitas yang tinggi menunjukkan bahwa sistem dapat membedakan antara jatuh dengan non-jatuh dengan baik. Meskipun metode AdaBoost dapat mendeteksi jatuh dengan sangat baik, metode ini terkadang masih salah mendeteksi aktivitas non-jatuh sebagai jatuh seperti berbaring dan sujud. Namun kekurangan ini dapat diatasi dengan menggunakan nilai *confidence level* sebagai *threshold* agar sistem dapat mendeteksi jatuh lebih akurat karena nilai *confidence level* merepresentasikan kemiripan data *skeletal* dengan model. Alasan lain mengapa metode AdaBoost lebih baik dibandingkan metode *thresholding* dalam mendeteksi jatuh yaitu karena nilai F-Score nya yang cenderung stabil. Nilai F-Score yang cenderung stabil menunjukkan bahwa sistem deteksi jatuh dengan metode AdaBoost dapat mendeteksi jatuh dengan baik meskipun menggunakan data yang berbeda.

4. KESIMPULAN

Kesimpulan yang didapatkan dari penelitian yang sudah dilakukan yaitu metode AdaBoost terbukti dapat mendeteksi jatuh dengan sangat baik dengan nilai F-Score yang cenderung stabil meskipun dilakukan pengujian dengan data yang berbeda dan nilai rata-rata yang tinggi yaitu 91,73% jika dibandingkan dengan metode *thresholding*. Performa dari metode *thresholding* tidak sebaik metode AdaBoost karena metode *thresholding* hanya melihat nilai kecepatan titik sendi tanpa mengetahui jenis gerakan yang dilakukan. Meski begitu baik metode AdaBoost maupun metode *thresholding* sering mengkategorikan aktivitas berbaring sebagai jatuh karena kemiripannya dengan jatuh baik dari segi nilai kecepatan titik sendi maupun gerakan

5. SARAN

Beberapa saran terkait penelitian yang telah dilakukan yaitu pengembangan agar sistem dapat melakukan deteksi jatuh ketika ada lebih dari satu orang pada satu ruangan. Saran lainnya yaitu penggunaan nilai *confidence value* pada metode AdaBoost agar dalam pendeteksian orang jatuh bisa lebih akurat sehingga bisa mengurangi nilai *false positive* dalam pendeteksian jatuh dengan metode AdaBoost. Untuk metode *thresholding*, digabungkan dengan pengecekan posisi titik sendi untuk mengurangi nilai *false positive*.

DAFTAR PUSTAKA

- [1] G. Bergen and M. R. Stevens, "Falls Prevention Awareness Day — Falls and Fall Injuries Among Adults Aged ≥ 65 Years — United States , 2014," vol. 65, no. 37, 2016.
- [2] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable Sensors for Reliable Fall Detection," *Clin. Ter.*, vol. 85, no. 2, pp. 179–204, 2005.
- [3] M. Alwan, P. J. Rajendran, S. Kelli, D. Mack, S. Dalali, and M. W. I, "A Smart and Passive Floor-Vibration Based Fall Detector for Elderly," pp. 3–7, 2006.
- [4] L. Jamhoury, "Understanding Kinect V2 Joints and Coordinate System," 2018. [Online]. Available: medium.com/@lisajamhoury/understanding-kinect-v2-joints-and-coordinate-system-4f4b90b9df16. [Accessed: 21-Nov-2018].
- [5] K. Gunadi, Liliana, and J. Tjitrokusmo, "Fall detection application using kinect," *Proc. - 2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol. Build. Intell. Through IOT Big Data, ICSIIT 2017*, vol. 2018–Janua, pp. 279–282, 2018.
- [6] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil, "Human Fall Detection from Depth Images using Position and Velocity of Subject," *Procedia Comput. Sci.*, vol. 105, no. Iris 2016, pp. 131–137, 2017.
- [7] Y. Angal and A. Jagtap, "Fall Detection System for Older People," *2016 IEEE Int. Conf. Adv. Electron. Commun. Comput. Technol.*, pp. 262–266, 2016.
- [8] G. Sriram, M. Vivek, S. K. Roy, and P. Sharan, "Spectral Analysis of Photonic crystal based Bio-Sensor using AdaBoost Algorithm," *2015 Int. Conf. Commun. Signal Process.*, pp. 1806–1810, 2015.
- [9] M. Rahman, *Beginning Microsoft Kinect for Windows SDK 2.0: Motion and Depth Sensing for Natural User Interface*. Montreal: Apress, 2017.
- [10] A. B. Rathod, "A Comparative Study on Distance Measuring Approches for Permutation Representations," *2016 IEEE Int. Conf. Adv. Electron. Commun. Comput. Technol.*, pp. 251–255, 2016.