BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab Perancangan adalah bagian yang menjelaskan bagaimana alat yang digunakan sebagai bahan penelitian yang dihubungkan menjadi satu kesatuan agar dapat menjalankan fungsinya. Bab implementasi adalah bab yang menjelaskan bagaimana implementasi beberapa program dengan fungsi-fungsi yang sudah disediakan dapat berjalan dengan baik dan akurat. Isi dari bab perancangan dan implementasi penelitian ini adalah menjelaskan secara rinci proses perancangan alat baik dari sisi perangkat keras maupun perangkat lunak. Bab ini juga menjelaskan bagaimana implementasi dari perangkat berdasarkan program. Bab implementasi akan membagi penjelasan satu per satu fungsi yang ada pada Arduino.

1.1 Perancangan Sistem

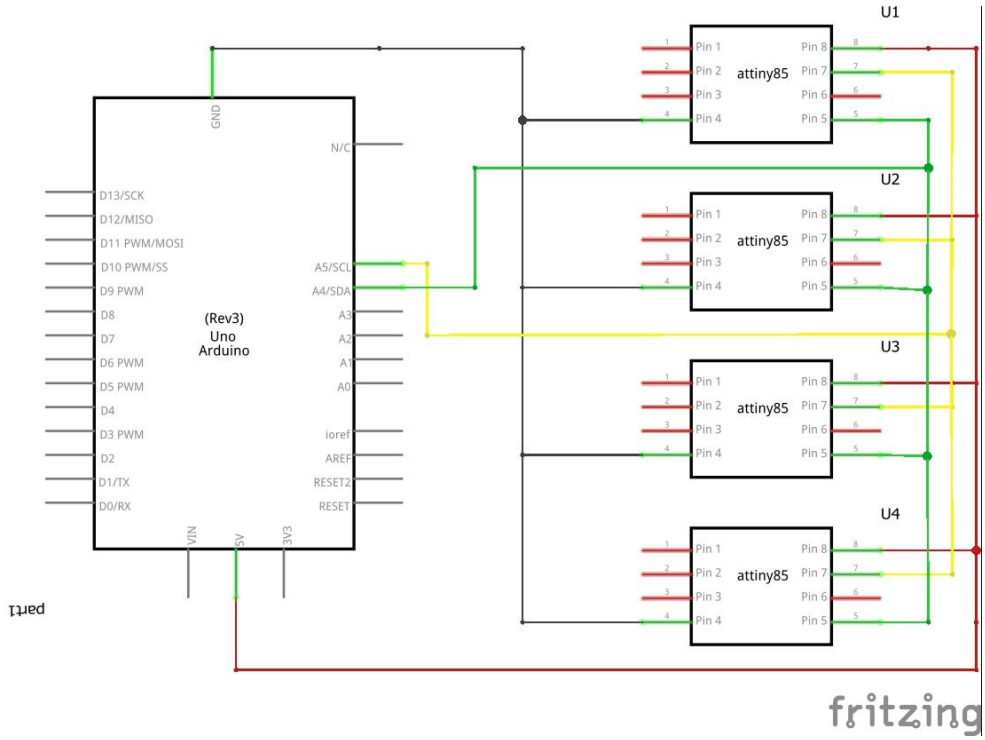
Pada sub bab perancangan sistem akan dibagi menjadi 2 bagian, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Masing-masing perancangan akan diberikan gambaran berupa skematik dan *flowchart*.

1.1.1 Perancangan Perangkat Keras

Pada Perancangan perangkat keras terbagi menjadi dua bagian, yaitu perancangan Modul Antarmuka dengan Modul Sensor dan perancangan Perangkat Modul Antarmuka Dengan Modul Komunikasi.

1.1.1.1 Perancangan Modul Antarmuka Dengan Modul Sensor

Untuk dapat membangun hubungan antara Modul Antarmuka dengan Modul Sensor yaitu menggunakan komunikasi I2C. Komunikasi berperan agar banyak sensor dapat terhubung dengan Modul Antarmuka yaitu dengan cara diparalel. Semua data sensor akan disimpan pada *Microcontroller* Attiny85, sehingga Arduino UNO langsung dihubungkan dengan pin Attiny85 tersebut. Jika terdapat banyak sensor, maka jumlah Attiny85 mengikuti jumlah sensor tersebut.



Gambar 0.1 Perancangan Modul Antarmuka Dengan Modul Sensor Secara I2C

Berdasarkan Gambar 5.1 ditunjukkan hubungan antara pin yang digunakan Arduino UNO dan Attiny85. Pin yang akan dipakai oleh Arduino UNO yaitu pin A4 (SDA) sebagai jalur sinkronisasi data dan A5 (SCL) sebagai jalur sinkronisasi *clock.*

Tabel 0.1 Koneksi Pin Arduino UNO dengan Attiny85

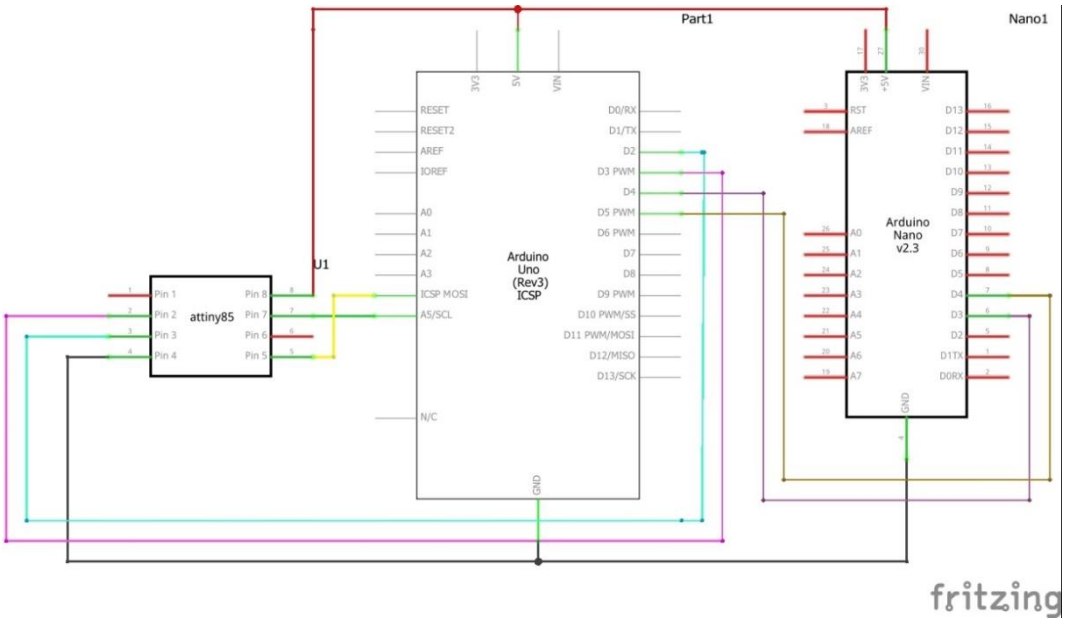
Pin Arduino UNO	Pin Attiny85
A4 (SDA)	SDA
A5 (SCL)	SCL
5V	VCC
GND	GND

Berdasarkan Tabel 5.1 di atas pin A4 (SDA) Arduino UNO akan terhubung dengan Pin SDA Attiny85. Pin A5 (SCL) Arduino UNO akan terhubung dengan Pin SCL Attiny85. Untuk menunjang daya perangkat akan disambungkannya pin VCC pada Attiny85 dengan pin 5V serta menghubungkan kedua perangkat dengan masing-masing pin *Ground*.

1.1.1.2 Perancangan Modul Antarmuka Dengan Modul Komunikasi

Untuk dapat membagun hubungan antara Modul Antarmuka dengan Modul Komunikasi menggunakan komunikasi UART. Komunikasi UART merupakan komunikasi serial, yang nantinya Modul Antarmuka bertindak sebagai *transmitter* dan Modul Komunikasi bertindak sebagai *receiver* data. Pada gambar skematik di

bawah terdapat tiga komponen yaitu *microcontroller* Arduino UNO sebagai Modul Antarmuka, *microcontroller* Arduino NANO sebagai penghubung Modul Antarmuka dengan modul Komunikasi ESP8266, serta *Microcontroller* Attiny85 sebagai penyimpan data dan penghubung antara Arduino UNO dengan Modul Komunikasi NRF24L01



Gambar 0.2 Perancangan Modul Antarmuka Dengan Modul Komunikasi Secara UART

Berdasarkan Gambar 5.2, ditunjukkan hubungan antara pin yang digunakan Arduino UNO dengan Arduino NANO sebagai koneksi dengan Modul Komunikasi ESP8266, dan Arduino UNO dengan Attiny85 yaitu komunikasi dengan NRF24L01. Pin yang akan dipakai oleh Arduino UNO dengan Atiiny85 dapat dilihat pada Tabel 5.2.

Tabel 0.2 Koneksi Pin Arduino UNO dengan Attiny85

Pin Arduino UNO	Pin Attiny85
Digital 2 (TX)	3 (RX)
Digital 3 (RX)	3 (TX)
5V	VCC
GND	GND

Pin RX dan TX yang digunkan oleh Arduino UNO adalah pin berdasarkan *SoftwareSerial* yang sudah dideklarasikan pada program. Berdasarkan Tabel 5.2 pin Digital 2 (TX) Arduino UNO akan terhubung dengan Pin 2 (RX) Attiny85. Pin 3 (RX) Arduino UNO akan terhubung dengan pin 2 (TX) Attiny85. Untuk memperoleh daya perangkat akan disambungkannya pin VCC pada Attiny85 dengan pin 5V serta menghubungkan kedua perangkat dengan masing-masing pin *Ground*.

Tabel 0.3 Koneksi Pin Arduino UNO dengan Arduino NANO

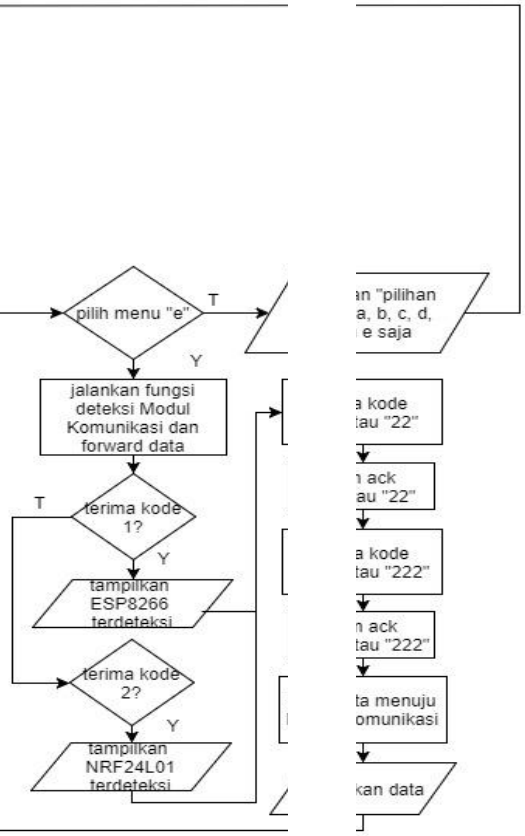
Pin Arduino UNO	Pin Arduino NANO
Digital 4 (TX)	6 (RX)
Digital 5 (RX)	7 (TX)
5V	5V
GND	GND

Pin RX dan TX yang digunkan oleh Arduino UNO adalah pin berdasarkan *SoftwareSerial* yang sudah dideklarasikan pada program. Berdasarkan tabel di atas pin Digital 4 (TX) Arduino UNO akan terhubung dengan pin digital 6 (RX) Arduino NANO. Pin 5 (RX) Arduino UNO akan terhubung dengan pin digital 7 (TX) Arduino NANO. Untuk memperoleh daya perangkat akan disambungkannya pin 5V pada kedua *Microcontroller* serta menghubungkan masing-masing pin *Ground*. Pin yang tersambung dapat dilihat pada Tabel 5.3.

1.1.2 Perancangan Perangkat Lunak

Pada sub bab ini terbagi menjadi Perancangan Modul Antarmuka Dengan Modul Sensor dan Perancangan Modul Antarmuka Dengan Modul Komunikasi. Sebagai gambaran umum dari perancangan perangkat lunak akan dijelaskan proses jalannya keseluruhan fungsi dengan menggunakan *flowchart*.

Gambar 5.3 merupakan digram alir secara keseluruhan sistem yang dirancang dengan berbagai fungsi. Program berjalan berurutan karena alur dari program adalah urut yaitu dengan memberikan *input*an “a” untuk deteksi sensor, “b” melihat spesifikasi sensor, “c ” meminta data semua sensor, “d” meminta data secara spesifik pada satu sensor, “e” untuk mendeteksi dan mem-*forward* data sensor ke Modul Komunikasi. Semua hasil proses ditampilkan pada serial monitor. Fungsi tersebut akan secara spesifik dijelaskan pada *flowchart* di bawah untuk masing-masing sub bab fungsinya.



Gambar 5.3 ***Flowchart*** **Perancangan Semua Fungsi Secara Umum**

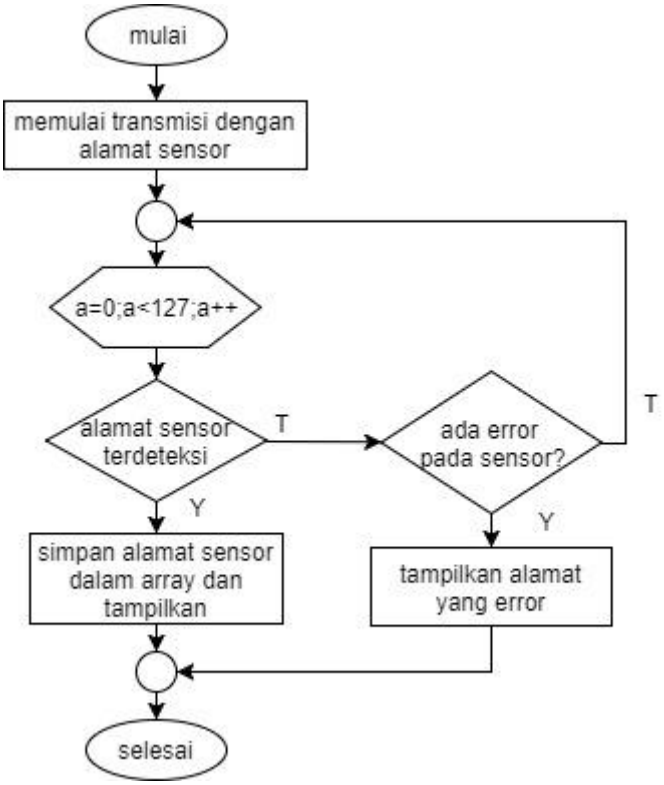
Gambar 0.3 *Flowchart* Perancangan Semua Fungsi Secara Umum

1.1.2.1 Perancangan Modul Antarmuka Dengan Modul Sensor

Perancangan Perangkat lunak Modul Antarmuka dengan Modul Sensor terbagi menjadi beberapa fungsi utama dan digunakan sebagai menu untuk *user*. Program dengan berbagai fungsi ini dijalankan pada Arduino IDE dengan meng-*install library Wired*. Proses perancangan programnya ditunjukkan pada gambar *flowchart* pada masing-masing sub bab fungsi.

a. Perancangan Fungsi Deteksi Sensor

Fungsi pertama adalah mendeteksi adanya modul sensor yang terkoneksi. Modul Antarmuka *standby* untuk menunggu modul sensor yang akan dihubungkan. Ketika ada sensor yang terhubung, maka dengan melakukan *looping* alamat 0-127 akan ditemukan alamat yang tepat, yaitu alamat sensor terhubung. Alamat tersebut ditampilkan pada serial monitor. Ketika ada sensor lain yang ditancapkan lagi maka Modul Antarmuka tetap melakukan *looping* untuk mencari alamat tersebut. Sensor yang dapat terkoneksi maksimal adalah 128 perangkat. Pada Gambar 5.4 akan ditunjukkan alur dari pendeteksian sensor melalui *flowchart*

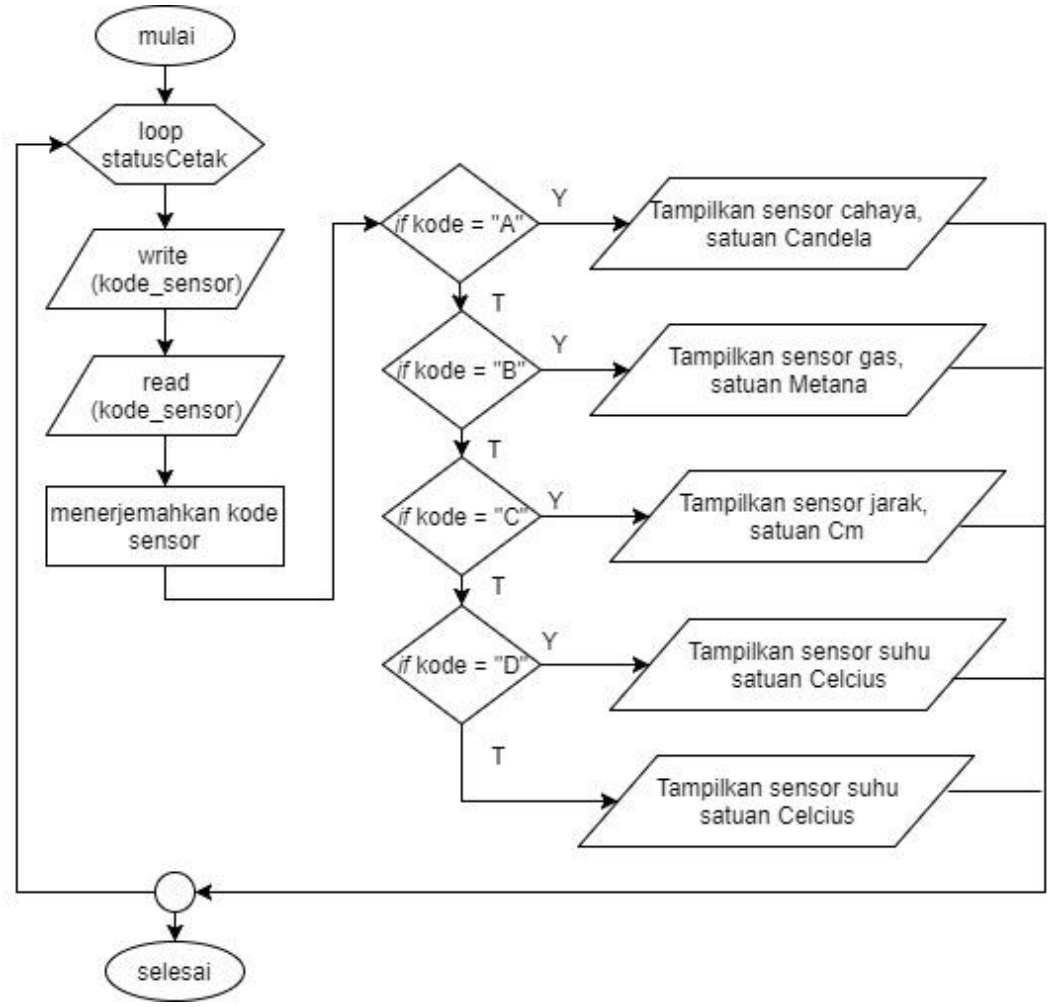


Gambar 0.4 *Flowchart* Deteksi Modul Sensor

b. Perancangan Fungsi Spesifikasi Sensor

Modul Antarmuka akan meminta suatu kode alfabet ke sensor dan kode tersebut akan diterjemahkan, untuk mengetahui nama dan tipe data apa yang dihasilkan data sensor. Program yang dirancang adalah menggunakan fungsi yang tersedia dari *library* I2C yaitu *Wire*.*write* untuk mengirimkan permintaan kode dan

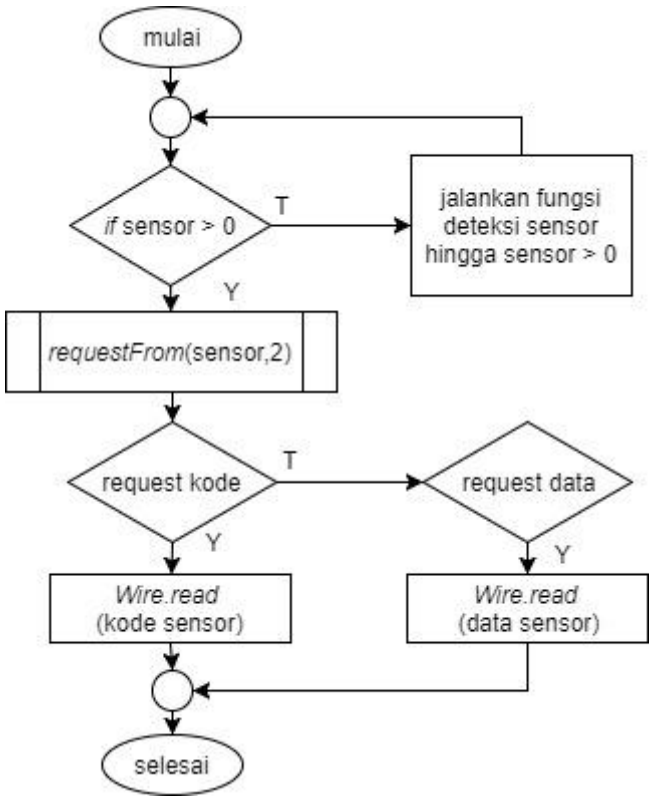
Wire.read untuk membaca kode tersebut. Kode tersebut terdiri dari huruf “A” yang berarti sensor cahaya, jika “B” adalah sensor gas, jika “C” adalah sensor jarak, dan jika “D” adalah sensor suhu. Melalui kode tersebut akan ditampilkan nama sensor dan satuan dari data sensor yang didapatkan. Dapat dilihat pada Gambar 5.5 *flowchart* di bawah ini alur programnya.



Gambar 0.5 *Flowchart* Fungsi Spesifikasi Sensor

c. Perancangan Fungsi Minta Data Ke Semua Sensor

User dapat meminta data secara sekaligus ke semua sensor yang terkoneksi melalui sebuah veriabel dalam bentuk *array*, dimana variabel tersebut berisi alamat-alamat sensor yang sudah terhubung dengan Modul Antarmuka.

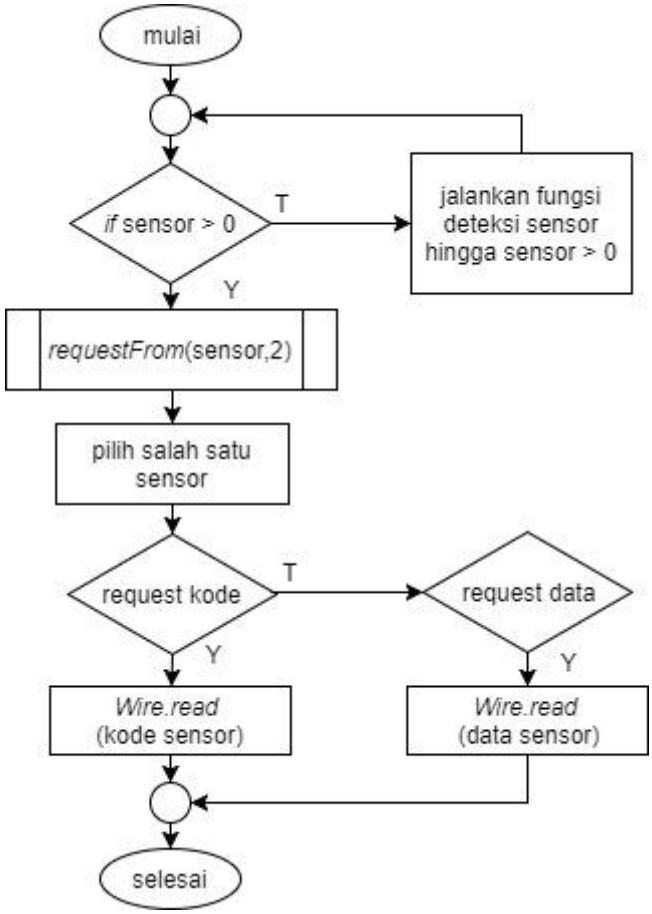


Gambar 0.6 *Flowchart* Minta Data Semua Sensor

Proses minta data dilakukan serentak dan ditampilkan pada serial monitor sesuai urutan sensor yang pertama kali terhubung dan dengan alamat terkecil. Untuk meminta data sensor, pertama sekali Modul antarmuka harus terlebih dahulu mengetahui kode dari sensor tersebut, kemudian baru akan mengurum data sensor karena sensor akan aktif ketika Modul Antarmuka sudah mengetahui kode sensor. Dapat dilihat pada Gambar 5.6 alur programnya.

d. Perancangan Fungsi Meminta Data Sesuai Alamat Sensor

User dapat meminta data secara spesifik pada salah satu sensor berdasarkan alamat sensor. Data yang diminta harus di-*request* terlebih dahulu. Data tersebut berupa data mentah dari sensor yang ditampilkan pada serial monitor. Proses ini dijalankan setelah kode sensor sudah diketahui oleh Modul Antarmuka. Berikut ini adalah Gambar 5.7 *flowchart* untuk meminta data.



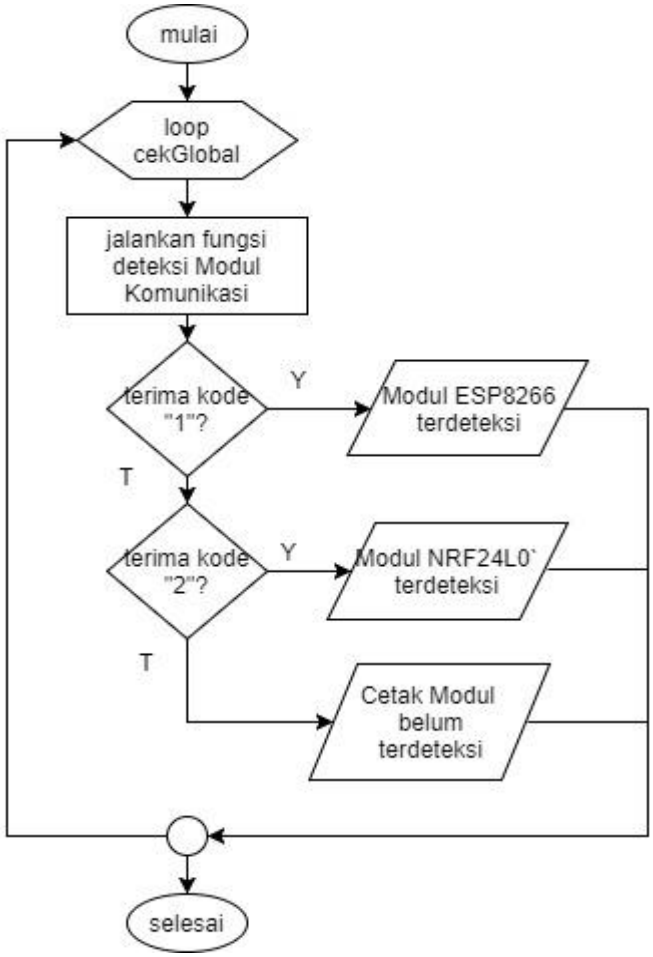
Gambar 0.7 *Flowchart* Fungsi Minta Data Sesuai Sensor

1.1.2.2 Perancangan Modul Antarmuka Dengan Modul Komunikasi

Perancangan Perangkat lunak Modul Antarmuka dengan Modul Komunikasi terbagi menjadi dua fungsi utama dan digunakan sebagai menu untuk *user*. Program dengan berbagai fungsi ini dijalankan pada Arduino IDE dengan meng-*install library SoftwareSerial*. Proses perancangan programnya ditunjukkan pada gambar *flowchart* di masing-masing sub bab fungsi.

a. Perancangan Fungsi Deteksi Modul Komunikasi

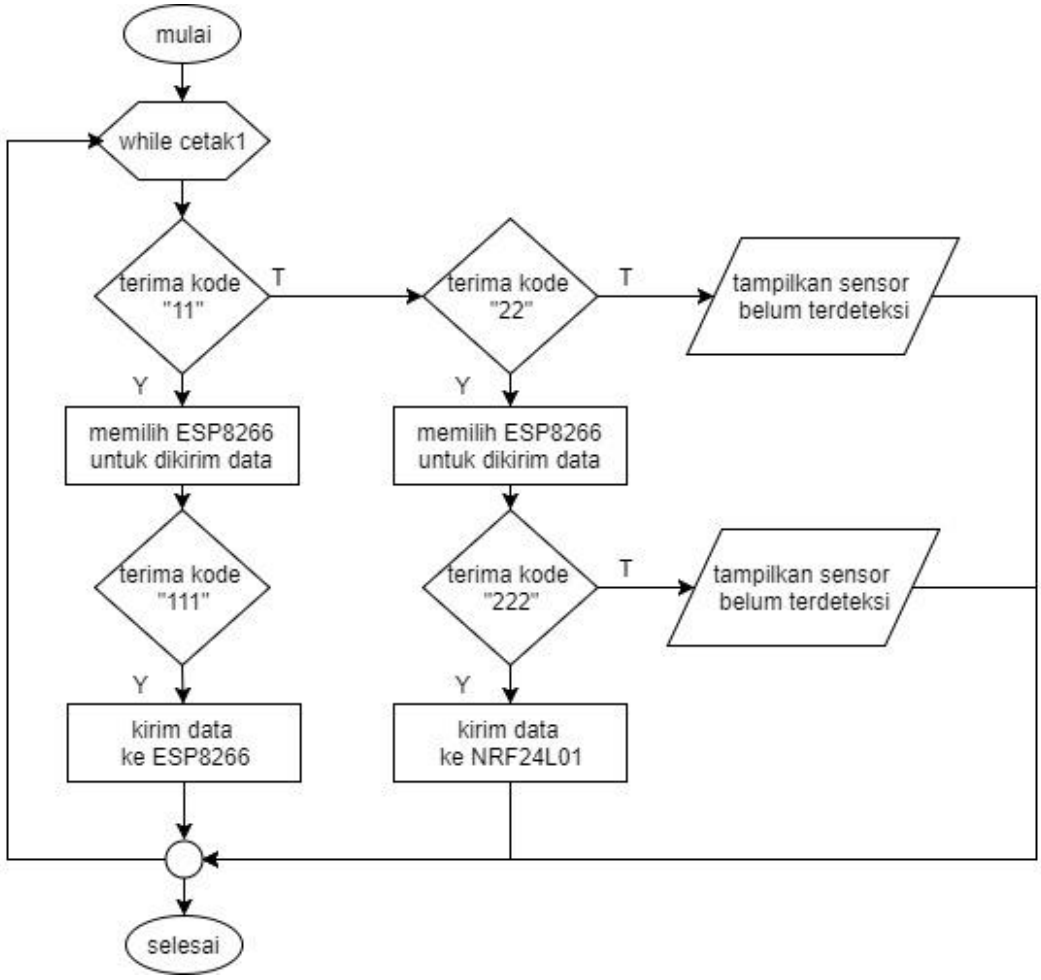
Perancangan fungsi mendeteksi adanya Modul Komunikasi ialah untuk mengetahui Modul Komunikasi yang terkoneksi, apakah Modul Komunikasi NRF24L01 atau Modul Komunikasi ESP8266 yang sedang terhubung. Ketika modul terhubung dan Modul Antarmuka menerima kode dari modul tersebut berupa angka “1” maka terdeteksi bahwa ESP8266 sedang terhubung, namun apabila angka “2” berarti yang sedang terhubung adalah NRF24L01. Kedua Modul Komunikasi dapat terkoneksi secara bersamaan. Pada Gambar 5.8 *flowchart* di bawah dapat dilihat alur pendeteksiannya.



Gambar 0.8 *Flowchart* Deteksi Modul Komunikasi

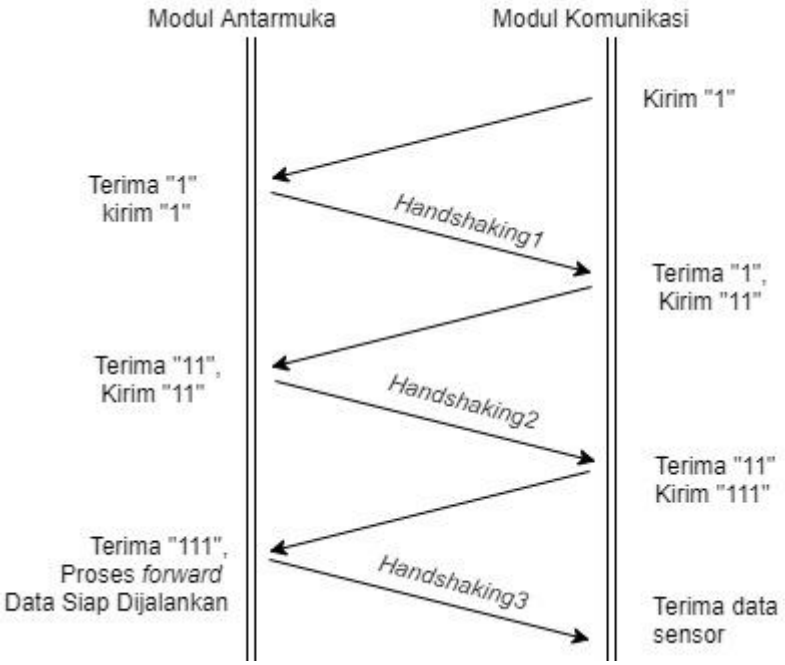
b. Perancangan Fungsi *Forward* Data

Perancangan fungsi *forward* data ialah meneruskan data yang diperoleh dari Modul Sensor ke Modul Komunikasi. Data sensor yang sudah diterima oleh Modul Antarmuka langsung diteruskan tanpa ada penyimpanan data jika modul komunikasi yang terkoneksi hanya satu. Jika yang terkoneksi dua perangkat, maka *user* dapat memilih untuk mengirimkan ke Modul Komunikasi yang mana. Pemilihan dapat dilakukan pada menu yang disediakan Modul Antarmuka. Pada Gambar 5.9 dapat dilihat alur *forward* datanya.



Gambar 0.9 *Flowchart* Fungsi *Forward* Data

Pada Gambar 5.10 merupakan proses *handshaking* antara Modul Antarmuka dan Modul Komunikasi. Setiap kali Modul Komunikasi mengirimkan kode angka dan diterima oleh Modul Antarmuka, maka Modul Antarmuka akan membalas kode tersebut dengan kode yang sama sebagai penginformasian bahwa kode itu sudah sampai dan perintah siap dijalankan



Gambar 0.10 Proses *Handshaking* Modul Antarmuka dengan Modul Komunikasi

1.2 Implementasi Sistem

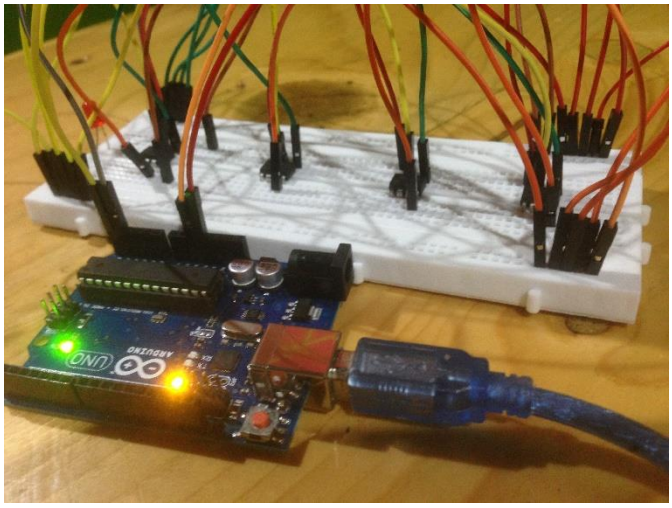
Pada implementasi sistem dilakukan ketika semua proses perancangan sistem telah dipenuhi. Pembahasan tentang implementasi sistem pada sub bab ini antara lain menjelaskan tentang implementasi *kode* program pada pada Arduino UNO IDE. Pada sub bab implementasi sistem akan dibagi menjadi 2 bagian, yaitu implementasi prangkat keras dan implementasi perangkat lunak. Pada implementasi akan dijelaskan melalui dokumentasi alat dan potongan-potongan program yang menjelaskan alur dari fungsi yang disediakan.

1.2.1 Implementasi Perangkat Keras

Bab implemantasi perangkat keras menampilkan dokumentasi dari perangkat yang sudah dibuat, sub bab ini akan terbagi menjadi dua yaitu implementasi perangkat Modul Antarmuka dengan Modul Sensor dan implementasi perangkat Modul Antarmuka dengan Modul Komunikasi.

1.2.1.1 Implementasi Modul Antarmuka Dengan Modul Sensor

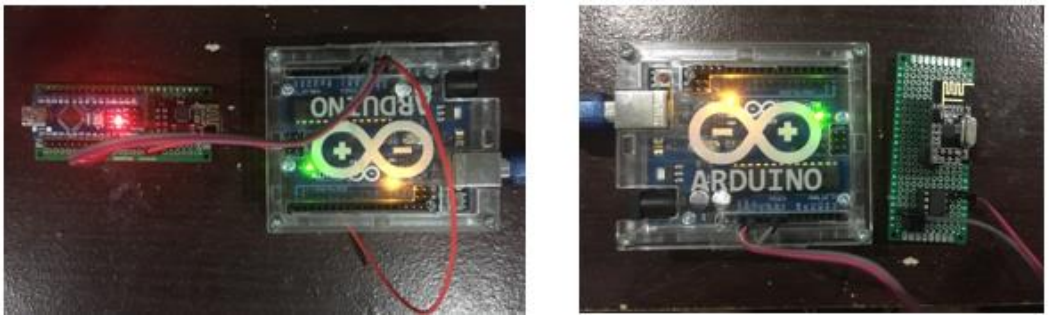
Implementasi perangkat keras ini dirancang sesuai bab 5.1.1.1 yaitu menggunakan sebuh *microcontroller* Arduino UNO yang dihubungkan dengan Attiny85 sesuai pinnya. Untuk penyambungannya menggunakan *jumper female-female* yang mana semua perangkat berada di atas *breadboard*. Hasil pengimplementasian perangkat keras Modul Antarmuka dan Modul sensor dapat dilihat pada Gambar 5.11.



Gambar 0.11 Implementasi Perangkat Keras Modul Antarmuka dan Sensor Dengan Komunikasi I2C

1.2.1.2 Implementasi Modul Antarmuka Dengan Modul Komunikasi

Implementasi perangkat keras ini dirancang sesuai bab 5.1.1 yaitu menggunakan sebuh *Microcontroller* Arduino UNO yang dihubungkan dengan Attiny85 dan Arduino NANO sesuai dengan pin serialnya. Untuk penyambungannya menggunakan *jumper*. Hasil pengimplementasian perangkat keras Modul Antarmuka dan Modul sensor dapat dilihat pada Gambar 5.12.



Gambar 0.12 Implementasi Perangkat Keras Modul Antarmuka dan Modul Komunikasi dengan Komunikasi UART

1.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menampilkan baris program pada setiap fungsinya. Sub bab ini terbagi menjadi dua yaitu implementasi Modul Antarmuka dengan Modul Sensor dan implementasi Modul Antarmuka dengan Modul Komunikasi.

1.2.2.1 Implementasi Modul Antarmuka Dengan Modul Sensor

Implementasi perangkat lunak menjelaskan proses realisasi program untuk Modul Antarmuka. Pada Implementasi Perangkat lunak dijalankan sepenuhnya melalui program dilakukan pada Arduino IDE. Untuk memulai program dilakukan inisialisasi *library* yang digunakan untuk mempermudah pemrograman beberapa

fungsi tertentu. *Library* yang digunakan adalah "*Wire*.h" yang dituliskan pada baris pertama.

Baris selanjutnya adalah Inisialisasi berbagai variabel global yang diperlukan untuk komunikasi dengan sensor seperti variabel alamat sensor, *array* penyimpan sensor yang terkoneksi, *array* penyimpan data sensor yang diterima, serta variabel lain yang dipanggil pada fungsi di bawahnya yang dapat dilihat pada baris 10-14.

Pada baris 16-20 terdapat fungsi dengan nama "inisialisasiStatus" yaitu untuk inisialisasi status sensor dengan status *true* atau *false*, jika *true* akan dijalankan, dan jika *false* maka menjalankan kondisi lain. Terdapat pula variable "cmdSensor" yaitu variabel yang akan dikirim ke sensor sebagai perintah untuk meminta kode sensor, juga "cmdData" sebagai perintah meminta data sensor. Dapat dilihat pada Gambar 5.13 penggalan programnya.

Gambar 0.13 Penggalan Program Inisialisasi dan Deklarasi Sensor

5.2.2.1a Implementasi Fungsi Deteksi Sensor

Fungsi deteksi sensor dimulai dengan inisialisasi fungsi terlihat pada baris pertama. Pencarian alamat melalui *looping* dari 1-127. Angka 127 didapatkan dari 2^7 untuk perangkat maksimal yang terhubung pada Modul Antarmuka. Pada baris 6 dan 7 adalah inisialisasi dalam memulai transmisi I2C menggunakan fungsi *Wire.beginTransmission*() dengan alamat sensor.

Terdapat percabangan dengan *if* yaitu dengan kondisi apabila "*error*==0" berarti ada sensor yang terhubung, kemudian memberikan *return* untuk *wire.endTransmission* dengan hasil sukses. Semua sensor yang terdeteksi

dimasukkan pada *array* dengan nama “Aktif” terlihat pada baris 12. Kondisi *if else* apabila “*error*==4” maka akan mencetak alamat yang *error*. Variabel *nDevices* akan selalu di-*increment* saat sensor bertambah. Potongan program dapat dilihat pada Gambar 5.14.

Gambar 0.14 Penggalan Program Fungsi Deteksi Sensor

5.2.2.1b Implementasi Fungsi Spesifikasi Sensor

Implementasi fungsi spesifikasi sensor menunjukkan kepada *user* nama dan satuan sensor. Dimulai dengan inisialisasi nama fungsi serta memulai transmisi I2C dengan fungsi *Wire.beginTransmission*(). Dalam fungsi ini dilakukan pembacaan kode dengan menuliskan "*Wire.read*(cmdSensor)” saat sensor pertama kali ditancapkan, tertulis pada baris ke-2 program. Melalui kode yang dibaca akan dilakukan percabangan dengan *if* pada baris 212-230 untuk mendeteksi kode yang diterima dan menyesuaikan dengan definisi kode tersebut. Dengan fungsi spesifikasi sensor *user* memiliki opsi untuk mengambil data dari sensor yang mana, karena sudah mengetahui spesifikasi dari semua sensor yang terhubung. Pada Gambar 5.15 dapat dilihat bagaimana alur program spesifikasi sensor.

PROGRESS_12 §

```
209    if (Wire.available()) {
210      if (statusCetak[i]) {
211        kode[i] = Wire.read();
212        if (kode [i] == 'A') {
213          Serial.print("Kode A = ");
214          Serial.println("Sensor Cahaya ");
215          Serial.println("Value Candela");
216        }
217        else if (kode[i] == 'B') {
218          Serial.print("Kode B = ");
219          Serial.println("Sensor Gas ");
220          Serial.println("Value Metana ");
221        }
222        else if (kode[i] == 'C') {
223          Serial.print("Kode C = ");
224          Serial.println("Sensor Jarak ");
225          Serial.println("Satuan : Cm ");
226        }
227        else if (kode[i] == 'D') {
228          Serial.print("Kode D = ");
229          Serial.println("Sensor Suhu ");
230          Serial.println("Satuan : Celsius ");
231        }
```

Gambar 0.15 Penggalan Program Fungsi Spesifikasi Sensor

5.2.2.1c Implementasi Fungsi Minta Data Ke Semua Sensor

Pada implementasi minta data ke semua sensor diawali dengan inisialisasi fungsi tersebut dan dilanjutkan dengan memulai transmisi dengan alamat sensor yang tersimpan pada *array* "Aktif[]" dengan perulangan *for* sebanyak jumlah sensor. Pada baris 4-7 adalah apabila "*statusDetect=false*" maka akan mengirimkan perintah meminta kode sensor dengan menuliskan *Wire.write*(cmdSensor). Variabel "CmdSensor" merupakan kode sensor yang memiliki nilai "x" yang nantinya akan dibandingkan dengan nilai "CmdSensor" yang terdapat pada Attiny85. Apabila nilai sama, sensor akan mengirimkan kode A, B, C, atau D. Pada baris 8-11 adalah kondisi apabila *"statusDetect=false*" maka akan menjalankan perintah *Wire.write*(cmdData), cmdData adalah variabel yang dikirim Modul Antarmuka untuk meminta data sensor.

Pada baris 14-16 terdapat fungsi *Wire.requestFrom*(aktif[i], 2), yaitu me-*request* sebanyak 2 byte kepada sensor. Pada baris 17-25 merupakan *nested if* dengan syarat percabangan *if* yang pertama apabila terdapat *input* dari *user*, maka perintah dalam percabangan *if* akan dijalankan. Syarat percabangan *if* yang kedua apabila "statusCetak=*true"* maka akan menampilkan kode sensor yang bersumber dari sensor.

Apabila syarat tidak terpenuhi maka percabangan *else* akan dijalankan, yang berfungsi untuk menerima data dengan fungsi *Wire.read*. Data sensor yang diterima disimpan pada variabel lb dan hb, agar sejumlah 4 digit bilangan dapat

ditampilkan. Variabel Lb dan hb di-*parsing* dengan tipe data *word* (2 byte). Berikut adalah Gambar 5.16 yaitu program implementasi minta data ke semua sensor.

Gambar 0.16 Penggalan Program Fungsi Minta Data Ke Semua Sensor

5.2.2.1d Implementasi Fungsi Meminta Data Sesuai Alamat Yang Dipilih.

Pada implementasi minta data sesuai alamat sensor dimulai dengan inisialisasi fungsi dan memulai transmisi I2C dengan sensor yang aktif. Pada baris ke 14 akan di-*request* alamat yang ditunjuk dari *array* "Aktif[]" dengan fungsi *Wire.requestFrom*(aktif[i], 2) dan akan dibaca dengan fungsi *Wire.read*(). Untuk

meminta data sesuai alamat *user* dapat memilih dari beberapa alamat yang tersedia, pemilihan alamat yaitu melalui menu yang disediakan pada serial monitor dengan kode program pada Gambar 5.17

```
77      Serial.println("Data Sesuai Alamat Modul\n");
78      Serial.println("Input 0 untuk memilih alamat pertama ");
79      Serial.println("Input 1 untuk memilih alamat kedua ");
80      Serial.println("Input 2 untuk memilih alamat ketiga ");
81      Serial.println("Input 3 untuk memilih alamat keempat ");
```

Gambar 0.17 Memilih Sensor Yang Aktif

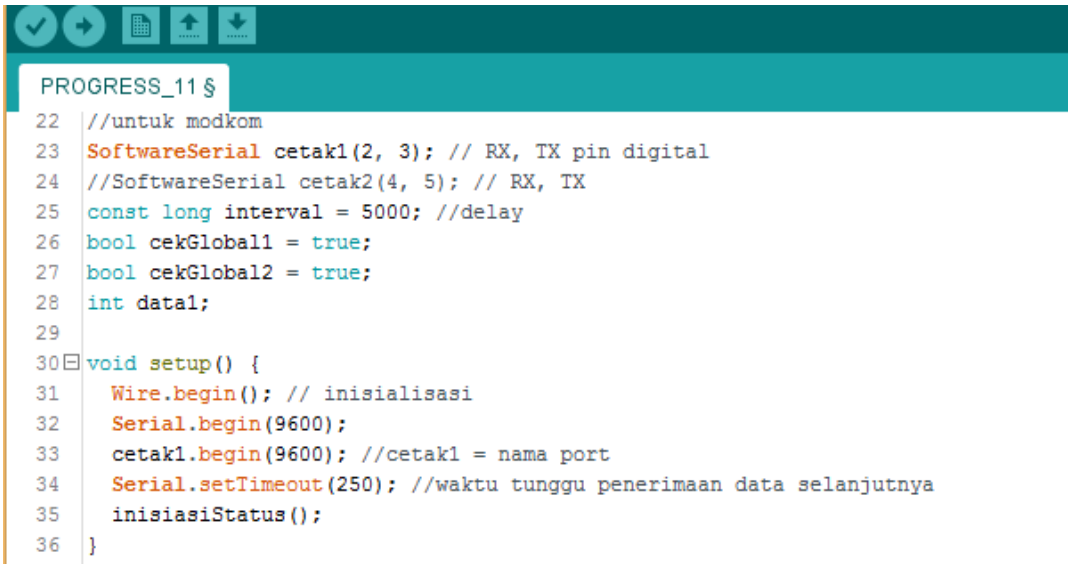
Baris 17-35 merupakan proses untuk membaca data pada variabel "byteReceived" dan akan ditampilkan dengan fungsi *serial.println*. Jika proses tidak berjalan, maka akan menampilkan "*Slave* gagal dideteksi". Pada fungsi ini ditampilkan terlebih dahulu kode sensor yang kemudian menampilkan data sensor. Program dapat dilihat pada gambar 5.18.

Gambar 0.18 Penggalan Program Fungsi Minta Data Pada Sensor Tertentu

1.2.2.2 Implementasi Modul Antarmuka Dengan Modul Komunikasi

Implementasi perangkat lunak menjelaskan proses realisasi program untuk Modul Antarmuka. Pada implementasi perangkat lunak dimulai dengan inisialisasi *library* "*SoftwareSerial*.h". Pada baris 23 merupakan inisialisasi pin RX dan TX untuk dua port serial pada pin Arduino UNO dengan nama cetak1. Pada baris 23-28 adalah inisialisasi variabel global seperti interval dan variabel data. Pada fungsi *setup* juga harus dideklarasikan nama port dengan *baudrate* 9600 dan *setTimeOut*

untuk *delay* saat fungsi "*parstInt*" dijalankan untuk menerima data Integer. Dapat dilihat pada Gambar 5.19 untuk potongan programnya.



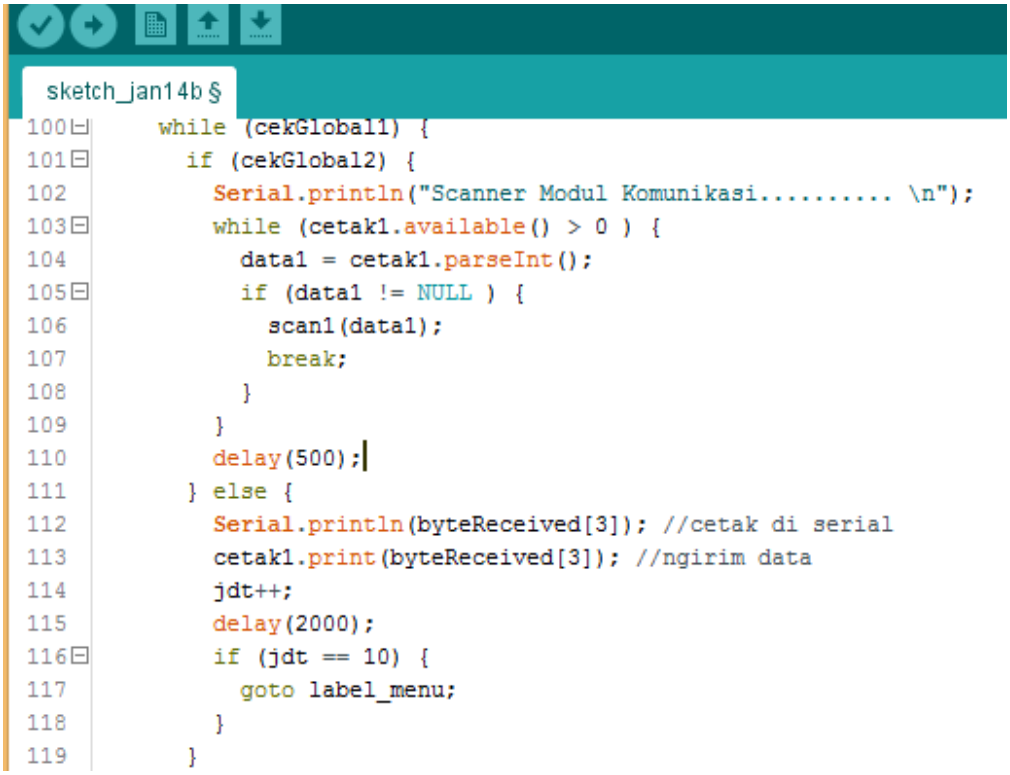
Gambar 0.19 Penggalan Program Inisialisasi dan Deklarasi untuk Modul Komunikasi

5.2.2.2a Implementasi Fungsi Deteksi Modul Komunikasi

Pada implementasi fungsi *scanner* Modul Komunikasi ini dilakukan di dalam fungsi void *loop*. Cetak1 merupakan nama port yang diberi kondisi percabangan *if* dengan syarat apabila ada Modul Komunikasi yang terhubung pada port tersebut yang menandakan ada data masuk dan akan diubah ke dalam bentuk Integer menggunakan fungsi *parseInt.*

Saat percabangan bernilai *false* maka akan dijalankan fungsi kirim data. Fungsi kirim data dijalankan melalui proses *handshaking* yang ada pada fungsi *Scan1.* Fungsi *Scan1* dapat dilihat pada sub bab *forward* data pada Gambar 5.20.

Fungsi yang digunakan oleh Modul Antarmuka untuk mengirim data atau perintah ke Modul Komunikasi menggunakan perintah perintah *serial.print(data)*. Fungsi tersebut digunakan sebagai *handshaking* dengan Modul Komunikasi untuk penentuan Modul Komunikasi yang dipilih, serta proses kirim data. Pemberian *delay* sebanyak 200 didasarkan pada sinkronisasi dengan Modul Komunikasi agar data dapat diterima dengan jumlah yang benar serta akurat. Penggalan programnya dapat dilihat pada Gambar 5.20.



Gambar 0.20 Penggalan Program Fungsi Deteksi Modul Komunikasi

5.2.2.2b Implementasi Fungsi *Forward* Data

Implementasi fungsi *forward* data dilakukan dengan syarat apabila *handshaking*1 dan *handshaking*2 sudah berjalan. Ketika port Cetak1 dideteksi dan Modul Komunikasi yang terdeteksi tersebut mengirimkan angka "1" maka Modul Antarmuka menerimanya dengan bentuk data Integer. Modul Komunikasi akan mengirim kembali angka "1" sebagai *acknowledgement* (ack) yang dapat dilihat pada baris 299. Proses ini dinamakan *Handshaking*1 yang mana Modul Komunikasi sudah terdeteksi dengan menerjemahkan kode "1" sebagai Modul Komunikasi ESP8266 dan kode "2" sebagai Modul Komunikasi NRF24L01.

Proses *Handshaking*2 yaitu saat Modul Komunikasi mengirimkan angka "11" dan Modul Antarmuka menerimanya dan membalas dengan angka "11" juga sebagai ack. Proses ini menandakan Modul Antarmuka sudah memilih untuk mengirim ke Modul Komunikasi yang dituju. Angka "11" untuk ESP8266 dan "22" untuk NRF24L01.

Proses *Handshaking3* adalah proses terakhir yaitu konfirmasi pengiriman data. Ketika Modul Antarmuka menerima angka "111" (untuk ESP8266) atau "222" (untuk NRF24L01) dari Modul Komunikasi maka menandakan Modul Komunikasi siap untuk menerima data sensor. Maka Proses kirim data akan dilaksanakan dengan perulangan sebanyak 10 kali untuk setiap pengirimannya. Untuk penggalan programnya dapat dilihat pada Gambar 5.21

Gambar 0.21 Penggalan Program *Handshaking* untuk *Forward* Data