

BAB 4 HASIL DAN PEMBAHASAN

4.1. Persiapan Kebutuhan

Pada bab ini akan dijelaskan beberapa tahapan bagaimana membangun dan mengimplementasikan rancangan tersebut. Alat dan aplikasi ini memiliki beberapa kebutuhan dalam menunjang keberhasilan pembuatan, berikut alat dan perangkat yang dibutuhkan:

- Smartphone Android
- ESP32
- Board LM2596
- Board ULN2003
- Motor Stepper 28BYJ-48
- Pipa tiang penyangga antena
- Wheel motor stepper
- Antena TV

4.2. Code pada Arduino

Code dibawah ini di compile pada aplikasi Arduino IDE, dan ESP32 akan membuka koneksi wifi dengan nama "TestAP" dan passwordnya "test1234".

Pada aplikasi Android nantinya akan memanggil ke alamat IP Address host yang default sudah di konfigurasi di kode tersebut, dan memanggil path R (untuk tombol kanan) atau (untuk tombol kiri).

```
#include <Stepper.h>
const int stepsPerRevolution = 200;
// Konfigurasi pin motor, melalui ULN2003
Stepper myStepper(stepsPerRevolution, 21, 18, 19,
5);

#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>

#define LED_BUILTIN 2 // LED internal ESP32

const char *ssid = "testAP"; //setel nama wifi
```

```

const char *password = "test1234"; //setel password wifi

WiFiServer server(80); //inisialisasi port 80

void setup() {
  // kecepatan motor diatur 90 rpm:
  myStepper.setSpeed(90);
  // initialize the serial port:
  Serial.begin(9600);

  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(115200);
  Serial.println();
  Serial.println("Configuring access point...");

  // konfigurasi nama dan password
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP(); /* Konfigurasi IP
Address host Default */
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.begin();

  Serial.println("Server started");
}

void loop() {
  // memperhatikan clients masuk
  WiFiClient client = server.available();

  if (client) { // jika mendapatkan client,
    Serial.println("New Client."); /* cetak pesan keluar
dari port serial */
    String currentLine = ""; /* membuat String untuk
menyimpan data yang masuk dari klien*/
    while (client.connected()) { /* ulang selama klien
terhubung */
      if (client.available()) { /* jika ada byte untuk dibaca
dari klien */
        char c = client.read(); // baca byte

```

```

Serial.write(c); // tampilkan di serial monitor
if (c == '\n') { // jika byte adalah karakter baris baru

    /* jika baris saat ini kosong, maka mendapat dua karakter baris
    baru berturut-turut, itu adalah akhir dari permintaan HTTP klien, jadi
    kirim respon */
    if (currentLine.length() == 0) {
        /* HTTP Headers selalu dimulai dengan kode respons (mis.
        HTTP / 1.1 200 OK)
        sehingga klien tahu apa yang akan terjadi */
        client.println("HTTP/1.1 200 OK");
        // Tipe Konten
        client.println("Content-
type:text/html");
        client.println();

        // HTTP response ends with another blank line:
        client.println();
        // berhenti dari perulangan:
        break;
    } else { // if you got a newline, then clear currentLine:
        currentLine = "";
    }
    } else if (c != '\r') { /* if you got anything else but
    a carriage return character */
        currentLine += c; // add it to the end of the currentLine
    }

    /* Cek untuk melihat apakah permintaan klien adalah "GET /H"
    atau "GET /L": */
    if (currentLine.endsWith("GET /R")) {
        // GET /H menyalakan LED
        digitalWrite(LED_BUILTIN, HIGH);
        myStepper.step(+64);
        // GET /L turns the LED off
        digitalWrite(LED_BUILTIN, LOW);
    }
    if (currentLine.endsWith("GET /L")) {
        // GET /H turns the LED on
        digitalWrite(LED_BUILTIN, HIGH);
        myStepper.step(-64);
    }
}
}

```

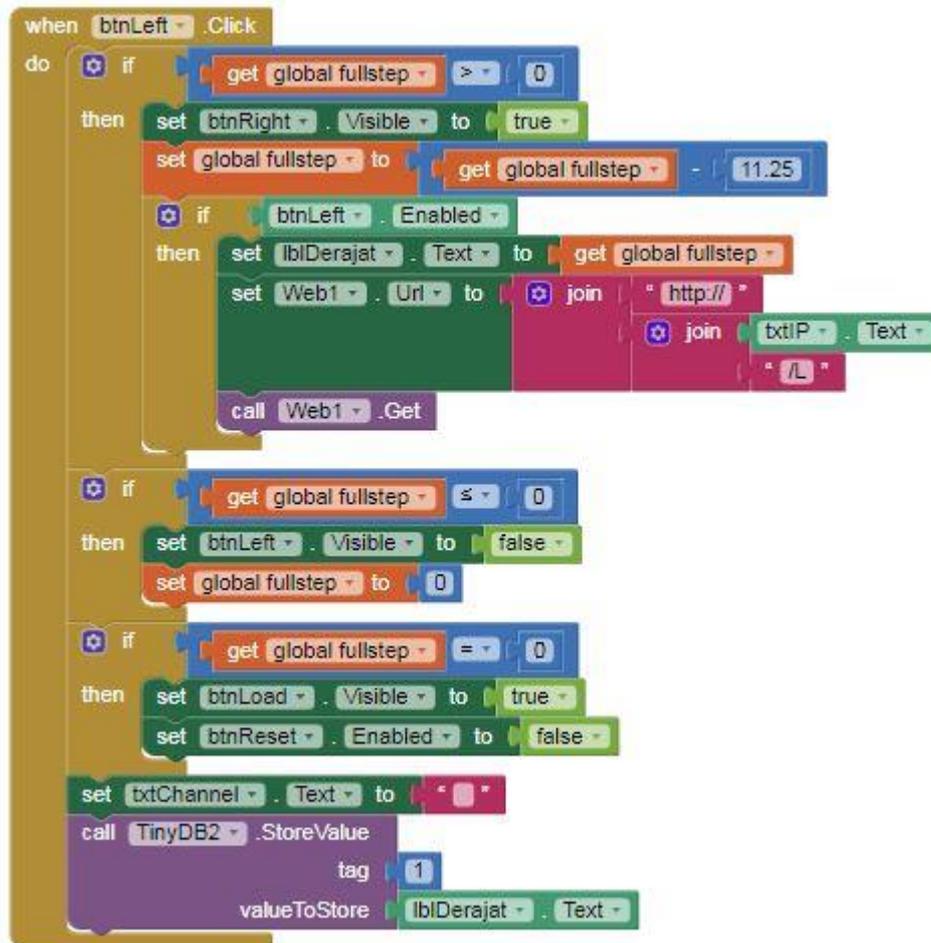
```
        // GET /L turns the LED off
        digitalWrite(LED_BUILTIN, LOW);
    }
}

// tutup koneksi:
client.stop();
Serial.println("Client Disconnected.");
}
}
```

Pada fungsi `myStepper.step(+64)` diinisialisasi nilai 64 yang artinya akan berputar 11,25 per derajat ke kanan. Pada uji coba yang sudah dilakukan jika fungsi `myStepper.step()` diinisialisasi nilai kurang dari 64 maka data request dari aplikasi android ke motor stepper akan mengalami delay putaran.

4.3. Code pada MIT App Inventor

Code pada MIT App Inventor berbentuk susunan puzzle, berikut ini potongan gambarnya.

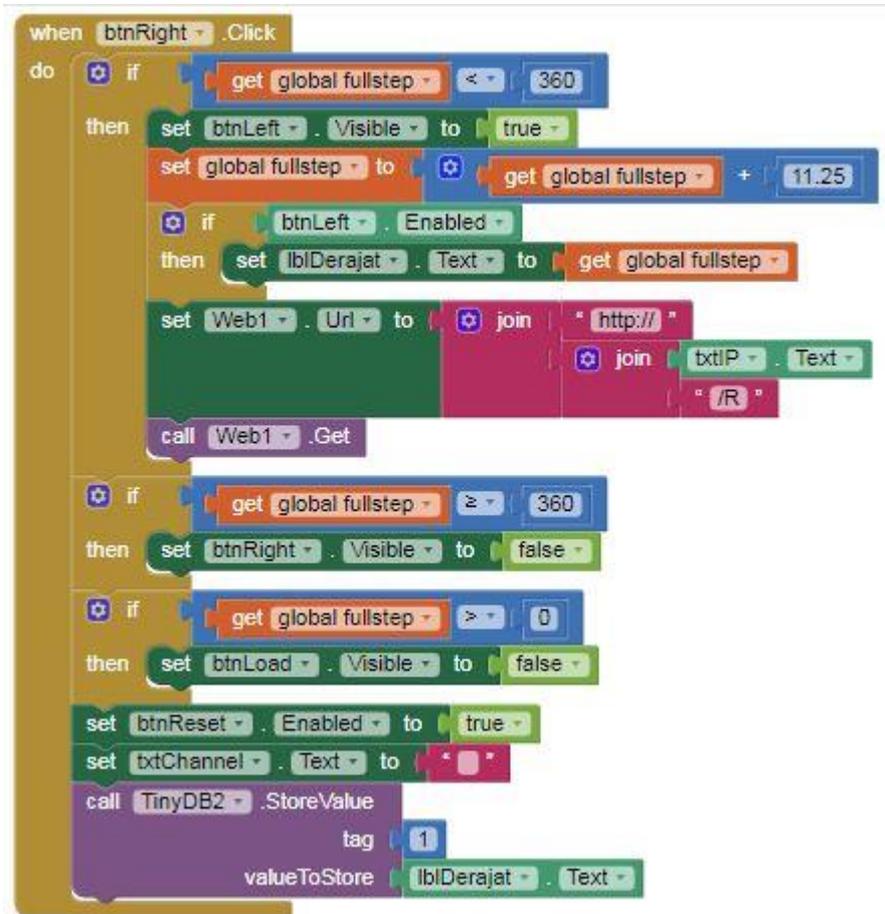


Gambar 4.1 Tombol kiri

Pada gambar 4.1 tersebut melakukan fungsi tombol kiri, yang nantinya jika ditekan maka akan melakukan pengecekan variabel globalnya lebih dari nol atau tidak, variabel nol adalah putaran derajatnya. Jika lebih dari nol maka variabel global dikurangi dengan nilai 11.25 yang nantinya nilai tersebut akan ditampilkan di label informasi derajat putaran. Pada variabel global diinisialisasi nol pada saat aplikasi diinstal. Lalu selanjutnya akan request ke alamat host ESP32 dengan ketentuan Smartphone harus tersambung pada alamat wifi yang sudah dikonfigurasi

dan memanggil fungsi HTTP GET dan ditambahkan dengan URL path "L" yang sudah dikonfigurasi pada kode arduino ESP32.

Setelah itu akan ada kondisi jika kurang sama dengan dari nol maka tombol button tersebut akan disembunyikan atau hidden.

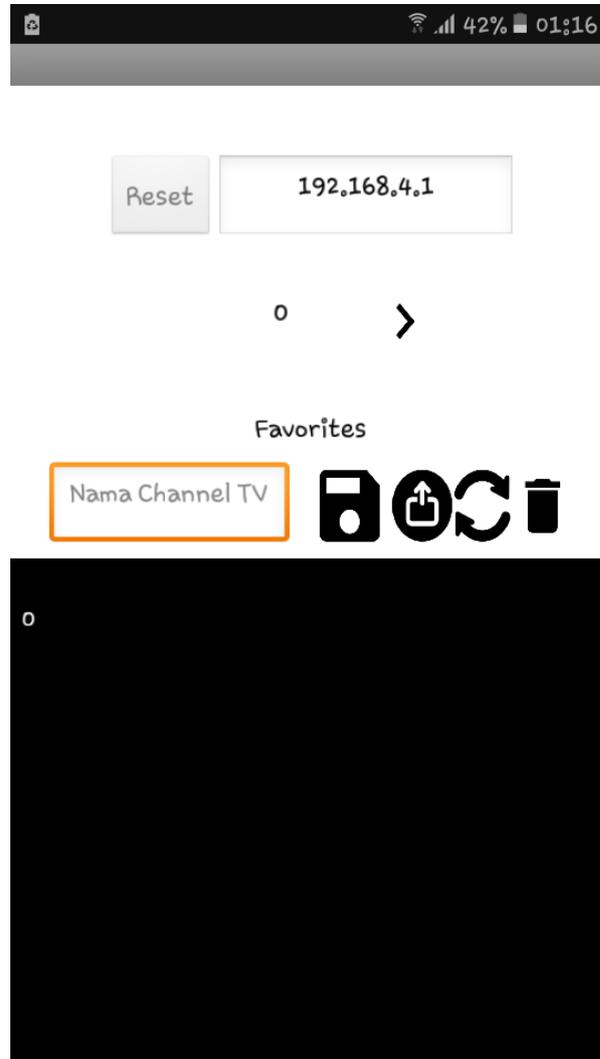


Gambar 4.2 Tombol Kanan

Pada gambar 4.2 fungsi sama saja dengan gambar gambar 4.1 yang membedakan kebalikannya, nilainya batasnya 360 adalah maksimal derajatnya dan URL pathnya.

4.4. Hasil pengujian aplikasi

Pada pengujian aplikasi ini awal aplikasi dijalankan 0 derajat, dan pada daftar list akan menampilkan lagi 0 derajat tetapi itu nantinya sebagai data putaran terakhir saat mau membuka aplikasi.



Gambar 4.3 Tampilan aplikasi saat pertama

Pada daftar list yang ke 2 sampai terakhir menampilkan informasi putaran derajat yang akan disimpan dan mengharuskan mengisi kolom channel TV yang diinginkan.



Gambar 4.4 Tampilan aplikasi tombol save

Pada fitur load data, aplikasi harus menekan tombol reset dahulu jadi anetena akan kembali ke posisi awal putaran ke nol derajat, setelah itu tombol load data akan muncul seperti pada gambar 4.5. User dapat memilih chanel TV yang sudah disave untuk di load datanya.



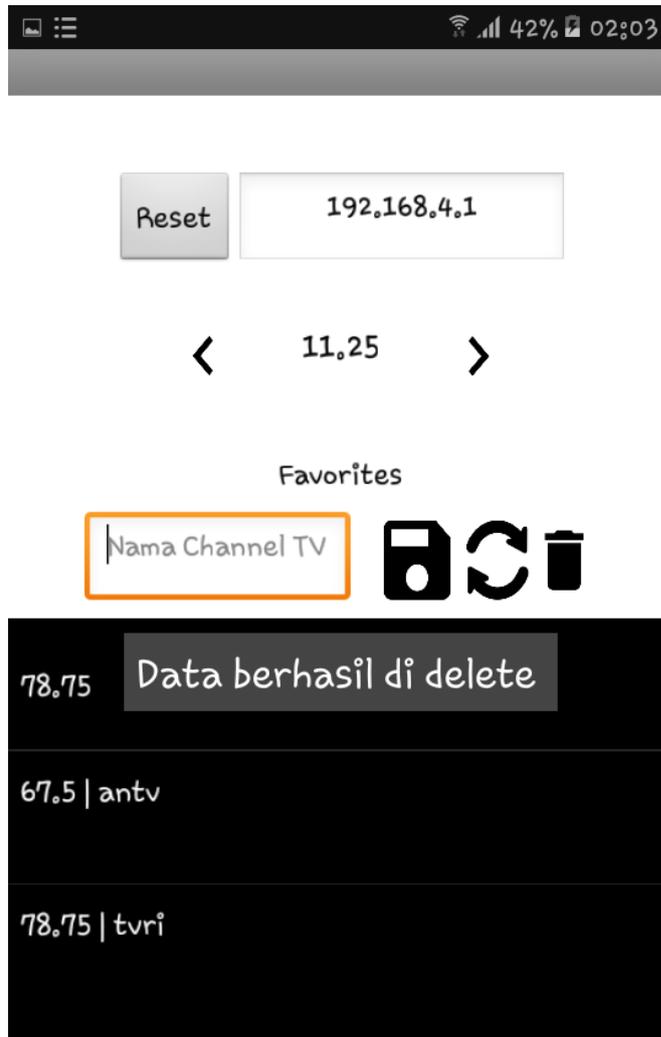
Gambar 4.5 Tampilan aplikasi sesudah tombol reset

Sesudah menekan tombol load, maka tombol load data akan terhidden otomatis jadi jika ingin meload data user harus menekan tombol reset dahulu dan memilih channel TV yang sudah disimpan, seperti pada gambar 4.6.



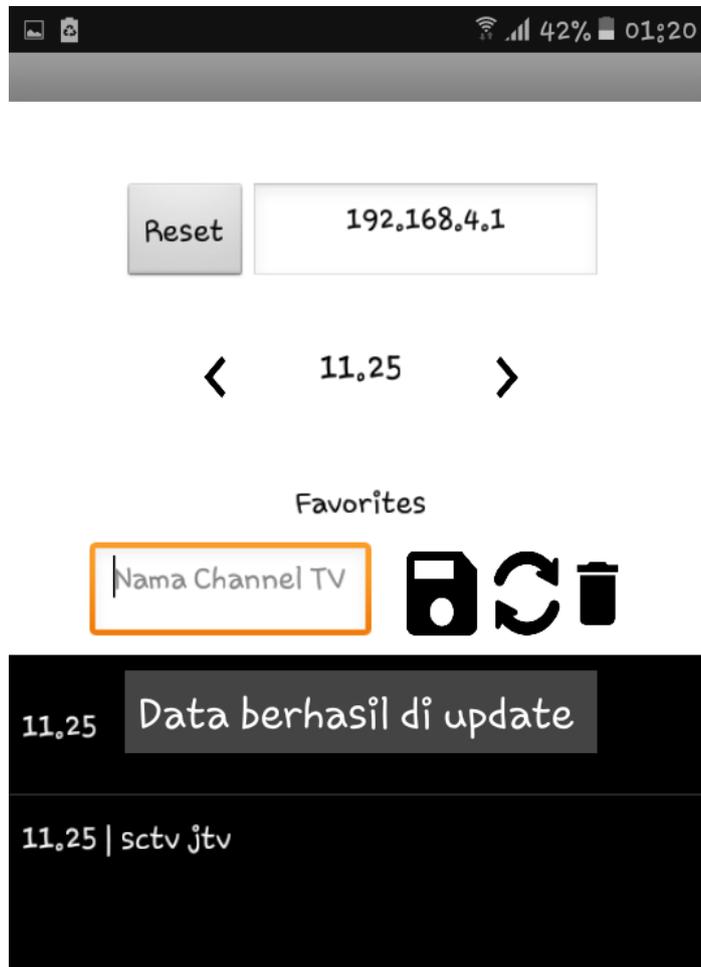
Gambar 4.6 Tampilan aplikasi sesudah tombol load

Pada Gambar 4.7 proses mendelete channel yang diinginkan, dengan memilih channel tv yang pilih pada tampilan list lalu tinggal tekan tombol delete / tombol trash.



Gambar 4.7 Tampilan aplikasi sesudah tombol hapus

Pada proses update akan menampilkan seperti pada gambar 4.8 dan melakukan proses mengubah atau menambahkan nama channel tv yang diinginkan, dengan memilih data channel tv tersebut lalu ubah pada kolom text nama channel TV.



Gambar 4.8 Tampilan aplikasi sesudah tombol update

4.5. Pengujian alat dan aplikasi

Untuk sambungan pada antena TV ke motor stepper ini menggunakan alat tambahan seperti gambar 4.9.



Gambar 4.9 Wheel motor stepper

Wheel motor stepper dimasukkan kedalam sambungan antenanya dan dibaut seperti pada gambar 4.10.



Gambar 4.10 Sambungan ke antena TV

Pada percobaan berikut kabel yang digunakan untuk memutar motor stepper sepanjang 3 meter dari microcontroller-nya. Sepanjang kabel tersebut data yang dikirimkan ke motor stepper tidak hilang.



Gambar 4.11 Pengujian alat dengan antena TV