

**RANCANG BANGUN SISTEM *MONITORING* HIDROPONIK
PADA *GREENHOUSE* DI PT INDMIRA**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Hasan Mubarak Gasmi

16524083

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta**

2020

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM *MONITORING* HIDROPONIK PADA *GREENHOUSE*

DI PT INDMIRA

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Hasan Mubarak Gasmi
16524083

الجمهورية الإسلامية
Yogyakarta, 22 Juli 2020

Menyetujui,

Pembimbing 1



Dwi Ana Ratna Wati, S.T., M.Eng.
NIK. 035240102

LEMBAR PENGESAHAN PENGUJI

SKRIPSI

RANCANG BANGUN SISTEM *MONITORING* HIDROPONIK PADA *GREENHOUSE*

DI PT INDMIRA

Dipersiapkan dan disusun oleh:

Hasan Mubarak Gasmi

16524083

Telah dipertahankan di depan dewan penguji

Pada tanggal: 13 Agustus 2020

Susunan dewan penguji

Ketua Penguji : **Dwi Ana Ratna Wati, S.T., M. Eng.,**

Anggota Penguji 1: **Medilla Kusriyanto, S.T., M.Eng.,**

Anggota Penguji 2: **Ida Nurcahyani, S.T., M.Eng.,**

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 24 Agustus 2020

Ketua Program Studi Teknik Elektro



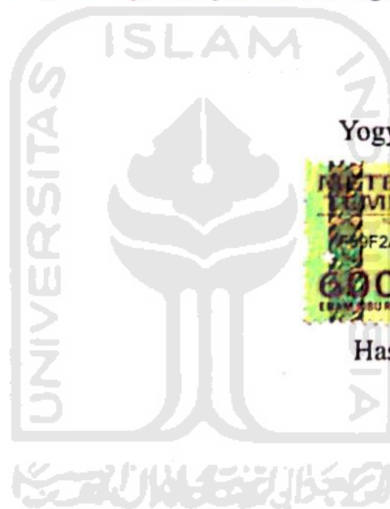
Yusuf Aziz Amrullah, S.T., M.Sc., Ph.D.

NIK. 045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.



Yogyakarta, 20 Juli 2020



Hasan Mubarak Gasmi

KATA PENGANTAR

Assalamualaikum. Wr. Wb.

Alhamdulillahirobbil ‘alamin, segala puji dan syukur penulis ucapkan karena berkat rahmat dan hidayahNya pelaksanaan masa kuliah di Universitas Islam Indonesia ini mencapai tahap akhir dan dapat menyelesaikan laporan skripsi ini dengan baik. Shalawat serta salam penulis haturkan kepada baginda Rasulullah Muhammad SAW. Laporan ini tidak akan terlaksana dengan baik tanpa bantuan dan masukan dari Dosen Pembimbing serta teman-teman dari Teknik Elektro. Untuk itu, penulis sampaikan terimakasih kepada pihak-pihak yang telah memberikan dukungan dan bimbingan dalam Tugas Akhir ini yaitu:

1. Bapak dan Ibu selaku kedua orang tua penulis, Kakak dan Adik penulis yang selalu mendoakan dan telah memberikan dukungan yang besar untuk menyelesaikan laporan Skripsi ini.
2. Ibu Dwi Ana Ratna Wati, S.T., M. Eng, selaku dosen pembimbing dari penulis yang sudah membantu dan membimbing penulis dengan penuh kesabaran dalam menyelesaikan laporan skripsi ini.
3. Bapak Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D. Selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia
4. Seluruh dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama menempuh kuliah dari semester pertama hingga akhir di Jurusan Teknik Elektro.
5. Teman-teman mahasiswa Teknik Elektro angkatan 2016 yang telah menemani masa-masa perantauan dan kebersamaannya selama berkuliah dan juga Bagas Dhanes K. dan Ayasi Bahifatih P. rekan seperjuangan di PT. Indmira
6. Mas Budi, mas Hanry Ario, mas Fahrul dan rekan-rekan di PT.Indmira yang telah membantu dan membimbing pelaksanaan Tugas Akhir di PT. Indmira

Terima kasih atas segala do'a dan dukungannya. Semoga Allah SWT membalas seluruh kebaikan kalian dan senantiasa meridhoi kita dalam setiap langkah menuju cita-cita Aamiin. Penulis menyadari laporan ini jauh dari sempurna sehingga penulis memohon maaf atas segala kekurangan dan kesalahan baik yang disengaja atau tidak disengaja.

Wassalamualaikum. Wr. Wb

ARTI LAMBANG DAN SINGKATAN

$^{\circ}\text{C}$: Derajat <i>Celcius</i>
ADC	: <i>Analog to Digital Converter</i>
DHT22	: <i>Digital Humidity Temperature Sensor - AM2302</i>
EEPROM	: <i>Electrically Erasable Programmable Read-Only Memory</i>
GSM	: <i>Global System for Mobile Communications</i>
HMI	: <i>Human Machine Interface</i>
HTC	: <i>Hygrometer temperature control</i>
IoT	: <i>Internet of Things</i>
L	: <i>Liter</i>
LCD	: <i>Liquid Crystal Display</i>
mL	: <i>Mililiter</i>
MODBUS	: Protocol Komunikasi Serial yang Dipublikasikan oleh Modicon
ms	: <i>Milliseconds</i>
PHP	: <i>Hypertext Preprocessor</i> (salah satu bahasa pemrograman untuk pengembangan Web)
PLC	: <i>Programmable Logic Controller</i>
RTC	: <i>Real Time Clock</i>
RX	: <i>Receiver</i>
SDCard	: Kartu Memori
TTL	: <i>Time to Live</i>
TX	: <i>Transmitter</i>
USB	: <i>Universial Serial Bus</i>
VDC	: <i>Voltage Direct Current</i>
Wi-Fi	: <i>Wireless Fidelity</i>

ABSTRAK

PT. Indmira merupakan salah satu perusahaan yang menerapkan konsep pertanian hidroponik dalam *greenhouse*. Hidroponik adalah salah satu teknik menanam tanpa menggunakan tanah melainkan menggunakan pipa dan air nutrisi. Namun, sistem ini perlu dipantau secara berkala dengan mengecek ke lokasi *greenhouse*. Karena, semakin besar umur tanaman jumlah konsumsi air semakin meningkat. Selain itu, sistem penanaman dalam *greenhouse* perlu memperhatikan kondisi *temperature* dan kelembapan ruangan. Oleh karena itu, dirancang sebuah sistem *monitoring* keadaan *greenhouse* dan pembacaan volume air pada *water inlet* berbasis Arduino. *Monitoring* dapat dipantau melalui sebuah aplikasi Android yang terhubung ke Google Firebase dan Thingspeak. Cara kerja dari sistem ini adalah Arduino *slave* yang diletakkan di tengah *greenhouse* mengirimkan data DHT22 (*temperature* dan kelembapan udara) ke Arduino *master* (ruang operator) menggunakan modul *serial* MAX485 dengan kabel 15 meter. Lalu, Arduino *master* mengirimkan data *temperature* dan kelembapan udara ke *database* setiap 20 menit. *Flow sensor* terhubung dengan Arduino *master* untuk membaca volume air setiap terjadi pengisian tangki dengan mengirimkan total volume air yang terbaca ke *database*. Pemantauan kondisi *greenhouse* dapat dilihat langsung ke ruang *control panel* maupun menggunakan aplikasi Android. Pada aplikasi terdapat sebuah notifikasi untuk menginformasikan ke pemilik kebun apabila kondisi *greenhouse* terlalu panas maupun kelembapan udara yang terlalu kering.

Kata kunci: *monitoring; arduino; flow sensor; dht22; android; hidroponik; greenhouse; temperature; kelembapan udara*



DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PENGESAHAN PENGUJI	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Studi Literatur	4
2.2 Tinjauan Teori.....	5
2.2.1 Arduino Mega 2560.....	5
2.2.2 DHT22	6
2.2.3 Modul GSM SIM800C	6
2.2.4 Flow Sensor	7
2.2.5 Solenoid Valve.....	8
2.2.6 Modul Serial MAX 485	8
2.2.7 Android	8

2.2.8 Regresi Linier	9
BAB 3 METODOLOGI	10
3.1 Desain Sistem	10
3.2 Perancangan Sistem	11
3.3 Cara Kerja Sistem	11
3.4 Gambaran <i>Prototype</i> Sistem <i>Monitoring</i>	13
3.5 Gambaran Antarmuka Android	14
3.6 Pengujian Sistem.....	15
BAB 4 HASIL DAN PEMBAHASAN.....	19
4.1 Kalibrasi Sensor.....	19
4.2 Pengujian Komunikasi RS485 Menggunakan Modul Serial MAX485	23
4.3 Menguji <i>Monitoring</i> Sistem Secara Keseluruhan.....	24
4.3.1 Pengujian <i>Monitoring Temperature</i> dan Kelembapan Udara.....	24
4.3.2 Pengujian <i>Monitoring Volume Air</i>	27
4.3.3 Pengujian Fungsi Aplikasi Android.....	28
BAB 5 KESIMPULAN DAN SARAN.....	29
5.1 Kesimpulan	29
5.2 Saran	30
DAFTAR PUSTAKA	31
LAMPIRAN	1

DAFTAR GAMBAR

Gambar 2.1 Mikrokontroler Arduino Mega2560.....	6
Gambar 2.2 Modul DHT22	6
Gambar 2.3 Modul GSM SIM800C	7
Gambar 2.4 <i>Flow Sensor</i> SEN-HZ21W1.....	7
Gambar 2.5 <i>Solenoid Valve</i> 24VDC.....	8
Gambar 2.6 Modul <i>serial</i> MAX 485	8
Gambar 2.7 Grafik garis regresi	9
Gambar 3.1. Diagram Sistem <i>Monitoring</i>	10
Gambar 3.2 Gambaran <i>green house</i> PT. Indmira.....	11
Gambar 3.3 <i>Flowchart</i> sistem <i>monitoring</i> Arduino (<i>Master</i>) bagian <i>setup</i>	11
Gambar 3.4 <i>Flowchart</i> sistem <i>monitoring</i> Arduino (<i>Master</i>) bagian <i>void loop</i>	12
Gambar 3.5 <i>Prototype</i> Arduino <i>Master</i>	13
Gambar 3.6 <i>Prototype</i> Arduino <i>Slave</i>	13
Gambar 3.7 Antarmuka aplikasi Android	14
Gambar 3.8 Hasil regresi data <i>temperature</i> DHT22	16
Gambar 3.9 Hasil regresi data kelembapan DHT22.....	17
Gambar 3.10 Data hasil regresi <i>flow sensor</i> data volume air dan gelas ukur.....	18
Gambar 4.1 Kalibrasi DHT22 dengan HTC V.02 <i>environment sensor</i>	19
Gambar 4.2 Kalibrasi <i>flow sensor</i> dengan gelas ukur	19
Gambar 4.3 Grafik data <i>temperature</i> dan kelembapan udara sebelum kalibrasi	20
Gambar 4.4 Grafik data <i>temperature</i> dan kelembapan udara sesudah kalibrasi	21
Gambar 4.5 Grafik hasil pengiriman data <i>temperature</i> ke <i>database</i>	25
Gambar 4.6 Grafik hasil perekaman data <i>temperature</i> pada SDCard.....	25
Gambar 4.7 Grafik hasil pengiriman data kelembapan ke <i>database</i>	26
Gambar 4.8 Grafik hasil perekaman data Kelembapan udara pada SDCard	26
Gambar 4.9 Hasil pengujian <i>monitoring volume</i> air pengisian setengah tangki.....	27
Gambar 4.10 Pengujian fungsi aplikasi Android	28

DAFTAR TABEL

Tabel 3.1 Kriteria Status Nilai Sensor di Aplikasi Android.....	14
Tabel 3.2 Data pengujian DHT22 terhadap HTC <i>environmet</i> sensor (sampel dari Lampiran 1)..	15
Tabel 4.1 Hasil pembandingan antara sensor dengan alat ukur sebelum kalibrasi	20
Tabel 4.2 Hasil pembandingan antara sensor dengan alat ukur sesudah kalibrasi.....	21
Tabel 4.3 Hasil terbaca komunikasi serial menggunakan MAX485.....	23
Tabel 4.4 Nilai volume air pada data penyimpanan <i>database</i> dan SDCard.....	27



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Kemajuan teknologi yang semakin pesat menjadikan kehidupan kita seakan tidak bisa lepas dari teknologi. Konsep tatanan kota cerdas salah satunya dipengaruhi oleh tingginya pengguna teknologi yang belakangan ini digemari diberbagai kota besar diseluruh dunia termasuk Indonesia. Mulai tahun 2010, menurut data penduduk yang diinformasikan oleh BPS (Badan Pusat Statistik) telah terjadi peningkatan persentase jumlah penduduk perkotaan dari 49,8% hingga mencapai 56,7% pada 2020 [1]. Tatanan kota cerdas (*smart city*) tidak akan sukses apabila tidak didukung dengan pengembangan konsep sistem yang cerdas (*smart system*). Pertanian perkotaan merupakan sebuah konsep pertanian lahan sempit yang sesuai untuk diterapkan di daerah perkotaan. Salah satu jenis dari konsep pertanian perkotaan adalah hidroponik.

Hidroponik adalah salah satu teknik menanam tanpa menggunakan tanah melainkan memanfaatkan air sebagai media tanamnya. Tanaman akan ditanam pada sebuah pipa dengan lubang tanam yang banyak sehingga tidak membutuhkan lahan yang besar. Beberapa hal yang perlu diperhatikan dalam budidaya tanaman hidroponik salah satunya adalah kandungan nutrisi yang diberikan ke tanaman. Kadar nutrisi digunakan untuk menggantikan unsur hara yang ada pada tanah. Selain itu, sistem hidroponik memerlukan pemantauan secara berkala dengan mengecek ke lokasi *greenhouse*. Karena, pada saat tanaman semakin besar jumlah konsumsi nutrisi semakin banyak. Sehingga, air nutrisi akan lebih cepat habis. Untuk ukuran tangki penampungan air 100L dapat habis dalam 3 hari. Oleh karena itu, perlu dirancang sebuah sistem untuk memantau keadaan *greenhouse* agar pada saat ditinggal ke luar kota, kondisi *greenhouse* akan tetap berjalan dan dapat dipantau dari jarak jauh.

Pada penelitian ini, implementasi sistem dilakukan di PT Indmira yang terletak di jl. Kaliurang Km.16, Kabupaten Sleman. Instalasi hidoponik berada dalam *greenhouse* dengan ukuran 12m x 9m x 3,5m. Sistem minimum yang digunakan adalah modul mikrokontroler Arduino. Penggunaan modul ini karena Arduino mudah dioperasikan dan implementasinya sudah banyak dipakai. Sistem yang akan diterapkan untuk *monitoring greenhouse* adalah memantau *temperature* dan kelembapan udara menggunakan sensor DHT22 yang terpasang di tengah *greenhouse* dengan jarak 15meter dari ruang *operator* dan pembacaan volume air pada *water inlet* untuk media tanam meja A. Android studio digunakan untuk merancang dan membuat aplikasi android sebagai media pemantauan *greenhouse* menggunakan *smartphone*. Sistem komunikasi

yang dipakai agar Arduino dapat berkomunikasi dengan aplikasi Android diperlukan modul GSM SIM800C dan *web server* 000webhost.com yang terhubung ke Google Firebase dan Thingspeak.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem *monitoring temperature*, kelembapan udara, dan volume air pada sistem hidroponik dalam *greenhouse*?
2. Bagaimana merancang sistem komunikasi antara Arduino dan Android?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data yang di pantau pada sistem *monitoring* adalah *temperature*, kelembapan udara, dan volume air.
2. Perangkat keras yang digunakan pada sistem ini adalah mikrokontroller Arduino Mega2560, Arduino Pro mini, modul serial Max RS485, *Flow Sensor*, DHT22, dan Modul GSM Sim800C.

1.4 Tujuan Penelitian

1. Merancang sistem *monitoring temperature* dan kelembapan udara pada *greenhouse* berbasis Arduino yang dapat memberikan informasi ke Android secara daring.
2. Merancang sistem *monitoring* pada *water inlet* agar dapat memantau jumlah penggunaan air yang masuk ke tangki dalam setiap pengisian.
3. Mendapatkan datalog pembacaan sensor DHT22 untuk melihat pengaruh kondisi *greenhouse* terhadap tanaman dengan berdasarkan data *temperature* dan kelembapan udara.

1.5 Manfaat Penelitian

Manfaat penelitian yang diharapkan adalah:

1. Mendapatkan desain *monitoring* hidroponik yang sesuai untuk memantau keadaan atau kondisi *greenhouse* jarak jauh.
2. Mengetahui kondisi *temperature* dan kelembapan udara pada *greenhouse* sehingga dapat memberikan perlakuan khusus pada tanaman ketika *temperature* terlalu tinggi maupun kelembapan udara yang terlalu kering.
3. Mendapatkan data jumlah penggunaan air dalam periode tertentu dengan memasang *flow sensor* pada jalur air ke tangki meja A.



BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Sebagai bahan referensi dalam menyelesaikan penelitian ini. Penulis melakukan studi literatur terkait hidroponik dan *monitoring* jarak jauh. Sudah cukup banyak penelitian lain yang membahas topik ini baik dari sisi pertanian, sistem otomasi, bahkan dampak lingkungan. Penelitian kali ini berfokus pada pemantauan kondisi *greenhouse* dengan memasang sensor dan mengirimkan data tersebut ke *database* agar dapat dipantau melalui *smartphone*.

Beberapa sistem *monitoring* telah diteliti, salah satunya adalah sistem automasi hidroponik untuk penanaman dalam ruangan yang dilakukan oleh Vaibhav Palande, Adam Zaheer, dan Kiran George [2]. Penelitian yang dilakukan menggunakan varian mikrokontroler, sensor, dan teknologi IoT (*Internet of Things*) untuk *monitoring* dan pengendalian. Sistem pemantauan yang mereka buat menggunakan antarmuka *web* maupun aplikasi ponsel. Melalui aplikasi tersebut pengguna dapat memonitor parameter sensor dan mendapatkan notifikasi bila terdapat kesalahan pembacaan parameter sensor.

Penelitian oleh Gigih Pamungkas, Ahmad Zatnika Purwalaksana, Mitra Djamal, Nina Siti Amina [3] membuat sebuah hidroponik sistem *Nutrient Film Technique* otomatis berbasis Arduino. Cara kerja dari sistem ini adalah sistem akan bekerja saat sensor ultrasonic membaca ketinggian air nutrisi di wadah *water pump* cukup, kemudian pengisian nutrisi akan berhenti. Nutrisi yang berada di wadah penampungan akan dialirkan ke pipa hidroponik menggunakan *water pump*. Saat nutrisi mulai habis sistem akan mengisi ulang wadah penampungan dengan nutrisi *AB Mix*.

D.Anandhavalli, K.S. Sangeetha, V. Priya Dharshini, dan B. Lukshana Fathima [4] melakukan penelitian tentang *smart meter* untuk pemanfaatan air secara IoT. Mikrokontroler yang digunakan adalah NodeMCU, dan dua buah *flow sensor* untuk pengukuran air yang masuk ke suatu tangki penampungan. Cara kerja dari sistem ini adalah ketika air mengalir melewati *flow sensor*, rotor pada *flow sensor* akan berputar mengikuti laju aliran air. Kecepatan berubah sesuai laju aliran air dan menghasilkan keluaran berupa sinyal pulsa yang terbaca oleh NodeMCU. Kemudian, nilai pulsa yang terbaca dikonversi ke satuan liter/menit menggunakan rumus konversi. Hasil dari pembacaan *flow sensor* dikirimkan ke penyimpanan internet menggunakan jaringan *Wi-Fi* agar pengguna dapat memantau nilai laju aliran dan dapat ditampilkan dalam bentuk grafik.

Penelitian tentang kendali *temperature* dan kelembapan udara dalam *greenhouse* daerah gurun dilakukan oleh Shigeki Hiwasawa, Mai Nakatsuka, Kunio Masui, Tsuyoshi Kawanami, dan

Katsuaki Shirai [5]. Sistem kendali dilakukan dengan mengendalikan ventilasi udara, *sprinkler* air dan pelindung radiasi matahari yang diletakkan di atas atap *greenhouse*. *Greenhouse* yang digunakan memiliki ukuran 10m x 30m x 2,5m dengan dinding dan atap kaca. Tujuan dari penelitian ini adalah menganalisis efek kendali terhadap *temperature* dan kelembapan udara dalam *greenhouse* serta dampak konsumsi air tanaman. Hasil yang didapatkan adalah *temperature* di ruma kaca menurun sesuai dengan rasio pelindung matahari yang semakin tebal dan ventilasi udara yang semakin lebar. Kelembapan udara dalam *greenhouse* turun menurun seiring bertambah besar ukuran pelindung matahari dan ventilasi udara.

Pada penelitian ini dilakukan perancangan sistem *monitoring* kondisi *greenhouse* menggunakan Arduino dan Android. *Prototype* sistem *monitoring* bertujuan untuk melengkapi sistem kendali PLC yang sudah ada di PT Indmira dengan mengirimkan data *temperature*, kelembapan udara, dan volume air pengisian tangki hidroponik ke *smartphone* pengguna. Karena, sistem kendali PLC tidak dapat melakukan pembacaan *flow sensor* dan hanya dapat dipantau ke ruang *operator* pada *greenhouse* menggunakan HMI (*Human Machine Interface*). Penelitian ini mengambil beberapa bagian sistem yang telah ada pada penelitian sebelumnya seperti penggunaan *flow sensor* dan *temperature* ruangan. Sistem yang belum lengkap kemudian dijadikan satu dan dibuat saling melengkapi.

2.2 Tinjauan Teori

2.2.1 Arduino Mega 2560

Arduino adalah salah satu mikrokontroler yang dapat digunakan untuk pengendalian sistem elektronik dan dapat dirancang sesuai dengan kebutuhan penggunanya. Terdapat 54 pin digital dengan 15 pin sebagai output PWM dan 16 input analog yang berfungsi untuk menerima sinyal dari komponen analog dengan resolusi ADC 10-bit. Sumber daya yang direkomendasikan untuk Arduino Mega adalah 7-12V namun dapat juga pada skala 6-20V. Terdapat juga pin RX dan TX yang dapat difungsikan untuk komunikasi serial. Arduino Mega menggunakan ATmega2560 yang memiliki memory sebesar 256 KB (8 KB digunakan untuk bootloader) [6].



Gambar 2.1 Mikrokontroler Arduino Mega2560

2.2.2 DHT22

DHT 22 adalah sensor *temperature* dan kelembapan udara [7]. DHT22 menggunakan daya sebesar 3,5 – 5,5 VDC dan dapat mengukur *temperature* dengan rentang nilai sebesar -40°C hingga 80°C dengan tingkat presisi $\pm 0,5^{\circ}\text{C}$ serta dapat mengukur kelembapan 0-100% dengan tingkat presisi $\pm 2-5\%$. Gambar 2.2 menunjukkan bentuk fisik sensor DHT22 (sensor *temperature* dan kelembapan udara).



Gambar 2.2 Modul DHT22

2.2.3 Modul GSM SIM800C

Modul GSM SIM800C adalah perangkat komunikasi data dengan menggunakan jaringan seluler. Modul ini dapat digunakan sebagai pengganti fungsi *handphone* karena dapat digunakan untuk *telephone*, sms, dan koneksi internet. *Supply voltage* modul ini adalah 5V-20V atau 3.4-4.2V jika menggunakan *battery lithium*. Untuk mengakses beberapa konfigurasi GSM modul ini dapat diakses menggunakan ATCommand dari Arduino maupun menggunakan USB to TTL. Untuk jaringan frekuensi yang dapat diakses adalah 850/900/1800/1900 MHz [8]. Gambar 2.3 menunjukkan bentuk fisik modul GSM SIM800C.



Gambar 2.3 Modul GSM SIM800C

2.2.4 Flow Sensor

Flow sensor adalah sensor yang digunakan untuk mengetahui laju aliran fluida [4]. *Flow sensor* biasanya digunakan untuk menghitung laju aliran, atau untuk menghitung volume fluida. Sensor yang digunakan adalah SEN-HZ21W1 yang berbahan *stainless steel* dengan kemampuan ukur skala 1 – 30L/min. *temperature* air yang dapat melalui aliran sensor ini kurang dari 80°C dan *Flow Conversion* 1 L air = 553 pulsa ±10%. Cara kerja sensor ini adalah dengan mengubah nilai pulsa yang terbaca ke satuan laju aliran (L/menit) maupun volume (mililiter atau liter). Rumus konversi pembacaan volume air adalah [4]:

$$P = \text{Pulsa (Hz)} \qquad P = C \times Q \qquad 2.1$$

C = Faktor Kalibrasi

$$Q = \frac{P}{C}$$

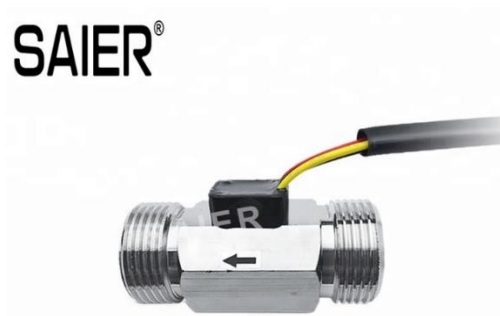
V = Volume Air (L)

$$V = \frac{Q \times T}{60}$$

T = Waktu (detik)

$$V = \frac{P \times T}{C \times 60} \qquad 2.2$$

Volume air didefinisikan dengan jumlah pulsa dibagi faktor kalibrasi. Faktor kalibrasi adalah variabel yang digunakan untuk mengatur akurasi sensor. Dengan menggunakan persamaan 2.1 dan 2.2 dapat diketahui besaran volume air yang masuk dalam satuan liter (L) dari debit air (L/m) dikalikan dengan lama penggunaan. Bentuk fisik *flow sensor* ditunjukkan pada Gambar 2.4.



Gambar 2.4 *Flow Sensor* SEN-HZ21W1

2.2.5 Solenoid Valve

Solenoid valve adalah keran otomatis yang berfungsi untuk membuka dan menutup aliran fluida. *Solenoid valve* bekerja secara otomatis dimana buka atau tutupnya ditentukan oleh tegangan listrik. Input tegangan untuk mengaktifkan *solenoid valve* membutuhkan tegangan 24VDC.



Gambar 2.5 *Solenoid Valve* 24VDC

2.2.6 Modul Serial MAX 485

MAX 485 adalah sebuah modul yang digunakan untuk melakukan komunikasi data serial RS485 dengan jarak yang jauh hanya dengan menggunakan 2 kabel jumper. Komunikasi dua arah menggunakan modul ini memungkinkan komunikasi serial jarak jauh hingga 1.200meter dengan mode *Half Duplex*. *Half duplex* merupakan *mode* komunikasi dua arah dengan cara bergantian dalam melakukan transmisi maupun menerima data [9]. Sumber tegangan untuk mengaktifkan modul ini membutuhkan tegangan 5VDC.



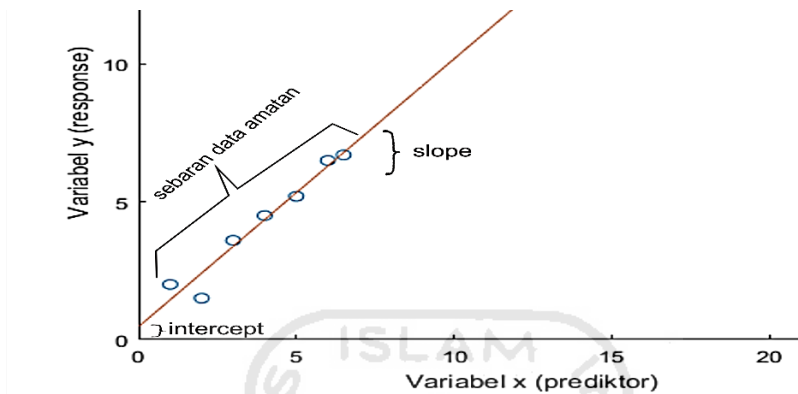
Gambar 2.6 Modul *serial* MAX 485

2.2.7 Android

Android adalah sebuah nama sistem operasi yang dirancang untuk *smartphone*, yaitu sebuah perangkat bergerak layar sentuh. Android dapat digunakan untuk menerima informasi dari peralatan elektronik dan menampilkan informasi tersebut. Selain itu, android juga dapat mengolah data dari perangkat elektronik dan mengendalikannya. Informasi tersebut dapat dikirim melalui perantara *bluetooth* atau internet.

2.2.8 Regresi Linier

Regresi linier adalah metode statistika yang digunakan untuk membentuk model hubungan antara variabel terikat (y) dengan satu atau lebih variabel bebas (x). Dalam metode regresi linier terdapat data yang digunakan sebagai bahan untuk membentuk persamaan regresi. Persamaan untuk regresi linier adalah $y = a + bx$. Dimana: y = variable dependen, a = konstanta, b = koefisien variable x (*slope*), dan x = variable independen. Contoh garis regresi digambarkan pada Gambar 2.7.



Gambar 2.7 Grafik garis regresi

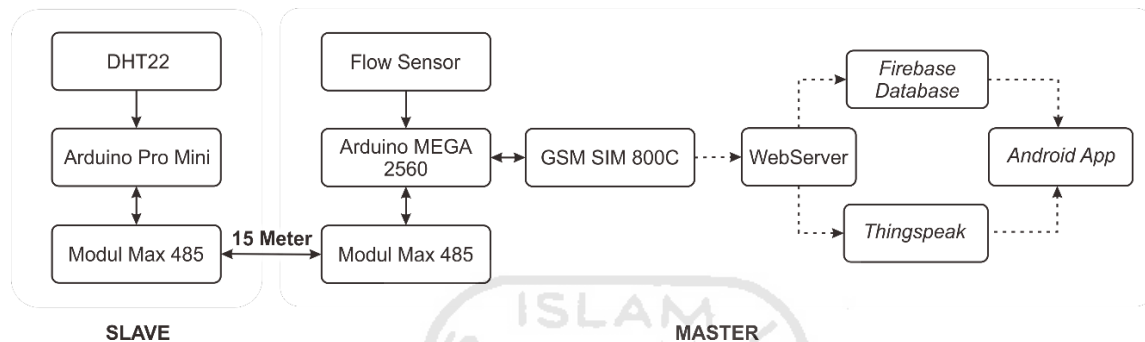


BAB 3

METODOLOGI

3.1 Desain Sistem

Pembuatan sistem *monitoring* memerlukan perancangan dan desain sistem yang baik untuk memudahkan dalam membuat *prototype*. Beberapa komponen yang dibutuhkan untuk membuat *prototype* sistem *monitoring* terdapat pada Gambar 3.1.



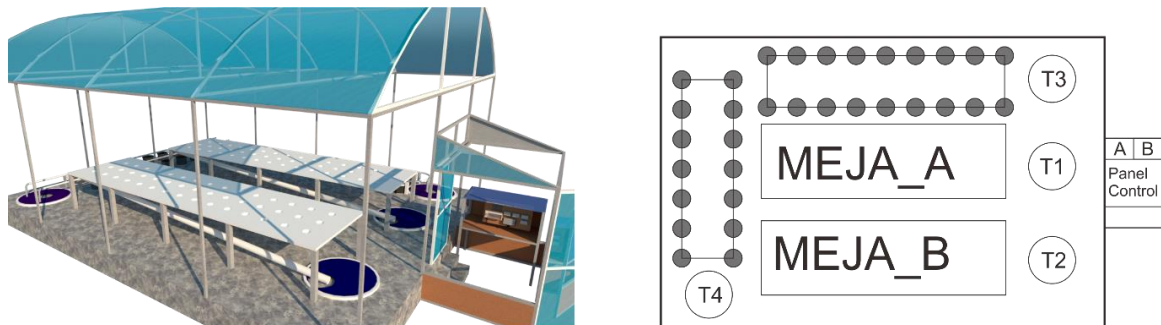
Gambar 3.1. Diagram Sistem *Monitoring*

Pada Gambar 3.1 terlihat bahwa terdapat dua arduino yang saling berkomunikasi menggunakan media perantara MAX485. Modul ini merupakan sebuah komponen tambahan yang digunakan untuk mengirim data DHT22 dari Arduino *Slave* ke Arduino *Master* menggunakan kabel dengan jarak 15 meter. MAX485 memiliki keunggulan dapat mengirimkan data sejauh 1200meter dan dapat mengurangi *noise* pada saat mengirimkan data.

Prinsip kerja dari sistem ini adalah memantau keadaan *greenhouse* dengan mengirimkan data *temperature*, kelembapan udara, dan volume air ke *smartphone* pemilik *greenhouse*. Pada mulanya, Arduino (*slave*) membaca *temperature* dan kelembapan udara. Kemudian, setiap 1 menit Arduino (*slave*) mengirimkan data ke Arduino (*Master*) yang berfungsi sebagai pengirim data ke *database*. Selain itu, Arduino (*Master*) dapat melakukan pembacaan volume air untuk tangki hidroponik Meja A (dapat dilihat pada gambar 3.2) menggunakan *flow sensor*. apabila ketinggian tangki hidroponik Meja A menunjukkan pembacaan air rendah, sistem kendali PLC akan mengaktifkan *Solenoid valve* sehingga katup *water inlet* terbuka. Kemudian, pada saat air melewati *flow sensor*, Arduino (*Master*) akan mendapatkan interupsi perubahan pulsa yang dikonversi dalam satuan mL (mililiter) ataupun satuan liter dan menampilkan pada LCD nilai laju aliran dan total air yang masuk. ketika sudah tidak ada perubahan nilai dari *flow sensor* nilai terakhir dari volume air dikirimkan ke *web server*. Pada saat terjadi perubahan nilai pada *database*, aplikasi android yang terhubung ke jaringan internet akan mendapatkan data terakhir yang dikirim oleh arduino (*master*). Terdapat 2 *database* yang digunakan untuk aplikasi android yaitu Google

Firestore yang digunakan untuk menampilkan data ke aplikasi dalam bentuk teks maupun angka, sedangkan *database* Thingspeak digunakan untuk menampilkan grafik ke aplikasi android.

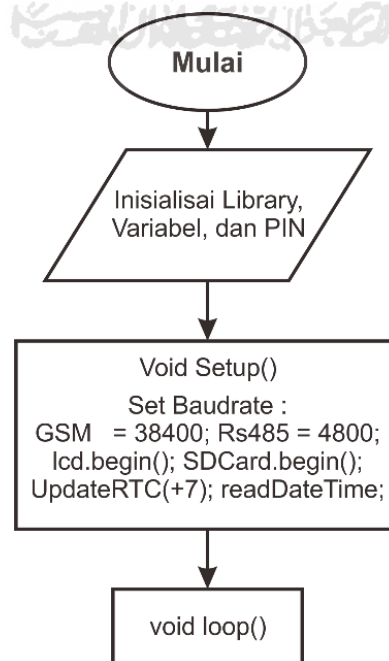
3.2 Perancangan Sistem



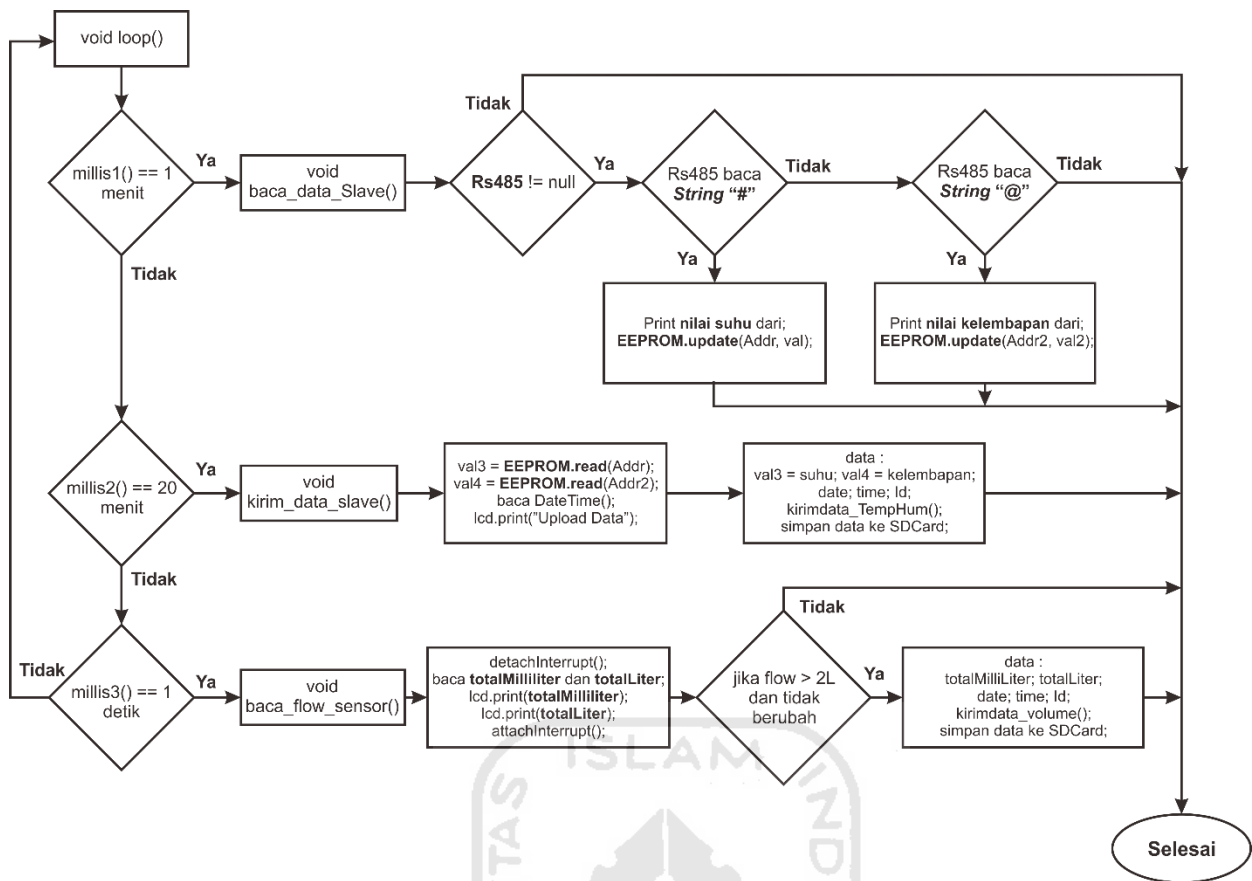
Gambar 3.2 Gambaran *green house* PT. Indmira

Perancangan sistem *monitoring* hidroponik pada *green house* di PT. Indmira digambarkan pada Gambar 3.2. Gambar 3.2 merupakan gambaran lahan yang ada di PT. Indmira dengan ukuran 15m x 9m x 3,5m. Lahan tersebut dimanfaatkan untuk beberapa media tanam yaitu Meja A, Meja B, *drip sistem* dan *duch bucket*. Kemudian, untuk membuat sistem *monitoring* pada *greenhouse* ditambahkan sensor DHT22 yang dipasang di tengah *greenhouse*. Pemasangan *flow sensor* untuk *water inlet* diletakkan pada tangki penampungan Meja A (T1) yang letaknya berdekatan dengan *panel control* menggunakan kabel sepanjang 6 meter.

3.3 Cara Kerja Sistem



Gambar 3.3 *Flowchart* sistem *monitoring* Arduino (*Master*) bagian *setup*



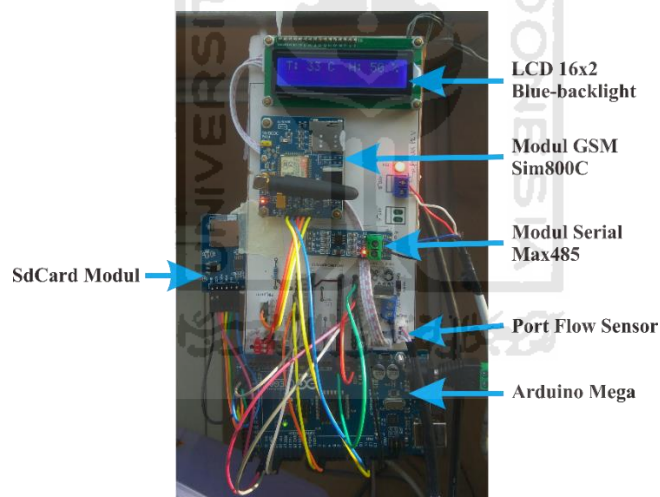
Gambar 3.4 Flowchart sistem monitoring Arduino (Master) bagian void loop

Sistem kerja sistem monitoring secara singkat dijelaskan dalam diagram alir ditunjukkan pada Gambar 3.3 dan 3.4. Cara kerja sistem monitoring dimulai dari memberikan input tegangan ke sistem. Beberapa inisialisasi library dan variabel program ada di tahap sebelum void setup (Gambar 3.3). Pada bagian setup terdapat program untuk menjalankan fungsi GSM, RS485, LCD, SDCard, dan updateRTC agar sesuai waktu di Indonesia (WIB). Kemudian masuk ke bagian fungsi loop yang terdiri dari 3 bagian fungsi millis. Millis adalah suatu fungsi yang berguna untuk menjalankan perhitungan waktu secara independent setiap milliseconds. Pada sistem ini fungsi millis digunakan sebagai fungsi interrupt untuk tiap bagian program dengan perhitungan waktu yang telah ditentukan. Pada bagian pertama terdapat fungsi millis yang berfungsi untuk melakukan eksekusi program untuk membaca data dari Arduino slave tiap 1 menit. Pada fungsi millis1() terdapat program pembacaan data DHT22 yang mana apabila Arduino master membaca data serial dari MAX485 data tersebut akan di parseInt (mengembalikan tipe data string ke integer) untuk mendapatkan data yang dikirim dari Arduino slave. Datanya adalah apabila terbaca String "C" setelahnya merupakan data temperature sehingga datanya di parsing dari string ke integer kemudian disimpan ke EEPROM(Addr) lalu ditampilkan ke LCD. Selanjutnya, jika terbaca String "D" data yang dikirim (slave) setelahnya adalah kelembapan udara kemudian disimpan ke EEPROM(Addr2) dan ditampilkan ke LCD.

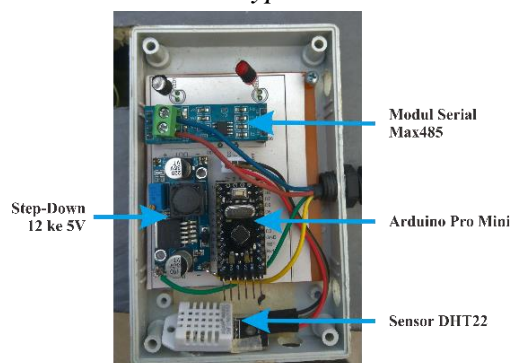
Pada bagian kedua terdapat fungsi *millis2()* yang digunakan untuk menjalankan program pengiriman data *temperature* dan kelembapan udara setiap 20 menit. Arduino akan mengirimkan data *temperature* dan kelembapan yang tersimpan di EEPROM ke *web server* dengan urutan data (Id, nilai *temperature*, nilai kelembapan, tanggal, waktu).

Program setelahnya adalah fungsi *millis3()* untuk pembacaan *Flow Sensor* SEN-HZ21W. cara kerja sensor ini cukup sederhana, prinsip dasarnya adalah perubahan pulsa PWM. Dalam program arduino terdapat perhitungan yang dipakai untuk mengkonversi nilai PWM ke laju aliran dengan satuan L/menit menggunakan library *flow sensor*. Setelah itu data laju aliran dan total air ditampilkan ke LCD, jika total pembacaan air sudah terhenti dan total air melebihi 2 liter maka arduino akan mengirimkan beberapa data dengan urutan (Id, total air (mL), total air (L), tanggal, dan waktu) ke *web server* yang terhubung ke Google Firebase dan Thingspeak menggunakan modul GSM Sim800C. Disamping itu, setiap ada proses pengiriman data ke *web server* data juga disimpan ke kartu memori sebagai data cadangan.

3.4 Gambaran *Prototype Sistem Monitoring*



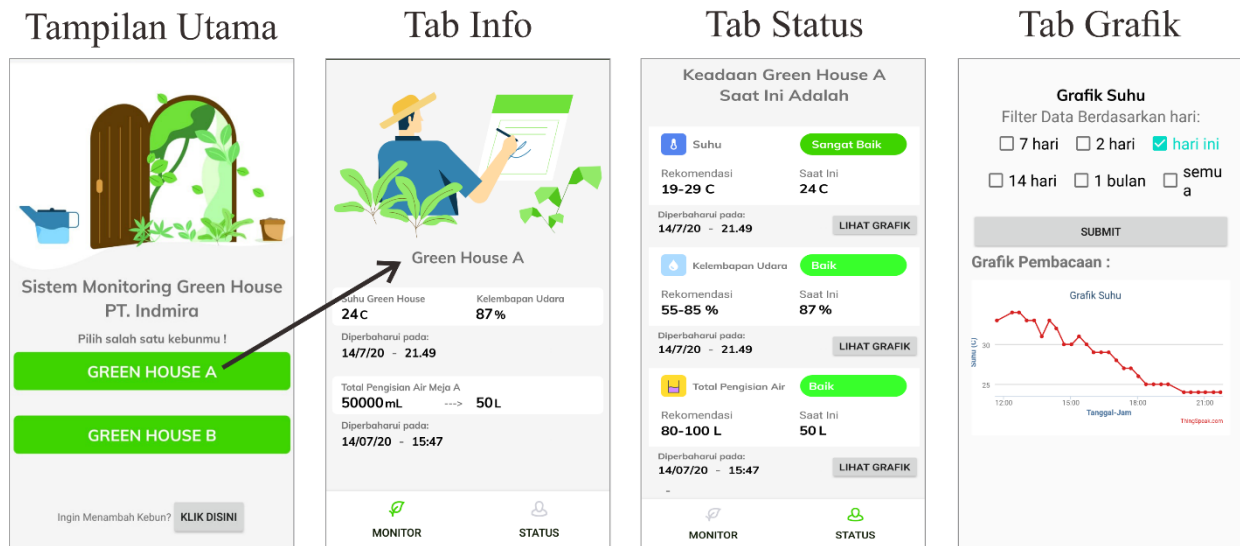
Gambar 3.5 *Prototype Arduino Master*



Gambar 3.6 *Prototype Arduino Slave*

Prototype sistem monitoring memiliki 2 perangkat keras. Perangkat keras pertama digunakan sebagai Master (Gambar 3.5) dan perangkat kedua sebagai slave (Gambar 3.6).

3.5 Gambaran Antarmuka Android



Gambar 3.7 Antarmuka aplikasi Android

Tampilan utama merupakan halaman awal saat aplikasi dibuka. *Flow chart* aplikasi secara lengkap terdapat pada lampiran 11. Pada tampilan utama terdapat beberapa pilihan menu yaitu *green house A*, *green house B*, dan tombol untuk menambahkan kebun (bag. bawah) jika pengguna nantinya akan menambahkan titik pemantauan. Pada saat halaman *greenhouse A* dibuka maka akan muncul halaman yang berisikan 2 *tab* yaitu: *tab info* dan *tab status*. Yang pertama adalah *tab info* yang berisi tampilan nilai-nilai dari sensor beserta tanggal dan waktu data dikirim. Pada bagian kedua ada *tab status* dengan animasi apabila nilai pembacaan sensor berada pada nilai rekomendasi maka status akan berubah sesuai kriteria pada tabel 3.1.

Tabel 3.1 Kriteria Status Nilai Sensor di Aplikasi Android

Nilai <i>temperature</i> (°C)	Status	Nilai Kelembapan (%)	Status	Nilai Volume Air (L)	Status
< 19	Dingin	< 55	Sangat Kering	<= 25	Sedikit
19 - 29	Sangat Baik	55 - 85	Sangat Baik	26 - 50	Cukup
30 - 35	Panas	86 - 95	Lembap	50 - 80	Baik
> 35	Sangat Panas	> 95	Sangat Lembap	80 - 105	Penuh
				> 105	Terlalu banyak

Pada *tab status* selain dapat menampilkan status nilai sensor yang terbaca dapat juga menampilkan grafik dengan cara menekan tombol “Lihat Grafik” pada tiap *layout* pembacaan sensor. *Tab grafik* memiliki *checkbox* untuk menampilkan grafik dengan *filter* data berdasarkan hari. Namun, secara *default* menampilkan grafik secara penuh.

3.6 Pengujian Sistem

Pengujian sistem dilakukan setelah selesai melakukan proses perancangan. Pada tahap pengujian terdiri dari beberapa langkah. Pengujian ini bertujuan untuk mengetahui tingkat akurasi dari tiap sensor yang digunakan. Langkah-langkah pengujian untuk penelitian ini yaitu:

1. Mengkalibrasi sensor DHT22

Sensor DHT22 menggunakan sebuah *library* Arduino yang bernama DHT.h. *library* ini sudah banyak digunakan dan dibagikan melalui forum diskusi Github. Cara kerjanya adalah Arduino membaca nilai keluaran DHT22 yang berupa nilai analog 0 sampai 1024 kemudian dikonversi menjadi nilai *temperature* maupun kelembapan udara. Proses kalibrasi meliputi beberapa tahapan yang pertama adalah merekam data *temperature* dan kelembapan DHT22 serta kalibrator HTC *Environmet* Sensor. Hasil data yang telah direkam kemudian diolah menggunakan regresi linier yang bertujuan untuk memperkecil selisih nilai DHT22 dan HTC *Environement* Sensor. Pada proses kalibrasi ini digunakan metode regresi linier sederhana dengan rumus:

$$Y = a + bX \quad 3.1$$

Dimana: Y = variabel terikat (response), a = konstanta, b = koefisien regresi (kemiringan), dan x = variable bebas (prediktor). Nilai a dan b dapat dihitung. Berikut adalah perhitungan sederhana bagaimana nilai a dan b diperoleh.

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \quad 3.2$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad 3.3$$

Tabel 3.2 Data pengujian DHT22 terhadap HTC *environmet* sensor (sampel dari Lampiran 1)

Data ke-	Variabel X - <i>temperature</i> DHT22 (⁰ C)	Variabel Y - <i>temperature</i> HTC (⁰ C)	x ²	y ²	xy
1	26.4	26.4	696.96	696.96	696.96
2	24.9	25.7	620.01	660.49	639.93
3	25	25.2	625	635.04	630
4	25.1	25.1	630.01	630.01	630.01
5	25.2	25.3	635.04	640.09	637.56
6	25.8	25.8	665.64	665.64	665.64
7	27	27.3	729	745.29	737.1
	$\sum x = 179.4$	$\sum y = 180.8$	$\sum x^2 = 4601.66$	$\sum y^2 = 4673.52$	$\sum xy = 4637.2$

Melalui persamaan 3.2 nilai a dapat dicari sebagai berikut,

$$a = \frac{(180.8)(4601.66) - (179.4)(4637.2)}{7(4601.66) - (179.4)^2}$$
$$a = \frac{66.448}{27.26}$$
$$a = 2.4375$$

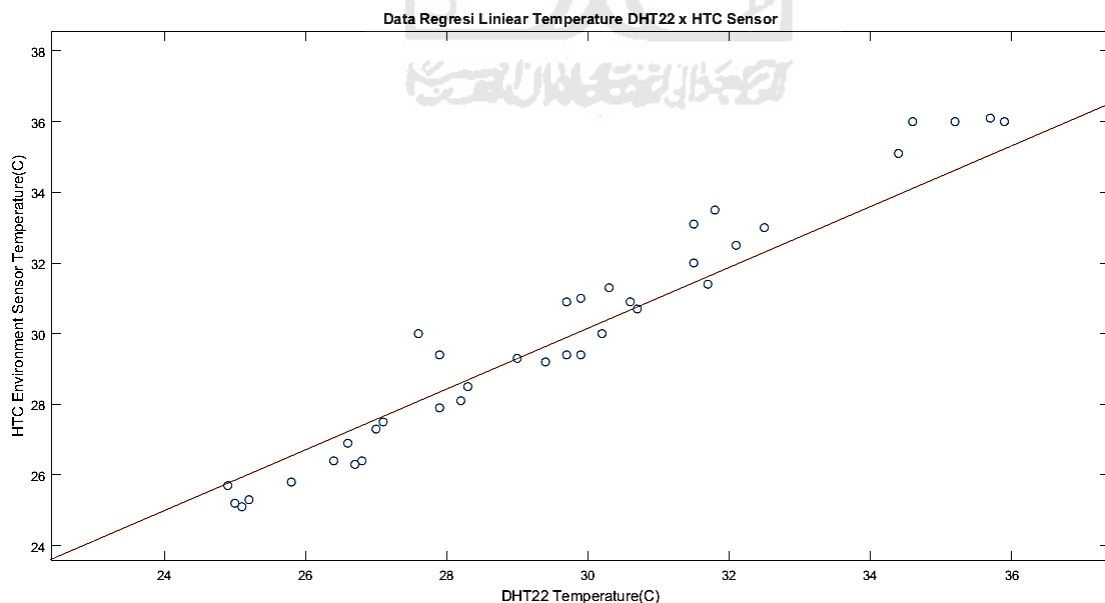
Untuk mencari nilai b digunakan persamaan 3.3 sebagai berikut

$$b = \frac{7(4637.2) - (179.4)(180.8)}{7(4601.66) - (179.4)^2}$$
$$a = \frac{24.88}{27.26}$$
$$a = 0.9126$$

Maka, diperoleh model persamaan regresi dari perhitungan di atas sebagai berikut,

$$Y = 2.4375 + 0.9126X$$

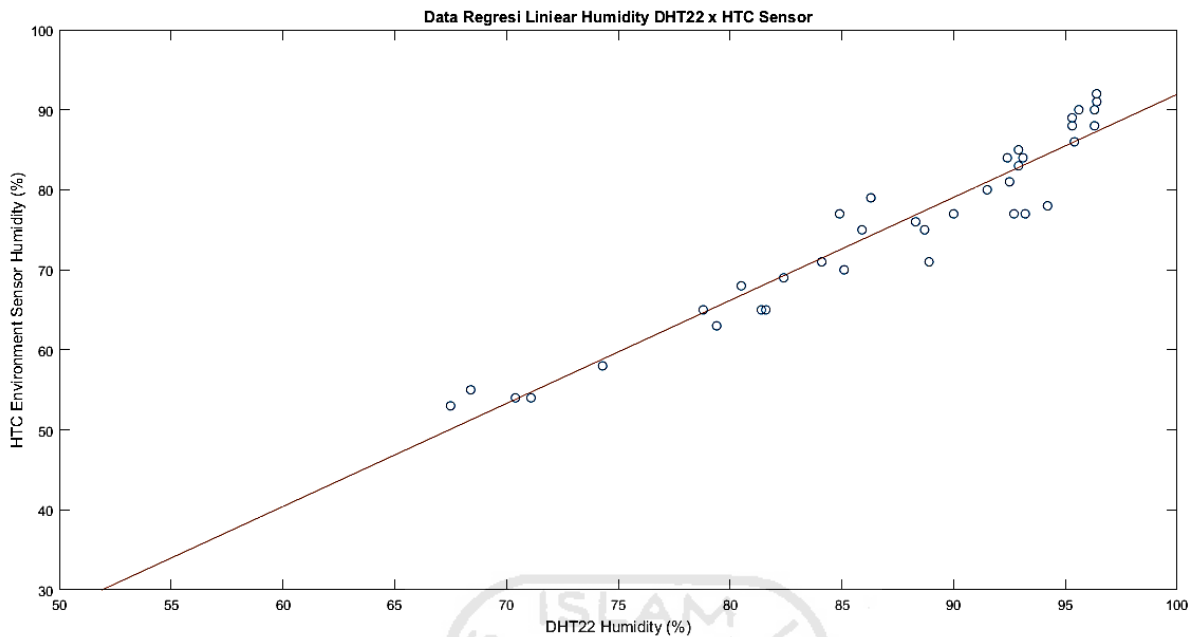
Hasil perhitungan tersebut hanya sebagai contoh bagaimana mendapatkan model persamaan regresi linier untuk *temperature* DHT22. Secara lengkapnya grafik hasil regresi data *temperature* ditunjukkan pada gambar 3.7, sedangkan grafik hasil regresi data kelembapan udara ditunjukkan pada Gambar 3.8. Seluruh data perekaman terdapat di lampiran 1 dan 2.



Gambar 3.8 Hasil regresi data *temperature* DHT22

Persamaan hasil regresi untuk data *temperature* DHT22 adalah:

$$Y = (-11.7535) + (9.09834)X \quad 3.4$$



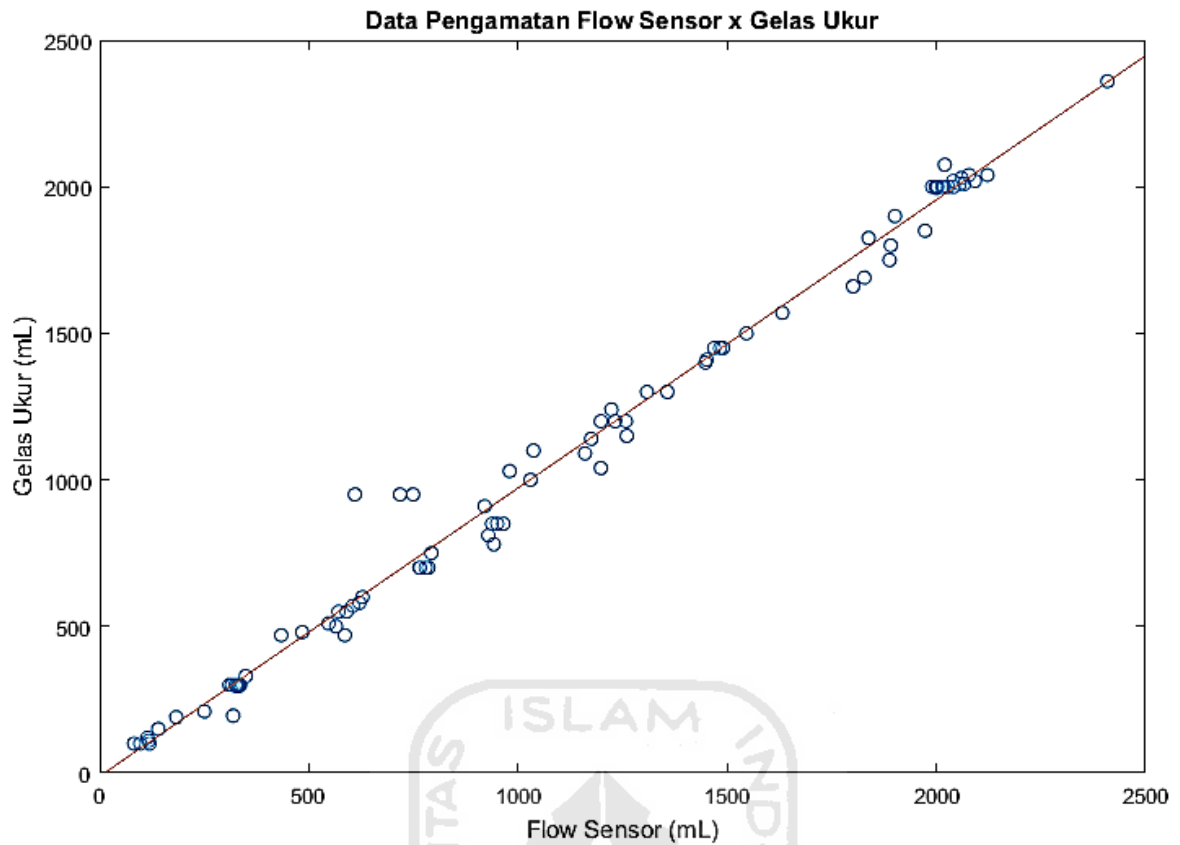
Gambar 3.9 Hasil regresi data kelembapan DHT22

Persamaan untuk hasil regresi data kelembapan udara adalah:

$$Y = (-36.8527) + (1.2879)X \quad 3.5$$

2. Mengkalibrasi *flow sensor* SEN-HZ21W1

Pada tahap ini dilakukan kalibrasi sensor dengan menggunakan 3 alat yaitu : *Solenoid valve*, *flow sensor*, dan gelas ukur takaran 2 Liter. *Solenoid valve* dikendalikan melalui PLC dengan *timer* agar air yang keluar tidak melebihi takaran gelas ukur. Kemudian, *flow sensor* akan membaca jumlah air yang masuk dan dibandingkan dengan air yang masuk ke dalam gelas ukur. Hasil perekaman data *flow sensor* dan gelas ukur dibandingkan menggunakan regresi linier, tujuannya untuk mendapatkan persamaan matematis yang dapat memperkecil nilai *error* pembacaan *flow sensor*. Data kalibrasi terdapat pada lampiran 4 dan 5. Grafik hasil regresi data *flow sensor* dan gelas ukur ditunjukkan pada Gambar 3.10.



Gambar 3.10 Data hasil regresi *flow sensor* data volume air dan gelas ukur

Persamaan untuk data hasil regresi data *flow sensor* adalah:

$$Y = (-11.7535) + (0.9834)X \quad 3.6$$

3. Menguji komunikasi RS485 Arduino *Master* dan Arduino *Slave*.

Pengujian komunikasi RS485 antar Arduino dilakukan dengan menambahkan modul Max485 dengan kabel telepon sepanjang 15 meter. Jalur yang digunakan hanya 2 kabel (yaitu kabel A dan kabel B). Pengujian dilakukan dengan cara manual yaitu dengan mencatat data terkirim dan data yang diterima melalui *serial monitor* oleh kedua perangkat serta mencatat durasi yang dibutuhkan untuk tiap pengiriman data.

4. Menguji sistem *monitoring* secara keseluruhan

Pengujian ini berupa pengecekan secara keseluruhan sistem mulai dari pembacaan sensor hingga data terkirim sampai ke pengguna aplikasi *monitoring*.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Kalibrasi Sensor

Sensor-sensor yang digunakan pada sistem *monitoring greenhouse* telah dikalibrasi menggunakan alat ukur yang sesuai parameter lingkungan. Kalibrasi sensor dilakukan dengan cara memperkecil selisih pembacaan sensor menggunakan metode regresi linier. Sensor DHT22 dikalibrasi dengan alat ukur HTC V.02 *environment sensor* dan *flow meter* SEN-HZ21W1 dikalibrasi dengan gelas ukur. Parameter yang terukur pada DHT22 merupakan hasil pembacaan sensor secara langsung mulai dari jam 6.00 – 15.00. Sedangkan parameter volume air diperoleh dengan cara memberikan *timer* pada *relay* untuk mengaktifkan *solenoid valve* dalam periode waktu tertentu. Gambar 4.1 dan 4.3 menunjukkan kegiatan pengambilan data untuk kalibrasi. Dari kegiatan kalibrasi tersebut didapatkan *error* pada masing-masing sensor seperti pada Tabel 4.1. kemudian dilakukan kalibrasi sensor dan diperoleh hasil yang lebih baik yang ditunjukkan pada Tabel 4.2. Data kalibrasi terdapat pada lampiran 1-6.

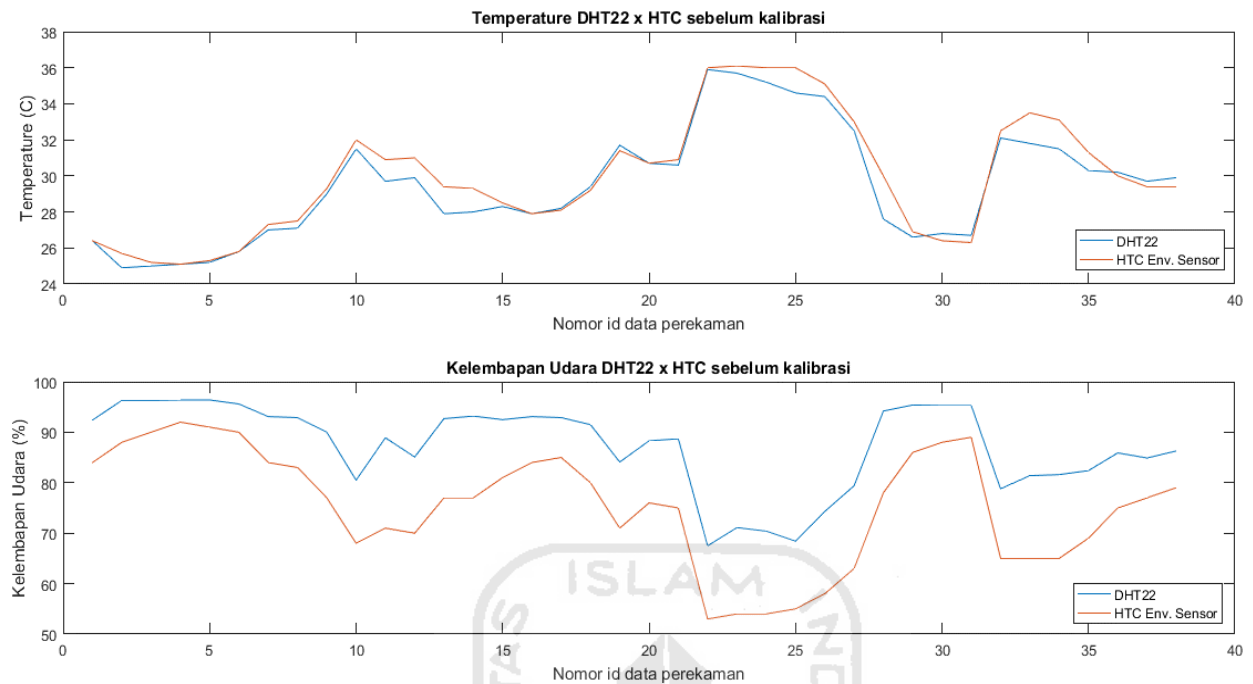


Gambar 4.1 Kalibrasi DHT22 dengan HTC V.02 *environment sensor*



Gambar 4.2 Kalibrasi *flow sensor* dengan gelas ukur

Setelah melakukan proses perekaman data DHT22 dan HTC *environment sensor* dihasilkan data *temperature* dan kelembapan udara yang ditampilkan dalam bentuk grafik seperti pada Gambar 4.3. untuk melihat data secara utuh dapat dilihat pada Lampiran 1 s.d 4.

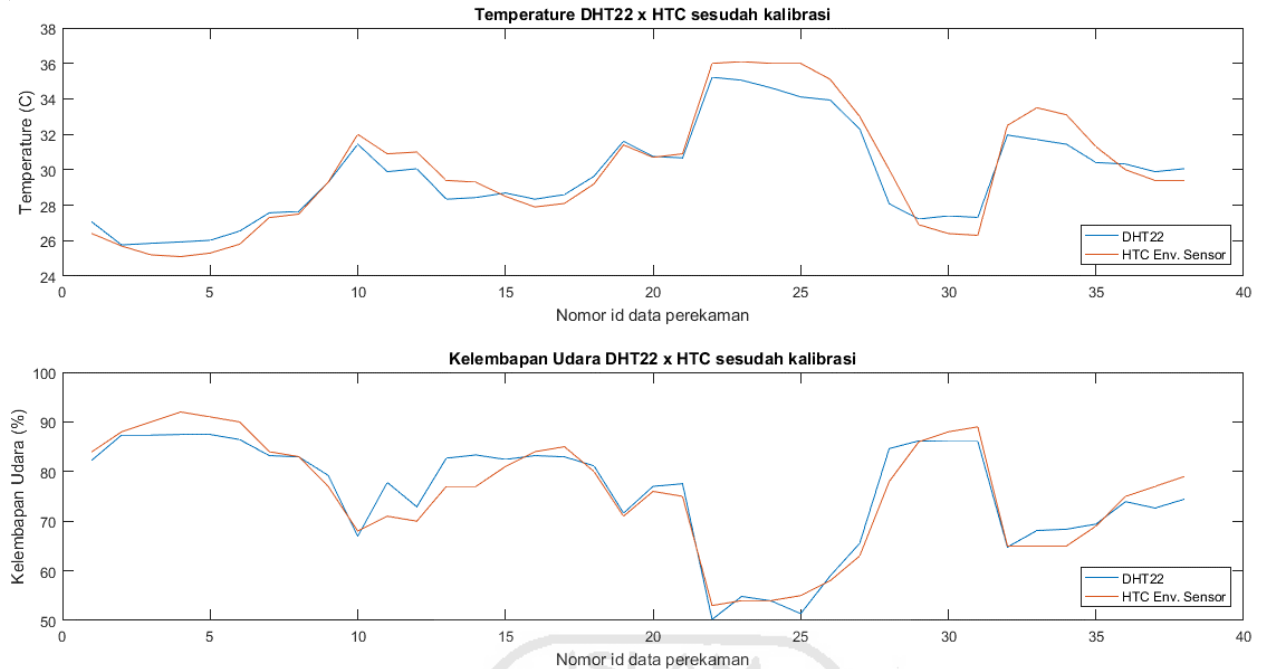


Gambar 4.3 Grafik data *temperature* dan kelembapan udara sebelum kalibrasi. Data yang tertampil pada grafik (Gambar 4.3) terdiri dari 5 tahapan pengujian yang dilakukan pada jam berbeda, namun ditampilkan dalam grafik dengan sumbu x dalam satuan urutan data terekam. Nilai rata-rata *error* pembacaan tiap pengujian dapat dilihat pada Tabel 4.1

Tabel 4.1 Hasil pembandingan antara sensor dengan alat ukur sebelum kalibrasi

Alat Ukur	Parameter	Rata-rata selisih pengukuran / uji coba					Rata-rata total error
		Error 1	Error 2	Error 3	Error 4	Error 5	
DHT22	Temperature (°C)	0.2	0.9	0.15	0.9	0.68	0.566
DHT22	Kelembapan udara (%)	6.8	14.3	11.3	15.7	10.93	11.806
Flow Sensor	Volume air (mL)	14.8	35.7	42.3	92.2	61.6	49.32

Kemudian, setelah dilakukan kalibrasi sensor menggunakan metode regresi linier diperoleh hasil pembacaan yang ditampilkan dalam bentuk grafik pada Gambar 4.4.



Gambar 4.4 Grafik data *temperature* dan kelembapan udara sesudah kalibrasi. Data yang tertampil pada grafik (Gambar 4.4) merupakan data yang telah dilakukan kalibrasi menggunakan metode regresi linier. Nilai selisih pembacaan rata-rata tiap pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil perbandingan antara sensor dengan alat ukur sesudah kalibrasi

Alat Ukur	Parameter	Rata-rata selisih pengukuran / uji coba					Rata-rata total error
		Error 1	Error 2	Error 3	Error 4	Error 5	
DHT22	Temperature (°C)	0.56	0.65	0.29	1.27	0.86	0.72
DHT22	Kelembapan udara (%)	2.49	3.57	1.37	2.49	2.22	2.43
Flow Sensor	Volume air (mL)	15.8	19.8	27.9	65.7	42.2	34.28

Error merupakan selisih nilai hasil pembacaan sensor dan pembandingnya (alat ukur). Pada Tabel 4.1 dan 4.2 terdapat nilai *error* 1 s.d. 5, nilai tersebut merupakan rata – rata selisih pengukuran tiap uji coba yang dilakukan pada jam berbeda. Contohnya pada pembacaan *temperature*, Tabel 4.1 memiliki rata-rata *error* sebesar 0.2 °C sebagai pengujian pertama (tiap pengujian terdapat ±7 data). Kemudian, dilakukan pengujian berikutnya pada jam berbeda dengan hasil rata-rata *error* 0.9 °C untuk pengujian ke-2. Dari total 5 pengujian *temperature* diperoleh rata-rata total *error* sebesar 0.56 °C untuk pembacaan *temperature*. Pembacaan *temperature* DHT22 tidak dilakukan kalibrasi karena nilai *error* dibawah 1 °C yaitu 0.56 °C dibanding pada saat dilakukan kalibrasi nilai *error* yang diperoleh menjadi lebih tinggi yaitu 0.76 °C (Tabel 4.2). Data yang lengkap dapat dilihat pada Lampiran 1 dan 2. Selain itu, pada Tabel 4.1 diperoleh rata-

rata total *error* pembacaan kelembapan udara sebesar 11.8 %. Nilai pembacaan kelembapan udara memiliki *error* cukup besar karena lebih dari 10 % dengan nilai kalibrator. Sehingga, pembacaan kelembapan udara perlu dilakukan kalibrasi agar hasilnya sesuai dengan nilai kalibrator. Setelah dilakukan kalibrasi rata-rata total *error* untuk pembacaan kelembapan udara menjadi 2,43 % (Tabel 4.2) dan hasilnya lebih mendekati nilai pembacaan pada kalibrator dibanding sebelum dilakukan kalibrasi menggunakan regresi linier.

Pada pembacaan volume air nilai *error* 1 s.d. 5 adalah rata-rata selisih pembacaan dalam tiap pengujian. Dimana, pembacaan volume air dilakukan 10 pengulangan dalam tiap pengujian. Perbedaan dalam tiap pengujian terletak pada durasi/waktu *solenoid valve* menyala. Contohnya, pada Tabel 4.1 pengujian pertama memiliki rata-rata *error* pembacaan volume air sebesar 14.8 mL dengan durasi *solenoid valve* menyala 2 detik (lihat pada Lampiran 5). Kemudian, pada pengujian ke-2 memiliki rata-rata *error* sebesar 35.7 mL dengan durasi *solenoid valve* menyala 3 detik. Pada Tabel 4.1 hasil pembacaan volume air memiliki nilai rata-rata total *error* sebesar ± 49.32 mL. perubahan nilai *error* pada volume air sering berubah-ubah dikarenakan laju aliran yang sering berubah seperti pada saat tangki sumber air kondisi penuh dan kondisi akan habis. Permasalahan tersebut dapat diatasi dengan membuat tangki sumber terisi otomatis ketika kondisi air tersisa 1/3 tangki. Pembacaan *flow sensor* dikalibrasi untuk mendapatkan hasil pembacaan yang sesuai dengan alat ukur. Setelah dilakukan kalibrasi pada Tabel 4.2 diperoleh nilai rata-rata *error* pembacaan volume air sebesar ± 34.28 mL. Hasil pembacaan sensor menjadi lebih baik dengan perbedaan *error* ± 15.04 mL.

4.2 Pengujian Komunikasi RS485 Menggunakan Modul Serial MAX485

Pengujian komunikasi RS485 dilakukan menggunakan kabel sepanjang 15 meter. Pengujian dilakukan dengan cara *manual* yaitu dengan merekam data terkirim dan data yang diterima melalui *serial monitor*. Hasil perekaman data ditunjukkan pada Tabel 4.3.

Tabel 4.3 Hasil terbaca komunikasi serial menggunakan MAX485

No	Data dikirim (Arduino <i>slave</i>)	Data diterima (Arduino <i>master</i>)	Status Data		Delay (ms)
	Waktu → Data	Waktu → Data	Behasil	Gagal/ Hilang	
1	15:28:35.111 -> 29.08	15:28:35.157 -> 29.08	√		46
2	15:28:40.163 -> 77.27	15:28:40.163 -> 77.27	√		0
3	15:28:45.134 -> 29.17	15:28:45.188 -> 29.17	√		54
4	15:28:50.180 -> 77.27	-		√	-
5	15:28:55.205 -> 29.17	-		√	-
6	15:29:00.194 -> 77.27	15:29:00.194 -> 77.27	√		0
7	15:29:05.242 -> 29.17	15:29:05.242 -> 29.17	√		0
8	15:29:10.208 -> 77.16	15:29:10.262 -> 77.16	√		54
9	15:29:15.273 -> 29.17	15:29:15.273 -> 29.17	√		0
10	15:29:20.270 -> 77.16	15:29:20.270 -> 77.16	√		0
11	15:29:25.279 -> 29.17	15:29:25.279 -> 29.17	√		0
12	15:29:30.310 -> 77.16	15:29:30.310 -> 77.16	√		0
13	15:29:35.308 -> 29.17	15:29:35.308 -> 29.17	√		0
14	15:29:40.346 -> 77.06	-		√	-
15	15:29:45.339 -> 29.17	15:29:45.339 -> 29.17	√		0
16	15:29:50.360 -> 76.96	15:29:50.360 -> 76.96	√		0
17	15:29:55.379 -> 29.17	15:29:55.379 -> 29.17	√		0
18	15:30:00.403 -> 76.96	15:30:00.403 -> 76.96	√		0
19	15:30:05.423 -> 29.17	-		√	-
20	15:30:10.397 -> 76.96	-		√	-
Total data			15 data terkirim	5 data gagal	
Rata-rata delay					10 ms

Pada Tabel 4.3 menampilkan hasil komunikasi menggunakan MAX485. Kecepatan pengiriman data dari Arduino *slave* ke Arduino *master* cukup baik dengan rata-rata 10 ms, meski terdapat beberapa proses penerimaan data dengan *delay* 40-60 ms. Dari 20 data terkirim dari Arduino *slave* terdapat 5 data yang hilang / *drop* pada saat diterima oleh Arduino Master. Artinya, terdapat 25% data yang tidak terbaca oleh Arduino *master*. Hal ini disebabkan oleh beberapa hal. Format pengiriman data dari Arduino *slave* ke Arduino *master* adalah sebagai berikut,

Kirim data:

string “#”; *delay* (); *data temperature*; *delay* (); *string* “@”; *delay* (); *data kelembapan*; *delay* ();

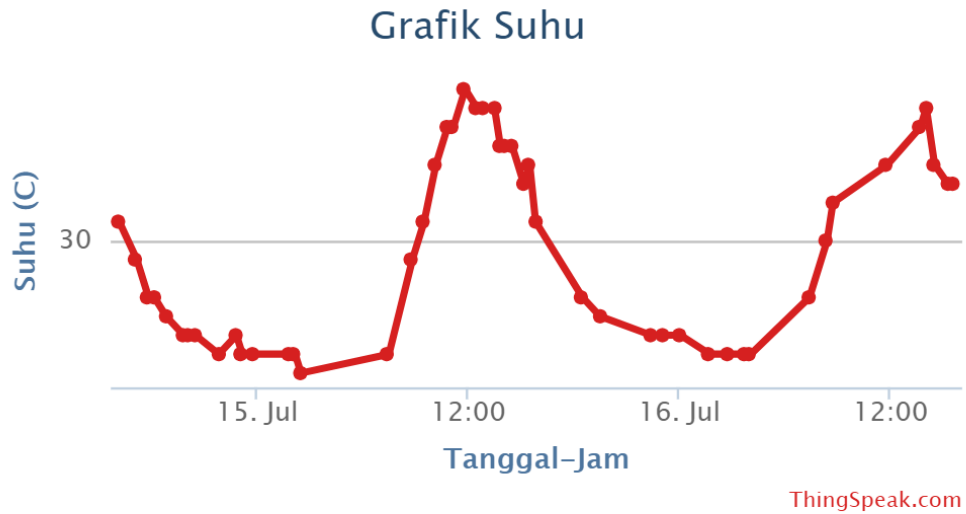
Penyebab pertama permasalahan tersebut adalah karena data yang dikirimkan terdapat 2 parameter yaitu data *temperature* dan kelembapan udara. Sehingga, pada saat diterima oleh *Arduino master* kata kunci *string* “#” untuk *temperature* dan *string* “@” untuk kelembapan udara tertukar sehingga datanya tidak terbaca. Hasil pembacaan yang berhasil memiliki urutan data “#data(*temperature*)” dan “@data(kelembapan)” sehingga jika data yang diterima adalah “#data(*temperature*)” dan “data(kelembapan)” maka data kelembapan menjadi tidak terbaca karena kata kunci “@” berada setelah data yang dikirimkan.

Penyebab kedua adalah metode yang digunakan untuk komunikasi antar *Arduino* adalah metode komunikasi serial yang paling sederhana yaitu dengan cara melakukan “*Serial.print(data)*” menggunakan jalur komunikasi RS485. Meski begitu penggunaan metode ini dapat berjalan baik, karena data dapat terkirim ke *Arduino master* dan dapat dikirimkan ke *database*. Metode lain yang dapat digunakan untuk komunikasi RS485 adalah MODBUS. Namun, komunikasi ini membutuhkan program yang cukup Panjang dan lebih rumit sehingga berdampak pada pemrosesan lain terganggu dan *Arduino* menjadi lambat. Untuk mengatasi beberapa data yang hilang digunakan *library* EEPROM. EEPROM *Arduino Mega* 2560 bersifat *non volatile* dan dapat menulis maupun menghapus data dengan *write cycle capacity* 100.000 - 1 juta kali. Data akan tetap tersimpan di alamat EEPROM *Arduino* meskipun perangkat kondisi *off*. Namun, yang menjadi kekurangan adalah tidak dapat menyimpan nilai *float* dan range nilai yang dapat disimpan adalah 0 – 255.

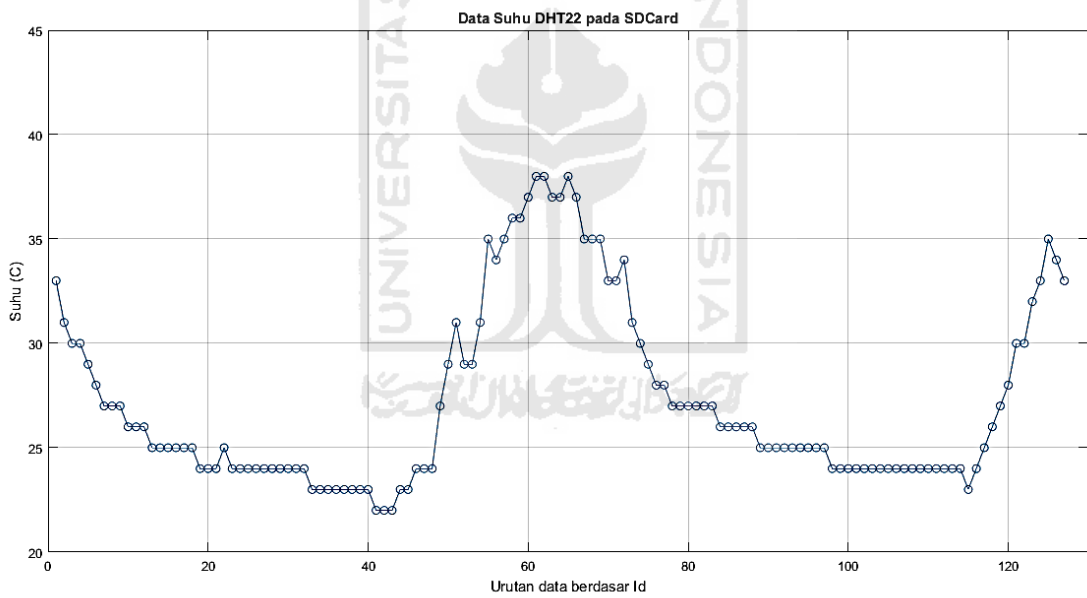
4.3 Menguji *Monitoring* Sistem Secara Keseluruhan

4.3.1 Pengujian *Monitoring Temperature* dan Kelembapan Udara

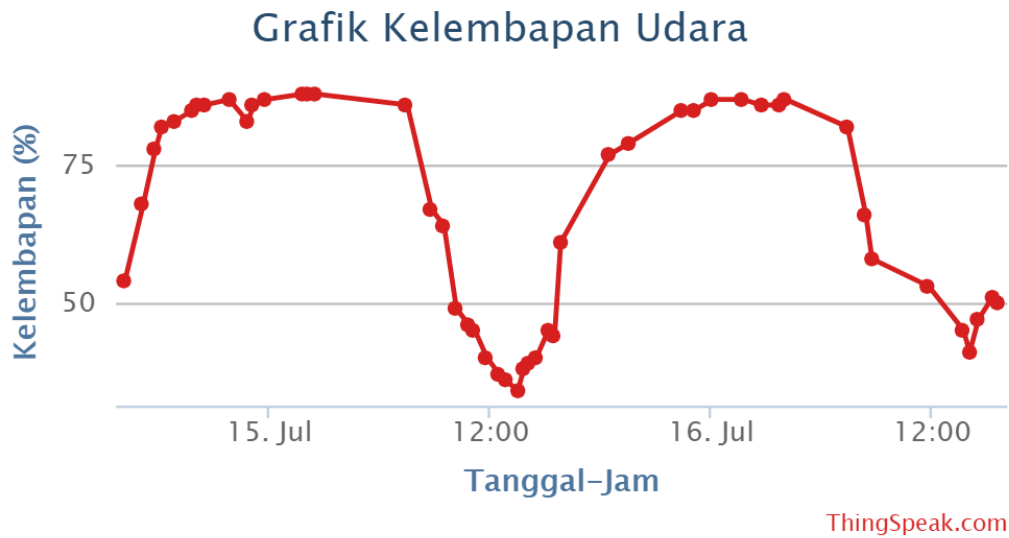
Pengujian sistem *monitoring* dilakukan di *green house* PT. Indmira dalam waktu 3 hari. Pengujian ini dilakukan dengan tujuan membandingkan data pembacaan sensor yang tersimpan di *SDCard* dengan hasil *monitoring* pada *Smartphone*. Program *Arduino* mengirimkan data *temperature* dan kelembapan udara ke *database* setiap 20 menit untuk melihat perubahan nilai yang signifikan. Data parameter *monitoring* terdapat pada lampiran 7. Dari hasil pengujian selama 3 hari diperoleh hasil pembacaan *temperature* dan kelembapan udara pada Gambar 4.3 sampai 4.6.



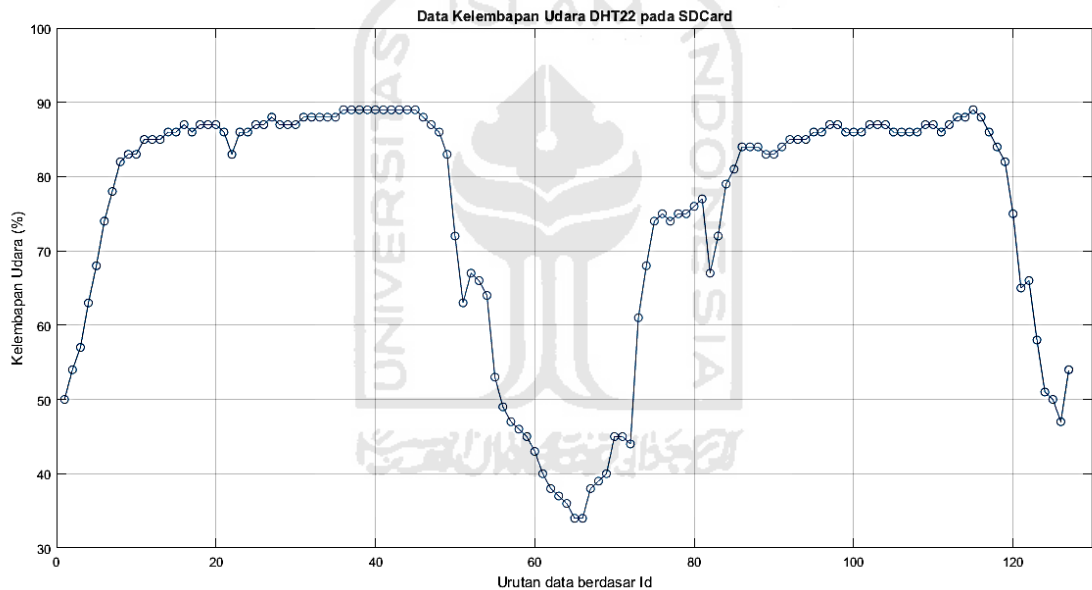
Gambar 4.5 Grafik hasil pengiriman data *temperature* ke *database*



Gambar 4.6 Grafik hasil perekaman data *temperature* pada SDCard



Gambar 4.7 Grafik hasil pengiriman data kelembapan ke *database*



Gambar 4.8 Grafik hasil perekaman data Kelembapan udara pada SDCard

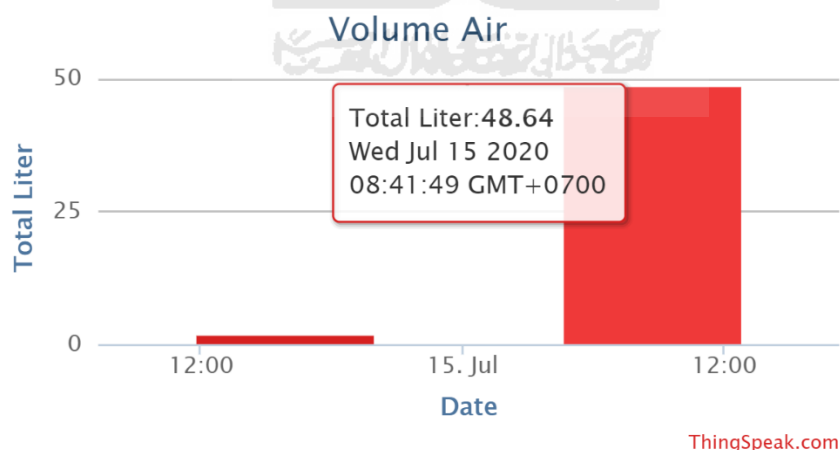
Hasil pengujian *monitoring temperature* dan kelembapan udara pada saat dibandingkan dengan data yang tersimpan di SDCard beberapa data tidak tertampil pada grafik *database*. Beberapa hal yang menyebabkan data tidak tertampil di *database* dipengaruhi pada proses pengiriman data. Penyebab pertama adalah jaringan internet dari SIM800 terkadang hilang sehingga menyebabkan proses pengiriman data menjadi terganggu. Penyebab lainnya adalah pada saat SIM800 terkoneksi dengan baik dan menyala sepanjang waktu proses pengiriman data dapat terganggu dikarenakan jumlah *query* data

yang sering. Sehingga, perlu menutup jaringan dan menghubungkan kembali SIM800 ke jaringan.

Penyebab selanjutnya adalah koneksi ke *web server* terputus (*504 connection timeout*). *Web server* yang digunakan pada sistem *monitoring* ini adalah “000webhostapp.com” dengan akun gratis. Akun gratis memiliki beberapa batasan seperti *reload web* yang lambat dan ketersediaan *server* yang tidak dapat bekerja selama 24/7/365 (sehari penuh) melainkan hanya 23/7/365 atau sekitar 5% waktu dalam sehari *web server* akan mati sehingga koneksi terputus, dalam keadaan tersebut ketika Arduino melakukan pengiriman data terjadi proses *load file PHP* yang lambat maupun terjadi “*504 connection timeout*”. Untuk mengatasi permasalahan tersebut ditambahkan sebuah kartu memori sebagai *datalog* pembacaan sensor apabila akan dilakukan pengolahan data lebih lanjut.

4.3.2 Pengujian *Monitoring Volume Air*

Pengujian ini dilakukan dengan cara membandingkan nilai yang tersimpan di SDCard dan data yang tertampil pada *smartphone*. Data yang terekam merupakan data pada saat sistem PLC melakukan pengisian tangki hidroponik meja A. Proses pengisian tangki dilakukan dalam periode waktu 2 – 4 hari. Keadaan pengisian dapat berubah-ubah disebabkan oleh konsumsi nutrisi bergantung pada besar dan umur tanaman, semakin besar tanaman konsumsi air akan cepat habis. Data *monitoring* parameter terlampir. Hasil pengujian selama 3 hari terdapat pada Gambar 4.5.



Gambar 4.9 Hasil pengujian *monitoring volume* air pengisian setengah tangki

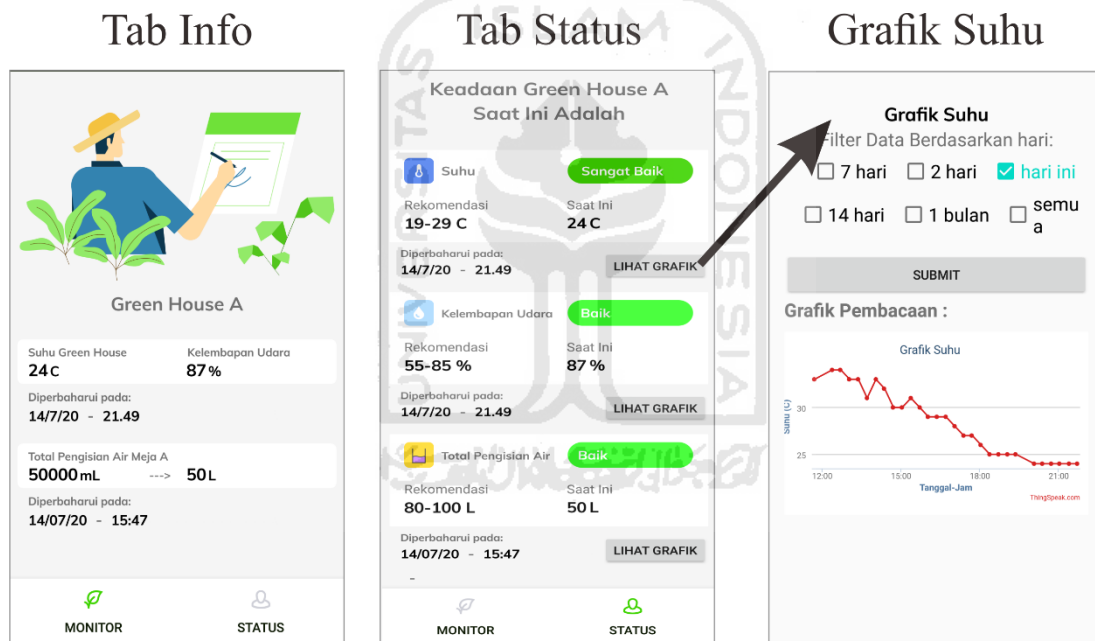
Tabel 4.4 Nilai volume air pada data penyimpanan *database* dan SDCard

Id	Waktu	Data pada SDCard	Data pada <i>Database</i>
1	2020-07-14 15:52:17 WIB	2.03 Liter	2.03 Liter
2	2020-07-15 08:41:49 WIB	48.64 Liter	48.64 Liter

Hasil pengujian *monitoring* volume air selama 3 hari yang tertampil pada grafik *database* dibandingkan dengan data yang tersimpan di SDCard dapat tertampil semua dan nilai yang ditampilkan sesuai. Namun, pengiriman dari Arduino ke *web server* dapat gagal dipengaruhi oleh sinyal maupun koneksi ke *web server* yang terputus. Pembacaan data volume air dapat menjadi acuan pemilik kebun untuk memantau jumlah air setiap pengisian dan periode pengisian tangki dalam sebulan sesuai dengan kondisi atau karakteristik tanaman hidroponik.

4.3.3 Pengujian Fungsi Aplikasi Android

Pengujian ini dilakukan untuk menguji fungsionalitas dari aplikasi dalam menampilkan data dari sensor dan menampilkan grafik. Gambar 4.6 menampilkan informasi kepada pengguna terkait hasil pembacaan sensor oleh Arduino yang tersimpan di *database*.



Gambar 4.10 Pengujian fungsi aplikasi Android

Pengujian aplikasi dari segi fungsional dapat berjalan dengan baik dengan menampilkan data dari *database* berdasarkan data terakhir beserta jam dan tanggal. Perubahan parameter status berubah sesuai dengan kriteria pada Tabel 3.1 dimana pada saat *temperature* terbaca di antara nilai 19 °C dan 29 °C status yang tertampil adalah “Sangat Baik”. Aplikasi Android dapat menampilkan notifikasi pada *notification bar* Android sesuai status pembacaan sensor *temperature*, kelembapan udara, dan volume air. Kemudian, untuk fungsi dari grafik dapat tertampil dengan baik dimana sumbu Y adalah nilai *temperature* dan sumbu X adalah nilai “jam-menit”.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perancangan, pembuatan, dan pengujian terhadap sistem *monitoring* hidroponik di PT. Indmira dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pembacaan sensor DHT22 memiliki rata-rata *error* untuk *temperature* sebesar 0,5⁰C dan kelembapan udara sebesar 2,43%. Sedangkan pembacaan *flow sensor* memiliki nilai rata-rata *error* sebesar 34,28 mL. nilai *error* tersebut menjadi parameter dalam setiap pembacaan sensor yang berlangsung.
2. Pengiriman data dari Arduino ke Android membutuhkan jaringan yang lebih stabil dibandingkan modul GSM SIM800. Karena, GSM SIM800 memiliki kekurangan seperti koneksi terputus tak terduga dan gagal mengirimkan data *monitoring* ke *database*. Selain itu, *web server* “000webhost” sesekali terjadi *timeout* dan *loading file php* yang lambat sehingga mempengaruhi proses pengiriman data dari Arduino ke *database* yang berakibat data gagal terkirim.
3. Sistem komunikasi Arduino *slave* dan Arduino *Master* masih memiliki kendala dalam metode pengiriman data. Karena, masih terjadi *error* penerimaan data di Arduino master sebesar 25% yang menyebabkan data tidak bisa ditampilkan secara *real time*. Penggunaan *memory internal* Arduino (EEPROM) membantu dalam menyimpan data *temperature* dan kelembapan udara sebelum dikirimkan ke *web server*.
4. Aplikasi android dapat digunakan dengan baik sesuai dengan fungsionalitasnya. Data yang terdapat pada database Google Firebase maupun Thingspeak dapat tertampil dengan baik.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilaksanakan perlu adanya perbaikan beberapa bagian untuk mendapatkan hasil yang lebih baik diantaranya adalah:

1. Memperbaiki metode komunikasi serial RS485 agar data yang dikirimkan dari Arduino Slave ke Arduino Master tidak ada yang hilang (*loss data*).
2. Memperbaiki UI/UX aplikasi *monitoring* android baik secara desain dan fitur-fitur yang lebih baik.
3. Memperbaiki metode pengiriman data dari Arduino ke *database* baik menggunakan GSMSIM800C maupun modul lain dengan jaringan internet.
4. Mengganti penyimpanan data yang dikirimkan Arduino *slave* ke *memory card* ataupun ke modul EEPROM *external* sebelum dikirim ke *web server* agar umur Arduino *master* menjadi lebih lama dan dapat menggunakan penyimpanan di EEPROM *external* lebih lama dengan batasan penulisan EEPROM yang lebih besar.



DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, “Persentase Penduduk Daerah Perkotaan menurut Provinsi, 2010-2035,” 2015. <http://www.bps.go.id/linkTabelStatis/view/id/1276>.
- [2] V. Palande, A. Zaheer, and K. George, “Fully Automated Hydroponic System for Indoor Plant Growth,” in *Procedia Computer Science*, 2018, vol. 129, pp. 482–488, doi: 10.1016/j.procs.2018.03.028.
- [3] G. Pamungkas, A. Z. Purwalaksana, M. Djamal, and N. S. Amina, “Rancang Bangun Hidroponik Sistem Nutrient Film Technique Otomatis Berbasis Arduino,” Bandung, 2017.
- [4] D. Anandhavalli, K. S. Sangeetha, V. P. Dharshini, and B. L. Fathima, “Smart Meter for Water Utilization using IoT,” pp. 4–7, 2018.
- [5] S. Hirasawa, M. Nakatsuka, K. Masui, T. Kawanami, and K. Shirai, “Temperature and Humidity Control in Greenhouses in Desert Areas,” *Agric. Sci.*, vol. 05, no. 13, pp. 1261–1268, 2014, doi: 10.4236/as.2014.513134.
- [6] Atmel, “Arduino Mega 2560 Datasheet,” *Power*, 2015. <http://www.robotshop.com/content/PDF/ArduinoMega2560Datasheet.pdf>.
- [7] L. Aosong Electronics Co, “Digital-output relative humidity & temperature sensor/module DHT22,” 2015. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>.
- [8] Shanghai SIMCom Wireless Solutions, “SIM800C_Hardware_Design_V1.02,” *simcom*, 2015. https://simcom.ee/documents/SIM800C-DS/SIM800C-DS_Hardware_Design_V1.01.pdf.
- [9] I. Maxim Integrated Products, “RS-485 / RS-422 Transceivers,” *Current*, 2014. <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>.

LAMPIRAN

Lampiran 1; Data sensor DHT22 (*temperature*) sebelum kalibrasi

No.	Percobaan ke-	Jam	Sensor ($^{\circ}\text{C}$)	Kalibrator ($^{\circ}\text{C}$)	Error ($^{\circ}\text{C}$)	Rata-rata error ($^{\circ}\text{C}$)
1	1	06:17	26.4	26.4	0	0.2 $^{\circ}\text{C}$
2		06:30	24.9	25.7	0.8	
3		06:38	25	25.2	0.2	
4		06:45	25.1	25.1	0	
5		07:00	25.2	25.3	0.1	
6		07:15	25.8	25.8	0	
7		07:30	27	27.3	0.3	
8	2	07:48	27.1	27.5	0.4	0.9 $^{\circ}\text{C}$
9		08:02	29	29.3	0.3	
10		08:15	31.5	32	0.5	
11		08:30	29.7	30.9	1.2	
12		08:33	29.9	31	1.1	
13		08:45	27.9	29.4	1.5	
14		08:47	28	29.3	1.3	
15	3	09:00	28.3	28.5	0.2	0.15 $^{\circ}\text{C}$
16		09:15	27.9	27.9	0	
17		09:30	28.2	28.1	0.1	
18		09:45	29.4	29.2	0.2	
19		10:00	31.7	31.4	0.3	
20		10:15	30.7	30.7	0	
21		10:30	30.6	30.9	0.3	
22	4	10:46	35.9	36	0.1	0.9 $^{\circ}\text{C}$
23		10:48	35.7	36.1	0.4	
24		10:49	35.2	36	0.8	
25		10:49	34.6	36	1.4	
26		11:09	34.4	35.1	0.7	
27		11:15	32.5	33	0.5	
28		11:30	27.6	30	2.4	
29	5	11:45	26.6	26.9	0.3	0.68 $^{\circ}\text{C}$
30		11:01	26.8	26.4	0.4	
31		11:15	26.7	26.3	0.4	
32		13:03	32.1	32.5	0.4	
33		13:15	31.8	33.5	1.7	
34		13:36	31.5	33.1	1.6	
35		13:45	30.3	31.3	1	
36		14:00	30.2	30	0.2	
37		14:15	29.7	29.4	0.3	
38		14:30	29.9	29.4	0.5	

Lampiran 2: Data DHT22 (*temperature*) sesudah kalibrasi

No.	Percobaan ke-	Sensor ($^{\circ}\text{C}$)	Kalibrator ($^{\circ}\text{C}$)	Error ($^{\circ}\text{C}$)	Rata-rata error ($^{\circ}\text{C}$)
1	1	27.05	26.4	0.65	0.56 $^{\circ}\text{C}$
2		25.76	25.7	0.06	
3		25.85	25.2	0.65	
4		25.93	25.1	0.83	
5		26.02	25.3	0.72	
6		26.54	25.8	0.74	
7		27.57	27.3	0.27	
8	2	27.65	27.5	0.15	0.65 $^{\circ}\text{C}$
9		29.29	29.3	0.01	
10		31.44	32	0.56	
11		29.89	30.9	1.01	
12		30.06	31	0.94	
13		28.34	29.4	1.06	
14		28.43	29.3	0.87	
15	3	28.69	28.5	0.19	0.29 $^{\circ}\text{C}$
16		28.34	27.9	0.44	
17		28.6	28.1	0.5	
18		29.63	29.2	0.43	
19		31.61	31.4	0.21	
20		30.75	30.7	0.05	
21		30.67	30.9	0.23	
22	4	35.22	36	0.78	1.27 $^{\circ}\text{C}$
23		35.05	36.1	1.05	
24		34.62	36	1.38	
25		34.11	36	1.89	
26		33.93	35.1	1.17	
27		32.3	33	0.7	
28		28.08	30	1.92	
29	5	27.22	26.9	0.32	0.869 $^{\circ}\text{C}$
30		27.4	26.4	1	
31		27.31	26.3	1.01	
32		31.96	32.5	0.54	
33		31.7	33.5	1.8	
34		31.44	33.1	1.66	
35		30.41	31.3	0.89	
36		30.32	30	0.32	
37		29.89	29.4	0.49	
38		30.06	29.4	0.66	

Lampiran 3; Data sensor DHT22 (kelembapan udara) sebelum kalibrasi

No.	Percobaan ke-	Jam	Sensor (%)	Kalibrator (%)	Error (%)	Rata-rata error (%)
1	1	06:17	92.4	84	8.4	6.78 %
2		06:30	96.3	88	8.3	
3		06:38	96.3	90	6.3	
4		06:45	96.39	92	4.39	
5		07:00	96.4	91	5.4	
6		07:15	95.6	90	5.6	
7		07:30	93.1	84	9.1	
8	2	07:48	92.9	83	9.9	14.32 %
9		08:02	90	77	13	
10		08:15	80.5	68	12.5	
11		08:30	88.9	71	17.9	
12		08:33	85.1	70	15.1	
13		08:45	92.7	77	15.7	
14		08:47	93.2	77	16.2	
15	3	09:00	92.5	81	11.5	11.3 %
16		09:15	93.1	84	9.1	
17		09:30	92.9	85	7.9	
18		09:45	91.5	80	11.5	
19		10:00	84.1	71	13.1	
20		10:15	88.3	76	12.3	
21		10:30	88.7	75	13.7	
22	4	10:46	67.5	53	14.5	15.75 %
23		10:48	71.1	54	17.1	
24		10:49	70.4	54	16.4	
25		10:49	68.4	55	13.4	
26		11:09	74.3	58	16.3	
27		11:15	79.4	63	16.4	
28		11:30	94.2	78	16.2	
29	5	11:45	95.4	86	9.4	10.93 %
30		11:01	95.3	88	7.3	
31		11:15	95.3	89	6.3	
32		13:03	78.8	65	13.8	
33		13:15	81.4	65	16.4	
34		13:36	81.6	65	16.6	
35		13:45	82.4	69	13.4	
36		14:00	85.9	75	10.9	
37		14:15	84.9	77	7.9	
38		14:30	86.3	79	7.3	

Lampiran 4: Data sensor DHT22 (kelembapan udara) sesudah kalibrasi

No.	Percobaan ke-	Sensor (%)	Kalibrator (%)	Error (%)	Rata-rata error (%)
1	1	82.31	84	1.69	2.49 %
2		87.34	88	0.66	
3		87.34	90	2.66	
4		87.46	92	4.54	
5		87.47	91	3.53	
6		86.44	90	3.56	
7		83.21	84	0.79	
8	2	82.96	83	0.04	3.57 %
9		79.22	77	2.22	
10		66.96	68	1.04	
11		77.8	71	6.8	
12		72.9	70	2.9	
13		82.7	77	5.7	
14		83.34	77	6.34	
15	3	82.44	81	1.44	1.37 %
16		83.21	84	0.79	
17		82.96	85	2.04	
18		81.15	80	1.15	
19		71.61	71	0.61	
20		77.02	76	1.02	
21		77.54	75	2.54	
22	4	50.2	53	2.8	2.49 %
23		54.84	54	0.84	
24		53.94	54	0.06	
25		51.36	55	3.64	
26		58.97	58	0.97	
27		65.54	63	2.54	
28		84.63	78	6.63	
29	5	86.18	86	0.18	2.221 %
30		86.05	88	1.95	
31		86.05	89	2.95	
32		64.77	65	0.23	
33		68.12	65	3.12	
34		68.38	65	3.38	
35		69.41	69	0.41	
36		73.93	75	1.07	
37		72.64	77	4.36	
38		74.44	79	4.56	

Lampiran 5: Data flow sensor sebelum kalibrasi

No.	Percobaan ke-	Durasi relay ON	Sensor (mL)	Gelas Ukur (mL)	Error (mL)	Rata-rata error (mL)
1	1	2 detik	250	210	40	14.8 mL
2			183	190	7	
3			349	330	19	
4			114	120	6	
5			140	150	10	
6			97	100	3	
7			117	110	7	
8			119	100	19	
9			119	100	19	
10			82	100	18	
11	2	3 detik	316	300	16	35.7 mL
12			331	295	36	
13			336	300	36	
14			310	300	10	
15			332	300	32	
16			331	300	31	
17			327	295	32	
18			316	300	16	
19			324	300	24	
20			319	195	124	
21	3	5 detik	547	510	37	42.3 mL
22			565	500	65	
23			621	580	41	
24			628	600	28	
25			484	480	4	
26			434	470	36	
27			586	470	116	
28			590	550	40	
29			571	550	21	
30			605	570	35	
31	4	8 detik	786	700	86	92.2 mL
32			779	700	79	
33			764	700	64	
34			766	700	66	
35			938	850	88	
36			929	810	119	
37			942	780	162	
38			950	850	100	
39			965	850	115	
40			793	750	43	
41	5	10 detik	1198	1040	158	61.6 mL
42			1232	1200	32	

No.	Percobaan ke-	Durasi relay ON	Sensor (mL)	Gelas Ukur (mL)	Error (mL)	Rata-rata <i>error</i> (mL)
43			1258	1200	58	
44			1198	1200	2	
45			1160	1090	70	
46			1030	1000	30	
47			1260	1150	110	
48			1258	1200	58	
49			1037	1100	63	
50			1175	1140	35	



Lampiran 6: Data flow sensor sesudah kalibrasi

No.	Percobaan ke-	Durasi relay ON	Sensor (mL)	Gelas Ukur (mL)	Error (mL)	Rata-rata error (mL)
1	1	2 detik	234	210	24	15.8 mL
2			168	190	22	
3			331	330	1	
4			100	120	20	
5			125	150	25	
6			83	100	17	
7			103	110	7	
8			105	100	5	
9			105	100	5	
10			68	100	32	
11	2	3 detik	299	300	1	19.8 mL
12			313	295	18	
13			318	300	18	
14			293	300	7	
15			314	300	14	
16			313	300	13	
17			309	295	14	
18			299	300	1	
19			306	300	6	
20			301	195	106	
21	3	5 detik	526	510	16	27.9 mL
22			543	500	43	
23			598	580	18	
24			605	600	5	
25			464	480	16	
26			415	470	55	
27			564	470	94	
28			568	550	18	
29			549	550	1	
30			583	570	13	
31	4	8 detik	761	700	61	65.7 mL
32			754	700	54	
33			739	700	39	
34			741	700	41	
35			910	850	60	
36			901	810	91	
37			914	780	134	
38			922	850	72	
39			937	850	87	
40			768	750	18	
41	5	10 detik	1166	1040	126	42.2 mL
42			1199	1200	1	
43			1225	1200	25	

44			1166	1200	34	
45			1128	1090	38	
46			1001	1000	1	
47			1227	1150	77	
48			1225	1200	25	
49			1008	1100	92	
50			1143	1140	3	



Lampiran 7: Data perekaman *temperature* dan kelembapan udara pada SDCard dan Database

Id	Tanggal	Waktu	<i>Temperature</i> pada SDCard (°C)	<i>temperature</i> pada Database (°C)	Kelembapan pada SDCard (%)	Kelembapan pada Database (%)
1	16/07/2020	10:41:13 WIB	31	-	57	-
2	16/07/2020	10:51:13 WIB	33	-	59	-
3	16/07/2020	11:01:13 WIB	32	-	54	-
4	16/07/2020	11:33:58 WIB	33	-	56	-
5	16/07/2020	11:48:19 WIB	34	34	53	53
6	16/07/2020	11:58:19 WIB	34	-	51	-
7	16/07/2020	12:08:19 WIB	34	-	50	-
8	16/07/2020	12:44:35 WIB	34	-	49	-
9	16/07/2020	12:44:35 WIB	33	-	49	-
10	16/07/2020	13:21:34 WIB	36	36	45	45
11	16/07/2020	13:21:34 WIB	37	37	41	41
12	16/07/2020	14:29:51 WIB	34	34	47	47
13	16/07/2020	14:51:17 WIB	33	-	50	-
14	16/07/2020	15:15:07 WIB	33	33	51	51
15	16/07/2020	15:35:07 WIB	33	33	50	50
16	16/07/2020	15:55:09 WIB	32	32	54	54
17	16/07/2020	16:15:07 WIB	31	31	56	56
18	16/07/2020	16:35:07 WIB	30	30	60	60
19	16/07/2020	16:55:06 WIB	29	29	61	61
20	16/07/2020	17:16:39 WIB	29	-	63	-
21	16/07/2020	17:35:08 WIB	28	28	66	66
22	16/07/2020	17:55:08 WIB	27	27	77	77
23	16/07/2020	18:15:08 WIB	26	26	83	83
24	16/07/2020	18:35:14 WIB	26	26	84	84
25	16/07/2020	18:55:08 WIB	26	-	84	-
26	16/07/2020	19:15:08 WIB	25	25	85	85
27	16/07/2020	19:36:39 WIB	25	-	85	-
28	16/07/2020	19:55:08 WIB	25	25	86	86
29	16/07/2020	20:15:08 WIB	25	25	86	86
30	16/07/2020	20:35:08 WIB	25	25	86	86
31	16/07/2020	20:55:08 WIB	25	25	87	87
32	16/07/2020	21:15:14 WIB	25	25	86	86
33	16/07/2020	21:35:08 WIB	25	25	86	86
34	16/07/2020	21:56:39 WIB	25	-	85	-
35	16/07/2020	22:16:21 WIB	24	24	86	86
36	16/07/2020	22:35:09 WIB	24	24	86	86
37	16/07/2020	22:55:09 WIB	24	24	86	86
38	16/07/2020	23:15:59 WIB	24	-	87	-
39	16/07/2020	23:35:06 WIB	23	23	87	87
40	16/07/2020	23:55:06 WIB	23	23	87	87

41	17/07/2020	00:15:07	WIB	23	23	88	88
42	17/07/2020	00:36:38	WIB	23	23	88	88
43	17/07/2020	00:55:07	WIB	23	23	88	88
44	17/07/2020	01:15:07	WIB	23	23	88	88
45	17/07/2020	01:35:07	WIB	23	23	88	88
46	17/07/2020	01:55:09	WIB	23	23	88	88
47	17/07/2020	02:15:07	WIB	23	23	89	89
48	17/07/2020	02:35:07	WIB	23	23	89	89
49	17/07/2020	02:56:38	WIB	23	-	89	-
50	17/07/2020	03:16:38	WIB	23	-	89	-
51	17/07/2020	03:35:36	WIB	22	-	89	-
52	17/07/2020	03:56:38	WIB	22	-	89	-
53	17/07/2020	04:16:38	WIB	22	-	89	-
54	17/07/2020	04:36:00	WIB	22	22	89	89
55	17/07/2020	04:55:09	WIB	22	22	89	89
56	17/07/2020	05:15:06	WIB	22	22	89	89
57	17/07/2020	05:35:07	WIB	22	22	90	90
58	17/07/2020	05:55:06	WIB	22	22	89	89
59	17/07/2020	06:15:05	WIB	22	22	89	89
60	17/07/2020	06:35:06	WIB	22	22	89	89
61	17/07/2020	06:56:37	WIB	23	-	89	-
62	17/07/2020	07:15:48	WIB	25	25	87	87
63	17/07/2020	07:35:06	WIB	26	26	85	85
64	17/07/2020	07:55:09	WIB	28	-	80	-
65	17/07/2020	08:15:55	WIB	28	28	75	75
66	17/07/2020	08:35:06	WIB	30	30	69	69
67	17/07/2020	08:55:06	WIB	31	31	61	61
68	17/07/2020	09:15:58	WIB	31	-	67	-

Lampiran 8: Data Penduduk Daerah Perkotaan menurut Provinsi,

7/17/2020

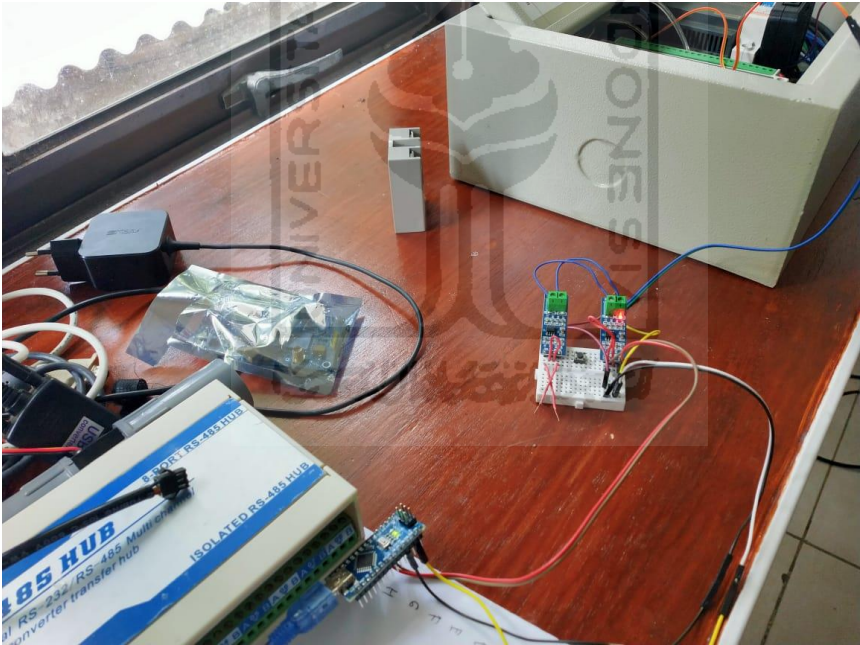
Persentase Penduduk Daerah Perkotaan Hasil Proyeksi Penduduk menurut Provinsi, 2015 - 2045

Persentase Penduduk Daerah Perkotaan menurut Provinsi, 2010-2035

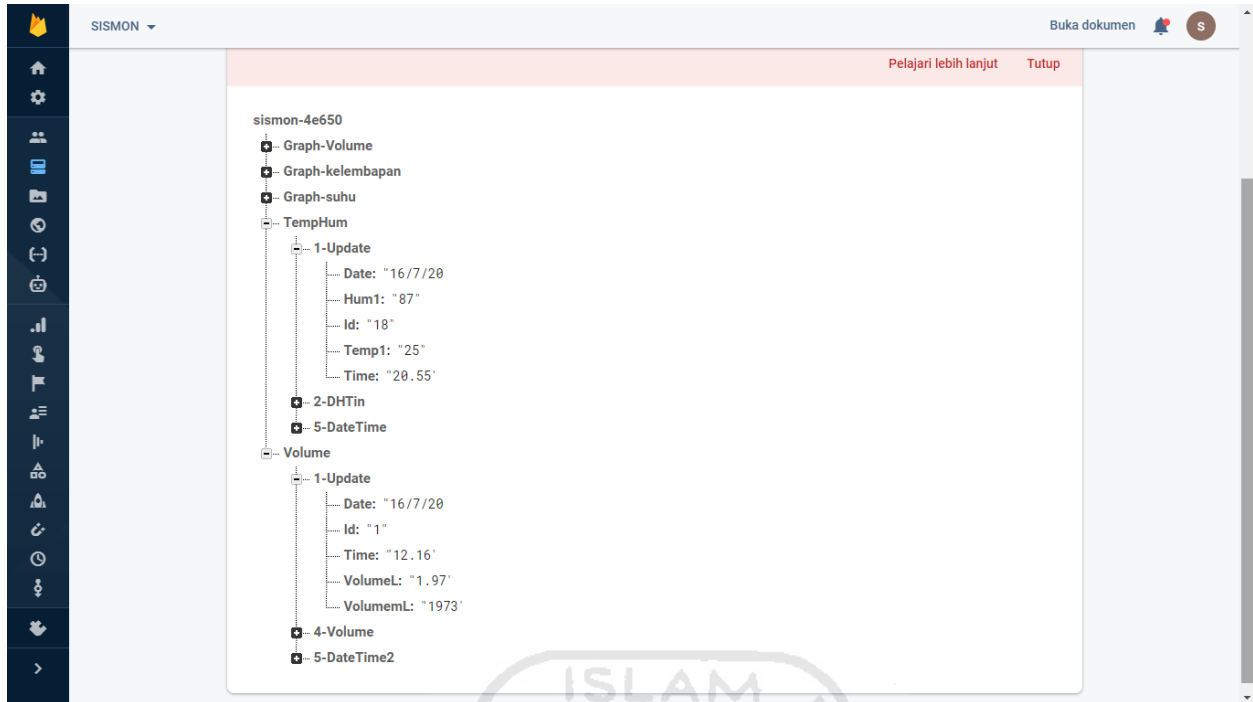
Provinsi	Tahun					
	2010	2015	2020	2025	2030	2035
Aceh	28,1	30,5	33,2	36,2	39,5	43,2
Sumatera Utara	49,2	52,6	56,3	60,1	64,1	68,1
Sumatera Barat	38,7	44,2	49,6	54,6	59,4	63,8
Riau	39,2	39,6	40,1	40,7	41,2	41,8
Jambi	30,7	32,0	33,3	34,8	36,5	38,2
Sumatera Selatan	35,8	36,5	37,3	38,2	39,1	40,1
Bengkulu	31,0	31,7	32,6	33,5	34,5	35,6
Lampung	25,7	28,3	31,3	34,6	38,3	42,4
Kepulauan Bangka Belitung	49,2	52,5	56,0	59,7	63,5	67,4
Kepulauan Riau	82,8	83,0	83,3	83,8	84,5	85,3
DKI Jakarta	100,0	100,0	100,0	100,0	100,0	100,0
Jawa Barat	65,7	72,9	78,7	83,1	86,6	89,3
Jawa Tengah	45,7	48,4	51,3	54,3	57,5	60,8
DI Yogyakarta	66,4	70,5	74,6	78,0	81,3	84,1
Jawa Timur	47,6	51,1	54,7	58,6	62,6	66,7
Banten	67,0	67,7	69,9	73,7	78,8	84,9
Bali	60,2	65,5	70,2	74,3	77,8	81,2
Nusa Tenggara Barat	41,7	45,4	49,4	53,6	58,1	62,7
Nusa Tenggara Timur	19,3	21,6	24,3	27,3	30,7	34,6
Kalimantan Barat	30,2	33,1	36,2	39,8	43,7	47,9
Kalimantan Tengah	33,5	36,6	40,2	44,1	48,3	52,9
Kalimantan Selatan	42,1	45,1	48,4	52,0	55,8	59,8
Kalimantan Timur	63,2	66,0	68,9	71,8	74,8	77,7
Sulawesi Utara	45,2	49,8	54,7	59,2	63,9	68,7
Sulawesi Tengah	24,3	27,2	30,5	34,2	38,4	43,1
Sulawesi Selatan	36,7	40,6	45,0	49,8	54,9	59,6
Sulawesi Tenggara	27,4	31,2	35,0	39,4	43,6	48,3
Gorontalo	34,0	39,0	44,0	48,9	53,5	58,4
Sulawesi Barat	22,9	22,9	23,0	23,0	23,1	23,1
Maluku	37,1	38,0	38,9	39,9	41,0	42,1
Maluku Utara	27,1	27,8	28,5	29,2	29,9	30,6
Papua Barat	29,9	32,3	34,9	37,8	40,9	44,4
Papua	26,0	28,4	31,2	34,2	37,7	41,5
INDONESIA	49,8	53,3	56,7	60,0	63,4	66,6

<https://www.bps.go.id/statictable/2014/02/18/1276/persentase-penduduk-daerah-perkotaan-menurut-provinsi-2010-2035.html>

Lampiran 9: Dokumentasi *build-up* Sistem Monitoring



Lampiran 10: Tampilan *database* Google Firebase



The screenshot displays the Google Firebase console interface for a project named 'sismon-4e650'. The database is structured as follows:

- sismon-4e650
 - Graph-Volume
 - Graph-kelembapan
 - Graph-suhu
 - TempHum
 - 1-Update
 - Date: "16/7/20"
 - Hum1: "87"
 - Id: "18"
 - Temp1: "25"
 - Time: "20.55"
 - 2-DHTin
 - 5-DateTime
 - Volume
 - 1-Update
 - Date: "16/7/20"
 - Id: "1"
 - Time: "12.16"
 - VolumeL: "1.97"
 - VolumemL: "1973"
 - 4-Volume
 - 5-DateTime2



Lampiran 11: Flow chart aplikasi android

