

**IMPLEMENTASI METODE *COMPREHENSIVE HYPERMEDIA
DESIGN* (Com HDM) PADA RANCANG BANGUN SISTEM
MARKETPLACE PEMESANAN TRAVEL BERBASIS WEB**

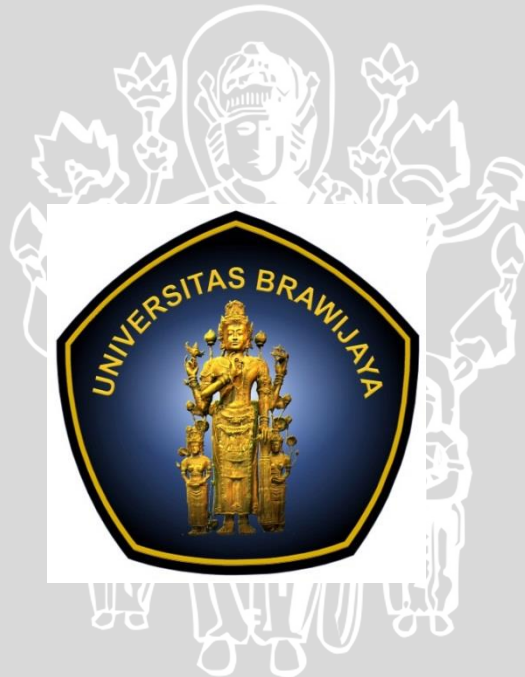
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Wildan Ryan Raditya

NIM: 0910683096



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI METODE *COMPREHENSIVE HYPERMEDIA DESIGN* (Com HDM)
PADA RANCANG BANGUN SISTEM *MARKETPLACE* PEMESANAN TRAVEL BERBASIS
WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :

Wildan Ryan Raditya

NIM: 0910683096

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Agustus 2016

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Denny Sagita R., S.Kom., M.Kom.

NIP: 198511242015041001

Issa Arwani, S.Kom., M.Sc.

NIP: 198309222012121003

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, Ph.D

NIP: 19670801 199203 1001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Agustus 2016

Wildan Ryan Raditya



KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Metode *Comprehensive Hypermedia Design* (Com HDM) Pada Rancang Bangun Sistem *Marketplace* Pemesanan *Travel* Berbasis *Web*”. Shalawat serta salam tak lupa penulis sampaikan kepada junjungan besar kita Nabi Muhammad SAW beserta keluarga dan para sahabat-Nya. Skripsi ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer di Jurusan Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya Malang.

Dalam skripsi ini masih banyak sekali kekurangan dan masih jauh dari kesempurnaan, hal ini dikarenakan keterbatasan kemampuan yang penulis miliki. Terdapat beberapa hambatan dan kesulitan dalam penulisan skripsi ini, tetapi Alhamdulillah dapat diselesaikan dengan baik. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada pihak yang telah memberikan bantuan baik secara langsung maupun tidak langsung, diantaranya:

1. Kedua orang tua saya Herijanto dan Suwarni yang telah melahirkan, membesarkan serta memberi dukungan setiap waktu.
2. Bapak Denny Sagita R., S.Kom., M.Kom. selaku Dosen Pembimbing I dan bapak Issa Arwani, S.Kom., M.Sc. selaku Dosen Pembimbing II yang telah memberikan ilmu dan saran selama penyusunan skripsi ini.
3. Bapak Agus Wahyu Widodo, S.T, M.Sc. selaku Ketua Program Studi Informatika yang telah memberikan kemudahan birokrasi.
4. Segenap dosen dan staff pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan.
5. Luthfi Aziz atas waktu dan tambahan ilmu yang di berikan untuk menyelesaikan penelitian ini.
6. Brilian A.V atas bantuan dalam menyelesaikan penelitian ini.
7. Yuri Pratama yang telah menemani setiap hari dan bantuannya.
8. Mas Toni atas ketersediaan tempat yang diberikan untuk mengerjakan penelitian ini.
9. Kakak Bonie yang bersedia menemani hingga terselesaikan penelitian ini.
10. Seluruh teman-teman TIF 2009 atas kebersamaannya dari awal perkuliahan hingga akhir.
11. Kakak-kakak tingkat dan adik-adik tingkat yang memberi dukungan selama perkuliahan.
12. Keluarga besar parkiran yang memberikan hiburan di waktu senggang.
13. Serta semua pihak yang telah membantu dan memberikan pengalaman berharga selama perkuliahan.

14. Alief Afini Swaranthy yang telah memberi semangat untuk segera menyelesaikan penelitian ini.
15. Kepada semua pihak yang tidak dapat saya sebutkan satu persatu atas segala kontribusi yang diberikan.

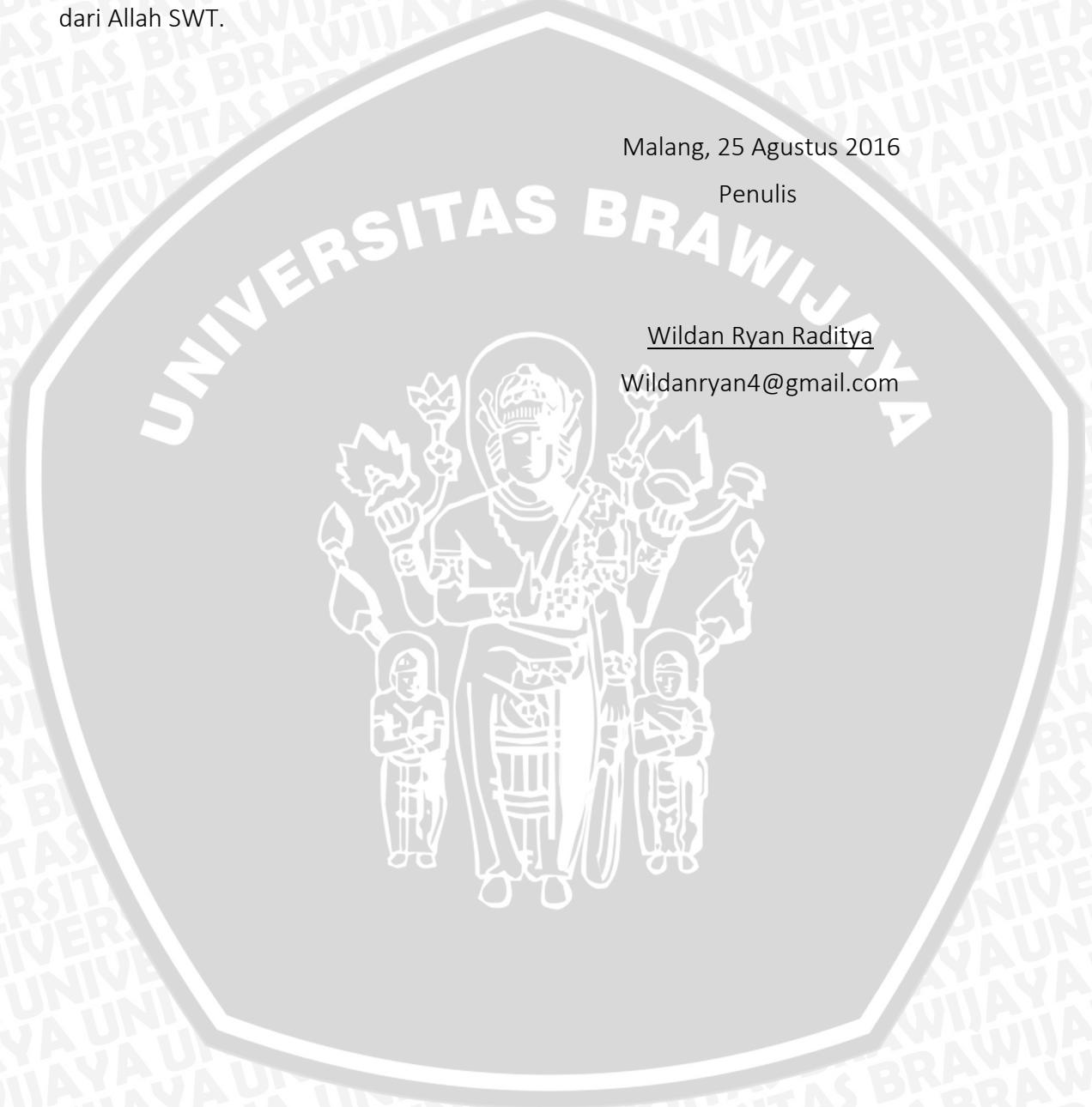
Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak dan semoga amal baik yang telah diberikan kepada penulis mendapat balasan dari Allah SWT.

Malang, 25 Agustus 2016

Penulis

Wildan Ryan Raditya

Wildanryan4@gmail.com



ABSTRAK

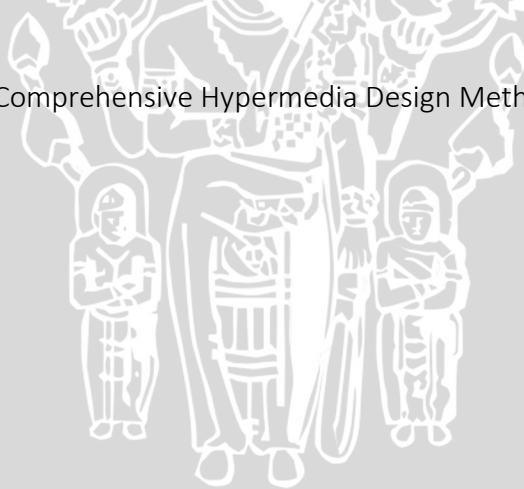
Di zaman modern, teknologi informasi disertai internet sangat berkembang. Keduanya saling berkaitan dan berpengaruh yang akhirnya terbentuk konsep teknologi informasi berbasis internet yang semakin banyak diminati dan digunakan dalam bisnis perusahaan di banyak aspek. Bisnis transportasi darat khususnya pada bidang pelayanan jasa dengan menggunakan mobil *travel* atau *shuttle* pada saat ini mulai banyak diminati. Hambatan dari bisnis *travel* atau *shuttle* adalah sulitnya pelanggan dalam mengetahui informasi letak alamat, ketersediaan tempat duduk, waktu keberangkatan, harga serta pemesanan masih dilakukan secara konvensional dengan datang ke tempat *travel*. Sebuah perangkat lunak "Sistem Marketplace Pemesanan Travel Berbasis Web" di buat untuk memudahkan pelanggan dalam melakukan pemesanan. Studi literatur, analisis kebutuhan sistem, perancangan data, perancangan perangkat lunak, implementasi, pengujian, pengambilan kesimpulan dan saran menjadi metode pembuatan aplikasi ini. Pada fase desain dalam perancangan akan di bantu dengan metode lain untuk membuat gambaran sebuah sistem aplikasi *web* yang memiliki desain alur informasi serta data yang jelas mudah di pahami. Com HDM (*Comprehensive Hypermedia Design Method*), sebuah metode desain *web* yang komprehensif dan sistematis untuk mendukung semua karakteristik dalam konteks desain aplikasi *web* ini. Pada fase pengujian perangkat lunak untuk fungsionalitas telah berfungsi seperti yang di harapkan. Untuk memenuhi aspek efisien atau memudahkan pengujian *usability* dilakukan dan memenuhi aspek dengan prosentase 88,2%.

Kata kunci: Com HDM (*Comprehensive Hypermedia Design Method*), mobil *travel*

ABSTRACT

In modern times, the information technology is highly developed with the Internet. Both are interrelated and influence that eventually formed the concept of Internet-based information technology is more and more popular and used in a business enterprise in many aspects. Land transportation business, especially in the field of services by car or shuttle travel at this time began in great demand. Barriers of business travel or customer shuttle is the difficulty in knowing the location of the address information, seat availability, time of departure, prices and bookings are still done conventionally by coming to the place of travel. A software "Travel Booking System Web-based Marketplace" created to enable customers to make a reservation. The study of literature, analysis of system requirements, design of data, software design, implementation, testing, pangambilan conclusions and suggestions become the method of making this application. In the design phase in the design will be assisted by another method to create a picture of a web application system which has a design flow of information and data that is obviously easier to understand. Com HDM (Hypermedia Design Comprehensive Method), a method of web design comprehensive and systematic to support all the characteristics in the context of this web application design. In the testing phase peningkatan functionality has software to function as expected. To meet or facilitate efficient aspects of usability testing is done and meets aspects with a percentage of 88.2%.

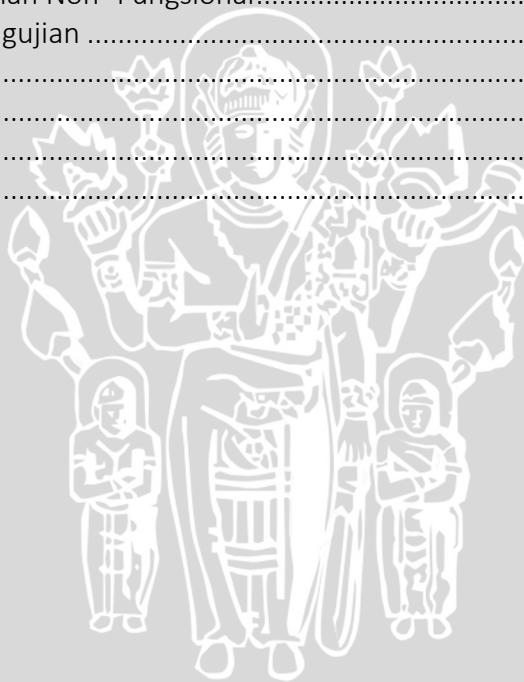
Keywords: Com HDM (Comprehensive Hypermedia Design Method), travel car



DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika pembahasan	3
BAB 2 DASAR TEORI	5
2.1 Kajian Pustaka	5
2.2 Rekayasa Perangkat Lunak (RPL)	5
2.3 <i>Software Development Life Cycle (SDLC) Waterfall</i>	6
2.4 <i>Unified Modelling Language</i>	6
2.4.1 <i>Use Case Diagram</i>	7
2.4.2 <i>Class Diagram</i>	8
2.4.3 <i>Activity Diagram</i>	10
2.4.4 <i>Sequence Diagram</i>	11
2.5 <i>MySQL Database Management System</i>	12
2.6 <i>Hyper Text Markup Language (HTML)</i>	13
2.7 <i>Cascading Style Sheet (CSS)</i>	13
2.8 <i>Javascript</i>	14
2.9 <i>Com HDM (Comprehensive Hypermedia Design Method)</i>	15
2.10 Pengujian Perangkat Lunak	20
2.10.1 <i>Black Box Testing</i>	20
2.10.2 <i>Usability Testing</i>	21
2.11 Skala Likert	21
BAB 3 METODOLOGI PENELITIAN	22
3.1 Analisis Kebutuhan Sistem	22
3.2 Perancangan Perangkat Lunak	23
3.3 Implementasi	25
3.4 Pengujian	25
BAB 4 ANALISIS DAN PERANCANGAN	26
4.1 Analisis Kebutuhan	27
4.1.1 Gambaran Umum Perangkat Lunak	27
4.1.2 Identifikasi Aktor	29
4.1.3 Kebutuhan Fungsional	30
4.1.4 Kebutuhan Non Fungsional	32

4.1.5 Use Case Diagram	32
4.1.6 Activity Diagram	41
4.2 Perancangan Perangkat Lunak	48
4.2.1 Squence Diagram	48
4.2.2 Conceptual Models	51
4.2.3 Navigation Models	54
4.2.4 User Interface Model	55
4.2.5 Perancangan penyimpanan data	62
BAB 5 IMPLEMENTASI	64
4.1 Implementasi	64
4.1.1 Implementasi Basis Data	64
4.1.2 Implementasi User Interface	64
4.1.3 Implementasi Program	70
BAB 6 PENGUJIAN	75
5.1 Pengujian	75
5.1.1 Pengujian Fungsional	75
5.1.2 Pengujian Non- Fungsional	88
5.2 Analisis Pengujian	90
BAB 7 Penutup	92
6.1 Kesimpulan	92
6.2 Saran	92
DAFTAR PUSTAKA	93



DAFTAR TABEL

Tabel 2.1 Keterangan simbol pada <i>use case diagram</i>	8
Tabel 2.2 <i>Multiplicity</i> dan artinya	9
Tabel 2.3 Keterangan simbol dalam <i>conceptual class model</i>	17
Tabel 2.4 Keterangan simbol dalam <i>navigation class model</i>	17
Tabel 2.5 Keterangan simbol dalam <i>navigation access and interaction model</i>	18
Tabel 2.6 Keterangan simbol transformasi	19
Tabel 2.7 Keterangan simbol dalam <i>user interface</i>	19
 Tabel 3.1 Daftar fitur implementasi	 23
 Tabel 4.1 Identifikasi aktor	 29
Tabel 4.2 Kebutuhan Fungsional	30
Tabel 4.3 Kebutuhan Non-Fungsional	32
Tabel 4.4 Spesifikasi Use Case Melakukan Register	33
Tabel 4.5 Use Case Melakukan Login	34
Tabel 4.6 Use Case Melakukan Pemesanan <i>Travel</i>	35
Tabel 4.7 Use Case Melakukan Konfirmasi Pemesanan <i>Travel</i>	36
Tabel 4.8 Use Case Lihat Order	36
Tabel 4.9 Use Case Lihat <i>Travel Publisher</i>	37
Tabel 4.10 Use Case Lihat List Vendor <i>Travel</i>	38
Tabel 4.11 Use Case Melakukan Edit Profil	38
Tabel 4.12 Use Case Mengelola Akun	39
Tabel 4.13 Use Case Mengatur Jadwal	39
Tabel 4.14 Use Case Melakukan Logout	40
 Tabel 6.1 Pengujian Fungsional Admin	 75
Tabel 6.2 <i>Functional Testing</i> Fungsional Login Admin	76
Tabel 6.3 <i>Functional Testing</i> Fungsional <i>Travel Publisher</i>	76
Tabel 6.4 <i>Functional Testing</i> Fungsional List <i>Travel</i>	77
Tabel 6.5 <i>Functional Testing</i> Mengelola Akun	77
Tabel 6.6 <i>Functional Testing</i> Logout	78
Tabel 6.7 Pengujian Fungsional Vendor <i>Travel</i>	78
Tabel 6.8 <i>Functional Testing</i> Register Vendor <i>Travel</i>	79
Tabel 6.9 <i>Functional Testing</i> Login Vendor <i>Travel</i>	79
Tabel 6.10 <i>Functional Testing</i> Mengatur Jadwal Vendor <i>Travel</i>	80
Tabel 6.11 <i>Functional Testing</i> Fungsional <i>Travel Publisher</i>	81
Tabel 6.12 <i>Functional Testing</i> Fungsional List <i>Travel</i>	81
Tabel 6.13 <i>Functional Testing</i> Lihat Order Vendor <i>Travel</i>	82
Tabel 6.14 <i>Functional Testing</i> Logout	82
Tabel 6.15 Pengujian Fungsional Customer	83
Tabel 6.16 <i>Functional Testing</i> Register Customer	83
Tabel 6.17 <i>Functional Testing</i> Login Customer	84
Tabel 6.18 <i>Functional Testing</i> Edit Profil Customer	85
Tabel 6.19 <i>Functional Testing</i> Pemesanan <i>Travel</i>	85

Tabel 6.20 <i>Functional Testing</i> Konfirmasi Pemesanan	86
Tabel 6.21 <i>Functional Testing</i> Lihat Order	86
Tabel 6.22 <i>Functional Testing</i> List Travel	87
Tabel 6.23 <i>Functional Testing</i> Travel Publisher	87
Tabel 6.24 <i>Functional Testing</i> Logout	88
Tabel 6.25 Kuesioner Pengujian <i>Usability</i>	89
Tabel 6.26 Hasil Data <i>Usability Testing</i>	89
Tabel 6.27 Hasil Pengujian Non-Fungsional	90



DAFTAR GAMBAR

Gambar 2.1 Model SDLC <i>Waterfall</i>	6
Gambar 2.2 Contoh <i>use case diagram</i>	7
Gambar 2.3 Contoh Class Diagram	10
Gambar 2.4 Contoh <i>Activity Diagram</i>	11
Gambar 2.5 <i>Sequence Diagram</i>	12
Gambar 2.6 Alur metode perancangan (Com HDM)	16
Gambar 2.7 Contoh alur perancangan biasa	16
 Gambar 3.1 <i>Flowchart</i> metode penelitian	22
Gambar 3.2 Gambaran umum perancangan data	24
Gambar 3.3 Gambaran umum implementasi aplikasi sistem	25
 Gambar 4.1 Diagram perancangan.....	26
Gambar 4.2 Diagram blok sistem <i>marketplace</i> pemesanan	27
Gambar 4.3 Diagram blok pendaftaran <i>vendor</i> pada sistem <i>marketplace</i>	28
Gambar 4.4 Skenario penyebaran informasi sistem <i>marketplace</i>	28
Gambar 4.5 <i>Use Case diagram</i>	33
Gambar 4. 6 Proses Melakukan <i>Register</i>	41
Gambar 4.7 Proses Melakukan <i>Login</i>	42
Gambar 4.8 Proses Melakukan Pemesanan <i>Travel</i>	43
Gambar 4.9 Proses Melakukan Konfirmasi Pemesanan <i>Travel</i>	43
Gambar 4. 10 Proses Lihat <i>Travel Publisher</i>	44
Gambar 4.11 Proses Lihat <i>List Vendor Travel</i>	44
Gambar 4.12 Proses <i>Customer</i> Melakukan Edit Profil.....	45
Gambar 4.13 Proses <i>Customer</i> Lihat <i>Order</i>	46
Gambar 4.14 Proses <i>Travel</i> Mengatur Jadwal	46
Gambar 4.15 Proses Admin Mengelola Akun	47
Gambar 4.16 Proses Melakukan <i>Logout</i>	48
Gambar 4.17 <i>Squence Login User</i>	49
Gambar 4.18 <i>Squence Pemesanan Travel</i>	50
Gambar 4.19 <i>Squence List Travel</i>	51
Gambar 4.20 <i>Conceptual Class Model</i>	52
Gambar 4.21 <i>Complex Processes Flow</i>	52
Gambar 4.22 <i>Conceptual Process Model</i>	53
Gambar 4.23 <i>Navigation Class Model</i>	54
Gambar 4.24 <i>Navigation Access Interaction Model</i>	55
Gambar 4.25 <i>User Interface</i> Halaman <i>Home</i>	55
Gambar 4.26 <i>User Interface</i> Halaman <i>Register</i>	56
Gambar 4.27 <i>User Interface</i> Halaman <i>Register Vendor Travel</i>	56
Gambar 4.28 <i>User Interface</i> Halaman <i>Register Customer</i>	57
Gambar 4.29 <i>User Interface</i> Halaman <i>Login</i>	57
Gambar 4.30 <i>User Interface</i> Halaman <i>Home Vendor Travel</i>	58
Gambar 4.31 <i>User Interface</i> Halaman <i>Add Schedule Travel</i>	58
Gambar 4.32 <i>User Interface</i> Halaman <i>List Travel</i>	59

Gambar 4.33 <i>User Interface</i> Halaman <i>Home Customer</i>	59
Gambar 4.34 <i>User Interface</i> Halaman <i>Edit Profil Customer</i>	60
Gambar 4.35 <i>User Interface</i> Halaman <i>Pemesanan</i>	60
Gambar 4.36 <i>User Interface</i> Halaman <i>Lihat Order</i>	61
Gambar 4.37 <i>User Interface</i> Halaman <i>Publisher</i>	61
Gambar 4.38 <i>User Interface</i> Halaman <i>Admin</i>	62
Gambar 4.39 <i>Gambar Entity Relationship Diagram</i>	63
Gambar 4.40 <i>Perancangan Penyimpanan Data</i>	63
Gambar 5.1 <i>Physical Diagram</i>	64
Gambar 5.2 Halaman <i>Login User</i>	65
Gambar 5.3 Halaman <i>Admin</i>	65
Gambar 5.4 Halaman <i>Register Vendor Travel</i>	66
Gambar 5.5 Halaman <i>Home Vendor Travel</i>	66
Gambar 5.6 Halaman <i>Add Schedule Vendor Travel</i>	67
Gambar 5.7 Halaman <i>Register Customer</i>	67
Gambar 5.8 Halaman <i>Home Customer</i>	68
Gambar 5.9 Halaman <i>Pemesanan</i>	68
Gambar 5.10 Halaman <i>Konfirmasi Pemesanan</i>	69
Gambar 5.11 Halaman <i>Edit Profil Customer</i>	69
Gambar 5.12 Halaman <i>Lihat Order</i>	70
Gambar 5.13 <i>Cuplikan Kode Register Vendor Travel</i>	71
Gambar 5.14 <i>Cuplikan Kode Register Customer</i>	72
Gambar 5.15 <i>Cuplikan Kode Tambah Jadwal</i>	73
Gambar 5.16 <i>Cuplikan Kode Pemesanan</i>	74
Gambar 5.17 <i>Cuplikan Kode Konfirmasi Pemesanan</i>	74

BAB 1 PENDAHULUAN

1.1 Latar belakang

Di zaman modern, teknologi informasi disertai internet sangat berkembang. Keduanya saling berkaitan dan berpengaruh yang akhirnya terbentuk konsep teknologi informasi berbasis internet yang semakin banyak diminati dan digunakan dalam bisnis perusahaan di banyak aspek. Beberapa perusahaan diantaranya menggunakan aplikasi web sebagai media penyampaian informasi penjualan (Siregar, 2008).

Dalam bidang penjualan, aplikasi *web* dijadikan sarana untuk melakukan transaksi jual beli baik barang atau jasa maupun produk yang lain karena akan membuat transaksi penjualan dapat dilakukan dengan mudah dan mengurangi waktu walau jarak berjauhan (Putra, 2013). Selain sarana untuk mempersingkat transaksi penjualan, aplikasi web juga dapat digunakan untuk menampung segala informasi yang dibutuhkan untuk menjual barang atau jasa, penerapan pada bisnis transportasi darat khususnya pada bidang pelayanan jasa dengan menggunakan mobil *travel* atau *shuttle* pada saat ini mulai banyak diminati (Sumaryadi, 2012). Tidak sedikit *travel* atau *shuttle* yang sudah memanfaatkan aplikasi web sebagai media penyampaian informasi.

Ada beberapa hambatan dalam penyampaian informasi yang berkaitan dengan *travel* baik melalui *website* atau *blog*. Hambatan dari bisnis *travel* atau *shuttle* adalah masih sedikit sistem untuk mewadahi vendor *travel* berakibat sulitnya pelanggan dalam mengetahui informasi letak alamat, ketersediaan tempat duduk, waktu keberangkatan, harga yang ditawarkan serta fasilitas yang didapatkan satu per satu. Dan untuk pemesanan masih dilakukan secara konvensional dengan menghubungi *customer service* atau datang ke tempat *travel* (Binus, 2012). Dalam bisnis ini kemudahan dalam mengakses informasi dan pemesanan berperan penting karena akan berpengaruh pada banyaknya pelanggan yang menggunakan layanan jasa *travel* dan berakhir dengan kepuasan pelanggan (Naftha, 2013). Oleh karena itu, dengan kita merancang dan membuat sebuah sistem aplikasi web yang memiliki desain alur informasi serta data yang jelas akan menjawab permasalahan diatas.

Untuk menuju solusi diatas agar pelanggan mudah melakukan aktifitas di dalam sistem untuk menemukan informasi yang mereka butuhkan dengan cepat dan tepat, maka hal yang perlu dilakukan adalah mencari dan menemukan metode perancangan desain yang tepat sehingga pada saat melakukan tahapan analisa sistem, metode ini akan dipakai sehingga diharapkan akan membantu dalam proses perancangan pada desain pembangunan aplikasi *web* yang sangat kompleks. Karena dibandingkan dengan desain *website* informasi dasar yang hanya memberikan informasi statis dengan sedikit interaksi, ada kebutuhan dalam struktur desain *web* yang berisi informasi kompleks dan spesifik. *Web* memiliki komposisi informasi yang signifikan seperti proses aliran datanya, interaksi *user*, struktur konten, *link* navigasi dinamis, dan presentasi *user interface* (Jantan, 2013).

Ada beberapa metode yang dapat digunakan salah satunya Com HDM (*Comprehensive Hypermedia Design Method*).

Com HDM (*Comprehensive Hypermedia Design Method*), sebuah metode desain *web* yang komprehensif dan sistematis untuk mendukung semua karakteristik dalam konteks desain aplikasi *web*. Com HDM telah terbukti sangat berguna dalam aliran pemodelan proses kompleks yang melibatkan interaksi kompleks antara *user* dan aplikasi *web*. Elemen pemodelan khusus disediakan untuk memfasilitasi analisis dalam mengorganisir dan mengintegrasikan informasi spesifik untuk menghasilkan model desain yang ekspresif pada strukturnya (Jantan, 2013).

Berangkat dari beberapa masalah tersebut, maka diperlukan sebuah perangkat lunak yang dapat membantu mengurangi masalah yang muncul. Salah satu cara yang digunakan dalam memecahkan permasalahan diatas adalah mengembangkan *software* (perangkat lunak) aplikasi *web* yang dapat membantu kinerja dari sistem *travel* dan mempermudah pelanggan dalam melakukan pemesanan *travel* dan melihat info dari vendor *travel* yang terdaftar.

Penulis juga akan menjelaskan secara rinci tentang bagaimana Com HDM dapat digunakan untuk merancang sistem pemesanan *travel* berbasis *web*, seluruh proses persyaratan dan model desain, untuk menyoroti kekuatan desain internal dari pendekatan ini.

1.2 Rumusan masalah

Berdasarkan permasalahan yang diangkat pada bagian latar belakang, maka rumusan masalah dikhususkan pada :

1. Bagaimana proses perancangan perangkat lunak sistem *marketplace* pemesanan *travel* berbasis *web* dengan menggunakan metode Com HDM ?
2. Bagaimana mengimplementasikan hasil perancangan dengan metode Com HDM pada perangkat lunak sistem *marketplace* pemesanan *travel* berbasis *web*?
3. Bagaimana skenario dan proses pengujian yang akan dilakukan dan diterapkan pada perangkat lunak sistem *marketplace* pemesanan *travel* berbasis *web* yang menggunakan metode perancangan Com HDM sehingga didapat hasil pengujian sesuai yang diharapkan?

1.3 Tujuan

Tujuan dari pengembangan perangkat lunak ini adalah untuk mengimplementasikan hasil perancangan metode *comprehensive hypermedia design* (Com HDM) pada rancang bangun sistem *marketplace* pemesanan *travel* berbasis *web* yang selanjutnya akan dilakukan pengujian untuk mengetahui sistem berjalan sesuai dengan yang di harapkan.

1.4 Manfaat

Adapun manfaat yang dapat diberikan setelah penelitian ini dilakukan antara lain:

1. Vendor *travel* dapat memberikan informasi dan mengelola jadwal melalui *website marketplace* pemesanan *travel*.
2. *Customer* dapat melakukan pemesanan *travel* melalui *website* dengan mudah.
3. *Customer* dapat mengetahui informasi *travel shuttle* dengan mudah karena sistem *marketplace* pemesanan *travel* adalah sistem terpadu.

1.5 Batasan Masalah

Agar diperoleh hasil pembahasan yang sesuai dengan apa diharapkan, maka perlu diberikan pembatasan masalah pada pengembangan perangkat lunak ini, yaitu :

1. Pembahasan difokuskan pada rekayasa perangkat lunak dari perancangan sistem dengan metode Com HDM (*Comprehensive Hypermedia Design Method*) pada sistem *marketplace* pemesanan *travel* yang berupa *Web Application*.
2. Pengembangan fitur pada perangkat lunak hanya sebatas pemesanan dan info tentang vendor *travel*.
3. Pengembangan fitur admin pada perangkat lunak hanya sebatas manajemen menghapus *customer* atau vendor *travel* untuk mengendalikan penyalahgunaan perangkat lunak.
4. *Framework PHP* yang dipakai adalah *Laravel*.
5. *Browser* yang dipakai untuk menjalankan dan menguji perangkat lunak perangkat lunak sistem *marketplace travel* berbasis *web* adalah *Google Chrome*.
6. Perancangan perangkat lunak sistem *marketplace travel* berbasis *web* pada fase desain menggunakan metode Com HDM (*Comprehensive Hypermedia Method*).
7. Pengujian fungsional dilakukan dengan pengujian validasi menggunakan pendekatan *black-box testing* dan pengujian *usability* dengan aspek efisiensi kepada pelanggan *travel* dan pihak *travel*.

1.6 Sistematika pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut :

Bab I Pendahuluan

Menguraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

Bab II Dasar Teori

Dasar teori berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian yang berkaitan dengan sistem *marketplace* pemesanan *travel* berbasis *web*

Bab III**Metodologi Penelitian**

Metodologi berisi metode yang lebih spesifik dalam penyelesaian masalah sistem *marketplace* pemesanan travel berbasis *web* dan juga dapat dipahami sebagai sebuah ilmu untuk mempelajari bagaimana sebuah penelitian dilakukan secara sistematis.

Bab IV**Analisis Dan Perancangan**

Pada bagian ini akan diuraikan bagaimana perancangan dan analisis data pada sistem yang akan di buat.

Bab V**Implementasi**

Pada bagian ini diuraikan bagaimana implementasi pembuatan aplikasi yang dilakukan berdasarkan perancangan dari sistem yang dibuat.

Bab VI**Pengujian**

Bagian ini berisikan pengujian dan analisa dari hasil pengujian terhadap perangkat lunak sistem *marketplace* pemesanan travel berbasis *web*.

Bab VII**Penutup**

Bagian ini memuat kesimpulan dan saran terhadap penelitian. Kesimpulan dan saran perangkat lunak sistem *marketplace* pemesanan travel berbasis *web* disajikan secara terpisah.



BAB 2 DASAR TEORI

Dasar teori berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Terdapat landasan teori dari sumber yang terkait dengan teori dan metode yang digunakan dalam penelitian implementasi metode Com HDM (*Comprehensive Hypermedia Method*) pada sistem marketplace pemesanan travel.

2.1 Kajian Pustaka

Penelitian terdahulu yang berjudul “Penerapan Metode Com HDM (*Comprehensive Hypermedia Design Method*) untuk Perpustakaan *online* (e-Library)” (Jantan, 2013) digunakan sebagai acuan pengembangan penelitian saat ini. Pada penelitian tersebut dijelaskan beberapa masalah yaitu penggunaan UML *Activity Diagram* untuk mendukung pandangan interaksi saat ini masih kurang fasilitas pemodelan. Untuk menyajikan bagaimana proses interaksi harus digambarkan dalam model desain navigasi. Selain itu, masalah fungsi navigasi juga disorot. Beberapa masalah fungsi navigasi yang tidak sesuai menjadi bahasan utama seperti *user* gagal untuk menemukan informasi yang diperlukan, *user* terkadang kebingungan dan tersesat di dalam sebuah *website* yang memiliki struktur halaman dan *sitemap* yang kompleks sehingga navigasi terkadang kurang efektif dalam merepresentasikan *sitemap website* yang kompleks tersebut, dan *user* mungkin memerlukan waktu ekstra untuk menemukan informasi yang diperlukan. Ini terjadi karena analisa dalam *hyperlink* dan struktur navigasi yang berantakan karena kurangnya fasilitas desain untuk menggambarkan dari elemen akses dalam metode desain saat ini. Oleh karena itu penulis menerapkan metode Com HDM sebagai fasilitas dalam melengkapi perancangan untuk menganalisa menggambarkan struktur navigasi dan *hyperlink* perpustakaan *online* (e-Library).

Dalam penelitian ini, penulis mencoba menerapkan metode yang telah disebutkan diatas yaitu Com HDM untuk sistem *marketplace* pemesanan travel sehingga permasalahan yang telah disebutkan diatas sebelumnya bisa diminimalisir atau bahkan dihindari serta untuk mendapatkan solusi atas navigasi perangkat lunak yang lebih informatif dan memaksimalkan pengalaman *user*.

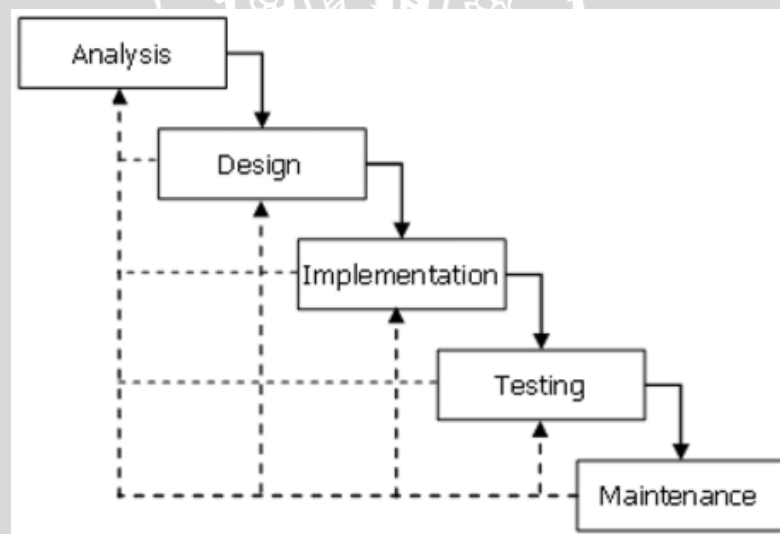
2.2 Rekayasa Perangkat Lunak (RPL)

Software Engineering adalah istilah yang lebih populer dibandingkan RPL. Tepatnya 1968 istilah ini mulai populer pada Konferensi Software Engineering yang digelar oleh *North Atlantic Treaty Organization*. Beberapa orang berasumsi bahwa RPL hanya membuat perangkat lunak komputer dan ilmu yang terbatas. Padahal terdapat perbedaan antara software dan program pada komputer. Salah satu perbedaan pada software adalah perintah-perintah yang dipakai untuk mengolah informasi dan bisa berupa program atau prosedur. Untuk program sendiri merupakan perintah yang di susun dan hanya dipahami komputer. Sedangkan prosedur sendiri merupakan perintah yang digunakan oleh pengguna dalam

memproses informasi. Rekayasa perangkat lunak menganut pemahaman yang terstruktur cara inilah yang paling efektif untuk menghasilkan perangkat lunak dengan memiliki kualitas tinggi karena pekerjaan yang mereka lakukan teratur. Metode yang dipilih juga pelajari dalam ilmu RPL. Para analis dan programmer akan memilih metode yang sesuai dengan kebutuhan perangkat lunak (Sommerville,2003).

2.3 Software Development Life Cycle (SDLC) Waterfall

Model *Waterfall* SDLC adalah proses sekuensial pengembangan perangkat lunak, dimana dalam prosesnya mengalir semakin ke bawah (seperti air terjun) melalui daftar tahapan yang harus dijalankan agar dapat membangun sebuah perangkat lunak komputer. Pada tahun 1970 Winston W.Royce mengusulkan model *Waterfall* untuk menggambarkan praktek rekayasa perangkat lunak. Model *Waterfall* mendefinisikan beberapa fase berurutan yang harus diselesaikan satu demi satu dan pindah ke tahap berikutnya hanya ketika fase sebelumnya yang benar-benar dilakukan. Untuk alasan ini, model *Waterfall* adalah rekursif di setiap fase dapat diulang tanpa henti sampai disempurnakan. Menggambarkan fase yang berbeda dari model SDLC *Waterfall* (Nursari,2012).



Gambar 2.1 Model SDLC *Waterfall*

2.4 Unified Modelling Language

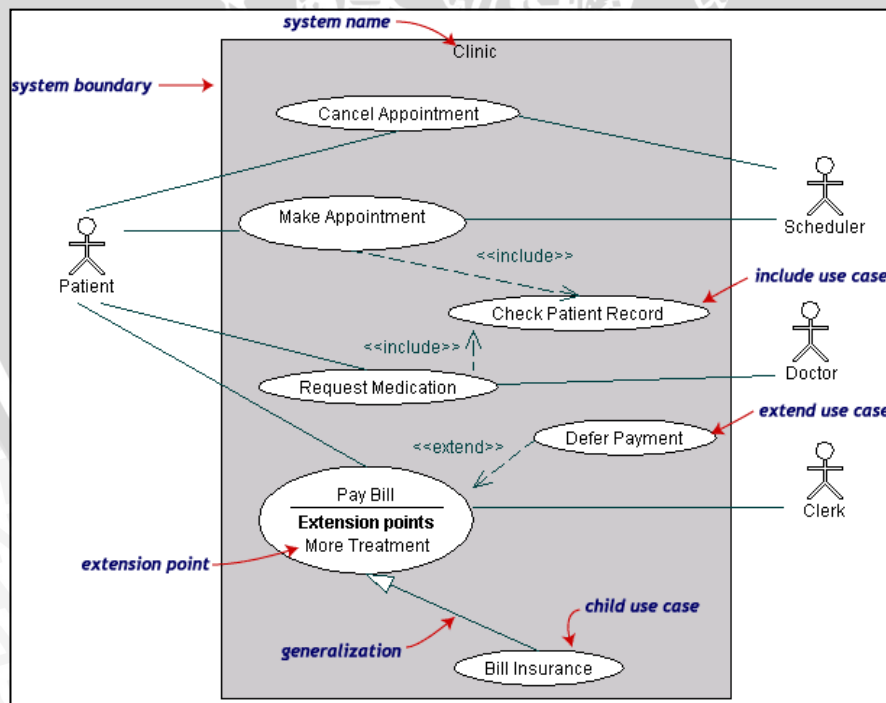
Model adalah bentuk sederhana dari kenyataan yang menyediakan penjelasan lengkap dari beberapa sudut pandang. Kita membangun sebuah model untuk dapat lebih memahami sistem yang akan kita buat. *Modelling* sangat penting karena membantu tim dalam memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan struktur dan *behavior* dari arsitektur sistem. Menggunakan bahasa *modelling* standard seperti UML (*Unified Modelling Language*), dapat membantu komunikasi antar anggota tim sehingga menghasilkan keputusan yang tidak ambigu (Sommerville,2011).

Menggunakan alat *visual modelling* dapat memfasilitasi manajemen model dan merawat konsistensi antara sistem artefak: *requirements*, *designs*, dan *implementations*. Kesimpulannya, *visual modelling* membantu meningkatkan kemampuan tim untuk mengatur kompleksitas perangkat lunak. Dalam membangun visual model dari suatu sistem, banyak diagram yang dibutuhkan untuk mewakili sistem, seperti beberapa contoh diagram di bawah ini (Sommerville,2011).

2.4.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apakah yang dapat diperbuat oleh sistem. Sebuah *use case* menggambarkan dan menjelaskan interaksi antara aktor dengan sistem. *Use case* menggambarkan dan menjelaskan sebuah pekerjaan tertentu, misalnya mendaftar ke sistem, masuk ke sistem, dan sebagainya. Sebuah entitas pengguna baik mesin dan manusia yang melakukan interaksi dengan sistem untuk melakukan pekerjaan tertentu dapat disebut sebagai aktor (Dharwiyanti,2003).

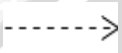
Penggunaan *use case diagram* membantu kita dalam menyusun kebutuhan fungsionalitas dari sistem, dapat digunakan sebagai alat untuk berkomunikasi dengan menunjukan hasil rancangan kepada klien, dan digunakan untuk merancang semua fitur yang ada pada sistem.



Gambar 2.2 Contoh *use case diagram*

Sumber : (Dharwiyanti,2003)

Tabel 2.1 Keterangan simbol pada *use case diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Dapat diartikan sebagai orang yang berinteraksi dengan sistem
3		<i>Generalization</i>	Menunjukkan hubungan umum ke khusus antara dua use case dimana yang satu memiliki fungsi umum
4		<i>Include</i>	Menunjukkan use case juga di jalankan untuk menjalankan use case yang lain
5		<i>Extend</i>	Menunjukkan tambahan use case tanpa menunjukkan adanya syarat tergantung pada use case yang di tambahkan.
6		<i>Association</i>	Untuk menunjukkan hubungan atau interaksi antara aktor dengan use case fungsi.
7		<i>System</i>	Menunjukkan bahwa sekumpulan use case adalah satu kesatuan sistem.
8		<i>Use Case</i>	Menunjukkan deskripsi fungsional dari sistem.

Sumber : (Dharwiyanti,2003)

Sebuah *use case* dapat memiliki kesamaan fungsionalitas dengan *use case* lain sebagai bagian dari proses yang terjadi dalam dirinya. Secara umum diartikan bahwa *use case* yang di miliki fungsinya akan dipanggil setiap kali *use case* yang memiliki fungsionalitas dari use case lain itu dieksekusi secara normal. Sebuah *use case* dapat di miliki fungsinya oleh lebih dari satu *use case*, sehingga use case yang fungsionalitasnya sama dapat dihindari dengan cara menarik keluar fungsionalitas yang sama (Dharwiyanti,2003).

Sebuah *use case* juga dapat memperluas perilaku *use case* lain dengan aturannya sendiri. Sementara spesialisasi *use case* dapat ditunjukan dengan adanya generalisasi dari satu *use case*.

2.4.2 Class Diagram

Dalam *diagram class* pandangan diberikan secara luas dari suatu perangkat lunak yaitu dengan menunjukan hubungan antar kelas mereka. Diagram class

sendiri bersifat statis, dengan digambarkannya keterkaitan kelas yang satu dengan kelas lainnya dengan kata lain untuk mengetahui hubungan apa yang terjadi (Wahono,2003). *Diagram Class* mempunyai 3 macam hubungan, sebagai berikut :

A. *Association*

Sebuah *association* adalah simbol penghubung antar dua kelas dalam diagram. Dalam melakukan kegiatan akan terjadi *association* antara dua kelas jika salah satu bagian dari kelas memiliki hubungan dengan yang lainnya.

B. *Aggregation*

Suatu hubungan yang menunjukkan satu kelasnya adalah bagian dari suatu kumpulan kelas. Pusat yang terdiri dari sekumpulan kelas disebut *Aggregation*. Contohnya : Detail pemesanan terdiri dari banyak pemesanan.

C. *Generalization*

Menunjukkan bahwa suatu kelas itu memiliki kelas turunan dengan satu kelasnya menjadi kelas umum dari kelas yang lain. *Generalization* memiliki hubungan tingkatan lebih khusus dari kelas yang memiliki fungsi sama tetapi lebih umum. Contoh : pembayaran adalah kelas umum dari tunai, transfer melalui bank, dan kredit.

Untuk tambahan bahwa *association* mempunyai 2 titik. Setiap satu titik memiliki penjelasan sendiri-sendiri dari hubungan kelas tersebut, contoh penjelasan titik: *produce* adalah garis vendor travel untuk menghubungkan setiap informasi travel.

Garis dengan panah disebut *navigability* dalam suatu hubungan antar kelas menggambarkan alur. Seperti dalam contoh : Detail pemesanan terdiri dari beberapa item, namun tidak bisa sebaliknya. Panah ini juga menjelaskan siapa “memiliki” implementasi dari *association*, dalam kasus ini Detail pemesanan memiliki beberapa Item. Garis hubungan tanpa adanya panah menunjukan hubungan yang bolak-balik (Wahono,2003).

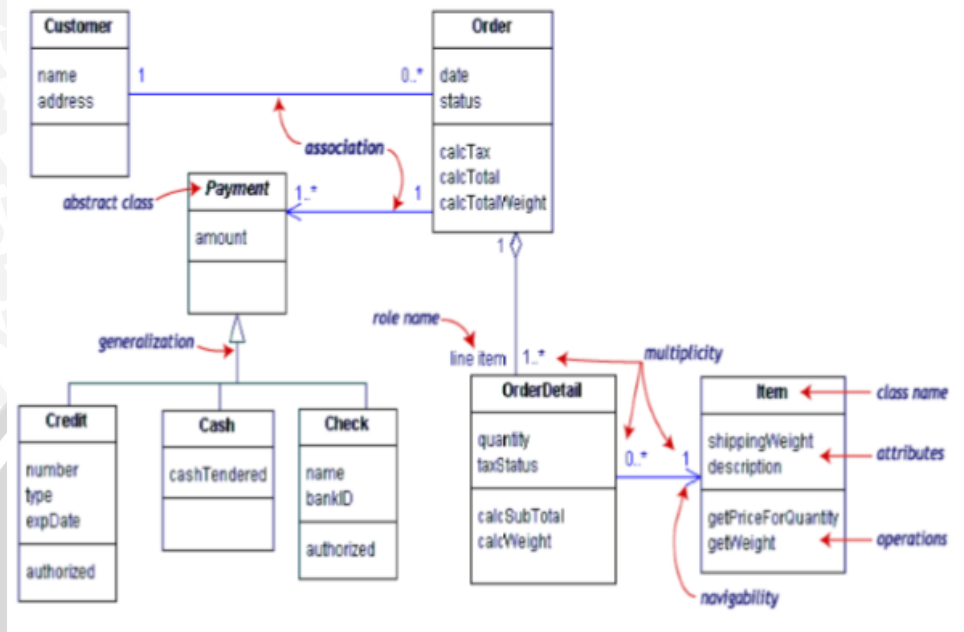
Pada titik garis hubungan juga akan terdapat *multiplicity*. *Multiplicity* berupa *single number* (angka tunggal) atau *range number* (angka batasan). Pada contoh, hanya bisa satu ‘Customer’ untuk setiap ‘Order’, tapi satu ‘Customer’ hanya bisa memiliki beberapa ‘Order’. Tabel dibawah mengenai *multiplicity* yang sering di gunakan :

Tabel 2.2 Multiplicity dan artinya

Multiplicities	Artinya
0..1	Nol atau satu bagian. Notasi n . . m menerangkan n sampai m bagian.
0..* or *	Tak hingga pada jangkauan bagian (termasuk kosong).
0..* or *	Tepat satu bagian
1..*	Sedikitnya hanya satu bagian

Sumber : (Wahono,2003)

Setiap diagram Class memiliki Class (kelas), *association*, dan *multiplicity*. Sedangkan *navigability* (alur arah) dan *role* (kegiatan) merupakan *optional* (tidak diharuskan).



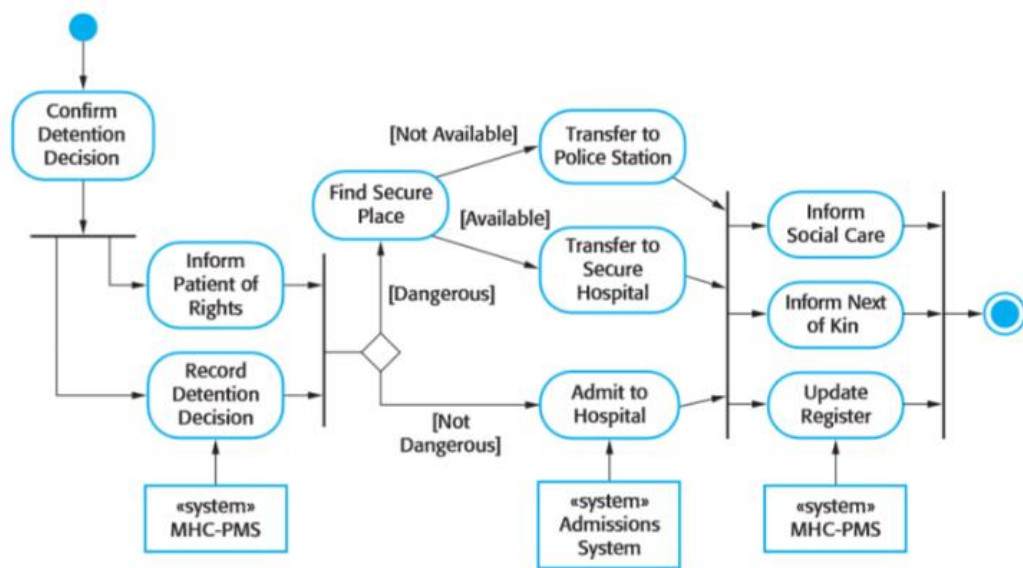
Gambar 2.3 Contoh Class Diagram

Sumber : (Wahono,2003)

2.4.3 Activity Diagram

Gambar 2.4 adalah sebuah *activity diagram* yang ditujukan untuk menunjukkan aktivitas yang membuat sebuah proses sistem dan aliran kontrol dari satu aktivitas ke aktivitas lain. Untuk memulai proses dimulai dengan *filled circle*, dan diakhiri dengan lambang lingkaran dalam lingkaran. Persegi dengan segi tumpul menggambarkan aktivitas.

Dalam UML *activity diagram*, garis panah menggambarkan alur kerja dari aktivitas satu ke aktivitas yang lain. Garis tebal digunakan untuk mengindikasikan koordinasi aktivitas. Saat alur mulai lebih dari satu aktivitas mengarah ke garis tebal maka semua aktivitas harus dilakukan sebelum melanjutkan ke tahap berikutnya. Saat alur dari garis tebal menuju aktivitas tertentu, maka aktivitas tersebut boleh dikerjakan secara paralel (Sommerville,2011).

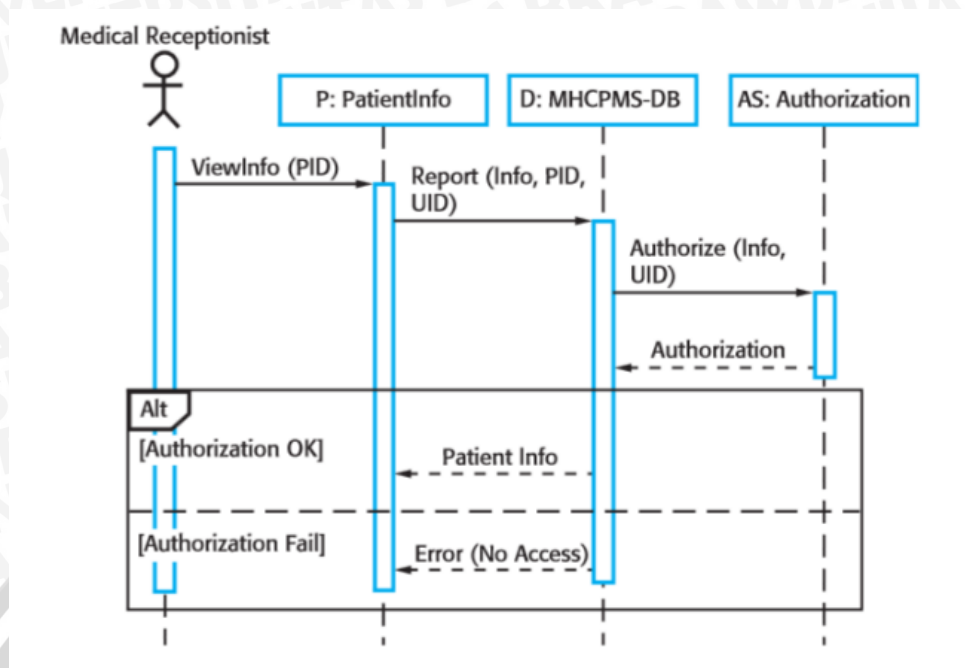


Gambar 2.4 Contoh Activity Diagram

Sumber: (Sommerville,2011)

2.4.4 Squence Diagram

Sequence diagram dalam UML digunakan untuk memodelkan interaksi antara aktor dan objek dalam sistem dan interaksi dengan dirinya sendiri. UML mempunyai banyak *syntax* untuk *sequence diagram* yang memungkinkan berbagai macam interaksi untuk dimodelkan. Seperti namanya, sebuah *sequence diagram* menunjukkan urutan dari interaksi yang digunakan selama *use case* tertentu. Gambar 2.5 adalah sebuah contoh *sequence diagram* yang menggambarkan notasi sederhana. Diagram ini memodelkan interaksi yang terjadi dalam tampilan *use case* informasi pasien, dimana resepsionis dapat melihat informasi pasien.



Gambar 2.5 Sequence Diagram

2.5 MySQL Database Management System

SQL digunakan untuk berkomunikasi dengan database. Menurut American National Standards Institute, SQL adalah bahasa untuk mengatur database yang saling berhubungan dalam sistem komputer. Script dalam bahasa SQL berguna untuk melakukan beberapa tugas seperti memperbarui data, menghapus data dan lain sebagainya. Ada beberapa sistem untuk mengatur data dalam database umum yang menggunakan SQL yaitu Oracle, Sybase, Microsoft SQL Server, dan lain-lain. Meskipun pada sistem mereka menggunakan SQL untuk mengatur data dalam database, banyak juga yang memiliki bahasa *script* tambahan khusus untuk sistem mereka (Saputra,2012). Ada dua jenis *script* perintah dalam SQL yaitu :

1. *Data definition language* (DDL)
Data definition language adalah *script* SQL yang berhubungan langsung dengan pendefinisian struktur database. Beberapa perintah DDL antara lain CREATE, ALTER, RENAME, DROP.
2. *Data manipulation Language* (DML)
Data manipulation language adalah *script* SQL yang berhubungan dengan manipulasi ,pengolahan data dalam tabel. Perintah SQL yang termasuk DML antara lain SELECT, INSERT, UPDATE, DELETE.

2.6 Hyper Text Markup Language (HTML)

Hyper text markup language adalah bahasa pemrograman yang digunakan *programer* untuk membuat halaman web atau lebih di kenal dengan HTML, memiliki fungsi menampilkan berbagai informasi ketika pengguna menggunakan browser untuk melihat informasi dan kumpulan tulisan yang digunakan sederhana yang ditulis kedalam format bahasa ASCII untuk menghasilkan tampilan web dengan menarik dan teratur. Dan dengan bahasa HTML inilah pengguna dapat melihat informasi dalam web yang menarik dan teratur (Klikkanan,2007).

Dokumen HTML hampir terlihat seperti kumpulan tulisan biasa, hanya saja dalam kumpulan tulisan ini sebuah kata atau kumpulan kata bisa memiliki tujuan perintah atau lebih dikenal dengan TAG khusus. Sebagai contohnya apabila *programer* akan membuat tulisan yang ditampilkan pada web menjadi tebal, maka yang harus dilakukan adalah `<b>TEBAL</b>`. Tanda `<b>` memiliki fungsi menjadikan tulisan tebal dalam web, selanjutnya *programer* memasukkan tulisan yang akan di buat tebal, sebagai penutup harus memakai tanda `</b>` agar intruksi bekerja sesuai yang di inginkan.

Secara garis besar, ada empat elemen dalam HTML:

1. *Structural*
Tanda yang menentukan level atau tingkatan dari tulisan yang di masukan contoh, `<h1>Bola</h1>` akan memerintahkan browser untuk menampilkan “Bola” sebagai teks tebal besar yang menunjukkan sebagai Heading 1,
2. *Presentationl*
Tanda ini memiliki fungsi mengganti tampilan dari sebuah atau sekumpulan tulisan dan level dalam tulisan tidak diperdulikan. Seperti `<b>boldface</b>` akan menampilkan **bold**. Tetapi tanda ini sudah mulai tidak digunakan karena digantikan oleh CSS dan tidak dianjurkan untuk mengatur tulisan.
3. *Hypertext*
Tanda yang menghubungkan dengan halaman atau sumber lain
4. *Elemen Widget*
Adalah tag untuk membuat objek-objek selain tulisan. Contohnya untuk tombol (`<button>`), list (`<li>`), dan garis horizontal (`<hr>`).

2.7 Cascading Style Sheet (CSS)

Ini adalah suatu bahasa pemrograman yang berguna bagi *programer* untuk merubah tampilan tulisan atau sekumpulan tulisan yang ditulis dalam tag HTML terlihat rapi dan menarik. Manfaat yang paling banyak dari bahasa pemrograman CSS adalah untuk merubah tampilan halaman web yang ditulis dalam tag HTML dan XHTML. Meskipun begitu, tag dari CSS sendiri dapat derguna untuk semua jenis tag XML 5 termasuk SVG dan XUL. World Wide Web Consortium (W3C) mengatur spesifikasi CSS (Mardiyanto,2011).

CSS digunakan oleh programmer untuk menentukan dan mengatur tampilan web termasuk warna dan jenis huruf agar terlihat rapi dan menarik. CSS biasa digunakan programmer untuk membedakan isi informasi (tag HTML) dengan yang digunakan untuk menampilkan dan merapikan (tag CSS). Perbedaan ini dapat meningkatkan kemudahan dalam melihat atau mengakses isi, serta akan memperlihatkan lebih banyak kemudahan juga dalam melakukan pengaturan pada tampilan. CSS dapat mengurangi atau menghindari pengulangan pada bagian isi.

Pada bahasa CSS, halaman dengan informasi yang sama dapat ditampilkan dan di desain dengan berbeda. Dengan contoh informasi dapat di sampaikan melalui suara atau dapat disampaikan langsung di tampilan pada layar dan lain sebagainya. Begitu juga dengan kumpulan informasi dengan tag HTML atau yang lainnya dapat juga ditampilkan dengan gaya tampilan, huruf, dan warna yang berbeda dengan menggunakan CSS.

Secara tidak langsung dengan menggunakan CSS akan lebih memudahkan dalam melakukan perbaikan dan mengatur ulang tampilan dokumen HTML. Karena antara isi dokumen HTML dan pengaturan format dokumen HTML dipisahkan dengan format yang berbeda. Format penulisan CSS secara umum memiliki dua bagian, yaitu : selector dan declaration. Selector adalah bagian pertama dari penulisan CSS sebelum tanda "{", sedangkan declaration adalah bagian kedua dari CSS yang berada di dalam tanda "{}" (Mardiyanto, 2011).

2.8 Javascript

JavaScript adalah bahasa scripting desainer *Web* yang digunakan untuk mengotomatisasi tugas-tugas dalam halaman *Web*. Sebelum JavaScript, halaman *web* hanya berisi gambar atau file audio. tag HTML bisa menggambarkan jalan, menentukan hal-hal seperti hyperlink, tabel, dan bentuk, dan memberitahu browser *Web* untuk menampilkan isi dari suatu file gambar, film, atau suara. Namun, HTML menyediakan cara untuk memberitahu browser untuk melaksanakan urutan instruksi (Andrianto, 2011).

Akibatnya, tanpa bahasa scripting seperti JavaScript, halaman *Web* tidak memiliki kemampuan pengolahan atau cara untuk berinteraksi dengan pengunjung situs. Misalnya, meskipun pengunjung dapat memasukkan informasi ke dalam bentuk HTML hanya memungkinkan browser mengirim data ke *Web* server, karena HTML tidak memiliki cara untuk memvalidasi atau melakukan hal lain dengan informasi yang pengunjung jenis atau pilihan pengunjung membuat pada formulir.

Karena desainer *Web* perlu cara untuk memberitahu browser *Web* untuk melakukan hal-hal selain menampilkan teks dan gambar dan mengirim data formulir kembali ke server *Web*, Netscape (bekerja sama dengan Sun Microsystems) mengembangkan JavaScript. Meskipun JavaScript adalah bahasa scripting dalam hal itu memungkinkan untuk menentukan satu set perintah yang diinginkan browser *Web* untuk mengeksekusi, JavaScript tidak untuk membuat program eksternal yang berjalan independen dari browser. Dengan demikian, JavaScript disebut script, karena memungkinkan Anda menulis statment (yaitu, satu set instruksi) yang akan diikuti browser.

Untuk menulis JavaScript, tidak dibutuhkan alat khusus, dapat menggunakan editor teks yang sama yang digunakan untuk membuat halaman *Web*. Selain itu, yang paling populer *Web* browser (seperti Netscape Navigator dan Internet Explorer) memiliki built-in mendukung JavaScript. Dengan demikian, pengunjung situs tidak perlu menginstal tambahan perangkat lunak pada komputer mereka, browser dapat mengeksekusi skrip berbasis JavaScript dengan menyisipkan di halaman HTML. Yang harus dilakukan adalah melampirkan laporan JavaScript antara satu set mulai dan script tag akhir (`<script> </script>`), yang akan memberitahu penjelajah informasi pada komputer untuk mengeksekusi laporan bukan menampilkan mereka pada layar.

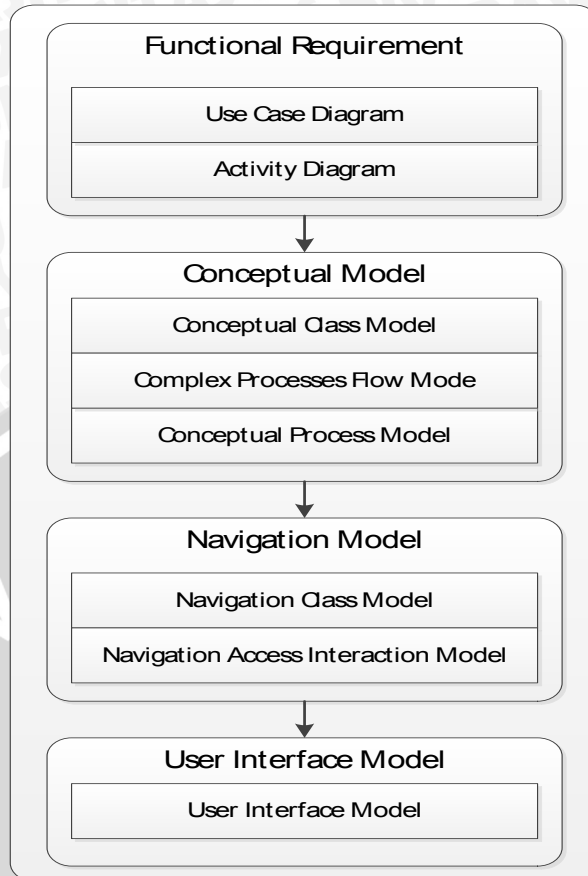
2.9 Com HDM (*Comprehensive Hypermedia Design Method*)

Sebuah metode desain yang komprehensif, sistematis, dan fleksibel untuk desain aplikasi *web*. Ini akan menggabungkan dimensi yang komprehensif dan sistematis dari tahap tiga desain yang disebut desain konseptual, desain navigasi, dan desain antarmuka user (Jantan, 2013). Secara umum, ada tiga tujuan utama dari Com HDM yaitu:

1. untuk model proses yang kompleks melalui desain dan informasi konseptual pemodelan yang sistematis.
2. untuk menyediakan link navigasi dan mekanisme struktur akses untuk mendukung arus proses dan interaksi mereka.
3. untuk menyajikan interaksi antara user *web* dan aplikasi *web* melalui abstrak tata letak antarmuka user.

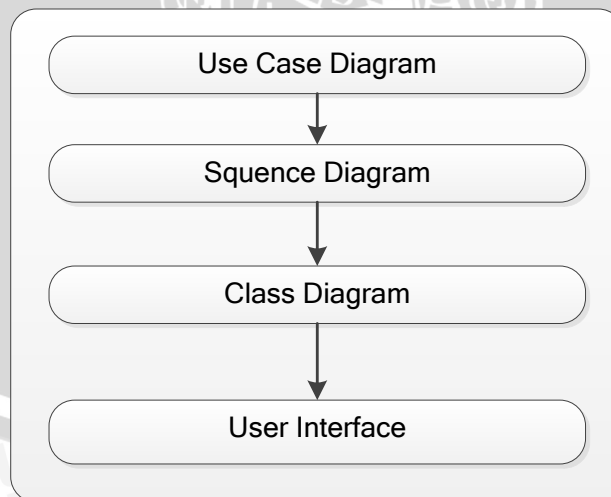
Com HDM dapat diterapkan dalam peningkatan desain iteratif untuk mendukung prinsip desain usability, mengarah ke siklus pengembangan yang sistematis dan fleksibel untuk diterapkan pada aplikasi *web*. Com HDM adalah metode pemodelan berorientasi objek generik berdasarkan *Unified Modeling Language* (UML). Ini adalah model berbasis pendekatan di mana model dibangun secara bertahap dengan desain terpisah untuk memberikan standar hasil yang lebih baik. Com HDM adalah metode desain yang membandingkan dan menggabungkan beberapa aspek desain di sini terbukti dimana metode desain yang ada dapat mengatasi beberapa tantangan utama khususnya dalam proses pemodelan interaksi yang kompleks (Jantan, 2013).

Oleh karena itu, Com HDM harus menghasilkan beberapa ide-ide baru melalui konsep pemodelan dan fitur desain melalui pendekatan metodologis nya. Dibawah ini adalah perbandingan alur metode perancangan Com HDM dengan contoh alur perancangan biasa:



Gambar 2.6 Alur metode perancangan (Com HDM)

Sumber : (Jantan,2013)



Gambar 2.7 Contoh alur perancangan biasa

Sumber : (Safitri,2010)

Tabel 2.3 Keterangan simbol dalam *conceptual class model*

Spesifik (Penulisan) Elemen Model Proses Konseptual		
Nama	Notasi	Deskripsi
Process Class	«process class» △	Didefinisikan sebagai cara yang sama seperti tindakan yang diambil oleh user untuk melakukan suatu kegiatan. The contoh objek dari kelas proses yang digunakan oleh user selama pelaksanaan urutan proses yang telah ditentukan.
Atomic Class	«atomic class» ○	Mewarisi kelas proses. Ini adalah satu proses yang dapat dilakukan dan dilaksanakan secara independen sebagai salah satu operasi / proses.
Non Atomic Class	«non atomic class» ⊙	Mewarisi kelas proses. Pelaksanaan non-atom harus dilakukan secara berurutan (mereka mungkin memiliki dependensi)
Process Container	«pro container» □	Kelompok dan partisi kelas proses dalam rangka untuk menunjukkan hubungan atau dependensi mereka. Hal ini untuk menentukan partisi contoh proses milik.
Process Link	«process link» →	Hubungan antara dua kelas terpisah; kelas konseptual untuk memproses kelas dan sebaliknya. Juga dikenal sebagai link eksternal yang menghubungkan orang-orang kelas melalui navigasi.
Action Link	«action link» - - - - ->	Hubungan antara operasi dalam proses yang sama kelas (kelas proses untuk memproses kelas). Hal ini untuk menunjukkan ketergantungan dari proses dan aliran informasi dalam kelas proses tertentu.
Database Link	«database link» ↔	Hubungan antara kelas konseptual atau kelas proses dan database kelas. Mewakili informasi dan data operasi seperti query, pencarian, masuk, dll
Database Class	«database class» ▣	Model Adanya basis data dalam domain aplikasi. Ini memberikan pandangan logis dari operasi database antara kelas proses (kelas atom dan kelas non-atom) dan kelas database.

Sumber : (Jantan,2013)

Tabel 2.4 Keterangan simbol dalam *navigation class model*

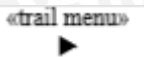
Modeling (Penulisan) Elemen Navigasi Kelas Model		
Nama	Notasi	Deskripsi

Navigation Class	«navigation class» 	kelas konseptual (CCM) yang erat terkait dengan domain aplikasi diturunkan sebagai kelas navigasi. Hal ini menyajikan adanya simpul navigasi di hyperspace navigasi.
Interaction Class	«interaction class» 	Hadir adanya proses interaksi dalam hyperspace navigasi. kelas interaksi harus selalu memiliki setidaknya referensi ke kelas navigasi.
Navigation Link	«navigation link» 	Juga dikenal sebagai hyperlink. Berasal dari asosiasi di Model Kelas Konseptual.

Sumber : (Jantan,2013)

Tabel 2.5 Keterangan simbol dalam *navigation access and interaction model*

Modeling (Penulisan) Elemen Navigasi Access dan Model Interaksi		
Nama	Notasi	Deskripsi
Index	«index» 	Indeks berisi sejumlah item nama terdaftar sewenang-wenang dengan link ke kelas navigasi sasaran. Setiap item memiliki nama yang unik dan memiliki link (target) untuk setiap contoh dari kelas navigasi.
Tree	«tree» 	Pohon memberikan pandangan struktur hirarki dari struktur indeks yang kompleks. Menu dapat diperluas atau runtuh (untuk orientasi terbaik).
Text Query	«text query» 	Item input teks interaktif yang menyediakan sebuah field input untuk mencari informasi. User harus memberikan kata kunci sebagai masukan.
Selectable List	«selectable list» 	Daftar Dipilih menyediakan fasilitas navigasi yang mudah-untuk-go. Beberapa item masukan tercantum disediakan dan mereka dibekukan. Ini memberi bimbingan lebih untuk user <i>web</i> sebagai mekanisme pencarian.
Guided Tours	«guided tours» 	tur menyediakan akses sekuensial memerintahkan ke sejumlah node navigasi atau contoh. Urutan telah ditetapkan untuk menegaskan urutan presentasi kelas navigasi.
Page	«page» 	Halaman menyediakan akses langsung ke sekelompok contoh di kelas navigasi. Halaman item dikelompokkan dan nomor dan masing-masing dari

		mereka memiliki hubungan sendiri ke lokasi target misalnya.
Menu		Menu menyediakan objek komposit (homogen) item menu yang telah tetap nama (beku) dan memiliki link khusus untuk kelas sasaran navigasi, kelas interaksi, atau struktur akses lainnya.
Trail Menu		menu Trail adalah menu dengan beberapa sub-item. Hal ini dapat diperluas dan runtuh dalam rangka memberikan tampilan yang dinamis dan akses yang lebih cepat dari menu.

Sumber : (Jantan,2013)

Tabel 2.6 Keterangan simbol transformasi

Keterangan Simbol Transformasi	
Stereotip Navigasi	Transformasi ke - User Interface Stereotip
«navigation node»	«UIPage» -> «framePage» -> «UIElement»
«interaction node»	«UIInteraction» -> «framePage» «UILogin» -> «framePage» «UISession» «UIElement»
«navigation link»	«accessStructure» «hyperlink» «formElement»
«access structure»	«UIElement» -> «accessStructure» «multimediaElement» «formElement»

Sumber : (Jantan,2013)

Tabel 2.7 Keterangan simbol dalam *user interface*

UI Stereotip	Deskripsi
«UIPage» User Interface Page	Model penyajian node navigasi. Bereaksi sebagai halaman yang dilihat informasi, hyperlink, dan struktur akses. Hal ini dapat dipartisi untuk mencari atau memisahkan informasi spesifik untuk tujuan mudah dilihat.
«UIInteraction» User Interface Interaction	Interaksi UI bereaksi sebagai halaman yang berisi proses interaksi. Ini adalah node tertentu yang memandang bagaimana transaksi dan operasi dapat dilakukan.
«framePage» Frame Webpage	Bingkai halaman partisi <i>web</i> ke daerah presentasi yang berbeda. Ini memisahkan halaman <i>web</i> untuk berpindah

	informasi untuk sederhana dan dipisahkan melihat tampilan.
«UILogin» Login Interface	Ini adalah node tertentu untuk memulai proses interaktif yang memerlukan identifikasi user <i>web</i> 'atau otentikasi.
«UISession» Session Interface	proses interaksi harus disimpan sebagai sesi. Sesi memberikan notasi untuk berhubungan setiap Halaman yang terlibat dalam sesi interaksi tertentu.
«UIElement» User Interface Element	UIElement mengumpulkan semua elemen antarmuka yang berkontribusi pada pembangunan sebuah halaman <i>web</i> . Hal ini disusun oleh tiga kelompok stereotip, yaitu Access Struktur, Multimedia Element, dan Form Element.

Sumber : (Jantan,2013)

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan elemen yang kritis dari SQA dan merepresentasikan tinjauan ulang yang menyeluruh terhadap spesifikasi, desain dan pengkodean. Uji coba merepresentasikan *error* atau kesalahan yang terjadi pada pengembangan software. Pada awal pembuatan, *developer* mengusahakan membuat perangkat lunak mulai perancangan yang abstrak sampai implementasi sesuai kebutuhan. Para pengembang membuat serangkaian uji kasus yang bertujuan untuk "membongkar" software yang mereka bangun. Pengembang software secara alami merupakan orang konstruktif. Developer melakukan uji coba adalah untuk melihat software yang dibuat memenuhi kebutuhan dan memiliki fungsi sesuai sebagaimana mestinya untuk meminimalisir kesalahan yang terjadi. Beberapa aturan tujuan ujicoba telah ditetapkan di buku yang ditulis oleh Glen Myers (Pressman,2010).

1. Ujicoba merupakan proses eksekusi program dengan tujuan untuk menemukan kesalahan.
2. Apabila telah dilakukan uji coba dan terjadi kemungkinan yang besar dalam menemukan keanehan-keanehan pada perangkat lunak yang belum terjadi,itu adalah uji coba yang baik.
3. Dan untuk kesuksesan uji coba ketika dapat menemukan keanehan yang belum terjadi.

2.10.1 Black Box Testing

Dalam metode ini pengembang software menitik beratkan pada pengujian fungsi perangkat lunak. Karena itu semua fungsi perangkat lunak dicoba sesuai dengan *standard* ujicoba blackbox dengan memasukan banyak kondisi oleh pengembang software. Blackbox dapat digunakan untuk melengkapi ujicoba whitebox dalam hal mencari keanehan software. Ada beberapa kategori untuk

berusaha mencari keanehan ketika dilakukan ujicoba blackbox, diantaranya (Pressman,2010):

1. Ketika fungsi itu tidak sesuai
2. Permasalahan pada UI
3. Permasalah pada pusat data atau pusat data dari luar yang di hubungan
4. Permasalah pada kecepatan dan ketepatan guna
5. kesalahan menghubungkan dan memutus sesi pengguna

2.10.2 Usability Testing

Proses pembelajaran mengenai pengguna dengan mengobservasi mereka menggunakan produk untuk mencapai tujuan tertentu yang membuat mereka tertarik (Rubin,2008). *Usability testing* dapat meliputi:

1. Mudah dipelajari
2. Mudah Digunakan
3. Menyenangkan
4. Intuitif

Pengujian ini memiliki fokus dalam mengobservasi pengguna dengan menguji menggunakan tugas yang sebenarnya. Pengujian melakukannya menggunakan dua cara yaitu:

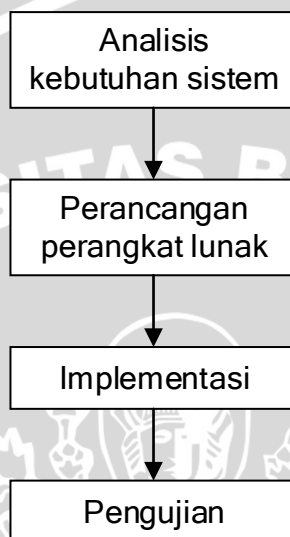
1. Pengujian formatif: dilakukan pada selama program sedang dibuat.
2. Pengujian summatif: dilakukan pada setelah program selesai dibuat.

2.11 Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum dan banyak digunakan dalam kuesioner riset berupa survei. Banyak penelitian berjenis survei deskriptif (gambaran) sering menggunakan skala ini. Pertanyaan yang di tanggap di dalam skala Likert oleh responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan yang di lontarkan dengan memilih salah satu dari pilihan jawaban yang tersedia. Skala Likert juga dapat digunakan untuk mengukur persepsi atau pendapat, sikap, perseorangan atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti (Janti,2014).

BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian membahas metode dan tahapan yang digunakan dalam pengembangan aplikasi sistem *marketplace* pemesanan *travel* berbasis *web*. Metode metode yang digunakan mengacu pada *software development life cycle waterfall* analisis kebutuhan sistem, perancangan perangkat lunak, implementasi, pengujian.



Gambar 3.1 *Flowchart* metode penelitian

3.1 Analisis Kebutuhan Sistem

Kegiatan analisis kebutuhan perangkat lunak meliputi analisis spesifikasi perangkat lunak. Metode analisis menggunakan bahasa pemodelan UML (*Unified Modeling Language*) *Use Case Diagram* dan *Activity Diagram*. *Use Case Diagram* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *user*. *Activity Diagram* digunakan untuk menggambarkan alir aktivitas dari setiap *use case* dalam sistem, bagaimana setiap alir berawal, keputusan yang mungkin terjadi, dan bagaimana akhirnya. Kebutuhan fungsional yang nantinya akan disediakan oleh aplikasi ini antara lain adalah :

1. Aplikasi harus mengimplementasi semua fitur yang telah dibuat sebelumnya berdasarkan acuan standar pemesanan *travel*. Berdasarkan pada literatur yang telah kami pelajari sebelumnya, akhirnya kami menentukan beberapa fitur yang akan diimplementasikan pada aplikasi kami. Berikut daftar fitur yang akan diimplementasikan pada aplikasi sistem pemesanan *travel* dilihat pada tabel 3.1.

Tabel 3.1 Daftar fitur implementasi

No.	Fitur	Penjelasan
1.	Sistem Pemesanan <i>Travel</i>	Menyediakan sistem standar pemesanan <i>travel</i> .
2.	Manajemen Jadwal	Dapat melihat jadwal yang tersedia untuk <i>customer</i> , dan untuk vendor dapat mengatur jadwal.
3.	Manajemen <i>Travel</i>	Mengetahui <i>vendor travel</i> yang terdaftar pada sistem.
4.	Manajemen <i>Customer</i>	Sistem dapat mengetahui <i>Customer</i> yang terdaftar.
5.	Melihat <i>order</i>	Sistem dapat memberikan informasi <i>order</i>
6.	Konfirmasi pemesanan	<i>Customer</i> mendapatkan konfirmasi pemesanan <i>travel</i> .

- Aplikasi tidak menyediakan fitur manajemen pembayaran sehingga proses pemesanan hanya sampai menampilkan verifikasi pemesanan. Selain itu sistem juga belum mengimplementasi sepenuhnya konsep POS (*Point of Sale*) dan manajemen travel maupun *customer* secara penuh. Aplikasi hanya akan mengadaptasi sesuai kebutuhan dan tujuan utama dari penelitian dan pembangunan aplikasi ini.

3.2 Perancangan Perangkat Lunak

Perancangan sistem berfungsi untuk mendefinisikan secara rinci kebutuhan dari aplikasi sistem pemesanan travel berbasis *web* sebagai acuan pembangunan aplikasi. Perancangan sistem menggunakan metode Com HDM (*Comprehensive Hypermedia Design Method*) dan ditambah dengan *sequence* diagram. Ada beberapa tahap dalam metode Com HDM yaitu :

- Functional Requirement*

Dalam *Functional Requirement* ini akan menggunakan *use case* dan *activity* diagram dimana bertujuan untuk mengetahui gambaran *user* sistem dan fitur apa yang akan dibuat dalam sistem.

2. *Conceptual Model*

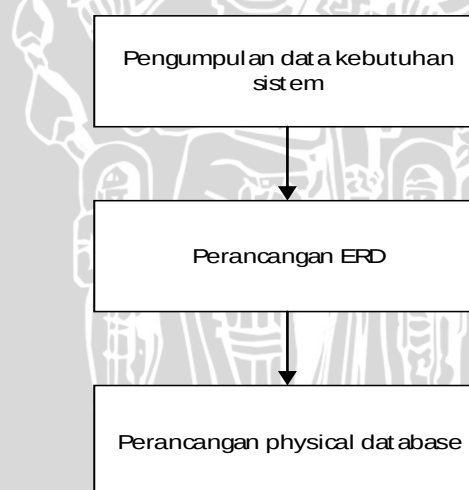
Conceptual Model akan diawali pembuatan CCM (*Conceptual Class Model*) yang terdiri dari *activity* diagram dan *class* diagram dengan menggunakan simbol-simbol *standart* untuk merancang struktur sistem. Dan dilanjutkan CPM (*Conceptual Process Model*) dengan cara merancang fitur kelas terpisah dan proses kompleks yang mengalir dalam satu proses tampilan desain.

3. *Navigational Model*

Idenya adalah bagaimana menyediakan mekanisme navigasi untuk mengakses informasi di *web*. Dalam *Navigational Model* terdiri dari NCM (*Navigation Class Model*) yang harus menyajikan kelas navigasi dan interaksi melalui *hyperlink*. Ini menjelaskan informasi apa yang *hyperlink* tawarkan. Model ini belum selesai, dan harus di tambah struktur akses navigasi. Dilanjutkan NAIM (*Navigation Access and Interaction Model*). Pemodelan ini dibagi menjadi dua bagian yang terpisah pertama menyediakan menu elemen akses, dan yang kedua menyediakan struktur akses.

4. Perancangan Data

Data yang dibutuhkan dalam penelitian ini adalah data perusahaan penyedia layanan *travel* dan data *customer*. Data dirancang menurut kebutuhan untuk dibuat diagram *physical database*.



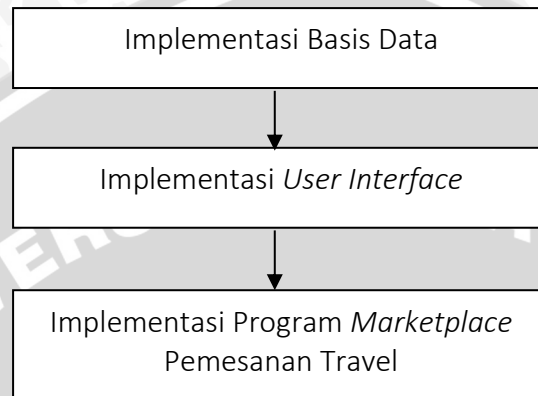
Gambar 3.2 Gambaran umum perancangan data

5. *User Interface Model*

Yang terakhir adalah UID (*User Interface Design*). Tujuan utama adalah untuk memberikan model grafis presentasi dan elemen setiap halaman *web*. Yang perlu diperhatikan pada desain struktur yaitu desain yang mengatur objek UI dalam dimensi tata letak dua halaman. Tetapi ini tergantung pada karakteristik sistem yang dibutuhkan.

3.3 Implementasi

Implementasi perangkat lunak sesuai dengan perancangan yang dijelaskan pada sub bab analisis dan perancangan perangkat lunak. Implementasi perangkat lunak ini berbasis *website* dengan menggunakan bahasa pemrograman *PHP* dengan *framework Laravel*, basis data dengan *MySQL-XAMPP*, dan *javascript* dengan library *Jquery* dan *Ajax*. Implementasi ini meliputi:



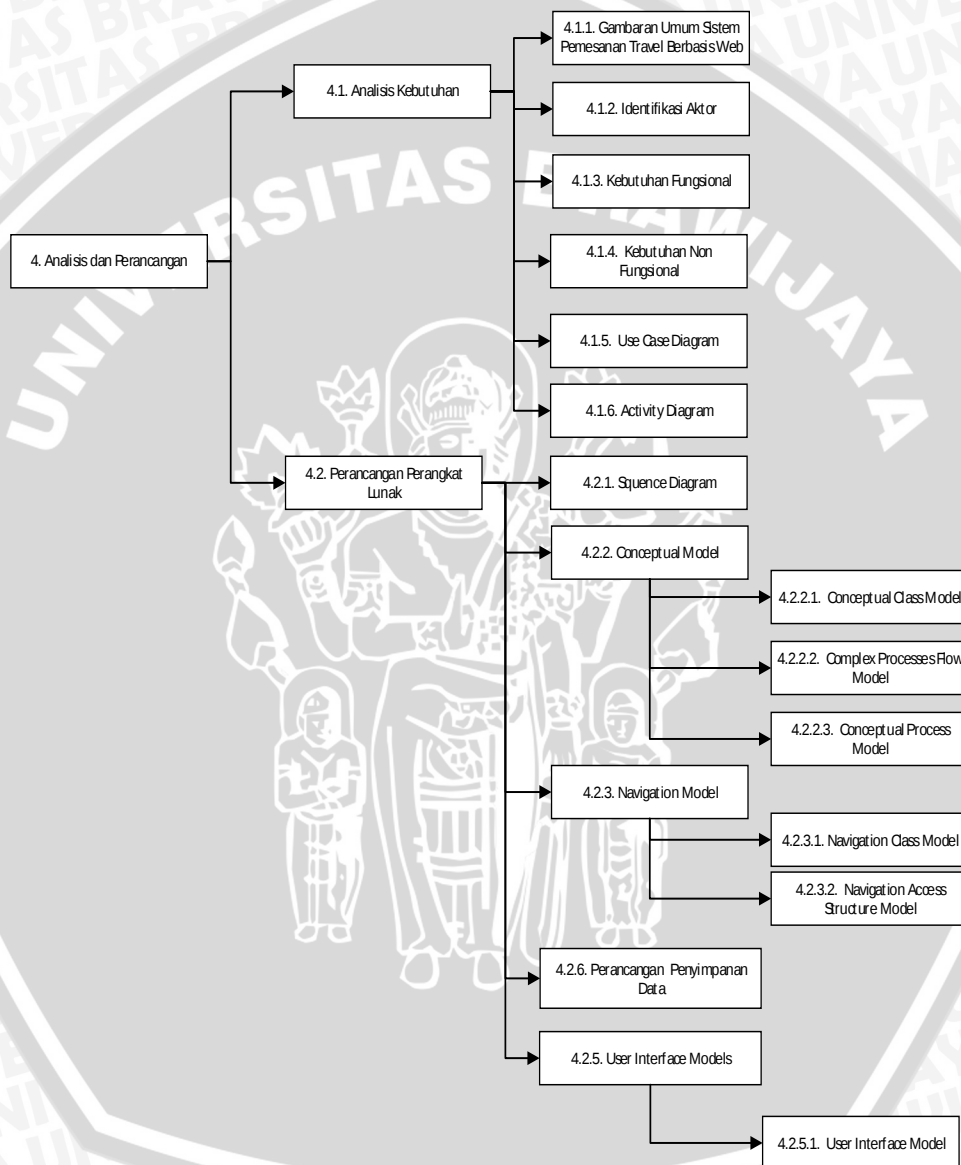
Gambar 3.3 Gambaran umum implementasi aplikasi sistem *marketplace pemesanan travel*

3.4 Pengujian

Pengujian sistem dilakukan untuk menguji fungsionalitas, apakah sesuai dengan kebutuhan yang diinginkan. Strategi pengujian fungsionalitas yang digunakan adalah pengujian validasi. Uji validasi berfungsi untuk menguji kerja setiap fitur aplikasi. Uji validasi menggunakan pendekatan *blackbox*. *Blackbox* bertujuan untuk menguji masukan dan keluaran data apakah sesuai dengan harapan. Dan strategi pengujian non fungsionalitas menggunakan uji *usability* untuk mengetahui apakah perangkat lunak memudahkan *user*.

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak sistem *marketplace* pemesanan *travel* menggunakan metode Com HDM (*Comprehensive Hypermedia Design Method*) Perancangan dilakukan dengan diagram perancangan yang digambarkan pada gambar 4.1.



Gambar 4.1 Diagram perancangan

Tahap analisis kebutuhan terdiri atas enam langkah tahap. Tahap pertama melakukan penjabaran tentang gambaran umum perangkat lunak sistem *marketplace* pemesanan *travel* berbasis web. Pada tahap kedua melakukan proses identifikasi aktor yang terlibat dalam sistem perangkat lunak.

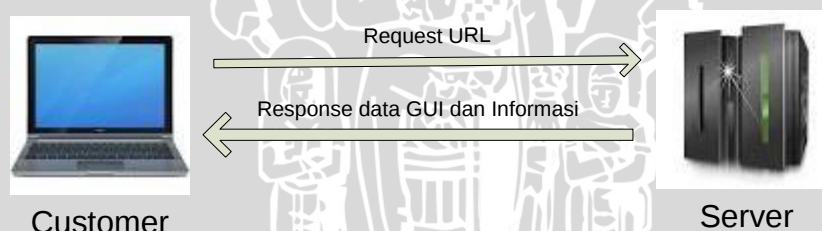
Perancangan perangkat lunak memiliki lima tahap, yaitu perancangan *use case diagram*, *activity diagram*, *conceptual design*, *navigation models*, perancangan dan penyimpanan data, tahap terakhir yaitu perancangan *user interface models*.

4.1 Analisis Kebutuhan

Tahap ini mengarah pada gambaran secara umum dari sistem *marketplace* pemesanan travel berbasis *web* dan hal ini bertujuan untuk pengumpulan, pemahaman dan keputusan fitur apa yang akan didapatkan oleh user. Tahap pertama adalah dengan menggambarkan secara umum perangkat lunak *marketplace* pemesanan travel berbasis *web*, analisis user yang menggunakan perangkat lunak disebut juga aktor, identifikasi data yang akan disimpan, penjabaran tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case* sebagai awal perancangan menggunakan Com HDM (*Comprehensive Hypermedia Design Method*). Tujuan dari analisis kebutuhan untuk menggambarkan kebutuhan yang harus disediakan oleh perangkat lunak supaya sesuai dengan kebutuhan user.

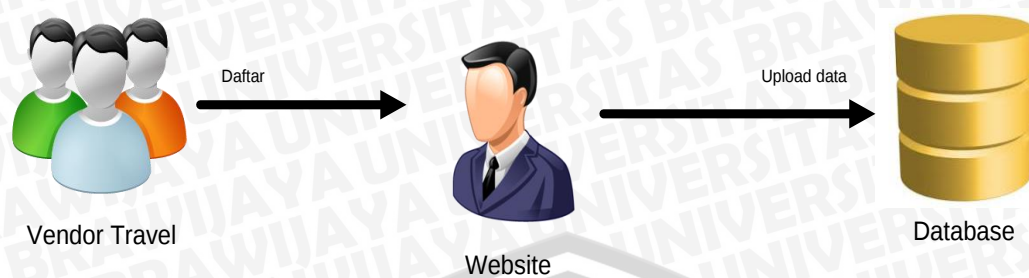
4.1.1 Gambaran Umum Perangkat Lunak

Fungsi perangkat lunak ini membantu menghubungkan antar vendor dan *customer* travel untuk menyampaikan informasi serta sistem pemesanan. Karena sistem yang ada saat ini berupa sistem konvensional. Masih banyak yang belum memanfaatkan *website* ataupun *blog*. Gambar 4.2 menunjukkan bagaimana sistem memberikan informasi kepada *customer*.



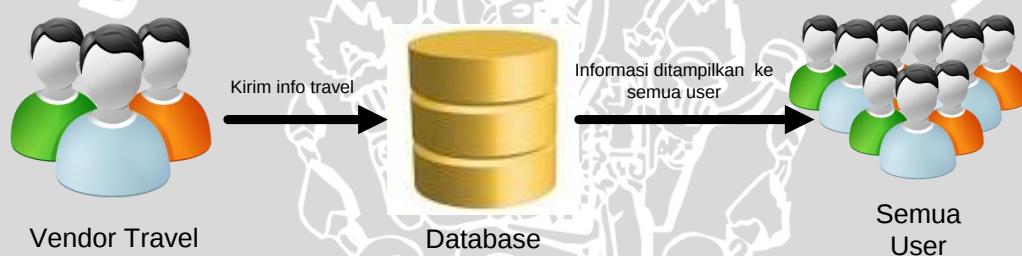
Gambar 4.2 Diagram blok sistem *marketplace* pemesanan *travel* berbasis *website*

Sistem memiliki fitur untuk vendor *travel* mendaftar pada sistem *marketplace* pemesanan *travel* berbasis *web*. Vendor *travel* yang telah memiliki akun dapat mengunggah foto dan memberikan informasi terkini tentang *travel* serta mengatur jadwal. Pada sistem pemesanan *travel* berbasis *website* informasi terkini dan akan ditampilkan pada halaman *website*. Diagram blok alur pendaftaran vendor *travel* pada sistem *marketplace* pemesanan *travel* berbasis *website* dijelaskan pada gambar 4.3.



Gambar 4.3 Diagram blok pendaftaran vendor pada sistem *marketplace* pemesanan *travel*

Sistem *marketplace* pemesanan *travel* merupakan pengembangan dari sistem konvensional *vendor travel* dalam menyampaikan informasi serta pemesanan. Sistem ini mengimplementasi skenario dari sistem konvensional pemesanan *travel*. Sistem *marketplace* pemesanan *travel* merupakan aplikasi yang memanfaatkan *website* dalam penyebaran informasi. Diagram blok skenario penyebaran informasi dijelaskan pada gambar 4.4.



Gambar 4.4 Skenario penyebaran informasi sistem *marketplace* pemesanan *travel*

Aplikasi web ini akan mengimplementasi fitur dari sistem konvensional pemesanan serta penyampaian informasi *travel* yang telah dijelaskan pada bab 3 sub bab 3.2 tentang analisis kebutuhan dengan beberapa pengembangan. Aplikasi ini memiliki 5 fitur utama yaitu: *register*, pemesanan *travel*, admin, *list vendor travel*, dan *travel publisher*.

1. Register

Fitur *register* ini dibagi menjadi dua yaitu :

a. Register Customer Baru

Daftar pelanggan baru berfungsi untuk *customer* yang belum terdaftar pada sistem. *Customer* diharuskan mendaftar untuk memesan tiket *travel*. Fitur ini juga mempermudah admin dalam mengontrol data *customer*.

b. *Register Vendor Baru*

Daftar vendor baru selain berfungsi untuk mendaftarkan vendor *travel* baru, fitur ini juga berfungsi untuk mempermudah admin dalam mengontrol vendor *travel*.

2. *Pemesanan Travel*

Pemesanan *travel* merupakan fitur yang memberikan layanan bagi *customer* yang telah terdaftar melalui *log in username* untuk memesan tiket mobil *travel* yang diinginkan dengan inputan teks berdasarkan penjemputan, tujuan, waktu, dan jumlah penumpang yang selanjutnya akan mendapatkan *form* konfirmasi pemesanan.

3. *Admin*

Admin merupakan fitur untuk admin sistem dengan memasukkan *id username* dan selanjutnya masuk ke halaman admin yang berisi list vendor *travel* dan pelanggan yang telah terdaftar dalam sistem pemesanan *travel*. Fitur ini juga digunakan untuk mengontrol data vendor *travel* dan pelanggan yang terdaftar.

4. *List Vendor Travel*

Fitur ini dapat digunakan pemesan untuk melihat list vendor *travel* yang terdaftar pada sistem pemesanan *travel* yang juga berisikan harga dan informasi tentang vendor *travel*.

5. *Travel Publisher*

Fitur *travel publisher* berisi informasi tentang *publisher web* yang bekerja sama.

4.1.2 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor yang akan berinteraksi dengan perangkat lunak sistem *marketplace* pemesanan *travel*. Tabel 4.1 memperlihatkan aktor – aktor yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
<i>Admin</i>	Admin adalah user <i>administrator</i> perangkat lunak sistem pemesanan <i>travel</i> yang memiliki fungsi menerima vendor yang mendaftar dan mengontrol seluruh akun baik akun pelanggan maupun akun vendor <i>travel</i> .
<i>Vendor Travel</i>	<i>Vendor travel</i> adalah user sistem pemesanan <i>travel</i> yang berperan sebagai penyedia layanan <i>travel</i> dengan cara melakukan pendaftaran dan mengisi identitas yang diperlukan.
<i>Customer</i>	<i>Customer</i> atau pelanggan adalah user sistem pemesanan <i>travel</i> yang dapat memanfaatkan jasa <i>travel</i> dengan cara melakukan

	pendaftaran dan mengisi informasi yang diperlukan untuk mencari <i>travel</i> yang diinginkan.
--	--

4.1.3 Kebutuhan Fungsional

Gambaran umum perangkat lunak yang dijelaskan pada 4.1.3 sebelumnya yang disimpulkan menjadi kebutuhan fungsional. Berikut ini adalah Tabel 4.2 yang berisi kebutuhan fungsional dan penjabarannya. Pada Tabel 4.2 ini menghubungkan antara kebutuhan fungsional dan fitur-fitur pada aplikasi yang digambarkan di *use case* diagram Gambar 4.5.

Tabel 4.2 Kebutuhan Fungsional

ID Kebutuhan Fungsional	Kebutuhan	Use Case ID	Use Case
KF-01	Perangkat lunak harus bisa melakukan <i>register customer</i>	UC-01	Melakukan <i>Register</i>
KF-02	Perangkat lunak harus bisa melakukan <i>register vendor travel</i>		
KF-03	Perangkat lunak harus mampu melakukan <i>login admin</i>	UC-02	Melakukan <i>Login</i>
KF-04	Perangkat lunak harus mampu melakukan <i>login customer</i>		
KF-05	Perangkat lunak harus mampu melakukan <i>login vendor travel</i>		
KF-06	Perangkat lunak harus mampu melakukan pemesanan <i>travel</i> untuk <i>customer</i>	UC-03	Melakukan <i>Pemesanan travel</i>
KF-07	Perangkat lunak harus mampu memberikan konfirmasi pemesanan kepada <i>customer</i>	UC-04	Melakukan <i>Konfirmasi pemesanan</i>
KF-08	Perangkat lunak harus mampu memberikan informasi order untuk <i>customer</i>	UC-05	Lihat <i>order</i>
KF-09	Perangkat lunak harus mampu memberikan informasi order untuk <i>vendor travel</i>		

KF-10	Perangkat lunak harus bisa memberikan informasi <i>travel publisher</i> untuk <i>customer</i>		
KF-11	Perangkat lunak harus bisa memberikan informasi <i>travel publisher</i> untuk <i>vendor travel</i>	UC-06	Lihat <i>Travel publisher</i>
KF-12	Perangkat lunak harus bisa memberikan informasi <i>travel publisher</i> untuk admin		
KF-13	Perangkat lunak harus bisa lihat keseluruhan <i>vendor travel</i> yang terdaftar untuk <i>customer</i>		
KF-14	Perangkat lunak harus bisa lihat keseluruhan <i>vendor travel</i> yang terdaftar untuk <i>vendor travel</i>	UC-07	Lihat <i>List vendor travel</i>
KF-15	Perangkat lunak harus bisa lihat keseluruhan <i>vendor travel</i> yang terdaftar untuk admin		
KF-16	Perangkat lunak harus bisa melakukan <i>edit</i> profil untuk mengupdate data <i>customer</i>		
KF-17	Perangkat lunak harus bisa manage jadwal <i>vendor travel</i>	UC-09	Mengatur jadwal
KF-18	Perangkat lunak harus bisa mengelola akun <i>vendor travel</i> dan <i>customer</i>	UC-10	Mengelola akun
KF-19	Perangkat lunak harus bisa melakukan <i>logout</i> untuk <i>customer</i>	UC-11	Melakukan <i>Logout</i>

KF-20	Perangkat lunak harus bisa melakukan <i>logout</i> untuk vendor <i>travel</i>		
KF-21	Perangkat lunak harus bisa melakukan <i>logout</i> untuk admin		

4.1.4 Kebutuhan Non Fungsional

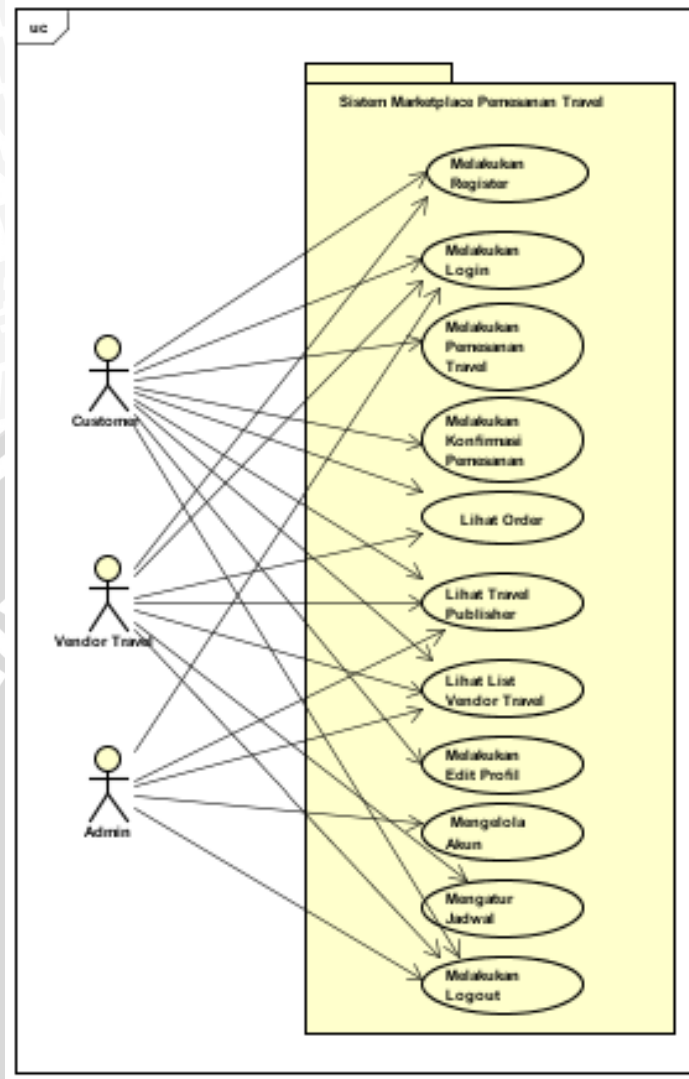
Hasil dari gambaran umum perangkat lunak selain menghasilkan kebutuhan fungsional juga menghasilkan kebutuhan non-fungsional. Dari hasil gambaran umum perangkat lunak diperoleh kriteria aplikasi yang harus dimiliki. Tabel 4.3 akan menjelaskan Kebutuhan non-fungsional yang didapatkan dari hasil gambaran perangkat lunak.

Tabel 4.3 Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional	
<i>Efficiency</i>	Aplikasi harus memberikan kemudahan pada <i>customer</i> untuk mengakses informasi serta pemesanan <i>travel</i> . Untuk vendor <i>travel</i> menjadi lebih mudah mengelola jadwal

4.1.5 Use Case Diagram

Sistem *marketplace* pemesanan *travel* berbasis *web* memiliki fitur sesuai dengan kebutuhan fungsional pada Tabel 4.2 dan digambarkan menggunakan *use case* pada Gambar 4.5 sebagai berikut :



Gambar 4.5 Use Case diagram

Penjelasan dari *use case* diagram pada gambar 4.5 masing-masing akan dijelaskan sebagai berikut:

A. Skenario Use Case Melakukan Register

Menjelaskan mengenai *use case* melakukan register. Di mana user dapat melakukan register apabila belum terdaftar pada sistem, *user* terdiri dari dua aktor yaitu *customer* dan *vendor travel*. Tabel 4.4 akan menjelaskan spesifikasi *use case* register.

Tabel 4.4 Spesifikasi Use Case Melakukan Register

Skenario Kasus Pada Sistem	
Nomor Use Case	UC-01
Nama	Melakukan Register
Tujuan	Agar vendor <i>travel</i> dan customer dapat melakukan register

Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana vendor travel dan <i>customer</i> dapat mendaftar pada sistem.
Aktor	<i>Customer dan Vendor Travel</i>
Skenario Utama	
Kondisi Awal	<i>Customer dan vendor travel membuka web browser.</i>
Aksi Aktor	Reaksi Sistem
1. Membuka <i>web</i> aplikasi 3. Tekan tombol login 5. Tekan tombol <i>register</i> 7. Memilih <i>register</i> sebagai <i>customer</i> atau <i>vendor travel</i> 9. Mengisi <i>form register</i> dan tekan tombol <i>register</i> untuk selesai	2. Menampilkan halaman <i>home</i> 4. Menampilkan <i>form login</i> 6. Menampilkan <i>menu register</i> 8. Menampilkan <i>form register</i>
Kondisi Akhir	<i>Customer dan vendor travel berhasil melakukan register</i>

B. Skenario *Use Case* Melakukan Login

Menjelaskan mengenai *use case* melakukan *login*. Di mana user melakukan *login* untuk masuk ke dalam sistem. Sistem, *user* terdiri dari tiga aktor yaitu *customer*, *vendor travel*, dan admin. Tabel 4.5 akan menjelaskan spesifikasi *use case* login.

Tabel 4.5 *Use Case* Melakukan Login

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-02
Nama	Melakukan <i>Login</i>
Tujuan	Agar Vendor travel, <i>customer</i> , dan admin dapat melakukan <i>login</i>
Deskripsi	<i>Diagram</i> ini akan menjelaskan bagaimana vendor travel, <i>customer</i> , dan admin memasukkan <i>id</i> dan kata sandi untuk <i>login</i> kedalam sistem.
Aktor	<i>Customer, Vendor Travel, Admin</i>
Skenario Utama	
Kondisi Awal	Membuka <i>web browser</i>
Aksi Aktor	Reaksi Sistem
1. Membuka <i>web</i> aplikasi 3. Tekan tombol <i>login</i> 5. Mengisi <i>username</i> dan <i>password</i>	2. Menampilkan halaman <i>home</i> 4. Menampilkan <i>form login</i> 6. Memeriksa <i>username</i> dan <i>password</i> 7. Menampilkan halaman <i>home</i> dengan <i>session id user</i>

Kondisi Akhir	<i>Customer</i> berhasil <i>login</i> dengan <i>session id</i> masing-masing.
---------------	---

C. Skenario *Use Case* Melakukan Pemesanan *Travel*

Menjelaskan mengenai *use case* melakukan pemesanan *travel*. Di mana *user* sudah melakukan *login* untuk masuk ke dalam sistem dengan *session customer*. *Customer* dapat mengisi data pada kolom pemesanan yang tersedia untuk mencari *travel* yang tersedia. Tabel 4.6 akan menjelaskan spesifikasi *use case* pemesanan *travel*.

Tabel 4.6 *Use Case* Melakukan Pemesanan *Travel*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-03
Nama	Melakukan Pemesanan <i>Travel</i>
Tujuan	<i>Customer</i> dapat melakukan pemesanan <i>travel</i> yang diinginkan.
Deskripsi	<i>Use Case</i> ini menjelaskan <i>customer</i> memilih tujuan,tanggal,serta <i>travel</i> yang diinginkan.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi Awal	<i>User</i> telah berhasil <i>login</i> dengan <i>session customer</i> dan masuk pada halaman <i>home</i>
Aksi Aktor	Reaksi Sistem
1. Membuka web aplikasi 3. Mengisi detail informasi dan menekan tombol <i>search</i> 5 Memilih dan tekan <i>order</i>	2. Menampilkan halaman <i>home</i> dengan id <i>customer</i> 4. Menampilkan vendor <i>travel</i>
Kondisi Akhir	<i>Customer</i> dapat melakukan order dari vendor <i>travel</i> yang diinginkan.

D. Skenario *Use Case* Melakukan Konfirmasi Pemesanan *Travel*

Menjelaskan mengenai *use case* melakukan konfirmasi pemesanan *travel*. Di mana *user* sudah melakukan *login* dan masuk ke dalam sistem dengan *session customer*. *Customer* telah memilih *travel* yang diinginkan dan mendapatkan konfirmasi pemesanan. Tabel 4.7 akan menjelaskan spesifikasi *use case* konfirmasi pemesanan *travel*.

Tabel 4.7 *Use Case* Melakukan Konfirmasi Pemesanan *Travel*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-04
Nama	Melakukan Konfirmasi Pemesanan <i>Travel</i>
Tujuan	Customer melakukan konfirmasi pemesanan <i>travel</i>
Deskripsi	<i>Use Case</i> ini menjelaskan <i>customer</i> melakukan konfirmasi pemesanan <i>travel</i> .
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi Awal	<i>Customer</i> telah berhasil <i>login</i> ke dalam sistem dan memesan <i>travel</i> yang di inginkan.
Aksi Aktor	Reaksi Sistem
1. Memilih <i>travel</i> dan tekan <i>order</i> 3. Tekan tombol konfirmasi untuk pesan	2. Menampilkan halaman konfirmasi
Kondisi Akhir	<i>Customer</i> berhasil melakukan konfirmasi <i>order</i>

E. Skenario *Use Case* Lihat *Order*

Menjelaskan mengenai *use case* konfirmasi lihat *order*. Dimana *customer* dan *Vendor Travel* sudah melakukan login dan masuk ke dalam sistem. *Customer* dan *Vendor Travel* dengan *session customer* telah melakukan pemesanan *travel* dan mendapatkan informasi order yang telah dipesan. *Customer* dan *Vendor Travel* dengan *session* masing-masing mendapatkan informasi *order* yang masuk dari pemesanan *customer*. Tabel 4.8 akan menjelaskan spesifikasi *use case* lihat *order*.

Tabel 4.8 *Use Case* Lihat *Order*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-05
Nama	Lihat <i>Order</i>
Tujuan	Memberi informasi kepada <i>customer</i> dan <i>vendor travel</i> tentang <i>travel</i> yang telah di pesan.
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana sistem memberikan informasi tentang <i>travel</i> yang telah di pesan kepada <i>customer</i> dan <i>vendor travel</i> .
Aktor	<i>Customer</i> dan <i>Vendor Travel</i>
Skenario Utama	
Kondisi Awal	<i>Customer</i> telah berhasil <i>login</i> ke dalam sistem dan masuk kedalam halaman <i>home</i> .
Aksi Aktor	Reaksi Sistem

1. Membuka web aplikasi 3. Menekan tombol lihat order	2. Menampilkan halaman home dengan session id user masing-masing 4. Menampilkan halaman order sesuai dari order customer
Kondisi Akhir	<i>Customer dan vendor travel</i> dapat melihat informasi travel yang telah di pesan.

F. Skenario Use Case Lihat Travel *Publisher*

Menjelaskan mengenai *use case* lihat travel *publisher*. Di mana *customer*, *vendor travel*, dan admin sudah melakukan *login* dan masuk ke dalam sistem. *Customer*, *vendor travel*, dan admin dapat melihat *publisher* yang bekerja sama dengan sistem. Tabel 4.9 akan menjelaskan spesifikasi *use case travel publisher*.

Tabel 4.9 *Use Case* Lihat Travel *Publisher*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-06
Nama	Lihat Travel <i>Publisher</i>
Tujuan	Memberi informasi kepada <i>customer</i> , <i>vendor travel</i> , dan admin tentang travel <i>publisher</i> yang bekerja sama dengan sistem pemesanan travel.
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana sistem memberikan dan menampilkan travel <i>publisher</i> yang bekerja sama kepada <i>Customer</i> , <i>Vendor Travel</i> , dan Admin.
Aktor	<i>Customer</i> , <i>Vendor Travel</i> , dan Admin
Skenario Utama	
Kondisi Awal	User telah berhasil <i>login</i> ke dalam sistem dan masuk kedalam halaman <i>home</i> .
Aksi Aktor	Reaksi Sistem
1. Membuka web aplikasi 2. Menekan tombol <i>publisher</i>	3. Menampilkan halaman <i>home</i> 4. Menampilkan travel <i>publisher</i>
Kondisi Akhir	<i>Customer</i> , <i>Vendor Travel</i> , dan Admin dapat melihat travel <i>publisher</i> yang bekerja sama.

G. Skenario *Use Case* Lihat List Vendor Travel

Menjelaskan mengenai *use case* lihat list vendor travel. Dimana *Customer*, *Vendor Travel*, dan Admin sudah melakukan login dan masuk ke dalam sistem. *Customer*, *Vendor Travel*, dan Admin dapat melihat informasi list vendor travel yang terdaftar pada sistem. Tabel 4.10 akan menjelaskan spesifikasi *use case* list vendor travel.

Tabel 4.10 Use Case Lihat List Vendor Travel

Skenario Kasus Pada Sistem	
Nomor Use Case	UC-07
Nama	Lihat List Vendor Travel
Tujuan	Memberikan informasi <i>vendor travel</i> yang terdaftar pada sistem kepada <i>Customer</i> , <i>Vendor Travel</i> , dan Admin
Deskripsi	Use Case ini menjelaskan bagaimana sistem menampilkan informasi tentang <i>vendor travel</i> yang terdaftar pada <i>Customer</i> , <i>Vendor Travel</i> , dan Admin
Aktor	<i>Customer</i> , <i>Vendor Travel</i> , dan Admin
Skenario Utama	
Kondisi Awal	<i>Customer</i> , <i>Vendor Travel</i> , dan Admin telah melakukan <i>login</i> dan berada pada halaman <i>home</i> sistem
Aksi Aktor	Reaksi Sistem
1. User menekan tombol <i>list travel</i> yang berada pada c	2. Sistem meneruskan ke <i>database vendor travel</i> dan selanjutnya menampilkan informasi list <i>vendor travel</i> yang terdaftar .
Kondisi Akhir	<i>Customer</i> , <i>Vendor Travel</i> , dan Admin dapat melihat informasi list <i>vendor travel</i> yang terdaftar pada sistem.

H. Skenario Use Case Melakukan Edit Profil

Menjelaskan mengenai *use case* melakukan edit profil. Dimana *customer* sudah melakukan login dan masuk ke dalam sistem. *Customer* dapat melakukan edit profil dalam sistem. Tabel 4.11 akan menjelaskan spesifikasi *use case* edit profil.

Tabel 4.11 Use Case Melakukan Edit Profil

Skenario Kasus Pada Sistem	
Nomor Use Case	UC-08
Nama	Melakukan Edit Profil
Tujuan	<i>Customer</i> dapat mengubah profil.
Deskripsi	Use Case ini menjelaskan bagaimana <i>customer</i> mengedit profilnya.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi Awal	<i>Customer</i> masuk ke dalam fitur profil
Aksi Aktor	Reaksi Sistem
1. Membuka <i>web</i> aplikasi 3. Menekan tombol <i>edit</i> profil 5. Melakukan <i>edit</i> profil	2. Menampilkan halaman <i>home</i> dengan <i>id customer</i> 4. Menampilkan halaman <i>edit</i> profil

6. Tekan tombol <i>update</i> untuk selesai	
Kondisi Akhir	Profil <i>customer</i> berubah sesuai dengan data yang diinputkan.

I. Skenario *Use Case* Mengelola Akun

Menjelaskan mengenai *use case* mengelola akun. Dimana *user* dengan *session* admin sudah melakukan *login* dan masuk ke dalam sistem. Admin dapat mengelola akun dari *customer* dan *vendor travel*. Tabel 4.12 akan menjelaskan spesifikasi *use case* manage akun.

Tabel 4.12 *Use Case* Mengelola Akun

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-09
Nama	Mengelola Akun
Tujuan	<i>Admin</i> dapat mengelola akun yang terdaftar.
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana <i>admin</i> mengelola akun <i>customer</i> dan <i>vendor travel</i>
Aktor	Admin
Skenario Utama	
Kondisi Awal	Admin masuk ke dalam fitur admin
Aksi Aktor	Reaksi Sistem
1. Membuka <i>web</i> aplikasi 3. Memilih menu halaman admin 5. Menekan tombol <i>delete</i> untuk akun yang akan di hapus 7. Menekan tombol hapus untuk selesai	2. Menampilkan halaman <i>home</i> dengan id admin 4. Masuk pada halaman admin 6. Menampilkan konfirmasi hapus
Kondisi Akhir	Akun <i>user</i> telah terhapus dari <i>database</i> sistem

J. Skenario *Use Case* Mengatur Jadwal

Menjelaskan mengenai *use case* mengatur jadwal. Dimana *user* dengan *session vendor travel* sudah melakukan *login* dan masuk ke dalam sistem. *Vendor travel* dapat mengelola jadwal baik menambah, mengedit, dan menghapus. Tabel 4.13 akan menjelaskan spesifikasi *use case* mengatur jadwal.

Tabel 4.13 *Use Case* Mengatur Jadwal

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-10
Nama	Mengatur Jadwal
Tujuan	<i>Vendor Travel</i> dapat mengatur jadwal <i>travel</i> .

Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana <i>user</i> mengatur jadwal berdasarkan ketersediaan armada <i>travel</i>
Aktor	<i>Vendor Travel</i>
Skenario Utama	
Kondisi Awal	<i>User</i> masuk ke dalam fitur jadwal
Aksi Aktor	Reaksi Sistem
1. Membuka web aplikasi 3. Menekan tombol <i>add schedule</i> 5. Menekan tombol <i>add, edit, atau delete</i> 7. Memasukan <i>detail</i> jadwal <i>travel</i> 8. Menekan tombol <i>add, edit, atau delete</i> setelah selesai	2. Menampilkan halaman <i>home</i> dengan <i>id vendor travel</i> 4. Masuk pada halaman jadwal 6. Masuk pada <i>form</i> jadwal
Kondisi Akhir	Jadwal <i>vendor travel</i> di tambahkan, di <i>edit</i> , atau dihapuskan

K. Skenario *Use Case* Melakukan *Logout*

Menjelaskan mengenai *use case* melakukan *logout*. Dimana *user* dengan *session* masing-masing yang sudah melakukan *login* dan masuk ke dalam sistem dapat keluar dari *session* masing-masing. Tabel 4.13 akan menjelaskan spesifikasi *use case* melakukan *logout*.

Tabel 4.14 *Use Case* Melakukan *Logout*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	UC-11
Nama	Melakukan <i>Logout</i>
Tujuan	<i>Customer, Vendor Travel</i> , dan Admin <i>logout</i> dari sistem
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana <i>Customer, Vendor Travel</i> , dan Admin keluar dari sistem
Aktor	<i>Customer, Vendor Travel</i> , dan Admin
Skenario Utama	
Kondisi Awal	<i>Customer, Vendor Travel</i> , dan Admin berada di halaman web
Aksi Aktor	Reaksi Sistem
1. Menekan tombol <i>id user</i> 3. Tekan tombol <i>logout</i> 5. Menekan tombol <i>logout</i> untuk keluar	2. Menampilkan halaman <i>id logout</i> 4. Menampilkan konfirmasi <i>logout</i>

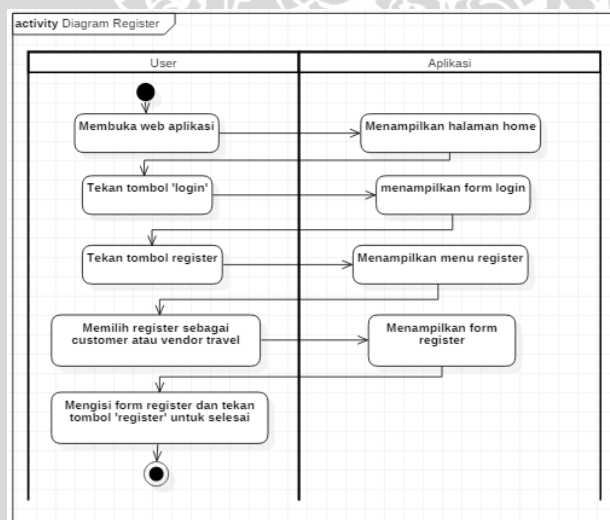
Kondisi Akhir	<i>Customer, Vendor Travel</i> , dan Admin keluar <i>dari session</i> masing-masing dan kembali ke halaman <i>home</i> .
---------------	--

4.1.6 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alir aktivitas dari setiap *use case* dalam sistem, bagaimana setiap alir berawal, keputusan yang mungkin terjadi, dan bagaimana akhirnya dari sistem *marketplace* pemesanan travel.

4.1.6.1 Proses Melakukan Register

Gambar 4.6 menunjukkan proses register *customer* dan vendor travel. Dimana *customer* dan vendor travel pertama kali membuka web aplikasi dan belum memiliki akun, sistem akan menampilkan halaman home dan *customer* dan vendor travel menekan tombol login. Sistem merespon dengan menampilkan form login lalu *user* memilih menu register. Sistem menampilkan menu register dengan dua pilihan menu. Pertama menu register untuk *customer* dan kedua menu register untuk vendor travel. *Customer* dan vendor travel dapat memilih salah satu dan sistem akan merespon dengan menampilkan form register. Setelah melengkapi data, *customer* dan vendor travel dapat menekan tombol register untuk selesai.

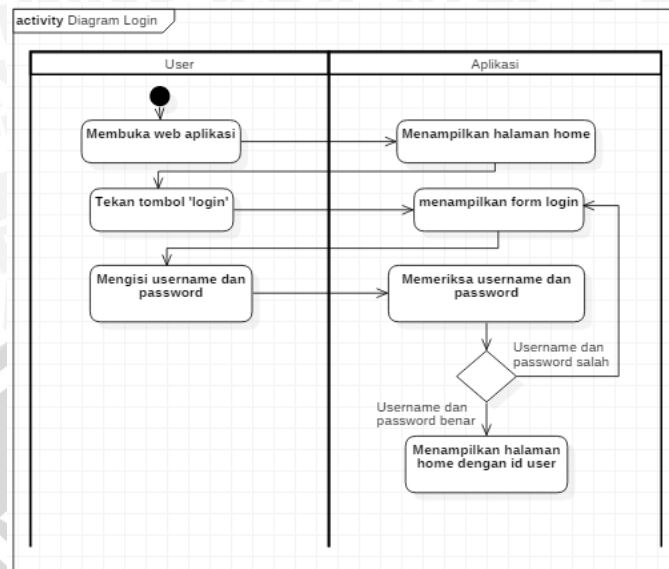


Gambar 4. 6 Proses Melakukan Register

4.1.6.2 Proses Melakukan Login

Gambar 4.7 menunjukkan proses login *customer*, vendor travel, dan admin. Dimana ketika *customer*, vendor travel, dan admin pertama membuka web aplikasi, sistem akan menampilkan halaman home dan *user* menekan tombol login. Sistem merespon dengan menampilkan form login lalu *customer*, vendor travel, dan admin memasukkan username dan password. Sistem memeriksa username dan password pada database untuk menentukan *customer*, vendor travel, dan admin akan masuk

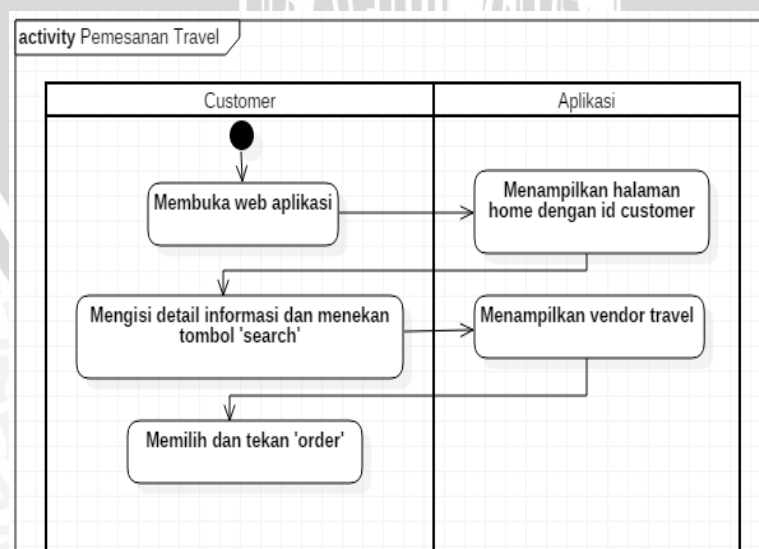
dengan id *customer*, vendor travel, atau sebagai admin. Berikutnya sistem akan menampilkan halaman home dengan id *customer*, vendor travel, dan admin.



Gambar 4.7 Proses Melakukan Login

4.1.6.3 Proses Melakukan Pemesanan Travel

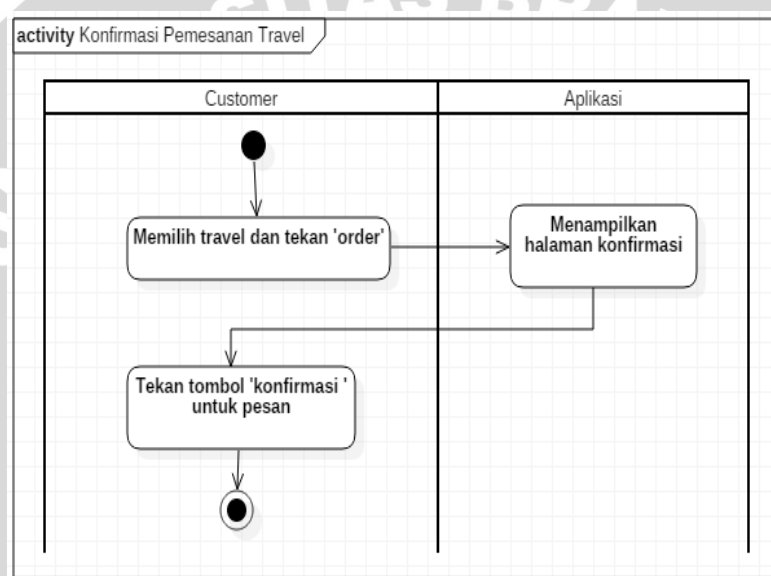
Gambar 4.8 menunjukkan *customer* melakukan pemesanan travel. Dimana akan dimulai dengan *customer* membuka sistem dan sistem menampilkan halaman home dengan id *customer*. Selanjutnya *customer* mengisi informasi yang dibutuhkan untuk memesan. Melalui tombol search, sistem akan menampilkan vendor travel yang sesuai dengan informasi yang telah dimasukan oleh *customer*. *Customer* memilih vendor travel yang sesuai di akhiri dengan menekan tombol order.



Gambar 4.8 Proses Melakukan Pemesanan *Travel*

4.1.6.4 Proses Melakukan Konfirmasi Pemesanan

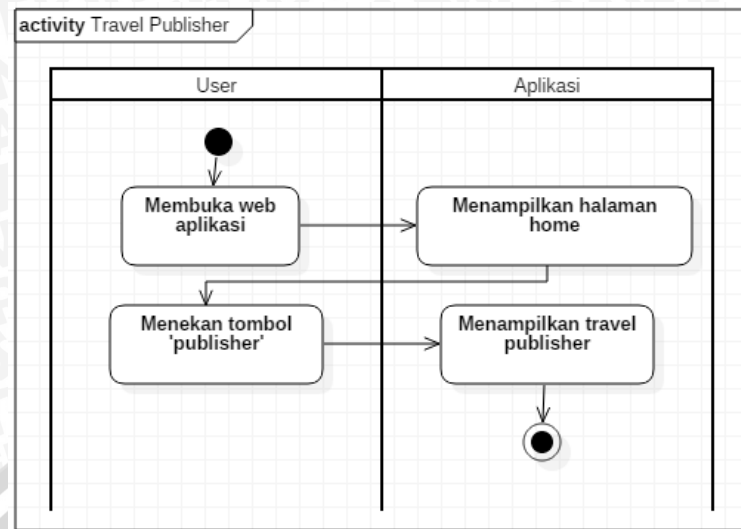
Gambar 4.9 menunjukkan *customer* melakukan konfirmasi pemesanan *travel*. Dimana akan dimulai dengan *customer* membuka sistem dan sistem menampilkan halaman home dengan id *customer*. Selanjutnya *customer* mengisi informasi yang dibutuhkan untuk memesan. Melalui tombol search, sistem akan menampilkan vendor *travel* yang sesuai dengan informasi yang telah dimasukan oleh *customer*. *Customer* memilih vendor *travel* yang sesuai di akhiri dengan menekan tombol order.



Gambar 4.9 Proses Melakukan Konfirmasi Pemesanan *Travel*

4.1.6.5 Proses Lihat *Travel Publisher*

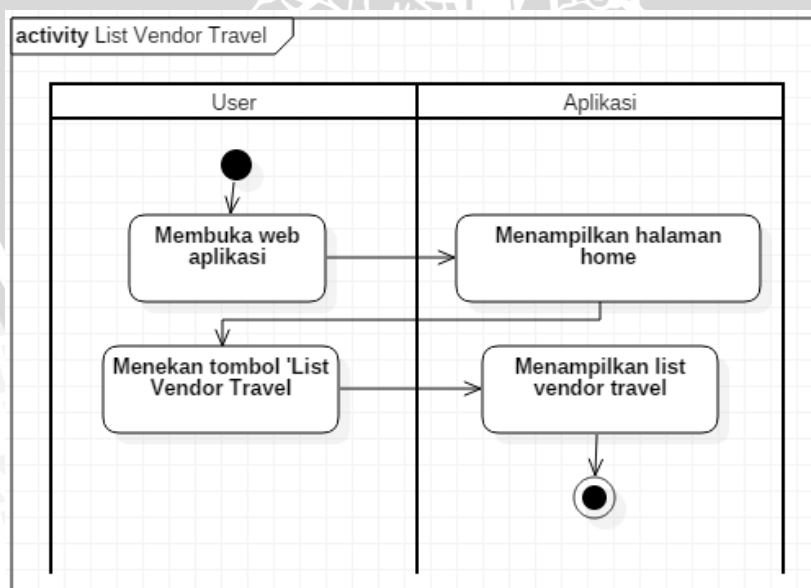
Gambar 4.10 menunjukkan *customer*, vendor *travel*, dan admin dapat melihat *travel publisher*. *Travel publisher* adalah menu informasi daftar *publisher* yang bekerja sama dengan sistem. *customer*, vendor *travel*, dan admin dapat melihat informasi dengan membuka sistem dan sistem akan menampilkan halaman home. *customer*, vendor *travel*, dan admin menekan tombol '*publisher*', selanjutnya sistem akan merespon dengan menampilkan daftar *publisher* yang berkerja sama.



Gambar 4. 10 Proses Lihat *Travel Publisher*

4.1.6.6 Proses Lihat *Vendor Travel*

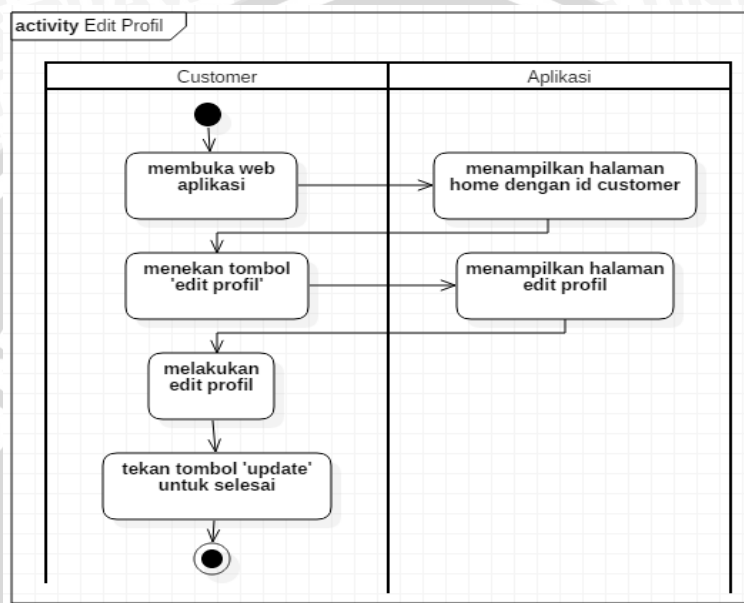
Gambar 4.11 menunjukkan *customer*, *vendor travel*, dan *admin* dapat melihat list *vendor travel*. List *vendor travel* adalah menu informasi daftar *vendor travel* yang bekerja sama dengan sistem. *customer*, *vendor travel*, dan *admin* dapat melihat informasi dengan membuka sistem dan sistem akan menampilkan halaman home. *customer*, *vendor travel*, dan *admin* menekan tombol '*list vendor travel*', selanjutnya sistem akan merespon dengan menampilkan daftar *vendor travel* yang berkerja sama.



Gambar 4.11 Proses Lihat *List Vendor Travel*

4.1.6.7 Proses Customer Melakukan Edit Profil

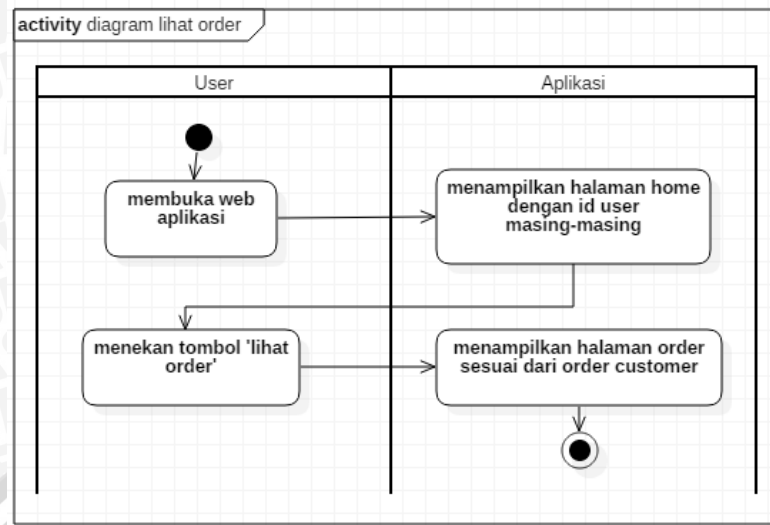
Gambar 4.11 menunjukkan *customer* dapat melakukan edit profil. Pertama *customer* membuka sistem. Sistem menampilkan halaman home dengan id *customer*. *customer* menekan tombol edit profil pada sistem. Dan sistem merespon dengan menampilkan halaman edit profil. *customer* melakukan edit profil dan menekan tombol 'update' untuk selesai.



Gambar 4.12 Proses *Customer* Melakukan Edit Profil

4.1.6.8 Proses Lihat Order

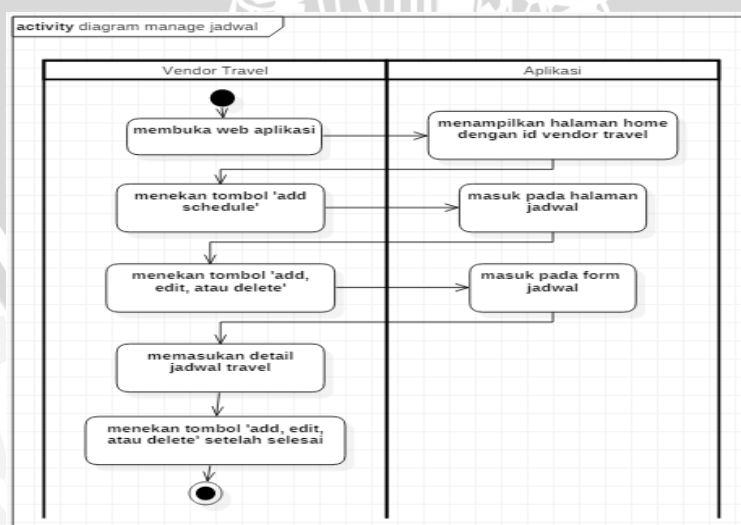
Gambar 4.12 menunjukkan bagaimana *customer* dan vendor travel dapat melihat order yang telah di pesan. *customer* dan vendor travel yang dapat melihat halaman order adalah *customer* dan vendor travel dengan membuka sistem. Sistem menampilkan halaman home dengan id *customer* dan vendor travel . Setelah itu *customer* dan vendor travel menekan tombol 'lihat order'. Dan sistem merespon menampilkan informasi order sesuai dengan order *customer*.



Gambar 4.13 Proses *Customer* Lihat Order

4.1.6.9 Proses *Vendor Travel* Mengatur Jadwal

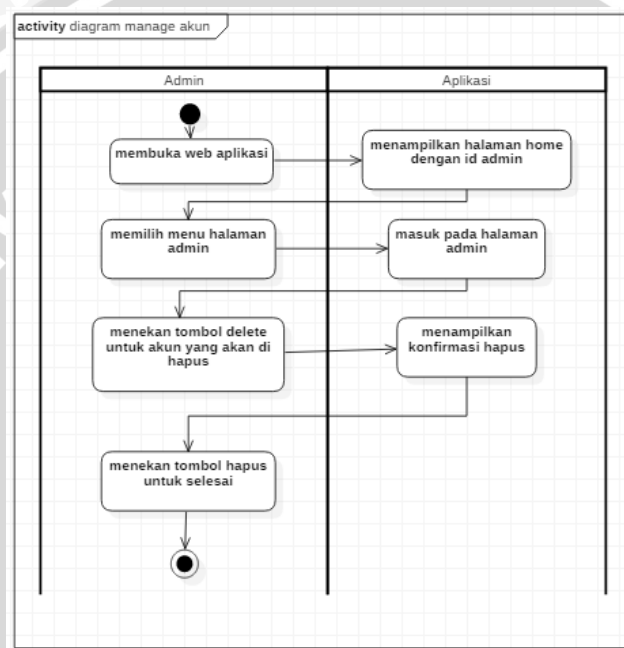
Gambar 4.13 menunjukkan proses vendor travel mengatur jadwal. Untuk mengatur jadwal, vendor travel dapat membuka sistem. Sistem menampilkan halaman home dengan id vendor travel. Selanjutnya vendor travel menekan tombol 'add schedule'. Sistem merespon dengan masuk pada halaman jadwal. Pada halaman tersebut tersedia tombol 'add, edit, dan delete' untuk mengatur jadwal. Setelah vendor travel memilih sesuai kebutuhan, sistem akan masuk pada form jadwal. Vendor travel bisa memasukkan, mengedit atau menghapus detail jadwal. Terakhir sistem menampilkan jadwal sesuai dengan inputan yang di masukan vendor travel.



Gambar 4.14 Proses *Travel* Mengatur Jadwal

4.1.6.10 Proses Admin Mengelola Akun

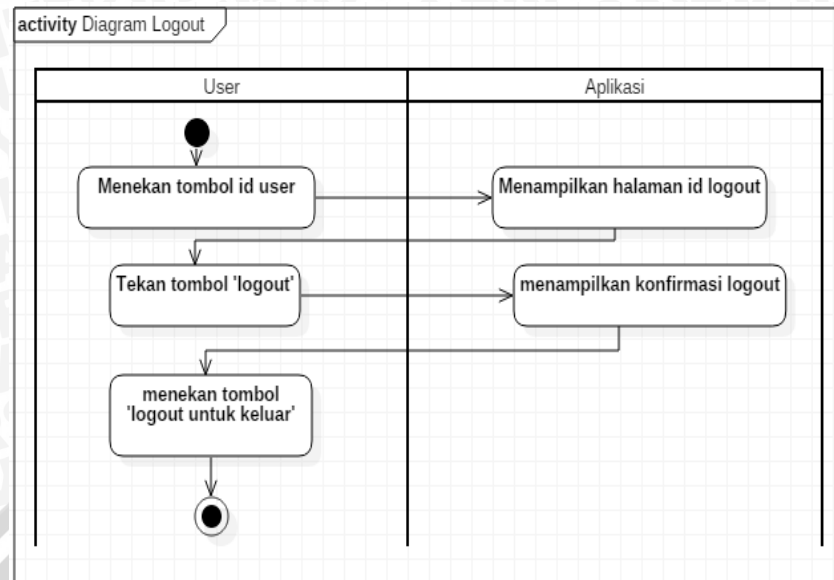
Gambar 4.14 menunjukkan admin menghapus pelanggan. Dimulai dengan admin membuka sistem. Sistem menampilkan halaman home dengan id admin. Admin menekan tombol 'admin page'. Sistem merespon dengan menampilkan daftar pelanggan dan admin menekan tombol delete pada sistem. Sistem menampilkan konfirmasi untuk menghapus. Admin melakukan konfirmasi dengan menekan tombol 'delete' untuk setuju. Sistem merespon dengan menghapus informasi pelanggan.



Gambar 4.15 Proses Admin Mengelola Akun

4.1.6.11 Proses Melakukan Logout

Gambar 4.15 menunjukkan *customer*, vendor travel, dan admin melakukan logout dari sistem. Dimulai dengan *customer*, vendor travel, dan admin menekan tombol id masing-masing. Sistem menampilkan halaman logout *customer*, vendor travel, dan admin. *Customer*, vendor travel, dan admin menekan tombol logout. Selanjutnya sistem merespon dengan menampilkan konfirmasi untuk logout. *customer*, vendor travel, dan admin menekan tombol logout untuk keluar dari session masing-masing dan sistem dapat digunakan kembali oleh user lain.



Gambar 4.16 Proses Melakukan Logout

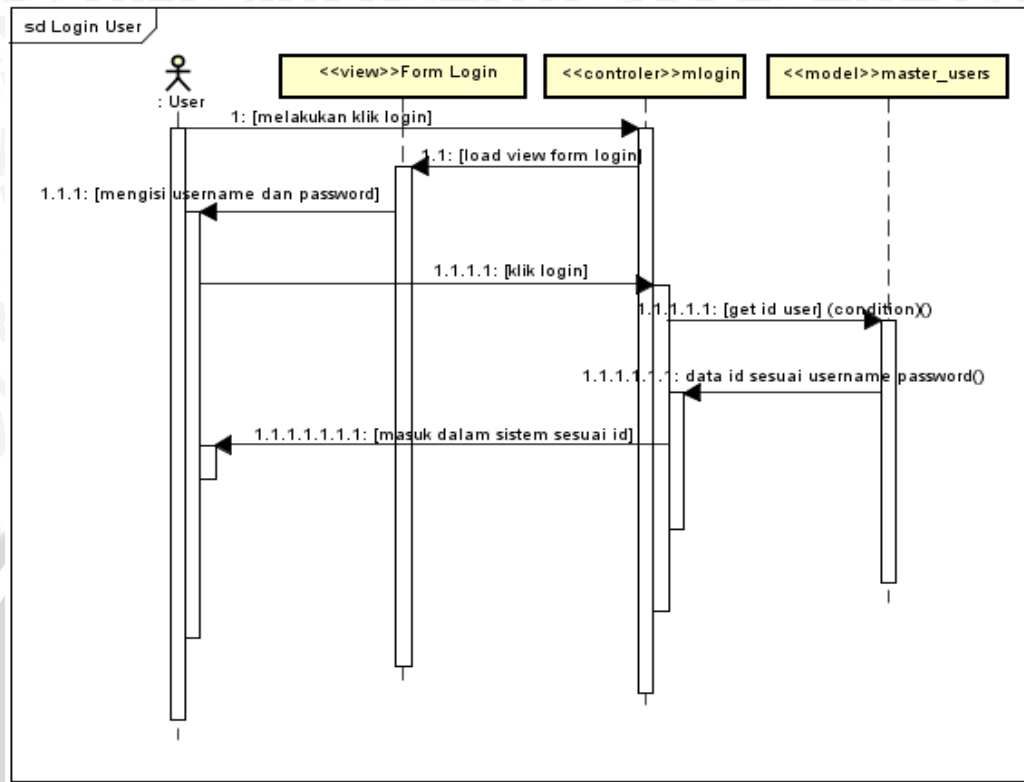
4.2 Perancangan Perangkat Lunak

Dalam metode Com HDM (*Comprehensive Hypermedia Design Method*) untuk perancangan perangkat lunak *marketplace* pemesanan travel berbasis web dilakukan beberapa tahap yaitu perancangan *conceptual design*, *navigation models*, dan *user interface models*.

4.2.1 Sequence Diagram

Sequence diagram bermanfaat untuk menentukan metode-metode (fungsi) yang terlibat dalam suatu kelas. Para ahli menggunakan pernyataan-pernyataan dari spesifikasi kebutuhan atau *activity diagram* yang dikembangkan dari suatu *use case* untuk mendapatkan pernyataan informal mengenai aplikasi yang dikembangkan. Jadi dapat disimpulkan bahwa pernyataan kebutuhan dan *activity diagram* yang merupakan deskripsi terperinci dari suatu *use case diagram*, kalimat-kalimat yang dijumpai merupakan kandidat kelas dan dapat diterjemahkan dalam bentuk *sequence diagram*.

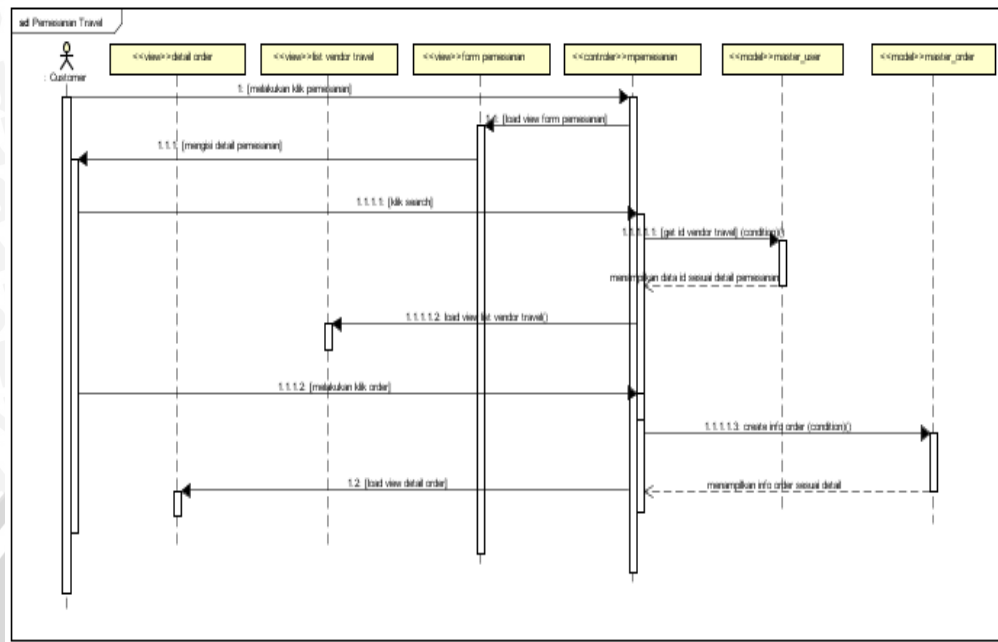
A. Login



Gambar 4.17 Squence Login User

Pada Gambar 4.19 menjelaskan *user* melakukan *login*. User terdiri dari tiga actor yaitu *customer*, *vendor travel*, dan *admin*. Setelah pengisian data *user* menekan tombol 'Login', method akan melakukan proses `username,password()` dalam aplikasi. Selanjutnya method `username,password ()` akan mengakses method `select data` yang berisi query `select data` pada database dan akan mengembalikan data tersebut untuk dilakukan `verify password()` pada aplikasi. Method `verify password()` mengakses method `select data` yang berisi query `select data` pada database dan akan mengembalikan data tersebut untuk dilakukan `session int()`. Method `session int()` mengakses method `pesan sukses()` dan user akan masuk sesuai session masing-masing.

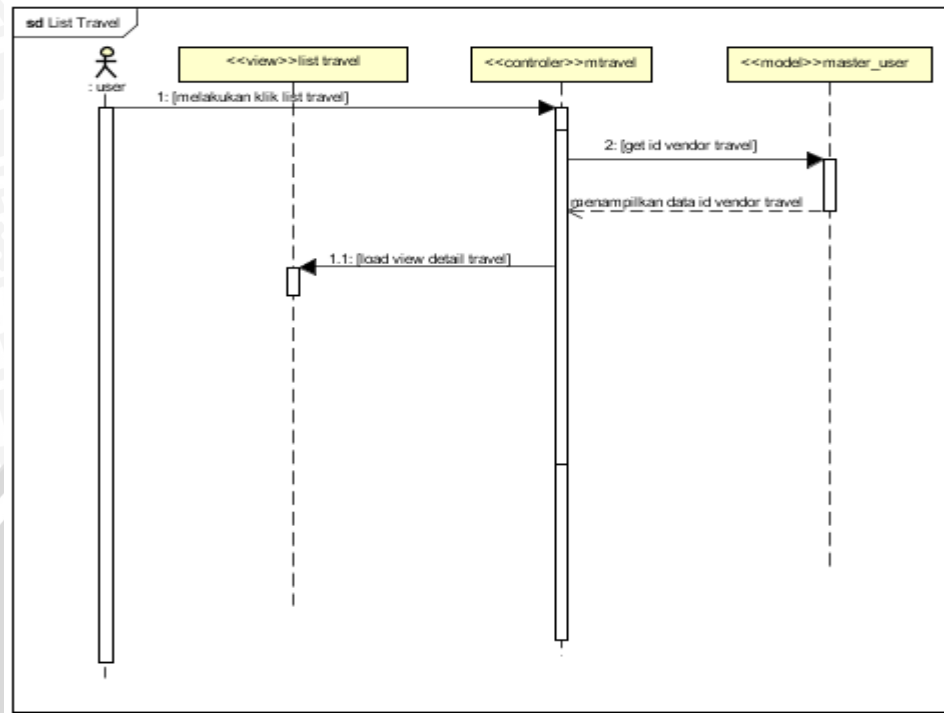
B. Pemesanan Travel



Gambar 4.18 Sequence Pemesanan Travel

Pada Gambar 4.20 menjelaskan customer melakukan pemesanan travel. Setelah pengisian data pada GUI dan menekan tombol 'search'. Method search akan mengirim data ke aplikasi dan memanggil method select data yang berisi query select data. Kemudian kembali dengan memanggil view hasil pencarian. Customer kembali memilih dan menekan 'order'. Method order akan mengirim data ke aplikasi dan memanggil method select data. Kemudian kembali dengan memanggil view data detail. Customer melakukan konfirmasi dan menekan tombol pesan. Method 'pesan' akan memanggil method insert data yang berisikan query insert pada database lalu dikembalikan dengan method respon memanggil view berupa pesan 'sukses'.

C. List Travel



Gambar 4.19 Sequence List Travel

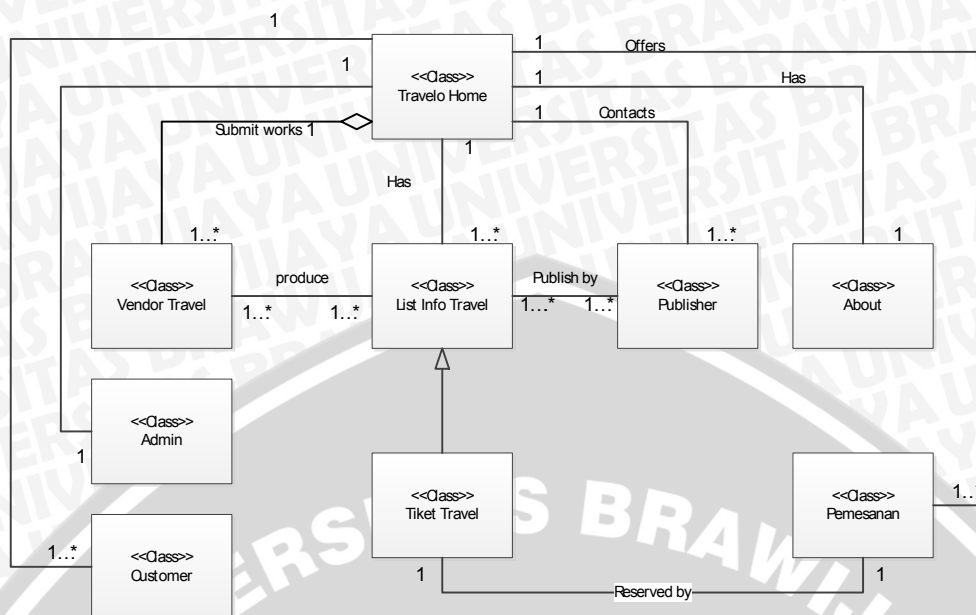
Pada Gambar 4.21 menjelaskan *user* melihat list travel. User terdiri dari tiga actor yaitu *customer*, *vendor travel*, dan *admin* yang dapat memilih menu list. Method list akan mengirim request pada aplikasi untuk memanggil method select data. Method select data berisi query select data pada database yang dikembalikan dengan memanggil view berupa hasil 'list travel'.

4.2.2 Conceptual Models

Conceptual models ini berfungsi untuk mendefinisikan kelas dan proses kompleks dalam objek. Proses desain pada tahap ini di bagi menjadi tiga yaitu Conceptual class model, Complex processes flow model, dan Conceptual process model. Seperti pada gambar 3.3 pada bab 3.

4.2.2.1 Conceptual Class Model

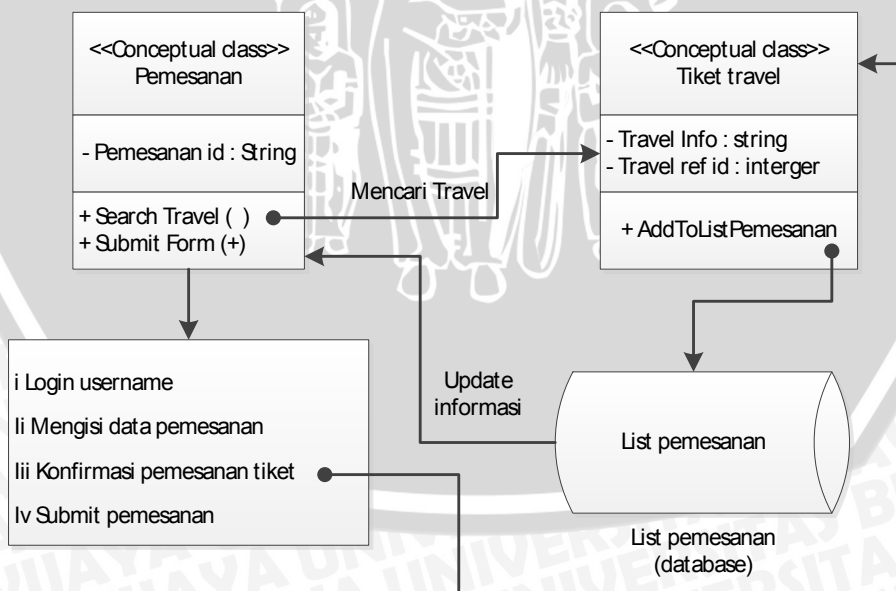
Conceptual class model adalah tahap awal desain, berfungsi untuk mendefinisikan kelas dengan menggunakan *Class diagram* tanpa memperhatikan aliran proses kompleks, navigasi link, dan *user interface* nya seperti pada gambar 4.19.



Gambar 4.20 Conceptual Class Model

4.2.2.2 Complex Processes Flow Model

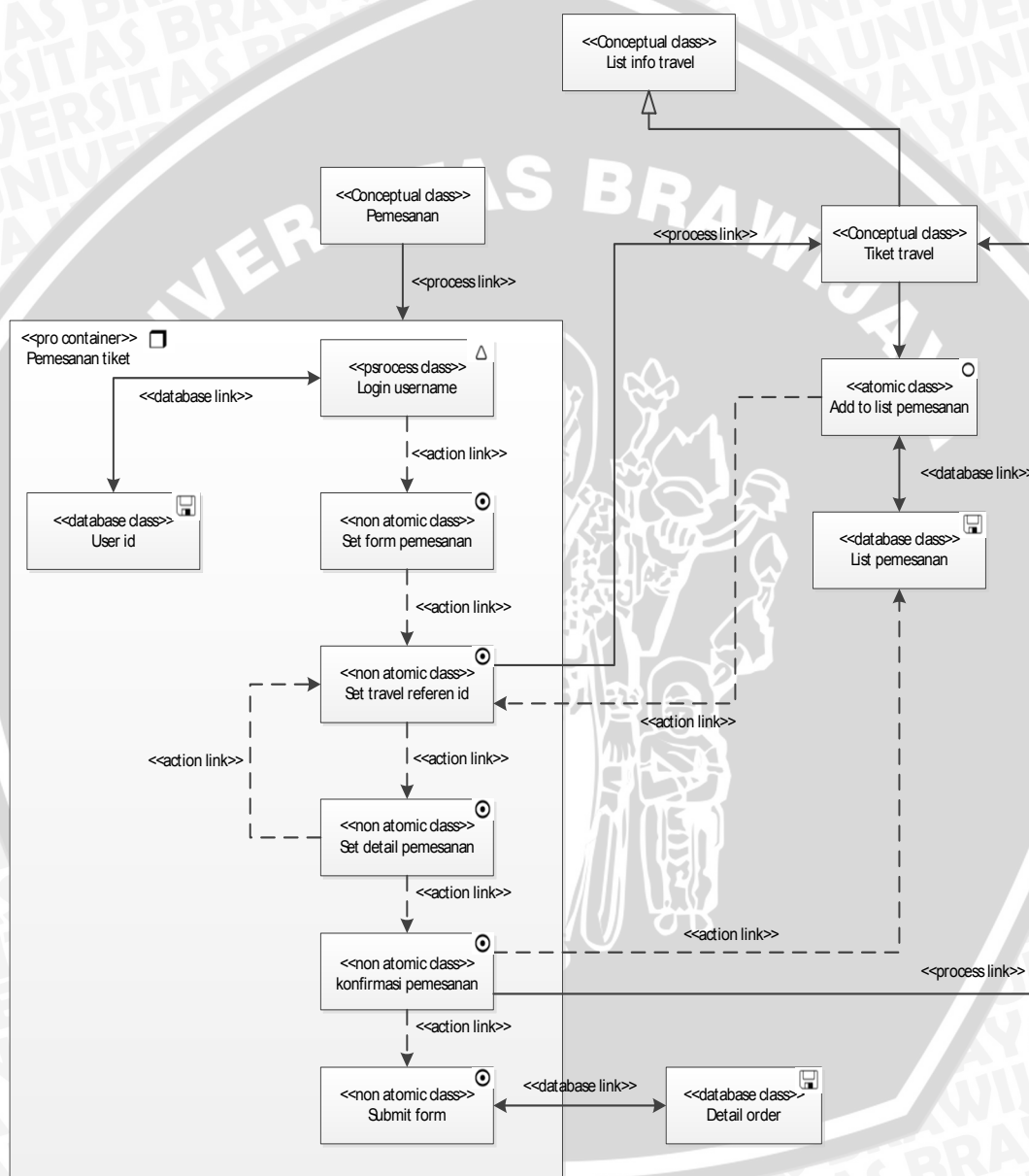
Complex processes flow model adalah tahap desain yang membantu analis atau desainer berfungsi mendefinisikan rincian kelas yang memiliki proses kompleks seperti pada gambar 4.20. Semua simbol yang di gunakan dapat dilihat pada tabel 3.2 yang telah di jelaskan pada bab 3.



Gambar 4.21 Complex Processes Flow

4.2.2.3 Conceptual Process Model

Tahap selanjutnya adalah *Conceptual process model*, dimana desain ini berfungsi untuk menjelaskan kelas dengan proses kompleks yang dapat di sisipkan diantara kelas-kelas pada desain yang saling berhubungan seperti pada gambar 4.21. Semua simbol yang di gunakan dapat dilihat pada tabel 3.2 yang telah di jelaskan pada bab 3.



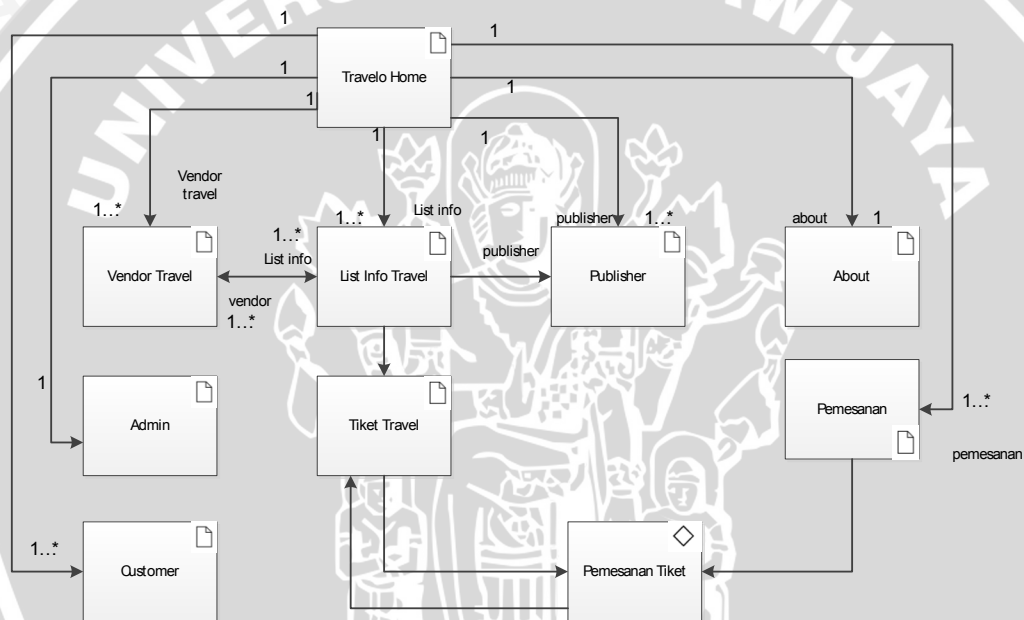
Gambar 4.22 Conceptual Process Model

4.2.3 Navigation Models

Navigation model adalah tahap selanjutnya dimana dalam desain ini akan di gambarkan mekanisme navigasi untuk mengakses informasi pada web. Tahap ini dibagi menjadi dua yaitu *navigation class model* dan *navigation access interaction model*. Seperti pada gambar 3.3 pada bab 3.

4.2.3.1 Navigation Class Model

Dalam *navigation class model* konseptual kelas di sajikan dalam perspektif navigasi. Pada gambar 4.22 menunjukkan bagaimana navigation class model di bangun dari desain konseptual yang menyajikan kelas navigasi, kelas interaksi, dan hyperlink agar user mencapai informasi yang di inginkan. Semua simbol yang di gunakan dapat dilihat pada tabel 3.3 yang telah di jelaskan pada bab 3.

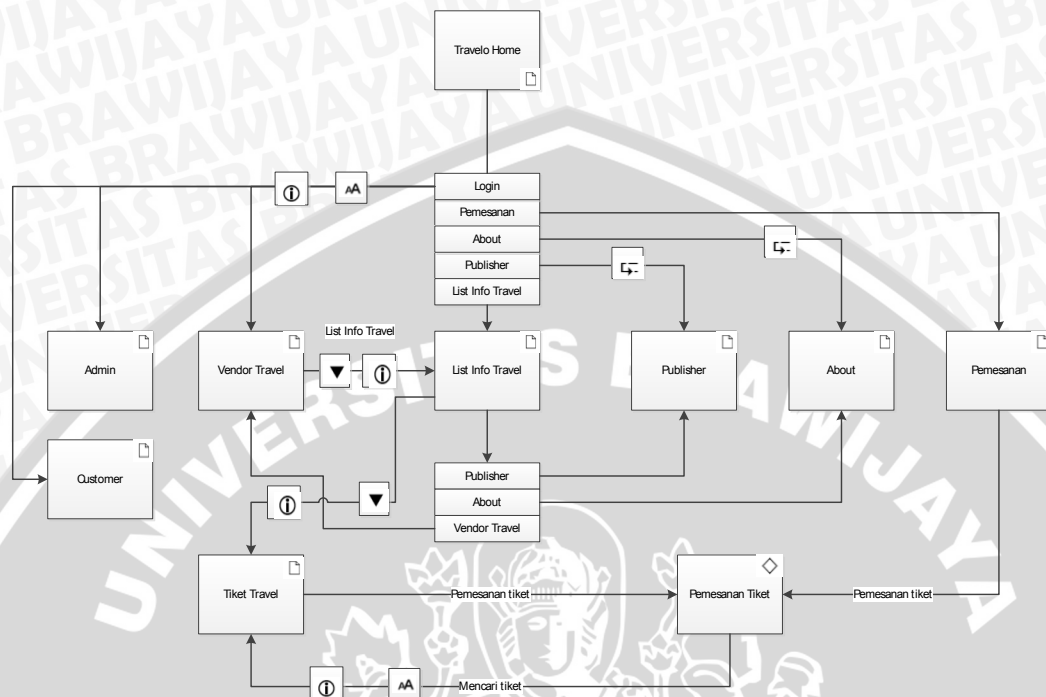


Gambar 4.23 Navigation Class Model

4.2.3.2 Navigation Access Interaction Model

Tahap selanjutnya adalah *navigation access interaction model*. Dimulai dengan menggunakan item menu. Tidak ada batasan berapa banyak item yang harus didefinisikan dalam menu. Ini tergantung pada kebutuhan navigasi dan informasi dari user web. Setelah menu yang digunakan dalam desain, analis harus menentukan struktur akses. Setiap struktur akses akan menyediakan informasi dalam berbagai bentuk presentasi. Penting bagi analis untuk dapat membuat keputusan yang baik pada useran simbol di struktur akses agar cocok dalam model sehingga user web dapat menelusuri informasi dengan sederhana, mudah, dan cepat. Pada gambar 4.23 ditunjukkan navigation access dan interaction model

sistem *marketplace* pemesanan travel. Semua simbol yang di gunakan dapat dilihat pada tabel 3.4 yang telah di jelaskan pada bab 3.

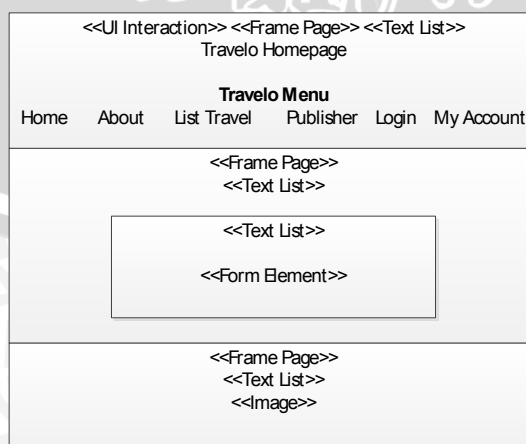


Gambar 4.24 Navigation Access Interaction Model

4.2.4 User Interface Model

User Interface Model ini adalah perancangan terakhir. Dimana di sini akan dibuat sesuai dari perancangan *Navigation Access Interaction Model* sebelum nya. Dan di sini akan di tunjukan masing-masing fungsi dari halaman menurut pada transformasi simbol pada tabel 3.5 yang telah di jelaskan pada bab 3.

4.2.4.1 User Interface Halaman Home

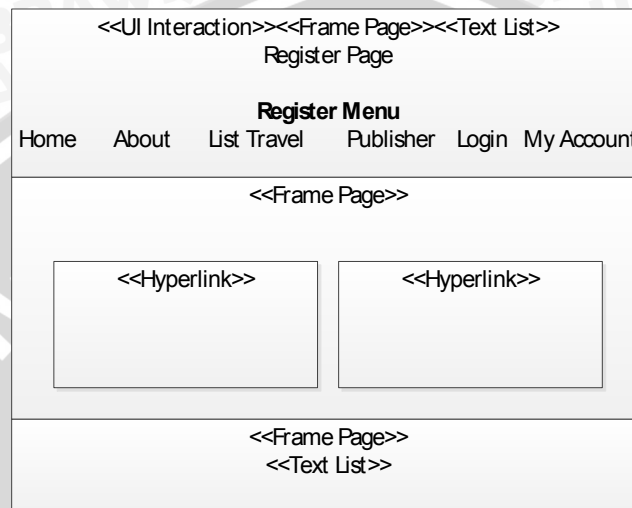


Gambar 4.25 User Interface Halaman Home

Keterangan :

- A. *UI Interaction* pada bagian *header* menunjukkan adanya interaksi pada halaman antara user dengan konten.
- B. Form Element menunjukkan ada nya form pada bagian konten yang dapat di isi oleh user.
- C. Image pada bagian *footer* menunjukkan adanya informasi dengan gambar.

4.2.4.2 User Interface Halaman Register

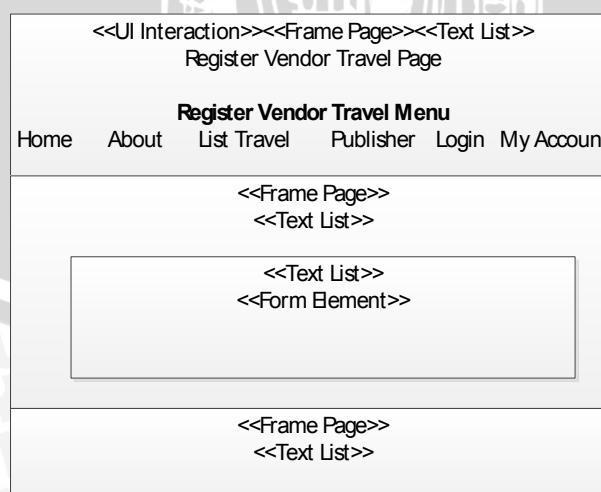


Gambar 4.26 User Interface Halaman Register

Keterangan :

- A. *UI Interaction* pada bagian *header* menunjukkan adanya interaksi pada halaman antara user dengan konten.
- B. Hyperlink menunjukkan ada link menuju informasi register pilihan user.
- C.

4.2.4.3 User Interface Halaman Register Vendor Travel



Gambar 4.27 User Interface Halaman Register Vendor Travel

Keterangan :

- UI Interaction* pada bagian *header* menunjukkan adanya interaksi pada halaman antara user dengan konten.
- Form Element* menunjukkan ada form register untuk vendor travel.

4.2.4.4 User Interface Halaman Register Customer

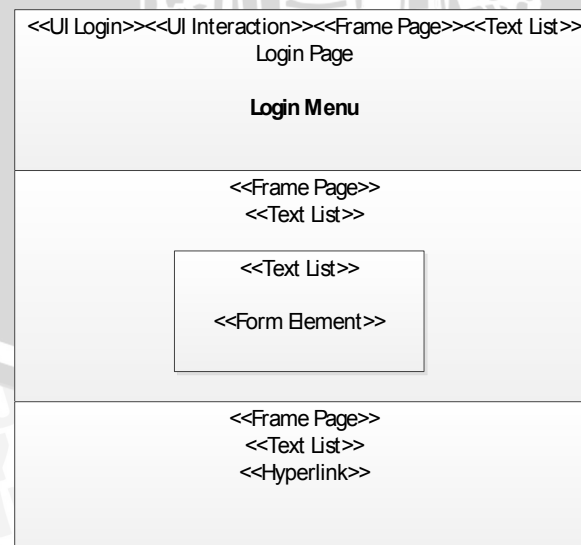


Gambar 4.28 User Interface Halaman Register Customer

Keterangan :

- UI Interaction* pada bagian *header* menunjukkan adanya interaksi pada halaman antara user dengan konten.
- Form Element* menunjukkan ada form register untuk *customer*.

4.2.4.5 User Interface Halaman Login

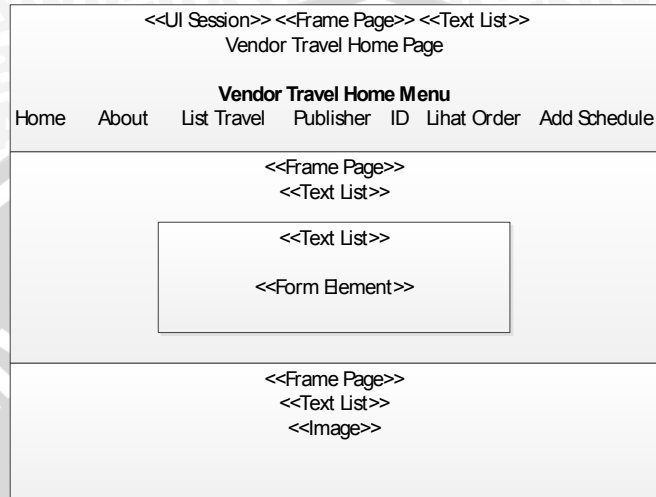


Gambar 4.29 User Interface Halaman Login

Keterangan :

- A. *UI Login* pada bagian *header* menunjukan halaman login.
- B. *UI Interaction* adanya interaksi pada halaman antara user dengan konten.
- C. *Form Element* menunjukan ada form login untuk user.

4.2.4.6 User Interface Halaman Home Vendor Travel

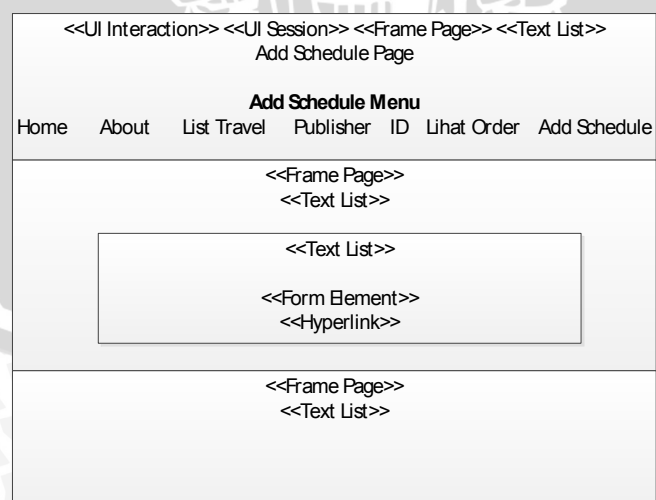


Gambar 4.30 User Interface Halaman Home Vendor Travel

Keterangan :

- A. *UI Session* pada bagian *header* menunjukan halaman *session* dari vendor travel.
- B. *Form Element* menunjukan form yang dapat di isi.

4.2.4.7 User Interface Halaman Add Schedule Travel

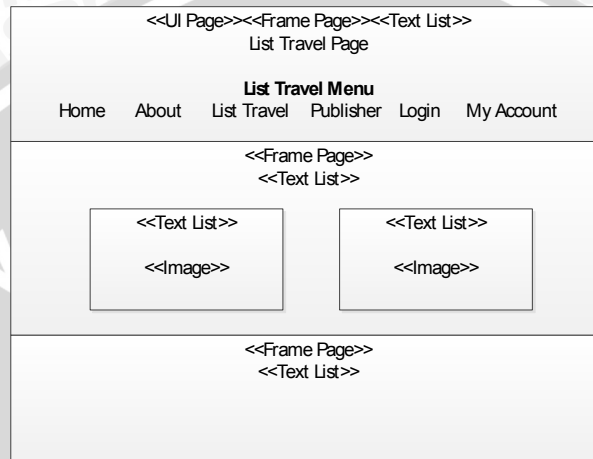


Gambar 4.31 User Interface Halaman Add Schedule Travel

Keterangan :

- UI Session* pada bagian *header* menunjukkan halaman *session* dari vendor travel .
- UI Interaction* menunjukkan adanya interaksi pada halaman antara user dengan konten.
- Form Element* menunjukkan form yang dapat di isi jadwal oleh vendor travel.

4.2.4.8 User Interface Halaman List Travel

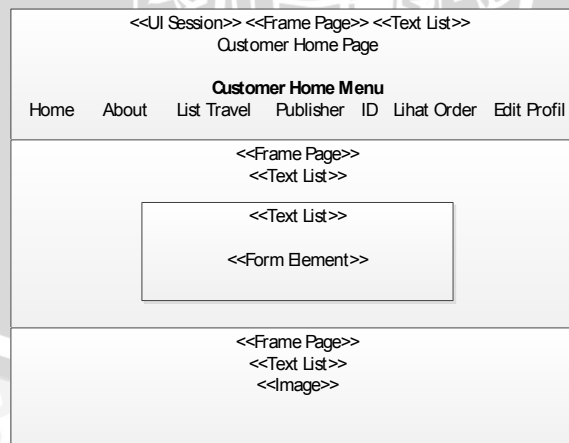


Gambar 4.32 User Interface Halaman List Travel

Keterangan :

- UI Page* pada bagian *header* menunjukkan halaman *tanpa* adanya interaksi pada halaman antara user dengan konten.
- Image* menunjukkan adanya informasi dengan gambar.

4.2.4.9 User Interface Halaman Home Customer

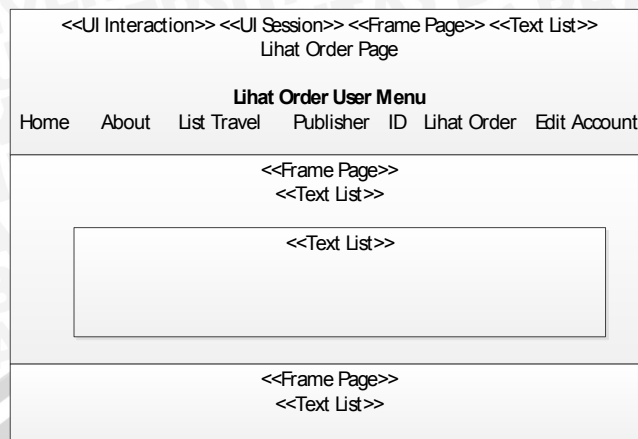


Gambar 4.33 User Interface Halaman Home Customer

Keterangan :

- UI Session* pada bagian *header* menunjukkan halaman *session* dari customer.
- Form Element* menunjukkan form yang dapat di isi.

4.2.4.12 User Interface Halaman Lihat Order



Gambar 4.36 User Interface Halaman Lihat Order

Keterangan :

- A. *UI Session* pada bagian *header* menunjukan halaman *session* dari user.
- B. *UI Interaction* menunjukan interaksi user dengan halaman.
- C. *Text List* pada konten menunjukan adanya informasi berupa *text*.

4.2.4.13 User Interface Halaman Publisher

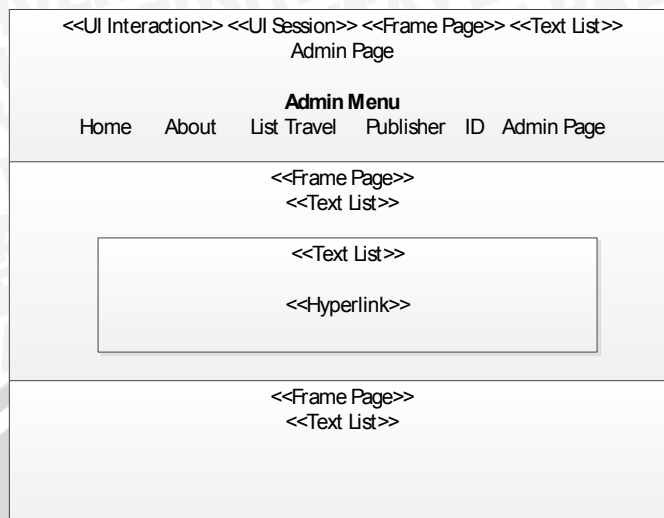


Gambar 4.37 User Interface Halaman Publisher

Keterangan :

- A. *UI Page* pada bagian *header* menunjukan halaman tanpa interaksi.
- B. *UI Image* menunjukan informasi dengan gambar pada konten.

4.2.4.14 User Interface Halaman Admin



Gambar 4.38 User Interface Halaman Admin

Keterangan :

- A. *UI Session* pada bagian *header* menunjukkan halaman *session* dari user.
- B. *UI Interaction* menunjukkan interaksi user dengan halaman.
- C. *Text List* pada konten menunjukkan adanya informasi berupa *text*.

4.2.5 Perancangan penyimpanan data

Perancangan data memiliki tujuan mendapatkan struktur data dan *database* yang akan di gunakan oleh perangkat lunak. Struktur data diperoleh berdasarkan kebutuhan perangkat lunak. Data kemudian disimpan dalam database. Pemetaan didalam database direpresentasikan dalam bentuk pemodelan *entity relationship diagram* (ERD). Struktur penyimpanan data sistem *marketplace* pemesanan tiket travel di tunjukan pada ERD berikut:

A. Tabel Schedule

Tabel schedule memiliki fungsi menyimpan data jadwal travel.

B. Tabel User

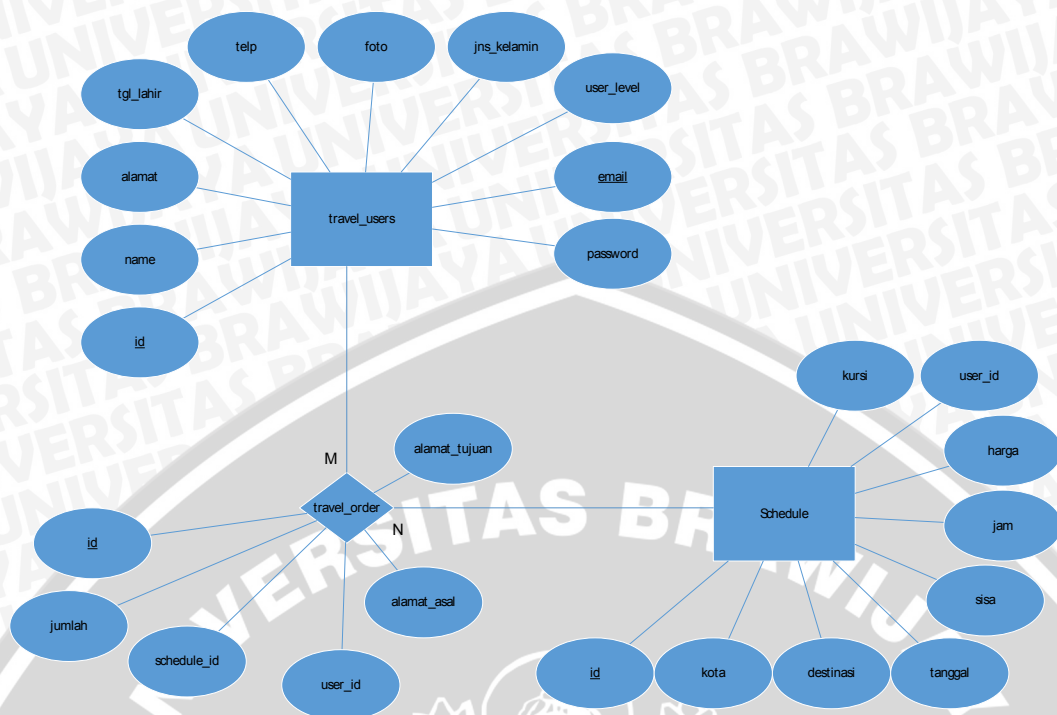
Tabel *user* memiliki fungsi menyimpan data *customer* dan admin.

C. Tabel Travel

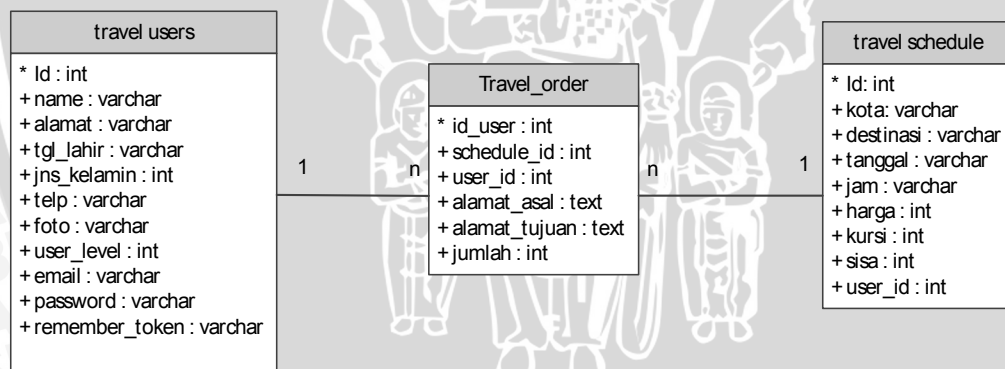
Tabel travel memiliki fungsi menyimpan data vendor travel.

D. Tabel Order

Tabel order memiliki fungsi menyimpan data order.



Gambar 4.39 Gambar Entity Relationship Diagram Sistem



Gambar 4.40 Perancangan Penyimpanan Data

BAB 5 IMPLEMENTASI

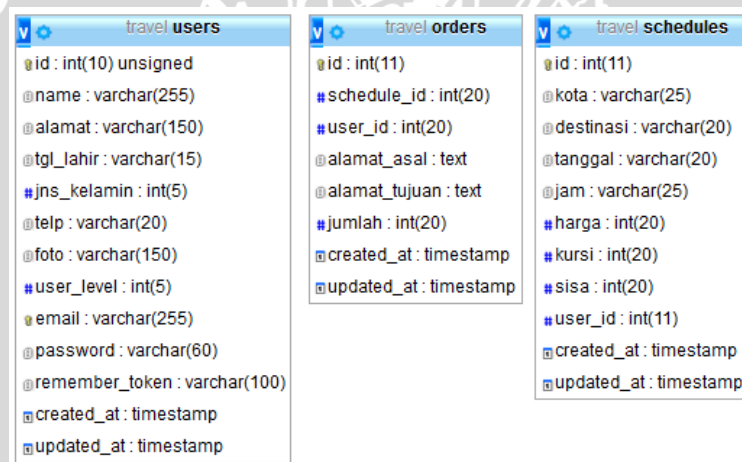
Pada bab ini akan dibahas tentang implementasi pembuatan aplikasi dan pengujian pada aplikasi yang dilakukan berdasarkan metode penelitian dan perancangan dari sistem yang telah dibuat lalu menguraikan hasil analisis dari pengujian.

4.1 Implementasi

Berdasarkan perancangan dan analisis kebutuhan yang telah dijelaskan sebelumnya, berikut adalah penjelasan tentang implementasi dari tiap tahapan proses. Implementasi yang dilakukan dibedakan menjadi implementasi yang dilakukan pada basis data, *user interface* dan program.

4.1.1 Implementasi Basis Data

Implementasi *database* dilakukan dengan *MySQL* berdasarkan *class diagram*. Hasil implementasi tersebut kemudian dimodelkan menjadi *physical diagram*. Gambar 5.1 berikut ini merupakan hasil pemodelan menjadi *physical diagram*.



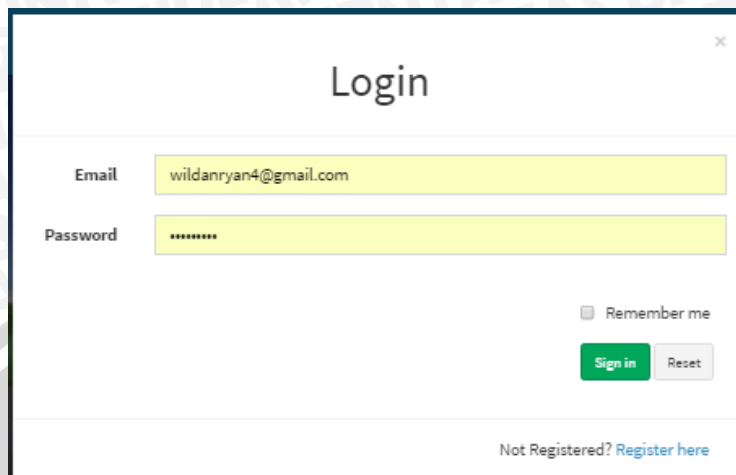
Gambar 5.1 *Physical Diagram*

4.1.2 Implementasi User Interface

Pada implementasi *user interface* ini dibagi menjadi tiga modul berdasarkan aktor yaitu modul admin, vendor travel, dan *user*. Di mana setiap modul akan menampilkan *user interface* dari fitur-fitur yang dimiliki sesuai dengan perancangan aplikasi.

4.1.2.1 User Interface Admin

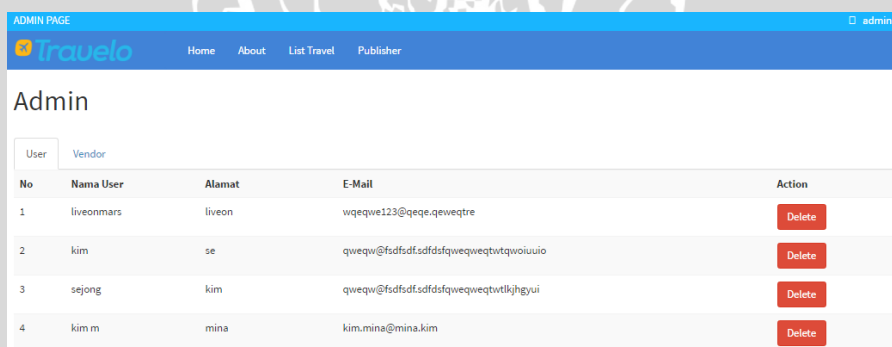
A. Halaman Login



Gambar 5.2 Halaman Login User

Gambar 5.2 adalah gambar tampilan dari bagaimana user melakukan login. Terdapat *form email* dan *password* untuk diisi oleh user. Di sini menentukan user masuk sebagai admin, *customer* atau vendor travel.

B. Halaman Admin



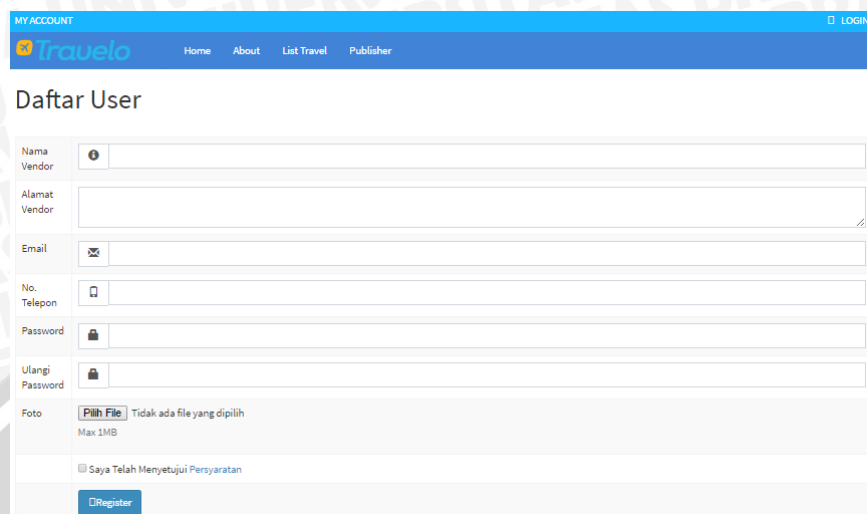
No	Nama User	Alamat	E-Mail	Action
1	liveonmars	liveon	wqeque23@qeque.qequeqtre	Delete
2	kim	se	qweqw@fsdfsdfsdfsfsqweqewtwtqwoiuio	Delete
3	sejong	kim	qweqw@fsdfsdfsdfsfsqweqewtwtlkguyui	Delete
4	kim m	mina	kim.mina@mina.kim	Delete

Gambar 5.3 Halaman Admin

Gambar 5.3 adalah gambar tampilan dari halaman admin. Halaman admin menampilkan list dari vendor travel dan *customer* yang telah terdaftar pada sistem. Terdapat fitur untuk menghapus vendor travel dan *customer*.

4.1.2.2 User Interface Vendor Travel

A. Halaman Register Vendor Travel



MY ACCOUNT LOGIN

Travelo Home About List Travel Publisher

Daftar User

Nama Vendor

Alamat Vendor

Email

No. Telepon

Password

Ulangi Password

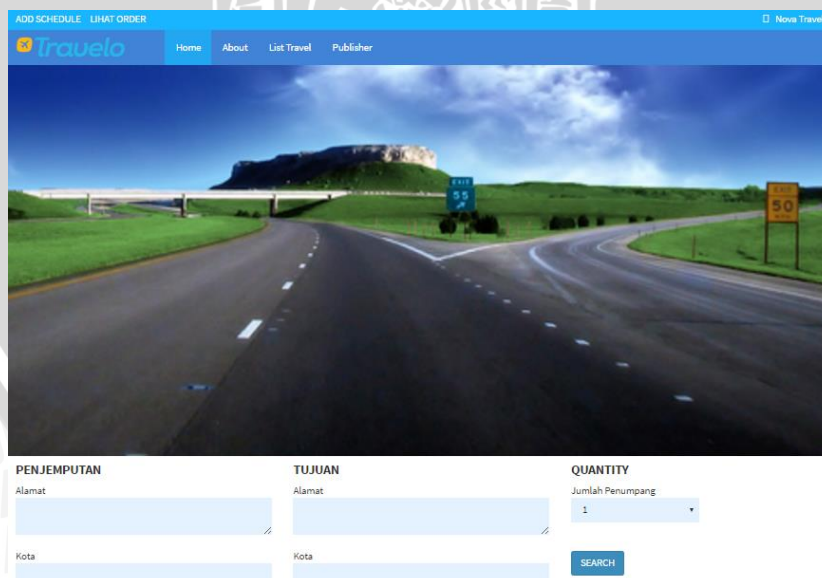
Foto Tidak ada file yang dipilih
Max 1MB

☐ Saya Telah Menyetujui Persyaratan

Gambar 5.4 Halaman *Register Vendor Travel*

Gambar 5.4 adalah gambar tampilan dari halaman register vendor travel. Halaman register vendor travel menampilkan *form* yang akan diisi vendor travel untuk mendaftar pada sistem.

B. Halaman *Home Vendor Travel*



ADD SCHEDULE LIHAT ORDER Novus Travel

Travelo Home About List Travel Publisher

PENJEMPUTAN

Alamat

Kota

TUJUAN

Alamat

Kota

QUANTITY

Jumlah Penumpang

Gambar 5.5 Halaman *Home Vendor Travel*

Gambar 5.5 adalah gambar tampilan dari halaman *session* dari vendor travel. Halaman *home* vendor travel menampilkan *menu add schedule* dan lihat order yang dapat di lihat oleh vendor travel pada sistem.

C. Halaman Add Schedule Vendor Travel

No	Dari	Tujuan	Tanggal	Waktu	Harga	Kursi	Action
1	Malang	Mojokerto	2016-05-01		75000	20	Edit Delete

[ADD](#)

Gambar 5.6 Halaman Add Schedule Vendor Travel

Gambar 5.6 adalah gambar tampilan dari halaman *add schedule* vendor travel. Halaman ini di gunakan untuk menambah, mengedit, dan menghapus jadwal vendor travel. Halaman ini hanya dapat di akses oleh vendor travel yang terdaftar pada sistem.

4.1.2.3 User Interface Customer

A. Halaman Register Customer

Daftar User

Nama

Alamat

Tanggal Lahir

Jenis Kelamin ☒ Laki laki ☐ Perempuan

No. Telepon

Email

Password

Ulangi Password

Foto Tidak ada file yang dipilih
Max 1MB

☐ Saya Telah Menyetujui Persyaratan

[Register](#)

Gambar 5.7 Halaman Register Customer

Gambar 5.7 adalah gambar tampilan dari halaman *register customer*. Terdapat form pada halaman ini untuk diisi oleh *customer* yang akan mendaftar pada sistem.

B. Halaman *Home Customer*

Gambar 5.8 Halaman *Home Customer*

Gambar 5.8 adalah gambar tampilan dari halaman session *customer*. Terdapat beberapa menu pada halaman home *customer* yaitu edit profil, lihat order dan form untuk pemesanan travel. Form pemesanan yang telah diisi selanjutnya akan di proses oleh sistem untuk mencari vendor travel.

C. Halaman Pemesanan

Gambar 5.9 Halaman Pemesanan

Gambar 5.10 adalah gambar tampilan dari halaman pemesanan. Halaman ini adalah proses lanjutan setelah *customer* mengisi form pada halaman home untuk mencari vendor travel yang di inginkan.

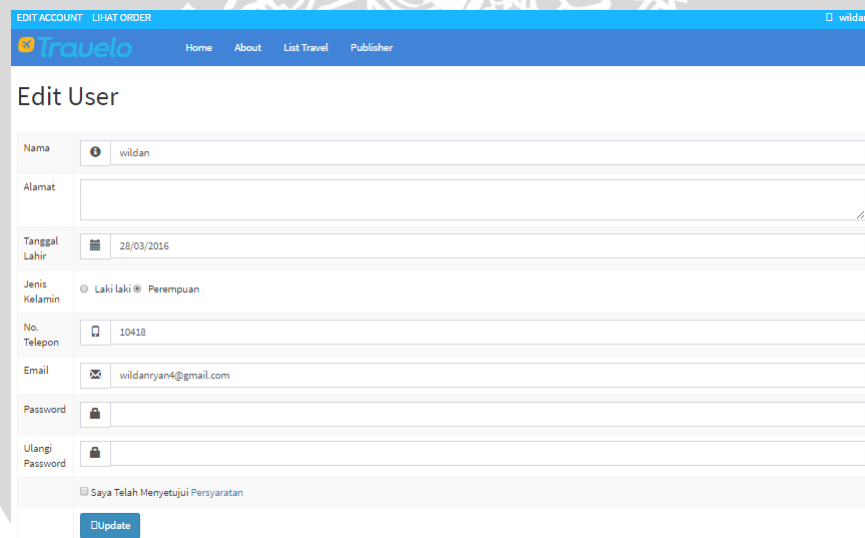
D. Halaman Konfirmasi Pemesanan



Gambar 5.10 Halaman Konfirmasi Pemesanan

Gambar 5.11 adalah gambar tampilan dari halaman konfirmasi pemesanan. Halaman ini berfungsi untuk memberikan konfirmasi kepada *customer*. Halaman ini berisi informasi travel yang di pesan dan total harga yang harus di bayar oleh *customer*.

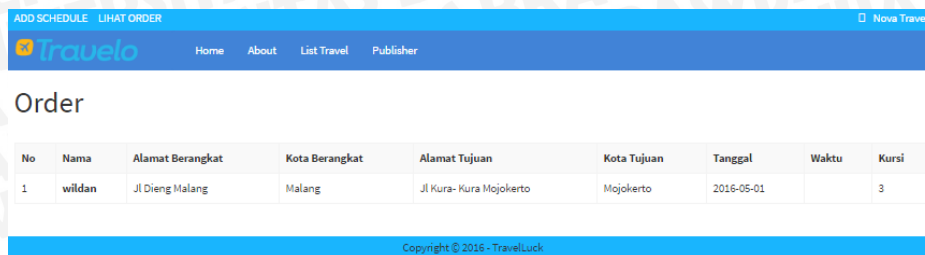
E. Halaman Edit Profil *Customer*



Gambar 5.11 Halaman *Edit Profil Customer*

Gambar 5.12 adalah gambar tampilan dari halaman edit profil *customer*. Halaman ini berfungsi untuk mengedit profil *customer*.

F. Halaman Lihat *Order*



Gambar 5.12 Halaman Lihat *Order*

Gambar 5.13 adalah gambar tampilan dari halaman lihat order. Halaman ini dapat di akses oleh vendor travel dan *customer* berfungsi untuk melihat *order* yang masuk dan melihat jadwal keberangkatan.

4.1.3 Implementasi Program

1. Register Vendor Travel

Berikut ini adalah potongan kode untuk *register vendor travel*.

```

1 {
2     $file = Input::file('image');
3     $file = array('image' => Input::file('image'));
4     $rules = array('image' => 'required|mimes:jpeg,bmp,png',); //mimes:jpeg,bmp,png
5     and for max size max:10000
6     $validator = Validator::make($file, $rules);
7     if ($validator->fails()) {
8         return Redirect::to('upload')->withInput()->withErrors($validator);
9     }
10    else {
11        if (Input::file('image')->isValid()) {
12            $destinationPath = 'uploads'; // upload path
13            $extension = Input::file('image')->getClientOriginalExtension(); // getting image
14            extension
15            $fileName = $data['email'].'.'.$extension; // renameing image
16            Input::file('image')->move($destinationPath, $fileName); // uploading file to given
17            path
18            echo "sukses";
19        }
20        else {
21            echo "gagal";
22        }
23    }
24 }
```

```

21     }
22     }
23
24     return User::create([
25         'name' => $data['name'],
26         'alamat' => $data['alamat'],
27         'tgl_lahir' => "",
28         'jns_kelamin' => "",
29         'telp' => $data['telp'],
30         'foto' => $fileName,
31         'user_level' => 0,
32         'email' => $data['email'],
33         'password' => bcrypt($data['password']),
34     ]);
35 }
36

```

Gambar 5.13 Cuplikan Kode *Register Vendor Travel*

2. Register Customer

Berikut ini adalah potongan kode untuk *register customer*.

```

1  protected function create(array $data)
2  {
3      $file = Input::file('image');
4      $file = array('image' => Input::file('image'));
5      $rules = array('image' => 'required|mimes:jpeg,bmp,png',); //mimes:jpeg,bmp,png
6      and for max size max:10000
7      $validator = Validator::make($file, $rules);
8      if ($validator->fails()) {
9          return Redirect::to('upload')->withInput()->withErrors($validator);
10     }
11     else {
12         if (Input::file('image')->isValid()) {
13             $destinationPath = 'uploads'; // upload path
14             $extension = Input::file('image')->getClientOriginalExtension(); // getting image
15             extension
16             $fileName = $data['email'].'.'.$extension; // renameing image

```

```

17      Input::file('image')->move($destinationPath, $fileName); // uploading file to given
18      path
19          echo "sukses";
20      }
21      else {
22          echo "gagal";
23      }
24  }
25
26      return User::create([
27          'name' => $data['name'],
28          'alamat' => $data['alamat'],
29          'tgl_lahir' => $data['tgl_lahir'],
30          'jns_kelamin' => $data['jns_kelamin'],
31          'telp' => $data['telp'],
32          'foto' => $fileName,
33          'user_level' => 1,
34          'email' => $data['email'],
35          'password' => bcrypt($data['password']),
36      ]);
37
38  }
39

```

Gambar 5.14 Cuplikan Kode *Register Customer*

3. Tambah Jadwal

Berikut ini adalah potongan kode untuk tambah jadwal.

```

1      Route::post('/addSc', function(){
2
3          $id_travel = Auth::user()->id;
4          $newArticle = new schedule();
5          $newArticle->kota = Input::get("kota");
6          $newArticle->destinasi = Input::get("destinasi");
7          $newArticle->tanggal = Input::get("tanggal");
8          $newArticle->jam = Input::get("jam");
9          $newArticle->harga = Input::get("harga");

```

```

10     $newArticle->kursi = Input::get("kursi");
11     $newArticle->sisa = Input::get("kursi");
12     $newArticle->user_id = $id_travel;
13     $newArticle->save();
14
15     return redirect('/schedule');
16 });
17
18

```

Gambar 5.15 Cuplikan Kode Tambah Jadwal

4. Pemesanan

Berikut ini adalah potongan kode untuk pemesanan.

```

1  protected function confirmDone () {
2      $newArticle = new \App\order();
3      $newArticle->schedule_id = Input::get("schedule_id");
4      $newArticle->user_id = Input::get("user_id");
5      $newArticle->jumlah = Input::get("kursi");
6      $newArticle->alamat_asal = \Session::get("alamatAsal");
7      $newArticle->alamat_tujuan = \Session::get("alamatTujuan");
8      $newArticle->save();
9
10     $newArticle = schedule::find(Input::get("schedule_id"));
11     $newArticle->sisa = $newArticle->sisa-Input::get("kursi");
12     $newArticle->save();
13
14     \Session::forget('alamatAsal');
15     \Session::forget('alamatTujuan');
16     \Session::forget('kotaAsal');
17     \Session::forget('kotaTujuan');
18     \Session::forget('jumlah');
19     \Session::forget('tanggal');
20
21     return redirect('/user-page');
22
23 }

```

Gambar 5.16 Cuplikan Kode Pemesanan

5. Konfirmasi Pemesanan

Berikut ini adalah potongan kode untuk konfirmasi pemesanan.

```

1  protected function confirmDone () {
2      $newArticle = new \App\order();
3      $newArticle->schedule_id = Input::get("schedule_id");
4      $newArticle->user_id = Input::get("user_id");
5      $newArticle->jumlah = Input::get("kursi");
6      $newArticle->alamat_asal = \Session::get("alamatAsal");
7      $newArticle->alamat_tujuan = \Session::get("alamatTujuan");
8      $newArticle->save();
9
10     $newArticle = schedule::find(Input::get("schedule_id"));
11     $newArticle->sisa = $newArticle->sisa-Input::get("kursi");
12     $newArticle->save();
13
14     \Session::forget('alamatAsal');
15     \Session::forget('alamatTujuan');
16     \Session::forget('kotaAsal');
17     \Session::forget('kotaTujuan');
18     \Session::forget('jumlah');
19     \Session::forget('tanggal');
20
21     return redirect('/user-page');
22
23 }
24

```

Gambar 5.17 Cuplikan Kode Konfirmasi Pemesanan

BAB 6 PENGUJIAN

5.1 Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah dibuat menggunakan *functional testing* dan *non-functional testing*. Di mana pada *non-functional testing* melibatkan *useability testing*. Pengujian ini memiliki tujuan untuk menjamin bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan fungsional dan kebutuhan non-fungsional yang telah dibuat.

5.1.1 Pengujian Fungsional

Functional testing adalah jenis *black box testing* yang mendasarkan kasus uji pada kebutuhan fungsional yang telah dibuat. Setiap fitur diuji dengan memberi masukan dan memeriksa keluaran apakah sesuai dengan. Pengujian ini berdasarkan spesifikasi kebutuhan fungsional yang telah dibuat pada setiap Aktor. Parameter yang digunakan dalam pengujian ini adalah valid dan tidak valid. Di mana pengujian fungsional dikatakan valid jika hasil sesuai dengan yang diharapkan dan gagal ketika hasil berbeda dengan apa yang diharapkan.

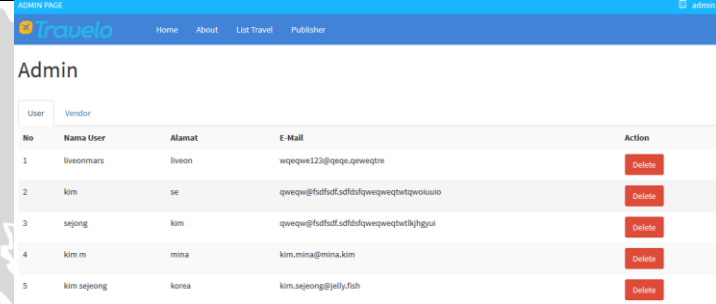
A. Pengujian Fungsional Admin

Tabel 6.1 Pengujian Fungsional Admin

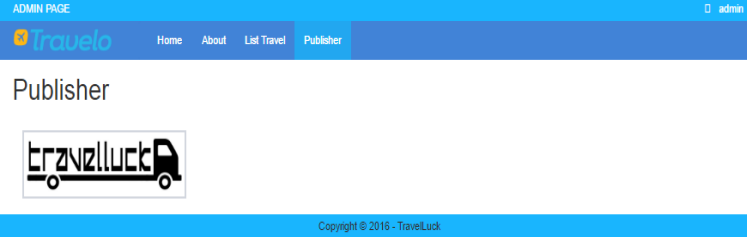
ID Kebutuhan Fungsional	ID <i>Functional Testing</i>	Kebutuhan	Hasil
KF-03	FT-01	Sistem bisa melakukan login untuk admin	Valid
KF-12	FT-02	Perangkat lunak harus bisa memberikan informasi travel publisher untuk admin	Valid
KF-15	FT-03	Perangkat lunak harus bisa lihat keseluruhan vendor travel yang terdaftar untuk admin	Valid
KF-18	FT-04	Perangkat lunak harus bisa mengelola akun vendor travel dan customer	Valid
KF-21	FT-05	Perangkat lunak harus bisa melakukan logout untuk admin	Valid

Berikut ini adalah penjelasan pengujian dari masing-masing ID *functional testing*:

Tabel 6.2 Functional Testing Fungsional Login Admin

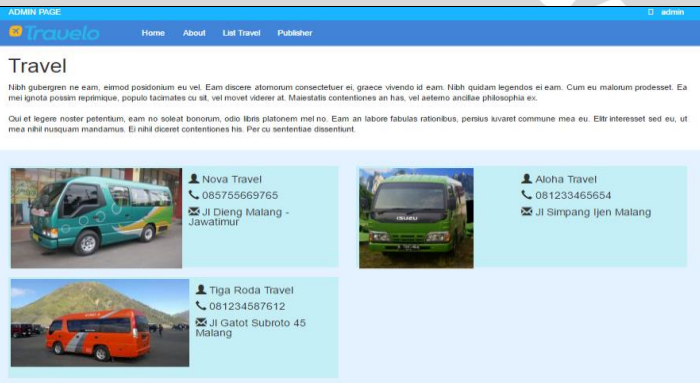
ID Fuctional Testing:	FT-01
Objek:	Pengujian Pada Login Admin
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional login untuk admin.
Langkah-Langkah:	1. Memilih menu login pada sistem 2. Memasukan email dan password admin 3. Memilih tombol <i>login</i>
Hasil yang Diharapkan:	User berhasil masuk ke dalam sistem sebagai admin
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.3 Functional Testing Fungsional Travel Publisher

ID Fuctional Testing:	FT-02
Objek:	Pengujian Pada Travel Publisher
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional memeberikan informasitravel publisher untuk admin.
Langkah-Langkah:	1. Memilih menu <i>login</i> pada sistem 2. Memasukan <i>email</i> dan <i>password</i> admin 3. Memilih tombol <i>login</i> 4. Memilih menu <i>travel publisher</i> pada halaman home
Hasil yang Diharapkan:	Admin berhasil mendapatkan informasi travel publisher
Tampilan:	

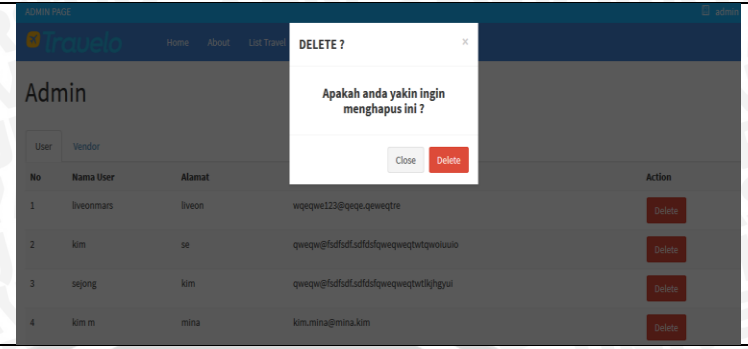
Hasil yang Diperoleh:	Valid
-----------------------	-------

Tabel 6.4 *Functional Testing Fungsional List Travel*

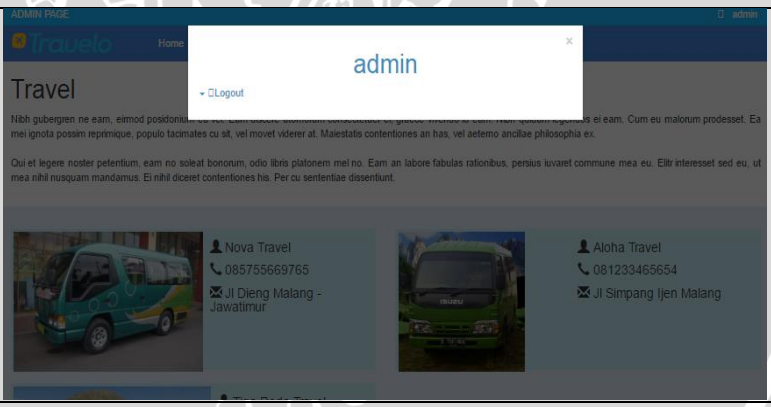
ID Fuctional Testing:	FT-03
Objek:	Pengujian Pada List Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional memberikan informasi list travel untuk admin.
Langkah-Langkah:	1. Memilih menu <i>login</i> pada sistem 2. Memasukan <i>email</i> dan <i>password</i> admin 3. Memilih tombol <i>login</i> 4. Memilih menu <i>list travel</i> pada halaman home
Hasil yang Diharapkan:	Admin berhasil mendapatkan informasi list travel
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.5 *Functional Testing Mengelola Akun*

ID Fuctional Testing:	FT-04
Objek:	Pengujian Fitur Admin Manage Akun
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional manage akun dengan menghapus akun vendor travel atau customer oleh admin.
Langkah-Langkah:	1. Memilih menu customer pada admin page 2. Memilih customer yang akan di hapus 3. Memilih tombol delete 4. Memilih tombol <i>delete</i> kembali pada konfirmasi
Hasil yang Diharapkan:	Customer berhasil di hapus dari sistem

Tampilan:	
Hasil Diperoleh:	yang Valid

Tabel 6.6 Functional Testing Logout

ID Fuctional Testing:	FT-05
Objek:	Pengujian Fitur Logout
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional logout untuk admin.
Langkah-Langkah:	1. Memilih menu logout pada halaman home 2. Memilih tombol logout pada halaman konfirmasi
Hasil yang Diharapkan:	Admin berhasil logout dari sistem
Tampilan:	
Hasil Diperoleh:	yang Valid

B. Pengujian Fungsional Vendor Travel

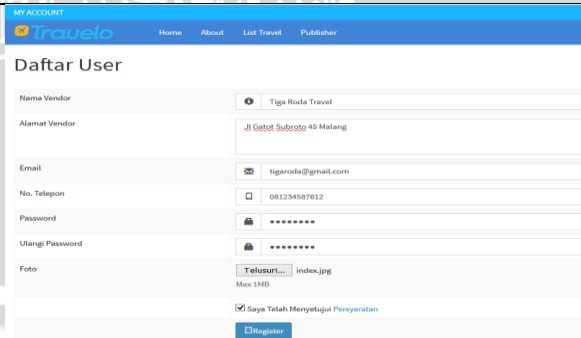
Tabel 6.7 Pengujian Fungsional Vendor Travel

ID Kebutuhan Fungsional	ID Functional Testing	Kebutuhan	Hasil
KF-02	FT-06	Sistem bisa melakukan register untuk vendor travel	Valid
KF-05	FT-07	Sistem bisa melakukan login untuk	Valid

		vendor travel.	
KF-17	FT-08	Sistem mempunyai fitur untuk manage jadwal vendor travel.	Valid
KF-11	FT-09	Perangkat lunak harus bisa memberikan informasi travel publisher untuk vendor travel	Valid
KF-14	FT-10	Perangkat lunak harus bisa lihat keseluruhan vendor travel yang terdaftar untuk vendor travel	Valid
KF-09	FT-11	Sistem mempunyai fitur untuk melihat order masuk untuk vendor travel	Valid
KF-20	FT-12	Perangkat lunak harus bisa melakukan logout untuk vendor travel	Valid

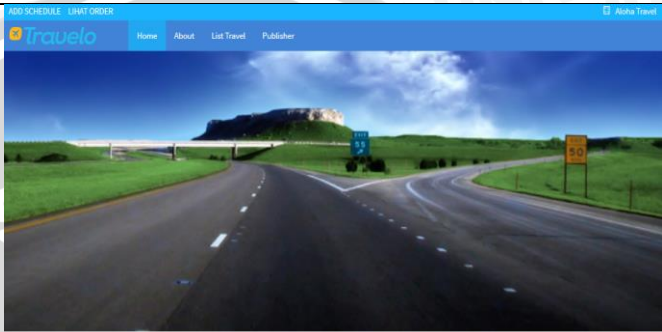
Berikut ini adalah penjelasan pengujian dari masing-masing ID *functional testing*:

Tabel 6.8 Functional Testing Register Vendor Travel

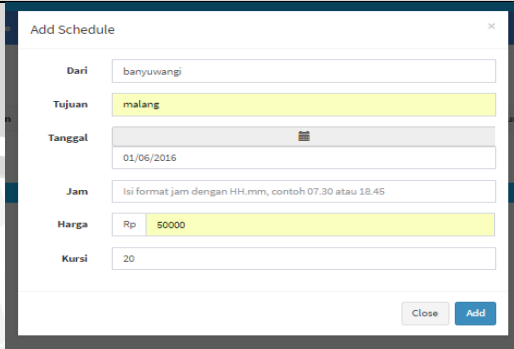
ID Fuctional Testing:	FT-06
Objek:	Pengujian Fitur Register Vendor Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional register untuk vendor travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Memilih menu register 3. Memilih tombol register vendor travel pada sistem 4. Mengisi indentitas yang diperlukan 5. Memilih tombol register setelah selesai
Hasil yang Diharapkan:	Vendor travel berhasil melakukan register
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.9 Functional Testing Login Vendor Travel

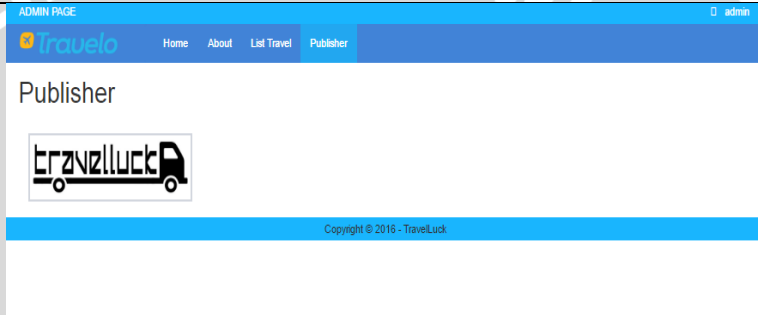
ID Fuctional Testing:	FT-07
Objek:	Pengujian Fitur Login Vendor Travel

Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional login untuk vendor travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password vendor travel 3. Memilih tombol login pada sistem
Hasil yang Diharapkan:	Vendor travel berhasil melakukan login
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.10 *Functional Testing* Mengatur Jadwal Vendor Travel

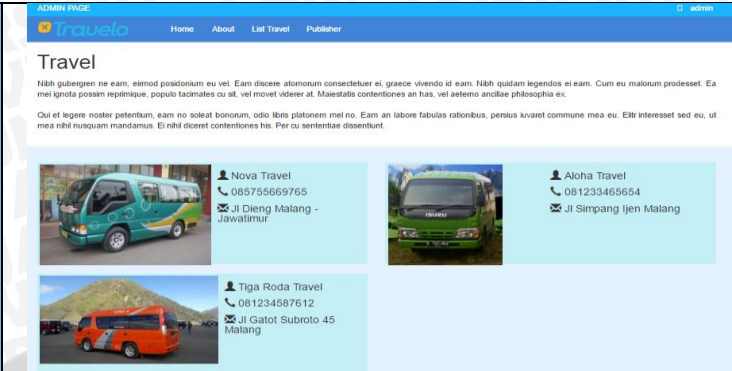
ID Fuctional Testing:	FT-08
Objek:	Pengujian Fitur Manage Jadwal Vendor Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional tambah jadwal untuk vendor travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password vendor travel 3. Memilih tombol login pada sistem 4. Memilih menu add schedule 5. Memilih menu add, edit, atau delete 6. Mengisi informasi jadwal 7. Memilih tombol add, edit, atau delete untuk selesai
Hasil yang Diharapkan:	Vendor travel berhasil memmanage jadwal
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.11 *Functional Testing Fungsional Travel Publisher*

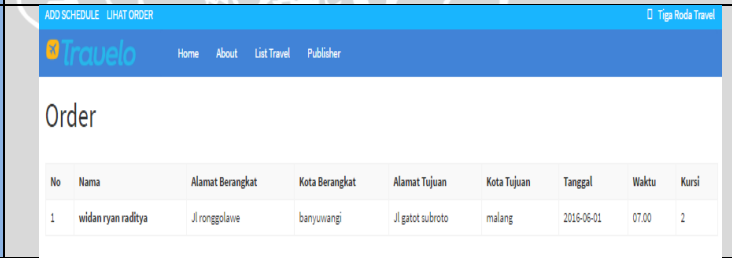
ID Fuctional Testing:	FT-09
Objek:	Pengujian Pada Travel Publisher
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional memeberikan informasitravel publisher untuk vendor travel.
Langkah-Langkah:	1. Memilih menu <i>login</i> pada sistem 2. Memasukan <i>email</i> dan <i>password</i> admin 3. Memilih tombol <i>login</i> 4. Memilih menu <i>travel publisher</i> pada halaman home
Hasil yang Diharapkan:	Vendor travel berhasil mendapatkan informasi travel publisher
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.12 *Functional Testing Fungsional List Travel*

ID Fuctional Testing:	FT-10
Objek:	Pengujian Pada List Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional memberikan informasi list travel untuk vendor travel.
Langkah-Langkah:	1. Memilih menu <i>login</i> pada sistem 2. Memasukan <i>email</i> dan <i>password</i> admin 3. Memilih tombol <i>login</i> 4. Memilih menu <i>list travel</i> pada halaman home
Hasil yang Diharapkan:	Vendor travel berhasil mendapatkan informasi list travel

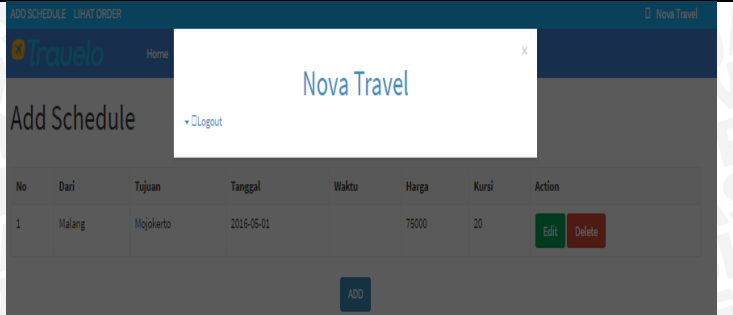
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.13 Functional Testing Lihat Order Vendor Travel

ID Fuctional Testing:	FT-11
Objek:	Pengujian Fitur Lihat Order Vendor Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional lihat order untuk vendor travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password vendor travel 3. Memilih tombol login pada sistem 4. Memilih menu lihat order
Hasil yang Diharapkan:	Vendor travel berhasil lihat order masuk
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.14 Functional Testing Logout

ID Fuctional Testing:	FT-12
Objek:	Pengujian Fitur Logout
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional logout untuk vendor travel.
Langkah-Langkah:	1. Memilih menu logout pada halaman home 2. Memilih tombol logout pada halaman konfirmasi
Hasil yang Diharapkan:	Vendor travel berhasil logout dari sistem

Tampilan:	
Hasil yang Diperoleh:	Valid

C. Pengujian Fungsional Customer

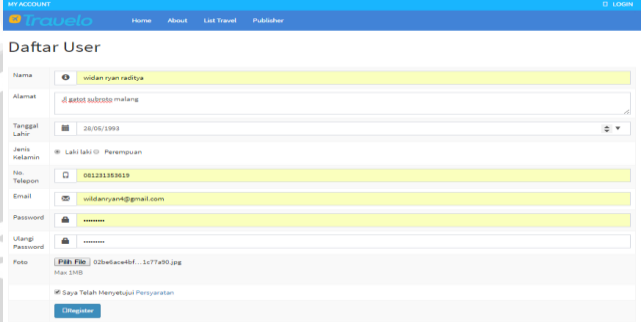
Tabel 6.15 Pengujian Fungsional Customer

ID Kebutuhan Fungsional	ID Functional Testing	Kebutuhan	Hasil
KF-01	FT-13	Sistem bisa melakukan register untuk Customer.	Valid
KF-04	FT-14	Sistem bisa melakukan login untuk Customer.	Valid
KF-16	FT-15	Sistem mempunyai fitur untuk edit profil.	Valid
KF-06	FT-16	Sistem mempunyai fitur untuk pemesanan travel.	Valid
KF-07	FT-17	Sistem mempunyai fitur untuk konfirmasi pemesanan travel.	Valid
KF-08	FT-18	Sistem mempunyai fitur untuk melihat order customer.	Valid
KF-13	FT-19	Sistem mempunyai fitur untuk melihat list travel.	Valid
KF-10	FT-20	Sistem mempunyai fitur untuk melihat travel publisher.	Valid
KF-19	FT-21	Perangkat lunak harus bisa melakukan logout untuk customer	Valid

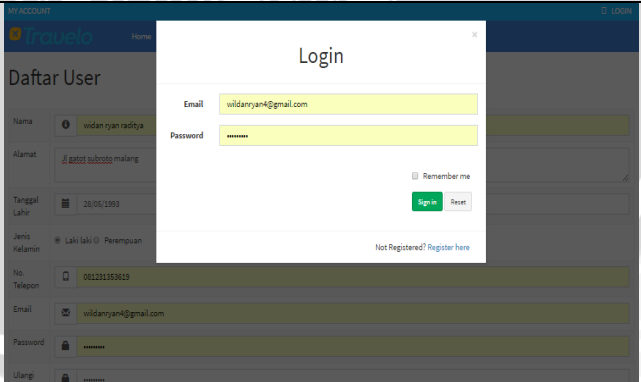
Berikut ini adalah penjelasan pengujian dari masing-masing ID *functional testing*:

Tabel 6.16 Functional Testing Register Customer

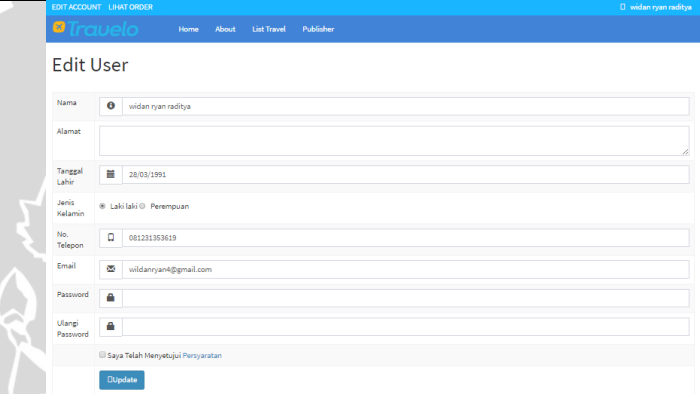
ID Fuctional Testing:	FT-13
Objek:	Pengujian Fitur Register Customer
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional register customer.

Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Memilih menu register 3. Memilih tombol register customer 4. Mengisi identitas yang diperlukan 5. Memilih tombol register setelah selesai
Hasil yang Diharapkan:	Customer berhasil melakukan register
Tampilan:	
Hasil yang Diperoleh:	Valid

Tabel 6.17 Functional Testing Login Customer

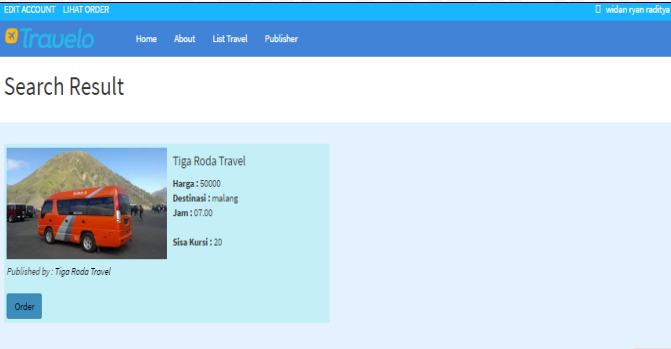
ID Fuctional Testing:	FT-14
Objek:	Pengujian Fitur Login Customer.
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional login untuk customer.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer 3. Memilih tombol login pada sistem
Hasil yang Diharapkan:	Customer berhasil melakukan login .
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.18 Functional Testing Edit Profil Customer

ID Fuctional Testing:	FT-15
Objek:	Pengujian Fitur Edit Profil Customer
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional edit profil customer.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer 3. Memilih tombol login pada sistem 4. Memilih menu edit profil 5. Mengedit identitas yang di perlukan 6. Memlih tombol update untuk selesai
Hasil yang Diharapkan:	Customer berhasil melakukan edit profil.
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.19 Functional Testing Pemesanan Travel

ID Fuctional Testing:	FT-16
Objek:	Pengujian Fitur Pemesanan
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional pemesanan travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer 3. Memilih tombol login pada sistem 4. Mengisi informasi pada kolom pemesanan 5. Memilih tombol search untuk mencari travel 6. Memilih tombol order untuk pesan
Hasil yang Diharapkan:	Customer berhasil melakukan pemesanan.

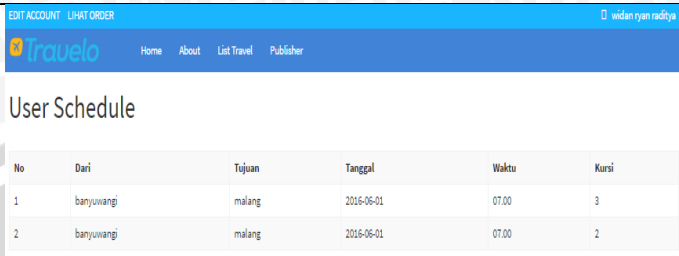
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.20 Functional Testing Konfirmasi Pemesanan

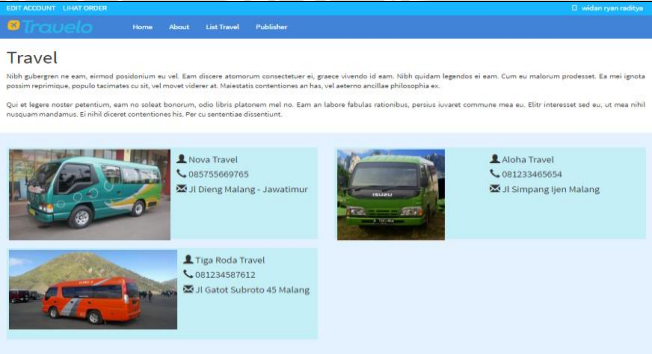
ID Fuctional Testing:	FT-17
Objek:	Pengujian Fitur Konfirmasi Pemesanan
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional konfirmasi pemesanan travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer 3. Memilih tombol login pada sistem 4. Mengisi informasi pada kolom pemesanan 5. Memilih tombol search untuk mencari travel 6. Memilih tombol order untuk pesan 7. Memilih tombol konfirmasi apabila informasi benar
Hasil yang Diharapkan:	Customer berhasil melakukan konfirmasi pemesanan.
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.21 Functional Testing Lihat Order

ID Fuctional Testing:	FT-18
Objek:	Pengujian Fitur Lihat Order Customer
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional lihat order customer.
Langkah-Langkah:	1. Memilih menu login pada halaman home

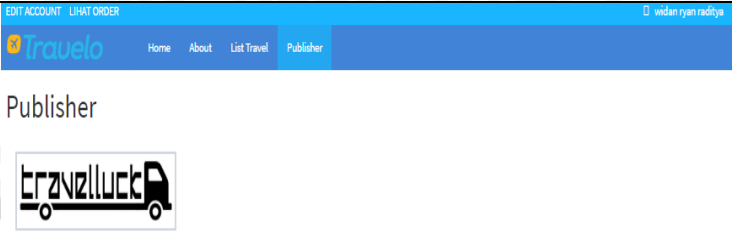
	2. Mengisi email dan password customer 3. Memilih tombol login pada sistem 4. Memilih tombol lihat order untuk customer
Hasil yang Diharapkan:	Customer berhasil melihat order.
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.22 Functional Testing List Travel

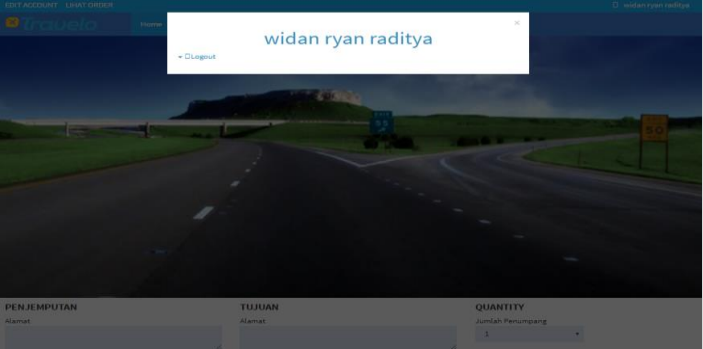
ID Fuctional Testing:	FT-19
Objek:	Pengujian Fitur List Travel
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional melihat list travel.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer 3. Memilih tombol login pada sistem 4. Memilih menu list travel
Hasil yang Diharapkan:	Customer berhasil melihat list travel.
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.23 Functional Testing Travel Publisher

ID Fuctional Testing:	FT-20
Objek:	Pengujian Fitur Travel Publisher
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional melihat travel publisher.
Langkah-Langkah:	1. Memilih menu login pada halaman home 2. Mengisi email dan password customer

	3. Memilih tombol login pada sistem 4. Memilih menu travel publisher
Hasil yang Diharapkan:	Customer berhasil melihat travel publisher.
Tampilan:	
Hasil yang Diperoleh:	Valid.

Tabel 6.24 *Functional Testing Logout*

ID Fuctional Testing:	FT-21
Objek:	Pengujian Fitur Logout
Tujuan:	Pengujian dilakukan untuk memastikan apakah sistem dapat memenuhi kebutuhan fungsional logout untuk vendor travel.
Langkah-Langkah:	1. Memilih menu logout pada halaman home 2. Memilih tombol logout pada halaman konfirmasi
Hasil yang Diharapkan:	Vendor travel berhasil logout dari sistem
Tampilan:	
Hasil yang Diperoleh:	Valid

5.1.2 Pengujian Non- Fungsional

Pada pengujian Non-Fungsional ini mengacu kepada 1 aspek yang telah dibuat pada perancangan yaitu *efficiency*. Untuk menguji aspek tersebut dibutuhkan *usability* testing. Pengujian ini dilakukan dengan cara membagikan kuesioner kepada calon *user*. Jenis kuesioner yang digunakan dalam pengujian *usability* ini adalah Skala Likert. Komponen pertanyaan yang digunakan untuk pengujian ditujukan pada Tabel 5.16 sebagai berikut:

Penilaian: 5 = Sangat Setuju 4 = Setuju 3 = Biasa
 2 = Tidak Setuju 1 = Sangat Tidak Setuju

Tabel 6.25 Kuesioner Pengujian *Usability*

No	Pertanyaan	Penilaian				
		R1	R2	R3	R4	R5
Aspek <i>Efficiency</i>						
1	Apakah aplikasi ini mempermudah dalam memperoleh informasi tentang <i>shuttle</i> atau vendor mobil travel?					
2	Apakah dengan menggunakan aplikasi ini dapat membantu vendor travel dalam mengelola jadwal?					
3	Apakah aplikasi ini mempermudah proses pemesanan pada <i>shuttle</i> atau vendor mobil travel?					

Pada Tabel 5.24 ditunjukkan pertanyaan-pertanyaan yang terdiri dari 1 aspek, yaitu aspek *efficiency*. Aspek *efficiency* menjelaskan seberapa cepat *user* mengoperasikan aplikasi untuk mendapatkan informasi. Dari pernyataan diatas didapat total jawaban sebanyak 40 jawaban dari 4 Responden yang bersangkutan yaitu Admin, Vendor travel, dan 2 customer. Hasil dari pengujian usability dijelaskan pada Tabel 5.25 berikut:

Tabel 6.26 Hasil Data *Usability Testing*

No	Pertanyaan	Penilaian				
		R1	R2	R3	R4	Nilai
1	Apakah aplikasi ini mempermudah dalam memperoleh informasi tentang <i>shuttle</i> atau vendor mobil travel?	4	4	4	4	4
2	Apakah dengan menggunakan aplikasi ini dapat membantu vendor travel dalam mengelola jadwal?	4	4	5	5	4.5

3	Apakah aplikasi ini mempermudah proses pemesanan pada <i>shuttle</i> atau vendor mobil travel?	5	4	5	5	4.75
Aspek <i>Efficiency</i>						4.41

Dari Hasil Pengujian pada Tabel 4.25 dapat ditemukan bahwa nilai tertinggi adalah elemen 3 yaitu “Apakah aplikasi ini mempermudah proses memberikan informasi nilai raport?” dengan nilai 4,75 dan Aspek *Efficiency* mendapatkan nilai rata-rata yaitu 4,41.

Dari *usability testing* yang sudah dilakukan maka dapat diperoleh hasil pengujian non-fungsional sebagai berikut:

Tabel 6.27 Hasil Pengujian Non-Fungsional

Kebutuhan <i>Non-Functional</i>		Nilai <i>Usability Testing</i>	Hasil
<i>Effecency</i>	Aplikasi harus memberikan kemudahan pada customer dalam melakukan pemesanan travel dan memberikan kemudahan vendor travel untuk mengelola jadwal serta memberikan informasi tentang travel <i>shuttle</i> .	Nilai 4,41 dengan prosentase 88,2%.	Tercapai

5.2 Analisis Pengujian

Proses Analisis Pengujian bertujuan untuk mendapatkan kesimpulan dari hasil pengujian perangkat lunak. Proses analisis mengacu pada hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisa pengujian fungsional.

A. Analisa Pengujian Fungsional

Proses analisis terhadap hasil pengujian fungsional dilakukan dengan memantau antara kinerja sistem dengan daftar kebutuhan. Hasil pengujian fungsionalitas pada *FT-01* sampai *FT-21* dengan metode *Black-Box Testing* secara keseluruhan adalah sangat baik. Hal ini dibuktikan dengan validnya semua pengujian fungsional berdasarkan kebutuhan fungsional. Jadi dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi dapat berjalan sesuai dengan spesifikasi kebutuhan yang telah dijabarkan pada analisis kebutuhan.

B. Analisa Pengujian Non Fungsional

Pada pengujian non-fungsional terdapat 1 aspek yang diuji yaitu aspek aspek *efficiency*. Pengujian yang dilakukan menggunakan pengujian *usability*. Dari

hasil pengujian *usability* didapatkan bahwa aplikasi dapat diterima oleh *user*. Nilai aspek *efficiency* 4.41 atau 88,2% dari nilai maksimal 5, menunjukkan bahwa aplikasi cukup memberikan kemudahan pada *customer* dalam melakukan pemesanan travel dan memberikan kemudahan vendor travel untuk mengelola jadwal serta memberikan informasi tentang travel *shuttle*.



BAB 7 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi dan pengujian yang telah dilakukan, maka diambil kesimpulan sebagai berikut:

1. Analisa pada sistem *marketplace* pemesanan travel menghasilkan kebutuhan fungsional, dan aktor yang berkaitan dengan aplikasi. Dari kebutuhan fungsional dirancang fitur-fitur pada aplikasi yaitu aplikasi dapat register (UC-01), login (UC-02), pemesanan travel (UC-03), konfirmasi pemesanan (UC-04), lihat order (UC-05), travel publisher (UC-06), list vendor travel (UC-07), edit profil (UC-08), Manage jadwal (UC-09), manage akun (UC-10), dan logout (UC-11). Kemudian dari fitur-fitur tersebut dibuat pemodelan dengan UML (*Unified Modelling Language*) yang menghasilkan *use case diagram*, *activity diagram*, dilanjutkan dengan menggunakan metode Com HDM (*Comprehensive Hypermedia Method*) untuk menggambarkan navigasi dan interaksi menuju informasi yang diperlukan.
2. Implementasi sistem *marketplace* pemesanan travel berdasarkan perancangan yang sudah dilakukan. Implementasi tersebut menghasilkan *physical diagram* dari *class diagram* yang telah dirancang sebelumnya. Selain itu dari perancangan pemodelan dihasilkan aplikasi sistem *markeplace* pemesanan travel berbasis *website* sesuai dengan fitur-fitur yang telah dibuat.
3. Pengujian sistem *marketplace* pemesanan travel dilakukan dengan pengujian fungsional. Dari pengujian fungsional didapatkan hasil bahwa aplikasi berjalan sesuai dengan kebutuhan fungsional dengan akurasi 100%. Pengujian non fungsional yang dilakukan berdasarkan kebutuhan non fungsional yaitu *efficiency*. Dari pengujian non fungsional didapatkan bahwa aplikasi cukup aplikasi membantu *customer* dalam melakukan pemesanan travel dan memberikan kemudahan vendor travel untuk mengelola jadwal serta memberikan informasi tentang travel *shuttle*.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi Sistem Marketplace Pemesanan Travel Berbasis Web selanjutnya adalah sebagai berikut:

1. Agar data dan implementasi metode ini dapat diimplementasikan dengan versi mobile
2. Penelitian lebih lanjut dapat menggunakan metode perancangan desain yang lain seperti *UML Extension Web Design Model* (UEWDM), karena metode desain terbukti berhasil membantu pada fase perancangan .

DAFTAR PUSTAKA

- Naftha, D.A., 2013. Penetapan Faktor Kualitas Layanan Jasa Travel Cipaganti Dengan Metode Principal Component Analysis. Ciputat . Tersedia di: <<http://library.gunadarma.ac.id>> [Diakses 2 Januari 2016].
- Pressman R.S., 2010. Software Engineering A Practitioners Approach. Ebook.
- Mubin, S.A., 2016. *UML-based Conceptual Design Approach for Modeling Complex Processes in Web Application*. International Journal of Applied Engineering Research. Volume 11, Number 6. Universiti Putra Malaysia.
- Jantan, A.H., Halin, A.A. & Mansor, E.I., 2013 .*Modeling Complex Processes, Navigations, and User Interface in Web Applications*. International Journal of Computer and Information Technology. Volume 02. Issue 05. Universiti Putra Malaysia.
- Sommerville, Ian. 2011 .*Software Engineering*. Ninth Edition. Addison-Wesley. New York.
- Sumari, P., Sulaiman, S., 2008. *Conceptual Modeling in Web-based Hypermedia Engineering with Com+HDM. Proceedings of the 4th International Conference on Information Technology and Multimedia*. Universiti Sains Malaysia.
- Andrianto, E., 2011. Tugas Pemrograman Web. Tersedia di: <<http://eprints.uad.ac.id/view/year/2011.html>> [Diakses 3 Februari 2016].
- Mardiyanto, S., 2011. Rancang Bangun Website Pertemanan Menggunakan Ajax Framework Untuk Komunitas Pjj. Politeknik Elektronika Negeri Surabaya.
- Simatupang, R.M., 2014. Penerapan Metode User Centered Design Untuk Perancangan Aplikasi Radio Steraming Berbasis Web. STMIK Budidarma Medan.
- Putra, R., 2013. Aplikasi Penjualan Berbasis Web Pada PT. PRATAPA NIRMALA Palembang. STMIK GI MDP Palembang.
- Sumaryadi, C., 2012. Peluang Bisnis pada Bidang Jasa Travel dan Shuttle. Tersedia di: <[http:// amikom.ac.id](http://amikom.ac.id)> [Diakses 10 Februari 2016].
- Siregar, A Ridwan., 2008. Pemanfaatan Teknologi Informasi untuk Pemasaran. USU e Repository.
- Dharwiyanti, Sri., 2003. Pengantar Unified Modeling Language (UML). www.ilmukomputer.com.
- Wahono, R.S., 2003. Pengantar Unified Modeling Language (UML). www.ilmukomputer.com.
- Nursari, S.R.C., 2012. Rekayasa Perangkat Lunak. Tersedia di: reezeki2011.wordpress.com.
- Binus., 2012. Tersedia di: repository.binus.ac.id [Diakses 12 Februari 2016].

Rubin, J., 2008. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.

Janti, S., 2014. Analisis Validitas Dan Reliabilitas Dengan Skala Likert Terhadap Pengembangan SI/TI Dalam Penentuan Pengambilan Keputusan PENERAPAN Strategic Planning Pada Industri Garmen.

Klikkanan., 2007. Tersedia di: www.klik-kanan.com [Diakses 15 Febuari 2016].

