



PROYEK AKHIR

PENDETEKSIAN ANOMALI DATA PADA HASIL PENGUKURAN SENSOR WIRELESS BODY AREA NETWORK

ANOMALOUS DATA DETECTION ON WIRELESS BODY AREA NETWORK MEASUREMENTS

Oleh :

Fajar Setiawan
NRP. 2110155005

Dosen Pembimbing :

M. Udin Harun Al Rasyid, Ph.D.
NIP. 198108082005011001

Isbat Uzzin Nadhori, S.Kom., M.T.
NIP. 197405052003121002

PROGRAM STUDI D4 LJ TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2017



PROYEK AKHIR

**PENDETEKSIAN ANOMALI DATA PADA
HASIL PENGUKURAN SENSOR WIRELESS
BODY AREA NETWORK**

***ANOMALOUS DATA DETECTION ON WIRELESS BODY
AREA NETWORKS MEASUREMENTS***

Oleh :

Fajar Setiawan
NRP. 2110155005

Dosen Pembimbing :

M. Udin Harun Al Rasvid, Ph.D.
NIP. 198108082005011001

Isbat Uzzin Nadhori, S.Kom., M.T.
NIP. 197405052003121002

**PROGRAM STUDI D4 LJ TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2017**

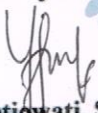
**PENDETEKSIAN ANOMALI DATA PADA
HASIL PENGUKURAN SENSOR WIRELESS
BODY AREA NETWORK**


Oleh :
Fajar Setiawan
NRP. 2110155005


Proyek Akhir ini Digunakan Sebagai Salah Satu Syarat Untuk
Memperoleh Gelar Sarjana Sains Terapan (S.ST)
di
Politeknik Elektronika Negeri Surabaya
2017

Disetujui oleh :

Tim Penguji Proyek Akhir :



1. Yuliana Setiowati, S.Kom., M.Kom.
NIP. 197807062002122003


2. Irma Wulandari, S.Si., M.Sc.
NIP. 198010032015042001


3. Iwan Syarif, S.Kom., M.Sc., Ph.D.
NIP. 196904041995121002

Dosen Pembimbing :


1. M. Udin Harun Al Rasvid, Ph.D.
NIP. 198108082005011001


2. Isbat Uzzin Nadhori, S.Kom., M.T.
NIP. 197405052003121002

Mengetahui,
Ketua Program Studi D4 Teknik Informatika
Departemen Teknik Informatika dan Komputer
Politeknik Elektronika Negeri Surabaya


Isbat Uzzin Nadhori, S.Kom., M.T.
NIP. 197405052003121002

ABSTRAK

Wireless Sensor Network (WSN) rentan terhadap berbagai macam kesalahan sensor dan pengukuran. Kelemahan ini menghambat efisiensi dari berbagai macam aplikasi WSN, seperti pada layanan kesehatan. Contohnya, kesalahan pengukuran sensor ini dapat memberikan false alarm dan kesalahan penanganan oleh perawat atau dokter. Maka dari itu, sebuah sistem untuk membedakan antara kondisi medis yang sesungguhnya dan false alarm dapat digunakan untuk meningkatkan kualitas sistem monitoring pasien secara remote dan layanan kesehatan lainnya. Pada proyek ini, sebuah pendekatan diusulkan untuk melakukan pendeteksian anomali dengan menggunakan sebuah metode prediksi. Tujuan dari proyek ini adalah membuat sebuah sistem yang digunakan untuk membedakan anomali dan data kesehatan yang benar pada data hasil pengukuran WBAN. Sistem ini memprediksi nilai parameter menggunakan data-data terdahulu dan membandingkannya dengan data yang didapatkan dari pengukuran sebenarnya. Perbedaan dari dua nilai tersebut dibandingkan dengan nilai threshold, yang disesuaikan secara dinamis. Lalu menggunakan majority voting untuk menentukan data tersebut anomali atau tidak. Pendekatan ini diujicobakan pada dataset kesehatan yang sesungguhnya dan membandingkan metode prediksi dan ukuran sliding window yang digunakan. Hasil implementasi menunjukkan keefektifan dari sistem, ditunjukkan dengan rasio pendeteksian tinggi dan rasio kesalahan yang kecil.

Kata kunci : *Wireless Body Area Network, pendeteksian anomali, metode prediksi, dynamic threshold, majority voting.*

ABSTRACT

Wireless Sensor Networks (WSN) are vulnerable to various sensor faults and faulty measurements. This vulnerability reduces the efficiency of many WSN applications, such as healthcare (WBAN). For example, faulty measurement from sensor gives a false alarm to healthcare personnel and lead to wrong patient's handling. Therefore, a system to differentiate between real medical condition and a false alarm will improve remote patient monitoring systems and other healthcare service using WBAN. In this project, a novel approach is proposed to do anomaly detection using prediction method. The objective of this project is to make a system which can differentiate between real medical conditions and false alarms. This system predicts a sensor value from historic values and compares it with actual data from real measurement. The difference is compared to a threshold value, which is dynamically adjusted. Then using majority voting algorithm to determine whether the data is an anomaly or not. The proposed approach has been applied to real datasets and compare the prediction methods and the size of the sliding window. Experimental results shows the effectiveness of the system, indicated by high Detection Rate and low False Positive Rate.

Keywords : *Wireless Body Area Network, anomaly detection, prediction method, dynamic threshold, majority voting.*

KATA PENGANTAR

Alhamdulillah Rabbil'aalamin, segala puji syukur kepada Allah SWT. Karena atas segala kemudahan dan kelapangan yang diberikan sehingga penulis dapat menyelesaikan proyek akhir ini yang berjudul: **“PENDETEKSIAN ANOMALI DATA PADA HASIL PENGUKURAN SENSOR WIRELESS BODY AREA NETWORK”**

Laporan proyek akhir ini disusun sebagai salah satu syarat lulus akademis dengan memperoleh gelar Sarjana Sains Terapan (S.ST) di Politeknik Elektronika Negeri Surabaya.

Terdapat beberapa literatur dan teori baik yang diperoleh dalam perkuliahan maupun dari luar perkuliahan yang digunakan dalam penyelesaian proyek akhir ini dan juga tidak lepas dari dukungan dosen pembimbing serta pihak-pihak lain yang telah banyak memberikan semangat dan bantuan.

Penulis menyadari bahwa buku proyek akhir ini masih memiliki banyak kekurangan. Untuk itu penulis memohon maaf sebesar-besarnya atas kekurangan-kekurangan yang ada pada buku proyek akhir ini dan mengharapkan saran dan kritik dari semua pihak demi kesempurnaan buku ini.

Besar harapan penulis agar buku proyek akhir ini dapat bermanfaat khususnya bagi penulis dan pembaca pada umumnya untuk menambah wawasan dan pengetahuan.

Surabaya, Januari 2017

Penulis

UCAPAN TERIMA KASIH

Puji dan syukur ke hadirat Allah SWT atas rahmat dan hidayah-Nya. Terima kasih layak disampaikan kepada pihak yang telah membantu terselesaikannya proyek akhir ini :

1. Kedua orang tua tercinta yang telah memberikan do'a, kasih sayang, semangat, nasihat dan nafkah yang tak bisa penulis balas semua kasih sayangnya.
2. Bapak Zaenal Arief, ST., MT. selaku Direktur Politeknik Elektronika Negeri Surabaya (PENS) yang telah memberikan dukungan untuk terselesaikannya Proyek Akhir ini.
3. Bapak Tri Harsono, M.Kom., P.hD. selaku Kepala Departemen Teknik Informatika dan Komputer Politeknik Elektronika Negeri Surabaya (PENS) yang membimbing hingga terselesaikannya Proyek Akhir ini.
4. Bapak M. Udin Harun Al Rasyid, S.Kom, Ph.D selaku dosen pembimbing pertama yang membimbing penulis hingga Proyek Akhir ini selesai.
5. Bapak Isbat Uzzin Nadhori, S.Kom, M.T. selaku dosen pembimbing kedua yang membimbing penulis hingga Proyek Akhir ini selesai.
6. Keluarga tercinta yang memberikan semangat dan dukungan dalam menjalankan pendidikan perkuliahan.
7. Teman-teman kelas D4LJ Informatika 2015 yang memberikan semangat dan kenangan yang bermanfa'at dan keren selama 1,5 tahun berjuang bersama.
8. Kepada teman-teman yang kamar kosnya sering saya tumpangi kalau sedang menunggu bimbingan atau kuliah.
9. Ehem.....
10. Dan semua pihak yang membantu dan memperlancar Proyek Akhir ini.

DAFTAR ISI

HALAMAN SAMPUL.....	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
UCAPAN TERIMA KASIH	vi
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
DAFTAR KODE	xi
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 PERUMUSAN MASALAH.....	3
1.3 TUJUAN	3
1.4 KONTRIBUSI.....	3
1.5 METODOLOGI	3
1.6 SISTEMATIKA PENULISAN	5
BAB II DASAR TEORI	7
2.1 DASAR PERMASALAHAN.....	7
2.2 PENELITIAN TERKAIT.....	8
2.3 KEUNIKAN.....	14
2.4 LANDASAN TEORI	16
2.4.1 Wireless Body Area Network	16
2.4.2 WEKA	17
2.4.3 Sequential Minimal Optimization Regression	18
2.4.4 Gaussian Process	19
2.4.5 PhysioNet	20

2.4.6	Confusion Matrix	21
BAB III PERANCANGAN DAN PEMBUATAN SISTEM		
3.1	DESKRIPSI UMUM.....	23
3.2	DIAGRAM ALUR PENDETEKSIAN ANOMALI	24
3.3	PERBANDINGAN METODE PREDIKSI	26
3.4	KONFIGURASI DATASET.....	27
3.5	PERANCANGAN APLIKASI.....	28
3.5.1	Pemasangan Library WEKA	29
3.5.2	Perancangan Antar-Muka	30
3.5.3	Konfigurasi Metode Prediksi.....	32
3.5.4	Pembuatan Grafik	40
3.5.5	Dynamic Threshold	42
BAB IV IMPLEMENTASI DAN ANALISA		
4.1	BAHAN DAN ALAT	43
4.2	UJI COBA APLIKASI.....	44
4.3	DATASET UNTUK IMPLEMENTASI	50
4.4	HASIL IMPLEMENTASI DAN ANALISA	52
BAB V PENUTUP		
5.1	KESIMPULAN	55
5.2	SARAN	55
DAFTAR PUSTAKA.....		
		61
BIODATA PENULIS.....		
		63

DAFTAR GAMBAR

Gambar 2.1	Diagram alir sistem	13
Gambar 2.2	Aliran data pada BAN	16
Gambar 3.1	Gambaran umum sistem.....	23
Gambar 3.2	Diagram alur pendeteksian anomali	24
Gambar 3.3	Grafik RMSE hasil uji coba metode prediksi	27
Gambar 3.4	Contoh dataset mimicdb dalam format arff.....	28
Gambar 3.5	Class weka.....	29
Gambar 3.6	Menambahkan library.....	29
Gambar 3.7	Import class-class weka.....	30
Gambar 3.8	Desain antar-muka.....	31
Gambar 3.9	Antar-muka setelah proses pendeteksian.....	32
Gambar 3.10	Import class-class weka	33
Gambar 3.11	Memilih atribut pada weka	36
Gambar 3.12	Pengaturan lag variables pada weka	38
Gambar 3.13	Memilih SMOReg pada Weka	39
Gambar 3.14	Grafik parameter	41
Gambar 4.1	Tampilan awal aplikasi.....	44
Gambar 4.2	Aplikasi menampilkan isi dataset	45
Gambar 4.3	Grafik dari dataset	46
Gambar 4.4	Mengatur metode prediksi & ukuran sliding window	47
Gambar 4.5	Hasil pendeteksian.....	48
Gambar 4.6	Grafik hasil pendeteksian	49
Gambar 4.7	Analisa hasil pendeteksian	50

DAFTAR TABEL

Tabel 2.1	Tabel perbandingan penelitian	15
Tabel 2.2	Tabel confusion matrix	21
Tabel 3.1	Method pada class Instances	34
Tabel 3.2	Method pada class WekaForecaster	35
Tabel 4.1	Tabel keterangan dataset	51
Tabel 4.2	Tabel ketepatan metode SMOReg (dalam %).....	52
Tabel 4.3	Tabel ketepatan metode Gaussian (dalam %)	53
Tabel 4.4	Tabel kecepatan pemrosesan (dalam detik/baris data) ...	54

DAFTAR KODE

Kode 3.1 Pembuatan forecaster dan pemilihan atribut.....	37
Kode 3.2 Pengaturan lag variables dengan kode.....	38
Kode 3.3 Pemilihan metode dengan menggunakan kode.....	39
Kode 3.4 Menjalankan prediksi	40

BAB I

PENDAHULUAN

Bab ini akan membahas mengenai berbagai macam pengenalan tentang proyek akhir ini. Didalamnya berisi penjelasan mengenai proyek akhir ini mulai dari latar belakang, rumusan masalah, tujuan dan manfaat, metodologi dan sistematika penulisan yang digunakan.

1.1 LATAR BELAKANG

Wireless Sensor Network atau sering disebut dengan WSN telah digunakan pada berbagai jenis aplikasi untuk domain-domain tertentu, seperti aplikasi kesehatan, aplikasi pertanian, dan aplikasi-aplikasi pengamatan kondisi lingkungan. Wireless Sensor Network dapat digunakan dalam bidang kesehatan untuk meningkatkan kinerja manajemen layanan kesehatan [1]. Sensor untuk kesehatan berukuran kecil dan mampu melakukan pekerjaan seperti mengambil berbagai parameter-parameter fisiologis, seperti detak jantung (HR), tekanan darah (BP), denyut nadi, *respiratory rate* (RESP) dan kadar oksigen dalam darah (SPO₂). Dalam pengukurannya, sensor-sensor tersebut harus dipasangkan pada tubuh subyek yang diamati dan secara terus-menerus diamati baik dari rumah sakit atau rumah [2]. Hasil dari sensor-sensor tersebut dapat digunakan lebih lanjut dalam aplikasi, contohnya seperti aplikasi penentuan penyakit dan aplikasi pengambil keputusan untuk penanganan orang sakit.

Pada suatu penelitian [3], dijelaskan bahwa anomali pada nilai data yang dihasilkan oleh sensor juga dapat disebabkan banyak alasan, seperti kesalahan pada hardware, sensor yang rusak, kehabisan energi, kalibrasi, gangguan elektromagnetik, pasien yang berkeringat, sensor yang terlepas, kegagalan pemakaian (*malfuction*), data yang diinjeksikan, pemakaian yang salah, dan sebagainya. Beberapa faktor diatas dapat menyebabkan data yang dikirimkan menjadi tidak akurat dan pada akhirnya

mengakibatkan pemberian informasi yang salah atau biasa dengan disebut *false alarm*.

Seperti yang telah disebutkan diatas, sangatlah penting untuk dapat mendeteksi ketidakakuratan data yang dikirimkan oleh sensor. Proses deteksi tersebut dilakukan sebelum aplikasi mengolahnya menjadi informasi yang ditampilkan pada pengguna agar tidak menyebabkan *false alarm*. Dalam kasus monitoring data, jumlah data akan menjadi sangat banyak seiring dengan berjalannya waktu. Sehingga, apabila tidak ada pemrosesan yang cepat terhadap data maka pendeteksian data yang tidak akurat menjadi tidak mungkin.

Beberapa metode telah dikembangkan untuk melakukan pendeteksian anomali data. Contohnya metode terdistribusi yang proses pendeteksiannya dilakukan pada masing-masing sensor node, akan tetapi metode ini membutuhkan sumber daya yang besar pada sisi sensor seperti daya baterai dan memori. Contoh lain adalah dengan menggunakan metode terpusat yang proses pendeteksiannya dilakukan pada level aplikasi yang tentunya setiap sensor node harus mengirim data secara bersamaan agar proses dapat berjalan dengan baik.

Pada proyek akhir ini, kami membangun suatu sistem yang dapat melakukan proses pendeteksian anomali data pada level aplikasi. Kami menggunakan metode terpusat karena dataset yang kami gunakan merupakan dataset yang datanya didapatkan dari hasil pengukuran menggunakan satu buah node. Untuk melakukan pendeteksian, kami menggunakan metode peramalan dengan memanfaatkan algoritma SMOReg disediakan oleh WEKA. Dengan menggunakan metode peramalan tidak diperlukan data training, karena data yang digunakan adalah data pengukuran sebelumnya. Metode peramalan tersebut digabungkan dengan *dynamic threshold* yang dapat memberikan nilai batas error sesuai dengan keadaan yang terjadi. Pada tahap akhir terdapat metode *majority voting* yang dapat membedakan antara alarm medis dan alarm anomali. Dengan menggunakan metode ini, kami dapat melakukan

proses pendeteksian anomali secara cepat dengan proses yang cepat dan memiliki tingkat ketepatan yang tinggi.

1.2 PERUMUSAN MASALAH

Berdasarkan dari uraian latar belakang diatas, maka dapat dirumuskan suatu permasalahan yaitu data anomali yang didapatkan dari hasil pengukuran sensor menyebabkan terjadinya *false alarm* yang dapat menyebabkan dampak-dampak negatif dalam pemberian layanan kesehatan. Selain itu, dibutuhkan sebuah algoritma pendeteksian anomali yang cepat agar dapat digunakan melakukan proses data parameters fisiologis yang datang tiap detik.

1.3 TUJUAN

Tujuan dari proyek akhir ini adalah membuat suatu sistem yang dapat digunakan untuk membedakan antara data kesehatan pasien yang memang benar-benar keadaan nyata dan data yang merupakan anomali atau kesalahan sehingga pengguna data kesehatan tahu di bagian mana data tersebut dapat digunakan. Serta membangun sebuah sistem pendeteksian anomali yang dapat melakukan proses data kurang dari satu detik untuk setiap data yang masuk ke dalam sistem.

1.4 KONTRIBUSI

Hasil dari tugas akhir ini diharapkan dapat dijadikan pertimbangan penggunaan algoritma pendeteksian anomali pada data-data hasil pengukuran parameter-parameter fisiologis dari seorang pasien oleh sensor WBAN.

1.5 METODOLOGI

Metodologi yang digunakan dalam proyek akhir ini sebagai berikut:

1. Studi pustaka

Tahap awal dari pengerjaan proyek akhir ini adalah mencari referensi untuk menunjang pengerjaan proyek akhir ini seperti

studi literatur terhadap konsep serta metode yang digunakan, pengumpulan data pendukung serta *software* yang akan digunakan. Sekaligus mencari landasan teori bagi pemecahan masalah yang dihadapi.

2. Pengumpulan Alat dan Bahan

Tahap ini merupakan tahapan dimana penulis melakukan persiapan, pengumpulan alat dan bahan yang berhubungan dengan proyek akhir ini.

Alat yang digunakan diantaranya adalah:

- Laptop dengan sistem operasi Windows 10 64-bit.
- WEKA Library.
- Dataset MIMICDB dari PhysioNet.

3. Perancangan Sistem

Dari literatur yang didapatkan, selanjutnya dilakukan analisa dan perencanaan terhadap implementasi WSN yang akan dibuat pada proyek akhir ini.

4. Pembuatan Sistem

Tahap implementasi dari perancangan sistem yang akan digunakan untuk menyelesaikan tugas akhir ini.

Alur dalam pengerjaan proyek akhir ini adalah sebagai berikut:

1. Menentukan dataset mana yang akan digunakan dalam proses pengerjaan proyek akhir ini.
2. Melakukan konfigurasi terhadap format dataset yang telah dipilih agar dapat digunakan dalam lingkungan aplikasi yang akan dibuat.
3. Melakukan percobaan berupa pengetesan dataset pada lingkungan WEKA.
4. Mencari dan mempelajari tentang library WEKA yang akan digunakan sebagai dasar pembuatan aplikasi.

5. Membuat aplikasi yang digunakan untuk melakukan pendeteksian anomali pada dataset dengan menggunakan java.
6. Melakukan proses pendeteksian anomali pada dataset dengan menggunakan aplikasi yang telah dibuat.
7. Menampilkan hasil pendeteksian berupa laporan berbentuk grafik yang menampilkan letak anomali pada masing-masing parameter kesehatan serta melakukan analisis terhadap hasil pendeteksian yang dilakukan.

5. Pengujian dan Analisa

Pengujian ini dimaksudkan untuk memastikan bahwa kinerja masing-masing sistem dapat berfungsi sesuai dengan yang tujuan.

Pengujian yang dilakukan meliputi:

- a. Menguji pembacaan dataset pada program.
- b. Menguji fitur yang ada pada sistem.
- c. Menguji sistem dengan mengganti metode prediksi dan ukuran *sliding window*.
- d. Menguji hasil pendeteksian anomali dengan menggunakan *Confusion Matrix*.

6. Pembuatan Laporan

Pada tahap ini dilakukan dokumentasi dari semua tahapan proses diatas. Dokumentasi disusun dalam bentuk laporan yang berisi tentang dasar teori dan metode yang digunakan serta hasil yang diperoleh selama pengerjaan proyek akhir.

1.6 SISTEMATIKA PENULISAN

Sistematika pembahasan yang akan diuraikan dalam buku laporan proyek akhir ini terbagi dalam bab-bab yang akan dibahas sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan berisi latar belakang, rumusan masalah, tujuan dan manfaat, metodologi dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini membahas teori-teori penunjang yang didapatkan dari beberapa hasil referensi serta sumber terkait lainnya yang berhubungan dengan penyelesaian proyek akhir ini. Selain dari literatur, terdapat juga penelitian-penelitian terdahulu yang berhubungan dengan proyek akhir ini.

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

Bab ini memuat perancangan sistem yang dibuat, meliputi perancangan sistem, perancangan database, dan perancangan desain antar-muka untuk pengguna.

BAB IV IMPLEMENTASI DAN ANALISA

Bab ini menampilkan dan menjelaskan seluruh hasil dan analisa dari implementasi proyek akhir.

BAB V PENUTUP

Bab yang berisi kesimpulan dari proyek tugas akhir, saran untuk pengembangan, perbaikan, maupun penyempurnaan pada aplikasi yang telah dibuat

DAFTAR PUSTAKA

Berisi tentang referensi-referensi yang telah digunakan sebagai landasan selama pembuatan proyek akhir ini.

BAB II

DASAR TEORI

Bab ini akan menjelaskan mengenai dasar permasalahan dari proyek ini, penelitian-penelitian serupa yang telah ada, dan penjelasan mengenai teori yang dipakai sebagai acuan serta pengenalan tentang teknologi yang digunakan pada proyek akhir ini.

2.1 DASAR PERMASALAHAN

Sensor untuk kesehatan berukuran kecil dan mampu melakukan pekerjaan seperti mengambil berbagai parameter-parameter fisiologis, seperti detak jantung (HR), tekanan darah (BP), denyut nadi, *respiratory rate* (RESP) dan kadar oksigen dalam darah (SPO_2). Dalam pengukurannya, sensor-sensor tersebut harus dipasangkan pada tubuh subyek yang diamati dan secara terus-menerus diamati baik dari rumah sakit atau rumah [1]. Hasil dari sensor-sensor tersebut dapat digunakan lebih lanjut dalam aplikasi, contohnya seperti aplikasi penentuan penyakit dan aplikasi pengambil keputusan untuk penanganan orang sakit.

Pada suatu penelitian [2], dijelaskan bahwa anomali pada nilai data yang dihasilkan oleh sensor juga dapat disebabkan banyak alasan, seperti kesalahan pada hardware, sensor yang rusak, kehabisan energi, kalibrasi, gangguan elektromagnetik, pasien yang berkeringat, sensor yang terlepas, kegagalan pemakaian (*malfunction*), data yang diinjeksikan, pemakaian yang salah, dan sebagainya. Beberapa faktor diatas dapat menyebabkan data yang dikirimkan menjadi tidak akurat dan pada akhirnya mengakibatkan pemberian informasi yang salah atau biasa dengan disebut *false alarm*. Pemberian informasi yang salah ini juga dapat mengakibatkan dampak pada aplikasi yang dapat memberikan output yang tidak tepat dan dapat menyebabkan pemberian perawatan dan obat secara sia-sia.

Maka dibutuhkan sebuah aplikasi yang dapat digunakan dalam lingkungan pelayanan kesehatan yang bertugas untuk mendeteksi atau

bahkan memperbaiki anomali data pada data keadaan pasien. Dengan begitu, data pasien yang didapatkan dari pengukuran menggunakan alat sensor-sensor kesehatan dapat digunakan sebagai acuan oleh tenaga kesehatan untuk memberikakan tindakan pengobatan atau yang lainnya kepada seorang pasien.

2.2 PENELITIAN TERKAIT

Terdapat beberapa penelitian yang memberikan kontribusi dalam pembuatan proyek akhir ini diantaranya adalah:

1. Penelitian dengan judul “**New and Fast Outlier Detection Scheme in WSN: NFDODS-WSN**” [3]. Peneliti fokus pada masalah pendeteksian outlier pada WSN. Pendeteksian outlier lebih banyak difokuskan pada konteks pengembang dan pengguna aplikasi yang jarang terjadi. Pada penelitian ini, penulis mengusulkan sebuah model yang berdasarkan perkiraan dari distribusi data sensor. Model yang diajukan memiliki beberapa teknik, yaitu model statistika untuk memodelkan data dari sensor yang didapatkan, *Kernel Density Estimation* untuk melakukan pemetaan terhadap data, pembobotan menggunakan metode pembobotan aritmatika dan pembobotan ekponensial, dan pendeteksian outlier dengan menghitung faktor deviasi (DEVF) dan *Normalized Deviation Factor* (NDEVF). Untuk melakukan percobaan terhadap metode yang diusulkan, peneliti menggunakan dataset yang didapatkan dari laboratorium penelitian Intel Berkeley yang berisi data dari 54 sensor dan dijalankan selama 42 hari. Setelah dilakukan percobaan, dapat disimpulkan bahwa algoritma yang diusulkan memberikan presisi yang tinggi dalam pendeteksian outlier. Penelitian ini bertujuan untuk membangun suatu sistem manajemen air untuk tanaman. Pada penelitian ini, penulis menggunakan sensor kelembaban dan suhu, lalu menggunakannya sebagai masukan dalam perhitungan fuzzy. Keluaran ditampilkan pada layar LCD berukuran 16x2.

2. Penelitian dengan judul “**Outliers detection and classification in wireless sensor networks**” yang diterbitkan oleh Egyptian Informatics Journal. Dalam penelitian ini, penulis memberikan suatu pendekatan untuk mendeteksi anomali data yang dapat membedakan antara *error* karena kerusakan sensor dan *error* karena suatu kejadian dengan menggolongkan hubungan *spatial* dan *temporal* antara kejadian yang diamati oleh titik sensor pada lingkungan jaringan yang dibatasi [4]. Terdapat 4 buah proses yang harus dilalui dalam penelitian ini. Tahap pertama adalah proses *pre-processing* yang dilakukan menggunakan metode pengklasteran untuk mengelompokkan data dalam kluster-kluster. Setelah itu melakukan proses *Outlier Detection* dengan cara melakukan pendeteksian outlier pada setiap kluster yang terbentuk dan melabeli masing-masing kluster apakah kluster tersebut merupakan kluster outlier atau tidak. Pendeteksian anomali tersebut dilakukan dengan menghitung jarak kluster dengan kluster lainnya lalu membandingkannya dengan standar deviasi dari jarak-jarak dalam kluster menggunakan rumus Euclidean. Setelah menentukan kluster mana saja yang merupakan anomali, selanjutnya adalah menentukan kluster anomali tersebut merupakan anomali karena kesalahan atau anomali karena sebuah kejadian. Penentuan jenis anomali ini dilakukan dengan melihat jumlah pembacaan dan jarak waktu terjadinya anomali. Apabila anomali terjadi pada banyak sensor dan pada waktu yang berdekatan maka anomali tersebut dianggap sebagai anomali kejadian, apabila tidak maka dianggap sebagai anomali karena kesalahan. Setelah semua proses diatas dilalui dan didapatkan hasil pendeteksian anomali, selanjutnya adalah mengukur tingkat kebenaran dari sistem yang telah dibuat. Peneliti melakukan perhitungan berdasarkan jumlah error pembacaan sensor dibagi dengan jumlah seluruh pembacaan sensor (persamaan 1). Jadi, semakin tinggi nilai persentase yang didapatkan, maka tingkat kepercayaan pada sistem semakin rendah.

$$Trust(s_i) = \left(1 - \left(\frac{N_{oi}}{N_i}\right) \times 100\right) \quad (1)$$

Dimana, $Trust(s_i)$ adalah tingkat kepercayaan pada sensor s_i yang merupakan presentase pembacaan error dari seluruh pembacaan, N_{oi} adalah jumlah pembacaan error pada sensor s_i , dan N_i adalah jumlah dari seluruh pembacaan untuk sensor s_i . Peneliti membandingkan hasil dari sistem yang dikembangkannya ini dengan algoritma COLLECT. Setelah dilakukan pengujian menggunakan dataset asli dan *synthetic dataset*, peneliti menemukan bahwa hasil dari algoritma yang dikembangkannya memiliki hasil yang lebih baik. Pada pengujian menggunakan dataset asli, algoritma ini memiliki waktu eksekusi lebih cepat dan tingkat pendeteksian lebih tinggi yang mencapai 100%. Pada pengujian menggunakan dataset sintetis, dilakukan beberapa kali pengujian dengan membedakan ukuran dari dataset. Hasilnya, algoritma ini memiliki waktu eksekusi yang lebih tinggi, tingkat pendeteksian mencapai 100% yang tidak pernah terjadi saat menggunakan COLLECT dan memiliki *false alarm* yang lebih kecil.

Penelitian ini bertujuan untuk membuat suatu alat yang digunakan untuk mengontrol suhu dan kelembaban tanaman anggrek. Pada proyek ini menggunakan 2 buah sensor yang terdiri dari 1 buah sensor suhu dan 1 buah sensor kelembaban. Sensor-sensor diatas diakumulasikan dengan potensiometer yang telah terintegrasi dengan suatu rangkaian. Setelah diproses, mikrokontroller akan memberikan perintah untuk menghidupkan atau mematikan aktuator yang berupa motor semprot dan kipas.

3. Penelitian dengan judul “Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies”. Penelitian ini bertujuan untuk melakukan *survey* terhadap berbagai strategi yang dilakukan dalam melakukan pendeteksian anomali pada WSN dan mengelompokkannya berdasarkan tipe-tipe anomali yang dideteksi [5]. Pada bagian awal, penulis menyebutkan dan mendefinisikan berbagai macam anomali yang terdapat pada

WSN berdasarkan beberapa kategori, yaitu anomali pada jaringan, anomali data, dan anomali titik sensor. Pada kategori anomali jaringan, terdapat 4 buah gejala yang terjadi, yaitu hilangnya sambungan yang dapat dideteksi dengan melihat tingkat pengiriman paket, sambungan tidak stabil yang dideteksi dengan adanya tingkat pengiriman paket yang tidak beraturan, *routing loop* yang terjadi apabila alamat yang dituju sama dengan alamat dari sumber, dan *broadcast storm* yang terjadi karena adanya sambungan yang terputus pada jaringan sensor sehingga titik sensor yang berhubungan dengan titik sensor tersebut akan melakukan *broadcast* secara terus menerus. Kategori selanjutnya adalah anomali pada titik sensor yang terdiri dari masalah panel matahari, penurunan energi baterai, kerusakan titik sensor yang dapat dideteksi apabila titik sensor tersebut tidak melakukan hubungan dengan titik sensor lain, dan *node reset* yang dapat diketahui dengan melihat *packet counter* pada titik sensor apakah memiliki nilai 0. Kategori yang terakhir adalah anomali data yang memiliki 3 buah gejala, yaitu *temporal anomalies* yang dilihat dari waktu terjadinya anomali, *spatial anomalies* yang dilihat dari variasi data, dan *spatiotemporal anomalies* yang merupakan gabungan dari 2 gejala sebelumnya. Anomali yang tidak termasuk pada 3 kategori diatas dimasukkan pada kategori “anomali lainnya”, seperti kesalahan aplikasi yang dapat dilakukan perbaikan secara offline. Setelah mendefinisikan tipe-tipe anomali, penulis juga melakukan penelitian tentang strategi-strategi yang dapat digunakan dalam melakukan pendeteksian anomali yang telah dikembangkan oleh orang lain sebelumnya. Berbagai strategi dianalisa berdasarkan beberapa kategori, yaitu konsep, status dari strategi apakah sebuah algoritma atau alat, type dari anomali apakah jaringan atau data atau titik sensor, arsitektur sistem apakah terdistribusi atau terpusat, dan penggunaannya. Setelah melakukan penelitian terhadap berbagai tipe anomali dan strategi pendeteksian anomali, peneliti menyimpulkan petunjuk untuk

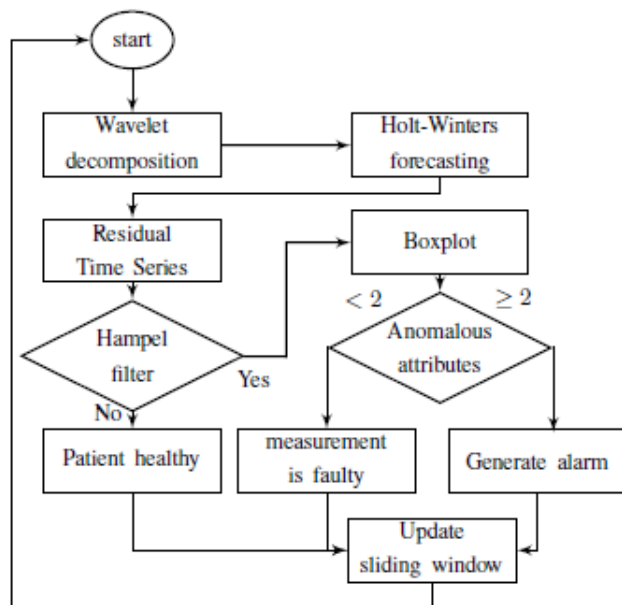
setiap desain dari sistem pendeteksian anomali. Berikut adalah beberapa poin yang disimpulkan oleh peneliti:

- a) Untuk melakukan pendeteksian anomali jaringan, dapat menggunakan metode terpusat apabila data jaringan dapat dikirimkan bersama paket. Metode terdistribusi dan *hybrid* lebih cocok untuk anomali jaringan yang lebih kompleks seperti *routing loop*, dimana titik sensor dapat melihat isi paket yang mereka kirimkan untuk mendeteksi *looping*.
- b) Untuk melakukan pendeteksian anomali pada titik sensor, dapat menggunakan arsitektur terpusat, terdistribusi, maupun *hybrid*. Tetapi untuk melakukan pencarian terhadap sumber dari anomali dengan menggunakan arsitektur terpusat tidak dapat dilakukan, karena hanya dapat memberikan informasi tentang sebab terjadinya anomali. Arsitektur terdistribusi memberikan pendeteksian yang lebih responsif.
- c) Untuk melakukan pendeteksian terhadap anomali data, pendeteksian terpusat lambat dibandingkan dengan yang terdistribusi. Metode *hybrid* lebih cocok karena dapat mendeteksi masalah secara lokal lalu melakukan analisis lebih lanjut.
- d) *Usability* merupakan faktor besar lain yang ada pada pendeteksian anomali pada WSN. Alat-alat pendeteksian juga harus menyediakan beberapa cara memberikan pemberitahuan, seperti email ataupun sms. Selain dapat memberikan pemberitahuan, alat juga harus menerima pengaturan tentang kapan, seberapa sering, dan seberapa penting sebuah pemberitahuan yang akan diterima. Selain dapat memberikan pemberitahuan, dapat juga ditambahkan fitur untuk menerima *feedback* dari pengguna.

Penelitian ini bertujuan untuk mendesain, membangun, dan mengimplementasikan WSN yang terhubung dengan ZigBee yang dihubungkan dengan CMS dan juga menghasilkan parameter GPS. Sistem ini diharapkan dapat membantu petani

mengevaluasi keadaan tanah dan membuat petani dapat bertindak dengan baik.

4. Penelitian dengan judul “Online Anomaly Detection in Wireless Body Area Network for Reliable Healthcare Monitoring” [6]. Penelitian ini bertujuan untuk mengusulkan sebuah pendekatan yang ringan untuk pendeteksian kesalahan pengukuran secara *online* dengan menganalisa data yang telah didapatkan dari WBAN. Kerangka yang diusulkan melakukan analisis data secara sekuensial menggunakan *smartphone* sebagai media pemrosesan. Tujuan utamanya adalah untuk membangkitkan alarm hanya pada saat pasien membutuhkan perawatan darurat dan membuang semua *false alarm* yang disebabkan oleh kesalahan pembacaan sensor atau kerusakan pada sensor.



Gambar 2.1 Diagram alir sistem [6]

Pendekatan yang diusulkan menggunakan *Discrete Haar wavelet transform* (DWT) sebagai dasar, *non-seasonal Holt-Winters* (NSHW) sebagai metode prediksi, *Hampel filter* dan *boxplot*. DWT, NSHW dan Hampel digunakan untuk mendeteksi deviasi spasial, dan boxplot untuk melakukan analisis temporal. Gambar 1 merupakan diagram alir dari pendekatan yang diusulkan, dimana 4 algoritma (DWT, NSHW, Hampel dan Boxplot) digunakan untuk memastikan bahwa alarm hanya akan dinyalakan apabila pasien dalam keadaan kritis.

Untuk melakukan percobaan terhadap pendekatan ini, peneliti menggunakan dataset MIMICDB dari PhysioNet. Percobaan dilakukan sebelum dan sesudah dimasukkannya anomali dalam dataset. Peneliti juga membandingkan hasil dari metode yang dikembangkan dengan hasil dengan menggunakan metode lainnya, yaitu *Mahalanobis Distance* (MD). Hasilnya, kedua metode tersebut memiliki tingkat pendeteksian 100%. Tetapi pada tingkat *false alarm*, pendekatan pada penelitian ini lebih baik karena memiliki nilai 67% sedangkan dengan menggunakan MD memiliki nilai 16%. Hasil tersebut membuktikan bahwa pendekatan dalam penelitian ini memiliki hasil yang lebih baik dibandingkan dengan menggunakan metode MD.

Penelitian ini bertujuan untuk mengukur dan memantau keadaan tanah dan keadaan iklim. Untuk memantau keadaan tanah sensor yang digunakan adalah sensor kelembaban tanah, kadar air tanah, suhu tanah, konduktifitas tanah, tingkat keasaman tanah dan untuk memantau iklim menggunakan sensor radiasi cahaya, kecepatan angin, suhu.

2.3 KEUNIKAN

Dari beberapa penelitian terkait diatas, dapat dibuat sebuah tabel perbandingan mengenai pendeteksian anomali yang telah dibuat. Berikut adalah tabel perbandingannya dengan ditambahkan spesifikasi dari pendeteksian anomali yang akan dibuat.

No.	Penelitian	Jenis	Metode	Arsitektur	Catatan/Algoritma
1	[3]	Simulasi	Statistika	Terdistribusi	KDE, NDEVF
2	[5]	Simulasi	Klaster	-	-
3	[6]	Survey	-	-	Menentukan arsitektur terbaik untuk jenis-jenis anomali
4	[7]	Simulasi	Prediksi, <i>filtering</i>	Terpusat	DWT, NSHW, Hampel filter, boxplot
Proyek ini		Simulasi Realtime	Prediksi, <i>dynamic threshold, voting</i>	Terpusat	SMOReg, <i>dynamic threshold, majority voting</i>

Tabel 2.1 Tabel perbandingan penelitian

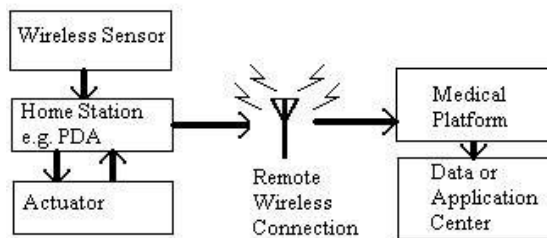
Pada tabel 2.1 dapat dilihat bahwa aplikasi atau sistem pendeteksian yang dibuat dalam proyek akhir ini memiliki beberapa keunikan dari penelitian lain yang memiliki tujuan yang hampir sama. Sistem yang dibuat dalam proyek akhir ini berjenis simulasi realtime karena meskipun data yang digunakan merupakan dataset dan bukan hasil pengukuran sendiri, tapi kami memperlakukannya seperti dalam keadaan pengukuran sebenarnya. Data dalam dataset akan diproses satu-persatu seperti pada saat data dari sensor masuk ke sistem secara satu-persatu. Selain itu, kami menggunakan metode prediksi untuk mendapatkan data pembandingan untuk data sensornya. Setelah itu dilakukan perhitungan dynamic threshold untuk nilai error yang dinamis sesuai dengan keadaan saat itu. Lalu dilakukan juga voting untuk membedakan keadaan anomali atau keadaan medis yang sebenarnya untuk meminimalkan kesalahan.

2.4 LANDASAN TEORI

2.4.1 Wireless Body Area Network [7]

Bertumbuhnya biaya layanan kesehatan pada negara-negara berkembang menjadi tantangan yang besar untuk pemerintah, penyedia layanan, dan industri layanan kesehatan. Ada perhatian besar tentang penggunaan teknologi *wireless* untuk melakukan *monitoring* pada pasien dengan biaya yang kecil dengan kualitas yang dapat diandalkan. *Wireless Body Area Network (WBAN)* adalah salah satu teknologi yang sedang tumbuh dan memiliki kemampuan yang signifikan untuk meningkatkan layanan kesehatan, membantu diagnosa, *disease-tracking*, dan prosedur-prosedur kesehatan yang terkait [8].

Aspek penting dalam WBAN adalah bahwa alat ini dapat memberikan komunikasi yang dapat diandalkan dengan alat-alat kesehatan, terutama yang ditanamkan dalam tubuh manusia. *Wireless Body Area Network* terdiri dari sensor-sensor yang murah, ringan, dan kecil yang dipasangkan pada tubuh, digabungkan dengan pakaian, serta dapat dipasangkan pada organ dalam tubuh. Tujuan utamanya adalah untuk memudahkan dokter dan staf medis untuk memonitor status medis pasien dengan aman dan mudah. Teknologi WBAN ini memberikan solusi yang murah dan efisien kepada orang-orang.



Gambar 2.2 Aliran data pada BAN [7]

Gambar 2.2 merupakan skema umum dari aliran data dalam penggunaan WBAN di dunia nyata. Data yang diperoleh dari sensor

akan diteruskan pada pusat dari sensor yang dapat berupa PDA, *smartphone* atau alat lainnya yang memiliki sumber daya yang lebih besar. Data tersebut akan digunakan untuk memberikan perintah pada aktuator dan juga dikirimkan pada suatu alat kesehatan menggunakan koneksi nirkabel. Alat kesehatan tersebut akan memroses data tersebut menggunakan aplikasi tertentu untuk mendapatkan informasi yang diinginkan.

2.4.2 WEKA [8]

Weka workbench adalah kumpulan algoritma *machine learning* dan alat *preprocessing* data. Alat ini didesain sehingga pengguna dapat dengan cepat mencoba metode-metode yang ada pada dataset dengan mudah. Weka dikembangkan di University of Waikato, Selandia Baru; dan merupakan singkatan dari *Waikato Environment for Knowledge Analysis*. Sistem ini ditulis menggunakan Java dan didistribusikan dibawah GNU General Public License. Weka dapat dijalankan hampir pada semua platform dan telah dicoba pada Sistem Operasi Linux, Windows, dan Macintosh [8].

Weka menyediakan implementasi dari algoritma *machine learning* yang dapat dengan mudah diimplementasikan pada dataset. Di dalamnya juga terdapat berbagai macam alat-alat untuk mentranformasikan dataset, seperti algoritma untuk diskritisasi dan *sampling*. Dapat juga digunakan untuk melakukan *preprocessing* terhadap dataset, memasukkannya pada skema *learning*, dan menganalisa hasil serta performansinya tanpa arus menuliskan baris kode sama sekali. Alat ini memiliki algoritma-algoritma untuk masalah-masalah utama dalam data mining: regresi, klasifikasi, pengklasteran, asosiasi, dan pemilihan atribut.

Weka dapat digunakan menggunakan berbagai macam cara, yang paling mudah adalah dengan menggunakan GUI yang biasa disebut *Explorer* yang berupa pilihan menu-menu dan form yang harus diisi untuk menggunakannya. Ada beberapa GUI lain yang bisa digunakan, yaitu *Knowledge Flow* yang dapat digunakan untuk

melakukan desain konfigurasi untuk pemrosesan data, *Experimenter* yang didesain untuk membantu menjawab mengenai hal-hal dasar saat menggunakan klasifikasi dan regresi seperti “metode mana yang paling bagus untuk masalah ini?”, dan *Workbench* adalah GUI gabungan yang memiliki fungsi 3 GUI yang lainnya. Selain menggunakan berbagai macam GUI diatas, pengguna juga dapat menggunakan CLI yang dapat digunakan dengan memasukkan perintah-perintah dalam bentuk teks [8]. Selain menggunakan metode-metode dari aplikasi resmi dari Weka, pengguna juga dapat menggunakan aplikasi yang dikembangkan sendiri dengan menggunakan *library* yang telah disediakan dalam bahasa Java.

2.4.3 Sequential Minimal Optimization Regression (SMOReg) [9]

Sequential Minimal Optimization Regression atau SMOReg merupakan pengembangan dari algoritma Sequential Minimal Optimization (SMO) [9] dan merupakan implementasi dari algoritma *Support Vector Regression* (SVR).

Andaikan kita mempunyai j data training $[(x_1, y_1), \dots, (x_j, y_j)]$ yang bisa merupakan data medis lampau yang telah didapatkan dari pengukuran menggunakan sensor. Sebuah fungsi $f(x)$ dengan deviasi sebesar ε (error) yang didapatkan dari data training menggunakan metode SVR. Error akan diabaikan selama nilainya tidak melebihi ε . Hal ini penting karena apabila error lebih dari ε maka dapat membuat data medis tidak beraturan. Untuk fungsi linier f sebagai berikut [9]:

$$f(x) = \langle w, x \rangle + b \text{ dengan } w \in X, b \in R \quad (2)$$

Persamaan 2 digunakan untuk mencari w terkecil yang bisa didapatkan dengan menyederhanakan persamaan $\|w\|^2 = \langle w, w \rangle$. Berikut adalah persamaannya [9]:

$$\text{Meminimalkan} \quad \frac{1}{2} \|w\|^2$$

Terhadap:

$$\begin{aligned} y_i - \langle w, w \rangle - b &\leq \varepsilon \\ \langle w, w \rangle + b - y_i &\leq \varepsilon \end{aligned} \quad (3)$$

Variabel slack ξ_i, ξ_i^* diperkenalkan untuk mengerjakan permasalahan pada persamaan 3 menjadi seperti berikut [9]:

$$\text{Meminimalkan} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i, \xi_i^*)$$

Terhadap:

$$\begin{cases} y_i - \langle w, w \rangle - b \leq \varepsilon \\ \langle w, w \rangle + b - y_i \leq \varepsilon \\ \xi_i, \xi_i^* \leq \varepsilon \end{cases} \quad (4)$$

$C > 0$ adalah konstanta yang digunakan untuk menentukan rentang error.

2.4.4 Gaussian Process [10]

Proses Gaussian adalah sebuah model statistik dimana proses terjadi pada domain yang berkelanjutan, seperti waktu. Sebuah proses Gaussian ditentukan oleh fungsi rata-rata $\mu(x) = E[Y_x]$ dan fungsi peragam $k(x_i, x_j) = E[(Y_{x_i} - \mu(x_i))(Y_{x_j} - \mu(x_j))]$ [11].

Regresi menggunakan proses Gaussian dapat diturunkan dari proses regresi Bayesian. Misal untuk setiap output y_i bergantung pada input x_i dengan fungsi berikut.

$$y_i = f(x_i) + \varepsilon_i \quad (5)$$

Dengan ε_i adalah variabel nilai error yang menyebar dengan rata-rata nol dan ragam σ^2 , sedangkan x_i adalah vektor input ke- i dengan

$i = 1, 2, \dots, n$. Apabila fungsi-fungsi f dihimpun dalam sebuah vektor menjadi $f = [f_1, f_2, \dots, f_n]$ maka menurut proses gaussian regresi:

$$f|X, \theta \sim N(0, K) \quad (6)$$

Dengan K adalah matrik $n \times n$ yang berantung pada X dan θ sedangkan θ adalah vektor parameter dari fungsi peragam. Setiap elemen $k(i, j)$ dari matrik K adalah $k(x_i, x_j)$ yang selanjutnya disebut fungsi peragam. Maka persamaan (5) dapat dinyatakan dalam bentuk vektor sebagai berikut:

$$y = f + \varepsilon \quad (7)$$

Dengan y adalah vektor hasil, f adalah vektor dari fungsi-fungsi regresi, dan ε adalah vektor nilai error.

2.4.5 PhysioNet [12]

PhysioNet adalah sebuah forum online untuk penyebaran dan bertukar sinyal-sinyal biomedical dan sebuah aplikasi *open source* untuk menganalisisnya. Situs ini menyediakan fasilitas untuk menganalisis data secara bersama dan mengevaluasi algoritma baru. Terdapat 2 komponen lain yang masih berhubungan dalam PhysioNet, yaitu PhysioBank dan PhysioToolkit.

PhysioBank adalah sebuah arsip online yang sangat besar dan terus berkembang. Arsip ini menyimpan rekaman digital sinyal-sinyal fisiologis dan data-data lain yang berhubungan untuk keperluan komunitas penelitian biomedical. Termasuk database kardiopulmonari, saraf, dan sinyal-sinyal biomedical. Data-data tersebut merupakan data yang didapatkan mulai dari pasien yang sehat sampai dari pasien dengan berbagai kondisi, seperti aritmia, gagal jantung, apnea tidur, penyakit saraf, dan penuaan. Koleksi ini termasuk data dari berbagai studi, yaitu yang dikembangkan dan disumbangkan oleh para anggota komunitas riset.

PhysioToolkit merupakan sebuah *library* aplikasi *open source* untuk pemrosesan dan analisis sinyal fisiologis, pendeteksian kejadian fisiologis menggunakan metode lama atau baru, penampilan dan karakterisasi sinyal secara interaktif, pembuatan database baru, simulasi, dan proses analisis lainnya. PhysioToolkit dapat diunduh dalam bentuk *source code* atau aplikasi yang terkompilasi untuk Linux, Solaris, atau MS-DOS.

2.4.6 Confusion Matrix [13]

Confusion Matrix berisi informasi tentang klasifikasi asli dan hasil sebuah prediksi klasifikasi dari sebuah sistem klasifikasi. Performa dari sistem klasifikasi seperti proyek akhir ini biasanya dievaluasi menggunakan data yang ada pada matrix [13].

		Prediksi	
		Positif	Negatif
Asli	Positif	TP	FN
	Negatif	FP	TN

Tabel 2.2 *Tabel confusion matrix*

Tabel 2.2 merupakan tabel yang menggambarkan *confusion matrix* yang digunakan untuk menghitung performansi dari sebuah aplikasi klasifikasi. Di dalam matrix terdapat 4 buah kondisi, yaitu:

1. True Positive (TP), adalah kondisi saat output program menyatakan positif anomali dan pada kondisi sebenarnya juga positif anomali.
2. False Positive (FP), adalah kondisi saat output program menyatakan positif anomali tetapi pada kondisi sebenarnya negatif anomali.
3. True Negative (TN), adalah kondisi saat output program menyatakan negatif anomali dan pada kondisi sebenarnya juga negatif anomali.

4. False Negative (FN), adalah kondisi saat output program menyatakan negatif anomali tetapi pada kondisi sebenarnya positif anomali.

Dengan menggunakan 4 kondisi diatas maka *Detection Rate* (DR) dapat dihitung dengan rumus persamaan 10:

$$DR = \frac{TP}{TP + FN} \quad (10)$$

Dan False Posotive Rate (FPR) dapat dihitung dengan menggunakan rumus persamaan 11:

$$FPR = \frac{FP}{FP + TN} \quad (11)$$

Semakin tinggi nilai DR dan rendah nilai FPR, maka dapat dikatakan bahwa program semakin baik karena program dapat mendeteksi adanya anomali dan keadaan yang benar.

BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

Dalam pembuatan suatu sistem harus dilakukan perencanaan dan perancangan sistem yang sesuai dengan tujuan serta permasalahan yang dihadapi. Bab ini akan membahas secara rinci mengenai perencanaan dan pembuatan sistem yang akan dibuat dalam proyek akhir ini.

3.1 DESKRIPSI UMUM

Pada proyek akhir yang saya kerjakan yaitu mengenai pembuatan aplikasi yang digunakan untuk mendeteksi adanya data anomali pada sebuah dataset yang didapatkan dari hasil pengukuran sensor-sensor kesehatan (WBAN). Berikut adalah gambaran umum desain proyek akhir saya.

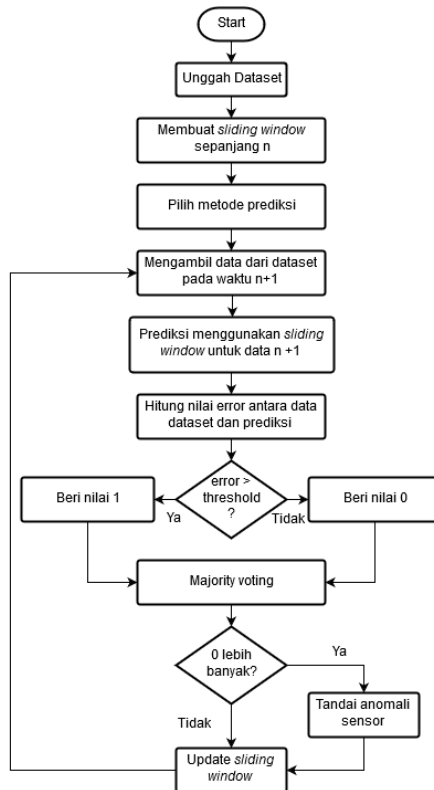


***Gambar 3.1** Gambaran umum sistem*

Gambar 3.1 merupakan ilustrasi gambaran sistem yang akan dibuat. Dataset merupakan sebuah file berformat .arff yang berisi hasil pengukuran pada beberapa parameter kesehatan pada tubuh manusia. Terdapat 5 parameter kesehatan yang dapat dilakukan pendeteksian anomali, yaitu tekanan darah, detak jantung, denyut nadi, pernapasan, dan kadar oksigen dalam darah. Format dataset harus disesuaikan dengan aplikasi yang dibuat, akan dijelaskan pada bagian berikutnya. Dataset ini diunggah ke dalam aplikasi pendeteksian anomali yang telah dibangun. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan dapat dijalankan pada platform desktop. Di dalam aplikasi, dataset akan dibuka dan dilakukan analisa pada setiap baris data yang ada di dalam dataset.

Pendeteksian anomali pada dataset ini menggunakan gabungan metode-metode yang dirangkai menjadi sebuah algoritma baru, diagram alur dari pendeteksian anomali ini akan dijelaskan pada bagian berikutnya. Setelah aplikasi melakukan pendeteksian anomali, pada aplikasi akan ditampilkan baris data mana saja yang merupakan sebuah anomali pada dataset yang telah diunggah. Selain menampilkannya pada aplikasi, aplikasi ini juga dapat melakukan pembuatan laporan berupa grafik yang menggambarkan parameter-parameter yang ada pada dataset dan memberikan tanda pada bagian-bagian yang merupakan anomali.

3.2 DIAGRAM ALUR PENDETEKSIAN ANOMALI



Gambar 3.2 Diagram alur pendeteksian anomali

Gambar 3.2 menunjukkan diagram alir pendeteksian anomali yang digunakan dalam aplikasi. Pendekatan yang digunakan akan menggunakan 2 algoritma: Sequential Minimal Optimization Regression (SMOReg) dan Gaussian untuk melakukan prediksi nilai, Dynamic Threshold (DT) untuk melakukan perhitungan error, dan Majority Voting (MV) untuk menentukan apakah data tersebut anomali atau tidak. Metode ini merupakan metode yang dikembangkan dalam penelitian [14] sebelumnya untuk tujuan yang hampir sama.

Gambar 3.2 menunjukkan alur dari aplikasi untuk melakukan pendeteksian anomali menggunakan 3 algoritma sebelumnya. Pengguna melangunggah sebuah dataset yang akan dilakukan proses pendeteksian anomali. Selanjutnya aplikasi akan membuat sebuah *sliding window* sepanjang yang diinginkan oleh pengguna. *Sliding window* ini merupakan *subset* dari dataset yang diunggah sebelumnya yang digunakan untuk pembuatan model prediksi dan perhitungan DT. Panjang dari *sliding window* ini dapat diatur dalam aplikasi. Setelah itu aplikasi akan membuat sebuah metode prediksi sesuai yang telah dipilih oleh pengguna pada aplikasi. Selanjutnya adalah mendapatkan nilai parameter-parameter kesehatan dari dataset. Setelah itu aplikasi akan melakukan prediksi nilai selanjutnya dengan menggunakan algoritma yang dipilih sebelumnya dengan menggunakan *sliding window* sebagai data training. Setelah data dari dataset dan nilai prediksi telah didapatkan, maka kedua nilai tersebut dibandingkan dan diambil selisihnya atau nilai errornya. Setelah mendapatkan nilai error dari masing-masing parameter, selanjutnya adalah membandingkan apakah nilai error tersebut melebihi nilai toleransi error/*threshold*. Nilai toleransi ini didapatkan dengan menghitung standar deviasi dari beberapa pengukuran sebelumnya yang akan terus berganti seiring dengan data-data yang diambil dari dataset dikalikan dengan sebuah konstanta yang akan dijelaskan pada bagian selanjutnya. Apabila nilai error melebihi *threshold*, maka parameter tersebut akan diberi nilai 1 yang berarti anomali dan apabila tidak melebihi, maka akan diberi nilai 0 yang berarti normal untuk parameter tersebut. Setelah proses pengecekan nilai error selesai, maka akan dilakukan proses *majority voting*. Proses *majority voting* ini digunakan untuk membedakan keadaan

anomali dengan keadaan medis yang benar. Proses tersebut dilakukan dengan melakukan perbandingan jumlah parameter dengan nilai 1 dan parameter dengan nilai 0. Apabila parameter dengan nilai 1 lebih banyak, maka data-data tersebut akan dianggap sebagai keadaan medis. Sebaliknya, apabila parameter dengan nilai 0 lebih banyak, maka data-data tersebut dianggap anomali pada sensor dan akan memberikan alarm terjadinya anomali tersebut. Setelah itu data akan disimpan pada variabel. Selanjutnya adalah melakukan pembaruan pada *sliding window* dan kembali menuju proses pengambilan data dari dataset. Proses tersebut akan terus berlanjut sampai semua data yang ada pada dataset telah dilakukan pendeteksian anomali.

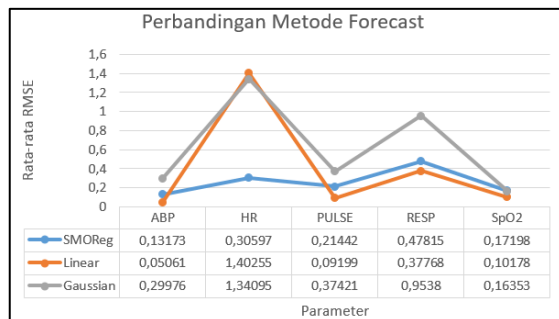
3.3 PERBANDINGAN METODE PREDIKSI

Untuk melakukan pendeteksian terhadap anomali, kami memerlukan sebuah metode prediksi yang dapat digunakan dalam melakukan prediksi terhadap data *timeseries*. Terdapat beberapa penelitian yang menggunakan sebuah metode prediksi untuk data *time series* [15][16]. Penelitian diatas menggunakan 2 metode yang berbeda dalam melakukan prediksi. Penelitian [15] menggunakan metode Gaussian dalam melakukan proses prediksi, sedangkan [16] menggunakan metode SMOReg dalam melakukan prediksi terhadap datanya.

Kami melakukan beberapa kali uji coba pada 3 metode prediksi pada WEKA. Metode yang kami uji coba adalah Linear Regression, SMO Regression dan Gaussian. Kami menggunakan dataset MIMICDB 221 yang didapat dari PhysioNet[12]. Uji coba yang kami lakukan adalah dengan mencoba menjalankan metode-metode tersebut pada dataset dengan merubah jumlah data yang digunakan. Jumlah data yang kami gunakan kami rubah sesuai kelipatan 10 dan berkisar antara 10-100 jumlah data. Jadi kami melakukan 10 kali percobaan pada 3 metode tersebut dengan dataset yang memiliki 5 buah parameter (ABP, HR, PULSE, RESP, SpO2). Pada setiap kali percobaan dilakukan perhitungan nilai error dengan menggunakan rumus *Root Mean Squared Error*

(RMSE) yang akan keluar saat proses prediksi pada WEKA selesai. Setelah semua percobaan selesai, kami menghitung rata-rata nilai RMSE setiap percobaan pada setiap variabel.

Gambar 3.3 menunjukkan grafik nilai RMSE pada setiap parameter. Dapat dilihat pada gambar 8, terdapat 3 buah garis yang mewakili metode yang diujicobakan. Pada parameter HR terdapat anomali yang sengaja diujicobakan untuk melihat apakah metode dapat digunakan dalam keadaan anomali. Garis metode SMOReg selalu konstan berada di bawah garis 0,6 yang menunjukkan bahwa metode tersebut dapat berjalan baik walaupun terdapat anomali yang terdapat pada parameter HR. Sedangkan garis metode Linear dan Gaussian memiliki nilai yang tinggi pada parameter HR karena terdapat anomali pada parameter tersebut. Metode Linear memiliki tingkat akurasi yang tinggi, tetapi waktu pemrosesan menggunakan metode ini terlalu lama sehingga kami tidak memasukkan metode ini ke dalam aplikasi.



Gambar 3.3 Grafik RMSE hasil uji coba metode prediksi

3.4 KONFIGURASI DATASET

Aplikasi yang kami bangun pada proyek akhir ini adalah sebuah aplikasi yang menggunakan weka sebagai inti. Dalam penggunaannya, weka hanya dapat melakukan operasi-operasinya pada dataset dengan format tertentu. Dataset yang digunakan dalam weka haruslah memiliki ekstensi .arff yang memiliki struktur khusus dalam filenya.

Dataset arff harus memiliki struktur sendiri dalam penulisan data didalamnya. Data arff memiliki 3 bagian utama, yaitu pendeklarasian nama relasi/data, pendeklarasian variable beserta tipe-tipenya, dan yang terakhir adalah data-data itu sendiri. masing-masing bagian memiliki sintaks yang berbeda-beda. Gambar 3.4 di bawah ini merupakan tampilan dari contoh dataset dengan format .arff yang dapat digunakan.

Dapat dilihat pada gambar 3.4 bahwa seperti yang telah dijeaskan sebelumnya, dataset arff dibagi menjadi beberapa bagian. Baris 1 merupakan pendeklarasian nama relasi atau dataset, yaitu “data221”. Pendeklarasiannya menggunakan sintaks “@relation [nama]”. Setelah itu harus ada bagian untuk melakukan pendeklarasian nama dan tipe variabel yang ada pada baris 3 sampai 8. Pendeklarasiannya dituliskan menggunakan sintaks “@attribute [nama] [tipe]”. Bagian terakhir merupakan bagian dimana data dari dataset itu dituliskan. Pada baris 10 dapat dilihat bahwa harus ada sintaks “@data” yang menandakan bahwa semua baris dibawahnya adalah data. Penulisan datanya iu sendiri dituliskan tiap baris dan setiap variabel dipisahkan menggunakan tanda koma.

```
1 @relation data221
2
3 @attribute time date 'HH:mm:ss dd/MM/yyyy'
4 @attribute abpmean numeric
5 @attribute HR numeric
6 @attribute PULSE numeric
7 @attribute RESP numeric
8 @attribute SpO2 numeric
9
10 @data
11 '12:05:02 18/05/1995',122,77,76,26,99
12 '12:05:03 18/05/1995',121,76,74,26,99
13 '12:05:04 18/05/1995',121,76,74,26,99
14 '12:05:05 18/05/1995',121,75,74,26,99
```

Gambar 3.4 Contoh dataset mimicdb dalam format arff

3.5 PERANCANGAN APLIKASI

Pada bagian ini akan dijelaskan bagaimana kami membangun aplikasi untuk melakukan pendeteksian anomali. Akan dijelaskan mulai proses pemasangan library weka, pembuatan antar-muka aplikasi, konfigurasi dari metode prediksi yang digunakan, sampai proses pembuatan report berupa grafik.

3.5.1 Pemasangan Library WEKA

Hal pertama yang dilakukan dalam pembuatan aplikasi ini adalah tentunya pembuatan proyek baru pada IDE Java, yang pada proyek ini kami menggunakan NetBeans IDE 8.0.1. Lalu memasang library weka ke dalam proyek tersebut. Berikut adalah langkah-langkah pemasangan library weka pada proyek mulai dari proses mengunduh library tersebut:

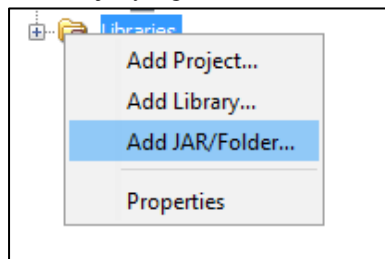
1. Unduh file weka.jar pada halaman web java2s.com dan pastikan memiliki class-class seperti gambar 3.5.

<http://www.java2s.com/Code/Jar/w/Downloadwekajar.htm>.

```
import weka.core.Instances;  
import weka.classifiers.evaluation.NumericPrediction;  
import weka.classifiers.timeseries.WekaForecaster;  
import weka.classifiers.functions.SMOreg;  
import weka.classifiers.functions.GaussianProcesses;  
import weka.classifiers.functions.LinearRegression;
```

Gambar 3.5 Class weka

2. Masukkan library weka pada proyek dengan menambahkannya pada folder Libraries seperti gambar 3.6. Lalu pilih file weka.jar yang telah diunduh.



Gambar 3.6 Menambahkan library

3. Import class-class library weka yang dibutuhkan pada aplikasi seperti gambar 3.7. Penjelasan dari class-class ini akan dijelaskan pada bagian 3.5.3.

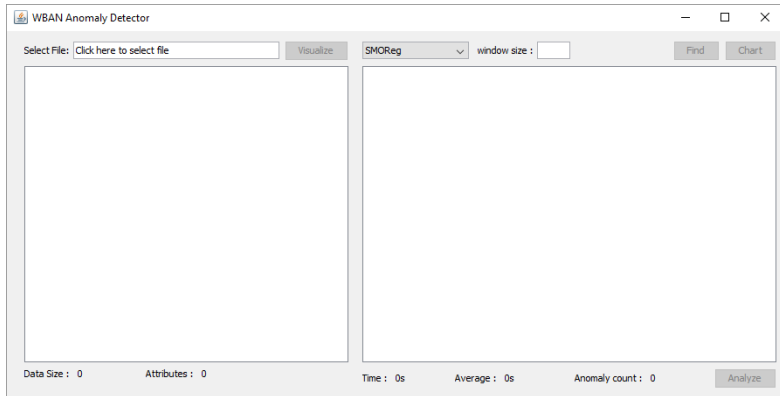
```
import weka.core.Instances;  
import weka.classifiers.evaluation.NumericPrediction;  
import weka.classifiers.timeseries.WekaForecaster;  
import weka.classifiers.functions.SMOreg;  
import weka.classifiers.functions.GaussianProcesses;  
import weka.classifiers.functions.LinearRegression;
```

Gambar 3.7 Import class-class weka

3.5.2 Perancangan Antar-Muka

Hal yang paling penting dari sebuah aplikasi adalah tampilan antar-muka. Tampilan antar-muka merupakan hal yang dapat dilihat oleh pengguna. Kenyamanan pengguna ditentukan dengan bagaimana bentuk maupun tata-letak dari fitur-fitur yang ada pada aplikasi yang mereka gunakan. Pada proyek ini, kami membuat sebuah aplikasi dengan tampilan antar-muka menggunakan tema *single-page*. Jadi, semua fitur dari aplikasi telah ada pada satu antar-muka tanpa harus berpindah-pindah.

Pada bagian ini akan dijelaskan bagaimana kami membangun tampilan antar-muka dari aplikasi ini. Aplikasi yang kami bangun menggunakan JFrame sebagai dasar dari tampilan, menggunakan tombol-tombol yang memiliki fungsi berbeda untuk menjalankan perintah yang akan dijelaskan selanjutnya, text area untuk melakukan pengunggahan dataset dan menampilkan hasil pendeteksian, serta label-label untuk memberikan penjelasan lebih lanjut pada aplikasi. Gambar 3.8 di bawah ini merupakan tampilan desain antar-muka dari aplikasi. Aplikasi dibagi menjadi 2 bagian besar, panel kiri dan kanan. Bagian panel kiri dikhususkan untuk semua proses input dan bagian panel kanan untuk semua proses output.

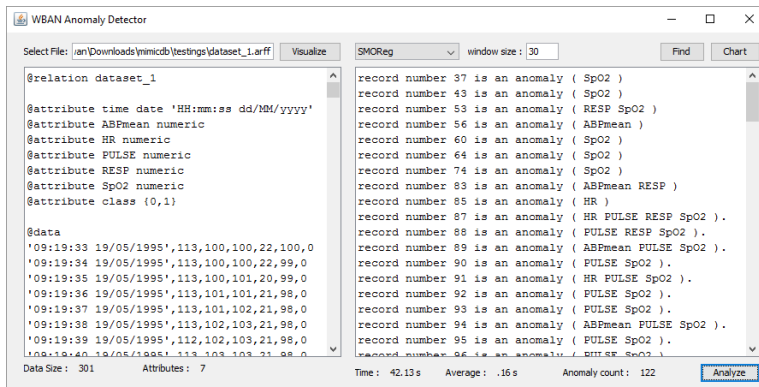


Gambar 3.8 Desain antar-muka

Pada panel kiri terdapat sebuah text-field yang digunakan untuk mengunggah file dataset ke dalam aplikasi. Lalu dibawah *text-field* tersebut terdapat sebuah *text-area* yang digunakan untuk menampilkan isi dari dataset apabila telah ada dataset yang diunggah menggunakan fitur sebelumnya. Dan pada bagian bawah terdapat 2 buah label yang menunjukkan ukuran dan jumlah attribut dari dataset yang telah diunggah.

Pada panel kanan bagian atas terdapat sebuah dropdown menu untuk memilih metode prediksi yang akan digunakan, sebuah *text-field* untuk memasukkan ukuran *sliding window*, 2 buah tombol, yaitu tombol “Analyze” dan tombol “Chart”. Tombol Analyze ini merupakan bagian utama dalam proyek akhir ini, yaitu tombol yang digunakan untuk melakukan pendeteksian anomali pada dataset yang diunggah. Hasil dari pendeteksian anomali ini berupa daftar baris mana saja yang merupakan anomali pada dataset yang diunggah. Tombol Chart digunakan untuk membuat dan membuka sebuah halaman html yang berisi grafik dari parameter-parameter fisiologis dataset dan area-area mana saja yang terjadi anomali pada dataset. Dan pada bagian bawah terdapat 3 buah label yang akan berisi nilai apabila proses pendeteksian anomali telah selesai. Nilai pada label-label tersebut menunjukkan waktu pendeteksian, rata-

rata pendeteksian tiap data, dan jumlah data yang dianggap sebagai anomali. Gambar 3.9 merupakan tampilan aplikasi setelah menjalankan pendeteksian anomaly pada sebuah dataset.



Gambar 3.9 Antar-muka setelah proses pendeteksian

3.5.3 Konfigurasi Metode Prediksi

Prediksi adalah bagian paling penting dalam proses pendeteksian anomali pada aplikasi ini. Permalan digunakan untuk memberikan nilai pembanding untuk data parameter yang didapatkan dari dataset. Maka dari itu, konfigurasi dari metode prediksi dalam aplikasi ini sangatlah penting untuk diperhatikan agar mendapat hasil yang tepat. Pada bagian ini akan dijelaskan bagaimana melakukan konfigurasi metode prediksi mulai dari pemasangan dataset, pembuatan sliding window, pembuatan lag maker dan cara menjalankan prediksinya itu sendiri.

Penulis akan menjelaskan bagaimana cara melakukan konfigurasi prediksi dengan mencontohkannya menggunakan aplikasi weka dan juga menjelaskan bagaimana cara menuliskan kode untuk aplikasi. Pada setiap langkahnya penulis akan memberikan contoh pada aplikasi weka terlebih dahulu dan menjelaskan koding untuk langkah tersebut. Berikut adalah langkah-langkahnya.

1. Import class weka pada project yang telah dibuat seperti contoh sebelumnya.

```
import weka.core.Instances;  
import weka.classifiers.evaluation.NumericPrediction;  
import weka.classifiers.timeseries.WekaForecaster;  
import weka.classifiers.functions.SMOreg;  
import weka.classifiers.functions.GaussianProcesses;  
import weka.classifiers.functions.LinearRegression;
```

Gambar 3.10 Import class-class weka

Dalam aplikasi pendeteksian anomali menggunakan bahasa pemrograman java yang memanfaatkan weka, diperlukan proses *import* class-class dari weka seperti pada gambar 3.9. pada gambar 3.10 di atas dapat dilihat bahwa diperlukan 6 buah class untuk aplikasi pendeteksian anomali. Berikut adalah penjelasan dan kegunaan masing-masing class yang di-*import* dalam aplikasi ini:

- a. Class weka.core.Instances

Merupakan sebuah class yang digunakan untuk menangani dataset. Class ini merupakan turunan dari class java.lang.Object. Dataset yang diunggah pada aplikasi akan dibaca menggunakan reader, dan reader tersebut akan digunakan sebagai parameter dalam pembuatan obyek Instances. Setelah dataset masuk ke dalam Instances ini, barulah dataset dapat diolah menggunakan *method* yang ada dalam weka.

Beberapa fungsi penting yang digunakan dalam aplikasi ini dapat dilihat pada tabel 3.1.

Tipe	Fungsi <i>Method</i>
void	add(Instance instance) Menambahkan satu instance pada akhir set.
Attribute	attribute(int index)

	Menghasilkan sebuah attribute pada index yang ditentukan.
AttributeStats	attributeStats(int index) Menghitung statistik nilai yang ada pada dataset untuk attribute tertentu.
void	delete(int index) Menghapus sebuah instance pada index tertentu.

Tabel 3.1 Method pada class Instances

- b. Class `weka.classifiers.timeseries.WekaForecaster`.
Merupakan class yang mengimplementasikan prediksi *timeseries* yang menggunakan skema regresi Weka. Class ini membutuhkan class `TSLagMaker` yang digunakan untuk menangani semua pembuatan attribut.
Tabel 3.2 merupakan daftar *method* dari class ini yang digunakan dalam aplikasi.

Tipe	Fungsi <i>Method</i>
void	buildForecaster(Instances insts, java.io.PrintStream... progress) Membuat sebuah model prediksi baru menggunakan data training yang ditentukan.
Attribute	attribute(int index) Menghasilkan sebuah attribute pada index yang ditentukan.
java.util.List<java.util.List<Numeric Prediction>>	forecast(int numSteps, java.io.PrintStream... progress)

	Menghasilkan hasil prediksi untuk <i>field</i> yang ditentukan sebanyak yang diinginkan.
TSLagMaker	getTSLagMaker() Mendapatkan TSLagMaker yang digunakan. TSLagMaker adalah sebuah <i>class</i> yang digunakan untuk membuat <i>lag variable</i> pada proses prediksi.
void	primeForecaster(Instances ins) Memberikan model dengan data historis untuk menghasilkan sebuah prediksi.
void	setBaseForecaster(Classifier f) Menentukan skema regresi Weka sebagai dasar yang digunakan.
void	setFieldsToForecast(String fields) Menentukan nama-nama atribut data yang dilakukan prediksi.

Tabel 3.2 Method pada class WekaForecaster

- c. Class weka.classifiers.functions.SMOReg.
Class SMOReg mengimplementasikan *support vector regression* (SVR) untuk regresi. Merupakan class yang berisi metode prediksi yang digunakan dalam fitur utama aplikasi proyek akhir ini.
- d. Class weka.classifiers.functions.GaussianProcesses.
Class SMOReg mengimplementasikan proses regresi menggunakan gaussian. Merupakan class yang berisi

metode prediksi yang digunakan dalam fitur utama aplikasi proyek akhir ini.

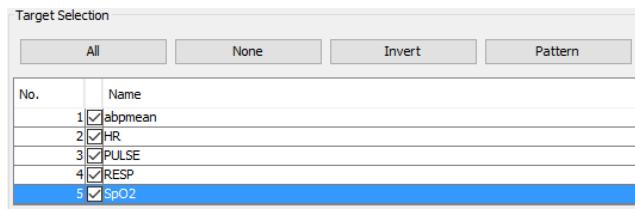
- e. `Class weka.classifiers.evaluation.NumericPredictions.`
Class ini digunakan untuk tipe data dari List yang digunakan untuk menampung hasil prediksi yang telah dilakukan oleh forecaster.

2. Menentukan dataset yang digunakan

Setelah memastikan semua class yang diperlukan sudah terpasang, selanjutnya adalah membuat sebuah Instances yang merupakan dataset yang akan dilakukan prediksi. Dalam aplikasi weka, pengguna dapat menggunakan tombol “Open File” untuk memilih file dataset yang digunakan. Sedangkan dalam aplikasi, kami memanfaatkan class Instances yang telah dimasukkan sebelumnya untuk membuat dataset dari file arff.

3. Memilih atribut yang akan dilakukan prediksi

Saat file sudah terpasang atau terunggah, selanjutnya adalah melakukan pembuatan sebuah obyek `WekaForecaster` dalam kode aplikasi. Sedangkan dalam aplikasi Weka, hal itu tidak perlu dilakukan. Setelah melakukan pembuatan obyek `WekaForecaster`, selanjutnya adalah memilih atribut mana saja yang akan dilakukan prediksi. Gambar 3.11 menunjukkan proses pemilihan atribut untuk prediksi yang dilakukan pada weka.



Gambar 3.11 Memilih atribut pada weka

Pada gambar 3.11, atribut yang dipilih untuk dilakukan prediksi adalah abpmean, HR, PULSE, RESP, dan SpO2. Pemilihannya dapat dilakukan dengan mencentang nama atribut yang diinginkan. Dalam kode, operasi ini dilakukan dengan menggunakan salah satu *method* yang ada pada class WekaForecaster. Kode 3.1 menunjukkan pembuatan obyek WekaForecaster dan penentuan atribut untuk proses prediksi.

```
1. //create new forecaster
2. WekaForecaster forecaster = new
   WekaForecaster();
3. //set the target
4. forecaster.setFieldsToForecast(String.join(",
   ", attributeNames);
```

Kode 3.1 Pembuatan forecaster dan pemilihan atribut

Pada kode 3.1 terdapat 2 buah perintah. Perintah pertama pada baris ke 2 digunakan untuk membuat forecaster baru, sedangkan baris ke 4 merupakan perintah untuk menentukan atribut mana saja yang akan dilakukan prediksi. Variabel *attributeNames* merupakan sebuah *array of string* yang berisi nama-nama atribut dan digabungkan menjadi sebuah string dengan tanda koma sebagai penghubung.

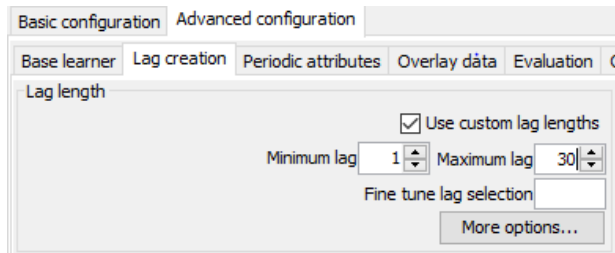
4. Membuat *Lag Variable*

Lag variables merupakan data-data terdahulu yang dijadikan sebagai data training dalam proses prediksi. Variabel ini juga akan dikatakan sebagai “*sliding window*”. Jadi variabel ini yang menentukan jumlah atau rentang data yang masih dapat dijadikan dasar melakukan operasi. Contohnya, apabila data yang dimiliki adalah sebuah data bulanan, lalu anda ingin memakai data 1 tahun sebelumnya, maka anda harus menginisialisasi nilai *lag variables* antara 1-12. Angka 1 menunjukkan bahwa variabel akan dibuat dari data dengan index -1 dari data yang akan diprediksi yang berarti dulan sebelumnya, dan angka 12 adalah maximum lag yang merupakan batas maksimum pembuatan variabel adalah waktu

-12 yang berarti 12 bulan sebelumnya. Apabila data yang anda miliki adalah data mingguan, maka *lag variables*-nya adalah 1-52.

Data yang kami gunakan dalam pendeteksian anomali ini adalah data tiap detik. Lalu kami ingin menggunakan data untuk prediksi kami sebanyak setengah menit data yang berarti *Lag Variables* kami mulai dari 1-30.

Dalam Weka, pembuatan variabel ini dilakukan pada panel “Lag Creation” pada bagian “Advance Configuration”. Dalam panel tersebut terdapat beberapa kotak yang bisa diisi dan yang dibutuhkan adalah kotak “minimum lag” dan “maximum lag”. Gambar 3.12 adalah contoh melakukan pengaturan *lag variabel* 1-30.



Gambar 3.12 Pengaturan *lag variables* pada weka

Dalam penulisan kode, pengaturan *lag variables* ini dilakukan dengan mengkonfigurasi obyek *WekaForecaster* yang menggunakan *method* pada *class* *TSLagMaker*. Kode 3.2 di bawah ini menunjukkan cara pengaturan *lag variabel* dengan kode.

```
1. forecaster.getTSLagMaker().setTimeStampField("time");  
2. forecaster.getTSLagMaker().setMinLag(1);  
3. forecaster.getTSLagMaker().setMaxLag(30);
```

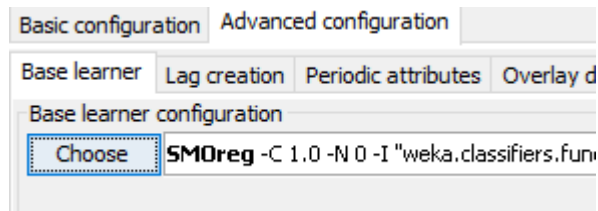
Kode 3.2 Pengaturan *lag variables* dengan kode

Pada kode 3.2 terdapat 3 buah baris yang berhubungan dengan pembuatan *lag variables*. Baris pertama digunakan untuk

menentukan atribut pada dataset yang digunakan sebagai penanda waktu. Baris kedua dan ketiga merupakan baris yang masing-masing berfungsi untuk mengatur *minimum* dan *maximum lag variables*.

5. Memilih metode prediksi.

Selanjutnya adalah memilih metode prediksi yang akan digunakan. Metode yang akan digunakan adalah metode SMOReg yang telah dijelaskan pada BAB II. Pada Weka, untuk melakukan pemilihan metode ini dapat dilakukan pada panel “Base learner” bagian “Advance configuration” dengan menekan tombol “Choose” dan memilih metode SMOReg. Gambar 3.13 adalah contoh pemilihan metode SMOReg pada aplikasi Weka.



Gambar 3.13 Memilih SMOReg pada Weka

Untuk melakukan pengaturan ini menggunakan kode pada aplikasi dapat dilakukan dengan menggunakan *method* pada obyek WekaForecaster. Kode 3.3 menunjukkan cara pemilihan metode dengan mengakses *method* `setBaseForecaster()` dan membuat obyek SMOReg sebagai parameternya. Pengaturannya hanya memerlukan satu baris perintah. Perintah ini akan berubah tergantung ada pilihan pengguna pada metode prediksi yang dipilih pada aplikasi.

```
1. forecaster.setBaseForecaster(new SMOReg());
```

Kode 3.3 Pemilihan metode dengan menggunakan kode

6. Melakukan prediksi.

Lima langkah diatas adalah hal-hal yang perlu dilakukan sebelum melakukan proses prediksi. Hal yang perlu dilakuakn selanjutnya adalah melakukan *build* pada model prediksi yang sudah dibangun sebelumnya. Dalam Weka, yang perlu dilakukan hanyalah tinggal menekan tombol “Start” untuk memulai prediksi. Dalam kode diperlukan beberapa langkah. Kode 3.4 menunjukkan perintah-perintah untuk menjalankan prediksi.

```
1. forecaster.buildForecaster(mimic,  
    System.out);  
2. forecaster.primeForecaster(mimic);  
3. List<List<NumericPrediction>> forecast =  
    forecaster.forecast(1, System.out);
```

Kode 3.4 Menjalankan prediksi

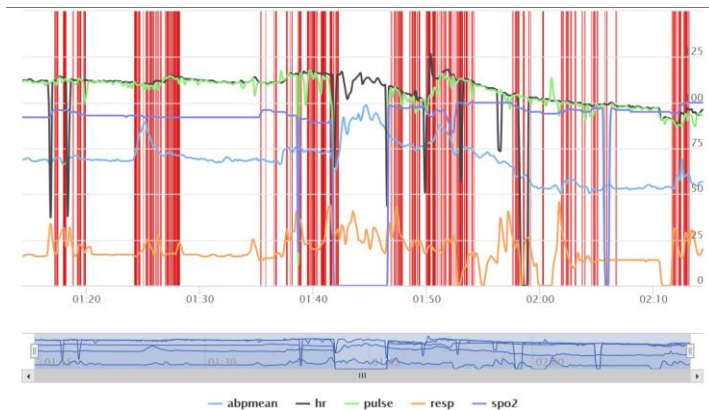
Pada kode 3.4 terdapat 3 baris kode. Baris pertama digunakan untuk membangun prediksi dengan data yang telah diberikan. Lalu pada baris ketiga dilakukan pembuatan sebuah variabel yang diisi dengan hasil prediksi dari *method* forecast() dengan parameter 1 yang menandakan jumlah data yang diprediksi. Kami menggunakan 1 karena hanya memerlukan 1 data prediksi setiap perulangan.

3.5.4 Pembuatan Grafik

Setelah semua proses pendeteksian anomali selesai, pada aplikasi akan ditampilkan daftar baris mana saja pada dataset yang merupakan anomali. Tampilan ini tentunya sulit dibaca oleh pengguna karena hanya berbentuk daftar tulisan. Untuk memudahkan pengguna melihat bagian mana saja pada dataset yang merupakan anomali, kami membuat sebuah grafik dengan file html untuk memudahkan pengguna. Ide dari grafik ini adalah menampilkan semua data parameter yang ada pada dataset pada satu grafik dan memberikan tanda pada bagian-bagian mana saja yang merupakan anomali dengan memberikan warna latar yang berbeda.

Gambar 3.14 merupakan tampilan grafik yang bagian anomalnya ditandai dengan warna latar merah.

Pada gambar 3.14 dapat dilihat bahwa terdapat 5 buah garis yang menunjukkan masing-masing nilai dari parameter dengan warna yang berbeda-beda. Lalu terdapat blok-blok merah pada grafik tersebut. Blok-blok merah itu adalah bagian-bagian dalam dataset yang menunjukkan adanya anomali. Lalu pada bagian bawah terdapat sebuah garis waktu yang dapat digunakan untuk mengatur rentang waktu yang ditampilkan dalam grafik. Grafik ini akan dibuat dan terbuka saat pengguna menekan tombol “Chart” pada aplikasi setelah proses analisa selesai.



Gambar 3.14 Grafik parameter

Penulis menggunakan Highstock sebagai library pembuat grafik. Dalam pembuatan sebuah grafik, diperlukan data-data dari dataset yang ada pada aplikasi java. Untuk mengirimkan data-data tersebut, maka dibuatlah beberapa file JSON untuk mempermudah. Aplikasi akan membuat file-file JSON sebanyak parameter dari dataset, yaitu 5 (ABPmean, HR, PULSE, Respiration, dan SpO2) dan sebuah file JSON untuk data area yang terdapat anomali.

Setelah pembuatan file-file JSON yang berisi data-data dari aplikasi pendeteksi anomali selesai dibuat, aplikasi secara langsung akan membuka sebuah file HTML yang didalamnya terdapat baris-baris kode untuk membaca data dari file-file JSON sebelumnya dan menampilkannya dalam bentuk grafik. Fitur grafik ini diharapkan dapat membantu pengguna dalam mengamati adanya anomali yang ada pada dataset.

3.5.5 Dynamic Threshold

Keadaan fisiologis pada seseorang dapat berubah-ubah berdasarkan beberapa faktor seperti umur, keadaan fisik, kondisi psikis. Maka dari itu, sebuah batas *error/threshold* yang statis tidak bisa digunakan sebagai batas error yang tepat pada setiap orang. Diperlukan sebuah kemampuan dalam aplikasi untuk menyesuaikan batas nilai error untuk setiap orang pada setiap waktunya. Kami menggunakan sebuah batas nilai error dinamis atau *dynamic threshold* (DT). DT dihitung berdasarkan keadaan terbaru dari seorang pasien. Dengan menggunakan nilai standar deviasi dari beberapa data fisiologis pasien terbaru yang dikalikan dengan sebuah konstanta sebagai DT, kami dapat memiliki sebuah batas nilai error yang dapat menyesuaikan dengan keadaan pasien. Kami menggunakan sebuah konstanta seperti yang telah dilakukan pada penelitian sebelumnya [6] yang juga membahas mengenai penggunaan *threshold* yang dinamis untuk data kesehatan. Pada penelitian tersebut, konstanta yang digunakan bernilai 1.95. Data yang digunakan untuk perhitungan DT akan berubah-ubah seiring dengan data baru yang masuk ke dalam sistem. *Dynamic threshold* ini dibuat sebanyak jumlah parameter pada dataset yang dideteksi.

BAB IV

IMPLEMENTASI DAN ANALISA

Pada bab ini akan dibahas tentang pengimplemantasian sistem pada studi kasus, pengujian, dan analisa proyek untuk melihat apakah proyek yang telah dikerjakan/dibuat telah sesuai dengan yang telah direncanakan pada awal penegrjaan. Pengujian dilakukan dengan cara melakukan percobaan secara berulang-ulang. Setelah melakukan percobaan, maka selanjutnya melakukan perhitungan tentang ketepatan pendeteksian anomali oleh sistem. Dengan adanya pengujian ini maka dapat diketahui adanya kekurangan yang ada pada proyek ini. Sehingga dapat dilakukan perbaikan jika terjadi kesalahan nantinya.

4.1 BAHAN DAN ALAT

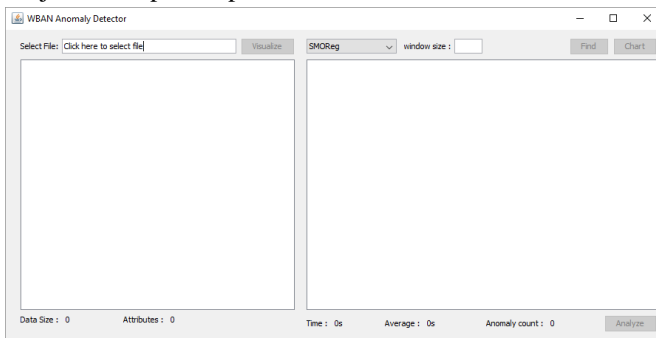
Bahan dan alat yang digunakan untuk pengujian dalam proyek akhir ini dibagi menjadi 3, yaitu perangkat keras, perangkat lunak atau aplikasi dan dataset yang digunakan. Berikut adalah spesifikasi bahan dan alat digunakan dalam pengujian proyek akhir ini:

- a. Perangkat keras (*hardware*)
Laptop ASUS A46 CM, dengan spesifikasi:
 - o Processor: Intel Core i5-3371 CPU @ 1.70GHz
 - o Memory RAM: 8.00 GB
 - o System: 64-bit
- b. Perangkat lunak (*software*)
 - Windows 10 64-bit.
 - Mozilla Firefox.
 - NetBeans IDE 8.0.1.
 - Library Weka
- c. Dataset
 - MIMICDB PhysioNet[12]

4.2 UJI COBA APLIKASI

Pada tahap ini, akan dilakukan proses uji coba terhadap semua fitur yang ada pada aplikasi. Pada aplikasi ini terdapat beberapa fungsi yang penting, seperti melakukan pengunggahan file, menampilkan file yang diunggah, melakukan pendeteksian dan menampilkannya, membuat grafik dari nilai-nilai parameter kesehatan dari dataset dan menampilkan kapan terjadi anomali, serta menghitung nilai ketepatan pendeteksian. Berikut adalah proses uji coba pengoperasian aplikasi dan uji coba fungsi-fungsi yang telah disebutkan sebelumnya.

1. Menjalankan aplikasi pendeteksian.

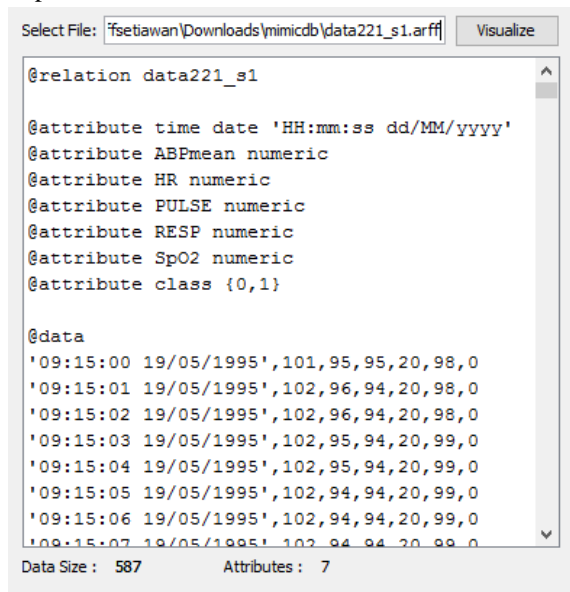


Gambar 4.1 Tampilan awal aplikasi

Gambar 4.1 merupakan tampilan awal dari aplikasi pendeteksian anomali yang dibuat dalam proyek akhir ini. Secara garis besar, aplikasi dibagi menjadi 2 panel, kiri dan kanan. Panel sebelah kiri dikhususkan sebagai tempat untuk menampilkan data-data tentang dataset yang diunggah, seperti unggah dataset, menampilkan isi dataset, menampilkan ukuran data dan atribut serta membuat grafik dari dataset yang diunggah. Panel kanan digunakan untuk operasi, seperti memilih metode prediksi, memasukkan ukuran *sliding window*, melakukan dan menampilkan pendeteksian serta menampilkan data-data tentang waktu pendeteksian yang telah dilakukan. Lalu dibagian bawah terdapat sebuah tombol yang digunakan untuk melakukan analisa pada pendeteksian yang telah dilakukan.

2. Melakukan pengunggahan dataset pada aplikasi.

Setelah menjalankan aplikasi, hal selanjutnya adalah melakukan pengunggahan dataset yang akan dianalisa. Untuk melakukannya, pengguna dapat mengakses *text field* dengan label “Select File”. Setelah itu akan muncul sebuah jendela baru yang digunakan untuk memilih dataset. Setelah dataset dipilih, aplikasi akan menampilkan sisi dari dataset tersebut ke dalam *text area* pada panel sebelah kiri. Gambar 4.2 menunjukkan tampilan aplikasi setelah pengguna mengunggah dataset ke dalam aplikasi.



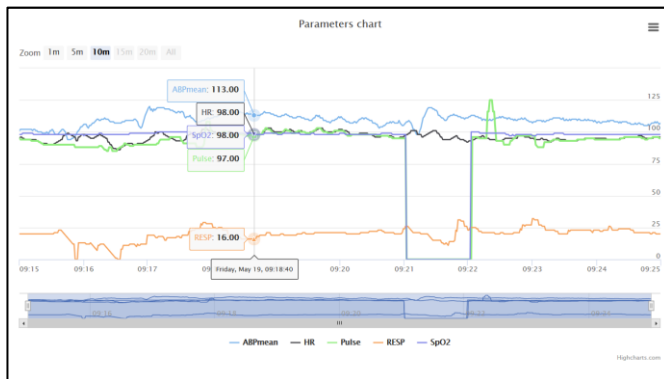
Gambar 4.2 Aplikasi menampilkan isi dataset

Dapat dilihat pada gambar 4.2 adalah tampilan aplikasi pada saat dataset telah diunggah. Isi dari *text field* pada bagian atas diganti menjadi direktori letak dari dataset pada komputer. Tombol “Visualize” yang sebelumnya tidak aktif menjadi aktif dan bisa ditekan. Lalu *text area* diisi dengan isi dari dataset yang telah

diunggah. Pada bagian bawah terdapat 2 label yang menunjukkan ukuran dari dataset dan jumlah atributnya.

3. Membuat grafik dari dataset.

Setelah pengguna mengunggah dataset ke dalam aplikasi, tombol “Visualize” yang ada pada samping *text field* akan menjadi aktif. Tombol ini digunakan untuk membuat sebuah grafik yang berisi data dari dataset yang telah diunggah. Saat pengguna menekan tombol ini, aplikasi akan membuat beberapa file berformat .json yang nantinya akan digunakan untuk menyimpan data-data yang akan dibaca oleh grafik. Grafik yang akan dibuat merupakan sebuah file .html yang bisa dibuka menggunakan *browser* pada komputer. Gambar 4.3 menunjukkan tampilan grafik dengan data yang diambil dari dataset yang diupload sebelumnya.

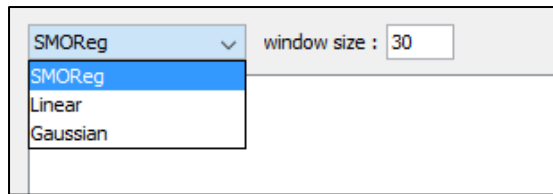


Gambar 4.3 Grafik dari dataset

Dapat dilihat pada gambar 4.3 diatas, data pada dataset yang ditampilkan pada gambar 4.2 dirubah menjadi bentuk grafik dalam file html yang dijlankan pada *browser*. Semua atribut pada dataset ditampilkan dalam grafik kecuali atribut *class*. Pada grafik tersebut terdapat fitur untuk melakukan perbesaran/*zoom*. Untuk memperbesar grafik dapat dilakukan dengan memilih salah satu pilihan di sebelah label “Zoom”, selain itu juga dapat melakukannya menggunakan fitur garis waktu di bagian bawah.

Data yang ditampilkan pada grafik dapat dipilih dengan cara memilih nama atribut pada bagian legenda. Apabila salah satu atribut dipilih, maka data dari atribut tersebut akan tampak atau hilang.

4. Mengatur metode prediksi dan ukuran *sliding window*.

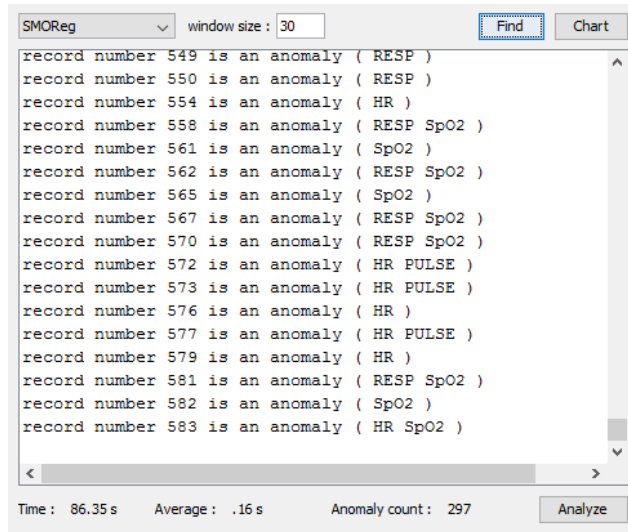


Gambar 4.4 Mengatur metode prediksi dan ukuran *sliding window*

Gambar 4.4 merupakan menu yang digunakan untuk melakukan pengaturan terhadap metode prediksi yang digunakan dalam pendeteksian anomali. Metode dapat dipilih dengan memilih salah satu pilihan yang ada pada *dropdown-list*. Selain memilih metode pemilihan, pengguna juga dapat melakukan pengaturan ukuran *sliding window* yang akan digunakan dalam proses pendeteksian anomali.

5. Melakukan proses pendeteksian.

Setelah melakukan pemilihan metode prediksi yang digunakan dan mengatur ukuran *sliding window* yang digunakan, maka aplikasi sudah siap digunakan untuk melakukan pendeteksian. Untuk melakukannya, pengguna dapat menekan tombol “Find” yang ada pada bagian atas panel sebelah kanan. Gambar 5.4 merupakan tampilan hasil pendeteksian anomaly oleh aplikasi ini.

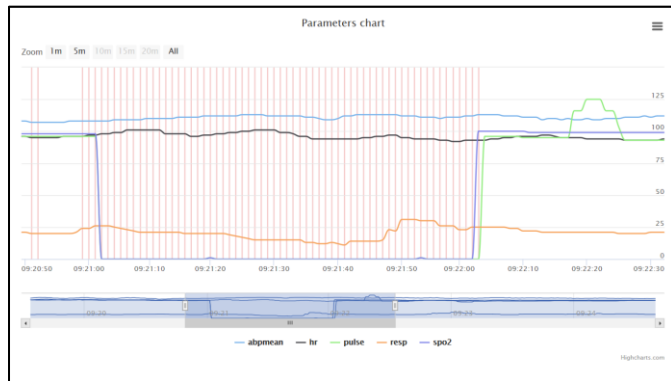


Gambar 4.5 Hasil pendeteksian

Pada gambar 4.5 dapat dilihat bahwa *text-area* yang sebelumnya kosong sudah terisi dengan beberapa baris tulisan. Baris-baris tulisan tersebut berisi informasi mengenai letak-letak terjadinya anomali pada dataset yang diunggah. Selain memberikan informasi mengenai letak anomali, terdapat informasi mengenai parameter mana yang menyebabkan terjadinya anomali.

Setelah proses pendeteksian selesai, pada bagian bawah akan terdapat beberapa informasi mengenai total waktu yang dibutuhkan untuk menganalisa dataset, rata-rata waktu yang digunakan untuk melakukan proses pendeteksian pada setiap baris data, dan jumlah anomali yang dapat dideteksi. Selain itu, tombol “Chart” pada bagian atas akan menjadi aktif dan dapat digunakan untuk memasukkan data-data anomali sebelumnya ke dalam grafik seperti pada bagian sebelumnya. Lalu tombol “Analyze” pada bagian bawah akan menjadi aktif dan dapat digunakan untuk melakukan analisa tentang tingkat akurasi dari pendeteksian anomali yang telah dilakukan sebelumnya.

6. Menampilkan nilai-nilai parameter dalam grafik.



Gambar 4.6 Grafik hasil pendeteksian

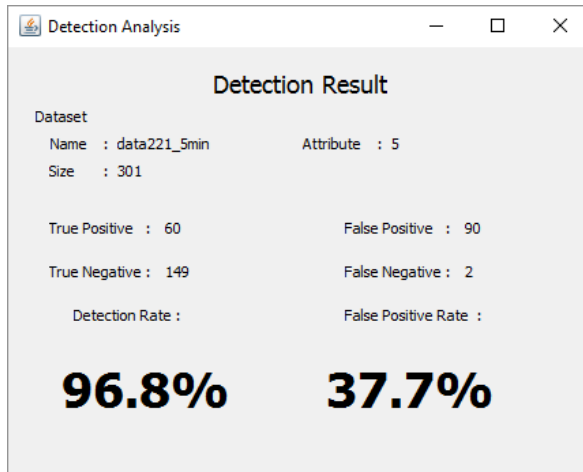
Gambar 4.6 menunjukkan *screenshot* dari fitur menampilkan hasil pendeteksian anomali pada sebuah grafik. Dapat dilihat bahwa grafik yang ditampilkan hampir sama dengan fitur grafik pada bagian sebelumnya, tetapi pada grafik ini terdapat beberapa garis merah pada sumbu x. Garis-garis merah ini menunjukkan letak-letak terjadinya anomali pada dataset yang diunggah. Grafik ini diharapkan dapat memudahkan pengguna dalam melihat data dan anomali.

7. Melakukan analisa terhadap hasil pendeteksian.

Setelah melakukan pendeteksian, selanjutnya adalah menganalisa tentang seberapa tepatkah pendeteksian yang telah dilakukan. Untuk melakukannya pengguna dapat menekan tombol “Analyze” yang ada pada bagian bawah panel sebelah kanan.

Gambar 4.7 menunjukkan *screenshot* dari fitur untuk menampilkan analisa dari hasil pendeteksian oleh aplikasi. Saat tombol “Analyze” ditekan, maka akan muncul window baru seperti pada gambar 4.7. Fitur analisa ini akan menghitung beberapa hal, seperti membandingkan hasil aplikasi dengan data

yang ada, perbandingan anomali yang terdeteksi dan yang tidak, serta pendeteksian yang salah. Hal inilah yang akan dianalisa secara manual dalam bagian berikutnya.



Gambar 4.7 Analisa hasil pendeteksian

4.3 DATASET UNTUK IMPLEMENTASI

Implementasi dan percobaan dilakukan pada dataset kesehatan yang sesungguhnya [12] menggunakan aplikasi pendeteksian yang telah dibuat. Pada bagian ini, penulis akan menyebutkan dan menjelaskan dataset-dataset mana saja yang akan digunakan dalam implementasi dari proyek akhir ini. Serta akan dijelaskan bagaimana penulis memilih dataset yang akan digunakan.

Pada *database* Multiple Intelligent Monitoring in Intensive Care (MIMICDB) PhysioNet [12] terdapat 121 dataset hasil pengukuran parameter-parameter kesehatan yang dilakukan pada pasien yang sesungguhnya. Setiap dataset memiliki parameter-parameter pengukuran yang berbeda-beda. Kami memilih dataset yang memiliki 5 parameter, yaitu tekanan darah (ABPmean), detak jantung (HR), denyut nadi (PULSE), pernapasan (Respiration), dan kadar oksigen dalam darah (SpO2). Selain memilih berdasarkan parameter, penulis juga memilih

dataset-dataset yang memiliki anomaly tanpa harus melakukan pembuatan anomaly.

Setiap dataset MIMICDB merupakan hasil pengukuran dengan rentang waktu yang berbeda, mulai dari 10 jam hingga 50 jam yang berarti dataset berisikan antara 36000 sampai 180000 *sample* pada setiap dataset. Dalam uji coba aplikasi pendeteksian anomaly ini, kami melakukan pemotongan terhadap dataset untuk menghemat waktu. Kami memotong dataset-dataset tersebut menjadi bagian yang berisi data selama 5 menit, atau sekitar 300 *sample* untuk setiap dataset yang digunakan. Kami membuat 4 buah dataset yang berisi data dan anomaly yang berbeda-beda pada setiap datasetnya yang diambil dari dataset MIMICDB. Berikut adalah data dari masing-masing dataset yang kami gunakan dalam percobaan.

Nama	Sumber	Waktu		Jumlah Data
		Dari	Sampai	
dataset_1	221n	19/05/1995 09:19:33	19/05/1995 09:24:40	301
dataset_2	054n	24/02/1995 21:02:00	24/02/1995 21:06:59	293
dataset_3	221n	19/05/1995 07:02:10	19/05/1995 07:07:10	304
dataset_4	414n	19/05/1995 13:39:57	19/05/1995 13:45:02	299

Tabel 4.1 Tabel keterangan dataset

Tabel 4.1 merupakan tabel yang menunjukkan dataset mana saja yang kami gunakan dalam proses percobaan untuk pengujian aplikasi. Dalam tabel terdapat keterangan sumber dari dataset, yaitu nomor dataset pada database MIMICDB. Kolom waktu menunjukkan waktu dimana dataset diambil dari dataset asli. Lalu terdapat kolom jumlah data yang menunjukkan jumlah *sample* data yang ada pada dataset tersebut.

4.4 HASIL IMPLEMENTASI DAN ANALISA

Terdapat 2 macam percobaan yang akan kami lakukan. Percobaan yang pertama adalah melakukan perubahan pada metode prediksi yang digunakan dalam aplikasi dan yang kedua adalah melakukan perubahan pada ukuran *sliding window* yang dipakai. Lalu juga ada 2 yang kami analisis dari hasil percobaan-percobaan yang kami lakukan, yaitu tingkat kebenaran dalam melakukan pendeteksian anomali dan waktu proses yang dibutuhkan.

Implementasi yang pertama adalah melakukan percobaan pada metode SMO Regression dengan mengganti-ganti ukuran *sliding window* yang digunakan dalam pendeteksian. Kami mencatat hasil analisa masing-masing percobaan seperti pada tabel 4.2.

Dataset	Akurasi pendeteksian					
	30		45		60	
	DR	FPR	DR	FPR	DR	FPR
dataset_1	100	4.6	100	1.7	100	1.7
dataset_2	100	4.5	100	2.2	100	2.2
dataset_3	100	4.9	100	3.7	100	4.5
dataset_4	95.7	10.9	100	8.0	100	7.0
Rata-Rata	98.92	6.225	100	3.9	100	3.8

Tabel 4.2 Tabel ketepatan metode SMOReg (dalam %)

Tabel 4.2 berisi hasil percobaan yang dilakukan dengan cara mengganti-ganti ukuran *sliding window* dan menggunakan metode SMOReg sebagai metode untuk melakukan prediksi. Sel tabel berwarna biru menunjukkan metode dengan hasil terbaik pada saat melakukan pengujian pada dataset tertentu. Dari percobaan yang telah dilakukan, dapat dilihat bahwa dengan memperbesar *sliding window* pada metode SMOReg dapat memperkecil rasio *false positive*. Sedangkan untuk rasio pendeteksian, nilainya selalu 100 % pada hampir padasemua percobaan. Sedangkan saat melakukan pendeteksian pada dataset nomor 4 yang

memiliki variasi data lebih banyak, tingkat pendeteksiannya tidak sampai 100% saat menggunakan ukuran 30 dan 100% saat menggunakan ukuran 45 & 60. Setelah melakukan rata-rata semua percobaan berdasarkan ukuran sliding window, dapat dikatakan bahwa ukuran *sliding window* 60 merupakan yang terbaik saat menggunakan metode SMOReg dengan rata-rata DR mencapai 100% dan rata-rata hanya FPR 3.8%.

Selanjutnya adalah melakukan percobaan pada metode Gaussian Process dengan mengganti-ganti ukuran sliding window yang digunakan dalam pendeteksian. Hasil percobaan dapat dilihat pada table 4.3.

Dataset	Akurasi pendeteksian					
	30		45		60	
	DR	FPR	DR	FPR	DR	FPR
dataset_1	100	2.9	100	1.7	100	1.2
dataset_2	100	12.0	100	2.6	100	2.2
dataset_3	100	4.1	100	3.7	100	3.4
dataset_4	95.7	8.3	100	6.3	100	5.3
Rata-Rata	98.92	6.82	100	3.57	100	3.02

Tabel 4.3 Tabel ketepatan metode Gaussian (dalam %)

Tabel 4.3 adalah hasil percobaan dengan menggunakan metode prediksi Gaussian yang dilakukan perubahan pada ukuran *sliding window*. Sel tabel berwarna biru menunjukkan metode dengan hasil terbaik pada saat melakukan pengujian pada dataset tertentu. Seperti pada percobaan dengan menggunakan SMOReg, percobaan ini selalu mencapai tingkat pendeteksian 100% pada 3 dataset awal, dan tidak pada dataset ke 4. Dapat dilihat bahwa dengan menggunakan ukuran 30 hasil yang didapatkan selalu paling buruk, sedangkan dengan menggunakan ukuran 60 hasil selalu paling baik. Dengan menggunakan nilai rata-rata sebagai acuan, ukuran 60 adalah yang terbaik untuk penggunaan metode prediksi Gaussian dengan nilai DR selalu mencapai angka 100% dan rata-rata FPR

3.02% dan tidak pernah mencapai nilai lebih dari 5.5% pada semua sliding window.

Dari tabel 4.2 dan 4.3 diatas, dapat dilihat bahwa pengaturan ukuran pada *sliding window* sangat berpengaruh terhadap proses pendeteksian yang dilakukan. Setelah dilakukan percobaan dengan metode prediksi dan dataset yang berbeda-beda, dapat dikatakan bahwa penggunaan metode Gaussian dengan menggunakan ukuran 60 adalah metode yang terbaik dengan nilai rasio pendeteksian mencapai 100% dengan rasio kesalahan 3.02%.

Selain melakukan percobaan untuk menguji tingkat ketepatan dalam pendeteksian, kami juga melakukan percobaan untuk menghitung waktu yang digunakan aplikasi dalam melakukan pemrosesan tiap data. Skenario yang digunakan sama dengan percobaan sebelumnya, yaitu dengan merubah metode prediksi dan ukuran *sliding window*. Selanjutnya kami mencatat hasil perhitungan waktu pada tabel 4.4 di bawah ini.

Dataset	Waktu					
	30		45		60	
	SMO	Gauss	SMO	Gauss	SMO	Gauss
dataset_1	0.16	0.12	0.33	0.32	0.65	0.58
dataset_2	0.14	0.14	0.33	0.31	0.62	0.62
dataset_3	0.16	0.15	0.34	0.34	0.66	0.63
dataset_4	0.16	0.12	0.34	0.3	0.6	0.6
Rata-Rata	0.15	0.13	0.34	0.31	0.64	0.60

Tabel 4.4 Tabel kecepatan pemrosesan (dalam detik/baris data)

Tabel 4.4 merupakan tabel hasil pengukuran waktu pemrosesan tiap *sample* data yang ada pada dataset. Dapat dilihat bahwa semakin besar ukuran *sliding window* yang digunakan, maka waktu yang dibutuhkan untuk memroses satu data menjadi semakin lama. Waktu yang dibutuhkan oleh kedua metode hampir sama untuk setiap ukuran *sliding*

window-nya. Saat menggunakan ukuran 30, waktu rata-rata yang dibutuhkan oleh program tidak pernah lebih dari 0,17 detik/baris data. Saat menggunakan ukuran 45, waktu paling lama adalah sekitar 0,35 detik/baris data. Dan apabila menggunakan ukuran 60, waktu pemrosesan yang dibutuhkan sekitar 0,6 detik/baris data. Dari hasil pengukuran rata-rata waktu yang dibutuhkan, dapat dilihat bahwa waktu yang dibutuhkan oleh metode Gaussian selalu lebih cepat dari metode SMOReg meskipun selisih waktunya kurang dari 0,1 detik.

[Halaman ini sengaja dikosongkan]

BAB V

PENUTUP

Bab ini merupakan akhir penulisan dari proyek akhir ini. Dalam bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari hasil penelitian, dan saran-saran yang bertujuan agar penelitian ini dapat diperbaiki dan dikembangkan di masa yang akan datang.

5.1 KESIMPULAN

Setelah melakukan berbagai tahapan mulai dari tahap perancangan, pembuatan sistem kemudian dilanjutkan pada tahap pengujian dan analisa, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem telah dapat melakukan pendeteksian anomali yang ada pada data hasil pengukuran anomali.
2. Pendeteksian anomali yang dilakukan pada percobaan menunjukkan performa yang baik dengan memberikan rasio pendeteksian (DR) yang sempurna pada hampir setiap percobaan dan memiliki rata-rata DR terbaik mencapai nilai 100% dan rata-rata nilai FPR tidak pernah melebihi angka 7%.
3. Metode prediksi Gaussian dengan ukuran *sliding window* 60 memberikan hasil yang terbaik.
4. Semakin besar ukuran *sliding window*, hasil pendeteksian akan semakin bagus pada dua metode prediksi yang digunakan.
5. Sistem dapat diimplementasikan pada kasus nyata karena memiliki waktu pemrosesan kurang dari 1 detik untuk tiap data.

5.2 SARAN

Dari hasil proyek akhir ini masih terdapat beberapa kekurangan serta dimungkinkan untuk dikembangkan lebih lanjut lagi. Oleh karenanya penulis merasa perlu menuliskan beberapa saran sebagai berikut:

- Pengembangan metode *dynamic threshold* dengan menggunakan metode *machine learning* dan pembobotan pada data training agar dapat memberikan dasar penentuan anomali yang lebih tepat.
- Mengimplementasikan aplikasi pendeteksian pada sistem pengukuran parameter kesehatan dengan menggunakan WBAN secara *realtime*.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] T. Yilmaz, R. Foster dan Y. Hao, "Detecting Vital Signs with Wearable Wireless Sensors," *Sensors* 2010, vol. 10, pp. 10837-10862, 2010.
- [2] O. Chipara, C. Lu, T. C. Bailey dan G.-C. Roman, "Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit," *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* , pp. 155-168, 2010.
- [3] V. Jha dan S. K. Srivastava, "New and Fast Outlier Detection Scheme in WSN: NFODS-WSN," *Journal of Global Research in Computer Science*, vol. III, no. 1, pp. 42-44, 2012.
- [4] A. Fawzy, H. M. Mokhtar dan O. Hegazy, "Outliers detection and classification in wireless," *Egyptian Informatics Journal*, p. 157–164, 2013.
- [5] R. Jurdak, X. R. Wang, O. Obst dan P. Valencia, "Chapter 12 Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies," *A. Tolk and L.C. Jain (Eds.): Intelligence-Based Systems Engineering*, pp. 309-325, 2011.
- [6] O. Salem, Y. Liu, A. Mehaoua dan R. Boutuba, "Online Anomaly Detection in Wireless Body Area Network for Reliable Healthcare Monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. XVIII, no. 5, pp. 1541-1551, 2014.
- [7] R. G. K dan D. B. K, "An Overview of Applications, Standards and Challenges in Futuristic Wireless Body Area Networks," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 180-186, 2012.
- [8] I. H. Witten, E. Frank, M. A. Hall dan C. J. Pal, "Introduction to Weka," dalam *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, Morgan Kaufmann, Fourth Edition, 2016., 2016, pp. 7-13.

- [9] A. Smola dan B. Schölkopf, "A tutorial on support vector regression," *Statistic and Computing*, no. 14, pp. 199-222, 2004.
- [10] C. E. Rasmussen dan C. K. I. Williams, *Gaussian Process for Machine Learning*, Massachusetts: MIT Press, 2006.
- [11] M. A. Mukid, A. H. Wigena dan Erfiani, "Regresi Proses Gaussian untuk Pemodelan Kalibrasi Spektroskopi," *Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA*, pp. 117-124, 2009.
- [12] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng dan H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet. Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. 215-220, 200.
- [13] "Confusion Matrix," [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Diakses 8 Desember 2016].
- [14] S. A. Haque, M. Rahman dan M. S. Aziz, "Sensor Anomaly Detection in Wireless Sensor Networks for Healthcare," *Sensors*, vol. 15, pp. 8764-8786, 2015.
- [15] T. L. Thorarinsdottir dan M. S. Johnson, "Probabilistic Wind Gust Forecasting Using Nonhomogeneous Gaussian Regression," *American Meteorology Society*, vol. 140, pp. 889-897, 2012.
- [16] V. Choubey, S. Mishra dan S. Pandey, "Time Series Data Mining in Real Time Surface Runoff Forecasting through Support Vector Machine," *International Journal of Computer Applications*, vol. 98, pp. 23-28, 2014.
- [17] PhysioNet, "PhysioNet LigthWAVE," PhysioNet, [Online]. Available: <https://www.physionet.org/lightwave/>. [Diakses 12 Januari 2017].

BIODATA PENULIS



Nama :

Fajar Setiawan

Tempat/Tanggal Lahir :

Surabaya, 14 Nopember 1994

Alamat :

Jl Tengger Raya II/42-A, Surabaya
60199

Email :

ffsetiawan@yahoo.co.id

Hobi :

Main, #B3IBSpacker-an,
nge- #PENSmania

Motto :

“Terima kasih kita dapat menghidupkan hari orang lain”

Riwayat Pendidikan :

- SD Negeri Kandangan II/620 Surabaya (2000-2006)
- SMP Negeri 20 Surabaya (2006-2009)
- SMA Negeri 11 Surabaya (2009-2012)
- D3 Teknik Informatika PENS (2012-2015)

[Halaman ini sengaja dikosongkan]