



MONITORING TEMPERATURE CHILLER WATER MENGUNAKAN ANDROID BERBASIS IOT (INTERNET OF THINGS)

Rachmat Dwi Pranata¹

¹Jurusan Informatika, Universitas Muhammadiyah Sidoarjo
pranataadwi5@gmail.com

STATUS ARTIKEL

Dikirim 10 April 2022
Direvisi 11 Mei 2022
Diterima 12 Juni 2022

Kata Kunci:

Kata Kunci1, IoT, Monitoring, Suhu

ABSTRAK

Temperatur chiller water dalam proses pembuatan pipa PVC merupakan faktor utama untuk menjaga kualitas finish good produk. Temperatur chiller water tersebut perlu dipantau agar setiap saat kondisi suhu tetap pada range yang ditentukan, sehingga dibutuhkan sistem pemantau yang real time dan dimanapun dapat diketahui. Salah satu alternatifnya adalah memanfaatkan teknologi internet melalui pendekatan Internet of Things (IoT). Tulisan ini bertujuan memberikan gambaran model pemantau pengendali suhu ruang berbasis IoT. Sensor Ds18b20 di implementasikan untuk monitoring chiller water dan alarm yang mengirimkan notifikasi pada smartphone android. Sistem yang dikembangkan menggunakan Node MCU sebagai pemroses data dan mengirimkan secara nirkabel menggunakan ESP 8266 menuju Internet dan diterima oleh smartphone. Hasil pengujian menunjukkan data temperatur yang terdeteksi mendekati temperatur yang terukur menggunakan thermometer standar.

1. PENDAHULUAN

1.1 Pendahuluan

Seiring dengan perkembangan teknologi yang semakin maju dari waktu ke waktu maka pengembang terus berusaha menciptakan sistem pemantauan yang lebih efisien, manusia menciptakan sistem keamanan mesin dapat berupa sistem pemantauan dapat didukung melalui teknologi yang terhubung dan terintegrasi dengan peralatan IT dan Electric yang dapat setting sesuai dengan kebutuhan, sebagai contoh bisa memunculkan warning alarm saat high temperatur pada water chiller.

Water chiller menjadi salah satu elemen yang penting dalam pembuatan produk contohnya dalam pembuatan produk pipa PVC pada PT. Wahana Tunas Utama Rucika. Standart temperatur untuk pembuatan produk pipa PVC 13°C sampai dengan maksimal 16°C. Apabila temperature melebihi standart atau batas maksimal yang telah ditentukan akan berakibat pada kegagalan produk pipa tersebut. Untuk menghindari kejadian tersebut operator harus lebih teliti dan konsiten untuk mengontrol temperatur water chiller agar sesuai dengan standart yang telah di tentukan. Sering juga terjadi kelalaian operator untuk menjaga stardart dan maksimal temperatur water chiller yang telah di tentukan, dampaknya pada gagal produk dan menambah reject pada produksi. Tim utility yang bertanggung jawab terhadap temperatur water chiller juga handle power listrik, distribusi air bersih, dan angin bertekanan

seluruh pabrik. Dengan itu mobilitas dan lingkup area kerja tim utility sangat luas bertentangan dengan radius sirine saat ada masalah dengan temperature water chiller yang ± 15 meter dan bersamaan dengan suara mesin lainnya. Itu yang membuat tim utility susah untuk mengontrol temperatur water chiller apabila bekerja di luar radius sirine.

1.2 Monitoring Berbasis IoT (Internet of Things)

Internet of Things (IoT) adalah kumpulan benda-benda (*things*), berupa perangkat fisik (*hardware/embedded system*) yang mampu bertukar informasi antar sumber informasi, operator layanan ataupun perangkat lainnya yang terhubung kedalam sistem sehingga dapat memberikan manfaat yang lebih besar. Untuk membuat aplikasi berbasis IoT membutuhkan mikrokontroler seperti NodeMCU ESP8266 atau mikrokontroler lainnya.

NodeMCU adalah *platform IoT open source*. NodeMCU *firmware* yang berjalan pada ESP8266 Wi-Fi SoC yang dirancang oleh *SistemEspressif* yang didasarkan pada Modul ESP-12. Istilah "NodeMCU" secara default mengacu pada *firmwareDevKit*. *Firmware* menggunakan bahasa *scriptingLua* dan dapat digunakan dalam beberapa projek seperti *lua-cjson*, dan *spiff*.



Gambar 1.1 NodeMCU ESP8266

Untuk menentukan alur program dan seperti apa nodeMCU ESP8266 akan bekerja, langkah pertama membuat code pada aplikasi arduino ide. Bahasa pemrograman yang digunakan adalah bahasa C.

```
Main | Arduino 1.8.12
File Edit Sketch Tools Help
Main secrets.h
#include <ESP8266WiFi.h>
#include "FirebaseESP8266.h"
#include "ThingSpeak.h"
#include "secrets.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2

#define FIREBASE_HOST "https://monitoring-suhu-1111-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "WZ1AgSituEZMuQRa79yhZtBuzRtAc32tFy7oQe4L"
#define WIFI_SSID "iPhone"
#define WIFI_PASSWORD "Beeneluv"
```

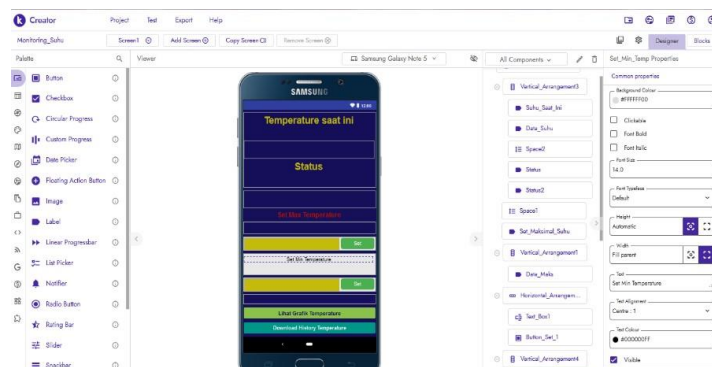
Gambar 1.2 Arduino ide

1.3 Firebase

Untuk membuat aplikasi berbasis IoT yang bertema monitoring temperature membutuhkan data realtime temperature yang di dapat dari inputan sensor yang terprogram dengan nodeMCU ESP8266. Firebase sebagai database yang dimiliki oleh google memiliki beberapa fitur yang dapat digunakan developer aplikasi android maupun IOS. Fitur yang sesuai dengan aplikasi yang saya buat adalah realtime database yang akan jadi penghubung antara hasil inputan dari nodeMCU dan menampilkan pada aplikasi android.

1.4 Aplikasi Android

Setelah membuat program pada nodeMCU dan membuat database pada firebase, selanjutnya membuat aplikasi android. Untuk pembuatan aplikasi android saya menggunakan kodular creator. Kodular adalah sebuah situs web, yang menyediakan tools yang untuk membuat aplikasi android dengan konsep drag-drop block programming.



Gambar 1.3 Kodular Creator

1.5 Sebelum Adanya Monitoring Temperature Menggunakan Smartphone

Sebelum adanya aplikasi ini, operator mengandalkan tampilan digital yang menampilkan hasil pembacaan temperature chiller water pada end point atau titik terjauh yang di aliri oleh chiller water. Apabila terdapat abnormality pada temperature sirine akan berbunyi. Diketahui dengan cara tersebut masih kurang flexibel dengan mobilitas area yang dijangkau operator sangat luas saat dildalam pabrik.

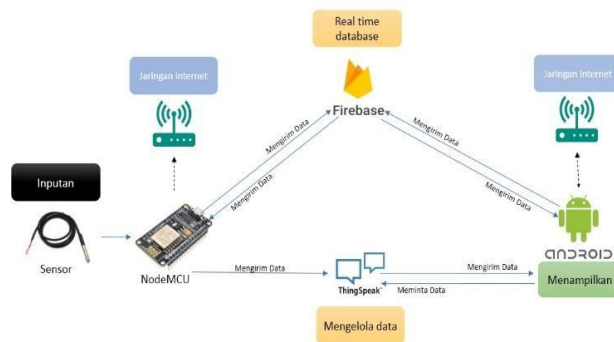
1.6 Langkah Berjalannya Aplikasi

- 1) Sensor DS18B20 terasang pada nodeMCU
- 2) NodeMCU mendapatkan power dc 3 volt
- 3) NodeMCU terhubung dengan jaringan internet
- 4) Smartphone terhubung dengan jaringan internet
- 5) Operator membuka aplikasi monitoring temperature
- 6) Operator setting batas maksimal & batas minimal tempeprature
- 7) Apabila temperature melebihi dari batas yang telah di setting, aplikasi akan mengirimkan push notifikasi.

1.7 Flow Proses Aplikasi

Sensor temperature yang sudah terpasang sebagai sebuah masukan. Sensor tersebut terhubung dengan mikrokontroler yang bertugas mengolah data yang didapat, diteruskan pada app server, dan di lanjutkan pada app smartphone. Mikrokontroler akan merespon

perubahan suhu dengan memunculkan peringatan sesuai dengan settingan standart yang telah di tetapkan. User menerima notifikasi pada smartphone apabila terjadi upnormality pada temperatur air bak chiller. sensor *temperature*, akan berperan sebagai masukan untuk nodeMCU. Firebase yang akan menerima masukan dari NodeMCU dan menampilkan data dengan *real time database*, setelah itu smartphone android kita yang terkoneksi dengan jaringan internet meminta data pada firebase dan menampilkan data tersebut melalui tampilan pada app yang ada pada smartphone android. NodeMCU juga mengirim data hasil pembacaan inputan pada *thingspeak* yang akan mengelola data tersebut menjadi grafik dan tabel history temperature. Thingspeak mengirimkan data pada smartphone untuk kebutuhan update grafik yang ada pada aplikasi, smartphone juga meminta data pada thingspeak pada saat pengguna aplikasi ingin melihat history temperature yang sudah dikelola thingspeak.



Gambar 1.4 Flow proses Aplikasi

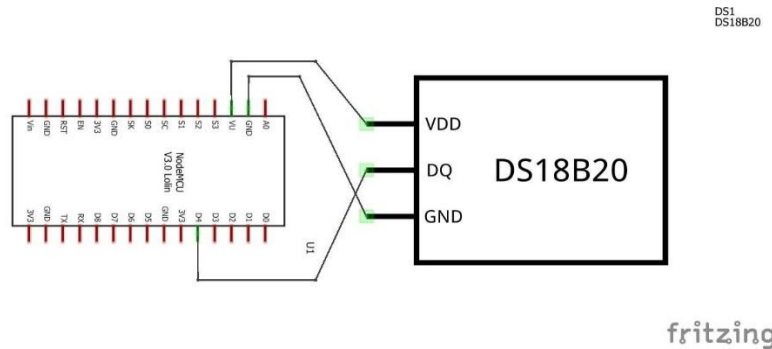
2. METODE

Metode yang digunakan yaitu memanfaatkan teknologi IoT yang berkembang pesat, banyak digunakan di segala bidang, menjadi improvement atau pembaruan bagi metode yang kuno, dan yang jelas mempermudah pekerjaan. Dengan menggunakan IoT dalam memonitoring temperature chiller water operator dapat melakukan pengecekan temperature tanpa harus berpindah tempat, operator juga dapat menghemat tenaga dan waktu kerja dalam pengecekan temperature. Warning temperature yang menggunakan cara lama yaitu dengan sirene juga kurang efektif saat operator tidak berada di radius sirene atau tidak mendengar suara sirene. Oleh karena operator telat dalam menangani problem pada temperature dan dapat menyebabkan efek pada hasil produksi (reject).

3. HASIL DAN PEMBAHASAN

3.1 Pembuatan Program pada NodeMCU

Untuk program pembacaan temperature menggunakan nodeMCU yang disambungkan dengan inputan yaitu sensor ds18b20, berikut koneksi antara sensor dan nodeMCU:



Gambar 3.1 Koneksi NodeMCU ESP8266 dengan Sensor ds18b20

Sensor ds18b20 memiliki 3 keluaran kabel yang berwarna Hijau, Kuning dan Orange. Kabel warna hijau sebagai koneksi ground, kabel warna kuning sebagai data, dan kabel warna orange sebagai tegangan. Berikut ini adalah kode program untuk maksimal dan minimal temperature yang di upload pada NodeMCU:

```
#include <ESP8266WiFi.h>
#include "FirebaseESP8266.h"
#include "ThingSpeak.h"
#include "secrets.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2

#define FIREBASE_HOST "https://monitoring-suhu-1111-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "WZ1AgSituEZMuQRa79yhZtBuzRtAc32tFy7oQe4L"
#define WIFI_SSID "iPhone"
#define WIFI_PASSWORD "Beeneluv"

unsigned long myChannelNumber = SECRET_CH_ID;
const char* myWriteAPIKey = SECRET_WRITE_APIKEY;

FirebaseData firebaseData0;

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
WiFiClient client;
```

Gambar 3.2 Coding Program

Setelah code program selesai dan di upload pada nodeMCU, berikut hasil running program pada nodeMCU ESP8266:

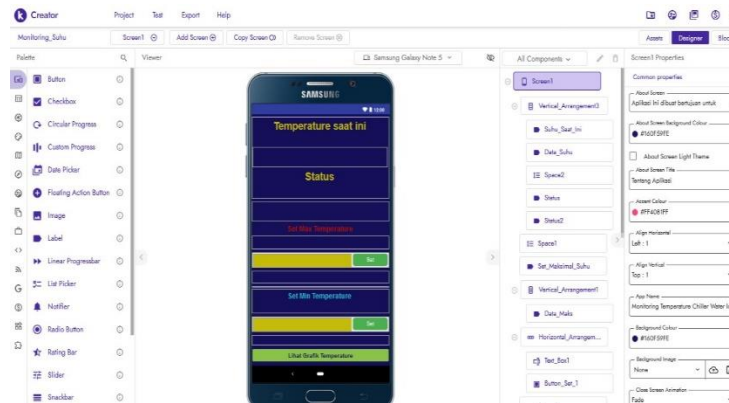


Gambar 3.3 Hasil Program

3.2 Pembuatan Aplikasi Android

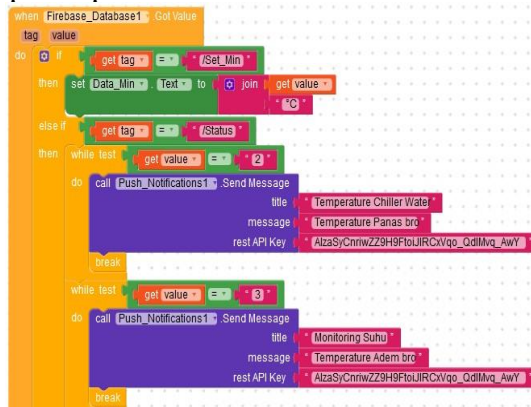
Pembuatan aplikasi android menggunakan website [creato.kodular](https://creato.kodular.com). Dalam membuat

aplikasi pada kodular pertama kita buat design user interface aplikasi terlebih dahulu. Berikut user interface aplikasi yang telah saya buat:



Gambar 3.4 Website kodulara.creator

Setelah menentukan design yang di inginkan untuk aplikasi, selanjutnya membuat coding blocks, berikut coding block pada aplikasi:



Gambar 3.5 Pembuatan Coding Blocks

3.3 Pengujian Aplikasi Android

Pastikan nodemcu mendapatkan power, terhubung dengan jaringan internet, dan sensor ds18b20 sudah terkoneksi dengan nodemcu. Kemudian buka aplikasi monitoring temperature chiller water pada smartphone, pastikan juga smartphone terhubung dengan jaringan internet. Rubah setting batas maksimal dan minimal untuk melakukan test apakah akan muncul push notifikasi yang menandakan muncul abnormality temperature pada smartphone. Berikut hasil test yang dilakukan:



Gambar 3.5 Hasil test Aplikasi

4. KESIMPULAN

Setelah dilakukan pembuatan alat ini dapat disimpulkan sebagai berikut :

1. Perancangan monitoring temperature ini dibuat dengan memanfaatkan koneksi internet dan NodeMCU untuk memberikan notifikasi pada aplikasi yang telah dibuat dan ter instal pada smartphone android.
2. Inputan yang dihasilkan dari pembacaan sensor ds18b20 dapat memberikan masukan pada mikrokontroler NodeMCU yang secara realtime merubah data yang ada pada firebase database.
3. Mikrokontroler NodeMCU pada tempat tidur dan aplikasi berbasis android yang telah dibuat dengan kodular dihubungkan melalui koneksi internet dengan metode IoT.
4. Setting batas maksimal & minimal temperatur dapat dilakukan pada aplikasi, dengan settingan temperature tersebut aplikasi dapat memberikan feedback berupa notifikasi pada smartphone dengan status yang sesuai dengan settingan temperature terhadap temperature realtime.
5. Aplikasi juga dapat melihat grafik temperature sesuai waktu yang telah di tentukan dan dapat mendownload history temperature dengan format file .xls .
6. Aplikasi monitoring temperature chiller water memunculkan notifikasi berdasarkan perubahan status temperature yang terbaca oleh sensor dan diterima oleh NodeMCU dan realtime database pada firebase. Setelah dilakukan proses pengujian aplikasi temperature chiller water yang dilakukan, dapat disimpulkan sebagai berikut:
7. Pada saat pengujian aplikasi menampilkan data temperature sesuai dengan pembacaan sensor dan realtime database.
8. Pada saat dilakukan pengujian setting temperature maksimal & minimal dari aplikasi alat akan merespon tindakan tetapi dengan sedikit waktu tunda. Lama waktu tunda tersebut ditentukan dari operator seluler yang kita pakai. Operator seluler dengan jaringan terbaik

maka waktu tundanya akan semakin sedikit. Berdasarkan percobaan yang telah dilakukan Telkomsel merupakan operator seluler terbaik.

9. Kondisi lingkungan dengan operator seluler komunikasi akan mempengaruhi aplikasi memunculkan notifikasi status temperature. Semakin buruk jaringan operator seluler yang tersedia maka akan semakin lama dalam mendapatkan kondisi terkini temperature.

5. DAFTAR PUSTAKA

- Admin. (2017, April 26). *Fungsi dan jenis-jenis chiller*. Diambil kembali dari Pengetahuan serta Informasi tentang MIGAS, Energi Terbarukan dan Pengilangan: <https://www.prosesindustri.com/2017/04/fungsi-danjenis-jenis-chiller.html>
- Admin. (2019, Oktober 22). *Arduino Sensor Suhu DS18B20* . Diambil kembali dari Ardutech.com: <https://www.ardutech.com/arduino-sensor-suhu-ds18b20/>
- Galih, S. (2017). *Aplikasi notifikasi keadaan suhu dan kelembaban udara*.
- Kartika Dwintaputri S, S. M. (2019). App Inventor. *Pengenalan App Inventor*.
- Maulana, J. (2018). *Implementasi model agile pada monitoring suhu kolam ikan dengan algoritma Fuzzy Logic berbasis IOT*.
- Muklis, R. K. (2018). *Perancangan Internet Of Things (IOT) Untuk Monitoring Suhu Budidaya Jamur*.
- Widiyaman, T. (2020, Oktober 17). *Mengenal Modul NodeMCU ESP8266, Sikecil yang handal untuk IoT*. Diambil kembali dari Warriornux: <https://www.warriornux.com/mengenal-nodemcu-esp8266-iot/>