

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab sebelumnya dijelaskan bahwa alat-alat yang diperlukan berupa perangkat lunak selama pelaksanaan penelitian ini. Penulis mengembangkan perangkat lunak berdasarkan acuan pada panduan *Rational Unified Process* (RUP). Pada *Rational Unified Process* terdapat empat fase pengembangan yang digunakan pada perangkat lunak yaitu fase insepisi, elaborasi, konstruksi, dan transisi. Di tiap fase terdiri atas pemodelan bisnis, kebutuhan, analisis dan desain, implementasi, dan pengujian. Pada bab ini akan membahas proses pengembangan perangkat lunak yang akan digunakan sebagai alat penelitian.

4.2 Fase Insepisi

Fase pertama pada proses pengembangan perangkat lunak adalah melakukan proses identifikasi terhadap sistem yang akan dikembangkan. Pada fase ini mencakup analisis sistem, identifikasi kebutuhan, perumusan kebutuhan pengujian, pemodelan diagram *use case* dan diagram aktivitas.

4.2.1 Pemodelan Bisnis

Prediksi tingkat kemacetan merupakan perhitungan yang digunakan untuk menghitung menentukan apakah jalan tersebut macet

atau tidak. Pada penelitian kali ini yaitu prediksi tingkat kemacetan dengan menggunakan metode *Backpropagation*. Perangkat lunak yang dikembangkan merupakan perangkat lunak yang berbasis desktop yang dapat digunakan untuk prediksi tingkat kemacetan. Perangkat lunak mampu melakukan prediksi tingkat kemacetan secara otomatis ketika dimasukkan data yang akan di coba. Masukan untuk perangkat lunak yang dikembangkan antara lain berupa *file* dengan format *.csv* pada proses pelatihan dan masukan *input* manual melalui *keyboard* pada proses prediksi. Keluaran yang dihasilkan berupa akurasi dari proses prediksi yang dilakukan.

4.2.2 Kebutuhan Sistem

Spesifikasi kebutuhan perangkat lunak prediksi tingkat kemacetan dibangun berdasarkan pada pemodelan bisnis. Fitur-fitur utama yang disediakan perangkat lunak antara lain fitur pelatihan dan juga pengujian.

a. Fitur Pelatihan

Fitur ini digunakan dalam prose pengaturan parameter yang akan di bangun antara lain untuk mengatur *epoch* dan *minimum error* dan juga dimana pola perangkat lunak yang dibangun dan data tingkat kemacetan akan dilatih.

b. Fitur Prediksi

Fitur ini digunakan untuk menguji pola pada perangkat lunak yang dibangun dan menguji prediksi tingkat kemacetan dimana data pengujian kemudian diprediksi dan disimpan.

Dalam perealisasiannya, kebutuhan fungsional dan non-fungsional harus dipenuhi oleh fitur-fitur perangkat lunak. Kebutuhan fungsional menjelaskan bahwa kebutuhan utama perangkat lunak yang dikembangkan atau layanan-layanan sistem dan bergantung kuat terhadap jenis perangkat lunak dan pengguna sistem yang dijelaskan pada tabel IV-1. Sedangkan, kebutuhan non fungsional menjelaskan bahwa kebutuhan penentuan atribut atau kualitas secara keseluruhan suatu sistem dan tidak harus dimiliki oleh tiap-tiap perangkat lunak, dalam artian hanya berfungsi sebagai pelengkap pada perangkat lunak agar lebih baik kinerjanya yang dijelaskan pada tabel IV-2.

Tabel IV-1 Kebutuhan Fungsional

No.	Kebutuhan
1	Perangkat lunak dapat memasukkan parameter dan melatih data pelatihan terhadap data rata-rata kendaraan.
2	Perangkat lunak dapat memprediksi dan mengetahui hasil akurasi dari prediksi kemacetan.

Tabel IV-2 Kebutuhan Non Fungsional

No.	Kebutuhan
1	Antarmuka yang dikembangkan haruslah mudah untuk dipahami oleh pengguna.

4.2.3 Analisis dan Desain

Pada fase insepisi, analisis dan desain merupakan tahapan terpenting untuk memahami kebutuhan perangkat lunak yang tepat dalam menyelesaikan permasalahan penelitian.

4.2.3.1 Analisis Perangkat Lunak

Kegiatan yang dilakukan pada tahapan ini adalah menganalisis kebutuhan perangkat lunak, normalisasi data, pembentukan pola arsitektur jaringan, dan prediksi.

a. Analisis Kebutuhan Perangkat Lunak

Dari pemodelan bisnis yang telah dijabarkan, untuk menyelesaikan permasalahan yang terjadi diperlukan perangkat lunak yang mampu memprediksi tingkat kemacetan secara otomatis. Setiap hari data rata-rata volume kendaraan semakin sulit untuk diprediksi secara manual sehingga menyulitkan untuk menentukan tingkat kemacetan. Kondisi tersebut membuat perangkat lunak harus memiliki kemampuan sebagai berikut.

1. Normalisasi data masukan;

2. Mengatur parameter untuk pola arsitektur jaringan yang akan dibuat;
3. Melatih data masukan;
4. Memprediksi tingkat kemacetan;
5. Menghasilkan akurasi prediksi dari data yang telah dimasukkan ke dalam perangkat lunak..

Pengembangan perangkat lunak dimulai dengan mengambil data sekunder rata-rata jumlah kendaraan di Dinas Perhubungan dan Transportasi Kota Palembang. Selanjutnya data yang didapat disimpan dalam format *.csv*. Tahap selanjutnya adalah melakukan proses normalisasi terhadap data yang didapat. Proses normalisasi adalah proses untuk mengambil hanya data yang dibutuhkan sehingga data yang diproses lebih sesuai. Setelah normalisasi data maka data siap untuk di olah

b. Analisis Data

Data yang dibutuhkan perangkat lunak yang dikembangkan yaitu data Trafik Kendaraan yang berguna sebagai masukan pada proses pelatihan dan pengujian algoritma *Backpropagation*. Data ini terdiri dari 4 variabel yaitu variable minggu 1, minggu 2, minggu 3, minggu 4. Data didapatkan dari Dinas Perhubungan dan Transportasi Kota Palembang dengan interval data diteliti perminggu untuk

kemudian dianalisis dan dibandingkan pada rentang data minggu selanjutnya. Data tersebut disimpan dalam format *.csv* dan dikategorikan menjadi 2 kategori, yaitu data pelatihan dan data target, Data yang digunakan adalah data Trafik Kendaraan dengan 1 tahun periode yaitu tahun 2015. Tabel 4.3 menunjukkan contoh data 4 variabel Trafik Kendaraan yang digunakan pada penelitian ini.

Tabel 4.3 Contoh Data 4 variabel Trafik Kendaraan (Sumber: Dinas Perhubungan dan Transportasi Kota Palembang).

Jalan	Bulan 1			
	Minggu 1	Minggu 2	Minggu 3	Minggu 4
Basuki Rahmat (Patal)	25343	24537	24408	23219
Basuki Rahmat (Polda)	22341	23271	21965	23859
A. Rivai (Charitas)	22081	23056	22965	23061
A. Rivai (Bukit)	26882	23349	23375	23107
Kol. H. Burlian (Polda)	39691	34276	35333	38537
Kol. H. Burlian (AAL)	30404	26238	28735	28588
Demang Lebar Daun (Polda)	21719	18044	21068	17874
Demang Lebar Daun (Bukit Besar)	15558	21139	17054	13996
Jendral Sudirman (RS Charitas)	42263	39315	36159	42440
Jendral Sudirman (Polda)	29711	31506	26122	28397
Veteran (RS. Charitas)	15541	15654	12379	14200
Veteran (Rajawali)	17038	14299	13655	13729

c. Analisis Normalisasi Data

Pada tahap ini data akan dirubah dengan rentang nilai data menjadi bilangan berkoma. Tujuan perubahan ini adalah agar data mudah dikenali polanya oleh JST pada fase pelatihan

dan pengujian. Adapun persamaan untuk normalisasi data sudah dijelaskan pada persamaan II.20. Tabel IV.4 menunjukkan contoh data 4 variabel Trafik Kendaraan yang telah dinormalisasi

Tabel 4.4 Contoh Data 4 variabel Trafik Kendaraan yang telah di normalisasi

Jalan	Bulan 1			
	Minggu 1	Minggu 2	Minggu 3	Minggu 4
Basuki Rahmat (Patal)	0,366814	0,40925807	0,505845	0,330535
Basuki Rahmat (Polda)	0,254472	0,35865046	0,403112	0,352826
A. Rivai (Charitas)	0,244742	0,35005596	0,445164	0,325032
A. Rivai (Bukit)	0,424407	0,36176847	0,462405	0,326634
Kol. H. Burlian (Polda)	0,90375	0,79856892	0,965265	0,864059
Kol. H. Burlian (AAL)	0,556208	0,47725456	0,687805	0,517536
Demang Lebar Daun (Polda)	0,231195	0,14970419	0,365391	0,144369
Demang Lebar Daun (Bukit Besar)	0,000636	0,27342501	0,196594	0,009299
Jendral Sudirman (RS Charitas)	1	1	1	1
Jendral Sudirman (Polda)	0,530275	0,68783978	0,577923	0,510884
Veteran (RS. Charitas)	0	0,05416533	0	0,016404
Veteran (Rajawali)	0,056021	0	0,053659	0

4.2.3.2 Konsep Pengujian Data

Pada tahap ini, menjelaskan bentuk pengujian yang perlu dilakukan. Berdasarkan dataset yang digunakan, berjumlah 576 data yang dibagi masing-masing menjadi 540 data latih dan 36 data uji dengan parameter berupa epoch dan eror. Epoch dan eror yang digunakan pada tiap percobaan akan bervariasi. Percobaan dilakukan beberapa kali untuk mengukur tingkat keakuratan yang dihasilkan oleh sistem, sesuai dengan data aslinya. Proses pengujian dilakukan sebanyak 40 kali percobaan

dengan percobaan pertama sebanyak 15 kali dengan menggunakan *Backpropagation* standar dan percobaan kedua sebanyak 25 kali dengan adanya variasi nilai eror dan epoch, untuk melihat pengaruh eror dan epoch terhadap akurasi yang dihasilkan oleh sistem.

4.2.3.3 Desain Perangkat Lunak

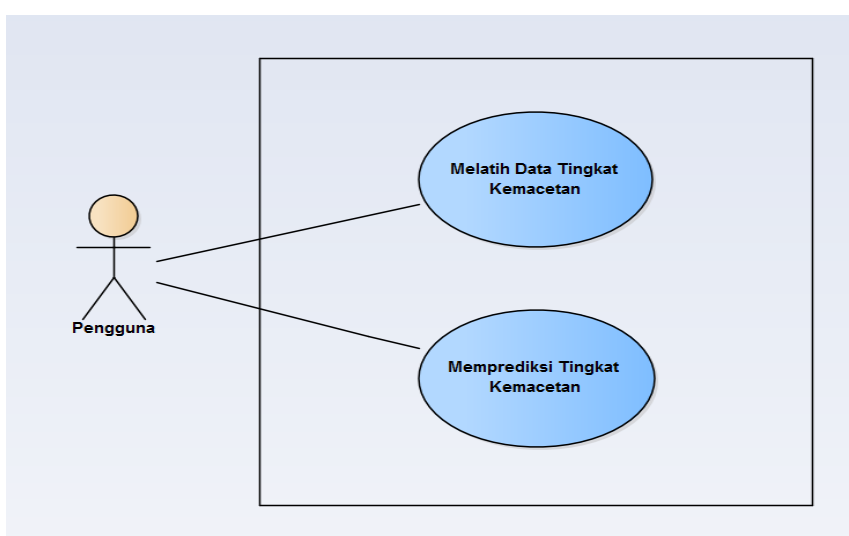
Desain dari perangkat lunak diilustrasikan ke dalam diagram *use case* dan diagram aktivitas.

1. Model *Use Case*

Pada subbab ini dijelaskan ilustrasi fungsionalitas yang dimiliki perangkat lunak yang dibangun dengan menggunakan pemodelan *Use Case*.

a) Diagram *Use Case*

Diagram *Use Case* menjelaskan kegiatan yang berhubungan oleh aktor (pengguna) terhadap perangkat lunak yang dikembangkan, dapat dilihat pada gambar IV.1



Gambar IV-1 Diagram *Use Case*

b) Tabel Definisi Aktor

Penelitian ini, aktor (pengguna) merupakan seseorang yang berinteraksi dengan system yang dijelaskan pada tabel IV-5.

Tabel IV-5 Definisi Aktor *Use Case*

Nomor	Use Case	Definisi
1	Pengguna	Pengguna merupakan orang yang berhubungan dengan perangkat lunak aplikasi untuk memasukkan data, mengatur parameter, melatih data dan memprediksi tingkat kemacetan.

c) Tabel Definisi *Use Case*

Definisi *Use case* yang akan dijelaskan pada tabel IV-6 merupakan penjelasan dari kerja perangkat lunak yang spesifik disimbolkan dengan sebuah notasi lingkaran yang agak lonjong atau berbentuk oval pada gambar IV-1.

Tabel IV-6 Definisi *Use Case*

Nomor	<i>Use Case</i>	Deskripsi
1	Melatih Data tingkat kemacetan	Kegiatan mengatur parameter yang akan digunakan untuk proses pelatihan serta memasukkan file data latih yang telah ditentukan untuk dilakukan proses pelatihan data berupa data rata-rata kendaraan per minggu setiap bulan.

2	Memprediksi tingkat kemacetan	Kegiatan memasukkan file data target, yaitu data yang digunakan sebagai perbandingan data yang akan diprediksi dan kemampuan program dalam menguji data rata-rata kendaraan per minggunya.
---	-------------------------------	--

d) Skenario *Use Case*

Skenario merupakan urutan yang bersifat spesifik berasal dari aksi dan interaksi antara aktor dan sistem. Berikut ini adalah skenario dari *use case* yang telah didefinisikan pada subbab sebelumnya.

1. Skenario *Use Case* Melatih Data rata-rata kendaraan

No	: 001
Nama <i>Use Case</i>	: Melatih Data rata-rata kendaraan
Aktor	: Pengguna
Tujuan	: Mengatur nilai parameter dan melatih data pelatihan rata-rata kendaraan.
Deskripsi	: <i>Use Case</i> ini menggambarkan interaksi antara system dan aktor dalam melakukan proses pelatihan dengan data masukan dalam bentuk <i>file excel 2015 (.csv)</i> .
Kondisi Awal	: Sistem aktif
Kondisi Akhir	: Data sudah dilatih

Skenario *use case* Melatih Data rata-rata kendaraan dijelaskan pada tabel

IV-7 .

Tabel IV-7. Skenario *Use Case* Melatih Data Rata-rata kendaraan.

Skenario Utama	
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pengguna menekan tombol Pelatihan.	
	2. Sistem menampilkan <i>form</i> Form Pelatihan.
3. Pengguna memasukkan data parameter yaitu nilai maximum epoch dan minimum error.	
4. Pengguna menekan tombol "Browse".	
	5. Sistem menampilkan direktori pencarian data
6. Pengguna memilih <i>file</i> data pelatihan dalam bentuk <i>file Excel 2015 (.csv)</i> .	
	7. Sistem menampilkan data pelatihan.

8. Pengguna menekan tombol mulai pelatihan	
	9. Sistem melakukan pelatihan data menggunakan metode <i>Backpropagation</i>
	10. Sistem menyimpan bobot
	11. Sistem menampilkan form Pelatihan selesai

2. Skenario *Use Case* Memprediksi Tingkat Kemacetan

No.	: 002
Nama <i>Use Case</i>	: Memprediksi tingkat kemacetan
Aktor	: Pengguna.
Tujuan	: Menguji data terhadap data pengujian untuk mendapatkan hasil prediksi dari tingkat kemacetan
Deskripsi	: <i>Use Case</i> ini mendeskripsikan interaksi antara sistem dan aktor dalam melakukan proses pengujian terhadap prediksi tingkat kemacetan
Kondisi Awal	: Data awal rata-rata kendaraan
Kondisi Akhir	: Memberikan kesimpulan dari prediksi terhadap tingkat kemacetan.

Skenario *use case* Memprediksi Tingkat Kemacetan dijelaskan pada tabel

IV-8 .

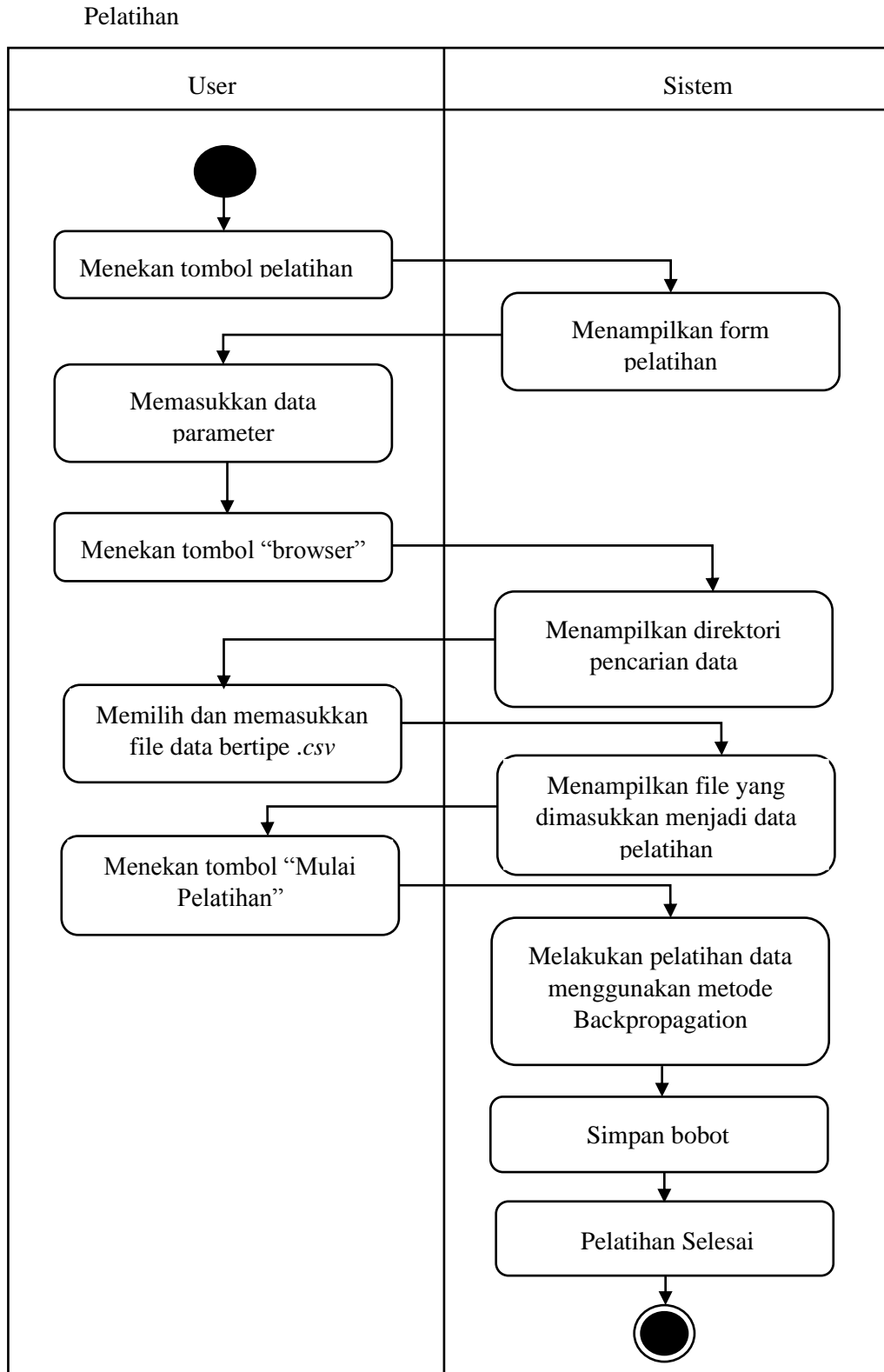
Tabel IV-8 Skenario *Use Case* Melakukan Prediksi Tingkat Kemacetan
Skenario Utama

Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol “Prediksi”.	
	2. Sistem menampilkan <i>form</i> Prediksi.
3. Pengguna menekan tombol browse	
	4. Sistem menampilkan direktori pencarian data
5. Pengguna memilih <i>file</i> data target prediksi dalam bentuk <i>file Excel 2015 (.csv)</i> .	
	6. Sistem menampilkan data pengujian.
7. Pengguna menekan tombol “Bobot v”	
	8. Sistem membaca bobot v dan menampilkan pesan “Bobot V Berhasil Di

	Load”
9. Pengguna menekan tombol “BobotW”	
	10. Sistem membaca bobot W dan menampilkan pesan “Bobot W Berhasil Di Load”
11. Pengguna menekan tombol “Proses”	
	12. Sistem melakukan perhitungan menggunakan metode backpropagation dalam proses prediksi
	13. Sistem menampilkan menampilkan hasil prediksi dan akurasi

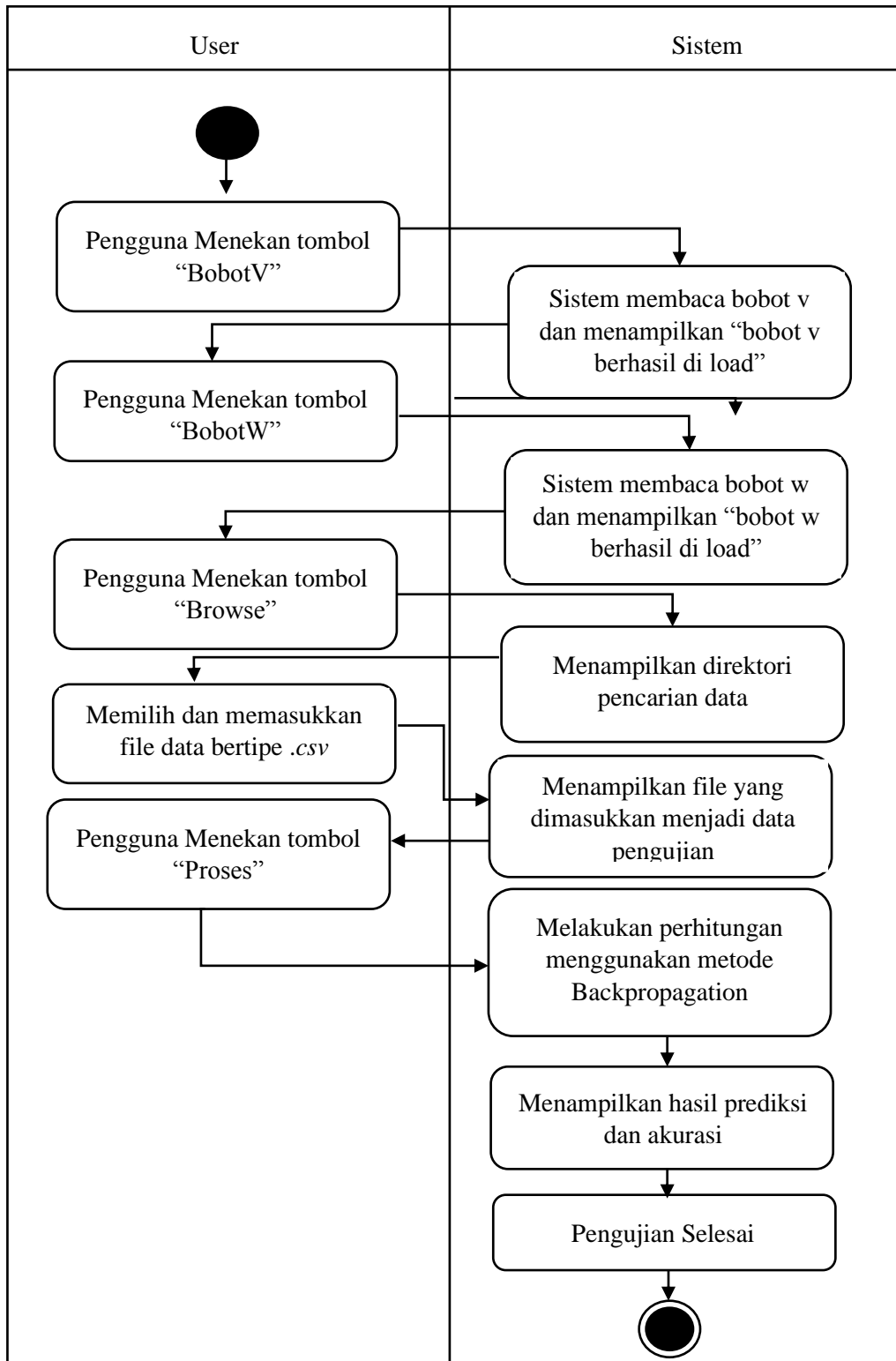
2. Diagram Aktivitas

Diagram aktivitas menggambarkan alur kerja dari sebuah sistem atau proses bisnis. Diagram aktivitas melatih data rata-rata kendaraan dapat dilihat pada gambar IV.2, diagram memprediksi tingkat kemacetan dapat dilihat pada gambar IV.3.



Gambar IV-2 Diagram Aktivitas Use Case Melatih Data Rata-rata kendaraan

Pengujian



Gambar IV-3 Diagram Aktivitas Use Case Memprediksi tingkat kemacetan

4.3 Fase Elaborasi

Fase kedua dalam pengembangan perangkat lunak ini adalah melakukan proses identifikasi pada sistem yang dikembangkan. Aktivitas yang terlibat mencakup pada perancangan data, perancangan antarmuka, identifikasi kebutuhan, perumusan kebutuhan pengujian, pemodelan diagram sequence, dan pembuatan dokumentasi.

4.3.1 Pemodelan Bisnis

Di subbab ini akan dibahas mengenai proses perancangan perangkat lunak yang akan dibangun. Perancangan dilakukan sesuai dengan hasil analisa yang telah dilaksanakan pada fase insepisi. Perancangan yang dibahas pada subbab ini meliputi perancangan data dan perancangan antar muka.

4.3.1.1 Perancangan Data

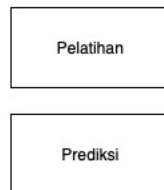
Perangkat lunak yang akan dikembangkan memiliki kemampuan prediksi terhadap tingkat kemacetan Adapun data yang akan melalui proses prediksi adalah data rata-rata kendaraan yang disimpan dalam file berformat *.csv*. Adapun data yang digunakan yaitu 4 variabel data rata-rata kendaraan antara lain minggu 1, minggu 2, minggu 3 dan minggu 4 berupa angka berbentuk bilangan bulat.

4.3.1.2 Perancangan Antar Muka

Di subbab ini membahas tentang perancangan *interface* dari perangkat lunak yang dikembangkan. Adapun rancangan antarmuka Menu

Utama digambarkan pada gambar IV-3 , rancangan antarmuka pelatihan pada gambar IV-4, rancangan antarmuka prediksi pada gambar IV-5

Prediksi Tingkat Kemacetan Menggunakan Metode Backpropagation



Gambar IV-4 Rancangan Antarmuka Form Menu

Gambar IV-4 merupakan halaman utama yang memiliki fungsi untuk melakukan prediksi tingkat kemacetan. Terdapat dua bagian pada antarmuka ini yaitu tombol untuk melakukan proses pelatihan dan prediksi.

Prediksi Tingkat Kemacetan Menggunakan Metode Backpropagation

Maximum Epoch
Minimum Error

Nama Jalan	<u>Minggu 1</u>	<u>Minggu 2</u>	<u>Minggu 3</u>	<u>Minggu 4</u>

Gambar IV-5 Rancangan Antarmuka Menu Pelatihan

Gambar IV-5 merupakan halaman pelatihan rata-rata jumlah kendaraan. Pada antarmuka ini disediakan tombol untuk memilih data dan tombol untuk memulai pelatihan.

Prediksi Tingkat Kemacetan Menggunakan Metode *Backpropagation*

Hasil Real

Nama Jalan	Minggu 1	Minggu 2	Minggu 3	Minggu 4

Hasil Sistem

Nama Jalan	Minggu 1	Minggu 2	Minggu 3	Minggu 4

Gambar IV-6 Rancangan Antarmuka Menu Prediksi

Gambar IV-6 merupakan halaman prediksi tingkat kemacetan. Pada antarmuka ini tersedia tombol untuk memasukkan data target, melakukan prediksi dan akan memunculkan secara otomatis hasil prediksi berupa Minggu 1, Minggu 2, Minggu 3 dan Minggu 4 prediksi serta tingkat akurasi dari tingkat kemacetan setiap minggunya.

4.3.2 Kebutuhan Sistem

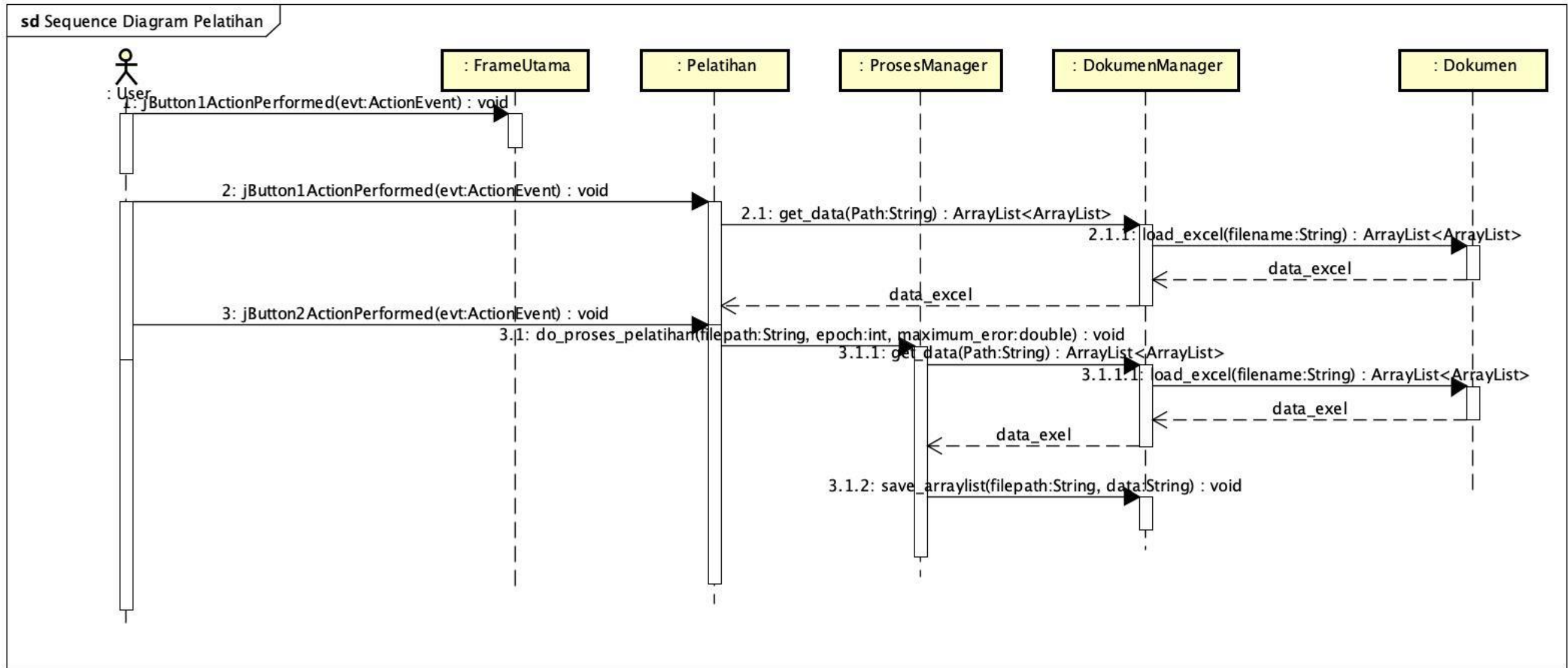
Subbab ini akan menjelaskan kebutuhan sistem dari perangkat lunak yang dibangun berdasarkan hasil analisis dan perancangan pada tahap selanjutnya. Untuk membangun perangkat lunak dalam penelitian ini dibutuhkan perangkat keras (*hardware*), perangkat lunak (*software*) dan bahasa pemrograman. Bahasa pemrograman yang digunakan untuk

implementasi perangkat lunak adalah Java. Perangkat keras yang digunakan pada tahap pengembangan dan penelitian ini adalah laptop dengan spesifikasi sebagai berikut:

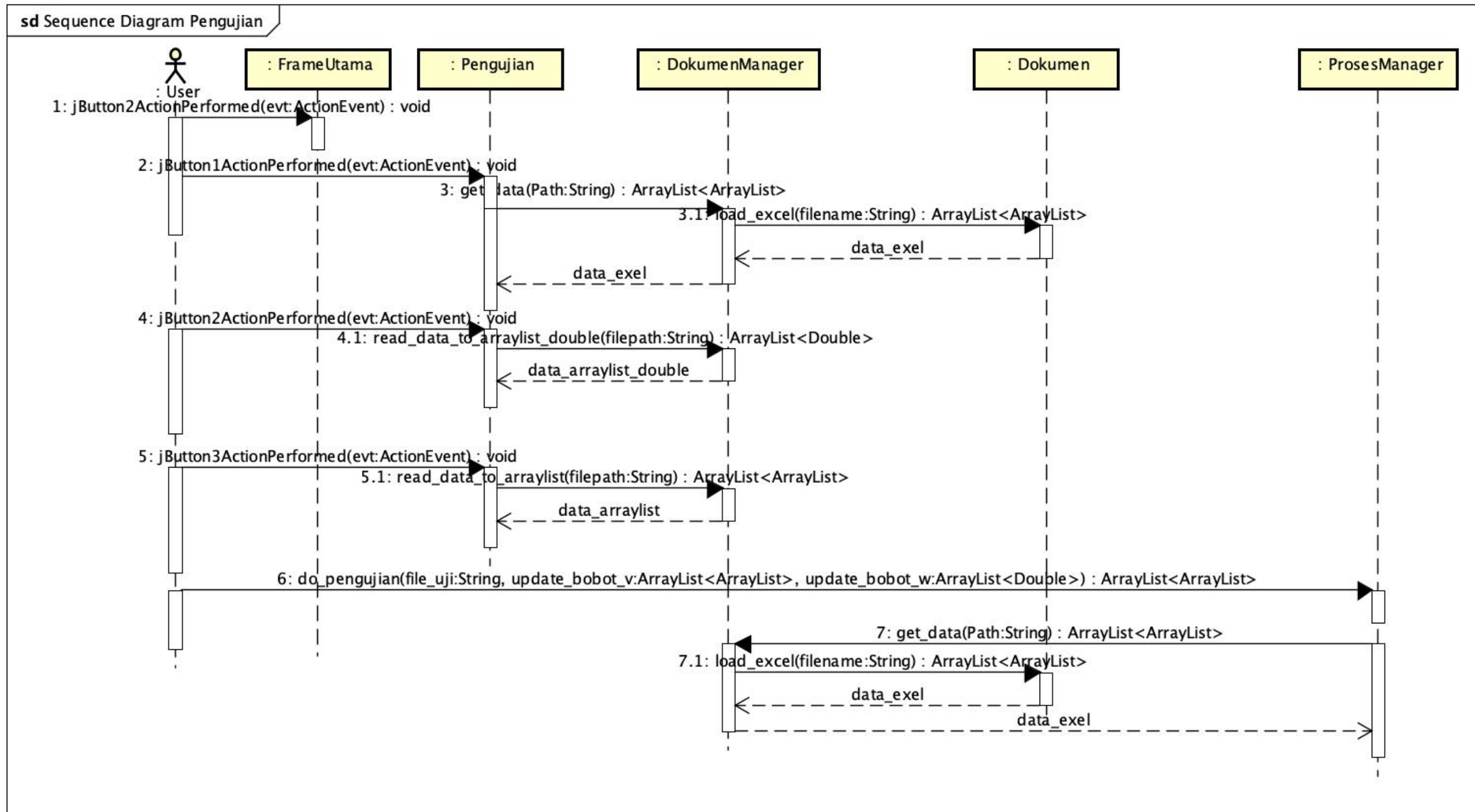
1. Laptop HP Helwett;
2. Processor Intel(R) Core(TM) i7 – 5500U @ 2.40 GHz 2.40 GHz;
3. RAM 8 GB ;
4. *Hard Disk* 455 GB .

4.3.3 Diagram Sequence

Diagram *sequence* adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek. Berdasarkan *use case* yang telah dibentuk, diagram sequence yang dibentuk pada pengembangan perangkat lunak ini berjumlah empat buah. Diagram sequence melatih data rata-rata jumlah kendaraan dapat dilihat pada gambar IV.6, diagram sequence memprediksi tingkat kemacetan dapat dilihat pada gambar IV.7



Gambar IV-7 Diagram *Sequence* Melatih Data rata-rata kendaraan



Gambar IV-8 Diagram *Sequence* Memprediksi Tingkat Kemacetan

4.4 Fase Konstruksi

Fase ini berfokus pada pengembangan perangkat lunak baik komponen utama maupun fitur-fitur pendukung dengan melakukan sederet iterasi. Pada setiap iterasi terdapat proses analisa, desain, implementasi, dan pengujian. Dalam proses pengembangannya dapat menggunakan konstruksi paralel agar mempercepat hasil perangkat lunak. Hasil yang diharapkan dari fase ini adalah sebuah produk perangkat lunak yang siap digunakan oleh *end-user*, yaitu sebuah produk perangkat lunak yang dapat digunakan sebagai alat penelitian.

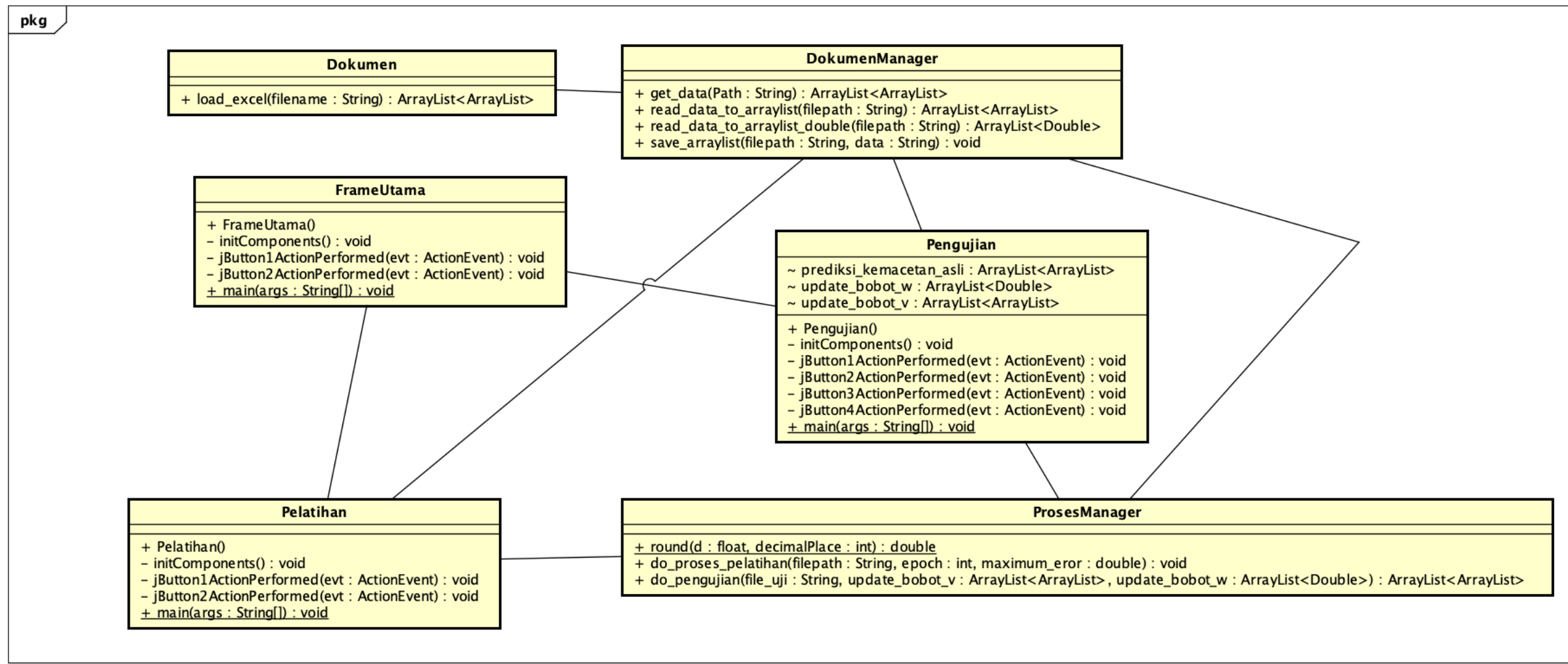
4.4.1 Kebutuhan Sistem

Dalam pengembangan perangkat lunak ini, perangkat lunak yang digunakan untuk implementasi yaitu:

1. Sistem operasi Windows 7 Home Premium 64-bit ;
2. *Compiler* Netbeans IDE 8.2 .
3. Visual Paradigm For Uml 8.0 Enterprise Edition.

4.4.2 Diagram Kelas

Diagram kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain. Terdapat serta dimasukkan pula atribut dan operasi. Terdapat 5 kelas yang terdiri dari 1 kelas *entity* (kelas Dokumen), 1 kelas *boundary* (kelas FrameUtama), dan 4 kelas *control* (Pelatihan, Pengujian, ProsesManager, DokumenManager). Gambar IV-9 menunjukkan hubungan antar kelas tersebut.



Gambar IV-9 Diagram Kelas Perangkat Lunak

4.4.3 Implementasi

Fase implementasi dalam konstruksi adalah mengembangkan perangkat lunak berdasarkan diagram kelas dan rancangan antarmuka yang telah dibuat dalam fase sebelumnya.

4.4.3.1 Implementasi Kelas

Kelas – kelas yang telah dirancang pada diagram kelas diimplementasikan dalam bahasa pemrograman Java. Tabel (IV–9) menunjukkan implementasi kelas dalam bahasa Java.

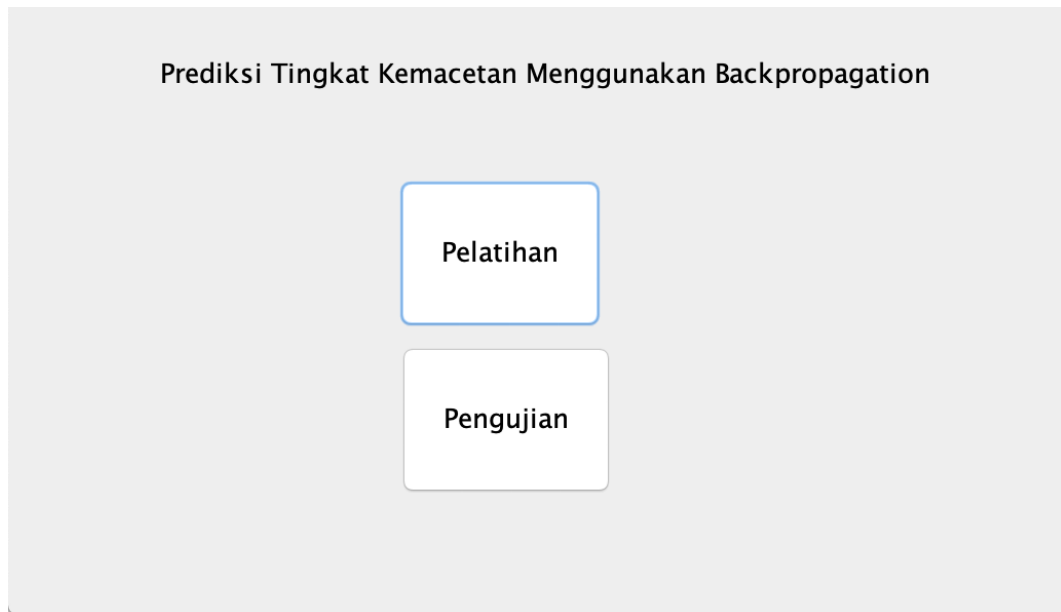
Tabel IV–9 Implementasi Kelas

No	Nama Kelas	Nama File	Keterangan
1.	FrameUtama	FrameUtama.java	Kelas FrameUtama merupakan kelas <i>boundary</i> yang menampilkan tampilan utama dari program.
2.	Pelatihan	Pelatihan.java	Kelas Pelatihan Merupakan kelas <i>boundary</i> yang menampilkan form pelatihan.
3.	Pengujian	Pengujian.java	Kelas Pengujian merupakan kelas <i>boundary</i> yang menampilkan form pengujian.
4.	ProsesManager	ProsesManager.java	Kelas ProsesManager

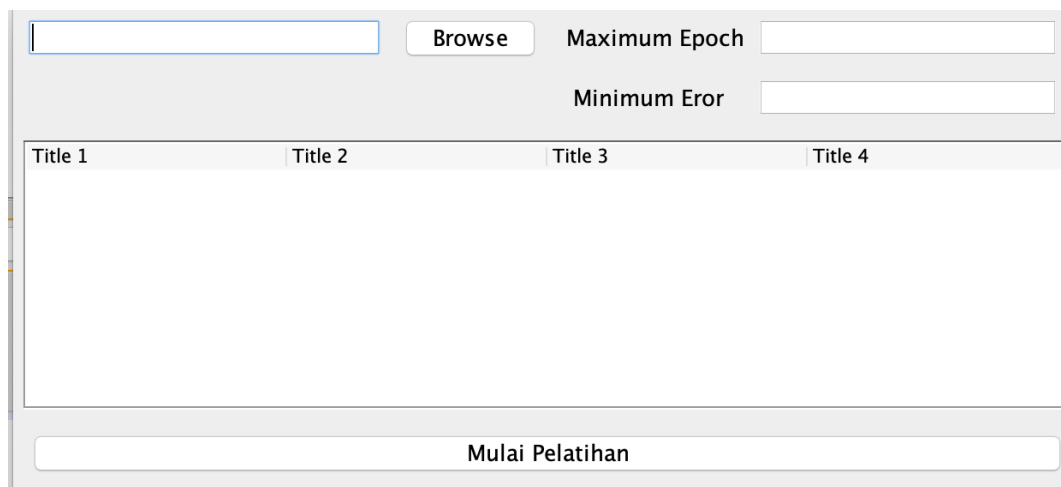
			merupakan kelas <i>controller</i> yang digunakan dalam proses prediksi Tingkat Kemacetan menggunakan metode <i>Backpropagation</i>
5.	DokumenManager	DokumenManager.java	Kelas DokumenManager merupakan kelas <i>controller</i> yang digunakan dalam proses melakukan Pengambilan data.
6.	Dokumen	Dokumen.java	Kelas Dokumen merupakan kelas <i>Entity</i> yang digunakan dalam proses pengambilan data dari exel.

4.4.3.2 Implementasi Antarmuka

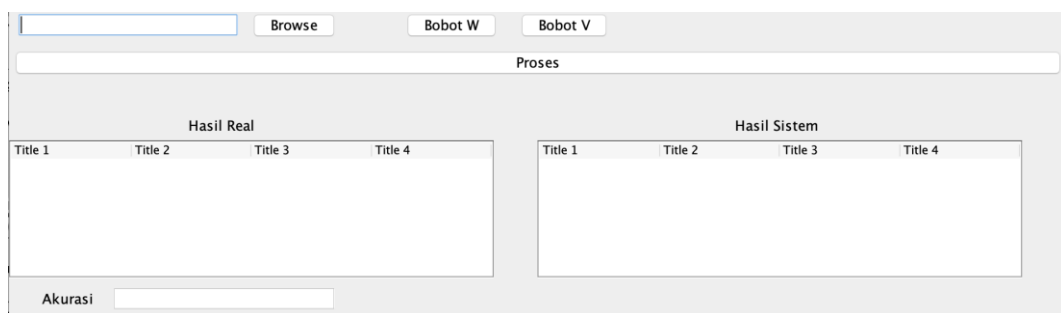
Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka pada fase elaborasi. Gambar IV-10 merupakan antarmuka halaman utama perangkat lunak, gambar IV-11 merupakan antarmuka pelatihan data, gambar IV-12 merupakan antarmuka prediksi.



Gambar IV-10 Antarmuka Halaman Menu Utama



Gambar IV-11 Antarmuka Pelatihan Data Rata-rata kendaraan



Gambar IV-12 Antarmuka Prediksi Tingkat Kemacetan

4.5 Fase Transisi

Pada subbab ini dibahas mengenai pengujian dari perangkat lunak prediksi tingkat kemacetan. Pengujian dilakukan berdasarkan perangkat lunak hasil pengembangan di fase konstruksi.

4.5.1 Pemodelan Bisnis

Pengujian perangkat lunak secara *black box* dan *white box* dengan terlebih dahulu membuat rencana pengujian berdasarkan *Use Case* yang dibuat pada fase insepisi.

4.5.2 Kebutuhan Sistem

Lingkungan pengujian yang digunakan pada fase transisi adalah perangkat keras yang sama saat membangun perangkat lunak prediksi tingkat kemacetan dengan spesifikasi sebagai berikut:

1. Laptop HP Helwett;
2. Processor Intel(R) Core(TM) i7 – 5500U @ 2.40 GHz 2.40 GHz;
3. RAM 8 GB ;
4. *Hard Disk* 455 GB .

Sedangkan perangkat lunak yang digunakan untuk implementasi yaitu:

1. Sistem operasi Windows 8.1 64-bit ;
2. *Compiler* Netbeans IDE 6.4.
3. Visual Paradigm For Uml 8.0 Enterprise Edition.
4. Balsamiq Mockup 3

4.5.3 Rencana Pengujian

Rencana pengujian pada perangkat lunak prediksi tingkat kemacetan digambarkan dalam tabel-tabel. Kolom pada tabel meliputi identifikasi, pengujian, jenis pengujian, serta tingkat pengujian.

4.5.3.1 Rencana Pengujian Use Case Melatih Data Rata-Rata Kendaraan

Rencana pengujian melatih data rata-rata kendaraan pada perangkat lunak berdasarkan *use case*.

Tabel IV–10 Rencana Pengujian Use Case Melatih Data Rata-Rata Kendaraan.

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U – 1 – 101	Mengisi data nilai parameter secara lengkap.	<i>Black Box</i>	Pengujian Unit
2.	U – 1 – 102	Mengisi data parameter secara tidak lengkap dan bukan berupa angka.	<i>Black Box</i>	Pengujian Unit
3.	U – 1 – 103	Memasukkan data pelatihan dengan <i>file Excel 2015 (.csv)</i> .	<i>Black Box</i>	Pengujian Unit
4.	U – 1 – 104	Memasukkan data pelatihan selain <i>file Excel 2015 (.csv)</i> .	<i>Black Box</i>	Pengujian Unit

5.	U – 1 – 105	Menekan tombol “Mulai Pelatihan”.	<i>Black Box</i>	Pengujian Unit
6.	U – 1 – 106	Melakukan keseluruhan data latih rata-rata kendaraan	<i>White Box</i>	Pengujian Unit

4.5.3.2 Rencana Pengujian Use Case Memprediksi Tingkat Kemacetan

Rencana pengujian memprediksi Tingkat Kemacetan pada perangkat lunak berdasarkan *usecase*.

Tabel IV–11 Rencana Pengujian *Use Case* Memprediksi Tingkat Kemacetan

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U – 2– 101	Memasukkan data prediksi dengan <i>file</i> <i>Excel 2015 (.csv)</i>	<i>Black Box</i>	Pengujian Unit
2.	U – 2 – 102	Memasukkan data prediksi selain <i>file</i> <i>Excel 2015 (.csv)</i> .	<i>Black Box</i>	Pengujian Unit
3.	U – 2– 103	Menekan tombol Bobot V	<i>Black Box</i>	Pengujian Unit
4.	U – 2 – 104	Menekan tombol Bobot W	<i>Black Box</i>	Pengujian Unit

5.	U – 2 – 105	Menekan tombol “Proses”.	<i>Black Box</i>	Pengujian Unit
6.	U – 2 – 106	Melakukan proses pengujian prediksi	<i>White Box</i>	Pengujian Unit

4.5.4 Implementasi

Berikut ini adalah kasus uji yang dilakukan terhadap perangkat lunak yang dibangun. Kasus uji dilakukan berdasarkan rencana uji yang telah dipaparkan sebelumnya.

4.5.4.1 Pengujian Use Case Melatih Data

Tabel IV-12 menerangkan pengujian melatih data Rata-Rata Kendaraan pada perangkat lunak berdasarkan Use Case.

Tabel IV-12. Pengujian Use Case Melatih Data Rata-Rata kendaraan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U - 1 - 101	Mengisi data parameter secara lengkap.	- Mengisi semua data parameter secara lengkap.	Maximum epoch = 10000 dan minium error = 0,001	Data parameter yang diinputkan ditampung.	Data parameter yang diinputkan ditampung sementara.	Terpenuhi
U - 1 - 102	Mengisi data parameter secara tidak lengkap dan bukan berupa	- Tidak Mengisi semua parameter secara lengkap.	Maximum epoch = 1000 dan minimum eror	- Tampil pesan <i>empty string</i> . - Pelatihan tidak dapat dijalankan	- Tampil pesan <i>empty string</i> . - Pelatihan tidak dapat dijalankan	Terpenuhi

	angka.		kosong			
U - 1 - 103	Memasukkan data pelatihan dengan <i>file Excel 2015 (.csv)</i> .	Menekan tombol "Browse".	Data rata-rata jumlah kendaraan <i>Excel 2015 (.csv)</i> .	Menampilkan rata-rata jumlah kendaraan <i>file excel 2015 (.csv)</i> yang dipilih.	Menampilkan data rata-rata jumlah kendaraan <i>excel 2015 (.csv)</i> yang dipilih.	Terpenuhi
U - 1 - 104	Memasukkan data pelatihan selain <i>file Excel 2015 (.csv)</i> .	Menekan tombol "Browse".	Memasukkan File Data selain <i>file Excel 2015 (.csv)</i> .	Menampilkan file saja, tidak terdapat data yang akan dimasukkan.	Menampilkan file saja, tidak terdapat data yang akan dimasukkan.	Terpenuhi
U - 1 - 105	Menekan tombol "Mulai	Menekan tombol "Mulai Pelatihan".	Table data rata-rata jumlah	-Menampilkan pesan "Pelatihan selesai".	-Menampilkan pesan "Pelatihan	Terpenuhi

	Pelatihan”.		kendaraan		selesai -	
U - 1 - 106	Melakukan keseluruhan data latihan rata-rata jumlah kendaraan	Melatih data rata-rata jumlah kendaraan	Masukan berupa file berformat .csv	Data berhasil dilatih dan mendapatkan bobot yang akan digunakan pada proses prediksi	Data berhasil dilatih dan mendapatkan bobot yang akan digunakan pada proses prediksi	Terpenuhi

4.5.4.2 Pengujian Use Case Memprediksi Tingkat Kemacetan

Tabel IV-11 menerangkan pengujian memprediksi Tingkat Kemacetan pada perangkat lunak berdasarkan Use Case.

Tabel IV-11. Pengujian Use Case Memprediksi Tingkat Kemacetan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U - 2 - 101	Memasukkan data	Menekan tombol	Data rata-rata	Menampilkan	Menampilkan data	Terpenuhi

	prediksi dengan <i>file Excel 2015 (.csv).</i>	“Browse”.	jumlah kendaraan <i>file Excel 2015 (.csv).</i>	data rata-rata jumlah kendaraan <i>file excel 2015 (.csv) yang</i> dipilih.	rata-rata jumlah kendaraan excel 2015 (.csv) yang dipilih.	
U – 2 – 102	Memasukkan data pelatihan selain <i>file Excel 2015 (.csv).</i>	Menekan tombol “Browse”.	memasukan <i>file selain Excel 2015 (.csv).</i>	- Tampil pesan <i>empty string.</i> - Pelatihan tidak dapat dijalankan	- Tampil pesan <i>empty string.</i> - Pelatihan tidak dapat dijalankan	Terpenuhi
U – 2 - 103	Menekan tombol “Bobot V”	Menekan tombol “Bobot V”		- Melakukan proses Load bobot V	Melakukan proses Load bobot V	Terpenuhi
U – 2 – 104	Menekan tombol “Bobot W”	Menekan tombol “Bobot W”		- Melakukan proses Load	Melakukan proses Load bobot W	Terpenuhi

				bobot W		
U - 2 - 105	Menekan tombol "Proses".	Menekan tombol "Proses".	-	- Melakukan proses prediksi tingkat kemacetan, - Menampilkan Hasil Prediksi dan tingkat akurasi.	- Melakukan proses prediksi tingkat kemacetan. - Menampilkan Hasil Prediksi dan tingkat akurasi.	Terpenuhi
U - 2 - 106	Menekan proses pengujian prediksi	Menekan proses pengujian prediksi	-	- Melakukan proses prediksi tingkat kemacetan, - Menampilkan	- Melakukan proses prediksi tingkat kemacetan. - Menampilkan Hasil Prediksi dan	Terpenuhi

				Hasil Prediksi dan tingkat akurasi.	tingkat akurasi.	
U - 2 - 107	Menekan tombol "Simpan"	- Menekan tombol "Simpan".	-Memilih direktori untuk lokasi penyimpanan file	- Data yang Diprediksi tersimpan.	Data yang Diprediksi tersimpan. - ..	Terpenuhi

4.6 Kesimpulan

Dalam pengembangan perangkat lunak yaitu media penelitian mengikuti metode RUP yang terdiri dari empat fase yaitu fase inepsi, elaborasi, konstruksi dan transisi. Fase inepsi melakukan identifikasi terhadap perangkat lunak yang dikembangkan, dengan aktivitas analisis sistem, identifikasi kebutuhan, perumusan kebutuhan pengujian, pemodelan diagram *use case* dan diagram aktivitas. *Use case* terdiri dari mengatur melatih data rata-rata jumlah kendaraan, dan melakukan prediksi.

Untuk perancangan antarmuka dan diagram sequence dilaksanakan pada fase elaborasi, dengan mengikuti pemodelan bisnis use case yang dilakukan pada fase sebelumnya. Fase konstruksi merupakan fase ketika melakukan proses koding dengan menjadikan diagram kelas sebagai tuntunan. Dan fase terakhir yaitu fase transisi, fase yang melakukan pengujian *black box* terhadap fitur perangkat lunak. Setelah melewati pengujian, perangkat lunak yang dikembangkan telah memenuhi *requirement* sehingga dapat digunakan untuk keperluan penelitian.