

**SISTEM *MONITORING* KEBUGARAN KARDIORESPIRASI
DENGAN *PULSE SENSOR* DAN *LOAD SENSOR*
MENGUNAKAN *NON-EXERCISE PREDICTION METHOD***

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rifqi Anshari
NIM: 135150300111012



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

PENGESAHAN

SISTEM MONITORING KEBUGARAN KARDIORESPIRASI DENGAN PULSE SENSOR
DAN LOAD SENSOR MENGGUNAKAN NON-EXERCISE PREDICTION METHOD

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

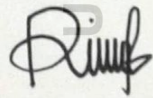
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rifqi Anshari
NIM: 135150300111012

Skripsi ini telah diuji dan dinyatakan lulus pada
12 Januari 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc
NIK: 201607 891009 1 001

Dosen Pembimbing II



Dahniel Syaury, S.T., M.T., M.Sc
NIK: 201607 870423 1 002

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D A
NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Januari 2018



Rifqi Anshari

NIM: 135150300111012

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Sistem *Monitoring* Kebugaran Kardiorespirasi Dengan *Pulse Sensor* Dan *Load Sensor* Menggunakan *Non-Exercise Prediction Method*” ini dapat terselesaikan.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh sebab itu, penulis menyampaikan rasa hormat dan terima kasih kepada:

1. Allah yang Maha Esa Karena atas kehendak dan nikmat-NYA laporan skripsi ini telah selesai dengan baik
2. Abdul Azis dan Helda Iriana selaku orang tua dari penulis beserta keluarga yang selalu mendukung dan memberikan semangat tanpa henti dalam pengerjaan laporan skripsi ini.
3. Rizal Maulana, S.T., M.T., M.Sc. selaku dosen pembimbing pertama yang telah membimbing dan memberikan solusi terhadap pengerjaan skripsi kepada penulis.
4. Dahnia Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing kedua yang selalu teliti dan memberikan arahan terkait penulisan laporan skripsi yang baik kepada penulis.
5. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika
6. Teman - teman penulis di program studi teknik komputer yang selalu memberikan dorongan dan semangat untuk segera menyelesaikan skripsi ini.
7. Teman – teman penulis di UKM Perisai Diri yang selalu memberikan singgungan membangun untuk terpacu secepat mungkin menyelesaikan skripsi.
8. Warung 999 yang selalu menjadi tempat nyaman untuk diskusi bersama dengan teman – teman seperjuangan.
9. Teman – teman lain yang selalu mendokan penulis yang tidak dapat penulis ungkapkan namanya satu persatu. Terimakasih atas semua doa yang sudah diberikan.

Penulis mengucapkan banyak terima kasih dan penulis sadar akan beberapa kekurangan pada laporan ini. Semoga laporan ini dapat memberikan manfaat dan referensi untuk melakukan penelitian dalam penyempurnaan kedepan.

Malang, 18 Januari 2018

Penulis

Rifqianshari07@gmail.com



ABSTRAK

Kebugaran kardiorespirasi berhubungan dengan kemampuan fungsi dari jantung, pembuluh darah, paru – paru, dan otot untuk menyuplai oksigen saat melakukan aktivitas fisik. Terdapat bukti bahwa memiliki kebugaran kardiorespirasi yang baik dapat menurunkan penyakit kardiovaskular seperti serangan jantung, nyeri dada dan stroke. Oleh karena itu penting untuk memiliki kebugaran kardiorespirasi yang baik. Kebugaran kardiorespirasi dapat di ukur melalui seberapa besar maksimum oksigen yang bisa diambil oleh tubuh (VO_{2max}). Terdapat metode untuk memprediksi VO_{2max} tanpa melibatkan kegiatan fisik dengan menggunakan denyut nadi istirahat, berat badan dan umur yaitu *non-exercise prediction method*. Pengukuran denyut nadi dan berat badan dalam perhitung VO_{2max} menggunakan 2 sensor yaitu *pulse sensor* dan *load sensor*. Untuk memudahkan dalam perhitungan sekaligus menampilkan hasil yang menarik maka digunakan perangkat berbasis OS android sebagai perantara dengan memanfaatkan modul bluetooth HC-05 sebagai media pertukaran data. Aplikasi yang dibangun pada perangkat berbasis OS android juga dilengkapi saran untuk meningkatkan VO_{2max} yang baik sehingga dapat menjadi pemandu bagi pengguna. Pengujian terhadap sensor dalam pengukuran VO_{2max} mendapatkan hasil *error* 8,12 % untuk *pulse sensor* dan hasil *error* 0,92% untuk *load sensor*. Jarak efektif maksimal yang dapat dilakukan modul *bluetooth* HC-05 yaitu 10 meter. Metode yang dilakukan dapat dengan mudah mengukur VO_{2max} seseorang tanpa harus melakukan kegiatan fisik sehingga memudahkan dalam pengawasan kebugaran kardiorespirasi tubuh.

Kata kunci: Kebugaran kardiorespirasi, VO_{2max} , *non-exercise prediction method*, *pulse sensor*, *load sensor*, modul *bluetooth* HC-05

ABSTRACT

Cardiorespiratory fitness connected with the ability function of heart, blood vessel, lungs and muscle to supply oxygen when do physical activity. There is a proof that own good cardiorespiratory fitness can reduce cardiovascular disease like heart attack and stroke. Therefore it's important to have good cardiorespiratory fitness. Cardiorespiratory fitness can be measured by level maximum oxygen body can take (VO_{2max}). There is a method for prediction of VO_{2max} without doing physical activity using heart rate rest, body weight and age (non-exercise prediction method). Measuring Heart rate and body weight for calculation VO_{2max} use 2 sensors, pulse sensor and load sensor. In order to facilitate VO_{2max} calculation and displaying attractive result, then used android OS-based devices as an intermediary by utilizing the HC-05 Bluetooth module as a medium of data exchange. Applications built on android OS based devices are also equipped with suggestions for improving good VO_{2max} so that it can be a guide for users. Tests on sensors in VO_{2max} measurements resulted in an error of 8.12% for pulse sensors and 0.92% error results for load sensor. The maximum effective range that can be done by the Bluetooth module HC-05 is 10 meters. The method performed can easily measure VO_{2max} without having to do physical activity so as to facilitate in the supervision of cardiorespiratory fitness of the body.

Keywords: *Cardiorespiratory fitness, VO_{2max} , non-exercise prediction method, pulse sensor, load sensor, bluetooth module HC-05*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika Pembahasan/Laporan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Dasar Teori	6
2.2.1 Kebugaran kardiorespirasi	6
2.2.2 Denyut Nadi	6
2.2.3 VO_{2max} dan <i>Non-Excercise Prediction Method</i>	7
2.2.4 Indeks Massa Tubuh.....	10
2.2.5 Arduino Nano	10
2.2.6 Modul <i>Bluetooth</i> HC-05	12
2.2.7 <i>Pulse Sensor</i> (SEN-11574)	13
2.2.8 <i>Load sensor</i>	14
2.2.9 HX711	16
2.2.10 OS android.....	17
BAB 3 METODOLOGI	19
3.1 Studi Literatur	19
3.2 Analisis Kebutuhan.....	20



3.2.1	Kebutuhan Perangkat Keras.....	21
3.2.2	Kebutuhan Perangkat Lunak.....	21
3.2.3	Kebutuhan Fungsional.....	21
3.3	Perancangan Sistem.....	21
3.3.1	Perancangan perangkat keras.....	21
3.3.2	Perancangan perangkat lunak.....	23
3.4	Implementasi Sistem.....	24
3.5	Pengujian Sistem dan Analisis Hasil.....	24
BAB 4	REKAYASA KEBUTUHAN.....	25
4.1	Gambaran Umum Sistem.....	25
4.2	Kebutuhan Sistem.....	25
4.2.1	Kebutuhan Perangkat lunak.....	26
4.2.2	Kebutuhan Perangkat Keras.....	26
4.2.3	Kebutuhan Fungsional.....	27
4.3	Batasan Desain Sistem.....	28
BAB 5	PERANCANGAN DAN IMPLEMENTASI.....	29
5.1	Perancangan Sistem.....	29
5.1.1	Perancangan Perangkat Keras.....	30
5.1.2	Perancangan Perangkat Lunak.....	31
5.2	Implementasi Sistem.....	38
5.2.1	Implementasi Perangkat Keras.....	38
5.2.2	Implementasi Perangkat Lunak.....	40
BAB 6	pengujian dan analisis.....	68
6.1	Pengujian <i>Pulse Sensor</i>	68
6.1.1	Prosedur Pengujian.....	68
6.1.2	Hasil Pengujian.....	68
6.1.3	Analisis Pengujian.....	69
6.2	Pengujian <i>Load Sensor</i>	69
6.2.1	Prosedur Pengujian.....	69
6.2.2	Hasil Pengujian.....	70
6.2.3	Analisis Pengujian.....	70
6.3	Pengujian Modul <i>Bluetooth</i>	70
6.3.2	Hasil Pengujian.....	71



6.3.3 Analisis Pengujian.....	73
6.4 Pengujian Tingkat Kebugaran Kardiorespirasi	73
6.4.2 Hasil Pengujian	73
6.4.3 Analisis Pengujian.....	75
BAB 7 Penutup	76
7.1 Kesimpulan.....	76
7.2 Saran.....	76
Daftar pustaka.....	78



DAFTAR TABEL

Tabel 2.1 Perbedaan pada penelitian sebelumnya.....	5
Tabel 2.2 Klasifikasi VO ₂ max Perempuan.....	8
Tabel 2.3 Klasifikasi VO ₂ max Laki - laki.....	8
Tabel 2.4 Kategori IMT	10
Tabel 2.5 Spesifikasi Arduino nano	11
Tabel 2.6 Spesifikasi Modul <i>Bluetooth</i> HC-05	13
Tabel 2.7 Spesifikasi <i>Load Sensor</i>	15
Tabel 2.8 Spesifikasi HX711.....	16
Tabel 5.1 Komponen Penyusun Panel Kontrol.....	47
Tabel 5.2 Komponen Penyusun <i>Layout</i> Informasi	48
Tabel 5.3 Komponen Penyusun <i>Layout</i> Masukan Informasi	49
Tabel 5.4 Komponen Penyusun <i>Layout</i> Masukan Data Sensor	50
Tabel 5.5 Komponen Penyusun <i>Layout</i> Hasil.....	51
Tabel 5.6 Komponen Penyusun <i>Layout</i> Saran VO ₂ max.....	53
Tabel 5.7 Komponen Penyusun <i>Layout</i> Saran Lanjutan VO ₂ max.....	55
Tabel 5.8 Komponen Penyusun <i>Layout</i> Denyut Nadi Latihan.....	55
Tabel 5.9 Komponen Penyusun <i>Layout</i> Saran IMT	56
Tabel 5.10 Komponen Penyusun <i>Non-visible Components</i>	57
Tabel 6.1 Hasil Pengujian Pulse Sensor	69
Tabel 6.2 Hasil Pengujian Load Sensor.....	70
Tabel 6.3 Pengujian Modul <i>Bluetooth</i> Pada Jarak 1 Meter	71
Tabel 6.4 Pengujian Modul <i>Bluetooth</i> Pada Jarak 5 Meter	71
Tabel 6.5 Pengujian Modul <i>Bluetooth</i> Pada Jarak 10 Meter	72
Tabel 6.6 Pengujian Modul <i>Bluetooth</i> Pada Jarak 12 Meter	72
Tabel 6.7 Pengujian Modul <i>Bluetooth</i> Pada Jarak 14 Meter	72
Tabel 6.8 Hasil Tingkat Kebugaran Terhadap Pengujian Yang Dilakukan	73

DAFTAR GAMBAR

Gambar 2.1 Arduino nano.....	10
Gambar 2.2 Arduino Nano Skematis.....	11
Gambar 2.3 Modul <i>Bluetooth</i> HC-05.....	12
Gambar 2.4 Skematis Modul <i>Bluetooth</i> HC-05	12
Gambar 2.5 <i>Pulse Sensor</i>	13
Gambar 2.6 Skematis <i>Pulse Sensor</i>	14
Gambar 2.7 <i>Load Sensor</i>	14
Gambar 2.8 Skematis <i>Load Sensor</i>	15
Gambar 2.9 Rangkaian <i>wheatstone Bridge</i> Dengan <i>Load Sensor</i>	16
Gambar 2.10 HX711	16
Gambar 2.12 Arsitektur Android.....	17
Gambar 2.13 MIT MIT APP <i>Inventor</i>	17
Gambar 3.1 Alur penelitian	19
Gambar 3.2 Diagram Blok Sistem	22
Gambar 5.1 Desain Sistem	29
Gambar 5.2 Skematik Arduino Nano dengan <i>Pulse Sensor</i>	30
Gambar 5.3 Skematik Arduino Nano dengan <i>Load Sensor</i>	31
Gambar 5.4 Alur Kerja Pada Aplikasi.....	33
Gambar 5.5 Fungsi Pada Aplikasi	34
Gambar 5.6 Arduino dengan <i>Pulse Sensor</i>	36
Gambar 5.7 Arduino dengan <i>load Sensor</i>	37
Gambar 5.8 Arduino nano dengan <i>Pulse Sensor</i>	38
Gambar 5.9 Penggunaan <i>Pulse Sensor</i>	39
Gambar 5.10 Arduino Nano dengan <i>Load Sensor</i>	39
Gambar 5.11 Cara Menggunakan <i>Load Sensor</i>	40
Gambar 5.12 Pustaka Pada Arduino Nano dengan <i>Pulse Sensor</i>	41
Gambar 5.13 Inisialisasi Letak Perangkat Pada Arduino Nano dengan <i>Pulse Sensor</i>	41
Gambar 5.14 <i>Setup</i> program Pada Arduino Nano dengan <i>Pulse Sensor</i>	41



Gambar 5.15 Mendeteksi Denyut Nadi Pada Arduino Nano dengan <i>Pulse Sensor</i>	43
Gambar 5.16 Variabel Pada Program Arduino Nano dengan <i>Pulse Sensor</i>	43
Gambar 5.17 Program Utama Pada Arduino Nano dengan <i>Pulse Sensor</i>	44
Gambar 5.18 Pustaka Pada Arduino Nano dengan <i>Load Sensor</i>	44
Gambar 5.19 Inisialisasi Letak Perangkat Pada Arduino Nano dengan <i>Load Sensor</i>	45
Gambar 5.20 <i>Setup</i> program Pada Arduino Nano dengan <i>Load Sensor</i>	45
Gambar 5.21 Program Utama Pada Arduino Nano dengan <i>Load Sensor</i>	46
Gambar 5.22 Panel Kontrol	46
Gambar 5.23 Layout Informasi	47
Gambar 5.24 Layout Masukan Informasi	48
Gambar 5.25 <i>Layout</i> Masukan Data Sensor	50
Gambar 5.26 Layout Hasil	51
Gambar 5.27 Layout Saran VO ₂ max	53
Gambar 5.28 Layout Saran Lanjutan VO ₂ max	54
Gambar 5.29 Layout Denyut Nadi Latihan	55
Gambar 5.30 <i>Layout</i> Saran IMT	56
Gambar 5.31 <i>Non-visible Components</i>	56
Gambar 5.32 <i>Blocks</i> mengantifkan Bluetooth	57
Gambar 5.33 <i>Blocks</i> Masukan Informasi	58
Gambar 5.34 <i>Blocks</i> Selesai	58
Gambar 5.35 <i>Blocks</i> Ukur Denyut Nadi	59
Gambar 5.36 <i>Blocks</i> Ukur Berat Badan	60
Gambar 5.37 <i>Blocks</i> Hasil	61
Gambar 5.38 <i>Blocks</i> Saran VO ₂ max	63
Gambar 5.39 <i>Blocks</i> Saran IMT	64
Gambar 5.40 <i>Blocks</i> Kembali Hasil	64
Gambar 5.41 <i>Blocks</i> Informasi	64
Gambar 5.42 <i>Blocks</i> Masukan data Ulang	65
Gambar 5.43 <i>Blocks</i> Ukur Ulang	65
Gambar 5.44 <i>Blocks</i> Keluar	66



Gambar 5.45 Blocks menyembunyikan/menampilkan layout Panel kontrol 67



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kebugaran kardiorespirasi berhubungan dengan kemampuan fungsi dari jantung, pembuluh darah, paru – paru, dan otot untuk menyuplai oksigen saat melakukan aktivitas fisik. Terdapat bukti bahwa memiliki kebugaran kardiorespirasi yang baik dapat menurunkan penyakit kardiovaskular seperti serangan jantung, nyeri dada dan stroke karena kebugaran kardiorespirasi meningkatkan sensitivitas insulin, transportasi glukosa , memperbaiki fungsi sistem saraf dan menurunkan denyut jantung (Lee et al., 2010). Oleh karena itu penting untuk memiliki kebugaran kardiorespirasi yang baik.

Kebugaran kardiorespirasi dapat ditingkatkan dengan latihan fisik. Latihan fisik yang dilakukan harus tepat dan tidak berlebihan sehingga terhindar dari risiko *over Training* yang mengakibatkan penurunan bukan peningkatan kemampuan dari tubuh. Latihan fisik yang dilakukan harus mengikuti Salah satu prinsip latihan yaitu meningkatkan secara bertahap beban saat latihan dengan tujuan tubuh akan terbiasa menghadapi beban tersebut. Dengan melakukan latihan fisik untuk meningkatkan kebugaran kardiorespirasi dapat membantu mencegah obesitas sampai 20 tahun ke depan karena dapat membantu menurunkan berat badan (Brien et al., 2007).

Kebugaran kardiorespirasi dapat di ukur melalui seberapa besar maksimum oksigen yang bisa diambil oleh tubuh (VO_{2max}). Pengukuran kebugaran kardiorespirasi melalui VO_{2max} menjadi alternatif termudah untuk mengetahui seberapa bugar tubuh seseorang (Lee et al., 2010). Seseorang memiliki kebugaran kardiorespirasi yang baik mempunyai nilai VO_{2max} berkisar antara 40,5 untuk laki – laki dan perempuan 35 (Heywood, 2006). *Monitoring* kebugaran pada saat ini seperti 2,4 KM *run test*, *quen college step test* dan cooper VO_{2max} *test* memerlukan serangkaian prosedur yang melibatkan kegiatan fisik serta perlu perhatian khusus untuk pelaksanaannya. Tetapi, terdapat metode untuk memprediksi VO_{2max} tanpa melibatkan kegiatan fisik (*non-exercise prediction method*) dengan menggunakan denyut nadi istirahat, berat badan dan umur (Rexhepi et al., 2014). Metode tersebut dipercaya dapat dengan mudah mengukur VO_{2max} seseorang tanpa harus melakukan kegiatan fisik.

Pada skripsi ini akan membuat sebuah alat bernama sistem *monitoring* kebugaran kardiorespirasi. Sistem *monitoring* kebugaran Kardiorespirasi adalah sistem yang memberitahu pengguna informasi kebugaran tubuh dan batas maksimal latihan yang dilakukan. Alat ini menggunakan denyut nadi dan sensor berat badan. Sensor membantu perhitungan VO_{2max} dan zona latihan dimana latihan tersebut tergolong aman atau *over training*. Alat ini diharapkan dapat membantu pengguna dalam mengetahui kebugaran tubuh dan dapat menjadi acuan agar berlatih lebih giat untuk mencapai kebugaran kardiorespirasi yang optimal tanpa melakukan latihan yang berlebihan.

1.2 Rumusan Masalah

Dari uraian latar belakang di atas, rumusan yang akan dibahas adalah sebagai berikut :

1. Bagaimana cara mendapatkan data dari *pulse sensor* untuk kemudian diolah untuk keperluan pengukuran denyut nadi ?
2. Bagaimana cara mendapatkan data dari *load sensor* kemudian diolah untuk keperluan pengukuran berat badan ?
3. Bagaimana perhitungan VO_{2max} dengan menggunakan denyut nadi, berat badan dan umur ?
4. Berapa jarak maksimal yang efektif untuk melakukan pertukaran data menggunakan modul *bluetooth* HC-05 ?
5. Bagaimana akurasi dari sistem dalam pengukuran denyut nadi dan berat badan ?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah ada maka tujuan dari pembuatan sistem ini yaitu :

1. Mengolah data dari *pulse sensor* untuk mengukur denyut nadi.
2. Mengolah data dari *load sensor* untuk mengukur berat badan.
3. Menemukan persamaan yang tepat untuk mengukur VO_{2max} .
4. Mengetahui jarak maksimal yang efektif untuk melakukan pertukaran data menggunakan modul *bluetooth* HC-05.
5. Pembuatan sistem dengan akurasi pengukuran yang baik.

1.4 Manfaat

Manfaat dari terimplementasikannya dan berjalan dengan seharusnya penelitian ini adalah dapat mengetahui kebugaran kardiorespirasi dan cara meningkatkannya karena dapat menanggulangi dini penyakit yang berhubungan dengan kardiovaskular.

1.5 Batasan masalah

Dalam penelitian ini untuk mempermudah pembuatan dan percobaan maka dibatasi :

1. Subjek dari penelitian ini adalah orang yang sehat secara fisik tidak memiliki kelainan jantung.
2. Subjek penelitian harus dalam keadaan isitirahat tidak sehabis melakukan olahraga berat.
3. Klasifikasi kebugaran hanya untuk usia 20 tahun keatas.

1.6 Sistematika Pembahasan/Laporan

Sistematika pembahasan/laporan memuat penjelasan mengenai tujuan dari masing – masing bab dalam laporan ini. Sistematika dalam laporan ini adalah sebagai berikut :

BAB 1 Pendahuluan

Berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, sistematika pembahasan, dan diakhiri dengan jadwal penelitian/pelaksanaan. Dalam bab ini secara umum pembahasannya berisi tentang harapan supaya pembaca bisa menemukan latar belakang atau alasan secara teoritis dari sumber bacaan. Dengan demikian disimpulkan bab ini menjadi dasar atau titik acuan metodologis dari bab-bab selanjutnya.

BAB 2 Landasan Kepustakaan

Memuat kajian teori yang meliputi pengertian komponen yang digunakan, dipaparkan tentang fungsi setiap komponen dan teori pendukung berjalannya sistem. Secara garis besar bab ini memuat tentang bagaimana sistem ini dibuat, komponennya apa saja. Lalu fungsi lain juga dimasukkan dalam bab ini.

BAB 3 Metodologi

Membahas metode yang akan digunakan dalam menyelesaikan permasalahan dalam pembuatan sistem *monitoring* kebugaran kardiorespirasi.

BAB 4 Rekayasa Kebutuhan

Menjelaskan secara rinci kebutuhan pada pelaksanaan.

BAB 5 Perancangan dan implementasi

Menjelaskan bagaimana rancangan sistem dan cara mengimplementasikannya agar sistem dapat berjalan dengan seharusnya.

BAB 6 Pengujian dan analisis

Memuat bagaimana cara sistem di uji dan hasil pengujian beserta analisis pada pengujian.

BAB 7 Penutup

Memuat kesimpulan akhir termasuk saran – saran untuk kepentingan pengembangan dari penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka membahas penelitian sebelumnya yang menjadi dasar pembuatan penelitian ini.

2.1 Kajian Pustaka

Kajian pustaka yang digunakan sebagai acuan pembuatan adalah *paper online* yang berjudul “*The Importance of Cardiorespiratory Fitness*”. Pada *paper* tersebut tersebut dijelaskan bahwa kebugaran Kardiorespirasi diperlukan karena kebugaran Kardiorespirasi akan mengurangi kemungkinan terkena penyakit kardiovaskular. Kebugaran kardiorespirasi dapat di ukur melalui maksimum pengambilan oksigen pada tubuh (VO_{2max}). Pengukuran VO_{2max} di jelaskan dalam *paper* “*Prediction of VO_{2max} based on age, body mass, and resting heart rate*” menemukan hubungan antara umur, berat badan dan HR_{rest} secara statistik dengan nilai korelasi sebesar 0.688 sehingga muncul metode *non-exercise prediction method*. Hasil dibandingkan dengan pengukuran VO_{2max} menggunakan metode *exercise based* yaitu *astrand bike*. Pada hasil tersebut dapat disimpulkan ada selisih perbedaan antar pengukuran langsung dan tidak langsung ($p = 0.782$), tetapi perbedaan tersebut tidak terlalu signifikan sehingga membuat ini layak untuk digunakan (Rexhepi et al, 2014).

Penggunaan *pulse sensor* untuk kebutuhan pengukuran denyut nadi didapat dari penelitian yang berjudul “*Rancang bangun alat pendeteksi denyut jantung dan suhu tubuh menggunakan pulse sensor dan sensor LM35*”. Pada penelitian tersebut menjelaskan bagaimana penggunaan *pulse sensor* untuk pengukuran denyut nadi. Peletakan sensor pada penelitian yaitu *pulse sensor* pada ujung jari. Peletakan sensor terbilang efektif karena dapat mengukur dengan tepat walaupun terdapat sedikit *error* dengan persentase 4.4% untuk *pulse sensor* dari pengukuran langsung (Akbar, 2016).

Penggunaan *load sensor* untuk pengukuran berat badan didapat dari penelitian yang berjudul “*Rancang bangun alat penghitung indeks massa tubuh*”. Pada penelitian tersebut menjelaskan bagaimana penggunaan *load sensor* untuk pengukuran denyut nadi. Dalam implementasinya *load sensor* menggunakan *load sensor* yang berada pada timbangan digital. Timbangan digital dirangkai ulang kemudian dihubungkan semua *load sensor* ke penguat sinyal dan diteruskan ke mikrokontroler untuk dibaca keluarannya. Hasil dari pengukuran berat badan mendapatkan *error* sebesar 3.7 % dibandingkan dengan pengukuran langsung menggunakan timbangan digital yang tidak dimodifikasi (Taalongonan et al., 2014).

Sistem *monitoring* kebugaran kardiorespirasi menggabungkan kajian pustaka yang disebutkan untuk menambahkan fitur pengukuran kebugaran, target latihan dan pengukuran indeks masa tubuh. Pada tabel 2.1 menjelaskan perbedaan pada penelitian sebelumnya yang digunakan dalam kajian pustaka ini.

Tabel 2.1 Perbedaan pada penelitian sebelumnya

No	Judul Penelitian	Persamaan Topik	Perbedaan Topik
1	<i>Prediction of VO2max based on age, body mass, and resting heart rate</i>	Prediksi VO2max menggunakan umur, berat badan dan HR _{rest}	Tidak di implementasikan
2	<i>Rancang bangun alat pendeteksi denyut jantung dan suhu tubuh menggunakan pulse sensor dan sensor LM35</i>	Mengukur denyut nadi menggunakan pulse sensor	<ol style="list-style-type: none"> 1. Tidak menggunakan arduino nano 2. Tidak menggunakan load sensor 3. Tidak mengukur berat badan 4. Tidak mengukur VO2max 5. Tidak mengukur denyut nadi latihan 6. Tidak mengukur indeks masa tubuh 7. Tidak menampilkan pada OS berbasis android
3	Rancang bangun alat penghitung indeks massa tubuh	<ol style="list-style-type: none"> 1. Mengukur berat badan menggunakan load sensor 2. Mengukur indeks masa tubuh 	<ol style="list-style-type: none"> 1. Tidak menggunakan arduino nano 2. Tidak menggunakan pulse sensor 3. Tidak mengukur denyut nadi 4. Tidak mengukur VO2max 5. Tidak mengukur denyut nadi latihan 6. Tidak menampilkan pada OS berbasis android

Penambahan fitur - fitur tersebut bertujuan agar terciptanya sistem yang dapat menjadi acuan untuk meningkatkan kebugaran Kardiorespirasi. Selain penambahan fitur tersebut alat *monitoring* kebugaran Kardiorespirasi juga dilengkapi dengan modul *bluetooth* yang dapat menghubungkannya dengan perangkat berbasis OS android demi terciptanya tampilan menarik dan mempunyai prinsip portabilitas.

2.2 Dasar Teori

2.2.1 Kebugaran kardiorespirasi

Kebugaran kardiorespirasi berhubungan dengan kemampuan fungsi dari jantung, pembuluh darah, paru – paru, dan otot. Mempunyai kebugaran kardiorespirasi yang tinggi berarti meningkatkan sensitivitas insulin, meningkatkan transportasi glukosa, memperbaiki fungsi sistem saraf dan menurunkan denyut jantung. Kebugaran kardiorespirasi juga sama halnya dengan ambilan oksigen maksimum dalam tubuh (VO_{2max}). Mempunyai kebugaran kardiorespirasi berarti mengurangi risiko terhadap penyakit kardiovaskular seperti serangan jantung, nyeri dada, atau stroke (lee et la, 2010). Hal ini menunjukkan pentingnya memiliki kebugaran fisik dalam kehidupan.

Kebugaran kardiorespirasi dapat ditingkatkan dengan latihan fisik. Latihan fisik yang dilakukan harus tepat dan tidak berlebihan sehingga terhindar dari risiko *over Training* yang mengakibatkan penurunan bukan peningkatan kemampuan dari tubuh. Latihan fisik yang dilakukan harus mengikuti Salah satu prinsip latihan yaitu meningkatkan secara bertahap beban saat latihan dengan tujuan tubuh akan terbiasa menghadapi beban tersebut. Dengan melakukan latihan fisik untuk meningkatkan kebugaran kardiorespirasi dapat membantu mencegah obesitas sampai 20 tahun ke depan karena dapat membantu menurunkan berat badan (Brien et al., 2007).

2.2.2 Denyut Nadi

Denyut nadi adalah hitungan kontraksi jantung yang dirambatkan sebagai gelombang pada dinding pembuluh darah per menit (BPM). Pada masa istirahat rata – rata denyut nadi manusia dewasa sekitar 70 BPM untuk laki – laki dan perempuan 75 BPM. Denyut nadi dapat diukur di beberapa bagian tubuh yaitu :

- a. Daerah pergelangan tangan
- b. Daerah leher
- c. Bagian dalam bahu
- d. Pangkal paha
- e. Daerah kaki bagian belakang
- f. Daerah kaki bagian tengah
- g. Daerah belakang lutut
- h. Daerah atas perut
- i. Di dada

Pengukuran denyut nadi dihitung permenit yang artinya dalam waktu 1 menit berhenti untuk mengetahui hasil denyut nadi atau dalam waktu 15 detik dengan hasil perhitungan di kali dengan 4. Denyut nadi memiliki beberapa kategori dan fungsi masing – masing yaitu :

1. Denyut nadi istirahat

Denyut nadi istirahat (HR_{rest}) adalah perhitungan denyut nadi yang didapatkan ketika sedang tidak sehabis melakukan kegiatan fisik apapun. Denyut nadi istirahat dipengaruhi oleh umur, latihan, suhu, emosi dan genetika.

2. Denyut nadi Maksimal

Denyut nadi maksimal (HR_{max}) adalah denyut maksimal yang dapat dicapai oleh seseorang. Denyut nadi maksimal dapat menjadi indikasi apakah kegiatan fisik yang kita lakukan berlebihan atau tidak. Denyut nadi maksimal dapat diukur dengan menggunakan rumus pada persamaan 2.1 (Jackson et al. 2007).

$$HR_{max} = 206,9 - (0,67 \times Umur) \quad (2.1)$$

3. Denyut nadi latihan

Denyut nadi latihan adalah jumlah denyut nadi yang ingin dicapai saat latihan. terdiri dari beberapa klasifikasi zona persentase yaitu :

a. Zona penyembuhan (60% - 70% HR_{max})

Latihan pada zona meningkatkan daya tahan dan kapasitas aerobik. Pada zona ini di mana lemak mudah terbakar yang dapat mengakibatkan penurunan berat badan sekaligus mengisi kembali otot dengan glycogen.

b. Zona aerobik (70% - 80% HR_{max})

Latihan pada zona ini akan meningkatkan sistem kardiovaskular yang mengakibatkan peningkatan kemampuan tubuh untuk transportasi oksigen dan membuang karbon dioksida.

c. Zona anaerobik (80% - 90%)

Pada zona ini energi utama yaitu lemak akan cepat terkuras dan glycogen dalam otot akan cepat habis. Dengan latihan yang tepat pada zona ini akan mencegah energi cepat habis sehingga tubuh akan terbiasa secara alami.

d. Zona merah (90% - 100% HR_{max})

Latihan pada zona memungkinkan tapi hanya sebentar. Harus memiliki kondisi yang benar – benar bugar jika tidak akan mengarah pada *over training*. Pengukuran target denyut nadi latihan dapat menggunakan persamaan 2.2 sebagai berikut (Karvonen, 1988) :

$$\text{Target} = ((HR_{max} - HR_{rest}) \times \%Zona) + HR_{rest} \quad (2.2)$$

2.2.3 VO_{2max} dan Non-Exercise Prediction Method

VO_{2max} adalah kadar maksimal oksigen yang bisa diambil oleh tubuh, VO_{2max} menentukan performa aerobik (Turnley, 2010). VO_{2max} memiliki satuan ml/kg/menit, dapat diklasifikasikan dalam beberapa kelompok umur dan jenis

kelamin. Klasifikasi pada kelompok tersebut memiliki beberapa kategori yaitu cukup, baik, baik sekali, sangat tinggi (Heywood, 2006).

Tabel 2.2 Klasifikasi VO₂max Perempuan

Umur	Jelek	Cukup	Baik	Baik sekali	Sangat tinggi
20 – 29	<36	36 – 39	40 - 43	44 – 49	>49
30 – 39	<34	34 – 36	37 – 40	41 – 45	>45
40 – 49	<32	32 – 34	35 – 38	39 – 44	>44
50 – 59	<25	25 – 28	29 – 30	31 – 34	>34
60 - 69	<26	26 – 28	29 - 31	32 – 35	>35
70 - 79	<24	24 - 26	27 - 29	30 - 35	>35

Sumber: Brianmac.co.uk

Tabel 2.3 Klasifikasi VO₂max Laki - laki

Umur	Jelek	Cukup	Baik	Baik sekali	Sangat tinggi
20 – 29	<42	42 – 45	46 – 50	51 – 55	>55
30 – 39	<41	41 – 43	44 – 47	48 – 53	>53
40 – 49	<38	38 – 41	42 – 45	46 – 52	>52
50 – 59	<35	35 – 37	38 – 42	43 – 49	>49
60 - 69	<31	31 – 34	35 – 38	39 – 45	>45
70 - 79	<28	28 - 30	31 - 35	36 - 41	>41

Sumber: Brianmac.co.uk

Pada saat melakukan pekerjaan berat yang terus bertambah tetapi tidak diikuti oleh kenaikan frekuensi denyut jantung dan frekuensi tersebut tetap konstan maka pada keadaan inilah ambilan oksigen seseorang dikatakan maksimum. Terdapat beberapa faktor yang mempengaruhi VO_{2max} antara lain :

1. Jenis kelamin
2. Umur
3. Komposisi dan ukuran tubuh
4. Genetika
5. Ketinggian
6. Latihan

VO_{2max} dapat ditingkatkan dengan latihan fisik. Latihan dengan intensitas yang tepat dapat dengan efektif meningkatkan kadar VO_{2max} seseorang. Latihan pada dengan intensitas tinggi lebih efektif meningkatkan kadar VO_{2max} dari pada latihan pada intensitas rendah (Helgerud et al., 2007). Latihan pada intensitas tinggi ditandai dengan denyut nadi mencapai persentase 80% - 100% dari denyut nadi maksimal dengan catatan berhati – hati untuk tidak melebihi kadar denyut nadi maksimal yaitu 100%. Terdapat beberapa metode latihan untuk meningkatkan VO_{2max} antara lain :

1. *Long Slow Distance Running* (LSD)

LSD diawali dengan pemanasan terlebih dahulu kemudian dilanjutkan dengan langkah sebagai berikut :

- a. Berlari dengan kecepatan sedang
 - b. Dilakukan pada kondisi denyut nadi 60 – 70 % dari denyut nadi maksimal
LSD dilakukan 3 – 5 kali dalam seminggu dengan maksimal jarak 15 mill atau maksimal 1 jam berlari.
2. *Lactate threshold running (LT)*
LT diawali dengan pemanasan terlebih dahulu kemudian dilanjutkan dengan langkah sebagai berikut :
- a. lari ringan 10 menit
 - b. Lari cepat dengan jarak antara 800 – 1200 meter
 - c. Lari ringan dengan jarak 400 meter
 - d. Ulang kembali langkah awal sampai total jarak lari cepat 5000 meter
- Untuk mendapat hasil yang maksimal dalam melakukan LT kecepatan berlari saat melakukan lari cepat tetap stabil tanpa penurunan kecepatan lari dan LT dilakukan pada saat kondisi denyut nadi berada pada 85% dari denyut nadi maksimal.
3. *15/15 interval running (15/15)*
15/15 diawali dengan pemanasan terlebih dahulu kemudian dilanjutkan dengan langkah sebagai berikut :
- a. Lari ringan selama 15 detik
 - b. Lari cepat selama 15 detik
 - c. Ulangi langkah tersebut sampai mencapai perulangan antara 12 – 20 kali
- Untuk mendapat hasil yang maksimal dalam melakukan 15/15 dilakukan pada saat kondisi denyut nadi berada pada 90% - 95% dari denyut nadi maksimal.
4. *4 x 4-min interval running (4 x 4)*
4 x 4 diawali dengan pemanasan terlebih dahulu kemudian dilanjutkan dengan langkah sebagai berikut :
- a. 4 menit lari cepat
 - b. 3 menit lari ringan
 - c. Ulangi langkah tersebut sampai mencapai total waktu 31 – 35 menit latihan
- Untuk mendapat hasil yang maksimal dalam melakukan 4 x 4 kecepatan berlari saat melakukan lari cepat tetap stabil tanpa penurunan kecepatan lari dan 4 x 4 dilakukan pada saat kondisi denyut nadi berada pada 90% - 95% dari denyut nadi maksimal.

Setelah selesai melakukan latihan tersebut harus melakukan pendinginan pada 70% dari denyut nadi maksimal untuk memulihkan tubuh dari latihan yang dilakukan.

Ada berbagai macam cara dalam pengukuran VO_{2max} yaitu dengan beban kerja maksimal menggunakan spidometer di sirkuit atau di dalam laboratorium atau estimasi VO_{2max} menggunakan beberapa metode beban kerja submaksimal seperti *Quen collage step test*. Tetapi terdapat metode pengukuran VO_{2max} tanpa harus melakukan serangkaian kegiatan (*non-exercise prediction method*) menggunakan persamaan yang melibatkan umur, berat badan dan HR_{rest} . Prediksi

VO_{2max} menggunakan *non-exercise prediction method* dapat ditulis menjadi persamaan 2.3 (Rexhepi et al, 2014).

$$VO_{2max} = 3.542 + (-0.014 \times Umur) + (0.015 \times BeratBadan) + (-0.011 \times HR_{rest}) \quad (2.3)$$

Hasil dari pada persamaan 2.3 adalah nilai VO_{2max} dalam satuan L/min, untuk dapat klasifikasi nilai dari VO_{2max} pada tabel 2.2 dan tabel 2.3 maka satuan tersebut harus diubah menjadi ml/kg/min dengan persamaan 2.4 (Brianmac.co.uk).

$$VO_{2max}(ml/kg/min) = \frac{VO_{2max}(L/min) \times 1000}{BeratBadan} \quad (2.4)$$

2.2.4 Indeks Massa Tubuh

Indeks massa tubuh atau IMT adalah nilai dari pembagian antara berat badan dengan tinggi badan, dapat dituliskan dengan persamaan 2.5.

$$IMT = BeratBadan(Kg) / (TinggiBadan(m))^2 \quad (2.5)$$

IMT merupakan salah satu metode paling mudah dalam pengukuran lemak tubuh. Mengetahui ukuran lemak dalam tubuh dapat membantu pencegahan penyakit seperti tekanan darah tinggi, kolesterol dan gangguan jantung. Hasil nilai dari perhitungan IMT memiliki beberapa kategori yang harus diketahui untuk pencegahan penyakit yaitu :

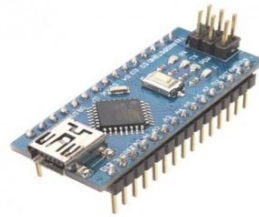
Tabel 2.4 Kategori IMT

IMT	Kategori
< 18,5	Kurang berat badan
18,5 – 22,9	Berat badan normal
≥ 23,0	Kelebihan berat badan
23,0 – 24,9	Berisiko obes
25,0 – 29,9	Obes I
≥ 30,0	Obes II

Sumber: *Center for Obesity Research and Education 2007*

2.2.5 Arduino Nano

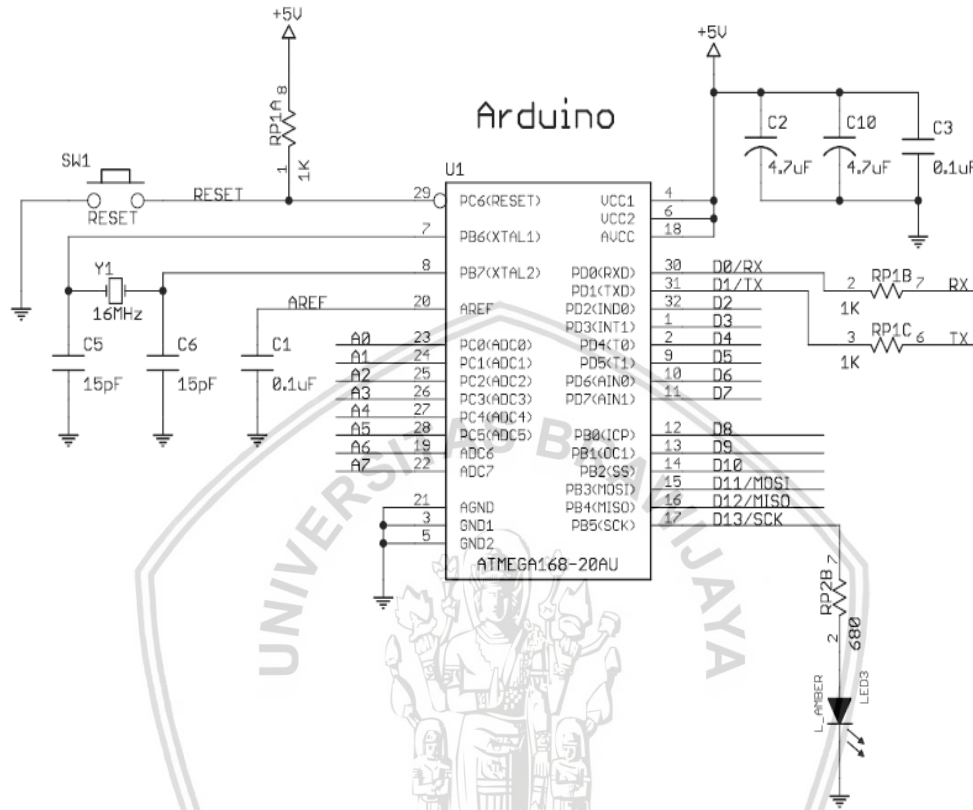
Arduino nano adalah sebuah minimum sistem yang menjalankan fungsi sebagai pengendali dan pusat menjalankan algoritma.



Gambar 2.1 Arduino nano

Sumber: Arduino.cc

Arduino nano adalah sebuah mikrokontroler yang menggunakan ATmega328p sebagai standarnya dan beberapa komponen pendukung untuk berjalannya sistem. Berikut adalah komponen penyusun Arduino nano yang dituangkan dalam skematis pada gambar 2.2.



Gambar 2.2 Arduino Nano Skematis

Sumber: Arduino.cc

Pada skematis yang tersusun dalam gambar 2.2 arduino nano mempunyai spesifikasi yaitu sebagai berikut :

Tabel 2.5 Spesifikasi Arduino nano

No	Spesifikasi	Keterangan
1	Mikrokontroler	Atmega328P
2	Voltase operasi	5 V
3	Voltase input	7 – 12 V
4	Arus Pin I/O	40 mA
5	Memori Flash	32 KB
6	SRAM	2 KB
7	EEPROM	1 KB
8	Kecepatan clock	16 MHz

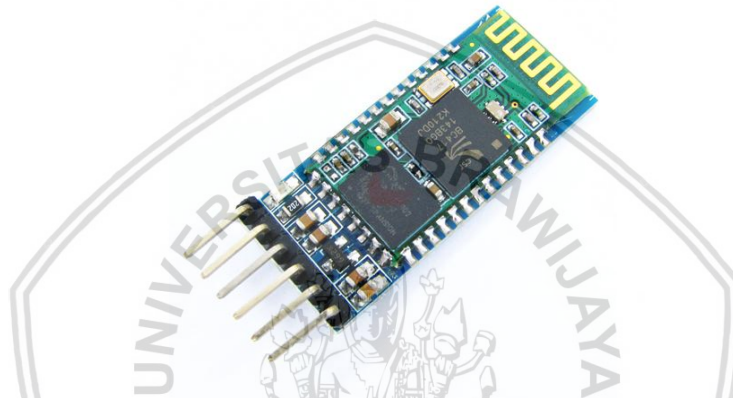
Sumber: Arduino.cc

Dalam menjalankan fungsinya arduino nano membutuhkan program arduino IDE. Program tersebut berfungsi sebagai perantara antar arduino dan programmer yang menggunakan arduino. Pada program itulah algoritma untuk menjalankan arduino nano dibuat dan di masukan ke dalam arduino nano.

2.2.6 Modul *Bluetooth* HC-05

Bluetooth adalah salah satu standar teknologi dalam jaringan nirkabel. *Bluetooth* memiliki jangkauan pendek karena menggunakan panjang gelombang UHF sebesar 2,4 – 2,48 GHz. Jarak jangkau dari *bluetooth* sekitar 10 meter.

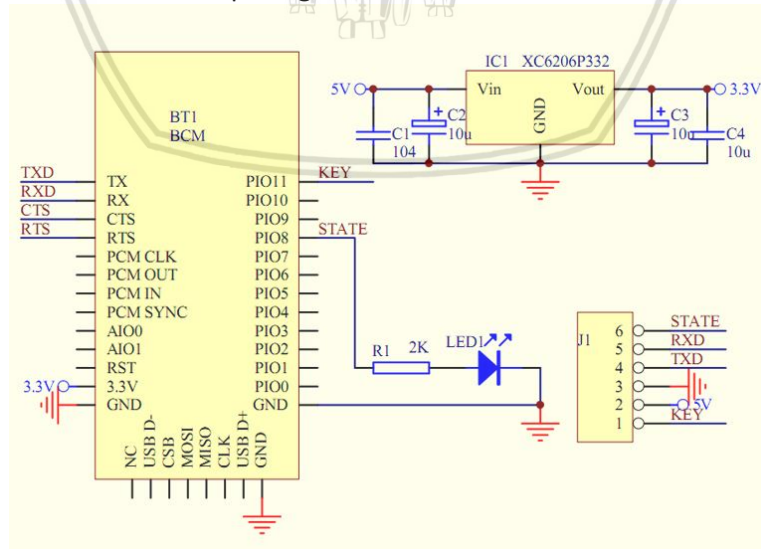
Modul *Bluetooth* HC-05 memungkinkan mikrokontroler untuk menggunakan fungsi *bluetooth* dan dapat bertukar data pada suatu perangkat.



Gambar 2.3 Modul *Bluetooth* HC-05

Sumber: *Datasheet Bluetooth Modul Bluetooth HC-05*

Berikut adalah komponen penyusun modul *bluetooth* HC-05 modul yang dituangkan dalam skematis pada gambar 2.4.



Gambar 2.4 Skematis Modul *Bluetooth* HC-05

Sumber: *Datasheet Bluetooth Modul Bluetooth HC-05*



Dalam penggunaannya modul *Bluetooth* HC-05 memiliki spesifikasi yang terdapat pada tabel 2.7.

Tabel 2.6 Spesifikasi Modul *Bluetooth* HC-05

Sensitifitas	-80dBm
Kekuatan transmisi	+4dBm RF
Voltase operasi	1,8 – 3.6 V

Sumber: *Datasheet* Modul *Bluetooth* HC-05

Dalam menjalankan fungsinya modul *bluetooth* HC-05 memiliki beberapa pin sebagai berikut :

1. EN
Berfungsi untuk mengaktifkan *AT Command Setup*. Secara *Default* mode yang digunakan adalah mode *data*. *AT command* adalah sekumpulan perintah dalam huruf kapital yang berguna untuk konfigurasi modul sedangkan mode *data* membuat modul siap untuk menerima ataupun mengirim data ke perangkat lain.
2. VCC
Pin untuk sumber tegangan, sumber tegangan yang dibutuhkan adalah 5 v untuk pengoperasiannya.
3. GND
Pin yang berfungsi sebagai *ground* pada modul. Hubungkan pin ini pada *ground* rangkaian agar modul dapat berfungsi dengan benar.
4. TXD
Pin yang berfungsi sebagai pengirim data dari modul.
5. RXD
Pin yang berfungsi sebagai penerima data dari modul.
6. State
Pin yang berfungsi sebagai pemberi informasi apakah perangkat terhubung atau tidak dengan perangkat lain.

2.2.7 *Pulse Sensor* (SEN-11574)

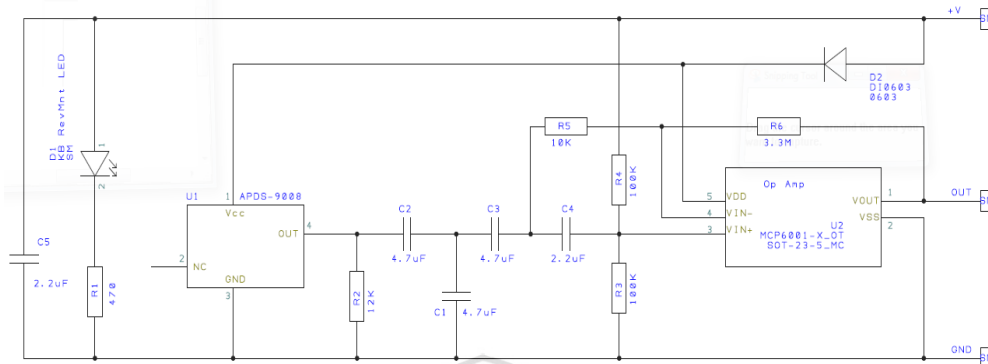
Pulse sensor digunakan untuk merekam denyut nadi pada tubuh. Dalam penggunaannya *pulse sensor* dapat diletakan di ujung jari, telinga ataupun bagian tubuh lain yang terdapat pembuluh darahnya.



Gambar 2.5 *Pulse Sensor*

Sumber: Pulsesensor.com

Cara kerja pulse sensor memancarkan cahaya dari LED kemudian pantulan dari LED tersebut akan di rekam nilainya oleh sensor cahaya untuk mendapatkan hasil dari denyut nadi.



Gambar 2.6 Skematis Pulse Sensor

Sumber: Theorycircuit.com

Pulse sensor terdiri dari 3 pin dalam penggunaannya yaitu :

1. Sumber tenaga
Operasi tenaga pada sensor ini diantara 3V – 5V agar dapat berfungsi dengan semestinya.
2. *Ground*
Sebagai *Ground* pada perangkat.
3. Sinyal
Sebagai keluaran hasil dari rekaman sensor. Hasil tersebut akan diolah kembali dalam program.

2.2.8 Load sensor

Load sensor bekerja dengan cara mengukur perubahan resistensi pada sensor yang terjadi karena tekanan yang diberikan.



Gambar 2.7 Load Sensor

Sumber: Sparkfun.com

Load sensor yang digunakan pada sistem adalah *load sensor* yang memiliki batas maksimal pengukuran sampai 50 Kg, adapun spesifikasi dari *load sensor* pada tabel 2.7.

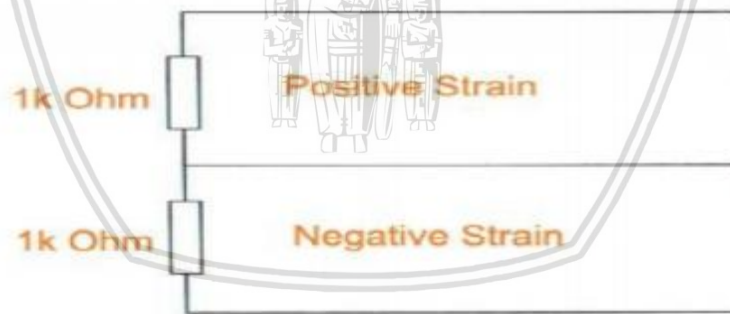
Tabel 2.7 Spesifikasi Load Sensor

Kesalahan komprehensif	0.05 mv/v
Sensitivitas keluaran	1.0 ± 0.1
Voltase eksitasi	< 10 V
Suhu operasi	0°C - 50°C

Sumber: Sparkfun.com

Load sensor dalam penggunaannya menggunakan 3 pin dapat dilihat pada gambar 2.8, Berikut adalah pin – pin pada *load sensor* :

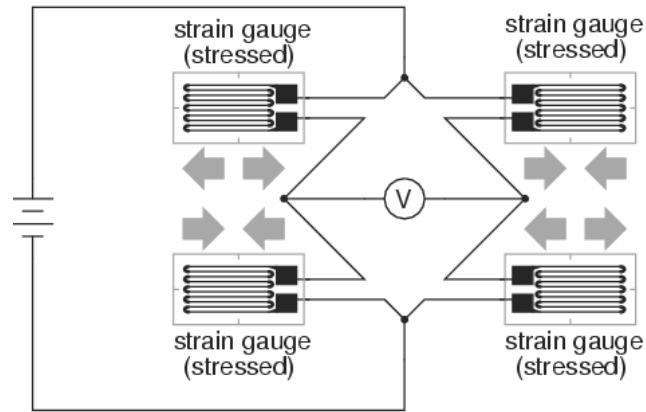
1. Sumber tenaga
Operasi tenaga pada sensor ini diantara < 10 V agar dapat berfungsi dengan semestinya.
2. *Ground*
Sebagai *Ground* pada perangkat.
3. Sinyal
Sebagai keluaran hasil dari rekaman sensor. Hasil tersebut akan diolah kembali dalam program.



Gambar 2.8 Skematis Load Sensor

Sumber: www.Sparkfun.com

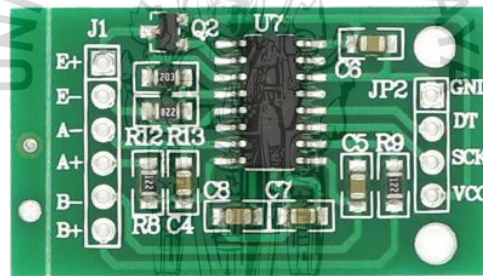
Load sensor dalam penggunaannya untuk mengukur berat badan membutuhkan 4 *load sensor* dihubungkan menggunakan rangkaian *wheatstone bridge* yang dapat dilihat pada gambar 2.9. Tujuan dari digunakannya *wheatstone bridge* adalah untuk dapat mengukur perubahan resistansi yang kecil pada sensor dan agar pengukuran dapat melebihi 50 Kg dari kemampuan awal sensor sehingga mampu untuk mengukur berat badan manusia.



Gambar 2.9 Rangkaian *wheatstone Bridge* Dengan *Load Sensor*
 Sumber: Sparkfun.com

2.2.9 HX711

HX711 adalah *analog digital converter* (ADC) yang merubah masukan analog menjadi digital sehingga dapat dibaca mikrokontroler dengan presisi 24 bit. HX711 didesain untuk pengukuran berat dan digunakan bersama dengan sensor tersusun secara *bridge* seperti *wheatstone bridge*.



Gambar 2.10 HX711
 Sumber: Datasheet HX711

Terdapat 2 saluran yang digunakan untuk digunakan yaitu saluran A dan saluran B dengan fitur *programmable gain amplifier* (*gain* yang dapat diprogram) . Saluran A dapat diprogram dengan *gain* 128 atau 64. Saluran B memiliki *gain* yaitu 32. HX711 memiliki masukan *clock* yang fleksibel sehingga dapat menerima *clock* eksternal, kristal atau *on-chip oscillator* tanpa perlu komponen tambahan.

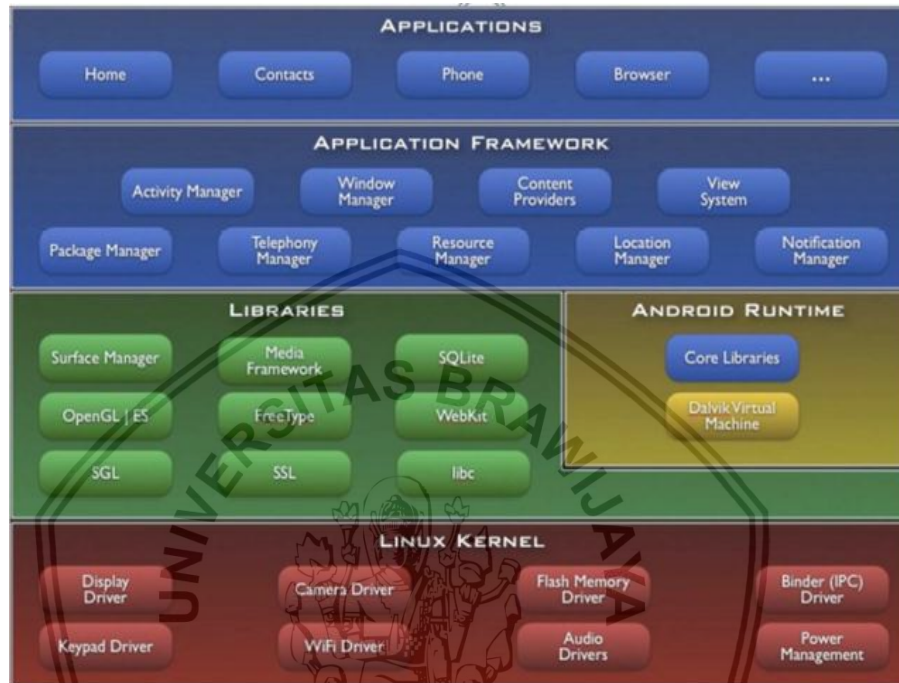
Tabel 2.8 Spesifikasi HX711

Voltase operasi	2.6 – 5.5 V
<i>Programmable gain amplifier</i>	32, 64, 128
Suhu operasi	-40°C - 85°C

Sumber: Datasheet HX711

2.2.10 OS android

Android adalah sistem operasi berbasis linux digunakan untuk perangkat *mobile* yang mencakup fungsi sistem operasi, *middleware*, dan aplikasi. Android bersifat terbuka yang artinya para pengembang aplikasi bebas berkreasi di dalamnya.



Gambar 2.11 Arsitektur Android

Sumber: Developer.android.com

Android menjadi salah satu sistem operasi untuk *smartphone* yang populer. Kepopuleran android membuat android menjadi pesaing utama Apple.

Pembuatan aplikasi berbasis android pada skripsi ini menggunakan aplikasi berbasis WEB MIT APP *Inventor*.



Gambar 2.12 MIT MIT APP *Inventor*

Sumber: Appinventor.mit.edu

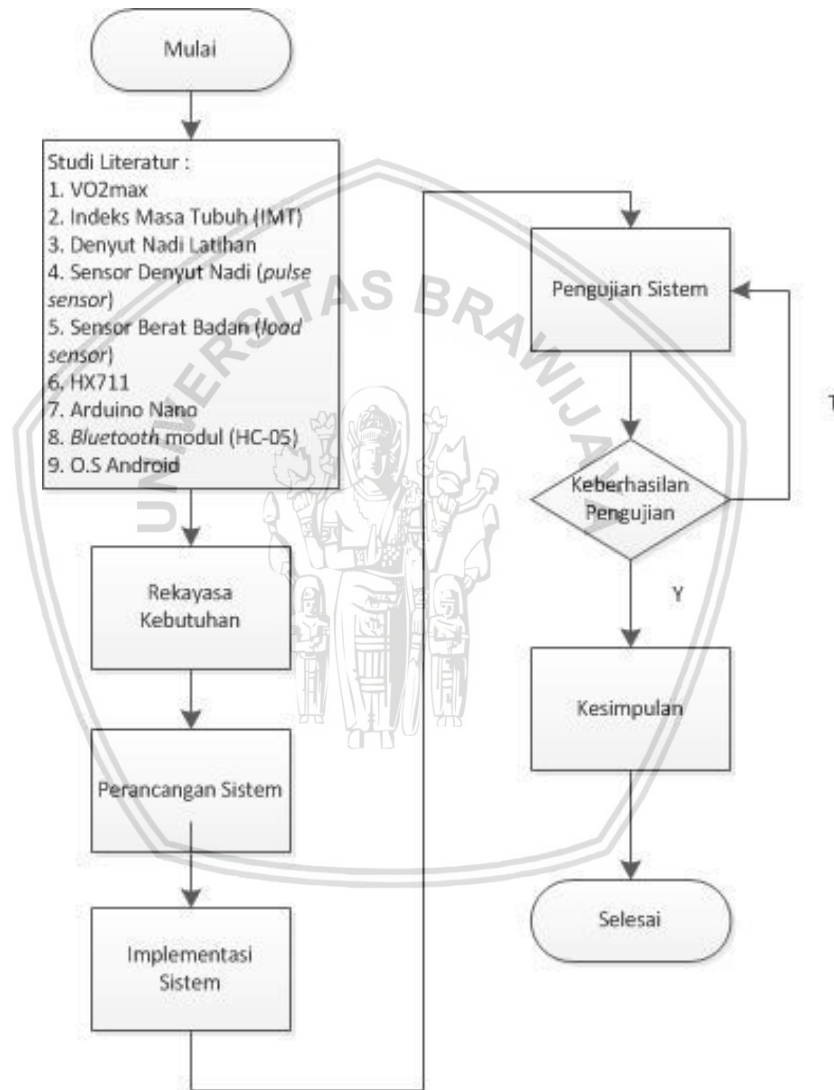


MIT APP *Inventor* menyederhanakan pembuatan aplikasi berbasis bahasa kompleks ke dalam *drag and drop building block* dalam bentuk visual. Hal tersebut memudahkan pembuatan aplikasi berbasis OS android.



BAB 3 METODOLOGI

Langkah Pembuatan dari sistem *monitoring* kebugaran kardiorespirasi ini dimulai dari studi literatur dan dasar teori guna mencari informasi mendukung dalam pembuatannya. Pembuatan sistem bersifat implementif yang artinya sistem diterapkan langsung ke dunia nyata bukan hanya analisis. Alur dalam pembuatan sistem ini terdapat pada gambar 3.1 yang menjelaskan sistematis jalannya pembuatan sistem.



Gambar 3.1 Alur penelitian

3.1 Studi Literatur

Studi literatur bertujuan untuk menyusun dasar teori yang digunakan untuk menunjang pembuatan sistem. Studi literatur yang dibutuhkan yaitu sebagai berikut :

1. VO_{2max}

Studi literatur tentang VO_{2max} bertujuan untuk mencari definisinya ,cara perhitungan dan hubungannya dengan kebugaran kardiorespirasi. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait VO_{2max} .
2. Indeks masa tubuh (IMT)

Studi literatur tentang IMT bertujuan untuk mencari definisinya dan cara perhitungannya. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait IMT.
3. Denyut nadi latihan
Studi literatur tentang suhu tubuh bertujuan untuk mencari definisinya dan cara pengukuran. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait denyut nadi latihan.
4. *Pulse sensor*

Studi literatur tentang *Pulse sensor* bertujuan untuk mencari cara penggunaan yang benar. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait *Pulse sensor*.
5. *Load sensor* (50kg)

Studi literatur tentang *Load sensor* bertujuan untuk mencari cara penggunaan yang benar. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait *Load sensor*.
6. HX711
Studi literatur tentang *Load sensor* bertujuan untuk mencari cara penggunaan yang benar. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait HX711.
7. Arduino nano
Studi literatur tentang arduino NANO bertujuan untuk mencari cara penggunaan yang benar. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait arduino NANO.
8. Modul *bluetooth* HC-05
Studi literatur tentang modul *Bluetooth* HC-05 bertujuan untuk mencari cara penggunaan yang benar. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait HC-05.
9. OS android
Studi literatur tentang OS android bertujuan untuk mencari definisi dan cara pembuatan aplikasi pada OS tersebut. Dilakukan dengan mencari penelitian terdahulu dan mengkaji informasi terkait OS android.

3.2 Analisis Kebutuhan

Analisis kebutuhan perangkat keras dan perangkat lunak bertujuan untuk mendapatkan system yang dapat diimplementasikan sebaik mungkin pada rancang bangun sistem *monitoring* kebugaran kardiorespirasi. Dengan tujuan sistem dapat memberikan hasil kebugaran tubuh yang akurat. Untuk berjalan dengan baik diperlukan masukan dari sensor yang digunakan dan pengolahan data yang tepat.

3.2.1 Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yaitu menganalisis kebutuhan perangkat keras apa saja yang akan digunakan untuk pembuatan sistem pada penelitian ini. Perangkat keras yang digunakan adalah :

1. Arduino nano
2. *Pulse sensor*
3. *Load sensor* (50 Kg)
4. HX711
5. Modul *Bluetooth* (HC-05)

3.2.2 Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak yaitu menganalisis kebutuhan perangkat lunak apa saja yang akan digunakan untuk pembuatan sistem pada penelitian ini. Perangkat lunak yang digunakan yaitu :

1. arduino IDE
2. MIT APP *Inventor*

3.2.3 Kebutuhan Fungsional

Analisis kebutuhan fungsional yaitu menganalisis kebutuhan fungsional apa saja yang akan diraih dalam berjalannya sistem. Kebutuhan fungsional meliputi :

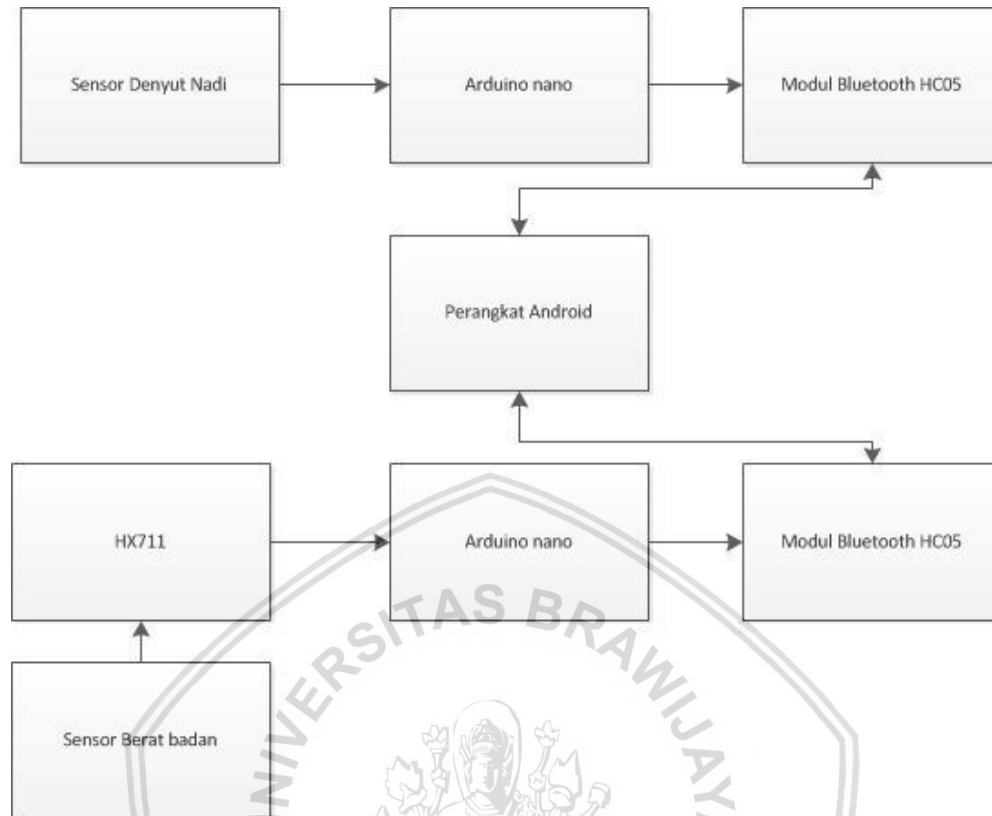
1. Perhitungan VO₂max menggunakan non-exercise prediction method dan klasifikasi hasil
2. Perhitungan IMT dan klasifikasi hasil.
3. Perhitungan zona latihan dan menampilkan hasil zona – zona latihan yang sebaiknya dilakukan.
4. Pemberi masukan terkait meningkatkan kebugaran kardiorespirasi dan cara menggapai nilai IMT yang ideal

3.3 Perancangan Sistem

Perancangan ditujukan untuk menjelaskan peletakan komponen dari sistem dan algoritma yang berjalan pada sistem. Perancangan terbagi menjadi dua yaitu perancangan perangkat keras dan perangkat lunak

3.3.1 Perancangan perangkat keras

Perancangan perangkat keras menjelaskan bagaimana tata letak perangkat keras pada sistem. Tata letak sistem akan menjadi acuan saat implemetasi dari sistem. Tata letak sistem dijelaskan dalam bentuk diagram blok. Diagram blok pada sistem dijelaskan pada gambar 3.2.



Gambar 3.2 Diagram Blok Sistem

penjelasan dari diagram blok pada gambar 3.2 adalah sebagai berikut :

1. *Pulse sensor* akan mengirimkan hasil perhitungan denyut nadi pada arduino. Sensor denyut nadi menggunakan ujung jari sebagai media pengukuran.
2. *Load sensor* akan mengirimkan hasil perhitungan berat badan pada arduino. *Load sensor* berfungsi seperti timbangan digital di mana pengguna berdiri di atasnya.
3. Arduino nano sebagai pusat pemrosesan data. Disinilah data yang diperoleh akan diproses untuk mendapatkan hasil yang diinginkan. Proses yang dilakukan dalam arduino nano meliputi :
 - a. Pembacaan denyut nadi
 - b. Pembacaan berat badan
4. HX711 sebagai ADC sekaligus *amplifier* untuk mendukung pembacaan dari *load sensor* sehingga dapat diproses oleh arduino nano
5. Modul *bluetooth* HC-05 sebagai penghubung antar arduino nano dan perangkat berbasis OS android.
6. perangkat berbasis OS android sebagai masukan dan keluaran hasil dari data di arduino nano. Disinilah pengguna mengendalikan jalannya sistem secara keseluruhan. Proses yang dilakukan dalam perangkat berbasis OS meliputi :
 - a. Perhitungan VO_{2max} menggunakan *non-exercise prediction method*
 - b. Klasifikasi VO_{2max}

- c. Perhitungan Denyut nadi latihan
- d. Perhitungan IMT
- e. Klasifikasi IMT
- f. Proses pemberian saran terkait peningkatan kebugaran kardiorespirasi
- g. Proses pemberian saran terkait capaian berat badan ideal

3.3.2 Perancangan perangkat lunak

Pada perancangan perangkat lunak ini akan menjelaskan alur dari proses yang berjalan dari sistem. Alur ini akan menjadi acuan algoritma yang akan diimplementasikan ke dalam sistem. Sistem ini bergerak ketika mendapat masukan dari pengguna. Pengguna akan memasukan pilihan sesuai kebutuhan dan sistem akan mencocokkan pilihan tersebut kemudian menjalankan fungsi sesuai pilihan yang dimasukan oleh pengguna. Pilihan yang tersedia mempunyai macam – macam fungsi yang berbeda yaitu sebagai berikut :

1. Fungsi hitung denyut nadi
Pada fungsi ini menjalankan perhitungan denyut nadi pengguna. Pengguna diharuskan meletakkan ujung jari ke *pulse sensor* untuk mendapatkan denyut nadi yang dibutuhkan.
2. Fungsi hitung berat badan
Pada fungsi ini menjalankan perhitungan berat badan berdasarkan masukan dari sensor. Pengguna harus berdiri di atas *load sensor* untuk mengukur berat badan.
3. Fungsi hitung IMT
Pada fungsi ini menjalankan perhitungan indeks masa tubuh (IMT). Pengguna akan melakukan masukan berupa tinggi badan dalam satuan senti meter secara manual dan berat badan menggunakan fungsi hitung berat badan. Hasil dari IMT akan diklasifikasikan apakah pengguna mempunyai berat badan seimbang atau tidak.
4. Fungsi hitung kebugaran kardiorespirasi
Pada fungsi ini menjalankan perhitungan VO_{2max} menggunakan *non-exercise prediction method* dengan masukan berupa jenis kelamin dan umur oleh pengguna secara manual, masukan denyut nadi menggunakan fungsi hitung denyut nadi dan masukan berat badan menggunakan fungsi hitung berat badan. Hasil dari Perhitungan akan diklasifikasikan apakah pengguna tergolong bugar atau tidak serta menampilkan saran bagaimana meningkatkannya.
5. Fungsi hitung denyut nadi latihan
Pada fungsi ini menjalankan perhitungan denyut nadi latihan dengan masukan denyut nadi menggunakan fungsi hitung denyut nadi. Hasil dari Perhitungan menampilkan berapa denyut batas nadi yang aman untuk latihan beserta fungsi dari zona – zona latihan tersebut berdasarkan persentasenya.

3.4 Implementasi Sistem

Pada implementasi semua perangkat keras dan perangkat lunak dipadukan untuk membuat sistem secara keseluruhan agar dapat digunakan. Perangkat keras disusun berdasarkan diagram blok pada gambar 3.1 dan pembuatan program yang akan di *upload* ke arduino nano melalui arduino IDE dilakukan. Pembuatan aplikasi berbasis OS android dilakukan oleh MIT APP *Inventor*.

3.5 Pengujian Sistem dan Analisis Hasil

Pada tahap pengujian terbagi menjadi dua, pengujian masukan dan pengujian fungsional. Pengujian masukan dilakukan dengan cara menguji keakuratan *pulse sensor* dan *load sensor*. Pengujian *pulse sensor* dilakukan dengan cara membandingkan hasil ukur dengan hasil ukur asli. Letakan ujung jari pada pergelangan tangan dan ukur denyut nadi selama 15 detik. BPM dihitung dengan mengalikan hasil perhitungan selama 15 detik dikalikan dengan 4. Hasil ukur tersebut dicatat seberapa besar *error* yang mungkin terjadi. Pengujian *load sensor* dilakukan dengan cara membandingkan hasil ukur dari sistem dan hasil ukur dengan menggunakan timbangan digital. Hasil ukur tersebut dicatat seberapa besar *error* yang mungkin terjadi.

Pengujian fungsional berjalannya sistem *monitoring* kebugaran kardiorespirasi dilakukan dengan cara melakukan tes terhadap 5 orang pengguna yang memiliki berbagai macam variasi frekuensi olahraga, umur, denyut nadi, berat badan dan jenis kelamin. 2 dari 5 orang subjek melakukan latihan untuk meningkatkan VO_{2max} sedangkan subjek yang tersisa mengurangi intensitas olahraga yang dilakukan. Hasil dicatat dan dibandingkan dampak setelah pengujian dilakukan terhadap nilai VO_{2max} . Pada tahap ini juga diuji apakah aplikasi pada OS android bekerja dengan baik sebagaimana mestinya dan seberapa jauh modul *bluetooth* dapat mengirimkan data. Analisis yang didapat dari pengujian dilakukan untuk mengetahui faktor – faktor yang mempengaruhi gagalannya sistem ketika berjalan.

BAB 4 REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan menjelaskan kebutuhan sistem secara rinci yang tidak dijelaskan pada bab sebelumnya. Rekayasa kebutuhan terdiri dari gambaran umum sistem, kebutuhan sistem dan batasan desain sistem.

4.1 Gambaran Umum Sistem

Kebugaran kardiorespirasi berhubungan dengan kemampuan fungsi dari jantung, pembuluh darah, paru – paru, dan otot untuk menyuplai oksigen saat melakukan aktivitas fisik. Terdapat bukti bahwa memiliki kebugaran kardiorespirasi yang baik dapat menurunkan penyakit kardiovaskular seperti serangan jantung, nyeri dada dan stroke karena kebugaran kardiorespirasi meningkatkan sensitivitas insulin, transportasi glukosa, memperbaiki fungsi sistem saraf dan menurunkan denyut jantung (Lee et al., 2010). Oleh karena itu penting untuk memiliki kebugaran kardiorespirasi yang baik.

Kebugaran kardiorespirasi dapat ditingkatkan dengan latihan fisik. Latihan fisik yang dilakukan harus tepat dan tidak berlebihan sehingga terhindar dari risiko *over Training* yang mengakibatkan penurunan bukan peningkatan kemampuan dari tubuh. Kebugaran kardiorespirasi dapat diukur melalui seberapa besar maksimum oksigen yang bisa diambil oleh tubuh (VO_{2max}). Pengukuran kebugaran kardiorespirasi melalui VO_{2max} menjadi alternatif termudah untuk mengetahui seberapa bugar tubuh seseorang (Lee et al., 2010).

Sistem *monitoring* kebugaran kardiorespirasi adalah sistem yang memberitahu pengguna informasi kebugaran kardiorespirasi tubuh dan batas maksimal latihan yang dilakukan. Sistem ini menggunakan *pulse sensor* dan *load sensor*. *Pulse sensor* dan *load sensor* membantu perhitungan VO_{2max} menggunakan *non-exercise prediction method* beserta klasifikasinya dengan tambahan masukan informasi dari pengguna berupa umur dan jenis kelamin. Selain perhitungan VO_{2max} sistem juga mengukur zona latihan dimana latihan tersebut tergolong aman atau *over training* dan indeks masa tubuh untuk mengetahui ideal atau tidak perbandingan tinggi dan berat badan tubuh. Sistem ini dapat membantu pengguna dalam mengetahui kebugaran tubuh dan dapat menjadi acuan pengguna agar berlatih lebih giat untuk mencapai kebugaran kardiorespirasi yang optimal tanpa melakukan latihan yang berlebihan.

4.2 Kebutuhan Sistem

Kebutuhan sistem berisi tentang penjelasan kebutuhan – kebutuhan untuk terwujudnya sistem. Tujuan dari kebutuhan sistem adalah untuk mempermudah desain dan implementasi dari sistem.

4.2.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak adalah kebutuhan yang menjembatani programmer dengan sistem. Kebutuhan perangkat untuk terwujudnya sistem adalah sebagai berikut :

1. Arduino IDE

Arduino IDE adalah perangkat lunak bagian dari arduino nano yang menjadi tempat algoritma sistem ditulis, *diupload* dan *dicompile*. Arduino IDE juga memiliki beberapa fitur lain seperti *serial monitor* dan *serial plotter* yang berfungsi sebagai penampil hasil dari program sehingga dapat diawasi oleh pengguna. Arduino IDE menggunakan bahasa pemrograman berbasis C yang memudahkan dalam penulisan kode programnya.

2. MIT APP *Inventor*

MIT APP *Inventor* adalah aplikasi berbasis WEB dimana berfungsi sebagai pembuat aplikasi berbasis OS android. MIT APP *Inventor* memudahkan pembuatan aplikasi dikarenakan pembuatannya berbasis visual sehingga pengguna tidak perlu bersentuhan dengan bahasa program yang rumit.

4.2.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras adalah keebutuhan yang bersifat benda nyata dan harus dipenuhi dalam prakteknya. Kebutuhan perangkat geasaaras terdiri dari :

1. Arduino nano

Sebagai pusat dari implementasi algoritma sistem. Arduino nano bertugas untuk mengontrol jalannya sistem dan menjadi tempat semua perangkat keras diinstalasi.

2. *Pulse Sensor*

Sensor yang bertugas untuk mendeteksi dan merekam denyut nadi untuk kebutuhan kalkulasi program. *Pulse sensor* diletakan pada ujung jari agar dapat berfungsi. Kalibrasi dari *pulse sensor* sangat dibutuhkan untuk mendapatkan hasil yang tepat.

3. *Load Sensor* (50 Kg)

Load Sensor bertugas untuk mengukur berat badan untuk kalkulasi progam. *Load sensor* berfungsi seperti timbangan digital dimana pengguna berdiri di atasnya.

4. HX711

HX711 sebagai ADC sekaligus *amplifier* untuk mendukung pembacaan dari *load sensor* sehingga dapat diproses oleh arduino nano. HX711 menjadi tempat *load sensor* dihubungkan sebelum masuk ke arduino nano



5. Modul *Bluetooth* HC-05

Modul *bluetooth* HC-05 bertugas sebagai penghubung perangkat berbasis OS android dengan arduino nano agar dapat bertukar data.

4.2.3 Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan yang bersifat teknis dan harus dicermati dengan baik untuk keberhasilan sistem. Kebutuhan fungsional pada sistem meliputi :

1. Perhitungan VO_{2max} dan klasifikasi hasil

Setelah mendapat hasil dari *pulse sensor* dengan masukan berupa jenis kelamin dan umur oleh pengguna secara manual, masukan denyut nadi menggunakan fungsi hitung denyut nadi dan masukan berat badan menggunakan fungsi hitung berat badan maka proses perhitungan VO_{2max} dimulai. Proses perhitungan VO_{2max} menggunakan *non-exercise prediction method* di persamaan 2.3 dan 2.4 kemudian klasifikasikan dari VO_{2max} berdasarkan pada tabel 2.2 untuk perempuan dan tabel 2.3 untuk laki – laki. Fungsi dari klasifikasi adalah untuk mengetahui seberapa bugar kondisi tubuh pengguna.

2. Perhitungan IMT dan klasifikasi hasil.

Setelah mendapat masukan berat badan dari *load sensor* dan tinggi badan secara manual oleh pengguna maka proses perhitungan IMT dimulai. Proses perhitungan IMT menggunakan persamaan 2.5 dan klasifikasi dari IMT berdasarkan pada tabel 2.5. Fungsi dari klasifikasi adalah mengetahui seberapa cocok kombinasi antar berat badan dan tinggi badan.

3. Perhitungan zona latihan dan menampilkan hasil klasifikasi.

Setelah mendapat hasil dari *pulse sensor* yang digunakan serta umur masukan manual oleh pengguna dan perhitungan HR_{max} dari persamaan 2.1 maka proses zona latihan dimulai. Proses perhitungan zona latihan menggunakan persamaan 2.2 dan klasifikasikan dari zona latihan berdasarkan dari persentase terhadap perhitungan HR_{max} . Fungsi dari klasifikasi adalah untuk mengetahui zona latihan yang sebaiknya dilakukan.

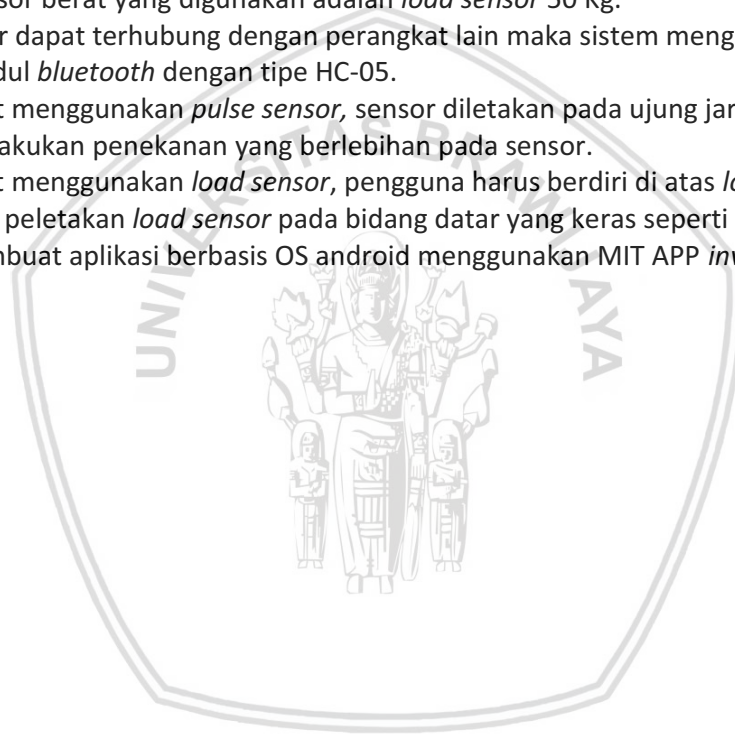
4. Pemberi masukan terkait meningkatkan kebugaran kardiorespirasi dan cara menggapai nilai IMT yang ideal

Setelah sistem mendapatkan hasil dari VO_{2max} dan IMT maka sistem dapat memberikan masukan terkait untuk meningkatkan kebugaran kardiorespirasi dan cara menggapai nilai IMT yang ideal. Saran untuk meningkat kardiorespirasi berupa latihan apa yang harus dilakukan dan batasan – batasan apa saja saat melakukannya. Saran untuk memiliki nilai IMT yang ideal berupa berat badan yang harus dicapai untuk kesesuaian dengan tinggi badan.

4.3 Batasan Desain Sistem

Saat implementasi dari sistem, sistem yang dibuat mempunyai batasan – batasan. Hal tersebut dikarenakan oleh kemampuan dari perangkat keras yang digunakan seperti *pulse sensor* yang sangat sensitif terhadap sentuhan sehingga dapat mempengaruhi hasil dari pengukuran denyut nadi serta ketelitian dari *load sensor*. Modul *bluetooth* yang digunakan memiliki jarak maksimal yaitu sekitar 10 meter yang mengakibatkan perangkat tidak dapat dikendalikan dengan jarak lebih dari kemampuan. Sistem memiliki batasan – batasan agar berjalan sesuai dengan harapan, berikut adalah batasan – batasan dari sistem :

1. Mikrokontroler yang digunakan adalah arduino nano.
2. Sensor denyut nadi yang digunakan adalah *pulse sensor*.
3. Sensor berat yang digunakan adalah *load sensor* 50 Kg.
4. Agar dapat terhubung dengan perangkat lain maka sistem menggunakan modul *bluetooth* dengan tipe HC-05.
5. Saat menggunakan *pulse sensor*, sensor diletakan pada ujung jari dan tidak melakukan penekanan yang berlebihan pada sensor.
6. Saat menggunakan *load sensor*, pengguna harus berdiri di atas *load sensor* dan peletakan *load sensor* pada bidang datar yang keras seperti lantai.
7. Pembuat aplikasi berbasis OS android menggunakan MIT APP *inventor*.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab perancangan dan implementasi menjelaskan perancangan dari sistem beserta implementasinya. Pada perancangan terdapat tiga pokok bahasan yaitu perancangan dalam perangkat keras, perangkat lunak dan algoritma jalannya sistem. Pada Implementasi berisi bagaimana sistem diterapkan dalam keadaan sebenarnya terdiri dari implementasi perangkat keras dan perangkat lunak.

5.1 Perancangan Sistem

Perancangan sistem terdiri perancangan perangkat keras dan perangkat lunak. Perancangan sistem berfungsi sebagai acuan dalam implementasi sistem. Sistem dalam penggunaannya dijelaskan pada gambar 5.1.



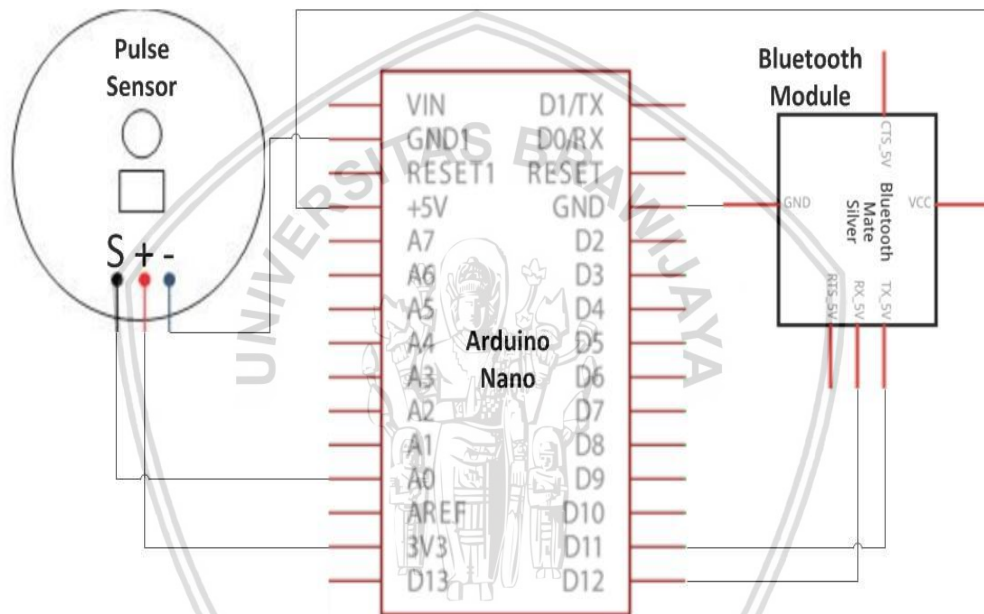
Gambar 5.1 Desain Sistem

Perangkat terpisah menjadi dua yaitu perangkat untuk mengukur denyut nadi yang digunakan pada ujung jari dan perangkat untuk mengukur berat badan yang digunakan dengan cara berdiri di atasnya. Masing – masing perangkat dikendalikan oleh perangkat berbasis OS android untuk melakukan pengukurannya. Untuk dapat melakukan pertukaran data masing – masing

perangkat dilengkapi dengan modul *bluetooth* dan mempunyai alamat masing – masing sebagai penanda.

5.1.1 Perancangan Perangkat Keras

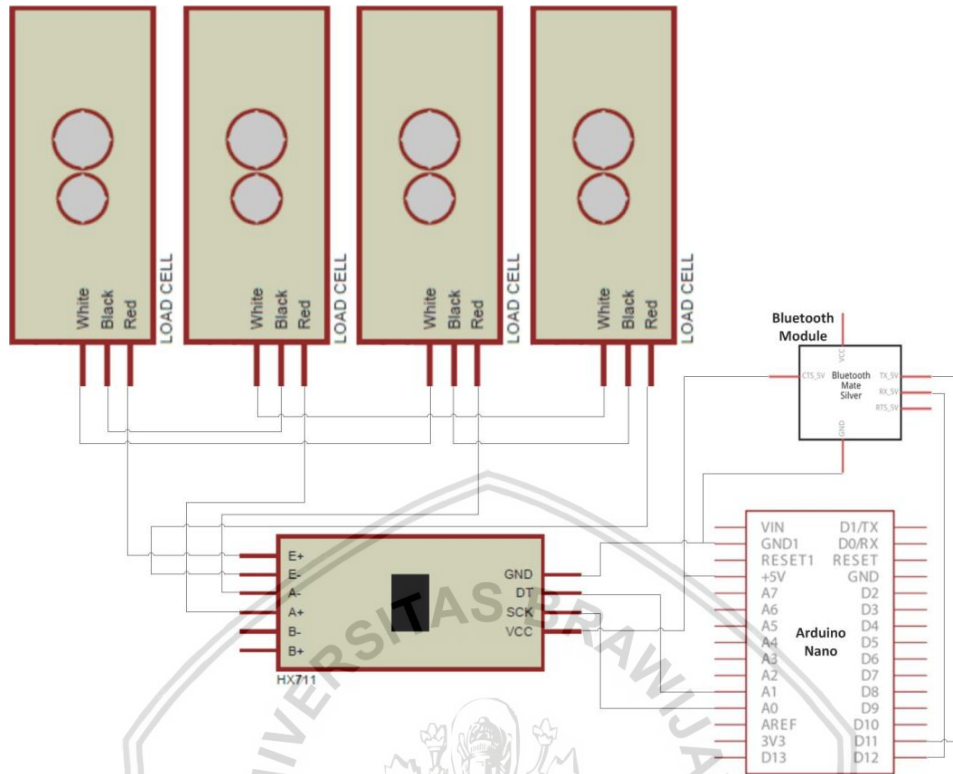
Arduino nano sebagai tempat modul *bluetooth* dan sensor untuk dapat menjalankan fungsinya yaitu pengukuran sekaligus mengirimkan data hasil pengukuran melalui modul *bluetooth* ke perangkat android. Sistem menggunakan 2 arduino nano dan 2 modul *bluetooth* untuk masing - masing *pulse sensor* dan *load sensor*. Pemasangan tersebut bertujuan agar dua perangkat dapat lebih ringkas mengingat posisi kedua perangkat tersebut sangat berjauhan yaitu di kaki dan di ujung jari. Bagaimana perangkat terhubung pada arduino nano dengan *pulse sensor* dituangkan pada gambar 5.2.



Gambar 5.2 Skematik Arduino Nano dengan *Pulse Sensor*

Pin voltase keluaran pada arduino nano memiliki dua voltase yaitu 3,3 V dan 5 V. Berdasarkan pada spesifikasi yang didapatkan dari *datasheet pulse sensor* dan modul *bluetooth* HC-05 masing – masing memiliki voltase operasi dengan jarak yang masih berada dalam jangkauan arduino nano. Hal tersebut menyebabkan fleksibilitas dalam penaruhan sumber tenaga untuk sensor dan modul. Selanjutnya pin – pin untuk pengoperasian sensor dan modul diletakan pada arduino nano yang akan di program oleh arduino IDE untuk berjalan sesuai kebutuhan dari sistem. Pin *signal* pada *pulse sensor* diletakan pada pin analog arduino nano karena hasil pengukuran dari *pulse sensor* berupa sinyal analog berbeda dengan pin pada mpdul *bluetooth* yang memberikan masukan/keluaran dalam bentuk digital.





Gambar 5.3 Skematik Arduino Nano dengan Load Sensor

Pada gambar 5.3 *load sensor* 50 Kg disusun menggunakan prinsip *wheatstone bridge* untuk mendapatkan pengukuran yang akurat dan dapat mengukur lebih dari 50 Kg. Kemudian *load sensor* dihubungkan ke HX711 yang berfungsi sebagai *analog to digital converter* untuk dapat dibaca oleh arduino nano. HX711 dan modul *bluetooth* memiliki kesamaan yaitu sumber tenaga menggunakan 5 V dan keluaran/masukan menggunakan pin digital.

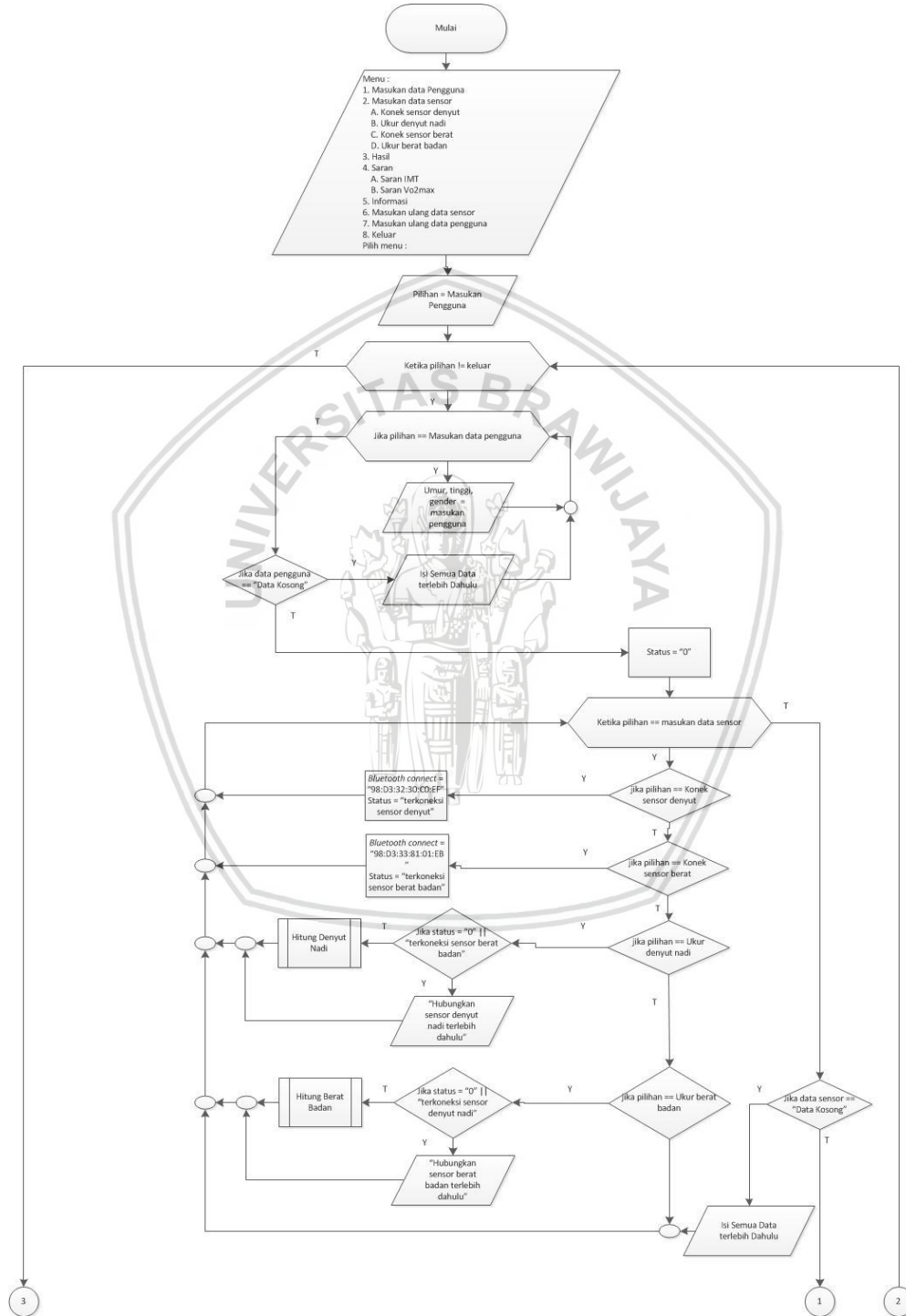
5.1.2 Perancangan Perangkat Lunak

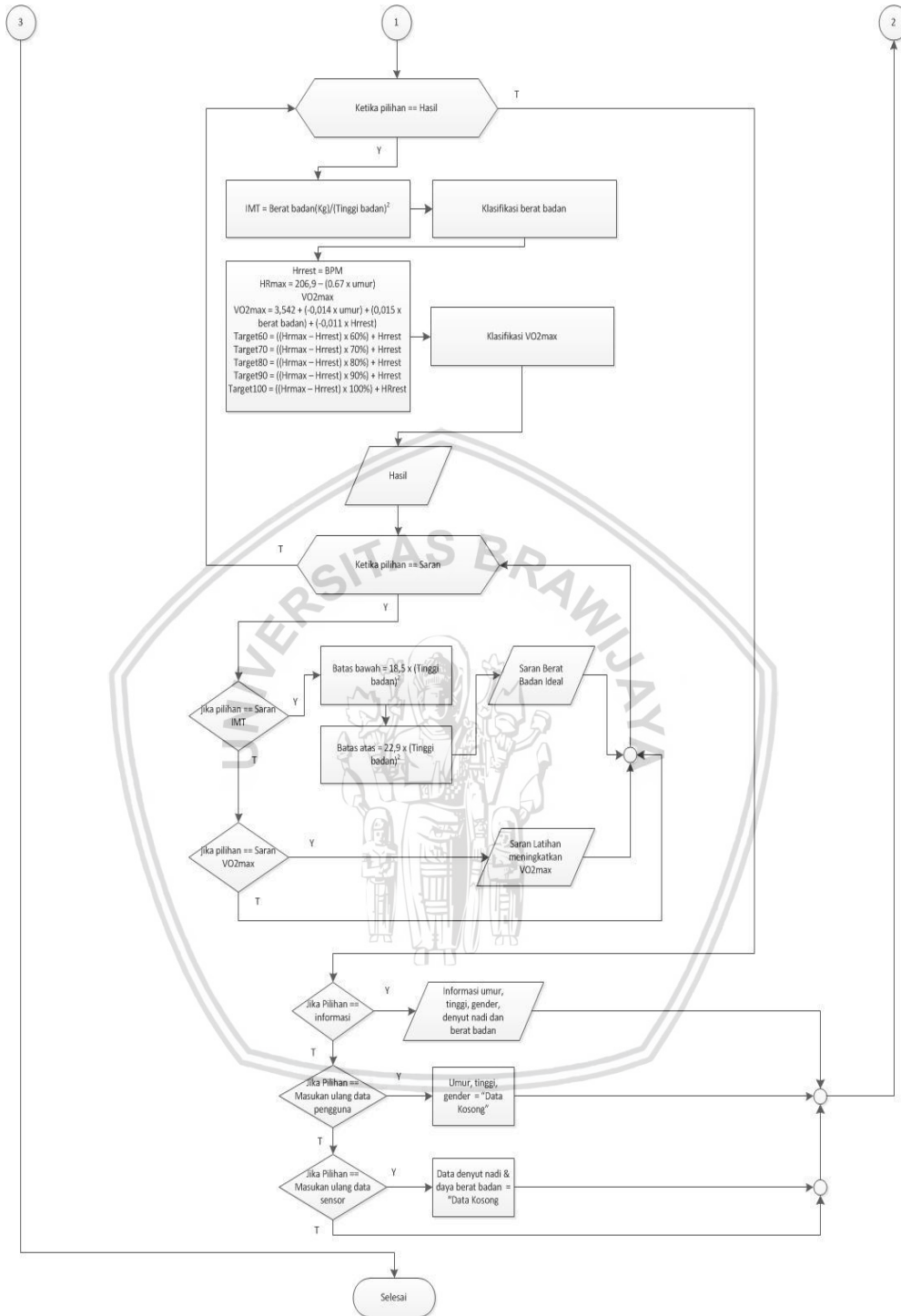
Perancangan perangkat lunak menggunakan arduino IDE sebagai tempat penulisan algoritma dalam bahasa C untuk mengatur jalannya perangkat keras dan MIT APP Inventor sebagai tempat pembuatan aplikasi berbasis OS android yang bertugas untuk mengontrol jalannya sistem. Dalam arduino IDE menggunakan pustaka *software serial* untuk kebutuhan berjalannya modul *bluetooth* HC-05 dan pustaka HX711 untuk kebutuhan berjalannya HX711. MIT APP Inventor tidak perlu menggunakan bahasa kompleks untuk mengoperasikannya karena MIT APP Inventor mempunyai konsep *drag and drop* dalam bentuk visual yang memudahkan dalam implementasi algoritma yang dibutuhkan dalam sistem.

Perancangan perangkat lunak berisi bagaimana algoritma dalam sistem berjalan. Algoritma digambarkan menggunakan *flowchart*. Perancangan perangkat lunak pada sistem terbagi menjadi dua yaitu perancangan perangkat lunak pada arduino dan pada MIT APP Inventor. Hasil dari perancangan akan diterapkan pada saat implementasi sistem.

5.1.2.1 Perancangan Perangkat Lunak Pada MIT APP Inventor

Algoritma pada aplikasi berbasis OS android yang dibuat menggunakan MIT APP Inventor berperan sebagai pemberi masukan data dari pengguna sekaligus pengontrol jalannya sistem. Alur kerja dari aplikasi dijelaskan pada gambar 5.4.



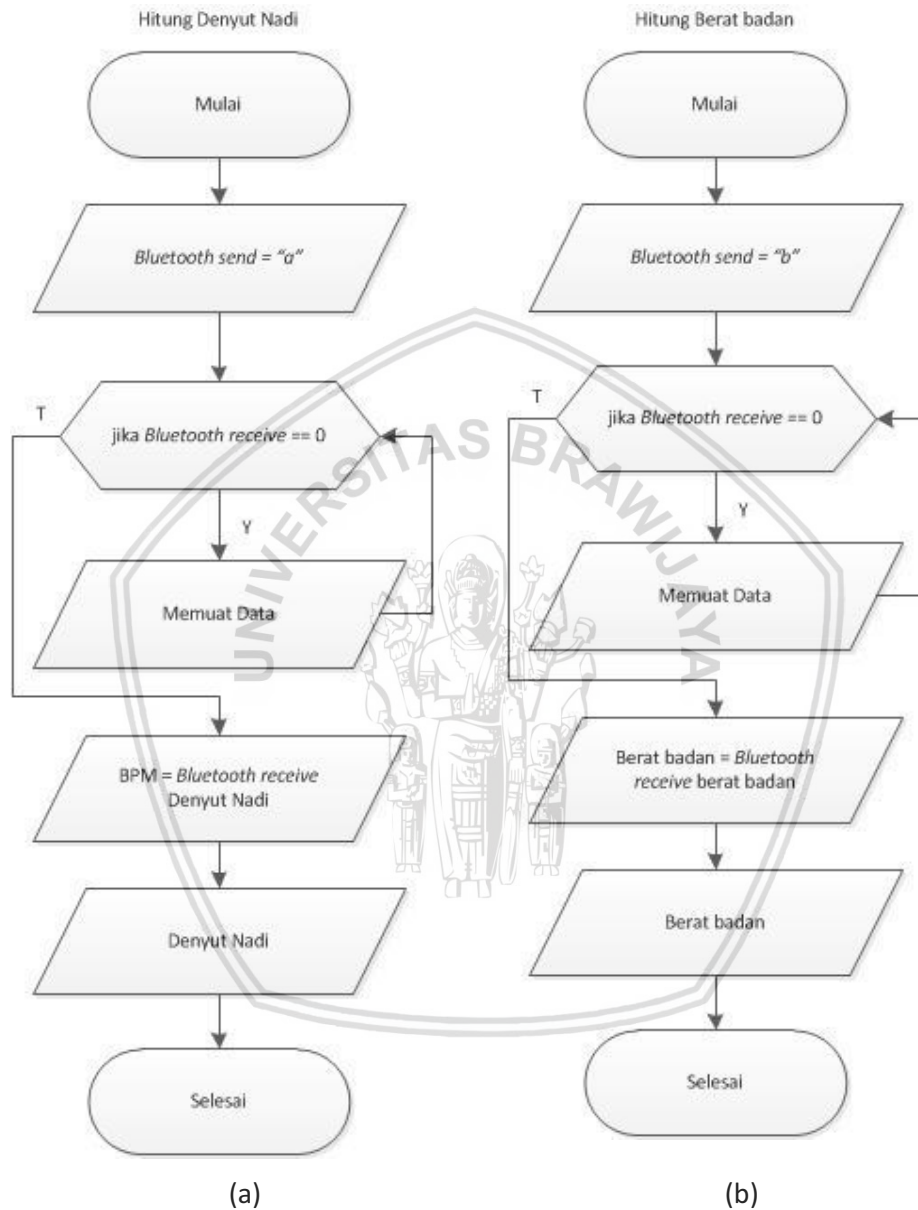


Gambar 5.4 Alur Kerja Pada Aplikasi

Masukan dari pengguna terbagi menjadi dua yaitu masukan informasi yang meliputi tinggi, umur, jenis kelamin dan masukan informasi yang melibatkan sensor. Masukan dari sensor akan memberikan aplikasi informasi denyut nadi dan berat badan. Untuk mendapatkan data sensor pengguna harus terhubung ke salah



satu sensor secara bergantian barulah dapat mendapatkan hasil dari pengukuran sensor. Dalam menjalankan sensor, aplikasi mempunyai fungsi yang berbeda – berbeda pada setiap sensor. Gambar 5.5 menjelaskan fungsi untuk menjalankan *pulse sensor* dan *load sensor*.



Gambar 5.5 Fungsi Pada Aplikasi

Kedua sensor memiliki fungsi yang hampir mirip yaitu mengirim teks ke arduino nano sebagai penanda untuk menjalankan pengukuran. Teks yang dikirim fungsi ukur denyut nadi adalah “a” sedangkan fungsi ukur berat badan adalah “b”. Selanjutnya aplikasi akan menunggu hasil pengukuran dari sensor, jika hasil pengukuran selesai maka pengguna dapat melanjutkan ke proses selanjutnya untuk melihat hasil dari perhitungan VO_{2max} dan IMT serta klasifikasinya.

Untuk menampilkan hasil dari masukan pada aplikasi terdapat beberapa perhitungan yang dilakukan. Perhitungan yang dilakukan berbeda - beda untuk setiap hasil yang diinginkan. Perhitungan yang dilakukan pada aplikasi yaitu sebagai berikut :

1. Perhitungan dan klasifikasi IMT
Perhitungan IMT menggunakan persamaan 2.5 dan klasifikasi berdasarkan pada tabel 2.5.
2. Perhitungan dan klasifikasi VO_{2max}
Perhitungan VO_{2max} menggunakan *non-exercise prediction method* di persamaan 2.3 dan persamaan 2.4 kemudian klasifikasi berdasarkan pada tabel 2.2 untuk perempuan dan tabel 2.3 untuk laki – laki.
3. Perhitungan denyut nadi latihan
Perhitungan denyut nadi latihan menggunakan persamaan 2.2. Denyut nadi latihan yang diukur yaitu 60%, 70%, 80%, 90% dan 100%.

pengguna dapat memunculkan saran terkait bagaimana mencapai nilai yang ideal. Saran yang dapat diberikan kepada pengguna meliputi sebagai berikut :

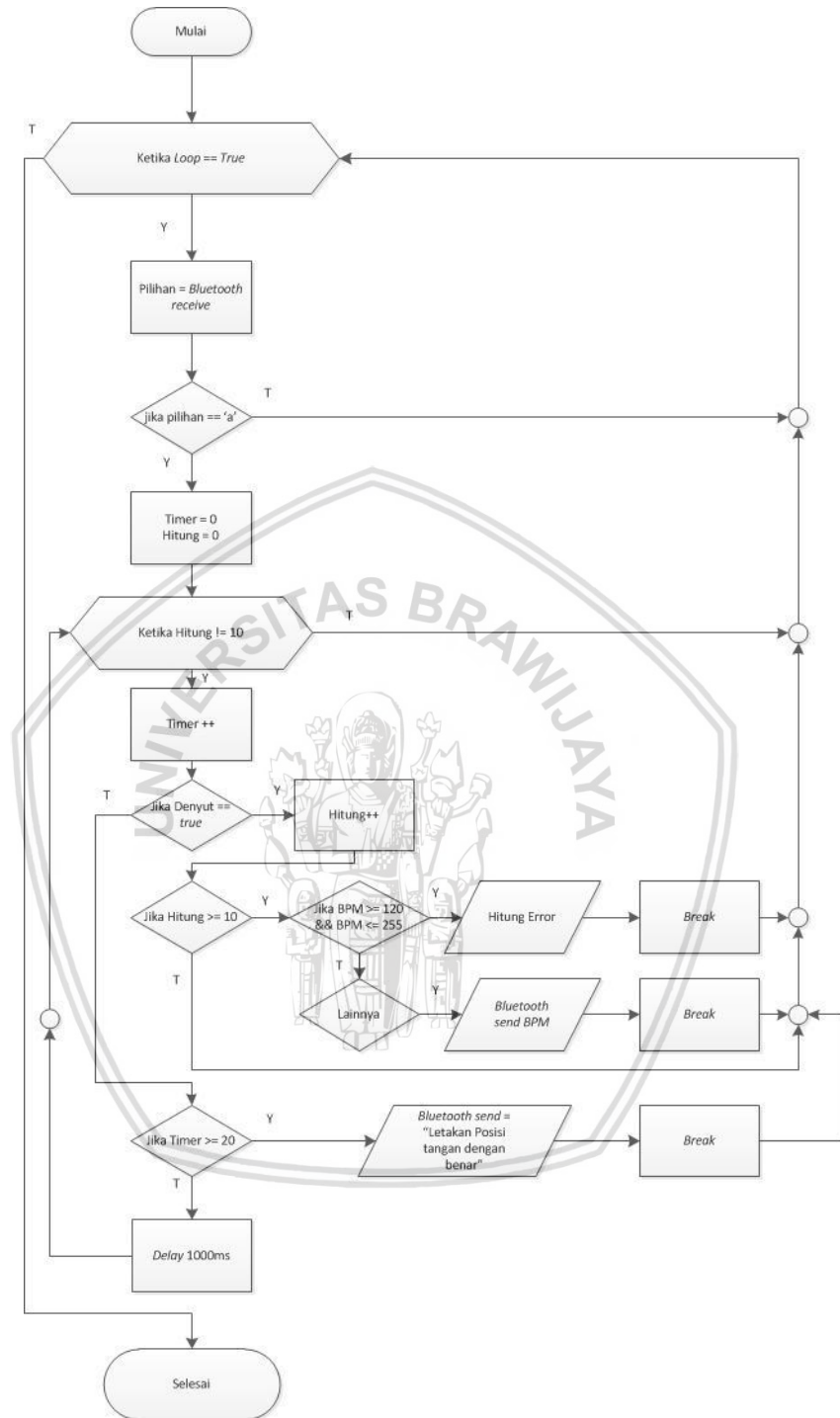
1. Saran VO_{2max}
Saran VO_{2max} menampilkan latihan – latihan efektif yang dapat dilakukan oleh pengguna untuk mencapai nilai VO_{2max} yang baik
2. Saran berat badan
Saran berat badan menampilkan rentang berat badan yang harus dicapai oleh pengguna untuk mendapatkan status berat badan ideal. Berat badan yang harus dicapai oleh pengguna dihitung dengan memodifikasi persamaan 2.5 sehingga muncul persamaan baru yang ditulis pada persamaan 5.1. Nilai a pada persamaan mengarah kepada capaian berat badan ideal yang diinginkan, dalam sistem ini menggunakan 18,5 dan 22,9 dikarenakan menurut tabel 2.5 rentang nilai diantara kedua nilai tersebut adalah berat badan ideal.

$$\text{Target} = a \times (\text{TinggiBadan(m)})^2 \quad (5.1)$$

Pengguna dapat kapan saja mengulang kembali informasi yang sudah dimasukan dengan cara memilih “masukan ulang data pengguna” jika ingin mengulang kembali informasi pengguna atau memilih “masukan ulang data sensor” jika hanya ingin mengulang kembali data sensor. Untuk menampilkan data yang sudah masuk maka pengguna dapat memilih pilihan “informasi” dan jika sudah selesai melakukan aktivitas pengguna dapat memilih “keluar” untuk menghentikan jalannya aplikasi.

5.1.2.2 Perancangan Perangkat Lunak Pada Arduino Nano

Arduino nano terpasang pada masing – masing sensor dan berjalan terpisah. Arduino nano akan menunggu perintah dari pengguna melalui aplikasi untuk menjalankan perangkat *pulse sensor* dan *load sensor*. Pada gambar 5.6 dijelaskan alur kerja dari arduino nano yang terpasang dengan *pulse sensor*.



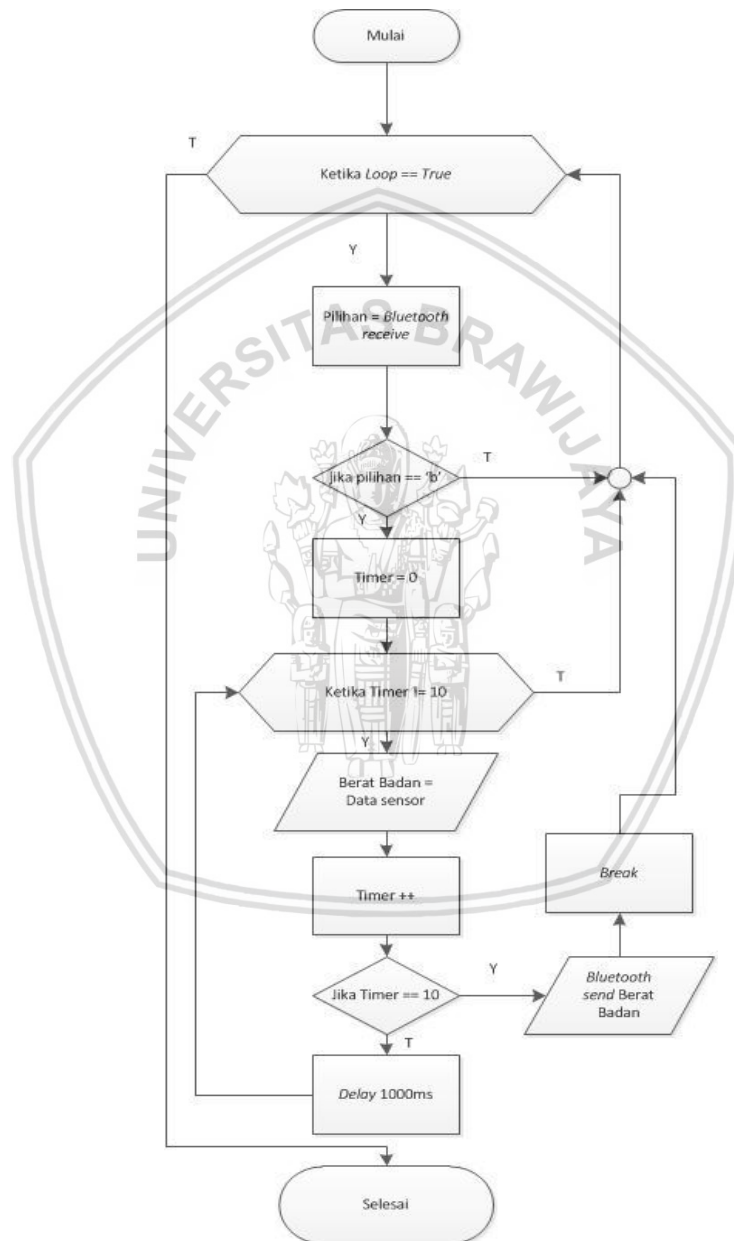
Gambar 5.6 Arduino dengan *Pulse Sensor*

Ketika perangkat mendapatkan perintah dari aplikasi ditandai dengan menerimanya pesan berupa "a" maka arduino nano akan mulai menjalankan *pulse sensor*. Data dari *pulse sensor* akan diproses, jika arduino nano berhasil mendapatkan 10 data dalam kurun waktu 20 detik barulah data akan dikirim ke



perangkat android melalui modul *bluetooth*. Jika data yang diterima mendapatkan hasil lebih dari 120 BPM maka data akan dianggap *error* kemudian pesan *error* akan dikirimkan ke perangkat android. Jika sensor tidak mendapatkan 10 data dalam waktu 20 detik maka akan dikirimkan ke perangkat android pesan “Letakan posisi tangan dengan benar”.

Pada gambar 5.7 menjelaskan alur kerja dari arduino nano yang terpasang dengan *load sensor*.



Gambar 5.7 Arduino dengan *load Sensor*

Ketika perangkat mendapatkan perintah dari aplikasi ditandai dengan menerimanya pesan berupa “b” maka arduino nano akan mulai menjalankan *load*

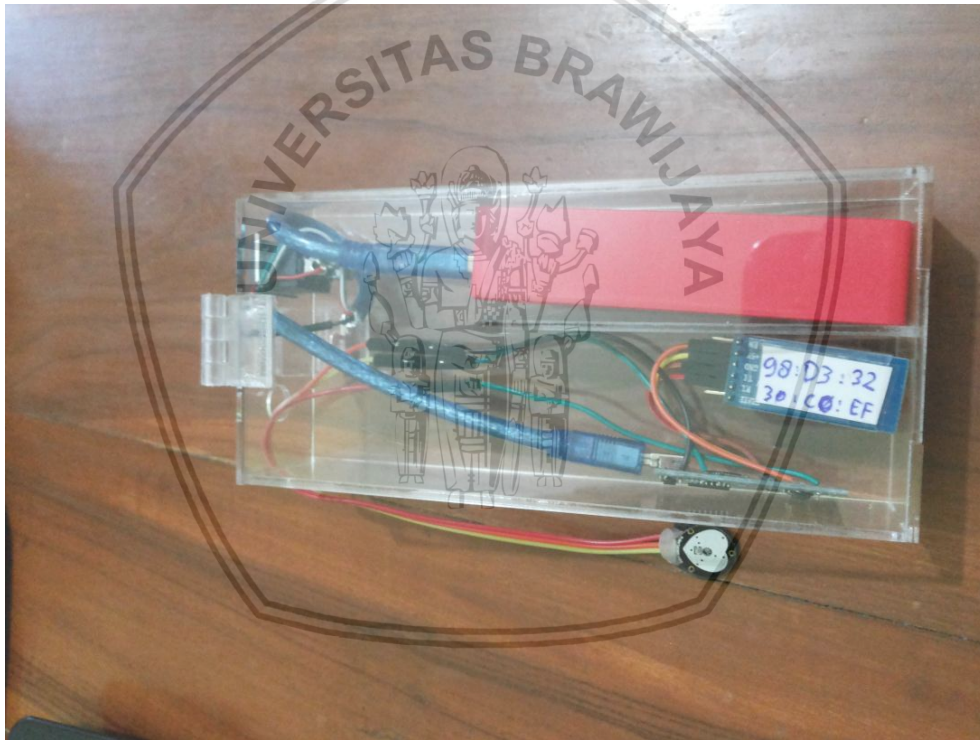
sensor. Data dari *load sensor* akan diproses, arduino akan menunggu 10 detik untuk mendapatkan hasil yang stabil dari data kemudian data akan dikirim ke perangkat android melalui modul *bluetooth*.

5.2 Implementasi Sistem

Implementasi sistem membahas tentang bagaimana sistem diimplementasi sesuai dengan perancangan yang sudah dibuat. Implementasi sistem terbagi menjadi dua bagian yaitu implementasi perangkat keras dan perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan bagaimana 2 arduino nano dan perangkat pendukung lainnya masing - masing terhubung. Pada gambar 5.8 menjelaskan bagaimana arduino nano dengan *pulse sensor*.



Gambar 5.8 Arduino nano dengan Pulse Sensor

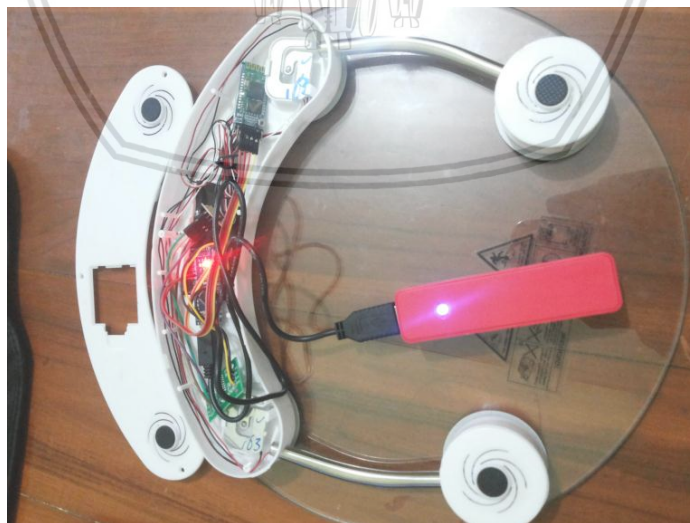
Pada gambar 5.8 perangkat diletakan dalam kubus terbuat dari acrylic. Perangkat didukung dengan *power bank* sebagai sumber tenaga untuk menjalankan arduino nano, modul *bluetooth* dan *pulse sensor*. *Pulse sensor* diletakan di luar kubus karena *pulse sensor* akan langsung berinteraksi dengan pengguna. Pada tempat dimana perangkat diletakan juga terdapat *switch* yang berfungsi untuk menyalakan *power bank* agar semua perangkat didalamnya aktif. Pada gambar 5.9 menjelaskan bagaimana arduino nano yang terhubung dengan *pulse sensor* digunakan.



Gambar 5.9 Penggunaan Pulse Sensor

Untuk memulai menggunakan perangkat, pengguna harus menyalakannya dengan cara menekan *switch* yang terletak di bagian atas perangkat. *Pulse sensor* digunakan dengan cara meletakkan ujung dari sensor ke ibu jari tangan pengguna.

Pada gambar 5.10 menjelaskan bagaimana arduino nano dengan *load sensor* dirangkai pada tempat timbangan yang sudah di modifikasi sehingga hanya *load sensor* pada timbangan yang dapat digunakan.



Gambar 5.10 Arduino Nano dengan Load Sensor

Perangkat arduino nano, modul *bluetooth* dan HX711 dimasukan ke dalam tempat yang sama sehingga bisa ditutup dengan rapi menyatu dengan timbangan digital.

Load sensor tersebar di 4 sisi timbangan dan dirangkai menggunakan prinsip *wheatstone bridge* kemudian dihubungkan ke HX711. Perangkat menggunakan *power bank* sebagai sumber tenaga untuk menjalankannya. Pengguna harus berdiri di atas perangkat untuk mendapatkan hasil pengukuran berat badan, gambar 5.11 menjelaskan bagaimana cara menggunakannya.



Gambar 5.11 Cara Menggunakan *Load Sensor*

Saat pengukuran perangkat harus diletakan di atas permukaan yang rata dan keras seperti lantai keramik karena akan mempengaruhi pengukuran sensor. Pengguna harus bertelanjang kaki saat pengukuran untuk mendapatkan pengukuran akurat dari berat badan dan keadaan kaki pengguna harus dalam keadaan kering untuk menghindari terpeleset dikarenakan alas dari timbangan terbuat dari kaca sehingga sangat rawan jika digunakan saat kaki tidak dalam keadaan kering.

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan bagaimana praktek dari diagram alir yang dijelaskan pada bab perancangan perangkat lunak. Implementasi perangkat lunak terbagi menjadi dua yaitu implementasi perangkat lunak pada arduino nano menggunakan arduino IDE dan implementasi perangkat lunak untuk pembuatan aplikasi berbasis OS android menggunakan MIT APP Inventor.

5.2.2.1 Implementasi Perangkat Lunak Pada Arduino Nano

Implementasi perangkat lunak pada arduino nano menggunakan arduino IDE sebagai perantara pembuatan program untuk menjalankan semua perangkat keras pada sistem. Arduino IDE menggunakan bahasa C yang mudah dipahami sehingga tidak mempersulit dalam pembuatan programnya. Implementasi perangkat lunak pada arduino nano sebagai berikut :

1. Implementasi perangkat lunak pada arduino nano dengan *pulse sensor*
 Arduino nano membutuhkan pustaka yang mendukung dalam menjalankan perangkatnya. Pada gambar 5.12 menjelaskan pustaka yang digunakan.

```
1 #include <SoftwareSerial.h>
```

Gambar 5.12 Pustaka Pada Arduino Nano dengan *Pulse Sensor*

Modul *bluetooth* membutuhkan pustaka software serial untuk dapat berjalan sebagaimana fungsinya. Selain pustaka yang digunakan Inisialisasi pin pada arduino dimana perangkat terpasang juga harus dimasukkan dalam kode program. Inisialisasi ini berguna untuk penggunaan lebih lanjut perangkat sensor dan modul dalam program agar rancangan sistem dapat diimplementasikan. Inisialisasi letak perangkat ke arduino nano dijelaskan pada gambar 5.13.

```
1 SoftwareSerial bt(3, 2); // TX, RX
2 int pulsePin = A0;
```

Gambar 5.13 Inisialisasi Letak Perangkat Pada Arduino Nano dengan Pulse Sensor

Pada arduino terdapat bagian dimana semua pengaturan awal dilakukan sebelum menjalankan program utama. Pada gambar 5.14 menjelaskan *setup* dalam program.

```
1 Serial.begin(115200);
2 bt.begin(9600);
3 interruptSetup();
```

Gambar 5.14 *Setup* program Pada Arduino Nano dengan Pulse Sensor

Serial.begin dan *bt.begin* melakukan pengaturan awal untuk *baudrate* dari program, *baudrate* sendiri mempunyai makna kecepatan data yang dikirim. Pengaturan *baudrate* menyesuaikan kebutuhan dalam program. *InterruptSetup* berfungsi untuk mengaktifkan *interrupt* yang berjalan di luar program utama untuk mendeteksi denyut nadi (BPM). Pada gambar 5.15 menjelaskan kode program untuk mendeteksi denyut nadi.

```
1 volatile int rate[10];
2 volatile unsigned long sampleCounter = 0;
3 volatile unsigned long lastBeatTime = 0;
4 volatile int P = 512;
5 volatile int T = 512;
6 volatile int thresh = 525;
7 volatile int amp = 100;
8 volatile boolean firstBeat = true;
9 volatile boolean secondBeat = false;
10 void interruptSetup(){
11     TCCR2A = 0x02;
12     TCCR2B = 0x06;
13     OCR2A = 0x7C;
14     TIMSK2 = 0x02;
15     sei();
16 }
17
```

```
18 ISR(TIMER2_COMPA_vect){
19   cli();
20   Signal = analogRead(pulsePin);
21   sampleCounter += 2;
22   int N = sampleCounter - lastBeatTime;
23
24   if(Signal < thresh && N > (IBI/5)*3){
25     if (Signal < T){
26       T = Signal;
27     }
28   }
29   if(Signal > thresh && Signal > P){
30     P = Signal;
31   }
32
33   if (N > 250){ //
34     if ( (Signal > thresh) && (Pulse == false) && (N >
35 (IBI/5)*3) ){
36       Pulse = true;
37       IBI = sampleCounter - lastBeatTime;
38       lastBeatTime = sampleCounter;
39       if(secondBeat){
40         secondBeat = false;
41         for(int i=0; i<=9; i++){
42           rate[i] = IBI;
43         }
44       }
45       if(firstBeat){
46         firstBeat = false;
47         secondBeat = true;
48         sei();
49         return;
50       }
51       word runningTotal = 0;
52       for(int i=0; i<=8; i++){
53         rate[i] = rate[i+1];
54         runningTotal += rate[i];
55       }
56       rate[9] = IBI;
57       runningTotal += rate[9];
58       runningTotal /= 10;
59       BPM = 60000/runningTotal;
60       QS = true;
61     }
62   }
63
64   if (Signal < thresh && Pulse == true){
65     Pulse = false;
66     amp = P - T;
67     thresh = amp/2 + T;
68     P = thresh;
69     T = thresh;
70   }
71
72   if (N > 2500){
73     thresh = 512;
74     P = 512;
75     T = 512;
```

```

76     lastBeatTime = sampleCounter;
77     firstBeat = true;
78     secondBeat = false;
79   }
80   sei();

```

Gambar 5.15 Mendeteksi Denyut Nadi Pada Arduino Nano dengan *Pulse Sensor*

Program pada gambar 5.15, menghitung BPM dengan cara menemukan denyut nadi kemudian menghitung rata – rata waktu antara gelombang denyut nadi kemudian membaginya dengan 60 detik. Denyut nadi ditemukan berdasarkan perubahan pada pembacaan sensor kemudian menandai perubahan tersebut dengan P dan T sehingga mudah untuk melakukan proses selanjutnya.

Untuk menjalankan program utama dibutuhkan variabel – variabel pendukung yang dimasukkan dalam kode program. Pada gambar 5.16 menjelaskan variabel – variabel yang digunakan dalam program.

```

1   String input ;
2   volatile int BPM;
3   volatile int Signal;
4   volatile int IBI = 600;
5   volatile boolean Pulse = false;
6   volatile boolean QS = false;

```

Gambar 5.16 Variabel Pada Program Arduino Nano dengan *Pulse Sensor*

Variabel volatile membantu proses *interrupt* dikarenakan variabel tersebut akan terus berubah ketika ada bagian program menggunakannya karena variabel tersebut langsung ditujukan ke *memory* tidak melalui register terlebih dahulu sehingga dapat lebih fleksibel terganti jika bagian dari *interrupt* mengubah nilai variabel tersebut. Variabel input digunakan sebagai penyimpan data hasil dari pengiriman melalui modul *bluetooth* oleh aplikasi.

Arduino nano yang berfungsi sebagai pusat proses data hanya akan berjalan ketika mendapat masukan perintah dari pengguna. Pada gambar 5.17 menunjukkan kode program utama sistem.

```

1   void loop() {
2     while (bt.available()) {
3       delay(10);
4       char data = bt.read();
5       input += data;
6     }
7     if ( input.length() > 0 ) {
8       if (input == "a") {
9         int hitung = 0;
10        int timer = 0;
11      }
12      while (hitung != 10) {
13        timer += 1;
14        if (QS == true) {
15          hitung += 1;
16          QS = false;

```



```

17     }
18
19     if (hitung >= 10) {
20         if (BPM >= 120 && BPM <= 255) {
21             bt.print("Perhitungan Error, Ulang kembali");
22         }
23         else {
24             bt.print(BPM);
25         }
26         break;
27     }
28
29     if (timer >= 20) {
30         bt.print("Letakan Posisi tangan dengan benar");
31     }
32     break;
33 }
34 delay(1000);
35 }
36 }
37 }
38 input = "";
39 }

```

Gambar 5.17 Program Utama Pada Arduino Nano dengan Pulse Sensor

Masukan perintah dari pengguna ditandai dengan dikirimkannya "a" ke arduino nano, setelah mendapatkan masukan tersebut barulah program akan mengeksekusi perhitungan denyut nadi untuk dikirimkan ke aplikasi hasilnya. Data dari *pulse sensor* akan diproses, denyut nadi ditemukan ditandai dengan variabel QS yang menjadi *true*, jika arduino nano berhasil mendapatkan 10 data dalam kurun waktu 20 detik barulah data akan dikirim ke perangkat android melalui modul *bluetooth*. Jika data yang diterima mendapatkan hasil lebih dari 120 BPM maka data akan dianggap *error* kemudian pesan *error* akan dikirimkan ke perangkat android. Jika sensor tidak mendapatkan 10 data dalam waktu 20 detik maka akan dikirimkan ke perangkat android pesan "Letakan posisi tangan dengan benar". Proses tersebut dilakukan dalam *loop while* dan *loop* akan berakhir ketika perintah *break* dijalankan.

2. Implementasi perangkat lunak pada arduino nano dengan *load sensor*

Arduino nano membutuhkan pustaka yang mendukung dalam menjalankan perangkatnya, perangkat tersebut yaitu modul *bluetooth* dan HX711. Pada gambar 5.18 menjelaskan pustaka yang digunakan.

```

1 #include <SoftwareSerial.h>
2 #include <HX711.h>

```

Gambar 5.18 Pustaka Pada Arduino Nano dengan Load Sensor

Pustaka yang digunakan akan mempermudah pembuatan program untuk menjalankan perangkat yang bersangkutan. Selain pustaka yang digunakan Inisialisasi pin pada arduino dimana perangkat terpasang juga harus dimasukan dalam kode program. Inisialisasi ini berguna untuk penggunaan lebih lanjut perangkat sensor dan modul dalam program agar rancangan

sistem dapat diimplementasikan. Inisialisasi letak perangkat ke arduino nano dijelaskan pada gambar 5.19.

```
1 SoftwareSerial bt(2, 3);
2 HX711 scale;
```

Gambar 5.19 Inisialisasi Letak Perangkat Pada Arduino Nano dengan Load Sensor

Pada arduino terdapat bagian dimana semua pengaturan awal dilakukan sebelum menjalankan program utama. Pada gambar 5.20 menjelaskan *setup* dalam program.

```
1 Serial.begin(115200);
2 bt.begin(9600);
3 scale.begin(A0, A1);
4 scale.set_scale(1722603.0/70.4);
5 scale.tare();
```

Gambar 5.20 Setup program Pada Arduino Nano dengan Load Sensor

`Serial.begin` dan `bt.begin` melakukan pengaturan awal untuk *baudrate* dari program, *baudrate* sendiri mempunyai makna kecepatan data yang dikirim. Pengaturan *baudrate* menyesuaikan kebutuhan dalam program. *Scale begin* menunjukkan letak dari HX711 terpasang pada arduino nano. *Set scale* dan *scale tare* berfungsi untuk kalibrasi dari sensor untuk mendapatkan pembacaan yang akurat. Nilai dari 70,4 Kg pada *set scale* didapat dari pengukuran langsung ke timbangan digital, kemudian dengan berat badan yang sama melakukan pengukuran ke perangkat dalam keadaan parameter *set scale* kosong nilai dari pembacaan tersebut dicatat sehingga menghasilkan nilai 1722603,0.

Arduino nano yang berfungsi sebagai pusat proses data hanya akan berjalan ketika mendapat masukan perintah dari pengguna. Pada gambar 5.21 menunjukkan kode program utama sistem.

```
1 void loop() {
2   while (bt.available()) {
3     delay(10);
4     char data = bt.read();
5     input += data;
6   }
7
8   if ( input.length() > 0 ) {
9     if (input == "b") {
10      int timer = 0 ;
11      while (timer != 10) {
12        timer += 1;
13
14        if (timer == 10) {
15          bt.print(scale.get_units(10), 1);
16
17          scale.power_down();
18          scale.power_up();
19
20          break;
```

```

21     }
22
23
24     scale.power_down();
25     delay(1000);
26     scale.power_up();
27     }
28 }
29 }
30 input = "";
31 }

```

Gambar 5.21 Program Utama Pada Arduino Nano dengan *Load Sensor*

Masukan perintah dari pengguna ditandai dengan dikirimkannya “b” ke arduino nano, setelah mendapatkan masukan tersebut barulah program akan mengeksekusi perhitungan berat badan untuk dikirimkan ke aplikasi hasilnya. Data dari *load sensor* akan diproses, arduino akan menunggu 10 detik dalam *loop while* untuk mendapatkan hasil yang stabil dari data kemudian data akan dikirim ke perangkat android melalui modul *bluetooth*. Data yang dikirim ke perangkat merupakan hasil rata – rata dari 10 perhitungan untuk mendapat hasil yang maksimal ditandai dengan kode program `scale.get_units(10)`.

5.2.2.2 Implementasi Perangkat Lunak Pada MIT APP Inventor

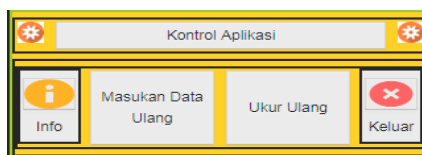
Implementasi perangkat lunak pada MIT APP Inventor menggunakan aplikasi berbasis WEB yang dapat diakses pada ai2.appinventor.mit.edu. MIT APP inventor terbagi menjadi 2 dalam pembuatan aplikasi yaitu :

1. Bagian *designer*

Designer berfungsi sebagai pembuat antar muka yang digunakan pengguna untuk mengontrol jalannya aplikasi. Pembuatan antarmuka terdiri dari beberapa komponen yang digabungkan untuk mencapai fungsi yang diinginkan. Komponen penyusun terpecah menjadi beberapa *layout* yang menjadi tempat dimana komponen diletakan. *Layout* yang digunakan dalam aplikasi terdiri dari :

a. Panel kontrol

Layout “panel kontrol” berfungsi sebagai pengontrol aplikasi secara keseluruhan. *Layout* “panel kontrol” memiliki fungsi yaitu menampilkan informasi, mengatur ulang data masukan pengguna, mengatur ulang data sensor dan menghentikan jalannya aplikasi. *Layout* pada “panel kontrol” dijelaskan pada gambar 5.22.



Gambar 5.22 Panel Kontrol

Dalam *layout* “panel kontrol” terdapat komponen penyusun yang dijelaskan pada tabel 5.1.

Tabel 5.1 Komponen Penyusun Panel Kontrol

Nama Komponen	Jenis Komponen	Fungsi
Kontrol aplikasi	Button	Menyembunyikan/menampilkan <i>layout</i> kontrol
Info	Button	Menyembunyikan/menampilkan informasi
Masukan data ulang	Button	Mengatur ulang kembali masukan pengguna kemudian mengembalikan pengguna ke proses memasukan informasi
Ukur ulang	Button	Mengatur ulang kembali masukan dari sensor kemudian mengembalikan pengguna ke proses memasukan data sensor
Keluar	Button	Menghentikan jalannya aplikasi

b. Informasi

Layout "informasi" memuat hasil masukan data pengguna dan data dari sensor. Isi dari *layout* "Informasi" meliputi jenis kelamin, umur, tinggi, denyut nadi dan berat badan dari pengguna. *Layout* pada "informasi" dijelaskan pada gambar 5.23.

The image shows a mobile application interface titled "Data Masukan". It features a header with an information icon and the title. Below the header, there are five input fields arranged in a grid:

- Jenis Kelamin- (Data Kosong)
- Umur- (Data Kosong)
- Tinggi- (Data Kosong)
- Denyut Nadi- (Data Kosong)
- Berat Badan- (Data Kosong)

Gambar 5.23 Layout Informasi

Dalam *layout* "informasi" terdapat komponen penyusun yang dijelaskan pada tabel 5.2.

Tabel 5.2 Komponen Penyusun *Layout* Informasi

Nama Komponen	Jenis Komponen	Fungsi
Hasil gender	<i>Label</i>	Menyimpan hasil dari informasi jenis kelamin pengguna
Hasil umur	<i>Label</i>	Menyimpan hasil dari informasi umur pengguna
Hasil tinggi	<i>Label</i>	Menyimpan hasil dari informasi tinggi badan pengguna
Hasil denyut	<i>Label</i>	Menyimpan hasil dari informasi denyut nadi pengguna
Hasil Berat	<i>Label</i>	Menyimpan hasil dari informasi berat badan pengguna

c. Masukan informasi

Layout "masukan informasi" berfungsi sebagai tempat dimana pengguna memasukkan data pribadi yang terdiri dari umur, tinggi dan jenis kelamin. *Layout* pada "masukan informasi" dijelaskan pada gambar 5.24.

Gambar 5.24 *Layout* Masukan Informasi

Informasi yang sudah dimasukan pengguna baru akan disimpan dalam aplikasi ketika pengguna memilih *button* "masukan" pada setiap bagian informasinya. Jika pengguna sudah memasukkan semua informasi yang dibutuhkan maka pengguna harus memilih "selesai" pada *layout* untuk melanjutkan ke proses selanjutnya. Dalam *layout* "masukan informasi" terdapat komponen penyusun yang dijelaskan pada tabel 5.3.

Tabel 5.3 Komponen Penyusun *Layout* Masukan Informasi

Nama Komponen	Jenis Komponen	Fungsi
Input umur	<i>Text box</i>	Tempat pengguna memasukan informasi tentang umur. Informasi yang dimasukan hanya berupa angka
Button umur	<i>Button</i>	Berfungsi untuk memasukan data umur pengguna untuk disimpan ke dalam aplikasi
Input tinggi	<i>Text box</i>	Tempat pengguna memasukan informasi tentang tinggi badan. Informasi yang dimasukan hanya berupa angka
Button tinggi	<i>Button</i>	Berfungsi untuk memasukan data tinggi badan pengguna untuk disimpan ke dalam aplikasi
Button laki	<i>Button</i>	Berfungsi untuk memasukan data jenis kelamin laki – laki untuk disimpan ke dalam aplikasi
Button perempuan	<i>Button</i>	Berfungsi untuk memasukan data jenis kelamin perempuan untuk disimpan ke dalam aplikasi
Selesai	<i>Button</i>	Berfungsi untuk menyembunyikan <i>layout</i> "masukan informasi kemudian" memunculkan <i>layout</i> masukan data sensor

d. Masukan data sensor

Layout "masukan data sensor" berfungsi sebagai tempat Di mana pengguna memasukan data denyut nadi dan berat badan. Pada bagian inilah aplikasi membutuhkan bantuan sensor untuk mendapatkan data dari denyut nadi dan berat badan dari *Layout* pada "masukan data sensor" dijelaskan pada gambar 5.25.



Gambar 5.25 *Layout Masukan Data Sensor*

Jika pengguna sudah memasukan semua informasi yang dibutuhkan maka pengguna harus memilih "hasil" pada *layout* untuk melanjutkan ke proses selanjutnya. Dalam *layout* "masukan data sensor" terdapat komponen penyusun yang dijelaskan pada tabel 5.4.

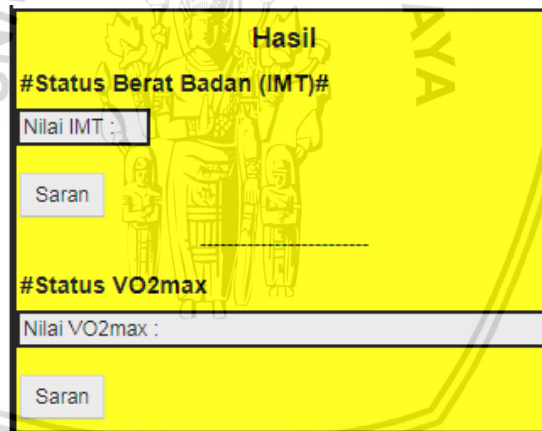
Tabel 5.4 *Komponen Penyusun Layout Masukan Data Sensor*

Nama Komponen	Jenis Komponen	Fungsi
Konek Denyut	<i>Button</i>	Berfungsi untuk menghubungkan perangkat ke arduino nano yang terhubung dengan <i>pulse sensor</i>
STdenyut	<i>Label</i>	Berfungsi untuk menunjukkan status koneksi perangkat ke arduino nano yang terhubung dengan <i>pulse sensor</i>
Konek Berat	<i>Button</i>	Berfungsi untuk menghubungkan perangkat ke arduino nano yang terhubung dengan <i>load sensor</i>
STberat	<i>Label</i>	Berfungsi untuk menunjukkan status koneksi perangkat ke arduino nano yang terhubung dengan <i>load sensor</i>
Ukur denyut	<i>Button</i>	Berfungsi untuk memicu arduino nano yang terhubung dengan <i>pulse sensor</i> untuk memulai perhitungan denyut nadi

Tabel 5.4 Komponen Penyusun Layout Masukan Data Sensor (Lanjutan)

Nama Komponen	Jenis Komponen	Fungsi
Ukur berat	<i>Button</i>	Berfungsi untuk memicu arduino nano yang terhubung dengan <i>load sensor</i> untuk memulai perhitungan berat badan
Hasil	<i>Button</i>	Berfungsi untuk menyembunyikan <i>layout</i> "masukan data sensor" kemudian memunculkan <i>layout</i> "hasil"

- e. Hasil
 Layout "hasil" memuat informasi hasil perhitungan dan klasifikasi dari VO_{2max} dan indeks masa tubuh berdasarkan data masukan dari pengguna. Pada *layout* ini pengguna dapat mengakses saran dari setiap hasil yang didapat. *Layout* pada "masukan data sensor" dijelaskan pada gambar 5.26.



Gambar 5.26 Layout Hasil

Dalam *layout* "hasil" terdapat komponen penyusun yang dijelaskan pada tabel 5.5.

Tabel 5.5 Komponen Penyusun Layout Hasil

Nama Komponen	Jenis Komponen	Fungsi
Nilai IMT	<i>Label</i>	Menampilkan hasil dari perhitungan nilai IMT
Status IMT	<i>Label</i>	Menampilkan hasil dari klasifikasi IMT

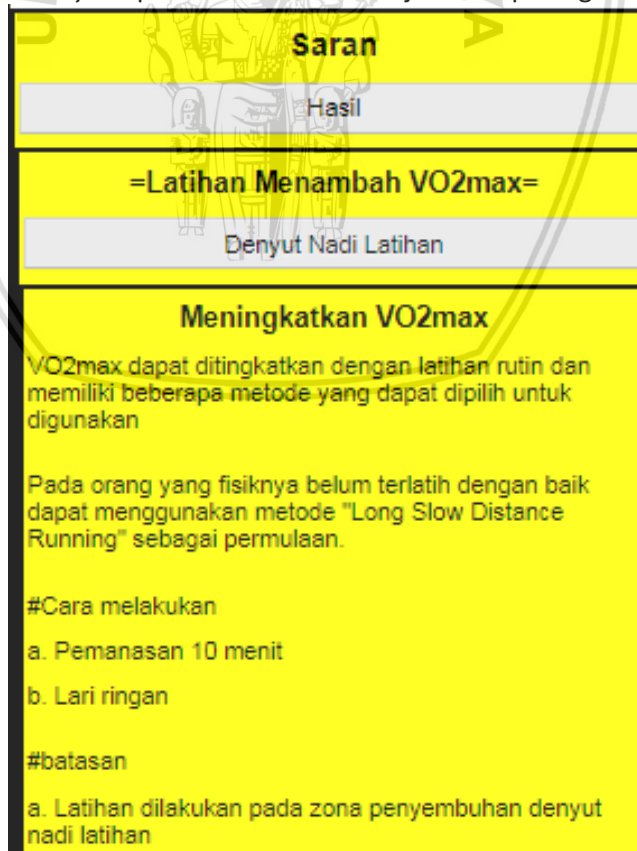


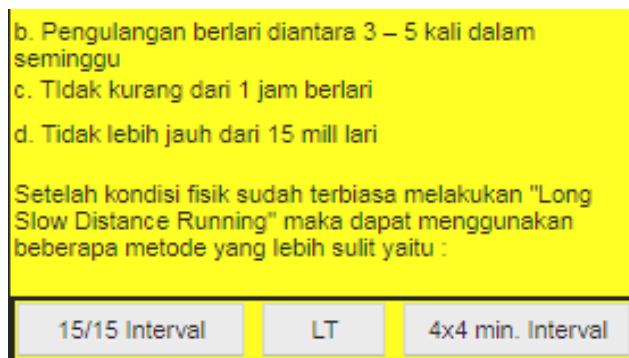
Tabel 5.5 Komponen Penyusun Layout Hasil (Lanjutan)

Nama Komponen	Jenis Komponen	Fungsi
Nilai VO _{2max}	Label	Menampilkan hasil dari perhitungan nilai VO _{2max}
Status VO _{2max}	Label	Menampilkan hasil dari klasifikasi VO _{2max}
Saran IMT	Button	Menyembunyikan layout "hasil" kemudian menampilkan layout "saran IMT"
Saran VO _{2max}	Button	Menyembunyikan layout "hasil" kemudian menampilkan layout "saran VO _{2max} "

f. Saran

Layout "saran" memuat tentang saran dari hasil perhitungan. Saran yang ditampilkan terbagi menjadi dua layout yaitu "saran VO_{2max}" dan "saran IMT". Layout pada "saran VO_{2max}" dijelaskan pada gambar 5.27.






Gambar 5.27 Layout Saran VO2max

Dalam *layout* "saran VO_{2max}" terdapat komponen penyusun yang dijelaskan pada tabel 5.6.

Tabel 5.6 Komponen Penyusun Layout Saran VO2max

Nama Komponen	Jenis Komponen	Fungsi
Hasil	Button	Menampilkan kembali <i>layout</i> "hasil" kemudian menyembunyikan <i>layout</i> "saran"
Denyut nadi latihan	Button	Menampilkan/menyembunyikan <i>layout</i> "denyut nadi latihan"
15/15 interval	Button	Menampilkan <i>layout</i> "15/15" kemudian menyembunyikan <i>layout</i> "saran"
LT	Button	Menampilkan <i>layout</i> "LT" kemudian menyembunyikan <i>layout</i> "saran"
4x4 min. interval	Button	Menampilkan <i>layout</i> "4x4 min. interval" kemudian menyembunyikan <i>layout</i> "saran"

Pada gambar 5.25 dan isi dari komponennya dijelaskan terdapat lanjutan dari *layout* "saran" yaitu *layout* "15/15 interval", " LT", "4x4 min. interval". Lanjutan dari *layout* dipisah ke beberapa bagian untuk memudahkan pengguna dalam membaca sehingga pengguna dapat lebih fokus saat membacara saran terkait VO_{2max} yang disajikan oleh aplikasi. Masing – masing *layout* dijelaskan pada gambar 5.28.



15/15 interval running


#Cara melakukan

- pemanasan 10 menit
- Lari ringan selama 15 detik
- Lari cepat selama 15 detik

#Batasan

- Kecepatan berlari tidak boleh berkurang dan selalu stabil
- Lakukan perulangan 12 – 20 kali
- Latihan dilakukan pada 90% - 95% denyut nadi latihan

(a)



Lactate threshold running (LT)


#Cara melakukan

- Pemanasan 10 menit dengan jogging ringan
- Lari cepat dengan jarak antara 800 – 1200 meter
- Lari ringan kembali skitar 400 meter
- Ulangi langkah sampai total jarak pada saat lari cepat 5000 meter

#Batasan

- Jarak total yang ditempuh saat berlari cepat dalam latihan adalah 5000 meter
- Kecepatan berlari tidak boleh berkurang dan selalu stabil
- Latihan Dilakukan pada 85% denyut nadi latihan.

(b)



4 x 4-min interval running

#Cara melakukan

- Pemanasan 10 menit
- 4 menit lari cepat
- 3 menit lari ringan

#Batasan

- Perulangan terbatas pada waktu pelaksanaan, sebaiknya sekitar 31 – 35 menit latihan
- Kecepatan berlari tidak boleh berkurang dan selalu stabil
- Latihan dilakukan pada 90% - 95% denyut nadi latihan

(c)

Gambar 5.28 Layout Saran Lanjutan VO_{2max}

Gambar 5.28(a) menjelaskan *layout* “15/15 interval”, gambar 5.28(b) menjelaskan *layout* “LT”, Gambar 5.28(c) menjelaskan *layout* “4x4 min.

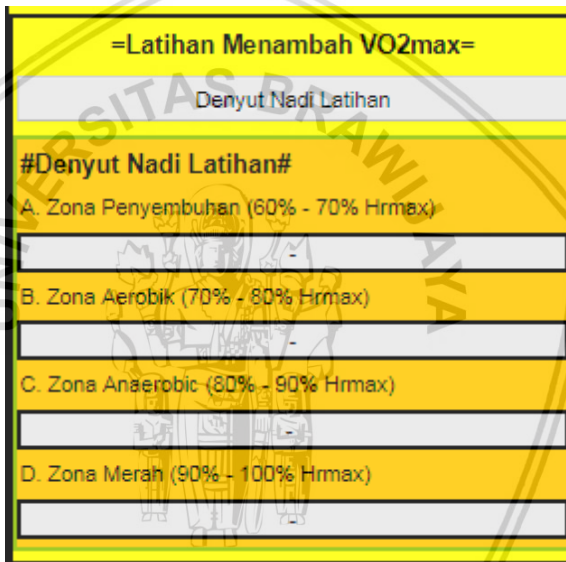


interval". Masing – masing dari *layout* pada gambar 5.28 mempunyai susunan komponen yang sama, komponen tersebut dijelaskan pada tabel 5.7.

Tabel 5.7 Komponen Penyusun *Layout* Saran Lanjutan VO_{2max}

Nama Komponen	Jenis Komponen	Fungsi
Kembali	<i>Button</i>	Menampilkan kembali <i>layout</i> "saran VO_{2max} "

Dalam *layout* "saran VO_{2max} " terdapat *layout* "denyut nadi latihan", *layout* ini berguna untuk mendukung informasi dari saran terkait cara meningkatkan VO_{2max} . Gambar 5.29 *layout* "denyut nadi latihan" ketika ditampilkan.



Gambar 5.29 *Layout* Denyut Nadi Latihan

Dalam *layout* "denyut nadi latihan" terdapat komponen penyusun yang dijelaskan pada tabel 5.8.

Tabel 5.8 Komponen Penyusun *Layout* Denyut Nadi Latihan

Nama Komponen	Jenis Komponen	Fungsi
Denyut nadi latihan	<i>Button</i>	Menampilkan/menyembunyi kan <i>layout</i> "denyut nadi latihan"

Selain *layout* "saran VO_{2max} ", dalam *layout* "saran" juga terdapat *layout* "saran IMT". Pemisahan dilakukan dengan tujuan agar pengguna dapat lebih fokus ke konten yang hanya diinginkan pengguna. *Layout* "saran IMT" yang dijelaskan pada gambar 5.30.



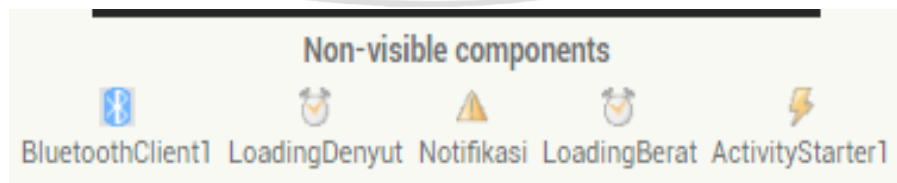
Gambar 5.30 *Layout Saran IMT*

Saran IMT akan menampilkan rentang berat badan yang harus dicapai pengguna untuk mendapatkan nilai IMT ideal. Dalam *layout* "saran IMT" terdapat komponen penyusun yang dijelaskan pada tabel 5.9.

Tabel 5.9 *Komponen Penyusun Layout Saran IMT*

Nama Komponen	Jenis Komponen	Fungsi
Hasil	<i>Button</i>	Menampilkan kembali <i>layout</i> "hasil" kemudian menyembunyikan <i>layout</i> "saran"
BAberat	<i>label</i>	Menyimpan hasil perhitungan batas atas dari rentang berat badan ideal
BBberat	<i>label</i>	Menyimpan hasil perhitungan batas bawah dari rentang berat badan ideal

Dalam mendukung jalannya aplikasi pada *designer* juga terdapat komponen tidak terlihat yang selanjutnya akan membantu dalam implementasi algoritmanya, komponen tersebut bernama *non-visible components*. *Non-visible components* yang digunakan pada aplikasi dijelaskan pada gambar 5.31.



Gambar 5.31 *Non-visible Components*

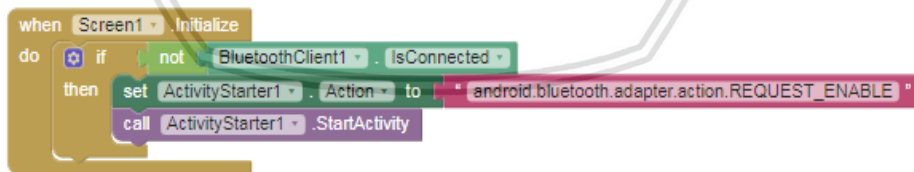
Dalam *non-visible components* terdapat komponen penyusun yang memiliki fungsi masing – masing untuk membantu aplikasi dalam menjalankan fungsi yang diinginkan. Komponen penyusun dari *non-visible components* dijelaskan dalam tabel 5.10.

Tabel 5.10 Komponen Penyusun *Non-visible Components*

Nama Komponen	Jenis Komponen	Fungsi
Bluetoothclient1	<i>Bluetooth client</i>	Mengatur konektivitas pada perangkat menggunakan <i>bluetooth</i>
Loadingdenyut	<i>Clock</i>	Mengatur perwaktuan dalam membuat proses menunggu hasil dari <i>pulse sensor</i>
Loadingberat	<i>Clock</i>	Mengatur perwaktuan dalam membuat proses menunggu hasil dari <i>load sensor</i>
Notifikasi	<i>Notifier</i>	Mengatur pembuatan notifikasi pada aplikasi
Activitystarter1	<i>Activitystarter</i>	Membantu aplikasi untuk dapat memanggil aktivitas pengaktifan <i>bluetooth</i> pada perangkat berbasis OS android

2. Bagian *blocks*

Blocks berfungsi sebagai implementasi dari algoritma pada *design* yang telah dibuat. *Blocks* berjalan secara paralel sehingga tidak perlu memperhatikan tata urutan dari penempatan *blocks*. *Event* pada *blocks* baru berjalan ketika kondisi dari *event* terpenuhi. Aplikasi baru akan bisa difungsikan maksimal ketika perangkat berbasis OS android terhubung melalui *bluetooth* ke arduino nano. Gambar 5.32 menjelaskan *blocks* untuk membuat perangkat mengaktifkan *bluetooth*.



Gambar 5.32 *Blocks* mengaktifkan Bluetooth

Aplikasi akan otomatis memanggil fungsi pada OS android untuk menampilkan *pop up* konfirmasi untuk menanyakan pengaktifan *bluetooth* pada perangkat berbasis OS android. Ketika *bluetooth* aktif maka pengguna dapat melanjutkan memasukan informasi yang dibutuhkan aplikasi untuk menjalankan fungsi utamanya.

Pengguna akan langsung ditampilkan *layout* “masukan informasi” sebagai awalan. Gambar 5.33 menjelaskan *blocks* untuk *layout* “masukan informasi” dari pengguna ke arduino nano.

```

when ButtonUmur .Click
do set Hasil_Umur . Text to Input_Umur . Text

when Button_Tinggi .Click
do set Hasil_Tinggi . Text to Input_Tinggi . Text

when Button_Laki .Click
do set Hasil_Gender . Text to " Laki - Laki "

when Button_Perempuan .Click
do set Hasil_Gender . Text to " Perempuan "
    
```

Gambar 5.33 *Blocks* Masukan Informasi

Komponen *text box* akan menyimpan masukan dari pengguna kemudian ketika pengguna menekan *button* “masukan” pada *layout* “masukan informasi” maka hasil dari masukan pengguna akan tersimpan dalam variabel tertentu untuk diproses selanjutnya. Jika pengguna belum memasukan semua data maka akan otomatis muncul peringatan dan proses akan tetap pada *layout* “masukan informasi”. Setelah selesai pengguna harus memilih “selesai” pada *layout* untuk memunculkan *layout* “masukan data sensor” yang berfungsi memasukan data sensor ke aplikasi. *Blocks* untuk pilihan “selesai” dijelaskan pada gambar 5.34

```

when ButtonPindahK .Click
do
  if Hasil_Umur . Text = " or ... or Hasil_Umur . Text = " Data ...
  then call Notifikasi .ShowAlert
      notice "Isi Semua Data terlebih Dahulu "
  else
    set Masukan_data_sensor . Visible to true
    set Masukan_Informasi . Visible to false
    set ButtonPindahK . Visible to false
    
```

Gambar 5.34 *Blocks* Selesai

Setiap perpindahan bagian proses aplikasi terjadi pada *screen* yang sama sehingga aplikasi hanya akan menyembunyikan atau memunculkan kembali bagian yang dibutuhkan ketika sedang berjalan. Pada *layout* “masukan data sensor” terdapat 2 bagian yaitu ukur denyut nadi dan ukur berat badan dimana setiap bagian memiliki *blocks* pendukung untuk dapat mendapat fungsi yang diinginkan yaitu untuk terhubung dengan sensor dan untuk menerapkan fungsi tunggu pada aplikasi ketika sensor masih melakukan pengukuran. *Blocks* masukan data sensor denyut nadi dijelaskan pada gambar 5.35.

```

when KonekDenyut .Click
do
  set STdenyut . Text to " "
  set STberat . Text to " "
  if call BluetoothClient1 .Connect
    address " 98:D3:32:30:C0:EF "
  then
    set STdenyut . Text to " terhubung "
    set STberat . Text to " tidak terhubung "
  else
    set STdenyut . Text to " gagal "
    set STberat . Text to " tidak terhubung "

```

(a)

```

when Kontroldenyut .Click
do
  if BluetoothClient1 . IsConnected and STdenyut . Text == " terhubung "
  then
    call BluetoothClient1 .SendText
      text " a "
    set LoadingDenyut . TimerEnabled to true
    call Notifikasi .ShowProgressDialog
      message " Memuat "
      title " Proses Perhitungan "
  else
    set STdenyut . Text to " Hubungkan Perangkat Terlebih Dahulu "

```

(b)

```

when LoadingDenyut .Timer
do
  if BluetoothClient1 . IsConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
  then
    call Notifikasi .DismissProgressDialog
    set HasilDenyut . Text to call BluetoothClient1 .ReceiveText
      numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    set LoadingDenyut . TimerEnabled to false
  else

```

(c)

Gambar 5.35 Blocks Ukur Denyut Nadi

Pada gambar 5.35(a) menunjukkan bagaimana cara perangkat terhubung dengan arduino nano yang terhubung dengan *pulse sensor*, hal tersebut ditandai dengan *blocks connect address* yang menunjuk ke alamat perangkat berada. Proses ini terjadi ketika pengguna menekan *button* “konek sensor denyut” pada *layout* “masukan data sensor”. Kemudian aplikasi akan menandai apakah status dari koneksi terhubung atau tidak, jika status menjadi “terhubung” maka barulah perhitungan denyut nadi dapat dilakukan. Pada gambar 5.35(b) proses *blocks* yang terjadi adalah mengirimkan teks “a” melalui modul *bluetooth* untuk menandai arduino nano untuk melakukan perhitungan denyut nadi. Aplikasi akan mengeluarkan *pop up* notifikasi berupa pemberitahuan “proses perhitungan”, pada saat ini aplikasi akan menunggu hasil dari arduino nano sampai selesai. Proses ini terjadi ketika pengguna menekan *button* “ukur denyut” pada *layout* “masukan data



sensor". Proses *blocks* pada gambar 5.35(c) akan menghentikan proses menunggu dari aplikasi dengan cara mematikan notifikasi yang berjalan ketika aplikasi menerima masukan dari arduino nano. Notifikasi pada aplikasi memanfaatkan fungsi *timer* untuk menandai mulainya proses menunggu sekaligus mengakhirinya dengan mengubah *timer enabled* menjadi *true* atau *false*.

Proses *blocks* ukur berat badan terpisah dengan ukur denyut nadi tetapi memiliki fungsi yang hampir mirip, proses *blocks* tersebut dijelaskan pada gambar 5.36.

```

(a)
when KonekBerat .Click
do
  set STdenyut .Text to ""
  set STberat .Text to ""
  if call BluetoothClient1 .Connect
     address "98:D3:33:81:01:EB"
  then
    set STberat .Text to "terhubung"
    set STdenyut .Text to "tidak terhubung"
  else
    set STberat .Text to "gagal"
    set STdenyut .Text to "tidak terhubung"
end

(b)
when KontrolBerat .Click
do
  if BluetoothClient1 .IsConnected and STberat .Text = "terhubung"
  then
    call BluetoothClient1 .SendText
      text "b"
    set LoadingBerat .TimerEnabled to true
    call Notifikasi .ShowProgressDialog
      message "Memuat"
      title "Proses Perhitungan"
  else
    set STberat .Text to "Hubungkan Perangkat Terlebih Dahulu"
end

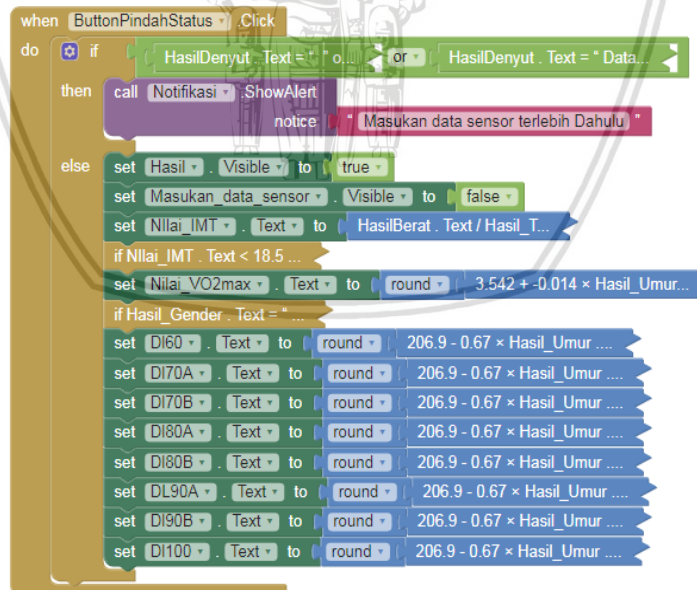
(c)
when LoadingBerat .Timer
do
  if BluetoothClient1 .IsConnected
  then
    call Notifikasi .DismissProgressDialog
    set HasilBerat .Text to call BluetoothClient1 .ReceiveText
      numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    set LoadingBerat .TimerEnabled to false
  else
end
  
```

Gambar 5.36 *Blocks* Ukur Berat Badan

Pada gambar 5.36(a) menunjukkan bagaimana cara perangkat terhubung dengan arduino nano yang terhubung dengan *load sensor*, hal tersebut

ditandai dengan *blocks connect address* yang menunjuk ke alamat perangkat berada. Kemudian aplikasi akan menandai apakah status dari koneksi terhubung atau tidak, jika status menjadi “terhubung” maka barulah perhitungan denyut nadi dapat dilakukan. Proses ini terjadi ketika pengguna menekan *button* “konek sensor berat” pada *layout* “masukan data sensor”. Pada gambar 5.36(b) proses *blocks* yang terjadi adalah mengirimkan *text* “b” melalui modul *bluetooth* untuk menandai arduino nano untuk melakukan perhitungan berat badan. Aplikasi akan mengeluarkan *pop up* notifikasi berupa pemberitahuan “proses perhitungan”, pada saat ini aplikasi akan menunggu hasil dari arduino nano sampai selesai. Proses *blocks* pada gambar 5.36(c) akan menghentikan proses menunggu dari aplikasi dengan cara mematikan notifikasi yang berjalan ketika aplikasi menerima masukan dari arduino nano. Notifikasi pada aplikasi memanfaatkan fungsi *timer* untuk menandai mulainya proses menunggu sekaligus mengakhirinya dengan merubah *timer enabled* menjadi *true* atau *false*.

Pada proses di *layout* “masukan data sensor”, mengukur denyut nadi dan berat badan tidak bisa berjalan bersamaan. Kekurangan tersebut dikarenakan modul *bluetooth* yang hanya bisa terhubung ke satu perangkat sehingga jika ingin mengukur pengukuran yang berbeda harus memutus koneksi ke salah satu perangkat terlebih dahulu. Jika semua data masukan sudah lengkap barulah pengguna dapat melihat hasil dari perhitungan VO_{2max} dan IMT beserta klasifikasi dengan cara memilih pilihan “hasil” pada *layout*. Gambar 5.37 menunjukkan proses *blocks* pada saat pengguna memilih “hasil”.



Gambar 5.37 *Blocks* Hasil

Setelah pengguna memilih “hasil” maka *layout* “masukan data sensor” akan disembunyikan kemudian *layout* “hasil” akan ditampilkan. Pada bagian ini terdapat beberapa perhitungan yang dilakukan yaitu sebagai berikut :

1. Perhitungan dan klasifikasi IMT

Perhitungan IMT menggunakan persamaan 2.5 dan klasifikasi berdasarkan pada tabel 2.5.
2. Perhitungan dan klasifikasi VO_{2max}

Perhitungan VO_{2max} menggunakan *non-exercise prediction method* di persamaan 2.3 dan persamaan 2.4 kemudian klasifikasi berdasarkan pada tabel 2.2 untuk perempuan dan tabel 2.3 untuk laki – laki.
3. Perhitungan denyut nadi latihan

Perhitungan denyut nadi latihan menggunakan persamaan 2.2. Denyut nadi latihan yang diukur yaitu 60%, 70%, 80%, 90% dan 100%. Setiap bagian dari hasil yang ditampilkan, pengguna dapat memunculkan saran terkait bagaimana mencapai nilai yang ideal. Saran yang dapat diberikan kepada pengguna meliputi sebagai berikut :

1. Saran VO_{2max}

Saran VO_{2max} menampilkan latihan – latihan efektif yang dapat dilakukan, *blocks* untuk menampilkan saran dijelaskan pada gambar 5.38.

```

when Saran_Vo2max .Click
do
  set Hasil .Visible to false
  set Saran .Visible to true
  set LLSaran_Beratt .Visible to false
  set LLSaran_vo2max .Visible to true
  set WPembuka_Saran_Vo2max .Visible to true
  set W151151 .Visible to false
  set W14x41 .Visible to false
  set WILT .Visible to false
    
```

(a)

```

when LT .Click
do
  set WPembuka_Saran_Vo2max .Visible to false
  set W151151 .Visible to false
  set W14x41 .Visible to false
  set WILT .Visible to true
    
```

(b)

```

when I151151 .Click
do
  set WPembuka_Saran_Vo2max .Visible to false
  set W151151 .Visible to true
  set W14x41 .Visible to false
  set WILT .Visible to false
    
```

(c)

```

when 14x41 .Click
do
  set WPembuka_Saran_Vo2max . Visible to false
  set W14x41 . Visible to true
  set W151515 . Visible to false
  set W1LT . Visible to false
  
```

(d)

```

when DenyutNadiLatihan .Click
do
  if Denyut_Nadi_latihan . Visible = true
  then set Denyut_Nadi_latihan . Visible to false
  else if Denyut_Nadi_latihan . Visible = false
  then set Denyut_Nadi_latihan . Visible to true
  
```

(e)

```

when kembaliSS1 .Click
do
  set Saran . Visible to true
  set LLSaran_Beratt . Visible to false
  set LLSaran_Vo2max . Visible to true
  set WPembuka_Saran_Vo2max . Visible to true
  set W151515 . Visible to false
  set W14x41 . Visible to false
  set W1LT . Visible to false
  
```

(f)

Gambar 5.38 Blocks Saran VO_{2max}

Gambar 5.38(a) menyembunyikan *layout* “hasil” kemudian menampilkan *layout* “saran VO_{2max}”. Saran VO_{2max} diawali dengan pengantar yang berisi olahraga ringan yang dapat dilakukan kemudian pengguna dapat memilih pilihan olahraga dengan tingkat yang lebih sulit yaitu *Lactate threshold running* (LT) pada gambar 5.38(b), 15/15 *interval running* (15/15) pada gambar 5.38(c) dan 4 x 4-min *interval running* (4 x 4) pada gambar 5.38 (d). Pada gambar 5.38(e) berfungsi untuk memunculkan atau menyembunyikan denyut nadi latihan. Gambar 5.36(f) merupakan *blocks* dari “kembali” yang berfungsi untuk mengembalikan pengguna ke saran awal dari meningkatkan VO_{2max}.

2. Saran IMT

Saran berat IMT menampilkan rentang berat badan yang harus dicapai oleh pengguna untuk mendapatkan status berat badan ideal. Saran hanya akan tampil ketika pengguna tidak memiliki berat badan ideal. *Blocks* untuk menampilkan saran dijelaskan pada gambar 5.39.

```

when Saran_Berat . Click
do
  set Hasil . Visible to false
  set Saran . Visible to true
  set LLSaran_Beratt . Visible to true
  set LLSaran_vo2max . Visible to false
  if HasilBerat . Text >= 18.5 and HasilBerat . Text <= 22.9
  then
    set HSangkaBerat1 . Text to "0"
    set HSangkaBerat2 . Text to "0"
    set HSberat . Text to "Kombinasi dari tinggi dan berat badan anda sudah..."
  else
    set HSberat . Text to "Untuk mendapatkan hasil ideal anda harus membuat..."
    set HSangkaBerat1 . Text to 18.5 x Hasil_Tinggi . Text ...
    set HSangkaBerat2 . Text to 22.9 x Hasil_Tinggi . Text ...
  
```

Gambar 5.39 Blocks Saran IMT

Berat badan yang harus dicapai oleh pengguna dihitung dengan menggunakan persamaan 5.1. Setiap saat pengguna dapat melihat kembali hasil secara keseluruhan setelah melihat semua saran, *blocks* tersebut dijelaskan pada gambar 5.40.

```

when PindahHasil . Click
do
  set Hasil . Visible to true
  set Saran . Visible to false
  set LLSaran_vo2max . Visible to false
  set LLSaran_Beratt . Visible to false
  
```

Gambar 5.40 Blocks Kembali Hasil

Pada *design* aplikasi terdapat *layout* “panel kontrol” yang akan selalu muncul pada bagian awal walau pengguna melakukan beberapa proses lain, panel tersebut berisi beberapa fungsi sebagai berikut :

1. Informasi

“Informasi” menampilkan data yang sudah dimasukan oleh pengguna, informasi tersebut disimpan dalam *layout* “informasi”. Pada gambar 5.41 menjelaskan *blocks* pada *layout* “Informasi”.

```

when hiddenInfo . Click
do
  if Informasi . Visible = true
  then
    set Informasi . Visible to false
  else if Informasi . Visible = false
  then
    set Informasi . Visible to true
  
```

Gambar 5.41 Blocks Informasi

Bagian informasi hanya menyembunyikan atau menampilkan kembali data yang dimasukan oleh pengguna.

2. Masukan data ulang

“Masukan data ulang” membuat pengguna kembali ke tempat *layout* “masukan informasi”, gambar 5.42 menjelaskan *blocks* pada *layout* “masukan informasi”. “Masukan data ulang” akan mengosongkan kembali semua data pengguna sehingga siap untuk dimasukan kembali.

```

when reset .Click
do
  set Masukan_data_sensor . Visible to false
  set Hasil . Visible to false
  set HasilDenyut . Text to "Data Kosong"
  set HasilBerat . Text to "Data Kosong"
  set Hasil_Gender . Text to "Data Kosong"
  set Hasil_Umur . Text to "Data Kosong"
  set Hasil_Tinggi . Text to "Data Kosong"
  set Masukan_Informasi . Visible to true
  set Saran . Visible to false
  set ButtonPindahK . Visible to true
  call Notifikasi .ShowAlert
  notice "Data pengguna di reset"
  
```

Gambar 5.42 *Blocks* Masukan data Ulang

3. Ukur ulang sensor

“Ukur ulang sensor” hampir sama seperti “Masukan data ulang”, tetapi pada bagian ini data yang dikosongkan hanya data sensor kemudian aplikasi akan kembali pada *layout* “masukan data sensor”. Gambar 5.43 menjelaskan *blocks* pada “Ukur ulang sensor”.

```

when ResetSensor .Click
do
  if Hasil_Umur . Text = " " or Hasil_Umur . Text = "Data ..."
  then
    call Notifikasi .ShowAlert
    notice "Data sensor belum terisi"
  else
    set Masukan_data_sensor . Visible to true
    set Hasil . Visible to false
    set Masukan_Informasi . Visible to false
    set Saran . Visible to false
    set ButtonPindahK . Visible to false
    set HasilDenyut . Text to "Data Kosong"
    set HasilBerat . Text to "Data Kosong"
    call Notifikasi .ShowAlert
    notice "Data sensor di reset"
  
```

Gambar 5.43 *Blocks* Ukur Ulang

4. Keluar

Pada aplikasi juga terdapat pilihan “keluar” yang berfungsi untuk menghentikan jalannya aplikasi. Selain memilih “keluar” pada *layout*

pengguna juga dapat menekan tombol *back* pada perangkat berbasis OS android untuk menghentikan jalannya aplikasi. Gambar 5.44 menjelaskan *blocks* ketika pengguna memilih “keluar” atau menekan tombol kembali.

```

when Keluar - .Click
do
  call Notifikasi - .ShowChooseDialog
    message "Keluar Dari Aplikasi ? "
    title "Peringatan ! "
    button1Text "Ya "
    button2Text "Tidak "
    cancelable false
  
```

(a)

```

when Screen1 - .BackPressed
do
  call Notifikasi - .ShowChooseDialog
    message "Keluar Dari Aplikasi ? "
    title "Peringatan ! "
    button1Text "Ya "
    button2Text "Tidak "
    cancelable false
  
```

(b)

```

when Notifikasi - .AfterChoosing
choice
do
  if compare texts get choice - = "Ya "
  then
    call BluetoothClient1 - .Disconnect
    close application
  
```

(c)

Gambar 5.44 Blocks Keluar

Blocks pada gambar 5.44(a) dan 5.44(b) akan memunculkan peringatan berupa konfirmasi apakah pengguna ingin keluar dari aplikasi, jika pengguna memilih “Ya” maka pilihan dari pengguna akan diproses pada *blocks* di gambar 5.44(c) untuk menghentikan koneksi dari *bluetooth* dan menutup aplikasi.

Panel kontrol dapat disembunyikan atau dimunculkan, hal ini bertujuan untuk meringkas tampilan sehingga pengguna tidak kebingungan karena sudah terlalu banyak *layout* yang muncul dalam satu *screen*. Gambar 5.45 menjelaskan *blocks* untuk memunculkan atau menyembunyikan panel kontrol.

```
when hiddenkontrol .Click  
do  
  if kontrol . Visible = true  
  then set kontrol . Visible to false  
  else if kontrol . Visible = false  
  then set kontrol . Visible to true
```

Gambar 5.45 Blocks menyembunyikan/menampilkan layout Panel kontrol



BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis menjelaskan bagaimana sistem diuji setelah implementasi selesai dilakukan kemudian menganalisis hasil pengujian tersebut untuk mencari kesimpulan dari kinerja sistem.

6.1 Pengujian *Pulse Sensor*

Tujuan dari pengujian ini adalah untuk mengetahui akurasi dari *pulse sensor* apakah memiliki akurasi yang baik atau tidak. Pengujian pada *pulse sensor* dilakukan dengan cara membandingkan hasil dari pengukuran manual dengan pengukuran menggunakan *pulse sensor* pada sistem. Pengukuran manual dilakukan dengan cara meletakkan ujung jari pada pergelangan tangan dimana subjek merasakan ada denyut nadi. Subjek menghitung berapa jumlah denyut nadi dalam waktu 15 detik kemudian mengalikannya dengan 4 agar mendapatkan BPM.

6.1.1 Prosedur Pengujian

Pengujian yang *pulse sensor* terhadap pengukuran manual dilakukan dengan prosedur sebagai berikut :

1. Subjek meletakkan ujung jari pada *pulse sensor*
2. Hasil pengukuran menggunakan *pulse sensor* dicatat
3. Subjek melakukan pengukuran manual
4. Hasil pengukuran manual dicatat
5. Hasil yang tercatat direkap dalam bentuk tabel
6. Setiap hasil yang telah direkap diproses dengan persamaan 6.1 untuk mencari persentase kesalahan dalam pengujian.

$$\frac{|\text{Pengukuran manual} - \text{Pengukuran sensor}|}{\text{Pengukuran manual}} \times 100\% \quad (6.1)$$

7. Hasil dari perhitungan persentase kesalahan dihitung rata – ratanya dari 10 kali pengukuran dengan subjek yang berbeda untuk mendapatkan hasil dari keseluruhan
8. Hasil keseluruhan kemudian dianalisis untuk mendapatkan kesimpulan dari pengujian

6.1.2 Hasil Pengujian

Hasil pengujian dimuat dalam bentuk tabel dan mengikuti prosedur pengujian. Hasil dari pengujian dari *pulse sensor* dimuat dalam tabel 6.1.

Tabel 6.1 Hasil Pengujian Pulse Sensor

Pengukuran ke -	Pengukuran Manual (BPM)	Pengukuran Sensor (BPM)	Persentase Kesalahan (%)
1	64	80	25 %
2	64	80	25 %
3	64	75	17,19 %
4	64	65	1,56 %
5	64	64	0 %
6	64	62	3,12 %
7	64	62	3,12 %
8	64	67	4,68 %
9	64	65	1,56 %
10	64	64	0 %
Rata – Rata			8,12 %

6.1.3 Analisis Pengujian

Menurut hasil pengujian pembacaan pada *pulse sensor* mengalami rata – rata kesalahan sebesar 8,12 %. Kesalahan yang terjadi dikarenakan *pulse sensor* sangat dipengaruhi oleh posisi ujung jari pada sensor dan tekanan yang diberikan pada sensor sehingga sebagian pembacaan terdapat kesalahan. Selain masalah tersebut *pulse sensor* juga membutuhkan waktu untuk stabil dalam pembacaan ketika pengguna pertama kali memegang sensor.

6.2 Pengujian Load Sensor

Tujuan dari pengujian ini adalah untuk mengetahui akurasi dari *load sensor* apakah memiliki akurasi yang baik atau tidak. Pengujian pada *load sensor* dilakukan dengan cara membandingkan hasil dari pengukuran manual dengan pengukuran menggunakan *load sensor* pada sistem. Pengukuran manual dilakukan dengan menggunakan timbangan digital.

6.2.1 Prosedur Pengujian

Pengujian *load sensor* terhadap pengukuran manual dilakukan dengan prosedur sebagai berikut :

1. Subjek meletakkan ujung jari pada *load sensor*
2. Hasil pengukuran menggunakan *load sensor* dicatat
3. Subjek melakukan pengukuran manual

4. Hasil pengukuran manual dicatat
5. Hasil yang tercatat direkap dalam bentuk tabel
6. Setiap hasil yang telah direkap diproses dengan persamaan 6.1 mencari persentase kesalahan dalam pengujian.
7. Hasil dari perhitungan persentase kesalahan dihitung rata – ratanya dari 5 kali pengukuran dengan subjek yang berbeda untuk mendapatkan hasil dari keseluruhan
8. Hasil keseluruhan kemudian dianalisis untuk mendapatkan kesimpulan dari pengujian

6.2.2 Hasil Pengujian

Hasil pengujian dimuat dalam bentuk tabel dan mengikuti prosedur pengujian. Hasil dari pengujian dari *load sensor* dimuat dalam tabel 6.2.

Tabel 6.2 Hasil Pengujian Load Sensor

Subjek	Pengukuran Manual (Kg)	Pengukuran Sensor (Kg)	Persentase Kesalahan (%)
1	55,5	54,8	1,26 %
2	67,6	67	0,88 %
3	46,5	45,8	1,5 %
4	59,9	59,2	0,11 %
5	68,4	67,8	0,88 %
Rata – Rata			0,92 %

6.2.3 Analisis Pengujian

Menurut hasil pengujian pembacaan pada *load sensor* mengalami rata – rata kesalahan sebesar 0,92 %. Kesalahan yang terjadi dikarenakan *load sensor* memiliki kesalahan komprehensif 0,05 mv/V sehingga dapat terjadi ketidakakuratan pembacaan pada sensor. Saat pengguna meletakkan kaki, posisi kaki yang sejajar atau bertumpu dengan baik pada perangkat juga dapat menjadi pengaruh karena sensor akan kesulitan dalam menstabilkan hasil yang keluar.

6.3 Pengujian Modul *Bluetooth*

Tujuan dari pengujian ini adalah untuk mengetahui jarak maksimal dari modul dalam mengirimkan data. Pengujian pada modul *bluetooth* dilakukan dengan cara mengirimkan data melalui *bluetooth* dari arduino nano ke perangkat berbasis OS android. Pengiriman data dilakukan pada jarak yang berbeda – beda yaitu 1 meter, 5 meter, 10 meter, 12 meter dan 14 meter.

Pengujian modul *bluetooth* untuk mengetahui jarak maksimal yang efektif untuk mengirimkan data tanpa kesalahan dilakukan dengan prosedur sebagai berikut :

1. Arduino nano dibuat mengirimkan data secara berkala setiap detik
2. Penguji berada pada jarak yang ditentukan sambil mengamati data yang masuk
3. Penguji mencatat 5 data yang masuk pada setiap jarak yang ditentukan
4. Hasil dari pengujian dimasukan ke dalam tabel untuk dianalisis.

6.3.2 Hasil Pengujian

Hasil pengujian dimuat dalam bentuk tabel dan mengikuti prosedur pengujian. Hasil dari pengujian setiap jarak dari modul *bluetooth* dimuat dalam tabel terdiri dari sebagai berikut :

- a. Pengujian dengan jarak 1 meter

Pada pengujian ini menggunakan jarak sejauh 1 meter, hasil dari pengujian terdapat pada tabel 6.3.

Tabel 6.3 Pengujian Modul *Bluetooth* Pada Jarak 1 Meter

No. pengujian	Data terkirim	Data diterima	Status
1	"Data terkirim 1"	"Data terkirim 1"	Berhasil
2	"Data terkirim 2"	"Data terkirim 2"	Berhasil
3	"Data terkirim 3"	"Data terkirim 3"	Berhasil
4	"Data terkirim 4"	"Data terkirim 4"	Berhasil
5	"Data terkirim 5"	"Data terkirim 5"	Berhasil

- b. Pengujian dengan jarak 5 meter

Pada pengujian ini menggunakan jarak sejauh 5 meter, hasil dari pengujian terdapat pada tabel 6.4.

Tabel 6.4 Pengujian Modul *Bluetooth* Pada Jarak 5 Meter

No. pengujian	Data terkirim	Data diterima	Status
1	"Data terkirim 1"	"Data terkirim 1"	Berhasil
2	"Data terkirim 2"	"Data terkirim 2"	Berhasil
3	"Data terkirim 3"	"Data terkirim 3"	Berhasil
4	"Data terkirim 4"	"Data terkirim 4"	Berhasil
5	"Data terkirim 5"	"Data terkirim 5"	Berhasil

c. Pengujian dengan jarak 10 meter

Pada pengujian ini menggunakan jarak sejauh 10 meter, hasil dari pengujian terdapat pada tabel 6.5.

Tabel 6.5 Pengujian Modul Bluetooth Pada Jarak 10 Meter

No. pengujian	Data terkirim	Data diterima	Status
1	"Data terkirim 1"	"Data terkirim 1"	Berhasil
2	"Data terkirim 2"	"Data terkirim 2"	Berhasil
3	"Data terkirim 3"	"Data terkirim 3"	Berhasil
4	"Data terkirim 4"	"Data terkirim 4"	Berhasil
5	"Data terkirim 5"	"Data terkirim 5"	Berhasil

d. Pengujian dengan jarak 12 meter

Pada pengujian ini menggunakan jarak sejauh 12 meter, hasil dari pengujian terdapat pada tabel 6.6.

Tabel 6.6 Pengujian Modul Bluetooth Pada Jarak 12 Meter

No. pengujian	Data terkirim	Data diterima	Status
1	"Data terkirim 1"	"Data terkirim 1"	Berhasil
2	"Data terkirim 2"	"Data terkirim 2"	Berhasil
3	"Data terkirim 3"	-	Gagal
4	"Data terkirim 4"	-	Gagal
5	"Data terkirim 5"	"Data terkirim 3"	Terlambat

e. Pengujian dengan jarak 15 meter

Pada pengujian ini menggunakan jarak sejauh 15 meter, hasil dari pengujian terdapat pada tabel 6.7.

Tabel 6.7 Pengujian Modul Bluetooth Pada Jarak 14 Meter

No. pengujian	Data terkirim	Data diterima	Status
1	"Data terkirim 1"	-	<i>Connection Lost</i>
2	"Data terkirim 2"	-	<i>Connection Lost</i>
3	"Data terkirim 3"	-	<i>Connection Lost</i>
4	"Data terkirim 4"	-	<i>Connection Lost</i>
5	"Data terkirim 5"	-	<i>Connection Lost</i>

6.3.3 Analisis Pengujian

Menurut hasil pengujian pada modul *bluetooth* terdapat kesalahan pengiriman data pada tabel 6.6 dijarak 12 meter dan pada tabel 6.7 perangkat berbasis OS android kehilangan koneksi dengan modul *bluetooth*. Kesalahan pada pengiriman diakibatkan oleh jangkauan maksimal dari modul sendiri yaitu kurang lebih 10 meter. Pada pengiriman dalam jarak kurang dari 10 meter data yang diterima tidak mengalami kesalahan sehingga penggunaan modul *bluetooth* harus dalam jangkauan jarak 10 meter untuk mendapatkan fungsi maksimal dari modul.

6.4 Pengujian Tingkat Kebugaran Kardiorespirasi

Pengujian bertujuan untuk menguji dampak dari latihan yang dijalani terhadap kebugaran kardiorespirasi. Pengujian ini dilakukan dengan cara menyurvei 5 subjek untuk mendapatkan informasi denyut nadi, berat badan dan intensitas latihan fisik yang dilakukan. 2 dari 5 subjek melakukan olahraga untuk meningkatkan kebugaran kardiorespirasi sedangkan 3 subjek yang tersisa mengurangi intensitas olahraga yang dilakukan.

Pengujian tingkat kebugaran kardiorespirasi terhadap intensitas latihan yang dilakukan dengan prosedur sebagai berikut :

1. Survei 5 subjek untuk mendapatkan informasi denyut nadi, berat badan, umur dan intensitas latihan fisik yang dilakukan
2. Menghitung kebugaran kardiorespirasi (VO_{2max}) menggunakan *non-exercise prediction method*, IMT dan hasil klasifikasi pada informasi yang didapatkan
3. Subjek ke-1 melakukan metode *15/15 interval running* dan subjek ke-2 melakukan metode *long slow distance running* dalam 1 Minggu
4. 3 subjek yang tersisa mengurangi intensitas olahraga yang dilakukan
5. Hasil dari pengujian dianalisis perbandingan antara data subjek saat pertama kali diambil dengan data setelah subjek melakukan metode pengujian

6.4.2 Hasil Pengujian

Hasil dari pengujian dimuat dalam tabel 6.8 untuk dibandingkan dampak yang terjadi pada tingkat kebugaran kardiorespirasi.

Tabel 6.8 Hasil Tingkat Kebugaran Terhadap Pengujian Yang Dilakukan

No.	Data Subjek Sebelum Metode Pengujian Dilakukan	Data Subjek Setelah Metode Pengujian Dilakukan
1	Umur : 23	
	Denyut Nadi : 68 BPM	Denyut Nadi : 68 BPM
	Berat Badan : 54,8 Kg	Berat Badan : 54 Kg
	Intensitas Latihan : 0 kali/Minggu	Intensitas Latihan : 3 kali/Minggu

Tabel 6.8 Hasil Tingkat Kebugaran Terhadap Pengujian Yang Dilakukan (Lanjutan)

No.	Data Subjek Sebelum Metode Pengujian Dilakukan	Data Subjek Setelah Metode Pengujian Dilakukan
1	Nilai VO _{2max} (ml/Kg/min) : 60	Nilai VO _{2max} (ml/Kg/min) : 61
	Tingkat Kebugaran : Sangat Tinggi	Tingkat Kebugaran : Sangat Tinggi
2	Umur : 22	
	Denyut Nadi : 64 BPM	Denyut Nadi : 64 BPM
	Berat Badan : 71 Kg	Berat Badan : 67,7 Kg
	Intensitas Latihan : 0 kali/Minggu	Intensitas Latihan : 3 kali/Minggu
	Nilai VO _{2max} (ml/Kg/min) : 51	Nilai VO _{2max} (ml/Kg/min) : 52
	Tingkat Kebugaran : Baik Sekali	Tingkat Kebugaran : Baik Sekali
3	Umur : 20	
	Denyut Nadi : 100 BPM	Denyut Nadi : 100 BPM
	Berat Badan : 47 Kg	Berat Badan : 48 Kg
	Intensitas Latihan : 7 kali/Minggu	Intensitas Latihan : 2 kali/Minggu
	Nilai VO _{2max} (ml/Kg/min) : 61	Nilai VO _{2max} (ml/Kg/min) : 60
	Tingkat Kebugaran : Sangat Tinggi	Tingkat Kebugaran : Sangat Tinggi
4	Umur : 21	
	Denyut Nadi : 88 BPM	Denyut Nadi : 88 BPM
	Berat Badan : 59,9 Kg	Berat Badan : 60,3 Kg
	Intensitas Latihan : 4 kali/Minggu	Intensitas Latihan : 3 kali/Minggu
	Nilai VO _{2max} (ml/Kg/min) : 53	Nilai VO _{2max} (ml/Kg/min) : 53
	Tingkat Kebugaran : Baik Sekali	Tingkat Kebugaran : Baik Sekali
5	Umur : 19	
	Denyut Nadi : 100 BPM	Denyut Nadi : 104 BPM
	Berat Badan : 47 Kg	Berat Badan : 47,3 Kg
	Intensitas Latihan : 3 kali/Minggu	Intensitas Latihan : 2 kali/Minggu
	Nilai VO _{2max} (ml/Kg/min) : 61	Nilai VO _{2max} (ml/Kg/min) : 60
	Tingkat Kebugaran : Sangat Tinggi	Tingkat Kebugaran : Sangat Tinggi

6.4.3 Analisis Pengujian

Menurut hasil perbandingan pada hasil pengujian di tabel 6.8 terdapat peningkatan nilai VO_{2max} terhadap subjek 1 dan 2. Subjek 2 yang melakukan 15/15 *interval running* mengalami peningkatan yang lebih tinggi dari pada subjek 1 yang melakukan *long slow distance running*, hal ini dikarenakan metode latihan yang dilakukan subjek 2 lebih berat dari pada subjek 1. Pada subjek 3 dan 5 terjadi peningkatan berat badan karena subjek mengurangi latihan yang biasa dilakukan kecuali pada subjek 4 yang mengalami peningkatan pada denyut nadi dan berat badan. Perubahan yang terjadi pada subjek 3 dan 5 mengakibatkan sedikit penurunan nilai VO_{2max} tetapi pada subjek 4 tidak terjadi penurunan yang berarti sehingga nilai VO_{2max} tetap stabil.



BAB 7 PENUTUP

Bab penutup memuat kesimpulan dan saran dari penelitian yang sudah dilakukan. Kesimpulan memuat garis besar dari perancangan sistem sampai dengan hasil pengujian dan analisis sedangkan saran memuat bayangan ke depan sistem sehingga dapat dikembangkan menjadi lebih baik.

7.1 Kesimpulan

Garis besar dari perancangan sistem sampai dengan hasil pengujian dan analisis yaitu :

1. Sistem dapat mengukur denyut nadi dengan memanfaatkan *pulse sensor* sebagai perantara yang digunakan pada ujung jari pengguna.
2. Sistem dapat mengukur berat badan dengan memanfaatkan *load sensor* sebagai perantara dibantu dengan ADC HX711 untuk membantu perhitungan. Pengguna harus berdiri di atas sensor untuk mendapat perhitungan berat badan.
3. Sistem dapat melakukan perhitungan kebugaran kardiorespirasi dengan menggunakan *non-exercise prediction method* dan ditambah dengan perhitungan indeks masa tubuh untuk melengkapi jalannya sistem. Sistem juga dapat memberikan saran sehubungan dengan hasil yang diberikan sehingga pengguna mengerti bagaimana mencapai kebugaran yang baik.
4. Arduino nano dapat terhubung dengan perangkat berbasis OS android dan mencapai prinsip portabilitas dengan memanfaatkan modul *bluetooth* dengan jarak maksimal 10 meter sebagai tempat bertukar data menggantikan kabel.
5. Pembuatan sistem untuk pengukuran denyut nadi dan berat badan menggunakan mendapatkan hasil akurasi yang tergolong baik dengan rata – rata *error* pada *pulse sensor* sebesar 8,12 % dan *load sensor* sebesar 0,92 %.

7.2 Saran

Saran memuat bayangan ke depan sistem sehingga dapat dikembangkan menjadi lebih baik yaitu :

1. Antarmuka sebagai interaksi antar pengguna dan perangkatnya sebaiknya memiliki tampilan dan tata letak dari bagian aplikasi yang baik sehingga pengguna lebih nyaman menggunakan tanpa harus memahami cara berjalannya aplikasi lebih lama.
2. *Casing* pada *pulse sensor* sebaiknya menggunakan prinsip *wearable device* sehingga pengguna dapat lebih mudah mengawasi kondisi tubuhnya dengan *realtime* saat diam maupun saat melaksanakan aktivitas olahraga.

3. Konektifitas antar perangkat sebaiknya terhubung manual dua arah tanpa harus memutus salah satu koneksi dari sensor terlebih dahulu untuk menggunakan sensor yang lainnya.



DAFTAR PUSTAKA

- Akbar, F., 2016. *Rancang bangun alat pendeteksi denyut jantung dan suhu tubuh menggunakan pulse sensor dan sensor LM35*. S1. Universitas Brawijaya. Tersedia di <<http://ptiik.ub.ac.id/skripsi>> [Diakses 25 Mei 2016]
- American College of Sport Medicine, 2008. *ACSM's Health-Related Physical Fitness Manual 2nd ed*. Philadelphia, PA: Lippincott Williams & Wilkins [electronic print] Tersedia di: < <http://ebook30.com/science/medicine/50959/acsrnshealthrelated-physical-fitness-assessmdnt-manual.html> > [Diakses 18 september 2016]
- Android., 2017. *Arsitektur Platform*. [online] Android. Tersedia di: < <https://developer.android.com/guide/platform/index.html?hl=id> > [Diakses 14 Juli 2017]
- Arduino, 2017. *Arduino Product*. [image online] Tersedia di: <<https://www.arduino.cc/en/Main/Products>> [Diakses 8 Oktober 2017]
- Brien SE, Katzmarzyk PT, Craig CL and Gauvin L, 2007. *Physical activity, cardiorespiratory fitness and body mass index as predictors of substantial weight gain and obesity: the Canadian physical activity longitudinal study*. Can J Public Health 98: 121–124.
- Brian Mac Sport Coach., 2016. *Heart Rate Zones*. [online] Brian Mac Sport Coach. Tersedia di: <<https://www.brianmac.co.uk/hrm1.htm>> [Diakses 11 November 2016]
- Brian Mac Sport Coach., 2016. *Maximum Heart Rate*. [online] Brian Mac Sport Coach. Tersedia di: < <https://www.brianmac.co.uk/maxhr.htm> > [Diakses 11 November 2016]
- Brian Mac Sport Coach., 2016. *VO₂ Max*. [online] Brian Mac Sport Coach. Tersedia di: <<https://www.brianmac.co.uk/vo2max.htm#vo2>> [Diakses 11 November 2016]
- Center for Obesity Research and Education.,2017. *Body Mass Index: BMI Calculator*. [Online] Tersedia di: <<http://www.core.monash.org/bmi.html> > [Diakses 8 Maret 2017]
- Danial et al., 2011. *Proporsi Indeks Massa Tubuh (IMT) Penderita Penyakit Jantung Koroner Di RSUP Haji Adam Malik, Medan* [pdf] Tersedia di: <<http://repository.usu.ac.id/handle/123456789/25638>> [Diakses 20 Januari 2017]
- GitHub., 2017. *An Arduino library to interface the Avia Semiconductor HX711 24-Bit Analog-to-Digital Converter (ADC) for Weight Scales*. [online] Bogde. Tersedia di: <<https://github.com/bogde/HX711>> [Diakses 14 Juli 2017]
- Hanum.Misra, 2004. *Pengaruh Latihan Aerobik Terhadap Ambilan Oksigen Maksimum (Vo₂ max) Pada Fase-Fase Siklus Menstruasi Wanita Usia 18-*

24 Tahun Yang Tidak Terlatih [pdf] Tersedia di: <<http://repository.usu.ac.id/xmlui/handle/123456789/6186?show=full>> [Diakses 20 Januari 2017]

Heywood, V., 2006. *The Physical Fitness Specialist Manual, The Cooper Institute for Aerobics Research*. Dallas : TX.

Helgerud, Jan et al., 2007. *Aerobic High-Intensity Intervals Improve VO2max More Than Moderate Training*. [pdf] Tersedia di: <https://www.researchgate.net/profile/Kjetil_Hoydal/publication/6407035_Aerobic_Highintensity_Intervals_Improve_VO2max_More_Than_Moderate_Training/links/563c69c008ae34e98c493e9d.pdf> [Diakses 3 Oktober 2017]

Howley et al ,1995. *Criteria for maximal oxygen uptake: review and commentary*. Med Sci Sports Exerc.

Irianto, D. D. P., 2004. *Berolahraga untuk Kebugaran & Kesehatan*. Yogyakarta: ANDI.

ITead Studio, 2010. *Datasheet Bluetooth HC-05*. [pdf] Tersedia di: <www.electronicastudio.com/docs/istd016A.pdf> [Diakses 20 Januari 2017]

Jackson, A.S. et al., 2007. *Estimating Maximum Heart Rate From Age: Is It a Linear Relationship? Medicine & Science in Sports & Exercise*, 39(5), p. 822-829

Jheri Turnley, B.S., 2010. *VO2max: How Can An Endurance Athlete Use It To Obtain Peak Performance?* [pdf] Tersedia di: <<http://henryortho.com/pdf/VO2max.pdf>> [Diakses 3 Oktober 2017]

Karvonen, J. and Vuorimaa, T, 1988. *Heart rate and exercise intensity during sports activities*. Sports Medicine

Lee et al., 2010. *Mortality trends in the general population: the importance of cardiorespiratory fitness*. [pdf] Tersedia di: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2951585/>> [Diakses 8 Maret 2017]

Massachusetts Institute of Technology., 2017. *Getting Started with MIT App Inventor 2*. [online] MIT App Inventor. Tersedia di: <<http://appinventor.mit.edu/explore/get-started?>> [Diakses 14 November 2017]

Olivia, Wilhelmina., 2012. *Hubungan Indeks Massa Tubuh dengan Kebugaran Fisik pada Mahasiswa Laki-Laki Fakultas Kedokteran Universitas Sumatera Utara Tahun Masuk*. [pdf] Tersedia di: <<http://repository.usu.ac.id/handle/123456789/31463>> [Diakses 20 April 2016]

Polar., 2016. *Your resting heart rate is telling you something – so listen*. [online] Polar. Tersedia di: <<https://www.polar.com/blog/resting-heart-rate>> [Diakses 8 Maret 2017]

- Pulse Sensor., 2016. *Pulse Sensor Amped*. [online] Pulse Sensor. Tersedia di: <<https://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1>> [Diakses 8 Maret 2017]
- Rahayu, Indri., 2013. *Kebugaran Jasmani Lanjutan* [pdf] Tersedia di: http://file.upi.edu/Direktori/FPOK/JUR._PEND._KESEHATAN_&_REKREASI/PRODI._KEPERAWATAN/198110192003122-NUR_INDRI_RAHAJU/KEBUGARAN_JASMANI_LANJUTAN_ARTIKEL.pdf [Diakses 20 April 2016]
- Rexhepi et al., 2014. *Prediction of VO₂max based on age, body mass, and resting heart rate* [pdf] Tersedia di: <<https://www.degruyter.com/downloadpdf/j/humo.2014.15.issue-/humo-2014-0003/humo-2014-0003.pdf>> [Diakses 11 Juli 2017]
- Sparkfun Electronics., 2017. *Load Cell Amplifier HX711 Breakout Hookup Guide*. [online] Sparkfun Electronics. Tersedia di: <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?_ga=2.47565220.220611380.1513215199-1251858349.1489277699> [Diakses 20 November 2017]
- Sparkfun Electronics., 2017. *Getting Started with Load Cells*. [online] Sparkfun Electronics. Tersedia di: <https://learn.sparkfun.com/tutorials/getting-started-with-load-cells?_ga=2.76875798.220611380.1513215199-1251858349.1489277699> [Diakses 20 November 2017]
- Taalongonan et al., 2014. *Rancang Bangun Alat Penghitung Indeks Massa Tubuh* [pdf] Tersedia di: <<http://download.portalgaruda.org/article.php?article=262306&val=1043&title=RANCANG%20BANGUN%20ALAT%20PENGHITUNG%20INDEKS%20MASSA%20TUBUH>> [Diakses 14 Juli 2017]
- Theory Circuit., 2017. *Pulse Sensor with Arduino*. [online] Theory Circuit. Tersedia di: <<http://www.theorycircuit.com/pulse-sensor-arduino>> [Diakses 14 Juli 2017]
- UTH, N. et al., 2004 Estimation of VO₂ max from the ratio between HR_{max} and HR_{rest} - the Heart Rate Ratio Method". [pdf] Tersedia di: <<https://www.degruyter.com/downloadpdf/j/humo.2014.15.issue-1/humo-2014-0003/humo-2014-0003.pdf>> [Diakses 11 Juli 2017]
- Theory Circuit., 2017. *Pulse Sensor with Arduino*. [online] Theory Circuit. Tersedia di: <<http://www.theorycircuit.com/pulse-sensor-arduino>> [Diakses 14 Juli 2017]