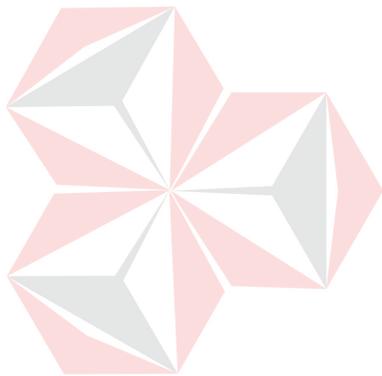


**RANCANG BANGUN APLIKASI SISTEM PEMESANAN TIKET
MENGUNAKAN WEB SERVICES**

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer



Oleh :

Nama : Gusti Ayu Putu Lili Santina

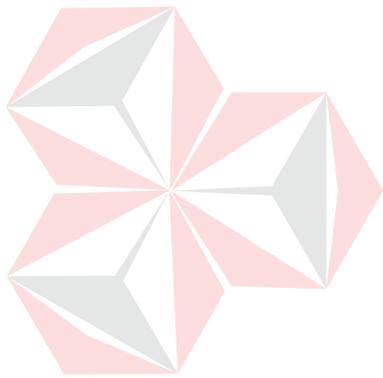
NIM : 01.41010.0002

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

2006



UNIVERSITAS
Dinamika

You won't know until you try

ABSTRAKSI

Aplikasi sistem pemesanan tiket merupakan suatu aplikasi yang digunakan *client* untuk memesan tiket dari biro perjalanan seperti tiket pesawat dan kapal laut yang dapat dilakukan melalui telepon seluler. Aplikasi client pada sistem pemesanan tiket ini dibuat menggunakan teknologi J2ME (Java 2 Micro Edition) yang didesain untuk pengguna peralatan nirkabel seperti telepon seluler.

Web service merupakan salah satu bentuk implementasi dari model arsitektur Multi-Tier yang menyediakan fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi sistem pemesanan tiket. XML (eXtensible Markup Language) merupakan format pertukaran data yang digunakan dalam web service.

Pada Tugas Akhir ini digunakan web service sebagai sarana penghubung antara client dengan biro perjalanan dan antara biro perjalanan dengan penyedia transportasi. Sistem ini memberikan kemudahan kepada pihak client untuk mengetahui jadwal dan *seat* yang tersedia pada masing-masing penyedia transportasi, serta melakukan pemesanan tiket darimana saja dan kapan saja mengingat sebagian besar masyarakat yang selalu *mobile* dan sudah terbiasa menggunakan telepon seluler.

KATA PENGANTAR

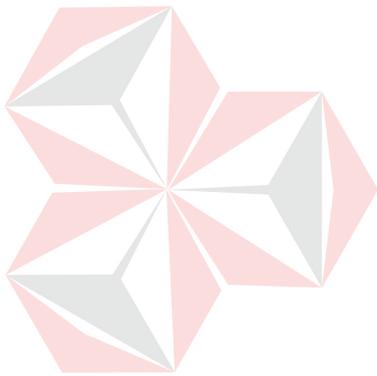
Puji syukur penulis panjatkan kehadirat Ida Sang Hyang Widhi Wasa, Tuhan Yang Maha Esa, karena berkat rahmatNya, penulis dapat menyelesaikan Tugas Akhir yang merupakan persyaratan dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM). Tugas akhir ini membahas tentang sistem pemesanan tiket menggunakan web services. Penyelesaian Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan kepada :

1. Bapak Budi Hermawan, S.Kom, sebagai dosen pembimbing II atas segala arahan dan bimbingannya.
2. Bapak Tutut Wuriyanto, M.Kom, sebagai dosen pembimbing I atas segala arahan dan bimbingannya.
3. Bapak Soetam Rizky, S.Kom atas segala arahan dan bimbingannya.
4. Mbak Lila dan Mbak Kris, selaku *contact person* pada PT. Gazali Inti International (GII) Tours and Travels, atas segala informasi, bimbingan, dan ketersediaan data.
5. Ajik, Ibu, Ita, Atu, Wayah, dan Ninik atas cinta, kasih sayang, dukungan moral, dan doanya.
6. Wi Lanang Arya, atas kesetiaan, cinta, kasih sayang, nasihat, dukungan, kepercayaan, doa, serta penantian yang cukup lama.
7. Ninok, Agung, Windi, Sujud, Tito, Erdwin, Kadek, April, Annisa, Puchan, Yuni '04, Ade '04 atas bantuan, dukungan, dan doanya.

8. Semua rekan-rekan yang tak mungkin dapat saya sebutkan satu persatu, atas segala motivasi, ide, dukungan serta doanya.

Semoga Ida Sang Hyang Widhi Wasa memberikan pahala yang setimpal kepada semua pihak yang telah membantu dalam pengerjaan Tugas Akhir ini.

Penulis menyadari masih banyak kekurangan dalam penulisan Tugas Akhir ini, dan mohon maaf apabila ada kata-kata yang kurang berkenan di hati. Semoga Tugas Akhir ini dapat memberikan manfaat bagi kita semua.



Surabaya, Agustus 2006

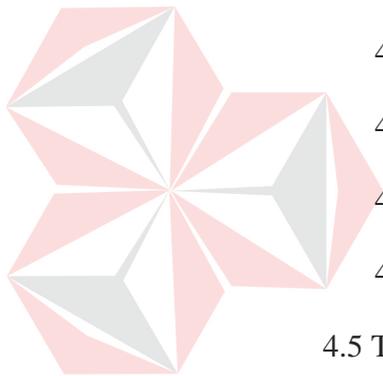
UNIVERSITAS
Dinamika
Penulis

DAFTAR ISI

	Halaman
ABSTRAKSI	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan	4
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Model Arsitektur N-Tier.....	6
2.2 Web Services	8
2.3 ASP.NET	17
2.4 Microsoft Access	21
2.5 Java 2 Micro Edition.....	22
2.5.1 Konfigurasi J2ME	23
2.5.2 J2ME Profiles.....	24
2.5.3 MIDlet.....	25
2.5.4 Antar Muka	27

	Halaman
2.5.5 Mobile Information Device Profile 2.0 (MIDP 2.0)	28
2.5.6 Generic Connection Framework	30
2.5.7 Record Management System (RMS)	31
2.5.8 Emulator J2ME	32
2.5.9 Komunikasi Data.....	34
2.6 Perancangan Sistem.....	36
2.6.1 Sistem Flowchart.....	37
2.6.2 Data Flow Diagram.....	38
2.6.3 Entity Relationship Diagram.....	40
BAB III PERANCANGAN SISTEM.....	41
3.1 Perancangan Model Arsitektur Multi-Tier	41
3.2 Perancangan Sistem.....	44
3.2.1 Pembuatan Sistem Flow.....	44
3.2.2 Pembuatan Diagram Berjenjang	47
3.2.3 Pembuatan DFD.....	47
3.2.4 Pembuatan ERD	52
3.2.5 Pembuatan Struktur Database	56
3.3 Perancangan Input Output	64
3.3.1 Rancangan Input Output pada Ponsel	64
3.3.2 Rancangan Input Output Web Untuk Administrator	70
BAB IV IMPLEMENTASI DAN EVALUASI	75
4.1 Kebutuhan Sistem.....	75
4.1.1 Kebutuhan perangkat keras	75

	Halaman
4.1.2 Kebutuhan perangkat lunak.....	76
4.2 Pembuatan Program	76
4.3 Tampilan Aplikasi Web Service.....	77
4.3.1 Web service penyedia transportasi.....	77
4.3.2 Web service biro perjalanan.....	79
4.4 Tampilan Aplikasi MIDlet pada Client	84
4.4.1 Tampilan pembuka.....	84
4.4.2 Tampilan menu utama.....	84
4.4.3 Tampilan pemesanan tiket.....	85
4.4.4 Tampilan konfirmasi	88
4.4.5 Tampilan pembatalan tiket.....	89
4.4.6 Tampilan lihat nomor booking.....	90
4.4.7 Tampilan about.....	91
4.5 Tampilan Aplikasi Web Administrator	91
4.5.1 Tampilan login	92
4.5.2 Tampilan web halaman utama	92
4.5.3 Tampilan web daftar pemesanan.....	93
4.5.4 Tampilan web update status	93
4.5.5 Tampilan web daftar harga	94
4.5.6 Tampilan web tambah data harga	95
4.5.7 Tampilan web daftar user.....	95
4.5.8 Tampilan web tambah data user.....	96



4.5.9 Tampilan web status seat.....	96
4.6 Tampilan Kesalahan	97
4.6.1 Tampilan kesalahan pada web service	97
4.6.2 Tampilan kesalahan pada MIDlet	100
4.7 Evaluasi Sistem.....	100
4.7.1 Analisa hasil uji coba pada aplikasi MIDlet	100
4.7.2 Analisa hasil uji coba pada aplikasi web administrator	102
4.7.3 Evaluasi kinerja sistem.....	103
4.7.4 Analisa hasil uji coba sistem yang digunakan.....	108
BAB V PENUTUP	109
5.1 Kesimpulan	109
5.2 Saran	110
DAFTAR PUSTAKA	111
LAMPIRAN.....	112



DAFTAR TABEL

	Halaman
Tabel 3.1	Tabel MasterKendaraan 56
Tabel 3.2	Tabel DetailKendaraan 57
Tabel 3.3	Tabel MasterTerminal 58
Tabel 3.4	Tabel MasterKelas 58
Tabel 3.5	Tabel MasterTrayek 59
Tabel 3.6	Tabel DetailTrayek 60
Tabel 3.7	Tabel MasterCustomer 60
Tabel 3.8	Tabel Pemesanan 61
Tabel 3.9	Tabel MasterPesan 62
Tabel 3.10	Tabel DaftarHarga 63
Tabel 3.11	Tabel Users 64
Tabel 4.1	Analisa hasil uji coba pada aplikasi MIDlet..... 101
Tabel 4.2	Analisa hasil uji coba pada aplikasi web Administrator 102
Tabel 4.3	Contoh permasalahan 104
Tabel 4.4	Tabel perhitungan waktu dengan kartu Simpati (dalam detik) 106
Tabel 4.5	Tabel perhitungan waktu dengan kartu XL Bebas (dalam detik) .. 106
Tabel 4.6	Tabel perhitungan waktu dengan kartu IM3 (dalam detik) 107
Tabel 4.7	Analisa hasil uji coba sistem yang digunakan 108

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model Arsitektur 3-Tier	6
Gambar 2.2 Model Aplikasi N-Tier	7
Gambar 2.3 Building block Web Services.....	10
Gambar 2.4 SOAP Messaging.....	13
Gambar 2.5 Contoh Web Service	18
Gambar 2.6 Fungsi HelloWorld().....	18
Gambar 2.7 Hasil dokumen XML fungsi HelloWorld()	19
Gambar 2.8 Fungsi Hitung()	19
Gambar 2.9 Hasil dokumen XML fungsi Hitung()	19
Gambar 2.10 Arsitektur ASP.NET	20
Gambar 2.11 Arsitektur tingkat tinggi dari sebuah aplikasi MIDP.....	25
Gambar 2.12 Siklus hidup dari sebuah MIDP	26
Gambar 2.13 Class Diagram dari user interface.....	27
Gambar 2.14 Hirarki java interface dalam GCF.....	31
Gambar 2.15 Emulator J2ME	33
Gambar 3.1 Model Arsitektur Multi-Tier	41
Gambar 3.2 Akses jaringan internet dari telepon seluler	43
Gambar 3.3 Hubungan Tiap-tiap Web Service Sistem Pemesanan Tiket	43
Gambar 3.4 Sistem Flow Pemesanan.....	45
Gambar 3.5 Sistem Flow Pembatalan	46
Gambar 3.6 Sistem Flow Konfirmasi Pemesanan	46
Gambar 3.7 Diagram Berjenjang.....	47

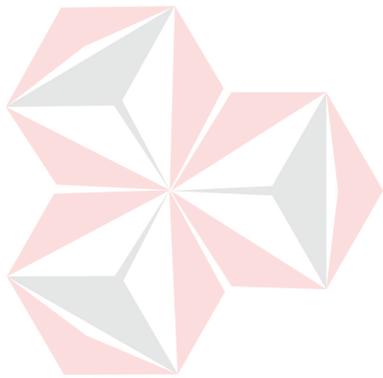
	Halaman
Gambar 3.8 DFD Level 0	48
Gambar 3.9 DFD Level 1.....	49
Gambar 3.10 DFD Level 2 Sub Proses Pemesanan Tiket	50
Gambar 3.11 DFD Level 2 Sub Proses Pembatalan Tiket.....	51
Gambar 3.12 DFD Level 2 Sub Proses Konfirmasi Tiket	52
Gambar 3.13 Conceptual Data Modelling (CDM) Penyedia Transportasi.....	53
Gambar 3.14 Physical Data Modelling (PDM) Penyedia Transportasi	54
Gambar 3.15 Conceptual Data Modelling (CDM) Biro Perjalanan.....	55
Gambar 3.16 Physical Data Modelling (PDM) Biro Perjalanan	55
Gambar 3.17 Rancangan Tampilan Pembuka	64
Gambar 3.18 Rancangan Tampilan Menu Utama	65
Gambar 3.19 Rancangan Tampilan Menu Pemesanan.....	65
Gambar 3.20 Rancangan Tampilan Jurusan	66
Gambar 3.21 Rancangan Tampilan Info Perjalanan	66
Gambar 3.22 Rancangan Tampilan Status Seat.....	67
Gambar 3.23 Rancangan Tampilan Entry Data Penumpang	67
Gambar 3.24 Rancangan Konfirmasi	68
Gambar 3.25 Rancangan Pembatalan	68
Gambar 3.26 Rancangan Pembatalan2	69
Gambar 3.27 Rancangan Lihat Nomor Booking	69
Gambar 3.28 Rancangan Detail Nomor Booking	70
Gambar 3.29 Rancangan Konfirmasi Tiket	70
Gambar 3.30 Rancangan Login	71

Gambar 3.31 Rancangan Web Halaman Utama	71
Gambar 3.32 Rancangan Web Daftar Pemesanan	72
Gambar 3.33 Rancangan Web Update Status	72
Gambar 3.34 Rancangan Web Daftar Harga	73
Gambar 3.35 Rancangan Web Tambah Data Harga	73
Gambar 3.36 Rancangan Web Daftar User.....	74
Gambar 3.37 Rancangan Web Tambah Data User.....	74
Gambar 4.1 Web service penyedia transportasi	77
Gambar 4.2 Web service biro perjalanan	79
Gambar 4.3 Contoh pemanggilan fungsi-fungsi pada web service	82
Gambar 4.4 Data yang dihasilkan dari pemanggilan fungsi web service1	82
Gambar 4.5 Data yang dihasilkan dari pemanggilan fungsi web service2	83
Gambar 4.6 Data yang dihasilkan dari pemanggilan fungsi web service3	83
Gambar 4.7 Tampilan pembuka	84
Gambar 4.8 Tampilan menu utama	84
Gambar 4.9 Tampilan awal tanggal dan kendaraan	85
Gambar 4.10 Tampilan untuk memilih tanggal berangkat	85
Gambar 4.11 Tampilan tujuan perjalanan	86
Gambar 4.12 Tampilan jadwal	86
Gambar 4.13 Tampilan status seat.....	87
Gambar 4.14 Tampilan entry data penumpang.....	87
Gambar 4.15 Tampilan informasi.....	88
Gambar 4.16 Tampilan konfirmasi1	88

	Halaman
Gambar 4.17 Tampilan konfirmasi2	89
Gambar 4.18 Tampilan pembatalan1	89
Gambar 4.19 Tampilan pembatalan2	90
Gambar 4.20 Tampilan lihat nomor booking	90
Gambar 4.21 Tampilan detail nomor booking.....	91
Gambar 4.22 Tampilan about	91
Gambar 4.23 Tampilan login	92
Gambar 4.24 Tampilan web halaman utama	92
Gambar 4.25 Tampilan web daftar pemesanan	93
Gambar 4.26 Tampilan web update status	94
Gambar 4.27 Tampilan web daftar harga	94
Gambar 4.28 Tampilan tambah data harga	95
Gambar 4.29 Tampilan web daftar user	95
Gambar 4.30 Tampilan tambah data user	96
Gambar 4.31 Tampilan web status seat	97
Gambar 4.32 Tampilan kesalahan jika inputan salah1	98
Gambar 4.33 Tampilan kesalahan jika inputan salah2.....	98
Gambar 4.34 Tampilan kesalahan jika tanpa inputan1	99
Gambar 4.35 Tampilan kesalahan jika tanpa inputan2	99
Gambar 4.36 Tampilan kesalahan jika tanggal berangkat lebih dari 1 bulan	100
Gambar 4.37 Tampilan kesalahan jika tanggal pemesanan kurang dari 3 hari sebelum tanggal berangkat	100

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Listing Program	112
Lampiran 2. Biodata Penulis	144



UNIVERSITAS
Dinamika

BAB I

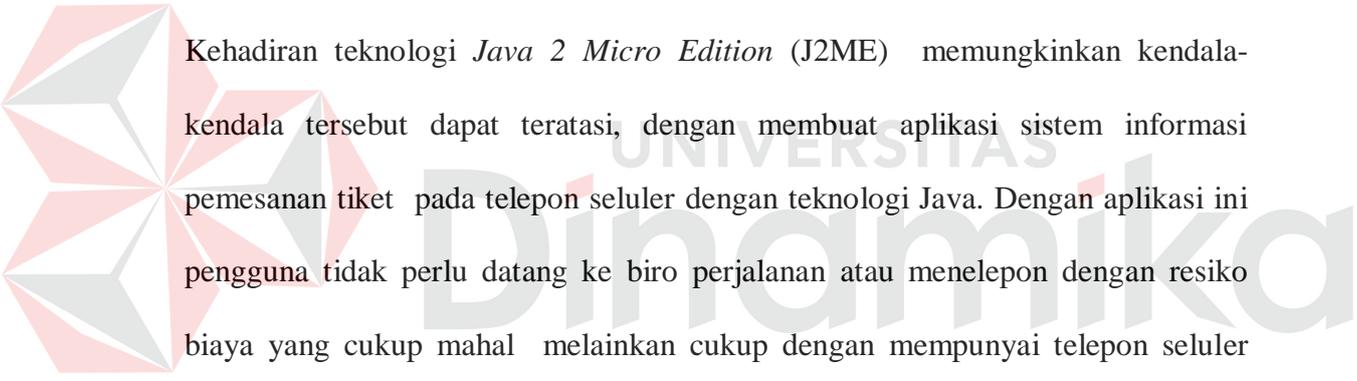
PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi informasi yang semakin canggih menuntut orang untuk mengembangkan sesuatu menjadi praktis dan efisien. Pada awalnya jika dua buah perusahaan ingin saling menukar informasi dilakukan melalui proses yang cukup rumit. Sebagai contoh proses pertukaran data yang terjadi antara biro perjalanan dengan perusahaan penyedia transportasi. Antara biro perjalanan dengan perusahaan penyedia transportasi tentu saja mempunyai *platform* yang berbeda karena dibangun oleh *developer* yang berbeda sehingga untuk melakukan pertukaran informasi sangat rumit. Saat ini muncul teknologi XML dan *web service*. Web service didesain untuk mendayagunakan jaringan global yang ada saat ini yang dikenal dengan *internet*. XML merupakan suatu format dokumen berbasis teks yang terdiri dari sejumlah tag seperti halnya HTML. Tag XML menjelaskan mengenai informasi dan struktur suatu dokumen. Aplikasi yang mampu mengartikan tag-tag yang ada sesuai dengan struktur dokumennya, akan mampu mendapatkan informasi yang terkandung dalam suatu dokumen XML. Dengan menggunakan format dokumen XML, memungkinkan suatu aplikasi dapat berkomunikasi dengan aplikasi lain yang berbeda platform. XML web service bersifat *independent* sebagai sarana pertukaran data, artinya tidak tergantung pada sistem operasi dan *device* yang digunakan.

Pada awalnya untuk memesan tiket, orang tersebut harus datang ke biro perjalanan, menelepon atau datang langsung ke tempat pembelian tiket. Biaya

untuk datang atau menelepon tentu saja mahal dan tidak dapat dilakukan oleh orang yang selalu aktif, tidak selalu berada di suatu tempat, atau orang-orang eksekutif yang tidak punya waktu untuk repot dengan mendatangi biro perjalanan. Sedangkan untuk memesan tiket melalui web tidak semua orang dapat melakukannya karena tidak semua orang mempunyai komputer atau terhubung ke internet dan tentu saja hal tersebut juga membutuhkan biaya yang tidak sedikit dan kurang praktis. Di jaman sekarang dimana orang-orang beranggapan bahwa memiliki telepon seluler telah menjadi suatu kebutuhan yang cukup penting, dituntutlah suatu teknologi yang dapat membantu orang-orang untuk dapat melakukan sesuatu yang praktis dan dapat dilakukan kapan saja dan dimana saja.



Kehadiran teknologi *Java 2 Micro Edition* (J2ME) memungkinkan kendala-kendala tersebut dapat teratasi, dengan membuat aplikasi sistem informasi pemesanan tiket pada telepon seluler dengan teknologi Java. Dengan aplikasi ini pengguna tidak perlu datang ke biro perjalanan atau menelepon dengan resiko biaya yang cukup mahal melainkan cukup dengan mempunyai telepon seluler yang mempunyai fasilitas *General Packet Radio Service* (GPRS) untuk terhubung ke web service biro perjalanan untuk mendapatkan informasi tentang perjalanan apa saja yang ada dan selanjutnya dapat memesan tiket yang diinginkan. Aplikasi sistem pemesanan tiket ini memungkinkan web service biro perjalanan dapat terhubung dengan web service penyedia transportasi, dan aplikasi telepon seluler dapat terhubung dengan web service biro perjalanan untuk mendapatkan informasi perjalanan dengan format data berupa XML. Dengan demikian XML memungkinkan web services yang berjalan di platform Windows dapat berkomunikasi dengan MIDlet yang berjalan di platform Symbian.

1.2 Perumusan Masalah

Dari uraian latar belakang, perumusan permasalahan dalam Tugas Akhir ini yaitu bagaimana membuat rancang bangun sistem pemesanan tiket menggunakan web service.

1.3 Pembatasan Masalah

Dalam pembuatan Tugas Akhir ini, permasalahan akan dibatasi pada hal-hal sebagai berikut:

1. Sistem dapat memberikan informasi jadwal pesawat terbang dan kapal laut dengan mengakses fungsi-fungsi pada web service.
2. Sistem ini hanya dapat melakukan proses pemesanan, konfirmasi pemesanan, dan pembatalan saja.
3. Sistem dapat menyimpan nomor booking ke dalam *Record Management System* (RMS) pada telepon seluler.
4. Hubungan yang terjadi antara web service biro perjalanan dengan web service perusahaan penyedia transportasi merupakan simulasi yang digunakan sebagai cara pertukaran data antar perusahaan.
5. Rancangan database beserta data-data yang ada pada tiap penyedia transportasi merupakan data simulasi, bukan data sebenarnya yang digunakan sebagai sumber penyedia data untuk aplikasi pemesanan tiket.
6. Sistem ini mengabaikan masalah pembayaran dan sekuritas (keamanan) pada transmisi data.
7. Sistem ini menggunakan teknologi web services, XML, dan J2ME.

1.4 Tujuan

Dibawah ini merupakan tujuan yang diharapkan dalam pembuatan Tugas Akhir ini:

1. Membuat dan menerapkan aplikasi web service sebagai sistem yang memungkinkan proses pertukaran informasi antara perusahaan penyedia transportasi, biro perjalanan, dan *client* pada sistem pemesanan tiket.
2. Menerapkan teknologi web service, XML, dan J2ME *mobile programming*.

1.5 Sistematika Penulisan

Sistematika yang digunakan dalam penulisan Tugas Akhir ini dibagi menjadi beberapa Bab dan Sub-Bab. Adapun pembagian Bab ini sebagai berikut:

BAB I : PENDAHULUAN

Bab ini mengulas beberapa masalah yang meliputi: latar belakang yang menjadi dasar pertimbangan timbulnya permasalahan, permasalahan yang disoroti, pembatasan masalah, tujuan pembuatan Tugas Akhir ini, dan dilanjutkan dengan Sistematika penulisan Tugas Akhir.

BAB II : LANDASAN TEORI

Bab ini memberikan uraian tentang teori yang digunakan dalam penyusunan tugas akhir. Menjelaskan Sistem Informasi yang digunakan serta beberapa teori yang berkaitan dengan sistem yang akan dirancang dalam penyusunan Tugas Akhir ini.

BAB III : METODE PENELITIAN DAN PERANCANGAN SISTEM

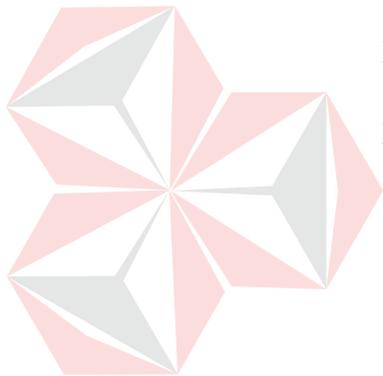
Bab ini membahas tentang pengidentifikasian masalah, perumusan masalah, batasan masalah, tujuan, studi lapangan dan pustaka, pengujian data lapangan, pembuatan model, evaluasi, dan kesimpulan dan saran.

BAB IV : IMPLEMENTASI DAN EVALUASI

Bab ini membahas tentang langkah-langkah dalam melakukan implementasi pada aplikasi ini, serta bagaimana penggunaannya. Selanjutnya melakukan evaluasi dari hasil implementasi tersebut.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dan saran dari perancangan dan pembuatan proyek ini.



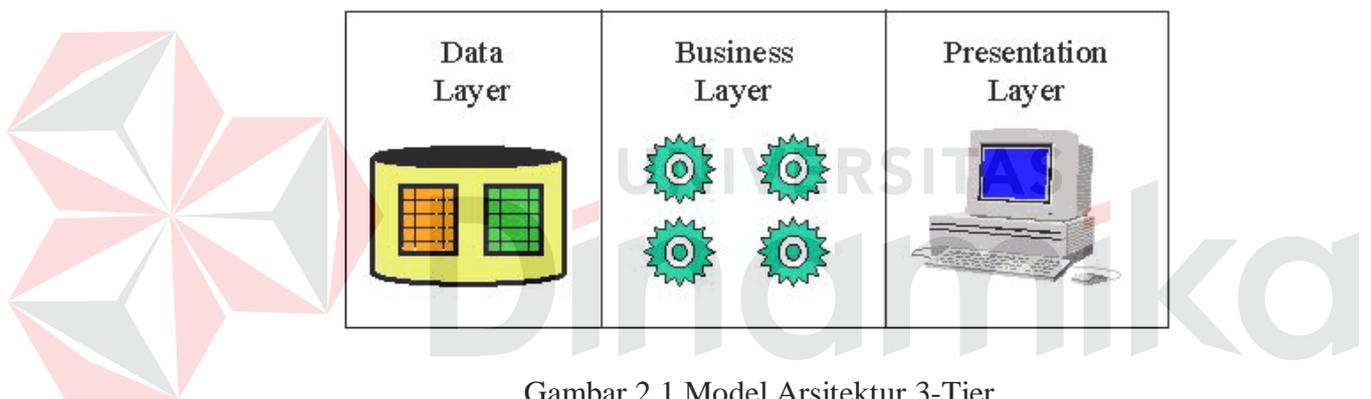
UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 Model Arsitektur N-Tier

Arsitektur N-Tier didasari oleh model arsitektur 3-Tier. Model 3-Tier ini memisahkan antara *Presentation Layer*, *Business Layer*, dan *Data Layer* dalam bagian yang berbeda. Dari Gambar 2.1 dapat kita lihat bagaimana komposisi dari arsitektur dengan model 3-Tier.



Gambar 2.1 Model Arsitektur 3-Tier

Berikut ini adalah penjelasan untuk setiap layer :

1. Presentation Layer

Presentation Layer merupakan bagian dari aplikasi yang akan berinteraksi dengan user. Layer ini menampilkan semua data yang diperlukan. Layer ini juga menerima input dan modifikasi user terhadap data.

2. Business Layer

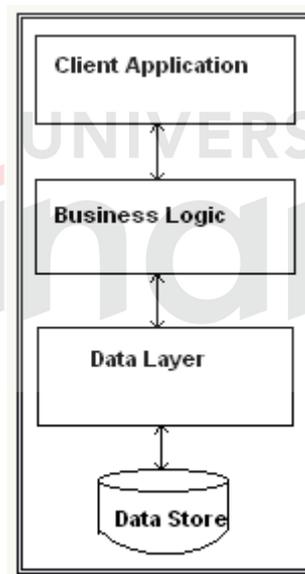
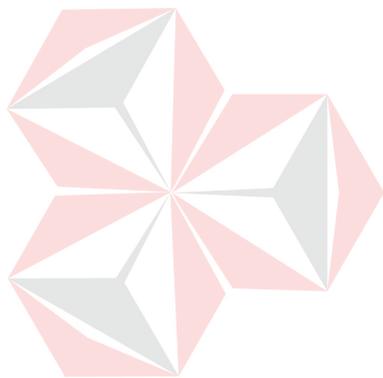
Business Layer adalah salah satu bagian yang ada dalam aplikasi client/server. Layer ini mengontrol semua data yang diakses dan meng-update data yang ada

dalam database. Layer ini biasanya dapat digunakan oleh modul-modul yang ada dalam aplikasi.

3. Data Layer

Data Layer merupakan tempat dimana aplikasi selesai melakukan proses manipulasi data. Layer ini bertanggung jawab terhadap penyimpanan data dan menyediakan data yang akan diberikan ke layer user interface.

Istilah model N-Tier sendiri muncul disebabkan karena suatu arsitektur aplikasi terdiri atas banyak tier di dalamnya seperti terlihat pada Gambar 2.2 berikut ini.



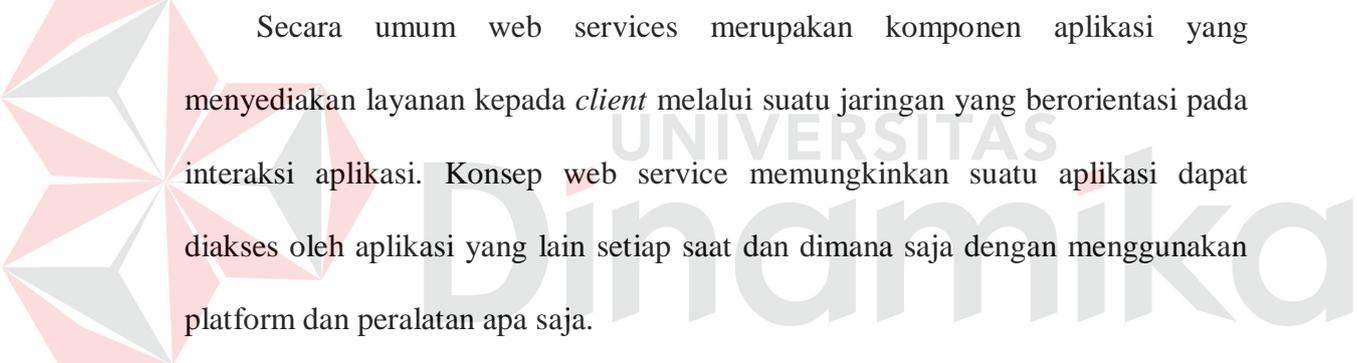
Gambar 2.2 Model Arsitektur N-Tier

Perbedaan yang jelas terlihat antara model arsitektur 3-Tier dengan N-Tier adalah pada bagian Data Layer. Pada arsitektur 3-Tier Data Layer berada dalam satu bagian dengan Data Store (tempat penyimpanan data). Sedangkan pada arsitektur N-Tier Data Layer dengan Data Store berada pada tempat yang terpisah

karena terdapat suatu proses lain sebelum data disimpan atau sebelum data dikirimkan ke Business Layer.

2.2 Web Services

Web services merupakan *programmable application logic* yang dapat diakses oleh program melalui protokol web standar dan tidak bergantung pada platform yang digunakan (Basiura, 2001). Menurut Galagher (2001) mengatakan web services merupakan cara yang sudah distandarkan untuk membagikan atau mendistribusikan fungsi-fungsi dari aplikasi melalui jaringan dengan menggunakan protokol web standar, seperti HTTP dan XML.



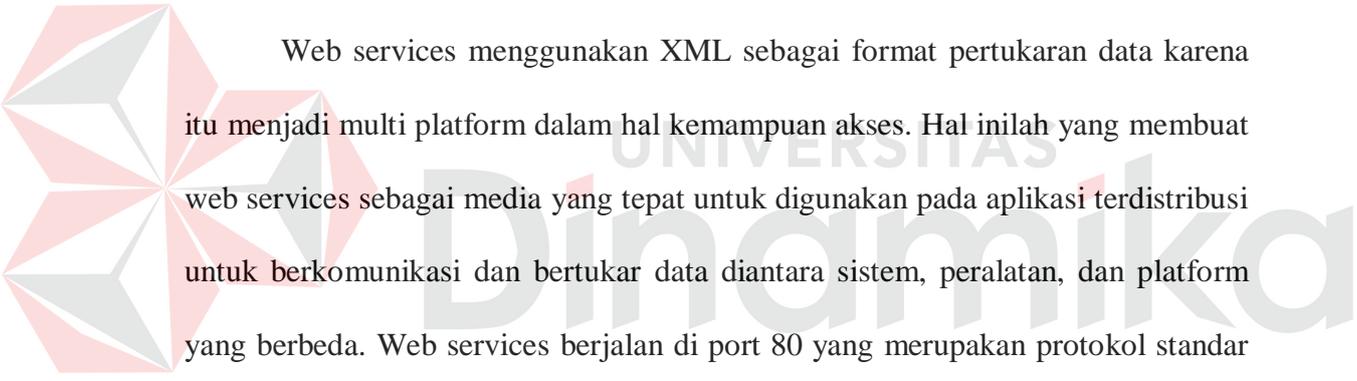
Secara umum web services merupakan komponen aplikasi yang menyediakan layanan kepada *client* melalui suatu jaringan yang berorientasi pada interaksi aplikasi. Konsep web service memungkinkan suatu aplikasi dapat diakses oleh aplikasi yang lain setiap saat dan dimana saja dengan menggunakan platform dan peralatan apa saja.

Web services merupakan perkembangan *distributed computing* dengan arsitektur N-Tier. Keuntungan yang paling mendasar yang ditawarkan oleh konsep web services adalah integrasi. Usaha untuk mengintegrasikan aplikasi, sistem, maupun platform yang berbeda sering mengalami kesulitan dan memerlukan proses yang panjang. Web services merupakan standar yang tepat sebagai alat pengintegrasian. Web services mampu mengintegrasikan aplikasi dan sistem dari platform yang berbeda karena menggunakan standar protokol web dalam interaksinya seperti TCP/IP, HTTP, XML, SOAP, UDDI.

Web services merupakan salah satu bentuk implementasi dari arsitektur model aplikasi N-Tier. Perbedaan web services dengan pendekatan N-Tier lainnya

adalah dari segi infrastruktur dan dokumen yang digunakan sebagai format pertukaran data.

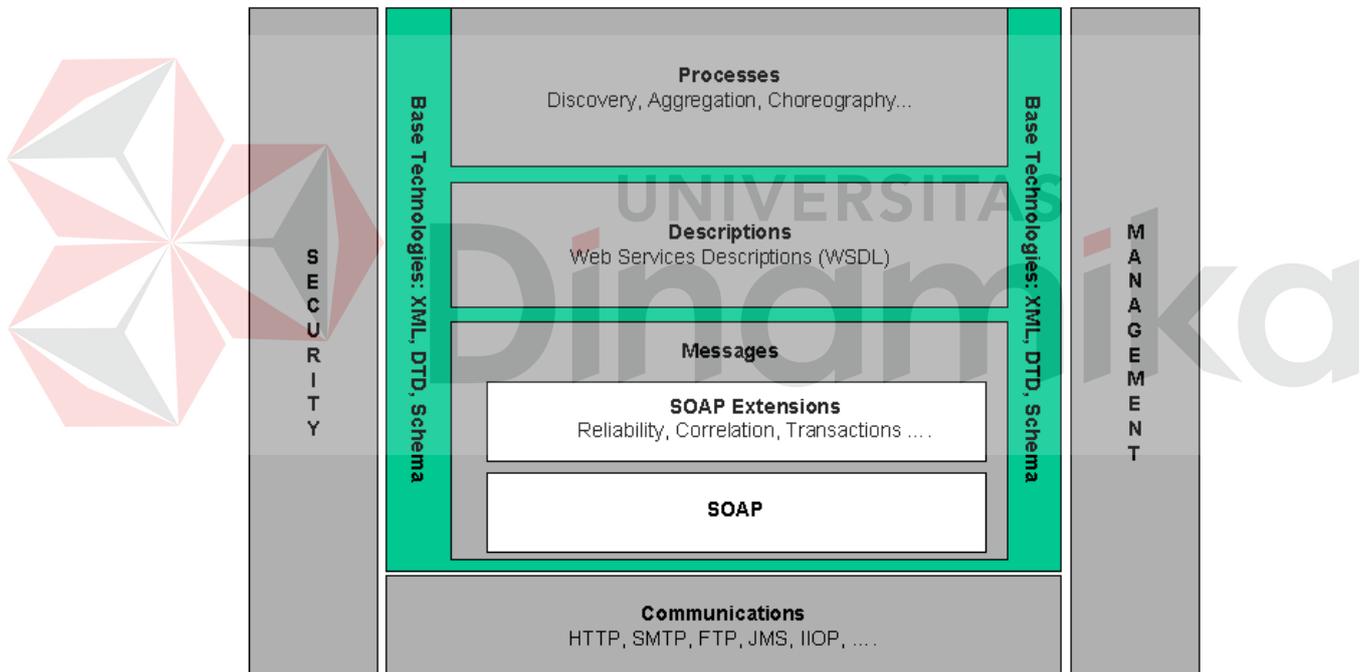
Selain itu web services bersifat platform independent. Web services dapat diimplementasikan di berbagai platform menggunakan bahasa pemrograman apapun. Web services tidak tergantung pada bahasa pemrograman ataupun sistem operasi yang digunakan. Hal ini memungkinkan program yang ditulis dalam bahasa yang berbeda dan pada platform yang berbeda untuk dapat saling berkomunikasi dengan cara yang standar. Web services yang dibuat dalam platform Microsoft dapat digunakan oleh aplikasi yang berada dalam platform lainnya.



Web services menggunakan XML sebagai format pertukaran data karena itu menjadi multi platform dalam hal kemampuan akses. Hal inilah yang membuat web services sebagai media yang tepat untuk digunakan pada aplikasi terdistribusi untuk berkomunikasi dan bertukar data diantara sistem, peralatan, dan platform yang berbeda. Web services berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian mengurangi resiko terblokir oleh *firewall*.

Dalam implementasinya web services tidak mempunyai tampilan yang bagus, karena web services termasuk dalam Business Layer. Artinya di dalam web services hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh suatu aplikasi. Walaupun tidak memiliki tampilan yang bagus, interfacenya dapat terus dikembangkan. Syarat untuk dapat memanfaatkan layanan yang disediakan oleh web services adalah kemampuan untuk berkomunikasi melalui TCP/IP dan kemampuan untuk memproses XML.

Walaupun web services bersifat independent, namun suatu web services dapat berhubungan dengan web services lainnya membentuk suatu kolaborasi untuk melakukan tugas tertentu. Beberapa vendor mulai berkolaborasi satu sama lain dengan konsep web services, diantaranya : IBM, Microsoft, Sun , Oracle. Contoh web services yang sudah jadi dan dipakai adalah web services keluaran Microsoft (Microsoft Passport) yaitu web services untuk username dan password yang sudah dipasang di web site Microsoft dan Hotmail. *Building block* web service dapat dilihat pada gambar berikut.



Gambar 2.3 Building block Web Services

1. *Universal Description, Discovery, and Integration (UDDI)*

Web services yang telah *publish* di internet disimpan dalam sebuah direktori. Proses mencari dan mendaftarkan web services didefinisikan dalam standar UDDI. Dengan UDDI kita bisa mencari web services berdasarkan

keyword untuk kategori tertentu. UDDI menyediakan registrasi web services dan menyediakan direktori yang berisi perusahaan-perusahaan yang dapat dicari berdasarkan nama perusahaan, di bidang apa perusahaan tersebut bergerak, dan di wilayah mana perusahaan tersebut menyediakan service. Setiap *record* dari perusahaan memiliki detail tentang tujuan dan lokasi web services yang dibuat.

UDDI bertujuan untuk menjembatani antara perusahaan yang mengembangkan dan mempublish web services dengan programmer yang berusaha untuk mencari dan menggunakan web services. Tanpa UDDI, interaksi dan penggunaan web services menuntut programmer untuk mengetahui secara jelas keberadaan service yang disediakan, tujuan, dan lokasi melalui mekanisme lain.

Saat ini terdapat tiga registrasi UDDI yang telah dipublikasikan, yaitu milik IBM (<http://www.ibm.com/services/uddi/>), SAP (<http://uddi.sap.com/>), dan Microsoft (<http://uddi.microsoft.com>). Informasi mengenai standar dan implementasi UDDI dapat dilihat di <http://www.uddi.org>.

2. *Web Services Definition Language* (WSDL)

Setelah web services ditemukan menggunakan UDDI, client yang akan menggunakan service yang disediakan web services perlu mengetahui cara untuk dapat berinteraksi dengan web services tersebut dan service apa saja yang disediakan. Fungsionalitas service yang disediakan oleh web services dipaparkan dalam bentuk *programmatic interface* menggunakan standar description language WSDL. WSDL merupakan suatu dokumen XML yang menjelaskan method-method apa saja yang tersedia dalam suatu web

services, parameter-parameter apa saja yang diperlukan untuk memanggil suatu method, dan apa hasil dari method yang akan dipanggil. WSDL menjelaskan bagaimana cara berinteraksi dengan web services dan mengandung informasi mengenai *message* yang dapat dikirim ke SOAP message dan jenis response yang akan dikembalikan oleh web services.

WSDL berisi detail bagi programmer dalam membuat client application untuk dapat menggunakan fungsionalitas yang disediakan oleh web services. Dokumen WSDL juga akan menunjukkan pada client bagaimana web services akan bertingkah laku. Notasi yang digunakan WSDL untuk menggambarkan format message yang digunakan berdasarkan pada standar XML Schema. WSDL juga menggambarkan protokol komunikasi apa saja yang digunakan untuk berinteraksi dengan service. WSDL menggambarkan segala sesuatu yang dibutuhkan untuk menulis program yang dapat bekerja dengan web services. Struktur dasar dari WSDL adalah :

```
<definitions> ...
  <types> ...
  <message name="CalculateSoapIn"> ...
  <portType name="Service1Soap"> ...
  <binding name="Service1Soap"> ...
  <service name="Service1"> ...
</definitions>
```

a. *Types*

Definisi tipe data yang *machine* dan *language independent*.

b. *Messages*

Berisi parameter fungsi (dipisahkan antara input dan output) dan deskripsi dokumen.

c. *Port Types*

Mengacu pada definisi message pada bagian message untuk mendeskripsikan signature dari fungsi (nama operasi, input parameter, output parameter).

d. *Bindings*

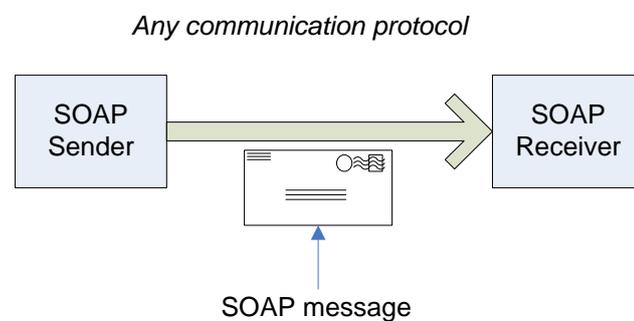
Menspesifikasikan informasi protocol untuk setiap operasi pada bagian port types.

e. *Services*

Menspesifikasikan informasi alamat port dari setiap binding pada sebuah services.

3. *Simple Object Access Protocol (SOAP)*

SOAP merupakan protokol untuk pertukaran informasi terstruktur pada sistem yang terdistribusi/terdesentralisasi. SOAP menggunakan teknologi XML untuk mendefinisikan kerangka struktur messagenya. SOAP didesain untuk bersifat *extensible* dan *simple*. Extensible artinya user mampu mengembangkan kemampuannya untuk kepentingan di masa mendatang.



Gambar 2.4 SOAP Messaging

Dalam konsep web services, SOAP merupakan mekanisme yang digunakan untuk mengirimkan message diantara client dengan web services. Message dalam hal ini berupa permintaan dari client yang menginstruksikan web services untuk melakukan pekerjaan tertentu dan dapat pula berupa respon dari service yang merupakan hasilnya. SOAP tidak bergantung pada model pemrograman ataupun sistem operasi. Hal ini berarti sebuah aplikasi client yang ditulis dalam bahasa pemrograman Java dapat mengerti message yang dikirim oleh web services yang ditulis dengan menggunakan C# atau Basic dengan editor Microsoft Visual Studio. Spesifikasi SOAP yang dipergunakan adalah yang dikeluarkan oleh *World Wide Web Consortium* (W3C) yaitu sebagai berikut :

a. SOAP 1.1

Nama namespace : <http://schemas.xmlsoap.org/soap/envelope/>

b. SOAP 1.2

Nama namespace : <http://www.w3c.org/2002/12/soap-envelope>

Bagian inti dari spesifikasi soap adalah kerangka message. Kerangka message SOAP terdiri atas elemen XML berupa : Envelope, Header, Body, dan Fault yang diambil dari namespace SOAP.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <hitung xmlns="http://tempuri.org/WebService1/Service1">
    <i>int</i>
  </hitung>
</soap:Body>
</soap:Envelope>
```

4. *Hypertext Transfer Protocol (HTTP)*

HTTP mendefinisikan bagaimana suatu pesan disusun dan dikirimkan serta bagaimana web server menanganinya. HTTP merupakan serangkaian aturan untuk pertukaran file (teks, grafik, gambar, suara, video, dan file multimedia lainnya) di dalam www. Berhubungan dengan protocol TCP/IP yang merupakan protokol utama dalam pertukaran informasi di internet, HTTP adalah protokol aplikasi.

Konsep dasar yang merupakan bagian dari HTTP termasuk ide bahwa di dalam file bisa terdapat referensi ke file lain dimana pemilihan file tersebut akan memicu terjadinya transfer request. Beberapa web server dalam memenuhi layanan akan HTML dan file-file lainnya yang dapat dilayaninya, menambahkan sebuah HTTP Daemon, yaitu : program yang disesain untuk menunggu HTTP request dan melayani request bila mana terjadi. HTTP request ini berasal dari HTTP client seperti web browser.

5. *eXtensible Markup Language (XML)*

Menurut vendor-vendor dalam industri software, XML merupakan langkah berikut dari evolusi dalam dunia web. XML digunakan untuk representasi digital dari dokumen-dokumen. (Charles F. GoldBard, Paul Prescord : p24).

XML merupakan sebuah standar W3C yang mengijinkan data untuk bersifat *portable* dan bebas mendeskripskan dirinya, sehingga dapat dipertukarkan dengan mudah antar aplikasi maupun devices pada berbagai platform. Jadi XML merupakan ide untuk membuat format data yang universal dan standar. XML adalah penyederhanaan subset SGML (Standard

Generalized Markup Language) yang memungkinkan SGML generik dapat dilayani, diterima dan diproses pada web dengan cara yang sekarang dipakai untuk HTML. XML dirancang untuk kemudahan implementasi dan untuk interoperabilitas dengan SGML dan HTML. Aplikasi XML biasanya didefinisikan dengan membuat sebuah *document type definition* (DTD), yang merupakan komponen optional pada dokumen XML.

XML digunakan untuk menjelaskan suatu data yang bersifat platform independent. XML merupakan suatu format dokumen yang berbasis teks. Dengan menggunakan format dokumen XML, web services memungkinkan suatu aplikasi "berbicara" dengan aplikasi lainnya. Web services dapat diimplementasikan dalam berbagai platform menggunakan bahasa pemrograman apapun yang mendukung web services, dan bisa digunakan oleh berbagai aplikasi di berbagai platform.

Seperti halnya HTML, XML juga merupakan markup language yang sama-sama digunakan untuk melakukan marking up data. Hanya saja yang membedakannya XML dapat menyimpan informasi yang terstruktur dari suatu dokumen dan dapat digunakan oleh aplikasi lain karena formatnya yang standar. Selain berfungsi sebagai markup language yaitu mekanisme yang digunakan untuk menjelaskan struktur dalam dokumen, menurut W3C XML didesain untuk mendeskripsikan data dan berfokus pada data, sementara itu HTML didesain untuk menampilkan data dan terfokus pada tampilan data.

XML juga menggunakan elemen yang ditandai dengan tag pembuka (<...>), tag penutup (</...>), dan atribut elemen (parameter yang dinyatakan

dalam tag pembuka). Perbedaannya dengan HTML, HTML mendefinisikan dari awal tag dan atribut yang dipakai di dalamnya, sedangkan XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Tag dalam XML menjelaskan mengenai informasi dan struktur dokumen. Aplikasi yang mampu mengartikan tag-tag yang ada sesuai dengan struktur dokumennya, akan mampu mendapatkan informasi yang terkandung dalam suatu dokumen XML.

2.3 ASP.NET

.NET platform merupakan suatu set kumpulan teknologi yang memungkinkan teknologi internet ditransformasikan ke dalam platform *distributed computing* dengan skalabilitas dan kompatibilitas tinggi (Amri, 2003).

ASP.NET merupakan suatu aplikasi berbasis web yang merupakan salah satu komponen utama di dalam Microsoft .NET Framework. ASP.NET adalah salah satu bahasa pemrograman yang bertujuan untuk membangun suatu aplikasi web.

ASP.NET merupakan penerus dari teknologi *Active Server Pages* (ASP). Namun ASP.NET tidaklah hanya versi perbaikan biasa dari ASP tetapi sudah merupakan inovasi teknologi baru yang memiliki banyak perbedaan dan keunggulan dari ASP. ASP.NET merupakan platform pembuatan aplikasi web yang menyatu dengan *.NET Framework* serta menyediakan fasilitas-fasilitas bagi developer untuk membangun aplikasi web. Salah satu fasilitas yang disediakan oleh ASP.NET adalah dukungan dalam pembuatan web services. File yang dihasilkan dalam pembuatan web service berekstensi *.asmx*. Contoh penggunaan web service menggunakan ASP.NET sebagai berikut :

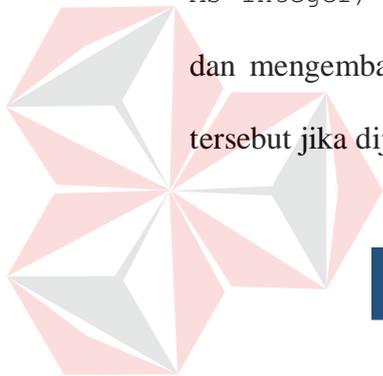
```

<WebMethod()> _
  Public Function HelloWorld() As String
    Return "Hello World"
  End Function

<WebMethod(Description:="Fungsi menggunakan parameter")> _
  Public Function hitung(ByVal i As Integer) As Integer
    hitung = i * 1000
  End Function

```

Web method merupakan suatu method yang dapat dijalankan di dalam web service yang telah dibuat. Pada contoh diatas, <WebMethod()> menunjukkan bahwa fungsi-fungsi yang terdapat di dalamnya akan dapat diakses oleh publik sebagai web method. Fungsi `Public Function HelloWorld()` mengembalikan nilai dengan tipe data *string*, sedangkan fungsi `Public Function hitung(ByVal i As Integer) As Integer` mendapatkan inputan atau parameter berupa *integer* dan mengembalikan nilai berupa *integer*. Berikut adalah tampilan fungsi-fungsi tersebut jika dijalankan :



Service1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [hitung](#)
Fungsi menggunakan nilai
- [HelloWorld](#)

Gambar 2.5 Contoh Web Service

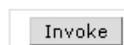
Service1

Click [here](#) for a complete list of operations.

HelloWorld

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.



Gambar 2.6 Fungsi HelloWorld()

```
<?xml version="1.0" encoding="utf-8" ?>
<string
xmlns="http://tempuri.org/WebService1/Service1">Hello
World</string>
```

Gambar 2.7 Hasil Dokumen XML Fungsi HelloWorld()

Service1

Click [here](#) for a complete list of operations.

hitung

Fungsi menggunakan nilai

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
i:	<input type="text" value="5"/>

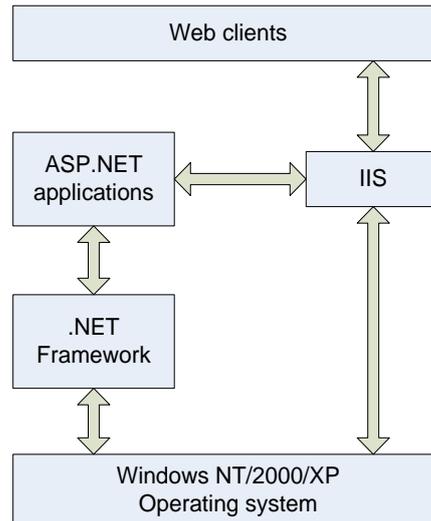
Gambar 2.8 Fungsi Hitung()

```
<?xml version="1.0" encoding="utf-8" ?>
<int
xmlns="http://tempuri.org/WebService1/Service1">5000</int>
```

Gambar 2.9 Hasil Dokumen XML Fungsi Hitung()

Fasilitas lain yang disediakan ASP.NET adalah *Web Form*. Web form memberikan kemudahan kepada pengguna dalam membuat halaman web berbasis form yang canggih dan dinamis. File yang dihasilkan dalam pembuatan web form berekstensi *.aspx*.

Untuk dapat menjalankan ASP.NET, selain dibutuhkan program ASP.NET, .NET Framework, juga dibutuhkan Web Server (IIS). Agar lebih bagaimana hubungannya, di bawah ini diberikan gambar arsitektur ASP.NET :



Gambar 2.10 Arsitektur ASP.NET

Gambar arsitektur ASP.NET diatas memiliki bagian-bagian yang terdiri dari:

1. Web Clients : client yang meminta request ke server. Pada umumnya software yang digunakan adalah web browser seperti *Internet Explorer (IE)*, *Netscape*, dan lain-lain.
2. Aplikasi ASP.NET : aplikasi yang dibuat dengan menggunakan ASP.NET.
3. IIS : web server yang terinstal pada server tempat aplikasi berada.
4. .NET Framework : model platform yang memudahkan dalam membangun lingkungan aplikasi yang terdistribusi di internet. Platform .NET Framework terdiri atas dua bagian utama yaitu :

a. *Common Language Runtime (CLR)*

CLR berfungsi untuk mengatur memori, mengatur thread, menjalankan kode, verifikasi keamanan kode, kompilasi, dan sistem service yang lain.

b. *.NET Framework Class Library (FCL)*

FCL berisi koleksi class yang dalam penggunaannya dapat diturunkan menjadi koleksi objek. Objek ini dapat digunakan untuk membangun aplikasi seperti menggunakan web forms atau XML web services.

5. Operating system : sistem operasi yang mendukung adalah Windows NT/2000/XP.

2.4 Microsoft Access

Merupakan salah satu program pengolah database yang cukup canggih, yang digunakan untuk mengolah beberapa jenis data dengan cara kerja pengoperasian yang cukup mudah. Banyak kemudahan yang akan kita peroleh jika bekerja dengan menggunakan Microsoft Access. Diantaranya kita dapat melakukan proses penyortiran, pengaturan data, pembuatan label data serta pembuatan laporan kegiatan sehari-hari

Access merupakan perangkat lunak dari Microsoft yang dapat digunakan sebagai tool untuk membuat database yang berisi tabel yang diperlukan oleh aplikasi yang nantinya menentukan rancangan antar muka untuk aplikasi tersebut. Dengan menggunakan Access, pembuatan database ini menjadi sangat mudah karena Access mudah dimengerti dan tidak terlalu rumit dalam proses pembuatannya.

Program database memungkinkan untuk bekerja dengan beberapa tabel. Dalam proses kerjanya, pengoperasian data pada tabel dalam database didukung oleh enam objek database, yaitu :

1. *Query* adalah sebuah objek database yang digunakan untuk menampilkan, menyunting dan menganalisis suatu data dengan cara lain.

2. *Form* adalah sebuah objek database yang digunakan untuk membuat kontrol-kontrol proses memasukkan, memeriksa dan memperbaiki data.
3. *Report* adalah sebuah objek yang digunakan untuk menampilkan data dengan format yang rendah.
4. *Pages* adalah sebuah objek khusus yang digunakan untuk menampilkan dan bekerja dengan data yang diambil dari internet atau intranet.
5. *Macro* adalah rangkaian dari beberapa perintah yang dapat disimpan dan dijalankan ulang secara otomatis.
6. *Module* adalah program-program yang ditulis *Access Basic*.

2.5 Java 2 Micro Edition (J2ME)

Java 2 Micro Edition (J2ME) merupakan subset dari Java 2 Standard Edition (J2SE) yang ditujukan untuk implementasi pada micro devices dan handheld yang tidak mampu mendukung secara penuh implementasi menggunakan J2SE.

Embedded system adalah produk-produk dengan komputer kecil berada di dalamnya, namun aplikasi yang bisa dimanfaatkan dari peralatan tersebut sangatlah spesifik. Contoh embedded system yang ada misalnya adalah aplikasi-aplikasi yang memanfaatkan microprocessor seperti televisi, sistem keamanan gedung, dan sebagainya.

J2ME sangat berguna untuk membangun sebuah aplikasi pada peralatan dengan jumlah memori dan kapasitas penyimpanan yang terbatas, serta kemampuan user interface yang terbatas seperti pada perangkat komunikasi bergerak berupa telepon seluler, PDA, dan sebagainya.

Seperti aplikasi umumnya yang menggunakan *Java Virtual Machine* (JVM), dalam J2ME digunakan pula virtual machine yang disebut *Kilobyte Virtual*

Machine (KVM). KVM adalah virtual machine yang sangat kecil dalam kebutuhan memorinya. Huruf K dalam KVM adalah singkatan dari *Kilobyte*, untuk menggambarkan betapa virtual machine ini bekerja pada total memori yang sedemikian kecil mulai dari 128 kilobyte hingga maksimal 512 kilobyte. Saat ini terdapat dua jenis aplikasi dari J2ME, yaitu:

1. *Walled garden application*, yaitu aplikasi yang berdiri sendiri yang berjalan pada handphone tanpa perlu mengakses sumber data eksternal melalui jaringan pembawa atau *carrier network*. Contoh dari aplikasi ini adalah kalkulator atau single player games.
2. *Network aware application*, yaitu aplikasi yang berinteraksi dengan jaringan.

Aplikasi ini memiliki kemampuan untuk mengakses sumber data eksternal. Contoh dari aplikasi ini adalah aplikasi email yang berada di dalam handphone, aplikasi untuk mendapatkan kembali data alamat-alamat yang tersimpan melalui jaringan, dan pengiriman email berbagai alamat melalui jaringan data.

2.5.1 Konfigurasi J2ME

Ada dua konfigurasi untuk J2ME, yaitu *Connected Limited Device Configuration* (CLDC) dan *Connected Device Configuration* (CDC). CLDC di disain untuk 16-bit atau 32-bit peralatan komputer kecil dengan jumlah memori yang terbatas. Peralatan CLDC biasanya memiliki memori antara 160KB sampai 512KB dengan bantuan batere dan mengimplementasikan KVM. Peralatan CLDC juga menggunakan koneksi *wireless* dengan jaringan bandwidth kecil dan kemungkinan tidak memiliki *user interface*. Contoh dari peralatan CLDC yaitu

pager, PDA, telepon seluler, dan peralatan lain yang memiliki memori antara 128KB sampai 512KB.

Peralatan CDC menggunakan arsitektur 32-bit, memiliki sedikitnya 2 megabyte kapasitas memori, dan mengimplementasikan seluruh fungsi JVM. Contoh dari peralatan CDC yaitu sistem navigasi, terminal *point-of-sale*, dan *smart phone*.

2.5.2 J2ME Profiles

Terdiri dari kelas-kelas java yang memungkinkan implementasi dari fitur-fitur untuk peralatan komputer kecil khusus atau yang biasa. Terdapat tujuh profile, yaitu:

1. *The Foundation Profile* digunakan dengan konfigurasi CDC dan adalah inti hampir di semua profile yang digunakan dengan konfigurasi CDC karena Foundation Profile mengandung inti kelas-kelas java.
2. *The Game Profile* juga digunakan dengan konfigurasi CDC dan mengandung kelas-kelas yang diperlukan untuk membuat aplikasi game untuk semua peralatan komputer kecil yang menggunakan konfigurasi CDC.
3. *The Mobile Information Device Profile (MIDP)* digunakan dengan konfigurasi CLDC dan mengandung kelas-kelas yang menyediakan penyimpanan lokal, *user interface*, dan kemampuan *networking* dari aplikasi yang berjalan pada *mobile computing device* seperti Palm OS. MIDP digunakan dengan aplikasi java wireless.
4. *The PDA Profile (PDAP)* digunakan dengan konfigurasi CLDC dan mengandung kelas-kelas yang memanfaatkan sumber daya canggih yang ditemukan dalam PDA. Profil-profil ini termasuk tampilan yang lebih baik

dan memori yang lebih besar dibandingkan dengan sumber daya biasa yang ditemukan dalam peralatan mobile MIDP (seperti telepon seluler).

5. *The Personal Profile* digunakan dengan konfigurasi CDC dan Foundation Profile dan mengandung kelas-kelas untuk mengimplementasikan user interface yang kompleks.
6. *The Personal Basis Profile* sama dengan *Personal Profile* yang digunakan dengan konfigurasi CDC dan *Foundation Profile*.
7. *The RMI Profile* digunakan dengan konfigurasi CDC dan *Foundation Profile* untuk menyediakan kelas-kelas *Remote Method Invocation* kepada kelas-kelas inti yang terkandung dalam Foundation Profile.

2.5.3 MIDlet

MIDlet adalah aplikasi yang dibuat menggunakan J2ME dengan profile MIDP. MIDP dikhususkan untuk digunakan pada handset dengan kemampuan komputasi, memori, keyboard, dan layar yang terbatas. Arsitektur tingkat tinggi

dari sebuah aplikasi MIDP ditunjukkan oleh gambar berikut:

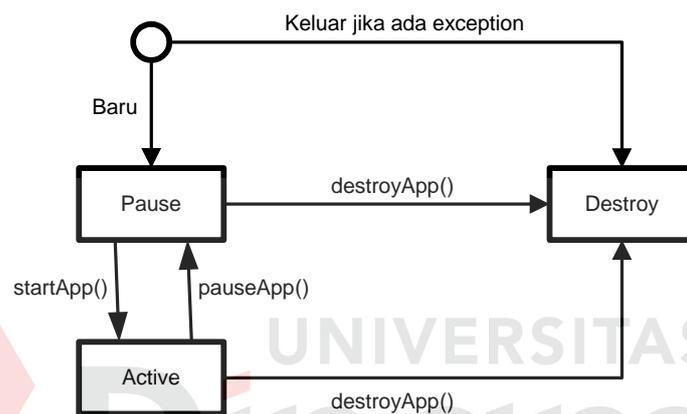


Gambar 2.11 Arsitektur tingkat tinggi dari sebuah aplikasi MIDP

Siklus hidup MIDlet

Siklus hidup dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS ini adalah lingkungan tempat siklus sebuah MIDlet diciptakan, dijalankan, dihentikan, maupun dihilangkan. AMS sering pula dinamakan dengan *Java Application Manager* (JAM).

MIDlet memiliki berapa *state*, yaitu *Pause*, *Active*, dan *Destroy* seperti terlihat pada gambar berikut:



Gambar 2.12 Siklus hidup dari sebuah MIDP

Ketika masing-masing *state* dipanggil, beberapa method standar yang bersesuaian dipanggil. Method-method ini merupakan bawaan dari class MIDlet.

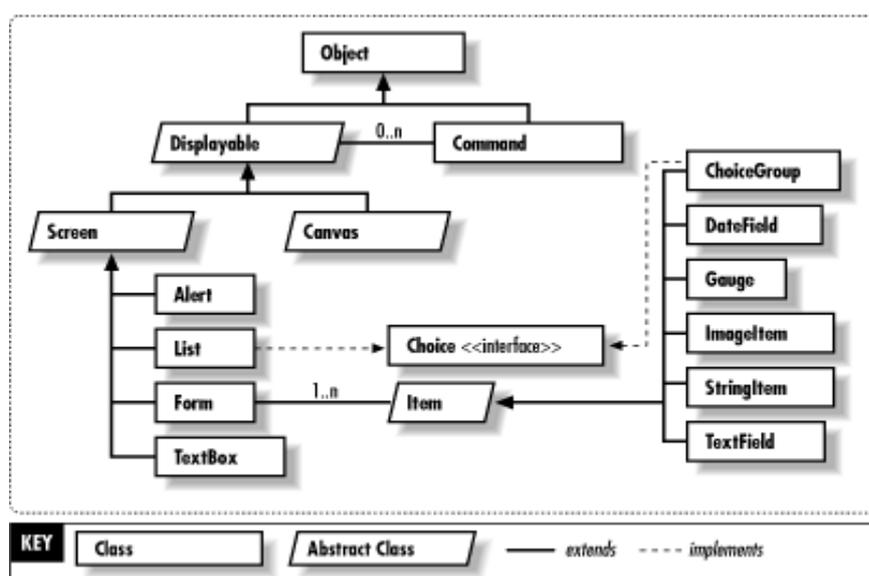
Dari gambar di atas terlihat bahwa:

- Ketika MIDlet pertama kali diciptakan dan diinisialisasi, MIDlet akan berada pada state “Pause”.
- Apabila terjadi kesalahan selama konstruksi MIDlet, MIDlet akan berpindah dari state “Pause” ke state “Destroy”, dan MIDlet batal diciptakan dengan jalan memanggil fungsi `destroyApp()`.
- Selanjutnya, ketika MIDlet dijalankan, MIDlet akan berada pada state “Active”, dalam hal ini fungsi yang dipanggil adalah `startApp()`.

- d. Akan tetapi, jika di tengah jalan MIDlet dihentikan sementara, MIDlet akan berada dalam state “Pause” dengan jalan memanggil fungsi `pauseApp()`. Pada state ini diperlukan proses *cleanup* terhadap *garbage collector* yang dihasilkan.

2.5.4 Antar muka

Antar muka terdiri dari class-class yang akan digunakan sebagai sarana interaksi antara aplikasi yang terdapat di dalam telepon seluler dengan user. Antar muka dari MIDP terbagi menjadi dua bagian yaitu high level user interface dan low level user interface. Low level user interface berbasis pada class `Canvas` sedangkan high level user interface berbasis pada class `Screen`. Low level user interface biasanya digunakan untuk mempresentasikan input atau output dengan bentuk gambar, seperti *chart*. Sedangkan High level user interface biasanya digunakan untuk mempresentasikan input atau output dengan bentuk form. Hirarki class user interface yang ada dalam J2ME adalah sebagai berikut:



Gambar 2.13 Class diagram dari user interface

2.5.5 Mobile Information Device Profile 2.0 (MIDP 2.0)

MIDP 2.0 mendukung beberapa fasilitas yang cukup baik. Fitur-fitur yang terdapat dalam MIDP 2.0 adalah :

1. Antar Muka

Banyak fitur-fitur baru yang terdapat dalam MIDP 2.0 yang akan membuat aplikasi MIDP semakin interaktif dan mudah untuk digunakan. MIDP 2.0 menyediakan tampilan yang lebih fleksibel untuk portabilitas aplikasi yang lebih baik. Layar form menyediakan lebih banyak kontrol berdasarkan pada ukuran tampilan dan atribut yang lain. Dalam MIDP 2.0 juga telah disediakan Custom Items, kontrol baru yang memungkinkan pengguna untuk membuat form sesuai keinginan mereka.

2. Multimedia

Dalam MIDP 2.0 sekarang terdapat *Audio Building Block* (ABB) yang merupakan bagian dari *Mobile Media API* (MMAPI). ABB memungkinkan pengguna untuk menambahkan audio seperti ringtone dan WAV file ke dalam aplikasi MIDP. MMAPI memungkinkan pengguna untuk menambahkan multimedia dalam aplikasi MIDP seperti *video streaming*.

3. Game

MIDP 2.0 juga menambahkan Game API yang menyediakan pondasi standar untuk membuat game. MIDP Game API menyertakan fungsionalitas game yang spesifik, seperti *sprites* dan *tiled layers*. Fungsi ini memudahkan pengguna dan menyediakan kontrol yang lebih baik pada gambar dan tampilan.

4. Konektivitas

MIDP 2.0 menambahkan suport untuk konektivitas standar dibawah HTTP, seperti HTTPS, datagram, sockets, server socket, dan komunikasi port serial.

5. Arsitektur *Push*

MIDP 2.0 memiliki model push pada server dimana MIDlet dapat diaktifkan ketika telepon seluler menerima informasi dari server. Seting aplikasi, dimana kombinasi dari seting standar telepon seluler dan seting dari pengguna, menentukan pemberitahuan kepada user untuk mengaktifkan MIDlet atau tidak ketika menggunakan MIDlet yang lain. Arsitektur *push* pada MIDP memungkinkan programmer untuk menggunakan kemampuan *event-driven* pada telepon seluler dan *carrier network*, juga termasuk *alerts*, pesan dan *broadcasts* menggunakan pendekatan standar aplikasi MIDP.

6. *Over-the-air (OTA) Provisioning*

Fitur baru yang sangat penting dalam MIDP 2.0 adalah kemampuan deploy secara dinamis dan mengupdate aplikasi secara langsung atau OTA. Spesifikai MIDP mendefinisikan bagaimana MIDlet ditemukan, diinstal, di-update, dan dihapus dari telepon seluler.

7. *End-to-end Security*

MIDP 2.0 mendukung HTTPS dan juga SSL dan WTLS yang memungkinkan transmisi data enkripsi. Dalam MIDP 2.0 kemanan utama melindungi pengaksesan data oleh akses yang tidak dikehendaki yang mengakibatkan kerusakan data.

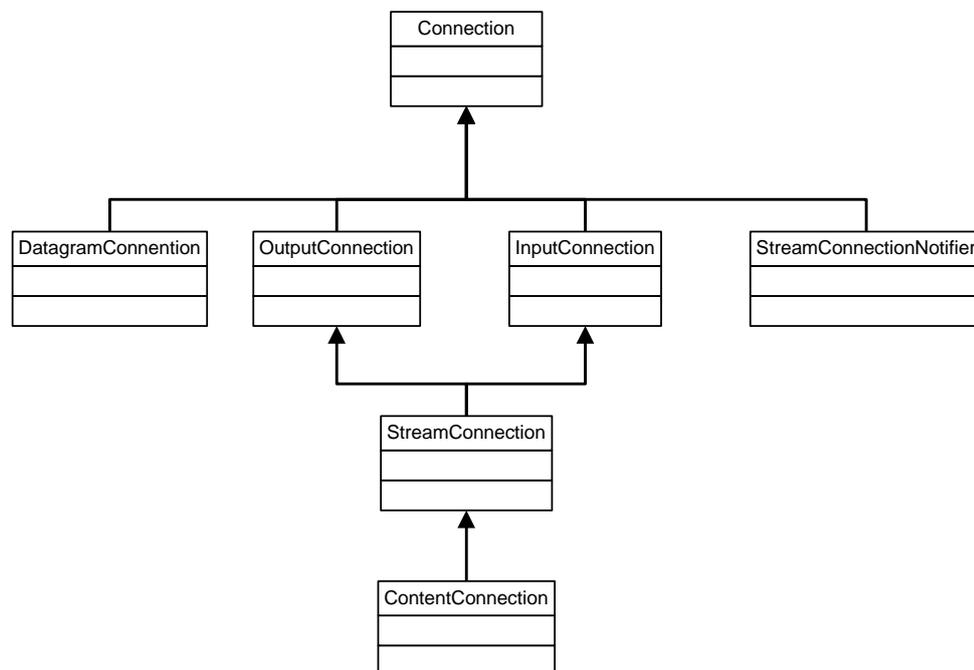
2.5.6 Generic Connection Framework (GCF)

CDLC merupakan standar lingkungan *runtime* J2ME yang digunakan pada telepon seluler. CDLC merupakan basis kerja bagi MIDP. Untuk fasilitas *networking* CDLC mendefinisikan GCF yang diimplementasikan dalam paket `javax.microedition.io`. GCF ini tidak sekedar menyediakan fasilitas I/O bagi interkoneksi jaringan, namun juga fasilitas I/O secara umum, misalnya untuk akses *Infrared Defice* (IrDA), file dan SMS. Khusus untuk interkoneksi dengan jaringan Internet berbasis web menggunakan *Hypertext Transport Protocol* (HTTP), CLDC menyediakan *interface* `javax.microedition.io.HttpConnection` untuk membuat aplikasi yang mengakses web (protokol HTTP).

Paket `javax.microedition.io` yang membungkus CGF untuk interkoneksi jaringan terdiri atas:

- a. Sebuah kelas java, yaitu kelas *Connector*
- b. Sebuah java *exception*, yaitu *ConnectionNotFoundException*
- c. Sembilan buah java *interface*, yaitu *Connection*, *ContentConnection*, *Datagram*, *DatagramConnection*, *InputConnection*, *OutputConnection*, *StreamConnection*, *StreamConnectionNotifier*, *HttpConnection*.

Khusus mengenai keterhubungan dari java *interface*, dapat dilihat pada gambar di bawah:



Gambar 2.14 Hirarki java interface dalam GCF

2.5.7 Record Management System (RMS)

RMS atau *Record Management System* merupakan implementasi basis data secara sederhana yang dipergunakan oleh J2ME. *Record store* merupakan kumpulan *record* seperti halnya tabel, yang direpresentasikan dalam bentuk *byte array* (Yu Feng and Jun Zhu, 2001). Record store memiliki karakteristik sebagai berikut :

1. Disimpan sebagai *record of array*.
2. Ukuran setiap *record* dapat berbeda-beda.
3. Setiap *record* memiliki *record id* yang unik.
4. Tidak memiliki metadata.
5. Tidak dapat didefinisikan dan dimanipulasi dengan SQL

Untuk mempergunakan RMS, sebuah MIDlet harus dibuat dengan mengimport paket `javax.microedition.rms`.

2.5.8 Emulator J2ME

Seperti pada Java Applet, aplikasi dapat dijalankan menggunakan web browser ataupun Java Interpreter sebelum proses pengembangan aplikasi, sedangkan MIDlet hanya dapat dijalankan dengan menggunakan emulator yang menyerupai perangkat mobile yang sebenarnya. Dari berbagai macam emulator yang disediakan oleh J2ME Wireles Tool Kit maupun yang disediakan oleh vendor telepon seluler, emulator menyediakan fungsi yang sama walaupun pada saat dijalankan memiliki sedikit perbedaan.

J2ME emulator adalah alat bantu berupa perangkat lunak yang digunakan untuk menjalankan aplikasi MIDlet pada PC dan menyimulasikan bagaimana status aplikasi MIDlet berjalan pada sebuah perangkat fisik mobile device.

Emulator menampilkan gambar dari mobile device yang sebenarnya pada layar sebuah PC, dan menjalankan aplikasi MIDlet pada layar. Keuntungan dari penggunaan J2ME emulator adalah :

1. Dapat menunda pelaksanaan tes pada perangkat fisik mobile device yang sebenarnya sampai dengan tahap akhir pengembangan aplikasi MIDlet.
2. Dapat melakukan tes terhadap MIDlet pada berbagai macam mobile device.
3. Dapat menelusuri aspek-aspek yang spesifik pada saat MIDlet dijalankan seperti pemanggilan class, pemanggilan method, dan garbage collector.



Gambar 2.15 Emulator J2ME

Dengan menggunakan emulator tentu saja tidak terlepas dari keterbatasan-keterbatasan untuk menyerupai atau menyamai mobile device yang sebenarnya. Keterbatasan emulator belum merupakan hal yang fatal, akan tetapi perlu diketahui sebelum mencobakan pada perangkat fisik mobile device yang sebenarnya. Beberapa aspek dari aplikasi yang tidak dapat diatur atau dicobakan pada emulator adalah:

a. Kecepatan Eksekusi

Mobile device sangat beraneka ragam dengan perangkat kerasnya, sehingga kecepatan prosesor tidak dapat diidentifikasi kecuali telah ditentukan untuk salah satu mobile device dan mengabaikan yang lainnya.

b. Ketersediaan Memori

Kecepatan prosesor sangat bervariasi untuk masing-masing perangkat mobile, begitu juga dengan memori yang akan digunakan aplikasi MIDlet. Ketersediaan memori sangat berpengaruh terhadap proses eksekusi MIDlet, sehingga perlu dilakukan tes terhadap MIDlet dengan keterbatasan-keterbatasan memori untuk setiap mobile device. Namun pada saat ini J2ME emulator tidak dapat menspesifikasikan berbagai macam ketersediaan memori untuk sebuah aplikasi MIDlet.

c. Application Manager

Pada saat ini tidak ada dukungan untuk *application management* pada emulator, *application manager* pada suatu device dapat mengatur aplikasi MIDlet seperti instalasi dan penghapusan aplikasi MIDlet.

2.5.9 Komunikasi Data

Manfaat terbesar dari piranti seluler adalah fleksibilitas koneksi dan akses. Piranti seluler membuat orang terhubung ke dunia luar setiap saat dari lokasi manapun. Kegunaan piranti seluler telah berubah. Saat ini telepon seluler bukan hanya sebagai alat komunikasi suara tetapi sudah menjadi “*mobile information device*” yaitu telepon seluler dengan kemampuan mengakses data. Untuk terhubung ke dunia luar, telepon seluler menggunakan komunikasi data berikut :

1. Global System for Mobile (GSM)

GSM merupakan jaringan komunikasi seluler yang berkembang di benua Eropa. GSM merupakan salah satu standart sistem komunikasi seluler digital yang diterima dan diterapkan oleh hampir seluruh negara. GSM dikembangkan oleh sebuah grup yang dibentuk oleh *Conference of European Post and Telecommunication* pada tahun 1982 dengan mengkombinasikan *Time Devision Multiple Access* (TDMA) dan *Frequency Division Multiple Acces* (FDMA). Saat ini layanan GSM tersedia dalam 4 frequency yaitu 450 Mhz, 900 Mhz, 1800 Mhz, dan 1900 Mhz. Sebuah jaringan GSM memiliki arsitektur yang terdiri atas 3 bagian :

- a. *Mobile Station*, merupakan piranti keras yang digunakan oleh pengguna untuk berkomunikasi; misalnya telepon seluler.
- b. *Base Station Subsystem*, digunakan untuk mengontrol radio yang terhubung dengan Mobile Station.
- c. *Network Subsystem*, digunakan untuk melakuikan pertukaran panggilan antara pengguna telepon seluler, antara pengguna telepon seluler dengan pengguna jaringan telepon digital dan juga melakukan aktifitas manajemen operasi.

2. General Packet Radio Service (GPRS)

GPRS yang termasuk dalam kelas 2.5 G adalah standar komunikasi data pada jaringan yang kecepatan transfernya mencapai 115 kbps. Dengan adanya GPRS ini jaringan bisa memisah paket data kecepatan tinggi dengan suara. Selain itu dengan GPRS pengguna bisa terus terkoneksi ke internet. Pengguna tidak perlu dial up terus menerus ketika akan melakukan koneksi ke internet. Tagihan internet

tidak berdasar lama waktu penggunaan internet namun berdasar banyaknya data yang dikirim/diterima.

Berbagai layanan komunikasi data baru dimungkinkan dengan adanya GPRS. Layanan-layanan komunikasi data ini pada umumnya memiliki karakteristik sebagai berikut :

a. Mobility

Kemampuan untuk menjaga kestabilan komunikasi suara dan data selama bergerak.

b. Immediacy

Mengijinkan pengguna untuk melakukan koneksi ketika diperlukan dimanapun pengguna berada tanpa mempedulikan lamanya koneksi.

c. Localization

Mengijinkan pengguna untuk dapat memperoleh layanan berbasis lokasi dimana pengguna berada.

2.6 Perancangan Sistem

Perancangan adalah langkah pertama dalam fase pengembangan untuk semua sistem atau produk yang terencana. Didefinisikan sebagai proses mengaplikasikan bermacam-macam teknik dan prinsip untuk tujuan pendefinisian suatu alat, proses, atau sistem dalam detail yang cukup untuk mencapai realisasi fisik. Tujuan dari perancangan adalah untuk menghasilkan suatu model atau representasi dari *entity* yang akan dibuat. Pendekatan perancangan perangkat lunak komputer melibatkan disiplin ilmu yang lain, berubah secara berkelanjutan, analisa yang lebih baik, dan pengetahuan yang terus berkembang.

Pentingnya perancangan perangkat lunak dapat dinyatakan dengan satu kata yaitu kualitas. Perancangan adalah tempat dimana kualitas dibantu dalam pengembangan software. Perancangan perangkat lunak adalah sebuah proses dimana permintaan diterjemahkan dalam representasi perangkat lunak. Perancangan adalah satu-satunya cara yang secara akurat menerjemahkan permintaan customer ke dalam sistem atau produk perangkat lunak yang sudah jadi. Perancangan sistem adalah dasar dari semua *software engineering*. Tanpa perancangan, kita beresiko membangun sistem yang tidak stabil. Beberapa karakteristik dari perancangan yang baik adalah sebagai berikut :

1. Perancangan haruslah *modular*, yaitu perangkat lunak haruslah secara logika dibagi dalam komponen-komponen yang menjalankan fungsi tertentu.
2. Perancangan haruslah berisi representasi data dan prosedur yang berbeda dan terpisah.

2.6.1 Sistem Flowchart

Sistem Flowchart merupakan alat bantu yang banyak digunakan untuk menggambarkan sistem secara fisik. Di dalam menggambarkan arus data dokumen serta proses yang dilakukan terhadap data dan dokumen dalam sistem, serta dengan cara menggunakan simbol-simbol standard yang biasa dipakai. Sistem flow dalam sistem ini adalah Sistem Flow Komputerisasi.

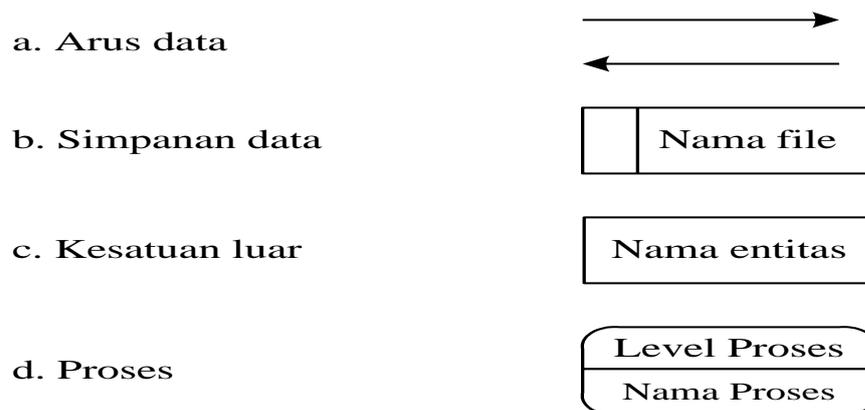
2.6.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan suatu metode representasi sistem dalam semua level detail yang diinginkan dengan menggunakan jaringan/hubungan simbol-simbol tertentu. Tujuan dari DFD ini adalah membangun jembatan semantik antara user dengan pengembang sistem, yakni :

1. *Graphical*, menjelaskan tanpa perlu menggunakan banyak kata.
2. *Logical Representation*, membuat model tentang apa yang dikerjakan sistem dan bukan model fisik tentang bagaimana sistem tersebut bekerja.
3. *Hierarchical*, menggambarkan sistem dalam setiap level detail yang diinginkan.
4. *Jargonless*, memungkinkan semua user untuk mengerti dan mengenalinya.

Sasaran yang diinginkan dari penggunaan DFD ini adalah membangun suatu model yang dapat menggambarkan sistem dan dapat dimengerti secara mudah. DFD merupakan dasar dari analisa sistem secara terstruktur. Perancangan sistem dengan menggunakan DFD ini diawali dengan masuknya arus data kedalam proses dan dihasilkan arus data yang keluar dari proses. Dan setiap proses dilengkapi dengan penjelasan yang lengkap mengenai identifikasi proses dan nama proses.

Simbol yang digunakan pada DFD dalam Tugas Akhir ini menggunakan model Gane & Sarson yaitu :



Arus data di DFD diberi simbol suatu panah. Arus data ini digunakan untuk menunjukkan arah aliran data dari proses, kesatuan luar dan file yang dibuat. Simpanan data merupakan nama file untuk menyimpan data atau untuk mengambil data sesuai proses apa yang sedang di kerjakan. Kesatuan luar merupakan kesatuan (entity) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem. Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang akan keluar dari proses. Identifikasi proses dapat berupa angka yang ditulis pada bagian atas proses yang digunakan sebagai nomor acuan dari proses. Nama proses menunjukkan kegiatan yang dilakukan oleh proses tersebut dan diletakkan dibawah identifikasi proses.

Pada DFD terdapat beberapa tingkatan, biasanya disebut dengan level yang dimulai dari level 0, level 1 dan seterusnya. DFD level 0 disebut juga dengan context diagram yang merupakan gambaran global dari suatu sistem. DFD level 1 dan seterusnya merupakan gambaran yang lebih detail dari setiap proses yang dilakukan dalam sistem.

2.6.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menginterpretasikan, menentukan dan mendokumentasikan kebutuhan untuk sistem pemrosesan database. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari aplikasi. Adapun elemen-elemen dari ERD ini adalah :

1. Entitas, adalah sesuatu yang dapat diidentifikasi di dalam lingkup pemakai, sesuatu yang penting bagi pemakai dari sistem yang akan dikembangkan.
2. Atribut, entitas memiliki atribut yang berfungsi untuk menjelaskan karakteristik dari entitas.
3. Pengidentifikasi, data-data entitas memiliki nama yang berfungsi untuk mengidentifikasi mereka. Sebuah identifikasi dapat bersifat unik atau tidak unik.

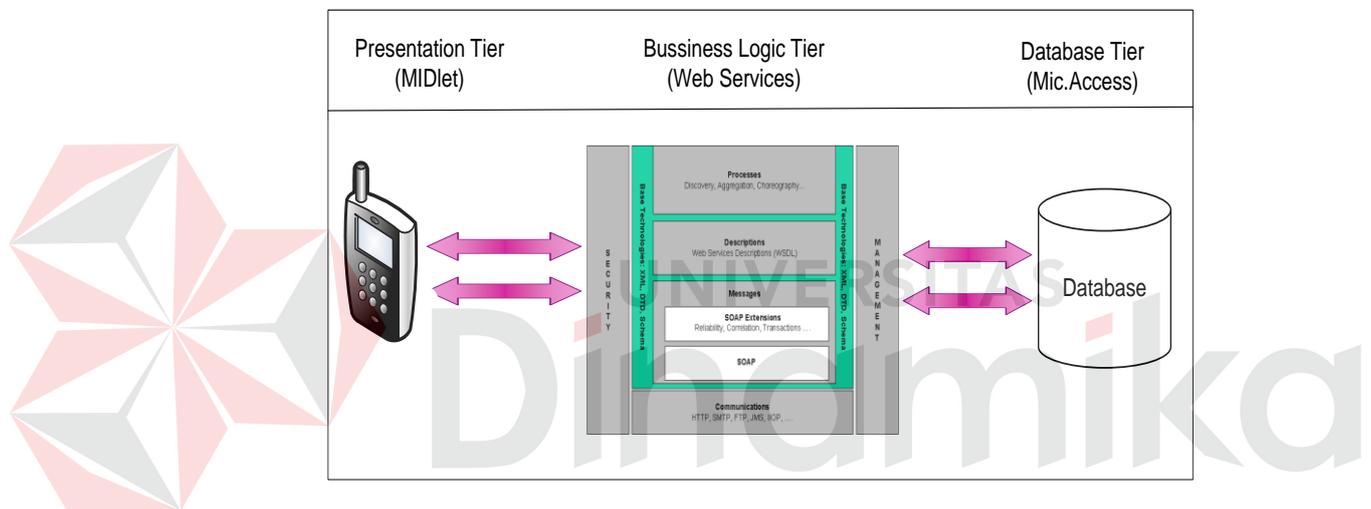
Hubungan atau relasi berfungsi untuk menunjukkan hubungan satu entitas dengan entitas yang lain. Hubungan ini boleh memiliki atribut. Banyaknya entitas dalam suatu relasi menunjukkan tingkat dari relasi yang bersangkutan, namun yang banyak digunakan dalam aplikasi-aplikasi adalah model yang menggunakan relasi tingkat dua atau yang disebut dengan hubungan biner. Hubungan biner ini memiliki tiga tipe yaitu hubungan biner satu ke satu, hubungan biner satu ke banyak dan hubungan biner banyak ke banyak.

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Perancangan Model Arsitektur Multi-Tier

Aplikasi sistem pemesanan tiket ini menggunakan model arsitektur multi-tier. Penggambaran secara umum dari arsitektur multi-tier dapat dilihat pada gambar dibawah.



Gambar 3.1 Model Arsitektur Multi-Tier

Aplikasi sistem pemesanan tiket ini dikembangkan dengan arsitektur Multi-Tier. Arsitektur Multi-Tier terdiri dari Presentation Tier, Bussiness Logic Tier, dan Database Tier. Berikut adalah penjelasan dari masing-masing tier.

1. Presentation Tier (MIDlet pada telepon seluler)

Aplikasi sistem pemesanan tiket pada bagian ini terdapat pada telepon seluler yang diimplementasikan dengan menggunakan MIDlet. Untuk melakukan pemesanan, telepon seluler dapat memperoleh data-data yang

diperlukan dengan memanggil fungsi-fungsi yang terdapat pada web service. Telepon seluler dapat terhubung dengan web service melalui GPRS.

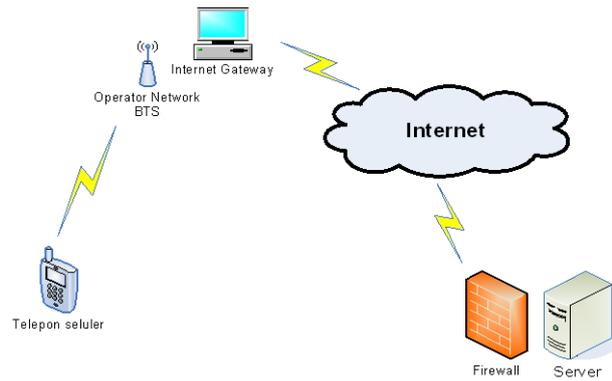
2. Bussiness Logic Tier (Web Service)

Bagian ini adalah tahap pengimplementasian web services dan web server. Di dalam web services terdapat banyak fungsi yang digunakan untuk mengambil data yang terdapat pada database. Fungsi-fungsi tersebut akan dipanggil oleh MIDlet telepon seluler untuk mendapatkan data yang diinginkan. Dengan meletakkan web service pada IIS maka fungsi-fungsi yang terdapat pada web service dapat dipanggil melalui akses internet. Setiap fungsi akan menghasilkan data dalam bentuk file XML untuk kemudian dikirim ke MIDlet.

3. Database Tier (Microsoft Access)

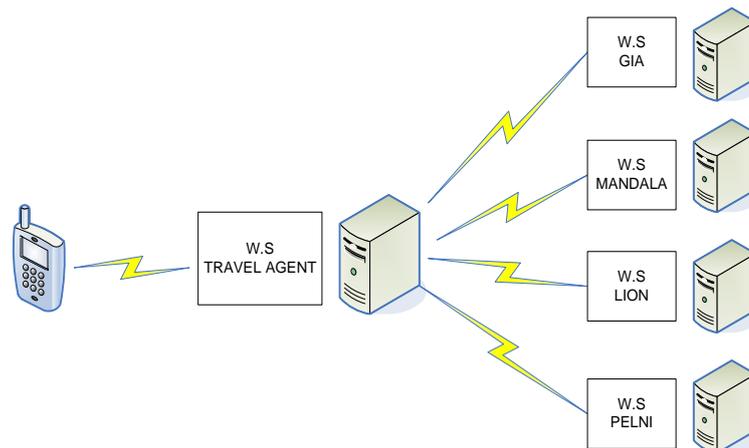
Bagian ini merupakan bagian untuk melakukan proses penyimpanan data. Sistem pemesanan tiket ini menggunakan Microsoft Access untuk keperluan penyimpanan data.

Aplikasi pada sistem ini nantinya akan menggunakan telepon seluler pada sisi client yang akan melakukan koneksi ke web service melalui jaringan Internet dengan dukungan GPRS. Operator penyedia layanan GPRS sebuah telepon seluler mampu berkomunikasi dengan jaringan Internet tanpa menggunakan alamat IP. Berikut ini dijelaskan bagaimana telepon seluler berkomunikasi dengan jaringan internet:



Gambar 3.2 Akses jaringan Internet dari telepon seluler

Sistem pemesanan tiket ini mensimulasikan web service biro perjalanan terhubung secara *online* ke web service tiap-tiap maskapai, dan perusahaan kapal laut nasional. Seperti yang telah dijelaskan pada batasan masalah, digunakan 3 contoh maskapai yaitu Garuda Indonesia Airways (GIA), Mandala Airlines, dan Lion Air, dan 2 jenis kapal laut nasional PELNI. Hubungan antara telepon seluler, web service biro perjalanan dan web service tiap-tiap perusahaan dapat dilihat pada bagan berikut :

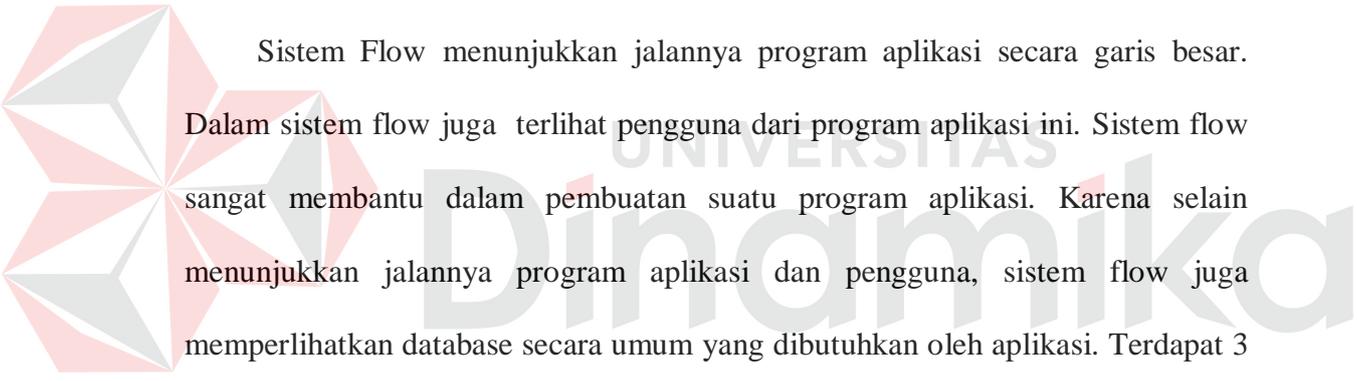


Gambar 3.3 Hubungan Tiap-tiap Web Service Sistem Pemesanan Tiket

3.2 Perancangan Sistem

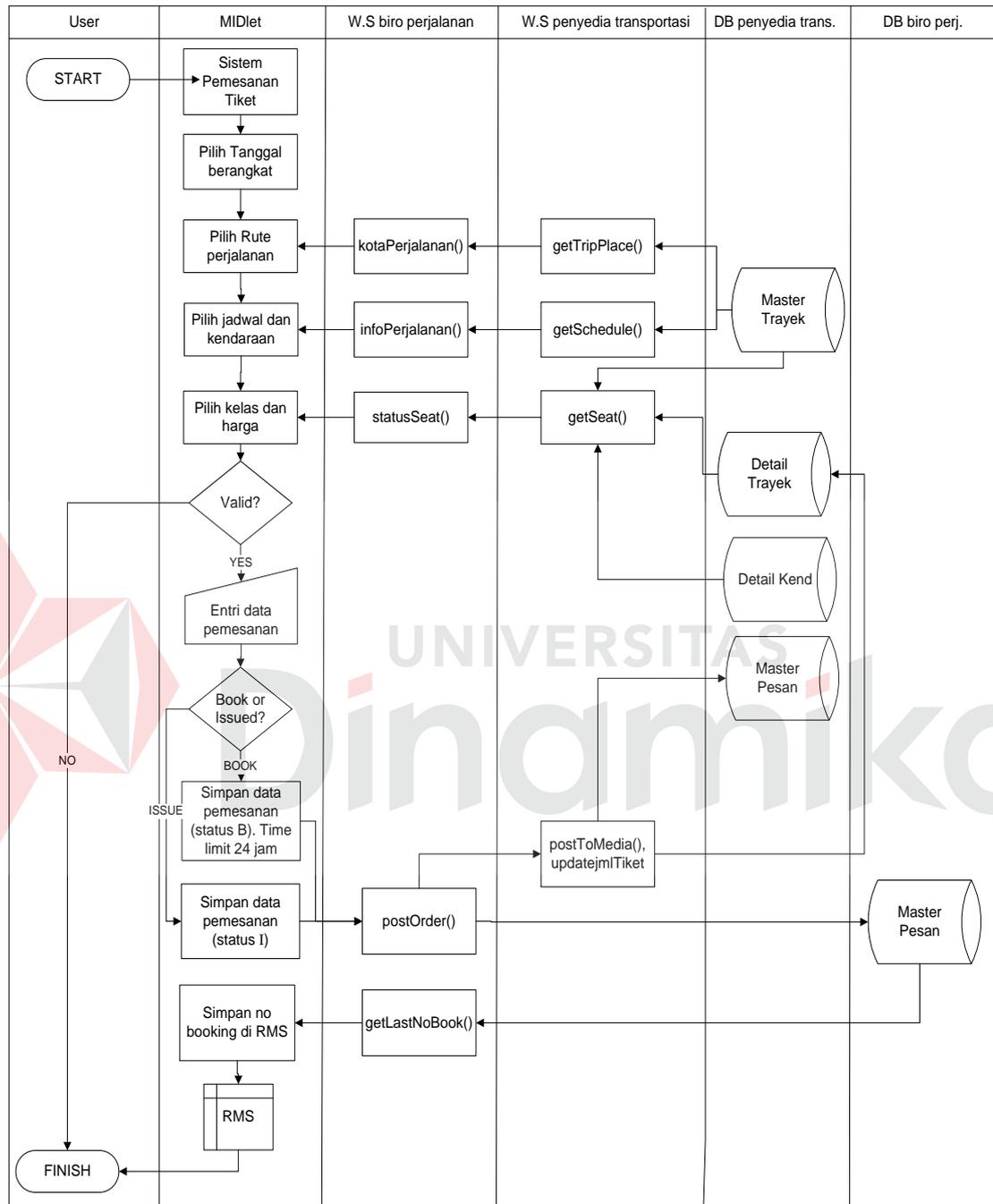
Sebelum membuat program aplikasi, terlebih dahulu dilakukan proses perancangan sistem. Hal ini dilakukan dengan tujuan supaya aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan. Dalam perancangan sistem ini ada beberapa tahapan-tahapan yang harus dilakukan. Adapun tahapan-tahapan dalam perancangan sistem yang dilakukan adalah pembuatan Sistem Flow, Diagram Berjenjang, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), dan Struktur Database yang digunakan dalam program aplikasi ini.

3.2.1 Pembuatan Sistem Flow



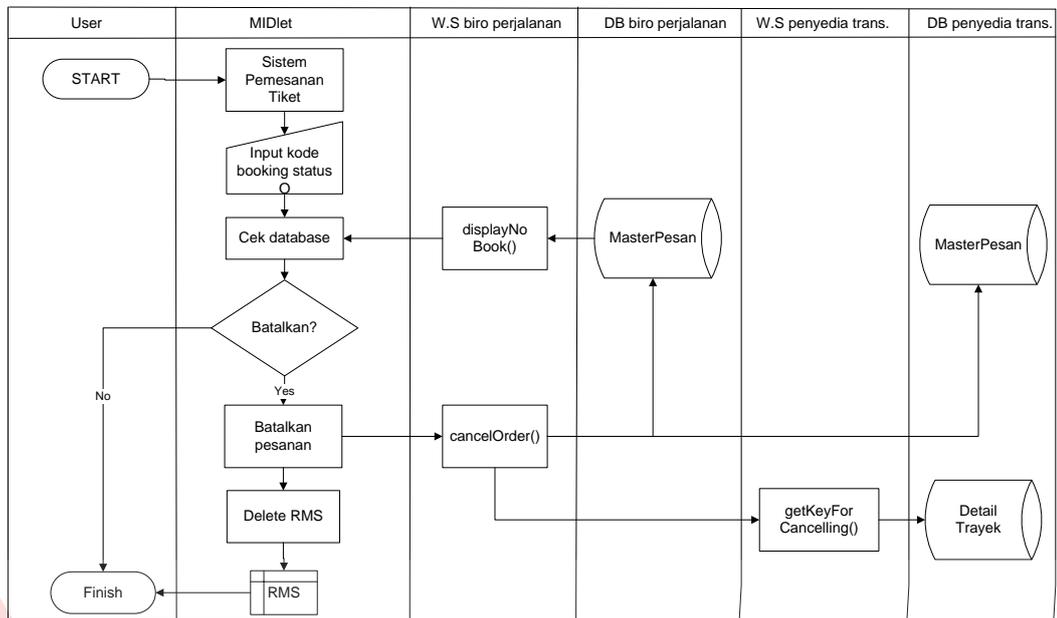
Sistem Flow menunjukkan jalannya program aplikasi secara garis besar. Dalam sistem flow juga terlihat pengguna dari program aplikasi ini. Sistem flow sangat membantu dalam pembuatan suatu program aplikasi. Karena selain menunjukkan jalannya program aplikasi dan pengguna, sistem flow juga memperlihatkan database secara umum yang dibutuhkan oleh aplikasi. Terdapat 3 macam sistem flow yang terdapat pada aplikasi sistem pemesanan tiket, yaitu :

a. Sistem Flow Pemesanan



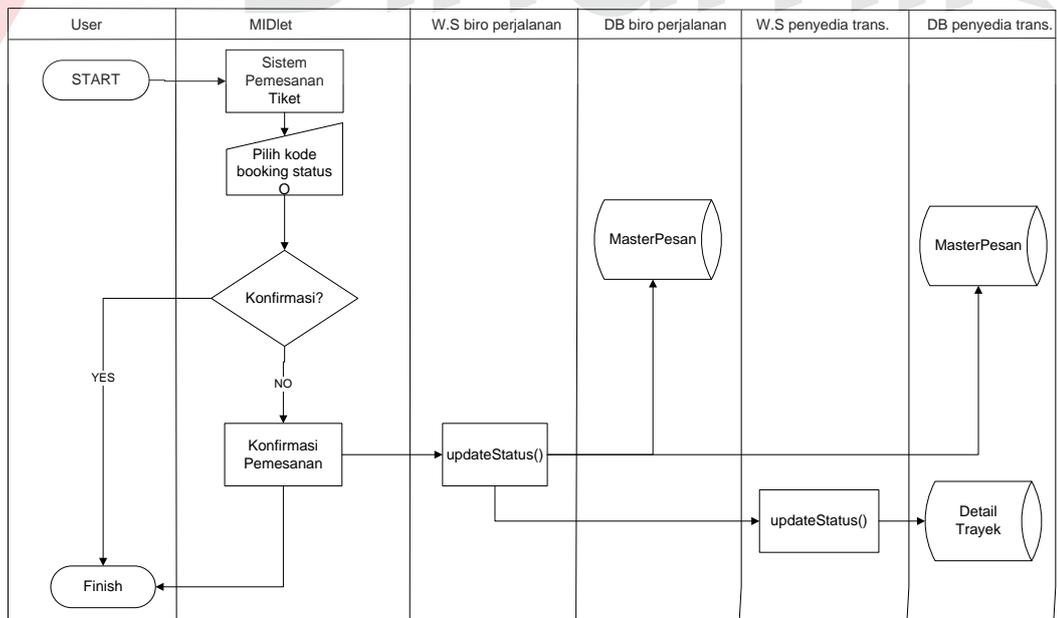
Gambar 3.4 Sistem Flow Pemesanan

b. Sistem Flow Pembatalan



Gambar 3.5 Sistem Flow Pembatalan

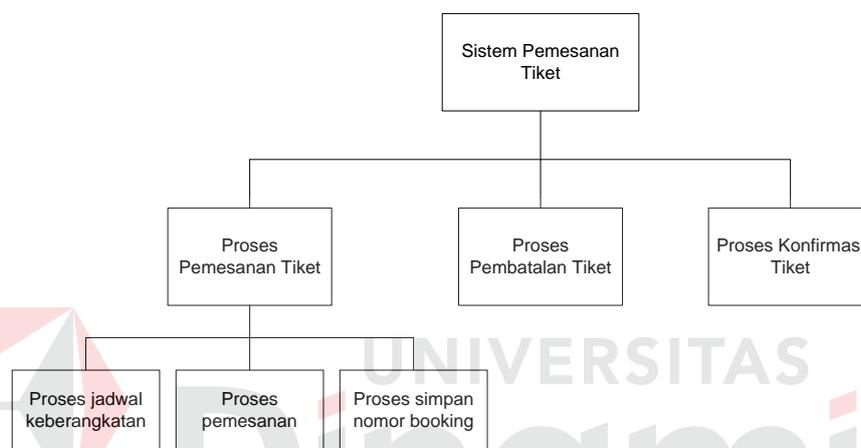
c. Sistem Flow Konfirmasi Pemesanan



Gambar 3.6 Sistem Flow Konfirmasi Pemesanan

3.2.2 Pembuatan Diagram Berjenjang

Diagram berjenjang menunjukkan secara garis besar proses-proses apa saja yang terjadi di dalam aplikasi sistem pemesanan tiket. Diagram berjenjang berguna untuk mempermudah pembuatan DFD sehingga pembuatan proses-proses dalam DFD cukup mengacu pada diagram berjenjang saja. Diagram berjenjang pada aplikasi ini dapat dilihat pada gambar berikut :



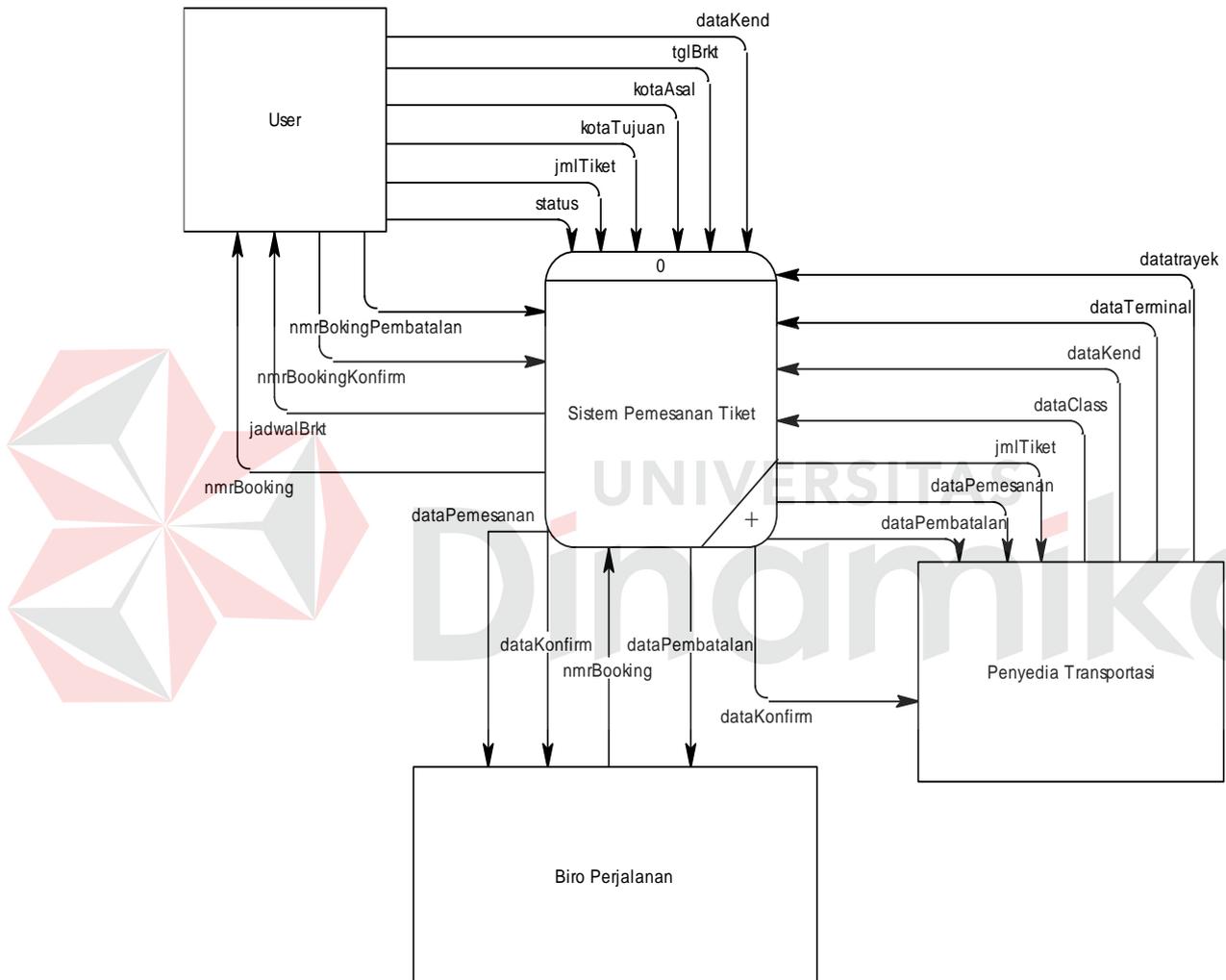
Gambar 3.7 Diagram Berjenjang

3.2.3 Pembuatan DFD

Data Flow Diagram berfungsi untuk menggambarkan proses aliran data yang terjadi dalam sistem dari tingkat tertinggi sampai yang terendah, yang memungkinkan kita untuk melakukan dekomposisi atau membagi sistem kedalam bagian-bagian yang lebih kecil dan lebih sederhana. Data Flow Diagram dari aplikasi ini dapat dilihat pada gambar berikut :

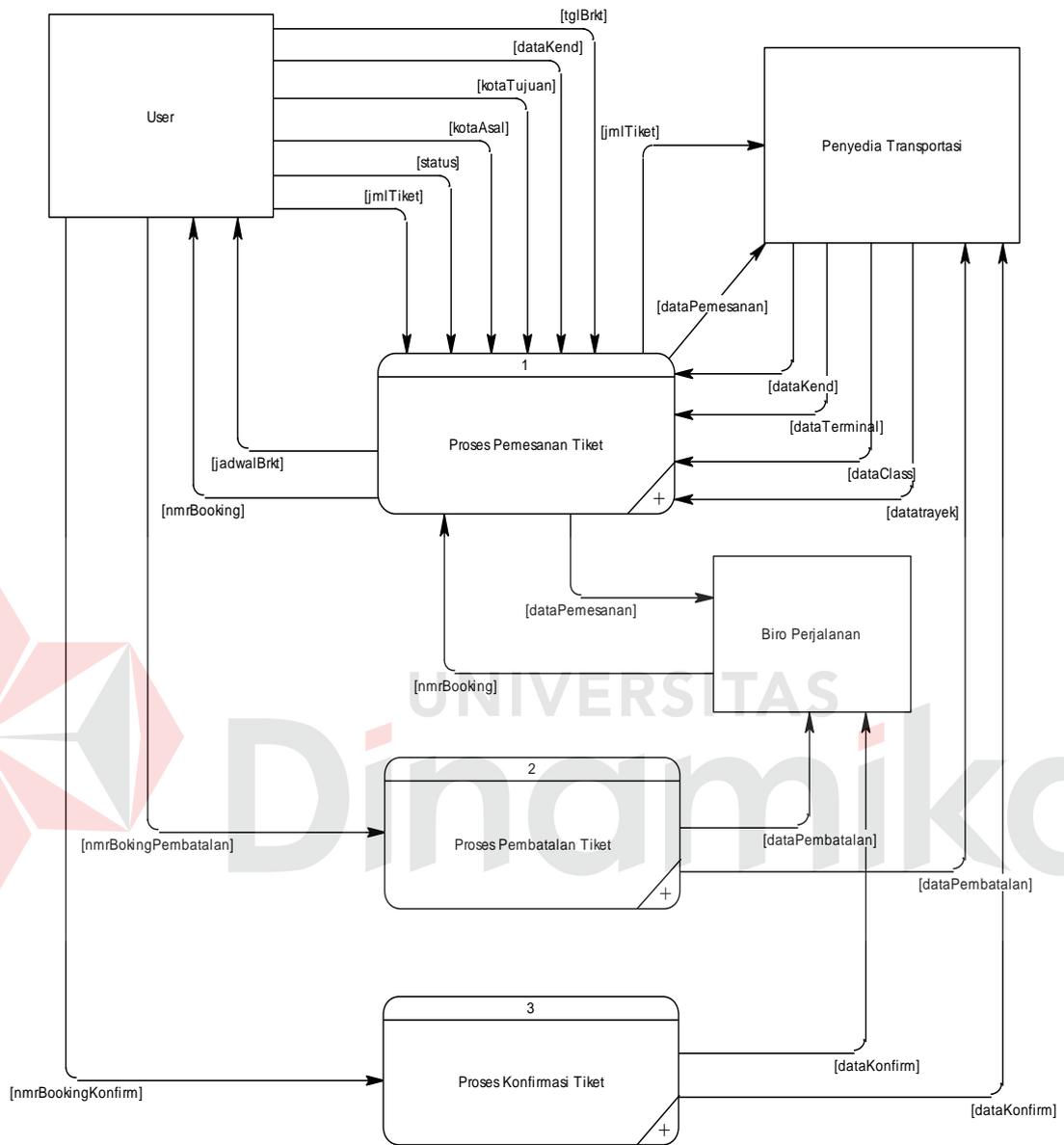
a. DFD Level 0

DFD level 0 menggambarkan proses aliran data yang terjadi dalam sistem secara garis besarnya. Selanjutnya DFD level 0 dapat didekomposisi menjadi DFD level 1 yang menjelaskan proses pada level yang lebih tinggi.



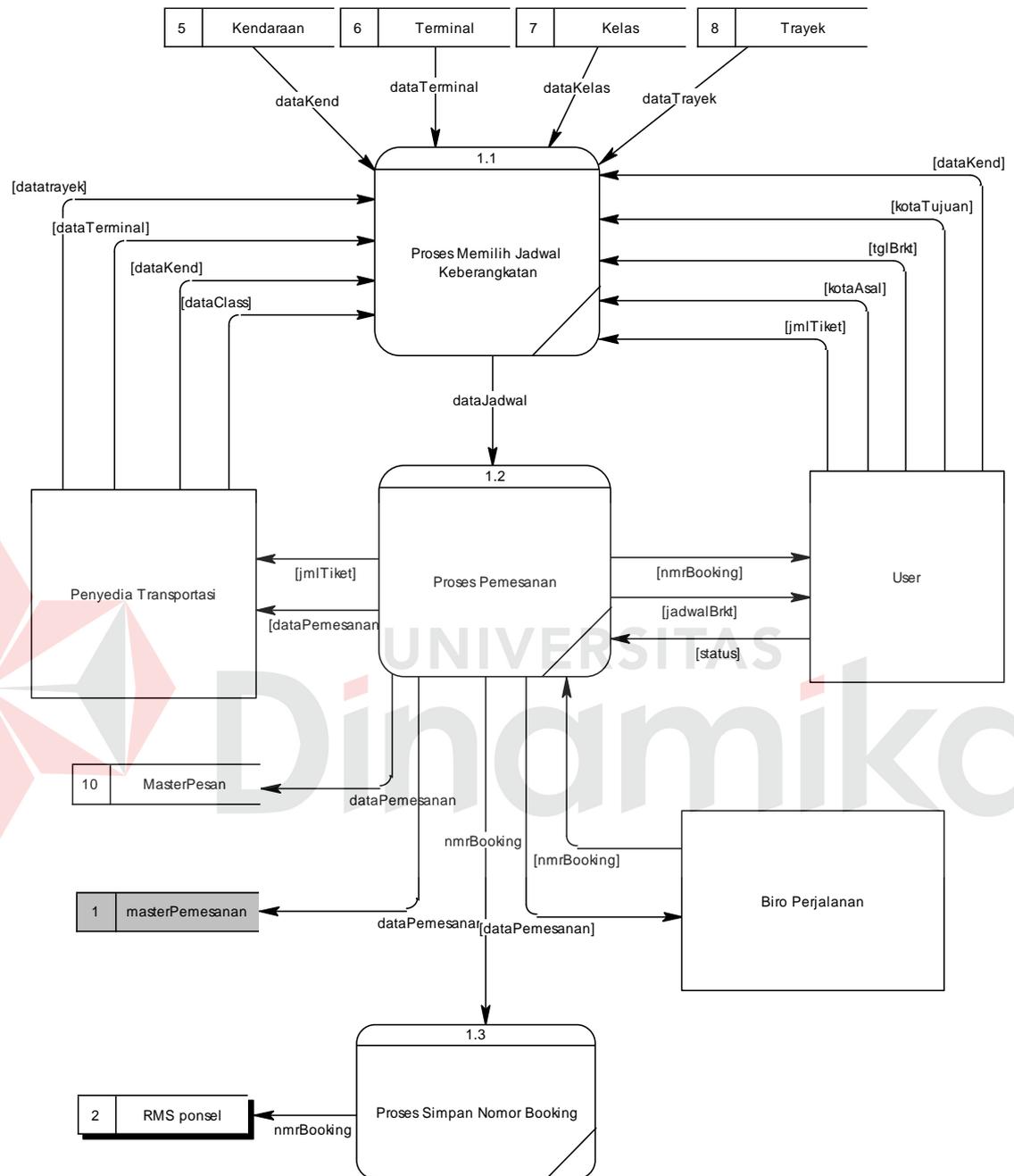
Gambar 3.8 DFD Level 0

b. DFD Level 1



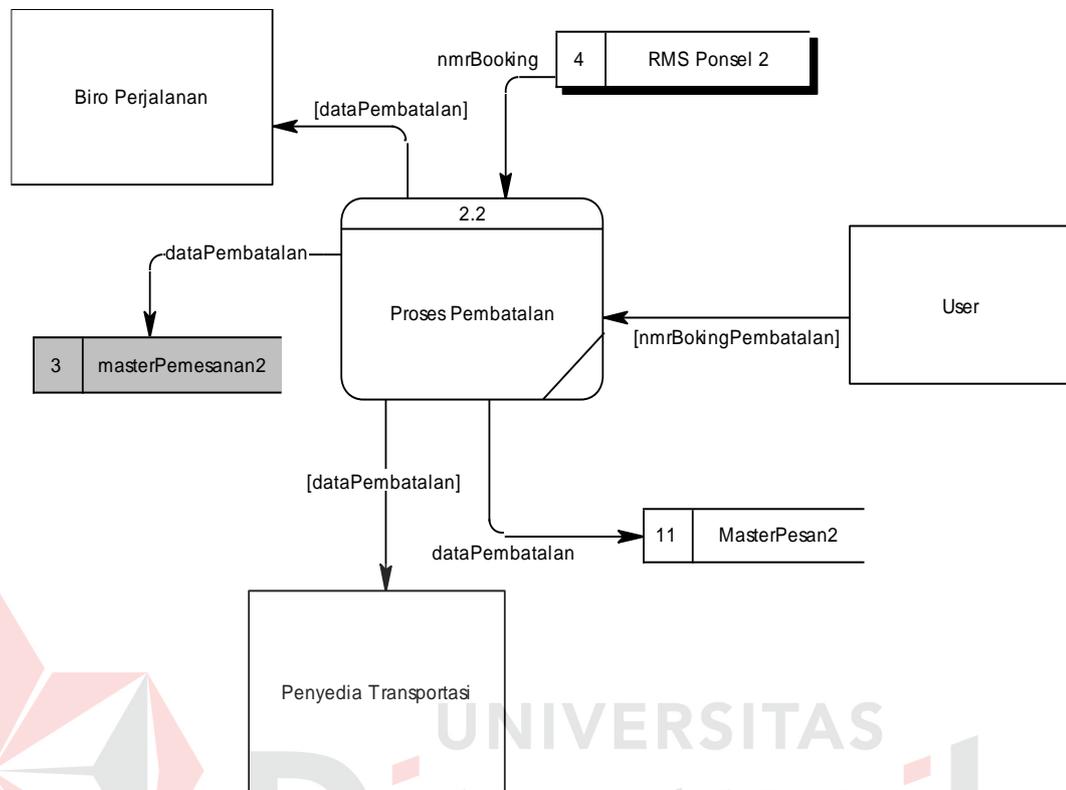
Gambar 3.9 DFD Level 1

c. DFD Level 2 Sub Proses Pemesanan Tiket



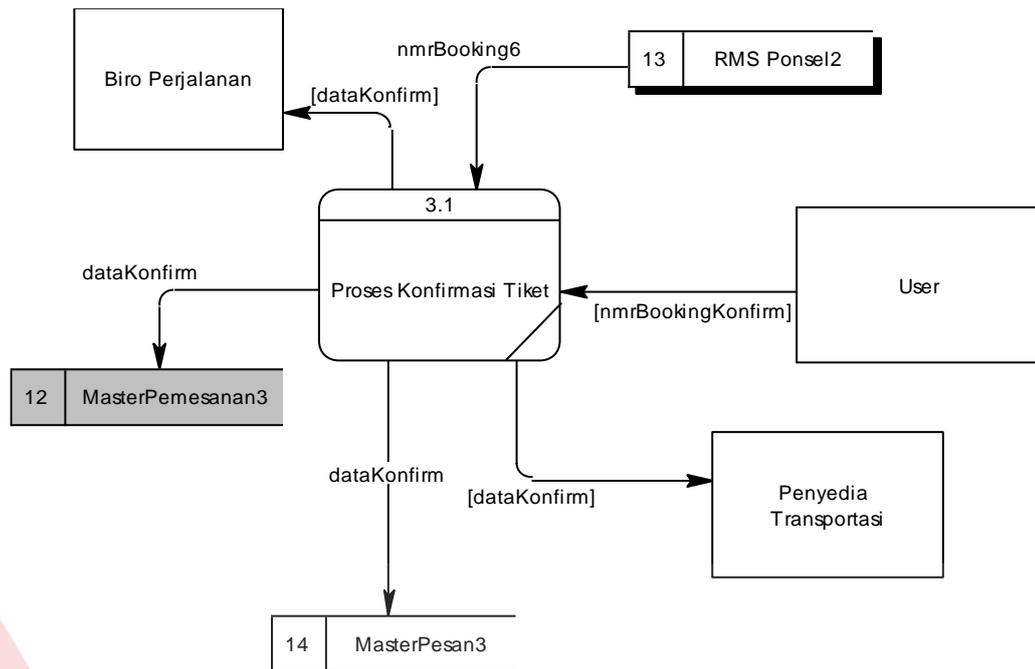
Gambar 3.10 DFD Level 2 Sub Proses Pemesanan Tiket

d. DFD Level 2 Sub Proses Pembatalan Tiket



Gambar 3.11 DFD Level 2 Sub Proses Pembatalan Tiket

e. DFD Level 2 Sub Proses Konfirmasi Tiket



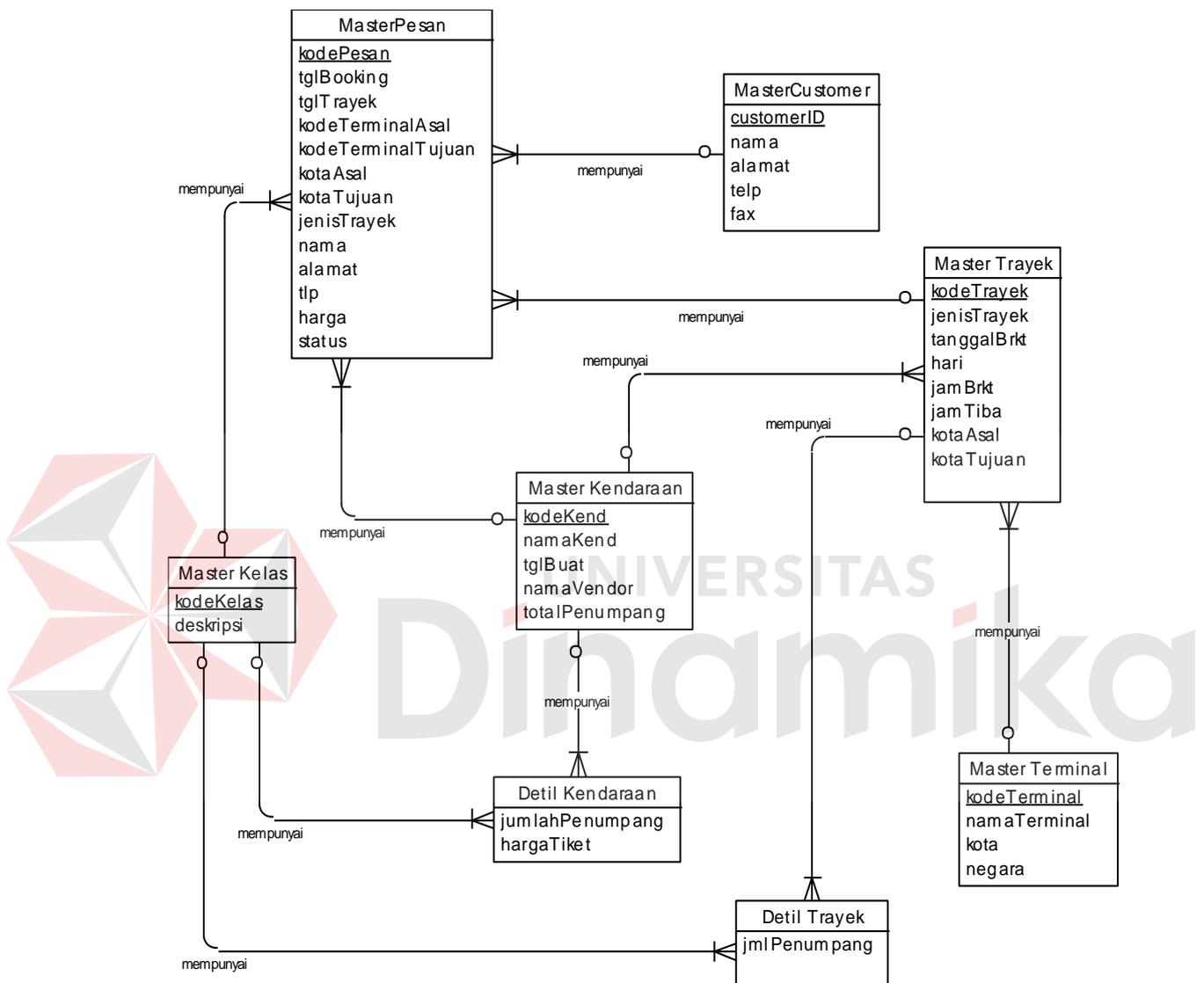
Gambar 3.12 DFD Level 2 Sub Proses Konfirmasi Tiket

3.2.4 Pembuatan ERD

ERD merupakan suatu desain sistem yang digunakan untuk merepresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan database. ERD juga menunjukkan hubungan (relasi) antar tabel. ERD terdiri atas *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)*. Pada sistem ini terdapat 2 macam ERD yaitu ERD dari sisi penyedia transportasi dan biro perjalanan.

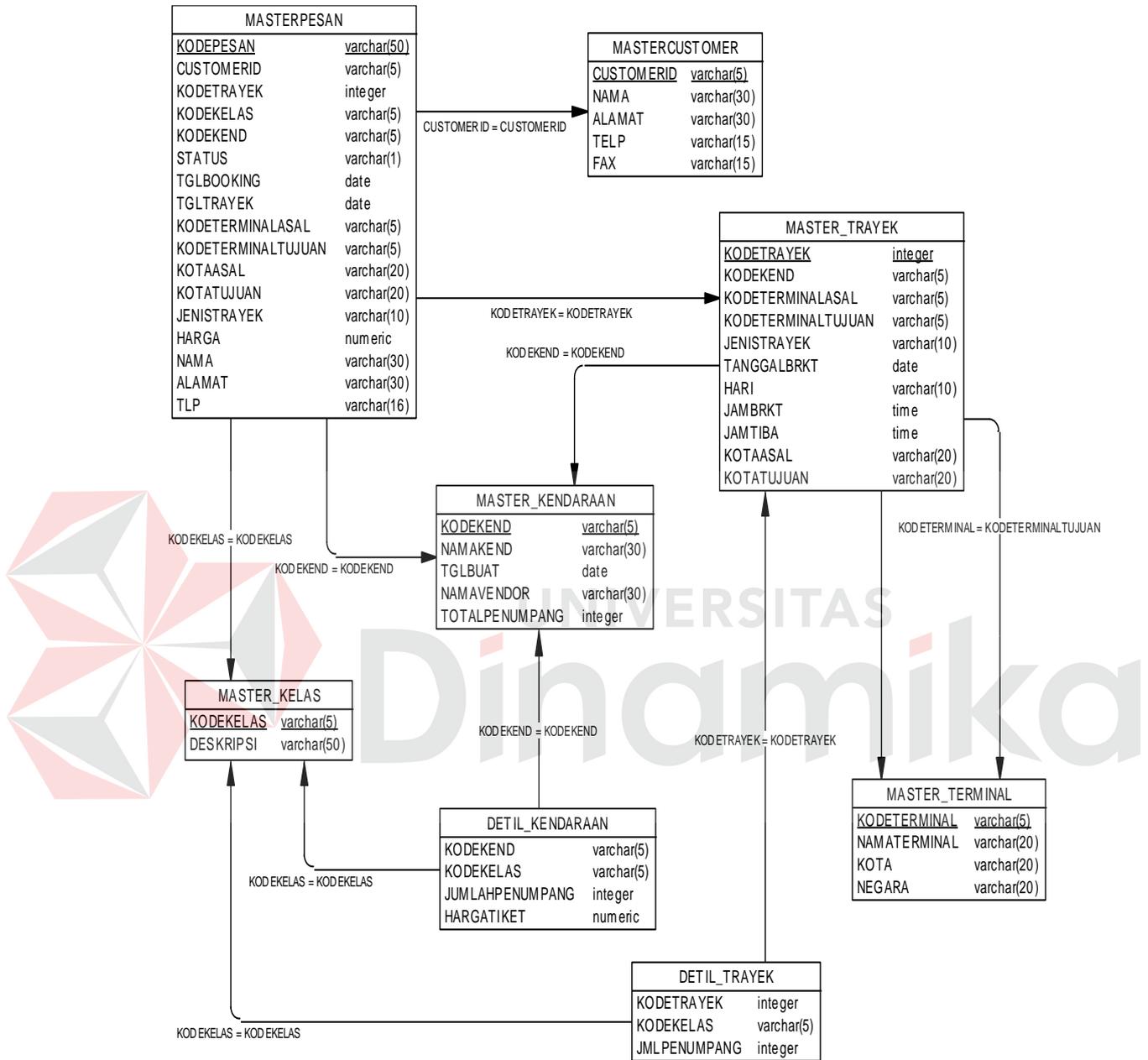
ERD Penyedia Transportasi

a. Conceptual Data Modelling (CDM) Penyedia Transportasi



Gambar 3.13 Conceptual Data Modelling (CDM) Penyedia Transportasi

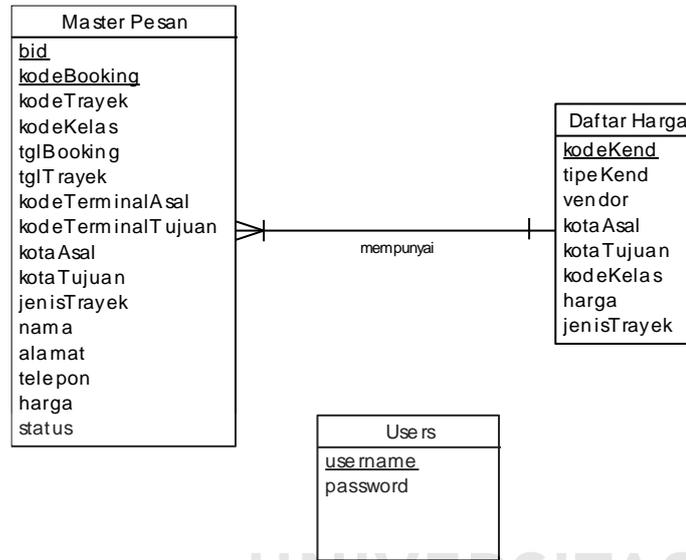
b. Physical Data Modelling (PDM) Penyedia Transportasi



Gambar 3.14 Physical Data Modelling (PDM) Penyedia Transportasi

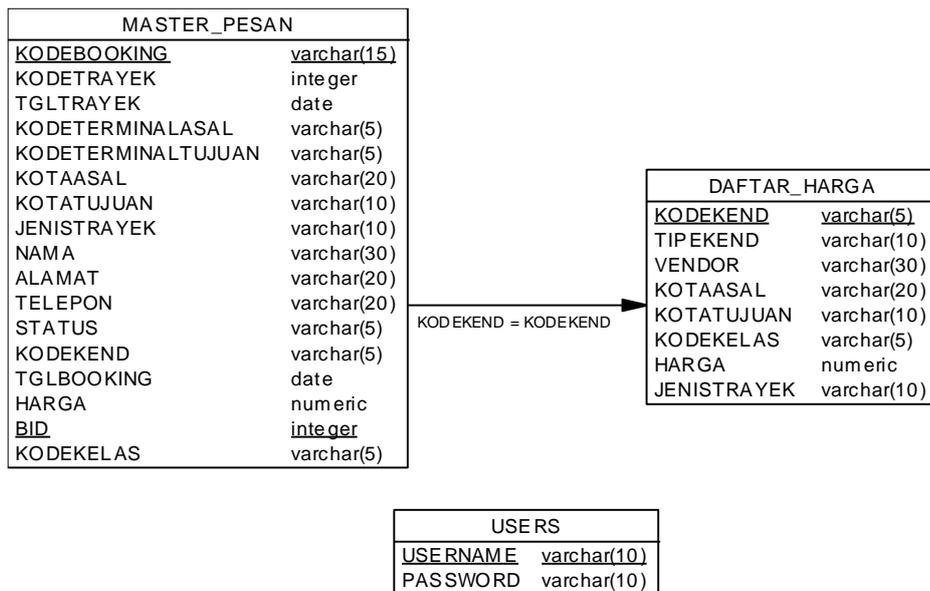
ERD Biro Perjalanan

a. Conceptual Data Modelling (CDM) Biro Perjalanan



Gambar 3.15 Conceptual Data Modelling (CDM) Biro Perjalanan

b. Physical Data Modelling (PDM) Biro Perjalanan



Gambar 3.16 Physical Data Modelling (PDM) Biro Perjalanan

3.2.5 Pembuatan Struktur Database

Dalam hal merancang struktur tabel yang diperlukan, meliputi nama tabel, nama *field*, tipe data, serta data pelengkap seperti *primary key*, *foreign key* dan sebagainya. Rancangan database aplikasi ini terdiri dari tabel-tabel sebagai berikut:

Struktur Database Penyedia Transportasi

1. Tabel MasterKendaraan

Nama Tabel	:	MasterKendaraan
Primary Key	:	kodeKend
Foreign Key	:	-
Fungsi	:	Untuk menyimpan data master kendaraan

Tabel 3.1. Tabel MasterKendaraan

Nama Field	Tipe Data	Key	Keterangan
kodeKend	Text	PK	Kode kendaraan
namaKend	Text		Nama kendaraan
tglBuat	Text		Tanggal pembuatan
namaVendor	Text		Nama perusahaan
totalPenumpang	Integer		Jumlah total kapasitas penumpang

2. Tabel DetailKendaraan

Nama Tabel : DetailKendaraan

Primary Key : kodeKend, kodeKelas

Foreign Key :

Fungsi : Untuk menyimpan data detail kendaraan

Tabel 3.2 Tabel DetailKendaraan

Nama Field	Tipe Data	Key	Keterangan
kodeKend	Text	PK	Kode kendaraan
kodeKelas	Text	PK	Kode kelas kendaraan
jumlahPenumpang	Integer		Jumlah penumpang tiap kelas
hargaTiket	Integer		Harga tiket tiap kelas

3. Tabel MasterTerminal

Nama Tabel : MasterTerminal

Primary Key : kodeTerminal

Foreign Key :

Fungsi : Untuk menyimpan data terminal

Tabel 3.3. Tabel MasterTerminal

Nama Field	Tipe Data	Key	Keterangan
kodeTerminal	Text	PK	Kode terminal
namaTerminal	Text		Nama Terminal
kota	Text		Nama kota
negara	Text		Nama Negara

4. Tabel MasterKelas

Nama Tabel : MasterKelas

Primary Key : kodeKelas

Foreign Key :

Fungsi : Untuk menyimpan data kelas kendaraan

Tabel 3.4. Tabel MasterKelas

Nama Field	Tipe Data	Key	Keterangan
kodeKelas	Text	PK	Kode kelas kendaraan
deskripsiKelas	Text		Keterangan tiap kelas

5. Tabel MasterTrayek

Nama Tabel : MasterTrayek

Primary Key : kodeTrayek, kodeKend

Foreign Key : kodeTerminalAsal, kodeTerminalTujuan

Fungsi : Untuk menyimpan data trayek

Tabel 3.5. Tabel MasterTrayek

Nama Field	Tipe Data	Key	Keterangan
kodeTrayek	Text	PK	Kode trayek
kodeKend	Text	PK	Kode kendaraan
jenisTrayek	Text		Jenis trayek apakah rutin atau ekstra
tglBooking	Date		Tanggal pemesanan
tglTrayek	Date		Tanggal trayek
jamBrkt	Time		Jam berangkat
jamTiba	Time		Jam tiba
hari	Text		Hari
kodeTerminalAsal	Text	FK	Kode terminal asal
kodeTerminalTujuan	Text	FK	Kode terminal tujuan
kotaAsal	Text		Kota asal
kotaTujuan	Text		Kota tujuan

6. Tabel DetailTrayek

Nama Tabel : DetailTrayek

Primary Key : kodeTrayek, kodeKelas

Foreign Key :

Fungsi : Untuk menyimpan data detail trayek

Tabel 3.6. Tabel DetailTrayek

Nama Field	Tipe Data	Key	Keterangan
kodeTrayek	Text	PK	Kode trayek
kodeKelas	Text	PK	Kode kelas
jumlahPenumpang	Integer		Jumlah penumpang pada trayek bersangkutan

7. Tabel MasterCustomer

Nama Tabel : MasterCustomer

Primary Key : customerID

Foreign Key :

Fungsi : Untuk menyimpan data biro perjalanan

Tabel 3.7. Tabel MasterCustomer

Nama Field	Tipe Data	Key	Keterangan
customerID	Text	PK	Kode biro perjalanan
nama	Text		Nama biro perjalanan
alamat	Text		Alamat
telp	Text		Nomor telepon
fax	Text		Nomor fax

8. Tabel MasterPesan

Nama Tabel	:	MasterPesan
Primary Key	:	kodePemesanan
Foreign Key	:	kodeKend, kodeTrayek, kodeKelas, customerID kodeTerminalAsal, kodeTerminalTujuan
Fungsi	:	Untuk menyimpan data pemesanan

Tabel 3.8 Tabel Pemesanan

Nama Field	Tipe Data	Key	Keterangan
kodePemesanan	Text	PK	Kode pemesanan
kodeTrayek	Text	FK	Kode trayek
kodeKelas	Text	FK	Kode kelas
kodeKend	Text	FK	Kode kendaraan
tglBooking	Date		Tanggal pemesanan
tglTrayek	Date		Tanggal trayek
kodeTerminalAsal	Text	FK	Kode terminal asal
kodeTerminalTujuan	Text	FK	Kode terminal tujuan
kotaAsal	Text		Kota asal
kotaTujuan	Text		Kota tujuan
jenisTrayek	Text		Jenis trayek apakah rutin atau ekstra
customerID	Text	FK	Kode biro perjalanan
nama	Text		Nama pemesan
alamat	Text		Alamat
tlp	Text		Nomor telepon
harga	Integer		Harga
status	Text		Status pemesanan

Struktur Database Penyedia Transportasi

9. Tabel MasterPesan

Nama Tabel	:	MasterPesan
Primary Key	:	kodeBooking
Foreign Key	:	bid, kodeTrayek, kodeKend, kodeTerminalAsal, kodeTerminalTujuan
Fungsi	:	Untuk menyimpan data pemesanan

Tabel 3.9 Tabel MasterPesan

Nama Field	Tipe Data	Key	Keterangan
bid	Text	PK	Kode bantuan
kodeBooking	Text	PK	Kode booking
kodeTrayek	Text	FK	Kode trayek
kodeKend	Text	FK	Kode kendaraan
tglBooking	Date		Tanggal pemesanan
tglTrayek	Date		Tanggal trayek
kodeTerminalAsal	Text	FK	Kode terminal asal
kodeTerminalTujuan	Text	FK	Kode terminal tujuan
kotaAsal	Text		Kota asal
kotaTujuan	Text		Kota tujuan
jenisTrayek	Text		Jenis trayek apakah rutin atau ekstra
nama	Text		Nama pemesan
alamat	Text		Alamat
tlp	Text		Nomor telepon
harga	Integer		Harga
status	Text		Status pemesanan

10. Tabel DaftarHarga

Nama Tabel	:	DaftarHarga
Primary Key	:	kodeKend
Foreign Key	:	kodeKelas
Fungsi	:	Untuk menyimpan data daftar harga tiap kendaraan

Tabel 3.10 Tabel DaftarHarga

Nama Field	Tipe Data	Key	Keterangan
kodeKend	Text	PK	Kode kendaraan
tipeKend	Text		Tipe kendaraan
vendor	Text		Nama perusahaan
kotaAsal	Text		Kota Asal
kotaTujuan	Text		Kota tujuan
kodeKelas	Text	FK	Kode kelas
harga	Integer		Harga
jenisTrayek	Text		Jenis trayek apakah rutin atau ekstra

11. Tabel Users

Nama Tabel	:	Users
Primary Key	:	username
Foreign Key	:	
Fungsi	:	Untuk menyimpan data administrator

Tabel 3.11 Tabel Users

Nama Field	Tipe Data	Key	Keterangan
username	Text	PK	Username
password	Text		Password
nama	Text		Nama pengguna

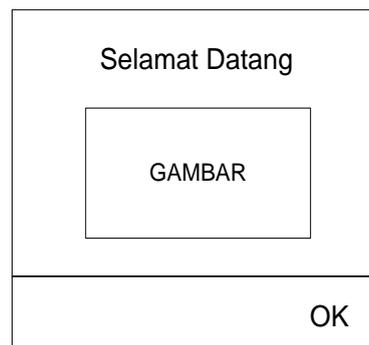
3.3 Perancangan Input Output

Rancangan input yang digunakan pada sistem ini dibuat berdasarkan konsep interaksi manusia dan komputer. Rancangan input output sistem ini terdiri atas dialog proses demi proses yang dilakukan selama pengguna memberikan input dan mendapatkan output sistem. Rancangan input output ini meliputi :

3.3.1 Rancangan Input Output Client Pada Telepon seluler

a. Tampilan Pembuka

Tampilan pembuka muncul pada saat aplikasi dijalankan. Jika ditekan tombol Ok maka aplikasi berpindah ke tampilan berikutnya.



Gambar 3.17 Rancangan Tampilan Pembuka

b. Tampilan Menu Utama

Tampilan berikut merupakan menu utama yang terdapat pada aplikasi sistem pemesanan tiket menggunakan web service.

MENU	
PEMESANAN	
PEMBATALAN	
LIHAT BOOKING	
KONFIRMASI	
KELUAR	OK

Gambar 3.18 Rancangan Tampilan Menu Utama

c. Tampilan Menu Pemesanan

Pada tampilan menu pemesanan, user memilih tanggal keberangkatan dan tipe kendaraan. Tipe kendaraan tersebut berupa pesawat dan kapal laut.

PEMESANAN	
Pilih Tanggal :	
<input type="text" value="Date Field"/>	
Pilih Tipe Kendaraan :	
<input checked="" type="radio"/> Pesawat <input type="radio"/> Kapal Laut	
KEMBALI	LANJUT

Gambar 3.19 Rancangan Tampilan Menu Pemesanan

d. Tampilan Jurusan

Misalkan user memilih perjalanan menggunakan pesawat, maka setelah memilih kendaraan pesawat, user memilih jurusan keberangkatan.

TUJUAN PERJ	
Pilih kota perjalanan :	
<input checked="" type="radio"/> Surabaya-Jakarta <input type="radio"/> Jakarta-Bali <input type="radio"/> Dst...	
KEMBALI	LANJUT

Gambar 3.20 Rancangan Tampilan Jurusan

e. Tampilan Info Perjalanan

Setelah memilih jurusan, user memilih jadwal keberangkatan tiap pesawat untuk rute yang dipilih.

INFO PERJALANAN	
Jadwal Penerbangan :	
<input checked="" type="radio"/> GARUDA – GIA001 Selasa-09.00 <input checked="" type="radio"/> GARUDA – GIA003 Kamis-13.00 <input checked="" type="radio"/> MANDALA – MAN001 Selasa-09.00 <input type="radio"/> Dst...	
KEMBALI	LANJUT

Gambar 3.21 Rancangan Tampilan Info Perjalanan

f. Tampilan Status Seat

Setelah itu akan muncul tampilan status seat yang tersedia dari jenis pesawat yang dipilih.

STATUS SEAT	
Pilih seat sesuai Class :	
<input checked="" type="radio"/>	Bisnis-sisa kursi:15- Rp.200.000
<input type="radio"/>	Eksekutif-sisa kursi:20- Rp.325.000
<input type="radio"/>	Dst...
BACK	LANJUT

Gambar 3.22 Rancangan Tampilan Status Seat

g. Tampilan Entry Data Penumpang

Setelah memilih kelas, user mengisi data diri meliputi jumlah tiket yang dipesan. Textbox nama akan muncul secara dinamis sesuai dengan jumlah tiket yang diinputkan.

DATA PENUMPANG	
Jumlah Tiket :	<input type="text" value="2"/>
Nama 1 :	<input type="text" value="Lili"/>
Nama 2 :	<input type="text" value="Putu"/>
Alamat :	<input type="text" value="Medokan"/>
Nomor Telepon :	<input type="text" value="8701910"/>
BACK	LANJUT

Gambar 3.23 Rancangan Tampilan Entry Data Penumpang

h. Tampilan Informasi

Setelah user menginputkan seluruh data, maka akan muncul tampilan konfirmasi yang berisi info pemesanan.

INFORMASI	
Anda telah berhasil melakukan pemesanan tiket pesawat GIA001 DPS-SBY Senin, 14 Mei 2006 10:00 2 class bisnis. Tekan Book untuk booking saja. Tekan Issued jika pasti brkt. Pemesanan yg Issued tidak dapat dibatalkan	
BOOK	ISSUED

Gambar 3.24 Rancangan Konfirmasi

i. Tampilan Pembatalan

Apabila user ingin membatalkan pesanan, maka user harus memilih nomor booking yang ingin dibatalkan.

No Booking	
Pilih nomor booking :	
<input checked="" type="radio"/> 12	<input type="radio"/> 13
BACK	BATALKAN

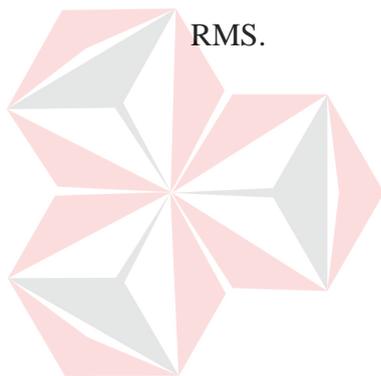
Gambar 3.25 Rancangan Pembatalan

Detail No Booking	
Nama : Lili Alamat : Medokan Kode Trayek : 3 Kode Kendaraan : GIA001 Tanggal Trayek : 5/14/2006 Kode Terminal Asal : JUA Kode TerminalTuj : SOEHAT Kota Asal : Surabaya Kota Tujuan : Jakarta	
BACK	BATALKAN

Gambar 3.26 Rancangan Pembatalan2

j. Tampilan Lihat Nomor Booking

Disini ditampilkan nomor-nomor booking yang telah tersimpan pada



RMS.

Lihat No Booking	
Daftar nomor booking :	
<input checked="" type="radio"/> 12	
<input type="radio"/> 13	
BACK	DETAIL

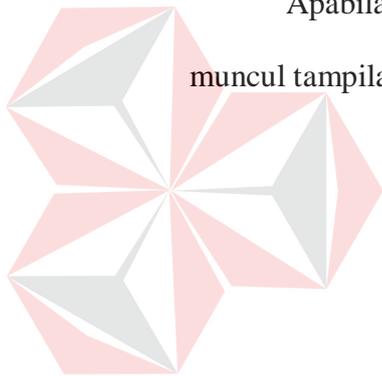
Gambar 3.27 Rancangan Lihat Nomor Booking

Detail No Booking
Nama : Lili Alamat : Medokan Kode Trayek : 3 Kode Kendaraan : GIA001 Tanggal Trayek : 5/14/2006 Kode Terminal Asal : JUA Kode TerminalTuj : SOEHAT Kota Asal : Surabaya Kota Tujuan : Jakarta
BACK

Gambar 3.28 Rancangan Detail Nomor Booking

k. Tampilan Konfirmasi Tiket

Apabila user ingin mengkonfirmasi pemesanan tiket maka akan muncul tampilan berikut ini.



No Booking
Pilih nomor booking yang akan di konfirmasi :
<input checked="" type="radio"/> 12 <input type="radio"/> 13
BACK KONFIRM

Gambar 3.29 Rancangan Konfirmasi Tiket

3.3.2 Rancangan Input Output Web Untuk Administrator

Agar proses bisnis berlangsung dengan baik, maka diperlukan perancangan tampilan bagi administrator untuk maintenance data pemesanan yang telah masuk.

a. Tampilan Login

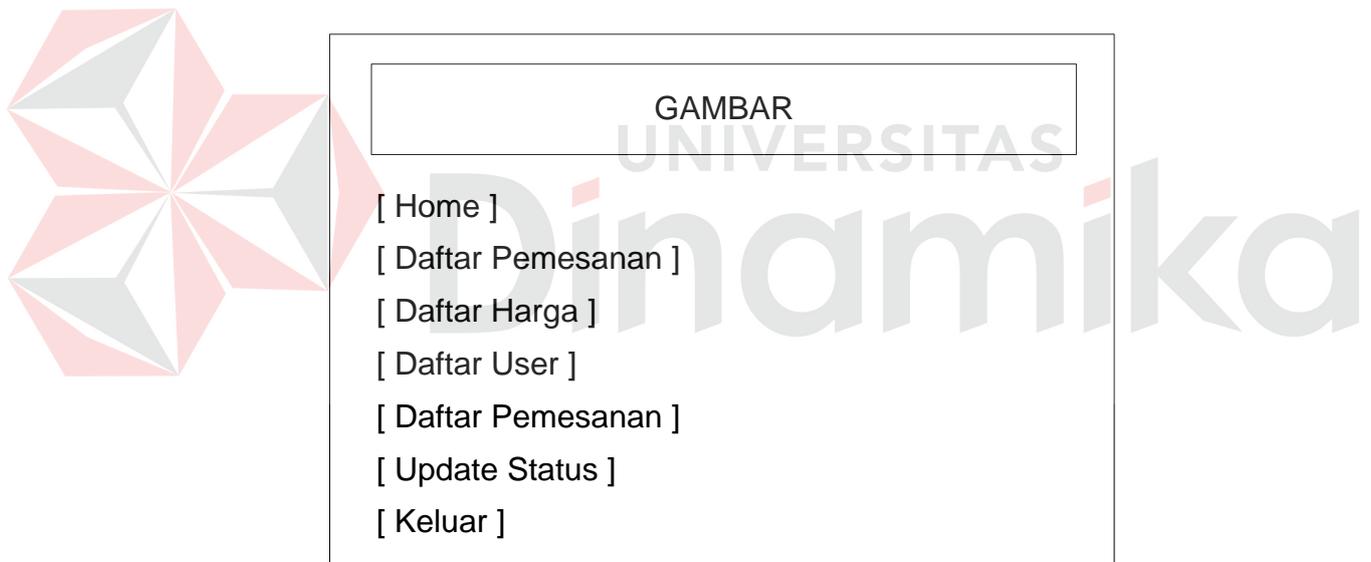


Username :

Password :

Gambar 3.30 Rancangan Login

b. Tampilan Web Halaman Utama



Gambar 3.31 Rancangan Web Halaman Utama

c. Tampilan Web Daftar Pemesanan

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Daftar Pemesanan

Kode Booking	Tgl Pesan	Kode Trayek	Tgl Brkt	Jenis Trayek	Nama	Alamat	Harga	Status

Gambar 3.32 Rancangan Web Daftar Pemesanan

d. Tampilan Web Update Status

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Update Status

Masukkan Kode Booking :

Set Status : O - Order C - Cancel

Gambar 3.33 Rancangan Web Update Status

e. **Tampilan Web Daftar Harga**

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Daftar Harga

[Tambah Data]

Kode Kendaraan	Tipe Kendaraan	Vendor	Kode Kelas	Harga	Jenis Trayek

Gambar 3.34 Rancangan Web Daftar Harga

f. **Tampilan Web Tambah Data Harga**

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Tambah Data Harga

Kode Kendaraan :

Tipe Kendaraan :

Vendor :

Kelas :

Harga :

Jenis Trayek : Rutin Ekstra

Gambar 3.35 Rancangan Web Tambah Data Harga

g. Tampilan Web Daftar User

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Daftar User

[Tambah Data]

Username	Password	Nama

Gambar 3.36 Rancangan Web Daftar User

h. Tampilan Web Tambah Data User

GAMBAR

[Home] [Daftar Pemesanan] [Daftar Harga] [Daftar User] [Daftar Pemesanan] [Update Status] [Keluar]

Tambah Data User

Username :

Password :

Verify Password :

Nama :

Gambar 3.37 Rancangan Web Tambah Data User

BAB IV

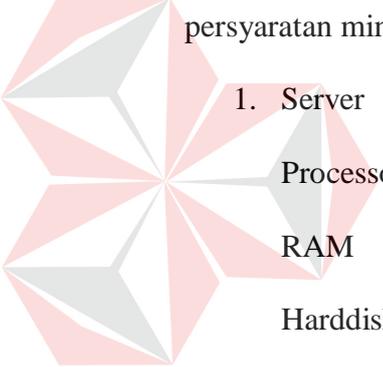
IMPLEMENTASI DAN EVALUASI

4.1 Kebutuhan Sistem

Aplikasi sistem pemesanan tiket menggunakan web service yang dijalankan pada telepon seluler ini membutuhkan perangkat keras dan perangkat lunak dengan spesifikasi tertentu agar dapat berjalan dengan baik.

4.1.1 Kebutuhan perangkat keras

Aplikasi sistem pemesanan tiket menggunakan web service membutuhkan persyaratan minimal perangkat keras sebagai berikut :

- 
1. Server
 - Processor : 1 GHz
 - RAM : 256 MB
 - Harddisk : 10 GB

2. Telepon seluler

Kebutuhan minimum telepon seluler untuk dapat menggunakan aplikasi sistem pemesanan tiket adalah dengan konfigurasi MIDP 2.0 dan CLDC 1.0. Untuk konektivitas menggunakan GPRS. Sebagai contoh telepon seluler yang mendukung MIDP 2.0 dan CLDC 1.0 adalah Nokia Series 40, 60 dan 80 (Nokia 6255, Nokia 7610 dan Nokia 9500) atau Siemens generasi 65 (Siemens CX70, Siemens 65).

4.1.2 Kebutuhan perangkat lunak

Kebutuhan perangkat lunak untuk aplikasi sistem pemesanan tiket menggunakan web service ini adalah :

1. Microsoft Visual Studio .NET 2003
2. Java 2 Standard Edition 1.4.2
3. Java 2 Micro Edition 2.2
4. Text Editor EditPlus
5. Database Microsoft Access XP
6. IIS web server

4.2 Pembuatan Program

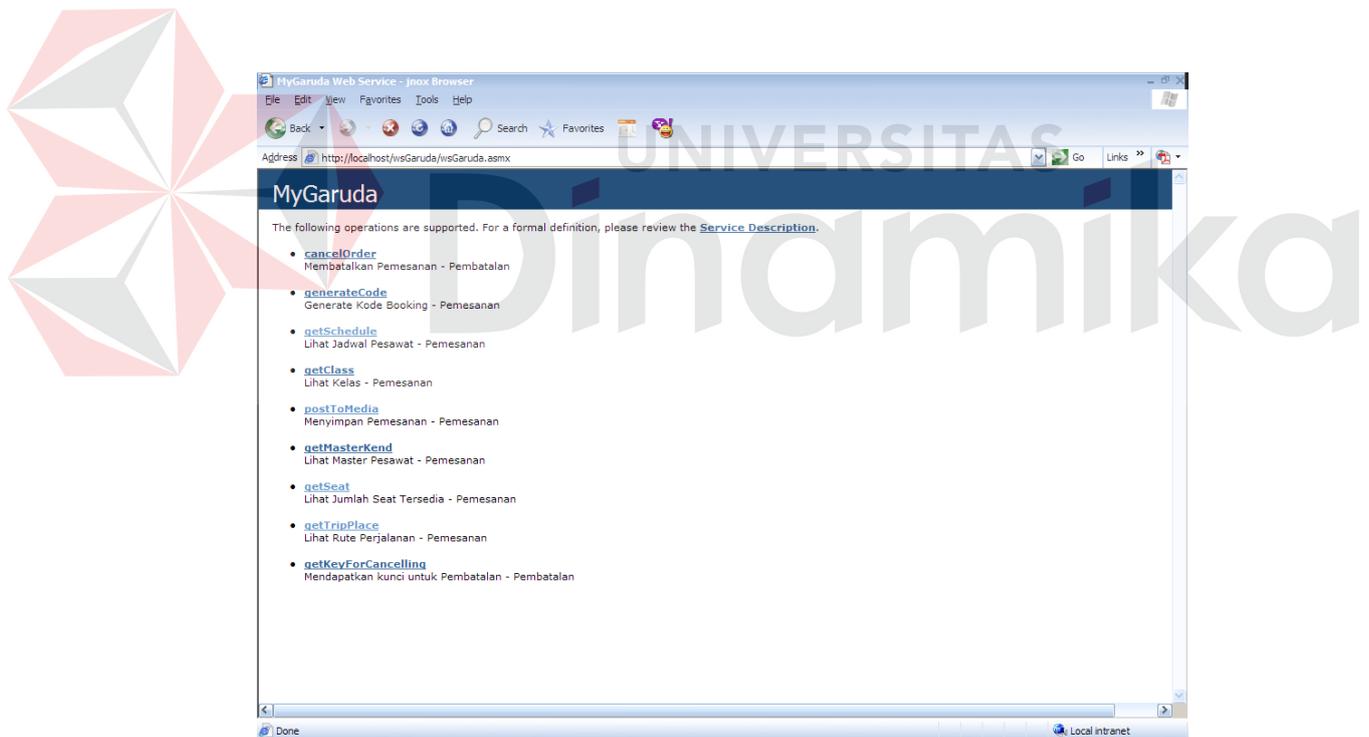
Aplikasi web service dibuat dengan ASP.NET Web Service dan disimpan dengan file berekstensi asmx. Aplikasi web service ini terdiri dari beberapa fungsi yang dapat dipanggil oleh aplikasi pemesanan tiket pada telepon seluler. Fungsi-fungsi tersebut akan mengirimkan data yang diperlukan dalam format XML. Web service sistem pemesanan tiket diletakkan pada web server. Web server yang digunakan adalah *Internet Information Services* (IIS). Sedangkan penulisan program aplikasi pemesanan tiket pada telepon seluler dibuat dengan menggunakan text editor EditPlus dan dijalankan menggunakan J2ME Wireless Toolkit. Tahap akhir pembuatan program MIDlet adalah pembuatan package berekstensi jad (Java Application Descriptor) dan jar (Java Archive) sehingga aplikasi MIDlet siap di-deploy pada telepon seluler. Untuk memasukkan package jad dan jar ke dalam telepon seluler dapat dilakukan dengan beberapa cara misalnya menggunakan kabel data, *cardreader*, *infra red*, *bluetooth*, atau melalui OTA Provisioning. Data dalam format XML yang didapat dari fungsi-fungsi

dalam web service akan diterjemahkan menjadi format String menggunakan XML *parser*. Dalam aplikasi ini XML parser yang digunakan adalah KXML.

Penulisan kode program dan perancangan tampilan aplikasi web dibuat menggunakan ASP.NET Web Application. Kode program aplikasi web ditulis dan disimpan dengan file berekstensi aspx. Database server yang digunakan oleh aplikasi web adalah Microsoft Access. Aplikasi web dapat dijalankan pada *internet browser* seperti Internet Explorer untuk melakukan pemeliharaan data.

4.3 Tampilan Aplikasi Web Services

4.3.1 Web services penyedia transportasi



Gambar 4.1 Web service penyedia transportasi

Fungsi-fungsi yang tersedia pada web service penyedia transportasi adalah sebagai berikut :

1. cancelOrder

Deskripsi : Mengupdate status pemesanan tiket

Parameter : kdBooking (String), status (String)

ReturnType : String

2. getSchedule

Deskripsi : Melihat jadwal keberangkatan

Parameter : kotaAsal (String), kotaTujuan (String)

ReturnType : Schedules (DataSet)

3. getClass

Deskripsi : Melihat kelas-kelas yang tersedia

Parameter : kodeTrayek (String)

ReturnType : Kelas (DataSet)

4. postToMedia

Deskripsi : Menyimpan pemesanan tiket

Parameter : kdBookTA (String), noTelp (String), kodeTrayek (String),

kodeKelas (String), kodeKendaraan (String), tanggalTrayek (String),

kodeTermAsal (String), kodeTermTujuan (String), kotaAsal (String),

kotaTujuan (String), jenisTrayek (String), tanggalPesan (String), nama

(String), alamat (String), status (String)

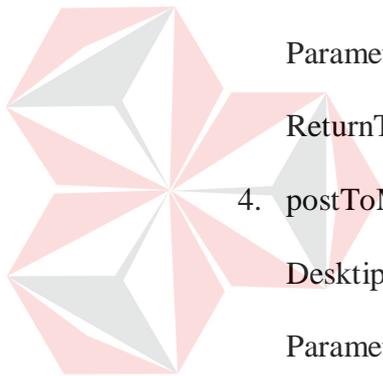
ReturnType : String

5. getMasterKend

Deskripsi : Melihat master kendaraan

Parameter : -

ReturnType : MasterKendaraan (DataSet)



6. getSeat

Deskripsi : Melihat jumlah seat tersedia

Parameter : kdTrayek (String)

ReturnType : SeatTersedia (DataSet)

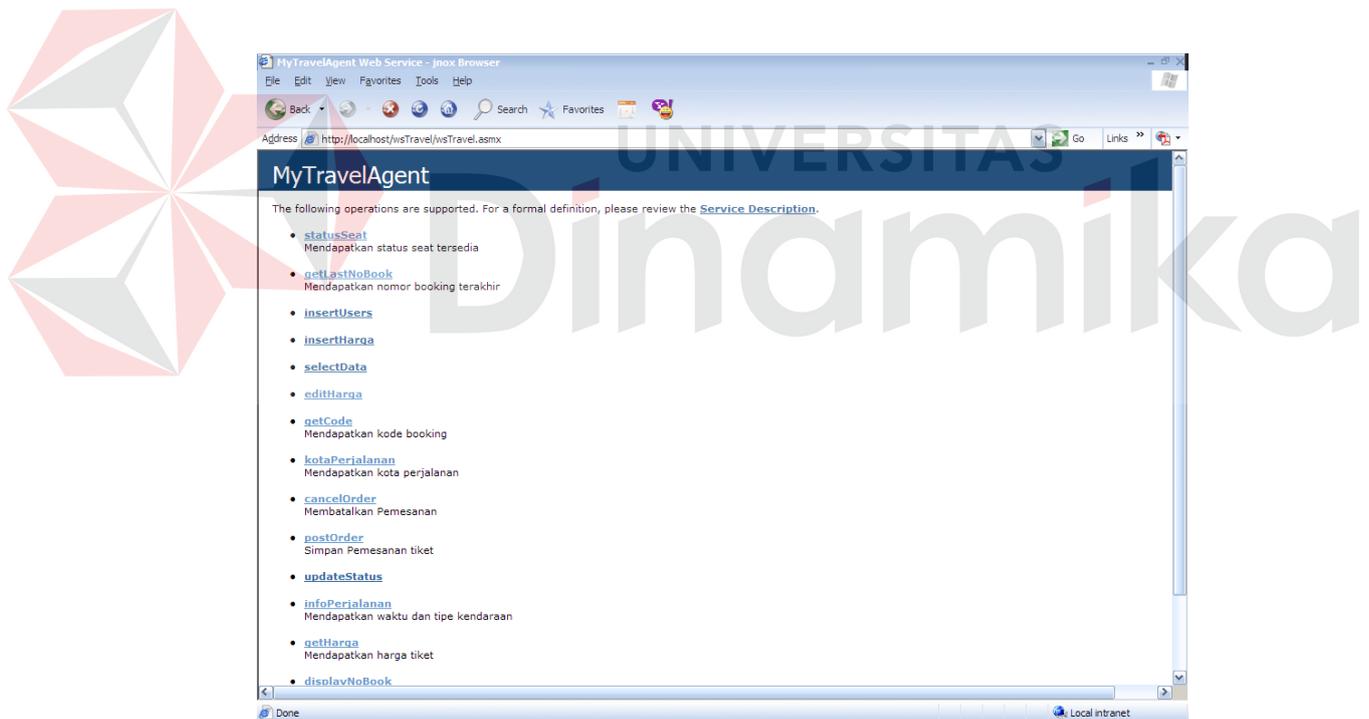
7. getTripPlace

Deskripsi : Melihat rute-rute perjalanan

Parameter : -

ReturnType : KotaPerjalanan (DataSet)

4.3.2 Web services biro perjalanan



Gambar 4.2 Web service biro perjalanan

Fungsi-fungsi yang tersedia pada web service biro perjalanan adalah sebagai berikut :

1. statusSeat

Deskripsi : Mendapatkan status seat yang tersedia. Memanggil fungsi `getSeat()` pada web service penyedia transportasi

Parameter : jenis (Integer), maskapai (Integer), kodeTrayek (String)

ReturnType : DataSet

2. kotaPerjalanan

Deskripsi : Mendapatkan rute-rute yang tersedia. Memanggil fungsi `getTripPlace()` pada web service penyedia transportasi

Parameter : jenis (Integer), maskapai (Integer)

ReturnType : DataSet

3. cancelOrder

Deskripsi : Mengupdate status pemesanan tiket. Memanggil fungsi `getKeyForCancelling()` pada web service penyedia transportasi

Parameter : maskapaiState (Integer), kdBooking (String), status (String)

ReturnType : String

4. postOrder

Deskripsi : Menyimpan pemesanan tiket. Memanggil fungsi `postToMedia()` pada web service penyedia transportasi

Parameter : maskapaiState (Integer), kodeTrayek (String), kodeKendaraan (String), kodeKelas (String), tanggalTrayek (String), kodeTermAsal (String), kodeTermTujuan (String), kotaAsal (String), kotaTujuan (String), jenisTrayek (String), nama (String), alamat (String), telp (String), status (String)

Return Type : String

5. infoPerjalanan

Deskripsi : Mendapatkan jadwal keberangkatan. Memanggil fungsi `getSchedule()` pada web service penyedia transportasi

Parameter : jenis (Integer), maskapai (Integer), kotaAsal (String), kotaTujuan (String)

Return Type : DataSet

6. displayNoBook

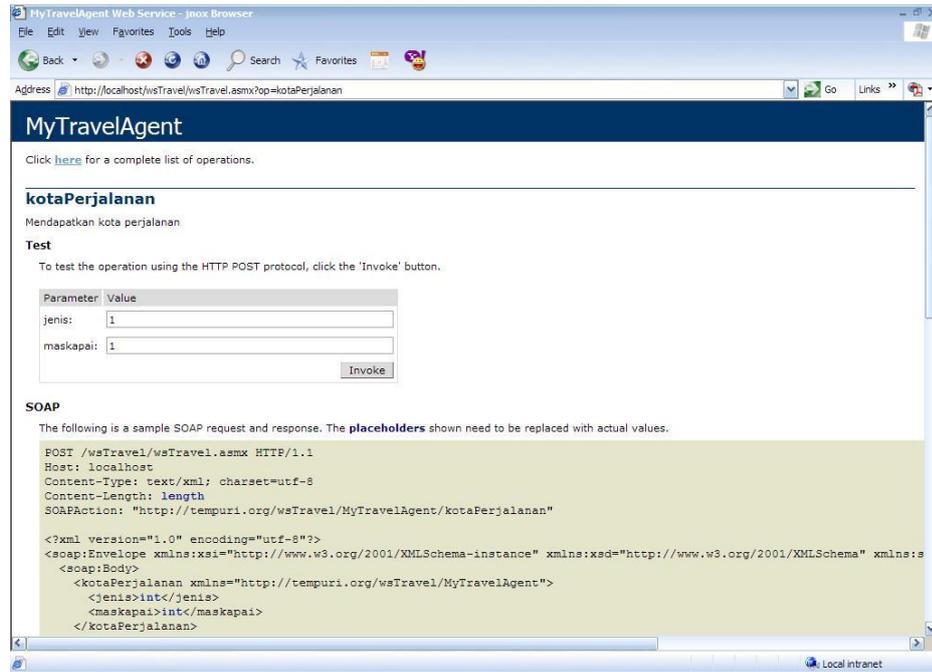
Deskripsi : Melihat detail kode booking

Parameter : kdBooking (String)

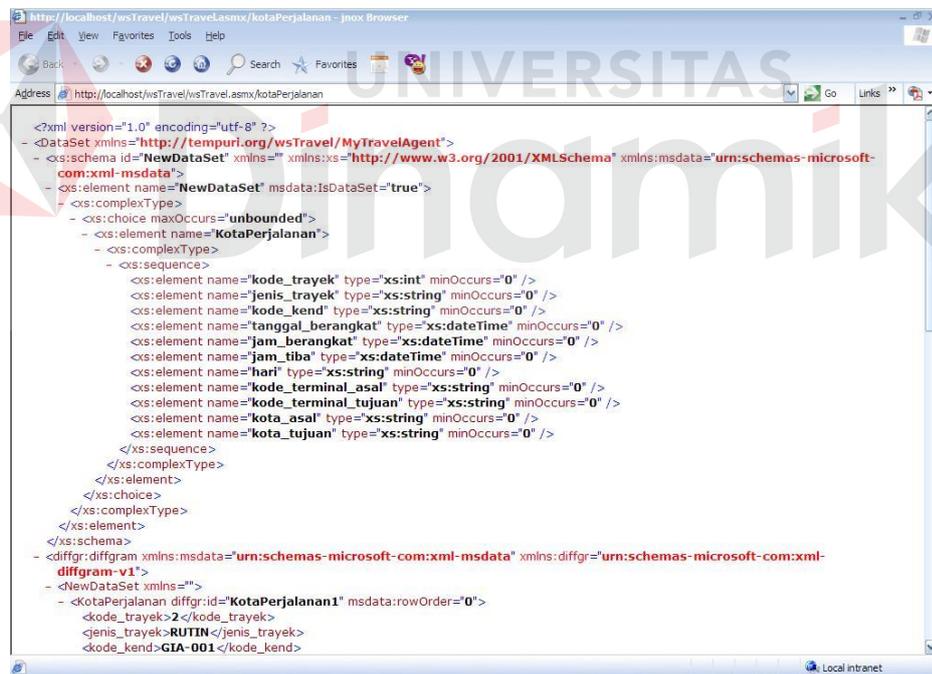
Return Type : DetailBooking (DataSet)

Fungsi-fungsi tersebut merupakan fungsi utama dalam proses pemesanan tiket. Diluar fungsi-fungsi tersebut terdapat beberapa fungsi yang lain, yang digunakan oleh administrator untuk melakukan maintenance data pemesanan tiket.

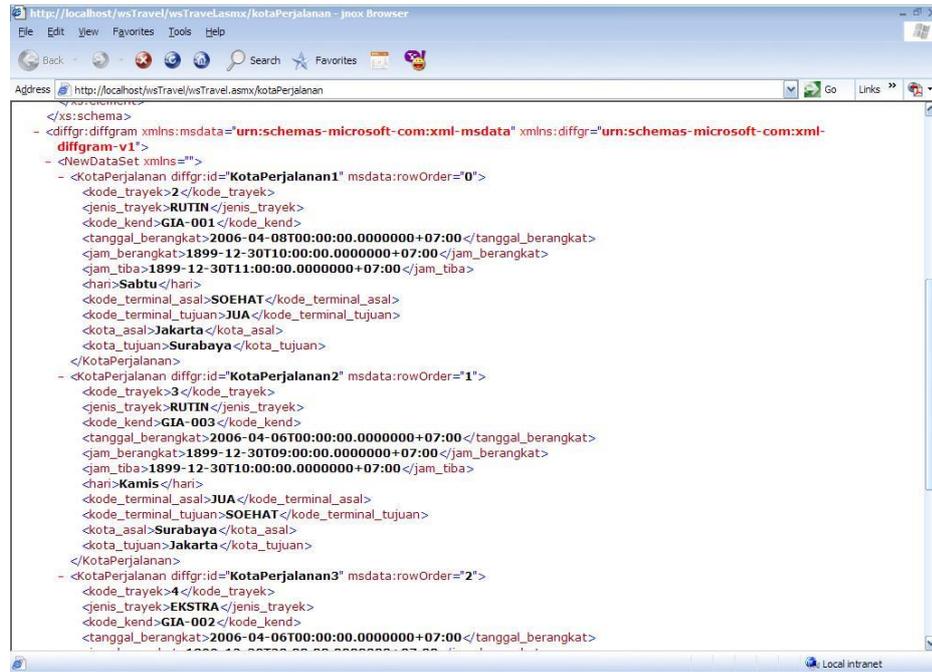
Berikut ini adalah salah satu contoh tampilan data yang dihasilkan oleh fungsi yang terdapat dalam web service. Contoh yang dipakai adalah fungsi `kotaPerjalanan()` pada web service biro perjalanan.



Gambar 4.3 Contoh pemanggilan fungsi-fungsi pada web service



Gambar 4.4 Data yang dihasilkan dari pemanggilan fungsi web service1



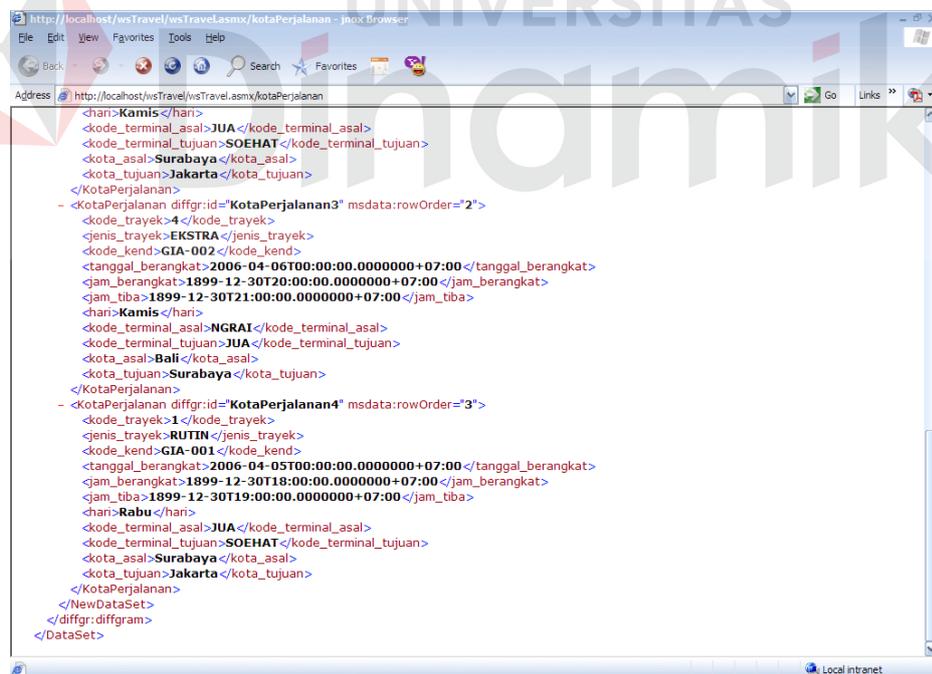
```

</xs:schema>
- <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-
diffgram-v1">
- <NewDataSet xmlns="">
- <KotaPerjalanan diffgr:id="KotaPerjalanan1" msdata:rowOrder="0">
- <kode_trayek>2</kode_trayek>
- <jenis_trayek>RUTIN</jenis_trayek>
- <kode_kend>GIA-001</kode_kend>
- <tanggal_berangkat>2006-04-08T00:00:00.000000+07:00</tanggal_berangkat>
- <jam_berangkat>1899-12-30T10:00:00.000000+07:00</jam_berangkat>
- <jam_tiba>1899-12-30T11:00:00.000000+07:00</jam_tiba>
- <hari>Sabtu</hari>
- <kode_terminal_asal>SOEHAT</kode_terminal_asal>
- <kode_terminal_tujuan>JUA</kode_terminal_tujuan>
- <kota_asal>Jakarta</kota_asal>
- <kota_tujuan>Surabaya</kota_tujuan>
- </KotaPerjalanan>
- <KotaPerjalanan diffgr:id="KotaPerjalanan2" msdata:rowOrder="1">
- <kode_trayek>3</kode_trayek>
- <jenis_trayek>RUTIN</jenis_trayek>
- <kode_kend>GIA-003</kode_kend>
- <tanggal_berangkat>2006-04-06T00:00:00.000000+07:00</tanggal_berangkat>
- <jam_berangkat>1899-12-30T09:00:00.000000+07:00</jam_berangkat>
- <jam_tiba>1899-12-30T10:00:00.000000+07:00</jam_tiba>
- <hari>Kamis</hari>
- <kode_terminal_asal>JUA</kode_terminal_asal>
- <kode_terminal_tujuan>SOEHAT</kode_terminal_tujuan>
- <kota_asal>Surabaya</kota_asal>
- <kota_tujuan>Jakarta</kota_tujuan>
- </KotaPerjalanan>
- <KotaPerjalanan diffgr:id="KotaPerjalanan3" msdata:rowOrder="2">
- <kode_trayek>4</kode_trayek>
- <jenis_trayek>EKSTRA</jenis_trayek>
- <kode_kend>GIA-002</kode_kend>
- <tanggal_berangkat>2006-04-06T00:00:00.000000+07:00</tanggal_berangkat>

```

Gambar 4.5 Data yang dihasilkan dari pemanggilan fungsi

web service2



```

- <hari>Kamis</hari>
- <kode_terminal_asal>JUA</kode_terminal_asal>
- <kode_terminal_tujuan>SOEHAT</kode_terminal_tujuan>
- <kota_asal>Surabaya</kota_asal>
- <kota_tujuan>Jakarta</kota_tujuan>
- </KotaPerjalanan>
- <KotaPerjalanan diffgr:id="KotaPerjalanan3" msdata:rowOrder="2">
- <kode_trayek>4</kode_trayek>
- <jenis_trayek>EKSTRA</jenis_trayek>
- <kode_kend>GIA-002</kode_kend>
- <tanggal_berangkat>2006-04-06T00:00:00.000000+07:00</tanggal_berangkat>
- <jam_berangkat>1899-12-30T20:00:00.000000+07:00</jam_berangkat>
- <jam_tiba>1899-12-30T21:00:00.000000+07:00</jam_tiba>
- <hari>Kamis</hari>
- <kode_terminal_asal>NGRAI</kode_terminal_asal>
- <kode_terminal_tujuan>JUA</kode_terminal_tujuan>
- <kota_asal>Bali</kota_asal>
- <kota_tujuan>Surabaya</kota_tujuan>
- </KotaPerjalanan>
- <KotaPerjalanan diffgr:id="KotaPerjalanan4" msdata:rowOrder="3">
- <kode_trayek>1</kode_trayek>
- <jenis_trayek>RUTIN</jenis_trayek>
- <kode_kend>GIA-001</kode_kend>
- <tanggal_berangkat>2006-04-05T00:00:00.000000+07:00</tanggal_berangkat>
- <jam_berangkat>1899-12-30T18:00:00.000000+07:00</jam_berangkat>
- <jam_tiba>1899-12-30T19:00:00.000000+07:00</jam_tiba>
- <hari>Rabu</hari>
- <kode_terminal_asal>JUA</kode_terminal_asal>
- <kode_terminal_tujuan>SOEHAT</kode_terminal_tujuan>
- <kota_asal>Surabaya</kota_asal>
- <kota_tujuan>Jakarta</kota_tujuan>
- </KotaPerjalanan>
- </NewDataSet>
- </diffgr:diffgram>
</DataSet>

```

Gambar 4.6 Data yang dihasilkan dari pemanggilan fungsi

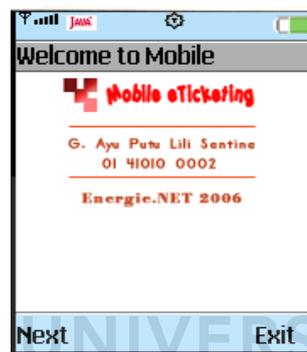
web service3

4.4 Tampilan Aplikasi MIDlet Pada Client

Berikut adalah tampilan program aplikasi sistem pemesanan tiket menggunakan web service :

4.4.1 Tampilan pembuka

Pada saat aplikasi pertama kali dijalankan, pada layar telepon selular akan muncul tampilan seperti di bawah ini :



Gambar 4.7 Tampilan pembuka

Tombol "Next" digunakan untuk pindah ke halaman berikutnya. Tombol "Exit" digunakan untuk keluar dari aplikasi.

4.4.2 Tampilan menu utama

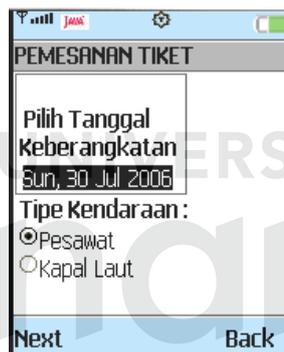
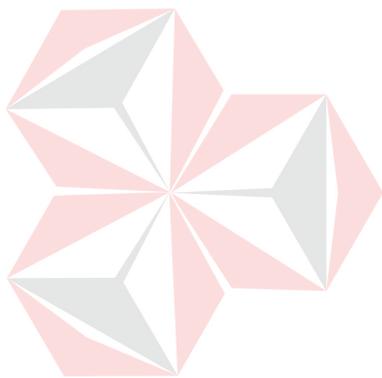


Gambar 4.8 Tampilan menu utama

Tampilan menu utama terdiri dari menu “Pemesanan Tiket”, “Pembatalan Tiket”, “Lihat Nomor Booking”, ”Konfirmasi” dan “About”. Untuk memilih menu dilakukan dengan cara mengarahkan kursor pada baris menu yang diinginkan, kemudian tekan tombol “OK”. Tombol “Exit” digunakan untuk keluar dari aplikasi.

4.4.3 Tampilan pemesanan tiket

Menu ”Pemesanan Tiket” terdiri dari pilihan tanggal dan 2 pilihan kendaraan yaitu ”Pesawat” dan ”Kapal Laut”.

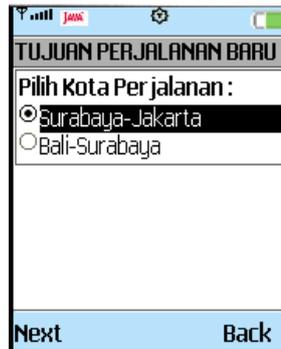


Gambar 4.9 Tampilan awal tanggal dan kendaraan



Gambar 4.10 Tampilan untuk memilih tanggal berangkat

Misalkan seorang user memilih kendaraan bertipe “Pesawat”, maka selanjutnya akan muncul tampilan info tujuan perjalanan sesuai kendaraan dan detail kendaraan yang dimaksud.



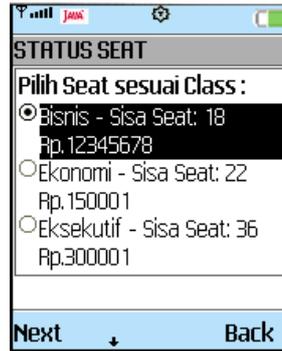
Gambar 4.11 Tampilan tujuan perjalanan

Tombol “Next” digunakan untuk memilih rute yang diinginkan dan berpindah ke halaman berikutnya yaitu tampilan jadwal maskapai dan jenis pesawat yang tersedia sesuai dengan rute yang dipilih.



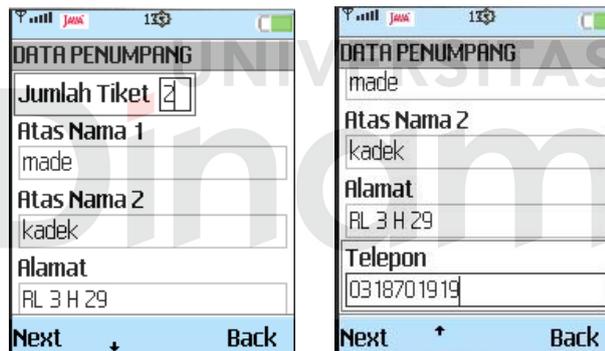
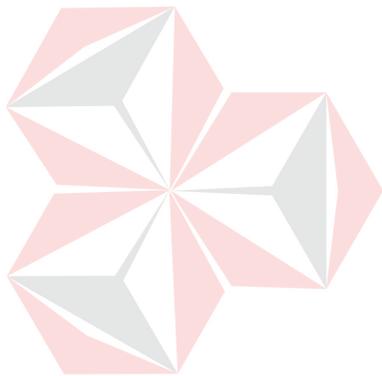
Gambar 4.12 Tampilan jadwal

Tombol “Next” digunakan untuk memilih jadwal yang diinginkan dan berpindah ke halaman berikutnya yaitu tampilan status seat yang tersedia.



Gambar 4.13 Tampilan status seat

Tombol “Next” digunakan untuk memilih kelas yang diinginkan dan berpindah ke halaman berikutnya yaitu tampilan entry data user serta jumlah tiket yang dipesan. Jumlah tiket yang dipesan tidak boleh melebihi 4 tiket.



Gambar 4.14 Tampilan entry data penumpang

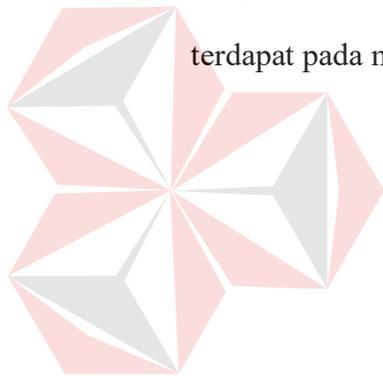
Textbox “Atas Nama” akan muncul secara dinamis sesuai inputan jumlah tiket. Tombol “Next” digunakan untuk memesan tiket. Jika pemesanan telah berhasil akan muncul halaman informasi untuk melakukan pemilihan apakah transaksi tersebut hanya berupa Booking (status “O”) atau *Issued* (status “I”). Status Booking berarti transaksi masih dapat dibatalkan dan diberi *time limit* selama 24 jam untuk melakukan konfirmasi (perubahan status “O” menjadi “I”). Status Issued berarti pemesanan tiket yang dilakukan tidak dapat dibatalkan lagi.



Gambar 4.15 Tampilan informasi

4.4.4 Tampilan konfirmasi

Menu "Konfirmasi" digunakan untuk mengkonfirmasi (merubah status "O" menjadi status "I") transaksi pemesanan tiket. Nomor-nomor booking yang terdapat pada menu Konfirmasi hanya nomor booking dengan status "O".



Gambar 4.16 Tampilan konfirmasi1

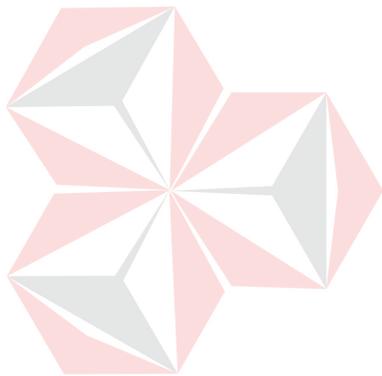
Nomor booking ditampilkan menggunakan Radio Button sehingga user dapat mengkonfirmasi nomor booking satu persatu. Untuk mengkonfirmasi dilakukan dengan memilih nomor booking yang diinginkan kemudian tekan tombol "Konfirmasi". Jika satu nomor booking telah dikonfirmasi, maka nomor booking tersebut akan hilang dari menu Konfirmasi.



Gambar 4.17 Tampilan konfirmasi2

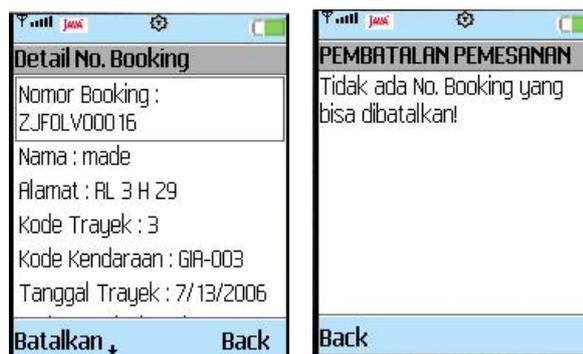
4.4.5 Tampilan pembatalan tiket

Menu "Pembatalan Tiket" berisi nomor-nomor booking yang memiliki status "O".



Gambar 4.18 Tampilan pembatalan1

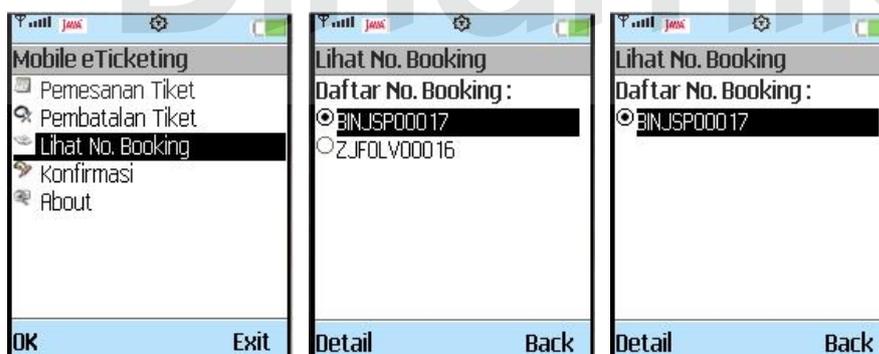
Untuk membatalkan tiket dilakukan dengan memilih nomor booking yang ingin dibatalkan kemudian tekan tombol "Lanjut", maka akan muncul detail info nomor booking bersangkutan. Untuk membatalkan dilakukan dengan menekan tombol "Batalkan".



Gambar 4.19 Tampilan pembatalan2

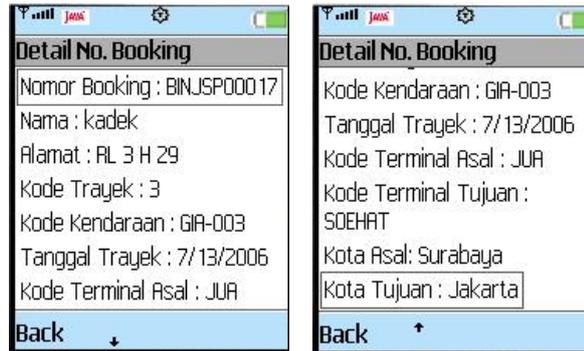
4.4.6 Tampilan lihat nomor booking

Menu "Lihat Nomor Booking" digunakan untuk melihat nomor-nomor booking yang terdapat dalam RMS dan melihat detail info tiap-tiap nomor booking. Pada awalnya terdapat 2 nomor booking dalam RMS, tetapi karena salah satu nomor booking dibatalkan yaitu nomor booking "ZJFOLV00016" maka hanya tersisa 1 nomor booking. Hal tersebut dapat dilihat pada gambar :



Gambar 4.20 Tampilan lihat nomor booking

Nomor booking ditampilkan menggunakan Radio Button. Untuk melihat detail nomor booking dilakukan dengan memilih nomor booking yang diinginkan kemudian tekan tombol "Detail".



Gambar 4.21 Tampilan detail nomor booking

4.4.7 Tampilan about



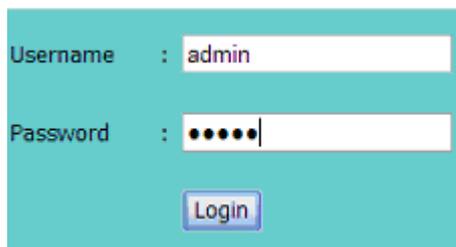
Gambar 4.22 Tampilan about

4.5 Tampilan Aplikasi Web Administrator

Berikut adalah tampilan aplikasi web yang digunakan untuk memelihara data pemesanan tiket pada biro perjalanan.

4.5.1 Tampilan login

Pada saat pertama kali dijalankan akan muncul tampilan login sebagai berikut :



Username : admin

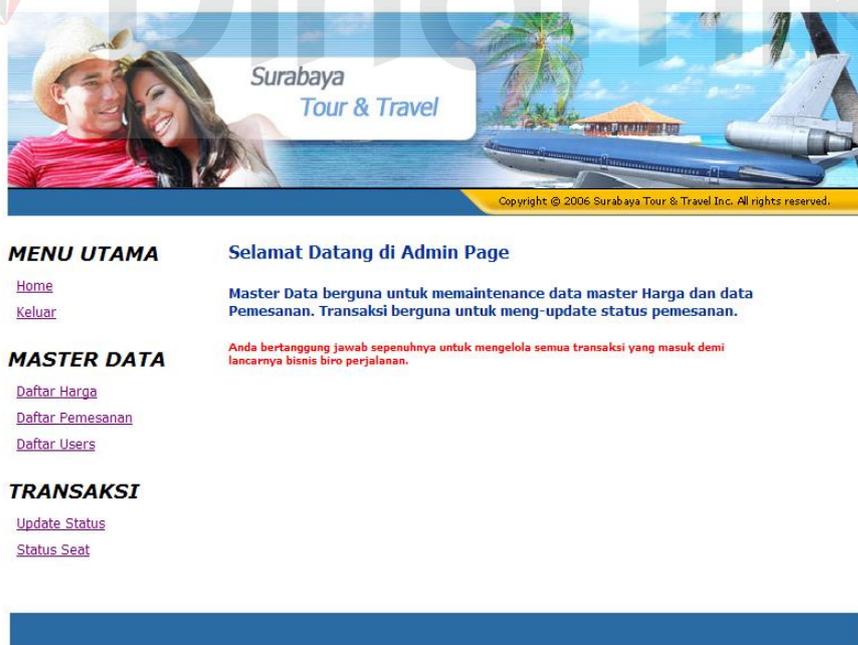
Password : ●●●●●

Login

Gambar 4.23 Tampilan login

Halaman login digunakan sebagai verifikasi seorang administrator yang berhak untuk melakukan akses terhadap aplikasi web ini. Pada halaman login seorang administrator diminta untuk memasukkan username dan password diikuti dengan penekanan tombol “Login”. Jika username dan password yang diinputkan benar, maka administrator akan masuk ke halaman utama.

4.5.2 Tampilan web halaman utama



Surabaya Tour & Travel

Copyright © 2006 Surabaya Tour & Travel Inc. All rights reserved.

MENU UTAMA

- [Home](#)
- [Keluar](#)

MASTER DATA

- [Daftar Harga](#)
- [Daftar Pemesanan](#)
- [Daftar Users](#)

TRANSAKSI

- [Update Status](#)
- [Status Seat](#)

Selamat Datang di Admin Page

Master Data berguna untuk memaintenance data master Harga dan data Pemesanan. Transaksi berguna untuk meng-update status pemesanan.

Anda bertanggung jawab sepenuhnya untuk mengelola semua transaksi yang masuk demi lancarnya bisnis biro perjalanan.

Gambar 4.24 Tampilan web halaman utama

Halaman utama berisikan *link-link* ke halaman “Daftar Harga”, “Daftar Pemesanan”, “Daftar Users”, “Update Status”, dan ”Status Seat”. Selain itu terdapat pula link “Home” dan “Keluar”.

4.5.3 Tampilan web daftar pemesanan

Halaman “Daftar Pemesanan” digunakan untuk melihat daftar pemesanan tiket yang masuk ke database.

[Home](#)
[Daftar Harga](#)
[Daftar Pemesanan](#)
[Daftar Users](#)
[Update Status](#)
[Keluar](#)

DAFTAR PEMESANAN

KODE BOOKING	TANGGAL PESAN	KODE TRAYEK	TANGGAL PERJALANAN	JENIS TRAYEK	NAMA	ALAMAT	HARGA	STATUS
8	5/22/2006 12:00:00 AM	1	22 Mei 2006	1	1	1	1	C
9		0					0	
10	5/29/2006 12:00:00 AM	2	6/5/2006	RUTIN	ninojk	loko	200000	C
11	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	Nino	DSI	200000	O
12	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NINI	DSI	2500000	O
13	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NINI	DSI	2500000	O
14	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NINI	DSI	2500000	O
15	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NINI	DSI	2500000	O
16	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NINI	DSI	2500000	O
17	6/6/2006 12:00:00 AM	1	10/10/2006	RUTIN	NinoG	DSI	200000	O

Gambar 4.25 Tampilan web daftar pemesanan

4.5.4 Tampilan web update status

Halaman “Update Status” digunakan untuk meng-update status dari suatu transaksi pemesanan tiket. Halaman ini berguna jika ada pembatalan manual dari user terhadap kode booking tertentu.

[Home](#) [Daftar Harga](#) [Daftar Pemesanan](#) [Daftar Users](#) [Update Status](#) [Keluar](#)

UPDATE STATUS

Masukkan Kode Booking :

 Garuda Airline

Set Status : Order Cancel Issued

Gambar 4.26 Tampilan web update status

4.5.5 Tampilan web daftar harga

Halaman “Daftar Harga” digunakan untuk melihat daftar harga dari setiap kode kendaraan.

[Home](#) [Daftar Harga](#) [Daftar Pemesanan](#) [Daftar Users](#) [Update Status](#) [Keluar](#)

DAFTAR HARGA

[\[Tambah Data\]](#)

KODE KENDARAAN	TIPE KENDARAAN	VENDOR	KODE KELAS	HARGA	JENIS TRAYEK	
GIA-001	PESAWAT	GARUDA	BIS	200001	RUTIN	Edit
GIA-001	PESAWAT	GARUDA	EKO	150001	RUTIN	Edit
GIA-001	PESAWAT	GARUDA	EKS	300001	RUTIN	Edit
GIA-002	PESAWAT	GARUDA	BIS	200001	RUTIN	Edit
GIA-002	PESAWAT	GARUDA	EKO	150001	RUTIN	Edit
GIA-002	PESAWAT	GARUDA	EKS	300001	RUTIN	Edit
LIO-001	Pesawat	Lion Air	BIS	2500000	Rutin	Edit
LIO-199	PESAWAT	LION AIR	EKS	10000	RUTIN	Edit
MAN-001	PESAWAT	MANDALA AJR	EKO	1000000	EKSTRA	Edit
KAP-001	KAPAL	BAJAK LAUT	EKS	100000	RUTIN	Edit
KAP-002	KAPAL	DEWA RUCI	EKS	100000	RUTIN	Edit
KER-001	KERETA	TURANGGA	EKS	200000	RUTIN	Edit

Gambar 4.27 Tampilan web daftar harga

Link “Tambah Data” digunakan untuk berpindah ke halaman “Tambah Data Harga”.

4.5.6 Tampilan web tambah data harga

Halaman “Tambah Data Harga” digunakan untuk menambah kode kendaraan baru dan daftar harga baru.

[Home](#)
[Daftar Harga](#)
[Daftar Pemesanan](#)
[Daftar Users](#)
[Update Status](#)
[Keluar](#)

TAMBAH DATA HARGA

Kode Kendaraan
 Tipe Kendaraan
 Vendor
 Kelas
 Harga
 Jenis Trayek Rutin Ekstra

Gambar 4.28 Tampilan tambah data harga

Tombol “Simpan” digunakan untuk menyimpan data, tombol “Batal” digunakan untuk membersihkan layar.

4.5.7 Tampilan web daftar user

Halaman “Daftar User” digunakan untuk melihat daftar administrator.

[Home](#)
[Daftar Harga](#)
[Daftar Pemesanan](#)
[Daftar Users](#)
[Update Status](#)
[Keluar](#)

DAFTAR USERS

[\[Tambah Data\]](#)

USERNAME	PASSWORD	RAMA
admin	admin	Administrator
lily	lily	Lily Kawai
coba	coba	gek coba
1		

Gambar 4.29 Tampilan web daftar user

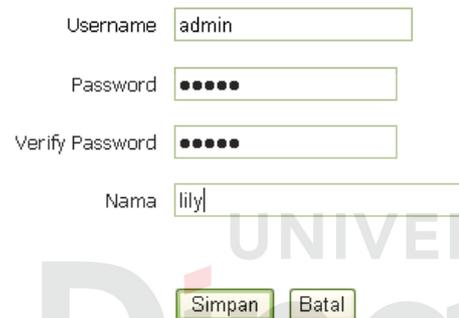
Link “Tambah Data” digunakan untuk berpindah ke halaman “Tambah Data User”.

4.5.8 Tampilan web tambah data user

Halaman “Tambah Data User” digunakan untuk menambah data administrator yang baru.

[Home](#) [Daftar Harga](#) [Daftar Pemesanan](#) [Daftar Users](#) [Update Status](#) [Keluar](#)

TAMBAH DATA USER



Username

Password

Verify Password

Nama

Gambar 4.30 Tampilan tambah data user

Tombol “Simpan” digunakan untuk menyimpan data, tombol “Batal” digunakan untuk membersihkan layar.

4.5.9 Tampilan web status seat

Halaman “Status Seat” digunakan untuk melihat sisa seat untuk setiap penyedia transportasi. Dengan demikian pihak biro perjalanan tidak harus menyimpan data-data sisa seat untuk setiap penyedia transportasi.

[Home](#) [Daftar Harga](#) [Daftar Pemesanan](#) [Daftar Users](#) [Update Status](#) [Status Seat](#) [Keluar](#)

STATUS SEAT

Garuda Airlines

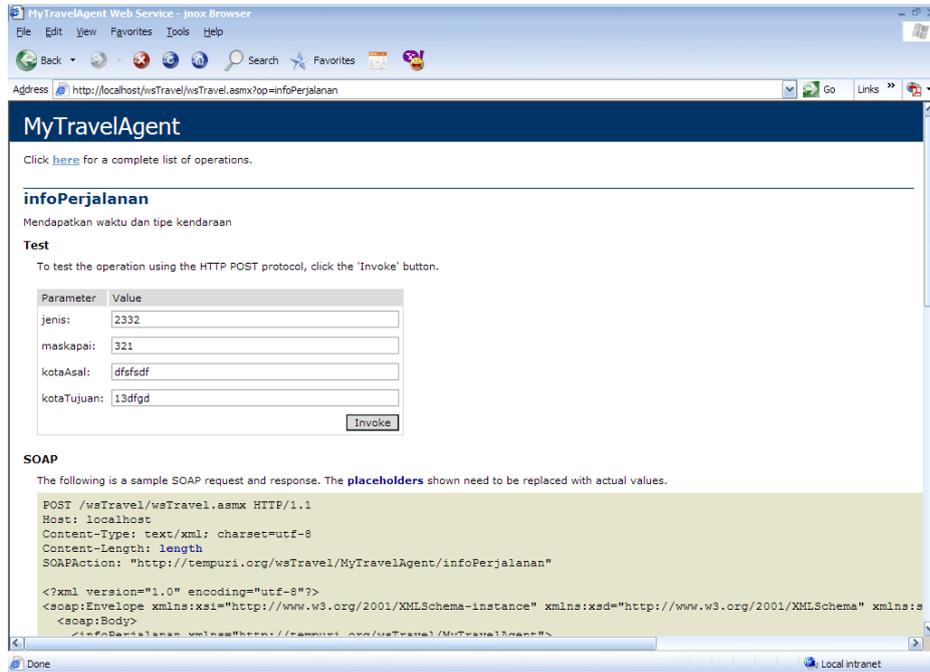
TANGGAL	KODE TRAYEK	KODE KENDARAAN	KODE KELAS	SEAT MASKAPAI	SEAT TERPAKAI	SISA SEAT
4/8/2006 12:00:00 AM	2	GIA-001	BIS	30	33	-3
4/8/2006 12:00:00 AM	2	GIA-001	EKO	30	30	0
4/8/2006 12:00:00 AM	2	GIA-001	EKS	40	37	3
4/6/2006 12:00:00 AM	3	GIA-003	BIS	40	29	11
4/6/2006 12:00:00 AM	3	GIA-003	EKO	40	4	36
4/6/2006 12:00:00 AM	3	GIA-003	EKS	20	1	19
4/5/2006 12:00:00 AM	1	GIA-001	BIS	30	14	16
4/5/2006 12:00:00 AM	1	GIA-001	EKO	30	8	22
4/5/2006 12:00:00 AM	1	GIA-001	EKS	40	4	36
1						

Gambar 4.31 Tampilan web status seat

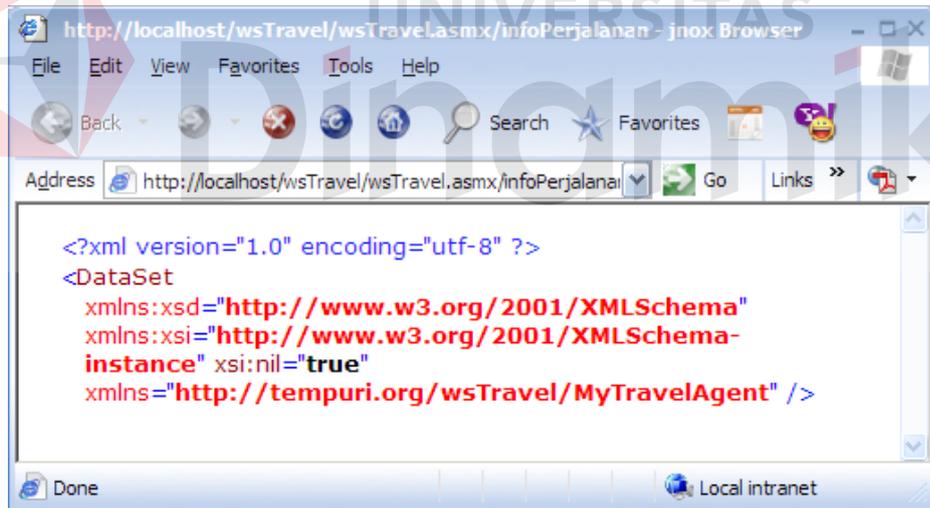
4.6 Tampilan Kesalahan

4.6.1 Tampilan kesalahan pada web service

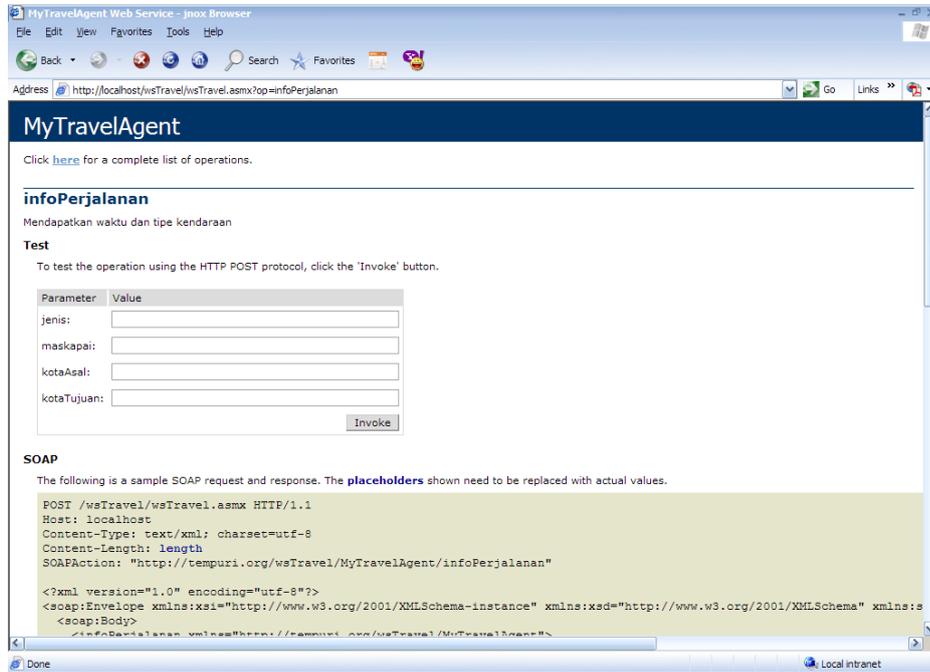
Berikut ini merupakan tampilan kesalahan pada aplikasi web service dan aplikasi MIDlet. Gambar 4.32 sampai 4.35 adalah tampilan kesalahan pada aplikasi web service jika terjadi kesalahan input data dan kesalahan jika tidak ada input data. Gambar 4.36 dan 4.37 adalah tampilan kesalahan pada MIDlet pada saat input tanggal keberangkatan.



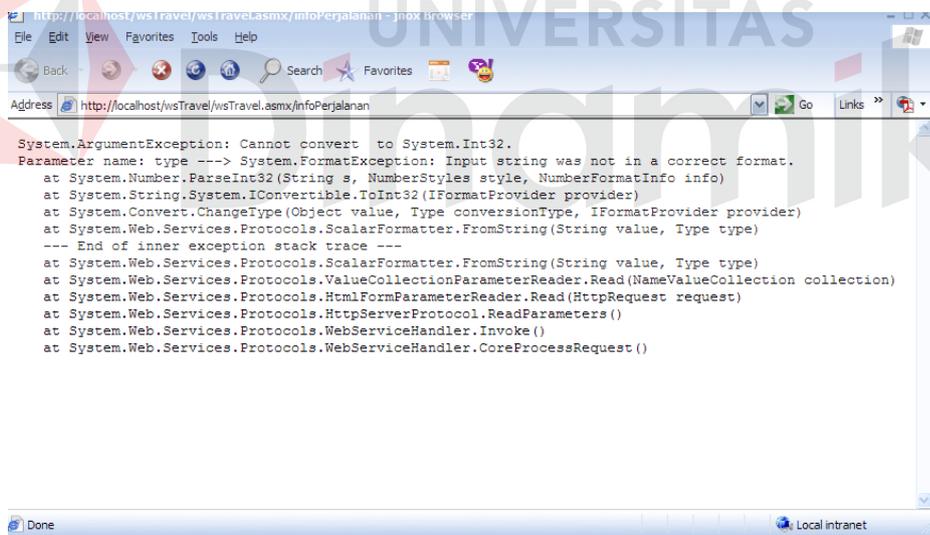
Gambar 4.32 Tampilan kesalahan jika inputan salah1



Gambar 4.33 Tampilan kesalahan jika inputan salah2



Gambar 4.34 Tampilan kesalahan jika tanpa inputan1

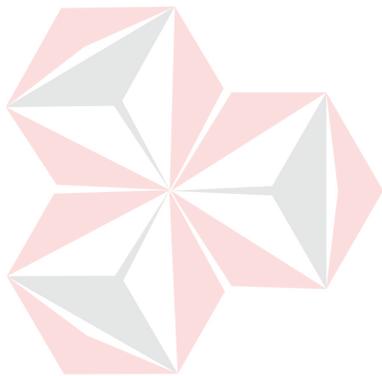


Gambar 4.35 Tampilan kesalahan jika tanpa inputan2

4.6.2 Tampilan kesalahan pada MIDlet



Gambar 4.36 Tampilan kesalahan jika tanggal berangkat lebih dari 1 bulan



Gambar 4.37 Tampilan kesalahan jika tanggal pemesanan kurang dari 3 hari sebelum tanggal berangkat

4.7 Evaluasi Sistem

4.7.1 Analisa hasil uji coba pada aplikasi MIDlet

Berikut ini adalah analisa hasil uji coba aplikasi MIDlet dalam mengakses fungsi-fungsi yang terdapat pada aplikasi web service. Hasil pengujian ini didapat setelah dilakukan *hosting* web service dan web administrator pada sebuah *free web hosting* yaitu www.somee.com. Aplikasi web service yang telah dihosting dapat dipanggil dengan alamat :

- a. <http://mobilbalap.somee.com/wsTravel/wsTravel.aspx> untuk web service biro perjalanan.
- b. <http://mobilbalap.somee.com/wsGaruda/wsGaruda.aspx> untuk web service Garuda Indonesia.
- c. <http://mobilbalap.somee.com/wsLion/wsLion.aspx> untuk web service Lion Air.
- d. <http://mobilbalap.somee.com/wsMandala/wsMandala.aspx> untuk web service Mandala Airlines.
- e. <http://mobilbalap.somee.com/wsKapal/wsKapal.aspx> untuk web service Kapal Laut.
- f. <http://mobilbalap.somee.com/Admin/login.aspx> untuk web administrator.

Tabel 4.1 Analisa hasil uji coba pada aplikasi MIDlet

Test Case ID	Tujuan	Input	Output yang diharapkan	Output sistem
1	Untuk mengetahui apakah fungsi kotaPerjalanan() dapat diakses oleh MIDlet	Pilih kendaraan dan tipenya	Tampilan tujuan perjalanan	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.11
2	Untuk mengetahui apakah fungsi infoPerjalanan() dapat diakses oleh MIDlet	Pilih tujuan perjalanan	Tampilan jadwal sesuai tujuan perjalanan	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.12
3	Untuk mengetahui apakah fungsi statusSeat() dapat diakses oleh MIDlet	Pilih tanggal berangkat	Tampilan status seat dan kelas	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.13

4	Untuk mengetahui apakah fungsi postOrder() dapat diakses oleh MIDlet	Pilih status Booking atau Issued	Tampilan nomor booking yang tersimpan dalam RMS	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.20
5	Untuk mengetahui apakah fungsi cancelOrder() dapat diakses oleh MIDlet	Pilih nomor booking yang ingin dibatalkan / dikonfirmasi	Tampilan nomor booking hilang dari menu pembatalan / konfirmasi	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.16, 4.17, 4.18, dan 4.19
6	Untuk mengetahui apakah fungsi displayNoBook() dapat diakses oleh MIDlet	Pilih nomor booking ingin dilihat detailnya	Tampilan detail nomor booking	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.20, dan 4.21

4.7.2 Analisa hasil uji coba pada aplikasi web Administrator

Berikut ini adalah analisa hasil uji coba aplikasi web Administrator dalam mengakses fungsi-fungsi yang terdapat pada web service.

Tabel 4.2 Analisa hasil uji coba pada aplikasi web Administrator

Test Case ID	Tujuan	Input	Output yang diharapkan	Output sistem
1	Untuk mengetahui apakah fungsi loginAuth() dapat diakses oleh web	Ketik username dan password	Admin masuk ke halaman utama	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.24

2	Untuk mengetahui apakah fungsi <code>selectData()</code> dapat diakses oleh web	Pilih menu daftar harga/daftar pemesanan/daftar user	Admin masuk ke halaman yang dipilih	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.25, 4.27, dan 4.29
3	Untuk mengetahui apakah fungsi <code>updateStatus()</code> dapat diakses oleh web	Ketik nomor booking, pilih pesawat, dan pilih status	Tampilan status yang telah berubah	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.26
4	Untuk mengetahui apakah fungsi <code>dispSeat()</code> dapat diakses oleh web	Pilih menu status seat	Tampilan status seat tiap-tiap penyedia transportasi	Sesuai dengan yang diharapkan. Terlihat pada gambar 4.31

4.7.3 Evaluasi kinerja sistem

Uji coba akan menggunakan parameter infrastruktur dan kepadatan jalur internet. Untuk parameter infrastruktur digunakan perbandingan 3 kartu SIM yang paling banyak digunakan pada operator telepon seluler yaitu Simpati, XL Bebas, dan IM3. Sedangkan untuk parameter kepadatan jalur internet menggunakan pembagian waktu pagi, siang, sore, dan malam. Diasumsikan telepon seluler yang digunakan telah mendukung aplikasi Java dengan MIDP 2 dan telah terinstal aplikasi tugas akhir ini. Contoh permasalahan yang digunakan adalah jika seseorang ingin melakukan pemesanan tiket dengan data-data input sebagai berikut :

Tabel 4.3 Contoh permasalahan

Kendaraan	Pesawat
Tipe	Garuda Indonesia Airlines
Tujuan	Surabaya-Jakarta
Jadwal berangkat	GIA-003 Kamis 09.00
Tanggal berangkat	Rabu 19 Juli 2006
Kelas	Bisnis
Jumlah tiket	2
Nama 1	Lili
Nama 2	Santina
Alamat	RL 3H 29 Surabaya
Telepon	0318701910
Status tiket	Booking

Detil proses yang dilakukan oleh user dalam menjalankan aplikasi ini untuk pemesanan akan dijelaskan melalui tahap-tahap berikut ini :

1. User mengaktifkan MIDlet dan memasuki menu utama.
2. User masuk ke dalam menu “Pemesanan” kemudian memilih kendaraan “Pesawat” dengan tipe “Garuda Indonesia Airlines”.
3. MIDlet mengambil data tujuan keberangkatan sesuai kendaraan dan tipe dari web service.....i
4. User memilih tujuan “Surabaya-Jakarta”. MIDlet mengambil dari web service jadwal keberangkatan tujuan Surabaya-Jakarta.....ii
5. User memilih tanggal berangkat.
6. MIDlet mengambil data seat dan kelas berdasarkan tanggal yang dipilih.....iii
7. User memilih kelas “Bisnis”.

8. User mengisi jumlah tiket dan mengisi data penumpang.
9. User memilih status “Booking”. MIDlet mengirim data pemesanan ke web service dan menyimpan nomor bookingiv
10. User kembali ke menu utama.
11. Pada menu “Lihat Nomor Booking” terdapat 2 nomor booking yaitu “CEFMNO00039” dan “GJJQST00038”.
12. User melihat detail nomor booking “CEFMNO00039”.
13. MIDlet mengambil detail info nomor booking tersebut.....v
14. User kembali ke menu utama kemudian memilih menu “Konfirmasi”.
15. User melakukan konfirmasi terhadap salah satu nomor booking yaitu nomor booking “CEFMNO00039”.....vi
16. User kembali ke menu utama kemudian memilih menu “Pembatalan Tiket”.
17. User melakukan pembatalan terhadap nomor booking yang tersisa yaitu nomor booking “GJJQST00038”. MIDlet mengambil detail info nomor booking tersebutvii
18. User melakukan pembatalan terhadap nomor booking tersebut.....viii

Untuk melakukan perhitungan waktu dibuat sebuah fasilitas tambahan yaitu pembuatan class `HitungWaktu.java`. Perhitungan waktu dilakukan dengan memanggil method `getDetik()` pada tiap-tiap proses diatas. Waktu yang diperlukan dalam melakukan detail proses dapat dijelaskan pada tabel 4.4, 4.5, dan 4.6. Waktu yang didapat merupakan waktu rata-rata setelah aplikasi dicoba sebanyak 3 kali.

Tabel 4.4 Tabel perhitungan waktu dengan kartu Simpati (dalam detik)

Proses	06.00-09.59	10.00-14.59	15.00-18.59	19.00-22.00
i	31	30	42	31
ii	0	0	0	0
iii	17	25	24	25
iv	25	38	35	33
v	5	9	20	19
vi	6	17	17	17
vii	6	15	12	18
viii	8	10	12	15

Tabel 4.5 Tabel perhitungan waktu dengan kartu XL Bebas (dalam detik)

Proses	06.00-09.59	10.00-14.59	15.00-18.59	19.00-22.00
i	25	24	25	31
ii	0	0	0	0
iii	20	23	23	20
iv	35	43	52	42
v	10	13	15	19
vi	10	13	10	12
vii	11	12	6	8
viii	7	7	6	9

Tabel 4.6 Tabel perhitungan waktu dengan kartu IM3 (dalam detik)

Proses	06.00-09.59	10.00-14.59	15.00-18.59	19.00-22.00
i	24	23	44	26
ii	0	0	0	0
iii	16	27	24	24
iv	40	62	38	35
v	10	12	9	8
vi	8	18	15	12
vii	6	13	16	28
viii	10	19	12	8

Untuk melakukan 1 kali transaksi booking normal (memesan tiket dan melakukan konfirmasi) memerlukan transfer data kurang lebih sebanyak 25 Kbyte. Tarif yang dikenakan untuk tiap-tiap operator dalam mentransfer data sebanyak 25 Kbyte adalah Rp. 750,- (untuk operator Simpati), Rp. 625,- (untuk operator XL Bebas), dan Rp. 500,- (untuk operator IM3).

Dengan melakukan tahap-tahap proses yang sama tiap operator menghasilkan waktu yang berbeda, perbedaan waktu yang terjadi dapat disebabkan perbedaan infrastruktur yang digunakan tiap operator.

4.7.4 Analisa hasil uji coba sistem yang digunakan

Tabel 4.7 Analisa hasil uji coba sistem yang digunakan

Test Case ID	Tujuan	Input	Output sistem
1	Untuk membuktikan bahwa aplikasi web service dapat diintegrasikan dengan bahasa pemrograman yang berbeda sehingga dapat diakses pihak yang berkepentingan	Pada aplikasi MIDlet digunakan bahasa pemrograman J2ME sebagai bahasa pemrograman mobile device	Sesuai dengan yang diharapkan. Sebab sistem berhasil diintegrasikan meskipun memiliki bahasa pemrograman yang berbeda. Hal ini dapat dilihat pada listing program
2	Untuk membuktikan bahwa aplikasi web service dapat diintegrasikan dengan bahasa pemrograman yang berbeda sehingga dapat diakses pihak yang berkepentingan	Pada aplikasi web admin digunakan bahasa pemrograman ASP.NET sebagai standar bahasa pemrograman web	Sesuai dengan yang diharapkan. Sebab sistem berhasil diintegrasikan meskipun memiliki bahasa pemrograman yang berbeda. Hal ini dapat dilihat pada listing program

BAB V

PENUTUP

5.1 Kesimpulan

Dari analisa yang telah diperoleh maka dapat disimpulkan bahwa :

1. Penggunaan aplikasi XML web service pada sistem pemesanan tiket dapat dinyatakan berhasil, sebab dapat memudahkan pihak biro perjalanan untuk memberikan informasi perjalanan kepada client tanpa harus menyimpan data-data perjalanan dari tiap-tiap perusahaan penyedia transportasi. Aplikasi MIDlet dapat memberikan informasi tentang jadwal keberangkatan pesawat atau kapal laut untuk kota asal dan tujuan tertentu, memberikan status seat pada tanggal tertentu dengan memanggil fungsi-fungsi yang disediakan oleh web service perusahaan penyedia transportasi.
2. Dengan memanfaatkan fasilitas koneksi HTTP aplikasi MIDlet mampu melakukan pemanggilan fungsi-fungsi atau mengakses sumber daya pada web service seperti layaknya pemesanan tiket melalui web. Aplikasi MIDlet yang melibatkan komunikasi jaringan seperti aplikasi sistem pemesanan tiket ini sangat tergantung pada operator penyedia layanan ponsel.
3. Dokumen XML sebagai format pertukaran data sangat baik digunakan untuk pertukaran data antar perusahaan yang menggunakan web service karena XML bersifat lintas platform dan sistem operasi. Namun untuk format pertukaran data antara web service dengan aplikasi client MIDlet pada telepon seluler, dokumen XML kurang baik untuk digunakan. Hal itu disebabkan karena dokumen XML memiliki format data kompleks jadi data

yang dikirim sangat besar sehingga memakan bandwidth jaringan cukup besar pula dan waktu yang diperlukan juga menjadi lebih lama.

5.2 Saran

Saran-saran yang direkomendasikan untuk pengembangan dan implementasi aplikasi MIDlet dan web service pada umumnya dan aplikasi sistem pemesanan tiket menggunakan web service pada khususnya adalah :

1. Agar transmisi data antara perusahaan dengan aplikasi MIDlet tidak memakan waktu lama dan menggunakan bandwidth yang besar, dapat diterapkan *StringBuilder* pada web service untuk mengubah format dokumen XML yang kompleks menjadi format String yang lebih singkat.
2. Untuk meningkatkan keamanan pada aplikasi web service dan aplikasi wireless maka dapat diterapkan *Hypertext Transport Protocol Secure* (HTTPS) atau HTTP yang memanfaatkan *Secure Sockets Layer* (SSL) untuk keamanan transmisi data.
3. Perlunya implementasi aplikasi web service dan aplikasi MIDlet yang melibatkan transaksi pembayaran. Hal ini sebaiknya didukung dengan adanya kerjasama dengan operator penyedia layanan ponsel atau bank, sehingga pembayaran dapat dimasukkan ke dalam tagihan atau dengan solusi lain yang lebih efektif.
4. Perlunya registrasi web service yang dirancang pada direktori UDDI untuk memudahkan dalam pencarian dan deskripsi web service.