

**SISTEM *MONITORING* DENYUT JANTUNG MENGGUNAKAN
NODEMCU DAN MQTT**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Falachudin Akbar
NIM: 135150301111073



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PERSETUJUAN

SISTEM *MONITORING* DENYUT JANTUNG MENGGUNAKAN
NODEMCU DAN MQTT

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Falachudin Akbar
NIM: 135150301111073

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc
NIK. 2016078910091001

Dosen Pembimbing II



Hurriyatul Fitriyah, S.T., M.Sc
NIP. 19851001 201504 2 003

Mengetahui
Ketua Jurusan Teknik Informatika



Tji Astoro Kurniawan, S.T., M.T., Ph.D
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, 16 Juli 2018



Falachudin Akbar

NIM: 135150301111073

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang selalu memberikan rahmat, hidayah, dan cahaya-Nya sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “Sistem *Monitoring* denyut jantung menggunakan Nodemcu dan MQTT”. Adapun maksud dari penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Program Teknologi Informasi dan Ilmu Komputer.

Penyelesaian skripsi ini tidak terlepas dari bimbingan dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Rizal Maulana, S.T., M.T., M.Sc selaku dosen pembimbing pertama dan Ibu Hurriyatul Fitriyah, S.T, M.Sc selaku dosen pembimbing kedua yang telah memberikan pengarahan dengan sabar terhadap penulis dalam perancangan hingga penyelesaian skripsi ini.
2. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku Ketua Program Studi Teknik komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya.
4. Ayahanda dan Ibunda beserta seluruh anggota keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti – hentinya memberikan do’a dan semangat demi terselesaikannya skripsi ini.
5. Seluruh civitas academica Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya yang telah memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini.
6. Seluruh teman – teman Program Studi Sistem Komputer Angkatan 2013 yang selalu memberikan dukungan, semangat, serta bantuan pikiran.
7. Dan seluruh pihak yang telah membantu dan memberikan masukan berharga bagi penulis selama penulis menjalani masa perkuliahan.

Malang, 31 Mei 2018

Falachudin Akbar

falachudin17@gmail.com

ABSTRAK

Denyut jantung dapat menjadi salah satu dasar untuk mengetahui kesehatan dasar. Gejala yang timbul dari denyut jantung yaitu takikardia dan bradikardia, yaitu kondisi tidak normal. Timbulnya gejala yang tidak normal pada denyut jantung dapat mengindikasikan suatu penyakit seperti gagal jantung dan gangguan pernafasan. Oleh karena itu penelitian ini melakukan perancangan alat untuk *monitoring* denyut jantung. Sistem *monitoring* denyut jantung ini mengukur denyut jantung dengan membaca nilai pulse sensor, setelah membaca nilai sensor akan dilakukan proses oleh nodemcu untuk mendapatkan BPM (*Beat per Minute*). Setelah mendapatkan nilai BPM, sistem akan melakukan pengecekan jika lebih dari 100 akan mengirimkan SMS bahaya Takikardia, jika kurang dari 60 akan mengirimkan SMS bahaya bradikardia. Sistem akan mengirimkan data BPM ke *channel thingspeak* menggunakan MQTT. Pada sistem akan selalu bersiap untuk menerima SMS permintaan denyut jantung yang selanjutnya akan membalas SMS dengan nilai BPM. Hasil dari pengujian fungsionalitas sistem dapat berhasil dilakukan dan hasil dari pengujian pembacaan denyut jantung menghasilkan persentase kesalahan sebesar 2.6%. Pada pengujian pengiriman data denyut jantung ke channel Thingspeak, pengiriman SMS peringatan, dan membalas sms dengan nilai denyut jantung terkini berhasil dapat dilakukan.

Kata kunci: denyut jantung, *MQTT*, *sms gateway*, *wearable device*

ABSTRACT

Heart rate is one of the basics to know basic health. Symptoms from the heart rate is tachycardia and bradycardia, that conditions are not normal. The incidence of abnormal symptoms in the heart rate may indicate a heart failed and breathing disorders. This research is designing tools for heart rate monitoring. This heart rate monitoring system measures the heart rate by reading the pulse sensor value, after that nodemcu will process reading heart rate value to get BPM (Beat per Minute). After getting the value of BPM, System check if more than 100 will send SMS danger tachycardia, if less than 60 will send SMS danger bradikardia. System will send BPM data to the thingspeak channel using MQTT. The system will always be ready to receive SMS request heart rate which will reply SMS with BPM value. The results of the system functionality test can be successfully performed and the results of the heart rate reading test resulted in a percentage error of 2.6%. In testing heart rate data transmission to the Thingspeak channel, sending SMS warnings, and replying to sms with the latest heart rate can be successfully done.

Keyword : heart beat , MQTT, SMS gateway, wearable device



DAFTAR ISI

PERSETUJUAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
KATA PENGANTAR.....	iii
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN TEORI	6
2.1 Kajian Pustaka	6
2.2 Dasar Teori.....	7
2.2.1 Takikardia dan Bradikardia.....	7
2.2.2 BPM (<i>beat per minute</i>).....	8
2.2.3 Tanda Vital	8
2.2.4 MQTT (<i>Message Queue Telemetry Transport</i>) Protokol.....	9
2.2.5 <i>Pulse sensor</i>	9
2.2.6 Mikrokontroler NodeMCU	10
2.2.7 Modul GSM Sim800L.....	12
2.2.8 Baterai Li-Po	13
2.2.9 Arduino IDE	14
2.2.10 Thingspeak	15
2.2.11 ThingView.....	16
BAB 3 METODOLOGI PENELITIAN	17

3.1 Metode Penelitian	17
3.2 Landasan Pustaka.....	18
3.3 Rekayasa Kebutuhan Sistem.....	18
3.4 Perancangan dan Implementasi	18
3.5 Pengujian	19
3.6 Penutup.....	19
BAB 4 REKAYASA KEBUTUHAN SISTEM.....	20
4.1 Gambaran Umum Sistem.....	20
4.1.1 Prespektif Sistem.....	21
4.1.2 Ruang Lingkup	21
4.1.3 Karakteristik Pengguna	21
4.1.4 Lingkungan Operasi Sistem.....	21
4.1.5 Asumsi dan ketergantungan	21
4.2 Kebutuhan Fungsional	22
4.3 Kebutuhan Sistem.....	22
4.3.1 Arduino IDE (<i>Intergrated Development Environtment</i>).....	23
4.3.2 <i>Thingspeak</i>	23
4.3.3 <i>Thingview</i>	23
4.3.4 <i>Library</i>	23
4.3.5 <i>AT Command</i>	24
4.4 Kebutuhan Perangkat Keras	24
4.4.1 <i>Pulse Sensor</i>	24
4.4.2 Mikrokontroler NodeMCU	25
4.4.3 Modul GSM Sim 800.....	25
4.4.4 Baterai Li-Po	25
BAB 5 PERANCANGAN DAN IMPLEMENTASI	26
5.1 Diagram Blok Sistem	26
5.2 Perancangan Sistem.....	26
5.2.1 Perancangan Perangkat Keras	27
5.2.2 Diagram Alur Perancangan Perangkat Lunak.....	28
5.2.3 Perancangan Perangkat Lunak.....	29
5.3 Implementasi Sistem	35

5.3.1 Implementasi Perangkat Keras	35
5.3.2 Implementasi Perangkat Lunak.....	37
BAB 6 PENGUJIAN	46
6.1 Pengujian Fungsional Sistem	46
6.1.1 Tujuan Pengujian.....	46
6.1.2 Alat yang Digunakan	46
6.1.3 Prosedur Pengujian	46
6.1.4 Hasil Pengujian	47
6.2 Pengujian Keakuratan Pulse Sensor	49
6.2.1 Tujuan Pengujian.....	49
6.2.2 Alat yang Digunakan	49
6.2.3 Prosedur Pengujian	49
6.2.4 Hasil Pengujian Secara Pembacaan Pulse Sensor	50
6.3 Pengujian Pengiriman Data ke <i>Channel</i> Thingspeak	51
6.3.1 Tujuan Pengujian.....	51
6.3.2 Alat yang Digunakan	51
6.3.3 Prosedur Pengujian	51
6.3.4 Hasil Pengujian Pengiriman Data ke <i>Channel</i> Thingspeak.....	51
6.4 Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung	53
6.4.1 Tujuan Pengujian.....	53
6.4.2 Alat yang Digunakan	53
6.4.3 Prosedur Pengujian	53
6.4.4 Hasil Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung.....	53
6.5 Pengujian Balasan SMS dengan karakter BPM	55
6.5.1 Tujuan Pengujian.....	55
6.5.2 Alat yang Digunakan	55
6.5.3 Prosedur Pengujian	55
6.5.4 Hasil Pengujian	55
BAB 7 PENUTUP	57
7.1 Kesimpulan.....	57
7.2 Saran	57
Daftar Pustaka.....	58





DAFTAR GAMBAR

Gambar 2.1 ECG kondisi normal, bradikardia, dan takikardia.....	7
Gambar 2.2 Mengukur Tanda Vital Denyut Jantung	8
Gambar 2.3 Arsitektur MQTT.....	9
Gambar 2.4 <i>Pulse sensor</i>	10
Gambar 2.5 Kaki <i>Pulse sensor</i>	10
Gambar 2.6 Komponen NodeMCU	11
Gambar 2.7 Komponen Sim800L	12
Gambar 2.8 Komponen Baterai Li-Po.....	13
Gambar 2.9 Arduino IDE	14
Gambar 2.10 Thingspeak	15
Gambar 2.11 ThingView.....	16
Gambar 3.1 Diagram Alir Metode Penelitian.....	17
Gambar 3.2 Diagram Blok Perancangan Sistem	18
Gambar 4.1 Gambaran Umum Sistem	20
Gambar 5.1 Diagram Blok Sistem	26
Gambar 5.2 Skematik Perancangan Perangkat Keras.....	27
Gambar 5.3 Diagram Alur Perancangan Perangkat Lunak.....	28
Gambar 5.4 Flowchart Koneksi ke Wifi.....	29
Gambar 5.5 Flowchart Koneksi MQTT	30
Gambar 5.6 Flowchart Pembacaan <i>Pulse sensor</i>	31
Gambar 5.7 Flowchart Pengiriman Nilai BPM ke Channel Thingspeak	32
Gambar 5.8 Flowchart Pengiriman SMS Peringatan.....	33
Gambar 5.9 Flowcart Membalas SMS dengan Nilai BPM	34
Gambar 5.10 Implementasi Perangkat Keras	35
Gambar 5.11 Implementasi Purwarupa Tampak Atas.....	36
Gambar 5.12 Implementasi Purwarupa Tampak Bawah	36
Gambar 6.1 Hasil Pengujian Fungsionalitas Aplikasi Android.....	47
Gambar 6.2 Hasil Pengujian Permintaan Nilai BPM	48
Gambar 6.3 Hasil Pengujian Pengiriman SMS peringatan	48

DAFTAR TABEL

Tabel 2.1 Karakteristik ESP8266.....	11
Tabel 2.2 Spesifikasi Baterai Li-Po.....	13
Tabel 5.1 Hubungan Pin antar Komponen.....	27
Tabel 5.2 Kode Sumber Nodemcu koneksi ke WiFi	37
Tabel 5.3 Kode Sumber Nodemcu Koneksi ke MQTT Broker (Thingspeak)	38
Tabel 5.4 Kode Sumber Pembacaan Pulse Sensor	39
Tabel 5.5 Kode Sumber Pengiriman Nilai BPM ke Channel Thingspeak	40
Tabel 5.6 Kode Sumber Pengiriman SMS Peringatan	41
Tabel 5.7 Kode Sumber Membalas SMS dengan Nilai BPM.....	42
Tabel 6.1 Hasil Pengujian Pembacaan Pulse Sensor	50
Tabel 6.2 Hasil Pengujian Pengiriman Data ke <i>Channel</i> Thingspeak	52
Tabel 6.3 Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung	54
Tabel 6.4 Hasil Pengujian Balasan SMS dengan karakter BPM.....	56



BAB 1 PENDAHULUAN

1.1 Latar belakang

Untuk menjaga agar tubuh bisa optimal, kita harus menjaga kesehatan. Kesehatan menjadi salah satu prioritas utama, karena kesehatan adalah aset berharga dalam hidup. Jika kesehatan terganggu maka akan mengalami penurunan produktifitas (Ali, 2009). Oleh karena itu kita harus menjaganya dari penyakit. Ada beberapa cara untuk menjaga kesehatan, salah satunya *monitoring* kondisi tubuh. Kondisi tubuh dapat kita lihat melalui kondisi tanda vital, karena itu mencerminkan aspek dasar kesehatan seseorang. (Manurung, 2017)

Tanda vital adalah suatu pengukuran fungsi tubuh yang paling dasar untuk mengetahui tanda klinis dan mengetahui suatu penyakit. Ada 4 tanda vital untuk bisa mengetahui kesehatan, yaitu pernafasan, denyut jantung, suhu tubuh, dan tekanan darah. Perubahan tanda vital dapat mengidentifikasi kebutuhan untuk dilakukan penanganan medis selanjutnya. Pengukuran tanda – tanda vital secara periodik merupakan cara yang cepat dan efisien untuk memantau perkembangan kondisi kesehatan fisik dan mengindikasi suatu gejala penyakit secara dini. (Morton, 2005)

Banyak sekali orang mengabaikan gejala yang mengakibatkan bahwa dirinya sedang terindikasi suatu gejala penyakit. Oleh karena itu, Pengecekan tanda vital secara teratur sangat dianjurkan untuk pemeriksaan diri dari sebuah penyakit. Salah satu pemeriksaan dini dengan tanda vital, yaitu pemeriksaan informasi denyut jantung. Ada beberapa yang bisa kita ukur untuk mengetahui gejala saat kita melakukan pemeriksaan denyut jantung, yaitu gejala takikardi dan bradikardi. (Gray, 2005)

Gejala Takikardia adalah gejala yang terjadi saat denyut jantung berdetak dengan cepat, yaitu melebihi 100 BPM. Gejala ini biasanya dapat terjadi pada saat orang yang tekanan intrakarnial meningkat, dan efek samping beberapa obat. Tekanan intrakarnial adalah mengalami peningkatan tekanan otak normal. Gejala Bradikardia adalah gejala yang terjadi saat denyut jantung berdetak dengan lambat, yaitu kurang dari 60 BPM. Gejala ini biasanya terjadi pada saat orang yang sedang demam, *congentif heart failure*, syok. Dari dua kondisi gejala tersebut memiliki resiko yang berbahaya karena dapat mengakibatkan kematian. Terlebih lagi itu adalah salah satu organ penting dalam tubuh manusia. Oleh karena itu pengecekan denyut jantung sangatlah penting guna pendeteksian penyakit. (Gray, 2005)

Ada beberapa cara pada saat ini untuk mendeteksi denyut jantung, yaitu dengan stetoskop dan manual menggunakan tangan. Petugas medis menggunakan alat stetoskop untuk mendeteksi denyut jantung yang diletakkan pada dada orang. Jika dengan cara manual yaitu dengan meletakkan jari telunjuk dan jari tengah diletakkan pada pergelangan tangan. Pergelangan tangan terdapat

Arteri Radialis yaitu, tempat untuk menghitung tekanan denyut jantung. Tempat mendeteksi denyut jantung yaitu pergelangan bagian depan sebelah bawah pangkal ibu jari tangan (*Arteri radialis*), dileher sebelah kiri/kanan depan otot *sterno cleido mastoidues* (*Arteri carolis*), dada sebelah kiri tepat di *apex* jantung (*Arteri temporalis*) dan pelipis. (Muffichatum, 2006)

Dengan perkembangan teknologi yang pesat, setiap orang memiliki *smartphone*. *Smartphone* menjadi alat untuk mendapatkan informasi. Salah satu informasi yang dapat diterima yaitu informasi kesehatan. Informasi kesehatan dapat dilakukan dengan *monitoring* kesehatan. Dengan mendapatkan *monitoring* kesehatan pada *smartphone* dapat menjaga kesehatan dengan mudah. (McGrath, 2013). Untuk *monitoring* kesehatan dapat dilakukan dengan menggunakan *smartphone*. Ada beberapa cara untuk mengirimkan data ke *smartphone* yaitu dengan komunikasi *Bluetooth* atau WiFi, namun pengiriman data dengan menggunakan komunikasi *Bluetooth* atau WiFi dilakukan secara offline. Untuk dapat melakukan *Monitoring* secara online dapat dilakukan dengan menggunakan protokol MQTT. Untuk dapat melakukan pengiriman protokol MQTT membutuhkan *broker* MQTT sebagai penampung agar dapat dilakukan penyimpanan pada server dan dapat ditampilkan melalui *smartphone*. Selain itu untuk mendapatkan informasi kesehatan dapat dilakukan dengan cara menerima SMS (*Short Messaging Service*) peringatan dan status kondisi kesehatan. Mengirimkan informasi kesehatan dengan menggunakan SMS dapat menjadi solusi terbaik dengan biaya yang murah. (Moon, 2015)

Berdasarkan permasalahan dan kondisi saat ini yang telah dijabarkan di latarbelakang tersebut, penulis mengusulkan judul skripsi “Sistem *Monitoring* Denyut Jantung Menggunakan NodeMCU dan MQTT”. Sistem ini menggunakan NodeMCU sebagai mikrokontroler, *pulse sensor* sebagai sensor untuk membaca denyut jantung, menggunakan protokol MQTT untuk *monitoring* secara online serta menggunakan modul GSM sim800 untuk mengirimkan SMS peringatan dan fitur mengirimkan balasan bpm jika diminta dengan sms.

Dalam penelitian sebelumnya yang dilakukan ikhsan *monitoring* denyut jantung dilakukan menggunakan *smartphone* melalui *Bluetooth*. Pengiriman oleh *Bluetooth* hanya dapat dilihat hasilnya jika sistem tersebut dekat dengan *smartphone* yang terkoneksi oleh *Bluetooth*. Oleh karena itu sistem ini menggunakan protokol mqtt digunakan sebagai sarana untuk mengirimkan data secara online sehingga dapat *monitoring* melalui *smartphone*. Protokol MQTT perlu menggunakan *broker*, oleh karena itu penelitian ini menggunakan *broker* yang disediakan oleh *Thingspeak* untuk bisa menampilkannya secara online.

Dengan sistem *monitoring* ini diharapkan masyarakat untuk bisa melakukan pengecekan denyut jantung dengan mudah, agar tidak terjadi terlambatnya penanganan suatu penyakit atau gejala yang tidak normal. Karena salah satu faktor yang menyebabkan penanganan yang terlambat ialah mengabaikan gejala – gejala awal yang ditimbulkan meskipun gejala – gejala tersebut tidak bisa langsung memvonis bahwa orang tersebut mengidap suatu penyakit tetapi alangkah baiknya jika memeriksakan segera ketika mengalami gejala – gejala awal

tersebut (Hanafy, 2013). Karena pengecekan sejak dini dapat membantu langkah medis selanjutnya yang akan dilakukan.

1.2 Rumusan masalah

Rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana merancang dan membuat sistem *monitoring* denyut jantung?
2. Bagaimana mengolah sinyal data dari *pulse sensor* menggunakan NodeMCU?
3. Bagaimana penggunaan protokol MQTT dalam mengirimkan data ke *channel Thingspeak*?
4. Bagaimana penggunaan Modul GSM Sim 800 dalam mengirimkan dan membalas SMS menggunakan?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Merancang dan membuat sebuah sistem yang bisa *monitoring* denyut jantung.
2. Mengolah sinyal data *pulse sensor* untuk mendeteksi denyut jantung yang akan di proses menggunakan NodeMCU.
3. Menggunakan protokol MQTT untuk *monitoring* menggunakan *channel Thingspeak* yang dapat dilihat hasilnya melalui *smartphone*.
4. Menggunakan modul GSM untuk mengirimkan SMS ke nomer tertentu dan membalas SMS nilai bpm jika mendapatkan SMS dengan karakter tertentu.

1.4 Manfaat

Manfaat yang diperoleh dari penerapan sistem *monitoring* denyut jantung menggunakan NodeMCU dan MQTT antara lain:

1. Memudahkan pengguna untuk *monitoring* denyut jantung pada seseorang.
2. Memudahkan pengguna untuk memantau denyut jantung dari waktu ke waktu secara online.
3. Menjadi sarana edukasi dalam pemanfaatan NodeMCU dan MQTT.

1.5 Batasan masalah

Batasan masalah pada penelitian ini adalah:

1. Denyut jantung dimonitor menggunakan *pulse sensor* dengan menggunakan pergelangan tangan sebagai tempat untuk mendeteksinya.
2. Fokus penelitian adalah tersampainya data denyut jantung ke *channel Thingspeak* menggunakan MQTT, mengirimkan SMS jika kondisi takikardia atau bradikardia dan membalas sms jika ada permintaan nilai BPM(*Beat per Minute*).

1.6 Sistematika pembahasan

Pembahasan dalam skripsi ini akan terbagi menjadi 6 bab, sub bab penelitian dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

Bab ini menjelaskan informasi tentang latar belakang dari masalah yang diangkat yaitu masalah kesehatan jantung yang perlu dipantau, rumusan masalah dijabarkan untuk menjabarkan masalah yang perlu diselesaikan dalam penelitian ini, batasan masalah dijabarkan untuk memberi batasan dari penelitian ini agar memiliki ruang lingkup, manfaat dijabarkan dengan harapan penelitian ini bisa bermanfaat, tujuan dijabarkan agar mendapatkan tujuan dari masalah dan solusi yang dilakukan dan sistematika pembahasan dari penelitian skripsi dijabarkan untuk mengetahui isi secara umum setiap bab.

BAB II Landasan Kepustakaan

Bab ini membahas tentang penelitian yang pernah ada sebelumnya dan teori yang mendukung penelitian skripsi. Akan dijelaskan bagaimana skripsi sebelumnya secara singkat, dari mulai cara kerja, alat komponen yang dipakai dan bagaimana hasil penelitian sebelumnya.

BAB III Metode Penelitian

Pada bab ini akan membahas metode dan langkah yang akan digunakan pada penelitian skripsi. Didalamnya terdapat metode penelitian yang akan menjelaskan tentang bagaimana langkah-langkah penelitian ini dilakukan. Landasan pustaka menjelaskan tentang referensi topik utama untuk menunjang penelitian.

BAB IV Rekayasa Kebutuhan Sistem

Pada bab ini akan membahas kebutuhan sistem. Kebutuhan sistem tersebut yaitu gambaran umum sistem akan dijelaskan mengenai bagaimana gambaran umum dari sistem yang dilakukan pada penelitian ini, kebutuhan fungsional dijabarkan untuk mengetahui fungsional utama dari sistem, kebutuhan sistem dijabarkan untuk mengetahui kebutuhan sistem dalam penelitian ini, kebutuhan perangkat keras dijabarkan untuk mengetahui kebutuhan perangkat keras dalam sistem, dan kebutuhan perangkat lunak dijabarkan untuk mengetahui kebutuhan perangkat lunak dalam sistem ini.

BAB V Perancangan dan Implementasi

Pada bab ini menguraikan proses implementasi dari dasar teori yang telah dipelajari sesuai dengan perancangan sistem. Pada perancangan sistem akan di bagi menjadi dua perancangan perangkat keras dan perancangan perangkat lunak, dan untuk implementasi akan menjelaskan implementasi perangkat keras dan perangkat lunak.

BAB VI Pengujian

Memuat hasil dari pengujian dari implementasi sistem yang telah dibuat. Disetiap pengujian akan menjelaskan tujuan pengujian, alat yang digunakan, prosedur pengujian, dan hasil pengujian. Pada pengujian ini akan melakukan 5 topik pengujian, yaitu pengujian fungsional sistem, pengujian pembacaan sensor, pengujian pengiriman data bpm ke *channel* thingspeak, pengujian pengiriman sms peringatan, dan pengujian balasan sms dengan karakter “BPM”.

BAB VII Penutup

Berisi kesimpulan atas penelitian yang telah diuji dan dianalisis serta saran untuk pengembangan sistem selanjutnya. Kesimpulan akan diambil berdasarkan rumusan masalah, tujuan dan pengujian yang telah dilakukan,.



BAB 2 LANDASAN TEORI

Pada penelitian ini pustaka yang dijadikan referensi adalah penelitian berjudul “*A microcontroller-based automatic heart rate counting system from fingertip*” oleh Mamun et al dan “Implementasi *Low Power Wearable Perangkat Sebagai Heart Rate Monitor* dengan Metode *State Machine*” oleh Ihsanurrahim. Dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah konsep dasar penjelasan tentang tanda vital, mikrokontroler Node MCU, *pulse sensor*, MQTT Protokol, serta buzzer dan pemrograman pada mikrokontroler.

2.1 Kajian Pustaka

Berikut ini ada penelitian sebelumnya yang terkait dengan sistem *monitoring* denyut jantung dengan menggunakan perhitungan denyut jantung berdasarkan *microcontroller-based* dan penelitian yang mendeteksi denyut jantung secara *low power wearable* perangkat yang ditampilkan pada *smartphone* dengan menggunakan Bluetooth.

Penelitian sebelumnya yang berjudul “*A microcontroller-based automatic heart rate counting system from fingertip*” oleh Mamun et al. Pada penelitiannya ini membahas sebuah sistem yang dapat melakukan penghitungan terhadap denyut jantung seseorang dengan meletakkan jarinya pada sensor tersebut dan hasilnya akan ditampilkan pada LCD. Sensor yang digunakan pada sistem ini terdiri dari infra merah dan *photodiode*. Cara kerja dari sensornya ialah dengan memancarkan sinar infra merah pada jari yang telah diletakkan di atasnya kemudian *photodiode* akan menangkap sinar pantulnya untuk kemudian merubahnya ke data digital agar dapat ditampilkan pada LCD sebagai data dari denyut jantung.

Selanjutnya penelitian dari Ihsanurrahim pada Gambar 2.2, yaitu “Implementasi *Low Power Wearable Perangkat Sebagai Heart Rate Monitor* dengan Metode *State Machine*”. Penelitian ini memonitor denyut jantung menggunakan metode *state machine* yang akan ditampilkan pada *smartphone* menggunakan Bluetooth. Pada sistem ini menggunakan metode *state machine* untuk menghasilkan konsumsi daya yang hemat. Hasil dari purwarupa sistem menghasilkan persentasi kesalahan 3.97% pada keadaan normal dan 7.91% pada keadaan olahraga. Uji kinerja sistem untuk mengirimkan data menggunakan Bluetooth memiliki efektif hingga 12-meter.

Dari dua penelitian diatas “*A microcontroller-based automatic heart rate counting system from fingertip*” oleh Mamun et al, yang dapat memonitor denyut jantung menggunakan LCD dan “Implementasi *Low Power Wearable Perangkat Sebagai Heart Rate Monitor* dengan Metode *State Machine*” oleh Ihsanurrahim menggunakan Bluetooth untuk memonitor denyut jantung pada *smartphone*.

Oleh karena itu, penelitian ini ingin mengembangkan sistem ini menggunakan protokol mqtt dan modul GSM. Dengan protokol mqtt, sistem ini dapat menampilkan hasil pengukuran denyut jantung secara online dengan

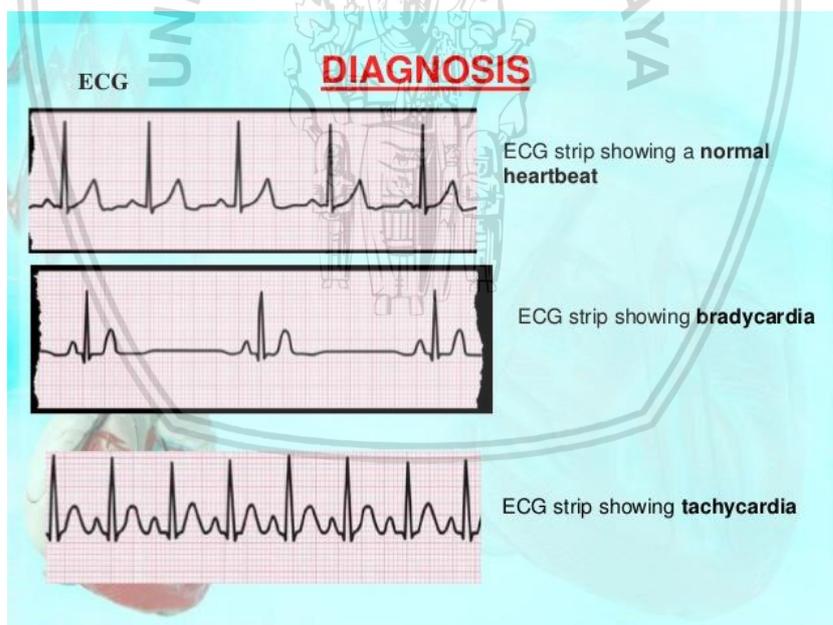
menggunakan smartphone. Monitoring dengan menggunakan protokol mqtt secara online ini dapat diaplikasikan jika pengguna sistem ini dapat menyimpan data kesehatan dan dapat dianalisis oleh dokter tanpa dokter perlu berada didekat pengguna sistem ini. Fitur tambahan selanjutnya yaitu menggunakan modul GSM. Modul GSM akan mengirmkan SMS peringatan dan membalas SMS nilai bpm jika menerima SMS dengan karakter tertentu.

2.2 Dasar Teori

Dalam subbab ini akan dijelaskan mengenai dasar teori yang digunakan dalam penelitian ini. Dasar Teori meliputi Takikardia dan Bradikardia, BPM(*beat per minute*), tanda vital, MQTT, pulse sensor, mikrokontroler nodemcu, modul GSM Sim800l, baterai li-po, arduino ide, thingspeak, thingview.

2.2.1 Takikardia dan Bradikardia

Takikardia dan Bradikardia adalah kondisi yang dapat dilihat dari denyut jantung. Kedua gejala tersebut terjadi jika ada kelainan pada denyut jantung atau denyut jantung tidak normal. Denyut jantung yang normal adalah 60-100 kali permenit. Jika denyut jantung kita per menit diatas 100 *beat per minute* berarti mengalami kondisi Takikardia, sedangkan jika dibawah 60 *beat per minute* berarti mengalami kondisi bradikardia (Delp, 1996).



Gambar 2.1 ECG kondisi normal, bradikardia, dan takikardia

Sumber : (Abjali.c, 2015)

Takikardia dapat dilihat dengan meningkatnya denyut jantung pada ruang bawah jantung, ruang atas jantung atau keduanya. Denyut jantung dikontrol oleh sinyal elektrik yang dikirim ke seluruh jaringan jantung. Ketika produksi sinyal dipercepat maka akan timbul takikardia. Sebaliknya dengan kondisi bradikardia jika produksi sinyal lambat akan timbul kondisi bradikardia. Kondisi Takikardia dan

Bradikardia dapat hilang bila seseorang beristirahat. Namun bila kondisi itu tetap berlangsung walaupun sudah beristirahat maka perlu melakukan pemeriksaan lebih lanjut karena itu bisa menjadi indikasi terhadap kelainan jantung. Berikut pada Gambar 2.1 menjelaskan perbedaan denyut jantung normal, gejala takikardia, dan gejala bradikardia.

2.2.2 BPM (*beat per minute*)

BPM(*beat per minute*) adalah satuan dari denyut jantung. Satuan ini berfungsi untuk melihat kondisi ritme denyut jantung. Ritme denyut jantung memiliki 3 kondisi yaitu, normal, takikardia, dan bradikardia. Untuk melakukan perhitungan BPM dengan cara manual yaitu menggunakan pergelangan tangan dan *stopwatch*. Langkahnya dengan cara meraba pergelangan tangan kanan dengan tangan kiri, rasakan adanya denyutan tangan kiri, hitung denyutan itu bersama dengan mulainya waktu pada *stopwatch*, setelah melakukan perhitungan denyut selama 60 detik, akumulasikan dan itulah nilai BPM. (Sari, 2016). Berikut pada Gambar 2.2 menjelaskan pembacaan denyut jantung secara manual dengan meletakkan jari tangan pada pergelangan tangan.



Gambar 2.2 Mengukur Tanda Vital Denyut Jantung

Sumber : (Mikrajuddin, 2007)

2.2.3 Tanda Vital

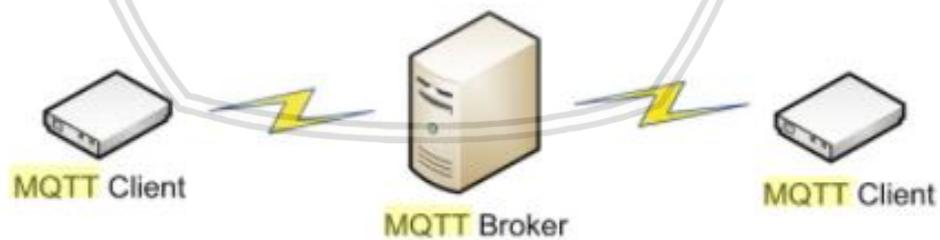
Tanda vital merupakan gabungan dua kata, yaitu tanda dan vital, yang merupakan terjemahan istilah dari bahasa Inggris yaitu *vital sign*. *Vital sign* adalah suatu tanda yang sifatnya objektif yang dapat berubah setiap saat yang mencerminkan hidup yang terdiri dari tekanan darah, respirasi, nadi dan suhu tubuh. Perubahan tanda vital dapat mengidentifikasi kebutuhan untuk dilakukan intervensi keperawatan dan medis.

Pengukuran tanda – tanda vital yang periodik merupakan cara yang cepat dan efisien untuk memantau perkembangan kondisi kesehatan fisik dan mengevaluasi respon terhadap intervensi keperawatan dan medis yang dilakukan. Pada semua kasus, termasuk persalinan pengukuran tanda – tanda vital adalah mencakup pengukuran tekanan darah, nadi, suhu tubuh dan pernapasan (Morton, 2005).

2.2.4 MQTT (Message Queue Telemetry Transport) Protokol

MQTT dikembangkan oleh Andy Stanford-Clark (IBM) dan Arlen Nipper (Eurotech; sekarang Cirrus Link) pada tahun 1999 untuk melakukan *monitoring* pipa minyak yang melalui gurun (García, 2018). Dengan tujuan untuk mempunyai sebuah protokol, dengan penggunaan bandwidth yang efisien serta konsumsi daya baterai yang kecil, karena perangkat terhubung dengan satelilite link dan dengan biaya yang sangat mahal pada saat itu.

MQTT (*Message Queue Telemetry Transport*) adalah protokol pesan ringan (*lightweight*) berbasis *publish-subscribe* digunakan di atas protokol TCP/IP atau WebSockets. Cara kerja dari protokol ini terdiri dari *publisher* dan *subscriber*. *Publisher* adalah sebuah perangkat yang selalu mengirim data menuju server atau langsung menuju *subscriber*. Sementara *subscriber* merupakan sebuah piranti yang melaksanakan *request* atau menerima pesan dari server atau *publisher*. MQTT bersifat terbuka, simpel dan didesain agar mudah diimplementasikan. Karakteristik ini membuatnya ideal untuk digunakan dalam banyak situasi, termasuk lingkungan terbatas seperti dalam konteks Internet of Things (IOT) dimana sistem membutuhkan jaringan yang terbatas. Pola pesan publish subscribe membutuhkan *broker* yang bertanggung jawab dalam mengirimkan atau mendistribusikan pesan kepada client dan server (García, 2018). *Broker* berada diantara *publisher* dan subsriber. Berikut pada Gambar 2.3 menjelaskan arsitektur MQTT.



Gambar 2.3 Arsitektur MQTT

Sumber : (Hassanien, 2018)

2.2.5 Pulse sensor

Sensor denyut jantung adalah perangkat keras untuk membaca denyut jantung (Bronzino, 2000). Pembacaan sensor memiliki dua tipe sensor, yaitu tipe sensor digital dan tipe sensor analog. Untuk pembacaan denyut jantung dibutuhkan nilai *threshold* untuk menentukan satu denyut jantung. Satu denyut

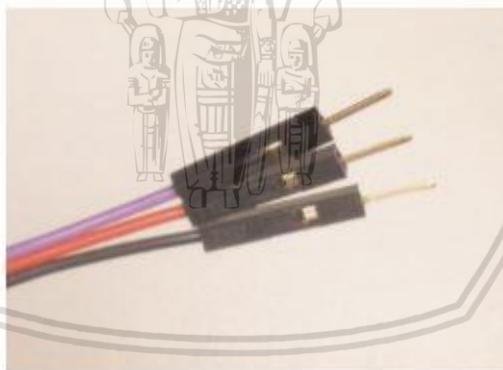
jantung dibaca saat pantulan sinar dari cahaya yang dipancarkan oleh sensor berubah intensitasnya. Saat berubah intensitasnya dapat diketahui itu sebuah satu denyut jantung atau bukan. Pengiriman data analog sensor denyut jantung menggunakan kabel ke mikrokontroler. Untuk sensor analog, pada mikrokontroler terdapat pin khusus untuk membaca sinyal yang dikirimkan sensor analog.

Pulse sensor merupakan sensor yang telah dirancang agar mudah digunakan oleh siapapun untuk mengukur banyaknya denyut jantung per satuan menit. Sensor ini bekerja dengan memancarkan sinar LED-nya pada bagian jari lalu cahaya yang dipantulkan ditangkap kembali oleh sensor cahaya (photodiode) untuk mendapatkan nilai kecerahan cahaya yang dipantulkan (World Famous Electronics llc, 2015). Berikut ini penampilan fisik dari *pulse sensor* terdapat pada Gambar 2.4



Gambar 2.4 *Pulse sensor*

Sumber: (World Famous Electronics llc, 2015)



Gambar 2.5 Kaki *Pulse sensor*

Sumber: (World Famous Electronics llc, 2015)

Pada Gambar 2.5 terdapat 3 kaki dari *pulse sensor*, kaki yang berwarna merah berfungsi sebagai sumber tegangan antara 3 Volt sampai 5 Volt, lalu kaki yang berwarna hitam merupakan ground dan kaki yang berwarna ungu adalah sinyal keluaran dari sensor tersebut.

2.2.6 Mikrokontroler NodeMCU

Mikrokontroler Nodemcu merupakan sebuah perangkat keras yang akan memproses dan menjalankan *code* program yang telah dibuat. Program yang telah diunggah di mikrokontroler akan dieksekusi sesuai setiap baris *code* program yang

dibuat. Untuk bisa membaca denyut jantung membutuhkan *pulse sensor* yang dimana sensor tersebut merupakan sensor analog, maka dalam sistem ini membutuhkan mikrokontroler yang spesifikasinya dapat membaca sensor analog. Sistem ini juga membutuhkan modul Wifi untuk dapat mengirmkan data secara online.

Mikrokontroler NodeMCU merupakan mikroprosesor open soucre yang berjalan pada Esp8266 WiFi SoC dari Espresssif System dan perangkat keras yang berbasis ESP12 module sehingga mendukung koneksi WiFi (Schwartz, 2017). NodeMCU dapat dihubungkan dengan komputer menggunakan kabel USB dan dapat diprogram dengan mudah karena NodeMCU kompatible dengan dua program editor, yaitu LUA dan Arduino IDE serta dapat diprogram dengan bahasa pemrograman C atau C++ yang di custom menjadi sebuah software khusus sebagai platform pemrograman. NodeMCU memiliki keunggulan sendiri dibanding dengan mikrokontroler lainnya, yaitu dari segi bentuk yang lebih kecil serta memiliki fungsi yang lebih lengkap dibanding Arduino Uno seperti kemampuan untuk terhubung ke jaringan *wireless* tanpa harus ada perangkat tambahan. Tampilan fisik NodeMCU seperti pada Gambar 2.6



Gambar 2.6 Komponen NodeMCU

Sumber: (<https://www.seeedstudio.com>)

Pada penelitian ini NodeMCU diprogram dengan bahasa C menggunakan *editor* Arduino IDE. Dengan teknologi *Low Power*, perangkat ini dapat aktif pada tegangan 3.3V. Beirkut Tabel 2.1 adalah spesifikasi lengkap dari NodeMCU:

Tabel 2.1 Karakteristik ESP8266

Parameter	Nilai
Protokol WiFi	802.11 b/g/n
Frekuensi Kerja	2.4G-2.5G (2400M-2483.5M)
Tegangan Operasi	3.0V-3.6V
Arus Operasi Sekitar	80mA



Protokol Jaringan	Ipv4, TCP/UDP/HTTP/FTP
Mode WiFi	Station/ softAP/ SoftAP+station

Sumber: (<https://www.nodemcu.com>)

Esp8266EX merupakan WiFi Chips yang paling bisa diintegrasikan dalam dunia industri. Dengan ukuran hanya 5mm x 5mm, Esp8266EX hanya memerlukan sedikit *circuit external* dan terintegrasi dengan Tensilica MCU 32-bit. Pada penelitian ini, Perangkat Esp8266 yang dimanfaatkan telah terhubung terhadap mikrokontroler NodeMCU. Esp8266 pada penelitian berfungsi sebagai modul WiFi tambahan di mikrokontroler agar dapat terhubung langsung dengan koneksi jaringan WiFi yang tersedia. Modul ini memiliki tiga mode WiFi, yaitu *station*, *access point* dan *both* (keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO.

2.2.7 Modul GSM Sim800L

Modul GSM Sim 800L merupakan modul GSM yang dibuat salah satunya untuk mengirimkan SMS (Choudhuri, 2017). Modul ini hanya berfungsi dengan kartu perdana GSM. Sim800L termasuk modul GSM jaringan *quad-band* yang dapat beroperasi dalam frekuensi GSM850MHz, EGSM900MHz, DCS100MHz, dan PCS1900MHz. Ukuran dari modul ini tergolong kecil yaitu 15,8*17,8*2,4mm. Untuk mendapatkan sinyal yang kuat modul ini menggunakan *antenna pad* yang mampu menjangkau jaringan lebih optimal. SIM card yang digunakan menggunakan SIM card bertipe micro SIM.

Ada beberapa pin dalam modul ini, yaitu vcc, gnd, tx, rx, reset, speaker+, speaker-, mic+, mic-, dtr dan yang terakhir ring. Pin vcc dan gnd berfungsi untuk catu daya modul GSM sim 800. Pin tr dan tx berfungsi untuk mengirimkan perintah agar bisa melakukan aksi terhadap modul GSM sim 800. Pin reset digunakan untuk mereset modul GSM sim 800. Pin speaker+ dan speaker – berfungsi untuk menggunakan fungsi speaker. Pin mic+ dan mic- berfungsi untuk menggunakan fungsi mic. Kebutuhan catu daya modul GSM sim 800 yaitu minimal 3,4 V dan maksimal 4.4 V. Berikut pada Gambar 2.7 menampilkan komponen SIM800L



Gambar 2.7 Komponen Sim800L

2.2.8 Baterai Li-Po

Sistem *monitoring* denyut jantung menggunakan nodemcu dan mqtt bersifat *wearable*. Sistem yang bersifat *wearable* membutuhkan baterai yang bisa dibawa kemana saja dan bentuknya kecil. Oleh karena itu. Baterai Li-Po dibutuhkan oleh sistem sebagai catu daya. Berikut pada Gambar 2.8 menampilkan gambar dari komponen baterai Li-Po.



Gambar 2.8 Komponen Baterai Li-Po

Baterai Li-Po memiliki kepanjangan dari *lithium polimer*. Ada beberapa cara kelebihan utama jika menggunakan baterai Li-Po, yaitu baterai Li-Po memiliki berat yang ringan dan tersedia dalam berbagai bentuk, baterai Li-Po memiliki kapasitas yang beragam, dari kebutuhan bentuk yang kecil sampai bentuk yang besar tergantung dari ukuran volt dan ampere (Bartsch, 2017). Berikut Tabel 2.2 spesifikasi baterai li-po yang digunakan pada penelitian ini.

Tabel 2.2 Spesifikasi Baterai Li-Po

Parameter	Nilai
Kapasitas	500 mAh
<i>Continuous Discharge Rate</i>	25C
<i>Voltage</i>	3.7V

2.2.9 Arduino IDE

Perangkat lunak ini adalah sesuatu perangkat lunak yang berfungsi sebagai tempat untuk menulis program yang dibutuhkan sistem. Selain berfungsi sebagai tempat untuk menulis, perangkat ini berfungsi untuk kompilasi program dan mengunggahnya ke mikrokontroler (Dennis, 2015). Didalam IDE juga terdapat *library* yang bisa digunakan untuk menunjang kebutuhan sistem. *Library* tersebut bisa sudah ada IDE ataupun kita bisa mengimpor dari sumber lain. *Library* merupakan suatu kumpulan set program yang dapat membantu pengembang untuk membuat *code* program.



Gambar 2.9 Arduino IDE

Untuk menggunakan perangkat lunak ini kita hanya menuliskan *code* set program yang biasanya satu baris kode program. *Library* berisi set dari beberapa fungsi tertentu yang berisi baris program. Perangkat ini sangat membantu pengembang untuk merancang programnya dan dapat memangkas waktu dalam membuat program. Arduino IDE dapat digunakan pada sistem operasi Microsoft Windows dan macOS. Versi Arduino yang digunakan pada penelitian ini menggunakan versi 1.8.5.

2.2.10 Thingspeak

Thingspeak adalah perangkat lunak yang dapat menampung data yang akan dikirim oleh sistem secara online (Molloy, 2016). Aplikasi ini memiliki beberapa fitur diantaranya, menampung data yang dikirimkan oleh *device* yang terkoneksi dengan internet, menganalisis data yang telah dikirim oleh *device*, memonitor data yang telah dikirim oleh *device*, dan mengontrol setiap *device* yang terkoneksi dengan Thingspeak



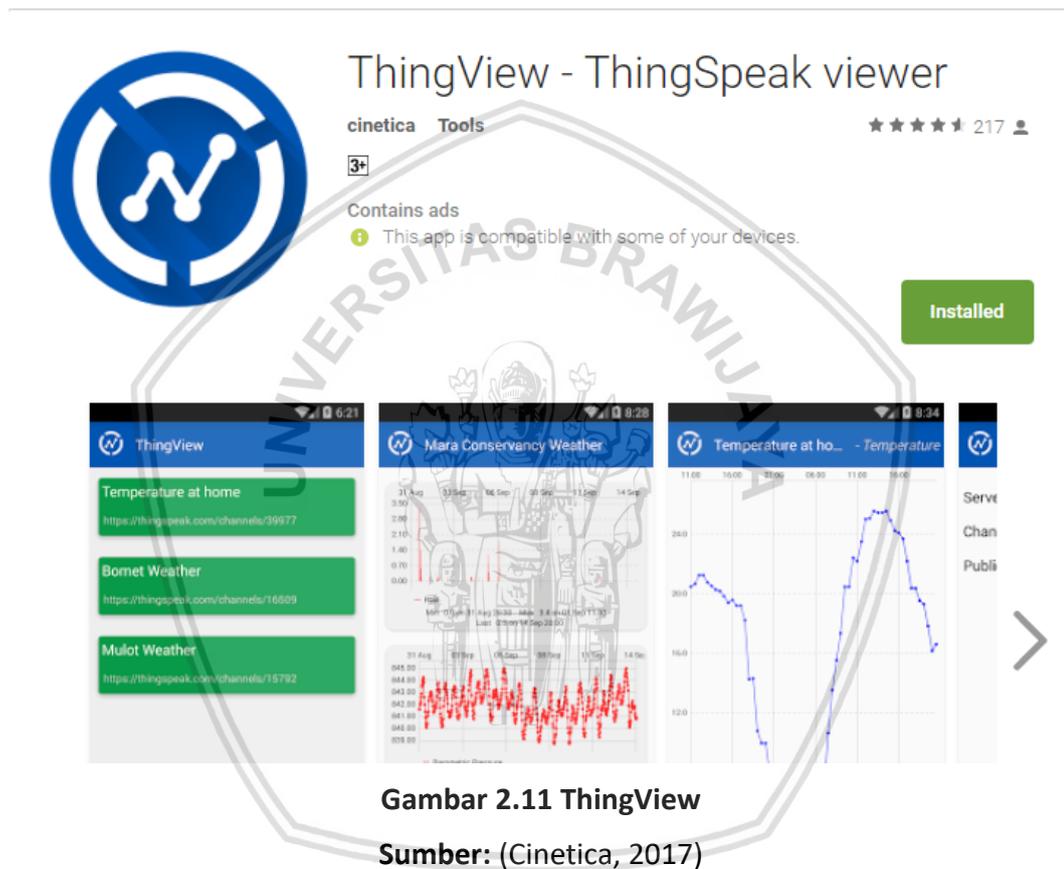
Gambar 2.10 Thingspeak

Sumber: (<https://thingspeak.com/>)

Pada penelitian ini Thingspeak digunakan sebagai *broker* MQTT. Dalam perangkat lunak ini kita dapat melihat hasil data secara grafik, dan kita bisa menyimpan data yang telah masuk di dalam website ini. Dalam website ini kita meletakkan data pada *Channel Thingspeak*. Didalam *Channel Thingspeak* kita bisa mengatur berapa *field* yang dibutuhkan, melihat *API Write* untuk digunakan untuk kunci pengiriman data.

2.2.11 ThingView

Thingview adalah aplikasi perangkat lunak berbasis android yang dibuat oleh developer cinetica ini. Aplikasi ini berfungsi untuk membaca data yang terdapat di *channel* thingspeak. Aplikasi ini dapat diunduh secara gratis di playstore google. Terdapat beberapa fitur yang dapat dilakukan dengan aplikasi ini yaitu, membaca lebih dari 1 *channel* Thingspeak, dapat menampilkan data secara grafik. Thingview memiliki ukuran yang tidak terlalu besar yaitu, 4.5Mb. Aplikasi ini telah diunduh sebanyak 10.000 pada Play Store (Cinetica, 2017).



Gambar 2.11 ThingView

Sumber: (Cinetica, 2017)

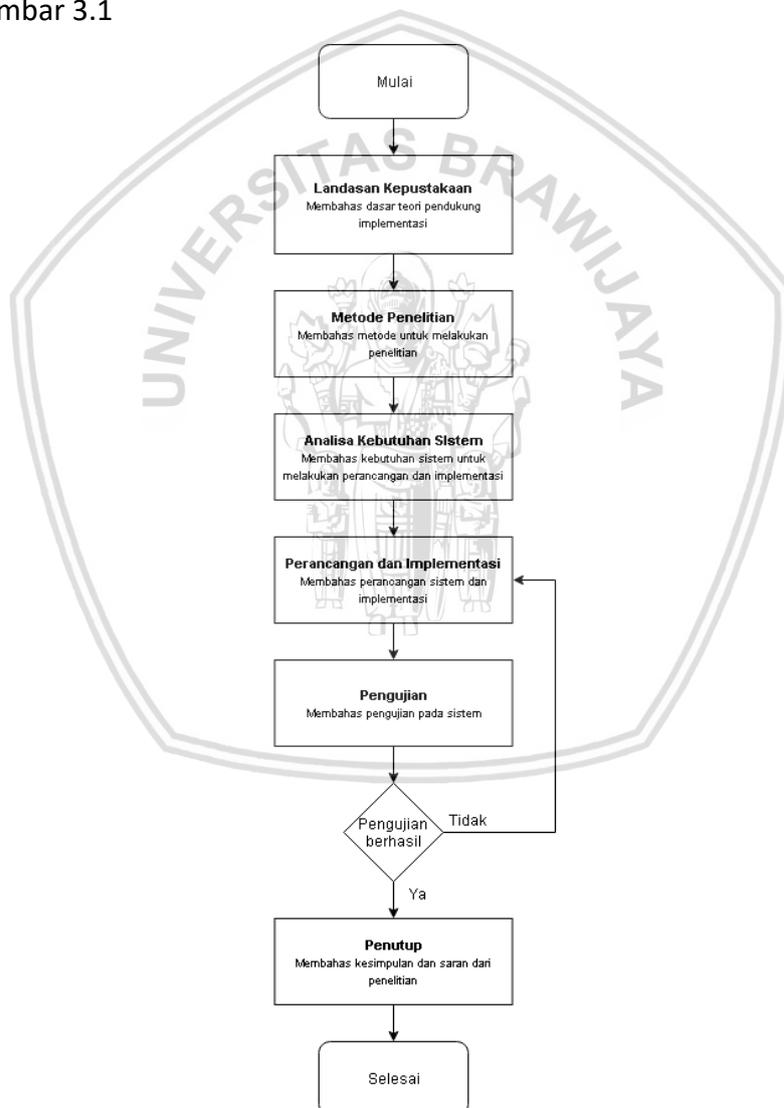
Untuk melihat data di *channel* thingspeak dibutuhkan *ID Channel* dan *API Write* untuk dapat melihat data pada *channel* tersebut. *ID Channel* dan *API Write* dapat dilihat di website thingspeak. Aplikasi Thingview ini bisa melihat data yang ada di *channel* thingspeak secara grafik. Grafik tersebut terdapat waktu, nilai dari data yang terkirimkan dari *publisher*



BAB 3 METODOLOGI PENELITIAN

3.1 Metode Penelitian

Bab ini menjelaskan metode dan langkah yang di tempuh dalam pengerjaan penelitian tentang sistem *monitoring* detak jantung menggunakan NodeMCU dan MQTT. Tipe penelitian ini adalah implementatif, yaitu dengan melakukan rekayasa kebutuhan, perancangan sampai implementasi. Langkah pertama yang dilakukan adalah mengumpulkan teori-teori pendukung dan mengemasnya ke dalam studi pustaka. Kemudian dilanjutkan dengan proses rekayasa kebutuhan sistem, perancangan, implementasi, pengujian, serta kesimpulan dan saran. Adapun beberapa proses yang dilaksanakan terhadap penelitian skripsi ini ditunjukkan pada Gambar 3.1



Gambar 3.1 Diagram Alir Metode Penelitian



3.2 Landasan Pustaka

Landasan pustaka digunakan untuk mencari literatur dan pengetahuan dari sumber yang terpercaya guna mendapatkan metode-metode ilmiah untuk memecahkan masalah yang muncul ketika proses melakukan pembuatan sistem. Beberapa literatur yang dapat digunakan, yaitu berupa buku, jurnal, laporan penelitian maupun artikel. Referensi utama yang diperlukan untuk menunjang penulisan skripsi ini berhubungan dengan kajian pustaka:

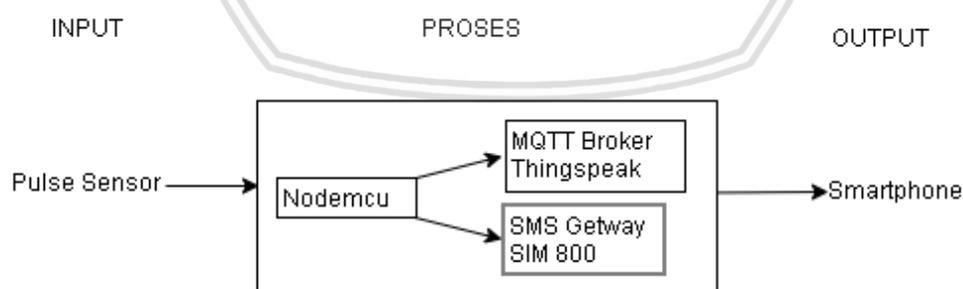
1. *Pulse sensor*.
2. *Wearable Perangkat*.
3. MQTT Protokol.
4. *SMS Gateway*.

3.3 Rekayasa Kebutuhan Sistem

Bab ini akan membahas rekayasa kebutuhan sistem untuk memenuhi kebutuhan perancangan sistem. Pada bab ini akan membahas gambaran umum sistem yang menjelaskan mengenai gambaran umum sistem yang akan dirancang dan diimplementasikan. Menjelaskan kebutuhan fungsional dan kebutuhan sistem yang akan menjelaskan kebutuhan fungsi utama dan sekunder dari sistem tersebut. Dan yang terakhir akan menjelaskan kebutuhan perangkat keras dan perangkat lunak yang dimana akan menjelaskan kebutuhan perangkat keras dan perangkat lunak untuk merancang dan mengimplementasi sistem.

3.4 Perancangan dan Implementasi

Perancangan dan implementasi sistem akan dilakukan setelah semua kebutuhan yang diperlukan oleh sistem sudah didapatkan dan sesuai dengan diagram blok sistem yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Blok Perancangan Sistem

Pada perancangan kali ini, NodeMCU adalah sistem kendali utama dari sistem yang dibentuk. Input pada sistem ini dihasilkan oleh *pulse sensor* yang nanti akan di proses oleh NodeMCU. Setelah NodeMCU memproses hasil dari *pulse sensor*, sistem kendali akan melakukan pengiriman data pada *Thingspeak* menggunakan protokol MQTT dan mengirim sms dengan menggunakan modul sim

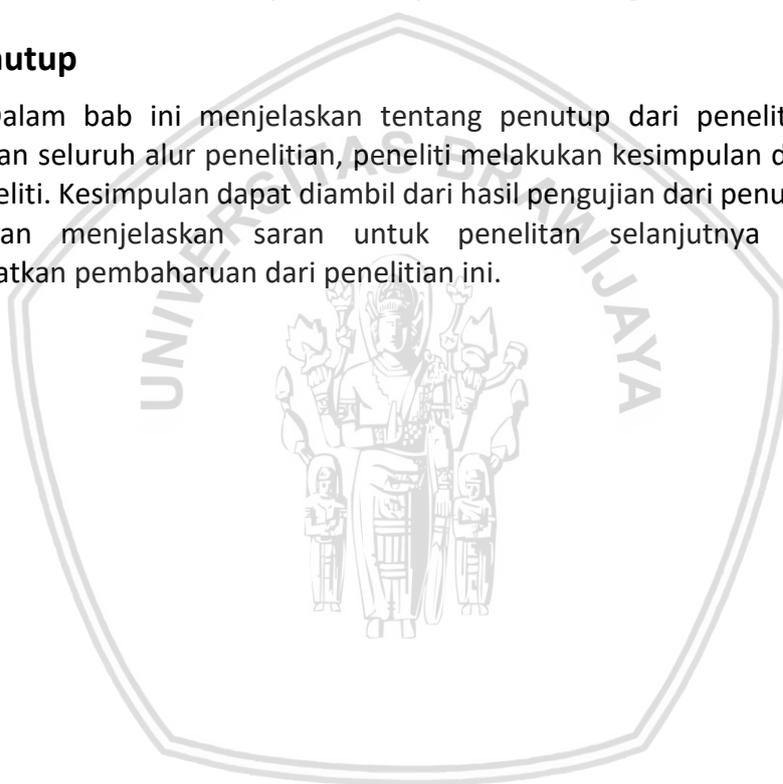
800L. Pada saat data sudah dikirimkan pada *Thingspeak*, kita bisa langsung melihat hasil *monitoring* dengan menggunakan Smartphone. Pada sistem ini juga dapat meminta data terkini denyut jantung dengan mengirim SMS pada sistem dan mengirimkan peringatan jika terjadi kondisi abnormal.

3.5 Pengujian

Pada Bab ini akan dilakukan pengujian. Pengujian keseluruhan sistem dilakukan agar bisa mengetahui perancangan dan implementasi yang dilakukan sebelumnya bisa berhasil atau tidak. Pengujian dilakukan berdasarkan fitur yang ditawarkan pada sistem. Pengujian pada penelitian ini meliputi fungsional sistem, pengujian keakuratan sensor, pengujian komunikasi data mqtt, pengiriman SMS peringatan, dan balasan SMS jika mendapatkan SMS dengan karakter "BPM".

3.6 Penutup

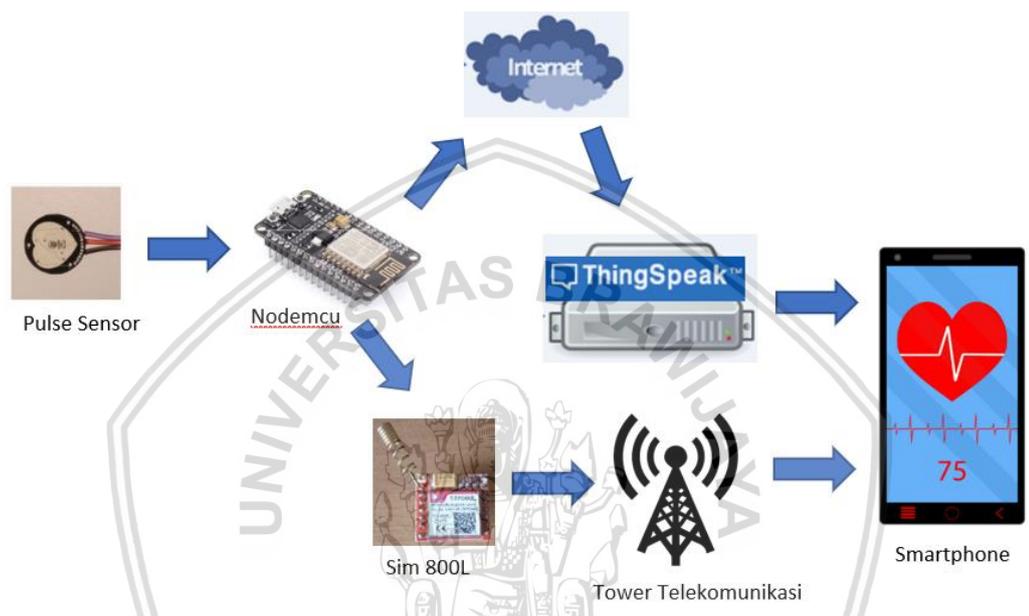
Dalam bab ini menjelaskan tentang penutup dari penelitian. Setelah melakukan seluruh alur penelitian, peneliti melakukan kesimpulan dari apa yang telah diteliti. Kesimpulan dapat diambil dari hasil pengujian dari penulisan. Bab ini juga akan menjelaskan saran untuk penelitian selanjutnya agar dapat mendapatkan pembaharuan dari penelitian ini.



BAB 4 REKAYASA KEBUTUHAN SISTEM

Dalam bab ini akan menjelaskan tentang rekayasa kebutuhan sistem yang harus dilengkapi untuk proses perancangan sampai proses implementasi sistem. Rekayasa kebutuhan ini gambaran dari kebutuhan fungsional dan kebutuhan sistem.

4.1 Gambaran Umum Sistem



Gambar 4.1 Gambaran Umum Sistem

Sistem ini dirancang untuk memudahkan seseorang untuk memantau denyut jantung secara *online* atau dapat mendapatkan data menggunakan internet. Sistem ini bekerja untuk mendapat nilai denyut jantung dengan membaca denyut jantung dengan menggunakan pulse sensor. Setelah sensor membaca denyut jantung, mikrokontroler Nodemcu akan mengolahnya dalam bentuk BPM (*Beat per minute*). Penelitian ini menggunakan protokol MQTT untuk mengirimkan data ke *Channel Thingspeak*. Untuk mengirimkan data perangkat membutuhkan koneksi WiFi untuk mengirimkan data secara *online*. Hasil pengiriman data dari perangkat ke *Channel Thingspeak* dapat dipantau melalui smartphone. Hasil data dari perangkat akan ditampilkan berupa satuan *beat per minute* (BPM). Sistem akan melakukan pengecekan denyut jantung jika lebih dari 100 per menit akan mengirimkan SMS peringatan takikardia dan jika kurang dan kurang dari 60 per menit akan mengirimkan SMS peringatan bradikardia. Sistem juga bersiap untuk menerima SMS dan akan membalasnya dengan nilai BPM. Dibawah ini akan dijelaskan prespektif sistem, ruang lingkup, karakteristik pengguna, lingkungan operasi sistem.

4.1.1 Prespektif Sistem

Sistem ini akan berjalan sesuai perencanaan jika sistem dapat membaca denyut jantung, dan menampilkan hasilnya dalam bentuk beat per minute. Hasil dari pembacaan denyut jantung dapat dilihat di pada layar *smartphone*. Terdapat dua cara untuk menampilkan pada layar *smartphone*, yaitu sistem akan mengirimkan data BPM pada *channel* thingspeak menggunakan MQTT dan mengirimkan balasan SMS dengan nilai BPM menggunakan modul gsm SIM800L. Sistem juga dapat mengirimkan SMS peringatan jika nilai BPM lebih dari 100 atau kurang dari 60.

4.1.2 Ruang Lingkup

Ruang lingkup pada sistem ini yaitu wearable perangkat yang dibuat berupa gelang dan menggunakan *pulse sensor* sebagai denyut jantung. Pembacaan *pulse sensor* menggunakan metode *Photoplethysmography* yaitu membaca perubahan yang terjadi dikulit dan membacanya dengan cara memancarkan sinar dan dilihat perubahan sinar kembalinya. Sistem ini masih tahap pengembangan oleh karena itu masih rentan terhadap gangguan luar, seperti cuaca dan material lain. Penggunaan sistem ini digunakan untuk penggunaan kegiatan normal.

4.1.3 Karakteristik Pengguna

Karakteristik pengguna pada sistem ini diperuntukan bagi seseorang yang ingin memantau kesehatan jantung dengan memantau denyut jantung secara berkala secara praktis dan datanya dapat disimpan sebagai riwayat kesehatan.

4.1.4 Lingkungan Operasi Sistem

Lingkungan operasi sistem pada sistem merupakan kebutuhan yang mendukung sistem dengan kondisi sebagai berikut:

1. Sumber tegangan yang dibutuhkan akan berjalan baik pada 3.3 volt.
2. Membutuhkan koneksi internet untuk mengirim data secara online.

4.1.5 Asumsi dan ketergantungan

Berikut beberapa asumsi dan ketergantungan sebagai persyaratan sistem sebagai berikut:

1. Hasil Pembacaan sensor akan akurat apabila peletakan sensor tepat dan tidak berubah ubah.
2. Untuk memenuhi kebutuhan *power microcontroler*, membutuhkan sumber tegangan antara 3.0v – 3.6v. jika dibawah itu maka *microcontroller* akan mati dan jika diatas itu maka komponen akan rusak karena kelebihan tegangan.

4.2 Kebutuhan Fungsional

Subbab ini menjelaskan kebutuhan fungsional dari sistem ini, kebutuhan fungsional dari sistem yang akan dirancang merupakan penjelasan dari fungsi yang harus ada dalam sistem dan dapat diperoleh oleh sistem tersebut. Kebutuhan sistem meliputi *input* dan *output* dari sistem, serta fungsi respon dari keduanya dalam berjalannya proses sistem. Kebutuhan fungsional dari sistem adalah sebagai berikut:

1. Membaca Denyut Jantung

Sistem membaca denyut jantung dari pergelangan tangan pengguna. Untuk membaca denyut jantung sistem menggunakan sensor yang menerapkan metode photoplethysmography yang salah satu metode untuk membaca keadaan denyut jantung melalui kulit yaitu dipergelangan tangan. Pembacaan denyut jantung kemudian diolah untuk bisa mendapatkan nilai dalam bentuk *Beat per Minute*(BPM).

2. Melihat Hasil Sistem

Untuk melihat hasil dari sistem, perangkat mengirimkan data denyut jantung menggunakan internet dan protokol komunikasi MQTT. Sistem ini menggunakan internet karena data akan dikirimkan secara online. Oleh karena itu sistem dapat dipantau melalui smartphone. Sistem ini menggunakan *broker* untuk bisa menyimpan data, yaitu menggunakan *Channel Thingspeak*. *Channel Thingspeak* sebagai penerima data dari perangkat dan dapat disimpan datanya untuk kebutuhan riwayat kesehatan.

3. Mengirimkan Data Menggunakan Wifi

Sistem ini mengirimkan data menggunakan Wifi yang terkoneksi dengan internet dan akan mengirimkan data denyut jantung ke *Channel Thingspeak*. Setelah data terkumpul di *Channel Thingspeak*, pengguna dapat melihat hasil pembacaan denyut jantung melalui smartphone berbasis android.

4. Mengirimkan SMS peringatan

Setelah melakukan pembacaan denyut jantung, akan dilakukan pengecekan nilai BPM, Jika nilai BPM lebih dari 100(Takikardia) atau kurang dari 60(Bradikardia) maka sistem akan mengirimkan sms peringatan.

5. Membalas SMS nilai bpm saat menerima sms dengan karakter "BPM".

Pada sistem ini juga akan melakukan pengiriman sms dengan nilai bpm jika mendapat sms yang berisi karakter "BPM". Sistem akan mengirimkan nilai bpm terikini saat mendapatkan sms.

4.3 Kebutuhan Sistem

Pada subbab ini akan dijelaskan kebutuhan sistem yang dibutuhkan untuk memenuhi persyaratan sistem yang diperlukan untuk merancang sistem yang akan di buat. Kebutuhan tersebut adalah sebagai berikut:

4.3.1 Arduino IDE (*Intergrated Development Environtment*)

Pada sistem penelitian ini menggunakan mikrokontroler nodemcu. Mikrokontroler nodemcu membutuhkan perangkat lunak untuk mengirimkan kode program, oleh karena itu sistem ini menggunakan arduino ide. Untuk mengirimkan kode program dari komputer ke mikrokontroler membutuhkan *library* yang nanti akan dijelaskan pada subbab selanjutnya. Arduino ide memiliki pilihan *board* untuk mengirimkan kode program. Pilihan *board* pada arduino ide belum terdapat pilihan *board* nodemcu. Pilihan *board* nodemcu dapat diunduh di internet.

4.3.2 Thingspeak

Pada sistem ini dibutuhkan *broker* mqtt sebagai perantara antara sistem dan *smartphone*. *Broker* mqtt yang digunakan pada sistem penelitian ini menggunakan Thingspeak. Thingspeak memiliki *channel* yang dapat dikirimkan data secara terus menerus. Data yang dikirimkan ke *channel thingspeak* membutuhkan dua kunci, yaitu *channel* id dan api write. *Channel* id berfungsi sebagai kunci awal untuk mengirimkan data, sedangkan api write berfungsi untuk menuliskan data pada *channel* thingspeak.

4.3.3 Thingview

Sistem dalam penelitian ini monitoring denyut jantung secara online menggunakan thingspeak. Untuk menampilkan data denyut jantung pada sistem ini menggunakan *smartphone*. Pada *smartphone* membutuhkan aplikasi untuk membaca data denyut jantung. Aplikasi untuk memonitoring denyut jantung pada sistem ini menggunakan thingview. Pada thingview membutuhkan *channel* id dan api write dari thingspeak.

4.3.4 Library

Dalam penelitian ini akan menggunakan beberapa *library*, antara lain:

1. ESP8266WiFi

Library ini berfungsi untuk membuat set fungsi untuk mengaktifkan wifi pada NodeMCU. Dengan *library* ini kita bisa mengkoneksikan NodeMCU dengan wifi.

2. Software Serial

Library ini berfungsi untuk menginialisasi pin digital menjadi pin tr/tx. Dengan mengubah pin digital menjadi pin tr/tx.

3. PubSubClient

Library dibutuhkan untuk bisa mengirimkan data ke thingspeak.

4.3.5 AT Command

Dalam penelitian ini akan menggunakan *AT Command* untuk mengirimkan perintah pada SIM800, antara lain:

1. AT+CSCLK

Pada Perintah ini berfungsi untuk mengkonfigurasi slow clock. Dalam sistem ini konfigurasi slow clock akan tidak diaktifkan.

2. AT+CMGF

Pada Perintah ini berfungsi untuk mengkonfigurasi *Set SMS System* menjadi text mode. Untuk pengiriman SMS perlu mengkonfigurasi menjadi text mode.

3. AT+CMGS

Pada Perintah ini berfungsi untuk mengisi nomer telfon yang akan dikirim SMS. Nomer telfon penerima sms dideklarasikan melalui perintah AT+CMGS.

4. AT+CNMI

Pada Perintah ini berfungsi untuk mengirimkan pesan baru. Perintah ini dilakukan saat hendak mengirimkan pesan baru.

5. AT+CMGL

Pada Perintah ini berfungsi untuk memberikan status pesan yang telah terbaca. Setelah ada pesan yang masuk, perintah ini dilakukan agar pesan yang telah dibaca memberikan status telah terbaca

6. AT+CLIP

Pada Perintah ini berfungsi untuk memanggil CLI(*calling line identity*) agar dapat mengirimkan pesan SMS.

7. AT+CMGDA

Pada Perintah ini berfungsi untuk menghapus SMS yang telah masuk. Perintah ini digunakan agar tidak terjadi penumpukan pesan dalam sim 800L

4.4 Kebutuhan Perangkat Keras

Pada subbab ini menjelaskan kebutuhan perangkat keras. Perangkat keras pada sistem ini meliputi nodemcu sebagai mikrokontroler, *pulse sensor* sebagai sensor denyut jantung, sim 800 sebagai modul GSM, dan baterai Li-Po sebagai catu daya. Berikut penjelasan dari bagian dari komponen yang dibutuhkan oleh sistem.

4.4.1 Pulse Sensor

Fitur utama pada sistem ini yaitu mengukur denyut jantung. Oleh karena itu agar sistem dapat mengukur denyut jantung, diperlukan sensor yang dapat membaca nilai denyut jantung. *Pulse sensor* adalah salah satu komponen yang

berfungsi untuk membaca nilai denyut jantung. Nilai denyut jantung akan dikirimkan ke mikrokontroler. Mikrokontroler akan mengolah pembacaan nilai sensor menjadi denyut jantung per menit.

4.4.2 Mikrokontroler NodeMCU

Kebutuhan perangkat keras pada sistem sebagai pemrosesan keseluruhan sistem membutuhkan perangkat keras mikrokontroler. Perangkat keras mikrokontroler pada sistem ini menggunakan Nodemcu. Sistem ini dirancang agar dapat mengirimkan denyut jantung secara online, mengirimkan SMS peringatan, dan membalas dengan nilai denyut jantung per menit. Kebutuhan dari perangkat ini yaitu, membutuhkan pin analog agar dapat membaca *pulse sensor*, membutuhkan pin digital agar dapat mengirimkan perintah ke sim800 menggunakan software serial untuk dijadikan trtx.

4.4.3 Modul GSM Sim 800

Salah satu perangkat keras yang dibutuhkan oleh sistem ini adalah modul GSM sim800. Modul GSM sim800 memiliki dua fungsi yaitu, berfungsi mengirimkan SMS peringatan dan membalas SMS dengan nilai denyut jantung per menit. Fitur yang berfungsi sebagai pengirim SMS peringatan terjadi jika kondisi denyut jantung per menit kurang dari 60 atau lebih dari 100. Sedangkan fitur yang berfungsi membalas SMS dengan nilai denyut jantung terjadi jika mendapatkan SMS dengan kata "BPM". Fitur kedua pada modul GSM sim800 ditambahkan agar ketika sistem tidak dapat menjalankan sistem monitoring denyut jantung melalui mqtt, pengguna dapat melihat dengan cara mengirimkan SMS pada sim800 dan selanjutnya sim800 dapat mengirimkan nilai denyut jantung per menit.

4.4.4 Baterai Li-Po

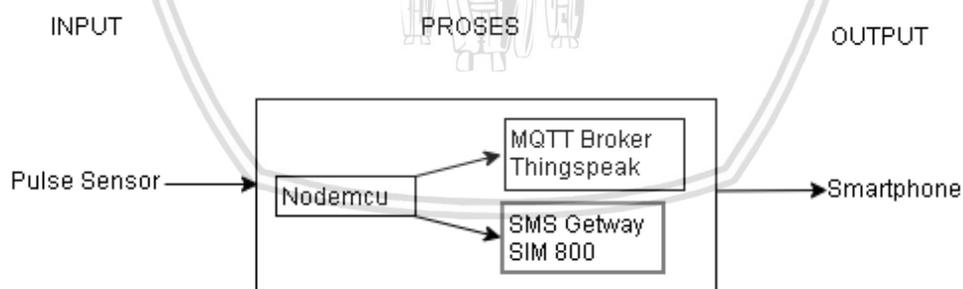
Sebuah sistem yang terdapat perangkat keras didalamnya membutuhkan catu daya agar seluruh sistem dapat berjalan. Catu daya pada sistem dalam penelitian ini menggunakan baterai li-po. Baterai li-po dibutuhkan untuk memberikan catu daya pada sistem. Perangkat keras ini memiliki kabel yang berwarna merah dan kabel yang berwarna hitam. Kabel yang berwarna merah sebagai vcc dan kabel yang berwarna hitam sebagai gnd. Kedua kabel tersebut dihubungkan pada mikrokontroler nodemcu dan modul GSM sim 800. Kabel yang berwarna merah dihubungkan pada vcc mikrokontroler nodemcu dan modul GSM sim800. Kabel yang berwarna hitam dihubungkan pada gnd mikrokontroler nodemcu dan modul GSM sim800.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab perancangan dan implementasi akan membahas tentang perancangan dan implementasi dari sistem monitoring denyut jantung menggunakan nodemcu dan mqtt. Sistem ini akan dimulai dari perancangan perangkat keras dan perancangan perangkat lunak, lalu dilanjutkan implemetasi perangkat keras dan implementasi perangkat lunak. Dalam bab ini akan menjelaskan lebih detail agar dapat mengerti dengan kegunaan sistem.

5.1 Diagram Blok Sistem

Tahap awal untuk memulai perancangan yaitu dengan merancang diagram blok sistem. Gambaran umum dari penelitian ini yaitu monitoring denyut jantung menggunakan nodemcu, mqtt, dan sms gateway seperti pada Gambar 5.1. Pada sistem ini menggunakan nodemcu sebagai mikrokontroler yang akan menerima data dari sensor untuk diolah agar mendapatkan nilai Bpm (*Beat per minute*) dan mengirimkan data secara online dengan menggunakan mqtt serta mengirimkan SMS ke nomer tertentu untuk mendapatkan SMS peringatan jika pengguna terkena takikardia dan bradikardia. Selain itu sistem ini juga bisa mengirimkan nilai BPM terkini dengan cara mengirim permintaan pada sistem menggunakan sms yang berisi karakter "BPM". Pengiriman MQTT membutuhkan broker MQTT, oleh karena itu pada sistem ini menggunakan broker Thingspeak. Broker MQTT menjadi jembatan antara nodemcu dan smartphone. Fitur yang terdapat pada sistem ini adalah pengguna dapat memonitor, menyimpan data denyut jantung secara online dan dapat dilihat oleh orang tertentu, dapat mengirimkan SMS peringatan ke nomer tertentu, dan juga dapat membalas SMS dengan nilai BPM terkini.



Gambar 5.1 Diagram Blok Sistem

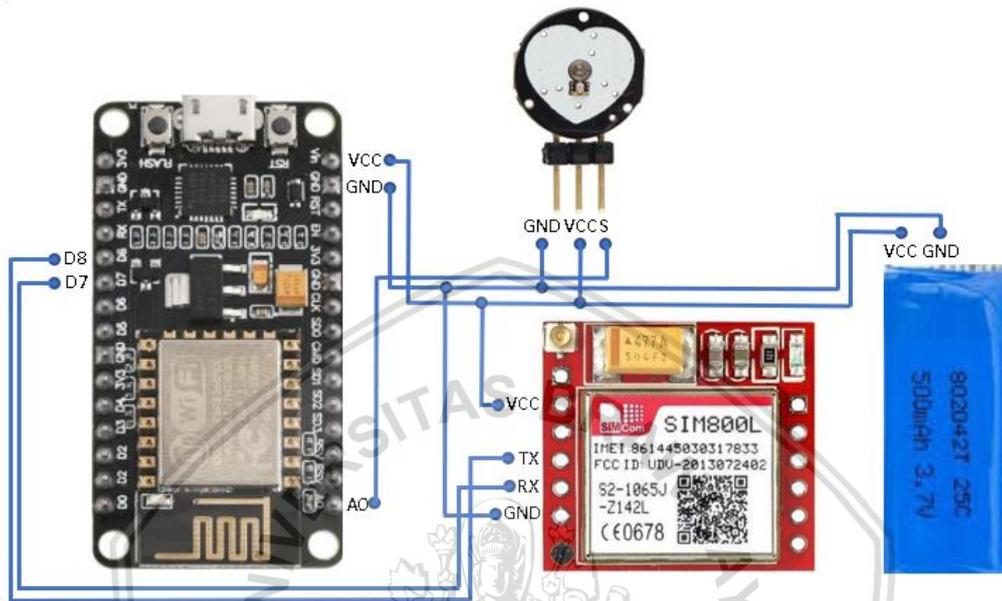
5.2 Perancangan Sistem

Perancangan sistem pada bab ini terdapat 3 subbab meliputi, perancangan perangkat keras, diagram alur perancangan perangkat lunak, dan yang terakhir perancangan perangkat lunak.



5.2.1 Perancangan Perangkat Keras

Pada perancangan perangkat keras akan dijelaskan skematik secara keseluruhan pada sistem. Skematik sistem akan menjelaskan hubungan antar komponen agar sistem dapat berjalan dengan baik. Berikut adalah Gambar 5.2 perancangan perangkat keras.



Gambar 5.2 Skematik Perancangan Perangkat Keras

Terdapat 4 buah komponen perangkat keras yang akan dirakit pada pcb. Daya pada sistem ini menggunakan baterai Li-Po, olehkarena itu vcc dan gnd pada baterai ini diparalel dengan 3 komponen lainnya, yaitu nodemcu, pulse sensor, dan sim800. Setelah daya setiap komponen terpenuhi, nodemcu sebagai mikrokontroler akan di hubungkan dengan pulse sensor dan sim800L. Pulse sensor sebagai sensor analog akan di hubungkan pada pin analog nodemcu yaitu pin S di hubungkan pada pin A0. Sim 800 akan di hubungkan dengan nodemcu dengan RX di hubungkan pada pin D7 dan TX di hubungkan pada D8. Berikut hubungan pin antar komponen dapat dilihat pada table 5.1.

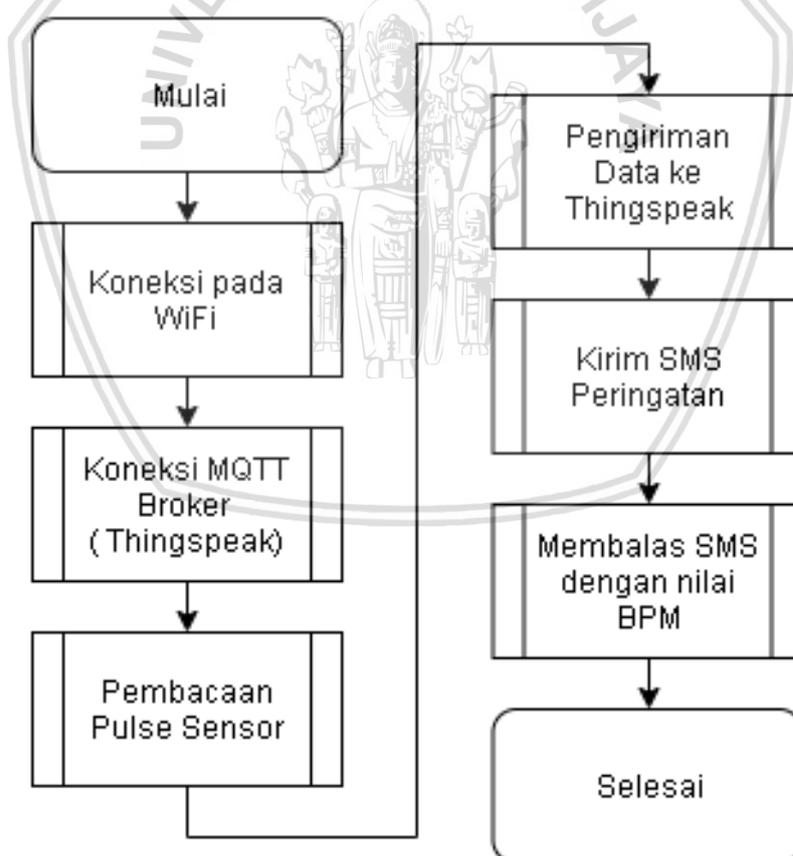
Tabel 5.1 Hubungan Pin antar Komponen

No	PIN Nodemcu	PIN SIM 800	PIN Pulse Sensor	PIN Baterai Li-Po
1	PIN VCC	PIN VCC	PIN VCC	PIN VCC
2	PIN GND	PIN GND	PIN GND	PIN GND
3	A0	-	PIN S	-

4	D7	-PIN RX	-	-
5	D8	-PIN TX	-	-

5.2.2 Diagram Alur Perancangan Perangkat Lunak

Subbab ini akan menjelaskan tentang diagram alur perangkat lunak yang akan dirancang untuk sistem ini. Bisa dilihat Gambar 5.3 , alur dari program perangkat lunak dimulai dari mengkoneksikan nodemcu pada WiFi. Setelah nodemcu terkoneksi dengan Wifi selanjutnya mengecek koneksi MQTT Broker telah terhubung dengan Thingspeak atau belum. Selanjutnya sistem akan melakukan pengukuran sensor. Pengukuran sensor akan dimasukkan pada variable BPM. Variabel akan melakukan penyeleksian 3 kondisi, yaitu kondisi normal, kondisi bradikardia, dan kondisi takikardia. Sistem ini akan mengirimkan sms peringatan jika melewati penyeleksian kondisi bradikardia dan kondisi takikardia, berikut kondisinya. Jika variabel BPM kurang dari 60 maka sistem akan mengirimkan sms peringatan “Awas Bradikardia”. Sedangkan Jika variabel BPM lebih dari 100 maka sistem akan mengirimkan sms peringatan “ Awas Takikardia”.



Gambar 5.3 Diagram Alur Perancangan Perangkat Lunak

Setelah melewati proses penyeleksian kondisi, sistem akan mengirim data variabel BPM ke Thingspeak setelah pembacaan sensor 60 detik. Sistem juga



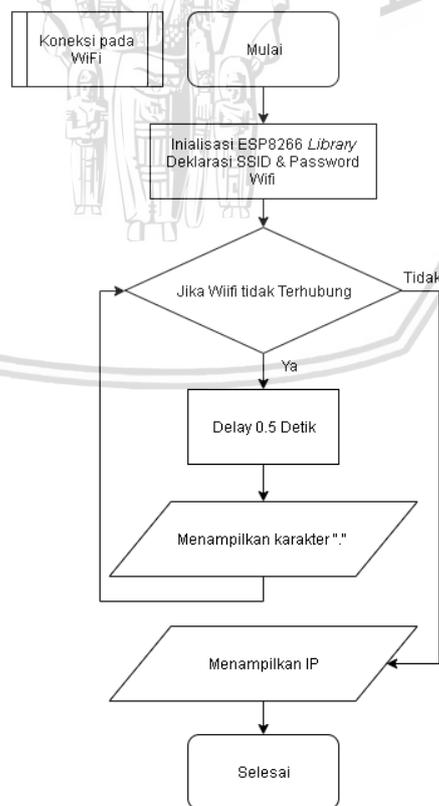
terdapat fitur membalas sms untuk mendapatkan nilai BPM terkini. Oleh karena itu sistem selalu bersiap untuk mendapatkan sms masuk. Sistem akan mengenali nomer telfon dan isi dari sms yang masuk. Jika sms terdapat karakter “BPM” maka sistem akan mengirmkan sms dengan nilai BPM terkini ke nomer telfon yang dari mengirmkan pesan “BPM”.

5.2.3 Perancangan Perangkat Lunak

Pada Sub bab ini, perancangan perangkat lunak menjelaskan bagaimana pembacaan pulse sensor, mengkoneksikan nodemcu pada *access point*, mengkoneksikan nodemcu ke MQTT broker(thingspeak), pengiriman nilai BPM ke *channel* thingspeak, pengiriman SMS peringatan, dan membalas SMS dengan nilai BPM. Berikut akan dijelaskan pada masing-masing bab.

5.2.3.1 Perancangan Nodemcu koneksi ke WiFi

Pada Gambar 5.4 Ini akan menjelaskan tentang perancangan mengkoneksikan nodemcu ke WiFi. Perancangan perangkat lunak pada komponen Nodemcu untuk melakukan koneksi dengan wifi membutuhkan *library ESP8266WiFi.h*. Untuk mengkoneksikan nodemcu ke wifi dibutuhkan *SSID* dan *Password* wifi. Setelah mengetahui *SSID* dan *Password* wifi, tuliskan *syntax* program untuk medeklarasikan *SSID* dan *Password* wifi pada kode program. Untuk memeriksa potongan kode program yang mengkoneksikan nodemcu ke wifi, perlu menuliskan kode program log untuk mengetahui ip nodemcu pada koneksi wifi.

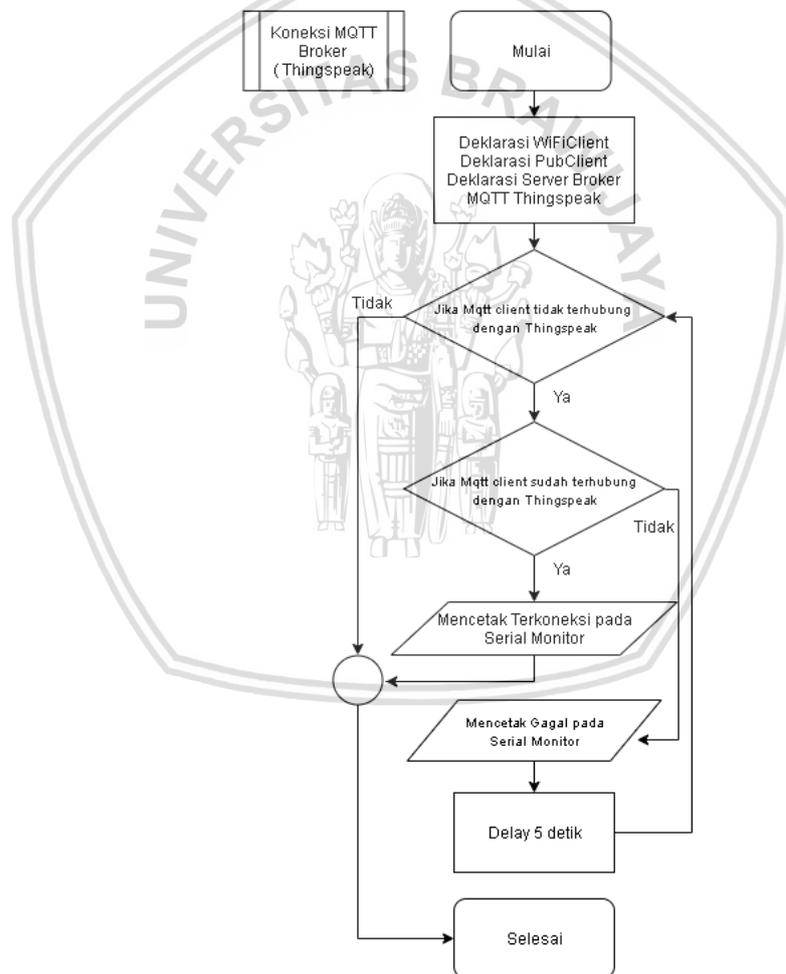


Gambar 5.4 Flowchart Koneksi ke Wifi



5.2.3.2 Perancangan Nodemcu Koneksi ke MQTT Broker (Thingspeak)

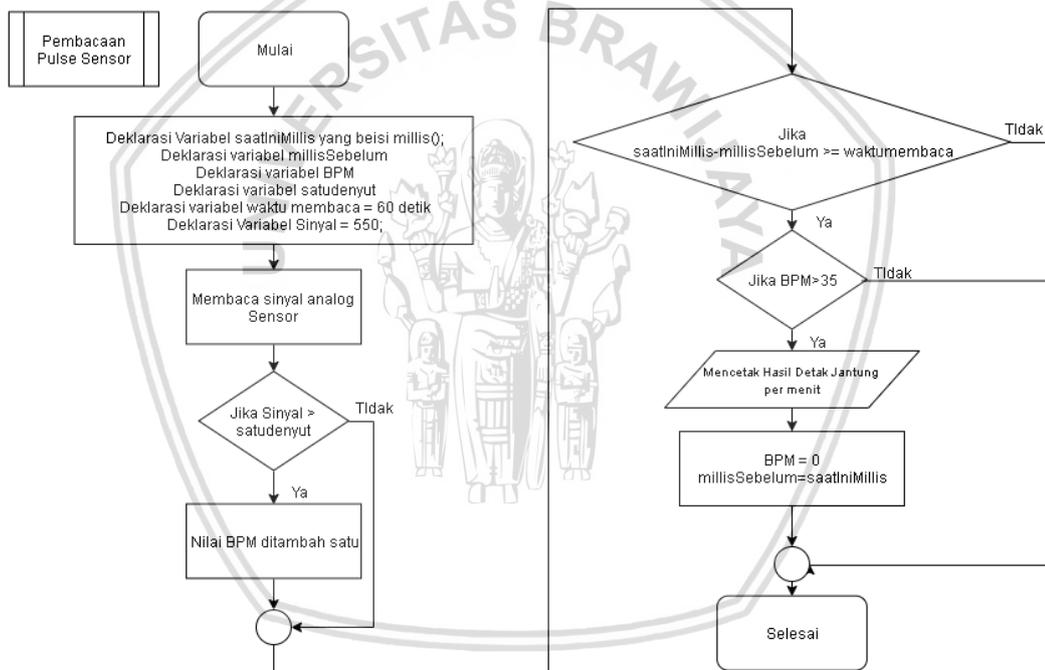
Pada Gambar 5.5 Ini akan menjelaskan tentang perancangan untuk menkoneksi *protocol* MQTT. Perancangan protokol MQTT bertujuan untuk mengirimkan data denyut jantung dari nodemcu ke thingspeak. Pengiriman dengan MQTT dibutuhkan koneksi internet. Koneksi internet dalam sistem ini menggunakan *Hotspot* pada smartphone. Membutuhkan tiga deklarasi untuk dapat mengkoneksi koneksi MQTT yaitu, deklarasi WiFiClient, PubClient, dan Server broker MQTT Thingspeak. Setelah itu akan melakukan looping sampai MQTT terkoneksi. Jika koneksi MQTT sudah terkoneksi maka akan menampilkan telah terkoneksi pada serial monitor, jika belum maka akan mencetak gagal pada serial monitor dan akan delay selama 5 detik dan akan mengulang kembali mengkoneksi dengan *server broker* MQTT thingspeak.



Gambar 5.5 Flowchart Koneksi MQTT

5.2.3.3 Perancangan Pembacaan Pulse Sensor

Pada Gambar 5.6 Ini akan menjelaskan tentang perancangan pembacaan pulse sensor. Pertama beberapa variabel pembantu untuk mendapatkan denyut jantung permenit, yaitu variabel waktu menggunakan millis() untuk menghitung denyut jantung selama satu menit, variabel BPM untuk menyimpan denyut jantung, dan variabel satudenyut untuk menentukan keluaran nilai sensor analog yang dapat dikategorikan sebagai satu denyut. Setelah deklarasi masuk ke pembacaan sensor pembacaan sensor menggunakan analogread dan dimasukkan pada variabel Sinyal. Variabel Sinyal akan dilakukan pengecekan nilainya lebih dari variabel satudenyut atau kurang dengan menggunakan percabangan jika variabel Sinyal lebih dari variabel satu denyut maka variabel BPM ditambah satu. Proses itu dilakukan selama 60 detik atau 1 menit. Setelah melewati 60 menit akan masuk kepercabangan jika nilai lebih dari 35 maka akan mencetak nilai BPM pada serial monitor. Setelah melakukan pembacaan nilai sensor selama 60 detik variabel BPM diberi nilai 0 agar dapat mengisi nilai BPM selanjutnya.

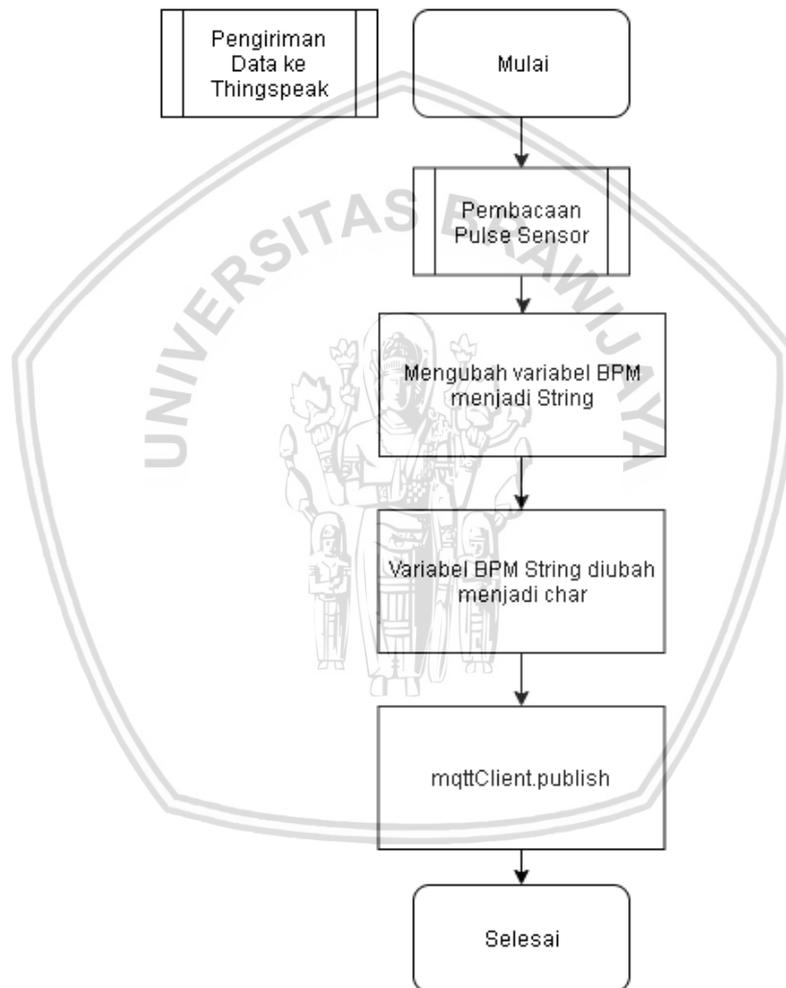


Gambar 5.6 Flowchart Pembacaan Pulse sensor



5.2.3.4 Perancangan Pengiriman Nilai BPM ke Channel Thingspeak

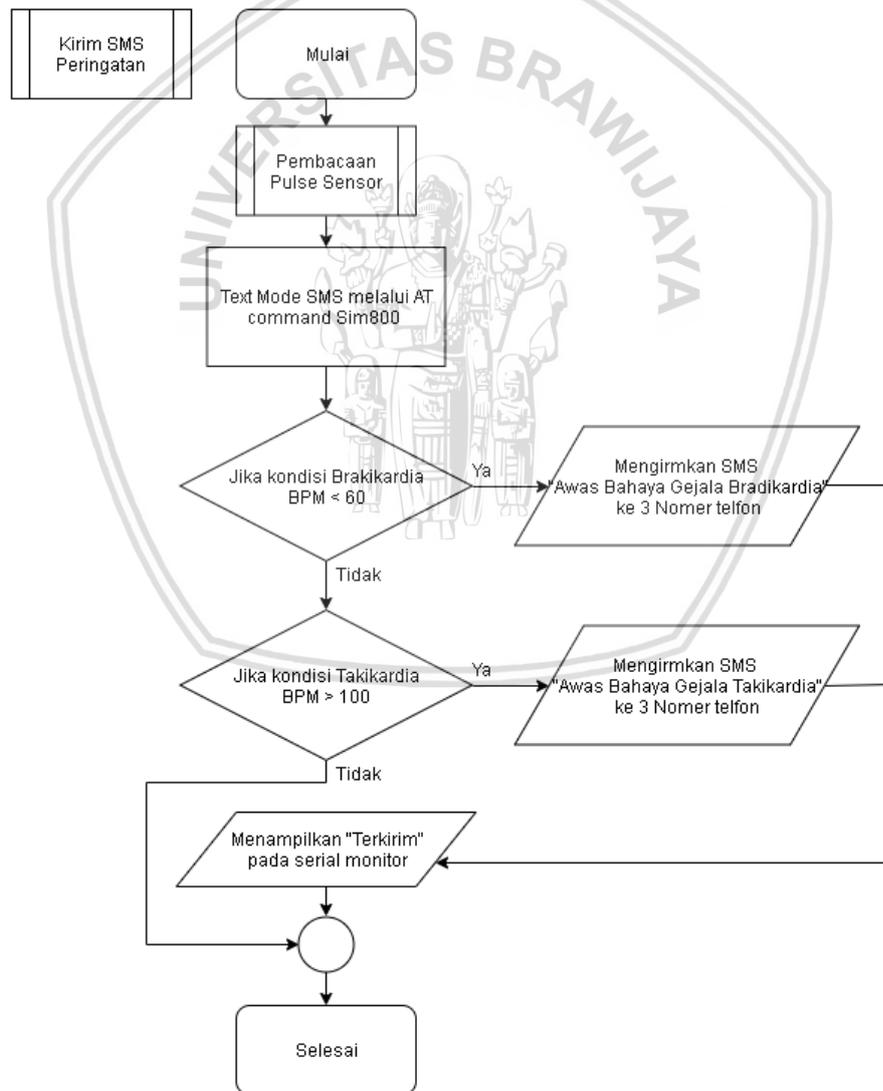
Pada Gambar 5.7 akan menjelaskan perancangan pengiriman nilai BPM ke *channel* thingspeak. Pengiriman nilai BPM akan dilakukan per karakter, oleh karena itu nilai variabel BPM akan diubah menjadi string dan setelah itu menjadi char. Langkahnya yaitu setelah mendapat nilai BPM selama satu menit sistem akan mengubah variabel yang menyimpan BPM dari integer menjadi String menggunakan `String [variabel] = String([variabel],DEC)`. Setelah menjadi String, variabel akan diubah menjadi variabel array bertipe data char. Setelah itu variabel akan dikirimkan menggunakan *method* `mqttClient.publish` bersama channel ID dan Write API Thingspeak.



Gambar 5.7 Flowchart Pengiriman Nilai BPM ke Channel Thingspeak

5.2.3.5 Perancangan Pengiriman SMS Peringatan

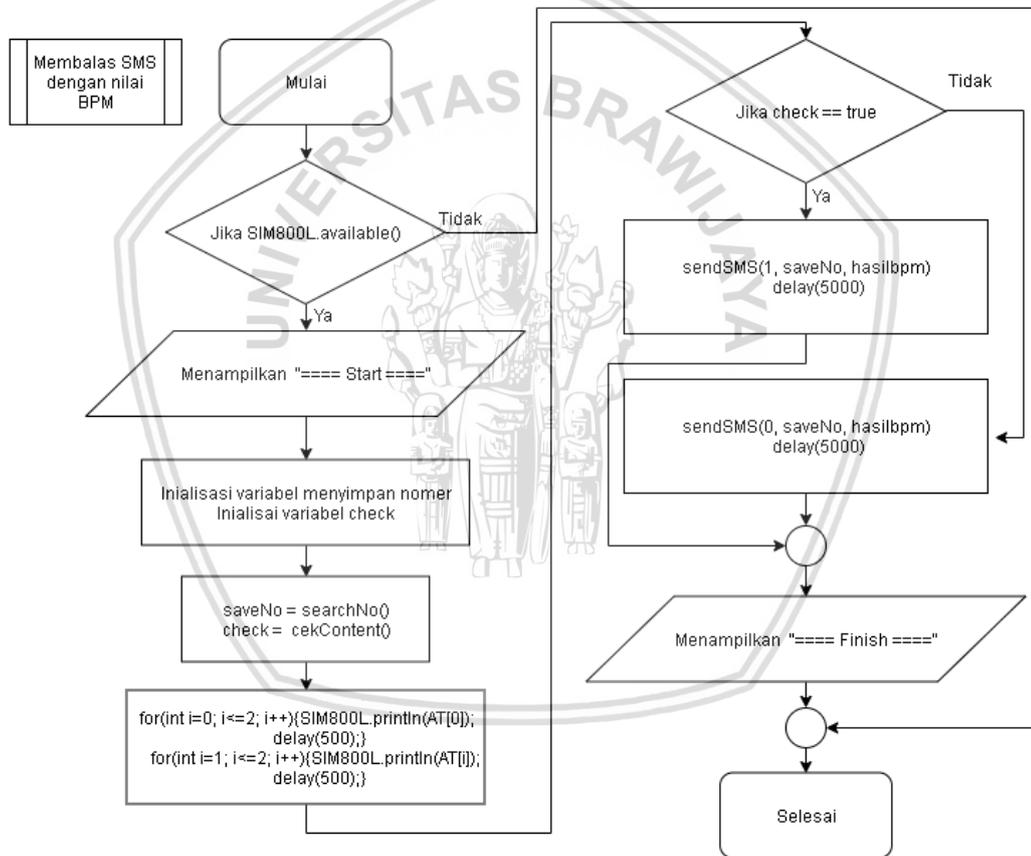
Pada Gambar 5.8 ini akan menjelaskan perancangan pengiriman sms peringatan. Pengiriman sms akan dilakukan saat kondisi tertentu yaitu, kondisi nilai BPM sesuai dengan kondisi Takikardia (kondisi BPM diatas 100) dan kondisi Bradikardia (kondisi BPM dibawah 60). Langkah untuk mengirimkan sms dengan Sim 800L, pertama membutuhkan masukan nilai kondisi BPM setelah 1 menit, selanjutnya mengirimkan perintah ATCommand ke Sim 800 L untuk mengaktifkan Text SMS mode. Setelah aktif Text SMS Mode, melakukan percabangan jika kondisi Bradikardia maka akan mengirimkan sms "Awat Bahaya Gejala Bradikardia" ke 3 nomer yang telah didaftarkan pada kodingan, dan jika kondisi Takikardia maka akan mengirimkan sms "Awat Bahaya Gejala Takikardia" ke 3 nomer yang telah didaftarkan pada kodingan. Setelah mengirimkan SMS peringatan salah satu kondisi maka akan menampilkan "Terkirim" pada serial monitor. Sistem tidak akan mengirimkan sms peringatan jika nilai antara nilai 60 sampai 100.



Gambar 5.8 Flowchart Pengiriman SMS Peringatan

5.2.3.6 Perancangan Membalas SMS dengan Nilai BPM

Pada Gambar 5.9 Ini akan menjelaskan tentang perancangan membalas SMS dengan nilai BPM. Sistem ini terdapat fitur membalas SMS nilai BPM terkini kepada siapapun yang mengirimkan SMS dengan karakter "BPM". Untuk dapat membalas SMS secara langsung, SIM800L harus selalu aktif dengan cara melakukan perulangan pengiriman perintah AT Command untuk Sim800L selalu *Mode Wake up*. Pertama sistem akan menampilkan "Start" pada serial monitor jika ada sms masuk. Untuk membalas SMS dibutuhkan identifikasi terhadap nomer telfon dan isi SMS. Nomer telfon dan isi SMS akan dimasukkan pada variabel yang nanti akan dicek apakah isi SMS adalah "BPM". Jika isi SMS adalah "BPM" maka sistem akan mengirimkan nilai BPM terkini, dan jika isi SMS bukan "BPM" maka akan mengirimkan pesan "karakter yang dimasukkan salah". Setelah mengirimkan balasan sistem akan menampilkan "Finish" pada serial monitor.



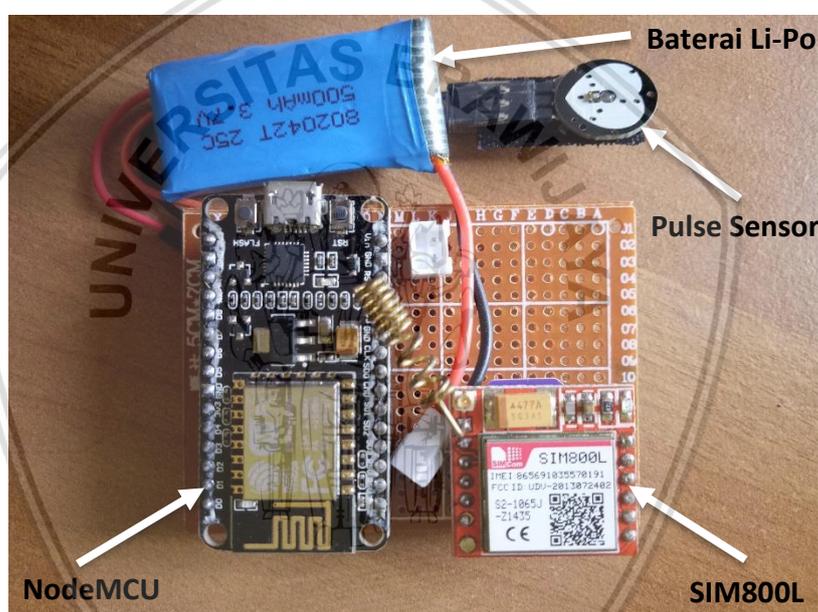
Gambar 5.9 Flowcart Membalas SMS dengan Nilai BPM

5.3 Implementasi Sistem

Setelah Melakukan Perancangan, Subbab ini akan melakukan implementasi dari subbab perancangan diatas. Implementasi akan dibagi menjadi 2 bagian, yaitu implementasi perangkat keras dan implementasi perangkat lunak. Implementasi perangkat keras akan menjelaskan komponen apa saja yang akan dirangkai menjadi satu kesatuan. Pada implementasi perangkat lunak akan menjelaskan fitur apa saja yang bisa diimplementasikan.

5.3.1 Implementasi Perangkat Keras

Pada subbab Impementasi perangkat keras akan menjelaskan hasil dari perancangan yang dilakukan sebelumnya. Pada tahap awal implementasi perangkat keras, menyambungkan setiap komponen di pcb 5cmX7cm. Komponen pcb terdapat nodemcu, sim800, dan kabel menuju *pulse sensor*. Berikut Gambar 5.10 menampilkan perangkat keras dan sistem.



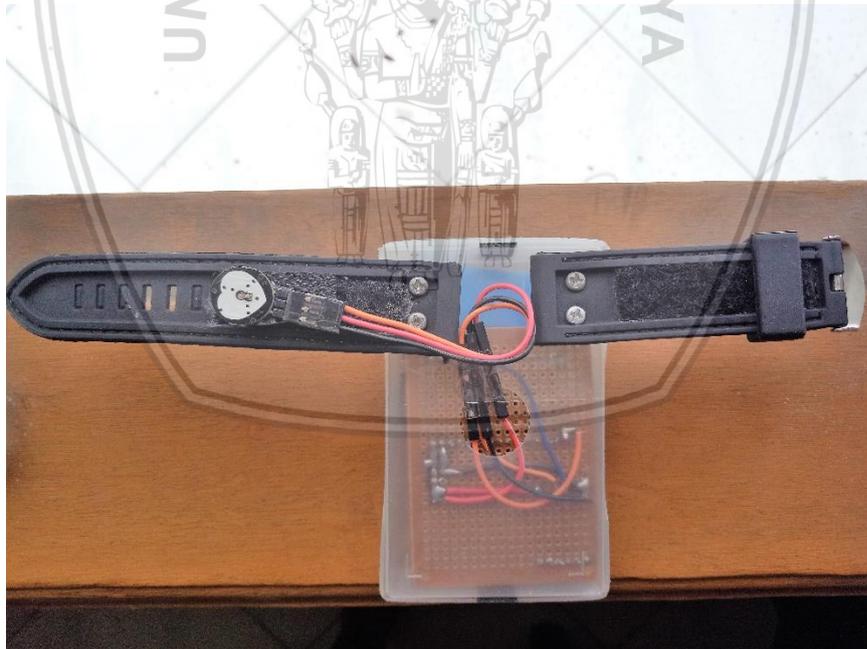
Gambar 5.10 Implementasi Perangkat Keras

Untuk menghidupkan perangkat keras sistem menggunakan baterai li-po. Pada *pulse sensor* menggunakan kabel agar dapat menjangkau titik pengelangan tangan untuk mendapatkan nilai BPM. Setelah perangkat keras sistem tergabung seluruh komponen yang dibutuhkan adalah *packaging* dari sistem ini. Karena sistem ini bersifat *wearable*, maka membutuhkan *packaging* berbentuk seperti jam.



Gambar 5.11 Implementasi Purwarupa Tampak Atas

Pada Gambar 5.11 menampilkan implementasi purwarupa tampak atas dari sistem. Purwarupa menggunakan kotak plastik sebagai pembungkus dari komponen nodemcu, sim800L, dan baterai li-po. Agar dapat diletakkan pada pergelangan tangan membutuhkan *strap* berbahan karet yang biasanya digunakan oleh jam tangan. Pada purwarupa menggunakan bahan strap berbahan karet agar dapat memudahkan sensor membaca denyut jantung karena elastisitasnya.



Gambar 5.12 Implementasi Purwarupa Tampak Bawah

Pada Gambar 5.12 menampilkan peletakan sensor di *strap*. Agar dapat memudahkan pemindahan sensor sesuai dengan pergelangan user pada *strap* ditambahkan *Velcro*. Sensor juga dihubungkan dengan nodemcu menggunakan kabel.

5.3.2 Implementasi Perangkat Lunak

Pada bab ini menjelaskan implementasi perangkat lunak yang meliputi, menyambungkan nodemcu ke koneksi Wifi, Implementasi MQTT *protocol*, pembacaan nilai pulse sensor, pengiriman nilai BPM ke *channel* thingspeak, pengiriman SMS peringatan, membalas SMS dengan nilai BPM terkini.

5.3.2.1 Implementasi Nodemcu koneksi ke WIFI

Tabel 5.2 Kode Sumber Nodemcu koneksi ke WiFi

Baris	Kode Sumber
1	#include "ESP8266WiFi.h"
2	void cekip(){
3	while (WiFi.status() != WL_CONNECTED) {
4	delay(500);
5	Serial.print(".");
6	}
7	// Jika terhubung akan serial print Wifi terhubung.
8	Serial.println("");
9	Serial.println("WiFi Terhubung");
10	// Serial Print Ip.
11	Serial.println(WiFi.localIP());
12	}

Untuk mengirimkan data ke Thingspeak, membutuhkan koneksi internet. Koneksi internet dalam sistem ini menggunakan hotspot *smartphone*. Untuk bisa mengkoneksikan WiFi menggunakan hotspot, dalam kodingan harus menggunakan *library* ESP8266 untuk mengaktifkan WiFi pada nodemcu seperti pada baris 1. Untuk mengetahui WiFi sudah terhubung atau belum, dapat dilakukan dengan membuat *method* untuk mencetak IP nodemcu dengan perulangan jika Wifi status belum terkoneksi maka akan menampilkan karakter "." sebagai tanda menunggu sampai Wifi terkoneksi seperti pada baris 3 sampai baris 6. Jika koneksi WiFi berhasil, sistem akan menampilkan "WiFi Terhubung" dan menampilkan IP Nodemcu pada *serial monitor* seperti pada baris 9 sampai baris 11.

5.3.2.2 Implementasi Nodemcu Koneksi ke MQTT Broker (Thingspeak)

Protokol MQTT digunakan untuk komunikasi pengiriman data dari nodemcu ke thingspeak. Untuk mengimplementasikan protokol ini menggunakan *library* pubClient pada baris 1 dan code program untuk medeklarasikan WiFiClient, mqttClient, dan server broker MQTT pada baris 4 sampai baris 9. Berikut Tabel 5.3 potongan code program untuk mengimplementasikan *protocol* mqtt. Untuk mengecek Koneksi MQTT sudah terkoneksi atau belum, dengan cara membuat *method*. Didalam *method* terdapat perulangan sampai koneksi MQTT terhubung, didalam loop terdapat percabangan jika MQTT sudah terhubung akan menampilkan “Terkoneksi” dan Jika MQTT belum terhubung akan menampilkan “Gagal” dan status gagal seperti pada baris 11 sampai 24. Pada baris 17 sampai baris 23 menjelaskan jika saat MQTT gagal terkoneksi akan melakukan delay selama 5 detik untuk menghubungkannya lagi.

Tabel 5.3 Kode Sumber Nodemcu Koneksi ke MQTT Broker (Thingspeak)

Baris	Kode Sumber
1	#include <PubSubClient.h>
2	// Pengaturan yang dibutuhkan untuk mengirimkan data dengan
3	mqtt.....
4	// Inialisasi WiFi client library
5	WiFiClient client;
6	// Inialisasi PuBSubClient library
7	PubSubClient mqttClient(client);
8	// Deklarasi server broker mqtt yaitu thingspeak
9	const char* server = "mqtt.thingspeak.com";
10	void cekkoneksimqtt(){
11	while (!mqttClient.connected()){
12	Serial.print("Menghubungkan koneksi MQTT...");
13	// koneksi ke MQTT broker
14	if (mqttClient.connect("WiFiClient")) {
15	Serial.println("Terkoneksi");
16	}
17	else { Serial.print("Gagal, rc=");
18	// Serial print status gagal
19	Serial.print(mqttClient.state());
20	Serial.println(" Mencoba lagi setelah 5 detik ");
21	// Tunggu 5 detik sebelum mencoba untuk konek lagi
22	delay(5000);
23	}
24	}
25	}
26	

5.3.2.3 Implementasi Pembacaan Pulse Sensor

Pada Subbab ini akan menjelaskan tentang Implementasi pembacaan sensor. Pada implementasi pembacaan sensor terdapat 2 *function* yaitu, *membaca_sensor()* pada baris 11 sampai 37 dan *enampuluhdetik()* pada baris 39 sampai 44. *Function* *membaca_sensor()* berfungsi untuk membaca nilai sensor dan akan melakukan percabangan jika sudah melewati 60 detik. Untuk dapat mengetahui pembacaan sensor yang telah melewati 60 detik menggunakan *method* *millis()* dan 3 variabel yaitu, variabel *saatIniMillis*, *millisSebelum*, *waktumembaca*. Pembacaan nilai sensor menggunakan bantuan variabel *BPM* dan variabel konstanta *tracehold* senilai 550 yang berguna untuk menandakan satu denyut seperti pada baris 4. Setelah pembacaan sensor selama 60 detik, maka dilakukan pemanggilan *function* *enampuluhdetik()*: yang berfungsi sebagai pemberian nilai variabel *hasilbpm* sebagai variabel global yang menyimpan nilai *BPM* terkini.

Tabel 5.4 Kode Sumber Pembacaan Pulse Sensor

Baris	Kode Sumber
1	<code>// Inialisai Variabel yang dibutuhkan untuk membaca</code>
2	<code>sensor.....</code>
3	<code>int Sinyal;</code>
4	<code>int satudenyut = 550;</code>
5	<code>int BPM=0;</code>
6	<code>unsigned long millisSebelum=0;</code>
7	<code>unsigned long saatIniMillis=0;</code>
8	<code>int waktumembaca=60000;</code>
9	<code>//inialisasi nilai BPM global</code>
10	<code>int hasilbpm=0;</code>
11	<code>void membaca_sensor() {</code>
12	<code> // pointer untuk membaca sensor selama 60 detik.</code>
13	<code> saatIniMillis = millis();</code>
14	<code> // pembacaan sensor diletakkan ke variabel Sinyal.</code>
15	<code> Sinyal = analogRead(PulseSensorPurplePin);</code>
16	<code> // Jika sesuai dengan nilai satu denyut akan bertambah 1</code>
17	<code> variabel BPM.</code>
18	<code> if (Sinyal > satudenyut) {</code>
19	<code> BPM += 1;</code>
20	<code> }</code>
21	
22	<code> Serial.print("Detik ke :");</code>
23	<code> Serial.println(saatIniMillis/1000);</code>
24	<code> Serial.print("Nilai Sinyal sensor :");</code>
25	<code> Serial.println(Sinyal);</code>
26	<code> Serial.print("Status bpm :");</code>
27	<code> Serial.println(BPM);</code>
28	
29	<code> // jika sudah melewati 60 detik akan mengirimkan sms jika</code>
30	<code> BPM<60Mem atau BPM>100</code>
31	<code> // dan mengirimkan ke Thingspeak.</code>
32	<code> if (saatIniMillis - millisSebelum >= waktumembaca) {</code>
33	<code> enampuluhdetik();</code>
34	<code> BPM = 0;</code>
35	<code> millisSebelum = saatIniMillis;</code>
36	<code> }</code>
37	<code>}</code>
38	
39	<code>void enampuluhdetik(){</code>
40	<code> hasilbpm = BPM;</code>



Baris	Kode Sumber
41	<code>Serial.print("Hasil Detak jantung=");</code>
42	<code>Serial.println(BPM);</code>
43	<code>Serial.println("-----");</code>
44	<code>}</code>

5.3.2.4 Implementasi Pengiriman Nilai BPM ke Channel Thingspeak

Pada Subbab ini menjelaskan implementasi dari pengiriman nilai BPM ke *channel* thingspeak. Agar dapat mengirimkan nilai sensor pada *channel* thingspeak membutuhkan koneksi terhadap server MQTT broker thingspeak yang telah dijelaskan pada subbab sebelumnya. Setelah sistem sudah terhubung dengan MQTT broker Thingspeak, sistem akan bersiap untuk mengirim data ke *channel* thingspeak. Pengiriman thingspeak membutuhkan variabel pembantu untuk mendeklarasikan nilai BPM seperti pada baris 3. Variabel bpm digunakan untuk mengisi nilai BPM terkini dengan cara mengisi nilai variabel sama dengan variabel global hasilbpm. Selanjutnya data diubah menjadi string dengan perintah “String data = String(bpm, DEC);” seperti pada baris 4. Pengiriman data hanya dapat dilakukan satu persatu per karakter, oleh karena itu nilai BPM setelah menjadi variabel String akan di ubah menjadi bentuk variabel array bernama msgBuffer dan bertipedata char seperti pada baris 6 sampai baris 9. Variabel msgBuffer akan dikirim dengan perintah “mqttClient.publish” bersama dengan *channel id* dan *write api* seperti pada baris 13 sampai baris 14.

Tabel 5.5 Kode Sumber Pengiriman Nilai BPM ke Channel Thingspeak

Baris	Kode Sumber
1	<code>// Fungsi yang dibutuhkan untuk mengirimkan data ke Thingspeak</code>
2	<code>void mqttpublish(){</code>
3	<code>int bpm=hasilbpm;</code>
4	<code>String data = String(bpm, DEC);</code>
5	<code>// mengambil data dari bpm</code>
6	<code>int length = data.length();</code>
7	<code>char msgBuffer[length];</code>
8	<code>// mengubah data string menjadi karakter buffer</code>
9	<code>data.toCharArray(msgBuffer, length+1);</code>
10	<code>Serial.println(msgBuffer);</code>
11	<code>// mengirimkan data ke Channel Thingspeak</code>
12	
13	<code>mqttClient.publish("channels/437132/publish/fields/field1/GIYK8ES</code>
14	<code>XCP8P93G8",msgBuffer);</code>
15	<code>Serial.print("Data Terkirim");</code>
16	<code>Serial.println(bpm);</code>
17	<code>}</code>

5.3.2.5 Implementasi Pengiriman SMS Peringatan

Salah satu fitur sistem ini yaitu mengirim SMS peringatan ke pengguna saat BPM (*beat per minute*) tidak normal. Untuk mengimplementasikan fitur ini menggunakan code program serial write AT command. Pada Tabel 5.6 terdapat potongan program yang berfungsi sebagai pengiriman SMS peringatan. Untuk dapat pengiriman SMS peringatan, pertama deklarasi baudrate untuk mengirimkan perintah AT Command menggunakan komunikasi serial seperti pada baris 2. Perintah AT Command untuk dapat mengirimkan sms, yaitu "AT+CMGF=1\r\n", perintah tersebut mengaktifkan SMS Text Mode yang berfungsi sebagai mode untuk mengirimkan sms seperti pada baris 3. Selanjutnya pada *function* kirim sms akan menerima parameter untuk mengetahui kondisi BPM apakah normal, bradikardia atau takikardia. Terdapat perulangan selama 3 kali karena sistem ini akan mengirimkan pesan ke 3 nomer berbeda pada baris 12. Jika nilai parameter bernilai 0 maka itu bertanda bradikardia dan akan mengirimkan pesan peringatan ke 3 nomer telfon, dan jika nilai parameter bernilai 1 maka itu bertanda takikardia dan akan mengirimkan pesan peringatan ke 3 nomer telfon pada baris 14 sampai baris 20. Sistem ini juga menggunakan variabel bantu untuk mengurangi *error* saat pengiriman SMS karena terjadi penumpukan pada serial yaitu, variabel b, c, dan s seperti pada baris 23 sampai baris 26.

Tabel 5.6 Kode Sumber Pengiriman SMS Peringatan

Baris	Kode Sumber
1	<code>void kirimsms(int code){</code>
2	<code> SIM800L.begin(9600);</code>
3	<code> SIM800L.write(ATkirim[0]);</code>
4	<code> delay(500);</code>
5	<code> //</code>
6	<code> for(int i=1; i<4; i++){</code>
7	<code> char b;</code>
8	<code> String s;</code>
9	<code> String c;</code>
10	<code> Serial.println(i);</code>
11	<code> SIM800L.write(ATkirim[i]);</code>
12	<code> delay(500);</code>
13	<code> if (code == 0){</code>
14	<code> SIM800L.write("Awas Bahaya Gejala Bradikardia");</code>
15	<code> delay(500);</code>
16	<code> } else if (code==1){</code>
17	<code> SIM800L.write("Awas Bahaya Gejala Takikardia");</code>
18	<code> delay(500);</code>
19	<code> }</code>
20	<code> SIM800L.write((char)26);</code>
21	<code> Serial.println("SMS send...");</code>
22	<code> while(SIM800L.available()){</code>
23	<code> b = SIM800L.read();</code>
24	<code> s += b;</code>
25	<code> }</code>
26	<code> c = s;</code>
27	<code> delay(5000);</code>
28	<code> }</code>
29	<code> Serial.println("Ter kirim...");</code>
30	<code>}</code>



5.3.2.6 Implementasi Membalas SMS dengan Nilai BPM

Implementasi perangkat lunak dalam subbab ini berfungsi sebagai membalas sms. Membalas SMS ini yaitu menerima SMS dengan karakter “BPM”, setelah itu mengirimkan nilai denyut jantung per menit ke nomer yang sama. Jika penerimaan karakter tidak sesuai oleh diatas, maka modul GSM sim800L akan mengirimkan kalimat bahwa kata kunci yang diberikan salah. Pada Tabel 5.7 akan dijelaskan bagaimana proses untuk membalas sms jika mendapat sms dari nomer tertentu dan akan membalas smsnya. Terdapat 5 *function* untuk dapat membalas sms yaitu, *balassms()* pada baris 1 sampai baris 38 sebagai induk dari *function* lainnya, *sim800setup()* pada baris 40 sampai baris 57 berfungsi untuk mengatur aturan agar SIM800L dapat menerima sms dan membaca isi pesan dan nomer dari pengirim , *searchNo()* pada baris 59 sampai baris 84 berfungsi mengambil nilai nomer telfon, *cekContent()* pada baris 86 sampai baris 116 berfungsi sebagai melihat dan memberikan kondisi bahwa isi dari pesan tersebut berkarakter “BPM” atau tidak, dan yang terakhir *sendSMS* pada baris 188 sampai baris 144 yang berfungsi untuk mengirimkan SMS yang berisi nilai BPM terkini. Dari ke 5 *function* akan dijelaskan lebih rinci agar dapat memahami proses dari setiap *function*

Tabel 5.7 Kode Sumber Membalas SMS dengan Nilai BPM

Baris	Kode Sumber
1	<code>void balassms(){</code>
2	<code> if (SIM800L.available()) {</code>
3	<code> Serial.println("==== Start ====");</code>
4	<code> saveNo = "";</code>
5	<code> bool check = false;</code>
6	<code> delay(500);</code>
7	
8	<code> //Mencari nomor pengirim sms</code>
9	<code> saveNo = searchNo();</code>
10	<code> //Pengecekan Isi SMS</code>
11	<code> check = cekContent();</code>
12	<code> Serial.println(saveNo);</code>
13	<code> Serial.println("Nomer handphone : 0"+saveNo);</code>
14	<code> delay(500);</code>
15	
16	<code> //Standby menerima sms</code>
17	<code> for(int i=0; i<=2; i++){</code>
18	<code> SIM800L.println(AT[0]);</code>
19	<code> delay(500);}</code>
20	<code> for(int i=1; i<=2; i++){</code>
21	<code> SIM800L.println(AT[i]);</code>
22	<code> delay(500);}</code>
23	
24	<code> //Balas SMS dengan kode benar dan salah</code>
25	<code> if (check == true){</code>
26	<code> sendSMS(1, saveNo, hasilbpm);</code>
27	<code> delay(5000);</code>
28	<code> Serial.println("==== Check Inbox ====");</code>
29	<code> }</code>
30	<code> else{</code>
31	<code> sendSMS(0, saveNo, hasilbpm);</code>
32	<code> delay(5000);</code>
33	<code> Serial.println("==== Check Inbox ====");</code>
34	<code> }</code>
35	<code> sim800setup();</code>
36	<code> Serial.println("==== Finish ====");</code>



Baris	Kode Sumber
37	}
38	}
39	
40	void sim800setup(){
41	char iB;
42	String iS;
43	String iC;
44	
45	for(int i=0; i<=4; i++){
46	SIM800L.println(ATsms[i]);
47	delay(1000);
48	}
49	
50	while (SIM800L.available()) {
51	iB = SIM800L.read();
52	iS += iB;
53	}
54	
55	iC = iS;
56	delay(2000);
57	}
58	
59	String searchNo(){
60	char b;
62	String s;
63	unsigned long rest = millis();
64	while(1) {
65	if (millis() - rest > 3000){
66	break;};
67	if (SIM800L.available()) {
68	b = SIM800L.read();
69	if (b == ''){
70	while(1) {
71	b = SIM800L.read();
72	s = s + b;
73	if(b == ''){
74	int pos = s.indexOf('');
75	String optor = s.substring(3,pos);
76	return optor;
77	break;
78	}
79	}
80	break;
81	}
82	}
83	}
84	
85	
86	bool cekContent(){
87	String s;
88	char c;
89	while (SIM800L.available()) {
90	c = SIM800L.read();
91	s += c;
92	}
93	int i = 1;
94	bool contentSMS;
95	unsigned long rest = millis();
96	while(1){
97	if(String(s.charAt(i)) == "B"){
98	i = i+1;
99	if(String(s.charAt(i)) == "P"){
100	i = i+1;
101	if(String(s.charAt(i)) == "M"){

Baris	Kode Sumber
102	contentSMS = true;
103	return contentSMS;
104	break;
105	}
106	}
107	}
108	i++;
109	if(i > 50){
110	contentSMS = false;
112	return contentSMS;
113	break;
114	}
115	}
116	}
117	
118	void sendSMS(int code, String saveNo,int bpm){
119	String data;
121	data = String(bpm);
122	SIM800L.write(ATsend[0]);
123	
124	delay(2000);
125	
126	String ATc = ATsend[1] + saveNo + ATsend[2];
127	const char* ATnomer = ATc.c_str();
128	SIM800L.write(ATnomer); // AT Mencoba mengirim pesan pada
129	Nomer yang dituju
130	delay(500);
131	if(code == 0){
132	SIM800L.write("Keyword yang dimasukan salah");
133	delay(500);
134	}
135	else if (code == 1){
136	SIM800L.print("Nilai Bpm : ");
137	delay(500);
138	SIM800L.print(data);
139	delay(500);
140	}
141	SIM800L.write((char)26);
142	delay(2000);
143	Serial.println("SMS Send...");
144	}

Pada *function* `balassms()` akan dijalankan loop agar sistem dapat bersiap untuk menerima sms dan mengirimnya kembali. Pertama *function* `balassms()` akan mengecek Serial SIM800L telah aktif atau belum, jika aktif akan melanjutkan perintah dibawahnya seperti pada baris 2. Nomer akan disimpan pada variabel `saveNo` dan isi pesan akan dicek sesuai dengan karakter "BPM" atau tidak seperti pada baris 9 dan baris 11. Jika isi pesan sesuai karakter "BPM" maka akan dimasukkan pada variabel `check` bertipedata boolean seperti pada baris 25 dan baris 30. Agar mempermudah *debugging* tambahkan menampilkan nomer telfon setelah mengambil nilai nomer telfon pada serial monitor. Selanjutnya terdapat potongan kode looping untuk mengirim perintah AT Command pada SIM800L agar *standby*. Proses selanjutnya melakukan pengecekan variabel `check` yang berisi nilai sesuai dengan karakter "BPM" atau tidak. Jika variabel `check` sama dengan true maka akan mengirim sms dengan nilai BPM terkini ke nomer yang tadi sudah di simpan pada variabel `saveNo`. Jika variabel `check` sama dengan false maka

akan mengirimkan sms ke nomer tadi dengan pesan “karakter yang anda masukkan salah”.

Pada *function* sim800setup terdapat perintah untuk mengirimkan AT Command pada SIM800L yang berfungsi untuk mengatur agar SIM800 dapat menerima sms antara lain perintahnya yaitu, AT+CMGF berfungsi untuk mengaktifkan jaringan GSM/GPRS, AT+CNMI=1,2,0,0,0 berfungsi untuk mengaktifkan SIM800 agar dapat mengirimkan SMS dan menerima SMS, AT+CMGL="\REC UNREAD\" berfungsi untuk memberikan status telah terbaca yang telah dibaca, AT+CLIP=1\r berfungsi untuk Mode pengiriman SMS, AT+CMGDA="\DEL ALL\" berfungsi untuk menghapus seluruh SMS.

Pada *function* searchNo() Berfungsi untuk mengambil nilai nomer telfon dari pesan yang masuk. Terdapat beberapa variabel pembantu yaitu variabel b, s, dan rest. Variabel b berfungsi untuk mencari batas awal pesan yang diterima dalam format SIM800L, format tersebut dapat dilihat di serial monitor. Variabel s berisi sebagian isi pesan yang masuk sebelum nomer telfon sesuai dengan format SIM800L. Variabel rest berfungsi untuk memberikan batas waktu proses pengerjaan. Cara kerjanya Variabel b akan menjadi pembanding agar dapat menemukan batas awal dan batas akhir yang didalamnya ada nilai nomer telfon, tentu setelah menentukan batas awal dan akhir nomer yang didalamnya ada nilai nomer telfon dapat diambil nomernya. Setelah mengetahui batas awal dan akhir, variabel s akan di subtring untuk mengambil nilai nomer telfon yang akan dimasukkan pada variabel optor. Variabel optor akan di return sebagai nilai keluaran *function* searchNo().

Pada *function* cekContent() berfungsi sebagai membaca isi pesan dan meidentifikasi apakah karakter isi dari pesan adalah “BPM” atau tidak. Membutuhkan variabel pembantu untuk mengetahui batas awal dan batas akhir agar diproses untuk identifikasi. Variabel s digunakan sebagai penampung isi dari pesan yang masuk. Setelah itu untuk mengetahui karakter “BPM” menggunakan *method* charAt untuk mengidentifikasi awal dengan karakter B, dan selanjutnya P dan M. Setelah pesan itu sesuai dengan karakter “BPM” maka akan melakukan pengembalian Boolean yaitu true. Tetapi jika tidak terdapat karakter “BPM” akan melakukan pengembalian bernilai false.

Pada *function* sendSMS terdapat parameter yang dibutuhkan untuk mengeksekusi perintah dibawahnya. Parameter tersebut yaitu, variabel code yang berfungsi sebagai nilai kondisi apakah isi pesan sesuai dengan karakter “BPM” atau tidak yang akan dikeluarkan oleh *function* cekContent(), saveNo adalah nilai variabel yang berisi nomer telfon yang mengirimkan permintaan BPM pada sistem, dan bpm adalah variabel yang berisi nilai BPM terkini.

BAB 6 PENGUJIAN

Pada bab ini menjelaskan tentang pengujian. Pengujian yang dilakukan antara lain pengujian fungsional sistem, pengujian keakuratan sensor, pengujian komunikasi data sistem menggunakan MQTT, pengiriman sms peringatan, dan membalas sms dengan nilai BPM.

6.1 Pengujian Fungsional Sistem

Dalam pengujian ini akan dilakukan pengujian fungsional sistem. Pengujian ini membutuhkan wifi berkoneksi internet agar bisa mengirim data ke Thinspeak, dalam pengujian ini akan menggunakan *hotspot* dari *smartphone*.

6.1.1 Tujuan Pengujian

Tujuan dari pengujian fungsional sistem ini yaitu untuk menguji keseluruhan dari fitur yang ada dalam sistem ini. Dalam sistem ini akan diuji dari sensor membaca denyut jantung, sistem mengirim data ke thingspeak, mengirimkan SMS di saat kondisi denyut jantung takikardia dan bradikardia dan membalas sms dengan nilai BPM terkini.

6.1.2 Alat yang Digunakan

Pada pengujian ini menggunakan beberapa alat sebagai berikut:

1. Laptop yang sudah terdapat aplikasi arduino IDE
2. USB Kabel data
3. Puwarupa penelitian
4. *Smartphone*
5. *Hostpot* berkoneksi internet

6.1.3 Prosedur Pengujian

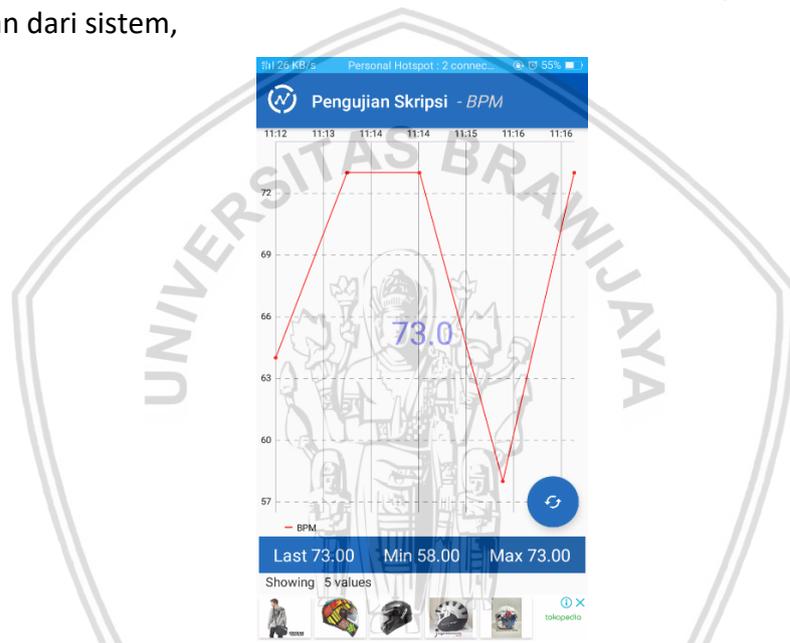
Prosedur pengujian dalam pengujian fungsional sistem, yaitu :

1. Gunakan Purwarupa sistem pada lengan manusia
2. Menghidupkan *hotspot smartphone*.
3. Membuka aplikasi Thingspeak, Isi *Channel* ID.
4. Melihat hasil dari pembacaan sensor di aplikasi Thingview.
5. Screenshot layar aplikasi.
6. Mengirimkan SMS pada sistem dengan Karakter "BPM".
7. Melihat hasil balasan SMS.
8. Screenshot pesan balasan yang berisi nilai BPM.
9. Mengamati Jika ada BPM kurang dari 60 atau lebih dari 100.

- 10. Jika ada kondisi pada point 9, lihat pesan masuk pada nomer yang terdaftar.
- 11. Screenshot pesan peringatan.

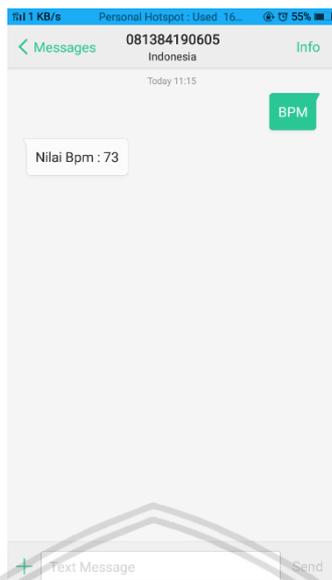
6.1.4 Hasil Pengujian

Pengujian dilakukan selama 5 menit. Dilihat dari hasil pengujian, sistem dapat mengirimkan nilai BPM ke channel thingspeak dengan baik sebanyak 5 kali. Pada aplikasi akan menampilkan grafik dari nilai BPM setiap 1 menit dan dapat melihat BPM terkini, nilai maksimum BPM, dan nilai minimum BPM selama pembacaan sensor dijalankan. Setelah melakukan pengujian fungsionalitas aplikasi android, selanjutnya mengirimkan permintaan nilai BPM dengan menggunakan karakter "BPM". Pada Gambar 6.1 menampilkan hasil dari pengujian dari sistem,



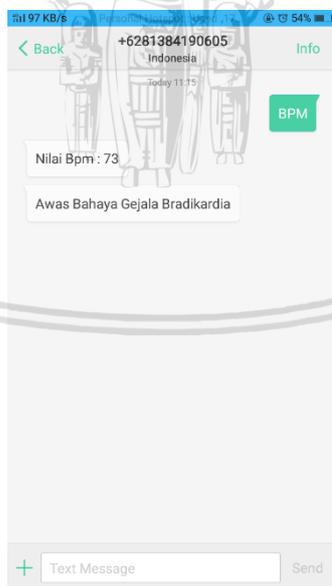
Gambar 6.1 Hasil Pengujian Fungsionalitas Aplikasi Android

Pada Gambar 6.2 menampilkan hasil dari pengujian permintaan Nilai dari BPM. Permintaan nilai BPM dapat dilakukan oleh siapapun dan nomer apapun. Setelah mengirimkan karakter "BPM" pada sistem maka sistem akan mengembalikan nilai BPM terkini dengan baik.



Gambar 6.2 Hasil Pengujian Permintaan Nilai BPM

Setelah melakukan pengujian 5 kali, terdapat nilai BPM kurang dari 60. Pada Gambar 6.3 menampilkan hasil pengujian pengiriman SMS peringatan. SMS peringatan akan dilakukan jika nilai BPM kurang dari 60 dan lebih dari 100. Setelah melakukan pengujian selama 5 kali, ternyata terdapat nilai BPM yang kurang dari 60. Sistem akan mengirimkan SMS peringatan "Awas Bahaya Gejala Bradikardia" ke 3 nomer yang terdaftar. Pengiriman SMS peringatan dapat dilakukan oleh sistem dengan baik.



Gambar 6.3 Hasil Pengujian Pengiriman SMS peringatan

6.2 Pengujian Keakuratan Pulse Sensor

Pengujian keakuratan sensor dilakukan perbandingan antara pembacaan sensor dengan pembacaan secara manual. Pembacaan secara manual dilakukan dengan menggunakan pergelangan tangan dan pembacaan sensor dilakukan pada pergelangan tangan. Dalam pengujian ini dilakukan selama 10 kali percobaan.

6.2.1 Tujuan Pengujian

Pengujian keakuratan sensor bertujuan sebagai tolak ukur keakuratan sensor pada pulse sensor. Hasil pengujian ini menghitung tingkat kesalahan saat menggunakan pulse sensor untuk membaca denyut jantung dan mengukur secara manual dengan *stopwatch*.

6.2.2 Alat yang Digunakan

Pada pengujian ini menggunakan beberapa alat sebagai berikut :

1. Laptop yang sudah terdapat aplikasi arduino IDE.
2. USB Kabel data.
3. Puwarupa penelitian.
4. *Stopwatch*

6.2.3 Prosedur Pengujian

Prosedur pengujian dalam pengujian fungsional sistem, yaitu :

1. Purwarupa sistem digunakan pada lengan subjek.
2. Saat pembacaan sensor di lengan subjek, dilakukan juga pengukuran denyut jantung secara manual.
3. Setelah 15 detik pengukuran denyut jantung secara manual dicatat.
4. Hasil pengukuran denyut jantung secara manual dicatat dan dikalikan 4 untuk mendapatkan denyut jantung selama 1 menit.
5. Tampilkan hasil kedua pengukuran dalam bentuk tabel.
6. Setelah itu lakukan perhitungan persentase kesalahan dari pengujian dengan menggunakan rumus,

$$\frac{|pengukuran\ manual - pengukuran\ sensor|}{pengukuran\ manual} \times 100\%$$

7. Persentase kesalahan dari pengujian kemudian dihitung rata-ratanya dari 5x pengambilan data.

6.2.4 Hasil Pengujian Secara Pembacaan Pulse Sensor

Dari Pengujian pembacaan sensor akan mendapatkan hasil pengukuran denyut jantung secara manual dan pengukuran menggunakan purwarupa. Hasil pengujian dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Pembacaan Pulse Sensor

Pengujian	Kondisi	Pengukuran manual (BPM)	Pengukuran sensor (BPM)	Kesalahan (%)
1	Normal	65	61	6.15
2	Normal	68	67	1.4
3	Normal	70	70	0
4	Normal	72	72	0
5	Normal	69	67	2.8
6	Normal	70	68	2.8
7	Normal	69	65	5.7
8	Normal	69	65	5.7
9	Normal	67	66	1.4
10	Normal	67	67	0
Rata – Rata Kesalahan				2.6

Pada Tabel 6.1 menampilkan hasil pengujian pembacaan *pulse sensor*. Setelah melakukan pengujian selama 10 kali percobaan didapatkan hasil pengujian yaitu persentasi dalam kesalahan dengan menggunakan *pulse sensor* yaitu sebesar 2.6. Persentasi kesalahan disebabkan keakuratan sensor masih belum dapat akurat 100%. Walaupun pengujian pembacaan sensor masih terdapat presentasi kesalahan namun dapat dikatakan sistem dapat berjalan sesuai yang diharapkan.

6.3 Pengujian Pengiriman Data ke *Channel* Thingspeak

Pada subbab ini akan menjelaskan pengiriman data BPM ke *channel* thingspeak. Pengujian ini akan dilakukan sebanyak 5 kali pengiriman dari nodemcu ke thingspeak. Pengiriman data BPM membutuhkan koneksi internet, oleh karena itu pengujian ini menggunakan *hotspot* pada *smartphone*.

6.3.1 Tujuan Pengujian

Tujuan pengujian komunikasi data menggunakan MQTT yaitu sebagai pengujian dari pengiriman data dari purwarupa sistem ke Thingspeak. Pengiriman ini dilakukan setelah data denyut jantung dibaca selama 60 detik. Pengiriman ini dilakukan menggunakan purwarupa, koneksi internet menggunakan wifi dan komputer yang terdapat aplikasi Arduino IDE. Pengujian akan dilakukan sebanyak 5 kali.

6.3.2 Alat yang Digunakan

Pada pengujian ini menggunakan beberapa alat sebagai berikut:

1. Laptop yang sudah terdapat aplikasi arduino IDE
2. USB Kabel data
3. Puwarupa penelitian
4. Smartphone
5. Hostpot berkoneksi internet

6.3.3 Prosedur Pengujian

1. Purwarupa sistem dihubungkan dengan komputer.
2. Membuka aplikasi Arduino IDE pada komputer.
3. Membuka Serial Monitor.
4. Melihat Hasil dari pengiriman data dari purwarupa ke *Channel* Thingspeak.
5. Membuka aplikasi thingview pada *smartphone*.
6. Mencatat nilai BPM di serial monitor dan aplikasi thingview.

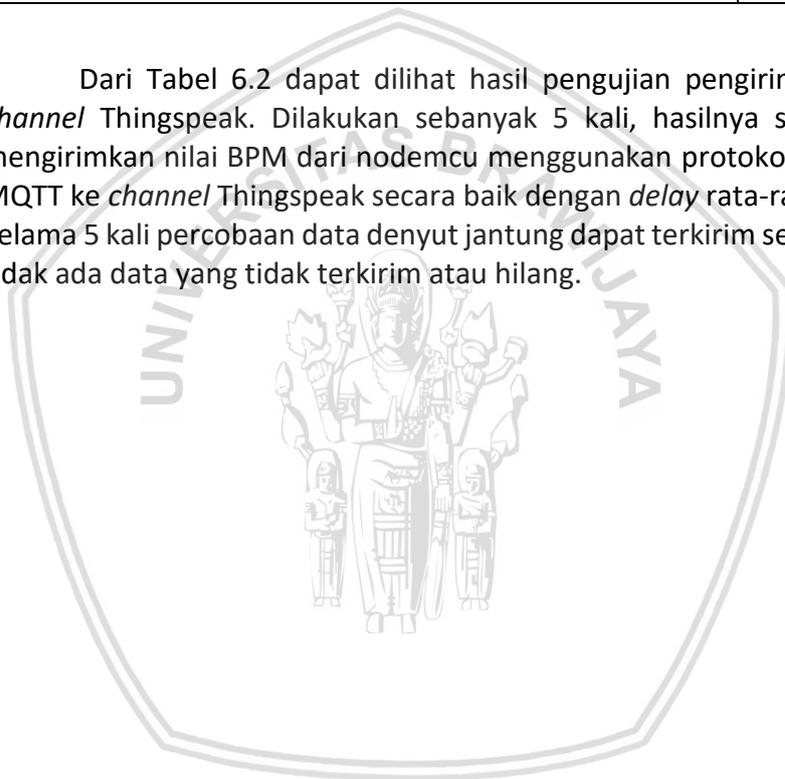
6.3.4 Hasil Pengujian Pengiriman Data ke *Channel* Thingspeak

Setelah melakukan pengujian sebanyak 5 kali untuk mengirimkan data ke *channel* Thingspeak. Berikut Pada Tabel 6.2 akan dijelaskan hasil dari pengujian. Pada Tabel 6.2 pengujian akan menjelaskan nilai bpm yang ada pada nodemcu, nilai bpm yang ada pada thingspeak dan dari kedua nilai tersebut akan dapat disimpulkan pada kolom hasil pengiriman apakah terkirim atau tidak. Pada kolom selanjutnya akan menampilkan delay yang dibutuhkan sistem untuk mengirimkan data dari nodemcu ke *channel* Thingspeak.

Tabel 6.2 Hasil Pengujian Pengiriman Data ke *Channel* Thingspeak

Pengujian	Nilai BPM Nodemcu	Nilai BPM Thingspeak	Hasil Pengiriman	Delay
1	64	64	Terkirim	1s
2	73	73	Terkirim	2s
3	73	73	Terkirim	1s
4	58	58	Terkirim	1s
5	73	73	Terkirim	1s
Rata-rata				1.2s

Dari Tabel 6.2 dapat dilihat hasil pengujian pengiriman data ke *channel* Thingspeak. Dilakukan sebanyak 5 kali, hasilnya sistem dapat mengirimkan nilai BPM dari nodemcu menggunakan protokol komunikasi MQTT ke *channel* Thingspeak secara baik dengan *delay* rata-rata 1.2 detik. Selama 5 kali percobaan data denyut jantung dapat terkirim semuanya dan tidak ada data yang tidak terkirim atau hilang.



6.4 Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung

6.4.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui apakah berhasil atau tidak pengiriman SMS peringatan. Untuk pengujian ini dilakukan secara manual yaitu, memasukkan nilai BPM secara manual untuk mendapatkan kondisi bradikardia dan takikardia. Pengujian ini akan dilakukan sebanyak 5 kali untuk pengiriman SMS peringatan Bradikardia dan 5 kali pengiriman SMS peringatan Takikardia.

6.4.2 Alat yang Digunakan

Pada pengujian ini menggunakan beberapa alat sebagai berikut:

1. Laptop yang sudah terdapat aplikasi arduino IDE
2. USB Kabel data
3. Puwarupa penelitian
4. *Smartphone*
5. *Stopwatch*

6.4.3 Prosedur Pengujian

1. Purwarupa sistem dihubungkan dengan komputer
2. Membuka aplikasi Arduino IDE pada komputer
3. Memberikan nilai BPM secara manual senilai <60 dan senilai >100 secara acak
4. *Upload* Program sampai selesai
5. Melakukan masing-masing sebanyak 5 kali percobaan nilai BPM senilai <60 dan >100 secara acak
6. Amati pada serial monitor dan *smartphone*
7. Mencatat hasil pengujian pada tabel

6.4.4 Hasil Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung

Setelah melakukan *upload* program amati pada serial monitor pengiriman data setelah melakukan pengukuran denyut jantung selama 60 detik. Untuk menghitung delay membutuhkan *stopwatch*, mulai menghitung saat tampilan pada serial monitor menunjukkan tanda data dikirim dan diakhiri saat data sudah bisa ditampilkan pada layar *smartphone*. Berikut pada Tabel 6.3 hasil dari pengujian, dilakukan masing-masing sebanyak 5 kali percobaan.

Tabel 6.3 Pengujian Pengiriman SMS Peringatan Kondisi Denyut Jantung

Pengujian	Nilai BPM (BPM)	Kondisi	Hasil Pengiriman	Delay (Second)
1	59	Takikardia	Terkirim	3s
2	50	Takikardia	Terkirim	4s
3	55	Takikardia	Terkirim	3s
4	40	Takikardia	Terkirim	5s
5	51	Takikardia	Terkirim	3s
6	101	Bradikardia	Terkirim	5s
7	120	Bradikardia	Terkirim	3s
8	125	Bradikardia	Terkirim	4s
9	111	Bradikardia	Terkirim	3s
10	123	Bradikardia	Terkirim	3s
Rata-rata				3.5s

Setelah melakukan pengujian, pada Tabel 6.3 menampilkan pesan sms yang dikirim oleh sistem jika BPM diberi nilai <60 dan nilai >100 secara acak. Sistem ini dapat melakukan pengiriman SMS peringatan dengan baik dengan delay rata-rata 3.5s. Jika BPM diberi nilai <60 maka kondisi tersebut termasuk kondisi Bradikardia, oleh karena itu sistem akan mengirimkan SMS peringatan “ Awas Bahaya Gejala Bradikardia” dan jika BPM diberi nilai >100 kondisi tersebut termasuk kondisi Takikardia, oleh karena itu sistem akan mengirmkan SMS peringatan “ Awas Bahaya Gejala Takikardia”. Untuk mengirimkan SMS peringatan, nomer telfon akan diinputkan dalam kode sumber.

6.5 Pengujian Balasan SMS dengan karakter BPM

6.5.1 Tujuan Pengujian

Tujuan dari pengujian balasan SMS dengan karakter BPM yaitu untuk menguji apakah sistem dapat membalas pesan permintaan dari nomer lain. Pengujian ini akan dilakukan 10 kali pengujian yaitu, 5 kali pengujian permintaan sms dengan karakter “BPM” dan 5 kali pengujian permintaan dengan karakter “minta BPM”.

6.5.2 Alat yang Digunakan

Pada pengujian ini menggunakan beberapa alat sebagai berikut:

1. Laptop yang sudah terdapat aplikasi arduino IDE
2. USB Kabel data
3. Puwarupa penelitian
4. *Smartphone*
5. Hostpot berkoneksi internet
6. *Stopwatch*

6.5.3 Prosedur Pengujian

1. Menghubungkan Purwarupa dengan komputer.
2. Membuka Aplikasi Arduino IDE.
3. Membuka Serial monitor.
4. Melakukan pengiriman sms dengan karakter “BPM sebanyak 5 kali
5. Mengamati serial monitor dan amati balasan SMS
6. Mencatat hasil pengujian pada tabel
7. Melakukan pengiriman sms dengan karakter “minta BPM sebanyak 5 kali
8. Mengamati serial monitor dan amati balasan SMS
9. Mencatat hasil pengujian pada tabel

6.5.4 Hasil Pengujian

Pada Tabel 6.4 menampilkan hasil pengujian balasan sms dengan karakter BPM. Setelah melakukan permintaan nilai BPM dengan mengirimkan karakter “BPM” pada sistem, sistem akan membalas pesan dengan nilai BPM terkini dan melakukan permintaan nilai BPM dengan karakter yang salah dengan karakter “Minta BPM”. Pada Tabel 6.4 pengujian terdapat isi karakter SMS dari peminta nilai BPM yang berisi karakter yang dikirimkan oleh peminta BPM, isi balasan SMS yang berisi balasan SMS dari sistem, pengiriman balasan menjelaskan apakah isi balasan sms telah terkirim atau tidak terkirim, dan yang terakhir delay adalah waktu yang dibutuhkan sistem untuk mengirimkan SMS.

Tabel 6.4 Hasil Pengujian Balasan SMS dengan karakter BPM

Pengujian	Isi Karakter SMS	Isi Balasan SMS	Pengiriman Balasan	Delay (Second)
1	BPM	Nilai Bpm : 68	Terkirim	3s
2	BPM	Nilai Bpm : 69	Terkirim	2s
3	BPM	Nilai Bpm : 70	Terkirim	3s
4	BPM	Nilai Bpm : 69	Terkirim	4s
5	BPM	Nilai Bpm : 71	Terkirim	4s
6	Minta BPM	keyword yang dimasukkan salah	Terkirim	3s
7	Minta BPM	keyword yang dimasukkan salah	Terkirim	2s
8	Minta BPM	keyword yang dimasukkan salah	Terkirim	3s
9	Minta BPM	keyword yang dimasukkan salah	Terkirim	2s
10	Minta BPM	keyword yang dimasukkan salah	Terkirim	3s
Rata – Rata Kesalahan				2.9s

Setelah melakukan pengujian balasan sms dengan karakter yang benar yaitu karakter “BPM”, pengujian selanjutnya pada mengirimkan sms ke sistem dengan karakter “minta BPM”. Pada Tabel 6.4 menampilkan hasil pengujian dengan karakter “minta BPM” dan sistem dapat mengirimkan balasan sms tidak sesuai dengan karakter yg dibutuhkan dengan delay rata-rata 2.6s . Jika sistem akan membaca tidak sesuai dengan karakter yang menjadi kata kunci maka sistem akan mengirimkan SMS “ keyword yang dimasukkan salah”.

BAB 7 PENUTUP

Bab penutup ini merupakan bab terakhir dari penelitian ini. Bab ini menjelaskan kesimpulan dari keseluruhan penelitian dan saran untuk penelitian selanjutnya dari topik yang diangkat pada penelitian ini. Kesimpulan akan diambil berdasarkan hasil dari pengerjaan purwarupa sistem sampai hasil pengujian dari sistem.

7.1 Kesimpulan

Kesimpulan yang diambil dari hasil perancangan, implementasi, dan pengujian sebagai berikut :

1. Sistem *monitoring* denyut jantung pada penelitian ini dapat disimpulkan bahwa penelitian ini berhasil membuat sistem sesuai dengan yang diharapkan. Dengan menggunakan nodemcu sebagai mikrokontroler, pulse sensor sebagai sensor pengambilan nilai denyut jantung, dan SIM800L sebagai modul gsm untuk mengirminkan sms dan membalas sms dengan karakter "BPM".
2. Pengolahan sinyal data denyut jantung pada pulse sensor selama 60 detik dapat dilakukan oleh mikrokontroler nodemcu. Pembacaan denyut jantung selama 60 detik dirancang menggunakan mikrokontroler nodemcu dan pulse sensor. Hasil pengujian pembacaan denyut jantung menggunakan pulse sensor menunjukkan kesalahan pembacaan pulse sensor sebesar 2.6%.
3. Sistem *monitoring* denyut jantung pada penelitian ini dapat disimpulkan bahwa penelitian ini berhasil menggunakan *protocol* MQTT sebagai alat komunikasi untuk pengiriman data. Dari perancangan sampai pengujian sistem ini dapat melakukan pengiriman data denyut jantung ke *channel* thingspeak berjalan sesuai dengan yang diharapkan. Hasil pengujian pengiriman data denyut jantung menggunakan MQTT sebanyak 5 kali dapat terkirim dengan delay rata-rata 1.2s.
4. Saat sistem mendeteksi kondisi bradikardia dan takikardia maka sistem akan mengirimkan SMS peringatan. Sistem ini dapat mengirimkan SMS peringatan saat kondisi bradikardia dan takikardia sesuai dengan harapan. Hasil pengujian yang dilakukan sebanyak 5 kali masing-masing dari kondisi, sistem dapat mengirimkan SMS peringatan dengan baik dengan delay rata-rata 3.5s.

7.2 Saran

Saran diberikan untuk pengembangan penelitian ini selanjutnya sebagai berikut:

1. Pengembangan selanjutnya dapat dilakukan untuk memasukkan nomer telfon yang akan dikirim SMS peringatan tidak dilakukan secara manual pada kodingan, tetapi dapat dilakukan dengan mengirimkan sms pada sistem.
2. Pengembangan selanjutnya menggunakan peletakan sensor di ujung jari tangan atau pada daun telinga.



Daftar Pustaka

- Abjali.c, 2015. *Arrythmia*. Indiana : s.n.
- Ali, M., 2009. *Pendidikan Untuk pembangunan Nasional*. Cetakan I ed. Jakarta: Imtima.
- Bartsch, R., 2017. *Drones in Society: Exploring the Strange New World of Unmanned Aircraft*. New York: Routledge.
- Bronzino, J. D., 2000. *Biomedical Engineering Handbook*. Danvers, Ma, USA: CRC PRes LLC.
- Choudhuri, K. B. R., 2017. *Learn Arduino Prototyping in 10 days*. Birmingham: Packt Publishing Ltd..
- Delp, M. H., 1996. *Mayor Diagnosis Fisik*. Philadelphia: Sauders Company.
- Dennis, A. K., 2015. *Raspberry Pi Home Automation with Arduino - Second Edition*. Birmingham: Packt Publishing Ltd..
- García, G., 2018. *Protocols and Applications for the Industrial Internet of Things*. Hershey PA: IGI GLoBal.
- Gray, H. H., 2005. *Kardiologi*. Edisi Keempat ed. Jakarta: Penerbit Erlangga.
- Hassanien, A. E., 2018. *The International Conference on Advanced Machine Learning Technologies and Applications*. Switzerland ed. Cham: Springer International Publisher.
- Manurung, N., 2017. *Sistem Endokrin*. Yogyakarta: Deepulish.
- Mikrajuddin, 2007. *IPA TERPADU SMP dan MTs 3A*. s.l.:Penerbit Erlangga.
- Molloy, D., 2016. *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*. Indianapolis: John Wiley & Sons, Inc..
- Morton, P. G., 2005. *Panduanb Pemeriksaan Kesehatan*. I ed. Jakarta: Penerbit buku Kedokteran EGC.
- Muffichatum, 2006. *Hubungan Antara Tekanan Panas, Denyut Nadi dan. Produktivitas Kerja Pada Pekerja Pandai Besi Paguyuban Wesi Aji*. s.l.:s.n.
- Schwartz, M., 2017. *ESP8266 Internet of Things Cookbook*. Birmingam: Packt Publishing Ltd..
- Wijaya, S. P., Christiyono, Y. & Sukiswo, 2010. Alat Pelacak Lokasi Berbasis GPS Via Komunikasi Seluler. *Transmisi*, pp. 82-86.
- World Famous Electronics llc, 2015. *Pulse sensor Getting Started Guide*. New York: World Famous Electronics llc.

