

Implementasi Aplikasi Report Menggunakan .Net Framework Studi Kasus PT Debesta Transport

Arfan sansprayada¹, Riva Abdillah Aziz², Edhi Prayitno³, Adi Chandra Setiawan⁴

Universitas Bina Sarana Informatika¹, STMIK Nusamandiri Jakarta², STMIK Nusamandiri Jakarta³, STMIK Nusamandiri Jakarta⁴,
arfan.anp@bsi.ac.id¹, riva.raz@nusamandiri.ac.id², edhi.epo@nusamandiri.ac.id³,
adi.dhe@nusamandiri.ac.id⁴

Abstract - This journal is explained about report application implementation that created with .Net technology at PT. DebestaTransport. PT. DebestaTransport is company that have business trucking service delivery container box both small and large. PT. DebestaTransport still create report with manual for management report, create report with manual make PT. DebestaTransport have trouble if want to create report with quickly and accurate. the method this research is 1. data collection the following: observation, interview, and literature review. 2. design application, 3. application development. the result this research is with uses report application that created with .Net technology, management PT. DebestaTransport can get report quickly and accurate.

Keywords: Logistic, System, Trucking.

Abstrak - Jurnal ini menjelaskan tentang implementasi aplikasi laporan yang dibuat dengan teknologi .Net di PT. Debesta Transport. PT. DebestaTransport adalah perusahaan yang memiliki bisnis jasa pengiriman peti kontainer baik kecil maupun besar. PT. DebestaTransport masih membuat laporan dengan manual untuk laporan manajemen, pembuatan laporan secara manual membuat PT. Debesta Transport mengalami kesulitan jika ingin membuat laporan dengan cepat dan akurat. Metode penelitian ini adalah 1. pengumpulan data sebagai berikut: observasi, wawancara, dan tinjauan literatur. 2. desain aplikasi, 3. pengembangan aplikasi. hasil penelitian ini adalah dengan menggunakan aplikasi laporan yang dibuat dengan teknologi .Net, manajemen PT. DebestaTransport bisa mendapatkan laporan dengan cepat dan akurat.

Kata Kunci : Logistik, Sistem, Trucking.

1. Pendahuluan

1.1 Latar Belakang Masalah

Sistem informasi dan teknologi komputer berkembang sangat dengan pesat sejalan dengan besarnya kebutuhan terhadap informasi.

Perkembangan teknologi informasi tidak lepas dari pesatnya perkembangan teknologi komputer, karena komputer merupakan media yang dapat memberikan kemudahan bagi manusia dalam menyelesaikan suatu pekerjaan. Perubahan dan dinamika masyarakat yang semakin cepat seiring dengan perkembangan jaman dan teknologi sehingga memerlukan kualitas informasi yang akurat cepat dan tepat teknologi informasi adalah salah satu contoh produk teknologi yang berkembang pesat yang dapat membantu manusia dalam mengolah data serta menyajikan sebuah informasi yang berkualitas. Untuk menyediakan informasi tersebut diperlukan suatu alat bantu atau media

untuk mengolah beraneka ragam data agar dapat disajikan menjadi sebuah informasi yang bermanfaat dengan kemasan yang menarik dan berpedoman pada criteria informasi yang berkualitas. Kemajuan teknologi yang berbasis komputer ini dimanfaatkan oleh perusahaan dengan menjadikan komputer sebagai alat untuk mengolah data menjadi suatu informasi yang cepat dan akurat. Perusahaan yang dapat lebih meningkatkan efektifitas dan efisiensi dalam menyelesaikan suatu pekerjaan. Distribusi barang merupakan kegiatan utama pada setiap perusahaan dalam memasarkan produknya.

PT Debesta Transport merupakan salah satu perusahaan yang bergerak dibidang jasa *trucking*, yaitu jasa pengiriman barang berupa peti kemas/container pada perusahaan yang membutuhkan untuk dikirim ke suatu tempat. Rute yang dikirim pun luas dari sabang sampai merauke. Perusahaan yang terletak di JL. K.H

Zainul arifin ini mempunyai kurang lebih 40 mobil *trucking* untuk pengangkutan peti kemas / container berukuran 20" atau 40".

1.2 Latar Belakang Masalah

Latar belakang masalah sebagai berikut

1. Masih manualnya sistem yang ada membuat proses menjadi terlalu rumit dan susah untuk dikerjakan.
2. Manajemen masih mengalami kesulitan dalam mendapatkan laporan kinerja perusahaan baik mengenai proses bisnis yang berjalan terlebih untuk laporan keuangan dengan cepat dan akurat.

1.3 Perumusan Masalah

Perumusan masalah sebagai berikut

1. Desain aplikasi mana yang cocok untuk menggantikan cara-cara manual yang saat ini digunakan?
2. Bagaimana dapat menyajikan laporan yang dibutuhkan oleh manajemen dengan cepat dan akurat?

1.4 Maksud dan Tujuan

Memperudahkan dari sisi karyawan untuk melakukan semua proses yang ada sehingga dengan adanya implementasi sistem ini mengintegrasikan seluruh divisi yang ada terhadap satu sistem. Hasil penelitian ini diharapkan dapat mengungkapkan berbagai permasalahan yang berkaitan dengan efektifitas kerja terhadap karyawan diantaranya adalah :

1. Menganalisa kebutuhan system yang dibutuhkan karyawan dari sisi *sales*, *finance* ataupun *management*.
2. Menganalisa seberapa besar sistem ini dapat memenuhi kebutuhan karyawan dan perusahaan mulai dari proses *job order* sampai ke proses laporan.

1.5 Metode Penelitian

1.5.1 Teknik Pengumpulan Data

Pengumpulan data dalam penelitian ini adalah sebagai berikut :

- a. Observasi
Metode observasi yaitu penulis melakukan pengamatan langsung turun kelapangan pada pegawai atau *management* PT Debesta Transport
- b. Wawancara
Melakukan wawancara langsung kepada salah satu pada pegawai PT Debesta Transport.
- c. Studi pustaka
Metode kepustakaan yaitu dengan mencari dan mempelajari buku buku atau jurnal yang relevan guna memberi pemahaman lebih baik

terhadap topik penulisan dan memperkaya pengetahuan tentang penulis tentang penelitian ilmiah.

1.5.2 Perancangan System

1.5.3 Pembuatan Aplikasi

1.6 Ruang Lingkup

Topik pembahasan yang terdapat pada penulisan ini adalah pembatasan pada penginputan dari sisi Sales mulai dari *quotation*, *job order*, surat ijin operasi (SIO), SIO *realisasi*, SIO biaya lain, *job order adjustment*, *customer invoice*. Sedangkan dari sisi divisi finance terdiri dari *customer payment*, bank *payment*, dan komisi supir. Sedangkan dari sisi laporan terdiri dari laporan *job order*, laporan SIO, laporan SIO *realisasi*, laporan SIO biaya lain, laporan *job order adjustment*, laporan *customer invoice*, laporan *customer invoice profit loss* dan laporan SIO komisi *customer*.

2. Bahan dan Metode

2.1 Konsep Dasar Model Pengembangan Sistem

Sistem adalah sekumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu (Jogiyanto, 2005:3). Sistem mempunyai sifat-sifat tertentu, yaitu:

1. *Components* adalah suatu sistem terdiri dari sejumlah komponen yang berinteraksi membentuk suatu kesatuan yang dapat berupa sub sistem atau bagian-bagian dari sistem.
2. *Boundary* adalah daerah yang membatasi antara satu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.
3. *Environments* adalah apapun yang berada diluar batas dari sistem yang mempengaruhi operasi sistem.
4. *Inteface* adalah media penghubung antara satu sub sistem dengan sub sistem lainnya.
5. *Input* adalah energi yang dimasukkan kedalam sistem, input dapat berupa *maintenance input* dan *signal input*.
6. *Output* adalah hasil energi yang telah diolah dan diklarifikasikan menjadi pengeluaran yang berguna dan sisa pembuangan.
7. *Process* adalah bagian yang mengolah masukan menjadi keluaran.
8. *Objectives* atau *Goal* sangat dibutuhkan oleh sistem untuk menentukan masuk yang dibutuhkan oleh sistem.

Sistem dapat diklarifikasikan dari berbagai sudut pandang, yaitu:

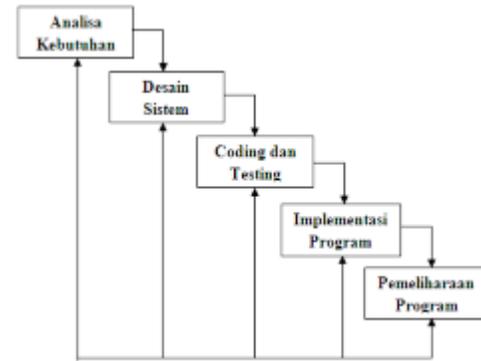
- a. *Abstract system* dan *Physical system*. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya teologika yaitu sistem yang berupa pemikiran-pemikiran hubungan manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi dan lain sebagainya.
- b. *Natural system* dan *Human made system*. Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat oleh manusia, misalnya rotasi perputaran bumi dan musim. Sistem buatan manusia adalah sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dan mesin, misalnya sistem informasi.
- c. *Deterministic system* dan *probabilistic system*. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi karena interaksi diantara bagian-bagiannya sudah dapat di deteksi dengan pasti. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Closed system dan *Open system*.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja otomatis tanpa adanya turut campur dari pihak luar. Secara teoritis sistem tertutup ini ada tetapi kenyataannya tidak ada sistem yang benar benar tertutup yang ada hanyalah *relatively closed system*. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atas sub sistem yang lainnya.

2.2 Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*Sequential linier*) atau alur hidup klasik (*classic life cycle*). Menurut M. Salahuddin (2011:26) "model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial akan terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*)". Berikut adalah gambar model *waterfall*:



Gambar 1. Waterfall

Sumber : Presman (2012)

Berikut adalah penjelasan tahapan-tahapan yang ada dalam model *waterfall*:

1. **Analisa Kebutuhan**
Merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara, atau *study* literatur. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem.
2. **Desain Sistem**
Proses desain ini menerjemahkan syarat kebutuhan dalam sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada: struktur data, arsitektur perangkat lunak, representasi *interface* dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen *software requirement*.
3. **Coding dan Testing**
Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh *user*. *Testing* adalah proses untuk mengidentifikasi kesalahan-kesalahan terhadap sistem tersebut.
4. **Penerapan Program**
Penerapan Program bisa diartikan sebagai *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah selesai akan digunakan oleh *user*.
5. **Pemeliharaan Program**
Perangkat lunak yang sudah disampaikan kepada pelanggan jika mengalami perubahan, perubahan tersebut bisa dikarenakan mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru), atau

karena pelanggan membutuhkan perkembangan sistem yang fungsional.

2.3 Unified Modeling Language (UML)

Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek'. Pemodelan (Modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML merupakan bahasa pemodelan dimana unsur-unsur dan aturan-aturan yang dimilikinya berfokus pada presentasi konseptual dan fisikal dari sistem. Unsur-unsur dan aturan-aturan tersebut dapat digunakan untuk merancang dan membaca model objek.

Kegunaan UML adalah sebagai berikut:

- a. UML sebagai bahasa visualisasi digunakan untuk merancang suatu model yang dapat dibaca oleh banyak orang dengan pengertian yang sama.
- b. UML bahasa pendefinisian digunakan untuk mendefinisikan dengan rinci seluruh hasil analisis, desain dan implementasi yang harus dilakukan dalam pengembangan sistem.
- c. UML sebagai bahasa dokumentasi digunakan untuk mendokumentasikan arsitektur beserta perinciannya, unsur-unsur yang dibutuhkan dalam pengembangannya. Serta perencanaan dan implementasi proyek secara keseluruhan dengan simbol-simbol yang mudah dimengerti.

1. Use Case Diagram

Deskripsi fungsi dari sebuah system dari perspektif pengguna. *Use Case* merupakan suatu urutan langkah-langkah yang menerangkan antara pengguna dan system yang saling terkait (**scenario**), baik termotivasi maupun secara manual, untuk tujuan umum pengguna.

Notasi yang digunakan dalam *use case diagram* adalah:

a. Actor

Sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan system (Munawar, 2005:63). *Actor* juga merupakan sesuatu yang memerlukan interaksi system untuk bertukar informasi.

b. Use case-symbol

Menggambarkan sekumpulan urutan, dimana setiap urutan mewakili intervensi antara *actor* dengan sistem. *Use case symbol* mempresentasikan fungsionalitas sistem secara keseluruhan.

2. Activity Diagram

Menurut Munawar (2005:109) "*Activity Diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus". Dimana secara grafis diagram ini menggambarkan aliran sekuntial dari kegiatan baik itu proses bisnis atau sebuah *use-case*. Diagram ini mempunyai peran seperti dengan *flowchart*, tetapi perbedaannya dengan *flowchart* adalah dimana diagram ini menggambarkan mekanisme kegiatan secara paralel.

Activity diagram sangat berguna untuk memodelkan kegiatan, menunjukkan tahapan, pengambilan keputusan dan percabangan yang akan dilakukan saat sebuah operasi dieksekusi dan untuk memodelkan hasil dari kegiatan tersebut.

Dibawah ini merupakan simbol-simbol yang ada pada diagram aktivitas:

a. Initial Node atau Start Point

Initial node yang digambarkan dengan simbol lingkaran padat, merupakan titik yang mengawali *activity diagram*.

b. Activity

Menggambarkan suatu kegiatan yang perlu dilakukan. Suatu *activity*. Suatu *activity* bisa berbentuk secara fisik, seperti *inspect forms*, atau secara elektronik, seperti *display create student screen*.

c. Control Flow

Panah yang menggambarkan sasaran yang mengawali kegiatan.

d. Fork (Percabangan)

Bar hitam dengan satu *flow* yang mengarah ke dalamnya dan beberapa *flow* lainnya meninggalkannya, merupakan bar sinkronisasi dimana kegiatan dapat dilakukan secara paralel.

e. Join (Penggabungan)

Bar hitam dengan beberapa *flow* mengarah ke dalamnya dan ada satu *flow* yang meninggalkannya, merupakan bar sinkronisasi dimana beberapa aktivitas yang mempunyai tujuan yang sama dan tujuan keduanya digabungkan menjadi satu tujuan.

f. Decision

Menggambarkan sebuah kegiatan untuk pengambilan keputusan.

g. Final Node atau End Point

Menggambarkan sebuah titik akhir dari sebuah *activity diagram*.

3. Sequence Diagram

Menurut Munawar (2005:87) "*Sequence Diagram* digunakan untuk

menggambarkan perilaku pada sebuah scenario". Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

Komponen utama pada *sequence diagram* terdiri dari:

- a. Obyek atau Participant
Obyek yang digambarkan dengan kotak segiempat bernama. Obyek diletakkan didekat bagian atas diagram dengan urutan dari kiri ke kanan. Setiap participant terhubung dengan garis titik-titik yang disebut *lifeline*.
- b. *Message* (Pesan)
Disimbolkan dengan sebuah garis dengan tanda panah. Sebuah *message* bergerak dari satu participant ke participant yang lain. Sebuah participant dapat mengirim sebuah *message* kepada dirinya sendiri.
- c. *Time*
Time adalah diagram yang mewakili waktu pada arah *vertical* (Munawar, 2005:88). Waktu dimulai dari atas kebawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dari pada *message* yang lebih dekat mengarah kebawah.

4. *Class Diagram*

Menurut Henderi (2008:6) "*class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML". *Class diagram* adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem.

Class diagram terdapat beberapa pemodelan UML terdiri dari: *Class-class*, struktur *class*, sifat *class* (*class behavior*), perkumpulan atau gabungan (*association*), pengumpulan atau kesatuan (*agregation*), ketergantungan (*depedency*), relasi-relasi turunannya, keberagaman dan indikator navigasi dan *role name* (peranan/tugas nama).

2.4 Entitiy Relationship Diagram (ERD)

Menurut "*Entitiy Relationship Model/ER_M* Merupakan suatu model data yang dikembangkan berdasarkan obyek". ER_M digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logik. ER_M digambarkan dalam bentuk diagram yang disebut

diagram ER (*ER Diagram/ER_D*). Elemen-elemen diagram ERD antara lain:

1. *Entity* (Entitas)
Menurut Fathansyah (2007:73) "Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain". Entitas terdapat nama dan kata benda yang dapat dikelompokan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).
2. *Relationship* (Relasi)
Menurut Fathansyah (2007:75) "Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda". *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat.
Pada umumnya *Relationship* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif).
3. Derajat *Relationship* (*Relationship Degree*)
Relationship adalah jumlah entitas yang berpartisipasi dalam satu *relationship*. Derajat *relationship* yang sering dipakai dalam ERD adalah:
 - a. *Unary Relationship*
Unary Relationship adalah model *relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *Recursive Relationship* dan *Reflective Relationship*.
 - b. *Binary Relationship*
Binary Relationship adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.
 - c. *Ternary Relationship*
Ternary Relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sehipak.
4. Atribut
Menurut Fathansyah (2007:74) "Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut. Secara umum atribut adalah sifat atau karakteristik dari setiap entitas maupun setiap *relationship* maksudnya atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship* sehingga sering dikatakan atribut adalah elemen dari setiap entitas atau *relationship*."

5. Kardinalitas (*Cardinality*)

Menurut Fathansyah (2007:77) “Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”.

Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

a. *One to One* (Satu ke Satu)

Yang berarti setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas yang kedua, dan begitu juga sebaliknya setiap entitas pada himpunan entitas yang kedua berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas pertama.

b. *One to Many* (Satu ke Banyak)

Yang berarti setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas yang kedua berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas pertama.

c. *Many to One* (Banyak ke Satu)

Yang berarti setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas yang kedua, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas yang kedua.

d. *Many to Many* (Banyak ke Banyak)

Yang berarti setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, dan demikian juga sebaliknya dimana setiap entitas pada himpunan entitas yang kedua dapat berhubungan dengan banyak entitas pada himpunan entitas yang pertama.

6. *Participation Constraint*

Participation Constraint merupakan batasan yang menjelaskan apakah keberadaan suatu *entity* terhadap pada hubungannya dengan *entity* lain. Terdapat dua macam *Participation Constraint*:

a. *Total Participation*

Keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain.

b. *Partial Participation*

Keberadaan suatu *entity* tidak tergantung pada hubungannya dengan *entity* lain.

Ketentuan-ketentuan dalam pembuatan ERD adalah:

1. Menentukan entitas menentukan peran, kejadian, lokasi, hal nyata dan konsep dimana penggunaan untuk menyimpan data.
2. Menentukan relasi menentukan hubungan antar pasangan entitas menggunakan matriks relasi.
3. Gambar ERD sementara entitas digambarkan dengan kotak, dan relasi digambarkan dengan garis.
4. Isi kardinalitas menentukan jumlah kejadian satu entitas untuk sebuah kejadian pada entitas yang berhubungan.
5. Tentukan kunci utama menentukan atribut yang mengidentifikasi satu dan hanya satu kejadian masing-masing entitas.
6. Gambar ERD berdasarkan kunci menghilangkan relasi *many to many* dan memasukan *primary* dan kunci tamu pada masing-masing entitas.
7. Menentukan atribut menentukan *field-field* yang diperlukan sistem.
8. Pemetaan atribut dengan cara memasang atribut dengan entitas yang sesuai.
9. Gambar ERD dengan atribut mengatur ERD dari langkah No. 6 dengan menambahkan entitas atau relasi yang ditemukan pada langkah No. 8.
10. Periksa hasil, apakah ERD sudah menggambarkan sistem yang akan dibangun atau dirancang.

3. Analisa Sistem Berjalan dan Program Usulan

3.1 Tinjauan Perusahaan

PT Debesta Transport merupakan perusahaan truck container yang berdiri dari tahun 2000 dimana dalam moto perusahaan mendedikasikan untuk memenuhi pelanggan secara baik dan melayani kepuasan pelanggan. PT Debesta Transport melayani semua terminal yang ada dipelabuhan Jakarta dan sekitarnya. Untuk masalah safety dan keamanan mobil truck yang dimiliki PT Debesta Transport dilengkapi dengan keamanan *GPS (Global Positioning System)* dimana terdeteksi selama 24 jam dan dapat melacak keberadaan posisi truck secara baik. Dengan berkembangnya dan banyaknya *customer* yang menggunakan jasa truck pada PT

Debesta Transport tersebut mengalami kendala terhadap kinerja perusahaan dan karyawan dalam implementasi sistem yang masih bersifat manual.

Dibutuhkan rasa kepercayaan yang sangat besar bagi *customer* untuk perusahaan karena begitu tidak mengenalnya mereka kepada perusahaan dan yang sangat terpenting dikarenakan jarak antar Negara yang sangat jauh. Setiap *customer* pasti mempunyai rasa tidak percaya terhadap perasaan disaat mereka ingin pertama kali melakukan kerjasama terhadap perusahaan. Tetapi semua rasa tidak percaya akan adipertanggung jawabkan oleh perusahaan untuk kepuasan *customer* salah satunya dengan efektifitas karyawan terhadap kinerja karyawan. Implementasi yang akan diterapkan pada PT Debesta Transport menggunakan .net framework yang sangat user friendly artinya dari sisi karyawan sangat nyaman dan enak digunakan serta dari sisi management sangat mudah untuk digunakan menarik semua laporan bila diperlukan. Bila sistem ini berjalan dengan baik dan lancar tidak memungkinkan bahwa efektifitas kerja akan mengimbangi pelayanan yang baik terhadap *customer* pada PT Debestaransport.

3.2 Sejarah Perusahaan

PT Debesta Transport berdiri akhir tahun 2000 an di Jakarta. Perusahaan penyedia jasa truck ini menangani pengiriman peti kemas dengan ukuran 20" atau ukuran 40" dari seluruh Indonesia. PT Debesta Transport ini mempunyai garasi untuk menampung truck yang dimiliki didaerah Tanjung Priok. Dimana secara keseluruhan PT Debesta Transport mempunyai armada kurang lebih 40 mobil truck yang siap untuk mengantar peti kemas *customer* ke tujuan dengan keamanan GPS yang terpasang pada mobil truck tersebut sehingga memberikan rasa aman terhadap *customer* akan barang yang akan dikirim menggunakan jasa PT Debesta Transport.

Perusahaan yang hanya bergerak dibidang jasa ini konsisten untuk selalu memberikan pelayanan kepada *customer* atau pelanggan untuk memberikan jasa yang terbaik dalam melakukan pengiriman. *Customer* atau pelanggan disini bersifat *corporate*. Artinya hanya perusahaan yang melakukan pengiriman dalam partai besar dalam bentuk petikemas yang bisa menggunakan jasa PT Debesta Transport.

3.3 Proses Bisnis

Pada dasarnya PT Debesta Transport adalah perusahaan *logistic*. Akan tetapi perusahaan ini hanya menyediakan jasa truck nya saja untuk melayani permintaan *customer* atau pelanggan yang membutuhkan jasa pengiriman barang

dalam bentuk peti kemas kepolosok seluruh Indonesia.

Proses order dimulai dimana dalam divisi sales melakukan penginputan dari *quotation* yaitu nota kesepatakatan harga dan bentuk container yang akan digunakan. Kemudian setelah proses *quotation* selesai maka akan dijalankan nya *job order* yang diinput kedalam system tersebut. *Job order* ini bersifat sementara artinya tergantung dari perubahan biaya yang nanti bisa berubah sewaktu waktu. Setelah *job order* diproses maka masuk kedalam surat ijin operasi (SIO) disini informasi sangat dibutuhkan antara karyawan PT Debesta Transport artinya tidak boleh ada kesalahan antara karyawan yang ada di Head Office dengan karyawan yang ada di Garasi atau di Tanjung Priok. Setelah SIO selesai maka masuk ke pada SIO realisasi dan SIO biaya lain yaitu proses yang merealisasikan biaya SIO secara akurat dan benar tanpa merekayasa proses biaya yang ada. Terakhir adalah proses *job order customer invoice* yaitu proses secara keseluruhan atau finalisasi proses yang ada pada *quotation*.

Sedangkan untuk proses laporan, pihak management dapat melihat proses proses yang ada pada laporan . laporan terdiri dari laporan *job order*, laporan SIO, laporan SIO realisasi, laporan SIO biaya lain, laporan *job order adjustment*, laporan *customer invoice*, laporan *customer invoice profit loss* dan laporan SIO komisi *customer*.

Diharapkan dengan proses yang ada dan pengimplementasian sistem yang dilakukan mempermudah bagi perusahaan khususnya dari sisi management untuk mendapatkan laporan profit dan benefit secara real time sedangkan dari sisi karyawan mempermudah pekerjaan dan menghapus sistem yang masih bersifat manual.

3.4 User Interface

3.4.1 User Interface Login

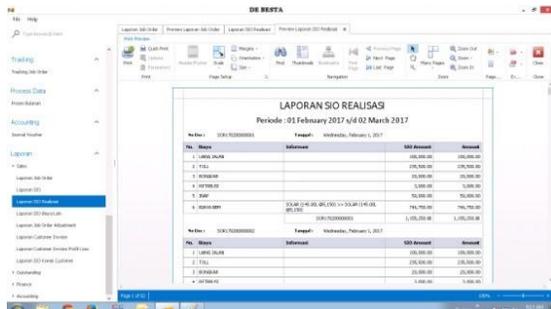
Berikut tampilan *login* pada *user interface* PT Debesta Transport



Gambar 2. User interface login

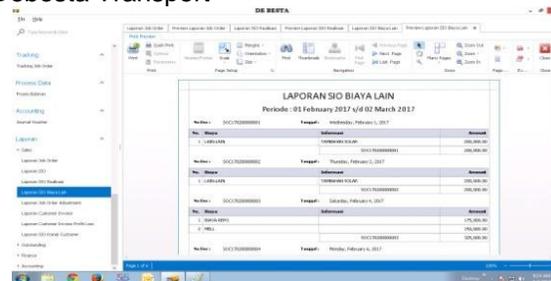
3.4.2 User Interface Laporan SIO Realisasi

Berikut tampilan laporan SIO Realisasi PT Debesta Transport



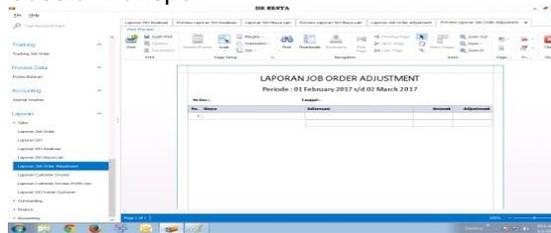
Gambar 3. User interface laporan SIO realisasi

3.4.3 User Interface Laporan SIO Biaya Lain
Berikut tampilan laporan SIO Biaya Lain PT Debesta Transport



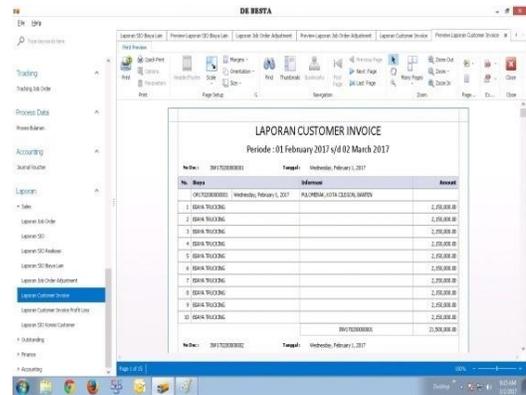
Gambar 4. User interface laporan SIO Biaya Lain

3.4.4 User Interface Laporan Job Order Adjusment
Berikut tampilan laporan Job Order Adjusment PT Debesta Transport



Gambar 5. User interface laporan Job Order Adjusment

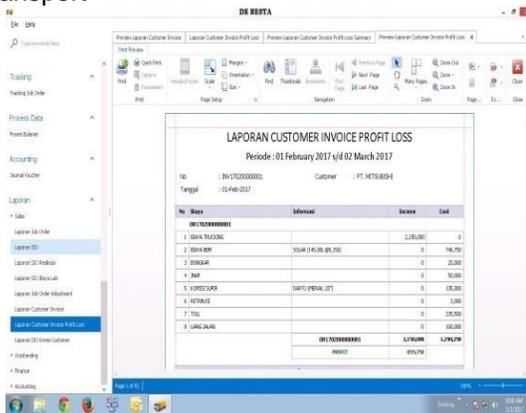
3.4.5 User Interface Laporan Customer Invoice
Berikut tampilan laporan customer payment PT Debesta Transport



Gambar 6. User interface laporan Customer Invoice

3.4.6 User Interface Laporan Invoice Profit Loss

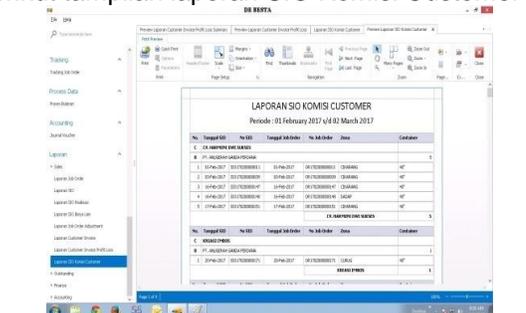
Berikut tampilan laporan invoice profit PT Debesta Transport



Gambar 7. User interface Laporan Invoice Profit Loss

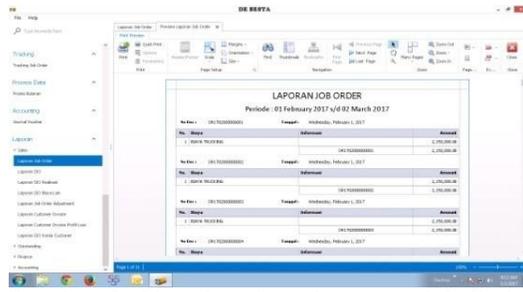
3.4.7 User Interface Laporan SIO Komisi Customer

Berikut tampilan laporan SIO Komisi Customer



Gambar 8. User interface Laporan SIO Komisi Customer

3.4.8 User Interface Laporan Job Order
Berikut tampilan laporan Job Order PT Debesta Transport



Gambar 9. User interface Laporan Job Order

4. Kesimpulan dan Saran

4.1 Kesimpulan

Kesimpulan yang didapat dari hasil implementasi aplikasi Report menggunakan *.net Framework* pada PT Debesta Transport adalah sebagai berikut :

1. Implementasi aplikasi Report menggunakan *.Net framework* ini dibangun untuk mendukung proses bisnis yang ada diperusahaan tersebut sehingga memudahkan dari sisi karyawan dan management perusahaan agar dapat meningkatkan efisiensi kerja dengan baik dan benar
2. Fitur fitur yang ada pada implementasi *.net framework* ini meliputi dari *fitur sales order*, *finance*, serta fitur laporan.

Implementasi pada PT Debesta Transport ini dibangun unukt memberikan kemudahan dari sisi karyawan dan memberikan laporan yang valid untuk management perusahaan.

4.2 Saran

Pengembangan implementasi aplikasi Report Menggunakan *.net framework* pada PT Debesta Transport ini masih banyak kekurangan serta perlu dikembangkan lagi. Dan untuk itu beberapa saran penulis antar lain :

1. Hendaknya penelitian ini dapat digunakan sebagai acuan penelitian berikutnya dengan mengaitkan aspek-aspek yang belum diungkapkan dan dikembangkan.
2. Penerapan implementasi pada PT Debesta Transport ini memberikan keleluasaan kepada karyawan dan management untuk meningkatkan efisiensi kerja serta memberikan laporan yang valid terhadap perusahaan

6. Pustaka

- [1] Jogiyanto. 2005. Analisis dan Desain Sistem Informasi. Yogyakarta: Penerbit Andi.
- [2] Shalahuddin. 2011. Analisis dan Desain Basis Data. Bandung:Modula.
- [3] Presman. 2005. Rekayasa Perangkat Lunak. Yogyakarta:Penerbit Andi.
- [4] Nugroho. 2010. Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP. Yogyakarta: Andi.
- [5] Munawar. 2005. Pemodelan Visual dengan UML. Yogyakarta: Graha Ilmu.
- [6] Fathansyah. 2002. Basis Data. Bandung:Graha Ilmu.