

BAB 3

METODOLOGI PENELITIAN

3.1 Jenis dan Pendekatan Penelitian

Penelitian ini diklasifikasikan sebagai penelitian terapan (*applied research*) dengan fokus pada pengembangan sistem (*system development research*). Penelitian terapan bertujuan untuk menghasilkan solusi teknis yang dapat diterapkan secara langsung dalam konteks permasalahan nyata melalui proses perancangan dan implementasi sistem [24].

Pendekatan metodologis yang digunakan adalah pendekatan rekayasa sistem (*systems engineering approach*). Pendekatan ini menekankan proses pengembangan sistem secara bertahap dan terstruktur, yang mencakup perancangan arsitektur, implementasi komponen, integrasi subsistem, serta pengujian performa. Pendekatan rekayasa sistem dipilih karena selaras dengan tujuan penelitian, yaitu membangun sistem yang mampu melakukan akuisisi, transmisi, penyimpanan, dan visualisasi data kendaraan secara terintegrasi.

Dari sudut pandang paradigma penelitian, penelitian ini berlandaskan paradigma postpositivistik. Paradigma ini memandang bahwa fenomena dapat diamati, diukur, dan dianalisis secara objektif melalui data empiris serta pengujian sistematis [25]. Oleh karena itu, evaluasi penelitian difokuskan pada pengujian kinerja sistem, keandalan komunikasi data, serta kemampuan sistem dalam menyajikan informasi kendaraan secara *real-time*.

Selain itu, kerangka metodologis penelitian ini juga merujuk pada prinsip *Design Science Research* (DSR), yang umum digunakan dalam penelitian pengembangan sistem informasi dan teknologi [26]. Dalam kerangka ini, penelitian tidak hanya berfokus pada proses pengembangan sistem, tetapi juga pada evaluasi hasil implementasi untuk memastikan bahwa sistem yang dibangun memiliki fungsi yang sesuai dan relevan dengan kebutuhan pengguna.

Dengan demikian, metodologi penelitian ini merupakan kombinasi antara penelitian terapan, pendekatan rekayasa sistem, paradigma postpositivistik, serta kerangka DSR, yang secara keseluruhan mendukung proses pengembangan sistem monitoring kendaraan berbasis data secara sistematis dan terukur.

3.2 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di lingkungan Laboratorium Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, serta pada kendaraan uji (*test vehicle*) untuk keperluan pengambilan data lapangan. Aktivitas penelitian di laboratorium mencakup pengembangan dan pengujian sistem, sedangkan pengujian lapangan dilakukan untuk memperoleh data komunikasi kendaraan secara langsung melalui jaringan *Controller Area Network* (CAN Bus) menggunakan port *On-Board Diagnostics II* (OBD-II).

Pengembangan perangkat lunak dilakukan pada lingkungan lokal yang mencakup layanan *backend* dan *frontend* berbasis web. Lingkungan ini digunakan untuk mendukung proses integrasi data, penyimpanan, serta visualisasi informasi kendaraan secara *real-time*.

Penelitian direncanakan berlangsung selama enam bulan, yaitu dari Juli hingga Desember 2025, dengan tahapan sebagai berikut:

- Juli 2025: Penyusunan rencana penelitian dan studi literatur.
- Agustus 2025: Perancangan sistem perangkat keras dan perangkat lunak.
- September 2025: Pengumpulan data CAN Bus melalui port OBD-II pada kendaraan uji.
- Oktober 2025: Implementasi dan integrasi sistem perangkat keras dan perangkat lunak.
- November 2025: Pengembangan dan integrasi modul analisis berbasis data.
- Desember 2025: Pengujian, evaluasi, dan optimasi sistem.

3.3 Prosedur Penelitian

Prosedur penelitian disusun untuk menggambarkan tahapan pelaksanaan penelitian secara menyeluruh dan sistematis. Setiap tahapan dirancang agar saling berkaitan dan mendukung pencapaian tujuan penelitian.

Secara umum, prosedur penelitian ini terdiri atas lima tahapan utama sebagai berikut:

1. Studi Literatur

Tahap awal dilakukan dengan mengkaji literatur yang relevan terkait jaringan

CAN Bus, sistem monitoring kendaraan, serta teknologi pendukung yang digunakan dalam penelitian. Studi ini berfungsi sebagai landasan teoritis dan referensi dalam perancangan sistem.

2. Perancangan Sistem

Berdasarkan hasil studi literatur, dilakukan perancangan sistem yang mencakup arsitektur perangkat keras, alur komunikasi data, serta rancangan perangkat lunak dan antarmuka visualisasi.

3. Implementasi dan Integrasi Sistem

Tahap ini mencakup perakitan perangkat keras, konfigurasi modul CAN Bus, serta pengembangan perangkat lunak. Seluruh komponen kemudian diintegrasikan untuk memastikan sistem dapat bekerja secara terpadu.

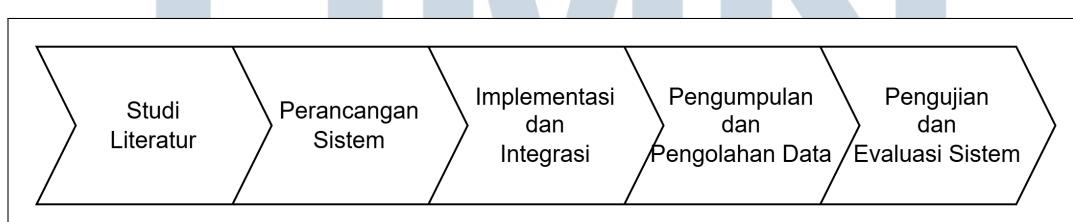
4. Pengumpulan dan Pengolahan Data

Sistem yang telah terintegrasi digunakan untuk mengambil data dari kendaraan uji melalui jaringan CAN Bus. Data yang diperoleh kemudian disimpan dan diolah untuk kebutuhan monitoring dan evaluasi.

5. Pengujian dan Evaluasi Sistem

Tahap akhir meliputi pengujian fungsional dan performa untuk memastikan sistem bekerja sesuai dengan rancangan dan mampu menampilkan data kendaraan secara *real-time*.

Alur keseluruhan prosedur penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1. Diagram alur penelitian

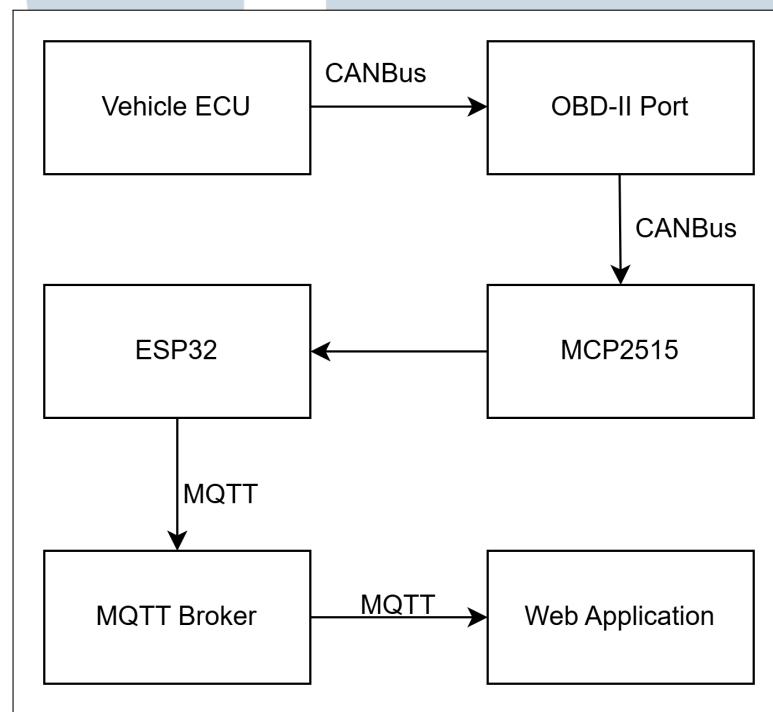
3.4 Perancangan Sistem

Perancangan sistem merupakan tahapan penting dalam penelitian ini karena menjadi dasar implementasi sistem monitoring kendaraan yang dikembangkan. Pada tahap ini, rancangan disusun untuk memastikan bahwa sistem mampu melakukan akuisisi data kendaraan, mengirimkan data secara *real-time*, serta

menampilkan informasi kendaraan dalam bentuk yang mudah dipahami oleh pengguna. Perancangan sistem mencakup perancangan arsitektur umum, perangkat keras, dan perangkat lunak yang saling terintegrasi.

3.4.1 Arsitektur Umum Sistem

Arsitektur sistem pada penelitian ini dirancang untuk mendukung proses pengambilan, pengiriman, penyimpanan, serta penyajian data kendaraan secara otomatis dan terintegrasi. Struktur arsitektur ini menghubungkan sistem elektronik kendaraan dengan perangkat *embedded* serta aplikasi berbasis web untuk mendukung proses monitoring kendaraan secara *real-time*.



Gambar 3.2. Diagram arsitektur umum sistem

Sebagaimana ditunjukkan pada Gambar 3.2, data kendaraan bersumber dari berbagai *Electronic Control Unit* (ECU) yang saling berkomunikasi melalui jaringan *Controller Area Network* (CAN Bus). Data tersebut dapat diakses melalui port *On-Board Diagnostics II* (OBD-II) yang tersedia pada kendaraan. Port ini berfungsi sebagai titik akses untuk membaca komunikasi data antar ECU tanpa mengganggu sistem kendaraan.

Data dari OBD-II diteruskan ke modul MCP2515 yang berfungsi sebagai *CAN Controller* sekaligus *CAN Transceiver*. Modul ini bertugas menangani proses

komunikasi CAN, termasuk penerimaan sinyal diferensial CAN-H dan CAN-L serta pengolahan frame CAN agar dapat dibaca oleh mikrokontroler ESP32.

Mikrokontroler ESP32 berperan sebagai node akuisisi data yang membaca frame CAN dari MCP2515, kemudian menyiarkan (*broadcast*) data tersebut melalui protokol *Message Queuing Telemetry Transport* (MQTT). Penggunaan MQTT dipilih karena karakteristiknya yang ringan, efisien, dan sesuai untuk sistem *Internet of Things* (IoT) yang membutuhkan komunikasi data *real-time* dengan latensi rendah.

Aplikasi web berfungsi sebagai klien yang berlangganan (*subscribe*) pada topik MQTT tertentu. Dengan mekanisme ini, aplikasi web dapat menerima data kendaraan secara langsung dan menampilkannya dalam bentuk visualisasi *real-time*. Selain itu, data yang diterima juga disimpan ke dalam basis data MongoDB untuk kebutuhan pencatatan dan analisis lanjut.

Secara umum, arsitektur sistem dalam penelitian ini dibagi ke dalam tiga lapisan fungsional utama sebagai berikut:

1. Lapisan Akuisisi Data

Lapisan ini berfungsi untuk melakukan pengambilan data kendaraan secara langsung dari ECU melalui jaringan *Controller Area Network* (CAN Bus).

Komponen yang terlibat pada lapisan ini meliputi ECU kendaraan, jaringan CAN Bus, modul MCP2515, serta mikrokontroler ESP32 yang bertugas membaca dan menyiarkan data kendaraan.

2. Lapisan Komunikasi

Lapisan ini berfungsi sebagai media pertukaran dan distribusi data dari perangkat *embedded* ke sistem aplikasi. Proses komunikasi data pada lapisan ini dilakukan menggunakan protokol *Message Queuing Telemetry Transport* (MQTT) dengan bantuan *MQTT Broker* sebagai perantara.

3. Lapisan Aplikasi

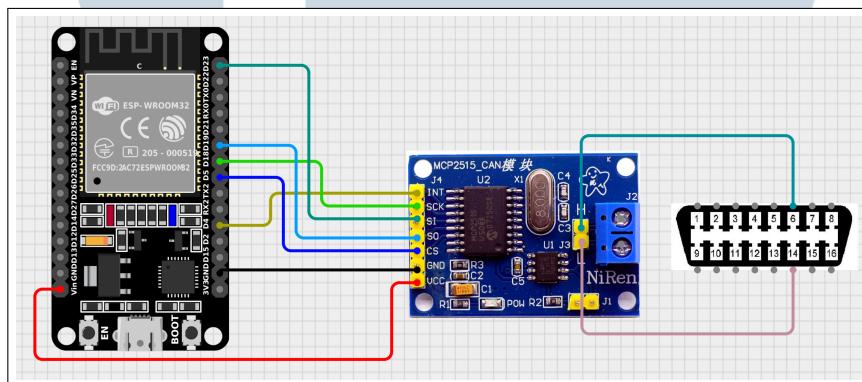
Lapisan ini berfungsi untuk menerima, menyimpan, dan menyajikan data kendaraan kepada pengguna. Lapisan aplikasi mencakup aplikasi web sebagai antarmuka visualisasi serta basis data MongoDB yang digunakan untuk penyimpanan dan pengelolaan data kendaraan.

Arsitektur sistem dirancang secara modular agar setiap komponen dapat dikembangkan atau dimodifikasi secara terpisah tanpa memengaruhi keseluruhan sistem.

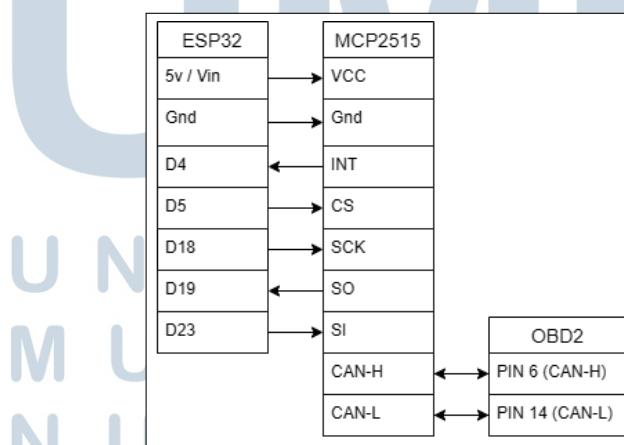
3.4.2 Perancangan Perangkat Keras

Perancangan perangkat keras bertujuan untuk menyediakan sistem akuisisi data yang mampu membaca komunikasi jaringan CAN Bus dari kendaraan serta mengirimkan data tersebut ke sistem aplikasi secara nirkabel. Rancangan perangkat keras difokuskan pada pemilihan komponen yang kompatibel, stabil, dan mudah diintegrasikan.

Komponen utama yang digunakan dalam rancangan ini meliputi mikrokontroler ESP32, modul CAN MCP2515, dan konektor OBD-II sebagai antarmuka ke jaringan kendaraan. ESP32 dipilih karena memiliki kemampuan komunikasi Wi-Fi terintegrasi serta mendukung antarmuka SPI yang dibutuhkan untuk berkomunikasi dengan modul MCP2515.



Gambar 3.3. Rancangan fisik koneksi perangkat keras ESP32, MCP2515, dan OBD-II



Gambar 3.4. Diagram koneksi pin antara ESP32, MCP2515, dan OBD-II

Koneksi antara ESP32 dan MCP2515 dilakukan melalui antarmuka SPI. Konfigurasi pin yang digunakan ditunjukkan pada Tabel 3.1.

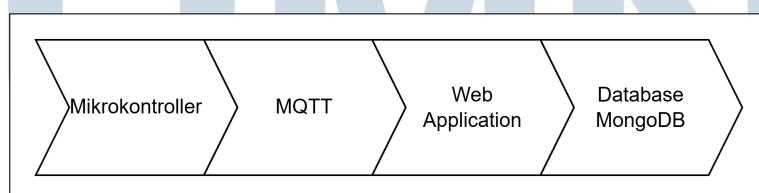
Tabel 3.1. Koneksi pin antara ESP32 dan MCP2515

ESP32	MCP2515
5V / Vin	VCC
GND	GND
D4	INT
D5	CS
D18	SCK
D19	SO
D23	SI

Modul MCP2515 dihubungkan ke kendaraan melalui konektor OBD-II menggunakan dua jalur utama, yaitu CAN-H (Pin 6) dan CAN-L (Pin 14). Kedua jalur ini membawa sinyal diferensial komunikasi CAN yang memungkinkan pembacaan data antar ECU secara langsung.

3.4.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak difokuskan pada pengembangan sistem yang mampu menerima, menyimpan, dan menampilkan data kendaraan yang diperoleh dari jaringan CAN Bus. Perangkat lunak dirancang untuk berkomunikasi dengan perangkat keras menggunakan protokol MQTT dan mendukung proses pencatatan data secara *real-time*.



Gambar 3.5. Diagram alur perancangan perangkat lunak sistem monitoring kendaraan

Sistem perangkat lunak terdiri atas beberapa komponen utama, yaitu mikrokontroler ESP32, MQTT Broker, aplikasi web, dan basis data MongoDB. ESP32 bertugas membaca data CAN dan menyiarannya dalam format JSON melalui MQTT. MQTT Broker berperan sebagai penyalur data yang memungkinkan banyak klien menerima data secara bersamaan.

Aplikasi web berfungsi sebagai antarmuka pengguna yang menampilkan data kendaraan secara *real-time* serta menyimpan data ke dalam basis data

MongoDB melalui layanan HTTP. Basis data MongoDB digunakan karena memiliki struktur yang fleksibel dan sesuai untuk penyimpanan data telemetri kendaraan.

3.5 Pengumpulan Data

Pengumpulan data dilakukan secara langsung pada kendaraan uji untuk memperoleh data komunikasi dari jaringan CAN Bus. Kendaraan yang digunakan dalam penelitian ini adalah Honda BR-V tahun 2017. Pemilihan kendaraan tersebut didasarkan pada ketersediaan dan kompatibilitasnya dengan standar komunikasi CAN dan OBD-II.

Pengambilan data dilakukan menggunakan dua konfigurasi perangkat, yaitu Arduino Mega 2560 dengan modul MCP2515 atau TJA1050, serta ESP32 dengan modul MCP2515. Data yang diperoleh dari mikrokontroler direkam melalui antarmuka USB dan disimpan menggunakan *script* Python dalam format CSV.

Setelah data diperoleh dalam format CSV, dilakukan proses identifikasi sinyal CAN untuk menentukan parameter kendaraan yang relevan, seperti kecepatan kendaraan, putaran mesin (RPM), dan suhu mesin. Proses ini menjadi dasar untuk pengolahan data pada tahap berikutnya.

3.6 Pengolahan dan Analisis Data

Pengolahan dan analisis data dilakukan untuk mengubah data mentah hasil pembacaan CAN Bus menjadi informasi yang dapat digunakan dalam sistem monitoring kendaraan. Proses ini mencakup pengolahan data awal, pemrosesan data secara *real-time* pada sistem *embedded*, serta analisis data historis.

Tahap awal dimulai dengan pengolahan data CSV menggunakan *Jupyter Notebook* untuk mengidentifikasi CAN ID dan menentukan rumus konversi nilai mentah menjadi parameter teknis. Selanjutnya, ESP32 menerapkan hasil konversi tersebut secara *real-time* sebelum menyiarkan data ke aplikasi web.

Data yang diterima oleh aplikasi web disimpan ke MongoDB dan digunakan untuk evaluasi operasional kendaraan, analisis tren penggunaan, serta pengembangan fitur monitoring lanjutan.

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen sistem bekerja sesuai dengan rancangan dan mampu beroperasi secara stabil dalam kondisi nyata. Pengujian difokuskan pada pengujian fungsional dan pengujian performa sistem.

3.7.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk memverifikasi bahwa sistem mampu membaca data CAN, mengirimkan data melalui MQTT, serta menampilkan data pada aplikasi web tanpa kesalahan komunikasi.

3.7.2 Pengujian Performa Sistem

Pengujian performa bertujuan mengukur latensi transmisi data, kecepatan pemrosesan, serta stabilitas sistem dalam menangani data *real-time*. Evaluasi ini memastikan sistem layak digunakan sebagai sistem monitoring kendaraan.

