

BAB III

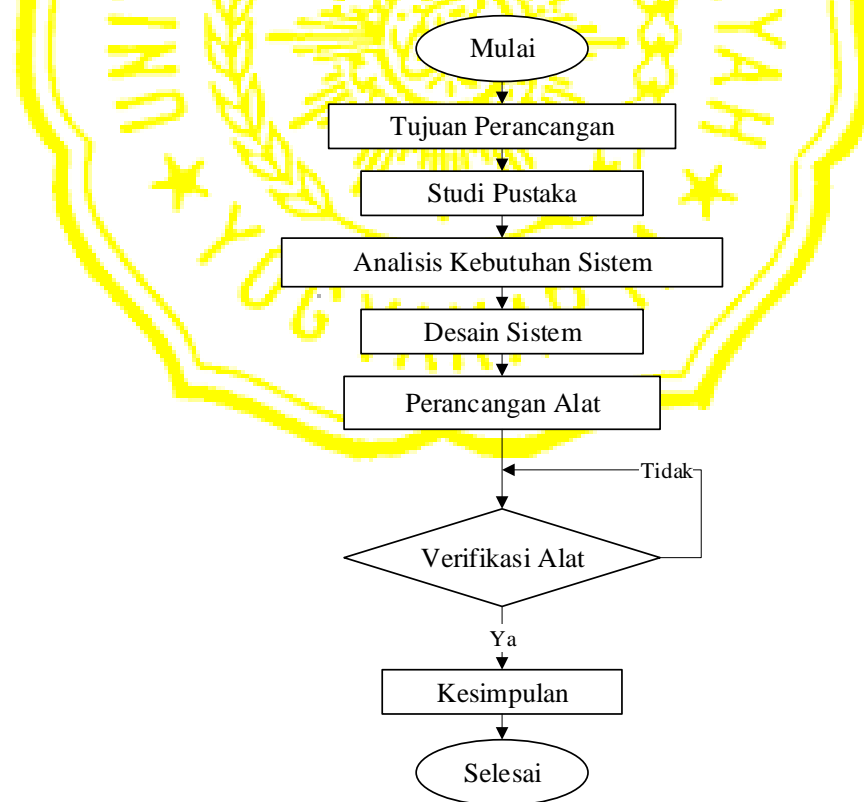
METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian untuk tugas akhir ini dilakukan dari bulan April 2019 sampai dengan bulan Juli 2019 di Laboratorium Teknik Elektro Universitas Muhammadiyah Yogyakarta.

3.2 Prosedur Penelitian

Prosedur penelitian merupakan langkah-langkah yang harus dikerjakan secara runtut dalam penelitian tugas akhir ini. Prosedur penelitian dalam tugas akhir ini digambarkan ke dalam diagram blok berikut ini :



Gambar 3.1 Diagram Blok Prosedur Penelitian Tugas Akhir

Diagram blok prosedur penelitian tugas akhir di atas dapat dijelaskan sebagai berikut :

1. Tujuan Perancangan

Penelitian tugas akhir ini dimulai dengan pembuatan tujuan pembuatan dasar tentang fungsi dan cara kerja dari pembuatan *data logger* untuk sensor suhu, kelembaban, dan tekanan udara dengan menggunakan Arduino Uno dan mengintegrasikan modul GSM/GPRS SIM900A sebaga media pengirim data ke *server database*.

2. Studi Pustaka

Langkah berikutnya yaitu studi pustaka. Pada tahap ini, studi pustaka dilakukan dengan mengumpulkan informasi dan referensi-referensi dari jurnal-jurnal dan informasi lainnya dari internet.

3. Analisis Kebutuhan Sistem

Setelah studi pustaka dilakukan maka akan dihasilkan kebutuhan sistem yang harus dicapai dan terpenuhi agar hasil dari alat yang dibuat sesuai dengan tujuan yang telah ditetapkan di awal. Kebutuhan sistem yang harus terpenuhi dalam sistem tugas akhir ini antara lain sebagai berikut :

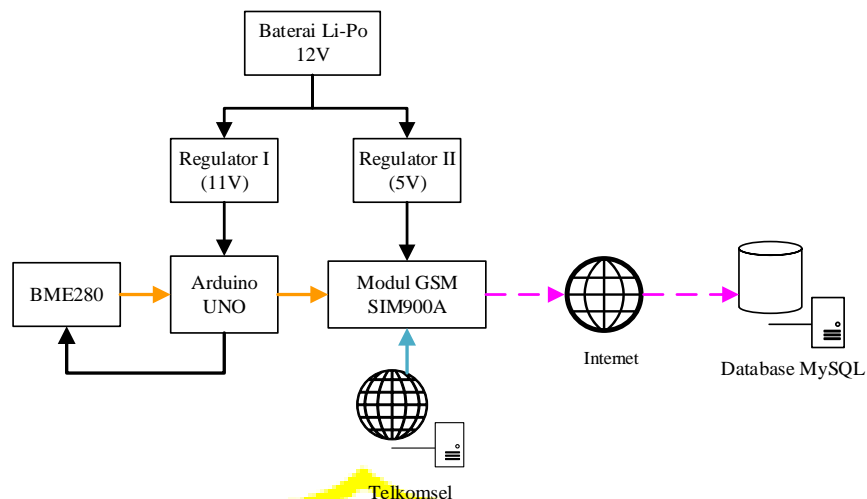
- 1) Komputer atau laptop yang digunakan
- 2) Mikrokontroler yang digunakan
- 3) Sensor yang digunakan sebagai pendeteksi tingkat suhu, kelembaban dan tekanan lokasi penelitian.

- 4) Modul GSM/GPRS sebagai media transfer data ke server *database*.
- 5) *Domain* dan *hosting* yang digunakan untuk server *database*.
- 6) Penyedia layanan seluler yang digunakan sebagai penyedia jaringan internet.
- 7) Komponen pendukung lainnya, yaitu : pin deret, pin *header*, kabel *jumper*, regulator, dan lain-lain.

4. Desain Sistem

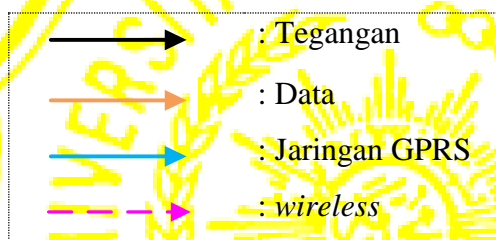
Ketika kebutuhan sistem telah selesai dianalisis maka langkah berikutnya adalah melakukan desain sistem. Desain data *logger* suhu, kelembaban, dan tekanan udara ini memiliki 2 (dua) bagian yaitu perangkat keras atau *hardware* dan perangkat lunak atau *software*. Desain perangkat keras terdiri atas pembuatan komponen elektronik agar antar komponen *hardware* yang digunakan terhubung dan dapat aktif ketika digunakan, sedangkan desain perangkat lunak meliputi pembuatan *database* MySQL dengan menggunakan CPanel, pembuatan program pendeteksi tingkat suhu, kelembaban dan tekanan udara, dan pembuatan program untuk mentransmisikan data sensor ke server *database* yang telah dibuat, serta pembuatan *script* PHP .

Diagram blok dari keseluruhan desain sistem dalam penelitian tugas akhir *data logger* untuk sensor suhu, kelembaban, dan tekanan udara menggunakan Arduino Uno dan modul GSM/GPRS SIM900A dapat ditunjukkan seperti pada Gambar 3.2 sebagai berikut :



Gambar 3.2 Diagram Blok Desain Sistem

Keterangan tanda panah :



Dari diagram blok di atas, sistem dalam penelitian tugas akhir ini memiliki prinsip kerja sebagai berikut :

1. Baterai Li-Po sebagai catu daya untuk Arduino Uno dan modul GSM/GPRS SM900A
2. Regulator I sebagai penstabil tegangan turunan ke Arduino dan regulator II sebagai penstabil tegangan turunan ke modul GSM/GPRS SIM900A
3. BME280 akan aktif dengan mendapatkan sumber tegangan dari Arduino Uno.
4. Arduino Uno akan menerima data tingkat temperatur, kelembaban, dan tekanan dari sensor BME280.

5. Modul GSM/GPRS SIM900A akan menerima data olahan dari Arduino Uno dengan komunikasi serial UART.
6. Modul GSM/GPRS SIM900A akan mentransmisikan data sensor dari Arduino Uno ke *database* MySQL menggunakan jaringan GPRS dari penyedia layanan seluler Telkomsel.
7. Modul GSM/GPRS SIM900A mengirimkan data sensor ke *database* MySQL melalui internet secara *wireless*.

Berikut adalah penjelasan untuk masing-masing bagian sistem dalam tugas akhir ini :

1) Desain Sistem Perangkat Keras

Perancangan desain sistem perangkat keras dalam tugas akhir ini menggunakan alat dan bahan sesuai dengan Tabel 3.1 sebagai berikut :

No	Alat	No	Bahan
1	Laptop	1	Arduino Uno
2	Tang Potong	2	Sensor BME280
3	Solder dan Tenol	3	Kartu Perdana SimPATI
4	Multimeter Digital	4	Modul SIM900A
5	Kabel <i>jumper male-female</i>	5	Papan PCB lubang 2,5mm
6	Mur & Baut M3	6	Pin Header <i>male-female</i>
7	T-Blok	7	Regulator <i>step down</i> DC-DC L2596
		8	Termometer Dinding

Tabel 3.1 Alat dan Bahan yang Digunakan

2) Desain Sistem Perangkat Lunak

Sistem perangkat lunak dibuat untuk memproses dan mengontrol proses kerja dari seluruh sistem yang telah dirancang. Desain sistem perangkat lunak dalam tugas akhir ini menggunakan Arduino IDE, CPanel dan Visual Studio Code. Arduino IDE digunakan untuk membuat program pendeteksian untuk sensor BME280, program untuk menghubungkan modul GSM SIM900A ke jaringan GPRS Telkomsel dan basis data (*database*) serta untuk membuat program untuk mentransfer data ke basis data. CPanel sebagai tempat pembuatan *database* dengan menggunakan layanan Telkom WebHosting. Visual Studio Code digunakan untuk pembuatan file PHP.

5. Perancangan Alat

Tahap berikutnya yaitu perancangan atau pembuatan rangkaian elektronik, penyambungan komponen elektronik ke mikrokontroler, dan penyambungan mikrokontroler ke modul GSM/GPRS.

6. Verifikasi Alat

Setelah alat selesai dibuat, maka langkah berikutnya adalah memverifikasi alat tersebut guna mengetahui apakah alat yang telah dibuat tersebut dapat bekerja sesuai dengan prinsip kerja yang telah ditetapkan dengan baik atau belum. Apabila masih terdapat *error* ataupun kesalahan lainnya maka dilakukan pengecekan dan perbaikan kembali sehingga alat dapat beroperasi sesuai dengan prinsip kerja dengan baik dan sesuai dengan tujuan awal dari pembuatan alat.

Ketika alat sudah dapat beroperasi normal maka akan dilakukan pengambilan data yang dibutuhkan dan langkah berikutnya.

7. Kesimpulan

Setelah alat diuji untuk pengambilan data yang dibutuhkan, langkah berikutnya adalah mengambil kesimpulan dari keseluruhan uji coba dan penelitian dari Tugas Akhir ini.

3.3 Perancangan Perangkat Keras

Perancangan perangkat keras dalam penelitian ini menggunakan beberapa komponen utama untuk mengatur proses kerja dari *data logger*. Komponen utama yang dimaksud adalah mikrokontroler, sensor dan modul GSM/GPRS. *Board* mikrokontroler yang digunakan adalah Arduino Uno. Sensor yang digunakan adalah sensor BME280 dari Bosch Sensortec. Modul GSM/GPRS yang digunakan adalah modul GSM/GPRS SIM900A.

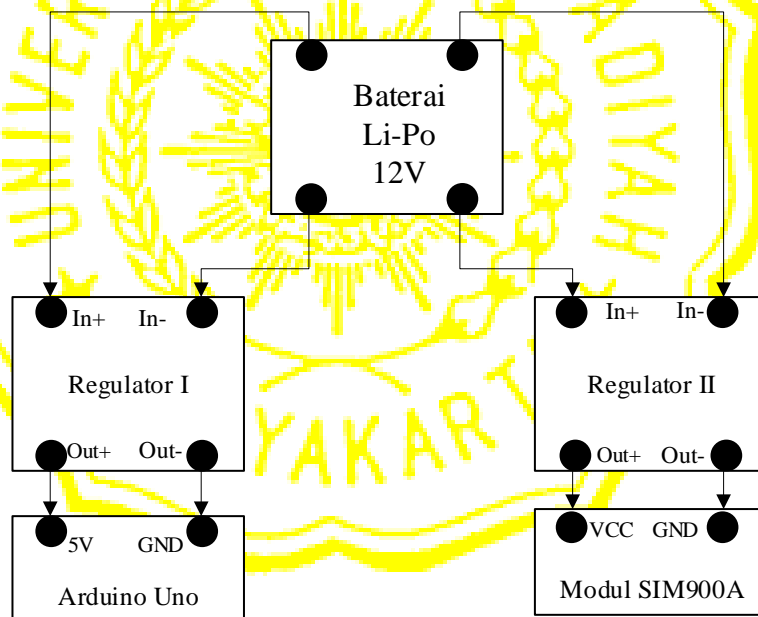
Perancangan perangkat keras (*hardware*) memerlukan tahapan perancangan utama diantaranya yaitu perancangan catu daya, koneksi sensor dengan Arduino Uno, dan komunikasi serial UART antara Arduino Uno dengan modul SIM900A. Masing-masing dari tahapan tersebut dapat dijelaskan sebagai berikut ini :

3.3.1. Perancangan Catu Daya

Perancangan catu daya dalam sistem perangkat keras untuk penelitian tugas akhir ini dimaksudkan untuk memberikan tegangan *input*-an eksternal untuk Arduino Uno dan modul GSM/GPRS SIM900A untuk menghindari adanya kerusakan regulator 5V Arduino Uno jika langsung dihubungkan ke modul SIM900A.

Catu daya yang dirancang dalam penelitian ini menggunakan 2 regulator *step-down* DC-DC L2596 yang berfungsi sebagai penstabil tegangan. Regulator pertama merupakan regulator dengan tegangan keluaran sebesar 11 V untuk Arduino Uno dan regulator ke-dua memiliki tegangan keluaran sebesar 5V untuk menyuplai tegangan pada modul GSM/GPRS SIM900A. Kedua regulator tersebut dihubungkan ke baterai Li-Po 3 sel dengan tegangan 12 V.

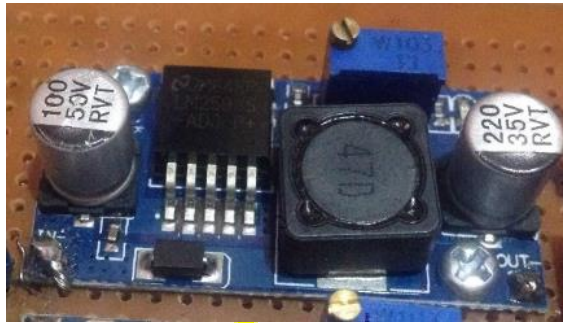
Perancangan catu daya dengan menggunakan 2 regulator dalam penelitian ini memiliki skema rangkaian yang sesuai dengan Gambar 3.3 sebagai berikut :



Gambar 3.3 Skema Rangkaian Catu Daya

Rangkaian catu daya dirancang dengan dengan cara diparalelkan ke dalam satu sumber daya. Cara ini dilakukan karena tegangan *input* yang dibutuhkan untuk modul SIM900A dan Arduino Uno berbeda sehingga dibutuhkan pembagian *output* tegangan yang berbeda dari sumber tegangan.

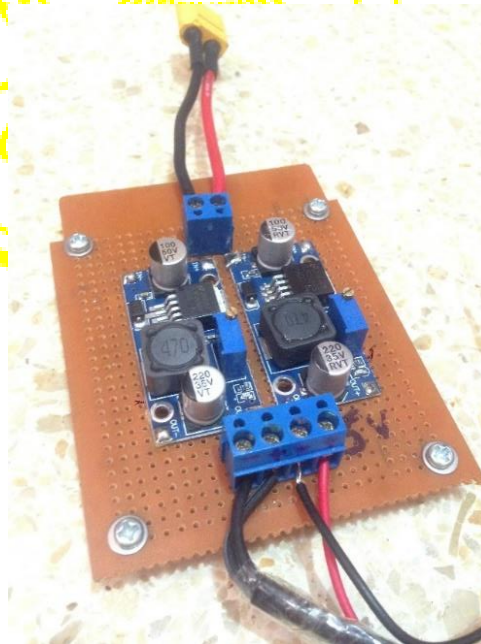
Kemudian untuk regulator *step-down* LM2596 yang digunakan dalam penelitian ini adalah sesuai Gambar 3.4 di bawah ini :



Gambar 3.4 Regulator *Step-Down* LM2596

LM2596 dipilih karena regulator ini adalah *switching regulator* yang mana memiliki efisiensi lebih tinggi daripada *linear regulator* sehingga akan lebih hemat dalam penggunaan tegangan dari sumber tegangan.

Rangkaian catu daya yang telah dibuat untuk menyuplai tegangan ke Arduino Uno dan modul GSM/GPRS SIM900A adalah sesuai dengan Gambar 3.5 sebagai berikut :



Gambar 3.5 Rangkaian Catu Daya

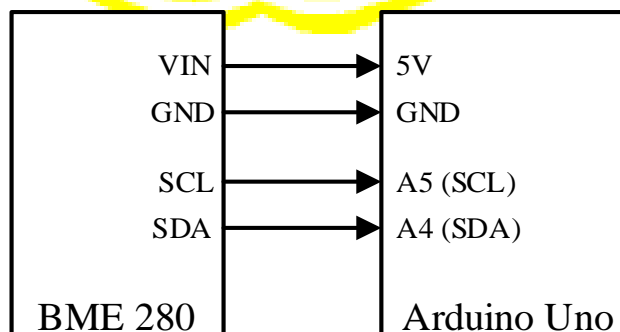
3.3.2. Perancangan Koneksi Sensor dengan Arduino Uno

Sistem dalam penelitian tugas akhir ini menggunakan sensor lingkungan yaitu sensor BME280. Sensor ini berfungsi untuk mendeteksi keadaan lingkungan sekitar yang meliputi tingkat temperatur, tingkat kelembaban, dan tingkat tekanan udara di sekitar lingkungan yang dijadikan sebagai objek pengukuran. Sensor BME280 yang digunakan memiliki tampilan sesuai Gambar 3.6 berikut ini :



Gambar 3.6 Tampilan Sensor BME280

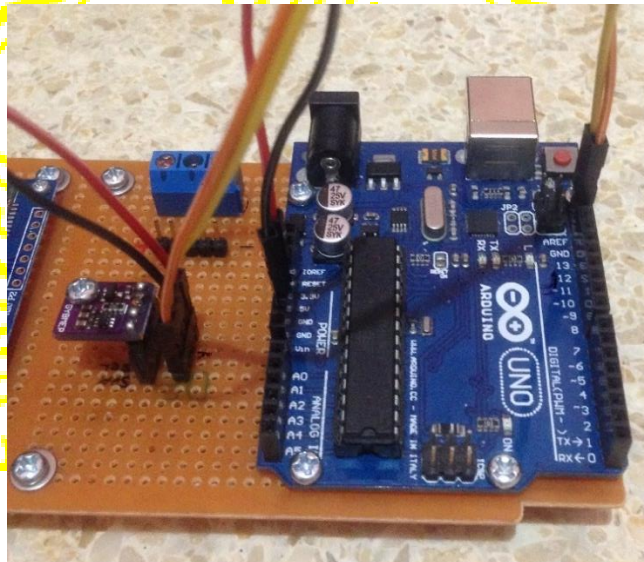
Untuk menghubungkan sensor dengan Arduino Uno diperlukan koneksi antar pin kedua *device* tersebut dengan tepat. Skema koneksi antara sensor BME280 dengan Arduino Uno sesuai dengan Gambar 3.7 berikut ini :



Gambar 3.7 Skema Rangkaian Koneksi Sensor BME280 dengan
Arduino Uno

Komunikasi yang digunakan dalam koneksi sensor BME280 dengan Arduino Uno adalah I2C (*Inter-Integrated Circuit*) . Komunikasi ini dipilih karena meskipun I2C bersifat *synchronous* seperti halnya SPI, namun dalam I2C hanya menggunakan dua kabel untuk komunikasi yaitu SCL (*Synchronous Clock*) dan SDA (*Synchronous Data*).

Hasil rangkaian koneksi antara sensor BME280 dengan Arduino Uno adalah sesuai Gambar 3.8 sebagai berikut :



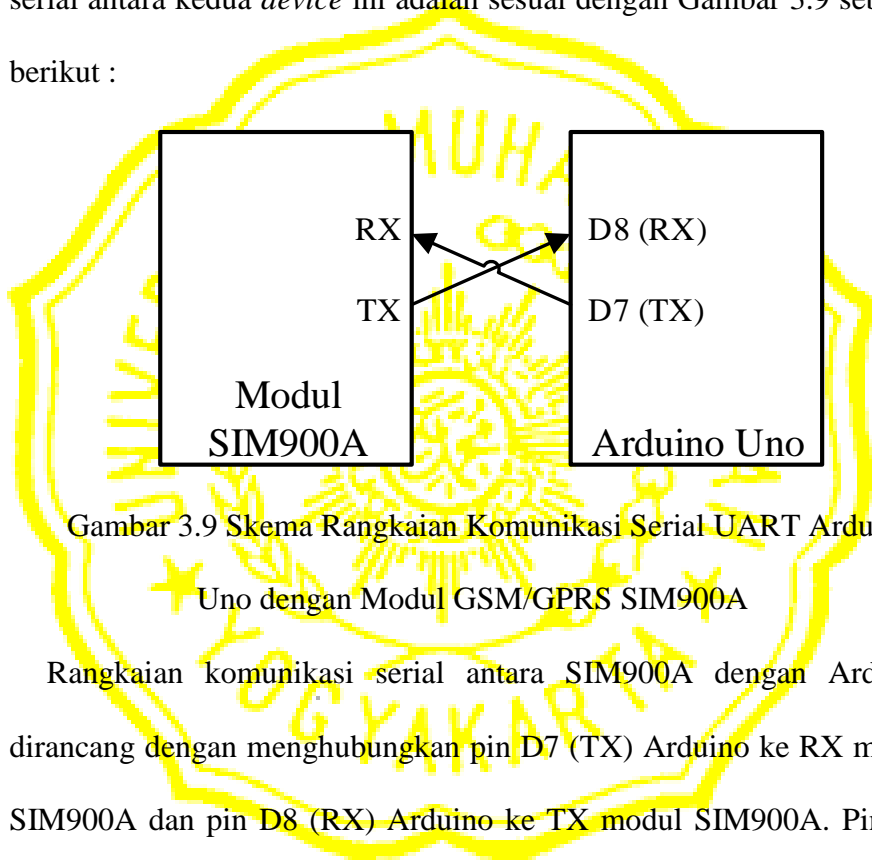
Gambar 3.8 Rangkaian Koneksi Sensor BME280 dengan Arduino Uno

3.3.3. Perancangan Komunikasi Serial UART

Perancangan komunikasi serial UART dalam sistem ini berfungsi untuk menghubungkan antara modul GSM/GPRS SIM900A dengan Arduino Uno. Arduino Uno akan mengirimkan data dari sensor ke modul SIM900A untuk kemudian akan dikirimkan ke *database* MySQL. Penggunaan komunikasi ini memiliki tujuan agar modul GSM/GPRS

dapat melakukan proses pengiriman data sensor ke *database* sesuai dengan perintah yang diinstruksikan dalam program di Arduino IDE.

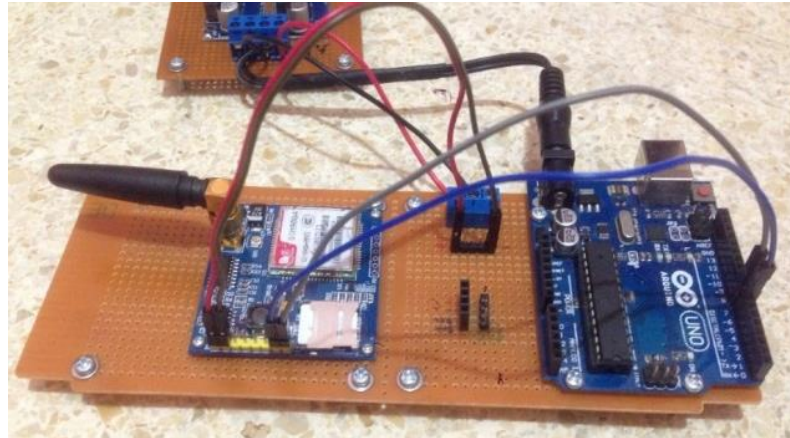
Pin yang digunakan untuk komunikasi serial pada Arduino Uno yaitu pada pin D7 (RX) dan pin D8 (TX). Sementara itu, pin yang digunakan untuk komunikasi serial pada modul GSM/GPRS SIM900A yaitu pada pin RX dan pin TX. Skema rangkaian untuk perancangan komunikasi serial antara kedua *device* ini adalah sesuai dengan Gambar 3.9 sebagai berikut :



Gambar 3.9 Skema Rangkaian Komunikasi Serial UART Arduino Uno dengan Modul GSM/GPRS SIM900A

Rangkaian komunikasi serial antara SIM900A dengan Arduino dirancang dengan menghubungkan pin D7 (TX) Arduino ke RX modul SIM900A dan pin D8 (RX) Arduino ke TX modul SIM900A. Pin D7 Arduino sebagai TX akan mengirimkan data ke RX modul SIM900A, kemudian setelah modul SIM900A menerima data dari TX Arduino akan mengirimkan data dari TX modul SIM900A ke pin D8 (RX) Arduino.

Sementara itu, hasil dari rangkaian komunikasi serial antara modul SIM900A dengan Arduino Uno yang telah dibuat adalah sesuai Gambar 3.10 sebagai berikut :



Gambar 3.10 Rangkaian Komunikasi Serial UART Arduino Uno dengan Modul GSM/GPRS SIM900A

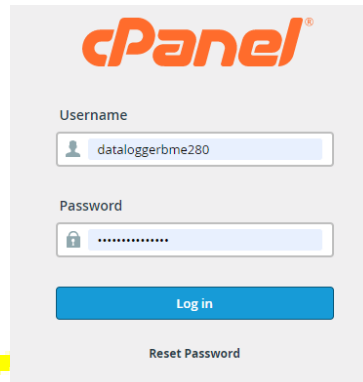
3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak dalam penelitian tugas akhir ini terdiri dari pembuatan *database* MySQL, pembuatan program untuk koneksi basis data dengan *file* PHP, dan pembuatan program untuk *sensing* dan mentransfer data untuk disimpan ke basis data. Proses perancangan perangkat lunak dapat dijelaskan sebagai berikut :

3.4.1. Pembuatan *Database* MySQL

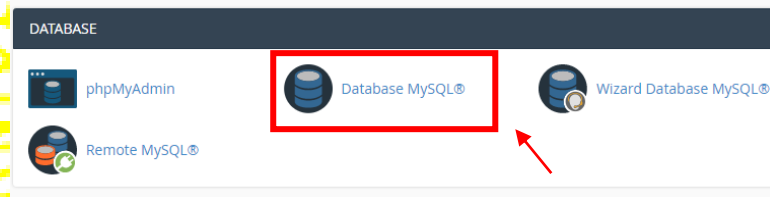
Pembuatan *database* membutuhkan *domain* dan *hosting*. Layanan *hosting* yang digunakan adalah layanan WebHosting dari penyedia layanan hosting Telkom. Pembuatan *database* dilakukan dengan membeli layanan WebHosting yang dapat aktif per – satu bulan dengan besar kuota 300MB dan mendaftarkan *domain* baru yang dapat aktif per – satu tahunnya. Setelah mendapatkan layanan WebHosting, maka dilakukan pembuatan *database* melalui menu MySQL Database dan pembuatan tabel pada menu PHPMyAdmin pada CPanel. Secara rinci, pembuatan *database* MySQL adalah sebagai berikut :

1. Melakukan *login* ke Cpanel utama dalam CPANEL. Seperti pada Gambar 3.11 sebagai berikut :



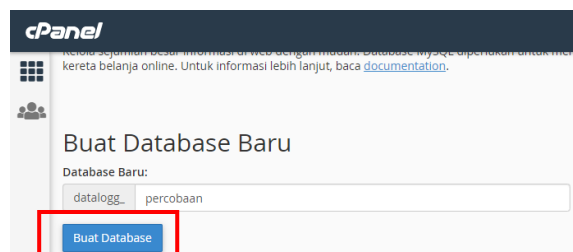
Gambar 3.11 Halaman Login ke CPANELUtama

2. Masuk ke menu Database, kemudian memilih DatabaseMySQL® sesuai pada Gambar 3.12 sebagai berikut:



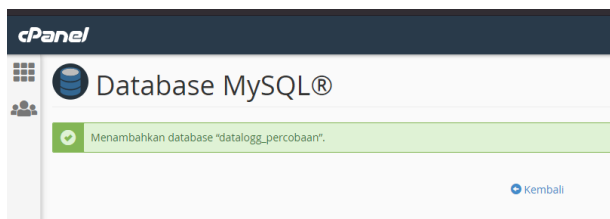
Gambar 3.12 Memilih Menu Database MySQL®

3. Memasukkan nama *database* yang akan digunakan pada kotak dialog yang disediakan, kemudian meng-klik tombol 'Buat Database'. Sebagaimana pada Gambar 3.13 sebagai berikut :



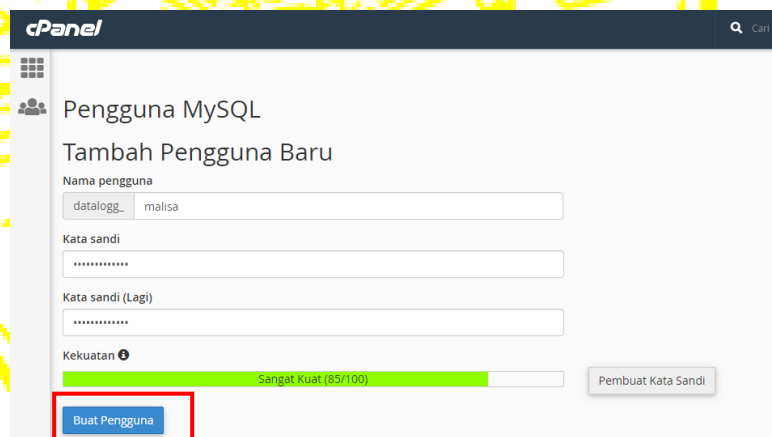
Gambar 3.13 Proses Memasukkan Nama *Database*

Ketika berhasil maka cPanel akan merespon dengan memberi notifikasi seperti pada Gambar 3.14 berikut ini :



Gambar 3.14 Respon Berhasil Menambahkan *Database* MySQL

4. Selanjutnya yaitu menambahkan pengguna berhak untuk *database* yang baru saja ditambahkan pada kotak dialog yang disediakan. Kemudian meng-klik 'Buat Pengguna' seperti Gambar 3.15 berikut ini:



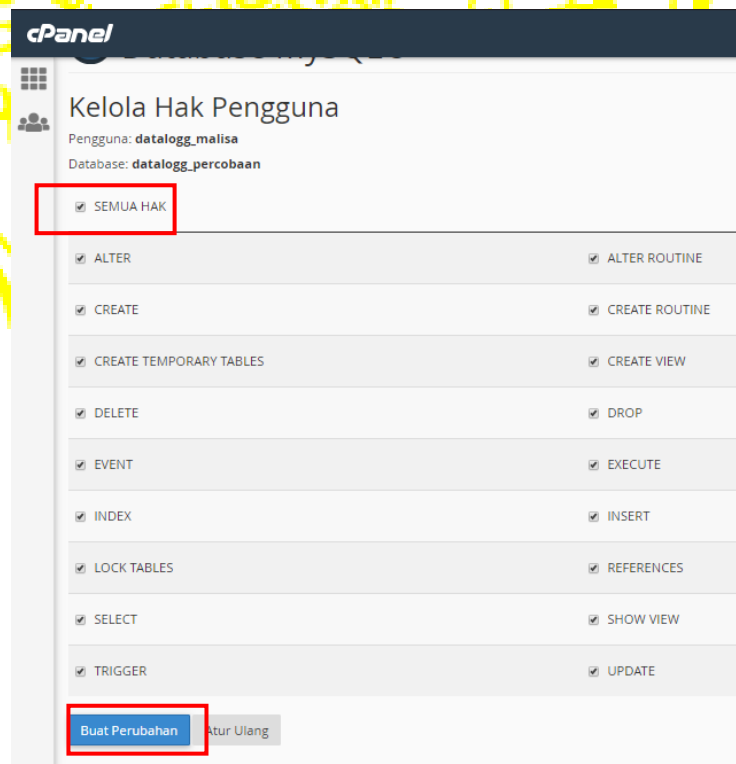
Gambar 3.15 Menambahkan Pengguna Baru untuk *Database* MySQL

5. Setelah itu, menambahkan pengguna baru tersebut sebagai pengguna *database* yang baru saja dibuat dan meng-klik tombol "Tambah", seperti pada Gambar 3.16 di bawah ini :



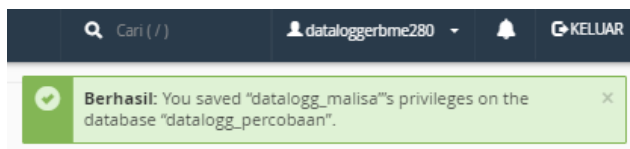
Gambar 3.16 Menambahkan Pengguna ke Database

6. Setelah tombol “Tambah” di-klik maka akan muncul permintaan “Kelola Hak Pengguna” untuk memilih hak kelola database. Maka yang dipilih adalah semua hak kelola dengan ceklis “Semua Hak” lalu meng-klik tombol “Buat Perubahan”, sesuai Gambar 3.17 sebagai berikut :



Gambar 3.17 Memilih Semua Hak Kelola untuk Database

Ketika berhasil, maka CPanel akan merespon dengan memberi notifikasi sesuai Gambar 3.18 sebagai berikut :



Gambar 3.18 Respon Berhasil Membuat Perubahan Hak Pengguna Database

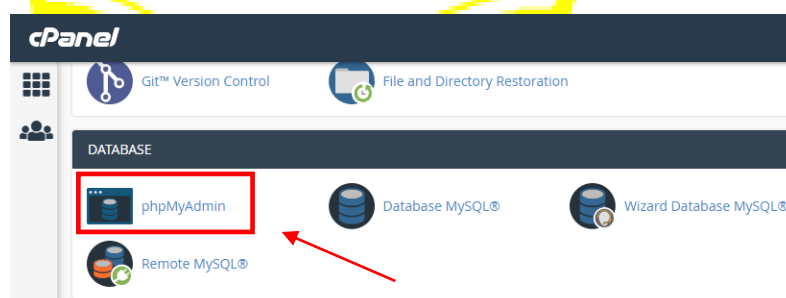
7. Maka proses pembuatan database di Database MySQL® sudah selesai dan dapat dilihat dalam daftar database yang ada, seperti Gambar 3.19 di bawah ini :

A screenshot of the MySQL database list in cPanel. The title is "Database Saat Ini". There is a search bar with "Cari" and a "Buka" button. Below the search bar is a table with the following columns: "Database", "Ukuran", "Pengguna Berhak", and "Tindakan". The table contains two rows of data. The second row, "datalogg_percobaan", is highlighted with a red border. The "Tindakan" column for each row contains "Ubah nama" and "Hapus" links.

Database	Ukuran	Pengguna Berhak	Tindakan
datalogg_bme280	0.00 MB	datalogg_malisa	Ubah nama Hapus
datalogg_percobaan	0.00 MB	datalogg_malisa	Ubah nama Hapus

Gambar 3.19 Daftar Database di Database MySQL®

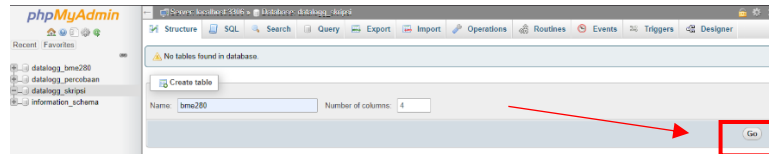
8. Proses selanjutnya yaitu pembuatan tabel untuk database di PHPMyAdmin. Langkah pertama yaitu masuk ke PHP MyAdmin, seperti yang ditunjukkan melalui Gambar 3.20 sebagai berikut :



Gambar 3.20 Memasuki PHPMyAdmin

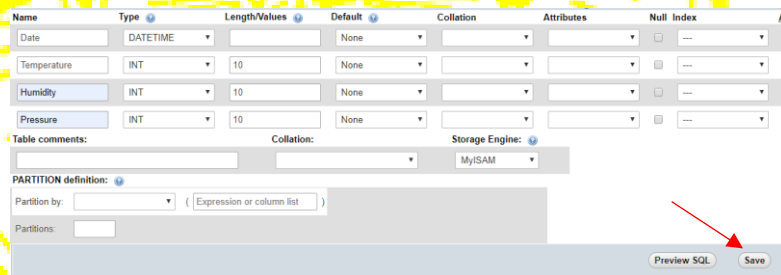
9. Selanjutnya yaitu mengisi nama tabel pada kotak "Create Table" dan menentukan jumlah kolom tabel pada "Column"

kemudian meng-klik tombol “Go”, seperti yang ditunjukkan dalam Gambar 3.21 sebagai berikut :



Gambar 3.21 Memasukkan Nama Tabel dan Jumlah Kolom Tabel

10. Berikutnya yaitu mengisi variabel yang dibutuhkan dalam tabel diantaranya yaitu *date*, *temperature*, *humidity*, dan *pressure*. Variabel *date* dibuat untuk mengetahui tanggal dan waktu saat *input* data sensor dilakukan. Pengisian komposisi yang dibutuhkan dalam tabel ditunjukkan melalui Gambar 3.22 sebagai berikut :



Gambar 3.22 Memasukkan Komposisi untuk Variabel dalam Tabel Database

11. Setelah komposisi yang dibutuhkan telah dimasukkan maka selanjutnya yaitu meng-klik tombol “Save” sehingga tabel *database* telah selesai dibuat dan tersimpan dalam CPanel.

3.4.2. Koneksi PHP ke Database MySQL

Setelah *database* selesai dibuat yang dilengkapi dengan tabelnya, maka selanjutnya diperlukan koneksi antara *file* PHP dengan *database*

MySQL agar data sensor dapat dimasukkan ke *database* dan dapat dipantau di CPanel dan dapat diakses melalui *browser*. Pembuatan program PHP menggunakan Visual Studio Code yang mana program terdiri atas 2 program inti. Pertama yaitu program koneksi ke *database* MySQL dan yang kedua adalah program memasukkan data sensor ke tabel di dalam *database* MySQL. Kedua program inti tersebut dibuat ke dalam satu *file* PHP yang disimpan di dalam XAMPP Control Panel.

Program tersebut adalah sebagai berikut :

```
<?php
$servername = "nama server";
$dbname = "nama database";
$username = "nama user database";
$password = "password database";
$conn = mysqli_connect($servername, $username, $password,
$dbname);
// mengecek koneksi
if (!$conn) {
    die("Koneksi gagal: ". mysqli_connect_error());
}
echo "Koneksi berhasil";
date_default_timezone_set('Asia/Jakarta');
$date=date('Y-m-d H:i:s');
$sql = "INSERT INTO data (Date, temperature, humidity, pressure) VALUES
('$date', '".$_GET["temperature"]."', '".$_GET["humidity"]."', '".$_GET["pres
sure"]."'";
if(mysqli_query($conn,$sql))
{
    echo ("OK" );
}
else
{
    echo "Fail: ". $sql. "<br/>". mysqli_error($conn);
}
mysqli_close($conn);
?>
```

Penjelasan masing-masing *code* di atas yaitu :

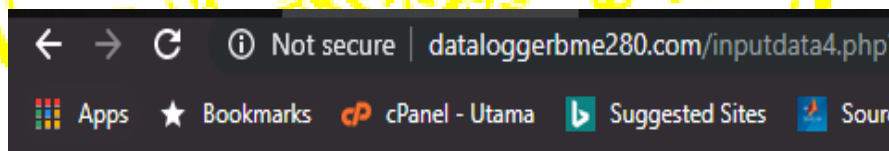
1. `$servername`, `$dbname`, `$username`, dan `$password` merupakan program untuk mendeklarasikan variabel yang akan dihubungkan,

yaitu *server*, nama *database*, nama *user/pengguna database*, dan *password* pengguna *database* yang akan dituju untuk dihubungkan dengan MySQL.

2. `$conn = mysqli_connect($servername, $username, $password, $database)`. Di dalam *code* ini terdapat fungsi utama yang digunakan dalam program utama yaitu “`mysqli_connect`” untuk membuka koneksi baru ke *server* MySQL. Huruf “i” pada “`mysqli_connect`” merupakan versi *update* dari MySQL atau yang sering disebut juga *improved* dari MySQL. Kemudian untuk `$servername`, `$username`, `$password`, dan `$database` merupakan parameter-parameter yang berfungsi sebagai spesifikasi untuk koneksi.
3. Fungsi yang terdapat dalam “if ... Else” dalam program ini bekerja dengan cara “if” akan mengeksekusi jika keadaan adalah benar dan “else” akan mengeksekusi jika keadaan salah.
4. Kode “`date_default_timezone_set('AsiaJakarta')`” merupakan kode untuk pengaturan zona waktu secara otomatis dimana zona waktu yang diatur dalam program ini mengikuti waktu di Jakarta.
5. Kode `$date=date('Y-m-d H:i:s')` merupakan pengaturan tanggal dan waktu. Tanggal di atur dengan format “*Year*” yaitu tahun, lalu “*month*” yaitu bulan dan “*day*”. Format ini berarti penanggalan berurutan dari tahun-bulan-hari. Format waktu atur dengan format “H:i:s” yaitu *hour:minute:second*” yang artinya berurutan dari jam-menit-detik.

6. Kode "INSERT INTO data (Date, temperature, humidity, pressure)" dalam "\$sql=" merupakan kode untuk memasukkan parameter yang berada dalam tanda kurung (...) ke dalam tabel "data". Kemudian untuk kode "VALUES ('\$date', '".\$_GET["temperature"]."', '".\$_GET["humidity"]."', '".\$_GET["pressure"]."')" dalam "\$sql=" merupakan perintah untuk memasukkan nilai atau *value* dari variabel yang dibutuhkan di tabel *database* MySQL dengan metode GET yang mana metode ditunjukkan dengan kode "\$_GET".

Jika *file* PHP untuk *input* data sudah terhubung dengan database maka dapat dicek melalui *browser* dengan alamat pencarian yaitu <http://dataloggerbme280.com/inputdata4.php> dan akan muncul respon "OK" pada jendela *website* seperti pada Gambar 3.23 berikut ini :

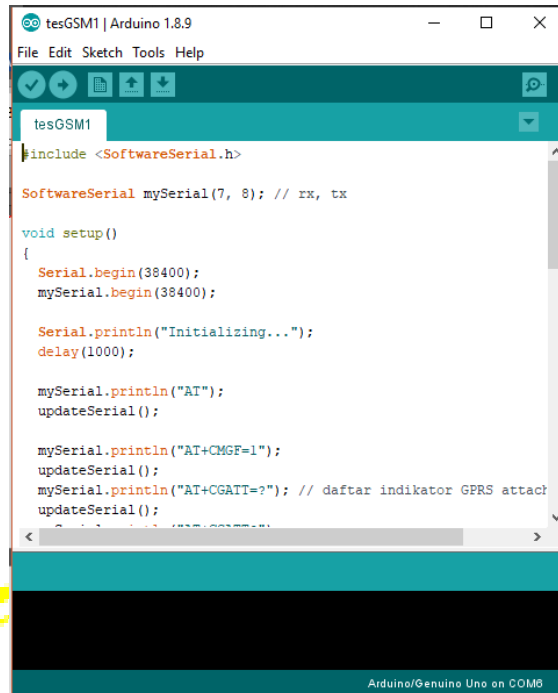


Koneksi berhasilOK

Gambar 3.23 Koneksi *File* PHP untuk *Input* Data Berhasil

3.5 Perancangan Program Arduino

Program yang dirancang di dalam tugas akhir ini terdiri atas dua program yaitu program untuk menguji kondisi sensor BME280 dan program untuk mengirimkan data dari sensor BME280 ke *server database* MySQL. Kedua program tersebut dilakukan dengan menggunakan Arduino IDE versi 1.8.9 dengan antarmuka seperti pada Gambar 3.24 berikut ini :



Gambar 3.24 Antarmuka Arduino IDE 1.8.9

3.5.1. Perancangan Program Arduino untuk Pengujian Sensor BME280

Program Arduino pertama dirancang dalam penelitian tugas akhir ini adalah program untuk pengujian sensor BME280 dengan tujuan agar sensor dapat membaca nilai dari keadaan sekitar yaitu tingkat temperatur, kelembaban, dan tekanan udara sehingga akan didapatkan hasil bahwa sensor ini berfungsi dengan baik. Program yang telah dirancang adalah sebagai berikut :

```

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme;
void setup() {
  Serial.begin(9600);

  if (!bme.begin(0x76)) {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
}

```

```

    }
  }
  void loop() {
    Serial.print("Suhu = ");
    Serial.print(bme.readTemperature());
    Serial.println("*C");

    Serial.print("Kelembaban = ");
    Serial.print(bme.readHumidity());
    Serial.println("%");

    Serial.print("Tekanan = ");
    Serial.print(bme.readPressure() / 100.0F);
    Serial.println("hPa");

    Serial.println();
    delay(5000);
  }

```

Program di atas dapat dijelaskan sebagai bahwa program yang dirancang untuk menguji sensor BME280 membutuhkan tiga pustaka yang dapat dilihat pada bagian “#include” diantaranya yaitu *Wire.h*, *AdafruitSensor.h* dan *Adafruit_BME280.h*. Pustaka *Wire.h* merupakan pustaka untuk komunikasi TWI yaitu SDA dan SCL. Untuk pengaturan parameter pengukuran tekanan udara didefinisikan pada bagian “#define”. Kemudian untuk penginisialan sensor bme280 diinisialkan dengan “bme” pada bagian “Adafruit_BME280 bme”.

Perintah untuk komunikasi serial di dalam program ini ditunjukkan dalam kode “Serial.begin(9600)”. Angka 9600 menunjukkan kecepatan transmisi data melalui komunikasi serial dengan kecepatan 9600 bit untuk setiap detik.

Terdapat dua program inti yang dijalankan di dalam program ini yaitu program di dalam bagian “void setup ()” dan program yang berada di dalam “void loop()”. Program yang berada di dalam “void setup()”

akan di jalankan satu kali, sedangkan program yang berada di dalam “void loop()” akan di jalan secara berulang sesuai dengan *delay* yang telah ditentukan yaitu 5000ms atau 5 detik pada kode “delay(5000)”.

Di dalam bagian “void loop()” terdapat perintah “Serial.print” dan “Serial.println” yang memiliki fungsi yang hamper sama. Keduanya berfungsi sebagai perintah untuk menampilkan data ke serial monitor, hanya saja jika “Serial.println” akan menampilkan ke dalam baris berikutnya.

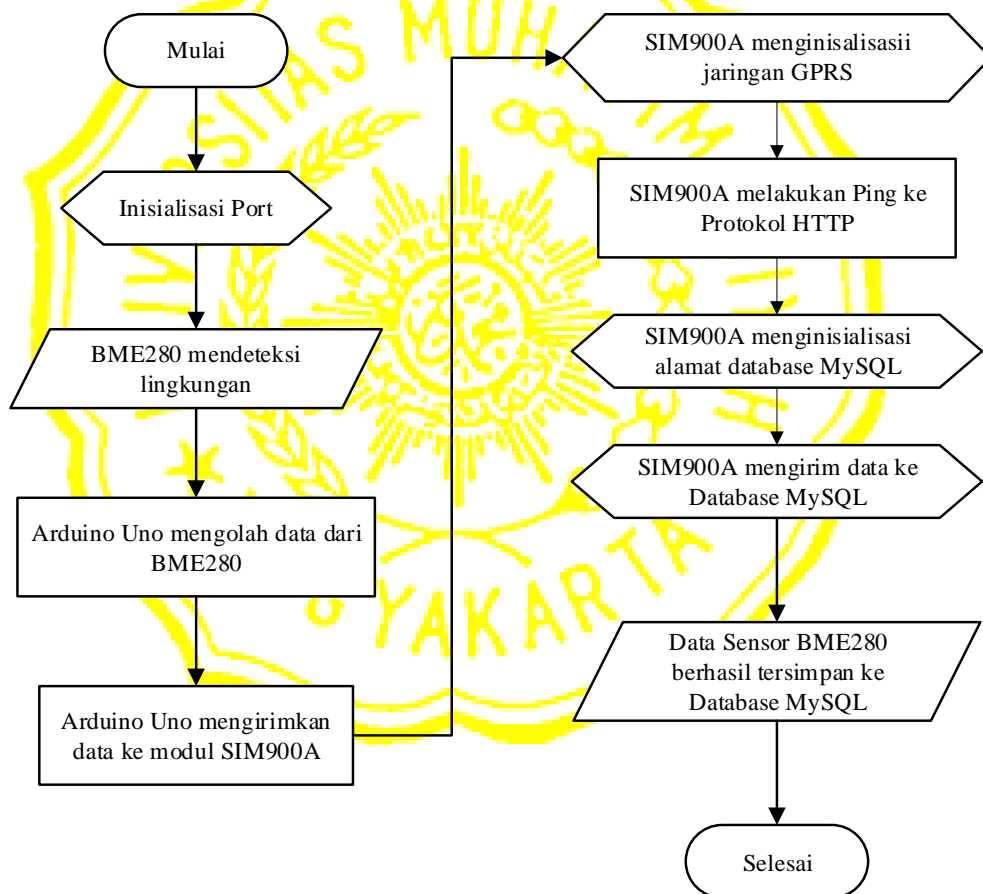
Kode “bme.readTemperature()”, “bme.readHumidity()”, dan “bme.readPressure()/100.0F” merupakan perintah untuk sensor BME280 membaca nilai dari masing-masing variabel yaitu *temperature* atau suhu, *humidity* atau kelembaban udara, dan *pressure* atau tekanan. Di dalam program ini terdapat aturan yang ditentukan di dalam kode untuk sensor membaca nilai dari tekanan udara yaitu hasil pembacaan akan dibagi dengan 100. Aturan ini ditunjukkan melalui kode “/100.0F”. Aturan ini ditentukan agar hasil pembacaan sesuai dalam satuan hPa yaitu hektoPascal dari satuan nilai sebenarnya yang terukur oleh sensor BME280 yaitu Pascal (Pa).

3.5.2. Perancangan Program untuk Pengiriman Data Sensor ke Server

Database MySQL

Program ke-dua yang dirancang di dalam penelitian tugas akhir ini adalah program untuk transmisi data sensor BME280 ke *server database* MySQL melalui modul GSM/GPRS SIM900A.

Program ini menggunakan satu masukan (*input*) dan satu keluaran (*output*) yang akan diolah oleh Arduino Uno. Masukan yang dimaksud yaitu sensor BME280 dan keluaran berupa data sensor tersebut yang kemudian akan dikirim ke *database* oleh modul GSM/GPRS SIM900A. Sebelum merancang program di Arduino IDE, dibuat *flowchart* untuk memudahkan dalam pembuatan program dengan memperjelas alur *step-by-step* jalannya sebuah program. *Flowchart* program yang dibuat adalah sesuai dengan Gambar 3.25 sebagai berikut :



Gambar 3.25 *Flowchart* Program Arduino

Dari *flowchart* di atas dapat dijelaskan yaitu program dimulai dengan inisialisasi perangkat dan komunikasi yang digunakan dimana perangkat yang digunakan adalah sensor BME280 dan modul SIM900A

dengan komunikasi serial *asynchronous*. Ketika sensor telah teridentifikasi aktif, maka sensor akan mendeteksi tingkat suhu, kelembaban, dan tekanan dari lingkungan yang dijadikan objek pengukuran. Data yang berhasil diambil oleh sensor BME280 akan diolah oleh Arduino Uno untuk kemudian akan dikirimkan ke modul SIM900A. Sebelum mengirimkan data ke *server database*, SIM900A harus melakukan inisialisasi konektivitas jaringan GPRS dari penyedia layanan seluler, yaitu Telkomsel. Setelah mendapatkan jaringan GPRS maka SIM900A akan melakukan tahap konektivitas ke HTTP dengan melakukan *ping* ke HTTP. Ketika *ping* tersebut berhasil, selanjutnya adalah proses inisialisasi alamat *database* yang akan dituju sehingga jalur ke *server database* yang dituju tersebut dapat tersambungkan. Setelah selesai melakukan inisialisasi alamat *database*, SIM900A akan mengirimkan data dari sensor BME280 ke *server database* MySQL dengan jeda waktu (*delay*) tertentu. Apabila data berhasil dikirimkan dan tersimpan maka data tersebut dapat dilihat melalui tabel di *server database* MySQL. Program ini dirancang sebagai program yang satu kali jalan, artinya program akan berjalan secara keseluruhan tanpa perulangan pada tahap tertentu untuk kondisi yang tidak diinginkan.

Program untuk mengirimkan data sensor BME280 ke *server database* MySQL ini memiliki dua bagian utama yaitu program yang akan dieksekusi satu kali di dalam bagian “void setup()” dan program yang akan dieksekusi secara berulang di dalam bagian “void loop()”. Perintah yang terdapat di dalam “void setup()” yaitu perintah untuk

komunikasi serial modul SIM900A, komunikasi serial sensor BME280 dan perintah untuk koneksi SIM900A dengan jaringan GPRS dan HTTP. Sedangkan perintah yang terdapat di dalam bagian “void loop()” terdiri atas perintah untuk mentransmisikan data dari sensor BME280 ke alamat tabel di dalam *server database* MySQL.

Di dalam program yang dibuat terdapat perintah *float* untuk mendefinisikan nilai dari variabel yang diinisialkan ke dalam nilai desimal. Variabel yang diinisialkan adalah variabel yang diukur oleh sensor BME280 yaitu *temperature* dengan “suhu”, *humidity* dengan “kelembaban”, dan *pressure* dengan “tekanan”.

Perintah AT untuk pengaturan modul SIM900A terdapat pada perintah “mySerial.println(“AT+.....”)”. Kode “println” mengartikan bahwa respon dari perintah yang diinstruksikan akan di tampilkan ke baris serikutnya setelah perintah AT ditampilkan. Format penulisan kode perintah AT di dalam program ini dibagi atas dua bagian yaitu perintah AT untuk koneksi GPRS sampai dengan instruksi untuk memulai koneksi HTTP yang akan dieksekusi satu kali dalam bagian “void setup()” dan perintah AT untuk pengaturan identitas pembawa ke HTTP sampai intruksi untuk mengirimkan data sensor ke alamat tabel di dalam *database* MySQL yang akan dieksekusi secara berulang pada bagian “void loop()”.

3.6 Perlakuan Pengujian

Perlakuan pengujian dibuat untuk mengetahui secara jelas mengenai cara kerja dari alat yang telah dibuat. Dalam rangka mendapatkan

keluaran yang diharapkan, maka diperlukan skenario pengujian alat yang bertahap di dalam tugas akhir ini. antara lain sebagai berikut :

3.6.1. Perlakuan Pengujian Database MYSQL

Pengujian *database* MySQL dilakukan dengan tujuan untuk memperoleh kesiapan dari *database* untuk dapat dijadikan sebagai tempat untuk menyimpan data sensor. Pengujian ini akan didapatkan hasil apakah *database* MySQL yang telah dibuat sudah siap dijadikan tempat penyimpanan data sensor atau belum. Pengujian ini dilakukan dengan tahapan sebagai berikut :

1. Memasukkan alamat *domain*, direktori *file*, dan nilai dari variabel yang akan dituju. Alamat *domain*, direktori *file*, dan nilai dari variabel yang akan dituju adalah sebagai berikut :

Alamat *domain* : dataloggerbme280.com

Direktori *file* : inputdata4.php

Nilai dari variabel : temperature sebesar 21, humidity sebesar 45,
dan pressure sebesar 34

2. Mengecek respon dari *browser* apakah koneksi dan *input* data yang dilakukan berhasil atau tidak.
3. Meninjau tabel *database* MySQL apakah sudah ter-*input* data untuk masing-masing variabel atau belum.

3.6.2. Perlakuan Pengujian Sensor BME280

Pengujian ini merupakan pengujian untuk mengetahui apakah sensor BME280 dapat berfungsi atau tidak sehingga akan didapatkan

hasil sensor siap untuk digunakan sebagai komponen untuk perancangan *data logger* dalam penelitian tugas akhir ini.

Pengujian sensor BME280 dilakukan bersama dengan thermometer dinding dengan cara menempatkan rangkaian sensor BME280 dan termometer pada tempat dan waktu yang sama sehingga akan didapatkan hasil apakah sensor BME280 dapat membaca nilai keadaan atau tidak. Pengujian ini dilakukan dengan tahapan sebagai berikut:

1. Menghubungkan kabel *jumper* sesuai dengan skema rangkaian BME280 ke Arduino Uno.
2. Menghubungkan *downloader* Arduino Uno ke laptop.
3. Mengatur lama waktu pengujian sensor dan thermometer.
4. Meng-*upload* program pengujian sensor BME280 di Arduino IDE.

3.6.3. Perlakuan Pengujian Pengiriman Data Ke Server *Database* MySQL menggunakan USB to TTL

Pengujian ini merupakan pengujian untuk mengetahui tahapan-tahapan perintah AT yang harus dimasukkan ke dalam program Arduino nantinya untuk mengirimkan data ke *server database* MySQL. Pengujian ini perlu dilakukan sebagai penanda jika perintah AT dapat merespon “OK” melalui USB to TTL maka seharusnya perintah-perintah AT tersebut juga dapat berfungsi dengan baik jika dijalankan melalui Arduino Uno, hanya saja format penulisan kode program yang berbeda.

Sebelum pengujian dapat dilakukan, dibutuhkan tahapan untuk pengecekan apakah *driver* USB to TTL sudah ter-*install* di dalam laptop dan beberapa pengaturan untuk serial monitor. Pengujian ini akan memanfaatkan aplikasi RealTerm Serial Capture Program sebagai media untuk berkomunikasi dengan modul GSM/GPRS SIM900A. Tahapan-tahapan yang dimaksud dapat dijelaskan sebagai berikut :

1. Menghubungkan rangkaian USB to TTL dan modul GSM/GPRS SIM900A ke laptop seperti pada Gambar 3.26 di bawah ini :

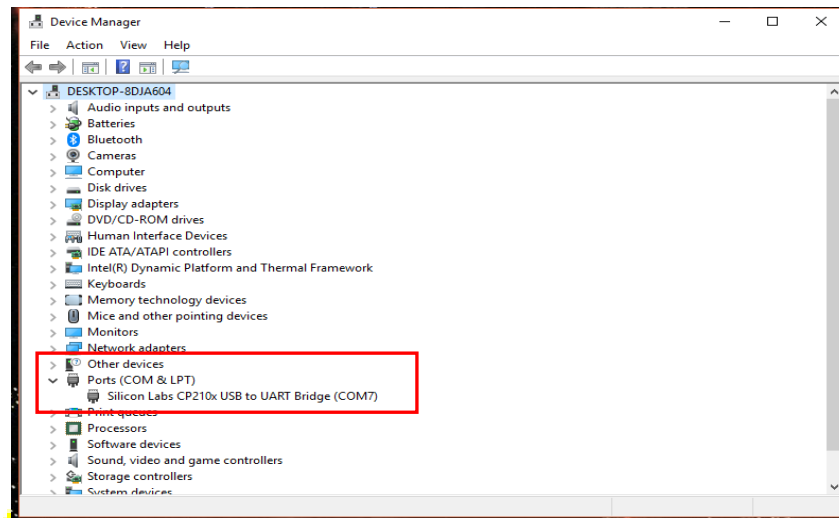


Gambar 3.26 Menghubungkan Modul GSM dan USB to TTL ke Laptop

2. Mengecek *driver* USB to TTL di *Device Manager*

Laptop yang digunakan dalam penelitian ini sudah ter-*install driver* untuk USB to TTL di *port* COM7 sebelumnya sehingga tidak memerlukan instalasi *driver* kembali. *Driver* tersebut dapat

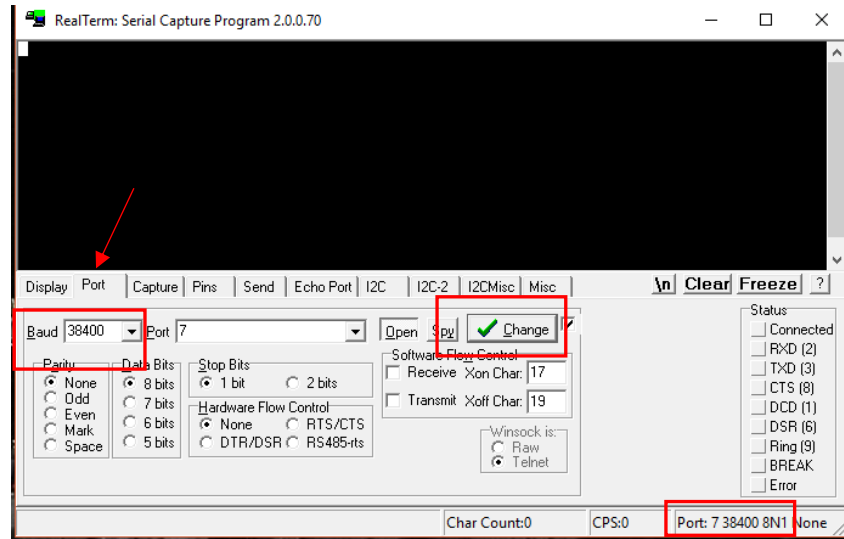
terlihat di daftar “Port(COM&UTP)” di dalam *device manager* seperti pada Gambar 3.14 sebagai berikut :



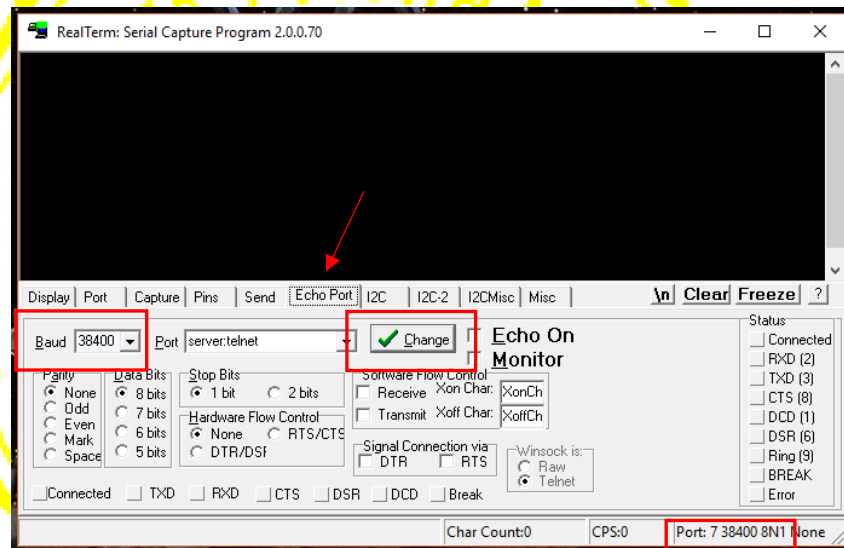
Gambar 3.27 Driver USB to TTL Sudah Tersedia di Laptop

3. Mengatur *baudrate* di RealTerm

Langkah berikutnya yaitu mengatur *baudrate* di “Port” dan “Echo Port”. *Baudrate* di kedua *port* tersebut harus sama karena apabila tidak sama maka perintah AT tidak dapat merespon OK, dengan kata lain tidak bekerja. Dalam mengatur *baudrate* ini dilakukan dengan memasukkan nilai *baudrate* yang dapat membuat modul SIM900A bekerja, kemudian mengklik tombol “Change” agar *baudrate* dapat ter-*setting*. Pengaturan *baudrate* di kedua *port* yang telah disebutkan sebelumnya dapat ditunjukkan melalui kotak bergaris warna merah pada Gambar 3.28 dan 3.29 sebagai berikut :



Gambar 3.28 Pengaturan *Baudrate* di Port

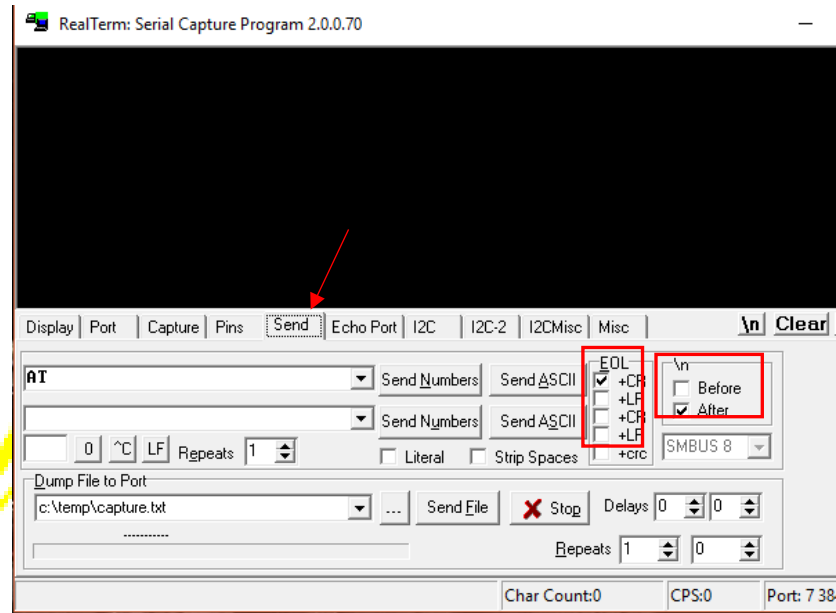


Gambar 3.29 Pengaturan *Baudrate* di Echo Port

4. Mengatur baris teks

Pengaturan baris teks bertujuan agar hasil dari respon perintah AT dapat ditampilkan dengan rapi dan jelas. Pengaturan dilakukan dengan masuk ke menu Send lalu mencentang “+CR” di bagian “EOL” dan mencentang “After” di bagian “\n”. Hal ini berarti respon dari perintah AT akan berada pada baris selanjutnya atau di bawahnya perintah AT, begitu pula dengan perintah-perintah

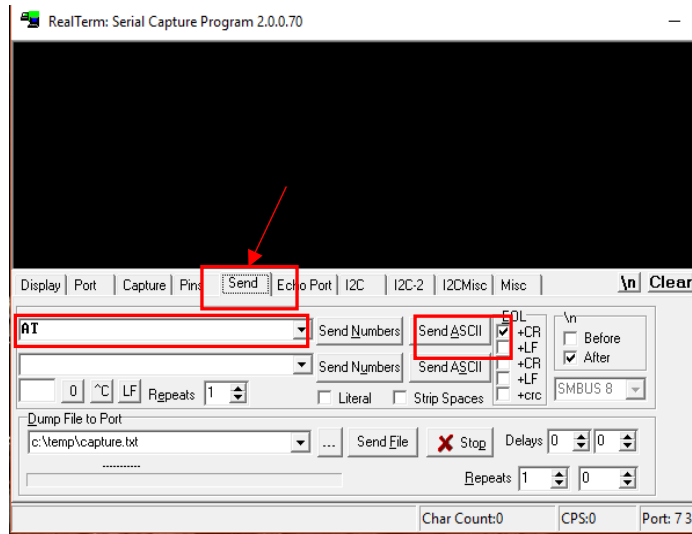
AT yang selanjutnya juga akan ditampilkan di baris berikutnya. Pengaturan ini ditunjukkan melalui kotak bergaris warna merah pada Gambar 3.30 sebagai berikut :



Gambar 3.30 Pengaturan Baris Teks

5. Mengirim perintah AT

Proses pengiriman perintah AT di RealTerm dilakukan di menu Send dengan mengetikkan perintah AT di kotak pengetikan yang disediakan, lalu mengklik tombol “Send ASCII”. Hal ini berarti perintah yang dikirimkan akan diproses dengan menggunakan aturan ASCII. Proses pengiriman perintah AT ditunjukkan melalui kotak bergaris warna merah pada Gambar 3.31 di bawah ini:



Gambar 3.31 Pengiriman Perintah AT di RealTerm

Setelah pengaturan selesai dilakukan maka tahap berikutnya yang dilakukan adalah memasukkan perintah-perintah AT yang dibutuhkan. Tahapan-tahapan perintah AT dalam hal ini memiliki urutan tertentu agar dapat mengirimkan data ke dalam *server database* MySQL. Hasil urutan tahapan perintah AT yang harus dimasukkan ke dalam program Arduino akan dijelaskan di dalam Bab IV.

3.6.4. Perlakuan Pengujian Pengiriman Data Sensor Ke Server Database MySQL Menggunakan Arduino Uno Tanpa Sensor BME280

Pengujian ini merupakan tahapan lanjutan dari pengujian sebelumnya yang mana setelah tahapan perintah AT didapatkan melalui pengujian dengan USB to TTL maka selanjutnya memasukkan tahapan perintah AT tersebut ke dalam program di Arduino IDE.

Pengujian ini dilakukan dengan menguji program yang dibuat di dalam Arduino IDE untuk dapat mengirimkan data secara berulang sesuai dengan variabel yang ada di dalam *database* MySQL. Namun

data yang dimasukkan tersebut masih data yang telah ditentukan bukan data hasil pembacaan oleh sensor BME280.

Program yang dibuat di dalam Arduino IDE tersebut membutuhkan format penulisan tertentu agar modul GSM/GPRS SIM900A dapat mengirimkan data dari Arduino Uno yang mana jika format penulisan tersebut tidak benar maka data tidak dapat tersimpan di dalam *server database* MySQL.

Tahapan dalam pengujian ini antara lain yaitu :

1. Menghubungkan kabel *jumper* sesuai dengan skema rangkaian catu daya dan komunikasi serial modul SIM900A dengan Arduino Uno.
2. Menghubungkan *downloader* Arduino ke laptop.
3. Meng-*upload* program yang telah dibuat.
4. Meninjau tabel *database* MySQL apakah data dapat tersimpan ke *server database* MySQL atau tidak.

3.6.5. Perlakuan Pengujian Pengiriman Data Sensor Ke Server Database MySQL Menggunakan Arduino Uno Dengan Sensor BME280

Pengujian ini merupakan pengujian final yang akan dilakukan di beberapa lokasi yang berbeda dan telah ditentukan. Apabila data dari sensor BME280 dapat tersimpan di *server database* MySQL maka hal tujuan dari perancangan sistem sudah berhasil.

Tempat-tempat yang dijadikan sebagai lokasi untuk pengujian ini antara lain sebagai berikut :

1. Lobi C Gedung Siti Walidah Universitas Muhammadiyah Yogyakarta.
2. Masjid Agung Manunggal Bantul, Yogyakarta.
3. Masjid Jogokariyan Yogyakarta.
4. Masjid Gedhe Kauman Yogyakarta.
5. Grha Sabha Pramana Universitas Gadjah Mada.

Kemudian pengujian dilakukan dengan alur yang dimulai dari lokasi pertama sampai lokasi ke- lima sebagai lokasi terakhir.

Sementara itu pengujian ini dilakukan dengan tahapan sebagai berikut :

1. Me-*restart* Arduino Uno untuk mempersiapkan Arduino Uno agar mengolah data dari awal di lokasi pengujian.
2. Kemudian melakukan peninjauan ke *database* MySQL untuk mengetahui apakah data dari sensor BME280 dapat tersimpan di *server database* MySQL atau tidak.