

# Implementasi Aplikasi Import Menggunakan Bootstrap Studi Kasus PT Anugerah Ganda Perdana

Arfan Sansprayada

*Abstract— This writing implement on Bootstrap application with case studies PT Anugerah Ganda Perdana a company engaged in the trucking services, which are companies that take care of the delivery of goods in the form of container/container either in bulk or small parties. This application describes the details about the flow of information about the flow of information about the way of the commencement order will begin the process to finish. Trucking services are needed at every company where the freight forwarder is business and growing even though it appears simple but who would have thought to have tremendous market potential. Benefit applications to be built is easier for the employee to perform the work in accordance with their respective division. While benefiting from the management is able to see the companys profit and loss in detail and clearly. PT Anugerah Ganda Perdana is one of the trucking service company that has a customer or customers very much which serve the needs of customers will both freight export and import. Implementation using the Bootstrap is an application that is used in PT Anugerah Ganda Perdana for supporting either of the employee or the management company in detail and complete so as to provide convenience for all those involved in PT Anugerah Ganda Perdana*

**Intisari—** Penulisan ini mengimplementasikan aplikasi *trucking* menggunakan *bootstrap* dengan studi kasus PT Anugerah Ganda Perdana sebuah perusahaan yang bergerak dibidang jasa *trucking*, yang mengurus pengiriman barang pada *container* atau petikemas. Aplikasi ini menggambarkan detail tentang alur informasi tata cara dimulainya order sampai selesai. Jasa *trucking* sangat dibutuhkan dimana jasa pengiriman barang merupakan bisnis yang semakin maju dan berkembang walau terlihat sederhana tapi mempunyai potensi pasar yang luar biasa. Manfaat aplikasi yang akan dibangun ini adalah mempermudah karyawan untuk melakukan pekerjaannya sesuai dengan divisi masing masing, sedangkan manfaat dari sisi *management* adalah dapat melihat *profit* dan *loss* perusahaan secara detail dan jelas. Implementasi menggunakan *Bootstrap* ini merupakan aplikasi yang digunakan pada PT Anugerah Ganda Perdana untuk mendukung sarana dari sisi karyawan ataupun *management* perusahaan secara detail dan lengkap.

**Keywords :** *Logistic, System, Import.*

## I. PENDAHULUAN

### A. 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 kriteria 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 Anugerah Ganda Perdana 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”.

### B. Identifikasi Permasalahan

1. Masih manualnya sistem yang ada membuat proses menjadi terlalu rumit dan susah untuk dikerjakan.
2. Keinginan management perusahaan untuk mengetahui proses secara detail dan lengkap baik dari segi laporan ataupun dari segi keuntungan perusahaan.

### C. Perumusan Masalah

Membutuhkan implementasi dari sisi divisi import untuk membuat master kurs dan membuat arrival notice dan invoice dalam divisi import

### D. Maksud dan Tujuan

Mempermudah 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

*Jurusan Manajemen Informatika, AMIK BSI Bogor, Jl Merdeka No 168 Ciwaringin Bogor (telp: 0251-835 3009; fax: 0274-4321982; e-mail: arfan.anp@bsi.ac.id )*

diharapkan dapat mengungkapkan berbagai permasalahan yang berkaitan dengan efektifitas kerja terhadap karyawan.

#### E. Metode Penelitian

##### 1. Teknik Pengumpulan Data

Dalam penelitian ini penulis menggunakan metode penelitian sebagai berikut :

###### a. Observasi

Metode observasi yaitu penulis melakukan pengamatan langsung turun kelapangan pada pegawai atau *management* PT Anugerah Ganda Perdana

###### b. Wawancara

Melakukan wawancara langsung kepada salah satu pada pegawai PT Anugerah Ganda Perdana.

###### 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

#### F. Ruang Lingkup

Topik pembahasan yang terdapat pada penulisan ini adalah pembatasan hanya dari sisi pembuatan master kurs dan pembuatan Arrival notice dan Invoice pada divisi Import.

## II. BAHAN DAN METODE

### 1. Konsep Dasar Model Pengembangan Sistem

Sistem adalah sekumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu [1]. Sistem mempunyai sifat-sifat tertentu, yaitu:

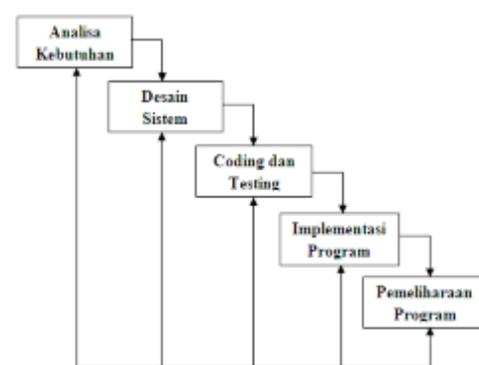
- Components* adalah suatu sistem terdiri dari sejumlah komponen yang berinteraksi membentuk suatu kesatuan yang dapat berupa sub sistem atau bagian-bagian dari sistem.
- Boundary* adalah daerah yang membatasi antara satu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batas suatu sistem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

Sistem dapat diklarifikasikan dari berbagai sudut pandang, yaitu:

- 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.
- 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 anantara manusia dan mesin, misalnya sistem informasi.
- Deterministic sytem* dan *probabilistic system*. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi karena interkasi 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. Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*Sequential linier*) atau alur hidup klasik (*classic life cycle*). 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*. [2]



Sumber : [3]

Gbr 1 Model Waterfall

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

- a. 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.
- b. 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*.
- c. 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.
- d. 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*.
- e. 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.

### 3. Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ [3]. 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 fisikial dari sistem. Unsur-unsur dan aturan-aturan tersebut dapat digunakan untuk merancang dan membaca model objek.

Kegunaan UML adalah sebagai bahasa visualisasi digunakan untuk merancang suatu model yang dapat dibaca oleh banyak orang dengan pengertian yang sama. UML juga merupakan bahasa pendefinisian yang digunakan untuk mendefinisikan dengan rinci seluruh hasil analisis, desain dan implementasi yang harus dilakukan dalam pengembangan sistem. Selain itu UML juga digunakan

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.

#### a. 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:

##### 1) Actor

Sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan system [4]. *Actor* juga merupakan sesuatu yang memerlukan interaksi system untuk bertukar informasi.

##### 2) Use case-symbol

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

#### b. Activity Diagram

*Activity Diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus”[5]. Dimana secara grafis diagram ini menggambarkan aliran sekuensial 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:

##### 1) Initial Node atau Start Point

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

##### 2) 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*.

##### 3) Control Flow

Panah yang menggambarkan sasaran yang mengawali kegiatan.

##### 4) 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.

##### 5) 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.

6) *Desicion*

Menggambarkan sebuah kegiatan untuk pengambilan keputusan.

7) *Final Node* atau *End Point*

Menggambarkan sebuah titik akhir dari sebuah *activity diagram*.

c. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario [5]. 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:

1) 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*.

2) *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.

3) *Time*

*Time* adalah diagram yang mewakili waktu pada arah *vertical* [6]. Waktu dimulai dari atas kebawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dari pada *message* yang lebih dekat mengarah kebawah.

d. *Class Diagram*

*Class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML [7]. *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).

4. *Entitiy Relationship Diagram* (ERD)

"*Entitiy Relationship Model/ER\_M* Merupakan suatu model data yang dikembangkan berdasarkan obyek"[6]. *ER\_M* digunakan untuk mejelaskan 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:

a. *Entity* (Entitas)

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain[8]. Entitas terdapat nama dan kata benda yang

dapat dikelompokan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

b. *Relationship* (Relasi)

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda [8]. *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).

c. Derajat *Relationship* (*Relationship Degree*)

*Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*. Derajat *relationship* yang sering dipakai dalam ERD adalah:

1) *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*.

2) *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.

3) *Ternary Relationship*

*Ternary Relationship* merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sehipak.

d. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut[9]. 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*.

e. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain [10]. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

1) *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.

2) *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.

3) *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.

4) *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.

f. *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* yang 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 memasangkan 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.

### III. ANALISA SISTEM DAN PROGRAM USULAN

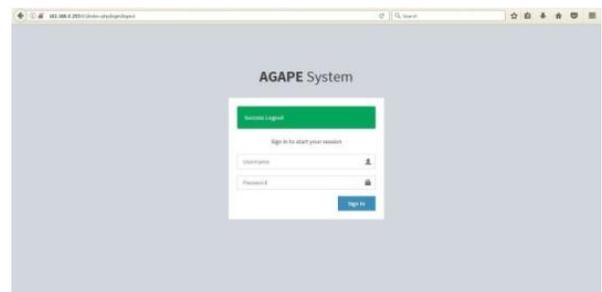
#### A. Proses Bisnis

Pada dasarnya PT Anugerah Ganda Perdana perusahaan *logistic*. Akan tetapi perusahaan ini hanya menyediakan jasa import juga dari sisi dokumen ataupun dalam proses pengiriman import terhadap pelayaran. Proses yang dibutuhkan hanya sekedar dalam pembuatan arrival notice dan invoice dan pembuatan master kurs untuk divisi import. Sehingga memudahkan dari divisi import dalam melakukan aktivitas karyawan. 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

#### B. User interface

##### 1. User interface Login

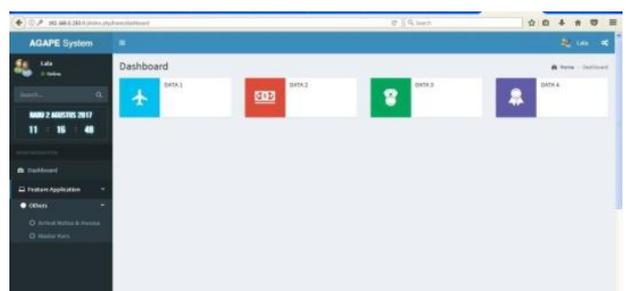
Berikut tampilan *login* pada *user interface* PT Anugerah Ganda Perdana



Gbr 2. User Interface Login

##### 2. User interface Menu

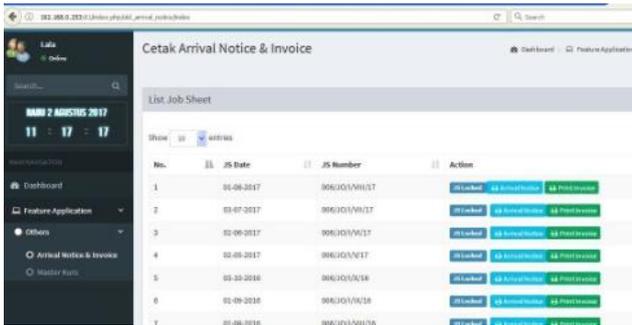
Berikut tampilan *Quotation* pada *user interface* PT. Anugerah Ganda Perdana



Gbr 3. User Interface Menu

##### 3. User interface Arrival Notice

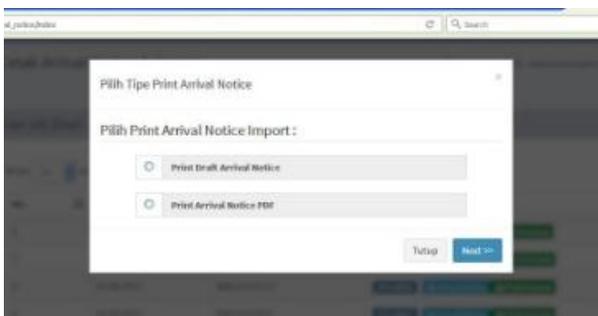
Berikut tampilan *Arrival Notice* pada *user interface* PT Anugerah Ganda Perdana



Gbr 4. User Interface Arrival Notice

4. User interface Print Arrival Notice

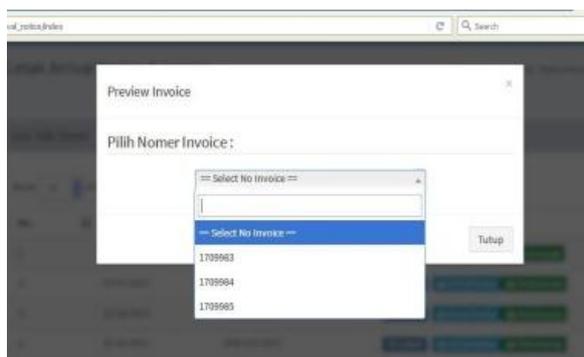
Berikut tampilan *Print Arrival Notice* pada user interface PT Anugerah Ganda Perdana



Gbr 5 User Interface Print Arrival Notice

5. User interface Print Invoice

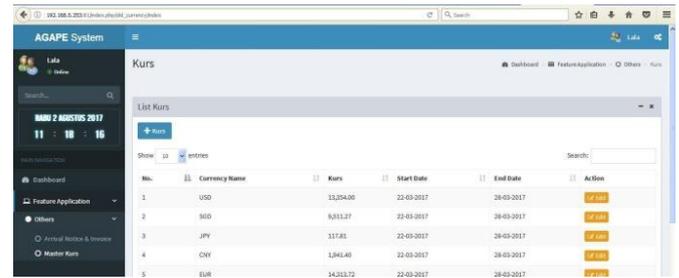
Berikut tampilan *Print Invoice* pada user interface PT Anugerah Ganda Perdana.



Gbr 6. User Interface Print Invoice

6. User interface Master Kurs

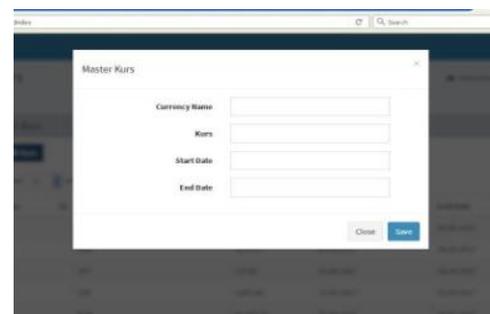
Berikut tampilan Master Kurs pada user interface PT Anugerah Ganda Perdana



Gbr 7. User Interface Master Kurs

7. User Interface Add Master Kurs

Berikut tampilan *Add Master Kurs* pada user interface PT Anugerah Ganda Perdana



Gbr 8 User Interface Add Master Kurs

IV. KESIMPULAN DAN SARAN

Kesimpulan yang didapat dari hasil implementasi aplikasi menggunakan *Bootstrap* pada PT Anugerah Ganda Perdana adalah sebagai berikut :

1. Implementasi aplikasi menggunakan *Bootstrap* ini dibangun untuk mendukung proses bisnis yang ada di perusahaan 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 *Bootstrap* ini meliputi Master Kurs dan penambahan invoice serta arrival notice.
3. Implementasi pada PT Anugerah Ganda Perdana ini dibangun untuk memberikan kemudahan dari sisi karyawan dan memberikan laporan yang valid untuk management perusahaan.

Pengembangan implementasi aplikasi *Bootstrap* pada PT Anugerah Ganda Perdana 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 Anugerah Ganda Perdana ini memberikan keleluasaan kepada karyawan dan management untuk meningkatkan efisiensi kerja serta memberikan laporan yang valid terhadap perusahaan.

## UCAPAN TERIMA KASIH

Sebagai penutup penulis mengucapkan banyak terima kasih kepada ALLAH SWT. Yang telah memberikan nikmat serta karunia didalam kehidupan dunia ini. Tak lupa Penulis mengucapkan kepada Ibu tercinta Hj. Sumiati yang telah rela berjuang melahirkan penulis hingga penulis sampai sekarang bisa seperti ini. I luv u Mom. Kepada istri penulis, Eka Herawati serta anak tersayang Keenan Athala Sansprayada yang telah mendukung segala aktifitas penulis selama ini. I luv u bunda dan athala. Penulis juga mengucapkan terima kasih kepada Bapak yansen wijaya dimana selaku direktur dari PT Anugerah Ganda Perdana yang telah memberikan penulis riset di perusahaan tersebut. Serta pihak lain yang tidak bisa penulis sebutkan satu persatu.

## RERENSI

- [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



Arfan sansprayada. Lahir di Jakarta 27 Oktober 1984. Lulus Pasca Sarjana (S2) Ilmu Komputer di STMIK Nusa Mandiri Jakarta, Saat ini aktif sebagai Dosen di AMIK BSI