



LEMBAR PENGESAHAN

**PERANGKAT LUNAK VISUALISASI PERJALANAN KERETA API
DENGAN MENGGUNAKAN PENDEKATAN SEMAPHORE,
DEADLOCK SOLUTION, DAN ALGORITMA DIJKSTRA**

Tugas Akhir

Program Studi Teknik Informatika

Fakultas Teknik

Universitas Widyatama

Oleh :

Esa Fauzi

0610P001

Telah disetujui dan disahkan di Bandung, Februari 2013

Pembimbing,

Sriyani Violina, S.T., M.T.

NID. 0401067407

Ka. Prodi Teknik Informatika,

Dekan Fakultas Teknik,

Abdullah Fajar, S.Si., M.Sc

NID. 0427117004

Setiadi Yazid, Ir., M.Sc., Ph.D.

NID. 0315085402

LEMBAR PENGESAHAN

Nama Mahasiswa : Esa Fauzi

NIM : 0610P001

Judul Skripsi : Perangkat Lunak Visualisasi Perjalanan Kereta Api Dengan Menggunakan Pendekatan *Semaphore*, *Deadlock Solution*, dan Algoritma *Dijkstra*

Yang bersangkutan dinyatakan lulus dengan nilai.....

PENILAI	TANDA TANGAN
Pembimbing : 1. Sriyani Violina, S.T., M.T.	
Penguji : 1. Dr. Savitri Galih, S.Si., M.T. 2. Sukenda, S.T., M.T.	

Mengetahui,

Bandung, Januari 2013

Ketua Jurusan Teknik Informatika,

Ketua Sidang,

SURAT PERNYATAAAN

Saya yang bertanda tangan dibawah ini :

Nama : Esa Fauzi

Tempat dan Tanggal Lahir : Bandung, 8 September 1989

Alamat Orang Tua : Jalan Yupiter III No 12 Bandung

Menyatakan bahwa skripsi ini adalah benar dan hasil karya saya sendiri. Bila terbukti tidak demikian, saya bersedia menerima segala akibatnya, termasuk pencabutan kembali gelar Sarjana Teknik yang telah saya peroleh

Bandung, Januari 2013

Esa Fauzi



ABSTRACT

The train is a land transport that has many advantages. However, in practice of the use, the train still has some problems such as delays and train wreck. Train delays can occur which one of them is caused by the deadlock. While the collision, may occur in several factors, one of them is a collision between trains. This train collision has many causes, but most of them are error caused by humans. Semaphore in OS is one method that can be applied to prevent trains collide. Semaphore can be chosen because its easy to use and simple. Meanwhile, to overcome the deadlock problem, it can be solved with the prevention, avoidance, or settlement after the deadlock occurs. Then, in addition to reducing delays, shortest path Dijkstra algorithm can be used. This algorithm has advantages over the other shortest path algorithms, that are one-to-many in finding the shortest routes so its will easy to search many destination from one location. In addition the pattern of this algorithm can make easier for the storage in database when all of the shortest path want to be stored.

The visualization software of this train travel is made by using a Semaphore approaches, Deadlock Solution, and Dijkstra Algorithms. The method that used for software development is iterative incremental method. This method is a method that started from the planning stage throught to deployment and has interaction turnover between the planning, analysis, design, implementation, and testing. This visualization software uses JavaFX as the programming language and Oracle 11G as the database.

This research result is a visualization software by using 2D object. Results of the research showed that this visualization can solving the problems of train collision and deadlocks. For the further development, this visualization is expected to use others methods such as monitor, using the parameters such as speed, acceleration, deceleration, and track condition and adding the system to create the map.

Key Words: Visualization, Train, Semaphore, Deadlocks, Dijkstra Algorithm

ABSTRAKSI

Kereta api merupakan transportasi darat yang memiliki banyak keunggulan. Namun, pada prakteknya penggunaan kereta api ini masih memiliki beberapa masalah diantaranya adalah keterlambatan dan kecelakaan. Keterlambatan kereta api bisa terjadi salah satunya disebabkan oleh *deadlock*. Kecelakaan kereta api ini terjadi diakibatkan oleh beberapa faktor diantaranya faktor manusia dan sarana dan prasarana, akan tetapi faktor penyebab yang paling tinggi adalah kesalahan yang disebabkan oleh manusia. *Semaphore* dalam OS merupakan salah satu metode yang dapat diterapkan untuk mencegah kereta api saling bertabrakan. *Semaphore* ini dapat dipilih karena penggunaannya yang mudah dan sederhana. Untuk mengatasi masalah *deadlock* dapat diselesaikan dengan melakukan pencegahan, penghindaran, atau penyelesaian setelah *deadlock* terjadi. Kemudian sebagai tambahan untuk mengurangi keterlambatan, algoritma *shortest path Dijkstra* dapat dipakai. Algoritma ini mempunyai keunggulan dibandingkan algoritma *shortest path* lainnya yaitu bersifat *one-to-many* dalam mencari rute terpendek sehingga memudahkan pencarian dari satu lokasi ke banyak lokasi lainnya. Selain itu pola algoritma ini dapat memudahkan penyimpanan ke dalam *database* apabila seluruh rute terpendek ingin disimpan ke dalam *database*.

Perangkat lunak visualisasi perjalanan kereta api ini dibuat dengan menggunakan pendekatan *Semaphore*, *Deadlock Solution*, dan Algoritma *Dijkstra*. Metode pengembangan perangkat lunak yang digunakan adalah metode *iterative incremental*. Metode ini merupakan metode yang tahapannya dimulai dari perencanaan awal (*planning*) sampai dengan *deployment* dan didalamnya terdapat perputaran interaksi antara *planning*, analisis, *design*, implementasi, serta *testing*. Sedangkan pada implementasinya, perangkat lunak visualisasi ini menggunakan JavaFX sebagai bahasa pemrogramannya dan Oracle 11G sebagai databasenya.

Penelitian ini menghasilkan perangkat lunak visualisasi dengan menggunakan obyek 2D. Hasil penelitian menunjukkan bahwa perangkat lunak ini dapat mengatasi masalah tabrakan kereta api dan *deadlock*. Untuk pengembangan selanjutnya diharapkan perangkat lunak ini dapat menggunakan metode-metode lain seperti monitor, menggunakan parameter-parameter untuk perangkat lunak seperti kecepatan, percepatan, perlambatan, dan kondisi track serta menambahkan sistem untuk membuat peta.

Kata Kunci : Visualisasi, Kereta Api, *Semaphore*, *Deadlock*, Algoritma *Dijkstra*.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas segala karunia-Nya penulis dapat menyelesaikan Tugas Akhir. Tugas akhir ini dibuat dan diajukan sebagai salah satu syarat menyelesaikan program Strata 1 (S1) di Jurusan Teknik Informatika Universitas Widyatama Bandung. Adapun judul tugas akhir ini adalah “Perangkat Lunak Visualisasi Perjalanan Kereta Api Dengan Menggunakan Pendekatan *Semaphore*, *Deadlock Solution*, dan Algoritma *Dijkstra*”.

Penyusunan Tugas Akhir ini tidak akan pernah selesai tanpa bantuan dan bimbingan dari berbagai pihak baik secara moral maupun spiritual yang tiada terkira nilainya. Oleh karena itu dengan seluruh rasa hormat dan ketulusan hati, penulis mengucapkan terimakasih kepada :

1. Ibu dan Bapak tercinta untuk doa restu, pengorbanan, cinta dan kasih sayang yang tercurah selama ini.
2. Ibu Sriyani Violina selaku dosen pembimbing tugas akhir penulis yang telah membimbing dan mengarahkan dengan ikhlas dan sabar. Semoga segala ilmu yang diberikan menjadi amal sholeh di sisi Allah SWT.
3. Seluruh Dosen Program Studi Teknik Informatika Universitas Widyatama yang telah mengajarkan ilmu-ilmu yang bermamfaat.
4. Seluruh Staff Administrasi Program Studi Teknik Informatika Universitas Widyatama yang telah memberikan kelancaran administrasi selama penulis mengerjakan Tugas Akhir ini.
5. Rekan-rekan mahasiswa Program Studi Teknik Informatika Universitas Widyatama yang telah memberikan saran dan masukan serta inspirasi yang sangat berharga bagi penulis.
6. Pihak-pihak yang tidak dapat disebutkan satu-persatu yang telah membantu baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa dalam pembuatan laporan tugas akhir ini masih jauh dari sempurna sebagai suatu karya ilmiah yang baik. Hal ini disebabkan masih rendah dan terbatasnya pengetahuan serta pengalaman yang dimiliki penulis, oleh karena itu penulis sangat mengharapkan kritik dan saran dari

pembaca untuk memperbaiki laporan ini. Akhir kata semoga laporan tugas akhir ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca umumnya.

Bandung, Januari 2013

Penulis



DAFTAR ISI

ABSTRAKSI	i
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR SIMBOL	xvi
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang Masalah	I-1
1.2 Perumusan Masalah	I-4
1.3 Tujuan	I-4
1.4 Ruang Lingkup	I-4
1.5 Model Pengembangan Perangkat Lunak	I-6
1.6 Sistematika Penulisan Laporan	I-8
BAB II LANDASAN TEORI	II-1
2.1 Konsep <i>Graph</i>	II-1
2.1.1 Definisi <i>Graph</i>	II-1
2.1.2 Jenis – Jenis <i>Graph</i>	II-1
2.1.3 Konsep dan karakteristik sistem	II-3
2.2 Algoritma Dijkstra	II-4
2.3 Geometri	II-8
2.3.1 Persamaan Parameter	II-8
2.3.2 Persamaan Baku Lingkaran	II-8
2.3.3 Rumus Titik Tengah	II-9
2.4 Deadlock	II-9
2.5 Thread	II-10
2.5.1 Definisi <i>Thread</i>	II-10
2.5.2 Keuntungan MultiThreading	II-11
2.5.3 Model Multithreading	II-12

2.6 Komunikasi Interproses	II-13
2.7 Semaphore	II-15
2.8 Komputer <i>Graphik</i> (Transformasi)	II-17
2.8.1 <i>Translation</i> (Mengeser)	II-18
2.9 <i>Tool</i> Pengembangan	II-19
2.9.1 Konsep Unified Modeling Language (UML)	II-19
2.9.1.1 <i>Use case</i> Diagram	II-19
2.9.1.2 <i>Sequence</i> Diagram	II-20
2.9.1.3 <i>Class</i> Diagram	II-21
2.9.1.4 <i>Deployment</i> Diagram	II-22
2.9.1.5 <i>Package</i> Diagram	II-23
2.9.2 Java	II-24
2.9.2.1 Java FX	II-25
2.9.2.2 Java FX 2.0	II-26
2.9.2.3 FXML	II-26
2.9.3 Oracle	II-27
2.10 Visualisasi	II-28
BAB III ANALISIS SISTEM	III-1
3.1 Analisis Perilaku Sistem	III-1
3.1.1 Proses menentukan lokasi awal keberangkatan dan tujuan dari kereta api	III-1
3.1.2 Penentuan rute kereta api	III-1
3.1.3 Proses perjalanan kereta api	III-2
3.1.4 Visualisasi perjalanan kereta	III-2
3.2 Gambaran Sistem	III-2
3.3 Analisis Perangkat Sistem	III-4
3.4 Analisis Aturan Bisnis	III-5
3.5 Asumsi	III-6
3.6 <i>Use Case</i> Model	III-6
3.6.1 <i>Use Case</i> Diagram	III-6
3.6.2 <i>Use case</i> Spesification	III-7

3.7	Lingkup Pengembangan Sistem	III-19
3.8	<i>Entry</i> Data Perjalanan Kereta	III-21
3.9	Penentuan Rute Perjalanan	III-22
3.9.1	Analisis Pembentukan Rute Perjalanan	III-22
3.10	Perjalanan Kereta	III-23
3.10.1	Analisis <i>Track</i>	III-23
3.10.2	Pemodelan <i>Track</i>	III-24
3.10.3	Analisis <i>Graph</i> Pada Peta	III-26
3.10.4	Aturan Pada Saat Perjalanan	III-28
3.11	Pengaturan Perjalanan Kereta	III-29
3.11.1	Identifikasi Masalah Perjalanan Kereta	III-29
3.11.2	Identifikasi Solusi Masalah Perjalanan Kereta	III-33
3.11.3	Identifikasi Solusi Masalah Deadlock	III-40
3.11.3.1	Monitoring	III-40
3.11.3.2	Desain <i>Track</i>	III-57
3.12	Visualisasi Data / Objek Perjalanan Kereta	III-58
3.12.1	Penggambaran Objek	III-58
3.12.2	Penskalaan	III-59
3.12.3	Pergerakan kereta	III-60
3.13	Fase Elaborasi	III-61
3.13.1	Use Case Pada Fase Elaborasi	III-61
3.13.2	Use Case Specification	III-62
3.13.3	Pendekatan per-Line	III-63
3.13.3	Domain Model	III-64
3.13	Daftar <i>Requirement</i>	III-65
 BAB IV PERANCANGAN SISTEM		 IV-1
4.1	Tujuan Perancangan	IV-1
4.2	Batasan Perancangan	IV-1
4.3	Deskripsi Perancangan	IV-2
4.3.1	Perancangan Data	IV-2
4.3.1.1	Data Transient	IV-2

4.3.1.2	Data persistent	IV-6
4.3.2	High Level Design	IV-12
4.3.2.1	Use Case Realization	IV-12
4.3.2.2	Logical View	IV-23
4.3.2.3	Rancangan User Interface	IV-31
 BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM		 V-1
5.1	<i>Prototype</i>	V-1
5.2	Struktur Menu	V-1
5.3	Deskripsi Proses	V-3
5.3.1	Form Login	V-3
5.3.2	<i>Primary</i> Menu (Menu Utama)	V-4
5.3.3	<i>Entry</i> Menu Jadwal	V-6
5.3.4	<i>View</i> Jadwal Menu	V-7
5.3.5	<i>Entry</i> Kereta Menu	V-8
5.3.6	<i>View</i> Menu Kereta	V-9
5.3.7	Menu Tentang Program	V-10
5.3.8	Pencarian Rute Terpendek	V-10
5.3.9	Perjalanan Kereta	V-12
5.4	Deployment Diagram	V-13
5.5	Strategi Pengujian	V-14
5.6.	Hasil Pengujian	V-15
 BAB VI KESIMPULAN DAN SARAN		 VI-1
6.1	Kesimpulan	VI-1
6.2	Saran	VI-2
 DAFTAR PUSTAKA		 VII-1
LAMPIRAN		

DAFTAR TABEL

Tabel 3.1 Aturan bisnis Entry Data Perjalanan Kereta	III-5
Tabel 3.2 Use case Specification Entry Data Perjalanan Kereta Api	III-7
Tabel 3.3 Use case Specification View available kereta	III-9
Tabel 3.4 Use case Specification Edit data perjalanan kereta	III-9
Tabel 3.5 Use case Specification Hapus data Perjalanan Kereta	III-10
Tabel 3.6 Use case Specification Tambah Data Perjalanan Kereta	III-11
Tabel 3.7 Use case Specification View Map	III-12
Tabel 3.8 Use case Specification View Jadwal Perjalanan	III-13
Tabel 3.9 Use case Specification Menjalankan Kereta	III-14
Tabel 3.10 Use case Specification Mengambil Rute Tercepat	III-14
Tabel 3.11 Use case Specification Memesan Vertex/Segmen	III-15
Tabel 3.12 Use case Specification Melepas Vertex/Segmen	III-15
Tabel 3.13 Use case Specification Memberhentikan Kereta	III-16
Tabel 3.14 Use case Specification Melapor ke Monitor	III-16
Tabel 3.15 Use case Specification Mencari Rute tercepat dari Setiap Vertex ke Vertex lain	III-17
Tabel 3.16 Use case Specification Menyimpan rute tercepat ke database	III-17
Tabel 3.17 Use case Specification Menerima Laporan Tidak dapat berjalannya Kereta	III-17
Tabel 3.18 Use case Specification Mendeteksi Deadlock	III-18
Tabel 3.19 Use case Specification Pasang Sistem	III-18
Tabel 3.20 Use case Specification membuat Resolusi Deadlock	III-19
Tabel 3.21 Informasi kereta ketergantungan	III-41
Tabel 3.22 Data laporan	III-45
Tabel 3.23 Identifikasi Kereta Bergantung	III-45
Tabel 3.24 Daftar lapor kereta	III-45
Tabel 3.25 Keterangan simbol peta	III-58
Tabel 3.26 Memesan Line	III-62
Tabel 3.27 Melepas Line	III-62
Tabel 3.28 Daftar Requirement	III-65
Tabel 4.1 Data Persistent Entity Jenis Kereta	IV-6

Tabel 4.2 Data Persistent Entity Kelas Kereta	IV-6
Tabel 4.3 Data Persistent Entity lokomotif Kereta	IV-6
Tabel 4.4 Data Persistent Entity Nama Kereta	IV-6
Tabel 4.5 Data Persistent Entity Kereta	IV-7
Tabel 4.6 Data Persistent Entity Vertex	IV-7
Tabel 4.7 Data Persistent Entity Stasiun	IV-7
Tabel 4.8 Data Persistent Entity Hasil Dijkstra	IV-7
Tabel 4.9 Data Persistent Entity Jadwal	IV-8
Tabel 4.10 Data Persistent Entity Penghubung	IV-8
Tabel 4.11 Data Persistent Entity Segmen	IV-8
Tabel 4.12 Data Persistent Entity Jenis Segmen belok	IV-9
Tabel 5.1 Pengujian Proses Perjalanan Kereta	V-15
Tabel 5.2 Pengujian Entry dan View Kereta	V-17
Tabel 5.3 Pengujian Entry Jadwal, View Jadwal, dan View stasiun	V-18
Tabel 5.4 Pengujian Pembentukan Rute Tercepat	V-19

DAFTAR GAMBAR

Gambar 1.1 Data Kecelakaan Kereta Api	I-2
Gambar 1.2 Penyebab Kecelakaan Kereta Api	I-2
Gambar 1.3 Fase dan <i>Discipline</i> pada <i>iterative development</i>	I-6
Gambar 2.1 <i>Graph</i> Berarah dan Berbobot	II-2
Gambar 2.2 <i>Graph</i> Tidak Berarah dan Berbobot	II-2
Gambar 2.3 <i>Graph</i> Berarah dan Tidak Berbobot	II-2
Gambar 2.4 <i>Graph</i> tidak Berarah dan Tidak Berbobot	II-3
Gambar 2.5 Contoh Adjacency Matrix	II-3
Gambar 2.6 Jarak antara Vertex u dan V selalu positif	II-4
Gambar 2.7 Dijkstra dengan simpul yang berdekatan	II-6
Gambar 2.8 Dijkstra dengan relaks Tetangga C	II-6
Gambar 2.9 Dijkstra yang Berdekatan dengan Vertex himpunan S	II-7
Gambar 2.10 Dijkstra Jalan dan Jalan Terpendek	II-8
Gambar 2.11 Deadlock	II-10
Gambar 2.12 Model Multithreading	II-12
Gambar 2.13 Model komunikasi (A) message passing (B) shared memory	II-14
Gambar 2.14 Ilustrasi Translasi	II-18
Gambar 2.15 Contoh Use case diagram	II-20
Gambar 2.16 Contoh Sequence Diagram	II-21
Gambar 2.17 Contoh Class Diagram	II-22
Gambar 2.18 Contoh Deployment Diagram	II-23
Gambar 3.1 Gambaran Sub Sistem	III-3
Gambar 3.2 Use Case Diagram Sistem Fase Insepsi	III-7
Gambar 3.3 Contoh Track Belok	III-24
Gambar 3.4 Contoh <i>Graph</i> Track Belok	III-24
Gambar 3.5 Contoh track Lurus	III-25
Gambar 3.6 Contoh <i>Graph</i> Track Lurus	III-25
Gambar 3.7 Contoh Track simpangan perempatan	III-25
Gambar 3.8 Contoh track Persimpangan perempatan-2	III-26
Gambar 3.9 Contoh <i>graph</i> persimpangan perempatan	III-26
Gambar 3.10 Contoh keterkaitan objek dengan vertex	III-27




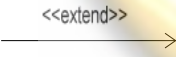
Gambar 3.11 Contoh kereta Tabrakan	III-30
Gambar 3.12 Contoh Antrian kereta	III-30
Gambar 3.13 Contoh kasus 2 kereta pada jalan lurus	III-31
Gambar 3.14 Contoh kasus 2 kereta pada persimpangan	III-31
Gambar 3.15 Contoh kasus 3 kereta pada persimpangan	III-32
Gambar 3.16 Contoh kasus 4 kereta pada persimpangan	III-32
Gambar 3.17 Peta kondisi	III-41
Gambar 3.18 Contoh kasus DFS	III-42
Gambar 3.19 Contoh Back edge	III-43
Gambar 3.20 Contoh forward	III-44
Gambar 3.21 <i>Graph</i> kasus deadlock	III-47
Gambar 3.22 Kasus pada pendekatan 1	III-49
Gambar 3.23 kasus pada pendekatan 2	III-49
Gambar 3.24 Kasus pada pendekatan 3	III-51
Gambar 3.25 Kasus pada pendekatan 4-1	III-53
Gambar 3.26 Kasus pada pendekatan 4-2	III-54
Gambar 3.27 Contoh desain 2 track untuk menangani deadlock	III-57
Gambar 3.28 Contoh desain track lonceng untuk mengani deadlock	III-57
Gambar 3.29 Contoh titik Bezier	III-59
Gambar 3.30 Contoh pembuatan Garis lurus	III-59
Gambar 3.31 Contoh track berbentuk garis lurus	III-60
Gambar 3.32 Use Case Diagam Fase Elaborasi	III-61
Gambar 3.33 Contoh Penerapan Pendekatan Line	III-64
Gambar 3.34 Domain Model	III-65
Gambar 4.1 <i>Conceptual Data Model</i>	IV-10
Gambar 4.2 <i>Physical Data Model</i>	IV-11
Gambar 4.3 Sequence Diagram Pencarian Rute terpendek	IV-12
Gambar 4.4 <i>Sequence Diagram Entry Jadwal</i>	IV-13
Gambar 4.5 Sequence Diagram Entry Kategori Kereta	IV-14
Gambar 4.6 <i>Sequence Diagram Entry Kereta</i>	IV-15
Gambar 4.7 <i>Sequence Diagram Entry Stasiun</i>	IV-16
Gambar 4.8 <i>Sequence Diagram Gerakan Kereta</i>	IV-18

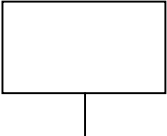
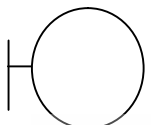
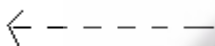


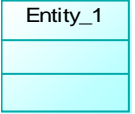


Gambar 4.9 <i>Sequence Diagram View Jadwal</i>	IV-19
Gambar 4.10 <i>Sequence Diagram View Kereta</i>	IV-20
Gambar 4.11 <i>Sequence Diagram View Stasiun</i>	IV-21
Gambar 4.12 <i>Sequence Diagram Menu Utama</i>	IV-22
Gambar 4.13 <i>Package Diagram</i>	IV-23
Gambar 4.14 <i>Diagram Kelas Menu Utama dan Login</i>	IV-25
Gambar 4.15 <i>Kelas Diagram Entry Jadwal dan View Jadwal</i>	IV-26
Gambar 4.16 <i>Kelas Diagram Entry Menu Kereta dan View Kereta</i>	IV-27
Gambar 4.17 <i>Kelas Diagram Entry Stasiun dan View Stasiun</i>	IV-28
Gambar 4.18 <i>Kelas Model Relasi Dijkstra</i>	IV-29
Gambar 4.19 <i>Kelas Diagram Perjalanan Kereta</i>	IV-30
Gambar 4.20 <i>Rancangan User Interface Entry Jadwal Perjalanan Kereta</i>	IV-31
Gambar 4.21 <i>Rancangan User Interface Entry Kereta</i>	IV-32
Gambar 4.22 <i>Rancangan User Interface Entry Stasiun</i>	IV-32
Gambar 4.23 <i>Rancangan User Interface View Jadwal</i>	IV-33
Gambar 4.24 <i>Rancangan User Interface View Kereta</i>	IV-34
Gambar 4.25 <i>Rancangan User Interface View Stasiun</i>	IV-35
Gambar 4.26 <i>Rancangan User Interface Menu utama</i>	IV-36
Gambar 5.1 <i>Struktur Menu</i>	V-2
Gambar 5.2 <i>Form Login</i>	V-3
Gambar 5.3 <i>Primary Menu</i>	V-5
Gambar 5.4 <i>Tampilan Entry Jadwal Kereta</i>	V-6
Gambar 5.5 <i>Tampilan View Jadwal Menu</i>	V-7
Gambar 5.6 <i>Tampilan Entry Kereta Menu</i>	V-8
Gambar 5.7 <i>Tampilan View Kereta Menu</i>	V-9
Gambar 5.8 <i>Tampilan Menu Tentang Program</i>	V-10
Gambar 5.9 <i>Tampilan SQL command line ketika memanggil fungsi count (jumlah data) dari tabel hasil_dijkstra-1</i>	V-10
Gambar 5.10 <i>Tampilan SQL command line ketika memanggil fungsi count (jumlah data) dari tabel hasil_dijkstra-2</i>	V-11
Gambar 5.11 <i>Data yang terisikan ke dalam tabel hasil dijkstra dilihat dari Oracle SQL Developer</i>	V-11



Gambar 5.12 Proses Perjalanan Kereta-1	V-12
Gambar 5.13 Proses Perjalanan Kereta-2	V-13
Gambar 5.14 Deployment Diagram	V-14



DAFTAR SIMBOL

SIMBOL	NAMA	PENGGUNAAN	KETERANGAN
	Actor	<i>Use case Diagram</i>	<i>User</i> dari sistem atau aplikasi. <i>User</i> bisa berupa manusia, mesin, sistem lain atau subsistem lain.
	<i>Use case</i>	<i>Use case Diagram</i>	Sebuah elemen pemodelan pada UML yang menggambarkan bagaimana <i>user</i> dari sebuah sistem berinteraksi dengan sistem tersebut untuk melakukan suatu pekerjaan yang berbeda.
	<i>Colaboration</i> <i>n</i>	<i>Use case Diagram</i>	Digunakan sebagai use case control atau sistem yang juga memiliki peran seperti aktor
	<i>Use (Relasi)</i>	<i>Use case Diagram</i>	Mengindikasikan bahwa suatu elemen memerlukan elemen yang lain untuk melakukan interaksi.
	<i>Extend</i> <i>Dependency</i>	<i>Use case Diagram</i>	Mengindikasikan suatu elemen merupakan perluasan perilaku dari elemen lain. Relasi <i>extend</i> digunakan di pemodelan <i>use case</i> untuk mengindikasikan suatu <i>use case</i> merupakan perluasan (pilihan) perilaku dari <i>use case</i> lain.
	<i>Include</i> <i>Dependency</i>	<i>Use case Diagram</i>	Mengindikasikan bahwa suatu elemen termasuk fungsi dari elemen target. Relasi <i>include</i> digunakan di pemodelan <i>use case</i> untuk menggambarkan bahwa suatu <i>use case</i> termasuk perilaku dari <i>use case</i> yang lain.

SIMBOL	NAMA	PENGGUNAAN	KETERANGAN
	<i>Lifeline</i>	<i>Sequence Diagram</i>	Merepresentasikan <i>class</i> /objek yang berinteraksi
	<i>Boundary</i>	<i>Sequence Diagram</i>	Objek <i>stereotype</i> yang memodelkan beberapa sistem boundary, seperti user interface
	<i>Message (return)</i>	<i>Sequence Diagram</i>	Aliran informasi diantara elemen yang berupa nilai kembalian
	<i>Message</i>	<i>Sequence Diagram</i>	Aliran informasi diantara elemen
	<i>Class</i>	<i>Class Diagram</i>	Sebuah <i>class</i> merupakan representasi dari objek, yang menggambarkan struktur dan perilaku dalam sistem. <i>Class</i> memiliki <i>attribute</i> (data) atau <i>method</i> (operasi). <i>Attribute</i> ditandai melalui penulisan dengan warna merah dan operasi ditandai melalui penulisan dengan warna hijau.
	<i>Entity</i>	<i>Conceptual Data Model</i>	Merupakan mekanisme penyimpanan atau <i>persistence</i> yang menangkap informasi atau ilmu pengetahuan dari sebuah sistem
	<i>One to Many (mandatory - dependency)</i>	<i>Conceptual Data Model</i>	Hubungan satu ke banyak antar dua table
	<i>One to One (mandatory)</i>	<i>Conceptual Data Model</i>	Hubungan satu ke satu antar dua tabel

SIMBOL	NAMA	PENGGUNAAN	KETERANGAN
 Relationship_1	<i>Many to Many</i>	<i>Conceptual Data Model</i>	Hubungan banyak ke banyak antar dua tabel
 Relationship_1	<i>Many to One</i>	<i>Conceptual Data Model</i>	Hubungan banyak ke satu antar table



DAFTAR LAMPIRAN

Lampiran A

Riwayat Hidup

Lampiran B

Kartu Bimbingan

Lampiran C

Formulir Revisi

