



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IS184853

# RANCANG BANGUN APLIKASI PENGUKUR TINGKAT STRES BERBASIS ANDROID

## *DESIGN AND DEVELOPMENT OF STRESS LEVEL MEASUREMENT APPLICATION ANDROID BASED*

DAUD MUHAJIR  
NRP. 05211540000058

Dosen Pembimbing  
Nisfu Asrul Sani, S.Kom., M.Sc.  
Faizal Mahananto, S.Kom., M.Eng., Ph.D.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019





**TUGAS AKHIR - IS184853**

# **RANCANG BANGUN APLIKASI PENGUKUR TINGKAT STRES BERBASIS ANDROID**

**DAUD MUHAJIR  
NRP. 05211540000058**

**Dosen Pembimbing  
Nisfu Asrul Sani, S.Kom., M.Sc.  
Faizal Mahananto, S.Kom., M.Eng., Ph.D.**

**DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*Halaman ini sengaja dikosongkan*



**UNDERGRADUATE THESIS - IS184853**

# **DESIGN AND DEVELOPMENT OF STRESS LEVEL MEASUREMENT APPLICATION ANDROID BASED**

**DAUD MUHAJIR  
NRP. 05211540000058**

**Supervisor  
Nisfu Asrul Sani, S.Kom., M.Sc.  
Faizal Mahananto, S.Kom., M.Eng., Ph.D.**

**DEPARTMENT OF INFORMATION SYSTEMS  
Faculty of Information and Communication Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*Halaman ini sengaja dikosongkan*

# LEMBAR PENGESAHAN

## RANCANG BANGUN APLIKASI PENGUKUR TINGKAT STRES BERBASIS ANDROID

### TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**DAUD MUHAJIR**  
**NRP. 05211540000058**

Surabaya, 11 Juli 2019

**KEPALA  
DEPARTEMEN SISTEM INFORMASI**



**Mahendrawathi ER, S.T., M.Sc., Ph.D.**  
**NIP. 19761011 200604 2 001**

*Halaman ini sengaja dikosongkan*



# LEMBAR PERSETUJUAN

## RANCANG BANGUN APLIKASI PENGUKUR TINGKAT STRES BERBASIS ANDROID

### TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**DAUD MUHAJIR**  
**NRP. 0521154000058**

Disetujui Tim Penguji : Tanggal Ujian : 11 Juli 2019  
Periode Wisuda : September 2019

**Nisfu Asrul Sani, S.Kom., M.Sc.**

  
(Pembimbing I)

**Faizal Mahananto, S.Kom., M.Eng., Ph.D.**

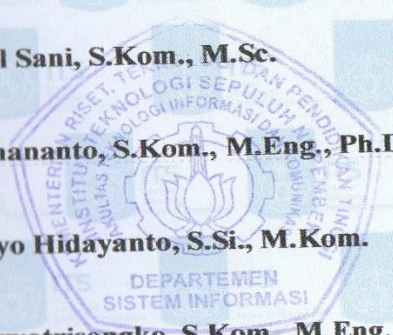
  
(Pembimbing II)

**Bekti Cahyo Hidayanto, S.Si., M.Kom.**

  
(Penguji I)

**Hatma Suryotrisongko, S.Kom., M.Eng.**

  
(Penguji II)



*Halaman ini sengaja dikosongkan*

# RANCANG BANGUN APLIKASI PENGUKUR TINGKAT STRES BERBASIS ANDROID

Nama : Daud Muhajir  
NRP : 0521154000058  
Departemen : Sistem Informasi ITS  
Pembimbing I : Nisfu Asrul Sani, S.Kom., M.Sc.  
Pembimbing II : Faizal Mahananto, S.Kom., M.Eng., Ph.D.

## ABSTRAK

*Stres adalah respon alami manusia terhadap tekanan ketika dihadapkan dengan tantangan dan terkadang situasi yang berbahaya. Bukti neurobiologis saat ini menunjukkan bahwa Heart Rate Variability (HRV) yang diperoleh dari sinyal electrocardiogram (ECG) dipengaruhi oleh stres dan mendukung penggunaannya untuk penilaian objektif kesehatan psikologis dan stres.*

*Pengambilan sinyal ECG dan analisis metrik HRV saat ini masih menggunakan sistem yang berbeda dan belum terintegrasi. Untuk itu dibuatlah sistem pengukur tingkat stres berbasis android yang dapat mengambil sinyal ECG secara wireless dan dapat mengetahui hasil analisis metrik HRV secara langsung.*

*Sistem yang telah dibangun menggunakan 3 sistem yang terintegrasi yaitu sistem mobile (android), sistem ESP8266 (mikrokontroler), dan sistem webserver. Menggunakan metode validasi NASA TLX, pengukuran tingkat stres dengan analisa metrik HRV berhasil mengklasifikasikan 7 dari 10 responden atau memiliki performa akurasi sebesar 70%.*

***Kata Kunci : stres, electrocardiogram, heart rate variability, metrik HRV***

*Halaman ini sengaja dikosongkan*

# **DESIGN AND DEVELOPMENT OF STRESS LEVEL MEASUREMENT APPLICATION ANDROID BASED**

Name : Daud Muhajir  
NRP : 0521154000058  
Department : Information Systems ITS  
Supervisor I : Nisfu Asrul Sani, S.Kom., M.Sc.  
Supervisor II : Faizal Mahananto, S.Kom., M.Eng., Ph.D.

## **ABSTRACT**

*Stress is a natural human response to stress when faced with challenges and sometimes dangerous situations. Current neurobiological evidence shows that Heart Rate Variability (HRV) obtained from electrocardiogram (ECG) signals is influenced by stress and supports its use for objective assessment of psychological health and stress.*

*ECG signal retrieval and HRV metric analysis currently still use different systems and are not yet integrated. For this reason an Android-based stress level measurement system was created that can take ECG signals wirelessly and can find out the results of HRV metrics analysis directly.*

*The system has been built using 3 integrated systems, the mobile system (android), the ESP8266 system (microcontroller), and the webserver system. Using the NASA TLX validation method, measuring stress levels with HRV metric analysis successfully classified 7 out of 10 respondents or has a performance accuracy of 70%.*

***Keywords : stress, electrocardiogram, heart rate variability, HRV metrics***

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan serta hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini yang merupakan salah satu syarat kelulusan di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Segeanp keluarga besar terutama kedua orang tua, kakak dan adik penulis, Bapak Dwi Windarto Rahmono, Ibu Heriyati, Bintang Fajar Alhidayah dan Hanif Abdullah yang senantiasa mendoakan, memberikan motivasi dan semangat, sehingga penulis mampu menyelesaikan pendidikan sarjana ini dengan baik.
2. Ibu Mahendrawathi Er, S.T., M.Sc., Ph.D. selaku Kepala Departemen Sistem Informasi ITS, Bapak Nisfu Asrul Sani, S.Kom., M.Sc selaku Ketua Program Studi Sarjana Departemen Sistem Informasi ITS, serta seluruh dosen pengajar beserta staf dan karyawan Departemen Sistem Informasi ITS selama penulis menjalani perkuliahan.
3. Bapak Nisfu Asrul Sani, S.Kom., M.Sc. sebagai dosen wali penulis selama menempuh pendidikan di Departemen Sistem Informasi ITS.
4. Bapak Nisfu Asrul Sani, S.Kom., M.Sc. dan Bapak Faizal Mahananto, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan mendukung dengan

memberikan ilmu, petunjuk, dan motivasi dalam penyelesaian tugas akhir ini.

5. Bapak Bekti Cahyo Hidayanto, S.Si., M.Sc. dan Bapak Hatma Suryotrisongko, S.Kom., M.Eng. selaku dosen penguji yang telah memberikan kritik dan saran dalam penyempurnaan tugas akhir ini.
6. Teman-teman Sistem Informasi angkatan 2015 (Lannister) yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.
7. Teman-teman laboratorium ADDI, RDIB, SE, dan MSI yang telah mempersilakan penulis bernaung dan mencari inspirasi dalam mengerjakan tugas akhir ini.
8. Seluruh pihak-pihak lainnya yang tidak dapat disebutkan satu per satu, yang telah membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Penyusunan laporan tugas akhir ini masih jauh dari kata sempurna, sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 11 Juli 2019

Penulis

Daud Muhajir



## DAFTAR ISI

HALAMAN JUDUL.....	I
LEMBAR PENGESAHAN.....	V
LEMBAR PERSETUJUAN.....	VII
ABSTRAK.....	IX
ABSTRACT.....	XI
KATA PENGANTAR.....	XIII
DAFTAR ISI.....	XV
DAFTAR GAMBAR.....	XIX
DAFTAR TABEL.....	XXI
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Relevansi.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Studi Literatur.....	7
2.2. Dasar Teori.....	8
2.2.1. Android.....	8
2.2.2. <i>Software Development Life Cycle</i> .....	9
2.2.3. <i>Unified Modeling Language</i> .....	9
2.2.4. ESP8266.....	11
2.2.5. Flutter.....	12
2.2.6. <i>Electrocardiogram</i> .....	13
2.2.7. Flask.....	15
2.2.8. <i>Testing</i> .....	15
BAB III METODOLOGI.....	17
3.1. Tahap Identifikasi.....	18
3.1.1. Referensi.....	18
3.1.2. Gambaran Umum Sistem.....	19
3.1.3. Observasi.....	19
3.1.4. Spesifikasi.....	19
3.2. Tahap Desain.....	20

3.2.1. Desain UI/UX .....	20
3.2.2. Perancangan Diagram .....	20
3.2.3. Dokumentasi Desain .....	20
3.3. Tahap Pengembangan .....	20
3.3.1. Pengembangan Android.....	21
3.3.2. Pengembangan ESP8266 .....	21
3.3.3. Pengembangan <i>Webserver</i> .....	21
3.3.4. Prototipe .....	21
3.4. Tahap Prototipe.....	21
3.4.1. Percobaan Prototipe .....	21
3.4.2. Perbaikan Prototipe .....	21
3.4.3. Aplikasi .....	22
3.5. Tahap Pengujian .....	22
3.5.1. Perancangan Pengujian .....	22
3.5.2. Implementasi Pengujian .....	22
3.5.3. Dokumentasi Pengujian .....	22
3.6. Tahap Deployment.....	22
3.6.1. Uji Responden.....	22
3.6.2. Analisa Hasil.....	22
3.6.3. Hasil Analisa.....	23
3.7. Tahap Dokumentasi .....	23
3.7.1. Penyusunan Laporan .....	23
3.7.2. Dokumen Tugas Akhir.....	23
BAB IV PERANCANGAN .....	25
4.1. Analisis Kebutuhan.....	25
4.1.1. Referensi .....	25
4.1.2. Observasi.....	27
4.2. Desain Sistem .....	28
4.2.1. Perancangan <i>Use Case</i> .....	28
4.2.2. Perancangan Arsitektur .....	31
4.2.3. Perancangan Desain UI/UX .....	32
4.2.4. Perancangan Sistem Mobile.....	35
4.2.5. Perancangan Sistem ESP8266.....	37
4.2.6. Perancangan Sistem <i>Webserver</i> .....	39
4.3. Perancangan <i>Test Case</i> .....	41
4.4. Perancangan Validasi.....	42
4.5. Perancangan <i>Deployment</i> .....	44
BAB V IMPLEMENTASI .....	47

5.1. Lingkungan Implementasi .....	47
5.2. Implementasi Desain UI/UX .....	49
5.3. Implementasi Sistem Android .....	50
5.4. Implementasi Sistem ESP8266.....	54
5.5. Implementasi Sistem Server .....	60
5.6. Implementasi Test Case.....	63
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>	<b>67</b>
7.1. Performa Sistem .....	67
7.1.1. Uji Integrasi .....	67
7.1.2. Uji Jarak.....	67
7.1.3. Uji Ketepatan Algoritma.....	67
7.1.4. Uji Kecepatan .....	68
7.2. NASA-TLX .....	71
7.2.1. Uji Keseragaman .....	73
7.2.2. Uji Kecukupan Data.....	74
7.3. Metrik HRV .....	75
7.4. Analisis Data <i>Train</i> .....	76
7.5. Analisis Data <i>Testing</i> .....	77
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>83</b>
7.1. Kesimpulan .....	83
7.2. Saran.....	83
<b>DAFTAR PUSTAKA .....</b>	<b>85</b>
<b>BIODATA PENULIS .....</b>	<b>89</b>

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 1.1. Efek stres pada performa dan kesehatan .....	2
Gambar 2.1. UML .....	10
Gambar 2.2. Wemos D1 R2 .....	11
Gambar 2.3. Modul AD8232.....	12
Gambar 2.4. Modul SD Card .....	12
Gambar 2.5. Sinyal Electrocardiogram .....	14
Gambar 3.1. Metodologi .....	18
Gambar 3.2. Gambaran umum sistem .....	19
Gambar 4.1. Use case aplikasi pengukur tingkat stres .....	29
Gambar 4.2. Deployment diagram .....	32
Gambar 4.3. Mockup oscilloscope.....	33
Gambar 4.4. Mockup test .....	33
Gambar 4.5. Mockup analyze .....	34
Gambar 4.6. Mockup result.....	34
Gambar 4.7. Mockup detail result.....	35
Gambar 4.8. Class diagram pada sistem android .....	36
Gambar 4.9. Activity diagram monitoring sinyal ECG.....	36
Gambar 4.10. Activity diagram melakukan tes .....	37
Gambar 4.11. Activity diagram melihat hasil tes .....	37
Gambar 4.12. Data flow diagram ESP8266 .....	38
Gambar 4.13. Data flow diagram webserver.....	39
Gambar 4.14. Flowchart deployment.....	45
Gambar 5.1. Tampilan Osiloskop dan Test.....	49
Gambar 5.2. Tampilan Analyze .....	50
Gambar 5.3. Tampilan Hasil .....	50
Gambar 5.4. Kode program oscilloscope .....	51
Gambar 5.5. Kode program merekam sinyal ECG .....	52
Gambar 5.6. Kode program analisis.....	53
Gambar 5.7. Kode program menampilkan hasil.....	54
Gambar 5.8. Rangkaian sistem ESP8266 .....	55
Gambar 5.9. Pendefinisian variabel ESP8266 .....	56
Gambar 5.10 Kode program setup ESP8266.....	57
Gambar 5.11. Kode program fungsi ESP8266 .....	59
Gambar 5.12. Kode program loop ESP8266.....	60
Gambar 5.13. Kode program menerima file csv .....	60

Gambar 5.14. Kode program deteksi hrv .....	61
Gambar 5.15. Kode program menghitung metrik hrv .....	61
Gambar 5.16. Kode program reads.....	62
Gambar 5.17. Kode program dashboard.....	63
Gambar 6.1. Contoh miss detect titik puncak.....	68
Gambar 6.2. Uji kecepatan transfer csv dan analisis .....	70

## DAFTAR TABEL

Tabel 2.1. Studi literatur.....	7
Tabel 4.1. Kebutuhan fungsional .....	26
Tabel 4.2. Kebutuhan non fungsional .....	27
Tabel 4.3. Tumpukan teknologi .....	27
Tabel 4.4. Pemetaan kebutuhan fungsional.....	28
Tabel 4.5. Use case description melihat oscilloscope .....	30
Tabel 4.6. Use case description memulai tes.....	30
Tabel 4.7. Use case description melihat hasil tes .....	31
Tabel 4.8. URI pada sistem ESP8266 .....	38
Tabel 4.9. Kolom pada table database.....	40
Tabel 4.10. URI pada server python .....	40
Tabel 4.11. Perancangan test case .....	41
Tabel 4.12. Dimensi NASA-TLX .....	42
Tabel 4.13. Interpretasi skor NASA-TLX.....	44
Tabel 5.1. Spesifikasi smartphone implementasi .....	47
Tabel 5.2. Spesifikasi laptop implementasi.....	47
Tabel 5.3. Teknologi pengembangan sistem mobile .....	48
Tabel 5.4. Teknologi pengembangan sistem server .....	48
Tabel 5.5. Teknologi pengembangan sistem ESP8266 .....	48
Tabel 5.6. Rangkaian modul AD8232 dengan Wemos D1 R2 .....	55
Tabel 5.7. Rangkaian modul SD Card dengan Wemos D1 R2 .....	55
Tabel 5.8. Implementasi test case.....	64
Tabel 6.1. Uji ketepatan algoritma .....	68
Tabel 6.2. Uji kecepatan transfer sinyal dari sensor.....	69
Tabel 6.3. Uji kecepatan transfer csv dan analisis .....	70
Tabel 6.4. Hasil NASA TLX data train.....	71
Tabel 6.5. Hasil metrik HRV data train time domain .....	75
Tabel 6.6. Hasil metrik HRV data train frequency domain....	75
Tabel 6.7. Hasil threshold .....	76
Tabel 6.8. Hasil klasifikasi metrik HRV data train .....	77
Tabel 6.9. Hasil NASA TLX data testing .....	78
Tabel 6.10. Hasil metrik HRV data testing time domain .....	80

Tabel 6.11. Hasil metrik HRV data testing frequency domain .....	80
Tabel 6.12. Hasil klasifikasi metrik HRV data testing .....	81



# **BAB I**

## **PENDAHULUAN**

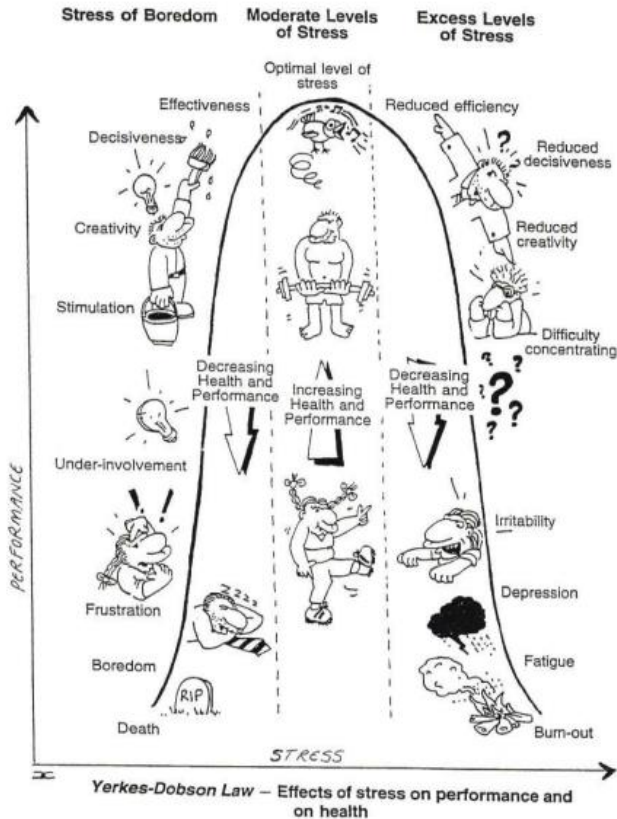
Pada bab pertama akan dijelaskan mengenai pendahuluan pengerjaan tugas akhir ini, akan diuraikan proses identifikasi masalah yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir, dan relevansi penelitian tugas akhir dengan bidang minat yang terdapat pada Departemen Sistem Informasi ITS. Berdasarkan uraian pada bagian ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir ini dapat dipahami.

### **1.1. Latar Belakang**

Stres adalah respon alami manusia terhadap tekanan ketika dihadapkan dengan tantangan dan terkadang situasi yang berbahaya. Tekanan itu bukan hanya tentang apa yang terjadi di sekitar kita, tetapi sering juga tentang tuntutan yang kita tempatkan pada diri kita sendiri [1]. Sebagai contoh tekanan yang datang dari eksternal ialah ketika kita menghadapi ujian, sedangkan yang berasal dari internal ialah kondisi pikiran kita yang mengkhawatirkan sebuah ujian.

Bagi kebanyakan orang, stres dipandang sebagai suatu hal yang negatif, namun stres juga dapat memacu kita untuk mencapai sesuatu hal yang baik. Misalnya, dengan tekanan kita menjadi termotivasi untuk memenangkan olimpiade, mengerjakan makalah, atau mempersiapkan ujian. Untuk itu, stres juga diperlukan, namun stres yang berlebih juga akan merugikan [2].

Stres yang tidak terkelola dapat memiliki pengaruh besar pada kemampuan, kinerja, kepercayaan diri, tingkat energi, dan kualitas hidup [3]. Gambar 1.1. dibawah merupakan kurva tipe U terbalik yang telah digunakan untuk menggambarkan efek stres terhadap kinerja. Hal itu menunjukkan ketika stres meningkat, begitu pula kinerja dan efisiensi. Namun jika stres terus meningkat melampaui titik optimal, kinerja mulai menurun (terlalu banyak stres) [4].



Gambar 1.1. Efek stres pada performa dan kesehatan

Setiap orang pasti mempunyai tuntutan dan tekanan, hal ini menjamin bahwa setiap manusia pasti mengalami stres. Stres tidak dapat dihindari namun dapat dikelola [2]. Untuk itu kita perlu mengukur tingkat stres untuk memantau tingkat stres yang kita alami saat ini. Bukti neurobiologis saat ini menunjukkan bahwa *Heart Rate Variability* (HRV) dipengaruhi oleh stres dan mendukung penggunaannya untuk penilaian objektif kesehatan psikologis dan stres [5].

HRV merupakan durasi antara dua puncak gelombang R yang berdekatan atau biasa disebut sebagai interval R-R pada sinyal

*electrocardiogram* (ECG) [6]. ECG merupakan suatu ukuran sinyal yang menunjukkan bagaimana aktivitas listrik dari otot jantung berubah dari waktu ke waktu selama setiap siklus jantung [7]. Perlu dilakukan perhitungan metrik HRV untuk mengenali tingkat stres yang memungkinkan analisis efek stres yang baik sebagai indeks stres psikologis dan memberikan wawasan tentang reaksi saraf otonom sistem stres [8].

Untuk pengambilan sinyal ECG saat ini masih menggunakan kabel serta analisis HRV pada sistem yang berbeda, belum terintegrasi sehingga tidak memungkinkan untuk mengetahui hasil analisis secara langsung.

Untuk itu dibuatlah sistem pengukur tingkat stres berbasis android yang dapat mengambil sinyal ECG secara *wireless* dan dapat mengetahui hasil analisis metrik HRV secara langsung. Sistem yang akan dibuat memanfaatkan mikrokontroler sebagai sistem untuk mendapatkan sinyal ECG dan sistem *webserver* untuk menganalisis metrik HRV.

Dengan demikian, pengerjaan tugas akhir ini diharapkan dapat terciptanya suatu sistem aplikasi berbasis android yang dapat menerima sinyal ECG secara *wireless* melalui mikrokontroler serta menghasilkan perhitungan metrik HRV secara langsung yang nantinya dapat digunakan sebagai indeks pengukur tingkat stres maupun penelitian lebih lanjut yang melibatkan metrik HRV.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah pada tugas akhir ini adalah sebagai berikut :

1. Bagaimana mendapatkan sinyal ECG secara *wireless* dan hasil analisis metrik HRV secara langsung?
2. Bagaimana mengembangkan aplikasi pengukur tingkat stres dengan teknologi *mobile*, mikrokontroler, dan *webserver*?

### 1.3. Batasan Masalah

Berdasarkan permasalahan yang telah diuraikan sebelumnya, adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Sistem aplikasi pengukur tingkat stres yang dibuat dalam bentuk *prototype*
2. Data sinyal ECG diperoleh secara langsung menggunakan sensor ECG pada mikrokontroler
3. Proses pengiriman data terjadi secara *wireless*
4. Perhitungan metrik HRV dilakukan pada sistem *Webserver*
5. Validasi tingkat stres menggunakan metode NASA-TLX
6. Sistem akan diujicobakan kepada 10 responden
7. Responden berjenis kelamin laki-laki dan berusia antara 21 sampai 23 tahun

Dengan demikian, selesainya tugas akhir ini adalah dengan terbentuknya aplikasi pengukur tingkat stres yang telah memenuhi kebutuhan, telah melewati tahapan pengujian.

### 1.4. Tujuan

Berdasarkan latar belakang dan rumusan masalah yang telah disebutkan sebelumnya, adapun tujuan dari penelitian tugas akhir ini adalah mengembangkan aplikasi pengukur tingkat stres secara *wireless* dan hasil analisis metrik HRV secara langsung, serta dapat digunakan untuk pengembangan lebih lanjut yang melibatkan penggunaan metrik HRV.

### 1.5. Manfaat

Manfaat yang diharapkan dapat diperoleh dari penelitian tugas akhir ini adalah sebagai berikut :

1. Mengembangkan aplikasi dalam dunia kesehatan sebagai alat pengukur tingkat stres.
2. Memberikan tambahan wawasan dan menyediakan referensi dalam pengembangan dunia kesehatan yang melibatkan metrik HRV.

### **1.6. Relevansi**

Tugas akhir ini disusun untuk memenuhi salah satu syarat kelulusan sebagai Sarjana Komputer di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi (FTIK) Institut Teknologi Sepuluh Nopember (ITS). Tugas akhir ini sesuai dengan penerapan mata kuliah dari laboratorium Infrastruktur dan Keamanan Teknologi Informasi (IKTI) Departemen Sistem Informasi FTIK ITS, yaitu Internet Untuk Segala dan Teknologi Bergerak. Selain itu, juga berkaitan dengan mata kuliah wajib, yaitu Konstruksi dan Pengujian Perangkat Lunak.

*Halaman ini sengaja dikosongkan*

## BAB II TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan mengenai studi literatur terhadap penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

### 2.1. Studi Literatur

Dalam proses pengerjaan penelitian tugas akhir ini, dilakukan pencarian penelitian-penelitian yang sudah dilakukan untuk dijadikan sebagai referensi dalam pengerjaan penelitian tugas akhir ini, seperti pada Tabel 2.1.

*Tabel 2.1. Studi literatur*

<b>1) Rancang Bangun Sistem Pengiriman Data Electrocardiograph dengan media Wi-Fi</b>
<b>Penulis; Tahun</b> Rijal Abdurrahman, Angga Rusdinar dan Agus Ganda Permana; 2017 [9]
<b>Pembahasan</b> Pada penelitian ini penulis melakukan pengiriman sinyal ECG menggunakan modul AD8232 dengan mikrokontroler Wemos D1 mini. Hasilnya sinyal dapat dikirimkan melalui media WI-Fi kepada <i>notebook</i> hingga jarak 100 cm. jarak antara <i>server</i> dan <i>client</i> tidak berpengaruh terhadap keakuratan data-data yang dikirimkan.
<b>Keterkaitan</b> Penelitian Tugas Akhir ini berkaitan dengan digunakannya sensor ECG dan mikrokontroler yang selanjutnya dikirimkan kepada <i>webservice</i> menggunakan media <i>Wi-Fi</i> .
<b>2) Heart Rate Variability Metrics for Fine-Grained Stress Level Assessment</b>
<b>Penulis; Tahun</b> Tânia Pereira, Pedro R. Almeida, João P.S. Cunha, Ana Aguiar; 2017 [8]
<b>Pembahasan</b>

<p>Pada penelitian ini bertujuan untuk mengevaluasi waktu dan domain dari metrik HRV. Untuk penilaian tingkat stres menggunakan <i>short-time window</i>. Metode yang dilakukan dengan menggunakan sinyal ECG dari 14 relawan yang di <i>monitoring</i> menggunakan <i>Vital Jacket</i> sementara mereka melakukan <i>Trier Social Stress Test (TSST)</i>. <i>Window length</i> yang digunakan mulai dari 220 s sampai 50 s dengan berbagai perhitungan metrik HRV <i>time series</i> dan <i>frequency domain</i>. Hasil metrik yang konsisten untuk membedakan fase stres dan tidak stres adalah AVNN, rMSSD, SDNN, dan PNN20. Dengan hasil yang paling baik adalah metrik AVNN dengan 50 s <i>window length</i>.</p>
<p><b>Keterkaitan</b>  Penelitian Tugas Akhir yang dilakukan ini berkaitan dengan penggunaan metrik HRV sebagai indeks yang menentukan tingkat stres.</p>
<p><b>3) Analisa Beban Kerja Mental dengan Metode NASA TLX pada Operator Kargo di PT. Dharma Bandar Mandala (PT. DBM)</b></p>
<p><b>Penulis; Tahun</b>  Muhammad Arasyandi dan Arfan Bakhtiar; 2016 [10]</p>
<p><b>Pembahasan</b>  Pada penelitian ini penulis menggunakan metode NASA TLX untuk mengetahui tingkat stres berdasarkan beban kerja mental pada operator kargo di PT Dharma Bandar Mandala (PT. DBM)</p>
<p><b>Keterkaitan</b>  Pada penelitian Tugas Akhir ini menggunakan metode NASA TLX untuk menentukan validasi perbandingan tingkat stres</p>

## 2.2. Dasar Teori

### 2.2.1. Android

Android merupakan sistem operasi seluler yang saat ini dikembangkan oleh Google, berdasarkan pada kernel linux dan dirancang terutama untuk perangkat seluler layar sentuh seperti ponsel pintar dan tablet. Ada banyak alasan mengapa semakin



banyak orang tertarik mempelajari cara mengembangkan aplikasi android. Tidak dapat disangkal Android adalah sistem operasi seluler paling populer, dengan hampir 2 miliar perangkat diaktifkan [11]. Android menawarkan pendekatan terpadu untuk pengembangan aplikasi pada perangkat seluler yang artinya pengembang hanya perlu mengembangkan untuk Android, dan aplikasi mereka dapat dijalankan pada berbagai perangkat yang diberdayakan oleh Android [12]. Kode program untuk Android bersifat *free* dan *open source*.

### **2.2.2. Software Development Life Cycle**

*Software Development Life Cycle* (SDLC) adalah proses alur kerja yang mendefinisikan tahapan inti dan kegiatan siklus pengembangan perangkat lunak. Dapat digunakan oleh analis sistem, perancang, dan pengembang untuk merencanakan dan mengimplementasikan sistem atau produk tepat waktu dan sesuai anggaran. SDLC ini terdiri dari langkah-langkah detail dan aktivitas yang menggambarkan bagaimana merancang, mengembangkan, memelihara, mengganti, mengubah, meningkatkan, menguji atau bahkan meluncurkan perangkat lunak. Dengan banyaknya metodologi pengembangan, seperti model *waterfall*, model *spiral*, model *agile*, model *prototype*, dan sebagainya, tidaklah mudah untuk memilih strategi yang tepat sehingga memungkinkan untuk menggabungkan beberapa metodologi agar sesuai dalam satu proyek [13].

### **Mobile Application Development Lifecycle**

*Mobile Application Development Lifecycle* (MADLC) adalah siklus untuk pengembangan aplikasi seluler. MADLC akan membantu pengembang aplikasi seluler dalam mengembangkan aplikasi seluler. Siklus ini mencakup fase berikut : Identifikasi, Desain, Pengembangan, Prototipe, Pengujian, *Deployment* dan Pemeliharaan [14].

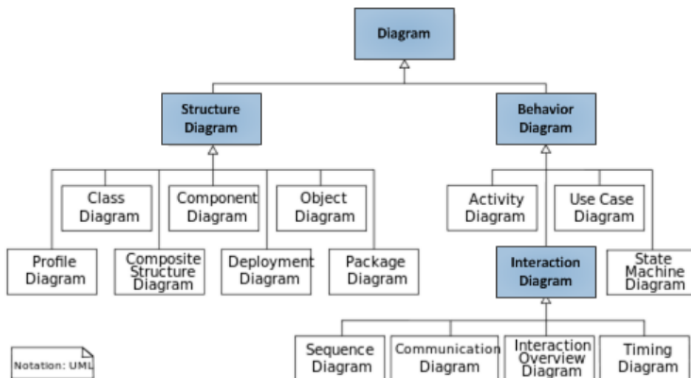
### **2.2.3. Unified Modeling Language**

*Unified Modeling Language* (UML) adalah bahasa pemodelan standar di bidang rekayasa perangkat lunak berorientasi objek. UML mencakup teknik notasi grafis untuk dibuat model visual

dari sistem perangkat lunak berorientasi objek. UML menggabungkan teknik pemodelan data, pemodelan bisnis, pemodelan objek, dan pemodelan komponen serta dapat digunakan di seluruh siklus pengembangan perangkat lunak dan lintas teknologi implementasi yang berbeda [15].

Untuk itu, UML menyediakan 14 tipe diagram yang dapat digunakan sesuai dengan kebutuhan pengembangan perangkat lunak, diagram-diagram tersebut dibagi menjadi 2 elemen yang berbeda [15], yaitu :

- *Structure diagrams* mewakili aspek struktural dari suatu sistem, adapun diagram yang termasuk kategori ini, yaitu : *class diagram*, *component diagram*, *composite structure diagram*, *deployment diagram*, *object diagram*, dan *package diagram*.
- *Behavior diagrams* mewakili aspek dinamis dari suatu sistem, adapun diagram yang termasuk kategori ini, yaitu : *activity diagram*, *communication diagram*, *interaction overview diagram*, *sequence diagram*, *state machine diagram*, *timing diagram*, dan *use case diagram*.



Gambar 2.1. UML

### 2.2.4. ESP8266

ESP8266 merupakan *microchip Wi-Fi* murah yang menggabungkan kemampuan TCP / IP *stack* dan mikrokontroler. Benda ini diperkenalkan oleh pabrik *Espressif Systems*. ESP8266 NodeMcu adalah perangkat yang kompleks, yang menggabungkan beberapa fitur papan Arduino biasa dengan kemungkinan menghubungkan ke internet. Modul dan mikrokontroler Arduino selalu menjadi pilihan tepat untuk memasukkan otomatisasi ke dalam proyek yang relevan. Tetapi modul-modul ini datang dengan sedikit kelemahan karena mereka tidak memiliki kemampuan WiFi bawaan, selanjutnya, kita perlu menambahkan protokol WiFi eksternal ke dalam perangkat ini untuk membuatnya kompatibel dengan saluran internet [16]. Salah satu tipe ESP8266 adalah Wemos D1 R2 yang dapat dilihat pada Gambar 2.2.

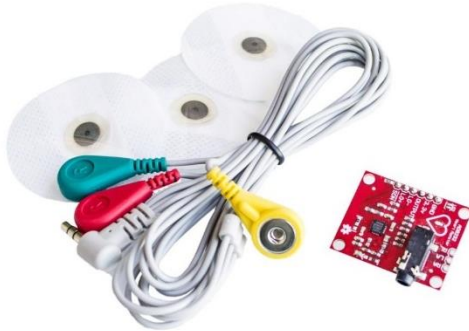


Gambar 2.2. Wemos D1 R2

### Modul AD8232

Modul AD8232 adalah blok pengkondisi sinyal terintegrasi untuk ECG dan aplikasi pengukuran biopotensial lainnya. Modul ini dirancang untuk mengekstrak, memperkuat, dan memfilter sinyal biopotensial ketika adanya *noise*, seperti yang diciptakan oleh gerakan atau penempatan elektroda jarak jauh. Desain ini memungkinkan untuk *ultralow* konverter analog-ke-digital (ADC) atau tertanam mikrokontroler untuk

mendapatkan sinyal *output* dengan mudah [17]. Modul AD8232 dapat dilihat pada Gambar 2.3.



Gambar 2.3. Modul AD8232

### Modul SD Card

Modul SD Card memungkinkan untuk menambahkan kemampuan penyimpanan data ke mikrokontroler dari kartu memori biasa. Modul ini memiliki *slot* untuk SD Card ukuran biasa, *regulator* tegangan, kapasitor dan *resistor* untuk SPI-bus [18]. Modul SD Card dapat dilihat pada Gambar 2.4.



Gambar 2.4. Modul SD Card

### 2.2.5. Flutter

Flutter merupakan kerangka kerja UI seluler Google untuk membuat antarmuka asli berkualitas tinggi pada iOS dan Android dalam waktu singkat. Flutter bekerja dengan kode yang sudah ada, digunakan oleh pengembang dan organisasi di seluruh dunia, bersifat *free* dan *open source* [19].

### **HTTP Package**

*Package* ini berisi serangkaian fungsi dan kelas tingkat tinggi yang membuatnya mudah untuk mengkonsumsi sumber daya HTTP. *Platform-independent*, dan dapat digunakan pada baris perintah dan browser [20].

### **Oscilloscope Package**

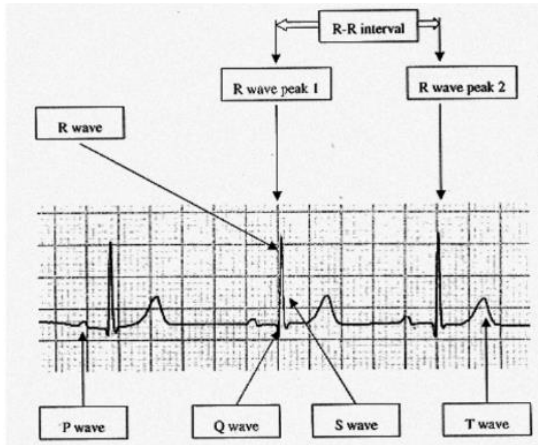
Oscilloscope merupakan *widget package* untuk flutter yang menampilkan grafis mirip dengan jejak pada osiloskop yang akan menampilkan nilai saat bergulir di layar. *Widget* ini menggunakan List <double> sebagai sumber data untuk ditampilkan dan akan menskalakan informasi agar sesuai dengan tampilan. Setiap kali nilai ditambahkan ke *dataset* titik lain diplot pada jejak, jejak ini akan meluas dari kiri ke kanan melintasi layar sampai mencapai akhir. Pada titik ini jejaknya kemudian bergulir untuk menunjukkan nilai-nilai baru [21].

#### **2.2.6. Electrocardiogram**

*Electrocardiogram* (ECG) merupakan suatu ukuran sinyal yang menunjukkan bagaimana aktivitas listrik dari otot jantung berubah dari waktu ke waktu selama setiap siklus jantung [7]. ECG ini diambil menggunakan elektrokardiograf yang ditampilkan melalui monitor atau dicetak pada kertas yang merupakan rekaman informasi kondisi jantung. Rekaman inilah yang biasa dipakai oleh dokter untuk menentukan kondisi jantung seorang pasien [22].

### **Heart Rate Variability**

*Heart Rate Variability* (HRV) merupakan variasi temporal antara urutan detak jantung berturut-turut yang dihasilkan karena perubahan terus menerus pada keseimbangan parasimpatis simpatis [6]. Pada ECG normal, defleksi ke atas maksimum dari kompleks QRS normal adalah puncak gelombang R, dan durasi antara dua puncak gelombang R yang berdekatan disebut sebagai interval R-R. Sinyal ECG memerlukan pengeditan sebelum analisis HRV dapat dilakukan, suatu proses yang membutuhkan penghapusan semua denyut yang berasal dari simpul sinus [6].



Gambar 2.5. Sinyal Electrocardiogram

HRV merupakan salah satu penanda yang menjanjikan, kemudahan dari derivasi dalam pengukuran ini telah mempopulerkan penggunaannya. Karena banyak perangkat komersial sekarang menyediakan pengukuran HRV secara otomatis, ahli jantung telah menyediakan dengan alat sederhana untuk penelitian dan studi klinis [23]. HRV diatur oleh sistem saraf otonom (parasimpatis simpatis) dan dipengaruhi oleh beberapa hal seperti umur, kondisi fisik dan stres [24].

Terdapat beberapa metrik HRV dapat digunakan untuk analisis ialah [25] :

- a. Mean, nilai rata-rata dari RR Interval
- b. SDNN (*Standard Deviation of NN*), nilai standar deviasi dari RR interval
- c. SDDS, nilai standar deviasi perbedaan antara RR interval yang berdekatan.
- d. RMSSD (*Root Mean Square of Successive Difference*), nilai akar kuadrat dari rata-rata jumlah kuadrat perbedaan antara RR Interval
- e. pNN20, nilai persentase jumlah perbedaan RR interval berturut-turut yang lebih besar dari 20 ms.

- f. pNN50, nilai persentase jumlah perbedaan RR interval berturut-turut yang lebih besar dari 50 ms
- g. LF (*Low Frquency*), Frekuensi yang berada pada antara 0,05 Hz sampai 0,15 Hz
- h. HF (*High Frequency*), Frekuensi yang berada pada antara 0,15 Hz sampai 0,4 Hz
- i. LF/HF, nilai LF dibagi dengan HF

### 2.2.7. Flask

Flask merupakan sebuah *web application framework* yang ditulis dalam bahasa pemrograman python yang dikembangkan oleh *Armin Ronacher*. *Web application framework* adalah kumpulan *library* dan *module* yang memungkinkan pengembang untuk membuat aplikasi website tanpa perlu untuk menghiraukan *low-level details* seperti protokol, *thread management*, dan lain-lain [26].

### QRSDetector

*QRSDetector* merupakan *library* dengan bahasa pemrograman python untuk mendeteksi titik puncak pada sinyal elektrokardiogram. Library ini menghasilkan deteksi titik puncak berada pada posisi ke berapa. Agar dapat dirubah menjadi HRV harus mencari interval dari titik puncak yang terdeteksi dalam satuan *milliseconds* (ms) [27].

### Hrvanalysis

*Hrvanalysis* merupakan *library* dengan bahasa pemrograman python untuk menghitung metrik HRV. Dengan *input* HRV, library ini dapat menghasilkan output analisis berupa *time domain* dan *frequency domain* [28].

### 2.2.8. Testing

*Testing* merupakan elemen penting dari siklus hidup pengembangan perangkat lunak., disebut juga sebagai kontrol kualitas perangkat lunak atau jaminan kualitas perangkat lunak. Tujuan dari testing adalah untuk memvalidasi dan memverifikasi dari aplikasi yang dibuat. Yang dimaksud validasi adalah apakah aplikasi yang dibuat sudah sesuai, sedangkan verifikasi

adalah apakah kode program, model, desain diagram, serta kebutuhan telah memenuhi spesifikasi [29]. Terdapat berbagai metode *testing* yang dapat digunakan dalam pengembangan perangkat lunak [30] :

- a. *Unit Tests*. Suatu pengujian untuk aspek terkecil dari sistem (function, unit, modul, komponen) yang dibuat apakah berjalan sesuai dengan yang diharapkan.
- b. *Integration Tests*. Suatu pengujian yang dilakukan ketika dua atau lebih unit yang diuji digabungkan ke dalam struktur yang lebih besar. Tes ini sering dilakukan pada kedua antarmuka antara komponen dan struktur yang lebih besar dibangun.
- c. *System Test*. Pengujian yang dilakukan oleh pengembang cenderung untuk menegaskan kualitas dari ujung ke ujung dari keseluruhan sistem. Tes ini didasarkan pada spesifikasi fungsional atau persyaratan sistem.
- d. *Acceptance Test*. Tes ini dilakukan ketika sistem selesai diserahkan dari pengembang ke pengguna. Tujuannya untuk mengecek apakah sistem yang dibuat telah memenuhi kebutuhan atau tidak.

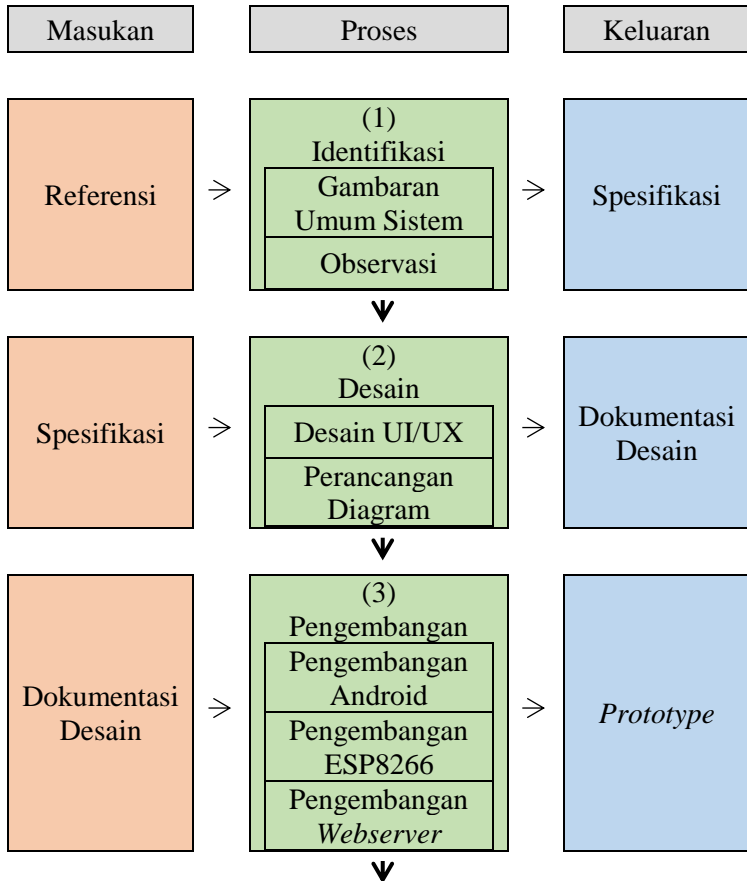
### **Blackbox Testing**

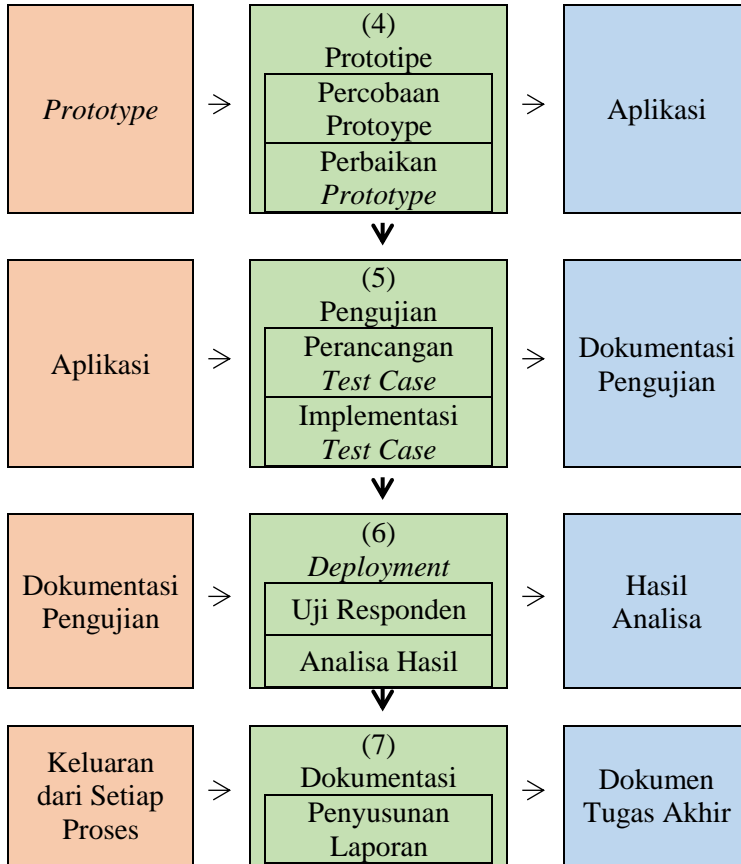
*Blackbox Testing* merupakan ujicoba yang berfokus pada kebutuhan fungsional perangkat lunak. Pengujian ini memungkinkan developer untuk membuat himpunan berbagai kondisi *input* yang akan melaith seluruh syarat-syarat fungsional suatu perangkat lunak. Pengujian ini berusaha untuk menemukan beberapa kesalahan seperti fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan struktur data atau akses database, kesalahan performa, serta kesalahan inisialisasi dan terminasi [31].



### BAB III METODOLOGI

Pada bagian ini akan menjelaskan mengenai tahapan yang dilakukan dalam pengerjaan tugas akhir menggunakan *software development life cycle* (SDLC) untuk pengembangan mobile (*Mobile Application Development Cycle*) yang telah disesuaikan beserta deskripsi dan penjelasannya.





Gambar 3.1. Metodologi

### 3.1. Tahap Identifikasi

Pada tahap ini akan dilakukan proses identifikasi mengenai kebutuhan dari sistem aplikasi yang akan dibuat. Hasil pada tahap ini nantinya akan menjadi data penunjang yang akan digunakan pada tahapan penelitian selanjutnya.

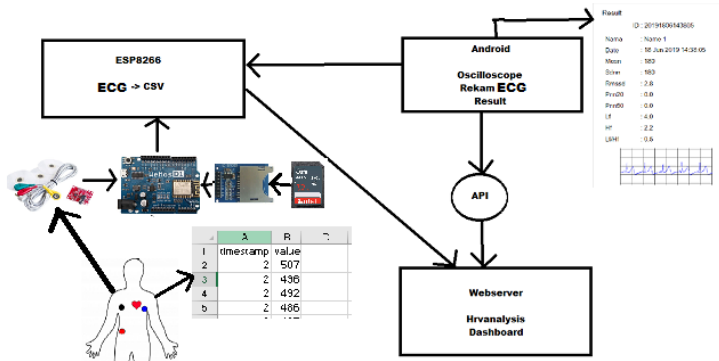
#### 3.1.1. Referensi

Referensi ini merupakan data masukan pertama dari dimulainya pengerjaan tugas akhir. Referensi yang dimaksud disini adalah merupakan informasi yang berguna serta hasil dari penelitian-

penelitian sebelumnya yang dihimpun kemudian digunakan sebagai landasan untuk pengerjaan tugas akhir ini. Ditambah juga dengan referensi kode program sejenis yang telah ada untuk dimanfaatkan pada kegiatan observasi.

### 3.1.2. Gambaran Umum Sistem

Sistem yang akan dibuat terdiri dari 3 sub-sistem yang saling terintegrasi yaitu ESP8266, Android, dan *webserver*. ESP8266 berperan sebagai pengambilan sinyal ECG yang disimpan dalam bentuk csv kemudian dikirimkan ke *webserver* untuk dianalisis. *Webserver* berperan sebagai sistem yang menganalisis metrik HRV, dan android sebagai sistem yang berinteraksi dengan pengguna.



Gambar 3.2. Gambaran umum sistem

### 3.1.3. Observasi

Pada tahap ini akan dilakukan observasi terhadap struktur kode program yang telah diimplementasikan pada masing-masing sistem, meliputi mekanisme jalannya kode program untuk membuat data yang dihasilkan dapat di *transfer* antar sistem.

### 3.1.4. Spesifikasi

Spesifikasi tugas akhir merupakan hasil keluaran yang didapat dari tahap identifikasi ini. Entitas ini akan mendefinisikan cakupan pengembangan aplikasi pada tugas akhir serta teknologi yang dipakai dalam pengembangan produk tugas

akhir ini. Spesifikasi tugas akhir akan digunakan sebagai masukan dalam proses pertama pada tahap desain sistem.

### **3.2. Tahap Desain**

Pada tahap ini dilakukan perancangan dan desain aplikasi pengukur tingkat stres. Perancangan dan desain menggunakan *Unified Modeling Language* (UML) serta desain pengalaman pengguna (*User Experience*) dan desain tampilan (*User Interface*).

#### **3.2.1. Desain UI/UX**

Pada tahap ini dilakukan desain melalui wireframe atau *mockup* menggunakan salah satu *toolkit* desain yang sudah ada. Desain dari UX ini dirancang tanpa perlu memperhatikan desain tampilan yang sebenarnya, hal yang terpenting dari desain UX adalah mempertimbangkan pedoman antarmuka yang ditargetkan oleh aplikasi, dalam penelitian tugas akhir ini adalah android. Setelah desain UX ditentukan, langkah selanjutnya adalah membuat desain UI. Sementara desain UX hanya berupa *mockup* hitam putih, fase pada desain UI ini adalah tempat dimana warna, grafik, dan lain-lain diselesaikan.

#### **3.2.2. Perancangan Diagram**

Pada tahapan ini akan dilakukan pembuatan diagram-diagram untuk pengembangan aplikasi pengukur tingkat stres yang disesuaikan oleh kebutuhan dari tahapan sebelumnya.

#### **3.2.3. Dokumentasi Desain**

Dokumentasi desain ini merupakan hasil dari tahap desain sistem yang akan digunakan dalam proses pertama pada tahap pengembangan.

### **3.3. Tahap Pengembangan**

Pengembangan merupakan tahapan pembuatan kode program berdasarkan desain sistem yang telah disepakati, meliputi pengembangan 3 sub-sistem yang telah dirancang.

### **3.3.1. Pengembangan Android**

Pada tahap ini dilakukan pembuatan kode program android menggunakan framework flutter dengan *package* utama *Oscilloscope* dan *Http*

### **3.3.2. Pengembangan ESP8266**

Pada tahap ini dilakukan pembuatan kode program pada ESP8266 menggunakan tipe Wemos D1 R2 dengan penambahan modul AD8232 untuk mengambil sinyal ECG dan Modul SD Card untuk menyimpan data sinyal ECG dalam bentuk csv

### **3.3.3. Pengembangan Webserver**

Pada tahap ini dilakukan pembuatan kode program pada webserver menggunakan framework flask dengan bahasa pemrograman python disertai dengan library utama QRSDetector untuk mencari nilai HRV dan Hrvanalysis untuk menganalisis metrik HRV

### **3.3.4. Prototipe**

Entitas ini merupakan *output* dari tahap pengembangan, sistem yang dihasilkan sesuai desain sistem yang telah dibuat pada tahap sebelumnya.

## **3.4. Tahap Prototipe**

Prototipe merupakan tahapan percobaan fungsionalitas yang telah dibuat kemudian melakukan perbaikan apabila terdapat fungsionalitas yang tidak berjalan hingga semua fungsionalitas terpenuhi

### **3.4.1. Percobaan Prototipe**

Pada tahap ini dilakukan percobaan pada aplikasi sederhana untuk mengetahui apakah fungsionalitas pada masing-masing sistem dapat berjalan atau tidak

### **3.4.2. Perbaikan Prototipe**

Pada tahap ini dilakukan fungsionalitas apabila terdapat fungsi yang masih tidak dapat berjalan.

### **3.4.3. Aplikasi**

Entitas ini merupakan bentuk aplikasi setiap sistem sudah saling terintegrasi yang siap digunakan dari awal hingga akhir untuk selanjutnya dilakukan tahap pengujian.

## **3.5. Tahap Pengujian**

Pada tahapan ini, akan dilakukan pengujian terhadap sistem yang telah dibangun dengan metode *blackbox testing*

### **3.5.1. Perancangan Pengujian**

Pada tahapan ini akan dirancang skenario pengujian untuk dilakukan pada tahapan berikutnya. Adapun perencanaan skenario pengujian berdasarkan fungsional kode program, meliputi seluruh fungsi yang telah dibangun.

### **3.5.2. Implementasi Pengujian**

Setelah rancangan skenario pengujian telah selesai maka akan dilakukan *blackbox testing* dengan menggunakan skenario pengujian yang telah dibuat sebelumnya.

### **3.5.3. Dokumentasi Pengujian**

Pada tahap ini merupakan *output* dari *blackbox testing* yang telah dilakukan.

## **3.6. Tahap Deployment**

Pada tahap ini dilakukan pengujian sistem yang telah dibuat kepada responden

### **3.6.1. Uji Responden**

Pengujian kepada responden dilakukan kepada 10 responden sesuai kriteria dengan metode validasi NASA TLX

### **3.6.2. Analisa Hasil**

Pada tahap ini dilakukan analisa terhadap klasifikasi tingkat stres berdasarkan metrik HRV dengan validasi metode NASA TLX menggunakan data *train*

### **3.6.3. Hasil Analisa**

Entitas ini merupakan luaran dari tahap analisa berupa tingkat akurasi pengklasifikasian tingkat stres berdasarkan metrik HRV pada data *testing*

## **3.7. Tahap Dokumentasi**

Pada tahapan ini, laporan tugas akhir akan dibuat yang akan mendokumentasikan setiap langkah yang telah dilakukan, hasil yang dihasilkan setiap langkah, serta kesimpulan dan saran untuk penelitian kedepannya.

### **3.7.1. Penyusunan Laporan**

Pada tahapan terakhir ini akan dilakukan pembuatan laporan dalam bentuk buku tugas akhir yang disusun sesuai format yang telah ditentukan. Buku ini berisi dokumentasi langkah-langkah pengerjaan tugas akhir secara rinci. Buku ini diharapkan dapat bermanfaat sebagai referensi untuk pengerjaan penelitian lain, serta sebagai acuan untuk pengembangan lebih lanjut terhadap topik pengembangan yang serupa.

### **3.7.2. Dokumen Tugas Akhir**

Adapun dokumen tugas akhir merupakan luaran dari keseluruhan proses penelitian yang telah dilakukan sesuai metodologi.

*Halaman ini sengaja dikosongkan*



## **BAB IV PERANCANGAN**

Pada bagian ini akan menjelaskan mengenai proses perancangan terhadap sistem aplikasi yang akan digunakan sebagai acuan dalam proses implementasi.

### **4.1. Analisis Kebutuhan**

Pada tahap pertama dari perancangan dimulai dengan mengumpulkan informasi terkait pembuatan aplikasi mengenai kondisi dan teknologi yang digunakan sistem aplikasi *existing* pengukur tingkat stres. Pengambilan informasi dilakukan dengan menggunakan dua buah metode, yaitu referensi dan observasi.

#### **4.1.1. Referensi**

Pada tahap ini dilakukan pengumpulan informasi-informasi berdasarkan penelitian sebelumnya, yang digunakan sebagai landasan untuk merancang sistem aplikasi pengukur tingkat stres. Telah ditemukan beberapa metode untuk mengukur tingkat stres yakni secara objektif dan subjektif. Secara objektif, mengukur tingkat stres dapat menggunakan *heart rate variability* yang diperoleh dari sinyal detak jantung. Sinyal detak jantung ini didapatkan dari sensor ECG.

Untuk pengambilan sinyal elektrokardiogram saat ini masih menggunakan kabel dan juga analisa *heart rate variability* dilakukan pada sistem yang berbeda belum terintegrasi sehingga tidak *realtime*. Terdapat juga pengiriman sinyal elektrokardiogram yang *wireless* menggunakan arduino namun untuk analisa *heart rate variability* tetap menggunakan sistem aplikasi berbeda yang belum terintegrasi.

Untuk itu pada tugas akhir ini dikembangkan sistem aplikasi yang dapat mengukur tingkat stres secara *wireless* dengan memanfaatkan analisis *heart rate variability* yang terintegrasi dengan 3 sub-sistem yakni android, ESP8266, dan Server yang dikembangkan menggunakan bahasa pemrograman python.

Berdasarkan beberapa referensi yang ditemukan, didapat beberapa kebutuhan fungsional dan non fungsional pada sistem aplikasi yang akan dikembangkan. Berikut merupakan kebutuhan dari pengembangan sistem aplikasi pengukur tingkat stres berbasis android:

### 1. Kebutuhan Fungsional

Tabel 4.1. Kebutuhan fungsional

Kode	Fungsional	Deskripsi
F1	Monitoring Sinyal ECG	Sistem dapat melakukan <i>monitoring</i> sinyal ECG berbentuk <i>oscilloscope</i> pada sistem <i>mobile</i> (android) yang terintegrasi dengan sensor ECG pada sistem ESP8266
F2	Merekam Sinyal ECG	Sistem dapat merekam sinyal ECG yang didapat dari sensor ECG pada sistem ESP8266 kemudian disimpan pada database MySQL dengan akses API pada sistem <i>webserver</i>
F3	Menganalisa Sinyal ECG	<i>Sistem webserver</i> dapat melakukan analisa sinyal ECG dengan memanfaatkan HRV untuk mendeteksi tingkat stres
F4	Melihat Hasil Analisa	Sistem <i>webserver</i> dan sistem android dapat menampilkan hasil analisa yang telah disimpan pada database MySQL yang diakses menggunakan API pada <i>webserver</i>

## 2. Kebutuhan Non Fungsional

Tabel 4.2. Kebutuhan non fungsional

Kode	Non Fungsional	Deskripsi
NF1	Integrasi	Sistem aplikasi terdiri dari 3 sub-sistem yang saling terintegrasi yakni Android, ESP8266, dan <i>Webserver</i>
NF2	Performa	Sistem memiliki akses yang cepat, stabil, dan akurat.

### 4.1.2. Observasi

Metode ini dilakukan dengan mengamati sistem aplikasi *existing* pengukur tingkat stres beserta kode program yang telah ada. Dari hasil observasi yang dilakukan didapatkan tumpukan teknologi yang akan digunakan untuk pengembangan sistem aplikasi pengukur tingkat stres ini, dapat dilihat pada pada Tabel 4.3.

Tabel 4.3. Tumpukan teknologi

No	Teknologi	Rincian
1	<i>Web Server</i>	Werkzeug
2	Bahasa Pemrograman	Python
3	<i>Framework</i>	Flask
4	<i>Database</i>	MySQL
5	<i>Platform</i>	Android
6	<i>Framework</i>	Flutter
7	Bahasa Pemrograman	Dart
8	<i>ESP8266</i>	Wemos D1 R2

Dari hasil referensi dan observasi yang dilakukan, dibuat pemetaan fitur sesuai dengan sistem yang akan digunakan untuk mengembangkan sistem aplikasi pengukur tingkat stres berdasarkan kebutuhan fungsional. Adapun tabel pemetaan yang dihasilkan dapat dilihat pada Tabel 4.4.

Tabel 4.4. Pemetaan kebutuhan fungsional

Sistem	Fitur
ESP8266	Mengirim satuan nilai sinyal ECG
	Merekam sinyal ECG dalam bentuk csv
	Mengirim csv
WebsERVER	Menerima sinyal ECG dalam bentuk csv
	Mendeteksi HRV dari file csv
	Menghitung (menganalisa) metrik HRV dari HRV yang telah dihasilkan
	Menyimpan hasil analisa
	Menampilkan hasil analisa
Android	Menerima satuan sinyal ECG dalam bentuk <i>oscilloscope</i>
	Melakukan tes
	Menampilkan hasil tes

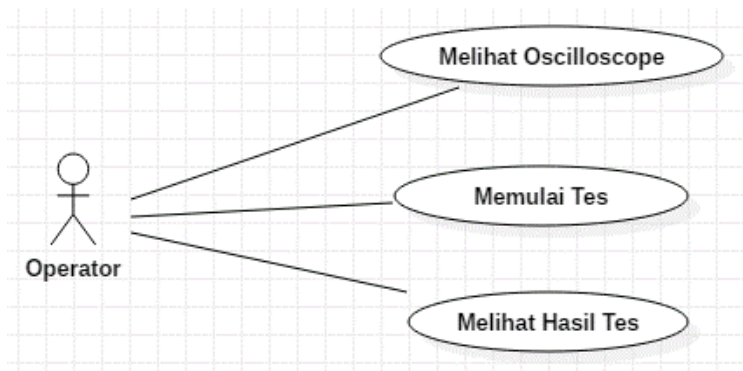
## 4.2. Desain Sistem

Desain sistem dibuat untuk menjadi acuan dari pengembangan sistem aplikasi pengukur tingkat stres. Desain sistem dibuat setelah didapatkan kebutuhan fungsional dan non fungsional yang telah dirancang, beserta pemetaan fitur pada masing-masing sistem yang akan dikembangkan. Desain sistem dilakukan dengan beberapa tahapan yang dibuat secara terstruktur untuk memudahkan proses pengembangan sistem aplikasi pengukur tingkat stres.

### 4.2.1. Perancangan Use Case

Desain *use case* dibuat dari hasil kebutuhan fungsional yang diiriskan dengan fitur yang akan dikembangkan. *Use case*

bertujuan untuk menunjukkan interaksi antara aktor dan sistem aplikasi pengukur tingkat stres. Adapun *use case* yang dibuat untuk sistem aplikasi pengukur tingkat stres ini, seperti pada Gambar 4.1.



Gambar 4.1. Use case aplikasi pengukur tingkat stres

Berdasarkan *use case* tersebut, nantinya sistem aplikasi pengukur tingkat stres ini digunakan oleh *operator* untuk mengukur tingkat stres seorang pasien yang dijalankan melalui *interface* pada sistem android. Nantinya sistem android ini saling berintegrasi dengan sistem ESP8266 dan sistem server yang dikembangkan dengan bahasa pemrograman python. Hingga terbentuklah suatu kesatuan sistem pengukur tingkat stres yang dapat memonitoring sinyal detak jantung (*oscilloscope*), melakukan tes tingkat stres, dan melihat hasilnya.

Selain pembuatan *use case*, juga dilakukan pembuatan deskripsi yang berisikan detail *behavior* dari setiap *use case* beserta pola interaksinya. *Use case description* pada pengembangan sistem pengukur tingkat stres dibagi menjadi 3 bagian sesuai dengan *use case* yang telah dibuat sebelumnya. Adapun *use case description* yang dibuat, seperti pada Tabel 4.5 hingga 4.7.

Tabel 4.5. Use case description melihat oscilloscope

<b>U1</b> <i>Use Case Name : Melihat Oscilloscope</i>	
<i>Primary Actor</i>	Operator
<i>Brief Description</i>	<i>Use case ini digunakan oleh aktor pada sistem android untuk monitoring sinyal ECG yang didapat dari sensor ECG pada sistem ESP8266.</i>
<i>Pre Condition :</i>	Sistem ESP8266 telah siap
<i>Basic Course :</i>	Operator menekan tab oscilloscope pada <i>bottom bar interface</i> sistem android. Sistem android akan menampilkan sinyal ECG yang diperoleh dari sistem ESP8266 dalam bentuk osiloskop.
<i>Alternate Course :</i>	<i>Oscilloscope pada Interface</i> sistem android tidak berjalan apabila sinyal ECG tidak didapatkan dari sistem ESP8266.

Tabel 4.6. Use case description memulai tes

<b>U2</b> <i>Use Case Name : Memulai Tes</i>	
<i>Primary Actor</i>	Operator
<i>Brief Description</i>	<i>Use case ini digunakan oleh aktor pada sistem android untuk mulai merekam sinyal ECG pada sistem ESP8266 dan memberhentikannya yang kemudian dikirim kepada sistem webserver python untuk dianalisis HRV nya. Kemudian hasilnya disimpan pada database MySQL.</i>
<i>Pre Condition :</i>	Sistem ESP8266 dan <i>Webserver</i> telah siap
<i>Basic Course :</i>	Aktor menekan tab ECG pada <i>bottom bar interface</i> sistem android. Aktor menekan tombol “START” untuk mulai merekam sinyal ECG. Waktu akan berjalan selama 5 menit,

kemudian menekan tombol “STOP”. Setelah itu Aktor memasukkan nama pasien, dan klik tombol “Analyze”, sistem *webservice* akan melakukan perhitungan dan menyimpan hasilnya.

*Alternate Course :*

*Interface* sistem pada android akan menampilkan notifikasi ketika menekan tombol “START” apabila sistem ESP8266 belum siap. *Interface* android akan menampilkan notifikasi apabila nama belum diisi ketika menekan tombol “Analyze” dan ketika *webservice* belum siap.

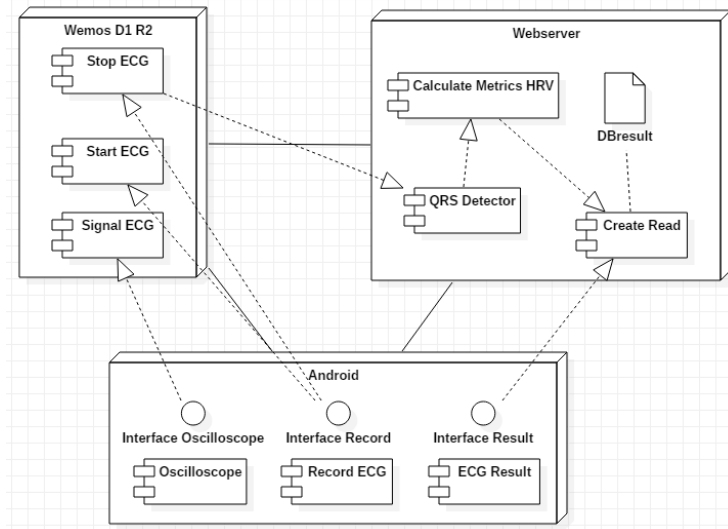
Tabel 4.7. Use case description melihat hasil tes

<b>U3</b>	<i>Use Case Name</i> : Melihat Hasil Tes
<i>Primary Actor</i> Operator	
<i>Brief Description</i> Use case ini digunakan oleh aktor pada sistem android dan sistem <i>webservice</i> dalam bentuk website untuk melihat hasil tes yang telah dilakukan.	
<i>Pre Condition</i> : Sistem <i>Webserver</i> telah siap	
<i>Basic Course</i> : Aktor menekan tab Hasil pada sistem <i>interface</i> android atau mengakses url <a href="http://localhost:5000/reads">http://localhost:5000/reads</a> pada <i>webservice</i> .	
<i>Alternate Course</i> : <i>Interface</i> pada sistem android menampilkan notifikasi apabila data hasil tidak dapat dimuat.	

#### 4.2.2. Perancangan Arsitektur

Arsitektur sistem aplikasi pengukur tingkat stres yang akan dikembangkan menggunakan 3 sub-sistem yakni *mobile* dengan sistem operasi android, ESP8266 dengan tipe Wemos D1 R2, dan Server yang dikembangkan menggunakan bahasa pemrograman Python. Masing-masing sistem mempunyai peran atau fitur masing-masing dalam membentuk suatu kesatuan sistem pengukur tingkat stres yang terintegrasi. Secara garis besar, sistem android berperan dalam hal *interface* karena

berhubungan langsung dengan aktor yang menjalankan sistem aplikasi ini. Untuk sistem pada Wemos D1 R2 berperan untuk mengambil sinyal detak jantung menggunakan sensor ECG. Untuk server python digunakan untuk proses analisa *heart rate variability* dari sinyal detak jantung yang diperoleh dari sistem Wemos D1 R2, kemudian hasil analisa tersebut disimpan menggunakan database MySQL. Untuk lebih jelasnya, berikut merupakan pola aritektur yang akan dibuat dapat dilihat pada Gambar 4.2.

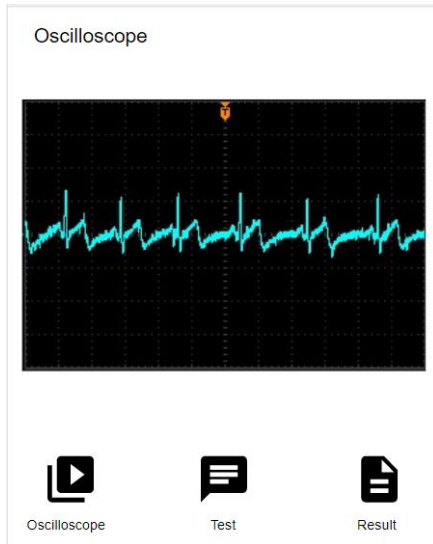


Gambar 4.2. Deployment diagram

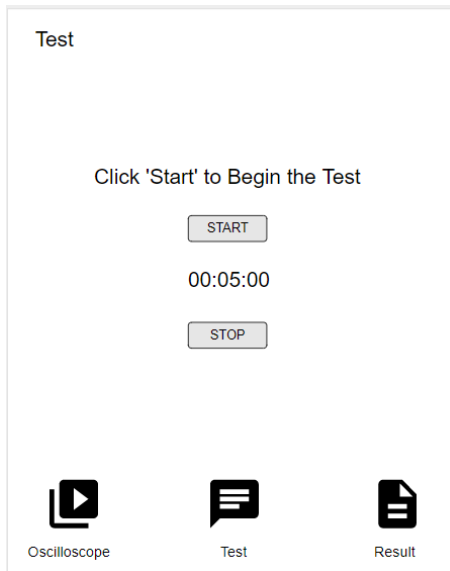
#### 4.2.3. Perancangan Desain UI/UX

Secara garis besar, desain UI/UX pada aplikasi pengukur tingkat stres ini terdiri dari 3 *button bottom bar* sesuai dengan 3 fungsi utama yang ada pada *use case*. Desain mockup untuk pengembangan aplikasi pengukur tingkat stres ini dapat dilihat pada Gambar 4.3 sampai 4.7.





Gambar 4.3. Mockup oscilloscope



Gambar 4.4. Mockup test

Test

Please Wait until the lamp of Wemos Turn Off





Please Input Name the Patient

Clík 'Analyze' to begin analyze HRV

And Please Wait ...  
Test Result Will be Display  
at the Result Tab

*Gambar 4.5. Mockup analyze*

### Result

▼ Name	▼ Date	▼ Mean	▼ Sdnn
Name 1 	18 Jun 2019 20:15:30	180	2.3
Name 2 	19 Jun 2019 21:32:43	158	2.5
Name 3 	20 Jun 2019 19:43:50	201	2.5
Name 4 	21 Jun 2019 20:30:12	167	1.8



Oscilloscope

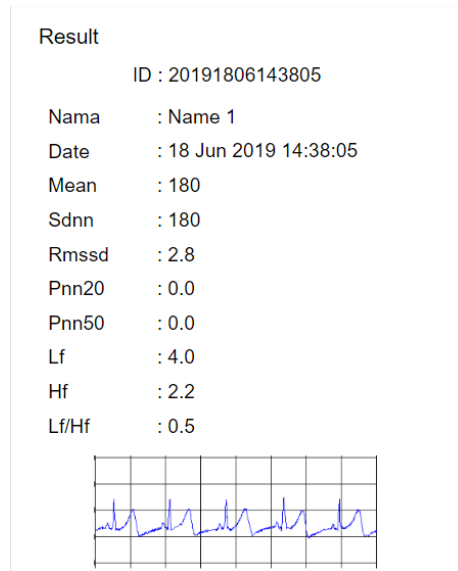


Test



Result

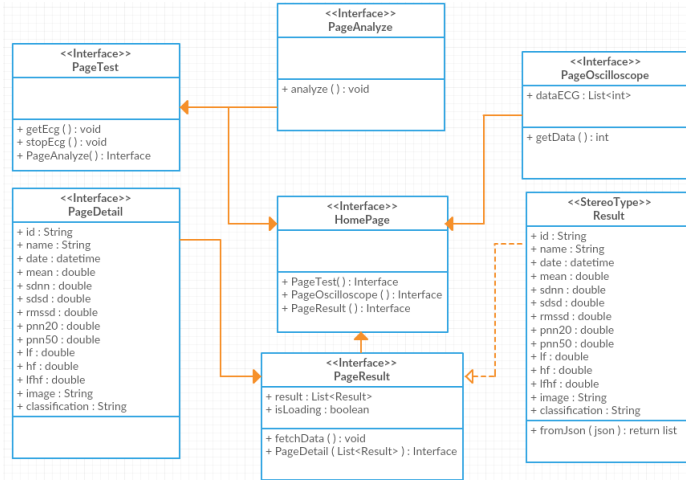
*Gambar 4.6. Mockup result*



*Gambar 4.7. Mockup detail result*

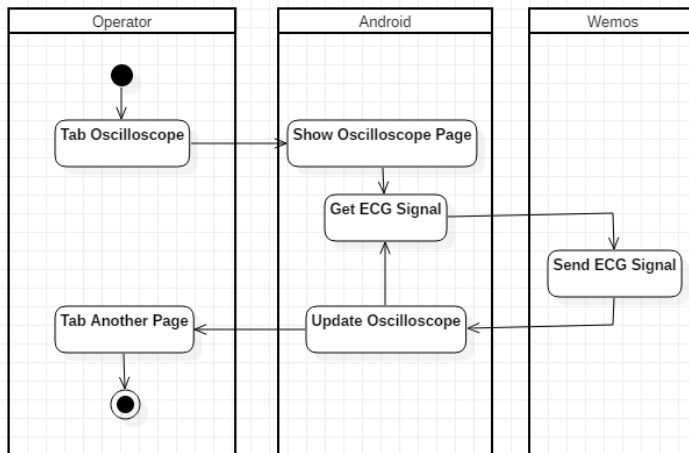
#### **4.2.4. Perancangan Sistem Mobile**

Pada sistem Android terdiri dari 3 fitur yakni monitoring sinyal ECG, melakukan tes, dan menampilkan hasil tes. Untuk membuat fitur tersebut, dibentuklah beberapa class yang digambarkan pada class diagram pada gambar 4.8.

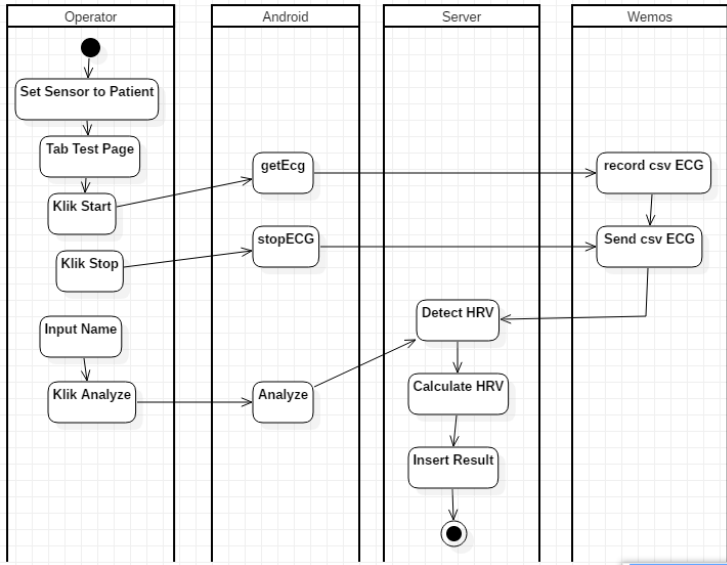


Gambar 4.8. Class diagram pada sistem android

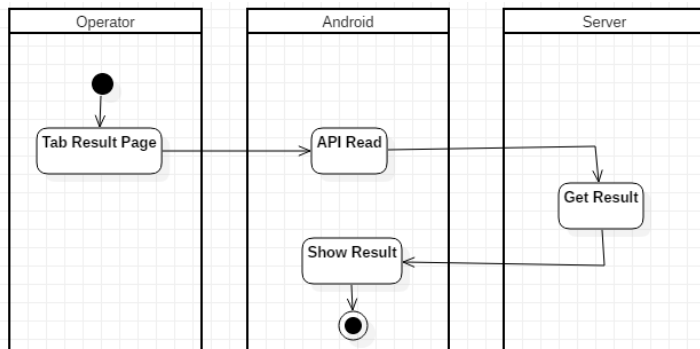
Untuk memperjelas aktifitas yang dilakukan pada masing-masing fitur tersebut, dijelaskan pada *activity diagram* diagram yang bisa dilihat pada Gambar 4.9 sampai 4.11



Gambar 4.9. Activity diagram monitoring sinyal ECG



Gambar 4.10. Activity diagram melakukan tes

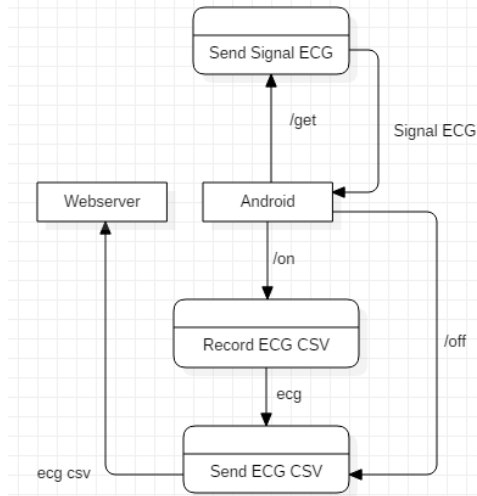


Gambar 4.11. Activity diagram melihat hasil tes

#### 4.2.5. Perancangan Sistem ESP8266

Pada sistem ESP8266 terdapat 3 fitur yakni mengirim sinyal satuan ECG, merekam sinyal ECG dalam bentuk csv, dan mengirimkan sinyal ECG dalam bentuk csv tersebut.

Untuk fitur mengirim sinyal ECG dalam bentuk satuan digunakan oleh sistem android untuk *monitoring* sinyal detak jantung dalam bentuk *oscilloscope*, secara garis besar cara kerja fitur ini adalah android mengakses URI “/get” untuk mengambil nilai sinyal ECG secara terus menerus yang kemudian nilai tersebut ditampilkan dalam bentuk *oscilloscope* pada *interface* android. Sistem ESP8266 akan mulai merekam sinyal ECG dalam bentuk csv ketika sistem android mengakses URI “/on”, dan akan berhenti merekam dan mengirimkannya kepada webserver ketika sistem android mengakses URI “/off”. Berikut gambar *data flow* diagram dapat dilihat pada gambar 4.12.



Gambar 4.12. Data flow diagram ESP8266

Berikut merupakan daftar URI beserta deskripsi fungsinya yang terdapat pada sistem ESP8266, dapat dilihat pada table 4.9

Tabel 4.8. URI pada sistem ESP8266

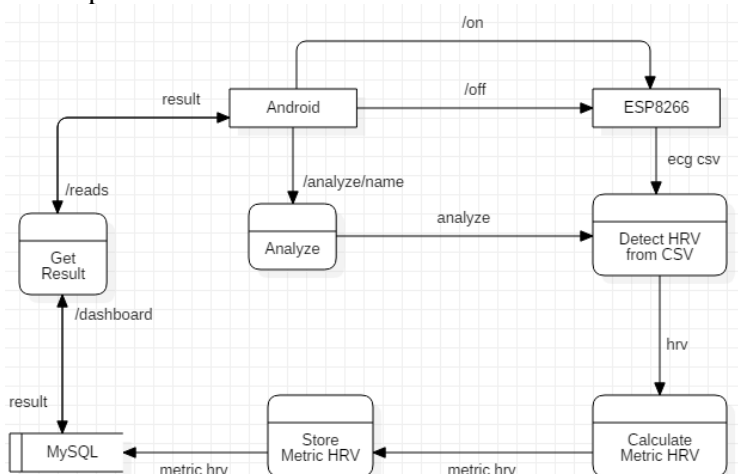
Method	URI	Deskripsi
GET	/get	Diakses oleh sistem android untuk mengambil sinyal ECG dalam bentuk satuan
GET	/on	Diakses oleh sistem android untuk mulai merekam sinyal ECG dalam bentuk csv

GET	/off	Diakses oleh sistem android untuk memberhentikan perekaman sinyal ECG dalam bentuk csv dan mengirimkannya ke webserver
-----	------	------------------------------------------------------------------------------------------------------------------------

#### 4.2.6. Perancangan Sistem *Webserver*

Pada sistem webserver ini (menggunakan bahasa pemrograman python) terdapat 6 fitur yakni menerima sinyal ECG dalam bentuk csv, mendeteksi HRV dari csv, menghitung atau menganalisa HRV, menyimpan hasilnya, dan menampilkannya.

Secara garis besar peran pada sistem webserver ini adalah sebagai penghubung antara sistem android dan sistem ESP8266 untuk melakukan tes. Serta proses utama dari tes tersebut yakni menerima sinyal detak jantung, menganalisa, meyimpan, dan menampilkannya. Berikut gambar *data flow diagram* dapat dilihat pada Gambar 4.13



Gambar 4.13. *Data flow diagram webserver*

Pada sistem server python ini menggunakan database untuk menyimpan hasil analisa yang MySQL. Dalam sistem ini hanya menggunakan 1 table yakni tabel "results" dengan daftar yang dapat dilihat pada tabel 4.9.

Tabel 4.9.. Kolom pada tabel database

No	Nama Kolom	Tipa Data
1	id	Varchar(14) PK
2	Name	Varchar(100)
3	Date	Datetime
4	Mean	Float(10,2)
5	Sdnn	Float(10,2)
6	Sdsd	Float(10,2)
7	Rmssd	Float(10,2)
8	Pnn20	Float(10,2)
9	Pnn50	Float(10,2)
10	Lf	Float(10,2)
11	Hf	Float(10,2)
12	Lfhf	Float(10,2)
13	File	Varchar(150)
14	image	Varchar(150)
15	Classification	Varchar(10)

Berikut merupakan daftar URI beserta deskripsi fungsinya yang terdapat pada sistem ESP8266, dapat dilihat pada table 4.10.

Tabel 4.10. URI pada server python

Method	URI	Deskripsi
GET	/analyze/<name>	Diakses oleh android untuk mulai menganalisa csv ecg yang telah dikirim oleh sistem ESP8266, dengan mengirimkan paramater name (nama dari subjek) kemudian menyimpan hasilnya ke database MySQL.
GET	/dashboard	Diakses oleh webserver python untuk menampilkan <i>dashboard</i> dari hasil tes



GET	/reads	Diakses oleh sistem android untuk menampilkan <i>dashboard</i> dari hasil tes
-----	--------	-------------------------------------------------------------------------------

### 4.3. Perancangan *Test Case*

Pada tahap ini merupakan tahapan dimana skenario untuk melakukan pengujian pada setiap *use case* yang dibuat. *Test case* dibuat berdasarkan metode *blackbox testing*. Tujuan dari pembuatan *test case* adalah untuk menghasilkan berbagai kemungkinan pada semua fungsi yang dibuat, apakah dapat berjalan dengan baik atau tidak. *Setiap use case* memiliki *test case* dengan skenario pengujian yang berbeda.

Skenario pengujian didasari oleh berhasil atau tidaknya fitur berdasarkan pemetaan fungsional yang telah dibuat sebelumnya. Berikut skenario beserta ekspektasi dari perancangan *test case* yang dibuat, seperti pada Tabel 4.11.

Tabel 4.11. Perancangan *test case*

Use Case	Kategori	Skenario
U1	Normal	Sistem ESP8266 telah siap dan Sistem android pada tab <i>oscilloscope</i>
	Alternatif 1	Sistem ESP8266 belum siap dan Sistem android pada tab <i>oscilloscope</i>
U2	Normal	Sistem android akses URI /getecg dan Sistem ESP8266 telah siap
	Normal	Sistem android akses URI /stopecg dan Sistem ESP8266 telah siap
	Normal	Sistem android akses URI /analyze/<name> dengan mengisi <i>field name</i> dan sistem webserver telah siap
	Alternatif 1	Sistem android akses URI /getecg dan Sistem ESP8266 belum siap
	Alternatif 2	Sistem android akses URI /stopecg dan Sistem ESP8266 belum siap

	Alternatif 3	<i>Sistem android akses URI /analyze/&lt;name&gt; tanpa mengisi field name</i>
	Alternatif 4	<i>Sistem android akses URI /analyze/&lt;name&gt; dengan mengisi field name dan webserver belum siap</i>
U3	Normal	Sistem android akses URI /reads dan webserver telah siap
	Alternatif 1	Sistem android akses URI /reads dan webserver belum siap

#### 4.4. Perancangan Validasi

Perlu dilakukan validasi untuk membandingkan hasil pengukuran tingkat stres yang dihasilkan pada sistem yang dibuat ini dengan metode validasi yang sudah ada. Pada penelitian ini, validasi menggunakan metode pengukuran tingkat stres secara subjektif yaitu NASA-TLX. Metode NASA-TLX merupakan metode pengukuran beban kerja mental dengan mempertimbangkan enam dimensi untuk menilai beban mental. Dari enam dimensi akan ditentukan pembobotan dimensi yang paling mempengaruhi kerja, dan dilanjutkan dengan penghitungan skor dari 0 – 100 pada setiap skala. Alasan menggunakan metode NASA-TLX adalah karena pada metode ini menggunakan suatu studi kasus untuk menentukan tingkat stres.

Pada penelitian ini menggunakan studi kasus pengerjaan Tugas Akhir, sehingga pada penelitian ini mengambil sampel pada 11 responden mahasiswa S1 yang sedang mengerjakan Tugas Akhir. Dibawah ini merupakan deskripsi 6 dimensi yang dijadikan pertanyaan terkait dengan studi kasus pengerjaan Tugas Akhir oleh mahasiswa S1, dapat dilihat pada tabel 4.12

*Tabel 4.12. Dimensi NASA-TLX*

Dimensi	Keterangan
Mental Demand (MD)	Seberapa besar aktivitas mental dan persepsi yang dibutuhkan untuk melakukan suatu pekerjaan (seperti berfikir, memutuskan,

	mengingat, merencanakan, dsb)? Apakah pekerjaan tersebut mudah atau sulit, sederhana atau kompleks?
Physical Demand (PD)	Seberapa besar aktivitas fisik yang dibutuhkan dalam melakukan suatu pekerjaan (seperti mendorong, menarik, mengontrol, memutar, dsb)? Apakah pekerjaan tersebut mudah atau sulit, lambat atau cepat, ringan atau berat?
Temporal Demand (TD)	Seberapa lama tekanan (pressure) yang dirasakan selama elemen pekerjaan berlangsung? Apakah pekerjaan perlahan dan santai atau cepat melelahkan?
Performance (OP)	Seberapa besar keberhasilan seseorang di dalam menyelesaikan pekerjaannya? Dan seberapa puas dengan hasil kinerja yang dicapai?
Frustration (FR)	Seberapa besar rasa tidak aman, putus asa, tersinggung, terganggu yang dirasakan selama bekerja?
Effort (EF)	Seberapa keras usaha (termasuk mental dan fisik) yang dibutuhkan dalam menyelesaikan suatu pekerjaan?

Langka pertama yang dilakukan ialah pembobotan, pada bagian ini responden diminta untuk memilih satu diantara dua dimensi yang dirasa lebih dominan menimbulkan beban mental terhadap pengerjaan Tugas Akhir. Kuisisioner yang diberikan berbentuk tabel perbandingan berpasangan yang terdiri dari 15 perbandingan berpasangan, dari kuisisioner ini dihitung jumlah pilihan dimensi yang dipilih dari setiap perbandingan pasangan dimensi yang dirasa lebih dominan. Jumlah pilihan dimensi inilah yang dijadikan bobot untuk setiap dimensi beban mental. Langkah kedua yang dilakukan ialah pemberian nilai tiap dimensi, pada bagian ini responden diminta untuk memberikan nilai pada tiap dimensi dengan skala 1-100 dimana 1 adalah nilai rendah atau sedikit berpengaruh pada pengerjaan Tugas

Akhir, sedangkan 100 adalah nilai tinggi atau banyak berpengaruh pada pengerjaan Tugas Akhir. Nilai yang diberikan adalah subjektif tergantung pada beban mental yang dirasakan oleh responden tersebut.

Langkah ketiga adalah perhitungan skor NASA-TLX, pada bagian ini dilakukan perhitungan skor NASA-TLX untuk mengetahui klasifikasi tingkat beban mental berdasarkan dari hasil skor NASA-TLX. Untuk mendapatkan skor NASA-TLX, bobot dimensi pada langkah pertama dikalikan dengan nilai dimensi pada langkah kedua, kemudian nilai perkalian tiap dimensi tersebut ditambahkan dan dibagi 15 (jumlah perbandingan berpasangan). Setelah mengetahui skor NASA-TLX dibawah ini adalah tabel klasifikasi berdasarkan penjelasan Hart dan Staveland dalam Hendrawan dkk dalam teori NASA-TLX, dapat dilihat pada tabel 4.13

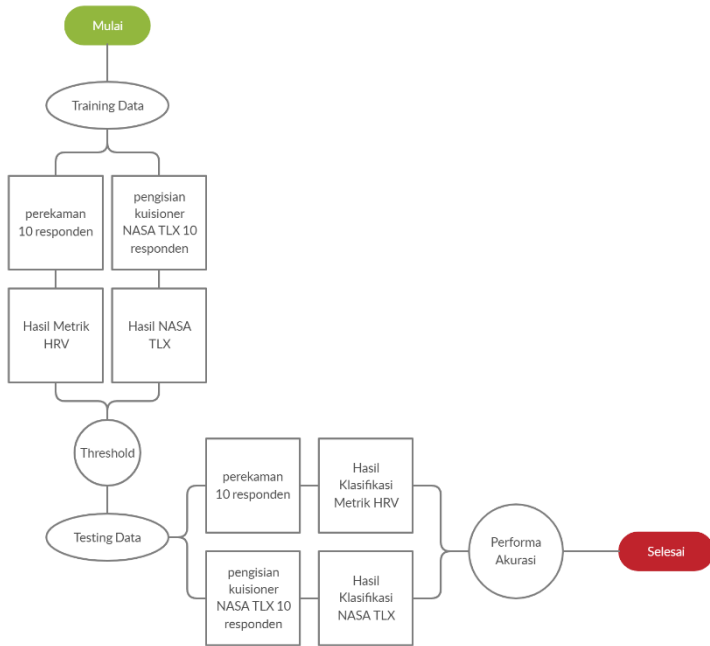
Tabel 4.8. Interpretasi skor NASA-TLX

Skor	Klasifikasi
<50	Rendah
50-80	Sedang
>80	Tinggi

#### 4.5. Perancangan *Deployment*

Pada tahap *deployment* dilakukan uji coba kepada responden dan analisa hasil. Uji coba kepada responden dilakukan pada waktu yang sama supaya hasil yang didapatkan seragam, waktu yang dipilih adalah malam hari ketika responden (Mahasiswa S1 tingkat akhir) sedang mengerjakan Tugas Akhir nya. Dilakukan *training data* dengan merekam sinyal ECG kepada 10 responden kemudian responden diminta untuk mengisi kuisisioner NASA TLX. Setelah itu hasilnya digunakan untuk mencari *threshold* sebagai landasan untuk menentukan tingkat stres dengan metrik HRV. Setelah diperoleh *threshold*, dilakukan *testing data* dengan merekam sinyal ECG kepada 10 responden dan mengisi kuisisioner NASA TLX. Dari hasil klasifikasi kedua metode tersebut diperoleh performa akurasi

pengukuran tingkat stres menggunakan metrik HRV dengan metode validasi NASA TLX. Gambar *flowchart deployment* dapat dilihat pada Gambar 4.14.



Gambar 4.14. *Flowchart deployment*

*Halaman ini sengaja dikosongkan*

## BAB V IMPLEMENTASI

Pada bagian ini akan menjelaskan mengenai proses implementasi terhadap sistem aplikasi pengukur tingkat stres sesuai dengan *output* dari proses perancangan yang telah dibuat.

### 5.1. Lingkungan Implementasi

Pengembangan sistem aplikasi pengukur tingkat stres menggunakan *smartphone* dan laptop dengan spesifikasi yang tertera pada Tabel 5.1. dan Tabel 5.2.

*Tabel 5.1. Spesifikasi smartphone implementasi*

Type	Luna G55
Processor	Octa Core 1.8 Ghz Cortex-A53
Memory (RAM)	4 GB
Sistem Operasi	Android v6.0 (Marshmallow)
Graphic Card	Mali-T860
WiFi	Wi-Fi 802.11 a/b/g/n/ac, Dual Band, Wi-Fi Direct, DLNA, Hotspot

*Tabel 5.2. Spesifikasi laptop implementasi*

Processor	Intel® Core™ i3-3217U CPU @1.8GHz
Memory (RAM)	8 GB
Sistem Operasi	Windows 10 Education 64Bit
Graphic Card	Nvidia GeForce GT 620M

Seperti yang telah ditentukan pada tahap perancangan bahwa sistem pengukur tingkat stres ini dikembangkan dengan 3 sistem yang saling terintegrasi yakni menggunakan sistem *mobile*, sistem server, dan sistem ESP8266. Beberapa teknologi pengembangan yang digunakan pada masing-masing sistem dapat dilihat pada tabel 5.3 sampai 5.5.

Tabel 5.3. Teknologi pengembangan sistem mobile

Bahasa Pemrograman	Dart ver 2.3
<i>Framework</i>	Flutter ver 1.5.4
<i>Editor (IDE)</i>	Android Studio Ver 3.4
<i>Plugin</i>	Oscilloscope ver 0.0.3
	Http ver 0.11.3
	Countdown ver 0.1.0

Tabel 5.4. Teknologi pengembangan sistem server

<i>Web Server</i>	Werkzeug ver 0.15.2
Bahasa Pemrograman	Python ver 3.7
<i>Framework</i>	Flask
<i>Database</i>	MySQL ver 1.4.29
<i>Editor (IDE)</i>	Atom ver 1.38.2
<i>Library</i>	Hrvalysis
	QRSDetector
	Flaskext.mysql
	Flask_paginate

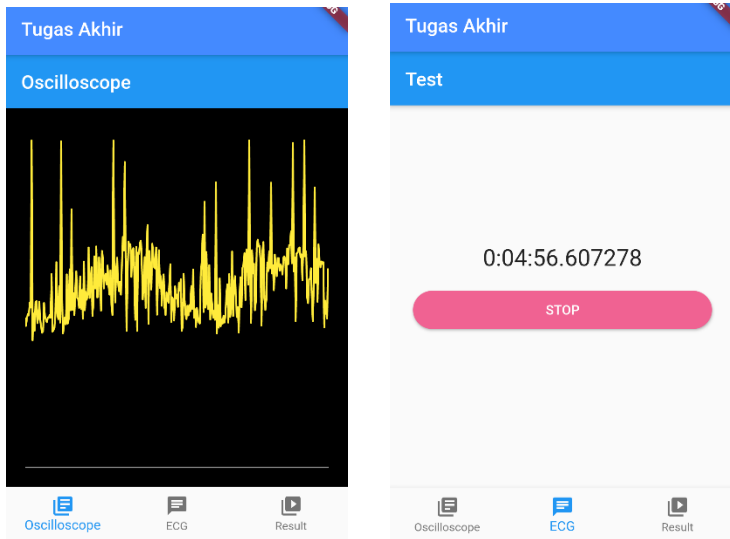
Tabel 5.5. Teknologi pengembangan sistem ESP8266

<i>Board</i>	ESP8266/Wemos D1 R2
<i>Editor (IDE)</i>	Arduino IDE
Komponen	Modul AD8232
	Modul SD Card
Perlengkapan	<i>Breadboard</i>
	Kabel Male to Male
	SD Card 16GB 80 MB/s
	<i>Power Adapter</i>

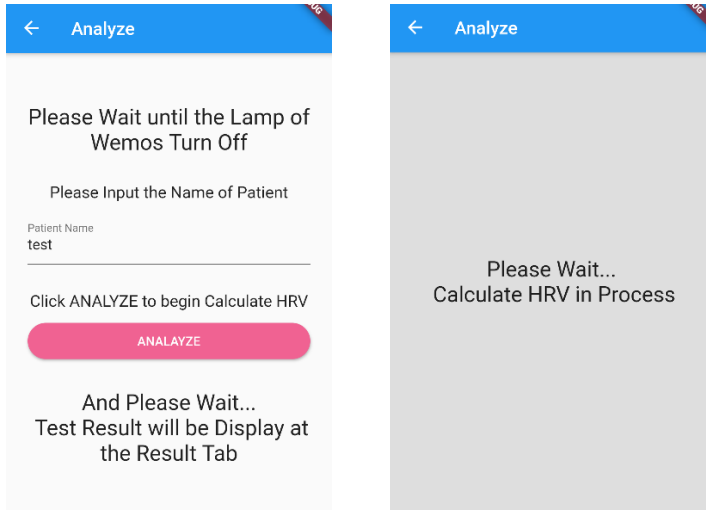


## 5.2. Implementasi Desain UI/UX

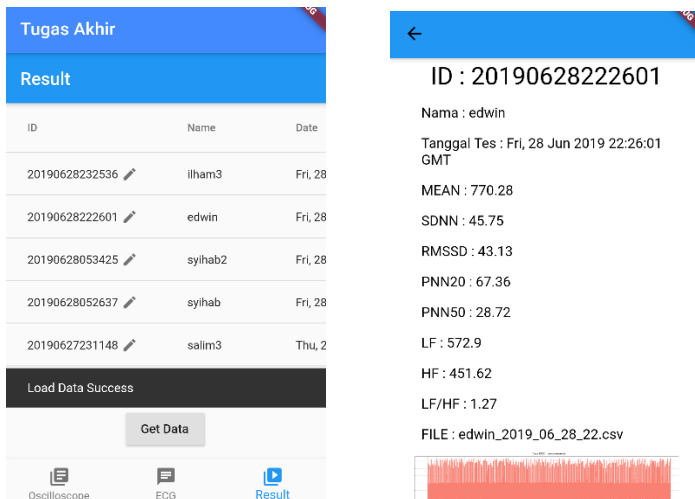
Pada pembuatan desain tampilan untuk mobile menggunakan *framework* Flutter yaitu merupakan *framework mobile UI cross-platform*, menggunakan bahasa pemrograman Dart. Berikut adalah hasil implementasi kode program dapat dilihat pada gambar 5.1 sampai 5.3



Gambar 5.1. Tampilan Osiloskop dan Test



Gambar 5.2. Tampilan Analyze



Gambar 5.3. Tampilan Hasil

### 5.3. Implementasi Sistem Android

Pada sistem android merupakan sistem yang berinteraksi dengan pengguna, terdiri dari 3 fitur utama yakni *monitoring*

sinyal ECG dalam bentuk *oscilloscope*, perekaman sinyal ECG serta analisis, dan menampilkan hasil analisis.

Pada monitoring sinyal elektrokardiogram, cara kerjanya yaitu sistem android mengakses URI “/get” dengan method GET pada sistem ESP8266 yang dilakukan secara terus menerus. Nilai yang didapat dimasukkan kedalam List, List inilah yang membentuk *oscilloscope* dengan menggunakan plugin *oscilloscope*. Kode program fitur ini dapat dilihat pada Gambar 5.4.

```
// Membuat list untuk menyimpan data ECG
List<double> dataECG = List();
Timer _timer;

Future<int> getData() async {
  // Meminta Data Signal ECG dalam bentuk satuan kepada ESP8266
  final String url = 'http://192.168.43.241/get';
  var res = await http.get(Uri.encodeFull(url), headers: { 'accept': 'application/json' });
  print(double.parse(res.body));
  setState(() {
    // Menambahkan satuan sinyal ECG ke dalam list
    dataECG.add(double.parse(res.body));
  });
  return 200;
}

// method untuk mengambil data satuan ecg secara terus menerus
_generateTrace(Timer t){
  // Fungsi mengambil data satuan ECG
  getData();
}

@override
initState() {
  super.initState();
  // menjalankan method untuk mengambil data terus menerus per 60ms
  _timer = Timer.periodic(Duration(milliseconds: 60), _generateTrace);
}

@override
void dispose() {
  _timer.cancel();
  super.dispose();
}
```

Gambar 5.4. Kode program *oscilloscope*

Pada fitur merekam sinyal elektrokardiogram cara kerjanya ialah sistem android mengakses URI “/on” dengan method GET pada sistem ESP8266 untuk memberitahu mulai perekaman sinyal elektrokardiogram dan tampilan android akan

menampilkan *countdown* sebagai penanda lamanya perekaman. Setelah waktu yang diinginkan perekaman telah sesuai (*countdown* 5 menit), kemudian sistem android mengakses URI “/off” dengan method GET pada sistem ESP8266 untuk memberitahu pemberhentian perekaman sinyal elektrokardiogram serta tampilan android berpindah ke tampilan analisis dan hasil rekaman akan dikirim ke sistem server oleh sistem ESP8266 untuk proses selanjutnya yaitu analisis. Kode program fitur merekam sinyal elektrokardiogram ini dapat dilihat pada Gambar 5.5.

```

void getECG() async {
  final response = await http.get("http://192.168.43.241/on");
}

void countdown(){
  cd = new Countdown(new Duration(minutes: 5));
  sub = cd.stream.listen(null);

  // start your countdown by registering a listener
  sub.onData((Duration d) {
    print(d);

    setState(() {
      this.countDownText = d.toString();
    });
  });

  // when it finish the onDone cb is called
  sub.onDone(() {
    print("Press STOP");

    setState(() {
      this.countDownText = "Press STOP";
    });
  });
}

@override
void dispose() {
  sub.pause();
  super.dispose();
}

static stopECG() async {
  final response = await http.get("http://192.168.43.241/off");
}

```

Gambar 5.5. Kode program merekam sinyal ECG

Pada fitur analisis, sistem android mengakses URI “/analyze/<name>” dengan method GET pada sistem server, analisis dilakukan di server setelah server mendapatkan hasil rekaman sinyal elektrokardiogram dalam bentuk csv yang dikirim oleh sistem ESP8266. Parameter name tersebut digunakan untuk identifikasi nama subjek (pasien) yang diukur tingkat stres nya. Kode program analisis pada sistem android dapat dilihat pada Gambar 5.6

```
Future<bool> analyze(String name) async {  
  var response = await http.get("http://192.168.43.197:5000/analyze/"+name);  
  if (response.statusCode == 200) {  
    return true;  
  } else {  
    return false;  
  }  
}
```

*Gambar 5.6. Kode program analisis*

Pada fitur menampilkan hasil, sistem android mengakses URI “/reads” dengan method GET pada sistem server, server akan mengirimkan json hasil analisis yang diambil dari database MySQL. Kode program menampilkan hasil pada sistem android dapat dilihat pada Gambar 5.7

```

//Membuat list kosong sebagai tempat menaruh hasil API (GET)
List<Result> list = List();
//Sebagai penanda proses pengambilan API sudah selesai atau belum
bool isLoading = false;

//method untuk mengambil data result API (GET)
getData() async {
  setState(() {
    | isLoading = true;
  });
  final response =
  await http.get("http://192.168.43.197:5000/reads");
  if (response.statusCode == 200) {
    list = (json.decode(response.body) as List)
      .map((data) => new Result.fromJson(data))
      .toList();
    _scaffoldState.currentState.showSnackBar(SnackBar(
      | content: Text("Load Data Success"),
    ));
    setState(() {
      | isLoading = false;
    });
  } else {
    _scaffoldState.currentState.showSnackBar(SnackBar(
      | content: Text("Load Data Failed"),
    ));
    throw Exception('Failed to load data');
  }
}
}

```

Gambar 5.7. Kode program menampilkan hasil

#### 5.4. Implementasi Sistem ESP8266

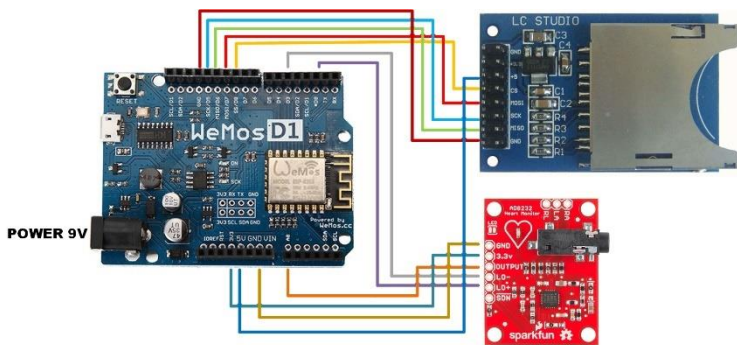
Sebelum melakukan penulisan kode program, perlu dilakukan perancangan rangkaian antara *board* dengan modul yang digunakan. Pada sistem ESP8266 ini menggunakan 2 modul yakni modul AD8232 dan modul SD Card dengan rangkaian yang dapat dilihat pada Tabel 5.6. dan Tabel 5.7 sedangkan untuk keseluruhan rangkaian dengan penggunaan *Breadboard* dan *Power Adapter* dengan aliran listrik 9V dapat dilihat pada Gambar 5.8

Tabel 5.6. Rangkaian modul AD8232 dengan Wemos D1 R2

Modul AD8232	Wemos D1 R2
GND	GND
3.3v	3.3v
OUTPUT	A0
LO-	D3
LO+	D0
SDN	-

Tabel 5.7. Rangkaian modul SD Card dengan Wemos D1 R2

Modul SD Card	Wemos D1 R2
GND	-
3.3v	-
5v	5v
CS	D8
MOSI	D7
SCK	D6
MISO	D5
GND	GND



Gambar 5.8. Rangkaian sistem ESP8266

Pembuatan kode program pada Arduino IDE pada umumnya terdiri dari 3 struktur yakni pendefinisian variabel, setup, dan loop. Pada sistem ini menggunakan beberapa library serta pendefinisian variabel dapat dilihat pada Gambar 5.9.

```

#include <SD.h>
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

const int chipSelect = D8;
File ecg;

const char* ssid = "dmhj";
const char* password = "muhajir11";
const char* host = "192.168.43.197";
ESP8266WebServer server(80); //Define server object

int lamp = 0;

```

Gambar 5.9. Pendefinisian variabel ESP8266

Variabel “chipSelect” dan “ecg” untuk keperluan pin SD Card dan file csv. Variabel “ssid”, “password”, “host”, dan “server(80)” untuk keperluan koneksi. Jaringan yang digunakan dalam sistem ESP8266 ini menggunakan *tethering* pada sistem android agar saling terhubung di jaringan yang sama. Variabel “lamp” untuk kontrol lampu dan penanda perekaman sinyal. Pada bagian setup dilakukan inisialisasi pin, SD Card, dan WiFi. Kode program untuk setup dapat dilihat pada Gambar 5.10



```

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Serial.print("\r\nWaiting for SD card to initialise...");
  if (!SD.begin(chipSelect)) { // CS is D8 in this example
    Serial.println("Initialising failed!");
    return;
  }
  Serial.println("Initialisation completed");
  // put your setup code here, to run once:
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(D0, INPUT); //setup for leads off detection LO +
  pinMode(D3, INPUT); // setup for leads off detection LO -
  pinMode(A0, INPUT);
  lamp = 0;
  delay(10);

  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.on("/on", startECG);
  server.on("/off", stopECG);
  server.on("/get", signalECG);
  server.begin(); //Start web server
  Serial.println("HTTP server started");
}

```

Gambar 5.10 Kode program setup ESP8266

Pada bagian setup diatas terdapat “server.on” dan beberapa URI, merupakan URI yang diakses oleh sistem lain agar dapat

berkomunikasi dengan sistem ESP8266. Pada URI “/get” diakses oleh sistem android untuk mengambil satuan nilai sinyal elektrokardiogram yang digunakan untuk pembentukan *oscilloscope*. Untuk URI “/on” diakses oleh sistem android untuk memberitahukan sistem ESP8266 untuk merekam, variabel “lamp” berubah menjadi 1 yang artinya lampu menyala dan proses rekaman sedang berlangsung. Untuk URI “/off” diakses oleh sistem android untuk memberitahukan bahwa perekaman sinyal berhenti dan mengirimkan hasilnya berupa csv kepada sistem server, variabel “lamp” berubah menjadi 0 menandakan bahwa lampu mati dan memberhentikan proses perekaman. Kode program ketiga fungsi tersebut dapat dilihat pada Gambar 5.11

```

void signalECG() {
    String signal = String(analogRead(A0));
    server.send(200, "application/json", signal);
}

void startECG() {
    lamp = 1;
    SD.remove("dataecg.csv");
    ecg = SD.open("dataecg.csv", FILE_READ|FILE_WRITE);
    ecg.print("timestamp");
    ecg.print(',');
    ecg.print("value");
    ecg.print('\n');
    String status = "Starting Get ECG Signal";
    server.send(200, "text/html", status);
}

void stopECG(){
    String stop = "Stop Get ECG Signal";
    lamp = 0;
    ecg.close();
    WiFiClient client;
    if (client.connect(host, 5000)) {
        File dataFile = SD.open("dataecg.csv");
        client.println(F("POST /files/dataecg.csv HTTP/1.1"));
        client.print(F("Host: daud"));
        client.println(host);
        client.println(F("Connection: close"));
        client.print(F("Content-Length: "));
        client.println(dataFile.size());
        client.println();
        while (dataFile.available()) {
            client.write(dataFile.read());
        }
        dataFile.close();
        client.stop();
    }
    server.send(200, "application/json", stop);
}

```

Gambar 5.11. Kode program fungsi ESP8266

Pada bagian *loop* dilakukan ini mendakan bahwa kode didalamnya dilakukan secara terus menerus. Disinilah proses perekaman sinyal elektrokardiogram dilakukan dengan variabel

“lamp” sebagai pendanda kapan waktu nya merekam dan tidak. Kode program *loop* tersebut dapat dilihat pada Gambar 5.12

```

void loop(void) {
  // put your main code here, to run repeatedly:
  server.handleClient();

  if (lamp == 1) {
    digitalWrite(LED_BUILTIN, LOW);
    ecg.print(2);
    ecg.print(',');
    ecg.print(analogRead(A0));
    ecg.print('\n');
    delay(2);
  } else if (lamp == 0) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
}

```

Gambar 5.12. Kode program *loop* ESP8266

## 5.5. Implementasi Sistem Server

Pengembangan sistem server ini berperan sebagai menerima hasil rekaman kemudian menganalisis nya dan menampilkan hasilnya. Fungsi menerima file ini diakses otomatis oleh sistem ESP8266 dengan mengakses URI “/files/<filename>” ketika sistem android memberhentikan proses perekaman sinyal. Kode program menerima file csv ini dapat dilihat pada Gambar 5.13

```

@app.route("/files/<filename>", methods=["POST"])
def post_file(filename):
    """Upload a file."""

    if "/" in filename:
        # Return 400 BAD REQUEST
        abort(400, "no subdirectories directories allowed")

    with open(os.path.join(UPLOAD_DIRECTORY, filename), "wb") as fp:
        fp.write(request.data)

    # Return 201 CREATED
    print('kirim csv selesai :')
    print(time.strftime("%H:%M:%S", time.localtime()))
    return "", 201

```

Gambar 5.13. Kode program menerima file csv

Untuk fitur analisis ini diakses oleh sistem android dengan URI “/analyze/<name>” dengan method GET setelah file csv telah selesai dikirim ke server oleh sistem ESP8266 dengan tanda matinya lampu pada wemos D1 R2. Proses analisis ini terdiri dari 2 tahap yakni mendeteksi hrv dan menghitung metrik hrv, kedua proses ini dilakukan dengan bantuan library. kemudian hasilnya disimpan di database MySQL. Untuk mendeteksi hrv (*heart rate variability*) ini *output* yang dihasilkan adalah berupa titik puncak ada di baris ke berapa, sedangkan *input* yang diperlukan untuk menghitung metrik hrv ialah jarak antar puncak dengan satuan *milisecond* (ms) sehingga perlu kode program tambahan agar berjalan dengan semestinya. Kode program analisis ini dapat dilihat pada Gambar. 5.14 dan 5.15

```
def rr_interval_detect(name):
    qrs_detector = QRSDetectorOffline(ecg_data_path="files/dataecg.csv", name=name,
    verbose=True, log_data=True, plot_data=True, show_plot=False)
    peaks_indices = qrs_detector.qrs_peaks_indices
    rr_interval = []
    index = 2
    length = len(peaks_indices)
    while index < length:
        rr_interval.append((peaks_indices[index] - peaks_indices[index-1])/450*1000)
        print((peaks_indices[index] - peaks_indices[index-1])/450*1000, end=" ")
        index = index + 1
    # print(rr_interval)
    return rr_interval
```

Gambar 5.14. Kode program deteksi hrv

```
@app.route('/analyze/<name>', methods=['GET'])
def calculate_metrics(name):
    print('Mulai perhitungan :')
    print(time.strftime("%H%M%S", time.localtime()))
    nn_list = rr_interval_detect(name)
    time_domain_features = get_time_domain_features(nn_list)
    frequency_domain_features = get_frequency_domain_features(nn_list)
```

Gambar 5.15. Kode program menghitung metrik hrv

Kemudian untuk fitur menampilkan hasil, sistem android mengakses URI “/reads” dengan method GET untuk memperoleh hasil analisa yang disimpan dalam database dan ditampilkan pada *interface* android. Untuk server dapat menampilkan hasilnya melalui halaman website dengan

mengakses URI “/dashboard”. Kode program menampilkan hasil tersebut dapat dilihat pada Gambar 5.16 dan 5.17.

```
@app.route('/reads', methods=['GET'])
def read():
    db = mysql.connect()
    cursor = db.cursor()
    get = ("SELECT * FROM results ORDER BY date DESC")
    cursor.execute(get)
    rows = cursor.fetchall()
    dataList = []
    if rows is not None:
        for item in rows:
            dataTempObj = {
                'id' : item[0],
                'name' : item[1],
                'date' : item[2],
                'mean' : item[3],
                'sdnn' : item[4],
                'sdsd' : item[5],
                'rmssd' : item[6],
                'pnn20' : item[7],
                'pnn50' : item[8],
                'lf' : item[9],
                'hf' : item[10],
                'lfhf' : item[11],
                'file' : item[12],
                'image' : item[13],
                'classification' : item[14],
            }
            dataList.append(dataTempObj)
        db.commit()
        db.close()
        return jsonify(dataList)
    else:
        return jsonify('data kosong')
```

Gambar 5.16. Kode program reads

```

@app.route('/dashboard')
def dashboard():
    page, per_page, offset = get_page_args(page_parameter='page',
                                          per_page_parameter='per_page')
    pagination_users = get_data(offset=offset, per_page=per_page)
    db = mysql.connect()
    cursor = db.cursor()
    get = ("SELECT * FROM results")
    cursor.execute(get)
    cursor.fetchall()
    total = cursor.rowcount
    db.commit()
    db.close()
    pagination = Pagination(page=page, per_page=per_page, total=total,
                           css_framework='bootstrap4')
    return render_template('dashboard.html',
                          users=pagination_users,
                          page=page,
                          per_page=per_page,
                          pagination=pagination,
                          )

def get_data(offset=0, per_page=10):
    db = mysql.connect()
    cursor = db.cursor()
    get = ("SELECT * FROM results ORDER BY date DESC")
    cursor.execute(get)
    rows = cursor.fetchall()
    db.commit()
    db.close()
    return rows[offset: offset + per_page]

```

Gambar 5.17. Kode program dashboard

## 5.6. Implementasi Test Case

Tes yang dilakukan menggunakan metode *blackbox testing* dengan skenario yang telah dirancang pada BAB IV. Hasil implementasi *test case* dapat dilihat pada Tabel 5.8.

Tabel 5.8. Implementasi test case

Use Case	Kategori	Skenario	Hasil
U1	Normal	Sistem ESP8266 telah siap dan Sistem android pada tab <i>oscilloscope</i>	<i>Interface</i> Android menampilkan sinyal detak jantung dalam bentuk <i>oscilloscope</i>
	Alternatif 1	Sistem ESP8266 belum siap dan Sistem android pada tab <i>oscilloscope</i>	<i>Oscilloscope</i> pada <i>Interface Android</i> tidak berjalan
U2	Normal	Sistem android akses URI /getecg dan Sistem ESP8266 telah siap	<i>Interface Android</i> mulai menampilkan <i>countdown</i> dan lampu Wemos menyala
	Normal	Sistem android akses URI /stopecg dan Sistem ESP8266 telah siap	<i>Interface Android</i> berpindah ke tampilan <i>analyze</i> . Lampu Wemos masih menyala hingga <i>file csv</i> selesai dikirimkan ke <i>Server</i> kemudian lampu Wemos mati.
	Normal	Sistem android akses URI /analyze/<name> dengan mengisi <i>field name</i> dan sistem webserver telah siap	<i>Interface Android</i> menampilkan tampilan <i>please wait</i> , dan berpindah ke <i>homepage</i> ketika selesai
	Alternatif 1	Sistem android akses URI	Interface Android menampilkan



		/getecg dan Sistem ESP8266 belum siap	<i>countdwn</i> dan Lampu Wemos tidak menyala
	Alternatif 2	Sistem android akses URI /stopecg dan Sistem ESP8266 belum siap	Interface Android berpindah ke tampilan <i>analyze</i> dan Lampu Wemos tidak menyala
	Alternatif 3	<i>Sistem android</i> akses <i>URI /analyze/&lt;name&gt;</i> tanpa mengisi <i>field name</i>	<i>Interface Android</i> menampilkan notifikasi harus mengisi <i>field name</i>
	Alternatif 4	<i>Sistem android</i> akses <i>URI /analyze/&lt;name&gt;</i> dengan mengisi <i>field name</i> dan <i>webserver</i> belum siap	Interface android menampilkan tampilan <i>please wait</i>
U3	Normal	Sistem android akses URI /reads dan webserver telah siap	<i>Interface android</i> menampilkan data hasil
	Alternatif 1	Sistem android akses URI /reads dan webserver belum siap	<i>Interface android</i> menampilkan tampilan <i>please wait</i>

*Halaman ini sengaja dikosongkan*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bagian ini akan menjelaskan mengenai hasil dan pembahasan dari pengembangan sistem aplikasi pengukur tingkat stres yang telah dibuat dalam pengerjaan tugas akhir ini.

#### **7.1. Performa Sistem**

Sistem aplikasi pengukur tingkat stres ini dijalankan pada *localhost* dengan spesifikasi seperti yang disebutkan pada BAB V bagian Lingkungan Implementasi menggunakan jaringan *thethering* dari *smartphone* yang digunakan untuk implementasi. Berdasarkan perancangan sistem, terdapat 2 kebutuhan *non fungsional* yaitu integrasi dan performa yang mencakup uji jarak, akurasi, dan kecepatan.

##### **7.1.1. Uji Integrasi**

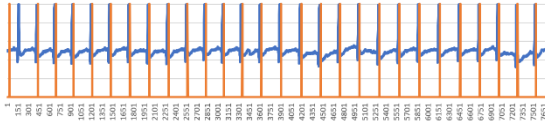
Pada pengujian integrasi telah di tes menggunakan *blackbox testing* pada BAB V yaitu implementasi *test case*, hasilnya ketiga sistem yang dikembangkan dapat terintegrasi saling berkomunikasi menjadi satu kesatuan sistem aplikasi pengukur tingkat stres.

##### **7.1.2. Uji Jarak**

Pada pengujian jarak dilakukan dengan cara menjalankan aplikasi pada android dengan jarak 8 meter dari server dan mikrokontroler. Hasilnya sistem dapat berjalan dengan baik hingga jarak 8 meter tanpa benda atau bangunan yang menghalangi.

##### **7.1.3. Uji Ketepatan Algoritma**

Pada uji ketepatan algoritma dilakukan dengan cara *plotting* hasil rekaman sinyal ECG. Kemudian mengecek berapa titik puncak yang *miss detect* dari keseluruhan titik puncak yang ada. Contoh *plotting* hasil sinyal ECG dan *miss detect* dapat dilihat pada Gambar 6.1. Pada gambar tersebut terdapat 1 *miss detect* pada 28 titik puncak, keseluruhan hasil *miss detect* dapat dilihat pada Tabel 6.1.



Gambar 6.1. Contoh miss detect titik puncak

Tabel 6.1. Uji ketepatan algoritma

Nama	Jumlah Puncak	Miss Detect	Akurasi
Indra	402	0	100%
Wahyu	498	1	99.7992%
Titus	473	1	99.78858%
Dani	365	0	100%
Aldi	480	1	99.79167%
Habib	392	1	99.7449%
Azzam	439	0	100%
Salim	460	1	99.78261%
Edwin	384	0	100%
Ilham	466	1	99.78541%

*Miss detect* yang terjadi berada pada titik puncak pertama, untuk menghasilkan deteksi yang sempurna 100%, perhitungan RR interval dimulai dari titik puncak ke 2. Sehingga dapat dikatakan data HRV yang diperoleh memenuhi syarat untuk proses pengolahan selanjutnya yaitu menghitung metrik HRV.

#### 7.1.4. Uji Kecepatan

Pada pengujian kecepatan, ada 3 aspek yang diukur yakni kecepatan transfer nilai sinyal ECG dari sensor dengan *delay* 2 milisecond (ms), *transfer file* csv dari sistem ESP8266 ke *webserver*, dan kecepatan perhitungan analisis HRV dari csv.

Pada pengukuran kecepatan *transfer* nilai sinyal ECG dari sensor dilakukan dengan cara merekam dengan lama waktu

berbeda, kemudian dicatat berapa baris atau nilai yang dihasilkan dari pengukuran tersebut. Dengan percobaan yang dilakukan diperoleh hasil rata-rata menghasilkan 449 bps (*beats per second*), artinya dalam waktu 1 detik menghasilkan 449 nilai. Hasil percobaan dapat dilihat pada Tabel 6.2.

Tabel 6.2. Uji kecepatan transfer sinyal dari sensor

Waktu (second)	Nilai (baris)	Beats per second
10 s	4631 baris	463 bps
30 s	13540 baris	451 bps
60 s	26809 baris	447 bps
120 s	53573 baris	446 bps
180 s	80059 baris	445 bps
240 s	106767 baris	445 bps
300 s	133244 baris	444 bps

Untuk pengukuran kecepatan *transfer file csv* dan analisis dilakukan dengan cara *print timestamp* saat mulaimerekam, berhenti merekam, csv diterima, mulai hitung, dan selesai hitung. Pada percobaan uji kecepatan ini dilakukan pada 10 responden dengan lama waktu perekaman sinyal ECG selama 5 menit. Hasilnya adalah rata-rata kecepatan pengiriman file csv selama 30,5 detik sedangkan rata-rata melakukan analisis selama 9,3 detik. Hasil dapat dilihat pada Tabel 6.3.

```
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.43.1 - - [27/Jun/2019 19:55:14] "GET /reads HTTP/1.1" 200 -
192.168.43.1 - - [27/Jun/2019 19:55:24] "GET /reads HTTP/1.1" 200 -
192.168.43.1 - - [27/Jun/2019 19:55:29] "GET /reads HTTP/1.1" 200 -
mulai merekam :
201802
192.168.43.1 - - [27/Jun/2019 20:18:03] "GET /getecg HTTP/1.1" 200 -
berhenti merekam :
202303
192.168.43.1 - - [27/Jun/2019 20:23:36] "GET /stopecg HTTP/1.1" 200 -
kirim csv selesai :
202336
192.168.43.241 - - [27/Jun/2019 20:23:36] "POST /files/dataecg.csv HTTP/1.1" 201 -
* Detected change in 'D:\tugas_akhir\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 319-722-137
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Detected change in 'D:\tugas_akhir\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 319-722-137
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
Mulai perhitungan :
202901
args peaks indices
[
  31    366    778    1178    1584    1982    2387    2793    3198    3485
  3820    4220    4625    5030    5430    5840    6245    6642    7053    7453
  7858    8263    8674    9092    9488    9893    10298    10609    11102    11508
  11915    12327    12727    13136    13542    13945    14355    14757    14978    15379
  15785    16191    16591    17000    17406    17812    18217    18630    19031    19436
  19841    20255    20656    21061    21465    21875    22274    22680    23092    23433
  23785    24192    24545    24828    25323    25745    26096    26449    26873    27156
```

Gambar 6.2. Uji kecepatan transfer csv dan analisis

Tabel 6.3. Uji kecepatan transfer csv dan analisis

Nama	Transfer csv (second)	Analisis (second)
Indra	32 s	8,37 s
Wahyu	31 s	9,6 s
Titus	31 s	7,63 s
Dani	29 s	8,31 s
Aldi	29 s	8,74 s
Habib	31 s	9,82 s
Azzam	28 s	8,29 s
Salim	31 s	10,84 s
Edwin	33 s	10,61 s
Ilham	32 s	10,79 s

## 7.2. NASA-TLX

Pengisian kuisioner NASA-TLX ini dilakukan kepada 10 responden mahasiswa S1 yang sedang mengerjakan Tugas Akhir. Responden berjenis kelamin laki-laki semua dan berusia antara 21 tahun sampai 23 tahun. Pengisian kuisioner dan tes dilakukan selama 2 hari sekitar pukul 21:00 – 23:30. Hasil kuisioner ini nantinya akan digunakan sebagai *data train* pencarian *threshold* untuk penentuan klasifikasi. Hasil kuisioner dapat dilihat pada Tabel 6.4.

Tabel 6.4. Hasil NASA TLX data train

Nama	Dimensi	Bobot	Rating	Bobot x Rating	
Indra	MD	2	70	140	WWL
	PD	0	50	0	1060
	TD	4	90	360	Skor
	OP	3	60	180	70,6
	FR	2	90	180	Klasifikasi
	EF	4	50	200	Sedang
Wahyu	MD	2	90	180	WWL
	PD	0	5	0	1275
	TD	5	90	450	Skor
	OP	1	85	85	85
	FR	4	80	320	Klasifikasi
	EF	3	80	240	Tinggi
Titus	MD	3	80	240	WWL
	PD	0	40	0	1214
	TD	4	87	348	Skor
	OP	1	60	60	80,9
	FR	5	92	460	Klasifikasi

	EF	2	53	106	Tinggi
Dani	MD	3	87	261	WWL
	PD	0	34	0	1137
	TD	2	56	112	Skor
	OP	3	78	234	75,8
	FR	4	80	320	Klasifikasi
	EF	3	70	210	Sedang
Aldi	MD	4	80	320	WWL
	PD	1	60	60	1100
	TD	2	60	120	Skor
	OP	0	60	0	73,3
	FR	4	70	280	Klasifikasi
	EF	4	80	320	Sedang
Habib	MD	4	95	380	WWL
	PD	1	30	30	1260
	TD	2	40	80	Skor
	OP	5	100	500	84
	FR	0	10	0	Klasifikasi
	EF	3	90	270	Tinggi
Azzam	MD	2	90	180	WWL
	PD	0	75	0	1251
	TD	2	80	160	Skor
	OP	4	80	320	83,4
	FR	2	83	166	Klasifikasi
	EF	5	85	425	Tinggi
Salim	MD	3	80	240	WWL
	PD	0	20	0	1080



	TD	3	75	225	Skor
	OP	3	65	195	72
	FR	5	70	350	Klasifikasi
	EF	1	70	70	Sedang
Edwin	MD	2	80	160	WWL
	PD	1	40	40	1200
	TD	2	70	140	Skor
	OP	4	80	321	80
	FR	1	90	90	Klasifikasi
	EF	5	90	450	Sedang
Ilham	MD	4	100	400	WWL
	PD	0	0	0	1100
	TD	2	50	100	Skor
	OP	5	100	500	73,3
	FR	3	30	90	Klasifikasi
	EF	1	10	10	Sedang

### 7.2.1. Uji Keseragaman

Untuk menguji keseragaman dari hasil kuisioner yang telah dibagikan kepada 10 kuisioner dapat dihitung menggunakan rumus

$$\bar{x} = \frac{\sum x_i}{n} = \frac{70,6 + 85 + \dots + 73,3}{10} = \frac{865,8}{10} = 77,83$$

$$\begin{aligned}
 \sigma &= \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}} \\
 &= \sqrt{\frac{(70,6 - 78,7)^2 + (85 - 78,7)^2 + \dots + (73,3 - 78,7)^2}{10 - 1}} \\
 &= \sqrt{\frac{266,06}{9}} = \sqrt{29,56} = 5,43
 \end{aligned}$$

$$BKA = \bar{x} + 3\sigma = 77,83 + 3(5,43) = 94,12$$

$$BKB = \bar{x} - 3\sigma = 77,83 - 3(5,43) = 61,54$$

Berdasarkan hasil kuisioner yang diperoleh skor paling atas adalah 85 dan skor paling bawah adalah 70,6. Nilai batas yang dihitung yaitu BKA sebesar 94,12 dan nilai batas bawah yang dihitung yaitu BKB sebesar 61,54. Sehingga diketahui bahwa tidak ada skor yang diperoleh melebihi batas kontrol. Artinya data yang diperoleh pada tahap sebelumnya dikatakan seragam. Hal ini sesuai dengan prinsip uji keseragaman dan data tersebut dapat digunakan untuk dalam pengujian selanjutnya.

### 7.2.2. Uji Kecukupan Data

Untuk tingkat kepercayaan 95% dan derajat ketelitian 10%

Tingkat kepercayaan 95%  $\rightarrow k = 2$

Tingkat ketelitian 10%  $\rightarrow s = 0,1$

$$\begin{aligned}
 N' &= \left( \frac{k/s \sqrt{(n \sum(x_i^2) - (\sum x_i)^2)}}{\sum x_i} \right)^2 \\
 &= \left( \frac{2/0,1 \sqrt{(10(60841,15) - (605750,89))}}{778,3} \right)^2 = 1,76
 \end{aligned}$$

$N' = 1,76$  dan  $N = 10$

Nilai  $N' < N$ , sehingga data mencukupi untuk menjadi bahan penelitian dengan tingkat kepercayaan 95% dan tingkat ketelitian 10%. Dapat diartikan bahwa sekurang-kurangnya 95 dari 100 nilai rata-rata dari data yang dicatat untuk suatu elemen kerja memiliki penyimpangan tidak lebih dari 10%. Oleh karena itu, jumlah data yang diambil dianggap cukup.

### 7.3. Metrik HRV

Setelah proses perekaman sinyal ECG selama 5 menit, kemudian mencari HRV nya dengan menentukan titik puncak dan dihitung interval tiap titik puncak. Setelah mendapatkan HRV kemudian dihitung metrik nya berdasarkan *time domain* dan *frequency domain*. Hasil perhitungan metrik *heart rate variability* dapat dilihat pada Tabel 6.5 dan 6.6

Tabel 6.5. Hasil metrik HRV data train time domain

Nama	Mean	Sdnn	Sdsd	Rmssd	Pnn20	Pnn50
Indra	733	51,96	44,03	44,03	63,75	24,75
Wahyu	739,84	36,69	31,19	31,19	50,25	9,6
Titus	622,24	33,39	17,76	17,77	21,02	0,85
Dani	806,1	47,4	30,8	30,81	56,47	10,74
Aldi	613,02	29,14	18,59	18,59	24,06	1,46
Habib	750,34	95,66	69,56	69,56	67,95	38,46
Azzam	670,19	41,65	33,39	33,39	13,04	0,92
Salim	639,43	40,11	24,46	24,46	34,93	5,46
Edwin	770,19	45,78	43,19	43,19	67,54	28,8
Ilham	630,11	37,56	42,08	42,08	18,75	1,29

Tabel 6.6. Hasil metrik HRV data train frequency domain

Nama	LF	HF	LF/HF
Indra	786,52	479,1	1,64
Wahyu	370,53	286,51	1,29

Titus	331,92	118,86	2,79
Dani	648,11	332,48	1,95
Aldi	194,61	220,63	0,88
Habib	2846,3	2668,66	1,07
Azzam	882,86	287,27	3,07
Salim	1000,24	327,01	3,06
Edwin	568,97	452,92	1,26
Ilham	173,79	210,37	0,83

#### 7.4. Analisis Data Train

Pada tahap ini dilakukan pencarian *threshold* sebagai dasar penentuan hasil klasifikasi tingkat stres oleh sistem berdasarkan dari perbandingan hasil NASA-TLX dan Metrik HRV. Berdasarkan hasil NASA-TLX dicari masing-masing *threshold* untuk tiap metrik HRV yang akan digunakan sebagai penentu klasifikasi metrik HRV apakah termasuk sedang atau tinggi. Hasil pencarian *threshold* dapat dilihat pada Table 6.7.

Tabel 6.7. Hasil *threshold*

Metrik	Rata-rata Sedang	Rata-rata Tinggi	Threshold
Mean	698,6417	695,6525	697,1471
Sdnn	41,99167	51,8475	46,91958
Sdsd	33,85833	37,975	35,91667
Rmssd	33,86	37,9775	35,91875
Pnn20	44,25	38,065	41,1575
Pnn50	12,08333	12,4575	12,27042
LF	562,04	1107,903	834,9713
HF	337,085	840,325	588,705
LF/HF	1,603333	2,055	1,829167

Melakukan klasifikasi menggunakan metrik HRV berdasarkan threshold yang diperoleh pada analisis perbandingan dengan metode NASA-TLX. Penentuan klasifikasi ditentukan dengan cara melabeli tiap metrik HRV dengan sedang atau tinggi berdasarkan threshold yang diperoleh pada tahap sebelumnya, dari kesembilan metrik label yang paling banyak dijadikan sebagai hasil klasifikasi. Hasil klasifikasi dengan metrik HRV pada *data train* diperoleh 7 dari 10 responden sesuai dengan hasil klasifikasi pada NASA-TLX, dapat dilihat pada Tabel 6.8.

Tabel 6.8. Hasil klasifikasi metrik HRV data train

Nama	Klasifikasi berdasarkan metrik HRV	Klasifikasi berdasarkan NASA-TLX
Indra	Sedang	Sedang
Wahyu	Sedang	Tinggi
Titus	Sedang	Tinggi
Dani	Sedang	Sedang
Aldi	Sedang	Sedang
Habib	Tinggi	Tinggi
Azzam	Sedang	Tinggi
Salim	Sedang	Sedang
Edwin	Sedang	Sedang
Ilham	Sedang	Sedang

### 7.5. Analisis Data Testing

Pada data testing dilakukan perekaman kepada 10 responden dan pengisian kuisisioner NASA TLX sehingga diperoleh performa akurasi dari sistem untuk pengukuran tingkat stres menggunakan metrik HRV dengan metode validasi NASA TLX. Hasil analisis data testing dapat dilihat pada Tabel 6.9 sampai Tabel 6.12.

Tabel 6.9. Hasil NASA TLX data testing

Nama	Dimensi	Bobot	Rating	Bobot x Rating	
Dani	MD	2	80	160	WWL
	PD	0	30	0	1000
	TD	2	50	100	Skor
	OP	4	70	280	66,6
	FR	3	60	180	Klasifikasi
	EF	4	70	280	Sedang
Aldi	MD	3	80	240	WWL
	PD	0	60	0	1160
	TD	3	80	240	Skor
	OP	1	70	70	77,3
	FR	3	70	210	Klasifikasi
	EF	5	80	400	Sedang
Habib	MD	2	80	160	WWL
	PD	3	80	240	1200
	TD	0	60	0	Skor
	OP	2	70	140	80
	FR	5	90	450	Klasifikasi
	EF	3	70	210	Sedang
Edwin	MD	3	85	255	WWL
	PD	1	70	70	1205
	TD	3	80	240	Skor
	OP	2	70	140	80,3
	FR	1	50	50	Klasifikasi
	EF	5	90	450	Tinggi

Azzam	MD	3	90	270	WWL
	PD	0	75	0	1289
	TD	1	80	80	Skor
	OP	2	82	164	85,93
	FR	4	85	340	Klasifikasi
	EF	5	87	435	Tinggi
Agung	MD	4	83	332	WWL
	PD	2	70	140	1118
	TD	5	80	400	Skor
	OP	0	65	0	74,53
	FR	2	63	126	Klasifikasi
	EF	2	60	120	Sedang
Ilham	MD	5	100	500	WWL
	PD	0	0	0	1240
	TD	2	50	100	Skor
	OP	4	100	400	82,66
	FR	1	30	30	Klasifikasi
	EF	3	70	210	Tinggi
Indra	MD	2	70	140	WWL
	PD	0	50	0	1090
	TD	4	60	240	Skor
	OP	4	70	280	72,6
	FR	2	80	160	Klasifikasi
	EF	3	90	270	Sedang
Wahyu	MD	5	90	450	WWL
	PD	0	10	0	1040
	TD	4	80	320	Skor

	OP	1	20	20	69,3
	FR	2	50	100	Klasifikasi
	EF	3	50	150	Sedang
Salim	MD	3	100	300	WWL
	PD	0	40	0	1100
	TD	5	82	410	Skor
	OP	1	60	60	82,6
	FR	4	85	340	Klasifikasi
	EF	2	65	130	Tinggi

Tabel 6.10. Hasil metrik HRV data testing time domain

Nama	Mean	Sdnn	Sdsd	Rmssd	Pnn20	Pnn50
Dani	720,10	37,15	21,62	21,62	33,82	2,70
Aldi	648,71	29,95	23,11	23,11	32,74	2,43
Habib	574,80	39,19	20,77	20,77	17,61	2,74
Edwin	653,20	37,05	28,43	28,43	47,33	6,44
Azzam	754,50	43,07	21,49	21,49	30,85	2,83
Agung	783,67	45,01	32,59	32,59	51,47	12,27
Ilham	717,16	68,52	96,17	96,17	61,86	29,34
Indra	755,86	53,89	54,52	54,52	75,77	37,63
Wahyu	712,50	36,25	27,01	27,01	41,65	6,30
Salim	616,63	59,41	30,58	30,58	34,45	6,72

Tabel 6.11. Hasil metrik HRV data testing frequency domain

Nama	LF	HF	LF/HF
Dani	437,89	117,53	3,73
Aldi	248,42	193,97	1,28
Habib	605,99	320,48	1,89



Edwin	554,48	246,99	2,24
Azzam	794	106,46	7,46
Agung	1098,73	304,31	3,61
Ilham	610,82	1740,11	0,35
Indra	880,29	710,41	1,24
Wahyu	264,78	128,24	2,06
Salim	1286,24	424,99	3,03

Tabel 6.12. Hasil klasifikasi metrik HRV data testing

Nama	Klasifikasi berdasarkan metrik HRV	Klasifikasi berdasarkan NASA-TLX
Dani	Sedang	Sedang
Aldi	Sedang	Sedang
Habib	Sedang	Sedang
Edwin	Sedang	Tinggi
Azzam	Sedang	Tinggi
Agung	Sedang	Sedang
Ilham	Tinggi	Tinggi
Indra	Tinggi	Sedang
Wahyu	Sedang	Sedang
Salim	Tinggi	Tinggi

Berdasarkan hasil diatas, klasifikasi dengan metrik HRV pada *data testing* diperoleh 7 dari 10 responden sesuai dengan hasil klasifikasi pada NASA-TLX, sehingga sistem mempunyai performa akurasi sebesar 70%.

*Halaman ini sengaja dikosongkan*

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Pada bagian ini akan menjelaskan mengenai kesimpulan dari pengerjaan tugas akhir ini dan saran untuk pengembangan perangkat lunak selanjutnya yang relevan.

#### **7.1. Kesimpulan**

Berikut merupakan kesimpulan yang dapat diambil dari pengembangan perangkat lunak yang telah dilakukan.

1. Sistem pengukur tingkat stres yang telah dibangun menggunakan 3 sub-sistem yang saling terintegrasi yaitu *mobile*, mikrokontroler, dan *webserver*.
2. Sistem dapat bekerja secara *wireless* dan mendapatkan hasil secara langsung dengan data sinyal ECG sebanyak 449 bps (*beats per second*) pada *delay* 2 ms, lama pengiriman csv selama 30,5 s dan proses analisis metrik HRV selama 9,3 s dengan lama waktu perekaman selama 5 menit.
3. Sistem yang telah dibangun mempunyai performa akurasi sebesar 70% dengan metode validasi NASA TLX.

#### **7.2. Saran**

Berikut merupakan saran untuk pengembangan lebih lanjut yang dapat dilakukan dalam menyempurnakan pengembangan perangkat lunak.

1. Melakukan validasi dengan metode pengukur tingkat stres yang objektif seperti tes kelenjar saliva atau kedipan mata.
2. Pengambilan sampel dilakukan kepada lebih banyak responden dan bervariasi.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Lifeline, "What is Stress?," Lifeline Australia, [Online]. Available:  
<https://www.lifeline.org.au/static/uploads/files/what-is-stress-wfvgiurqqawx.pdf>. [Diakses 03 March 2019].
- [2] University of Regina, "What is Stress," 1998. [Online]. Available:  
<https://www.uregina.ca/student/counselling/assets/docs/pdf/what-is-stress.pdf>. [Diakses 03 March 2019].
- [3] Civil Service Employee Assistant Service, "Understanding Stress," 2015. [Online]. Available:  
<https://www.cseas.per.gov.ie/wp-content/uploads/CSEAS-Understanding-Stress-Booklet.pdf>. [Diakses 03 March 2019].
- [4] C. Newton, "The Stress Curve," 2017. [Online]. Available:  
<http://www.clairenewton.co.za/images/resources/posters/the-stress-curve.pdf>. [Diakses 03 March 2019].
- [5] K. Hye-Geum, C. Eun-Jin, B. Dai-Seg, L. H. Young dan K. Bon-Hoon, "Stress and Heart Rate Variability: A Meta-Analysis and Review of the Literature," *Psychiatry Investigation*, vol. 3, no. 15, pp. 235-245, 2018.
- [6] K. Nasim, H. A. Jahan dan A. S. Syed, "Heart Rate Variability - A Review," *Journal of Basic and Applied Sciences*, vol. 7, no. 1, pp. 71-77, 2011.
- [7] D. M. Anthoni, V. Sarah dan I. A. Paul, "Basic ECG Theory, Recordings, and Interpretation," *Handbook of Cardiac Anatomy, Physiology, and Device*, vol. 15, pp. 191-201, 2005.
- [8] T. Pereira, P. R. Almeida, J. P. Cunha dan A. Aguiar, "Heart rate variability metrics for fine-grained stress level assessment," *Computer Methods and Programs in Biomedicine*, vol. 148, pp. 71-80, 2017.

- [9] R. Abdurrohman, A. Rusdinar dan A. G. Permana, "Rancang Bangun Sistem Pengiriman Data Electrocardiograph Dengan Media Wi-Fi," *e-Proceeding of Engineering* , vol. 4, no. 3, pp. 3270-3277, 2017.
- [10] M. Arasyandi dan A. Bakhtiar, *Analisa Beban Kerja Mental dengan Metode NASA TLX Pada Operator Kargo di PT. Dharma Bandar Mandala*, Semarang: Universitas Diponegoro, 2016.
- [11] C. Aliferi, "Android Programming Cookbook," 2016. [Online]. Available: <http://enos.itcollege.ee/~jpoial/allalaadimised/reading/Android-Programming-Cookbook.pdf>. [Diakses 03 March 2019].
- [12] Tutorialspoint, "Android Application Development," 2014. [Online]. Available: [https://www.tutorialspoint.com/android/android\\_tutorial1.pdf](https://www.tutorialspoint.com/android/android_tutorial1.pdf). [Diakses 03 March 2019].
- [13] Y. T. W. Tiky, "Software Development Lifecycle," 2016. [Online]. Available: [https://www.cse.ust.hk/~rossiter/independent\\_studies\\_projects/software\\_development/software\\_development\\_report.pdf](https://www.cse.ust.hk/~rossiter/independent_studies_projects/software_development/software_development_report.pdf). [Diakses 03 March 2019].
- [14] V. Tejas dan K. Anand , "Modeling the Mobile Application Development Lifecycle," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2014.
- [15] B. Padmanabhan, "UNIFIED MODELING LANGUAGE OVERVIEW," *Principles of Software Engineering*, 2012.
- [16] A. A. Dahoud dan M. Fezari, *NodeMCU V3 For Fast IoT Application Development*, Al-Zaytoonah University Amman, 2018.
- [17] O. T. Way, "Single-Lead, Heart Rate Monitor Front End (AD8232)," Analog Devices, 2018. [Online]. Available:

- <https://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf>. [Diakses 4 July 2019].
- [18] Electrokit, “SD Card Reader Module,” [Online]. Available: [https://www.electrokit.com/uploads/productfile/41015/41015739\\_-\\_SD\\_Card\\_Module.pdf](https://www.electrokit.com/uploads/productfile/41015/41015739_-_SD_Card_Module.pdf). [Diakses 4 July 2019].
- [19] Flutter Dev, “Flutter,” Google, [Online]. Available: <https://flutter.dev/>. [Diakses 03 March 2019].
- [20] F. Dev, “http | Dart Package,” Google, 3 April 2019. [Online]. Available: <https://pub.flutter-io.cn/packages/http>. [Diakses 4 July 2019].
- [21] F. Dev, “Oscilloscope,” Google, 14 September 2018. [Online]. Available: <https://pub.dartlang.org/packages/oscilloscope>. [Diakses 03 March 2019].
- [22] T. L. Hartono, F. D. Setiaji dan I. Setyawan, “Alat Bantu Analisis Heart Rate Variability,” *Techné Jurnal Ilmiah Elektroteknika*, vol. 12, no. 2, pp. 141-157, 2013.
- [23] Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology, “Heart Rate Variability - Standards of measurement, physiological interpretation, and clinical use,” *European Heart Journal*, vol. 17, pp. 354-381, 1996.
- [24] Firstbeat, “How to Analyze Stress from Heart Rate & Heart Rate Variability: A Review of Physiology,” [Online]. Available: <https://assets.firstbeat.com/firstbeat/uploads/2015/10/How-to-Analyze-Stress-from-Heart-Rate-Variability.pdf>. [Diakses 03 March 2019].
- [25] D. Andersson, *Real-time ECG for Objectives Stress Level Measurement*, Linköping: Linköping University, 2017.

- [26] Tutorialspoint, “Flask Website Application Framework,” 2015. [Online]. Available: [https://www.tutorialspoint.com/flask/flask\\_tutorial.pdf](https://www.tutorialspoint.com/flask/flask_tutorial.pdf). [Diakses 4 July 2019].
- [27] M. Sznajder dan M. Łukowska, “Python Online and Offline ECG QRS Detector based on the Pan-Tomkins algorithm,” 12 July 2018. [Online]. Available: [https://github.com/renatosc/qrs\\_detector](https://github.com/renatosc/qrs_detector). [Diakses 4 July 2019].
- [28] R. Champseix, “Heart Rate Variability analysis,” 27 December 2018. [Online]. Available: <https://github.com/robinchampseix/hrvanalysis>. [Diakses 4 July 2019].
- [29] A. M. Kenneth, “Introduction to Software Testing,” 2012. [Online]. Available: <https://www.cs.colorado.edu/~kena/classes/5828/s12/lectures/05-introtesting.pdf>. [Diakses 03 March 2019].
- [30] L. Luo, “Software Testing Techniques Technology Maturation and Research Strategy,” [Online]. Available: <https://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf>. [Diakses 03 March 2019].
- [31] A. Liana, “Blackbox Testing,” [Online]. Available: [http://rifiana.staff.gunadarma.ac.id/Downloads/files/26083/Teknik\\_Pengujian\\_perangkat\\_Lunak\\_BlackBox.pdf](http://rifiana.staff.gunadarma.ac.id/Downloads/files/26083/Teknik_Pengujian_perangkat_Lunak_BlackBox.pdf). [Diakses 4 July 2019].



## BIODATA PENULIS



Penulis bernama Daud Muhajir, lahir di Sidoarjo pada tanggal 11 Juli 1997, merupakan anak kedua dari tiga bersaudara. Penulis telah menempuh jenjang pendidikan formal di beberapa sekolah, yaitu : SD Negeri Gondolayu Yogyakarta (2003 - 2009), SMP Negeri 12 Yogyakarta (2009 - 2012), dan SMA Negeri 6 Yogyakarta (2012 - 2015). Penulis melanjutkan pendidikan sarjana di Departemen Sistem Informasi Fakultas Teknologi

Informasi dan Komunikasi (FTIK) Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2015 yang terdaftar sebagai mahasiswa dengan NRP 0521154000058.

Selama menjadi mahasiswa, penulis aktif mengikuti beberapa organisasi kemahasiswaan, diantaranya Himpunan Mahasiswa Sistem Informasi pada tahun 2016-2017, Lembaga Dakwah Kampus Jamaah Masjid Manarul Ilmi pada tahun 2016-2018, Lembaga Dakwah Jurusan Kajian Islam Sistem Informasi pada tahun 2016-2018, dan Koperasi Mahasiswa dr.Angka ITS pada tahun 2015-2019.

Selain itu juga mengikut beberapa kegiatan kepanitiaan, seperti Information Systems Expo 2016, berbagai macam pelatihan manajemen, bisnis, dan spiritual, serta kegiatan lainnya.

Pada Juni - Agustus 2018, penulis melaksanakan kegiatan magang di PT Gamatechno Indonesia sebagai *Web Developer* pada Proyek Aglis. Penulis dapat dihubungi melalui email [daud.muhajir11@gmail.com](mailto:daud.muhajir11@gmail.com).

*Halaman ini sengaja dikosongkan*