BAB IV
PROSES PEMBUATAN ALAT

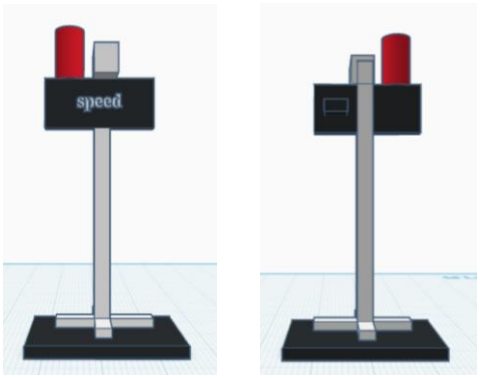
4.1 Pembuatan Perangkat Keras (Hardware)

Pembuatan perangkat keras sistem deteksi kecepatan kendaraan ini bertujuan untuk merealisasikan sistem berbasis sensor radar mmWave DFRobot C4001 dan mikrokontroler ESP32 yang dapat mendeteksi kecepatan kendaraan secara real-time. Sensor radar mmWave berfungsi dengan prinsip efek Doppler, di mana perubahan frekuensi gelombang yang dipantulkan oleh kendaraan digunakan untuk menghitung kecepatan. Data yang diperoleh dikirimkan ke mikrokontroler ESP32 melalui komunikasi UART, yang kemudian mengarahkan data tersebut untuk ditampilkan pada LED Dot Matrix Display P10 dan dikirim ke server ThingSpeak melalui Wi-Fi untuk pemantauan jarak jauh.

Mikrokontroler ESP32 berperan sebagai pusat pemrosesan, mengelola dua jalur data: pertama untuk tampilan lokal dan kedua untuk pengiriman data ke cloud. Sistem ini juga dilengkapi dengan komponen peringatan berupa lampu warning dan buzzer yang aktif saat kendaraan melaju melebihi batas kecepatan yang ditentukan. Ketika sensor mendeteksi pelanggaran kecepatan, ESP32 mengendalikan lampu warning dan buzzer untuk memberi peringatan visual dan auditori kepada pengendara dan pengawas, dengan daya yang diatur melalui driver relai atau modul transistor.

Untuk pengendalian tampilan LED Dot Matrix, Arduino Nano digunakan sebagai pengendali tambahan yang menerima data dari ESP32 dan mengatur tampilan menggunakan komunikasi SPI. Sistem ini dirakit pada PCB khusus yang mengoptimalkan penggunaan komponen dan memastikan efisiensi daya. Setelah perakitan, sistem diuji dengan kendaraan nyata untuk memastikan akurasi deteksi kecepatan dan fungsionalitas pengiriman data ke ThingSpeak. Dengan kemampuan pemantauan jarak jauh melalui aplikasi dan penambahan

peringatan langsung, sistem ini menyediakan solusi yang efektif dan efisien untuk meningkatkan keselamatan lalu lintas.



Gambar 4. 1 Desain 3D Alat Sistem Deteksi Kendaraan

4.1.1 Alat dan Bahan

Proses selanjutnya dalam pembuatan perangkat keras adalah melakukan persiapan terhadap alat dan bahan yang akan digunakan untuk merakit sistem sesuai rancangan yang telah dijelaskan sebelumnya. Tahapan ini meliputi inventarisasi seluruh komponen elektronik utama seperti sensor, mikrokontroler, modul tampilan, serta perangkat pendukung lainnya. Semua komponen disiapkan dalam jumlah dan spesifikasi yang sesuai untuk memastikan sistem dapat bekerja secara optimal.

Berikut adalah daftar lengkap bahan dan komponen sistem deteksi kecepatan kendaraan berbasis sensor mmWave C4001 dan ESP32:

Tabel 4. 1 Komponen yang digunakan

No	Nama Bahan	Jumlah
1	ESP32 DevKit V1	1 buah
2	Sensor Radar mmWave C4001	1 buah
3	Arduino Nano	1 buah
4	LED Dot Matrix Display P10 (32x16cm)	1 buah
5	Step-down Regulator LM2596	1 buah
6	Adaptor 12V DC 10A	1 buah

7	PCB (Printed Circuit Board)	1 lembar
8	Header Pin Male/Female	Secukupnya
9	Kabel Jumper	Secukupnya
10	Socket DC Male/Female	1 pasang
11	Konektor Terminal Screw	1 buah
12	Resistor dan kapasitor pendukung	Secukupnya
13	Lampu Warning Buzzer	1 buah
15	Kabel data USB	1 buah

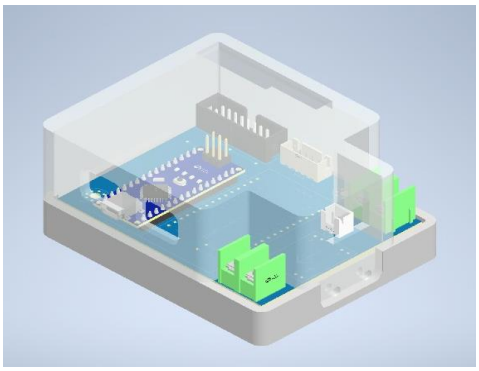
Tabel 4. 2 Alat yang Digunakan

No	Nama Alat	Jumlah
1	Multimeter	1 buah
2	Solder Listrik	1 buah
3	Timah Solder	1 gulung
4	Gunting Kabel	1 buah
5	Cutter	1 buah
6	Obeng Set	1 set
7	Alat Tulis	1 set
8	Bor PCB	1 buah
9	Breadboard (uji awal)	1 buah

4.1.2 Langkah-Langkah Pembuatan

Berikut merupakan tahapan-tahapan pembuatan perangkat keras untuk sistem deteksi kecepatan kendaraan berbasis sensor mmWave dan ESP32 yang dilakukan secara sistematis mulai dari persiapan, perakitan, hingga pengujian awal:

1. Pembuatan 3D Box Komponen Sistem Deteksi Kecepatan Kendaraan



Gambar 4. 2 Desain 3D Box Komponen

- Menyusun seluruh komponen utama yang telah tercantum pada Tabel 4.1 dan Tabel 4.2.
- Memastikan spesifikasi tiap komponen sesuai kebutuhan, termasuk ESP32, sensor radar mmWave C4001, Arduino Nano, LED Dot Matrix P10, modul step-down LM2596, dan Dual Mosfet Driver D4148.
- Melakukan pengecekan awal pada setiap komponen menggunakan multimeter untuk memastikan tidak ada kerusakan.

2. Perancangan dan Pemasangan di PCB:
   - Desain awal PCB dibuat menggunakan software EasyEDA berdasarkan wiring diagram sistem.
   - Komponen seperti ESP32, Arduino Nano, dan header pin disusun pada PCB sesuai dengan jalur komunikasi UART dan SPI.
   - Jalur daya dan ground dirancang dengan lebar cukup untuk mencegah drop tegangan, terutama untuk suplai LED Dot Matrix.

3. Penyolderan Komponen:
   - Proses penyolderan dimulai dari komponen kecil seperti header pin hingga modul besar seperti regulator dan konektor power.
   - Jalur TX sensor radar mmWave disolder ke pin RX2 (GPIO16) pada ESP32.

- Pemasangan step-down dilakukan dengan menyolder input VIN dan GND ke jalur power adaptor, dan output 5V ke Arduino dan LED.

4. Penyusunan dan Penempatan Komponen:
   - Komponen-komponen utama diposisikan sesuai tata letak PCB untuk menjaga sirkulasi udara dan meminimalkan interferensi sinyal.
   - Kabel sensor radar dipasang di bagian luar box untuk diarahkan ke jalur pembacaan objek kendaraan.
   - Header pin tambahan dipasang untuk kemudahan pemrograman atau modifikasi di masa depan.

5. Pemasangan Catu Daya:
   - Adaptor 12V DC dihubungkan ke input power menggunakan konektor terminal screw.
   - Output adaptor diteruskan ke modul step-down yang mengubahnya menjadi 5V untuk Arduino Nano dan LED Dot Matrix.
   - Ground dari semua komponen disatukan untuk menjaga kestabilan sistem.

6. Pengujian Koneksi dan Fungsionalitas:
   - Setelah semua komponen terpasang, dilakukan pengujian koneksi menggunakan multimeter untuk memastikan tidak ada sambungan yang putus atau short.
   - Sistem dinyalakan pertama kali untuk melihat apakah sensor dapat mengirimkan data kecepatan ke ESP32 dan ditampilkan pada LED Dot Matrix.
   - ESP32 juga diuji untuk memastikan data terkirim ke platform ThingSpeak dan muncul pada aplikasi monitoring di smartphone.

7. Perakitan Akhir dan Penataan Fisik:
    - Setelah pengujian berhasil, PCB dan seluruh komponen dimasukkan ke dalam box yang sudah dibuat.
    - Bagian depan box dilubangi untuk menampilkan LED Dot Matrix agar dapat terbaca oleh di lapangan.
    - Posisi sensor radar diatur dengan sudut dan jarak yang tepat agar dapat membaca kendaraan yang melintas dengan akurat.

8. Finalisasi Sistem dan Integrasi dengan Internet (Cloud)
    - Setelah semua rangkaian selesai dirakit dan diuji satu per satu, sistem kembali diuji secara menyeluruh untuk memastikan semua bagian bekerja dengan baik.
    - Pengujian dilakukan dengan cara menyalakan alat selama beberapa waktu untuk memastikan tidak ada komponen yang panas berlebihan atau tidak berfungsi.
    - Sensor radar diuji apakah bisa mendeteksi kecepatan kendaraan dan mengirimkannya ke ESP32 dengan benar.
    - Data dari ESP32 kemudian dicek apakah sudah bisa dikirim ke Arduino Nano dan ditampilkan di LED Dot Matrix dengan jelas.
    - Selain itu, ESP32 juga diuji apakah sudah bisa mengirim data ke server ThingSpeak melalui Wi-Fi.
    - ThingSpeak berfungsi sebagai penyimpanan data online (cloud) yang dapat diakses kapan saja melalui internet.
    - Hasil data yang dikirim ke ThingSpeak juga dapat dilihat dari smartphone, sehingga pengguna bisa memantau kecepatan kendaraan dari jarak jauh.
    - Setelah semua bagian berhasil bekerja secara bersamaan mulai dari pendeteksian, penampilan data, hingga pengiriman ke server, maka sistem dinyatakan sudah siap digunakan di lapangan.

4.2 Pembuatan Perangkat Lunak (Software)

Pembuatan perangkat lunak dalam sistem deteksi kecepatan kendaraan ini bertujuan untuk mengintegrasikan seluruh fungsi sensor, mikrokontroler, tampilan data, serta koneksi ke platform cloud IoT agar sistem dapat berjalan secara otomatis dan real-time. Proses ini mencakup tahapan penulisan kode program (pemrograman), pengujian fungsionalitas sensor, komunikasi antar perangkat, hingga pengiriman data ke server cloud dan aplikasi monitoring di smartphone.

Perangkat lunak dirancang menggunakan bahasa pemrograman Arduino (C/C++) yang ditanamkan ke dalam mikrokontroler ESP32 dan Arduino Nano. ESP32 berperan sebagai pusat pemrosesan data, yang bertugas membaca data kecepatan dari sensor radar mmWave C4001 melalui komunikasi serial UART, kemudian mengirimkan data ke dua jalur: pertama, untuk ditampilkan secara lokal melalui LED Dot Matrix; dan kedua, untuk dikirim ke platform ThingSpeak melalui koneksi Wi-Fi.

Program utama yang ditanam pada ESP32 berisi beberapa fungsi penting, antara lain: inisialisasi sensor dan Wi-Fi, pembacaan data serial dari sensor, parsing data kecepatan, serta pengiriman data ke ThingSpeak menggunakan metode HTTP POST. Selain itu, terdapat juga pengaturan waktu delay dan kondisi jika koneksi Wi-Fi terputus agar sistem dapat melakukan percobaan koneksi ulang secara otomatis.

Sementara itu, Arduino Nano menerima data kecepatan dari ESP32 melalui komunikasi serial, lalu meneruskannya ke tampilan LED Dot Matrix P10 menggunakan komunikasi SPI. Pada bagian ini, program ditulis menggunakan pustaka DMD2 atau pustaka DotMatrix lainnya yang mendukung tampilan angka secara digital pada modul LED 32x16 cm.

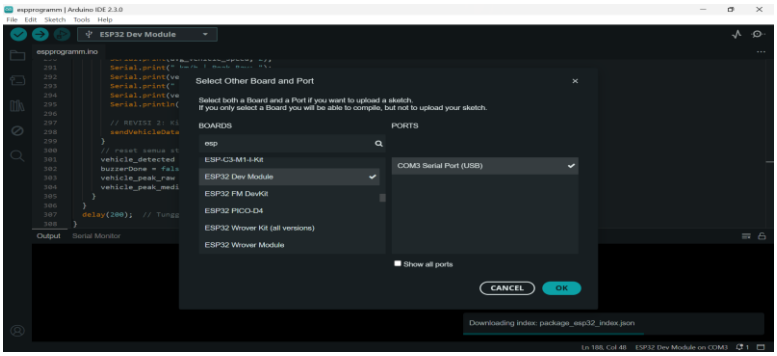
Penggunaan ThingSpeak sebagai platform IoT memungkinkan data kecepatan kendaraan yang ditangkap oleh alat dapat dikirimkan dan disimpan ke server cloud. Data ini selanjutnya dapat divisualisasikan dalam bentuk grafik pada dashboard ThingSpeak yang dapat diakses kapan saja oleh pengguna. Dashboard

ini juga dapat dibuka melalui browser atau aplikasi monitoring di smartphone, sehingga memudahkan pemantauan kecepatan kendaraan dari lokasi yang jauh.

Perangkat lunak juga dirancang untuk memastikan efisiensi penggunaan memori dan kestabilan sistem, dengan memperhatikan logika pemrosesan data, penggunaan variabel global, serta pemrosesan delay agar tidak mengganggu pengiriman data secara periodik.

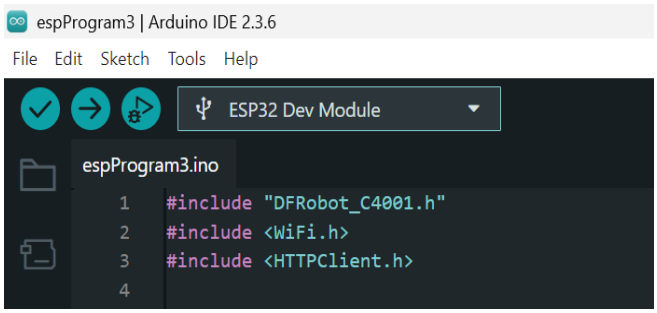
Dengan perangkat lunak yang telah dirancang dan diuji, sistem dapat bekerja secara otomatis mulai dari pembacaan data sensor, pengolahan data, penampilan hasil, hingga pengiriman ke server. Hal ini memungkinkan sistem beroperasi secara mandiri di lapangan dengan tetap memberikan akses informasi kepada pengguna secara real-time melalui internet.

4.2.1 Pembuatan Program ESP32 pada Arduino IDE



Gambar 4. 3 Pemilihan Board Program ESP32

Pada tahap ini, perangkat lunak untuk mengendalikan sistem deteksi kecepatan kendaraan dirancang menggunakan Arduino IDE sebagai platform pemrograman utama. Proses pemrograman dimulai dengan membuka aplikasi Arduino IDE yang telah terinstal pada komputer atau laptop. Langkah pertama yang dilakukan adalah melakukan konfigurasi awal dengan memilih board yang sesuai, yaitu ESP32 Dev Module untuk mikrokontroler ESP32. Pemilihan board dilakukan melalui menu Tools > Board > ESP32 Dev Module untuk ESP32, dilanjutkan dengan menentukan port komunikasi (COM port) yang sesuai dengan perangkat yang sedang terhubung.

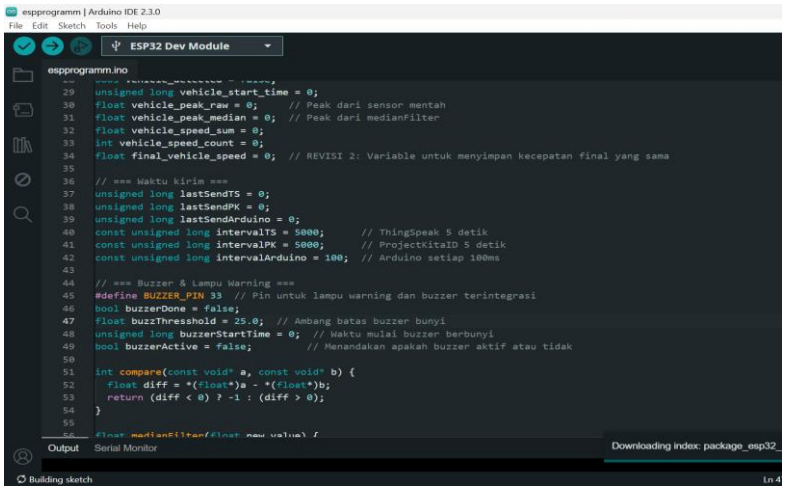


Gambar 4. 4 Library Program ESP32

Langkah selanjutnya dalam pemrograman ESP32 menggunakan Arduino IDE adalah mengimpor pustaka (library) yang dibutuhkan. Library ini digunakan agar sistem dapat mengenali dan memanfaatkan fungsi-fungsi yang tersedia untuk sensor dan komunikasi internet. Beberapa pustaka penting yang digunakan antara lain DFRobot_C4001.h, WiFi.h, dan HTTPClient.h.

Keterangan:

1. DFRobot_C4001.h adalah pustaka resmi dari DFRobot yang digunakan untuk berkomunikasi dengan sensor radar mmWave C4001 melalui antarmuka UART. Pustaka ini memungkinkan mikrokontroler untuk membaca data kecepatan dari sensor secara langsung.
2. WiFi.h adalah pustaka bawaan dari ESP32 yang memungkinkan perangkat terhubung ke jaringan Wi-Fi. Ini penting karena sistem Anda akan mengirimkan data ke server online secara real-time.
3. HTTPClient.h digunakan untuk membuat permintaan HTTP POST atau GET ke server. Dalam sistem Anda, pustaka ini dipakai untuk mengirim data ke platform seperti ThingSpeak atau aplikasi smartphone melalui koneksi internet.

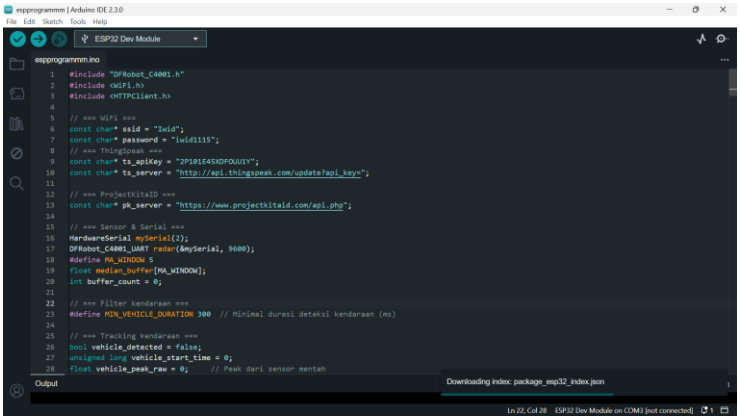


Gambar 4. 5 Program Data Thingspeak dan Aplikasi Smartphone

Pada bagian program ini, sistem menggunakan variabel lastSendTS, lastSendPK, dan lastSendArduino untuk mengatur interval pengiriman data ke platform ThingSpeak, Aplikasi Smartphone, dan Arduino Nano. Interval pengiriman ditentukan oleh konstanta intervalTS, intervalPK, dan intervalArduino, masing-masing sebesar 5 detik untuk ThingSpeak dan Aplikasi, serta 1 detik untuk Arduino Nano. Sistem membandingkan waktu saat ini yang diperoleh melalui fungsi millis() dengan waktu terakhir pengiriman yang disimpan dalam variabel lastSendTS, lastSendPK, atau lastSendArduino. Ketika selisih waktu melebihi interval yang ditentukan, maka sistem akan memanggil fungsi pengiriman data yang sesuai, yaitu sendToThingSpeak(), sendToProjectKitaID(), atau pengiriman data ke Arduino Nano untuk ditampilkan pada LED Dot Matrix.

Proses ini memastikan bahwa pengiriman data ke tiap platform dilakukan secara real-time namun tidak saling mengganggu, karena interval pengiriman ke Arduino lebih cepat untuk mendukung tampilan lokal, sedangkan pengiriman ke platform cloud dilakukan dengan interval lebih panjang agar tidak membebani koneksi Wi-Fi. Pendekatan ini menggunakan prinsip non-blocking timing, sehingga sistem tetap dapat membaca data dari sensor, memprosesnya, dan mengupdate tampilan LED Dot Matrix secara simultan tanpa menunda proses lainnya. Bagian program ini berperan penting dalam pemantauan kecepatan kendaraan berbasis IoT karena memungkinkan data kecepatan terkirim secara

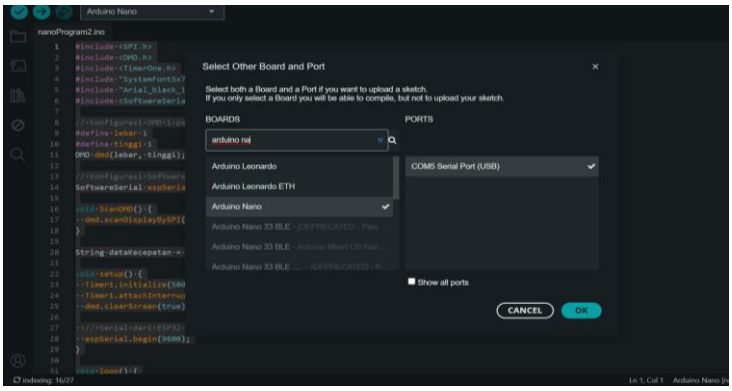
berkala ke platform cloud maupun ditampilkan secara lokal, mendukung monitoring real-time dan pengambilan keputusan cepat di lapangan.



Gambar 4. 6 Program ESP32 Arduino IDE

Setelah program berhasil diverifikasi tanpa adanya kesalahan, langkah berikutnya adalah melakukan proses unggah (upload) program ke papan ESP32. Proses ini dilakukan dengan cara menekan ikon Upload (→) di bagian atas Arduino IDE. Proses ini akan mengirimkan seluruh baris kode program ke memori ESP32 melalui USB.

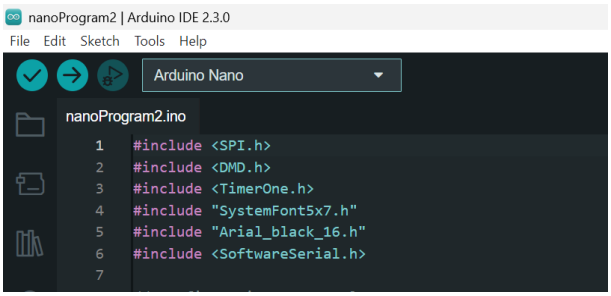
4.2.2 Pembuatan Program Arduino Nano pada Arduino IDE



Gambar 4. 7 Pemilihan Board Program Arduino Nano

Pada tahap ini, perangkat lunak untuk mengendalikan LED Dot Matrix P10 dirancang menggunakan Arduino IDE. Proses pemrograman dimulai dengan membuka aplikasi Arduino IDE yang telah diinstal pada komputer. Setelah itu, dilakukan konfigurasi awal dengan memilih board yang sesuai, yaitu Arduino

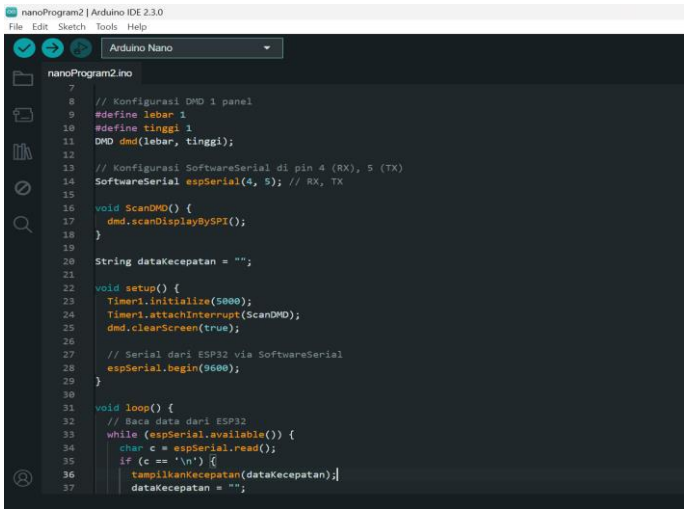
Nano, melalui menu Select Board dan menentukan port komunikasi (COM port) yang terhubung dengan Arduino Nano.



Gambar 4. 8 Library Program ESP32

Program sistem deteksi kecepatan kendaraan menggunakan beberapa library yang mendukung pengolahan data dan tampilan pada LED Dot Matrix serta komunikasi serial. Library <SPI.h> digunakan untuk mengatur komunikasi serial dengan protokol SPI (Serial Peripheral Interface) antara Arduino Nano dan modul LED Dot Matrix P10, memungkinkan pengiriman data secara sinkron dan cepat. Library <DMD.h> berfungsi sebagai driver untuk mengontrol LED Dot Matrix, menyediakan fungsi untuk menampilkan teks, angka, atau animasi secara real-time, termasuk pengaturan refresh display agar tampilan tetap stabil dan tidak berkedip. Library <TimerOne.h> digunakan untuk mengatur timer berbasis interrupt, sehingga pembaruan tampilan LED Dot Matrix dapat dilakukan secara periodik dan presisi tanpa mengganggu proses utama program.

Selain itu, library "SystemFont5x7.h" dan "Arial_black_16.h" menyediakan font yang digunakan pada LED Dot Matrix, di mana SystemFont5x7 adalah font kecil dengan ukuran 5x7 piksel untuk menampilkan teks sederhana, sedangkan Arial_black_16 adalah font besar yang lebih jelas untuk menampilkan angka kecepatan kendaraan secara mudah dibaca dari jarak jauh. Library <SoftwareSerial.h> memungkinkan ESP32 atau Arduino membuat port serial tambahan melalui pin digital, yang digunakan untuk komunikasi data antara ESP32 dan Arduino Nano, sehingga data kecepatan kendaraan dapat dikirim secara paralel ke LED Dot Matrix untuk tampilan lokal.



Gambar 4. 9 Program Arduino Nano

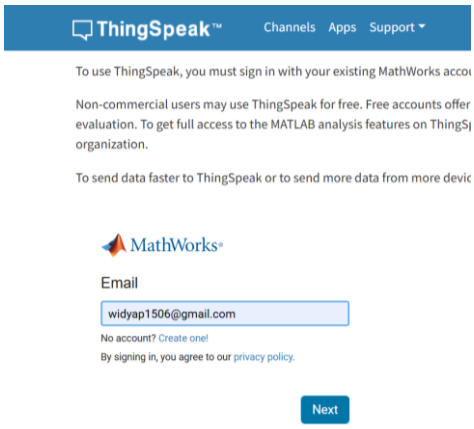
Setelah program berhasil diverifikasi tanpa adanya kesalahan, langkah berikutnya adalah melakukan proses unggah (upload) program ke papan Arduino Nano. Proses ini dilakukan dengan cara menekan ikon Upload (→) di bagian atas Arduino IDE. Proses ini akan mengirimkan seluruh baris kode program ke memori Arduino Nano melalui USB.

4.2.3 Perangkat Lunak Platform ThinkSpeak

ThingSpeak adalah platform berbasis cloud computing yang dirancang khusus untuk mendukung pengembangan Internet of Things (IoT). Platform ini dapat menerima, menyimpan, memproses, dan menampilkan data dari berbagai perangkat IoT secara real-time melalui koneksi internet. ThingSpeak sangat populer digunakan karena kemudahannya dalam integrasi dengan mikrokontroler seperti ESP32, Arduino, maupun Raspberry Pi, serta dukungannya untuk analisis data berbasis MATLAB.

Dalam penelitian ini, ThingSpeak dimanfaatkan sebagai media pemantauan kecepatan kendaraan secara real-time. Semua data kecepatan yang dibaca oleh sensor mmWave C4001 dan diproses oleh mikrokontroler ESP32 akan dikirimkan ke server ThingSpeak melalui koneksi Wi-Fi. Data yang tersimpan di ThingSpeak dapat dilihat secara langsung melalui web browser maupun aplikasi smartphone.

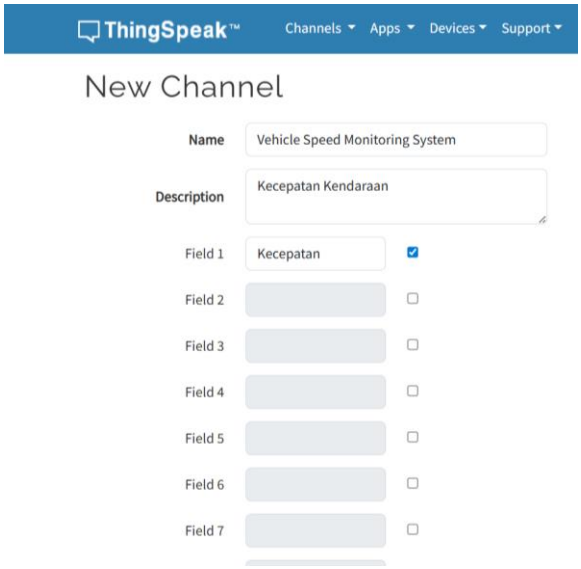
Keunggulan ThingSpeak terletak pada kemampuannya menampilkan data dalam bentuk grafik visual (line chart) yang mudah dipahami. Pengguna juga dapat mengatur rentang waktu data yang ingin ditampilkan, seperti per jam, per hari, bahkan per minggu. Hal ini sangat berguna untuk memantau pola kecepatan kendaraan dalam jangka waktu tertentu. Selain itu, ThingSpeak menyediakan fitur API (Application Programming Interface) yang memungkinkan komunikasi data dua arah. Namun, pada penelitian ini fitur API hanya digunakan untuk pengiriman data (upload) dari ESP32 ke ThingSpeak secara berkala.



Gambar 4. 10 Akun ThingSpeak

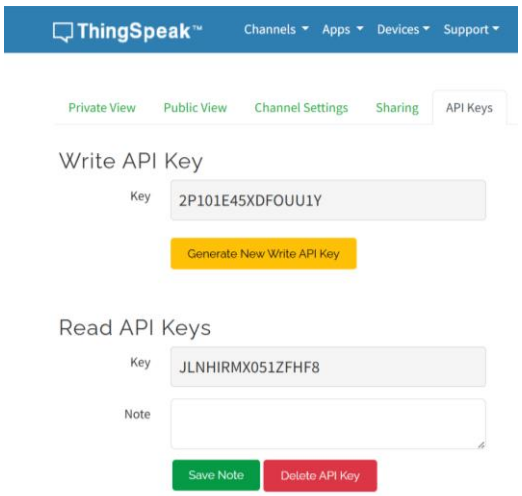
Tahap pertama untuk menggunakan ThingSpeak adalah membuat akun pada situs resmi thingspeak.com. Setelah proses registrasi selesai, dilakukan pembuatan channel baru sebagai wadah untuk menyimpan data dari sensor.

Dalam pembuatan channel ini, peneliti memberikan nama channel yang sesuai dengan tujuan penelitian, misalnya Vehicle Speed Monitoring System. Selanjutnya, mengaktifkan Field 1 yang digunakan khusus untuk menyimpan nilai kecepatan kendaraan. Jika dalam penelitian ini digunakan lebih dari satu parameter, maka dapat mengaktifkan Field tambahan, namun pada proyek ini cukup menggunakan satu field saja.



Gambar 4. 11 Channel ThingSpeak

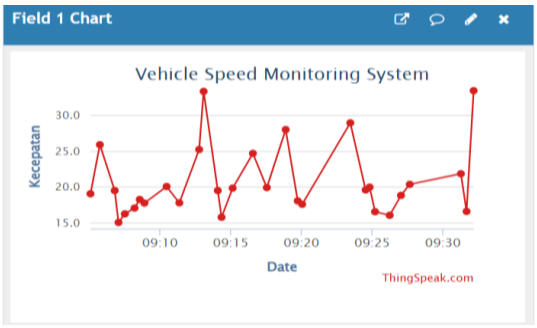
Setelah channel berhasil dibuat, ThingSpeak menyediakan API Key yang berfungsi sebagai kode autentikasi. API Key ini wajib dimasukkan ke dalam program ESP32 agar setiap pengiriman data ke server ThingSpeak dapat diverifikasi dan diterima. Tanpa API Key yang benar, ThingSpeak tidak akan menerima data dari perangkat.



Gambar 4. 12 API Key ThingSpeak

Langkah berikutnya adalah penghubungan ThingSpeak dengan ESP32 dilakukan melalui pemrograman di Arduino IDE. Pada bagian awal program, dilakukan konfigurasi koneksi Wi-Fi dengan memasukkan SSID dan password

jaringan internet yang digunakan. Kemudian, dibuat struktur URL pengiriman data yang memuat API Key beserta nilai pembacaan sensor yang sudah diproses. Nilai kecepatan yang diperoleh dari sensor mmWave C4001 awalnya dalam satuan meter per detik (m/s), lalu dikonversi menjadi kilometer per jam (km/h) sebelum dikirimkan ke ThingSpeak. Pengiriman data dilakukan menggunakan metode HTTP GET yang disediakan oleh library HTTPClient.h. Interval waktu pengiriman data diatur dalam program agar tidak terlalu sering sehingga server ThingSpeak dapat memproses data dengan baik dan tidak menolak pengiriman.



Gambar 4. 13 Grafik Data ThingSpeak

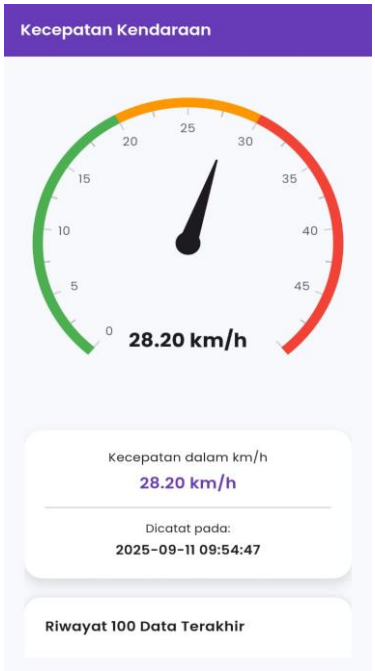
Setelah data berhasil dikirim, ThingSpeak akan secara otomatis menyimpannya di channel yang telah dibuat. Data ini kemudian divisualisasikan dalam bentuk grafik pada halaman channel, di mana setiap titik pada grafik merepresentasikan kecepatan kendaraan pada waktu tertentu. Pengguna dapat memantau data ini secara real-time dari mana saja, baik melalui web browser maupun aplikasi mobile ThingSpeak. Dengan adanya fitur pengaturan rentang waktu tampilan data, pengguna dapat memilih untuk melihat data terbaru atau menelusuri riwayat data untuk keperluan analisis.

Dengan memanfaatkan ThingSpeak, sistem deteksi kecepatan kendaraan yang dibangun menjadi lebih fleksibel karena data dapat dipantau dari jarak jauh tanpa memerlukan koneksi langsung ke perangkat. Hal ini sangat membantu dalam proses analisis performa sistem, pengujian di lapangan, serta pemantauan kondisi lalu lintas secara real-time. ThingSpeak berperan sebagai penghubung antara perangkat keras di lapangan dengan antarmuka visual yang dapat diakses secara

global, sehingga menjadikan sistem ini efektif, portabel, dan praktis digunakan di berbagai lokasi.

4.2.4 Perangkat Lunak Platform Aplikasi

Aplikasi smartphone yang digunakan pada penelitian ini berfungsi sebagai antarmuka pemantauan data kecepatan kendaraan secara real-time yang dikirimkan langsung dari mikrokontroler ESP32 melalui koneksi internet. Aplikasi ini dirancang dengan tampilan antarmuka yang sederhana namun informatif, sehingga memudahkan pengguna untuk membaca dan memahami data yang ditampilkan.



Gambar 4. 14 Platform Aplikasi

Tampilan utama aplikasi memperlihatkan indikator berbentuk speedometer digital dengan jarum penunjuk dan skala kecepatan mulai dari 0 hingga lebih dari 45 km/h. Warna skala dibagi menjadi tiga zona, yaitu zona hijau untuk kecepatan rendah, zona kuning untuk kecepatan normal, dan zona merah untuk kecepatan tinggi. Tepat di tengah speedometer, terdapat angka besar yang menunjukkan nilai kecepatan terkini dalam satuan kilometer per jam (km/h).

Selain indikator kecepatan, aplikasi juga menampilkan keterangan “Kecepatan dalam km/h” diikuti oleh nilai kecepatan terbaru, serta waktu

pencatatan data yang menunjukkan kapan data tersebut terakhir diterima. Informasi ini memudahkan pengguna untuk mengetahui apakah data yang tampil adalah pembacaan terbaru dari sensor atau data lama.

Bagian bawah tampilan aplikasi memuat riwayat pembacaan kecepatan terakhir yang disajikan dalam bentuk daftar. Pada penelitian ini, riwayat tersebut menampilkan hingga 100 data kecepatan terakhir yang telah diterima oleh aplikasi. Fitur ini berguna untuk melihat pola perubahan kecepatan kendaraan selama periode tertentu tanpa harus membuka platform pemantauan lain seperti ThingSpeak.

Penghubungan aplikasi smartphone dengan ESP32 dilakukan melalui pemrograman di Arduino IDE. Pada bagian awal program, dilakukan konfigurasi koneksi Wi-Fi dengan memasukkan SSID dan password jaringan internet yang digunakan. Selanjutnya, ESP32 diprogram untuk mengirimkan data kecepatan yang diperoleh dari sensor mmWave C4001 secara langsung ke alamat server atau endpoint yang digunakan oleh aplikasi smartphone.

Nilai kecepatan yang diperoleh dari sensor awalnya berada dalam satuan meter per detik (m/s), kemudian dikonversi menjadi kilometer per jam (km/h) di dalam program sebelum dikirimkan. Proses pengiriman data ini memanfaatkan protokol HTTP atau WebSocket agar data dapat diterima aplikasi dengan cepat. Aplikasi smartphone kemudian menampilkan data tersebut dalam bentuk indikator speedometer digital serta riwayat pembacaan terakhir.

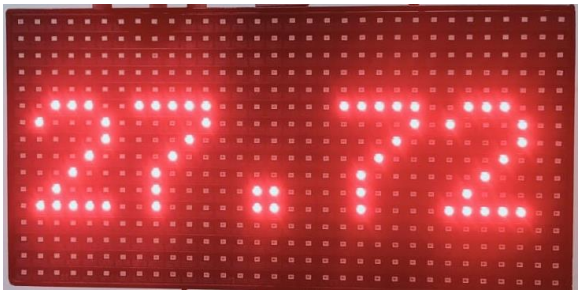
Interval waktu pengiriman data diatur di dalam program agar aplikasi selalu menerima pembaruan data secara real-time namun tetap menjaga kestabilan koneksi dan mencegah beban jaringan yang berlebihan. Dengan metode ini, pengguna dapat memantau kecepatan kendaraan langsung dari ponsel tanpa melalui perantara platform cloud seperti ThingSpeak, sehingga data dapat diterima lebih cepat.

Dengan adanya aplikasi ini, proses monitoring menjadi lebih cepat karena pengguna dapat memeriksa data kecepatan kendaraan secara langsung melalui

ponsel tanpa memerlukan perangkat tambahan. Data ditampilkan secara real-time dan terhubung langsung dengan sistem, sehingga apabila terdapat perubahan kecepatan pada kendaraan yang terdeteksi oleh sensor mmWave C4001 dan informasi tersebut dapat segera diketahui oleh pengguna melalui aplikasi.

4.2.5 Perangkat Lunak Pengendali Dot Matrix

Pada penelitian ini, LED Dot Matrix digunakan sebagai media tampilan kecepatan kendaraan secara langsung di lokasi alat. Data yang ditampilkan berasal dari hasil pembacaan sensor mmWave C4001 yang diolah oleh mikrokontroler ESP32, kemudian dikirim ke Arduino Nano untuk dikendalikan dan ditampilkan di LED Dot Matrix.



Gambar 4. 15 Tampilan LED Dot Matrix P10

LED Dot Matrix memiliki berbagai keunggulan yang mendukung penerapan sistem ini. Salah satunya adalah kemampuan menampilkan informasi dengan ukuran besar dan keterbacaan tinggi sehingga mudah dilihat oleh pengguna meskipun dari jarak jauh. Modul ini juga hemat daya, tahan lama, dan mampu menampilkan teks atau angka secara real-time. Pada sistem ini, LED Dot Matrix dimanfaatkan untuk memberikan informasi kecepatan kendaraan dalam satuan kilometer per jam (km/h) dengan tampilan numerik yang besar dan jelas.

Langkah awal pemanfaatan LED Dot Matrix dalam penelitian ini adalah menentukan jenis modul yang sesuai dan menghubungkannya dengan Arduino Nano sebagai pengendali. Arduino Nano dipilih karena memiliki kinerja yang stabil dalam mengendalikan tampilan LED Dot Matrix, terutama ketika menggunakan library DMD.h yang dirancang khusus untuk mengontrol panel berbasis HUB75. Penggunaan library ini mempermudah pengaturan teks, posisi

tampilan, dan pembaruan data tanpa perlu membuat kode kendali LED secara manual.

Program pada Arduino Nano dibuat di Arduino IDE dan berfungsi untuk menerima data kecepatan kendaraan yang dikirim oleh ESP32 melalui komunikasi serial. Data tersebut kemudian diproses dan ditampilkan dalam format angka pada LED Dot Matrix. Untuk menjaga kualitas tampilan, program memanfaatkan fungsi refresh display dari library DMD.h sehingga angka yang ditampilkan tetap stabil tanpa flicker (berkedip).

Setelah sistem berjalan, setiap data kecepatan kendaraan yang diterima oleh Arduino Nano dari ESP32 akan langsung diperbarui di LED Dot Matrix secara real-time. Dengan demikian, orang yang berada di sekitar alat dapat langsung melihat kecepatan kendaraan yang melintas tanpa harus mengakses platform pemantauan jarak jauh seperti ThingSpeak atau aplikasi smartphone.

Dengan memanfaatkan LED Dot Matrix sebagai media tampilan lokal, sistem deteksi kecepatan kendaraan yang dibangun menjadi lebih informatif dan efektif. Informasi dapat tersampaikan secara cepat kepada pihak yang memerlukannya, baik itu pengguna di lapangan, petugas, atau masyarakat umum. Hal ini menjadikan LED Dot Matrix sebagai komponen penting dalam sistem ini karena berperan langsung dalam memberikan informasi visual yang mudah diakses dan dipahami.