

Sistem *Data Logging* Sederhana Berbasis *Internet Of Things* untuk Pemantauan Suhu Tubuh dan Detak Jantung

Brian Sahuleka, Resmana Lim, Petrus Santoso
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia
E-Mail: m23411022@john.petra.ac.id

Abstrak—Teknologi *Internet Of Things* sudah banyak menarik perhatian dunia saat ini. Pengembangan ini menghasilkan banyak pendekatan yang mengusulkan suatu pemanfaatan *Internet Of Things* dalam aplikasi *data logging* secara *wireless*. Berdasarkan perluasan IOT dan khususnya *wearable device*, remote *data logging* walaupun tanpa server data IOT, penggunaannya telah menjadi suatu tren yang besar dalam dunia kesehatan.

Namun cakupannya masih sangat luas untuk diselami dan yang berkembang saat ini adalah IOT pada *wearable device*. Tugas akhir ini menghadirkan pendekatan dalam membangun sistem *data logging* nirkabel menggunakan teknologi IOT pada server Thingspeak, dan berfokus pada data dari suhu tubuh dan detak jantung. Ada dua Mikrokontroler yang akan digunakan dalam proyek ini yang mana akan digunakan untuk berkomunikasi dengan server IOT.

wireless data logging ini dapat digunakan untuk pengiriman dari sebuah node ke server, tetapi keakuratan waktu penerimaan pada server sangat bergantung pada kondisi kestabilan jaringan.

Kata Kunci— *Internet of Things*, Analog, Mikrokontroler, *Data logging*, WiFi, Server

I. PENDAHULUAN

Teknologi *Internet Of Things* (IoT) berkembang begitu cepat, berbagai peralatan elektronik, sistem kontrol, dan aplikasi mobile mulai terhubung di dalam suatu jaringan komputer yang kompleks [1]. Persaingan dalam teknologi komunikasi pun berkembang dengan cepat, perangkat-perangkat yang digunakan mulai saling berkomunikasi, melakukan *monitoring*, dan bahkan melakukan pertukaran informasi di dalam satu jaringan global. Teknologi ini dikenal dengan teknologi *Internet Of Things* [2].

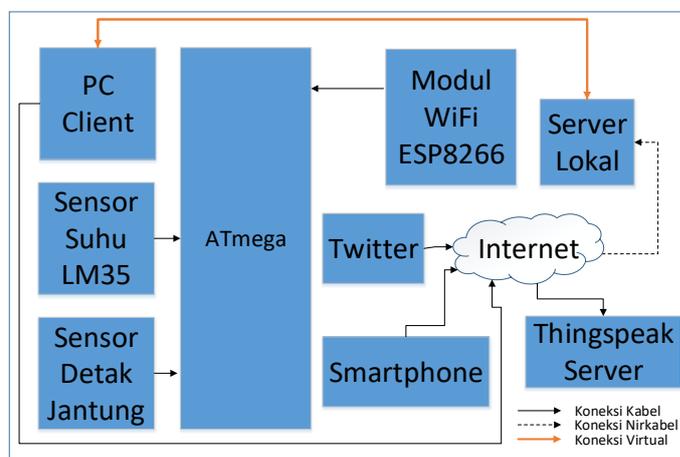
Dalam beberapa tahun terakhir *smartphone* menjadi terkenal dengan fitur-fitur teknologi canggih layaknya komputer yang bisa dibawa kemana-mana. Setelah itu muncul jam tangan dengan fitur yang menyerupai *smartphone* dengan inovasi yang lebih mutakhir daripada jam tangan biasa. Jam tangan ini memiliki fitur seperti *alarm*, telepon sms, pedometer, informasi cuaca, pengendali kamera jarak jauh, kalender, stopwatch, dan bahkan sebagai *heartrate* monitor. Teknologi *internet of things* yang mulai bergerak ke *wearable device* membuat penggunaan IoT menjadi semakin populer pada masa kini [3]. Android dengan Android Wear Smartwatch dan Apple dengan Apple Watch, mulai bersaing dalam mempercanggih teknologi komunikasi pada *wearable device* yang mereka kembangkan. Akan tetapi teknologi komunikasi yang canggih pun membutuhkan sumber daya yang tidak sedikit. Inovasi pada produk membuat dana penelitian yang besar pun tidak menjadi masalah dihabiskan untuk mendapatkan sebuah inovasi [4]. Mahalnya sebuah harga

inovasi dan kualitas perangkat keras yang dibuat membuat produk teknologi *internet of things* menjadi sulit dijangkau dan dibuat dengan tangan sendiri menyerupai kualitas aslinya. Terintegrasinya perangkat keras dan perangkat lunak dalam suatu perangkat membuat implementasi dari *Internet Of Things* juga membutuhkan sumberdaya yang andal dalam mengelolanya [5]. Pada tugas akhir ini akan dibuat sebuah perangkat *wearable data logger* sederhana yang mudah dikelola dengan memanfaatkan teknologi *internet of things* sebagai sarana untuk memantau suhu tubuh dan detak jantung manusia dengan biaya yang terjangkau.

II. PERANCANGAN SISTEM

A. Gambaran Umum Sistem

Desain dari sistem ini terdiri dari perangkat mikrokontroler yang terhubung secara serial dengan modul *wifi*. Perangkat mikrokontroler ini kemudian dapat terhubung ke internet dengan akses poin yang sudah diatur untuk terkoneksi. Data analog sensor yang dibaca oleh mikrokontroler ini kemudian dikirim ke *wifi* melalui port serial yang nantinya dikirim langsung ke server Thingspeak.com untuk ditampilkan dalam bentuk grafik. Desain sistem secara keseluruhan dapat dilihat pada Gambar 1. Data grafik pada halaman Thingspeak.com kemudian di-embedd pada halaman server web lokal untuk ditampilkan dalam bentuk grafik yang sama. Ketika data suhu dan detak jantung pada server Thingspeak mencapai *threshold* yang sudah diatur atau melebihi batas normal, maka server akan mengirimkan notifikasi pada akun twitter yang telah terdaftar untuk memberitahu bahwa data yang telah dikirim ada yang tidak normal.



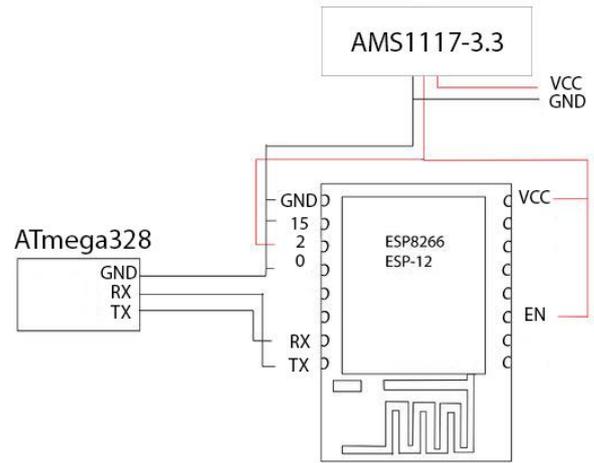
Gambar 1. Desain Sistem.

Layanan dari Sistem ini berupa layanan Autentikasi ke Access Point, layanan HTTP POST pada ESP8266 dengan Thingspeak dan Layanan Notifikasi dari Thingspeak.

B. Desain Desain Hardware

Desain *hardware* yang digunakan pada sistem ini terdiri dari Laptop sebagai Klien PC (*Personal Computer*) dan juga sebagai server untuk menampilkan halaman web yang data grafiknya diambil dari Thingspeak.

Rangkaian mikro pada sistem ini menggunakan ATmega328P sebagai prosesor dengan eksternal kristal sebesar 16 MHz. Sensor yang ada yaitu sensor suhu dan detak jantung langsung terhubung pada pin input analog, dan data hasil bacaan langsung diproses oleh mikrokontroler. Mikrokontroler ATmega328P juga terhubung secara serial dengan modul WiFi ESP8266, dan data yang diperoleh dari sensor kemudian dikirim ke wifi untuk kemudian dikirim ke server.



Gambar 2. Skematik ESP8266 dengan ATmega328P

Tabel 1. Daftar Penggunaan Pin Arduino Mega 2560

Pin	Fungsi
Vin	Sumber tegangan Arduino
GND	Ground
2	TX ESP8266
3	RX ESP8266
1	Reset
5	LED pulse detection
6	PWM LED pulse detection
8	DC pada LCD
9	Reset pada LCD
10	CS pada LCD
11	MOSI pada LCD
12	MISO pada LCD
13	SCLK pada LCD
A0	Output Sensor Detak Jantung
A3	Output Sensor Suhu LM35

Sumber : Arduino [6]

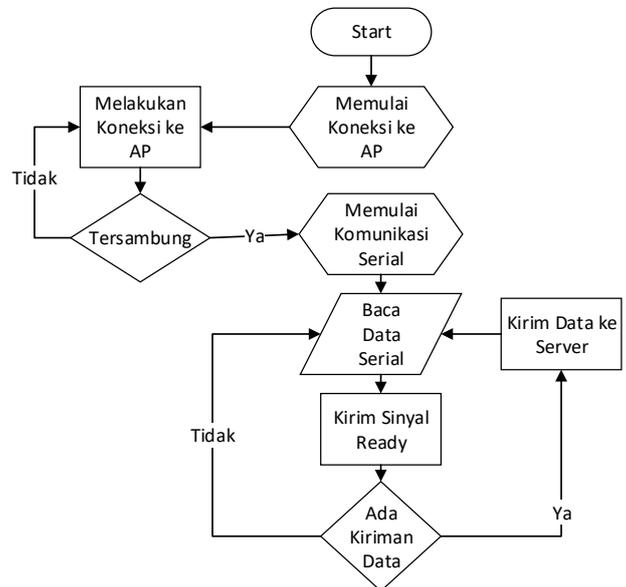
Sistem ini bekerja dengan menghubungkan TFT LCD 2.2” sebagai tampilan sederhana untuk menampilkan data hasil bacaan dari sensor. Karena tegangan LCD adalah 3.3V maka diperlukan modul regulator tambahan untuk mengubah tegangan kerja sistem yaitu 5V menjadi 3.3V sehingga LCD dapat bekerja dengan baik [7]. Output dari mikrokontroler selain daya yang terhubung dengan LCD pun harus diubah menjadi 3.3V dengan menggunakan IC *level shifter* sehingga level tegangan menjadi sesuai.

Pada sistem ini ESP8266 merupakan modul WiFi yang berkomunikasi dengan ATmega328P untuk menerima data hasil bacaan dari kedua sensor. Data yang diterima itu kemudian dikirim ke server untuk ditampilkan. Pada Gambar 2 komunikasi serial antara ESP8266 dengan ATmega328P ini sama seperti rangkaian komunikasi serial biasa, pin RX pada ATmega328P terhubung dengan pin TX pada ESP8266, begitu pula dengan pin TX pada ATmega328P terhubung dengan pin RX pada ESP8266, serta ground antara keduanya saling terhubung. Berikut gambar rangkaian antara ATmega328P dengan ESP8266 ditunjukkan pada gambar berikut.

C. Desain Software ESP8266

Desain *software* pada tugas akhir ini dibagi menjadi tiga komponen utama yaitu, desain *software* pada ESP8266 WiFi, *software* pada ATmega328P untuk membaca data analog sensor, menampilkan, dan mengirimkannya, serta *HTML code* pada server web untuk menampilkan grafik hasil pembacaan dan notifikasi.

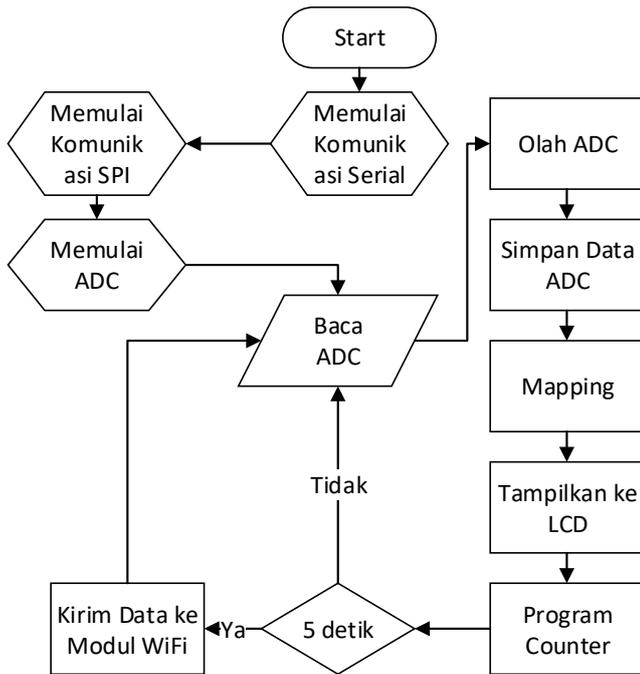
Desain *software* pada aplikasi ini bertujuan untuk mengirimkan data yang diterima dari ATmega328 ke server yang telah ditentukan. Servernya ialah server Thingspeak dan server lokal pada laptop. Flowchart dari *software* ini dapat dilihat pada Gambar 3.



Gambar 3. Flowchart Program ESP8266 WiFi

D. Desain Software pada ATmega328P

Gambar 5. Program Pembacaan Nilai Sensor LM35



Gambar 4. Flowchart sistem kerja ATmega328P

Pemrograman pada ATmega328P terbilang cukup kompleks karena tugasnya yang pertama ialah menjalankan proses pembacaan data analog dari pin analog, menyimpan data tersebut dan kemudian menampilkannya pada layar LCD. Setelah data yang dibaca selesai disimpan data itu yang nantinya akan dikirim melalui serial ke ESP8266 WiFi. Secara garis besarnya *flowchart* dari ATmega328 dapat dilihat pada Gambar 4 di atas.

Pada flowchart dapat dilihat pertama yang dilakukan mikrokontroler adalah memulai komunikasi serial, setelah memulai komunikasi serial dilanjutkan dengan memulai komunikasi SPI untuk LCD, lalu dilanjutkan lagi dengan melakukan pembacaan data analog. Ada dua data yang dibaca pada sistem ini, yang pertama ialah membaca data suhu pada pin A3 dan membaca data detak jantung pada pin A0, data pada pin A3 dihitung menggunakan tegangan referensi internal 1,1V dan kemudian disimpan ke dalam variabel.

Berbeda dengan pin A3 data pada pin 0 dibaca dengan menggunakan *interrupt* dan TIMER sebagai vektor *interrupt* dan pembacaannya setiap 2ms. Setelah kedua data dibaca, data-data tersebut kemudian disimpan ke dalam suatu variabel, untuk ditampilkan ke LCD maupun dikirim ke ESP8266 WiFi. Pada mikrokontroler selanjutnya akan menghitung sampai 5 detik untuk melakukan pengiriman data pada WiFi. Ketika perhitungan belum mencapai 5 detik program akan terus melakukan pembacaan pada data sensor.

E. Pembacaan Sensor Suhu

```
void setup() {
    analogReference (INTERNAL);
}
void loop() {
    reading = analogRead(A3);
    temp = reading/9.31;
}
```

Pada tahapan ini, ATmega328P melakukan pembacaan data yang dikirimkan oleh sensor-sensor yang terhubung pada pin analog. Pembacaan sensor suhu LM35 dapat dilihat dari kode program pada Gambar 5. Pada Gambar 5 dilakukan penentuan tegangan referensi internal dari ATmega328P. Tegangan internal ATmega328P adalah sebesar 1,1V dan skala pembacaan analog adalah sebesar 10 bit atau berkisar dari 0 sampai 1023.

F. Pembacaan Sensor Detak Jantung

Pada tahapan ini sensor melakukan pembacaan pada detak jantung manusia. Detak jantung dibaca setiap 2 ms menggunakan program *interrupt*. *interrupt* diatur sehingga ketika timer menghitung sampai 2 ms, maka selanjutnya ialah menjalankan program perhitungan detak jantung. Perhitungan dilakukan dengan membaca tinggi pulsa yang dihasilkan oleh data sensor, ketika pulsa mencapai batas ketinggian tertentu, maka program akan melakukan perhitungan jumlah pulsa per menit berdasarkan interval waktu yang ada. Data hasil perhitungan waktu ini kemudian yang akan disimpan untuk ditampilkan maupun dikirim pada server. Berikut kode program untuk melakukan pembacaan data dan menjalankan fungsi perhitungan detak jantung.

```
void setup() {
    TCCR2A = 0x02;
    TCCR2B = 0x06;
    OCR2A = 0x7C;
    TIMSK2 = 0x02;
    sei();
}
ISR(TIMER2_COMPA_vect) {
    cli();
    beatCalculation();
    sei();
}
```

Gambar 6. Program Detak Jantung dengan Menggunakan Interrupt

Pada Gambar 6 ada tiga fungsi yang menjalankan fungsi *interrupt*, *cli()*, *beatCalculation()*, dan *sei()*. Fungsi yang menjalankan program perhitungan detak jantung berada pada fungsi *beatCalculation()*, fungsi mendeteksi detak jantung dari sensor yang berskala digital dari 0 sampai 1023. Nilai-nilai ini kemudian dihitung dengan batas tertentu untuk mendeteksi adanya sinyal yang masuk dari sensor, yang kemudian dihitung kisaran detak per menit.

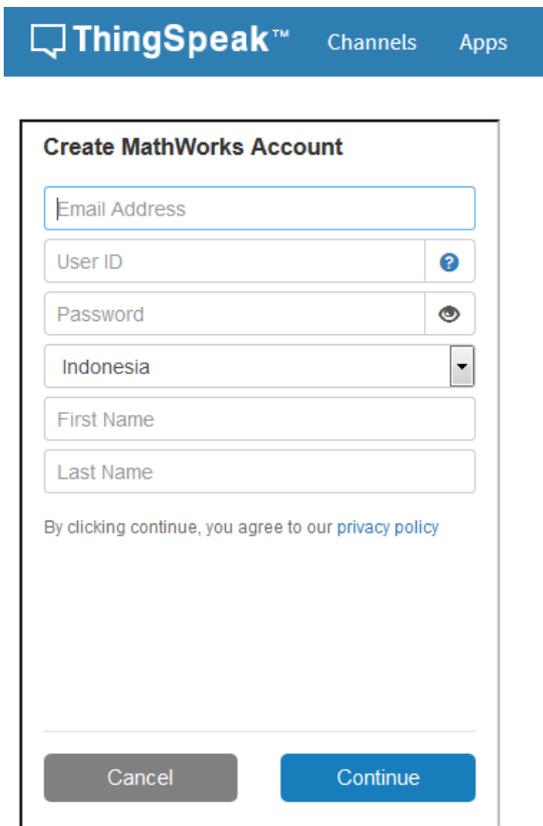
G. Algoritma Tampilan pada LCD.

Pada LCD variabel yang menyimpan data Sensor setelah diolah data tersebut kemudian ditampilkan ke koordinat layar. Data yang diolah ditampilkan per digit berdasarkan fungsi perulangan. Berikut penggalan kode program untuk tampilan digit data. Algoritma yang dijalankan kode program tampilan ini adalah sebagai berikut. Pada tahap awal program melakukan pendeteksian *digit* pada nilai yang tersimpan, nilai yang disimpan tersimpan dalam bentuk *array*. Proses penyimpanan nilai dilakukan dua tahap, penyimpanan pertama melakukan penyimpanan nilai sebelumnya, sedangkan penyimpanan kedua ialah penyimpanan nilai setelahnya. Saat

ditampilkan program akan membandingkan *digit* data sebelum dan sesudah, jika data tersebut berbeda maka program akan menampilkan yang baru, sedangkan jika *digit* data sama dengan sebelumnya maka program tidak melakukan apa-apa terhadap *digit* yang sudah tercetak.

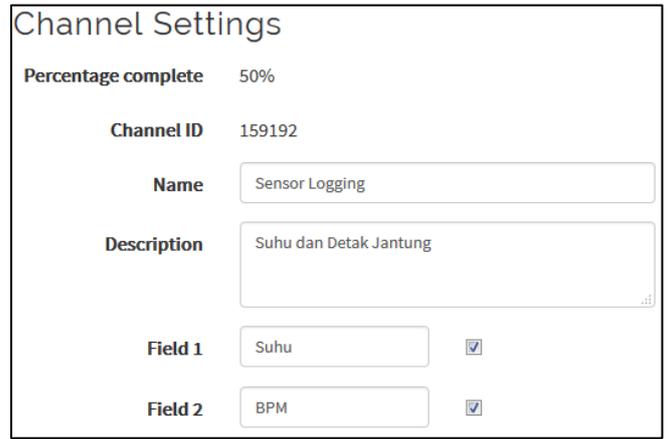
H. Pengaturan Server Thingspeak

Thingspeak merupakan server Internet of Things yang cukup diminati karena konfigurasinya yang mudah, untuk memulai Thingspeak pertama-tama kita harus membuat akun pada server tersebut dengan alamat email yang kita miliki. Setelah mendapat akun pada server tersebut, maka langkah selanjutnya ialah mengatur dashboard atau tampilan awalnya dengan perangkat kita yang akan digunakan untuk melakukan pengiriman data. Gambar berikut menunjukkan halaman ketika kita memulai menggunakan Thingspeak, dan akan diarahkan untuk membuat akun jika belum terdaftar sebagai pengguna Thingspeak.



Gambar 7. Pembuatan Akun pada Thingspeak

Pada Gambar 7 setelah akun selesai dibuat, pada tampilan awal Thingspeak akan diarahkan untuk membuat *channel* untuk perangkat yang akan kita hubungkan dengan server. Gambar 8 menunjukkan pembuatan *channel* yang menggunakan *field* sebagai parameter pengiriman data yang dikirimkan melalui Modul WiFi ESP8266 menggunakan HTTP POST. Kode pengiriman WiFi dapat dilihat pada penggalan kode dari Gambar 9.



Gambar 8. Pembuatan Channel pada Thingspeak

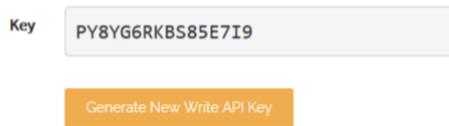
```
void server1() {
    String apiKey =
    "PY8YG6RKBS85E7I9";
    kiriman = "&field1="; kiriman
    += suhu;
    kiriman += "&field2="; kiriman
    += "detak";

    const char* server =
    "thingspeak.com";
    if (client.connect(server, 80))
    {
        String postStr =apiKey;
        postStr += kiriman;
        postStr +=
        "\r\n\r\n";
        client.print("POST /update
        HTTP/1.1\n");
        client.print("Host:
        thingspeak.com\n");
        client.print("Connection:
        close\n");
        client.print("X-
        THINGSPEAKAPIKEY: "+apiKey+"\n");
        client.print("Content-Type:
        application/x-www-form-
```

Gambar 9. Kode Program Pengiriman Data Ke Field Thingspeak

Pada Gambar 10 pembuatan channel dilakukan dengan melakukan pengaturan untuk membuat apakah channel yang akan digunakan bersifat public atau private. Tujuan pembuatan ini ialah membuat hak pengaksesan terhadap data yang diolah. Pada tugas akhir ini channel dibuat public supaya dapat diakses oleh server lain untuk melakukan embed terhadap grafik dari Thingspeak. Setelah channel berhasil dibuat maka API (Application Programming Interface) Key akan otomatis ter-generate untuk dilakukan sebagai penghubung antara Thingspeak dan modul WiFi. Jika ada permasalahan mengenai koneksi antara perangkat dengan API Key kita bisa melakukan generate ulang API Key. Tampilan halaman API Key dapat dilihat pada Gambar berikut.

Write API Key

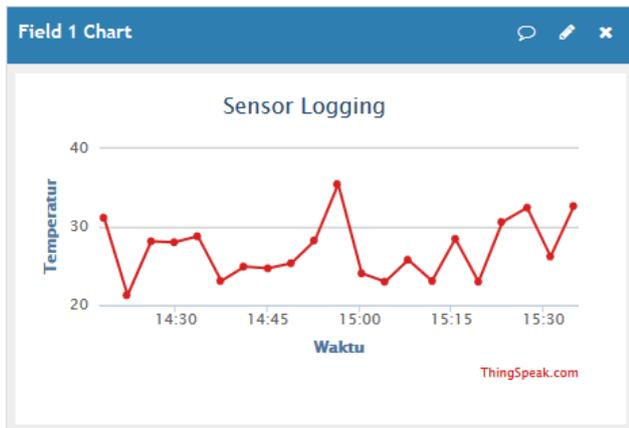


Read API Keys



Gambar 10. API Key pada Thingspeak

Pada API Key terdapat write API Key dan Read API Key, yang digunakan dalam sistem ini ialah write API Key karena kita akan menuliskan data pada server. Setelah API Key sudah tergenerate maka selanjutnya kita dapat menempatkannya pada kode program di atas. Tampilan awal grafik pada My Channel sebelum data dikirim akan terlihat kosong, setelah data dikirimkan dari perangkat, maka grafika akan terlihat seperti Gambar 11.



Gambar 11. Tampilan Grafik pada Thingspeak

Grafik yang dapat ditampilkan dapat diatur sedemikian hingga terlihat seperti yang kita inginkan, tampilan grafik yang penulis gunakan adalah tampilan grafik dinamis dengan 10 data per tampilan dengan skala otomatis.

III. PENGUJIAN SISTEM

A. Pengujian Konektivitas ke Akses Poin

Pengujian ini dilakukan untuk mengetahui apakah konektivitas ke jaringan internet dapat terjalin dan dapat melakukan koneksi ke server.

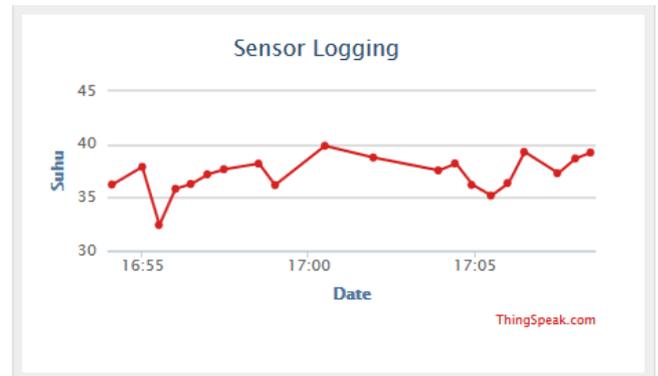


Gambar 12. Pengujian Konektivitas WiFi terhadap Akses Poin

Pada Gambar 12 Pengujian dilakukan pada Modul WiFi yang terhubung dengan serial monitor untuk melakukan pengiriman data. Dari pengujian ini diperoleh hasil yang baik dengan tidak ada kesalahan pada saat melakukan koneksi.

B. Pengujian Pembacaan pada Sensor Suhu

Pengujian ini dilakukan untuk mengetahui ketepatan dalam melakukan pengukuran oleh perangkat. Sensor yang akan diuji dipasang pada pin yang sudah ditentukan, setelah semua terhubung langkah selanjutnya ialah mencatat hasil yang diperoleh dari data pengukuran. Data pada pengukuran dibawah ini otomatis diambil dalam rentan waktu pengiriman selama 20 detik dan waktu penerimaan pada server Thingspeak berkisar 20 detik [8], total waktu pengukuran adalah sebesar 15 menit.



Gambar 13. Pengujian Konektivitas WiFi terhadap Akses Poin

Jika kondisi jaringan tidak terlalu padat, maka pada kondisi ideal proses pengiriman data akan berlangsung dengan interval sekitar 20 detik. Jadi total data ideal yang akan terkirim adalah berjumlah 15 data. Pada Gambar 13 dapat terlihat ada beberapa data yang terhilang pada saat pengiriman.

Tabel 2. Pengujian Pengukuran Suhu (16.55-17.10)

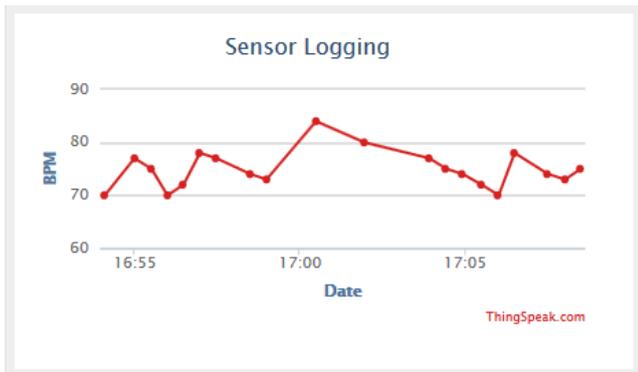
20 detik ke	Suhu Termometer	Suhu 5 menit pertama	Suhu 5 menit Kedua	Suhu 5 menit Ketiga
1	36,19	37,89	39,87	35,19
2	36,19	-	-	-
3	36,19	32,43	-	-
4	36,19	35,83	-	36,37
5	36,19	36,29	38,77	-
6	36,19	37,19	-	39,3
7	36,19	-	-	-
8	36,19	-	37,56	-
9	36,19	37,67	-	37,3
10	36,19	-	-	38,68
11	36,19	-	38,21	39,24
12	36,19	38,2	36,2	-
13	36,19	36,18	-	-
14	36,19	-	-	-
15	36,19	-	-	-

Berdasarkan data pada Tabel 2 di atas diperoleh presentase keberhasilan pengiriman data suhu tubuh dari hasil 5 menit

pertama yaitu 8 dari 15 data adalah sebesar 53,33%, pengukuran 5 menit kedua menghasilkan keberhasilan sebesar 33,33% karena 5 dari 15 data yang berhasil terkirim, dan pengukuran 5 menit ketiga adalah sebesar 40% karena hanya ada 6 data yang terkirim.

C. Pengujian Pembacaan Sensor Detak Jantung

Pengujian ini memiliki tujuan untuk mengetahui ketepatan dalam membaca detak jantung pada tubuh manusia. Detak jantung pada awalnya diukur menggunakan cara manual yaitu dengan menghitung jumlah detak yang ditimbulkan selama satu menit dan diperoleh hasil 80 denyut per menit. Jumlah denyut ini selanjutnya yang akan dijadikan sebagai standar untuk menghitung error dari sensor. Berikut gambar pengujian selama 20 menit pada server Thingspeak.



Gambar 14. Grafik Detak Jantung pada Server Thingspeak.

Sama seperti grafik pengukuran suhu tubuh, Grafik pada Gambar 14 juga memiliki kesamaan dalam penerimaan data pada server dimana ada data yang hilang pada saat proses pengiriman.

Tabel 3. Pengujian Detak Jantung (16.55-17.10)

20 detik ke	Suhu Termometer	Detak 5 menit pertama	Detak 5 menit Kedua	Detak 5 menit Ketiga
1	80	77	84	72
2	80	-	-	-
3	80	75	-	-
4	80	70	-	70
5	80	72	80	-
6	80	78	-	78
7	80	-	-	-
8	80	-	77	-
9	80	77	-	74
10	80	-	-	73
11	80	-	75	75
12	80	74	74	-
13	80	73	-	-
14	80	-	-	-
15	80	-	-	-

Pada Tabel 3 di atas, dapat terlihat ada 9 data yang berhasil terkirim dari 15 data yang dikirim dalam interval 20 detik, dan terdapat 6 data yang hilang ketika dikirim. Tabel 8 di atas, dapat terlihat ada 5 data yang berhasil terkirim dari 15 data yang dikirim dalam interval 20 detik, dan terdapat 10 data yang

hilang ketika dikirim. Sedangkan pada Tabel 4.6 di atas, dapat terlihat ada 6 data yang berhasil terkirim dari 15 data yang dikirim dalam interval 20 detik, dan terdapat 9 data yang hilang ketika dikirim.

Berdasarkan data pada Tabel di atas diperoleh presentase keberhasilan pengiriman data detak jantung pada 5 menit pertama yaitu 9 dari 15 data adalah sebesar 60%, pengukuran 5 menit kedua menghasilkan keberhasilan sebesar 33,33% karena 5 dari 15 data yang berhasil terkirim, dan pengukuran 5 menit ketiga adalah sebesar 40% karena hanya ada 6 data yang terkirim.

D. Pengujian Tampilan LCD Untuk Suhu Dan Detak Jantung

Pengujian ini bertujuan untuk melihat apakah koordinat tampilan data yang ditetapkan sesuai yang diharapkan, dari pengujian tampilan ada beberapa masalah yang terjadi seperti terjadi drop tegangan pada module AMS1117-3.3, sehingga solusi untuk memperbaikinya penulis menambahkan modul tambahan untuk memperbaiki keadaan tersebut. Tampilan akhirnya dapat berjalan dengan normal seperti pada Gambar 15.



Gambar 15. Tampilan Data pada LCD

E. Pengujian pada server

Pengujian ini dilakukan untuk menguji apakah server yang menangani data maupun notifikasi berjalan dengan baik, pada Gambar di bawah ini dapat dilihat tampilan yang menunjukkan tampilan server lokal dengan embedded grafik yang berasal dari Thingspeak.

Hasil pengujian menunjukkan bahwa server berjalan dengan baik dan dapat mengambil konten yang bersangkutan dari server Thingspeak, selain pengujian data grafik, pengujian tampilan jarum penunjuk juga berjalan dengan baik.

IV. KESIMPULAN

Kesimpulan yang didapat dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Koneksi WiFi terhadap akses poin berjalan dengan baik dengan tidak ada kesalahan, tetapi pada kondisi jaringan yang tidak stabil terkadang koneksi lambat, namun perangkat tetap melakukan proses pengiriman.

2. Pengiriman data ke server berjalan dengan memiliki kendala pada saat pembaruan data pada grafik dengan tingkat keberhasilan sebesar 53,3% seperti pada pengujian yang dikarenakan karena kondisi jaringan yang pada saat itu sedang terdapat kepadatan trafik.
3. Area Pembacaan data denyut tidak memiliki sensitivitas yang baik karena diperlukan pengaturan sensor pada posisi denyut sehingga denyut dapat terdeteksi dengan baik.

DAFTAR PUSTAKA

- [1] A. Junaidi, "INTERNET OF THINGS, SEJARAH, TEKNOLOGI DAN PENERAPANNYA: REVIEW," *J. Ilm. Teknol. Inf. Terap.*, 2015.
- [2] R. Weber and R. Weber, *Internet of things: legal perspectives*. 2010.
- [3] E. Mouser, "The Internet of Things," *Wearable Devices and the Internet of Things*, 2016.
[Online]. Available:
<http://www.mouser.co.id/applications/article-iot-wearable-devices/>.
- [4] G. Moore and I. Benbasat, "Development of an instrument to measure the perceptions of adopting an information technology innovation," *Inf. Syst. Res.*, 1991.
- [5] B. Santoso, M. Rahmah, T. Setiasari, and P. Sularsih, "PERKEMBANGAN DAN MASA DEPAN TELEMEDIKA DI INDONESIA," *researchgate.net*, 2014.
- [6] Arduino, "ATmega168/328-Arduino Pin Mapping," *ATmega168/328-Arduino Pin Mapping*, 2016. [Online]. Available:
<https://www.arduino.cc/en/Hacking/PinMapping168>.
- [7] ElecFreaks, "2.2S" TFT LCD: TFT01-2.2S," *2.2S" TFT LCD: TFT01-2.2S*, 2016. [Online]. Available:
http://www.elecFreaks.com/wiki/index.php?title=2.2S%2522_TFT_LCD:_TFT01-2.2S.
- [8] Thingspeak, "Thingspeak Community," *Send Data to ThingSpeak with Arduino*, 2016.
[Online]. Available:
<http://community.thingspeak.com/tutorials/arduino/send-data-to-thingspeak>.