



UNIVERSITAS
Dinamika

**KLASIFIKASI SINYAL JANTUNG PHONOCARDIOGRAM
MENGUNAKAN METODE *LONG SHORT-TERM MEMORY (LSTM)***

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

MISKIYANTO

17410200044

FAKULTAS TEKNOLOGI DAN INFORMATIKA

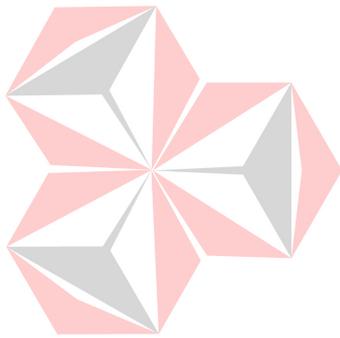
UNIVERSITAS DINAMIKA

2021

**KLASIFIKASI SINYAL JANTUNG PHONOCARDIOGRAM
MENGUNAKAN METODE *LONG SHORT-TERM MEMORY (LSTM)***

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



**UNIVERSITAS
Dinamika**

Disusun Oleh:

**Nama : Miskiyanto
NIM : 17410200044
Program Studi : S1 Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2021

TUGAS AKHIR

KLASIFIKASI SINYAL JANTUNG PHONOCARDIOGRAM MENGUNAKAN METODE *LONG SHORT-TERM MEMORY (LSTM)*

Dipersiapkan dan disusun oleh:

Miskiyanto

NIM : 17410200044

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 31 Agustus 2021

Susunan Dewan Pembahas

Pembimbing:

- I. Dr.Jusak
NIDN: 0708017101
- II. Ira Puspasari, S.Si., M.T.
NIDN: 0710078601

Pembahas:

Pauladie Susanto, S.Kom., M.T.
NIDN: 0729047501


Digitally signed by Universitas Dinamika
Date: 2021.08.31 21:40:35 +08'00'


Digitally signed by Universitas Dinamika
Date: 2021.08.31 09:54:35 +07'00'


Digitally signed by Universitas Dinamika
Date: 2021.09.02 13:51:36 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana


Digitally signed by Universitas Dinamika
Date: 2021.09.02 14:16:12 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



*“Jika ingin tahu kompleksnya algoritma masalah dalam kehidupan, maka
menikahlah!”*

UNIVERSITAS
Dinamika

Kupersembahkan kepada eppak, emmak, yang selalu memberi dukungan, doa, dan kontribusi besar dalam diri saya hingga mampu menyelesaikan tulisan ini. Serta semua orang tanpa terkecuali yang telah ikut dalam proses penulisan ini .



UNIVERSITAS
Dinamika

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Miskiyanto
NIM : 17410200044
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **KLASIFIKASI SINYAL JANTUNG
PHONOCARDIOGRAM MENGGUNAKAN METODE
LONG SHORT-TERM MEMORY (LSTM)**

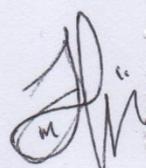
Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui **memberikan** kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 13 Juli 2021

Yang menyatakan




Miskiyanto

NIM : 17410200044

ABSTRAK

Jantung adalah organ vital yang digunakan dalam kegiatan sehari-hari. Namun, berdasarkan survei World Health Organization (WHO), terdapat 33% kematian disebabkan oleh Penyakit Jantung Koroner (PJK). Pemeriksaan jantung secara rutin diharapkan dapat membantu pasien dalam memonitoring kondisi jantungnya. Teknik untuk mendengarkan suara jantung dengan menggunakan elektronik atau tradisional *stethoscope*, sebuah metode lama namun sangat efektif dalam melakukan diagnosis terhadap sejumlah penyakit kardiovaskular. Namun, hasil pemeriksaan yang didasarkan pendengaran dokter, juga menjadi kendala dalam menentukan hasil pemeriksaan jantung, karena merupakan hasil subjektifitas. Hal ini menjadikan analisis pendeteksian terhadap karakteristik sinyal suara jantung secara otomatis penting untuk dilakukan supaya tidak terjadi kesalahan diagnosa pada saat perekaman sinyal suara jantung. Dengan berkembangnya teknik klasifikasi secara otomatis dengan menggunakan *machine learning* maupun *deep learning*, telah banyak upaya yang dilakukan untuk menganalisa sinyal PCG. Dengan adanya hal tersebut, maka penulis membuat sebuah sistem untuk melakukan klasifikasi secara otomatis terhadap sinyal jantung PCG. Metode yang digunakan pada penelitian ini adalah *Long Short – Term Memory* (LSTM) dengan menggunakan variasi dataset sebanyak 3240 yang terbagi kedalam 70% data latih dan 30% data validasi. Akurasi yang didapatkan dari metode LSTM dengan menggunakan 5 hidden layer sebesar 91% untuk akurasi *training* dan 81.7% untuk nilai akurasi *test*. Dengan dilakukan sebanyak beberapa kali percobaan didapatkan hasil semakin banyak jumlah layer disetiap layer maka waktu yang dibutuhkan untuk proses training data juga semakin lama.

Kata Kunci: *Long Short – Term Memory, Phonocardiogram, Deep Learning*

KATA PENGANTAR

Segala puji bagi Tuhan, karena dengan rahmatNya penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “Klasifikasi Sinyal Jantung Phonocardiogram Menggunakan Metode *Long Short-Term Memory* (LSTM)”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Teknik pada program studi S1 Teknik Komputer Universitas Dinamika. Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

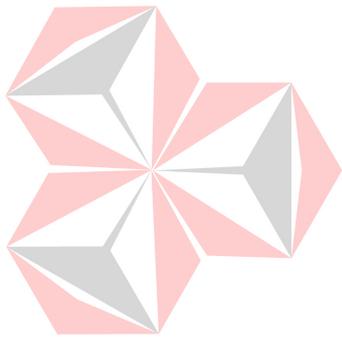
1. Orang Tua penulis, yang telah memberikan kontribusi besar atas apa yang telah penulis lakukan hingga dapat menyelesaikan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika, dan selaku Penguji dalam Tugas Akhir ini.
4. Bapak Dr. Jusak selaku Dosen pembimbing I yang selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Ibu Ira Puspasari, S.Si., M.T., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Seluruh rekan Tim Penelitian Identifikasi Kelainan Sinyal Suara Jantung, terimakasih karena telah memberikan kesempatan kepada penulis untuk bergabung dan melakukan penelitian terhadap sinyal suara jantung.
7. Seluruh rekan Tim Robotika Universitas Dinamika yang telah memotivas penulis dalam menyelesaikan Tugas Akhir ini.

8. Untuk Ekasari Oktarina, S.Kom., yang telah menjadi teman diskusi dalam Tugas Akhir ini.

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun

Surabaya, 22 Juni 2021

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvi
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
BAB II	5
LANDASAN TEORI.....	5
2.1 Jantung.....	5
2.2 Suara Jantung	5
2.3 Prinsip Kerja Jantung	6
2.4 Fast Fourier Transform.....	7
2.5 Phonocardiogram (PCG)	8
2.6 <i>Python</i>	8
2.7 Keras.....	9

2.8	Tensorflow.....	9
2.9	Google Colab.....	10
2.10	Long Short-Term Memory Neural Network (LSTM)	11
2.11	Pengujian Klasifikasi Sinyal.....	13
BAB III.....		15
METODOLOGI PENELITIAN		15
3.1	Model Perancangan	15
3.1.1	Sinyal Normal dan Abnormal	15
3.1.2	Pre-processing.....	16
3.1.3	Proses	16
3.1.4	Analisis Data	16
3.2	Model Program.....	17
BAB IV		22
HASIL DAN PEMBAHASAN		22
4.1	Pengujian Segmentasi.....	22
4.1.1	Tujuan Pengujian Segmentasi	22
4.1.2	Alat Yang Digunakan Untuk Pengujian Segmentasi	22
4.1.3	Prosedur Pengujian Program Segmentasi	23
4.1.4	Hasil Pengujian Program Segmentasi	23
4.2	Pengujian Fast Fourier Transform (FFT)	24
4.2.1	Tujuan Pengujian FFT	24
4.2.2	Alat Yang Digunakan Untuk Pengujian Segmentasi	24
4.2.3	Hasil Pengujian FFT	24
4.3	Pengujian Pelatihan Data Sinyal Suara Jantung PCG Dengan LSTM ...	26
4.3.1	Tujuan Proses Pelatihan	26
4.3.2	Alat yang dibutuhkan dalam proses pengujian pelatihan.....	26

4.3.3	Prosedur pengujian proes pelatihan	26
4.4	Mencoba Hasil Model Latih.....	38
4.5	Analisa Hasil	38
BAB V	40
PENUTUPAN	40
5.1	Kesimpulan.....	40
5.2	Saran	41
DAFTAR PUSTAKA	42
LAMPIRAN	45
Lampiran 1	Source Code Program	45
Lampiran 2	Hasil Turnitin.....	55
BIODATA PENULIS	64

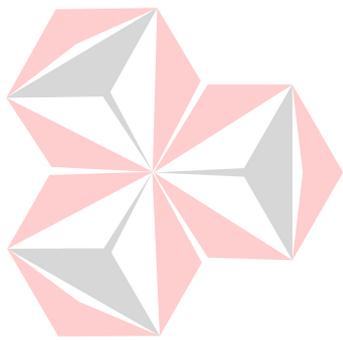


UNIVERSITAS
Dinamika

DAFTAR GAMBAR

Gambar 2. 1 Sinyal Suara Jantung Phonocardiogram.....	6
Gambar 2. 2 Anatomi Jantung	7
Gambar 3. 1 Perancangan kalsifikasi sinyal jantung PCG.....	15
Gambar 3. 2 Proses training sinyal jantung PCG.....	16
Gambar 3. 3 Blok diagram program pelatihan dengan LSTM.....	17
Gambar 3. 4 Plotting sinyal suara jantung PCG wav original.....	18
Gambar 3. 5 Plot gambar sinyal FFT.....	19
Gambar 4. 1 Program segmentasi.....	22
Gambar 4. 2 Hasil plotting sinyal suara jantung PCG tanpa proses segmentasi....	23
Gambar 4. 3 Hasil plotting segmentasi sinyal suara jantung PCG.....	23
Gambar 4. 4 Program FFT (<i>Fast Fourier Transform</i>).....	24
Gambar 4. 5 Hasil plotting sinyal suara jantung PCG segmentasi.....	25
Gambar 4. 6 Hasil plotting sinyal suara jantung PCG FFT	25
Gambar 4. 7 Hasil grafik percobaan pertama.....	28
Gambar 4. 8 <i>Confusion Matrix</i> percobaan pertama	29
Gambar 4. 9 Hasil grafik percobaan kedua.....	31
Gambar 4. 10 <i>Confusion Matrix</i> percobaan kedua.....	32
Gambar 4. 11 Hasil grafik percobaan ketiga.....	34
Gambar 4. 12 Konfusi matrix percobaan ketiga	34
Gambar 4. 13 Hasil grafik percobaan keempat.....	36
Gambar 4. 14 Konfusi matrix percobaan keempat.....	37
Gambar 4. 15 Konfusi matrix ujicoba model.....	38
Gambar L2. 1 Hasil Plagiasi Halaman 1.....	55
Gambar L2. 2 Hasil Plagiasi Halaman 2.....	56
Gambar L2. 3 Hasil Plagiasi Halaman 3.....	57
Gambar L2. 4 Hasil Plagiasi Halaman 4.....	58
Gambar L2. 5 Hasil Plagiasi Halaman 5.....	59

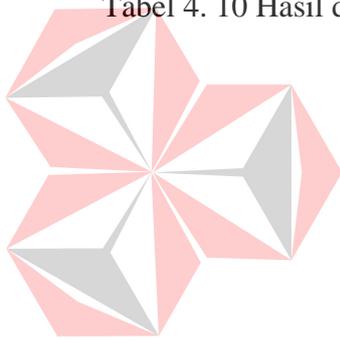
Gambar L2. 6 Hasil Plagiasi Halaman 6.....	60
Gambar L2. 7 Hasil Plagiasi Halaman 7.....	61
Gambar L2. 8 Hasil Plagiasi Halaman 8.....	62
Gambar L2. 9 Hasil Plagiasi Halaman 9.....	63



UNIVERSITAS
Dinamika

DAFTAR TABEL

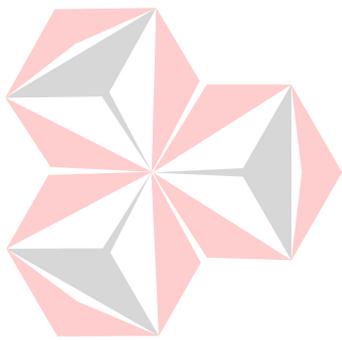
Tabel 3. 1 <i>Confusion matrix</i> hasil klasifikasi.....	20
Tabel 4. 1 Data input pelatihan	27
Tabel 4. 2 Hasil training pada percobaan pertama	27
Tabel 4. 3 Performa matrik percobaan pertama	29
Tabel 4. 4 Hasil training pada percobaan kedua	30
Tabel 4. 5 Performa matrik percobaan kedua	32
Tabel 4. 6 Hasil training pada percobaan ketiga	33
Tabel 4. 7 Performa matrik percobaan ketiga	35
Tabel 4. 8 Hasil training pada percobaan keempat	35
Tabel 4. 9 Performa matrik percobaan keempat	37
Tabel 4. 10 Hasil dari keempat percobaan	39



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1	Source Code Program	45
Lampiran 2	Hasil Turnitin.....	55



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jantung adalah organ vital yang digunakan dalam kegiatan sehari-hari. Namun, berdasarkan survei World Health Organization (WHO), terdapat 33% kematian disebabkan oleh Penyakit Jantung Koroner (PJK) (Organization, n.d.). Pemeriksaan jantung secara rutin diharapkan dapat membantu pasien dalam memonitoring kondisi jantungnya.

Terdapat setidaknya 3 teknik dapat digunakan dalam melakukan monitoring jantung. Elektrokardiogram (EKG) adalah teknik yang paling populer digunakan dengan melakukan perekaman terhadap aktifitas elektronik yang terdapat pada jantung, Sementara *photoplethysmography* (PPG), merupakan teknik perhitungan laju aliran darah [Nichols, 2019] dengan menggunakan sensor cahaya sebagai instrumen pengukuran. Teknik monitoring jantung yang lain adalah *phonocardiogram* (PCG), merupakan teknik perekaman sinyal suara jantung beserta murmur yang terbentuk selama siklus jantung bekerja. Aktivitas mekanisme jantung menghasilkan beberapa suara diantaranya adalah pada suara jantung normal terdapat dua ketukan yang dikenal sebagai suara jantung pertama (S1), dan suara jantung ke dua (S2), dan terdapat suara jantung lain dikenal dengan suara jantung abnormal ketiga (S3), suara jantung abnormal keempat (S4), murmur yang muncul ketika jantung dalam keadaan abnormal [Ismail, 2018] Tingkat kompleksitas yang terkandung pada PCG membuat PCG lebih unggul dibandingkan kedua teknik lainnya (W Phanphaisarn, 2011).

Teknik untuk mendengarkan suara jantung dengan menggunakan elektronik atau tradisional *stethoscope*, sebuah metode lama namun sangat efektif dalam melakukan diagnosis terhadap sejumlah penyakit kardiovaskular. Namun, hasil pemeriksaan yang didasarkan pendengaran dokter, juga menjadi kendala dalam menentukan hasil pemeriksaan jantung, karena merupakan hasil subjektifitas. Hal ini menjadikan analisis pendeteksian terhadap karakteristik sinyal suara jantung secara otomatis penting untuk dilakukan supaya tidak terjadi kesalahan diagnosa pada saat perekaman sinyal suara jantung.

Telah dilakukan beberapa penelitian untuk melakukan identifikasi sinyal suara jantung secara otomatis untuk dapat menentukan posisi S1 dan S2, juga murmur yang terdapat pada sinyal jantung PCG. Beberapa penelitian telah dilakukan yaitu oleh (Jusak, et al., 2020) dengan melakukan identifikasi S1 dan S2 terhadap 75 siklus sinyal jantung pada bayi normal. Durasi rata-rata T11 dan durasi rata-rata T12 dalam penelitian ini dapat digunakan sebagai acuan untuk menentukan durasi sinyal suara jantung janin normal. Pada tahun 2019 telah dilakukan sebuah penelitian untuk mengetahui letak murmur dan juga S1 dan S1 pada sebuah sistem IoT. (Puspasari, et al., 2019). Pengembangan CEEMD dilakukan pada penelitian yang dilakukan oleh jusak dengan melakukan identifikasi secara semi otomatis pada sinyal suara jantung yaitu dengan metode Largest Interval Heart Sounds Detection (LiHSD) menghasilkan nilai akurasi suara jantung normal menunjukkan 98%, sedangkan untuk anomali suara jantung sampel deteksi menghasilkan nilai akurasi mulai dari 89% hingga 97,5%. (Jusak, et al., 2021)

Dengan berkembangnya teknik klasifikasi secara otomatis dengan menggunakan *machine learning* maupun *deep learning*, telah banyak upaya yang dilakukan untuk menganalisa sinyal PCG secara otomatis seperti yang dilakukan oleh Tschanen pada tahun 2016 dengan menggunakan data pada *physionet* dengan menunjukkan hasil tingkat sensitivitas, spesififikasi dan skore sebesar 96%, 83% dan 89% (M, et al., 2016) Pada penelitian yang dilakukan Chen dengan metode DNN menghasilkan tingkat akurasi sebesar 91.12% (T, et al., 2017). Kompleksitas pada sinyal jantung PCG menjadi tantangan terbesar dalam proses klasifikasi, sehingga pada tahun 2019, dilakukan sebuah penelitian segmentasi sinyal PCG (Behbani, 2019). Penelitian tersebut menunjukkan hasil yang baik, namun waktu yang digunakan dalam melakukan proses cenderung lama. Banyaknya penelitian tentang klasifikasi sinyal jantung telah dirangkum oleh Suyi Li pada tahun 2018 (Li, et al., 2020).

Long Short-Term Neural Network (LSTM) salah satu metode *machine learning* pada kategori RNN. Dengan melakukan proses masukan berupa data sekuensial. LSTM didesain untuk mempelajari data apa yang akan dipakai maupun yang tidak dipakai. Karena LSTM mempunyai sebuah neuron yang didalamnya memiliki gates, setiap gate berfungsi untuk mengatur memori pada setiap neuron

itu sendiri. Karena LSTM melakukan proses pengolahan dengan data masukan berupa data sekuensial, maka LSTM banyak digunakan untuk pemrosesan audio, teks, video, dan data time series.

Banyaknya penelitian yang dilakukan menunjukkan bahwa klasifikasi terhadap sinyal PCG sangat penting dilakukan. Maka, dengan melakukan klasifikasi sinyal jantung dengan menggunakan metode LSTM diharapkan hasil penelitian ini dapat membantu paramedis dalam melakukan identifikasi pada jantung pasien normal dan abnormal secara lebih cepat, dan memiliki akurasi yang tinggi.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan dalam Tugas Akhir ini yaitu: bagaimana melakukan klasifikasi sinyal jantung dalam bentuk PCG dengan menggunakan metode LSTM untuk dikategorikan sebagai sinyal jantung normal dan abnormal.

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut yaitu:

1. Dataset sinyal jantung berasal dari *physionet*.
2. Kondisi jantung pasien abnormal ditentukan dengan munculnya murmur pada hasil klasifikasi.
3. Jumlah sample 3240 data normal dan abnormal *phonocardiogram*

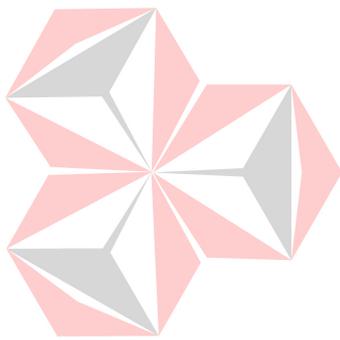
1.4 Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah di atas, tujuan penelitian dalam Tugas Akhir adalah melakukan klasifikasi sinyal jantung dalam bentuk PCG dengan menggunakan metode LSTM untuk dikategorikan sebagai sinyal jantung normal dan abnormal.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut :

1. Proses pendeteksian penyakit kardiovaskular dengan lebih cepat dan efektif, sehingga pasien tidak perlu melakukan banyak tahapan untuk memonitoring keadaan jantungnya.
2. Pengambilan keputusan secara lebih objektif diharapkan dapat membantu para dokter dalam melakukan analisis terhadap pasien dengan adanya klasifikasi sinyal jantung secara otomatis.



UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 Jantung

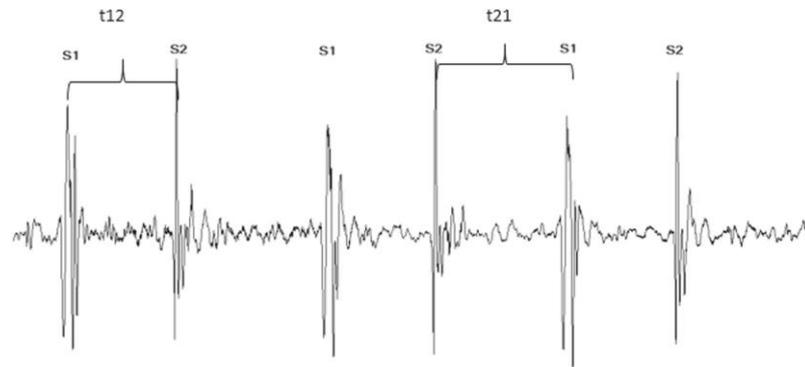
Jantung organ penting dalam tubuh bekerja memompa darah sehingga mengalir keseluruh tubuh dan mengembalikan kejantung. Jantung terdiri dua bagian yaitu serambi (atrium) pada bagian atas, dan bilik (ventricile) pada bagian bawah. Darah dipompa dari satu ruangan ke ruangan yang lain menggunakan otot jantung. Katup jantung akan membuka tiap terjadi proses pemompaan hingga darah dapat mengalir menuju ruangan yang dituju. Kemudian katup menutup dalam mencegah akirah darah yang berbalik (Maisyaroh, 2012).

2.2 Suara Jantung

Suara lub-dub adalah dua suara berbeda yang dihasilkan dari detak jantung yang dapat didengarkan melalui stetoskop. Suara jantung pertama (S1) atau yang disebut juga dengan suara lub yang disebabkan oleh penutupan katup triscupid dan mitral (atrioventrikular), dan mencegah lairah dalarah membalik. Penutupan katu dapat menimbulkan suara dub yang kemudian disebut suara jantung kedua (S2), semilunar (*aortic* dan *pulmonary*) yang membebaskan darah ke sistem sirkulasi paru-paru dan sistemik. (Maisyaroh, 2012).

Aktivitas mekanis jantung dapat didengar menggunakan stetoskop tradisional atau elektronik. Auskultasi atau mendengarkan sinyal suara jantung adalah metode lama tetapi sangat efektif diagnosis untuk sejumlah penyakit kardiovaskular. Aktivitas ini dapat digunakan untuk mendengarkan suara yang dihasilkan oleh aktifitas jantung (Ismail, et al., 2018).

Pada dasarnya dalam keadan normal bunyi suara jantung terdiri dari dua yaitu S1 dan S2 seperti ditunjukkan Gambar 2.1.

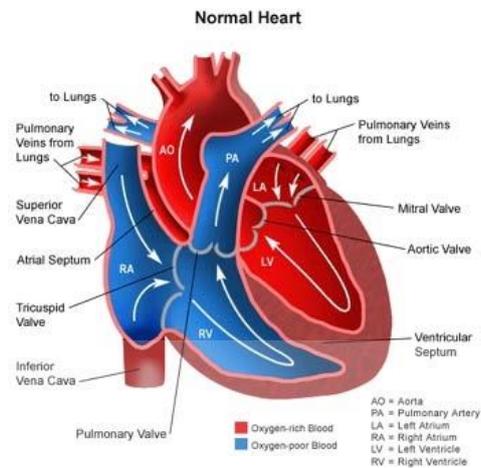


Gambar 2. 1 Sinyal Suara Jantung Phonocardiogram
 Sumber : (Ismail, et al., 2018)

2.3 Prinsip Kerja Jantung

Jantung bekerja melalui suatu mekanisme yang berulang dan terjadi terus menerus, disebut juga dengan siklus jantung, sehingga dapat dilihat secara visual atau disebut dengan detak jantung. Frekuensi sinyal suara jantung umumnya antara 20-200 Hz. Melalui mekanisme bolak-balik, jantung berkontraksi untuk mengosongkan isi jantung, bersantai dan mengisi kembali darah. Secara sirkulasi, jantung melewati masa sistolik, masa kontraksi dan pengosongan isinya (darah), dan masa diastolik, masa relaksasi dan pengisian jantung dengan darah. Kedua atrium (atrium) melebar dan berkontraksi pada saat yang sama, dan kedua ventrikel (ventrikel) juga melebar dan berkontraksi pada saat yang sama untuk melakukan mekanisme ini. (Barclay, 2020).

Dua sisi jantung yaitu sisi kanan dan kiri berfungsi sebagai pompa yang terpisah, walaupun secara anatomik jantung adalah satu organ. Jantung dibagi menjadi dua bagian, yaitu sisi kanan dan kiri yang terbagi lagi menjadi empat ruang. Darah yang kembali ke jantung diterima oleh bilik bagian atas (atrium). Dan dipindahkan ke bilik bagian bawah (ventrikel), yang bertugas memompa darah dari jantung. Vena adalah pembuluh darah yang mengembalikan darah dari jaringan ke atrium. Arteri adalah pembuluh darah yang berfungsi mengangkut darah untuk menjauhi ventrikel. (Barclay, 2020) seperti yang terlihat pada Gambar 2.2.



Gambar 2. 2 Anatomi Jantung
Sumber : (Barclay, 2020)

2.4 Fast Fourier Transform

Fast Fourier Transform (FFT) ditemukan oleh J. Fourier pada tahun 1822 kemudian dikembangkan pada 1965. Prinsip kerja *fourier transform* yaitu membagi sinyal menjadi frekuensi yang berbeda, dalam fungsi eksponensial yang kompleks. Merupakan metode yang sangat efektif, untuk menghitung koefisien barisan berhingga dari Fourier diskrit ke data kompleks (Sipasulta, et al., 2014). Karena menyimpan lebih banyak waktu daripada metode tradisional, transformasi Fourier cepat merupakan aplikasi penting dari hasil penelitian di berbagai bidang, seperti analisis spektrum, pemrosesan sinyal suara dan optik, dan desain filter digital. Prinsip utama dari algoritma FFT adalah menguraikan perhitungan transformasi Fourier diskrit dari barisan sepanjang N menjadi transformasi Fourier diskrit kontinu yang lebih kecil dan transformasi Fourier diskrit yang lebih kecil. Cara menerapkan prinsip ini telah menghasilkan berbagai algoritma yang berbeda, yang semuanya dapat meningkatkan kecepatan perhitungan.

Fast Fourier Transform merupakan algoritma, untuk menghitung Transformasi Fourier diskrit dengan cepat dan efisien. Banyaknya sinyal dalam sistem komunikasi yang kontinu, transformasi *fourier* digunakan untuk sinyal kontinu. Transformasi Fourier didefinisikan dengan rumus berikut:

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft} dt$$

Dimana $s(f)$ sinyal domain frekuensi, $s(t)$ sinyal domain waktu, dan $e^{-j2\pi ft}$ konstanta sebuah sinyal, f frekuensi dan t untuk waktu.

2.5 Phonocardiogram (PCG)

Phonocardiogram (PCG) atau disebut stetoskop elektrik, adalah perangkat yang digunakan untuk mendengar suara jantung. *Phonocardiogram* adalah teknik menelusuri suara jantung dan merekam getaran. Melalui sebuah transduser mikrofon suara jantung direkam dan ditampilkan pada osiloskop.

Dalam mengindikasi laju dan ritme jantung saat memompa darah, dapat menggunakan suara – suara jantung yang direkam. Selain itu juga memberikan informasi tentang efektifitas pemompaan jantung. Dan juga dapat memberi informasi aktifitas katup-katup jantung. Dengan ditampilkan secara visual suara jantung dapat digunakan lebih efisien oleh dokter (Debbal, 2008).

Lub-dub atau suara jantung pertama (S1) dan suara jantung kedua (S2), dihasilkan oleh suara jantung. Suara pertama (S1) dapat disebabkan oleh dua hal yaitu, penutupan katup atrioventrikular dan kontraksi otot – otot jantung. Suara yang kedua (S2) terjadi karena penutupan katup semilunaris. Jarak antara suara jantung pertama (S1) dan suara jantung (S2) memiliki selisih dimana suara jantung pertama (S1) mempunyai waktu sedikit lebih lama dibanding (S2) (Debbal, 2008). Interval diantara kedua suara sinyal jantung terdapat disebut sistole dan diastole.

2.6 Python

Merupakan bahasa pemrograman tinggi yang diciptakan oleh Guido Van Rossum pada tahun 1989 di Amsterdam, Belanda. Bahasa pemrograman python sangatlah mudah dalam penulisan *syntax* programnya. Python juga merupakan bahasa pemrograman yang dapat berjalan diberbagai sistem operasi. Python dapat berjalan pada sistem operasi WINDOWS, UNIX, Linux, dan IOS. Python bahasa pemrograman dengan portabilitas yang tinggi.

Bahasa pemrograman python adalah jenis pemrograman yang banyak diminati untuk saat ini karena kesederhanaannya. Meskipun sederhana tapi pustaka yang ada sangat lengkap misalnya modul – modul untuk keperluan jaringan, antarmuka grafis, analisis dan komputasi numerik, *Hypertext* (HTML, XML, dan lain-lain),

akses *database*, dan berbagai hal lainnya. Kode bahasa pemrograman python mudah untuk dibaca baik oleh yang sudah terbiasa dengan bahasa pemrograman lain maupun yang baru memulai. Kemudahan bahasa pemrograman python karena python sendiri menggunakan interpreter sebagai penerjemahnya. Bahasa pemrograman python disediakan gratis yang bisa didapat pada situs www.python.org.

2.7 Keras

Keras adalah level tertinggi untuk *framework* pada *Deep Learning* yang digunakan Python, dimana kapasitasnya berjalan diatas Theano dan Tensorflow. Dibuat oleh Francois Chollet, keuntungan terpenting dalam penggunaan Keras adalah untuk menghemat waktu dengan API tingkat tinggi yang mudah dan handal digunakan saat pembuatan prototipe dengan sebuah ide yang cepat. Keras memungkinkan kita mengkonstruksi Tensorflow dan Theano dengan cara lebih mudah untuk dipahami dan mudah digunakan tanpa harus dengan cara menulis kode *boilerplate* berlebih untuk membangun model berbasis *Neural Networks*. Kemudahan fleksibilitas dan kesederhanaan adalah alasan utama yang menjadikan popularitas Keras. Selain memberikan akses yang mudah untuk *esoteric library*, Keras memastikan kita bisa dapat memanfaatkan *library* yang ditawarkan.

2.8 Tensorflow

Tensorflow adalah putaka perangkat lunak sumber terbuka untuk *machine learning* yang dirilis oleh Google pada bulan November 2015. Tensorflow didasarkan pada sistem internal yang dipakai oleh Google dalam memberdayakan penelitian dan sistem produksinya. Tensorflow dibuat sebagai upaya Google dalam mengembangkan Theano dengan menyediakan antarmuka yang mudah dipahami saat digunakan pada *Deep Learning*, *neural networks*, dan *machine learning* yang berfokus pada pembuatan prototype yang cepat dan penerapan konstruksi model. Seperti halnya Theano pada Tensorflow juga menyediakan fungsi untuk simbol matematika, yang kemudian diterjemahkan kedalam grafik komputasi. Grafik ini kemudian dikompilasi menjadi bahasa pemrograman dengan tingkatan yang lebih rendah dan dieksekusi secara efisien. Seperti Theano, Tensorflow juga mendukung CPU dan GPU tanpa hambatan. Namun pada faktanya Tensorflow akan bekerja

lebih baik pada TPU, atau yang dikenal dengan *Tensor Processing Unit* yang ditemukan oleh Google. Selain memiliki Python API, Tensorflow juga terbuka untuk API's untuk C++, Haskell, Java, dan Bahasa Pemrograman Go. Salah satu perbedaan utama yang dimiliki Tensorflow dibandingkan dengan Theano adalah dukungan untuk operasi tingkat tinggi, yang dapat memudahkan dalam proses *machine learning*, dan berfokus pada pengembangan model serta penerapan pada produksi dan penyajian model beberapa mekanisme (https://tensorflow.org/serving/serving_basic). Juga dokumentasi dan penggunaannya Theano sulit dipahami, namun tidak dengan Tensorflow yang mudah di pahami dan diimplementasikan. Untuk lebih detil tentang dokumentasi Tensorflow bisa mengunjungi website berikut <https://tensorflow.org/>.

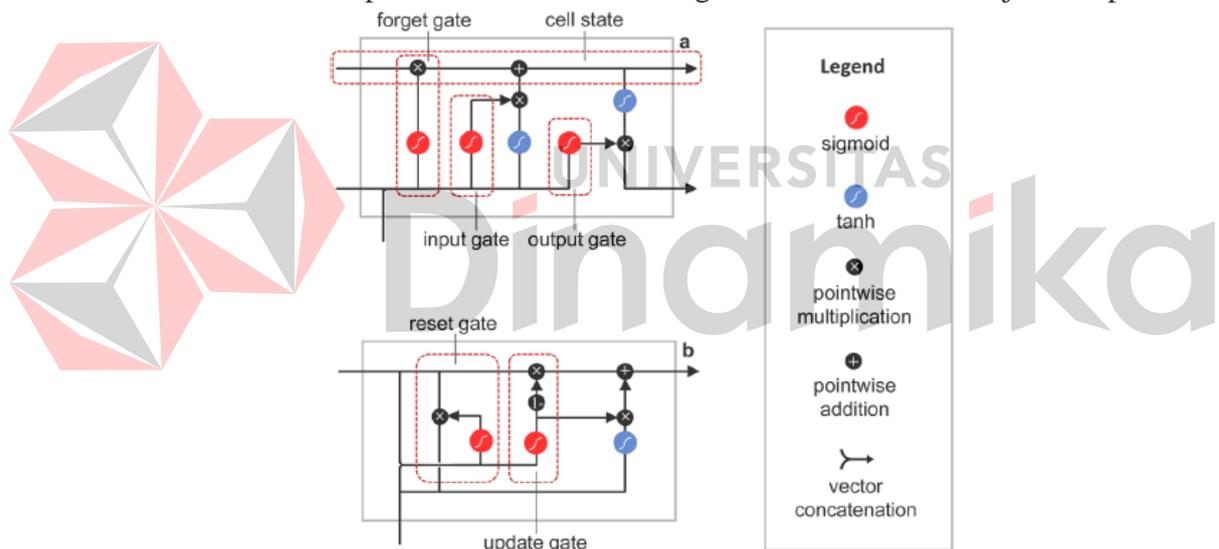
2.9 Google Colab

Pesatnya perkembangan bahasa pemrograman Python, telah membuat google untuk mencentuskan *online integrated environment development* (IDE), yang disebut Google Interactive Notebook, atau biasa dikenal dengan Google Colab lewat situs resminya www.colab.research.google.com. Dengan jenis lingkungan yang digunakan adalah Jupyter Notebook, dan ekstensi file adalah `ipynb`. Dari sisi *software*, Google Colab telah menyiapkan sebagian besar *library* yang dibutuhkan. Dalam penelitian ini *library* yang dibutuhkan adalah Keras, TensorFlow, NumPy, Pandas dan program pendukung lainnya, misalnya untuk membuat grafik melalui Matplotlib.

Dari segi perangkat keras, Google Colab memberikan fasilitas yang cukup mumpuni untuk membuat sebuah program yang dibuat menggunakan bahasa pemrograman python. Yaitu berupa media penyimpanan yang terintegrasi dengan Google Drive, serta menyediakan komputasi yang berspesifikasi tinggi. Selama fungsi server terjamin dan koneksi Internet lancar, sebagian besar pemrosesan tidak akan menimbulkan masalah bagi Google Colab. Dalam penelitian ini, fokusnya adalah pada GPU yang saat ini banyak digunakan sebagai dukungan DL. GPU yang melakukan pemrosesan paralel kompatibel dengan pemrosesan data DL dengan struktur *multi-core*.

2.10 Long Short-Term Memory Neural Network (LSTM)

Pada jaringan syaraf tiruan yang sedang dilatih, sering memiliki masalah pada hilangnya atau vanishing gradien pada Recurrent Neural Network saat memproses data sequential yang Panjang (Houdt, et al., 2020). Untuk menghindari kekurangan hal tersebut maka sebuah arsitektur bernama Long Short Term Memory (LSTM) diperkenalkan. Long Short – Term Memory Neural Network (LSTM) merupakan pengembangan dari Recurrent Neural Network (RNN), LSTM pertama kali diusulkan oleh Hochreiter dan Schmidhuber (Hochreiter & Schmidhuber, 1997) ditingkatkan oleh Graves (Graves, 2013). Arsitektur LSTM berisi sekelompok sel memori yang terhubung secara cyclicaly, dan setiap unit LSTM dilengkapi dengan input gate, forget gate, dan output gate. Gate berfungsi untuk mengontrol cara keadaan internal dipertahankan atau dibuang. Struktur unit ditunjukkan pada



Bertujuan untuk melakukan analogi bagaimana model LSTM bekerja, diasumsikan sebuah jaringan terdiri dari N blok pemrosesan dan M input. Umpan ke depan dalam pengulangan ini sistem saraf dijelaskan pada rumus dibawah ini.

Block Input. Langkah ini dikhususkan untuk memperbarui komponen *input gate*, yang dikombinasikan antara input $x^{(t)}$ saat ini dan output dari unit LSTM tersebut $y^{(t-1)}$ pada iterasi terakhir. Ini dapat dilakukan seperti yang digambarkan di bawah ini:

$$g^{(t)} = \sigma(b_g + U_g x^{(t)} + W_g h^{(t-1)}), \quad (1)$$

di mana U_g dan R_g masing-masing adalah bobot yang terkait dengan $x^{(t)}$ dan $h^{(t-1)}$, sedangkan b_g adalah singkatan dari bias *weight vector*.

Input Gate. Pada langkah ini, penulis memperbarui *input gate* yang menggabungkan arus

input $x^{(t)}$, output dari unit LSTM $h^{(t-1)}$ dan nilai sel $c^{(t-1)}$ dalam iterasi terakhir.

Persamaan berikut menunjukkan prosedur ini:

$$f^{(t)} = \sigma(b_f + U_f x^{(t)} + W_f h^{(t-1)} + p_f \odot c^{(t-1)}), \quad (2)$$

\odot dimana menunjukkan perkalian titik-bijaksana dari dua vektor, W_f , U_f dan p_f adalah bobot yang terkait dengan $x^{(t)}$, $h^{(t-1)}$ dan $c^{(t-1)}$, masing-masing, sedangkan b_o mewakili untuk

vektor bias yang terkait dengan komponen ini. Pada langkah sebelumnya, lapisan LSTM menentukan informasi mana yang harus dipertahankan dalam status sel jaringan $c^{(t)}$. Ini termasuk pemilihan nilai kandidat $g^{(t)}$ yang berpotensi ditambahkan ke status sel, dan nilai aktivasi $f^{(t)}$ dari *input gate*.

Pada langkah sebelumnya, lapisan LSTM menentukan informasi mana yang harus dipertahankan dalam status sel jaringan $c^{(t)}$. Ini termasuk pemilihan nilai kandidat $f^{(t)}$ yang berpotensi ditambahkan ke status sel, dan nilai aktivasi $f^{(t)}$ dari *input gate*.

Forget Gate. Pada langkah ini, unit LSTM menentukan informasi mana yang harus dihapus dari status sel sebelumnya $c^{(t-1)}$. Oleh karena itu, nilai aktivasi $o^{(t)}$ dari *forget gate* pada langkah waktu t dihitung berdasarkan input saat ini $h^{(t)}$, output $h^{(t-1)}$ dan status $c^{(t-1)}$ dari sel memori pada sebelumnya langkah waktu $^{(t-1)}$, koneksi lubang intip, dan suku bias b_f dari *forget gate*. Hal ini dapat dilakukan sebagai berikut:

$$o^{(t)} = \sigma(b_o + U_o x^{(t)} + W_o h^{(t-1)} + p_o \odot c^{(t-1)}), \quad (3)$$

di mana U_o , W_o dan p_o adalah bobot yang terkait dengan $x^{(t)}$, $h^{(t-1)}$ dan $c^{(t-1)}$, masing-masing, sedangkan b_o menunjukkan untuk vektor bobot bias.

Cell. Langkah ini menghitung nilai sel, yang menggabungkan input block $x^{(t)}$, nilai input gate $i^{(t)}$ dan *forgot gate* $o^{(t)}$, dengan nilai sel sebelumnya. Ini dapat dilakukan seperti yang digambarkan di bawah ini:

$$c^{(t)} = \sigma(b_o \odot g^{(t)} + c^{(t-1)} \odot o^{(t)}), \quad (4)$$

Output Gate. Langkah ini menghitung *output gate*, yang menggabungkan arus

input $x^{(t)}$, output dari unit LSTM $h^{(t-1)}$ dan nilai sel $c^{(t-1)}$ dalam iterasi terakhir. Ini dapat dilakukan seperti yang digambarkan di bawah ini:

$$s^{(t)} = \sigma(b_s + U_s x^{(t)} + W_s h^{(t-1)} + p_s \odot c^{(t-1)}), \quad (5)$$

dimana U_s , W_s dan p_s adalah bobot yang terkait dengan $x^{(t)}$, $h^{(t-1)}$ dan $c^{(t-1)}$, masing-masing, sedangkan b_s menunjukkan untuk vektor bobot bias.

Block Output. Akhirnya, menghitung output blok, yang menggabungkan nilai sel saat ini $c^{(t)}$ dengan nilai *block output* saat ini sebagai berikut:

$$h^{(t)} \sigma = \sigma(c^{(t)}) \odot s^{(t)}, \quad (5)$$

Dalam langkah di atas, σ , g dan h menunjukkan fungsi aktivasi non-linier.

Nilai sigmoid $\sigma(x) = \frac{1}{1+e^{1-x}}$ digunakan sebagai fungsi aktivasi gerbang, sedangkan tangen hiperbolik $g(x)=h(x) = \tanh(x)$ sering digunakan sebagai input blok dan fungsi aktivasi keluaran.

Dimana σ merepresentasikan sigmoid function dan bobot 0-1, dan $g^{(t)}$, $f^{(t)}$, $o^{(t)}$, $s^{(t)}$ merupakan input gate, forget gate, cell dan output gate state unit. Dan b , U , dan W merupakan bias, input weight, dan circular weight (Chung, et al., 2014).

Semua LSTM layer dikoneksikan dengan softmax function dengan rumus fungsi :

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}, \quad (6)$$

2.11 Pengujian Klasifikasi Sinyal

Akurasi, sensitivitas dan spesifikasi biasanya digunakan untuk mengukur kinerja dari sebuah metode klasifikasi pada medis. Langkah-langkah ini digunakan untuk menilai ketepatan usulan metode (Gharehbaghi, 2015). Maka dilakukan perhitungan rata-rata spesifisitas (S_p) ditambah sensitivitas (S_e) sebagai berikut:

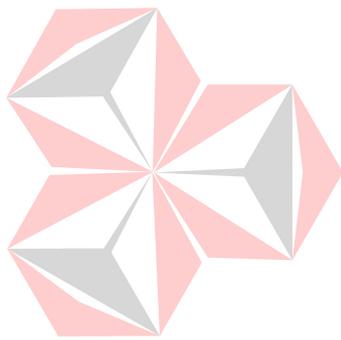
$$score = \frac{S_e + S_p}{2} \quad (7)$$

Dimana

$$S_e = \frac{A_n}{A_a + A_n} \quad (8)$$

$$S_p = \frac{N_n}{N_a + N_n} \quad (9)$$

Dengan nilai A_a , N_n merepresentasikan klasifikasi abnormal dan normal bernilai benar dan A_n , N_a merepresentasikan klasifikasi abnormal dan normal bernilai salah.

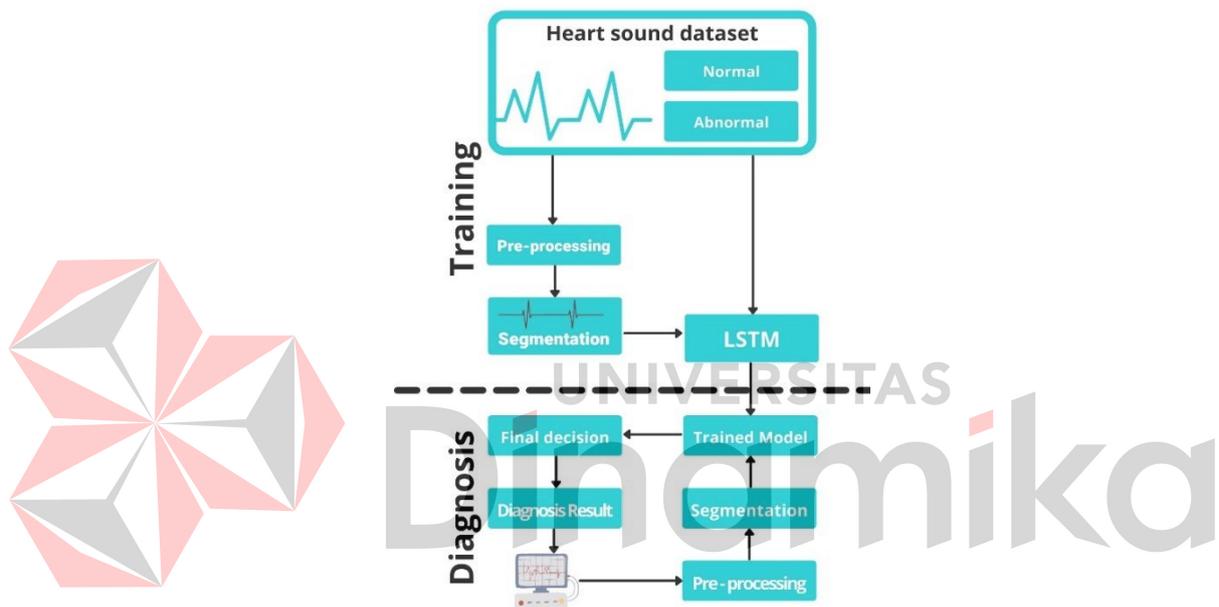


UNIVERSITAS
Dinamika

BAB III METODOLOGI PENELITIAN

3.1 Model Perancangan

Perancangan klasifikasi sinyal jantung dilakukan menggunakan metode LSTM seperti yang terdapat pada gambar 4. Terdapat beberapa tahapan dalam melakukan proses klasifikasi yaitu pre processing, segmentation, dan analisis data.



Gambar 3. 1 Perancangan kalsifikasi sinyal jantung PCG

3.1.1 Sinyal Normal dan Abnormal

Tahapan *pre processing* memiliki arti penting karena penggunaan stetoskop digital menyebabkan *noise* yang ikut tercampur saat proses perekaman sinyal jantung. Data yang digunakan pada proses klasifikasi memiliki beberapa klasifikasi. Data yang digunakan pada proses klasifikasi memiliki beberapa klasifikasi. Data berdasarkan suara sinyal jantung normal dan abnormal yang didapat pada database Physionet dengan topik *Classification of Normal/Abnormal Heart Sound Recordings: the PhysioNet/Computing in Cardiology Challenge* tahun 2016. Data berupa sinyal jantung normal dan abnormal yang diambil dari *physionet*. Data dibagi menjadi 3 bagian dengan rasio 60%, 30% dan 10% masing – masing untuk data train, validasi dan test. Data abnormal ditandai dengan adanya murmur.

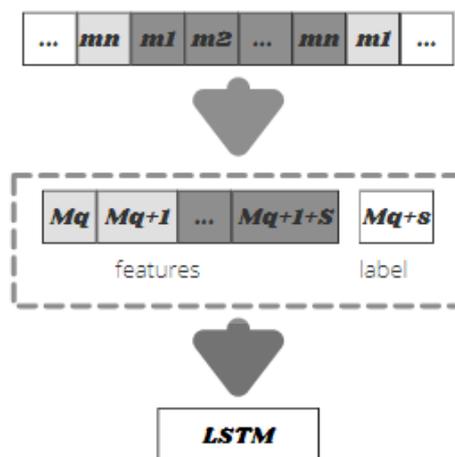
3.1.2 Pre-processing

Data sinyal jantung diubah dari wav menjadi *data time series* sebagai input. Sebelum dijadikan input *neural network*, dataset terlebih dahulu di *denoising* untuk menghilangkan *noise* pada sinyal. Dengan proses *denoising* sinyal jantung yang masih memiliki *noise* akan dibuang dan yang mengandung informasi penting akan dipertahankan. Selain itu jika diperlukan, juga diubah frekuensi samplingnya untuk mendapatkan hasil sinyal yang mendekati sempurna setelah di *training*. Input berupa nilai amplitudo terhadap waktu pada sinyal yang dikumpulkan menjadi satu dalam tabel time series, kemudian data – data tersebut disimpan dalam bentuk csv sebagai nilai inputan dari *neural network*.

3.1.3 Proses

Proses *training* pada jaringan LSTM diilustrasikan pada gambar 5, dilakukan dengan menjadikan tiap siklus sinyal menjadi sebuah model generatif. Setiap satu siklus sinyal PCG dilatih, dan tiap siklus menjadi fitur dalam *dataset* pelatihan. Satu siklus sinyal yang dilatihkan juga disertakan data *latih* label. Sinyal PCG satu siklus berpasangan dengan label membentuk sebuah data yang utuh yang kemudian menjadi data untuk di ingat saat proses *training* menggunakan LSTM.

Gambar 3. 2 Proses training sinyal jantung PCG



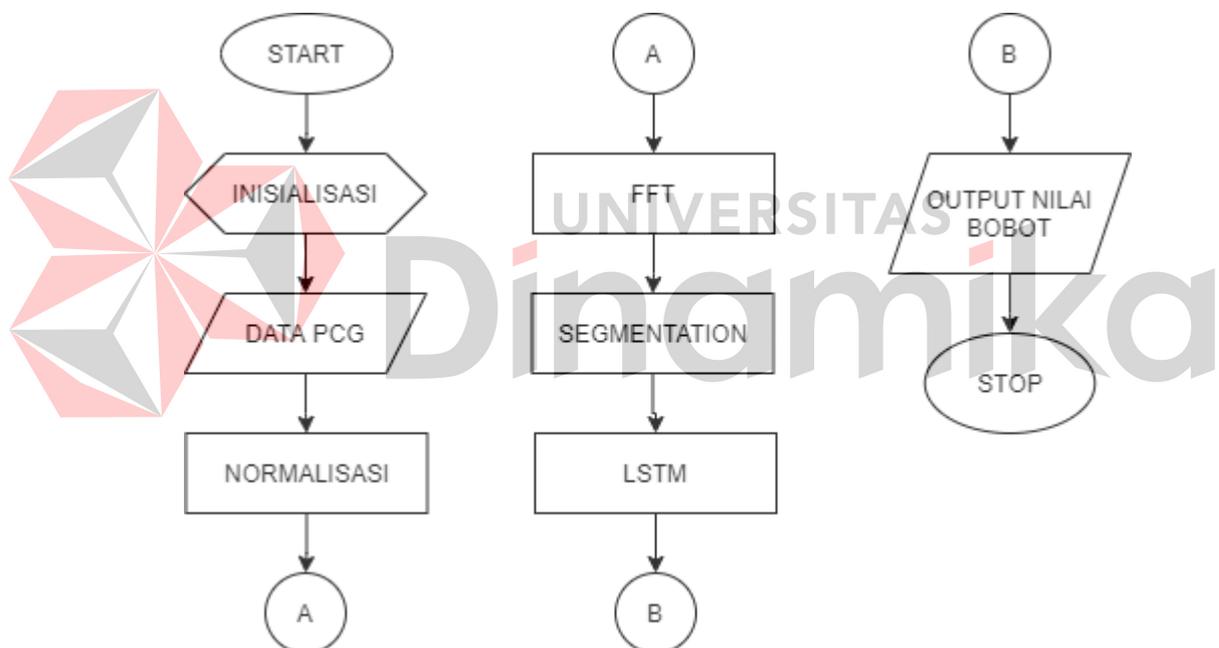
3.1.4 Analisis Data

Setelah dilakukan proses *training*, dilakukan proses test dengan menggunakan data test yang telah disediakan untuk melihat keberhasilan dari metode LSTM dalam melakukan klasifikasi. Parameter uji dinilai dari tingkat

keberhasilan metode LSTM dalam melakukan klasifikasi data. Tingkat keberhasilan dari sebuah klasifikasi pada bidang medis dilihat dari seberapa besar tingkat akurasi, sensitifitas dan juga spesififikasi dari hasil klasifikasi. Pengukurannya dilihat dari melakukan rata-rata terhadap sensitifitas dan spesififikasi. Hasil akurasi yang diinginkan adalah diatas 80% karena dengan nilai tersebut proses klasifikasi dianggap berhasil.

3.2 Model Program

Tahapan alur program untuk *training* data dapat dilihat pada gambar 3.3 di bawah. Dimana proses *learning* akan dijelaskan sebagai berikut:



Gambar 3. 3 Blok diagram program pelatihan dengan LSTM

1. Inisialisasi

Pada tahap ini program pertama kali menjalankan pustaka yang dibutuhkan untuk menjalankan program secara keseluruhan. Pustaka yang paling utama yaitu keras, dimana keras merupakan pustaka pada python yang banyak digunakan untuk menjalankan program *deep learning*. Kemudian juga ditambahkan pustaka numpy

untuk proses aritmatika, dan pustaka sklearn untuk penghitungan nilai konvolusi matrix.

Terdapat beberapa pustaka yang digunakan untuk menjalankan program secara keseluruhan. Untuk pustaka `from scipy.io import wavfile` digunakan sebagai fungsi pemnacaan data yang berupa audio wav. Untuk proses pelatihan menggunakan pustaka keras. Sedangkan untuk operasi aritmatika menggunakan pustaka numpy.

2. Data PCG

Data seperti yang sudah dijelaskan pada subbab model perancangan dimana data didapat dari *Classification of Normal/Abnormal Heart Sound Recordings: the PhysioNet/Computing in Cardiology Challenge* tahun 2016. Data merupakan kumpulan suara sinyal jantung berupa wav. Berikut salah satu contoh sinyal wav yang sudah di plot.



Gambar 3. 4 Plotting sinyal suara jantung PCG wav original

3. Normalisasi

Normalisasi merupakan suatu cara yang digunakan untuk memaksimalkan nilai aplitudo dan frekuensi. Proses normalisasi diperoleh dengan membagi nilai tiap suara sinyal jantung pcg dengan nilai sampling yaitu dua dipangkat empat belas. Perhitungan nilai normalisasi dapat diperoleh dari hasil persamaan rumus berikut:

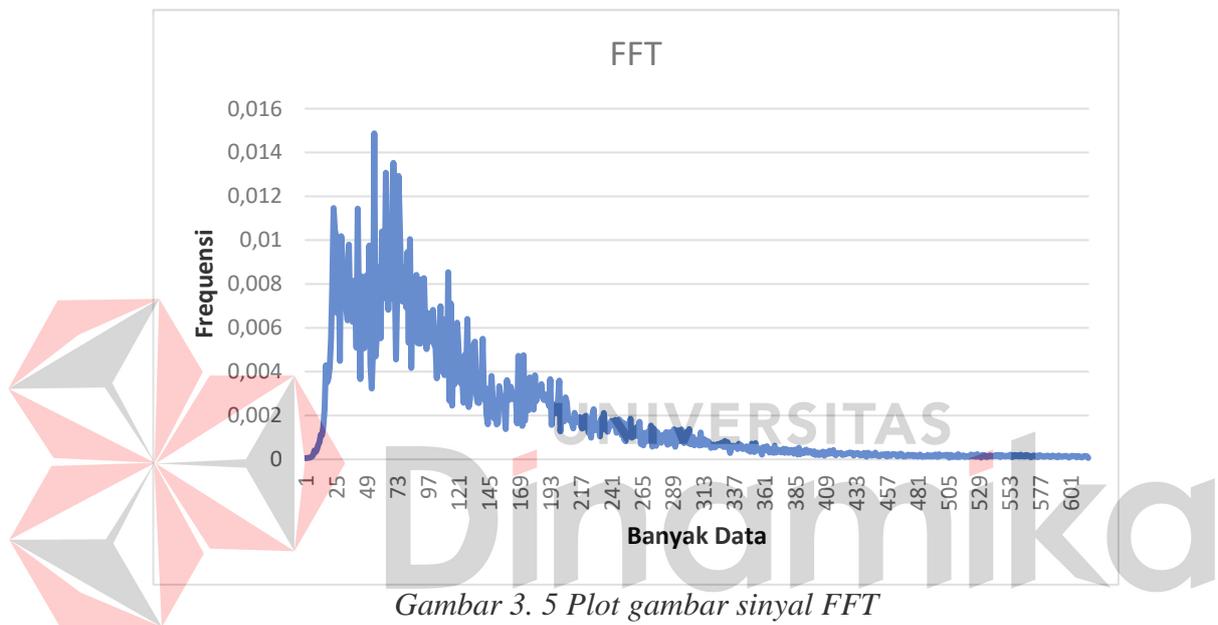
$$x_{norm} = \frac{x_{in}}{x_{max}}$$

Dimana x_{norm} adalah nilai sinyal suara jantung pcg yang telah dinormalisasi dan x_{in} adalah nilai masukan dari sinyal suara jantung pcg asli, x_{max} adalah nilai maksimum dari nilai sinyal suara jantung pcg. Semua nilai dari sinyal suara jantung pcg dibagi dengan nilai tertinggi dari sinyal suara jantung pcg untuk menghasilkan

nilai maksimum sama dengan satu. Tujuan normalisasi adalah untuk mendapatkan nilai maksimum dari amplitudo sinyal suara jantung pcg.

4. Fast Fourier Transform (FFT)

Nilai FFT yang diambil merupakan nilai keluaran kompleks yang dijadikan nilai absolut yang kemudian disebut dengan magnitude. Untuk plotting sinyal jantung yang telah di FFT ditunjukkan sebagai gambar berikut.



Gambar 3. 5 Plot gambar sinyal FFT

5. Segmentasi

Segmentasi dilakukan dengan membagi sinyal menjadi beberapa siklus dimana tiap siklus dibatasi menjadi 16000 data. Dengan adanya segmentasi pula sinyal yang didapat merupakan sinyal yang paling dibutuhkan dan membuang sinyal yang tidak perlu sehingga mempercepat proses *learning*.

6. Modeling

Yaitu proses menyusun desain sel jaringan yang digunakan untuk nantinya dijadikan acuan model pada *learning*. Pada perancangan model memiliki struktur model yang disusun untuk memenuhi nilai bobot keluaran yang diinginkan. Untuk mengetahui keberhasilan dari model yang dibuat dapat dilihat dari nilai akurasi dan *loss*. Nilai akurasi merupakan nilai yang digunakan untuk menentukan tingkat

keberhasilan model yang dibuat. Nilai *loss* adalah ukuran kesalahan jaringan, dan tujuannya adalah untuk meminimalkannya.

7. Nilai Bobot

Merupakan nilai acuan yang digunakan untuk mengetahui apakah memenuhi atau tidak jika nantinya data training dijadikan model untuk data prediksi. Parameter uji dinilai dari tingkat keberhasilan metode LSTM dalam melakukan klasifikasi data. Tingkat keberhasilan dari sebuah klasifikasi pada bidang medis dilihat dari seberapa besar tingkat akurasi, sensitifitas dan juga spesififikasi dari hasil klasifikasi. Pengukurannya dilihat dari melakukan rata-rata terhadap sensitifitas dan spesififikasi. Hasil akurasi yang diinginkan adalah diatas 80% karena dengan nilai tersebut proses klasifikasi dianggap berhasil. Matrik konfusi adalah bentuk matrik yang menunjukkan tingkat keberhasilan sebuah model pelatihan sinyal suara jantung ppg dalam mengklasifikasi berdasarkan dataset. Perhitungan *confusion matrix* didapat dari rumus sebagai berikut:

Tabel 3. 1 *Confusion matrix* hasil klasifikasi

1 (Positif) 0 (Negatif)

Prediksi	1 (Positif)	TP (True Positif)	FN (False Negatif)
	0 (Negatif)	FN (False Negatif)	TN (True Negatif)

Untuk merepresentasikan hasil proses klasifikasi pada *confusion matrix* disimbolkan dengan 4 istilah pada table 3.1.

1. True Positif – Untuk data positif yang diprediksi benar.
2. True Negatif – Untuk data negatif yang diprediksi benar.
3. False Positif – Untuk data negatif namun diprediksi positif.
4. False Negatif – Untuk data positif namun diprediksi negatif.

Dari hasil *confusion matrix* kemudian diproses untuk mendapat nilai *performance matrix* yang digunakan sebagai parameter kinerja model yang telah dibuat. *Performance matrix* terdiri dari *accuracy*, *precision*, *recall*, dan *F1 score* yang di implementasikan menggunakan rumus berikut:

1. Accuracy

Digunakan untuk mengetahui keakuratan model dalam mengklasifikasi dengan benar. Akurasi dapat digunakan untuk mengetahui rasio model dalam memprediksi Nilai akurasi dapat di ketahui melalui persamaan berikut:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2. Precision (Positive Predictive Value)

Precision menggambarkan tingkat akurasi antara data yang diminta dan hasil prediksi yang diberikan oleh model.

$$precision = \frac{TP}{TP+FP} \quad (2)$$

3. Recall atau Sensitivity (True Positive Rate)

Recall menggambarkan keberhasilan model dalam mengambil informasi.

$$precision = \frac{TP}{TP+FN} \quad (3)$$

4. Rumus F1 – Score

$$F1 - Score = 2x \frac{Recall*Precesion}{Recall+Precesion} \quad (4)$$

BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil dari beberapa sample penelitian tugas akhir ini. Pengujian dilakukan pada meliputi pengujian pada perangkat lunak.

4.1 Pengujian Segmentasi

Pengujian segmentasi dilakukan untuk penyesuaian data input sebagai data latih sesuai dengan yang dibutuhkan.

4.1.1 Tujuan Pengujian Segmentasi

Tujuan dari pengujian adalah memotong data sinyal suara jantung pcg sesuai dengan yang dibutuhkan dalam penelitian tugas akhir ini. Sinyal suara jantung telah dipotong sesuai dengan yang dibutuhkan untuk menjadi dataset latih. Pada penelitian tugas akhir ini sinyal suara jantung yang awalnya terdiri dari antara 60000 hingga 70000 data di potong menjadi 16000 data. Berikut merupakan potongan program dari segmentasi.

```
def segmentasi(sndList):  
    segList = []  
    for snd in sndList:  
        sndNP = np.asarray(snd)  
        if sndNP.size <= 16000:  
            sndSeg = sndNP  
        elif sndNP.argmax() > 8000 and (sndNP.size - sndNP.argmax()) >= 8000:  
            sndSeg = snd[sndNP.argmax() - 8000:sndNP.argmax()+8000]  
        elif sndNP.argmax() < 8000 and sndNP.size - sndNP.argmax() > 8000:  
            sndSeg = sndNP[0:16000]  
        elif sndNP.argmax() > 8000 and sndNP.size - sndNP.argmax() <= 8000:  
            sndSeg = sndNP[sndNP.size - 16000:sndNP.size]  
        segList.append(sndSeg)  
  
    return segList
```

Gambar 4. 1 Program segmentasi

4.1.2 Alat Yang Digunakan Untuk Pengujian Segmentasi

1. Laptop
2. Python
3. Google Colab

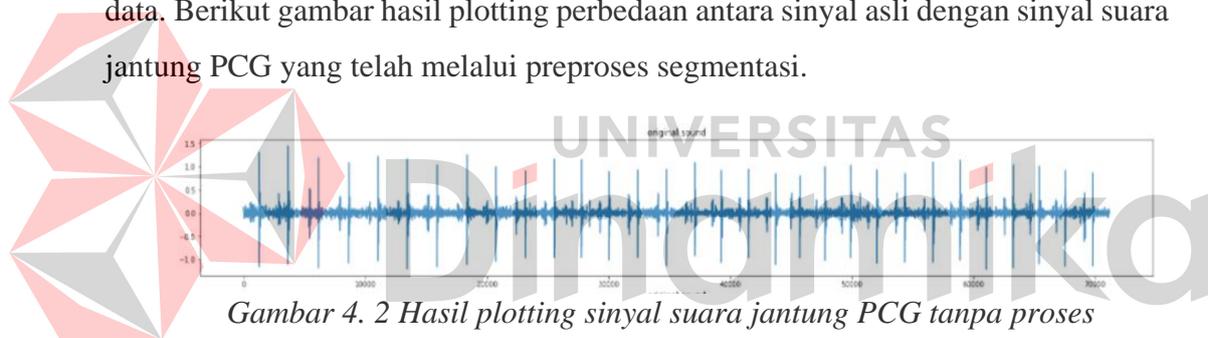
4.1.3 Prosedur Pengujian Program Segmentasi

Agar program dapat dijalankan maka perlu prosedur pengujian sebagai berikut:

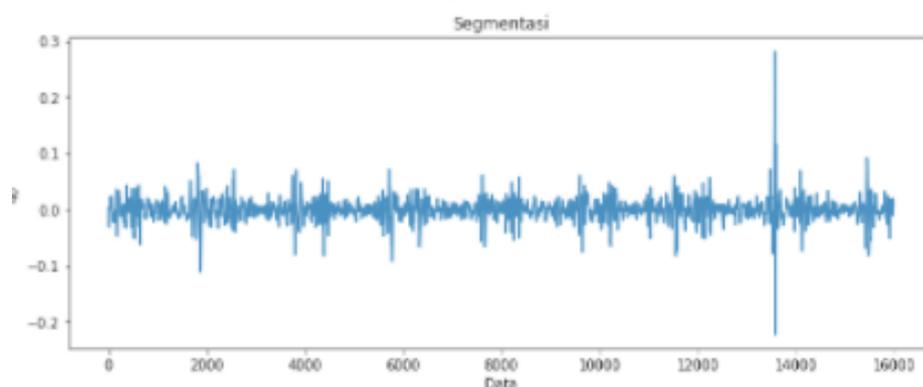
1. Membuka Google Colab pada web browser.
2. Membuka program segmentasi.
3. Mengatur variable nilai banyaknya data yang akan di segmentasi.
4. Jalankan program segmentasi.

4.1.4 Hasil Pengujian Program Segmentasi

Sinyal suara jantung pcg yang didapat pada physionet memiliki data sebanyak 60000 hingga 70000 data. Kemudian data tersebut di segmentasi menjadi 16000 untuk menyesuaikan dengan data latih dan mempercepat proses pada saat training data. Berikut gambar hasil plotting perbedaan antara sinyal asli dengan sinyal suara jantung PCG yang telah melalui preproses segmentasi.



Gambar 4. 2 Hasil plotting sinyal suara jantung PCG tanpa proses segmentasi



Gambar 4. 3 Hasil plotting segmentasi sinyal suara jantung PCG

4.2 Pengujian Fast Fourier Transform (FFT)

Pengujian FFT dilakukan untuk penyesuaian data input LSTM

4.2.1 Tujuan Pengujian FFT

Tujuan dari menjalankan program ini adalah untuk mengetahui kesesuaian nilai yang akan dijadikan data latih. Karena data latih yang diperlukan LSTM dalam proses pelatihan berupa dataset yang berbentuk time series, maka dengan FFT akan tepat jika digunakan untuk mengubah frekuensi domain terhadap waktu. Untuk nilai dari FFT yang diambil berupa nilai absolut yang sebelumnya keluaran FFT sendiri berbentuk bilangan complex yang terdiri dari bilangan riil dan bilangan imajiner. Dengan nilai absolut maka akan memudahkan pada saat proses pelatihan karena data nilai yang telah disesuaikan. Berikut potongan dari program FFT.



```
[ ] def fftProcess4(sndList, cutOffIdx, offset):
    sndFFTLList = []
    fList = []
    NFFT = 16384
    Fs = 2000 # Fs = Frekuensi Sampling
    for idx, snd in enumerate(sndList):
        L = len(snd) # L = Panjang Sinyal Suara jantung PCG yang telah disegmentasi
        Ypre = fft(snd, NFFT)/L # Komputasi FFT dan Normalisasi
        Y = 2*np.abs(Ypre[0:NFFT//2]) # harga absolut dari fft
        sndFFTLList.append(average(Y[0:cutOffIdx], 4)) # Mengambil rata rata dari data di titik ke 4
        f = Fs/2*np.linspace(0.0, 1, NFFT//2+1)
        fList.append(f)
    return sndFFTLList, fList
```

Gambar 4. 4 Program FFT (Fast Fourier Transform)

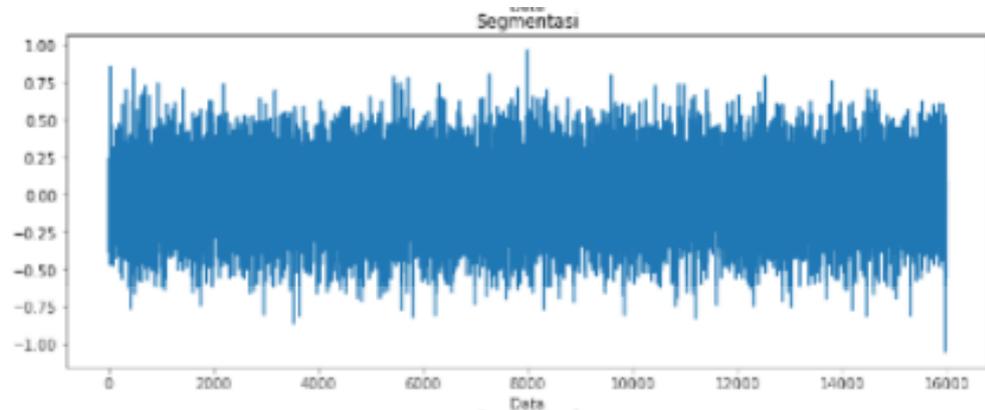
4.2.2 Alat Yang Digunakan Untuk Pengujian Segmentasi

1. Laptop
2. Python
3. Google Colab

4.2.3 Hasil Pengujian FFT

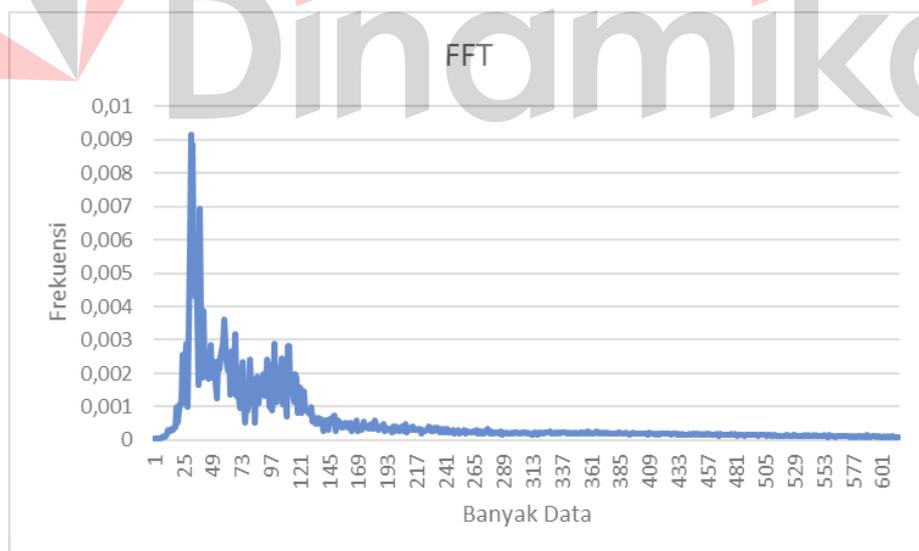
Hasil dari sinyal suara jantung yang awalnya memiliki data sebanyak antara 60000 hingga 70000 data kemudian di segmentasi hingga menyesuaikan data yang digunakan sebanyak 16000 data. Dari data hasil segmentasi kemudian dijadikan bahan untuk input yang selanjutnya diproses dalam program FFT, proses FFT ini dilakukan untuk mentransformasi dari sinyal suara jantung yang meliki domain terhadap frekuensi menjadi bentuk data sinyal domain terhadap waktu. Dan berikut

merupakan hasil dari plotting sinyal yang di segmentasi dan sinyal yang yang melalui preproses FFT.



Gambar 4. 5 Hasil plotting sinyal suara jantung PCG segmentasi

Dari gambar diatas dapat kita lihat Sinyal dimulai dari data sebanyak 16000 diakhir index, dengan normalisasi data tersebut kemudian memiliki amplitudo yang sama yaitu batas atas adalah (1) dan batas bawah adalah (-1).



Gambar 4. 6 Hasil plotting sinyal suara jantung PCG FFT

4.3 Pengujian Pelatihan Data Sinyal Suara Jantung PCG Dengan LSTM

Pada pengujian proses pelatihan dilakukan dengan menggunakan program yang telah dirancang dan dijalankan di google colab.

4.3.1 Tujuan Proses Pelatihan

Tujuan dari pengujian ini dilakukan untuk mengetahui proses pelatihan dapat digunakan dan berfungsi dengan baik, serta untuk mendapatkan nilai bobot sesuai dengan yang dibutuhkan dalam penelitian ini.

4.3.2 Alat yang dibutuhkan dalam proses pengujian pelatihan.

Agar proses pengujian pelatihan ini dapat berjalan sesuai dengan penelitian ini maka diperlukan beberapa alat sebagai berikut :

1. Komputer/ Laptop.
2. Python.
3. Google Colab.

4.3.3 Prosedur pengujian proses pelatihan

Prosedur pengujian:

1. Buka web browser di komputer / laptop dan kemudian buka google colab.
2. Jalankan program LSTM di google colab.
3. Pilih nilai input dan nilai hidden layer pada model, pelatihan dilakukan sebanyak 4 kali dengan variasi data sebagai berikut:
 - a. Input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 512, dan hidden layer kelima 512.
 - b. Input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 1024, dan hidden layer kelima 512.
 - c. Input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 1024, dan hidden layer kelima 1024.

- d. Input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 2048, dan hidden layer kelima 1024.
4. Masukkan data keluaran yang diinginkan dari program pelatihan berupa nilai epochs sebanyak 500 dan nilai batch sebanyak 500.

Tabel 4. 1 Data input pelatihan

Penelitian Ke-	Input	Hidden 1	Hidden 2	Hidden 3	Hidden 4	Hidden 5
1	615	1024	2048	2048	512	512
2	615	1024	2048	2048	1024	512
3	615	1024	2048	2048	1024	1024
4	615	1024	2048	2048	2048	1024

4.3.4 Hasil pengujian proses pelatihan

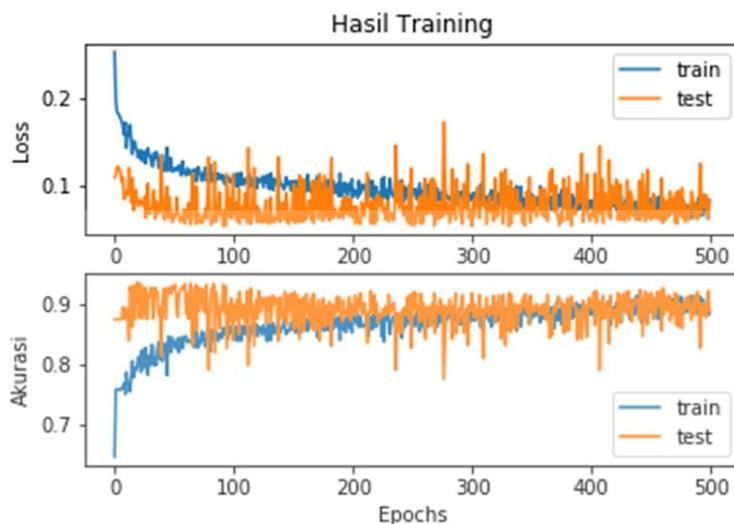
1. Hasil Pengujian dengan nilai epochs 500, batch 500, input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 512, dan hidden layer kelima 512. Dan berikut merupakan tabel hasil training yang diambil berdasarkan model yang telah ditentukan.

Tabel 4. 2 Hasil training pada percobaan pertama

No.	Epochs	Accuracy	Loss	Validasi Accuracy	Validasi Loss
1	50	0.8391	0.1160	0.9321	0.0604
2	100	0.8532	0.1112	0.8416	0.1125
3	150	0.8593	0.1051	0.8796	0.0753
4	200	0.8541	0.1078	0.8930	0.0645
5	250	0.8774	0.0869	0.8292	0.1200
6	300	0.8884	0.0833	0.8930	0.0701
7	350	0.8818	0.0838	0.9156	0.0582
8	400	0.8955	0.0746	0.9053	0.0602
9	450	0.8977	0.0744	0.8940	0.0779
10	500	0.8849	0.0841	0.8909	0.0831

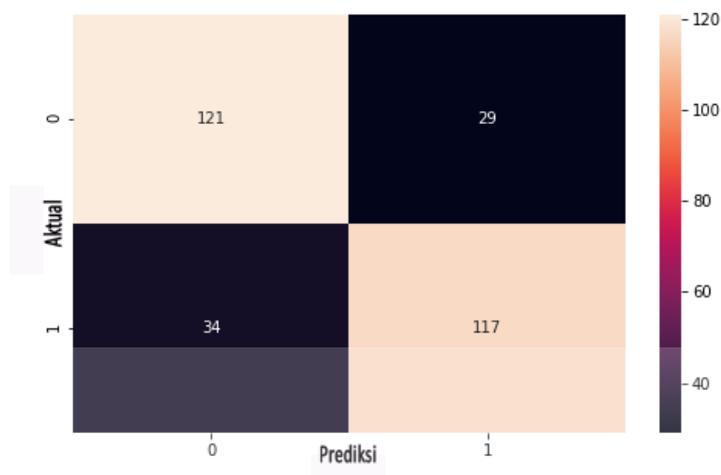
Pada tabel 4.2 menunjukkan hasil training dari 500 epoch dimana nilai akhir dari proses tersebut berupa nilai akurasi train 88% dan akurasi test 73%. Apabila

proses training diatas diplot dalam sebuah grafik maka akan tampil seperti yang ditunjukkan grafik dibawah.



Gambar 4. 7 Hasil grafik percobaan pertama

Dari grafik pada gambar 4.7 didapat hasil fungsi akurasi dan loss yang masih fluktuatif karena nilai akurasi training lebih tinggi daripada nilai akurasi dengan data test. Kemudian hasil akurasi test yang telah diperoleh diolah kembali untuk mendapatkan hasil *confusion matrix*. Dengan nilai test yang didapat maka akan diperoleh nilai true positif (TP) sebanyak 117, true negatif (TN) sebanyak 121, false positif (FP) sebanyak 29, dan false negatif (FN) sebanyak 34. Jika divisualiasasi maka diperoleh gambar 4.8.

Gambar 4. 8 *Confusion Matrix* percobaan pertama

Dari hasil *confusion matrix* yang telah didapat maka akan didapatkan nilai *performance matrix* yang akan dijadikan acuan dalam mengukur kinerja pada model yang telah dibuat. Dari Dari model yang telah dibuat berdasarkan percobaan pertama maka akan menghasilkan *performance matrix* sebagaimana yang telah tersaji pada tabel 4.3 berikut.

Tabel 4. 3 *Performa matrik percobaan pertama*

	Precision	Recall	F1-Score	Support
Normal	0.78	0.81	0.79	150
Abnormal	0.80	0.77	0.79	151
Accuracy			0.79	301
Macro Avg	0.79	0.79	0.79	301
Weight Avg	0.79	0.79	0.79	301

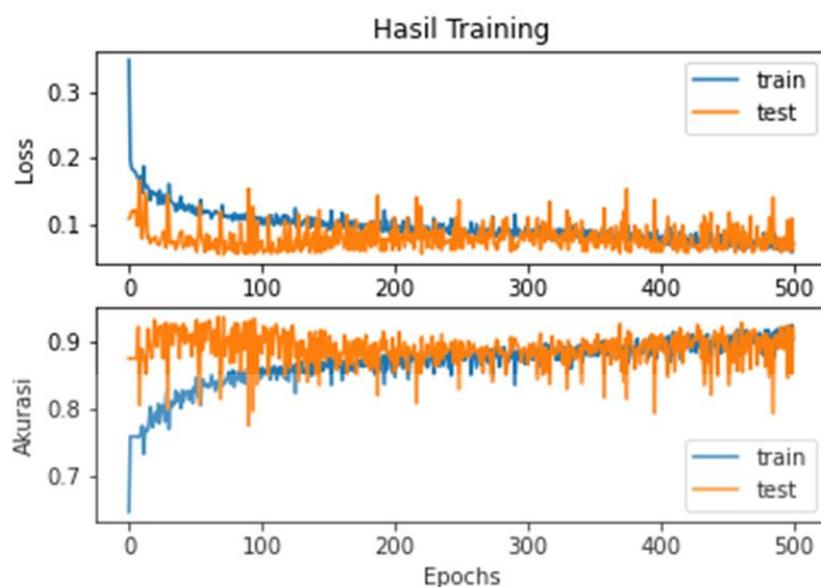
Dari tabel 4.3 sesuai dengan rumus yang ditetapkan sebelumnya maka didapatkan hasil berupa sensitivitas 79% dan spesifitas 79%. Sesuai dengan rumus yang dijelaskan pada bab 2, kita dapat mengetahui nilai akurasi yang kemudian disebut dengan *score* dengan rumus dimana nilai sensitivitas dijumlahkan dengan nilai spesifitas dibagi dua. Maka dengan rumus yang telah ditetapkan pada percobaan pertama didapat nilai akhir atau *score* sebesar 79%.

2. Hasil Pengujian dengan nilai epochs 500, batch 500, input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 1024, dan hidden layer kelima 512. Dan berikut merupakan tabel hasil training yang diambil berdasarkan model yang telah ditentukan.

Tabel 4. 4 Hasil training pada percobaan kedua

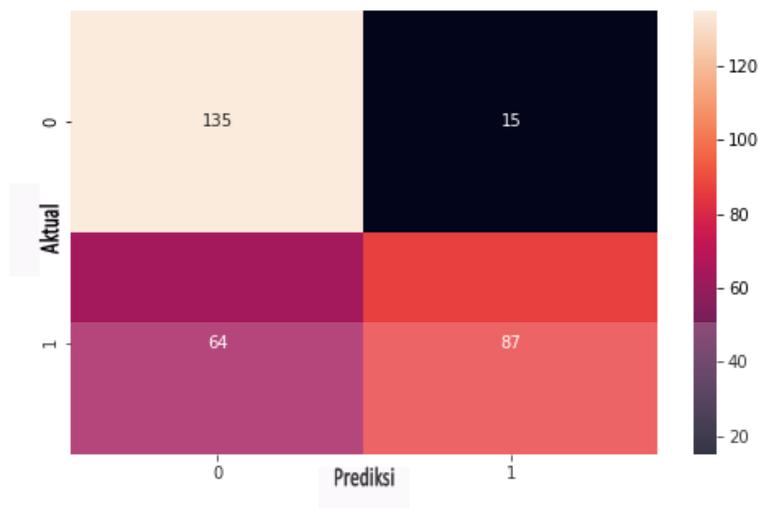
No.	Epochs	Accuracy	Loss	Validasi Accuracy	Validasi Loss
1	50	0.8285	0.1215	0.8642	0.1128
2	100	0.8479	0.1114	0.0589	0.9239
3	150	0.8558	0.1059	0.9136	0.0603
4	200	0.8743	0.0900	0.8724	0.0910
5	250	0.8660	0.0998	0.9198	0.0591
6	300	0.8898	0.0807	0.8899	0.0648
7	350	0.8907	0.0812	0.8529	0.1065
8	400	0.8920	0.0787	0.8920	0.0764
9	450	0.9039	0.0729	0.8827	0.0887
10	500	0.8973	0.0768	0.9280	0.0656

Pada tabel 4.4 menunjukkan hasil training dari 500 epoch dimana nilai akhir dari proses tersebut berupa nilai akurasi train 0.884 dan akurasi test 0.724. Apabila proses training diatas diplot dalam sebuah grafik maka akan tampil seperti yang ditunjukkan grafik dibawah.



Gambar 4. 9 Hasil grafik percobaan kedua

Dari grafik pada gambar 4.9 didapat hasil fungsi akurasi dan loss yang masih fluktuatif karena nilai akurasi training lebih tinggi daripada nilai akurasi dengan data test. Kemudian hasil akurasi test yang telah diperoleh diolah kembali untuk mendapatkan hasil *confusion matrix*. Dengan nilai test yang didapat maka akan diperoleh nilai true positif (TP) sebanyak 87, true negatif (TN) sebanyak 135, false positif (FP) sebanyak 15, dan false negatif (FN) sebanyak 64. Jika divisualiasasi maka diperoleh gambar 4.10.



Gambar 4. 10 *Confusion Matrix* percobaan kedua

Dari hasil *confusion matrix* yang telah didapat maka akan didapatkan nilai *performance matrix* yang akan dijadikan acuan dalam mengukur kinerja pada model yang telah dibuat. Dari model yang telah dibuat berdasarkan percobaan kedua maka akan menghasilkan *performance matrix* sebagaimana yang telah tersaji pada tabel 4.5 berikut.

Tabel 4. 5 *Perfoma matrik percobaan kedua*

	Precision	Recall	F1-Score	Support
Normal	0.66	0.93	0.77	150
Abnormal	0.88	0.52	0.66	151
Accuracy			0.72	301
Macro Avg	0.77	0.72	0.71	301
Weight Avg	0.77	0.72	0.71	301

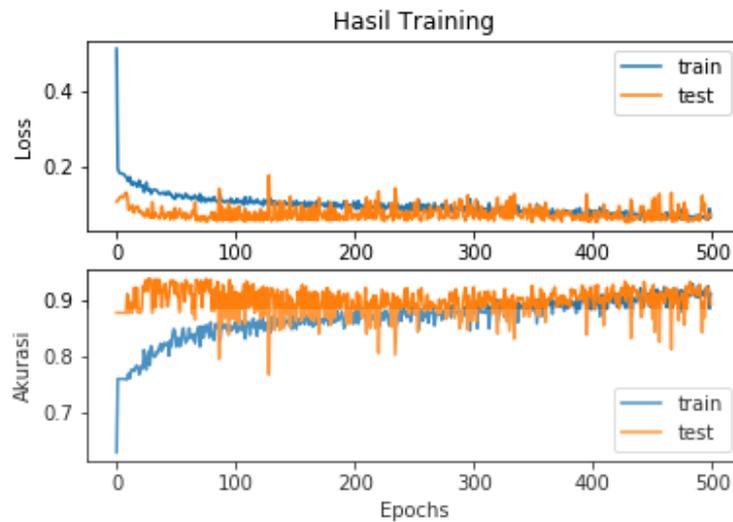
Dari tabel 4.5 sesuai dengan rumus yang ditetapkan sebelumnya maka didapatkan hasil berupa sensitivitas 72% dan spesifitas 77%. Sesuai dengan rumus yang dijelaskan pada bab 2, kita dapat mengetahui nilai akurasi yang kemudian disebut dengan *score* dengan rumus dimana nilai sensitivitas dijumlahkan dengan nilai spesifitas dibagi dua. Maka dengan rumus yang telah ditetapkan pada percobaan kedua didapat nilai akurasi atau *score* sebesar 74,5%.

3. Hasil Pengujian dengan nilai epochs 500, batch 500, input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 1024, dan hidden layer kelima 1024. Dan berikut merupakan tabel hasil training yang diambil berdasarkan model yang telah ditentukan.

Tabel 4. 6 Hasil training pada percobaan ketiga

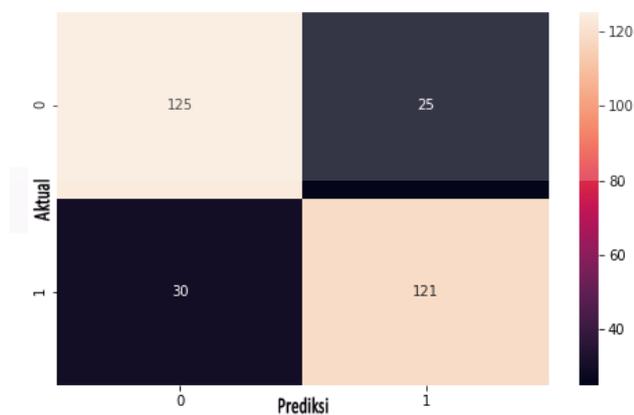
No.	Epochs	Accuracy	Loss	Validasi Accuracy	Validasi Loss
1	50	0.8435	0.1147	0.9105	0.0915
2	100	0.8576	0.1037	0.9136	0.0590
3	150	0.8726	0.0976	0.8817	0.8726
4	200	0.8571	0.1081	0.8827	0.0717
5	250	0.8801	0.0887	0.9105	0.0661
6	300	0.8735	0.0931	0.8858	0.0757
7	350	0.8893	0.0824	0.9239	0.0537
8	400	0.8818	0.0846	0.9074	0.0651
9	450	0.9127	0.0676	0.9177	0.0563
10	500	0.9101	0.0674	0.8909	0.0763

Pada tabel 4.6 menunjukkan hasil training dari 500 epoch dimana nilai akhir dari proses tersebut berupa nilai akurasi train 0.913 dan akurasi test 0.817. Apabila proses training diatas diplot dalam sebuah grafik maka akan tampil seperti yang ditunjukkan grafik dibawah.



Gambar 4. 11 Hasil grafik percobaan ketiga

Dari grafik pada gambar 4.11 didapat hasil fungsi akurasi dan loss yang masih fluktuatif karena nilai akurasi training lebih tinggi daripada nilai akurasi dengan data test. Kemudian hasil akurasi test yang telah diperoleh diolah kembali untuk mendapatkan hasil *confusion matrix*. Dengan nilai test yang didapat maka akan diperoleh nilai true positif (TP) sebanyak 121, true negatif (TN) sebanyak 125, false positif (FP) sebanyak 25, dan false negatif (FN) sebanyak 30. Jika divisualiasasi maka diperoleh gambar 4.12.



Gambar 4. 12 Konfusi matrix percobaan ketiga

Dari hasil confusion matrix yang telah didapat maka akan didapatkan nilai *performance matrix* yang akan dijadikan acuan dalam mengukur kinerja pada model yang telah dibuat. Dari Dari model yang telah dibuat berdasarkan percobaan ketiga maka akan menghasilkan *performance matrix* sebagaimana yang telah tersaji pada tabel 4.7 berikut.

Tabel 4. 7 Performa matrik percobaan ketiga

	Precision	Recall	F1-Score	Support
Normal	0.81	0.83	0.82	150
Abnormal	0.83	0.80	0.81	151
Accuracy			0.82	301
Macro Avg	0.82	0.82	0.82	301
Weight Avg	0.82	0.82	0.82	301

Dari tabel 4.7 sesuai dengan rumus yang ditetapkan sebelumnya maka didapatkan hasil berupa sensitivitas 82% dan spesifitas 82%. Sesuai dengan rumus yang dijelaskan pada bab 2, kita dapat mengetahui nilai akurasi yang kemudian disebut dengan *score* dengan rumus dimana nilai sensitivitas dijumlahkan dengan nilai spesifitas dibagi dua. Maka dengan rumus yang telah ditetapkan pada percobaan ketiga didapat nilai akurasi atau *score* sebesar 82%.

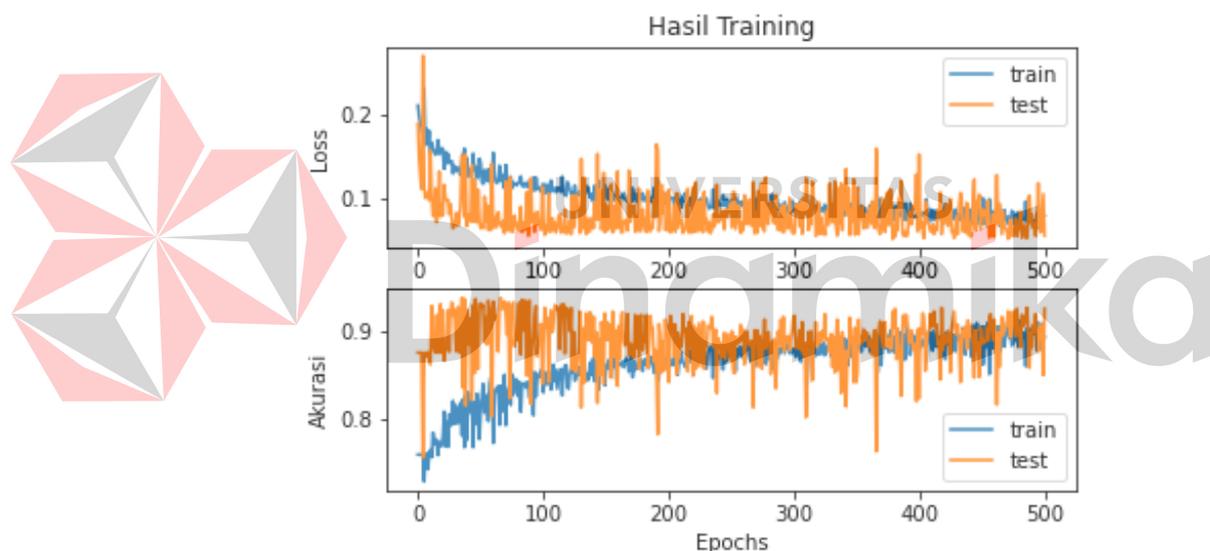
4. Hasil Pengujian dengan nilai epochs 500, batch 500, input layer sebanyak 615, hidden layer pertama 1024, hidden layer kedua 2048, hidden layer ketiga 2048, hidden layer keempat 2048, dan hidden layer kelima 1024. Dan berikut merupakan tabel hasil training yang diambil berdasarkan model yang telah ditentukan.

Tabel 4. 8 Hasil training pada percobaan keempat

No.	Epochs	Accuracy	Loss	Validasi Accuracy	Validasi Loss
1	50	0.7773	0.1482	0.0725	0.9321
2	100	0.8466	0.1165	0.8405	0.1145
3	150	0.8391	0.1160	0.0595	0.9218
4	200	0.0928	0.8726	0.9074	0.0613
5	250	0.8814	0.0880	0.8683	0.0934
6	300	0.8805	0.0852	0.9167	0.0576

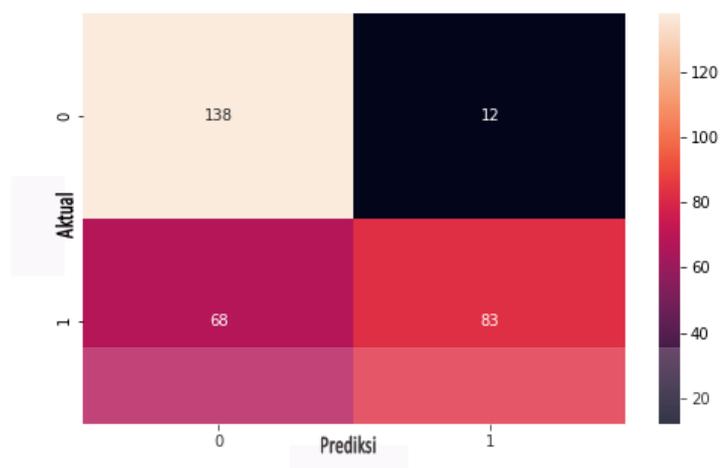
No.	Epochs	Accuracy	Loss	Validasi Accuracy	Validasi Loss
7	350	0.8871	0.0803	0.8426	0.1074
8	400	0.8805	0.0802	0.8241	0.1534
9	450	0.9065	0.0697	0.8940	0.0797
10	500	0.8951	0.0795	0.9270	0.0551

Pada tabel 4.8 menunjukkan hasil training dari 500 epoch dimana nilai akhir dari proses tersebut berupa nilai akurasi train 0.900 dan akurasi test 0.734. Apabila proses training diatas diplot dalam sebuah grafik maka akan tampil seperti yang ditunjukkan grafik dibawah.



Gambar 4. 13 Hasil grafik percobaan keempat

Dari grafik pada gambar 4.13 didapat hasil fungsi akurasi dan loss yang masih fluktuatif karena nilai akurasi training lebih tinggi daripada nilai akurasi dengan data test. Kemudian hasil akurasi test yang telah diperoleh diolah kembali untuk mendapatkan hasil *confusion matrix*. Dengan nilai test yang didapat maka akan diperoleh nilai true positif (TP) sebanyak 83, true negatif (TN) sebanyak 138, false positif (FP) sebanyak 12, dan false negatif (FN) sebanyak 68. Jika divisualiasasi maka diperoleh gambar 4.14.



Gambar 4. 14 Konfusi matrix percobaan keempat

Dari hasil *confusion matrix* yang telah didapat maka akan didapatkan nilai *performance matrix* yang akan dijadikan acuan dalam mengukur kinerja pada model yang telah dibuat. Dari Dari model yang telah dibuat berdasarkan percobaan ketiga maka akan menghasilkan *performance matrix* sebagaimana yang telah tersaji pada tabel 4.9 berikut.

Tabel 4. 9 Performa matrik percobaan keempat

	Precision	Recall	F1-Score	Support
Normal	0.67	0.92	0.78	150
Abnormal	0.87	0.55	0.67	151
Accuracy			0.73	301
Macro Avg	0.77	0.73	0.73	301
Weight Avg	0.77	0.73	0.72	301

Dari tabel 4.9 sesuai dengan rumus yang ditetapkan sebelumnya maka didapatkan hasil berupa sensitivitas 73% dan spesifitas 77%. Sesuai dengan rumus yang dijelaskan pada bab 2, kita dapat mengetahui nilai akurasi yang kemudian disebut dengan *score* dengan rumus dimana nilai sensitivitas dijumlahkan dengan nilai spesifitas dibagi dua. Maka dengan rumus yang telah ditetapkan pada percobaan keempat didapat nilai akurasi atau *score* sebesar 75%.

4.4 Mencoba Hasil Model Latih

Setelah selesai proses pealtihan model diuji coba sesuai dengan data *test* yang ada. Pada uji coba kali ini data test yang digunakan sebanyak 8 data sinyal suara jantung *phonocardiogram* berbentuk audio wav. Sebelumnya data model latih yang disimpan di panggil kembali untuk proses uji coba. Dari hasil ujicoba yang dilakukan didapat hasil gambar tabel confusion matrix berikut.



Gambar 4. 15 Konfusi matrix ujicoba model

Dari hasil yang diperlihatkan gambar dapat kita lihat untuk nilai TP sebanyak 4, TN sebanyak 2 dan FP sebanyak 2, Serta FN nya tidak terdeteksi karena data latih sudah sesuai sebanyak 8 data.

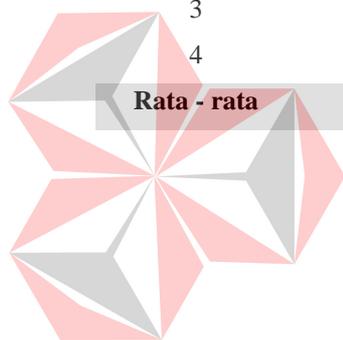
4.5 Analisa Hasil

Dari keempat percobaan yang telah dilakukan maka dapat dilihat hasil yang baik seperti ditunjukkan pada percobaan ketiga. Akurasi test sebesar 81% dan grafik yang tidak terlalu fluktuatif, ini menandakan bahwa data yang loss lebih kecil daripada pada percobaan pertama sebesar 73%, percobaan kedua 72%, percobaan keempat 73% dan dapat disimpulkan bahwa kemungkinan overfitting lebih kecil dibanding yang lain (Junita & Bachtiar, 2019). Sesuai dengan rumus yang

dijelaskan pada bab 2, kita dapat mengetahui nilai akurasi yang kemudian disebut dengan *score* dengan rumus dimana nilai sensitivitas dijumlahkan dengan nilai spesifitas dibagi dua. Maka dengan rumus yang telah ditetapkan pada percobaan pertama hingga percobaan keempat didapat kesimpulan hasil sesuai dengan tabel 4.10.

Tabel 4. 10 Hasil dari keempat percobaan

Percobaan Ke -	Nilai Train	Nilai Test	Sensivitas	Spesivitas	Score (Akurasi)
1	88.4%	73.1%	79%	79%	79%
2	88.4%	72.4%	72%	77%	74.5%
3	91.3%	81.7%	82%	82%	82%
4	90%	73.4%	77%	73%	75%
Rata - rata	89.525%	75.15%	77.5%	77.75%	77.6%



UNIVERSITAS
Dinamika

BAB V

PENUTUPAN

Pada bab ini penulis akan menguraikan tentang kesimpulan dan saran dari hasil pengujian yang telah dilakukan pada penelitian Tugas Akhir ini. Dari hasil kesimpulan dan saran semoga bisa menjadi pengembangan dalam penelitian berikutnya.

5.1 Kesimpulan

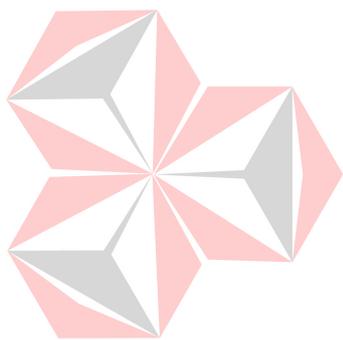
Dari hasil pengujian klasifikasi sinyal jantung PCG menggunakan LSTM maka didapat beberapa kesimpulan yaitu:

1. Dari data yang berupa sinyal suara jantung *phonocardiogram* berektensi wav dan melalui proses normalisasi implementasi menggunakan *long short-term memory* dapat dijalankan.
2. Hasil akurasi model *long short-term memory* dapat dilihat dari grafik dan nilai *confusion matrix*.
3. Pengujian model yang telah dibuat dengan variasi dataset sebanyak 3240 yang dibagi kedalam 70% untuk kebutuhan data latih dan 30% untuk validasi mendapatkan nilai rata – rata akurasi dari 4 kali percobaan sebesar 80% dan data *loss* sebesar 7%
4. Pada pengujian akurasi menggunakan model yang bervariasi nilai *hidden* layernya terhadap variasi proses iterasi, dengan dilakukan sebanyak 4 kali percobaan didapatkan hasil semakin banyak jumlah layer disetiap layer maka waktu yang dibutuhkan untuk proses training data juga semakin lama.
5. Proses pengujian dengan nilai akurasi terbesar didapat dengan variasi model yang diujikan pada proses percobaan ketiga yaitu dengan nilai akhir akurasi sebanyak 91%.
6. Hasil *confusion matrix* yang didapat dari empat kali percobaan disajikan dalam bentuk visual, kemudian didapatkan hasil *performance matrix* yang disajikan dalam bentuk tabel.

5.2 Saran

Adapun saran untuk pengembangan pada penelitian ini agar lebih baik, terdapat beberapa saran yaitu:

1. Menambahkan proses *denoising* untuk menghilangkan derau pada sinyal suara jantung phonocardiogram yang didapat dari physionet sehingga dapat menambah nilai keakuratan model dalam mengklasifikasi.
2. Mengubah bentuk model yang dibuat untuk memaksimalkan proses training agar tidak *overfitting*
3. Untuk hasil maksimal sebaiknya data yang digunakan berupa data sinyal suara jantung pcg yang dijadikan dataset dapat ditambah lag



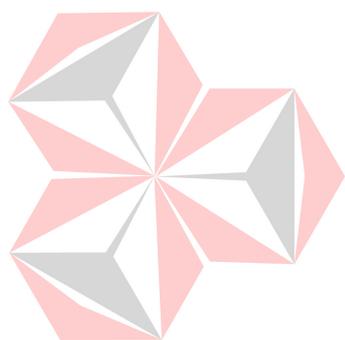
UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Barclay, T., 2020. *Heart*. [Online] Available at: [www.innerbody.com:http://www.innerbody.com/image/card01.html#full-description](http://www.innerbody.com/image/card01.html#full-description) [Accessed 15 April 2020].
- Behbani, S., 2019. A hybrid Algoritm fo Heart Sound Segmentation based on Phonocardioram. *Journal of Medical Engineering & Technology*, pp. 1-15.
- Chung, J. et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Emprint Ervix*.
- Debbal, S. M., 2008. Computerized Heart Sounds Analysis. *Computer in Biology and Medicine*, 38(2), pp. 263-280.
- Gharehbaghi, A. B. M. S. B. J. & A. P., 2015. A Novel Method for Discrimination between Innocent and Pathological Heart Murmurs. *Medical Engineering & Physics*, 37(7), pp. 674-682.
- Graves, A., 2013. Generating Sequences with Recurrent Neural Networks. *Computation Science*.
- Gusmanto, Marindani, E. D. & Sanjaya, B. W., 2016. Rancang Bangus Sistem Peringatan Dini Dan Pelacakan Pada Kendaraan Sepeda Motor Dengan Menggunakan Mikrokontroler Arduino Nano. *Jurusan Teknik Elektro Fakultas Teknik Universitas Tanjungpura*.
- Heryanto, M. A. & Suprijono, H., 2011. APLIKASI GELOMBANG ULTRASOUND PADA TONGKAT PUTIH UNTUK PERINGATAN DINI BAGI PENYANDANG TUNA NETRA. *Jurnal Dian*.
- Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, Volume 9, pp. 1735-1780.
- Ismail, S., Siddiq, I. & Akram, U., 2018. Localization dan Classification of Heart Beat in Phonocardiography Signal - a Comprehensive Review. *EURASIP Journal anda Advances Signal*, Volume 26, pp. 1-27.

- Junita, V. & Bachtiar, F. A., 2019. Klasifikasi Aktivitas Manusia menggunakan Algoritme Decision Tree C4.5 dan Information Gain untuk Seleksi Fitur. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(10), pp. 9426-9433.
- Jusak, J., Puspasari, I. & Kusumawati, W. I., 2021. A Semi-automatic Heart Sounds Identification Model and Its Implementation in Internet of Things Devices. *Advances in Electrical and Computer Engineering*, Volume 21, p. 1.
- Jusak, J., Puspasari, I., Kusumawati, W. I. & Oktarina, E. S., 2020. Model Identifikasi Sinyal Jantung Pertama (S1) dan Sinyal Jantung Kedua (S2) pada Janin. *Jurnal Rekayasa Elektronika*, Volume 16, pp. 50-56.
- Li, S., Li, F., Tang, S. & Xiong, W., 2020. A Review of Computer-Aided Heart Sound Detection Technique. *BiomED Research International*, Volume 2020.
- Maisyaroh, S., 2012. *Rancang Bangun Instrumentasi Elektrokardiografi Berbasis Pc Menggunakan Sound Card*. Medan: Fakultas MIPA Jurusan S1 Fisika, Universitas Negeri Medan.
- M, T. et al., 2016. *Heart sound classification using deep structured*. Vancouver, s.n.
- Orgaanization, W. H., n.d. *Mortality Due Cardiovascular Diseases in World*. [Online]
Available at: <https://www.medicalnewstoday.com/articles/282929>
[Accessed 15 April 2020].
- Pambudi, G. W., 2020. *Belajar Arduino from Zero to Hero (jilid 1)*. Wonogiri: s.n.
- Puspasari, I., Jusak, J., Kusumawati, I. W. & Oktarina, E. S., 2019. *A New Heart Sound Signal Identification Approach Suitabled for Smart Healthcare System*. Indonesia, IEEE.
- Santoso, H., 2015. *Panduan Praktis Arduino Untuk Pemula*. Trenggalek: s.n.
- Sipasulta, R. Y., Lumenta, ST., MT., A. S. & Sompie, ST., MT., S. R., 2014. Simulasi Sistem Pengacak Sinyal Dengan Metode FFT (Fast Fourier Transform). *E-journal Teknik Elektro dan Komputer*, Issue 2301-8402.

- Supriyadi, T., 2018. Tingkat Pintar Sebagai Alat Bantu Pemantau Keberadaan Penyandang Tunanetra Melalui Smartphone. *Seminar Nasional Teknik Elektro*, pp. 181-191.
- Tangdiongan, R. C. G., Allo, E. K. & Sompie, S. R. U. A., 2017. Rancang Bangun Alat Bantu Mobilitas Penderita Tunanetra Berbasis Microcontroller Arduino Uno. *E-Journal Teknik Elektro dan Komputer vol.6 no.2*.
- T, C., S, Y. L. & Ho et al., 2017. S1 and S2 heart sound recognition. *IEEE Transactions on Biomedical*, Volume 64, pp. 372-380.
- W Phanphaisarn, e. a., 2011. Heart detection and diagnosis based on ECG and. *Medical Devices: Evidence and Research*, pp. 133-144.



UNIVERSITAS
Dinamika