

**SKRIPSI**  
**DETEKSI GANGGUAN TIDUR MENGGUNAKAN**  
**METODE JARINGAN SARAF TIRUAN BERBASIS**  
**INTERNET OF THINGS (IoT)**

Oleh:  
**Moch Aripin**  
**065118215**



**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS PAKUAN**  
**BOGOR**  
**2023**

**SKRIPSI**  
**DETEKSI GANGGUAN TIDUR MENGGUNAKAN**  
**METODE JARINGAN SARAF TIRUAN BERBASIS**  
**INTERNET OF THINGS (IoT)**

Diajukan sebagai salah satu syarat untuk memperoleh  
Gelar Sarjana Komputer Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam

Oleh :  
**Moch Aripin**  
**065118215**



**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS PAKUAN**  
**BOGOR**  
**2023**

## HALAMAN PENGESAHAN

JUDUL : DETEKSI GANGGUAN TIDUR MENGGUNAKAN METODE  
JARINGAN SARAF TIRUAN BERBASIS INTERNET OF  
THINGS (IOT)

NAMA : Moch Aripin

NPM : 065118215

**Mengesahkan,**

Pembimbing Pendamping  
Program Studi Ilmu Komputer  
FMIPA – UNPAK



Teguh Puja Negara, S.Si., M.Si.

Pembimbing Utama  
Program Studi Ilmu Komputer  
FMIPA – UNPAK



Prof. Dr.-Ing. Soewarto Hardhienata.

**Mengetahui,**

Ketua Program Studi Ilmu Komputer  
FMIPA – UNPAK



Arie Qur'ania, M.Kom.

Dekan  
FMIPA-UNPAK



Asep Denih, S.Kom., M.Sc., Ph.D.

## PERNYATAAN KEASLIAN KARYA TULIS

Dengan ini saya menyatakan bahwa :

Nama : Moch Aripin  
NPM : 065118215  
Program Studi : Ilmu Komputer  
Fakultas : Matematika dan Ilmu Pengetahuan Alam Universitas  
Pakuan Bogor

Menyatakan bahwa sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian – bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar – benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai pertauran yang berlaku.

Bogor, 15 Agustus 2023



Moch Aripin  
065118215

## **PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

---

Saya yang bertandatangan di bawah ini :

Nama : Moch Aripin  
NPM : 065118215  
Judul Skripsi : Deteksi Gangguan Tidur Menggunakan Metode Jaringan Saraf Tiruan Berbasis Internet Of Things (Iot)

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, 15 Agustus 2023



Moch Aripin  
065118215

## RIWAYAT HIDUP



**Moch Aripin** lahir di Sukabumi pada tanggal 14 Juni 2000 sebagai anak kedua dari tiga bersaudara dari pasangan Bapak Mahpudin dan Ibu Nani Sumarni. Penulis memulai Pendidikan Sekolah Dasar pada tahun 2006 di SDN Cipelang Gede dan lulus pada tahun 2012. Setelah lulus dari Sekolah Dasar pada tahun 2012 penulis melanjutkan Pendidikan Sekolah Menengah Pertama di MTS Syamsul Ulum dan lulus pada tahun 2015. Setelah lulus dari Pendidikan Sekolah Menengah Pertama pada tahun 2015 penulis melanjutkan Pendidikan ke Sekolah Menengah Kejuruan di SMK Teknologi Plus Padjadjaran dengan bidang keahlian Teknik Komputer dan Jaringan dan penulis lulus pada tahun 2018. Setelah lulus dari Sekolah Menengah Kejuruan pada tahun 2018 penulis melanjutkan jenjang Pendidikan ke salah satu perguruan tinggi di Kota Bogor yaitu Universitas Pakuan, Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer. Selama di Universitas Pakuan, Penulis pernah aktif sebagai anggota biasa di Himpunan Mahasiswa Ilmu Komputer (HIMAKOM) periode 2020 – 2022. Pada bulan Juli 2023 penulis telah menyelesaikan penelitian dengan judul “Deteksi Gangguan Tidur Menggunakan Metode Jaringan Saraf Tiruan Berbasis Internet Of Things (Iot)”.

## RINGKASAN

**Moch Aripin 2023.** Deteksi Gangguan Tidur Menggunakan Metode Jaringan Saraf Tiruan Berbasis Internet Of Things (Iot). Dibimbing oleh **Prof. Dr.-Ing. Soewarto Hardhienata dan Teguh Puja Negara, S.Si., M.Si.**

Menurut Gava et al. (2019), tidur adalah aktivitas yang sangat penting bagi manusia karena memungkinkan otak untuk memberikan tubuh waktu untuk beristirahat secara total. Saat seseorang tertidur, kelenjar bawah otak akan melepaskan hormon pertumbuhan yang membantu dalam regenerasi sel dan pertumbuhan tubuh. Oleh karena itu, tidur memegang peran penting dalam hidup manusia. Namun, menurut Hnin et al. (2018), jika kualitas tidur buruk, hal ini akan meningkatkan risiko depresi, gangguan pada sistem kekebalan tubuh, penyakit kronis, bahkan kematian. Salah satu penyebab dari kualitas tidur yang buruk adalah adanya gangguan tidur. Alat diagnostik standar untuk mengidentifikasi gangguan tidur atau sleep apnea dalam dunia medis adalah *Polysomnography* (PSG). Meskipun demikian, alat ini tidak selalu efektif karena membutuhkan pemasangan banyak elektroda pada tubuh pasien, memakan waktu lama dan meminta pasien untuk tidur di rumah sakit, sehingga hasil pengukurannya kurang akurat. Oleh karena itu, diperlukan teknik diagnostik alternatif yang bisa dilakukan secara praktis di rumah tanpa memerlukan perawatan di rumah sakit (Hung, 2018). Rekaman sinyal Electrocardiogram (ECG) terbukti efektif dan efisien dalam mendeteksi gangguan sleep apnea (Sharma, Raval, dan Acharya, 2019). Maka dari itu dibutuhkan teknik untuk mendiagnosis sleep apnea yang dapat dilakukan secara portable di rumah tanpa memerlukan laboratorium tidur rumah sakit. Berdasarkan uraian masalah yang dituliskan diatas, pada penelitian (Moch Aripin, 2023) yang berjudul “Deteksi Gangguan Tidur Menggunakan Metode Jaringan Saraf Tiruan Berbasis Internet Of Things (Iot)”. Sistem deteksi yang akan dibuat akan menggunakan mikrokontroler ESP32 untuk ekstraksi ciri dari ECG dan proses klasifikasi. Harapannya dari sistem deteksi ini untuk mengetahui gangguan tidur seperti CSA dan OSA akan dilakukan dengan menggunakan mikrokontroler. Sistem ini memiliki fitur yang dapat dilihat melalui website dan mempermudah dokter untuk memantau hasil rekaman sinyal pasien secara real-time. Fitur ini membantu dokter dalam melakukan diagnosis awal sebelum dilakukan oleh dokter spesialis dengan mengetahui hasil rekaman sinyal pasien selama 1 malam. Dalam penelitian ini, diperoleh manfaat untuk membantu pasien dengan gejala gangguan tidur dengan memberikan penanganan yang lebih dini. Data yang terkumpul juga bisa memberikan informasi yang lebih akurat bagi tenaga medis, khususnya dokter, karena proses pengumpulan data di rumah sakit berbeda dengan saat pasien melakukannya sendiri di rumah.

## KATA PENGANTAR

Puji Syukur kehadiran Allah SWT, karena rahmat dan hidayah-Nya penulis dapat menyelesaikan karya tulis skripsi yang berjudul **“DETEKSI GANGGUAN TIDUR MENGGUNAKAN METODE JARINGAN SARAF TIRUAN BERBASIS INTERNET OF THINGS (IOT)”** sebagai salah satu syarat dalam mendapatkan gelar sarjana.

Dalam penulisan tugas akhir ini, penulis mengucapkan banyak terimakasih yang sebesar besarnya kepada :

1. Bapak Prof. Dr.-Ing. Soewarto Hardhienata selaku pembimbing utama yang telah berkenan meluangkan waktunya, memberikan petunjuk penulisan laporan ini serta memberikan dorongan moral dan motivasi kepada penulis selama ini.
2. Bapak Teguh Puja Negara, S.Si., M.Si. selaku pembimbing pendamping yang telah meluangkan waktunya dan memberikan arahan.
3. Ibu Arie Qurania, M.Kom. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Pakuan.
4. Orang Tua yang telah memberikan perhatian dan senantiasa memberikan dorongan moral, materil dan motivasi serta doanya kepada penulis.
5. Sahabat, teman seperjuangan yang telah membantu menyelesaikan penulisan laporan tugas akhir ini.
6. Semua pihak yang telah membantu penulis dalam melaksanakan penyusunan laporan ini.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna. Untuk itu penulis mengharapkan kritik dan saran yang sifatnya membangun demi perbaikan ke arah kesempurnaan. Walaupun demikian, penulis berharap karya tulis ini dapat bermanfaat bagi penulis pada khususnya dan para pembaca pada umumnya.

Penulis,

Moch Aripin



## DAFTAR ISI

<b>HALAMAN PENGESAHAN</b> .....	<b>i</b>
<b>PERNYATAAN KEASLIAN KARYA TULIS</b> .....	<b>ii</b>
<b>PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA</b> .....	<b>iii</b>
<b>RIWAYAT HIDUP</b> .....	<b>iv</b>
<b>RINGKASAN</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>DAFTAR ISI</b> .....	<b>vii</b>
<b>DAFTAR GAMBAR</b> .....	<b>x</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Tujuan Penelitian.....	2
1.3. Ruang Lingkup Penelitian .....	2
1.4. Manfaat Penelitian.....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>4</b>
2.1 Tinjauan Pustaka .....	4
2.1.1 Gangguan Tidur .....	4
2.1.2 Elektrokardiogram (EKG).....	5
2.1.3 Sinyal Elektrokardiogram .....	6
2.1.4 Modul Sensor AD8232 .....	6
2.1.5 Microcontroller ESP32.....	7
2.1.6 Jaringan Syaraf Tiruan (JST) .....	7
2.1.7 Heart Rate Variability (HRV) .....	8
2.1.8 Internet of Things (IoT) .....	10
2.2 Penelitian Terdahulu.....	10
2.3 Tabel Perbandingan .....	11
<b>BAB III METODE PENELITIAN</b> .....	<b>12</b>
3.1 Metode Penelitian.....	12
3.1.1 Perencanaan Rancangan Penelitian.....	12
3.1.2 Studi Referensi .....	12
3.1.3 Desain Elektrik.....	13

3.1.4.	Pengadaan Komponen.....	13
3.1.5.	Pengujian Komponen .....	13
3.1.6.	Implementasi Elektrik .....	13
3.1.7.	Desain Perangkat Lunak .....	13
3.1.8.	Implementasi <i>Software</i> .....	13
3.1.9.	Uji Software .....	13
3.1.10.	Desain Sistem Mekanik.....	14
3.1.11.	Integrasi .....	14
3.1.12.	Uji Keseluruhan.....	15
3.1.13.	Aplikasi .....	15
3.2	Alat dan Bahan .....	16
3.2.1	Alat.....	16
3.2.2	Bahan.....	16
<b>BAB IV RANCANGAN DAN IMPLEMENTASI .....</b>		<b>17</b>
4.1.	Perencanaan Rancangan Penelitian .....	17
4.1.1.	Analisa Kebutuhan <i>Hardware</i> .....	17
4.1.2.	Analisa Kebutuhan <i>Software</i> .....	17
4.2.	Studi Referensi .....	18
4.3.	Desain Elektrik .....	18
4.4.	Pengadaan Komponen .....	20
4.5.	Pengujian Komponen .....	20
4.5.1.	Pengujian ESP32.....	20
4.5.2.	Pengujian Sensor AD8232 .....	20
4.6.	Implementasi Elektrik .....	21
4.7.	Desain Software .....	22
4.8.	Implementasi Software.....	23
4.9.	Uji Software .....	23
4.10.	Desain Mekanik .....	24
4.11.	Implementasi Mekanik .....	24
4.12.	Integrasi .....	24
<b>BAB V HASIL DAN PEMBAHASAN .....</b>		<b>25</b>
5.1.	Hasil Penelitian.....	25
5.2.	Test Fungsional Keseluruhan Sistem ( <i>Overall Testing</i> ).....	26
5.2.1.	Pengujian Struktural.....	26

5.2.2. Pengujian Fungsional .....	26
5.2.3. Uji Coba Validasi .....	34
<b>BAB VI KESIMPULAN DAN SARAN.....</b>	<b>40</b>
6.1. Kesimpulan.....	40
6.2. Saran .....	40
<b>DAFTAR PUSTAKA .....</b>	<b>41</b>
<b>LAMPIRAN.....</b>	<b>43</b>

## DAFTAR GAMBAR

Gambar 1. Sinyal OSA dan CSA .....	4
Gambar 2. Alat Elektrodigram .....	5
Gambar 3. Perancangan Penempatan Elektroda Pada Tubuh .....	6
Gambar 4. Sinyal Elektrokardiogram .....	6
Gambar 5. Modul Sensor AD8232.....	7
Gambar 6. Microcontroller ESP32.....	7
Gambar 7 Arsitektur Jaringan Syaraf Tiruan .....	8
Gambar 8 Jaringan Syaraf Tiruan dengan Multilayer Perceptrons (MLP) .....	8
Gambar 9 Frequency FFT .....	9
Gambar 10. Metode Penelitian Hardware Programming.....	12
Gambar 11. Uji Software .....	14
Gambar 12. Uji Keseluruhan.....	15
Gambar 13. Diagram Blok Sistem .....	18
Gambar 14. Desain Elektrik .....	19
Gambar 15 Konektivitas ESP32.....	20
Gambar 16 Pengujian Sensor AD8232 .....	21
Gambar 17 Pengujian 3.3V Sensor .....	21
Gambar 18 Implementasi Elektrik .....	21
Gambar 19. Flowchart Sistem.....	22
Gambar 20 Implementasi Software.....	23
Gambar 21 Uji Software .....	23
Gambar 22 Desain Mekanik Sistem.....	24
Gambar 23 Integrasi .....	24
Gambar 24 Tampilan Tampak Atas Alat .....	25
Gambar 25 Tampilan Tampak Dalam Alat.....	25
Gambar 26 Pengujian ESP32 .....	27
Gambar 27 Contoh Penempatan Elektroda .....	28
Gambar 28 Pembacaan Sinyal Sensor AD8232.....	28
Gambar 29 EasyFFT .....	29
Gambar 30 Perhitungan FFT.....	29
Gambar 31 nnstart Matlab.....	31
Gambar 32 Function Output NNstart.....	31
Gambar 33 Halaman Dashboard Alat .....	34

## DAFTAR TABEL

Tabel 1. Acuan Nilai Normal, OSA dan CSA .....	5
Tabel 2 Penelitian Terdahulu .....	10
Tabel 3 Perbandingan Penelitian.....	11
Tabel 4. Konfigurasi pin Sensor AD8232.....	19
Tabel 5. Konfigurasi Baterai 18560 .....	20
Tabel 6 Pengujian Struktural Sistem.....	26
Tabel 7 Pengujian Fungsional Power Supply .....	26
Tabel 8 Pengujian Fungsional ESP32 .....	27
Tabel 9 Penempatan Elektroda.....	27
Tabel 10 Data pembacaan Sensor AD8232 antara 2 puncak R-Peaks.....	28
Tabel 11 Perhitungan FFT .....	30
Tabel 12 Nilai LF dan HF .....	30
Tabel 13 Nilai Normalisasi Input.....	31
Tabel 14 Bobot dan Bias Hidden Layer.....	32
Tabel 15 Bobot dan Bias Output Layer.....	32
Tabel 16 Uji Coba Validasi Sensor AD8232 .....	34
Tabel 17 Klasifikasi Jaringan Syaraf Tiruan.....	36
Tabel 18 Klasifikasi Jaringan Syaraf Tiruan.....	37
Tabel 19 Uji Coba Validasi Kecepatan Akses Halaman Dashboard .....	38

## DAFTAR LAMPIRAN

Lampiran 1 Perhitungan FFT .....	43
Lampiran 2 Define Function JST .....	48
Lampiran 3 Code Function Menghitung JST .....	48

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Menurut Gava et al. (2019), tidur adalah aktivitas yang sangat penting bagi manusia karena memungkinkan otak untuk memberikan tubuh waktu untuk beristirahat secara total. Saat seseorang tertidur, kelenjar bawah otak akan melepaskan hormon pertumbuhan yang membantu dalam regenerasi sel dan pertumbuhan tubuh. Oleh karena itu, tidur memegang peran penting dalam hidup manusia. Namun, menurut Hnin et al. (2018), jika kualitas tidur buruk, hal ini akan meningkatkan risiko depresi, gangguan pada sistem kekebalan tubuh, penyakit kronis, bahkan kematian. Salah satu penyebab dari kualitas tidur yang buruk adalah adanya gangguan tidur.

Menurut Ananda et al. (2021), di Indonesia, lebih dari 28 juta orang atau 10% dari penduduk negara ini dilaporkan menderita gangguan tidur. Gangguan tidur di Indonesia sangat tinggi, terutama pada anak usia 0-36 bulan, dengan tingkat prevalensi sebesar 31% menurut penelitian oleh Retnosari, Irwanto, dan Herawati (2021). *World Health Organization* (WHO) menyatakan bahwa jenis gangguan tidur pada manusia meliputi *Sleep-Disordered Breathing* (SDB), *REM Behaviour Disorder* (RBD), dan *Restless Legs Syndrome* (RLS). Dari ketiga jenis gangguan tidur tersebut, SDB adalah yang paling sering dialami, mencapai 20% - 40%. SDB meliputi *Obstructive Sleep Apnea* (OSA) dan *Central Sleep Apnea* (CSA), dengan CSA yang jarang terjadi dibandingkan dengan OSA, namun lebih berbahaya karena dapat mengakibatkan komplikasi seperti stroke dan gagal jantung (Culebras dan Anwar, 2018). CSA adalah penyakit gangguan tidur yang terjadi karena otak tidak mengirimkan sinyal ke tubuh, sehingga otot yang mengontrol pernapasan berhenti sejenak saat seseorang tidur (McLaren, Bin-Hasan, dan Narang, 2019).

Alat diagnostik standar untuk mengidentifikasi gangguan tidur atau sleep apnea dalam dunia medis adalah *Polysomnography* (PSG). Meskipun demikian, alat ini tidak selalu efektif karena membutuhkan pemasangan banyak elektroda pada tubuh pasien, memakan waktu lama dan meminta pasien untuk tidur di rumah sakit, sehingga hasil pengukurannya kurang akurat. Oleh karena itu, diperlukan teknik diagnostik alternatif yang bisa dilakukan secara praktis di rumah tanpa memerlukan perawatan di rumah sakit (Hung, 2018). Rekaman sinyal Electrocardiogram (ECG) terbukti efektif dan efisien dalam mendeteksi gangguan sleep apnea (Sharma, Raval, dan Acharya, 2019).

Beberapa penelitian terdahulu yang telah dilakukan diantaranya mengenai gangguan tidur antara lain. Ratnasari (2022) Menganalisis gangguan tidur menggunakan durasi QRS dan RR interval menggunakan metode *Neural Network*, sistem dirancang sebagai pendeteksi aktifitas listrik jantung, dengan 3 buah elektroda menempel pada dada untuk merekam lalu diekstraksi fitur RR interval dan durasi QRS hasilnya ditampilkan pada smartphone dengan konektivitas Bluetooth. Namun alat ini hanya bisa mengirim data apabila terhubung dalam 1 perangkat bluetooth dan apabila jarak lebih dari 5 meter dalam ruangan dan 10 meter pada berbeda ruangan maka tidak bisa melakukan pengiriman data. Penelitian

lainnya dilakukan oleh Nguyen (2022) Menganalisis perbandingan antara *Central Sleep Apnea* dengan *Obstructive sleep apnea* menggunakan ResNet50.

Berdasarkan permasalahan dan penelitian terkait, maka dibutuhkan teknik untuk mendiagnosis sleep apnea yang dapat dilakukan secara portable di rumah tanpa memerlukan laboratorium tidur rumah sakit. Harapannya dari sistem deteksi ini untuk mengetahui gangguan tidur seperti CSA dan OSA akan dilakukan dengan menggunakan mikrokontroler. Sistem ini memiliki fitur yang dapat dilihat melalui website dan mempermudah dokter untuk memantau hasil rekaman sinyal pasien secara real-time. Fitur ini membantu dokter dalam melakukan diagnosis awal sebelum dilakukan oleh dokter spesialis dengan mengetahui hasil rekaman sinyal pasien selama 1 malam.

Sistem deteksi yang akan dibuat akan menggunakan mikrokontroler ESP32 untuk ekstraksi ciri dari ECG dan proses klasifikasi. Modul AD8232 digunakan sebagai sensor untuk mendapatkan sinyal ECG dari jantung pasien. Sistem ini menggunakan 3 elektroda dengan teknik *Fontaine bipolar precordial lead* (F-ECG) dan elektroda tersebut diletakkan di tengah dada atas, tengah dada bawah, dan di kanan sejajar dengan elektroda sebelumnya sebagai acuan. Sistem ini memiliki akurasi dan waktu komputasi yang baik dan dapat diterapkan saat dirumah. Modul WiFi pada ESP01 digunakan untuk mengirimkan hasil klasifikasi ke database dan dapat dilihat melalui website secara langsung. Sistem ini membutuhkan kemampuan deteksi yang portabel, cepat dan akurat. Adanya sistem ini diharapkan dapat membantu dokter memantau kondisi pasien saat tidur karena hasil deteksi dapat disimpan dan digunakan untuk pemeriksaan lebih lanjut.

## **1.2. Tujuan Penelitian**

Penelitian ini bertujuan untuk merancang alat pendeteksi gangguan tidur menggunakan metode jaringan saraf tiruan berbasis internet of things (IoT).

## **1.3. Ruang Lingkup Penelitian**

Ruang lingkup penelitian ini meliputi :

1. Penelitian ini mendeteksi dua dari macam-macam gangguan tidur yaitu Central Sleep Apnea dan Obstructive Sleep Apnea.
2. Proses pengecekan kondisi subjek atau pasien wajib dalam posisi berbaring atau tidur.
3. Hanya aktivitas sinyal biolistrik tubuh yang dapat terdeteksi oleh sensor.
4. Data sinyal *Sleep-Disordered Breathing* didapatkan dari *database online physionet (St. Vincent's University Hospital / University College Dublin Sleep Apnea Database)*.
5. Penggunaan fitur *Low* dan *High Frequency* hanya berkisar pada powernya diluar *frequency* tidak digunakan.
6. Alat ini juga minimal bisa mendekati kinerja sleep apnea meter yang sudah ada di pasaran.



#### **1.4. Manfaat Penelitian**

Manfaat penelitian adalah sebagai berikut:

Dalam penelitian ini, diperoleh manfaat untuk membantu pasien dengan gejala gangguan tidur dengan memberikan penanganan yang lebih dini. Data yang terkumpul juga bisa memberikan informasi yang lebih akurat bagi tenaga medis, khususnya dokter, karena proses pengumpulan data di rumah sakit berbeda dengan saat pasien melakukannya sendiri di rumah.

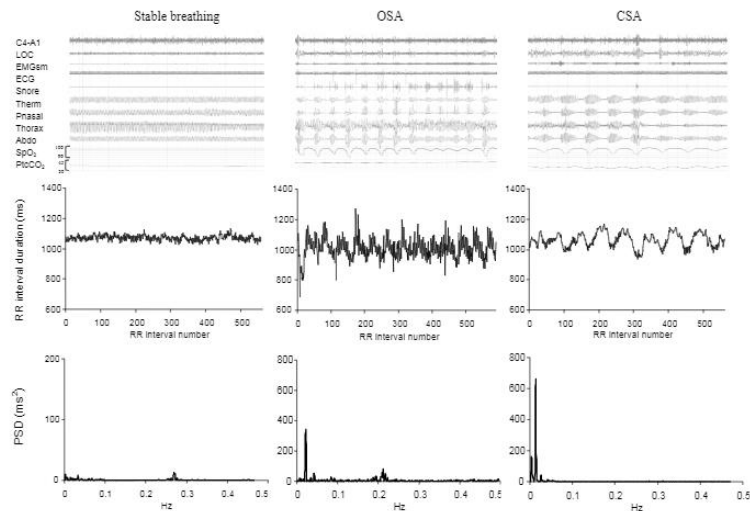
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

##### 2.1.1 Gangguan Tidur

Gangguan tidur merupakan kondisi tidak normal atau sulit bernafas selama tidur berlangsung. Gangguan pernafasan ini membuat pasiennya mengalami insomnia atau gangguan tidur dan merasa tidak nyaman ketika terbangun dari tidurnya yang berakibat pada kurang produktif dalam menjalani harinya. Menurut penelitian (Sateia, 2014) ada beberapa tipe gangguan tidur, yang sering terjadi ada 2 yaitu: *Obstructive Sleep Apnea* (OSA) dan *Central Sleep Apnea* (CSA). OSA ditandai dengan sumbatan yang berulang pada saluran pernafasan dibagian atas selama tidur yang menyebabkan pengurangan atau penghentian total pernafasan (Kaplan & Ulualp, 2022). Sedangkan, CSA terjadi ketika otak tidak mengirimkan sinyal yang benar ke jalannya pernafasaan saat manusia tertidur dimana kondisi ini menyebabkan pengurangan atau penghentian pernafasan saat tidur (Felix et al., 2016). Perbedaan sinyal OSA dan CSA ditunjukkan pada gambar



Gambar 1. Sinyal OSA dan CSA

Sumber: (Szollosi et al., 2007)

Dapat dilihat bahwa sinyal OSA cenderung kasar dan memiliki nilai frekuensi kecil sedangkan CSA memiliki sinyal yang jauh perbedaan intervalnya dan relative memiliki frekuensi yang tinggi. Dimana nilai ini didapatkan dengan rumus *Fast Fourier Transform* (FFT) kemudian di ambil nilai powernya yang dijelaskan pada tinjauan pustaka *Heart Rate Variability*. Fitur yang digunakan adalah *frequency Low* (LF) dan *High. Low Frequency* (HF) adalah nilai fitur data ECG yang diambil dan dilakukan proses *Fast Fourier Transform* dan diambil pada nilai *frequency* 0.04 – 0.15 sedangkan *High Frequency* pada nilai 0.15 – 0.5 dari data itu kemudian dibuat Tabel acuan nilai Normal, CSA dan OSA maupun normal dapat dilihat pada tabel 1.

Tabel 1. Acuan Nilai Normal, OSA dan CSA

No	HF	LF	Kelas
1	1.81	1.38	Normal
2	0.78	0.85	OSA
3	0.57	0.45	CSA

Sumber: (Szollosi et al., 2007)

### 2.1.2 Elektrokardiogram (EKG)

Elektrokardiogram (EKG) merupakan alat untuk merekam aktivitas kelistrikan pada jantung manusia. Alat ini juga digunakan untuk menganalisa atau mendiagnosik dari bentuk gelombang sinyal yang dihasilkan. Gelombang sinyal yang dihasilkan memiliki detak jantung, irama jantung dan morfologi pada manusia. Informasi yang dikirim dari jantung manusia antara lain.

1. Posisi dari jantung.
2. Banyak detak jantung.
3. Irama dari jantung
4. Lokasi dari *Myocardial Ischemia*

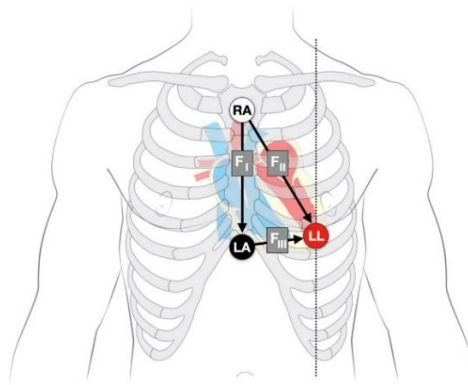
EKG biasanya dilakukan oleh dokter di rumah sakit untuk mendeteksi aktivitas jantung dengan memasang elektroda pada tubuh pasien. Karena dokter yang lebih mengetahui sinyal EKG normal dan tidak normalnya. Alat elektrokardiogram dapat dilihat pada Gambar 2.



Gambar 2. Alat Elektrodiogram

Sumber: (Florenzia, 2019)

Perancangan penempatan elektroda menunjukkan pemasangan elektroda yang digunakan saat melakukan pengujian. Pada pengujian ini menggunakan 3 buah elektroda dengan teknik penempatan *Fontain bipolar precordial leads* (F-ECG). Penempatan elektroda dapat dilihat pada Gambar 3.



Gambar 3. Perancangan Penempatan Elektroda Pada Tubuh

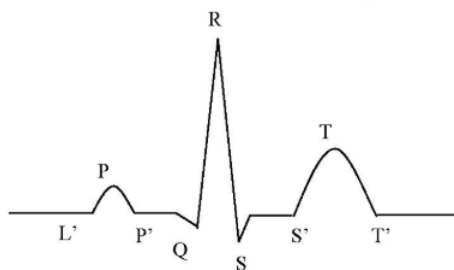
Sumber: (litfl.com, 2022)

Pada Gambar 3 menunjukkan penempatan 3 buah elektroda (RA, LA, LL) sehingga ketika pengujian alat pasien atau orang yang ingin melakukan deteksi dini harus sesuai dengan perancangan penempatan elektroda. Dimana elektroda warna merah RA diletakkan pada dada tengah atas sebagai VCC. Kemudian elektroda warna kuning LA diletakkan data tengah bawah sebagai Ground dan elektroda LL warna hijau diletakkan pada dada kanan sebagai referensi.

### 2.1.3 Sinyal Elektrokardiogram

Sinyal elektrokardiogram menggambarkan aktivitas listrik jantung. Aktivitas listrik berhubungan dengan impuls yang berjalan melalui jantung. Informasi yang diberikan yaitu: detak jantung, ritme, dan morfologi. Gelombang yang dihasilkan terdiri dari gelombang P, QRS kompleks dan gelombang T. Gambar 4 menggambarkan dari bentuk sinyal EKG normal. Gelombang P merupakan cerminan dari depolarisasi dari atrium kanan dan kiri. QRS kompleks merupakan hubungan dengan depolarisasi *ventrikel* kanan dan kiri. Gelombang T merupakan cerminan repolarisasi *ventrikel*.

Sumber: (Wang, Plataniotis and Hatzinakos, 2006)



Gambar 4. Sinyal Elektrokardiogram

### 2.1.4 Modul Sensor AD8232

Dibutuhkan untuk membaca aktivitas sinyal jantung yang nantinya sebagai input untuk klasifikasi Jaringan Syaraf Tiruan, maka perangkat yang dipilih adalah Modul sensor AD8232. Modul sensor AD8232 ini menggunakan 3 buah elektroda yang akan ditempelkan pada dada atas, dada tengah dan dada bagian bawah. Pembacaan modul sesor AD8232 akan masuk melalui pin analog pada

mikrokontroler ESP32. Sinyal yang masuk nantiya akan dilakukan ekstraksi *High Frequency* dan *Low Frequency*.



Gambar 5. Modul Sensor AD8232

### 2.1.5 Microcontroller ESP32

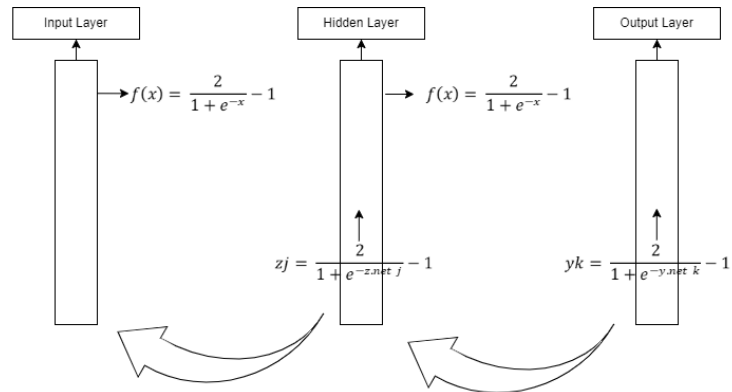
ESP32 adalah nama mikrocontroller yang dirancang oleh Espressif Systems. ESP32 diperkenalkan sebagai solusi jaringan WiFi mandiri yang menawarkan sebagai jembatan dari microcontroller ke pengontrol ke WiFi selain itu juga mampu menjalankan aplikasi mandiri (Kolban, 2018).



Gambar 6. Microcontroller ESP32

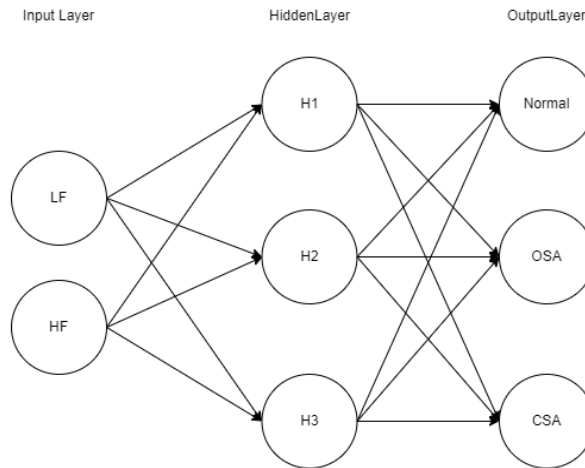
### 2.1.6 Jaringan Syaraf Tiruan (JST)

*Syaraf Tiruan* merupakan metode *Machine Learning* (ML) yang cara kerjanya menyerupai otak manusia. Metode ini melakukan proses belajar dengan melakukan perhitungan nilai bobot berdasarkan lapisannya. Terdapat 3 lapisan pada *JST* yaitu: *input layer*, *hidden layer* dan *output layer*. *Input layer* merupakan banyak dari fitur yang akan dimasukkan kedalam arsitekturnya biasanya *input layer* dinyatakan dalam bentuk vektor  $P$  dengan  $L$  baris (Fahruzi, et al., 2020). *Hidden layer* merupakan lapisan antara *input layer* dengan *output layer* dalam *hidden layer* ini akan menentukan *neuron* yang nantinya akan memiliki sekumpulan pembobot *weight* dari *input layer* untuk menghasilkan *output layer*. *Output layer* merupakan lapisan terakhir dari *neuron* yang merupakan hasil dari *output* sistem. Gambaran dari JST dapat dilihat pada Gambar 7.



Gambar 7 Arsitektur Jaringan Syaraf Tiruan

*Jaringan Syaraf Tiruan* yang digunakan dalam penelitian ini adalah jenis *feedforward* yang disebut *multilayer perceptron* (MLP). MLP memiliki fungsi aktivasi nonlinear yang memungkinkan JST untuk mempelajari kegiatan berdasarkan data masa lalu dan memberikan keputusan terhadap data yang belum pernah dipelajari sebelumnya. Rincian setiap layer dapat dilihat pada Gambar 8.



Gambar 8 Jaringan Syaraf Tiruan dengan *Multilayer Perceptrons* (MLP)

*Jaringan Syaraf Tiruan* yang digunakan dalam penelitian ini memiliki struktur dengan dua lapisan input, tiga lapisan tersembunyi, dan tiga lapisan output. Proses pembelajaran menggunakan metode *feedforward*, yang melakukan perhitungan terhadap bobot-bobot dari lapisan input hingga lapisan output. Aktivasi yang digunakan untuk menghitung probabilitas adalah aktivasi *softmax*.

### 2.1.7 Heart Rate Variability (HRV)

*Heart rate variability* adalah pengukuran kecepatan detak jantung. Detak jantung ini juga dapat dirasakan dengan mengambil denyut nadi pada leher atau pada pergelangan tangan. HRV bervariasi setiap orang secara alami akan menurun seiring dengan bertambahnya usia. HRV yang rendah sering dikaitkan dengan tingkat stres tinggi dan adanya kondisi penyakit lainnya. HRV ini juga sering digunakan untuk ekstraksi fitur yang nantinya untuk mendeteksi atau menganalisis

suatu penyakit terutama pada sinyal EKG. Pada penelitian ini menggunakan fitur *High Frequency* (LF) dan *Low Frequency* (HF) yang dihitung menggunakan rumus *Fast Fourier Transform* (FFT).

$$S(f) = \int_{-\infty}^{\infty} s(t)Wdt \tag{1}$$

$S(f)$  : Sinyal dalam domain frekuensi (*frequency domain*)

$S(t)$  : Sinyal dalam domain waktu (*time domain*)

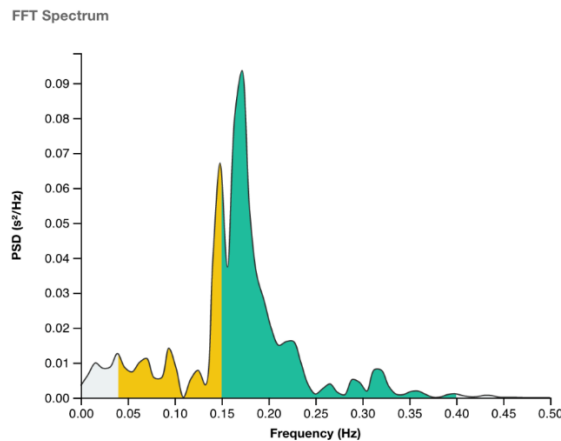
$W$  : Konstata nilai sebuah sinyal

Nilai  $W$  dapat dilihat pada rumus dibawah ini :

$$W = e^{\frac{-2\pi j}{N}}$$

$$W = \cos \frac{\pi}{40} - j \sin \frac{\pi}{40}$$

Dari perhitugnan FFT diatas akan didapatkan nilai *frequency* yang nantinya akan dilakukan perhitungan nilai HF dan LF pada hasil *frequency* tersebut. Sebagai ilustrasi gambaran sinyal seperti gambar dibawah ini.



Gambar 9 Frequency FFT

Sumber: (elitehrv.com)

Yang nantinya akan diambil HF dan LFnya dimana power *Frequency* dari LF berkisar antara 0.04 – 0.15 Hz dan HF berkisar antara 0.15 – 0.4 Hz yang akan dirata-rata dan data itu akan digunakan dalam input JST.

### 2.1.8 Internet of Things (IoT)

IoT adalah perangkat apapun yang terhubung ke internet sehingga dapat berkomunikasi dengan perangkat lain. IoT juga sebagai network perangkat perangkat yang terhubung dan setiap perangkat memiliki alamat IP yang berbeda. Teknologi internet memberikan kemudahan salah satunya dalam bidang monitoring polusi udara. (Suriansyah, 2018).

## 2.2 Penelitian Terdahulu

Penelitian terdahulu tentang gangguan tidur ada yang pernah dilakukan diantaranya sebagai berikut:

Tabel 2 Penelitian Terdahulu

1.	Nama	Ratnasari
	Judul	“Sistem Pendeteksi <i>Central Sleep Apnea</i> Menggunakan Metode <i>Neural Network</i> Dengan Fitur <i>RR Interval</i> dan Durasi QRS.”
	Isi	Menganalisis gangguan tidur menggunakan Durasi QRS dan <i>RR Interval</i> Menggunakan metode <i>Neural Network</i>
2.	Nama	Nguyen, et al.,
	Judul	“A novel deep learning-based approach for sleep apnea detection using single-lead ECG signals.”
	Isi	Menganalisis pebandingan antara <i>Central Sleep Apnea</i> dengan <i>Obstructive Sleep Apnea</i> Menggunakan <i>ResNet50</i>
3.	Nama	Moch Aripin
	Judul	Deteksi gangguan tidur menggunakan metode jaringan syaraf tiruan berbasis Internet of Things
	Isi	Merancang alat pendeteksi gangguan tidur menggunakan metode jaringan saraf tiruan



### 2.3 Tabel Perbandingan

Perbandingan penelitian sebagai berikut:

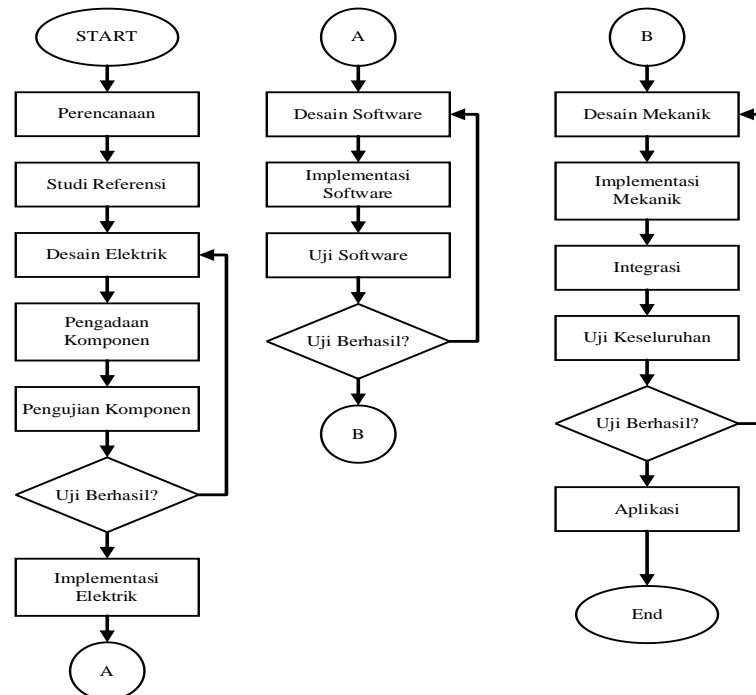
Tabel 3 Perbandingan Penelitian

No	Penelitian dan Tahun	Input			Proses	Metode			Output Interface		
		HRV	Elektroda	Single-lead		ESP32	Neural Network/JST	ResNet50	LCD	WEB	Internet of Things
1	Ratnasari et al (2022)		✓			✓		✓			
2	Nguyen et al (2022)		✓	✓			✓				
3	Moch Aripin (2023)	✓	✓	✓	✓	✓			✓	✓	

## BAB III METODE PENELITIAN

### 3.1 Metode Penelitian

Metode yang digunakan dalam penelitian “Deteksi gangguan tidur menggunakan metode jaringan syaraf tiruan berbasis *Internet of Things*” ini menggunakan metode penelitian bidang hardware programming yang ditunjukkan pada gambar berikut



Gambar 10. Metode Penelitian Hardware Programming

#### 3.1.1. Perencanaan Rancangan Penelitian

Dalam perencanaan penelitian ini, terdapat beberapa hal penting yang harus diperhatikan dan dipertimbangkan, antara lain :

1. Penentuan topik dan kerangka awal penelitian
2. Estimasi kebutuhan alat dan bahan
3. Estimasi anggaran
4. Perangkat lain, dan
5. Penerapan dari model alat yang telah dirancang.

#### 3.1.2. Studi Referensi

Setelah perencanaan yang dilakukan telah matang, maka dilanjutkan penelitian awal dari hardware yang akan dibuat, dimulai dari pemilihan dan pengesanan komponen (alat dan bahan), memilih komponen yang tepat dan sesuai merupakan hal yang membutuhkan ketelitian. Hal ini akan memberikan pengaruh terhadap hasil akhir dari penelitian ini.

### **3.1.3. Desain Elektrik**

Dalam merancang desain elektrik terdapat beberapa hal yang harus diperhatikan, antara lain :

1. Sumber tegangan dan pembagian daya untuk masing-masing komponen
2. Kebutuhan daya untuk Mikrokontroler yang akan digunakan
3. Desain skema rangkaian
4. Pengetesan skema listrik yang telah dirancang.

### **3.1.4. Pengadaan Komponen**

Pengadaan komponen adalah tahap persiapan pengumpulan komponen-komponen yang akan di pakai nantinya agar pada saat proses perakitan tidak terhenti karena kekurangan komponen. Setelah pengadaan komponen selesai lalu dilanjut ke proses pengujian komponen.

### **3.1.5. Pengujian Komponen**

Dalam pengetesan komponen dilakukan terhadap fungsi dari masing-masing komponen yang akan digunakan sesuai kebutuhan dari sistem yang sebelumnya sudah didesain.

### **3.1.6. Implementasi Elektrik**

Implementasi elektrik adalah pengimplementasian dari gambaran rangkaian desain listrik yang telah dibuat sebelumnya.

### **3.1.7. Desain Perangkat Lunak**

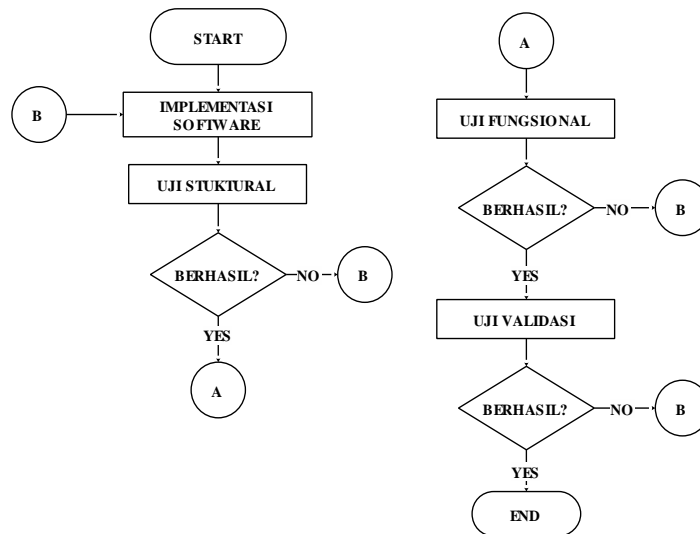
Dalam desain perangkat lunak yang digunakan dalam penelitian ini menggunakan perangkat lunak MS. Office, Google Chrome, Fritzing, Arduino IDE 1.8.10, Visual Studio Code dan untuk Bahasa pemrograman yang digunakan yaitu *pemrograman* Bahasa C++.

### **3.1.8. Implementasi Software**

Implementasi *software* adalah pengimplementasian dari gambaran desain *software* yang telah di buat sebelumnya. Kemudian setelah pengimplementasian *software* selesai di lanjut ke tahap uji *software*.

### **3.1.9. Uji Software**

Pengujian *software* dilakukan agar desain yang telah dibuat sebelumnya sesuai dengan yang di inginkan sehingga pada saat penelitain bisa berfungsi dengan baik. Uji software meliputi uji struktural, uji fungsional dan uji validasi.



Gambar 11. Uji Software

### 3.1.9.1. Uji Struktural

Uji struktural pada software untuk mengetahui apakah software yang telah di buat dapat berfungsi dengan benar atau tidak.

### 3.1.9.2. Uji Fungsional

Uji fungsional untuk mengintegrasikan sistem *software* yang telah di desain sebelumnya.

### 3.1.9.3. Uji Validasi

Uji Validasi untuk menguji kinerja dari software yang telah dibuat apakah software tersebut dapat berjalan dengan baik atau tidak.

### 3.1.10. Desain Sistem Mekanik

Dalam perancangan perangkat keras, desain mekanik merupakan hal penting. Tahap desain sistem mekanis merupakan tahap dilakukannya pertimbangan meliputi kebutuhan sistem yang akan dibuat terhadap desain mekanik, diantaranya:

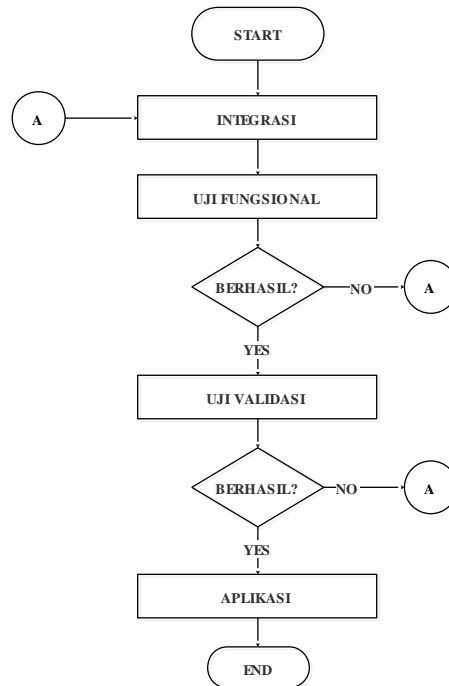
1. Bentuk dan ukuran PCB (*Printed Circuit Board*)
2. Ketahanan dan fleksibilitas terhadap lingkungan
3. Penempatan modul-modul elektronik
4. Pengetesan sistem mekanik yang telah di rancang
5. Bentuk desain ukuran interface hardware.

### 3.1.11. Integrasi

Modul listrik yang diintegrasikan dengan *software* di dalam kontrollernya, kemudian diintegrasikan dalam struktur mekanik yang telah dirancang. Lalu dilakukan uji keseluruhan. Uji keseluruhan meliputi uji fungsional, dan uji validasi.

### 3.1.12. Uji Keseluruhan

Pada tahapan ini dilakukan pengujian fungsi dari keseluruhan sistem. Pengetesan ini bertujuan untuk mengetahui apakah sistem yang dibuat sesuai dengan rancangannya atau tidak. Bila ada sistem yang tidak dapat bekerja dengan baik maka harus dilakukan proses perakitan ulang pada setiap desain sistemnya.



Gambar 12. Uji Keseluruhan

#### 3.1.12.1. Uji Fungsional

Uji fungsional dilakukan untuk mengintegrasikan sistem listrik dan software yang telah di desain. Tes ini dilakukan untuk meningkatkan performa dari perangkat lunak untuk pengontrolan desain listrik dan mengeliminasi error (bug) dari software yang telah dibuat.

#### 3.1.12.2. Uji Validasi

Tahap ini bertujuan untuk menguji kinerja dari alat yang telah dibuat apakah alat tersebut dapat berjalan dengan baik atau tidak.

#### 3.1.13. Aplikasi

Pengoptimalan dilakukan untuk meningkatkan performa dari aplikasi yang telah dirancang. Lalu optimasi ditekankan pada desain mekanik dan perangkat lunak agar penggunaan lebih maksimal dan tidak terjadi error.

### **3.2 Alat dan Bahan**

Untuk mendukung proses penelitian diperlukan beberapa alat dan bahan.

#### **3.2.1 Alat**

Adapun alat yang akan digunakan adalah sebagai berikut:

1. Hardware
  - Laptop Asus A456uf
  - Multi Meter
2. Software
  - Arduino Uno
  - Visual Studio Code
  - Ms. Office
  - Visio
  - Matlab
  - Chrome Browser

#### **3.2.2 Bahan**

Adapun bahan bahan yang akan digunakan sebagai berikut

1. Mikrokontroler ESP32
2. Baterai 18650
3. Modul Sensor AD8232
4. Kabel
5. Buku panduan skripsi dan tugas akhir Prodi Ilmu Komputer Fakultas MIPA.  
Jurnal referensi yang terkait dengan penelitian ini

## **BAB IV**

### **RANCANGAN DAN IMPLEMENTASI**

#### **4.1. Perencanaan Rancangan Penelitian**

Tahapan perencanaan proyek penelitian adalah tahapan kegiatan dari proses pembuatan sistem. Komponen yang dibutuhkan dalam perencanaan sistem yang akan dibangun yaitu mikrokontroler ESP32, sensor AD8232 sebagai pembacaannya. Metode yang digunakan menggunakan metode jaringan syaraf tiruan.

##### **4.1.1. Analisa Kebutuhan *Hardware***

Dalam melakukan penelitian dibutuhkan analisa kebutuhan hardware diantaranya :

1. Pemilihan Modul Mikrokontroller

Modul mikrokontroller yang digunakan untuk membangun sistem ini hanya menggunakan ESP32. ESP32 sudah terintegrasi dengan berbagai fitur layaknya mikrokontroler dan sudah bisa terkoneksi dengan wifi tanpa harus di sambungkan dengan perangkat NODMCU8266 untuk terhubung ke internet.

2. Pemilihan Sensor

Dalam membuat system ini menggunakan sensor AD8232. Sensor AD8232 adalah sensor yang sering digunakan dalam aplikasi pengukuran aktivitas jantung non-invasif. Sensor ini memiliki kemampuan untuk mendeteksi sinyal listrik yang dihasilkan oleh jantung dan mengubahnya menjadi sinyal analog yang dapat diolah lebih lanjut. Keunggulan dari sensor AD8232 adalah kemampuannya untuk menghilangkan kebisingan dan interferensi elektromagnetik, sehingga memberikan hasil yang akurat dan stabil. Selain itu, sensor ini juga mudah digunakan dan dapat diintegrasikan dengan berbagai platform mikrokontroler dan sistem elektronik lainnya. Dalam pemilihan sensor AD8232, penting untuk mempertimbangkan kebutuhan aplikasi, akurasi yang diinginkan, serta kemudahan integrasinya dengan sistem yang ada. Dengan kemampuannya yang handal dan kinerja yang stabil, sensor AD8232 menjadi pilihan yang baik dalam pengukuran aktivitas jantung yang dapat digunakan dalam berbagai bidang, termasuk pemantauan kesehatan, sistem pemantauan pasien, dan proyek-proyek elektronik lainnya.

##### **4.1.2. Analisa Kebutuhan *Software***

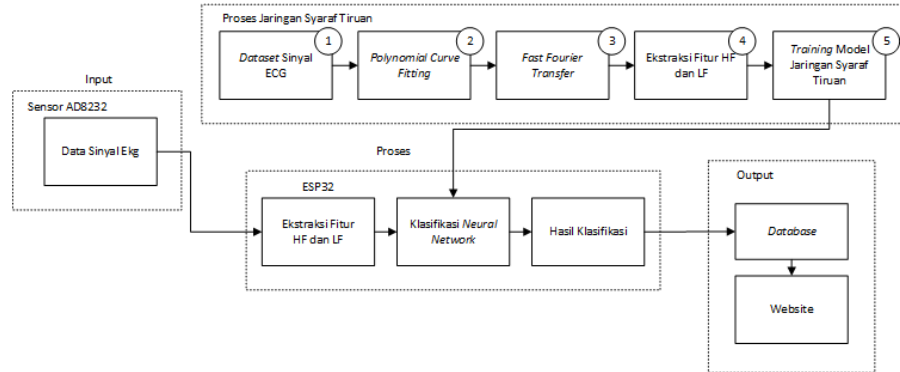
Analisa kebutuhan software untuk alat *deteksi gangguan tidur menggunakan metode jaringan saraf tiruan berbasis internet of things (iot)*. Visual Studio Code merupakan sebuah perangkat lunak yang digunakan untuk mempermudah pembuatan dan pengembangan sistem yang akan dibangun mulai dari menuliskan *source* program sampai *upload* hasil kompilasi, dan uji coba secara terminal serial. Adapun *software* untuk membuat *Aplikasi Adroid* yaitu Visual Studio Code. Visual Studio Code adalah teks editor *open-source*.

## 4.2. Studi Referensi

Setelah melakukan tahapan perencanaan sistem, kemudian dilanjutkan dengan penelitian awal dari sistem yaitu melakukan perancangan rangkaian mekanik serta komponen dari model sistem untuk memastikan bahwa semua komponen dapat berfungsi dengan baik dan dapat digunakan secara optimal. Sistem ini menggunakan sebuah Mikrokontroler ESP32. Input sistem menggunakan sensor AD8232. Output system berupa tampilan pada Web.

## 4.3. Desain Elektrik

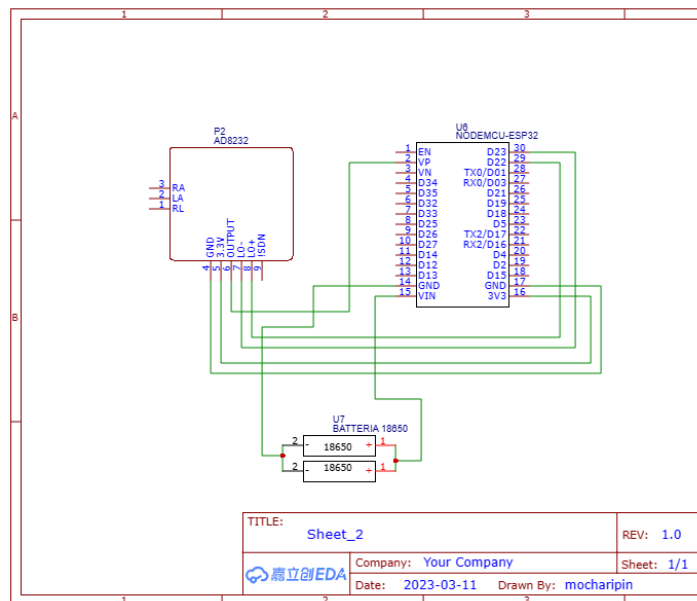
Berikut merupakan diagram blok sistem yang akan dibuat



Gambar 13. Diagram Blok Sistem

Gambar 13 merupakan ilustrasi blok diagram perangkat keras yang dirancang pada penelitian ini. Pada diagram input terdapat 18650 yang digunakan untuk mensuplay daya ke dalam ESP32, kemudian terdapat sensor AD8232 yang digunakan sebagai input untuk membaca aktivitas sinyal listrik pada jantung dengan menggunakan 3 buah elektroda. Selanjutnya pada diagram proses Arduino akan melakukan pengambilan fitur dari sensor AD8232 yang nantinya akan dilakukan klasifikasi Jaringan Syaraf Tiruan dimana nilai bobot dari Jaringan Syaraf Tiruan didapatkan dari proses *training* dari *software* matlab. Dataset didapatkan dari *physionet* yang dilakukan *polynomial curve fitting*, hal tersebut dilakukan agar sinyal nilai tengahnya adalah di nol. Setelah itu dilakukan proses ekstraksi *fast fourier transfer* dan diambil fitur *high frequency* dan *low frequency* dan melakukan *training* model Jaringan Syaraf Tiruan. Hasil dari klasifikasi akan dikirimkan ke dalam *database*. Pada diagram output akan menerima data yang dikirimkan oleh ESP32 dan akan ditampilkan ke dalam website dengan menggunakan *framework Laravel*.





Gambar 14. Desain Elektrik

Komponen dari diagram blok disusun sedemikian rupa mengikuti skematik yang ditunjukkan pada Gambar 14. Sensor AD8232 dan ESP32 mendapatkan catu daya dari ESP32.3V dan ESP32 mendapatkan daya dari baterai 18650 yang terhubung ke VIN dan GND.

Berdasarkan skematik yang sudah dirancang, maka bisa dirancang konfigurasi pin untuk masing-masing sensor dan aktuator. Konfigurasi pin yang dirancang adalah konfigurasi untuk setiap komponen dalam sistem. Setiap komponen dalam sistem mencakup sensor AD8232, ESP32 dan Baterai 18650. Konfigurasi pin untuk keseluruhan sensor dan aktuator pada sistem adalah sebagai berikut:

Tabel 4. Konfigurasi pin Sensor AD8232

ESP32	Sensor AD8232
GND	GND
3.3V	3.3V
Pin VP	OUTPUT
Pin D23	LO-
Pin D22	LO+

Tabel 4 menunjukkan konfigurasi pin ESP32 terhadap sensor AD8232. Sensor AD8232 beroperasi menggunakan sumber tegangan 3.3V dan GND dari ESP32. Untuk pin output terhubung ke pin VP, pin LO- dan LO+ dari sensor AD8232 terhubung ke pin D23 dan D22 ESP32. Pada sensor AD8232 terdapat 1 pin tambahan yaitu pin SDN, namun tidak digunakan karena tidak mempengaruhi fungsionalitas sistem yang dirancang.

Tabel 5. Konfigurasi Baterai 18560

ESP32	Baterai 18560
VIN	Positif (+)
GND	Positif (-)

Tabel 5 menunjukkan konfigurasi pin ESP32 terhadap Baterai 18560 ESP32 langsung mendapatkan daya dari Baterai 18560 yang masuk melalui pin VIN dan GND ESP32. Kemudian daya tersebut disebar ke Modul WiFi ESP01 dan Sensor AD8232 melalui pin 3.3v ESP32.

#### 4.4. Pengadaan Komponen

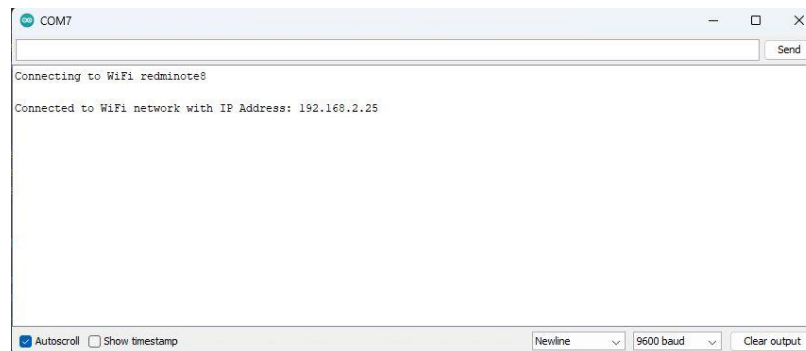
Setelah komponen semuanya sudah terkumpul maka selanjutnya di lakukan pengujian pada setiap komponen.

#### 4.5. Pengujian Komponen

Pada tahap ini dilakukan pengetesan semua komponen yang akan digunakan menggunakan multimeter, pengetesan ini meliputi input dan output voltase dari komponen dan sensor, dan pengetesan menggunakan Platform serial monitor dengan cara melihat output dari masing-masing komponen.

##### 4.5.1. Pengujian ESP32

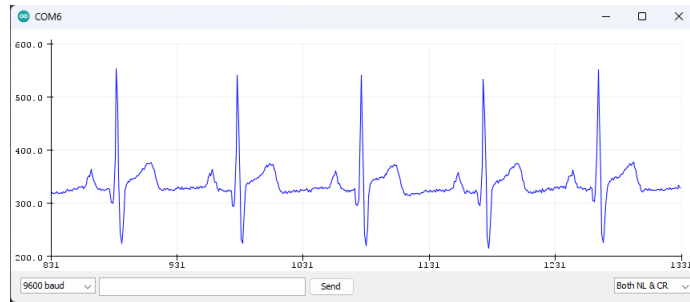
Untuk mengetahui apakah ESP32 dapat berfungsi dan terkoneksi dengan baik antara ESP32 dengan software Arduino IDE, maka proses ini dilakukan dengan melakukan proses compile program dan upload program pada software Arduino IDE yang di tampilkan pada serial monitor.



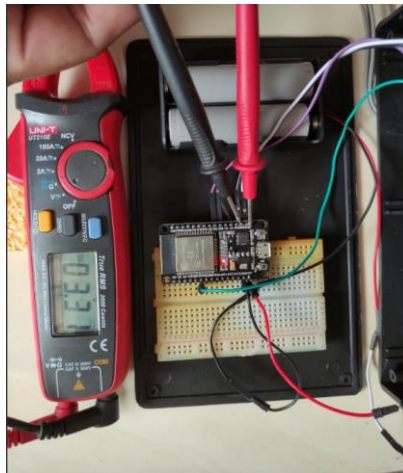
Gambar 15 Konektivitas ESP32

##### 4.5.2. Pengujian Sensor AD8232

Dilakukan pengujian pada sensor AD8232 yang dilakuakn dengan menghubungkan sensor ke mikrokontroler ESP32 yang nantinya akan ditampilkan pada serial monitor dan akan dihitung apakah mendapatkan tegangan 3.3V. Terhubung dengan kabel sensor ground sedangkan kabel multitester positif dengan pin analog atau output digital sensor.



Gambar 16 Pengujian Sensor AD8232

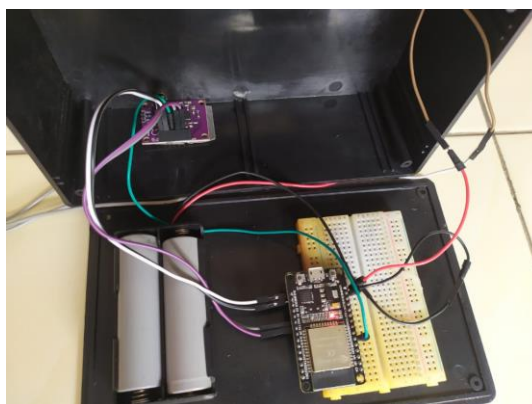


Gambar 17 Pengujian 3.3V Sensor

Dari hasil pengujian tersebut diketahui output dari beberapa sensor AD8232 yang hubungkan dengan serial monitor sesuai dengan fungsi sensor dan komponen. Tegangan input sebesar 5V menghasilkan tegangan output yang beragam nilainya, ini sudah sesuai dengan kebutuhan setiap sensor dan komponen.

#### 4.6. Implementasi Elektrik

Pada tahap ini adalah tahap dari desain elektrik yang telah dibuat sebelumnya untuk gambaran dari implementasi elektrik dapat dilihat pada gambar dibawah:

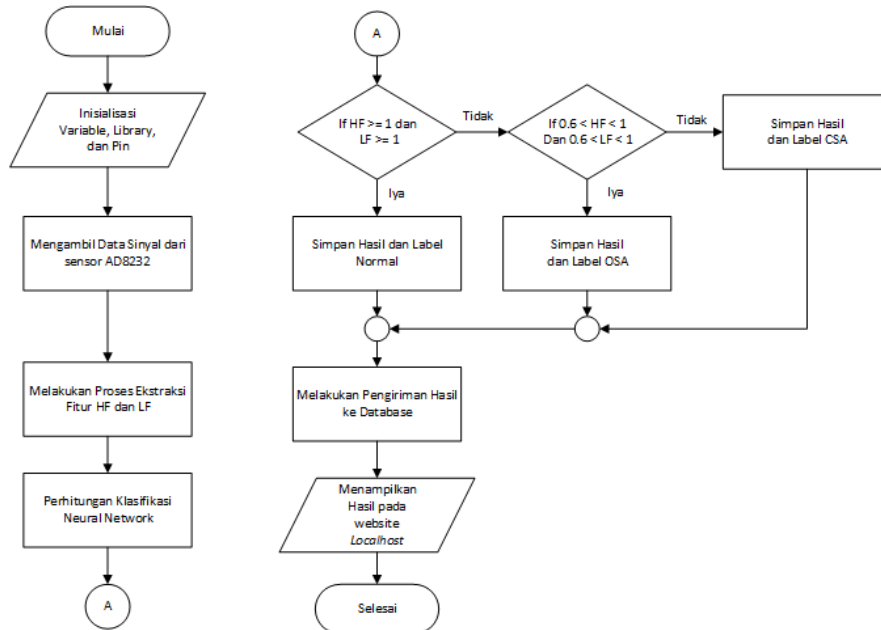


Gambar 18 Implementasi Elektrik

Pada Gambar 18 terlihat Implementasi perkabelan yang digunakan dimana untuk pendistribusian power menggunakan *project board* yang kemudian didistribusikan kedalam sensor AD8232 dan ESP32 sesuai dengan perancangan.

#### 4.7. Desain Software

Untuk desain perangkat lunak Deteksi gangguan tidur menggunakan metode jaringan syaraf tiruan berbasis *Internet of Things* dapat dilihat pada diagram alir berikut:

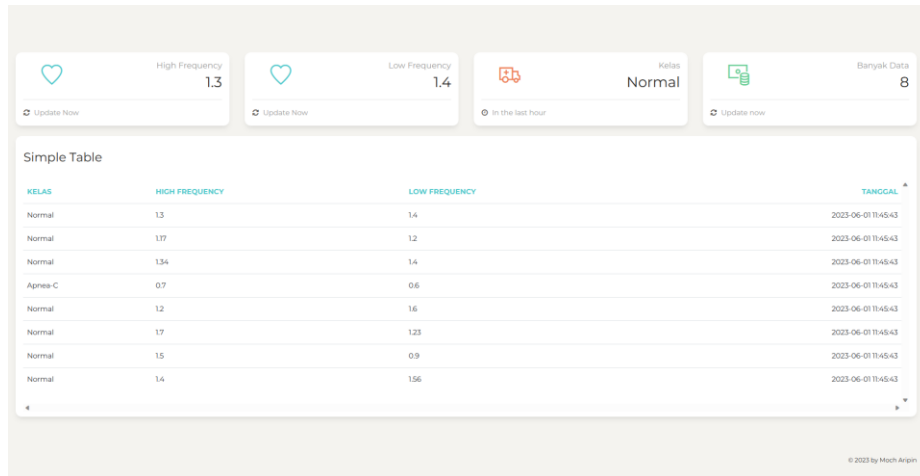


Gambar 19 Flowchart Sistem

Pada Gambar 19 berisi poin-poin penting dari sebuah program secara keseluruhan yang dijelaskan secara umum. Proses dimulai dengan inisialisasi variable, library dan pin yang dibutuhkan dalam sistem nantinya akan masuk de proses akuisisi sinyal jantung. Pada proses akuisisi sinyal jantung akan mengambil data aktivitas sinyal listrik pada jantung, akuisisi sinyal ini sangat penting dikarenakan akan digunakan sebagai input untuk klasifikasi nantinya. Setelah proses akuisisi sinyal akan dilakukan proses ekstraksi fitur *High Frequency* dan *Low Frequency* yang nantinya akan merubah domain waktu menjadi domain frekuensi. Setelah didapatkan hasil ekstraksinya akan dilakukan proses klasifikasi menggunakan metode Jaringan Syaraf Tiruan yang nilai bobotnya didapatkan dari hasil *training* dari matlab dari hasil tersebut dilakukan proses pengiriman ke dalam *database*. Proses pengiriman *database* dilakukan untuk menyimpan hasil dari ekstraksi dan hasil klasifikasi sehingga datanya nantinya akan bisa diperiksa belih lanjut oleh dokter spesialis. Dan terakhir datanya akan ditampilkan pada website dengan menggunakan *framework* Laravel.

#### 4.8. Implementasi Software

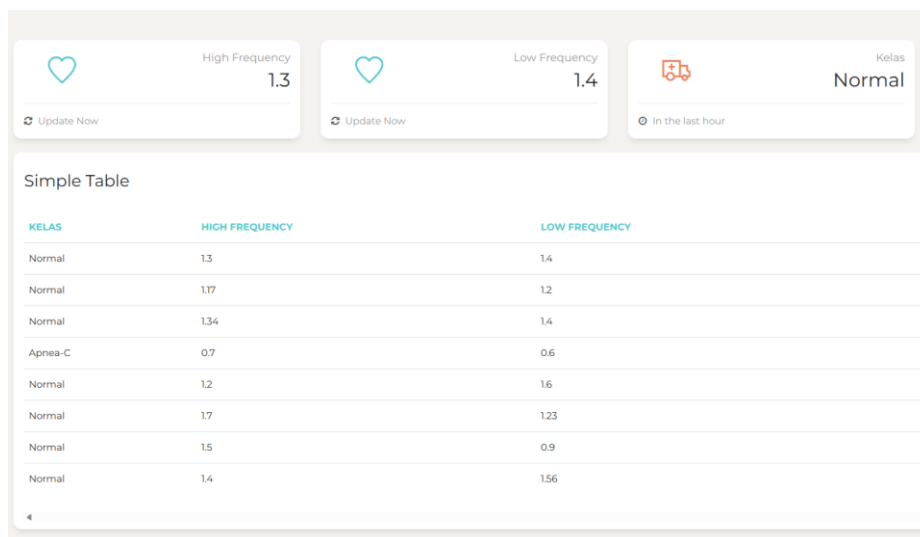
Pada tahap implementasi software ini adalah gambaran dari desain software yang telah dibuat sebelumnya.



Gambar 20 Implementasi Software

#### 4.9. Uji Software

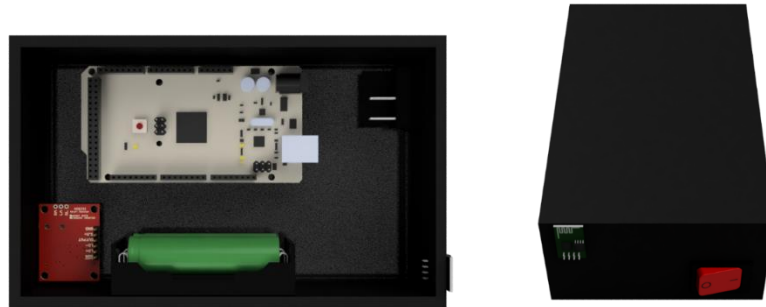
Uji software dilakukan dengan mengamati nilai hasil deteksi pada alat. Apakah software dapat berfungsi seperti rancangan sistem yang sudah dibuat sebelumnya atau tidak. Untuk gambar uji software dapat dilihat pada gambar berikut.



Gambar 21 Uji Software

#### 4.10. Desain Mekanik

Berikut adalah desain sistem mekanik seperti pada gambar 18:



Gambar 22 Desain Mekanik Sistem

Desain sistem mekanik di atas dibuat menggunakan *software* SolidWorks Untuk desain sistem mekanik tampak sebuah satu perangkat konveyor untu di terapkan beberapa sensor untuk menunjang sistem alat tersebut.

#### 4.11. Implementasi Mekanik

Pada tahap implementasi mekanik ini adalah tahap pembuatan alat dari gambar desain mekanik yang sudah dibuat sebelumnya kemudian setelah pengimplementasian selesai dilakukan integrasi atau perakitan.

#### 4.12. Integrasi

Dalam proses integrasi ini dilakukan berdasarkan dari proses desain mekanik, desain kelistrikan maupun desain perangkat lunak sehingga akan menjadi satu keseluruhan dari suatu alat. Berikut ini merupakan integrasi sistem yang berjalan dapat dilihat pada gambar dibawah ini.



Gambar 23 Integrasi

## **BAB V**

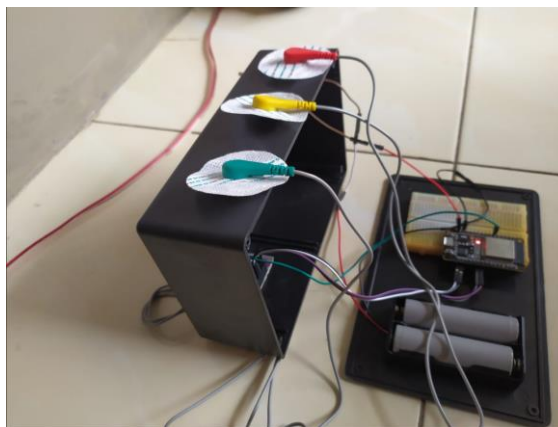
### **HASIL DAN PEMBAHASAN**

#### **5.1. Hasil Penelitian**

Pada tahap sebelumnya telah dijelaskan proses perancangan dan implementasi penelitian yang telah dilakukan. Hasil penelitian ini penulis menyelesaikan beberapa hal yang menjadi acuan referensi untuk mendapatkan hasil yang maksimal dengan desain model seefisien mungkin. Sistem ini menggunakan sensor AD8232 untuk melakukan akuisis sinyal yang nantinya akan diolah menggunakan metode Jaringan Syaraf Tiruan yang nantinya akan dikirimkan ke website dan dapat dilakukan monitoring untuk memantau pasiennya. Dimana peletakkan alatnya ditaruh didalam box supaya alatnya tetap bisa portable dan bisa dibawa kemana-mana. Berikut tampak alat tersebut.



Gambar 24 Tampilan Tampak Atas Alat



Gambar 25 Tampilan Tampak Dalam Alat

## 5.2. Test Fungsional Keseluruhan Sistem ( *Overall Testing* )

Tahapan ini dilakukan pengujian seluruh fungsi sistem, mulai dari pengujian hardware program, user interface website dan dashboard pada website. Jika ada sistem yang tidak bisa berfungsi sebagaimana mestinya maka akan dilakukan proses implementasi mekanik pada sistem. Uji keseluruhan meliputi uji fungsional, dan uji validasi.

### 5.2.1. Pengujian Struktural

Pada tahapan ini dilakukan pengujian yang bertujuan untuk mengetahui apakah jalur sirkuit sudah terhubung dengan benar sehingga sistem dapat berfungsi dengan baik. Pengujian ini dilakukan dengan mencoba semua jalur-jalur sirkuit dengan menggunakan multimeter. Berikut tabel hasil pengujian struktural sistem terdapat pada Tabel 6 berikut.

Tabel 6 Pengujian Struktural Sistem

No	Komponen Sistem	Terhubung Pada Pin ESP32	Keterangan
1	Sensor AD8232	Pin D23, pin D22	Terhubung
2	Battery	Pin Vin dan GND	Terhubung

### 5.2.2. Pengujian Fungsional

Dilakukan pengujian fungsional untuk mengetahui apakah setiap komponen dapat berkerja dengan baik. Beberapa komponen yang diuji yaitu power supply, Microcontroller ESP32. Pada proses pengujian ini hanya dua komponen tersebut yang dilakukan pengujian dikarenakan power supply dan ESP32 menunjang sistem kelistrikan pada output tegangan masing-masing komponen. Pengujian dilakukan dengan menggunakan multimeter dan program alat. Pada pengujian power supply dilakukan dengan cara mengukur voltase output yaitu 12V dan 5V.

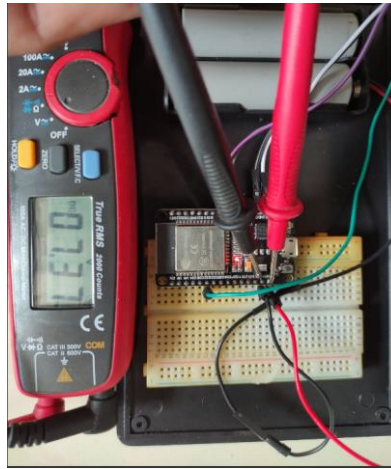
Tabel 7 Pengujian Fungsional Power Supply

No	Suplai Voltase	Pengukuran	Kondisi
1	5v	4.89 V	Baik
2	12v	12.15 V	Baik

Pengujian pada ESP32 dilakuan untuk mengetahui apakah pin dapat difungsikan dengan baik, dengan cara memasukan program terlebih dahulu dan perintahkan setiap pin untuk menghasilkan output HIGH atau LOW. Jika pada kondisi HIGH maka pin seharusnya menghasilkan tegangan sebesar 5V sedangkan



saat LOW pin akan menghasilkan 0V. Pengujian fungsional ESP32 dapat dilihat pada tabel berikut.



Gambar 26 Pengujian ESP32

Tabel 8 Pengujian Fungsional ESP32

No	Pin No	Pengukuran Saat HIGH	Pengukuran Saat LOW	Kondisi
1	16	3.33 v	0.01 V	Baik
2	17	3.24 v	0.00 V	Baik
3	33	3.31 v	0.01 V	Baik

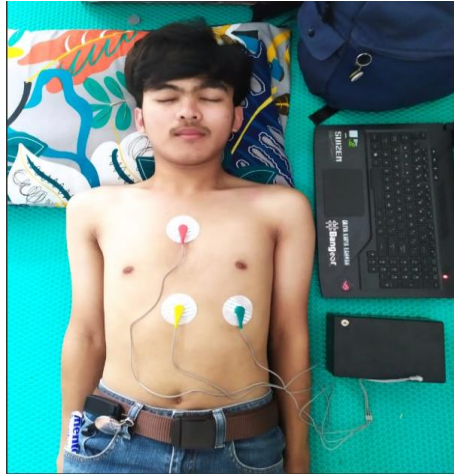
### 5.2.2.1. Pengujian Keseluruhan Sistem

Setelah beberapa rangkaian pengujian yang telah dilakukan pada setiap komponen yang ada, maka tahap selanjutnya akan dilakukan pengujian keseluruhan pada sistem yang dibuat. Pada tahap ini akan dilakukan pengujian sesuai dengan diagram blok yang sudah diterapkan sebelumnya, mulai dari pembacaan sensor AD8232 kemudian akan dilakukan perhitungan *Fast Fourier Transform* (FFT) dan akan dilakukan perhitungan Jaringan Syafar Tiruan (JST) yang nantinya akan menghasilkan kelasnya.

1. Pengujian pertama pasien harus dipasang elektroda sesuai dengan rancangan awal penempatan elektrodanya. Dapat dilihat bahwa elektroda ditetapkan sesuai dengan table berikut.

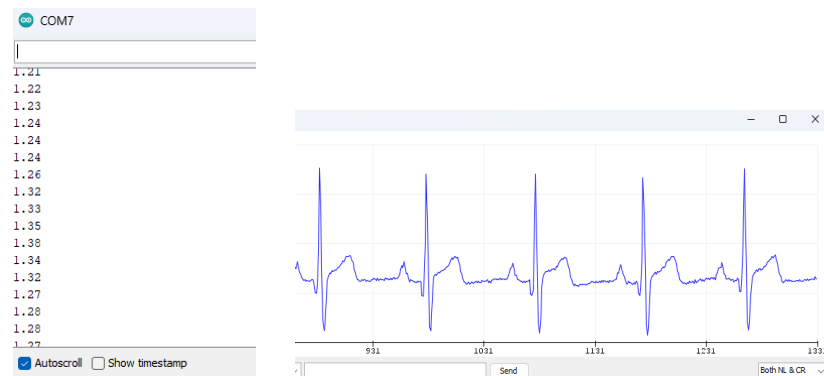
Tabel 9 Penempatan Elektroda

Warna	Peletakkan	Fungsi
Merah (RA)	Dada tengah atas	Untuk mengetahui aktivitas listrik di jantung.
Kuning (LA)	Dada tengah bawah (tulang rusuk terakhir)	Untuk menentukan arah dan magnitude sinyal yang melewati jantung.
Hijau (LL)	Dada kanan (bawah puting)	Untuk melihat interpretasi keseluruhan ECG.



Gambar 27 Contoh Penempatan Elektroda

- Setelah dipastikan elektroda terpasang, dilakukan dengan melakukan pembacaan sinyal dengan sensor AD8232 yang nantinya akan dilakukan proses perhitungan FFT pada ESP32. Berikut contoh sinyal yang dihasilkan pada sensor AD8232.



Gambar 28 Pembacaan Sinyal Sensor AD8232

- Selanjutnya Ketika data didapatkan dari pembacaan sensor AD8232 akan dilakukan perhitungan FFT yang nantinya akan didapatkan nilai *High Frequency* (HF) dan *Low Frequency* (LF). Perhitungan dan code FFT pada ESP 32 dapat dilihat dibawah ini.

Tabel 10 Data pembacaan Sensor AD8232 antara 2 puncak R-Peaks

No	Data
1	1,12
2	1,26
3	1,77
4	1,76
5	0,96
	.....
76	1,91

77	1,25
78	1,05
79	1,20

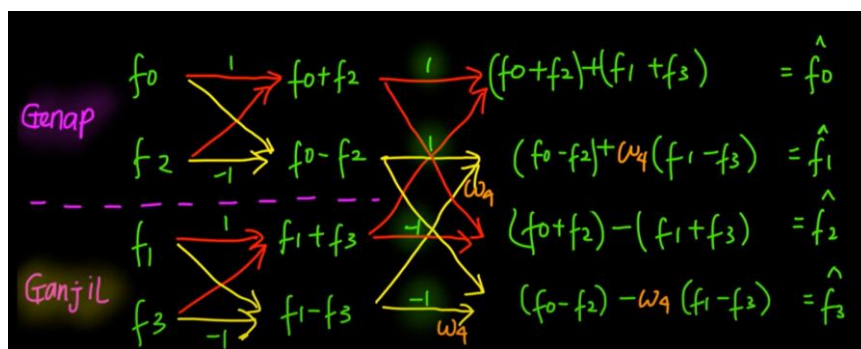
Pada tabel 10 merupakan data yang didapatkan pada pembacaan sinyal yang nantinya akan dilakukan proses FFT untuk didapatkan nilai LF dan HF-nya. Nilai sinyal itu didapatkan dari nilai antara 2 puncak *R-peak* pada sistem. Untuk perhitungan pada ESP32 menggunakan library langsung dari Arduino yaitu EasyFFT. Dimana sudah disediakan function tinggal memasukan data sesuai dengan function yang disediakan. Library dapat dibaca dan dilihat pada link berikut :

<https://projecthub.arduino.cc/abhilashpatel121/easyfft-fast-fourier-transform-fft-for-arduino-03724d> untuk melihat code jelasnya dapat dilihat pada lampiran 1.



Gambar 29 EasyFFT

Untuk perhitungan manual FFT dapat melihat gimana proses sinyal dijadikan nilai frequency pada gambar berikut untuk mempermudah dalam menghitung FFT nantinya.



Gambar 30 Perhitungan FFT

Pada gambar terlihat bahwa nilai akan dibagi menjadi 2 yaitu untuk ganjil dan genap yang nantinya akan dilakukan cross perhitungan dengan menggunakan  $W$  dimana nilai  $W$  memiliki rumus.

Jadi perhitungan manual FFT disajikan dalam Tabel Berikut.

Tabel 11 Perhitungan FFT

Data Sinyal	FFT	Nilai
1,12	$(f_0+f_2)+(f_1+f_3)$	1,24
1,26	$(f_0-f_2)+w_4(f_1-f_3)$	1,25
1,77	$(f_0+f_2)-(f_1+f_3)$	3,74
1,76	$(f_0-f_2)-w_4(f_1-f_3)$	3,73
0,96	$(f_2+f_4)+(f_3+f_5)$	2,49
...		
1,91	$(f_{74}-f_{76})+w_4(f_{73}-f_{75})$	0,8
1,25	$(f_{74}+f_{76})-(f_{73}+f_{75})$	0,78
1,05	$(f_{74}-f_{76})-w_4(f_{73}-f_{75})$	1,23
1,20	$(f_{76}+f_{78})+(f_{75}+f_{77})$	1,11

Kemudian akan dihitung nilai frequencynya dimana pada system ini menggunakan frequency 1,28 sesuai dengan frequency pada saat melakukan ekstrak menggunakan matlab. frequencynya didapatkan dengan rumus sebagai berikut.

$$i = \frac{i \times \text{frequency}}{N}$$

Kemudian dari nilai frequency tersebut akan lakukan rata-rata berdasarkan frequencynya dimana untuk nilai LF yaitu antara frequency 0,04 – 0,15 dan HF 0,15 – 0,4. Sehingga dapat diambil nilai sebagai berikut:

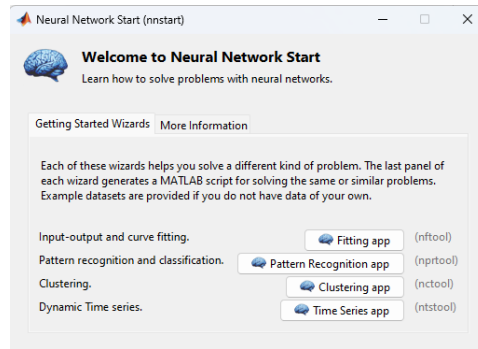
Tabel 12 Nilai LF dan HF

LF	[1,24 1,25 3,74 3,73 2,49]	2,49
HF	[0,9 0,67 2,041 2,669 ]	1,57

$$X = \frac{\sum f_i}{\sum f_i}$$

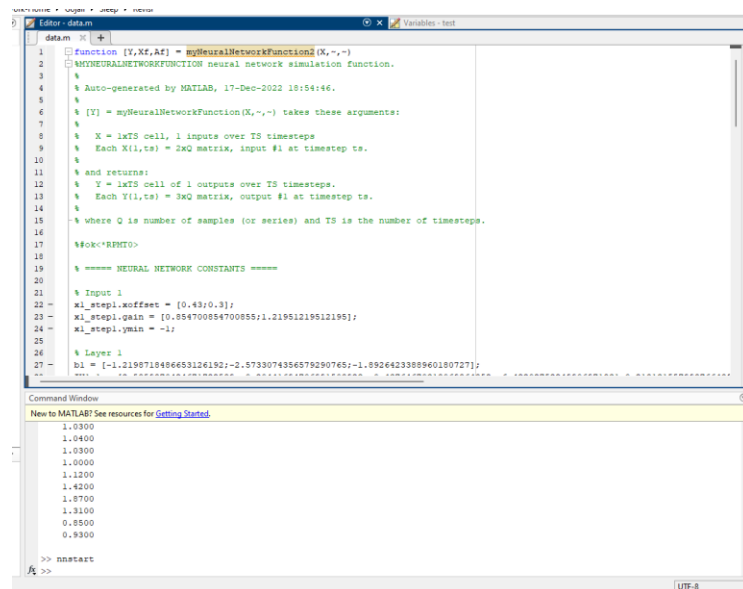
Dari data nilai frequency tabel diatas akan dilakukan rata-rata menggunakan rumus rata-rata diatas dan akan didapatkan nilai LF sebesar 2,49 dan HF sebesar 1,57.

- Setelah dapatkan nilai HF dan LFnya akan dilakukan perhitungan JST pada ESP32 yang nilai bobot trainingnya didapatkan pada proses training menggunakan software matlab dengan tools yang Namanya nnstart.



Gambar 31 nnstart Matlab

Dimana nanti didapatkan nilai bobot dan beberapa nilai yang nantinya akan dimasukkan kedalam ESP32 untuk dilakukan perhitungan nantinya. Berikut function yang dihasilkan dalam proses nnstart pada software matlab.



Gambar 32 Function Output NNstart

Tabel 13 Nilai Normalisasi Input

Xoffset	0.43; 0.3
Gain	0.854700854700855; 1.21951219512195
Ymin	-1; -1

Tabel diatas merupakan nilai normalisasi yang didapatkan dari proses training pada matlab yang digunakan untuk menormalisasi input yang masuk kedalam proses pada ESP32 nantinya. Dimana xoffset akan mengurangkan nilai 2 input. Selanjutnya akan dikali dengan nilai gain dan akan dikurang pada nilai ymin. Setelah dilakukan proses normalisasi pada inputnya akan dilakukan proses perhitungan bobot berdasarkan training pada matlab. Dimana akan dilakukan

perhitungan pada *hidden layer* untuk didapatkan hasil masing masing berat pada nodenya. Nilai bobot *hidden layer* dapat dilihat pada tabel dibawah.

Tabel 14 Bobot dan Bias *Hidden Layer*

Node 1	2.5859870424671722589; -0.20441694706991508923
Node 2	-0.43764670318069864852; -6.4220275224990697183
Node 3	0.21813155799876648988; -8.2769669728104435791
Bias <i>Hidden Layer</i>	-1.2198718486653126192; -2.5733074356579290765; -1.8926423388960180727

Pada tabel 14 merupakan bobot hasil training pada matlab yang akan digunakan sebagai perhitungan pada ESP32 dimana nanti akan dilakukan perkalian dari nilai normalisasi input dan akan di tambah dengan nilai bias untuk mendapatkan hasilnya. Setelah dilakukan perhitungan nilai bobot pada setiap node pada *hidden layer* akan dimasukkan kedalam perhitungan nilai bobot output layer untuk diketahui nilai probabilitas dari setiap kelas yang nantinya dikeluarkan sebagai output. Nilai bobot output layer dapat dilihat pada tabel dibawah.

Tabel 15 Bobot dan Bias *Output Layer*

Node 1	1.54809825246362; -17.9704244951953; -5.27619712455950
Node 2	1.09308242834741; 18.8669573710212; 1.15381792557886
Node 3	-3.16842678459342; -0.243377681700155; 4.18028398115929
Bias <i>Output Layer</i>	-0.889216745884705; -2.13672130940209; 3.16497735285852

Pada tabel 15 akan dilakukan proses perhitungan nilai berat yang akan dihasilkan yang nantinya akan dihitung nilai probabilitas dari setiap kelas yang digunakan yaitu kelas Normal, OSA dan CSA. Yang terbesar merupakan kelas tersebut.

5. Setelah didapatkan nilai bobot pada matlab akan dilakukan proses perhitungan pada ESP32 dimana nilai yang didapat HF dan LF akan dilakukan perhitungan secara langsung dan akan menghasilkan langsung kelasnya. Proses perhitungan dan code programnya pada lampiran 2.
6. Setelah melakukan define nilai bobotnya akan dilakukan define function JST untuk menghitung nilai JST nya. Code function untuk menghitung JST pada ESP32 dapat dilihat pada lampiran 3.
7. Dapat dilihat bahwa yang dilakukan pertama kali yaitu melakukan normalisasi pada input nilai nantinya yang kemudian akan dilakukan perhitungan berdasarkan nilai bobotnya sesuai dengan table yang didapatkan pada matlabnya. Secara manual perhitungannya dapat dilihat pada table perhitungan manual berikut.

Data LF dan HF	
LF	HF
2,49	1,57

	Nilai	Rumus	Hasil
LF	2,49	Nilai - xofset	$2,49 - 0,43 = 2,06$
HF	1,57	Nilai - xofset	$1,57 - 0,3 = 1,27$

	Nilai	Rumus	Hasil
LF	2,06	Nilai x gain	$2,06 \times 0,86 = 1,78$
HF	1,27	Nilai x gain	$1,27 \times 1,22 = 1,55$

	Nilai	Rumus	Hasil
LF	1,78	Nilai + ymin	$1,78 + (-1) = 0,78$
HF	1,55	Nilai + ymin	$1,55 + (-1) = 0,55$

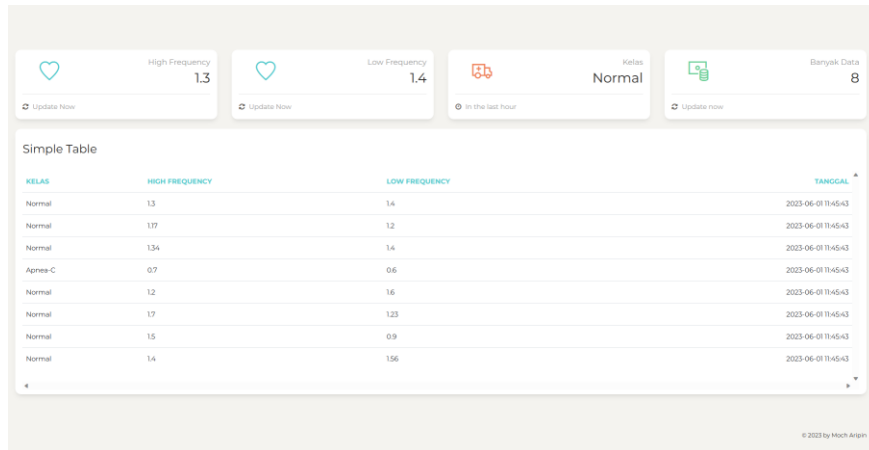
	Rumus Hidden	tansig_apply	Hasil
Normal	$H = (W*LF) + (W*HF) + Bias$	$2.0 / (1.0 + \exp(-2.0 * H)) - 1.0$	-0,49
CSA	$H = (W*LF) + (W*HF) + Bias$	$2.0 / (1.0 + \exp(-2.0 * H)) - 1.0$	-1
OSA	$H = (W*LF) + (W*HF) + Bias$	$2.0 / (1.0 + \exp(-2.0 * H)) - 1.0$	-1

	Rumus Output	Hasil
Normal	$H = (W*Normal) + (W*CSA) + (W*OSA) + Bias$	9,27
CSA	$H = (W*Normal) + (W*CSA) + (W*OSA) + Bias$	-31,1
OSA	$H = (W*Normal) + (W*CSA) + (W*OSA) + Bias$	6,03

	Rumus Exp	Probabilitas	Hasil
Normal	Num = EXP(Normal)	$P = Num / SUM(Num)$	0,95
CSA	Num = EXP(CSA)	$P = Num / SUM(Num)$	0,02
OSA	Num = EXP(OSA)	$P = Num / SUM(Num)$	0,03

Jadi berdasarkan perhitungan manual JST didapatkan hasil dimana nilai probabilitas Normal lebih tinggi sehingga termasuk kelas Normal.

- Selanjutnya akan dikirimkan kedalam website system monitoring agar dapat dipantau dengan baik oleh pasien. Dimana pengirimannya menggunakan WiFi modul pada ESP32. Berikut tampilan system monitoringnya.



Gambar 33 Halaman Dashboard Alat

### 5.2.3. Uji Coba Validasi

Tahap ini bertujuan untuk menguji kinerja dari seluruh sensor yang akan digunakan nantinya apakah sensor tersebut dapat berfungsi dengan baik atau tidak. Sensor diuji dengan membandingkan antar nilai pembacaan dari sensor dengan alat ukur yang sebenarnya. Pengujian ini dilakukan dengan cara menguji dari nilai kemungkinan kesalahan yang dapat terjadi pada komponen-komponen yang diimplementasikan model penelitian ini.

#### 5.2.3.1. Uji Coba Validasi Sensor AD8232

Pengujian pembacaan sensor AD8232 bertujuan untuk mengetahui tingkat error dari pembacaan sensor AD8232. Pengujian dilakukan dengan membandingkan jumlah denyut jantung atau *Bit Per Minute* (BPM) melalui pembacaan sensor dan manual selama 10 detik kemudian dikali dengan 6. BPM dari pembacaan sensor akan dibandingkan dengan BPM perhitungan manual. Pengujian dilakukan pada 10 subjek berbeda dengan waktu 10 detik per subjeknya.

Tabel 16 Uji Coba Validasi Sensor AD8232

Subject	BPM Manual	BPM Sensor AD8232	Error (%)
1	72	72	0%
2	72	72	0%
3	84	78	7.14%
4	72	72	0
5	72	66	8.33%
6	78	78	0
7	78	72	7.69%
8	84	84	0
9	78	78	0



10	72	78	8.33%
Rata-rata <i>error</i>			3.15%

Berdasarkan perbandingan pada Tabel 16, terdapat perbedaan antara pengukuran manual dan menggunakan sensor AD8232. Oleh karena itu, persentase kesalahan dapat dihitung dengan menggunakan rumus pada Persamaan dibawah ini.

$$Presentase\ error = \frac{|BPM\ Manual - BPM\ Sensor|}{BPM\ Manual} \times 100\%$$

Setelah didapatkan presentase *error* dari pengujian sensor akan dilakukan perhitungan rata-rata presentase *error* dengan menggunakan rumus pada persamaan dibawah ini.

$$Rata - rata\ presentase\ error = \frac{Total\ error\ (\%)}{Jumlah\ pengujian}$$

$$Rata - rata\ presentase\ error = \frac{31.51\%}{10}$$

$$Rata - rata\ presentase\ error = 3.15\%$$

Setelah didapatkan nilai rata-rata *error* akan dilakukan perhitungan tingkat akurasi sensor AD8232 dengan menggunakan rumus pada Persamaan dibawah ini.

$$Akurasi\ sensor\ AD8232 = 100\% - error$$

$$Akurasi\ sensor\ AD8232 = 100\% - 3.15\%$$

$$Akurasi\ sensor\ AD8232 = 96.85\%$$

Hasil analisis dari pengujian pembacaan sensor AD8232 memiliki akurasi sebesar 96.85%. Adapun kendala saat pembacaan yang membuat nilai akurasinya berkurang karena sensor AD8232 memiliki sensitivitas terhadap power terutama saat laptop melakukan pengisian daya. Oleh sebab itu sangat disarankan untuk melakukan pembacaan dengan tidak melakukan pengisian daya laptop

### 5.2.3.2. Uji Coba Data Latih

Pengujian data latih adalah pengujian untuk melihat kemampuan suatu algoritma yang dipakai dengan model yang dihasilkan apakah memiliki tingkat yang efektif atau sebaliknya. Pengujian ini menggunakan dataset yang didapatkan dari physionet. Dimana dari dataset ini juga akan dilakukan pengujian untuk melihat tingkat error dari system jika dikasih dataset untuk menentukan model dari JST-nya.

Tabel 17 Klasifikasi Jaringan Syaraf Tiruan

<b>No.</b>	<b>LF</b>	<b>HF</b>	<b>Hasil Klasifikasi</b>	<b>Kelas Aktual</b>
1	0.90	0.97	Normal	Normal
2	0.77	0.92	Normal	Normal
3	0.82	0.88	Normal	Normal
4	0.81	0.88	Normal	Normal
5	0.93	0.88	Normal	Normal
6	1.01	1.03	Normal	Normal
7	1.02	1.03	Normal	Normal
8	0.91	1.03	Normal	Normal
9	0.75	1.03	Normal	Normal
10	0.79	1.03	Normal	Normal
11	0.89	0.97	Normal	Normal
12	1.05	0.97	Normal	Normal
13	0.92	0.97	Normal	Normal
14	0.92	1.00	Normal	Normal
15	0.89	1.02	Normal	Normal
16	0.89	1.04	Normal	Normal
17	0.90	1.04	Normal	Normal
18	0.87	1.04	Normal	Normal
19	0.96	1.03	Normal	Normal
20	0.89	1.02	Normal	Normal
21	0.85	0.90	OSA	OSA
22	0.96	0.90	OSA	OSA
23	1.00	0.90	CSA	OSA
24	0.97	0.89	OSA	OSA
25	1.01	0.89	OSA	OSA
26	1.22	0.68	OSA	CSA
27	1.12	0.68	CSA	CSA
28	1.01	0.67	CSA	CSA
29	1.09	0.67	CSA	CSA
30	1.15	0.67	CSA	CSA

Dari hasil pengujian sebanyak 30 kali menggunakan dataset untuk pengujian kelas data akan dihitung nilai error apakah daset tersebut valid atau benar sesuai dengan kelas sebenarnya atau tidak. Dimana dari tabel tersebut sebanyak 28 data dinyatakan valid dan 2 tidak dikarenakan nilai OSA dan CSA yang memiliki nilai cukup mendekati.

$$\text{Presentase error} = \frac{|\text{Nilai valid} - \text{banyak data}|}{\text{banyak data}} \times 100\%$$

$$\text{Presentase error} = \frac{|28 - 30|}{30} \times 100\%$$

$$\text{Presentase error} = 6,67\%$$

Jadi tingkat ketidakakuratan model dalam prediksi dan klasifikasi dataset *physionet* ditunjukkan oleh presentase error 6,67%. Presentase error menunjukkan seberapa dekat atau jauh hasil prediksi model dengan nilai sebenarnya dalam dataset. Untuk menentukan seberapa besar tingkat kesalahan yang dapat diterima atau diperlukan untuk ditingkatkan melalui pengoptimalan model.

### 5.2.3.3. Uji Coba Data Uji

Pengujian Data Uji adalah pengujian untuk memastikan system yang digunakan berfungsi sesuai yang diinginkan dengan menggunakan algoritma yang dipilih atau tidak. Pengujian secara langsung dilakukan dengan mengisi kuisisioner *Stop-Bang* dan *Epworth Sleepiness Scale* untuk mengetahui yang diuji benar pasien normal atau mengalami gejala gangguan tidur. Pengujian dilakukan untuk mengetahui nilai dari akurasi serta peforma dari Jaringan Syaraf Tiruan, pengujian dengan menggunakan 20 subjek. Hasil pengujian metode klasifikasi dapat dilihat pada tabel dibawah ini.

Tabel 18 Klasifikasi Jaringan Syaraf Tiruan

No.	LF	HF	Hasil Klasifikasi	Kelas Aktual
1	2.36	0.83	NORMAL	Normal
2	1.92	1.22	NORMAL	Normal
3	1.72	1.29	NORMAL	Normal
4	2.49	1.57	NORMAL	Normal
5	1.98	1.28	NORMAL	Normal
6	1.05	0.9	OSA	Normal
7	2.02	1.36	NORMAL	Normal
8	1.38	1.12	NORMAL	Normal
9	2.38	1.9	NORMAL	Normal
10	1.78	1.24	NORMAL	Normal

11	1.45	0.98	OSA	OSA
12	1.34	1.11	OSA	OSA
13	1.23	1.23	OSA	OSA
14	1.45	1.05	NORMAL	OSA
15	1.18	1.26	OSA	OSA
16	1.48	1.21	OSA	OSA
17	1.23	1.1	NORMAL	OSA
18	1.36	1.4	OSA	OSA
19	1.76	1.12	OSA	OSA
20	1.19	1.03	NORMAL	OSA

Setelah dilakukan pengujian system dengan data asli akan dilakukan perhitungan akurasi klasifikasi dengan menggunakan rumus dibawah ini.

$$Akurasi = \frac{Total\ data\ benar}{Jumlah\ total\ pengujian} \times 100\%$$

$$Akurasi = \frac{16}{20} \times 100\%$$

$$Akurasi = 0.8 \times 100\% = 80\%$$

Jadi hasil akurasi yang didapatkan dalam pengujian ini sebesar 80% ini menunjukkan system ini sangat efektif untuk mendeksi gangguan tidur. Kesalahan pembacaan pada system biasanya disebabkan oleh sensor yang rentan terkena gangguan magnetic atau benda yang memiliki sifat logam sehingga terkadang mengganggu saat pembacaan.

#### 5.2.3.4. Uji Coba Waktu Komputasi

Pengujian waktu komputasi klasifikasi merupakan pengujian untuk mengetahui waktu melakukan klasifikasi dengan metode Jaringan Syaraf Tiruan. pada pengujian ini akan dilakukan saat proses klasifikasi Jaringan Syaraf Tiruan dan dilihat waktu saat mekalukan klasifikasi sampai keluar hasilnya:

Tabel 19 Uji Coba Validasi Kecepatan Akses Halaman Dashboard

Data Uji ke -	Waktu Komputasi (ms)
1	6 ms
2	7 ms
3	8 ms
4	10 ms
5	6 ms

6	7 ms
7	7 ms
8	8 ms
9	9 ms
10	8 ms

Dari hasil pengujian yang dilakukan diatas akan dilakukan rata-rata waktu komputasi pada 10 subjek untuk didapatkan hasilnya. Perhitungan rata-rata waktu komputasi menggunakan rumus pada dibawah ini.

$$\text{Rata - Rata waktu komputasi} = \frac{\text{Total nilai waktu}}{\text{Jumlah subjek}}$$

$$\text{Rata - Rata waktu komputasi} = \frac{\sum 6+7+8+10+6+7+7+8+9+8}{10}$$

$$\text{Rata - Rata waktu komputasi} = \frac{76}{10} = 7,6 \text{ ms}$$

Jadi hasil dari analisis pada pengujian waktu komputasi Klasifikasi Jaringan Syaraf Tiruan didapatkan hasil rata-rata 7,6 ms. Nilai kecepatan ini dipengaruhi oleh nilai fitur yang keluar jika fitur bernilai lebih tinggi akan menghasilkan komputasi yang cukup tinggi sekisar 10 ms. Hal ini dikarenakan ESP32 membutuhkan waktu untuk menghitung nilai bobot yang ada.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1. Kesimpulan**

Kesimpulan pada penelitian ini berdasarkan dengan pengujian dan analisis yang telah dilakukan dan dapat ditarik kesimpulan:

1. Sensor ECG AD8232 pada ESP32 berhasil mendeteksi aktivitas listrik dan memperoleh data sinyal jantung secara akurat. Tingkat akurasi yang diperoleh dari pengujian kinerja sensor AD8232 dalam mendeteksi aktivitas kelistrikan jantung sebesar 96,85%.
2. Tingkat akurasi klasifikasi menggunakan Jaringan Syaraf Tiruan sebesar 80% berdasarkan proses klasifikasi menggunakan Jaringan Syaraf Tiruan pada 20 data uji yang dihitung berdasarkan jumlah kelas Normal, OSA dan CSA tingkat akurasi ini dinilai cukup baik.
3. Waktu komputasi rata-rata sistem monitoring pendeteksi *Sleep-disordered Breathing* menggunakan metode Jaringan Syaraf Tiruan adalah 7,6 ms, berdasarkan rentang waktu komputasi 6 ms hingga 10 ms yang diperoleh setelah pengujian.

#### **6.2. Saran**

Berdasarkan kesimpulan yang didapatkan pada penelitian ini, ada beberapa saran yang bisa diberikan untuk penelitian-penelitian selanjutnya, yaitu:

1. Penggunaan sensor yang semakin canggih memungkinkan elektroda ditempatkan pada bagian tubuh manusia yang lebih nyaman digunakan, sehingga letaknya tidak mengganggu pengambilan data atau menyebabkan redaman saat pengguna bergerak.
2. Menambahkan data primer saat melakukan proses training data dan membuat model. Hal ini bertujuan untuk menambah nilai akurasi yang dihasilkan sehingga datanya lebih bagus.

## DAFTAR PUSTAKA

- Ananda, I.D., Kurniawan, R., Yanti, N. and Ihsani, F.,** 2021. Sistem Pakar untuk Mendiagnosis Insomnia Menggunakan Metode Dempster Shafer. *JIMP: Jurnal Informatika Merdeka Pasuruan*, 6(3), pp.1–8.
- Culebras, A. and Anwar, S.,** 2018. *Sleep Apnea Is a Risk Factor for Stroke and Vascular Dementia. Current Neurology and Neuroscience Reports*, <https://doi.org/10.1007/s11910-018-0855-1>.
- Fahruzi, I., Purnama, K.E., Purnomo, M.H., Kunci, K., Ecg, R., Tidur, G. and Svm, M.P.,** 2020. *Asesmen ECG-Apnea Satu Sadapan untuk Peningkatan Akurasi Klasifikasi Gangguan Tidur Berdasarkan AdaBoost (Single Lead ECG-Apnea Recordings Assessment for Improved Accuracy in Classification of Sleep Disorder based on AdaBoost). Jurnal Nasional Teknik Elektro dan Teknologi Informasi |*, .
- Felix, O., Amaddeo, A., Olmo Arroyo, J., Zerah, M., Puget, S., Cormier-Daire, V., Baujat, G., Pinto, G., Fernandez-Bolanos, M. and Fauroux, B.,** 2016. Central sleep apnea in children: experience at a single center. *Sleep Medicine*, 25, pp.24–28. <https://doi.org/10.1016/j.sleep.2016.07.016>.
- Kaplan, A. and Ulualp, S.O.,** 2022. **Assessment of Central Sleep Apnea Events in Children with Sleep-Disordered Breathing.** *Sleep Disorders*, 2022, pp.1–7. <https://doi.org/10.1155/2022/2590337>.
- Mandal, U.G.V.G.A.B.A.B.U.C.B.K.T.,** 2008. Optimization of Metformin HCl 500 mg Sustained Release Matrix Tablets Using Artificial Neural Network (ANN) Based on Multilayer Perceptrons (MLP) Model. *Chem. Pharm. Bull.*, 56, pp.150–155.
- Nguyen, A.-T., Nguyen, T., Le, H.-K., Pham, H.-H. and Do, C.,** 2022. A novel deep learning-based approach for sleep apnea detection using single-lead ECG signals. [online] Available at: <<http://arxiv.org/abs/2208.03408>>.
- Ratanasari, D., Widasari, E. R. & Maulana, R.,** 2022. Sistem Pendeteksi Central Sleep Apnea Menggunakan Metode Neural Network dengan Fitur RR Interval dan Durasi QRS. *J-PTIHK*, III(5), pp. 1-5.
- Retnosari, G.Y., Irwanto, I. and Herawati, L.,** 2021. Prevalence and characteristics of sleep problems of Indonesian children in 0 – 36 months old. *Jurnal Kedokteran dan Kesehatan Indonesia*, 12(1), pp.28–33. <https://doi.org/10.20885/jkki.vol12.iss1.art6>.
- Suriansyah, Mohamad Iqbal dan Prihastuti Harsani** 2018 Air Pollution Monitoring for Bogor Smart City Based Internet of Things and Social Media (Twitter) IOP Conf. Series: Materials Science and Engineering 621 (2019) 012006

**Szollosi, I., Krum, H., Kaye, D. and Naughton, M.T.,** 2007. Sleep apnea in heart failure increases heart rate variability and sympathetic dominance. *Sleep*, 30(11), pp.1509–1514. <https://doi.org/10.1093/sleep/30.11.1509>.

**Wang, Y., Plataniotis, K.N. and Hatzinakos, D.,** 2006. Integrating analytic and appearance attributes for human identification from ECG signals. In: *Biometrics Symposium, BCC 2006*. <https://doi.org/10.1109/BCC.2006.4341627>



## LAMPIRAN

### Lampiran 1 Perhitungan FFT

```
Code EasyFFT
/*
//Example data:
int data[64]={14, 30, 35, 34, 34, 40, 46, 45, 30, 4, -26, -48, -55, -49, -37,
-28, -24, -22, -13, 6, 32, 55, 65, 57, 38, 17, 1, -6, -11, -19, -34,
-51, -61, -56, -35, -7, 18, 32, 35, 34, 35, 41, 46, 43, 26, -2, -31, -50,
-55, -47, -35, -27, -24, -21, -10, 11, 37, 58, 64, 55, 34, 13, -1, -7};
*/

//-----//
byte sine_data [91]=
{
0,
4, 9, 13, 18, 22, 27, 31, 35, 40, 44,
49, 53, 57, 62, 66, 70, 75, 79, 83, 87,
91, 96, 100, 104, 108, 112, 116, 120, 124, 127,
131, 135, 139, 143, 146, 150, 153, 157, 160, 164,
167, 171, 174, 177, 180, 183, 186, 189, 192, 195, //Paste this at top of
program
198, 201, 204, 206, 209, 211, 214, 216, 219, 221,
223, 225, 227, 229, 231, 233, 235, 236, 238, 240,
241, 243, 244, 245, 246, 247, 248, 249, 250, 251,
252, 253, 253, 254, 254, 254, 255, 255, 255, 255
};
float f_peaks[5]; // top 5 frequencies peaks in descending order
//-----//

void setup()
{
  Serial.begin(250000);
}

void loop() {

/*
//example
FFT(data,64,100); //to get top five value of frequencies of X having 64
sample at 100Hz sampling
Serial.println(f_peaks[0]);
Serial.println(f_peaks[1]);
*/
}
```

```

delay(99999);
*/

/*
after ruing above FFT(), frequencies available at
f_peaks[0],f_peaks[1],f_peaks[2],f_peaks[3],f_peaks[4],
*/
}

//-----FFT Function-----//

float FFT(int in[],int N,float Frequency)
{
/*
Code to perform FFT on arduino,
setup:
paste sine_data [91] at top of program [global variable], paste FFT function at
end of program
Term:
1. in[] : Data array,
2. N : Number of sample (recommended sample size 2,4,8,16,32,64,128...)
3. Frequency: sampling frequency required as input (Hz)

If sample size is not in power of 2 it will be clipped to lower side of number.
i.e, for 150 number of samples, code will consider first 128 sample, remaining
sample will be omitted.
For Arduino nano, FFT of more than 128 sample not possible due to mamory
limitation (64 recomended)
For higher Number of sample may arise Mamory related issue,
Code by ABHILASH
Contact: abhilashpatel121@gmail.com
Documentation:https://www.instructables.com/member/abhilash\_patel/instructables/
2/3/2021: change data type of N from float to int for >=256 samples
*/

unsigned int data[13]={ 1,2,4,8,16,32,64,128,256,512,1024,2048 };
int a,c,l,f,o,x;
a=N;

for(int i=0;i<12;i++) //calculating the levels
{ if(data[i]<=a){o=i;} }

int in_ps[data[o]]={}; //input for sequencing
float out_r[data[o]]={}; //real part of transform
float out_im[data[o]]={}; //imaginory part of transform

```

```

x=0;
for(int b=0;b<o;b++)          // bit reversal
{
    c1=data[b];
    f=data[o]/(c1+c1);
    for(int j=0;j<c1;j++)
    {
        x=x+1;
        in_ps[x]=in_ps[j]+f;
    }
}

for(int i=0;i<data[o];i++)    // update input array as per bit reverse order
{
    if(in_ps[i]<a)
    {out_r[i]=in[in_ps[i]];}
    if(in_ps[i]>a)
    {out_r[i]=in[in_ps[i]-a];}
}

int i10,i11,n1;
float e,c,s,tr,ti;

for(int i=0;i<o;i++)          //fft
{
    i10=data[i];              // overall values of sine/cosine :
    i11=data[o]/data[i+1];    // loop with similar sine cosine:
    e=360/data[i+1];
    e=0-e;
    n1=0;

    for(int j=0;j<i10;j++)
    {
        c=cosine(e*j);
        s=sine(e*j);
        n1=j;

        for(int k=0;k<i11;k++)
        {
            tr=c*out_r[i10+n1]-s*out_im[i10+n1];
            ti=s*out_r[i10+n1]+c*out_im[i10+n1];

            out_r[n1+i10]=out_r[n1]-tr;
            out_r[n1]=out_r[n1]+tr;

            out_im[n1+i10]=out_im[n1]-ti;

```

```

        out_im[n1]=out_im[n1]+ti;

        n1=n1+i10+i10;
    }
}

/*
for(int i=0;i<data[o];i++)
{
Serial.print(out_r[i]);
Serial.print("\ "); // un comment to print RAW o/p
Serial.print(out_im[i]); Serial.println("i");
}
*/

//---> here onward out_r contains amplitude and our_in conntains frequency (Hz)
for(int i=0;i<data[o-1];i++) // getting amplitude from complex number
{
    out_r[i]=sqrt(out_r[i]*out_r[i]+out_im[i]*out_im[i]); // to increase the
speed delete sqrt
    out_im[i]=i*Frequency/N;
    /*
    Serial.print(out_im[i]); Serial.print("Hz");
    Serial.print("\ "); // un comment to print freuency bin
    Serial.println(out_r[i]);
    */
}

x=0; // peak detection
for(int i=1;i<data[o-1]-1;i++)
{
    if(out_r[i]>out_r[i-1] && out_r[i]>out_r[i+1])
    {in_ps[x]=i; //in_ps array used for storage of peak number
    x=x+1;}
}

s=0;
c=0;
for(int i=0;i<x;i++) // re arraange as per magnitude
{
    for(int j=c;j<x;j++)
    {
        if(out_r[in_ps[i]]<out_r[in_ps[j]])
        {s=in_ps[i];

```

```

        in_ps[i]=in_ps[j];
        in_ps[j]=s;}
    }
    c=c+1;
}

for(int i=0;i<5;i++)    // updating f_peak array (global variable)with
descending order
{
    f_peaks[i]=out_im[in_ps[i]];
}

}

float sine(int i)
{
    int j=i;
    float out;
    while(j<0){j=j+360;}
    while(j>360){j=j-360;}
    if(j>-1 && j<91){out= sine_data[j];}
    else if(j>90 && j<181){out= sine_data[180-j];}
    else if(j>180 && j<271){out= -sine_data[j-180];}
    else if(j>270 && j<361){out= -sine_data[360-j];}
    return (out/255);
}

float cosine(int i)
{
    int j=i;
    float out;
    while(j<0){j=j+360;}
    while(j>360){j=j-360;}
    if(j>-1 && j<91){out= sine_data[90-j];}
    else if(j>90 && j<181){out= -sine_data[j-90];}
    else if(j>180 && j<271){out= -sine_data[270-j];}
    else if(j>270 && j<361){out= sine_data[j-270];}
    return (out/255);
}

//-----//

```

## Lampiran 2 Define Function JST

```
1  const int InputNodes = 2;
2  const int HiddenNodes = 3;
3  const int OutputNodes = 3;
4  int idxI, idxJ;
5
6  const float w1[HiddenNodes][InputNodes] = {
7      {2.5859870424671722589, -0.20441694706991508923},
8      {-0.43764670318069864852, -6.4220275224990697183},
9      {0.21813155799876648988, -8.2769669728104435791}};
10
11 const float b1[HiddenNodes] = {
12     -1.2198718486653126192,
13     -2.5733074356579290765,
14     -1.8926423388960180727};
15
16 const float w2[OutputNodes][HiddenNodes] = {
17     {1.54809825246362, -17.9704244951953, -5.27619712455950},
18     {1.09308242834741, 18.8669573710212, 1.15381792557886},
19     {-3.16842678459342, -0.243377681700155, 4.18028398115929}};
20
21 const float b2[] = {
22     -0.889216745884705,
23     -2.13672130940209,
24     3.16497735285852};
25
26 const float xoffset[InputNodes] = {0.43, 0.3};
27 const float gain[InputNodes] = {0.854700854700855, 1.21951219512195};
28 const float ymin[InputNodes] = {-1, -1};
29 float probabilities[OutputNodes];
```

## Lampiran 3 Code Function Menghitung JST

### 1. Function Tansig Apply

```
1  float tansig_apply(float n)
2  {
3      return 2.0 / (1.0 + exp(-2.0 * n)) - 1.0;
4  }
```

## 2. Define Function dan Normalization Process

```
1 String getClassificationNM(float lf, float hf)
2 {
3     float input[InputNodes];
4     float output[OutputNodes];
5     float hidden[HiddenNodes];
6     double norm_input[InputNodes];
7     int indexMaks;
8     float maks;
9     float numerator[3] = {0};
10    float denominator = 0;
11
12    input[0] = lf;
13    input[1] = hf;
14    // input normalization
15    for (idxI = 0; idxI < InputNodes; idxI++)
16    {
17        norm_input[idxI] = input[idxI] - xoffset[idxI];
18        norm_input[idxI] = norm_input[idxI] * gain[idxI];
19        norm_input[idxI] = norm_input[idxI] + ymin[idxI];
20    }
21
22    // empty memory
23    for (idxI = 0; idxI < HiddenNodes; idxI++)
24    {
25        hidden[idxI] = 0;
26    }
27
28    // empty memory
29    for (idxI = 0; idxI < OutputNodes; idxI++)
30    {
31        output[idxI] = 0;
32    }
```

### 3. Perhitungan Hidden Layer dan Output Layer

```
1
2 // compute hidden layer
3 for (idxI = 0; idxI < HiddenNodes; idxI++)
4 {
5     hidden[idxI] = b1[idxI];
6     for (idxJ = 0; idxJ < InputNodes; idxJ++)
7     {
8         hidden[idxI] += w1[idxI][idxJ] * norm_input[idxJ];
9     }
10    hidden[idxI] = tansig_apply(hidden[idxI]);
11 }
12
13 // calculate the output of the output units
14 for (idxI = 0; idxI < OutputNodes; idxI++)
15 {
16     output[idxI] = b2[idxI];
17     for (idxJ = 0; idxJ < HiddenNodes; idxJ++)
18     {
19         output[idxI] += w2[idxI][idxJ] * hidden[idxJ];
20     }
21 }
```



#### 4. Perhitungan Probabilitas

```
1 // Calculate Probabilitas kelas
2 numerator[3] = {0};
3 denominator = 0;
4 for (idxI = 0; idxI < 3; idxI++)
5 {
6     numerator[idxI] = exp(output[idxI]);
7     denominator += numerator[idxI];
8 }
9 if (denominator == 0)
10 {
11     denominator = 1;
12 }
13 for (idxI = 0; idxI < 3; idxI++)
14 {
15     probabilities[idxI] = numerator[idxI] / denominator;
16 }
17
18 // find nilai kelas terbesar
19 maks = probabilities[0];
20 indexMaks = 0;
21 for (idxI = 1; idxI < OutputNodes; idxI++)
22 {
23     if (probabilities[idxI] > maks)
24     {
25         maks = probabilities[idxI];
26         indexMaks = idxI;
27     }
28 }
29
30 // mengembalikan nilai terbesar
31 if (indexMaks == 0)
32 {
33     return "Normal"; // Normal
34 }
35 else if (indexMaks == 1)
36 {
37     return "Apnea-C"; // APNEA-C
38 }
39 else
40 {
41     return "Apnea-O"; // APNEA-O
42 }
43 }
44
```