BAB III
PEMBAHASAN

3.1 Jadwal Kerja Praktek

Kerja praktek dilaksanakan di PT. Astra International, Tbk yang beralamat di Jl. Gaya Motor Raya No. 8 Sunter II, Jakarta Utara. Adapun pelaksanaan kerja praktek dimulai pada tanggal 5 Juli 2010 sampai dengan 20 Agustus 2010.

3.2 Teknik Kerja Praktek

Dalam pelaksanaan kerja praktek ini dibagi kedalam beberapa tahapan kegiatan antara lain :

1. Pembuatan surat permohonan kerja praktek ditujukan untuk PT. Astra International, Tbk
2. Pengajuan permohonan kerja praktek di PT. Astra International, Tbk yang beralamat di Jl. Gaya Motor Raya No. 8 Sunter II, Jakarta Utara
3. PT. Astra International, Tbk memberikan surat balasan permohonan kerja praktek
4. Pelaksanaan kerja praktek

Dalam pelaksanaan kerja praktek ini dibagi kedalam beberapa tahapan kegiatan untuk mengetahui permasalahan yang ada didalam pelaksanaan kegiatan kerja praktek ini.

1. Tahapan analisis sistem

   Setelah mendapatkan data yang cukup, langkah selanjutnya adalah kegiatan analisis, kegiatan analisis terdiri dari: analisis perangkat keras, analisis perangkat lunak pembangun sistem dan analisis user

2. Perancangan sistem dan perangkat lunak

   Setelah menganalisis sistem, selanjutnya adalah merancang sistem *webservice* dan menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar.

3. Implementasi dan pengujian Unit

   Setelah membuat perancangan sistem dan perangkat lunak, langkah selanjutnya adalah perancangan perangkat lunak yang sudah dirancang direalisasikan sebagai serangkaian program atau unit program dan pengujian unit meliputi instalasi aplikasi di *server* dan melakukan percobaan aplikasi oleh user di computer *client*

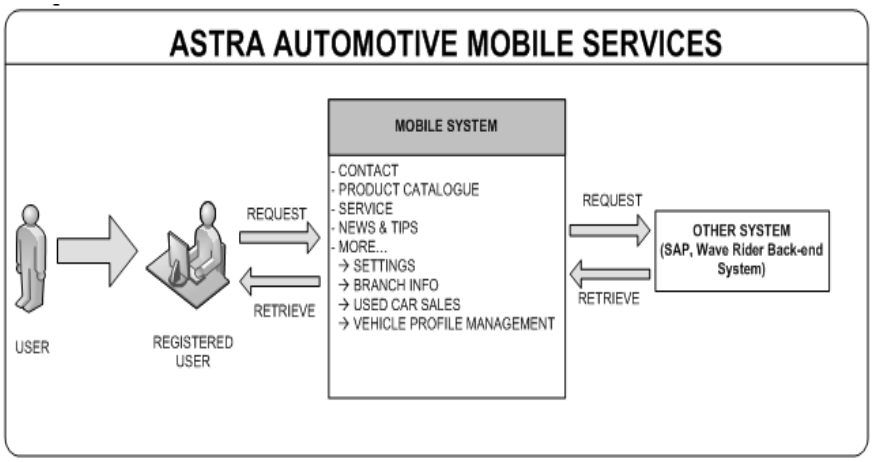
3.3 Data Hasil Kerja Praktek

3.3.1 Analisis Masalah

Sesuai dengan hasil penelitian yang didapat, aplikasi WaveRider harus didukung oleh sebuah sistem yang mumpuni agar dapat mengakses database dan bertukar informasi dikarenakan terdapat perbedaan platform. Berdasarkan hal tersebut maka diperlukan sistem webservice agar masalah tersebut dapat diatasi.

3.3.2 Analisis Sistem

3.3.2.1 High Level Proses



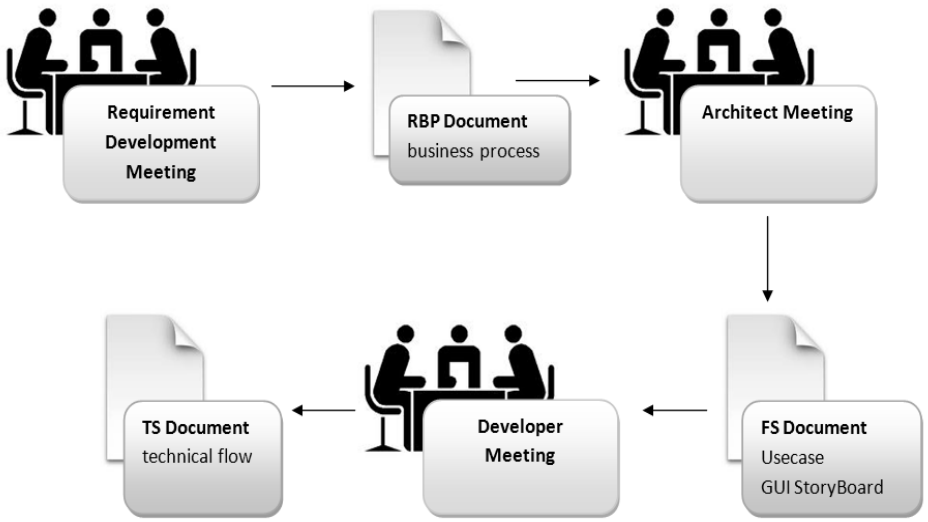
Gambar III.1 High Level Proses

User harus terdaftar terlebih dahulu sebelum dapat menggunakan aplikasi ini. Setelah *user* menginstall WaveRider ke dalam perangkat iPhonenya, *user* baru dapat menggunakan fitur-fitur yang disediakan. Aplikasi di iPhone mengumpulkan data dengan cara mengirim *request* ke web service.

3.3.2.2 Proses Analisis Kebutuhan di PT. Astra International Tbk

Proses analisis kebutuhan di PT Astra International, Tbk. diawali dengan adanya pertemuan dengan seorang Business Process Analyst dengan calon pengguna aplikasi dalam forum yang disebut dengan Requirement Development Meeting atau RDM. Pada RDM akan dibicarakan mengenai kebutuhan pengguna. BPA memiliki tanggung jawab penuh untuk memahami kebutuhan pengguna dan

proses bisnisnya. Dari RDM yang dilakukan, dihasilkan sebuah dokumen yang disebut dengan Requirement Business Process atau RBP.



Gambar III.2 Proses Analisis Kebutuhan

Setelah terbentuknya sebuah RBP, BPA akan mengadakan pertemuan dengan para arsitek aplikasi dalam sebuah Architect Meeting. Pada forum ini akan dibicarakan mengenai arsitektur aplikasi beserta GUI Story Board dari aplikasi secara detail. Setelah pertemuan tersebut, terbentuklah dokumen Functional Spesification Document atau FSD.

Setelah menerima FSD dari BPA, tim Developer akan merancang Technical Spesification Design atau TSD. Proses pengembangan aplikasi harus sesuai dengan TSD yang telah dibuat.

Jika aplikasi sudah dinilai jadi oleh tim Developer, tim Training and Testing akan melakukan pengetesan aplikasi berdasarkan FSD. Setelah dianggap layak, aplikasi akan diserahkan ke *Production* untuk proses deployment aplikasi.

3.3.2.3 Web Service

Standar protokol web service yang digunakan oleh WaveRider adalah SOAP dan WSDL. Standar tersebut digunakan karena kedua standar tersebut sudah terdapat pada framework .NET sebagai standar komunikasi antar platform.

1. SOAP

SOAP merupakan kependekan dari *Simple Object Access Protocol*. SOAP adalah protokol yang digunakan untuk pertukaran informasi yang terstruktur dalam implementasi *web service*. Dalam pertukaran informasi tersebut, format yang digunakan adalah XML atau *Extensible Markup Language*. XML memiliki banyak kelebihan dibandingkan dengan format lainnya, yaitu lebih mudah dibaca dengan bahasa manusia, lebih mudah untuk dideteksi kesalahannya, dan dapat digunakan pada banyak *platform*.

1. WSDL

WSDL atau *Web Services Description Language* adalah bahasa berbasis pada XML yang menggambarkan sebuah model sebagai representasi dari *web service*. WSDL versi 1.1 dan sebelumnya memiliki kepanjangan *Web Service Definition Language*. WSDL mendefinisikan sebuah *service* sebagai kumpulan

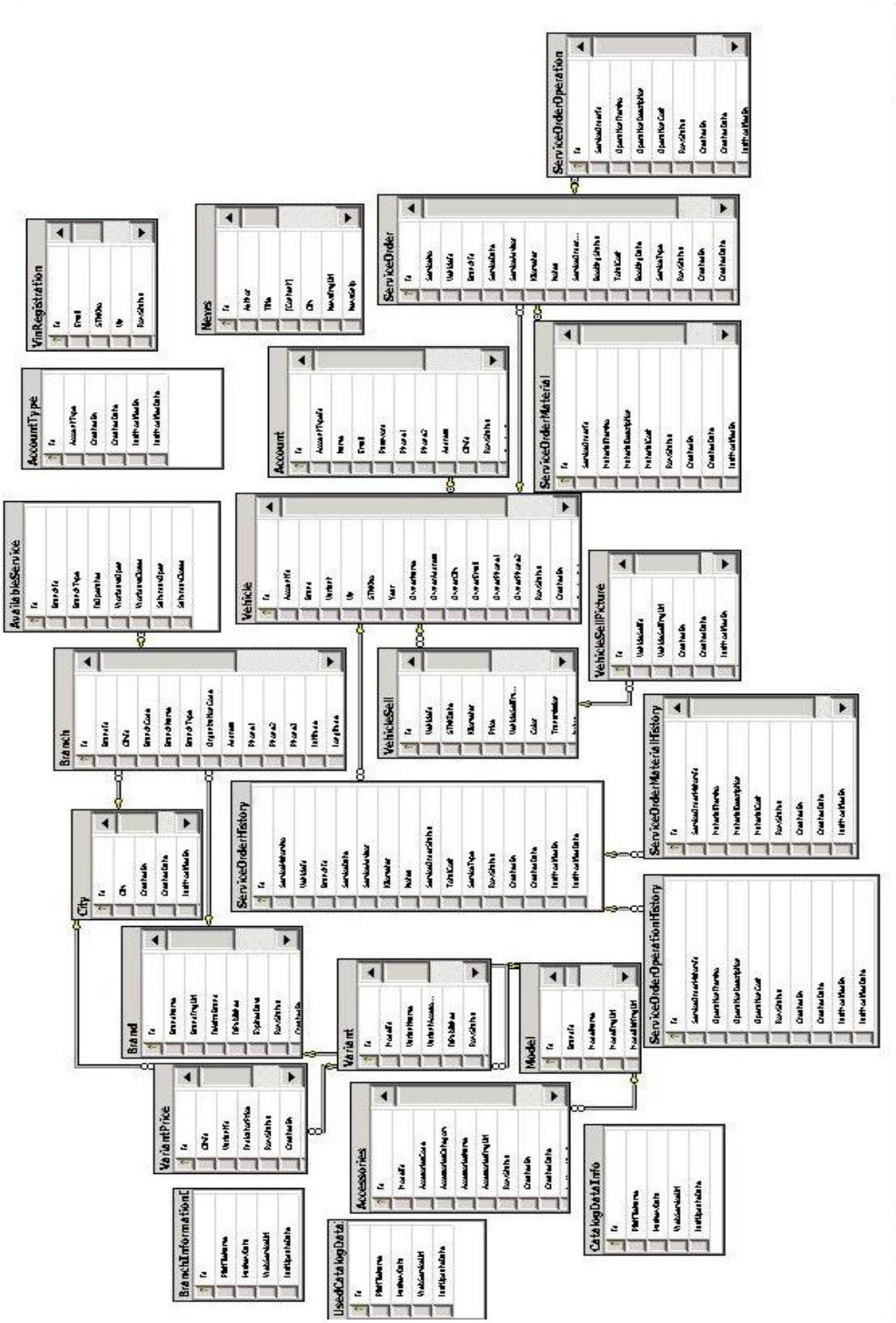
dari *port* atau *endpoint* yang biasanya diwakili oleh URL HTTP. WSDL menyediakan pola XML sebagai spesifikasi dari *service* tersebut.

WSDL sering dikombinasikan dengan SOAP dan skema XML dalam penerapannya. Program disisi *client* yang sudah terkoneksi dengan web service dapat membaca dokumen WSDL untuk mengetahui *service* apa saja yang tersedia.

3.3.3 Analisis Basis Data

Usulan untuk diagram E-R yaitu terdapat kunci yang unik (*primary key*) pada setiap entitas (tabel induk) yang dapat membedakan dengan atribut lainnya sehingga tabel tersebut dapat dijadikan referensi untuk tabel yang lainnya.

Dari usulan diatas maka dapat dibuat diagram E-R untuk sistem yang terdri dari beberapa tabel yang dibutuhkan untuk menyimpan data. Untuk melihat keterhubungan antar tabel yang ada maka akan digambarkan sebagai berikut :

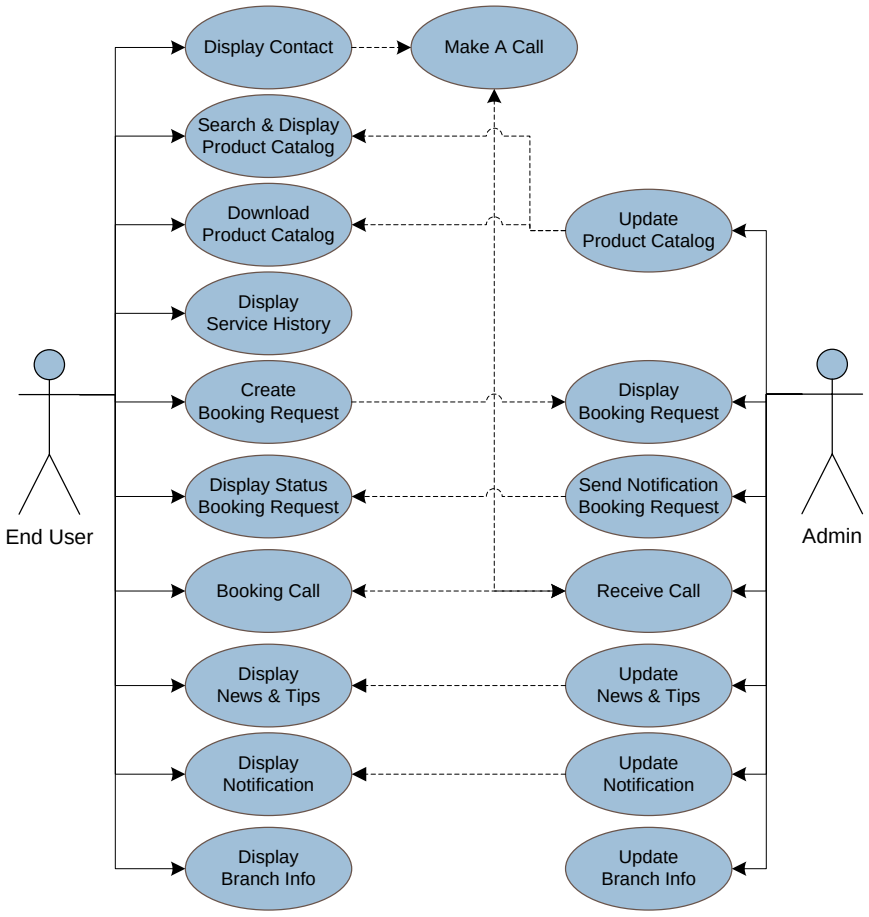


Gambar III.3 Hubungan Antar Tabel

3.3.4 Desain

3.3.4.1 Usecase

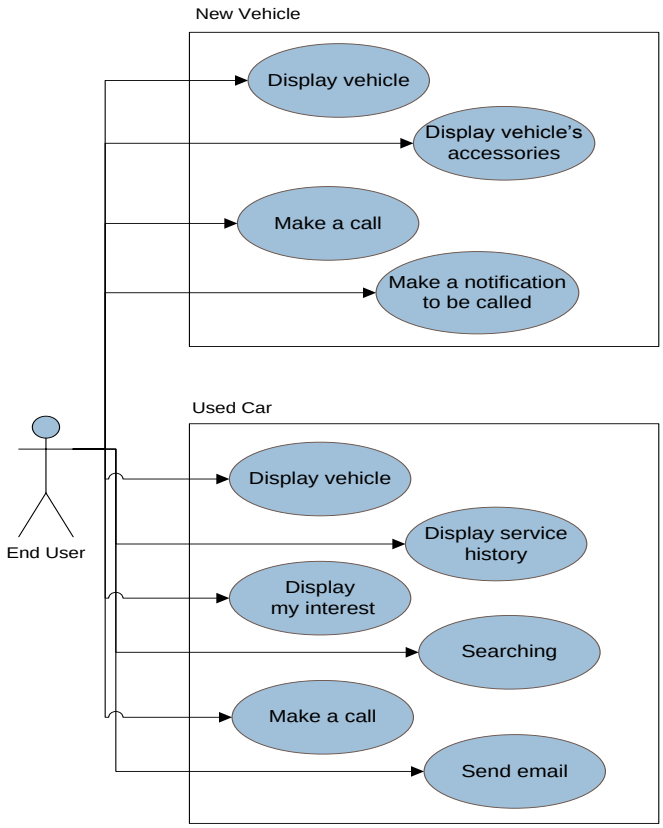
AAMS merupakan singkatan dari Astra Automotive Mobile Services yang merupakan salah satu proses bisnis yang ada pada aplikasi WaveRider, yang memiliki fungsi dalam pembookingan waktu service kendaraan pelanggan secara mobile.



Gambar III.4 Usecase Diagram

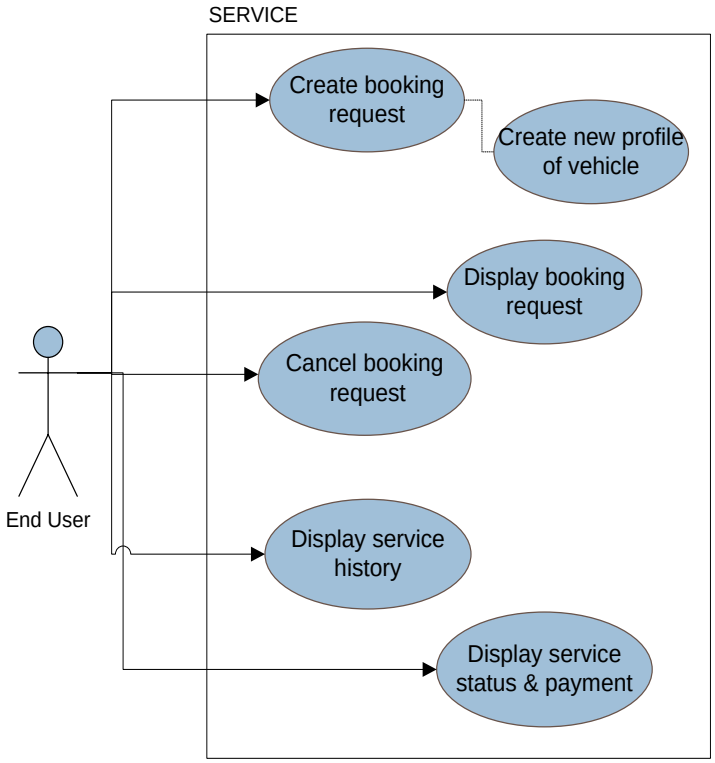
Product Catalogue berisi daftar kendaraan sebagai produk otomotif Astra Automotive Group, terdiri dari kendaraan dan aksesories baru serta mobil bekas. *End user* dapat melihat katalog hirarki produk dari detail model dan merek. Dari info detail kendaraan baru, pelanggan dapat melihat varian, warna, harga dan juga aksesoris. Jika pelanggan tertarik pada salah satu produk, mereka dapat mengkontak Astra World untuk mendapatkan infofrmasi detail dari kendaraan baru terserbut.

Di dalam *Product catalogue* juga disediakan suatu *search engine* yang memiliki fungsi untuk memudahkan pelanggan dalam mencari kendaraan yang dia cari.



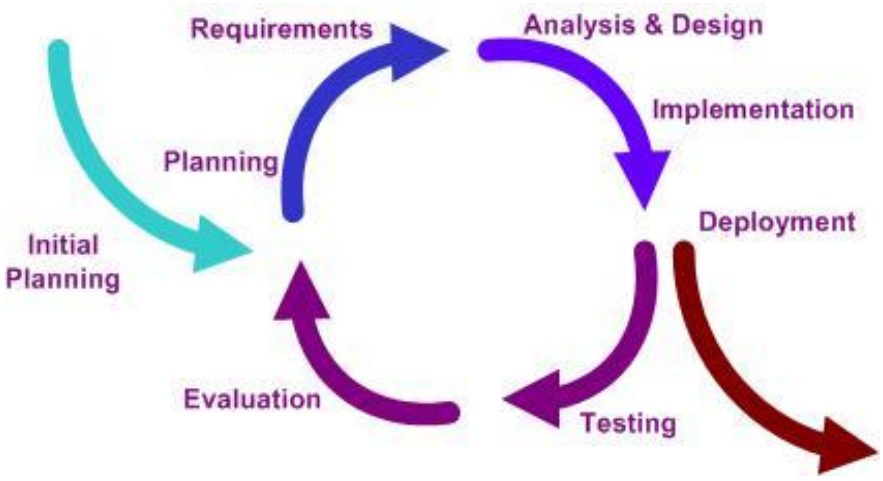
Gambar III.5 Usecase Product Catalogue

Service pada fitur AAMS berisi pemesanan Permintaan, *History Service*, dan Status *Service* dan Pembayaran. Pengguna dapat membuat permintaan pesanan, melihat status *booking* dan pembatalan pesanan pada menu Pemesanan Permintaan. Pemesanan permintaan tidak terbatas pada satu kendaraan yang dimiliki *user* saja, namun bisa lebih, yaitu dengan membuat profil baru untuk kendaraan yang lain.



3.3.5 Pengembangan Aplikasi

Pada pengembangan Aplikasi, Project Wave Rider menggunakan Metode Incremental Developtment Model. Untuk lebih jelasnya silahkan lihat gambar di bawah ini yang akan menjelaskan tentang alur dari model pengembangan Aplikasi dengan menggunakan Incremental Development Model.



Gambar III.6 Incremental Development Model

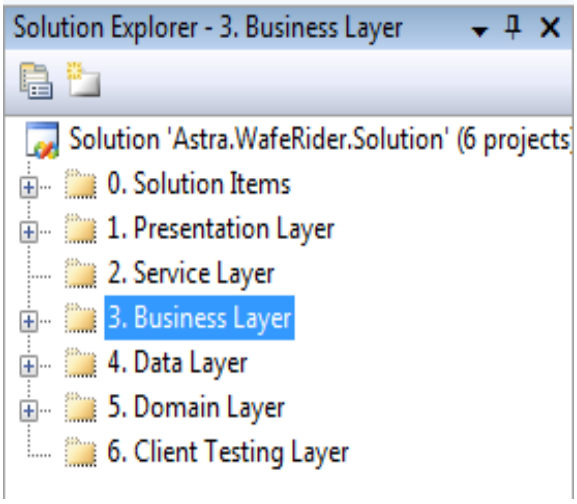
Aplikasi WaveRider ini dimulai dengan perencanaan aplikasi yang kemudian dilanjutkan pada tahap perencanaan kebutuhan aplikasi, yang terus berlanjut hingga analisis dan Design Aplikasi yang nantinya akan digunakan, setelah itu berlanjut pada Implementasi penggunaan. Setelah aplikasi dirasa cukup siap, diadakan testing terhadap aplikasi tersebut, kemudian dilakukan evaluasi terhadap kinerja dari aplikasi tersebut. Bila dirasa masih banyak kekurangan, maka proses akan kembali masuk ke dalah tahapan planning.

Di tengah-tengah tahap pengembangan aplikasi ini, terkadang sering terjadi perombakan kebutuhan dan design pihka pengembang, sehingga tidak menutup kemungkinan perubahan dasar pada aplikasi sering terjadi.

3.4 Implementasi Sistem

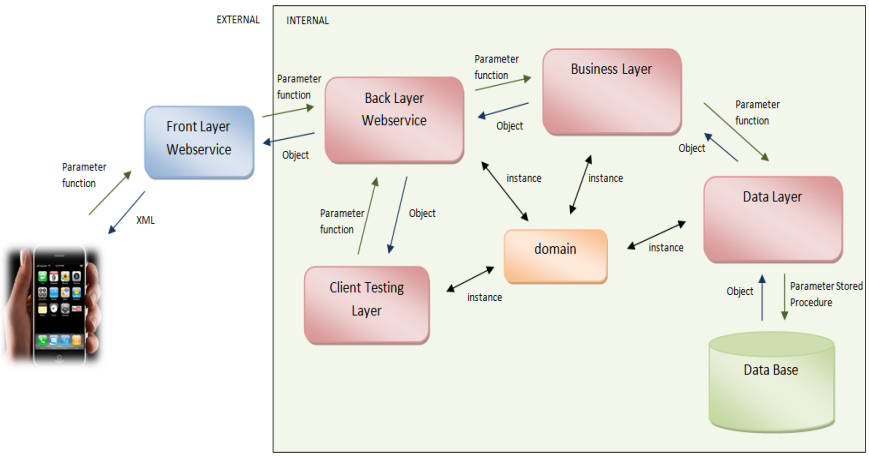
3.4.1 Struktur Aplikasi

Struktur aplikasi suatu project memiliki peranan penting dalam pembuatan perangkat lunak di PT Astra International, Tbk. Programmer harus berpegang pada struktur aplikasi dalam pengembangan perangkat lunak tersebut. Tujuan utamanya adalah efisiensi kinerja dan kejelasan alur sistem. Semakin bagus arsitektur aplikasi diharapkan menghasilkan aplikasi dengan kinerja yang efisien dan alur sistem yang tertata rapi. Jika suatu saat ada pengembangan lebih lanjut dari aplikasi yang sistemnya tertapa rapi, maka proses pengembangan akan lebih mudah.



Gambar III.7 Struktur Aplikasi

Secara garis besar, aristektur aplikasi WaveRider terdiri dari 5 lapisan/layer, yaitu Service Layer, Business Layer, Data Layer, Domain Layer, dan Client Testing Layer seperti pada Gambar di bawah ini



Gambar III.8 Alur Aplikasi

Gambar di atas menunjukkan Service Layer berperan sebagai jembatan utama untuk komunikasi antara aplikasi yang terinstal di iPhone dengan logika pengolahan data di web service. Business Layer adalah alur logika pengolahan data dari Data Layer menjadi informasi yang siap dikirim oleh Service Layer. Data Layer berperan sebagai jembatan utama web service dengan server database, sehingga data siap diolah oleh Business Layer. Domain Layer disediakan sebagai objek yang menjadi cikal bakal XML. Client Testing Layer berperan pada saat proses testing web service.

3.4.1.1 Service Layer

Service Layer berperan sebagai jembatan utama untuk komunikasi antara aplikasi yang terinstal di iPhone dengan logika pengolahan data di web service.

Aplikasi pada iPhone akan mengirim request ke web service yang disertai dengan parameter-parameter tertentu sesuai dengan kebutuhan. Layer yang

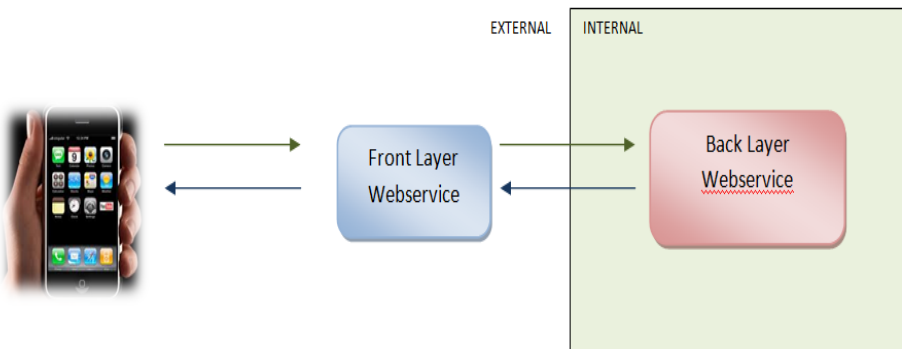
pertama kali menerima adalah service front-layer. Karena service front-layer ini tidak memiliki akses terhadap data, service front-layer meneruskan request tersebut ke service back-layer.

Service back-layer ini lah yang nantinya akan memanggil class yang ada di business layer untuk mengolah data sesuai dengan bisnis proses yang direncanakan.

1. Service Front-Layer

Layer yang pertama kali menerima adalah service front-layer. Karena service front-layer ini tidak memiliki akses terhadap data, service front-layer meneruskan request tersebut ke service back-layer.

Pada layer ini, tidak terjadi pengolahan data. Layer ini hanya mengirim parameter kepada instance service back-layer dan mengirim XML yang diterima dari service back-layer.

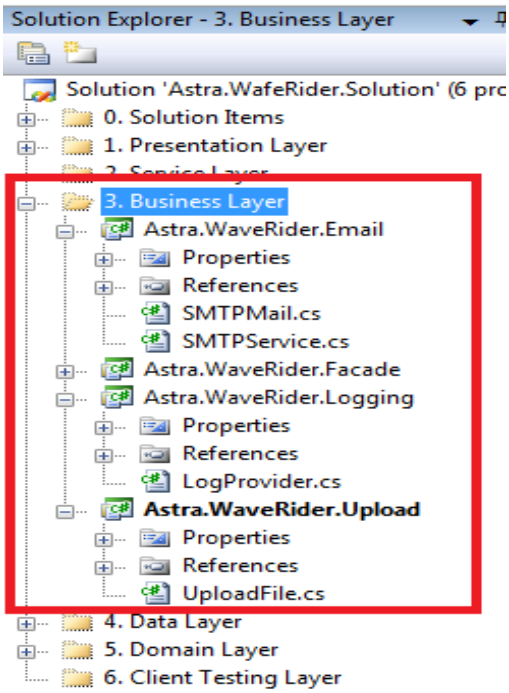


Gambar III.9 Service Front Layer

Gambar di atas menunjukkan bahwa layer ini juga tidak memiliki akses ke banyak kelas seperti yang dapat dilakukan oleh service back-layer. Tujuannya adalah untuk mencegah adanya akses yang tertentu yang dapat membahayakan sistem. Dengan terbatasnya akses yang dimiliki oleh service front-layer, akses dari aplikasi external, dalam hal ini iPhone, tidak dapat melakukan banyak hal kecuali sesuai dengan apa yang tersusun pada service front-layer.

2. Business Layer

Business Layer pada aplikasi web service WaveRider ini berguna sebagai tempat pengubah data mentah menjadi informasi sesuai dengan proses bisnis masing-masing fungsi.



Gambar III.10 Business Layer

Dalam penerapannya, business layer sendiri terbagi menjadi beberapa bagian, upload, logging, façade, dan email seperti pada Gambar di atas. Upload bertugas untuk menangani permasalahan upload yang dilakukan oleh user eksternal. Logging berguna sebagai pencatat error yang terjadi selama proses eksekusi. Façade adalah otak dari business layer, tempat logika yang mewakili proses bisnis masing-masing fungsi. Email berperan dalam menangani email yang dikirim melalui aplikasi.

3.4.1.2 Data Layer

Data layer ini menangani masalah data dalam aplikasi. Layer ini lah yang memiliki akses ke database. Layer ini lah yang mengubah data dari dalam bentuk entities dair entity framework ke dalam objek domain sesuai dengan yang ada pada Domain Layer.

1. Data Access

Pada Data Layer, hanya terdapat satu project di dalamnya yaitu Data Access. Pada Data Access ini lah aplikasi mengakses data melalui Stored Procedure yang telah disusun sebelumnya di database melalui entitiy framework. Data-data dari database tersebut nantinya diterima data access dalam bentuk entitas dari entity framework. Data access ini juga lah yang mengubah data mentah tersebut ke dalam bentuk domain-domain.

2. Domain Layer

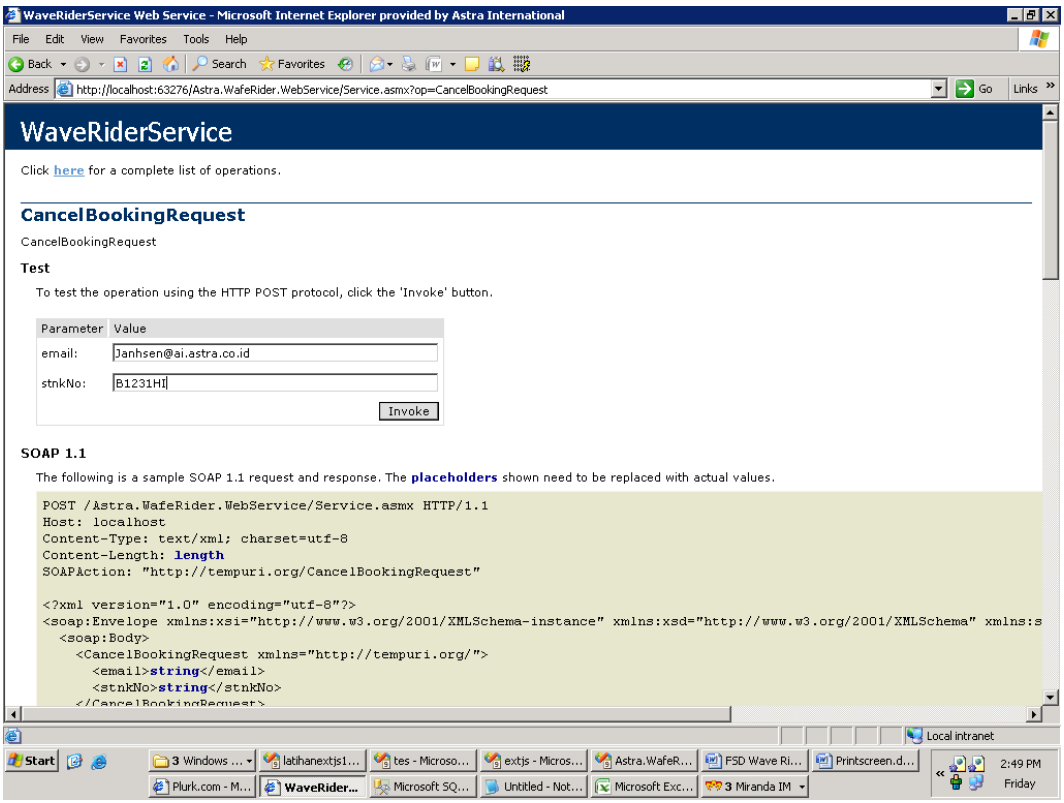
Domain Layer ini berfungsi sebagai objek utama yang diinstance oleh berbagai kelas untuk keperluan transfer data. Basis utama yang digunakan untuk menyusun domain layer adalah request format XML yang dibutuhkan oleh aplikasi pada iPhone.

3. Client Testing Layer

Client Testing Layer ini berfungsi untuk membantu melakukan testing terhadap web service yang telah selesai dibuat. Jika parameter inputan untuk function web service masih bersifat primitive, testing dapat dilakukan langsung melalui UI web servicenya. Namun jika parameter inputan untuk function web service sudah bersifat referensi (menggunakan domain), maka testing tidak dapat dilakukan langsung melalui UI web servicenya. Oleh karena itu lah layer ini dibuat.

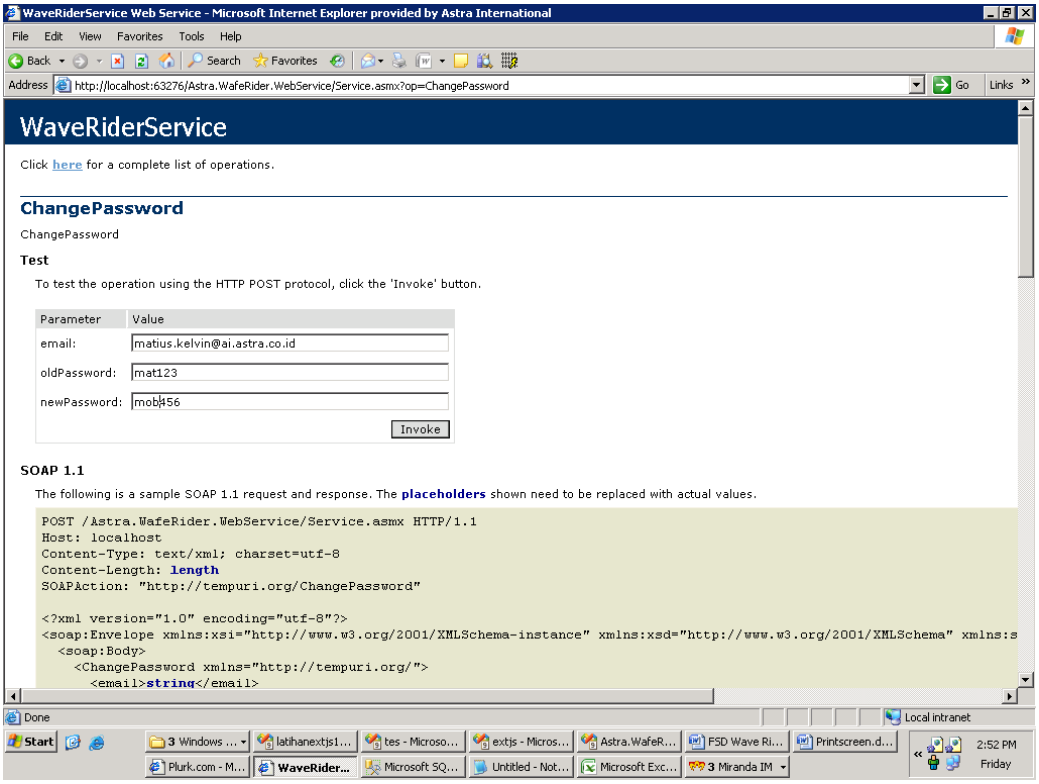
3.4.2 Function Web Service

3.4.2.1 Cancel Booking Request



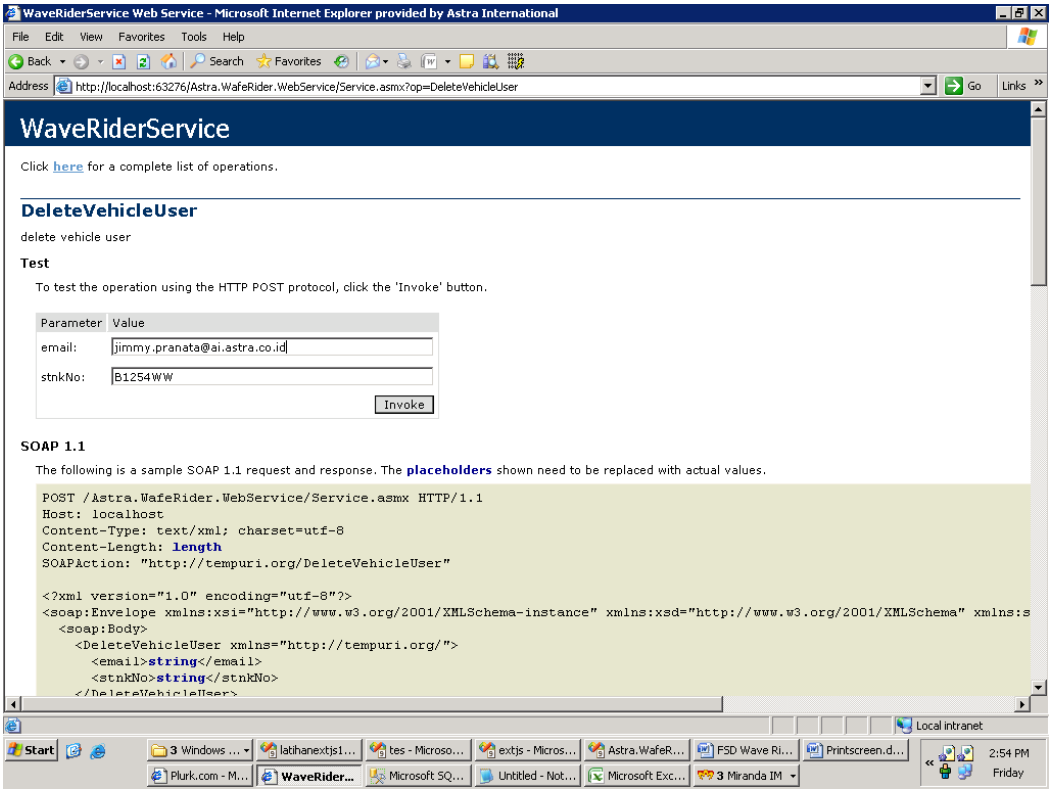
Gambar III.11 Cancel Booking Request

3.4.2.2 Change Password



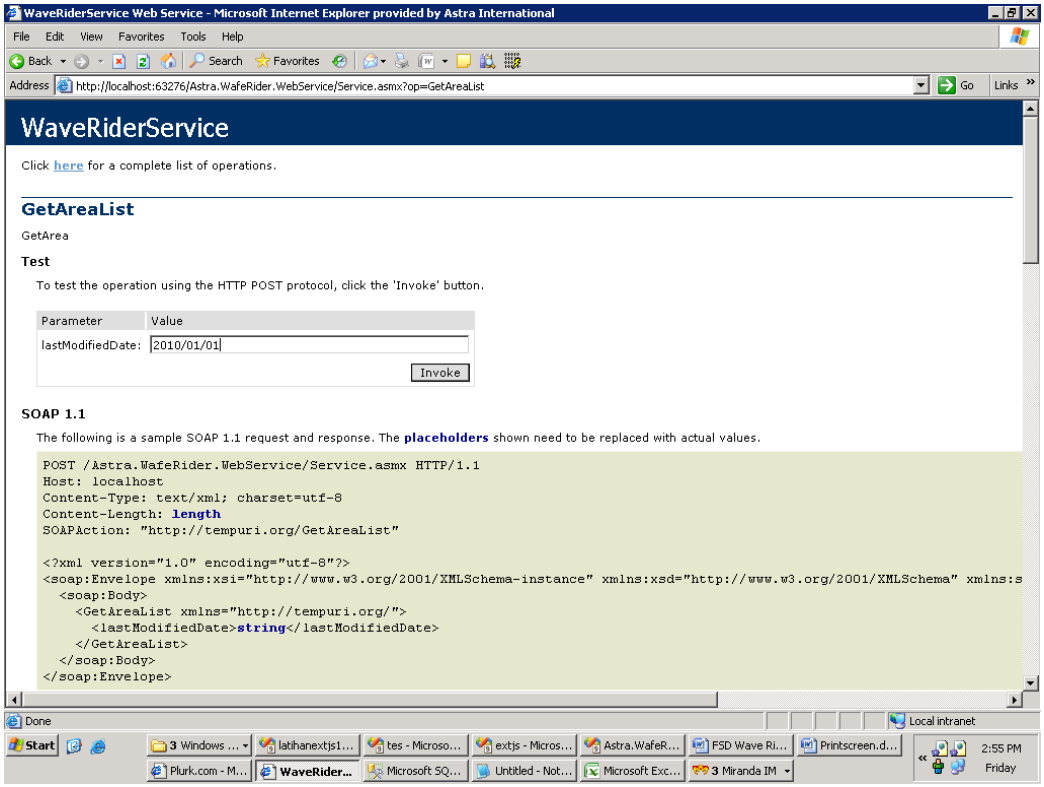
Gambar III.12 Change Password

3.4.2.3 Delete Vehicle User



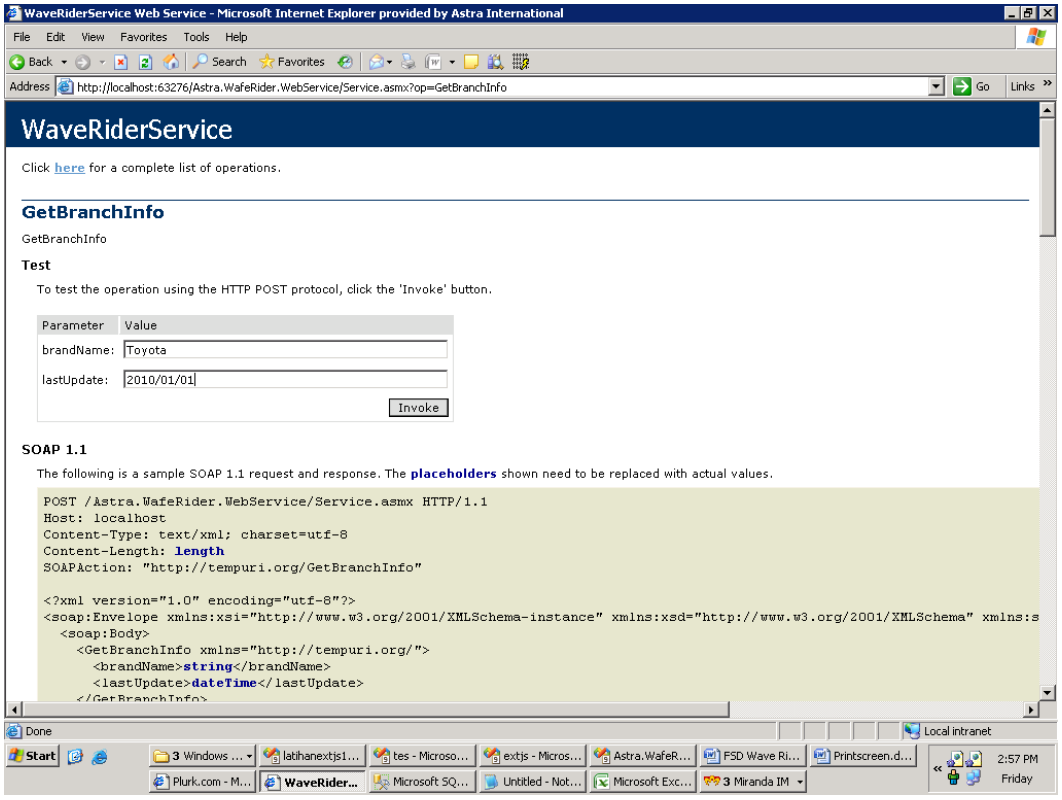
Gambar III.13 Delete Vehicle User

3.4.2.4 Get Area List



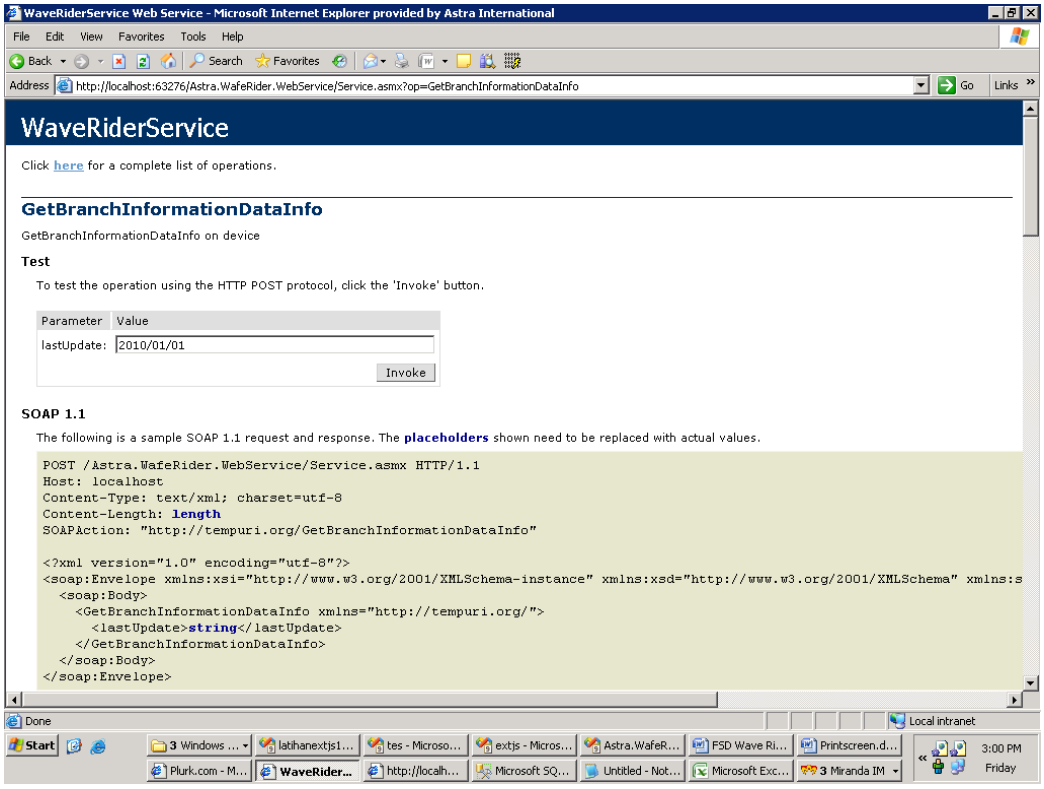
Gambar III.14 Get Area List

3.4.2.5 Get Branch Info



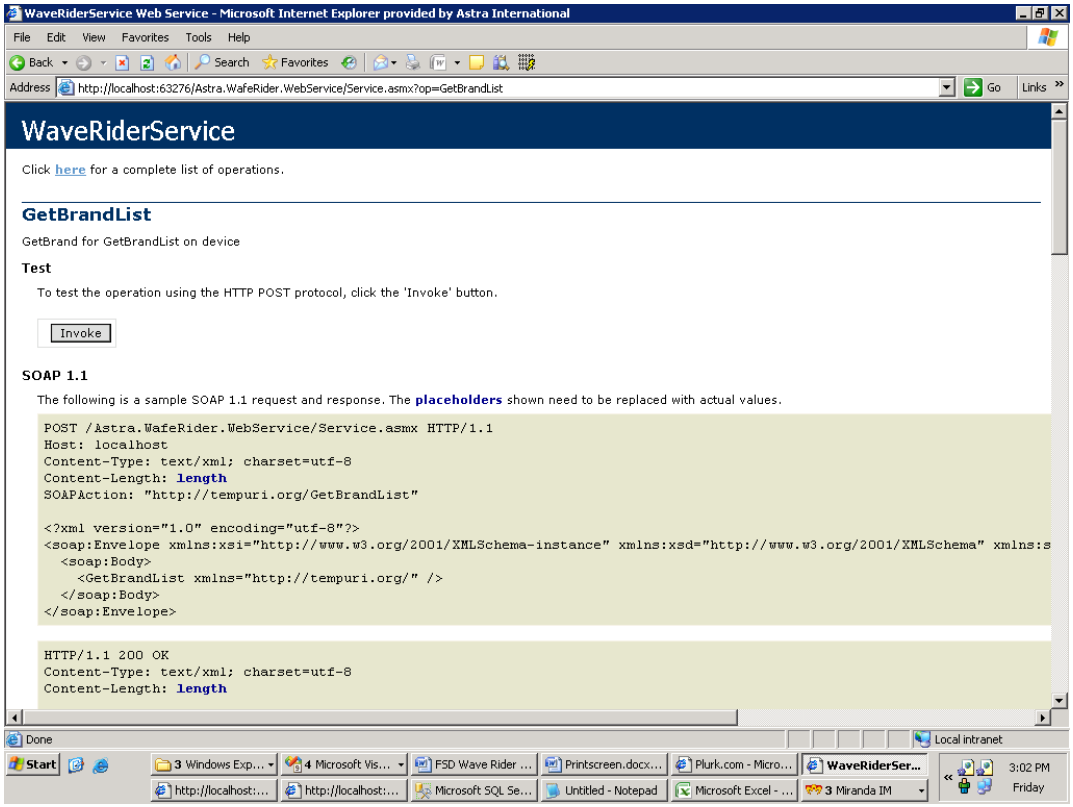
Gambar III.15 Get Branch Info

3.4.2.6 Get Branch Information Data Info



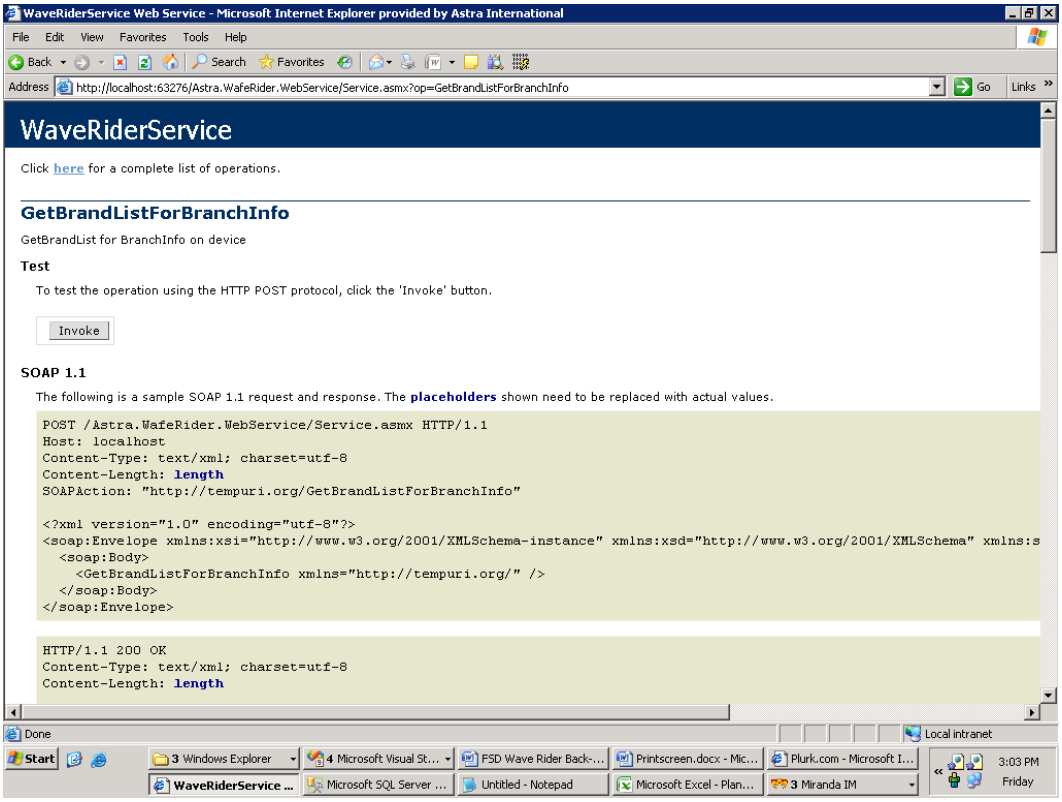
Gambar III.16 Get Branch Information Data Info

3.4.2.7 Get Branch List



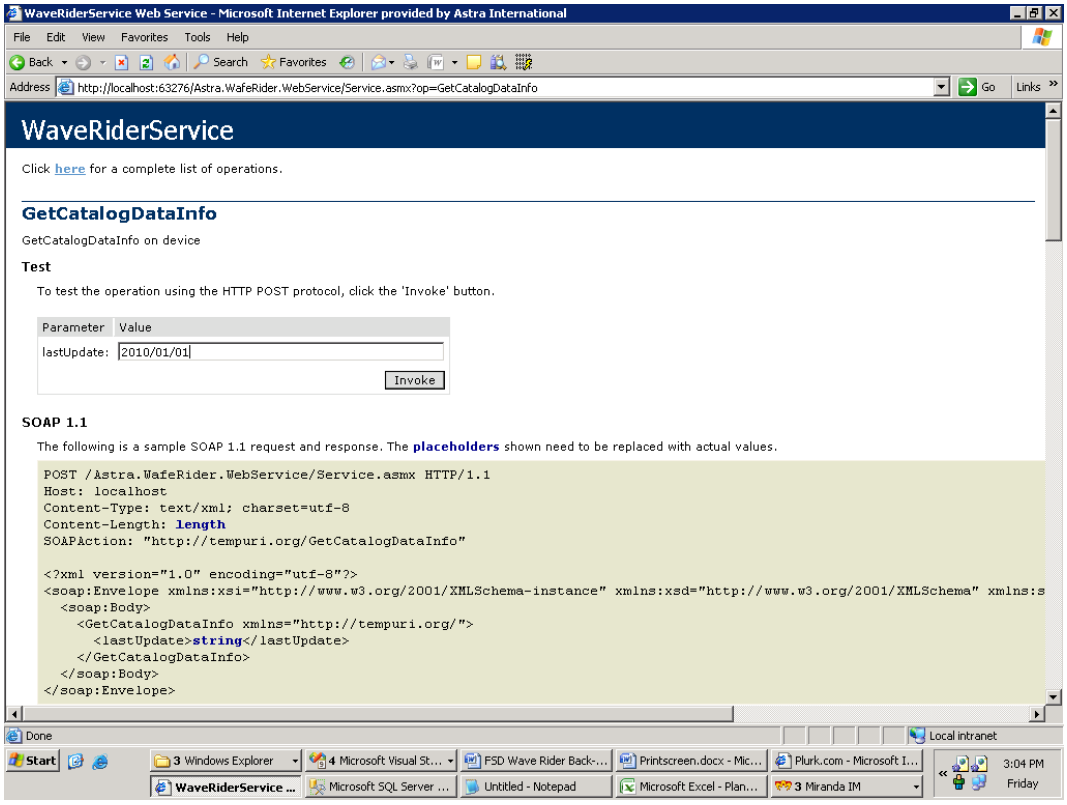
Gambar III.17 Get Brand List

3.4.2.8 Get Brand List for Branch Info



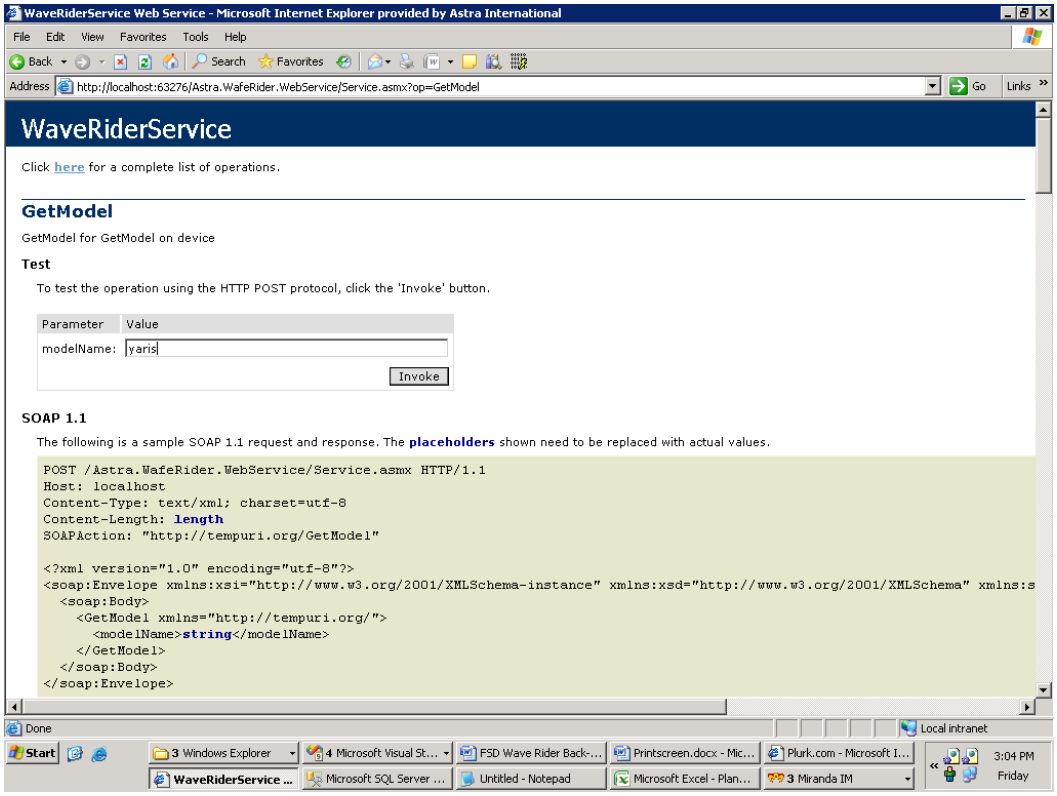
Gambar III.18 Get Brand List for Branch Info

3.4.2.9 Get Catalog Data Info



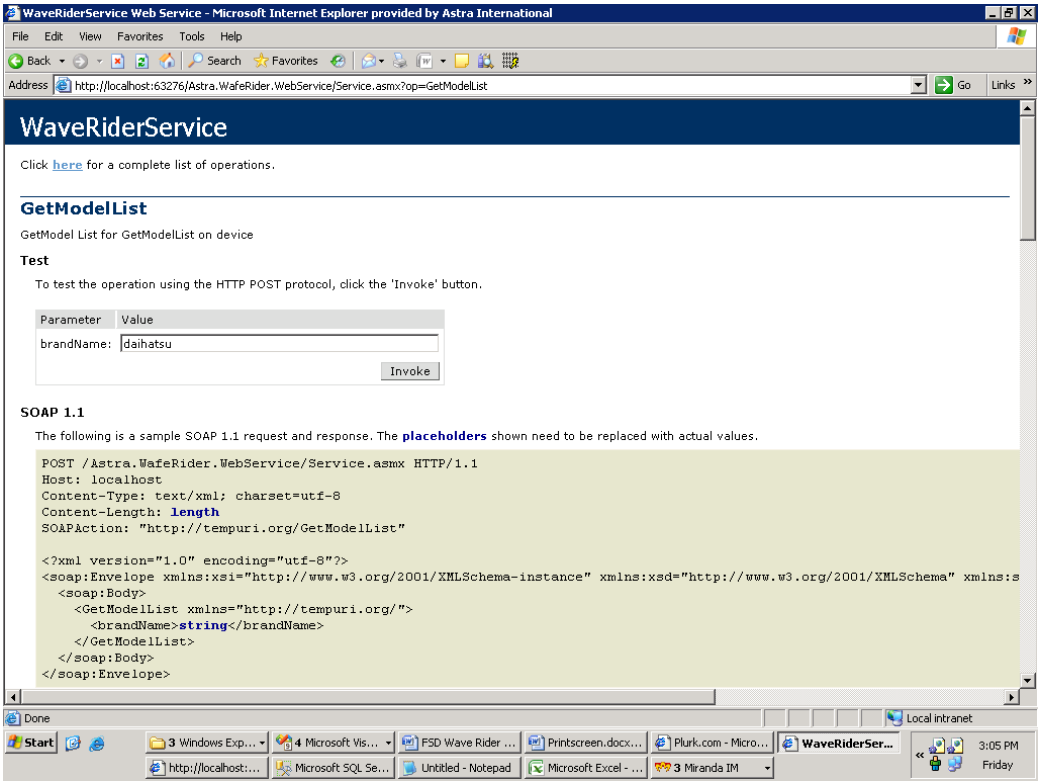
Gambar III.19 Get Catalog Data Info

3.4.2.10 Get Model



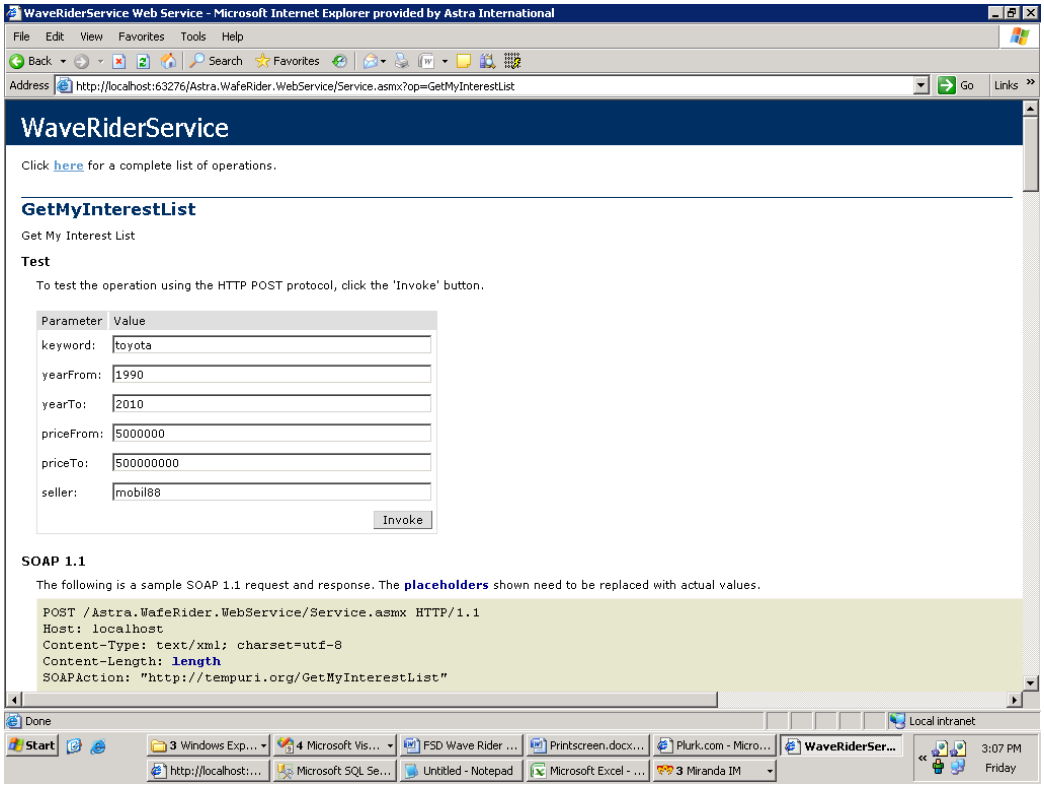
Gambar III.20 Get Model

3.4.2.11 Get Model List



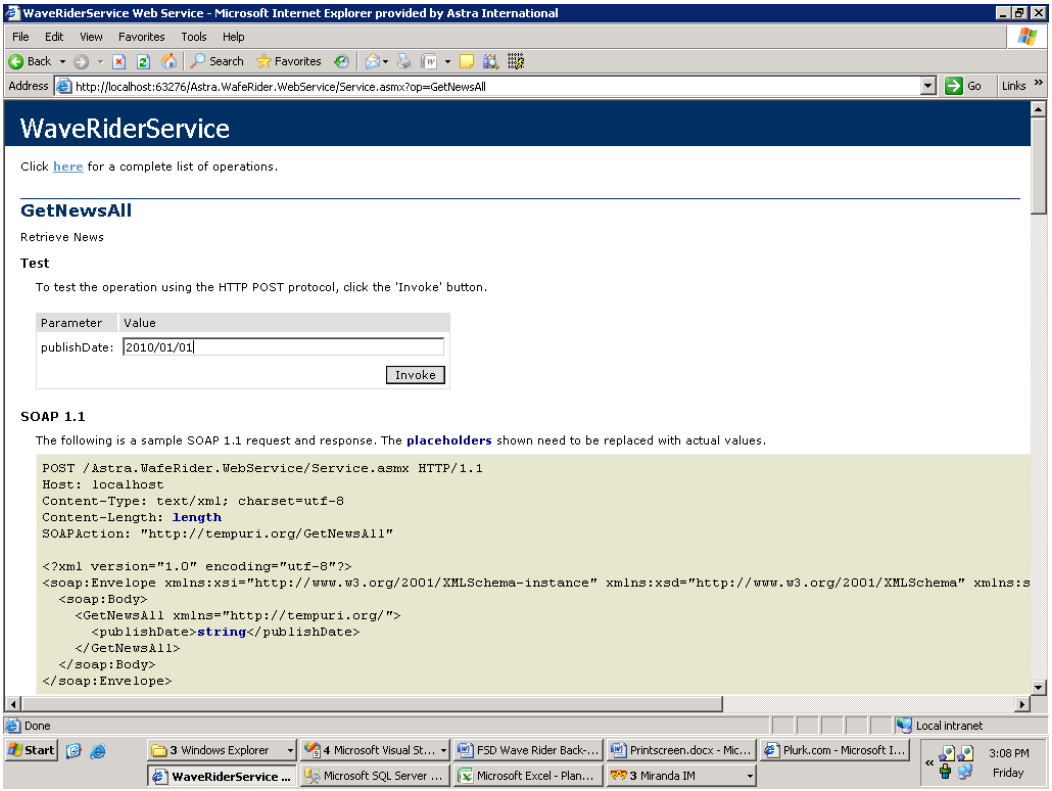
Gambar III.21 Get Model List

3.4.2.12 Get My Interest List



Gambar III.22 Get My Interest List

3.4.2.13 Get News All



Gambar III.23 Get News All