

Sistem Monitoring Kendaraan Logistik Secara *Real-Time* Berbasis *Internet of Things* (IoT)

1st Ghifari Munawar
Departemen Teknik Komputer dan
Informatika
Politeknik Negeri Bandung
Bandung, Indonesia
ghifari.munawar@polban.ac.id

Abstract—Biaya transportasi merupakan salah satu komponen terbesar dalam proses logistik, sehingga diperlukan pengelolaan yang lebih efisien. Penelitian ini bertujuan mengembangkan sistem pelacakan kendaraan logistik berbasis web yang memanfaatkan teknologi *Internet of Things* (IoT) untuk meningkatkan kualitas pengelolaan logistik, khususnya pada aspek transportasi. Sistem ini memungkinkan pelaku logistik untuk memantau kendaraan secara *real-time*, termasuk posisi dan histori kecepatan kendaraan. Fokus pengembangan sistem terletak pada modul pelacakan dan penelusuran (*tracking dan tracing*), yang memungkinkan monitoring posisi kendaraan, jalur yang telah ditempuh, serta kecepatan kendaraan. Alat GPS yang digunakan memiliki komponen GPS, GPRS, dan sensor kecepatan yang terintegrasi langsung dengan kendaraan. Aplikasi web yang dikembangkan menggunakan *framework* ASP.NET MVC dengan basis data SQL Server, serta menggunakan Google Map API untuk visualisasi peta. Hasil pengujian menunjukkan bahwa integrasi data antara perangkat GPS dan aplikasi web berjalan dengan baik, memungkinkan monitoring kendaraan secara *real-time*.

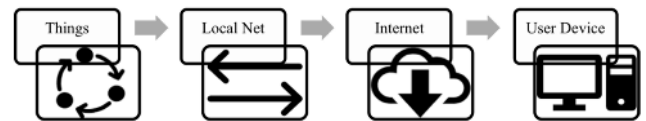
Keywords—*Internet of Things* (IoT), Sistem Monitoring, Pelacakan Kendaraan, Aplikasi Web.

I. LATAR BELAKANG

Pertumbuhan pesat sektor logistik dan kebutuhan akan solusi yang dapat meningkatkan efisiensi menjadi peluang penting bagi para pelaku industri untuk turut serta dalam menghadapi berbagai tantangan logistik [1]. Sistem logistik nasional merupakan elemen kunci yang mendukung perekonomian suatu negara. Namun, kurangnya efisiensi dalam sistem ini dapat menjadi penghambat bagi aktivitas bisnis di Indonesia, yang pada akhirnya menurunkan daya saing produk lokal. Menurut Lembaga Pengkajian Penelitian dan Pengembangan Ekonomi (LP3EI), biaya transportasi adalah komponen terbesar dalam biaya logistik nasional yang secara keseluruhan mencapai 24% dari produk domestik bruto (PDB), sehingga optimalisasi dalam pengelolaan transportasi logistik sangat diperlukan [2], [3].

Untuk itu, penerapan teknologi informasi dalam logistik harus ditingkatkan, salah satunya melalui pengembangan sistem pelacakan kendaraan logistik yang memungkinkan pemantauan pergerakan secara *real-time*. Teknologi *Internet of Things* (IoT) kini menjadi tren global yang diterapkan di berbagai sektor, termasuk logistik [4]. Dengan IoT, kendaraan logistik dapat dilengkapi dengan perangkat yang mampu memberikan data *real-time*, seperti lokasi (longitude dan latitude) serta kecepatan kendaraan. IoT juga memungkinkan integrasi yang aman dengan sistem server, berkat pengiriman data langsung dari perangkat ke server secara *point-to-point*.

Pertukaran informasi pada sistem IoT dapat bekerja tanpa campur tangan manusia secara *real-time* satu sama lainnya (lihat pada Gambar 1), hal ini dapat membantu meningkatkan transparansi dan efisiensi rantai pasok, termasuk dalam pelacakan aset logistik.



Gambar 1. Pertukaran informasi pada IoT [4]

Penelitian ini bertujuan untuk mengembangkan sistem pelacakan kendaraan berbasis web yang memanfaatkan IoT agar dapat meningkatkan efisiensi transportasi logistik. Dengan adanya monitoring *real-time* terhadap posisi dan kecepatan kendaraan, diharapkan akan mendukung peningkatan kualitas logistik serta efisiensi dalam pengelolaan biaya transportasi. ASP.NET MVC digunakan sebagai *framework* dalam mengembangkan aplikasi web, baik untuk teknologi front-end maupun back-end, sementara MS SQL Server digunakan sebagai database management system (DBMS). Selain itu, Google Map API digunakan sebagai tools untuk pemetaan, dengan beberapa fungsional yang tersedia seperti map display, marker, geocoding, dan geolocation services.

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengembangkan sistem pelacakan kendaraan logistik secara *real-time* dengan memanfaatkan IoT ?
2. Bagaimana fungsionalitas sistem yang dihasilkan?

Sesuai dengan rumusan masalah tersebut, adapun tujuan dari penelitian ini adalah:

1. Mengembangkan sistem pelacakan kendaraan logistik secara *real-time* berbasis IoT.
2. Menguji fungsionalitas sistem yang dikembangkan.

II. TINJAUAN PUSTAKA

A. *Internet of Things* (IoT)

Internet of Things (IoT) merepresentasikan tahap baru dalam transformasi digital bagi masyarakat dan ekonomi, di mana objek-objek fisik dapat terhubung melalui jaringan komunikasi untuk bertukar data mengenai status atau kondisi lingkungannya, sebagaimana diilustrasikan pada Gambar 2.

Teknologi ini kini telah diterapkan di berbagai sektor, seperti *smart home/building*, transportasi, kesehatan, dan logistik [5]. Implementasi IoT memungkinkan pengumpulan data yang akurat dan *real-time*, sehingga objek-objek yang sebelumnya tidak terhubung ke internet kini dapat memberikan informasi berharga yang mendukung pengambilan keputusan yang lebih efektif dan efisien.



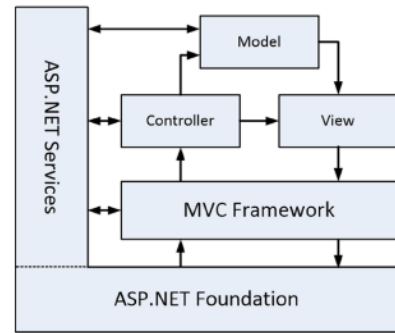
Gambar 2. Ekosistem pada IoT [5]

B. Framework ASP.NET MVC

Framework ASP.NET MVC adalah pendekatan terstruktur untuk pengembangan aplikasi web yang memanfaatkan pola desain *Model-View-Controller* (MVC). Arsitektur ini memisahkan aplikasi menjadi tiga komponen yang saling berhubungan, yang memungkinkan pengorganisasian kode yang efisien dan pengembangan paralel. Framework ini bermanfaat untuk pengembangan aplikasi web yang *scalable* dan *maintainable*, karena mendukung pemisahan masalah yang jelas dan mendukung pengembangan berbasis pengujian. Berikut ini adalah aspek-aspek utama dalam merancang kerangka kerja ASP.NET MVC [6]:

1. **Model:** Bertanggung jawab untuk mengelola data dan logika bisnis aplikasi. Komponen ini secara langsung menangani data yang dioperasikan di aplikasi, termasuk mengambil data dari database, memprosesnya, dan mengembalikannya ke *controller* atau *view* sesuai kebutuhan.
2. **View:** Bertugas menyajikan data kepada pengguna. Komponen ini merupakan *user interface* aplikasi, yang menampilkan data yang disediakan model dalam format yang dapat dipahami dan diakses oleh pengguna. *View* bertanggung jawab untuk menampilkan *user interface* dan memperbaruinya sebagai respon terhadap perubahan model.
3. **Controller:** Bertindak sebagai perantara antara *Model* dan *View*. *Controller* memproses input pengguna, berinteraksi dengan *model* untuk mengambil atau memperbarui data, dan menentukan *view* mana yang akan ditampilkan. *Controller* menginterpretasikan tindakan pengguna, seperti klik atau input keyboard, dan mengubahnya menjadi perintah untuk *model* atau *view*.

Berikut ini adalah arsitektur MVC pada *framework* ASP.NET:



Gambar 3. Arsitektur MVC pada ASP.NET [7]

C. MS SQL Server

MS SQL Server adalah sistem manajemen basis data relasional (RDBMS) yang dikembangkan oleh Microsoft, yang dirancang untuk menyimpan, mengambil, dan mengelola data secara efisien. MS SQL Server mendukung berbagai aplikasi, mulai dari aplikasi *stand alone* hingga aplikasi berskala besar yang terhubung ke internet dengan banyak pengguna bersamaan. SQL Server terutama digunakan pada *platform* sistem operasi Microsoft Windows dan dikenal sebagai salah satu RDBMS terkemuka di pasaran. Sistem ini menggunakan MS-SQL dan Transact-SQL sebagai bahasa pemrograman utamanya, yang didasarkan pada aljabar relasional untuk manipulasi data dan kontrol akses [8].

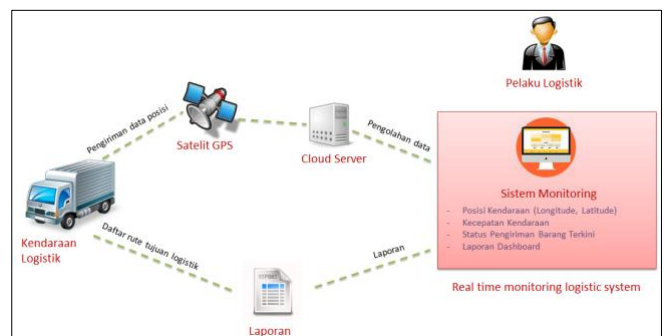
D. Google Map API

Google Maps API merupakan *library* yang memungkinkan pengembang untuk mengintegrasikan Google Maps ke dalam aplikasinya, serta menyediakan kemampuan pemetaan yang dinamis. API ini telah menjadi standar dalam pemetaan online karena fiturnya yang lengkap dan kemudahan penggunaannya. API ini telah mendukung berbagai fungsi, termasuk identifikasi lokasi, pemetaan rute, dan analisis lalu lintas, sehingga membuatnya mudah diimplementasikan pada berbagai aplikasi [9]. Salah satu penerapan API ini adalah untuk menghitung jumlah kendaraan, seperti pada studi [10], yang bermanfaat untuk menentukan waktu tempuh, kondisi lalu lintas, bahkan untuk mendeteksi adanya kecelakaan. Google MAP API telah dikembangkan ke berbagai aplikasi pemetaan lainnya, seperti contoh aplikasi berikut [9]:

1. runtastic (<https://www.runtastic.com/>)
2. all state (<https://goodlife.allstate.com/goodhome/>)
3. sun surveyor (<http://www.sunsurveyor.com/>)

III. SPESIFIKASI DAN RANCANGAN SISTEM

A. Deskripsi Umum Sistem



Gambar 4. Gambaran Umum Sistem

Sistem yang akan dikembangkan adalah aplikasi web pelacakan kendaraan logistik dengan pemanfaatan teknologi IoT. Fokus pada studi ini adalah rancangan sistem monitoring dan integrasi dengan IoT agar komunikasi data dapat berjalan dengan baik secara point-to-point, artinya alat IoT yang dipasangkan pada kendaraan akan mengirimkan data geolocation-nya secara langsung ke server web, sehingga informasi lokasi dapat dilacak secara real-time. Adapun fitur utama yang dikembangkan pada sistem monitoring ini sebagaimana tertulis pada Tabel I.

TABLE I. FITUR UTAMA APLIKASI

ID	Deskripsi
FR1	Otorisasi dan Otentikasi Sistem
FR2	Manajemen Kendaraan
FR3	Manajemen Kurir
FR4	Manajemen Pengiriman
FR5	Tracking Kendaraan
FR6	Dashboard View
FR7	Backend Service untuk Integrasi IoT

B. Spesifikasi Komponen IoT yang digunakan



Gambar 5. Komponen Alat IoT

Tabel II menunjukkan bill of material (BOM) alat IoT yang digunakan dalam sistem:

TABLE II. BILL OF MATERIAL (BOM)

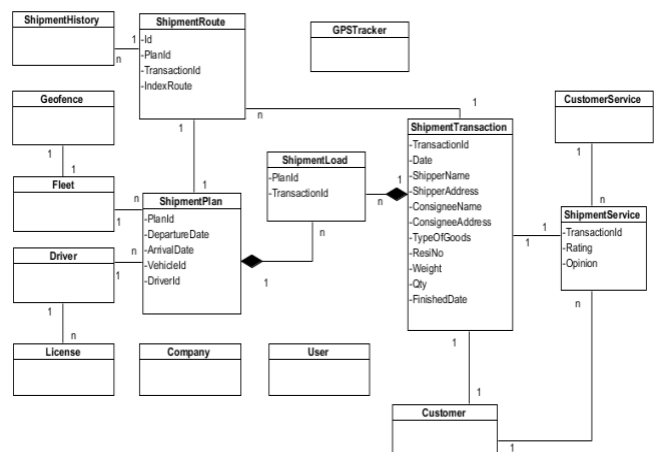
Part Type	Designator	Footprint	Qty
IC STM 32F103RET	IC2	LQFP64	1
IC 24C256	IC1	SM D SOIC8	1
SIM 908CL	JP3	SIM 908CL	1
SOKET SIM 908			1
HOLDER SIM CARD (8PIN)	JP10	HOLDER_SIM B	1
KRISTAL 8M HZ	X2	XTAL1	1
KRISTAL 32.768KHz	X1	AXIAL	1
TRANSISTOR BC847	T1, T2, T3, T4	SOT23	4
LM 1117	REG1	REGSM D	1
LM 2576-5VOLT	REG2	TO263-5	1
DIODA 1N4002	DR1, DR2, DR3, DR4	DSM D	4
KAPASITOR 10uF	C7, CR4	SM D 0805	2

Part Type	Designator	Footprint	Qty
KAPASITOR 100nF	C1, C2, C3, C4, C5, C6, C8, C9, CR1, CR3	SM D 0805	10
KAPASITOR 22pF	C10, C13, C14	SM D 0805	3
KAPASITOR 10pF	C11, C12	SM D 0805	2
RESISTOR 47K	R26, R27	SM D 0805	2
RESISTOR 10K	R15, R16, R18, R19, R20, R33	SM D 0805	6
RESISTOR 4K7	R14, R17, R22, R23, R24, R25	SM D 0805	6
RESISTOR 1K5	R6, R7, R8, R9, R10	SM D 0805	5
RESISTOR 1K	R3, R4, R5	SM D 0805	3
RESISTOR 220	R11, R12, R13, R21	SM D 0805	4
RESISTOR 22	R28, R29, R30	SM D 0805	3
RESISTOR 0	R31, R32	SM D 0805	2
IDC 40x2 Type L (@PCB)	JP7	IDC10	
SW ITCH PUSHBUTON	S1	PBUTTON	1
WHITE HOUSING 2PIN	JP2	FCP2	1
LED HIJAU	LED3	SM D 0805	1
LED KUNING	LED2	SM D 0805	1
LED MERAH	LED4	SM D 0805	1
LED BIRU	LED1	SM D 0805	1
SPACER 1 CM			4

*) Keterangan: Dimensi PCB 107mm x 70 mm, Dual Layer, Tebal PCB 1.6mm.

C. Perancangan Sistem

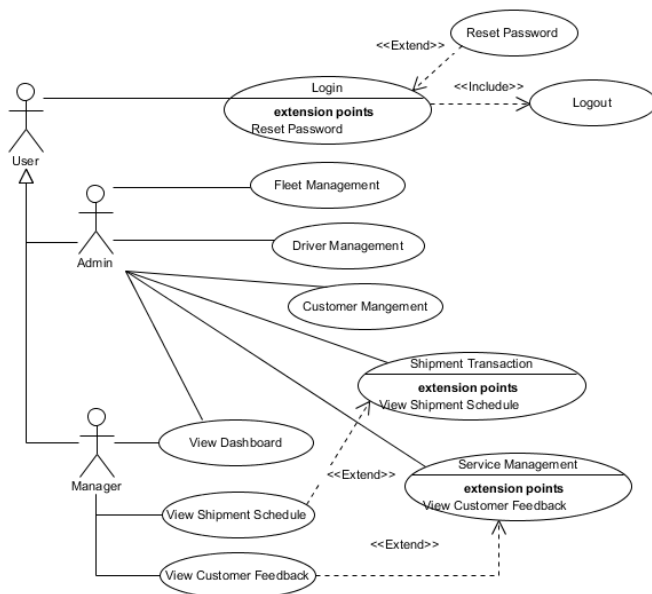
Pada tahap ini dirancang model desain user interface (UI) sistem sesuai kebutuhan, mulai dari tampilan UI pada tiap halaman sistem, mekanisme integrasi data dengan alat IoT, desain database, dll. Tahap ini yang akan menjadi dasar dalam proses implementasi prototipenya. Berikut ini adalah salah satu rancangan yang dikembangkan dalam bentuk diagram model domain:



Gambar 6. Diagram Model Domain

Diagram model domain pada gambar 6 di atas menggambarkan relasi antar entitas. Sebagai contoh, ShipmentTransaction memiliki relasi dengan entitas lainnya, yaitu ShipmentLoad, ShipmentRoute, ShipmentService, dan Customer. Relasi tersebut memiliki multiplicity berupa one to one, dan one to many, dimana hal ini berarti terdapat property pada kelas ShipmentTransaction yang dimiliki/digunakan oleh entitas lainnya.

Kebutuhan fungsional yang telah ditentukan selanjutnya dimodelkan dalam bentuk diagram use case, sebagaimana dapat dilihat pada Gambar 7. Terdapat dua aktor utama dalam rancangan ini, yakni admin, dan manager. Admin memiliki fasilitas penuh untuk mengakses beberapa fitur utama, termasuk pengelolaan kendaraan, kurir, pelanggan, transaksi pengiriman, dan tampilan dashboard. Sementara manager hanya memiliki akses untuk melihat laporan dalam bentuk dashboard maupun feedback pelanggan.



Gambar 7. Diagram Use Case

D. Implementasi Sistem

Tahap ini dilakukan dengan kegiatan membangun source code program, mengkompilasi program, memperbaiki kesalahan kode, dan mengeksekusi program. Dalam melakukan implementasi prototipe dalam penelitian ini, peneliti menggunakan perangkat lunak (software), perangkat keras (hardware) dan framework / tools nya. Adapun perangkat yang digunakan dalam penelitian ini ditunjukkan pada Tabel III berikut :

TABLE III. SOFTWARE, HARDWARE, DAN TOOLS PENDUKUNG

Komponen	Spesifikasi
1. Software	
Sistem Operasi	Windows
Integrated Development Environment (IDE)	Visual Studio
Framework	.NET
Bahasa Pemrograman	C# / C
Designer Tools	UML for Visual Studio

Komponen	Spesifikasi
1. Software	
2. Hardware	
CPU	Processor Intel Core i5
Memory	12 GB
Harddisk	Minimal 5 GB ruang kosong
I/O Tools	Mouse & Keyboard
3. Tools Pendukung	
Framework	ASP.NET MVC
Object Relational Mapping (ORM) Tools	Entity Framework
API	Web API, Google Map API

E. Mekanisme Integrasi dengan Alat IoT

Pada tahap ini, sistem monitoring yang telah dibangun (Web API) selanjutnya diintegrasikan dengan alat IoT, dengan fungsional utamanya adalah agar pengiriman data berupa posisi kendaraan (longitude dan latitude) serta kecepatan kendaraan (Km/h) dapat diterima oleh aplikasi web dengan baik. Hal ini dilakukan dengan mengatur alamat pengiriman data ke server pada alat IoT dengan rentang waktu tertentu (misal: pengiriman data setiap 5 detik sekali). Secara spesifik, atribut data yang dikirimkan ke server berupa id device, latitude, longitude, date, satellites, speedOTG, dan course. Adapun gambar dibawah ini menunjukkan alat IoT yang dipasangkan pada kendaraan logistik:



Gambar 8. Alat IoT untuk Kendaraan Logistik

Adapun kode program dibawah ini adalah source code yang disematkan pada alat IoT untuk mengirimkan data tersebut ke aplikasi web.

```
File: HTTP.c

#include "main.h"

unsigned char url1[100] = IPSEVER
unsigned char appsl[100] =
"gpstrackers/updatelocation?";

void SetUpHTTP(void){

    int8_t answer;

    answer =
sendATcommand("AT+SAPBR=3,1,\"Contype\", \"GPRS\"
", "OK", 10000000);
    printf("answer = %d\n", answer);
    answer =
sendATcommand("AT+SAPBR=3,1,\"APN\", \"internet\"
", "OK", 10000000);
```

```

printf("answer = %d\n", answer);
answer =
sendATcommand("AT+SAPBR=3,1,\"USER\",\"\");
"OK", 10000000);
printf("answer = %d\n", answer);
answer =
sendATcommand("AT+SAPBR=3,1,\"PWD\",\"\");
"OK", 10000000);
printf("answer = %d\n", answer);
answer = sendATcommand("AT+HTTPTERM",
"OK", 10000000);
printf("answer = %d\n", answer);
answer = sendATcommand("AT+SAPBR=1,1",
"OK", 10000000);
vTaskDelay(5000);
printf("answer = %d\n", answer);
}
....
....

// posting data ke server
sprintf(aux_str,
"AT+HTTPPARA=\"URL\", \"http://%s/%s",
url1,apps1);
ModemSendString(aux_str,
strlen(aux_str));
vTaskDelay(1000);
sprintf(frame,
"data=%s,%s,%s,%s,%s,%s,%s,%s,%s\",idDevice,long
itude,latitude, altitude, date, satellites,
speedOTG, course);

ModemSendString(frame, strlen(frame));
vTaskDelay(1500);

answer =
sendATcommand("AT+HTTPACTION=0",
"+HTTPACTION:0,200", 10000000);

printf("answer = %d\n", answer);

return answer;
}

```

F. Mekanisme Pengujian

Tahap pengujian merupakan suatu proses yang dapat menentukan apakah sistem yang dibangun telah sesuai dengan spesifikasi yang dirancang atau tidak. Teknik pengujian dilakukan dengan menguji fungsionalitas sistem yang telah dibangun secara blackbox, dengan tiap-tiap butir kebutuhan fungsional sebagaimana yang telah didefinisikan pada Tabel I.

IV. HASIL IMPLEMENTASI DAN PENGUJIAN

A. Hasil Implementasi Sistem

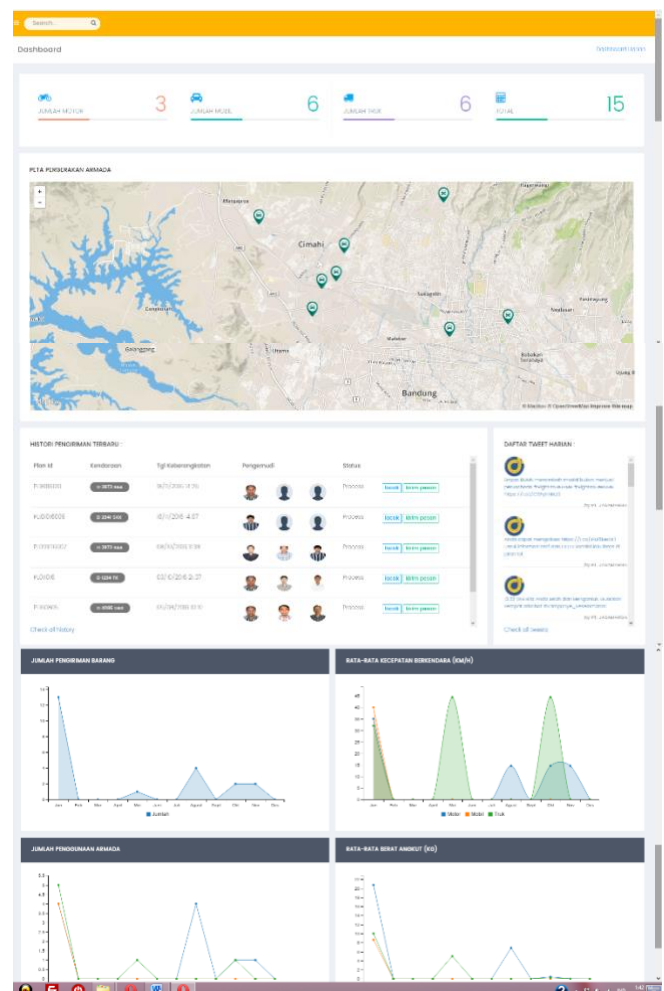
Sistem pelacakan kendaraan logistik dengan memanfaatkan perangkat IoT telah berhasil dikembangkan. Setiap pergerakan kendaraan dapat divisualisasikan secara real-time pada sistem, baik posisi kendaraan, maupun kecepatan berkendarnya. Tabel IV dibawah ini adalah status implementasi dari kebutuhan fungsional yang telah didefinisikan sebelumnya:

TABLE IV. STATUS IMPLEMENTASI

ID	Deskripsi	Status Implementasi
FR1	Otorisasi dan Otentikasi Sistem	Terdapat dua aktor utama, yakni admin dan <i>manager</i> dengan aksesibilitas sistem yang berbeda sesuai dengan peranannya.

ID	Deskripsi	Status Implementasi
FR2	Manajemen Kendaraan	Telah dikembangkan halaman pengelolaan kendaraan dengan fungsionalitas <i>Create, Read, Update, Delete</i> (CRUD).
FR3	Manajemen Kurir	Telah dikembangkan halaman pengelolaan kurir dengan fungsionalitas <i>Create, Read, Update, Delete</i> (CRUD).
FR4	Manajemen Pengiriman	Telah dikembangkan halaman transaksi pengiriman barang dengan fungsionalitas <i>Create, Read, Update, Delete</i> (CRUD).
FR5	Tracking Kendaraan	Halaman <i>tracking</i> kendaraan telah berjalan dengan baik dan dapat memvisualisasikan pergerakan kendaraan
FR6	Dashboard View	Dashboard berhasil menampilkan statistik aktivitas logistik
FR7	Backend Service untuk Integrasi IoT	Web service telah selesai dikembangkan untuk menerima data dari alat IoT

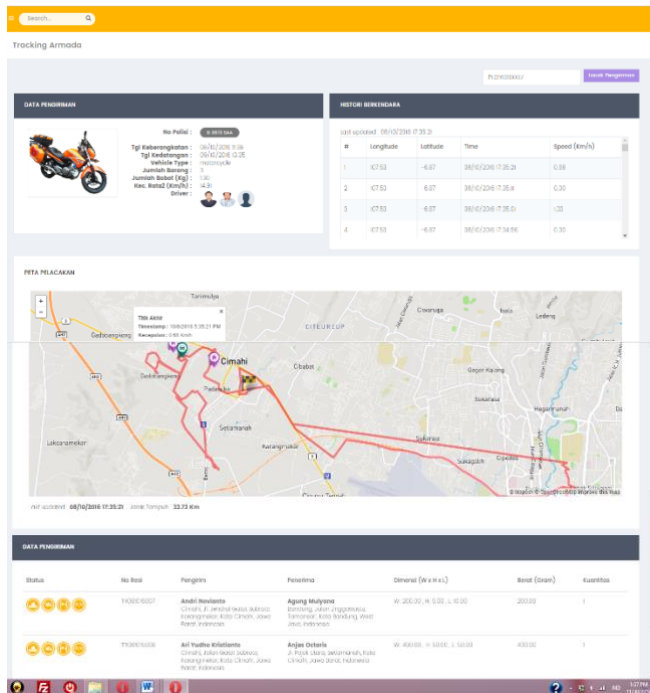
Adapun contoh tampilan halaman web yang telah dikembangkan dapat dilihat pada Gambar 9, Gambar 10, dan Gambar 11.



Gambar 9. Halaman Dashboard

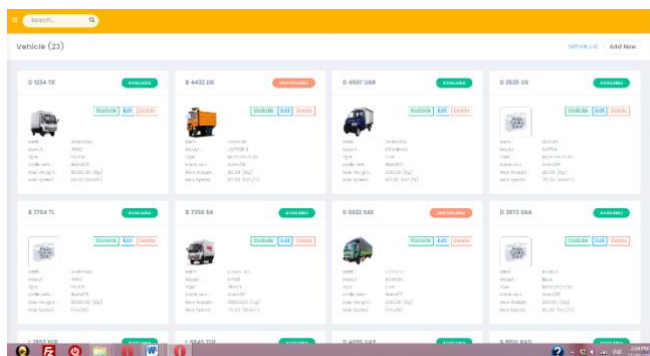
Halaman Dashboard: merupakan halaman yang memberikan informasi terkini mengenai proses logistik dari berbagai modul. Termasuk informasi peta pergerakan armada,

status pengiriman, informasi lalu lintas melalui sosial media, dan berbagai statistik lainnya.



Gambar 10. Halaman Tracking Kendaraan

Halaman Tracking Kendaraan: merupakan halaman yang digunakan untuk melakukan pelacakan pada kendaraan logistik, melihat posisi kendaraan secara real-time, serta historis rute yang dilaluinya. Pada modul tracking ini setiap kendaraan logistik akan dapat divisualisasikan jalur-jalur yang dilaluinya serta lokasi tujuan yang sudah dilewatinya. Sehingga para pelaku logistik dapat memantau pergerakan kendaraannya secara real-time pada sistem.



Gambar 11. Halaman Vehicle untuk Manajemen Kendaraan

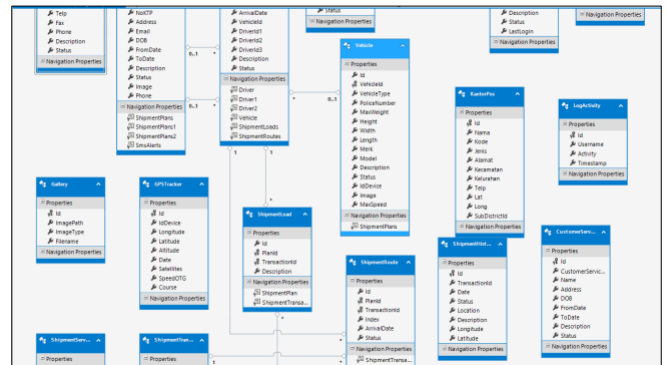
Halaman Vehicle: merupakan halaman yang digunakan untuk pengelolaan kendaraan, baik menambahkan data kendaraan, mengubah, menghapus, maupun pencarian kendaraan. Termasuk juga untuk melihat informasi terkini status available / unavailable.

B. Implementasi ASP.NET MVC

Dalam mengembangkan modul MVC dengan framework ASP.NET, digunakan teknologi *Object Relational Mapping* (ORM). Tools ORM yang digunakan adalah Entity Framework (EF). EF dapat diimplementasikan dengan dua pendekatan, yakni Code First dan Database First. Pada studi

ini, pendekatan yang dilakukan adalah Code First. Pendekatan ini memungkinkan pengembang untuk membuat database dari kelas Model yang dibangun, dan tersinkronisasi secara otomatis pada DBMS seperti MS SQL Server.

Gambar 12 dibawah ini menunjukkan tampilan desain database hasil implementasi ORM.



Gambar 12. Tampilan Desain Database melalui ORM

C. Hasil Pengujian Sistem

Pengujian fungsionalitas sistem telah dilakukan terhadap 36 skenario dengan status pass (berhasil). Dibawah ini adalah contoh beberapa skenario pengujian pada fungsional utama, sebagaimana ditunjukkan pada Tabel V.

TABLE V. HASIL PENGUJIAN FUNGSIONALITAS SISTEM

ID Pengujian	Deskripsi	Keluaran yang diharapkan	Hasil Uji
PG01	Mengisi <i>Username & Password</i> dengan benar	<i>System</i> akan menerima akses <i>Login</i> masuk ke halaman Admin.	<i>Pass</i>
PG02	Klik tombol 'Lacak' pada halaman <i>Maps GPS Tracker</i> dan menampilkan data	<i>System</i> akan menuju halaman <i>Maps GPS Tracker</i> dan menampilkan data	<i>Pass</i>
PG03	Klik tombol 'Vehicle' pada menu	<i>System</i> akan menampilkan Data Statistik Kendaraan	<i>Pass</i>
PG04	Pada Modul Tracking, Mengisi Kode Pengiriman klik tombol 'Lacak Pengiriman'	<i>System</i> menampilkan Keseluruhan Data Transaksi, Vehicle, Lokasi Terkini Vehicle	<i>Pass</i>
PG05	Pada Modul Tracking, pilih kode pengiriman pada daftar tabel	<i>System</i> akan menampilkan hasil pelacakan sesuai kode pengiriman yang diinputkan user	<i>Pass</i>

V. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi web monitoring untuk pelacakan kendaraan logistik menggunakan teknologi Internet of Things (IoT). Sistem ini memanfaatkan beberapa teknologi, yaitu framework ASP.NET MVC, MS SQL Server sebagai database management system, dan Google Map API untuk visualisasi peta. Dengan pemasangan perangkat GPS secara embedded pada kendaraan logistik, sistem dapat mengirim data secara langsung ke server setiap lima detik, memungkinkan pemantauan real-time atas posisi (longitude dan latitude) dan kecepatan kendaraan.

Berdasarkan hasil uji fungsionalitas, integrasi data antara perangkat GPS dan aplikasi web beroperasi dengan baik, sehingga sistem ini dapat dimanfaatkan oleh pelaku logistik untuk mendapatkan informasi terkini mengenai kondisi kendaraan di lapangan, membantu optimalisasi pengambilan keputusan terkait operasional logistik.

Untuk penelitian selanjutnya teknologi IoT yang digunakan dalam penelitian ini adalah alat GPS yang memiliki fitur untuk mengirimkan lokasi kendaraan (longitude dan latitude) serta sensor kecepatan untuk mengukur kecepatan saat berkendaranya saja. Untuk kedepannya, alat IoT dapat dikembangkan dengan fitur yang lebih luas, seperti sensor bahan bakar, sensor suhu, kamera, dll. Sehingga pemanfaatan IoT menjadi lebih maksimal dalam membantu sistem dalam memonitor keseluruhan proses logistik.

UCAPAN TERIMAKASIH

Penulis menyampaikan ucapan terimakasih kepada Unit Penelitian dan Pengabdian kepada Masyarakat (UPPM) Politeknik Negeri Bandung dan Lab SCCIC ITB yang telah memberikan dukungan dan fasilitas selama penelitian berlangsung. Penulis juga menyampaikan ucapan terimakasih kepada Mas Ricky di Lab SCCIC ITB yang telah membantu dalam pengembangan IoT, serta mas Anjas dan mba Wanti yang telah membantu penulis dalam pengembangan aplikasi web.

REFERENCES

- [1] G. Munawar, "THE WEB SERVICE PROTOTYPE ON DELIVERY SYSTEM IN THE IMPLEMENTATION OF ENTERPRISE SERVICE BUS," *Ethos: Jurnal Penelitian dan Pengabdian (Sains & Teknologi)*, vol. 5, no. 1, pp. 63–68, Jan. 2017.
- [2] Kementerian Perhubungan Republik Indonesia, "TINGGINYA BIAYA LOGISTIK DI INDONESIA," Feb. 24, 2012. Accessed: May 05, 2024. [Online]. Available: <https://dephub.go.id/post/read/tingginya-biaya-logistik-di-indonesia-10694>
- [3] Republika, "Tidak Kompetitif, Biaya Logistik Indonesia Tertinggi di Dunia," Feb. 14, 2012. Accessed: Mar. 05, 2024. [Online]. Available: <https://ekonomi.republika.co.id/berita/lzdkvn/tidak-kompetitif-biaya-logistik-indonesia-tertinggi-di-dunia>
- [4] G. V. Ivankova, E. P. Mochalina, and N. L. Goncharova, "Internet of Things (IoT) in logistics," *Materials Science and Engineering*, 2019.
- [5] P. Tadejko, "Application of Internet of Things in Logistics – Current Challenges," *Economics and Management*, vol. 7, no. 4, 2015.
- [6] A. Majeed and I. Rauf, "MVC Architecture: A Detailed Insight to the Modern Web Applications Development," vol. 1.
- [7] A. Freeman, "Pro ASP.NET MVC 5 Platform," in *Pro ASP.NET MVC 5 Platform*, A. Freeman, Ed., Berkeley, CA: Apress, 2014, pp. 3–8. doi: 10.1007/978-1-4302-6542-9_1.
- [8] M. McQuillan, *Introducing SQL Server*. Berkeley, CA: Apress, 2015. doi: 10.1007/978-1-4842-1419-0.
- [9] Google Maps Platform, "Build awesome apps with Google's knowledge of the real world." Accessed: Jun. 01, 2024. [Online]. Available: <https://developers.google.com/maps/>
- [10] D. Sudarsono and M. R. Arief, "Aplikasi Perhitungan Jumlah Kendaraan Dengan Menggunakan Google Maps API".