

# REKAYASA KOMPONEN WEB SERVICE GENERAL LEDGER BERBASIS MICROSOFT .NET

**Suhadi Lili, Yudhi Purwananto, Kurniawan E. Permana**

Jurusan Teknik Informatika,  
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5913804  
Email : suhadi@its-sby.edu

## ABSTRAK

*Kebutuhan akan akuntansi menjadi suatu kebutuhan yang begitu mendasar bagi perusahaan baik yang skala kecil, menengah sampai yang besar. Tak pernah terlepas sedikitpun kegiatan ekonomi di suatu perusahaan dari akuntansi dan bagian-bagian yang ada di dalamnya. Salah satu bagian dari akuntansi itu adalah General Ledger atau Buku Besar. General Ledger atau Buku Besar menjadi inti dan pokok dari akuntansi.*

*Pada penelitian ini dikembangkan sebuah komponen yang dapat digunakan untuk membangun web service General Ledger, dimana fungsinya adalah untuk melakukan proses penjurnalan sehingga dapat secara otomatis menghasilkan laporan-laporan keuangan. Tahapan untuk membuatnya adalah dengan waterfall development model dimulai dari pencarian masalah, membuat model class, pengaturan layer hingga aplikasi dapat diimplementasikan. Dalam ujicoba yang dilakukan, komponen telah dapat berfungsi dengan baik pada saat diimplementasikan, dengan melakukan pengisian data master, pengisian jurnal dan pembuatan laporan-laporan akuntansi.*

**Kata kunci :** *Komponen, Web Service, General Ledger, Akuntansi, Microsoft .NET*

## 1. PENDAHULUAN

Dalam dunia modern, kebutuhan akan akuntansi bagi perusahaan telah menjadi bagian yang tak terpisahkan. Hampir di semua perusahaan yang ada saat ini baik itu yang besar, menengah ataupun kecil menggunakan akuntansi sebagai dasar dalam kegiatan keuangannya. Dan salah satu bagian penting dari akuntansi itu adalah General Ledger (buku besar)

Posisi buku besar dalam akuntansi sangat esensial dan secara keseluruhan buku besar berfungsi sebagai berikut :

1. Tempat mencatat perubahan posisi keuangan
2. Pemberian informasi keuangan bilamana diperlukan secara efisien dan efektif
3. Sumber informasi penyusunan daftar sisa (trial balance)
4. Sumber informasi penyusunan laporan keuangan

Pada perusahaan-perusahaan besar, dimana sistem keuangannya telah demikian kompleks dan lengkap, dipastikan terdapat berbagai macam perbedaan platform database ataupun sistem operasi pada masing-masing sub bagian dari perusahaan. Adanya kenyataan seperti demikian menyebabkan proses pengiriman informasi keuangan perusahaan secara global memakan waktu, resiko dan tenaga.

Untuk itulah dibutuhkan layanan Web Service General Ledger, sehingga pada saat perusahaan itu

ingin memposting data dapat dengan langsung dilakukan secara online

Permasalahan-permasalahan yang dihadapi dalam pembuatan komponen ini adalah :

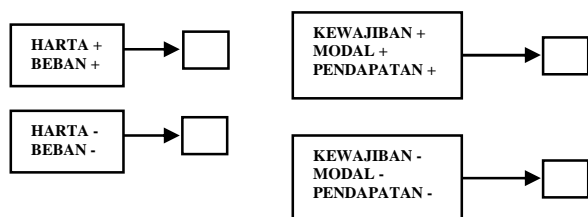
1. Bagaimana merancang arsitektur perangkat lunak komponen General Ledger
2. Bagaimana memodelkan data yang memungkinkan perubahan struktur account
3. Bagaimana membuat laporan-laporan akuntansi yang bersesuaian
4. Bagaimana mengimplementasikan desain sistem Web Service General Ledger menggunakan teknologi berbasis Microsoft .NET

## 2. TEORI AKUNTANSI

Dalam dunia akuntansi modern, pencatatan transaksi-transaksi didasarkan kepada klasifikasi pos-pos keuangan perusahaan ke dalam elemen-elemen yang disebut **perkiraan** (disebut juga rekening / akun atau dalam bahasa Inggris disebut *accounts*). Setiap perkiraan menyatakan posisi keuangan yang bersesuaian yang dinyatakan dalam satuan moneter (mata uang) tertentu. Suatu sistem akuntansi memiliki sekumpulan perkiraan yang disebut **daftar / diagram perkiraan** (*Chart of Accounts*).

Oleh karena kelak terlihat bahwa seluruh perkiraan dalam diagram perkiraan seolah-olah memantau hukum “ketetapan jumlah entitas

ekonomi” yang direalisasikan dalam bentuk neraca / *balance*, sedemikian rupa sehingga akan jelas bagaimana pergerakan entitas ekonomi (yang direpresentasikan dalam satuan moneter tertentu) dari satu pos ke pos lainnya. Sifat dari perkiraan ditunjukkan oleh gambar 1.



**Gambar 1. Sifat Perkiraan**

Pada setiap kegiatan yang berkaitan dengan akuntansi maka dapat dipastikan terdiri dari lima kelompok perkiraan yang berperan besar, yaitu :

1. Aset (*assets*)
2. Kewajiban (*liability*)
3. Modal (*equity*)
4. Pendapatan (*income / revenue*)
5. Biaya (*cost / expenses*)

Pembicaraan mengenai bisnis (dunia usaha) selalu dikaitkan dengan bagaimana cara memperoleh modal (kapital). Modal perusahaan berasal dari hak kekayaan pemilik (*owner's equity*). Setiap modal akan menjadi harta kekayaan (*assets*). Tetapi setiap harta kekayaan bukan hanya bersumber dari modal, melainkan dapat diperoleh dari pihak kreditur yang kita sebut kewajiban (*liabilities*). Kedudukan harta, modal dan kewajiban berkaitan satu sama lain yang disebut unsur-unsur dasar perubahan posisi keuangan. Perubahan posisi keuangan selalu berada dalam keseimbangan, oleh karena itu disebut **Keseimbangan Neraca**. Setiap terjadi perubahan posisi keuangan terlebih dahulu dilakukan transaksi secara tertulis yang kita sebut surat bukti pencatatan. Surat bukti inilah yang harus dicatat kedalam keseimbangan neraca, dan dari persamaan akuntansi dapat disusun laporan keuangan.

Laporan keuangan yang akan dihasilkan oleh system dalam sistem ini adalah :

1. Laporan Jurnal Umum : Laporan tentang aktivitas transaksi yang terjadi
2. Laporan Buku Besar : Laporan tentang catatan perkiraan yang sejenis
3. Laporan RugiLaba : Laporan tentang ikhtisar pendapatan (*income*) dan beban (*Expense*) secara sistematis hingga dapat diketahui dengan jelas laba/rugi bersih
4. Laporan Neraca : Laporan daftar ikhtisar harta, kewajiban dan modal dalam jangka waktu tertentu

### 3. **FRAMEWORK .NET DAN WEB SERVICES**

*Framework .NET* adalah sebuah platform komputing baru yang dirancang untuk menyederhanakan pengembangan aplikasi pada lingkungan terdistribusi pada Internet. Bahasa yang didukung oleh *framework .NET* adalah Visual Basic.NET, C++, ASP.NET, dan C#. *.NET* juga dapat digunakan oleh bahasa pemrograman dari pihak ketiga seperti Eiffel, COBOL, dan Perl.

*Framework* adalah komponen dari sistem operasi Microsoft Windows yang menyediakan model pemrograman untuk membangun, mendeploy dan menjalankan aplikasi berbasis web, aplikasi smart client dan XML web service. *Framework .NET* memberikan nilai bisnis lebih yaitu yang membutuhkan waktu yang lebih cepat untuk suatu aplikasi bisa dijual dipasaran, meningkatkan fleksibility system dan mengurangi biaya, baik biaya mendesain, membangun ataupun biaya pembuatan.

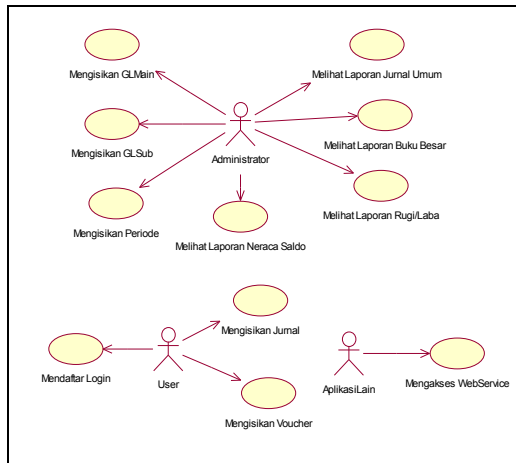
Web Services merupakan salah satu fitur penting yang dibawa oleh *Framework .NET*. Web Services dibuat untuk mengatasi perbedaan sistem operasi, bahasa pemrograman, dan model objek. Web Services terdiri dari objek-objek dan metode-metode yang dapat dipanggil dari client apapun melalui HTTP. Web Services dibangun diatas SOAP. SOAP merupakan protokol standar untuk pengiriman data lewat HTTP. SOAP memudahkan pengiriman data dan struktur lewat web.

SOAP memungkinkan aplikasi-aplikasi yang ditulis dengan bahasa yang berbeda-beda dan berjalan di berbagai platform untuk membuat *Remote Procedure Call (RPC)* secara efektif meskipun melalui firewall. SOAP menggunakan port 80 yang disediakan untuk jalur HTTP sehingga memungkinkan pemanggilan prosedur-prosedur yang berada di balik firewall.

*Framework .NET* akan membangkitkan file WSDL (*Web Services Description Language*) di server. Sedangkan pada web server host halaman-halaman .aspx dibentuk WSDL Proxy. WSDL merupakan format XML yang menggambarkan bagaimana sebuah Web Service dapat dipanggil, argumen-argumen apa yang dibutuhkan, dan sebagainya.

### 4. **PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK**

Perancangan proses secara keseluruhan dimodelkan dengan menggunakan UML. Diagram yang digunakan disini adalah *Use Case Diagram* serta *Sequence Diagram* dan *Class Diagram* dari tiap-tiap *Use Case*. Pertama-tama, perlu dibuat *Use Case Diagram* untuk menggambarkan kerja sistem secara global melalui aktor-aktor yang terlibat beserta peran yang dilakukan oleh tiap aktor. Gambar 2. menunjukkan *Use Case Diagram* untuk sistem yang dibuat.



**Gambar 2.** Use case diagram dari sistem.

Dari *Use Case Diagram* tersebut dapat dilihat adanya dua jenis aktor yang terlibat dalam sistem yaitu :

1. Administrator, sebagai pengelola dan pemelihara sistem.
2. User, sebagai pihak luar yang mengakses informasi, dalam hal ini adalah untuk melakukan proses penjurnalan.

Perancangan data fisik ini akan menggambarkan struktur fisik dari penyimpanan basis data dan menggambarkan detail lengkap dari penyimpanan data dan alur pengaksesan untuk basis data, yaitu bagaimana data tersebut disimpan di dalam basis data. Bentuk dari rancangan data fisik dapat dilihat pada gambar 4.

Komponen perangkat lunak yang dibuat pada penelitian ini bertujuan untuk melakukan penjurnalan terhadap system untuk kemudian akan dapat dihasilkan laporan keuangan. Sebagai contoh di sini penjurnalan yang dilakukan oleh user membutuhkan urutan proses yang harus dilakukan.

Urutan proses yang harus dilakukan adalah administrator akan melakukan memasukkan data master yang terdiri dari data GLM (General Ledger Main), data GLS (General Ledger Sub) dan data periode yang sekaligus mengaktifkan periode yang diinginkan aktif. Setelah itu user dapat melakukan proses penjurnalan ataupun memasukkan data pada transaksi Voucher.

Setelah itu akan dapat dihasilkan laporan-laporan yang dikehendaki, dalam hal ini adalah laporan jurnal umum, laporan buku besar, laporan rugi laba dan laporan neraca. Untuk mendapatkan laporan yang diinginkan, terlebih dulu dimasukkan kriteria dan kategori laporan yang diinginkan. Kriteria yang dapat dipilih adalah untuk GLMID, GLSID, tanggal mulai dan tanggal akhir. Dan untuk diketahui bahwa laporan yang dihasilkan adalah proses yang didapat dari data pada penjurnalan.

### Arsitektur Three Tier

Latar belakang dari arsitektur three tier adalah pemisahan aplikasi ke dalam tiga komponen besar service yaitu layer presentasi atau layanan user, layer middle atau layanan bisnis dan layer data atau layanan data. Ketiga layer ini tidak harus berhubungan secara fisik dalam satu komputer di satu lokasi tetapi lebih seperti pembagian layer logika dalam suatu aplikasi.

Seperti yang ditunjukkan dalam gambar 3, pada aplikasi yang di buat untuk penelitian ini, arsitektur three tier dapat dikelompokkan secara lebih khusus Web Client Application, XML Web Service dan OLEDB. Sehingga pengkhususan pada layer Business Object atau XML Web Service itu dapat lebih dipersempit dalam komponen yang telah disediakan secara khusus pada framework .NET.

Aplikasi web yang dibuat menggunakan arsitektur *three-tier* yang terdiri dari :

1. *Application Layer*, atau disebut juga *Presentation Layer*, menyediakan antarmuka untuk pengguna sistem. Dalam sistem ini, *Application Layer* berisi kumpulan Web Form (.aspx).
2. *Business Service Layer*, berisi aturan-aturan bisnis dari aplikasi. *Business Service* menghubungkan antara *Application* dan *Data Access Layer*.
3. *Data Access Layer*, merupakan *layer* yang berhubungan langsung dengan basis data, misalnya melalui query.

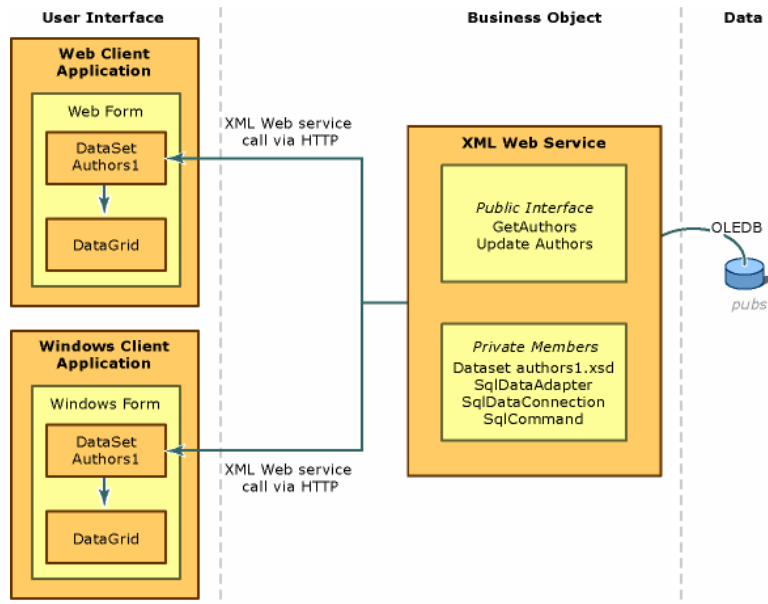
Pembagian ini dimaksudkan untuk menunjukkan modularitas dari perangkat lunak serta memastikan web service yang digunakan bekerja dengan baik di berbagai komputer. Server kedua terdapat pada server yang berada pada sistem yang sama.

Penjelasan dari arsitektur yang ada di gambar 5 adalah sebagai berikut :

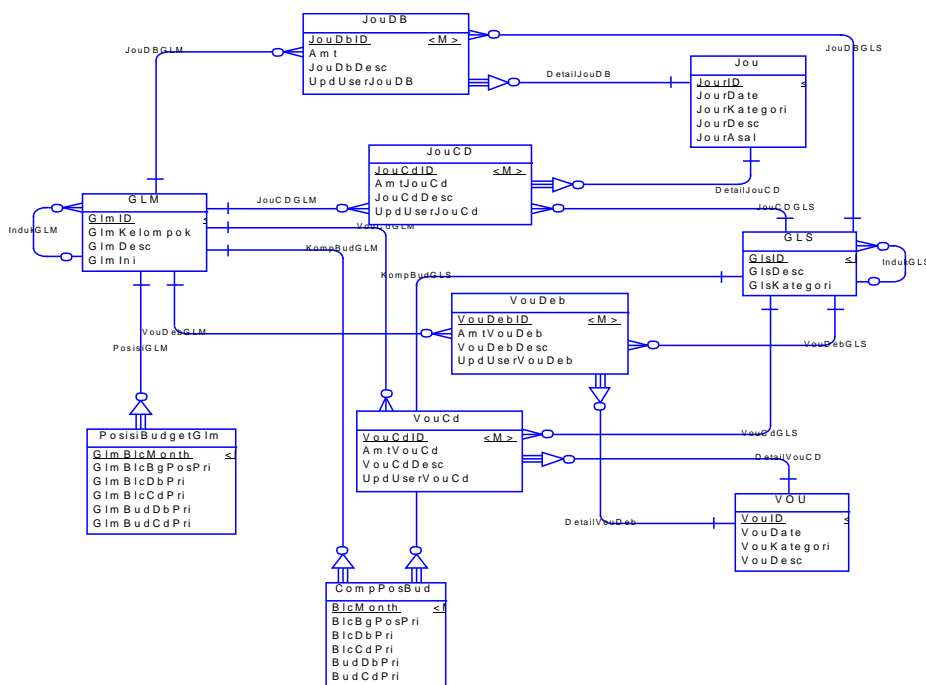
1. Administrator memanggil halaman aspx lewat browser. Browser melakukan HTTP request ke web server dan web server akan mengirimkan output HTML ke browser.
2. Dilakukan proses pengisian data master.
3. Proses penjurnalan dapat dilakukan dengan didasarkan pada data master yang telah dimasukkan.
4. Penjurnalan dapat dilakukan langsung dengan mengakses pada web server admin ataupun pada web server user.
5. Halaman-halaman aspx khusus untuk administrator yang berada di Application Layer di server utama akan mengakses class-class yang ada di Business Service Layer secara langsung.
6. Halaman-halaman aspx untuk pengguna akan mengakses fungsi-fungsi di Business Service Layer di server admin melalui web service.
7. Server yang diakses pengguna atau server IIS membuat Web Service Proxy sebagai replikasi

dari metode-metode yang ada di web service agar metode-metode tersebut tidak dieksekusi langsung di server user. Hubungan ke web service ini dilakukan melalui SOAP dan web service mengirimkan hasil yang diminta dalam bentuk XML.

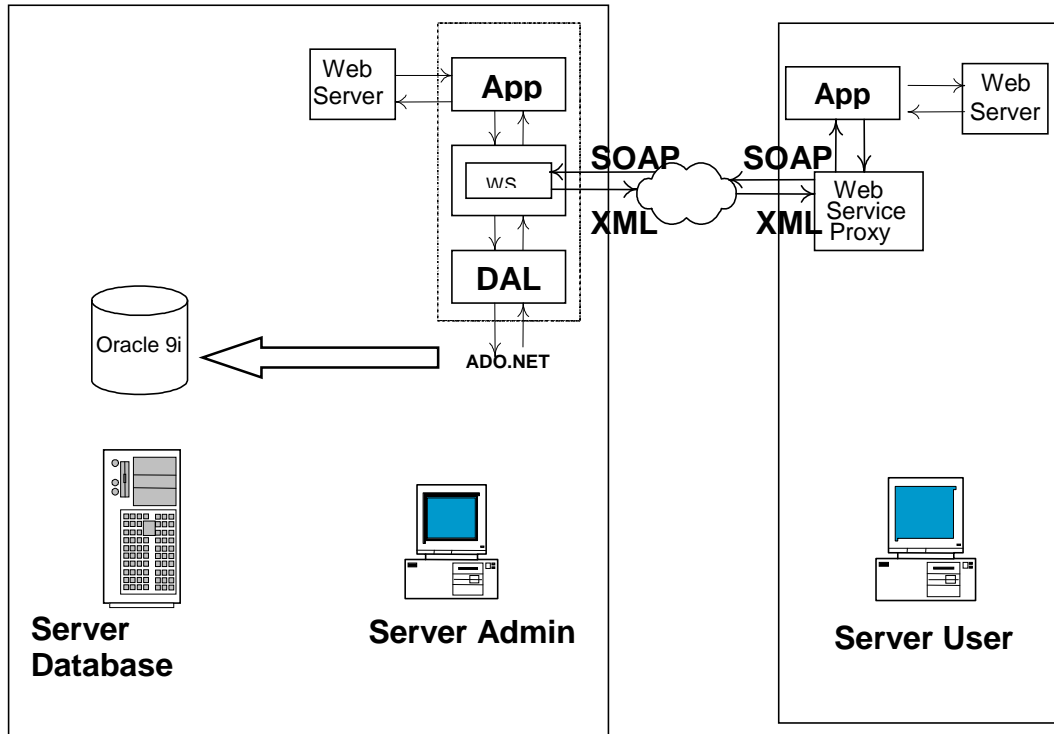
Pengguna memanggil halaman aspx lewat browser.



**Gambar 3. Arsitektur Three Tier**



**Gambar 4. Diagram Data Fisik**



Gambar 5. Arsitektur sistem

Dalam pengembangannya, aplikasi ini disusun dalam beberapa layer sebagai berikut :

1. Data Access Layer : adalah layer dimana transaksi haruslah melalui layer ini terlebih dahulu sebelum mendapatkan, menginputkan ataupun menghapus data.
2. Business Rules Layer : adalah layer yang berisi tentang logika dan aturan yang dipakai dengan segala kemungkinan-kemungkinan transaksi yang berlaku dalam aplikasi itu. Biasanya juga berisi kondisi-kondisi yang diperkirakan.
3. User Service Layer : adalah layer yang berhubungan langsung dengan user, berupa form, view, laporan dan sebagainya. Sehingga pada layer ini sering menjadi acuan dari sebagian orang untuk menentukan bagus tidaknya aplikasi. Padahal ini adalah salah satu layer dari beberapa layer yang ada. Dan semuanya membentuk suatu kesatuan yang sinergis
4. GLServices : adalah layer berhubungan dengan Web Service. Pada layer ini berisikan method-method yang hendak diservicekan. Khusus pada layer ini hanya mereferensikan methodnya dari layer Data Access ataupun Business Logic.

## 5. HASIL UJI COBA

Pada uji coba ini diberikan dua skenario yang ditujukan untuk mengetahui fungsionalitas dari perangkat lunak yang dibuat, yaitu :

1. Pengisian jurnal atau penjurnalan.
2. Dihasilkan laporan keuangan

The screenshot shows the 'Entry Journal' form in the 'GENERAL LEDGER' application. The form includes fields for Date, Category, Transaction ID, and Description. It also has a 'Detail' section with fields for GL Sub ID, GL Main ID, and Description. A table at the bottom shows a list of journal entries with columns for GLMID, GLSID, DESKRIPSI, DK, and JUMLAH.

GLMID	GLSID	DESKRIPSI	DK	JUMLAH	Remove
1000101	0101	Pajak Kendaraan Bermotor	D	100000000	Remove

Gambar 6. Form untuk memasukkan jurnal.

### Uji Coba Skenario Pertama

Skenario ini dimulai dengan user melakukan proses penjurnalan dengan data terlampir. Pada gambar 6 ditunjukkan form pada waktu dilakukan proses penjurnalan.

Pada saat ini akan terjadi proses pengecekan terhadap tanggal, transaksi ID dan keseimbangan neraca. Jika salah satu tidak memenuhi maka tidak akan dapat dilakukan proses penyimpanan. Kemudian yang perlu diperhatikan adalah karena data GLM ataupun GLS memiliki induk masing-masing, maka data yang telah dimasukkan akan secara otomatis mengupdate data induknya. Hanya saja pada daftar detail yang ditampilkan adalah data GLM dan GLS yang dipilih langsung. Hasil dari proses ini akan dapat dilihat pada saat melihat hasil laporan.

### Uji Coba Skenario Kedua

Skenario kedua dimulai dengan memasukkan kriteria laporan yang diinginkan. Terdapat form untuk memasukkan kriteria, seperti ditunjukkan oleh gambar 7. Dan dihasilkan empat laporan keuangan, yang meliputi Laporan Jurnal Umum, Laporan Buku Besar, Laporan Neraca Saldo, dan Laporan Laba Rugi, seperti yang ditunjukkan oleh gambar 8 sampai gambar 11.

Setelah dimasukkan kriteria yang diinginkan yaitu GLSID dan atau GLMID, tanggal mulai dan tanggal akhir, maka akan didapatkan laporan yang dikehendaki.



**Gambar 7. Form untuk memasukkan kriteria yang diinginkan.**

### Analisis Hasil

Dari uji coba yang dilakukan terhadap perangkat lunak yang dibuat, terlihat perangkat lunak telah bekerja dan menjalankan semua fungsi yang diharapkan.

Tanggal	No. Bukti	GLM ID	Nama Akun	D/K	Nilai
01/01/2003	1	3000306 4000302	Pembangunan Gedung Gubernur Bangunan Gedung Struktur Binaan	0 K	75.000.000 50.000.000
01/01/2003	2	3000301 5000302	Penerimaan bantuan tanah Tanah Berkas	0 K	100.000.000 100.000.000
01/01/2003	3	3000309 5000301	Penerimaan hibah kendaraan Arah Angkutan Bibah	0 K	50.000.000 50.000.000
01/01/2003	4	3000301 5000301	Penerimaan dana hibah Kas dan Bibah Bibah	0 K	50.000.000 50.000.000

**Gambar 8. Laporan Jurnal Umum.**

Pada skenario pertama hal ini terlihat dari proses penjurnalan yang dapat melaksanakan pengecekan dan memberikan pesan kesalahan pada saat salah memasukkan data dan juga berhasil melakukan penyimpanan dalam database.

Proses dihasilkan laporan keuangan juga telah berhasil. Terlihat dari pelaksanaan skenario kedua yang menampilkan laporan keuangan sesuai dengan kriteria yang dimasukkan. Adanya kriteria tersebut menunjukkan kemudahan bagi pengguna dalam mendapatkan laporan yang diinginkan.

Tanggal	Deskripsi	Debet	Kredit	Saldo
01/01/2003	Penerimaan Dana Hibah	50.000.000	0	50.000.000
01/01/2003	Penerimaan Hibah kendaraan	50.000.000	100.000.000	50.000.000
TOTAL		100.000.000	100.000.000	

**Gambar 9. Laporan Buku Besar.**

Kode	GLM Description	Saldo Awal	Mutasi Debet	Mutasi Kredit	Saldo Akhir
3	AKTIVA	0	275.000.000	0	275.000.000
30001	AKTIVA LANCAR	0	50.000.000	0	50.000.000
3000101	Kas dan Bank	0	50.000.000	0	50.000.000
3000102	Sisa Belanja	0	0	0	0
3000103	Piutang Prabadi	0	0	0	0
3000104	Piutang Pajak	0	0	0	0
3000105	Piutang Lain-lain	0	0	0	0
3000106	Persediaan Bibah	0	0	0	0
3000107	Manfaat	0	0	0	0
3000107	Batasnya Dibayar Dimuka	0	0	0	0
30002	INVESTASI JANGKA PANJANG	0	0	0	0
3000201	Investasi dalam Saham	0	0	0	0
3000202	Investasi dalam Obligasi	0	0	0	0
30003	AKTIVA TETAP	225.000.000	0	225.000.000	225.000.000
3000301	Tanah	100.000.000	0	100.000.000	100.000.000
3000302	Jalan dan Jembatan	0	0	0	0
3000303	Bangunan As (Hutang)	0	0	0	0
3000304	Instansi	0	0	0	0
3000305	Jaringan	0	0	0	0
3000306	Bangunan Gedung	75.000.000	0	75.000.000	75.000.000
3000307	Musiman dan Tugu	0	0	0	0
3000308	Arah dan Benda	0	0	0	0
3000309	Alat Angkutan	50.000.000	0	50.000.000	50.000.000
3000310	Ukuk	0	0	0	0
3000311	Alat Perikanan	0	0	0	0
3000312	Alat Kantor dan Rumah Tangga	0	0	0	0
3000313	Alat Studio dan Alat	0	0	0	0

**Gambar 10. Laporan Neraca Saldo.**

Kode	GLN Description	Saldo Awal	Mutasi Debet	Mutasi Kredit	Saldo Akhir
100010	Pajak Kendaraan Bermotor	0	0	0	0
1000102	Pajak Kendaraan Di Atas Air	0	0	0	0
1000103	Pajak Kendaraan Di Bawah Air	0	0	0	0
1000104	Pajak Salinan Dokumen Di Atas Air	0	0	0	0
1000105	Pajak Salinan Dokumen Di Bawah Air	0	0	0	0
1000106	Pajak Penghasilan dan Pemertanian Air Elewah Tanah	0	0	0	0
1000107	Pajak Penghasilan dan Pemertanian Air Pemukiman	0	0	0	0
100020	Pembelian Perbaikan	0	0	0	0
1000201	Pembelian Perbaikan Fasilitas	0	0	0	0
1000202	Pembelian Perbaikan Fasilitas Dasar	0	0	0	0
1000203	Pembelian Perbaikan Biaya Cetak Peta	0	0	0	0
1000204	Pembelian Perbaikan Kapal Perikanan	0	0	0	0
100030	Bagian Laba Perusahaan Milik Daerah	0	0	0	0
1000301	Bagian Laba Lembaga Keuangan Bank	0	0	0	0
1000302	Bagian Laba Lembaga Keuangan Non-Bank	0	0	0	0
1000401	Bip Hasi Pajak	0	0	0	0

**Gambar 11. Laporan Laba Rugi.**

## 6. KESIMPULAN

Beberapa kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut :

1. Rekayasa Komponen WebService General Ledger ini memiliki arsitektur multitier yang dirancang untuk keperluan pengelolaan penjumlahan General Ledger, yaitu Data Access Layer, Business Logic Layer, User Service Layer dan Web Service Layer. Kesemuanya merupakan sebuah bagian atau komponen aplikasi yang membentuk integrasi dan dapat digunakan kembali setiap saat.
2. Pembuatan sistem aplikasi ini memungkinkan dapat dilakukannya proses perubahan struktur akuntansi sesuai dengan kebutuhan sistem akuntansi yang berlaku. Struktur pemodelan ini dibentuk sedemikian rupa sehingga perubahan yang dapat dilakukan selain perubahan akuntansi, juga dapat dilakukan pada struktur organisasi perusahaan.
3. Pada tahap laporan telah dibuat empat macam bentuk laporan yang menampilkan data-data

penjumlahan yang telah dilakukan. Laporan itu adalah laporan penjumlahan, laporan buku besar, laporan rugi laba dan laporan neraca.

## Kemungkinan Pengembangan

Agar dapat menghasilkan sistem Web Service yang lebih kompleks, pengembangan dapat diarahkan dengan mempertimbangkan hal-hal berikut:

1. Melakukan proses penjumlahan yang terintegrasi dengan bagian-bagian akuntansi yang lain seperti Inventory, Invoice, Account Payable ataupun Account Receivable.
2. Untuk menghasilkan data-data penjumlahan yang lebih reliable dan mendapatkan daya guna yang lebih besar, aplikasi ini dapat dihubungkan dengan bagian akuntansi yang lain menjadi aplikasi enterprise

## 7. DAFTAR PUSTAKA

1. Atqona, "General Ledger", ATS Accounting 1.0, 2000
2. Edi Purwono, "Mendesain dan memprogram aplikasi akuntansi (General Ledger) dengan FoxBase", <http://members.tripod.com/~tipon/tip.htm>, 1998
3. IbuySpy, "IbuySpy Application Series", [www.ibuyspy.com](http://www.ibuyspy.com), 2002
4. Microsoft, "MSDN Librarys", <http://msdn.microsoft.com>, 2002
5. Crowe, Malcolm K. "Compiler Writing Tools Using C#", <http://cis.paisley.ac.uk/crow-ci0>, 2002
6. GotDotNet, "The Microsoft.NET Framework Community" <http://www.gotdotnet.com>, 2003