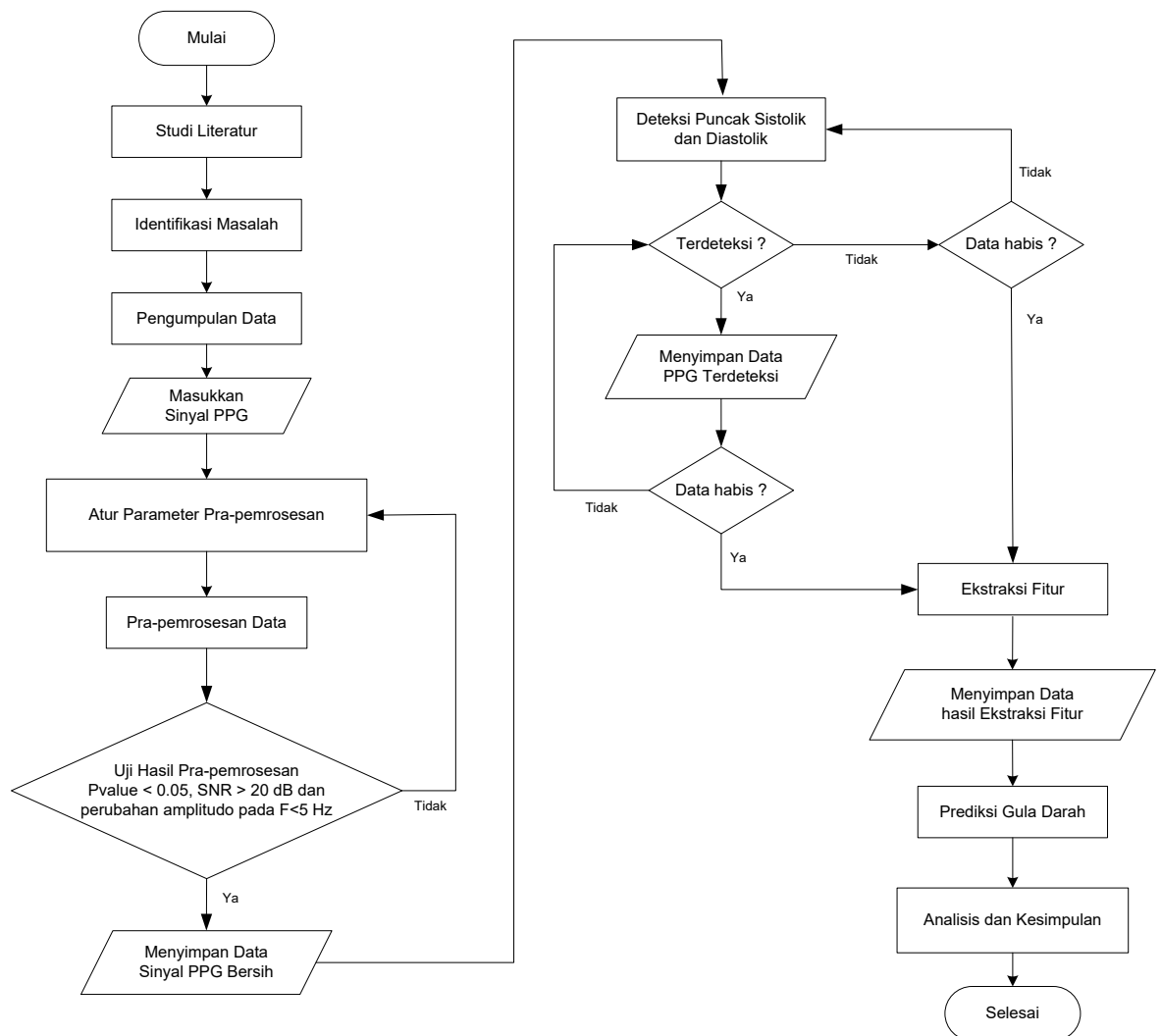


BAB III

METODE PENELITIAN

3.1 Diagram Alur Penelitian



Gambar 3. 1 Diagram Alur Penelitian

Pada Gambar 3.1 ditampilkan langkah-langkah penyusunan Tugas Akhir diantaranya sebagai berikut :

1. Mulai penelitian

Langkah pertama memulai penelitian.

2. Studi literatur

Langkah kedua, melakukan studi literatur yang bertujuan untuk merumuskan masalah dan mendapatkan teori yang mendasarinya. Teori yang digunakan dalam penelitian ini diantaranya mengenai gula darah dan metode pengukurannya, *photoplethysmography*, pra-pemrosesan sinyal PPG, analisis kualitas sinyal PPG, deteksi puncak sistolik dan diastolik, analisis *time-frequency*, fitur ekstraksi sinyal PPG, hubungan fitur ekstraksi dengan gula darah, model prediksi gula darah, dan perangkat SWADETEC. Referensi ini bersumber dari buku, jurnal nasional maupun internasional.

3. Identifikasi Masalah

Langkah ketiga, melakukan identifikasi masalah yang bersumber dari informasi terkini mengenai diabetes di Indonesia dan dari beberapa jurnal yang dibaca. Pada bagian ini tujuan yang diperoleh yaitu mengetahui masalah yang ada mengenai pengukuran gula darah konvensional yakni secara invasif.

4. Pengumpulan Data

Langkah keempat yaitu pengumpulan data yang digunakan untuk analisis. Data ini berasal dari dua sumber yang berbeda. Data pertama diperoleh dari perangkat SWADETEC yang dilengkapi dengan sensor yang menghasilkan sinyal *photoplethysmography* (PPG). Data kedua diperoleh dari *platform Mendeley Data* yang berjudul “*The dataset of photoplethysmography signals collected from a pulse sensor to measure blood glucose level*” yang mencakup sinyal PPG mentah serta parameter fisiologis terkait, seperti kadar gula darah yang diukur menggunakan glukometer, tinggi badan, berat badan, usia dan

jenis kelamin. Penggunaan kedua dataset ini menunjang penelitian untuk mengoptimalkan analisis dan memperkuat hasil yang diperoleh.

5. Masukkan Sinyal PPG

Langkah kelima, masukkan sinyal PPG dengan data fisiologis lainnya (gula darah, tinggi badan, berat badan, usia dan jenis kelamin), ke dalam sistem untuk diproses lebih lanjut.

6. Atur Parameter Pra-pemrosesan

Langkah keenam, mengatur parameter pada setiap metode pra-pemrosesan. Pada proses *smooth* diatur parameter *smooth factor* menjadi 0.25 untuk menghasilkan sinyal PPG yang halus (terbebas dari *noise*) yang dapat mengganggu analisis. Kemudian pada filter *bandpass* diatur *cut-off* frekuensi rendah sebesar 0.5 Hz dan frekuensi tinggi 5 Hz untuk mempertahankan informasi fisiologis yang terkandung dalam sinyal PPG.

7. Pra-pemrosesan Data

Langkah ketujuh, melakukan pra-pemrosesan data untuk memastikan sinyal dengan kualitas yang cukup baik yang akan digunakan dalam proses ekstraksi fitur. Pengolahan sinyal ini dilakukan menggunakan *software* MATLAB R2021a. Pada proses ini dilakukan *detrend*, *smoothing* dan *filtering* pada sinyal *Photoplethysmography* (PPG) yang sudah diatur parameternya.

8. Uji Hasil Pra-pemrosesan Data

Langkah kedelapan, setelah dilakukan pra-pemrosesan data maka selanjutnya melakukan pengujian hasil pra-pemrosesan untuk memastikan kualitas sinyal PPG. Pengujian yang dilakukan dengan memeriksa stasioneritas (Uji ADF)

divalidasi dengan parameter $p\text{-value} < 0.05$, memeriksa perubahan amplitudo pada frekuensi (Uji spektrum frekuensi) divalidasi dengan adanya perubahan pada frekuensi > 5 Hz, memeriksa kualitas sinyal PPG yang terganggu oleh *noise* (Uji SNR) divalidasi dengan parameter $\text{SNR} > 20$ dB. Jika semua parameter sudah sesuai, maka proses dilanjutkan ke tahap selanjutnya. Jika tidak, perlu adanya perbaikan dengan mengatur ulang parameter pra-pemrosesan data.

9. Menyimpan Data Sinyal PPG Bersih

Langkah kesembilan, menyimpan data sinyal PPG yang sudah dilakukan pra-pemrosesan untuk di analisis lebih lanjut.

10. Deteksi Puncak Sistolik dan Diastolik

Langkah kesepuluh, melakukan deteksi puncak sistolik dan diastolik pada sinyal PPG yang sudah bersih. Deteksi ini dilakukan menggunakan metode gradien. Proses ini bertujuan untuk memastikan bahwa sinyal relevan dan valid yang digunakan dalam analisis lebih lanjut. Sinyal PPG yang terdeteksi puncak sistolik dan puncak diastolik menandakan bahwa data yang digunakan mencerminkan siklus jantung yang sesuai dan dapat digunakan dalam ekstraksi fitur. Data sinyal PPG yang terdeteksi di simpan, sementara sinyal PPG yang tidak terdeteksi tidak disimpan atau dilanjutkan analisis.

11. Ekstraksi Fitur

Langkah kesebelas, melakukan ekstraksi fitur untuk prediksi gula darah. Dalam ekstraksi fitur ini sinyal PPG dalam domain waktu di transformasikan menjadi domain waktu-frekuensi. Transformasi sinyal ini dilakukan dengan

menggunakan *analisis time-frequency* dengan metode *Short Time Fourier Transform* (STFT) dimana, outputnya berupa spektogram. Dari spektogram tersebut di ekstraksi fitur yang berkaitan dengan kadar gula darah seperti nilai frekuensi sesaat dan nilai entropi spektral.

12. Menyimpan Data hasil Ekstraksi Fitur

Langkah kedua belas, menyimpan data fitur ekstraksi yang diperoleh dari proses sebelumnya kemudian digabungkan dengan data parameter fisiologis lainnya menjadi satu dataset. Dataset itu akan dijadikan sebagai bahan prediksi gula darah menggunakan *Toolbox Regression Learner*.

13. Prediksi Gula Darah

Langkah ketiga belas, melakukan prediksi kadar gula darah menggunakan *toolbox regression learner* pada MATLAB R2021a dengan berbagai macam metode regresi. Pada *toolbox* ini akan diperoleh model regresi yang sudah dilatih dan di evaluasi untuk prediksi kadar gula darah, sehingga diperoleh nilai kadar gula darah *non-invasive*.

14. Analisis dan Kesimpulan

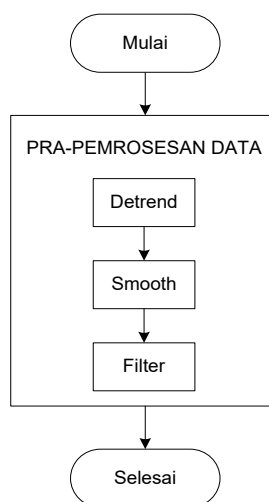
Langkah keempat belas, dilakukan evaluasi hasil model prediksi gula darah untum memahami kinerja model berdasarkan data uji. Kemudian dibuat kesimpulan hasil analisis dan memberikan rekomendasi untuk pengembangan lebih lanjut.

15. Selesai

Langkah lima belas, penelitian selesai.

3.2 Diagram Alur Pra-pemrosesan Data

Tahap pra-pemrosesan data adalah tahap awal dalam penelitian ini. Proses ini dilakukan untuk menghasilkan sinyal PPG yang bebas dari *noise* atau gangguan yang dapat mempengaruhi analisis. Pra-pemrosesan ini dilakukan dengan 3 tahapan utama yaitu *detrend* bertujuan untuk menghilangkan *baseline* maupun tren pada sinyal sehingga menghasilkan sinyal yang lebih stabil, *smooth* bertujuan untuk mengurangi *noise* pada sinyal PPG dengan metode *moving-mean* dan *filter* bertujuan untuk menghilangkan *noise* dengan frekuensi rendah maupun tinggi yang terdapat pada data sinyal PPG menggunakan filter *Bandpass*. Adapun Gambar 3.2 menggambarkan urutan dan rincian tahapan dalam perancangan proses pra-pemrosesan ini.

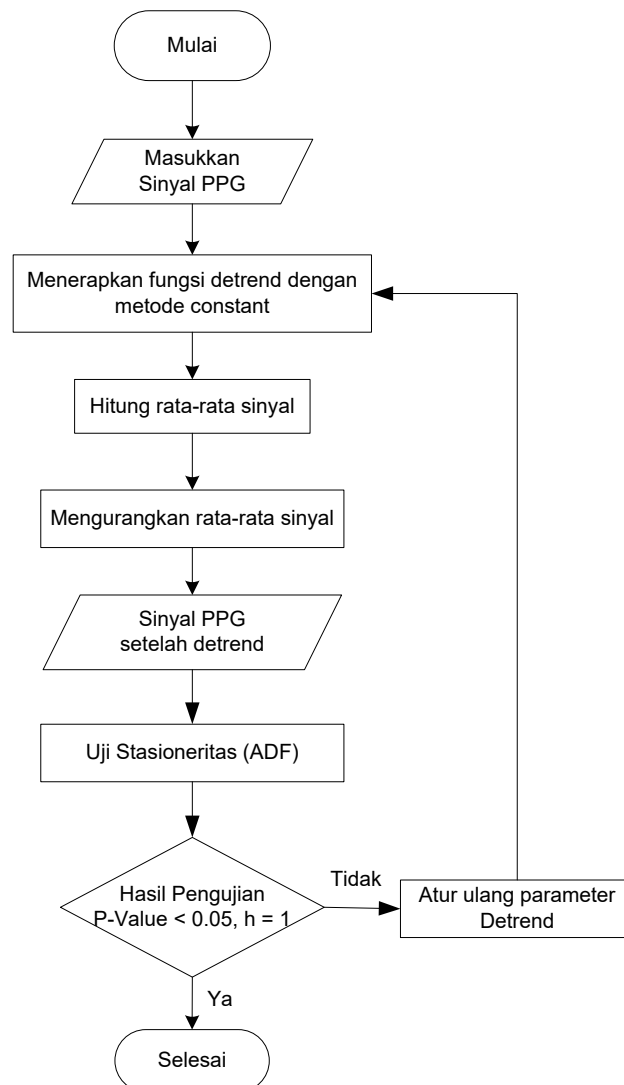


Gambar 3. 2 Diagram Alur Pra-pemrosesan Data

3.2.1 *Detrend* Sinyal PPG

Langkah pertama dalam pra-pemrosesan adalah proses *detrend* yang bertujuan untuk menghilangkan komponen DC atau *offset* dari sinyal PPG. Pada tahap ini, sinyal mentah yang diperoleh dari sensor sering kali memiliki pergeseran nilai rata-rata akibat *noise* atau artefak lingkungan. Dalam implementasinya, proses

detrend dilakukan menggunakan fungsi *detrend constant* bawaan MATLAB. Fungsi ini secara otomatis menghitung rata-rata dari sinyal, kemudian mengurangkannya dari data asli untuk mendapatkan sinyal dengan nilai tengah yang mendekati nol.



Gambar 3. 3 Diagram Alur *Detrend*

Pada Gambar 3.3 ditunjukkan langkah-langkah dalam proses *detrend* beserta penjelasan rinci setiap langkahnya sebagai berikut :

1. Mulai *Detrend* Sinyal PPG

2. Masukkan Sinyal PPG

Masukkan sinyal PPG yang akan digunakan sebagai bahan analisis data.

3. Menerapkan fungsi *detrend* dengan metode *Constant*

Detrend dengan menggunakan metode *constant* bertujuan untuk menghilangkan komponen *mean* (rata-rata) dari sinyal PPG. Dalam pemrosesan sinyal ini tren jangka panjang yang stabil akan mempengaruhi analisis sinyal dengan metode *detrend constant* dipastikan bahwa sinyal yang dianalisis memiliki nilai *mean* nol, sehingga fluktuasi sinyal di sekitar *mean* lebih jelas terlihat dan analisis menjadi lebih akurat.

4. Hitung rata-rata sinyal

Menghitung rata-rata sinyal (nilai konstan) yang merepresentasikan titik tengah dari sinyal atau *offset* sinyal.

5. Mengurangkan rata-rata sinyal

Cara kerja *detrend* dengan metode *constant* ini adalah mengurangi nilai sinyal PPG sebenarnya dengan nilai rata-rata sinyal PPG. Matlab akan secara otomatis melakukan pengurangan terhadap nilai rata-rata sinyal PPG dari setiap titik data dalam sinyal.

6. Sinyal PPG setelah *Detrend*

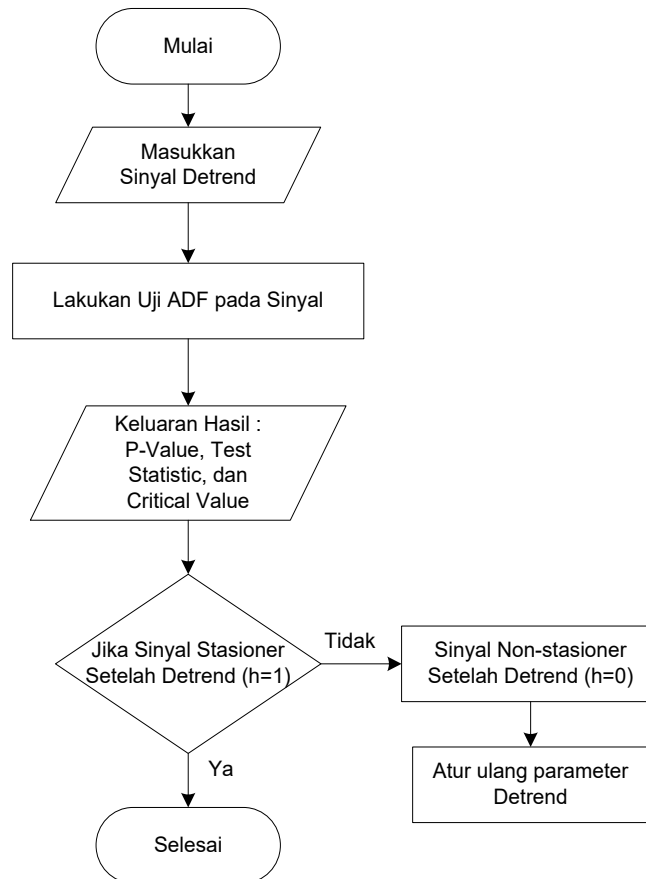
Kemudian, sinyal PPG yang sudah di *detrend* di visualisasikan dalam bentuk plot MATLAB.

7. Uji Stasioneritas ADF (*Augmented Dickey-Fuller*)

Sinyal PPG setelah *detrend* kemudian diuji stasioneritas-nya dengan menggunakan metode ADF (*Augmented Dickey-Fuller*) yang bertujuan

melihat apakah sinyal PPG stasioner atau non-stasioner. Diagram alur pengujian stasioneritas dengan metode ADF ditunjukkan pada Gambar 3.4.

8. Selesai *Detrend* Sinyal PPG



Gambar 3. 4 Diagram Alur Uji ADF (*Augmented Dickey-Fuller*)

Proses uji ADF dalam pengujian hasil *detrend* melibatkan beberapa langkah sebagai berikut :

1) Mulai uji ADF

Memulai Uji ADF (*Augmented Dickey-Fuller*).

2) Masukkan Sinyal *Detrend*

Masukkan sinyal yang telah melalui proses *detrend* untuk diuji stasioneritasnya.

3) Lakukan Uji ADF pada Sinyal *Detrend*

Menerapkan uji ADF untuk menentukan apakah sinyal bersifat stasioner atau non-stasioner.

4) Keluaran Hasil

Menyimpan atau menampilkan nilai *P-Value*, *Test Statistic* dan *Critical Value* dari hasil uji ADF yang merupakan hasil pengujian.

5) Cek Stasioneritas Sinyal

Hasil pengujian ADF tergantung nilai hipotesis yang ditampilkan. Apabila:

3) $h = 1$ maka sinyal bersifat stasioner setelah *detrend*, sehingga dapat dilanjutkan ke proses selanjutnya.

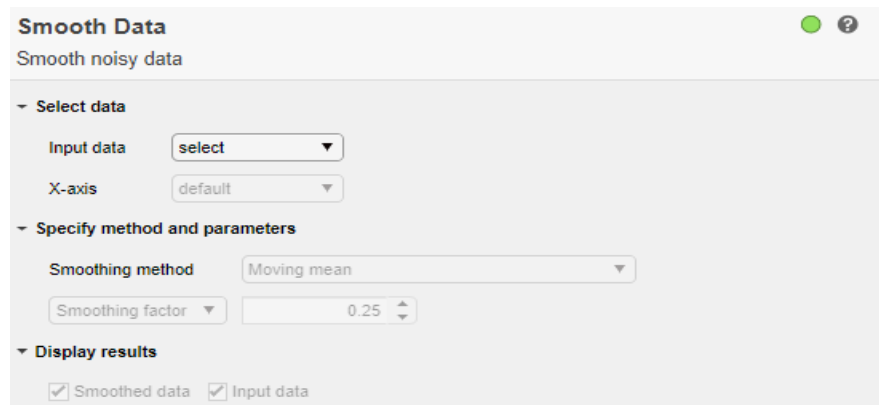
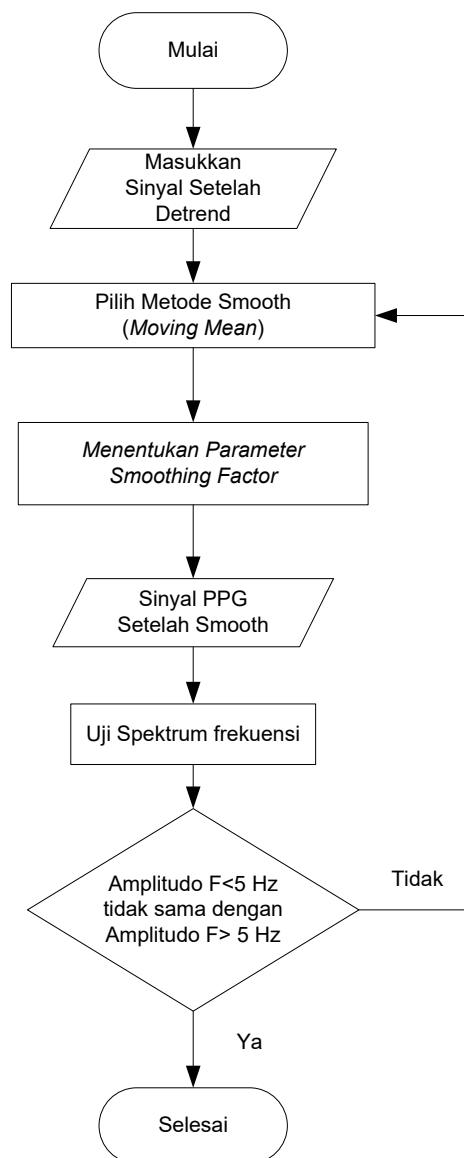
4) $h = 0$ maka sinyal bersifat non-stasioner setelah *detrend*, sehingga pengaturan ulang parameter *detrend* harus dilakukan.

6) Selesai Uji ADF

Selesai melakukan uji ADF (*Augmented Dickey-Fuller*).

3.2.2 Smooth Sinyal PPG

Proses *smooth* dilakukan menggunakan *toolbox smooth data* yang tersedia pada MATLAB. *Toolbox* ini berisikan beberapa opsi metode penghalusan data dan parameter yang dapat diatur sesuai kebutuhan analisis dapat ditunjukkan pada Gambar 3.5. Proses *smooth* bertujuan untuk mengurangi *noise* pada sinyal PPG dengan cara menghaluskan sinyal dengan berbagai metode *smooth*. Metode *smooth* yang digunakan pada penelitian ini adalah *moving mean*. Metode ini bekerja dengan menghitung rata-rata dari beberapa titik data di sekitar setiap titik sinyal, sehingga menghasilkan sinyal yang lebih halus (terbebas dari *noise*).

Gambar 3. 5 Tampilan *Toolbox Smooth Data*Gambar 3. 6 Diagram Alur *Smooth* Sinyal PPG

Pada Gambar 3.6 ditunjukkan langkah-langkah dalam proses *smooth*. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai *Smooth* Sinyal PPG

2. Masukkan Sinyal Setelah *Detrend*

Masukkan sinyal PPG setelah *detrend* yang akan digunakan dalam analisis.

3. Pilih Metode *Smooth*

Pilih *Moving-Mean* sebagai metode *smooth* yang akan diterapkan.

4. Menentukan Parameter

Tentukan nilai parameter *smoothing factor* sebesar 0.25. Nilai ini dapat disesuaikan apabila hasil *smooth* masih mengandung *noise* yang akan berpengaruh pada kualitas sinyal.

5. Sinyal PPG Setelah *Smooth*

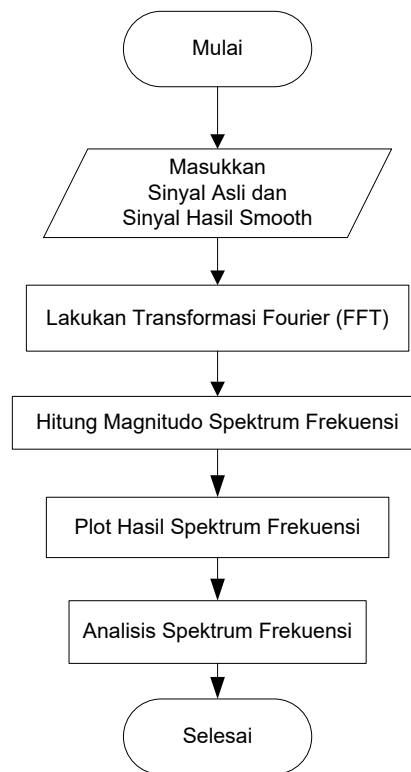
Sinyal PPG yang sudah di *smooth* di visualisasikan dalam bentuk plot MATLAB.

6. Uji Spektrum Frekuensi Sinyal PPG

Sinyal yang sudah di *smooth* kemudian di uji dengan melihat bagaimana distribusi frekuensi sinyal berubah setelah diterapkan proses *smooth*. Tujuan dari pengujian ini adalah untuk melihat apakah proses *smooth* berhasil mengurangi *noise* tanpa merusak komponen frekuensi penting dalam sinyal.

Diagram alur pengujian spektrum frekuensi sinyal PPG ditunjukkan pada Gambar 3.7.

7. Selesai *Smooth* Sinyal PPG



Gambar 3. 7 Diagram Alur Uji Spektrum Frekuensi Pada Sinyal

Gambar 3.7 menunjukkan proses uji spektrum frekuensi dalam pengujian hasil *smooth* melibatkan beberapa langkah sebagai berikut :

1) Mulai uji Spektrum Frekuensi

Memulai uji spektrum frekuensi pada sinyal asli dan sinyal hasil *smooth*.

2) Masukkan Sinyal

Masukkan sinyal asli dan sinyal hasil *smooth* untuk di proses lebih lanjut.

3) Lakukan *Transformasi Fourier* (FFT)

Lakukan *transformasi fourier* (FFT) pada kedua sinyal untuk mengubah data dari domain waktu ke domain frekuensi.

4) Hitung *Magnitude* Spektrum Frekuensi

Menghitung *magnitude* spektrum frekuensi dari hasil transformasi sinyal tersebut yang menunjukkan kekuatan sinyal pada pada setiap frekuensi.

5) Plot Hasil Spektrum Frekuensi

Plot spektrum frekuensi pada sinyal asli dan sinyal hasil *smooth*. Plot ini menunjukkan distribusi energi sinyal dalam domain frekuensi.

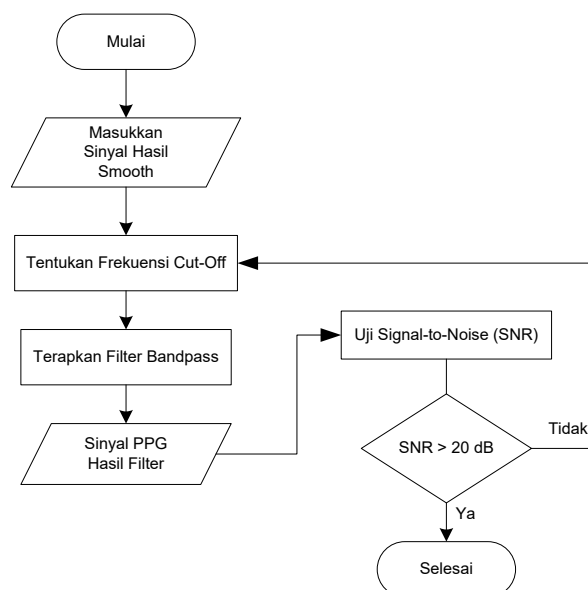
6) Analisis Spektrum Frekuensi

Analisis hasil spektrum frekuensi dengan membandingkan spektrum frekuensi dari kedua sinyal tersebut dan perhatikan apakah komponen frekuensi tinggi yang menunjukkan *noise* berkurang atau tidak serta komponen penting tetap terlihat jelas atau tidak. Jika ada pengurangan berarti hasil *smooth* berhasil dilakukan.

7) Selesai Uji Spektrum Frekuensi

3.2.3 Filter Sinyal PPG

Proses *filter* merupakan tahap pra-pemrosesan yang bertujuan untuk menghilangkan *noise* dengan frekuensi rendah maupun tinggi yang terdapat pada sinyal PPG.



Gambar 3. 8 Diagram Alur *Filter* Sinyal PPG

Pada gambar 3.8 ditunjukkan langkah-langkah dalam proses *filter* pada sinyal *Photoplethysmography* (PPG) sebagai berikut :

1. Mulai *Filter* Sinyal PPG

2. Masukkan Sinyal Hasil *Smooth*

Masukkan sinyal hasil *smooth* yang akan digunakan dalam proses *filter*.

3. Tentukan Frekuensi *Cut-Off*

Penelitian ini menguji beberapa kombinasi frekuensi *cut-off filter bandpass*, yaitu 0,3–4 Hz, 0,5–5 Hz, 0,7–8 Hz, dan 1–4 Hz untuk mengevaluasi pengaruhnya terhadap kualitas sinyal PPG. Rentang 0,5-5 Hz dipilih sebagai acuan utama berdasarkan literatur yang menyatakan bahwa informasi fisiologis utama PPG berada dalam rentang tersebut.

4. Terapkan *Filter Bandpass*

Filter ini bekerja dengan membatasi sinyal pada rentang frekuensi tertentu sesuai dengan *cut-off* bawah atau *cut-off* atas yang telah ditentukan.

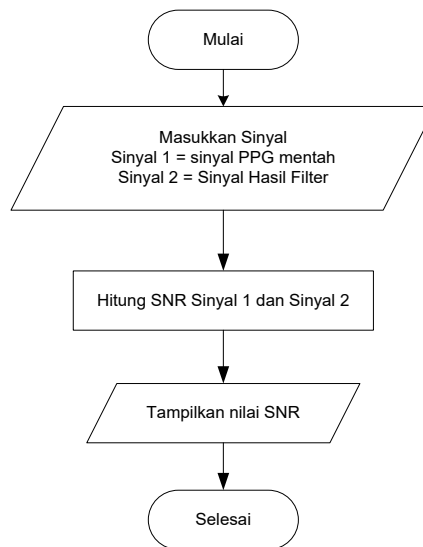
5. Sinyal PPG Hasil *Filter*

Plot sinyal PPG untuk melihat apakah frekuensi yang dipertahankan sesuai.

6. Uji *Signal to-Noise Ratio* (SNR)

Selanjutnya pengujian *Signal to-Noise* (SNR) yaitu uji perbandingan antara sinyal informasi (PPG yang dibutuhkan) dengan *noise* yang terdapat pada sebuah sinyal analisis. Semakin tinggi nilai SNR semakin baik kualitas sinyal informasi yang terkandung dalam sinyal tersebut. Tujuan pengujian ini adalah menentukan kualitas dari sebuah sinyal yang terganggu oleh *noise*. Diagram alur pengujian *Signal to-Noise* (SNR) sinyal PPG ditunjukkan pada Gambar 3.9.

7. Selesai *Filter* Sinyal PPG



Gambar 3. 9 Diagram Alur Uji *Signal to-Noise* (SNR)

Proses uji *Signal to-Noise* (SNR) dalam pengujian hasil *filter* melibatkan beberapa langkah berikut :

- 1) Mulai Uji SNR
- 2) Masukkan Sinyal

Masukkan sinyal PPG untuk dihitung nilai *Signal to-Noise* (SNR)-nya :

- Sinyal 1 = sinyal PPG mentah (sebelum pra-pemrosesan).
- Sinyal 2 = sinyal hasil *filter* (sesudah pra-pemrosesan).

- 3) Hitung nilai SNR

Hitung nilai SNR dari kedua sinyal tersebut menggunakan persamaan 2.6 yang tertera pada Bab 2 untuk melihat perbandingan sinyal 1 dan sinyal 2.

- 4) Tampilkan nilai SNR

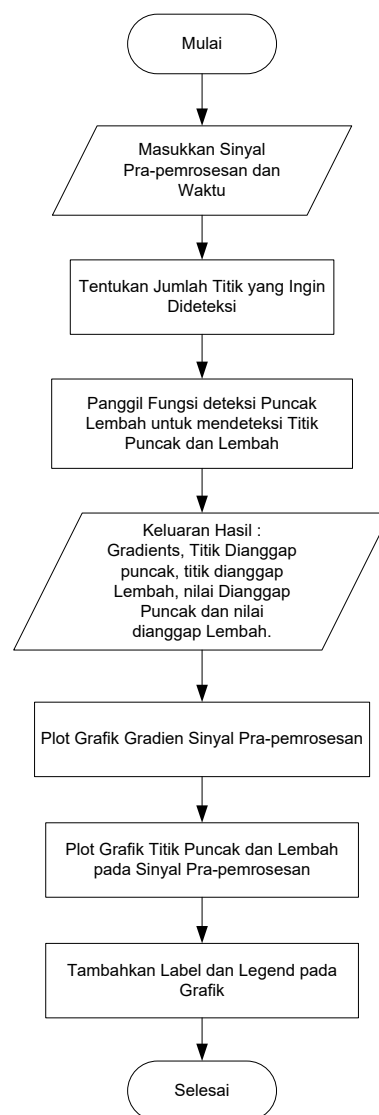
Tampilkan dan bandingkan nilai SNR dari sinyal 1 dan sinyal 2.

- 5) Selesai Uji SNR

3.3 Diagram Alur Deteksi Puncak Sistolik dan Diastolik

Validasi sinyal pra-pemrosesan dilakukan untuk memastikan bahwa sinyal yang digunakan untuk ekstraksi fitur memiliki pola fisiologis yang sesuai dengan siklus kardiovaskular. Untuk itu, puncak sistolik dan diastolik pada sinyal pra-pemrosesan dideteksi menggunakan metode gradien.

3.3.1 Deteksi Puncak Lembah



Gambar 3. 10 Diagram Alur Deteksi Puncak Lembah

Gambar 3.10 menunjukkan diagram alur proses deteksi puncak lembah.

Proses ini terdiri dari beberapa langkah diantaranya sebagai berikut :

1. Mulai Deteksi Puncak Dan Lembah

2. Masukkan Sinyal Pra-pemrosesan dan Waktu

Masukkan sinyal setelah pra-pemrosesan dan waktu untuk proses deteksi puncak dan lembah.

3. Menentukan Jumlah Titik yang Ingin Dideteksi

Menentukan jumlah titik yang akan dianalisis pada sinyal pra-pemrosesan ini penting untuk membatasi titik yang dianalisis secara optimal. Jumlah titik harus sesuai dengan data yang dimiliki, yakni tidak sedikit dan tidak terlalu banyak tujuannya agar semua titik terdeteksi dan proses deteksi tidak memakan waktu yang lama.

4. Panggil Fungsi deteksi Puncak Lembah

Fungsi deteksi Puncak Lembah yang sudah tersedia (ditunjukkan pada Gambar 3.10) digunakan untuk mendeteksi puncak dan lembah pada sinyal pra-pemrosesan. Hasil yang diperoleh berupa variabel `lembahPPG`, `lembahPPGIndex`, `puncakPPG` dan `puncakPPGIndex`.

5. Keluaran Hasil

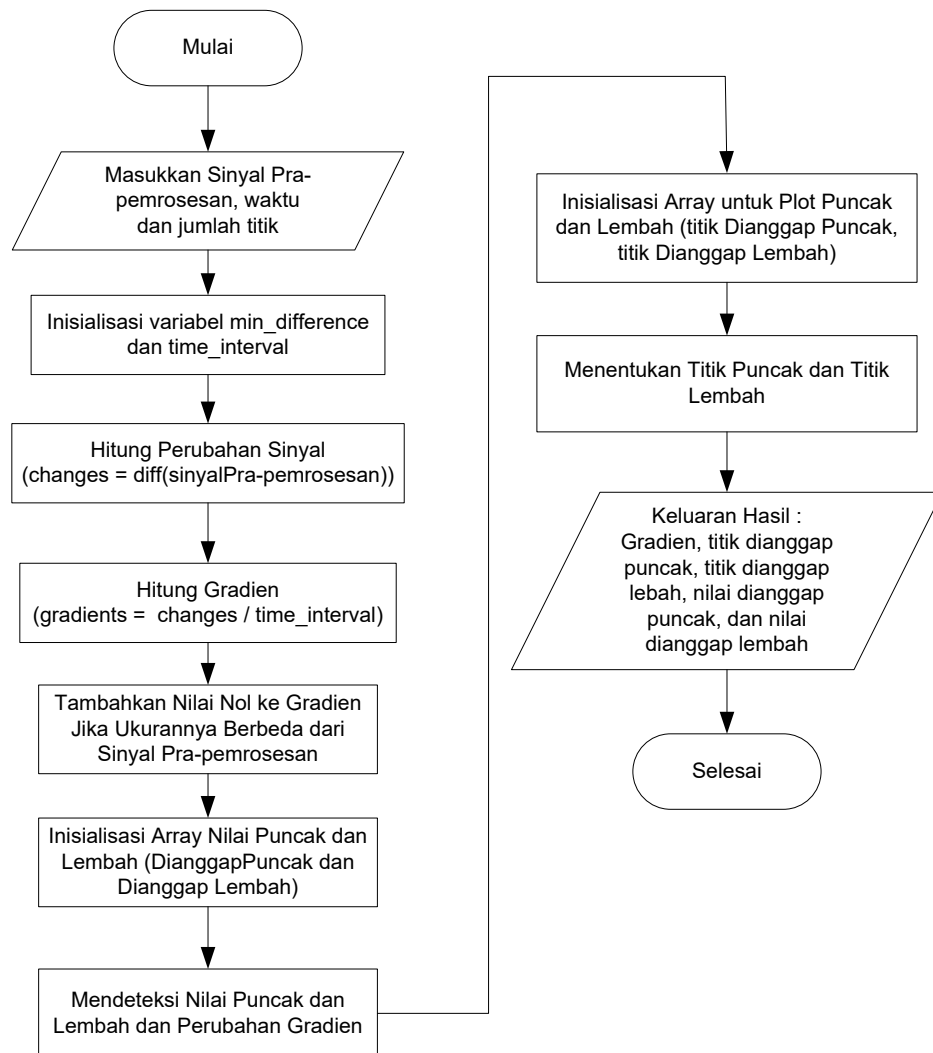
Hasil yang diperoleh berupa variabel seperti `gradients`, `titikDianggapPuncak`, `titikDianggapLembah`, `nilaiDianggapPuncak`, `nilaiDianggapLembah`.

6. Plot Hasil Deteksi

Hasil deteksi puncak dan lembah divisualisasikan dengan menampilkan grafik gradien sinyal pra-pemrosesan yang menunjukkan gradien terhadap

waktu. Selain itu, visualisaikan grafik sinyal pra-pemrosesan yang telah terdeteksi titik puncak dan lembah. Kemudian, tambahkan judul, label sumbu waktu dan amplitudo dan *legend* untuk membedakan puncak dan lembah sehingga hasil dapat mudah dipahami.

7. Selesai Deteksi Puncak dan Lembah



Gambar 3. 11 Diagram Alur Fungsi Deteksi Puncak Lembah

Gambar 11 menunjukkan fungsi deteksi Puncak Lembah yang digunakan dalam proses deteksi puncak dan lembah melalui beberapa tahap diantaranya sebagai berikut :

1. Mulai fungsi deteksi puncak dan lembah

2. Masukkan sinyal Pra-pemrosesan, Waktu dan Jumlah Titik

Masukkan data sinyal pra-pemrosesan, waktu dan jumlah titik yang akan digunakan dalam proses deteksi puncak dan lembah.

3. Inisialisasi Variabel

variabel *min_difference* diinisialisasi dengan nilai 0.005 sebagai ambang batas minimum untuk membedakan antara puncak dan lembah, sedangkan *time_interval* di atur dengan nilai 1 sebagai interval waktu. Nilai *min_difference* dapat diatur agar tetap lebih kecil dari perbedaan antara puncak sistolik dan diastolik berdasarkan karakteristik sinyal tujuannya mencegah deteksi puncak kecil yang bukan merupakan puncak sistolik dan diastolik, yang tidak terfilter pada tahap pra-pemrosesan.

4. Hitung Perubahan Sinyal (*Changes*)

Gunakan fungsi *diff* untuk menghitung perubahan antara setiap pasangan titik bertetangga dalam sinyal pra-pemrosesan. Fungsi ini menghasilkan informasi tentang perbedaan nilai sinyal pada setiap waktu.

5. Hitung Gradien

Gradien dihitung dengan membagi perubahan sinyal (*changes*) dengan interval waktu (*time_interval*). Gradien ini kemudian digunakan untuk mengidentifikasi titik puncak dan lembah dalam sinyal.

6. Tambahkan Nilai Nol ke Gradien

Apabila ukuran gradien lebih kecil dari sinyal pra-pemrosesan, maka tambahkan nilai nol untuk menyesuaikan ukuran keduanya.

7. Inisialisasi *Array* Nilai Puncak dan Lembah

Buat *array* `nilaiDianggapPuncak` dan `nilaiDianggapLembah` untuk menyimpan nilai-nilai puncak dan lembah yang terdeteksi selama proses identifikasi.

8. Mendeteksi nilai Puncak dan Lembah dari Perubahan Gradien

Titik puncak terdeteksi ketika gradien berbalik dari positif ke negatif, sementara titik lembah terdeteksi ketika gradien berbalik dari negatif ke positif.

9. Inisialisasi *Array* untuk Plot Puncak dan Lembah

Buat *array* untuk menyimpan posisi titik-titik yang dianggap sebagai puncak dan lembah, yang dapat digunakan dalam visualiasi hasil deteksi.

10. Menentukan Titik Puncak dan Lembah pada Sinyal

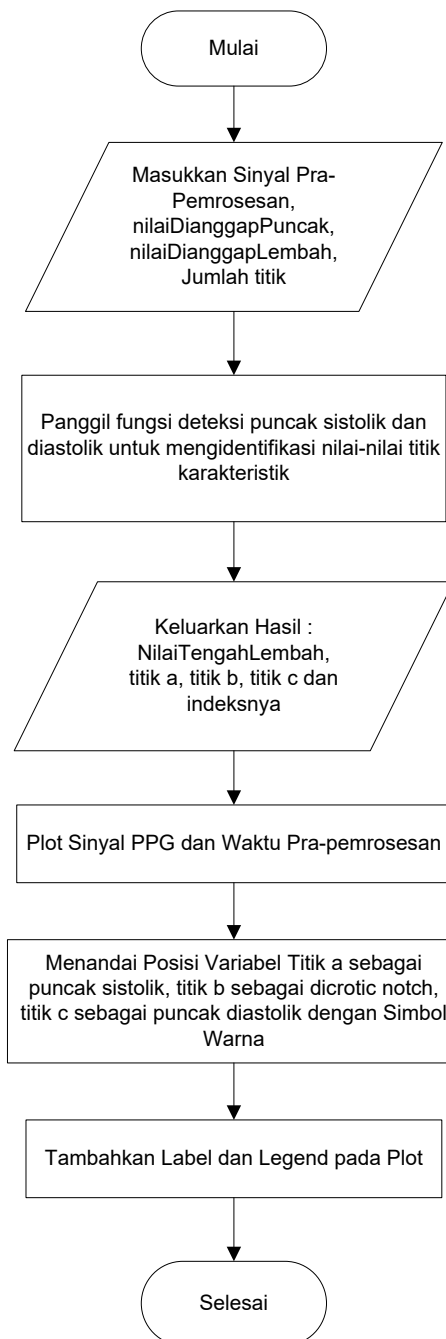
Menentukan nilai puncak dan lembah pada sinyal PPG yang telah melalui tahap pra-pemrosesan untuk menentukan titik puncak dan lembah pada sinyal.

11. Keluaran Hasil

Hasil yang diperoleh berupa variabel-variabel seperti gradien, `TitikDianggapPuncak`, `TitikDianggapLembah`, `NilaiDianggapPuncak`, `NilaiDianggapLembah`.

12. Selesai fungsi deteksi puncak lembah

3.3.2 Deteksi Puncak Sistolik dan Diastolik



Gambar 3. 12 Diagram Alur Deteksi Puncak Sistolik dan Puncak Diastolik

Pada Gambar 3.12 menunjukkan proses deteksi puncak sistolik dan diastolik melalui beberapa tahap diantaranya sebagai berikut :

1. Mulai Deteksi Puncak Sistolik dan Puncak Diastolik

2. Masukkan Sinyal Pra-Pemrosesan dan Variabel lainnya

Masukkan sinyal pra-pemrosesan, nilai dianggap puncak, nilai dianggap lembah dan jumlah titik.

3. Panggil Fungsi Deteksi Puncak Sistolik dan Puncak Diastolik

Fungsi deteksi puncak sistolik dan puncak diastolik yang sudah tersedia (ditunjukkan pada Gambar 3.12) digunakan untuk mendeteksi puncak sistolik dan puncak diastolik pada sinyal pra-pemrosesan.

4. Keluarkan Hasil :

Hasil yang diperoleh berupa nilai tengah lembah, titik a, titik b, titik c dan indeksinya.

5. Plot Sinyal dan Waktu PPG Pra-pemrosesan

Hasil deteksi puncak sistolik dan diastolik divisualisasikan dengan menampilkan grafik sinyal pra-pemrosesan yang telah terdeteksi titik a, titik b, dan titik c.

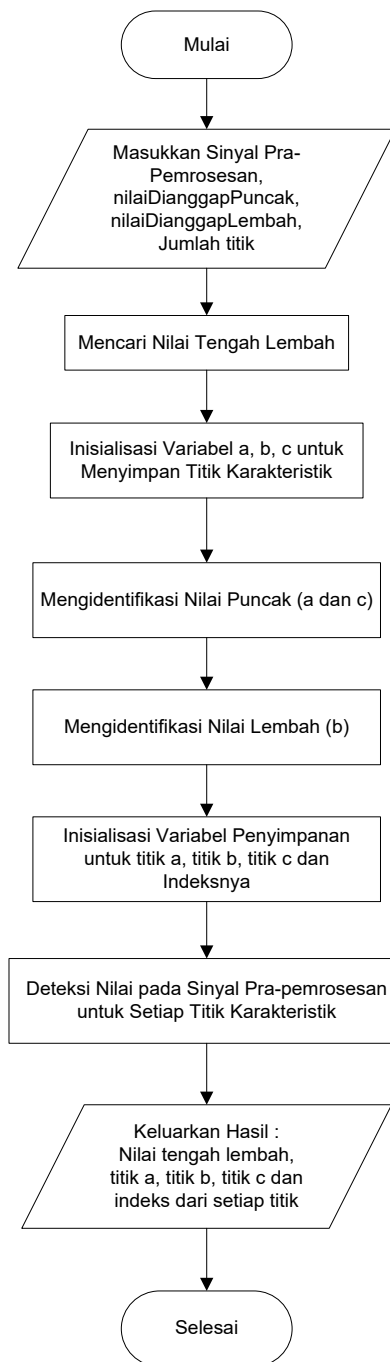
6. Menandai Posisi Variabel Titik a, Titik b, dan Titik c

Menandai posisi titik a sebagai puncak sistolik, titik b sebagai titik dicrotic notch, titik c sebagai titik puncak diastolik dengan berbagai simbol agar memudahkan dalam membedakan titik yang terdeteksi.

7. Tambahkan Label dan Legend pada Plot

Tambahkan judul, label sumbu waktu dan amplitudo dan *legend* untuk membedakan titik yang terdeteksi sehingga hasil dapat mudah dipahami.

8. Selesai Deteksi Puncak Sistolik dan Puncak Diastolik



Gambar 3. 13 Diagram Alur Fungsi Deteksi Puncak Sistolik dan Diastolik

Pada Gambar 3.13 menunjukkan fungsi deteksi puncak sistolik dan diastolik melalui beberapa tahap diantaranya sebagai berikut :

1. Mulai Fungsi Deteksi Puncak Sistolik Dan Diastolik
2. Masukkan Variabel untuk Analisis

Masukkan variabel *input* yaitu sinyal pra-pemrosesan, nilai dianggap puncak, nilai dianggap lembah dan jumlah titik.

3. Mencari Nilai Tengah Lembah

Menghitung nilai rata-rata diantara titik-titik lembah untuk referensi.

4. Inisialisasi Variabel a, b, c untuk Menyimpan Titik Karakteristik

Variabel a, b dan c diinisialisasi untuk menyimpan nilai dan indeks titik karakteristik yaitu puncak (a dan c) dan lembah (b).

5. Mengidentifikasi Nilai Puncak (a dan c)

Algoritma yang telah dibuat mendeteksi titik-titik puncak pada sinyal berdasarkan parameter nilaiDianggapPuncak.

6. Mengidentifikasi Nilai Lembah (b)

Algoritma yang telah dibuat mendeteksi titik-titik lembah pada sinyal berdasarkan parameter nilaiDianggapLembah.

7. Inisialisasi Variabel Penyimpanan pada Setiap Titik

Inisialisasi variabel penyimpanan untuk titik a, titik b, titik c dan indeksnya.

8. Deteksi Nilai pada Sinyal Pra-pemrosesan

Algoritma memindai sinyal untuk memastikan bahwa nilai-nilai puncak dan lembah serta indeksnya benar terdeteksi.

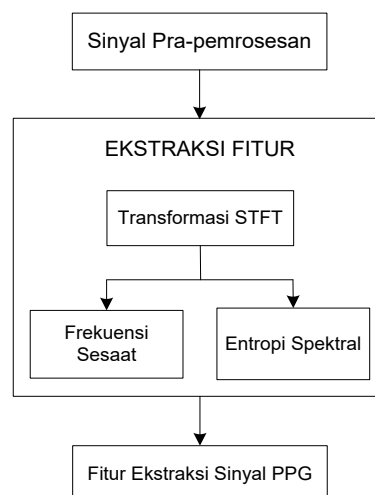
9. Keluarkan Hasil

Hasil yang diperoleh berupa nilai tengah lembah, titik a, titik b, titik c dan indeksnya.

10. Selesai Fungsi Deteksi Puncak Sistolik Dan Diastolik

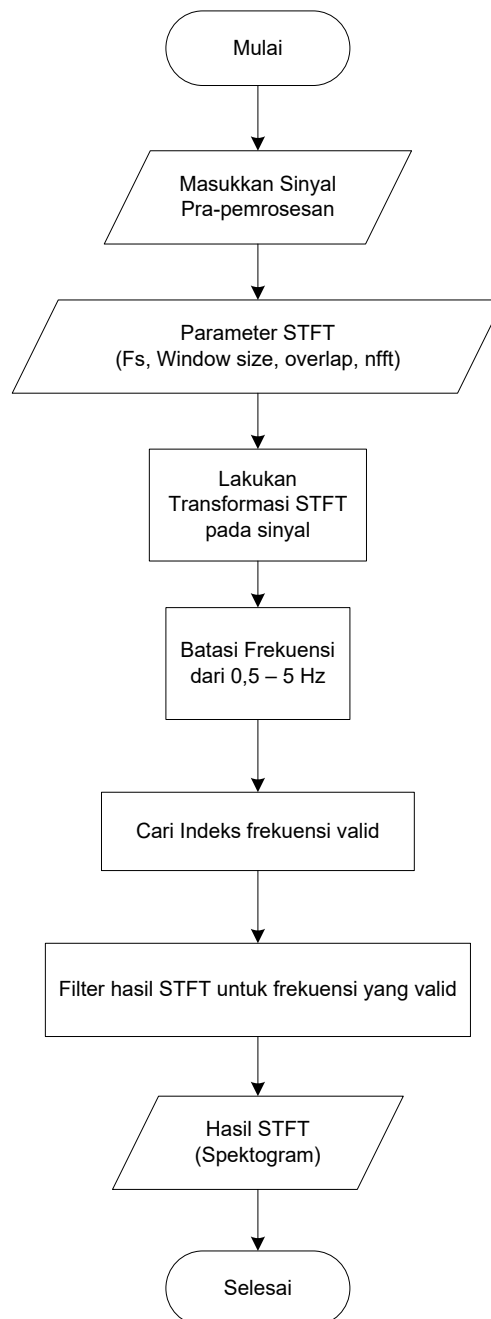
3.4 Diagram Alur Ekstraksi Fitur

Selanjutnya tahap ekstraksi fitur, tahap ini merupakan proses mengambil informasi penting berupa representasi numerik dari data PPG yang menggambarkan aspek kondisi fisiologis yang bermanfaat dari data tersebut. Data fitur ekstraksi yang diperoleh dapat dilakukan analisis lebih lanjut yaitu prediksi kadar gula darah seperti yang ditunjukkan pada gambar 3.14. Tahap ekstraksi fitur ini menggunakan analisis *time-frequency* yang bertujuan untuk mentransformasikan sinyal PPG dari domain waktu ke domain waktu-frekuensi. Salah satu metode untuk menganalisis *time-frequency* tersebut adalah menggunakan *Short-Time Fourier Transform* (STFT) dengan memilih parameter STFT yang akan digunakan. Hasil dari transformasi ini yaitu spektrogram. Dari spektrogram ini dapat terlihat komponen frekuensinya berubah seiring waktu. Setelah spektrogram diperoleh, maka langkah selanjutnya adalah mengekstraksi fitur yang terkandung dalam sinyal PPG yaitu frekuensi sesaat dan entropi spektral. Dimana fitur ini akan dijadikan sebagai bahan prediksi kadar gula darah.



Gambar 3. 14 Diagram Alur Ekstraksi Fitur Sinyal PPG.

3.4.1 Tranformasi Sinyal PPG (*Short-Time Fourier Transform*)



Gambar 3. 15 Diagram Alur Transformasi Sinyal PPG (STFT)

Pada Gambar 3.15, ditunjukkan langkah-langkah dalam proses transformasi Sinyal PPG menggunakan metode *Short-Time Fourier Transform*. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai Transformasi Sinyal PPG

2. Masukkan sinyal pra-pemrosesan

Sinyal yang dimasukkan pada proses ini adalah sinyal yang sudah melalui tahap pra-pemrosesan dan sudah terdeteksi adanya puncak sistolik dan diastolik.

3. Masukkan parameter STFT

Untuk melakukan transformasi sinyal PPG ini dibutuhkan beberapa parameter diantaranya frekuensi sampling (F_s), *Overlap*, *nfft* dan *window size*.

4. Lakukan Transformasi Sinyal PPG

Melakukan transformasi sinyal PPG menggunakan metode *Short-Time Fourier Transform*.

5. Batasi Frekuensi

Membatasi frekuensi pada rentang 0,5 Hz hingga 5 Hz bertujuan untuk memfokuskan sinyal PPG yang berkaitan dengan aktivitas sistem kardiovaskular, seperti denyut jantung dan variasi pembuluh darah sehingga memudahkan dalam proses ekstraksi fitur. Dengan membatasi frekuensi ini, gangguan dari *noise* atau komponen frekuensi yang tidak relevan dapat diminimalkan sehingga analisis menjadi spesifik terhadap informasi yang diinginkan.

6. Cari Indeks Frekuensi yang Valid

Mencari indeks frekuensi yang valid dilakukan untuk memastikan bahwa sinyal yang dianalisis berada pada rentang frekuensi yang sudah ditentukan.

7. *Filter* Hasil STFT untuk Frekuensi yang Valid

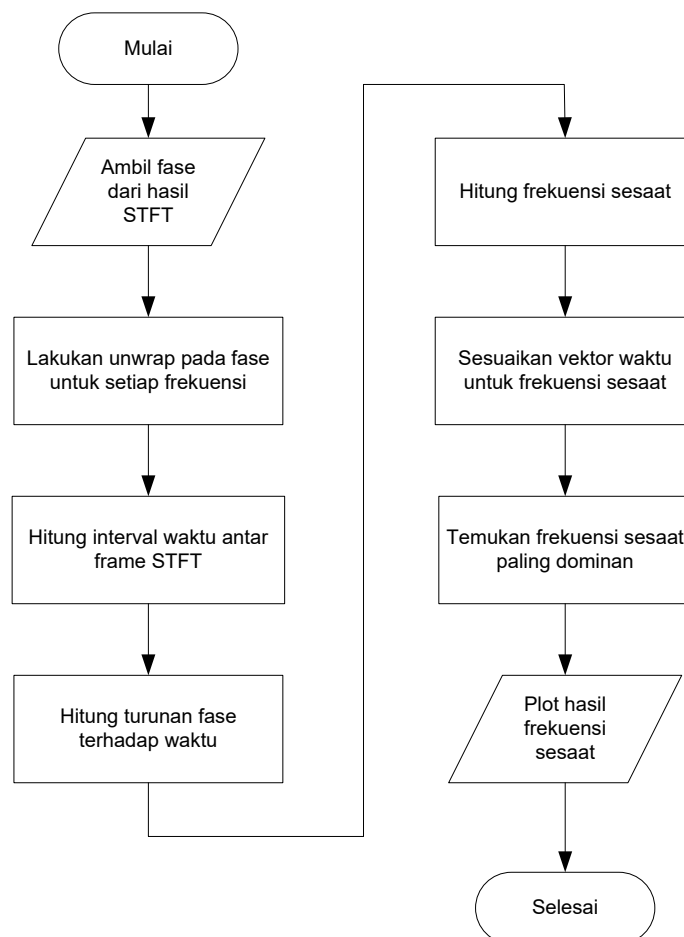
Penyaringan hasil transformasi sinyal berdasarkan indeks frekuensi yang valid bertujuan untuk memfokuskan analisis pada fitur yang relevan.

8. Hasil transformasi sinyal (STFT)

Hasil transformasi sinyal PPG menggunakan metode STFT ini direpresentasikan dalam bentuk spektrogram. Data sinyal PPG yang telah ditransformasikan serta di *filter* digunakan untuk analisis lebih lanjut yaitu perhitungan fitur ekstraksi untuk prediksi kadar gula darah.

9. Selesai Transformasi Sinyal PPG

3.4.2 Fitur Frekuensi Sesaat



Gambar 3. 16 Diagram Alur Frekuensi Sesaat

Pada Gambar 3.16, ditunjukkan langkah-langkah perhitungan frekuensi sesaat. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai Perhitungan Frekuensi Sesaat
2. Ambil Fase Dari Hasil STFT

Setelah STFT menghasilkan representasi frekuensi dari sinyal, data frekuensi ini memiliki amplitudo (kekuatan) dan fase (posisi gelombang). Pengambilan fase dilakukan karena frekuensi sesaat dihitung dari perubahan fase terhadap waktu.

3. Lakukan *Unwrap* Pada Fase untuk Setiap Frekuensi

Fase biasanya dibatasi pada rentang tertentu ($-\pi$ hingga π). Ketika fase melampaui batas tersebut, nilainya “melompat” kembali ke rentang awal. Untuk itu, proses unwrap diperlukan untuk memastikan fase terus meningkat atau menurun tanpa adanya nilai lompatan.

4. Hitung Interval Waktu Antar *Frame* STFT

Selanjutnya perhitungan interval waktu antar *frame* untuk menentukan kecepatan perubahan fase, karena di dalam proses STFT sinyal dipecah menjadi beberapa segmen pendek (*frame*).

5. Hitung Turunan Fase Terhadap Waktu

Turunan fase adalah perubahan fase dari satu *frame* ke *frame* berikutnya, dibagi dengan waktu antar *frame*. Hitung turunan ini menunjukkan seberapa cepat fase berubah terhadap waktu.

6. Hitung Frekuensi Sesaat

Hitung frekuensi sesaat menggunakan persamaan 2.8 yang ditunjukkan pada Bab 2. Frekuensi sesaat ini merupakan hasil dari turunan fase terhadap waktu, dibagi dengan konstanta 2π . Ini memberikan frekuensi aktual pada setiap waktu dalam sinyal hasil STFT.

7. Sesuaikan Vektor Waktu untuk Frekuensi Sesaat

Karena frekuensi sesaat dihitung dari perubahan antar *frame*, hasilnya sedikit berbeda dibandingkan sinyal asli. Untuk itu, diperlukan penyesuaian waktu agar grafik atau analisis frekuensi sesaat sesuai dengan waktu sebenarnya dari sinyal. Tanpa penyesuaian waktu ini, frekuensi sesaat akan terlihat seperti tidak sinkron dengan sinyal aslinya.

8. Temukan Frekuensi Sesaat yang Paling Dominan

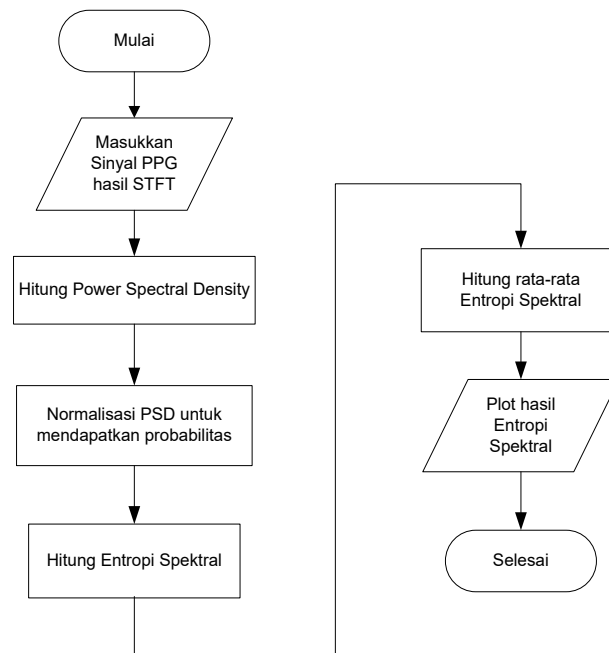
Dari semua frekuensi sesaat yang dihitung, di cari frekuensi sesaat paling dominan yaitu frekuensi yang paling sering muncul atau memiliki amplitudo terbesar.

9. Plot Hasil Frekuensi Sesaat

Menampilkan plot hasil frekuensi sesaat paling dominan. Nilai frekuensi sesaat paling dominan ini merupakan fitur ekstraksi yang akan digunakan dalam prediksi kadar gula darah digabungkan menjadi satu dataset.

10. Selesai Perhitungan Frekuensi Sesaat

3.4.3 Fitur Entropi Spektral



Gambar 3. 17 Diagram Alur Entropi Spektral

Pada Gambar 3.17, ditunjukkan langkah-langkah perhitungan entropi spektral. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai Perhitungan Entropi Spektral
2. Masukkan Sinyal PPG Hasil STFT

Masukkan sinyal PPG yang telah ditransformasikan menggunakan metode STFT. Sinyal ini sama dengan sinyal yang digunakan dalam perhitungan frekuensi sesaat.

3. Hitung *Power Spectral Density* (PSD)

Hitung *power spectral density* (PSD) untuk mendapatkan distribusi energi sinyal dalam domain frekuensi. PSD ini digunakan untuk mengetahui bagaimana energi sinyal tersebar di berbagai frekuensi.

4. Normalisasi PSD untuk Mendapatkan Probabilitas

Normalisasi dilakukan dengan membagi PSD pada setiap frekuensi dengan total PSD. Hal ini akan menghasilkan nilai probabilitas (antara 0 hingga 1) untuk setiap frekuensi, yang digunakan dalam perhitungan entropi. Probabilitas ini menjadi dasar untuk menghitung ketidakpastian dalam distribusi energi.

5. Hitung Entropi Spektral

Hitung entropi spektral menggunakan persamaan 2.9 yang ditunjukkan pada Bab 2.

6. Hitung Rata-Rata Entropi Spektral

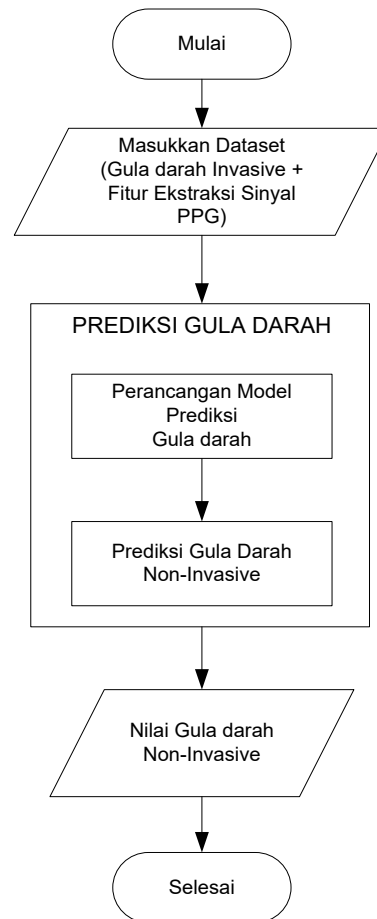
Setelah entropi spektral dihitung untuk setiap segmen waktu atau frekuensi, rata-rata dari semua nilai entropi spektral diambil untuk memberikan gambaran keseluruhan entropi sinyal.

7. Plot Hasil Entropi Spektral

Visualisasikan rata-rata entropi spektral dengan memplot hasil terhadap waktu atau frekuensi. Nilai rata-rata entropi spektral ini merupakan fitur ekstraksi yang akan digunakan dalam prediksi kadar gula darah digabungkan menjadi satu dataset.

8. Selesai Perhitungan Entropi Spektral

3.5 Diagram Alur Prediksi Gula Darah

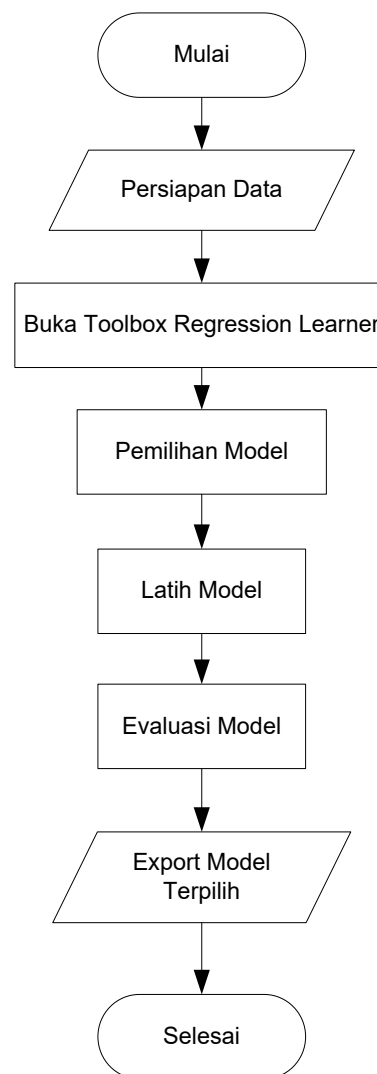


Gambar 3. 18 Diagram Alur Prediksi Gula Darah

Prediksi kadar gula darah dilakukan menggunakan *Toolbox Regression Learner* di MATLAB R2021a. Berdasarkan diagram alur yang ditunjukkan pada Gambar 3.18 langkah pertama yang dilakukan adalah mempersiapkan dataset yang akan digunakan untuk merancangan model prediksi. Dataset ini terdiri dari 2 nilai fitur ekstraksi dan nilai kadar gula darah yang di ukur secara invasif. Kedua, melakukan perancangan model prediksi dengan dataset yang sudah dibuat. Model prediksi yang digunakan adalah model regresi yang menghasilkan nilai evaluasi

terbaik. Ketiga, lakukan prediksi gula darah menggunakan model regresi terpilih sehingga didapatkan nilai gula darah non-invasif.

3.5.1 Diagram Alur Perancangan Model Prediksi Gula darah



Gambar 3. 19 Diagram Alur Perancangan Model Prediksi Gula Darah

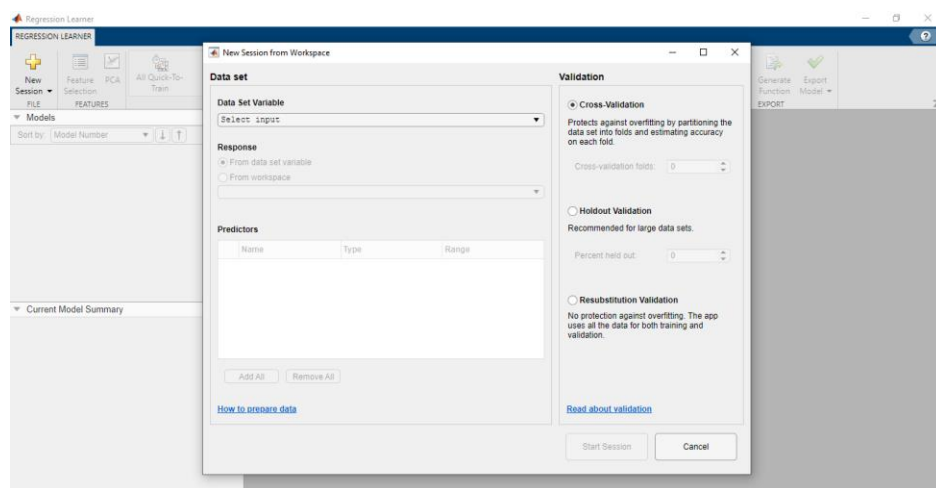
Pada gambar 3.19, ditunjukkan alur proses yang dilakukan dalam perancangan model prediksi kadar gula darah. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai Perancangan Model Prediksi

2. Persiapan Data

Dataset yang digunakan adalah data fitur ekstraksi dari sinyal PPG yang telah diproses sebelumnya dan data kadar gula darah invasif (target). Pisahkan dataset menjadi data latih (*training data*) dan data uji (*testing data*) dengan 70% untuk data latih dan 30% untuk data uji.

3. Buka *Regression Learner Toolbox*



Gambar 3. 20 Tampilan *Toolbox Regression Learner*

Membuka *regression learner toolbox* di MATLAB. Bisa dilakukan melalui menu Apps atau ketik *regressionLearner* di *Command window*. Kemudian klik *new session* dan pilih *from workspace* jika dataset berada di *workspace*. Pilih tabel data latih (*training data*) seperti yang ditunjukkan pada Gambar 3.20.

4. Pemilihan Model

Setelah data latih dimasukkan ke *toolbox*, tentukan variabel target dan fitur. Kemudian aktifkan juga opsi *validation* dengan menggunakan *5-Fold Cross-Validation* untuk menghindari *overfitting*. Setelah itu, gunakan opsi *All*

Models di *Regression Learner* untuk mencoba berbagai model regresi yang tersedia, seperti *Linear Regression*, *Support Vector Machines* (SVM), *Decision Trees*, *ensemble*, dan *Gaussian Process Regression* (GPR).

5. Latih Model

Klik *Train All* untuk melatih semua model. Kemudian hasil pelatihan dapat diamati seperti *Root Mean Squared Error* (RMSE), *R-squared*, dan lain-lain untuk memilih model terbaik.

6. Evaluasi Model

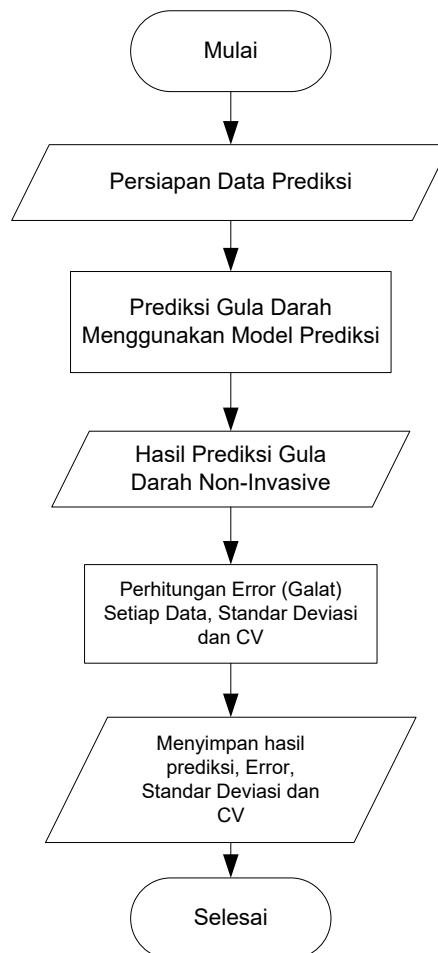
Gunakan data uji (*testing data*) untuk mengevaluasi performa model. Evaluasi hasil pengujian dilihat dari nilai RMSE dari setiap model yang di uji. Model dengan nilai RMSE terkecil dipilih sebagai model terbaik.

7. *Export* Model terpilih

Model terpilih kemudian di *export* ke *workspace* MATLAB dan disimpan sebagai variabel (misalnya, *trainedModel*) atau disesuaikan dengan nama variabel yang diinginkan. Variabel ini digunakan sebagai fungsi prediktor untuk melakukan prediksi kadar gula darah secara non-invasif.

8. Selesai Perancangan Model Prediksi

3.5.2 Diagram Alur Prediksi Gula Darah *Non-Invasive*



Gambar 3. 21 Diagram Alur Prediksi Gula darah *Non-invasive*

Pada gambar 3.21, ditunjukkan alur proses yang dilakukan dalam prediksi gula darah secara *non-invasive*. Berikut adalah penjelasan rinci dari setiap langkah dalam diagram alur tersebut :

1. Mulai Prediksi Gula Darah *Non-Invasive*
2. Persiapan Data Prediksi

Data prediksi ini merupakan data fitur ekstraksi sinyal PPG (frekuensi sesaat dan entropi spektral) dan data gula darah secara *invasive* Data prediksi ini di *import* ke MATLAB untuk dilakukan prediksi.

3. Prediksi Gula Darah *Non-Invasive* menggunakan Model Terpilih

Gunakan model terpilih untuk memprediksi gula darah dengan mengetik *trainedModel.predictFcn(Dataprediksi)* pada *command window*. Setelah itu, hasil prediksi akan muncul pada *command window*.

4. Hasil Prediksi Gula Darah *Non-Invasive*

Hasil prediksi ditampilkan langsung dalam *command window* dan disimpan menjadi variabel gula darah *non-invasive*.

5. Perhitungan *Error* (Galat) Setiap Data dan Akurasi Pengukuran

Hitung *error* pada setiap data dengan :

- 1) Menghitung selisih antara nilai gula darah *non-invasive* (prediksi) dengan nilai gula darah *invasive* (*gold standar*) :

$$\text{Selisih} = (\text{nilai gula darah non-invasif}) - (\text{nilai gula darah invasif})$$

- 2) Hitung persentase *error* dengan persamaan 2.16 yang tertera pada Bab 2.

- 3) Hitung standar deviasi dari keseluruhan data menggunakan persamaan 2.17 yang tertera pada Bab 2.

- 4) Hitung koefisien variasi (CV) dari nilai standar deviasi dan rata-rata *error* menggunakan persamaan 2.18 yang tertera pada Bab 2.

6. Menyimpan Hasil Prediksi, Error, Standar Deviasi dan CV

Buat tabel untuk menyimpan data hasil prediksi dan *error* dengan nama HasilPrediksi yang berisi nilai gula darah *invasive*, gula darah *non-invasive* dan persentase *error* (galat). Selain itu, simpan hasil perhitungan nilai standar deviasi serta koefisien variasinya (CV).

7. Selesai Prediksi Gula Darah *Non-Invasive*.