

**DETEKSI JATUH MENGGUNAKAN ALGORITMA FEED-FORWARD ARTIFICAL
NEURAL NETWORK (FFANN)**

SKRIPSI



Oleh:

ANITA AMALIA HAK BISYU

H13114302

Pembimbing Utama : Dr. Eng. Armin Lawi, M.Eng

Pembimbing Pertama : Andi Galsan Mahie, S.Si., M.Si.

Penguji : 1. Dr. Diaraya, M.Ak

2. Andi Kresna Jaya, S.Si., M.Si

3. Drs. Muhammad Zakir, M.Si.

PROGRAM STUDI ILMU KOMPUTER

JURUSAN MATEMATIKA

KULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

DECEMBER 2018



Optimization Software:
www.balesio.com

**DETEKSI JATUH MENGGUNAKAN ALGORITMA FEED-FORWARD ARTIFICIAL
NEURAL NETWORK (FFANN)**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Ilmu Komputer Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin Makassar

ANITA AMALIA HAK BISYU

H13114302

**PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

DECEMBER 2018



LEMBAR PERYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

DETEKSI JATUH MENGGUNAKAN ALGORITMA FEED-FORWARD ARTIFICAL NEURAL NETWORK (FFANN)

adalah benar hasil karya saya sendiri bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun

Makassar, 19 Desember 2018

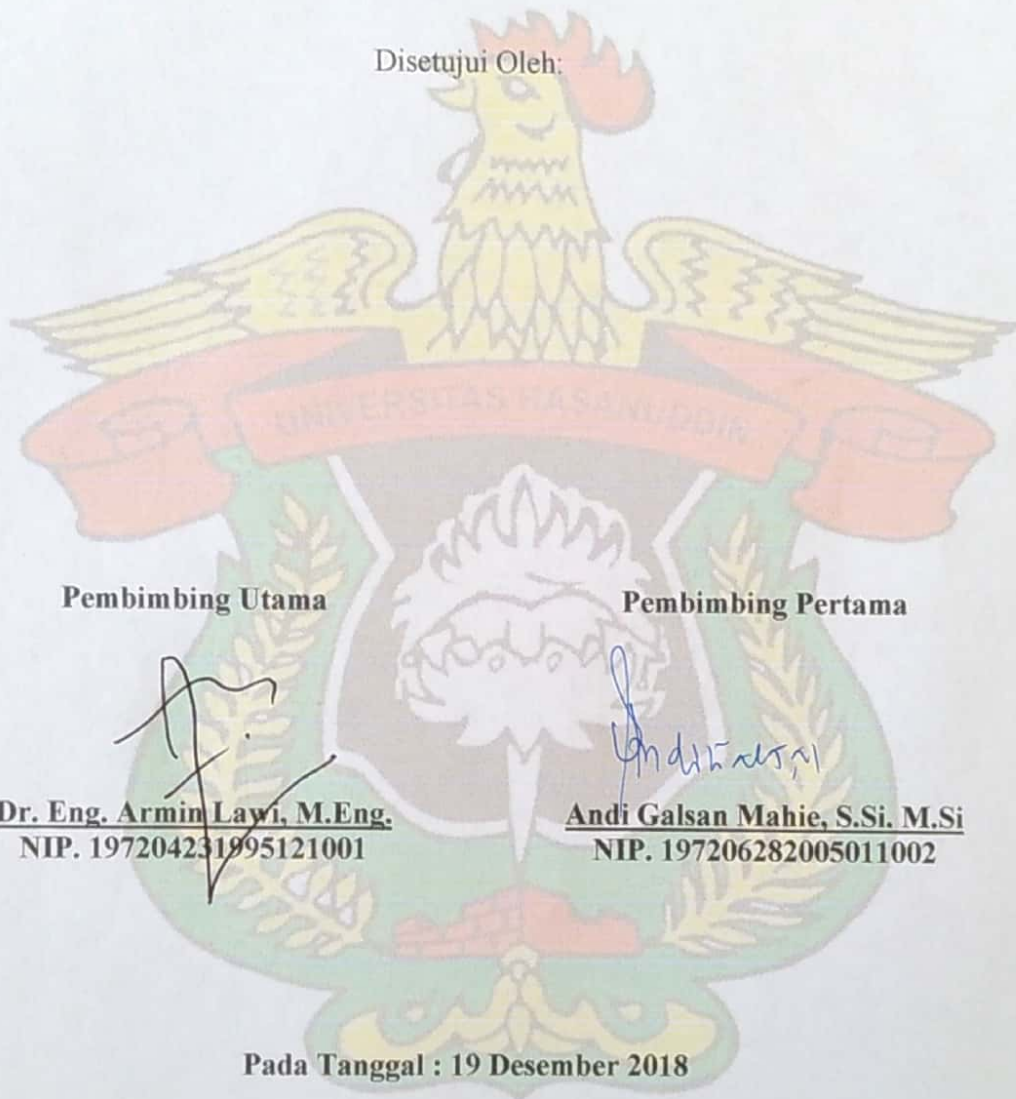


Anita Amalia Hak Bisyu

NIM. H13114302

DETEKSI JATUH MENGGUNAKAN ALGORITMA FEED-FORWARD ARTIFICIAL
NEURAL NETWORK (FFANN)

Disetujui Oleh:



Pembimbing Utama

Pembimbing Pertama


Dr. Eng. Armin Lawi, M.Eng.
NIP. 197204231995121001


Andi Galsan Mahie, S.Si. M.Si
NIP. 197206282005011002

Pada Tanggal : 19 Desember 2018



Optimization Software:
www.balesio.com

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Anita Amalia Hak Bisyu

NIM : H13114302

Program Studi : Ilmu Komputer

Judul Skripsi : Deteksi Jatuh menggunakan Algoritma Feed-Forward Artificial Neural Network (FFANN)

Telah berhasil dipertahankan dihadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

1. Ketua : Dr. Diaraya, M.Ak
2. Sekretaris : Andi Kresna Jaya, S.Si., M.Si.
3. Anggota : Drs. Muhammad Zakir, M.Si.
4. Anggota : Dr. Eng. Armin Lawi, M.Eng
5. Anggota : Andi Galsan Mahie, S.Si., M.Si.

Tanda Tangan

(.....)
(.....)
(.....)
(.....)
(.....)

Ditetapkan di : Makassar

Tanggal : 19 Desember 2018

v



KATA PENGANTAR

Segala puji penulis panjatkan kehadirat Allah SWT, yang senantiasa melimpahkan rahmat dan karunia-Nya. Shalawat dan salam senantiasa penulis kirimkan kepada Baginda Rasulullah SAW, yang telah mengajarkan kebenaran dan membimbing umat – umatnya ke arah yang benar. Rasa syukur yang tak terkira atas segala nikmat yang telah diberikan terutama nikmat kesehatan, kesempatan dan kemudahan yang dikaruniakan kepada penulis dalam menyelesaikan tugas akhir yang berjudul **Deteksi Jatuh menggunakan Algoritma Feed-Forward Artificial Neural Network**.

Penyusunan tugas akhir ini tentunya tidak lepas dari bantuan berbagai pihak baik moril maupun materil. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang tulus dan penghargaan yang tak terhingga untuk ayahanda **Ir. H. Abdul Munir Hak Bisyu, M.Sc** dan ibunda **Ir. Hj. Yusni Ermita Saleh, MM** atas segala dukungan, doa, restu, nasehat dan motivasi yang tak henti-hentinya mereka berikan kepada penulis untuk menggapai cita-cita. Juga ucapan terima kasih untuk adinda **Andi Amirul Hak Bisyu** dan **Andi Ashraf Hak Bisyu** yang selalu mendukung penulis dalam penulisan skripsi ini, serta untuk seluruh keluarga besar, dan terkhusus untuk almarhum nenek tercinta **Hj. Aisyah A. Jeppe**, terima kasih atas doanya.

Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada :

1. Ibu **Rektor Universitas Hasanuddin** beserta jajarannya, Bapak **Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam beserta jajarannya**, dan seluruh pihak birokrasi atas pengetahuan dan kemudahan-kemudahan yang diberikan, baik dalam bidang akademik maupun bidang kemahasiswaan.
2. Bapak Prof. **Dr. Amir Kamal Amir, M.Sc.**, selaku Ketua Jurusan Matematika, dan Bapak **Dr. Amran, S.Si., M.Si.**, selaku Sekretaris Jurusan, serta Bapak **Dr. Diaraya, M.Ak.,**

a Program Studi Ilmu Komputer yang telah memberikan banyak bantuan is menjalani pendidikan. Terima kasih juga untuk segenap jajaran Pegawai rusan Matematika atas bantuannya dalam pengurusan akademik selama ini.



3. Bapak **Dr. Eng. Armin Lawi, M.Eng**, selaku pembimbing utama, atas segala ilmu, nasehat, dan kesabaran dalam membimbing penulis serta meluangkan waktu di sela-sela rutinitas yang begitu padat hingga skripsi ini dirampungkan, dan Bapak **Andi Galsan Mahie, S.Si., M.Si.**, selaku pembimbing pertama, untuk segala ilmu, nasehat, dan kesabaran dalam membimbing dan mengarahkan penulis, serta bersedia meluangkan waktunya untuk mendampingi penulis sejak awal penyusunan hingga akhir perampungan skripsi ini.
4. Bapak **Dr. Hendra, S.Si, M.Kom**, selaku ketua tim penguji sekaligus Penasehat Akademik (PA) penulis untuk segala ilmu, nasehat, saran dan motivasi yang diberikan kepada penulis mulai dari perkuliahan hingga penyusunan skripsi ini, dan Ibu **Naimah Aris, S.Si, M.Math**, selaku sekretaris penguji serta Bapak **Dr. Loeky Haryanto, MS. M.Sc.**, selaku anggota tim penguji yang telah bersedia meluangkan waktu untuk memberikan saran dan arahan kepada penulis dalam penyusunan skripsi ini.
5. Saudara-saudara **Keluarga Cemara (Nurul Hardiyanti, Nur Nilamyani, Yolanda Gabyriella Ferandji, Yaumil Agus Awal, dan Fuad Fadhil Azzar)** yang telah menemani penulis selama perkuliahan, yang telah meluangkan waktu dan berbagi suka-duka dan kebersamaan selama menuntut ilmu.
6. Bapak **Benny Enrico Panggabean, Ilham, dan Bima Satria Yudha Mohammad**, yang tak henti-hentinya selalu membantu dan menyemangati penulis mulai awal hingga akhir penyusunan skripsi ini.
7. Teman-teman seperjuangan **Ilmu Komputer 2014 (Nadya, Jo, Luki, Dila, Danti, Firda, Yuyu, Nuhi, Aspar, Syam, Tio, Hajar, Darul, dll)** yang membantu dan memberi support penulis dalam penyusunan skripsi ini.
8. Adik-Adik **Ilmu Komputer 2015, 2016, 2017, dan 2018 (Fika, Fachrul, Arka, Rian, dll)**
9. Rekan-rekan **KKN UNHAS Gelombang 96 Kec. Pattalassang Kab. Takalar (Juju, Venny, Nabil, Firman, dan Andri)** yang telah menjadi keluarga baru selama KKN dan menjadikan KKN sebagai momen yang membahagiakan.
10. Semua pihak yang telah banyak berpartisipasi, baik secara langsung maupun tidak langsung dalam penyusunan skripsi ini yang tak sempat penulis sebutkan satu per satu.



Semoga segala bantuan yang dengan tulus ditujukan kepada penulis mendapatkan balasan dari Allah SWT. Mudah-mudahan tulisan ini memberikan manfaat kepada semua pihak yang membutuhkan dan terutama untuk penulis.

Makassar, 19 Desember 2018

Anita Amalia Hak Bisyu



**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIK**

Sebagai civitas akademik Universitas Hasanuddin saya yang bertanda tangan di bawah ini:

Nama : Anita Amalia Hak Bisyu
NIM : H 131 14 302
Program Studi : Ilmu Komputer
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Non-eksklusif (*Non-exclusive Royalty- Free Right*)** atas tugas akhir saya yang berjudul:

“Deteksi Jatuh menggunakan Feed-Forward Artificial Neural Network (FFANN)”

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal, 18 Desember 2018

Yang menyatakan,



c Bisyu

ABSTRAK

Kejatuhan dapat menyebabkan kondisi yang fatal, bahkan kematian. Sistem monitor jatuh diperlukan untuk mengurangi tingkat kematian orang jatuh dalam kelompok potensi tinggi jatuh. Deteksi peristiwa jatuhnya dapat dideteksi menggunakan sensor ponsel yang ada. Data sensor akselerometer dan gyroscope akan diambil untuk mengenali gerakan kemudian dibandingkan dengan data sensor jatuh untuk mengidentifikasi jatuh.

Dalam makalah ini, kami menggunakan deteksi berdasarkan ambang batas dan umpan maju jaringan saraf untuk membedakan peristiwa jatuh antara kegiatan sehari-hari. Hasil penelitian menunjukkan bahwa sistem yang diusulkan dapat berhasil memperoleh akurasi sistem pendeteksi jatuh hingga 95,00% dari 80 percobaan jatuh, dan akurasi rata-rata 98,33% dari ADL 120 kali dari total percobaan.



ABSTRACT

Fall can cause fatal conditions, even death. A fall monitor system is necessary to reduce the rate of fall fatalities of people in group of high potential of fall. Fall event detection can be detected using existing mobile phone sensor. Accelerometer and gyroscope sensor data will retrieved to recognize the motion then compared to the falling sensor data to identify fall.

In this paper, we use threshold based-fall detection and feed-forward neural network to distinguish the falling event between daily activities. The result showed that the proposed system could successfully obtained accuracy of the fall detection system till of 95.00% of the 80 trials fall, and an average accuracy of 98.33% of the ADL 120 times the total experiment.



DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori.....	6
A. Sensor Accelerometer dan Gyroscope pada Smartphone.....	6
B. Algoritma Fall-Detection.....	8
C. Jaringan Saraf Tiruan (Artificial Neural Network).....	11
D. Feed-forward Artificial Neural Network (FFANN).....	16
E. Artificial Neural Network Backpropagation.....	17
F. Metode Fixed-Sized Overlapping Windows (FOSW).....	24
G. Ukuran Kinerja Sensitivitas, Spesifisitas, dan Akurasi.....	25
2.2 Kerangka Konseptual.....	27
BAB III METODE PENELITIAN.....	28
3.1 Tahapan Penelitian.....	28
A. Aplikasi deteksi jatuh menggunakan FFANN.....	29
B. Perancangan Database.....	29
C. Penentuan Nilai Threshold.....	29
D. Pengujian Deteksi jatuh.....	29
E. Perancangan Model Deteksi Jatuh.....	31
F. Pemilihan Lokasi Penelitian.....	31
G. Perancangan Sistem.....	31



3.4	Sumber Data.....	32
3.5	Instrumen Penelitian	32
BAB IV HASIL DAN PEMBAHASAN		33
4.1	Pelabelan Kejadian Jatuh dan Tidak Jatuh menggunakan Sensor Accelerometer dan Gyroscope.....	33
A.	Sensor Accelerometer dan Gyroscope.....	33
B.	Pemrosesan Signal Sensor.....	33
C.	Segmentasi Data.....	34
D.	Ekstraksi Fitur.....	34
E.	Proses labeling menggunakan <i>Threshold-based Algorithm</i>	36
4.2	Model Deteksi Jatuh menggunakan Feed Forward Neural Network.....	38
A.	Ekstraksi Fitur Neural Network.....	39
B.	Normalisasi.....	39
C.	Penyusunan Arsitektur Neural Network.....	40
D.	Model Feed-forward Artificial Neural Network.....	42
E.	Deteksi Jatuh menggunakan model neural network.....	49
4.3	Analisis Kinerja Model Deteksi Jatuh FFANN	50
A.	Kinerja Model Deteksi Jatuh FFANN dengan Data Training 70% dan Data Testing 30%	51
B.	Kinerja Model Deteksi Jatuh FFANN dengan Data Training 70% dan Data Testing 30%.....	53
BAB V KESIMPULAN DAN SARAN		56
5.1	Kesimpulan	56
5.2	Saran.....	57
DAFTAR PUSTAKA		58





Optimization Software:
www.balesio.com

DAFTAR GAMBAR

Gambar 1 Sistem Kordinat	6
Gambar 2 Sistem Kordinasi Smartphone	7
Gambar 3 Gyroscope	7
Gambar 4 Sistem kordinat gyroscope pada Smartphone vertikal.....	8
Gambar 5 Flowchart Algoritma Fall-Detection.....	11
Gambar 6 Prinsip jaringan saraf tiruan.....	13
Gambar 7 Jaringan dengan banyak lapisan (multilayer net)	14
Gambar 8 Model FFNN dengan Backpropagation Training	19
Gambar 9 Flowchart Tahapan Penelitian	28
Gambar 10 Arsitektur Fall Detection System.....	31
Gambar 11 Grafik Accelerometer dan gyroscope untuk kejadian jatuh.....	38
Gambar 12 Grafik Accelerometer dan Gyroscope untuk ADL	38
Gambar 13 Model FFANN.....	42
Gambar 14 Grafik Performa MSE pada proses training dan validation pada proses training untuk data training 70% dan data testing 30%	51
Gambar 15 Grafik Gradien, jumlah error validasi, dan learning rate pada proses training untuk data training 70% dan data testing 30%	52
Gambar 16 Grafik Performa MSE pada proses training dan validation pada proses training untuk data training 80% dan data testing 20%	53
Gambar 17 Grafik Gradien, jumlah error validasi, dan learning rate pada proses training untuk data training 70% dan data testing 30%	54



DAFTAR TABEL

Tabel 1 Confusion Matrix.....	25
Tabel 2 Skenario Simulasi Deteksi Jatuh	30
Tabel 3 Daftar nilai threshold	36
Tabel 4 Karakteristik subjek penelitian	37
Tabel 5 Jumlah Aktivitas yang disimulasikan	37
Tabel 6 Fitur yang digunakan pada Neural Network.....	39
Tabel 7 Dataset Input FFNN.....	39
Tabel 8 Nilai MSE pada proses training dan validation Neural Network	40
Tabel 9 Nilai Bobot dan Bias pada Lapisan Input untuk $\lambda = 0.01$ untuk Data Training 70%.....	43
Tabel 10 Nilai Bobot dan Bias pada lapisan tersembunyi untuk $\lambda = 0.01$ untuk Data Training 70%.....	43
Tabel 11 Nilai Bobot dan Bias pada lapisan input untuk $\lambda = 0.005$ untuk Data Training 70%	44
Tabel 12 Nilai Bobot dan Bias pada Lapisan Tersembunyi untuk $\lambda = 0.005$ untuk Data Training 70%.....	44
Tabel 13 Nilai Bobot dan Bias pada lapisan input untuk $\lambda = 0.001$ untuk Data Training 70%	44
Tabel 14 Nilai Bobot dan Bias pada Lapisan Tersembunyi untuk $\lambda = 0.001$ untuk Data Training 70%.....	45
Tabel 15 Nilai epoch, MSE dan gradien model neural network.....	45
Tabel 16 Nilai Bobot dan Bias pada Lapisan Input untuk $\lambda = 0.01$ untuk Data Training 80%.....	46
Tabel 17 Nilai Bobot dan Bias pada lapisan tersembunyi untuk $\lambda = 0.01$ untuk Data Training 80%	47
Tabel 18 Nilai Bobot dan Bias pada lapisan input untuk $\lambda = 0.005$ untuk Data Training 80%	47
Tabel 19 Nilai Bobot dan Bias pada Lapisan Tersembunyi untuk $\lambda = 0.005$ untuk Data Training 80%	47
Tabel 20 Nilai Bobot dan Bias pada lapisan input untuk $\lambda = 0.001$ untuk Data Training 80%	48
Tabel 21 Nilai Bobot dan Bias pada Lapisan Tersembunyi untuk $\lambda = 0.001$ untuk Data Training 80%	48
Tabel 22 Nilai epoch, MSE dan gradien model neural network.....	49
Tabel 23 Confussion Matriks pada proses testing untuk data testing 30%	52
Tabel 24 Nilai Sensitivitas, Spesifisitas, dan Akurasi pada proses training, validation and testing untuk data training 70%(90% training, 10% validation) dan data testing 30%.....	52
Tabel 25 Confussion Matriks pada proses testing untuk data testing 20%	54
Tabel 26 Nilai Sensitivitas, Spesifisitas, dan Akurasi pada proses training, validation and testing untuk data training 80%(90% training, 10% validation) dan data testing 20%.....	54



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada umumnya, manusia tentu tidak dapat terhindar dari keterbatasan secara fisik, seperti keterbatasan penglihatan, kemampuan berjalan, dan pendengaran. Dengan kondisi tersebut, potensi manusia mengalami peristiwa jatuh cukup tinggi. Menurut Data Rumah Sakit Dr.Wahidin Sudirohusodo, Pada tahun 2017 mulai dari bulan januari sampai bulan september, Korban insiden jatuh berusia golongan geriatri (usia lanjut 60 tahun ke atas) ada 6 orang dimana 3 diantaranya tidak cedera dan 3 lainnya mengalami cedera. Pada usia seperti ini tingkat risiko jatuh lebih tinggi khususnya pasien geriatri yang menderita beberapa penyakit dan masalah biopsikososial.

Peristiwa jatuh dapat terjadi di berbagai tempat, sehingga ada berbagai hal yang bisa memperburuk keadaan daripada jatuh. Beberapa akibat yang ditimbulkan oleh jatuh, seperti rasa sakit, kelemahan tubuh, cacat, bahkan dapat meningkatkan risiko kematian yang bergantung daripada berbagai faktor seperti tempat terjadinya jatuh. Minimumnya kualitas penanganan korban jatuh ini salah satunya terjadi karena durasi yang lama antara waktu jatuh dan waktu penanganan. Hal ini disebabkan berbagai faktor, antara lain adalah kurangnya pengawasan sehingga informasi kejadian jatuh yang tidak menyebar oleh pihak terdekat sehingga memakan waktu yang lama untuk pasien ditangani tim medis dan memperburuk kondisi pasien jatuh.

Pengawasan sangat diperlukan bagi manusia yang termasuk golongan orang yang berpotensi jatuh tinggi terutama lanjut usia. Pengawasan dapat dilakukan oleh keluarga atau orang terdekat. Orang yang bertugas mengawasi tentunya untuk selalu berada di dekat lanjut usia agar lebih mudah mengetahui dan memberikan pertolongan ketika jatuh terjadi. Akan tetapi, sebagai anggota keluarga



atau kerabat dekat tidak mungkin selamanya berada di dekat pengguna tersebut untuk melakukan pengawasan secara langsung. Oleh karena itu, pengawasan dapat dilakukan secara tidak langsung dengan memanfaatkan teknologi yang ada saat ini.

Dengan kemajuan teknologi saat ini yang sangat pesat terjadi terutama pada teknologi smartphone android yang banyak di gunakan, salah satunya adalah teknologi sensor, antara lain accelometer dan gyroscope. Data-data yang ada dari smartphone ini kemudian digunakan untuk mendeteksi berbagai aktivitas yang dilakukan pengguna. Melalui data-data yang ada ini, kita bisa mengontrol hidup mereka.

Sistem kemudian akan mengidentifikasi kejadian jatuh berdasarkan data-data yang ada melalui perbandingan data sensor ketika orang jatuh ini dengan data yang direkam pada saat kejadian jatuh.

Penelitian mengenai jatuh ini berkaitan dengan penelitian-penelitian lain mengenai pengenalan aktivitas. (Kwapisz, Weiss, & Moore, 2010) menggunakan WISDM (*Wireless Sensor Data Mining*) platform untuk mengambil data sensor dari 29 pengguna yang berjalan, *jogging*, menaiki tangga, menuruni tangga, duduk, dan berdiri yang dibagi menjadi segmen per-10 detik dan menghasilkan fitur berdasarkan 200 pembacaan data mentah accelerometer dari *smartphone* yang terkandung dalam setiap segmen per-10 detik. Sedangkan (Cherbumroong, Cang, Atkins, & Yu, 2013) menggunakan alat pengenalan aktivitas yang dipakai di pergelangan tangan yang jauh lebih praktis dan biaya terjangkau hingga menjadi sistem *assisted living* yang praktis. Adapun penelitian lain, (Benaissa, Koppen, & Yoshida, 2017) yang tidak menggunakan koleksi data yang berkomunikasi secara wireless. Caranya adalah menggabungkan pengenalan aktivitas menggunakan sensor badan dengan melacak posisi pengguna dengan sensor accelerometer di ruangan 3 dimensi untuk mengetahui

pengguna dan pengenalan emosi melalui deteksi detak jantung dan nafas
ografis untuk memonitor emosi dari pengguna dengan memfokuskan objek
khusus untuk usia lanjut. Banyak penelitian lainnya deteksi jatuh yang juga
suskan objek penelitian untuk usia lanjut.



Selain menggunakan accelerometer, mendeteksi gerak jatuh bisa dengan pendekatan orientasi tubuh menggunakan gyroscope. (Zhang, Li, McCullagh, Nugent, & Zheng, 2013) mendeteksi jatuh secara real-time menggunakan sensor dengan cara mendeteksi gerak transisi antar postur untuk mengetahui apakah gerak jatuh itu normal seperti meloncat, merebahkan diri atau abnormal dengan mengetahui orientasi dari sensor gyroscope.

Metode perbandingan ambang batas (*threshold*) juga digunakan dalam berbagai penelitian deteksi jatuh. (Rakhman, Nugroho, Widyawan, & Kurnianingsih, 2014) menggunakan algoritma berdasarkan *threshold*. Data dari dua sensor tersebut lalu diekstrak menghasilkan nilai absolut dari pengurangan nilai maksimum dengan minimum, nilai accelerometer, dan sudut jatuh antara pengguna dengan vector gravitasi akan dibandingkan dengan *threshold* yang sudah ditentukan lalu jika terdeteksi jatuh maka akan mengaktifkan sistem alarm dan peringatan. (Arifin, Wibisono, & Pratomo, 2013) juga melakukan penelitian yang sama, kemudian pada penelitian keduanya (Wibisono, Arifin, Pratomo, Ahmad, & Ijtihadie, 2013) ia menemukan bahwa setelah nilai delta kedua sensor dan sudut jatuh sudah memenuhi syarat ambang batas, nilai minimum dari sensor accelerometer juga dibandingkan dengan *threshold* untuk mengetahui aktivitas sebelumnya disengaja atau tidak. Jika jatuh terdeteksi maka dikirimkan peringatan bersama titik koordinat lokasi pengguna. Untuk meningkatkan akurasi deteksi geraknya, (Li, Stankovic, Hanson, Barth, & Zhou, 2009) bahkan menggunakan dua node sensor TEMPO (Technology-Enabled Medical Precision Observation) 3.0 yang mencakup accelerometer tri-aksial dan gyroscope tri-aksial yang diletakkan pada bagian dada dan kaki.

Untuk meningkatkan akurasi dan mengurangi jumlah *false alarm*, algoritma berdasarkan *threshold* ini bisa digabungkan dengan model Feed-forward Network (FFNN). (Sengto & Leauhatong, 2012) melakukan penelitian akan Backpropagation Neural Network (BPNN) setelah membedakan gerak lambat dari jatuh dengan aktivitas gerak tiba-tiba lainnya dengan menggunakan sensor accelerometer tri-aksial yang dipakai pada pinggang. (Yodpijit



& Sittiwanchai, 2017) menggunakan FFANN (Feed-forward Artificial Neural Network) bersama dengan algoritma *backpropagation* menghasilkan nilai akurasi, sensitivitas, dan spesifitas yang lebih tinggi dengan menggunakan Fallz 1.0 yang mencakup sensor accelerometer dan gyroscope yang digunakan pada pinggang pengguna.

Pada penelitian ini, algoritma berdasarkan *threshold* juga digabungkan dengan model FFANN dengan algoritma *backpropagation* yang menggunakan sensor accelerometer dan gyroscope sebagai sensor gerak untuk deteksi gerak tubuh dan pergerakan untuk mendeteksi keadaan jatuh pada *smartphone*.

Struktur dari skripsi ini disusun sebagai berikut. Bab I membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penulisan, serta organisasi skripsi. Bab II membahas mengenai landasan teori, konsep dasar yang mendasari pokok permasalahan dalam tulisan ini. Serta penelitian terkait. Bab III berisi waktu dan tempat penelitian, tahapan penelitian, rancangan sistem, sumber data, dan instrumen penelitian.

1.2 Rumusan Masalah

1. Bagaimana menggunakan sensor accelerometer dan gyroscope pada smartphone untuk memberi label kejadian jatuh dan tidak jatuh?
2. Bagaimana mendeteksi jatuh menggunakan model Feed-Forward Artificial Neural Network?
3. Bagaimana menganalisis kinerja model deteksi jatuh FFANN?

1.3 Batasan Masalah



Keadaan jatuh yang dimaksud adalah jatuh pada lantai, simulasi menggunakan matras.

Data sensor dari aplikasi hanya berupa dari accelerometer dan gyroscope.

1.4 Tujuan Penelitian

Dengan memperhatikan latar belakang dan rumusan masalah diatas maka tujuan dari penelitian ini adalah

1. Menggunakan sensor accelerometer dan gyroscope pada smartphone untuk deteksi jatuh
2. Mendeteksi jatuh menggunakan Feed-Forward Artificial Neural Network
3. Menguji akurasi, sensitifitas, dan spesifisitas aplikasi dalam mengklasifikasi gerak jatuh

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, diharapkan berguna untuk mendeteksi jatuh lebih akurat sehingga dapat mendapatkan penanganannya lebih cepat.

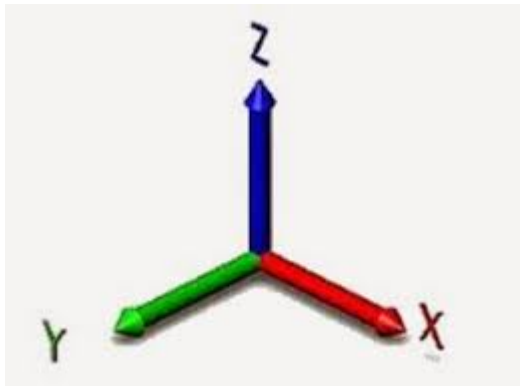


BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

A. Sensor Accelerometer dan Gyroscope pada Smartphone



Gambar 1 Sistem Kordinat

Sensor accelerometer mengukur akselerasi atau percepatan bahwa perangkat mengalami perubahan yang relatif sesuai dengan tiga sumbu x , y , dan z . Pada umumnya sistem menggunakan data ini untuk mengetahui apakah smartphone dalam orientasi berdiri (portrait) atau memanjang (landscape) (Gunawan, 2017).

Sistem koordinat smartphone didasarkan pada layar dan orientasi default smartphone. Sumbu x , y , dan z adalah seperti ditunjukkan pada Gambar 1 dan bekerja sebagai berikut:

1. **Sumbu x**

Horisontal, dengan nilai positif ke kanan dan nilai negatif ke kiri

2. **Sumbu y**

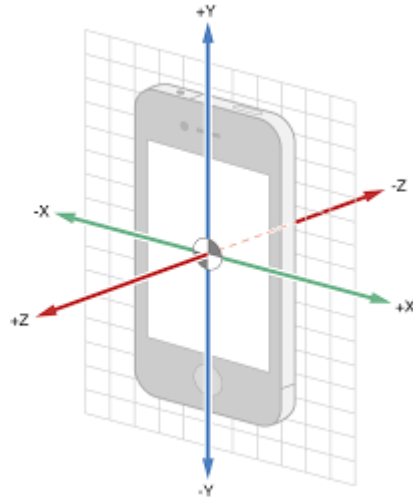
Vertikal, dengan nilai positif ke atas dan nilai negatif ke bawah

Sumbu z

Nilai positif keluar dari layar ke arah depan dan nilai negatif dibelakang layar (titik nol nol terletak di layar).



Saat pengguna memindahkan ponsel, sumbu mengikuti gerakan telepon dan tidak bertukar tempat.



Gambar 2 Sistem Kordinasi Smartphone

Orientasi smartphone (atau posisi relatif terhadap magnet utara) dapat dipantau dengan menggunakan sensor orientasi, yang menyediakan sudut rotasi 3D di sepanjang tiga sumbu (pitch, roll, yaw), dilambangkan dengan $(\omega_x, \omega_y, \omega_z)$, seperti yang ditunjukkan pada Gambar 3. Adapun sumbunya sebagai berikut (Zhang, Li, McCullagh, Nugent, & Zheng, 2013):



Gambar 3 Gyroscope



1. Pitch (ω_x)

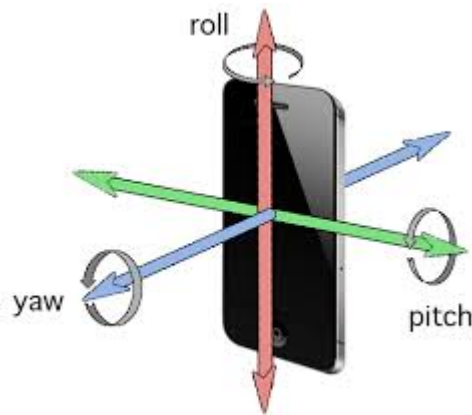
Pitch mengukur derajat putaran di sekitar sumbu X; kisaran nilai adalah -180° sampai 180° , dengan nilai positif ketika sumbu z positif bergerak ke arah sumbu Y positif. Ini sekitar 0° saat sumbu X vertikal; Ini adalah sekitar $\pm 90^\circ$ ketika sumbu Y vertikal; itu sekitar $\pm 180^\circ$ saat bagian atas layar mengarah ke tanah.

2. Roll (ω_y)

Roll mengukur derajat rotasi di sekitar sumbu Y, $-90^\circ \leq \omega_y \leq 90^\circ$, dengan nilai positif saat zaxis positif bergerak menuju sumbu X positif. Ini sekitar 0° saat sumbu Y vertikal; Ini adalah sekitar $\pm 90^\circ$ ketika Xaxis vertikal.

3. Yaw (ω_z)

Yaw mengukur derajat rotasi di sekitar sumbu Z, $0^\circ \leq \omega_z \leq 360^\circ$. Ini digunakan untuk mendeteksi arah kompas. Seperti $\omega_z = 0^\circ$ atau 360° , Utara; $\omega_z = 180^\circ$, Selatan; $\omega_z = 90^\circ$, Timur; $\omega_z = 270^\circ$, Barat.



Gambar 4 Sistem kordinat gyroscope pada Smartphone vertikal

B. Algoritma Fall-Detection

Algoritma fall detection merupakan algoritma pendeteksi aktivitas fisik user saat terjatuh. Terdapat beberapa langkah untuk mendeteksi aktivitas fisik pada saat terjatuh. Adapun langkah-langkahnya yaitu penghitungan nilai α dari sensor accelerometer, penghitungan nilai skalar omega (ω)



dari sensor gyroscope, penghitungan nilai skalar theta (θ) dari sumbu x sensor accelerometer, dan pencarian nilai threshold dari nilai omega (ω). Setelah nilai alpha, omega, dan theta didapatkan maka akan dilakukan pengecekan. Pengecekan yang dilakukan meliputi pengecekan nilai alpha realtime dan nilai omega realtime terhadap nilai training yang telah ditentukan, pengecekan nilai alpha terhadap threshold, pengecekan nilai omega terhadap threshold, serta pengecekan nilai theta terhadap nilai training yang telah ditentukan (Arifin, Wibisono, & Pratomo, 2013).

1. Penghitungan nilai sensor

Adapun rumus dalam pencarian nilai α , ω , dan θ adalah seperti berikut (Arifin, Wibisono, & Pratomo, 2013):

a. Penghitungan nilai alpha (α)

Penghitungan nilai alpha (α) didapat dari kuadrat dari penjumlahan sumbu x, y, dan z dari sensor accelerometer seperti (1).

$$\alpha_t = \sqrt{\alpha_{t_x}^2 + \alpha_{t_y}^2 + \alpha_{t_z}^2} \quad (1)$$

$\alpha_{A_x}^2$ = kuadrat nilai alpha pada sumbu x

$\alpha_{A_y}^2$ = kuadrat nilai alpha pada sumbu y

$\alpha_{A_z}^2$ = kuadrat nilai alpha pada sumbu z

Nilai alpha (α) digunakan pada algoritma fall detection untuk mendeteksi aktivitas statis, mencari nilai threshold yang kemudian akan dibandingkan antara data realtime dengan data training.

b. Penghitungan nilai omega (ω)

Penghitungan nilai omega (ω) didapat dari kuadrat dari penjumlahan sumbu roll, pitch, dan yaw dari sensor gyroscope seperti (2).

$$\omega_t = \sqrt{\omega_{t_x}^2 + \omega_{t_y}^2 + \omega_{t_z}^2} \quad (2)$$

$\omega_{A_x}^2$ = kuadrat nilai omega pada sumbu roll



$\omega_{A_y}^2$ = kuadrat nilai omega pada sumbu pitch

$\omega_{A_z}^2$ = kuadrat nilai omega pada sumbu azimuth

Nilai omega (ω) digunakan pada algoritma fall detection untuk mendeteksi aktivitas statis, mencari nilai threshold yang kemudian akan dibandingkan antara data realtime dengan data training.

c. Penghitungan nilai theta (θ)

Penghitungan nilai theta (θ) didapat dari arc cosinus dari pembagian sumbu x sensor accelerometer terhadap pusat gravitasi g sebesar 9,8 m/s² seperti (3).

$$\theta_t = \frac{\cos^{-1} \alpha_{t_x}}{g} \quad (3)$$

α_{A_x} = nilai *alpha* pada sumbu x

g = konstanta gravitasi bumi (9,8 m/s²)

Nilai *theta* (θ) ini merupakan nilai sudut antara posisi tubuh dengan vektor gravitasi.

2. Penghitungan nilai rentang sensor

$$\Delta_\alpha = \max\{\alpha_t \dots \alpha_{t-n}\} - \min\{\alpha_t \dots \alpha_{t-n}\} \quad (4)$$

$$\Delta_\omega = \max\{\omega_t \dots \omega_{t-n}\} - \min\{\omega_t \dots \omega_{t-n}\} \quad (5)$$

Nilai delta atau rentang atau jangkauan dari sensor bertujuan untuk membedakan gerak statis dan dinamis. Nilai yang didapatkan kemudian masing-masing dibandingkan dengan nilai t_α dan t_ω .

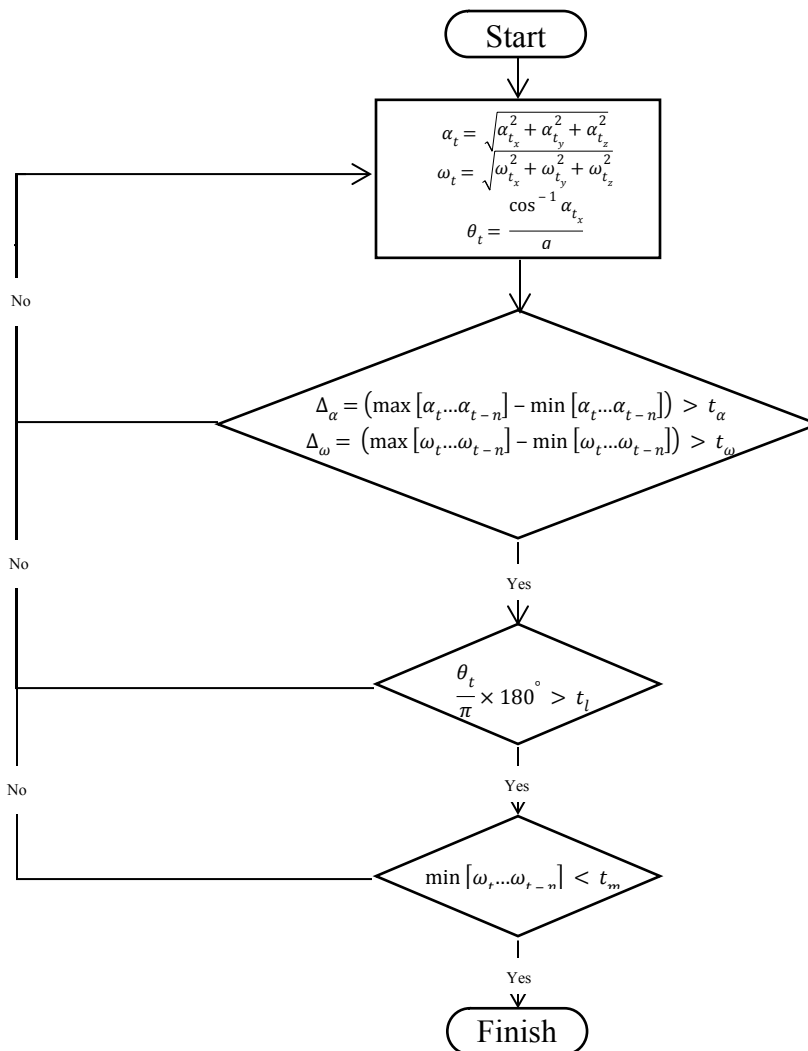
3. Perbandingan nilai theta

Nilai sudut pada theta digunakan pada algoritma *fall detection* untuk mendeteksi postur *user*, apakah postur *user* termasuk dalam kondisi terbaring atau tidak. Nilai yang didapatkan ini kemudian dibandingkan dengan threshold t_θ .



4. Penghitungan nilai minimal sensor accelerometer pada sumbu Y

Nilai minimal ini digunakan untuk mengetahui apakah aktivitas jatuh terjadi secara sengaja atau tidak sengaja (secara tiba-tiba). Nilai ini juga dibandingkan (Wibisono, Arifin, Pratomo, Ahmad, & Ijtihadie, 2013).



Gambar 5 Flowchart Algoritma Fall-Detection



C. Jaringan Saraf Tiruan (Artificial Neural Network)

Jaringan Saraf Tiruan (JST) adalah prosesor yang terdistribusi besar-besaran secara parallel yang dibuat dari unit proses sederhana, yang mempunyai kemampuan untuk menyimpan pengetahuan berupa pengalaman dan dapat digunakan untuk proses lain (Haykin, 2009).

Jaringan saraf tiruan tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, ke dalam jaringan saraf tiruan dimasukkan pola-pola masukan (dan keluaran) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima (Puspitaningrum, 2006).

Jaringan saraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan saraf biologis manusia, dengan asumsi bahwa (Riadi, 2016):

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung.
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- Untuk menentukan keluaran, Setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlah masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang.

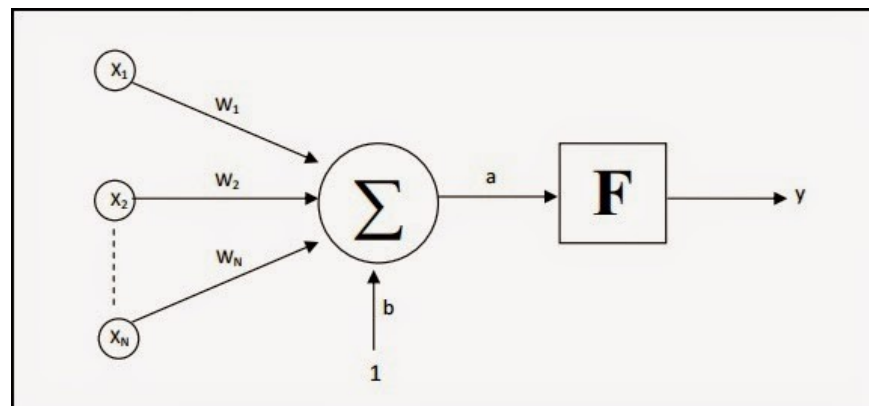
1. Prinsip Jaringan Syaraf Tiruan

Prinsip jaringan saraf tiruan (JST) ditentukan oleh tiga elemen dasar model jaringan saraf, yaitu:

- 1) Satu set dari sinapsis, atau penghubung yang masing-masing digolongkan oleh bobot atau kekuatannya.



- 2.) Sebuah penambah untuk menjumlahkan sinyal-sinyal input. Ditimbang dari kekuatan sinaptik masing-masing neuron.
 - 3.) Sebuah fungsi aktivasi untuk membatasi amplitudo output dari neuron.
- Fungsi ini bertujuan membatasi jarak amplitudo yang diperbolehkan oleh sinyal output menjadi sebuah angka yang terbatas. Prinsip jaringan saraf tiruan secara sederhana digambarkan di bawah ini (Haykin, 2009):



Gambar 6 Prinsip jaringan saraf tiruan

Pada gambar diatas, Y menerima masukan dari neuron x_1 , x_2 , dan x_3 , dengan bobot hubungan masing-masing adalah w_1 , w_2 , dan w_3 . Ketiga impuls neuron yang ada dijumlahkan menjadi (Siang, 2004):

$$Net = x_1w_1 + x_2w_2 + x_3w_3 \quad (6)$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y=f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot (Siang, 2004).



arsitektur Jaringan Saraf Tiruan

ada jaringan saraf tiruan, neuron-neuron akan dikumpulkan dalam sebuah lapisan yang disebut dengan lapisan neuron (neuron layers). Neuron-neuron

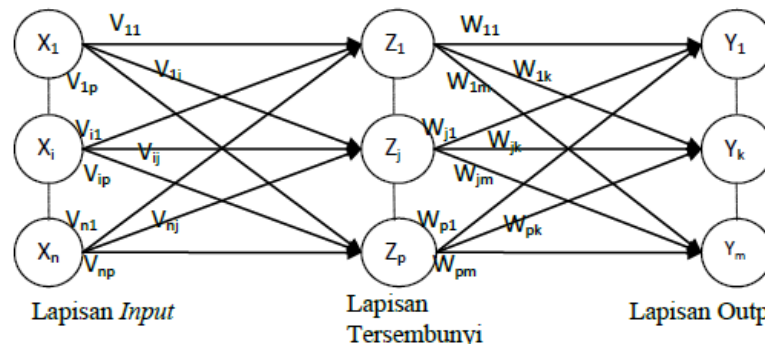
pada satu lapisan akan dihubungkan dengan lapisan-lapisan lainnya. Informasi yang didapatkan pada sebuah neuron akan disampaikan ke semua lapisan-lapisan yang ada, mulai dari lapisan masukan sampai dengan lapisan keluaran melalui lapisan tersembunyi (hidden layer). Pada jaringan saraf tiruan ini tiga lapisan bukanlah sebuah struktur umum karena beberapa jaringan saraf ada yang tidak memiliki lapisan tersembunyi. Menurut (Haykin, 2009), secara umum ada tiga jenis arsitektur dari Jaringan Saraf Tiruan yaitu:

a. Jaringan dengan lapisan tunggal (single layer net)

Di dalam Jaringan Saraf Tiruan dengan satu layer, neuron-neuron diorganisasi dalam bentuk layer-layer. Dalam bentuk paling sederhana dari Jaringan Saraf Tiruan dengan satu layer, kita mempunyai sebuah input layer dari node sumber di mana informasi diproyeksikan ke output layer dari neuron tapi tidak bisa sebaliknya. Dengan kata lain, jaringan ini adalah tipe feed forward. Input layer dari node sumber tidak dihitung karena tidak ada perhitungan yang dilakukan. Jaringan dengan banyak lapisan (multilayer net).

b. Jaringan ini merupakan jaringan dengan satu atau lebih lapisan tersembunyi (hidden layer). Jaringan multi lapis ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan jaringan lapis tunggal, namun pelatihannya mungkin lebih rumit. Pada beberapa kasus, pelatihan pada jaringan ini lebih baik karena memungkinkan bagi jaringan untuk memecahkan masalah yang tidak dapat diselesaikan jaringan berlapis tunggal karena jaringan tidak bisa dilatih untuk menampilkan secara benar (Haykin, 2009).





Gambar 7 Jaringan dengan banyak lapisan (multilayer net)

3. Mode Pelatihan

Untuk membentuk jaringan neural network, hal yang penting lainnya yaitu memperhatikan penentuan nilai bobot pada proses training atau pelatihan. Metode pelatihan pada Neural network dibagi menjadi dua jenis, yaitu (Faussett, 1994):

a. Pelatihan Terawasi (*Supervised Learning*)

Pelatihan ini dilakukan dengan adanya urutan vektor pelatihan, atau pola yang masing-masing terkait dengan vektor target output. Bobot kemudian disesuaikan untuk algoritma pembelajaran. Proses ini dikenal sebagai pelatihan terawasi.

b. Pelatihan tak Terawasi (*Unsupervised Learning*)

Pada pelatihan ini jaring saraf mengatur segala kinerja dirinya sendiri, mulai dari masukan vektor hingga menggunakan data training untuk melakukan pembelajaran

4. Fungsi Aktivasi

Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan atau tidak. Ada beberapa fungsi aktivasi yang sering digunakan dalam Neural Network, antara lain (Siang, 2004):

Fungsi Linier (Identitas)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya.



Fungsi ini dirumuskan sebagai:

$$y = x \quad (7)$$

b. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode Backpropagation. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. Fungsi sigmoid biner dirumuskan sebagai:

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

dengan: $f'(x) = f(x)[1 - f(x)] \quad (9)$

D. Feed-forward Artificial Neural Network (FFANN)

Feed-Forward Artificial Neural Network adalah jaringan saraf tiruan dimana koneksi antara simpul tidak membentuk siklus. Dengan demikian, itu berbeda dari jaringan saraf berulang.

Jaringan saraf feedforward adalah tipe jaringan syaraf tiruan yang pertama dan paling sederhana yang dibuat. Dalam jaringan ini, informasi bergerak hanya satu arah, maju, dari node input, melalui node tersembunyi (jika ada) dan ke node output. Tidak ada siklus atau loop dalam jaringan.

FFANN merupakan salah satu model neural network yang sering digunakan. Penggunaan jaringan ini sering juga digunakan bersamaan dengan Algoritma Backpropagation.

Setiap neuron di lapisan tertentu terhubung dengan semua neuron di lapisan berikutnya. Hubungan antara neuron X_i dan ke Y_j ditandai oleh koefisien berat w_{ij} dan neuron dengan oleh koefisien threshold b (Gbr. 2). Koefisien berat



mencerminkan tingkat kepentingan koneksi yang diberikan dalam jaringan saraf. Nilai output (aktivitas) dari neuron Y_j ditentukan oleh Persamaan (10) dan (11). Itu menyatakan bahwa (Svozil, Kvasnicka, & Pospichal, 1997):

$$Y_j = f(y_j) \quad (10)$$

$$y_j = b + \sum_{i=1}^n w_{ij} x_i \quad (11)$$

di mana Y_{in_j} adalah potensi dari neuron Y_j dan fungsi $f(Y_{in_j})$ adalah fungsi transfer yang disebut (penjumlahan dalam Persamaan (11) dilakukan atas semua neuron j yang mentransfer sinyal ke neuron Y_j). Koefisien ambang dapat dipahami sebagai koefisien berat dari koneksi dengan neuron j yang ditambahkan secara formal, dimana $X_i = 1$ (disebut bias) (Svozil, Kvasnicka, & Pospichal, 1997).

Untuk fungsi transfer itu sebagai berikut (Svozil, Kvasnicka, & Pospichal, 1997):

$$f(y_j) = \frac{1}{1 + e^{-y_j}} \quad (12)$$

Proses adaptasi yang diawasi bervariasi koefisien ambang f_i dan koefisien berat badan untuk meminimalkan jumlah perbedaan kuadrat antara nilai output yang dihitung dan dibutuhkan. Ini dicapai dengan meminimalkan tujuan fungsi E (Svozil, Kvasnicka, & Pospichal, 1997):

$$E = \frac{1}{n} \sum_{j=1}^p Y_j - \hat{Y}_j \quad (13)$$

di mana Y_j dan \hat{Y}_j , adalah vektor yang terdiri dari aktivitas yang dihitung dan diperlukan dari neuron output dan penjumlahan berjalan di atas semua neuron output (Svozil, Kvasnicka, & Pospichal, 1997).



Artificial Neural Network Backpropagation

Pengertian Backpropagation

Backpropagation adalah algoritma pembelajaran untuk memperkecil tingkat error dengan cara menyesuaikan bobotnya berdasarkan perbedaan output dan target yang diinginkan. Backpropagation juga merupakan sebuah metode sistematis untuk pelatihan multilayer JST. Backpropagation dikatakan sebagai algoritma pelatihan multilayer karena Backpropagation memiliki tiga layer dalam proses pelatihannya, yaitu input layer, hidden layer dan output layer, dimana backpropagation ini merupakan perkembangan dari single layer network (Jaringan Layer Tunggal) yang memiliki dua layer, yaitu input layer dan output layer. Dengan adanya hidden layer pada backpropagation dapat menyebabkan besarnya tingkat error pada backpropagation lebih kecil dibanding tingkat error pada single layer network. Hal tersebut dikarenakan hidden layer pada backpropagation berfungsi sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target output yang diinginkan.

2. Arsitektur Backpropagation

Arsitektur algoritma backpropagation terdiri dari tiga layer, yaitu input layer, hidden layer dan output layer. Pada input layer tidak terjadi proses komputasi, namun pada input layer terjadi pengiriman sinyal input X ke hidden layer. Pada hidden dan output layer terjadi proses komputasi terhadap bobot dan bias dan dihitung pula besarnya output dari hidden dan output layer tersebut berdasarkan fungsi aktivasi tertentu. Dalam algoritma backpropagation ini digunakan fungsi aktivasi sigmoid biner, karena output yang diharapkan bernilai antara 0 sampai 1.

Tiga layer backpropagation adalah input layer, hidden layer dan output layer. Pada input layer, inputan divariabelkan dengan X_n . Pada hidden layer, terdapat bobot (V_{ij}) dan bias (V_{0j}), serta Z sebagai data hidden layer. Pada output layer terdapat bobot (W_{ij}) dan bias (W_{0j}) dengan data output divariabelkan dengan Y .

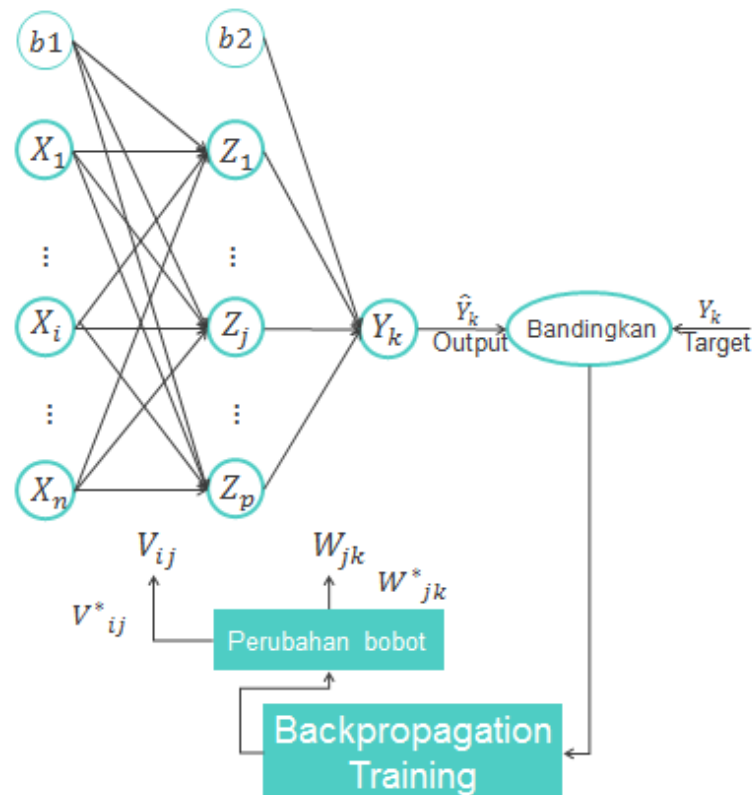


3. Algoritma Backpropagation

Algoritma backpropagation adalah sebuah algoritma untuk memperkecil tingkat error dengan menyesuaikan bobot berdasarkan perbedaan output dan target yang diinginkan.

Secara umum algoritmanya terdiri dari tiga langkah utama, yaitu :

- 1) Pengambilan input
- 2) Penelusuran error
- 3) Penyesuaian bobot



Gambar 8 Model FFNN dengan Backpropagation Training



Pada pengambilan input, terlebih dahulu dilakukan inisialisasi bobot, kemudian masuk ke dalam algoritma proses backpropagation yang terdiri dari komputasi maju yang bertujuan untuk menelusuri besarnya error dan

komputasi balik untuk mengupdate dan menyesuaikan bobot. Dalam mengupdate bobot dapat dilakukan dengan dua cara, yaitu tanpa momentum dan dengan momentum. Namun, yang dijelaskan di bawah ini dalam mengupdate bobotnya dilakukan tanpa memperhatikan besarnya momentum. Dengan demikian dalam metode backpropagation, algoritma yang harus dilakukan adalah inisialisasi bobot, komputasi feed forward dan backpropagation dan inisialisasi kondisi stopping berdasarkan nilai batas error atau jumlah batas epoch. Epoch merupakan rangkaian langkah dalam pembelajaran ANN. Satu epoch diartikan sebagai satu kali pembelajaran ANN.

Algoritma lengkap sebuah pembelajaran ANN adalah sebagai berikut :

Langkah 0 :

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- Tetapkan : Maksimum Epoch, Target Error, dan Learning Rate ().
- Inisialisasi : Epoch = 0.

Langkah 1 :

Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE > Target Error):

$$Epoch = Epoch + 1 \quad (14)$$

Langkah 2 :

Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan langkah-langkah berikutnya.



Pada dasarnya proses algoritma backpropagation terdiri dari 2 jenis komputasi, yaitu komputasi maju (feed forward) dan komputasi balik

(backpropagation).

Komputasi maju digunakan sebagai algoritma untuk menghitung nilai aktivasi yang ada pada semua neuron baik yang ada di lapis tersembunyi atau hidden layer maupun lapis keluaran atau output layer. Berikut ini merupakan algoritma yang digunakan dalam komputasi maju.

- a. Neuron yang berada pada lapisan tersembunyi melakukan perhitungan nilai total masukan dari lapisan tersembunyi dengan cara menjumlahkan perkalian sinyal masukan dengan bobot antara lapisan masukan (input layer) dan lapisan tersembunyi (hidden layer) dan nilai bias.
- b. Neuron tersebut melakukan perhitungan nilai aktivasi. Sinyal keluaran dari lapisan tersembunyi berguna sebagai lapisan masukan pada lapisan keluaran. Kemudian neuron yang berada pada lapisan keluaran melakukan perhitungan untuk menghitung nilai keluaran.
- c. Neuron keluaran tersebut menghitung nilai aktivasi dengan menerapkan fungsi aktivasi pada lapisan keluarannya.

Algoritma proses feed forwardnya adalah :

Langkah 3 :

Tiap-tiap unit input ($X_i, i=1,2,\dots,n$) menerima sinyal X_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

Langkah 4 :

- Tiap-tiap unit pada suatu lapisan tersembunyi ($Z_j, j=1,2,\dots,p$) menjumlahkan sinyal-sinyal input terbobot :

$$z_j = b1_j + \sum_{i=1}^n X_i V_{ij} \quad (15)$$



- gunakan fungsi aktivasi untuk menghitung sinyal output :

$$Z_j = f(z_j) \quad (16)$$

$$f(z_j) = \frac{1}{1 + e^{-z_j}} \quad (17)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Langkah 5 :

- Tiap-tiap unit output Y_k ($k=1,2,\dots,m$) menjumlahkan sinyal-sinyal input terbobot:

$$y_k = b2_k + \sum_{i=1}^p Z_i W_{jk} \quad (18)$$

- gunakan fungsi aktivasi untuk menghitung sinyal output :

$$Y_k = f(y_k) \quad (19)$$

$$f(y_k) = \frac{1}{1 + e^{-y_k}} \quad (20)$$

- kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Backpropagation merupakan sebuah algoritma yang berfungsi untuk melakukan perhitungan balik dari neuron keluaran agar memiliki nilai bobot yang sesuai dalam jaringan neural network. Dengan komputasi balik ini nilai error atau kesalahan bisa dikurangi dengan cukup baik. Berikut ini adalah gambaran kasar dari algoritma komputasi balik yang digunakan pada Artificial Neural Network:



- i. Sinyal keluaran yang dihasilkan pada komputasi maju kemudian dicocokkan dan dilakukan perhitungan untuk menghitung selisih antara target dengan sinyal keluaran yang ada pada neuron keluaran.
- ii. Hasil perhitungan ini kemudian digunakan untuk menyesuaikan bobot hubungan antara lapisan keluaran dengan semua neuron yang berada pada lapisan tersembunyi.
- iii. Setelah itu kirimkan sinyal kesalahan ke dalam lapisan tersembunyi sehingga setiap neuron yang berada pada lapisan tersembunyi bisa menyesuaikan beban yang ada agar nilai keluarannya mempunyai nilai yang mendekati dengan target.

Algoritma proses backpropagationnya adalah sebagai berikut :

Langkah 6 :

- Tiap-tiap unit output Y_k ($k=1,2,\dots,m$) menerima target pola yang berhubungan dengan pola input pelatihan, hitung informasi errornya :

$$\delta = (t_k - Y_k) f'(Y_{in_k}) \quad (21)$$

$$\varphi_{jk} = \delta_k Z_j \quad (22)$$

$$\beta_k = \delta_k \quad (23)$$

- Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta W_{jk} = \alpha \varphi_{jk} \quad (24)$$

- Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta b_k = \alpha \beta_k \quad (25)$$

Langkah 7 :

- Tiap-tiap unit tersembunyi Z_j ($j=1,2,\dots,p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya :



$$\delta_{in_j} = \sum_{k=1}^m \delta_{2_k} W_{jk} \quad (26)$$

- Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error :

$$\delta 1_j = \delta_{in_j} f'(Z_{in_j}) \quad (27)$$

$$\varphi 1_{ij} = \delta 1_j X_j \quad (28)$$

$$\beta 1_j = \delta 1_j \quad (29)$$

- Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta V_{ij} = \alpha \varphi 1_{ij} \quad (30)$$

- Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta b 1_j = \alpha \beta 1_j \quad (31)$$

Langkah 8 :

- Tiap-tiap unit output Y_k ($k=1,2,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$) :

$$W_{jk}^* = W_{jk} + \Delta W_{jk} \quad (32)$$

$$b_{2_k}^* = b_{2_k} + \Delta b_{2_k} \quad (33)$$

- Tiap-tiap unit tersembunyi Z_j ($j=1,2,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,\dots,n$) :

$$V_{ij}^* = V_{ij} + \Delta V_{ij} \quad (34)$$

$$b_{1_j}^* = b_{1_j} + \Delta b_{1_j} \quad (35)$$

Langkah 9 :

Hitung nilai MSE yang menunjukkan performa dari jaringan.



F. Metode Fixed-Sized Overlapping Windows (FOSW)

Metode FOSW (*Fixed Overlapping Size Window*) berfungsi untuk memisahkan data Time-series ke dalam vektor inputan tanpa kehilangan informasi dari data untuk mempersiapkan input dari data sensor.

Untuk ukuran window l dengan 50% overlapping, polanya adalah: dengan

$$Pola = \begin{bmatrix} 1 & 2 & \dots & l \\ \frac{l}{2} + 1 & \frac{l}{2} + 2 & \dots & \frac{l}{2} + l \\ \vdots & \vdots & \ddots & \vdots \\ \frac{i-1}{2}l + 1 & \frac{i-2}{2}l + 2 & \dots & \frac{i-1}{2}l + l \end{bmatrix} \quad (36)$$

$$0 \leq i \leq \frac{2 \times \text{Banyak Data}}{l} - 1.$$

G. Ukuran Kinerja Sensitivitas, Spesifisitas, dan Akurasi

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. *Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Prasetyo, 2012).

Tabel 1 Confusion Matrix

		Data Aktual	
		Jatuh	Tidak Jatuh
Data Prediksi	Jatuh	TP	FN
	Tidak Jatuh	FP	TN



Uji coba akurasi deteksi dilakukan untuk mengetahui tingkat akurasi algoritma *fall detection* ditinjau dari beberapa aktivitas fisik *user* pada saat terjatuh. Uji coba ini menggunakan perhitungan *true positive*, *true negative*, *false positive*, dan *false negative* (Arifin, Wibisono, & Pratomo, 2013).

1.) *True Positive* (TP)

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

2.) *True Negative* (TN)

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

3.) *False Positive* (FP)

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

4.) *False Negative* (FN)

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

$$\text{Sensitivitas} = \frac{TP}{TP + FN} \quad (37)$$

$$\text{Spesifitas} = \frac{TN}{TN + FP} \quad (38)$$

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (39)$$



2.2 Kerangka Konseptual

Menurut Data Rumah Sakit Dr.Wahidin Sudirohusodo, Pada tahun 2017 mulai dari bulan januari sampai bulan september, Korban insiden jatuh berusia golongan geriatri (usia lanjut 60 tahun ke atas) ada 6 orang dimana 3 diantaranya tidak cedera dan 3 lainnya mengalami cedera. Tingkat risiko jatuh lebih tinggi terjadi khususnya pasien geriatri yang menderita beberapa penyakit dan masalah biopsikososial



Pengawasan pada korban yang memiliki resiko jatuh yang tinggi yang minim sehingga menyebabkan penanganan kejadian jatuh menjadi terhambat



Solusi yang ditawarkan atas permasalahan tersebut adalah dibuatnya suatu aplikasi yang mempunyai sensor deteksi kejadian jatuh dengan akurasi yang tinggi



Aplikasi ini nantinya akan mendeteksi terjadinya peristiwa jatuh



Diharapkan dengan adanya Aplikasi yang mendeteksi jatuh akan membantu untuk mendeteksi jatuh lebih akurat sehingga dapat mendapatkan penanganannya lebih cepat.



Gambar 9 Kerangka Konseptual

