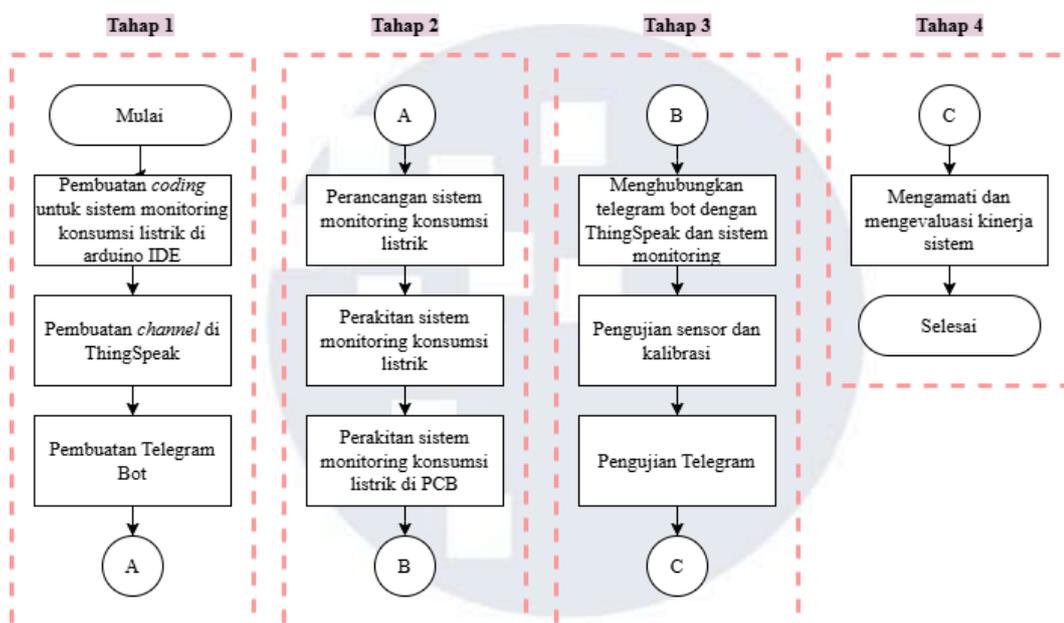


BAB III

METODE PERANCANGAN DAN EKSPERIMEN

3.1 Metode Perancangan

Dalam pelaksanaannya tugas akhir ini dibagi menjadi empat tahapan yang terlihat pada Gambar 3.1 di bawah ini.



Gambar 3.1 Diagram Alir Pembuatan Sistem Akuisisi data Konsumsi Listrik

Berdasarkan diagram alir pembuatan sistem akuisisi data konsumsi listrik, tugas akhir ini terbagi menjadi empat tahap utama yaitu pembuatan sistem perangkat lunak, pembuatan sistem perangkat keras, pengujian, dan pengamatan. Tahap pertama adalah pembuatan sistem perangkat lunak yang terbagi menjadi tiga bagian yaitu pembuatan kode program menggunakan Arduino IDE, pembuatan *channel* untuk visualisasi data di *ThingSpeak*, dan pembuatan Telegram Bot sebagai tempat interaksi antara pengguna dan sistem perangkat keras. Tahap kedua adalah pembuatan sistem perangkat keras sistem akuisisi data konsumsi listrik dimulai dari perancangan hingga perakitan seluruh komponen sistem ke dalam papan sirkuit cetak (PCB). Setelah sistem selesai dibuat, tahap ketiga yaitu pengujian dilakukan

untuk memastikan keakuratan sistem akuisisi data dalam mengukur tegangan, arus, dan faktor daya. Selain menguji keakuratan sistem akuisisi data, Telegram Bot diuji untuk menemukan waktu yang dibutuhkan Telegram Bot memproses perintah yang diberikan pengguna. Terakhir adalah pengamatan yang mengamati kinerja sistem secara keseluruhan dan mengambil kesimpulan mengenai efektivitas sistem dalam melakukan akuisisi data.

3.2 Alat dan Bahan

3.2.1 Perangkat Lunak

Pembuatan sistem perangkat lunak menggunakan aplikasi Arduino IDE, *website ThingSpeak*, Telegram Bot, dan laptop. Arduino IDE digunakan untuk menulis program kode sistem akuisisi data. *ThingSpeak* menyimpan dan memvisualisasikan data dari sistem akuisisi data dalam bentuk grafik. Telegram Bot sebagai *interface* atau antarmuka antara sistem dengan pengguna. Laptop merupakan tempat untuk menjalankan Arduino IDE, *ThingSpeak*, dan Telegram Bot.

3.2.2 Perangkat Keras

Pembuatan sistem perangkat keras menggunakan mikroendali NodeMCU ESP8266, *relay 2 Channel*, sensor PZEM-004T, steker listrik, stop kontak, jumper, kabel, isolasi listrik, solder, timah, *blackbox*, dan PCB. NodeMCU ESP8266 merupakan mikroendali yang terintegrasi langsung dengan WiFi. *Relay 2 Channel* digunakan untuk mengontrol aliran listrik yang masuk ke peralatan elektronik. PZEM-004T merupakan modul sensor yang digunakan untuk mengukur tegangan, arus, dan daya listrik yang dikonsumsi. Steker listrik menghubungkan sistem akuisisi data ke tegangan tinggi untuk menghidupkan peralatan elektronik. Stop kontak menghubungkan sistem akuisisi data dengan peralatan elektronik yang diukur. Jumper menghubungkan antara komponen dan mikroendali. Kabel digunakan untuk menghantarkan aliran listrik. Isolasi listrik digunakan untuk mencegah arus listrik mengalir keluar. Solder digunakan untuk melelehkan timah yang digunakan untuk

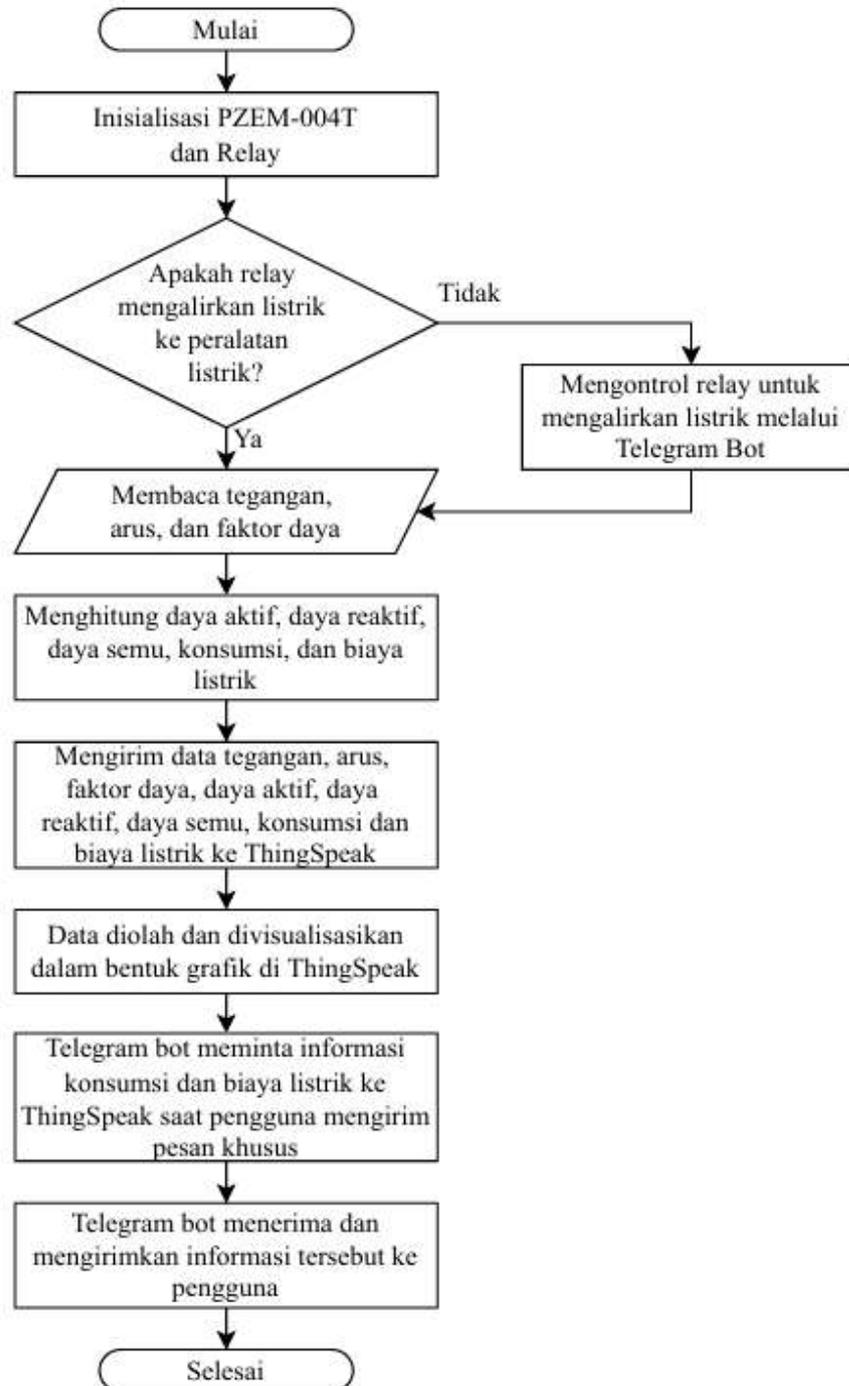
menghubungkan antara komponen. Blackbox digunakan untuk menyimpan sistem akuisisi data. PCB digunakan untuk meletakkan jumper dan kabel.

3.3 Alur Kerja Sistem

Alur kerja sistem akuisisi data memiliki lima tahap seperti terlihat di Gambar 3.2. Lima tahapan utama dalam proses kerja sistem adalah pengontrolan aliran listrik oleh *relay*, pengukuran oleh sensor PZEM-004T, pengolahan data oleh mikrokontroler, visualisasi data melalui *ThingSpeak*, serta interaksi pengguna menggunakan Telegram Bot. Tahap pertama dimulai dari pengontrolan aliran listrik ke peralatan elektronik oleh *relay* yang dikendalikan melalui perintah yang dikirim oleh pengguna di Telegram Bot.

Selanjutnya pada tahap pengukuran, sensor PZEM-004T yang telah diinisialisasi membaca parameter tegangan, arus listrik, dan faktor daya dari peralatan yang sedang beroperasi. Data hasil pengukuran tersebut kemudian masuk ke tahap pengolahan yang dilakukan oleh mikrokontroler NodeMCU ESP8266. Data tegangan, arus, dan faktor daya listrik akan dikelola untuk mendapatkan daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya dari peralatan listrik. Setelah pengolahan data selesai, mikrokontroler akan mengirimkan kedelapan data parameter listrik ke platform *ThingSpeak* melalui koneksi internet. Namun, apabila mikrokontroler tidak terhubung ke jaringan WiFi, maka pengguna perlu terlebih dahulu menyambungkannya ke internet agar proses pengiriman data dapat berjalan.

Tahap berikutnya adalah visualisasi data, *ThingSpeak* akan menerima, menyimpan, serta menyajikan data yang dikirimkan dalam bentuk grafik yang ditampilkan pada dasbor di dalam *channel ThingSpeak*. Visualisasi ini bertujuan untuk memberikan gambaran konsumsi energi secara *real-time* dan historis kepada pengguna. Terakhir, pada tahap interaksi pengguna, sistem memungkinkan pengguna untuk berkomunikasi melalui Telegram Bot dengan mengirimkan perintah tertentu. Perintah ini kemudian akan diproses oleh sistem, dan Telegram Bot akan memberikan respons sesuai data yang diminta.



Gambar 3.2 Diagram Alir Sistem Akuisisi data

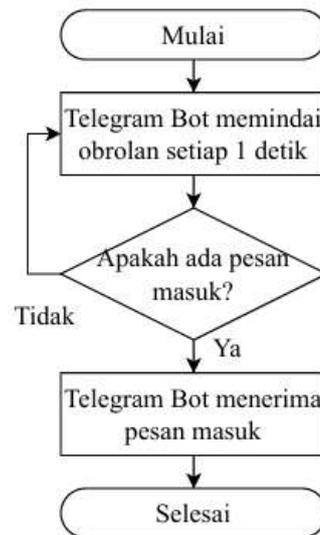
3.4 Pembuatan Perangkat Lunak

3.4.1 Pemrograman

Pemrograman sistem akuisisi data konsumsi listrik dilakukan menggunakan perangkat lunak Arduino IDE. Struktur program dibagi menjadi beberapa bagian kode yaitu sebagai berikut:

- 1) Variabel global berfungsi untuk menyimpan angka maupun karakter di dalam program. Variabel ini dapat dipanggil oleh semua fungsi di dalam program. Tugas akhir ini menggunakan beberapa variabel berikut:
 - a. Pustaka luar
 - b. Nama dan sandi *WiFi*
 - c. Token, chat ID, dan variabel telegram bot
 - d. Pin *relay* dan PZEM-004T yang terhubung dengan mikrokendali
 - e. Kode API dari saluran *ThingSpeak* dan interval pengiriman data ke *ThingSpeak*
- 2) Void `setup ()` adalah fungsi yang berjalan hanya satu kali saat program pertama kali dijalankan. Void ini berfungsi untuk melakukan konfigurasi awal dan menetapkan kondisi awal. Tugas akhir ini menggunakan void `setup` berikut:
 - a. Inisialisasi serial monitor
 - b. Menjadikan *relay* sebagai keluaran dan mengatur *relay* dalam kondisi HIGH atau nonaktif.
 - c. Menghubungkan mikrokendali dengan *WiFi*
 - d. Inisialisasi Telegram bot dan *ThingSpeak*
- 3) Void `pengukuran1 ()` berfungsi untuk mengukur tegangan, arus, faktor daya, daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya peralatan 1 dari pembacaan sistem sensor 1.
- 4) Void `pengukuran2 ()` berfungsi untuk mengukur tegangan, arus, faktor daya, daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya peralatan 2 dari pembacaan sistem sensor 2.

- 5) Void `botUpdates ()` berfungsi untuk menerima pesan masuk dari pengguna. Alur kerja void ini dapat dilihat di Gambar 3.3 di bawah ini.

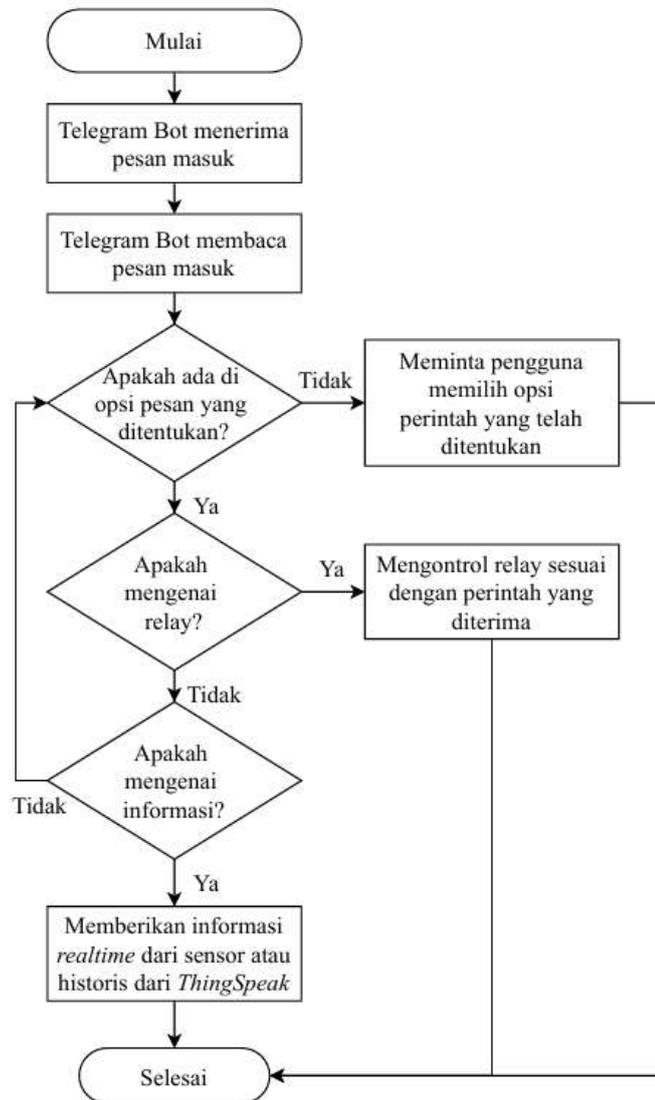


Gambar 3.3 Diagram alir telegram bot menerima pesan masuk

Berdasarkan diagram alir, Telegram Bot memindai obrolan setiap 1 detik untuk melihat apakah ada pesan masuk. Ketika ada pesan masuk, Telegram Bot akan menerima pesan tersebut,

- 6) Void `handleNewMessages ()` berfungsi untuk mengatur respons telegram ketika mendapatkan perintah dari pengguna. Alur kerja void ini dapat dilihat di Gambar 3.4 di bawah ini.



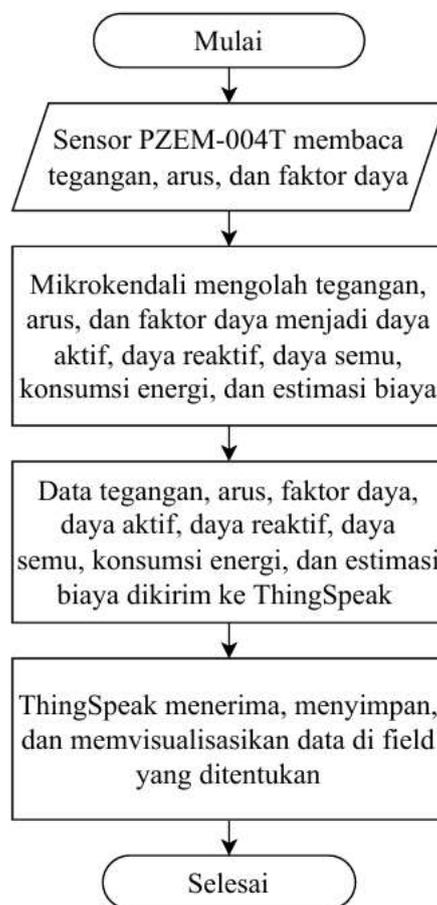


Gambar 3.4 Diagram alir telegram bot merespons pesan masuk

Berdasarkan diagram alir, Telegram Bot menerima dan membaca yang dikirimkan oleh pengguna. Setiap pesan yang masuk akan dianalisis, kemudian dijawab sesuai dengan jenis perintah yang diterima. Telegram Bot akan mengontrol *relay* berdasarkan perintah yang masuk, baik untuk mengaktifkan maupun menonaktifkan aliran listrik ke perangkat elektronik. Jika perintah yang diberikan berkaitan dengan permintaan informasi, Telegram Bot akan mengirimkan data dari sensor untuk permintaan data *real-time*, dan membiarkan mengirimkan tautan URL *ThingSpeak* untuk permintaan data historis. Telegram

Bot dalam sistem ini terbatas pada pengendalian *relay* dan penyampaian informasi konsumsi, sehingga, perintah di luar kedua fungsi tersebut akan di arahkan ke opsi yang ada.

- 7) Void *ThingSpeak* () berfungsi membaca data tegangan, arus, daya, konsumsi energi, dan estimasi biaya dari mikrokendali dan mengirimkannya ke saluran *ThingSpeak*. Alur kerja void ini dapat dilihat di Gambar 3.5 di bawah ini.



Gambar 3.5 Diagram alir *ThingSpeak*

Berdasarkan diagram alir, sensor membaca tegangan dan arus listrik. Data yang diperoleh kemudian dikirim ke mikrokendali untuk diolah menemukan daya listrik, konsumsi energi, serta estimasi biaya penggunaan listrik. Setelah proses pengolahan data selesai, seluruh data akan dikirimkan secara otomatis ke

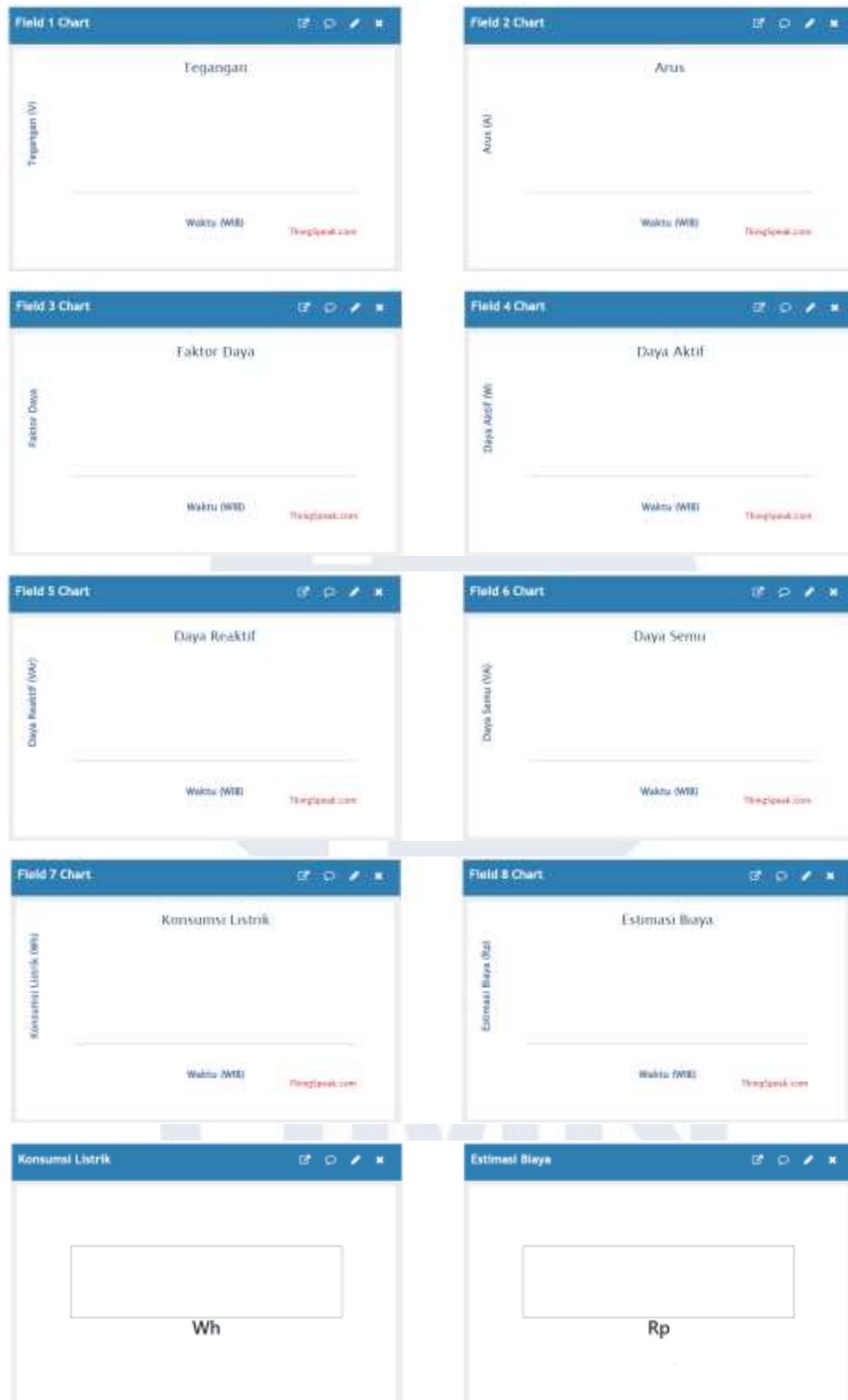
platform *ThingSpeak* melalui jaringan internet. *ThingSpeak* kemudian akan menerima data tersebut, menyimpannya dalam *field* yang telah ditentukan, dan menampilkannya dalam bentuk grafik visual pada dasbor untuk memudahkan proses pemantauan dan analisis.

8) Void Loop berfungsi menjalankan kode program secara berulang selama mikroendali menyala. Tugas akhir ini menggunakan void loop untuk menjalankan pembacaan data oleh sensor, Telegram Bot, dan *ThingSpeak*.

3.4.2 Pembuatan *channel* di *ThingSpeak*

Channel pada *ThingSpeak* digunakan untuk penyimpanan dan visualisasi data dari sistem akuisisi konsumsi listrik. *ThingSpeak* merupakan platform berbasis *cloud* yang mendukung perangkat IoT untuk mengirimkan dan memantau data secara real-time. *ThingSpeak* memberikan dua jenis API Key yaitu *Write API Key* dan *Read API Key* yang akan dimasukkan ke dalam kode program sistem. *Write API Key* digunakan oleh mikroendali untuk mengirimkan data ke *ThingSpeak*, sedangkan *Read API Key* digunakan untuk membaca data yang telah tersimpan.

Langkah pertama dalam pembuatan *channel* di *ThingSpeak* adalah membuat akun di *ThingSpeak*. Setelah memiliki akun dilakukan langkah berikutnya yaitu membuat *channel* baru dan melakukan konfigurasi awal berupa penamaan *channel* dan pembuatan *field* yang berfungsi sebagai tempat penyimpanan dan visualisasi data yang dikirimkan dari perangkat IoT. Terdapat delapan *field* dalam *channel ThingSpeak* yang digunakan untuk menyimpan data tegangan, arus, faktor daya, daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya listrik yang diukur oleh sensor dan dikirim melalui mikroendali NodeMCU ESP8266. Terdapat delapan *field* yang digunakan untuk menyimpan data parameter listrik. Delapan *field* ini akan diisi oleh akuisisi data konsumsi listrik dari sensor seperti terlihat pada Gambar 3.6.

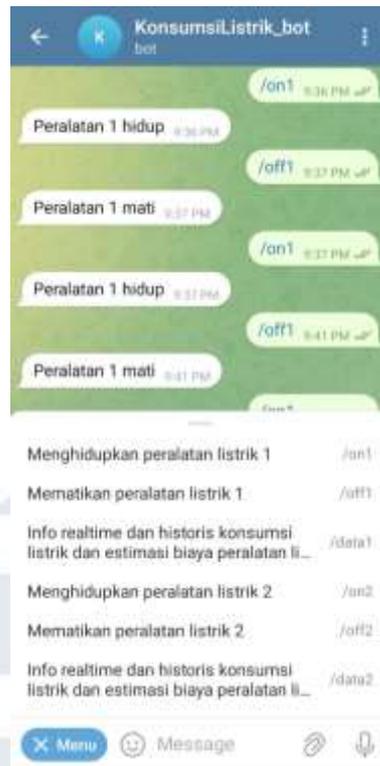


Gambar 3.6 Antarmuka *ThingSpeak*

3.4.3 Pembuatan Telegram Bot

Telegram Bot digunakan sebagai antarmuka komunikasi antara pengguna dan sistem akuisisi data konsumsi listrik. Pengguna dapat mengontrol *relay* secara jarak jauh dan memperoleh informasi konsumsi listrik melalui Telegram Bot dengan memberikan perintah khusus. Pembuatan Telegram Bot dimulai dengan membuat bot baru melalui BotFather di Telegram dan meminta Telegram ID menggunakan IDBot. API bot dari BotFather dan Telegram ID berfungsi untuk menghubungkan program dengan bot yang telah dibuat. Perintah khusus pada Telegram Bot dibuat di BotFather, berikut adalah perintah yang ada di Telegram Bot sistem akuisisi data konsumsi listrik:

- 1) /on1 : menyalakan *relay* peralatan 1 dan menginformasikan peralatan 1 hidup
- 2) /off1 : mematikan *relay* peralatan 1 dan menginformasikan peralatan 1 mati
- 3) /on2 : menyalakan *relay* peralatan 2 dan menginformasikan peralatan 2 hidup
- 4) /off2 : mematikan *relay* peralatan 2 dan menginformasikan peralatan 2 mati
- 5) /data1 : menampilkan data *real-time* dan historis peralatan 1
- 6) /data2 : menampilkan data *real-time* dan historis peralatan 2

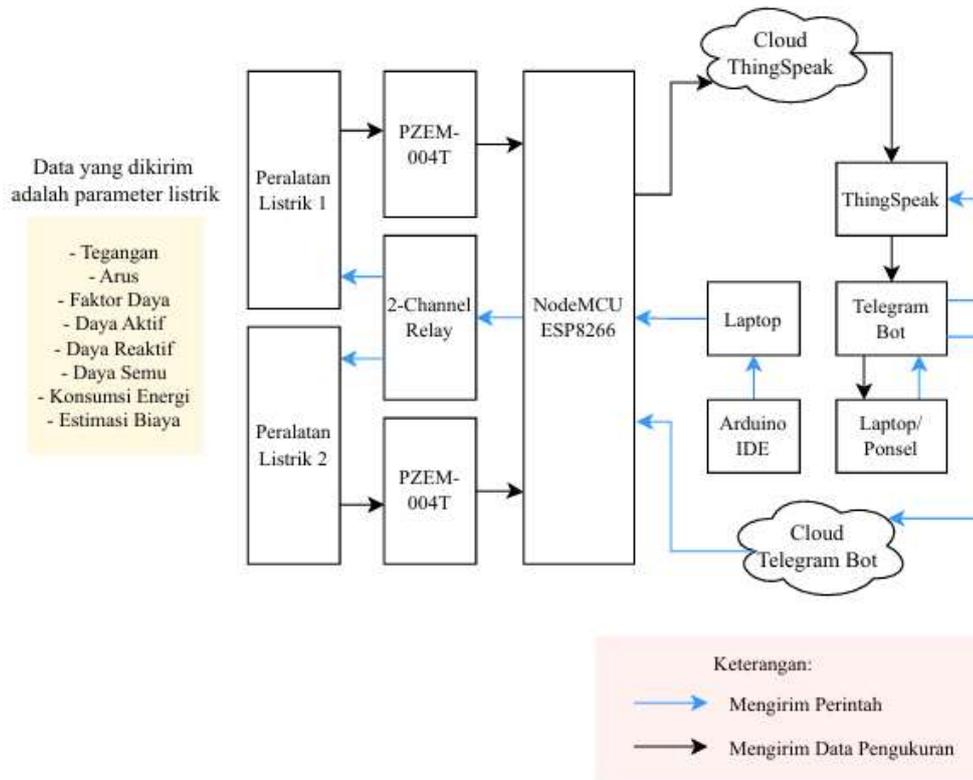


Gambar 3.7 Antarmuka Telegram Bot

3.5 Pembuatan Perangkat Keras

3.5.1 Blok Diagram

Gambar 3.8 merupakan diagram blok sistem akuisisi data konsumsi listrik berbasis IoT. Rangkaian sistem terbagi menjadi dua bagian yaitu perangkat keras sebagai *transmitter* atau TX dan perangkat lunak sebagai *receiver* atau RX. TX bertugas untuk mengirim data pengukuran dan RX bertugas untuk menerima data pengukuran dan mengolahnya untuk ditampilkan kepada pengguna. Perangkat keras terdiri dari mikrokontroler NodeMCU ESP8266, sensor PZEM-004T, dan *relay* 2-Channel. Perangkat lunak terdiri dari *ThingSpeak* yang menyimpan data dan memvisualisasikan grafik data di dasbor, serta Telegram bot yang berinteraksi dengan pengguna.



Gambar 3.8 Rangkaian Sistem Akuisisi data Konsumsi Listrik buat siklus

Sensor PZEM-004T berperan dalam membaca parameter tegangan, arus, dan faktor daya dari peralatan yang terhubung. Data yang diperoleh kemudian diproses oleh mikrokontroler untuk menghitung daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya penggunaan listrik. Setelah proses pengolahan, mikrokontroler akan mengirimkan data tersebut ke platform *ThingSpeak* untuk disimpan dan divisualisasikan dalam bentuk grafik. Sementara itu, Telegram Bot digunakan sebagai antarmuka interaktif antara pengguna dan sistem, yang memungkinkan pengguna untuk memperoleh informasi konsumsi listrik secara *real-time* maupun historis, serta mengendalikan *relay* melalui perintah tertentu. Fungsi *relay* dalam sistem ini adalah untuk mengatur aliran listrik ke peralatan elektronik, sehingga pengguna dapat mengaktifkan atau menonaktifkan peralatan listrik secara jarak jauh.

3.5.2 Rangkaian Sistem

Rangkaian sistem akuisisi data konsumsi listrik terdiri dari mikrokontroler NodeMCU ESP8266, sensor PZEM-004T, dan *relay*. Pin pada mikrokontroler dihubungkan dengan pin di sensor dan *relay* seperti pada tabel 3.1 di bawah ini.

Tabel 3.1 Rangkaian Sistem Akuisisi Data

NodeMCU ESP8266	PZEM-004T Peralatan 1
5 V	5 V
Gnd	Gnd
D5	Tx
D6	Rx
NodeMCU ESP8266	PZEM-004T Peralatan 2
5 V	5 V
Gnd	Gnd
D7	Tx
D8	Rx
NodeMCU ESP8266	Relay
5 V	VCC
Gnd	GND
D2	IN 1
D1	IN 2

Pin D5 mikrokontroler berfungsi sebagai receiver dihubungkan dengan pin Tx PZEM-004T peralatan 1. Pin D6 mikrokontroler berfungsi sebagai transmitter, pin ini dihubungkan dengan pin Rx PZEM-004T peralatan 1. Pin D7 mikrokontroler berfungsi sebagai receiver, pin ini dihubungkan dengan pin Tx PZEM-004T peralatan 2. Pin D8 mikrokontroler berfungsi sebagai transmitter, pin ini dihubungkan dengan pin Rx PZEM-004T peralatan 2. Pin D2 mikrokontroler berfungsi untuk mengontrol aliran listrik peralatan 1, pin ini dihubungkan dengan pin IN 1 *relay*. Pin D1 mikrokontroler berfungsi untuk mengontrol aliran listrik peralatan 2, pin ini dihubungkan dengan pin IN 2 *relay*. Pin 5V mikrokontroler berfungsi mengalirkan daya ke sensor dan *relay*, pin ini dihubungkan dengan pin VCC sensor dan *relay*. Pin GND mikrokontroler berfungsi mengalirkan kembali daya yang tidak terpakai oleh sensor dan *relay* ke mikrokontroler, pin ini dihubungkan dengan pin GND sensor dan *relay*.

3.6 Metode Pengujian

3.6.1 Pengujian PZEM-004T

Sensor PZEM-004T diuji untuk mengetahui tingkat akurasi sensor dalam membaca parameter listrik tegangan, arus, dan faktor daya sebelum dan setelah *adjustment*. Tiga parameter ini digunakan untuk menghitung daya aktif, daya reaktif, daya semu, konsumsi energi, dan estimasi biaya. Pengujian ini bertujuan untuk memastikan hasil pengukuran sistem sensor sesuai dengan kondisi aktual, sehingga perhitungan lima parameter listrik lainnya lebih akurat.

Proses pengujian dilakukan dengan membandingkan hasil pengukuran dari sensor PZEM-004T dengan alat ukur standar yaitu wattmeter digital DDS108 untuk mengukur tegangan, arus, dan faktor daya. Beban listrik yang digunakan dalam pengujian adalah *charger earphone*, *charger ponsel*, kipas, dan pengering rambut yang merupakan peralatan rumah tangga yang biasa digunakan sehari-hari. Setiap peralatan listrik dihubungkan ke sistem akuisisi data secara terpisah, pengukuran dilakukan sebanyak empat kali dengan interval waktu 15 detik.

Hasil pengukuran dari sensor dan alat ukur digunakan untuk menghitung persentase eror dan akurasi. Untuk meningkatkan persentase akurasi sistem akuisisi data, PZEM-004T akan disesuaikan dengan nilai ukur wattmeter digital. Setelah *adjustment*, data akan kembali dibandingkan untuk melihat apakah ada peningkatan persentase akurasi dan penurunan persentase eror.

3.6.2 Pengujian Telegram Bot

Pengujian waktu respons Telegram Bot bertujuan untuk mengetahui seberapa cepat sistem merespons perintah yang dikirimkan oleh pengguna karena mempengaruhi kepuasan pengguna. Parameter yang diuji meliputi waktu yang dibutuhkan sejak pengguna mengirimkan perintah hingga sistem memberikan respons yang sesuai, baik berupa perubahan status *relay* maupun pengiriman data konsumsi listrik.

Metode pengujian dilakukan dengan mencatat waktu saat perintah dikirim melalui aplikasi Telegram pada perangkat pengguna, kemudian mencatat waktu saat respons dari bot diterima. Pengujian dilakukan terhadap perintah-perintah yang ada. Setiap perintah diuji sebanyak lima kali untuk memperoleh rata-rata waktu respons.

Untuk perintah `/on1`, `/off1`, `/on2`, dan `/off2`, waktu respons dihitung dari saat pesan perintah dikirim hingga *relay* berubah status (dinyalakan atau dimatikan) dan bot mengirimkan konfirmasi ke pengguna. Sedangkan untuk perintah `/data1` dan `/data2`, waktu dihitung sejak perintah dikirim hingga data energi, estimasi biaya, dan tautan ke *ThingSpeak* ditampilkan di aplikasi Telegram.

