

Sistem Pemantauan Pasien berbasis IoT menggunakan ESP8266 dan Arduino

Stefanus Adi Putra Pratama, Djoni Haryadi Setiabudi
Program Studi Informatika,
Fakultas Teknologi Industri,
Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp (031) – 2983455, Fax. (031) - 8417658
stefanus.apps@gmail.com, djonihs@peter.petra.ac.id

ABSTRAK

Pemantauan kesehatan adalah masalah utama di dunia saat ini. Karena kurangnya pemantauan kesehatan yang tepat, pasien dapat menderita masalah kesehatan yang serius. Ada banyak perangkat jaman sekarang untuk memantau kesehatan pasien yang dimanfaatkan oleh pakar kesehatan untuk mengawasi pasien. Namun karena tidak terintegrasi dengan IoT maka metode ini dapat menambah beban tenaga medis untuk melakukan pemeriksaan dan tidak bisa dilakukan kapanpun dan dimanapun. Penelitian ini dilakukan untuk memudahkan seorang dokter maupun kerabat dari seorang penderita gangguan jantung untuk melakukan pengawasan secara *real Time*. Perangkat Arduino akan melakukan pembacaan pada tiap sensor dan dengan menggunakan layanan dari *ThingSpeak* dan IFTTT, pembacaan data dapat diterima dan divisualisasikan, kemudian secara otomatis dapat dicatat dalam Google Sheets sehingga dapat di akses baik melalui internet maupun aplikasi *mobile*. Kombinasi antara IoT dan Arduino adalah salah satu cara untuk mengenalkan metode IoT dalam dunia kesehatan. Dengan memanfaatkan IoT, pemantauan dapat lebih dipermudah karena tidak terbatas ruang. penyampaian informasi yang *real-time* juga memudahkan pengambilan keputusan bila terjadi keadaan kritis.

Kata Kunci : Pemantauan Pasien, Arduino, ESP8266, *ThingSpeak*, IFTTT

ABSTRACT

Health monitoring is a major problem in the world today. Due to lack of proper health monitoring, patients can suffer serious health problems. There are many devices today to monitor the health of patients that health experts use to monitor their patients but because it is not integrated with IoT, this method can add to the burden of medical personnel to conduct examinations and cannot be done anytime, anywhere. This research was conducted to make it easier for a doctor and a relative of a person with heart problems to conduct surveillance in Real Time. The Arduino device will read each sensor and by using services from ThingSpeak and IFTTT, data readings can be received and visualized, then automatically recorded on Google Sheets so that they can be accessed either via the internet or mobile applications. The combination of IoT and Arduino is one way to introduce the IoT method. By utilizing IoT, monitoring can be made easier because it is not limited in space, real-time delivery of information also facilitates decision making in the event of a critical situation.

Keywords : Patient Monitoring, Arduino, ESP8266, *ThingSpeak*, IFTTT.

1. PENDAHULUAN

Pemantauan kesehatan adalah masalah utama di dunia saat ini. Karena kurangnya pemantauan kesehatan yang tepat, pasien dapat menderita masalah kesehatan yang serius. Ada banyak perangkat jaman sekarang untuk memantau kesehatan pasien yang pakar kesehatan memanfaatkan untuk mengawasi pasien mereka, seperti Holter Monitor. Holter Monitor adalah perangkat kecil yang dapat dikenakan dan dapat melacak irama jantung seseorang. Untuk Memonitor detak jantung biasanya seorang pasien butuh untuk menggunakan alat Holter selama 1 sampai 2 hari [1]. Dengan banyaknya teknologi perawatan kesehatan baru, IoT dapat dengan cepat merevolusi industri perawatan kesehatan.

Selanjutnya Elektrokardiogram (EKG) adalah tes sederhana untuk mengukur dan merekam aktivitas listrik jantung. Tes ini menggunakan mesin pendeteksi impuls listrik yang disebut elektrokardiograf. Elektrokardiograf akan menerjemahkan impuls listrik menjadi grafik yang ditampilkan pada layar pemantau, namun karena tidak terintegrasi dengan IoT maka metode ini dapat menambah beban tenaga medis untuk melakukan pemeriksaan dan tidak bisa dilakukan kapanpun dan dimanapun.

Berdasarkan latar belakang diatas maka pada skripsi ini akan dilakukan penelitian tentang Sistem Pemantauan Kesehatan berbasis IoT yang memiliki kelebihan yaitu akan mencatat detak jantung dan suhu tubuh pasien berdasarkan usia serta mengirimkan peringatan *email* setiap kali pembacaan data detak jantung melampaui nilai kritis. Detak jantung dan pembacaan suhu tubuh dicatat di *Spreadsheet*. Kemudian dengan bantuan aplikasi *Webhook* yang merupakan *HTTP Callback* akan memicu IFTTT yang berfungsi sebagai *automation* dan *push notification* [2] ketika pasien kritis sehingga kesehatan pasien dapat dipantau dari mana saja melalui internet.

2. LANDASAN TEORI

2.1. Sistem Hardware

Dalam pembuatan sistem pemantauan pasien ini akan digunakan Arduino. Arduino adalah mikrokontroler yang berfungsi sebagai pengatur perangkat-perangkat lain yang terhubung dengan board arduino. Arduino bersifat *open-source* sehingga dapat dengan mudah dikembangkan oleh semua orang dengan bebas. Arduino yang akan digunakan dalam pembuatann sistem pemantauan pasien adalah Arduino Uno. Arduino Uno merupakan *board* mikrokontroler berbasis ATmega328P.



Gambar 1. Arduino Uno

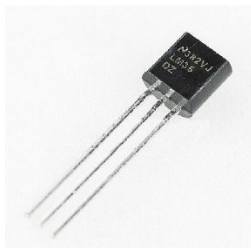
Untuk pemrograman penulis menggunakan arduino IDE yang memakai bahasa C dan memiliki *library* yang dapat ditambahkan sesuai kebutuhan dan dapat dipanggil pada saat awal sesuai fungsi yang diinginkan. Dalam pembuatan sistem pemantauan pasien digunakan beberapa *module* pendukung Arduino antara lain :

ESP8266 *Wi-Fi* Module adalah module Wifi yang sebagian besar digunakan untuk pengembangan aplikasi IoT [3]. Dalam penelitian ini ESP8266 digunakan untuk menghubungkan perangkat langsung dengan *Wi-Fi* menggunakan *AT Command*.



Gambar 2. ESP8266 Wi-Fi Module

LM35 *Temperature Sensor* digunakan untuk mengukur suhu tubuh pasien. Kisaran suhu pengoperasiannya adalah dari -55°C hingga 150°C . LM35 dapat dioperasikan menggunakan pin 5V atau 3.3V Arduino.



Gambar 3. LM35 Temperature Sensor

Pulse Heart rate sensor adalah sensor pembaca detak jantung plug-and-play untuk Arduino. dalam penelitian ini Pulse Heart rate sensor digunakan untuk mengukur detak jantung pasien. dibagian depan sensor terdapat logo berbentuk hati, sisi inilah yang melekat pada kulit pasien. di bagian depan sensor ada lubang kecil dimana LED menyala dari belakang. LED menyinari ujung jari dan sensor membaca jumlah cahaya yang memantul kembali.



Gambar 4. Pulse Heart Rate Sensor

Push Button adalah komponen yang menghubungkan antara 2 titik dalam circuit ketika ditekan. Dalam penelitian ini digunakan untuk tambahan fitur yaitu Panic Button sehingga pasien dapat memberikan peringatan langsung kepada dokter tanpa menunggu hasil pembacaan data di *ThingSpeak*.



Gambar 5. Push Button

2.2. Aplikasi

2.2.1 ThingSpeak

ThingSpeak adalah platform analisis berbasis IoT yang dapat memvisualisasi dan menganalisa lalu lintas data. Dengan menggunakan situs *ThingSpeak*, penulis dapat memonitor data data pasien seperti suhu tubuh dan temperature tubuh. kemudian *ThingSpeak* mengendalikan sistem melalui Internet menggunakan *Channels* dan halaman web yang disediakan oleh *ThingSpeak*. *ThingSpeak* mengumpulkan data dari sensor, menganalisis serta memvisualisasikan data serta memicu reaksi jika pembacaan data melampaui batas tertentu



Gambar 6. ThingSpeak

2.2.2 IFTTT (If This Then That)

Platform IFTTT digunakan untuk menghubungkan *ThingSpeak* ke layanan email sehingga pesan peringatan dapat dikirim kapan pun pasien berada dalam kondisi kritis. Banyak *applets* yang mendukung keperluan *IoT* sudah disediakan di IFTTT untuk

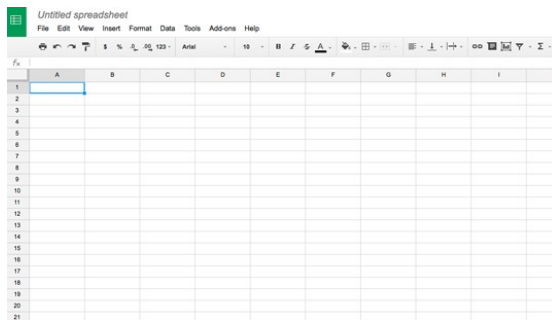
automation alat seperti notifikasi [2]. Kemudian dengan memanfaatkan aplikasi IFTTT akan memberikan *alert* kepada dokter pengawas. Perancangan juga akan memanfaatkan aplikasi IFTTT yang tersedia di *Google Store*.



Gambar 7. IFTTT

2.2.3 Google Sheets

Google Sheet adalah aplikasi *Spreadsheet Online* yang memungkinkan pengguna untuk membuat dan memformat sheet [4]. Dalam penelitian ini, aplikasi Google Sheet digunakan sebagai fasilitas penyedia *Spreadsheet* untuk mencatat data pasien.



Gambar 8. Google Sheet

2.2.4 Library

Berikut adalah *Library* yang digunakan untuk pembuatan perangkat ini : 1) ESP8266: Digunakan untuk mengontrol modul Wi-Fi sehingga memungkinkan Arduino dapat tersambung dengan internet menggunakan modul tersebut. 2) *Pulse Rate Sensor*: Digunakan pulse sensor agar dapat terhubung dengan Arduino. 3) Timer: Digunakan untuk mengatur interval pengambilan data.

2.3. Tinjauan Studi

Menurut Muhlis Agung Saputro et al. menyatakan bahwa “akurasi perhitungan menggunakan pulse sensor dan sensor suhu LM35 terbilang akurat. Rata-rata error yang dihasilkan dari perhitungan detak jantung menggunakan pulse sensor sebanyak 2.83%. Sedangkan rata-rata *error* yang dihasilkan pada perhitungan suhu tubuh menggunakan sensor suhu LM35 sebanyak 0.72%.” [5]

Menurut Muhammad Ridhan Firdaus bahwa dalam penggunaan alat monitoring detak jantung, seorang pasien tidak melakukan

banyak pergerakan karena dapat menyebabkan timbulnya *Noise* yang dapat merubah hasil sinyal BPM. Oleh karena itu dalam penelitian ini akan membahas tentang pengawasan detak jantung pasien ketika beristirahat [6]

3. ANALISIS DESAIN SISTEM

3.1 Analisis Sistem lama

Pemantauan detak jantung pada seseorang sebenarnya bisa dilakukan secara manual dan tanpa alat, namun metode yang digunakan kurang akurat yaitu dengan cara menghitung detak jantung menggunakan jari dan dihitung jumlah detak jantung dalam waktu 15 detik kemudian hasilnya dikalikan 4. Metode ini memang mudah diterapkan namun kurang maksimal hasilnya karena selain tidak mempertimbangkan suhu tubuh seorang pasien, seorang dokter haruslah mencatat dan menghitung secara langsung.

Selain secara manual, ada juga penghitungan detak jantung yang menggunakan alat EKG (Elektrokardiogram) yang dapat mengukur dan merekam aktifitas detak jantung. Namun alat ini belum terintegrasi dengan IoT sehingga untuk melakukan pengecekan dan pencatatan detak jantung tetap harus dalam pengawasan tenaga ahli seperti dokter atau perawat.

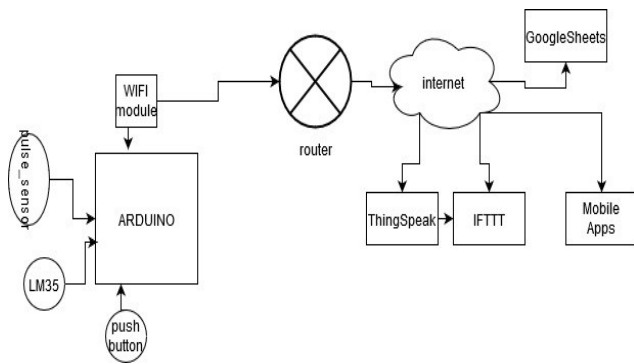
Berdasarkan latar belakang di atas, maka pada skripsi ini akan dicoba untuk membuat perangkat yang mempermudah seorang dokter dalam melakukan pengawasan detak jantung kepada pasiennya secara *real-time* berdasarkan umur seorang pasien, pengawasannya dapat dilakukan dimanapun dan kapanpun karena perangkat ini akan terintegrasi dengan IoT. Selain itu dokter bisa mendapatkan peringatan tanda darurat secara langsung melalui email dan aplikasi IFTTT jika pembacaan detak jantung melebihi atau dibawah normal. Dalam pembuatan perangkat, dibutuhkan beberapa komponen listrik untuk menjalankannya, antara lain Arduino Uno, *LM35 Temperature Sensor*, *Pulse Rate Sensor*, *Wi-Fi Module* dan *Push Button*. Masing masing perangkat memiliki peran dan fungsi, seperti Arduino Uno yang menjadi pusat kontrol dan sumber daya bagi komponen lain, *LM35 Temperature Sensor* yang berperan untuk mendeteksi temperatur tubuh pasien, *Pulse rate sensor* yang berperan untuk mendeteksi detak jantung pasien, dan *push button* sebagai tombol panik untuk pasien jika pasien merasa dalam kondisi darurat.

Arduino Uno juga dilengkapi dengan *Wi-Fi Module* yang berfungsi untuk menghubungkan perangkat dengan Internet sehingga dapat melakukan pencatatan pada *Spreadsheet* serta automation IFTTT.

3.2 Desain

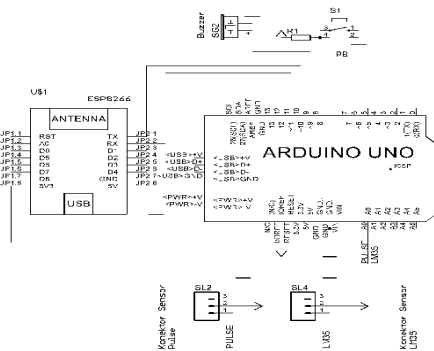
Desain perangkat ini meliputi desain rangkaian listrik, desain arsitektur sistem dan *flowchart* cara kerja perangkat untuk berhubungan dengan IFTTT, *Spreadsheet* serta *email* sehingga dapat memberikan laporan. Seperti pada gambar 9, pencatatan dimulai dari Pulse Sensor yang langsung terhubung pada pasien, kemudian dengan menggunakan WiFi Module, perangkat dapat terhubung dengan router yang kemudian akan menghubungkan ke internet sehingga *ThingSpeak* dapat melakukan monitoring, kemudian dengan menggunakan aplikasi ThingHTTP akan

memicu IFTTT untuk melakukan pencatatan data ke Spreadsheet dan notifikasi.



Gambar 9. Desain Arsitektur Sistem

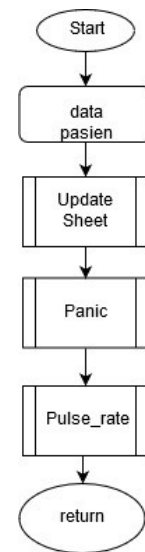
3.2.1 Rangkaian Listrik



Gambar 10. Rangkaian Listrik

Gambar 10 merupakan gambar rangkaian listrik pada perangkat Arduino di Gambar 9 sebelumnya. Pulse sensor dan Push button adalah perangkat yang langsung berinteraksi dengan user, kemudian LM35 dapat mendeteksi temperature tubuh dan ESP8266 untuk menghubungkan perangkat ke router yang tersambung ke internet. Koneksi antar komponen ke Board Arduino antara lain adalah : *Signal Pin* pada *Pulse Sensor* tersambung pada A0 di Arduino Uno, kemudian VCC Pin dari *Pulse Sensor* Tersambung pada 5V Arduino Uno, dan koneksi terakhir dari *pulse sensor* adalah pin GND tersambung pada GND Arduino. Koneksi selanjutnya adalah Vout pada LM35 *Temperature Sensor* terhubung pada A1 di Arduino Uno. Kemudian koneksi Tx pada ESP8266 terhubung pada pin 10, Rx pada ESP8266 terhubung pada pin 11, CH_PD dan VCC terhubung pada 3.3 V, dan koneksi terakhir dari ESP8266 adalah pin GND tersambung pada pin GND pada Arduino. Kemudian untuk perangkat terakhir adalah push button yang terhubung dengan digital pin 8 pada Arduino.

3.2.1 Flowchart Utama Perangkat



Gambar 11. Flowchart Utama

Gambar 11 menjelaskan tentang flowchart utama perangkat, dimana data pasien diterima melalui sensor, kemudian dimonitor oleh *ThingSpeak* dan dicatat ke *Google Sheet*. Langkah selanjutnya adalah pengecekan panic, panic hanya memerlukan data dari tombol panic saja. Kemudian dilakukan pengecekan pulse rate dari data detak jantung yang didapatkan dari pasien.

4. IMPLEMENTASI SISTEM

4.1 Pemrograman Arduino

Meliputi Source code pada Arduino Uno untuk melakukan pengambilan data pada pasien. Langkah pertama adalah melakukan pengecekan apakah Sensor dan ESP8266 sudah terhubung.

4.1.1 Pengecekan Sensor dan ESP8266

Meliputi Source code pada Arduino Uno untuk melakukan pengambilan data pada pasien. Langkah pertama adalah melakukan pengecekan apakah Sensor dan ESP8266 sudah terhubung.

4.1.2 Fungsi Pengolahan Data

Fungsi selanjutnya adalah untuk melakukan pengolahan data yang diambil dari pasien dan dikirim ke ESP8266.

4.2 Pemrograman ESP8266

Meliputi Source code pada ESP8266 untuk melakukan pengiriman data ke *ThingSpeak*.

4.3 Setting pada IFTTT

4.3.1 Pembuatan Link untuk melakukan Trigger Event

Pada bagian service di web IFTTT, cari service webhooks yang digunakan untuk melakukan fungsi web request. Di menu webhooks, pilih opsi *documentation*. Kemudian pada EventName akan menggunakan name event *patient_info* untuk melakukan trigger request *patient_info* dan *panic* untuk fungsi *panic*. Untuk merubah EventName hanya dilakukan perubahan pada event name saja, Link keseluruhan akan tetap sama

4.4 Setting pada *ThingSpeak*

4.4.1 Membuat channel untuk menampung data dari *Arduino*

Field 1 pada channel *ThingSpeak* digunakan untuk menampung data heart rate, kemudian field 2 digunakan untuk menampung data temperature dan field 3 digunakan untuk menampung data push button.

4.5 Pembuatan Fungsi Pulse Rate

Menjelaskan tentang proses reaksi *alert* jika data yang didapat dari sensor detak jantung yang terhubung pada pasien tercatat lebih dari 60.

4.5.1 Pembuatan Applets *patient_info*

Pembuatan applets *patient_info* yang nantinya berhubungan dengan service react di *ThingSpeak* untuk melakukan update data di Google Sheets. Value yang akan ditampilkan akan disesuaikan dengan fields yang ada pada Server *ThingSpeak*.

4.5.2 Pembuatan Service *ThingHTTP patient_monitor*

Pembuatan service *ThingHTTP patient_monitor* pada server *ThingSpeak* sehingga dapat berhubungan dengan applets *patient_info* pada IFTTT.

4.5.3 Pembuatan Service React *patient_info*

Pembuatan Service React pada server *ThingSpeak* sehingga dapat berbubungan dengan Applets *patient_info* yang sudah dibuat menggunakan service webhooks pada IFTTT.

4.6 Pembuatan Fungsi Panic

4.6.1 Pembuatan Applets *Panic*

Pembuatan *applets panic* yang akan berhubungan dengan *service react panic* di *ThingSpeak* untuk melakukan pengiriman *Email* yang sudah ditentukan jika tombol *panic* ditekan. Ketika tombol *panic* ditekan, maka value *panic* akan menjadi *true* yang kemudian akan tercatat pada *ThingSpeak* dengan angka 1.

4.6.2 Pembuatan Service *ThingHTTP panic*

Pembuatan service *ThingHTTP panic* pada server *ThingSpeak* sehingga dapat berhubungan dengan applets *panic* pada IFTTT.

4.6.3 Pembuatan Service React *panic*

Pembuatan Service React pada server *ThingSpeak* sehingga dapat berbubungan dengan Applets *patient_info* yang sudah dibuat menggunakan service webhooks pada IFTTT.

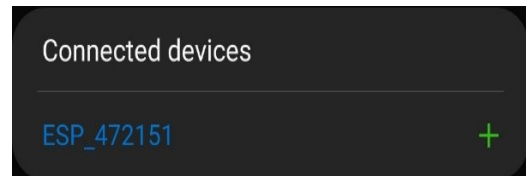
5. PENGUJIAN SISTEM

5.1 Pengujian Aplikasi

Pengujian perangkat pendeteksi detak jantung dan suhu tubuh ini dilakukan dengan cara percobaan pada sensor sensor dan push button yang ada pada perangkat, koneksi perangkat dengan server *ThingSpeak*, dan apakah fungsi fungsi *ThingSpeak* dan IFTTT sudah berjalan.

5.1.1 Pengujian Koneksi ke Internet

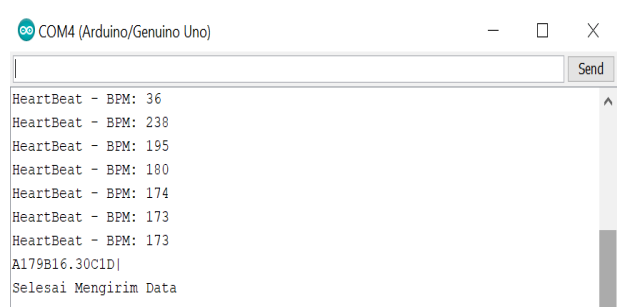
Dalam penggunaan perangkat dibutuhkan koneksi internet yang pada penelitian ini akan menggunakan koneksi dengan rata rata kecepatan *download* 10.79 Mbps dan rata-rata kecepatan *upload* 1.25 Mbps, untuk itu dilakukan pengujian apakah perangkat sudah tersambung ke WiFi dan ke api.*ThingSpeak.com*. koneksi dibutuhkan agar data yang sudah didapatkan oleh perangkat dapat di kirim ke server *ThingSpeak*.



Gambar 12. Perangkat dan WiFi terhubung

5.1.2 Pengujian Pembacaan Data Pasien

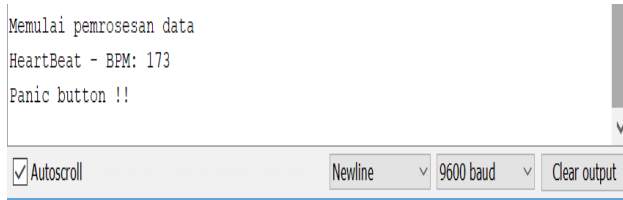
Pengujian pembacaan data dilakukan dengan cara menjalankan code pada *Arduino IDE* dengan serial monitor. Serial Monitor akan menampilkan *value* detak jantung setiap pembacaan dan *value* temperatur akan ditampilkan juga di akhir pembacaan sebelum dikirimkan ke server *ThingSpeak*.



Gambar 13. Value dari data-data yang akan dikirim ke server *ThingSpeak*

5.1.3 Pengujian pembacaan Push Button

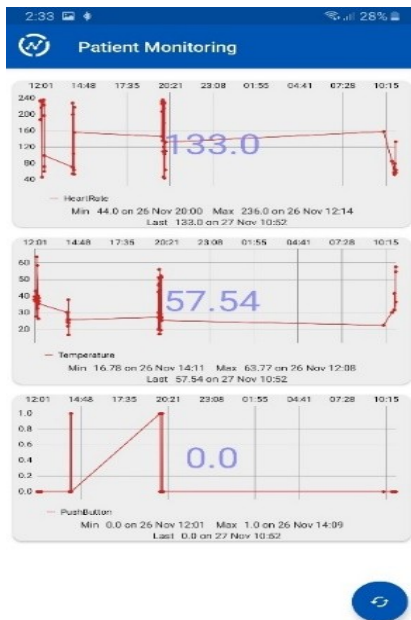
Push Button digunakan untuk menjalankan fungsi panic, dimana jika button di tekan akan terdengar bunyi beep yang dihasilkan oleh buzzer yang sudah terpasang pada perangkat. Kemudian push button juga akan menghasilkan value antara 0 atau 1 yang akan dikirim ke server *ThingSpeak* untuk menjalankan fungsi panic.



Gambar 14. Tombol push button pada perangkat

5.1.4 Pengujian Channel *ThingSpeak*

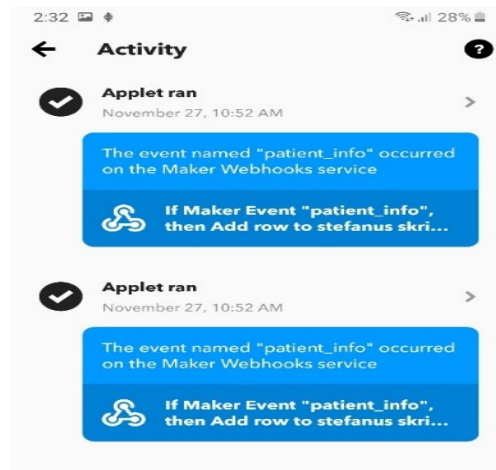
Ketika pembacaan data sudah berjalan pada perangkat, maka data data tersebut akan dikirimkan ke server *ThingSpeak* untuk dicatat dan ditampilkan visualisasinya berupa charts. Chart data dapat di akses melalui channel *ThingSpeak* baik dari web maupun aplikasi *ThingSpeak* di mobile device.



Gambar 15. Visualisasi data di *ThingSpeak*

5.1.5 Pengujian Fitur *patient_info*

Ketika *patient_info* berjalan maka akan muncul notifikasi pada aplikasi IFTTT di mobile device, kemudian akan ada activity yang memberikan informasi kapan fungsi ini dijalankan serta hasil fungsi *patient_info* ini dapat langsung di akses melalui Google Sheets. Data otomatis terupdate ketika fungsi dijalankan.



Gambar 16. Activity *patient_info*

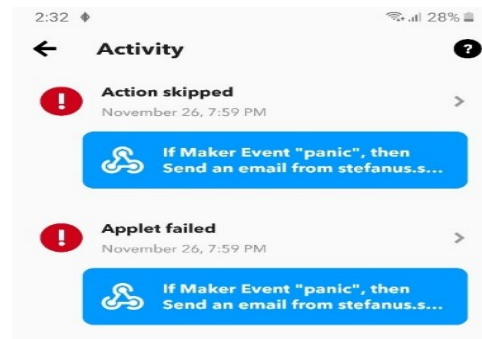
Format yang dihasilkan di Google Sheets berupa Tanggal, Fungsi atau Event IFTTT yang sedang mengirimkan data, Heart Rate pada kolom ketika dan Temperatur pada kolom keempat.

9	November 27, 2019 at 10:52AM	patient_info	64	54.
10	November 27, 2019 at 10:52AM	patient_info	133	57.

Gambar 17. data hasil pembacaan tampil pada Google Sheets

5.1.6 Pengujian Fitur *panic*

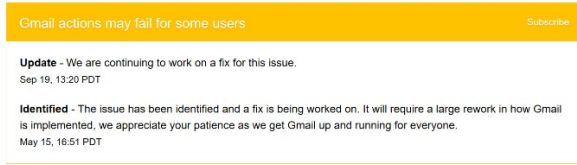
Sama seperti Fungsi *patient_info*, fitur *panic* juga menghasilkan notifikasi bila fungsi dijalankan dengan informasinya akan di tampilkan di bagian Activity. Namun untuk saat ini terjadi masalah teknis antara IFTTT dan Gmail sehingga fungsi tidak dapat berjalan semestinya.



Gambar 18. Pesan error fungsi tidak berjalan

Fungsi tidak berjalan dengan baik karena terjadi error dengan Service Gmail dan IFTTT. Berdasarkan pernyataan resmi dari pihak IFTTT pada tanggal 19 September lalu bahwa error pada service gmail sudah diidentifikasi dan sedang dilakukan penanganan agar service dapat berjalan kembali.

IFTTT



Gambar 19. Pernyataan resmi IFTTT tentang error Service Gmail

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari pengujian sistem, dapat disimpulkan beberapa hal berikut :
1) Dengan memanfaatkan IoT, pengecekan detak jantung pasien dapat dilakukan secara *remote* sehingga mempermudah dilakukannya pengawasan. Kebanyakan perangkat pengecekan kesehatan sampai saat ini masih *offline*.
2) Dengan rata-rata kecepatan *download* 10.79 Mbps dan rata-rata kecepatan *upload* 1.25 Mbps, data pembacaan dari Arduino rata-rata dapat di terima kurang dari satu menit sekali menurut pembacaan waktu yang ada pada laporan di Google Sheets.
3) Ketika data sudah diterima oleh *ThingSpeak*, notifikasi dapat langsung di terima dan laporan detak jantung sudah otomatis ter *update* pada google Sheets dan laporan dapat segera di akses baik *web-based* maupun *mobile application*.
4) Tidak dapat melakukan perbaikan segera bila terjadi kendala pada services yang disediakan baik oleh Thingpeak maupun IFTTT.

6.2 Saran

Berikut merupakan beberapa hal yang dapat dijadikan saran untuk pengembangan aplikasi selanjutnya :
1) Penggunaan sensor yang lebih canggih sehingga pembacaan data menjadi lebih akurat.
2) Mikrocontroller Arduino didukung oleh banyak sekali komponen dan perangkat kelistrikan dan sensor sensor lain sehingga dapat dengan mudah melakukan penambahan fitur.

7. DAFTAR PUSTAKA

- [1] Holter monitor - Mayo Clinic. (2017, March 28). Retrieved from <https://www.mayoclinic.org/tests-procedures/holter-monitor/about/pac-20385039>
- [2] IFTTT. (n.d.). IFTTT. Retrieved from <https://ifttt.com/about>
- [3] SP8266 WiFi Module | Sensors & Modules. (2019). Retrieved from <https://www.electronicwings.com/sensors-modules/esp8266-wifi-module>
- [4] Google Sheets: Free Online Spreadsheets for Personal Use. (2019). Retrieved from <https://www.google.com/sheets/about/>
- [5] Saputro, M., Widasari, E., & Fitriyah, H. (2017). Implementasi Sistem Monitoring Detak Jantung dan Suhu Tubuh Manusia Secara Wireless. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(2), 148-156. Diambil dari <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/53>
- [6] Firdaus, M. R. (2017). Monitoring Detak Jantung Suhu Dan Infus Pada Pasien Berbasis Mikrokontroler. Retrieved from <http://alumni.unikom.ac.id>