**PROYEK EMBEDDED SYSTEM**

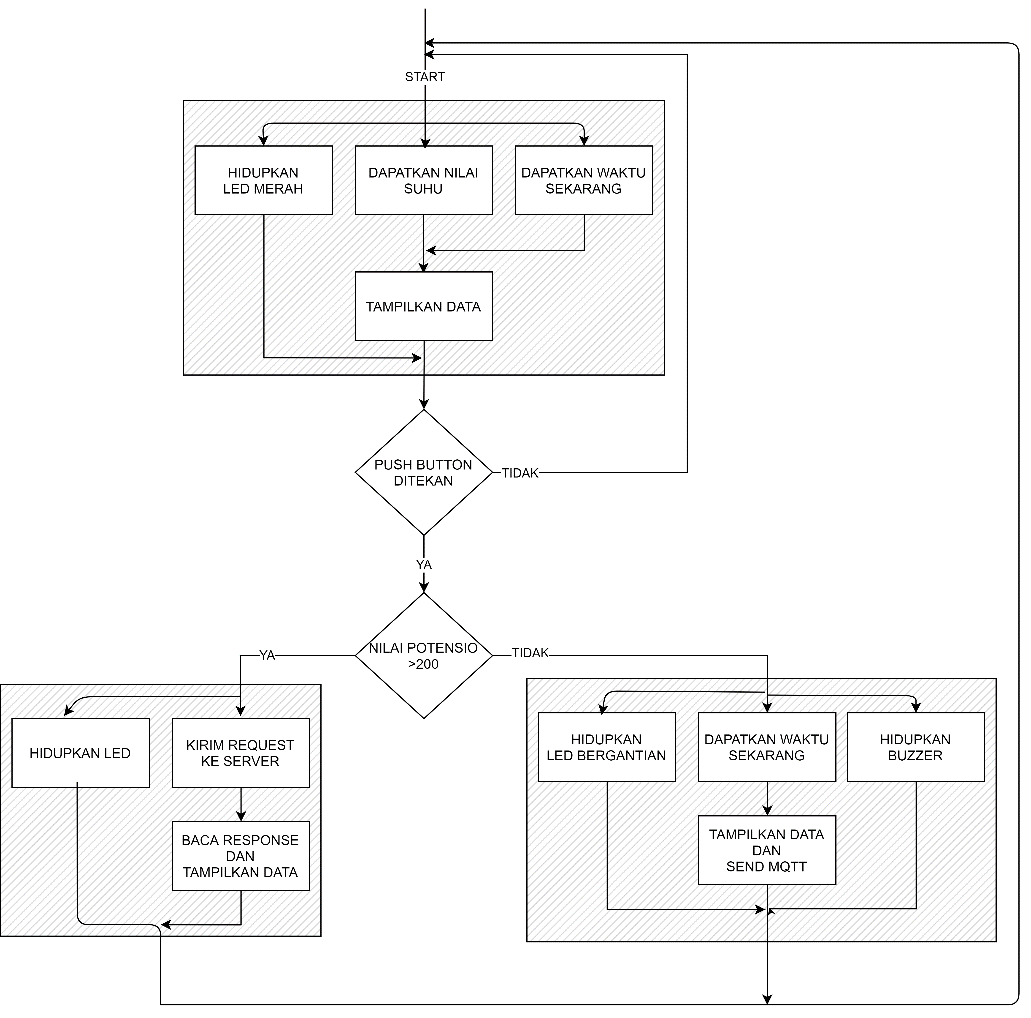
Cosmas Yustianus Billiarta Gondokusuma – 217116585

Proyek ini dibuat untuk memenuhi kelulusan dari mata kuliah *Embedded System*, dengan membuat sebuah sistem kecil yang memanfaatkan penggunaan ESP8266 NodeMCU. Pada sistem ini digunakan NodeJS sebagai *web service*, MQTT Broker yang disediakan oleh Eclipse (<https://mosquitto.org/download/>), MySQL yang disediakan oleh XAMPP secara lokal, dan Arduino IDE untuk melakukan pemrograman pada NodeMCU. Semua terhubung menggunakan WIFI yang tersambung dengan jaringan lokal.

Pada proyek ini terdapat input sebagai berikut. Input digital yang digunakan adalah sensor suhu dan kelembaban DHT11, dan Input analog yang digunakan adalah potensiometer dan *pushbutton*. Sedangkan output dihasilkan oleh 2(dua) buah LED berwarna biru dan merah, LCD, dan buzzer untuk menghasilkan suara.

**Desain Sistem**

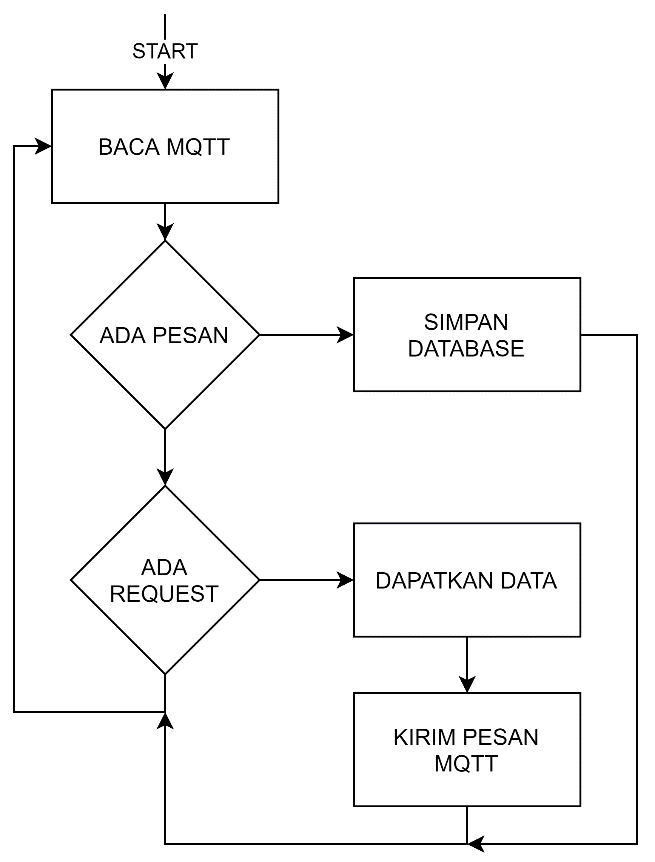
Pada proyek ini, terdapat 2(dua) sistem yaitu sistem NodeMCU dan sistem web service NodeJS. Sistem NodeMCU melakukan input dan mengeluarkan output, dan sistem NodeJS melakukan pembacaan dan penulisan database hasil input dari NodeMCU.



**Gambar 1 Alur Sistem NodeMCU**

Pada Gambar 1, ditampilkan alur sistem dari NodeMCU. Sistem ini mendapatkan input berupa pushbutton, sensor suhu dan kelembaban DHT11, dan potensiometer, dan output yang digunakan adalah dua buah LED, buzzer, dan LCD. Ketika NodeMCU dihidupkan, sistem akan menampilkan fase pertama, yaitu menampilkan waktu beserta tanggal dan suhu yang didapatkan dari DHT11 dan menyalakan LED merah. Dilakukan pengecekan terhadap pushbutton, jika ditekan, maka akan dilakukan pengecekan juga terhadap nilai yang dikembalikan oleh potensiometer.

Jika nilai potensiometer tersebut lebih dari 200, maka dilakukan GET request kepada server NodeJS yang terhubung pada jaringan lokal yang kemudian akan mengembalikan respons yang dibaca oleh NodeMCU yang ditampilkan pada LCD beserta menyalakan LED. Jika nilai potensiometer kurang dari 200, maka didapatkan waktu saat ini dan dikirimkan pada server MQTT yang akan dibaca oleh server NodeJS untuk dimasukkan kedalam database dan menyalakan LED secara bergantian dan membunyikan buzzer untuk mendapatkan perhatian.

****

**Gambar 2 Alur Sistem NodeJS**

Pada Gambar 2, dijelaskan secara garis besar bagaimana cara kerja dari server web service yang menggunakan NodeJS. Terdapat beberapa *package* yang digunakan yaitu: express, mqtt, dan mysql, yang dapat di-*download* melalui npm. MQTT yang dipakai pada package tersebut terdapat sebuah routine yang melakukan pengecekan pada topik yang telah di-*subscribe* pada awal konfigurasi aplikasi, maka ketika NodeMCU mengirimkan pada pesan kepada topik yang sudah di-subscribe oleh NodeJS (dalam kasus ini adalah “proyek/kirim”) NodeJS akan mendapatkan pesan tersebut dan menyimpannya pada database MySQL. Ketika NodeMCU melakukan request GET pada server melalui <http://localhost:3000/getEntry>, server NodeJS akan mendapatkan data pada database MySQL dan mengirimkannya melalui pesan pada topik MQTT yang sudah di-subscribe oleh NodeMCU (dalam kasus ini adalah “proyek/terima”) yang kemudian akan ditampilkan oleh NodeMCU. Juga terdapat *endpoint* yang digunakan untuk melihat data pada database dalam bentuk tabel yang di-*update* setiap terdapat data baru yang memanfaatkan AJAX.

**Gambar rangkaian**

****

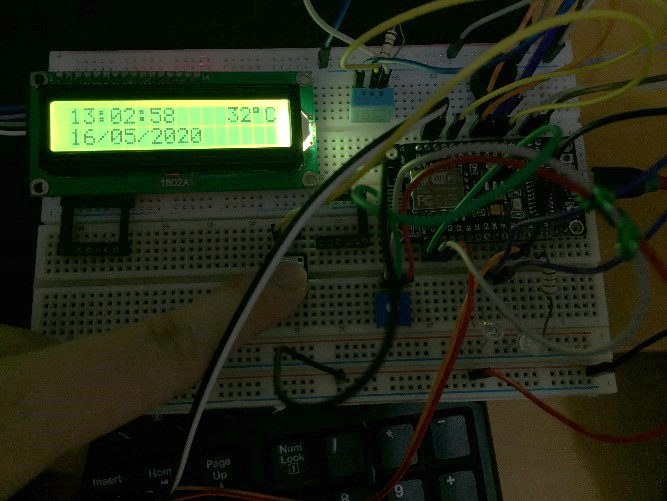
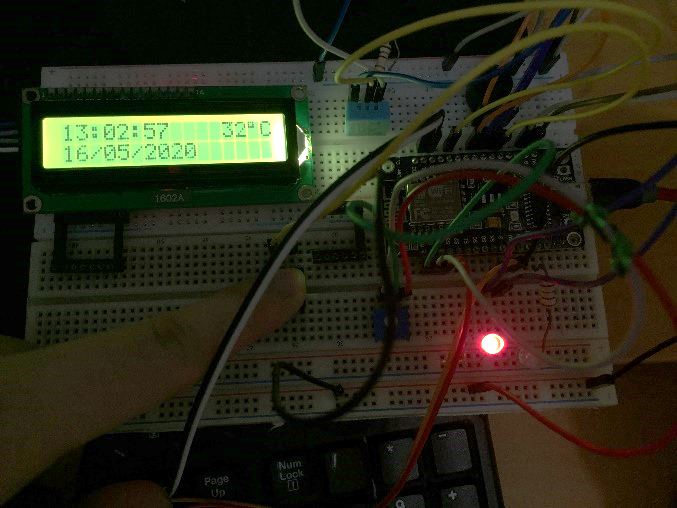
**Gambar 3 Illustrasi Rangkaian NodeMCU**

Rangkaian ini dibentuk menggunakan komponen-komponen yang telah disebut di awal, yaitu: NodeMCU, LED merah dan biru, potensiometer, pushbutton, buzzer, DHT11, dan LCD. Untuk mempermudah pembacaan rangkaian, dapat dilihat Tabel 1 di bawah ini.

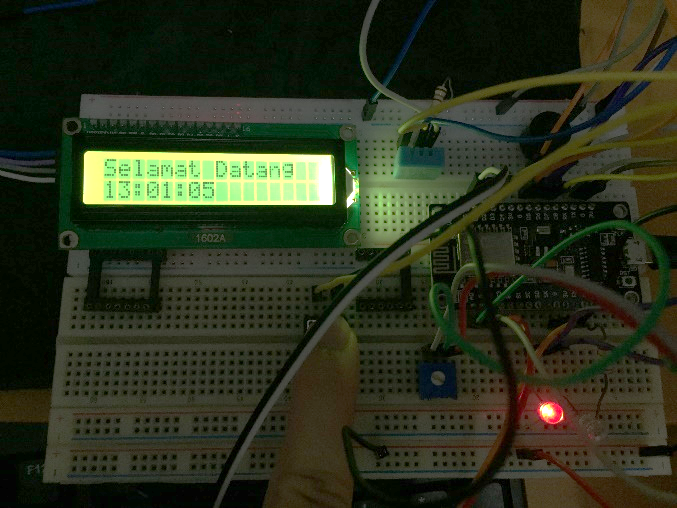
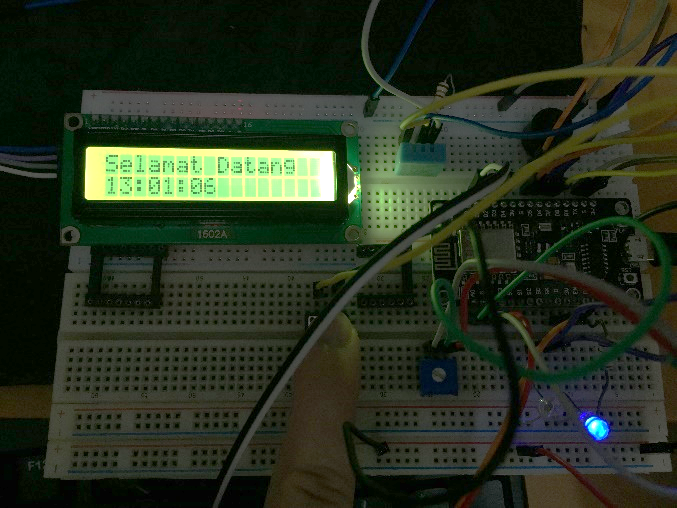
|  |  |  |  |
| --- | --- | --- | --- |
| Nomor | Komponen | Pin | Keterangan |
| 1 | Buzzer | D5 |  |
| 2 | DHT11 | D4 | Diberi resistor pada data. |
| 3 | LED Merah | D6 | Diberi resistor. |
| LED Biru | D7 |
| 4 | LCD 16x2 | D1, D2, VU | Menggunakan pin SCL dan SDA, dan VU untuk *power in*. |
| 5 | Potensiometer | A0 |  |
| 6 | Pushbutton | D8 |  |

**Tabel 1 Detail Penggunaan Komponen**

**Hasil**

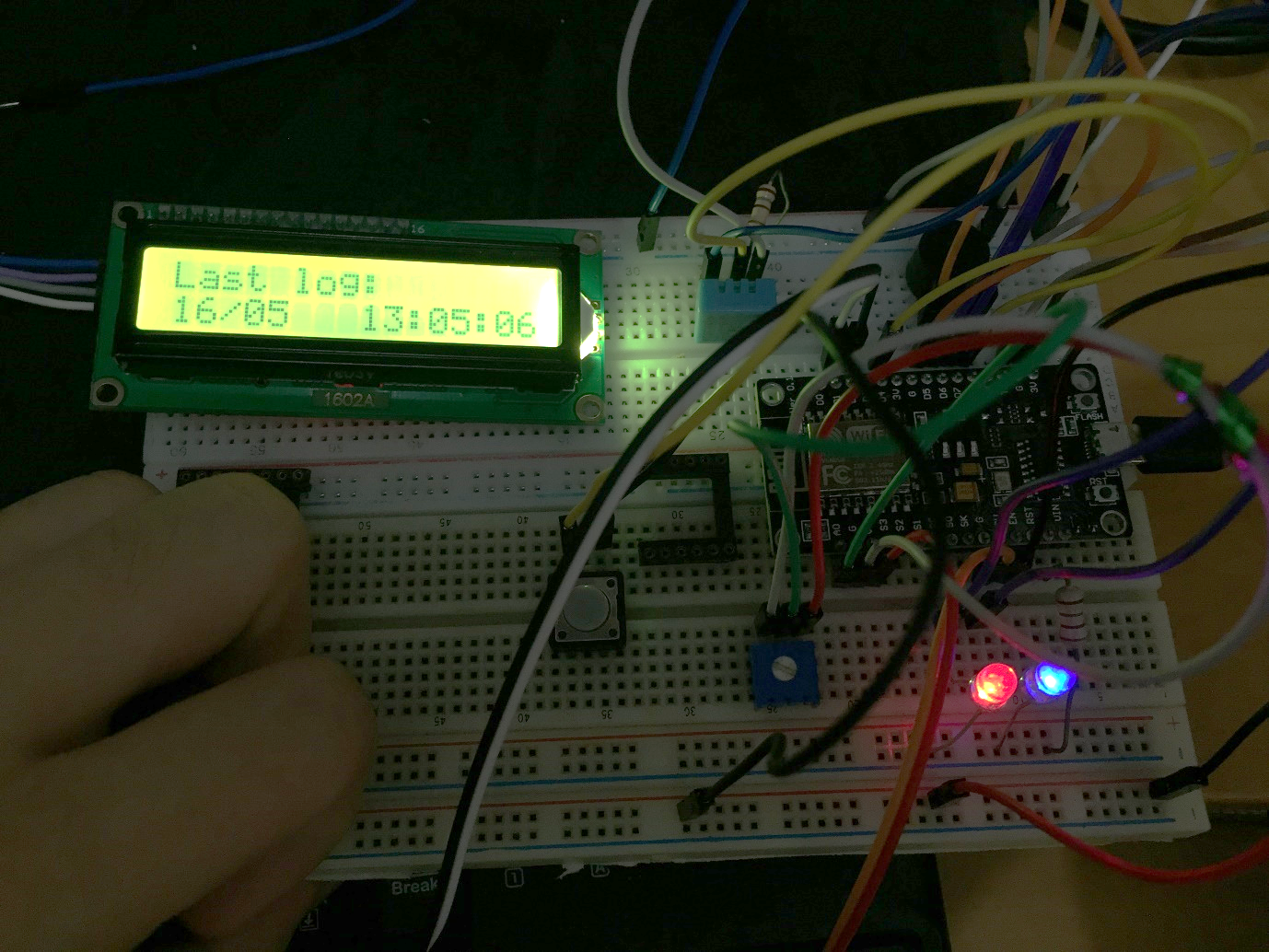
Di bawah ini merupakan hasil dari program yang dibuat pada NodeMCU. Pada saat *idle*, yang terjadi saat NodeMCU baru dinyalakan atau kembali dari kondisi lainnya, LCD(Liquid Crystal Display) menampilkan suhu saat ini yang diukur oleh DHT11 dan waktu beserta tanggal yang didapatkan melalui koneksi server NTP. Selain menampilkan tulisan pada LCD, salah satu LED, yang berwarna merah, dihidup-matikan setiap 200 mili sekon seperti pada Gambar 4a dan Gambar 4b.

**Gambar 4a dan Gambar 4b Kondisi pada saat idle**

Ketika pushbutton ditekan, maka NodeMCU akan mengirimkan pesan kepada server MQTT atau *logging* seperti yang sudah dijelaskan pada bagian desain sistem di atas, maka NodeMCU akan menampilkan informasi waktu yang dicatat pada LCD, membunyikan buzzer, dan menyalakan LED dengan bergantian, seperti pada Gambar 5a dan Gambar 5b.

**Gambar 5a dan Gambar 5b Kondisi pada saat logging**

Ketika pushbutton ditekan dan nilai dari potensiometer lebih dari 200, maka NodeMCU akan mendapatkan *log* terahkir dari database. NodeMCU akan menampilkan log tersebut pada LCD dan menyalakan kedua LED, seperti pada Gambar 6.

****

**Gambar 6 Kondisi ketika mendapatkan data**

**Segmen program**

Pada bagian pemrograman, terdapat dua buah bagian yang telah ditentukan yaitu setup() dan loop(). Dalam segmen program ini, pada bagian loop() dibagi menjadi *fucntion* dan *method* untuk mempermudah *debugging* dan pembacaan kode program. Berdasarkan pada file “.ino“, maka di bawah ini merupakan penjelasan dari segmen-segmen program tersebut.

Pada baris ke-1 hingga baris ke-8, dituliskan library-library yang digunakan untuk menjalankan keseluruhan program seperti: DHT.h untuk menggunakan DHT11, ArduinoMqttClient untuk melakukan komunikasi MQTT, LiquidCrystal\_I2C untuk komunikasi dengan LCD ber-*interface* I2C, dan lain-lainnya.

Pada baris ke-11 hingga baris ke-40, dilakukan inisiasi pada variabel dan *instance* yang akan dipakai pada program. Kemudian pada baris-baris selanjutnya adalah function dan method mulai dari baris ke-43 yang bisa dilihat pada Tabel 2 di bawah ini.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Baris | Nama | Fungsi |
| 1 | 42-44 | poolMQTT | Melakukan *pooling* pada server MQTT. |
| 2 | 45-57 | printLCD | Menampilkan tulisan pada layar berdasarkan input yang diberikan dan menampilkan suhu pada layar. |
| 3 | 58-65 | tambahNol | Menambahkan angka nol “0” pada input yang dikembalikan dalam bentuk String. |
| 4 | 66-68 | baca\_potensio | Mengembalikan nilai potensiometer yang ditangkap secara analog yang sudah di-*remap* dari 0 hingga 255. |
| 5 | 69-70 | Baca\_DHT | Mendapatkan nilai suhu dan kelembaban dari sensor DHT11. |
| 6 | 73-82 | connectKeMQTT | Melakukan koneksi kepada server MQTT yang sudah ditentukan. |
| 7 | 84-87 | getWaktuSekarang | Mengembalikan waktu dan tanggal dalam bentuk teks. |
| 8 | 88-99 | bunyi\_beep | Membunyikan buzzer berdasarkan pengaturan yang ditentukan. |
| 9 | 100-102 | matiin\_beep | Mematikan buzzer yang menyala. |
| 10 | 103-106 | analogLampu | Menyalakan LED berdasarkan nilai input yang diberikan. |
| 11 | 107-156 | nyalain\_lampu | Menyalakan lampu berdasarkan nilai potensiometer dan kondisi sistem (idle, logging, atau ambil data). |
| 12 | 158-165 | nyalaSuara | Menyalakan dan mematikan buzzer sesuai dengan konfigurasi yang dibuat. |
| 13 | 165-171 | kirimMQTT | Mengirimkan pesan kepada topik “proyek/kirim” berisikan tanggal dan waktu |
| 14 | 172-185 | cekMQTT | Mengembalikan nilai respons dari server NodeJS dengan melakukan GET request. |
| 15 | 186-191 | resetLCD | Menghapus semua teks pada LCD menjadi bersih. |
| 16 | 192-218 | serverLCD | Menentukan teks yang ditampilkan pada LCD berdasarkan kondisi saat dijalankan. |
| 17 | 219-241 | gantiMode | Mengganti mode dan melakukan aktivitas berdasarkan kondisi saat dijalankan. |
| 18 | 243-303 | getNtpTime dan sendNTPpacket | Melakukan koneksi terhadap server NTP untuk mendapatkan waktu, seperti mendapatkan waktu menggunakan RTC. |
| 19 | 305-344 | setup | Melakukan setup awal atau inisiasi pada program ketika NodeMCU menyala, seperti: mengatur koneksi pin, menyambungkan pada WIFI dan server MQTT, menyalakan LCD dan mengatur koneksi NTP. |
| 20 | 345-372 | loop | Mengeksekusi method-method yang dalam interval yang ditentukan. |

**Tabel 2 Detail Function dan Method pada Segmen Program**