

III.3 Realisasi

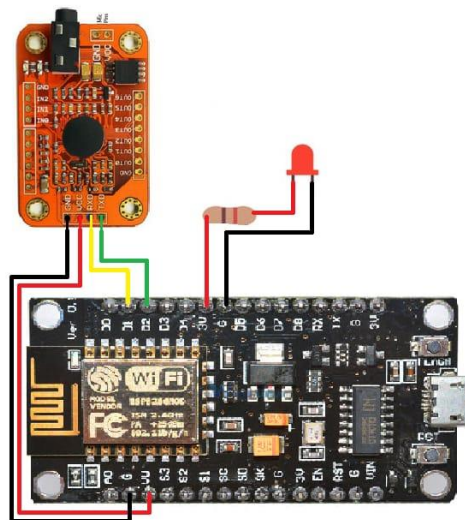
Bagian ini merupakan realisasi dari bagian blok diagram beberapa bagian yang sudah dikerjakan meliputi realisasi perangkat keras dan perangkat lunak.

III.3.1 Realisasi Perangkat Keras

Pada bagian ini merupakan tahapan realisasi perangkat keras meliputi realisasi perangkat PCB, realisasi perakitan, realisasi pengkabelan dan realisasi mekanik pendukung.

III.3.1.1 Realisasi Pengkabelan

Bagian ini merupakan realisasi pengkabelan dari konsep blok diagram yang telah dibuat untuk memperlihatkan integrasi mikrokontroler dengan modul yang digunakan pada proyek tugas akhir ini.



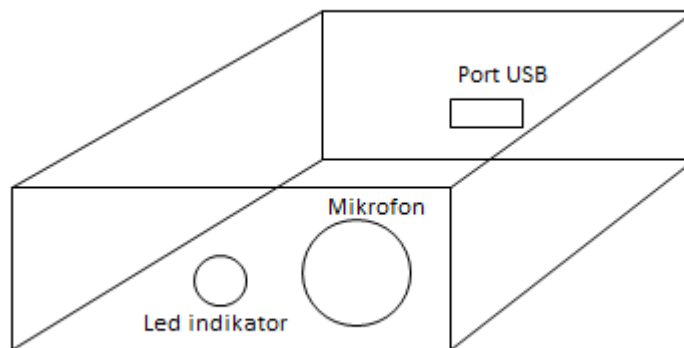
Gambar III.9 Wiring Diagram Perangkat *Hardware*

Gambar III.9 menunjukkan integrasi *Voltage USB (VU)* dan Gnd dari Node MCU ESP8266 ke pin Vcc dan Gnd pada modul *Voice Recognition V3*. Kemudian pin GPIO5 (D1) ke Rx dan pin GPIO4 (D2) ke Tx sebagai komunikasi data serial

dan dipasang sebuah led sebagai indikator yang terhubung seri dengan resistor 330 ohm ke 3,3 V pada Node MCU ESP8266.

III.3.1.2 Realisasi Mekanik Pendukung

Pada realisasi perangkat *hardware* pengintegrasian mikrokontroler dengan modul *Voice Recognition V3* akan dikemas menggunakan kotak hitam sebagai casing dengan ukuran $P \times L \times T = 8,5 \text{ cm} \times 7,5 \text{ cm} \times 3,5 \text{ cm}$, kemasan tersebut didesain menggunakan *software Microsoft word 2010*.

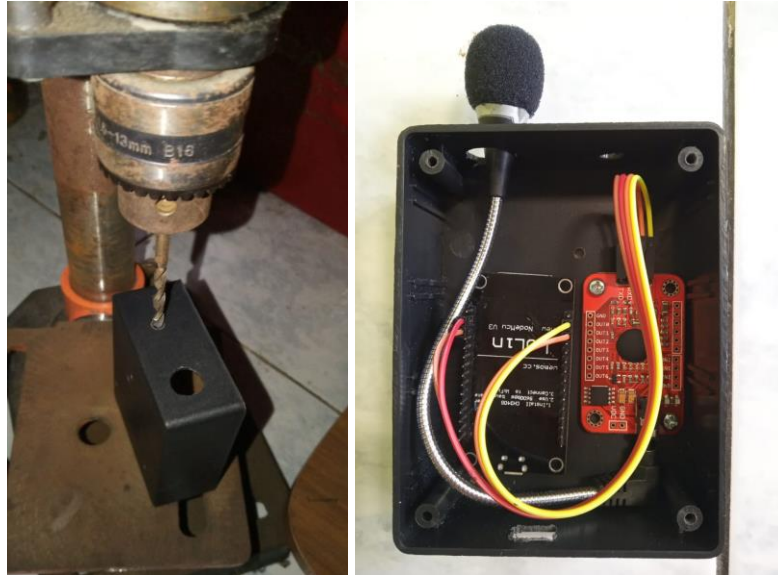


Gambar III.10 Konsep Casing

Gambar III.10 merupakan konsep casing yang dibuat untuk mengemas perangkat *hardware*. Pada konsep tersebut ditunjukkan bagian casing disesuaikan dengan kebutuhan modul yang digunakan.

III.3.1.3 Realisasi Kemasan Alat

Bagian ini menjelaskan realisasi kemasan alat meliputi proses pembuatan casing alat hingga jadi dan dapat digunakan tahapan yang dilakukan dengan menyesuaikan tata letak komponen atau modul dengan kotak hitam yang digunakan kemudian dilakukan proses *drilling* atau proses pengeboran pada casing untuk tata letak komponen pada modul.



Gambar III.11 Proses *Drilling* dan Uji Casing



Gambar III.12 Kemasan Perangkat Hardware

III.3.2 Realisasi Perangkat Lunak

Pada bagian ini merupakan tahapan realisasi perangkat lunak meliputi realisasi program yang di upload pada *hardware*, realisasi program pembuatan aplikasi dan realisasi *database*.

III.3.2.1 Realisasi Program *Hardware*.

```
void printSigTrain(uint8_t *buf, uint8_t len)
{
    if(len == 0){
        Serial.println(F("Train With Signature Finish."));
        return;
    }
    else{
        Serial.print(F("Success: "));
        Serial.println(buf[0], DEC);
    }
    Serial.print(F("Record "));
    Serial.print(buf[1], DEC);
    Serial.print(F("\t"));
    switch(buf[2]){
        case 0:
            Serial.println(F("Trained"));
            break;
    }
}
```

Gambar III.13 Program *Sigtrain*

Program sigtrain merupakan program untuk mengsetting kata yang diucapkan sebagai perintah suara yang akan digunakan sintak if-else digunakan untuk membandingkan variable dengan suatu logika, bila variable bernilai benar maka pernyataan 1 akan dieksekusi bila tidak (else) maka pernyataan 2 akan dieksekusi. `if(len == 0) Serial.println(F("Train With Signature Finish."))` perintah kondisi untuk merecord dan apabila tidak merecord maka else untuk menyimpan dengan print "success" di deklarasikan pada `(buf[0], DEC)` ketika merecord kata perintah dideklarasikan pada `(buf[1], DEC)`; dan pada switch case 0 membandingkan dvariabel dengan beberapa konstanta dan jika sesuai maka di print.

```

void printLoad(uint8_t *buf, uint8_t len)
{
    if(len == 0){
        Serial.println(F("Load Successfully.));
        return;
    }
    else{
        Serial.print(F("Load success: "));
        Serial.println(buf[0], DEC);
    }
    for(int i=0; i<len-1; i += 2){
        Serial.print(F("Record "));
        Serial.print(buf[i+1], DEC);
        Serial.print(F("\t"));
        switch(buf[i+2]){
            case 0:
                Serial.println(F("Loaded"));
                break;

```

Gambar III.14 Program *Load*

Pada Gambar III.14 merupakan program *load* atau proses menyimpan perintah suara yang telah di *record*. If-else merupakan sintak kondisi untuk menyimpan data, int i = 0 merupakan sebuah data yang diinputkan pada sigtrain i<len-1; i += 2) maka untuk perintah yang disimpan berikutnya (buf [i+1])

```
const char* ssid = "AA_Daus.net";
const char* password = "netdausaa";

IPAddress host(192,168,1,80);
```

Gambar III.15 Konfigurasi SSID dan *Password* jaringan

Gambar III.15 menunjukkan program untuk konfigurasi SSID dan Password yang disesuaikan dengan konfigurasi *access point* dan konfigurasi *IPAddress* sebagai host yang akan digunakan dengan catatan IP tersebut satu jaringan.

```
// Connect to WiFi network -----
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
// Mengatur WiFi -----
// WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// Print status Connect -----
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
```

Gambar III.16 Koneksi Jaringan

Gambar III.16 menunjukkan program koneksi jaringan WiFi pada perangkat *hardware* ke perangkat server. `WiFi.begin (ssid,password)` perintah koneksi disambungkan dengan ssid dan password yang telah disesuaikan pada sintak konfigurasi SSID dan Password. Apabila WiFi dengan status *connected* maka print sebuah status dengan delay 0,5 detik keterangan jaringan terkoneksi dengan *IPAddress* yang digunakan.

```

void loop()
{
    const char* url;

    int ret;
    ret = myVR.recognize(buf, 50);
    if(ret>0) {
        switch(buf[1]) {
            case KamarON:
                // turn ON Kamar
                url="KamarON";
                break;

            case PompaON:
                // turn ON Pompa
                url="PompaON";
                break;

            case KipasON:
                // turn ON Kipas
                url="KipasON";
                break;

            case KamarOFF:
                // turn OFF Kamar
                url="KamarOFF";
                break;

            case PompaOFF:
                // turn OFF Pompa
                url="PompaOFF";
                break;

            case KipasOFF:
                // turn OFF Kipas
                url="KipasOFF";
                break;

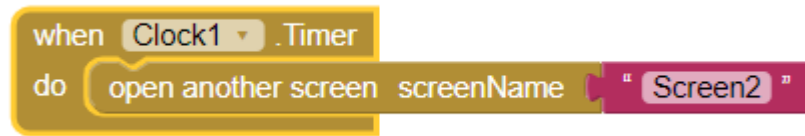
            break;
        default:
            Serial.println("Record function undefined");
            break;
        }
    }
}

```

Gambar III.17 Program Perintah Suara

Pada Gambar III.17 Program perintah suara yang akan diberikan sesuai dengan perintah suara yang telah disimpan pada myVR.recognize perintah yang dilakukan menggunakan switch case untuk membandingkan suatu variable dengan beberapa konstanta, konstanta diawali dengan “case” dan setiap blok case diakhiri dengan “break”. case KamarON merupakan sintak perintah untuk menyalakan lampu kamar dengan get url “KamarON” begitupun dengan perintah fitur selanjutnya dan apabila pengucapan perintah suara tidak sesuai dengan kata yang sudah di *trained* maka akan tampil keterangan *undefined* kata perintah yang diucapkan tidak terdefiniskan.

III.3.2.3 Realisasi Program *Software Application*.



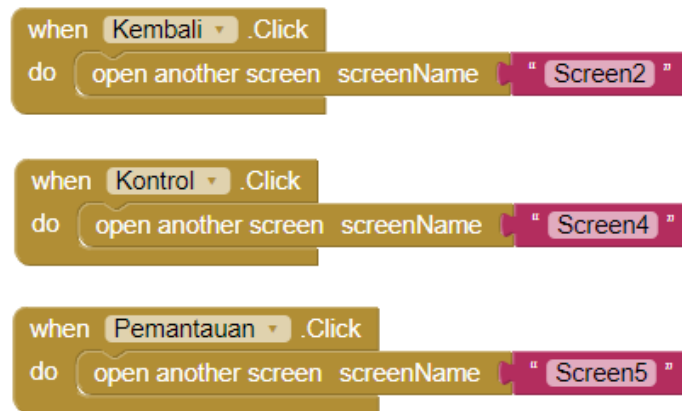
Gambar III.18 Program Blok *Splash Screen*

Realisasi sistem software dirancang pada MIT App Inventor dapat dilihat pada gambar III.18 merupakan program blok *splash screen* yang telah diatur dengan interval waktu 3 detik, ketika aplikasi dibuka program *clock* ini akan berjalan otomatis sesuai interval waktu yang ditentukan kemudian meminta untuk membuka *Screen2* merupakan halaman *login*.



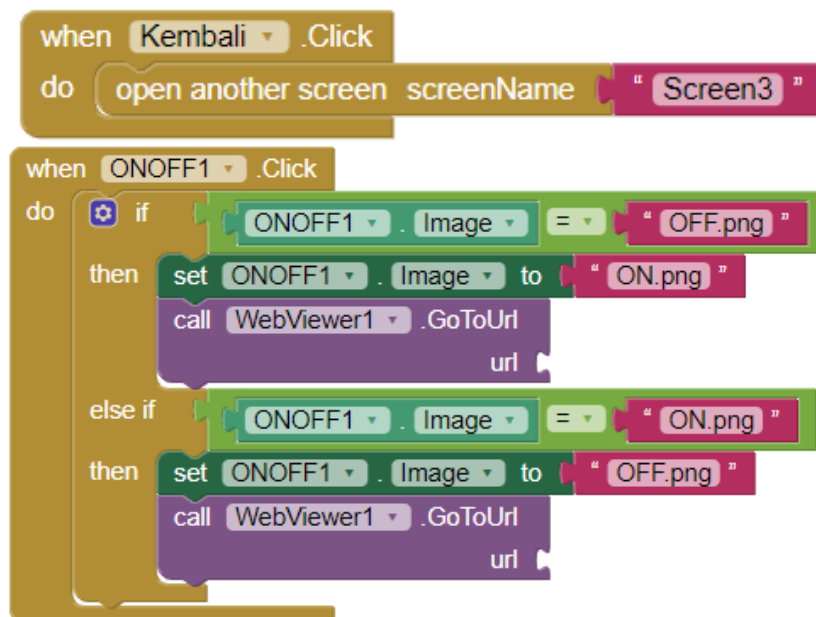
Gambar III.19 Program Blok halaman *Login*

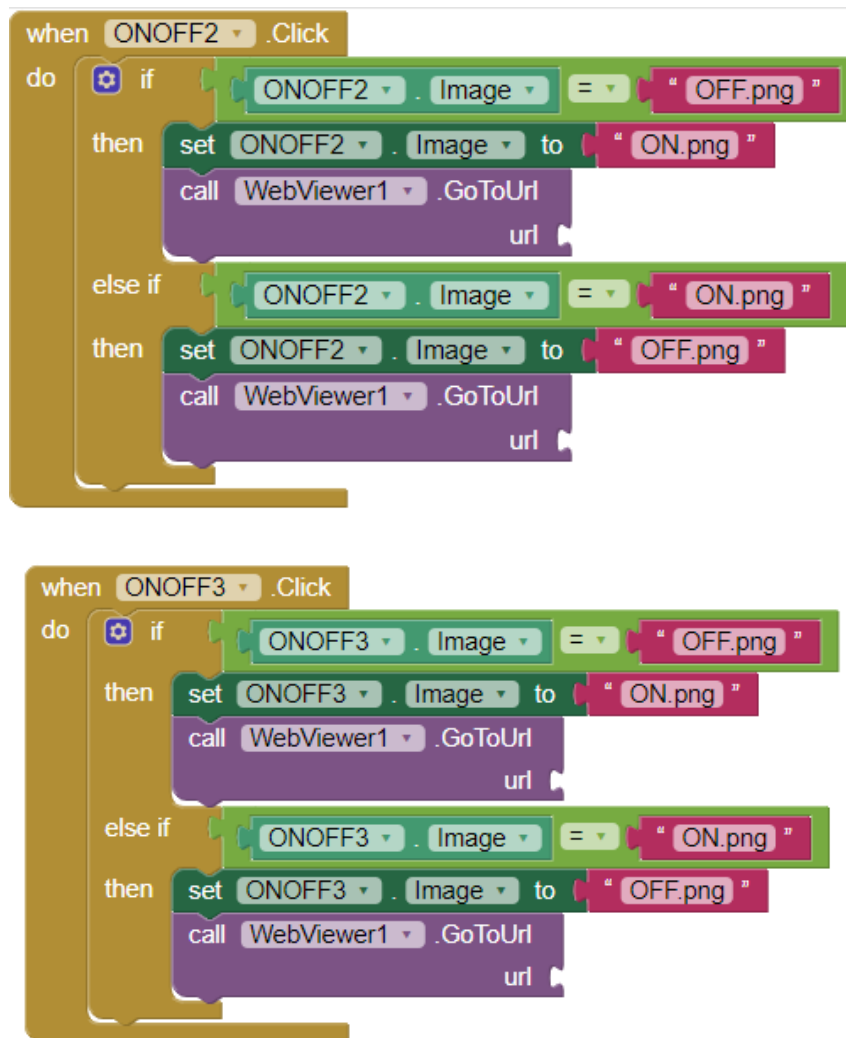
Pada Gambar III.19 dapat dilihat ketika tombol masuk diklik jika kondisi `TextBox = Text = Sholehsoleh23@gmail.com` dan `PasswordTextBox = Text = 23111998` maka akan masuk pada halaman *Screen3* halaman main menu dan apabila kondisinya tidak sesuai maka akan memanggil *Notifier* dengan catatan "Username atau password salah".



Gambar III.20 Program Blok *Button* Kembali, Kontro dan Pemantauan

Pada Gambar III.20 merupakan beberapa blok program tombol, untuk tombol kembali ketika diklik maka akan masuk pada Screen2 merupakan halaman login. Kemudian ketika klik tombol kontrol maka halaman akan dilanjutkan ke Screen4 merupakan halaman fitur kontrol dan ketika klik tombol pemantauan maka akses akan diteruskan pada Screen5 sebagai halaman pemantau fitur.





Gambar III.21 Program Blok *Button* Kembali dan Switch On/Off Saklar

Pada Gambar III.21 terdapat program blok tombol kembali apabila diklik maka akan masuk pada halaman sebelumnya Screen3 kemudian terdapat 3 blok program switch on/off tombol saklar apabila diklik akan menggulir warna tombol sesuai berdasarkan gambar yang ditentukan, apabila digulir pada on/off maka akan memanggil url pada penyedia basis data untuk request menyalakan atau mematikan fitur. ONOFF1= lampu kamar, ONOFF2 = pompa air dan ONOFF3 = kipas angin.