

# Sistem *Fall Detection* Pada Orang Lanjut Usia Menggunakan Metode Klasifikasi *Support Vector Machine* (SVM)

<sup>1</sup> Nadya Putri Yuliani, <sup>2</sup> Raditiana Patmasari, <sup>3</sup> Yuli Sun Hariyani

<sup>1,2</sup> S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup> D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

<sup>1</sup> nadyayuliani@student.telkomuniversity.ac.id , <sup>2</sup> raditiana@telkomuniversity.ac.id , <sup>3</sup> yulisun@tass.telkomuniversity.ac.id

**Abstract** - Falling is an accident that known as a major health risk to cause injuries. The risk for serious injuries is increase when a person remains unconscious or immobilized and cannot call for help after falling. Fall detection devices can help to limit the major health risk because of falling by reducing the time between the fall and the arrival of aid, increasing the likelihood of successful treatment. Fall detection system in this study aim to design a system that use image processing to detect a fall in three-stage, video acquisition, feature extraction and fall detection. Video acquisition is done using the fall detection dataset. In the second stage, image processing is used for feature extraction using Gaussian Mixture Model for object segmentation, ellipse model to determine the orientation, Motion History Image to calculate the motion coefficient and Support Vector Machine as the classification method. System performance analysis is carried out in several factors, namely the appropriate type of SVM, the position of the camera used and high accuracy. Based on these factors, it can be seen that the most suitable SVM type is the RBF Kernel and the camera position in the fall detection dataset (FDD) produces the highest accuracy which is 95%.

**Keywords** — *fall detection, image processing, accuracy*

**Abstrak** - Jatuh merupakan salah satu kejadian tidak disengaja yang dapat menyebabkan luka yang berbahaya. Risiko jatuh akan bertambah saat orang yang jatuh tidak bisa memanggil bantuan. Alat *fall detection* dapat membantu mengurangi waktu antara kejadian jatuh dengan kedatangan pertolongan medis sehingga dapat mencegah adanya risiko berat pada kejadian jatuh. Sistem *fall detection* pada penelitian ini dibuat dengan menggunakan metode pengolahan citra yang terdiri dari 3 tahap yaitu akuisisi video, ekstraksi ciri dan deteksi kejadian jatuh. Akuisisi video dilakukan dengan menggunakan dataset *fall detection*. Proses ekstraksi ciri dilakukan dengan menggunakan *Gaussian Mixture Model* untuk segmentasi objek, *ellipse model* untuk menentukan orientasi, *Motion History Image* (MHI) untuk menghitung *motion coefficient* dan *Support Vector Machine* (SVM) sebagai metode klasifikasi kejadian jatuh. Analisis kinerja sistem dilakukan dalam beberapa faktor yaitu tipe SVM yang sesuai, posisi kamera yang digunakan dan tingkat akurasi yang lebih besar. Berdasarkan faktor-faktor tersebut dapat diketahui bahwa tipe SVM yang paling sesuai adalah RBF Kernel dan posisi kamera pada *the fall detection dataset* (FDD) menghasilkan nilai terbaik dengan nilai akurasi kinerja sistem yang didapatkan adalah 95%.

**Kata Kunci** - *fall detection, pengolahan citra, akurasi*

## I. Pendahuluan

Jatuh merupakan salah satu kejadian tidak disengaja yang berbahaya, khususnya pada orang lanjut usia yang membutuhkan pertolongan medis secepatnya. Jatuh dapat menyebabkan luka yang berbahaya diantaranya adalah tulang patah, luka pada kulit, trauma pada kepala dan hilang kesadaran. Risiko jatuh akan bertambah saat orang yang jatuh tidak bisa memanggil bantuan. Orang lanjut usia yang tinggal sendiri dapat mengalami kesulitan untuk memanggil bantuan secepatnya. Setidaknya 28-35% orang lanjut usia jatuh sekali atau lebih tiap tahun [1]. Hampir setengah dari orang lanjut usia yang tidak sadar setelah jatuh selama lebih dari 1 jam tanpa mendapatkan pertolongan dapat menyebabkan kematian dalam rentang 6 bulan [2]. Skenario tersebut menjadi alasan mengapa banyak penelitian mengenai *fall detection* sebagai pencegah adanya risiko berat pada kejadian jatuh. *Fall detection* dapat membantu mengurangi waktu antara kejadian jatuh dengan kedatangan pertolongan medis.

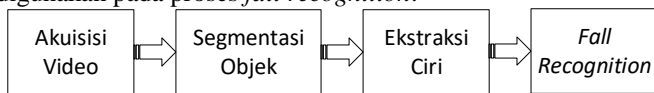
*Fall detection* dapat berupa sebuah alat yang dipakai di badan dengan menggunakan sensor untuk mendeteksi kejadian jatuh seperti *accelerometer* atau sensor *inertial measurement unit* (IMU) [3]. *Fall detection* dapat pula berupa alat yang berbasis video. Metode berbasis video pada sistem *fall detection* merupakan sistem sederhana yang menggunakan kamera sebagai alat bantu pendeteksi jatuh. Pengolahan video dengan metode *Gaussian Mixture Model*, *Elliptical Approximation* dan MHI dapat mendeteksi kejadian jatuh dengan akurasi mencapai 90% [4]. Pengolahan video dengan menggunakan metode *Gaussian Mixture Model*, *ellipse model* dan *Hidden Markov Model* mempunyai akurasi sebesar 97,47% [5].

Sistem *fall detection* pada penelitian ini terdiri dari 3 tahap yaitu akuisisi video, ekstraksi ciri dan deteksi kejadian jatuh. Pada tahap akuisisi video, kamera dipasang di salah satu ruangan rumah yang mempunyai persentase kejadian jatuh tinggi. Pada tahap ekstraksi ciri, *Gaussian Mixture Model* digunakan untuk memisahkan *background* dengan objek yang

akan dideteksi. Metode *blob* digunakan untuk mencari nilai tinggi dan lebar dari objek yang telah dipisahkan oleh *Gaussian Mixture Model*. *Ellipse model* digunakan untuk mencari nilai orientasi dari objek yang sedang dideteksi. *Motion History Image* (MHI) digunakan untuk menghitung *motion coefficient* dari objek yang bergerak. Klasifikasi *Support Vector Machine* (SVM) digunakan untuk mengklasifikasikan skenario kegiatan normal dan kejadian jatuh. Saat kejadian jatuh terjadi, sistem dapat mendeteksi kejadian jatuh dan membedakan kejadian jatuh dengan kejadian tidak jatuh. Pada penelitian ini, tipe SVM yang sesuai, posisi kamera yang digunakan dan tingkat akurasi dibahas untuk menentukan sistem *fall detection* yang memiliki kinerja terbaik.

**II. Metode Penelitian**

Gambar 1 merupakan diagram sistem mengenai proses pengolahan video. Video yang telah didapat akan mengalami 2 proses sebelum masuk ke algoritma *fall recognition*. Segmentasi objek digunakan untuk memisahkan antara objek yang akan di deteksi dari *background* sebuah citra. Sedangkan ekstraksi ciri digunakan untuk membuat database yang akan digunakan pada proses *fall recognition*.



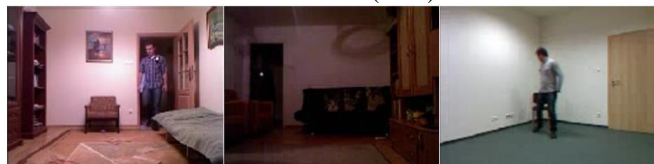
Gambar 1. Diagram Blok Sistem

**A. Akuisisi Video**

Akuisisi video dilakukan dengan menggunakan *dataset*. *Dataset* yang digunakan adalah *the fall detection dataset* (FDD) [6] dan *UR Fall Detection Dataset* [7]. *Dataset* dibagi menjadi dua tipe, yaitu *dataset* kejadian jatuh dan kejadian tidak jatuh. Selain itu *dataset* digunakan untuk dua tahap, yaitu tahap latihan dan tahap uji.



Gambar 2. The Fall Detection Dataset (FDD)



Gambar 3. UR Fall Detection Dataset

**B. Segmentasi Objek**

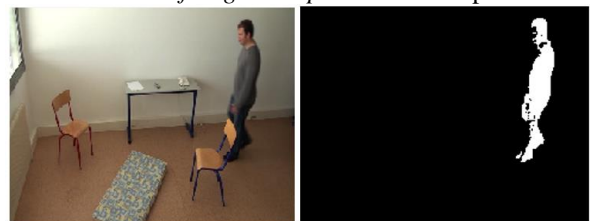
Pada tahap segmentasi objek, objek yang akan dideteksi yaitu objek yang sedang bergerak, dipisahkan dengan *background*. Proses ini dilakukan dengan menggunakan metode *Gaussian Mixture Model*. GMM secara garis besar bekerja dengan cara menangkap perubahan piksel dalam

interval waktu tertentu. GMM bertujuan untuk mendeteksi perubahan dari setiap piksel agar bisa mendeteksi objek bergerak yang disebut dengan *foreground* dan objek diam yang disebut *background*. Probabilitas distribusi piksel  $X_t$  dapat dimodelkan sebagai persamaan *mixture of K Gaussian* berikut [8]:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \eta(X_t; \mu_{i,t}, \Sigma_{i,t}) \dots\dots\dots (1)$$

Dimana  $K$  adalah nomor dari distribusi Gaussian,  $\omega_{i,t}$  adalah perkiraan berat dari distribusi Gaussian ke  $i$  saat waktu  $t$ ,  $\eta$  adalah fungsi kemungkinan kepadatan dari Gaussian ke  $i$ ,  $\mu_{i,t}$  adalah rata-rata dari nilai Gaussian ke  $i$  saat waktu  $t$ ,  $\Sigma_{i,t}$  adalah matriks kovariansi dari Gaussian ke  $i$  saat waktu  $t$ .

Piksel yang merupakan bagian dari objek bergerak akan termasuk dalam parameter *foreground pixel*. Hasil dari tahap ini akan berbentuk citra biner dengan *background pixel* berwarna hitam dan *foreground pixel* berwarna putih.



Gambar 4. Segmentasi Objek Menggunakan GMM

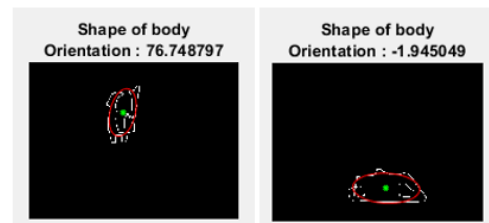
**C. Ekstraksi Ciri**

Ekstraksi ciri dilakukan pada objek yang telah di segmentasi untuk mendapatkan nilai *aspect ratio*, *orientation* dan *motion coefficient*. *Aspect ratio* merupakan nilai perbandingan antara lebar dan tinggi dari objek yang bergerak.

$$AspectRatio(n) = \frac{W(n)}{H(n)} \dots\dots\dots (2)$$

Dimana  $W(n)$  dan  $H(n)$  merupakan lebar dan tinggi dari objek bergerak yang sedang dideteksi pada saat *frame*  $n$ .

*Ellipse model* merupakan sebuah model yang dapat digunakan untuk menghitung orientasi. Orientasi menentukan sudut dari objek yang sedang dideteksi. Sudut ini merupakan sudut yang dibentuk oleh major axis dari ellipse model dan garis horizontal dari citra.



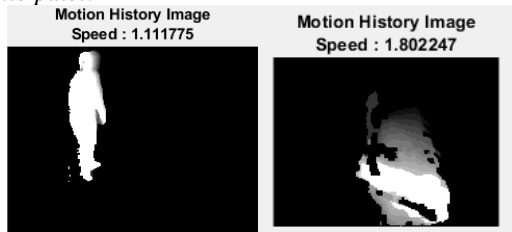
Gambar 5. Citra Orientasi

MHI menggambarkan informasi mengenai objek berdasarkan kecerahan citra dengan menghitung perubahan piksel pada lokasi yang sama pada periode waktu tertentu [9]. *Motion History Image* digunakan untuk menghitung *motion coefficient* dari objek yang bergerak. Saat kejadian jatuh terjadi, nilai dari *motion coefficient* akan bernilai besar. Pada citra MHI, pergerakan yang baru saja terjadi akan mempunyai

piksel yang lebih cerah intensitasnya dibanding dengan piksel yang lain.

$$C_{motion} = \frac{\sum_{Pixels(x,y) \in blob} H_T(x,y,t)}{\# pixels \in blob} \dots\dots\dots (3)$$

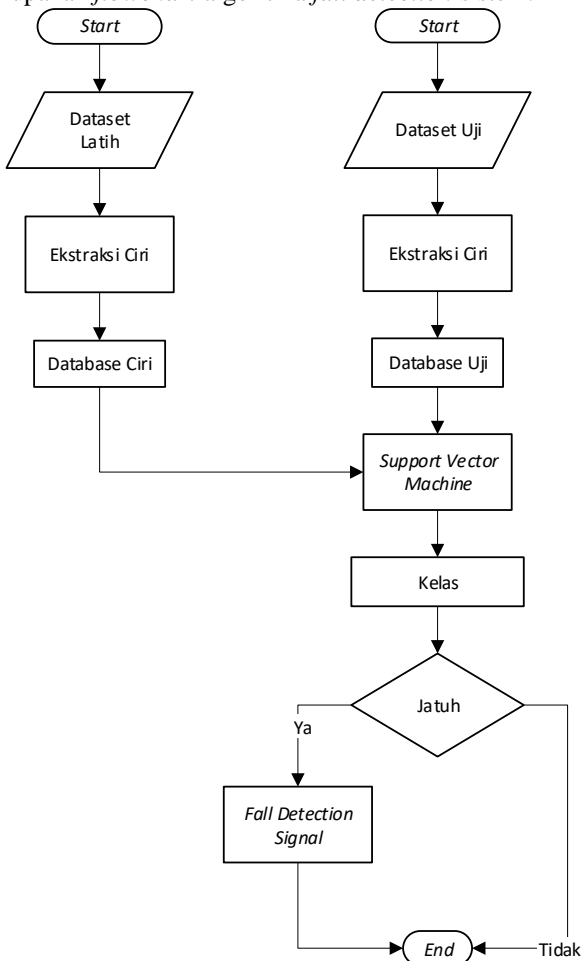
Dimana  $\sum_{Pixels(x,y) \in blob} H_T(x,y,t)$  adalah jumlah piksel yang merupakan bagian dari *gray pixel*,  $\# pixels \in blob$  adalah jumlah piksel yang merupakan penjumlahan bagian dari *gray pixel* dan *white pixel*.



Gambar 6. Citra MHI

**D. Fall Recognition**

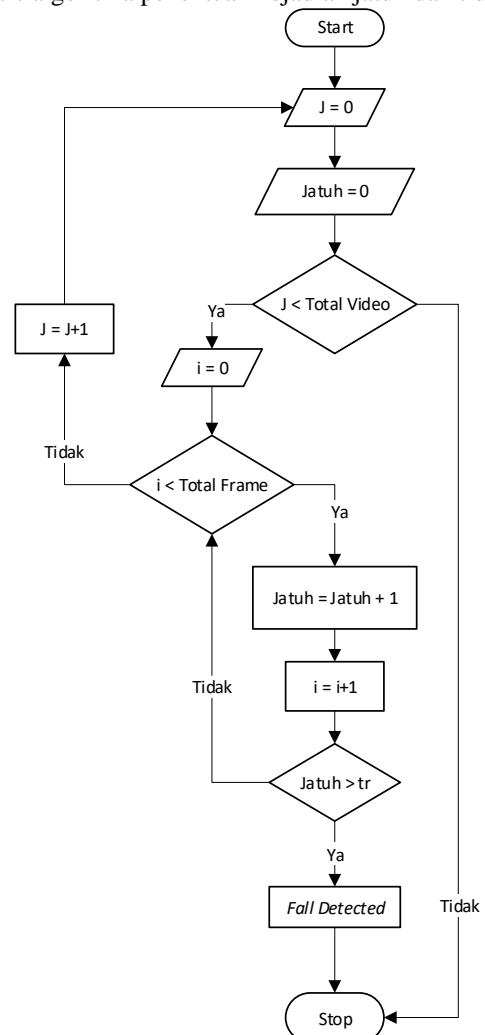
Variabel *motion coefficient*, *aspect ratio* dan *orientation* digunakan untuk pengambilan keputusan dalam sistem *fall detection* berdasarkan algoritma *fall detection*. Gambar 7 merupakan *flowchart* algoritma *fall detection* sistem.



Gambar 7. Flowchart Fall Detection

Seluruh ciri pada tahap pelatihan dan pengujian disimpan didalam sebuah database yang berisi *motion coefficient*, *aspect ratio*, *orientation* dan kelas. Kelas 1 merupakan kelas kejadian jatuh sedangkan kelas 2 merupakan kejadian tidak jatuh. Data ciri yang didapat dari proses *training* akan menghasilkan sebuah *hyperplane* yang dapat membedakan kelas kejadian jatuh dan kelas kejadian tidak jatuh. Dalam menentukan *hyperlane*, terdapat 2 tipe SVM yang berbeda yaitu linear dan nonlinear. Pada penelitian ini digunakan tipe linear dan salah satu tipe nonlinear yaitu RBF Kernel.

Ketika sistem telah menentukan kelas dari tiap *frame* data uji, sistem menentukan apakah video yang diproses merupakan kejadian jatuh atau tidak jatuh. Gambar 8 merupakan *flowchart* algoritma penentuan kejadian jatuh dan tidak jatuh.



Gambar 8. Flowchart Penentuan Kejadian Jatuh

Variabel *i* menunjukkan banyaknya *frame* tiap video, variabel *j* menunjukkan banyaknya video pada proses tahap uji sedangkan variabel *tr* merupakan *threshold*. *Threshold* menggambarkan durasi lamanya kejadian jatuh terjadi. Ketika pada satu video terdapat *frame* yang merupakan kelas jatuh sebanyak lebih dari *threshold* secara berturut-turut maka

sistem mengambil keputusan bahwa video yang diuji merupakan kejadian jatuh. Setelah proses berakhir, sistem akan melakukan perintah untuk memberikan indikasi bahwa kejadian jatuh terdeteksi.

**E. Skenario Penelitian**

Pengujian sistem *fall recognition* dilakukan dengan menghitung kinerja sistem dalam melakukan proses deteksi. Dalam menghitung kinerja sistem, satuan *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN) digunakan untuk mendeskripsikan hasil dari pengujian.

Tabel 1. Satuan Kinerja

		Kejadian Sebenarnya	
		Kelas Jatuh	Kelas Tidak Jatuh
Prediksi	Kelas Jatuh	<i>True Positive</i>	<i>False Positive</i>
	Kelas Tidak Jatuh	<i>False Negative</i>	<i>True Negative</i>

Paramater-parameter yang digunakan dalam menghitung kinerja sistem adalah *sensitivity*, *specificity*, *precision* dan *accuracy*. *Sensitivity* menghitung persentase banyaknya kejadian jatuh yang terdeteksi benar sebagai kejadian jatuh. *Specificity* menghitung persentase banyaknya kejadian tidak jatuh yang terdeteksi benar sebagai kejadian tidak jatuh. *Precision* menghitung persentase banyaknya kejadian jatuh yang diprediksi dengan benar. *Accuracy* menghitung persentase seberapa besar sistem dapat mendeteksi kejadian dengan benar.

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots (4)$$

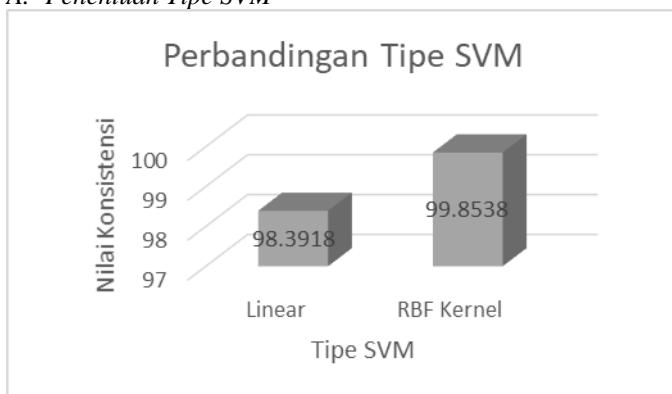
$$Specificity = \frac{TN}{TN+FP} \times 100\% \dots\dots\dots (5)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots (6)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \dots\dots\dots (7)$$

**III. Hasil dan Pembahasan**

**A. Penentuan Tipe SVM**



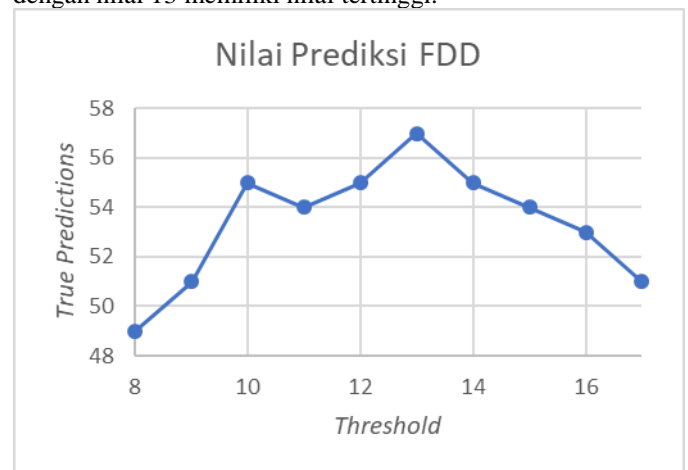
Gambar 9. Perbandingan Tipe SVM

Untuk menentukan tipe SVM yang digunakan, yaitu tipe linear dan nonlinear, dilakukan pengujian untuk membandingkan kedua tipe sehingga dapat dipilih tipe yang memiliki nilai konsistensi paling tinggi. Nilai konsistensi tipe merupakan nilai perbandingan kelas yang diperoleh dari satu pengujian dengan pengujian sebelumnya.

Pada sistem yang menggunakan tipe linear, konsistensi hasil kelas yang diperoleh bernilai 98.3918 sedangkan pada sistem tipe RBF Kernel nilai konsistensi hasil kelas yang diperoleh bernilai 99.8538. Sehingga pada pengujian selanjutnya tipe SVM yang digunakan adalah RBF Kernel

**B. Pengujian Dataset FDD**

Pengujian ini dilakukan dengan menggunakan dataset uji berjumlah 60 video yang terdiri dari 36 kejadian jatuh dan 24 kejadian tidak jatuh. Dalam memilih nilai *threshold*, dilakukan pengujian sebanyak 10 kali dengan nilai *threshold* yang berbeda. Untuk menentukan *threshold* terbaik yang dapat digunakan maka dapat dilihat dari penjumlahan antara kasus *true positive* dan *true negative*. Nilai penjumlahan ini merupakan nilai berapa banyak kelas yang di prediksi oleh sistem sesuai dengan kelas aslinya. Gambar 10 menggambarkan grafik nilai prediksi sistem. *Threshold* dengan nilai 13 memiliki nilai tertinggi.

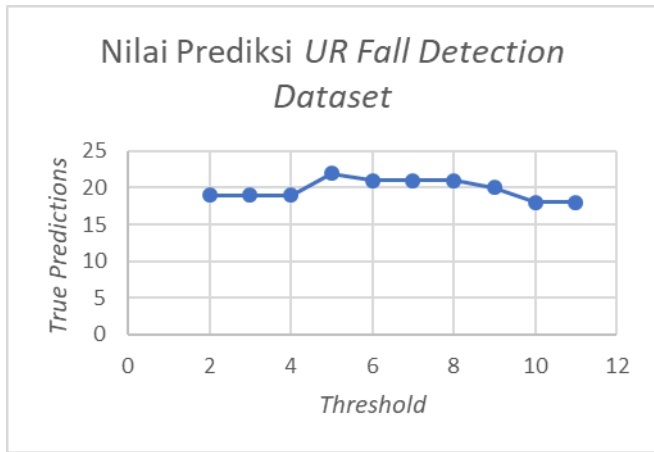


Gambar 10. Nilai Prediksi FDD

**C. Pengujian UR Fall Detection Dataset**

Pengujian ini dilakukan dengan menggunakan dataset uji berjumlah 25 video yang terdiri dari 10 kejadian jatuh dan 15 kejadian tidak jatuh. Dalam memilih nilai *threshold*, dilakukan pengujian sebanyak 10 kali dengan nilai *threshold* yang berbeda.

Untuk menentukan *threshold* terbaik yang dapat digunakan maka dapat dilihat dari penjumlahan antara kasus *true positive* dan *true negative*. Nilai penjumlahan ini merupakan nilai berapa banyak kelas yang di prediksi oleh sistem sesuai dengan kelas aslinya. Gambar 11 menggambarkan grafik nilai prediksi sistem. *Threshold* dengan nilai 5 memiliki nilai tertinggi.



Gambar 11. Nilai Prediksi UR Fall Detection Dataset

**D. Perhitungan Kinerja**

Pengujian sistem dilakukan dengan menggunakan 2 dataset. Dataset yang digunakan adalah *the fall detection dataset* (FDD) [4] dan *UR Fall Detection Dataset* [5]. Pada pengujian dataset FDD, kamera diposisikan menyerupai penempatan kamera CCTV sedangkan pada pengujian dataset *UR Fall Detection Dataset* kamera diposisikan menyerupai penempatan kamera rekam tv. Tabel 2 menunjukkan perbandingan nilai kinerja dari pengujian dataset FDD dan pengujian *UR Fall Detection Dataset*.

Tabel 2. Tabel Perbandingan Kinerja

	FDD (%)	UR Fall Detection Dataset (%)
Sensitivity	97.222	80
Specificity	91.667	93.333
Precision	94.594	88.889
Accuracy	95	88

Pada pengujian dataset FDD didapatkan nilai akurasi sebesar 95% sedangkan pada pengujian *UR Fall Detection Dataset* didapatkan nilai akurasi sebesar 88%. Pada pengujian dataset FDD, posisi kamera menghasilkan lebih banyak skenario jatuh dengan orientasi yang berbeda. Hal ini menyebabkan sistem dapat membedakan kejadian jatuh dan kejadian non jatuh dengan lebih baik.

**IV. Kesimpulan**

Berdasarkan hasil analisis dan pengujian maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan pengujian yang dilakukan dalam menentukan tipe SVM, sistem tipe RBF Kernel merupakan tipe terbaik karena memiliki nilai konsistensi hasil prediksi kelas lebih besar yaitu 99.8538.
2. Berdasarkan pengujian yang dilakukan menggunakan 2 dataset yang berbeda untuk menentukan nilai threshold, pengujian menggunakan dataset FDD memiliki tingkat keberhasilan tertinggi dengan nilai threshold yaitu 13. Pada pengujian menggunakan *UR Fall Detection Dataset*

memiliki tingkat keberhasilan tertinggi dengan nilai threshold yaitu 5.

3. Pada pengujian menggunakan dataset FDD nilai akurasi kinerja sistem yang didapatkan adalah 95% sedangkan pada pengujian menggunakan *UR Fall Detection Dataset* nilai akurasi kinerja sistem yang didapatkan adalah 88%.
4. Pada pengujian menggunakan dataset FDD, posisi kamera yang diposisikan menyerupai penempatan kamera CCTV menghasilkan lebih banyak skenario jatuh dengan orientasi yang berbeda. Hal ini menyebabkan sistem dapat membedakan kejadian jatuh dan kejadian non jatuh dengan lebih baik.

**V. Daftar Pustaka**

- [1] WHO, "Good Health Adds Life to Years.Global brief for World Health Day 2012," *World Heal. Organ.*, 2012.
- [2] S. R. Lord, C. Sherrington, H. B. Menz, and J. C. T. Close, *Falls in older people: Risk factors and strategies for prevention, second edition*. 2007.
- [3] M. Firmansyah, "RANCANG BANGUN SISTEM FALL DETECTION UNTUK ORANG LANJUT USIA BERBASIS INERTIAL MEASUREMENT UNIT," *TEKTRIKA - J. Penelit. dan Pengemb. Telekomun. Kendali, Komputer, Elektr. dan Elektron.*, 2019.
- [4] M. Krekovic *et al.*, "A method for real-time detection of human fall from video," *MIPRO, 2012 Proc. 35th Int. Conv.*, 2012.
- [5] H. L. U. Thuc, P. Van Tuan, and J. N. Hwang, "An effective video-based model for fall monitoring of the elderly," in *Proceedings - 2017 International Conference on System Science and Engineering, ICSSE 2017*, 2017.
- [6] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification," *J. Electron. Imaging*, 2013.
- [7] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, 2014.
- [8] K. Tra and T. V. Pham, "Human fall detection based on adaptive background mixture model and HMM," in *International Conference on Advanced Technologies for Communications*, 2013.
- [9] Q. Feng, C. Gao, L. Wang, M. Zhang, L. Du, and S. Qin, "Fall detection based on motion history image and histogram of oriented gradient feature," in *2017 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2017 - Proceedings*, 2018.

