

**RANCANG BANGUN PROTOTIPE SISTEM MONITORING
DETAK JANTUNG DENGAN MENGGUNAKAN
SMARTPHONE ANDROID BERBASIS
MIKROKONTROLLER ARDUINO**



VICKY ANGGITASARI

5215117031

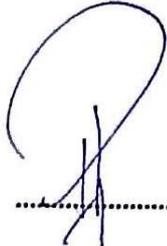
**Skripsi ini Ditulis untuk Memenuhi Sebagian Persyaratan
dalam Memperoleh Gelar Sarjana**

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA

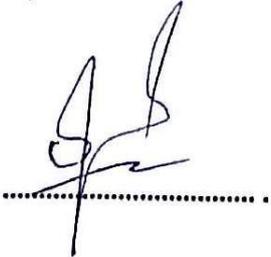
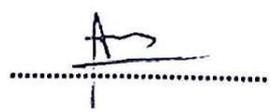
FAKULTAS TEKNIK UNIVERSITAS NEGERI JAKARTA

2017

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Drs. Wisnu Djatmiko, MT</u> (Dosen Pembimbing I)		27-Februari-2017
<u>Drs. Pitoyo Yuliatmojo, MT</u> (Dosen Pembimbing II)		27 Februari 2017

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Drs. Jusuf Bintoro, MT</u> (Ketua Sidang)		27 Februari 2017
<u>Dr. Moch. Sukardjo, M.Pd</u> (Sekretaris)		24-2-2017
<u>Aodah Djamah, Ph.D</u> (Dosen Ahli)		24 Feb 2017

Tanggal Lulus : 14 Februari 2017

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini menyatakan bahwa:

1. Karya tulis skripsi saya dengan judul “Rancang Bangun Prototipe Sistem Monitoring Detak Jantung Dengan Menggunakan Smartphone Android Berbasis Mikrokontroler Arduino Uno” adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Februari 2017

Yang membuat pernyataan

Vicky Anggitasari

NIM : 5215117031

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat, karunia dan hidayahnya, sehingga penulis dapat menyelesaikan skripsi dengan judul “ Rancang Bangun Prototipe Sistem Monitoring Detak Jantung dengan Menggunakan *Smartphone* Android berbasis Mikrokontroler Arduino ” yang merupakan persyaratan untuk meraih gelar Sarjana Pendidikan Teknik Elektronika pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.

Penulis menyadari bahwa dalam mengerjakan skripsi ini ditemukan berbagai kesulitan. Oleh karena itu skripsi ini tidak akan terwujud dengan baik tanpa adanya bimbingan, dorongan, saran – saran dan bantuan dari berbagai pihak. Maka pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Drs. Wisnu Djatmiko, M.T, selaku Dosen Pembimbing I
2. Drs. Pitoyo Yuliatmojo, M.T, selaku ketua Program Studi Pendidikan Teknik Elektronika FT-UNJ dan Dosen Pembimbing II
3. Drs. Nanang Arif Guntoro M.Si selaku Pembimbing Akademik
4. Kedua Orang tua dan keluarga serta teman-teman yang senantiasa mendoakan atas keberhasilan dan kelancaran dalam studi

Semoga segala kebaikan, keikhlasan, kesabaran dan doa yang telah diberikan kepada penulis akan mendapat balasan yang berlipat ganda dari Allah SWT. Akhir kata, semoga skripsi penulis dapat bermanfaat bagi yang membacanya.

Penulis

ABSTRAK

Vicky Anggitasari, *Rancang Bangun Prototipe Sistem Monitoring Detak Jantung dengan Menggunakan Smartphone Android berbasis Mikrokontroller Arduino*. Skripsi. Jakarta. Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta, 2016. Dosen Pembimbing: Drs. Wisnu Djatmiko, M.T dan Drs. Pitoyo Yuliatmojo, M.T.

Tujuan yang hendak dicapai dari penelitian pada pembuatan skripsi ini adalah peneliti mampu merancang dan merealisasikan prototipe sistem monitoring detak jantung manusia menggunakan sistem kendali Mikrokontroller Arduino UNO R3 dengan *input Pulse sensor* dan *output* berupa data yang ditampilkan pada aplikasi di *smartphone* Android

Penelitian ini memonitoring banyaknya detak jantung manusia permenit (Bpm) menggunakan *smartphone* android. Penyimpanan hasil pemeriksaan disimpan pada *smartphone* android uno R3 dengan menggunakan aplikasi yang dibuat dengan App *Inventor* yang penyimpanannya menggunakan basis data *TinyDB*. *Pulse sensor* adalah sensor pendeteksi detak jantung manusia yang dirancang khusus untuk Arduino dengan batas maksimal Bpm yg muncul adalah 240 Bpm tanpa batas bawah.

Hasil pengujian dengan membandingkan hasil antara PULSE OXIMETER dan prototipe dengan cara menempelkan ibu jari sebelah kanan dengan prototipe dan ibujari sebelah kiri pada Pulse Oximeter menunjukkan presentase error antara Pulse Oxymeter dan prototipe sistem monitoring detak jantung tidak lebih dari 5%, sehingga prototipe sistem monitoring detak jantung layak dan dapat digunakan.

Kata kunci : Arduino uno R3, Detak Jantung Manusia, Monitoring, Smartphone Android, pulse sensor.

ABSTRACT

Vicky Anggitasari, Design Build Prototype Heartbeat Monitoring System Using Android Smartphone Arduino-based microcontroller. Essay. Jakarta. Study Program Electronic Engineering, Faculty of Engineering, Universitas Negeri Jakarta, 2016. Supervisor: Drs. Vishnu Djatmiko, M.T and Drs. Pitoyo Yuliatmojo, M.T.

Goals to be achieved by research on the making of this thesis is the researcher is able to design and realize a prototype system using a human heartbeat monitoring control system microcontroller Arduino UNO R3 with Pulse sensor input and output of data displayed in applications on Android smartphone

This study monitoring many human heartbeat per minute (Bpm) using android smartphone. Storage of test results stored on android smartphone uno R3 using applications created with App Inventor uses a database that is stored TinyDB. Pulse sensor is human heartbeat detection sensors designed specifically for Arduino with a maximum limit naturally arises Bpm 240 Bpm no lower limit.

The test results by comparing the results between PULSE OXIMETER and prototypes by placing the thumb of the right hand with a prototype and thumb left on the Pulse Oximeter shows the percentage of error between Pulse Oxymeter and prototype system monitoring heart rate is not more than 5%, so the prototype monitoring system heartbeat feasible and can be used.

Keywords: Arduino Uno R3, Human Heart Beat, Monitoring, Android Smartphone, pulse sensor.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xiii

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah.....	1
1.2 Identifikasi Masalah	2
1.3 Pembatasan Masalah	3
1.4 Perumusan Masalah.....	3
1.5 Tujuan Penelitian.....	3
1.6 Manfaat Penelitian	4

BAB II TINJAUAN PUSTAKA

2.1 Kerangka Teoritik	5
2.1.1 Merancang Bangun.....	5
2.1.2 Alat	5
2.1.3 Prototipe Sistem	6
2.1.4 Monitoring.....	6
2.1.5 Detak Jantung	8
2.1.6 Detak Jantung Manusia	8
2.1.7 Sistem Monitoring Detak Jantung Manusia	11
2.1.8 Smartphone.....	12
2.1.9 Android.....	13
2.1.10 Arduino Uno R3	15

2.1.11	Arduino IDE.....	17
2.1.12	Pulse Sensor	18
2.1.13	Bluetooth HC 05.....	20
2.1.14	App Inventor	21
2.1.15	Basis Data (Database)	22
2.1.16	TinyDB.....	25
2.1.17	Metode Research and Development (RnD)	26
2.1.17.1	Definisi	26
2.1.17.2	Langkah-langkah R&D.....	27
2.2	Kerangka Berpikir	27
2.2.1	Blog Diagram	29
2.2.2	Flowchart.....	31
2.3	Hipotesis Penelitian.....	33

BAB III METODOLOGI PENELITIAN

3.1	Tempat dan Waktu Penelitian	34
3.2	Metode Pengembangan Produk.....	34
3.2.1	Tujuan Pengembangan	34
3.2.2	Metode Pengembangan	34
3.2.3	Sasaran Produk.....	35
3.2.4	Instrumen.....	35
3.3	Prosedur Pengembangan	35
3.3.1	Tahap Penelitian dan Pengumpulan Informasi.....	35
3.3.2	Tahap Perencanaan.....	36
3.3.3	Tahap Desain Produk	36
3.3.3.1	Rancangan Perangkat Keras	36
3.3.3.2	Perancangan Perangkat Lunak.....	37
a.	Perancangan Program Arduino	37
b.	Perancangan Aplikasi Android	48
c.	Perancangan Basis data TinyDB.....	52
3.4	Teknik Pengumpulan Data	56
3.4.1	Rangkaian Catu Daya.....	56

3.4.2	Rangkaian Modul Bluetooth HC 05.....	57
3.4.3	Rangkaian Pulse sensor.....	57
3.4.4	Rangkaian Indikator LED.....	57
3.5	Teknik Analisis Data.....	58
3.5.1	Pengujian Catu Daya.....	58
3.5.2	Pengujian Modul Bluetooth HC 05.....	59
3.5.3	Pengujian Pulse sensor.....	59
3.5.4	Pengujian Indikator LED.....	59

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1	Hasil Pengembangan Produk.....	60
4.1.1	Lokasi Penelitian.....	60
4.1.2	Implementasi Desain Prototipe.....	60
4.1.2.1	Implementasi Desain Prototipe Perangkat Keras.....	60
4.1.2.2	Implementasi Desain Prototipe Perangkat Lunak.....	62
4.2	Kelayakan Produk.....	63
4.3	Efektifitas Produk.....	66
4.3.1	Hasil Pengujian Catu Daya.....	67
4.3.2	Hasil Pengujian Modul Bluetooth HC 05.....	67
4.3.3	Hasil Pengujian Pulse sensor.....	68
4.3.4	Hasil Pengujian Indikator LED.....	69
4.4	Pembahasan.....	70

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan.....	71
5.2	Saran.....	71

DAFTAR PUSTAKA	73
-----------------------	-------	-----------

DAFTAR TABEL

Tabel 2.1	Detak jantung per menit saat istirahat	10
Tabel 3.1	Penggunaan Pin pada Arduino Uno	37
Tabel 3.2	menunjukkan kriterian pengujian catu daya.....	58
Tabel 3.3	Kriteria pengujian pada modul Bluetooth HC-05	59
Tabel 3.4	Kriteria Pengujian pulse sensor.....	59
Tabel 3.5	Kriteria Pengujian Indikator LED	59
Tabel 4.1	Hasil Implementasi Desain Prototipe	61
Tabel 4.2	Hasil implementasi prototipe perangkat lunak.....	62
Tabel 4.3	Hasil pengujian perbandingan	64
Tabel 4.4	Tampilan menu histori	65
Tabel 4.5	Hasil Pengujian Catu Daya	67
Tabel 4.6	Hasil pengujian modul bluetooth HC 05.....	68
Tabel 4.7	Hasil pengujian Pulse Sensor	69
Tabel 4.8	Hasil pengujian Indikator Led.....	70

DAFTAR GAMBAR

Gambar 2.1	<i>smartphone</i> dengan sistem operasi Android.....	13
Gambar 2.2	Arduino Uno R3	15
Gambar 2.3	Tampilan Sketsa Arduino IDE	18
Gambar 2.4	<i>Pulsesensor</i>	19
Gambar 2.5	Kit <i>Pulsesensor</i>	20
Gambar 2.6	Modul <i>Bluetooth</i> HC 05	21
Gambar 2.7	Tampilan <i>App Inventor</i>	22
Gambar 2.8	pengaplikasian <i>TinyDB</i> pada <i>App Inventor</i>	26
Gambar 2.9	Langkah – langkah Metode R&D.....	27
Gambar 2.10	Diagram Blog prototipe sistem monitoring detak jantung manusia.....	29
Gambar 2. 11	blog komponen prototipe sistem monitoring detak jantung Manusia	30
Gambar 2. 12	Flowchart arduino prototipe sistem monitoring detak jantung Manusia	31
Gambar 2. 13	Flowchart aplikasi android prototipe sistem monitoring detak jantung manusia.....	32
Gambar 3.1	Rancangan desain perangkat keras	36
Gambar 3.2	Flowchart perancangan program arduino	38
Gambar 3.3	Flowchart pengiriman perintah.....	39
Gambar 3.4	Flowchart penentuan batas gelombang	41
Gambar 3.5	Gelombang pulsa	42
Gambar 3.6	Flowchart untuk mencari detak jantung	43
Gambar 3.7	Flowchart untuk menghitung detak jantung	45
Gambar 3.8	Flowchart pengiriman data	47
Gambar 3.9	<i>ScreenHome</i>	49
Gambar 3.10	Tampilan <i>screen</i> periksa.....	50
Gambar 3.11	Tampilan <i>screen</i> daftar	51
Gambar 3.12	Tampilan <i>screen</i> histori	51
Gambar 3.13	Tampilan <i>screen</i> pengaturan.....	52
Gambar 3.14	program <i>TinyDB</i> untuk menu daftar.....	53

Gambar 3.15	pemrograman <i>TinyDB</i> untuk menu periksa.....	55
Gambar 3.16	Pemrograman pada menu histori	56
Gambar 3.17	<i>powerbank</i>	56
Gambar 3.18	Rangkaian <i>bluetooth</i> HC 05 dengan Aduino Uno R3	57
Gambar 3.19	Rangkaian <i>pulsesensor</i> dengan Arduino	57
Gambar 3.20	Rangkaian Indikator LED dengan Arduino.....	58
Gambar 4.1	Pengujian Arduino.....	60

DAFTAR LAMPIRAN

LAMPIRAN 1 : Instrumen Penelitian

LAMPIRAN 2 : List Program Arduino dan Interrupt

LAMPIRAN 3 : Program App Inventor

LAMPIRAN 4 : APDS 9008

LAMPIRAN 5 : Pulse Sensor

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Ilmu pengetahuan di bidang teknologi mengalami perkembangan yang pesat, perkembangan tersebut mulai dari perkembangan pola pikir manusia hingga manusia mampu membuat suatu alat yang dapat digunakan untuk memudahkan kerja manusia dan mampu menggantikan peran manusia. Salah satunya dalam pemeriksaan medis terutama dalam hal pengukuran sebagai alat kontrol kesehatan.

Kegiatan manusia yang kian hari kian padat membuat kebanyakan manusia mengesampingkan kesehatannya. Hal ini terbukti dengan banyaknya manusia yang hanya akan melakukan pemeriksaan medis apabila sudah merasakan keanehan atau sakit pada tubuhnya. Padahal sakit itu sendiri sebenarnya dapat dicegah dengan rutin melakukan pemeriksaan medis. Hal utama yang paling diperhatikan dalam pemeriksaan medis adalah jantung.

Jantung adalah organ penting dalam tubuh manusia yang difungsikan untuk memompa darah ke seluruh tubuh. Oleh karena itu, monitoring jantung merupakan hal yang penting untuk dilakukan mengingat tubuh manusia secara *continue* melakukan sirkulasi darah ke seluruh organ tubuh lainnya. Dengan mengetahui denyut jantung, dapat diketahui kondisi kesehatan seseorang. Frekuensi detak jantung tersebut memberikan informasi mengenai bagaimana keadaan jantung, cepat lambatnya implus jantung, ada tidaknya gangguan pembentukan implus dan gangguan fungsi jantung.

Dalam istilah kedokteran, ilmu yang mempelajari aktivitas listrik jantung atau yang sering disebut *Elektrokardiograf* dapat di rekam dan di gambarkan dalam bentuk grafik menggunakan mesin EKG (*Electrocardiogram*). Sebenarnya pemeriksaan dan pengukuran frekuensi denyut jantung dapat di lakukan secara manual, namun hal ini masih banyak mendapat kendala karena tidak akurat.

Saat ini dipasaran sudah beredar sebuah alat pendeteksi dan penghitung detak jantung yang sering disebut *pulse oximeter*. Namun, kelemahan *pulse oximeter* adalah tidak menyimpan data (histori) sehingga penggunaanya tidak dapat membuat perbandingan antara pemeriksaan-pemeriksaan yang sebelumnya.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang di atas maka dapat diidentifikasi masalah sebagai berikut :

1. Bagaimana membuat alat monitoring detak jantung manusia sebagaimana sistem kerja *Pulse Oximeter* ?
2. Bagaimana membuat sistem pendeteksi detak jantung menggunakan *Pulse sensor*?
3. bagaimana cara mengirimkan data dari *pulse sensor* ke Android tentang banyaknya detak jantung manusia per-sampling waktu yang sudah ditetapkan menggunakan Arduino Uno R3?
4. Bagaimana cara menyimpan data hasil pemeriksaan (histori) pada aplikasi android yang dibuat dengan App *Inventor*?

1.3 Pembatasan Masalah

Batasan masalah yang didapat dari hasil uraian latar belakang dan identifikasi masalah adalah sebagai berikut:

1. Heart rate sensor yang digunakan adalah XD-58C pulse heart rate sensor.
2. Mikrokontroller yang digunakan adalah mikrokontroller arduino UNO R3.
3. Waktu sampling yang diambil untuk mengambil data detak jantung adalah 60 detik/1 menit (Bpm).
4. monitoring dibedakan berdasarkan nomor ID yang mencakup nama, usia, alamat dan jenis kelamin.
5. Dalam pengukuran tidak dilakukan observasi pada penderita suatu penyakit.
6. Program yang digunakan bersumber dari situs *website* modul Heart rate sensor.

1.4 Perumusan Masalah

Perumusan masalah yang didapat berdasarkan hasil latar belakang masalah, identifikasi masalah dan batasan masalah adalah : “Bagaimana merancang bangun prototipe sistem monitoring detak jantung manusia pada *smartphone* android menggunakan *pulse sensor* berbasis arduino uno R3?”

1.5 Tujuan Penelitian

Sesuai masalah yang telah dirumuskan, maka tujuan yang hendak dicapai dari penelitian pada pembuatan skripsi ini adalah peneliti mampu merancang dan merealisasikan prototipe sistem monitoring detak jantung manusia menggunakan sistem kendali Mikrokontroller Arduino UNO R3 dengan *input Pulse sensor* dan *output* berupa data yang ditampilkan pada aplikasi di *smartphone* Android.

1.6 Manfaat Penelitian

Manfaat pada penelitian ini adalah untuk mengembangkan teknologi khususnya dalam bidang teknik Instrumentasi dan kendali serta dalam bidang kesehatan dengan membuat prorotipe sistem monitoring detak jantung manusia pada smartphone android dengan *input Pulse sensor* dan arduino uno R3 sebagai pengolah data.

BAB II

TINJAUAN PUSTAKA

2.1 Kerangka Teoritik

2.1.1 Merancang Bangun

Berdasarkan Kamus Besar Bahasa Indonesia tahun 2005, Merancang merupakan kegiatan mengatur segala sesuatu sebelum mengerjakan atau melakukan sesuatu. Sedangkan bangun adalah susunan yang memiliki struktur dan wujud. Artinya, merancang bangun merupakan kegiatan mengatur pengerjaan sebuah benda sehingga akhirnya memiliki struktur dan wujud sesuai dengan rancangan yang sudah dibuat.

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2002).

Berdasarkan definisi-definisi yang telah dikemukakan di atas pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisis ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

2.1.2 Alat

Menurut Kamus Besar Bahasa Indonesia tahun 2005, alat adalah suatu benda yang dipakai untuk mengerjakan sesuatu; perkakas, perabot, yang dipakai untuk mencapai maksud.

Berdasarkan pengertian di atas alat adalah benda yang digunakan untuk mengerjakan sesuatu yang fungsinya adalah untuk mempermudah pekerjaan.

2.1.3 Prototipe Sistem

Prototipe merupakan model kerja dasar dari pengembangan sebuah program (*software*) atau perangkat lunak.

Dalam bidang desain, prototipe atau purwarupa atau disebut juga dengan arketipe adalah bentuk awal sebagai contoh atau standar ukuran dari sebuah entitas. Sebuah prototipe dibuat sebelum dikembangkan atau justru dibuat khusus untuk pengembangan sebelum dibuat dalam skala sebenarnya atau sebelum diproduksi secara massal.

Sistem adalah susunan dari dua atau lebih komponen yang membentuk satu kesatuan dengan satu (Santoso, 2013).

Berdasarkan definisi di atas dapat disimpulkan, bahwa prototipe sistem adalah model pengembangan sebuah program perangkat keras (*hardware*) yang diintegrasikan dengan perangkat lunak (*software*) sebagai contoh atau standar ukuran dari sebuah entitas yang disusun dari dua atau lebih komponen yang membentuk satu kesatuan dengan satu tujuan.

2.1.4 Monitoring

Keberhasilan dalam mencapai tujuan, separuhnya ditentukan oleh rencana yang telah ditetapkan dan setengahnya lagi fungsi oleh pengawasan atau monitoring. Pada umumnya, manajemen menekankan terhadap pentingnya kedua fungsi ini, yaitu perencanaan dan pengawasan (*monitoring*) (Conor, 1974).

Kegiatan monitoring dimaksudkan untuk mengetahui kecocokan dan ketepatan kegiatan yang dilaksanakan dengan rencana yang telah disusun. Monitoring digunakan pula untuk memperbaiki kegiatan yang menyimpang dari rencana, mengoreksi penyalahgunaan aturan dan sumber-sumber, serta untuk mengupayakan agar tujuan dicapai seefektif dan seefisien mungkin. Berdasarkan kegunaannya, William Travers Jerome menggolongkan monitoring menjadi delapan macam, sebagai berikut:

1. Monitoring yang digunakan untuk memelihara dan membakukan pelaksanaan suatu rencana dalam rangka meningkatkan daya guna dan menekan biaya pelaksanaan program.
2. Monitoring yang digunakan untuk mengamankan harta kekayaan organisasi atau lembaga dari kemungkinan gangguan, pencurian, pemborosan, dan penyalahgunaan.
3. Monitoring yang digunakan langsung untuk mengetahui kecocokan antara kualitas suatu hasil dengan kepentingan para pemakai hasil dengan kemampuan tenaga pelaksana.
4. Monitoring yang digunakan untuk mengetahui ketepatan pendelegasian tugas dan wewenang yang harus dilakukan oleh staf atau bawahan.
5. Monitoring yang digunakan untuk mengukur penampilan tugas pelaksana.
6. Monitoring yang digunakan untuk mengetahui ketepatan antara pelaksanaan dengan perencanaan program.
7. Monitoring yang digunakan untuk mengetahui berbagai ragam rencana dan kesesuaiannya dengan sumber-sumber yang dimiliki oleh organisasi atau lembaga.

2.1.5 Detak Jantung

Detak jantung adalah debaran yang dikeluarkan oleh jantung akibat aliran darah melalui jantung (Herman, 2009). Dari debaran tersebut dapat dihitung jumlah denyutan jantung per satuan waktu. Denyut jantung yang optimal untuk setiap individu berbeda-beda tergantung pada kapan waktu mengukur detak jantung tersebut (saat istirahat atau setelah melakukan pergerakan). Variasi dalam detak jantung sesuai dengan jumlah oksigen yang diperlukan oleh tubuh saat itu.

2.1.6 Detak Jantung Manusia

Detak jantung terjadi akibat adanya dua mekanisme pada jantung yaitu sistole dan diastole. Sistole merupakan suatu fase dimana serambi relaksasi, serta bilik dari jantung berkontraksi. Adanya kontraksi ini menyebabkan daerah dalam ruang bilik bertekanan tinggi, serta terjadi gerakan peristaltik sehingga darah akan mengalir ke ruangan yang bertekanan lebih rendah, yakni menuju arteri, untuk selanjutnya darah tersebut akan beredar ke organ-organ melalui pembuluh darah.

Sedangkan diastole merupakan suatu fase saat serambi kontraksi serta bilik relaksasi. Pada saat terjadi kontraksi serambi terjadi tekanan yang lebih besar pada ruang serambi akibat kontraksinya tersebut, sehingga adanya tekanan yang lebih besar pada bagian serambi ini dibandingkan dengan daerah bilik, maka darah akan mengalir menuju bilik yang bertekanan tinggi (Venti, 2010).

Normalnya terdapat dua bunyi yang terjadi setiap satu siklus jantung. Yang pertama “lub” rendah yang agak memanjang (bunyi pertama), yang disebabkan oleh mulainya vibrasi oleh penutupan mendadak katub mitral dan trikuspid pada awal sistole bilik. Yang kedua “dub” bernada tinggi yang lebih singkat (bunyi

kedua), yang disebabkan oleh vibrasi yang disertai dengan penutupan katup aorta dan pulmonalis tepat setelah akhir sistole bilik. Bunyi ketiga bernada rendah yang lunak terdengar sekitar sepertiga jalan melalui diastole dalam banyak individu muda normal. Lalu bersamaan dengan masa pengisian bilik yang cepat dan mungkin karena vibrasi dimulai oleh aliran masuk darah. Bunyi keempat kadang-kadang dapat terdengar segera sebelum bunyi pertama sewaktu tekanan serambi tinggi atau bilik kaku dalam pengisian dan jarang terdengar dalam dewasa normal.

Frekuensi jantung dapat dibedakan sesuai dengan kegiatan yang dilakukan yaitu saat sedang beristirahat dan saat sedang berolahraga. Detak jantung istirahat adalah detak jantung yang diukur saat istirahat dan tidak setelah melakukan aktivitas. Pada orang dewasa yang sehat, saat sedang istirahat maka denyut jantung yang normal adalah sekitar 60-100 denyut per menit (bpm). Jika didapatkan denyut jantung yang lebih rendah saat sedang istirahat, pada umumnya menunjukkan fungsi jantung yang lebih efisien dan lebih baik kebugaran kardiovaskularnya. (Edward R. Laskowski, M.D, 2015). Tabel 2.1 berikut ini menjelaskan bagaimana kriteria detak jantung manusia berdasarkan umur dan jenis kelamin pada saat istirahat berdasarkan Top End Sport Resting Heart Rate Table tahun 2013.

Tabel 2.1 Detak jantung per menit saat istirahat

	Detak jantung istirahat perempuan						Detak jantung istirahat laki – laki					
Umur	18-25	26-35	36-45	46-55	56-65	65+	18-25	26-35	36-45	46-55	56-65	65+
Kondisi												
Sangat baik sekali	56-60	54-59	54-59	54-60	54-65	54-59	49-55	49-54	50-56	50-57	51-56	50-55
Sangat baik	61-65	60-64	60-65	61-65	60-64	60-64	56-61	55-62	57-62	58-63	57-61	56-61
Baik	66-89	65-88	65-89	66-89	65-88	65-88	62-85	62-85	63-86	64-87	62-87	62-85
Lebih dari cukup	90-93	89-92	90-93	90-93	89-93	89-92	86-89	86-90	87-90	88-91	88-91	86-89
Cukup	94-98	93-96	94-97	94-97	94-97	93-96	90-93	92-94	91-95	92-96	92-95	90-93
Kurang	99-104	97-102	99-104	98-103	98-103	97-104	94-101	95-101	96-102	97-103	96-101	94-99
Buruk	105+	103+	105+	104+	104+	104+	102+	102+	103+	104+	102+	100+

Detak jantung seseorang saat beraktifitas misalnya sedang berolahraga memiliki batasan yang berbeda-beda berdasarkan umur. Adanya batasan maksimal dalam beraktifitas dapat dihitung dengan menghitung denyut nadi pada pergelangan tangan.

Denyut nadi maksimal adalah maksimal denyut nadi yang dapat dilakukan pada saat melakukan aktivitas maksimal. (gustia, 2011) Menurut DR Suhantoro, cara yang aman adalah mengukur denyut nadi maksimal (DNM). DNM adalah denyut nadi maksimal yang dihitung berdasarkan rumusan $DNM = 220 - \text{Umur}$, kemudian dikalikan dengan intensitas membakar lemak 60-70 persen DNM.

$DNM = 220 - \text{umur}$. Kemudian mengkalikan dengan 60 dan 70 persen. Hasil dari perkalian 60 persen adalah batas ringan, sedangkan hasil perkalian 70 persen adalah batas atas. Misalnya, seseorang dengan usia 23 tahun.

$$\begin{aligned} \text{DNM} &= 220 - 23 \\ &= 197 \text{ Bpm} \end{aligned}$$

$$197 \times 60\% = 118,2 \text{ atau } 118 \text{ Bpm}$$

$$197 \times 70\% = 137,9 \text{ atau } 138 \text{ Bpm}$$

Artinya, seseorang dengan usia 23 tahun harus berhenti sejenak saat berolahraga apabila detak jantungnya sudah lebih dari 138 Bpm.

2.1.7 Sistem Monitoring Detak Jantung Manusia

Topik tentang menghitung maupun memantau sinyal denyut jantung ini telah banyak dibahas dalam bentuk tulisan dan penelitian, Jurnal yang dibahas (Isnaeni, 2008) mengenai pembuatan alat yang dapat merekam denyut jantung berbasis komputer menggunakan sensor elektroda yang mampu menangkap sinyal denyut jantung. (Harahap, 2013) tentang sistem pengukuran detak jantung manusia menggunakan media online dengan jaringan *WI-FI* berbasis *PC*. (Galuh, 2014) tentang alat pengukur detak jantung menggunakan *pulse sensor* berbasis *arduino uno R3* yang diintegrasikan menggunakan *bluetooth*.

Dari beberapa topik tersebut dapat disimpulkan bahwa sistem monitoring detak jantung manusia adalah sistem pengukuran detak jantung manusia yang dilakukan secara terus – menerus persatuan waktu yang sudah ditetapkan BPM (Beat PerMinute) dengan menggunakan suatu alat bantu yaitu prototipe sistem monitoring detak jantung manusia yang kemudian hasil pengujiannya dapat dilihat sebagai data.

2.1.8 Smartphone

Smartphone atau dalam bahasa Indonesia berarti ponsel cerdas merupakan sebuah telepon genggam yang kecanggihannya hampir menyerupai komputer. Ponsel cerdas dapat dibedakan dengan telepon genggam biasa dengan dua cara fundamental, yakni bagaimana mereka dibuat dan apa yang mereka bisa lakukan. (Wood, 2011)

Bagi kebanyakan orang, ponsel cerdas hanyalah sebuah perangkat keras yang dioperasikan dengan menggunakan perangkat lunak. Dengan banyaknya aplikasi yang tersedia bahkan terus berkembang menyajikan kemudahan-kemudahan bagi manusia yang menggunakannya. Tidak hanya itu, kecanggihan pada ponsel pintar ini dapat memangkas banyaknya waktu yang ditempuh untuk melakukan suatu hal. Seperti misalnya, surat elektrik atau yang sering kita kenal dengan nama *email*. Dapat dikirim oleh siapa saja, kapan saja dan dimana saja. Bahkan tidak memerlukan jasa kurir pengirim surat biasa yang terbilang jauh lebih mahal ketimbang mengirim email.

Sistem operasi yang dapat ditemukan di ponsel cerdas adalah *Symbian OS*, *iOS*, *RIM BlackBerry*, *Windows Mobile*, *Linux*, *Palm*, *WebOS* dan *Android*. *Android* dan *WebOS* dibuat oleh *Linux*, dan *iOS* dibuat oleh *BSD* dan sistem operasi *NeXTSTEP* berhubungan dengan *Unix*. Sedangkan perkembangan sistem operasi yang paling banyak digunakan manusia khususnya di Indonesia sendiri adalah *Android*.

2.1.9 Android

Menurut Teguh Arifianto (2011 : 1), android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Hermawan (2011 : 1), Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

Beberapa *smartphone* dengan sistem operasi android dapat di lihat pada gambar 2.1 di bawah ini.



Gambar 2.1 *smartphone* dengan sistem operasi Android

Android memiliki empat karakteristik sebagai berikut :

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan *open source*, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools

yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Beberapa fitur utama dari Android antara lain WiFi *hotspot*, *Multi-touch*, *Multitasking*, GPS, *accelerometers*, support java, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, *Bluetooth HC 05*, Wi-Fi, LTE & WiMAX) serta juga kemampuan dasar *handphone* pada umumnya

2.1.10 Arduino Uno R3

Arduino UNO R3 adalah arduino dengan mikrokontroller ATmega328. Mikrokontroler (Syahwil, 2013) adalah sebuah sistem komputer fungsional dalam sebuah *chip*, yang mempunyai masukan dan keluaran kendali dengan program yang bisa ditulis dan dihapus dengan cara membaca dan menulis data. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program atau keduanya) dan perlengkapan *input output*. Mikrokontroler dibedakan spesifikasinya berdasarkan produsen dan tipenya.

Bentuk dari arduino uno R3 itu sendiri dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2 Arduino Uno R3

Berikut adalah spesifikasi Arduino UNO dengan Mikrokontroler ATmega 328 (Syahwil, 2013):

1. Sumber (Catu Daya)

Arduino dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal yang dapat berasal dari adapter AC-DC atau baterai. Batas tegang *input* yang digunakan adalah 6 – 20 Volt, namun disarankan tegangan *input* yang digunakan berkisar 7-12 Volt saja, karena jika lebih dari 12 Volt maka *board* akan cepat panas dan rusak.

Adapun pin *power supply* pada arduino UNO adalah :

- a. Vin : pin tegangan *input* eksternal (baterai atau adaptor)
- b. 5 V : Keluaran pin yang diatur 5 V, dapat digunakan untuk mengaktifkan modul.
- c. 3v3 : Suplai 3,3 Volt dengan arus maksimum 50 mA.
- d. GND : pin Ground yang terdapat 3 pin.
- e. IOREF. Pin IOREF mendukung Arduino untuk menggunakan tegangan referensi yang digunakan mikrokontroler.

2. MemoriPin

Memori yang digunakan Arduino UNO adalah memori mikrokontroler ATmega 328 dengan memori 32 KB (0.5 KB digunakan untuk *bootloader*) juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang mana dapat dibaca tulis dengan *library* EEPROM).

- a. *Input* dan *Output*

Arduino UNO memiliki 14 pin digital *input/output* dengan arus DC 20 mA.

Masing-masing pin memiliki fungsi – fungsi khusus diantaranya:

1. Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (Rx) dan mengirimkan (TX) data serial TTL. Pin serial terhubung ke pin yang sesuai dari chip ATmega8U2 USB to TTL Serial.
2. Interupsi Eksternal : 2 dan 3. Pin Interupsi Eksternal dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, tepi naik atau perubahan nilai.
3. PWM : terdapat 6 pin 8-bit *Output* PWM yaitu pin 3, 5, 6, 9, 10, 11.
4. SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin SPI mendukung komunikasi SPI menggunakan *library SPI*.

2.1.11 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah suatu program spesial yang memperbolehkan pengguna untuk menulis sketsa program pada papan Arduino dengan model bahasa *Processing*, dimana ketika pengguna *upload* sketsa program pada *board* Arduino, kode-kode tersebut diterjemahkan dalam bahasa C sehingga dapat dilewatkan ke mikrokontroler AVR (Banzi, 2008: 20).

Hal ini menjadi sangat penting karena dengan perubahan bahasa C pengguna pemula yang sulit menggunakan bahasa C akan dipermudah dalam pemrograman mikrokontroler. Penggunaan inisialisasi awal, *looping* dan penggunaan variabel menjadi kode program yang sering digunakan dalam Arduino IDE.

Arduino IDE pertama dibuat dalam versi 1.0, kini seiring bertambahnya berbagai simulasi yang menggunakan arduino, Arduino IDE terus diperbaharui untuk menunjang perangkat-perangkat atau modul yang akan dipasang pada Arduino. Gambar 2.3 merupakan contoh tampilan program pada Arduino IDE.

```

void setup(){
  pinMode(blinkPin,OUTPUT); // pin that will blink to your
  pinMode(fadePin,OUTPUT); // pin that will fade to your
  Serial.begin(9600);
  while (!Serial) {
  }
  bluetooth.begin(9600);
  interruptSetup(); // sets up to read Pulse Senso
}

void loop(){
  //Serial.println("Processing('S', Signal);
  if (QS == true){
    fadeRate = 255;
    Serial.println("Pulse " +

```

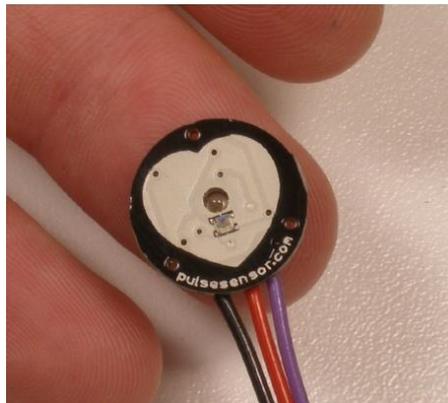
Gambar 2. 3 Tampilan Sketsa Arduino IDE

Program arduino terdiri dari dua blok. Blok pertama adalah *void setup()* dan blok kedua adalah *void loop()*. Berikut keterangan dari masing-masing blok:

- Blok *void setup ()*: berisi kode program yang hanya dijalankan sekali sesaat setelah arduino di hidupkan atau di-*reset*. Merupakan bagian persiapan atau inisialisasi program
- Blok *void loop()*; berisi kode program yang akan di jalankan terus menerus. Merupakan tempat untuk program utama.

2.1.12 *Pulse sensor*

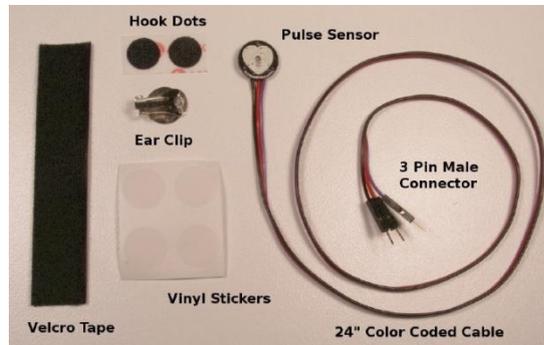
Sensor yang digunakan dalam pembuatan prototipe sistem monitoring detak jantung manusia ini adalah *Pulse sensor*. *Pulse sensor* adalah sensor detak jantung yang dirancang khusus untuk Arduino. *Pulse sensor* ini merupakan kategori sensor yang *easy-to-use* karena sifatnya yang *open source* (Gitman : 2013). Kemudahan lain dari sensor ini adalah 3 kabel utama yang dapat langsung dihubungkan ke Arduino untuk memprogram data. Selain itu, sensitivitas yang dimiliki oleh sensor ini bahkan dapat digunakan untuk mendeteksi detak jantung dari pembuluh darah terkecil sekalipun, misalnya pada ujung jari atau daun telinga. Gambar 2.4 di menunjukkan bentuk fisik dari *pulse sensor*.



Gambar 2.4 *Pulse sensor*

Pada *Pulse sensor* digunakan LED berwarna hijau, karena sensor cahaya yang digunakan yaitu APDS-9008 memiliki puncak sensitivitas sebesar 565nm. Dalam hal ini LED hijau memiliki panjang gelombang 495-570 nm sehingga sesuai dengan kebutuhan sensor tersebut

Di pasaran, biasanya *pulse sensor* dijual lengkap dengan beberapa kit atau perlengkapan pendukung seperti yang ditunjukkan pada gambar 2.5.



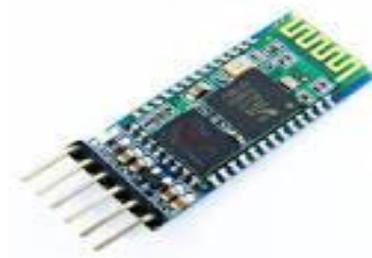
Gambar 2.5 Kit *Pulse sensor*

Perlengkapan yang termasuk dalam *pulse sensor* antara lain:

1. Kabel warna. Kabel dengan panjang sekitar 24 inci sudah dilengkapi dengan *male header connector* yang dapat dihubungkan langsung ke Arduino tanpa solder.
2. Klip telinga. Dengan disertakannya klip telinga dalam perlengkapan *pulse sensor* ini membuat pengguna tidak perlu mencari lagi klip yang cocok untuk digunakan pada sensor.
3. 2 buah Velcro Dots. Digunakan untuk mengikatkan *Pulse sensor* pada ujung jari kita. Kita tidak perlu lagi mencari alat perekat lain untuk memasang sensor ini.
4. 3 buah Stiker Transparan. Stiker dapat digunakan pada bagian depan *Pulse sensor* untuk melindunginya dari jari kita yang berminyak dan telinga kita yang berkeringat.
5. *Pulse sensor* memiliki 3 lubang di sekitar tepi luar yang dapat memudahkan kita untuk menjahitnya.

2.1.13 *Bluetooth HC-05*

Modul *bluetooth* HC 05 digunakan untuk menghubungkan ke sistem GPS, komputer, Laptop, ponsel dan banyak lagi. Dimana fungsi dari *bluetooth* HC 05 ini digunakan untuk mengontrol suatu alat elektronik yang kemudian dapat dikontrol dalam jangkauan sekitar 30 kaki atau lebih tapi itu benar-benar tergantung pada banyak variabel lain. Hal ini memungkinkan prototipe sistem monitoring detak jantung manusia agar dapat menerima basis data hasil pemeriksaan. Tampilan modul *bluetooth* HC 05 dapat dilihat pada gambar 2.6.

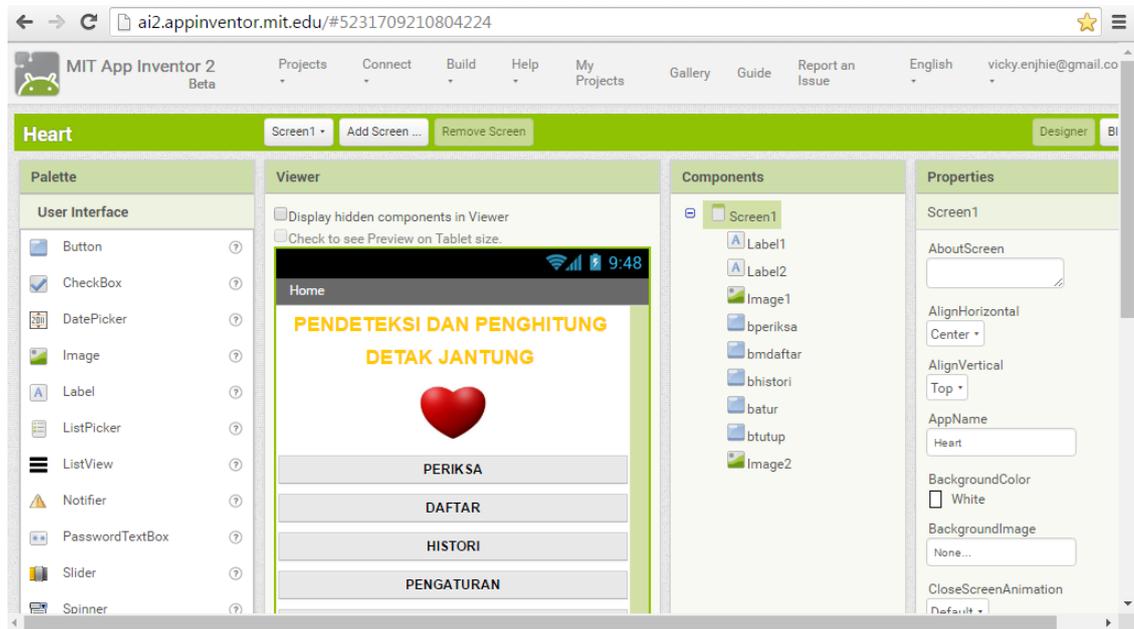


Gambar 2.6 modul *Bluetooth HC 05*

2.1.14 *App Inventor*

App Inventor Android atau *Google App Inventor* adalah aplikasi berbasis web open source yang awalnya dikembangkan oleh Google, dan saat ini dikelola oleh *Massachusetts Institute of Technology* (MIT) (Hilmi, 2014). *App Inventor* memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi Android. *App Inventor* menggunakan antarmuka grafis yang memungkinkan pengguna men-drag-and-drop objek visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat Android. Begitupun dengan *coding*, kita tidak perlu menulis kode program yang amat

sangat panjang, cukup dengan men-drag-and-drop seperti halnya menyusun *puzzle*. Tampilan *App Inventor* dapat dilihat pada gambar 2.7.



Gambar 2.7 Tampilan *App Inventor*

2.1.15 Basis Data (*Database*)

Basis data (*database*) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media (Kadir, 1999). Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya.

Terdapat beberapa aturan yang harus dipatuhi pada file basis data agar dapat memenuhi kriteria sebagai suatu basis data, yaitu:

1. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada file basis data.

2. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada field yang sama untuk beberapa file dengan kunci yang sama.
3. Data terisolasi, disebabkan oleh pemakaian beberapa file basis data. Program aplikasi tidak dapat mengakses file tertentu dalam sistem basis data tersebut, kecuali program aplikasi dirubah atau ditambah sehingga seolah-olah ada file yang terpisah atau terisolasi terhadap file yang lain.
4. Keamanan data, berhubungan dengan masalah keamanan data dalam sistem basis data. Pada prinsipnya file basis data hanya boleh digunakan oleh pemakai tertentu yang mempunyai wewenang untuk mengakses.
5. Integrasi data, berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

Membangun basis data adalah langkah awal dari pembuatan sebuah aplikasi. Keberhasilan dalam membangun basis data akan menyebabkan program lebih mudah dibaca, mudah dikembangkan dan mudah mengikuti perkembangan perangkat lunak. Berikut ini diuraikan mengenai komponen-komponen yang terdapat dalam basis data.

- a. Tabel

Tabel adalah kumpulan dari suatu field dan record. Dalam hal ini biasanya field ditunjukkan dalam bentuk kolom dan record ditunjukkan dalam bentuk baris.

b. Field

Field adalah sebutan untuk mewakili suatu record. Misalnya seorang pegawai dapat dilihat datanya melalui field yang diberikan padanya seperti nip, nama, alamat, dan lain-lain.

c. Record

Record adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu isi data secara lengkap. Satu record mewakili satu data atau informasi tentang seseorang misalnya, nomor daftar, nama pendaftar, alamat, tanggal masuk.

d. Primary Key

Primary key adalah suatu kolom (field) yang menjadi titik acuan pada sebuah tabel, bersifat unik dalam artian tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut, dan dalam satu tabel hanya boleh ada satu primary key.

e. Foreign Key

Foreign key atau disebut juga kunci relasi adalah suatu kolom dalam tabel yang digunakan sebagai “kaitan” untuk melengkapi satu hubungan yang didapati dari tabel induk, dan biasanya hubungan yang terjalin antar tabel adalah satu ke banyak (one to many).

f. Index

Index adalah struktur basis data secara fisik, yang digunakan untuk optimalisasi pemrosesan data dan mempercepat proses pencarian data.

Basis data adalah kumpulan file yang saling berkaitan. Pada model data relasional hubungan antar file direlasikan dengan kunci relasi (relation key), yang merupakan kunci utama dari masing- masing file. Perancangan basis data yang tepat akan menyebabkan paket program relasional akan bekerja secara optimal.

Relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam. Demikian pula untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan attribute dalam satu file.

1. One to one relationship

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan table dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.

2. One to many relationship

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda pada salah satu hubungan.

3. Many to many relationship

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda pada kedua hubungan tersebut.

2.1.16 *TinyDB*

TinyDB adalah komponen non terlihat (*non-visible*) pada *drawer* dasar desain. *TinyDB* digunakan untuk menyimpan data yang akan tersedia setiap kali aplikasi dijalankan. *TinyDB* sangat sederhana, hanya membutuhkan *tag* (nama) dan nilai. Setelah user keluar dari aplikasi dan menjalankannya kembali, nilai tersebut masih ada di *TinyDB* dan dapat diambil dengan menggunakan *tag*.

Sebagai ilustrasi, *tag* adalah nama *file* dan data adalah sebuah *file*. Cara memuat data kembali ke sebuah aplikasi adalah seperti memuat *file*. Sebut itu dengan nama, dan diambil dari *TinyDB* kemana variabel yang sudah kita tentukan.

Setiap satu *TinyDB* hanya dapat digunakan sebagai penyimpanan data per aplikasi. Apabila dalam dua aplikasi menggunakan satu *TinyDB* maka aplikasi tidak akan bekerja sebagaimana yang sudah dirancang sebelumnya. Pengaplikasian *TinyDB* dalam *App Inventor* dapat dilihat pada gambar 2.8.



Gambar 2.8 pengaplikasian *TinyDB* pada *App Inventor*

2.1.17 Metode *Research and Development* (R&D)

2.1.17.1 Definisi

Metode *Research and Development* atau yang berarti Metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut (Sugiyono, 2011: 297).

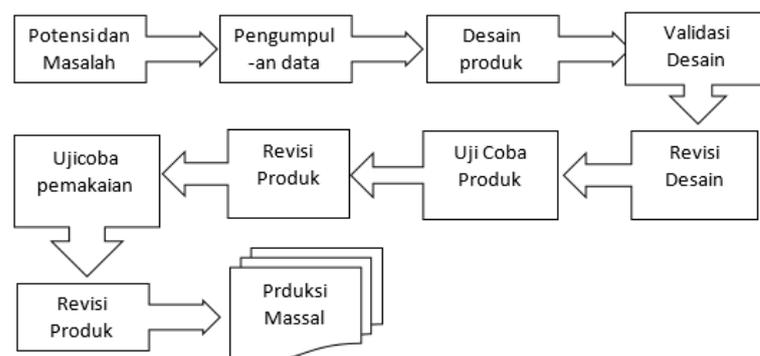
National Science Board dalam “*Research and Development Essential Foundation for U.S Competitiveness in A Global Economy*” (2008 : *endnotes*) dalam buku Putra (2011: 68) menguraikan Definisi Penelitian dan Pengembangan (R&D) :

“R&D adalah metode penelitian yang secara sengaja, sistematis, bertujuan / diarahkan untuk mencaritemukan, merumuskan, memperbaiki, mengembangkan, menghasilkan, menguji keefektifan produk, model, metode / strategi / cara / jasa.”

Berdasarkan definisi-definisi yang telah dikemukakan di atas maka dapat disimpulkan bahwa R&D adalah suatu metode yang membutuhkan pengetahuan untuk mengembangkan, menghasilkan suatu produk secara efektif.

2.1.17.2 Langkah - langkah R&D

Langkah – langkah *Research and Development* (R&D) ditunjukkan pada gambar 2.9.



Gambar 2.9 Langkah – langkah Metode R&D

2.2 Kerangka Berpikir

Dengan landasan teori yang sudah ditulis pada halaman sebelumnya, pada rancang bangun prototipe sistem monitoring detak jantung manusia menggunakan

Smartphone Android Berbasis Mikrokontroler Arduino Uno, dapat diketahui bahwa Arduino Uno R3 merupakan komponen utama. Arduino berperan sebagai otak dalam prototipe sistem monitoring detak jantung manusia. Arduino Uno R3 diaktifkan melalui sumber tegangan AC. Dari sumber tegangan tersebut, Arduino akan mensuplai tegangan untuk *pulse sensor* dan *bluetooth* HC 05. Pulse sensor akan diberi tegangan oleh Arduino Uno R3 sebesar 5V dan *bluetooth* HC 05 mendapatkan tegangan sebesar 3,3 V. Ketika semua komponen telah aktif, maka alat akan mulai bekerja. *Pulse sensor* akan mendeteksi detak jantung pada manusia dan kemudian data itu diproses oleh Arduino Uno R3. Ketika proses perhitungan detak jantung telah selesai, data kemudian dikomunikasikan secara serial dan dikirimkan melalui *Bluetooth* HC 05 menuju *Smartphone* Android. Untuk dapat memahami rancang bangun prototipe sistem monitoring detak jantung manusia terlebih dahulu menentukan jenis-jenis komponen pendukung yang dibutuhkan dalam pembuatannya. Adapun tahapan-tahapan yang akan dilakukan adalah sebagai berikut:

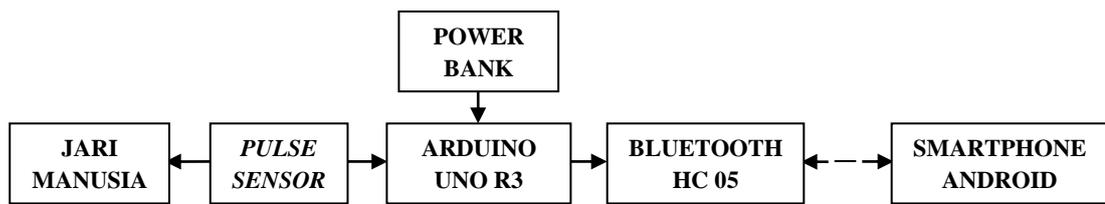
- 1) Mempelajari pengukuran detak jantung manusia.
- 2) Mendesain rancangan bangun prototipe sistem monitoring detak jantung manusia menggunakan komponen utama *pulse sensor*, arduino, modul *bluetooth* HC 05 dan basis data *TinyDB*.
- 3) Merealisasikan rancangan prototipe sistem monitoring detak jantung manusia dengan menggunakan *pulse sensor* sebagai sensor utama yang kemudian diproses oleh Arduino Uno R3. Hasil penghitungan data dikirim oleh *Bluetooth* HC 05 ke penyimpanan basis data *TinyDB* pada

aplikasi *App Inventor* yang sudah di desain sehingga basis data dapat dilihat pada *smartphone* android.

- 4) Menguji rancangan prototipe sistem monitoring detak jantung manusia pada rancangan yang telah direalisasikan sehingga dapat digunakan untuk menyimpan hasil monitoring detak jantung manusia.

2.2.1 Blog Diagram

Sebelum membuat prototipe sistem monitoring detak jantung manusia, terlebih dahulu merancang susunan keseluruhan sistem. Rancangan dapat dilihat pada Gambar 2.10.



Gambar 2.10 Diagram Blog prototipe sistem monitoring detak jantung manusia

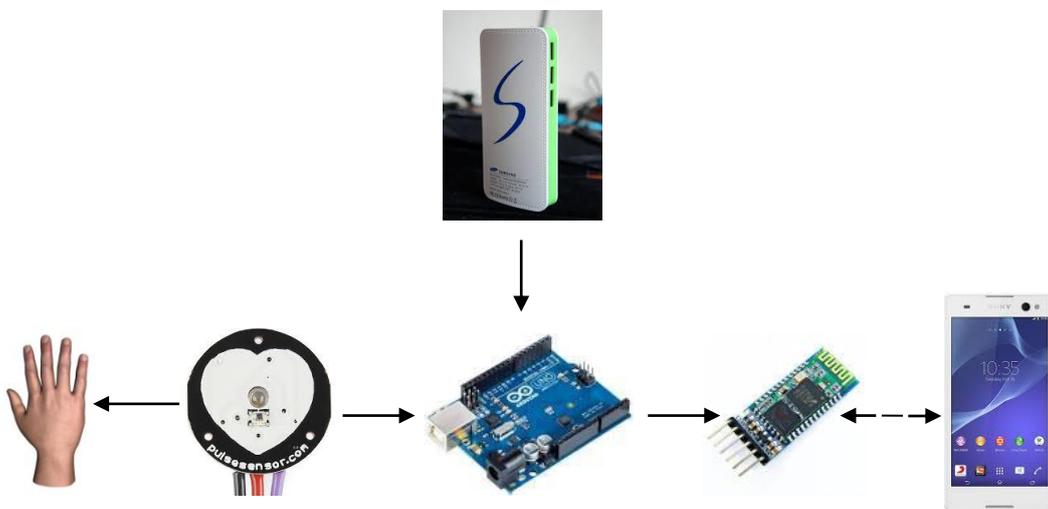
Berdasarkan gambar 2.10 maka rancang bangun prototipe sistem monitoring detak jantung manusia dapat dijelaskan sebagai berikut:

1. Jari manusia merupakan masukan yg dideteksi oleh *pulse sensor*.
2. *Pulse sensor* digunakan sebagai masukan pendeteksi detak jantung manusia. Masukan tersebut merupakan masukan sinyal analog yang nanti akan diproses di Arduino.
3. Power Bank sebagai *power supply unit*.
4. Arduino melakukan proses *coding* untuk pembacaan data yang diberikan oleh *pulse sensor* yang kemudian disesuaikan dengan perubahan data yang

diinginkan. Proses *coding* pada arduino dilakukan dengan bahasa pemrograman IDE (*Integrated Development Environment*).

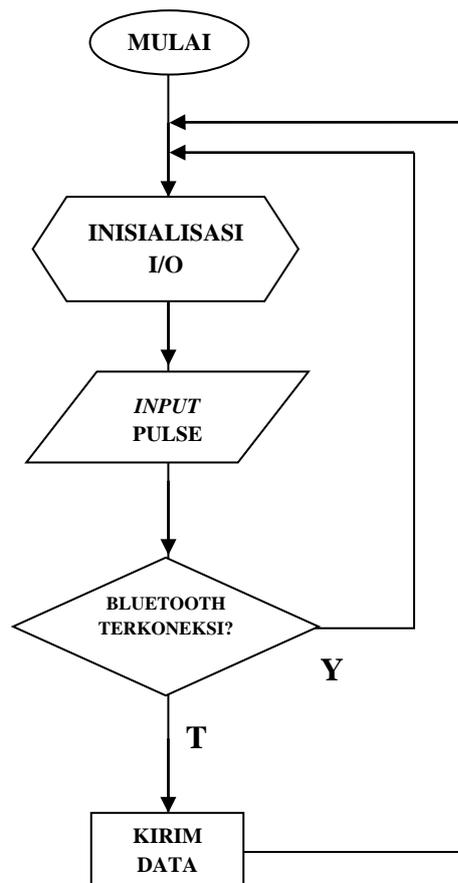
5. *Bluetooth* HC 05 digunakan sebagai perantara antara Arduino dengan *smartphone* Android dalam komunikasi data hasil pengukuran detak jantung manusia.
6. *Smartphone* Android digunakan untuk melihat hasil pengukuran detak jantung manusia yang sudah dikomunikasikan oleh modul *bluetooth HC 05*.

Selain blok diagram, adapun blok komponen yang dirancang pada prototipe sistem monitoring detak jantung manusia, berikut **Gambar 2.11** blok komponen di bawah ini.



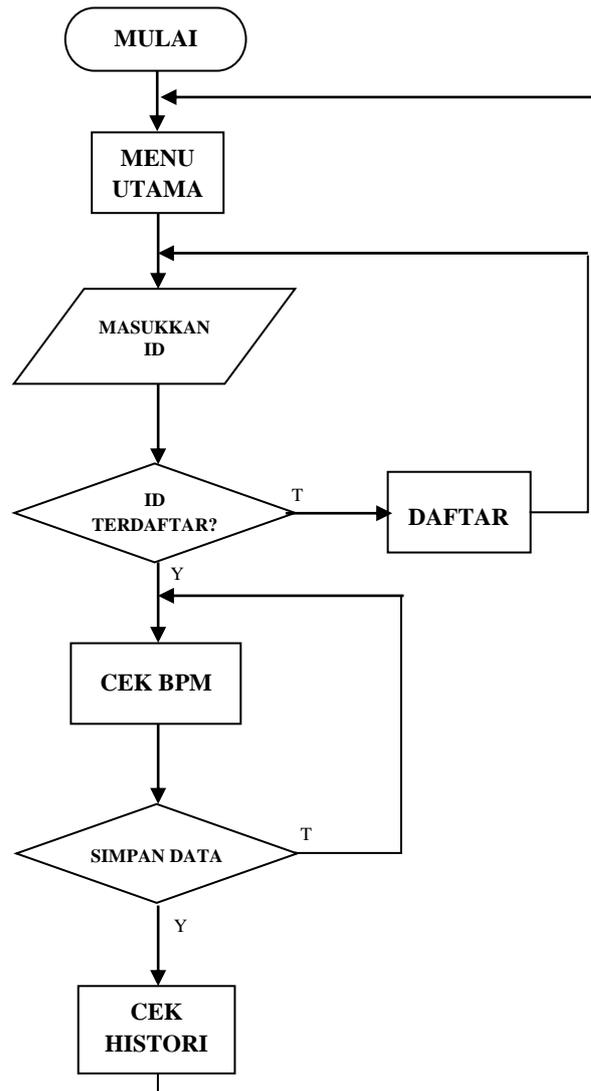
Gambar 2.11 blok komponen prototipe sistem monitoring detak jantung manusia

2.2.2 Flowchart



Gambar 2.12 Flowchart arduino prototipe sistem monitoring detak jantung manusia

Dalam flowchart pada gambar 2.12 yang menunjukkan sistem arduino dari prototipe sistem monitoring detak jantung manusia ini menjelaskan bahwa langkah awal dalam pengoperasian alat tersebut yaitu dengan melakukan inisialisasi I/O yang menghubungkan Arduino dengan *Smartphone* Android untuk mengirim data hasil deteksi detak jantung manusia oleh *Pulse sensor*. Barulah setelah itu *pulse sensor* mendeteksi adanya detak jantung pada tubuh manusia kemudian menghitungnya sesuai dengan acuan BPM. Setelah itu data dikirim melalui data serial dan *bluetooth* HC 05. Alat pendeteksi dan penghitung detak jantung manusia ini akan terus beroperasi selama alat belum dimatikan.



Gambar 2.13 Flowchart aplikasi android prototipe sistem monitoring detak jantung manusia

Penjelasan dari flowchart pada Gambar 2.12 dan 2.13 adalah sebagai berikut; untuk mengoperasikan aplikasi ini pengguna harus memasukkan id, apabila belum terdaftar dan tidak memiliki id maka pengguna harus melakukan pendaftaran id.

Selanjutnya pengguna akan langsung dapat melakukan cek detak jantung permenit, apabila pengguna ingin menyimpan data hasil cek tersebut dapat langsung disimpan sebagai data yang dapat dilihat di menu histori pada aplikasi tersebut.

2.3 Hipotesis Penelitian

Pada hipotesis penelitian ini yaitu rancang bangun prototipe sistem monitoring detak jantung manusia ini akan berkerja dengan *input* dari *pulse sensor* yang dapat mendeteksi detak jantung manusia dengan arduino uno R3 sebagai pengolah data yang kemudian dikomunikasikan secara serial oleh *bluetooth* HC 05 sehingga hasil pemeriksaan dapat dilihat pada aplikasi yang dibuat dengan menggunakan App *Inventor* dan kemudian disimpan sebagai basis data pada *TinyDB*.

BAB III

METEDOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Bengkel Mekanik Teknik Elektro Universitas Negeri Jakarta, pada bulan Januari 2016 sampai Juni 2016.

3.2 Metode Pengembangan Produk

3.2.1 Tujuan Pengembangan

Mengacu pada tujuan penelitian dan perumusan masalah yang telah dibahas sebelumnya, maka tujuan pengembangan produk adalah sebagai berikut:

1. Membuat prototipe sistem monitoring detak jantung manusia sebagaimana sistem kerja *Pulse oximeter*.
2. Membuat sistem pendeteksi detak jantung menggunakan *Pulse sensor*.
3. Membuat sistem monitoring detak jantung manusia dengan cara menyimpan histori data pada basis data *TinyDB*.

3.2.2 Metode Pengembangan

Berdasarkan metode penelitian yang sudah dibahas sebelumnya maka peneliti menggunakan metode pengembangan Research dan Development (R&D). Namun, peneliti hanya menggunakan 6 langkah metode R&D sesuai kebutuhan peneliti, yaitu :

1. Potensi dan Masalah
2. Pengumpulan data
3. Desain Produk
4. Validasi Desain

5. Perbaiki Desain
6. Uji Coba Produk

3.2.3 Sasaran Produk

Adapun sasaran dari penelitian ini adalah sebagai berikut :

1. Memberikan pengetahuan dalam pembuatan prototipe sistem monitoring detak jantung manusia menggunakan basis data *TintDB* pada aplikasi App *Inventor*.
2. Media pembelajaran bagi mahasiswa dalam teknik pengukuran dan instrumentasi di dunia elektronika.
3. Memberikan pengetahuan dalam pembuatan alat pendeteksi detak jantung dengan menggunakan *pulse sensor* dan arduino uno R3.

3.2.4 Instrumen

Instrumen yang digunakan dalam penelitian adalah *pulse oximeter* dan avometer. *Pulse oximeter* memiliki batas bawah 30 Bpm dan batas atas 240 Bpm. Sedangkan avometer yang digunakan adalah multimeter digital SANWA CD800a.

3.3 Prosedur Pengembangan

3.3.1 Tahap Penelitian dan Pengumpulan Informasi

Tahap penelitian dan pengumpulan informasi disini merupakan analisis kebutuhan sistem. Kebutuhan suatu sistem pada umumnya yaitu perangkat input dan output yang digunakan dalam sistem tersebut.

Pada penelitian ini, perangkat-perangkat yang dibutuhkan untuk membuat prototipe sistem monitoring detak jantung manusia telah dijelaskan pada bab sebelumnya yang meliputi perangkat pendukung sistem pengendali yaitu Arduino

Uno dan Bluetooth HC-05, *Pulse sensor*, dan basis data *TinyDB* sebagai output yang digunakan dalam alat ini.

3.3.2 Tahap Perencanaan

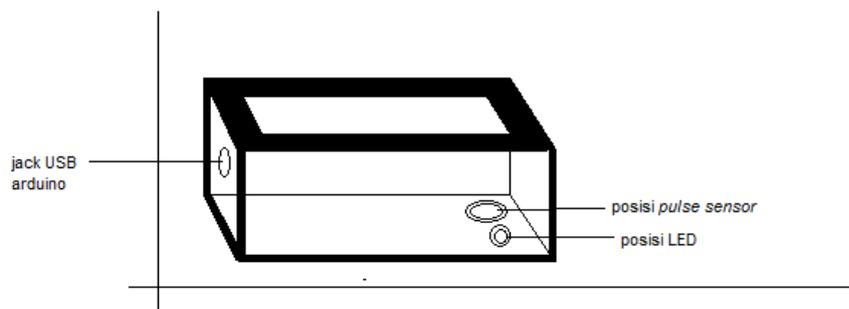
Tahap perencanaan berisi kerangka berpikir peneliti dalam pembuatan prototipe sistem monitoring detak jantung manusia yang telah dijelaskan pada bab sebelumnya meliputi blok diagram yang dapat dilihat pada gambar 2.10, dan flowchart keseluruhan sistem yang terdapat pada gambar 2.12 dan gambar 2.13.

3.3.3 Tahap Desain Prototipe

Tahap desain prototipe sistem monitoring detak jantung manusia merupakan suatu rencana atau gagasan yang mempunyai suatu tujuan yang terarah dalam melakukan suatu penelitian untuk menghasilkan atau meningkatkan karya yang dibuat.

3.3.3.1 Rancangan Perangkat Keras

Pada rancangan desain perangkat keras alat pendeteksi dan penghitung detak jantung harus bersifat *portable* agar mudah dibawa kemana-mana. Dengan menggunakan kotak bekas adaptor dengan ukuran 10 x 6 x 3 cm. Rancangan desain prototipe dapat dilihat pada gambar 3.1 di bawah ini.



gambar 3.1 Rancangan desain prototipe

Sedangkan untuk penggunaan pin pada arduino, dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 Penggunaan Pin pada Arduino Uno

No	Perangkat	Pin perangkat	Pin Arduino Uno
1	<i>Pulse sensor</i>	Sinyal	A0
		GND	GND
		+	5V
2	Bluetooth	RXD	-3
		TXD	2
		3,5	3,5
		GND	GND
3	LED	-	GND
		+	-5

3.3.3.2 Perancangan Perangkat Lunak

Perangkat lunak dalam penelitian ini adalah berupa pemrograman untuk arduino dan aplikasi android. Seperti yang telah dijelaskan pada bab sebelumnya, untuk pemrograman arduino menggunakan perangkat lunak arduino IDE. Sedangkan untuk pemrograman aplikasi android menggunakan aplikasi App *Inventor*.

a. Perancangan Program Arduino

Perancangan program arduino menggunakan *software* IDE Arduino menggunakan bahasa *processing* dalam pemrogramannya yang didapat dari website resmi *pulse sensor*. Perancangan program dibuat berdasarkan flowchart pada gambar 3.2 berikut ini.

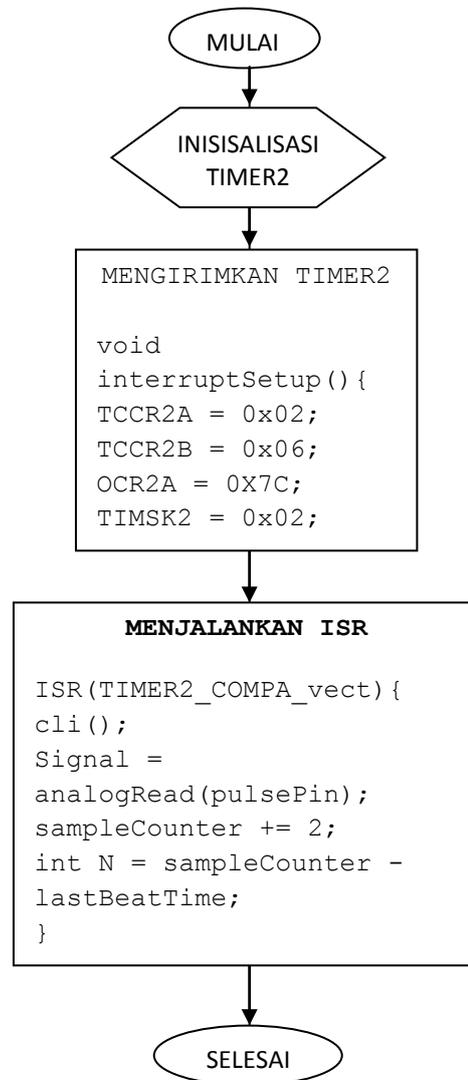


Gambar 3.2 flowchart perancangan program arduino

Untuk penjelasan proses secara lengkap dapat dilihat penjabaran proses beserta penjelasan listing programnya.

a.1 Pengiriman Perintah Program

proses pengiriman perintah program dibuat berdasarkan flowchart pada gambar 3.3 di bawah ini.



gambar 3.3 flowchart pengiriman perintah

Pertama, penting untuk memiliki sample rate dengan resolusi yang cukup tinggi untuk mendapatkan pengukuran waktu yang dapat diandalkan antara setiap denyut. Untuk melakukannya dibuatlah *Timer2*, 8 bit hardware timer pada *Atmega328 (UNO)*, sehingga melempar interupsi setiap milidetik. Hal ini

memberikan tingkat sampel 500Hz dan resolusi waktu *beat-to-beat* 2ms. Ini akan menonaktifkan *PWM* output pada pin 3 dan 11.

```
void interruptSetup(){
// inisialisasi timer 2 untuk melempar/mengirim interrupt
setiap 2mS.
TCCR2A = 0x02;    // menonaktifkan PWM pada pin digital 3 dan
                  // 11, dan menjadi mode CTC
TCCR2B = 0x06; // sebagai pembanding, precaler 256
OCR2A  = 0x7C;  // mengatur dengan puncak perhitungan ke 124
                  // Dengan sample rate 500Hz
TIMSK2 = 0x02;  // mengaktifkan interrupt agar sesuai dengan
                  // TIMER2 dan OCR2A
sei();          // memastikan interrupts global diaktifkan
}
```

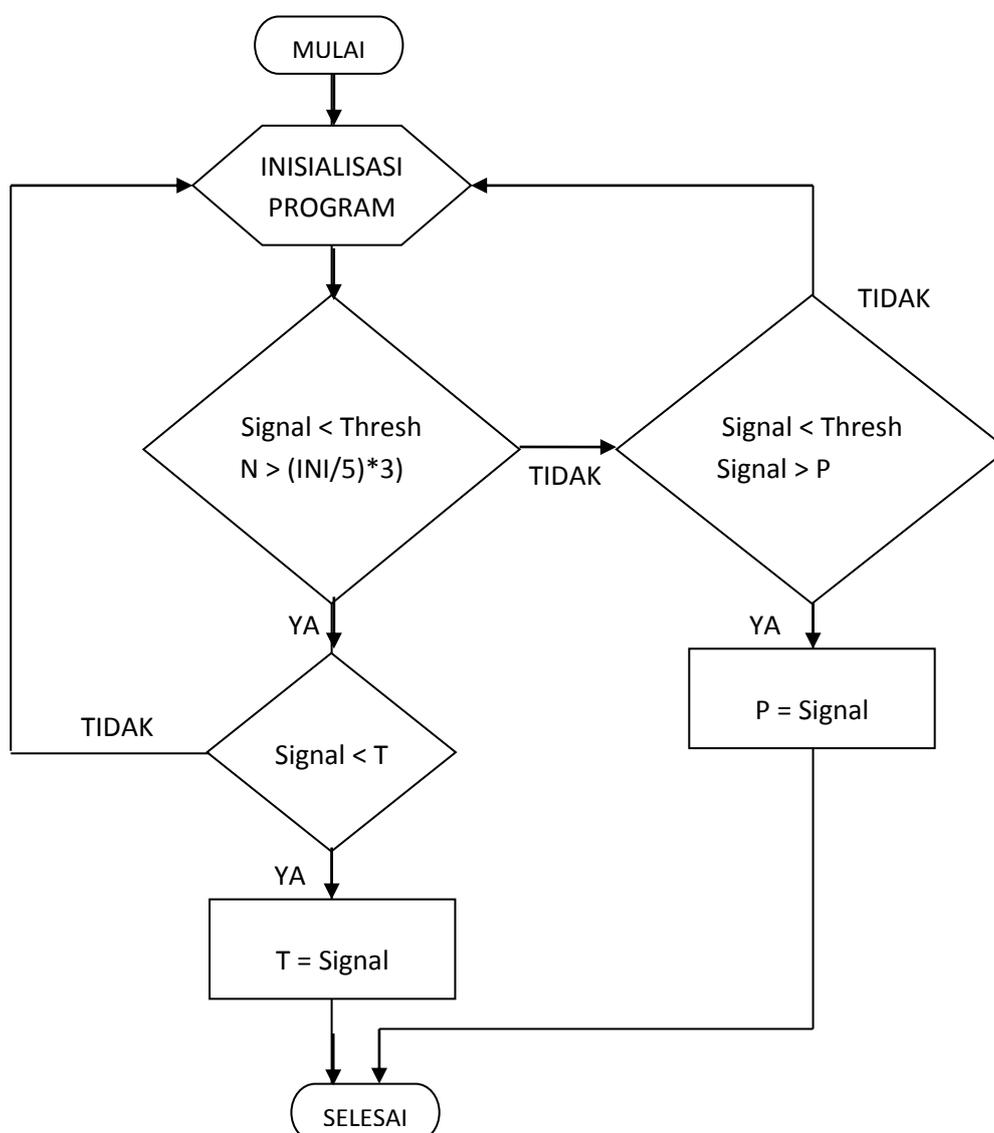
Hal lain yang diperlukan adalah *ISR* vektor yang benar pada langkah berikutnya. Perangkat Arduino menggunakan *ISR (TIMER2-COMPA-vect)* yang fungsinya dipanggil setiap 2 milidetik seperti program di bawah ini.

```
// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.
// Timer 2 akan membaca setiap 2mS
ISR(TIMER2_COMPA_vect){ // trigger ketika Timer2 menghitung
sampai 124
cli();                // menonaktifkan interrupt ketika
                      // berlangsung
Signal = analogRead(pulsePin); // membaca pulse sensor
sampleCounter += 2;      // menjaga waktu pengukuran dalam
                          // mS dengan variabel
int N = sampleCounter - lastBeatTime // memonitoring waktu
                                      // dari detak terakhir
                                      // untuk menghindari
                                      // noise
```

Pertama yang harus dilakukan adalah mengambil pembacaan analog dari *Pulse sensor*. Selanjutnya, pada variabel kenaikan *sampleCounter*. Variabel *sampleCounter* adalah apa yang digunakan untuk malacak waktu. Variabel *N* akan membantu menghindari adanya *noise*.

a.2 Penentuan Batas Titik Puncak dan Lembah Gelombang Pulsa

Selanjutnya kita melacak nilai-nilai tertinggi dan terendah dari gelombang pulsa, untuk mendapatkan ukuran yang akurat dari amplitudo. Gambar 3.4 di bawah ini adalah flowchart penentuan batas gelombang.



Gambar 3.4 Flowchart penentuan batas gelombang

```

// MENENTUKAN PUNCAK DAN LEMBAH GELOMBANG PULSA
if(Signal < thresh && N > (IBI/5)*3){ // menghindari noise
                                                dengan menunggu 3/5
                                                dari IBI terakhir

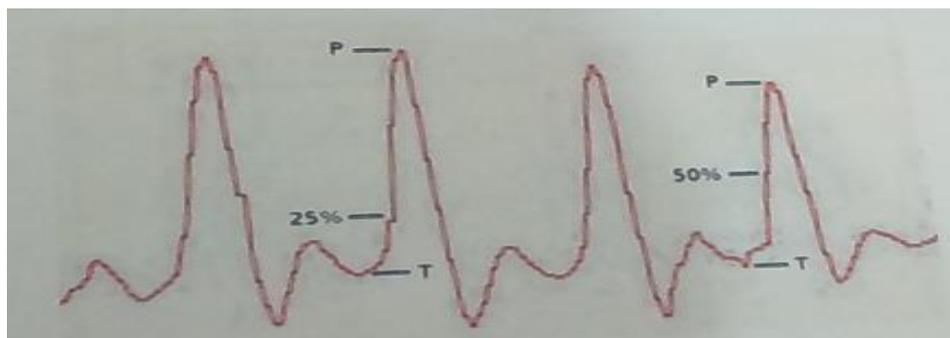
if (Signal < T){ // T adalah lembah
T = Signal;    // menjaga tetap pada poin terendah pada
                gelombang pulsa
        }
}

if(Signal > thresh && Signal > P){ // kondisi thresh membantu
                                    menghindari noise
P = Signal;    // P adalah puncak
} // menjaga agar tetap pada poin tertinggi pada gelombang pulsa

```

Variabel *thresh* diinisialisasi di 512 (titik tengah rentang analog) dan perubahan selama menjalankan waktu untuk melacak titik di 50% dari *amplitude*. Ada jangka waktu 3/5 *IBI* yang harus dilewati sebelum T (*trough*/lembah) untuk menghindari noise dan pembacaan palsu.

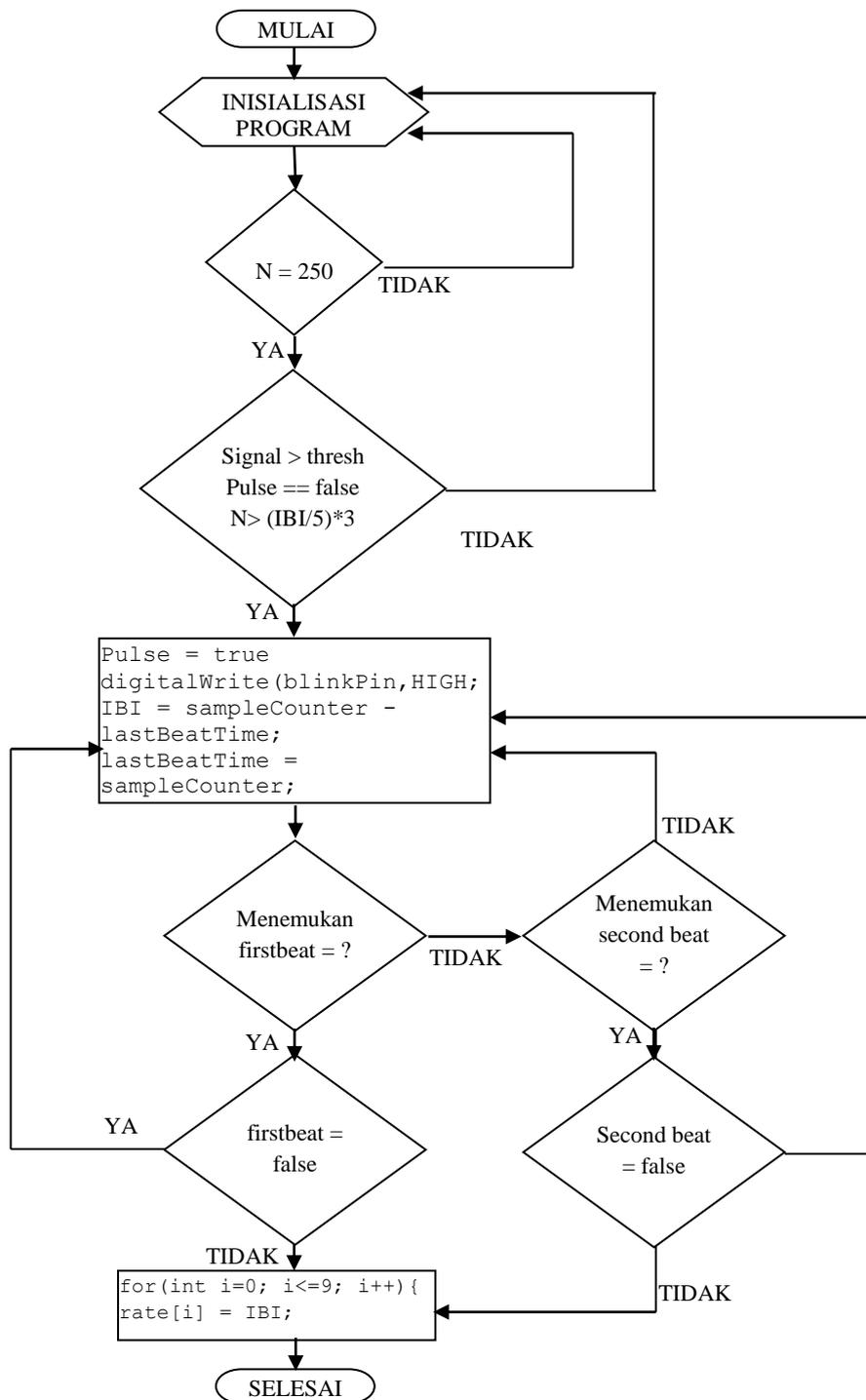
Untuk penggambaran gelombang pulsa pada saat ada di titik puncak maupun lembah dapat dilihat pada gambar 3.5.



Gambar 3.5 gelombang pulsa detak jantung manusia

a.3 Mencari Detak Jantung

Proses pencarian detak jantung dapat dilihat pada gambar 3.6 yaitu gambar flowchart pencarian detak jantung.



Gambar 3.6 flowchart untuk mencari detak jantung

```

// Mencari Detak Jantung
// signal melonjak setiap waktu ada
if (N > 250){ // menghindari noise frekuensi tinggi
if ((Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
Pulse = true; // mengatur pulse flag saat berfikir adanya pulsa
digitalWrite(blinkPin,HIGH); // menghidupkan led pada pin 13
IBI = sampleCounter - lastBeatTime; // mengukur waktu antara
// detak dalam mS
lastBeatTime = sampleCounter; // melacak waktu untuk pulsa
// berikutnya

```

Sebelum mencari detak jantung, harus melewati jumlah minimal waktu. Hal ini membantu menghindari kebisingan/*noise* frekuensi tinggi. 250 milidetik minimum N menempatkan batas atas 240BPM. Ketika gelombang naik melewati nilai “*thresh*” dan $\frac{3}{5}$ dari *IBI* terakhir yang telah berlalu, maka ditemukanlah *first beat*.

Bit berikutnya digunakan untuk memastikan kita mulai dengan nilai BPM realistis pada start up.

```

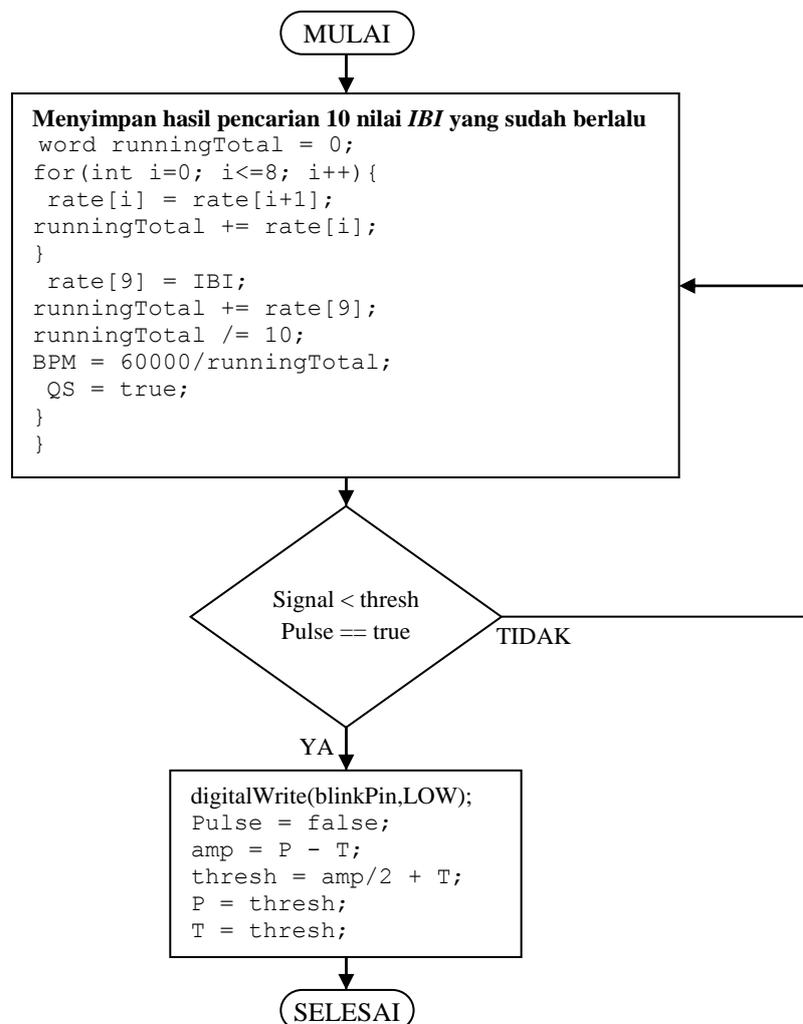
if(secondBeat){ // jika ini adalah bit kedua, maka bit kedua =
// benar
secondBeat = false; // jika salah, maka bersihkan flag bit kedua
for(int i=0; i<=9; i++){ // menjalankan total pencarian untuk
// mendapatkan BPM yang realistis
// pada saat dimulai
rate[i] = IBI;
}
}
if(firstBeat){ // jika ini pertama kalinya menemukan bit maka bit
// pertama = benar
firstBeat = false; // jika salah, maka bersihkan flag bit pertama
secondBeat = true; // mengatur flag bit kedua
sei(); // mengaktifkan interupsi lagi
return; // karena ilai IBI tidak dapat diandalkan maka akan
// dibatalkan
}

```

Boolean first beat diinisialisasi sebagai benar dan *scond beat* diinisialisasi sebagai palsu pada saat memulai. Mulai saat pertama kali ditemukan detak dan sampai sejauh ini dalam *ISR* maka akan dikembalikan oleh “*return*” dalam *firstbeat* bersyarat. Kondisi tersebut akan berakhir dengan membuang pembacaan *IBI* pertama, karena pembacaan dinyatakan salah.

a.4 Menghitung Detak Jantung

Dalam menghitung detak jantung, untuk mengetahui proses pelaksanaannya dapat dilihat pada gambar 3.7 yang merupakan flowchart dari penghitungan detak jantung.



Gambar 3.7 Flowchart untuk menghitung detak jantung

```

// menyimpan hasil pencarian 10 nilai IBI 10 yang sudah berlalu
word runningTotal = 0;    // bersihkan variabel runningTotal
for(int i=0; i<=8; i++){  // pergeseran data dalam tingkat array
rate[i] = rate[i+1];      // drop nilai IBI terakhir
runningTotal += rate[i]; // jumlahkan nilai 9 IBI sebelum terakhir
}
rate[9] = IBI;           // menambahkan IBI terakhir untuk tingkat array
runningTotal+=rate[9];//menambahkan IBI terakhir untuk
                        runningTotal
runningTotal /= 10;     // rata-ratakan 10 nilai IBI terakhir
BPM = 60000/runningTotal; // berapa banya detak dalam setiap
                        menitnya, itulah BPM
QS = true;              // set Quantified Self
}
}

```

Pertama, kita ambil variabel besar `runningTotal` untuk mengumpulkan nilai *IBI* dengan menggunakan fungsi tingkat yang ditambah. Kemudian ditambahkan ke `runningTotal`. Dan proses selanjutnya mencari rata-ratanya. Proses terakhir adalah mencari *BPM* dengan ketentuan 6000ms dibagi dengan rata-rata hasil `running total` tersebut. Kemudian ditemukanlah *BPM* yang telah diukur.

Ada beberapa kondisi lainnya berakhir tida stabil sehingga perlu mengikat off sebelum selesai, seperti meneukan kondisi tidak berdetak.

```

if(Signal<thresh &&Pulse == true){//ketika nilai mulai turun
                        atau menghilang,perhitungan
                        detak berakhir
digitalWrite(blinkPin,LOW); // matikan LED pin 13
Pulse = false; // mereset pulse flag dan dimulai lagi dari awal
amp = P - T; // menentukan amplitudo
thresh = amp/2 + T; // mengatur thresh pada 50% dari amplitudo
P = thresh; // mereset untuk waktu berikutnya
T = thresh;
}

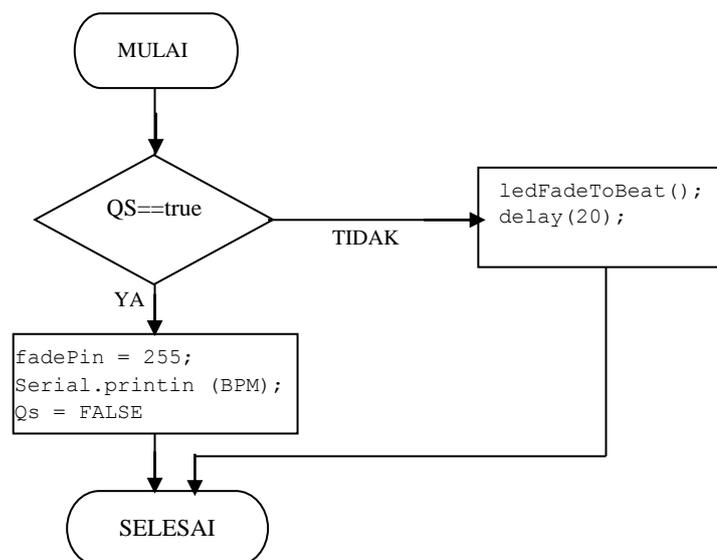
```

Pulse dinyatakan berlaku selama kenaikan-kenaikan sinyal *pulse sensor* menemukan detak. Ketika sinyal melintas turun, bisa dikatakan pulsa berakhir.

Kemudian amplitudo gelombang yang baru saja berlalu diukur dan nilai *thresh* diperbaharui. P dan T direset ke *thresh* baru.

a.5. Pengiriman Data

Dalam pengiriman data *bluetooth* yang digunakan merupakan seri *HC-05* dimana tidak dapat mengawali koneksi ke perangkat lain. Oleh karena itu, *bluetooth* secara terus menerus mengirimkan data hingga ada perangkat lain meminta koneksi untuk berpasangan . Ketika telah ditemukan, maka data digital jumlah detak jantung dalam satuan *BPM* tadi akan diintegrasikan secara serial melalui *bluetooth* menuju *smartphone* android. Kemudian android akan menampilkan data tersebut secara *real time* selama koneksi berlangsung, berikut ini adalah flowchart pengiriman data pada gambar 3.8



Gambar 3.8 flowchart pengiriman data

Dalam *listing* programnya, digunakan perintah “Serial.printin(BPM);”. Perintah tersebut secara otomatis melalui komunikasi serial akan menampilkan hasil perhitungan dari *BPM*.

```

void loop(){
if (QS == true){      // Qualitified self flag benar ketika arduino
                        menemukan detak
fadeRate = 255;      // mengatur variabel 'faderate' menjadi 255 dan
                        melemahkan LED dengan pulsa
Serial.print("BPM=");
Serial.println(BPM);
//bluetooth.print("BPM=");
bluetooth.println(BPM);
QS = false;
    }
ledFadeToBeat();
delay(20);

```

Kemudian untuk pengaturan kecepatan pengiriman maka akan digunakan perintah “`serial.begin(9600)`”. Kecepatan yang mampu digunakan oleh *buetooth HC-05* hanya sebesar *9600 Baudrate*.

```

void setup(){
pinMode(blinkPin,OUTPUT); // pin yang akan berkedip sesuai detak
                            jantung
pinMode(fadePin,OUTPUT);  // pin yang akan melemah (fade) sesuai
                            detak jantung

    Serial.begin(9600);
while (!Serial) {
    }
bluetooth.begin(9600);
interruptSetup(); // sets up untuk membaca sinyal pulse sensor
                  setiap 2mS
}

```

b. Perancangan Aplikasi Android

Perancangan aplikasi android menggunakan aplikasi *App Inventor*. Aplikasi dirancang dengan membuat 5 menu pada *screen home* seperti tampak pada gambar 3.9.



Gambar 3.9 *Screen Home*

Pada *Screen home* terdapat 5 menu diantara lain:

1. Menu periksa

Pada menu periksa terdapat beberapa opsi yang dapat dipilih untuk melakukan pemeriksaan detak jantung diantaranya adalah:

a. Pilih ID

Sebelum melakukan pemeriksaan pengguna harus memilih id yang sudah didaftarkan sebelumnya.

b. Pilih *bluetooth*

Sebelum memulai pemeriksaan, pengguna harus memilih bluetooth yang digunakan di alat yang sudah dibuat yaitu HC 05.

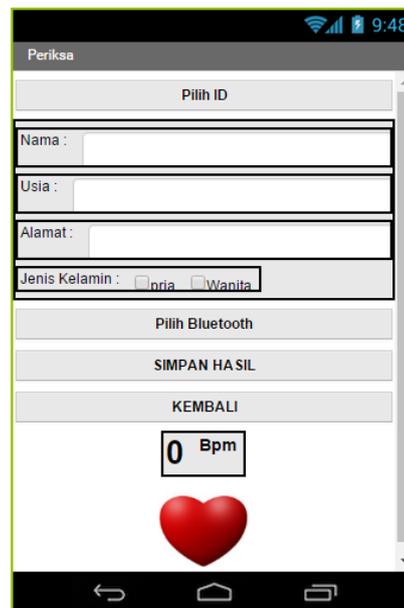
c. Simpan hasil

Setelah melakukan pemeriksaan, pengguna dapat menekan opsi simpan hasil untuk menyimpan hasil pemeriksaan.

d. Kembali

Saat menekan opsi kembali, pengguna akan diarahkan kembali pada screen home.

Tampilan *screen* periksa dapat dilihat pada gambar 3.10 di bawah ini.



Gambar 3.10 Tampilan screen periksa

2. Menu daftar

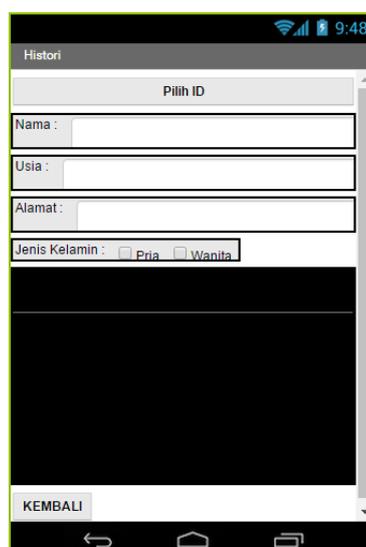
Pada menu daftar, pengguna harus mengisi data diri untuk mendapatkan nomor id, data yg harus diisi diantaranya adalah; nama, usia, alamat dan jenis kelamin. Setelah menekan tombol simpan, pengguna akan mendapatkan nomor id. Tampilan pada menu daftar dapat dilihat pada gambar 3.11 di bawah ini.



Gambar 3.11 Tampilan *screen* daftar

3. Menu histori

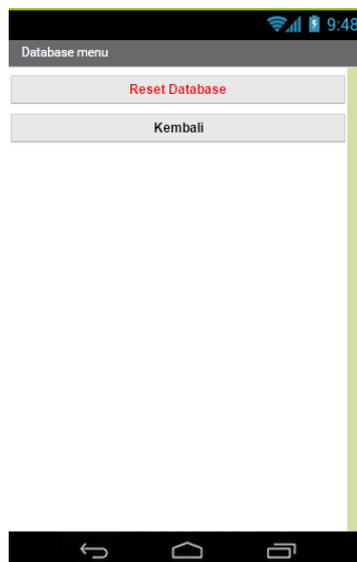
Pada menu histori ada pilihan opsi untuk memilih id pengguna. Saat pengguna sudah memilih id maka akan muncul data diri dari id yg sudah dipilih tersebut, begitu juga dengan data detak jantung yang sudah disimpan sebelumnya. Tampilan *screen* histori dapat dilihat pada gambar 3.12 di bawah ini.



Gambar 3.12 Tampilan *screen* histori

4. Menu pengaturan

Pada menu pengaturan terdapat opsi untuk *me-reset database*. Hal ini diperlukan agar data tersimpan dapat dihapus seluruhnya. Dengan menghapus seluruh data, maka nomor id akan mengulang kembali ke nomor 1000 saat memulai kembali melakukan pendaftaran. Tampilan pada *screen* pengaturan dapat dilihat pada gambar 3.13 di bawah ini.



Gambar 3.13 Tampilan *screen* pengaturan

5. Menu keluar aplikasi

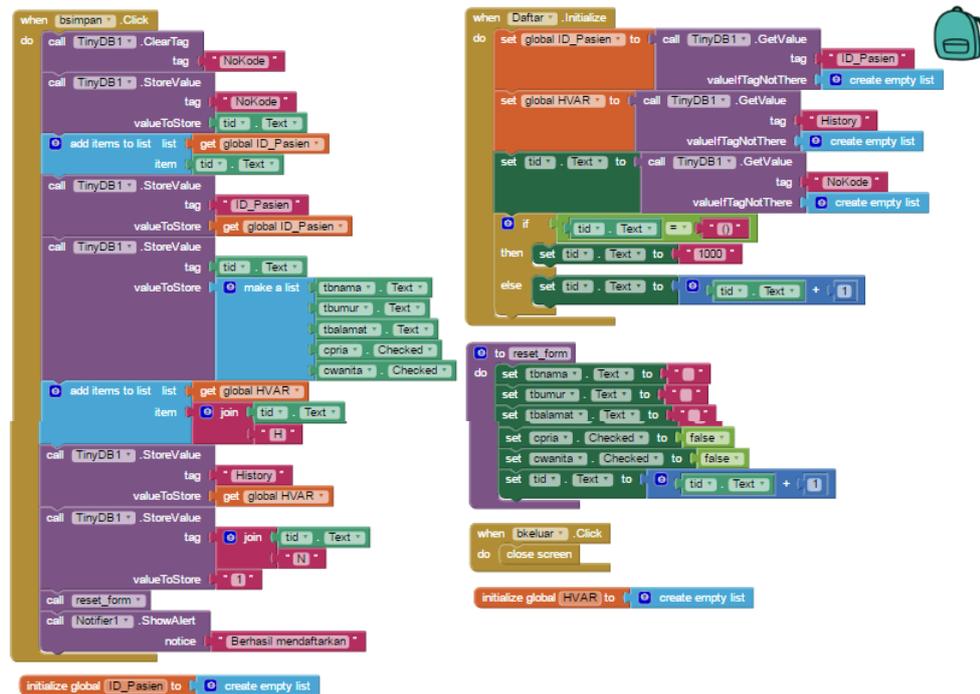
Menu keluar aplikasi mengarahkan pengguna keluar dari aplikasi.

c. Perancangan Basis data *TinyDB*

Perancangan basis data *TinyDB* mencakup 3 menu dari 5 menu utama, yaitu:

1. Menu daftar

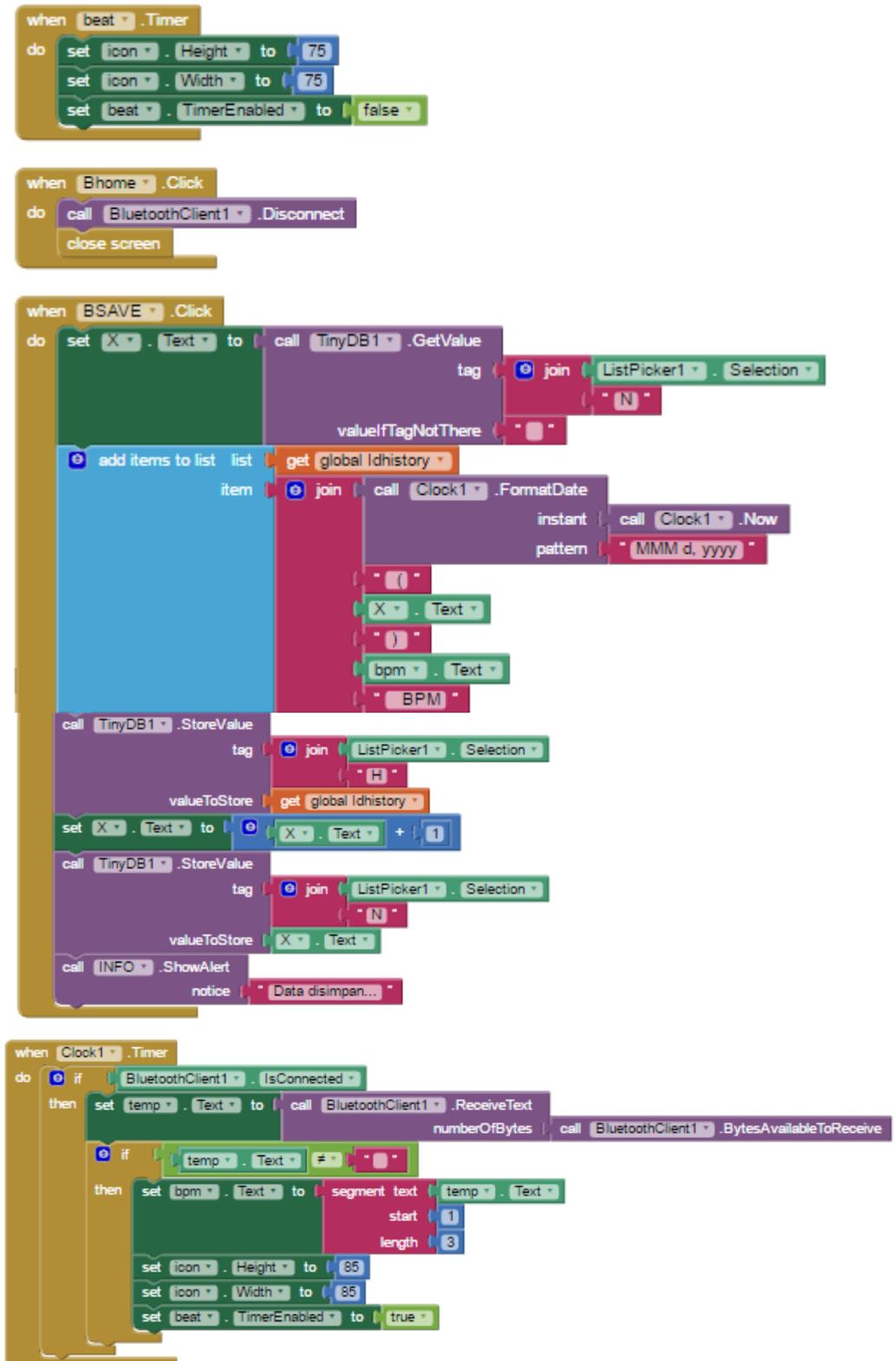
Pada menu daftar *TinyDB* digunakan untuk menyimpan data diri pengguna dan memberikan nomor id pada setiap data yang masuk untuk nantinya digunakan untuk membuka menu periksa dan histori. Adapun pemrograman *TinyDB* untuk menu daftar dapat dilihat pada gambar 3.14 di bawah ini.

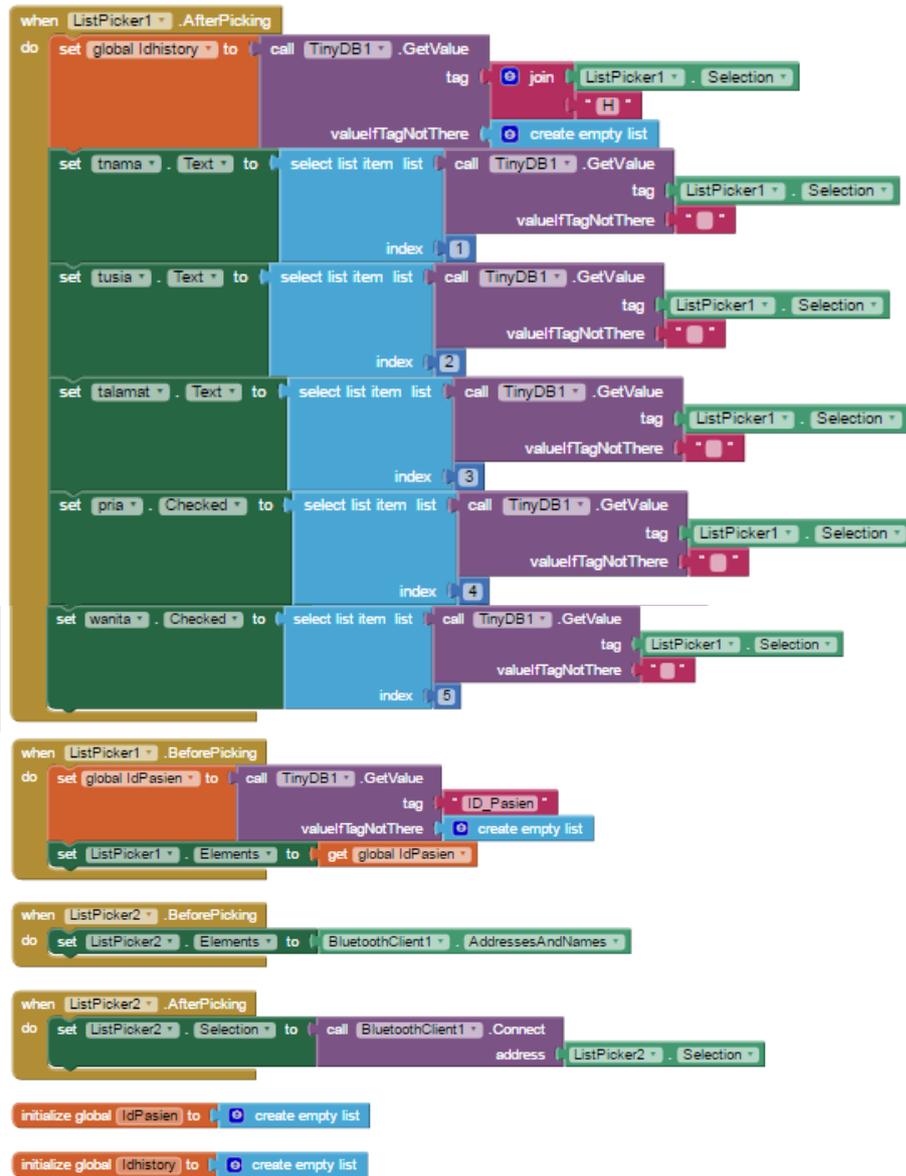


Gambar 3.14 program *TinyDB* untuk menu daftar

2. Menu periksa

Pada menu periksa, data di *TinyDB* yang digunakan pada menu daftar untuk menyimpan data diri pengguna dipanggil kembali, barulah dilakukan pemilihan komunikasi *bluetooth* untuk menjalankan pemeriksaan. Pemrograman yang digunakan untuk menjalankan *screen* periksa dapat dilihat pada gambar 3.15 di bawah ini.

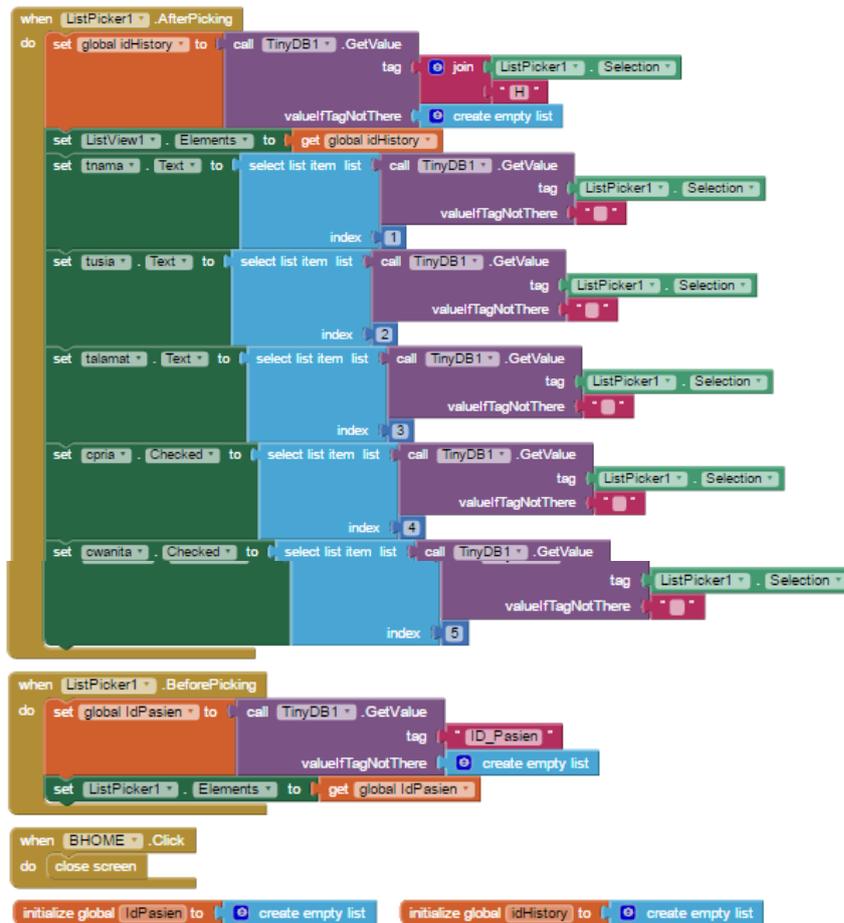




Gambar 3.15 pemrograman *TinyDB* untuk menu periksa

1. Menu histori

Pada menu histori, data pengguna yang disimpan di *TinyDB* pada menu daftar dipanggil kembali, begitu juga dengan data yang sudah disimpan pada menu periksa berupa hasil pemeriksaan. Pemrograman pada menu histori dapat dilihat pada gambar 3.16 di bawah ini.



Gambar 3.16Pemrograman pada menu histori

3.4 Teknik Pengumpulan Data

3.4.1 Rangkaian Catu Daya

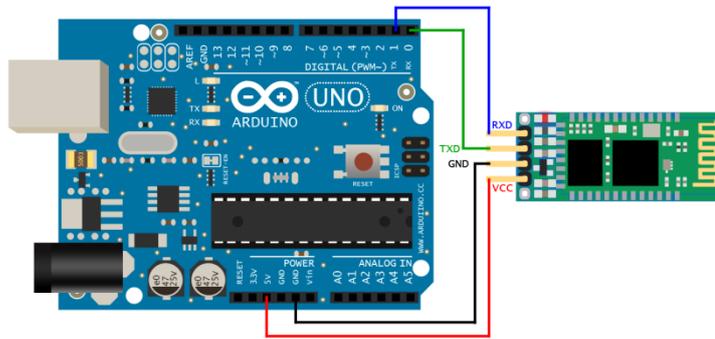
Sumber tegangan yang digunakan dalam penelitian ini adalah *powerbank* dengan kapasitas 1800 mAh dan keluaran 5 Volt DC seperti gambar 3.17 di bawah ini,



Gambar 3.17*powerbank*

3.4.2 Rangkaian Modul *Bluetooth* HC-05

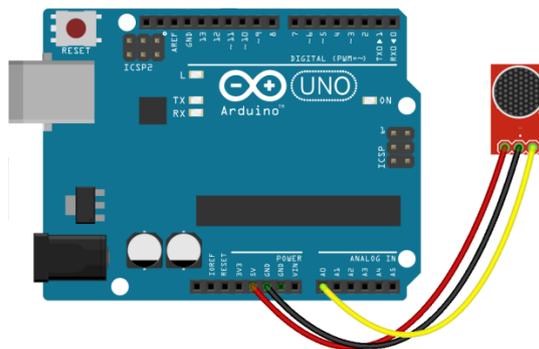
Rangkaian modul *bluetooth* yang digunakan adalah modul *bluetooth* HC-05 yang dipasangkan langsung dengan Arduino Uno R3 seperti gambar 3.18 di bawah ini.



Gambar 3.18 rangkaian *bluetooth* HC 05 dengan Aduino Uno R3

3.4.3 Rangkaian *Pulse sensor*

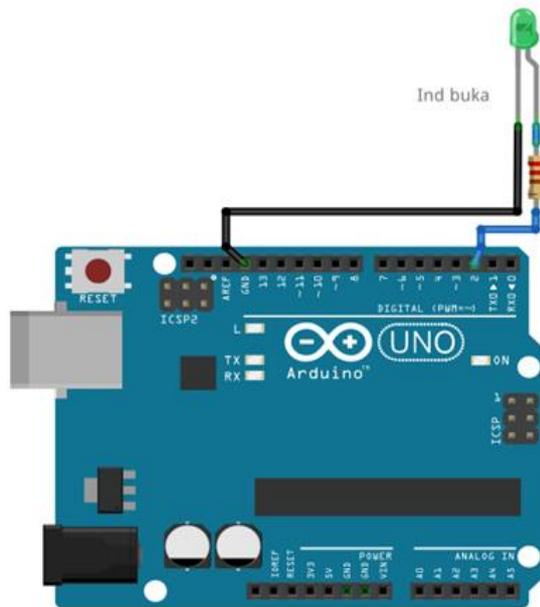
Rangkaian *pulse sensor* adalah sensor untuk mendeteksi detak jantung manusia yang dirancang khusus untuk Arduino dan perancangannya dapat dilihat pada gambar 3.19 di bawah ini.



Gambar 3.19 Rangkaian *pulse sensor* dengan Arduino

3.4.4 Rangkaian Indikator LED

Rangkaian Indikator LED yang digunakan adalah sebuah LED sebagai indikator saat *pulse sensor* mendeteksi adanya detak jantung manusia. Rangkaian indikator LED tersebut dapat dilihat pada gambar 3.20 di bawah ini.



Gambar 3.20 Rangkaian Indikator LED dengan Arduino

3.5 Teknik Analisis Data

Mengacu pada metode penelitian RnD dan Aspek yang dievaluasi pada instrumen maka teknik analisis data yang dilakukan pada penelitian ini yaitu dengan menguji prototipe yang sudah dibuat. Adapun pengujian yang dilakukan pada prototipe sebagai berikut :

3.5.1 Pengujian Catu Daya

Pada pengujian catu daya, sumber tegangan utama yaitu *powerbank* dengan kapasitas 1800 mAh dan keluaran 5 Volt DC, keluaran regulator 3.3 Volt dari pin 3.3 V Arduino dan keluaran tegangan dari pin 5V Arduino diukur sesuai kriterianya. Tabel 3.2 menunjukkan kriteria pengujian catu daya.

Tabel 3.2 menunjukkan kriteria pengujian catu daya

No	Sumber tegangan/ catu daya	Kriteria pengukuran	Hasil pengukuran
1	<i>Powerbank</i>	5 Volt DC	
2	Pin Vcc 5 Volt Arduino	5 Volt DC	

3.5.2 Pengujian Modul *Bluetooth* HC-05

Pengujian pada modul bluetooth HC-05 dilakukan dengan melihat apakah modul bluetooth HC-05 terkoneksi dengan baik dengan smartphone android. Kriteria pengujian pada modul Bluetooth HC-05 ditunjukkan pada tabel 3.3

Tabel 3.3 Kriteria pengujian pada modul *Bluetooth* HC-05

No	Bentuk Komunikasi	Kriteria Pengujian	Hasil Pengujian (Gambar)
1	<i>Bluetooth</i> HC-05	<i>Bluetooth</i> terkoneksi dengan <i>smartphone</i> android	

3.5.3 Pengujian *Pulse sensor*

Pengujian pada *pulse sensor*, dilakukan dengan mengamati gelombang oscilloscope. *Pulse sensor* dihubungkan dengan arduino, catu daya dan oscilloscope. Berikut Kriteria Pengujian *pulse sensor* pada tabel 3.4

tabel 3.4 Kriteria Pengujian *pulse sensor*

NO	Kondisi	Kriteria Pengecekan	Gambar
1	Tidak ditempel jari	Tidak ada gelombang di oscilloscope	
2	Ditempel jari	Ada gelombang di oscilloscope	

3.5.4 Pengujian Indikator LED

Pada pengujian Indikator LED, LED akan menyala sesuai kondisi darah pada area tubuh yang ditempelkan di *pulse sensor*. Berikut Kriteria Pengujian Indikator LED pada tabel 3.5

Tabel 3.5 Kriteria Pengujian Indikator LED

NO	Kondisi Darah	Kriteria Pengecekan	Hasil Pengujian (gambar)
1	Lebih Pekat	Led indikator akan menyala	
2	Kurang Pekat	Led indikator akan mati	

BAB IV

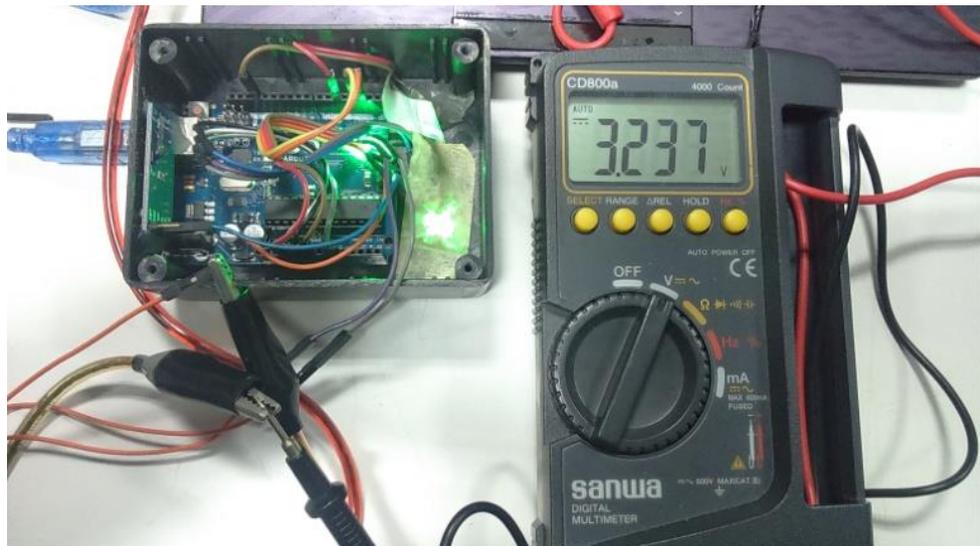
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Pengembangan Produk

Berdasarkan penyusunan blok diagram, flowchart beserta desain produk yang telah dirancang, maka berikut gambaran lokasi penelitian, karakteristik responden dan hasil implementasi dari desain produk yang telah dibuat.

4.1.1 Lokasi Penelitian

Penelitian dilaksanakan di Bengkel Mekanik Teknik Elektro Universitas Negeri Jakarta. Salah satu penelitian yang dilakukan adalah pengujian Arduino Uno R3 yang ditunjukkan pada gambar 4.1.



Gambar 4.1 Pengujian Arduino

4.1.2 Implementasi Desain Prototipe

4.1.2.1 Implementasi Desain Prototipe perangkat keras

Hasil Implementasi desain prototipe perangkat keras yang telah dibuat dapat ditunjukkan pada tabel 4.1 di bawah ini.

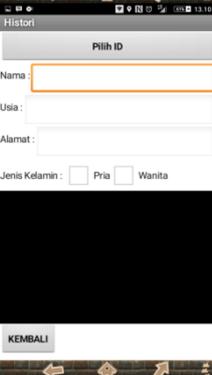
Tabel 4.1 Hasil Implementasi Desain Prototipe

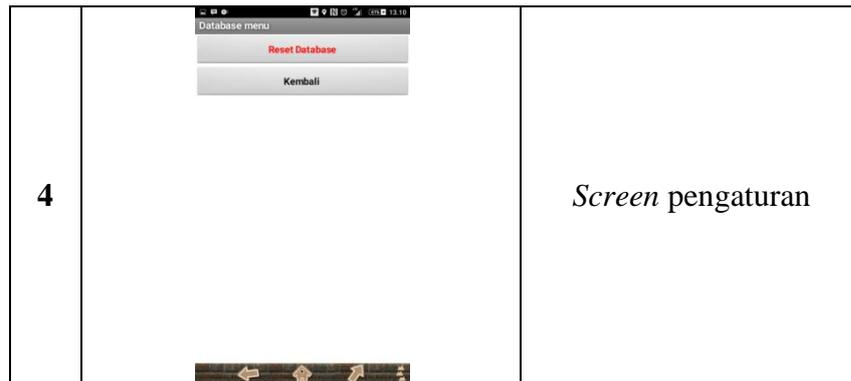
No	Hasil Implementasi	Keterangan
1		Tampak atas
2		Tampak Samping
3		Tampak Bawah

4.1.2.2 Implementasi prototipe perangkat lunak

Hasil implementasi prototipe perangkat lunak yaitu pembuatan aplikasi dengan menggunakan App Inventor dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil implementasi prototipe perangkat lunak

No	Hasil Implementasi	Keterangan
1		Screen home
2		Screen periksa
3		Screen histori



4.2 Kelayakan Produk

(Indra, 2010) Satu-satunya cara yang satu dapat menguji validitas teori ini adalah melalui percobaan. Apa yang menegaskan teori adalah pencocokan yang tepat antara diamati eksperimental dan hasil prediksi teoritis. Tingkat kesalahan, yang merupakan perbedaan antara nilai-nilai percobaan dapat diamati dan predikat akan membawa kehancuran bagi model teoritis. Bahkan perubahan kecil atau kesalahan, berarti bahwa ada sesuatu yang kita tidak mengerti dan perlu account untuk itu.

Cara menghitung presentase error adalah dengan mengurangi hasil prototipe dengan hasil pulse oximeter yang kemudian dibagi dengan hasil dari pulse oximeter lalu dikalikan 100.

Misalnya, hasil prototipe adalah 81 dan hasil pulse oximeter adalah 82, setelah dikurangi hasilnya adalah 1, kemudian dibagi 82 yang merupakan hasil pulse oximeter yaitu 0,012. Setelah dikalikan 100 maka hasilnya adalah 1,2. Maka presentase errornya adalah 1,2.

Sebuah prototipe layak untuk di uji coba atau tidak dengan melihat perbandingan antara prototipe dengan PULSE OXIMETER. Pengujian dilakukan

dengan cara menempelkan ibu jari sebelah kanan pada prototipe dan ibu jari sebelah kiri pada Pulse Oximeter. Hasil dari pengujian prototipe dapat dilihat pada tabel 4.3.

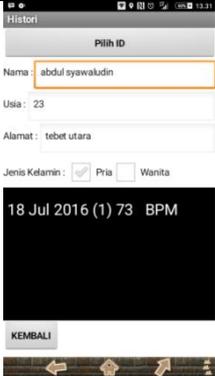
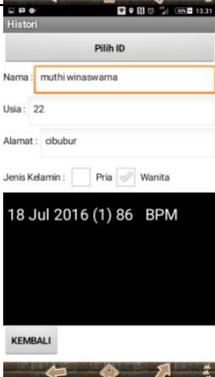
Tabel 4.3 hasil pengujian perbandingan

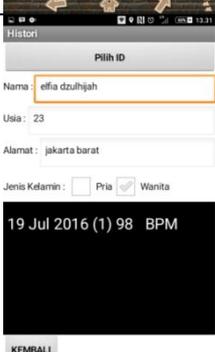
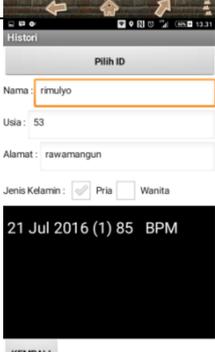
No. ID	Nama	Usia	JK	PO (Bpm)	Prototipe (Bpm)	Error (%)
1000	Abdul Syawaludin	23	Pria	74	71	4,05
1001	Hani Nurlailawati	23	Wanita	95	95	0
1002	Muthi winaswarna	22	Wanita	87	87	0
1003	Muhammad Ilman	23	Pria	92	93	1,08
1004	Naomi Christin	23	Wanita	81	82	1,23
1005	Elfia Dzulhijjah	23	Wanita	78	78	0
1006	Rimulyo	53	Pria	87	87	0

Berdasarkan hasil pengujian dengan membandingkan hasil antara PULSE OXIMETER dengan prototipe pada tabel 4.3 dapat dilihat bahwa presentase error antara Pulse Oxymeter dan prototipe sistem monitoring detak jantung tidak lebih dari 5%, sehingga prototipe sistem monitoring detak jantung layak untuk digunakan.

Kelayakan produk juga dilihat dari keberhasilan aplikasi yang dibuat dalam menyimpan data hasil pemeriksaan. Tampilan hasil pemeriksaan yang ada di menu histori dapat dilihat pada tabel 4.4.

Tabel 4.4 Tampilan menu histori

No. ID	Histori Pemeriksaan
1000	
1001	
1002	
1003	

1004	
1005	
1006	

4.3 Efektifitas Produk

Efektifitas produk dapat dilihat berdasarkan hasil implementasi dan pengujian dari perangkat keras dan perangkat lunak yang telah dibuat. Adapun bentuk hasil pengujian dapat dijelaskan sebagai berikut :

4.3.1 Hasil Pengujian Catu Daya

Pengujian catu daya yaitu dengan mengukur powerbank berkapasitas 1800 mAh dan keluaran 5 Volt DC. Hasil pengujian dari rangkaian catu daya dapat dilihat pada tabel 4.5.

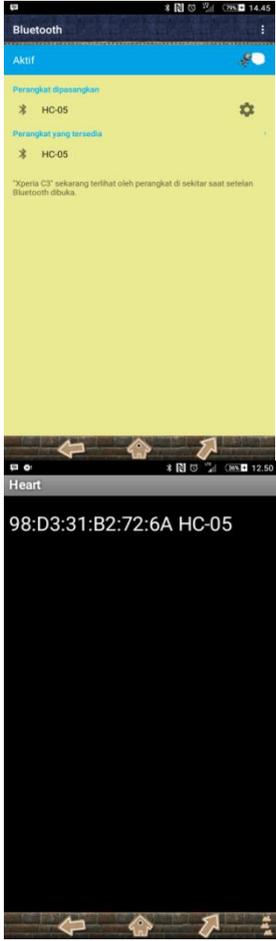
Tabel 4.5 Hasil Pengujian Catu Daya

No	Sumber tegangan/ catu daya	Kriteria pengukuran	Hasil pengukuran	Gambar
1	Powerbank	5 Volt DC	4,51 Volt	
3	Pin Vcc 5 Volt Arduino	5 Volt DC	4,51 Volt	
4	Pin 3.3V Arduino	3,3 Volt DC	3,23 Volt	

4.3.2 Hasil Pengujian Modul *Bluetooth* HC 05

Pengujian pada modul *bluetooth* HC-05 dilakukan dengan melihat apakah modul *bluetooth* HC-05 terkoneksi dengan baik dengan *smartphone* android. Kriteria pengujian pada modul *Bluetooth* HC-05 ditunjukkan pada tabel 4.6.

Tabel 4.6 Hasil pengujian modul *bluetooth* HC 05

No	Bentuk Komunikasi	Kriteria Pengujian	Hasil Pengujian (Gambar)
1	<i>Bluetooth</i> HC-05	<i>Bluetooth</i> terkoneksi dengan <i>smartphone</i> android	

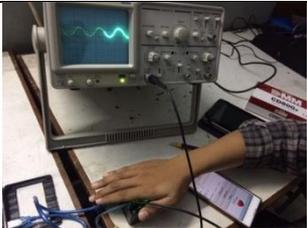
Hasil uji coba menunjukkan Bluetooth HC 05 dapat terkoneksi dengan baik sesuai dengan kriteria yang diinginkan.

4.3.3 Hasil Pengujian Pulse sensor

Uji coba sensor dilakukan dengan mengamati dengan menggunakan oscilloscope untuk melihat adanya gelombang yang menunjukkan bekerjanya sebuah *pulse sensor*. Saat pulse sensor mendapat input dari jari tangan manusia, maka akan muncul gambar gelombang pada oscilloscope, begitu juga sebaliknya.

Pengaturan pada oscilloscope dalam pengujian ini yaitu 2 time/div dan 5 volt/div . Hasil pengujian *pulse sensor* dapat dilihat pada tabel 4.7.

Tabel 4.7 hasil pengujian Pulse Sensor

NO	Kondisi Darah	Kriteria Pengecekan	Indikator	Gambar
1	Lebih Pekat	tidak ditempelkan jari	Tidak ada gelombang	
2	Kurang Pekat	Ditempelkan jari	Ada gelombang	

Hasil uji coba menunjukkan pulse sensor dalam kondisi sesuai dengan kriteria yang diinginkan.

4.3.4 Hasil Pengujian Indikator LED

Pada pengujian Indikator LED, LED akan menyala sesuai kondisi darah pada area tubuh yang ditempelkan di pulse sensor. Hasil pengujian indikator LED dapat dilihat pada tabel 4.8.

Tabel 4.8 hasil pengujian Indikator Led

NO	Kondisi Darah	Kriteria Pengecekan	Hasil Pengujian(gambar)
1	Lebih Pekat	Led indikator akan menyala	
2	Kurang Pekat	Led indikator akan mati	

4.4 Pembahasan

Pada prototipe alat pendeteksi dan penghitung detak jantung yang sudah diuji dapat dilihat bahwa pulse sensor dapat bekerja sesuai dengan kriteria yang sudah diberikan, hal ini sudah dapat dipastikan pada pengujian sebelumnya yang hasilnya dapat dilihat pada tabel 4.7.

Pada pengujian modul bluetooth HC 05 dapat dilihat pada tabel 4.6 bahwa modul bluetooth HC 05 sudah dapat terkoneksi dengan baik pada smartphone android. Hal ini memungkinkan pengiriman data dari pulse sensor dan Arduino ke smartphone android.

Pada pengujian indikator LED, LED dapat berkedip sesuai kriteria yang sudah ditentukan dengan hasil deteksi yang sudah dibaca oleh pulse sensor.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan perancangan, implementasi, pengujian, dan analisis rancang bangun prototipe sistem monitoring detak jantung manusia dengan pulse sensor dan penyimpanan data hasil pemeriksaan menggunakan TinyDB berbasis Arduino UNO R3, dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan program yang sudah dibuat, batas maksimal Bpm yg muncul adalah 240 Bpm tanpa batas bawah.
2. Pengujian dilakukan dengan cara menempelkan ibu jari sebelah kanan pada prototipe dan ibujari sebelah kiri pada Pulse Oximeter.
3. Berdasarkan hasil pengujian dengan membandingkan hasil antara PULSE OXIMETER dengan prototipe pada tabel 4.3 dapat dilihat bahwa presentase error antara Pulse Oxymeter dan prototipe sistem monitoring detak jantung tidak lebih dari 5%, sehingga prototipe sistem monitoring detak jantung layak untuk digunakan.

5.2 Saran

Berdasarkan penelitian yang telah dilaksanakan dan kesimpulan yang didapatkan, maka saran untuk penyempurnaan dan pengembangan lebih lanjut dari penelitian ini antara lain:

1. Pembuatan aplikasi android menggunakan software selain *App Inventor* agar dapat melakukan penghapusan data satuan sehingga tidak perlu menghapus seluruh data.
2. Melakukan perbandingan dengan alat yg digunakan di rumah sakit seperti EKG atau elektrokardiograf agar mendapatkan hasil perbandingan yang lebih akurat.
3. Mikrokontroller arduino yang digunakan sebaiknya adalah arduino nano mengingat pin yang ada di arduino nano sama dengan arduino uno. Namun, arduino nano memiliki ukuran lebih kecil dari arduino nano sehingga bentuk alat yang dibuat akan lebih efisien.

DAFTAR PUSTAKA

- Anonimous. 2013. *Cara Menghitung Persentase*.
<http://rumushitung.com/2012/12/10/cara-menghitung-persentase>. Diakses 01Oktober 2016
- Arduino, 2013, Arduino Uno R3. [http://www. http://arduino.cc/en/Main/ArduinoBoardUno/](http://www.arduino.cc/en/Main/ArduinoBoardUno/).
- Ari BeniSantoso, Martinus, danSugiyanto, “Pembuatan Otomasi Pengaturan Kereta Api, Pengereman, dan Palang Pintu pada Rel Kereta Api Mainan Berbasis Mikrokontroler”.*Jurnal FEMA*. Vol.1 No.1, Januari 2013
- Artanto, Dian. 2007. *Interaksi Arduino dan Labview* . Jakarta : PT Elex Media Komputindo.
- Bluetooth HC. 2013, Bluetooth HC-05 Serial Module .
<http://hcmodule.com/2013/06/19/using-the-hc-06-bluetooth-module/>
- Conor. 1974. *Monitoring Pekerjaan*. Yogyakarta.
- Med. Suhantoro (2012). Pentingnya Memperhatikan Detak Jantung Sebelum Olahraga.
<http://health.detik.com/read/2012/01/10/152522/1811778/763/pentingnya-memperhatikan-detak-jantung-sebelum-olahraga>. Diakses 09 Juni 2016
- Edward R. Laskowski, M.D. (2015). What's a normal resting heart rate?. *Healthy Lifestyle Fitness*.
<http://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/heart-rate/faq-20057979>. Diakses 09 Juni 2016
- Gustia, Irna. 2011. Waspadai Kram Jantung saat Berolahraga Keras.
<http://health.detik.com/read/2011/02/05/111505/1560639/763/waspadai%20kram%20jantung>. Diakses 23 Agustus 2016.
- Hilmi, Zul. (2014). *Membuat Aplikasi Android Lebih Mudah dengan Google App Inventor*.
<https://www.codepolitan.com/membuat-aplikasi-android-lebih-mudah-dengan-google-app-inventor/>. Diakses 20 Juli 2016.
- Jerome, William Travers.1961. *Executive Control: The Catalyst*. John Wiley & Sons, Inc.
- Kadir, Abdul. 1999. *Konsep dan Tuntunan Praktis BasisData*, Penerbit Andi, Yogyakarta
- Kroenke, M.D. 1998. *Database Processing*, Prentice Hall International, Inc.,
- Nuryati, Venti. 2010. *Rancang Bangun Alat Pendeteksi dan Penghitung Detak Jantung dengan Asas Doppler*. Universitas Indonesia. Depok.

Herman B. Rahmatika Phd, AIF.2009. *Buku Ajar Fisiologi Jantung*. Penerbit EGC. Jakarta

Pulse Sensor Amped Getting Started Guide <http://pulse-sensor.googlecode.com/files/PulseSensorAmpedGettingStartedGuide.pdf>. &https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino.git diakses 12 Mei 2016.

Sugiyono. (2011). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.

Syahwil Muhammad 2013. *Panduan Mudah Simulasi dan Praktek Mikrokontroler Arduino*. Penerbit Andi. Yogyakarta.

LAMPIRAN

LAMPIRAN 1

Instrumen Penelitian

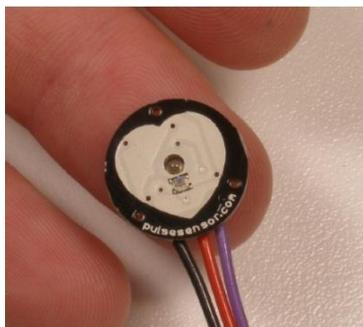
1. Multimeter digital sanwa CD800a



2. Puse Oximeter



3. Pulse Sensor



4. Led indikator



LAMPIRAN 2

List program Arduino & interrupt

1. List program Arduino

```

#include <SoftwareSerial.h>;

SoftwareSerial bluetooth(2, 3); // RX, TX

char inChar;

String inString = "";

int pulsePin = 0;

int blinkPin = 13;

int fadePin = 5;

int fadeRate = 0;

volatile int BPM;

volatile int Signal;

volatile int IBI = 600;

volatile boolean Pulse = false;

volatile boolean QS = false;

void setup(){

  pinMode(blinkPin,OUTPUT); // pin that will blink to your
  heartbeat!

  pinMode(fadePin,OUTPUT); // pin that will fade to your
  heartbeat!

  Serial.begin(9600);

  while (!Serial) {

  }

  bluetooth.begin(9600);

  interruptSetup(); // sets up to read Pulse
  Sensor signal every 2mS

}

void loop(){

  //sendDataToProcessing('S', Signal);

  if (QS == true){

    fadeRate = 255;

```

```
Serial.print("BPM=");  
Serial.println(BPM);  
//bluetooth.print("BPM=");  
bluetooth.println(BPM);  
QS = false;  
}  
ledFadeToBeat();  
delay(20);  
}  
void ledFadeToBeat(){  
    fadeRate -= 15;  
    fadeRate = constrain(fadeRate,0,255);  
    analogWrite(fadePin,fadeRate);  
}
```

2. List program interrupt

```

volatile int rate[10];    // array to hold last ten IBI values

volatile unsigned long sampleCounter = 0;    // used to
determine pulse timing

volatile unsigned long lastBeatTime = 0;    // used to find IBI

volatile int P =512; // used to find peak in pulse wave, seeded

volatile int T = 512;    // used to find
trough in pulse wave, seeded

volatile int thresh = 512;    // used to find instant moment
of heart beat, seeded

volatile int amp = 100;    // used to hold amplitude of pulse
waveform, seeded

volatile boolean firstBeat = true;    // used to seed rate
array so we startup with reasonable BPM

volatile boolean secondBeat = false;    // used to seed rate
array so we startup with reasonable BPM

void interruptSetup(){    // Initializes Timer2 to throw an
interrupt every 2mS.

    TCCR2A = 0x02;    // DISABLE PWM ON DIGITAL PINS 3 AND 11,
AND GO INTO CTC MODE

    TCCR2B = 0x06;    // DON'T FORCE COMPARE, 256 PRESCALER

    OCR2A = 0X7C;    // SET THE TOP OF THE COUNT TO 124 FOR
500Hz SAMPLE RATE

    TIMSK2 = 0x02;    // ENABLE INTERRUPT ON MATCH BETWEEN
TIMER2 AND OCR2A

    sei();    // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}

// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.

// Timer 2 makes sure that we take a reading every 2
milliseconds

ISR(TIMER2_COMPA_vect){    // triggered when Timer2 counts to
124

    cli();    // disable interrupts while we do this

    Signal = analogRead(pulsePin); // read the Pulse Sensor

    sampleCounter += 2;    // keep track of the time in
mS with this variable

```

```

    int N = sampleCounter - lastBeatTime;          // monitor the
time since the last beat to avoid noise

    // find the peak and trough of the pulse wave

    if(Signal < thresh && N > (IBI/5)*3){          // avoid
dichrotic noise by waiting 3/5 of last IBI

        if (Signal < T){                          // T is the trough

            T = Signal;    // keep track of lowest point in pulse wave

        }

    }

    if(Signal > thresh && Signal > P){              // thresh
condition helps avoid noise

        P = Signal;                               // P is the peak

    }          // keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT

// signal surges up in value every time there is a pulse

if (N > 250){                                     // avoid high frequency noise

    if ( (Signal > thresh) && (Pulse == false) && (N >
(IBI/5)*3) ){

        Pulse = true;    // set the Pulse flag when we think there
is a pulse

        digitalWrite(blinkPin,HIGH);             // turn on pin 13 LED

        IBI = sampleCounter - lastBeatTime;      // measure
time between beats in mS

        lastBeatTime = sampleCounter;            // keep track
of time for next pulse

        if(secondBeat){                          // if this is the
second beat, if secondBeat == TRUE

            secondBeat = false;                  // clear secondBeat flag

            for(int i=0; i<=9; i++){              // seed the
running total to get a realistic BPM at startup

                rate[i] = IBI;

            }

        }

    }

    if(firstBeat){                               // if it's the first time we found
a beat, if firstBeat == TRUE

        firstBeat = false;                       // clear firstBeat flag

```

```

        secondBeat = true;          // set the second beat flag
        sei();                      // enable interrupts again
        return;                    // IBI value is unreliable so discard it
    }

    // keep a running total of the last 10 IBI values
    word runningTotal = 0; // clear the runningTotal variable
    for(int i=0; i<=8; i++){ // shift data in the rate array
        rate[i] = rate[i+1]; // and drop the oldest IBI value
        runningTotal += rate[i]; // add up the 9
oldest IBI values
    }

    rate[9] = IBI; // add the latest IBI to the rate array
    runningTotal += rate[9]; // add the latest IBI to
runningTotal

    runningTotal /= 10; // average the last 10 IBI values

    BPM = 60000/runningTotal; // how many beats
can fit into a minute? that's BPM!

    QS = true; // set Quantified
Self flag

// QS FLAG IS NOT CLEARED INSIDE THIS ISR
    }

}

    if (Signal < thresh && Pulse == true){ // when the values
are going down, the beat is over

        digitalWrite(blinkPin,LOW); // turn off pin 13 LED
        Pulse = false;// reset the Pulse flag so we can do it again
        amp = P - T; // get amplitude of the pulse wave
        thresh = amp/2 + T; // set thresh at 50% of the amplitude
        P = thresh; // reset these for next time
        T = thresh;
    }

    if (N > 2500){ // if 2.5 seconds go by without a beat
        thresh = 512; // set thresh default
    }

```

```
    P = 512;                // set P default
    T = 512;                // set T default

    lastBeatTime = sampleCounter; // bring the lastBeatTime
up to date

    firstBeat = true;      // set these to avoid noise
    secondBeat = false;    // when we get the heartbeat back
}

sei();                    // enable interrupts when youre done!
} // end isr
```

LAMPIRAN 3

Program App *inventor*

1. Home

```

when bmdaftar .Click
do open another screen screenName "Daftar"

when bperiksa .Click
do open another screen screenName "Periksa"

when bhistor .Click
do open another screen screenName "Histori"

when batur .Click
do open another screen screenName "Db"

when btutup .Click
do close application
  
```

2. Menu daftar

```

when bsimpan .Click
do
  call TinyDB1 .ClearTag
  tag "NoKode"
  call TinyDB1 .StoreValue
  tag "NoKode"
  valueToStore tid .Text
  add items to list list
  item tid .Text
  call TinyDB1 .StoreValue
  tag "ID_Pasien"
  valueToStore get global ID_Pasien
  call TinyDB1 .StoreValue
  tag tid .Text
  valueToStore make a list
  tnama .Text
  tbumur .Text
  tbalamat .Text
  cpria .Checked
  owanita .Checked
  add items to list list
  item join tid .Text
  call TinyDB1 .StoreValue
  tag "History"
  valueToStore get global HVAR
  call TinyDB1 .StoreValue
  tag join tid .Text
  valueToStore "1"
  call reset_form
  call Notifier1 .ShowAlert
  notice "Berhasil mendaftarkan"

initialize global ID_Pasien to create empty list
  
```

```

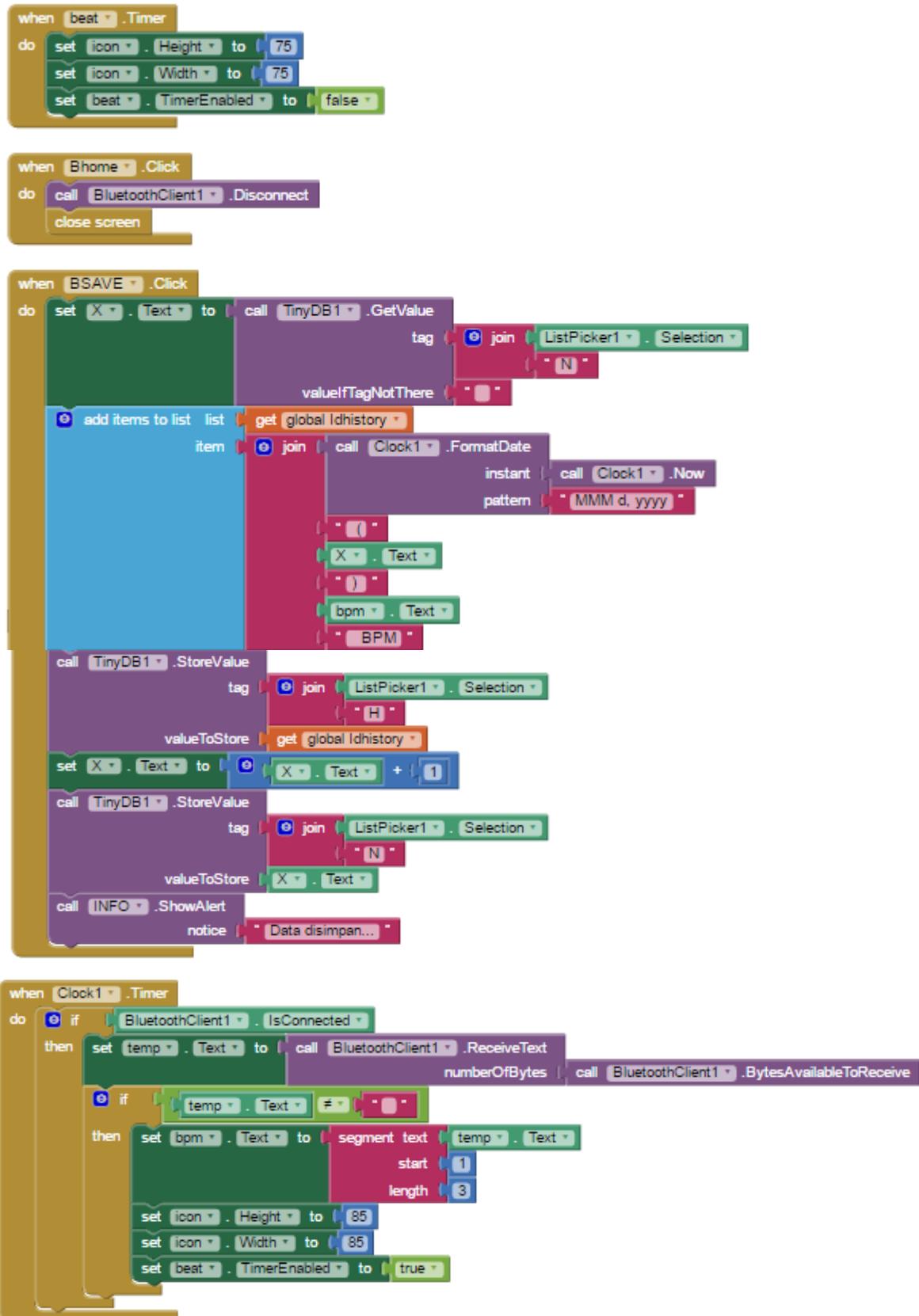
when Daftar .Initialize
do
  set global ID_Pasien to call TinyDB1 .GetValue
  tag "ID_Pasien"
  valueIfTagNotThere create empty list
  set global HVAR to call TinyDB1 .GetValue
  tag "History"
  valueIfTagNotThere create empty list
  set tid .Text to call TinyDB1 .GetValue
  tag "NoKode"
  valueIfTagNotThere create empty list
  if tid .Text == ""
  then set tid .Text to "1000"
  else set tid .Text to tid .Text + 1

to reset_form
do
  set tnama .Text to ""
  set tbumur .Text to ""
  set tbalamat .Text to ""
  set cpria .Checked to false
  set owanita .Checked to false
  set tid .Text to tid .Text + 1

when bkeluar .Click
do close screen

initialize global HVAR to create empty list
  
```

3. Menu periksa



```

when ListPicker1 .AfterPicking
do
  set global Idhistory to call TinyDB1 .GetValue
  tag join ListPicker1 . Selection
  valueIfTagNotThere "H"
  create empty list
  set tnama . Text to select list item list
  call TinyDB1 .GetValue
  tag ListPicker1 . Selection
  valueIfTagNotThere " "
  index 1
  set tusia . Text to select list item list
  call TinyDB1 .GetValue
  tag ListPicker1 . Selection
  valueIfTagNotThere " "
  index 2
  set talamat . Text to select list item list
  call TinyDB1 .GetValue
  tag ListPicker1 . Selection
  valueIfTagNotThere " "
  index 3
  set pria . Checked to select list item list
  call TinyDB1 .GetValue
  tag ListPicker1 . Selection
  valueIfTagNotThere " "
  index 4
  set wanita . Checked to select list item list
  call TinyDB1 .GetValue
  tag ListPicker1 . Selection
  valueIfTagNotThere " "
  index 5

when ListPicker1 .BeforePicking
do
  set global IdPasien to call TinyDB1 .GetValue
  tag "ID_Pasien"
  valueIfTagNotThere create empty list
  set ListPicker1 . Elements to get global IdPasien

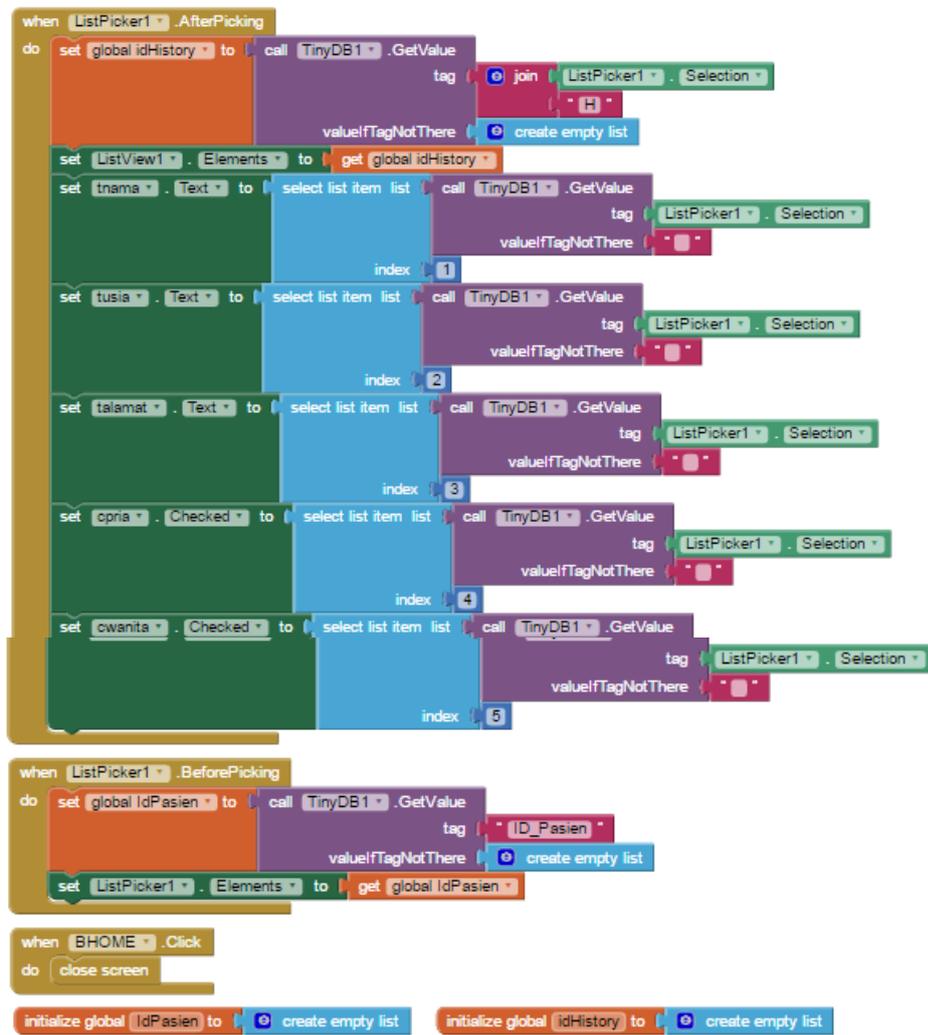
when ListPicker2 .BeforePicking
do
  set ListPicker2 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker2 .AfterPicking
do
  set ListPicker2 . Selection to call BluetoothClient1 .Connect
  address ListPicker2 . Selection

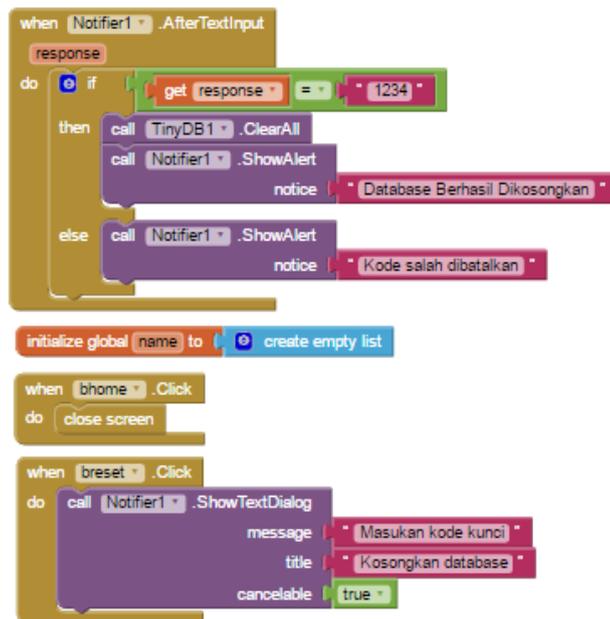
initialize global IdPasien to create empty list
initialize global Idhistory to create empty list

```

4. Menu histori



5. Menu pengaturan



Riwayat Hidup



Vicky Anggitasari, lahir di Pati, 04 Juli 1993. Penulis bertempat tinggal di jl. Danau Mahalona 23 blok FD 25 no. 21 Vila Tangerang Regensi 2, Pasar Kemis, Gelam Jaya, Tangerang. Mahasiswi yang hobi memasak dan berwirausaha ini pernah menempuh pendidikan dari tingkat taman kanak-kanak di TK Wijaya Kusuma, Kotabumi, Tangerang. Kemudian melanjutkan pendidikan dasar di SDN Gebang Raya I, Tangerang. Setelah itu, penulis melanjutkan pendidikan menengah SMP dan SMA di Pondok Pesantren La Tansa, Parakansantri, Lebak, Banten. penulis melanjutkan pendidikan ke jenjang lebih tinggi di Universitas Negeri Jakarta Jurusan Teknik Elektro, Program Studi Pendidikan Teknik Elektronika S1. Penulis dapat dihubungi melalui alamat email di vicky.enjhie@gmail.com atau nomor telepon (021) 5923114.