



HEART & MIND TOWARDS EXCELLENCE

**IMPLEMENTASI *MULTI-USER SHARED ACCESS* UNTUK PEMANTAUAN
DATA *PERSONAL HEART RATE* MENGGUNAKAN ANDROID**

TUGAS AKHIR

Program Studi

S1 Teknik Komputer

Oleh :

YAHYA NUGROHO

15410200014

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

**IMPLEMENTASI *MULTI-USER SHARED ACCESS* UNTUK
PEMANTAUAN DATA *PERSONAL HEART RATE*
MENGUNAKAN ANDROID**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik

Disusun Oleh :

Nama : Yahya Nugroho

NIM : 15410200014

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

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

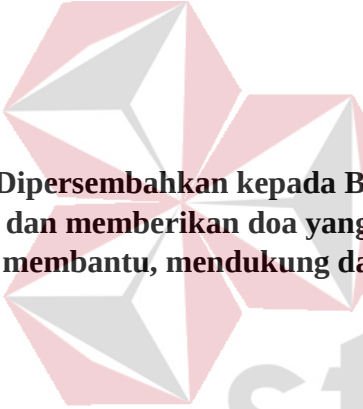
2019



“Terus kejar berbagai kemungkinan meski keberhasilan hanya 0.0001%”

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA



Dipersembahkan kepada Bapak, Ibu dan Keluarga saya yang selalu memotivasi dan memberikan doa yang terbaik kepada saya serta semua orang yang selalu membantu, mendukung dan memotivasi agar terus belajar dan berkarya lebih baik dari hari kemarin

stikom
SURABAYA

TUGAS AKHIR
IMPLEMENTASI MULTI-USER SHARED ACCESS UNTUK PEMANTAUAN
DATA PERSONAL HEART RATE MENGGUNAKAN ANDROID

Dipersiapkan dan disusun oleh

Yahya Nugroho

NIM : 15410200014

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji

Pada : 5 Agustus 2019

Susunan Dewan Penguji

Pembimbing

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

NIDN. 0729047501

II. Musayyanah, S.ST., M.T.

NIDN. 0730069102

Pembahas

I. Dr. Susijanto Tri Rasmana, S.Kom., M.T.

NIDN. 0727097302

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



FAKULTAS TEKNOLOGI
DAN INFORMATIKA

stikom

SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Yahya Nugroho
NIM : 15410200014
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **IMPLEMENTASI MULTI-USER SHARED ACCESS UNTUK PEMANTAUAN DATA PERSONAL HEART RATE MENGGUNAKAN ANDROID**

Menyatakan dengan sesungguhnya bahwa:

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

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 5 Agustus 2019



NIM : 15410200014

ABSTRAK

Olahraga adalah bagian penting dalam menjaga kesehatan. Salah satu jenis olahraga yang murah dan mudah dilakukan adalah lari (*jogging*). Olahraga lari di tempat *fitness* memiliki kelebihan karena adanya *monitoring* dari *Personal Trainer* (PT). PT adalah seseorang yang membantu dalam membimbing kita ketika sedang berolahraga agar berjalan dengan tepat. Agar PT dapat memberikan saran untuk hasil yang maksimal, dibutuhkan pencatatan nilai *heart rate* (HR) ketika latihan.

Pada penelitian sebelumnya telah dilakukan pembuatan *prototype* alat untuk pencatatan HR yang sudah berbasis IoT, namun manajemen *database* data yang dikirim oleh ESP8266-01 masih di *setting* secara manual, dan *devices* yang digunakan masih *single user*. Pada penelitian ini, penulis telah membangun sebuah sistem dimana setiap *device user* dapat terintegrasi dan dapat dipantau data *history* bersamaan secara *multi-user shared access*.

Sistem dirancang dengan menggunakan Arduino Pro Mini sebagai mikrokontrollernya, ESP-01 sebagai transmisi dengan Firebase, dan Firebase sebagai *database*. Berdasarkan pengujian 3 *devices* dengan beberapa variasi *on-off* bersamaan maupun bergantian menggunakan *database* Firebase berstruktur JSON *tree*, terbukti dapat menangani data untuk keperluan *multi-user shared access*. Firebase dapat menerima data secara *realtime* dan data yang dikirimkan tertata rapi menggunakan JSON *buffer*.

Kata Kunci : Firebase, *Internet of things*, *Heart rate*, *Multi-user shared access*.

KATA PENGANTAR

Atas berkat rahmat Tuhan Yang Maha Esa, maka penulis berhasil menyelesaikan dan menyusun naskah Tugas Akhir yang berjudul *Implementasi Multi-User Shared Access Untuk Pemantauan Data Personal Heart Rate Menggunakan Android*. 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 Sistem 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.S.T., M.T., selaku dosen pembimbing kedua yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
6. Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., selaku dosen pembahas yang telah memberikan arahan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
7. Seluruh dosen pengajar Program Studi S1 Sistem 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 Sistem Komputer dan semua pihak yang telah memberikan dorongan dan bantuan baik secara langsung maupun secara tidak langsung.
9. Teman-teman yang selalu memberikan dorongan baik itu dari dalam maupun dari luar kampus, terimakasih telah menemani baik dalam suka maupun duka.

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

Surabaya, 5 Agustus 2019

Penulis

DAFTAR ISI

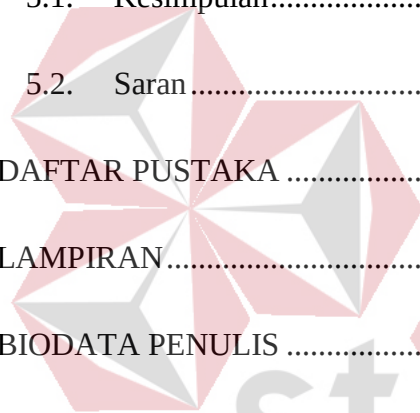
	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xv
DAFTAR TABEL	xviii
DAFTAR LAMPIRAN.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1 Definisi <i>Multi-User Shared Access</i>	5
2.2 Definisi Heart Rate	6
2.3 Target Heart Rate (THR).....	6
2.4 Arduino Pro Mini	8

2.5	ESP 8266 - 01.....	9
2.6	USB To TTL	10
2.7	UART	10
2.8	Buzzer.....	11
2.9	Grove – Finger Clip Heart Rate Sensor.....	12
2.10	Li-po Battery YTong 702030	13
2.11	Battery Charger TP4056.....	13
2.12	RTC DS3231	14
2.13	Firebase (Database)	15
2.14	Android Studio	16
2.15	Arduino IDE	17
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM		19
3.1	Metode Penelitian.....	19
3.2	Perancangan Perangkat Keras	20
3.2.1.	Rancangan <i>Hardware</i>	20
3.2.2.	Desain Alat.....	23
3.3	Perancangan <i>Software</i>	24
3.3.1.	<i>Flowchart</i> Arduino Pro Mini	24
3.3.2.	<i>Flowchart</i> ESP8266-01.....	26

3.3.3.	JSON Tree Firebase	27
3.3.4.	Flowchart Aplikasi Android.....	28
BAB IV HASIL UJI DAN PENGAMATAN		29
4.1.	Pengujian Upload Program Mikrokontroler Arduino Pro Mini	29
4.1.1.	Tujuan	29
4.1.2.	Peralatan yang digunakan	29
4.1.3.	Cara Pengujian	30
4.2.	Pengujian Upload Program ESP8266-01	32
4.2.1.	Tujuan	32
4.2.2.	Peralatan yang digunakan	32
4.2.3.	Cara pengujian	32
4.3.	Pengujian Tombol, LED, dan Buzzer	36
4.3.1.	Tujuan	36
4.3.2.	Peralatan yang digunakan	36
4.3.3.	Cara pengujian	36
4.4.	Pengujian RTC DS3231	39
4.4.1.	Tujuan	39
4.4.2.	Peralatan yang digunakan	40
4.4.3.	Cara pengujian	40

4.5.	Pengujian OLED 128x64	42
4.5.1.	Tujuan	42
4.5.2.	Peralatan yang digunakan	42
4.5.3.	Cara pengujian	43
4.6.	Pengujian Grove – Finger Clip Heart Rate Sensor.....	46
4.6.1.	Tujuan	46
4.6.2.	Peralatan yang digunakan	46
4.6.3.	Cara pengujian	46
4.7.	Pengujian Pengiriman Data ESP01 - Firebase (Create, Read, Update)	49
4.7.1.	Tujuan	49
4.7.2.	Peralatan dan yang digunakan.....	49
4.7.3.	Cara pengujian	50
4.8.	Pengujian Pengiriman Data Antara Arduino – ESP01 – Firebase	53
4.8.1.	Tujuan	53
4.8.2.	Peralatan yang digunakan	54
4.8.3.	Cara pengujian	54
4.9.	Pengujian Monitoring History HR pada Android	57
4.9.1.	Tujuan	57
4.9.2.	Peralatan yang digunakan	58

4.9.3. Cara pengujian	58
4.10. Pengujian Keseluruhan Sistem	61
4.10.1. Tujuan	61
4.10.2. Peralatan yang digunakan	61
4.10.3. Cara pengujian	62
BAB V KESIMPULAN.....	76
5.1. Kesimpulan.....	76
5.2. Saran.....	76
DAFTAR PUSTAKA	77
LAMPIRAN.....	78
BIODATA PENULIS	108



INSTITUT BISNIS
 DAN INFORMATIKA
stikom
 SURABAYA

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Gambaran multi-user shared access	5
Gambar 2.2. Bentuk fisik dan pin Arduino Pro Mini.....	9
Gambar 2.3. Bentuk fisik dan pin ESP8266-01	10
Gambar 2.4. Bentuk fisik USB to TTL	10
Gambar 2.5. Bentuk fisik buzzer.....	11
Gambar 2.6. Bentuk fisik Grove-finger clip heart rate sensor	12
Gambar 2.7. Bentuk fisik baterai Ytong 702030	13
Gambar 2.8. Bentuk fisik dan pin modul TP4056	14
Gambar 2.9. Bentuk fisik RTC DS3231	15
Gambar 2.10. Contoh Struktur JSON tree	16
Gambar 2. 11. Tampilan software Arduino IDE.....	18
Gambar 2.12. Tampilan serial monitor Arduino IDE	18
Gambar 3.1. Diagram alur penelitian.....	19
Gambar 3.2. Diagram alur sistem	20
Gambar 3.3. Skema diagram blok Hardware per-komponen.....	21
Gambar 3.4. Diagram Skematik.....	22
Gambar 3.5. Peletakan sensor HR pada jari.....	23
Gambar 3.6. Peletakan alat pada lengan atas	23
Gambar 3.7. Flowchart Arduino Pro Mini.....	24
Gambar 3.8. Flowchart ESP-01	26

Gambar 3.9. Struktur JSON tree Firebase.....	27
Gambar 3.10. Flowchart aplikasi Android.....	28
Gambar 4.1. Setting pada menu Tools untuk Arduino Pro Mini	30
Gambar 4.2. Pesan status pada Arduino IDE	31
Gambar 4.3. Hasil tes upload program.....	31
Gambar 4.4. Rangkaian upload program ESP-01 dengan USB to TTL	33
Gambar 4.5. Download boards ESP8266.....	34
Gambar 4.6. Setting pada menu Tools untuk ESP-01.....	34
Gambar 4.7. Setting serial monitor untuk memantau hasil tes	35
Gambar 4.8. ESP-01 sudah terhubung dengan <i>access point</i>	36
Gambar 4.9. Rangkaian IO Switch, Push, LED, dan Buzzer.....	37
Gambar 4.10. Hasil serial monitor tes switch/push button	39
Gambar 4.11. LED menyala pada male mode	39
Gambar 4.12. Rangkaian Arduino Pro Mini dan RTC DS3231	40
Gambar 4.13. Cara memasukkan library.....	41
Gambar 4.14. Hasil output data RTC serial monitor	42
Gambar 4.15. Rangkaian Arduino Pro Mini dan Adafruit OLED 128x64	43
Gambar 4.16. Hasil display OLED 128x64	45
Gambar 4.17. Rangkaian Arduino Pro Mini dan Sensor HR.....	47
Gambar 4.18. Hasil pembacaan HR.....	48
Gambar 4.19. Auth key firebase untuk ESP-01	50
Gambar 4.20. Nama firebase host untuk ESP-01.....	51
Gambar 4.21. Setting rule database Firebase	51

Gambar 4.22. Hasil pada serial monitor dan database Firebase	53
Gambar 4.23. Hasil data pada Firebase dan serial monitor	57
Gambar 4.24. Icon login akun Google	58
Gambar 4.25. Form .xml login.....	59
Gambar 4.26. Form .xml pilih user	59
Gambar 4.27. Date-Time Picker (1).....	60
Gambar 4.28. Form history HR (1).....	60
Gambar 4.29. Date-Time Picker (2).....	60
Gambar 4.30. Form history HR (2).....	60
Gambar 4.31. Format data yang dikirim ke ESP-01 (Mode Run)	63
Gambar 4.32. <i>History</i> HR user_1 tanggal 15/06/19.....	70
Gambar 4.33. <i>History</i> HR user_2 tanggal 15/06/19.....	70
Gambar 4.34. <i>History</i> HR user_3 tanggal 15/06/19.....	71
Gambar 4.35. <i>History</i> HR user_1 tanggal 20/07/19.....	71
Gambar 4.36. <i>History</i> HR user_2 tanggal 20/07/2019.....	72
Gambar 4.37. <i>History</i> HR user_3 tanggal 20/07/2019.....	72
Gambar 4.38. PT 3 dan user 3 menampilkan data user_3.....	73
Gambar 4.39. PT 2 dan user 2 menampilkan data user_2.....	74
Gambar 4.40. PT 3 dan user 3 menampilkan data user_3.....	74

DAFTAR TABEL

	Halaman
Tabel 4.1. Selisih perbandingan hasil sensor HR dengan Blood Pressure Monitor....	48
Tabel 4.2. History pencatatan HR tanggal 15/06/2019	64
Tabel 4.3. History pencatatan HR tanggal 20/07/2019	67



DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1. PROGRAM ARDUINO PRO MINI.....	78
LAMPIRAN 2. PROGRAM ESP-01.....	83
LAMPIRAN 3. PROGRAM XML activity_main	85
LAMPIRAN 4. PROGRAM JAVA MainActivity	86
LAMPIRAN 5. PROGRAM XML activity_pt	87
LAMPIRAN 6. PROGRAM JAVA pt_activity.....	88
LAMPIRAN 7. PROGRAM XML activity_chart_user	89
LAMPIRAN 8. PROGRAM JAVA Chart_User	90
LAMPIRAN 9. PROGRAM JAVA Chart_fragment	92
LAMPIRAN 10. PROGRAM AndroidManifest.xml	93
LAMPIRAN 11. DATA USER 1 TANGGAL 15-06-2019 PADA FIREBASE.....	94
LAMPIRAN 12. DATA USER 2 TANGGAL 15-06-2019 PADA FIREBASE.....	96
LAMPIRAN 13. DATA USER 3 TANGGAL 15-06-2019 PADA FIREBASE.....	98
LAMPIRAN 14. DATA USER 1 TANGGAL 20-07-2019 PADA FIREBASE.....	101
LAMPIRAN 15. DATA USER 2 TANGGAL 20-07-2019 PADA FIREBASE.....	103
LAMPIRAN 16. DATA USER 3 TANGGAL 20-07-2019 PADA FIREBASE.....	105

BAB I

PENDAHULUAN

1.1 Latar Belakang

Olahraga adalah bagian penting dalam menjaga kesehatan. Salah satu jenis olahraga yang murah dan mudah dilakukan adalah lari (*jogging*). Olahraga lari di tempat *fitness* memiliki kelebihan karena adanya *monitoring* dari *Personal Trainer* (PT). PT adalah seseorang yang membantu dalam membimbing olahragawan ketika sedang berolahraga, agar berjalan dengan tepat. PT dapat memberikan masukan untuk hasil olahraga yang maksimal dengan pencatatan dan *review history* dari catatan nilai *heart rate* (HR) ketika latihan. *Target heart rate* (THR) adalah sebuah parameter ketepatan ketika berolahraga untuk mendapatkan manfaat atau hasil yang maksimal dari latihan.

Pada penelitian yang telah dilakukan sebelumnya telah dilakukan pembuatan *prototype* alat untuk pencatatan HR yang sudah berbasis IoT, namun masih terdapat beberapa kekurangan. Yakni manajemen *database* data dari yang dikirim oleh ESP8266-01 masih di *setting* secara manual, dan *devices* yang digunakan masih *single user* (Musayyanah, 2018).

Oleh karena itu maka alat pencatat data *personal heart rate* yang telah dibuat sebelumnya akan dikembangkan menjadi *multi-user shared access*, yang mana akan dituangkan pada penelitian ini yang berjudul “**IMPLEMENTASI MULTI-USER SHARED ACCESS UNTUK PEMANTAUAN DATA PERSONAL HEART RATE MENGGUNAKAN ANDROID**”. Teknik *multi-user shared access*

merupakan salah satu metode komunikasi *multi-user* untuk teknologi IoT. Teknik ini cocok digunakan untuk *devices* yang memiliki konsumsi daya rendah dan *low cost* (Yuan, 2018). Agar data semua *user* terakses secara *real time* oleh PT, maka tugas akhir ini mengadopsi teknik *multi-user shared access*.

Sistem mengintegrasikan antara *hardware* dan *software*. *Hardware* terhubung dengan *access point* (AP) untuk mengirimkan data *heart rate* (HR) ke *database* Firebase, yang nantinya *user* atau *Personal Trainer* (PT) dapat melihat *history* dari HR melalui aplikasi Android yang dibuat.

1.2 Rumusan Masalah

Melihat dari latar belakang diatas, maka dapat disimpulkan permasalahan dalam pengerjaan tugas akhir adalah sebagai berikut :

1. Bagaimana merancang *hardware* untuk pembacaan HR menggunakan mikrokontroler.
2. Bagaimana merancang *database* (Firebase) untuk perekaman hasil deteksi *Heart Rate* secara *multi-user*.
3. Bagaimana mengintegrasikan antara *hardware*, *software*, dan *database* secara *real time* untuk *multi-user*.

1.3 Batasan Masalah

Adapun yang menjadi batasan masalah dari tugas akhir ini agar pembahasan tidak meluas adalah sebagai berikut :

1. Mikrokontroler yang digunakan adalah Arduino Pro Mini yang terhubung dengan ESP8266-01.

2. Aplikasi desktop yang digunakan untuk pembuatan *software* android adalah aplikasi Android Studio.
3. Banyak *devices* (alat) yang digunakan atau user berjumlah 3 buah.
4. Perhitungan untuk data HR max memiliki nilai intensitas 75% (Jogging).
5. Perhitungan THR menggunakan metode *Karvonen*.
6. Konsumsi daya pada *devices* diabaikan.
7. Pembacaan THR dianggap sudah benar berdasarkan penelitian yang sudah dilakukan sebelumnya, penulis hanya berfokus pada *multi-user devices*.

1.4 Tujuan

Berdasarkan rumusan masalah diatas, adapun tujuan dari tugas akhir atau penelitian yang ingin dicapai adalah :

1. Dapat merancang *hardware* untuk pembacaan HR menggunakan mikrokontroler
2. Dapat merancang *database* Firebase untuk perekaman hasil deteksi *Heart Rate* secara *multi-user*.
3. Dapat mengintegrasikan antara *hardware*, *software*, dan *database* secara *real time* untuk *multi-user*.

1.5 Sistematika Penulisan

BAB 1 PENDAHULUAN

Pada bab ini dituliskan penjelasan latar belakang permasalahan sistem pencatatan HR yang ada pada saat ini dan apa yang telah dilakukan pada penelitian sebelumnya. Penulis menjelaskan tujuan untuk dilanjutkannya penelitian tersebut

agar *device* yang dipantau dapat secara *Multi User Shared Access* oleh PT menggunakan database Firebase melalui aplikasi Android.

BAB II LANDASAN TEORI

Pada bab ini disampaikan perihal tentang landasan teori sistem, aplikasi yang digunakan, perangkat-perangkat yang digunakan, serta teori-teori dalam menyelesaikan masalah yang sesuai dengan topik dan fokus.

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan tentang langkah-langkah yang dilakukan penulis dalam melakukan penelitian untuk merancang sistem, yang mencakup Hardware (rangkaiian elektronik), software (coding), struktur *JSON tree* database Firebase, dan rancangan aplikasi sederhana untuk menyelesaikan permasalahan tugas akhir.

BAB IV HASIL PENGUJIAN DAN PENGAMATAN

Pada bab ini dibahas mengenai tujuan, bahan-bahan, dan langkah-langkah pengujian yang digunakan dalam merancang sistem. Mulai dari perangkat keras (*hardware*), perangkat lunak (*software*), pengaturan pengiriman komunikasi *serial*, dan pengaturan data pada *database* Firebase pada penelitian untuk mendapatkan hasil uji terhadap sistem yang dibuat dalam menyelesaikan permasalahan tugas akhir.

BAB V PENUTUP

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

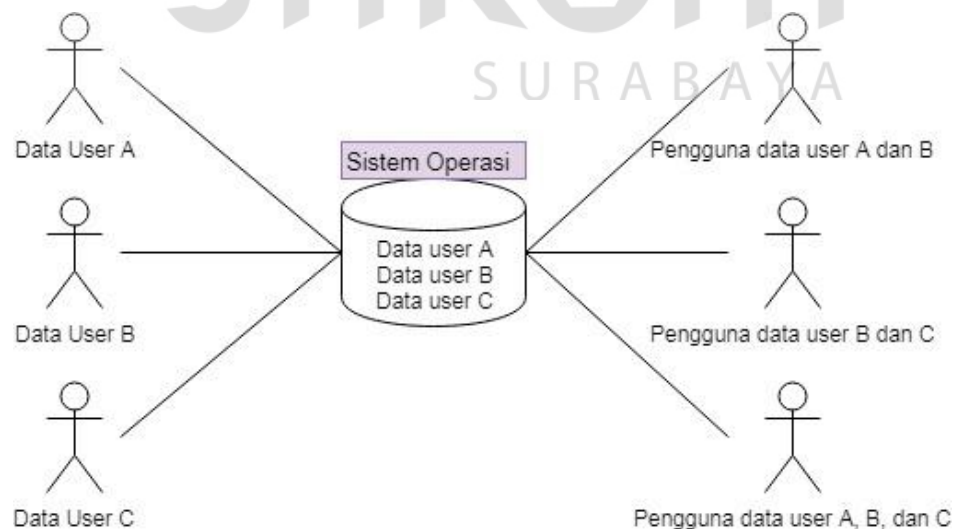
BAB II

LANDASAN TEORI

2.1 Definisi *Multi-User Shared Access*

Multi-user atau pengguna ganda adalah perangkat lunak (sistem operasi) yang memperbolehkan akses oleh beberapa pengguna dalam waktu bersamaan ke sistem operasi atau aplikasi tersebut. Istilah lawannya yaitu *single-user* mengacu kepada suatu sistem operasi yang hanya bisa digunakan oleh satu pengguna setiap saat (Indah Indriyana, 2012). Sedangkan *shared access* adalah istilah dimana sebuah data dapat diakses secara bersamaan oleh banyak *user*.

Secara keseluruhan, *multi-user shared access* pada sistem adalah sebuah konsep dari sistem banyak *user* yang diijinkan mengakses ke dalam sistem operasi atau aplikasi dan data, yang dapat diakses oleh semua *user*. Penggambaran *multi-user shared access* dapat dilihat pada gambar 2.1.



Gambar 2.1. Gambaran *multi-user shared access*

2.2 Definisi Heart Rate

Heart rate adalah jumlah detak jantung persatuan waktu atau yang biasa disebut dengan *beats per minute* (bpm), merupakan salah satu acuan untuk menentukan ketepatan intensitas olahraga. Nilai *heart rate* bisa didapat di belakang lutut, pergelangan tangan dan jari, leher, pelipis, dan kaki secara manual dengan menghitung sendiri detak jantung di tiap detiknya, namun kurang akurat. Untuk mendapatkan nilai *heart rate* yang lebih akurat, dibutuhkan alat monitoring *heart rate*, seperti yang digunakan oleh perawat atau dokter pada umumnya.

Tiap individu memiliki nilai *heart rate* yang berbeda, tergantung jumlah kadar oksigen yang terdapat pada tubuh. Semakin tinggi kadar oksigen maka semakin rendah nilai *heart rate* orang tersebut. Tinggi rendahnya *heart rate* juga dipengaruhi oleh aktifitas fisik (orang rutin olahraga dan tidak rutin berbeda), suhu udara, tingkat emosi, dan posisi tubuh. Nilai *heart rate* normal rata-rata berkisar antara 60-100bpm untuk usia 18 tahun keatas. Lamanya peningkatan *heart rate* seseorang juga dipengaruhi oleh rutinitas sehari-hari yang berpengaruh pada daya tahan tubuh (Mendia, 2016).

2.3 Target Heart Rate (THR)

THR adalah target *heart rate* yang ingin dicapai ketika berolahraga untuk mendapatkan manfaat atau hasil yang maksimal dari latihan (tidak terlalu ringan dan tidak terlalu berat). Adapun intensitas latihan fisik dibagi menjadi beberapa zona, yaitu :

1. Zona I

Zona paling rendah, biasanya zona untuk pemula atau orang yang jarang berolahraga, dengan nilai intensitas 50-60%. Contoh dari aktifitas zona I adalah berjalan kaki.

2. Zona II

Zona ini terdiri dari kegiatan yang membuat kalori terbakar dan membuat nafas lebih berat. Contoh dari aktifitas zona II adalah berjalan kaki cepat. Nilai intensitasnya 60-70%.

3. Zona III

Zona ini biasanya digunakan untuk ketahanan tubuh. Contoh dari aktifitas zona III adalah *jogging* atau *race walking*. Nilai intensitasnya 70%-80%.

4. Zona IV

Zona ini memiliki intensitas yang agak lebih berat dari jogging, seperti lari kecil naik tangga dengan santai. Nilai intensitasnya 80%-90%.

5. Zona V

Zona ini memiliki intensitas yang paling tinggi. Contoh dari aktifitas zona V adalah lari sprint, Nilai intensitasnya 90-100%.

Agar perhitungan tepat, sebaiknya dilakukan bertahap. Sebagai contoh ketika kita baru memulai kegiatan olahraga kembali setelah lama tidak berolahraga, nilai intensitas dimulai dari yg terendah di *zone* 50% untuk minggu pertama. Setelah itu bertahap naik ke *zone* 75%, lalu 6 minggu kemudian ketika masih rutin melakukan kegiatan olahraga naik ke THR *zone* 85%.

Metode Karvonen adalah salah satu metode untuk menghitung *target heart rate* (THR). Menggunakan nilai *heart rate* maksimal (HR max), *rest heart rate* (HR istirahat), dan intensitas dalam perhitungannya. Besarnya nilai *target heart rate* (THR) dapat dihitung menggunakan persamaan sebagai berikut :

$$\text{THR} = \{(\text{HR max} - \text{HR istirahat}) * \text{intensitas}\} + \text{HR istirahat}$$

Keterangan :

1. Perhitungan HR max untuk wanita dan pria berbeda (She, J., 2015), berikut

rumusnya :

$$\text{HR max (wanita)} = 206 - (0.88 * \text{usia})$$

$$\text{HR max (laki-laki)} = 206.9 - (0.67 * \text{usia})$$

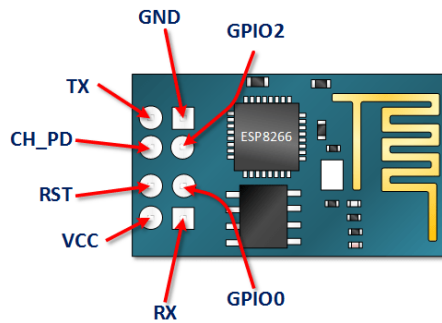
2. HR istirahat adalah nilai *heart rate* ketika bangun tidur atau tidak melakukan aktivitas.
3. Intensitas didapat dari nilai persentase zona yang telah dijelaskan diatas.

2.4 Arduino Pro Mini

Arduino pro mini hampir sama seperti Arduino pada umumnya, namun ukurannya lebih kecil berdimensi 3,3 x 1,8 cm. Arduino pro mini menggunakan mikrokontroler ATmega 328, memiliki 14 digital pin I/O (6 bisa sebagai output PWM), 6 input untuk analog, *on-board resonator*, *reset button*, dan lubang untuk pin *header*. Terdapat dua jenis Arduino pro mini, yang pertama berjalan pada 3,3V (memiliki clock 8MHz), sedangkan yang kedua berjalan pada 5V (memiliki clock 16MHz). Kedua versi memiliki SRAM sebesar 2kB, EEPROM sebesar 1kB, Flash memory sebesar 32kB, dan arus DC pada tiap pin sebesar 40mA. Terdapat pin Rx-Tx maupun I2C juga untuk khusus komunikasi serial, sama seperti Arduino jenis yang



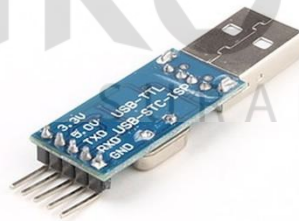
Modul WiFi ESP 8266 - 01 adalah salah satu jenis dari berbagai macam modul WiFi ESP 8266. Modul ini bisa juga berdiri sendiri seperti arduino, tetapi hanya memiliki 2 pin saja sebagai I/O. Modul sudah terintegrasi dengan protocol TCP/IP sehingga mikrokontroler (ESP 8266-01) dapat terhubung dengan jaringan WiFi. Modul ESP 8266-01 dapat diisi perintah atau program dengan cara *flashing* terlebih dahulu. Penjelasan penggunaan pin yang terdapat ESP8266-01 dapat dilihat pada gambar 2.2.



Gambar 2.3. Bentuk fisik dan *pin* ESP8266-01

2.6 USB To TTL

USB to TTL adalah *USB to serial converter* yang memungkinkan konektivitas antara USB dan *interface* UART. Terdiri dari 5 pin, yaitu pin 3.3v, Rx, Tx, GND, dan 5v. Dibutuhkan ketika akan menghubungkan antara Arduino Pro Mini dengan laptop, baik itu untuk komunikasi ataupun *upload sketches*. Gambar 2.2. adalah bentuk fisik dari USB to TTL



Gambar 2. 4. Bentuk fisik USB to TTL

2.7 UART

Komunikasi serial adalah suatu komunikasi yang mengirimkan data per bit secara bergantian dan berurutan. Untuk pengiriman data, komunikasi serial lebih lambat dibanding komunikasi paralel. Namun komunikasi serial dapat mengurangi

jumlah keluaran dan jumlah kabel yang dibutuhkan dalam komunikasinya. Komunikasi serial terdiri dari 2 macam, yaitu *serial synchronous* dan *serial asynchronous*. Pada *serial synchronous* sebelum melakukan koneksi diperlukan sinkronisasi *clock* antara pengirim dan penerima, sedangkan *serial asynchronous* tidak. *Serial asynchronous* terdiri dari transmitter (TX) dan receiver (RX) untuk kabel transmisi datanya, berbeda dengan *serial synchronous* yang terdapat di protokol SPI (*Serial Peripheral Interface*) dan I2C (*Inter-Integrated Circuit*) dimana protokolnya membutuhkan dua kabel untuk transmisi data, yaitu untuk data dan *clock*.

2.8 Buzzer

Buzzer adalah sebuah komponen yang mengubah getaran listrik menjadi getaran suara. Buzzer terdiri dari kumparan yang dipasang pada diafragma, kemudian kumparan tersebut dialiri arus dan menjadi elektromagnet, kumparan tadi tertarik kedalam atau keluar tergantung polaritas, pergerakan diafragma karena kumparan inilah yang membuat udara bergetar sehingga timbul suara. Buzzer biasa digunakan sebagai indikator atau sebagai penanda (alarm). Gambar 2.5. adalah bentuk fisik dari Buzzer.



Gambar 2.5. Bentuk fisik buzzer

2.9 Grove – Finger Clip Heart Rate Sensor

Modul sensor ini menggunakan PAH8001EI-2G, sebuah “CMOS-*process optical*” yang memiliki kinerja tinggi dan penggunaan daya yang rendah, dengan LED hijau dan DSP terintegrasi yang berfungsi sebagai sensor *heart rate*. LED hijau ditembakkan ke aliran darah, lalu memantul kembali ke optik yang nantinya pantulan tadi diproses untuk menghasilkan data detak jantung yang akurat per detik. Modul ini mengintegrasikan STM32, *reserved SWD interface* memungkinkan pengguna untuk melakukan program ulang STM32. Gambar 2.6. adalah bentuk fisik dari Grove-finger clip *heart rate sensor*.



Gambar 2.6. Bentuk fisik Grove-finger clip *heart rate sensor*

Spesifikasi :

1. Bekerja pada temperatur -20 ~ +60°C
2. Interface menggunakan I2C (SCL dan SDA)
3. Area Heart rate sensor seluas 3.0 x 4.7 mm
4. Konsumsi daya sangat rendah, masuk ke mode *power saving* ketika tidak ada interaksi dengan sensor
5. STM32F103 terintegrasi
6. *Reserved SWD Interface*

2.10 Li-po Battery YTong 702030

Baterai ini menjadi sumber tegangan yang akan digunakan pada rangkaian sistem. Baterai li-po tidak menggunakan cairan untuk elektrolitnya, melainkan menggunakan polimer elektrolit kering yang berbentuk lapisan plastik tipis. Baterai li-po memiliki kekurangan, yakni lemahnya aliran pertukaran ion yang melalui elektrolit polimer kering yang menyebabkan penurunan pada charging dan discharging rate. Tetapi baterai Li-Po lebih aman dan juga memiliki siklus hidup yang lebih panjang. Gambar 2.7. adalah bentuk fisik dari baterai Li-po Ytong 702030.



Gambar 2.7. Bentuk fisik baterai Ytong 702030

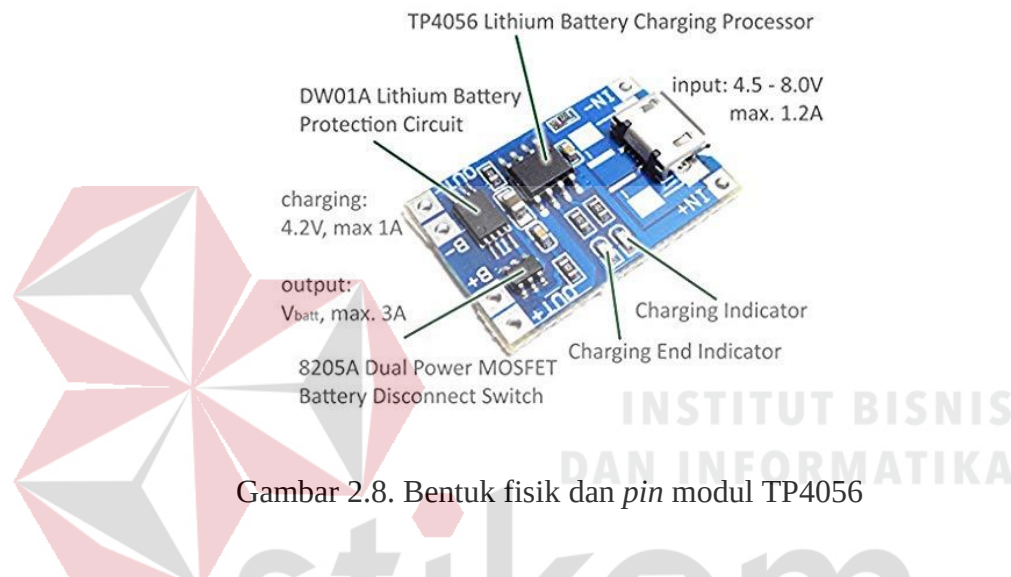
Spesifikasi baterai Li-Po YTong 702030 :

1. Tegangan : 3.7v
2. Ampere : 260mAh
3. Burst : 25°C
4. Dimensi : 30*16*8 mm
5. Daya Tahan : 6-12 Menit (tergantung penggunaan)

2.11 Battery Charger TP4056

TP4056 dapat digunakan untuk mengisi baterai jenis *Lithium*. Input menggunakan *micro-USB* (seukuran *handphone*) yang bertegangan 4.5-5v. Memiliki

2 lampu indikator, warna merah ketika sedang cas, dan warna biru ketika baterai sudah penuh. Pemasangan positif dan negatif output BAT+/- pada modul ke baterai harus benar-benar diperhatikan, karena jika pemasangan terbalik akan merusak chip. Modul bekerja pada suhu $-10^{\circ}\text{C} \sim 85^{\circ}\text{C}$. Bentuk fisik dan penjelasan penggunaan pin yang terdapat pada modul TP4056 dapat dilihat pada gambar 2.8.



Gambar 2.8. Bentuk fisik dan *pin* modul TP4056

2.12 RTC DS3231

RTC DS3231 merupakan sebuah modul untuk *real time clock* untuk pewaktu *digital*, yang memiliki IC EEPROM tipe AT24C32. *Interfaces* yang digunakan untuk mengakses modul ini menggunakan I2C (SCL dan SDA). RTC memiliki presisi waktu yang sangat akurat, dapat menghitung secara *real time* detik, menit, jam, hari, tanggal, bulan, dan tahun (hingga 2100). Memiliki berat 8g dengan dimensi 38x22x14mm. Dapat bekerja menggunakan tegangan 3,3-5,55v yang biasanya berasal dari baterai ataupun power yang dihubungkan dengan arduino langsung. Gambar 2.9. adalah bentuk fisik dari RTC DS3231.



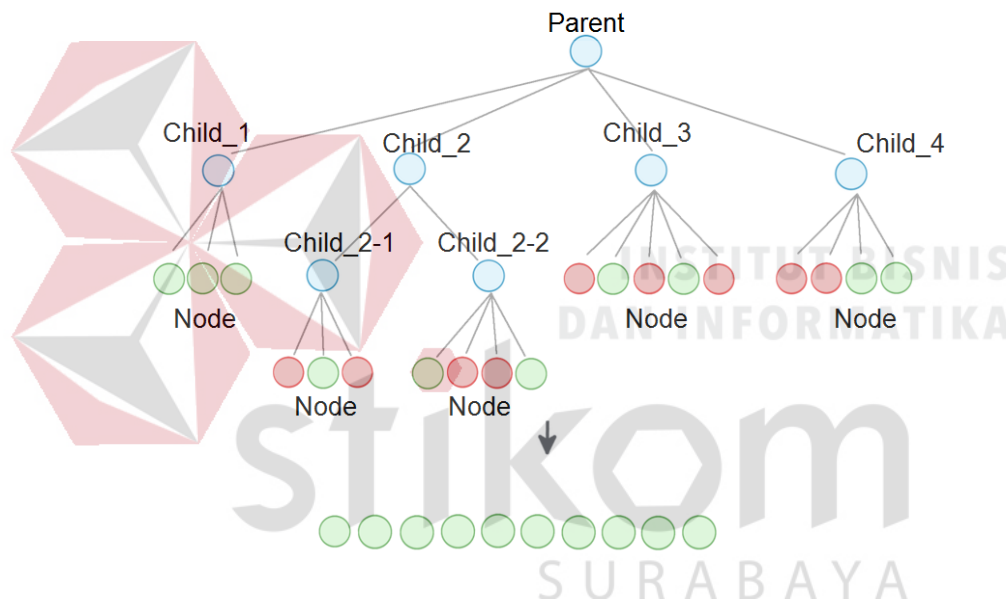
Gambar 2.9. Bentuk fisik RTC DS3231

2.13 Firebase (Database)

Firebase merupakan layanan yang disediakan oleh google dengan tujuan mempermudah pengembang aplikasi, menyediakan *database* yang dapat diakses secara *real time* sebagai *Backend as a service*. Database Firebase juga menyediakan layanan untuk sinkronisasi yang memungkinkan data pengembang aplikasi dapat terhubung dan disimpan ke Firebase Cloud untuk kepentingan CRUD (Create, Read, Update, dan Delete) maupun *remoting*, untuk mengaktifkan layanan ini dapat dilakukan dengan cara memberikan atau mendeklarasikan *library*-nya terlebih dahulu. Dapat di integrasikan dengan iOS, Android, Java, JavaScript, Node.js, dan Swift. Layanan Firebase terbagi dalam tiga pilihan, yaitu : BLAZE (bayar sesuai penggunaan), FLAME (\$25 per bulan), dan SPARK (gratis).

Firebase merupakan database non-SQL, tetapi juga dapat menyimpan data bertipe String, Long, dan Boolean. Tidak seperti database yang menggunakan SQL pada umumnya yang menggunakan tabel dan baris, penambahan data pada Firebase disimpan lalu menjadi objek (node) baru pada JSON *tree*. Penambahan data yang baru tadi akan menjadi *node* baru pada struktur JSON *tree*. *Node* dapat diibaratkan

seperti simpul-simpul yang berisi data, dan *node* tadi juga bisa memiliki cabang yang mana tiap cabang dapat memiliki data juga. Fitur *push* pada Firebase dapat digunakan untuk pengisian suatu data kedalam struktur *JSON tree* ataupun menambah *node* baru. Gambar 2.10. adalah contoh struktur dari *JSON tree*, setiap *child* memiliki parent yang sama. *Child* tadi dapat diisi *child* dalam jumlah yang banyak juga dan dapat bercabang menjadi kompleks. Di dalam *child* tersebut terdapat *node* yang berisi nilai dari data yang disimpan.



Gambar 2.10. Contoh Struktur *JSON tree*

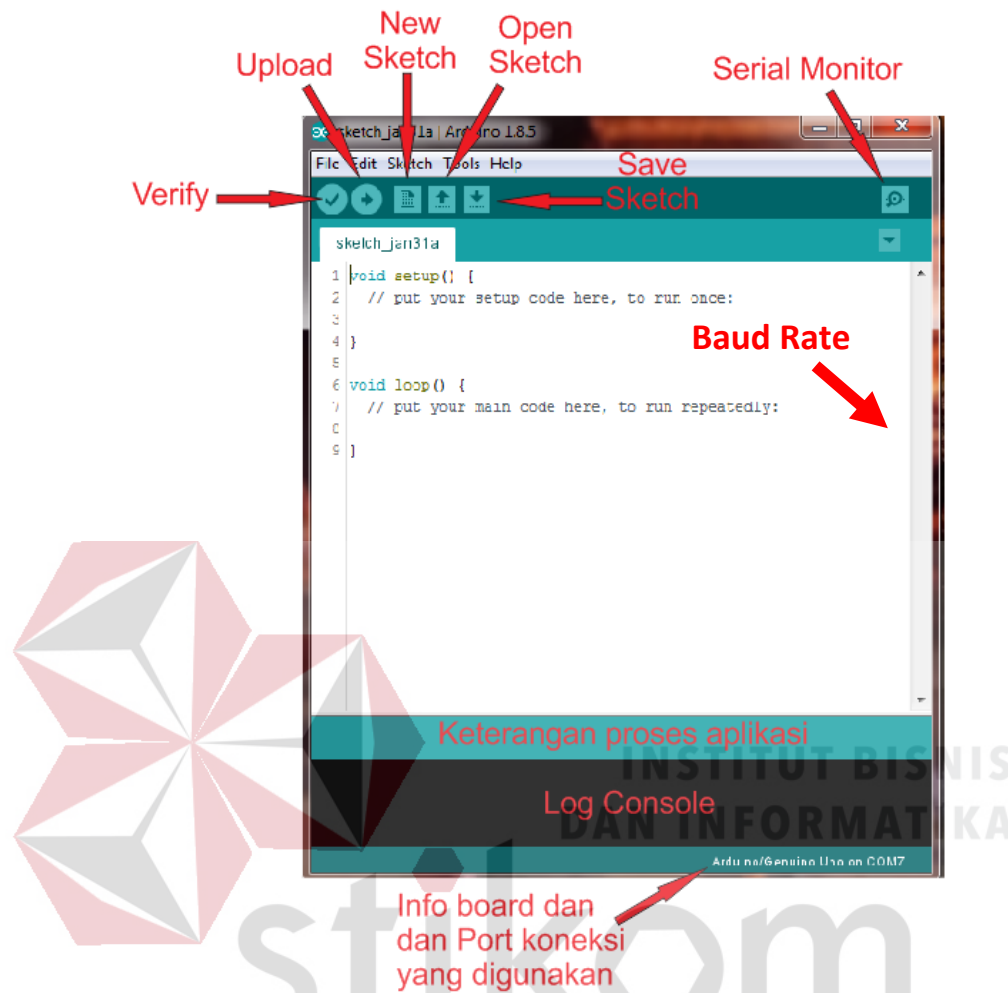
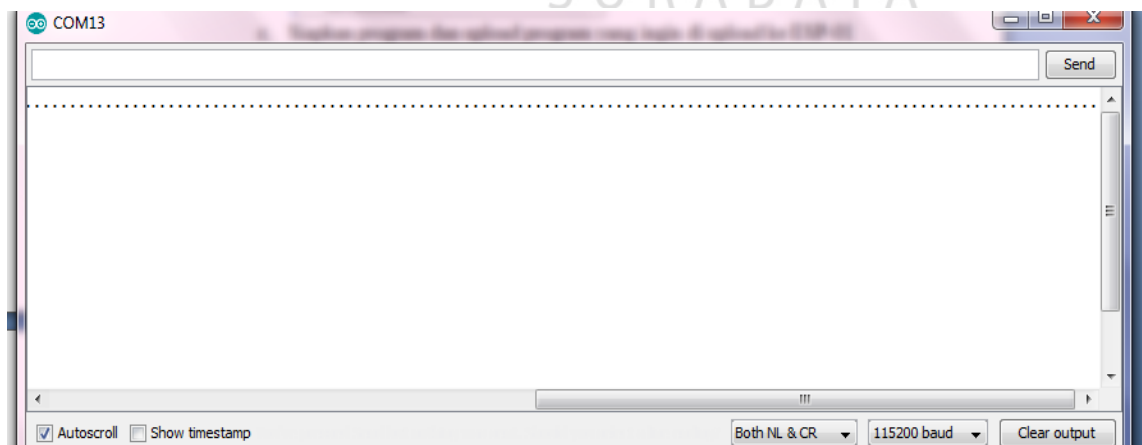
2.14 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) yang khusus untuk membangun aplikasi pada platform android. Android Studio berbasis IntelliJ IDEA. Bahasa utama yang digunakan oleh Android Studio adalah bahasa Java, sedangkan bahasa untuk membuat tampilan (*layout*) menggunakan bahasa XML. Saat ini sudah banyak ads-on support tambahan yang dapat

ditambahkan pada Android Studio, salah satunya adalah Firebase. Android Studio terintegrasi dengan Android Software Development Kit (SDK) yang digunakan untuk Debug (*deploy*) ke perangkat android (Developer, 2017), yang mana versi Android pada saat setup pertama program dan versi perangkat Android yang akan digunakan harus diperhatikan.

2.15 Arduino IDE

Arduino IDE merupakan *software* layanan dari pembuat mikrokontroler Arduino, digunakan untuk membuat program perintah yang akan di *load* ke Arduino. Aplikasi Arduino IDE menggunakan bahasa C++, penambahan komponen diluar produk arduino juga dapat dilakukan dengan memasukkan *library*nya terlebih dahulu. Kode program yang akan dimasukkan ke Arduino biasa disebut *sketch* atau *source code* yang berekstensi “.ino”. Struktur pemrogramannya terdiri dari dua bagian, yaitu *void setup* dan *void loop* seperti pada gambar 2.11. *Void setup* berisi baris yang digunakan untuk inisialisasi, dieksekusi hanya sekali saja ketika Arduino dihidupkan. *Void loop* berisi baris *coding* yang akan dieksekusi secara terus menerus selama arduino menyala. Dan untuk penambahan *library* diletakkan diatas baris *sketch* sebelum *Void setup*. Arduino IDE juga memiliki fitur *serial monitor* yang berguna untuk *debugging* program, dengan cara menyesuaikan *baudrate* pada *serial monitor* dan *baudrate* pada program yang kan dimasukkan (*upload*) ke mikrokontroler seperti yang terdapat pada gambar 2.12.

Gambar 2. 11. Tampilan *software* Arduino IDEGambar 2.12. Tampilan *serial monitor* Arduino IDE

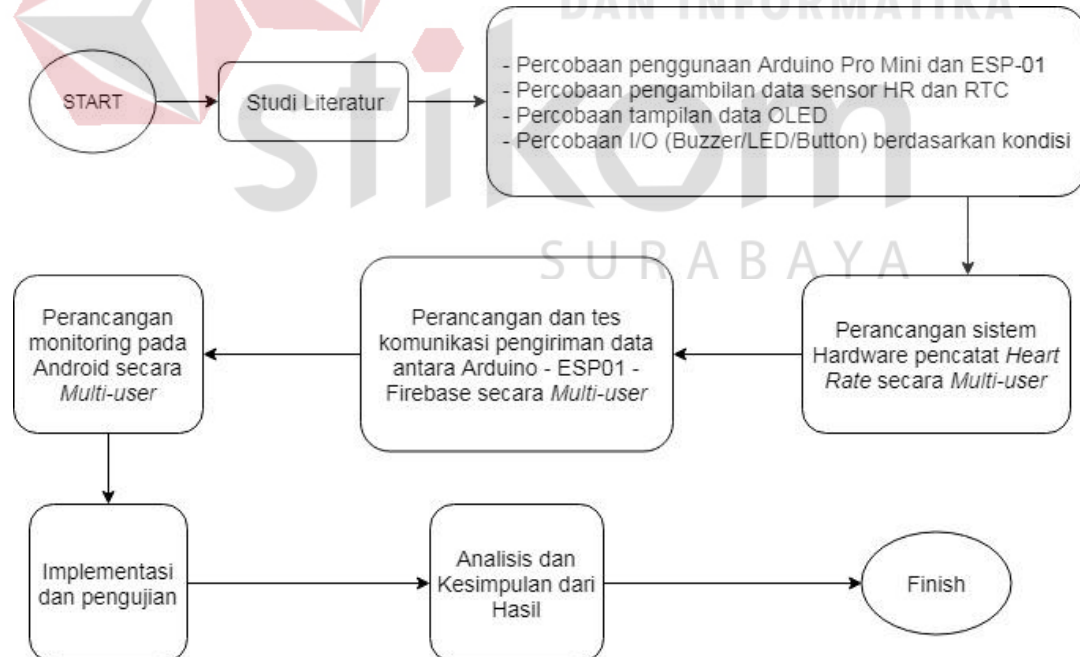
BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Metode penelitian digunakan untuk pembuatan alat, baik dari perangkat keras maupun perangkat lunak dengan melakukan studi pustaka, baik dari buku maupun internet. Alur pengerjaan penelitian terdapat pada gambar 3.1., penulis berusaha mengumpulkan data-data, informasi, dan konsep yang bersifat teori yang berhubungan dengan penelitian ini.

Dari data-data dan teori yang diperoleh maka dapat disusun rancangan perangkat serta rancangan pemrogramannya. Setelah dilakukan perancangan dilakukan integrasi dan pengujian dari seluruh rancangan yang telah dibuat.

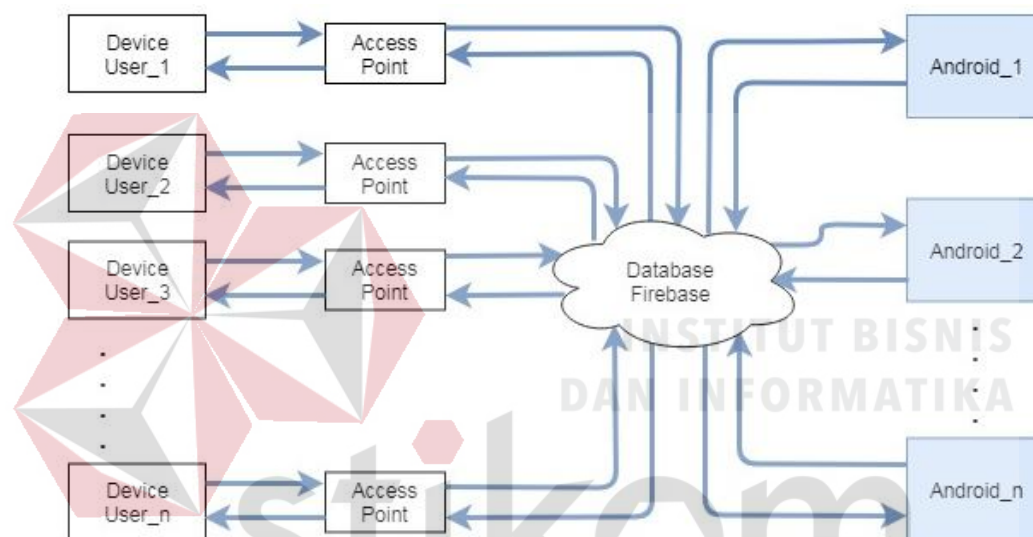


Gambar 3.1. Diagram alur penelitian

3.2 Perancangan Perangkat Keras

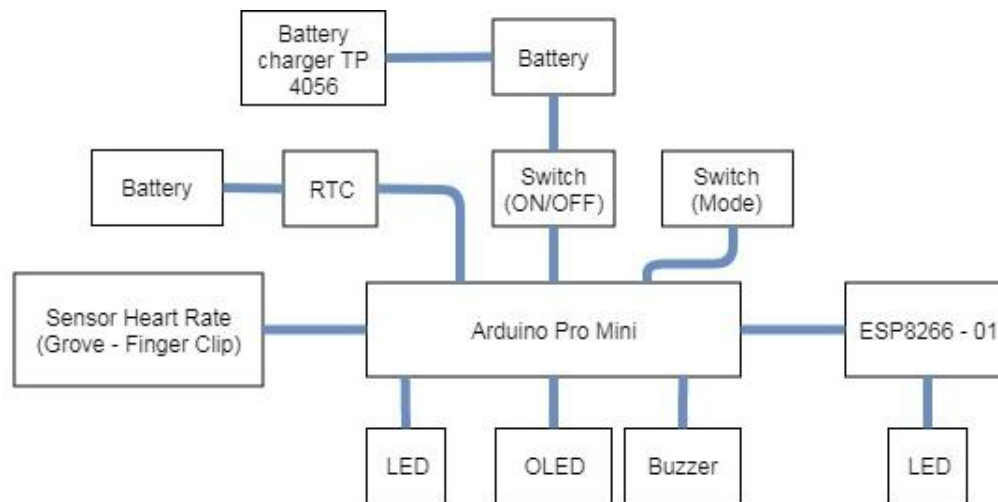
3.2.1. Rancangan *Hardware*

Konsep sistem yang akan dibuat adalah implementasi dari *Internet of Things* (IoT) yang diterapkan menggunakan mikrokontroler untuk memantau *Heart Rate* secara *multi-user*. Terdapat alat untuk pendeteksian *Heart Rate*, adapun konsep sistemnya adalah sebagai berikut :



Gambar 3.2. Diagram alur sistem

Dari gambar 3.2. diatas, data nilai *Heart Rate* (HR) yang diperoleh dari alat yang dibuat akan disimpan di Firebase secara *multi-user*, dan ditampilkan ke aplikasi Android. Input dari sistem ini adalah nilai dari sensor HR dan RTC yang dipantau secara terus menerus untuk mencapai THR. Apabila THR tercapai, maka akan terdapat tanda pada *devices* berupa suara buzzer. Data THR dihitung menggunakan metode *Karvonen*. Data disajikan berupa grafik garis secara real time, pencatatan tanggal sebelumnya juga dilakukan dan disimpan apabila sewaktu-waktu ingin melihat *history* nilai HR.



Gambar 3.3. Skema diagram blok *Hardware* per-komponen

Dari gambar 3.3., Arduino Pro Mini, ESP-01, dan RTC mendapatkan daya dari baterai yang di-charge menggunakan kabel *usb micro* melalui modul TP4056. Arduino Pro mini dan ESP-01 berjalan ketika *switch* ditekan on, sedangkan RTC terus menyala mendapatkan daya dari baterai jam. *Switch Mode* digunakan untuk pengecekan apakah user ingin mode inisialisasi (untuk pembacaan HR istirahat dan menghitung THR) atau mode *run*. Push button reset digunakan jika ada kesalahan penekanan ketika memasukkan usia. Nilai HR didapatkan dari sensor *Heart Rate* dan nilai waktu dari RTC. Data yang telah diolah Arduino Pro Mini diteruskan ke ESP-01 melalui jalur Rx-Tx, ESP-01 digunakan sebagai media pengiriman data ke *database* Firebase. Ketika ESP-01 mendapatkan IP maka LED putih akan mati. Saat nilai HR mencapai THR, LED hijau indikator menyala. Dan ketika terjadi masalah pada sensor dan RTC, suara *buzzer* akan berbunyi. Hasil dari data yang diolah ditampilkan ke OLED melalui jalur I2C. Gambar 3.4. adalah gambar rangkaian Skematik untuk gambaran keseluruhan dari komponen yang digunakan.

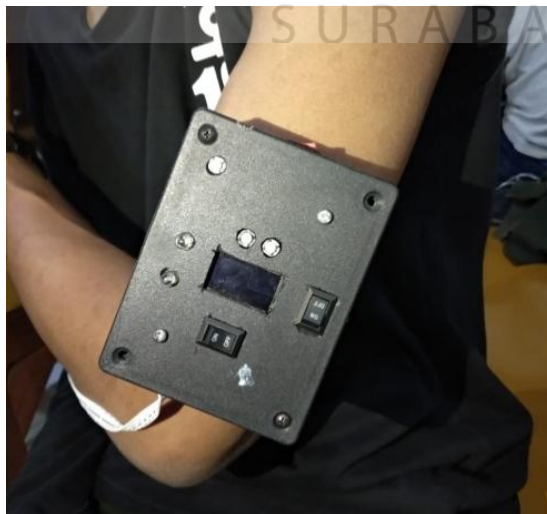
Gambar 3.4. Diagram Skematik

3.2.2. Desain Alat

Desain alat menggunakan kotak hitam plastik biasa yang beredar di pasaran, dengan dimensi 10cm x 7.5 cm x 3.5 cm (panjang x lebar x tinggi) yang dilubangi untuk akses OLED, LED, *switch/push button*, dan kabel untuk sensor HR. Sensor HR diletakkan pada jari *user* seperti pada gambar 3.5., sedangkan alat (kotak) diletakkan di lengan atas seperti *arm band* yang digunakan orang lari pada umumnya seperti pada gambar 3.6. Alat direkatkan pada lengan menggunakan *strap*.



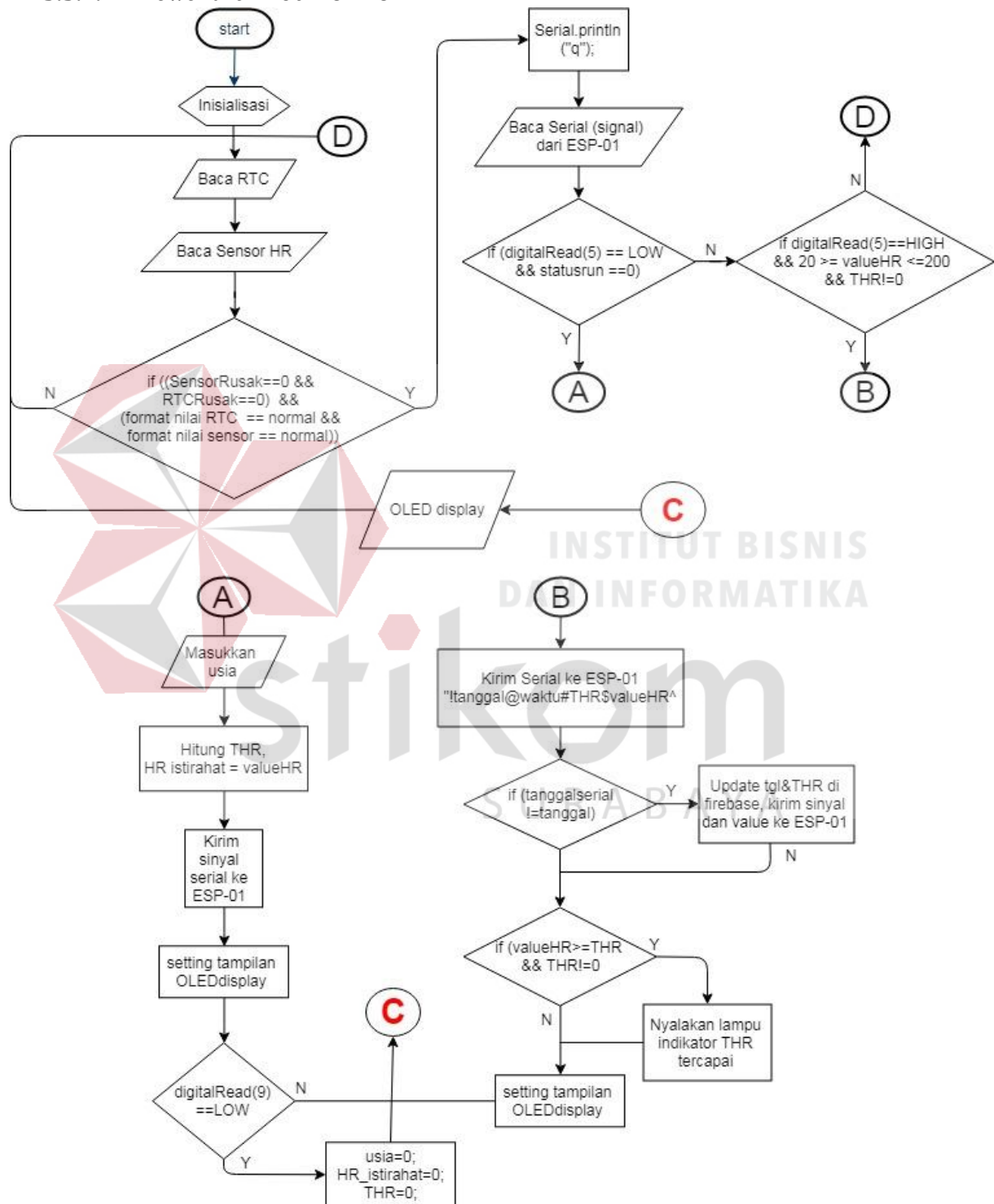
Gambar 3.5. Peletakan sensor HR pada jari



Gambar 3.6. Peletakan alat pada lengan atas

3.3 Perancangan Software

3.3.1. Flowchart Arduino Pro Mini



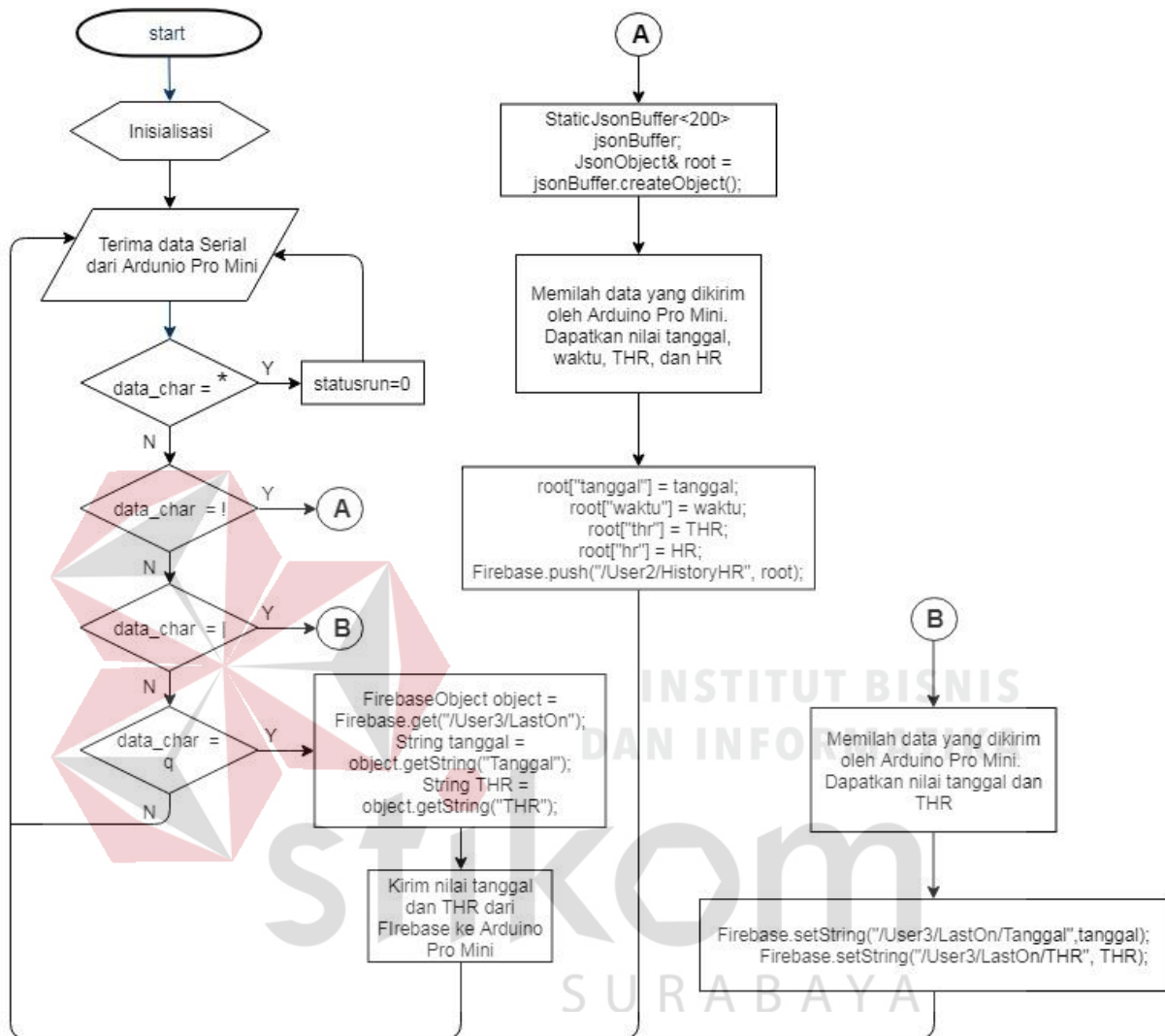
Gambar 3.7. Flowchart Arduino Pro Mini

Dari gambar 3.7., Awalnya Arduino Pro Mini melakukan inisialisasi terlebih dahulu lalu dilakukan pembacaan nilai dari sensor HR dan RTC. Jika didapat komponen bermasalah maka buzzer berbunyi, sedangkan jika tidak terdapat masalah dalam pembacaan selanjutnya Arduino Pro Mini mengirimkan serial “q” ke ESP-01. Setelah itu Arduino Pro Mini mendapatkan *feedback*, membaca serial dari data yang dikirimkan oleh ESP-01 untuk pembacaan kapan terakhir kali *device* menyala (tidak menggunakan EEPROM). Jika tanggal yang diterima dari ESP-01 tidak sama seperti dengan tanggal pada RTC maka nilai variabel *statusrun* bernilai 0, sedangkan jika sama maka *statusrun* bernilai 1. Hal ini digunakan untuk menghindari masalah jika tiba-tiba *user* mematikan *device*, agar nilai THR tidak berubah-ubah.

Masuk ke *mode* inisialisasi, *User* memasukkan usia dan kemudian dilakukan perhitungan nilai THR berdasarkan nilai HR istirahat sebelum *user* berolahraga. Setelah itu Arduino Pro Mini mengirimkan sinyal bahwa sedang di *mode* inisialisasi, kemudian OLED menampilkan nilai dari data yang telah diolah. Jika ada kesalahan *setting* maka penekanan tombol *reset* akan membuat semua nilai menjadi nol.

Masuk *mode run*, Arduino Pro Mini mengirimkan data *serial* untuk nilai tanggal, waktu, THR, dan HR ke ESP-01. Kemudian dilakukan pengecekan, apakah tanggal yang dikirimkan via komunikasi *serial* oleh ESP-01 tadi sama dengan tanggal nilai RTC, jika tidak sama maka akan dilakukan *update* pada *child* Last_On *node* tanggal dan THR pada Firebase. Arduino Pro Mini melakukan pengiriman sinyal untuk melakukan *update*. Jika nilai THR tercapai maka LED hijau akan menyala, hasil Run ditampilkan pada OLED.

3.3.2. Flowchart ESP8266-01



Gambar 3.8. Flowchart ESP-01

Gambar 3.8. adalah alur program dari ESP-01. ESP-01 digunakan untuk mengirimkan data dari Arduino Pro Mini ke Firebase (sebagai perantara). Awalnya ESP-01 menerima sinyal dari Arduino Pro mini apakah sedang *mode run*, *mode inisialisasi*, atau *mode update*.

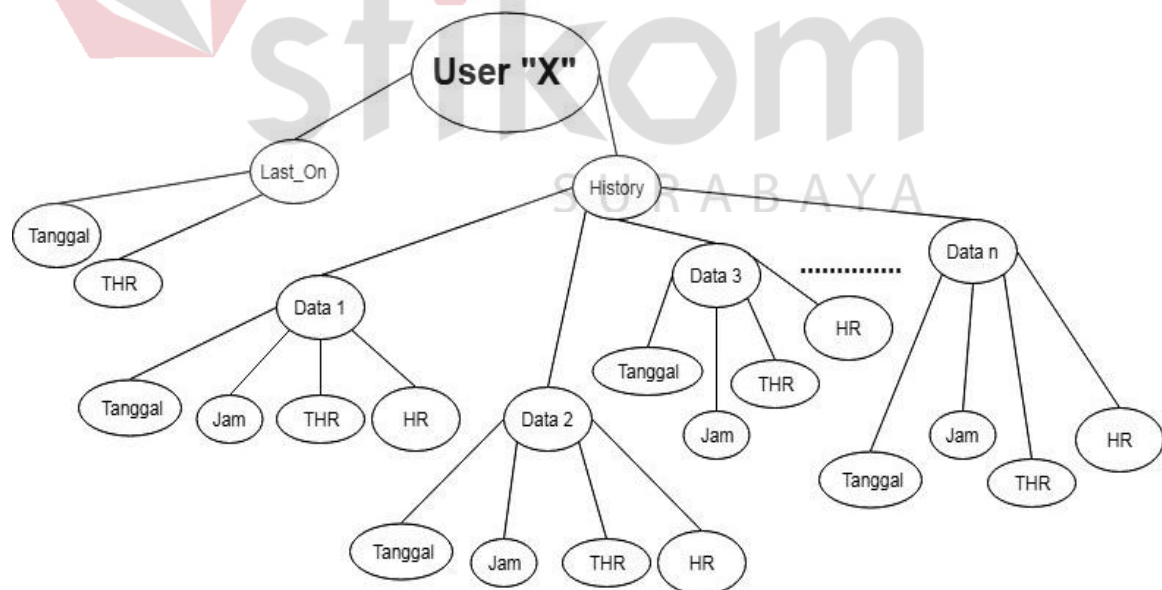
Ketika didapat sinyal mode init “*” *status run* menjadi 0, sedangkan ketika didapat sinyal mode run “!” *status run* menjadi 1. Ketika *status run* bernilai 1, ESP-

01 melakukan deklarasi membuat JSON *object* untuk JSONbuffer (digunakan untuk *buffer* data yang dikirimkan ke Firebase). ESP-01 akan menerima variabel tanggal, waktu, THR, dan HR. Kemudian variabel tersebut akan dikirim ke Firebase membentuk *node* data dan *child* baru dengan fitur `firebase.push("/data_path".isi_data)`.

Ketika didapat sinyal “q”, ESP-01 mengambil nilai data dari *node* Last_On untuk mendapatkan nilai THR dan tanggal kapan terakhir kali *device* dinyalakan. Setelah didapat hasil dikirimkan ke Arduino Pro Mini untuk penguncian tanggal.

Ketika didapat sinyal update “|”, ESP-01 menerima data tanggal dan THR kemudian data tersebut dimasukkan ke *node* Last-ON dan *child* tanggal THR untuk *update* bahwa *device* sudah dinyalakan pada tanggal yang berbeda.

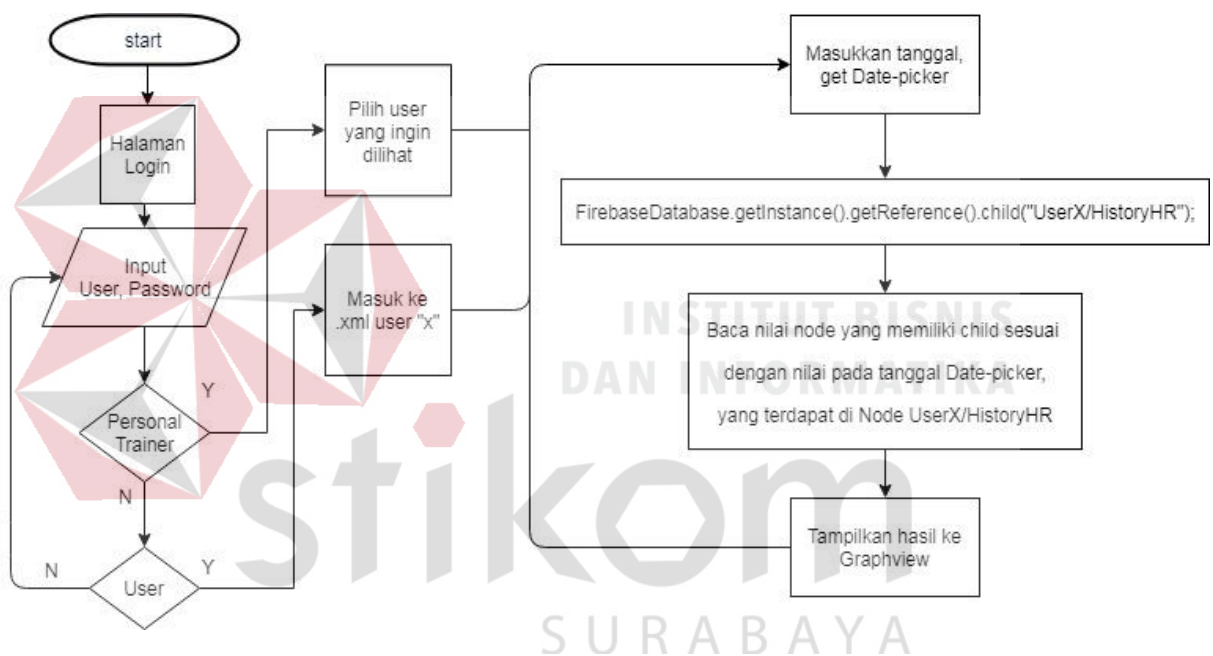
3.3.3. JSON Tree Firebase



Gambar 3.9. Struktur JSON tree Firebase

Dari gambar 3.9., user ‘X’ adalah *child* awal dari data *user*. Dari *child* user ‘X’ terbagi lagi menjadi *child* Last_On dan History. *Child* Last_On berisi *node* yang akan digunakan untuk menyimpan kapan terakhir kali *user* pernah menyala. *Child* History berisi data yang di *push* terus menerus oleh *device*, yang mana didalam *child* data baru tersebut terdapat *node* data yang berisikan tanggal, jam, THR, dan HR.

3.3.4. Flowchart Aplikasi Android



Gambar 3.10. Flowchart aplikasi Android

Dari gambar 3.10., aplikasi masuk ke halaman *login* terlebih dahulu, kemudian memasukkan *username* dan *password*. Jika *user* maka langsung mengakses ke *.xml history user*, Jika PT maka mengarah ke halaman pilih *user* terlebih dahulu. Penampilan data dilakukan dengan memilih tanggal berapakah data yang ingin diambil. Setelah itu Android melakukan pemanggilan dan *get* data pada Firebase, hasil ditampilkan pada Graphview.

BAB IV

HASIL PENGUJIAN DAN PENGAMATAN

Pada bab ini penulis akan memaparkan hasil dari analisis pengujian pada penelitian yang telah dilakukan. Adapun tahap yang dilakukan pada penelitian kali ini, yaitu pengujian perangkat keras (*hardware*) per komponen dan juga perangkat lunak (*software*). Adapun *output* yang diinginkan dari pengujian perangkat keras secara keseluruhan adalah history HR dan THR yang disimpan di Firebase, secara *multi-user*, dan akurasi pengiriman data secara *realtime* berdasarkan waktu dari RTC. Kemudian, pembacaan sensor HR dibandingkan dengan alat pembaca *Blood Pressure Monitor* milik OMRON untuk pengujian akurasi dari sensor yang digunakan.

4.1. Pengujian Upload Program Mikrokontroler Arduino Pro Mini

4.1.1. Tujuan

Tujuan dilakukan pengujian upload program pada Arduino Pro Mini karena cara uploadnya berbeda dengan Arduino Uno yang bisa langsung terupload.

4.1.2. Peralatan yang digunakan

1. Arduino Pro Mini
2. USB to TTL
3. Kabel jumper female-female
4. Arduino IDE

4.1.3. Cara Pengujian

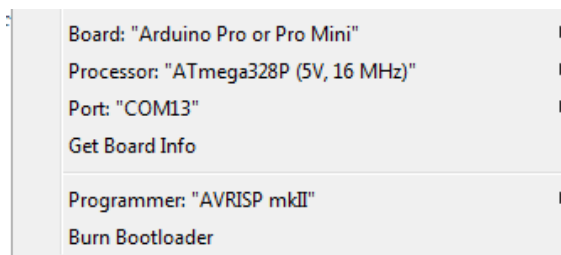
- a. Hidupkan laptop dan buka aplikasi Arduino IDE, siapkan program yang akan di upload (pada percobaan ini akan di upload program untuk menyalakan pin D12). Upload program berikut :

```
void setup () {
    pinMode(12, OUTPUT);
}
void loop () {
    digitalWrite (12, LOW);
}
```

- b. Hubungkan pin USB to TTL ke Arduino Pro Mini ke pin-pin dengan urutan sebagai berikut :

1. 5v USB to TTL -> VCC Arduino Pro Mini
2. GND USB to TTL -> GND Arduino Pro Mini
3. Rx USB to TTL -> Tx Arduino Pro Mini
4. Tx USB to TTL -> Rx Arduino Pro Mini

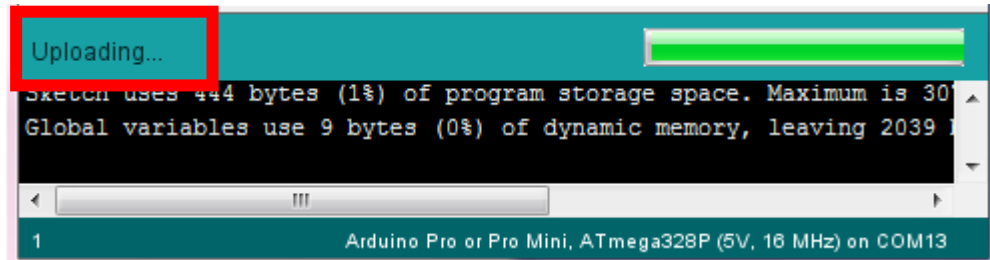
- c. Ubah jenis board pada menu Tools -> Board -> Arduino Pro or Pro Mini, pilih jenis processornya juga (pada percobaan ini menggunakan ATmega328P (5v, 16MHz)). Ubah juga Port pilih COM yang tersedia seperti pada gambar 4.1.



Gambar 4.1. Setting pada menu Tools untuk Arduino Pro Mini

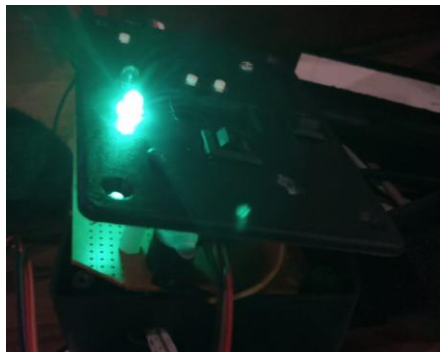
- d. Upload Program, tunggu hingga proses compiling selesai.

- e. Ketika pesan status “Compiling sketch...” berubah menjadi “uploading..” seperti pada gambar 4.2., dan LED pada USB to TTL berkedip, tekan tombol reset pada Arduino Pro Mini. Tunggu sebentar hingga LED berkedip selesai.



Gambar 4.2. Pesan status pada Arduino IDE

- f. Jika sukses maka pesan status akan berubah menjadi “Done Uploading”, jika gagal maka akan stuck di “Uploading..”. maka yang harus dilakukan adalah dengan cara cabut USB to TTL dari laptop dan upload kembali program, kembali ke langkah (c). jika masih bermasalah restart laptop anda.
- g. Ketika upload sukses, pin D12 menyala sehingga LED menyala seperti pada gambar 4.3. Hal ini terjadi karena coding pada void setup mendeklarasikan pin D12 sebagai output, yang kemudian pin D12 tadi diberi nilai high. *Script* dari Arduino IDE sudah masuk ke *microcontroller* Arduino Pro Mini.



Gambar 4.3. Hasil tes upload program

4.2. Pengujian Upload Program ESP8266-01

4.2.1. Tujuan

Tujuan dilakukan pengujian upload program pada ESP8266-01 ini agar pembaca mengetahui cara upload program langsung *flash* menggunakan USB to TTL, karena untuk uploadnya ada beberapa cara yang kurang tepat di internet dan bahkan susah dipahami.

4.2.2. Peralatan yang digunakan

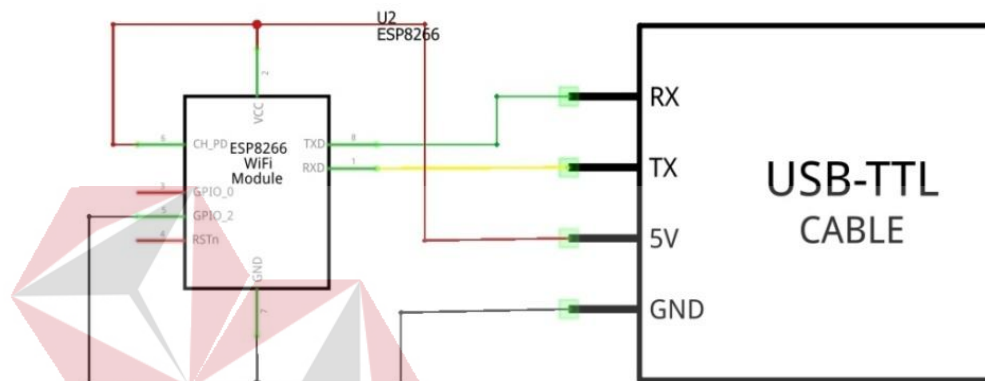
1. ESP8266-01
2. USB to TTL
3. Header sisir male
4. Socket header female
5. Jumper female-female
6. Timah
7. Solder
8. PCB lubang (kecil)
9. Arduino IDE
10. Laptop / PC

4.2.3. Cara pengujian

- a. Siapkan rangkaian dan hubungkan sesuai gambar 4.4 dengan urutan sebagai berikut :
 1. CH_PD dan VCC ESP-01 -> 3.3v USB to TTL
 2. Tx Esp-01 -> Rx USB to TTL

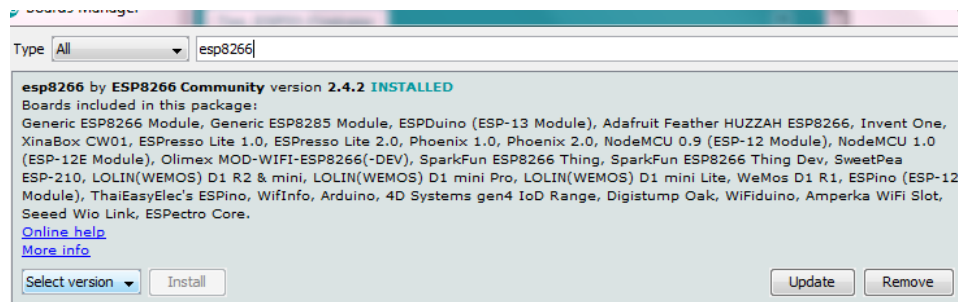
3. Rx Esp-01 -> Tx USB to TTL
4. GPIO_2 dan GND Esp-01 -> GND USB to TTL

(Lepas GPIO_2 dengan GND ketika program sudah selesai terupload, jika tidak dilepas ketika ESP-01 *lost power* dan nyala kembali, program yang terupload tadi akan hilang).



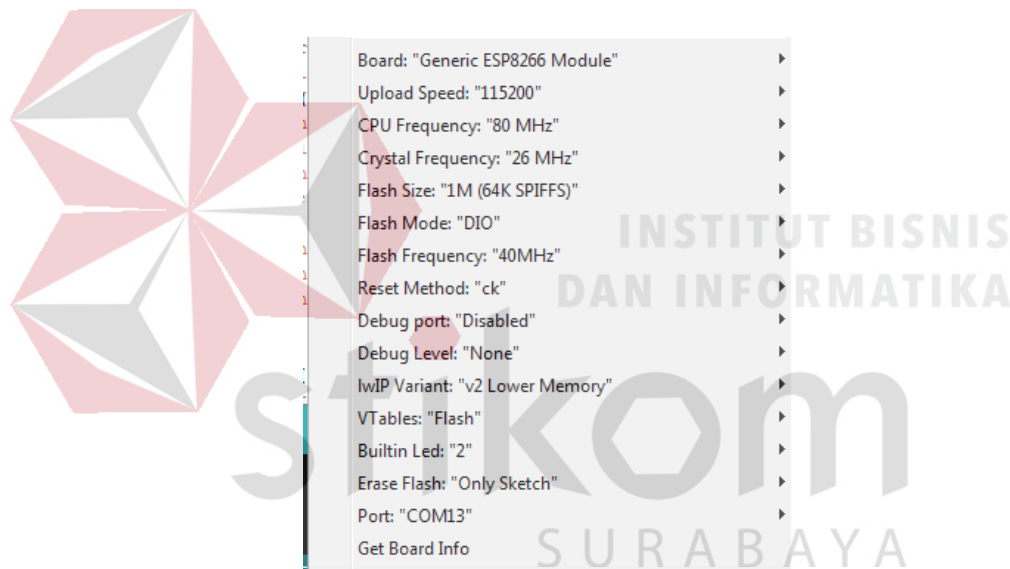
Gambar 4.4. Rangkaian upload program ESP-01 dengan USB to TTL

- b. Buka Arduino IDE, cek apakah ada board Generic ESP8266 pada menu Tools -> Board -> Generic ESP8266 Module.
- c. Jika ada lanjut ke step (e), Jika tidak ada maka buka file -> Preferences -> Additional Boards Manager URLs -> masukkan link :
https://arduino.esp8266.com/stable/package_esp8266com_index.json
- d. Buka menu Tools -> Board -> Board Manager -> ketikkan ESP8266, pilih versi -> install. Adapun contoh menu yang dibuka seperti pada gambar 4.5.



Gambar 4.5. Download boards ESP8266

- e. Kemudian ikuti setting pada menu Tools pada gambar 4.7., ubah juga Port nya (pilih COM yang tersedia) seperti pada gambar 4.6.



Gambar 4.6. Setting pada menu Tools untuk ESP-01

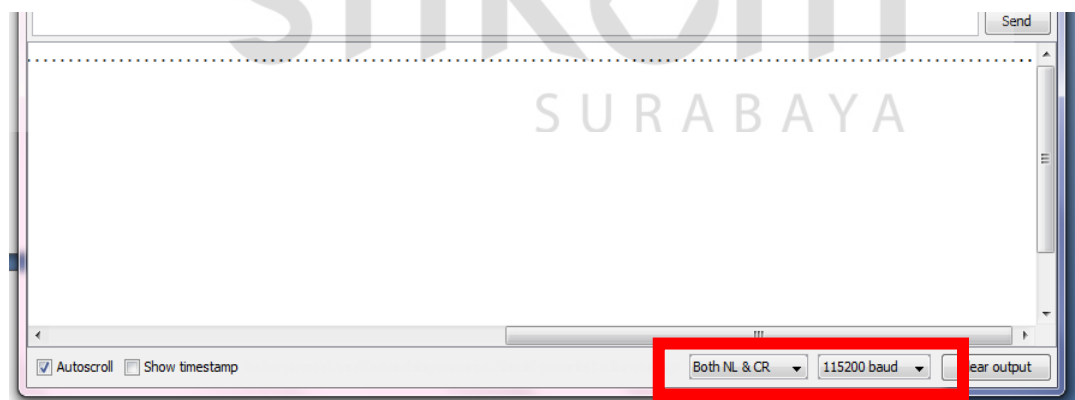
- f. Siapkan program dan upload program yang ingin di upload ke ESP-01 (pada percobaan ini dilakukan upload program tes untuk koneksi ke *access point*, alamat IP didapatkan secara dinamis). Upload program sebagai berikut :

```

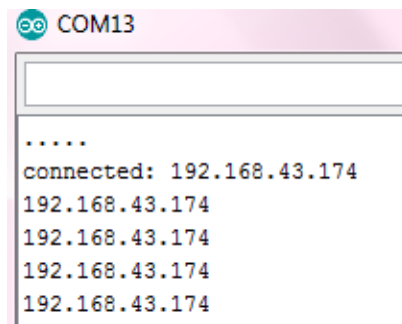
#include <ESP8266WiFi.h>
#define WIFI_SSID "Freyja" ///Nama Wifi
#define WIFI_PASSWORD "UtaMacross" ///Password Wifi
void setup() {
    Serial.begin(115200);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("connecting");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }
}
void loop() {
    Serial.println(WiFi.localIP());
    delay(1000);
}

```

- g. Ubah *baudrate* menjadi 115200 (menyesuaikan program) dan Both NL&CR.
- h. Lihat pada gambar 4.8., ketika sukses terhubung dengan *access point* maka ESP-01 mengirimkan string menampilkan alamat IP yang didapat, jika tidak maka hanya akan ditampilkan string “.” seperti pada gambar 4.7. Dari hasil percobaan, program yang di upload ke ESP-01 berhasil dan berjalan dengan baik. ESP-01 mendapatkan alamat IP secara dinamis.



Gambar 4.7. Setting serial monitor untuk memantau hasil tes



Gambar 4.8. ESP-01 sudah terhubung dengan *access point*

4.3. Pengujian Tombol, LED, dan Buzzer

4.3.1. Tujuan

Tujuan dilakukan pengujian ini agar pembaca mengetahui cara penggunaan push / switch button sebagai input, dan LED / Buzzer sebagai output menggunakan pin digital.

4.3.2. Peralatan yang digunakan

1. Arduino Pro Mini
2. USB to TTL
3. Push Button
4. Switch
5. LED
6. Buzzer 5v (kecil)
7. Arduino IDE

4.3.3. Cara pengujian

- a. Siapkan komponen dan hubungkan dengan urutan berikut :

D dan Buzzer terhubung der

D dan Buzzer terhubung der

- b. Upload program berikut untuk tes percobaan proses input menggunakan switch dan push button, output menggunakan LED dan Buzzer pada pin digital.

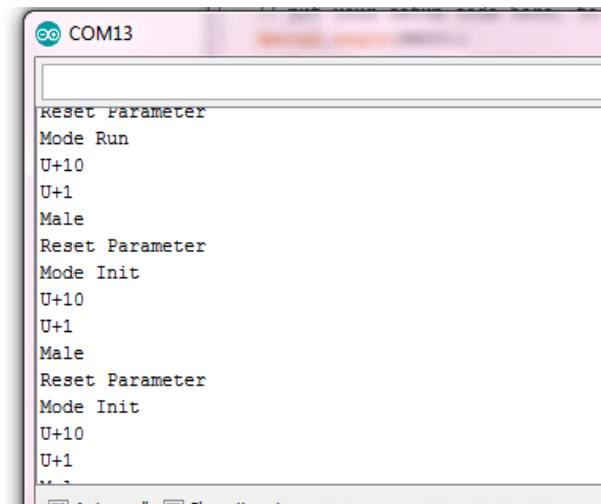

```

void setup() {
  Serial.begin(9600);
  pinMode(5, INPUT); ///Switch init atau Run
  pinMode(6, INPUT); ///Push Button Usia +10
  pinMode(7, INPUT); ///Push Button Usia +1
  pinMode(8, INPUT); ///Switch Jenis Kelamin
  pinMode(9, INPUT); ///Push Button Reset Settingan
  pinMode(10, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  if (digitalRead(5) == HIGH) {
    Serial.println("Mode Run");
  }
  else if (digitalRead(5) == LOW) {
    Serial.println("Mode Init");
  }
  if (digitalRead(6) == LOW) {
    Serial.println("U+10");
  }
  if (digitalRead(7) == LOW) {
    Serial.println("U+1");
  }
  if (digitalRead(8) == HIGH) {
    Serial.println("Male");
    digitalWrite(12, HIGH);
  }
  else if (digitalRead(8) == LOW) {
    Serial.println("Female");
    digitalWrite(13, HIGH);
  }
  if (digitalRead(9) == LOW) {
    Serial.println("Reset Parameter");
  }
  delay(500);
}

```

- c. Ketika rangkaian dioperasikan, tampilan *serial print* yang ditampilkan pada gambar 4.10. sesuai dengan kondisi *input* yang ditekan pada rangkaian. LED menyala seperti pada gambar 4.11. ketika *switch* pada pin *digital* 8 bernilai HIGH. Rangkaian pada gambar 4.9. dan program yang di *upload* pada Arduino Pro Mini berhasil dijalankan.



Gambar 4.10. Hasil serial monitor tes switch/push button



Gambar 4.11. LED menyala pada male mode

4.4. Pengujian RTC DS3231

4.4.1. Tujuan

Tujuan dilakukan pengujian ini agar pembaca mengetahui bagaimana cara set waktu dan melakukan pembacaan waktu pada RTC DS3231 menggunakan jalur komunikasi I2C pada Arduino Pro Mini.

4.4.2. Peralatan yang digunakan

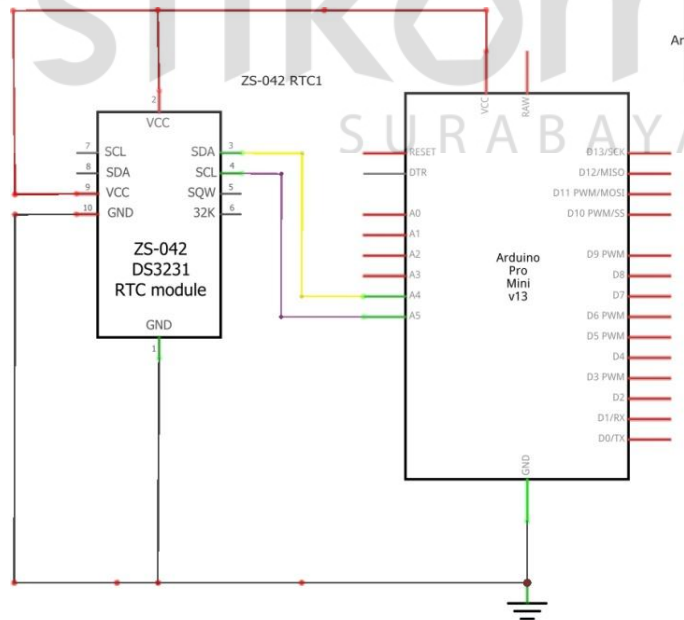
1. RTC DS3231
2. Arduino Pro Mini
3. USB to TTL
4. Arduino IDE

4.4.3. Cara pengujian

- a. Siapkan komponen dan hubungkan rangkaian seperti pada gambar 4.12.

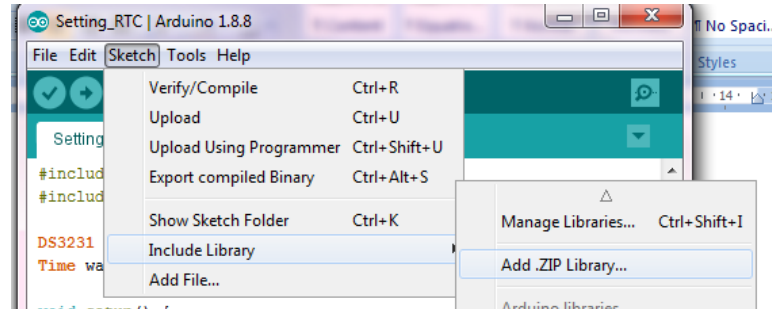
dengan urutan berikut :

1. Sambungkan USB to TTL ke Arduino Pro Mini (langkah sub bab 4.1).
2. SCL DS3231 -> A5 Arduino Pro Mini
3. SDA DS3231 -> A4 Arduino Pro Mini
4. VCC DS3231 -> VCC Arduino Pro Mini
5. GND DS3231 -> GND Arduino Pro Mini



Gambar 4.12. Rangkaian Arduino Pro Mini dan RTC DS3231

- b. Tambahkan driver RTC DS3231 (download di google). Buka menu Sketch -> Include Library -> Add .ZIP Library -> Pilih file .ZIP yang sudah di download seperti pada gambar 4.13.

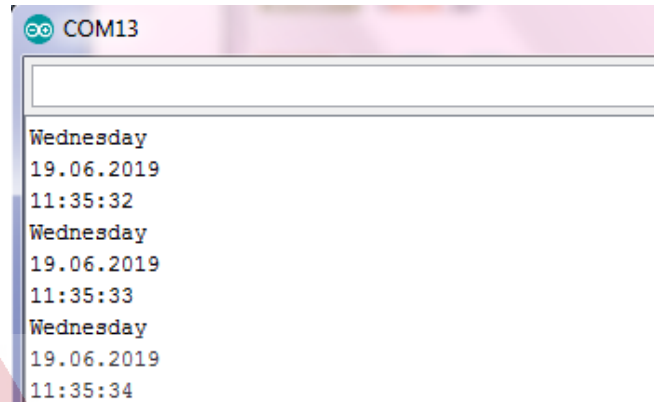


Gambar 4.13. Cara memasukkan library

- c. Tes percobaan cara pembacaan dan *setting data* waktu pada RTC DS3231, jangan lupa sertakan headernya RTC DS3231. Upload program sebagai berikut :

```
#include <DS3231.h>
#include <Wire.h>
DS3231 rtc(SDA, SCL);
Time waktu;
void setup() {
  Serial.begin(9600);
  Wire.begin();
  rtc.begin();
  rtc.setDOW(WEDNESDAY); // Set Hari
  rtc.setTime(5, 8, 9); // JJ:MM:DD (24hr format)
  rtc.setDate(9, 6, 2019); // Set tanggal dd/mm/yy
}
void loop() {
  Serial.println(rtc.getDOWStr(FORMAT_LONG));
  Serial.println(rtc.getDateStr(FORMAT_LONG));
  waktu = rtc.getTime();
  int dataJam = waktu.hour;
  int dataMenit = waktu.min;
  int dataDetik = waktu.sec;
  Serial.print(dataJam);
  Serial.print(':');
  Serial.print(dataMenit);
  Serial.print(':');
  Serial.println(dataDetik);
  delay(1000);
}
```

- d. Pada gambar 4.14., cek *serial monitor* didapat hasil nilai dari RTC. Dapat disimpulkan bahwa nilai tanggal dan waktu telah masuk ke RTC dan menampilkan data dari RTC telah berhasil.



Gambar 4.14. Hasil output data RTC serial monitor

4.5. Pengujian OLED 128x64

4.5.1. Tujuan

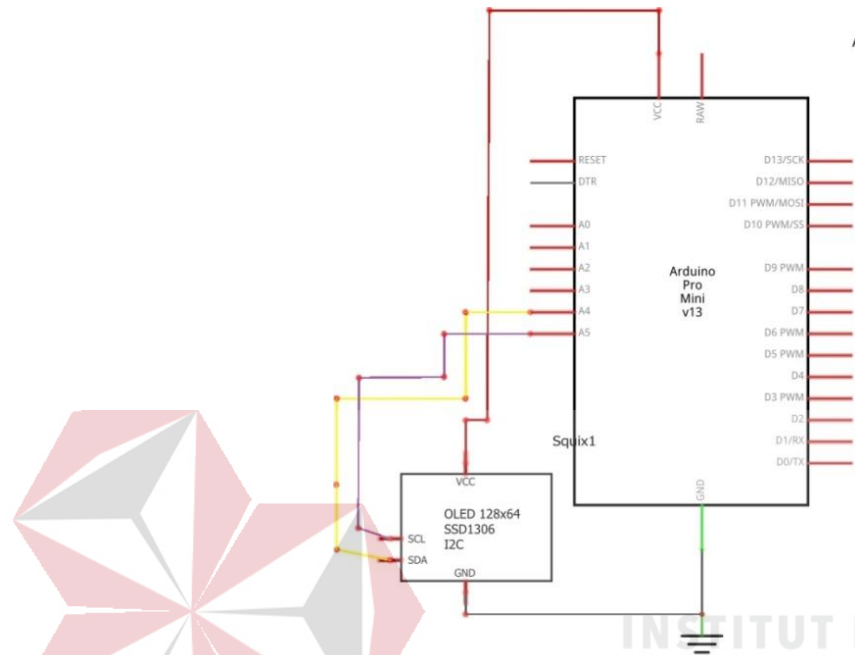
Tujuan dilakukan pengujian ini agar pembaca mengetahui cara menggunakan display OLED 128x64 sebagai output untuk menampilkan nilai dari sensor, RTC, dan status running program. Menggunakan komunikasi I2C juga pada penggunaannya.

4.5.2. Peralatan yang digunakan

1. Arduino Pro Mini
2. USB to TTL
3. Kabel jumper female-female
4. Adafruit OLED 128x64
5. Arduino IDE

4.5.3. Cara pengujian

- a. Hubungkan rangkaian seperti pada gambar 4.15. dengan urutan berikut :



Gambar 4.15. Rangkaian Arduino Pro Mini dan Adafruit OLED 128x64

1. Sambungkan USB to TTL dengan Arduino Pro Mini (sesuai pada langkah sub bab 4.1).
 2. SCL DS3231 -> A5 Arduino Pro Mini
 3. SDA DS321 -> A4 Arduino Pro Mini
 4. VCC DS3231 -> VCC Arduino Pro Mini
 5. GND DS3231 -> GND Arduino Pro Mini
- b. Tambahkan library Adafruit OLED128x64 (download di google) dengan cara yang sama seperti pada langkah (b) pada sub bab 4.4.3.
- c. Upload program untuk tes percobaan tulisan yang akan ditampilkan ke display OLED 128x64.

```

#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <splash.h>
#include <Wire.h>
#define OLED_ADDR 0x3C
///Resolusi 128*64
Adafruit_SSD1306 display(-1);
//Adafruit_SSD1306(int8_t RST = -1);
String jam="12",menit="56",detik="43";
int valueHR=85, HR_istirahat=68;
int usia=20;
float miliss;
float THR=168.89;

void setup() {
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
}

void tampilwaktu() {
  ///tanggal jam menit miliss
  for(int x=39;x<=93;x++){
    display.drawPixel(x,10,WHITE);
    display.drawPixel(x,0,WHITE);
  }
  for(int y=0;y<=10;y++) {
    display.drawPixel(38,y,WHITE);
    display.drawPixel(93,y,WHITE);
  }
  display.setCursor(42,2);
  display.print(jam);
  display.print(":");
  display.print(menit);
  display.print(":");
  display.print(detik);
}

void tampilusia(){
  display.setCursor(8,13);
  display.print("Usia");
  display.setCursor(13,23);
  display.print(usia);
}

void tampilHR() {
  display.setCursor(58,13);
  display.print("HR");
  display.setCursor(58,23);
  display.print(valueHR);
}

```

```

void tampilTHR(){
    display.setCursor(100,13);
    display.print("THR");
    display.setCursor(90,23);
    display.print(THR);
}

void tampilmoderun() {
    display.setCursor(120,2);
    display.print("R");
}

void milisstampil(){
    display.setCursor(2,2);
    display.print(int(miliss/1000));
    display.print("s");
}

void loop() {
    display.clearDisplay();
    display.setTextSize(1); //tinggi=7px
    display.setTextColor(WHITE);
    display.clearDisplay();
    tampilmoderun();
    milisstampil();
    tampilTHR();
    tampilwaktu();
    tampilusia();
    tampilHR();
    display.display();
    delay(200);
    miliss = miliss+200;
}

```

- d. Dari gambar 4.16., OLED berhasil menampilkan *string* yang terdapat pada *script*. Penataan tampilan pada OLED dapat dilakukan dengan cara mengatur koordinat-koordinat (*pixel*) dimanakah *string* tersebut akan ditampilkan.



Gambar 4.16. Hasil display OLED 128x64

4.6. Pengujian Grove – Finger Clip Heart Rate Sensor

4.6.1. Tujuan

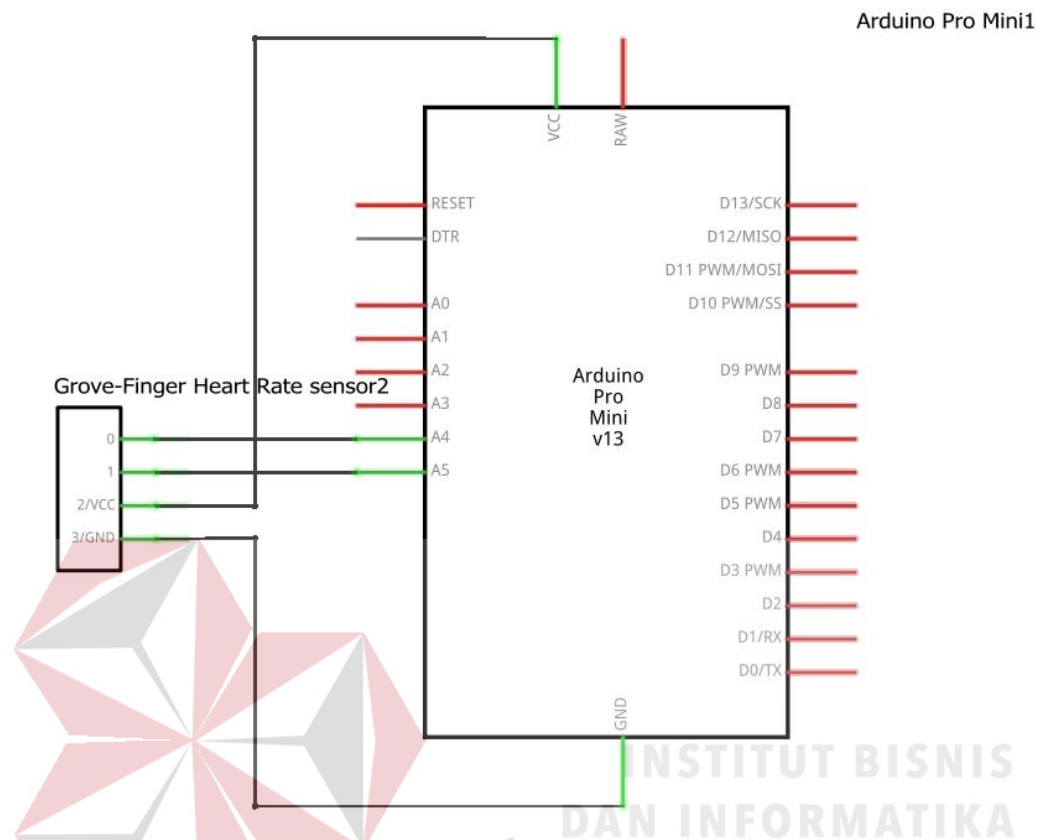
Tujuan dilakukan pengujian ini agar pembaca mengetahui bagaimana cara menggunakan Grove – Finger Clip Heart Rate Sensor menggunakan software Arduino IDE. Komunikasi yang digunakan adalah komunikasi jalur I2C yang telah disediakan pada Arduino Pro Mini.

4.6.2. Peralatan yang digunakan

1. Arduino Pro Mini
2. USB to TTL
3. Kabel jumper male-female
4. Grove – Finger Clip Heart Rate Sensor
5. Arduino IDE
6. Laptop/PC

4.6.3. Cara pengujian

- a. Hubungkan rangkaian seperti pada gambar 4.17. dengan urutan berikut :
 1. Sambungkan USB to TTL dengan Arduino Pro Mini (sesuai pada langkah sub bab 4.1).
 2. SCL sensor HR -> A5 Arduino Pro Mini
 3. SDA sensor HR -> A4 Arduino Pro Mini
 4. VCC sensor HR -> VCC Arduino Pro Mini
 5. GND sensor HR -> GND Arduino Pro Mini



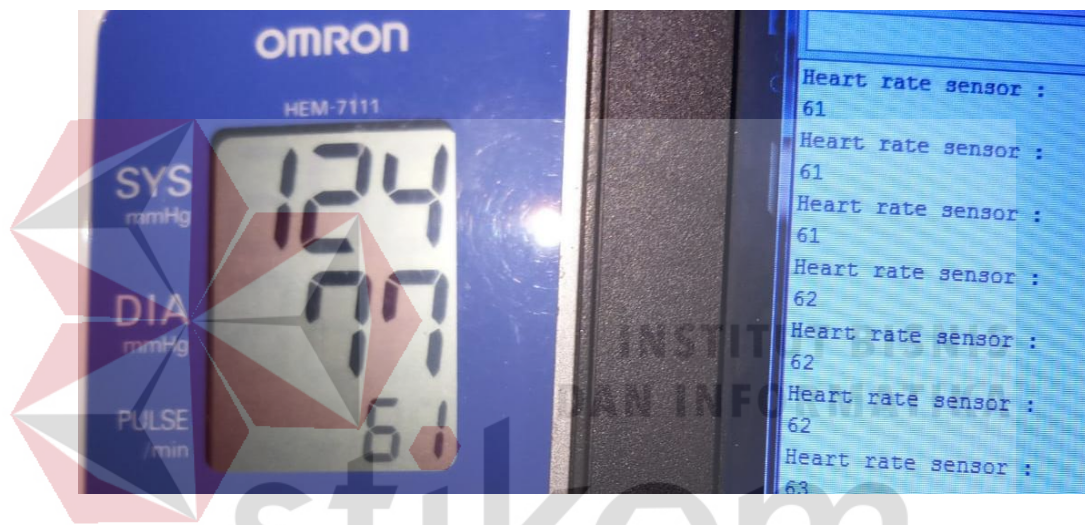
Gambar 4.17. Rangkaian Arduino Pro Mini dan Sensor HR

- b. Upload program untuk tes percobaan membaca nilai sensor *heart rate* pada Arduino Pro Mini. Upload program sebagai berikut :

```
#include <Wire.h>
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Wire.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("Heart rate sensor :");
  Wire.requestFrom(0xA0 >> 1, 1);    // request 1 bytes, slave
  while(Wire.available()) {
    unsigned char c = Wire.read();    // receive HR value(byte)
    Serial.println(c, DEC);
  }
  delay(500);
}
```

- c. Pengujian pembacaan sensor HR dilakukan dengan membaca hasil data yang ditampilkan pada serial monitor, dibandingkan dengan hasil HR (*pulse*) dari Blood Pressure Monitor OMRON seperti pada gambar 4.18. Hasil data sensor HR yang diambil berdasarkan data yang masuk bersamaan dengan keluarnya nilai pada alat *Blood Pressure Monitor*. Sehingga hasil selisih perbandingan terlihat pada Tabel 4.1.



Gambar 4.18. Hasil pembacaan HR

Tabel 4.1. Selisih perbandingan hasil sensor HR dengan *Blood Pressure Monitor*

No	Sampel	HR Max (bpm)	Alat Pembaca <i>Heart Rate</i>		Error (%)
			Finger Clip (bpm)	<i>Blood Pressure Monitor</i> OMRON (bpm)	
1	A	197	62	61	1.64
2	B	199	68	69	1.47
3	C	190	64	64	0
4	D	196	63	62	1.58
5	E	195	72	73	1.37
Rata-rata kesalahan (%)					1.21

Dari hasil percobaan sensor Grove - *finger clip heart rate sensor* dan Blood Pressure Monitor OMRON pada tabel diatas, disimpulkan bahwa perbedaan nilai antara sensor Grove - *finger clip* tidak jauh berbeda dengan sensor pembaca lain, dimana untuk perbandingan dengan pembaca *Blood Pressure Monitor* OMRON didapatkan selisih error sebesar 1.21%.

4.7. Pengujian Pengiriman Data ESP01 - Firebase (Create, Read, Update)

4.7.1. Tujuan

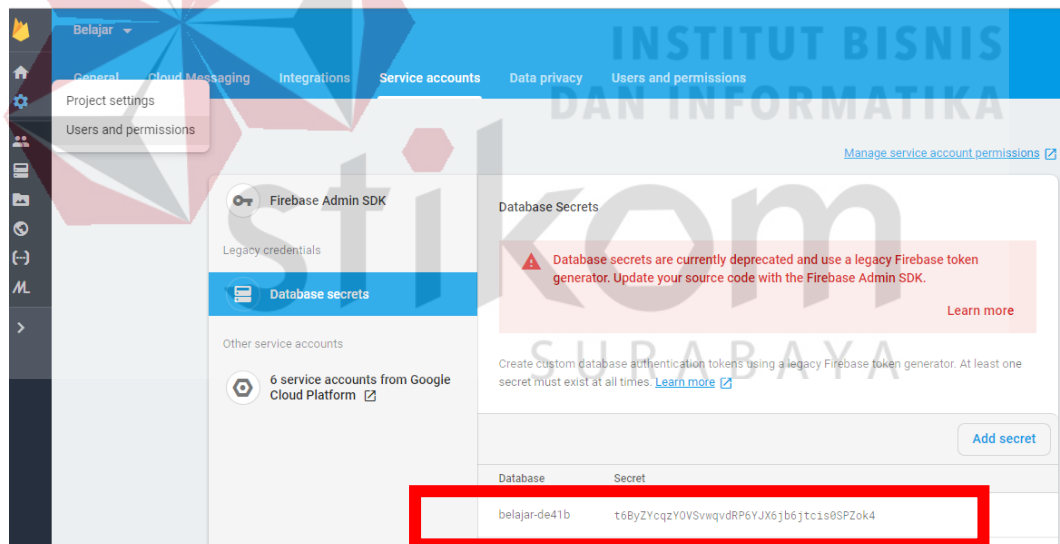
Tujuan dilakukan pengujian ini adalah untuk mengetahui bagaimanakah cara penggunaan fitur *syntax* dalam penggunaan Firebase pada ESP-01 untuk keperluan mengoperasikan CRU (Create, Read, Update) pada data yang ada di database Firebase.

4.7.2. Peralatan dan yang digunakan

1. Rangkaian yang digunakan sama seperti pada rangkaian sub bab 4.2.
2. Laptop/PC
3. Arduino IDE
4. Web Browser
5. Akun google
6. Database Firebase.
7. *Access point*.

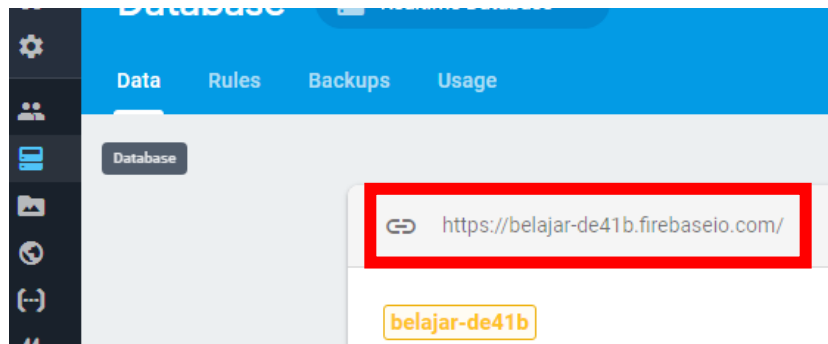
4.7.3. Cara pengujian

- a. Sambungkan dan hubungkan rangkaian sesuai dengan sub bab 4.2. Jangan lupa lepas GPIO0 dengan GND jika sudah upload.
- b. Setting database Firebase pada web www.firebaseio.com
 1. Login ke firebase, klik “Get Started”.
 2. *Create* database Firebase baru :
New project -> masukkan nama project -> Create project.
 3. Dapatkan Firebase *host* dan *auth key* seperti pada gambar 4.19. dan 4.20.
 4. Buka menu settings -> Users and Permissions -> Service accounts -> Database secrets -> Copy secret untuk *auth key*.



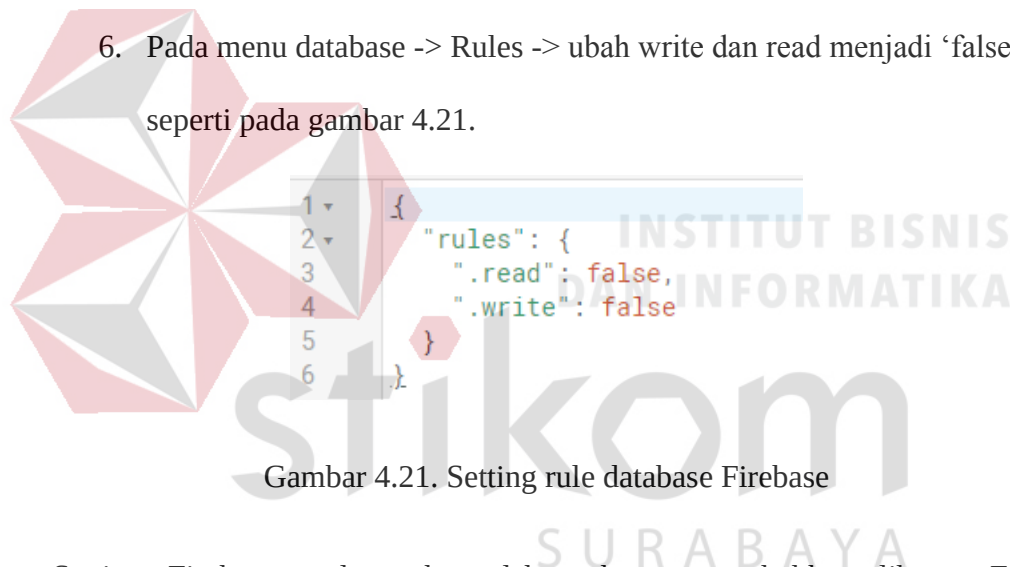
Gambar 4.19. *Auth key* firebase untuk ESP-01

5. Buka menu database -> Data -> copy nama firebase host (belajar-de41b.firebaseio.com) untuk mendapatkan *host*.



Gambar 4.20. Nama firebase host untuk ESP-01

6. Pada menu database -> Rules -> ubah write dan read menjadi 'false', seperti pada gambar 4.21.



Gambar 4.21. Setting rule database Firebase

- c. Setting Firebase pada web sudah, sekarang tambahkan library Firebase (*download* di google) dengan cara yang sama seperti pada langkah (b) pada sub bab 4.4.3. jika belum ada.
- d. Upload program untuk percobaan tes kirim data dari ESP-01 ke firebase untuk keperluan *create*, *update*, *delete* (perhatikan comment program tentang penggunaan *syntax-syntax* yang akan digunakan). Jangan lupa setting arduino IDE untuk ESP nya, sama seperti sub bab 4.2.3. Upload program ke ESP-01 sebagai berikut :

```

#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define WIFI_SSID "Freyja" ///Nama Wifi
#define WIFI_PASSWORD "UtaMacross" ///Password Wifi
#define FIREBASE_HOST "belajar-de41b.firebaseio.com"
#define FIREBASE_AUTH "t6ByZYcqzYOVSvwqvvdRP6YJX6jb6jtcis0SPZok4"
int x=0, y=0, a=0, b=0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

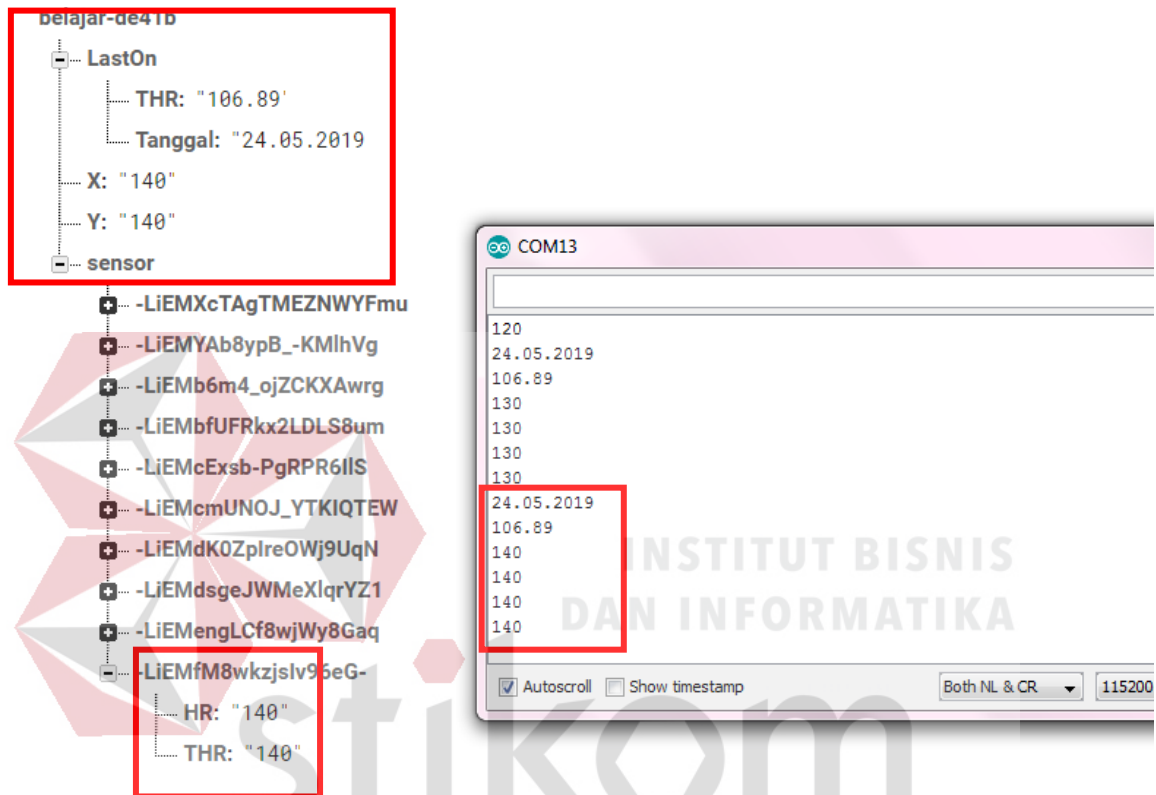
void loop() {
  x=x+10;
  y=y+10;
  String vx = String(x);
  String vy = String(y);
  a=a+10;
  b=b+10;
  String va = String(a); String vb = String(b);
  ///// Tambah child data baru /////
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  root["HR"] = va;
  root["THR"] = vb;
  String name = Firebase.push("/sensor", root);

  ///// Set data /////
  Firebase.setString("X", vx);
  Firebase.setString("Y", vy);

  ///// Ambil data /////
  FirebaseObject object = Firebase.get("/LastOn");
  String tanggal = object.getString("Tanggal");
  String THR = object.getString("THR");
  Serial.println(tanggal);
  Serial.println(THR);
  Serial.println(vx);
  Serial.println(vy);
  Serial.println(va);
  Serial.println(vb);
  delay(1000);
}

```

- e. Dari hasil data *serial monitor* dan pencocokan data pada Firebase pada gambar 4.22., didapat hasil yang sama. Dapat disimpulkan bahwa *syntax* pada *script* untuk CRU dapat digunakan.



Gambar 4.22. Hasil pada serial monitor dan database Firebase

4.8. Pengujian Pengiriman Data Antara Arduino – ESP01 – Firebase

4.8.1. Tujuan

Tujuan dilakukan pengujian ini adalah untuk mengetahui bagaimanakah cara mengkomunikasikan data yang berasal dari Arduino jika ingin mengupload nilai data ke database Firebase, yang mana menggunakan ESP-01 sebagai perantaranya.

4.8.2. Peralatan yang digunakan

1. USB to TTL
2. Kabel jumper
3. Arduino Pro Mini
4. ESP-01
5. Firebase
6. Arduino IDE
7. Web Browser
8. *Access Point*

4.8.3. Cara pengujian

- a. Program ESP-01 dan program Arduino dibedakan, diibaratkan seperti mikro berkomunikasi dengan mikro menggunakan komunikasi serial.
- b. Siapkan program yang akan di upload dari Arduino, semisal data string “!15.05.2019@17:30#168.73\$68^” akan dicoba untuk dikirimkan. Upload program ke Arduino Pro Mini menggunakan rangkaian yang sama seperti pada sub bab 4.1.
- c. Siapkan program yang akan di upload di ESP-01, pada percobaan ini ESP-01 akan memilah data yang diterima dan di proses untuk disimpan ke dalam database Firebase.
- d. Upload program ke ESP-01 menggunakan rangkaian yang sama seperti pada sub bab 4.2. Berikut adalah program yang akan di *upload* :

```

#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <Firebase.h>
#include <FirebaseArduino.h>
///set wifi
#define WIFI_SSID "Freyja" ///Nama Wifi
#define WIFI_PASSWORD "UtaMacross" ///Password Wifi
#define FIREBASE_HOST "belajar-de41b.firebaseio.com"
#define FIREBASE_AUTH "t6ByZYcqzYOVSvwqvdrP6YJX6jb6jtcis0SPZok4"
#define LED 2
String dataSerial = "";
String tanggal, waktu, THR, HR;
short int statusrun = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("|");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  digitalWrite(LED, HIGH);
  do {
    digitalWrite(LED, LOW);
    char c = Serial.read();
    if (c == '*') {
      statusrun = 0;
      Firebase.setString("/User1/LastOn/Status", "TES");
      Serial.println("TES");
    } else if (c == '!') {
      statusrun = 1;
    }
  }

  if (statusrun == 1) {
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    if (c == ':' || c == '.' || c == '0' || c == '1' ||
        c == '2' || c == '3' || c == '4' || c == '5' ||
        c == '6' || c == '7' || c == '8' || c == '9') {
      dataSerial += c;
    }
  }
}

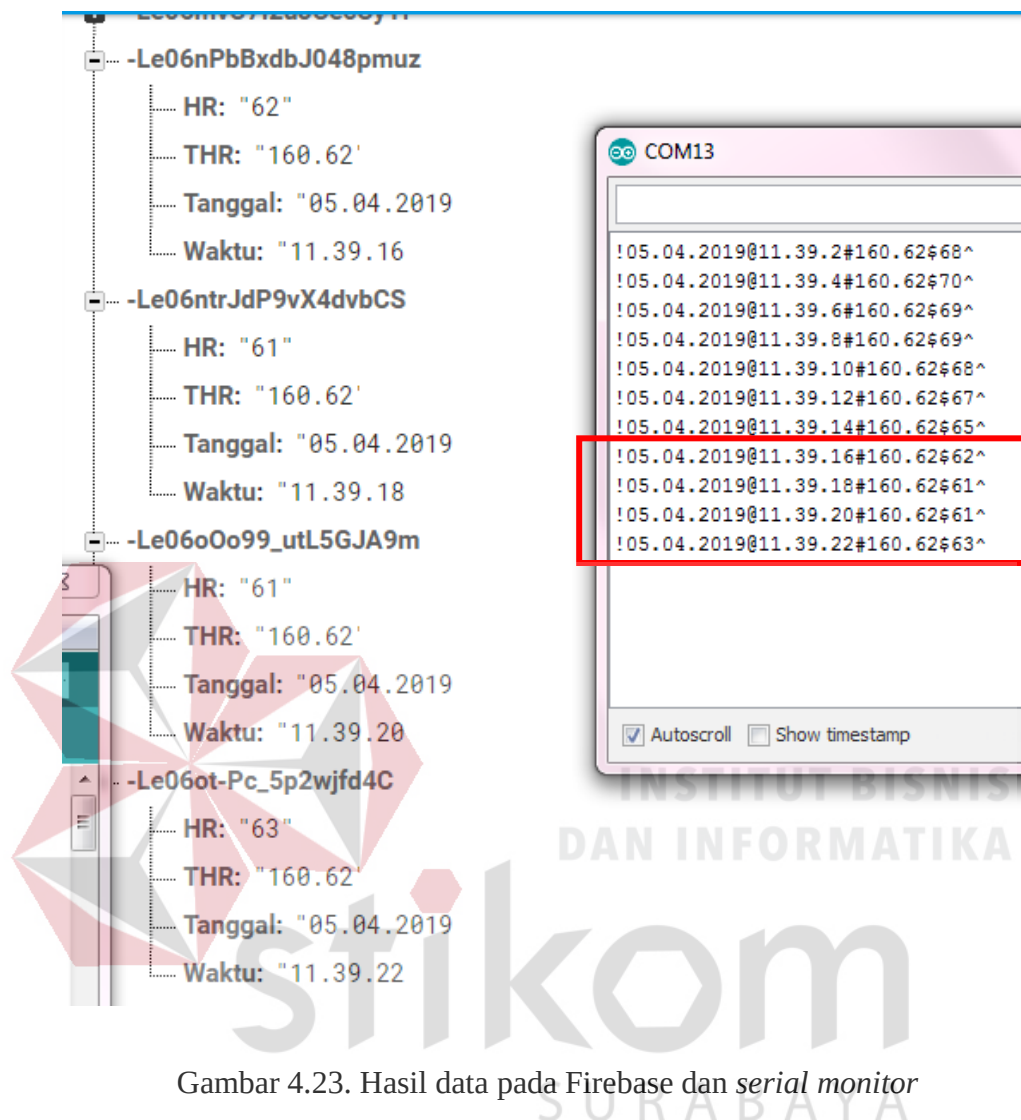
```

```

else if (c == '@') {
  ///Kirim tanggal
  tanggal = "";
  tanggal = dataSerial;
  dataSerial = "";
}
else if (c == '#') {
  ///Kirim waktu
  waktu = "";
  waktu = dataSerial;
  dataSerial = "";
}
else if (c == '$') {
  ///Kirim THR
  THR = "";
  THR = dataSerial;
  dataSerial = "";
}
else if (c == '^') {
  ///Kirim HR
  HR = "";
  HR = dataSerial;
  root["Tanggal"] = tanggal;
  root["Waktu"] = waktu;
  root["THR"] = THR;
  root["HR"] = HR;
  Firebase.push("/User/HistoryHR", root);
  dataSerial = "";
}
}
} while (Serial.available() > 0);
}

```

- e. Cek hasil pada Firebase apakah sama dengan serial monitor (data yang dikirimkan oleh Arduino Pro Mini adalah yang ditampilkan di serial monitor).
- f. Dari hasil yang didapat pada gambar 4.23., dapat disimpulkan bahwa data yang dikirim oleh Arduino Pro Mini yang diterima ESP-01 telah dikirimkan, dan cara pemilahan data untuk Firebase dapat diaplikasikan untuk keseluruhan sistem.



Gambar 4.23. Hasil data pada Firebase dan serial monitor

4.9. Pengujian Monitoring History HR pada Android

4.9.1. Tujuan

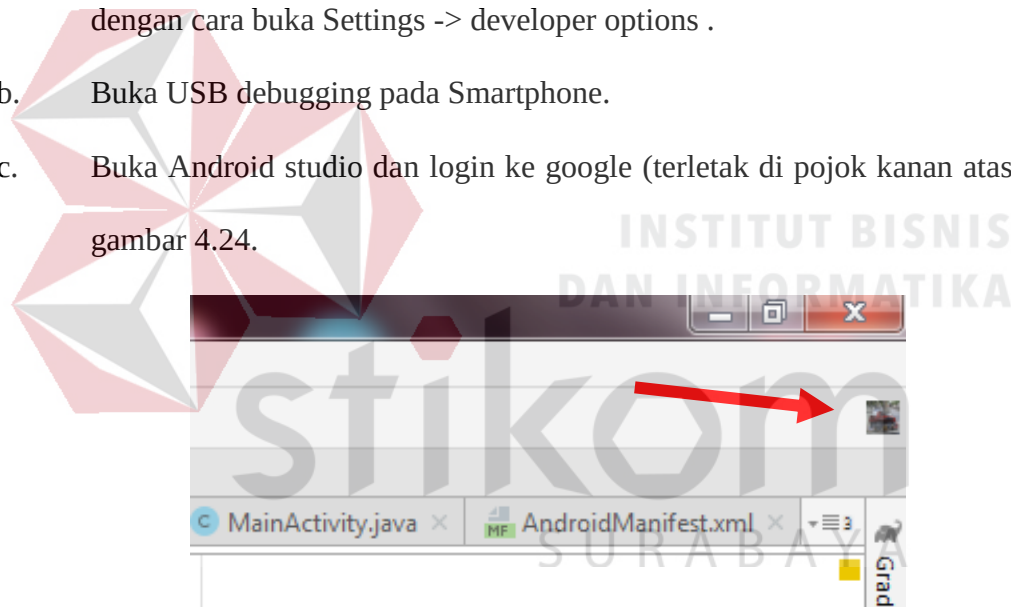
Tujuan dilakukan pengujian ini adalah untuk mengetahui bagaimanakah cara menggunakan Android Studio untuk menampilkan pengambilan data pada tiap *form-formnya*, terutama pada *form* ketika menampilkan data history HR dan THR.

4.9.2. Peralatan yang digunakan

1. Android Studio 3.4.1 dan SDK ver. 28
2. Smartphone (*Operating system* Oreo keatas)
3. Firebase
4. Koneksi internet

4.9.3. Cara pengujian

- a. Buka *Develop Mode* pada Smartphone terlebih dahulu, dapat dilakukan dengan cara buka Settings -> developer options .
- b. Buka USB debugging pada Smartphone.
- c. Buka Android studio dan login ke google (terletak di pojok kanan atas), lihat gambar 4.24.



Gambar 4.24. Icon login akun Google

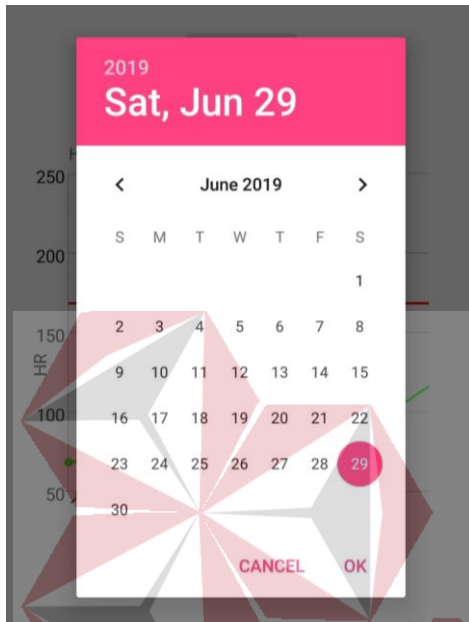
- d. Sinkronasikan Android Studio dengan Firebase, buka menu Tools -> Firebase -> Authentication -> Connect to Firebase. Secara otomatis akan ke *direct* untuk login ke akun Firebase.
- e. Masukkan program yang akan di *Upload* (contoh program yang akan di upload ada di lampiran).

- f. Hasil .apk dapat berjalan, form login seperti pada gambar 4.25. Jika login user maka langsung masuk ke Form tampil history user, jika login PT maka PT akan memilih *user* mana yang akan dilihat *history*nya terlebih dahulu seperti pada gambar 4.26.

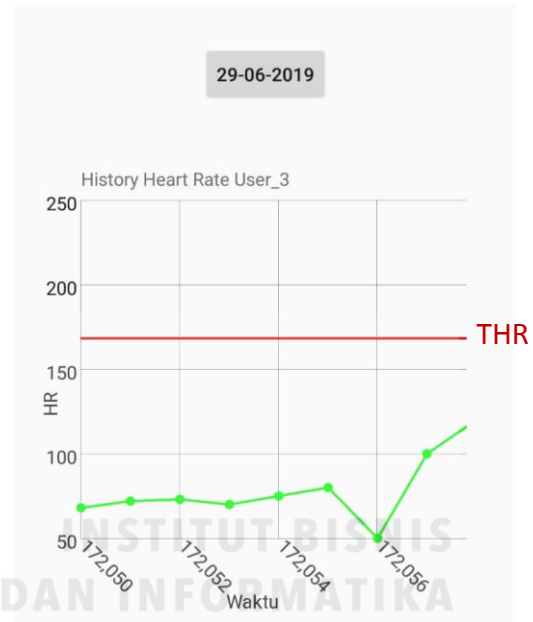
The image contains two screenshots of an Android application. The top screenshot, labeled 'Gambar 4.25. Form .xml login', shows a login form with a heart icon and an ECG line. It has two input fields: 'Username' with the value 'PT_3' and 'Password' with the value 'pt3'. A 'LOGIN' button is at the bottom. The bottom screenshot, labeled 'Gambar 4.26. Form .xml pilih user', shows a screen with three buttons labeled 'USER 1', 'USER 2', and 'USER 3'. A large watermark 'stikom SURABAYA' is visible across both screenshots.

- g. Kemudian memilih tanggal berapakah data *history* HR yang akan ditampilkan. Gambar 4.28. adalah hasil data yang dipilih oleh Date-Time Picker pada gambar

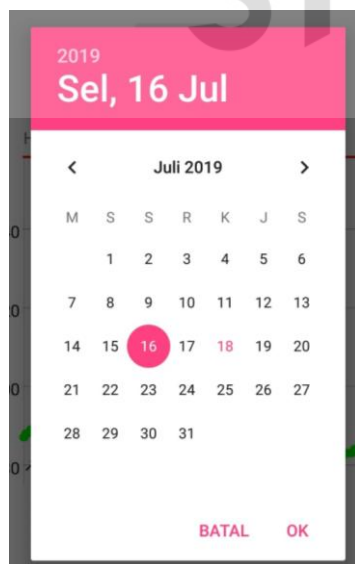
4.27., dan gambar 4.30 adalah hasil data yang dipilih oleh Date-Time Picker pada gambar 4.29. Nilai THR ditandakan oleh garis berwarna merah, dan nilai history HR ditandakan oleh garis berwarna hijau.



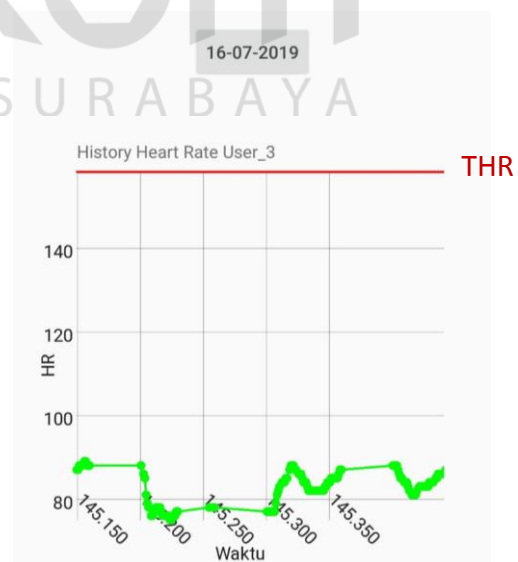
Gambar 4.27. Date-Time Picker (1)



Gambar 4.28. Form *history* HR (1)



Gambar 4. 29. Date-Time Picker (2)



Gambar 4. 30. Form *history* HR (2)

Dari hasil penyajian data pada gambar 4.28. didapat tulisan waktu 172052. maksud dari waktu tersebut adalah jam 17, menit 20, detik 52. Hal tersebut juga berlaku pada gambar 4.30.

- h. Dari hasil percobaan yang telah dilakukan didapat kesimpulan bahwa *syntax* Firebase pada Android Studio dapat digunakan dan dapat berjalan dengan baik.

4.10. Pengujian Keseluruhan Sistem

4.10.1. Tujuan

Pengujian keseluruhan meliputi pengujian mulai dari Hardware hingga data dapat terkirim ke *Software* (aplikasi). *Software* berupa aplikasi android berupa .apk dimana aplikasi tersebut dapat berfungsi untuk memantau history HR dari *user* oleh PT secara *realtime*. Bagian *hardware* berupa pemasangan setiap modul mulai dari input sensor, RTC, *switch*, dan *push button* sampai ke output berupa OLED dan hasil di database Firebase. Setiap *user* akan dilakukan pengambilan data dan diuji selama 70 detik dengan beberapa-beberapa kemungkinan skenario ketika ketiga alat dijalankan.

4.10.2. Peralatan yang digunakan

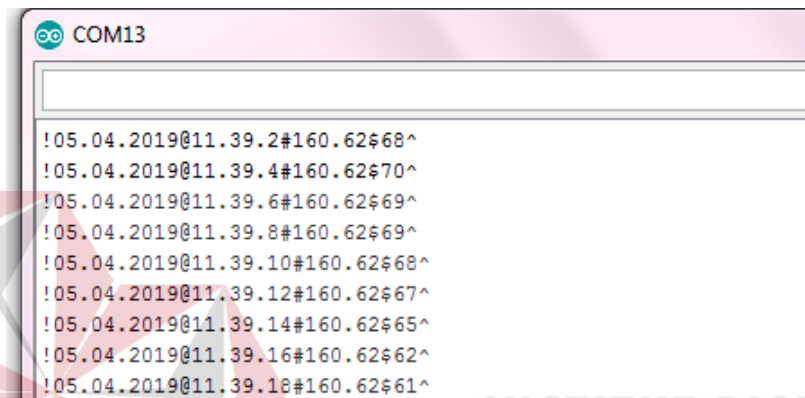
1. Arduino IDE
2. Android Studio 3.4.1 dan SDK ver. 28
3. Smartphone (*Operating system* Oreo keatas)
4. Firebase
5. *Access Point*

6. Arduino Pro Mini
7. ESP8266-01
8. USB to TTL
9. Adafruit OLED 128x64
10. RTC DS3231
11. Kabel Jumper
12. Buzzer
13. LED
14. Push dan Switch button
15. Li-Po Battery Ytong 702030
16. Modul charger battery TP4056

4.10.3. Cara pengujian

- a. Siapkan program yang akan di upload ke Arduino Pro Mini, ESP8266-01, dan Android (tertera pada lampiran).
- b. Lepas USB to TTL setelah upload program, hubungkan setiap perangkat seperti pada gambar 3.4. (sumber daya berasal dari baterai).
- c. Hidupkan perangkat, masuk ke init mode, masukkan usia (tekan push button), tekan switch untuk menentukan jenis kelamin, dan baca HR pada tiap *device*. Nilai THR didapatkan. Tekan tombol reset jika ketika setting ada kesalahan. OLED digunakan sebagai display.
- d. LED putih akan menyala jika ada masalah di bagian pengiriman data ESP-01. LED hijau akan menyala ketika THR telah tercapai ketika berolahraga.

- e. Ketika ditekan mode run, pembacaan history HR dan pengiriman HR akan dikirimkan ke Firebase.
- f. Lihat hasil pada serial monitor untuk komunikasi serial ketika mode run dan mode init, apakah data yang dikirimkan oleh Arduino Pro Mini sesuai format seperti pada gambar 4.31. atau tidak.



Gambar 4.31. Format data yang dikirim ke ESP-01 (Mode Run)

- g. Nyalakan *access point*, dilakukan pengujian untuk *multi-user* selama 70 detik dengan pengambilan sample setiap detik, dengan ketentuan sebagai berikut :
 1. Detik ke 0 : user_1 mati, user_2 dan user_3 menyala.
 2. Detik ke 10 : user_1 dinyalakan, semua user hidup.
 3. Detik ke 20 : user_2 mati, user_1 dan user_3 menyala.
 4. Detik ke 30 : user_1 dan user_2 mati, user_3 menyala.
 5. Detik ke 40 : user_1 dan_2 dinyalakan, ketiga user menyala.
 6. Detik ke 50 : semua user dimatikan.
 7. Detik ke 60 : semua user dinyalakan.
 8. Detik ke 70 : semua user dimatikan.

- h. Data dari Arduino Pro Mini diterima oleh ESP-01, hasil pilahan data pada ESP-01 kemudian dikirimkan ke Firebase. Lihat hasil pada laman web browser Firebase. Setiap hasil data masuk ke tiap *node* pada struktur JSON *tree* Firebase. *Child* baru ditambahkan setiap ada data baru yang di *push*.
- i. Berikut hasil data yang disimpan pada Firebase menggunakan scenario langkah (g) yang ditampilkan pada tabel 4.2. untuk data tanggal 15/06/2019, dan tabel 4.3. untuk data pada tanggal 20/07/2019.

Tabel 4. 2. *History* pencatatan HR tanggal 15/06/2019

Tanggal	Detik ke-	Waktu	HR user (bpm)		
			1	2	3
15/06/19	1	13:30:22	-	63	62
15/06/19	2	13:30:23	-	63	63
15/06/19	3	13:30:24	-	62	63
15/06/19	4	13:30:25	-	83	65
15/06/19	5	13:30:26	-	85	67
15/06/19	6	13:30:27	-	84	75
15/06/19	7	13:30:28	-	76	73
15/06/19	8	13:30:29	-	73	78
15/06/19	9	13:30:30	-	68	85
15/06/19	10	13:30:31	-	68	78
15/06/19	11	13:30:32	60	70	80
15/06/19	12	13:30:33	60	85	82
15/06/19	13	13:30:34	61	85	82
15/06/19	14	13:30:35	60	87	85
15/06/19	15	13:30:36	63	102	100
15/06/19	16	13:30:37	62	102	101

Tanggal	Detik ke-	Waktu	HR user (bpm)		
			1	2	3
15/06/19	17	13:30:38	62	103	-
15/06/19	18	13:30:39	62	102	99
15/06/19	19	13:30:40	60	115	99
15/06/19	20	13:30:41	61	114	100
15/06/19	21	13:30:42	81	113	120
15/06/19	22	13:30:43	81	-	118
15/06/19	23	13:30:44	80	-	117
15/06/19	24	13:30:45	70	-	120
15/06/19	25	13:30:46	68	-	119
15/06/19	26	13:30:47	68	-	135
15/06/19	27	13:30:48	67	-	132
15/06/19	28	13:30:49	65	-	130
15/06/19	29	13:30:50	65	-	129
15/06/19	30	13:30:51	63	-	142
15/06/19	31	13:30:52	-	-	135
15/06/19	32	13:30:53	-	-	137
15/06/19	33	13:30:54	-	-	145
15/06/19	34	13:30:55	-	-	144
15/06/19	35	13:30:56	-	-	132
15/06/19	36	13:30:57	-	-	130
15/06/19	37	13:30:58	-	-	133
15/06/19	38	13:30:59	-	-	133
15/06/19	39	13:31:00	-	-	132
15/06/19	40	13:31:01	-	-	135
15/06/19	41	13:31:02	65	137	145
15/06/19	42	13:31:03	68	136	140
15/06/19	43	13:31:04	64	136	142

Tanggal	Detik ke-	Waktu	HR user (bpm)		
			1	2	3
15/06/19	44	13:31:05	63	135	-
15/06/19	45	13:31:06	63	136	152
15/06/19	46	13:31:07	62	135	135
15/06/19	47	13:31:08	62	138	155
15/06/19	48	13:31:09	61	138	154
15/06/19	49	13:31:10	61	136	155
15/06/19	50	13:31:11	-	-	156
15/06/19	51	13:31:12	-	-	-
15/06/19	52	13:31:13	-	-	-
15/06/19	53	13:31:14	-	-	-
15/06/19	54	13:31:15	-	-	-
15/06/19	55	13:31:16	-	-	-
15/06/19	56	13:31:17	-	-	-
15/06/19	57	13:31:18	-	-	-
15/06/19	58	13:31:19	-	-	-
15/06/19	59	13:31:20	-	-	-
15/06/19	60	13:31:21	-	-	-
15/06/19	61	13:31:22	68	127	-
15/06/19	62	13:31:23	68	125	147
15/06/19	63	13:31:24	65	125	147
15/06/19	64	13:31:25	65	125	153
15/06/19	65	13:31:26	63	143	144
15/06/19	66	13:31:27	62	145	144
15/06/19	67	13:31:28	61	145	145
15/06/19	68	13:31:29	60	149	145
15/06/19	69	13:31:30	60	151	143
15/06/19	70	13:31:31	-	150	143

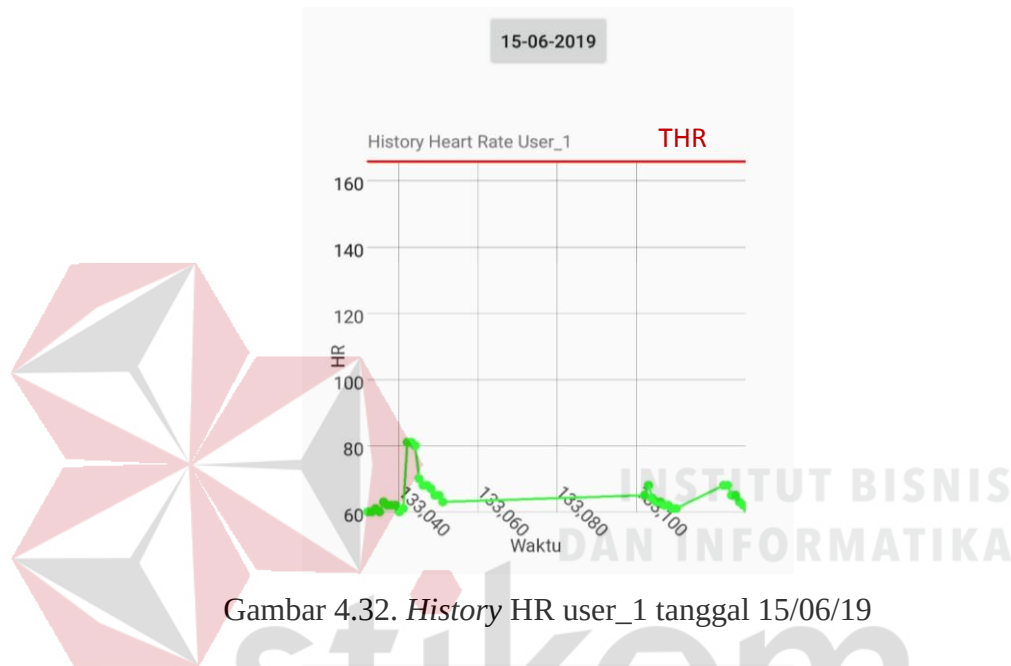
Tabel 4. 3. *History* pencatatan HR tanggal 20/07/2019

Tanggal	Detik ke-	Waktu	HR user (bpm)		
			1	2	3
20/07/19	1	19:43:27	-	63	64
20/07/19	2	19:43:28	-	-	63
20/07/19	3	19:43:29	-	80	-
20/07/19	4	19:43:30	-	80	62
20/07/19	5	19:43:31	-	79	62
20/07/19	6	19:43:32	-	79	74
20/07/19	7	19:43:33	-	79	91
20/07/19	8	19:43:34	-	80	96
20/07/19	9	19:43:35	-	80	100
20/07/19	10	19:43:36	-	82	-
20/07/19	11	19:43:37	83	-	103
20/07/19	12	19:43:38	84	81	105
20/07/19	13	19:43:39	84	83	107
20/07/19	14	19:43:40	86	87	108
20/07/19	15	19:43:41	84	92	109
20/07/19	16	19:43:42	84	94	-
20/07/19	17	19:43:43	-	96	110
20/07/19	18	19:43:44	83	98	111
20/07/19	19	19:43:45	83	99	111
20/07/19	20	19:43:46	83	100	112
20/07/19	21	19:43:47	82	-	113
20/07/19	22	19:43:48	82	-	-
20/07/19	23	19:43:49	82	-	113

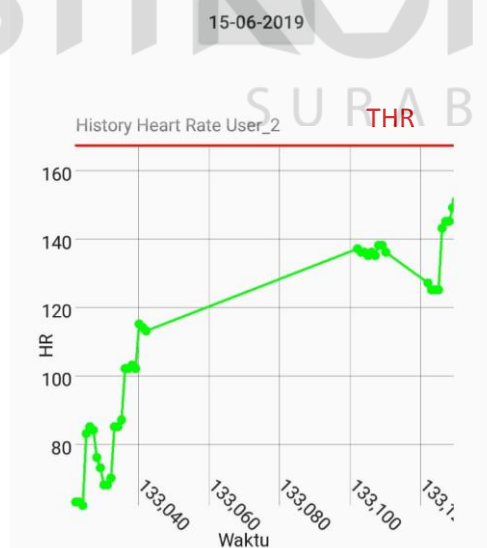
Tanggal	Detik ke -	Waktu	HR user (bpm)		
			1	2	3
20/07/19	24	19:43:50	-	-	115
20/07/19	25	19:43:51	82	-	110
20/07/19	26	19:43:52	82	-	112
20/07/19	27	19:43:53	82	-	112
20/07/19	28	19:43:54	82	-	-
20/07/19	29	19:43:55	81	-	111
20/07/19	30	19:43:56	-	-	110
20/07/19	31	19:43:57	-	-	112
20/07/19	32	19:43:58	-	-	118
20/07/19	33	19:43:59	-	-	120
20/07/19	34	19:44:00	-	-	120
20/07/19	35	19:44:01	-	-	-
20/07/19	36	19:44:02	-	-	120
20/07/19	37	19:44:03	-	-	120
20/07/19	38	19:44:04	-	-	121
20/07/19	39	19:44:05	-	-	118
20/07/19	40	19:44:06	-	-	116
20/07/19	41	19:44:07	78	104	-
20/07/19	42	19:44:08	-	105	116
20/07/19	43	19:44:09	81	105	112
20/07/19	44	19:44:10	81	105	111
20/07/19	45	19:44:11	81	105	110
20/07/19	46	19:44:12	81	-	112
20/07/19	47	19:44:13	81	106	112

Tanggal	Detik ke -	Waktu	HR user (bpm)		
			1	2	3
20/07/19	48	19:44:14	-	106	113
20/07/19	49	19:44:15	81	107	113
20/07/19	50	19:44:16	80	108	112
20/07/19	51	19:44:17	-	109	-
20/07/19	52	19:44:18	-	-	-
20/07/19	53	19:44:19	-	-	-
20/07/19	54	19:44:20	-	-	-
20/07/19	55	19:44:21	-	-	-
20/07/19	56	19:44:22	-	-	-
20/07/19	57	19:44:23	-	-	-
20/07/19	58	19:44:24	-	-	-
20/07/19	59	19:44:25	-	-	-
20/07/19	60	19:44:26	-	-	-
20/07/19	61	19:44:27	-	104	100
20/07/19	62	19:44:28	79	101	100
20/07/19	63	19:44:29	79	101	102
20/07/19	64	19:44:30	79	100	102
20/07/19	65	19:44:31	79	100	105
20/07/19	66	19:44:32	79	102	105
20/07/19	67	19:44:33	80	101	108
20/07/19	68	19:44:34	80	102	108
20/07/19	69	19:44:35	80	103	106
20/07/19	70	19:44:36	82	104	105

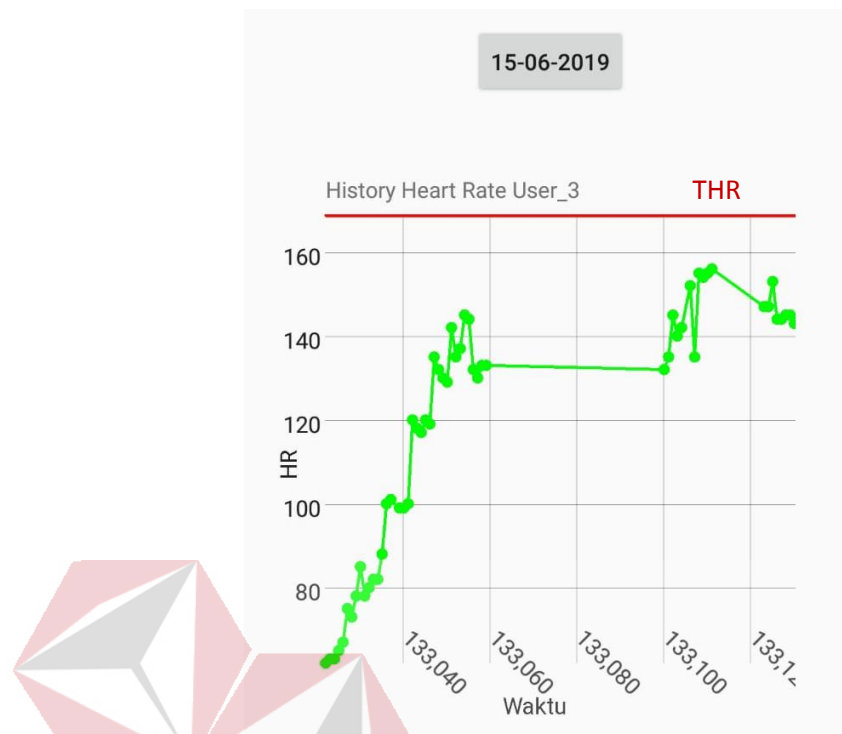
- j. Setelah data disimpan pada Firebase, data *history* HR tiap *user* ditampilkan pada Android. Berikut hasil data semua *user* pada tanggal 15/06/2019 yang ditampilkan pada gambar 4.32., 4.33., dan 4.34. Dan hasil data tanggal 20/07/2019 yang ditampilkan pada gambar 4.35., 4.36., dan 4.37.



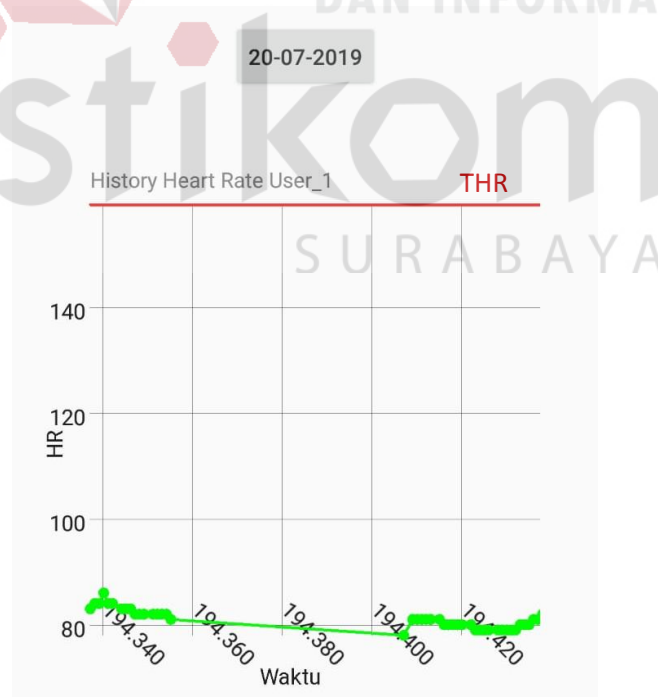
Gambar 4.32. *History* HR user_1 tanggal 15/06/19



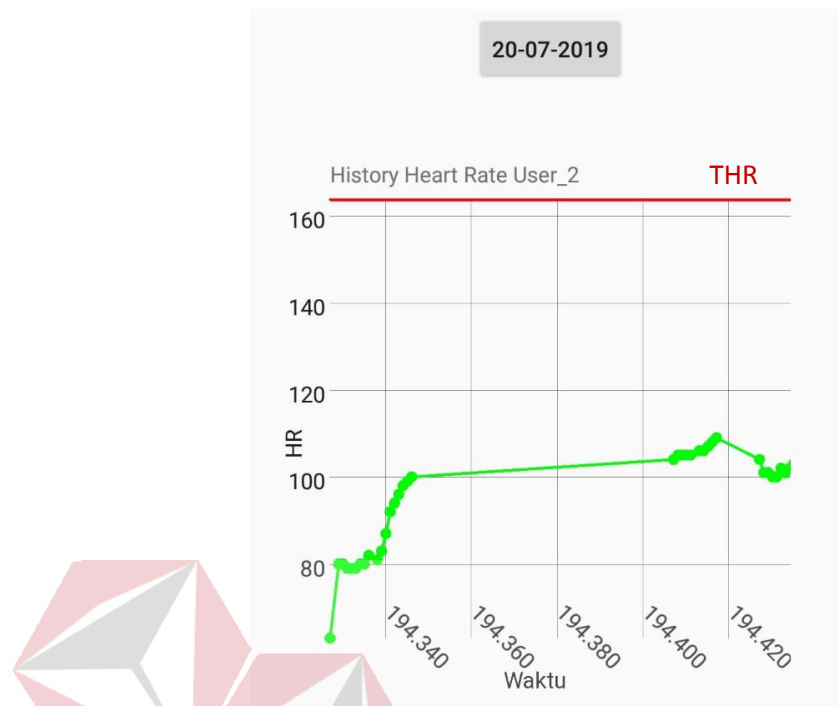
Gambar 4.33. *History* HR user_2 tanggal 15/06/19



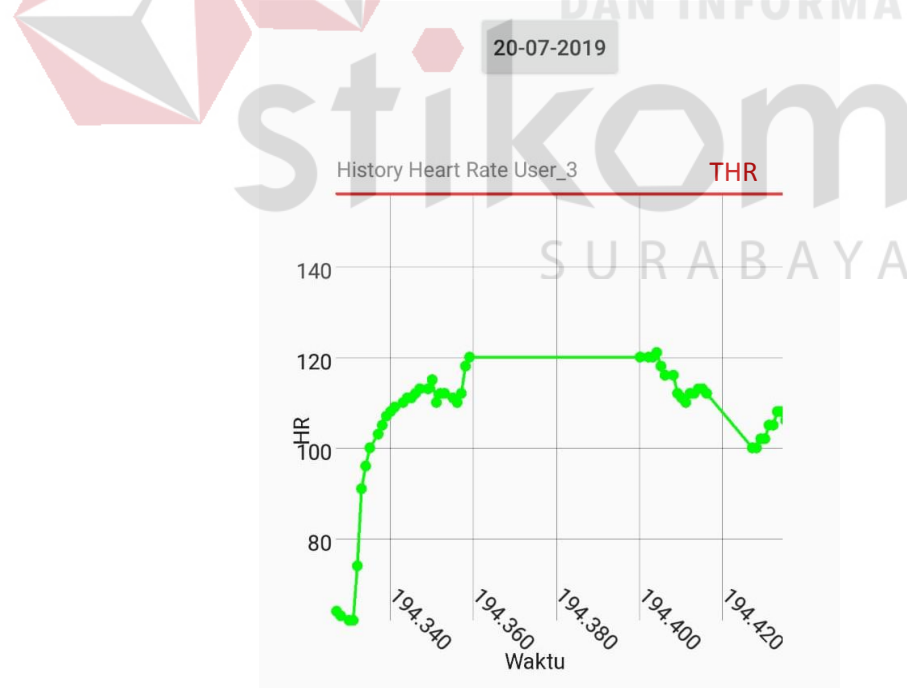
Gambar 4.34. History HR user_3 tanggal 15/06/19



Gambar 4. 35. History HR user_1 tanggal 20/07/19



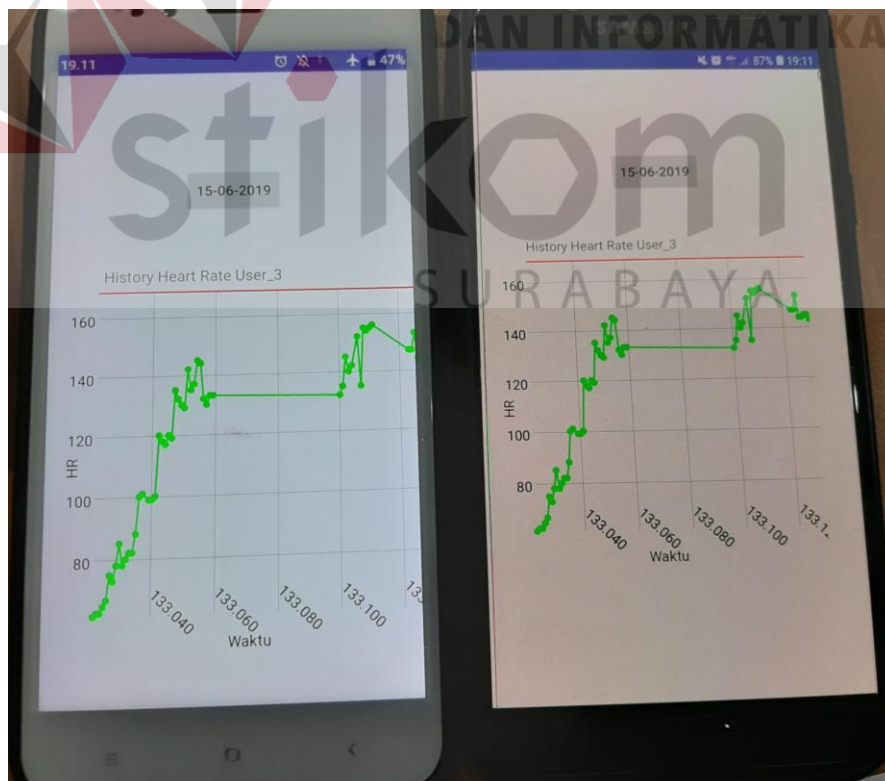
Gambar 4. 36. *History HR user_2 tanggal 20/07/2019*



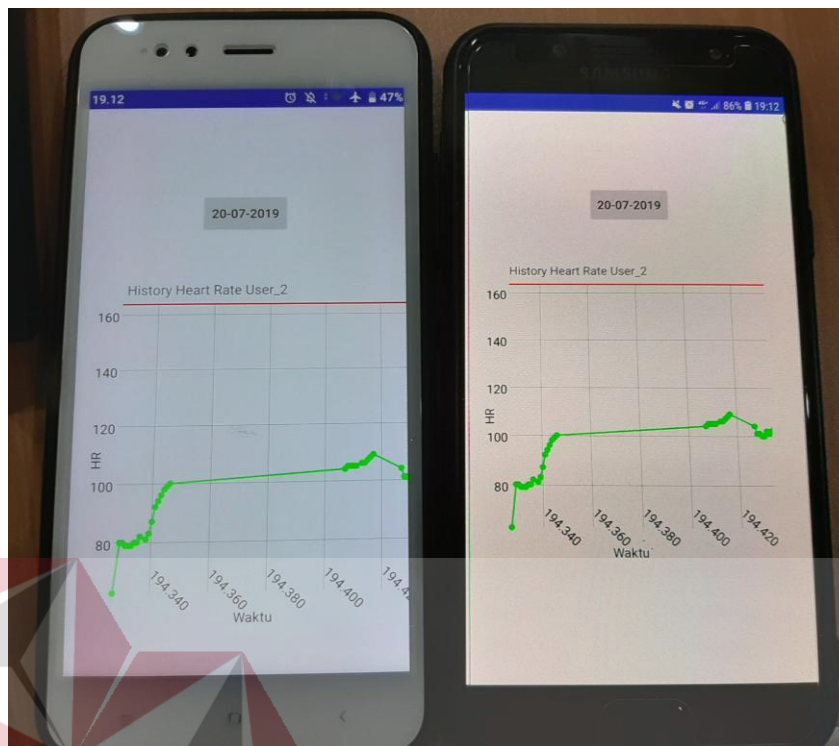
Gambar 4. 37. *History HR user_3 tanggal 20/07/2019*

Pada sumbu X axis terdapat nilai waktu 133040 dan seterusnya, maksud dari tampilan tersebut adalah 2 digit awal merupakan jam, 2 digit selanjutnya adalah menit, dan 2 digit terakhir adalah detik. Semisal X axis bernilai 133040, berarti data waktu tersebut jam 13 menit 30 detik 40.

- k. Selanjutnya dilakukan pengujian *shared access* dengan menggunakan dua *smartphone*. Apakah data tersebut bisa diakses secara bersamaan oleh PT dan *user*. Gambar 4.38. adalah tampilan ketika PT 3 (kiri) dan *user* 3 (kanan) mengakses data *user_3* pada tanggal 15/06/19, gambar 4.39. adalah tampilan ketika PT 2 (kiri) dan *user* 2 (kanan) mengakses data *user_2* pada tanggal 20/07/19, dan gambar 4.40 adalah tampilan ketika PT 3 (kiri) dan *user* 3 (kanan) mengakses data *user_3* pada tanggal 20/07/19.



Gambar 4. 38. PT 3 dan *user* 3 menampilkan data *user_3*



Gambar 4. 39. PT 2 dan user 2 menampilkan data user_2



Gambar 4. 40. PT 3 dan user 3 menampilkan data user_3

- l. Lihat pada tabel 4.2. dan 4.3., beberapa isi tabel ada yang kosong berdasarkan waktu (detik ke-) ketika dijalankan. Berikut penjelasannya :
 1. Pada tabel 4.3. detik ke- 28, 35, dan 41 pada user_3 terdapat nilai HR yang kosong, hal tersebut terjadi karena terdapat delay pengiriman waktu dari RTC sehingga data pada detik tersebut terlewat.
 2. Pada tabel 4.2. detik ke- 50 pada user_1 dan user_2 terdapat nilai HR yang kosong juga, hal tersebut terjadi karena pada detik tersebut ESP-01 belum mengirimkan data ke Firebase sedangkan catu daya dari baterai sudah dimatikan (masih ditampung di JSON buffer).
 3. Pada tabel 4.2. detik ke- 40 pada user_1 dan user_2 juga kosong, hal tersebut terjadi karena ketika ditekan ON, RTC pada detik tersebut terlewat dan langsung masuk pada detik selanjutnya (detik ke- 40).
- m. Dari percobaan keseluruhan didapatkan hasil bahwa sistem telah terintegrasi dengan baik mulai dari penyimpanan data dari *devices* ke *database* secara *multi-user*, hingga penampilan data secara *shared access* pada aplikasi Android.
- n. *JSON buffer* digunakan untuk *buffer* data yang akan dikirim terlebih dahulu, supaya tertata rapi dalam antrian dan tidak ada data yang hilang ketika koneksi jelek (data tersimpan dalam antrian pada *JSON buffer*). Ketika koneksi bagus, antrian yang tertahan tadi langsung dikirimkan semua. Jika tidak menggunakan *JSON buffer* maka yang terjadi adalah ketika koneksi jelek sedangkan nilai pada variabel-variabel belum terkirim, data lama pada variabel akan tertimpa data baru yang diterima ESP-01. Sehingga data lama tidak tersimpan pada Firebase.

BAB V

KESIMPULAN

5.1. Kesimpulan

1. Mikrokontroller Arduino Pro Mini dan ESP-01 dapat digunakan untuk perancangan sistem pembacaan *history*. Hasil pengujian Grove – *finger clip heart rate* sensor dibandingkan dengan OMRON *Blood Pressure Monitor* didapat tingkat akurasi 98,79%. Penggunaan RTC dan sensor *heart rate* dapat diintegrasikan dengan baik.
2. Pengiriman data secara *multi-user* dapat dilakukan dengan cara melakukan penataan data (diatur pada ESP-01) membentuk *child* dan *node* baru, yang dikirimkan oleh tiap *user* pada Firebase dengan struktur database JSON *tree*.
3. Berdasarkan pengujian yang dijelaskan pada sub bab 4.10., sinkronisasi pengiriman data antara Arduino Pro Mini, ESP-01, Firebase, dan aplikasi Android dapat dilakukan. Fitur JSON *buffer* Firebase pada ESP-01 diperlukan untuk *buffer data* yang akan dikirim supaya tertata rapi dalam antrian dan tidak ada data yang hilang ketika ESP-01 menerima data baru dari Arduino Pro Mini.

5.2. Saran

1. Posisi sensor perlu diperbaiki lagi.
2. Pengaturan atau desain perangkat pada sensor HR perlu dirapikan.
3. *User interface* pada apk Android perlu dikembangkan lagi sesuai dengan kondisi dan kebutuhan pelari.

DAFTAR PUSTAKA

- Indriyanna, Indah, 2012, *Sistem Penjualan Retail Mini Market Dengan VB6.0*. Yogyakarta : ALIF Media.
- Mendia I, M. K. 2016. *Assesment of Heart Rate Variability Threshold from Incremental Test Treadmill in Five Cross-Country Skiing*. Journal.Pone.
- Musayyanah, Ira P., Pauladie S. 2018. *Monitoring Target Heart Rate (THR) untuk optimalisasi latihan lari berbasis Internet of Things*. *Teknika : Engineering and Sains Journal*, Volume 2, Nomor 2, Desember 2018, 87-94.
- She, J., Nakamura, H., Makino, K. et al. *Selection of suitable maximum-heart-rate formulas for use with Karvonen formula to calculate exercise intensity*. *Int. J. Autom. Comput.* (2015) 12: 62.
- Yuan Z., Yu G., dkk. 2018. *Multi-User Shared Access for Internet of Things*. Tidak diterbitkan. ZTE Corporation.