

PERANCANGAN DAN PEMBANGUNAN PERANGKAT LUNAK BERORIENTASI ARSITEKTUR SERVIS (SOA) DENGAN PENDEKATAN WORKFLOW PADA DOMAIN CASH BANK DAN GENERAL LEDGER ERP

Dwi Sunaryono¹⁾, Riyanarto Sarno²⁾, Victor Hariadi³⁾, Yusuf Kurniawan⁴⁾

Teknik Informatika, Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Jl. Arief Rahman Hakim, Surabaya, 60111 Indonesia

E-mail : dwi@its-sby.edu¹⁾, riyanarto@its-sby.edu²⁾, victor@its.sby.edu³⁾,
ucubkurniawan@gmail.com⁴⁾

Abstrak

Pada perusahaan, skala kecil, menengah ataupun besar, akuntansi keuangan adalah bagian penting. Pengelolaan akuntansi keuangan dalam perusahaan antara lain adalah manajemen kas / bank dan buku besar. Pada paper ini, dikembangkan sebuah perancangan dan pembangunan perangkat lunak Cash Bank (CB) dan General Ledger (GL). CB berfungsi untuk menangani sistem pencatatan pemasukan dan pengeluaran kas, bank, cek dan giro. GL digunakan untuk pencatatan ayat jurnal, mengatur rekening, mengatur periode akuntansi serta mencetak laporan laba rugi dan neraca keuangan. Aplikasi ini menggunakan arsitektur berorientasi servis (SOA) dengan sistem Model-View-Controller (MVC) dan Workflow berbasis .NET. Hasil pengujian menunjukkan bahwa perancangan dan pembuatan perangkat lunak berhasil melakukan pencatatan berbagai fungsionalitas penting CB dan GL antara lain pemasukan dan pengeluaran melalui kas, bank, cek dan giro, pencatatan ayat jurnal, mengatur rekening, mengatur periode akuntansi serta mencetak laporan laba rugi dan neraca keuangan. Secara umum, perangkat lunak telah berhasil memenuhi spesifikasi CB dan GL yang dibutuhkan pada sistem ERP dan terintegrasi dengan domain fungsi yang lain.

Kata kunci: CB, GL, MVC, SOA, Workflow

1. PENDAHULUAN

Kondisi dunia bisnis saat ini telah berkembang menjadi semakin kompleks, semakin kompetitif, bergerak dengan cepat serta semakin sulit untuk diprediksi. Agar dapat bersaing dan sukses, perusahaan perlu memadukan bisnis dan sumber daya IT yang dimiliki agar dapat secara fleksibel mengakomodasi adanya perubahan untuk kemudian dilakukan adaptasi terhadap perubahan tersebut secara cepat dan tepat.

Dari pembahasan diatas dapat disimpulkan bahwa kebutuhan akan aplikasi Enterprise Resource Planning (ERP) dalam perusahaan merupakan kebutuhan yang penting, agar perusahaan dapat memadukan bisnis dan sumber daya. Pada ERP terdapat banyak sekali modul-modul yang dibuat dan di integrasikan, antara lain adalah Finance / Accounting, Sales, Customer Relationship Manager, Inventory, Manufacturing, Human Resource Management, dan lainnya. Pada paper ini, Aplikasi dibangun pada fungsional domain Cash Bank dan General Ledger yang termasuk dalam paket Finance / Accounting.

Cash Bank atau CB adalah domain fungsional yang menangani pencatatan transaksi penerimaan dan pengeluaran uang. Cash Bank melakukan pencatatan transaksi berdasarkan Invoice yang diterima dari Account Payable dan Account Receivable, kemudian setiap pencatatan tersebut akan diakumulasi dan diproses menjadi sebuah jurnal. Dan General Ledger atau GL adalah bagian penting yang dibutuhkan oleh direksi perusahaan untuk mengambil keputusan. Karena pada bagian ini mereka bisa mengetahui seluruh informasi keuangan yang dibutuhkan dalam bentuk jurnal atau laporan sesuai dengan kebutuhan.

Selanjutnya, tentu saja masalah integrasi dengan sistem menjadi suatu tantangan tersendiri, sehingga konsep CB dan GL ini perlu dibangun dalam sistem yang berbasis servis (*service-oriented*). Arsitektur sistem ini lebih dikenal dengan nama Service Oriented Architecture (SOA). SOA secara lebih spesifik berhubungan dengan pembangunan layanan bisnis bebas yang dapat dikombinasikan menjadi proses bisnis pada level tinggi dan solusi dalam konteks enterprise. Dengan menerapkan metode SOA, permasalahan yang ada terkait dengan perkembangan dunia bisnis dapat terselesaikan. SOA merupakan suatu metode pengembangan berbasis arsitektur yang memodularisasi sistem informasi menjadi services dengan cara mengelompokkan proses bisnis perusahaan. Dengan kombinasi hal-hal di atas, maka suatu bentuk sistem Cash Bank dan General Ledger yang ideal mampu memberi nilai positif yang optimal bagi perusahaan.

2. TINJAUAN PUSTAKA

2.1 Enterprise Resource Planning

ERP atau perencanaan sumber daya perusahaan adalah sistem informasi yang diperuntukkan bagi perusahaan *manufacturing* maupun jasa yang berperan mengintegrasikan dan mengotomasi proses bisnis yang berhubungan dengan aspek operasi, produksi maupun distribusi perusahaan tersebut.



Gambar 1 Sistem Global ERP

Jadi ERP adalah sebuah termologi yang diberikan kepada sistem informasi yang mendukung transaksi atau operasi sehari-hari dalam pengelolaan sumber daya perusahaan meliputi dana, manusia, mesin, suku cadang, waktu, material dan kapasitas [1]. Keuntungan penggunaan ERP diantaranya adalah integrasi data keuangan, standarisasi proses operasi, standarisasi data dan informasi, penurunan *inventory* dan tenaga kerja, peningkatan servis level dan kontrol keuangan dan penurunan waktu yang dibutuhkan untuk mendapatkan informasi. Sistem global dari ERP bisa dilihat pada Gambar 1

2.2 Manajemen Kas / Bank

Domain fungsional Cash Bank secara umum menangani pencatatan transaksi penerimaan dan pengeluaran uang. Cash Bank melakukan pencatatan transaksi berdasarkan Invoice yang diterima, kemudian setiap pencatatan tersebut akan diakumulasi dan diproses menjadi sebuah jurnal [2]. Jurnal-jurnal yang dihasilkan cash bank akan dimanfaatkan oleh fungsional domain lain seperti GL untuk membuat laporan keuangan. Dalam perangkat lunak terdapat empat proses bisnis yang ditangani dalam domain CB, yaitu Cash Management, Bank Management, Cheque Management, Giro Management

2.3 General Ledger

Buku besar umum (*General Ledger*) adalah buku akun keuangan, yang mencerminkan pengaruh keuangan dari transaksi setelah dibukukan dari berbagai jurnal [3]. Secara umum buku besar memiliki beberapa tugas utama, diantaranya adalah pengaturan akun perkiraan (*Chart Of Account*), pengaturan periode akuntansi, pencatatan jurnal dan mencetak laporan keuangan, baik itu laporan laba rugi atau laporan neraca keuangan [4].

2.4 Service-Oriented Architecture (SOA)

SOA atau arsitektur berorientasi layanan adalah suatu gaya arsitektur sistem yang membuat dan menggunakan proses bisnis dalam bentuk paket layanan sepanjang siklus hidupnya. SOA juga mendefinisikan dan menentukan arsitektur teknologi informasi yang dapat menunjang berbagai aplikasi untuk saling bertukar data dan berpartisipasi dalam proses bisnis. Fungsi-fungsi ini tidak terikat dengan sistem operasi dan bahasa pemrograman yang mendasari aplikasi-aplikasi tersebut [5].

SOA membagi fungsi-fungsi menjadi unit-unit yang berbeda (layanan), yang dapat didistribusikan melalui suatu jaringan dan dikombinasikan serta digunakan ulang untuk membentuk aplikasi bisnis. Layanan-layanan ini saling berkomunikasi dengan mempertukarkan data antar mereka atau dengan mengkoordinasikan aktivitas antara dua atau lebih layanan. Konsep SOA sering dianggap didasari atau berkembang dari konsep-konsep yang lebih lama dari komputasi terdistribusi dan pemrograman modular.

2.5 Windows Communication Foundation

WCF atau *Windows Communication Foundation* adalah salah satu teknologi yang memungkinkan aplikasi dalam lingkungan terdistribusi dan berkomunikasi satu sama lain. WCF adalah model pemrograman lengkap untuk membangun aplikasi berorientasi layanan. Teknologi ini memungkinkan pengembang untuk membangun solusi aman, handal, dan mendukung transaksi, yang dapat terintegrasi lintas platform serta mampu beroperasi dengan investasi yang ada [6].

2.6 Windows Workflow Foundation

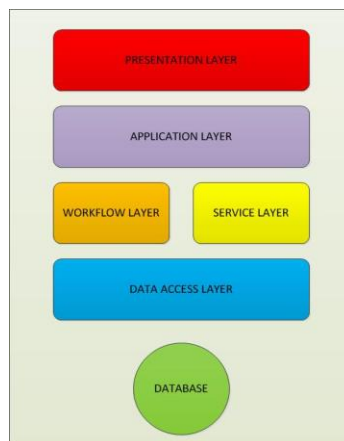
WF merupakan sebuah teknologi yang menyediakan sebuah API, *in-process workflow engine*, dan desainer untuk mengimplementasikan proses sebagai sebuah WF dalam aplikasi yang berbasis .NET. Sebuah WF merupakan serangkaian langkah-langkah atau fase dalam pemrograman. Setiap fase dimodelkan dalam WF sebagai sebuah Activity dan framework .NET menyediakan sebuah *library Activity* yang mencakup keseluruhan aktivitas aktivitas. Sekumpulan aktivitas dapat disusun secara visual dalam beberapa WF menggunakan Workflow Designer[7].

3. METODOLOGI PENELITIAN

Metode penelitian yang dipakai pada paper ini menggunakan pendekatan pembuatan perangkat lunak yaitu tahapan analisis dan desain.

3.1 Analisis Sistem

Domain permasalahan utama yang diangkat dalam paper ini adalah bagaimana mengimplementasikan suatu aplikasi berbasis web untuk menangani permasalahan-permasalahan yang pada umumnya terdapat di perusahaan manufaktur, khususnya pada bagian proses bisnis/ domain fungsional General Ledger (Buku Besar Umum) dan juga Cash Bank Management ?



Gambar 2. Arsitektur Perangkat Lunak

Berangkat dari hal tersebut, secara umum domain permasalahan utama ini dibagi-bagi menjadi beberapa subdomain sesuai dengan subproses bisnis yang berintegrasi dalam General Ledger maupun Cash Bank Management, antara lain adalah manajemen kas, manajemen bank, manajemen cek dan manajemen giro, kemudian untuk domain General Ledger antara lain adalah manajemen nomer akun, manajemen periode akuntansi, manajemen jurnal, buku besar dan mencetak laporan keuangan.

Subbagian-subbagian tersebut yang membentuk suatu kesatuan buku besar umum dan manajemen kas bank juga selanjutnya harus terhubung dengan domain fungsional lain dalam sistem ERP yang diimplementasikan, terutama dalam hal pengiriman dan pengambilan data yang akan digunakan dalam pengolahan informasi masing-masing domain fungsional. Contoh hubungan antara domain fungsional buku besar umum dan manajemen kas bank dengan beberapa domain fungsional lainnya dalam sistem ERP yang diterapkan dalam aplikasi ini adalah:

1. Domain Inventory untuk menulis ayat jurnal inventory
2. Domain Account Payable untuk menulis ayat jurnal suatu hutang belum dibayar perusahaan
3. Domain Account Receivable untuk menulis ayat jurnal suatu piutang perusahaan
4. Domain Manufacturing untuk menulis ayat jurnal work in process
5. Hubungan internal cash bank dengan General Ledger.

Perangkat lunak yang dibangun mengakomodir beberapa fungsi utama pada domain fungsional GL dan CB

dalam sebuah sistem ERP. Aplikasi yang dibangun memiliki arsitektur *service-oriented* dan menerapkan fitur *workflow* oleh *Windows Workflow Foundation* untuk menangani proses-proses transaksi dalam beberapa subdomainnya.

3.2 Desain Sistem

Desain sistem dibagi menjadi beberapa bagian yaitu desain skenario kasus penggunaan, arsitektur, servis dan workflow.

1) Desain Arsitektur

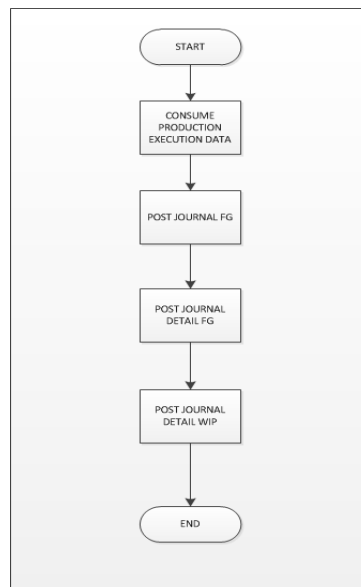
Arsitektur dari perangkat lunak ini mempunyai 5 layer, yaitu Data Acces Layer, Service Layer, Workflow Layer dan Presentation Layer. Desain arsitektur ditunjukkan pada Gambar 2.

2) Desain Servis

Web service yang dirancang berguna sebagai servis *atomic* dipakai oleh proses *compositing* oleh *workflow* maupun proses bisnis domain fungsi yang lain di dalam sistem ERP.

3) Desain Workflow

Workflow yang digunakan dalam perangkat lunak ini ada dua jenis, yaitu *workflow* yang berfungsi sebagai *composite* dari beberapa *service atomic* serta workflow yang merupakan navigasi halaman, sehingga terlihat jelas alur kerja halaman perangkat lunak. Salah satu desain workflow dalam perangkat ini dapat dilihat pada Gambar 3.



Gambar 3. Desain Workflow

4. HASIL PENGUJIAN

4.1. Pengujian Fungsionalitas

Dalam perangkat lunak yang dikembangkan, output fungsionalitas utama yang paling diharapkan untuk domain Cash Bank adalah keberhasilan perangkat lunak untuk mencatat semua aktifitas proses bisnis pengeluaran atau pemasukan kas, bank, cek dan giro. Dan untuk domain General Ledger adalah pengaturan akun perkiraan, periode akuntansi, pencatatan jurnal, buku besar dan laporan keuangan, yaitu laporan laba rugi dan laporan neraca keuangan yang selalu di terbitkan pada akhir periode akuntansi perusahaan. Pada laporan laba rugi, terdapat banyak informasi yang ditampilkan, mulai dari berapa total Sales, Sales return, Cost Of Good Sold, Expense dan Interest dalam periode tertentu, yang nantinya akan menghasilkan informasi net income dari perusahaan.

Pada laporan neraca, juga terdapat informasi yang ditampilkan. Mulai dari aset perusahaan yang terdiri dari Cash and Cash Equivalent, Trade Receivable, Inventory, Property, dan Goodwill. Liabilitas perusahaan yang terdiri dari Trade Payable, Short-Term Borrowing, Long-Term Borrowing, dan Ekuitas yang terdiri atas Shareholder Equity, Shared Capital dan juga Retained Earning. Setelah dilakukan pengujian untuk tiga tenant (dua perusahaan manufaktur sepeda dan satu perusahaan manufaktur sereal), semua fungsionalitas baik itu domain Cash Bank dan General Ledger sudah berjalan dengan baik dan memenuhi semua keluaran yang diharapkan.

4.1.1. Pengujian Menambah Penerimaan Kas Dari Invoice

Pengujian dilakukan terhadap subbagian pembuatan data penerimaan kas dari transaksi yang berasal dari *Invoice*, pengujian ini dimulai ketika pengguna memilih menu *Cash ReceiptInvoice* dan berakhir dengan menampilkan daftar penerimaan kas. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 1, sedangkan hasilnya ditunjukkan pada Gambar 4 dan Gambar 5.

Tabel 1 Skenario Pengujian Menambah Penerimaan Kas Dari *Invoice*

| | |
|-----------------------|--|
| Kode | PF-001 |
| Tujuan Pengujian | Menguji fungsi menambah data penerimaan kas dari transaksi <i>Invoice</i> |
| Kondisi Awal | Pengguna sudah <i>login</i> dan masuk ke dalam halaman daftar penerimaan kas |
| Data Masukan | <ol style="list-style-type: none"> Memilih <i>Invoice Number</i> “ARSI20130528001” Kolom <i>Transaction Code</i> diisi dengan “AR” Kolom <i>Amount</i> diisi dengan “39600” Kolom <i>Notes</i> diisi dengan “Pembayaran Kayu” Kolom <i>Unique Code</i> diisi dengan “AR.001” Kolom <i>Date</i> diisi dengan “10/02/2013” |
| Prosedur Pengujian | <ol style="list-style-type: none"> Menekan tombol <i>Create</i> Mengisi kolom sesuai data masukan Menekan tombol <i>Create</i> |
| Hasil yang Diharapkan | Data masukan untuk penerimaan kas dari <i>Invoice</i> yang baru dapat masuk ke dalam basis data |
| Hasil yang Diperoleh | Data penerimaan kas dari <i>Invoice</i> yang baru berhasil masuk ke dalam basis data |
| Kesimpulan | Proses menambah data penerimaan kas dari <i>Invoice</i> berhasil |
| Kondisi Akhir | Pengguna berada pada halaman daftar penerimaan kas |

Gambar 4 Menambah Data Penerimaan Kas

| Cash Receipt Code | Reference No. | Amount | Notes | Transaction Date | Action |
|-------------------|-----------------|------------|-----------------|------------------|--|
| AR.001 | ARSI20130528001 | 39600.0000 | Pembayaran Kayu | 02 Oct 2013 | detail edit delete |
| | | | | | |
| | | | | | |
| | | | | | |

Gambar 5 Hasil Pengujian Penambahan Data Penerimaan Kas

4.1.2. Pengujian Menambah Penerimaan Kas Dari Non-Invoice

Pengujian ini dilakukan terhadap subbagian pembuatan data penerimaan kas dari transaksi yang berasal dari *non-Invoice*, pengujian ini dimulai ketika pengguna memilih menu *Cash ReceiptNon-Invoice* dan berakhir dengan menampilkan daftar penerimaan kas. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 2, sedangkan hasilnya ditunjukkan pada Gambar 6 dan Gambar 7

Tabel 2 Skenario Pengujian Menambah Penerimaan Kas Dari *Non-Invoice*

| | |
|-----------------------|--|
| Kode | PF-002 |
| Tujuan Pengujian | Menguji fungsi menambah data penerimaan kas dari transaksi <i>non-Invoice</i> |
| Kondisi Awal | Pengguna sudah <i>login</i> dan masuk ke dalam halaman daftar penerimaan kas |
| Data Masukan | <ol style="list-style-type: none"> 1. Kolom <i>Receipt Source</i> diisi dengan “Hutang Andre” 2. Kolom <i>Transaction Code</i> diisi dengan “AR” 3. Kolom <i>Amount</i> diisi dengan “900000” 4. Kolom <i>Notes</i> diisi dengan “Pembayaran Hutang Andre” 5. Kolom <i>Unique Code</i> diisi dengan “AR.002.01” 6. Kolom <i>Date</i> diisi dengan “10/01/2013” |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Menekan tombol <i>Create</i> 2. Mengisi kolom sesuai data masukan 3. Menekan tombol <i>Create</i> |
| Hasil yang Diharapkan | Data masukan untuk penerimaan kas dari <i>non-Invoice</i> yang baru dapat masuk ke dalam basis data |
| Hasil yang Diperoleh | Data penerimaan kas dari <i>non-Invoice</i> yang baru berhasil masuk ke dalam basis data |
| Kesimpulan | Proses menambah data penerimaan kas dari <i>non-Invoice</i> berhasil |
| Kondisi Akhir | Pengguna berada pada halaman daftar penerimaan kas |

Gambar 6 Menambah Data Penerimaan Kas Non-Invoice

| Cash Receipt Code | Reference No. | Amount | Notes | Transaction Date | Action |
|-------------------|-----------------|-------------|-------------------------|------------------|--|
| AR.002.01 | Hutang Andre | 900000.0000 | Pembayaran Hutang Andre | 01 Oct 2013 | detail edit delete |
| AR.001 | ARS120130528001 | 39600.0000 | Pembayaran Kayu | 02 Oct 2013 | detail edit delete |

Gambar 7 Hasil Pengujian Penambahan Data Penerimaan Kas Non-Invoice

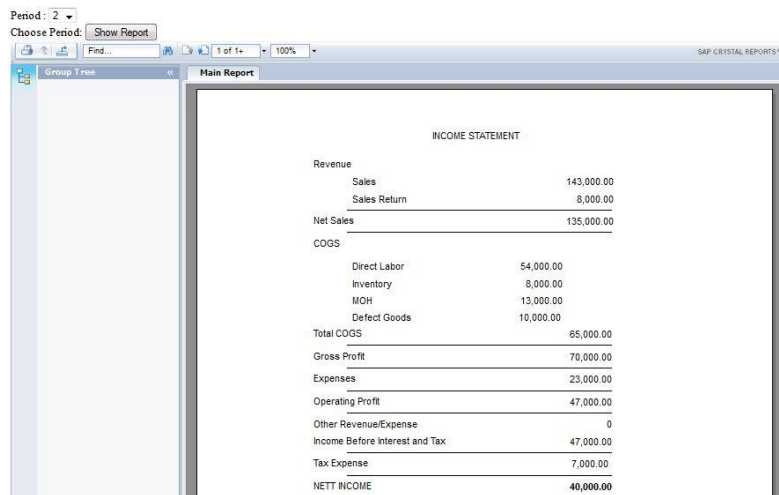
4.1.3. Pengujian Mencetak Laporan Laba Rugi

Pengujian ini dilakukan terhadap fungsionalitas mencetak laporan laba rugi. Pengujian ini dimulai ketika pengguna mengakses halaman Report dan berakhir pada halaman tersebut. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 3 dan Gambar 9.

Tabel 3 Skenario Pengujian Mencetak Laporan Laba Rugi

| | |
|------------------|--|
| Kode | PF-018 |
| Tujuan Pengujian | Menguji fungsi mencetak laporan laba rugi |
| Kondisi Awal | Pengguna sudah <i>login</i> dan masuk ke dalam halaman laporan laba rugi |
| Data Masukan | 1. Kolom <i>Period</i> diisi dengan ” 2” |

| | |
|-----------------------|--|
| Prosedur Pengujian | 1. Menekan tombol <i>Show Report</i> |
| Hasil yang Diharapkan | Laporan laba rugi dapat ditampilkan |
| Hasil yang Diperoleh | Laporan laba rugi berhasil ditampilkan |
| Kesimpulan | Proses mencetak laporan laba rugi berhasil |
| Kondisi Akhir | Pengguna berada pada halaman laporan laba rugi |



Period: 2
Choose Period: Show Report

Group Tree

Main Report

INCOME STATEMENT

| | |
|--------------------------------|------------|
| Revenue | |
| Sales | 143,000.00 |
| Sales Return | 8,000.00 |
| Net Sales | 135,000.00 |
| COGS | |
| Direct Labor | 54,000.00 |
| Inventory | 8,000.00 |
| MOH | 13,000.00 |
| Defect Goods | 10,000.00 |
| Total COGS | 85,000.00 |
| Gross Profit | 70,000.00 |
| Expenses | 23,000.00 |
| Operating Profit | 47,000.00 |
| Other Revenue/Expense | 0 |
| Income Before Interest and Tax | 47,000.00 |
| Tax Expense | 7,000.00 |
| NETT INCOME | 40,000.00 |

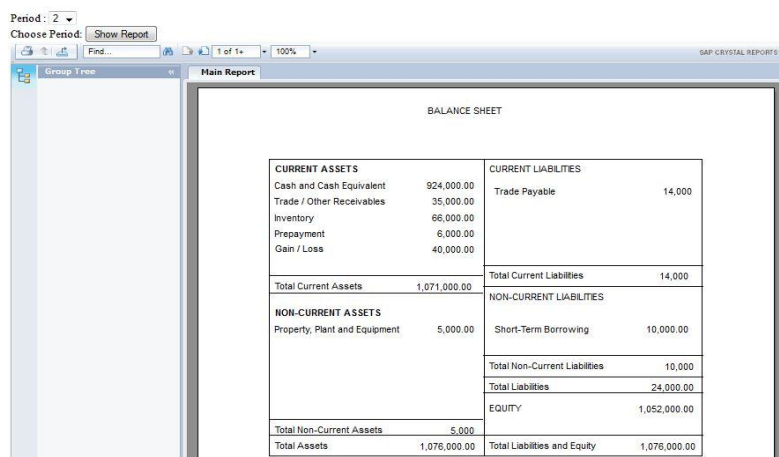
Gambar 8 Hasil Pengujian Laporan Laba Rugi

4.1.4. Pengujian Mencetak Laporan Neraca Keuangan

Pengujian ini dilakukan terhadap fungsionalitas mencetak laporan neraca keuangan. Pengujian ini dimulai ketika pengguna mengakses halaman *Report* dan berakhir pada halaman tersebut. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 4 dan Gambar 9

Tabel 4 Skenario Pengujian Mencetak Laporan Neraca Keuangan

| | |
|-----------------------|--|
| Kode | PF-019 |
| Tujuan Pengujian | Menguji fungsi mencetak laporan neraca keuangan |
| Kondisi Awal | Pengguna sudah <i>login</i> dan masuk ke dalam halaman laporan neraca keuangan |
| Data Masukan | 1. Kolom <i>Period</i> diisi dengan "2" |
| Prosedur Pengujian | 1. Menekan tombol <i>Show Report</i> |
| Hasil yang Diharapkan | Laporan neraca keuangan dapat ditampilkan |
| Hasil yang Diperoleh | Laporan neraca keuangan berhasil ditampilkan |
| Kesimpulan | Proses mencetak laporan neraca keuangan berhasil |
| Kondisi Akhir | Pengguna berada pada halaman laporan neraca keuangan |



Period: 2
Choose Period: Show Report

Group Tree

Main Report

BALANCE SHEET

| | | | |
|-------------------------------|--------------|-------------------------------|--------------|
| CURRENT ASSETS | | CURRENT LIABILITIES | |
| Cash and Cash Equivalent | 924,000.00 | Trade Payable | 14,000 |
| Trade / Other Receivables | 35,000.00 | | |
| Inventory | 66,000.00 | | |
| Prepayment | 6,000.00 | | |
| Gain / Loss | 40,000.00 | | |
| Total Current Assets | 1,071,000.00 | Total Current Liabilities | 14,000 |
| NON-CURRENT ASSETS | | NON-CURRENT LIABILITIES | |
| Property, Plant and Equipment | 5,000.00 | Short-Term Borrowing | 10,000.00 |
| | | | |
| | | Total Non-Current Liabilities | 10,000 |
| | | Total Liabilities | 24,000.00 |
| | | EQUITY | 1,052,000.00 |
| Total Non-Current Assets | 5,000 | | |
| Total Assets | 1,076,000.00 | Total Liabilities and Equity | 1,076,000.00 |

Gambar 9 Hasil Pengujian Laporan Neraca Keuangan

4.2. Pengujian Workflow

Implementasi workflow dalam perangkat lunak terdapat dalam proses bisnis pencatatan jurnal, workflow yang dibuat berguna sebagai composite dari beberapa atomic web service, yaitu Workflow Post Journal yang terdiri dari atomic service PostJournalDebit dan PostJournalCredit yang mempunyai parameter chart of account atau

akun perkiraan dalam setiap pemanggilannya. Setelah dilakukan pengujian untuk tiga tenant (dua perusahaan manufaktur sepeda dan satu perusahaan manufaktur sereal), workflow yang dibuat sudah bisa mencatat jurnal secara otomatis sesuai dengan parameter yang diinginkan

5. SIMPULAN

Dari proses pengerjaan selama perancangan, implementasi, dan proses pengujian aplikasi yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Perangkat lunak yang dirancang dan dibangun dalam paper ini dapat mengakomodasi beberapa proses-proses bisnis untuk domain fungsional Cash Bank yaitu manajemen kas, bank, cek dan giro dan manajemen rekening. Domain fungsional General Ledger yaitu manajemen akun perkiraan, manajemen periode akuntansi, mencatat ayat jurnal, melihat buku besar dan mencetak laporan laba-rugi dan laporan neraca keuangan.
2. Perangkat lunak yang dirancang dan dibangun dalam paper ini telah mengimplementasikan bentuk arsitektur berbasis (Service Oriented Architecture), yaitu pada tingkat service layer aplikasi dengan menggunakan teknologi WCF Service.
3. Perangkat lunak yang dirancang dan dibangun dalam paper ini telah dibangun dengan menggunakan pendekatan Workflow. Pendekatan Workflow ini diimplementasikan dengan menggunakan teknologi Windows Workflow Foundation pada subproses-subproses bisnis yang mungkin berubah alur prosesnya, seperti subproses mencatat jurnal otomatis dan juga mengatur navigasi dari halaman-halaman web dari aplikasi.

6. DAFTAR RUJUKAN

- [1] S.R. Magal and J. Word, *Integrated Business Processes with ERP System*, 1st ed. Wiley, 2011.
- [2] G. Ventyana, *Rancang Bangun Aplikasi Cash Bank*. Surabaya, 2011.
- [3] C.S. Warren, J. M. Reeve, and J. Duchac, *Principle Of Accounting*, 22nd ed. United States: Cengage Learning, 2006.
- [4] I. Gumilar, *Rancang Bangun Aplikasi General Ledger*. Surabaya, 2011.
- [5] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall, 2005.
- [6] Microsoft. (2013, Jul.) MSDN Microsoft. [Online]. <http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx>.
- [7] M. Collins, *Beginning WF*, 1st ed, Apress, 2010.