

## DAFTAR PUSTAKA

- Abdul Kadir.(2017).*Pemrograman Arduino Dan Processing*. Jakarta : Penerbit Elex Media Komputindo
- Abdul Kadir.(2018). *Arduino Dan Sensor*. Yogyakarta : Penerbit Andi Publisier
- Handayani Saptaji W.(2015).*Mudah Belajar Mikrokontroller Dengan Arduino* : Penerbit Widya Media
- Ir. Sutarno, M,Sc.(2014).*Instrumentasi Industri Dan Kontrol Proses*. Yogyakarta : Penerbit Graha Ilmu
- Irfan Indra Kurniawan.(2017).*fansElectronics.com*.
- Ivan Brokhotkov.(2015).*ESP 8266 Arduino Core Documentation Release 2.4.0*
- Ajie.(2016).*Saptaji.com*
- M. Fajar Wicaksana, Hidayat.(2017).*Mudah Belajar Mikrokontroler Arduino* : Penerbit Informatika
- Tim Fakultas Teknologi Industri (2019). *Pedoman Tugas Akhir*. Universitas PGRI Adi Buana Surabaya.
- Ratih Listiyani.(2018). *Dasar Listrik Dan Elektronika*. Yogyakarta : Penerbit CV Budi Utama
- Risna, Harrizki Arie Perdana. (2014). *Rancang Bangun Aplikasi Monitoring Penggunaan Air PDAM Berbasis Mikrokontroler Arduino Uno*. Teknik informatika STMIK Atma Luhur, Kepulauan Bangka Belitung
- Saiful Anwar. (2015). *Perancangan Sistem Pengisian Bahan Bakar Premium dan Solar Bersubsidi dengan Teknologi RFID Sebagai Pembatas Jumlah Beli dan Pembeli*. Jurusan Teknik Elektro, Teknik Elektro Universitas Diponegoro, Semarang
- Winda Gissella Ginting. (2017) . *Rancang Bangun Alat Ukur Debit Air Berbasis Mikrokontroler Arduino Uno Dengan Menggunakan Sensor Yf-201*.



# UNIVERSITAS PGRI ADI BUANA SURABAYA

## FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)

### BERITA ACARA BIMBINGAN SKRIPSI

Form Skripsi-03



Nama	: HARIS ATMAJAYA
NIM	: 193609010
Program Studi	: TEKNIK ELEKTRO
Pembimbing	: Drs. BUDI PRJO SEMBODO, ST., M.KOM
Periode Bimbingan	: <u>Genap</u> (Genap*) Tahun 2020 / 2021
Judul Skripsi	PERANCANGAN SMART FUEL STATION AND MANAGEMENT SYSTEM BERBASIS ARDUINO DAN INTERET OF THINGS

#### KEGIATAN KONSULTASI / BIMBINGAN

No	Tanggal	Materi pembimbingan	Keterangan	Paraf
1	28 Sep 2020	BAB I Pendahuluan, Latar Belakang	Ace	
2	2 Okt 2020	BAB I Rumusan masalah, Manfaat	Ace	
3	10 Okt 2020	BAB II Tinjauan Pustaka	Ace	
4	20 Okt 2020	BAB III Rancangan Penelitian, Konsep	Ace	
5	2 Nov 2020	BAB III Populasi Dan Pemilihan sampel	Ace	
6	18 Nov 2020	BAB III Pengumpulan Data, Metode Analisa	Ace	
7	3 Des 2020	BAB IV Data Hasil Pengujian	Ace	
8	20 Des 2020	BAB IV Hasil Analisa Data, Pembahasan	Ace	
9	5 Jan 2021	BAB V Kesimpulan dan Saran	Ace	
10	18 Jan 2021	Pustaka	Ace	

Dinyatakan selesai tanggal : ...18... JANUARI..... 2021.

Mengetahui,  
Ketua Program Studi,  
  
Atha Syam S.S., M.S.

Pembimbing,  
  
BUDI PRJO S. ST. KOM

Surabaya, 18 Januari 2021  
Mahasiswa,  
  
HARIS ATMAJAYA



# UNIVERSITAS PGRI ADI BUANA SURABAYA

## FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)

### FORM REVISI SKRIPSI

Nama Mahasiswa : HARIS ATMAJAYA  
NIM : 193609010  
Fakultas / Progdi : FAKULTAS TEKNIK / TEKNIK ELEKTRO  
Judul Skripsi : PERANCANGAN SMART FUEL STATION  
AND MANAGEMENT SYSTEM BERBASIS  
ARDUINO DAN INTERNET OF THINGS  
Ujian Tanggal : 1 Februari 2021

No Bab.	Tanggal	Materi Konsultasi	Keterangan Catatan	Tanda Tangan Penguji
I	9-2-2021	- Kesalahan pengukuran	} ale	
II		- ditambahkan pada kompila		
III		- Saran untuk peneliti berikutnya		
IV		- bisa diintegrasikan		
V		- pada head unit mobil		
		- SOP penggunaan alat		

Disetujui Dosen Penguji

Pada Tanggal, 9-2-2021

Penguji I,

Penguji II,

( ATMIASHI, ST. MT )

- Penyelesaian Revisi paling lambat 2 minggu dari pelaksanaan Ujian Skripsi.
  - Pengetikan, penjilidan, penandatanganan Skripsi dan mengumpulkan Skripsi paling lambat 2 minggu dari revisi.
- Apabila sampai batas waktu tersebut ( point 1, a dan b ) mahasiswa belum menyelesaikan revisi dan tanda tangan, maka Ujian dinyatakan Gugur.
- Foto copy Form Revisi diserahkan ke Program Studi.
  - Skripsi yang sudah direvisi diserahkan ke Fakultas tiga eksemplar untuk dijilid.

## Lampiran 1

# SOP Penggunaan Smart Fuel Station And Management System Berbasis Arduino dan Internet Of Things

1. Setting WIFI pada router / hotspot untuk SSID = **SPBU MANDIRI**, Password = **pertalite**
2. Sambungkan selang inlet dengan penampungan (Ember, galon, tanki, kran air)
3. Sambungkan / tancapkan mesin ke power supply 220 VAC.
4. **Switch ON switch no 1** terlebih dahulu, kemudian **Switch ON switch no 2** setelah Display menyala dan terhubung pada **SPBU MANDIRI**.
5. Bila mesin sudah tersambung dengan wifi, buka web browser pada smartphone / laptop  
Masukkan **IP address 192.168.43.80** ( Atau sesuai pada Display / LCD )
6. Login Admin, User name = **Admin**, Password = **operator**  
Untuk Melakukan pengisian / Top Up Saldo :
  1. Pilih Pelanggan A atau Pelanggan B pada Isian Top Up Pelanggan
  2. Masukkan Nilai Rupiah pada Isian Top Up Senilai.
  3. Klik Top Up, dan simpan data pada halaman berikutnya sebagai konfirmasi.  
Untuk melihat mutasi / histori transaksi pelanggan, Klik pada Mutasi pelanggan A atau B  
Untuk keluar dari halaman admin, klik Keluar
7. Login Pelanggan :
  - Pelanggan A, User name = **a**, password = **a**
  - Pelanggan B. User name = **b**, password = **b**Untuk melakukan pembelian BBM :
  1. Rencanakan pengisian pada Pompa 1 atau Pompa 2
  2. Masukkan selang nozle ke tempat pengisian dan buka kran nozle.
  3. Pada halaman pelanggan, Pilih **Pompa 1** atau **Pompa 2**
  4. Masukkan **nominal Rupiah** pembelian.  
(BBM hanya jenis pertalite dengan harga **Rp. 6450 /L**.)
  5. Pilih / klik **ISI**
  6. Pastikan BBM akan keluar dari nozle dan nominal pembelian serta volume yang dikeluarkan ditampilkan pada Display / LCD.
  7. Aliran akan tertutup bila volume yang dikeluarkan sudah tercapai.
  8. Tutup kembali kran nozle dan letakkan selang pada tempatnya.
  9. Klik keluar bila sudah selesai pengisian, dan back pada browser bila ingin kembali pada halaman utama untuk melihat histori transaksi atau melakukan pembelian ulang BBM.
8. Matikan mesin dengan **Switch 1 & 2** pada posisi **OFF**
9. Cabut kabel power supply
10. Cabut selang inlet dengan penampungan.

Note : Untuk hapus / Clear semua history, masukkan **192.168.43.80/reset**

## Lampiran 2

### Listing Program Perancangan Smart Fuel Station And Management System Berbasis Arduino Dan Internet of Things

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WebServer.h>
#include <EEPROM.h>
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <FlowMeter.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

#define TIME_SAVE 30000

/*
#define STASSID
#define STASSID "Intan-HotSpot"
#define STAPSK ""
#endif
*/

#define STASSID
#define STASSID "SPBU MANDIRI"
#define STAPSK "pertalite"
#endif

#define Channel_Water_Meter 776132
WiFiClient client;
IPAddress server_addr(153,92,10,119); // IP of the MySQL *server* here
char user[] = "u5620516_arif_nugroho"; // MySQL user login username
char passworddb[] = "arif_nugroho"; // MySQL user login password

MySQL_Connection conn(&client);
MySQL_Cursor* cursor;

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //

#define keypad_addr 0x20
#define user_web_A "a"
#define password_web_A "a"
#define user_web_B "b"
#define password_web_B "b"
#define user_web_ADMIN "admin"
```

```

#define password_web_ADMIN "operator"
//Harga per liter
const int ratebb = 6450;

String ServerName = "http://app.ptsasmito.com/";
String password_lcd_A = "A456";
String password_lcd_B = "B456";

const byte ROWS = 4; // Four rows

const byte COLS = 4; // Three columns

// Define the Keymap
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 0, 1, 2, 3 };
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 4, 5, 6, 7 };
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
// Create instance of the Keypad name I2C_Keypad and using the PCF8574 chip
Keypad_I2C I2C_Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS, keypad_addr,
PCF8574 );

bool stateKeypad = false;
int countState = 0;
int TimeKeypad = millis();
int countbbm;
float calibrationFactorA = 7.5;
float calibrationFactorB = 7.3;
volatile long Meter1Pulse; // Edit
volatile long Meter2Pulse; // Edit
float flowRate1; // haris Edit
float flowRate2; // haris Edit
unsigned int flowMilliLitres1; // Edit
unsigned int flowMilliLitres2; // Edit
unsigned int totalMilliLitres1; // Edit
unsigned int totalMilliLitres2; // Edit

String vpassword;
String vPompa;
String vNominal;
String apiKey = "ED27G2PYDSJYCZPL"; // <<<<< YOUR API KEY
const char* ssid = STASSID;
const char* password = STAPSK;
int SisaMeterA,SisaMeterB;
int VolA,VolB, TotalVol;

```



```

unsigned long timergo = millis();
ESP8266WebServer server(80);
int statusCode;
String KamarS;
String StatusMenu="";
float DiPakai;
// connect a flow meter to an interrupt pin (see notes on your Arduino model for pin
numbers)
FlowMeter Meter1 = FlowMeter(D3);
FlowMeter Meter2 = FlowMeter(D4);

#define RelayA D5
#define RelayB D6
#define BuzzerA D7
#define BuzzerB D8

String hhtml="";

// set the measurement update period to 1s (1000 ms)
const unsigned long period = 1000;
unsigned long updatePeriod; // Edit
//unsigned long updatePeriod = 1000; // test SPBU 2-edit
unsigned long BuzzerOnPeriod = 1000;

// define an 'interrupt service handler' (ISR) for every interrupt pin you use
void ICACHE_RAM_ATTR Meter1ISR() {
    // let our flow meter count the pulses
    Meter1Pulse++; // Edit
    //Meter1.count(); // test SPBU 2-edit
}

// define an 'interrupt service handler' (ISR) for every interrupt pin you use
void ICACHE_RAM_ATTR Meter2ISR() {
    // let our flow meter count the pulses
    Meter2Pulse++; // Edit
    //Meter2.count(); // test SPBU 2-edit
}

long vkode;
//Check if header is present and correct
bool is_authenticated() {
    Serial.println("Enter is_authenticated");
    if (server.hasHeader("Cookie")) {
        Serial.print("Found cookie: ");
        String cookie = server.header("Cookie");
        Serial.println(cookie);
        if (cookie.indexOf("ESPSESSIONID=1") != -1) {
            Serial.println("Authentication Successful");
            return true;
        }
    }
}

```

```

Serial.println("Authentication Failed");
return false;
}

//login page, also called for disconnect
void handleLogin() {
String msg;
if (server.hasHeader("Cookie")) {
Serial.print("Found cookie: ");
String cookie = server.header("Cookie");
Serial.println(cookie);
}
if (server.hasArg("DISCONNECT")) {
Serial.println("Disconnection");
server.sendHeader("Location", "/login");
server.sendHeader("Cache-Control", "no-cache");
server.sendHeader("Set-Cookie", "ESPSESSIONID=0");
server.send(301);
return;
}
if (server.hasArg("USERNAME") && server.hasArg("PASSWORD")) {
if (server.arg("USERNAME") == user_web_ADMIN && server.arg("PASSWORD") ==
password_web_ADMIN) {
server.sendHeader("Location", "/operator");
server.sendHeader("Cache-Control", "no-cache");
server.sendHeader("Set-Cookie", "ESPSESSIONID=1");
server.send(301);
Serial.println("Log in Successful");
return;
} else
if (server.arg("USERNAME") == user_web_A && server.arg("PASSWORD") ==
password_web_A) {
server.sendHeader("Location", "/kamar");
server.sendHeader("Cache-Control", "no-cache");
server.sendHeader("Set-Cookie", "ESPSESSIONID=1");
server.send(301);
Serial.println("Log in Successful");
KamarS = "KAMARA";
return;
} else
if (server.arg("USERNAME") == user_web_B && server.arg("PASSWORD") ==
password_web_B) {
server.sendHeader("Location", "/kamar");
server.sendHeader("Cache-Control", "no-cache");
server.sendHeader("Set-Cookie", "ESPSESSIONID=1");
server.send(301);
Serial.println("Log in Successful");
KamarS = "KAMARB";
return;
}
}
msg = "Wrong username/password! try again.";

```



```

    Serial.println("Log in Failed");
}
String content = hhtml;
content += "<div id='card'> <div id='card-content'> <div id='card-title'> <h2>LOGIN</h2>
<div class='underline-title'></div> </div></div>";
content += "<form action='/login' method='POST'><label for='user-email' style='padding-
top:13px'>&nbsp;User Name :</label><br>";
content += "<input id='user-email' class='form-content' type='text' name='USERNAME'
autocomplete='on' required /> <div class='form-border'></div><br>";
content += "<label for='user-password' style='padding-
top:22px'>&nbsp;Password</label><br> <input id='user-password' class='form-content'
type='password' name='PASSWORD' required />";
content += "<div class='form-border'></div><br>";
content += " <input id='submit-btn' type='submit' name='SUBMIT' value='Submit'
/></form></div>"; // </td></tr></table> ";
content += "</body></html>";
server.send(200, "text/html", content);
}

```

//root page can be accessed only if authentication is ok

```

void handleRoot() {
    Serial.println("Enter handleRoot");
    String header;
    if (!is_authenticated()) {
        server.setHeader("Location", "/login");
        server.setHeader("Cache-Control", "no-cache");
        server.send(301);
        return;
    }
    String content = "<html><body style=background-size:cover;background-
image:url('http://3.bp.blogspot.com/-fjcfki4GQ0A/VZFtYi539dl/AAAAAAAAUYo/yN-
164wBMem/s1600/Water%2BHD%2BWallpapers%2B%2B%25287%2529.jpg');>";
    content += "<H2>Halo, selamat datang di sistem kami!</H2><br>";
    if (server.getHeader("User-Agent")) {
        content += "the user agent used is : " + server.getHeader("User-Agent") + "<br><br>";
    }
    content += "Untuk masuk ke sistem silahkan anda login terlebih dahulu <a
href='\"/login?DISCONNECT=YES\">disconnect</a></body></html>";
    server.send(200, "text/html", content);
}

```

```

void handleKamar() {
    if (!is_authenticated()) {
        server.setHeader("Location", "/kamar");
        server.setHeader("Cache-Control", "no-cache");
        server.send(301);
        return;
    }
}

```

```

if (server.hasArg("POMPA")) {
    if (server.arg("POMPA") == "POMPA 1") {

```

```

if (KamarS == "KAMARA") {
    if ((server.arg("NOMINAL").toInt()) <= SisaMeterA) {
        Meter1.reset();
        Meter1Pulse = 0;
        totalMilliLitres1 = 0; // Edit
        //TotalVol = 0; // test SPBU 2-edit
        vPompa = "A";
        StatusMenu = "JRENGA";
        vNominal = server.arg("NOMINAL");
        VolA = server.arg("NOMINAL").toInt();
        VolA = ((float) VolA/ratebb)*1000;
        digitalWrite(RelayA,HIGH);
        updatePeriod = millis();
        String content = hhtml;
        content = "Proses pengisian <br>";
        content = String(VolA) + "<br>";
        content += "Keluar klik disini <a
href=\"/login?DISCONNECT=YES\">keluar</a></body></html>";
        server.send(200, "text/html", content);
        return;
    }
    String content = hhtml;
    content = "Saldo lebih kecil dari nominal <br>";
    content += "Keluar klik disini <a
href=\"/login?DISCONNECT=YES\">keluar</a></body></html>";
    server.send(200, "text/html", content);
    return;
} else if (KamarS == "KAMARB") {
    if ((server.arg("NOMINAL").toInt()) <= SisaMeterB) {
        Meter1.reset();
        Meter1Pulse = 0;
        totalMilliLitres1 = 0; // Edit
        //TotalVol = 0; // test SPBU 2-edit
        vPompa = "A";
        StatusMenu = "JRENGB";
        vNominal = server.arg("NOMINAL");
        VolA = server.arg("NOMINAL").toInt();
        VolA = ((float) VolA/ratebb)*1000;
        digitalWrite(RelayA,HIGH);
        updatePeriod = millis();
        String content = hhtml;
        content = "Proses pengisian <br>";
        content = String(VolA) + "<br>";
        content += "Keluar klik disini <a
href=\"/login?DISCONNECT=YES\">keluar</a></body></html>";
        server.send(200, "text/html", content);
        return;
    }
    String content = hhtml;
    content = "Saldo lebih kecil dari nominal <br>";

```

```

        content += "Keluar klik disini <a
href=\\\"/login?DISCONNECT=YES\\\">keluar</a></body></html>";
        server.send(200, "text/html", content);
        return;
    }
} else if (server.arg("POMPA") == "POMPA 2") {
    if (KamarS == "KAMARA") {
        if ((server.arg("NOMINAL").toInt()) <= SisaMeterA) {
            Meter2.reset();
            Meter2Pulse = 0;
            totalMilliLitres2 = 0; // Edit
            //TotalVol = 0; // test SPBU 2-edit
            vPompa = "B";
            StatusMenu = "JRENGA";
            vNominal = server.arg("NOMINAL");
            VoLA = server.arg("NOMINAL").toInt();
            VoLA = ((float) VoLA/ratebb)*1000;
            digitalWrite(RelayB,HIGH);
            updatePeriod = millis();
            String content = hhtml;
            content = "Proses pengisian <br>";
            content = String(VoLA) + "<br>";
            content += "Keluar klik disini <a
href=\\\"/login?DISCONNECT=YES\\\">keluar</a></body></html>";
            server.send(200, "text/html", content);
            return;
        }
        String content = hhtml;
        content = "Saldo lebih kecil dari nominal <br>";
        content += "Keluar klik disini <a
href=\\\"/login?DISCONNECT=YES\\\">keluar</a></body></html>";
        server.send(200, "text/html", content);
        return;
    } else if (KamarS == "KAMARB") {
        if ((server.arg("NOMINAL").toInt()) <= SisaMeterB) {
            Meter2.reset();
            Meter2Pulse = 0;
            totalMilliLitres2 = 0; // Edit
            //TotalVol = 0; // test SPBU 2-edit
            vPompa = "B";
            StatusMenu = "JRENGB";
            vNominal = server.arg("NOMINAL");
            VoLA = server.arg("NOMINAL").toInt();
            VoLA = ((float) VoLA/ratebb)*1000;
            digitalWrite(RelayB,HIGH);
            updatePeriod = millis();
            String content = hhtml;
            content = "Proses pengisian <br>";
            content = String(VoLA) + "<br>";
            content += "Keluar klik disini <a
href=\\\"/login?DISCONNECT=YES\\\">keluar</a></body></html>";

```

```

        server.send(200, "text/html", content);
        return;
    }
    String content = hhtml;
    content = "Saldo lebih kecil dari nominal <br>";
    content += "Keluar klik disini <a
href=\"/login?DISCONNECT=YES\">keluar</a></body></html>";
    server.send(200, "text/html", content);
    return;
}
}
}

if (KamarS == "KAMARA")
    kamar("pelanggana");
else if (KamarS == "KAMARB")
    kamar("pelangganb");

}

void kamar(String vkamar)
{
    String content = hhtml;
    //<html><body style=background-size:cover;background-
image:url('http://3.bp.blogspot.com/-fjcfki4GQ0A/VZFtYi539dl/AAAAAAAAUYo/yN-
164wBMeM/s1600/Water%2BHD%2BWallpapers%2B%2B%25287%2529.jpg');>";
    content += "<form action='/kamar' method='POST'>";
    content += "<H2>Selamat Datang di Sistem Pengisian Bahan Bakar Prabayar</H2><br>";
    content += "<table><tr><th>Lokasi</th><th>Tanggal Transaksi</th><th>Nilai
Transaksi</th><th>Jenis Transaksi</th><th>Saldo</th></tr>";
    String vSisa;

    if (conn.connect(server_addr, 3306, user, passworddb))
        Serial.println("OK.");
    else
        Serial.println("FAILED.");

    // create MySQL cursor object
    cursor = new MySQL_Cursor(&conn);
    delay(200);

    if (vkamar=="pelanggana") {
        char QUERY_POP[]="SELECT lokasi,tanggal_transaksi,nilai_transaksi,jenis_transaksi,saldo
FROM u5620516_arif_nugroho.vpelanggana";
        char query[128];

        MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
        sprintf(query, QUERY_POP, 9000000);
        // Execute the query

        cur_mem->execute(query);

```

```

// Fetch the columns and print them
column_names *cols = cur_mem->get_columns();
for (int f = 0; f < cols->num_fields; f++) {
    Serial.print(cols->fields[f]->name);
    if (f < cols->num_fields-1) {
        Serial.print(',');
    }
}
Serial.println();
int i=1;
// Read the rows and print them
row_values *row = NULL;
do {
    row = cur_mem->get_next_row();
    content += "<tr>";
    if (row != NULL) {
        for (int f = 0; f < cols->num_fields; f++) {
            if (f==3 and i==1) {
                vSisa = String(row->values[f]);
            }
            content += "<td> " + String(row->values[f]) + "</td>";
            Serial.print(row->values[f]);
            if (f < cols->num_fields-1) {
                Serial.print(',');
            }
        }
        i++;
        Serial.println();
    }
    content += "</tr>";
} while (row != NULL);
// Deleting the cursor also frees up memory used
delete cur_mem;
} else
if (vkamar=="pelangganb") {
    char QUERY_POP[]="SELECT lokasi,tanggal_transaksi,nilai_transaksi,jenis_transaksi,saldo
FROM u5620516_arif_nugroho.vpelangganb";
    char query[128];

    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
    sprintf(query, QUERY_POP, 9000000);
    // Execute the query

    cur_mem->execute(query);
    // Fetch the columns and print them
    column_names *cols = cur_mem->get_columns();
    for (int f = 0; f < cols->num_fields; f++) {
        Serial.print(cols->fields[f]->name);
        if (f < cols->num_fields-1) {
            Serial.print(',');
        }
    }
}

```

```

}
Serial.println();
int i=1;
// Read the rows and print them
row_values *row = NULL;
do {
    row = cur_mem->get_next_row();
    content += "<tr>";
    if (row != NULL) {
        for (int f = 0; f < cols->num_fields; f++) {
            if (f==3 and i==1) {
                vSisa = String(row->values[f]);
            }
            content += "<td> " + String(row->values[f]) + "</td>";
            Serial.print(row->values[f]);
            if (f < cols->num_fields-1) {
                Serial.print(',');
            }
        }
        i++;
        Serial.println();
    }
    content += "</tr>";
} while (row != NULL);
// Deleting the cursor also frees up memory used
delete cur_mem;
}

//content += "</table><br><table><tr><td>Saldo anda sebesar :</td><td> Rp. <input
type='text' value="+vSisa+" name='SALDO' placeholder='vsisa' readonly></td></tr>";
content += "<tr><td>Pilih Pompa </td><td><input type='text' name='POMPA'
list='pompa'/> <datalist id='pompa'> <option>POMPA 1</option> <option>POMPA
2</option> </datalist></td></tr>";
content += "<tr><td>Masukkan nominal </td><td> Rp. <input type='text' value=0
name='NOMINAL' placeholder='nominal'></td></tr></table>";
content += "<input type='submit' name='SUBMIT' value='ISI'></form><br>";
content += "Untuk membatalkan klik disini <a
href=\"/login?DISCONNECT=YES\">Batal</a></body></html>";
server.send(200, "text/html", content);

}

void handleMutasia() {
    Serial.println("Enter Operator");
    if (!is_authenticated()) {
        server.sendHeader("Location", "/login");
        server.sendHeader("Cache-Control", "no-cache");
        server.send(301);
        return;
    }
}

```

```

String vSisa;
String content = hhtml;
content += "<table><tr><th>Lokasi</th><th>Tanggal Transaksi</th><th>Nilai
Transaksi</th><th>Jenis Transaksi</th><th>Saldo</th></tr>";
if (conn.connect(server_addr, 3306, user, passworddb))
    Serial.println("OK.");
else
    Serial.println("FAILED.");

// create MySQL cursor object
cursor = new MySQL_Cursor(&conn);
delay(200);

char QUERY_POP[]="SELECT lokasi,tanggal_transaksi,nilai_transaksi,jenis_transaksi,saldo
FROM u5620516_arif_nugroho.vpelanggana";
char query[128];

MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
sprintf(query, QUERY_POP, 9000000);
// Execute the query

cur_mem->execute(query);
// Fetch the columns and print them
column_names *cols = cur_mem->get_columns();
for (int f = 0; f < cols->num_fields; f++) {
    Serial.print(cols->fields[f]->name);
    if (f < cols->num_fields-1) {
        Serial.print(',');
    }
}
Serial.println();
int i=1;
// Read the rows and print them
row_values *row = NULL;
do {
    row = cur_mem->get_next_row();
    content += "<tr>";
    if (row != NULL) {
        for (int f = 0; f < cols->num_fields; f++) {
            if (f==3 and i==1) {
                vSisa = String(row->values[f]);
            }
            content += "<td> " + String(row->values[f]) + "</td>";
            Serial.print(row->values[f]);
            if (f < cols->num_fields-1) {
                Serial.print(',');
            }
        }
        i++;
        Serial.println();
    }
}

```



```

        content += "</tr>";
    } while (row != NULL);
    // Deleting the cursor also frees up memory used
    delete cur_mem;

content += "</table></body></html>";
server.send(200, "text/html", content);
return;

}

void handleMutasib() {
    Serial.println("Enter Operator");
    if (!is_authenticated()) {
        server.sendHeader("Location", "/login");
        server.sendHeader("Cache-Control", "no-cache");
        server.send(301);
        return;
    }

    String vSisa;
    String content = hhtml;
    content += "<table><tr><th>Lokasi</th><th>Tanggal Transaksi</th><th>Nilai  

Transaksi</th><th>Jenis Transaksi</th><th>Saldo</th></tr>";
    if (conn.connect(server_addr, 3306, user, passworddb))
        Serial.println("OK.");
    else
        Serial.println("FAILED.");

    // create MySQL cursor object
    cursor = new MySQL_Cursor(&conn);
    delay(200);

    char QUERY_POP[]="SELECT lokasi,tanggal_transaksi,nilai_transaksi,jenis_transaksi,saldo  

FROM u5620516_arif_nugroho.vpelangganb";
    char query[128];

    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
    sprintf(query, QUERY_POP, 9000000);
    // Execute the query

    cur_mem->execute(query);
    // Fetch the columns and print them
    column_names *cols = cur_mem->get_columns();
    for (int f = 0; f < cols->num_fields; f++) {
        Serial.print(cols->fields[f]->name);
        if (f < cols->num_fields-1) {
            Serial.print(',');
        }
    }
}

```

```

Serial.println();
int i=1;
// Read the rows and print them
row_values *row = NULL;
do {
    row = cur_mem->get_next_row();
    content += "<tr>";
    if (row != NULL) {
        for (int f = 0; f < cols->num_fields; f++) {
            if (f==3 and i==1) {
                vSisa = String(row->values[f]);
            }
            content += "<td> " + String(row->values[f]) + "</td>";
            Serial.print(row->values[f]);
            if (f < cols->num_fields-1) {
                Serial.print(',');
            }
        }
        i++;
        Serial.println();
    }
    content += "</tr>";
} while (row != NULL);
// Deleting the cursor also frees up memory used
delete cur_mem;

content += "</table></body></html>";
server.send(200, "text/html", content);
return;

}

void handleReset() {
    if (!is_authenticated()) {
        server.sendHeader("Location", "/login");
        server.sendHeader("Cache-Control", "no-cache");
        server.send(301);
        return;
    }

    if (conn.connect(server_addr, 3306, user, passworddb)) {
        char QUERY_DEL[]="DELETE FROM u5620516_arif_nugroho.voucher";
        MySQL_Connection conn(&client);
        MySQL_Cursor* cursor;

        cursor = new MySQL_Cursor(&conn);

        if (conn.connected())
            cursor->execute(QUERY_DEL);

        delay(200);
    }
}

```

```

EEPROMWritelong(0,0);
delay(200);
EEPROMWritelong(4,0);
delay(200);
EEPROMWritelong(8,0);
delay(200);
EEPROMWritelong(12,0);
delay(200);
EEPROM.commit();

SisaMeterA = 0;
SisaMeterB = 0;

String content = hhtml;
content += "Reset berhasil <br>";
content += "Untuk membatalkan klik disini <a
href=\"/login?DISCONNECT=YES\">Batalkan</a></body></html>";
server.send(200, "text/html", content);
return;
} else {
String content = hhtml;
content += "Reset gagal <br>";
content += "Untuk membatalkan klik disini <a
href=\"/login?DISCONNECT=YES\">Batalkan</a></body></html>";
server.send(200, "text/html", content);
return;
}
}

void handleOperator() {

Serial.println("Enter Operator");
//String header;
if (!is_authenticated()) {
server.setHeader("Location", "/login");
server.setHeader("Cache-Control", "no-cache");
server.send(301);
return;
}

if (server.hasArg("NOMINAL")) {
if (server.arg("NOMINAL").toInt() > 0) {
String content = hhtml;
//content += "<body style=background-size:cover;background-
image:url('http://3.bp.blogspot.com/-fjcfki4GQ0A/VZFtYi539dl/AAAAAAAAUYo/yN-
164wBMeM/s1600/Water%2BHD%2BWallpapers%2B%2B%25287%2529.jpg');>";

```

```

        content += "<form action='/operator' method='POST'><H2>Top up akan diisi sebesar
:</H2><br><H3>";
        content += "<table><tr><td>Pelanggan </td>";
        content += "<td><input type='text' name='KAMAR' value='' +
String(server.arg(\"KAMAR\")) + \"' placeholder='kamar' readonly></td></tr>";
        content += "<tr><td>Top Up senilai</td>";
        content += "<td><input type='text' name='NOMINAL2' value='' +
String(server.arg(\"NOMINAL\")) + \"' placeholder='nominal2' readonly></td></tr>";
        //content += "<tr><td>Kode Pengisian</td>";
        content += "<td><input type='hidden' name='KODE' value='' +
String(random(1000000,2000000)) + \"' placeholder='kode' readonly></td></tr>";
        //content += "<tr><td>Volume bahan bakar yang didapat sebesar</td><td>" +
String((server.arg(\"NOMINAL\").toInt()/rateair)*1000) + "Liter </td></tr>";
        content += "</table><input type='submit' name='SUBMIT' value='Simpan
Data'></form> <br>";
        content += "Untuk membatalkan klik disini <a
href='/login?DISCONNECT=YES'>Batalkan</a></body></html>";
        server.send(200, "text/html", content);
        return;
    }
}

```

```

if (server.hasArg("KODE")) {
    int kode = server.arg("KODE").toInt();
    int nominal = server.arg("NOMINAL2").toInt();
    String kamar = server.arg("KAMAR");
    String content = hhtml;
    //"<html><body style=background-size:cover;background-
image:url('http://3.bp.blogspot.com/-fjcfki4GQ0A/VZFtYi539dl/AAAAAAAAUYo/yN-
164wBMeM/s1600/Water%2BHD%2BWallpapers%2B%2B%25287%2529.jpg');>";
    Serial.print("Isi Senilai :");Serial.println(String(nominal));
    if (kamar == "PELANGGAN A") {
        SisaMeterA = nominal+SisaMeterA;
        if (conn.connect(server_addr, 3306, user, passworddb)) {
            char INSERT_DATA[]="insert into u5620516_arif_nugroho.voucher
(pelanggan,nilai_transaksi,saldo,jenis_transaksi) values('%s','%d','%d','%s)";
            char query[128];
            Serial.println("OK.");
            delay(1000);
            // Initiate the query class instance
            MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);

            sprintf(query, INSERT_DATA, "pelanggana", nominal,SisaMeterA,"TOPUP SALDO");
            // Execute the query
            cur_mem->execute(query);
            delete cur_mem;

            delay(200);
            EEPROMWritelong(0,kode);
            delay(200);
            EEPROMWritelong(4,SisaMeterA);

```

```

        delay(200);
        content += "<H2>Data suda tersimpan!</H2><br><H3>";
    }
    else {
        Serial.println("FAILED.");
        content += "<H2>Data gagal tersimpan!</H2><br><H3>";
    }
} else if (kamar == "PELANGGAN B") {
    SisaMeterB = nominal+SisaMeterB;
    if (conn.connect(server_addr, 3306, user, passworddb)) {
        char INSERT_DATA[]="insert into u5620516_arif_nugroho.voucher
(pelanggan,nilai_transaksi,saldo,jenis_transaksi) values('%s','%d','%d','%s)";
        char query[128];
        Serial.println("OK.");
        delay(1000);
        // Initiate the query class instance
        MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);

        sprintf(query, INSERT_DATA, "pelangganb", nominal,SisaMeterB,"TOPUP SALDO");
        // Execute the query
        cur_mem->execute(query);
        delete cur_mem;

        delay(200);
        EEPROMWritelong(8,kode);
        delay(200);
        EEPROMWritelong(12,SisaMeterB);
        delay(200);
        content += "<H2>Data suda tersimpan!</H2><br><H3>";
    }
    else {
        Serial.println("FAILED.");
        content += "<H2>Data gagal tersimpan!</H2><br><H3>";
    }
}

EEPROM.commit();
content += "Untuk pelanggan " + kamar + "<br>";
content += "Untuk mengisi kembali klik disini <a href=\"/operator\">Isi
Kembali</a><br>";
content += "Keluar klik disini <a
href=\"/login?DISCONNECT=YES\">keluar</a></H3></body></html>";

server.send(200, "text/html", content);
return;
}

```

String content = hhtml;

```

//content += "<body style=background-size:cover;background-
image:url('http://3.bp.blogspot.com/-fjcfki4GQ0A/VZFtYi539dl/AAAAAAAAUYo/yN-
164wBMeM/s1600/Water%2BHD%2BWallpapers%2B%2B%25287%2529.jpg');>";

content += "<form action='/operator' method='POST'>";
content += "<H2>Selamat Datang di Sistem Pengisian Bahan Bakar Prabayar</H2><br>";

content += "<table border=2 bordercolor=#0000FF><tr><th colspan=2><H2>Sisa Nominal
Pelanggan</H2></th></tr>";
content += "<tr><th><H2>Pelanggan A</H2></th><th><H2>Pelanggan
B</H2></th></tr>";
content += "<tr><td>" + String(SisaMeterA);

content += "</td>";
content += "<td>" + String(SisaMeterB);

//content += "</td></tr><tr><td><a href='\"/mutasi\">Keluar</a></td>";
content += "</td></tr><tr><td><a href='\"/mutasia\">Mutasi</a></td>";
content += "<td> <a href='\"/mutasib\">Mutasi</a></td></tr>";
content += "</table>";
content += "Untuk melakukan Top Up isi data berikut, kemudian tekan tombol Top Up :";
content += "<table> <tr><td>";
content += "Top Up Pelanggan</td><td><input type='text' name='KAMAR' list='kamar'/">
<datalist id='kamar'> <option>PELANGGAN A</option> <option>PELANGGAN B</option>
</datalist></td></tr>";
content += "<tr><td>Top Up Senilai</td><td><input type='text' value=0 name='NOMINAL'
placeholder='nominal'></td></tr>";
content += "</table>";
content += "<input type='submit' name='SUBMIT' value='Top Up'></form><br>";
content += "Untuk membatalkan klik disini <a
href='\"/login?DISCONNECT=YES\">Keluar</a></body></html>";
server.send(200, "text/html", content);
// Free resources

}

void EEPROMWritelong(int address, long value)
{
byte four = (value & 0xFF);
byte three = ((value >> 8) & 0xFF);
byte two = ((value >> 16) & 0xFF);
byte one = ((value >> 24) & 0xFF);

//Write the 4 bytes into the eeprom memory.
EEPROM.write(address, four);
EEPROM.write(address + 1, three);
EEPROM.write(address + 2, two);
EEPROM.write(address + 3, one);
}

long EEPROMReadlong(long address)

```

```

    {
        long four = EEPROM.read(address);
        long three = EEPROM.read(address + 1);
        long two = EEPROM.read(address + 2);
        long one = EEPROM.read(address + 3);

        //Return the recomposed long by using bitshift.
        return ((four << 0) & 0xFF) + ((three << 8) & 0xFFFF) + ((two << 16) & 0xFFFFF) + ((one
        << 24) & 0xFFFFFFF);
    }

```

```

//no need authentication
void handleNotFound() {
    String message = "File Not Found\n\n";
    message += "URI: ";
    message += server.uri();
    message += "\nMethod: ";
    message += (server.method() == HTTP_GET) ? "GET" : "POST";
    message += "\nArguments: ";
    message += server.args();
    message += "\n";
    for (uint8_t i = 0; i < server.args(); i++) {
        message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
    }
    server.send(404, "text/plain", message);
}

```

```

void setup(void) {
    hhtml = "<html><head><title>Smart Gasoline Station</title> <link rel='icon'
href='https://icon-library.com/images/petrol-icon/petrol-icon-1.jpg' type = 'image/x-
icon'>";
    hhtml += "<style>#card {background: #fbfbfb;border-radius: 8px;box-shadow: 1px 2px 8px
rgba(0, 0, 0, 0.65);height: 410px;margin: 6rem auto 8.1rem auto;width: 329px;}";
    hhtml += "#card-content {padding: 12px 44px;} #card-title {font-family: 'Raleway Thin',
sans-serif;letter-spacing: 4px;padding-bottom: 23px;padding-top: 13px;text-align: center;}";
    hhtml += ".underline-title {background: -webkit-linear-gradient(right, #a6f77b,
#2ec06f);height: 2px;margin: -1.1rem auto 0 auto;width: 89px;}";
    hhtml += "a {text-decoration: none;} label {font-family: 'Raleway', sans-serif; font-size:
11pt;}";
    hhtml += "#forgot-pass {color: #2dbd6e;font-family: 'Raleway', sans-serif;font-size:
10pt;margin-top: 3px;text-align: right;}";
    hhtml += ".form {align-items: left;display: flex;flex-direction: column;}";
    hhtml += ".form-border {background: -webkit-linear-gradient(right, #a6f77b,
#2ec06f);height: 1px;width: 100%;}";
    hhtml += ".form-content {background: #fbfbfb;border: none;outline: none;padding-top:
14px;}";
    hhtml += "#signup {color: #2dbd6e;font-family: 'Raleway', sans-serif;font-size:
10pt;margin-top: 16px;text-align: center;}";

```



```
html += "#submit-btn {background: -webkit-linear-gradient(right, #a6f77b,  
#2dbd6e);border: none;border-radius: 21px;box-shadow: 0px 1px 8px #24c64f;cursor:  
pointer;";  
html += "color: white;font-family: 'Raleway SemiBold', sans-serif;height: 42.3px;margin: 0  
auto;margin-top: 50px;transition: 0.25s;width: 153px; justify-content: center;}";  
html += "#submit-btn:hover {box-shadow: 0px 1px 18px #24c64f;}";  
html += "</style></head>";  
html += "<body style=background-size:cover;background-  
image:url('https://indonesiaexpat.biz/wp-content/uploads/2020/01/IMG_0193.jpg');>";  
//https://images.wisegeek.com/gas-station-at-night.jpg');>";
```

```
randomSeed(1);  
EEPROM.begin(512);
```

```
Serial.begin(115200);  
lcd.begin(16,2);  
lcd.setBacklight  
(100);  
// delay(5000);  
// lcd.clear();
```

```
WiFi.mode(WIFI_STA);  
WiFi.hostname("ESP8266");  
WiFi.begin(ssid, password);  
Serial.println("");
```

```
// Wait for connection  
lcd.setCursor(0, 0);  
lcd.print("Connecting to ");  
lcd.setCursor(0, 1);  
int vloop = 0;  
while (WiFi.status() != WL_CONNECTED) {  
  delay(500);  
  lcd.print(".");  
  Serial.print(".");  
  vloop += 1;  
  if (vloop >= 17) {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Connected to ");  
    lcd.setCursor(0, 1);  
    vloop = 1;  
  }  
}  
Serial.println("");  
Serial.print("Connected to ");
```

```
Wire.begin(D2,D1);  
lcd.clear();  
Serial.println(ssid);  
lcd.setCursor(0, 1);
```

```

lcd.print(ssid);
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("IP Address");
lcd.setCursor(0, 1);
lcd.print(WiFi.localIP());
delay(5000);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
Serial.print("Connecting to DB... ");

if (conn.connect(server_addr, 3306, user, passworddb))
  Serial.println("OK.");
else
  Serial.println("FAILED.");

// create MySQL cursor object
cursor = new MySQL_Cursor(&conn);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Initial Data A");

//Initial Pelanggan A
if (EEPROMReadlong(4) < 0)
  SisaMeterA = 0;
else
  SisaMeterA = EEPROMReadlong(4);
  Serial.print("Sisa Pelanggan A :");Serial.println(SisaMeterA);
//Initial Pelanggan B
if (EEPROMReadlong(12) < 0)
  SisaMeterB = 0;
else
  SisaMeterB = EEPROMReadlong(12);
  Serial.print("Sisa Pelanggan B :");Serial.println(SisaMeterB);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Untuk Masuk Sistem");
lcd.setCursor(0, 1);
lcd.print("Tekan Hurf A");

I2C_Keypad.begin();

server.on("/", handleRoot);
server.on("/login", handleLogin);
server.on("/operator", handleOperator);
server.on("/mutasia", handleMutasia);
server.on("/mutasib", handleMutasib);
server.on("/reset", handleReset);
server.on("/kamar", handleKamar);

```

```
server.on("/inline", []() {
  server.send(200, "text/plain", "this works without need of authentication");
});
```

```
server.onNotFound(handleNotFound);
//here the list of headers to be recorded
const char * headerkeys[] = {"User-Agent", "Cookie"};
size_t headerkeyssize = sizeof(headerkeys) / sizeof(char*);
//ask server to track these headers
server.collectHeaders(headerkeys, headerkeyssize);
server.begin();
Serial.println("HTTP server started");
```

```
pinMode(RelayA, OUTPUT);
pinMode(RelayB, OUTPUT);
pinMode(BuzzerA, OUTPUT);
pinMode(BuzzerB, OUTPUT);
pinMode(D3, INPUT);
pinMode(D4, INPUT);
digitalWrite(D3, HIGH); // Edit
digitalWrite(D4, HIGH); // Edit
//attachInterrupt(D3, Meter1ISR, RISING); // test SPBU 2-edit
//attachInterrupt(D4, Meter2ISR, RISING); // test SPBU 2-edit
attachInterrupt(D3, Meter1ISR, FALLING); // Edit
attachInterrupt(D4, Meter2ISR, FALLING); // Edit
Meter1Pulse = 0; // Edit
Meter2Pulse = 0; // Edit
flowRate1 = 0.0; // Edit
flowRate2 = 0.0; // Edit
flowMilliLitres1 = 0; // Edit
flowMilliLitres2 = 0; // Edit
totalMilliLitres1 = 0; // Edit
totalMilliLitres2 = 0; // Edit
updatePeriod = 0; // Edit
// sometimes initializing the gear generates some pulses that we should ignore
Meter1.reset();
Meter2.reset();
updatePeriod = millis();
BuzzerOnPeriod = millis();
}
```

```
void loop(void) {
  server.handleClient();
  HTTPClient http;
  int vLiter;

  if (VolA >= 0)
  {
    digitalWrite(BuzzerA,LOW);
```

```

} else
{
digitalWrite(BuzzerA,HIGH);
delay(200);
digitalWrite(BuzzerA,LOW);
}

if ((StatusMenu == "JRENGA") && (VoIA >= 0)) {
if((millis() - updatePeriod) > 1000) // Only process counters once per second
{
if (vPompa == "A") {
detachInterrupt(D3); // Edit
flowRate1 = ((1000.0 / (millis() - updatePeriod)) * Meter1Pulse) / calibrationFactorA; //
Edit
updatePeriod = millis(); // Edit
//flowMilliLitres1 = (flowRate1 / 60) * 1000; // Edit
flowMilliLitres1 = (3.8 / 60) * 1000; // Edit hasil percobaan didapatkan 63.3 ml/s
totalMilliLitres1 += flowMilliLitres1; // Edit
VoIA = VoIA - flowMilliLitres1; // Edit
//Meter1.tick(period); // test SPBU 2-edit
//TotalVol += ((Meter1.getCurrentVolume()/calibrationFactorA) * 1000.0f); // test
SPBU 2-edit
//VoIA = VoIA - ((Meter1.getCurrentVolume()/calibrationFactorA) * 1000.0f); // test
SPBU 2-edit
//VoIA = VoIA - totalMilliLitres1; // Edit
//unsigned int frac; // Edit
vLiter = ((float)vNominal.toInt()/ratebb)*1000;
lcd.clear();
String rupiah = "Rp. "+vNominal;
lcd.setCursor(16-rupiah.length(),0);
lcd.print(rupiah);
String totalan = String(totalMilliLitres1) + " ml"; // Edit
//String totalan = String(TotalVol) + " ml"; // test SPBU 2-edit
lcd.setCursor(16-totalan.length(),1);
lcd.print(totalan);
Meter1Pulse = 0; // Edit
attachInterrupt(D3, Meter1ISR, FALLING); // Edit
updatePeriod = millis(); // Edit
} else if (vPompa == "B") {
detachInterrupt(D4); // Edit
flowRate2 = ((1000.0 / (millis() - updatePeriod)) * Meter2Pulse) / calibrationFactorB; //
Edit
updatePeriod = millis(); // Edit
//flowMilliLitres2 = (flowRate2 / 60) * 1000; // Edit
flowMilliLitres2 = (4.45 / 60) * 1000; // Edit hasil percobaan didapatkan 74.1 ml/s
totalMilliLitres2 += flowMilliLitres2; // Edit
//unsigned int frac; // Edit
//Meter2.tick(period); // test SPBU 2-edit
//TotalVol += ((Meter2.getCurrentVolume()/calibrationFactorB) * 1000.0f); // test
SPBU 2-edit

```

```

    //VolA = VolA - ((Meter2.getCurrentVolume()/calibrationFactorB) * 1000.0f); // test
SPBU 2-edit
    VolA = VolA - flowMilliLitres2; // Edit
    vLiter = ((float)vNominal.toInt()/ratebb)*1000;
    lcd.clear();
    String rupiah = "Rp. "+vNominal;
    lcd.setCursor(16-rupiah.length(),0);
    lcd.print(rupiah);
    //VolB += 107;
    //String totalan = String(VolB) + " ml";
    String totalan = String(totalMilliLitres2) + " ml"; // Edit
    //String totalan = String(TotalVol) + " ml"; // test SPBU 2-edit
    lcd.setCursor(16-totalan.length(),1);
    lcd.print(totalan);
    Meter2Pulse = 0; // Haris Edit
    attachInterrupt(D4, Meter2ISR, FALLING); // Edit
    updatePeriod = millis(); // Edit
    //updatePeriod = millis(); // test SPBU 2-edit
}
}
} if ((StatusMenu == "JRENGB") && (VolA >= 0)) {
    if((millis() - updatePeriod) > 1000) // Only process counters once per second
    {
        if (vPompa == "A") {
            detachInterrupt(D3); // Edit
            flowRate1 = ((1000.0 / (millis() - updatePeriod)) * Meter1Pulse) / calibrationFactorA; //
Edit
            updatePeriod = millis(); // Edit
            //flowMilliLitres1 = (flowRate1 / 60) * 1000; // Edit
            flowMilliLitres1 = (3.8 / 60) * 1000; // Edit Hasil percobaan didapatkan 63.3 ml/s
            totalMilliLitres1 += flowMilliLitres1; // Edit
            VolA = VolA - flowMilliLitres1; // Edit
            //unsigned int frac; // Edit
            //Meter1.tick(period); // test SPBU 2-edit
            //TotalVol += ((Meter1.getCurrentVolume()/calibrationFactorA) * 1000.0f); // test
SPBU 2-edit
            //VolA = VolA - ((Meter1.getCurrentVolume()/calibrationFactorA) * 1000.0f); // test
SPBU 2-edit
            vLiter = ((float)vNominal.toInt()/ratebb)*1000;
            lcd.clear();
            String rupiah = "Rp. "+vNominal;
            lcd.setCursor(16-rupiah.length(),0);
            //lcd.setCursor(16-String(vLiter).length()-1,0);
            //lcd.print(String(vLiter));
            lcd.print(rupiah);
            String totalan = String(totalMilliLitres1) + " ml"; // Edit
            //String totalan = String(TotalVol) + " ml"; // test SPBU 2-edit
            lcd.setCursor(16-totalan.length(),1);
            lcd.print(totalan);
            Meter1Pulse = 0; // Edit
            attachInterrupt(D3, Meter1ISR, FALLING); // Edit

```

```

        updatePeriod = millis(); // Edit
        //updatePeriod = millis(); // test SPBU 2-edit
    } else if (vPompa == "B") {
        detachInterrupt(D4); // Edit
        flowRate2 = ((1000.0 / (millis() - updatePeriod)) * Meter2Pulse) / calibrationFactorB; //
Edit
        updatePeriod = millis(); // Edit
        //flowMilliLitres2 = (flowRate2 / 60) * 1000; // Edit
        flowMilliLitres2 = (4.45 / 60) * 1000; // Edit Hasil percobaan didapatkan 74.1 ml/s
        totalMilliLitres2 += flowMilliLitres2; // Edit
        VolA = VolA - flowMilliLitres2; // Edit
        //unsigned int frac; // Edit
        //Meter2.tick(period); // test SPBU 2-edit
        //TotalVol += ((Meter2.getCurrentVolume()/calibrationFactorB) * 1000.0f); // test
SPBU 2-edit
        //VolA = VolA - ((Meter2.getCurrentVolume()/calibrationFactorB) * 1000.0f); // test
SPBU 2-edit
        vLiter = ((float)vNominal.toInt()/ratebb)*1000;
        lcd.clear();
        String rupiah = "Rp. "+vNominal;
        lcd.setCursor(16-rupiah.length(),0);
        //lcd.setCursor(16-String(vLiter).length()-1,0);
        //lcd.print(String(vLiter));
        lcd.print(rupiah);
        String totalan = String(totalMilliLitres2) + " ml"; // Edit
        //String totalan = String(TotalVol) + " ml"; // test SPBU 2-edit
        lcd.setCursor(16-totalan.length(),1);
        lcd.print(totalan);
        Meter2Pulse = 0; // Edit
        attachInterrupt(D4, Meter2ISR, FALLING); // Edit
        updatePeriod = millis(); // Edit
        //updatePeriod = millis(); // test SPBU 2-edit
    }
}
} else if ((StatusMenu == "JRENGA") && (VolA <= 0))
{
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Update data");
    if (conn.connect(server_addr, 3306, user, passworddb)) {
        char INSERT_DATA[]="insert into u5620516_arif_nugroho.voucher
(pelanggan,nilai_transaksi,saldo,jenis_transaksi) values('%s','%d','%d','%s')";
        char query[128];
        Serial.println("OK.");
        delay(1000);
        // Initiate the query class instance
        MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
        SisaMeterA = SisaMeterA-vNominal.toInt();
        if (vPompa == "A") {
            digitalWrite(RelayA,LOW);

```

```

        sprintf(query, INSERT_DATA, "pelanggan", vNominal.toInt(),SisaMeterA,"ISI\~DISP-
1");
    }
    else if (vPompa == "B") {
        digitalWrite(RelayB,LOW);
        sprintf(query, INSERT_DATA, "pelanggan", vNominal.toInt(),SisaMeterA," ISI\~DISP-
2");
    }
    // Execute the query
    cur_mem->execute(query);
    delete cur_mem;

    delay(200);
    EEPROMWritelong(4,SisaMeterA);
    delay(200);
    EEPROM.commit();
    lcd.setCursor(0,1);
    lcd.print("berhasil");
    StatusMenu = "pass";
}
} else if ((StatusMenu == "JRENGB") && (VoIA <= 0))
{
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Update data");
    if (conn.connect(server_addr, 3306, user, passworddb)) {
        char INSERT_DATA[]="insert into u5620516_arif_nugroho.voucher
(pelanggan,nilai_transaksi,saldo,jenis_transaksi) values('%s','%d','%d','%s)";
        char query[128];
        Serial.println("OK.");
        delay(1000);
        // Initiate the query class instance
        MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
        SisaMeterB = SisaMeterB-vNominal.toInt();
        if (vPompa == "A") {
            digitalWrite(RelayA,LOW);
            sprintf(query, INSERT_DATA, "pelangganb", vNominal.toInt(),SisaMeterB,"ISI\~DISP-
1");
        }
        else if (vPompa == "B") {
            digitalWrite(RelayB,LOW);
            sprintf(query, INSERT_DATA, "pelangganb", vNominal.toInt(),SisaMeterB,"ISI\~DISP-
2");
        }
        // Execute the query
        cur_mem->execute(query);
        delete cur_mem;

        delay(200);
        EEPROMWritelong(12,SisaMeterB);

```



```

    delay(200);
    EEPROM.commit();
    lcd.setCursor(0,1);
    lcd.print("berhasil");
    StatusMenu = "pass";
}
}

char keyKeypad = I2C_Keypad.getKey(); //storing pressed key value in a char
//Akses KeyPad
if (keyKeypad != NO_KEY)
{
    TimeKeypad = millis();
    if ((stateKeypad == false) && (keyKeypad == 'A'))
    {
        stateKeypad = true;
        lcd.clear();
        keyKeypad = '*';
    }

    if (stateKeypad == true)
    {

        if (keyKeypad == '*') { /*
            if (StatusMenu == "ISIBBA") {
                StatusMenu = "Menu";
                vNominal = "";
                digitalWrite(RelayA,LOW);
                digitalWrite(RelayB,LOW);
                //keyKeypad = 'A';
            } else */{
                lcd.clear();
                StatusMenu = "pass";
                vpassword = "";
                lcd.setCursor(0,0);
                lcd.print("Masukkan Password");
                lcd.setCursor(0,1);
            }
        } else
        if (keyKeypad == '#') {
            if (StatusMenu == "pass") {
                if ((password_lcd_A == vpassword)) {
                    lcd.clear();
                    StatusMenu = "KAMARA";
                    vpassword = "";
                    KamarS = "KAMARA";
                    lcd.setCursor(0,0);
                    lcd.print("Info Saldo > A");
                    lcd.setCursor(0,1);
                    lcd.print("Isi Bensin > B");
                }
            }
        }
    }
}

```

```

else if (password_lcd_B == vpassword){
    lcd.clear();
    StatusMenu = "KAMARB";
    vpassword = "";
    KamarS = "KAMARB";
    lcd.setCursor(0,0);
    lcd.print("Info Saldo > A");
    lcd.setCursor(0,1);
    lcd.print("Isi Bensin > B");

}
else {
    lcd.clear();
    StatusMenu = "pass";
    vpassword = "";
    lcd.setCursor(0,0);
    lcd.print("Masukkan Password");
    lcd.setCursor(0,1);
}
} //end if pass
else if (StatusMenu == "ISIBBA") {
    if (vNominal.toInt() == 0) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Nominal salah ");
        delay(500);
        keyKeypad = '*';
    } else
    if (SisaMeterA < vNominal.toInt())
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Saldo kurang");
        delay(200);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("S : Rp. " +String(SisaMeterA));
        lcd.setCursor(0,1);
        lcd.print("I : Rp. " +String(vNominal.toInt()));
        vNominal = "";
        StatusMenu = "ISIA";
        keyKeypad = 'A';
    } else {
        TotalVol = 0;
        VolA = vNominal.toInt();
        VolA = ((float) VolA/ratebb)*1000;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(String(VolA));
        if (vPompa == "A")
            digitalWrite(RelayA,HIGH);
    }
}

```

```

        else digitalWrite(RelayB,HIGH);
        StatusMenu = "JRENGA";
    }
}
else if (StatusMenu == "ISIBBB") {
    if (vNominal.toInt() == 0) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Nominal salah ");
        delay(500);
        keyKeypad = '*';
    } else
    if (SisaMeterB < vNominal.toInt())
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Saldo kurang");
        delay(200);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("S : Rp. " + String(SisaMeterB));
        lcd.setCursor(0,1);
        lcd.print("I : Rp. " + String(vNominal.toInt()));
        vNominal = "";
        StatusMenu = "ISIB";
        keyKeypad = 'A';
    } else {
        VoIA = vNominal.toInt();
        VoIA = ((float) VoIA/ratebb)*1000;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(String(VoIA));
        if (vPompa == "A")
            digitalWrite(RelayA,HIGH);
        else digitalWrite(RelayB,HIGH);
        StatusMenu = "JRENGB";
    }
}
} else if ((keyKeypad == 'A') && (StatusMenu == "KAMARA")) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Saldo anda");
    lcd.setCursor(0,1);
    lcd.print("Rp. ");lcd.print(SisaMeterA);
    StatusMenu = "KAMARA";
} else if ((keyKeypad == 'A') && (StatusMenu == "KAMARB")) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Saldo anda");
    lcd.setCursor(0,1);
    lcd.print("Rp. ");lcd.print(SisaMeterB);

```

```

    StatusMenu = "KAMARB";
} else if ((keyKeypad == 'B') && (StatusMenu == "KAMARB")) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("PUMP 1 > A");
    lcd.setCursor(0,1);
    lcd.print("PUMP 2 > B");
    StatusMenu = "ISIB";
} else if ((keyKeypad == 'B') && (StatusMenu == "KAMARA")) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("PUMP 1 > A");
    lcd.setCursor(0,1);
    lcd.print("PUMP 2 > B");
    StatusMenu = "ISIA";
} else if ((StatusMenu == "ISIA") && (keyKeypad == 'A')){
    vPompa = "A";
    Meter1.reset();
    VolA = 0;
    VolB = 0;
    vNominal = "";
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Masukkan Nominal");
    lcd.setCursor(0,1);
    StatusMenu="ISIBBA";
} else if ((StatusMenu == "ISIA") && (keyKeypad == 'B')){
    vPompa = "B";
    VolA = 0;
    VolB = 0;
    vNominal = "";
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Masukkan Nominal");
    lcd.setCursor(0,1);
    StatusMenu="ISIBBA";
} else if ((StatusMenu == "ISIB") && (keyKeypad == 'A')){
    vPompa = "A";
    VolA = 0;
    VolB = 0;
    vNominal = "";
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Masukkan Nominal");
    lcd.setCursor(0,1);
    StatusMenu="ISIBBB";
} else if ((StatusMenu == "ISIB") && (keyKeypad == 'B')){
    vPompa = "B";
    vNominal = "";
    VolA = 0;
    VolB = 0;

```

```

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Masukkan Nominal");
        lcd.setCursor(0,1);
        StatusMenu="ISIBBB";
    }
    else {
        vNominal += keyKeypad;
        vpassword += keyKeypad;

        lcd.print(keyKeypad);
    }

}

else
{
    if (((millis() - TimeKeypad) > 60000) && (StatusMenu != "JRENG"))
    { /*
        lcd.noBacklight();
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Masuk Sistem");
        lcd.setCursor(0,1);
        lcd.print("Tekan Huruf A");
        stateKeypad = false;
    */}
    else
        lcd.setBacklight(100);
}

}

void OffMeteran(int Kamar,int vSisa)
{
    if (vSisa <= 20)
    {
        Serial.print("Habis");
        digitalWrite(Kamar,HIGH);
    }
    else digitalWrite(Kamar,LOW);
}

```