

SKRIPSI

Deteksi Tekanan Darah Berdasarkan Sinyal PPG

dengan Metode *Statistical Analysis* dan *Extreme Learning Machine*



ALDAFFAN SHEVA GHIFARI WYDIANDHIKA

PROGRAM STUDI S1 TEKNIK BIOMEDIS

DEPARTEMEN FISIKA

FAKULTAS SAINS DAN TENOLOGI

UNIVERSITAS AIRLANGGA

2022

**DETEKSI TEKANAN DARAH BERDASARKAN SINYAL PPG DENGAN
METODE *STATISTICAL ANALYSIS* DAN *EXTREME LEARNING MACHINE***

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik
Bidang Teknik Biomedis pada Fakultas Sains dan Teknologi Universitas Airlangga

Oleh

ALDAFFAN SHEVA GHIFARI WYDIANDHIKA

NIM: 081811733019

Disetujui Oleh

Pembimbing I



Dr. Riries Rulaningtyas S.T., M.T.

NIP. 197903152003122002

Pembimbing II



Osmalina Nur Rahma S.T., M.Si.

NIP. 199109302019032018

LEMBAR PENGESAHAN

Judul : Deteksi Tekanan Darah berdasarkan sinyal PPG dengan Metode *Statistical Analysis* dan *Extreme Learning Machine*
Penyusun : Aldaffan Sheva Ghifari Wydiandhika
NIM : 081811733019
Tanggal Sidang : Selasa, 16 Agustus 2022

Disetujui oleh,

Pembimbing I,



Dr. Riries Rulaningtvas S.T., M.T.

NIP. 197903152003122002

Mengetahui,

Ketua Departemen Fisika.



Herri Trilaksana, S.Si., M.Si., Ph.D.

NIP. 197712282003121003

Pembimbing II,



Osmalina Nur Rahma S.T., M.Si.

NIP. 199109302019032018

Mengetahui,

Koordinator Program Studi S1 Teknik
Biomedis,



Dr. Riries Rulaningtvas, S.T., M.T.

NIP. 197903152003122002

PEDOMAN PENGGUNAAN SKRIPSI

Skripsi ini tidak dipublikasikan, namun tersedia di perpustakaan dalam lingkungan Universitas Airlangga, diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun dan harus menyebutkan sumbernya sesuai kebiasaan ilmiah.

Dokumen skripsi ini merupakan hak milik Universitas Airlangga.

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Aldaffan Sheva Ghifari Wydiandhika
NIM : 081811733019
Program Studi : Teknik Biomedis
Judul Skripsi : **Deteksi Tekanan Darah berdasarkan sinyal PPG
dengan Metode *Statistical Analysis* dan *Extreme
Learning Machine***

Menyatakan dengan sesungguhnya bahwa bagian atau keseluruhan Skripsi ini tidak pernah diajukan untuk mendapat gelar akademis pada bidang studi dan/atau universitas lain dan tidak pernah dipublikasikan/ditulis oleh individu selain penyusun kecuali bila dituliskan dengan format kutipan dalam isi Skripsi.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku di Universitas Airlangga. Pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Surabaya,

Yang membuat pernyataan,



Aldaffan Sheva Ghifari Wydiandhika

081811733019

SKRIPSI DETEKSI TEKanan DARAH... ALDAFFAN SHEVA G W

Aldaffan Sheva Ghifari Wydiandhika, 2022. **Deteksi Tekanan Darah berdasarkan sinyal PPG dengan Metode *Statistical Analysis* dan *Extreme Learning Machine***. Skripsi dibawah bimbingan Dr. Riries Rulaningtyas S.T., M.T. dan Osmalina Nur Rahma S.T., M.Si. Program studi S – 1 Teknik Biomedis, Departemen Fisika, Fakultas Sains dan Teknologi, Universitas Airlangga.

ABSTRAK

Hipertensi merupakan salah satu faktor penyebab penyakit seperti stroke, infark miokard, kematian kardiovaskular, hingga kematian akibat naiknya tekanan darah. Pengukuran tekanan darah umumnya dilakukan dengan metode *non – invasive*, yaitu menggunakan tensimeter. Namun, penggunaan tensimeter baik manual ataupun digital tidak dapat dilakukan secara kontinyu, sehingga tidak dapat digunakan untuk monitoring dalam jangka panjang. Seiring perkembangan teknologi, *Photoplethysmography* (PPG) dapat dikembangkan untuk monitoring tekanan darah atau deteksi kondisi hipertensi. Oleh karena itu, perlu dibuat sistem yang dapat menghipertensi dari sinyal PPG yang memiliki nilai akurasi yang tinggi. Penelitian ini menggunakan data sekunder yang didapat dari pihak Komite Etik Pusat Medis Universiti Kebangsaan Malaysia dalam bentuk *raw* data dengan subjek sebanyak 57 orang dengan 30 orang subjek normal dan 27 orang subjek hipertensi. Sebelum diklasifikasi, data ini diolah ke dalam 4 proses *pre – processing*, yaitu *detrend* data, *noise filtering* dengan *Band Pass Filter* (BPF), normalisasi data dengan metode min – max, dan segmentasi data. Fitur untuk klasifikasi hipertensi yang digunakan dalam penelitian ini adalah *statistical analysis*, yaitu nilai *skewness*, *prominence*, dan *width*. Fitur – fitur ini kemudian dibagi menjadi 3 tahapan, yaitu *skewness*, *peak analysis*, dan gabungan antara *skewness* dan *peak analysis*. Fitur ini kemudian akan digunakan sebagai *input* klasifikasi hipertensi menggunakan sistem *extreme learning machine* (ELM). Hasil klasifikasi dari ELM adalah berupa dua kelas, yaitu normal atau hipertensi. Dalam penelitian ini, *hidden neuron* yang digunakan dimulai dari 10, 50, 100, 500, 1000, dan 1500. Hasil akurasi tertinggi mencapai 89.66% untuk *training* dan 75.72% untuk *testing* dengan menggunakan 1000 *hidden neuron* yang didapat pada fitur gabungan antara *skewness* dan *peak analysis*.

Kata kunci : Hipertensi, *statistical analysis*, *skewness*, *peak analysis*, ELM

Aldaffan Sheva Ghifari Wydiandhika, 2022. **Deteksi Tekanan Darah berdasarkan sinyal PPG dengan Metode *Statistical Analysis* dan *Extreme Learning Machine***. Undergraduate thesis supervised by Dr. Riries Rulaningtyas S.T., M.T. and Osmalina Nur Rahma S.T., M.Si. Biomedical Engineering bachelor program, Department of Physics, Science and Technology Faculty, Airlangga University.

ABSTRACT

Hypertension is one of the factors causing diseases such as stroke, myocardial infarction, cardiovascular death, and death due to increased blood pressure. Blood pressure measurement is generally carried out by non-invasive methods, namely using a sphygmomanometer. However, the use of both manual and digital sphygmomanometers cannot be carried out continuously, so they cannot be used for long – term monitoring. Along with technological developments, Photoplethysmography (PPG) can be developed for monitoring blood pressure or detecting hypertension conditions. Therefore, it is necessary to create a system that can classify hypertension from PPG signals that have a high accuracy value. This research uses secondary data obtained from the Ethics Committee of the Universiti Kebangsaan Malaysia Medical Center in the form of raw data with 57 subjects with 30 normal subjects and 27 hypertension subjects. Before being classified, this data is processed into 4 pre-processing processes, which is detrend data, noise filtering with Band Pass Filter (BPF), data normalization using the min – max method, and data segmentation. Classification features of hypertension used in this research is statistical analysis, which is the value of skewness, prominence, and width. These features are divided into 3 stages, skewness, peak analysis, and a combination of skewness and peak analysis. All of this feature will be used as input for hypertension classification using the extreme learning machine (ELM) system. The results of the classification of the ELM are divided into two classes, which is normal or hypertension. In this research, the hidden neurons used started from 10, 50, 100, 500, 1000, and 1500. The highest accuracy results reached 89.66% for training and 75.72% for testing using 1000 hidden neurons obtained on the combined features of skewness and peak analysis. The highest ELM result accuracy reached 89.66% for training and 75.72% for testing using 1000 hidden neurons obtained on the combined features of skewness and peak analysis.

Keywords : *Hypertension, statistical analysis, skewness, peak analysis, ELM*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat rahmat dan hidayah – Nya skripsi dengan judul “**Deteksi Tekanan Darah berdasarkan sinyal PPG dengan metode *statistical analysis* dan *Extreme Learning Machine***” dapat diselesaikan. Penyusunan skripsi ini merupakan bentuk syarat akademik guna mendapatkan kesempatan untuk melakukan sidang skripsi.

Penulis menyadari bahwa penyusunan skripsi ini tidak dapat berjalan tanpa bantuan, arahan, dan bimbingan dari berbagai pihak. Oleh karenanya, penulis mengucapkan banyak terimaaksih kepada pihak yang telah bersedia membantu, meluangkan waktu, dan pengetahuannya khususnya kepada :

1. Allah SWT yang telah memberikan rahmat dan hidayah – Nya sehingga penulis dapat menyelesaikan naskah skripsi.
2. Isydianto Adji N. B dan Aniek Soellahmi selaku orang tua serta kepada seluruh keluarga yang telah membantu, mendukung, dan mendoakan selama dalam waktu perkuliahan dan penyusunan skripsi.
3. Bapak Herri Trilaksana, S.Si., M.Si., Ph.D., selaku Ketua Departemen Fisika Fakultas Sains dan Teknologi, Universitas Airlangga
4. Ibu Dr. Riries Rulaningtyas S.T., M.T., M.Si selaku Ketua Program Studi Teknik Biomedis Departemen Fisika Fakultas Sains Dan Teknologi, Universitas Airlangga
5. Ibu Dr. Riries Rulaningtyas S.T., M.T., sebagai dosen pembimbing pertama dan kepada Ibu Osmalina Nur Rahma S.T., M.Si. sebagai dosen pembimbing kedua yang telah banyak meluangkan waktunya untuk membimbing penulisan skripsi ini.
6. Bapak dan Ibu dosen Program Studi Teknik Biomedis Departemen Fisika Fakultas Sains Dan Teknologi, Universitas Airlangga yang telah memberikan ilmu yang bermanfaat bagi penulis.
7. Teman – teman penulis yang telah mendukung dan membantu selama masa kuliah hingga penulisan skripsi ini.

Penulis menyadari skripsi ini tidak luput dari kekurangan. Oleh karena itu, sangat diharapkan kritik dan saran yang membangun sehingga skripsi ini dapat memberikan manfaat bagi bidang pendidikan dan penerapan Teknik Biomedis

Surabaya. 11 Agustus 2022

Penulis

Aldaffan Sheva Ghifari Wydiandhika

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERNYATAAN	i
LEMBAR PENGESAHAN	ii
PEDOMAN PENGGUNAAN SKRIPSI	iii
LEMBAR PERNYATAAN KEASLIAN	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II.....	5
TINJAUAN PUSTAKA.....	5
2.1 Hipertensi	5
2.2 <i>Photoplethysmograph</i> (PPG).....	6
2.2.1 PPG <i>Waveform</i>	6
2.2.2 Fisiologi PPG	7
2.2.3 PPG untuk Pengukuran Hipertensi.....	9
2.3 Filter	12
2.3.1 <i>Band Pass Filter</i> (BPF).....	12
2.4 Fitur Statistik	13
2.4.1 <i>Skewness</i> (Kesimetrisan).....	14
2.4.2 <i>Peak Analysis</i>	14

2.5	<i>Extreme Learning Machine (ELM)</i>	15
BAB III.....		18
METODE PENELITIAN.....		18
3.1	Tempat dan Waktu Penelitian	18
3.2	Alat dan Bahan Penelitian	18
3.3	Prosedur Penelitian.....	18
3.3.1	Pengumpulan Data	18
3.3.2	<i>Pre – Processing</i>	20
3.3.3	Ekstraksi Fitur	21
3.3.4	Tahap Klasifikasi	23
3.3.5	Tahap Pengujian dan Analisis Data	26
BAB IV		28
HASIL DAN PEMBAHASAN.....		28
4.1	Pengumpulan Data.....	28
4.2	<i>Pre – processing</i>	29
4.3	Ekstraksi Fitur	34
4.4	Hasil Klasifikasi dan Pengujian Menggunakan ELM	42
BAB V.....		46
KESIMPULAN DAN SARAN.....		46
5.1	Kesimpulan.....	46
5.2	Saran	46
DAFTAR PUSTAKA		47
LAMPIRAN		50

DAFTAR GAMBAR

Gambar 2.1 Komponen pulsatil (AC) dari sinyal PPG dan elektrokardiogram (EKG) terkait (Allen, 2014)	7
Gambar 2.2 Skema bentuk gelombang PPG yang dihasilkan karena absorbansi cahaya dalam jaringan, sebagai fungsi waktu (Tamura <i>et al.</i> , 2014).	9
Gambar 2.3 Bentuk gelombang PPG model parametrik (Chowdhury <i>et al.</i> , 2020)	11
Gambar 2.4 Bentuk gelombang PPG model nonparametrik (Tjahjadi, Ramli and Murfi, 2020)	11
Gambar 2.5 Block Diagram <i>Band Pass Filter</i> (Tony R. Kuphaldt, 2007)	12
Gambar 2.6 Rangkaian <i>Band Pass Filter</i> (BPF) (Tony R. Kuphaldt, 2007)	13
Gambar 2.7 Plot magnitude <i>Band Pass Filter</i> (BPF) (Huelsman, 1993)	13
Gambar 2.8 Struktur dari <i>Extreme Learning Machine</i> (Ding <i>et al.</i> , 2015)	16
Gambar 3.1 Diagram langkah penelitian	18
Gambar 3.2 Tahapan <i>pre – processing</i> .	20
Gambar 3.3 Fitur <i>skewness</i>	22
Gambar 3.4 Fitur <i>Peak Analysis</i> .	22
Gambar 3.5 Diagram proses <i>training</i> .	25
Gambar 3.6 Diagram proses <i>testing</i> .	26
Gambar 4.1 Data subjek normal (atas) dan hipertensi (bawah) yang diambil dalam bentuk <i>raw</i> .	29
Gambar 4.2 Sinyal data normal (atas) dan hipertensi (bawah) setelah dilakukan <i>detrend</i> data.	30
Gambar 4.3 Data sinyal normal (atas) dan hipertensi (bawah) setelah difilter dengan BPF.	32
Gambar 4.4 Sinyal hasil normalisasi pada data normal (atas) dan hipertensi (bawah)	33
Gambar 4.5 Hasil segmentasi sinyal per segmen untuk data normal (atas) dan hipertensi (bawah).	34
Gambar 4.6 Contoh data yang memiliki <i>peak</i> rendah.	36
Gambar 4.7 Pengambilan fitur <i>prominence</i> dan <i>width</i> pada matlab.	37

Gambar 4.8 Data yang terdeteksi memiliki 1 *peak* (atas), 2 *peak* (tengah), dan 3 *peak* (bawah).

38

DAFTAR TABEL

Tabel 3.1 Tabel parameter pasien yang diukur	19
Tabel 4.1 <i>Range</i> parameter dari subjek	28
Tabel 4.2 Tabel hasil informasi data normal yang didapat dari fitur <i>findpeaks</i> .	35
Tabel 4.3 Tabel hasil informasi data hipertensi yang didapat dari fitur <i>findpeaks</i> .	35
Tabel 4.4 Data nilai <i>skewness</i> yang didapatkan untuk sinyal hipertensi dan normal	38
Tabel 4.5 Data fitur <i>skewness</i> sebagai masukkan ELM setelah pelabelan	40
Tabel 4.6 Data fitur <i>peak analysis</i> sebagai masukkan ELM setelah pelabelan	41
Tabel 4.7 Data fitur gabungan <i>skewness</i> dan <i>peak analysis</i> sebagai masukkan ELM setelah pelabelan	42
Tabel 4.8 Hasil akurasi pada masing – masing fitur dalam berbagai macam <i>hidden neuron</i>	44
Tabel 4.9 Hasil lama proses pada masing – masing fitur dalam berbagai macam <i>hidden neuron</i> dalam sekon	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada masa sekarang, hipertensi merupakan salah satu penyakit yang sering ditemui. Menurut *World Health Organization* (WHO), hipertensi merupakan suatu keadaan dimana peningkatan darah sistolik berada diatas batas normal, yaitu lebih dari 140 mmHg dan tekanan darah diastolik lebih dari 90 mmHg. Hipertensi dapat diketahui melalui karakteristik tekanan darah yang dihasilkan oleh tubuh. Tekanan darah merupakan parameter penting untuk deteksi dini penyakit jantung karena berhubungan dengan gejala hipertensi maupun hipotensi (Tjahjadi, Ramli and Murfi, 2020). Tekanan darah mendorong aliran darah pada pembuluh darah dan berperan penting dalam dinamika aliran darah di setiap interval detak jantung (Li and Laleg-Kirati, 2021). Tekanan darah mengukur daya dari pompa jantung yang diberikan ke arteri saat beredar ke seluruh tubuh. Hasil pengukuran tekanan darah terdiri dari tiga parameter, yaitu *Diastolic Blood Pressure* (DBP), *Systolic Blood Pressure* (SBP), dan *Mean Arterial Pressure* (MAP) dalam satuan milimeter air raksa (mmHg). (Tjahjadi, Ramli and Murfi, 2020). Pengukuran tekanan darah dapat memberikan informasi mengenai kondisi jantung pada tubuh seseorang.

Hipertensi merupakan salah satu faktor penyebab penyakit seperti stroke, infark miokard, kematian kardiovaskular, hingga kematian akibat naiknya tekanan darah (Pradono, 2014). Hipertensi menjadi penyakit tidak menular yang paling banyak paling umum bersamaan dengan diabetes. Menurut data WHO, hipertensi menyumbang hampir 46,2% dan 4% dari total kematian akibat penyakit tidak menular (Fitriyani *et al.*, 2019). Fitriyani dkk pada tahun 2019 menyebutkan bahwa sekitar 639 juta orang dewasa terkena hipertensi dan diperkirakan mencapai hampir satu miliar orang dewasa pada tahun 2025. Sedangkan López-Martínez dkk pada tahun 2018 menuliskan bahwa di AS hipertensi mengalami peningkatan pada populasi orang dewasa dan dapat dilihat pada *National Health and Nutrition*

Examination Survey (NHANES) yang dilakukan oleh *National Center for Health Statistics*.

Pengukuran tekanan darah umumnya dilakukan dengan metode *non – invasive*, yaitu menggunakan tensimeter. Namun, penggunaan tensimeter baik manual ataupun digital tidak dapat dilakukan secara kontinyu, sehingga tidak dapat digunakan untuk monitoring dalam jangka panjang. Seiring perkembangan teknologi, *Photoplethysmography* (PPG) dapat dikembangkan untuk monitoring tekanan darah atau deteksi kondisi hipertensi. Teknologi PPG telah digunakan dalam berbagai perangkat medis yang tersedia secara komersial untuk mengukur saturasi oksigen, tekanan darah, curah jantung, menilai fungsi otonom dan mendeteksi penyakit pembuluh darah perifer (Allen, 2014). Metode pengukuran menggunakan PPG untuk pengukuran tekanan darah menawarkan solusi yang lebih nyaman daripada metode konvensional (Tjahjadi, Ramli and Murfi, 2020). Selain itu, metode PPG dapat melakukan pengukuran secara kontinyu karena PPG sendiri merupakan teknologi optik yang sederhana dan efisien untuk mengukur perubahan volume darah. PPG juga diharapkan dapat digunakan sebagai alat monitoring secara *real time* untuk kedepannya, dimana selain dapat memonitoring kadar oksigen, PPG juga dapat digunakan sebagai monitoring hipertensi atau mungkin dapat digunakan seperti ECG.

Penelitian terkait pengukuran tekanan darah dengan PPG berkembang menjadi topik hangat yang sering dilakukan. Sinyal yang dihasilkan PPG dapat dipisahkan menjadi sistolik dan diastolik. Dari sinyal sistolik dan diastolik ini, nantinya dapat diklasifikasikan antara penderita hipertensi ataupun yang normal. Namun, salah satu tantangan dalam penggunaan sinyal PPG untuk klasifikasi hipertensi adalah belum ditemukannya feature yang dapat digunakan secara efektif dan memiliki akurasi yang tinggi. Para peneliti masih melakukan pengkajian ulang dengan mencoba berbagai macam fitur untuk melihat tingkat akurasinya. Seperti pada penelitian yang dilakukan oleh Li dan Laleg – Kirati pada tahun 2021 yang menggunakan fitur *semi-classical signal analysis* (SCSA) dimana hasilnya

menghasilkan performa yang sebanding jika dibandingkan dengan standar *Advancement of Medical Instrumentation* (AAMI). Penelitian lain seperti yang dilakukan oleh Tjahjadi, Ramli, dan Murfi pada tahun 2020, menggunakan fitur kesimetrisan (*skewness*) untuk mengetahui kualitas sinyal PPG. *Skewness* sendiri merupakan metode paling optimal untuk menilai *Signal Quality Index* (SQI) dalam sinyal PPG.

Fitur – fitur diatas nantinya akan dikombinasikan sebagai input dari sebuah *software* yang didalamnya sudah terdapat *Artificial Intelligence*. Penggunaan *artificial intelligence* dalam deteksi hipertensi telah membuktikan hasil yang signifikan dalam dunia medis. Oleh karenanya, saat ini telah banyak referensi penelitian terkait diagnosis hipertensi dengan beragam model fitur dan *artificial intelligence*. Salah satunya adalah penelitian yang dilakukan oleh Zhang, dkk pada tahun 2019, dimana mereka melakukan prediksi tingkat tekanan darah menggunakan *Gradient Boosting Decision Tree* (GBDT). Penelitian ini menghasilkan tingkat akurasi diatas 70% untuk tekanan sistolik dan akurasi diatas 64% untuk tekanan diastolik dengan waktu prediksi kurang dari 0.1 detik.

Pengklasifikasian dilakukan dengan mengolah data yang didapat melalui ekstraksi fitur yang kemudian dimasukkan ke dalam *artificial intelligence*. Penelitian yang dilakukan oleh Tjahyadi dan Ramli pada tahun 2020 menggunakan *K- Nearest Neighbors* (KNN) menghasilkan nilai F1 dalam tiga klasifikasi secara berurutan sebesar 83.34%, 94.84%, dan 88.49% (Tjahyadi and Ramli, 2020). Namun, pada penggunaan KNN diperlukan waktu yang lama terutama untuk *training process*. Penelitian lain yang dilakukan adalah membandingkan performa *Support Vector Machine* (SVM) dan *Extreme Learning Machine* (ELM). Didapatkan bahwa metode ELM memiliki akurasi yang lebih tinggi dari SVM. ELM telah terbukti menjadi pengklasifikasi dengan kinerja terbaik dibandingkan dengan pengklasifikasi lainnya bahkan dalam objek tertentu dapat mencapai akurasi hingga 99,2 % (Zhang, Zhang and Tian, 2016). Berdasarkan penelitian – penelitian tersebut, penulis melakukan penelitian untuk membuat *software* yang

dapat mengklasifikasi hipertensi dari sinyal PPG. Data yang didapat dilakukan *detrend*, *filtering*, normalisasi, dan kemudian di ekstraksi fitur menggunakan fitur *peak analysis* dan *skewness*. Hasil ekstraksi fitur akan dimasukkan ke dalam *artificial intelligence*, yaitu *machine learning* jenis *Extreme Learning Machine* (ELM) untuk diklasifikasi menjadi penderita hipertensi ataupun normal.

1.2 Rumusan Masalah

1. Fitur apa yang optimal untuk sistem klasifikasi hipertensi dari sinyal PPG?
2. Berapa tingkat akurasi tertinggi sistem klasifikasi hipertensi menggunakan metode ELM berbasis sinyal PPG?

1.3 Batasan Masalah

1. Sistem klasifikasi hanya mengklasifikasi menjadi dua, yaitu normal dan hipertensi
2. Kondisi pasien saat pengambilan data tidak memiliki penyakit lain (komplikasi) berdasarkan uji klinis

1.4 Tujuan Penelitian

1. Mengetahui fitur yang optimal untuk sistem klasifikasi hipertensi dari sinyal PPG
2. Mendapatkan tingkat akurasi tertinggi untuk sistem klasifikasi hipertensi menggunakan metode ELM berbasis sinyal PPG

1.5 Manfaat Penelitian

Penelitian ini diharapkan memiliki beberapa manfaat sebagai berikut :

1. Sebagai media penunjang bagi peneliti dalam mengembangkan fitur klasifikasi hipertensi.
2. Sebagai dasar penelitian selanjutnya untuk dapat dikembangkan sebagai metode klasifikasi hipertensi.
3. Sebagai media untuk membantu masyarakat dalam menjaga tekanan darah sehingga dapat mengurangi resiko terkena penyakit lain yang ditimbulkan akibat adanya hipertensi.

BAB II

TINJAUAN PUSTAKA

2.1 Hipertensi

Menurut KBBI, hipertensi adalah tekanan darah atau denyut jantung yang lebih tinggi daripada keadaan normal karena penyempitan pembuluh darah atau gangguan lainnya. Batas tekanan darah normal adalah pada 140 mmHg untuk tekanan sistolik dan 90 mmHg untuk tekanan diastolik (Pradono, 2014). Yang menjadi masalah yaitu apabila dibiarkan hipertensi dapat menimbulkan masalah kardiovaskular yang menjadi fatal. Peningkatan tekanan darah ini menurunkan risiko fungsi otak, jantung, dan ginjal (Lim, Solmi and Cortese, 2021).

Pada tahun 2010, hipertensi diidentifikasi sebagai faktor risiko utama untuk beberapa penyakit global, terhitung 9.4 juta kematian disebabkan oleh hipertensi (Lim, Solmi and Cortese, 2021). Data *World Health Organization* (WHO) pada tahun 2015, sekitar 1.13 Miliar orang di dunia mengidap hipertensi, yang berarti 1 dari tiap 3 orang di dunia mengalami hipertensi. Jumlah penyandang hipertensi terus meningkat setiap tahunnya, diperkirakan pada tahun 2025, penderita hipertensi akan mencapai angka 1.5 miliar. Di Indonesia sendiri, menurut data yang diambil oleh *Sample Registration System* (SRS) tahun 2014 menunjukkan bahwa hipertensi dengan komplikasi menempati urutan kelima sebagai penyebab kematian pada semua umur. Berdasarkan Riset Kesehatan Dasar (Riskesdas) yang dilakukan Kemenkes pada tahun 2018 memperlihatkan bahwa hasil pengukuran pengidap hipertensi di tiap daerah pada penduduk usia 18 tahun sebesar 34.1% dan terendah pada angka 22.2%. Selain itu, hipertensi yang terjadi pada kelompok umur 31 – 44 memiliki persentase tahun (31.6%), umur 45 – 54 tahun (45.3%), umur 55 – 64 tahun (55.2%).

Hipertensi biasanya ditandai dengan gejala – gejala seperti penyakit lain pada umumnya, yaitu mual, muntah, sakit kepala, ataupun nyeri dada. Adanya hipertensi ini biasanya disebabkan karena faktor keturunan. Selain dari faktor keturunan,

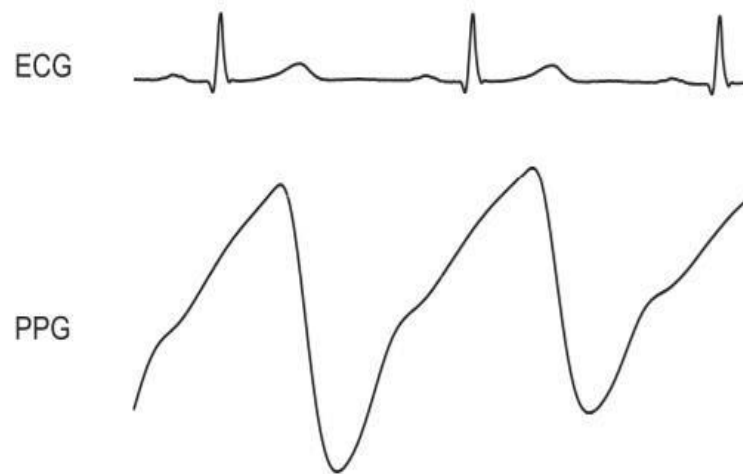
hipertensi juga bisa disebabkan karena perilaku dari penderita. Faktor umur, hubungan dengan pasangan, konsumsi garam, konsumsi kopi, dan rutinitas konsumsi obat hipertensi berhubungan erat dengan penyebab hipertensi (Artiyaningrum and Azam, 2016). Faktor lain seperti konsumsi minuman beralkohol, kebiasaan merokok, kurangnya aktifitas fisik, serta kurangnya konsumsi sayur dan buah juga menjadi faktor yang meningkatkan risiko terkena hipertensi (Pradono, 2014).

2.2 Photoplethysmograph (PPG)

Salah satu metode yang sedang populer untuk mengetahui tingkat tekanan darah adalah *photoplethysmograph* (PPG). PPG merupakan salah satu metode non-invasif yang simple dan mudah diaplikasikan. Untuk memunculkan sinyal PPG dibutuhkan beberapa komponen optoelektronik, yaitu *light emitting diode* (LED) dan *photodetector*. LED adalah sumber cahaya yang dapat digunakan untuk menerangi pembuluh darah sehingga perubahan perfusi kecil dapat dipantau di *photodetector*. Perfusi diukur sebagai kecepatan pengiriman darah ke jaringan (Tjahjadi, Ramli and Murfi, 2020). Prinsip dasar PPG adalah mendeteksi perbedaan laju absorpsi cahaya yang disebabkan oleh perubahan aliran darah di pembuluh mikrovaskuler yang ditangkap oleh *photodiode* (PD) (Lin *et al.*, 2015).

2.2.1 PPG Waveform

Bentuk gelombang PPG sering disebut dengan komponen 'AC' dan biasanya memiliki frekuensi dasar yang biasanya bernilai sekitar 1 Hz, tergantung pada detak jantung seperti pada Gambar 2.1. Komponen AC ini bersifat *superimposed* pada komponen quasi yang besar yang berhubungan dengan jaringan dan volume darah. Sedangkan komponen DC bervariasi secara perlahan karena respirasi, aktivitas vasomotor, dan gelombang vasokonstriktor (Allen, 2014).



Gambar 2.1 Komponen pulsatil (AC) dari sinyal PPG dan elektrokardiogram (EKG) terkait (Allen, 2014)

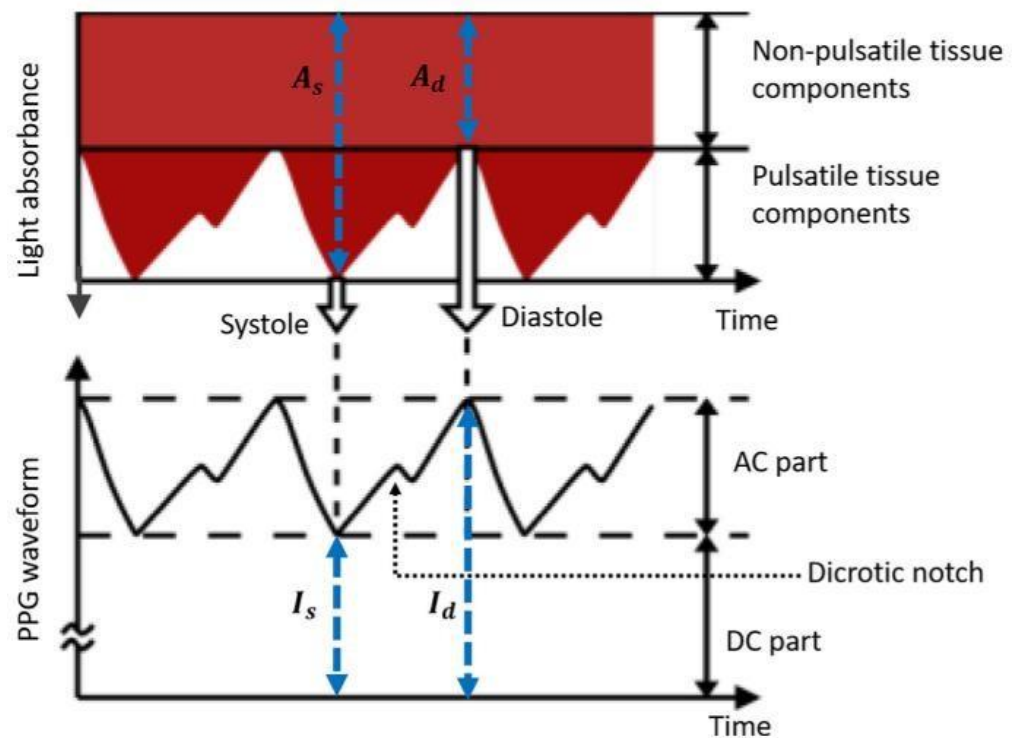
2.2.2 Fisiologi PPG

Photoplethysmograph menggunakan absorptivitas cahaya yang dihasilkan dari variasi fisiologis jaringan selama siklus jantung berjalan. Siklus jantung ini terdiri dari kondisi sistolik dan diastolik. Dalam keadaan sistolik, darah akan dipompa ke seluruh tubuh termasuk ke dalam jaringan perifer, sehingga tekanan darah sistolik akan meningkat dan menghasilkan absorbansi cahaya dalam jaringan dibandingkan dengan keadaan diastolik. Secara umum, perubahan relatif dalam absorbansi cahaya ini menimbulkan bentuk gelombang pulsatil PPG yang sinkron dengan setiap detak jantung (Moyle, 2003).

Intensitas PPG yang direkam (I) memiliki hubungan yang terbalik dengan absorbansi cahaya (A). Berdasarkan efek absorbansi cahaya, intensitas PPG dapat dibagi menjadi komponen DC dan AC. Komponen DC nonpulsatile dihasilkan karena adanya penyerapan cahaya dibagian jaringan nonpulsatile seperti otot, tulang, dan sebagainya. Sedangkan komponen AC pulsatile didapat dari penyerapan cahaya dalam arteri yang berdenyut dan bervariasi bersamaan dengan detak jantung. Dalam komponen AC terdapat dua fase dimana, fase tepi naik (*anacrotic*) yang berhubungan dengan kondisi sistolik dan fase tepi jatuh

(*catatrotic*) yang berhubungan dengan kondisi diastolik. Dalam sinyal PPG didapati pula *dicrotic notch* yang merupakan penanda akhir dari fase sistolik dan awal dari diastolik (Kyriacou and Chatterjee, 2022).

Absorbansi cahaya pada komponen jaringan pulsatile menghasilkan bagian AC dalam bentuk gelombang PPG. Absorbansi dalam komponen jaringan nonpulsatile berkontribusi pada bagian DC yang bervariasi secara perlahan dari bentuk gelombang PPG. Absorbansi di bagian sistolik dan diastolik dilambangkan sebagai A_s dan A_d , sedangkan intensitas cahaya yang sesuai dilambangkan sebagai I_s dan I_d . Korespondensi kualitatif antara A dan I yang ditunjukkan dalam Gambar 2.2. Komponen DC dari bentuk gelombang PPG merupakan indikasi sinyal optik yang ditransmisikan yang terdeteksi dari pantulan jaringan dan bergantung pada struktur jaringan dan volume darah rata – rata pada arteri dan vena. Komponen DC akan berubah perlahan sesuai adanya respirasi. Komponen AC menunjukkan perubahan volume darah yang terjadi pada fase sistolik dan diastolik dari siklus jantung. Frekuensi dasar komponen AC bergantung pada detak jantung (Tamura *et al.*, 2014).



Gambar 2.2 Skema bentuk gelombang PPG yang dihasilkan karena absorpsi cahaya dalam jaringan, sebagai fungsi waktu (Tamura *et al.*, 2014).

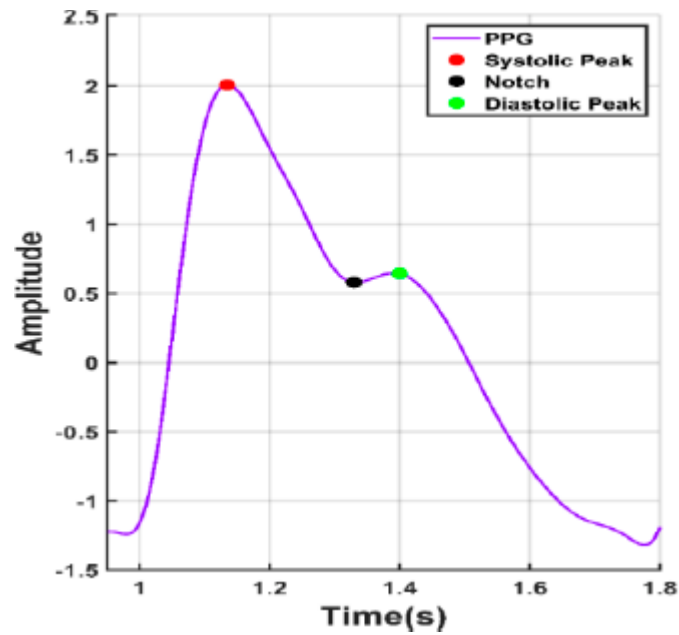
2.2.3 PPG untuk Pengukuran Hipertensi

PPG biasanya digunakan untuk mengukur kadar oksigen yang ada dalam darah. Salah satu contoh alat yang sering menggunakan PPG adalah pada *pulse oximeter*. Selain itu, PPG juga sering digunakan sebagai pengukur detak jantung (Lin *et al.*, 2015). *Photoplethysmograph* (PPG) digunakan selama beberapa dekade untuk mengukur jumlah cahaya yang diserap atau dipantulkan oleh pembuluh darah di jaringan tubuh. Teknologi PPG adalah teknologi serbaguna dengan biaya rendah yang dapat mencakup berbagai aspek pengawasan kardiovaskular termasuk identifikasi saturasi oksigen darah, denyut jantung, perkiraan tekanan darah, curah jantung, respirasi, penuaan arteri, kontrol endotel, aliran darah mikro-vaskular, dan fungsi otonom (Chowdhury *et al.*, 2020).

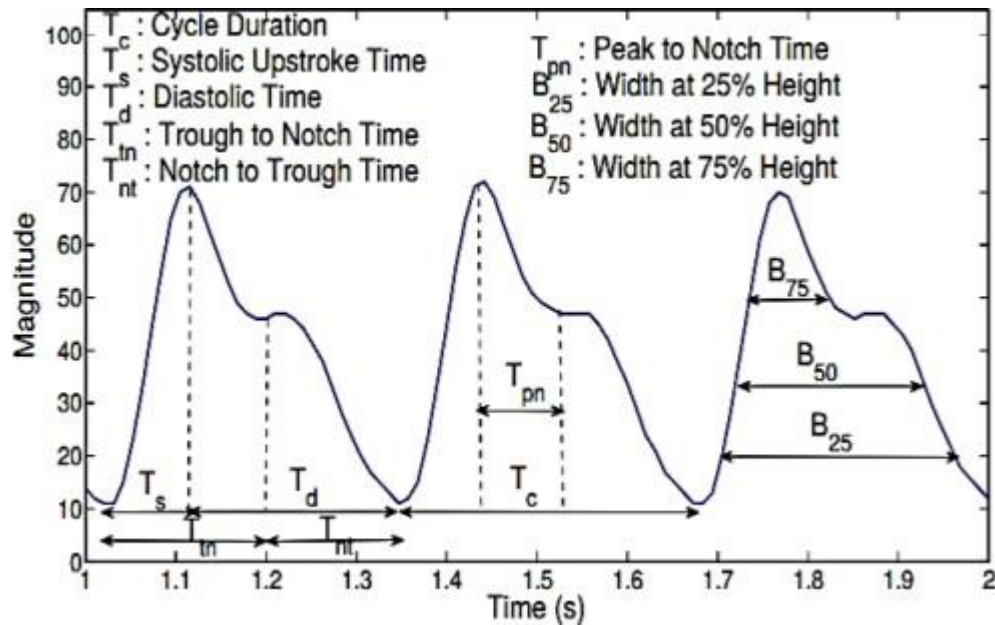
PPG akan menangkap perubahan warna yang terjadi dalam darah. Perubahan warna dan penyerapan cahaya diatur oleh sirkulasi darah dalam tubuh, yang

dipengaruhi oleh sistol dan diastol jantung yang berdetak. Tekanan darah seseorang akan meningkat ketika ada kontraksi otot jantung selama sistol yang mendorong darah menuju perifer. Propagasi gelombang darah ini menghasilkan peningkatan tekanan darah. Hal ini menghasilkan sinyal periodik yang jelas dan eksplisit dan puncak diastol yang halus. Setiap siklus ini berulang secara periodik yang menghasilkan satu puncak sistol dan diastol, dimana biasanya puncak pertama lebih mudah dideteksi dibandingkan dengan puncak kedua. Hubungan antara PPG dan tekanan darah dapat diketahui dari perubahan kecepatan aliran darah, yang biasanya disebut dengan *pulse wave velocity* (PWV). Ketika pembuluh darah berkontraksi, kecepatan aliran darah dan tekanan meningkat sehingga perubahan kecepatan ini akan mempengaruhi intensitas cahaya yang ditangkap oleh PPG (Slapni Č Ar, Mlakar and Luštrek, 2019).

Saat ini, ada dua cara untuk menentukan tekanan darah hanya dengan menggunakan PPG. Yang adalah model parametrik yang mencoba mengekstrak parameter tertentu seperti periode sistolik, detak jantung, dan diastolik dari setiap sinyal PPG. Pendekatan kedua melibatkan model nonparametrik, yang mencoba mengekstrak fitur spesifik dalam domain frekuensi atau domain waktu (Tjahjadi, Ramli and Murfi, 2020).



Gambar 2.3 Bentuk gelombang PPG model parametrik (Chowdhury *et al.*, 2020)



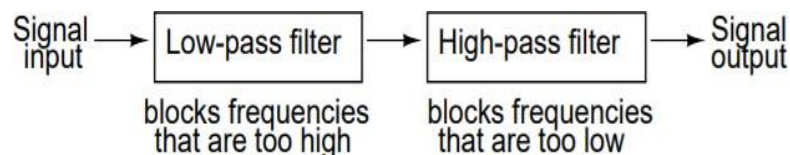
Gambar 2.4 Bentuk gelombang PPG model nonparametrik (Tjahjadi, Ramli and Murfi, 2020)

2.3 Filter

Dalam sebuah sinyal, terkadang hanya dibutuhkan beberapa bagian dari keseluruhan frekuensi atau bisa juga diartikan bahwa frekuensi yang dibutuhkan dalam sebuah sinyal hanya frekuensi tertentu saja. Untuk mengatasi hal ini dibutuhkan sebuah rangkaian yang dapat menyaring satu frekuensi atau rentang frekuensi tertentu secara selektif. Rangkaian yang dirancang untuk melakukan pemilihan frekuensi ini disebut filter (Tony R. Kuphaldt, 2007). Kegunaan filter yang paling utama adalah untuk menyingkirkan *noise* (gangguan) yang terdapat dalam suatu sinyal.

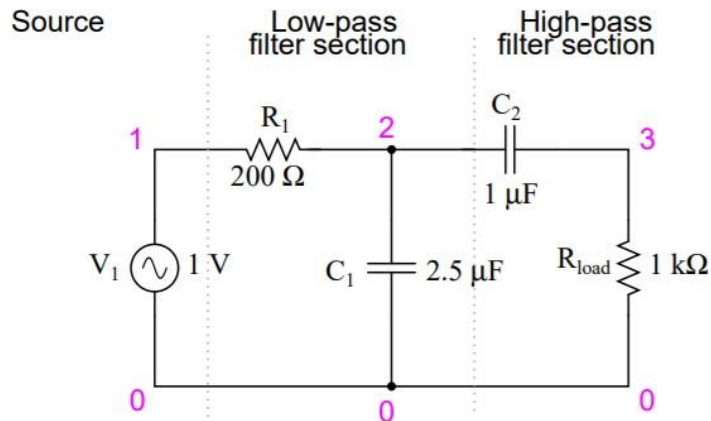
2.3.1 *Band Pass Filter (BPF)*

Pada suatu kondisi tertentu, terdapat masalah dimana frekuensi tertentu perlu dilakukan penyaringan dari kondisi sinyal aslinya. Filter yang dibutuhkan untuk menyelesaikan masalah ini, dapat dirancang dengan menggabungkan *low pass filter* dan *high pass filter*. Filter ini disebut *band pass filter* (BPF) (Tony R. Kuphaldt, 2007). Ilustrasi dari pembuatan BPF dapat digambarkan sebagai berikut:



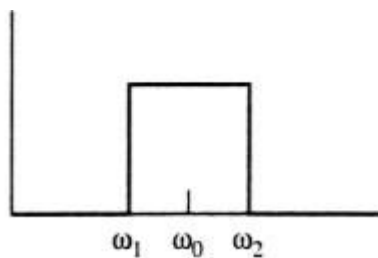
Gambar 2.5 Block Diagram *Band Pass Filter* (Tony R. Kuphaldt, 2007)

BPF dapat meloloskan sinyal dengan frekuensi yang tidak terlalu tinggi, namun juga frekuensi yang tidak terlalu rendah. Berikut merupakan contoh rangkaian BPF:



Gambar 2.6 Rangkaian *Band Pass Filter* (BPF) (Tony R. Kuphaldt, 2007)

Dalam BPF, pita frekuensi yang diinginkan dipusatkan di sekitar frekuensi 0 yang disebut frekuensi pusat. Pita frekuensi di bawah dan di atas passband disebut *stopbands*. Frekuensi di mana *passband* dan *stopband* bertemu disebut frekuensi bandedge. Plot magnitudo yang ideal dari *band pass filter* ditunjukkan sebagai plot pada Gambar 2.3 (Huelsman, 1993).



Gambar 2.7 Plot magnitude *Band Pass Filter* (BPF) (Huelsman, 1993)

2.4 Fitur Statistik

Ekstraksi fitur adalah sebuah metode utama dalam melakukan diagnosis, klasifikasi, pengelompokkan, pengenalan, dan deteksi. Metode ekstraksi fitur ini digunakan untuk memperoleh informasi sebanyak mungkin dari citra yang dianalisis. Yang menjadi tantangan adalah pemilihan dan efektifitas dari fitur yang akan diterapkan pada citra tersebut. Banyak metode yang digunakan untuk

mengekstrak fitur, yang mungkin bergantung pada fitur geometris, fitur statistik, fitur tekstur, dan fitur warna (K. Mutlag *et al.*, 2020).

Ekstraksi fitur dalam pengenalan pola didasarkan pada model matematis untuk mereduksi pola representatif. Representatif dimensi yang lebih rendah yang didasarkan pada pola deskriptif disebut fitur. Hal ini memainkan dalam menentukan metode pola pemisahan. Pemilihan fitur memiliki pengaruh penting pada beberapa hal, seperti akurasi klasifikasi, waktu klasifikasi, jumlah data, dan biaya klasifikasi (A. Meyer-Baese and Schmid, 2014).

Salah satu jenis ekstraksi fitur adalah dengan metode statistik. Dalam model data statistik, eksperimental dianggap sebagai distribusi statistik. Ketika eksperimen diulang, ada beberapa variasi acak dalam pengukuran yang tidak dapat dijelaskan dengan desain eksperimental. Hal inilah yang disebut dengan statistik (Nilsson, 2007). Fitur statistik merupakan konsep yang paling banyak digunakan dalam *data science*. Fitur ini biasanya merupakan teknik pertama yang akan diterapkan saat akan mengolah kumpulan data dan membahas hal – hal seperti bias, varians, mean, median, persentil, dan banyak lainnya.

2.4.1 Skewness (Kesimetrisan)

Fitur pertama yang digunakan dalam penelitian ini adalah *Skewness* (kesimetrisan). *Skewness* merupakan fitur untuk mengetahui tingkat simetris. Fitur ini tergolong fitur statistic karena memerlukan adanya nilai *mean*, *median*, dan *mode* sebagai pembanding tingkat simetris. Pada masa sekarang, penggunaan fitur *skewness* mengalami peningkatan khususnya ketika dalam memodelkan data. *Skewness* didefinisikan dengan nilai negative atau positif yang menandakan tingkat kesimetrisan relatif dari distribusi normal (Abdulah, 2019).

2.4.2 Peak Analysis

Fitur kedua yang digunakan adalah *peak analysis*. *Peak* merupakan keadaan yang disebut sebagai keadaan optimal puncak atau keadaan zona optimal. Atau secara umum, titik puncak dalam sinyal memiliki nilai tertinggi yang terletak pada

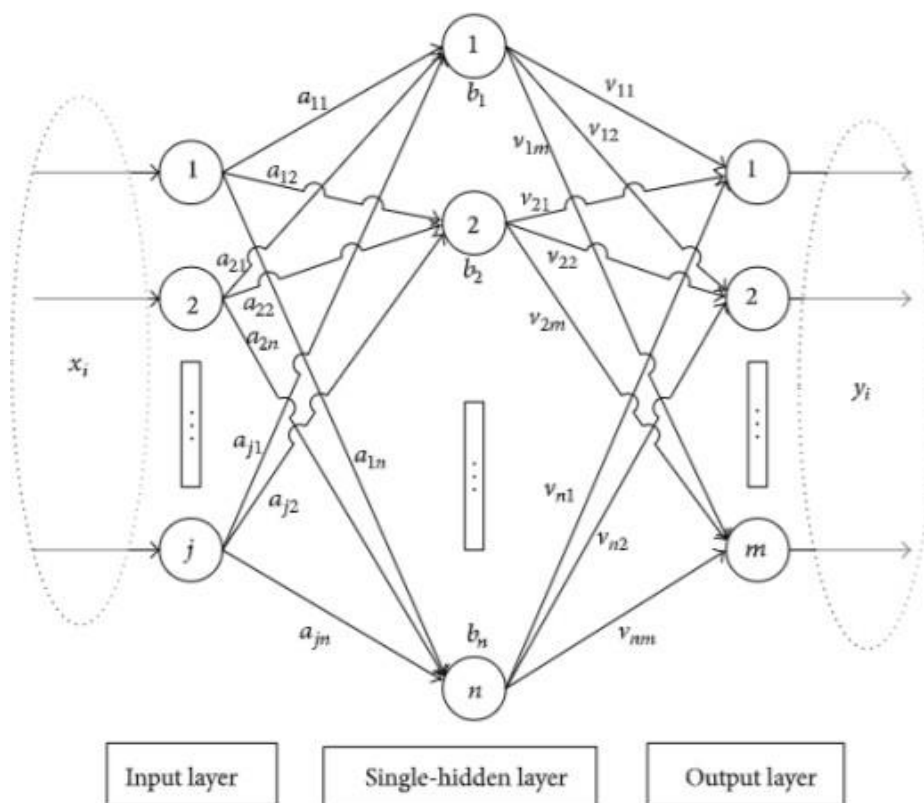
waktu dan lokasi tertentu. Namun, klasifikasi puncak hanyalah langkah pertama dalam mendeteksi peristiwa apa pun yang terkait dengan variasi sinyal seperti PPG, EEG, ECG, dan ECA (Adam *et al.*, 2019). Dalam analisis *peak*, terdapat dua fitur yang digunakan yaitu *prominence* dan *width*. Menurut *MathWorks*, *prominence* mengukur seberapa menonjol puncak tersebut karena ketinggian intrinsiknya dan lokasinya relatif terhadap puncak lainnya. Sedangkan *width* mengukur lebar dari *peak* tersebut.

2.5 *Extreme Learning Machine* (ELM)

Extreme Learning Machine (ELM) merupakan salah satu bagian dari jaringan saraf tiruan yang disebut dengan *single hidden layer feedforward neural network* (SLNs). Metode ini memiliki *learning speed* yang ribuan kali lebih cepat dibandingkan metode *feedforward* konvensional yang lain, seperti *Backpropagation* dan performansi generalisasi yang lebih baik (Ã, Zhu and Siew, 2006). ELM tidak melatih bobot input ataupun bias seperti jaringan saraf yang lain. Dengan menggunakan *node* yang memberikan nilai output maksimal, dan dengan parameter – parameter input *weight* dan bias yang dipilih secara random, ELM memiliki *learning speed* yang cepat dan *good generalization performance*.

Extreme Learning Machine (ELM) telah diusulkan sebagai paradigma yang sangat efisien dari *single hidden layer feedforward neural network* (SLFN). Menurut model ini, satu-satunya parameter yang dipelajari adalah bobot (*weight*) yang menghubungkan *hidden layer* dan output, sedangkan *hidden nodes* diinisialisasi secara acak yang kemudian diperbaiki tanpa penyetelan berulang (Mariniello *et al.*, 2021). Karena satu-satunya variabel bebas yang perlu dipelajari adalah hubungan antara *hidden layer* dan *hidden output* (bobot) dan apabila fungsi aktivasi terdiferensiasi tak terhingga, maka matriks *output hidden layer* dapat ditentukan dan akan memberikan pendekatan nilai target sebaik yang diinginkan. ELM sebagai model linear yang berujung pada pemecahan sistem linear. ELM sangat efisien dan cenderung mencapai optimal global. Studi teoritis telah

menunjukkan bahwa dengan hidden nodes yang dihasilkan secara acak, ELM mempertahankan kemampuan pendekatan SLFN yang universal (Tan, 2016).



Gambar 2.8 Struktur dari *Extreme Learning Machine* (Ding et al., 2015)

ELM dapat dimodelkan secara matematis dengan sederhana berdasar Gambar 2.7 sebagai berikut. Misalkan N adalah jumlah node dari *hidden layer* dan m mewakili banyaknya output layer dengan $g(x)$ merupakan fungsi aktivasi *hidden layer*. Untuk setiap N sampel yang berbeda $x_i \in R^N \times R^j$, $y_i \in R^N \times R^m$ ($i = 1, 2, \dots, N$) maka output dari *hidden layer* dapat ditunjukkan oleh persamaan (1). Kemudian untuk hubungan antara output hidden layere dengan output dari output layer dituliskan sebagaimana persamaan 2.

$$h = g(ax+b) \dots\dots\dots (1)$$

$$h(x_i)V = y_i, i = 1, 2, \dots, N \quad \dots\dots\dots (2)$$

Sehingga persamaan bisa ditulis menjadi

$$HV = Y \quad \dots\dots\dots (3)$$

Dimana

$$H = \begin{bmatrix} g(a_1, b_1, x_1) & g(a_1, b_1, x_2) & \dots & g(a_n, b_n, x_N) \\ g(a_2, b_2, x_1) & g(a_2, b_2, x_2) & \dots & g(a_n, b_n, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ g(a_n, b_n, x_1) & g(a_n, b_n, x_2) & \dots & g(a_n, b_n, x_N) \end{bmatrix} \dots\dots\dots (4)$$

$$V = \begin{bmatrix} F^T_1 \\ I \\ T^T \\ I^T_2 \\ \vdots \\ T^T \\ [^v_n]_{n \times m} \end{bmatrix}, Y = \begin{bmatrix} F^T_1 \\ I \\ T^T \\ I^T_2 \\ \vdots \\ T^T \\ [^y_n]_{N \times m} \end{bmatrix} \dots\dots\dots (5)$$

dimana $a_i = [a_{i1}, a_{i2}, \dots, a_{in}]^T$ adalah *weight* yang menjadi penghubung antara input node ke – i dengan *hidden layer*. b_j adalah bias dari *hidden node* ke j dan $v_j = [v_{j1}, v_{j2}, \dots, v_{jn}]^T$ adalah *weight* yang menghubungkan *hidden node* ke – j ke output layer (Ding *et al.*, 2015).

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan selama kurang lebih enam bulan di Laboratorium Instrumentasi Medis dan Laboratorium Komputer, Teknik Biomedis, Departemen Fisika, Fakultas Sains dan Teknologi, Universitas Airlangga.

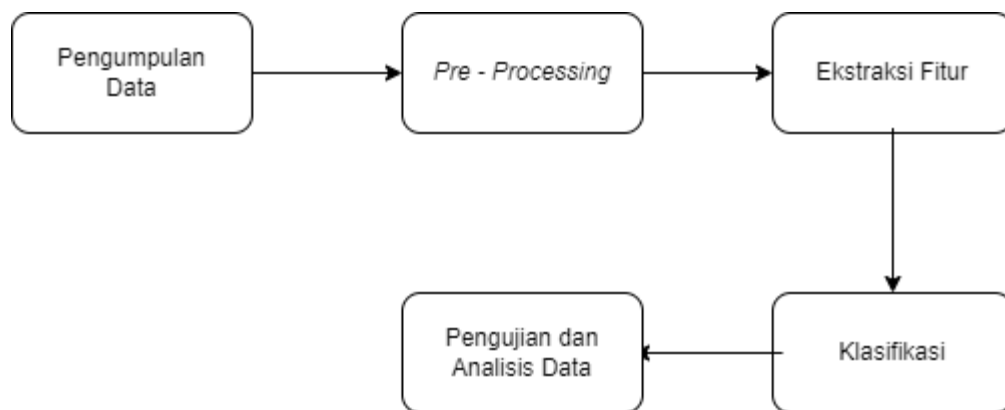
3.2 Alat dan Bahan Penelitian

Untuk menunjang pelaksanaan penelitian yang akan dilakukan, dibutuhkan alat dan bahan antara lain :

1. Laptop Asus dengan sistem operasi windows 11 dan RAM 16 GB
2. Software Matlab 2021a

3.3 Prosedur Penelitian

Prosedur penelitian terdiri atas beberapa tahapan. Secara garis besar digambarkan dengan diagram blok rancangan tahapan pada Gambar 3.1



Gambar 3.1 Diagram langkah penelitian

3.3.1 Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data sekunder yang

didapatkan dari pihak Komite Etik Pusat Medis Universiti Kebangsaan Malaysia dalam bentuk *raw* data. Data ini telah mendapat persetujuan tertulis yang diperoleh dari setiap subjek. Subjek data yang didapat pada penelitian ini adalah laki-laki dan perempuan dengan rentang usia 20 – 40 tahun yang sehat atau memiliki hipertensi.

Kriteria eksklusi untuk semua subjek dengan diabetes mellitus dan penyakit inflamasi kronis. Kondisi ini mungkin terkait dengan kerusakan vaskular lanjut dan dapat mengganggu skrining untuk lesi vaskular awal. Jumlah total naracoba adalah 57 orang, dengan naracoba sehat sebanyak 30 orang (7 laki-laki dan 23 perempuan) dan naracoba dengan hipertensi sebanyak 27 orang (3 laki-laki dan 24 perempuan).

Data yang dikumpulkan diambil menggunakan perekaman yang dilakukan dengan posisi naracoba berbaring terlentang pada suhu berkisar antara 20-25°C.

Subjek diminta untuk istirahat selama 5 menit dengan manset tekanan darah ditempatkan di lengan kanan. Tekanan darah brakialis diukur dengan menggunakan sistem Vicorder (*SMT Medical Wuerzburg*, Jerman). Sistem ini mampu mengubah bentuk gelombang brakialis menjadi bentuk gelombang aorta untuk memperkirakan tekanan darah sentral dengan menggunakan fungsi transfer umum brakialis ke aorta. Hasil pengukuran tekanan darah brakialis menggunakan sistem Vicorder ini nantinya akan digunakan sebagai acuan dalam menghitung preforma dari sistem berbasis *Artificial Intelligence* yang menjadi luaran pada penelitian ini.

Parameter yang digunakan dapat dilihat pada Tabel 3.1

Tabel 3.1 Tabel parameter pasien yang diukur

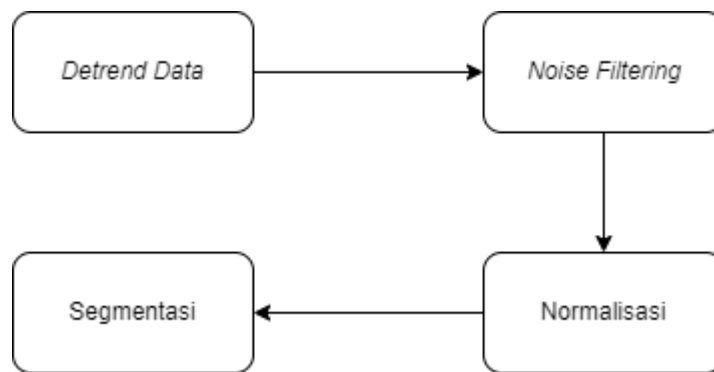
Parameters
Age (Tahun)
Weight (Kg)
Height (cm)
bSBP (mmHg)
bDBP (mmHg)

Heart Rate (bpm)

Pada saat yang sama dengan pengukuran tekanan darah brakialis, sebuah probe jari dilekatkan pada jari telunjuk kanan dan perekaman sinyal PPG dilakukan selama 2 menit menggunakan Port serial modul pulse-oximeter NiVARIx 1.0, Universiti Kebangsaan Malaysia. Sinyal PPG nantinya masih berupa *raw data*. Sinyal tersebut kemudian akan dilakukan *pre-processing* sebelum dilakukan proses analisis atau klasifikasi data.

3.3.2 Pre – Processing

Raw data sinyal PPG yang diperoleh dari tahap pengambilan data selanjutnya dilakukan *pre-processing*. Tahap *pre-processing* dilakukan seperti Gambar 3.2 berikut



Gambar 3.2 Tahapan *pre – processing*.

Detrend digunakan untuk menghilangkan trend yang muncul terhadap sinyal output dan input, sedangkan pemfilteran dilakukan untuk menghilangkan riak/*noise* yang ada pada sinyal. Filter yang digunakan untuk menghilangkan noise adalah *Band Pass Filter* (BPF) metode *Hamming* dengan rentang frekuensi 0.6-15 Hz. Dari hasil tersebut, nantinya sinyal yang bersih dari riak dan juga telah bersih dari gangguan sinyal DC akan dilakukan proses normalisasi untuk mengubah skala data dalam rentang 0 hingga 1.

Normalisasi data dalam penelitian ini akan menggunakan prinsip normalisasi min – max. Normalisasi ini dilakukan dengan cara mengurangi nilai pada tiap data

dengan nilai minimum dari data tersebut yang kemudian dibagi dengan selisih antara nilai maksimum dan nilai minimum data tersebut. Model matematika dari normalisasi dapat dilihat pada persamaan 3.1.

$$x = \frac{xold - xmin}{xmax - xmin} \dots\dots\dots (3.1)$$

Dimana x adalah nilai data hasil normalisasi, $xold$ adalah nilai data, $xmin$ adalah nilai minimum data, dan $xmax$ adalah nilai maksimum data. Setelah data dilakukan normalisasi, data akan dipotong menjadi tiap segmen. Dalam penelitian ini, segmentasi dilakukan berdasarkan pemotongan siklus titik puncak yang mengindikasikan titik sistol dan diastol dari sinyal PPG sehingga hasil segmentasi akan menunjukkan adanya satu siklus sistol dan diastol pada masing – masing segmen. Pemotongan siklus ini dilakukan dengan cara mendeteksi adanya *valley* pada sebelum sinyal sistol dan *valley* sesudah sinyal diastol.

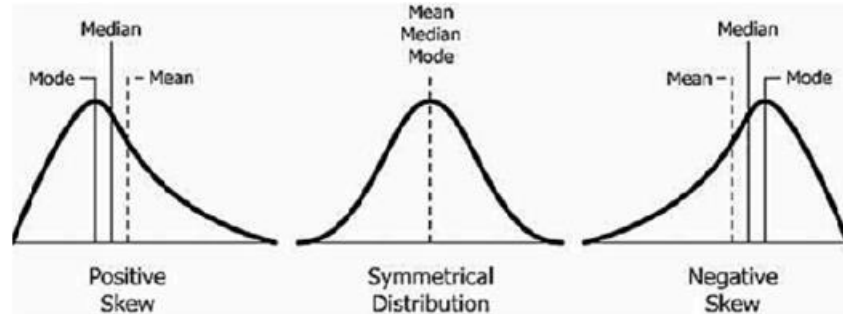
3.3.3 Ekstraksi Fitur

Ekstraksi fitur dilakukan untuk mendapatkan ciri dari hasil *pre-processing* sinyal sebagai nilai input untuk klasifikasi hipertensi. Setelah didapatkan sinyal PPG yang terbebas dari *noise*, langkah selanjutnya adalah melakukan segmentasi secara otomatis untuk mendapatkan satu segmen PPG. Segmen PPG ditandai dengan nilai minimal suatu sinyal. Sinyal PPG hasil segmentasi kemudian dipastikan apakah sudah termasuk sebagai *unfitted signal* atau bukan.

Selanjutnya akan dicari fitur berdasarkan domain waktu. Fitur pada domain waktu diantaranya adalah T_s (*systolic upstroke time*), T_d (*diastolic time*) serta mean untuk menentukan kesimetrisan satu segmen (*skewness signal quality*). Apabila distribusi nilainya simetris maka *skewness* akan bernilai mendekati nol. Visualisasi fitur yang digunakan dalam domain waktu terlihat pada Gambar 3.3. Fitur – fitur ini akan digunakan sebagai fitur untuk AI untuk mengklasifikasikan normal dan hipertensi. Model matematika dari fitur ini dapat dituliskan seperti yang tertulis pada persamaan 3.2.

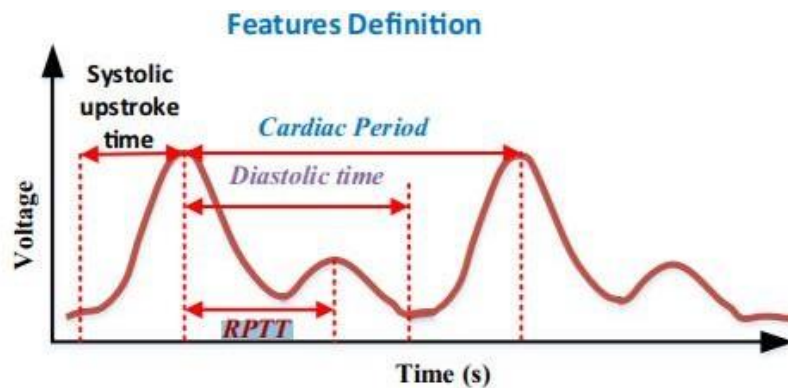
$$S_{SQI} = \frac{\sum_{i=1}^N (A_i - \tilde{A})^3}{(N-1)\sigma^3} \dots\dots\dots (3.2)$$

Dimana S_{SQI} adalah indeks kualitas sinyal *skewness*, N adalah jumlah variable dalam distribusi, σ adalah standar distribusi, A_i adalah variable acak, dan \tilde{A} adalah rata – rata distribusi.



Gambar 3.3 Fitur *skewness*

Fitur selanjutnya yang digunakan dalam penelitian ini adalah fitur *peak analysis*. Dalam *peak analysis* akan dicari dimana letak puncak yang mengindikasikan sistolik dan diastolik, *width* dari puncak sistolik dan diastolik, serta *prominence* dari puncak sistolik dan diastolik. Fitur ini nantinya dapat digunakan untuk mengetahui nilai *peak* yang mengindikasikan *systolic peak* dan *diastolic peak*. *Prominence* sendiri nantinya dapat membantu untuk mengetahui nilai RPTT (Gambar 3.4).



Gambar 3.4 Fitur *Peak Analysis*.

3.3.4 Tahap Klasifikasi

Setelah melakukan ekstraksi fitur, tahap selanjutnya adalah tahap klasifikasi. Metode yang digunakan untuk *Classifier* pada penelitian ini adalah ELM karena metode ini memiliki kelebihan dalam meningkatkan hasil akurasi dan waktu pelatihan yang cepat (Ä, Zhu and Siew, 2006). Metode ini memiliki dua proses didalamnya, yaitu proses *training* dan proses *testing*.

1. Proses *Training*

- Data pelatihan dari hasil ekstraksi fitur diinputkan. Data tersebut berupa nilai *skewness* dan *prominence*.
- Data pelatihan dinormalisasi pada rentang 0 sampai 1.
- Menginisiasi bobot awal (W_{jk}) dan bias secara acak dengan ukuran matriks $j \times k$, dimana j adalah banyak *input* node dan k adalah banyak *hidden* node.
- Menghitung output matriks *hidden layer* (H) dengan fungsi aktivasi sigmoid untuk memetakan nilai pada interval 0 sampai 1 dengan persamaan (3.3)

$$H = \frac{1}{(1 + e^{-(X \cdot W^T + \text{ones}(N_{\text{train}}, 1) * b))}} \dots \dots \dots (3.3)$$

Dimana :

H = Output *hidden layer*

X = Matriks input data

b = Bias

W^T = Matriks transpose dari input bobot

$\text{Ones}(N_{\text{training}}, 1)$ = Matriks dengan ukuran baris yang sama dengan jumlah data training dan 1 kolom dengan semua elemen yang bernilai 1.

- Menghitung matriks moore – penrose pseudo. Perhitungan Moore - Penrose generalized inverse ditunjukkan pada persamaan (3.4)

$$H^+ = (H^T \cdot H)^{-1} \cdot H^T \dots \dots \dots (3.4)$$

Dimana :

H^+ = Inverse matriks Moore – Penrose pseudo

H = Matriks output *hidden layer*

H^T = Transpose matriks output *hidden layer*

- Menghitung output *weight* (β) yang dihasilkan oleh *hidden layer* dan output layer menggunakan persamaan (3.5)

$$\beta = H^+ \cdot Y \dots\dots\dots (3.5)$$

Dimana :

β = Output *weight*

H^+ = Inverse matriks Moore – Penrose pseudo

Y = Matriks target

- Menghitung hasil prediksi (\hat{Y}) yang didapat melalui proses perkalian matriks output *hidden layer* dengan output *weight* menggunakan persamaan (3.6)

$$\hat{Y} = H \cdot \beta \dots\dots\dots (3.6)$$

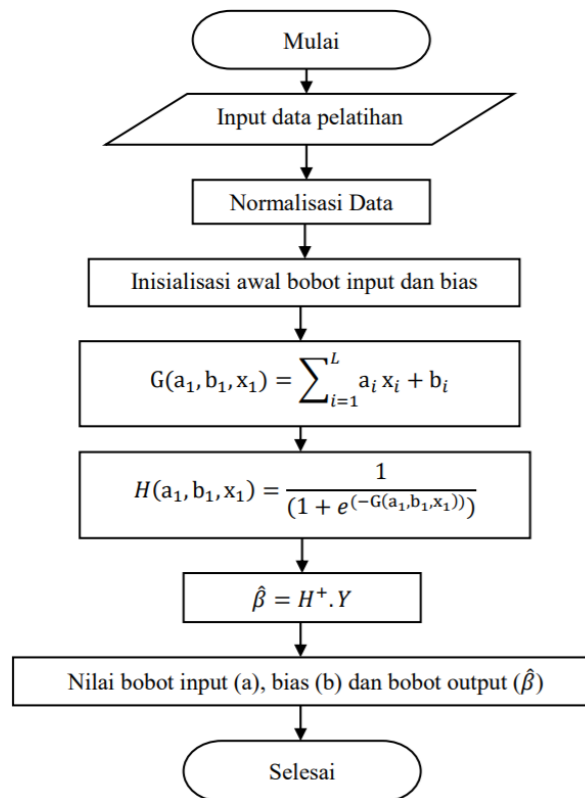
Dimana :

\hat{Y} = Hasil prediksi

H = Matriks output *hidden layer*

B = Matrteks output *weight*

Diagram dari proses *training* dapat digambarkan pada Gambar 3.5

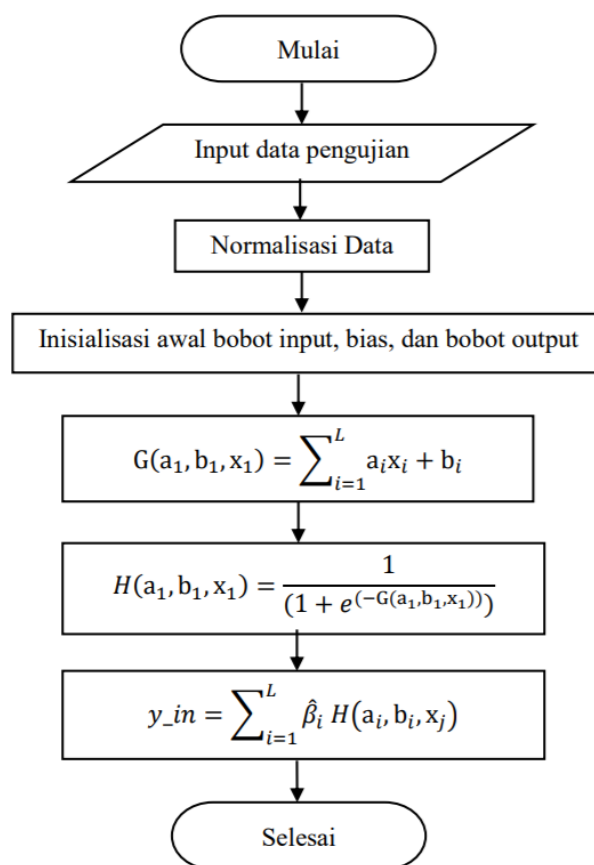


Gambar 3.5 Diagram proses *training*.

2. Proses *testing*

- Menginput data pengujian
- Menginput data hasil normalisasi
- Melakukan *setting* input *weight* (W_{jk}), bias, dan output *weight* (β) yang didapat dari proses *training* yang telah dilakukan.
- Menghitung matriks output *hidden layer* (H) menggunakan persamaan (3.2)
- Menghitung hasil prediksi (\hat{Y}) menggunakan persamaan (3.6)
- Menghitung nilai dari output target

Diagram dari proses *testing* dapat digambarkan pada Gambar 3.6



Gambar 3.6 Diagram proses *testing*.

3.3.5 Tahap Pengujian dan Analisis Data

Tahapan ini dilakukan untuk mengetahui fitur apa yang memiliki tingkat akurasi yang tinggi untuk digunakan pada data pengambilan menggunakan metode ELM dan fitur apa yang memiliki tingkat keefektifan yang baik dalam mengklasifikasikan hipertensi. Fitur yang akan digunakan yaitu *skewness*, *peak analysis*, dan kombinasi *skewness – peak analysis*. Akurasi akan didapatkan dari nilai rata – rata yang kemudian akan dilakukan pengulangan sebanyak tiga kali sesuai dengan banyak fitur yang digunakan.

Selain dari akurasi, data dapat dievaluasi sensitifitas dan spesifisitasnya. Sensitifitas adalah metrik yang mengevaluasi kemampuan model untuk memprediksi hal positif yang sebenarnya dari setiap kategori yang tersedia.

Spesifisitas adalah metrik yang mengevaluasi kemampuan model untuk memprediksi negatif sebenarnya dari setiap kategori yang tersedia. Metrik ini berlaku untuk model kategoris apa pun. Rumus untuk menghitung sensitifitas dituliskan dalam persamaan 3.7 dan rumus untuk menghitung spesifisitas dituliskan dalam persamaan 3.8

$$\text{Sensitifitas} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \dots\dots\dots (3.7)$$

$$\text{Spesifisitas} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positives}} \dots\dots\dots (3.8)$$

BAB IV

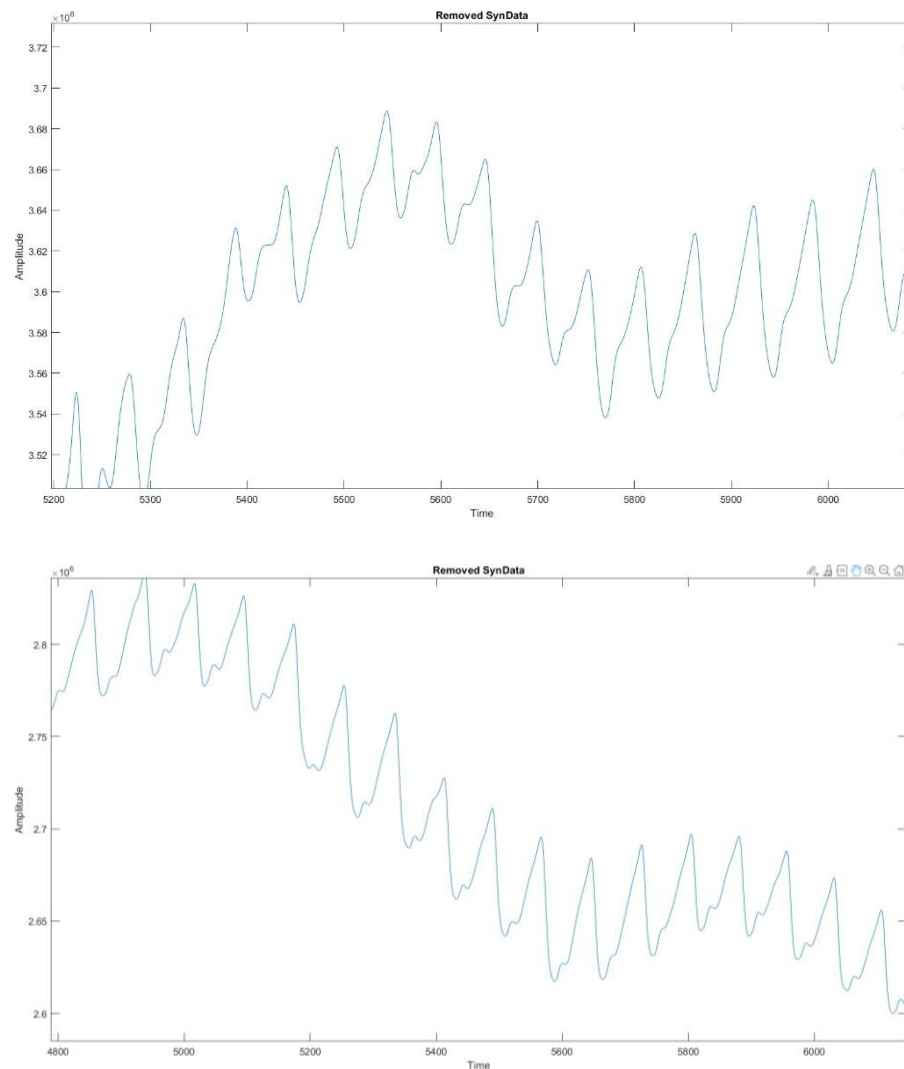
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Data penelitian ini diperoleh dari Komite Etik Pusat Medis Universiti Kebangsaan Malaysia dalam bentuk *raw* data dalam format (.mat), dimana untuk data normal sebanyak 30 subjek dan untuk hipertensi sebanyak 27 subjek. Subjek yang diambil akan dilihat juga parameter yang lain sesuai pada Tabel 3.1 sebagai syarat pengambilan data yang dilakukan oleh pakar. Hasilnya, data yang didapatkan memiliki *range* untuk masing – masing parameter sesuai dengan Tabel 4.1. Data tersebut kemudian dipanggil dalam matlab melalui program ppgtb.mlx dan terlihat seperti pada Gambar 4.1.

Tabel 4.1 *Range* parameter dari subjek

Parameter	Normal	Hipertensi
Age (Tahun)	26 – 43	22 – 59
Weight (Kg)	41 – 68.9	46 – 65.6
Height (cm)	150 – 176	147 – 175
bSBP (mmHg)	110 – 139	140 – 200
bDSP (mmHg)	61 – 89	61 – 124
Heart rate (bpm)	50 – 80	55 – 83



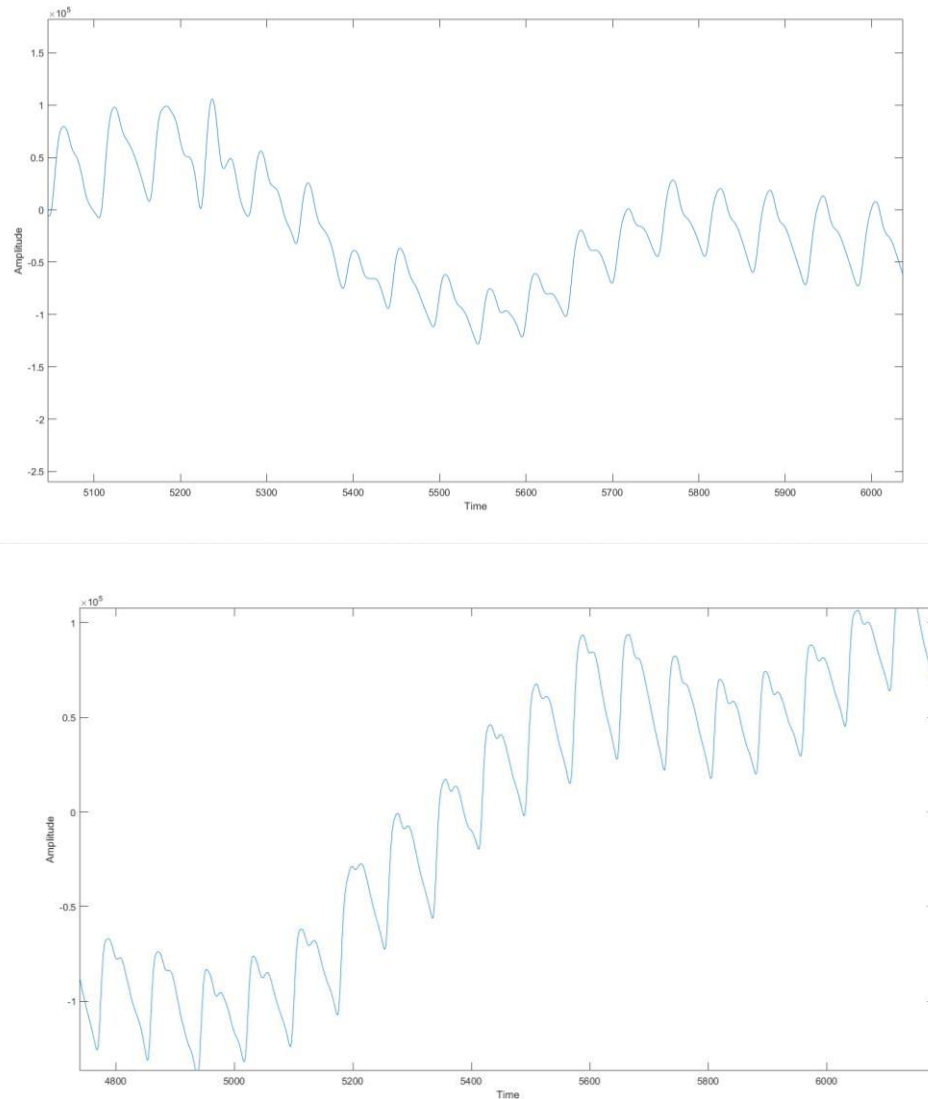
Gambar 4.1 Data subjek normal (atas) dan hipertensi (bawah) yang diambil dalam bentuk *raw*.

Setelah data tersebut sudah dikumpulkan, data akan diolah melalui empat tahap *pre – processing*, yaitu *detrend* data, *noise filtering*, normalisasi, dan segmentasi.

4.2 *Pre – processing*

Pre – processing merupakan tahapan yang bertujuan untuk mempermudah pembacaan sinyal. Pada penelitian ini, perlakuan *pre – processing* yang pertama adalah *detrend* data. Sebelum dilakukan *detrend*, data dipanggil terlebih dahulu dengan cara

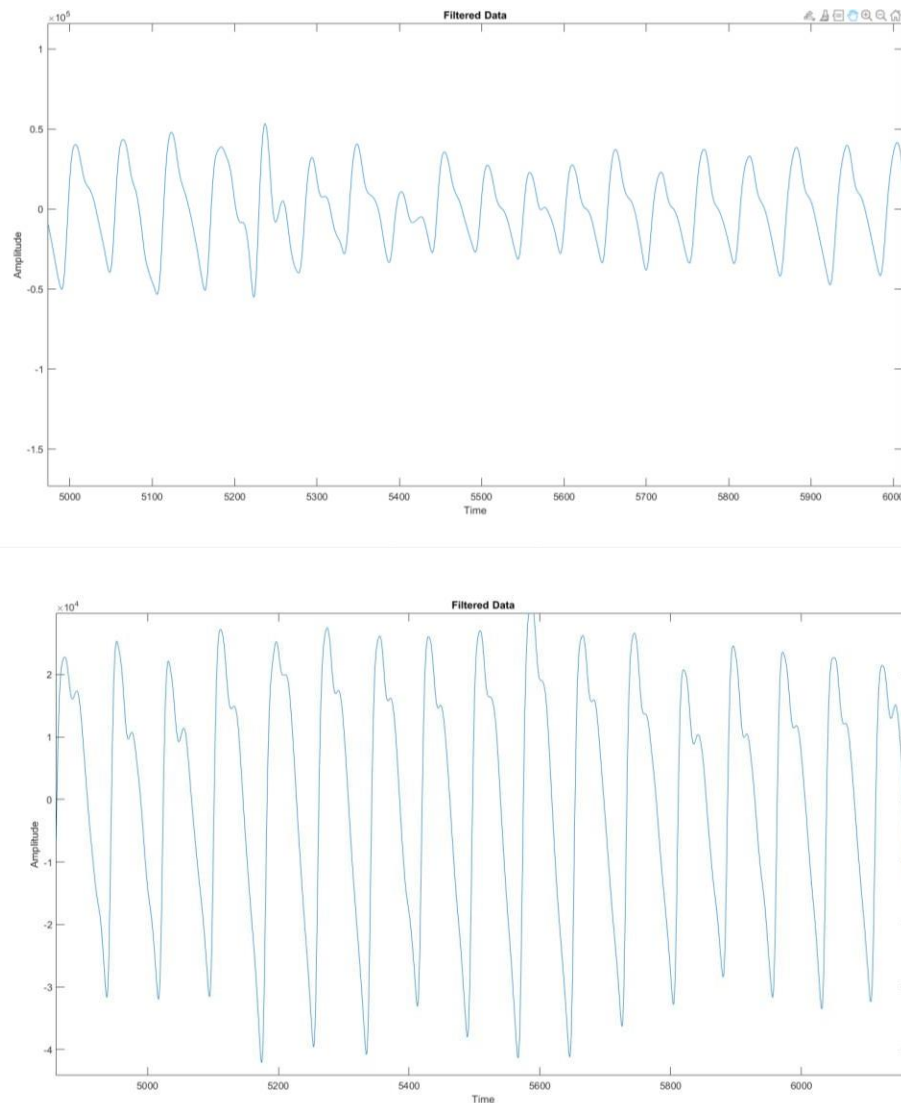
melakukan *load* pada data yang sudah disimpan dalam folder. *Detrend* data ini bertujuan untuk menghilangkan trend yang muncul pada sinyal atau bisa disebut *baseline wander* yang terjadi akibat adanya frekuensi kecil yang muncul, seperti detak jantung. Hasil dari *detrend* data dapat dilihat pada Gambar 4.2.



Gambar 4.2 Sinyal data normal (atas) dan hipertensi (bawah) setelah dilakukan *detrend* data.

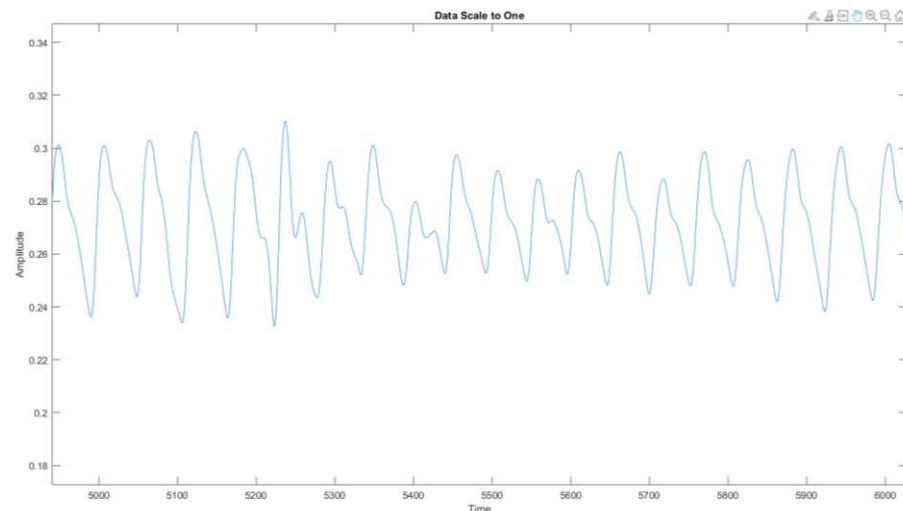
Setelah melakukan *detrend*, noise sinyal data akan diproses dengan menggunakan *Band Pass Filter* (BPF) metode *Hamming* yang memiliki frekuensi diantara 0.6 – 15

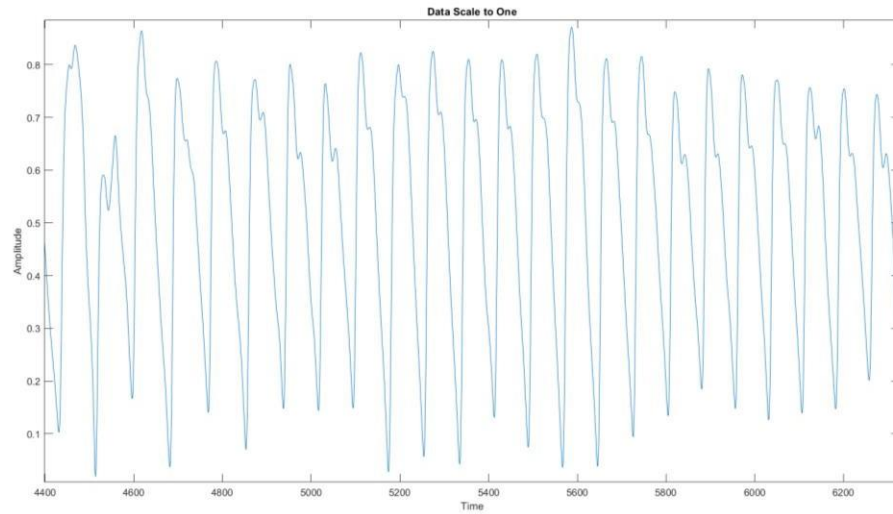
Hz. Frekuensi sampling yang digunakan untuk BPF ini adalah 100 Hz. Filter BPF dibuat melalui fungsi yang disimpan dalam program dengan nama BPF.m. Pemilihan metode *Hamming window* didasarkan pada lebar lobusnya yang relatif sempit. Selain itu, *Hamming* juga memiliki lebar atenuasi yang cukup baik sehingga cocok untuk digunakan pada sinyal PPG. Informasi yang didapat dalam sinyal PPG terletak pada frekuensi diantara 0.05 – 20 Hz (Madona, 2016). Oleh karenanya, penentuan frekuensi dengan rentang 0,6 – 15 Hz diakibatkan adanya *noise* dari berbagai hal, seperti detak jantung ataupun *electrical noise*. Hasil dari pemfilteran dengan BPF dapat dilihat pada Gambar 4.3.



Gambar 4.3 Data sinyal normal (atas) dan hipertensi (bawah) setelah difilter dengan BPF.

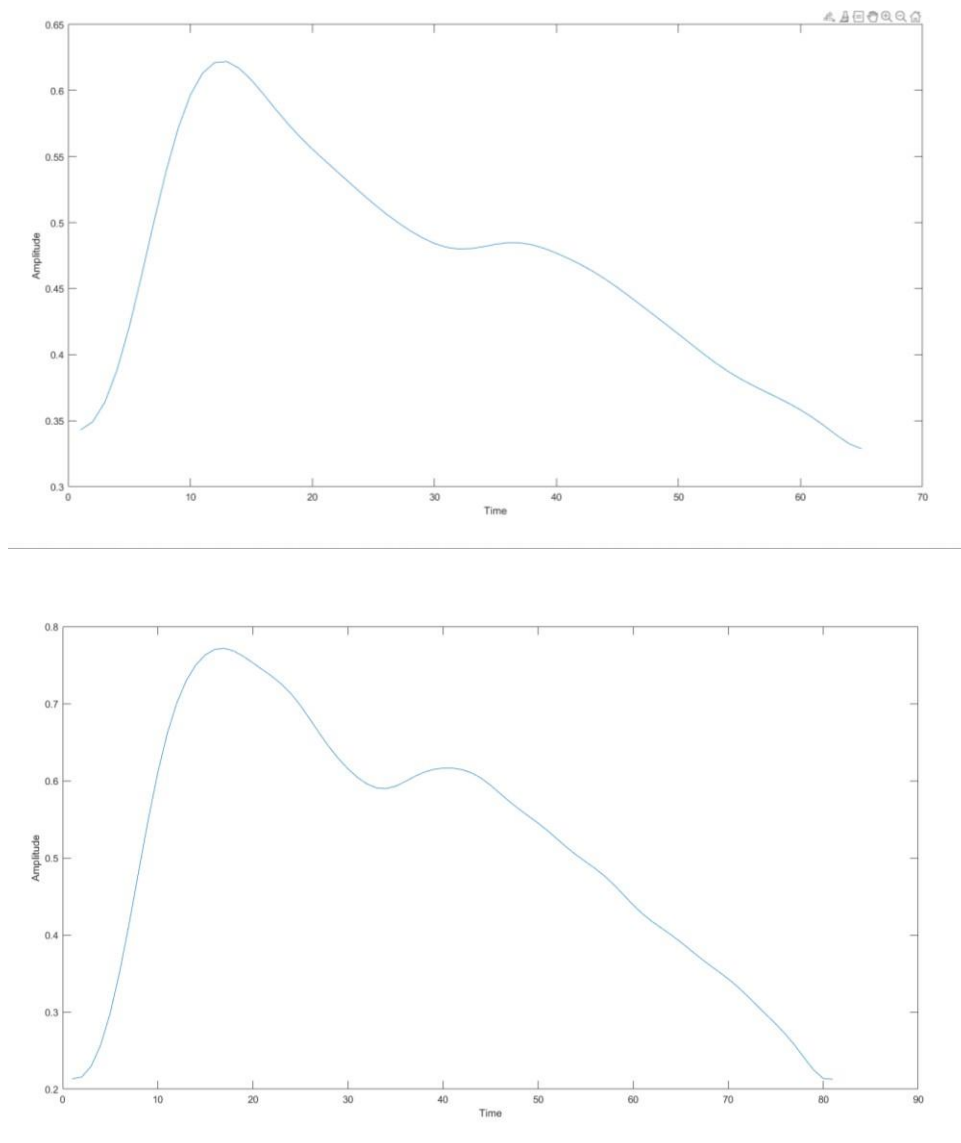
Kemudian, *pre - processing* dilanjutkan dengan tahap normalisasi dengan tujuan agar skala sinyal berada pada range antara 0 – 1. Normalisasi mempermudah pembacaan data sehingga dalam pengaplikasian fitur akan terlihat lebih jelas dibandingkan sebelum dinormalisasi. Pada penelitian ini, normalisasi dilakukan dengan menghitung selisih antara nilai sinyal tertinggi dan terendah yang dilambangkan dengan PP_PPG1. Kemudian, nilai sinyal masing – masing akan dikurangi dengan nilai minimal pada sinyal. Hasilnya akan dibagi dengan hasil PP_PPG1. Perhitungan untuk normalisasi dan hasil dari normalisasi ditampilkan pada Gambar 4.4.





Gambar 4.4 Sinyal hasil normalisasi pada data normal (atas) dan hipertensi (bawah)

Setelah mendapatkan data yang sudah dinormalisasi, data akan dipotong menjadi bagian satu segmen atau bisa disebut segmentasi. Segmentasi ini bertujuan agar data dapat terlihat dengan jelas bentuk sinyalnya sehingga fiturnya dapat diketahui nilai puncak yang mengindikasikan titik sistol dan diastolnya. Segmentasi sinyal dilakukan melalui dua tahapan dimana yang pertama dilakukan secara otomatis melalui code pada bagian pendeteksian *valley* dan pemotongan pulsa per satu segmen (*single pulse plot*). Pendeteksian *valley* dilakukan dengan menghitung berapa banyak lembah yang ada. Disini pendeteksian segmen didasari pada prinsip bahwa satu segmen diapit dengan dua lembah. Diawali dengan melakukan *looping* nilai $k = 2$ hingga sepanjang $(\text{sig}) - 1$. Inisiasi k dimulai dari dua dikarenakan pendeteksian puncak segmen berada diantara dua lembah, sehingga puncak akan terdeteksi apabila nilai k yang sekarang lebih kecil dari k sebelumnya dan lebih kecil dari setelahnya, maka sinyal inilah yang dapat dianggap sebagai puncak. *Valley* yang ditebak harus memiliki *treshold* pada nilai tertentu agar titik *valley* memiliki standard ketinggian yang sama. Pada penelitian ini, *treshold valley* yang digunakan adalah kurang dari 0.42. Setelah itu, hasilnya dikumpulkan dalam variabel yang diberi nama “segment”. Segment inilah yang kemudian disimpan satu per satu sebagai sinyal input untuk pengklasifikasi dengan ELM nantinya. Hasil segmentasi sinyal dapat dilihat pada Gambar 4.5 berikut.



Gambar 4.5 Hasil segmentasi sinyal per segmen untuk data normal (atas) dan hipertensi (bawah).

4.3 Ekstraksi Fitur

Hasil yang didapatkan dari proses segmentasi selanjutnya akan diproses melalui ekstraksi fitur seperti yang sudah dijelaskan pada subbab 3.3.3. Ekstraksi fitur yang digunakan dalam penelitian ini adalah fitur statistik berupa *peak analysis* dan *skewness*. Pada matlab, fitur *peak analysis* dapat dicari dengan cara menghitung nilai *peak*. Matlab memiliki sebuah fungsi *Findpeaks* yang dapat digunakan untuk mencari sebuah

peak dari sebuah sinyal. Dalam matlab, sebuah *peak* dari sinyal bisa dianggap *peak* apabila memiliki nilai lebih tinggi dari nilai sebelumnya dan juga dari nilai setelahnya. Pada penelitian ini, nilai puncak dapat memberikan empat informasi. Namun, untuk penggunaan dalam fitur hanya akan digunakan dua bagian. Empat informasi yang didapatkan dalam fitur *Findpeaks* adalah letak *peak* terhadap sumbu x, letak *peak* terhadap sumbu y, *prominence* dan *width*. Keempat informasi ini dilambangkan dengan variabel yang diapit dengan kurung siku '[pks, locs, widths, proms]' secara berurutan. Dari keempat informasi ini, yang digunakan untuk fitur adalah *widths* dan *proms*. Data informasi dari fitur *findpeaks* yang didapat terlihat pada Tabel 4.2 untuk normal dan 4.3 untuk hipertensi.

Tabel 4.2 Tabel hasil informasi data normal yang didapat dari fitur *findpeaks*.

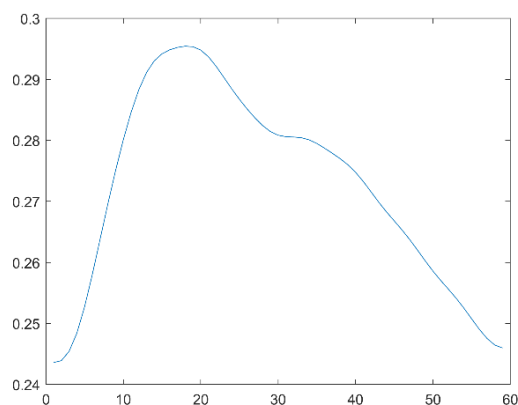
Segmen	Fitur <i>Peak</i>			
	<i>pks</i>	<i>locs</i>	<i>widths</i>	<i>proms</i>
1	0.7994	16	39.5057	0.5687
2	[0.7717;0.6163]	[17;40]	[47.1357;7.3918]	[0.5584;0.0266]
3	[0.7421;0.6448]	[16;38]	[47.3751;5.4464]	[0.5294;0.0228]
4	[0.7335;0.6529]	[21;39]	[51.2441;3.5055]	[0.5672;0.0042]
5	[0.7495;0.6716]	[19;40]	[50.5033;5.4562]	[0.5832;0.0116]
6	0.7220	20	49.9999	0.5043
7	[0.7135;0.6303]	[18;38]	[48.9307;5.1423]	[0.4919;0.0106]
...

Tabel 4.3 Tabel hasil informasi data hipertensi yang didapat dari fitur *findpeaks*.

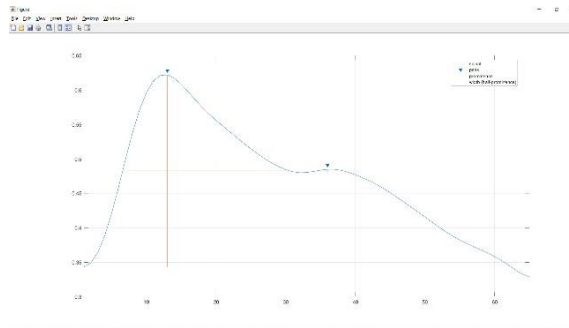
Segmen	Fitur <i>Peak</i>			
	<i>pks</i>	<i>locs</i>	<i>widths</i>	<i>proms</i>
1	0.7994	16	39.5057	0.5687
2	[0.7717;0.6163]	[17;40]	[47.1357;7.3918]	[0.5584;0.0266]
3	[0.7421;0.6448]	[16;38]	[47.3751;5.4464]	[0.5294;0.0228]

4	[0.7335;0.6529]	[21;39]	[51.2441;3.5055]	[0.5672;0.0042]
5	[0.7495;0.6716]	[19;40]	[50.5033;5.4562]	[0.5832;0.0116]
6	0.7220	20	49.9999	0.5043
7	[0.7135;0.6303]	[18;38]	[48.9307;5.1423]	[0.4919;0.0106]
...

Selain itu, pada matlab tinggi *peak* yang ingin dideteksi dapat ditentukan batasnya dengan menginputkan fungsi ‘MinPeakHeight’. Pada penelitian ini, nilai batas yang digunakan adalah 0.2. Nilai ini didasarkan pada adanya sinyal yang memiliki *peak* sangat rendah, salah satunya seperti pada Gambar 4.6. Pada *peak analysis* sendiri akan diambil dua buah nilai, yaitu *prominence* dan *width* yang kemudian hasilnya disimpan dalam bentuk .mat untuk masing – masing segmen sinyal. Nilai *prominence* dan *width* yang diambil didasarkan pada pengukuran data oleh matlab seperti pada Gambar 4.7.

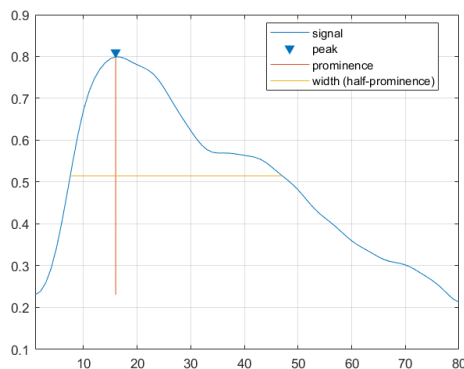


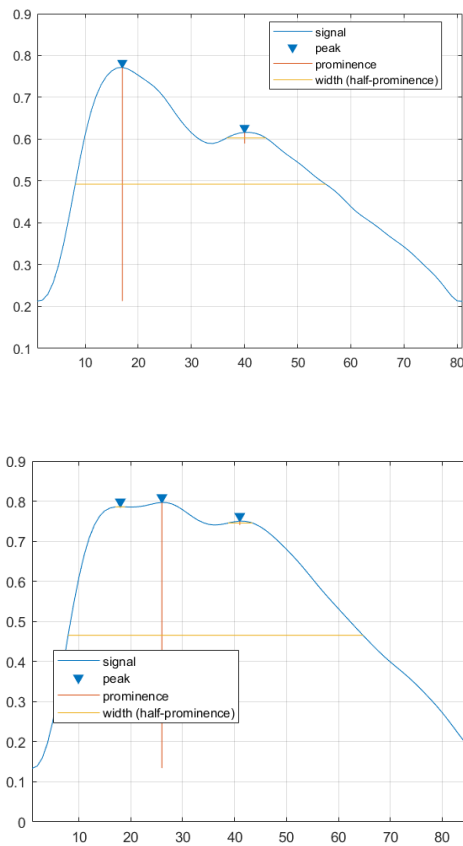
Gambar 4.6 Contoh data yang memiliki *peak* rendah.



Gambar 4.7 Pengambilan fitur *prominence* dan *width* pada matlab.

Dari proses pengambilan fitur *peak analysis* ini, didapati bahwa ada sinyal data yang tidak terdeteksi puncak titik keduanya sehingga nilai puncak yang didapati hanya puncak pertama saja. Hal ini disebabkan karena pendeteksian titik puncak oleh matlab didasari pada adanya turunan (lembah) sebelum sinyal mengalami kenaikan sehingga untuk mengatasinya data baris kedua diisi dengan nilai 0 agar data memiliki kesesuaian dengan data yang mendeteksi puncak yang mengindikasikan puncak sistol dan diastol, yaitu berbentuk matriks 2x1. Selain itu, pada data didapati pula sinyal yang terdeteksi memiliki tiga puncak yang disebabkan karena adanya data yang kurang bagus sehingga menyebabkan matriks *prominence* dan *width* memiliki ukuran 3x1. Dikarenakan data yang dibutuhkan terletak pada baris kedua dan ketiga yang mengindikasikan puncak untuk sinyal sistol dan diastolnya, maka dilakukan pengambilan data hanya pada baris dua dan tiga.





Gambar 4.8 Data yang terdeteksi memiliki 1 *peak* (atas), 2 *peak* (tengah), dan 3 *peak* (bawah).

Selanjutnya, untuk pengambilan fitur *skewness*, matlab sudah memiliki *syntax* yang dapat mengetahui nilai *skewness* secara langsung. Dengan memanggil fungsi ‘*skewness*’ secara otomatis nilai *skewness* dari data tersebut bisa langsung diketahui. Nilai *skewness* ini mewakili kondisi dari *mean*, *mode*, dan *median* seperti yang dijelaskan pada subbab 3.3.3. Sama seperti *peak analysis*, hasil dari fitur *skewness* yang didapatkan akan disimpan dalam bentuk *.mat* untuk tiap segmen sinyal. Hasil dari nilai *skewness* ini dapat dilihat pada Tabel 4.4 berikut.

Tabel 4.4 Data nilai *skewness* yang didapatkan untuk sinyal hipertensi dan normal

Segmen	Nilai <i>Skewness</i>	
	Hipertensi	Normal

1	0.1429	-0.0492
2	-0.2534	-0.2989
3	-0.3958	-0.2456
4	-0.4629	-0.1304
5	-0.4055	-0.1955
6	-0.4856	-0.2355
7	-0.3521	-0.2137
...

Setelah nilai masing – masing fitur didapatkan, fitur akan dipisah menjadi bagian *training* dan *testing* untuk *input Extreme Learning Machine* (ELM). Pembagian untuk *training* dan *testing* adalah sebesar 70% untuk *training* dan 30% untuk *testing* untuk tiap – tiap data normal dan hipertensi. Sehingga, untuk *training* didapatkan 1115 data normal dan 1110 data hipertensi. Sedangkan untuk *testing* didapatkan sebanyak 474 data normal dan 469 data hipertensi. Pemisahan data antara *training* dan *testing* hanya didasarkan pada urutan data saja. Setelah memisahkan data untuk *training* dan *testing*, data dari masing – masing kondisi normal dan hipertensi untuk *training* akan digabungkan. Perlakuan yang sama juga diberlakukan untuk data *testing*.

Nilai yang didapatkan pada proses *training* pada fitur *prominence* memiliki rata – rata sebesar [0.4964 0.0041] untuk data normal dan [0.4901 0.0132] untuk data hipertensi serta nilai rata – rata *training* pada fitur *width* adalah sebesar [43.2547 1.8970] untuk data normal dan untuk data hipertensi [43.7305 2.3913]. Sedangkan nilai rata – rata fitur *prominence* pada data *testing* sebesar [0.7182 0.0178] untuk data normal dan [0.5696 0.1958] untuk data hipertensi. Sedangkan rata – rata nilai untuk fitur *width* sebesar [32.2264 3.6301] pada data normal dan pada hipertensi sebesar [34.7565 12.2670]. Dapat dilihat bahwa nilai *prominence* normal lebih tinggi dibandingkan dengan nilai *prominence* hipertensi. Namun, sebaliknya untuk nilai *width* pada hipertensi memiliki rata – rata lebih tinggi dibandingkan pada data normal. Hal

ini disebabkan karena adanya masalah pada pemilahan data sehingga data yang didapatkan untuk klasifikasi tergolong kurang bagus.

Pada fitur *skewness* sendiri, didapatkan nilai rata – ratanya untuk *training* -0.1913 pada normal dan -0.2444 pada hipertensi. Sedangkan rata – rata nilai untuk *testing* didapat -0.0173 pada data normal dan pada data hipertensi sebesar -0.1214. Nilai negatif ini menunjukkan bahwa sinyal data yang didapat tidak simetris sesuai dengan subbab 3.3.3. Dari data didapatkan bahwa nilai hipertensi semakin jauh dari nilai 0 sehingga data hipertensi lebih tidak simetris dibandingkan data normal (Tjahjadi and Ramli, 2020).

Hasil dari proses ekstraksi fitur ini, akan didapatkan tiga buah file dalam bentuk .mat yang masing – masing terdiri dari *prominence.mat*, *width.mat*, dan *skewness.mat* untuk masing – masing sinyal data normal dan hipertensi. Dari ketiga file ini nantinya akan diuji menjadi tiga tahap. Tahap yang pertama adalah menggunakan fitur *skewness*, yang kedua adalah fitur *peak analysis* yang merupakan gabungan dari nilai fitur *prominence* dan *width*, dan yang terakhir adalah gabungan dari ketiganya. Setelah itu, masing – masing file fitur yang telah dikombinasikan ini diberi pelabelan. Untuk parameter normal diberi label ‘1’ sedangkan untuk parameter hipertensi diberi label ‘2’. Ketiga parameter ini lah nanti yang digunakan sebagai input untuk menguji fitur manakah yang memiliki akurasi tertinggi untuk digunakan dalam klasifikasi menggunakan *Extreme Learning Machine* (ELM).

Tabel 4.5 Data fitur *skewness* sebagai masukan ELM setelah pelabelan

Segmen	Label	Nilai <i>Skewness</i>
1	1	-0.1292
2	1	-0.1088
3	1	-0.1591
4	1	-0.1530
5	1	-0.1620
6	1	-0.1623

7	1	-0.2781
...
475	2	-0.2237
476	2	-0.2329
477	2	-0.1820
478	2	-0.2166
479	2	-0.2364
480	2	-0.1818
481	2	-0.2433
...

Tabel 4.6 Data fitur *peak analysis* sebagai masukan ELM setelah pelabelan

Segmen	Label	<i>Prominence</i>		<i>Width</i>	
		<i>peak 1</i>	<i>peak 2</i>	<i>peak 1</i>	<i>peak 2</i>
1	1	0.7616	0.0051	41.6906	2.9789
2	1	0.7413	0.0059	41.0277	3.1805
3	1	0.7982	0.0002	42.4938	0.6285
4	1	0.7790	0.0002	42.4965	0.6066
5	1	0.7068	0.0051	41.9592	2.9515
6	1	0.7670	0.0014	41.9592	1.5728
7	1	0.6424	0	48.4616	0
...
475	2	0.8433	0	41.2911	0
476	2	0.8125	0	40.7132	0
477	2	0.8095	0	39.8384	0
478	2	0.8561	0	41.8585	0
479	2	0.8182	0	40.2054	0
480	2	0.8160	0	39.6234	0

481	2	0.8573	0	41.6891	0
...

Tabel 4.7 Data fitur gabungan *skewness* dan *peak analysis* sebagai masukan ELM setelah pelabelan

Segmen	Label	Skewness	Prominence		Width	
			<i>peak 1</i>	<i>peak 2</i>	<i>peak 1</i>	<i>peak 2</i>
1	1	-0.1292	0.7616	0.0051	41.6906	2.9789
2	1	-0.1088	0.7413	0.0059	41.0277	3.1805
3	1	-0.1591	0.7982	0.0002	42.4938	0.6285
4	1	-0.1530	0.7790	0.0002	42.4965	0.6066
5	1	-0.1620	0.7068	0.0051	41.9592	2.9515
6	1	-0.1623	0.7670	0.0014	41.9592	1.5728
7	1	-0.2781	0.6424	0	48.4616	0
...
475	2	-0.2237	0.8433	0	41.2911	0
476	2	-0.2329	0.8125	0	40.7132	0
477	2	-0.1820	0.8095	0	39.8384	0
478	2	-0.2166	0.8561	0	41.8585	0
479	2	-0.2364	0.8182	0	40.2054	0
480	2	-0.1818	0.8160	0	39.6234	0
481	2	-0.2433	0.8573	0	41.6891	0
...

4.4 Hasil Klasifikasi dan Pengujian Menggunakan ELM

Proses klasifikasi menggunakan ELM terbagi menjadi dua proses, yaitu proses *training* dan proses *testing*. Pada kedua proses ini, input yang digunakan untuk pembelajaran ELM adalah hasil dari ekstraksi fitur yang berbentuk matriks berisikan parameter seperti pada Tabel 4.5, 4.6, dan 4.7. *Array* tersebut berisikan sejumlah data

yang sudah digabungkan antara data normal dan hipertensi yang dibagi menjadi dua, yaitu sebanyak 2225 untuk *training* dan 943 untuk *testing*. Proses ELM yang digunakan dalam penelitian ini adalah dengan fungsi aktivasi ‘sig’ yang digunakan untuk sinyal berbentuk fungsi sigmoid.

Selanjutnya, proses *training* diawali dengan menentukan jumlah *hidden neuron* sebagai salah satu parameter yang dibutuhkan ELM. Pada penelitian ini, *hidden neuron* yang diujikan menggunakan 6 macam *hidden neuron* dimulai dari 10, 50, 100, 500, 1000, dan 1500. Proses pembelajaran ini dilakukan berulang – ulang hingga didapatkan nilai parameter yang terbaik, yang kemudian akan dipanggil sebagai *input* pada proses *testing*. File yang disimpan ini berisikan informasi mengenai ciri *input*, *neuron output layer*, *neuron hidden layer*, *weight* (bobot), dan bias dari proses *training* pada ELM. Selanjutnya proses *testing* dilakukan dengan menggunakan data *testing* dengan menginputkan parameter yang didapat dari proses *training*. Kemudian, proses ini diuji coba pada tiap – tiap masukkan fitur, yaitu *skewness*, *peak analysis*, dan gabungan antara *skewness* dan *peak analysis*.

Hasil yang didapat pada proses ELM, ditunjukkan pada Tabel 4.8 yang menunjukkan bahwa semakin besar *hidden neuronnya*, semakin tinggi pula akurasinya. Hal ini disebabkan karena semakin tingginya *hidden neuron* akan memperbanyak opsi pilihan untuk bobot dan bias yang bagus sehingga hal ini akan mempengaruhi tingkat akurasi. Pada Tabel 4.8 dapat dilihat bahwa akurasi terbaik didapatkan pada fitur gabungan antara *skewness* dan *peak analysis* pada jumlah *hidden neuron* 1000, yaitu sebesar 89.66% dan 75.72% untuk masing – masing proses *training* dan *testing*. Hasil ini memiliki nilai akurasi yang lebih baik jika dibandingkan dengan masukkan fitur *skewness* saja ataupun *peak analysis* dimana akurasi tertinggi untuk *skewness* sendiri berada pada jumlah *hidden neuron* 50 dan 100 yang menghasilkan 60.40% akurasi *training* dan 34.99% akurasi *testing*. Sedangkan akurasi tertinggi ketika masukkan hanya fitur *peak analysis* didapatkan pada *hidden neuron* 500 dengan akurasi 81.93% untuk *training* dan 65.64% untuk *testing*. Selain itu, dapat dilihat pula semakin banyak

fitur, nilai akurasi yang dihasilkan juga semakin tinggi. Hal ini juga disebabkan karena adanya fitur yang makin banyak akan menambah opsi pilihan pada proses ELMnya.

Tabel 4.8 Hasil akurasi pada masing – masing fitur dalam berbagai macam *hidden neuron*

Jumlah <i>Hidden Neuron</i>	<i>Skewness</i>		<i>Peak Analysis</i>		<i>Skewness + Peak Analysis</i>	
	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
10	60.76%	27.57%	47.28%	81.12%	54.65%	82.18%
50	60.40%	34.99%	60.58%	48.46%	59.96%	67.02%
100	60.40%	34.99%	70.61%	53.98%	72.58%	64.26%
500	60.40%	34.89%	81.93%	65.64%	87.42%	71.69%
1000	60.40%	34.89%	83.10%	62.99%	89.66%	75.72%
1500	60.40%	34.89%	84.63%	58.85%	91.46%	64.58%

Berdasar Tabel 4.8 didapati bahwa ketika akurasi *training* tinggi, akurasi *testing* akan mengecil. Begitupula ketika akurasi *testing* tinggi, akurasi *training* akan mengecil. Hal ini disebabkan karena data yang dipilih untuk *training* bukanlah data yang paling bagus sehingga ketika melakukan *training* tidak bisa mencapai akurasi 100%. Dari hasil juga terlihat bahwa semakin besar jumlah *hidden neuron*, semakin lama juga proses yang dilakukan. Hal ini disebabkan karena *hidden neuron* yang banyak akan memakan waktu lebih besar dalam mencari opsi pilihan bobot dan bias yang bagus. Lama waktu proses dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil lama proses pada masing – masing fitur dalam berbagai macam *hidden neuron* dalam sekon

Jumlah <i>Hidden</i>	<i>Skewness</i>	<i>Peak Analysis</i>	<i>Skewness + Peak Analysis</i>
-------------------------	-----------------	----------------------	-------------------------------------

<i>Neuron</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
10	0.2031	0.1875	0.2344	0.0469	0.1250	0.0313
50	1.3281	0.0313	1.5781	0.0781	1.2031	0.1094
100	2.5625	0.1250	2.6250	0.0313	3.2969	0.0469
500	10.6563	0.1250	22.5781	0.0625	17.5625	0.1094
1000	21.2813	0.1094	24.5000	0.1406	20.6250	0.0938
1500	30.3438	0.1250	36.6094	0.1250	58.6875	0.2188

Setelah akurasi tertinggi didapatkan, perhitungan untuk sensitifitas dan spesifisitas dilakukan. Hasilnya didapatkan bahwa nilai sensitifitas sebesar 78.89% dan nilai spesifisitas sebesar 72.57% dengan masing – masing didapati bahwa data yang dinilai *true positif* sebanyak 370 dan data yang bernilai *true negatif* sebanyak 344.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diperoleh simpulan sebagai berikut:

1. Gabungan fitur *skewness* dan *peak analysis* menghasilkan nilai fitur yang paling optimal dibandingkan dengan fitur *skewness* atau fitur *peak analysis* saja. Terbukti dengan didapatkannya hasil klasifikasi tertinggi yang dapat membedakan sinyal normal dan hipertensi.
2. Tingkat akurasi sistem klasifikasi hipertensi menggunakan metode *extreme learning machine* (ELM) dalam penelitian ini sangat baik. Hasil akurasi terbaik yang didapat adalah sebesar 89.66% untuk *training* dan 75.72% untuk *testing* dengan menggunakan *hidden neuron* yang berjumlah 1000. Hasil ini lebih baik dibandingkan dengan jumlah *hidden neuron* 10, 50, 100, 500, dan 1500 yang menghasilkan akurasi *training* ataupun *testing* lebih rendah.

5.2 Saran

Dari hasil penelitian dan analisis yang telah dilakukan, saran yang dapat diberikan untuk perkembangan penelitian selanjutnya adalah:

1. Penggunaan data yang digunakan dipilah secara benar antara *training* dan *testing*, sehingga dihasilkan akurasi *training* dan *testing* yang lebih tinggi.
2. Penambahan fitur dapat dilakukan untuk mendapatkan nilai akurasi yang tinggi.
3. Dapat ditambahkan metode *cross – validation* agar data dapat diketahui nilai generalisasinya.

DAFTAR PUSTAKA

- A. Meyer-Baese and Schmid, V. J. (2014) *Feature Selection and Extraction*. doi: 10.1016/B978-0-12-409545-8.00002-9.
- Ã, G. H., Zhu, Q. and Siew, C. (2006) ‘Extreme learning machine : Theory and applications’, 70, pp. 489–501. doi: 10.1016/j.neucom.2005.12.126.
- Abdulah, E. K. (2019) ‘Analyzing Skewed Data with the Epsilon Skew Gamma Distribution’, (January 2013).
- Adam, A. *et al.* (2019) ‘A Review on Time-domain Peak Detection and Classification Algorithms for Electroencephalogram Signals’, 01(02), pp. 115–121.
- Allen, J. (2014) ‘Photoplethysmography and its application in clinical physiological measurement Photoplethysmography and its application in clinical’, (September). doi: 10.1088/0967-3334/28/3/R01.
- Artiyaningrum, B. and Azam, M. (2016) ‘FAKTOR-FAKTOR YANG BERHUBUNGAN DENGAN KEJADIAN HIPERTENSI TIDAK TERKENDALI PADA PENDERITA YANG MELAKUKAN PEMERIKSAAN RUTIN’, 1(1), pp. 12–20.
- Chowdhury, M. H. *et al.* (2020) ‘Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques’.
- Ding, S. *et al.* (2015) ‘EEG Classification’, 2015.
- Fitriyani, N. L. *et al.* (2019) ‘Development of Disease Prediction Model Based on Ensemble Learning Approach for Diabetes and Hypertension’, *IEEE Access*, 7, pp. 144777–144789. doi: 10.1109/ACCESS.2019.2945129.
- Huelsman, L. P. (1993) ‘Analog Electrical Filters’, pp. 519–530.
- K. Mutlag, W. *et al.* (2020) ‘Feature Extraction Methods : A Review Feature Extraction Methods : A Review’. doi: 10.1088/1742-6596/1591/1/012028.

- Kyriacou, P. A. and Chatterjee, S. (2022) *The origin of photoplethysmography*. First Edit, *Photoplethysmography*. First Edit. Elsevier Ltd. doi: 10.1016/b978-0-12-823374-0.00004-9.
- Li, P. and Laleg-Kirati, T. M. (2021) ‘Central Blood Pressure Estimation from Distal PPG Measurement Using Semiclassical Signal Analysis Features’, *IEEE Access*, 9, pp. 44963–44973. doi: 10.1109/ACCESS.2021.3065576.
- Lim, L., Solmi, M. and Cortese, S. (2021) ‘Neuroscience and Biobehavioral Reviews Association between anxiety and hypertension in adults : A systematic review and meta-analysis’, *Neuroscience and Biobehavioral Reviews*, 131(September), pp. 96–119. doi: 10.1016/j.neubiorev.2021.08.031.
- Lin, Z. *et al.* (2015) ‘Heart Rate Estimation Using Wrist-acquired Photoplethysmography Under Different Types of Daily Life Motion Artifact’, pp. 489–494.
- Madona, P. (2016) ‘Akuisisi Data Sinyal Photoplethysmograph (PPG) Menggunakan Photodioda’, *Jurnal Elektro dan Mesin Terapan*, 2(2), pp. 32–41. doi: 10.35143/elementer.v2i2.187.
- Mariniello, G. *et al.* (2021) ‘Automation in Construction Layout-aware Extreme Learning Machine to Detect Tendon Malfunctions in Prestressed Concrete Bridges using Stress Data’, *Automation in Construction*, 132(October), p. 103976. doi: 10.1016/j.autcon.2021.103976.
- Moyle (2003) ‘Pulse Oximetry , 2nd ed .’, p. 7279.
- Nilsson, R. (2007) *Statistical Feature Selection With Applications in Life Science*.
- Pradono, J. (2014) ‘FAKTOR-FAKTOR YANG MEMENGARUHI TERJADINYA HIPERTENSI DI DAERAH PERKOTAAN (Analisis Data Riskesdas 2007)’, *Gizi Indonesia*, 33(1), pp. 59–66. doi: 10.36457/gizindo.v33i1.83.
- Slapni Ć Ar, G., Mlakar, N. and Luštrek, M. (2019) ‘Blood pressure estimation from

- photoplethysmogram using a spectro-temporal deep neural network', *Sensors (Switzerland)*, 19(15). doi: 10.3390/s19153420.
- Tamura, T. *et al.* (2014) 'Wearable photoplethysmographic sensors—past and present', *Electronics*, 3(2), pp. 282–302. doi: 10.3390/electronics3020282.
- Tan, P. (2016) 'Applying Extreme Learning Machine to Classification of EEG BCI', pp. 228–232.
- Tjahjadi, H. and Ramli, K. (2020) 'Noninvasive blood pressure classification based on photoplethysmography using K-nearest neighbors algorithm: A feasibility study', *Information (Switzerland)*, 11(2). doi: 10.3390/info11020093.
- Tjahjadi, H., Ramli, K. and Murfi, H. (2020) 'Noninvasive Classification of Blood Pressure Based on Photoplethysmography Signals Using Bidirectional Long Short-Term Memory and Time-Frequency Analysis', *IEEE Access*, 8, pp. 20735–20748. doi: 10.1109/ACCESS.2020.2968967.
- Tony R. Kuphaldt (2007) 'Lessons In Electric Circuits, Volume II – AC'.
- Zhang, L., Zhang, D. and Tian, F. (2016) 'SVM and ELM: Who Wins? Object Recognition with Deep Convolutional Features from ImageNet', 1, pp. 249–263. doi: 10.1007/978-3-319-28397-5_20.

LAMPIRAN**Lampiran 1. Kode Program *Pre – Processing* raw data, yaitu ppgtb.mlx**

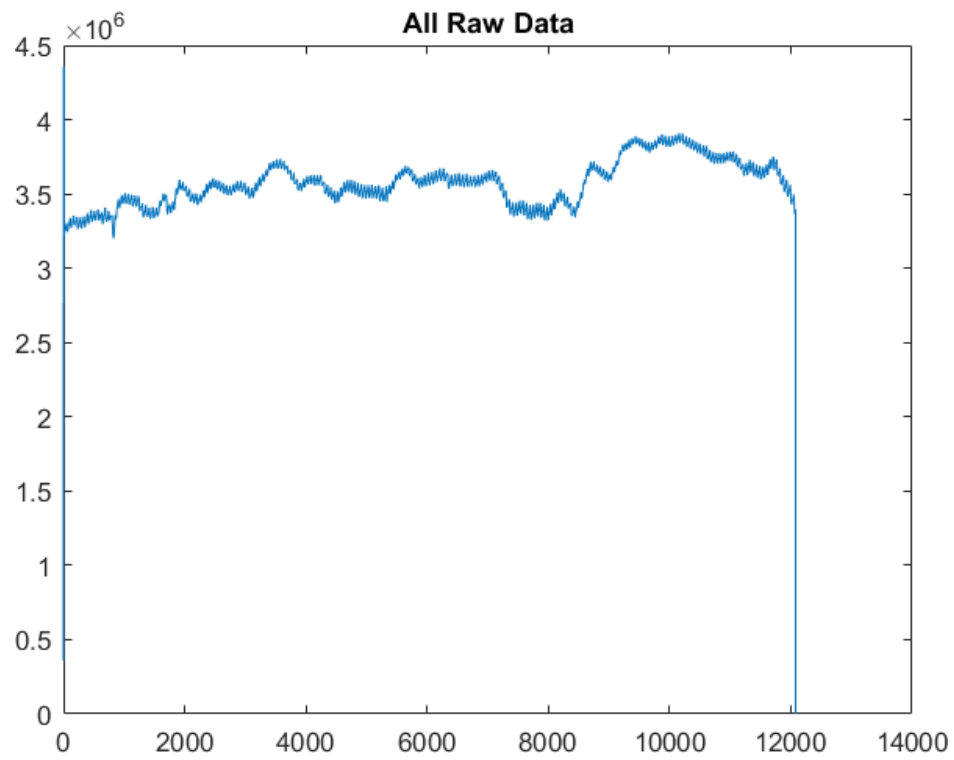
```
clc; clear all;
```

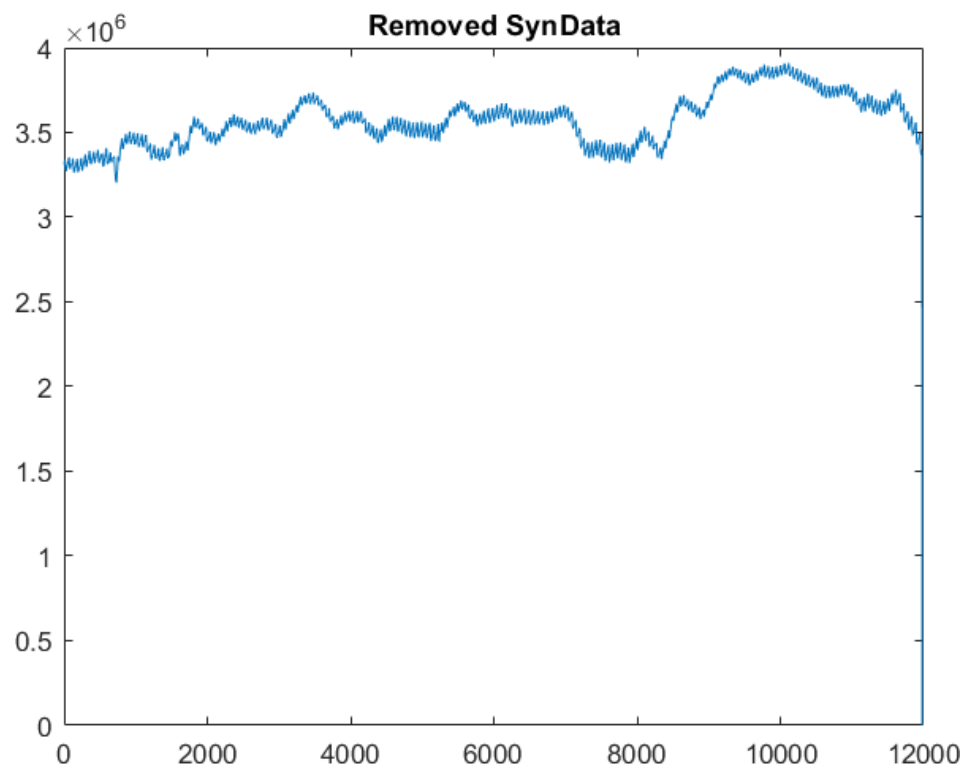
Step : 1

Load data from directory and plot

```
for data = 1
    x=load(['E:\Ilmu\Skripsi\PPG\Data File\Normal\Fix
Data\Data\NN',num2str(data),'.txt']);
    figure(1);
    plot(x);
    title('All Raw Data')
```

```
x1 = x(100:end);
figure(2);
plot(x1);
title('Removed SynData')
```

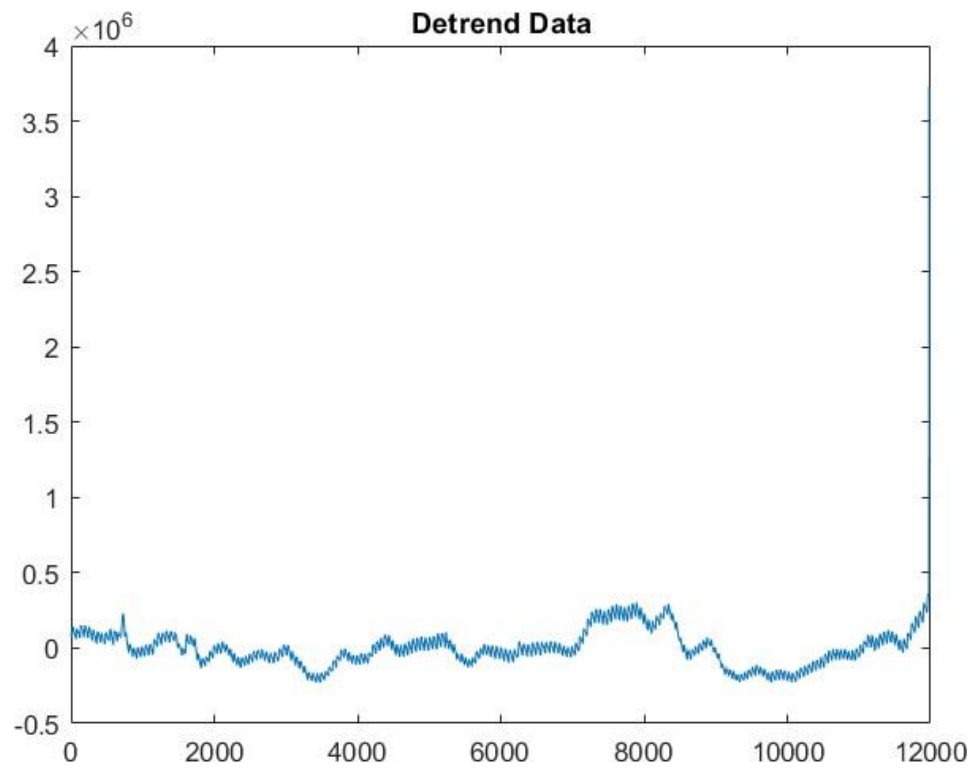





Step : 2

Remove trends from output - input data

```
x2=detrend(-x1);  
figure(3);  
plot(x2);
```

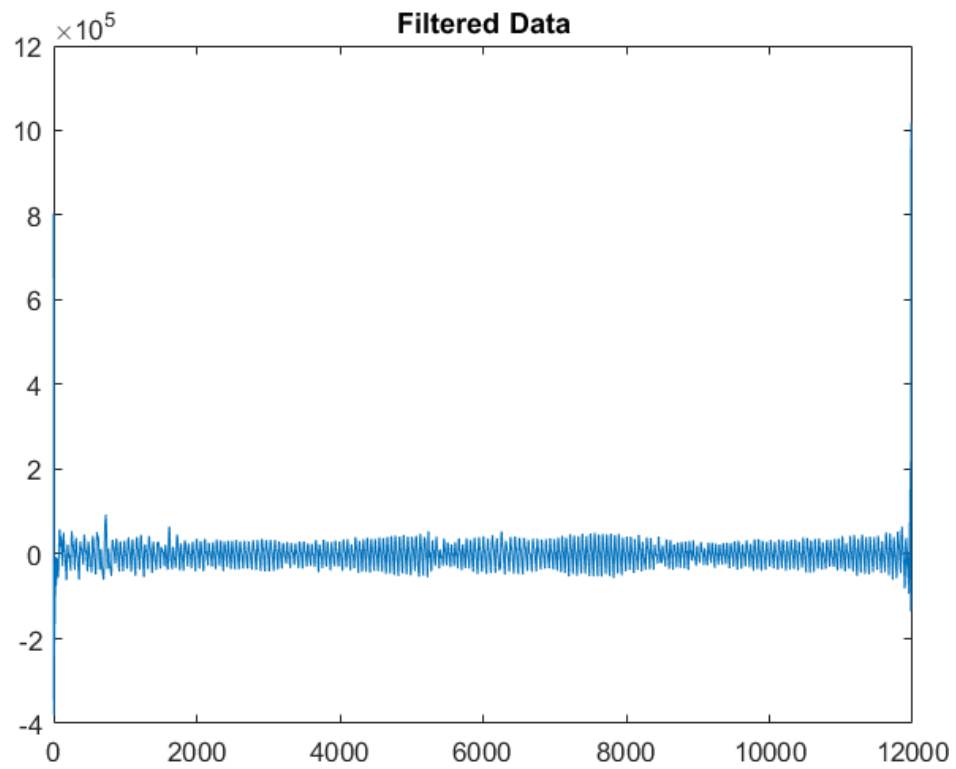


```
title('Detrend Data')
```

Step : 3

Remove noise

```
x3=BPF(x2,100,0.6,15);  
figure(4);  
plot(x3);
```

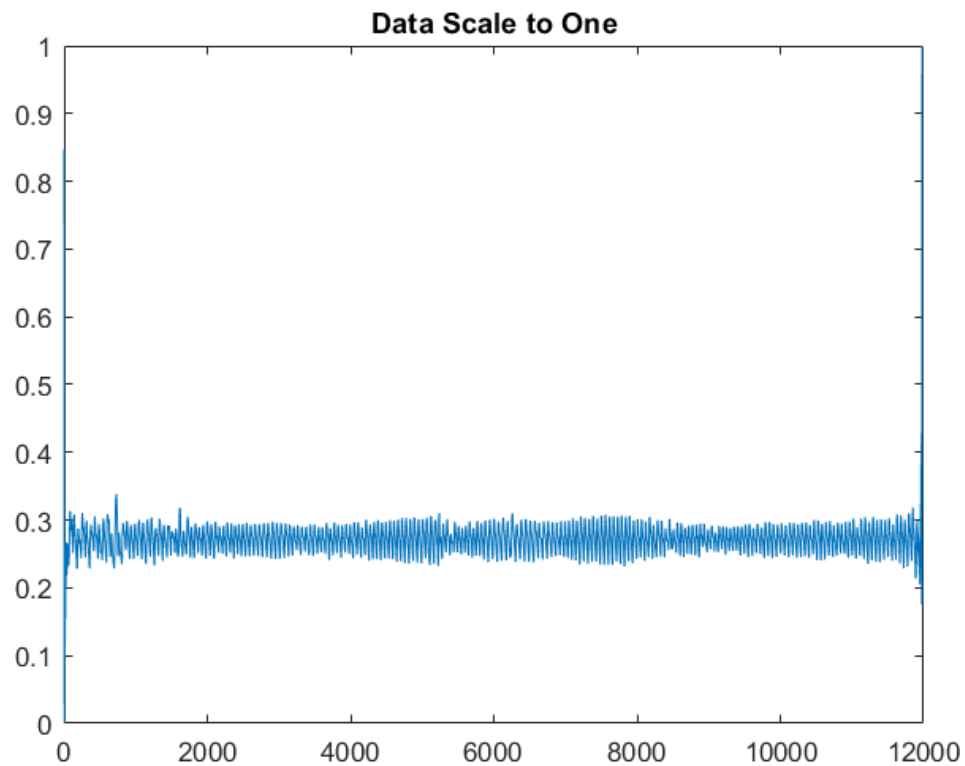


```
title('Filtered Data')
```

Step : 4

Scale both segments to have similar amplitudes close to one

```
PP_PPG1=max(x3)-min(x3);  
x4=(x3-min(x3))/PP_PPG1;  
figure(5);  
plot(x4);
```



```
title('Data Scale to One')
```

Detect the valey

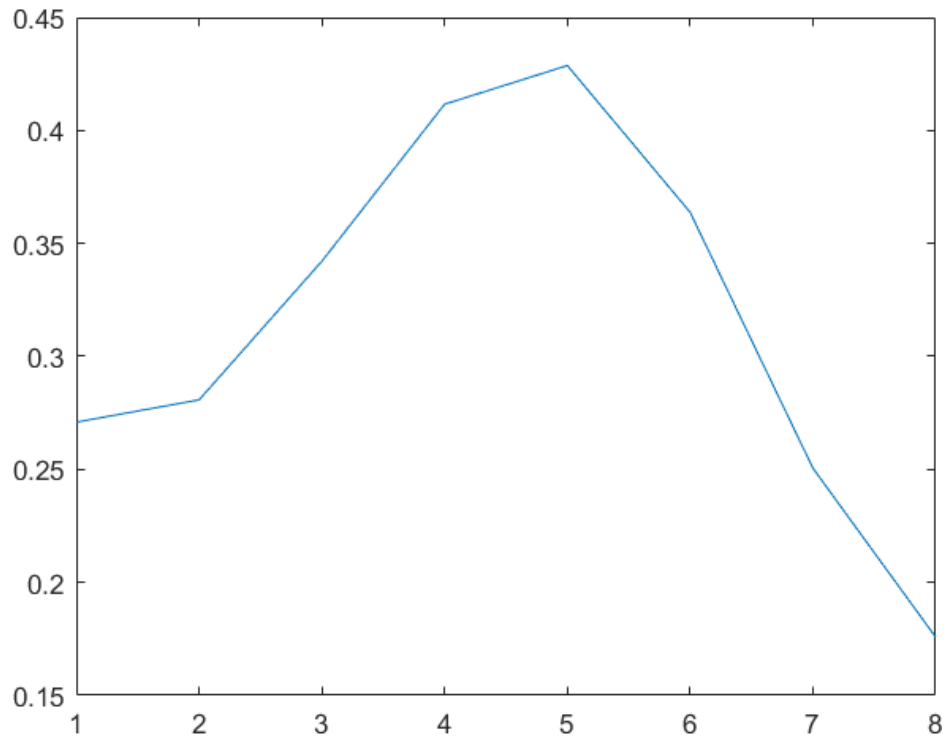
```
sig=x4;
valley_count=0;
valley={};
segment={};
for k=2:length (sig)-1 %k=1: length (sig) dari 1 tdk bs krn (k-1=0),
length(sig) tdk bs krn (k+1= tdk ada)
    if (sig(k)<sig(k-1)&& sig(k)<sig(k+1)&& sig(k)<0.42) %Tresshold
0.42
        vall=sig(k);
        valley=[valley,vall];
        k=k;
        segment=[segment,k];
        valley_count=valley_count+1;
    end
end
valley_count
```

```
valley_count = 378
```

Step 5

Single Pulse plot

```
m = {};  
p = 0;  
x5 = {};  
for k=1:(length(segment)-1)  
    x_seg=x4(cell2mat(segment(k)):cell2mat(segment(k+1)));  
    h=figure(6);  
    plot(x_seg);  
    mall=mean(x_seg);  
    p = p + mall;  
    q = p/(valley_count-1);  
    m=[m,mall];  
    fig_name = strcat('E:\Ilmu\Skripsi\PPG\Data File\Normal\New  
Data\FIG_',num2str(data),'_',num2str(k));  
    print(h,fig_name,'-dpng','-r400');  
    var_name=['E:\Ilmu\Skripsi\PPG\Data File\Normal\New  
Data\N',num2str(data),'_',num2str(k),'.mat'];  
    save(var_name,'x_seg')  
end
```



end

Lampiran 2. BPF.m

```
% Functionality: Band pass filtering via FFT
% This program filters any input sequence by a BPF
% defined by stop frequencies f1 & f2
% when sampling frequency = Fs
% Input sequence length has to be at least 2000
% samples; the filtered output is of same length
% for faster processing you can comment all
% statements between %*%
% written by Zahedi
%=====
function [so] = BPF(si,Fs,f1,f2)

Nmax=length(si);
```

```

si=si.*hamming(1,Nmax);
if mod(Nmax,2)==0
else
    Nmax=Nmax-1;
    si=si(1:Nmax);
end
t=0:Nmax-1; f=0:Fs/Nmax:Fs-Fs/Nmax;
z=fft(si);
% Define BPF
L1=ceil(f1*Nmax/Fs); L2 = ceil(f2*Nmax/Fs);
mask=[1e-10*ones(1,L1) ones(1,L2-L1) 1e-
10*ones(1,ceil(Nmax/2-L2))];
mask=[mask fliplr(mask)];
z=z.*mask';

so=real(ifft(z));

return

```

Lampiran 3. Kode Program fitur.mlx

Hipertensi (Training : 1-1110 ; Test : 1111-1579)

Skewness : 1579

Prominence : 1579

Width : 1579

Normal (Training : 1-1115 ; Test : 1116-1589)

Skewness : 1589

Prominence : 1589

Width : 1589

```

clc; clear;
for data = 6
    x = load(['E:\Ilmu\Skripsi\PPG\Data File\Normal\Rename\N',
(' ',num2str(data),' '), '.mat']);
    x = struct2cell(x);
    x = cell2mat(x);
    plot(x)
    skew = skewness(x);

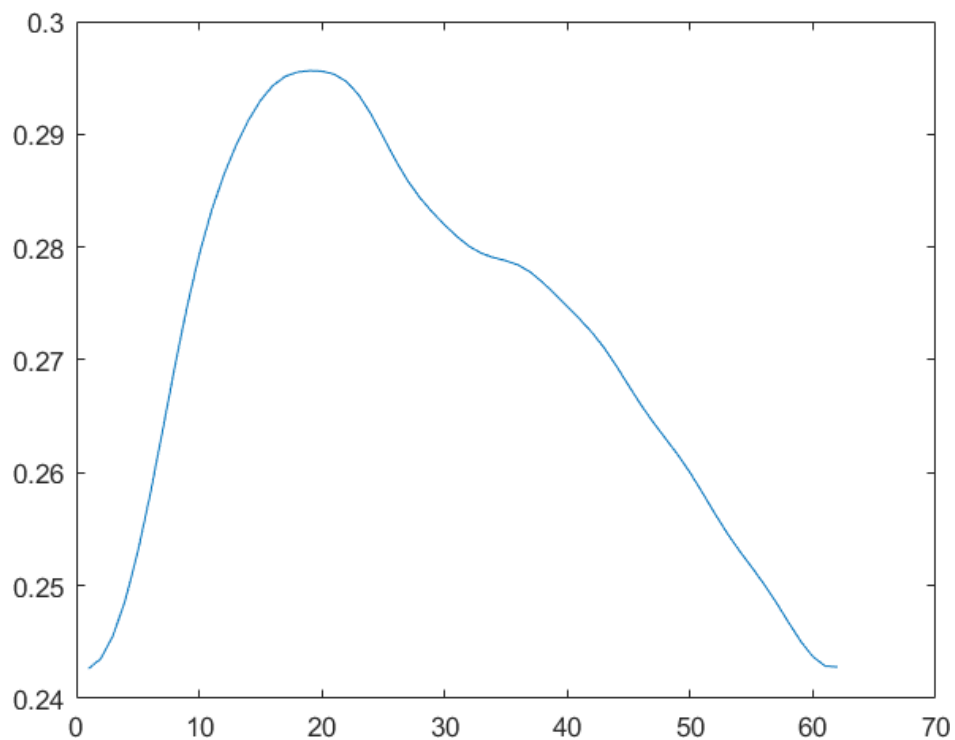
```



```

var_name=['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Skewness\S_',num2str(data)];
save(var_name,'skew')
[pks,locs,widths,proms] = findpeaks(x,"MinPeakHeight",0.2);
var_name=['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Prominence\P_',num2str(data)];
save(var_name,'proms')
var_name=['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Width\W_',num2str(data)];
save(var_name,'widths')
end

```



Lampiran 4. Kode Program Fitur_Train.mlx

Penggabungan Fitur Training

```
clc; clear;
```

TRAINING

Prominence

```

%NORMAL
PNTrain = [];
for data = 1:1115
    pn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Training\Prominence\P_',num2str(data)]);
    pn = struct2cell(pn);
    pn = cell2mat(pn);
    if length(pn) == 2
        PNTrain = [PNTrain;pn'];
    elseif length(pn) == 1
        PNTrain = [PNTrain;[pn,0]];
    else
        PNTrain = [PNTrain;[pn(2),pn(3)]];
    end
end
a1 = mean(PNTrain)
a1 = 1x2
    0.4964    0.0041
s1 = std(PNTrain)
s1 = 1x2
    0.2135    0.0251
%HIPERTENSI
PHTrain = [];
for data = 1:1110
    phpt = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Training\Prominence\P_',num2str(data)]);
    phpt = struct2cell(phpt);
    phpt = cell2mat(phpt);
    if length(phpt) == 2
        PHTrain = [PHTrain;phpt'];
    elseif length(phpt) == 1
        PHTrain = [PHTrain;[phpt,0]];
    else
        PHTrain = [PHTrain;[phpt(2),phpt(3)]];
    end
end
a2 = mean(PHTrain)
a2 = 1x2
    0.4901    0.0132
s2 = std(PHTrain)
s2 = 1x2
    0.2010    0.0848

```

Width

```

%NORMAL

```

```

WNTrain = [];
for data = 1:1115
    wn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Training\Width\W_',num2str(data)]);
    wn = struct2cell(wn);
    wn = cell2mat(wn);
    if length(wn) == 2
        WNTrain = [WNTrain;wn'];
    elseif length(wn) == 1
        WNTrain = [WNTrain;[wn,0]];
    else
        WNTrain = [WNTrain;[wn(2),wn(3)]];
    end
end
b1 = mean(WNTrain)
b1 = 1x2
    43.2547    1.8970
s3 = std(WNTrain)
s3 = 1x2
    11.5902    3.4905

%HIPERTENSI
WHTrain = [];
for data = 1:1110
    wh = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Training\Width\W_',num2str(data)]);
    wh = struct2cell(wh);
    wh = cell2mat(wh);
    if length(wh) == 2
        WHTrain = [WHTrain;wh'];
    elseif length(wh) == 1
        WHTrain = [WHTrain;[wh,0]];
    else
        WHTrain = [WHTrain;[wh(2),wh(3)]];
    end
end
b2 = mean(WHTrain)
b2 = 1x2
    43.7305    2.3913
s4 = std(WHTrain)
s4 = 1x2
    10.9811    6.8358

```

Skewness

```

%NORMAL
for data = 1:1115
    sn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Training\Skewness\S_', num2str(data)]);
    sn = struct2cell(sn);
    sn = cell2mat(sn);
    SNTrain(data,:) = [sn];
end
c1 = mean(SNTrain)
c1 = -0.1913
s5 = std(SNTrain)
s5 = 0.1640

%HIPERTENSI
for data = 1:1110
    sh = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Training\Skewness\S_', num2str(data)]);
    sh = struct2cell(sh);
    sh = cell2mat(sh);
    SHTrain(data,:) = [sh];
end
c2 = mean(SHTrain)
c2 = -0.2444
s6 = std(SHTrain)
s6 = 0.1701

```

COMBINE

```

%Skewness + Peak Analysis
x1 = [SNTrain PNTrain(:,1) PNTrain(:,2) WNTrain(:,1) WNTrain(:,2);
SHTrain PHTrain(:,1) PHTrain(:,2) WHTrain(:,1) WHTrain(:,2)];
%Skewness
x2 = [SNTrain; SHTrain];
%Peak Analysis
x3 = [PNTrain(:,1) PNTrain(:,2) WNTrain(:,1) WNTrain(:,2); PHTrain(:,1)
PHTrain(:,2) WHTrain(:,1) WHTrain(:,2)];
for i = 1:1115
    grup_a(i) = {'1'};
end
for i = 1:1110
    grup_b(i) = {'2'};
end

t1=[grup_a(1:1115) grup_b(1:1110)];

```

```

t1=str2double(t1)';
train_data=[t1 x1];
f1 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TrainSkewness+PA'];
save(f1,'train_data');

t2=[grup_a(1:1115) grup_b(1:1110)];
t2=str2double(t2)';
train_data=[t2 x2];
f2 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TrainSkewness'];
save(f2,'train_data');

t3=[grup_a(1:1115) grup_b(1:1110)];
t3=str2double(t3)';
train_data=[t3 x3];
f3 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TrainPA'];
save(f3,'train_data');

```

Lampiran 5. Kode Program Fitur_Test.mlx

Penggabungan Fitur Testing

```
clc; clear;
```

TESTING

Prominence

```

%NORMAL
PNTTest = [];
for data = 1116:1589
    pn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Test\Prominence\P_',num2str(data)]);
    pn = struct2cell(pn);
    pn = cell2mat(pn);
    if length(pn) == 2
        PNTTest = [PNTTest;pn'];
    elseif length(pn) == 1
        PNTTest = [PNTTest;[pn,0]];
    else
        PNTTest = [PNTTest;[pn(2),pn(3)]];
    end
end
a1 = mean(PNTTest)
a1 = 1x2

```

```

0.7182    0.0178

%HIPERTENSI
PHTest = [];
for data = 1111:1579
    phpt = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Prominence\P_',num2str(data)]);
    phpt = struct2cell(phpt);
    phpt = cell2mat(phpt);
    if length(phpt) == 2
        PHTest = [PHTest;phpt'];
    elseif length(phpt) == 1
        PHTest = [PHTest;[phpt,0]];
    else
        PHTest = [PHTest;[phpt(2),phpt(3)]];
    end
end
a2 = mean(PHTest)
a2 = 1x2
    0.5696    0.1958

```

```

%NORMAL
WNTTest = [];
for data = 1116:1589
    wn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Test\Width\W_',num2str(data)]);
    wn = struct2cell(wn);
    wn = cell2mat(wn);
    if length(wn) == 2
        WNTTest = [WNTTest;wn'];
    elseif length(wn) == 1
        WNTTest = [WNTTest;[wn,0]];
    else
        WNTTest = [WNTTest;[wn(2),wn(3)]];
    end
end
b1 = mean(WNTTest)
b1 = 1x2
    32.2264    3.6301

```

```

%HIPERTENSI
WHTest = [];
for data = 1111:1579

```

```

    wh = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Width\W_',num2str(data)]);
    wh = struct2cell(wh);
    wh = cell2mat(wh);
    if length(wh) == 2
        WHTest = [WHTest;wh'];
    elseif length(wh) == 1
        WHTest = [WHTest;[wh,0]];
    else
        WHTest = [WHTest;[wh(2),wh(3)]];
    end
end
b2 = mean(WHTest)
b2 = 1x2
    34.7565    12.2670

```

```

%NORMAL
i = 1;
for data = 1116:1589
    sn = load(['E:\Ilmu\Skripsi\PPG\Data
File\Normal\Test\Skewness\S_',num2str(data)]);
    sn = struct2cell(sn);
    sn = cell2mat(sn);
    SNTTest(i,:) = [sn];
    i = i+1;
end
c1 = mean(SNTTest)
c1 = -0.0176

```

```

%HIPERTENSI
j = 1;
for data = 1111:1579
    sh = load(['E:\Ilmu\Skripsi\PPG\Data
File\Hipertensi\Test\Skewness\S_',num2str(data)]);
    sh = struct2cell(sh);
    sh = cell2mat(sh);
    SHTest(j,:) = [sh];
    j = j+1;
end
c2 = mean(SHTest)
c2 = -0.1214

```

COMBINE

```

%Skewness + Peak Analysis
x1 = [SNTTest PNTTest(:,1) PNTTest(:,2) WNTTest(:,1) WNTTest(:,2); SHTest
PHTest(:,1) PHTest(:,2) WHTest(:,1) WHTest(:,2)];
%Skewness
x2 = [SNTTest; SHTest];
%Peak Analysis
x3 = [PNTTest(:,1) PNTTest(:,2) WNTTest(:,1) WNTTest(:,2); PHTest(:,1)
PHTest(:,2) WHTest(:,1) WHTest(:,2)];
for i = 1116:1589
    grup_a(i) = {'1'};
end
for i = 1111:1579
    grup_b(i) = {'2'};
end

t1=[grup_a(1116:1589) grup_b(1111:1579)];
t1=str2double(t1)';
test_data=[t1 x1];
f1 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TestSkewness+PA'];
save(f1,'test_data');

t2=[grup_a(1116:1589) grup_b(1111:1579)];
t2=str2double(t2)';
test_data=[t2 x2];
f2 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TestSkewness'];
save(f2,'test_data');

t3=[grup_a(1116:1589) grup_b(1111:1579)];
t3=str2double(t3)';
test_data=[t3 x3];
f3 = ['E:\Ilmu\Skripsi\PPG\Data File\Fitur Fix\TestPA'];
save(f3,'test_data');

```

Lampiran 6. Kode Program elm_train.m

```

function [TrainingTime, TrainingAccuracy, TestingTime]
= elm_train(TrainingData_File, Elm_Type,
NumberofHiddenNeurons, ActivationFunction)

% Usage: elm_train(TrainingData_File, Elm_Type,
NumberofHiddenNeurons, ActivationFunction)
% OR:      [TrainingTime, TrainingAccuracy] =
elm_train(TrainingData_File, Elm_Type,
NumberofHiddenNeurons, ActivationFunction)

```



```

%
% Input:
% TrainingData_File      - Filename of training data
set
% Elm_Type               - 0 for regression; 1 for
(bboth binary and multi-classes) classification
% NumberofHiddenNeurons - Number of hidden neurons
assigned to the ELM
% ActivationFunction      - Type of activation
function:
%                           'sig' for Sigmoidal
function
%                           'sin' for Sine function
%                           'hardlim' for Hardlim
function
%
% Output:
% TrainingTime           - Time (seconds) spent on
training ELM
% TrainingAccuracy       - Training accuracy:
%                           RMSE for regression or
correct classification rate for classification
%
% MULTI-CLASSE CLASSIFICATION: NUMBER OF OUTPUT
NEURONS WILL BE AUTOMATICALLY SET EQUAL TO NUMBER OF
CLASSES
% FOR EXAMPLE, if there are 7 classes in all, there
will have 7 output
% neurons; neuron 5 has the highest output means
input belongs to 5-th class
%
% Sample1 regression: [TrainingTime,
TrainingAccuracy, TestingAccuracy] =
elm_train('sinc_train', 0, 20, 'sig')
% Sample2 classification: elm_train('diabetes_train',
1, 20, 'sig')
%
% %%%      Authors:      MR QIN-YU ZHU AND DR GUANG-
BIN HUANG
% %%%      NANYANG TECHNOLOGICAL UNIVERSITY,
SINGAPORE

```

```

        %%%      EMAIL:      EGBHUANG@NTU.EDU.SG;
GBHUANG@IEEE.ORG
        %%%      WEBSITE:
http://www.ntu.edu.sg/eee/icis/cv/egbhuang.htm
        %%%      DATE:      APRIL 2004

%%%%%%%%%%%% Macro definition
REGRESSION=0;
CLASSIFIER=1;
Elm_Type=CLASSIFIER;
NumberofHiddenNeurons=10; %dimulai dari kecil
ActivationFunction='sig';
%%%%%%%%%%%% Load training dataset
load('E:\Ilmu\Skripsi\PPG\Data File\Fitur
Fix\TrainPA.mat')
T=train_data(:,1)'; %Target
P=train_data(:,2:size(train_data,2))';
clear train_data; %   Release raw training data array

NumberofTrainingData=size(P,2); %otomasi jumlah fitur
NumberofInputNeurons=size(P,1);

if Elm_Type~=REGRESSION
    %%%%%%%%%%%%% Preprocessing the data of
classification
        sorted_target=sort(T,2);
        label=zeros(1,1) %   Find and save in 'label'
class label from training and testing data sets
        label(1,1)=sorted_target(1,1);
        j=1;
        for i = 2:NumberofTrainingData
            if sorted_target(1,i) ~= label(1,j)
                j=j+1
                label(1,j) = sorted_target(1,i)
            end
        end
        number_class=j
        NumberofOutputNeurons=number_class

    %%%%%%%%%%%%% Processing the targets of training
        temp_T=zeros(NumberofOutputNeurons,
NumberofTrainingData)

```

```

for i = 1:NumberofTrainingData
    for j = 1:number_class
        if label(1,j) == T(1,i)
            break;
        end
    end
    temp_T(j,i)=1
end
T=temp_T*2-1
end %
end if of Elm_Type

%%%%%%%%%% Calculate weights & biases
train_start = cputime

%%%%%%%%%% Random generate input weights InputWeight
(w_i) and biases BiasofHiddenNeurons (b_i) of hidden
neurons
InputWeight=rand(NumberofHiddenNeurons,NumberofInputN
eurons)*2-1
BiasofHiddenNeurons=rand(NumberofHiddenNeurons,1)
tempH=InputWeight*P
clear P; %
Release input of training data
ind=ones(1,NumberofTrainingData)
BiasMatrix=BiasofHiddenNeurons(:,ind) %
Extend the bias matrix BiasofHiddenNeurons to match
the demention of H
tempH=tempH+BiasMatrix

%%%%%%%%%% Calculate hidden neuron output matrix H
switch lower(ActivationFunction)
    case {'sig','sigmoid'}
        %%%%%%%%% Sigmoid
        H = 1 ./ (1 + exp(-tempH))
    case {'sin','sine'}
        %%%%%%%%% Sine
        H = sin(tempH);
    case {'hardlim'}
        %%%%%%%%% Hard Limit
        H = hardlim(tempH);

```

```

        %%%%%%%%% More activation functions can be
added here
end
clear tempH; %
Release the temporary array for calculation of hidden
neuron output matrix H

%%%%%%%%%%%% Calculate output weights OutputWeight
(beta_i)
OutputWeight=pinv(H') * T'
end_time_train=cputime;
TrainingTime=end_time_train-train_start %
Calculate CPU time (seconds) spent for training ELM
disp(TrainingTime)

%%%%%%%%%%%% Calculate the training accuracy
Y=(H' * OutputWeight)' %
Y: the actual output of the training data
if Elm_Type == REGRESSION
    TrainingAccuracyR=sqrt(mse(T - Y))
% Calculate training accuracy (RMSE) for regression
case
    output=Y;
end
clear H;

if Elm_Type == CLASSIFIER
%%%%%%%%%%%% Calculate training & testing
classification accuracy
    MissClassificationRate_Training=0;

    for i = 1 : size(T, 2)
        [x, label_index_expected]=max(T(:,i));
        [x, label_index_actual]=max(Y(:,i));
        output(i)=label(label_index_actual);
        if label_index_actual~=label_index_expected

MissClassificationRate_Training=MissClassificationRate_Training+1;
        end
    end
end
end

```

```

    TrainingAccuracy=1-
    MissClassificationRate_Training/NumberOfTrainingData
end

if Elm_Type~=REGRESSION
    save('elm_model', 'NumberOfInputNeurons',
    'NumberOfOutputNeurons', 'InputWeight',
    'BiasofHiddenNeurons', 'OutputWeight',
    'ActivationFunction', 'label', 'Elm_Type');
else
    save('elm_model', 'InputWeight',
    'BiasofHiddenNeurons', 'OutputWeight',
    'ActivationFunction', 'Elm_Type');
end

%testing
%%%%%%%%%% Load testing dataset
load('E:\Ilmu\Skripsi\PPG\Data File\Fitur
Fix\TestPA.mat');
TV.T=test_data(:,1)';
TV.P=test_data(:,2:size(test_data,2))';
clear test_data; %
Release raw testing data array

NumberOfTestingData=size(TV.P,2);

load elm_model.mat;

if Elm_Type~=REGRESSION

    %%%%%%%%%%% Processing the targets of testing
    temp_TV_T=zeros(NumberOfOutputNeurons,
    NumberOfTestingData);
    for i = 1:NumberOfTestingData
        for j = 1:size(label,2)
            if label(1,j) == TV.T(1,i)
                break;
            end
        end
        temp_TV_T(j,i)=1;
    end
    TV.T=temp_TV_T*2-1;

```

```

end                                                    %
end if of Elm_Type

%%%%%%%%%%%% Calculate the output of testing input
test_start = cputime;
tempH_test=InputWeight*TV.P;
clear TV.P;                                     % Release input of testing
data
ind=ones(1,NumberofTestingData);
BiasMatrix=BiasofHiddenNeurons(:,ind);          %
Extend the bias matrix BiasofHiddenNeurons to match
the demention of H
tempH_test=tempH_test + BiasMatrix;
switch lower(ActivationFunction)
    case {'sig','sigmoid'}
        %%%%%%%%% Sigmoid
        H_test = 1 ./ (1 + exp(-tempH_test));
    case {'sin','sine'}
        %%%%%%%%% Sine
        H_test = sin(tempH_test);
    case {'hardlim'}
        %%%%%%%%% Hard Limit
        H_test = hardlim(tempH_test);
        %%%%%%%%% More activation functions can be
added here
end
TY=(H_test' * OutputWeight)';                    %
TY: the actual output of the testing data

test_end = cputime;
TestingTime = test_end - test_start              %Calculate
CPU time (seconds) spent by ELM predicting the whole
testing data

if Elm_Type == REGRESSION
    TestingAccuracy=sqrt(mse(TV.T - TY))          %
Calculate testing accuracy (RMSE) for regression case
    outputtest=TY;
end

if Elm_Type == CLASSIFIER

```

```

%%%%%%%%%% Calculate training & testing
classification accuracy
    MissClassificationRate_Testing=0;

    for i = 1 : size(TV.T, 2)
        [x, label_index_expected]=max(TV.T(:,i));
        [x, label_index_actual]=max(TY(:,i));
        outputtest(i)=label(label_index_actual);
        if label_index_actual~=label_index_expected

MissClassificationRate_Testing=MissClassificationRate
_Testing+1;
            end
        end
    end
    TestingAccuracy=1-
MissClassificationRate_Testing/NumberofTestingData;
end
save('acc','TrainingAccuracy','TestingAccuracy','Trai
ningTime','TestingTime')
save('elm_output','outputtest');

```

Lampiran 7. Kode Program find_thehighest_acc.m

```

clc;clear;

[TrainingTime,TrainingAccuracy,TestingTime] =
elm_train;
load acc.mat;
TrainingAccuracy;
TestingAccuracy
TestingTime

while TestingTime<0.001
    elm_train;
    load acc.mat;
end

```

Lampiran 8. Kode Program SensiSpesi.mlx

```

clc;clear

```

SPESIFISITAS

```

x = load('E:\Ilmu\Skripsi\ELM\S+PA\elm_output(1500-91,64).mat');
x = struct2cell(x);
x = cell2mat(x);

TN = 0;
FP = 0;

for i = 1:474
    y = x(1,i);
    if y == 1
        TN = TN+1;
    else
        FP = FP+1;
    end
end
TN
TN = 311
FP
FP = 163
Spesi = (TN/(TN+FP))*100
Spesi = 65.6118

```

Sensitifitas

```

TP = 0;
FN = 0;

for i = 475:943
    y = x(1,i);
    if y == 2
        TP = TP+1;
    else
        FN = FN+1;
    end
end
TP
TP = 298
FN
FN = 171
Sensi = (TP/(TP+FN))*100
Sensi = 63.5394

```