Rekayasa Perangkat Lunak Komersial Akuntansi Dengan Analisis Rasio Untuk Usaha Jasa

Yudi Wahyudi
Universitas Surakarta

ABSTRACT: Although small in scale, field services business in Indonesia is starting required to hold the actual accounting systems, standards, able to analyze the ratio and easily implemented. Meanwhile, not all stakeholders of small businesses that exist have the ability to manage its accounting system independently. Moreover, paying an accountant.
Keywords: Computer Accounting

Abstraksi : Walaupun dalam skala kecil, bidang usaha jasa di Indonesia saat ini sudah mulai dituntut untuk menyelenggarakan sistem akuntansi yang aktual, standar, mampu melakukan analisa rasio dan mudah diimplemetasikan. Sementara itu, tidak semua *stakeholder* usaha-usaha kecil yang ada memiliki kemampuan untuk mengelola sistem akuntansinya secara mandiri. Apalagi membayar seorang akuntan.
Kata Kunci : Komputer Akuntansi

1. LATAR BELAKANG

Disebut aktual karena data yang ada dan tersaji harus selalu dapat diperbarui dengan cepat agar sesuai dengan kondisi nyata. Disebut standar karena tata-cara, mekanisme dan sajian data keluaran harus sesuai standar nasional yang berlaku (Sistem Akuntansi Indonesia atau SAI), disebut mampu melakukan analisa rasio karena sistem tersebut secara akurat harus mampu melakukan analisis dengan hasil yang akurat dalam waktu singkat. Disebut mudah diimplementasikan karena penerapan sistem yang ada harus dapat diselenggarakan oleh *stakeholder* sendiri atau sumberdaya manusia pengganti yang tidak harus seorang akuntan.

Di lain pihak, dalam kasanah teknologi informasi telah berkembang metode atau teknik informatika yang semakin canggih dan berbasis komputer sehingga masalah tersebut sebenarnya dapat diatasi dengan menyediakan perangkat lunak khusus untuk mendukung sistem akuntansi yang telah ada.

Masalahnya, dewasa ini di pasar komersial hanya tersedia perangkat lunak komputer untuk sistem akuntansi dari luar negeri yang mahal dan tidak selalu sesuai dengan SAI. Sehingga penelitian dan perekayasaannya, layak diupayakan dan dijadikan topik dari penelitian ini.

2. RUMUSAN MASALAH

Berdasarkan masalah yang ada dapat diperoleh rumusan masalah sebagai berikut :

1. Di pasar komersial hanya tersedia perangkat lunak komputer untuk sistem akuntansi yang sebagian besar berasal dari luar negeri yang mahal dan tidak selalu sesuai dengan SAI
2. Bagaimana melakukan rekayasa perangkat lunak komersial akuntansi dengan analisis rasio untuk usaha jasa

3. BATASAN MASALAH

Sebagaimana layaknya sebuah karya ilmiah lainnya, maka untuk menghindari bias atau distorsi atas penelitian ini maka berikut ini adalah batasan masalah :

1. Area penelitian untuk sistem akuntansi dibatasi hanya pada SAI
2. Rekayasa sistem (*system engineering*) hanya dibatasi pada analisis dan perancangan sistem untuk menghasilkan spesifikasi kebutuhan perangkat lunak (*software requirement system*).

4. TUJUAN

1. Membuktikan bahwa rekayasa perangkat lunak komersial akuntansi dengan analisa rasio usaha jasa layak untuk dilaksanakan
2. Menyediakan perangkat lunak komersial akuntansi dengan analisis rasio usaha jasa yang ringan dan mudah digunakan oleh siapa saja yang membutuhkan

5. Manfaat Penelitian

1. Bagi para pengguna sistem komputer, hasil penelitian ini dapat digunakan atau diterapkan sebagai sub sistem pendukung dari sistem akuntansi yang sudah ada
2. Bagi Akademik, hasil penelitian ini dapat digunakan sebagai sumbangan pemikiran ilmiah bagi civitas akademik

6. RANCANGAN STRUKTUR PHISIK DATA (*PHYSICAL DATA STRUCTURE DESIGN*)

Desain atau rancangan struktur phisik data sangat terpengaruh oleh model dari hasil rancangan konseptual dan logikal. Pada struktur phisik data, spesifikasi yang akan mengidentifikasi setiap field sudah disesuaikan dengan kondisi yang diminta oleh mesin basisdata (*database server*). Karena 100% (seratus prosen) manipulasi basisdata pada perangkat lunak ini dirancang untuk menggunakan *structured query language* (standard SQL-192) maka untuk menjamin kesuksesan setiap komando *query* tunggal maupun yang berupa *query* terstruktur seluruh tabel (*database file*) yang terlibat di dalam perangkat lunak ini bertipe *dynaset*. Seluruh relasi antar tabel tidak ditentukan oleh *indexing method* tetapi oleh *query* secara langsung. Rancangan struktur phisik yang demikian diharapkan dapat menjamin pula dinamika struktur dari database yang bersangkutan.

Tabel Struktur Data Phisik (*Physical Data Structure*)

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	DirKerja	Text	250	No
1	DirDB	Text	250	No
2	Berkas	Text	40	No
3	IniPerk	Boolean	1	No
4	IniKode	Boolean	1	No
5	IniPenggun a	Boolean	1	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D1.2:Config.Perusahaan

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Perusahaan	Text	50	Yes
1	Pimpinan	Text	40	Yes
2	Sertifikat	Text	50	Yes
3	Inisiasi	Boolean	50	Yes
4	Aktivasi	Boolean	50	Yes

Keterangan :
D1 untuk kepentingan teknis pada Inisiasi Awal bertambah dengan field-field legalitas penggunaan program tetapi dipisahkan dalam sebuah database file tersendiri berspesifikasi Config.Perusahaan..

Tabel Struktur Data Phisik (*Physical Data Structure*)

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Nomor	Text	10	No
1	Perkiraan	Text	50	No
2	Golongan	Text	1	No
3	Keterangan	Boolean	1	No

Keterangan : xxx adalah nama database untuk satu siklus akuntansi

Tabel Struktur Data Phisik (*Physical Data Structure*)
D3:xxx.Kode

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Kode	Text	10	No
1	Debet	Text	10	No
2	Kredit	Text	10	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D4:xxx.Pengguna

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Ekaun	Text	20	No
1	Nama	Text	50	No
2	Sandi	Text	10	No
3	Hak	Text	1	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D5:xxx.Transaksi

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	IDTransaksi	Text	20	No
1	Tanggal	Date/Time	8	No
2	Bukti	Text	30	No
3	Pelaku	Text	20	No
4	Oponen	Text	50	No
5	Sudah	Boolean	1	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D6:xxx.Jurnal

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	IDTransaksi	Text	20	No
1	Tanggal	Date/Time	8	No
2	Kode	Text	10	No
3	Nomor	Text	10	No
4	DK	Text	1	No
5	Nominal	Currency	8	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D7.1:xxx.RL_jdl_xxx

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Judul	Text	100	No

Tabel Struktur Data Phisik (*Physical Data Structure*)
D7.2:xxx.RL_xxx

Ordinal Position	Field Name	Field Type	Field Size	Allow Zero Length
0	Nomor	Text	20	No
1	Perkiraan	Text	70	No
2	Ket	Text	1	No
3	Saldo	Currency	8	No
4	Jumlah	Currency	8	No

7. PEMBUATAN PROTOTIPE (*PROTOTYPING*)

Karena rekayasa perangkat lunak ini merupakan rekayasa perangkat lunak komersial maka tidak dibutuhkan prototipe seperti halnya pada perekayasaan perangkat lunak yang *taylor made*. Hanya saja, pada saat *teamwork* yang dibentuk khusus untuk penelitian ini melakukan *meeting* untuk berdiskusi dalam menterjemahkan kehendak rancangan perangkat lunak yang telah terbentuk sebelumnya, didapat beberapa penggalan-penggalan prototipe yang berasal dari usulan para anggota *teamwork*. Berikut ini adalah penjelasan perihal prototipe-prototipe yang berhasil dikoleksi dari diskusi-diskusi tersebut.

8. Awal Eksekusi

Pada saat aplikasi dieksekusi, aplikasi dengan segera akan menampilkan sebuah jendela lepas yang oleh industri perangkat lunak biasa disebut dengan *Splash*. Tetapi sebelum hal tersebut dilakukan, aplikasi ini akan mengaktifkan terlebih dahulu 3 buah modul yang terdiri dari sebuah modul yang khusus mengurusi *flagging* pengguna yang sedang *login* (ModPengguna.bas), modul khusus untuk menangani interaksi pengguna dengan tombol *keyboard* dan *mouse* (ModMouseKey.bas) dan sebuah modul yang memuat variabel-variabel yang bersifat *global*. Berikut adalah tampilan dari *splash* aplikasi ini.

Gambar *Splash*

9. Inisiasi Awal

Program/aplikasi ini memiliki sebuah basisdata khusus (aplikasi lain biasanya menggunakan berkas *plain text* yang diberi ekstensi **.ini*) yang diberi tajuk sebagai *config.mdb* (Lihat Tabel 3.2 dan Tabel 3.3 untuk melihat struktur phisik dari basisdata ini). Basisdata ini keberadaannya dapat digunakan sebagai indikator yang menunjukkan apakah aplikasi ini sudah berjalan ataukah aplikasi ini baru saja ditanam (di-*install*) dan belum aktif. Jika *config.mdb* tidak diketemukan, aplikasi akan secara otomatis mengeksekusi bagian aplikasi yang khusus dipergunakan untuk melakukan inisiasi awal.

Proses inisiasi awal dimulai dengan meminta pengguna (seharus *Administrator*) untuk memasukkan hal legal yang berhubungan dengan hak pemakaian aplikasi.

Gambar *Inisiasi Awal Sertifikasi Aktivasi Aplikasi*

Dari Gambar 4.2. terlihat bahwa Administrator harus memasukkan Nama Perusahaan yang akan menggunakan aplikasi ini, Nama Pimpinan dari perusahaan yang bersangkutan dan nomor sertifikat pembelian. Nomor sertifikat yang menunjukkan keabsahan pembelian ini dapat diketemukan di bagian atas dari sertifikat pembelian yang pada saat aplikasi ini dipasarkan seharusnya disertakan di dalam kemasan.

Jika nomor sertifikat yang dimasukkan oleh Administrator benar maka aplikasi akan mengakses *serial number* dari *processor* dan *primary HDD,* melakukan *encoding* dengan algoritma tertentu dan kemudian hasilnya ditampilkan di *textbox* “Kode Inisiasi”. Aplikasi juga melakukan proses tertentu yang hasilnya adalah “Kode Aktivasi”. Kode Aktivasi ini “disimpan” oleh aplikasi. Tidak diperlihatkan kepada pengguna.

Kode Inisiasi kemudian oleh Administrator harus di-*SMS*-kan ke pihak *vendor* yang memiliki *server* dengan aplikasi pelayanan yang dapat melakukan proses untuk membuat Kode Aktivasi

yang sama dengan yang ada di sistem Pengguna (termasuk memeriksa apakah nomor telepon Administrator yang bersangkutan sudah terdaftar sebagi pengguna legal ataukah tidak).

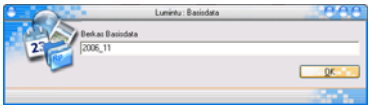
Setelah *server* dari *vendor* secara otomastis melakukan *replay* dan diterima oleh Administrator, maka Administrator harus memasukkan Kode Aktivasi tersebut di tempat yang telah tersedia. Apabila Kode Aktivasi dari *vendor* yang dimasukkan cocok dengan Kode Aktivasi yang ada di sistem pengguna dan informasi *EULA* telah disetujui oleh pengguna maka jika kemudian pengguna menekan tombol *Terus*>>>> maka proses inisiasi awal akan terus berlanjut.

Gambar *Inisiasi Awal Inisiasi Basisdata*

Pada saat proses inisiasi awal sudah ke inisiasi basisdata, Administrator masih diberi kesempatan untuk melakukan koreksi atas inisiasi sertifikasi yang telah dilakukannya atau bahkan dapat membatalkan proses inisiasi awal ini. Jika Administrator memutuskan untuk terus melakukan proses inisiasi awal, maka Administrator harus "memberitahu" terlebih dahulu *path* dan nama basisdata yang akan digunakannya (basisdata aktif) kepada aplikasi. Jika Administrator tidak hafal dan/atau takut salah menuliskan *path* dan nama basisdata yang dikehendakinya (apalagi kalau basisdatanya tidak tersedia) maka Administrator dapat menekan tombol komando "Tunjuk" yang berada tepat di sebelah kanan *textbox* yang sedianya digunakan untuk menuliskan *path* dan nama basisdata. Jika demikian maka di layar monitor akan tertampil *independent window* khusus untuk mengatur atau melakukan manajemen atas basisdata milik pengguna. Berikut adalah *snapshot* daripadanya.

Gambar *Jendela Direktori Basisdata*

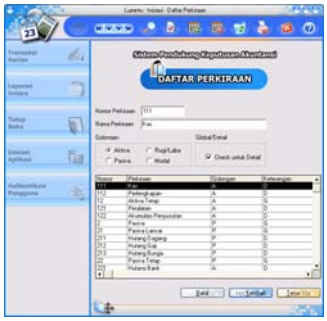
Administrator (pengguna) dapat menentukan *path* dan nama basisdata dengan menggunakan "explorer" khusus basisdata aplikasi ini. Tentu saja karena aktivitas ini berlaku pada saat inisiasi awal (yang berarti belum tersedia basisdata-nya) maka Administrator harus membuat basisdata baru. Caranya dengan melakukan *right-click* di area berkas basisdata. Berikut ini adalah tampilan *dialogbox* untuk memasukkan nama berkas basisdata baru.



Gambar *Dialogbox Memasukkan Nama Berkas Basisdata Baru*

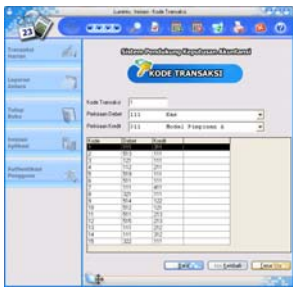
Selain membuat basisdata baru, explorer juga bisa melayani pengguna untuk masalah sederhana seperti menghapus basisdata yang sudah tidak diperlukan lagi dan mengaktifkan basisdata yang sudah ada. Jika pengguna menghendaki pelayanan manajemen berkas yang kompleks, lebih baik menggunakan Microsoft™ Windows® Explorer yang sudah tersedia di sistem operasi.

Inisiasi awal selanjutnya adalah memasukkan Daftar Perkiraan (*Chart Of Account*) yang akan digunakan dalam proses *classifying*. Lihat Gambar 4.6. Jadi Administrator sebaiknya telah mempersiapkan Daftar Perkiraan yang berlaku atau akan diberlakukan di perusahaan yang berangkutan. Walaupun demikian, asalkan transaksi untuk pos tertentu belum dilakukan dan dengan tetap berpedoman pada prinsip-prinsip dasar akuntansi, Administrator dapat meng-*update* Daftar Perkiraan pada saat siklus sudah berjalan. Berikut adalah subprogram yang digunakan untuk pemeliharaan data Daftar Perkiraan.



Gambar *Inisiasi Awal Daftar Perkiraan*

Subprogram untuk pemeliharaan data ini (dan juga sub-subprogram yang sejenis pada aplikasi ini) memiliki keunikan pada satuan-satuan kendalinya (*controls*). Tidak sebagaimana aplikasi lain yang ditulis dengan menggunakan Microsoft™ Visual Basic®, selain menggunakan *toolbar* non-*intrinsic* subprogram ini juga menggunakan *keystroke* sebagai perangkat kendali alternatif.Setiap transaksi pasti mengandung satu atau lebih item-item yang menunjukkan setiap obyek transaksi pada transaksi yang bersangkutan. Kemudian, untuk setiap transaksi harus di-*post*-kan ke dalam buku besar sehingga agar proses tersebut dapat bekerja secara otomatis maka aplikasi ini harus memiliki pedoman untuk pekerjaan tersebut. Setelah memasukkan Daftar Perkiraan Administrator harus pula menyiapkan jenis-jenis transaksi (diwakili oleh sebuah Kode Transaksi) yang diprediksikan akan terjadi dan "alamat" *post*-nya sesuai dengan Daftar Perkiraan. Berikut adalah *snapshot* dari subprogram untuk memasukkan Kode Transaksi ke dalam aplikasi.



Gambar *Inisiasi Awal Kode Transaksi*

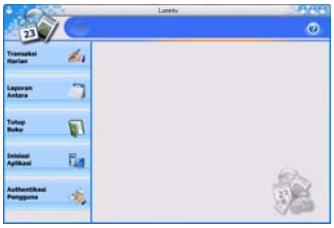
Aplikasi yang baru saja di-*install* sebenarnya tidak atau belum memiliki seorang penggunapun. Jadi secara otomatis yang melakukan proses instalasi dianggap sebagai calon Administrator dan harus memasukkan Ekaun, Nama dan Sandi yang akan digunakannya.

Gambar *Inisiasi Awal Authentikasi*

Setelah Administrator selesai dengan authentikasi untuk dirinya maka selesai juga proses inisiasi awal ini. *Workarea* akan dikosongkan dan aplikasi siap menerima komando-komando selanjutnya dari pengguna.

10. Sajian Dan Area Kerja

Pada saat inisiasi awal, sebenarnya sajian dan area kerja dari aplikasi ini sudah terlihat. Di bagian kiri jendela (bertipe *multiple document interface*) disediakan khusus untuk meletakkan sajian. Sajian tarik bawah (*pulldown menu*) yang biasanya menjadi tipe sajian standard di aplikasi komersial tidak disertakan. Alasan *teamwork* pengembang bersetuju untuk menghilangkan pelayanan ini karena aplikasi ini bukan aplikasi utilitas yang kompleks seperti aplikasi-aplikasi komersial yang lain tetapi justru cenderung monolitik.



Gambar *MDI Dengan Sajian dan Workarea*

Kinerja dari sajian ini sudah dijelaskan di Bab 3 tentang Rancangan Antarmuka Pengguna (*User Interface Design*) dan lebih rinci lagi juga dijelaskan di Bab 3 tentang Pembuatan Prototipe (*Prototyping*). Pada saat tidak digunakan untuk menampilkan sebuah subrogram, *workarea* diisi dengan sebuah subprogram yang disebut *Latar.frm* sehingga seolah *workarea* tersebut memiliki *background image* yang posisinya dapat menyesuaikan diri dengan keadaan.

11. Login Pengguna

Sajian tidak akan memberi tanggapan kepada pengguna sebelum pengguna yang akan

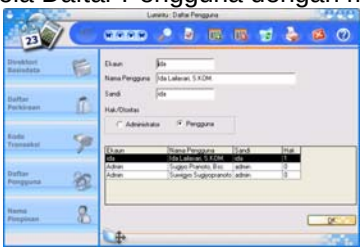
mempergunakan aplikasi ini *sign-in* atau *login*. Aplikasi ini hanya memiliki 2 tingkat *privileges*. Yaitu *privilege* untuk Administrator dan untuk pengguna lain seperti misalnya pembuka dari perusahaan pengguna.

Gambar *Snapshot Login*

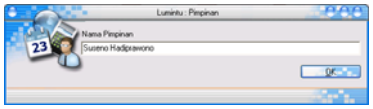
12. Inisiasi

Inisiasi dilakukan jika terjadi perubahan-perubahan yang harus dilakukan selama aplikasi ini dalam keadaan aktif dipergunakan. Dilakukan untuk *updating* beberapa *reference* penting seperti misalnya Penetapan basisdata aktif (Direktori Basisdata), menambah atau mengurangi *record* Daftar Perkiraan, Kode Transaksi, Daftar Pengguna dan Nama Pimpinan atau Pejabat lain yang akan menandatangani laporan keuangan.

Dengan pelayananan subprogram Direktori Basisdata ini pengguna dapat membuka arsip yang mana saja dan dapat membuat laporan-laporan ulang berupa salinan lunak dan keras jika diperlukan. Dan jika diperlukan, Administrator dapat mengelola Daftar Pengguna dengan mudah.



Gambar *Snapshot Inisiasi Daftar Pengguna*

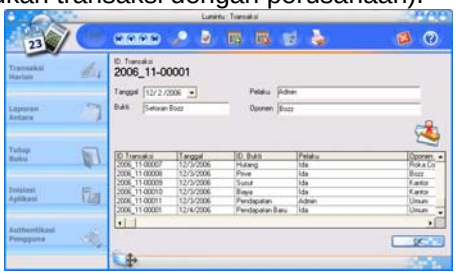


Gambar *Snapshot Inisiasi Pimpinan*

13. Identifikasi Transaksi

Transaksi dimulai dengan memilih sajian Transaksi Harian. Dengan memilih sajian ini maka subprogram yang memberi pelayanan untuk masalah ini akan aktif.

Pengguna pertamakali harus memasukkan identitas dari transaksi yang bersangkutan, yaitu : ID. Transaksi (di-*generate* oleh aplikasi), Tanggal Transaksi (*default*-nya tanggal sistem saat yang bersangkutan, Pelaku (otomatis, pengguna yang sedang *log*), dan Oponen (pihak lain yang melakukan transaksi dengan perusahaan).

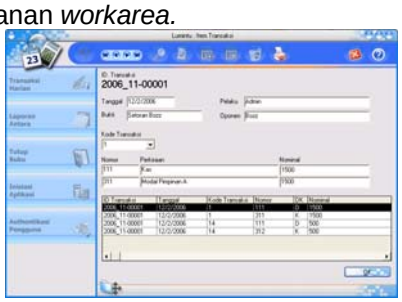


Gambar *Snapshot Transaksi*

ID. Transaksi adalah kode yang bersifat *composite*. Terdiri dari nama periode akuntansi yang bersangkutan (sama dengan nama basisdata aktif) dan setelah diberi karakter pemisah berupa tanda *hyphen,* ditambah dengan 5 digit bilangan pencacah. Bilangan pencacah ini akan dimulai dari 00001 untuk setiap harinya.

14. Memasukkan ItemTransaksi

Rincian transaksi yang terdiri dari satu atau lebih item transaksi kemudian dimasukkan setelah sebelumnya meng-*click* aiken yang berada di bagian kanan *workarea.*



Gambar *Snapshot Item Transaksi*

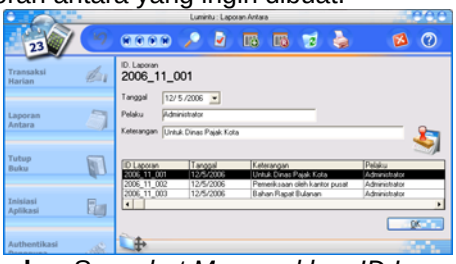
Dari proses memasukkan item transaksi ini akan terlihat betapa bergunanya *data reference* yang berupa Kode Transaksi. Dengan adanya kode ini aplikasi akan dapat secara langsung melakukan *posting*. Setiap kali kita memasukkan data item transaksi maka secara otomatis akan terbentuk 2 buah *record.*

15. Laporan Antara

Pada kebanyakan aplikasi utilitas akuntansi, hampir semua aplikasi tersebut tidak memiliki layanan untuk membuat laporan keuangan (apalagi analisis rasio) tanpa harus menutup buku terlebih dahulu. Aplikasi ini justru memiliki layanan tersebut. Dengan layanan ini pengguna dapat menerbitkan laporan antara tanpa dibatasi dengan waktu. Laporan antara seperti ini sangat penting untuk

mendukung proses pengambilan keputusan disebuah lembaga bisnis.

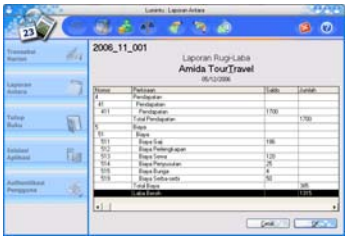
Pengguna harus memasukkan ID dari laporan antara yang ingin dibuat.



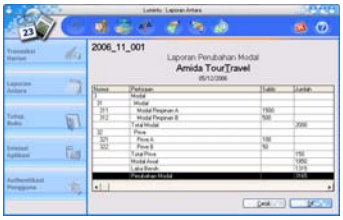
Gambar *Snapshot Memasukkan ID Laporan Antara*

16. Pratayang Laporan Antara

Setelah meng-*click* aiken yang ada di bagian kanan *workarea*, pengguna akan memperoleh *preview* dari laporan antara saat itu. Laporan-laporan di-*generate* oleh subprogram laporan antara ini terdiri dari laporan-laporan keuangan dan laporan-laporan hasil analisis rasio.

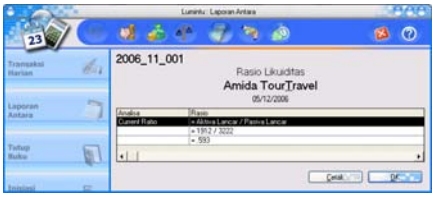


Gambar *Snapshot Laporan Antara Rugi-Laba*

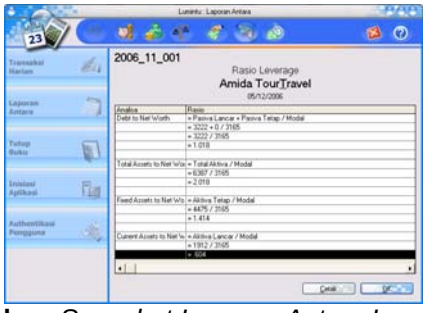


Gambar *Snapshot Laporan Antara Perubahan Modal*

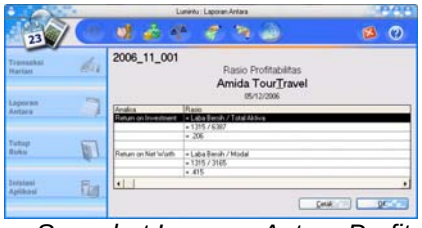
Gambar *Snapshot Laporan Antara Neraca*



Gambar *Snapshot Laporan Antara Analisis Rasio Likuiditas*



Gambar *Snapshot Laporan Antara Leverage*



Gambar *Snapshot Laporan Antara Profitabilitas*

Setiap laporan dapat dicetak tanpa bantuan aplikasi *report generator* seperti *halnya Crystal Report*® di kertas *pre-printed* dengan logo perusahaan yang bersangkutan.

17. Laporan Akhir

Pada saat dipilih sajian laporan Akhir, pengguna harus memberi konfirmasi bahwa pengguna yang bersangkutan yakin dengan apa yang akan dilakukannya karena proses ini tidak dapat diulang. Aplikasi akan melakukan proses pembentukan laporan keuangan dan analisa rasio akhir, menutup buku dan membentuk basisdata baru (untuk periode atau siklus akuntansi berikutnya).

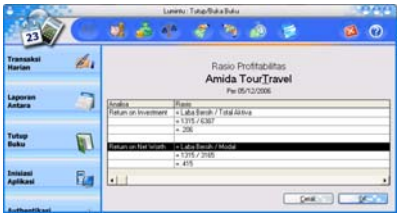
Pada saat membentuk basisdata baru ini aplikasi akan menyalin data referensi yang dibutuhkan dari periode sebelumnya serta "menarik" saldo untuk setiap perkiraan sebagai saldo awal pada jurnal yang baru.

Proses yang bekerja secara otomatis ini akan sangat membantu pengguna. Selain mempersingkat waktu, mempermudah pekerjaan dan menghasilkan perhitungan yang akurat.

Gambar *Snapshot Konfirmasi Proses Tutup/Buka Buku*

18. Pratayang Laporan Akhir

Seluruh Laporan sama benar dengan laporan antara. Perbedaannya hanya tersedianya sebuah aiken di *toolbar* untuk membuka buku baru.



Gambar *Snapshot Pratayang Laporan Akhir*

19. Pembentukan Gugus Kerja (*Teamwork Building*)

Tujuan akhir dari perekayasaan sebuah perangkat lunak tentu saja adalah perangkat lunak yang akan dihasilkan. Dalam hal pencapaian tujuan akhir perekayasaan, pengembangan perangkat lunak komersial bahkan memiliki masalah kritikal yang seringkali tidak diperhatikan dalam perekayasaan perangkat lunak non-komersial yang lebih bersifat tertutup. Dari hasil studi pustaka dan observasi atas beberapa perangkat lunak komersial (terutama yang berkaitan dengan akuntansi), selain tidak diperbolehkan adanya *bug* pada manajemen proses diperoleh beberapa fakta yang menunjukkan keunikan dari perangkat lunak komersial.

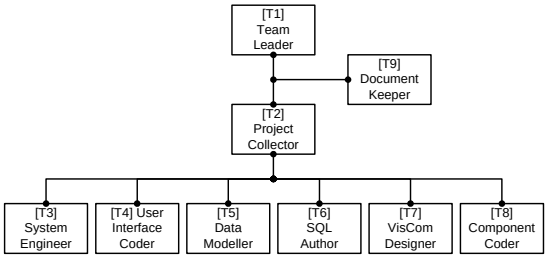
Dan kemudian, walaupun tidak langsung berkaitan dengan manajemen proses sebagai bagian inti dari sebuah perangkat lunak, *comfortablity level* atau tingkat kenyamanan pengguna dalam berinteraksi dengan program sangat ditentukan oleh desain visual dan kinerja dari *user interface*. Menurut beberapa penelitian, peran atau pengaruh *user interface* terhadap keberhasilan sebuah perangkat lunak di pasar mencapai angka sampai dengan 80 prosen. Jadi, keberhasilan sebuah perangkat lunak komersial sangat ditentukan oleh keberhasilan *software engineer* dalam merancang *user interface* dan kinerja dari *user interface* yang bersangkutan. Hal ini bahkan memunculkan cabang tersendiri di dalam disiplin atau ilmu *deskomvis* (desain komunikasi visual).

Berdasarkan hal tersebut maka rekruitmen atau pembentukan gugus kerja dari perekayasaan perangkat lunak ini harus memiliki paling tidak sumber daya manusia dengan keahlian sebagai berikut :

1. ***Team Leader;*** sebagai koordinator dan sekaligus pimpinan yang ditugasi untuk membawa tim ke tujuan, posisi ini mensyaratkan senioritas seorang *sofware engineer*. Secara umum sumber daya pemegang tanggungjawab ini harus berpengalaman dan terbiasa bekerja dalam gugus kerja di bidang *IT* khususnya dibidang perekayasaan perangkat lunak.
2. ***Analyst and Designer;*** anggota tim ini terdiri dari paling tidak satu orang yang memiliki kemampuan untuk melakukan analisis dan perancangan sistem dan sekaligus perangkat lunak (*system and software engineer*). *Analyst and Designer* dapat dibantu oleh paling tidak seorang *database modeller* dan dokumentator yang mengerti benar bidang yang menjadi tanggung jawabnya.
3. ***Coder/Programmer;*** seluruh dokumentasi perancangan yang terdiri dari catatan-catatan hasil analisis dan permodelan yang telah berhasil dilakukan oleh *analyst and designer* harus dapat diterjemahkan oleh bagian tim ini ke dalam bentuk program. Mulai dari prototipe, *build 1* s/d *n* dan versi finalnya.
4. ***Visual Communication Designer;*** atau yang lajim disebut sebagai seorang *graphics designer* adalah sumber daya manusia yang diberi tanggungjawab untuk merancang perangkat-perangkat komunikasi dengan pengguna dalam bentuk *graphical user interface* di dalam perangkat lunak yang akan dibangun.

20. Struktur Organisasi Gugus Kerja (*Teamwork*)

Keunikan karakter yang harus ada pada perangkat lunak komersial seperti telah diuraikan tersebut berpengaruh pada struktur organisasi dari *teamwork* yang akan melakukan rekayasa perangkat lunak komersial. Berikut ini adalah diagram struktur organisasi gugus kerja yang dibentuk khusus untuk penelitian ini.



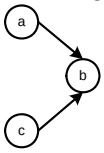
Gambar 4.26 : *Struktur Organisasi Gugus Tugas*

21. Sumber Daya Manusia (*Human Resources*)

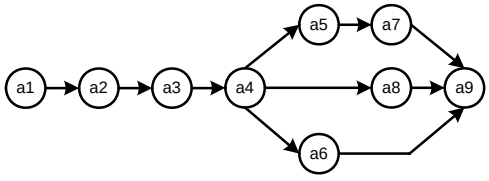
Keterbatasan akses terhadap sumber daya manusia yang dinilai mampu dan memenuhi kualifikasi memaksa beberapa posisi rangkap terjadi. Gugus kerja yang terdiri dari seorang *team leader, project collector, system engineer, user interface coder, data modeller, SQL author, visual communication designer, component coder* dan *document keeper* (9 posisi) hanya diemban oleh 5 (lima) orang personel.

22. Jadual Waktu (*Time Schedule*)

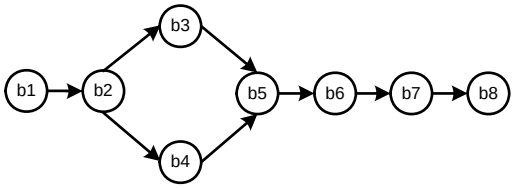
Jadual waktu dibentuk setelah *team leader* melakukan observasi, *survey* dan studi pustaka, gugus kerja telah terbentuk dan telah menerima *briefing* atau penjelasan proyek penelitian kepada seluruh anggota gugus kerja sehingga setiap anggota gugus kerja dapat memberikan kontribusi yang optimal terhadap upaya pembuatan skedul atau jadual waktu ini. Berikut ini adalah dokumentasi jadual waktu yang disajikan dengan menggunakan diagram *pert*.



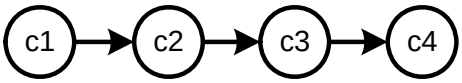
Gambar *Jadual Waktu Global Dalam Diagram Pert*



Gambar *Jadual Waktu Analisis Dan Perancangan Dalam Diagram Pert*



Gambar *Jadual Waktu Implementasi Dalam Diagram Pert*



Gambar *Jadual Waktu Administratif Dalam Diagram Pert*

Kemudian diagram *pert* pada Gambar 4.3 sampai denngan Gambar 4.5 diterjemahkan ke dalam diagram *gantt* agar daripadanya dapat diperoleh rencana anggaran biaya (RAB) khusus untuk sumber daya manusia.

Setiap rinci pekerjaan yang ada seharusnya tidak hanya dikaitkan dengan unsur waktu saja, setiap rinci pekerjaan harus pula dikaitkan dengan unsur peralatan, perlengkapan dan sumber daya manusia yang terlibat dalam pekerjaan ini.

23. PENGUJIAN PERANGKAT LUNAK

Pengujian atau evaluasi terhadap aplikasi ini merupakan tahapan akhir yang tidak dapat diabaikan begitu saja. Keberhasilan aplikasi ini sangat tergantung dari keberhasilan dari pengujian yang dilakukan. Karena pengujian merupakan suatu kejadian untuk mengidentifikasikan keberhasilan kelengkapan, keamanan dan kualitas pada aplikasi yang bersangkutan (Joko Nurjadi – PC Media, 2006). Berikut adalah jenis-jenis testing yang dilakukan atas aplikasi yang dihasilkan : (1) *Crash* (2) *Anomali* (3) *Fault* dan (4) *Mistake*. Sedangkan tahapan pengujian dimulai dari (1) *Unit/Component Testing* (2) *Integration Testing* (3) *System Testing* (4) *Acceptance Testing* dan (5) *Regression Testing*.

24. Unit/Component Testing

Unit/Component Testing sebenarnya telah dilakukan secara otomatis pada saat pengembangan aplikasi karena perangkat pengembangan yang digunakan merupakan sebuah IDE (*Integrated Development Environment*) yang sangat terpadu. Yaitu Microsoft Visual Basic versi ke 6.0. Oleh karena Tahap ini tidak dilaporkan kecuali jika tahap selanjutnya ternyata ditemui adanya *bug*.

25. Integration Testing

Pengujian dilakukan terhadap fungsi-fungsi sesuai *SRS* dan dengan modus penilai ber-skala 5.

Tabel 4.11. : *Hasil Integration Testing*

No	Responden	Uji1	Uji 2	Uji3	Uji 4
1.	Muhammad Ibrahim	4,00	3,00	4,00	3,80
2.	Fajar Widodo	3,90	3,20	4,00	2,90
3.	Yudi Wahyudi	3,80	3,20	3,80	3,10
4.	Muhammad Husin	4,00	3,50	4,00	3,50
5.	Wendro Prasetyo	4,00	3,40	3,80	3,80
	Hasil Rata-Rata Uji	3,94	3,26	3,92	3,42

Uji 1. Perbandingan Jumlah *bug* dengan fungsi sistem

Uji 2. Dapat menunaikan setiap fungsinya dengan baik dan benar

Uji 3. Sistem Kendali dan Navigasi mudah dan nyaman penggunaannya

Uji 4. Memiliki perangkat untuk manajemen berkas

26. System Testing

Sebenarnya pengujian harus pula dilakukan terhadap Sistem Operasi akan tetapi dengan asumsi bahwa Sistem Operasi telah teruji sebelumnya oleh pihak *vendor* maka pengujian hanya dilakukan terhadap komponen-komponen yang berupa *Dynamic Link Library* dan *ActiveX* kecuali yang standard (bawaan Sistem Operasi). Yang *intrinsic* (standard) maupun yang berupa *plug-in*. Hasilnya terlihat pada tabel berikut ini.

Tabel *Hasil System Testing*

Komponen	Hasil Uji
ActiveSkin 4.0 Type Library	<OK>
CDlg/Common Dialog	<OK>
Microsoft HTML Object Library	<NA>
Microsoft Windows Common Control s 5.0 (SP2)	<OK>
Microsoft Windows Common Control s – 2 5.0 (SP2)	<OK>
Shockwave Flash ActiveX Component	<NA>

27. Acceptance Testing

Acceptance Testing dilakukan gugus kerja sebagaimana pengujian sebelumnya.

Tabel 4.13. : *Hasil Acceptance Testing*

Responen	Keluwesan	Kemudahan	Muatan	Tampilan	Manfaat
Muhammad Ibrahim	3,00	3,20	3,80	3,90	3,50
Fajar Widodo	3,60	3,30	3,90	4,00	3,50
Yudi Wahyudi	3,50	3,40	3,60	3,30	3,50
Muhammad Husin	3,20	3,70	3,70	3,80	3,80
Wendro Prasetyo	3,20	3,50	3,80	4,00	3,80
Rata-Rata Uji	3,30	3,42	3,76	3,80	3,62

28. Regression Testing

Karena metode yang digunakan adalah metode *Waterfall* maka untuk setiap tahapan pengujian dilakukan berkali-kali dan hasil pengujian yang diterakan pada masing-masing tahapan adalah resultan dari setiap pengujian yang dilakukan. Sehingga dari fakta tersebut maka secara teoritis pengujian pada tahapan ini tidak perlu dilakukan.

Dalam seluruh pengujian diketemukan 144 *bug* pada berbagai komponen dan fungsi tetapi sesuai dengan prinsip-pinsip *PLC* (*Programming Life Cycle*), *bug* tersebut telah berhasil diatasi satu demi satu dan bertahap

KESIMPULAN

Kesimpulan dari penelitian ini adalah :

1. Setelah dilakukan uji implementasi maka perangkat lunak yang kemudian diberi Tajuk Lumintu versi 1.0 terbukti dapat memenuhi spesifikasi perangkat lunak komersial.
2. Hasil perancangan Sistem Informasi Akuntansi Dengan Analisa Rasio Untuk Usaha Jasa telah berhasil diimplementasikan dalam bentuk Aplikasi berbasis *desktop* dengan perangkat pengembang utama *Microsoft™ Visual BASIC® 6.0*
3. Setelah dilakukan uji implementasi maka perangkat lunak yang kemudian diberi Tajuk Lumintu versi 1.0 terbukti dapat memenuhi spesifikasi perangkat lunak komersial.

Pustaka

[1] Fatoni, Tri Irianto Tjendrowasono (2008), Sistem Informasi Akademik Pada Lembaga Pendidikan Alfa Farma Husada Surakarta, Jurnal Speed *Volume 1 No 1 - 2008 - ijns.org*

[2] Ayu Fiska Nurryna (2008), Sistem Informasi Akademik Universitas Surakarta Berbasis Web, Jurnal Speed *Volume 1 No 1 - 2008 - ijns.org*

[3] Hermanto, Ayu Fiska Nurina, Haryani (2008), Pembuatan Sistem Informasi Nilai Raport Berbasis

[4] Website di SMP Negeri 7 Surakarta, Jurnal Speed *Volume 1 No 1 - 2008 - ijns.org*