



**EFISIENSI DAYA UNTUK PANTAUAN DATA *HEART RATE*  
MENGUNAKAN METODE *IDLE TIME-DEEP SLEEP***

**TUGAS AKHIR**

**Program Studi  
S1 Teknik Komputer**

**Oleh:**

**Yonanta Ferdinan Susanto**

**15.41020.0025**

**INSTITUT BISNIS  
DAN INFORMATIKA**

**stikom**  
SURABAYA

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

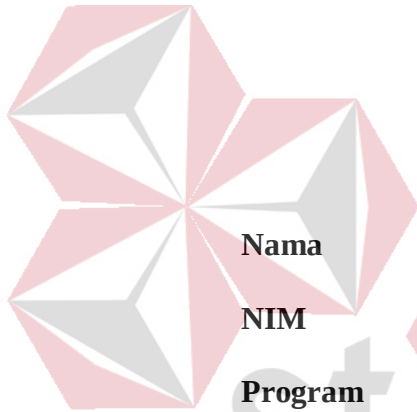
**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2019**

**EFISIENSI DAYA UNTUK PANTAUAN DATA *HEART RATE*  
MENGUNAKAN METODE *IDLE TIME-DEEP SLEEP***

**TUGAS AKHIR**

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



**Oleh :**

**Nama : Yonanta Ferdinan Susanto**

**NIM : 15.41020.0025**

**Program : S1 (Strata Satu)**

**Jurusan : Teknik Komputer**

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

**2019**

**TUGAS AKHIR**  
**EFISIENSI DAYA UNTUK PANTAUAN DATA *HEART RATE***  
**MENGUNAKAN METODE *IDLE TIME-DEEP SLEEP***

dipersiapkan dan disusun oleh

**Yonanta Ferdinan Susanto**

**NIM : 15.41020.0025**

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. Musayyanah, S.ST., M.T.**

**NIDN : 0730069102**

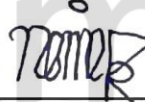
Penguji

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

**NIDN : 0716117302**



8/8/2019



Tugas Akhir ini telah diterima sebagai salah satu persyaratan



untuk memperoleh gelar Sarjana

FAKULTAS TEKNOLOGI  
DAN INFORMATIKA

**stikom**  
SURABAYA

**Dr. Jusak**

**NIDN : 0708017101**

**Dekan Fakultas Teknologi dan Informatika**

1/8/19

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

## PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Yonanta Ferdinan Susanto

NIM : 15.41020.0025

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya : **EFISIENSI DAYA UNTUK PANTAUAN DATA HEART RATE MENGGUNAKAN METODE IDLE TIME-DEEP SLEEP**

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 kesarjanaan yang telah diberikan kepada saya.

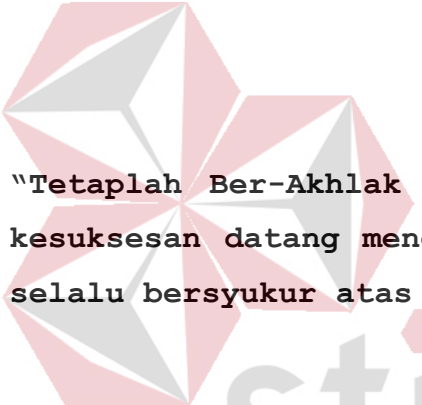
Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 22 Juli 2019

Yang menyatakan

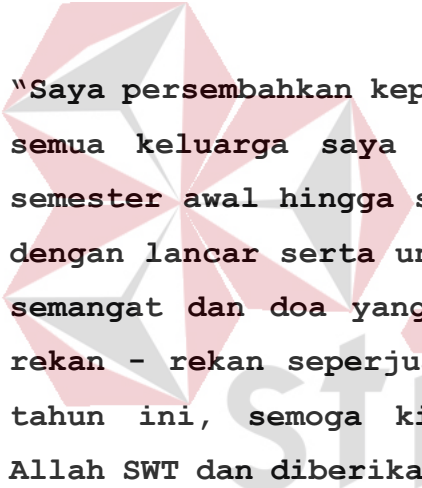


Yonanta Ferdinan S  
NIM : 15.41020.0025



"Tetaplah Ber-Akhlak dalam mengerjakan sesuatu hingga kesuksesan datang menghampiri, namun jangan lupa untuk selalu bersyukur atas semua yang telah didapatkan"

stikom  
SURABAYA



"Saya persembahkan kepada kedua orang tua saya, adik dan semua keluarga saya yang selalu mendukung saya dari semester awal hingga sekarang bisa menyelesaikan skripsi dengan lancar serta untuk seseorang yang selalu memberi semangat dan doa yang terbaik untuk saya, dan seluruh rekan - rekan seperjuangan dalam menimba ilmu selama 4 tahun ini, semoga kita senantiasa selalu dilindungi Allah SWT dan diberikan kesuksesan dunia maupun akhirat"

## ABSTRAK

Perkembangan teknologi saat ini sudah sangat berkembang pesat, terlebih pada teknologi yang bersifat IoT (*Internet Of Things*). Suatu sistem Monitoring hardware yang bersifat *wearable* dan *IoT* serta hemat daya diperlukan dan dapat diterapkan untuk memudahkan proses pantauan. Sistem dan *prototype* ini sebelumnya sudah dilakukan pada penelitian (Musayyanah,2018) yang berjudul Monitoring THR (Target Heart Rate) untuk optimalisasi latihan lari berbasis IoT (*Internet Of Things*). Namun dalam penelitian ini terdapat kekurangan yaitu pada bagian konsumsi daya, maka dari itu algoritma Efisiensi *Idle-Time Deep Sleep* akan diimplementasikan pada penggabungan hardware sensor Grove Finger Clip Heart Rate, RTC (*Real-Time Clock*), modul Transmisi *Wireless* (Modul ESP8266) dan Arduino Pro Mini. Digunakan untuk melakukan pantauan data HR (*Heart Rate*) dan selanjutnya akan masuk *Database Firebase* untuk memudahkan pelatih melakukan pantauan. Dengan menggunakan algoritma tersebut, maka penghematan daya pada *prototype* berhasil didapatkan. Ditunjukkan dengan hasil data durasi waktu pada fungsi `LowPower.powerDown` rata – rata berkisar 80,56 menit, untuk `LowPower.idle` rata – rata berkisar 73,3 menit sedangkan untuk `Delay(1000)` rata – rata berkisar 39,9 menit. *Prototype* tetap dapat berjalan dengan baik serta sesuai fungsi pada penelitian sebelumnya yang telah dilakukan.

**Kata Kunci :** *Internet Of Things, Olahraga, Heart Rate, Idle-Time Deep Sleep, LowPower.powerDown, LowPower.idle, Delay(1000), Firebase.*

## KATA PENGANTAR

Atas berkat rahmat Allah SWT maka penulis berhasil menyelesaikan dan menyusun naskah Tugas Akhir yang berjudul Efisiensi Daya Untuk Pantauan Data *Heart Rate* Menggunakan Metode *Idle-Time Deep Sleep*. Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa untuk meraih gelar kesarjanaan di Jurusan Teknik Komputer, Fakultas Teknologi dan Informatika, Institut Bisnis dan Informatika Stikom Surabaya.

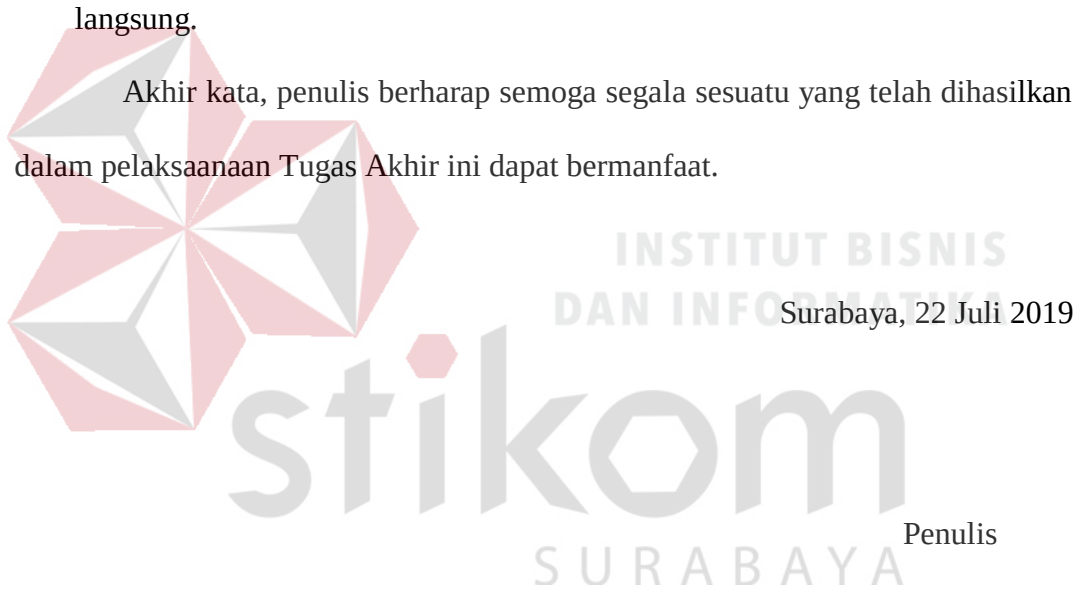
Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Budi Jatmiko, M.Pd., selaku Rektor Institut Bisnis dan Informatika Stikom Surabaya.
2. Bapak Dr. Jusak, selaku Dekan Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika Stikom Surabaya.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Prodi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku dosen pembimbing pertama yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
5. Ibu Musayyanah, S.ST., M.T. selaku dosen pembimbing kedua yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.



6. Bapak Heri Pratikno, M.T., MTCNA., MTCRE. selaku dosen pembahas yang telah memberikan arahan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya yang telah mendidik dan memberikan motivasi kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan baik.
8. Rekan-rekan mahasiswa S1 Teknik Komputer dan semua pihak yang telah memberikan dorongan dan bantuan baik secara langsung maupun secara tidak langsung.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat.



## DAFTAR ISI

|                                                         |      |
|---------------------------------------------------------|------|
| ABSTRAK .....                                           | vii  |
| KATA PENGANTAR .....                                    | viii |
| DAFTAR ISI.....                                         | x    |
| DAFTAR GAMBAR .....                                     | xiv  |
| DAFTAR TABEL .....                                      | xvii |
| BAB I.....                                              | 1    |
| 1.1 Latar Belakang .....                                | 1    |
| 1.2 Rumusan Masalah .....                               | 2    |
| 1.3 Batasan Masalah.....                                | 3    |
| 1.4 Tujuan .....                                        | 3    |
| 1.5 Sistematika Penulisan.....                          | 3    |
| BAB II.....                                             | 5    |
| 2.1 Konsep <i>IoT</i> .....                             | 5    |
| 2.2 <i>Efisiensi Daya</i> .....                         | 6    |
| 2.3 <i>Wearable Device</i> .....                        | 6    |
| 2.4 <i>Idle Time – Deep Sleep Algorithm</i> . ....      | 7    |
| 2.5 <i>Arduino Pro Mini Pro Mini</i> .....              | 8    |
| 2.6 <i>Grove Finger Clip Heart Rate Sensor</i> .....    | 10   |
| 2.7 <i>RTC DS1307</i> .....                             | 11   |
| 2.8 <i>LCD OLED (Organic Light Emiting Diode)</i> ..... | 13   |
| 2.9 <i>ESP8266</i> .....                                | 15   |
| 2.10 <i>Battery Lipo</i> .....                          | 17   |
| 2.11 <i>Switch On Off</i> .....                         | 19   |
| 2.12 <i>Voltage Sensor</i> .....                        | 21   |
| 2.13 <i>Firebase</i> .....                              | 22   |

|                |                                                                      |    |
|----------------|----------------------------------------------------------------------|----|
| 2.14           | <i>Arduino IDE</i> .....                                             | 23 |
| BAB III.....25 |                                                                      |    |
| 3.1            | Metode Penelitian.....                                               | 25 |
| 3.2            | Perancangan Perangkat Keras .....                                    | 25 |
| 3.2.1          | Perancangan <i>part</i> komponen.....                                | 26 |
| 3.3            | Perancangan Program .....                                            | 33 |
| 3.3.1          | <i>Flowchart</i> Sistem pantauan data <i>heart rate</i> .....        | 33 |
| 3.3.2          | <i>Flowchart</i> ESP8266.....                                        | 37 |
| 3.3.3          | Perancangan Software.....                                            | 39 |
| 3.3.4          | Fungsi <i>LowPower.powerDown</i> .....                               | 43 |
| 3.3.4          | Fungsi <i>LowPower.Idle</i> .....                                    | 45 |
| 3.3.5          | Prosedur konfigurasi <i>Arduino IDE</i> .....                        | 47 |
| 3.3.6          | Prosedur membuat <i>Database Firebase</i> .....                      | 50 |
| BAB IV .....58 |                                                                      |    |
| 4.1            | Pengujian <i>Arduino Pro Mini Pro Mini</i> .....                     | 58 |
| 4.1.1          | Tujuan Pengujian <i>Arduino Pro Mini Pro Mini</i> .....              | 58 |
| 4.1.2          | Peralatan Pengujian <i>Arduino Pro Mini Pro Mini</i> .....           | 58 |
| 4.1.3          | Prosedur Pengujian <i>Arduino Pro Mini Pro Mini</i> .....            | 58 |
| 4.1.4          | Hasil Pengujian <i>Arduino Pro Mini Pro Mini</i> .....               | 59 |
| 4.2            | Pengujian <i>Sensor Grove Finger Clip Heart Rate</i> .....           | 60 |
| 4.2.1          | Tujuan Pengujian <i>Sensor Grove Finger Clip Heart Rate</i> .....    | 60 |
| 4.2.2          | Peralatan Pengujian <i>Sensor Grove Finger Clip Heart Rate</i> ..... | 60 |
| 4.2.3          | Prosedur Pengujian <i>Sensor Grove Finger Clip Heart Rate</i> ....   | 60 |
| 4.2.4          | Hasil Pengujian <i>Grove Finger Clip Heart Rate Sensor</i> .....     | 61 |
| 4.3            | Pengujian <i>RTC (Real-Time Clock)</i> .....                         | 61 |
| 4.3.1          | Tujuan Pengujian <i>RTC (Real-Time Clock)</i> .....                  | 61 |

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| 4.3.2 Peralatan Pengujian RTC ( <i>RealTime-Clock</i> ) .....                               | 61 |
| 4.3.3 Prosedur Pengujian RTC ( <i>RealTime-Clock</i> ) .....                                | 62 |
| 4.3.4 Hasil Pengujian RTC ( <i>RealTime-Clock</i> ).....                                    | 63 |
| 4.4 Pengujian Komponen ESP8266 .....                                                        | 64 |
| 4.4.1 Tujuan Komponen ESP8266.....                                                          | 64 |
| 4.4.2 Peralatan Pengujian ESP8266 .....                                                     | 64 |
| 4.4.3 Prosedur Pengujian ESP8266.....                                                       | 64 |
| 4.4.4 Hasil Pengujian ESP8266 .....                                                         | 66 |
| 4.5 Pengujian Komponen <i>Display</i> LCD OLED ( <i>Organic Light Emitting Diode</i> )..... | 68 |
| 4.5.1 Tujuan dari Pengujian Komponen LCD OLED ( <i>Organic Light Emitting Diode</i> ).....  | 68 |
| 4.5.2 Peralatan Pengujian Komponen LCD OLED ( <i>Organic Light Emitting Diode</i> ).....    | 69 |
| 4.5.3 Prosedur pengujian Komponen LCD OLED ( <i>Organic Light Emitting Diode</i> ).....     | 69 |
| 4.5.4 Hasil Pengujian Komponen LCD OLED ( <i>Organic Light Emitting Diode</i> ).....        | 70 |
| 4.6 Efisiensi daya dari pengamatan waktu.....                                               | 71 |
| 4.6.1 Dengan Fungsi Delay (1000) .....                                                      | 71 |
| 4.6.2 Dengan Fungsi LowPower.powerDown .....                                                | 74 |
| 4.6.3 Dengan Fungsi LowPower.idle .....                                                     | 76 |
| 4.6.4 Grafik Data Keseluruhan Fungsi.....                                                   | 79 |
| 4.7 Nilai Ampere Dari Masing – Masing Fungsi.....                                           | 79 |
| BAB V.....                                                                                  | 81 |
| 5.1 Kesimpulan .....                                                                        | 81 |
| 5.2 Saran.....                                                                              | 81 |

|                      |    |
|----------------------|----|
| DAFTAR PUSTAKA ..... | 82 |
| LAMPIRAN.....        | 83 |



## DAFTAR GAMBAR

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| Gambar 2.1 <i>Arduino Pro Mini Pro Mini</i> .....                                           | 8  |
| Gambar 2.2 <i>Sensor Grove Finger Clip Heart Rate</i> .....                                 | 11 |
| Gambar 2.3 <i>RTC (Real-Time Clock)</i> .....                                               | 13 |
| Gambar 2.4 <i>LCD OLED (Organic Light Emiting Diode)</i> .....                              | 14 |
| Gambar 2.5 <i>ESP8266</i> .....                                                             | 16 |
| Gambar 2.6 <i>Baterai Lipo</i> .....                                                        | 18 |
| Gambar 2.7 <i>Switch On Off</i> .....                                                       | 19 |
| Gambar 2.8 <i>Sensor Voltage</i> .....                                                      | 21 |
| Gambar 2.9 <i>Logo Database Firebase</i> .....                                              | 22 |
| Gambar 2.10 <i>Logo Software Arduino Pro Mini IDE</i> .....                                 | 23 |
| Gambar 3.1 <i>Diagram Blok Sistem</i> .....                                                 | 25 |
| Gambar 3.2 <i>Diagram Sistem</i> .....                                                      | 26 |
| Gambar 3.3 <i>Rangkaian Sensor Grove Finger Clip Heart Rate</i> .....                       | 27 |
| Gambar 3.4 <i>Rangkaian ESP8266</i> .....                                                   | 28 |
| Gambar 3.5 <i>Rangkaian RTC (Real-Time Clock)</i> .....                                     | 29 |
| Gambar 3.6 <i>Rangkaian LCD OLED (ORGANIC LIGHT EMITING DIODE)</i> ..                       | 29 |
| Gambar 3.7 <i>Rangkaian Catu Daya</i> .....                                                 | 30 |
| Gambar 3.8 <i>Rangkaian Prototype Keseluruhan</i> .....                                     | 32 |
| Gambar 3.9 <i>Flowchart Keseluruhan Arduino Pro Mini Efisiensi Daya Heart Rate</i><br>..... | 34 |
| Gambar 3.10 <i>Flowchart Proses Kirim Data Serial</i> .....                                 | 36 |
| Gambar 3.11 <i>Flowchart Proses Kirim Data Protokol</i> .....                               | 37 |
| Gambar 3.12 <i>Flowchart Sistem Keseluruhan ESP8266</i> .....                               | 38 |

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Gambar 3.13 <i>Flowchart</i> Kirim Format Data Ke Database Firebase.....  | 39 |
| Gambar 3.14 Tentukan <i>Port</i> .....                                    | 47 |
| Gambar 3.15 <i>Verify Program</i> .....                                   | 48 |
| Gambar 3.16 Tentukan Board dan Konfigurasi ESP .....                      | 49 |
| Gambar 3.17 <i>Verify Program</i> ESP .....                               | 49 |
| Gambar 3.18 Upload Program .....                                          | 50 |
| Gambar 3.19 Buka Konsol Firebase .....                                    | 51 |
| Gambar 3.20 Tampilan awal database .....                                  | 51 |
| Gambar 3.21 Add Project.....                                              | 52 |
| Gambar 3.22 Tentukan Lokasi .....                                         | 53 |
| Gambar 3.23 Tampilan nama atabase .....                                   | 53 |
| Gambar 3.24 <i>Loading</i> pembuatan database .....                       | 54 |
| Gambar 3.25 <i>Database Realtime</i> .....                                | 55 |
| Gambar 3.26 <i>Security Realtime Database</i> .....                       | 55 |
| Gambar 3.27 Fungsi Baca Dan Tulis Database .....                          | 56 |
| Gambar 3.28 <i>publish database</i> .....                                 | 56 |
| Gambar 3.29 Menu Konsol .....                                             | 57 |
| Gambar 4.1 Program Berhasil Diupload.....                                 | 59 |
| Gambar 4.2 Output Serial Sensor <i>Grove Finger Clip Heart Rate</i> ..... | 61 |
| Gambar 4.3 Output Data Serial RTC ( <i>RealTime – Clock</i> ).....        | 63 |
| Gambar 4.4 Hasil Pengujian ESP8266 (1).....                               | 67 |
| Gambar 4.5 Hasil Pengujian ESP8266 (2).....                               | 68 |
| Gambar 4.6 Tampilan LCD OLED (Organic Light Emitting Diode) .....         | 70 |
| Gambar 4.7 Grafik Data Waktu Durasi Fungsi Delay(1000) .....              | 73 |

|                                                                |    |
|----------------------------------------------------------------|----|
| Gambar 4.8 Grafik Data Fungsi LowPower.powerDown.....          | 76 |
| Gambar 4.9 Grafik Data Durasi Waktu Fungsi LowPower.idle ..... | 78 |
| Gambar 4.10 Grafik Data keseluruhan Fungsi Efisiensi.....      | 79 |





## DAFTAR TABEL

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Tabel 2.1 Karakteristik dan Keterangan <i>Arduino Pro Mini Pro Mini</i> .....        | 9  |
| Tabel 2.2 Karakteristik dan Keterangan Sensor <i>Grove Finger Clip Hear Rate</i> ... | 11 |
| Tabel 2.3 Deskripsi Pin <i>LCD OLED (ORGANIC LIGHT EMITING DIODE)</i> ....           | 14 |
| Tabel 2.4 Karakteristik Dan Keterangan Baterai Lipo.....                             | 18 |
| Tabel 4.1 Tabel Hasil Pengamatan Waktu Fungsi Delay(1000).....                       | 72 |
| Tabel 4.2 Tabel Hasil Pengamatan Waktu Fungsi LowPower.powerDown.....                | 75 |
| Tabel 4.3 Tabel Hasil Pengamatan Waktu Fungsi LowPower.idle .....                    | 77 |
| Tabel 4.4 Tabel Nilai Avometer.....                                                  | 80 |



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Gaya hidup sehat dan olahraga adalah suatu aktifitas rutin dan kebiasaan yang dilakukan oleh masyarakat pada era milenial sekarang ini, contohnya seperti makan makanan bergizi dan olahraga yang melatih denyut jantung. Salah satu olahraga yang sangat mudah dilakukan oleh masyarakat untuk melatih denyut jantung adalah dengan melakukan olahraga berlari, karena olahraga ini tidak membutuhkan teknik yang rumit dan tidak membutuhkan biaya yang relatif mahal untuk melakukan olahraga tersebut, olahraga lari sangat mempengaruhi kondisi jantung, khususnya bagi orang yang jarang sekali melakukan olahraga. Jika aktifitas olahraga tidak dicatat/dipantau oleh seorang pelatih olahraga/ personal trainer maka dapat berdampak buruk bagi pelaku olahraga tersebut, bahkan bisa menyebabkan kematian karena serangan jantung. Teknologi *IoT* pada masa sekarang telah mengalami perkembangan yang pesat dan telah diterapkan di berbagai alat yang telah mendukung dan membutuhkan konsep *IoT* sehingga melahirkan *wearable device*, namun untuk kondisi sekarang peralatan tersebut masih belum bisa dikatakan sempurna, karena masih memiliki berbagai kekurangan yang harus diperbaiki, khususnya pada masalah konsumsi daya yang dibutuhkan untuk menjalankan berbagai alat yang telah diciptakan. maka dari itu diperlukan sebuah penelitian lanjutan agar alat-alat tersebut sangat ramah penggunaanya dan lebih mudah dalam menggunakannya. Namun untuk mengimplementasikan konsep *IoT* tidak semudah yang dibayangkan saat ini, karena dituntut untuk memiliki kemampuan Power Management yang baik (Martinez, et al., 2015). Implementasi

teknologi IoT dapat digunakan pada perangkat dengan protokol yang mendukung IoT, seperti protokol *Zigbee (Xbee)*, protokol *Lorawan* dan protokol *wifi*. Namun perangkat tersebut membutuhkan biaya yang besar untuk satu node transmisi. Implementasi IoT biasanya digunakan pada protokol *Wifi (IEEE 802.11)*, contohnya pada modul ESP8266 atau Wemos, namun kedua perangkat tersebut mengkonsumsi daya yang besar, seperti yang telah dilakukan pada penelitian (Musayyanah,2018). “Monitoring Hasil Tembakau Berbasis IoT di Baristan Surabaya .Pada penelitian (Anneke, 2018) (menggunakan metode *idle time deep sleep* untuk pantauan tanaman tembakau, dimana metode tersebut dapat menghemat baterai 75% sesuai dengan jadwal pantauan.

Penerapan *wearable device* yang mendeteksi HR (*Heart Rate*) Berbasis *IoT* membutuhkan konsumsi daya yang besar (Musayyanah,2018). Untuk mengatasi kelemahan masalah daya, maka tugas akhir ini menerapkan *Idle-Time Deep Sleep* pada prototype pengambilan data HR (*Heart Rate*).

## 1.2 Rumusan Masalah

Adapun permasalahan yang akan dihadapi dalam pengerjaan tugas akhir ini diantaranya adalah :

1. Bagaimana menerapkan metode *Idle Time-Deep Sleep* pada *prototype* Monitoring Heart Rate (HR) ?
2. Bagaimana Hasil perbandingan pengukuran Efisiensi daya pada alat yang menggunakan metode *Idle Time – Deep Sleep* dan tanpa menggunakan Metode tersebut?

### 1.3 Batasan Masalah

Untuk menghindari pembahasan yang lebih luas terkait dengan perancangan efisiensi pantauan data *heart rate*. Maka penelitian ini ditentukan pada ruang lingkup sebagai berikut:

1. Menggunakan Sensor *Grove Finger Clip Heart Rate*.
2. Menggunakan *Microcontroler Arduino Pro Mini Pro Mini*.
3. Tampilan Monitor Menggunakan *LCD OLED (ORGANIC LIGHT EMITTING DIODE)*.
4. Menggunakan Baterai Lipo 3.7V 260mAh.
5. Hanya Menguji ketahanan Catu Daya.

### 1.4 Tujuan

Berdasarkan rumusan masalah yang diuraikan di atas, maka tujuan penelitian ini adalah melakukan penelitian dan uji coba Algoritma *Idle Time – Deep Sleep* terhadap *prototype monitoring Heart Rate* untuk menambah fitur efisiensi daya pada *prototype* tersebut.

### 1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini disusun menjadi 5 (lima) garis besar bab pembahasan, yaitu sebagai berikut :

#### **BAB I            PENDAHULUAN**

Pada bab ini menyajikan pembahasan, mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dari, dan sistematika penulisan.

#### **BAB II          LANDASAN TEORI**

Pada bab ini disajikan teori penunjang dari permasalahan, diantaranya Konsep *IoT*, *efisiensi daya*, *wearable device*, *algoritma idle time – deep sleep*, *Arduino Pro Mini Pro Mini*, *Sensor Grove Finger Clip Heart Rate*, *RTC (RealTime-Clock) ds1307*, *ESP8266*, *Voltage sensor*, *LCD OLED (ORGANIC LIGHT EMITTING DIODE)*, *Baterai lipo 3.7v 260mAh*.

### **BAB III METODE PENELITIAN**

Pada bab ini disajikan mengenai tahapan perancangan perangkat lunak dan perangkat keras *Prototype* pemantau data *Heart Rate*. Dijelaskan proses pembuatan sub – sub program dan metode percobaan yang digunakan untuk menguji dan analisis Efisiensi algoritma.

### **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini membahas tentang meliputi pengujian Komponen *Protypet*, pengujian Fungsi *Efisiensi*

### **BAB V PENUTUP**

Pada bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan penelitian selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep *IoT*

Secara umum *internet of things* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Yang memungkinkan daya pengendalian, monitoring, komunikasi, kerja sama dengan berbagai perangkat keras, berbagi data serta memvirtualisasikan segala hal nyata ke dalam bentuk internet, melalui jaringan internet atau disebut juga M2M (*machine to machine*). (Aston,2009)

Selain itu dengan berkembangnya teknologi internet saat ini maka dapat menjadikan fungsi internet lebih beragam seperti hal nya untuk sarana yang memudahkan untuk pengawasan dan pengendalian barang fisik maka konsep IoT ini sangat memungkinkan untuk digunakan hampir pada seluruh kegiatan sehari-hari, mulai dari penggunaan perorangan, perkantoran, rumah sakit, pariwisata, industri, transportasi, konserverasi hewan, pertanian dan peternakan, sampai ke pemerintahan.

Dalam tujuan tersebut, IoT memiliki peran penting dalam pengendalian pemakaian listrik, sehingga pemakaian listrik dapat lebih hemat sesuai kebutuhan mulai dari tingkat pemakaian pribadi sampai ke industri. Tentunya selain untuk tujuan penghematan IoT juga dapat dipakai sebagai sarana kemajuan usaha, dengan sistem monitoring maka kebutuhan usaha dapat lebih terukur.

## 2.2 Efisiensi Daya

*Efisiensi* adalah ukuran tingkat penggunaan sumber daya dalam suatu proses. Semakin hemat/sedikit penggunaan sumber daya, maka prosesnya dikatakan semakin efisien. Proses yang efisien ditandai dengan perbaikan proses sehingga menjadi lebih murah dan lebih cepat. Selain itu *efisiensi* juga dapat diartikan sebagai ketepatan cara (usaha, kerja) dalam menjalankan sesuatu (dengan tidak membuang waktu, tenaga, biaya); kedayagunaan; ketepatangunaan; kesangkilan; kemampuan menjalankan tugas dengan baik dan tepat (dengan tidak membuang waktu, tenaga, biaya) (Setiawan, 2019)

## 2.3 Wearable Device

Perangkat *wearable*, menurut (Sofian, 2013), merupakan generasi selanjutnya dari *smart device*. Sebuah perangkat dengan teknologi mutakhir yang digabungkan dengan perangkat-perangkat yang sudah akrab dengan keseharian kita, seperti jam tangan, kacamata, pakaian, dan lain sebagainya yang mengantarkan tren baru dalam industri mobile. Bisa dikatakan teknologi sekarang sudah mengalami pergeseran yang semula portable menuju wearable. Sementara untuk IoT ia menjelaskan, “IoT adalah sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus guna berbagi data, mengontrol benda-benda fisik via gadget, dan banyak manfaat lainnya. Cara kerjanya adalah dengan memanfaatkan sebuah argumentasi pemrograman yang mana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis melalui *Internet*.”

Kesimpulannya adalah sebuah *wearable device* belum tentu dapat terhubung dengan *Internet* dan *IoT* pun juga belum tentu merupakan sebuah *wearable device*. Meskipun demikian, keduanya juga saling mendukung dan sebenarnya teknologi *wearable device* adalah bagian perkembangan *IoT*.

#### **2.4 Idle Time – Deep Sleep Algorithm.**

Dalam algoritma ini memungkinkan untuk menghemat konsumsi daya yang digunakan, dalam pengiriman data ada 3 mode yang berbeda, antara lain :

1. *Transmit Mode* : dalam mode ini konsumsi daya saat mengirimkan data adalah paling tertinggi dari mode lainnya.
2. *Idle Mode* : Penerima dalam mode siaga dapat menerima atau mengirim, hal itu membutuhkan daya karena harus melihat medium nirkabel sepanjang waktu untuk menentukan apakah ada transmisi paket yang terjadi. mode ini membutuhkan daya yang sedikit lebih rendah daripada mode terima.
3. *Sleep Mode* : Mode tidur memiliki konsumsi daya yang sangat rendah. Antarmuka jaringan baru tidak dapat menerima atau mengirim dalam mode ini kecuali jika dibangunkan ke mode siaga dengan instruksi eksplisit. dibutuhkan waktu yang terbatas untuk beralih dari mode tidur ke mode siaga dan sebaliknya.

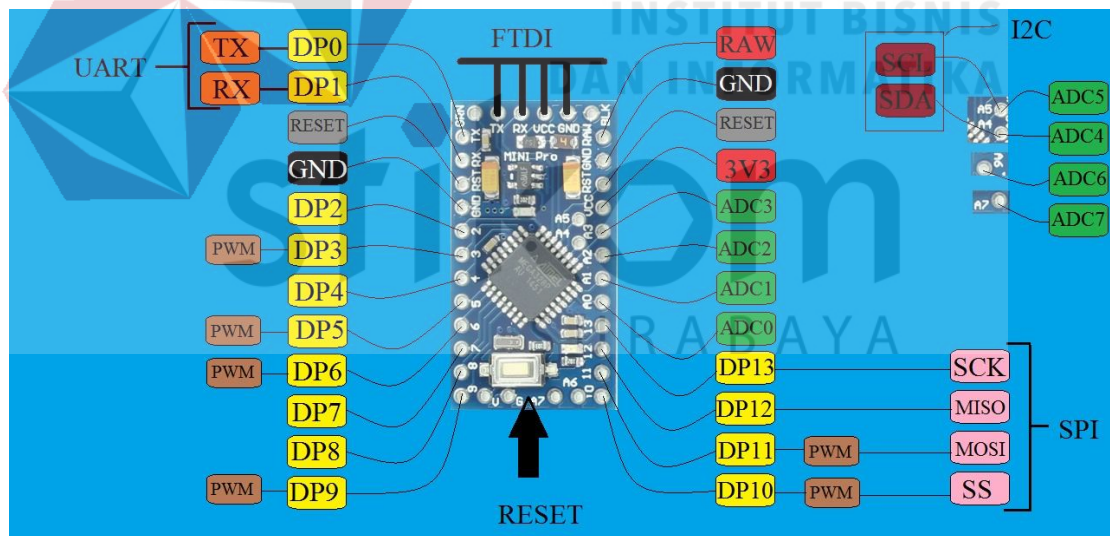
Dalam dunia mikrokontroller, metode ini bisa diterapkan di komponen pendukung Arduino Pro Mini seperti ESP8266 hal ini akan menempatkan MCU dalam mode tidur nyenyak. Mode tidur nyenyak yang memungkinkan pengoptimalan daya dengan waktu bangun paling lambat, kecuali peripheral RTC (RealTime – Clock)



dihentikan. CPU dapat dibangunkan hanya menggunakan RTC (RealTime – Clock) atau menggunakan pada pin external sebagai interupt.

## 2.5 *Arduino Pro Mini Pro Mini.*

*Arduino Pro Mini Pro Mini* adalah *board* mikrokontroler berbasis *ATmega168*. Papan kontroler ini dilengkapi dengan *Arduino Pro Mini* bootloader bawaan. *Arduino Pro Mini* ini memiliki 14 pin input / output digital (6 di antaranya dapat digunakan sebagai output PWM), 8 input analog, resonator *on-board*, tombol reset, dan lubang untuk memasang header pin. Papan dapat dihubungkan ke PC menggunakan port USB dan papan dapat berjalan dengan daya USB. Ada dua versi *Arduino Pro Mini Pro Mini*. Satu berjalan di 3.3V dan 8 MHz, yang lainnya di 5V dan 16 MHz.



Gambar 2.1 *Arduino Pro Mini Pro Mini*

Tabel 2.1 Karakteristik dan Keterangan Arduino Pro Mini Pro Mini

|     |                               |                                                                                                                                                   |
|-----|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | <i>Microcontroller</i>        | <i>ATmega 168</i>                                                                                                                                 |
| 2.  | <i>Operating Voltage</i>      | <i>3.3V or 5V</i>                                                                                                                                 |
| 3.  | <i>Input Voltage</i>          | <i>3.35 – 12 V(3.3V model) or 5 – 12V(5V model)</i>                                                                                               |
| 4.  | <i>Digital I/O Pins</i>       | <i>14(of which 6 provide PWM output)</i>                                                                                                          |
| 5.  | <i>Analog Input Pins</i>      | <i>8</i>                                                                                                                                          |
| 6.  | <i>DC Current per I/O Pin</i> | <i>40 mA</i>                                                                                                                                      |
| 7.  | <i>Flash Memory</i>           | <i>16KB (of which 2 KB used by bootloader)</i>                                                                                                    |
| 8.  | <i>SRAM</i>                   | <i>1 KB</i>                                                                                                                                       |
| 9.  | <i>EEPROM</i>                 | <i>512 bytes</i>                                                                                                                                  |
| 10. | <i>Clock Speed</i>            | <i>8 MHz (3.3V model) or 16 MHz (5V model)</i>                                                                                                    |
| 11. | <i>RAW</i>                    | <i>For supplying a raw (regulated) voltage to the board</i>                                                                                       |
| 12. | <i>VCC</i>                    | <i>The regulated 3.3 or 5 volt supply</i>                                                                                                         |
| 13. | <i>GND</i>                    | <i>Ground pins</i>                                                                                                                                |
| 14. | <i>RX</i>                     | <i>Used to receive TTL serial data</i>                                                                                                            |
| 15. | <i>2 and 3</i>                | <i>Digital I/O pins. These pins can also be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value</i> |
| 16. | <i>3, 5, 6, 9, 10, and 11</i> | <i>Digital I/O pins. They can also be configured to provide 8-bit PWM output</i>                                                                  |
| 17. | <i>10, 11, 12 and 13</i>      | <i>Digital I/O pins. They can also be configured as SPI pins;</i>                                                                                 |

|     |                  |                                                                                                  |
|-----|------------------|--------------------------------------------------------------------------------------------------|
|     |                  | <i>10 - (SS), 11 - (MOSI), 12 - (MISO) and 13 - (SCK)</i>                                        |
| 18. | <i>A0 to A3</i>  | <i>Analog input pins</i>                                                                         |
| 19. | <i>A4 and A5</i> | <i>Analog input pins. They can also be used as IIC pins;<br/><br/>A4 - (SDA) and A5 - (SCL).</i> |
| 20. | <i>A6 and A7</i> | <i>Analog input pins</i>                                                                         |
| 21. | <i>Reset</i>     | <i>The microcontroller can be reset by bringing this pin low</i>                                 |

## 2.6 Grove Finger Clip Heart Rate Sensor

*Grove Finger Clip Heart Rate* adalah sensor yang bekerja dengan memanfaatkan cahaya. Saat sensor ini diletakkan di permukaan kulit, sebagian besar cahaya akan diserap dan dipantulkan oleh organ dan jaringan (kulit, tulang, otot, darah) namun sebagian cahaya akan melewati jaringan tubuh yang cukup kecil. Ketika jantung memompa darah melalui tubuh, dari setiap denyut yang terjadi, timbul gelombang pulsa (jenis seperti gelombang kejut) yang bergerak di sepanjang arteri dan menjalar ke jaringan kapiler di mana sensor pulsa terpasang. Sensor ini bekerja berdasarkan teknologi yang dapat mengukur variasi gerakan darah pada pembuluh darah manusia. Sensor ini mendeteksi detak jantung yang terintegrasi dengan STM32F102 (ARM Cortex-M3), dan dilengkapi dengan antarmuka SWD yang memungkinkan user memprogram ulang.

Adapun fitur yang dimiliki oleh sensor *Grove Finger Clip Heart Rate* ini adalah sebagai berikut :

Tabel 2.2 Karakteristik dan Keterangan Sensor *Grove Finger Clip Heart Rate*

|                  |                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Model            | <i>Grove Finger Clip Heart Rate</i>                                                                                                                                 |
| Tegangan         | 3.5 – 5V DC                                                                                                                                                         |
| Konsumsi daya    | 1.5 mA pada kondisi normal, 160 $\mu$ A pada kondisi <i>Sleep Mode</i> ,40 $\mu$ A pada kondisi <i>Sleep Mode2</i> , 15 $\mu$ A pada kondisi <i>power down mode</i> |
| Temperatur kerja | -20 sampai +60°C                                                                                                                                                    |
| Max Frame Rate   | 3000 Fps                                                                                                                                                            |
| Ukuran           | 3.0 * 4.7 cm                                                                                                                                                        |

Gambar 2.2 Sensor *Grove Finger Clip Heart Rate*

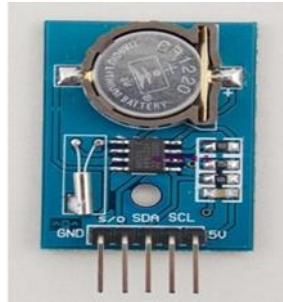
## 2.7 RTC DS1307

Komponen *Realtime clock* merupakan komponen IC penghitung waktu yang dapat difungsikan sebagai sumber data waktu baik berupa data jam, hari, bulan

maupun tahun. Komponen DS1307 ini berupa IC yang perlu dilengkapi dengan komponen pendukung lainnya seperti *crystal* sebagai sumber *clock* dan *Battery External* 3,6 Volt sebagai sumber energi cadangan agar fungsi penghitung tetap berjalan. Bentuk dari komunikasi data dari IC RTC adalah I2C yang merupakan kepanjangan dari *Inter Integrated Circuit*. Komunikasi dengan jenis ini hanya menggunakan 2 jalur komunikasi yaitu SCL dan SDA, dan semua Mikrokontroler sudah dilengkapi dengan fitur komunikasi 2 jalur ini, termasuk diantaranya adalah Arduino Pro Mini, dan tingkat ketelitian error pada RTC (RealTime-Clock) adalah 1 menit di setiap tahunnya.

Adapun spesifikasi dan fitur yang dimiliki RTC (*Realtime-Clock*) adalah sebagai berikut :

1. Pin Vcc / pin nomor 8 berfungsi sebagai sumber energi listrik utama. Tegangan kerja dari komponen ini adalah 5 volt, dan ini sesuai dengan tegangan kerja dari Arduino Pro Mini.
2. Pin GND / pin no 4 harus terhubung dengan ground yang dimiliki oleh komponen RTC dengan ground dari *battery back-up* / *battery CMOS*.
3. SCL berfungsi untuk saluran clock sebagai komunikasi data antara *Microcontroller* dengan RTC (*Realtime Clock*).
4. SDA berfungsi sebagai saluran Data untuk komunikasi data antara *Microcontroller* dengan RTC
5. X1 dan X2 berfungsi sebagai saluran clock yang memiliki sumber dari *crystal external*
6. Vbatt / Baterai CMOS Berfungsi sebagai baterai saluran energi listrik eksternal, agar fungsi penghitung tetap berjalan.



Gambar 2.3 RTC (*Real-Time Clock*)

## 2.8 *LCD OLED (Organic Light Emitting Diode)*

*LCD OLED (Organic Light Emitting Diode)* merupakan media display output pada modul Arduino Pro Mini atau controller lainnya. LCD ini memiliki kelebihan di sisi *pixel* nya yang sangat tajam serta tidak memerlukan cahaya tambahan (*Backlight*) sehingga sangat hemat dalam konsumsi daya. Namun *LCD OLED (Organic Light Emitting Diode)* memiliki kekurangan yaitu ukuran yang relative kecil serta kebanyakan masih satu warna meskipun sudah ada beberapa yang sudah berwarna (RGB) tentunya dengan harga yang sedikit mahal. Namun akhir – akhir ini *LCD OLED (Organic Light Emitting Diode)* tampil dengan interface komunikasi I2C dengan controller, sebelumnya *LCD OLED (Organic Light Emitting Diode)* jenis ini masih menggunakan koneksi serial.

Adapun Spesifikasi yang disajikan dalam *LCD OLED (Organic Light Emitting Diode)* ini adalah :

1. Ukuran LCD + Board : 2.7 x 2.7 cm
2. Ukuran layer LCD : 2.65 x 1.5 cm
3. Resolusi Layar : 128 x 64 pixel dengan 160° sudut pandang mata.
4. Warna Pixel : Kombinasi(RGB), Putih, Biru.
5. Komunikasi : Menggunakan komunikasi I2C

6. Tegangan yang disarankan : 3.3 – 5 V



Gambar 2.4 LCD OLED (Organic Light Emitting Diode)

Tabel 2.3 Deskripsi Pin LCD OLED (Organic Light Emitting Diode)

| Pin No: | Pin Name:           | Description                                                                         |
|---------|---------------------|-------------------------------------------------------------------------------------|
| 1       | Ground (Gnd)        | Connected to the ground of the circuit                                              |
| 2       | Supply (Vdd,Vcc,5V) | Can be powered by either 3.3V or 5V                                                 |
| 3       | SCK (D0,SCL,CLK)    | The display supports both IIC and SPI, for which clock is supplied through this pin |
| 4       | SDA (D1,MOSI)       | This is the data pin of the both, it can either be used for IIC or for SPI          |

|   |                         |                                                                                       |
|---|-------------------------|---------------------------------------------------------------------------------------|
| 5 | <i>RES(RST,RESET)</i>   | <i>When held to ground momentarily this pin resets the module</i>                     |
| 6 | <i>DC (A0)</i>          | <i>This is command pin, can either be used for SPI or for IIC</i>                     |
| 7 | <i>Chip Select (CS)</i> | <i>Normally held low, used only when more than one SPI device is connected to MCU</i> |

## 2.9 ESP8266

*ESP8266* adalah modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti *Arduino Pro Mini* agar dapat terhubung langsung dengan jaringan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3V dengan memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis *ESP8266* yang digunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya yang ada pada mikrokontroler. Perangkat ini bisa diprogram menggunakan *ESPlorer* dan untuk firmware berbasis *NodeMCU* bisa menggunakan *putty* sebagai terminal kontrol, dan bisa juga diprogram menggunakan *Arduino Pro Mini IDE*, dengan penambahan library *ESP8266* pada *board manager*.





Gambar 2.5 ESP8266

Adapun fitur dan spesifikasi ESP8266 adalah sebagai berikut :

1. *802.11 b/g/n protocol*
2. *Wi-Fi Direct (P2P), soft-AP*
3. *Integrated TCP/IP protocol stack*
4. *Integrated TR switch, balun, LNA, power amplifier and matching network*
5. *Integrated PLL, regulators, and power management units*
6. *+19.5dBm output power in 802.11b mode*
7. *Integrated temperature sensor*
8. *Supports antenna diversity*
9. *Power down leakage current of  $< 10\mu A$*
10. *Integrated low power 32-bit CPU could be used as application processor*
11. *SDIO 2.0, SPI, UART*
12. *STBC,  $1 \times 1$  MIMO,  $2 \times 1$  MIMO*
13. *A-MPDU & A-MSDU aggregation &  $0.4\mu s$  guard interval*
14. *Wake up and transmit packets in  $< 2ms$*
15. *Standby power consumption of  $< 1.0mW$  (DTIM3)*

Datasheet ESP8266 adalah sebagai berikut :

1. *Peripheral Bus : UART/SDIO/SPI/I2C/I2S/IR RemoteControl*  
: *GPIO/PWM*
2. *Operating Voltage : 3.0 – 3.6 v*
3. *Operating Current : 80mA*
4. *Operating Temperature : -40°C - +125°C*
5. *Package Size : 5x5mm*
6. *Internal Interface : N/A*

### 2.10 Battery Lipo

Baterai lipo merupakan baterai yang biasa digunakan pada dunia RC (*Remote Control*) dimana baterai ini memiliki daya yang lebih besar dibandingkan dengan baterai jenis lainnya yang mempunyai berat yang sama, ada beberapa kelebihan baterai lipo, antara lain :

1. Baterai LiPo memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran.
2. Baterai LiPo memiliki kapasitas penyimpanan energi listrik yang besar.
3. Baterai LiPo memiliki tingkat *discharge rate* energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang elektronik.

Namun baterai lipo tergolong baterai yang memiliki harga yang lebih mahal daripada baterai jenis lainnya, dan performa baterai lipo yang tinggi juga mengakibatkan umur baterai yang tergolong pendek sekitar 300 – 400 kali siklus pengisian baterai.



Gambar 2.6 Baterai Lipo

Tabel 2.4 Karakteristik Dan Keterangan Baterai Lipo

| S/N | Detail                        | Parameters               | Remarks                                                                                 |
|-----|-------------------------------|--------------------------|-----------------------------------------------------------------------------------------|
| 1   | <i>Rated Voltage</i>          | 3.7 V                    |                                                                                         |
| 2   | <i>Rated Capacity</i>         | 2000Mah                  | discharge with 0.2C to 2.75V after fully charge within 1h, measuring the discharge time |
| 3   | <i>Limited charge voltage</i> | 4.2V                     |                                                                                         |
| 4   | <i>Internal resistance</i>    | $\leq 160\text{m}\Omega$ |                                                                                         |
| 5   | <i>charge mode</i>            | C.C/C.V.                 |                                                                                         |
| 6   | <i>Charge time</i>            | 6H                       | Standard charging 0.2C 400mA                                                            |
| 7   | <i>Max Charge Current</i>     | 2000mA                   |                                                                                         |

### 2.11 Switch On Off

Merupakan saklar tekan yang berfungsi untuk menyambung dan memutus arus listrik dari sumber tegangan arus listrik ke perangkat atau komponen yang menyambung ke sumber tegangan arus listrik tersebut.



Gambar 2.7 Switch On Off

Spesifikasi :

*Circuit/Function : DPST/ON-OFF*

*SPST/ON-OFF*

*DPST/Momentary ON-OFF*

*SPST/Momentary ON-OFF*

*DPST/ON-OFF, illuminated*

*SPST/ON-OFF, illuminated*

*Terminal Type : 4.8mm × 0.8mm quick connect solder tail*

*Contact Gap :  $\geq 3\text{mm}$*

*Insulation Spacing :  $\geq 8\text{mm}$*

*Contact Resistance ( $m\Omega$ ) :  $< 20$  Initially*

*Dielectric Strength (kV) : Up to 3.5*

*Insulation Resistance (MΩ) : > 50 (500V DC between the Open Contacts)*

*Initially*

*Endurance*

*Mechanical Life :  $\geq 100,000$  Cycles*

*Electrical Life :  $\geq 20,000$  Cycles at Full Load*

*Panel Cut-out*

*Minimum Panel Thickness : 1mm*

*Cut-out Size : 19.3mm × 13mm*

*Maximum Panel Thickness : 3mm*

*Cut-out Size : 19.5mm × 13mm*

*Operating Force (N)*

*Maximum Double-Pole : 10*

*Maximum Single-Pole : 7*

*Material*

*Button/ Body : Nylon 6/6*

*Illuminated Button : Polycarbonate*

*Terminals : Copper Alloy*

*Contacts : Silver Alloy*

*Soldering Parameters*

*Soldering Iron Wattage : 50W*

*Tip Temperature : 300°C*

*Soldering time : Within 3 Seconds in Off Position*

## 2.12 Voltage Sensor



Gambar 2.8 Sensor Voltage

Modul sederhana namun sangat berguna, menggunakan pembagi potensial untuk mengurangi tegangan input apa pun dengan faktor 5. Hal ini memungkinkan untuk menggunakan input analog mikrokontroler untuk memonitor tegangan jauh lebih tinggi daripada yang mampu dirasakannya. Misalnya dengan rentang input analog 0V - 5V, dapat mengukur tegangan hingga 25V. Modul ini juga mencakup terminal sekrup yang mudah digunakan untuk koneksi kabel yang mudah dan aman. Modul ini didasarkan pada prinsip desain pembagi tegangan resistif, & dapat membuat tegangan input konektor terminal merah hingga 5 kali lebih kecil. Tegangan input analog Arduino Pro Mini hingga 5V, menghasilkan tegangan input modul deteksi tegangan tidak lebih besar dari  $5V \times 5 = 25V$  (jika Anda menggunakan sistem 3.3V, tegangan input tidak lebih besar dari  $3.3V \times 5 = 16.5V$  ).

Spesifikasi :

1. *Divider Ratio: 5:1*

2. *Resistor Tolerance: 1%*
3. *Max Input Voltage: 25V*
4. *Resister Value: 30K/7.5K Ohm*

*Datasheet :*

1. *GND - This is where you connect the low side of the voltage you are measuring.*

*Caution: This is the same electrical point as your Arduino Pro Mini ground.*

2. *VCC - This is where you connect the high side of the voltage your are measuring (0 - 25V)*

3. *S: This connects to your Arduino Pro Mini analog input.*

4. *-: This connects to your Arduino Pro Mini ground.*

5. *+: This is not connected. It does absolutely nothing.*

### 2.13 *Firestore*



Gambar 2.9 Logo Database *Firestore*

*Firestore* adalah suatu layanan dari *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya *Firestore*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua fitur yang menarik dari *Firestore* yaitu *Firestore Remote Config* dan *Firestore Realtime Database*. Selain itu terdapat fitur

pendukung untuk aplikasi yang membutuhkan pemberitahuan yaitu *Firebase Notification*.

#### 2.14 *Arduino IDE*



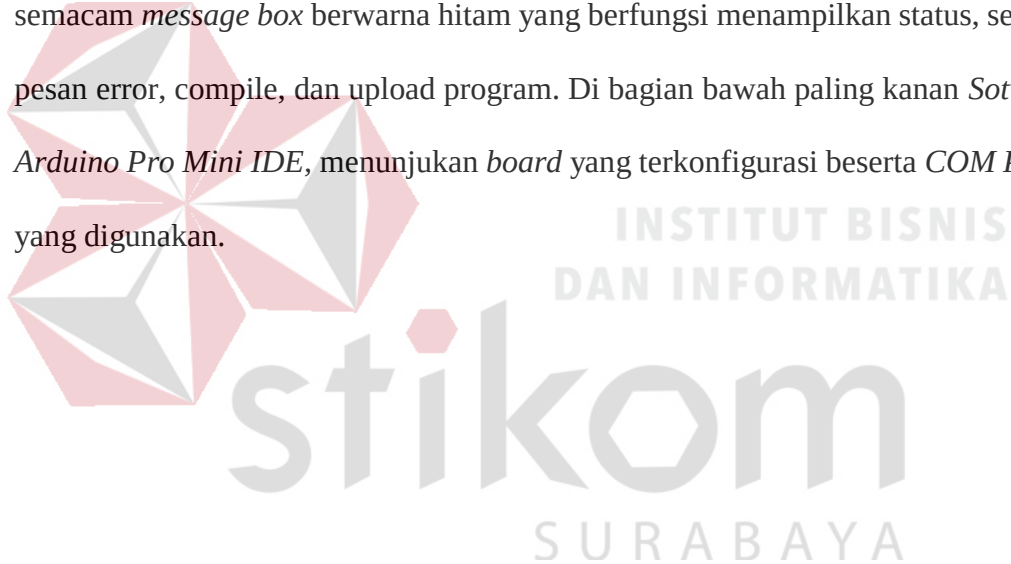
Gambar 2.10 Logo Software Arduino Pro Mini IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino Pro Mini dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman. Arduino Pro Mini menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino Pro Mini (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino Pro Mini telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino Pro Mini dengan mikrokontroler. Arduino Pro Mini IDE dibuat dari bahasa pemrograman JAVA. Arduino Pro Mini IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi input dan output menjadi lebih



mudah. Arduino Pro Mini IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino Pro Mini IDE khusus untuk pemrograman dengan Arduino Pro Mini.

Program yang ditulis dengan menggunakan *Arduino Pro Mini Software* (IDE) disebut sebagai *sketch*. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi .ino. Teks editor pada Arduino Pro Mini Software memiliki fitur” seperti cutting/paste dan seraching/replacing sehingga memudahkan kamu dalam menulis kode program. Pada Software Arduino Pro Mini IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan error, compile, dan upload program. Di bagian bawah paling kanan *Sotware Arduino Pro Mini IDE*, menunjukan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan.

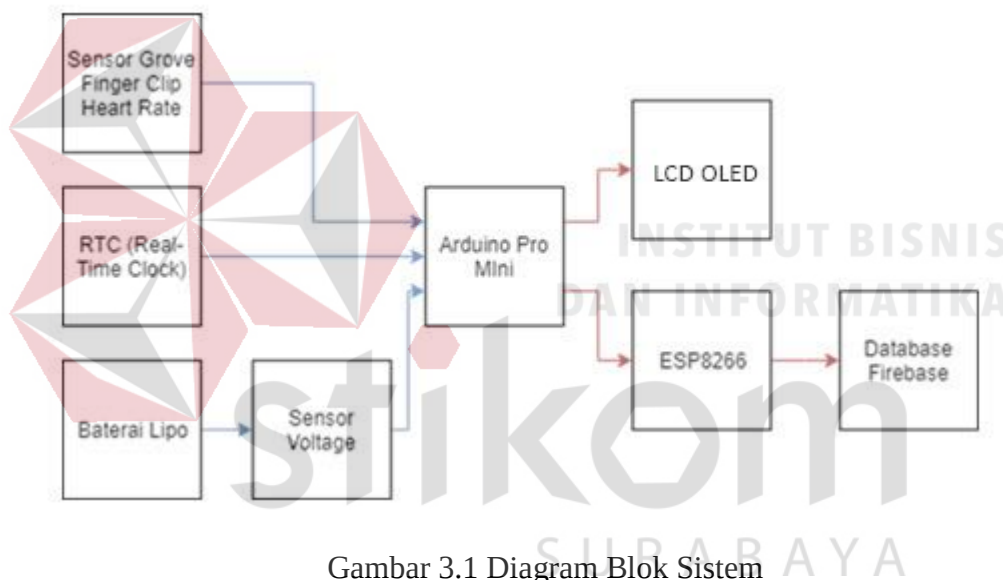


## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

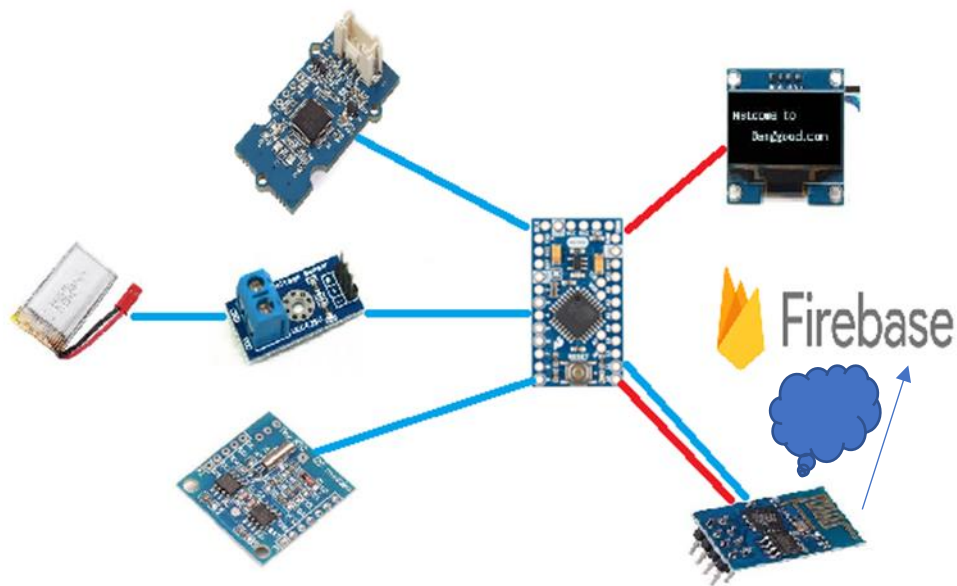
Pada bab ini akan dibahas dan dijelaskan mengenai beberapa tahap yang akan dikerjakan. Mulai dari perencanaan (sistem pengendali), pengujian (sistem pengendali), dan pembuatan laporan penelitian. Perancangan ini dibuat supaya pengerjaan pada tahapan selanjutnya dapat dilakukan dengan lancar dan berurutan. Sesuai dengan perencanaan sebelumnya.



Gambar 3.1 Diagram Blok Sistem

#### 3.2 Perancangan Perangkat Keras

Pada tahap ini akan dijelaskan bagaimana cara merancang perangkat keras alat dan dijelaskan komponen apa saja yang bertugas sebagai *input* data, komponen pengolah data dan komponen penyaji data yang digunakan dan cara kerja alatnya.



Gambar 3.2 Diagram Sistem

Keterangan :

Merah : Output.

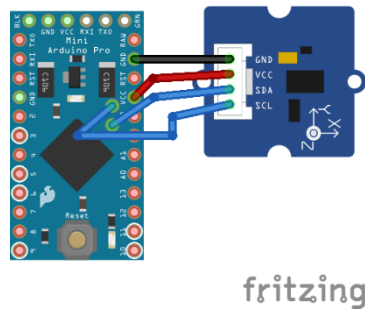
Biru : Input.

Cloud : Transmisi/pengiriman data.

### 3.2.1 Perancangan part komponen.

#### A. Sensor Grove Finger Clip Heart Rate

Pada gambar 3.3 merupakan rangkaian sensor *Grove Finger Clip Heart Rate* yang digunakan untuk sensor hitung detak jantung dan keluaran dari sensor ini akan dikirimkan ke *Arduino Pro Mini Pro Mini* untuk ditampilkan pada *LCD OLED (Organic Light Emitting Diode)* dan untuk data yang dikirim *ESP8266* ke *Database Firebase*.



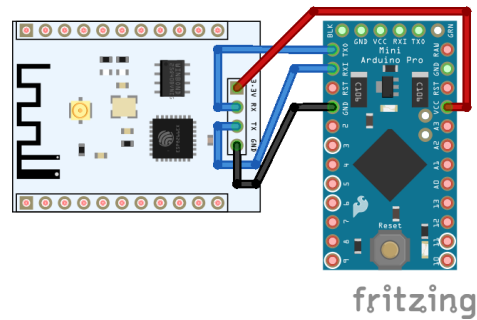
Gambar 3.3 Rangkaian Sensor *Grove Finger Clip Heart Rate*

Keterangan *port* pada Gambar 3.3:

1. Pin GND pada sensor terhubung dengan pin GND pada *Arduino Pro Mini Pro Mini*.
2. Pin VCC pada sensor terhubung dengan pin VCC pada *Arduino Pro Mini Pro Mini*.
3. Pin SDA pada sensor terhubung dengan pin A4/SDA pada *Arduino Pro Mini Pro Mini*.
4. Pin SCL pada sensor terhubung dengan pin A5/SCL pada *Arduino Pro Mini Pro Mini*.

#### B. ESP8266

Pada gambar 3.4 merupakan rangkaian *ESP8266* yang digunakan untuk mengirim data dari mikrokontroller *Arduino Pro Mini Pro Mini* dan kemudian dikirimkan ke *Database Firebase*.



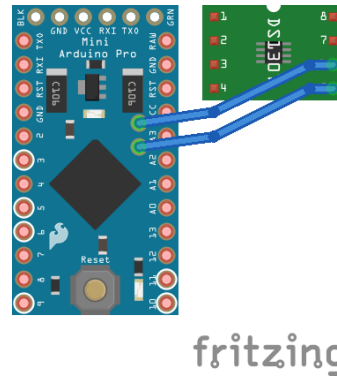
Gambar 3.4 Rangkaian *ESP8266*

Keterangan *port* pada Gambar 3.4:

1. Pin VCC pada ESP8266 terhubung dengan pin VCC pada *Arduino Pro Mini*.
2. Pin GND pada ESP 8266 terhubung dengan pin GND pada *Arduino Pro Mini*.
3. Pin RX pada ESP8266 terhubung dengan pin TX pada *Arduino Pro Mini*.
4. Pin TX pada ESP8266 terhubung dengan pin RX pada *Arduino Pro Mini*.

#### C. RTC (*Real-Time Clock*) DS1307

Pada gambar 3.5 merupakan rangkaian *RTC(Real-Time Clock) DS1307* yang digunakan untuk menunjukkan data waktu secara *real time*, kemudian data waktu tersebut dikirimkan ke microkontroller *Arduino Pro Mini* kemudian dikirimkan ke ESP 8266 yang selanjutnya dikirim ke *Database Firebase*.



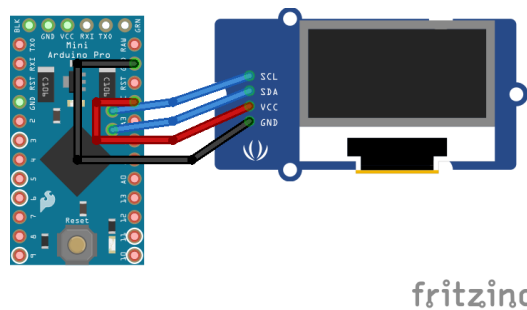
Gambar 3.5 Rangkaian RTC (*Real-Time Clock*)

Keterangan *port* pada Gambar 3.5:

1. Pin SDA/6 pada RTC terhubung dengan PIN SDA pada Microkontroler *Arduino Pro Mini Pro Mini*.
2. Pin SCL/5 pada RTC terhubung dengan PIN SCL pada Microkontroler *Arduino Pro Mini Pro Mini*.

#### D. LCD OLED (*Organic Light Emitting Diode*)

Pada gambar 3.6 merupakan rangkaian *LCD OLED (Organic Light Emitting Diode)* yang terhubung dengan *Arduino Pro Mini Pro Mini* yang digunakan untuk menampilkan data dari sensor *Heart Rate* dan sisa daya baterai.



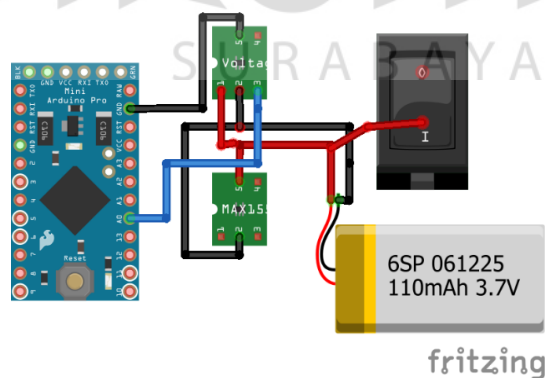
Gambar 3.6 Rangkaian LCD OLED (*Organic Light Emitting Diode*)

Keterangan *port* pada Gambar 3.6:

1. Pin VCC pada LCD OLED (*Organic Light Emiting Diode*) terhubung dengan pin VCC pada Arduino Pro Mini Pro Mini.
2. Pin GND pada LCD OLED (*Organic Light Emiting Diode*) terhubung dengan pin GND pada Arduino Pro Mini Pro Mini.
3. Pin SDA pada LCD OLED (*Organic Light Emiting Diode*) terhubung dengan pin SDA/A4 pada Arduino Pro Mini Pro Mini.
4. Pin SCL pada LCD OLED (*Organic Light Emiting Diode*) terhubung dengan pin SCL/A5 pada Arduino Pro Mini Pro Mini.

E. Baterai, sensor *voltage* dan *charger*.

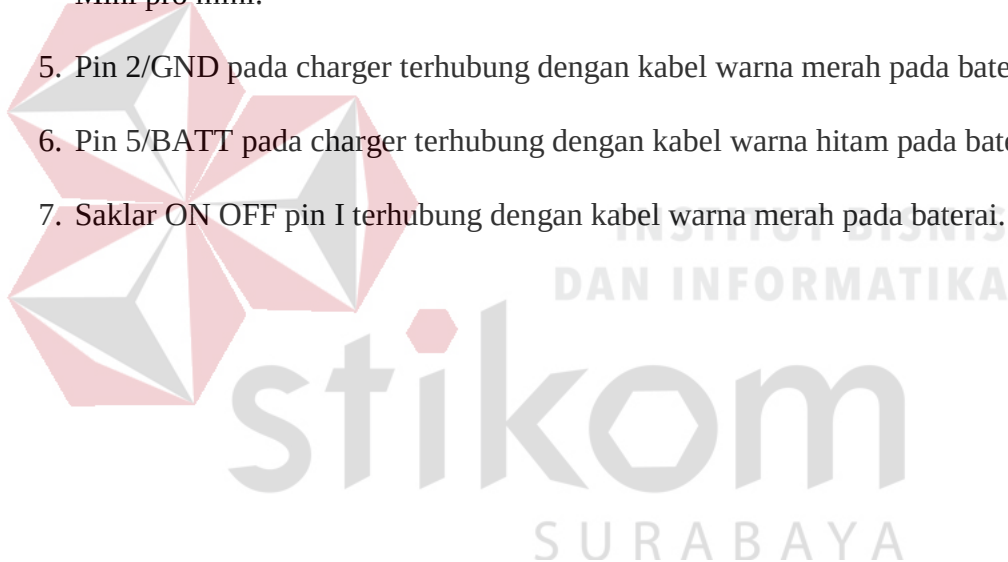
Pada gambar 3.7 merupakan rangkaian catu daya dan sensor voltase yang berfungsi untuk mensuplai daya pada alat/*device* dan untuk mengetahui voltase yang masih ada pada baterai. Dan rangkaian *charger* digunakan untuk men-*charge* baterai ketika daya nya habis.



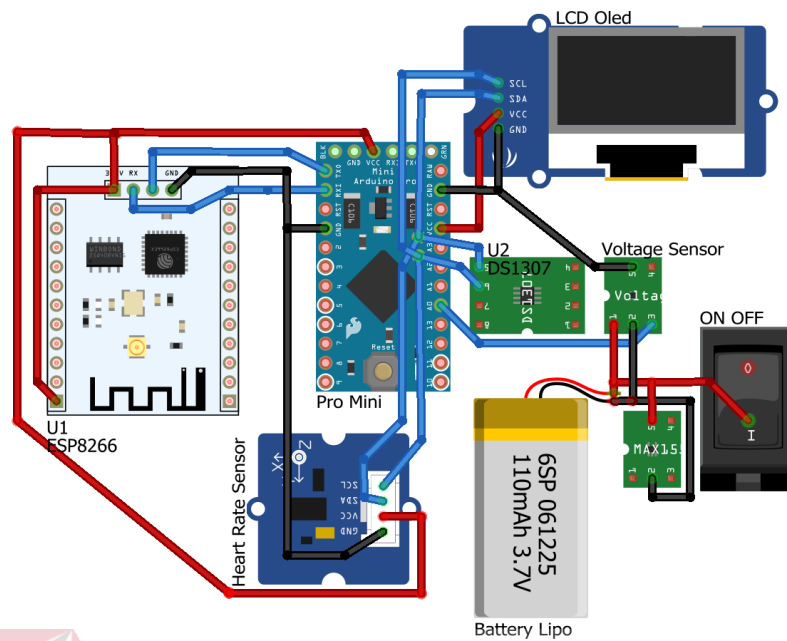
Gambar 3.7 Rangkaian Catu Daya

Keterangan *port* pada Gambar 3.7:

1. Pin 1/IN pada sensor voltage terhubung dengan kabel warna merah pada baterai.
2. Pin 2/GND pada sensor voltage tergubung dengan kabel warna hitam pada baterai.
3. Pin 3/EN pada sensor voltage terhubung dengan pin A0 pada Arduino Pro Mini pro mini.
4. Pin 5/OUT pada sensor voltage terhubung dengan pin GND pada Arduino Pro Mini pro mini.
5. Pin 2/GND pada charger terhubung dengan kabel warna merah pada baterai.
6. Pin 5/BATT pada charger terhubung dengan kabel warna hitam pada baterai.
7. Saklar ON OFF pin I terhubung dengan kabel warna merah pada baterai.







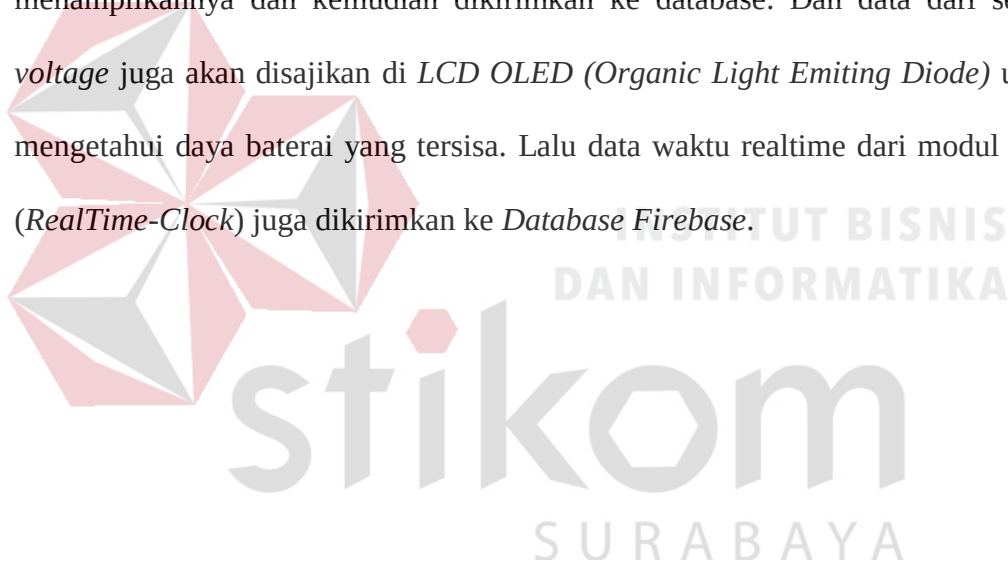
Gambar 3.8 Rangkaian *Prototype* Keseluruhan

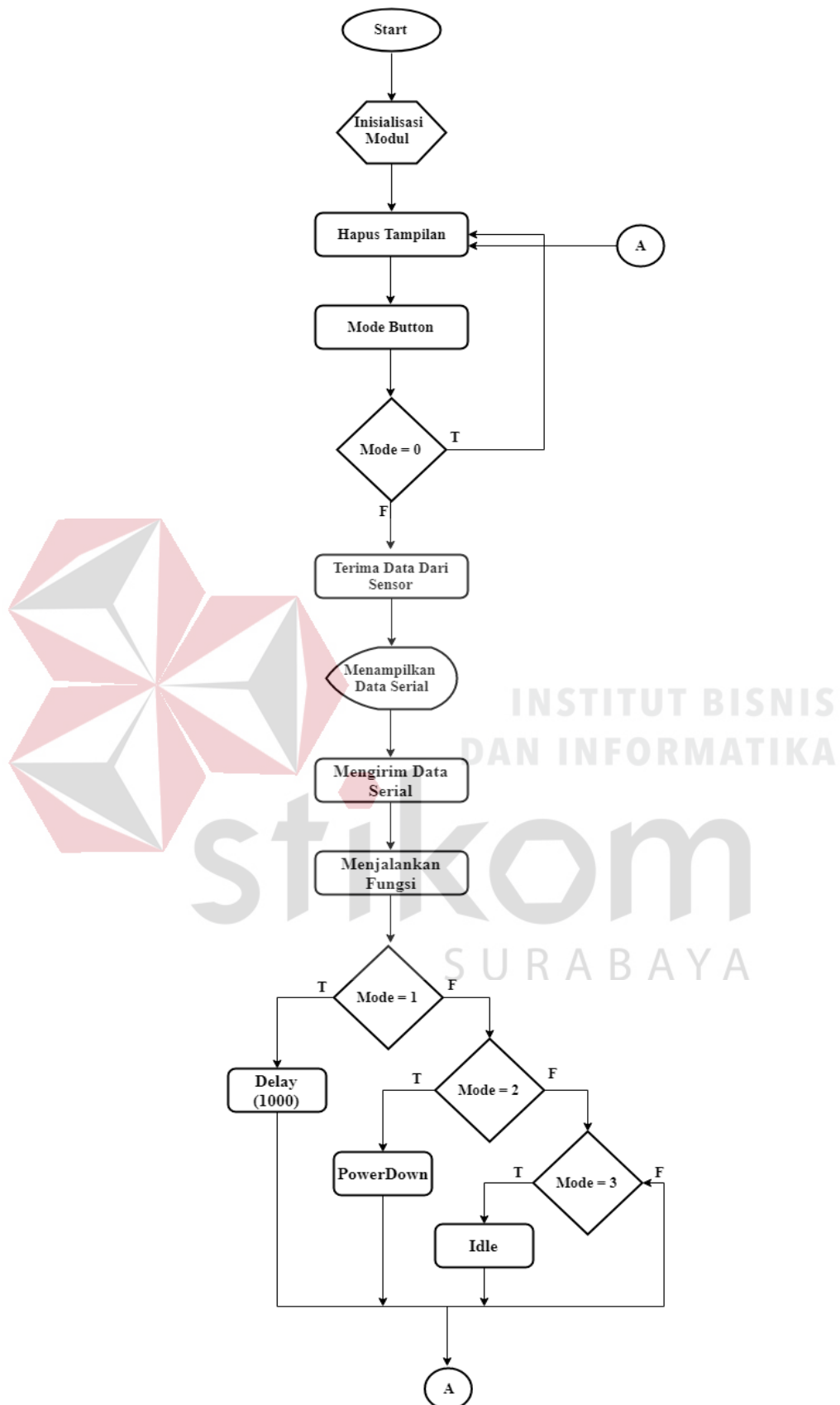
Dalam penjelasan kali ini akan dijelaskan bagaimana cara kerja *prototype* seperti pada Gambar 3.8, disini cara kerja *prototype* adalah dengan membaca sinyal dari sensor *Grove Finger Clip Heart Rate* kemudian sinyal akan dibaca oleh mikrokontroller *Arduino Pro Mini* dan ditampilkan di *LCD OLED (ORGANIC LIGHT EMITING DIODE)*, dan fungsi dari *ESP8266* adalah untuk transmisi data melalui sinyal *wifi*, yang nantinya bisa di akses di *desktop*, dan komponen *RTC* digunakan untuk menampilkan waktu ambil data sedangkan komponen sensor *voltage* digunakan untuk mengetahui voltase batrei yang tersisa. Untuk komponen yang dimatikan akan dianalisis terlebih dahulu, komponen mana yang sekiranya mengkonsumsi daya yang besar. Dan *switch On Off* disini digunakan untuk menghidupkan dan mematikan *prototype*.

### 3.3 Perancangan Program

#### 3.3.1 *Flowchart* Sistem pantauan data *heart rate*

Pada gambar 3.9 *flowchart* dapat dijelaskan bagaimana sistem pantauan data heart rate bekerja. Ketika sistem sudah menyala, maka pembacaan data *heart rate*, daya baterai dan data waktu RTC (RealTime – Clock) oleh sensor sudah bisa dilakukan dan data dari sensor akan diolah di *Arduino Pro Mini Pro Mini* untuk kemudian disajikan dalam bentuk nilai *heart rate* guna mempermudah pantauan maka digunakan *LCD OLED (Organic Light Emiting Diode)* untuk menampilkannya dan kemudian dikirimkan ke database. Dan data dari sensor *voltage* juga akan disajikan di *LCD OLED (Organic Light Emiting Diode)* untuk mengetahui daya baterai yang tersisa. Lalu data waktu realtime dari modul RTC (*RealTime-Clock*) juga dikirimkan ke *Database Firebase*.





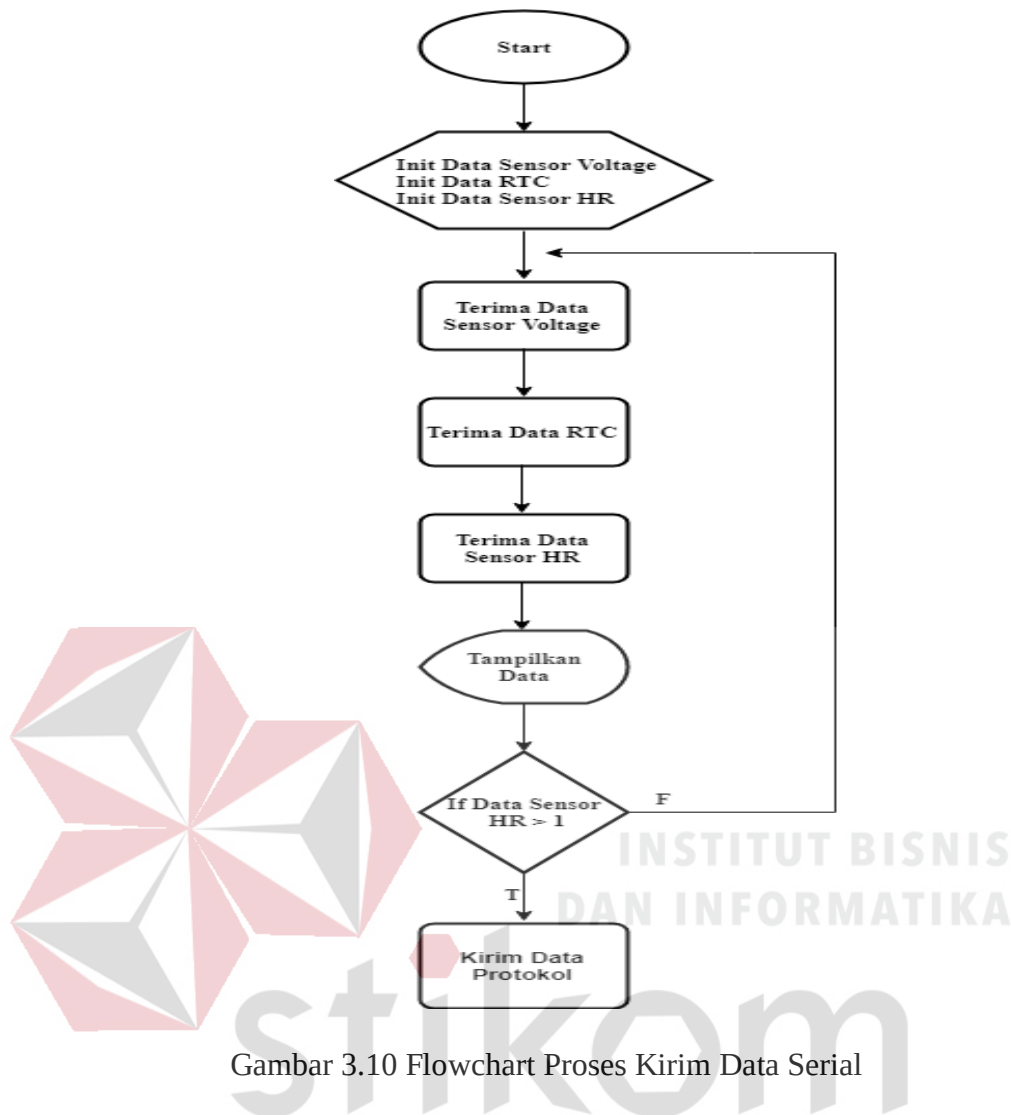
Gambar 3.9 *Flowchart* Keseluruhan Arduino Pro Mini Efisiensi Daya *Heart Rate*

Dari data yang diperoleh dari sensor detak jantung, sensor daya dan RTC (*RealTime – Clock*) maka selanjutnya akan diolah di dalam Arduino Pro Mini dan dikirimkan ke ESP 8266 untuk selanjutnya dikirim ke Database Firebase, data yang dikirimkan adalah data nilai keluaran sensor jantung, data daya baterai dari sensor daya kemudian data waktu realtime dari modul RTC (*RealTime – Clock*). Sehingga pada database akan muncul ketiga data tersebut. Pada sistem ini juga ditambahkan fitur switch untuk merubah tiap mode efisiensi.

Pada sistem ini juga diberikan fungsi berupa *lowPower.powerDown*, *lowPower.idle* dan *delay(1000)*. Untuk membandingkan efisiensi daya dan yang dikonsumsi serta durasi waktu jika menambahkan algoritma atau fungsi tersebut.

Perubahan fungsi dapat dilakukan dengan cara merubah *switch* yang ada pada *prototype* Efisiensi Daya Untuk Pantauan Data *Heart Rate* Menggunakan Metode *Idle Time – Deep Sleep*

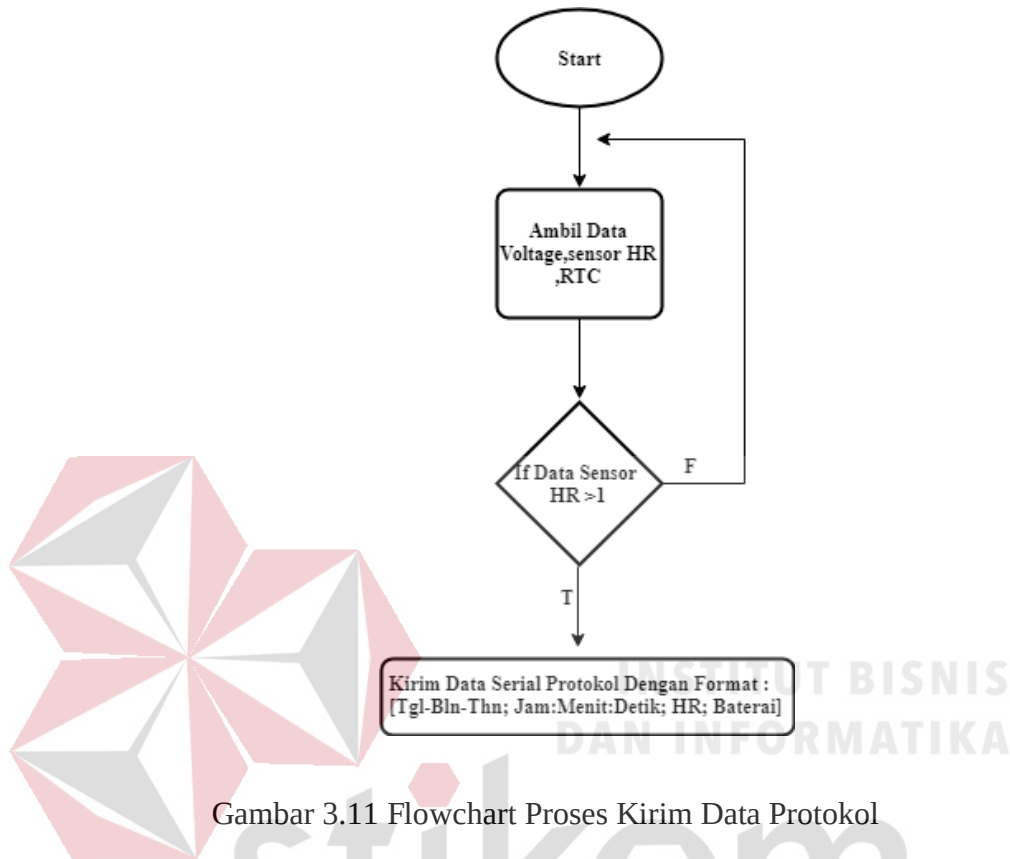




Gambar 3.10 Flowchart Proses Kirim Data Serial

Dalam gambar 3.10 flowchart ini menjelaskan alur pengiriman data serial, mulai dari menerima data dari masing – masing komponen *input* lalu masuk ke Arduino Pro Mini pro mini untuk diolah datanya, kemudian akan di tampilkan pada LCD OLED (*Organic Light Emitting Diode*) , dan untuk data sensor *Grove Finger Clip Heart Rate* ketika nilai masih kurang dari 1 dan sama dengan 1 maka Arduino Pro Mini belum mengirimkan data ke ESP8266, ketika data sudah lebih dari 1 nilainya, maka Arduino Pro Mini akan mengirim data protokol/format bentuk data yang pengirimannya sudah ditentukan. Pengiriman data dilakukan langsung secara bersamaan satu protokol/format namun dalam satu siklus pengiriman dikirimkan

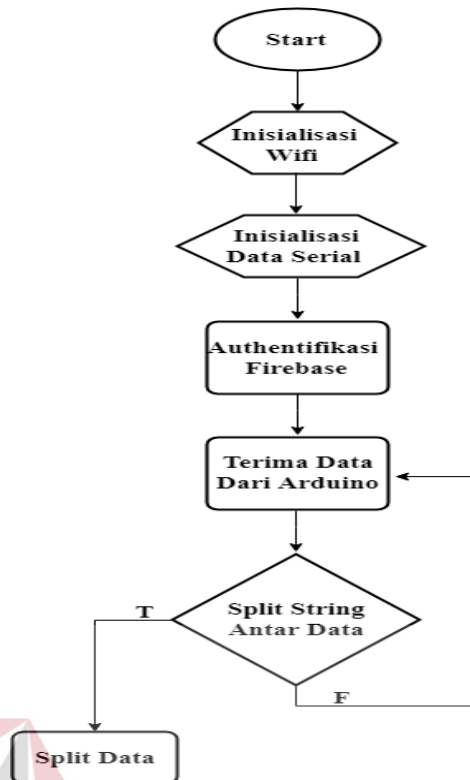
per satu *character*. Kemudian sampai di ESP8266 akan di split data yang kemudian menjadi format untuk dikirimkan ke *Database Firebase*.



Gambar 3.11 Flowchart Proses Kirim Data Protokol

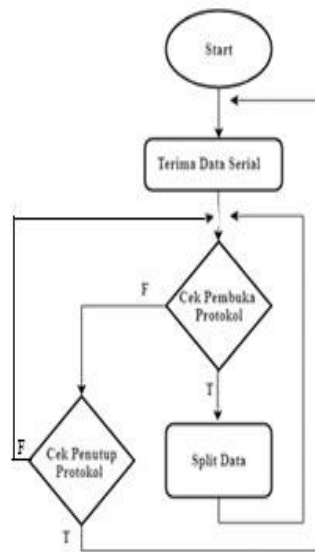
### 3.2.2 Flowchart ESP8266

*Flowchart ESP8266* pada gambar 3.12 dijelaskan bahwa mula – mula data dikirimkan dari Arduino Pro Mini ke ESP8266. Kemudian data tersebut di *split*/di pecah oleh ESP8266 untuk kemudian dikirimkan ke Database Firebase. Durasi pengiriman ke firebase ditentukan oleh fungsi algoritma efisiensi yaitu *lowPower.powerDown*, *lowPower.idle* dan *delay*, karena dalam fungsi bisa ditentukan untuk waktu pengiriman data nya per berapa detik sekali.



Gambar 3.12 Flowchart Sistem Keseluruhan ESP8266

Jadi sebelum mengoperasikan ESP8266 untuk mentransmisikan data yaitu dengan menentukan *SSID* dan *PASSWORD* yang digunakan ESP8266 sebagai sarana mengirim data, kemudian tentukan *HOST* dan *AUTHENTICATION* pada firebase. Lalu ESP8266 akan membaca data karakter dari *port* serial dan akan melakukan pemecahan data untuk dikirim ke firebase, kemudian akan dilakukan pemecahan data lagi antara data pertama yang terkirim dengan data selanjutnya, tujuannya untuk agar tiap data yang dikirim ke firebase tidak akan berantakan dan tetap berurutan dengan baik. Lalu program akan terus mengulang selama data yang dikirim dari *Mikrokontroller Arduino Pro Mini* masih berjalan.



Gambar 3.13 Flowchart Kirim Format Data Ke Database Firebase

Format data serial yang diperoleh dari *Arduino Pro Mini Pro Mini* masih belum beraturan maka dari itu perlu dilakukan proses *split* data untuk memecah data menjadi beberapa bagian yang setelah itu akan dikirimkan ke *Database Firebase* melalui jaringan koneksi *wifi*, protokol disini maksudnya adalah format penulisan data yang akan dikirimkan ke *Database Firebase*.

### 3.2.3 Perancangan Software

#### A. Program Arduino Pro Mini

##### 1. LCD OLED (Organic Light Emiting Diode) (*Organic Light Emiting Diode*)

Untuk mendeklarasikan Komponen *LCD OLED (Organic Light Emiting Diode)* pada *board Arduino Pro Mini* maka digunakan *syntax* seperti dibawah ini:

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
```



Setelah pendeklarasian *LCD OLED (Organic Light Emitting Diode)* maka langkah selanjutnya adalah melakukan pengiriman data string ke tampilan

### *LCD OLED (Organic Light Emitting Diode)*

```
display.clearDisplay();
display.setTextSize(2); //tinggi=7px
display.setTextColor(WHITE);
display.setCursor(38,10);
display.print(c);
display.setTextSize(1);
display.setCursor(90,2);
display.print(voltageString);
display.display();
```

## 2. RTC (*Real Time Clock*)

Untuk mendeklarasikan Komponen *RTC* pada board Arduino Pro Mini maka digunakan *syntax* seperti dibawah ini:

```
RTC_DS1307 rtc;
if(!rtc.begin())
{
    Serial.println("RTC Not Working");
    while(true);
}
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
```

Setelah melakukan pendeklarasian *RTC* maka langkah selanjutnya adalah melakukan pengambilan data berupa hari,tanggal,bulan,tahun dan kemudian diubah dalam bentuk *string*.

```
DateTime now = rtc.now();
Serial.print("[");
Serial.print(now.day(), DEC);
Serial.print("-");
Serial.print(now.month(), DEC);
Serial.print("-");
Serial.print(now.year(), DEC);
Serial.print(";");
Serial.print(now.hour(), DEC);
Serial.print(":");
Serial.print(now.minute(), DEC);
Serial.print(":");
Serial.print(now.second(), DEC);
```

## 3. Sensor *Heart Rate*.

Untuk mendeklarasikan Komponen *Sensor Grove Finger Clip Heart Rate* pada board Arduino Pro Mini maka digunakan *syntax* seperti dibawah ini:

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

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

#### 4. Sensor Voltage

Untuk mendeklarasikan Komponen *Sensor Grove Finger Clip Heart Rate* pada board Arduino Pro Mini maka digunakan *syntax* seperti dibawah ini:

```
analogReference (INTERNAL);
Serial.begin(9600);
float readVoltage()
{
  float raw = analogRead(A0);
  float x = raw/1023;
  float voltage = x*AREF*4;
  return voltage;
}
```

Setelah itu yaitu proses konversi data dari analog reference ke dalam bentuk persen(%), *syntax* nya seperti dibawah ini:

```
voltage = readVoltage();
prsn_voltage = (voltage/3.11) * 100.0;
voltageString = String(prsn_voltage,0) + "%";
```

#### 5. Protocol

Untuk pengiriman data ESP8266 maka menggunakan format seperti dibawah ini [tanggal-bulan-tahun;jam:menit:detik;heartrate;kapasitas baterai]:

```
Serial.print("[");
Serial.print(now.day(), DEC);
Serial.print("-");
Serial.print(now.month(), DEC);
Serial.print("-");
Serial.print(now.year(), DEC);
Serial.print(";");
```

```

Serial.print(now.hour(), DEC);
Serial.print(":");
Serial.print(now.minute(), DEC);
Serial.print(":");
Serial.print(now.second(), DEC);
Serial.print(";");
Serial.print(c, DEC);
Serial.print(";");
Serial.print(voltageString);
Serial.print(";");
Serial.println("]");

```

## B. Program ESP8266

### 1. *Firestore*

Untuk mengatur wifi dan meng - autentifikasi firebase, maka menggunakan syntax dibawah ini:

```

#define WIFI_SSID "Hidden" ///Nama Wifi
#define WIFI_PASSWORD "akul2345" ///Password Wifi
#define FIREBASE_HOST "tugasakhir1-1838b.firebaseio.com"
#define FIREBASE_AUTH "FFtgftAKiiDQqW2p1VxsN2A9qZFwD5uqAOD8j6"

```

Setelah itu koneksi ke SSID, syntax nya sebagai berikut:

```

Serial.begin(9600);

//untuk koneksi ssid
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());

```

### 2. *Split Data Serial*

```

do {
    //untuk pemecah split string antar data
    char c = Serial.read();
    if (c=='[') {
        statusrun=1;
    }
} while (1);

```

```

    } else if (c==']') {
        statusrun=0;
    }

    if(statusrun == 1) {
        StaticJsonBuffer<200> jsonBuffer;//menyiapkan 200 array
        karakter sbllm di kirim
        JsonObject& root = jsonBuffer.createObject();
        //di copy
        if(c=='%'||c=='.'||c==':'||c=='-'
        ||c=='0'||c=='1'||c=='2'||c=='3'||c=='4'||c=='5'||c==
        '6'||c=='7'||c=='8'||c=='9') {
            dataSerial += c;
        }
        else if (c==';') { // pemisah antar data
            if(string_data == 0) // pemisah data pertama
            {
                tanggal = dataSerial;
                dataSerial = "";
                string_data++;
            }
            else if(string_data == 1)
            {
                jam = dataSerial;
                dataSerial = "";
                string_data++;
            }
            else if(string_data == 2)
            {
                heart_rate = dataSerial;
                dataSerial = "";
                string_data++;
            }
            else if(string_data == 3)
            {
                baterai = dataSerial;
                dataSerial = "";
                string_data = 0;
            }
        }
    }
}

```

### 3.3.4 Fungsi LowPower.powerDown

Fungsi LowPower.PowerDown merupakan fungsi efisiensi yang ada pada internal Arduino Pro Mini, fungsi ini digunakan untuk memaksimalkan hemat daya pada Arduino Pro Mini karena dalam fungsi ini ada beberapa fungsi didalam *board Arduino Pro Mini* yang secara sistem akan dimatikan, sehingga akan mengurangi penggunaan daya pada board Arduino Pro Mini itu sendiri. Berikut adalah definisi dari fungsi *LowPower.PowerDown*.

*LowPower.PowerDown* adalah menempatkan mikrokontroller pada kondisi hampir mati atau sedang dalam daya terendahnya ketika tidak melakukan suatu perintah kirim data atau terima data, entah itu data dari sensor maupun inputan lainnya juga. Sehingga jika tidak dalam terima dan kirim, daya yang digunakan rendah. Namun ketika sedang melakukan pengiriman, maka daya yang digunakan normal namun dalam pengiriman data dan penerimaan data tentunya ada delay yang diterapkan dalam Arduino Pro Mini, ketika sedang mengalami delay, mikrokontroller jika menggunakan fungsi ini akan tertidur sesaat dan ketika tidur daya yang dibutuhkan sangat sedikit atau kurang dari daya saat kondisi normal, sehingga efisiensi bisa didapatkan dengan fungsi ini.

Berikut adalah isi dari fungsi *LowPower.PowerDown* :

*Name: LowPower.PowerDown*

*Description: Putting microcontroller into power down state. This is the lowest current consumption state. Use this together with external pin interrupt to wake up through external event triggering (example: RTC clockout pin, SD card detect pin).*

*Argument Description*

1. *Period : Duration of low power mode. Use SLEEP\_FOREVER to use other wake up resource:*

*SLEEP\_15MS - 15 ms sleep*  
*SLEEP\_30MS - 30 ms sleep*  
*SLEEP\_60MS - 60 ms sleep*  
*SLEEP\_120MS - 120 ms sleep*  
*SLEEP\_250MS - 250 ms sleep*  
*SLEEP\_500MS - 500 ms sleep*  
*SLEEP\_1S - 1 s sleep*  
*SLEEP\_2S - 2 s sleep*  
*SLEEP\_4S - 4 s sleep*  
*SLEEP\_8S - 8 s sleep*  
*SLEEP\_FOREVER - Sleep without waking up through WDT*

2. *ADC : ADC module disable control. Turning off the ADC module is basically removing the purpose of this low power mode.*

*ADC\_OFF - Turn off ADC module*

*ADC\_ON - Leave ADC module in its default state*

3. *BOD : Brown Out Detector (BOD) module disable control:*

*BOD\_OFF - Turn off BOD module*

*BOD\_ON - Leave BOD module in its default state*

Jadi dalam fungsi *LowPower.PowerDown* periode waktunya bisa ditentukan seperti pada bagian *Period* diatas. Kemudian dengan mematikan *ADC* ketika tidak digunakan ternyata juga mempunyai pengaruh besar terhadap konsumsi daya pada Board Arduino Pro Mini, seperti dijelaskan pada bagian *ADC* diatas dan yang terakhir adalah mematikan *BOD* seperti yang sudah dijelaskan diatas.

### 3.3.4 Fungsi *LowPower.Idle*

Fungsi *LowPower.Idle* merupakan fungsi efisiensi yang ada pada internal Arduino Pro Mini, fungsi ini digunakan untuk hemat daya pada Arduino Pro Mini karena dalam fungsi ini ada beberapa fungsi didalam *board Arduino Pro Mini* yang secara sistem akan dimatikan, sehingga akan mengurangi penggunaan daya pada board Arduino Pro Mini itu sendiri. Berikut adalah definisi dari fungsi *LowPower.Idle*.

*LowPower.Idle* Adalah menempatkan *MCU* dalam mode *IDLE/Siaga*. Mode ini memungkinkan pengoptimalan daya dengan waktu bangun tercepat. CPU akan terhenti sesaat untuk lebih mengurangi konsumsi daya, dan dapat secara manual dilakukan menonaktifkan *clocking* modul dan sumber *clock*.

1. *period* Duration of low power mode. Use *SLEEP\_FOREVER* to use other wake

*up resource:*

- (a) *SLEEP\_15MS* - 15 ms sleep
- (b) *SLEEP\_30MS* - 30 ms sleep
- (c) *SLEEP\_60MS* - 60 ms sleep
- (d) *SLEEP\_120MS* - 120 ms sleep
- (e) *SLEEP\_250MS* - 250 ms sleep
- (f) *SLEEP\_500MS* - 500 ms sleep
- (g) *SLEEP\_1S* - 1 s sleep
- (h) *SLEEP\_2S* - 2 s sleep
- (i) *SLEEP\_4S* - 4 s sleep
- (j) *SLEEP\_8S* - 8 s sleep
- (k) *SLEEP\_FOREVER* - Sleep without waking up through WDT

## **2. *adc* ADC module disable control:**

- (a) *ADC\_OFF* - Turn off ADC module
- (b) *ADC\_ON* - Leave ADC module in its default state

## **3. *timer4* Timer 4 module disable control:**

- (a) *TIMER4\_OFF* - Turn off Timer 4 module
- (b) *TIMER4\_ON* - Leave Timer 4 module in its default state

## **4. *timer3* Timer 3 module disable control:**

- (a) *TIMER3\_OFF* - Turn off Timer 3 module
- (b) *TIMER3\_ON* - Leave Timer 3 module in its default state

## **5. *timer1* Timer 1 module disable control:**

- (a) *TIMER1\_OFF* - Turn off Timer 1 module
- (b) *TIMER1\_ON* - Leave Timer 1 module in its default state

## **6. *timer0* Timer 0 module disable control:**

- (a) *TIMER0\_OFF* - Turn off Timer 0 module
- (b) *TIMER0\_ON* - Leave Timer 0 module in its default state

## **7. *spi* SPI module disable control:**

- (a) *SPI\_OFF* - Turn off SPI module
- (b) *SPI\_ON* - Leave SPI module in its default state

## **8. *usart1* USART1 module disable control:**

- (a) *USART1\_OFF* - Turn off USART1 module
- (b) *USART1\_ON* - Leave USART1 module in its default state

## **9. *twi* TWI module disable control:**

- (a) *TWI\_OFF* - Turn off TWI module
- (b) *TWI\_ON* - Leave TWI module in its default state

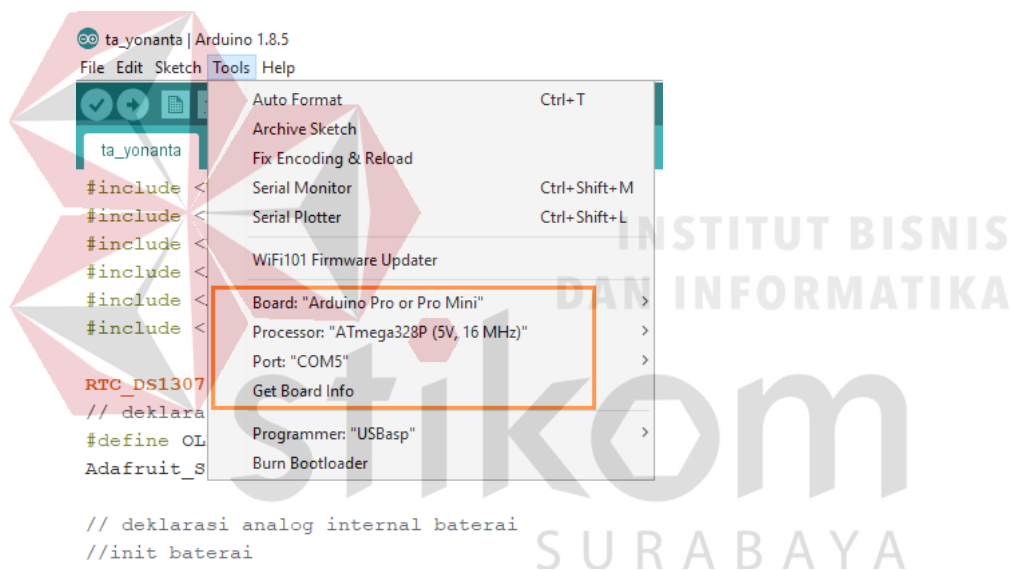
## **10. *usb* USB module disable control:**

- (a) *USB\_OFF* - Turn off USB module
- (b) *USB\_ON* - Leave USB module in its default state

### 3.3.5 Prosedur konfigurasi Arduino IDE

#### 1. Arduino Pro Mini Pro Mini

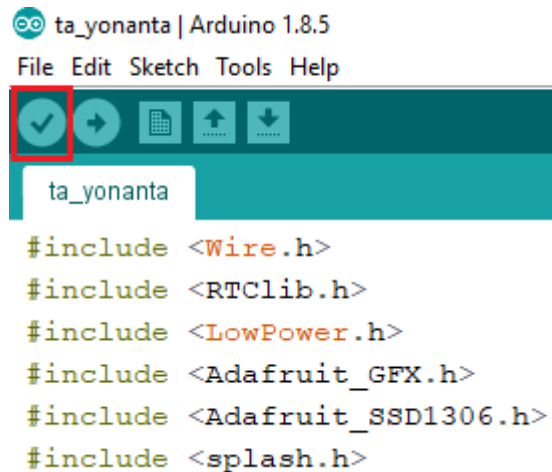
- a. Pertama kali yang harus dilakukan adalah klik menu *tools* pada Arduino Pro Mini IDE kemudian menentukan Arduino Pro Mini jenis apa yang dipakai, untuk Tugas Akhir kali ini menggunakan Arduino jenis Arduino Pro Mini 5V 16MHz untuk pembuatan Tugas Akhir, kemudian tentukan *port* yang digunakan untuk *upload* program *Arduino Pro Mini* seperti pada Gambar 3.14.



Gambar 3.14 Tentukan *Port*

- b. Setelah selesai menkonfigurasi, kemudian untuk memverifikasi program klik pada ikon centang pada pojok kiri atas, berfungsi untuk mengecek program sebelum diupload ke *Arduino Pro Mini*.



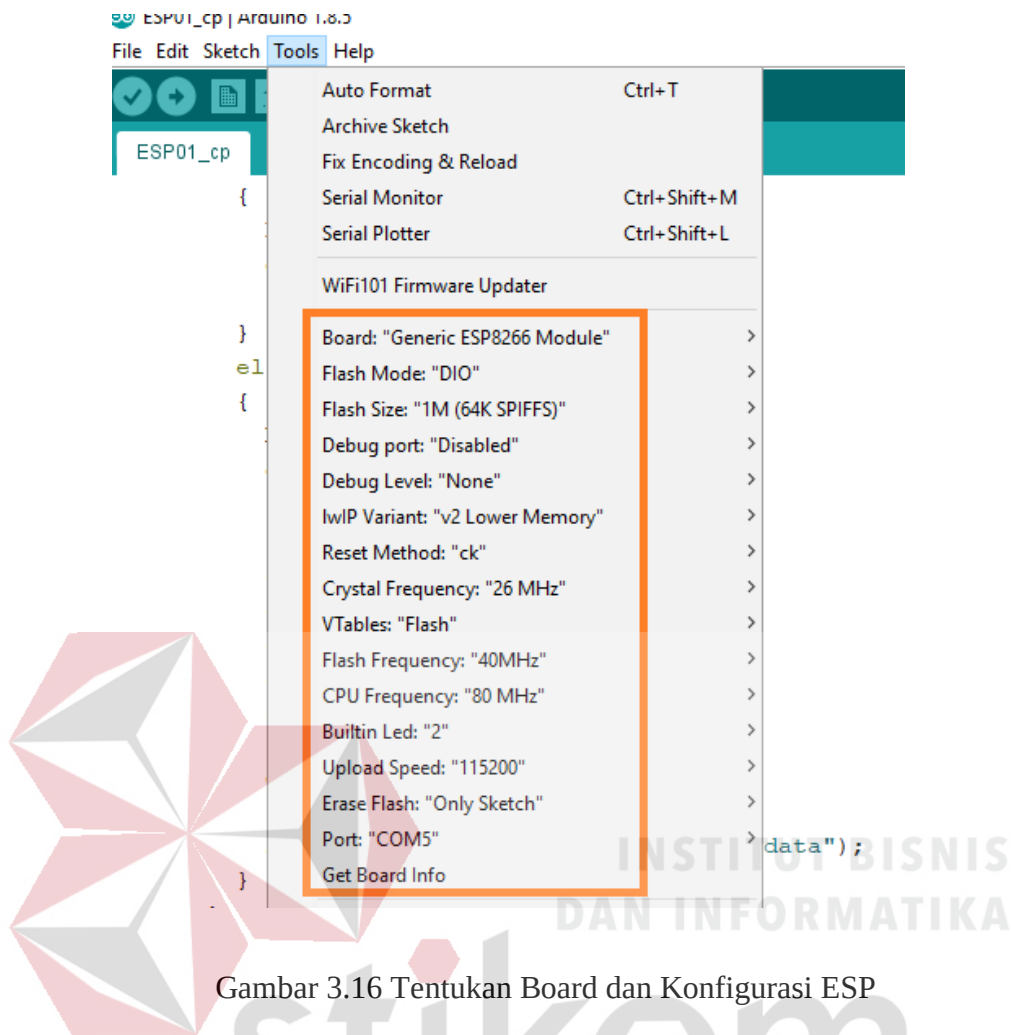


Gambar 3.15 *Verify* Program Arduino Pro Mini

- c. Kemudian jika tidak terdapat kesalahan pada program maka klik ikon panah/upload, fungsinya adalah untuk mengupload program yang sudah di tulis ke Arduino Pro Mini.

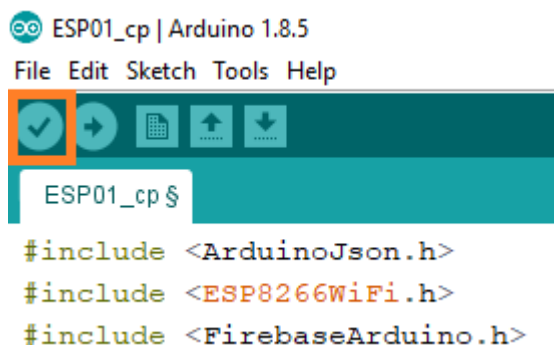
## 2. ESP8266

- a. Pertama kali yang harus dilakukan adalah klik menu tools pada Arduino IDE kemudian menentukan ESP apa yang dipakai, dalam tugas akhir ini yang digunakan adalah ESP8266 kemudian atur semuanya seperti pada Gambar 3.16.



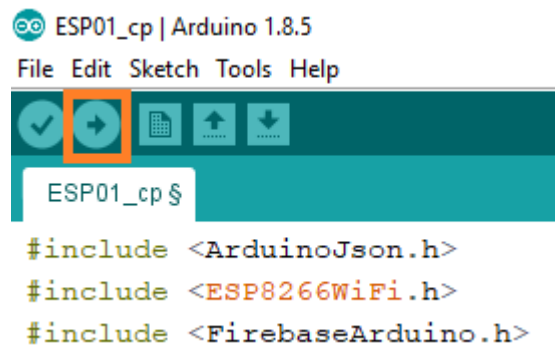
Gambar 3.16 Tentukan Board dan Konfigurasi ESP

- b. Jika sudah, maka klik ikon centang pada bagian kiri atas untuk memverifikasi apakah ada program yang salah atau tidak sebelum di upload ke Arduino Pro Mini, seperti pada Gambar 3.17.



Gambar 3.17 Verify Program ESP8266

C. Kemudian jika tidak terdapat kesalahan pada program maka klik ikon panah/upload, fungsinya adalah untuk meng-upload program yang sudah ditulis ke Arduino Pro Mini, seperti pada Gambar 3.18.

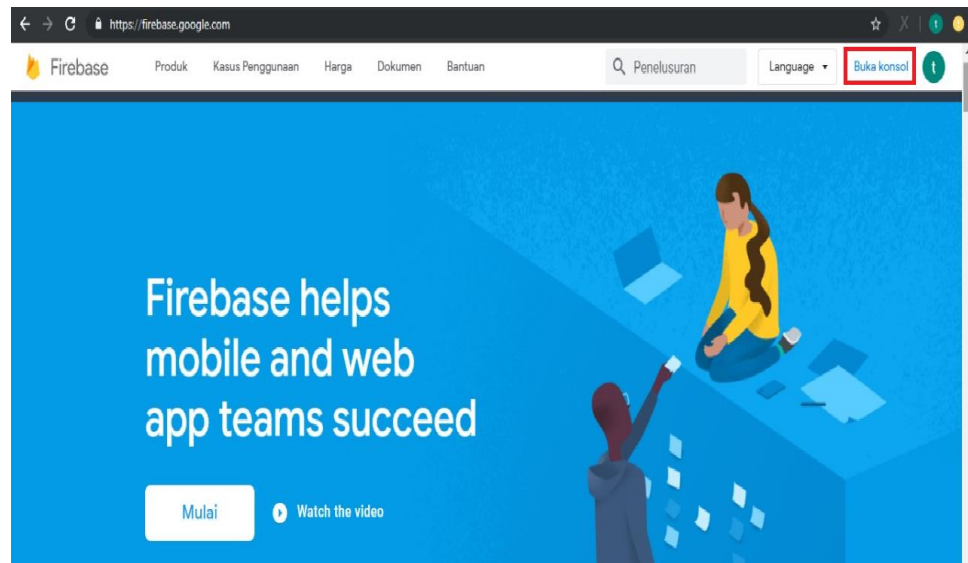


Gambar 3.18 Upload Program

### 3.3.6 Prosedur membuat *Database Firebase*

Database pada proyek ini digunakan untuk menyimpan data dari alat pemantau data heart rate. Maka disini akan dijelaskan pembuatan Database Firebase langkah demi langkah.

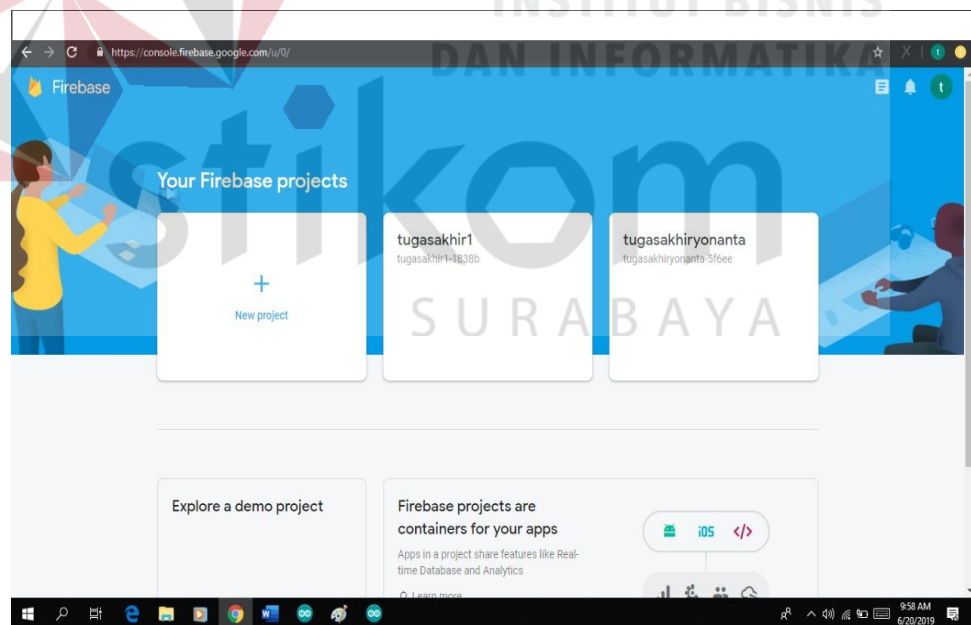
1. Pastikan *chrome* telah terhubung dengan akun *google*, karen untuk pembuatan Database Firebase harus menggunakan akun *google* yang tersambung di *chrome*.
2. Setelah itu buka [www.firebase.com](http://www.firebase.com), maka akan terdapat tampilan seperti Gambar 3.19.



Gambar 3.19 Buka Konsol Firebase

Setelah itu pilih menu buka konsol untuk membuat database.

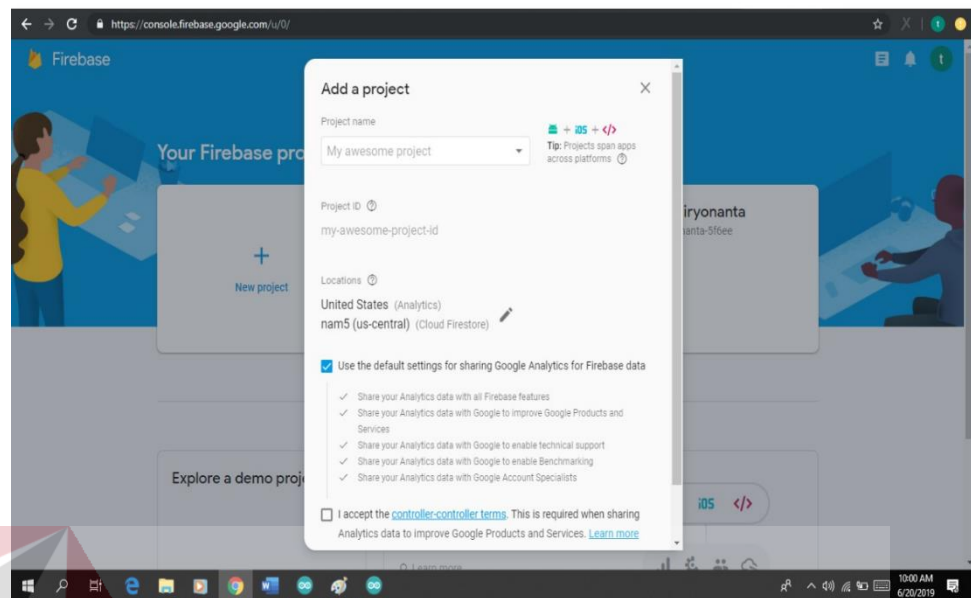
3. Setelah itu akan ada tampilan seperti Gambar 3.20.



Gambar 3.20 Tampilan awal database

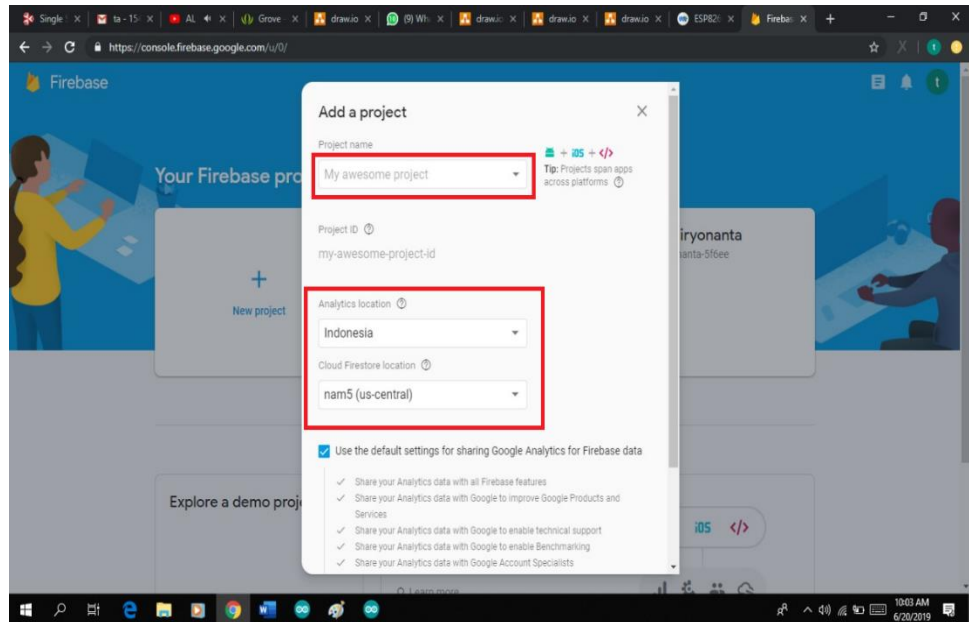
Setelah terdapat tampilan seperti gambar, maka pilih *New Project* untuk membuat *database* nya.

4. Setelah itu akan terdapat tampilan seperti Gambar 3.21.



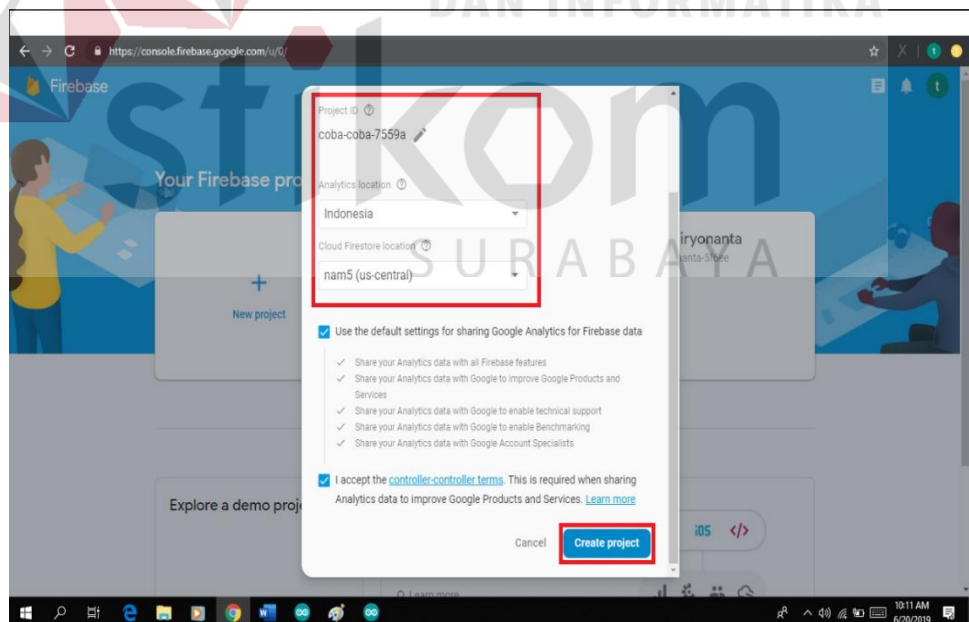
Gambar 3.21 Add Project

Setelah terdapat tampilan seperti Gambar 3.21, maka langkah selanjutnya adalah menamai proyek *database*, dan mengganti lokasi menjadi “Indonesia”, seperti pada Gambar 3.22.



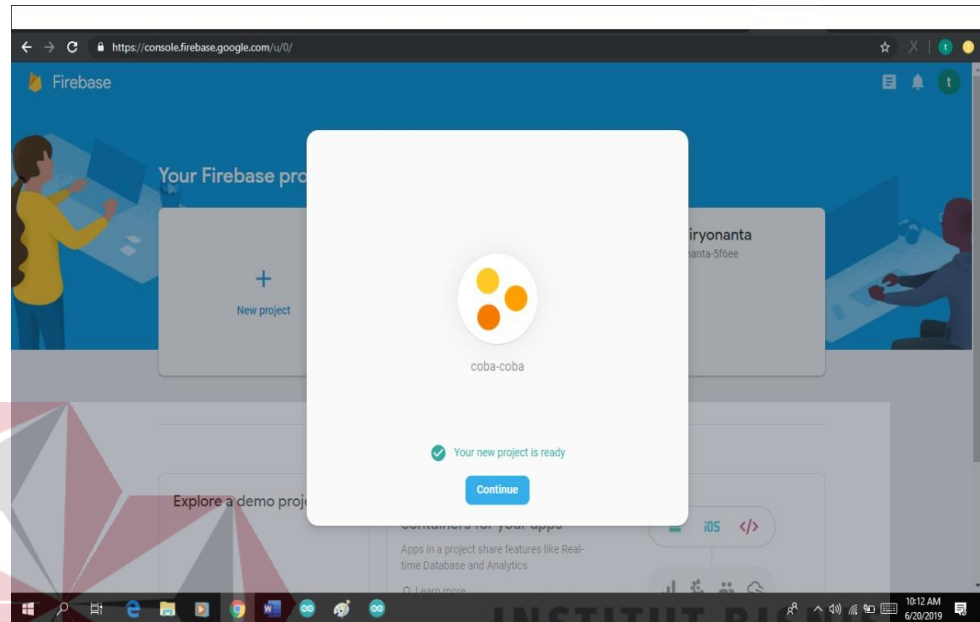
Gambar 3.22 Tentukan Lokasi

5. Setelah sudah memberi nama proyek dan mengganti lokasi, maka akan terdapat tampilan seperti Gambar 3.23.



Gambar 3.23 Tampilan nama atabase

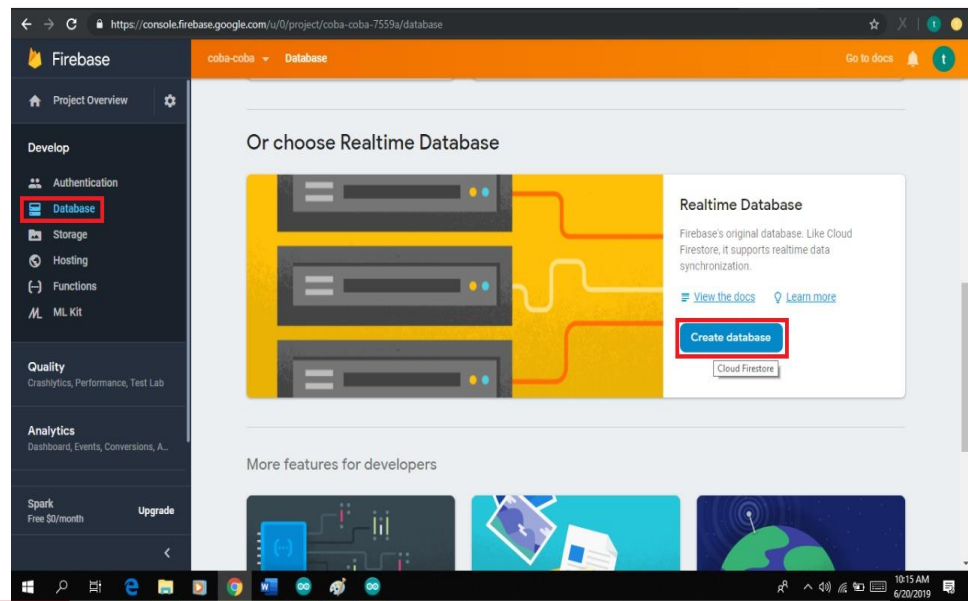
Ketika tampilan sudah seperti gambar, maka langkah selanjutnya pilih *Create Project*. Lalu tunggu proses pembuatan data, hingga muncul tulisan *Your Project Is Ready* seperti Gambar 3.24.



Gambar 3.24 Loading pembuatan database

Setelah selesai klik *continue* untuk lanjut ke proses selanjutnya.

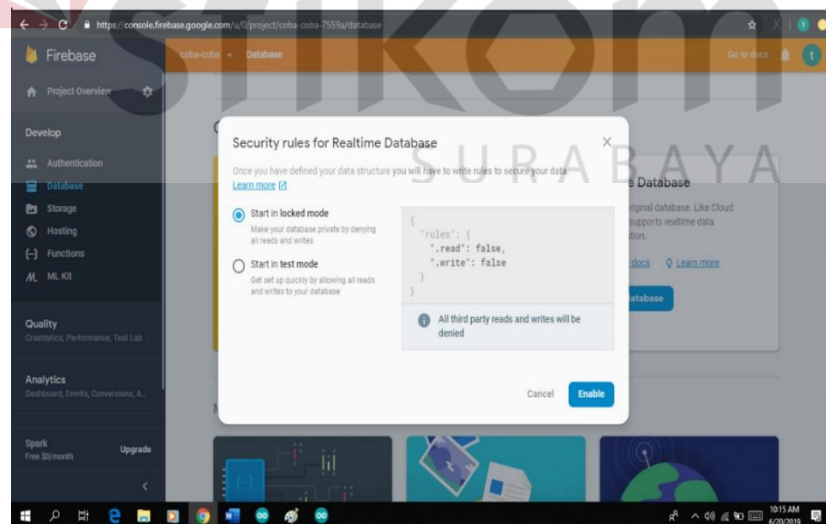
6. Setelah itu pilih menu *Database > Reltime Database* seperti Gambar 3.25.



Gambar 3.25 Database Realtime

Setelah terdapat tampilan seperti gambar, maka pilih *Create Database*

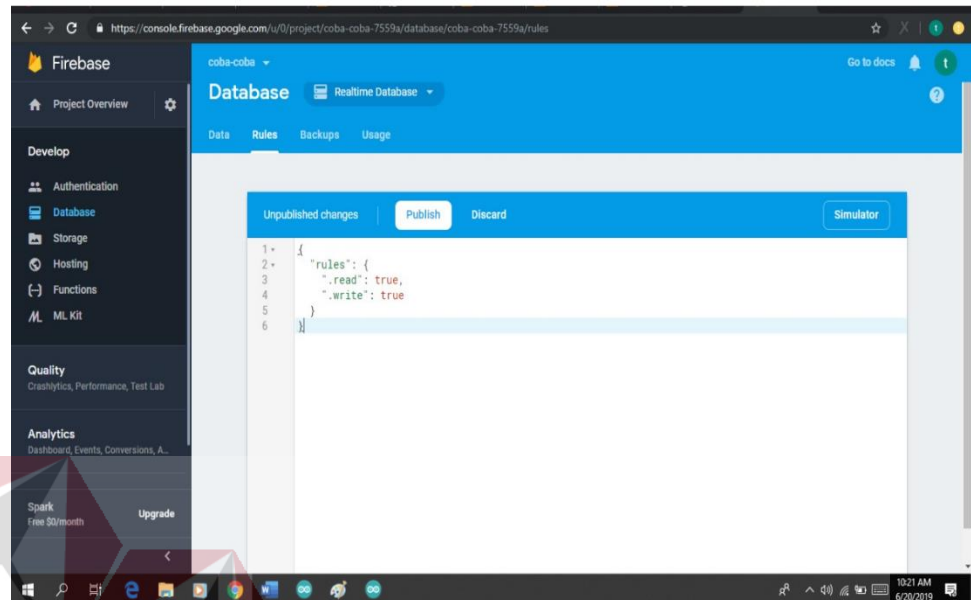
7. lalu akan muncul tampilan seperti Gambar 3.26 dan langkah selanjutnya adalah memilih *Enable*.



Gambar 3.26 Security Realtime Database

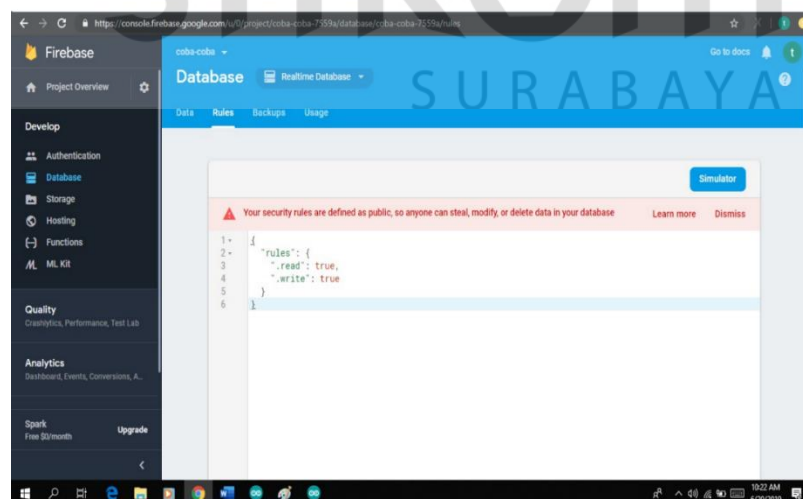


8. Setelah itu pilih rules dan ganti false menjadi true, agar database bisa berfungsi baca dan tulis, seperti pada Gambar 3.27.



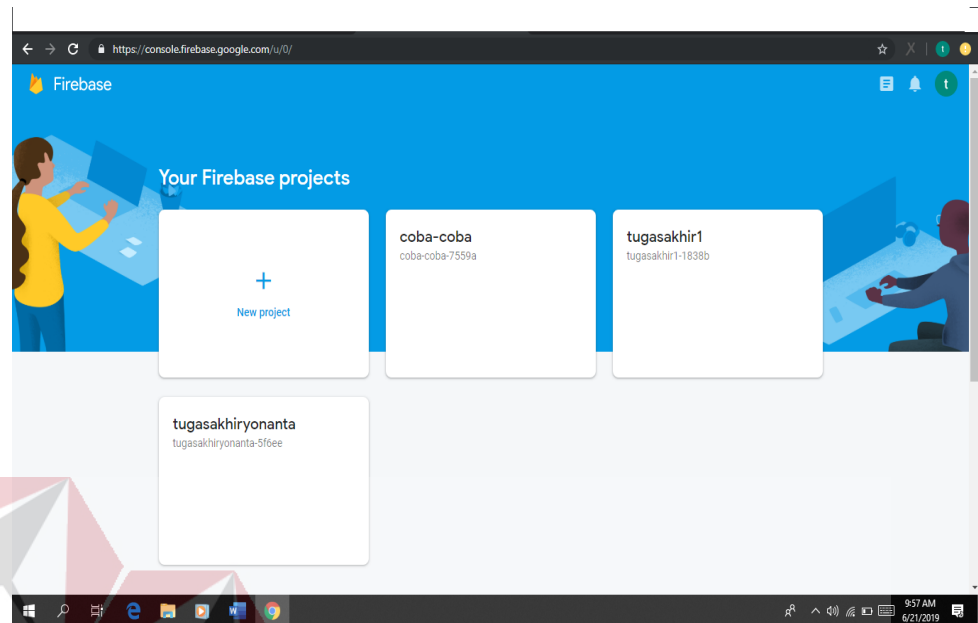
Gambar 3.27 Fungsi Baca Dan Tulis Database

9. Setelah itu pilih *publish* dan akan muncul tampilan seperti Gambar 3.28 dibawah ini:



Gambar 3.28 *publish database*

Maka pembuatan database telah selesai dan sudah dapat dilihat pada menu konsol seperti pada Gambar 3.29.



Gambar 3.29 Menu Konsol

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Hasil pembahasan dan pengujian yang dilakukan oleh penulis akan dijelaskan pada bab ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah diajukan dan dikerjakan. Tahapan pengujian yang akan dilakukan meliputi pengujian Arduino Pro Mini Pro Mini, Sensor grove finger clip, ESP8266, RTC (Real-Time Clock), LCD OLED (ORGANIC LIGHT EMITTING DIODE)

#### **4.1 Pengujian Arduino Pro Mini Pro Mini**

##### **4.1.1 Tujuan Pengujian Arduino Pro Mini Pro Mini**

Pengujian *Arduino Pro Mini Pro Mini* bertujuan untuk mengetahui apakah *Arduino Pro Mini Pro Mini* dapat berjalan dengan baik serta dapat mengeksekusi program yang menggunakan bahasa pemrograman *C* pada *software Arduino Pro Mini IDE*.

##### **4.1.2 Peralatan Pengujian Arduino Pro Mini Pro Mini**

1. *Arduino Pro Mini Pro Mini 5v*
2. *Software Arduino IDE*
3. *Kabel USB*
4. *Laptop*

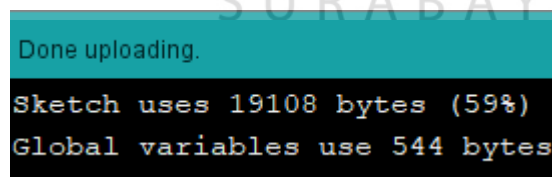
##### **4.1.3 Prosedur Pengujian Arduino Pro Mini Pro Mini**

1. Menyambungkan Kabel USB ke Port Uploading yang ada pada *Arduino Pro Mini Pro Mini*.

2. Sambungkan Kabel USB ke Port USB pada komputer/*laptop*
3. Pilih jenis Arduino Pro Mini yang digunakan.
4. Pilih *port COM* pada *software Arduino Pro Mini IDE* untuk jalur port yang digunakan.
5. Setelah itu mulai tulis program pada lembar kerja yang telah disediakan oleh *software*.
6. Verifikasi program untuk melihat ada yang salah atau tidak pada program, setelah tidak ada, maka bisa dilanjutkan *upload*/menjalankan program.
7. Setelah selesai maka langkah selanjutnya adalah Memberikan nama sesuai keinginan dengan ekstensi \*.ino pada file tersebut.
8. Tentukan *directory* penyimpanan pada file tersebut.

#### 4.1.4 Hasil Pengujian Arduino Pro Mini Pro Mini

Apabila *program* tidak ditemukan *error*, maka akan muncul tulisan Done Uploading pada message dibagian bawah seperti pada gambar 4.1 , itu menandakan bahwa program sudah di upload pada *board Arduino Pro Mini pro mini*.



```
Done uploading.  
Sketch uses 19108 bytes (59%)  
Global variables use 544 bytes
```

Gambar 4.1 Program Berhasil Diupload

## 4.2 Pengujian Sensor Grove Finger Clip Heart Rate

### 4.2.1 Tujuan Pengujian Sensor Grove Finger Clip Heart Rate

Pengujian *Sensor Grove Finger Clip Heart Rate* bertujuan untuk mengetahui apakah *Sensor Grove Finger Clip* dapat berjalan dengan baik pada *Arduino Pro Mini Pro Mini* serta dapat mengukur detak jantung dengan baik..

### 4.2.2 Peralatan Pengujian Sensor Grove Finger Clip Heart Rate

1. *Arduino Pro Mini Pro Mini*
2. *Sensor Grove Finger Clip*
3. *Software Arduino Pro Mini IDE*
4. Kabel USB
5. Laptop
6. Kabel Jumper

### 4.2.3 Prosedur Pengujian Sensor Grove Finger Clip Heart Rate

1. Menyambungkan Kabel *USB* ke *Port Uploading* yang ada pada *Arduino Pro Mini Pro Mini*.
2. Menyambungkan Kabel *USB* ke *port USB* yang ada pada Laptop.
3. Menyambungkan *port i2c* yang ada pada sensor *Grove Finger Clip Heart Rate* ke *port i2c* pada *Arduino Pro Mini pro mini*.
4. Membuat Program Sederhana untuk sensor. Untuk pembacaan sensor.

```
#include <Wire.h>
void setup() {
  Serial.begin(9600);
  Serial.println("heart rate sensor:");
  Wire.begin();
}
void loop() {
  Wire.requestFrom(0xA0 >> 1,1);
  while(Wire.available()) {
```

```

        unsigned char c = Wire.read();
        Serial.println(c, DEC);
    }
    delay(500);
}

```

#### 4.2.4 Hasil Pengujian Grove Finger Clip Heart Rate Sensor

Apabila *program* tidak ditemukan *error*, maka akan muncul *done uploading* baru. Contoh hasil *running program* dapat dilihat pada *serial monitor* ditunjukkan pada Gambar 4.3 dibawah ini:



Gambar 4.2 Output Serial Sensor Grove Finger Clip Heart Rate

### 4.3 Pengujian RTC (Real-Time Clock)

#### 4.3.1 Tujuan Pengujian RTC (Real-Time Clock)

Pada pengujian komponen *RTC (Real-Time Clock)* yaitu untuk mengetahui apakah bisa berjalan dengan baik ketika disambungkan dengan Arduino Pro Mini Pro Mini. Dan melihat keluaran waktu nya.

#### 4.3.2 Peralatan Pengujian RTC (RealTime-Clock)

1. *Arduino Pro Mini Pro Mini*
2. *RTC (Real-Time Clock)*

3. *Software Arduino Pro Mini IDE*
4. *Kabel USB*
5. *Laptop*
6. *Kabel Jumper*

#### 4.3.3 Prosedur Pengujian RTC (*RealTime-Clock*)

1. Menyambungkan kabel USB Ke port Uploading yang ada pada Arduino Pro Mini Pro Mini
2. Menyambungkan Kabel USB ke Port USB yang ada pada Komputer/*Laptop*
3. Menyambungkan Port RX TX yang ada pada RTC (*RealTime-Clock*) ke Port RX TX yang ada pada Arduino Pro Mini Pro Mini, namun pemasangannya adalah dibalik, RX dari RTC (*RealTime-Clock*) menyambung TX yang ada pada Arduino Pro Mini Pro Mini, dan TX dari RTC (*RealTime-Clock*) menyambung ke RX yang ada pada Arduino Pro Mini Pro Mini.
4. Menulis Program sederhana pada software Arduino Pro Mini IDE, seperti pada program dibawah ini:

```
#include <Wire.h>
#include <RTCLib.h>
#include <LowPower.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <splash.h>

RTC_DS1307 rtc;
if(!rtc.begin())
{
    Serial.println("RTC Not Working");
    while(true);
}
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
Serial.print("[");
    Serial.print(now.day(), DEC);
    Serial.print("-");
    Serial.print(now.month(), DEC);
    Serial.print("-");
```

```

Serial.print(now.year(), DEC);
Serial.print(";");
Serial.print(now.hour(), DEC);
Serial.print(":");
Serial.print(now.minute(), DEC);
Serial.print(":");
Serial.print(now.second(), DEC);
Serial.print(";");
Serial.print(c, DEC);
Serial.print(";");

```

#### 4.3.4 Hasil Pengujian RTC (*RealTime-Clock*)

Apabila *program* tidak ditemukan *error*, maka akan muncul *done Uploading* baru serta menampilkan *program* yang diinginkan di *serial monitor*. Contoh hasil *running program* dapat ditunjukkan pada Gambar 4.4 dibawah ini:



Gambar 4.3 Output Data Serial RTC (*RealTime – Clock*)



## 4.4 Pengujian Komponen ESP8266

### 4.4.1 Tujuan Komponen ESP8266

Pada pengujian komponen ESP8266 bertujuan untuk mengetahui apakah ESP 8266 bisa bekerja dengan baik serta menerima dan mengirim data dengan baik.

### 4.4.2 Peralatan Pengujian ESP8266

1. *Arduino Pro Mini Pro Mini*
2. ESP 8266
3. *Software Arduino Pro Mini IDE*
4. Kabel USB
5. *Laptop*
6. *Kabel Jumper*

### 4.4.3 Prosedur Pengujian ESP8266

1. Menyambungkan Kabel USB ke port Uploading yang ada pada Arduino Pro Mini Pro Mini.
2. Menyambung Kabel USB Ke port USB yang ada pada Laptop/Komputer.
3. Menyambungkan ESP8266 dengan Arduino Pro Mini Pro Mini.
4. Tulis program di *software Arduino Pro Mini IDE*, seperti dibawah ini:

```
#include <Arduino Pro MiniJson.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino Pro Mini.h>

//set wifi
//init wifi
#define WIFI_SSID "Hidden" ///Nama Wifi
#define WIFI_PASSWORD "aku12345" ///Password Wifi

//Authentifikasi firebase
init firebase
#define FIREBASE_HOST "tugasakhir1-1838b.firebaseio.com"
```

```

#define FIREBASE_AUTH "FFtgftAKiIDQqW2p1VxsN2A9qZFwD5uqAO
D8j6NP"

//inisialisasi data
//init data serial
short int string_data = 0;
int data_ke = 0;
int statusrun = 0;
String tanggal, jam, heart_rate, baterai;
String dataSerial = "";

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  //untuk koneksi ssid
  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());
  //untuk koneksi firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED_BUILTIN, LOW);

  do {
    //untuk pemecah split string antar data
    char c = Serial.read();
    if (c=='[') {
      statusrun=1;
    } else if (c==']') {
      statusrun=0;
    }
  }

  if(statusrun == 1) {
    StaticJsonBuffer<200> jsonBuffer;//menyiapkan 200
    array karakter sbkm di kirim
    JsonObject root = jsonBuffer.createObject();
    //di copy
    if(c=='%'||c=='.'||c==':'||c=='-'
    ||c=='0'||c=='1'||c=='2'||c=='3'||c=='4'||c=='5'
    ||c=='6'||c=='7'||c=='8'||c=='9') {
      dataSerial += c;
    }
    else if (c==';') { // pemisah antar data
      if(string_data == 0) //pemisah data pertama
      {
        tanggal = dataSerial;
        dataSerial = "";
        string_data++;
      }
    }
  }
}

```

```

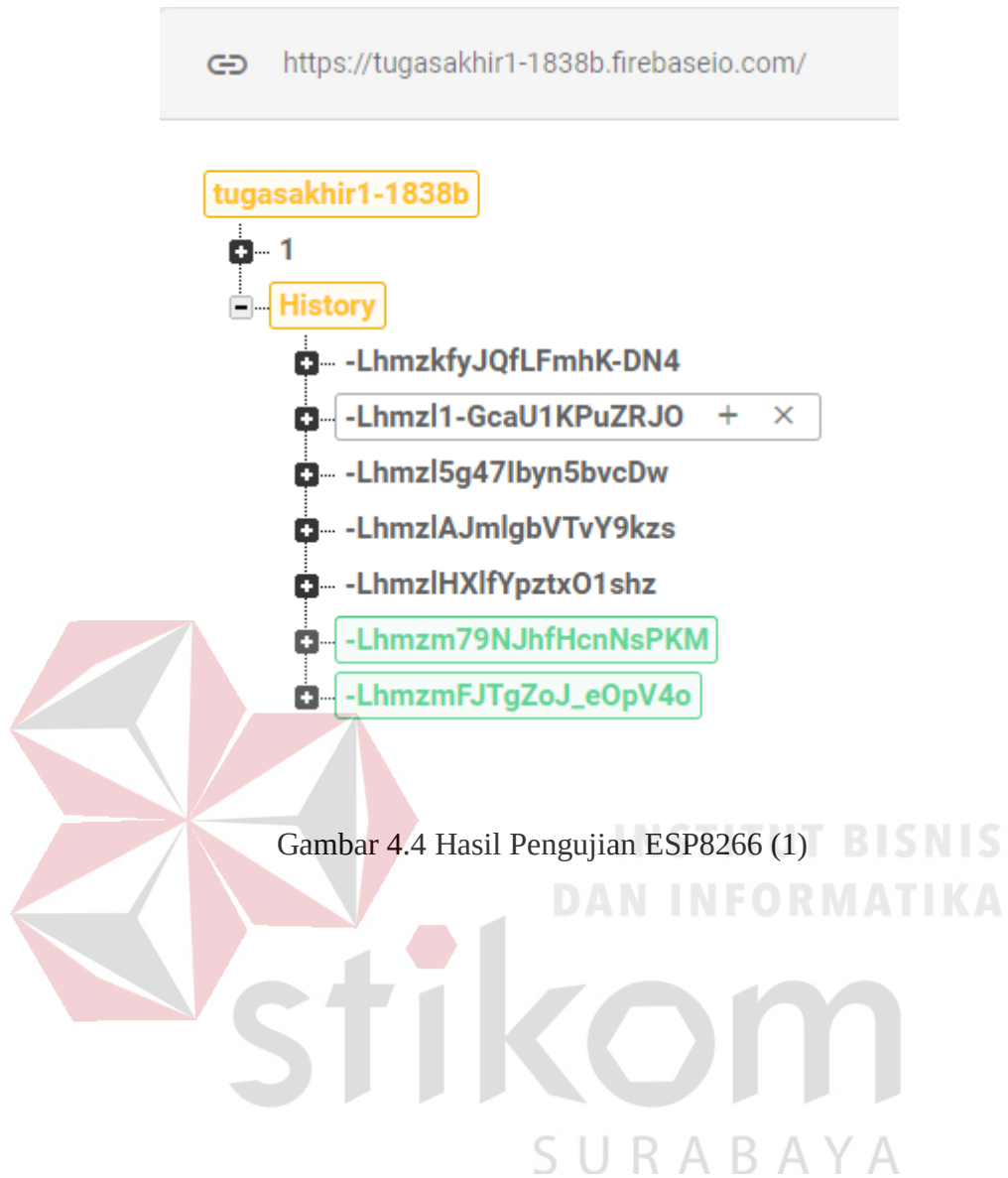
else if(string_data == 1)
{
    jam = dataSerial;
    dataSerial = "";
    string_data++;
}
else if(string_data == 2)
{
    heart_rate = dataSerial;
    dataSerial = "";
    string_data++;
}
else if(string_data == 3)
{
    baterai = dataSerial;
    dataSerial = "";
    string_data = 0;

    root["Data_ke"] = data_ke;
    root["Tanggal"] = tanggal;
    root["Jam"] = jam;
    root["Heart_Rate"] = heart_rate;
    root["Battery"] = baterai;
    Firebase.push("/History", root);
    data_ke++;
    //digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("Berhasil mengirim data");
}
}
} while(Serial.available()>0);
}

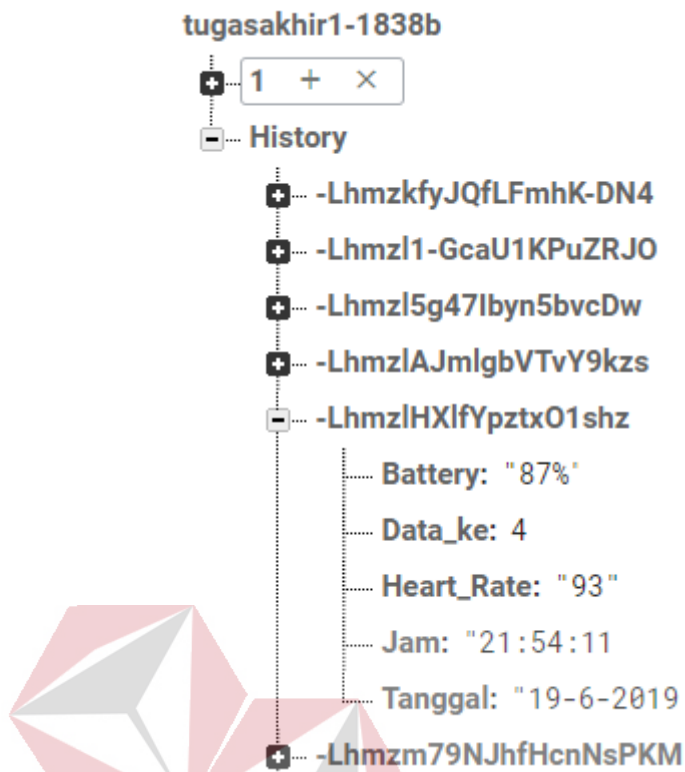
```

#### 4.4.4 Hasil Pengujian ESP8266

Dalam pengujian ini, dapat dilihat bahwa ESP8266 Sudah berhasil mengirimkan data dan memecah data dari mikrokontroler dan dikirim ke Database Firebase, seperti gambar dibawah ini:



Gambar 4.4 Hasil Pengujian ESP8266 (1)



Gambar 4.5 Hasil Pengujian ESP8266 (2)

#### 4.5 Pengujian Komponen *Display* LCD OLED (*Organic Light Emitting Diode*)

##### 4.5.1 Tujuan dari Pengujian Komponen LCD OLED (*Organic Light Emitting Diode*)

Tujuan pengujian LCD OLED (*Organic Light Emitting Diode*) adalah untuk mengetahui apakah komponen LCD OLED (*Organic Light Emitting Diode*) telah bekerja dengan baik dan menampilkan data dengan baik atau belum, maka dilakukan pengujian untuk menampilkan data dari Arduino Pro Mini Pro Mini.

#### 4.5.2 Peralatan Pengujian Komponen LCD OLED (*Organic Light Emitting Diode*)

1. *Arduino Pro Mini Pro Mini.*
2. *LCD OLED (Organic Light Emitting Diode) .*
3. *Software Arduino Pro Mini IDE.*
4. Kabel USB
5. *Laptop*
6. Kabel *Jumper*

#### 4.5.3 Prosedur pengujian Komponen LCD OLED (*Organic Light Emitting Diode*)

1. Menyambungkan Kabel USB ke port Uploading yang ada pada Arduino Pro Mini Pro Mini.
2. Menyambung Kabel USB Ke port USB yang ada pada Laptop/Komputer.
3. Menyambungkan LCD OLED (*Organic Light Emitting Diode*) dengan Arduino Pro Mini Pro Mini. Pin SDA dari LCD OLED (*Organic Light Emitting Diode*) menyambung pin A4 pada Arduino Pro Mini pro mini, dan pin SCL pada LCD OLED (*Organic Light Emitting Diode*) menyambung Pin A5 pada Arduino Pro Mini Pro mini.
4. Tulis program pada software Arduino Pro Mini IDE untuk menguji LCD OLED (*ORGANIC LIGHT EMITTING DIODE*) .

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay(); //hapus display LCD
Wire.begin();
Serial.print("[");
Serial.print(now.day(), DEC);
Serial.print("-");
Serial.print(now.month(), DEC);
Serial.print("-");
Serial.print(now.year(), DEC);
```

```

Serial.print(";");
Serial.print(now.hour(), DEC);
Serial.print(":");
Serial.print(now.minute(), DEC);
Serial.print(":");
Serial.print(now.second(), DEC);
Serial.print(";");
Serial.print(c, DEC);
Serial.print(";");
Serial.print(voltageString);

```

#### 4.5.4 Hasil Pengujian Komponen LCD OLED (*Organic Light Emitting Diode*)

Hasil pengujian LCD OLED (*Organic Light Emitting Diode*) sudah dapat dilihat bahwa LCD OLED (*Organic Light Emitting Diode*) sudah bisa menampilkan data sensor dan sisa baterai yang dikirimkan oleh Arduino Pro Mini pro mini. Hasilnya dapat dilihat pada gambar dibawah ini:



Gambar 4.6 Tampilan LCD OLED (*Organic Light Emitting Diode*)

OLED sudah berhasil menampilkan data dari sensor dan data dari sensor voltase.

#### 4.6 Efisiensi daya dari pengamatan waktu

Tujuan pengamatan waktu adalah, untuk mengetahui tingkat efisiensi dari ketiga fungsi yang ditanamkan pada *prototype* ini, semakin lama durasi waktu *prototype* menyala ketika menggunakan fungsi tersebut, maka dapat memudahkan untuk menyimpulkan bahwa fungsi mana yang lebih efisien daya.

##### 4.6.1 Dengan Fungsi Delay (1000)

Tujuan pengujian Fungsi Delay(1000) adalah untuk mengetahui durasi waktu penggunaan *prototype* jika hanya menggunakan fungsi Delay(1000) dari Arduino Pro Mini Pro Min..

##### A. Peralatan Pengujian Delay(1000)

1. *Prototype* Pantauan data *Heart Rate*.
2. *Handphone* Sebagai *Router Prototype*.

##### B. Prosedur Pengujian Delay(1000)

1. Pasang alat dan sensor pada jari tangan.
2. Mulai hidupkan alat.
3. Ubah fungsi pada *prototype* dengan menggeser switch ke arah On.
4. *Upload Program* sampai selesai.
5. Lihat waktu pada data awal pada Database Firebase ketika baterai masih 100%, kemudian lihat data di database firebase kembali untuk melihat data terakhir saat kondisi baterai 70%, kemudian lihat data awal pada Database Firebase saat kondisi baterai menunjukkan 69% untuk menandakan waktu terakhir pengamatan.

##### C. Hasil Pengujian Delay(1000)

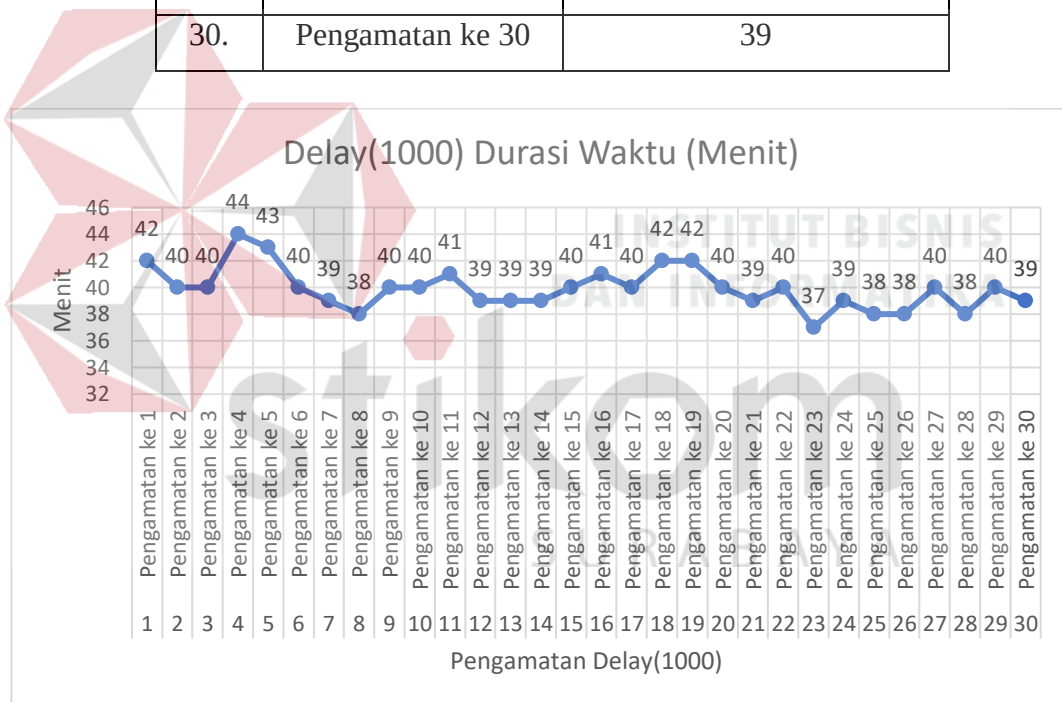


Dalam hasil pengujian kali ini akan ditampilkan dalam bentuk tabel dan menunjukkan durasi waktu dalam satuan menit. Hasil waktu pengujian didapatkan setelah melihat data pada *Database Firebase*, kemudian data waktu pengambilan awal dikurangi dengan data waktu pengambilan akhir, sehingga didapatkan selisih menit dari setiap pengambilan data, kemudian data selisih tersebut yang diambil untuk dimasukkan ke dalam tabel pengamatan.

Tabel 4.1 Tabel Hasil Pengamatan Waktu Fungsi Delay(1000)

| <b>Delay(1000)</b> |                  |                      |
|--------------------|------------------|----------------------|
| No                 | Pengamatan ke -  | Durasi Waktu (menit) |
| 1.                 | Pengamatan ke 1  | 42                   |
| 2.                 | Pengamatan ke 2  | 40                   |
| 3.                 | Pengamatan ke 3  | 40                   |
| 4.                 | Pengamatan ke 4  | 44                   |
| 5.                 | Pengamatan ke 5  | 43                   |
| 6.                 | Pengamatan ke 6  | 40                   |
| 7.                 | Pengamatan ke 7  | 39                   |
| 8.                 | Pengamatan ke 8  | 38                   |
| 9.                 | Pengamatan ke 9  | 40                   |
| 10.                | Pengamatan ke 10 | 40                   |
| 11.                | Pengamatan ke 11 | 41                   |
| 12.                | Pengamatan ke 12 | 39                   |
| 13.                | Pengamatan ke 13 | 39                   |
| 14.                | Pengamatan ke 14 | 39                   |
| 15.                | Pengamatan ke 15 | 40                   |
| 16.                | Pengamatan ke 16 | 41                   |
| 17.                | Pengamatan ke 17 | 40                   |
| 18.                | Pengamatan ke 18 | 42                   |
| 19.                | Pengamatan ke 19 | 42                   |
| 20.                | Pengamatan ke 20 | 40                   |

| Delay (1000) |                  |                      |
|--------------|------------------|----------------------|
| No.          | Pengamatan ke -  | Durasi Waktu (menit) |
| 21.          | Pengamatan ke 21 | 39                   |
| 22.          | Pengamatan ke 22 | 40                   |
| 23.          | Pengamatan ke 23 | 37                   |
| 24.          | Pengamatan ke 24 | 39                   |
| 25.          | Pengamatan ke 25 | 38                   |
| 26.          | Pengamatan ke 26 | 38                   |
| 27.          | Pengamatan ke 27 | 40                   |
| 28.          | Pengamatan ke 28 | 38                   |
| 29.          | Pengamatan ke 29 | 40                   |
| 30.          | Pengamatan ke 30 | 39                   |



Gambar 4.7 Grafik Data Waktu Durasi Fungsi Delay(1000)

#### D. Nilai Rata – Rata Waktu Pengamatan Delay(1000).

$$\frac{\text{Jumlah Data Waktu}}{\text{Jumlah Pengamatan}} = \frac{1197}{30} = 39,9 \text{ menit}$$

#### 4.6.2 Dengan Fungsi *LowPower.powerDown*

Tujuan pengujian Fungsi *LowPower.powerDown* adalah untuk mengetahui durasi waktu penggunaan *prototype* jika hanya menggunakan fungsi *LowPower.powerDown* dari Arduino Pro Mini Pro Mini

##### A. Peralatan Pengujian *LowPower.PowerDown*

1. *Prototype* Pantauan data Heart Rate.
2. *Handphone* Sebagai *Router Prototype*.

##### B. Prosedur Pengujian *LowPower.PowerDown*

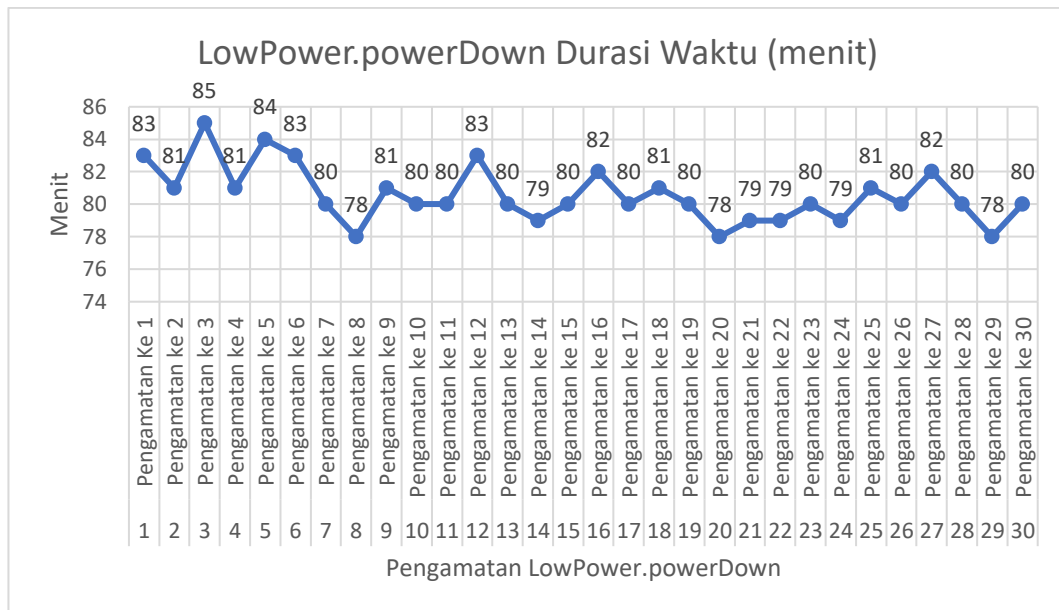
1. Pasang alat dan sensor pada jari tangan.
2. Mulai hidupkan alat.
3. Ubah fungsi pada *prototype* dengan menggeser switch ke arah On.
4. *Upload Program* sampai selesai.
5. Lihat waktu pada data awal pada Database Firebase ketika baterai masih 100%, kemudian lihat data di database firebase kembali untuk melihat data terakhir saat kondisi baterai 70%, kemudian lihat data awal pada Database Firebase saat kondisi baterai menunjukkan 69% untuk menandakan waktu terakhir pengamatan.

##### C. Hasil Pengujian *LowPower.PowerDown*

Dalam hasil pengujian kali ini akan ditampilkan dalam bentuk tabel dan menunjukkan durasi waktu dalam satuan menit. Hasil waktu pengujian didapatkan setelah melihat data pada *Database Firebase*, kemudian data waktu pengambilan awal dikurangi dengan data waktu pengambilan akhir, sehingga didapatkan selisih menit dari setiap pengambilan data, kemudian data selisih tersebut yang diambil untuk dimasukkan ke dalam tabel pengamatan.

Tabel 4.2 Tabel Hasil Pengamatan Waktu Fungsi LowPower.powerDown

| <b>LowPower.powerDown</b> |                  |                      |
|---------------------------|------------------|----------------------|
| No                        | Pengamatan Ke -  | Durasi Waktu (menit) |
| 1.                        | Pengamatan Ke 1  | 83                   |
| 2.                        | Pengamatan ke 2  | 81                   |
| 3.                        | Pengamatan ke 3  | 85                   |
| 4.                        | Pengamatan ke 4  | 81                   |
| 5.                        | Pengamatan ke 5  | 84                   |
| 6.                        | Pengamatan ke 6  | 83                   |
| 7.                        | Pengamatan ke 7  | 80                   |
| 8.                        | Pengamatan ke 8  | 78                   |
| 9.                        | Pengamatan ke 9  | 81                   |
| 10.                       | Pengamatan ke 10 | 80                   |
| 11.                       | Pengamatan ke 11 | 80                   |
| 12.                       | Pengamatan ke 12 | 83                   |
| 13.                       | Pengamatan ke 13 | 80                   |
| 14.                       | Pengamatan ke 14 | 79                   |
| 15.                       | Pengamatan ke 15 | 80                   |
| 16.                       | Pengamatan ke 16 | 82                   |
| 17.                       | Pengamatan ke 17 | 80                   |
| 18.                       | Pengamatan ke 18 | 81                   |
| 19.                       | Pengamatan ke 19 | 80                   |
| 20.                       | Pengamatan ke 20 | 78                   |
| 21.                       | Pengamatan ke 21 | 79                   |
| 22.                       | Pengamatan ke 22 | 79                   |
| 23.                       | Pengamatan ke 23 | 80                   |
| 24.                       | Pengamatan ke 24 | 79                   |
| 25.                       | Pengamatan ke 25 | 81                   |
| 26.                       | Pengamatan ke 26 | 80                   |
| 27.                       | Pengamatan ke 27 | 82                   |
| 28.                       | Pengamatan ke 28 | 80                   |
| 29.                       | Pengamatan ke 29 | 78                   |
| 30.                       | Pengamatan ke 30 | 80                   |



Gambar 4 8 Grafik Data Fungsi LowPower.powerDown

#### D. Nilai Rata – Rata Waktu Pengamatan LowPower.powerDown

$$\frac{\text{Jumlah Data Waktu}}{\text{Jumlah Pengamatan}} = \frac{2417}{30} = 80,56 \text{ menit}$$

#### 4.6.3 Dengan Fungsi LowPower.idle

Tujuan pengujian Fungsi LowPower.idle adalah untuk mengetahui durasi waktu penggunaan *prototype* jika hanya menggunakan fungsi *LowPower.idle* dari *Arduino Pro Mini Pro Mini*.

##### A. Peralatan Pengujian LowPower.idle

1. *Prototype* Pantauan data Heart Rate.
2. *Handphone* Sebagai Router *Prototype*.

##### B. Prosedur Pengujian LowPower.idle

1. Pasang alat dan sensor pada jari tangan.
2. Mulai hidupkan alat.

3. Ubah fungsi pada *prototype* dengan menggeser switch ke arah On.
4. *Upload Program* sampai selesai.
5. Lihat waktu pada data awal pada Database Firebase ketika baterai masih 100%, kemudian lihat data di database firebase kembali untuk melihat data terakhir saat kondisi baterai 70%, kemudian lihat data awal pada *Database Firebase* saat kondisi baterai menunjukkan 69% untuk menandakan waktu terakhir pengamatan.

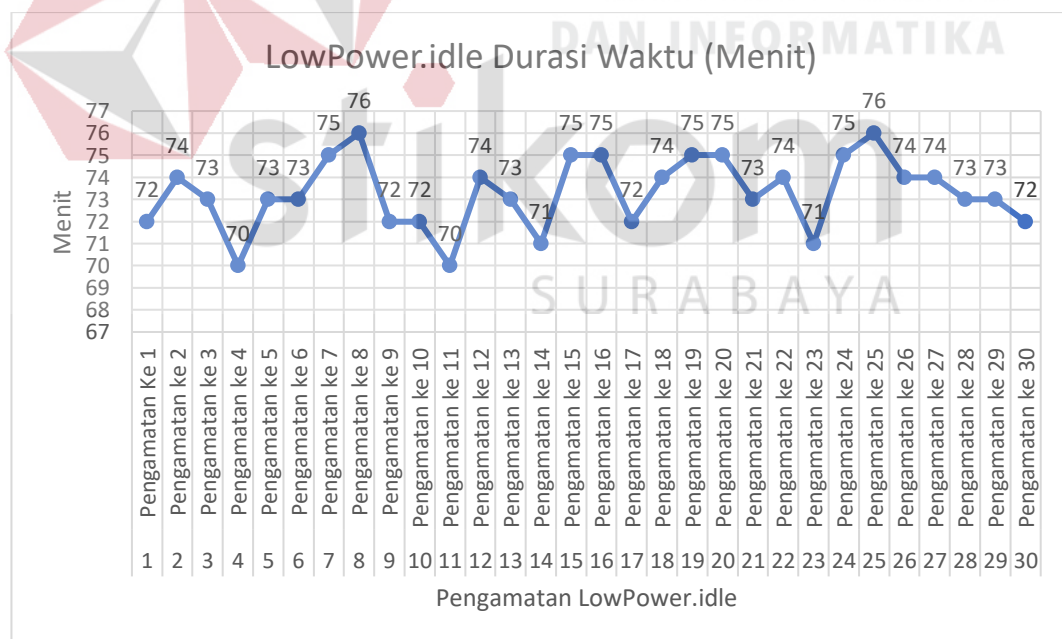
### C. Hasil Pengujian LowPower.idle

Dalam hasil pengujian kali ini akan ditampilkan dalam bentuk tabel dan menunjukkan durasi waktu dalam satuan menit. Hasil waktu pengujian didapatkan setelah melihat data pada *Database Firebase*, kemudian data waktu pengambilan awal dikurangi dengan data waktu pengambilan akhir, sehingga didapatkan selisih menit dari setiap pengambilan data, kemudian data selisih tersebut yang diambil untuk dimasukkan ke dalam tabel pengamatan.

Tabel 4.3 Tabel Hasil Pengamatan Waktu Fungsi LowPower.idle

| LowPower.idle |                  |                      |
|---------------|------------------|----------------------|
| No            | Pengamatan ke -  | Durasi Waktu (menit) |
| 1.            | Pengamatan ke 1  | 72                   |
| 2.            | Pengamatan ke 2  | 74                   |
| 3.            | Pengamatan ke 3  | 73                   |
| 4.            | Pengamatan ke 4  | 70                   |
| 5.            | Pengamatan ke 5  | 73                   |
| 6.            | Pengamatan ke 6  | 73                   |
| 7.            | Pengamatan ke 7  | 75                   |
| 8.            | Pengamatan ke 8  | 76                   |
| 9.            | Pengamatan ke 9  | 72                   |
| 10.           | Pengamatan ke 10 | 72                   |
| 11.           | Pengamatan ke 11 | 70                   |
| 12.           | Pengamatan ke 12 | 74                   |
| 13.           | Pengamatan ke 13 | 73                   |

| LowPower.idle |                  |                      |
|---------------|------------------|----------------------|
| No.           | Pengamatan ke -  | Durasi Waktu (menit) |
| 14.           | Pengamatan ke 14 | 71                   |
| 15.           | Pengamatan ke 15 | 75                   |
| 16.           | Pengamatan ke 16 | 75                   |
| 17.           | Pengamatan ke 17 | 72                   |
| 18.           | Pengamatan ke 18 | 74                   |
| 19.           | Pengamatan ke 19 | 75                   |
| 20.           | Pengamatan ke 20 | 75                   |
| 21.           | Pengamatan ke 21 | 73                   |
| 22.           | Pengamatan ke 22 | 74                   |
| 23.           | Pengamatan ke 23 | 71                   |
| 24.           | Pengamatan ke 24 | 75                   |
| 25.           | Pengamatan ke 25 | 76                   |
| 26.           | Pengamatan ke 26 | 74                   |
| 27.           | Pengamatan ke 27 | 74                   |
| 28.           | Pengamatan ke 28 | 73                   |
| 29.           | Pengamatan ke 29 | 73                   |
| 30.           | Pengamatan ke 30 | 72                   |

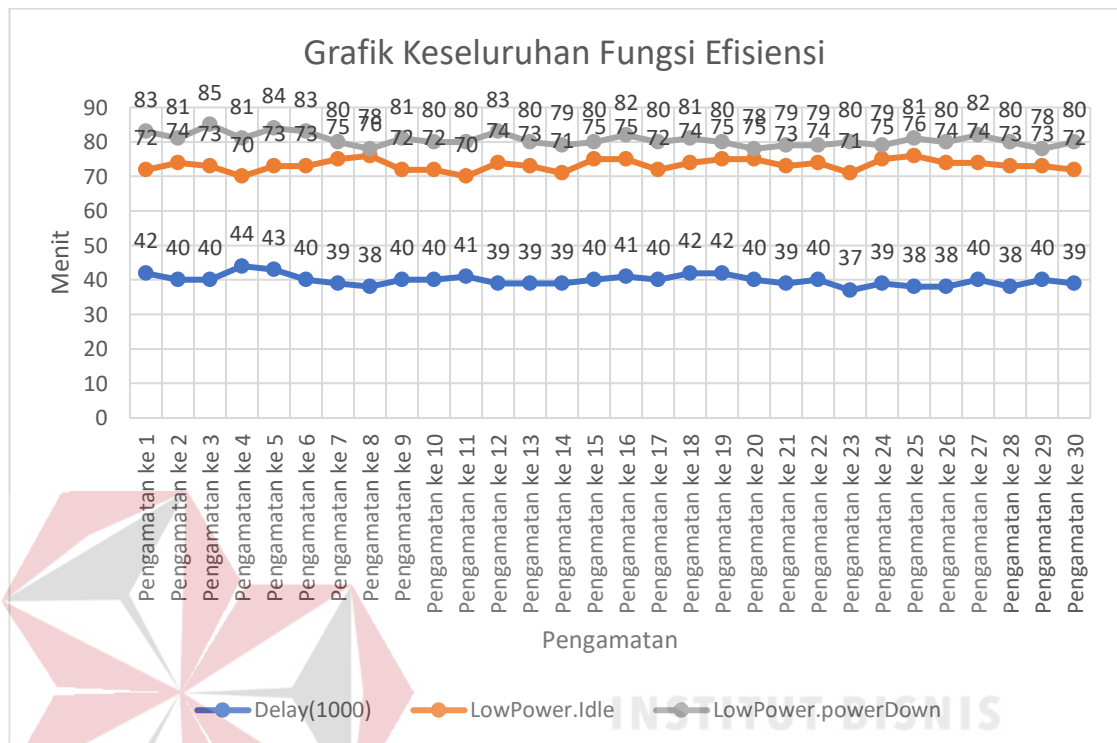


Gambar 4.9 Grafik Data Durasi Waktu Fungsi LowPower.idle

#### D. Nilai Rata – Rata Waktu Pengamatan LowPower.idle

$$\frac{\text{Jumlah Data Waktu}}{\text{Jumlah Pengamatan}} = \frac{2199}{30} = 73,3 \text{ menit}$$

#### 4.6.4 Grafik Data Keseluruhan Fungsi



Gambar 4.10 Grafik Data keseluruhan Fungsi Efisiensi

#### 4.7 Nilai Ampere Dari Masing – Masing Fungsi

Dari percobaan dan pengamatan yang dilakukan menghasilkan data nilai ampere yang muncul pada alat pengukuran Avometer, dari percobaan dan pengamatan tersebut digunakan alat pengukuran Avometer Analog, nilai yang dihasilkan relative hampir sama yaitu berkisar antara 90 – 101 mA, namun terjadi perbedaan kondisi jarum yang menunjukkan nilai pada Avometer, perbedaanya adalah, ketika menggunakan Fungsi Delay(1000), maka Jarum akan stabil pada angka 101mA tidak ada penurunan nilai sama sekali, namun ketika digunakan fungsi LowPower.idle, maka kondisi jarum mengalami penurunan nilai berkisar pada angka 94 mA, namun secara cepat akan kembali ke angka 101 mA kembali, kemudian pada percobaan menggunakan fungsi Power.powerDown maka



didapatkan hasil nilai yang juga sama 101 mA, namun untuk kondisi jarumnya mengalami penurunan hingga angka berkisar 90 mA dengan gerakan yang lambat untuk kembali pada angka 101 mA. Dan pada pengujian menggunakan Avometer digital untuk fungsi Delay(1000) menunjukkan nilai 42,3mA, dan menggunakan fungsi LowPower.idle menunjukkan nilai 95,2mA sedangkan ketika menggunakan fungsi LowPower.powerDown menunjukkan nilai 92,1mA.

Terdapat perbedaan nilai terhadap Avometer digital dan analog, namun yang sangat terlihat signifikan adalah ketika menggunakan Avometer digital, perbandingan ini dimaksudkan untuk memberikan data dari perbedaan pengamatan daya ketika menggunakan Avometer digital dan analog. Untuk penyebabnya harus dikaji lebih ulang karena disini penulis hanya berfokus pada cara menghemat daya menggunakan algoritma efisiensi.

Tabel 4.4 Tabel Nilai Avometer

| No | Fungsi             | Avometer Analog | Avometer Digital |
|----|--------------------|-----------------|------------------|
| 1. | Delay(1000)        | 101 mA          | 42,3mA           |
| 2. | LowPower.idle      | 94 mA – 101mA   | 95,2mA           |
| 3. | LowPower.powerDown | 90mA – 101mA    | 92,1mA           |

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Dari perancangan *program* dan alat hingga pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Metode *Idle Time – Deep Sleep* pada *prototype Monitoring Heart Rate* diterapkan dengan menggunakan *function powerDown* dan *idle* dari *Library LowPower*.
2. Fungsi *LowPower.powerDown* memiliki tingkat efisiensi yang paling tinggi dibandingkan *LowPower.idle* dan *Delay(1000)*
3. Durasi Waktu dari ketiga fungsi adalah berbeda dimana *LowPower.powerDown* rata – rata berkisar 80,56 menit, dan *LowPower.idle* rata – rata berkisar 73,3 menit sedangkan *Delay(1000)* rata – rata berkisar 39,9 menit.

#### 5.2 Saran

Dalam pengembangan selanjutnya dapat dilakukan integrasi dengan *smartphone / android* dalam ekstensi *.apk* sehingga lebih *portable* dan bisa dijangkau oleh setiap lapisan masyarakat dan pengamatan data bisa jadi lebih mudah. Dan ketika menguji *Prototype* ini sebaiknya jaringan dipastikan stabil, agar pembacaan dan pengiriman data di Database Firebase berjalan dengan lancar, serta dalam pengemasan *Prototype* sebaiknya menggunakan desain yang mengikuti bentuk *Prototype* sehingga alat terlihat lebih ringkas dan bagus.

## DAFTAR PUSTAKA

- Arduino. (n.d.). *LowPower.deepSleep()*. Retrieved from [www.Arduino.cc:https://www.arduino.cc/en/Reference/LowPowerDeepSleep](http://www.Arduino.cc:https://www.arduino.cc/en/Reference/LowPowerDeepSleep)
- Arduino. (n.d.). *LowPower.idle*. Retrieved June 27, 2019, from [www.Arduino.cc:https://www.arduino.cc/en/Reference/LowPowerIdle](http://www.Arduino.cc:https://www.arduino.cc/en/Reference/LowPowerIdle)
- madcoffee. (2015, February 18). *Home Automation Community*. Retrieved from Home Automation Community: <http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/>
- Martinez, B., Monton, M., Member, IEEE, & Prades, J. D. (2015). The Power Of Models : Modeling Power Consumption for IoT Device. *IEEE Sensors Journal*, 1-3.
- mrguen. (2018, October 8). *Low-Power/LowPower.h*. Retrieved June 29, 2019, from [www.GitHub.com:https://github.com/rocketscream/Low-Power/blob/master/LowPower.h](http://www.GitHub.com:https://github.com/rocketscream/Low-Power/blob/master/LowPower.h)
- Musayyanah, Puspasari, I., & Susanto, P. (2018). Monitoring Target Heart Rate (THR) Untuk Latihan Lari Berbasis Internet Of Things. *Engineering and Sains Journal*, 87 - 94.
- Pelayo, R. (2018). *Arduino Lipo battery Monitor*. Retrieved from Microcontroller Tutorials: <https://www.teachmemicro.com/arduino-lipo-battery-monitor/#>
- Rintiasti, A., Bangsawan, H. T., Suhartono, A. A., Mahmudah, L., & Bahari, N. S. (2018). Pemantauan Proses Fermentasi Tembakau Menggunakan Platform IoT. *Desiminasi Hasil Litbang Baristand Industri Surabaya*, -.
- Rohner, A. (2015, August 8). *How to modify an Arduino Pro Mini (clone) for low power consumption*. Retrieved from My Notebook: <https://andreasrohner.at/posts/Electronics/How-to-modify-an-Arduino-Pro-Mini-clone-for-low-power-consumption/>
- Samsugi, S., Ardiansyah, & Kastutara, D. (2018). Arduino dan Modul Wifi ESP8266 Sebagai Media Kendali Jarak Jauh dengan Antarmuka Berbasis Android. *Jurnal Teknoinfo*, 12, 23 - 27.
- Verma, S. K. (2011). An Efficient Power Mode Scheduling Algorithm For Adhoc Network. *International Journal Of Engineering Science And Technology (IJEST)*, 1770 - 1775.