



HEART & MIND TOWARDS EXCELLENCE

**RANCANG BANGUN ALAT PENDETEKSI TIDUR MENGGUNAKAN  
SENSOR *HEART RATE* PADA *SECURITY GUARD* BERBASIS IOT  
(*INTERNET OF THINGS*)**

**TUGAS AKHIR**

**Program Studi  
S1 Teknik Komputer**

**Oleh :**

**FREEAN ASVIAN**

**15410200069**

---

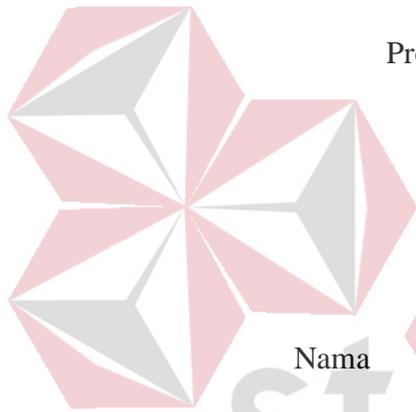
**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2019**

**RANCANG BANGUN ALAT PENDETEKSI TIDUR MENGGUNAKAN  
SENSOR *HEART RATE* PADA *SECURITY GUARD* BERBASIS IOT  
(*INTERNET OF THINGS*)**

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik



**Disusun Oleh :**

Nama : Freean Asvian

NIM : 15.41020.0069

Program : S1 (Strata Satu)

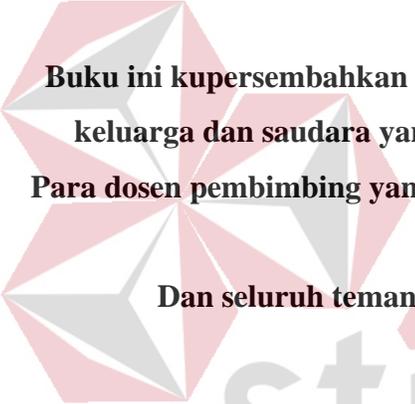
Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2019**

*“Kejarlah dunia. Akhirot tetap no.1”*





**Buku ini kupersembahkan kepada ibu / bapak, adik / kakak, dan semua keluarga dan saudara yang terus mendukung dan mendoakan saya. Para dosen pembimbing yang selalu memberikan motivasi dan bimbingan kepada saya. Dan seluruh teman-teman yang selalu memotivasi saya.**

**stikom**  
SURABAYA

**TUGAS AKHIR**

**RANCANG BANGUN ALAT PENDETEKSI TIDUR MENGGUNAKAN  
SENSOR HEART RATE PADA SECURITY GUARD BERBASIS IOT  
(INTERNET OF THINGS)**

Dipersiapkan dan disusun oleh

**Freean Asvian**

**NIM : 15.41020.0069**

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji

Pada : Juli 2019

**Susunan Dewan Penguji**

Pembimbing

**I. Pauladie Susanto, S.Kom., M.T.**

NIDN. 0729047501

**II. Weny Indah Kusumawati, S.Kom., M.MT.**

NIDN. 0721047201

Pembahas

**I. Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN. 0716117302

Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar sarjana



FAKULTAS TEKNOLOGI  
DAN INFORMATIKA

**Dr. Jusak**  
Dekan Fakultas Teknologi dan Informatika

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

## SURAT PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Freean Asvian  
NIM : 15.41020.0069  
Program Studi : SI Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **RANCANG BANGUN ALAT PENDETEKSI  
TIDUR MENGGUNAKAN SENSOR *HEART  
RATE* PADA *SECURITY GUARD* BERBASIS  
*IOT (INTERNET OF THINGS)***

Menyatakan dengan sesungguhnya bahwa:

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

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Agustus 2019



Freean Asvian

NIM: 15.41020.0069

## ABSTRAK

Satpam merupakan suatu kelompok petugas yang dibentuk oleh badan usaha yang bertujuan untuk melakukan keamanan fisik dalam rangka mengamankan aset dan properti perusahaan. Untuk memastikan keamanan dan mencegah kerugian maupun kerusakan yang disengaja maupun tidak disengaja. Dalam menjaga aset suatu perusahaan, Satpam harus serius dan tidak boleh tertidur karena Satpam memiliki peran utama dalam menjaga keamanan aset suatu perusahaan dan instansi.

Pada penelitian ini penulis membuat rancang bangun alat pendeteksi tidur menggunakan sensor *Heart Rate* pada *security guard* berbasis IoT (*Internet of Things*). Tujuannya agar Satpam lebih fokus dalam bertugas menjaga aset dan properti perusahaan/instansi. Pada rancang bangun tersebut telah ditetapkan penulis bahwa detak jantung orang yang normal sekitar 60-120 Bpm. Orang yang beraktivitas pada umumnya sekitar (detak jantung  $> 80$ ) sedangkan orang yang beristirahat sekitar (detak jantung  $< 80$ ), sehingga penulis menetapkan apabila Satpam dengan detak jantung  $< 80$  maka alat akan mendeteksi tertidur. Nilai detak jantung dan keterangan tertidur atau tidak, akan dikirimkan ke *cloud* Firebase sebagai monitoringnya.

Hasil pengujian yang diperoleh dengan membandingkan Satpam tertidur dengan yang beraktivitas. Dari 6 sampel untuk HR yang tertidur  $< 80$  Bpm dan yang beraktivitas  $> 80$  Bpm. Pengiriman data dari sensor *node* menuju *cloud* Firebase memiliki *delay* dengan rata-rata *delay* 65.9555 detik.

Kata Kunci: *Heart Rate*, *Security Guard*, Firebase, Detak Jantung.

## KATA PENGANTAR

Puji syukur saya panjatkan kepada ALLAH SWT yang telah memberikan rahmat dan kesempatannya sehingga saya bisa menyelesaikan Tugas Akhir ini dengan baik. Penulis mengambil judul “Rancang Bangun Alat Pendeteksi Tidur Menggunakan Sensor *Heart Rate* Pada *Security Guard* Berbasis IOT (*Internet of Things*)” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Institut Bisnis dan Informatika Stikom Surabaya

Dalam menyelesaikan Tugas Akhir ini, saya mendapatkan bantuan serta dukungan dari beberapa pihak yang bersangkutan, maka saya ingin mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmatnya dan hidayahnya penulis dapat menyelesaikan Tugas Akhir ini.
2. Orang Tua dan Seluruh Keluarga yang telah memberikan dukungan dan doanya sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku dosen pembimbing 1 serta Ketua Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya, yang telah banyak membantu serta mendukung kepada penulis sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.
4. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing 2 yang selalu senantiasa memberikan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.

5. Bapak Heri Praktikno, M.T., MTCNA., MTCRE., selaku dosen pembahas yang senantiasa memberikan masukan dan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
6. Seluruh dosen pengajar Program Studi S1 Teknik Komputer yang telah mendidik dan memberikan motivasinya kepada penulis selama menempuh perkuliahan di Institut Bisnis dan Informatika Stikom Surabaya.
7. Teman-teman yang seperjuangan Angkatan 2015, adik, dan kakak angkatan Jurusan S1 Teknik Komputer yang membantu dan memotivasi sehingga penulis bisa menyelesaikan penyusunan buku Tugas Akhir ini.
8. Seluruh pihak yang bersangkutan yang tidak bisa penulis tuliskan satu persatu yang membantu dalam penyusunan buku Tugas Akhir, baik secara langsung maupun tidak langsung.

Penulis berharap semoga penyusunan buku Tugas Akhir ini bisa bermanfaat dan berguna untuk semua pihak, kritik dan saran akan sangat bermanfaat karena penulis mengetahui bahwa di dalam pengerjaan buku Tugas Akhir ini masih jauh dari kesempurnaan.

Surabaya, Agustus 2019

Penulis

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Pembatasan Masalah.....	3
1.4 Tujuan .....	3
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1 Definisi Tidur.....	5
2.2 Sensor <i>Pulse (Heart Rate)</i> .....	6
2.3 Arduino Pro Mini .....	8
2.4 USB to TTL.....	10
2.5 UART.....	10
2.6 <i>Buzzer</i> .....	11
2.7 Baterai Li-Po 1S SyMA .....	12

2.8 <i>Battery Charger TP4056</i> .....	13
2.9 <i>Firestore</i> .....	14
2.10 <i>Wireshark</i> .....	15
<b>BAB III PERANCANGAN SISTEM</b> .....	16
3.1 <i>Metode Penelitian</i> .....	16
3.1.1 <i>Diagram Blok System Utama</i> .....	16
3.1.2 <i>Topologi Jaringan</i> .....	18
3.2 <i>Perancangan Perangkat Keras</i> .....	19
3.3 <i>Perancangan Software</i> .....	21
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	33
4.1 <i>Pengujian Heart Rate dengan Tensimeter Denyut Jantung</i> .....	33
4.1.1 <i>Tujuan Pengujian Heart Rate dengan Tensimeter Denyut Jantung</i> .	33
4.1.2 <i>Peralatan Pengujian Heart Rate dengan Tensimeter Denyut Jantung</i> ..	33
4.1.3 <i>Prosedur Pengujian Heart Rate dengan Tensimeter Denyut Jantung</i> ...	34
4.1.4 <i>Hasil Pengujian Heart Rate dengan Tensimeter Denyut Jantung</i> ....	36
4.2 <i>Pengujian Arduino Pro Mini Dengan Sensor Heart Rate</i> .....	38
4.2.1 <i>Tujuan Pengujian Arduino Pro Mini Dengan Sensor Heart Rate</i> ...	38
4.2.2 <i>Peralatan Pengujian Arduino Pro Mini Dengan Sensor Heart Rate</i>	38
4.2.3 <i>Prosedur Pengujian Arduino Pro Mini Dengan Sensor Heart Rate</i> .	38
4.2.4 <i>Hasil Pengujian Arduino Pro Mini Dengan Sensor Heart Rate</i> ..	40
4.3 <i>Pengujian ESP8266</i> .....	41
4.3.1 <i>Tujuan Pengujian ESP8266</i> .....	41
4.3.2 <i>Peralatan Pengujian ESP8266</i> .....	41

4.3.3	Prosedur Pengujian ESP8266.....	41
4.3.4	Hasil Pengujian ESP8266.....	43
4.4	Pengujian Komunikasi Serial.....	44
4.4.1	Tujuan Pengujian Komunikasi Serial.....	44
4.4.2	Peralatan Pengujian Komunikasi Serial.....	44
4.4.3	Prosedur Pengujian Komunikasi Serial.....	45
4.4.4	Hasil Pengujian Komunikasi Serial.....	45
4.5	Pengujian Firebase.....	46
4.5.1	Tujuan Pengujian Firebase.....	46
4.5.2	Peralatan Pengujian Firebase.....	46
4.5.3	Prosedur Pengujian Firebase.....	47
4.5.4	Hasil Pengujian Firebase.....	47
4.6	Pengujian <i>Delay</i> .....	48
4.6.1	Tujuan Pengujian <i>Delay</i> .....	48
4.6.2	Peralatan Pengujian <i>Delay</i> .....	49
4.6.3	Prosedur Pengujian <i>Delay</i> .....	49
4.6.4	Hasil Pengujian <i>Delay</i> .....	51
<b>BAB V PENUTUP.....</b>		<b>53</b>
5.1	Kesimpulan.....	53
5.2	Saran.....	53
<b>DAFTAR PUSTAKA.....</b>		<b>55</b>
<b>BIODATA PENULIS.....</b>		<b>68</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Sensor <i>Heart Rate</i> .....	7
Gambar 2. 2 Arduino Pro Mini dan Pemetaan Pinnya.....	8
Gambar 2. 3 Bentuk Fisik USB to TTL .....	10
Gambar 2. 4 Bentuk Fisik <i>Buzzer</i> .....	11
Gambar 2. 5 Bentuk Fisik Baterai Li-Po 1S SyMA.....	12
Gambar 2. 6 Bentuk Fisik Modul TP4056.....	14
Gambar 3. 1 Diagram Blok Sistem Utama.....	16
Gambar 3. 2 Topologi Jaringan.....	18
Gambar 3. 3 Diagram Blok Perangkat Keras Alat Pendeteksi Tidur.....	19
Gambar 3. 4 Disain <i>Casing</i> Perangkat Keras.....	21
Gambar 3. 5 <i>Flowchart</i> Pemrograman Arduino Pro Mini .....	22
Gambar 3. 6 Tampilan Awal Arduino IDE.....	23
Gambar 3. 7 Rangkaian <i>Heart Rate</i> , Arduino Pro Mini dan USB to TTL .....	24
Gambar 3. 8 Tampilan <i>Board</i> Arduino Pro Mini .....	24
Gambar 3. 9 Tampilan Serial Monitor .....	25
Gambar 3. 10 <i>Flowchart</i> ESP8266 .....	26
Gambar 3. 11 Skematik ESP8266 dan USB to TTL.....	26
Gambar 3. 12 Mengganti <i>Board Generic ESP8266 Module</i> .....	27
Gambar 3. 13 Tampilan <i>Flash</i> ESP8266.....	27

Gambar 3. 14 Tampilan Awal Firebase .....	28
Gambar 3. 15 Tampilan Membuat <i>New Projek</i> .....	29
Gambar 3. 16 Tampilan Projek <i>Database</i> .....	29
Gambar 3. 17 Tampilan Aturan Firebase.....	30
Gambar 3. 18 Tampilan Akun Layanan Firebase .....	31
Gambar 3. 19 Tampilan <i>Link Database</i> Firebase.....	31
Gambar 4. 1 Skematik Arduino Pro Mini dengan Sensor <i>Heart Rate</i> .....	34
Gambar 4. 2 <i>Tensimeter</i> Detak Jantung.....	35
Gambar 4. 3 Skematik <i>Prototype Sensor Node</i> .....	39
Gambar 4. 4 Tampilan Serial Monitor .....	40
Gambar 4. 5 Skematik ESP8266 dengan USB to TTL.....	41
Gambar 4. 6 Tampilan <i>Board</i> Arduino IDE.....	43
Gambar 4. 7 Tampilan <i>Connected</i> ESP8266.....	44
Gambar 4. 8 Tampilan Hasil Pengujian <i>Heart Rate</i> .....	46
Gambar 4. 9 Skematik Rangkaian Alat.....	47
Gambar 4. 10 Tampilan Hasil Pengujian pada Firebase.....	48
Gambar 4. 11 Tampilan Awal <i>Wireshark</i> .....	49
Gambar 4. 12 Tampilan <i>Capture</i> Pada Jaringan TCP.....	50
Gambar 4. 13 <i>Delay</i> Yang Tersimpan Pada <i>Excel</i> .....	51

## DAFTAR TABEL

	Halaman
Tabel 4. 1 Tabel Pengujian <i>Prototype</i> Dengan <i>Tensimeter</i> .....	36
Tabel 4. 2 Tabel Pengujian <i>Delay</i> .....	51



## DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1 Program Arduino Pro Mini .....	57
LAMPIRAN 2 Program ESP8266 .....	58
LAMPIRAN 3 <i>Datasheet Sensor Heart Rate</i> .....	60



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pekerja keamanan (*security guard*) atau yang biasa dikenal dengan Satpam adalah suatu kelompok petugas yang dibentuk oleh instansi/badan usaha/proyek untuk melakukan keamanan fisik dalam rangka mengamankan aset, instansi, proyek bangunan, properti atau tempat pemantauan peralatan untuk memastikan keamanan dan mencegah kerugian maupun kerusakan yang disengaja. Satpam memiliki peran yang utama dalam menjaga aset milik perusahaan/instansi, apabila seorang Satpam tidak serius dalam pekerjaannya maka perusahaan/instansi akan mendapatkan kerugian. Bukan hanya perusahaan/instansi yang membutuhkan Satpam akan tetapi Universitas, sekolah, tempat ibadah, juga membutuhkannya demi menjaga aset-aset yang berharga. Satpam yang bertugas pada malam hari sangat rawan dengan tertidur.

Untuk mengawasi agar Satpam tidak tertidur dan fokus dalam kerjanya, pimpinan perusahaan/instansi yang bersangkutan harus memonitoring Satpamnya agar tidak terjadi hal yang tidak diinginkan. Biasanya seorang pimpinan cukup melihat rekaman dari CCTV, akan tetapi dengan keterbatasan CCTV yang tidak bisa semua tempat dapat *direcord* dan tidak bisa dimonitoring secara *realtime* dari jauh maka pengawasan tersebut kurang efisien.

Selain CCTV, penelitian lainnya yaitu alat monitoring cek *point* Satpam berfungsi untuk mengawasi kerja Satpam untuk pengontrolan ruangan (gedung) pada perusahaan yang membutuhkan penjadwalan teratur. Satpam dalam melaksanakan

kegiatan pengontrolan akan diberikan *remote control* dan setiap melakukan pengontrolan ke ruangan, harus memasukkan ID-nya, sehingga laporan pengontrolan ruangan mana dan jam berapa Satpam melakukan pengontrolan dapat diketahui. ID yang telah dimasukkan Satpam akan diterima oleh *receiver* dan diteruskan ke server pusat disuatu ruangan yang khusus. (SOFYAN, 2006)

Implementasi IoT telah banyak dilakukan dibanyak bidang seperti sistem pengontrolan dan monitoring. Segala sesuatu mengenai IoT akan terkoneksi dengan internet menggunakan *cloud* komunikasi, salah satunya adalah *Firebase*.

*Firebase* menyediakan fitur untuk perangkat IoT yang bisa dibilang lengkap. Salah satu fiturnya yaitu *Firebase Realtime Database* yang mana layanan ini menyediakan antarmuka pemrograman aplikasi yang memungkinkan data dari aplikasi disinkronkan di seluruh *client* yang tersimpan di *cloud Firebase*.

Pada Tugas Akhir ini, penulis memberikan solusi atas permasalahan sebagaimana yang telah dibahas diatas, yaitu melalui penerapan berbasis IoT (*Internet Of Things*) agar data dari sensor *Heart Rate* bisa dibaca dan dimonitoring pada jarak yang jauh untuk memberikan informasi tentang status *security guard* yang berjaga dan *output Buzzer* untuk membangunkan Satpam agar tidak tertidur.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas dapat dirumuskan permasalahannya, yaitu:

1. Bagaimana membedakan orang yang tidur dan yang beraktivitas dengan sensor *Heart Rate*?

2. Bagaimana menampilkan data denyut nadi dari sensor *Heart Rate* menuju web *server* Firebase?
3. Bagaimana mengintegrasikan sensor *Heart Rate* dengan Arduino Pro Mini?

### 1.3 Pembatasan Masalah

Dalam perancangan ini, terdapat beberapa batasan masalah, antara lain:

1. Pengujian untuk Satpam laki-laki dan perempuan yang sehat serta tidak mempunyai riwayat penyakit dalam.
2. Perangkat yang digunakan yaitu Arduino Pro Mini dan sensor *Heart Rate*.
3. Web *server* yang digunakan adalah Firebase.

### 1.4 Tujuan

Tujuan perancangan alat ini adalah:

1. Mengetahui perbedaan orang yang tidur dan yang beraktivitas dengan sensor *Heart Rate*.
2. Mengetahui data sensor *Heart Rate* yang terkirim menuju web *server* Firebase
3. Mengintegrasikan sensor *Heart Rate* dengan Arduino Pro Mini.

### 1.5 Sistematika Penulisan

## BAB I PENDAHULUAN

Pada bab ini dituliskan penjelasan latar belakang permasalahan sistem pencatatan *Heart Rate* yang ada pada saat ini dan apa yang telah dilakukan pada

penelitian sebelumnya. Penulis menjelaskan tujuan untuk dilanjutkannya penelitian tersebut agar alat yang diharapkan bisa lebih simple dan praktis dalam pemakaiannya.

## **BAB II LANDASAN TEORI**

Pada bab ini disampaikan perihal tentang landasan teori sistem, aplikasi yang digunakan, perangkat-perangkat yang digunakan, serta konsep-konsep dalam menyelesaikan masalah yang sesuai dengan topik dan fokus diantaranya seperti Definisi Tidur, Sensor *Pulse (Heart Rate)*, Arduino Pro Mini, USB to TTL, UART, *Buzzer*, Baterai Li-Po 1S SyMA, *Battery Charger TP4056*, Firebase, dan *Wireshark*.

## **BAB III PERANCANGAN SISTEM**

Pada bab ini dijelaskan tentang rancangan sistem yang dibuat mencakup *Hardware* (rangkaian), *software (coding)*, dan *database* yang digunakan untuk menyelesaikan permasalahan Tugas Akhir diantaranya seperti Metode Penelitian, Perancangan Perangkat Keras, dan Perancangan *Software*.

## **BAB IV HASIL DAN PEMBAHASAAN**

Pada bab ini dibahas mengenai hasil dari penelitian, hasil pengamatan, dan hasil uji dari penelitian terhadap sistem yang dibuat untuk menyelesaikan permasalahan Tugas Akhir diantaranya seperti Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung, Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate*, Pengujian ESP8266, Pengujian Komunikasi Serial, Pengujian Firebase, dan Pengujian *Delay*.

## **BAB V PENUTUP**

Pada bab ini disampaikan tentang kesimpulan hasil analisis dari rancangan sistem yang dibuat dengan tujuan menjawab penelitian yang dilakukan, serta saran yang penulis ingin berikan untuk kedepannya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Definisi Tidur**

Tidur adalah suatu aktivitas keadaan bawah sadar dimana orang tersebut dapat dibangunkan dengan pemberian rangsang, baik itu pemberian rangsang sensorik maupun rangsang lainnya. Tidur memiliki fungsi restorasi yang penting untuk termoregulasi dan cadangan energi tubuh. Tidur merupakan proses normal yang bersifat aktif, teratur, berulang, reversibel yang dibutuhkan oleh otak untuk menunjang proses fisiologinya.

Semua makhluk hidup mempunyai irama kehidupan yang sesuai dengan beredarnya waktu dalam siklus 24 jam. Pada saat tidur tenaga yang hilang dipulihkan dan terjadi pelepasan otot. Perekaman listrik dari permukaan otak luar kepala dapat menunjukkan adanya aktivitas listrik yang terus-menerus timbul dalam otak. Intensitas dan pola aktivitas listrik sangat ditentukan oleh besarnya derajat ekstasi otak sebagai akibat dari keadaan tidur, siaga, dan penyakit otak serta psikosis. Fisiologi tidur dapat diterangkan melalui gambaran aktivitas sel-sel otak selama tidur. Aktivitas tersebut dapat direkam dalam alat EEG. Untuk merekam tidur, cara yang digunakan adalah dengan EEG *Polygraphy*.

Terdapat beberapa jenis klasifikasi gelombang otak, diantaranya adalah sebagai berikut:

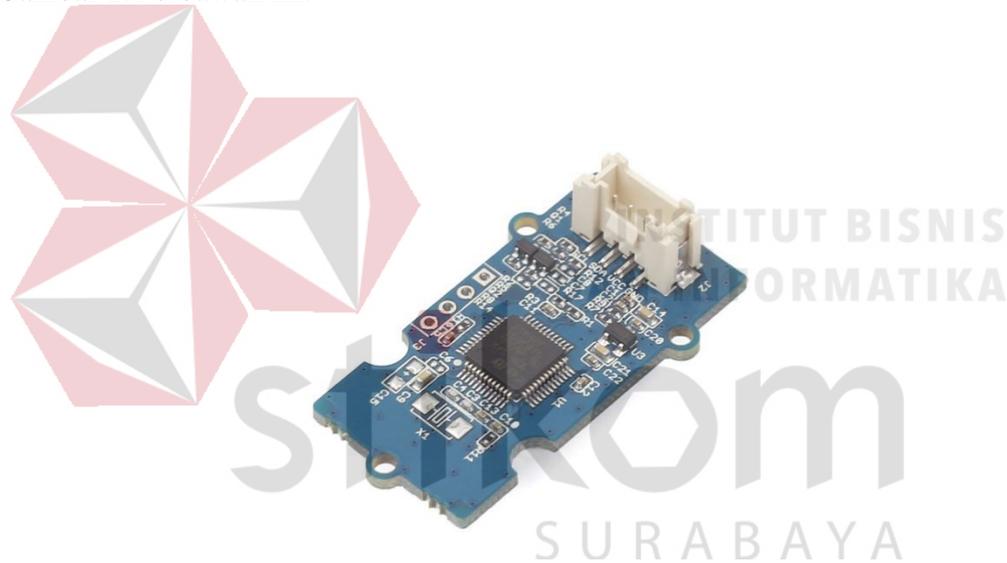
1. **Gelombang Alfa**, memiliki frekuensi 8 – 13 Hz, dan gelombang amplitude antara 10 – 15 mV. Gelombang alfa paling jelas terdapat pada daerah oksipital atau parietal.
2. **Gelombang Teta**, memiliki frekuensi 4 – 7 Hz, dan gelombang amplitudo bervariasi. Gelombang teta timbul di daerah parietal dan temporal anak-anak.
3. **Gelombang Beta**, memiliki frekuensi 14 Hz bahkan lebih, dan gelombang amplitude kecil yaitu rata-rata 25 mV. Gelombang beta selalu terekam di daerah parietal dan frontal kulit kepala selama aktivitas ekstra pada sistem saraf pusat.
4. **Gelombang Delta**, memiliki semua gelombang EEG dengan frekuensi di bawah 3,5 Hz, dengan amplitude serta lokalisasi bervariasi. Gelombang delta timbul pada korteks dan tak bergantung pada aktivitas di regio bawah otak. (Dita, 2009)

## 2.2 **Sensor Pulse (Heart Rate)**

Sensor *pulse (Heart Rate)* bekerja dengan memanfaatkan cahaya. Ketika sensor diletakkan dipermukaan kulit, cahaya yang diserap atau dipantulkan oleh organ dan jaringan seperti kulit, tulang, otot, dan darah. Sebagian cahaya akan melewati jaringan tubuh yang cukup tipis. Apabila jantung memompa darah melalui tubuh, dari setiap denyut yang terjadi, timbul gelombang *pulse* (jenis seperti gelombang kejut) yang bergerak sepanjang arteri dan menjalar ke jaringan kapiler dimana sensor *pulse* terpasang. Sensor *pulse (Heart Rate)* dirancang untuk mengukur *Inter Beat Interval* (IBI). IBI merupakan selang waktu pada denyut jantung dalam milidetik dengan waktu momen sesaat dari jantung berdetak. Adapun BPM adalah setiap detik dari rata-rata setiap 10 kali IBI. Jadi, ketika mikrokontroler dinyalakan dan berjalan dengan sensor

*pulse* yang disambungkan ke *pin* analog yang sudah disediakan. (SeedTechnology, 2018)

Cara kerja dari sensor *Heart Rate* adalah salah satu jari tangan dimasukkan ke blok *heartbeat transducer*, dimana cahaya LED menembus permukaan kulit dan daging lalu diterima oleh LDR. Frekuensi aliran darah yang dipompa oleh jantung akan menimbulkan gelombang (*pulse*), LDR akan menerima sinyal dari LED berdasarkan gelombang dari aliran darah (Darwinsa, 2011). Sensor *Heart Rate* bisa dilihat pada Gambar 2.1 dibawah ini.



Gambar 2. 1 Sensor *Heart Rate*  
(Sumber : [http://wiki.seeedstudio.com/Grove-Finger-clip\\_Heart\\_Rate\\_Sensor/#with-mbed](http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/#with-mbed))

Spesifikasi *Heart Rate*:

1. Konsumsi daya sangat rendah, mode hemat daya selama gerakan tanpa sentuhan.
2. STM32F103 terintegritas.
3. Area sensor detak jantung hanya 3,0 X 4,7mm.

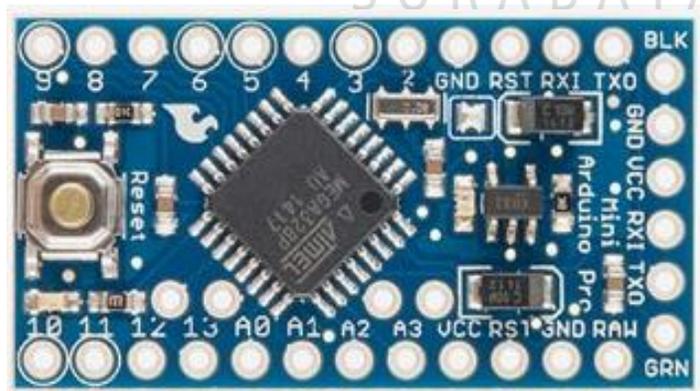
4. Antarmuka I2C dan SWD yang dicadangkan.

5. Suhu kerja  $-20 \sim +60^{\circ}\text{C}$ .

### 2.3 Arduino Pro Mini

Arduino Pro Mini adalah suatu mikrokontroler dengan Atmega328, memiliki 6 *input* analog, tombol *reset*, *resonator on-board*, dan lubang untuk *pin header*, 14 digital *pin input / output* 6 diantaranya sebagai *output* PWM serta memiliki *pin* I2C, 8 buah *input* analog yang diberi tanda dengan A0 – A7. Masing-masing analog memiliki resolusi 10 bits (1024 nilai). Arduino Pro Mini memiliki 2 versi yaitu 3.3V dengan 8 MHz dan 5V dengan 16 MHz.

Arduino ini dirancang dan diproduksi oleh *SparkFun Electronics*, mendukung dengan kabel FTDI atau *board breakout* terhubung ke-6 *pin header*, atau dengan tegangan 3,3V atau 5V tergantung pada modelnya pada *pin* VCC. Di *board* Arduino Pro Mini ada tegangan regulator yang dapat menerima daya sampai 12V DC. (Putra, 2018)



Gambar 2. 2 Arduino Pro Mini dan Pemetaan Pinnya  
(Sumber : <https://www.arduino.cc>)

Adapun spesifikasi Arduino Pro Mini sebagai berikut:

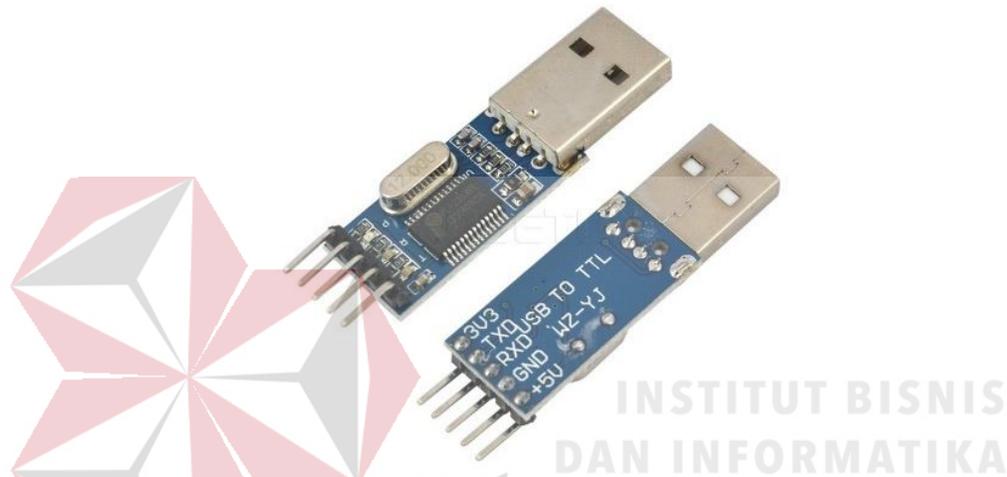
1. Mikrokontroler : ATmega328P.
2. Tegangan Operasi : 5V dan 3.3V (2 model).
3. *Pin Input* Analog : 6 buah.
4. *Pin* Digital I/O : 14 buah dan 6 *pin* PWM.
5. Arus DC per *pin* I/O : 40 mA.
6. SRAM : 2 KB.
7. *Flash Memory* : 32 KB dan 0.5 KB untuk *bootloader*.
8. Berat : 5 gram.
9. EEPROM : 1 KB.
10. Ukuran : 33 mm x 18 mm.
11. *Clock Speed* : 8 Mhz (model 3.3V) dan 16 Mhz (model 5V).

Beberapa fungsi pin Arduino Pro Mini diantaranya:

1. Serial: *pin* 0 (RX) digunakan untuk *receiver* dan *pin* 1 (TX) digunakan untuk *transmitter*. *Pin* ini terhubung sesuai *chip* FTDI USB to TTL *Serial*.
2. PWM: *pin* 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8 – bit dengan menggunakan fungsi `analogWrite()`.
3. LED: *pin* 13 terhubung *built – in* LED yang dikendalikan oleh *pin* digital.
4. *External Interrupt*: *pin* 2 dan *pin* 3 digunakan untuk mengaktifkan *interrups*.
5. SPI: *pin* 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*. (Firdausi, 2018)

## 2.4 USB to TTL

USB to TTL adalah USB *to serial converter* yang memungkinkan konektivitas antara USB dan *interface* UART. Terdiri dari 5 *pin*, yaitu *pin* 3.3V, Rx, Tx, GND, dan 5V. Dibutuhkan ketika akan menghubungkan antara Arduino Pro Mini dengan laptop, baik itu untuk komunikasi ataupun *upload sketches*.



Gambar 2. 3 Bentuk Fisik USB to TTL

(Sumber : <http://www.elektrojo.com/home/1689-pl2303-module-usb-to-ttl.html>)

## 2.5 UART

Komunikasi serial adalah suatu komunikasi yang mengirimkan data per bit secara bergantian dan berurutan. Untuk pengiriman data, komunikasi serial lebih lambat dibanding komunikasi paralel. Namun komunikasi serial dapat mengurangi jumlah keluaran dan jumlah kabel yang dibutuhkan dalam komunikasinya (Adam, 1980). Komunikasi serial terdiri dari 2 macam, yaitu *serial synchronous* dan *serial asynchronous*. Pada *serial synchronous* sebelum melakukan koneksi diperlukan sinkronisasi *clock* antara pengirim dan penerima, sedangkan *serial asynchronous* tidak. *Serial asynchronous* terdiri dari *transmitter* (TX) dan *receiver* (RX) untuk kabel

transmisi datanya, berbeda dengan *serial synchronous* yang terdapat di protokol SPI (*Serial Peripheral Interface*) dan I2C (*Inter-Integrated Circuit*) dimana protokolnya membutuhkan dua kabel untuk transmisi data, yaitu untuk data dan *clock*.

## 2.6 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik kedalam atau keluar tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. (WWW.FARNELL.COM, 2016) *buzzer* biasanya digunakan sebagai indikator pada suatu alat yang dibuat adapun Gambar *buzzer* bisa dilihat pada Gambar 2.4



Gambar 2. 4 Bentuk Fisik *Buzzer*

## 2.7 Baterai Li-Po 1S SyMA

Baterai Li-Po adalah singkatan dari *Lithium Polymer*, baterai ini bersifat cair (*Liquid*), menggunakan elektrolit polimer yang padat dan mampu menghantarkan daya lebih cepat. Baterai ini hasil dari pengembangan dari *Lithium Ion* dan termasuk baterai yang ramah lingkungan.

Beberapa kelebihan baterai Li-Po:

1. Ramah lingkungan.
2. Lebih aman.
3. Fleksibel bisa dibuat berdasarkan kebutuhan.
4. Lebih ringan.

Beberapa kekurangan baterai Li-Po:

1. Harga baterai yang mahal.
2. Butuh perawatan khusus untuk isi ulang, seperti jangan sampai habis baru diisi ulang.
3. Biaya manufaktur mahal.
4. Usia baterai pendek.



Gambar 2. 5 Bentuk Fisik Baterai Li-Po 1S SyMA

Spesifikasi baterai LiPo 1S SyMA:

1. Tegangan: 1S1P / 1 Cell / 3.7V.
2. *Ampere*: 1200mAh.
3. Pengeluaran: 25°C.
4. Dimensi: 55x28x9 mm.
5. Berat: 27g.

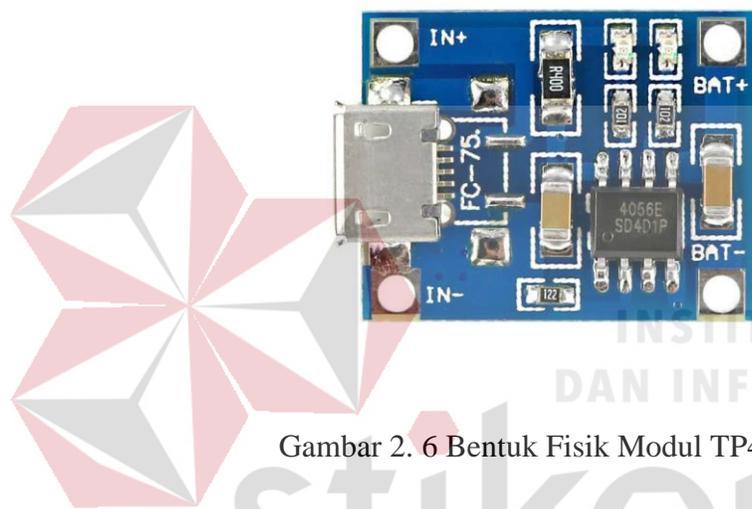
## 2.8 *Battery Charger TP4056*

TP4056 dapat digunakan untuk mengisi baterai jenis *Lithium*. *Input* menggunakan *micro-USB* (seukuran *handphone*) yang bertegangan 4.5-5V. Memiliki 2 lampu indikator, warna merah ketika sedang cas, dan warna biru ketika baterai sudah penuh. Pemasangan positif dan negatif *output* BAT+/- pada modul ke baterai harus benar-benar diperhatikan, karena jika pemasangan terbalik akan merusak *chip*. Modul ini sangat sederhana dan simpel dengan komponen SMD, yang membuat modul ini dapat langsung digunakan tanpa tambahan apapun. Selain bentuk yang simpel, modul ini memiliki sistem proteksi yang baik dan *charging* dengan presisi yang tinggi. Terdapat 2 buah LED sebagai indikator untuk mengetahui status baterai penuh atau habis.(Abduracman, 2018)

Adapun spesifikasi dari modul *Charger TP4056* adalah:

1. Arus *charging* 1A (*maximal*).
2. Kepresisian *charging* 1.5%.
3. Metode *charging* *Linier*.

4. 2 buah LED sebagai indikator, yaitu merah sebagai proses *charging* dan biru *full charging*.
5. Dapat beroperasi pada temperature  $-10^{\circ}\text{C}$  sampai  $+85^{\circ}\text{C}$ .
6. Tegangan *output* 4.2V.
7. Tegangan *input* berkisaran dari 4.5V – 5.5V.



Gambar 2. 6 Bentuk Fisik Modul TP4056

## 2.9 Firebase

Firebase adalah sebuah layanan infrastruktur *backend-as-a-service* (BaaS) yang diakuisisi oleh google pada oktober 2014 (Tamplin J, 2018). Firebase memiliki fitur *Analytics, Cloud Messaging, Authentication, Realime Database, Storage, Hosting, Test Lab, Crsah Reporting, Notifications, Remote Config, App Indexing, Dynamic Links, Invites, AdWords, dan Ad Mob*. Semua fitur tersebut dikemas dalam sebuah SDK Firebase tunggal sehingga dengan kemudahan yang ditawarkan para pengembang perangkat lunak dapat fokus untuk memecahkan masalah *costumer*

melalui perangkat lunak yang dibuatnya dan tidak memerlukan banyak waktu dalam membangun infrastruktur yang kompleks. (Bashofi & Abidin, 2018).

Dua fitur yang paling menarik dari firebase adalah *Remote Config* dan *Firestore Database*. *Remote Config* adalah fitur yang memungkinkan developer mengganti beberapa konfigurasi aplikasi android/iOS tanpa harus memberikan *update* aplikasi *via Play Store/App Store*. Salah satu konfigurasi yang bisa dimanipulasi adalah seperti warna dan tema aplikasi.

## 2.10 Wireshark

*Wireshark* merupakan *software* untuk melakukan analisis lalu lintas pada sebuah jaringan komputer dan memiliki fungsi-fungsi yang berguna untuk professional jaringan, peneliti, administrator jaringan bahkan pengembangan piranti lunak jaringan. *Software* ini dapat menangkap paket-paket data, menghitung *delay*, paket *loss* dan informasi yang berjalan pada suatu jaringan. Berbagai jenis format protokol dengan mudah ditangkap dan dianalisis. Oleh karena itu tak jarang *software* ini dinamakan dengan *sniffing* (memperoleh informasi penting seperti *password* dan *email* atau *account-account* yang lain) dengan menangkap paket-paket yang berjalan didalam jaringan dan menganalisisnya (Ids, n.d.). Cara kerja *wireshark* ada 2 yaitu:

1. Merekam semua paket yang melewati *wifi* maupun LAN.
2. Hasil rekaman dapat dianalisis melalui beberapa protokol seperti *tcp*, *http*, *udp*, dan sebagainya.

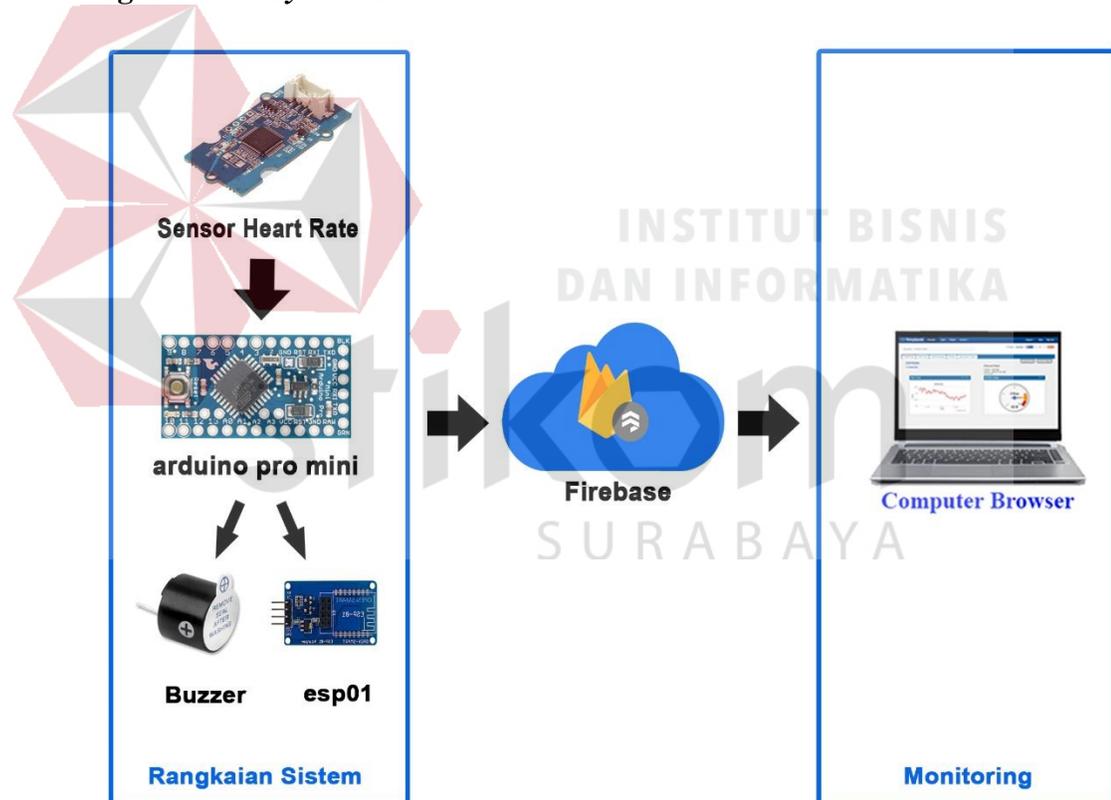
## BAB III

### PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Pada bab ini membahas tentang bagaimana proses komunikasi dan model sistem yang akan diterapkan:

##### 3.1.1 Diagram Blok *System* Utama



Gambar 3. 1 Diagram Blok Sistem Utama

Pada Gambar 3.1, dapat dilihat bahwa sensor *pulse (Heart Rate)* akan dihubungkan dengan Arduino Pro Mini. Data dari sensor *Heart Rate* dikirim ke

mikrokontroler Arduino Pro Mini melalui *pin* I2C (A4 dan A5), data yang diterima akan diproses dengan pemilahan  $HR < 80$  maka menampilkan keterangan bahwa Satpam tertidur begitu sebaliknya apabila  $HR > 80$  maka menampilkan keterangan Satpam beraktivitas. Setelah data dipilah Arduino lalu dikirimkan ke *cloud* Firebase melalui modul ESP8266. Apabila  $HR < 80$  maka *Buzzer* akan menyala yang tujuannya untuk membangunkan Satpam yang tertidur, dan dapat dimonitoring melalui *smartphone* / komputer.

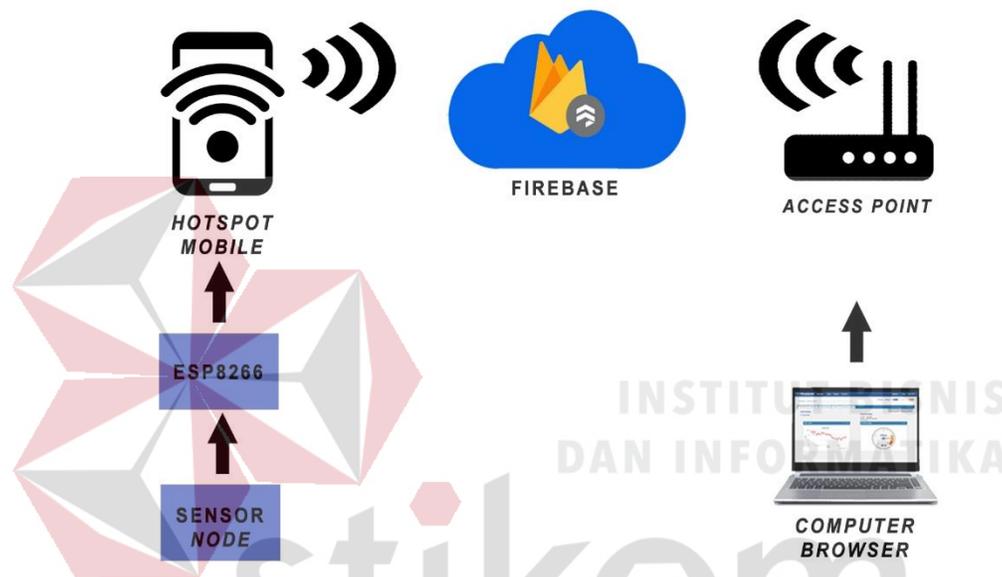
Pengujian yang dilakukan yaitu pengambilan data denyut nadi dari 6 orang dengan syarat tidak memiliki riwayat penyakit dalam. Contoh penyakit dalam seperti diabetes dan hepatitis, seseorang yang memiliki riwayat diabetes cenderung denyut nadinya rendah. (Hidayah, 2012) Sedangkan hepatitis denyut nadinya cenderung tinggi. (Suoth et al., 2014) dan berbagai penyakit dalam lainnya yang bisa mempengaruhi denyut jantung/nadi. Dengan memasang sensor *Heart Rate* pada pergelangan tangan *security guard* yang terhubung pada Arduino Pro Mini. hasil data yang diambil akan dikirimkan menuju *cloud* Firebase melalui modul *wifi* ESP8266. Untuk pengujian akurasi data yang dibaca akan dipresisikan dengan *tensimeter* denyut jantung.

Kemudian dilakukan pengujian kesesuaian data yang diambil oleh alat yang dibuat (sensor *Heart Rate*) dan diproses oleh Arduino Pro Mini dengan cara:

1. Memasang alat yang dibuat (sensor *Heart Rate*) pada Satpam kemudian mengambil kesimpulan apakah data yang diambil sudah sesuai dengan hasil yang diharapkan dan tidak ada kesalahan.

2. Mencoba alat yang dibuat kepada beberapa Satpam sebagai sampel dan mengambil kesimpulan apakah data yang diharapkan sudah sesuai.

### 3.1.2 Topologi Jaringan

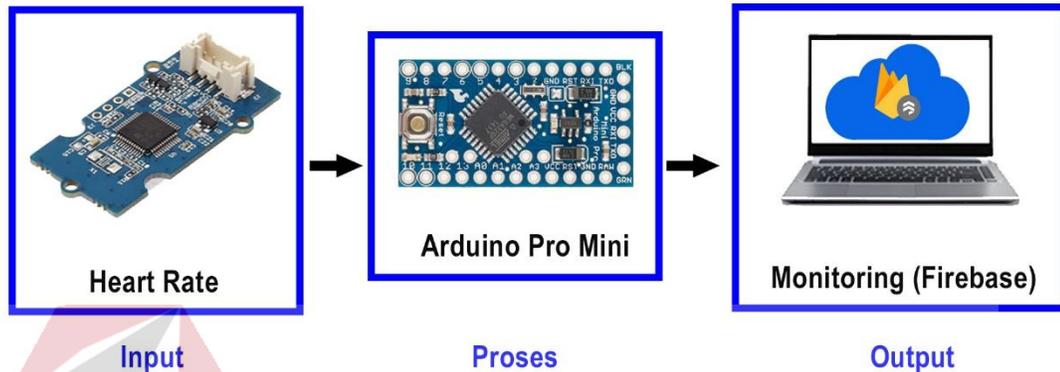


Gambar 3. 2 Topologi Jaringan

Pada Gambar 3.2, dapat dilihat bahwa sensor *node* mengirimkan data ke ESP8266 yang terkoneksi dengan *hotspot mobile*. Data disimpan di *cloud* Firebase, untuk monitoring data yang tersimpan dalam Firebase menggunakan *computer browser* yang terkoneksi dengan *access point*.

### 3.2 Perancangan Perangkat Keras

Gambar blok diagram sistem deteksi tidur dengan sensor *Heart Rate* ditunjukkan pada Gambar 3.3 berikut ini:



Gambar 3.3 Diagram Blok Perangkat Keras Alat Pendeteksi Tidur

Berikut penjelasan setiap bagian dari blok diagram alat pendeteksi tidur:

1. *Input.*

*Heart Rate* sensor, sebagai sensor untuk mengambil data denyut jantung Satpam.

2. *Proses.*

Arduino Pro Mini, sebagai pengelola data yang diterima dari sensor *Heart Rate* sehingga didapatkan nilai *output*.

3. *Output.*

Firestore, sebagai monitoring *realtime database* yang bisa menyimpan dan memonitoring data yang ditampilkan melalui *web server*.

*Buzzer*, sebagai notifikasi *alarm* kepada Satpam apabila nilai *output* yang didapatkan terdeteksi mengantuk.

Komponen yang digunakan untuk rancang bangun alat pendeteksi tidur pada *security guard* sebagai berikut:

1. Sensor *Heart Rate*.
2. Arduino Pro Mini.
3. ESP8266.
4. *Battery*.
5. *Buzzer*.
6. Laptop atau PC.
7. *Casing*.

Penjelasan dari masing-masing komponen sebagai berikut:

1. Sensor *Heart Rate*: sensor untuk deteksi detak jantung yang mana pengambilannya dilakukan dengan menempelkannya ke ujung jari, pergelangan tangan, dan daun telinga.
2. Arduino Pro Mini: mikrokontroler yang digunakan pada perangkat keras, dan disesuaikan dengan disain *casing* yang sudah ditentukan.
3. *Buzzer*: apabila data dari *Heart Rate* ( $HR > 80$ ) maka *Buzzer on*.
4. ESP8266: modul yang digunakan untuk transmisi data ke *cloud* Firebase.
5. *Casing*: disain yang digunakan untuk *packaging* perangkat keras yang dirancang sedemikian rupa, harapannya agar Satpam yang berjaga tidak kesulitan. Disain ditunjukkan pada Gambar 3.4 sebagai berikut:

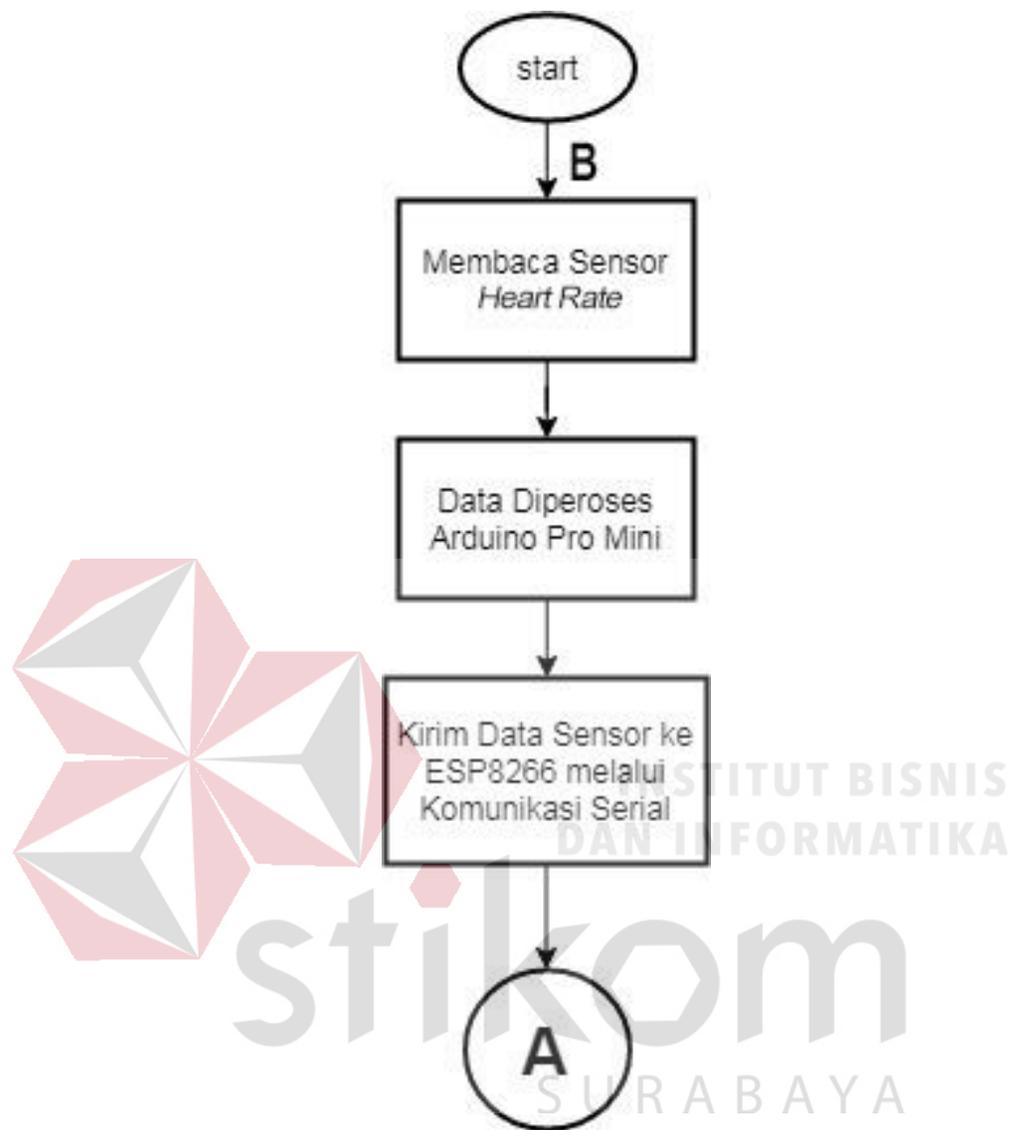


Gambar 3. 4 Disain Casing Perangkat Keras

### 3.3 Perancangan Software

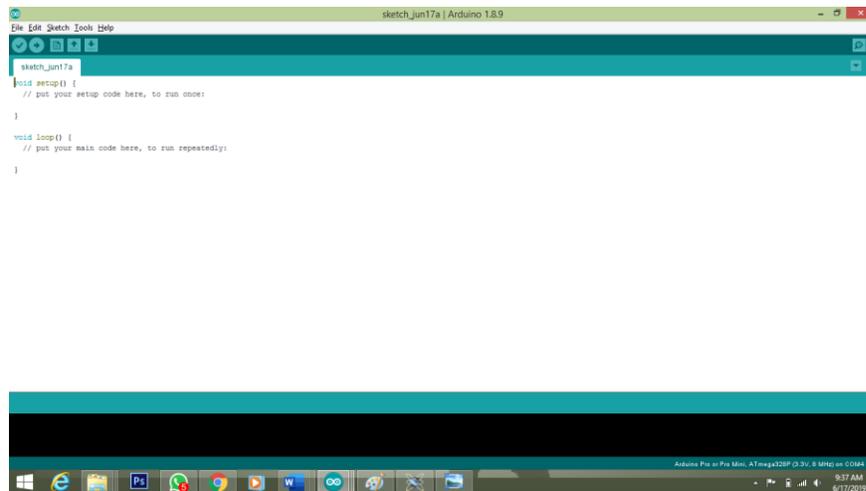
#### A. Flowchart Arduino Pro Mini

Dalam proses pengambilan data sensor *Heart Rate* bisa dilakukan melalui langkah-langkah dari proses yang ditunjukkan pada Gambar 3.5 di bawah ini:



Gambar 3. 5 *Flowchart* Pemrograman Arduino Pro Mini

1. Menjalankan aplikasi Arduino IDE seperti Gambar 3.6 di bawah ini:

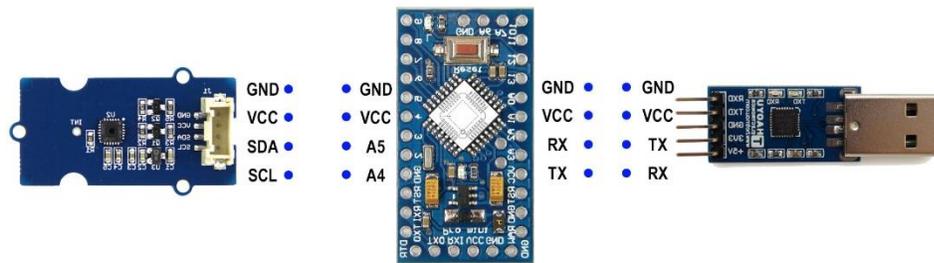


Gambar 3. 6 Tampilan Awal Arduino IDE

2. Memasukkan *library* I2C, Komunikasi Serial, dan *source code* yang sudah disediakan pada *website pulse Heart Rate* sensor. Berikut penggalan kodingnya:

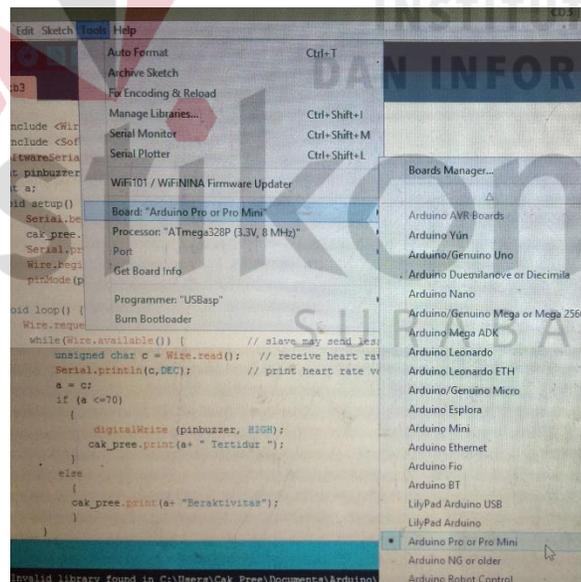
```
#include <Wire.h>
#include <SoftwareSerial.h>
//SoftwareSerial cak_pree(11,12);
int pinBuzzer = 13;
String hasil;
void setup() {
    Serial.begin(9600);
    Serial.println("Heart Rate sensor:");
    Wire.begin();
    pinMode(pinBuzzer, OUTPUT);
}
void loop() {
Wire.requestFrom(0xA0 >> 1, 1);
// request 1 bytes from slave device
// while(Wire.available()) { // slave may send less than
// requested
unsigned char c = Wire.read(); // receive Heart Rate value
// (a byte)
Serial.println(c, DEC); // print Heart Rate value
```

3. Merangkai sensor *Heart Rate* dengan Arduino Pro Mini dan USB to TTL, seperti Gambar 3.7 di bawah ini:



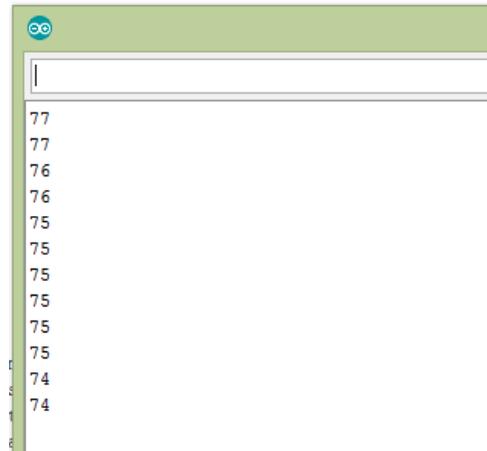
Gambar 3. 7 Rangkaian *Heart Rate*, Arduino Pro Mini dan USB to TTL

4. Mengganti *board* Arduino Pro Mini dan *port com* dengan cara membuka *Tools – Board – Arduino Pro Mini*, dan men-*setting port* sesuai dengan yang digunakan, seperti Gambar 3.8 di bawah ini:



Gambar 3. 8 Tampilan *Board* Arduino Pro Mini

5. Meng-*upload* program dan membuka serial monitor untuk melihat data dari sensor *Heart Rate*, seperti Gambar 3.9 di bawah ini:



Gambar 3. 9 Tampilan Serial Monitor

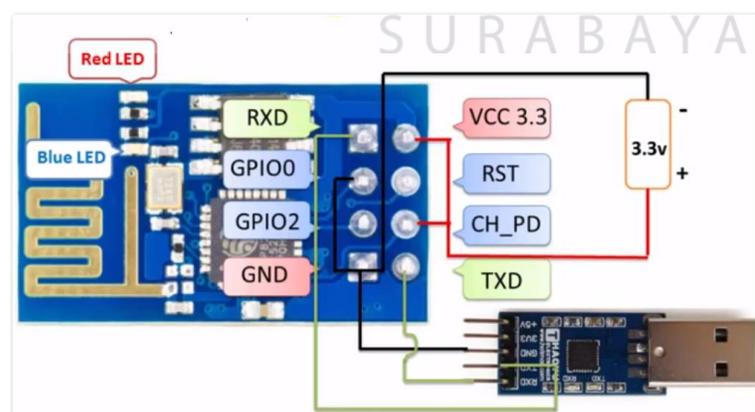
B. *Flowchart* pada ESP8266

Dalam proses pengiriman data sensor *Heart Rate* bisa dilakukan melalui langkah-langkah dari proses yang ditunjukkan pada Gambar 3.10 di bawah ini:



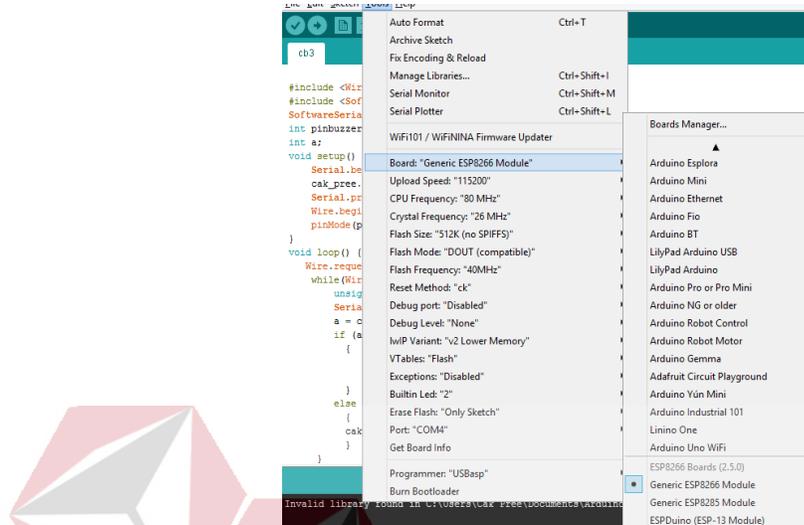
Gambar 3. 10 Flowchart ESP8266

1. Sebelum ESP8266 digunakan di *flash* terlebih dahulu dengan cara meng-*upload source code* ESP8266. Berikut ini adalah skematik rangkaian untuk ESP8266 bisa dilihat di Gambar 3.11 di bawah ini:



Gambar 3. 11 Skematik ESP8266 dan USB to TTL

2. Mengganti *Board* dengan Generic ESP8266 module, seperti Gambar 3.12 di bawah ini:



Gambar 3. 12 Mengganti *Board* Generic ESP8266 Module

3. Meng-*upload* program ESP8266 dengan USB to TTL yang sudah dirangkai, ditunggu hingga proses 100% dengan menekan tombol *reset* seperti Gambar 3.13 di bawah ini:



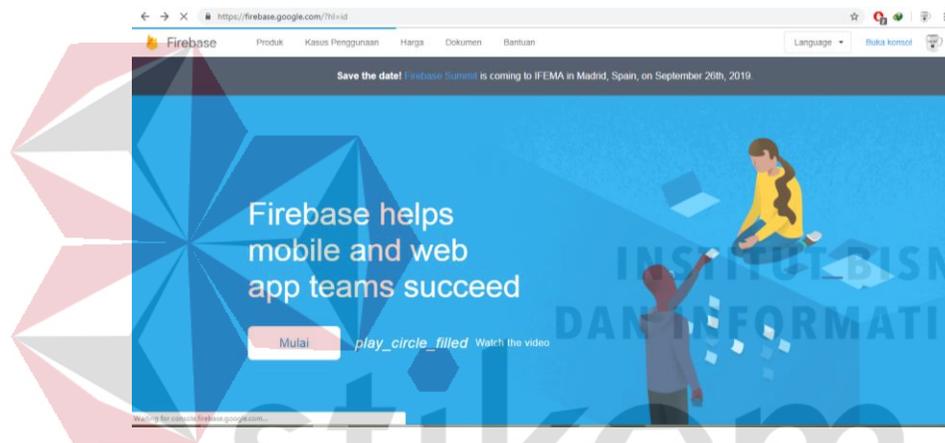
Gambar 3. 13 Tampilan *Flash* ESP8266

4. Melihat ESP8266 berhasil *connect* atau tidak bisa dilihat di Serial Monitor.

### C. Mengatur Pada Firebase

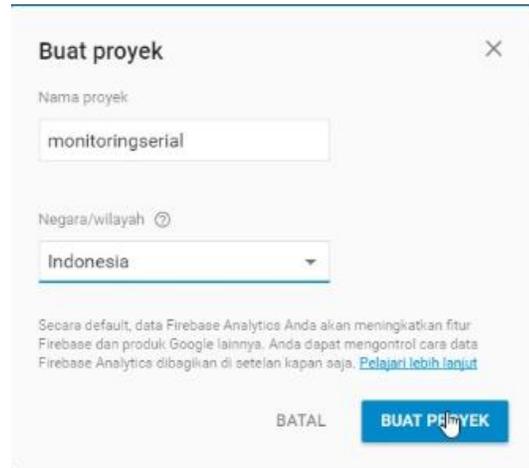
Dalam proses monitoring data sensor *Heart Rate* pada Firebase bisa dilakukan melalui langkah-langkah sebagai berikut:

1. Membuat *console* pada *web* resmi Firebase yaitu : <https://firebase.google.com>. Kemudian mengklik mulai. Seperti Gambar 3.14 di bawah ini:



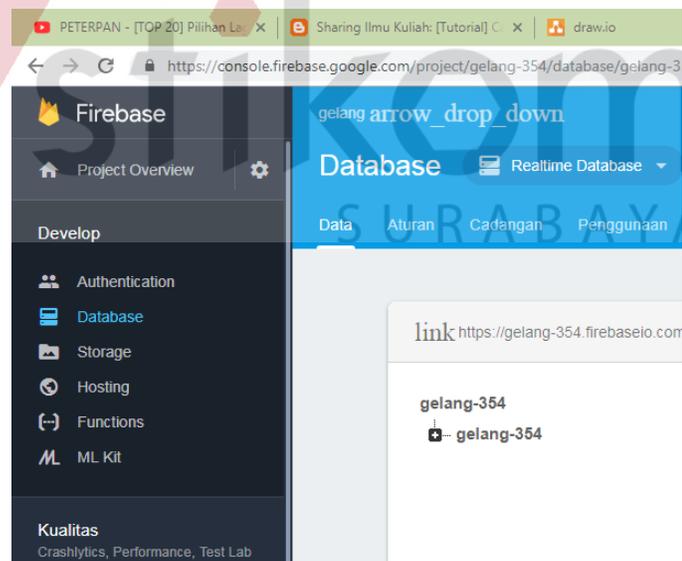
Gambar 3. 14 Tampilan Awal Firebase

2. Setelah itu akan ada pilihan dengan membuat proyek baru, mengisi nama proyek dan negara. Seperti Gambar 3.15 di bawah ini:



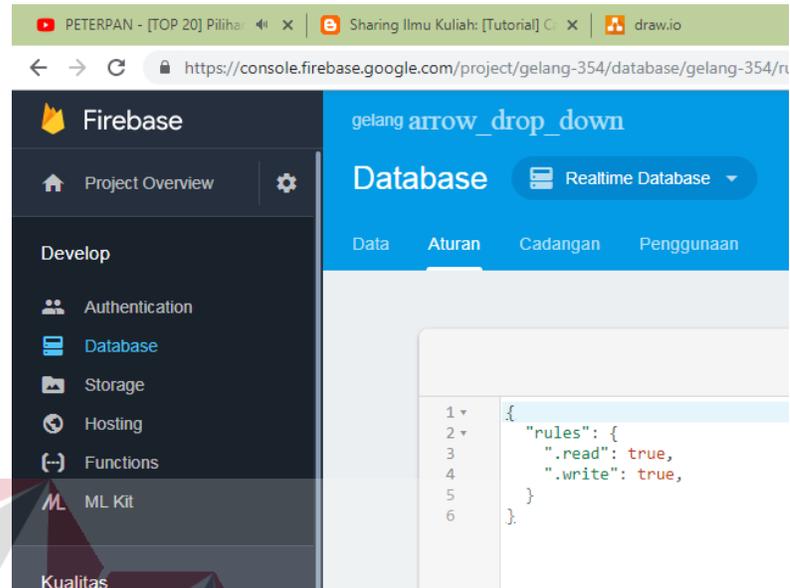
Gambar 3. 15 Tampilan Membuat *New* Proyek

3. Mengisikan data sesuai yang dibutuhkan, sebagai contoh penulis mengisikan data awal dengan gelang-354. Seperti Gambar 3.16 di bawah ini:



Gambar 3. 16 Tampilan Proyek *Database*

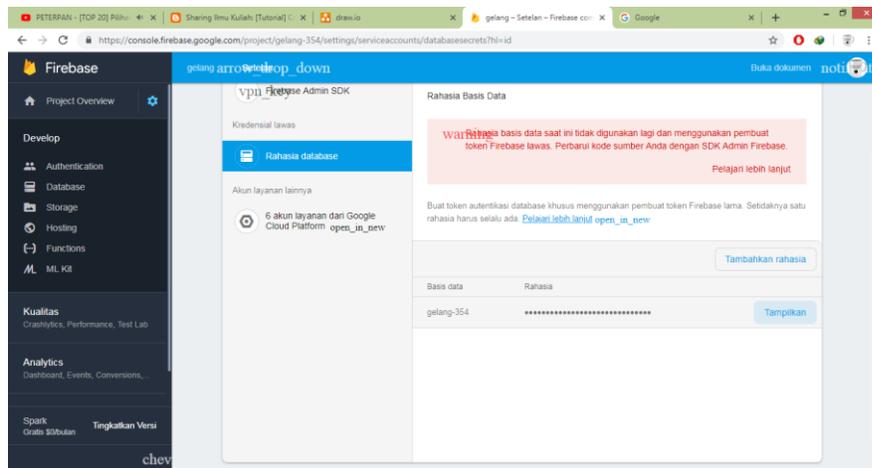
4. Mengganti aturan menjadi *true*, seperti Gambar 3.17 di bawah ini:



Gambar 3. 17 Tampilan Aturan Firebase

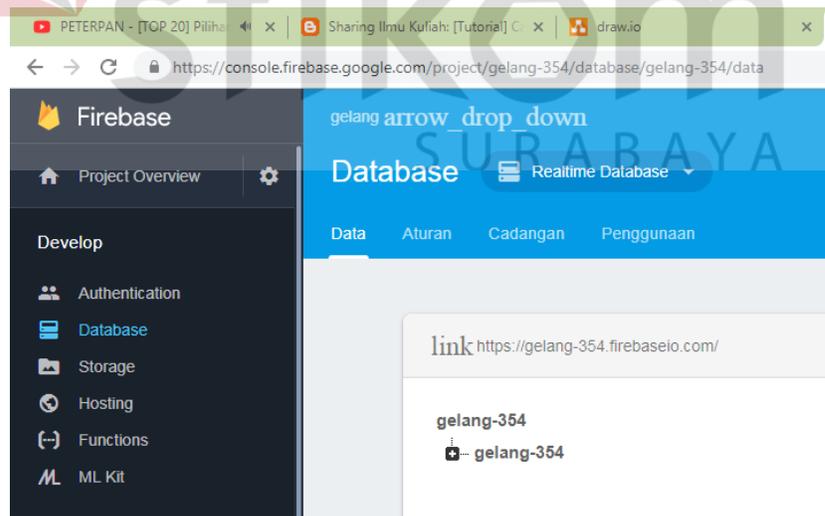
5. Memasukkan rahasia basis data ke Arduino IDE dengan cara *setting* – akun layanan – rahasia *database*, untuk mengkoneksikan Arduino dengan Firebase.

Seperti pada Gambar 3.18 di bawah ini:



Gambar 3. 18 Tampilan Akun Layanan Firebase

6. Memasukkan link pada Firebase ke Arduino IDE dengan cara *Database – Data – Link*, yaitu link *gelang-354.Firebaseio.com* Seperti pada Gambar 3.19 di bawah ini:



Gambar 3. 19 Tampilan *Link Database* Firebase

#### D. *Delay*

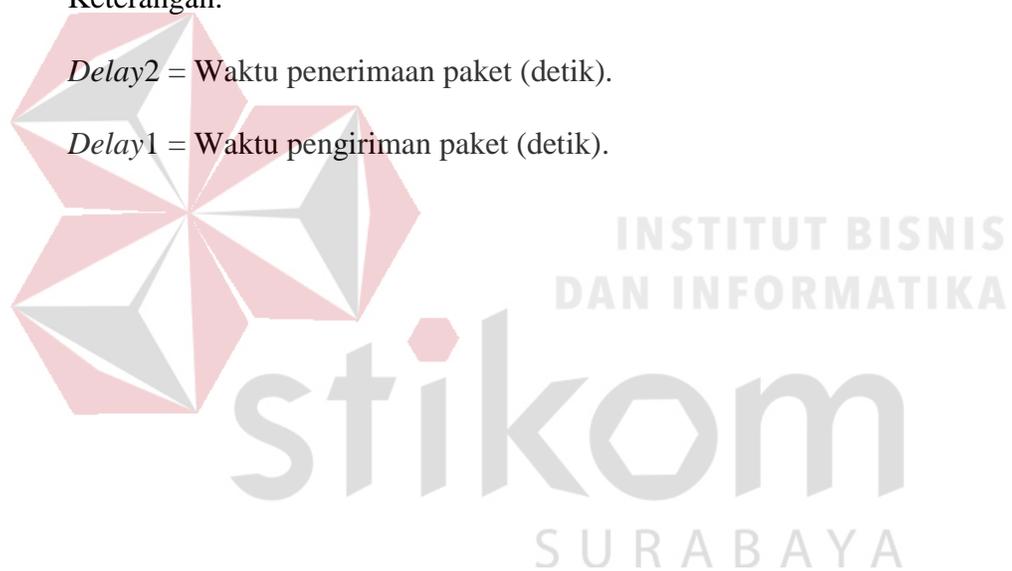
*Delay* adalah waktu tunda dalam proses transmisi data dari pengirim dan penerima, satuan *delay* adalah (detik). Contohnya paket data dari terminal 1 ke terminal 2, dalam perjalanan data tersebut mengalami propagasi sehingga membutuhkan waktu untuk sampai ke tujuan yaitu terminal 2. *Delay* memiliki rumus sebagai berikut:

$$Delay = Delay2 - Delay1$$

Keterangan:

*Delay2* = Waktu penerimaan paket (detik).

*Delay1* = Waktu pengiriman paket (detik).



## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pada bab ini, penulis akan membahas hasil pengujian dari alat yang sudah dirancang. Tahapan pengujian yang akan dilakukan meliputi pengujian tingkat keakurasian alat yang dibuat dengan *tensimeter* denyut jantung, pengujian penyimpanan data dari Firebase dan pengujian Satpam tertidur dengan yang beraktivitas.

#### **4.1 Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung**

##### **4.1.1 Tujuan Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung**

Pengujian ini dilakukan agar *Heart Rate* sensor yang dibuat bisa berjalan dengan baik dan tingkat keakurasian antara HR dengan *tensimeter* denyut jantung *range error*-nya tidak terlalu besar.

##### **4.1.2 Peralatan Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung**

Peralatan yang dibutuhkan untuk pengujian *heart rate* dengan *tensimeter* denyut jantung adalah sebagai berikut:

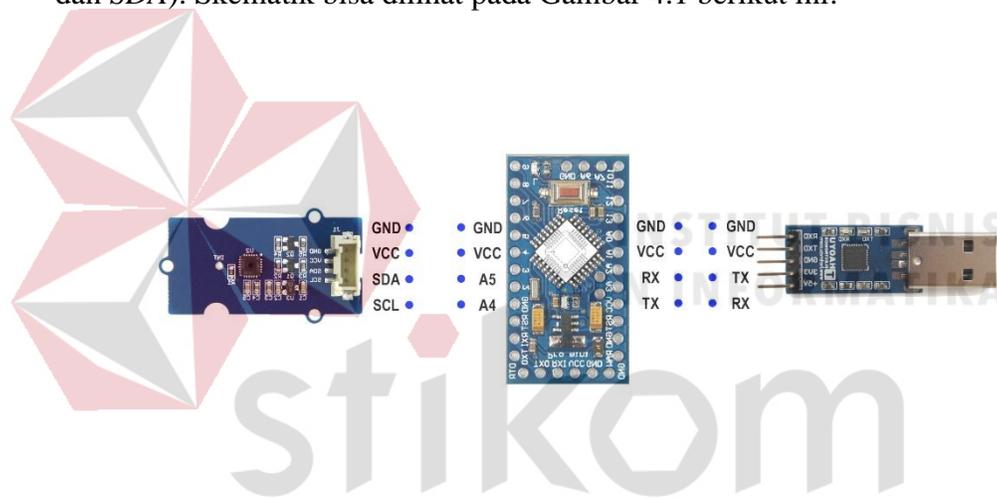
1. Sensor *Heart Rate*.
2. Arduino Pro Mini.
3. USB to TTL.
4. Arduino IDE.
5. *Tensimeter* deyak jantung.

## 6. Sampel Satpam.

### 4.1.3 Prosedur Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung

Adapun prosedur pengujian *heart rate* dengan *tensimeter* denyut jantung adalah sebagai berikut:

1. Menyambungkan Arduino Pro Mini dengan Arduino IDE menggunakan USB to TTL, lalu *Heart Rate* sensor disambungkan ke *pin* I2C Arduino Pro Mini (*SCL* dan *SDA*). Skematik bisa dilihat pada Gambar 4.1 berikut ini:



Gambar 4. 1 Skematik Arduino Pro Mini dengan Sensor *Heart Rate*

2. Menjalankan program Arduino Pro Mini berikut ini:

```
#include <Wire.h>
#include <SoftwareSerial.h>
///SoftwareSerial cak_pree(11,12);
int pinBuzzer = 13;
String hasil;
void setup() {
    Serial.begin(9600);
    Serial.println("Heart Rate sensor:");
    Wire.begin();
    pinMode(pinBuzzer, OUTPUT);
}
void loop() {
```

```

Wire.requestFrom(0xA0 >> 1, 1); //request 1 bytes from slave device
//while(Wire.available()) { // slave may send less than requested
unsigned char c = Wire.read(); // receive Heart Rate value (a byte)

    Serial.println(c,DEC);      // print Heart Rate value
    Serial.println("!");
    int a;
    a =c;

    if (a<=80)
    {
        digitalWrite (pinBuzzer, HIGH);
    }
    else
    {
        digitalWrite (pinBuzzer, LOW);
    }
    delay(500);

```

3. *Mem-verify* terlebih dahulu sebelum meng-*upload*, agar tidak ada *error*.
4. Membandingkan data dari sensor *Heart Rate* dengan *tensimeter* detak jantung, *tensimeter* detak jantung yang digunakan, bisa dilihat pada Gambar 4.2 berikut ini:



Gambar 4. 2 *Tensimeter* Detak Jantung

#### 4.1.4 Hasil Pengujian *Heart Rate* dengan *Tensimeter* Denyut Jantung

Pada Tabel 4.1 adalah hasil dari pengujian sensor *Heart Rate* dengan Arduino Pro Mini dan *Tensimeter* denyut jantung.

Tabel 4. 1 Tabel Pengujian *Prototype* Dengan *Tensimeter*

Sampel	Percobaan	Nilai HR Beraktivitas (Bpm)	Nilai HR Beristirahat (Bpm)	HR (Bpm)		Error
				<i>Finger clip</i>	<i>Tensimeter</i>	
<b>Revido</b>	1	109	78	81	81	0
	2	90	75	81	81	0
	3	93	68	81	81	0
	4	89	70	81	81	0
	5	89	72	82	81	1
	6	86	70	82	81	1
<b>Rata-rata error</b>						0.33
<b>Afrizal</b>	1	93	79	70	70	0
	2	92	78	70	70	0
	3	94	73	70	70	0
	4	95	73	70	70	0
	5	95	72	71	70	1
	6	90	70	72	70	2
<b>Rata-rata error</b>						0.33
<b>Agung</b>	1	97	69	77	77	0
	2	87	70	77	77	0
	3	83	68	77	77	0
	4	82	72	79	77	2
	5	81	70	77	77	0
	6	81	70	79	77	2
<b>Rata-rata error</b>						0.33

<b>Fahmi</b>	1	91	79	83	83	0
	2	90	79	83	83	0
	3	83	77	81	83	2
	4	80	71	82	83	1
	5	83	71	81	83	2
	6	80	69	83	83	0
<b>Rata-rata error</b>						1
<b>Sukus</b>	1	109	77	80	80	0
	2	96	75	80	80	0
	3	94	80	81	80	1
	4	91	70	81	80	1
	5	91	68	80	80	0
	6	90	75	80	80	0
<b>Rata-rata error</b>						0.33
<b>Affandi</b>	1	102	78	85	85	0
	2	97	75	85	85	0
	3	94	75	85	85	0
	4	90	72	85	85	0
	5	91	68	84	85	1
	6	78	70	84	85	1
<b>Rata-rata error</b>						0.33

Berdasarkan teori yang diperoleh normalnya orang dewasa beraktivitas berkisar 80-120 Bpm dan orang yang beristirahat < 75 Bpm. (Sandi, 2013) Dari hasil tabel 4.1 pengujian sampel diatas dapat disimpulkan bahwa detak jantung orang yang beraktivitas dengan orang yang beristirahat berbeda. Rata-rata orang yang beraktivitas detak jantungnya > 80, sedangkan orang yang beristirahat/tidur detak jantungnya lebih

rendah yaitu  $< 80$ . Pengujian ini mengambil sampel dengan kriteria umur 20 – 30 tahun, dan tidak memiliki riwayat penyakit dalam yang dapat mempengaruhi kinerja detak jantung. Hasil pengujian akurasi sensor *node* dengan *tensimeter* detak jantung pada sampel diatas dapat disimpulkan bahwa rata-rata *error* yaitu 0.386.

## **4.2 Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate***

### **4.2.1 Tujuan Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate***

Pengujian ini dilakukan agar integrasi antara Arduino Pro Mini dengan sensor *Heart Rate* bisa berjalan dengan baik.

### **4.2.2 Peralatan Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate***

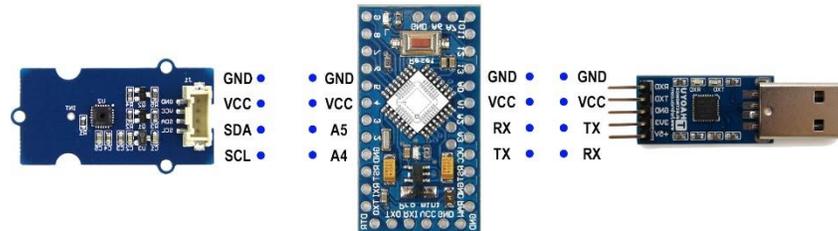
Peralatan yang dibutuhkan untuk pengujian arduino pro mini dengan sensor *heart rate* adalah sebagai berikut:

1. Aduino Pro Mini.
2. USB to TTL.
3. Arduino IDE.
4. Program Arduino.
5. Sensor *Heart Rate*.

### **4.2.3 Prosedur Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate***

Adapun prosedur pengujian arduino pro mini dengan sensor *heart rate* adalah sebagai berikut:

1. Menyambungkan Arduino Pro Mini dengan USB to TTL. Bisa dilihat pada Gambar 4.3 berikut ini:



Gambar 4. 3 Skematik *Prototype* Sensor Node

2. Menjalankan aplikasi Arduino IDE dan memasukkan program untuk mencoba komunikasi serial antara Arduino Pro Mini dengan sensor *Heart Rate*. Berikut ini adalah penggalan kodingannya:

```
#include <Wire.h>
#include <SoftwareSerial.h>
//SoftwareSerial cak_pree(11,12);
int pinBuzzer = 13; String hasil;
void loop() {

Wire.requestFrom(0xA0 >> 1, 1);
//request 1 bytes from slave device

//while(Wire.available()) {
//slave may send less than requested
unsigned char c = Wire.read();

//receive Heart Rate value (a byte)
Serial.println(c,DEC); // print Heart Rate value
Serial.println("!");
int a;
a =c;
```

```

if (a<=80)
{
  digitalWrite (pinBuzzer, HIGH);
}
else
{
  digitalWrite (pinBuzzer, LOW);
}
delay(500);
}

```

3. *Mem-verify* terlebih dahulu sebelum meng-*upload*, agar tidak ada *error*.
4. Membuka serial monitor pada pojok kanan atas untuk melihat hasilnya.

#### 4.2.4 Hasil Pengujian Arduino Pro Mini Dengan Sensor *Heart Rate*

Hasil pengujian Arduino Pro Mini dan sensor *Heart Rate* menampilkan komunikasi serial bisa dilihat pada Gambar 4.4 berikut ini:



Gambar 4. 4 Tampilan Serial Monitor

Dari pengujian diatas dapat disimpulkan bahwa Arduino Pro Mini dapat membaca data dari sensor *Heart Rate* dengan baik. Sensor *Heart Rate* mengirimkan data melalui *pin* I2C pada Arduino Pro Mini dan menampilkannya ke serial monitor. Data yang ditampilkan sesuai dengan yang diharapkan.

### 4.3 Pengujian ESP8266

#### 4.3.1 Tujuan Pengujian ESP8266

Pengujian ESP8266 bertujuan untuk melakukan komunikasi data serial, serta program dapat berjalan dengan baik.

#### 4.3.2 Peralatan Pengujian ESP8266

Peralatan yang dibutuhkan untuk pengujian ESP8266 adalah sebagai berikut:

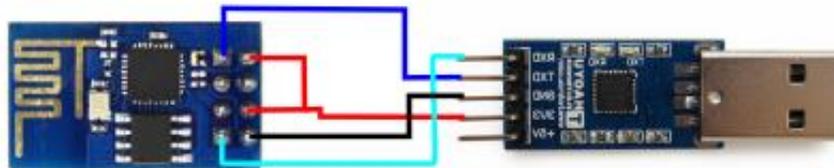
1. ESP8266.
2. USB to TTL.
3. Arduino IDE.

#### 4.3.3 Prosedur Pengujian ESP8266

Adapun prosedur pengujian ESP8266 adalah sebagai berikut:

1. Menyambungkan ESP8266 dengan USB to TTL. Ketentuannya bisa dilihat pada

Gambar 4.5 berikut ini:



Gambar 4. 5 Skematik ESP8266 dengan USB to TTL

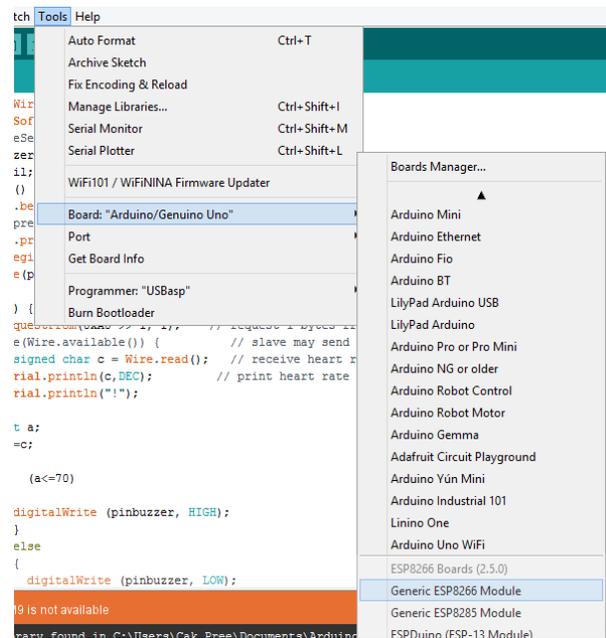
2. Menjalankan aplikasi arduino IDE dan jalankan progam ESP8266. Berikut ini adalah penggalan kodingannya:

```
#define FIREBASE_HOST "gelang-354.Firebaseio.com"
#define FIREBASE_AUTH "CG8eySGYjT92B58WlkyJFQS9KoSpdAlAJdnFXQiY"
#define WIFI_SSID "iPhone Freean"
#define WIFI_PASSWORD "cakpreee"
String dataSerial = "";
String kirim, HR;
int intHR;

void setup() {
  Serial.begin(9600);    // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```

3. Mengganti *board* dengan modul ESP8266 GENERIC, seperti pada Gambar 4.6 berikut ini:

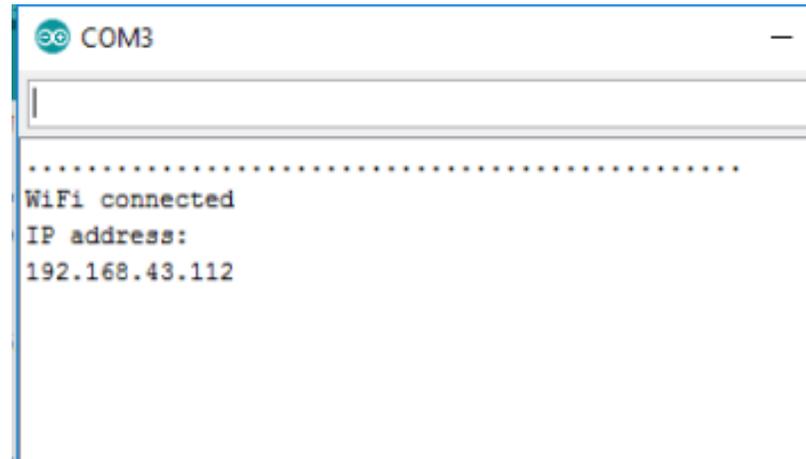


Gambar 4. 6 Tampilan *Board* Arduino IDE

4. *Mem-verify* terlebih dahulu sebelum meng-*upload* program agar tidak ada *error*, setelah itu meng-*upload* program.
5. Membuat *hostpot wifi* menggunakan *smartphone* dengan *SSID* iPhone Freean dan *PASS* cakpreee.

#### 4.3.4 Hasil Pengujian ESP8266

Hasil pengujian ESP8266 terkoneksi pada *hostpot* yang sudah dibuat bisa dilihat pada Gambar 4.7 berikut:



Gambar 4. 7 Tampilan *Connected* ESP8266

#### 4.4 Pengujian Komunikasi Serial

##### 4.4.1 Tujuan Pengujian Komunikasi Serial

Tujuan pengujian komunikasi serial dilakukan dengan Arduino Pro Mini dan ESP8266. Modul ini akan melakukan komunikasi serial sehingga nilai dari sensor *Heart Rate* yang disambung dengan Arduino Pro Mini bisa mengirim ke ESP8266.

##### 4.4.2 Peralatan Pengujian Komunikasi Serial

Peralatan yang dibutuhkan untuk pengujian komunikasi serial adalah sebagai berikut:

1. Arduino Pro Mini.
2. ESP8266.
3. Sensor *Heart Rate*.
4. Arduino IDE.

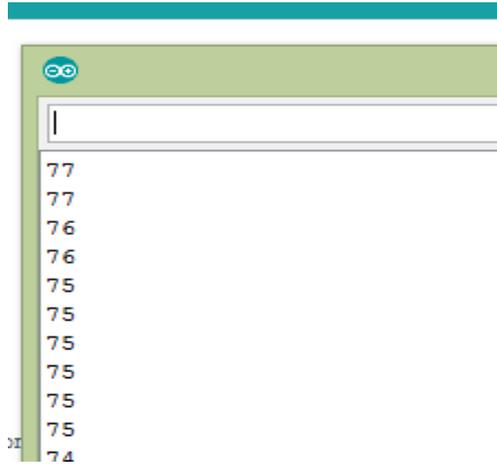
#### 4.4.3 Prosedur Pengujian Komunikasi Serial

Adapun prosedur pengujian komunikasi serial adalah sebagai berikut:

1. Menjalankan aplikasi Arduino IDE.
2. Menghubungkan *pin* RX (*Receiver*) Arduino Pro Mini dengan *pin* TX (*Transmitter*) ESP8266.
3. Menghubungkan *pin* TX (*Transmitter*) Arduino Pro Mini dengan *pin* RX (*Receiver*) ESP8266.
4. *Mem-verify* terlebih dahulu sebelum meng-*upload* program agar tidak ada *error*, setelah itu meng-*upload* program.
5. Menghubungkan sensor *Heart Rate* dengan *pin* I2C (SCL dan SDA) pada Arduino Pro Mini.
6. Mengamati hasil pada serial monitor.

#### 4.4.4 Hasil Pengujian Komunikasi Serial

Hasil dari pengujian komunikasi serial dapat dilakukan dengan cara melihat pada serial monitor pada Arduino IDE. Ketika Arduino menerima data dari sensor *Heart Rate*, Arduino mengirimkan data tersebut melalui *pin* RX (*Receiver*) yang dihubungkan dengan *pin* TX (*Transmitter*) pada ESP8266. Setelah membaca data dari Arduino Pro Mini maka data tersebut dikirimkan ke *cloud* Firebase. ESP8266 akan mengirimkan timbal balik pada Arduino Pro Mini dengan *pin* TX (*Transmitter*) dan diterima oleh Arduino Pro Mini dengan *pin* RX (*Receiver*) bahwa data sudah diterima dengan baik. Hasil serial monitor bisa dilihat pada Gambar 4.8 berikut ini:



Gambar 4. 8 Tampilan Hasil Pengujian *Heart Rate*

## 4.5 Pengujian Firebase

### 4.5.1 Tujuan Pengujian Firebase

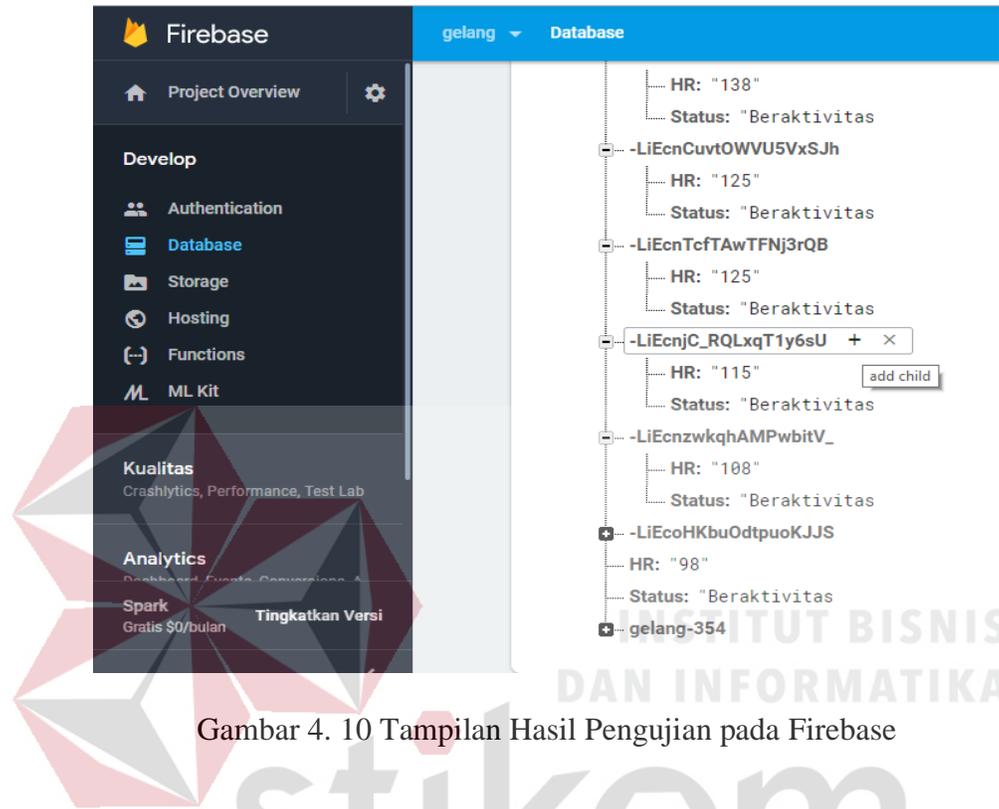
Pengujian Firebase dilakukan untuk melihat data yang dikirimkan dari sensor *node* menuju Firebase sudah sesuai dengan yang diharapkan atau tidak.

### 4.5.2 Peralatan Pengujian Firebase

Peralatan yang dibutuhkan untuk pengujian Firebase adalah sebagai berikut:

1. Arduino Pro Mini.
2. Sensor *Heart Rate*.
3. ESP8266.
4. Laptop.





Gambar 4. 10 Tampilan Hasil Pengujian pada Firebase

Dari gambar diatas dapat disimpulkan bahwa pengiriman data dari sensor *node* menuju *cloud* Firebase berjalan dengan baik. Firebase menampilkan nilai detak jantung dan keterangan apakah Satpam tersebut tertidur atau tidak.

## 4.6 Pengujian *Delay*

### 4.6.1 Tujuan Pengujian *Delay*

Pengiriman data dari sensor *node* ke Firebase dilakukan menggunakan *wireshark*. Dengan memperhitungkan jaringan *wifi* yang digunakan, tujuannya agar mengetahui berapa *delay* pengiriman data dari sensor *node* ke Firebase.

#### 4.6.2 Peralatan Pengujian *Delay*

Peralatan yang dibutuhkan untuk pengujian *delay* adalah sebagai berikut:

1. Sensor *Heart Rate*.
2. Arduino Pro Mini.
3. ESP8266.
4. Jaringan *wifi*.
5. Aplikasi *wireshark*.
6. Laptop.

#### 4.6.3 Prosedur Pengujian *Delay*

Adapun prosedur pengujian *delay* adalah sebagai berikut:

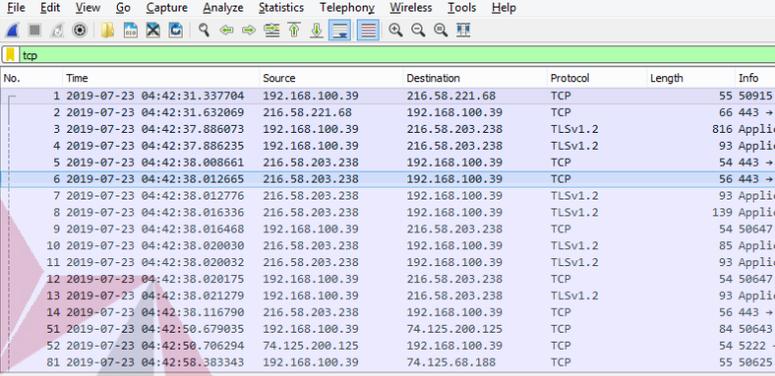
1. Menjalankan aplikasi *wireshark*, dan memilih jaringan *wifi* pada *capture*. Seperti pada Gambar 4.11 berikut ini:



Gambar 4. 11 Tampilan Awal *Wireshark*

2. Membuka *website* Firebase dan menyalakan sensor node untuk mendapatkan data detak jantung.

- Menunggu 1-2 menit agar data yang diterima oleh Firebase sesuai dengan kondisi detak jantung sebenarnya.
- Melihat *delay* pada *wireshark*, dengan cara mengklik *stop* pada *wireshark* dan menuliskan *tcp/http* pada *filter*. Seperti pada Gambar 4.12 berikut ini:



No.	Time	Source	Destination	Protocol	Length	Info
1	2019-07-23 04:42:31.337704	192.168.100.39	216.58.221.68	TCP	55	50915
2	2019-07-23 04:42:31.632069	216.58.221.68	192.168.100.39	TCP	66	443 →
3	2019-07-23 04:42:37.886073	192.168.100.39	216.58.203.238	TLSv1.2	816	Applic
4	2019-07-23 04:42:37.886235	192.168.100.39	216.58.203.238	TLSv1.2	93	Applic
5	2019-07-23 04:42:38.008661	216.58.203.238	192.168.100.39	TCP	54	443 →
6	2019-07-23 04:42:38.012665	216.58.203.238	192.168.100.39	TCP	56	443 →
7	2019-07-23 04:42:38.012776	216.58.203.238	192.168.100.39	TLSv1.2	93	Applic
8	2019-07-23 04:42:38.016336	216.58.203.238	192.168.100.39	TLSv1.2	139	Applic
9	2019-07-23 04:42:38.016468	192.168.100.39	216.58.203.238	TCP	54	50647
10	2019-07-23 04:42:38.020030	216.58.203.238	192.168.100.39	TLSv1.2	85	Applic
11	2019-07-23 04:42:38.020032	216.58.203.238	192.168.100.39	TLSv1.2	93	Applic
12	2019-07-23 04:42:38.020175	192.168.100.39	216.58.203.238	TCP	54	50647
13	2019-07-23 04:42:38.021279	192.168.100.39	216.58.203.238	TLSv1.2	93	Applic
14	2019-07-23 04:42:38.116790	216.58.203.238	192.168.100.39	TCP	56	443 →
51	2019-07-23 04:42:50.679035	192.168.100.39	74.125.200.125	TCP	84	50643
52	2019-07-23 04:42:50.706294	74.125.200.125	192.168.100.39	TCP	54	5222 -
81	2019-07-23 04:42:58.383343	192.168.100.39	74.125.68.188	TCP	55	50625

Frame 1: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0  
 Ethernet II, Src: HonHaiPr\_ad:dc:05 (f8:2f:a8:ad:dc:05), Dst: HuaweiTe\_8c:7f:4a (78:58:60:8c:7f:4a)  
 Internet Protocol Version 4, Src: 192.168.100.39, Dst: 216.58.221.68  
 Transmission Control Protocol, Src Port: 50915, Dst Port: 443, Seq: 1, Ack: 1, Len: 1

Gambar 4. 12 Tampilan *Capture* Pada Jaringan *TCP*

- Menyimpan *delay* yang telah *capture*, dengan cara *file – Export Packet Dissections – As CSV*.
- Membuka *file* yang telah disimpan dengan *excel* dan menghitung *delay*-nya. Seperti pada Gambar 4.13 berikut ini:

No.	Time	Source	Destinasi	Protocol	Length	Info												
1	51:46.3	172.20.10	172.20.10	DNS	87	Standard query 0x94ea A sqm.telemetry.microsoft.com												00:00.3
2	51:46.5	172.20.10	172.20.10	DNS	154	Standard query response 0x94ea A sqm.telemetry.microsoft.com CNAME sqm.telemetry.mi												00:00.0
3	51:46.5	172.20.10	65.55.252	TCP	66	49272 > 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1											00:00.0	
4	51:46.6	65.55.252	172.20.10	TCP	66	443 > 49272 [SYN, ACK] Seq=0 Ack=1 Win=22400 Len=0 MSS=1400 WS=4 SACK_PERM=1											00:00.0	
5	51:46.6	172.20.10	65.55.252	TCP	54	49272 > 443 [ACK] Seq=1 Ack=1 Win=65792 Len=0												00:00.0
6	51:46.6	172.20.10	65.55.252	TLSv1.2	242	Client Hello												00:00.0
7	51:46.6	65.55.252	172.20.10	TCP	54	443 > 49272 [ACK] Seq=1 Ack=189 Win=22588 Len=0												00:02.0
8	51:48.6	172.20.10	35.201.97	TLSv1.2	82	Application Data												00:00.1
9	51:48.6	172.20.10	35.201.97	TCP	55	[TCP Retransmission] 49244 > 443 [ACK] Seq=0 Ack=1 Win=256 Len=1												00:00.0
10	51:48.6	35.201.97	172.20.10	TCP	54	443 > 49244 [ACK] Seq=1 Ack=29 Win=6114 Len=0												00:00.0
11	51:48.7	35.201.97	172.20.10	TCP	66	[TCP Dup ACK 10#1] 443 > 49244 [ACK] Seq=1 Ack=29 Win=6114 Len=0 SLE=0 SRE=1												00:02.1
12	51:50.8	fe80::19d5:fd02::12		DHCPv6	151	Solicit XID: 0xc754a CID: 000100012406e60fc45444790d6												00:01.4
13	51:52.2	35.201.97	172.20.10	TLSv1.2	194	Application Data												00:00.1
14	51:52.2	172.20.10	35.201.97	TCP	54	49244 > 443 [ACK] Seq=29 Ack=81 Win=255 Len=0												00:00.3
15	51:52.6	35.201.97	172.20.10	TLSv1.2	145	Application Data												00:00.1
16	51:52.6	172.20.10	35.201.97	TCP	54	49244 > 443 [ACK] Seq=29 Ack=172 Win=255 Len=0												00:00.1
17	51:52.8	172.20.10	172.217.11	TCP	84	49198 > 5222 [PSH, ACK] Seq=1 Ack=1 Win=254 Len=30 [TCP segment of a reassembled PDU												00:00.1
18	51:52.8	172.217.11	172.20.10	TCP	54	5222 > 49198 [ACK] Seq=1 Ack=31 Win=6520 Len=0												00:00.1
19	51:52.9	35.201.97	172.20.10	TLSv1.2	179	Application Data												00:00.0
20	51:52.9	172.20.10	35.201.97	TCP	54	49244 > 443 [ACK] Seq=29 Ack=297 Win=254 Len=0												00:00.6
21	51:53.5	172.20.10	172.20.10	DNS	75	Standard query 0x5504 A play.google.com												00:00.1
22	51:53.5	172.20.10	172.20.10	DNS	171	Standard query response 0x5504 A play.google.com A 172.217.194.100 A 172.217.194.113 A 1												00:00.0
23	51:53.7	172.20.10	172.217.11	UDP	1392	62901 > 443 Len=1350												00:00.1
24	51:53.7	172.217.11	172.20.10	UDP	1392	443 > 62901 Len=1350												00:00.0
25	51:53.7	172.20.10	172.217.11	UDP	1392	62901 > 443 Len=1350												00:00.0
26	51:53.8	172.217.11	172.20.10	UDP	78	443 > 62901 Len=36												00:00.0
27																		00:00.0

Gambar 4. 13 Delay Yang Tersimpan Pada Excel

#### 4.6.4 Hasil Pengujian Delay

Hasil *delay* yang didapat dengan perhitungan dengan menggunakan *wireshark* bisa dilihat pada Tabel 4.2 berikut ini:

Tabel 4. 2 Tabel Pengujian Delay

Sampel	Percobaan	Spek Jaringan	Delay (detik)
Revido	1	Wifi (10 mbps)	36.2
	2	Wifi (10 mbps)	57.4
	3	Wifi (10 mbps)	68.1
	4	Wifi (10 mbps)	33.8
	5	Wifi (10 mbps)	55.2
	6	Wifi (10 mbps)	50.0
<b>Rata-rata delay (detik)</b>			50.1166
Afrizal	1	Wifi (10 mbps)	109.1
	2	Wifi (10 mbps)	79.6
	3	Wifi (10 mbps)	57.0
	4	Wifi (10 mbps)	63.5
	5	Wifi (10 mbps)	119.9
	6	Wifi (10 mbps)	165.0
<b>Rata-rata delay (detik)</b>			99.0166
Agung	1	Wifi (10 mbps)	64.4
	2	Wifi (10 mbps)	87.1
	3	Wifi (10 mbps)	79.2

	4	Wifi (10 mbps)	129.4
	5	Wifi (10 mbps)	80.7
	6	Wifi (10 mbps)	55.6
<b>Rata-rata delay (detik)</b>			82.7333
<b>Fahmi</b>	1	Wifi (10 mbps)	62.1
	2	Wifi (10 mbps)	51.6
	3	Wifi (10 mbps)	42.5
	4	Wifi (10 mbps)	86.9
	5	Wifi (10 mbps)	65.7
	6	Wifi (10 mbps)	34.6
<b>Rata-rata delay (detik)</b>			57.2333
<b>Sukus</b>	1	Wifi (10 mbps)	51.1
	2	Wifi (10 mbps)	66.1
	3	Wifi (10 mbps)	57.5
	4	Wifi (10 mbps)	67.8
	5	Wifi (10 mbps)	52.7
	6	Wifi (10 mbps)	55.8
<b>Rata-rata delay (detik)</b>			58.5
<b>Affandi</b>	1	Wifi (10 mbps)	65.6
	2	Wifi (10 mbps)	51.8
	3	Wifi (10 mbps)	48.0
	4	Wifi (10 mbps)	42.7
	5	Wifi (10 mbps)	42.2
	6	Wifi (10 mbps)	38.5
<b>Rata-rata delay (detik)</b>			48.1333

Hasil pengujian *delay* pada Tabel 4.2 dapat disimpulkan bahwa pengiriman data dari sensor *node* ke Firebase menggunakan jaringan *wifi* memiliki *delay* dengan rata-rata *delay* 65.9555 detik.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Hasil dari pengujian rancang bangun alat pendeteksi tidur pada *security guard* berbasis IoT (*Internet of Things*) pada Tugas Akhir ini terdapat beberapa kesimpulan, diantaranya:

1. Disimpulkan bahwa orang yang beristirahat/tidur memiliki detak jantung lebih rendah dari pada orang yang beraktivitas. Orang yang beristirahat/tidur memiliki rata-rata detak jantung  $< 80$  Bpm, sedangkan orang yang beraktivitas memiliki rata-rata detak jantung  $> 80$  Bpm. Hasil pengujian akurasi sensor *node* dengan *tensimeter* detak jantung memiliki rata-rata *error* yaitu 0.386.
2. Data dari sensor *Heart Rate* yang dikirimkan ke Firebase dapat dimonitoring dengan baik. Firebase menampilkan nilai detak jantung dan keterangan apakah Satpam tertidur atau tidak. Pengiriman data menggunakan ESP8266 ke *cloud* Firebase memiliki *delay* rata-rata 65.9555 detik.
3. Data sensor *Heart Rate* dikirimkan ke Arduino Pro Mini melalui *pin* I2C. dengan pembacaan 1 detik sekali.

#### 5.2 Saran

Adapun saran yang diberikan oleh penulis adalah sebagai berikut:

1. Untuk membuat sistem monitoring Satpam dibutuhkan catu daya yang cukup. Hasil perhitungan catu daya pada rancang bangun ini  $\pm 3.5$  jam sedangkan pada umumnya Satpam bertugas dalam shift berkisar 8 jam.
2. Demi kenyamanan Satpam dalam bertugas disarankan desain rancang bangun alat lebih diperkecil, agar lebih efisien dan ergonomis.



## DAFTAR PUSTAKA

- Abduracman. (2018). *Rancang bangun tongkat tunanetra dengan sensor ultrasonik berbasis mikrokontroler atmega8*. 2018.
- Bashofi, M. E., & Abidin, R. Z. (2018). *IMPLEMENTASI FIREBASE PADA SISTEM KENDALI LAMPU JARAK JAUH BERBASIS ANDROID*. Univesitas Yudharta Pasuruan.
- Darwinsa, E. (2011). *RATEMONITOR SEPULUH DENYUT PER SATUAN WAKTU DENGAN TRANSMISI RADIO FREKUENSI ( RF )*.
- Dita, D. (2009). *PENGARUH EKSTRAK BIJI PALA (Myristica fragrans Houtt) DOSIS 7,5 mg/ 25grBB TERHADAP WAKTU INDUKSI TIDUR DAN LAMA WAKTU TIDUR MENCIT BALB / C yang DIINDUKSI THIOPENTAL*. *Fakultas Kedokteran Universitas Diponegoro*, 1–31.
- Firdausi, N. A. (2018). *PROTOTIPE ALAT MONITORING DETAK JANTUNG PRTABEL MENGGUNAKAN ARDUINO PRO MINI DAN BLUETOOTH BERBASIS ANDROID*.
- Hidayah, A. (2012). *Tingkat Pengetahuan Pasien Diabetes Mellitus tentang Risiko Terjadinya Ulkus Kaki Diabetes di Poli Klinik Penyakit Dalam Rumah Sakit Umum Pusat Haji Adam Malik Medan*.
- Ids, M. M. A. (n.d.). *1 2 3, 4*. 214–222.
- Putra, N. D. (2018). *Wireless Smart Tag Device Sebagai Sistem Keamanan Rumah Sistem Keamanan Rumah*. *FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA*.
- Sandi, N. (2013). *HUBUNGAN ANTARA TINGGI BADAN, BERAT BADAN, INDEKS MASSA TUBUH, DAN UMUR TERHADAP FREKUENSI DENYUT NADI ISTIRAHAT SISWA SMKN-5 DENPASAR*. *1(1)*, 38–44.
- SeedTechnology. (2018). "GROVE-FINGER-CLIP- HATE RATE SENSOR." Retrieved from [http://wiki.seeedstudio.com/Grove-Finger-clip\\_Heart\\_Rate\\_Sensor/](http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/)
- SOFYAN, A. (2006). *PERANCANGAN ALAT MONITORING CEK POINT SATPAM*.
- Suoth, M., Bidjuni, H., Malara, R. T., Studi, P., Keperawatan, I., Kedokteran, F., ...

Manado, R. (2014). *PENDAHULUAN Pelayanan kesehatan yang diberikan puskesmas merupakan pelayanan yang menyeluruh yang meliputi pelayanan kuratif ( pengobatan ), rehabilitative kesehatan ). Pelayanan tersebut ditujukan kepada semua penduduk dengan tidak membedakan jenis kelim. 2.*

WWW.FARNELL.COM. (2016). BUZZER. Retrieved from  
<http://www.farnell.com/datasheets/2171929.pdf>

