

Rancang Bangun Aplikasi Business Intelligence Berbasiskan Arsitektur Aplikasi Akuntansi Accurate

Philbert Lukito Setiawan¹, Adi Suryaputra Paramita²

Abstrak— Perkembangan teknologi berdampak pada segala aspek, salah satunya adalah bisnis. Terdapat sebuah fenomena teknologi yang jika tidak dapat diatur dengan baik dapat menimbulkan dampak negatif. Fenomena ini disebut *big data*. Untuk menjawab fenomena ini, digunakanlah *business intelligence* yang menjadi solusi dengan memberikan *additional value* pada data. Dengan BI, pelaku usaha dapat melakukan analisa terhadap usaha yang dapat digunakan sebagai bahan penyusunan strategi. Di sisi lain, perkembangan teknologi pada dunia usaha juga mulai tampak di Indonesia. Salah satu aplikasi yang paling banyak digunakan oleh para pelaku usaha adalah *accurate*. Dikarenakan *target market* dari aplikasi *accurate* pasar general, *accurate* tidak memiliki beberapa fitur khusus yang diperlukan oleh perusahaan. Melalui penelitian ini diharapkan dapat menciptakan *business intelligence* pada arsitektur *accurate* yang dapat memberikan *additional value* bagi perusahaan.

Penelitian ini berfokus pada perancangan aplikasi *business intelligence* serta pembuatan *query* data yang mengubah data dari *database accurate* menjadi data yang siap dianalisa. Pembuatan aplikasi menggunakan bahasa pemrograman *visual basic* dan *database firebird*. Proses analisa kebutuhan dilakukan dengan cara wawancara, studi aplikasi, studi komparasi dan studi *database*. Berdasarkan analisa, *database accurate* dipetakan menjadi empat subjek besar yaitu *sales, purchase, inventory* dan *person* yang menjadi sumber data dari *business intelligence* yang dibuat. Untuk proses uji coba, penelitian menggunakan *database* real perusahaan yang telah menggunakan aplikasi *accurate*. Dari hasil uji coba, dapat disimpulkan bahwa aplikasi telah berhasil membuat analisa berdasarkan sumber data yang telah dibuat.

Kata Kunci: *Business Intelligence*, aplikasi akuntansi, *Accurate*, rancang bangun, *big data*

¹ Mahasiswa Program Studi Teknik Informatika, Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 INDONESIA (telp: 031-745 1699; fax: 031-745 1698; e-mail: plukito01@student.ciputra.ac.id)

² Dosen Program Studi Teknik Informatika, Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 INDONESIA (telp: 031-745 1699; fax: 031-745 1698; e-mail: plukito01@student.ciputra.ac.id)

Abstract— Technological developments had an impact on all aspects including in business. There is a problem when technology can't be regulated properly cause a negative impact, this called *big data*. The solution for this problem is *business intelligence* which providing additional value in the data. When BI implemented, businesses can perform an analysis of the business and can be used as a strategy formulation. On the other hand, the development of technology in the business world also began in Indonesia. One of the applications most used by user is *accurate*. Due to the target market of accurate application of general market, *accurate* doesn't have some special features that are required by the company. The purpose of this research is expected to create *business accurate intelligence* on an architecture that can provide additional value for the company.

This research focuses on the design and manufacture of *business intelligence* applications query the data that converts data from the *database accurate* data is ready to be analyzed. Making an application using *Visual Basic* programming language and *database firebird*. The process of needs analysis was conducted by interview, application studies, comparative studies and *database* studies. Based on the analysis, *accurate database* is mapped into four major subjects, namely *sales, purchase, inventory* and the *person* who becomes the source of the data from *business intelligence* made. For the test, the research uses a *database* of real companies that have used *accurate* application. From the test results, can be concluded that the application has successfully made the analysis based on source data that has been created.

Keywords: *Business Intelligence*, *Accounting*, *Accurate*, *Development*, *Big Data*

I. PENDAHULUAN

Perkembangan teknologi merubah segala aspek kehidupan manusia, tak terkecuali dalam dunia bisnis. Terdapat sebuah fenomena teknologi yang jika tidak diatur dengan baik dapat menimbulkan dampak yang buruk bagi bisnis. Fenomena tersebut dinamakan *big data*. *Big data* dapat menjadi bumerang bagi perusahaan dikarenakan tidak dapat memberikan nilai tambah namun biaya yang besar bagi perusahaan. Di sinilah *business intelligence* dapat berperan. Menurut [1], *business intelligence* adalah sebuah istilah yang menggabungkan arsitektur, tools, *database, analytical tools*, aplikasi, dan metodologi. Dengan mengaplikasikan teknologi *business intelligence* pada perusahaan, semula data yang tidak memiliki nilai ekonomis bagi perusahaan dapat diubah menjadi bahan bernilai tinggi untuk menentukan strategi bisnis kedepan.

Salah satu aplikasi akuntansi yang paling banyak digunakan di Indonesia adalah *accurate*. Saat ini pengguna aplikasi *accurate* telah mencapai 80.000 pengguna per tanggal September 2014 [2]. Salah satu kunci mengapa aplikasi ini menjadi pemimpin pasar adalah dengan strategi pengembangan produk secara terus menerus dan konsisten yang mengikuti perkembangan teknologi yang ada serta fitur yang dimilikinya difokuskan pada kebutuhan secara umum perusahaan.

Akibat dari strategi ini adalah terdapat beberapa fitur yang tidak dimiliki oleh *accurate* yang dibutuhkan oleh perusahaan. Salah satunya adalah *accurate* tidak memiliki *tools business intelligence* pada aplikasinya. Dengan demikian adalah sebuah kesempatan yang baik untuk merancang aplikasi *business intelligence* yang berbasiskan arsitektur aplikasi *accurate*, dengan melihat perkembangan fenomena *big data* yang semakin marak di Indonesia. Disamping itu, diharapkan penelitian ini juga dapat mendorong perkembangan dunia bisnis di Indonesia dan memberikan wawasan bagi dunia usaha skala menengah melalui penyediaan *tools business intelligence*.

II. LANDASAN TEORI

A. Business Intelligence

Menurut buku *Business Intelligence – A Managerial Approach* [1], *business intelligence* adalah sebuah istilah yang menggabungkan arsitektur, *tools*, *database*, *analytical tools*, aplikasi, dan metodologi. Tujuan utama dari BI adalah memungkinkan akses interaktif (terkadang secara *real time*) kepada data, untuk memungkinkan manipulasi data dan untuk memberikan kemampuan terhadap manajer bisnis dan analis untuk melakukan analisis yang tepat.

Stackowiak, Rayman and Greenwald mengartikan BI sebagai sebuah proses pengambilan data dalam jumlah yang besar, menganalisa data tersebut, dan mempresentasikan laporan tingkat tinggi yang merangkum esensi dari data tersebut menjadi dasar dari langkah bisnis yang memungkinkan manajemen untuk membuat keputusan dasar bisnis setiap harinya [3].

Dari pengertian diatas, dapat disimpulkan bahwa BI membantu perusahaan dalam menyusun strategi atau keputusan bisnis dengan cara menyediakan informasi yang telah diolah dari data dalam jumlah yang besar yang dimiliki oleh perusahaan itu sendiri.

Terdapat 4 komponen dasar dari BI yang saling bersinergi agar sebuah BI dapat berfungsi yaitu

- *Data Warehouse*

Data warehouse berperan sebagai sumber data dari BI. Menurut [4], “*A data warehouse is a subject oriented, nonvolatile, time variant collection of data in support of management decisions*” yang berarti bahwa sebuah *data warehouse* merupakan koleksi data yang terorientasi pada subyek, tidak mengalami perubahan, serta memiliki rentang waktu yang cukup lebar yang

berfungsi dalam mendukung pengambilan keputusan management.

- *Business Analytics*

Secara singkat adalah kumpulan peralatan untuk memanipulasi, menambang dan menganalisa data yang terdapat di dalam *data warehouse*. Peralatan – peralatan ini kemudian dibagi menjadi 2 bagian besar yakni

- *Report and Queries*

Termasuk didalamnya segala bentuk pelaporan baik secara statis (tidak berubah) ataupun dinamis sesuai dengan perubahan data dan setiap macam query yang ada seperti *drill down*, *multidimensional view*, pengelompokan, dsb.

- *Data, text dan Web mining dan peralatan matematika level atas dan statistik*

Data mining merupakan adalah sebuah proses untuk menemukan relasi atau informasi yang tidak diketahui didalam *database* besar ataupun *data warehouse* dengan menggunakan peralatan *intelligent*. Text mining adalah semiautomated proses ekstraksi pola dari data tidak terstruktur dalam jumlah yang besar. Perlu diingat bahwa perbedaan dengan data mining adalah sumber data dari data mining adalah data terstruktur yang terdapat di dalam *database*. Sedangkan *web mining* adalah sebuah proses penemuan relasi intrinsik (informasi menarik dan berguna) dari data web dimana diekspresikan dalam bentuk text, linkage atau informasi berguna.

Dari penjelasan di atas dapat disimpulkan bahwa ke 3 hal diatas memiliki kesamaan yakni sebuah proses ekstraksi pola atau relationship yang dalam prosesnya dibantu oleh berbagai peralatan *intelligent*.

- *Business Performance Management*

BPM memiliki berbagai nama lain didalam literatur bisnis seperti CPM atau *Corporate Performance Management*, EPM (*Enterprise Performance Management*), dan SEM (*Strategic Enterprise Management*) [5], BPM mengacu pada bisnis proses, metodologi, metrik, dan teknologi yang digunakan oleh enterprise untuk mengukur, memonitor dan mengelola performa bisnis. BPM mencakup 3 komponen penting yakni,

- Sekumpulan proses terintegrasi proses analisa dan perputaran tertutup dalam management yang didukung oleh teknologi dimana memetakan aktifitas operasional dan finansial.
- Peralatan bagi bisnis untuk menjelaskan tujuan strategis and kemudian mengukur serta mengelola performa terhadap tujuan.
- Sekumpulan proses inti yang termasuk didalamnya perencanaan keuangan dan operasional, konsolidasi dan pelaporan,

pemodelan, analisa, dan memonitor KPI (*Key Performance Indicator*) yang terhubung dengan strategi organisasi.

- *User Interface*

User Interface atau UI adalah sebuah program yang mengontrol tampilan untuk user dan yang memungkinkan user untuk berinteraksi dengan sistem. Salah satu UI yang digunakan adalah *dashboard*. *Dashboard* menyediakan penglihatan visual secara komprehensif dari pengukuran performa perusahaan, tren dan pengecualian. Peralatan ini mengintegrasikan informasi dari beberapa area bisnis. Selain *dashboard*, hal yang termasuk ke dalam UI adalah segala bentuk alat penyiaran informasi seperti *corporate portals*, *digital cockpits*, dsb.

B. Database

1) Pengertian Database

Dalam buku *Database Fundamentals*, *database* didefinisikan sebagai repositori data yang didesain untuk mendukung penyimpanan data secara efisien, pengambilan data dan perbaikan. Berbagai tipe *database* diciptakan untuk mendukung berbagai kebutuhan industry[6].

Elmasri mendefinisikan *database* lebih dibatasi pada arti implisit khusus yakni [7] :

1. *Database* merupakan penyajian suatu aspek dari dunia nyata.
2. *Database* merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tidak memiliki arti, tidak dapat disebut sebagai basis data.
3. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa user dan beberapa aplikasi yang sesuai dengan kepentingan user.

2) Firebird SQL

Firebird adalah *database* SQL yang berbasis teknologi RDBMS. Dikembangkan sejak tahun [9], *Firebird* adalah sebuah *database* bersifat *open source* yang berbasis pada *source code* Borland. Versi terakhir yang telah dikeluarkan oleh *firebird* adalah versi 2.5.4. Terdapat beberapa fitur serta alasan yang membuat *firebird* banyak digunakan dalam pengembangan aplikasi.

- Open Source penggunaan *firebird* tidak membutuhkan biaya apapun.
- Mendukung *platform* yang secara umum digunakan (Windows, Linux, MacOS, HP-UX, AIX, Solaris, dsb)
- Multi-generations Arsitektur
- Menggunakan bahasa SQL
- *Logging* dan *Monitoring*
- *Standard security*

Didukung oleh banyak *developer tool* (Visual Studio, eclipse, php, dsb)

C. Metadata

Metadata adalah data yang menjelaskan tentang data. *Metadata* menjelaskan struktur dari dan pengertian lainnya tentang data yang sangat berkontribusi pada keefektifan atau ketidakefektifan dari penggunaan data yang ada. Berdasarkan penelitian yang dilakukan oleh Mehra diindikasikan bahwa hanya sedikit organisasi yang mengerti *metadata* dan lebih sedikit lagi yang mengetahui serta mengimplementasikan strategi *metadata*[8].

Sedangkan menurut dokumen yang diterbitkan oleh (National Information Standards Organization (NISO) n.d.), *metadata* berarti informasi terstruktur yang menjelaskan, menggambarkan dan membuat lebih mudah dipahami, digunakan dan diolah dari sumber informasi.

D. Aplikasi Accurate

Accurate adalah sebuah aplikasi akuntansi yang dikembangkan oleh CPSSoft yang berpusat di Jakarta. Pertama kali diluncurkan pada November 1999 dengan versi 1.0, aplikasi ini terus dikembangkan dan disesuaikan dengan perkembangan zaman serta peraturan IFRS (*International Financial Reporting Standards*). Genap 16 tahun sejak pertama kali peluncurannya, *accurate* telah mencapai versi 4.0. Aplikasi *accurate* sendiri dibagi menjadi 3 varian yakni *Standard*, *Deluxe*, dan *Enterprise Version* yang perbedaan utamanya terletak pada modul yang tersedia.

E. UML

UML (Unified Modeling Language) adalah salah satu bagian dari keluarga notasi secara grafik yang membantu mendeskripsikan dan mendesain sistem software yang berbasis object oriented (OO) style [10]. UML adalah standar terbuka yang diatur oleh Object Management Group (OMG) serta terlahir dari kumpulan berbagai OO graphical modeling.

Di dalam bukunya yang berjudul *UML Distilled*, Fowler mengungkapkan bahwa secara umum UML digunakan untuk 2 hal yakni sebagai *blueprint* dan *skect*. Perbedaan utama dari keduanya adalah *blueprint* berfokus pada kelengkapan pencatatan dan detil. Sedangkan *sketch* berfokus pada penekanan informasi penting sehingga dalam pengaplikasiannya *sketch* bersifat tidak lengkap. Kedua cara penggunaan tersebut dapat diaplikasikan pada tipe penggunaan UML yakni *forward engineering*, dimana UML dibuat sebelum proses *coding* dan *reverse engineering* yang membuat UML setelah proses pembuatan. Keduanya memiliki fungsi masing sehingga dapat disesuaikan dengan kebutuhan.

Seiring dengan berjalannya waktu, UML terus mengalami perkembangan. Versi UML yang digunakan penulis adalah UML V2.0 dimana pada versi terdapat

perubahan paling banyak dari versi sebelumnya. UML V2.0 memiliki 13 tipe diagram yang memiliki kegunaan tertentu.

F. XML

XML adalah singkatan dari *Extensible Markup Language*. Dalam bukunya berjudul *Beginning XML* [11] penulis mengungkapkan bahwa XML merupakan sebuah bahasa yang memiliki tujuan untuk memudahkan komputer memahami sebuah data, kemampuan memberikan *metadata*, dan kemampuan untuk dapat digunakan secara universal. Sebelum XML dikenal, terdapat sebuah bahasa pemrograman berbasis teks yang disebut dengan SGML (*Standard Generalized Markup Language*). SGML didesain sebagai sebuah standar dalam membuat penanda data (*markup*) untuk segala tujuan dan digunakan pada sistem manajemen dokumen skala besar. Salah satu bahasa yang berbasis SGML yang paling dikenal adalah HTML (*HyperText Markup Language*). HTML menggunakan banyak konsep SGML di dalam bahasanya. Tujuannya adalah agar HTML dapat digunakan pada aplikasi apapun yang memiliki kemampuan membaca dan agar HTML dapat hanya dibuat dengan teks editor simpel, hal ini dikarenakan HTML hanya berbasis teks. Namun walaupun berhasil, jangkauan dari HTML sangatlah terbatas. HTML hanya dapat menampilkan sebuah konten sesuai dengan petunjuk namun tidak mengetahui konten itu sendiri sehingga ketika membutuhkan pertukaran data, hal ini tidaklah memungkinkan. Oleh sebab itulah XML diciptakan untuk menjawab permasalahan ini. Serupa dengan HTML, XML juga merupakan bagian dari SGML dengan tujuan yang sama, namun mengurangi kompleksitas setinggi mungkin.

Hal penting yang harus dimengerti dari XML adalah XML tidaklah benar – benar sebuah bahasa, namun lebih kepada sebuah standar untuk bahasa yang sesuai dengan kriteria XML. Dengan kata lain, XML menjelaskan *syntax* yang dibuat oleh bahasa yang kita ciptakan. Dengan demikian, XML sangatlah mudah diterapkan pada berbagai sistem dikarenakan kemampuannya tersebut.

III. ANALISIS DAN DESAIN

A. Analisa Kebutuhan

Sebuah sistem yang baik adalah sebuah sistem yang dapat mengakomodasi kebutuhan dari penggunanya. Oleh karena itu, perlu dilakukan analisa kebutuhan yang baik agar dapat menangkap kebutuhan dari konsumen. Analisa menggunakan metode:

1) Wawancara

Narasumber dari kegiatan wawancara ini adalah Bpk. Darwin Tjoe selaku pendiri CPSSoft yang mengembangkan *software accurate*.

Adapun kesimpulan dari wawancara tersebut adalah target pasar yang dimiliki oleh aplikasi *accurate* berfokus

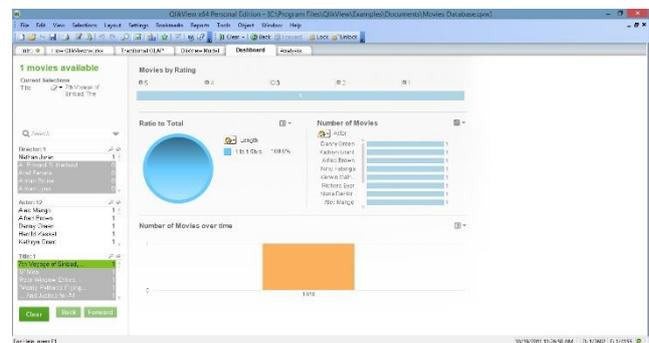
pada perusahaan skala kecil hingga menengah, secara spesifik jenis usaha yang ditargetkan adalah perusahaan *trading*, perusahaan kontraktor dan perusahaan manufaktur yang telah tercemrin dari fitur – fitur yang disediakan oleh aplikasi *accurate*.

2) Studi Aplikasi

Studi aplikasi dilakukan dengan menggunakan serta menganalisa aspek – aspek penting dari aplikasi. Hal ini penting untuk menentukan kebutuhan dari konsumen *accurate*. Beberapa hal yang dianalisa dari aplikasi adalah report yang dihasilkan aplikasi, modul – modul yang digunakan dan cara kerja transaksi.

3) Studi Komparasi

Studi komparasi dilakukan dengan mencoba aplikasi *business intelligence* lainnya yakni QlikView 11. Adapun hal – hal yang diperhatikan oleh penulis dalam melakukan studi adalah fitur, tampilan, serta kebutuhan sumber data yang dimiliki oleh aplikasi pembanding. Berikut salah satu contoh tampilan QlikView 11 yang terlihat pada gambar 1.



Gambar 1. Tampilan Dashboard QlikView11

4) Analisa Database Accurate

Database yang dimiliki oleh aplikasi *accurate* memiliki tabel 231 buah yang seluruhnya berfungsi untuk mengakomodasi kebutuhan penyimpanan data. Namun dalam penelitian kali ini, tidak semua tabel akan dibahas namun hanya tabel yang berhubungan dengan transaksi *Purchase*, *Sales*, *Inventory* dan *Person* yang akan menjadi fokus utama dari penulis. Keempat transaksi tersebut dipilih dikarenakan keempat transaksi tersebut adalah transaksi yang paling sering digunakan di dalam kegiatan perusahaan.

B. Desain Sistem

Aplikasi *business intelligence* memiliki 8 *use case* seperti yang tampak pada gambar 2. Kedelapan *use case* tersebut menggambarkan seluruh proses yang terjadi di dalam aplikasi.

Adapun seperti yang terlihat di dalam gambar 2, aplikasi hanya memiliki 1 *user*. Hal ini disebabkan karena aplikasi dirancang agar dapat digunakan oleh pengguna

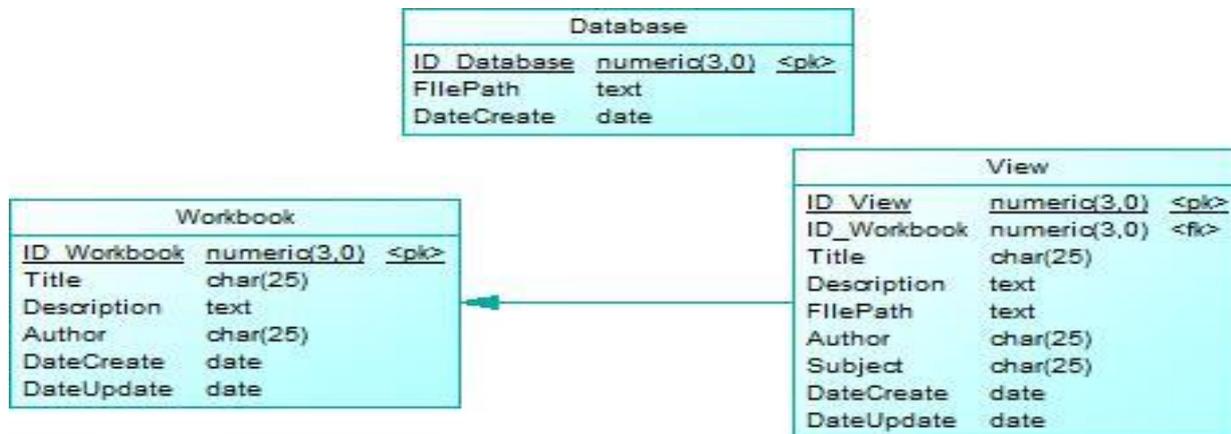


Gambar 2. Arsitektur Aplikasi

accurate.

1) Desain Database

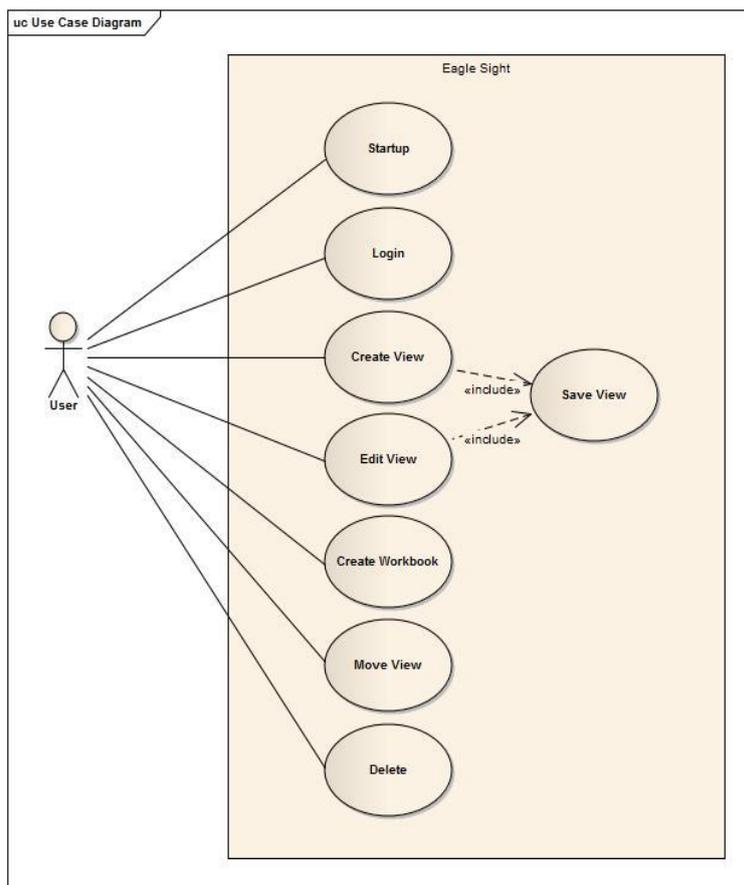
Untuk menyimpan data, aplikasi memiliki database tersendiri yang tergambar dalam gambar 3.



Gambar 3. Physical Data Model Database Aplikasi

Gambar 4 menjelaskan tampilan susunan database aplikasi yang hanya terdiri dari 3 tabel yakni database untuk menyimpan lokasi file database accurate, workbook

untuk mengelompokkan view, dan view yang berfungsi untuk menyimpan lokasi file XML view.



Gambar 4. Use Case Diagram

2) Database Mapping

Salah satu sifat dari *data warehouse* adalah *subject oriented*. Oleh karena itu *database accurate* perlu ditransformasikan ke bentuk yang lain agar data – data yang ada dapat dianalisa sesuai dengan kebutuhan. Sesuai dengan analisa yang telah dilakukan, maka terdapat 4 domain subjek yang akan dipetakan. Pemetaan tersebut didasarkan pada *field – field* yang dibutuhkan pada proses query berdasarkan subjek. Dengan memetakan *field* yang benar maka analisa sumber data perusahaan dapat lebih akurat serta komprehensif sehingga proses analisa perusahaan dapat berjalan dengan lebih baik.

- *Purchase*

Tabel 1. Purchase Subject Field

Asal Tabel	Alias
APCHEQ.CHEQUEID	PR ID
APCHEQ.PAYMENTDATE	PR Date
APCHEQ.SEQUENCENO	No PR
APINV.APINVOICEID	PI ID
APINV.DPAMOUNT	DP Amount
APINV.INVOICEAMOUNT	PI Amount
APINV.INVOICEDATE	PI Date
APINV.INVOICENO	PI No
APINV.ISDPPO	PI Cek DP
APINV.OWING	Hutang pada PI
APINV.PAIDAMOUNT	Jumlah terbayar PI
APINVCHQ.DISCOUNT	Disc PR
APINVCHQ.PAYMENTAMOUNT	Jumlah PR
PERSONDATA.NAME	Vendor
PO.DP	DP pada PO
PO.POAMOUNT	Jumlah PO
SHIPMENT.NAME	Metode Pengiriman
TAX.TAXNAME	Pajak
TERMOPMT.TERMNAME	Term

Tabel 1 menggambarkan semua transaksi *purchase* yang terjadi pada perusahaan.

- *Inventory*

Tabel 2. Inventory Subject Field

Asal Tabel	Alias
ITEM.ITEMNO	No Item
ITEM.ITEMTYPE	Tipe Item

ITEM.MINIMUMQTY	Min Quantitas Item
ITEMCATEGORY.NAME	Nama Kategori
ITEMHISTORY.COST	Biaya Item
ITEMHISTORY.DESCRPTION	Deskripsi Trans
ITEMHISTORY.INVOICEID	ID Transaksi
ITEMHISTORY.QUANTITY	Quantitas Trans
ITEMHISTORY.SELLINGPRICE	Harga Jual Item
ITEMHISTORY.TXDATE	Tanggal Transaksi
ITEMHISTORY.TXTPYE	Tipe Transaksi
TAX.TAXNAME	Pajak

Tabel 2 berisikan *field – field* pada transaksi *inventory*. Pada tabel ini, seluruh histori transaksi terekam dengan jelas.

- *Sales*

Tabel 3. Sales Subject Field

Asal Tabel	Alias
ARINV.ARINVOICEID	SI ID
ARINV.DPAMOUNT	Jumlah DP SI
ARINV.GETFROMOTHER	SI Cek DP
ARINV.INVOICEAMOUNT	Jumlah SI
ARINV.INVOICEDATE	SI Date
ARINV.INVOICENO	No SI
ARINV.OWING	Hutang SI
ARINV.PAIDAMOUNT	Jumlah Terbayar SI
ARINVPMT.DISCTAKENAMOUNT	Jumlah Diskon SR
ARINVPMT.PAYMENTAMOUNT	Jumlah SR
ARMPT.PAYMENTDATE	SR Date
ARMPT.PAYMENTID	SR ID
ARMPT.SEQUENCENO	No SR
PERSONDATA.NAME	Customer
SHIPMENT.NAME	Metode Pengiriman
SO.DP	DP SO
SO.INVAMOUNT	Jumlah SO
TAX.TAXNAME	Pajak
TERMOPMT.TERMNAME	Term

Tabel 3 berisikan detail transaksi penjualan yang dilakukan oleh perusahaan. Detil tersebut mencakup DP yang dilakukan, metode pengiriman, tanggal pembayaran, dsb.

• Person

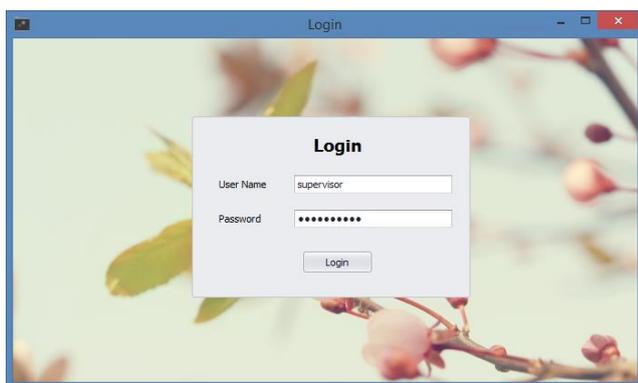
Tabel 4. Person Subject Field

Asal Tabel	Alias
APINV.PAIDAMOUNT	Total Tran. Vendor
ARINV.PAIDAMOUNT	Total Tran. Cust.
CURRENCY.CURRENCYNAME	Nama Kurs
CUSTOMERTPYE.TYPENAME	Tipe Customer
PERSONBALANCE.AMOUNT1	Jumlah Balance
PERSONBALANCE.OPENINGBAL	Jumlah Bal. Awal
PERSONDATA.ADDRESSLINE 1	Alamat Baris 1
PERSONDATA.ADDRESSLINE 2	Alamat Baris 2
PERSONDATA.CITY	Alamat Kota
PERSONDATA.COUNTRY	Alamat Negara
PERSONDATA.CREDITLIMIT	Batas Kredit
PERSONDATA.ID	ID Person
PERSONDATA.NAME	Nama Person
PERSONDATA.PERSONNO	No Person
PERSONDATA.PERSONTYPE	Tipe Person
PERSONDATA.STATEPROV	Alamat Provinsi
PERSONDATA.SUSPEND	Status Suspend
TAX.TAXNAME	Pajak
TERMOPMT.TERMNAME	Term Pembayaran

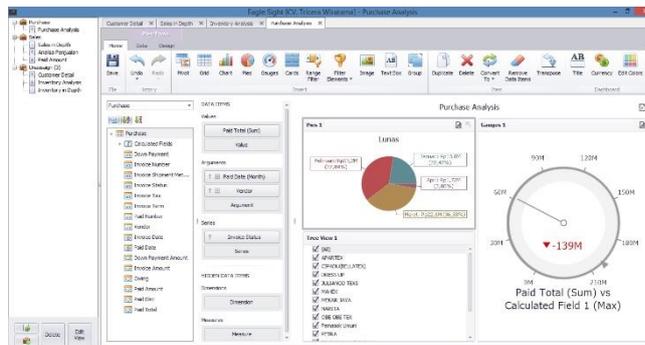
Sedangkan tabel 4 menggambarkan transaksi yang dilakukan tiap *customer / vendor*. Pada tabel ini dapat terlihat status hubungan antara perusahaan dengan *customer / vendor* dengan jelas.

3) User Interface

Berikut salah satu contoh tampilan dari aplikasi yang telah dibuat.



Gambar 1. User Interface - Login



Gambar 2. User Interface - Home

IV. PENGUJIAN

Proses pengujian dilakukan dengan cara mengganti sumber data dari aplikasi menjadi data *real* yang digunakan perusahaan dan membuat skenario penggunaan aplikasi. Tujuan dari pengujian ini adalah memastikan bahwa aplikasi dapat berjalan sesuai dengan yang diharapkan.

A. Profil Perusahaan

- Nama : X (Dirahasiakan)
- Jenis Usaha : Manufaktur
- Barang : Busana Muslim Wanita
- Lokasi : Jombang
- Lama Penggunaan : 4 Bulan

B. Skenario

Skenario disesuaikan dengan *use case* yang telah disusun sebelumnya. Oleh karena itu dibuatlah 7 skenario yakni *login, save view, edit view, save workbook, edit workbook, delete, move view*. Berikut salah satu contoh pengujian skenario yang telah dilakukan.

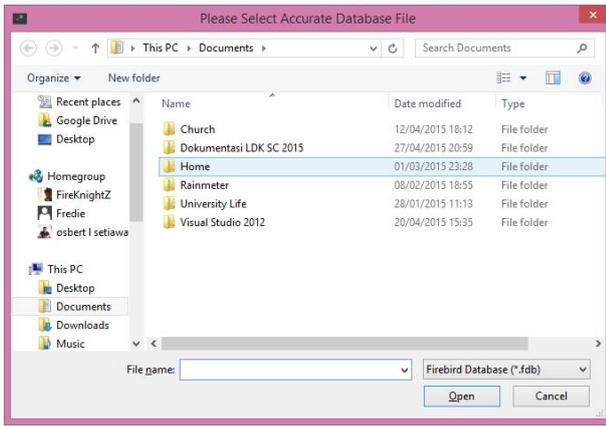
1) Login

Skenario ini akan menggambarkan proses *login* pertama kalinya oleh *user* (belum ada data mengenai *database* tersimpan)



Gambar 3 Skenario Login – Startup (Lokasi Kosong)

User diminta untuk mencari *file database* pada komputer. Untuk mencari *file*, dapat langsung mengisi *path* pada *textbox* atau menggunakan *OpenFileDialog* dengan mengklik tombol di sebelah kanan *textbox* yang tergambar pada gambar 4.



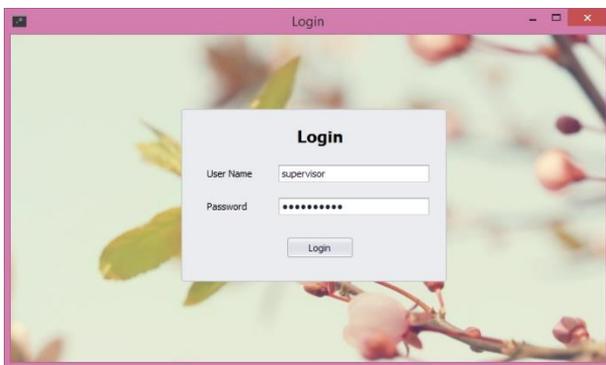
Gambar 4 Skenario Login - OpenFileDialog

Gambar 5 menunjukkan bahwa lokasi telah terisi dan user dapat lanjut untuk login.



Gambar 5 Skenario Login - Startup (Lokasi Terisi)

User memasukan password dan username standar versi accurate yakni username: supervisor dan password: supervisor. Untuk memverifikasi kredensial dan masuk ke dalam aplikasi, user menekan tombol “login”



Gambar 6 Skenario Login - Login

Jika username dan password yang dimasukan benar, maka user akan diarahkan kepada halaman utama dari aplikasi.

C. Hasil Pengujian

Berdasarkan pengujian yang telah dilaksanakan, aplikasi dapat berjalan dengan baik sesuai dengan skenario yang telah dibuat. Namun dari pengujian juga ditemukan masalah pada bagian loading data dari database accurate. Proses loading memakan waktu

sekitar 2 menit 20 detik yang berakibat hang pada aplikasi.

Setelah dilakukan uji coba, didapati bahwa masalah disebabkan oleh query yang tidak optimal sehingga mengakibatkan proses query yang lama. Untuk itu peneliti melakukan perubahan yang signifikan pada query yang berakibat pada berubahnya waktu tunggu menjadi kurang dari 1 detik. Contoh hasil akhri query tergambar pada gambar 11.

```

Select max(a.ARINVOICEID) as "SI ID",
max(p.PAYMENTID) as "SR ID", max(pd.Name) as
"Customer"
, max(a.INVOICENO) as "SI No",
max(p.SEQUENCENO) as "SR No"
, max(a.INVOICEDATE) as "SI Date",
max(p.PAYMENTDATE) as "SR Date"
, max(IIF(a.GETFROMOTHER = 0, a.INVOICEAMOUNT,
so.INVAMOUNT)) as "SI Amount"
, sum(ap.PAYMENTAMOUNT) as "SR Amount"
, sum(IIF(ap.DISTAKENAMOUNT = 0, null,
ap.DISTAKENAMOUNT)) as "SR Disc"
, Max(IIF(a.GETFROMOTHER = 0, IIF(a.DPAMOUNT =
0, 'No', 'DP'), IIF(a.GETFROMOTHER = 2, 'DP',
'No')))) as "DP"
, Max(IIF(a.GETFROMOTHER = 0, IIF(a.DPAMOUNT =
0, NULL, a.DPAMOUNT), IIF(so.dp = 0, null,
so.DP))) as "DP Amount"
, max(IIF(a.owing = 0, null, a.owing)) as
"Owing"
, sum((IIF(a.GETFROMOTHER = 0, a.DPAMOUNT,
so.DP)) + ap.PAYMENTAMOUNT) as "Paid Total"
, max(IIF(T.TERMNAME is not null, T.TERMNAME,
'-')) as "SI Term", max(IIF(S.NAME is NULL, '-',
S.NAME)) as "SI Shipment Method"
, max(ta.TAXNAME) as "SI Tax"
, Max(IIF(a.GETFROMOTHER = 0, IIF(a.OWING =0,
'Lunas', iif((a.PAIDAMOUNT + a.dpamount) /
a.INVOICEAMOUNT >= 0.5, '>50%',
IIF((a.PAIDAMOUNT + a.dpamount) /
a.INVOICEAMOUNT =0, 'Belum Bayar', '<50%'))),
IIF(SO.DP / so.INVAMOUNT >= 0.5, '>50%',
IIF(so.DP / so.INVAMOUNT =0, 'Belum Bayar',
'<50%')))) as "Status Pembayaran"
From ARINV a
Full Join ARINVPMT ap on a.ARINVOICEID =
ap.ARINVOICEID
Left Join ARPMT P on ap.PAYMENTID = p.PAYMENTID
Left Join ARINVDet ad on a.ARINVOICEID =
ad.ARINVOICEID
Left Join SO SO on ad.SOID = so.SOID
Left Join TERMOPMT T on a.TERMSID = t.TERMID
Left Join Shipment S on a.SHIPVIA = s.SHIPID
Left Join Tax Ta On a.TAX1ID = Ta.TAXID
Left Join PersonData PD on a.CUSTOMERID = pd.ID
Where (a.getfromother = 0 or a.getfromother is
null or (a.getfromother = 2 and so.PROCEED =
0)) and a.DELIVERYORDER = 0 and
ap.PAYMENTAMOUNT > 0 and (ad.SEQ = 1)
Group By ad.ARINVOICEID
Order by (case when max(a.ARINVOICEID) is null
then 0 end), 1
    
```

Gambar 11. Snippet Query Sales - Revisi

V. KESIMPULAN

1. Aplikasi mampu melakukan *query* terhadap *database accurate* sesuai subjek maupun *database* aplikasi yang telah ditentukan sebelumnya dengan cepat dan tanpa gangguan.
2. Aplikasi mampu untuk mengolah sumber data dalam bentuk *datatable* di dalam *view* dalam berbagai bentuk seperti *chart*, *gauge*, dsb.
3. Aplikasi mampu melakukan fungsi *drilldown* sesuai dengan parameter yang diinputkan oleh *user*.
4. Aplikasi mampu menyimpan hasil *view* dalam bentuk *file XML* sehingga dapat *diload* kembali ketika dibutuhkan.
5. Semua fitur pendukung aplikasi seperti *rename file*, *edit properties*, *delete file*, dsb dapat berjalan dengan baik.

DAFTAR PUSTAKA

- [1] Turban, Efraim, Ramesh Sharda, Dursun Delen, and David King. *Business Intelligence - A Managerial Approach*. Vol. 2. Prentice Hall, 2010.
- [2] Admin CPSSoft. *Why CPSSoft? - CPSSoft*. September 2014. <http://cpssoft.com/about> (accessed Mei 16, 2015).
- [3] Stackowiak, Robert, Joseph Rayman, and Rick Greenwald. *Oracle Data Warehousing and Business Intelligence Solutions*. Indianapolis: Wiley Publishing, Inc, 2007.
- [4] Inmon, W.H. "Building the Data Warehouse." (Wiley Computer Publishing) 3 (2002).
- [5] Colbert, John. *Captain Jack and the BPM Market: Performance Management in Turbulent Times | Business Performance Management content from Business Finance*. Juni 1, 2009. <http://businessfinancemag.com/business-performance-management/captain-jack-and-bpm-market-performance-management-turbulent-times> (accessed Mei 13, 2014).
- [6] Sharma, Neeraj, et al. *Database Fundamentals*. Markham, ON: IBM Corporation, 2010.
- [7] Elmasri, Ramez, and Shamkant B Navathe. *Fundamentals of Database System*. 3rd. Boston, Massachusetts: Addison-Wesley, 2000.
- [8] Mehra, V. "Building a Metadata-Driven Enterprise: A Holistic Approach." *Business Intelligence Journal* 10, no. 3 (Summer 2005).
- [9] Admin, Firebird. *Firebird: About Firebird*. n.d. <http://www.firebirdsql.org/en/about-firebird/> (accessed April 21, 2015).
- [10] Fowler, Martin. *UML Distilled - A Brief Guide to the Standard Object Modeling Language*. 3rd. Boston: Perason Education, Inc., 2004.
- [11] Hunter, David, et al. *Begining XML*. 4th. Indianapolis, Indiana: Wiley Publishing, Inc, 2007.