SIMULASI SISTEM *DATA LOGGING* BERBASIS *INTERNET OF THINGS* UNTUK MELACAK AKTIVITAS KENDARAAN DENGAN APLIKASI ANDROID

Muhammad Fadhli Adam, Purwantoro, Dadang Yusup
Informatika, Universitas Singaperbangsa Karawang
Jl. HS. Ronggowaluyo, Telukjambe Timur, Kabupaten Karawang
2010631170095@student.unsika.ac.id

ABSTRAK

Dalam industri logistik modern, kendaraan tidak hanya berfungsi sebagai alat transportasi, tetapi juga sebagai komponen penting dalam *supply chain* yang harus dipantau secara ketat untuk menjaga efisiensi dan keandalan operasional. Pemantauan aktivitas kendaraan menjadi sangat penting, mengingat berbagai permasalahan yang sering muncul, seperti perilaku pengemudi yang tidak sesuai standar serta kurangnya sistem pelacakan kendaraan berbasis Internet of Things (IoT) yang terintegrasi secara real-time dengan aplikasi Android. Penelitian ini bertujuan untuk mengembangkan sistem pelacakan aktivitas kendaraan berbasis IoT yang terintegrasi dengan aplikasi Android menggunakan Firebase Realtime Database. Sistem ini dirancang dengan menggunakan metode penelitian yang terdiri dari empat tahap: analisis sistem, perancangan sistem, pembuatan program, dan pengujian sistem. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu mlakukan transfer data aktivitas kendaraan secara real-time dan memantau posisi, jarak tempuh, kecepatan, serta durasi perjalanan dengan akurat melalui aplikasi Android. Implementasi sistem ini diharapkan dapat mengurangi perilaku buruk pengemudi dan meningkatkan efisiensi operasional dalam sektor logistik.

Kata kunci : *Internet of Things, Android, Node-Red, GPS, Pelacakan Kendaraan*

1. PENDAHULUAN

Dalam industri logistik, pemantauan aktivitas kendaraan menjadi sangat penting. Namun, seringkali terjadi permasalahan seperti perilaku pengemudi yang tidak sesuai standar, seperti ugal-ugalan atau berhenti terlalu lama, yang dapat mengganggu estimasi waktu pengiriman dan menurunkan efisiensi operasional [1].

Kurangnya sistem pelacakan kendaraan berbasis IoT yang terintegrasi secara *real-time* dengan aplikasi Android juga menjadi kendala dalam pemantauan yang efektif.

Oleh karena itu, penelitian ini memiliki tujuan untuk memberikan solusi dengan mengintegrasikan IoT dengan aplikasi Android menggunakan Firebase Realtime Database yang memungkinkan pelacakan aktivitas kendaraan secara *real-time* dan akurat [2].

Mikrokontroler ESP32, dengan dukungan WiFi dan Bluetooth, menjadi pilihan ideal karena fleksibilitas dan daya tahannya [3].

Sistem ini menggunakan modul GPS untuk memperoleh data lokasi kendaraan, yang kemudian dikirimkan melalui jaringan internet untuk pemantauan akurat. Node-RED berperan sebagai backend, menghubungkan ESP32 melalui MQTT dengan Firebase Realtime Database, memastikan data dikirim secara efisien. Google Maps API dalam aplikasi Android memungkinkan visualisasi rute perjalanan kendaraan dan analisis pola perjalanan [4].

Simulasi dilakukan sebagai langkah awal untuk memvalidasi desain dan memastikan seluruh fungsinya bekerja sesuai harapan. Simulasi ini dapat membantu mengidentifikasi potensi masalah teknis yang mungkin muncul saat implementasi. Selain itu, simulasi juga memungkinkan tim pengembang untuk mengevaluasi kinerja sistem dalam berbagai skenario, sehingga risiko kesalahan dapat diminimalkan dan biaya tambahan yang tidak diinginkan dapat dihindari [5].

Dengan adanya sistem pelacakan kendaraan berbasis IoT ini, diharapkan efisiensi operasional perusahaan logistik akan meningkat. Sistem ini juga berperan dalam menjaga keamanan kendaraan, karena perilaku mengemudi dapat dipantau dan dikendalikan. Diharapkan sistem ini dapat memberikan kontribusi positif terhadap perkembangan industri logistik, dengan menciptakan lingkungan operasional yang lebih aman, efisien, dan berkelanjutan..

2. TINJAUAN PUSTAKA

2.1. Perhitungan Aktivitas Kendaraan

Pada penelitian ini rumus vicenty digunakan untuk menghitung jarak tempuh berdasarkan titik koordinat yang didapat dari GPS. Rumus Vincenty merupakan metode matematika yang dapat digunakan untuk menghitung azimuth geodetik dengan lebih akurat daripada menggunakan konsep segitiga bola. Metode ini merupakan teknik perhitungan yang lebih cermat dalam menentukan arah dan jarak antara dua titik geografis di permukaan bumi [6].

Di dalam pustaka internal Android, kita bisa menemukan fungsi bernama Location.directionBetween(). Fungsi ini bisa dipakai dalam bahasa pemrograman kotlin dan java untuk mengestimasi jarak antara dua titik koordinat.

Kemudian untuk menghitung durasi berkendara, data waktu dikirim bersamaan dengan data lokasi ke *database*. Data waktu berbentuk *timestamp* dengan format “yyyy-M-d H:mm:ss” yang dimana format tersebut harus dikonversikan terlebih dahulu ke satuan menit untuk mendapatkan durasi berkendara.

Selanjutnya, jika jarak tempuh dan durasi berkendara telah didapat, maka perhitungan kecepatan kendaraan dapat dilakukan. Kecepatan kendaraan pada sistem ini menggunakan satuan kilometer per jam sehingga durasi harus diubah terlebih dahulu dari satuan menit ke satuan jam. Jika sudah, maka kecepatan kendaraan dihitung dengan pembagian jarak (km) dan durasi (jam).

2.2. *Internet of Things* (IoT)

Konsep dari *Internet of Things* (IoT) pertama kali diperkenalkan oleh Kevin Asthon pada tahun 1999. Dalam presentasinya, ia menekankan bahwa sebagian besar data di internet dimasukkan oleh manusia ke dalam sistem, namun seringkali terdapat kesalahan dalam kualitas dan kuantitas data tersebut. Sebagai solusi, sistem ini akan lebih efektif jika dapat terhubung langsung ke sensor yang terkoneksi ke internet untuk mengambil data. IoT adalah teknologi modern yang memanfaatkan konektivitas internet yang selalu aktif dengan menghubungkan berbagai perangkat ke jaringan internet melalui sensor dan selalu aktif. Konsep ini memanfaatkan konektivitas internet yang selalu tersambung secara luas, yang merupakan inti dari IoT [7].

2.3. *Message Queue Telemetry Transport* (MQTT)

Message Queue Telemetry Transport, yang dikenal sebagai MQTT, adalah protokol komunikasi M2M yang beroperasi pada tingkat aplikasi dan mengirim pesan ringan. Protokol MQTT menjamin pengiriman semua pesan, bahkan ketika koneksi terganggu. MQTT menggunakan metode pengiriman publish/subscribe. Pesan dalam MQTT dikirim ke broker dan mencakup topik yang dipublikasikan oleh publisher. Topik tersebut kemudian diproses dan diteruskan ke subscriber sesuai dengan permintaan pengguna [8].

2.4. Node-Red

Node-RED adalah alat berbasis browser yang dirancang untuk pengembangan aplikasi *Internet of Things* (IoT). Dengan lingkungan pemrograman visualnya, pengguna dapat dengan mudah membuat aplikasi dalam bentuk "flow". Sebagai bahasa pemrograman visual, Node-RED tidak berfokus pada pembuatan aplikasi sebagai urutan kode, tetapi lebih pada alur program [8].

2.5. Wokwi

Wokwi merupakan platform simulator untuk *Electronics Development Board* seperti Arduino, ESP32, dan Raspberry Pi Pico. Keunggulan Wokwi dibandingkan platform simulasi lainnya adalah kelengkapan komponen pendukung seperti sensor, tombol, dan LED yang lebih lengkap. Selain itu, simulasi di Wokwi dapat terhubung ke server asli. Wokwi juga menyediakan beragam pilihan board, termasuk Arduino, ESP32, dan Raspberry Pi Pico. Antarmuka Wokwi hampir mirip dengan perangkat lunak pemrograman sebenarnya, termasuk fitur untuk menambahkan library [9].

2.6. Android

Android merupakan sistem operasi seluler yang berbasis pada kernel Linux yang telah dimodifikasi, serta memanfaatkan perangkat lunak sumber terbuka lainnya. Sistem ini dirancang terutama untuk perangkat seluler berlayar sentuh, seperti smartphone dan tablet. Android pertama kali diperkenalkan pada September 2008, dikembangkan oleh Open Handset Alliance dengan dukungan komersial dari Google [10].

2.7. Google Maps API

Google Maps API merupakan fungsi-fungsi pemrograman yang disediakan oleh Google untuk mengintegrasikan Google maps ke dalam Web atau aplikasi yang sedang dibuat [4]. Pada sistem ini, Google Maps Api digunakan untuk menampilkan peta rute bersepeda berdasarkan list longitude dan latitude yang terdapat pada basis data.

2.8. Firebase

Firebase adalah solusi komprehensif dari Google yang memungkinkan developer membangun aplikasi mobile dan web dengan cepat. Fitur unggulannya adalah *database real-time* yang memungkinkan sinkronisasi data secara instan antar perangkat [2]. Salah satu kelebihan yang dimiliki oleh Firebase adalah, *database* ini menyimpan data secara lokal ketika suatu perangkat tidak terhubung dengan akses internet [11].

2.9. ESP32

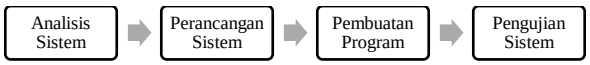
ESP32 merupakan mikrokontroler yang dikembangkan oleh Espressif System yang merupakan pengembangan dari mikrokontroler ESP8266. Secara spesifikasi ESP32 sangat lengkap sehingga mikrokontroler ini sangat tepat jika kita gunakan untuk pengaplikasian dari sistem yang menggunakan konsep *Internet of Things* (IoT), sebab mikrokontroler ini dapat berkomunikasi melalui jaringan WiFi, BLE, dan Bluetooth [3].

2.10. *Global Positioning System* (GPS)

GPS (*Global Positioning System*) adalah teknologi navigasi berbasis satelit yang memungkinkan pengguna mengetahui lokasi mereka di permukaan bumi. Sebagai satu-satunya sistem satelit navigasi global yang sepenuhnya operasional, GPS menyediakan informasi yang akurat tentang lokasi, waktu, arah, dan kecepatan. Sistem ini pertama kali dikembangkan oleh Departemen Pertahanan Amerika Serikat dan dirancang untuk memenuhi kebutuhan militer, namun kini juga digunakan untuk tujuan sipil seperti survei dan pemetaan [12].

3. METODE PENELITIAN

Perancangan sistem *data logging* berbasis *Internet of Things* ini menggunakan metode penelitian dengan kerangka kerja yang terdiri dari 4 tahap sebagai berikut:



Gambar 1. Metode penelitian

a. Analisis Sistem, pada tahap ini dilakukan analisis untuk menentukan kebutuhan-kebutuhan apa saja yang diperlukan dalam membangun sistem ini sesuai dengan tujuan penelitian.
b. Perancangn Sistem, pada tahap ini dilakukan perancangan sistem input, proses dan output yang akan digunakan dan dihasilkan oleh sistem.
c. Pembuatan Program, pada tahap ini pembuatan program pada *embedded system* dilakukan sesuai dengan perancangan yang telah dibuat pada tahap sebelumnya.
d. Pengujian Sistem pada tahap ini pengujian sistem dilakukan untuk mengetahui bahwa sistem telah sesuai dengan kebutuhan.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Sistem

Pada tahap ini dilakukan analisis untuk mengetahui dan menentukan kebutuhan-kebutuhan yang diperlukan oleh sistem sehingga dapat sesuai dengan spesifikasi dan tujuan dari pembuatan sistem ini.

4.2. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional diperlukan untuk menjelaskan proses-proses yang akan dijalankan oleh sistem, informasi yang dibutuhkan dan dihasilkan, layanan yang harus disediakan, serta respons dan tindakan sistem dalam berbagai situasi. Adapun kebutuhan fungsional pada sistem ini meliputi:

a. *Embedded system* mampu menangkap data GPS melalui modul GPS
b. *Embedded system* mampu mengirim data GPS ke *backend*
c. *Backend* mampu menyimpan data GPS ke *database*
d. Aplikasi Android mampu mengambil data GPS dari *database*
e. Aplikasi android mampu memvisualisasikan aktivitas kendaraan (lokasi, jarak tempuh, durasi dan kecepatan)

4.3. Analisis Kebutuhan Non Fungsional

Selain kebutuhan fungsional, berdasarkan hasil analisis terdapat beberapa kebutuhan non fungsional pada sistem ini. Berikut tabel kebutuhan perangkat keras yang terdapat informasi dari spesifikasi perangkat keras yang dibutuhkan untuk sistem ini:

Tabel 1. Kebutuhan perangkat keras

No	Perangkat Keras	Keterangan
1	Ponsel	• RAM: 2GB atau lebih • Penyimpanan Internal: 8GB atau lebih • Sistem Operasi: Android 8 (Oreo) atau terbaru
2	Komputer	• Sistem Operasi: Windows (8, 10, atau 11), Linux, macOS (10.14 Mojave atau terbaru), ChromeOS • Ram: 4GB atau lebih • Prosesor: Intel Core generasi ke-2 atau yang lebih baru, atau CPU AMD dengan dukungan untuk *Windows Hypervisor* • Penyimpanan minimum yang tersedia: 64 GB
3	*Embedded System*	• Mikrokontroler ESP-32 • Modul GPS

Selain perangkat keras, sistem ini memerlukan perangkat lunak untuk dapat dijalankan dengan baik. Berikut Tabel 2 yang mencakup informasi perangkat lunak yang dibutuhkan untuk sistem ini:

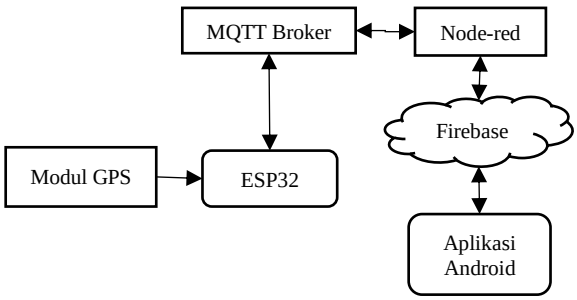
Tabel 2. Kebutuhan perangkat lunak

No	Perangkat lunak	Keterangan
1	IDE	Android Studio IDE, Arduino IDE, dan Node-Red
2	Simulator IoT	Wokwi
3	MQTT Broker	Eclipse Mosquitto
4	*Database*	Firebase Realtime Database

4.4. Analisis Sistem

Pada fase ini, alur sistem yang akan dirancang ditetapkan dalam bentuk diagram blok, rancangan *database*, serta rancangan rangkaian elektronik. Representasi visual ini bertujuan untuk menggambarkan interaksi antara pengguna dan sistem, yang didasarkan pada analisis kebutuhan sebelumnya.

4.5. Diagram Blok



Gambar 2. Diagram blok sistem

Pada Gambar 2 menunjukkan alur kerja sistem *data logging* untuk melacak aktivitas kendaraan berbasis IoT, di mana modul GPS mengirim data GPS ke ESP32, yang kemudian meneruskannya ke MQTT Broker. Data ini diolah oleh Node-RED dan dikirim ke

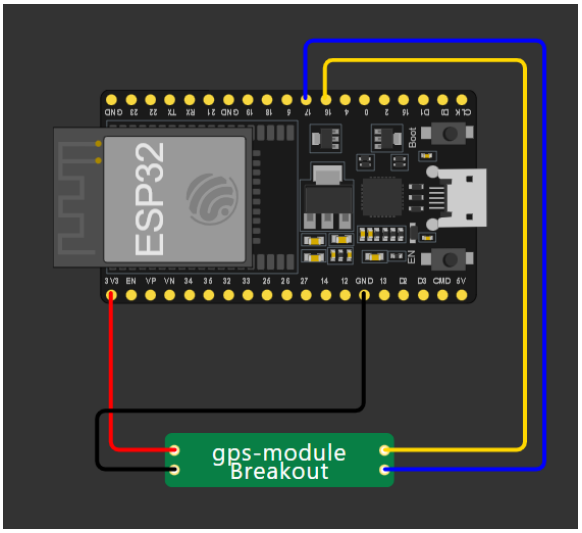
Firebase untuk penyimpanan. Pengguna akhir dapat mengakses data GPS secara *real-time* melalui aplikasi Android yang terhubung ke Firebase.

4.6. Rancangan *Database*

Gambar 3. Rancangan *database*

Pada Gambar 3 menunjukkan rancangan *database* yang digunakan dalam sistem ini. Dalam *database* tersebut, node route memiliki banyak *child* dengan struktur data yang seragam, yaitu ‘lat’ dan ‘long’ dengan tipe data numerik, serta ‘GPS Time’ dan ‘time’ dengan tipe data *string*. Data ini merepresentasikan lokasi kendaraan yang akan divisualisasikan dalam aplikasi Android.

4.7. Rancangan Rangkaian Elektronik



Gambar 4. Rancangan rangkaian elektronik

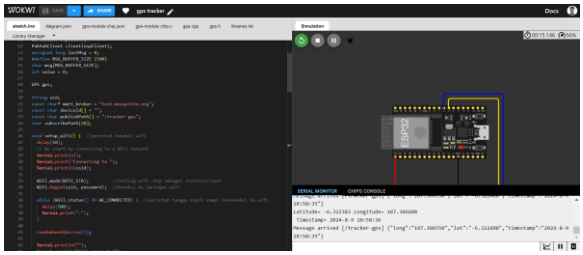
Gambar 4 menunjukan rancangan rangkaian elektronik dari *embedded system* pelacak aktivitas kendaraan. Dapat dilihat bahwa modul GPS terhubung dengan mikrokontroller ESP32 melalui 4 pin. Berikut tabel alokasi pin pada rancangan rangkaian elektronik:

Tabel 3. Alokasi pin pada *embedded system*

ESP32	Modul GPS
3V3	VCC
GND	GND
Pin 16	RX
Pin 17	TX

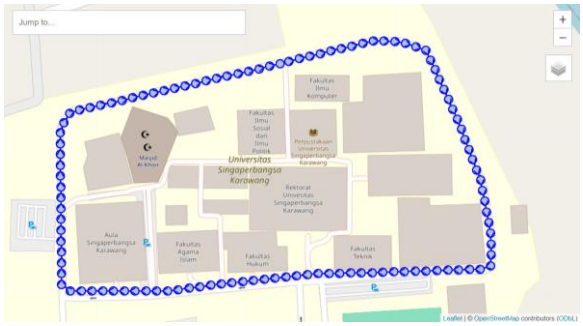
4.8. Pembuatan Program simulasi *Embedded System*

Untuk melakukan simulasi dan pembuatan program *embedded system*, penelitian ini menggunakan platform simulator elekronik online bernama wokwi.



Gambar 5. Pembuatan program dan simulasi *embedded system* di wokwi

Karena modul GPS belum tersedia pada wokwi, simulasi pelacakan lokasi memerlukan pembuatan modul GPS menggunakan komponen *custom chip* yang diprogram untuk menghasilkan data dalam format yang sesuai dengan modul GPS asli. Secara umum, modul GPS bekerja dengan menangkap dan memproses sinyal dari satelit, kemudian menghasilkan data dalam format NMEA. Oleh karena itu, untuk mensimulasikan pelacakan, rute kendaraan dibuat menggunakan NMEA generator di situs web nmeagen.org. Pada situs web ini, rute kendaraan dibuat dengan menggambar titik-titik pada peta yang menunjukkan setiap lokasi yang dilalui. Selain itu, terdapat fitur "multi-point line" yang memungkinkan penentuan kecepatan dan menggambar banyak titik lokasi secara linear.



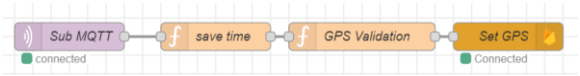
Gambar 6. Rute simulasi aktivitas kendaraan

Dalam penelitian ini, rute simulasi aktivitas kendaraan dirancang seolah-olah berputar di sekitar area kampus Universitas Singaperbangsa Karawang dengan kecepatan rata-rata sekitar 20 km/jam. Rute tersebut kemudian dihasilkan dalam bentuk file

berformat NMEA. Data dari file tersebut dimasukkan ke dalam kode program modul GPS di Wokwi.

Setelah kode pada modul GPS selesai dibuat, langkah selanjutnya adalah mengembangkan kode untuk mikrokontroler yang berfungsi menangkap dan mendekode data NMEA dari modul GPS, sehingga titik koordinat dapat diperoleh. Data tersebut kemudian dikirim ke MQTT broker melalui topik yang sesuai.

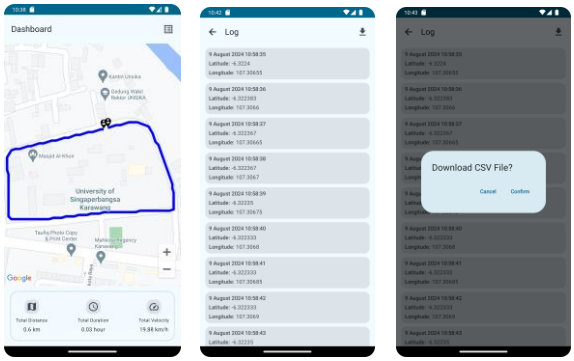
4.9. Pembuatan *Backend*



Gambar 7. Alur node-red

Dalam pengembangan backend, Node-RED digunakan sebagai penghubung antara MQTT broker pada *embedded system* dan *database*. Alur sistem ini terdiri dari tiga node utama: "MQTT In" untuk menangkap data dari topik MQTT, "Function" untuk menandakan waktu data diterima dari MQTT (save time) dan memvalidasi data sebelum disimpan (GPS Validation), serta "Firebase Out" untuk menyimpan data GPS yang valid ke Firebase Realtime Database.

4.10. Pembuatan Aplikasi Android

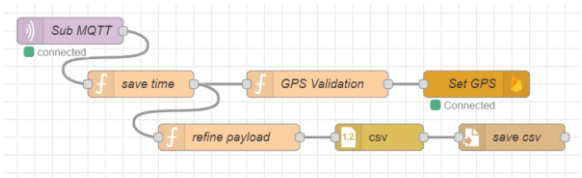


Gambar 8. Antarmuka aplikasi *data logging*

Dalam pembuatan aplikasi *data logging* aktivitas kendaraan di Android, aplikasi ini diprogram untuk mengambil data yang telah disimpan oleh backend di Firebase Realtime Database. Aplikasi *data logging* ini memiliki dua halaman utama, yaitu halaman dashboard dan halaman log. Halaman dashboard menampilkan visualisasi rute kendaraan di peta menggunakan Google Maps, serta ringkasan aktivitas kendaraan yang meliputi total jarak, total waktu, dan kecepatan rata-rata. Di halaman log, ditampilkan daftar titik koordinat kendaraan beserta waktu saat berada di setiap titik tersebut. Selain itu, halaman log juga dilengkapi dengan tombol unduh untuk menyimpan data logger dalam format CSV di penyimpanan internal perangkat.

4.11. Pengujian Sistem

Pada penelitian ini, sistem dilakukan dengan beberapa pengujian. Pengujian pertama membandingkan data titik koordinat dan waktu yang dihasilkan oleh file NMEA pada MQTT broker setelah ditambahkan objek “time” dan data pada aplikasi android. Pengujian kedua, dilakukan pembandingkan visualisasi rute pada peta di aplikasi android dengan rute di NMEA, Pengujian terakhir, *unit testing* dilakukan pada fungsi perhitungan ringkasan aktivitas kendaraan untuk memastikan perhitungan tersebut sudah tepat.



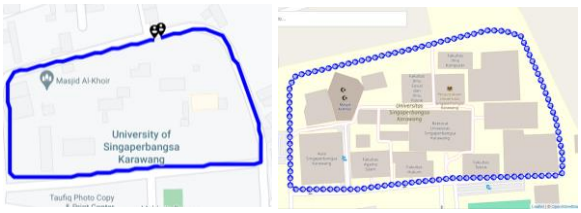
Gambar 9. Alur penyimpanan data dari MQTT ke CSV

Untuk membandingkan kedua data, data dari MQTT broker perlu disimpan terlebih dahulu dalam format CSV. Pada Node-RED, *payload* data yang dikirim dari MQTT, setelah ditambahkan objek “time”, disesuaikan agar penamaan key-nya konsisten dengan data pada aplikasi Android atau *database*. Proses konversi ke format CSV dilakukan menggunakan node "CSV" di Node-RED, yang berfungsi untuk mengubah objek dalam *payload* menjadi CSV. Selanjutnya, file CSV tersebut disimpan ke penyimpanan perangkat melalui node "Write File".

Time	MQTT			Android			Err
	Latitude	Longitude	GPS Time	Latitude	Longitude	GPS Time	
8/11/2024 19:41:23	-6.3224	107.307	8/9/2024 10:58:35	-6.3224	107.307	8/9/2024 10:58:35	SUCCESS
8/11/2024 19:41:28	-6.3224	107.307	8/9/2024 10:58:35				FAIL
8/11/2024 19:41:32	-6.3224	107.307	8/9/2024 10:58:36	-6.3224	107.307	8/9/2024 10:58:36	SUCCESS
8/11/2024 19:41:37	-6.3224	107.307	8/9/2024 10:58:37	-6.3224	107.307	8/9/2024 10:58:37	SUCCESS
8/11/2024 19:41:41	-6.3224	107.307	8/9/2024 10:58:38	-6.3224	107.307	8/9/2024 10:58:38	SUCCESS
8/11/2024 19:41:45	-6.3224	107.307	8/9/2024 10:58:38				FAIL
8/11/2024 19:41:50	-6.3224	107.307	8/9/2024 10:58:39	-6.3224	107.307	8/9/2024 10:58:39	SUCCESS
8/11/2024 19:41:55	-6.3223	107.307	8/9/2024 10:58:40	-6.3223	107.307	8/9/2024 10:58:40	SUCCESS
8/11/2024 19:42:00	-6.3223	107.307	8/9/2024 10:58:40				FAIL
8/11/2024 19:42:05	-6.3223	107.307	8/9/2024 10:58:41	-6.3223	107.307	8/9/2024 10:58:41	SUCCESS
8/11/2024 19:42:10	-6.3223	107.307	8/9/2024 10:58:42	-6.3223	107.307	8/9/2024 10:58:42	SUCCESS
8/11/2024 19:42:14	-6.3224	107.307	8/9/2024 10:58:43	-6.3224	107.307	8/9/2024 10:58:43	SUCCESS
8/11/2024 19:42:19	-6.3224	107.307	8/9/2024 10:58:43				FAIL
8/11/2024 19:42:24	-6.3224	107.307	8/9/2024 10:58:44	-6.3224	107.307	8/9/2024 10:58:44	SUCCESS
8/11/2024 19:42:29	-6.3224	107.307	8/9/2024 10:58:45	-6.3224	107.307	8/9/2024 10:58:45	SUCCESS

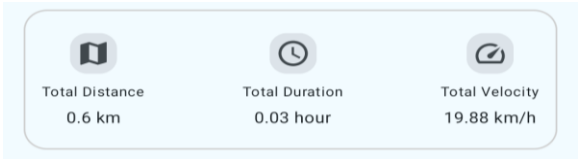
Gambar 10. Hasil pengujian perbandingan data MQTT dengan Aplikasi Android

Setelah data pada MQTT broker dan aplikasi android disimpan di dalam file CSV, maka dilakukan perbandingan kedua data GPS tersebut. Pengujian dilakukan menggunakan 15 buah sampel yang dimana data tersebut berupa titik koordinat kendaraan (Latitude dan Longitude), waktu data ditangkap oleh backend (Time), dan waktu lokasi dihasilkan oleh GPS (GPS Time). Hasil pengujian dinyatakan berhasil karena dapat dilihat pada Gambar 10, data yang dikirim oleh *embedded system* ke MQTT terdapat 4 data duplikat. Sedangkan, pada aplikasi android data tersebut benilai kosong dikarenakan backend telah berhasil menghapus data duplikat tersebut sehingga aplikasi dapat menggunakan data tersebut dengan baik.



Gambar 11. Perbandingan rute kendaraan di aplikasi android dengan NMEA

Pada pengujian selanjutnya Gambar 11 menunjukan bahwa aplikasi telah memvisualisasikan rute kendaraan yang sama dengan rute pada NMEA melalui Google Maps. Sehingga berdasarkan perbandingan tersebut, dapat dinyatakan pengujian visualisasi rute pada peta di aplikasi telah berhasil.



Gambar 12. Ringkasan aktivitas kendaraan pada halaman dashboard

Selanjutnya, pengujian dilakukan untuk memastikan bahwa perhitungan ringkasan aktivitas kendaraan, termasuk total jarak, durasi, dan kecepatan pada halaman dashboard, telah sesuai. Pengujian ini dilakukan menggunakan metode *unit testing* pada berbagai metode yang menghitung ringkasan aktivitas kendaraan, dengan memanfaatkan library JUnit 4 di Android Studio. Sebelum pengujian dimulai, skenario pengujian dipersiapkan terlebih dahulu untuk memastikan hasil yang diperoleh sesuai dengan yang diharapkan.

Tabel 4. Skenario *unit testing* pada ringkasan aktivitas kendaraan

Skenario	Fungsi	Input	Ekspektasi Output
Hitung Jarak dari Dua Titik	calcTotal Distance	LatLng(0.0, 0.0), LatLng(1.0, 1.0)	± 156900.0 meter
Hitung Durasi dalam Jam	durationIn Hour	startTime = 2024-08-12T10:00:00, endTime = 2024-08-12T11:00:00	1.0 jam
Hitung Kecepatan	calcVelocity	durationSecond = 3600, distance = 1000	1.0 km/h

Gambar 13. Hasil pengujian *unit testing*

Setelah pengujian dilakukan menggunakan skenario yang tercantum pada Tabel 4, setiap skenario pengujian dinyatakan berhasil, sebagaimana ditunjukkan pada Gambar 13. Hasil pengujian menunjukkan bahwa output yang dihasilkan sesuai dengan ekspektasi berdasarkan input yang diberikan, sehingga pengujian dianggap berhasil.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem pelacakan kendaraan berbasis IoT yang terintegrasi dengan aplikasi Android. Melalui simulasi embedded system dan backend, sistem ini mampu mengirim data aktivitas kendaraan secara *real-time*. Pengujian perbandingan peta dan *unit testing* untuk ringkasan aktivitas kendaraan menunjukkan bahwa sistem ini mampu memantau aktivitas kendaraan, seperti posisi pada peta, jarak tempuh, kecepatan, dan durasi perjalanan, dengan akurat melalui aplikasi Android. Implementasi sistem ini diharapkan dapat menjadi solusi efektif dalam mengurangi perilaku buruk pengemudi dan meningkatkan efisiensi operasional di sektor logistik.

Untuk penilitian selanjutnya disarakan untuk menambahkan fitur yang lebih kompleks seperti pemanfaatan algoritma *machine learning* untuk analisis perilaku yang lebih mendalam, notifikasi batas ketentuan kecepatan atau rute kendaraan, dan implementasi sistem dengan modul yang lebih akurat. Selain itu, peningkatan keamanan sistem juga diperlukan untuk melindungi privasi data dan mencegah akses yang tidak sah.

DAFTAR PUSTAKA

[1] Rombekila and B. L. Entamoing, "PROTOTYPE SMART HOME BERBASIS IoT DENGAN HANDPHONE ANDROID MENGGUNAKAN NODEMCU ESP32," *Jurnal Teknik AMATA*, vol. 3, no. 1, pp. 32-37, 2022

[2] R. Ricat, "APLIKASI GOOGLE MAPS API SISTEM INFORMASI PARIWISATA SUMATERA BARAT BERBASIS WEB," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 8, no. 1, pp. 137-142, 2020

[3] A. F. Oklilas, A. R. Passarella, S. and M. A. Buchari, "PENINGKATAN KEMAMPUAN SISWA SMKN1TANJUNG PANDANBELITUNGDALAM SIMULASIONLINESISTEMPALANG PINTUKERETAAPI," *Jurnal Pengabdian Kolaborasi dan Inovasi IPTEKS*, vol. 1, no. 5, pp. 698-704, 2023

[4] Y. K. Nugraha and A. Hajar, "Pemanfaatan Informasi Geospasial Dasar (IGD) untuk Analisis Penyimpangan Arah Kiblat Bangunan Masjid secara Masal," *Jurnal Teknik: Media Pengembangan Ilmu dan Aplikasi Teknik*, vol. 21, no. 2, pp. 202-214, 2023

[5] F. D. Ikram, I. A. Febryanti, I. S. Auliya, A. J. Pardede and D. O. Radianto, "Peningkatan

Keamanan Dan Penyelamatan Korban Insiden Di Wilayah," *Konstruksi: Publikasi Ilmu Teknik, Perencanaan Tata Ruang dan Teknik Sipil,* vol. 2, no. 2, pp. 189-196, 2024

[6] A. P. W. Hendrawan and N. P. Agustini, "Simulasi Kendali Dan Monitoring Daya Listrik Peralatan Rumah Tangga Berbasis ESP32," *ALINIER Journal of Artificial Intelligence & Applications,* vol. 3, no. 1, pp. 54-68, 2022

[7] A. Hakim, P. A. Saputra and F. Hendajani, "PERANCANGAN RUMAH PINTAR BERBASIS IOT UNTUK MEMONITOR DAN MENGONTROL PERANGKAT RUMAH MENGGUNAKAN CISCO PACKET TRACER 8.1.1," *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK),* vol. 7, no. 1, pp. 192-199, 2023

[8] I. K. H. Dwipayana, I. H. Santoso and N. Bogi, "RANCANG BANGUN SISTEM TRACKING PENDAKI BERBASIS *INTERNET OF THINGS* DENGAN MODUL LORA," *e-Proceeding of Engineering,* vol. 8, no. 6, pp. 11829-11838, 2021

[9] R. L. Ditha, S. T. Faulina and Wisnumurti, "RANCANG BANGUN APLIKASI LAYANAN PENGADUAN PADA," *Jurnal Informatika dan Komputer (JIK),* vol. 14, no. 2, pp. 25-35, 2023

[10] Y. Darnita and M. , "RANCANG BANGUN APLIKASI MOBILEPENJADWAL PERKULIAHAN DENGAN FIREBASEDENGAN REALTIME NOTIFICATIO," *Jurnal Pseudocode,* vol. 8, no. 1, pp. 58-65, 2021

[11] R. Andrianto and M. H. Munandar, "APLIKASI E-COMMERCE PENJUALAN PAKAIAN BERBASIS," *Journal Computer Science and Information Technology (JCoInT),* vol. 3, no. 1, pp. 20-29, 2022

[12] Cartrack, "Harga yang Harus Ditanggung Perusahaan dari Perilaku Mengemudi Sopir yang Tidak Aman," PT CARTRACK TECHNOLOGIES INDONESIA, 22 February 2021. [Online]. Available: https://www.cartrack.id/id/harga-yang-harus-ditanggung-perusahaan-dari-perilaku-mengemudi-sopir-yang-tidak-aman. [Accessed 12 August 2024]