

LAMPIRAN

1. Source Code Keseluruhan

a. AlcoholChecker

```
#include "Alcohol.h"  
#include "MonitorLCD.h"  
#include "ClientServer.h"  
#include "global_define.h"
```

```
Alcohol alcohol;  
MonitorLCD lcd("indonesia");  
const int OPA = 0;  
const int OPD = 8;
```

```
void setup() {  
    Serial.begin(BAUD_PRIMARY);  
    pinMode(OPD, INPUT);  
    serverStart();  
    lcd.initiate();  
}
```

```
void loop() {  
    int alcoholValue = analogRead(OPA),  
        percentage = alcohol.getPercentage();  
    char group = alcohol.getGroup();  
    String jsonValues[5] = {  
        String(percentage),  
        String(group),  
        alcohol.getEtanol(),  
        alcohol.getDisease(),  
        alcohol.getDescription()  
    };  
    alcohol.initiate(alcoholValue);  
    alcohol.updateValues();  
    printAlcoholStatus(alcohol);  
    Serial.print("Analog Value: ");  
    Serial.println(alcoholValue);  
    serverWatch(jsonValues);  
    lcd.render(percentage, group);  
    delay(1000);  
}
```

b. Alcohol.h

```
#include "global_define.h"
/*
|-----
| Deskripsi Alkohol
|-----
|
| Definisi deskripsi untuk masing - masing golongan alkohol. Deskripsi
| alkohol akan berganti otomatis sesuai dengan nilai input udara yang
| diterima melalui sensor MQ 3
|
*/
#define DESC_A "Belum akan mengalami mabuk, tetapi tetap memiliki efek
kurang baik bagi tubuh"
#define DESC_B "Alkohol pada kadar ini sudah cukup tinggi"
#define DESC_C "Kadar alkohol paling tinggi yang boleh di konsumsi oleh
manusia"
#define DESC_D "Menyebabkan kematian"

/*
|-----
| Grup Alkohol
|-----
|
| Definisi grup untuk masing - masing golongan alkohol. Grup alkohol juga
| akan berganti otomatis sesuai dengan nilai input udara yang diterima
| melalui sensor MQ 3
|
*/
#define GROUP_A 200
#define GROUP_B 300
#define GROUP_C 400
#define GROUP_D 500

/**
 * Implementasi dari perubahan data alkohol.
 *
 * Class yang bertugas untuk memanipulasi data alkohol
 * saat sebelum atau sesudah scan kadar alkohol dilakukan.
 *
 */
```

```

class Alcohol
{
private:
/**
 * Nilai analog dari sensor MQ3.
 *
 * @property int
 * @accessor private
 */
int value;

/**
 * Nilai persentasi alkohol
 *
 * @property int <Nilai berkisar antara 0% s/d 100%>
 * @accessor private
 */
int percentage;

/**
 * Grup alkohol.
 *
 * @property char[1]
 * @accessor private
 */
char group;

/**
 * Kadar etanol alkohol
 *
 * @property String
 * @accessor private
 */
String etanol;

/**
 * Keterangan tambahan alkohol.
 *
 * @property String
 * @accessor private
 */
String description;

```

```

/**
 * Penyakit yang disebabkan oleh alkohol.
 *
 * @property String
 * @accessor private
 */
String disease;

boolean groupA() {
    return this->value > GROUP_A;
}

boolean groupB() {
    return this->value > GROUP_B;
}

boolean groupC() {
    return this->value > GROUP_C;
}

boolean groupD() {
    return this->value > GROUP_D;
}

void updateDescription () {
    if (this->groupA())
        this->description = DESC_A;

    if (this->groupB())
        this->description = DESC_B;

    if (this->groupC())
        this->description = DESC_C;

    if (this->groupD())
        this->description = DESC_D;
}

void updateDisease() {
    if (this->groupA())
        this->disease = NIHIL;
}

```

```

if (this->groupB())
    this->disease = "Hati,Jantung";

if (this->groupC())
    this->disease = "Hati,Jantung";

if (this->groupD())
    this->disease = "Hati,Jantung";
}

void updateGroup() {
    if (this->groupA())
        this->group = 'A';

    if (this->groupB())
        this->group = 'B';

    if (this->groupC())
        this->group = 'C';

    if (this->groupD())
        this->group = 'D';
}

void updateEtanol() {
    if (this->groupA())
        this->etanol = "1-5";

    if (this->groupB())
        this->etanol = "5-20";

    if (this->groupC())
        this->etanol = "20-50";

    if (this->groupD())
        this->etanol = "50-100";
}

void updatePercentage() {
    if (this->groupA())
        this->percentage = 5;
}

```

```

    if (this->groupB())
        this->percentage = 21;

    if (this->groupC())
        this->percentage = 42;

    if (this->groupD())
        this->percentage = 60;
}

public:
/**
 * Mengambil nilai deskripsi alkohol.
 *
 * @return String
 */
String getDescription() {
    return this->description;
}

/**
 * Mengambil nilai penyakit alkohol.
 *
 * @return String
 */
String getDisease() {
    return this->disease;
}

/**
 * Mengambil nilai grup alkohol.
 *
 * @return char
 */
char getGroup() {
    return this->group;
}

/**
 * Mengambil nilai etanol alkohol.
 *

```

```

* @return String
*/
String getEtanol() {
    return this->etanol;
}

/**
 * Mengambil nilai persentasi alkohol.
 *
 * @return String
 */
int getPercentage() {
    return this->percentage;
}

/**
 * Inisialisasi nilai yang didapat dari
 * sensor MQ3
 *
 * @param int value
 * @return void
 */
void initiate(int value) {
    this->value = value;
}

void updateValues() {
    if (this->value < 200) {
        this->percentage = 0;
        this->group = '-';
        this->etanol = NIHIL;
        this->disease = NIHIL;
        this->description = NIHIL;
    }
    else {
        this->updatePercentage();
        this->updateGroup();
        this->updateEtanol();
        this->updateDisease();
        this->updateDescription();
    }
}
}

```

```

};

/**
 * Print status alkohol pada serial.
 *
 * @params Alcohol alcohol <Objek dari kelas alkohol>
 * @return void
 */
void printAlcoholStatus(Alcohol alcohol) {
    Serial.println("===== ALCOHOL STATUS =====");

    Serial.print("Percentage: ");
    Serial.print(alcohol.getPercentage());
    Serial.println("%");

    Serial.print("Etanol: ");
    Serial.print(alcohol.getEtanol());
    Serial.println(" (%)");

    Serial.print("Golongan: ");
    Serial.println(alcohol.getGroup());

    Serial.print("Penyakit: ");
    Serial.println(alcohol.getDisease());

    Serial.print("Deskripsi: ");
    Serial.println(alcohol.getDescription());

    Serial.println("=====");
    Serial.println();
}

```

c. ClientServer.h

```

#include <SPI.h>
#include <Ethernet.h>
#include <EthernetClient.h>
#include <EthernetServer.h>
#include "global_define.h"

#define DEFAULT_MAC { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }
#define HTTP_PORT 80

/**

```



```

* Port monitor server.
*
* @var int
*/
const int BAUD = 9600;

const byte MAC[] = DEFAULT_MAC;

String jsonKeys[5] = {
  "percentage", "group", "etanol", "disease", "description"
};

// Inisialisasi server.
EthernetServer server(HTTP_PORT);

IPAddress IP(192, 168, 0, 101);

void(* restart)(void) = 0;

void serverStart() {
  Ethernet.begin(MAC, IP);

  server.begin();

  Serial.print("Server is at ");
  Serial.println(Ethernet.localIP());
}

void serverWatch(String jsonValues[5]) {
  EthernetClient client = server.available();

  if (client) {
    Serial.println("new client");

    boolean currentLineIsBlank = true;

    while (client.connected()) {
      if (client.available()) {
        char clientRead = client.read();
        Serial.write(clientRead);

        if (clientRead == '\n' && currentLineIsBlank) {

```

```

client.println("HTTP/1.1 200 OK");
client.println("Access-Control-Allow-Origin: *");
client.println("Content-Type: application/json;charset=utf-8");
client.println("Server: Arduino");
client.println("Connection: close");
client.println();
client.print("{}");

for (int i = 0; i < 5; i++) {
  client.print("\n");
  client.print(jsonKeys[i]);
  client.print("\n: ");

  client.print("\n");
  client.print(jsonValues[i]);
  client.print("\n");

  if (i < 4) {
    client.print(", ");
  }
}

client.print("{}");

break;
}

if (clientRead == '\n') {
  currentLineIsBlank = true;
}
else if (clientRead != '\r') {
  currentLineIsBlank = false;
}
}
}

delay(1);

client.stop();
Serial.println("client disconnected.");
delay(5000);
restart();

```

```

    }
}
d. MonitorLCD.h
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "global_define.h"

#define COLUMNS 16
#define ROWS 2
#define GREETING_EN "Welcome"
#define GREETING_ID "Selamat Datang"

class MonitorLCD {
private :
    String lang;
    LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, COLUMNS, ROWS);
    void greeting() {
        lcd.setCursor(2, 0);
        lcd.print("*** Hallo ***");
        if (this->lang == "english") {
            lcd.setCursor(5, 1);
            lcd.print(GREETING_EN);
        }
        else {
            lcd.setCursor(1, 1);
            lcd.print(GREETING_ID);
        }
    }

public:
    String text;
    MonitorLCD(String lang) {
        this->lang = lang;
    }
    void initiate() {
        lcd.begin();
        lcd.backlight();
        this->greeting();
        delay(2000);
        lcd.clear();
    }
}

```

```
void render(int percentage, char group) {  
    lcd.setCursor(0, 0);  
    lcd.print(String("Alcohol: ") + String(percentage) + String("%"));  
  
    lcd.setCursor(0, 1);  
    lcd.print("Golongan: ");  
    lcd.print(group);  
}  
};
```