

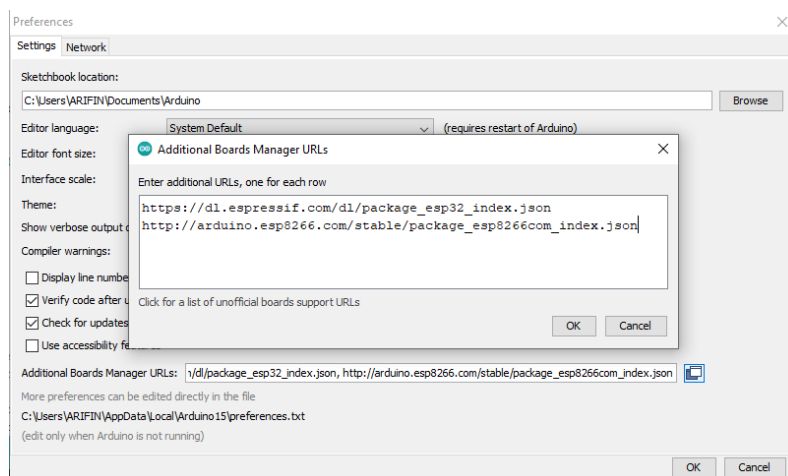
BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Pengujian sistem dilakukan untuk mengetahui apakah perangkat keras serta perangkat lunak yang akan dan telah dibuat dapat bekerja dengan baik. Pengujian sistem yang dilakukan yaitu meliputi pengujian perangkat keras atau alat, perangkat lunak sistem dan pengujian keseluruhan sistem yang telah dibuat.

4.1 Pembuatan Program ESP32

Pada bagian pembuatan program esp32 ini bertujuan agar keseluruhan dari sistem yang akan dibuat akan kompatibel dengan tools arduino. Hal tersebut diperlukan dikarenakan terdapat perbedaan perangkat yang digunakan, dimana tools arduino IDE sebenarnya hanya diperuntukan untuk penggunaan perangkat dengan turunan arduino. Maka dari itu perlu dilakukan konfigurasi agar tools arduino IDE dapat mengenali perangkat yang digunakan pada penelitian kali ini, yaitu esp32. Berikut adalah beberapa tahapannya:



Gambar 4.1 Penggunaan Library ESP32

Pada gambar diatas merupakan bagian yang digunakan untuk mengambil library esp32 yang diambil dari version control yaitu github. Dengan memasukkan library tersebut esp32 dapat dikenali oleh tools arduino IDE sebagai perangkat yang mendukung untuk dilakukan pemrograman. Berikut adalah hasil pembuatan program pada esp32:

```
void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}
```

Done compiling.

Sketch uses 890974 bytes (67%) of program storage space. Max allowed is 1310720 bytes.

Global variables use 40000 bytes (12%) of dynamic memory, leaving 280000 bytes free. Max allowed is 300000 bytes.

1 ESP32 Dev Module, Disabled, Default 4MB with spiffs

Gambar 4.2 Pembuatan Program ESP32

Pada gambar diatas merupakan indikator bahwa tools arduino IDE telah berhasil melakukan kompilasi *source code* yang digunakan untuk perangkat esp32.

4.2 Program Pengiriman Data MQTT

Pada bagian ini merupakan bagian pembuatan program yang digunakan untuk melakukan pengiriman data dari perangkat keras esp32 menuju server menggunakan protokol MQTT. Berikut adalah program yang dibuat:

```
if (KirimDataSuhu > Batas ){
  myBot.sendMessage(id, "Suhune Panas BOS");
  myBot.sendMessage(id, (KirimDataSuhu));
  Serial.println("Pesan Terkirim");
  Serial.println("SUHU PANAS");
  client.publish("iotdev/SuhuKubikel", String(KirimDataSuhu));
  client.publish("iotdev/KeteranganSuhuKubikel", "Panas");
}else {
  Serial.println("SUHU AMAN");
  client.publish("iotdev/SuhuKubikel", String(KirimDataSuhu));
  client.publish("iotdev/KeteranganSuhuKubikel", "AMAN");
}
```

Gambar 4.3 Program Pengiriman Data MQTT

Pada gambar diatas merupakan program yang digunakan untuk mengirimkan data menuju server dengan menggunakan protokol MQTT. Dimana fungsi yang digunakan untuk melakukan tugas tersebut adalah *client publish*, dimana fungsi tersebut memastikan pengiriman data berdasarkan topik yang telah ditentukan yang diikuti oleh variabel yang berisikan data pemantauan suhu.

4.3 Pembuatan Program Koneksi Dengan Server

Pada bagian ini merupakan pembuatan program yang digunakan untuk menghubungkan perangkat keras dengan server. Dimana server disini bersifat publik ip, sehingga dapat diakses melalui jaringan luar. Berikut merupakan program yang digunakan untuk menghubungkan dengan server:

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include "CTBot.h"

const char* ssid = "wifi"; //sesuaik
const char* password = "password"; /.
const char* mqtt_server = "202.154.56
const char* mqtt_user = "skripsi"; //.
const char* mqtt_password = "2022"; /.
```

Gambar 4.4 Program Menghubungkan Dengan Server

Pada gambar diatas merupakan program yang dibuat untuk menghubungkan perangkat keras dengan server. Pada bagian ini terdapat inisialisasi alamat server yang digunakan, dimana hal tersebut bertujuan untuk menentukan *end point* dari pengiriman data yang dilakukan.

4.4 Konfigurasi Virtual Private Server

Pada bagian ini merupakan tahapan untuk melakukan konfigurasi pada

sever. Hal tersebut bertujuan untuk mengakomodir dari ekosistem internet of things yang diterapkan, mulai dari database, mqtt hingga website. Berikut merupakan tampilan service dari berbagai sistem yang berjalan pada virtual private server:

```

iotskrip@iot-skripsi: ~
CPU[          ] 0.7%] Tasks: 36, 64 thr; 1 running
Mem[|||||||] 180M/985M] Load average: 0.00 0.04 0.02
Swp[|||||] 255M/1.92G] Uptime: 60 days, 02:32:18

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
10548 iotskrip   20    0 32120  4564  3788  R   0.7  0.5   0:00.69 http
1 root      20    0  220M  6292  4088  S   0.0  0.6   2:03.07 /lib/systemd/systemd --system --deser
386 root      20    0  103M   704   556  S   0.0  0.1   0:00.09 /sbin/lvmetad -f
955 iotskrip   20    0 76688  3360  3360  S   0.0  0.3   0:00.01 /lib/systemd/systemd --user
956 iotskrip   20    0  253M   288    0  S   0.0  0.0   0:00.00 (sd-pam)
967 daemon     20    0 28340  1744  1536  S   0.0  0.2   0:00.05 /usr/sbin/atd -f
970 messagebu 20    0 50320  3172  2588  S   0.0  0.3   2:43.52 /usr/bin/dbus-daemon --system --adre
993 root      20    0 71172  3916  2916  S   0.0  0.4   0:07.52 /lib/systemd/systemd-logind
1122 root      20    0  165M  3316  3092  S   0.0  0.3   0:00.00 /usr/bin/python3 /usr/bin/networkd-di
996 root      20    0  165M  3316  3092  S   0.0  0.3   0:00.08 /usr/bin/python3 /usr/bin/networkd-di
27977 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.05 /usr/bin/lxcfs /var/lib/lxcfs/
10163 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.02 /usr/bin/lxcfs /var/lib/lxcfs/
10167 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.01 /usr/bin/lxcfs /var/lib/lxcfs/
10168 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.01 /usr/bin/lxcfs /var/lib/lxcfs/
10171 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.01 /usr/bin/lxcfs /var/lib/lxcfs/
10172 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
10173 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
10175 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
10176 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
10177 root      20    0  630M  3504  1412  S   0.0  0.3   0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
1009 root      20    0  630M  3504  1412  S   0.0  0.3   0:24.65 /usr/bin/lxcfs /var/lib/lxcfs/
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

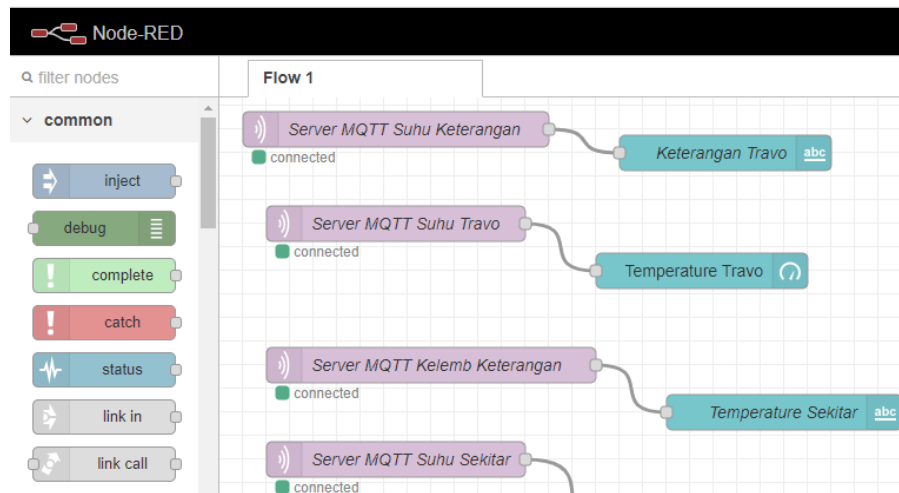
```

Gambar 4.5 Menjalankan Service Server

Pada gambar diatas merupakan tampilan service yang sedang berjalan pada server pemantauan suhu klem kubikel ini, dimana semua service sudah berjalan dengan baik dan juga sudah berjalan selama 60 hari. Hal tersebut ditunjukkan pada uptime yang menunjukkan 60 hari 2 jam 32 menit serta 18 detik.

4.5 Konfigurasi Website

Pada bagian ini merupakan tahapan untuk melakukan konfigurasi pada website. Dimana pada penelitian ini website yang digunakan dibangun dengan framework nodored yang menggunakan konsep *low code*. Berikut merupakan konfigurasi dari website pemantauan suhu:



Gambar 4.6 Konfigurasi Website

Pada gambar diatas merupakan adalah bagian yang digunakan untuk melakukan konfigurasi pada website. Dimana penggunaan framework node-red ini tidak banyak menggunakan *source code* dalam membangun website.

4.6 Konfigurasi Database Mysql

Pada bagian ini merupakan tahapan untuk melakukan konfigurasi pada database mysql. Dimana konfigurasi ini meliputi pembuatan tabel-tabel yang nantinya akan digunakan untuk menyimpan data-data dari pemantauan suhu kubikel incoming. Berikut merupakan konfigurasinya:

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	AJ C
No	INT	36	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
su_hu_travo	VARCHAR	36	None			<input type="checkbox"/>	---	<input type="checkbox"/>
su_hu_sekitar	VARCHAR	36	None			<input type="checkbox"/>	---	<input type="checkbox"/>
timestamp	DATETIME		CURRENT_TIME		on update CURR	<input type="checkbox"/>	---	<input type="checkbox"/>

Gambar 4.7 Konfigurasi Database

Pada gambar diatas adalah konfigurasi dari database yang digunakan untuk

menyimpan data pemantauan. Dimana terdapat 4 kolom dengan tipe data yang berbeda-beda, seperti *integer*, *varchar* dan juga *datetime*. Hal tersebut bertujuan agar database nantinya dapat menampung data dalam jumlah besar serta dapat bekerja secara optimal. Hal tersebut dikarenakan proses penyimpanan data dari teknologi internet of things merupakan salah satu sumber dari teknologi big data.

4.7 Implementasi

Pada implementasi sistem ini bertujuan untuk melakukan semua proses kinerja alat dari awal hingga akhir mulai dari sistem perangkat keras hingga sistem perangkat lunak. Dimana pada tahapan awal ini adalah dengan melakukan implementasi terhadap perangkat keras yang digunakan, dimana terdapat beberapa perangkat seperti papan pcb, sensor mlx90614 dan juga mikrokontroler esp32. Berikut merupakan tampilan dari perangkat keras yang nantinya akan digunakan :



Gambar 4.8 Pemasangan Perangkat Keras

Pada gambar diatas merupakan perangkat yang nantinya akan digunakan untuk melakukan pemantauan suhu terhadap klem sambungan kubikel incoming

pada gardu induk Ngagel. Dimana perangkat tersebut akan dirangkai dalam satu kesatuan untuk dapat saling berkomunikasi dan mengirimkan sebuah data. Perangkat ini nantinya akan mampu membaca suhu dari klem sambungan kubikel incoming dari jarak 50 cm, sehingga tidak mengganggu fungsi dari perangkat kubikel incoming itu sendiri. Berikut merupakan potongan program yang nantinya akan digunakan pada perangkat keras ini:

```

if (KirimDataSuhu > Batas ){
  myBot.sendMessage(id, "Suhune Panas BOS");
  myBot.sendMessage(id, (KirimDataSuhu));
  Serial.println("Pesan Terkirim");
  Serial.println("SUHU PANAS");
  client.publish("iotdev/SuhuKubikel", String(KirimDataSuhu).c_str(),
  client.publish("iotdev/KeteranganSuhuKubikel", String("Suhu Object
}else {
  Serial.println("SUHU AMAN");
  client.publish("iotdev/SuhuKubikel", String(KirimDataSuhu).c_str()
  client.publish("iotdev/KeteranganSuhuKubikel", String("Suhu Object
}

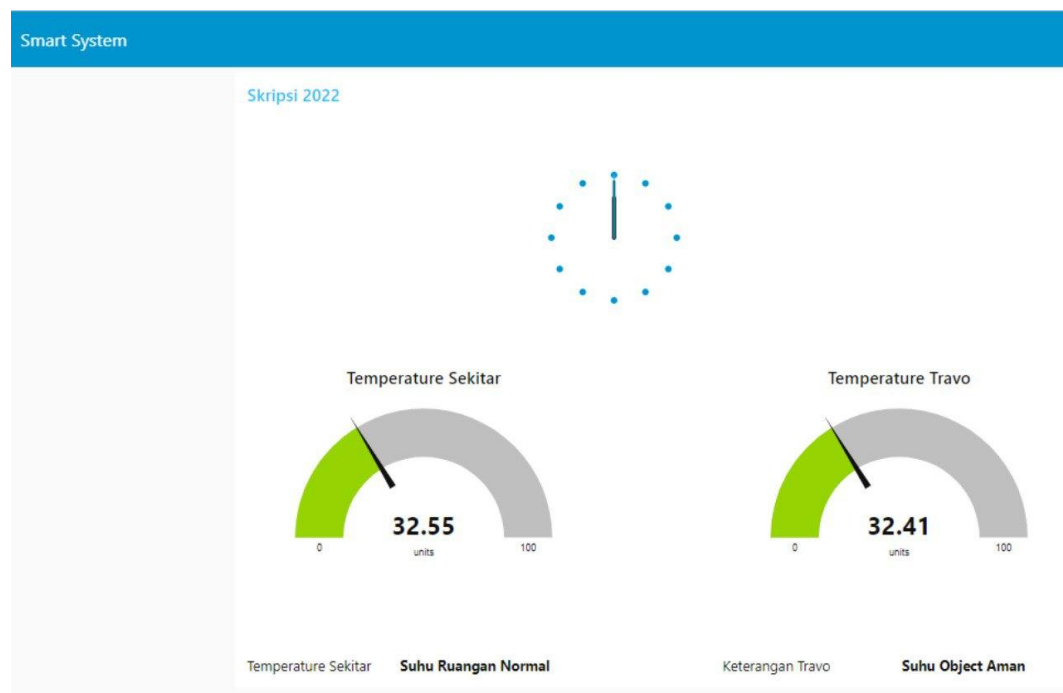
if(SuhuSekitar > Batas2 ){
  Serial.println(SuhuSekitar);
  client.publish("iotdev/SuhuSekitarKubikel", String(SuhuSekitar).c
  client.publish("iotdev/KeteranganSuhuSekitarKubikel", String("Suh
}else{

```

Gambar 4.9 Program Keseluruhan Sistem

Pada gambar diatas merupakan potongan dari sebagian program yang digunakan untuk melakukan pembacaan suhu dari klem sambungan kubikel incoming. Selain itu program tersebut juga terdapat beberapa fungsi seperti mengirim data kepada server dan juga mengirimkan sebuah notifikasi pada aplikasi telegram. Setelah data dikirimkan pada server, selanjutnya adalah data tersebut akan ditampilkan pada sebuah dashboard pemantauan data menggunakan node-red. Pada source code tersebut tertulis perintah client publish yang memiliki

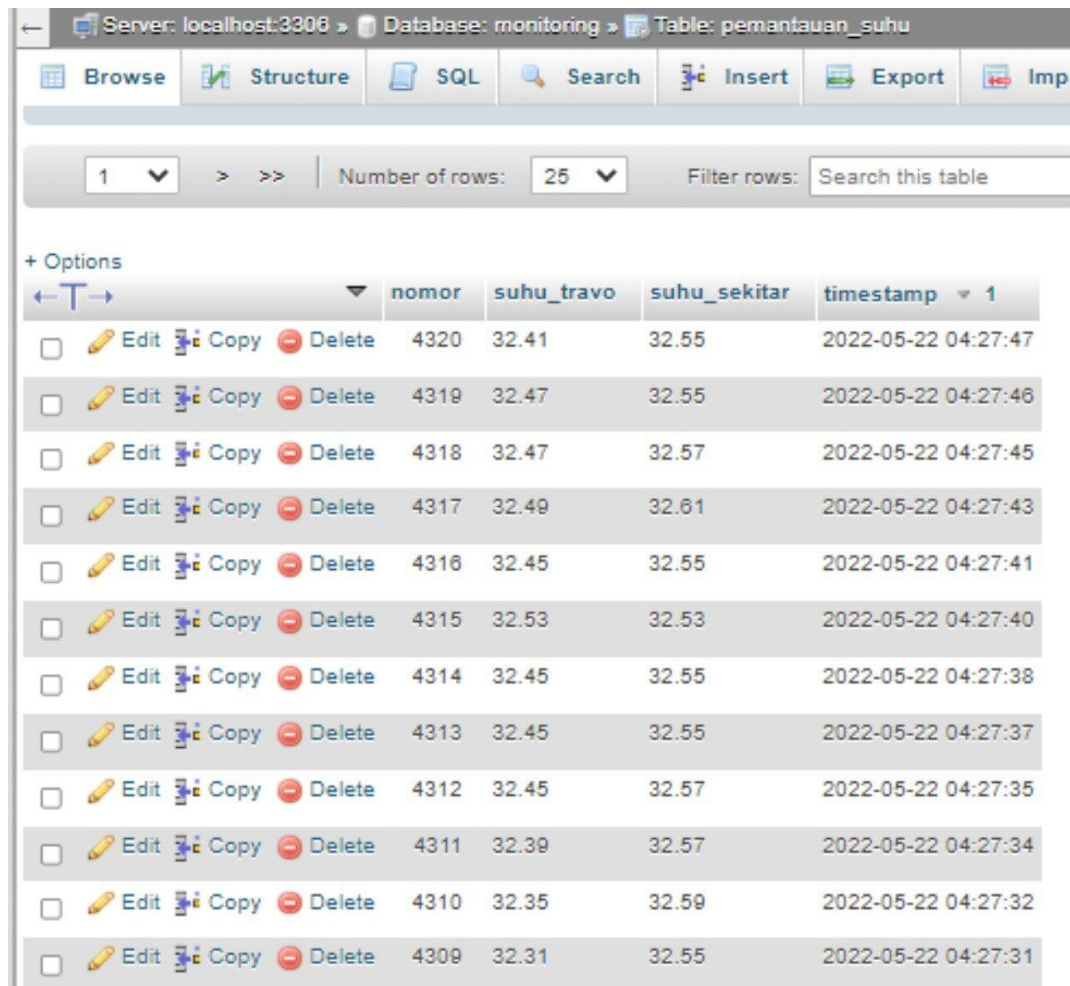
fungsi data yang ada dalam mikrokontroler akan dipublish kepada server dengan topik yang sudah ditentukan sebelumnya. Setelah dari server, data tersebut akan ditampilkan dalam sebuah dashboard yang lebih nyaman untuk dilihat oleh pengguna atau memiliki user interface yang mudah dipahami oleh orang awam.



Gambar 4.10 Tampilan Dashboard Node-red

Pada gambar diatas merupakan hasil akhir dari website node-red dashboard yang nantinya akan digunakan sebagai user interface bagi pengguna dalam melakukan monitoring terhadap kondisi dari suhu klem kubikel incoming. Untuk dapat mengakses web tersebut nantinya pengguna hanya perlu membuka sebuah browser dan mengakses link <https://192.168.100.69:1880/ui> yang secara otomatis akan mengarah pada dashboard pemantauan suhu klem kubikel incoming itu sendiri. Kemudian data yang sudah ditampilkan pada sebuah dashboard akan

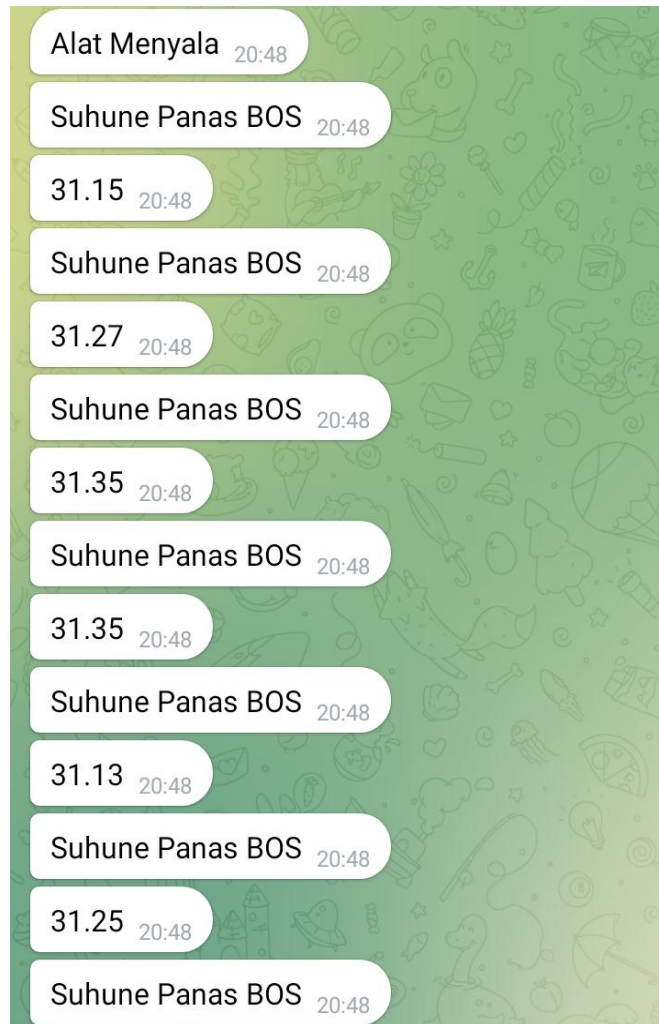
secara otomatis tersimpan ke dalam sebuah database MYSQL. Berikut adalah tampilannya:



	nomor	suhu_travo	suhu_sekitar	timestamp
<input type="checkbox"/>	4320	32.41	32.55	2022-05-22 04:27:47
<input type="checkbox"/>	4319	32.47	32.55	2022-05-22 04:27:46
<input type="checkbox"/>	4318	32.47	32.57	2022-05-22 04:27:45
<input type="checkbox"/>	4317	32.49	32.61	2022-05-22 04:27:43
<input type="checkbox"/>	4316	32.45	32.55	2022-05-22 04:27:41
<input type="checkbox"/>	4315	32.53	32.53	2022-05-22 04:27:40
<input type="checkbox"/>	4314	32.45	32.55	2022-05-22 04:27:38
<input type="checkbox"/>	4313	32.45	32.55	2022-05-22 04:27:37
<input type="checkbox"/>	4312	32.45	32.57	2022-05-22 04:27:35
<input type="checkbox"/>	4311	32.39	32.57	2022-05-22 04:27:34
<input type="checkbox"/>	4310	32.35	32.59	2022-05-22 04:27:32
<input type="checkbox"/>	4309	32.31	32.55	2022-05-22 04:27:31

Gambar 4.11 Tampilan Data MYSQL

Pada gambar diatas merupakan hasil penyimpanan data yang sebelumnya sudah dikirimkan oleh mikrokontroler kepada server. Data yang disimpan adalah data ketinggian suhu dari klem kubikel dan juga waktu dilakukannya pemantauan tersebut berlangsung. Selain itu sistem ini juga akan mengirimkan sebuah pesan singkat melalui aplikasi telegram jika suhu dari kubikel incoming sedang diatas normal, berikut tampilannya:



Gambar 4.12 Notifikasi Aplikasi Telegram

Pada gambar diatas adalah sebuah notifikasi yang dilakukan oleh sistem monitoring kubikel incoming jika kondisi dari suhu sambungan klem memiliki suhu diatas normal. Dengan notifikasi ini nantinya petugas dapat melakukan tindakan jika kondisi dari suhu klem sambungan kubikel incoming terjadi kelebihan suhu.

Berikut adalah tabel data pemantauan dari klem kubikel incoming:

No	Suhu Objek (MLX90614)	Suhu Objek (Thermogun)	Suhu Sekitar (MLX90614)	Suhu Sekitar (Pengukur Suhu Manual)
1.	36.8 C	36.7 C	38.8 C	38.8 C
2.	40.1 C	40.0 C	41.2 C	41.4 C
3.	45.0 C	45.0 C	45.9 C	45.9 C
4.	43.5 C	43.5 C	44.2 C	44.1 C
5.	41.2 C	41.0 C	43.3 C	43.3 C
6.	41.4 C	41.3 C	42.5 C	43.5 C

Tabel 4.1 Hasil Komparasi Pemantauan Suhu

Berdasarkan tabel pemantauan yang ditunjukkan diatas, merupakan komparasi dengan peralatan yang biasa dilakukan pada gardu induk. Dimana terdapat *error gap* sebesar 0,1 sampai 0,2 yang terjadi antara peralatan yang dibuat pada penelitian tugas akhir ini yang menggunakan sensor MLX90614. Nilai tersebut masih mendapatkan toleransi dari pihak gardu induk dengan batas toleransi *error gap* sebesar 2 Derajat *celcius*. Dengan begitu pada penelitian yang dilakukan pada tugas akhir ini dapat bekerja secara optimal.

4.8 Pengujian Perangkat Keras

Pada tahapan pengujian perangkat keras ini akan meliputi beberapa bagian

didalamnya, seperti pengujian pada mikrokontroler dimana pada penelitian tugas akhir ini akan menggunakan jenis ESP32 untuk mengatur dari keseluruhan sistem. Selanjutnya akan dilakukan pula pada sensor yang digunakan yaitu sensor MLX90614, dimana nantinya sensor tersebut akan digunakan sebagai perangkat yang digunakan untuk melakukan pemantauan terhadap suhu dari klem sambungan kubikel incoming. Pengujian perangkat keras disini dilakukan untuk nantinya dapat mengetahui bagaimana kinerja dari setiap komponen perangkat-perangkat keras yang akan diterapkan selama penelitian tugas akhir ini berlangsung.

4.8.1 Pengujian Mikrokontroler ESP32

Pada tahapan ini merupakan bagian yang digunakan untuk mengetahui apakah mikrokontroler ESP32 dapat bekerja dengan baik serta sesuai dengan konsep yang sudah ditentukan di awal penelitian. Kemudian tahapan ini juga akan digunakan untuk memastikan apakah mikrokontroler ESP32 mampu untuk terhubung dengan jaringan WiFi yang memiliki koneksi internet dengan baik. Selain itu pada tahapan ini juga akan memastikan bahwa perangkat mikrokontroler mampu untuk mendukung untuk pembuatan sistem monitoring suhu klem sambungan kubikel incoming yang nantinya akan diintegrasikan dengan sensor yang akan membaca suhu dari klem sambungan kubikel incoming, mengirimkan data kepada server, mengirimkan pemberitahuan kepada aplikasi telegram. Dengan pengujian mikrokontroler ini diharapkan ketika nantinya dilakukan pengujian sistem secara menyeluruh, maka akan meminimalisir untuk terjadinya sebuah kesalahan.

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

pinMode(smokeA0, INPUT);
pinMode(sensor_pin, INPUT_PULLUP);

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
client.setServer(mqtt_server, 1883);
client.setCallback(receivedCallback);

```

Gambar 4.13 Source Code Pengujian ESP32

Pada pengujian ESP32 kali ini adalah mengunggah source code diatas ke ESP32. Namun sebelumnya ketika akan mengunggah program kedalam esp32, terlebih dahulu ada beberapa hal yang harus dirubah untuk mendukung proses pengiriman data kepada esp32. Seperti halnya upload speed untuk pengiriman data ke esp32 harus memiliki jumlah sebesar 115200, selain itu juga untuk frekuensi yang digunakan adalah sebesar 40 Hz. Berikut ini adalah hasil dari pengujian koneksi ESP32 :

```

Connecting to punkseat
.....
WiFi connected
IP address:
192.168.100.134

```

Gambar 4.14 Hasil Pengujian ESP32

Gambar diatas adalah hasil pengaplikasian source code diatas kepada ESP32. Setelah di unggah ke ESP32, program akan berjalan sesuai dengan

perintah perintah yang sudah ditentukan yaitu dapat menyambung dengan baik ke jaringan wifi. Ketika serial monitor menampilkan sebuah pesan “connecting”, menandakan bahwa mikrokontroler akan berusaha menghubungkan terhadap sebuah jaringan, ketika mikrokontroler tersebut mengalami kegagalan untuk terhubung kedalam sebuah jaringan maka akan muncul pesan “Connecting Failed” dan pesan tersebut akan terus mengulang dan mencoba untuk menghubungkan, namun jika tetap gagal maka serial monitor akan terus mengirim pesan “connecting failed” dan “connecting”. Namun ketika program yang dikirimkan kepada mikrokontroler ESP32 berhasil terhubung kepada sebuah jaringan, maka akan muncul pesan “connected” dan pesan tersebut hanya akan muncul sebanyak satu kali, namun ketika saat sudah terhubung dan secara tiba-tiba sambungan pada jaringan terputus, maka akan langsung memunculkan pesan “Connecting” sampai perangkat tersebut terhubung kembali.

4.8.2 Pengujian Sensor MLX90614

Pada tahapan pengujian sensor ini bertujuan untuk mengetahui kinerja serta akurasi dari sensor yang nantinya akan digunakan untuk melakukan pemantauan terhadap suhu klem sambungan kubikel incoming. Dimana nantinya pada sensor ini akan bekerja dengan mengeluarkan sebuah inframerah yang berasal dari titik *transceiver* pada bagian sensor hingga sinar inframerah tersebut mengenai sebuah objek. Kemudian sinar inframerah itu pula yang digunakan untuk membantu dalam proses transmisi data kepada titik *receiver*, dimana titik tersebut berfungsi untuk menerima data yang sebelumnya sudah dilakukan dengan proses pemantauan inframerah. Setelah itu data tersebut akan dikirimkan kepada

mikrokontroler untuk selanjutnya akan diolah kembali menjadi sebuah data yang dapat digunakan untuk memantau kondisi suhu dari klem sambungan kubikel *incoming*. Berikut merupakan program yang digunakan untuk melakukan pengujian terhadap sensor MLX90614 yang digabungkan dengan mikrokontroler ESP32:

```
Serial.print("Suhu Sekitar = ");
Serial.print(mlx.readAmbientTempC());
Serial.print("*C\tSuhu Objek = ");
Serial.print(mlx.readObjectTempC());
Serial.println("*C");

String Batas = "30";
KirimDataSuhu = String(mlx.readObjectTempC());
SuhuSekitar = mlx.readAmbientTempC();
Serial.print("Isi Data: ");
Serial.println(KirimDataSuhu);
```

Gambar 4.15 Source Code Pengujian MLX90614

Pada gambar program pengujian yang ada di atas merupakan sebuah algoritma yang digunakan untuk melakukan pemantauan suhu dari klem kubikel *incoming*. Pada program tersebut juga memiliki sebuah algoritma, dimana data yang nantinya akan berupa dari dalam bentuk analog akan dikonversi kedalam sebuah data dalam bentuk digital. Hal tersebut digunakan agar nantinya para petugas dapat lebih mudah untuk melakukan sebuah pemantauan terhadap suhu yang didapat oleh sensor. Berikut merupakan hasil dari pengujian sensor yang ditampilkan pada sebuah serial monitor arduino:

```
Connecting to punkseat
.....
WiFi connected
IP address:
192.168.100.134
Suhu Sekitar: 32 Derajat
Suhu Objek: 45 Derajat

Suhu Sekitar: 31 Derajat
Suhu Objek: 44 Derajat
```

Gambar 4.16 Hasil Pengujian Sensor MLX90614

Pada gambar diatas merupakan sebuah hasil pemantauan terhadap suhu dari klem sambungan kubikel incoming yang dilakukan oleh sensor MLX90614. Dimana data yang ditampilkan pada serial monitor adalah data yang sudah dikonversi dari data yang sebelumnya berbentuk analog menjadi sebuah data yang berbentuk digital. Pada gambar tersebut menunjukkan bahwa suhu dari klem sambungan kubikel incoming yang diukur menggunakan sensor MLX90614 memiliki suhu di angka 45 derajat dengan satuan *celcius*. Kemudian data tersebut juga akan dibandingkan dengan sebuah peralatan konvensional yang biasa digunakan oleh para petugas, hal tersebut bertujuan untuk membandingkan apakah data yang didapat sudah sesuai atau belum. Selanjutnya data tersebut akan digunakan pula sebagai acuan, jika nantinya sebuah sensor mendeteksi suhu dari klem sambungan kubikel incoming diatas 36 derajat, maka sistem akan mengirimkan sebuah pemberitahuan kepada petugas melalui aplikasi telegram. Kemudian data yang sudah didapat akan dikirimkan kepada server melalui protokol MQTT. berikut merupakan program yang digunakan untuk melakukan pengiriman data suhu klem sambungan klem kubikel incoming menuju sebuah

server.

```
String Batas = "36";

String Batas2 = "36";
KirimDataSuhu = String(mlx.readObjectTempC());
SuhuSekitar = String(mlx.readAmbientTempC());
Serial.print("Isi Data Object: ");
Serial.println(KirimDataSuhu);
Serial.print("Isi Data Sekitar: ");
Serial.println(SuhuSekitar);
```

Gambar 4.17 Pengujian Pengiriman Data MLX90614

Pada gambar diatas merupakan program yang digunakan untuk melakukan pengiriman data yang berasal dari sebuah mikrokontroler menuju server melalui protokol MQTT. Dimana pada program tersebut terdapat sebuah fungsi yang digunakan untuk melakukan pengiriman data kepada server dengan isian data yang berasal dari sebuah variabel data suhu. Data yang berada pada variabel data suhu adalah sebuah data yang berisikan nilai dari pengukuran suhu klem sambungan kubikel incoming yang dilakukan sebelumnya. Berikut merupakan hasil merupakan hasil dari pengiriman data yang dilakukan oleh mikrokontroler yang diuji coba menggunakan sebuah aplikasi MQTT dashboard



Gambar 4.18 Mqtt Dashboard

Pada gambar diatas menunjukkan bahwa data yang sebelumnya dikirimkan oleh mikrokontroler kepada server sudah dapat ditampilkan pada sebuah dashboard pengujian menggunakan MQTT dashboard. Dimana data yang ditampilkan pada aplikasi MQTT dashboard ini adalah data yang dapat diakses melalui jaringan apapun secara online dan dimanapun berada selamat terkoneksi kedalam jaringan internet

4.9 Pengujian Perangkat Lunak

Pada tahapan pengujian perangkat lunak disini bertujuan untuk memastikan semua bagian dari sistem monitoring pemantauan suhu klem sambungan kubikel incoming yang akan dilakukan pada penelitian tugas akhir ini dapat berjalan dengan baik dan sesuai dengan konsep yang sudah ditentukan diawal. Pengujian perangkat lunak sendiri nantinya akan meliputi beberapa tahapan, seperti pengujian protokol pengiriman data MQTT, pengujian server atau *virtual cloud server*, pengujian dashboard node-red dan pengujian database MYSQL.

4.9.1 Pengujian Protocol MQTT

Pada pengujian ini bertujuan untuk mengetahui berfungsi atau tidaknya protokol MQTT yang nantinya akan digunakan sebagai pen jembatan komunikasi antara perangkat keras dengan perangkat lunak yang dilakukan pada penelitian ini. Selain itu pengujian protokol MQTT ini digunakan untuk mengetahui kesiapan dari sebuah broker untuk kemudian akan diterapkan pada keseluruhan sistem. Pada tahap awal pengujian protokol ini adalah dengan melakukan pemasangan protokol pada sisi server yang sebelumnya sudah memiliki ip publik, sehingga dapat diakses melalui jaringan luar. Server yang digunakan

sendiri menggunakan operating system ubuntu 20.04 dengan konsep command line interface.

```
kentang@L745: ~
login as: kentang
kentang@192.168.43.249's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

6 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Mar  9 13:15:27 2022 from 192.168.0.118
kentang@L745:~$
```

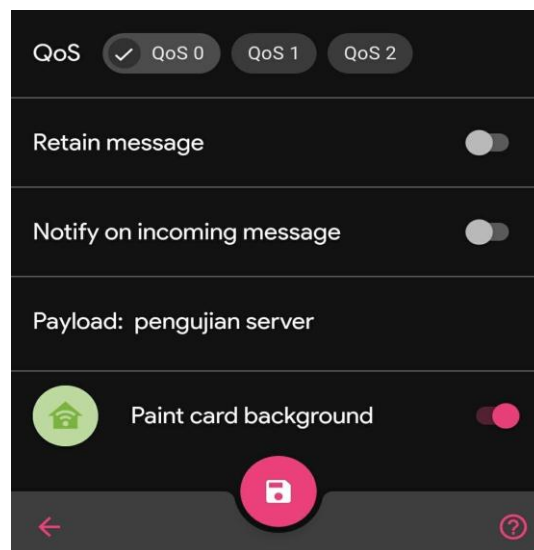
Gambar 4.19 Virtual Private Server

Pada gambar diatas merupakan bagian yang menunjukkan tampilan awal dari sistem operasi yang ada di virtual private server, dimana gambar tersebut menunjukkan pengguna dianggap masuk sebagai root. Selain itu pada awal masuk ke dalam sebuah server akan terdapat beberapa indikator, seperti aktivitas terakhir mengakses dan memperbaharui tanggal dari sistem operasi berdasarkan time zone waktu pertama kali memasang sistem operasi. Setelah itu akan dilakukan pengujian dari versi protokol MQTT yang digunakan pada penelitian tugas akhir ini, dimana nantinya server tersebut akan digunakan mampu diakses dari jaringan luar.

```
ipin@ipin-Lenovo-G40-70:~$ mosquitto -v
[ 2191.470092]~DLT-32260~INFO ~FIFO /tmp/dlt cannot be opened. Retrying late
r...
1626239162: mosquitto version 1.6.9 starting
1626239162: Using default config.
1626239162: Opening ipv4 listen socket on port 1883.
1626239162: Error: Address already in use
[ 2201.489959]~DLT-32260~WARNING ~Lost log messages in user buffer when exiting: 6
ipin@ipin-Lenovo-G40-70:~$
```

Gambar 4.20 Versi Protokol MQTT

Pada gambar diatas merupakan tampilan yang menunjukkan virtual private server telah terpasang protokol MQTT, dimana dengan ditunjukkan versi yang terpasang yaitu menggunakan versi mosquitto v1.6.9. Versi tersebut merupakan versi yang terbaru dari aplikasi protokol MQTT yang menggunakan mosquitto, dimana versi tersebut yang dianggap stabil dari versi peluncuran-peluncuran yang lainnya. Setelah protocol MQTT berhasil dipasang pada sebuah server, maka selanjutnya adalah bagian yang digunakan untuk melakukan pengujian apakah server MQTT dapat menerima data yang dikirimkan oleh pengguna, berikut merupakan aplikasi MQTT dashboard yang digunakan untuk melakukan mengirimkan data:



Gambar 4.21 Aplikasi Pengujian MQTT

Pada gambar diatas merupakan sebuah aplikasi yang digunakan untuk melakukan percobaan mengirimkan sebuah data dummy dalam bentuk String dengan keterangan yaitu “pengujian server”. Jika pengujian pengiriman data tersebut sukses maka pada server pun juga akan mendapatkan sebuah data yang dikirimkan oleh aplikasi MQTT dashboard. Berikut merupakan hasil dari

pengujian penerimaan data dari sisi server:



```

bakso@DESKTOP-GVSCO0E: ~
bakso@DESKTOP-GVSCO0E:~$ mosquitto_sub -h 172.17.53.123 -t iotdev/# -u skripsi -P 2022
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server
Pengujian Server

```

Gambar 4.22 Server Menerima Data

Pada gambar diatas menunjukkan bahwa server mampu menerima data yang sebelumnya sudah dikirimkan oleh aplikasi MQTT dashboard. Dimana pada tampilan command line tersebut ditunjukkan data dengan keterangan “pengujian server” yang sudah sesuai dengan data yang dikirimkan oleh aplikasi. Hal tersebut membuktikan bahwa server MQTT sudah dapat digunakan dengan baik.

4.9.2 Pengujian Penampilan Data Pada Dashboard *Node-red*

Setelah pengujian perangkat-perangkat keras yang nantinya akan digunakan pada penelitian ini dan pengujian server yang sudah dipasang protokol MQTT sebelumnya, maka selanjutnya adalah pengujian untuk menampilkan data hasil pemantauan suhu yang dilakukan oleh sensor pada dashboard node-red. Tahapan pengujian ini adalah menampilkan sebuah data dummy yang sebelumnya sudah dikirim oleh perangkat keras pada server. Namun sebelum itu perlu dilakukan menjalankan framework node-red pada virtual private server.

```

kentang@L745: ~
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Mar  9 13:21:28 2022 from 192.168.43.203
kentang@L745:~$ node-red
9 Mar 13:37:24 - [info]

Welcome to Node-RED
=====

9 Mar 13:37:24 - [info] Node-RED version: v2.1.4
9 Mar 13:37:24 - [info] Node.js version: v16.13.2
9 Mar 13:37:24 - [info] Linux 5.13.0-32-generic x64 LE
9 Mar 13:37:24 - [info] Loading palette nodes
9 Mar 13:37:26 - [info] Dashboard version 3.1.3 started at /ui
9 Mar 13:37:26 - [info] Settings file   : /home/kentang/.node-red/settings.js
9 Mar 13:37:26 - [info] Context store  : 'default' [module=memory]
9 Mar 13:37:26 - [info] User directory : /home/kentang/.node-red
9 Mar 13:37:26 - [warn] Projects disabled : editorTheme.projects.enabled=false
9 Mar 13:37:26 - [info] Flows file     : /home/kentang/.node-red/flows.json
9 Mar 13:37:26 - [info] Server now running at http://127.0.0.1:1880/
9 Mar 13:37:26 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

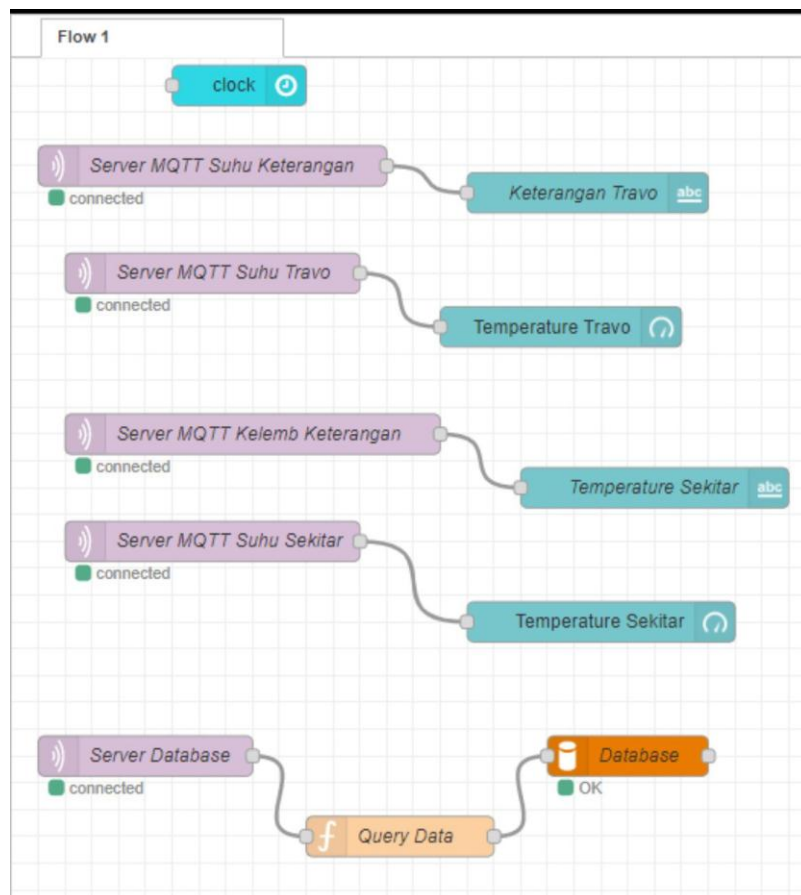
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

9 Mar 13:37:26 - [info] Starting flows
9 Mar 13:37:26 - [info] Started flows

```

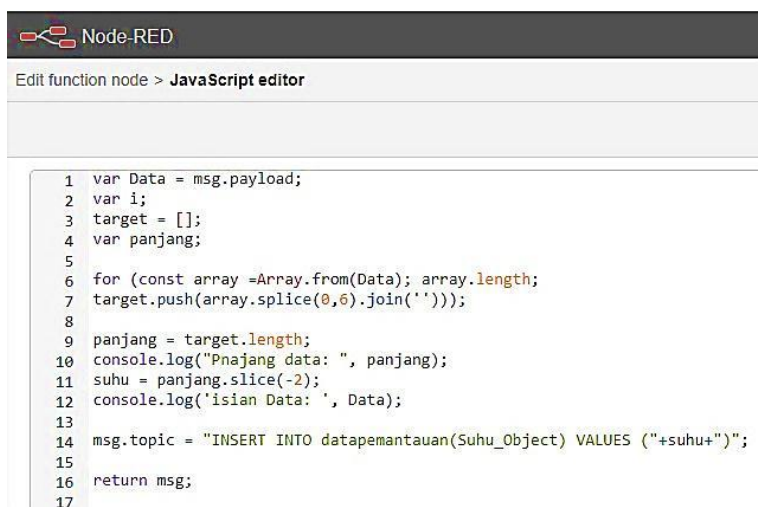
Gambar 4.23 Menjalankan Node-red Dashboard

Pada gambar diatas merupakan sebuah proses untuk menjalankan framework node-red melalui command line, dimana dengan perintah “node-red” secara otomatis program node-red akan berjalan dengan sendirinya. Program node-red sendiri akan berjalan pada sebuah localhost, namun pada penelitian tugas akhir ini pengguna nantinya dapat mengakses ip ataupun domain yang sebelumnya sudah dipasang pada sisi server. Setelah itu yang selanjutnya akan dilakukan adalah proses untuk konfigurasi pada workspace dari node-red. Setelah sudah berjalan, tinggal mengakses local ip dengan port 8000 yang selanjutnya akan dilakukan konfigurasi seperti yang ditunjukkan pada gambar dibawah ini:



Gambar 4.24 Konfigurasi Node-red Dashboard

Pada gambar diatas merupakan sebuah tahapan konfigurasi yang dilakukan pada web dashboard node-red, dimana hal tersebut dibutuhkan agar nantinya data yang dikirim oleh sensor akan dapat ditampilkan pada dashboard. Selain itu konfigurasi ini meliputi inialisasi dari server yang akan digunakan, dengan memasukkan ip publik dari server mqtt yang digunakan. Selanjutnya proses tersebut akan diterapkan pada sensor-sensor yang lainnya pula. Kemudian tahapan selanjutnya adalah melakukan konfigurasi untuk memberikan perintah dari framework node-red untuk dapat melakukan query data menuju aplikasi penyimpanan yaitu database MYSQL, seperti yang ditunjukkan pada gambar dibawah ini :



```

1 var Data = msg.payload;
2 var i;
3 target = [];
4 var panjang;
5
6 for (const array =Array.from(Data); array.length;
7 target.push(array.splice(0,6).join('')));
8
9 panjang = target.length;
10 console.log("Panjang data: ", panjang);
11 suhu = panjang.slice(-2);
12 console.log('Isian Data: ', Data);
13
14 msg.topic = "INSERT INTO datapemantauan(Suhu_Object) VALUES (" +suhu+)";
15
16 return msg;
17

```

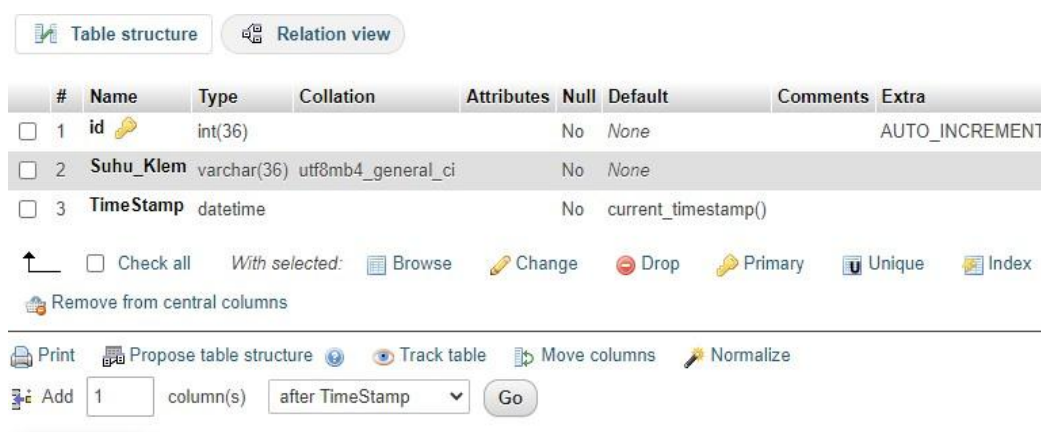
Gambar 4.25 Query Data Kedalam Database

Pada gambar diatas menunjukkan bahwa source code yang digunakan untuk melakukan query ke dalam database adalah menggunakan bahasa pemrograman javascript. Dimana pada source code tersebut juga akan dilakukan inisialisasi dari nama database yang digunakan, nama kolom yang digunakan dan juga terdapat algoritma yang digunakan untuk melakukan parsing data. Hal tersebut digunakan agar data yang dikirim oleh mikrokontroler tidak dipecah dalam beberapa bagian, namun akan dilakukan pemisahan data-data tersebut pada sisi mikrokontroler agar tidak membuat proses pada mikrokontroler terlalu berat sehingga kinerja dari perangkat-perangkat akan tetap stabil dan tahan lama.

4.9.3 Pengujian Database MYSQL

Pada tahapan pengujian perangkat lunak database ini merupakan bagian yang digunakan untuk melakukan konfigurasi dan juga untuk memastikan semua bagian dalam database dapat bekerja dengan baik sebagai perangkat lunak yang akan menyimpan data. Dimana pada penelitian tugas akhir ini akan digunakan perangkat lunak penyimpanan data dengan jenis MYSQL. Tahapan awal pada

penelitian tugas akhir ini adalah dengan melakukan konfigurasi dengan membuat beberapa kolom - kolom yang nantinya akan berisi sebuah data yang sebelumnya dikirim oleh mikrokontroler yang didapat dari pemantauan sensor yang digunakan. Pada penelitian tugas akhir ini akan dimasukkan beberapa data yang digunakan untuk pengolahan data selanjutnya, dimana data yang dimasukkan ke dalam tabel nantinya adalah data suhu dari klem sambungan kubikel incoming dan juga data waktu pemantauan. Berikut merupakan konfigurasi yang dilakukan pada perangkat lunak database MYSQL:



The screenshot shows the MySQL Table Structure view for a table. It features a table with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id	int(36)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	Suhu_Klem	varchar(36)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 3	TimeStamp	datetime			No	current_timestamp()		

Below the table, there are various control options including 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', and 'Remove from central columns'. At the bottom, there are options for 'Print', 'Propose table structure', 'Track table', 'Move columns', and 'Normalize'. A 'Go' button is also present at the bottom right.

Gambar 4.26 Tabel Database MYSQL

Pada gambar diatas menunjukkan beberapa kolom yang berjumlah 4 pada sebuah tabel database MYSQL, dimana pada kolom pertama akan digunakan sebagai inisialisasi dari penomoran dari tabel. Pada kolom penomoran sendiri akan diterapkan tipe Auto Increment, dimana dengan tipe tersebut dapat membuat database secara otomatis memberikan label penomoran yang nantinya dapat dimanfaatkan untuk penentuan kata kunci berdasarkan nomor urut. Selain itu terdapat pula bagian kolom data suhu akan menggunakan jenis tipe data varchar, karena nantinya data yang dikirimkan oleh mikrokontroler bisa jadi berubah data

bilangan desimal maupun bilangan bulat. Maka dari itu perlunya disiapkan dari segi infrastruktur agar nantinya database dapat secara fleksibel menerima data dalam bentuk apapun. Begitupun pula dengan kolom ketiga yang berisikan data kelembaban yang menggunakan tipe data varchar. Sedangkan pada bagian kolom history waktu pengambilan data pemantauan suhu klem sambungan kubikel incoming ini akan dibuat secara otomatis menyesuaikan data masuk dengan zona waktu dimana perangkat ini digunakan, jadi tidak perlu lagi memasukkan waktu secara manual ke dalam dashboard ataupun database nantinya, karena semua sudah dikerjakan secara otomatis. Kemudian untuk database sendiri nantinya akan dipasang pada sebuah virtual private server yang berada di cloud. Sehingga sebelum melakukan konfigurasi pada database MYSQL perlu dilakukan beberapa tahap instalasi perangkat lunak pendukung lainnya. Seperti melakukan pemasangan web server, dimana jenis web server yang digunakan pada penelitian tugas akhir ini menggunakan jenis Apache yang bersifat open source. Selain itu dilakukan pula pemasangan bahasa pemrograman PHP Hypertext Preprocessor dengan versi keluaran 7.2 yang disarankan oleh beberapa jurnal karena sudah stabil.

4.9.4 Pengujian Telegram Bot

Pada tahapan pengujian sistem notifikasi menggunakan telegram bot ini digunakan untuk memastikan apakah sistem pemberitahuan menggunakan aplikasi telegram dapat bekerja dengan baik. Pada tahapan ini nantinya juga akan meliputi konfigurasi yang dilakukan pada aplikasi telegram, dengan memanfaatkan bot yang sebelumnya sudah dikembangkan oleh orang lain dengan

lisensi terbuka dan dapat digunakan bebas oleh orang lain. Berikut merupakan bagian program yang digunakan untuk mengirimkan sebuah pesan menuju aplikasi telegram:

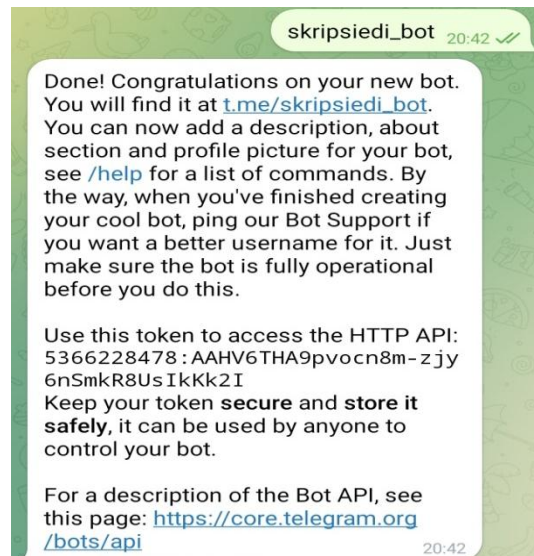
```
myBot.setTelegramToken(token);
if (myBot.testConnection()) {
    Serial.println("Koneksi Bagus");
} else {
    Serial.println("Koneksi Jelek");
}

myBot.sendMessage(id, "Alat Menyala");
Serial.println("Pesan Terkirim");

}
```

Gambar 4.27 Program Mengirim Data Menuju Telegram

Pada gambar diatas merupakan bagian program yang digunakan untuk melakukan pengiriman pesan melalui Rest Api menuju aplikasi telegram. Terdapat beberapa fungsi yang digunakan untuk inialisasi data yang digunakan, seperti inialisasi dari token api untuk menghubungkan perangkat keras dengan aplikasi telegram. Selain itu terdapat pula bagian yang digunakan untuk mengisikan teks atau isi pesan yang akan disampaikan kepada pengguna melalui aplikasi telegram. Selanjutnya adalah melakukan konfigurasi pada bagian aplikasi telegram dengan menggunakan “bot father”. Berikut merupakan konfigurasinya :



Gambar 4.28 Konfigurasi Pada Aplikasi Telegram

Gambar diatas adalah bagian yang digunakan untuk melakukan pemasangan bot yang nantinya akan digunakan untuk menerima data dari mikrokontroler. Setelah itu pada bagian ini pula akan digunakan untuk mengambil sebuah token yang digunakan untuk inialisasi dari api telegram nantinya. Berikut merupakan hasil dari chat telegram bot yang menerima pesan dari mikrokontroler:



Gambar 4.29 Pesan Notifikasi Melalui Telegram

Pada gambar diatas merupakan bagian yang menunjukkan bahwa proses pengiriman pesan yang dilakukan oleh mikrokontroler pada aplikasi telegram

dapat berjalan dengan baik. Dimana pada aplikasi telegram akan menerima pesan dengan isian “suhu dari kubikel mengalami kenaikan”, hal tersebut menunjukkan bahwa aplikasi telegram akan menerima pesan jika kondisi dari suhu diatas batas normal yang sudah ditentukan dan harus segera dilakukan tindakan selanjutnya oleh para petugas.