

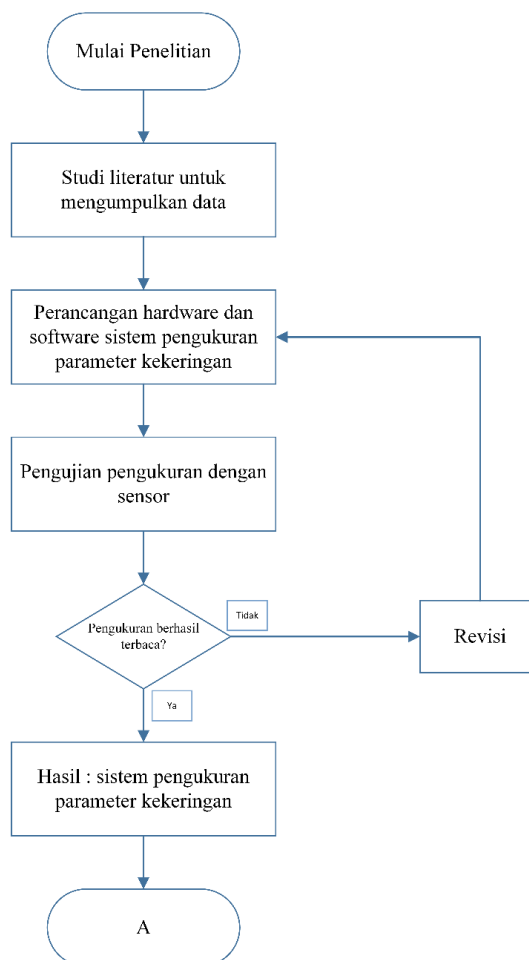
BAB III METODE PENELITIAN

3.1 Tahapan Metode Penelitian

Metode yang digunakan untuk melaksanakan penelitian ini adalah metode eksperimen, diawali dengan merencanakan sebuah rancangan terinci untuk menghasilkan sebuah sistem yang diharapkan. Adapun sistem yang diharapkan ialah sebuah sistem deteksi dini kekeringan yang dapat mengukur parameter kekeringan dan mengolahnya menjadi sebuah data digital. Data tersebut kemudian dikirimkan sebagai sebuah notifikasi terjadwal kepada pengguna yang berlangganan serta menampilkannya pada sebuah *dashboard website*.

1. Perancangan sistem pengukuran indikator kekeringan

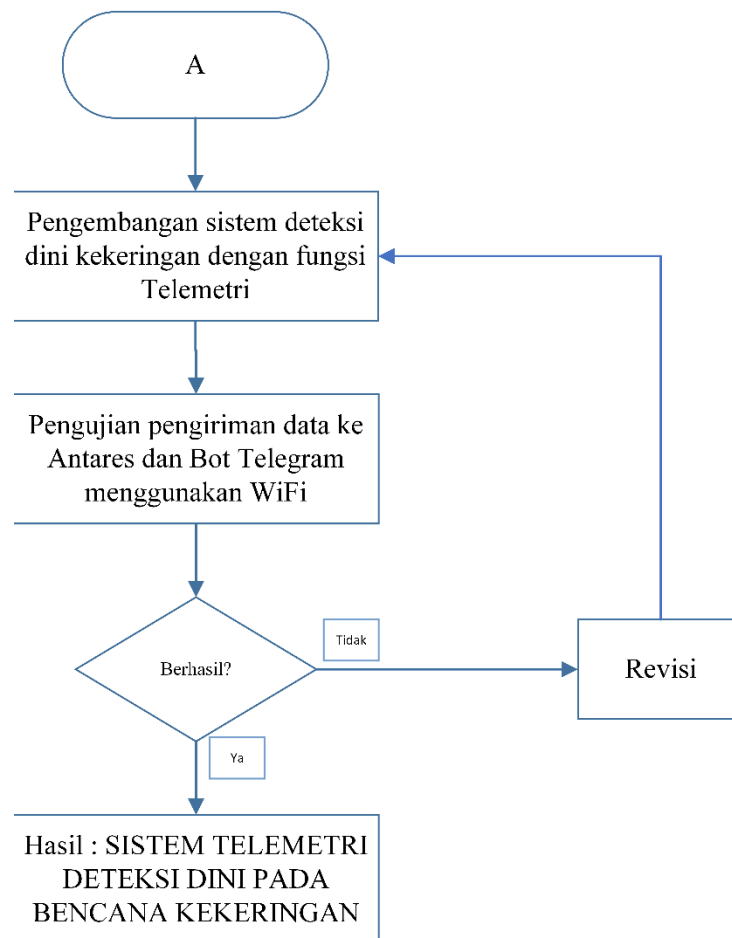
Gambar 3.1 dibawah menampilkan diagram alur perancangan sistem pengukuran parameter bencana kekeringan.



Gambar 3.1 Diagram alur untuk sistem pengukuran parameter

2. Pengembangan sistem deteksi dini kekeringan

Gambar 3.2 menampilkan diagram alur perancangan sistem deteksi dini bencana kekeringan, yang mana merupakan pengembangan lanjut dari sistem sebelumnya, yaitu sistem pengukuran parameter bencana kekeringan. Pengembangan ini mencakup pengiriman data ke sebuah *platform Internet of Things* yaitu Antares dan ke Bot Telegram dalam bentuk notifikasi kepada pengguna yang berlangganan.



Gambar 3.2 Diagram alur sistem deteksi dini kekeringan

Studi literatur merupakan langkah awal dalam penelitian ini, dimana mengumpulkan informasi yang diperlukan dalam meneliti sistem deteksi kekeringan. Informasi yang dikumpulkan adalah informasi mengenai parameter yang biasa digunakan dalam pengamatan bencana kekeringan, bagaimana indikasi apabila akan terjadi bencana kekeringan, serta sistem serupa yang telah diteliti dan dibuat baik di dalam negeri maupun di luar negeri.

Langkah selanjutnya adalah perancangan sistem yang dapat mengukur parameter bencana kekeringan yang telah ditentukan. Ada dua komponen penting dalam perancangan sistem ini, yaitu komponen *hardware* dan komponen *software*. Semua komponen *hardware* yang diperlukan dirangkai menjadi satu yang terintegrasi dengan mikrokontroler ESP32. Dilakukan juga perancangan *software*, yaitu program yang dapat membaca nilai dari pengukuran sensor-sensor oleh ESP32. Program ditulis menggunakan bahasa pemrograman C++ dalam aplikasi Arduino IDE dan diunggah ke *board* mikrokontroler dengannya.

Setelah *hardware* dan *software* selesai dibuat, tahap selanjutnya adalah melakukan pengujian pembacaan parameter dengan sensor untuk memastikan pengukuran dapat dilakukan.

Apabila sistem sudah dapat mengukur parameter dan ESP32 dapat menerima data pengukuran tersebut, sistem akan dikembangkan lebih lanjut menjadi sistem deteksi dini kekeringan (*drought early warning system*). Sistem tersebut merupakan hasil dari pengembangan *software*, dimana program sebelumnya ditambahkan dengan program pengiriman data yang terintegrasi dengan mikrokontroler ESP32. Sehingga menghasilkan sistem deteksi dini kekeringan yang dapat mengirimkan notifikasi ke *Bot* Telegram serta mengirimkan data ke sebuah *platform Internet of Things*, yaitu Antares yang nantinya data tersebut dapat digunakan sebagai data yang ditampilkan pada sebuah *website dashboard*.

Tahap akhir dari penelitian ini adalah pengambilan kesimpulan dan saran yang didasarkan pada tujuan, rumusan masalah penelitian serta dari pembahasan dan analisis sistem yang telah dibuat.

3.2 Sistem Deteksi Dini Kekeringan (*Drought Early Warning System*)

Perancangan sistem deteksi dini kekeringan terdiri dari purwarupa sistem pengukuran yang dapat mengukur parameter indikator bencana kekeringan. Berikut merupakan perancangan sistem deteksi dini kekeringan.

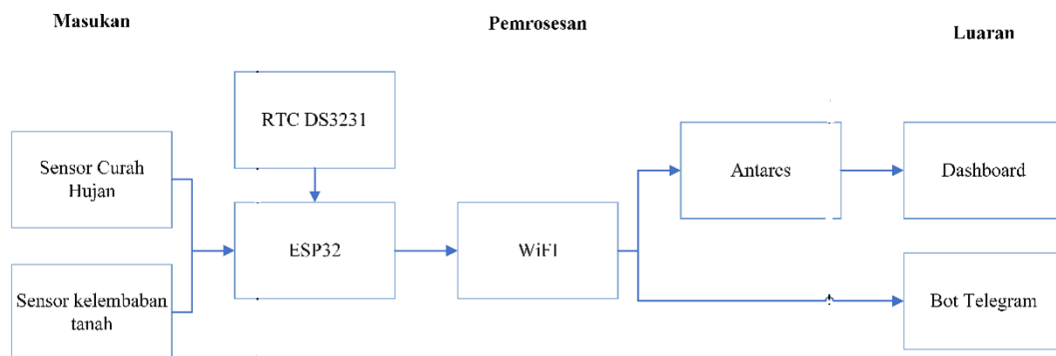
3.2.1 Perangkat Penunjang Penelitian

Perangkat penunjang penelitian yang digunakan terbagi menjadi 2 bagian, yaitu *hardware* dan *software*. Adapun *hardware* yang digunakan adalah sensor curah

hujan, sensor kelembaban tanah, ESP32, Baterai, RTC DS3231. Adapun *software* yang digunakan diantaranya adalah POSTMAN, Arduino IDE, Visual Studio Code, Google Chrome, dan ANTARES (*platform Internet of Things cloud storage*).

3.2.2 Prinsip Kerja

Prinsip kerja sistem deteksi dini kekeringan tergambar melalui diagram blok yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Sistem deteksi dini kekeringan

Diagram blok pada Gambar 3.3 menjelaskan bahwa sistem deteksi dini kekeringan terbagi menjadi 3 bagian yaitu masukan, pemrosesan dan luaran setelah diunggah program pada *board* ESP32. Pada masukan, Langkah selanjutnya adalah perancangan sistem yang dapat mengukur parameter bencana kekeringan yang telah ditentukan. ESP32 akan menerima masukan data terukur dari berbagai sensor yang sudah diintegrasikan.

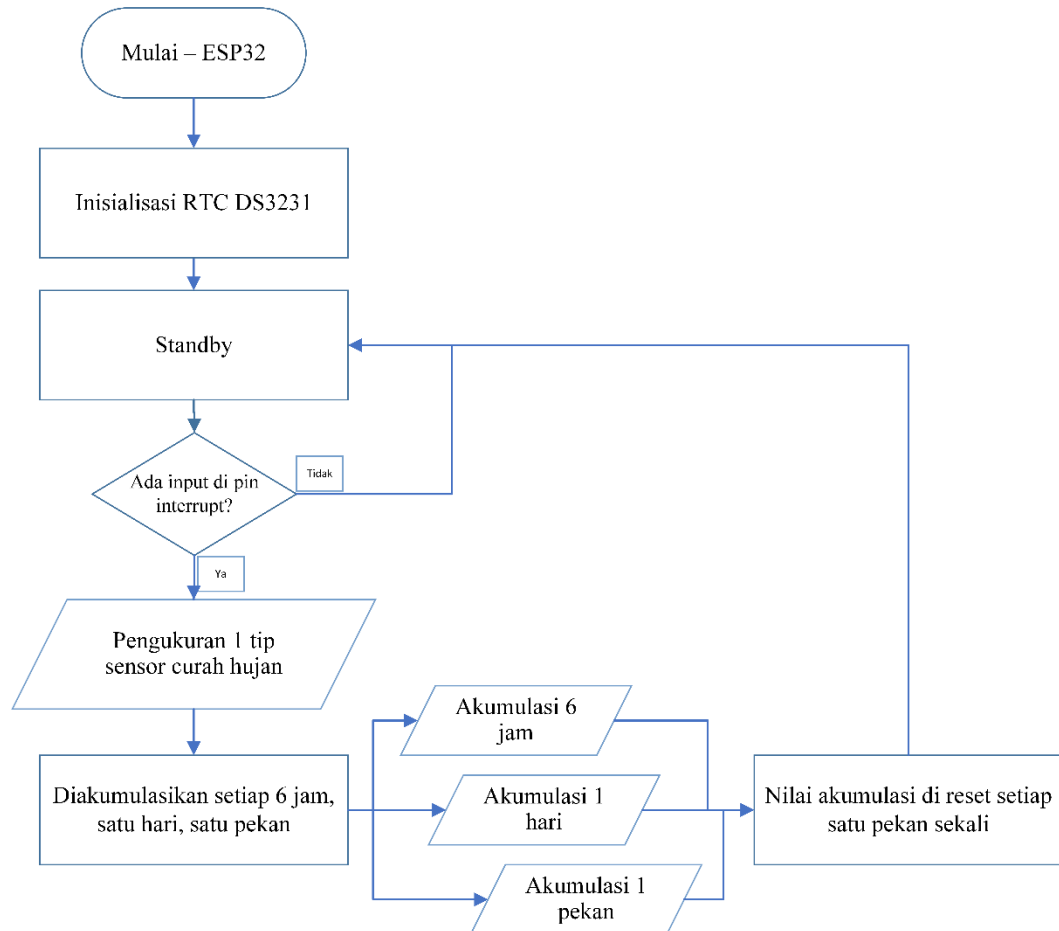
Pada pemrosesan, ESP32 akan memproses data dari sensor-sensor serta mengirimkan data tersebut ke platform ANTARES dan *Bot* Telegram menggunakan konektivitas Wi-Fi. RTC DS32 merupakan modul yang berfungsi sebagai *Real Time Clock*, atau penunjuk jam bagi mikrokontroler ESP32, modul ini adalah bagian penting dari sensor curah hujan dalam proses pembacaan curah hujan.

Data yang disimpan pada platform ANTARES selanjutnya dengan menggunakan metode GET, *website* dapat mendapatkan data pengukuran yang nantinya ditampilkan pada *dashboard*.

3.2.3 Algoritma

Terdapat beberapa algoritma dalam perancangan sistem deteksi dini kekeringan, diantaranya sebagai berikut:

1. Algoritma pembacaan curah hujan oleh sensor curah hujan

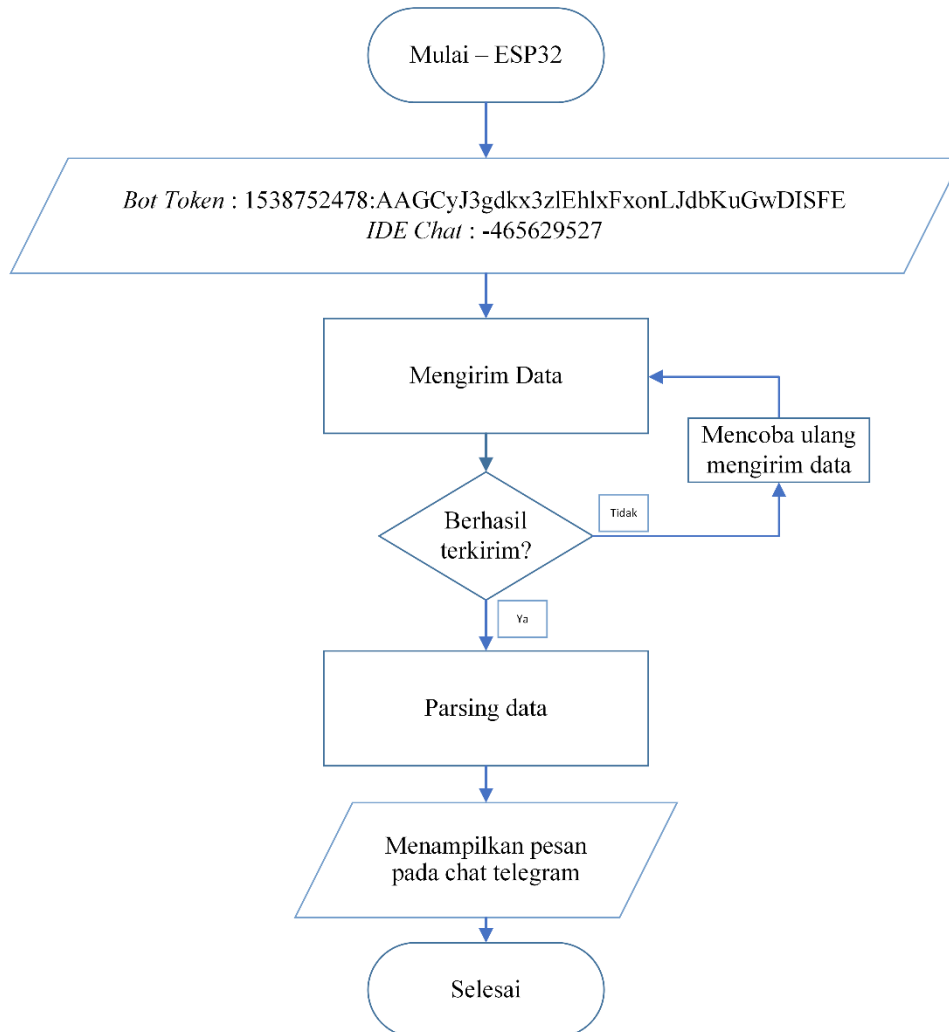


Gambar 3.4 Diagram blok status pembacaan curah hujan

Diagram blok pada Gambar 3.4 menggambarkan algoritma pembacaan curah hujan oleh sensor curah hujan:

- Tahap pertama merupakan tahap inisialisasi *Real Time Clock*, ESP32 akan membaca data waktu pada RTC DS3231.
- Setelah melakukan inisialisasi, ESP32 akan berada dalam modus standby sampai menerima input pada pin *interrupt*.
- Apabila ESP32 menerima input *pulse* pada pin *interrupt*, ESP32 akan menghitung tiap *pulse* sebagai satu tip pada sensor curah hujan. Satu tip bernilai 0.70 mm curah hujan.
- Perhitungan tip diakumulasi setiap enam jam, satu hari dan satu pekan.
- Setiap pekan perhitungan tip akan dihapus, dimulai kembali dari nilai 0 mm curah hujan.

2. Algoritma pengiriman data ke Bot Telegram (tidak berhasil harus apa?)



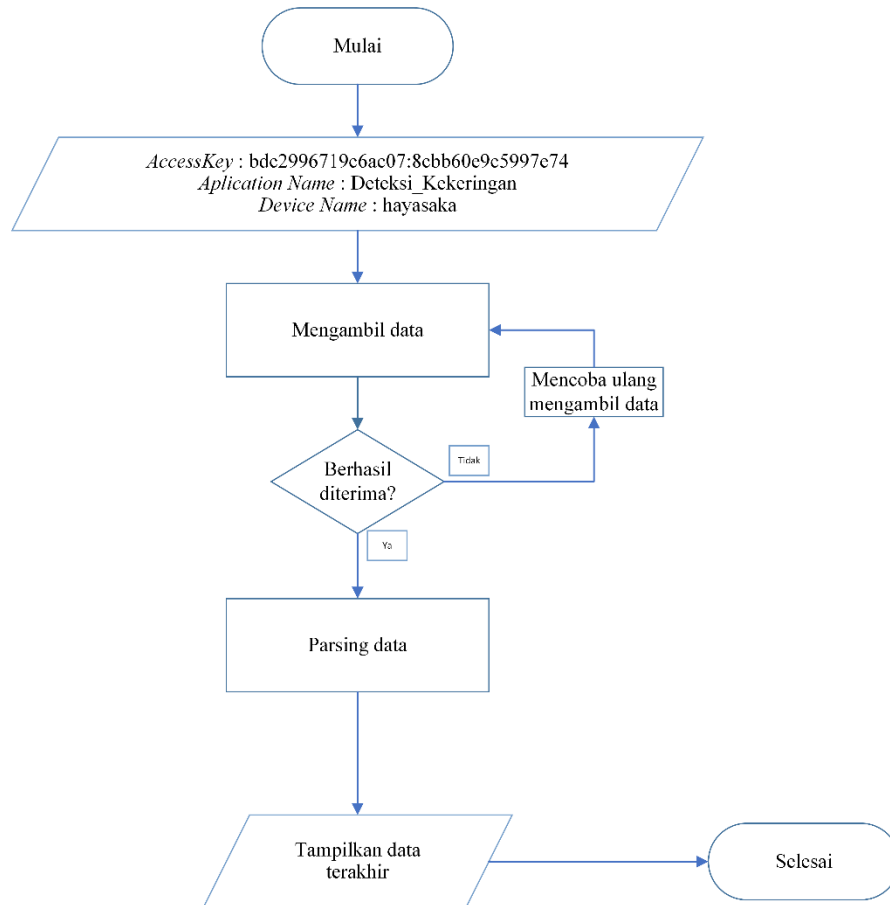
Gambar 3.5 Diagram pengiriman data ke Bot Telegram

Diagram blok pada Gambar 3.5 menggambarkan algoritma pengiriman data ke Bot Telegram:

- Tahap pertama merupakan inisialisasi pada ESP32, aplikasi akan membaca input data yang diperlukan dari Telegram untuk melakukan pengiriman data, yaitu data *Bot Token*, *id Chat*, dan data *final* yang akan dikirimkan oleh perangkat.
- Setelah semua data siap, sistem akan melakukan percobaan pengiriman data ke Bot Telegram.
- Jika data berhasil diterima oleh Telegram maka Telegram akan melakukan *parsing* data (memilih data mana yang diperlukan untuk ditampilkan sebagai notifikasi kepada pelanggan).

- d. Tahap terakhir Telegram akan menampilkan data sebagai notifikasi pesan kepada pelanggan.

3. Algoritma penampilan data pada *dashboard website*



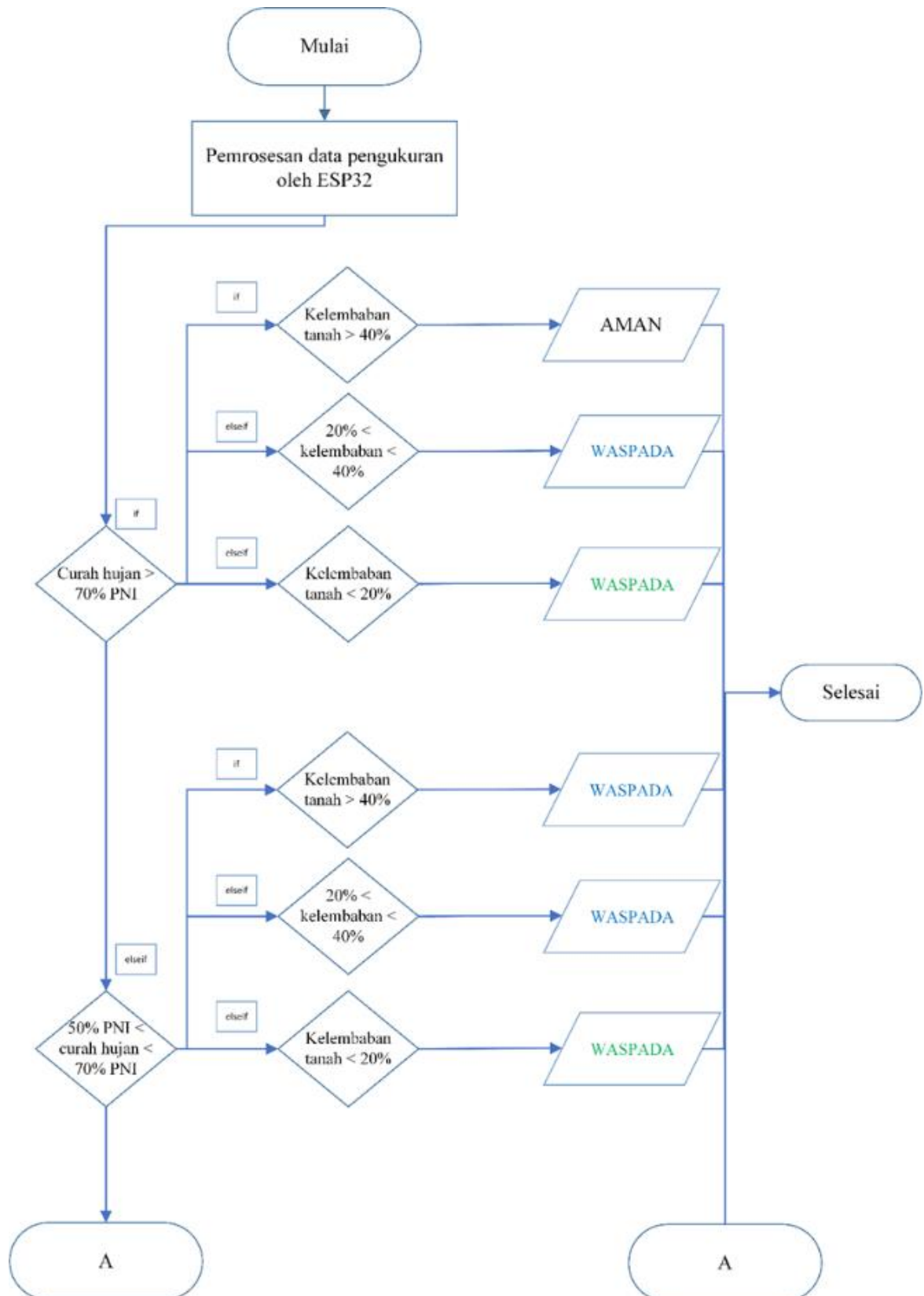
Gambar 3.6 Diagram blok penampilan data pada dashboard website

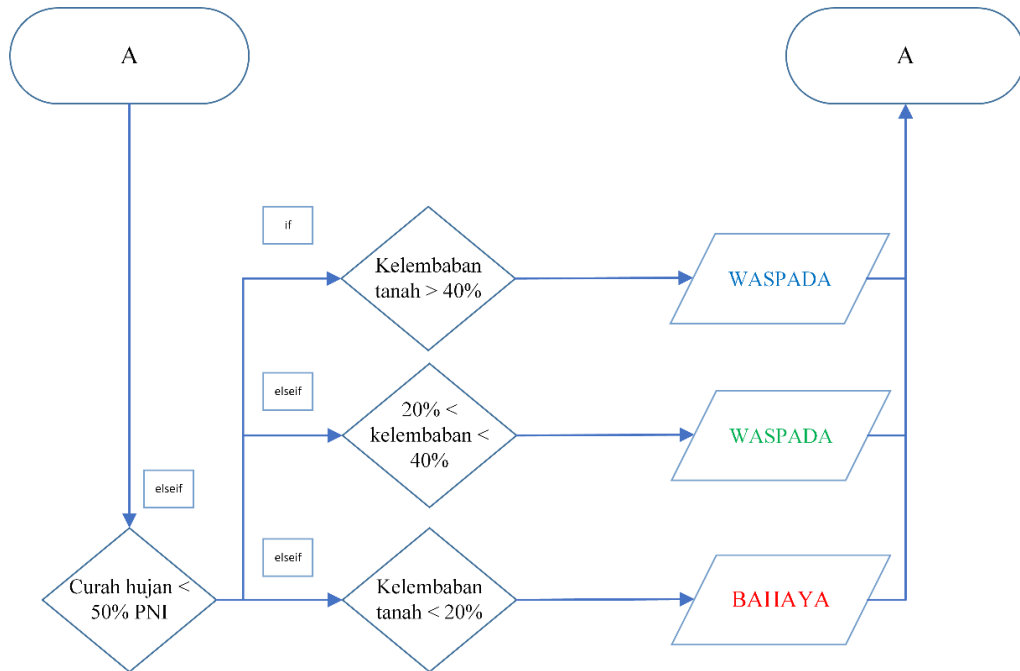
Diagram blok pada Gambar 3.6 menggambarkan algoritma pengambilan data dari Antares oleh untuk ditampilkan pada *dashboard*:

- Tahap pertama merupakan tahap inialisasi dimana aplikasi akan membaca input data yang dibutuhkan dari Antares untuk melakukan pengambilan data (*get*), yaitu data *accesskey*, *application name*, dan *device name*.
- Tahap kedua aplikasi akan melakukan pengambilan data (*get*) dari platform ANTARES.
- Jika data berhasil diterima oleh aplikasi maka aplikasi akan melakukan *parsing* data (memilih data mana yang diperlukan untuk ditampilkan sebagai notifikasi kepada pelanggan).

- d. Tahap terakhir aplikasi *website* akan menampilkan data pengukuran yang telah di-*parsing* sebelumnya.

4. Algoritma notifikasi ke Telegram berdasarkan data pengukuran





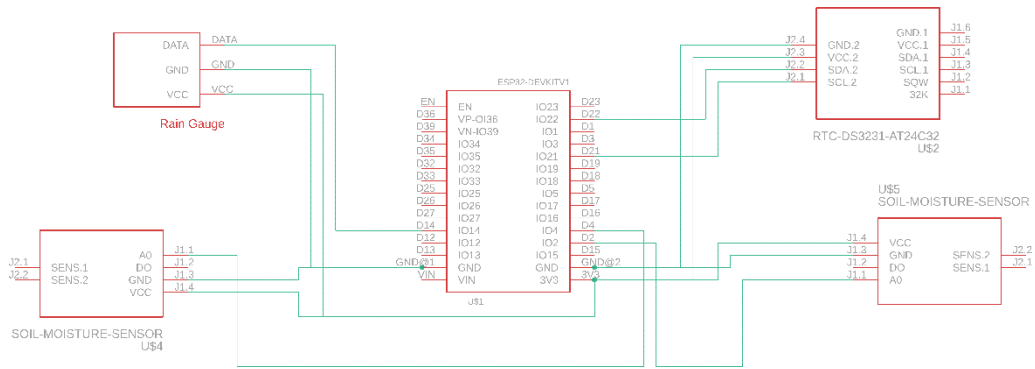
Gambar 3.7 Diagram blok notifikasi ke Telegram berdasarkan data pengukuran Diagram blok pada Gambar 3.7 menggambarkan informasi notifikasi ke Telegram berdasarkan data pengukuran parameter. Ada tiga jenis notifikasi yang didasarkan kepada nilai parameter curah hujan dan kelembaban tanah, yaitu Aman, Waspada dan Bahaya yang dirincikan pada tabel 3.1.

Tabel 3.1 Detil parameter kekeringan

AND	Curah hujan > 70% PNI	50% PNI < curah hujan < 70% PNI	Curah hujan < 50% PNI
Kelembaban tanah > 40%	AMAN	WASPADA	WASPADA
20% < kelembaban < 40%	WASPADA	WASPADA	WASPADA
Kelembaban tanah < 20%	WASPADA	WASPADA	BAHAYA

3.2.4 Perancangan perangkat keras (hardware)

Perancangan perangkat dimulai dengan merancang rangkaian elektronika pada mikrokontroler dan sensor-sensor yang digunakan. Gambar 3.8 merupakan skematik rancangan rangkaian dari perangkat deteksi dini kekeringan dengan detail pengaturan pin pada tabel 3.1.



Gambar 3.8 Skematik rangkaian

Sebagai sumber daya mikrokontroler, akan digunakan sebuah *shield battery* 18650 yang langsung disambungkan ke port Micro USB ESP32 menggunakan kabel. Shield battery digunakan sebagai catu daya bagi ESP32, dapat memberikan tegangan sebesar 3.3V dimana merupakan tegangan ideal bagi ESP32.

Tabel 3.2 Detail pengaturan pin rangkaian

No.	Pin Modul	ESP32
1	Sensor kelembaban tanah 1	2
2	Sensor kelembaban tanah 2	4
3	<i>Rain Gauge</i>	14
4	SDA – RTC DS3231	21
5	SCL – RTC DS3231	22

3.2.5 Perancangan Kode Program pada Arduino IDE

Kode (*sketch*) program Arduino merupakan sebuah *coding* dengan bahasa pemrograman C++ yang merupakan bagian *software* dari sistem deteksi dini kekeringan. Kode pada pengembangan sistem deteksi dini kekeringan terdiri dari program pembacaan waktu dari RTC DS 3231, program pembacaan curah hujan dari sensor curah hujan, program pembacaan sensor kelembaban tanah, serta dua program konektivitas yaitu pengiriman data ke *Bot Telegram* dan *Platform Internet of Things ANTARES*. Ada empat *library* yang akan digunakan, yaitu:

- ANTARESESP32HTTP sebagai *library* untuk koneksi perangkat dengan *platform Internet of Things ANTARES*.
- UniversalTelegramBot sebagai *library* koneksi perangkat dengan Bot Telegram dan mengirim pesan notifikasi.

- c. RTCLib sebagai *library* untuk mengakses fungsi *Real Time Clock* dari RTC DS3231.
- d. WiFiClientSecure sebagai *library* untuk koneksi perangkat dengan WiFi.