

Aplikasi Optimalisasi Muat Barang Dengan Penerapan Algoritma Dynamic Programming Pada Persoalan Integer Knapsack

Daniel Jahja Surjawan^{#1}, Irene Susanto^{#2}

Jurusan SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri No. 65 Bandung

¹danieljahjas@outlook.com

²laurenciaren@hotmail.com

Abstract — In the era of globalization, the use of information technology become a necessity that can not be separated in running a business in all instances. The need of information technology is not only used as a supporter of the process, but also has become a major part of which should go well. It was felt also by the CV. Labatrans engaged in courier services, which often occurs excess or shortage even when the goods are loaded into a vehicle that will deliver. There has been no control of the goods to be delivered, so that sometimes exceed the limit load capacity of the vehicle. This cause a heap of goods in the warehouse, and the main problem consumers will be disappointed because the goods delivered are not accepted at the right promised time. Another problem occurs when consumers asked where the goods they are shipped, they want to know where their goods for some reason such as perishable goods or food. And usually the staff of the company can not give an answer because of limited information. The purpose of this research is to create an application that can handle tracking of goods delivered and the load optimization by applying Dynamic Programming Algorithms on the issue of Integer Knapsack.

Keywords — Shipping, Goods Load Optimization, Integer Knapsack, Track the existence of Goods

I. PENDAHULUAN

Pada era globalisasi saat ini, penggunaan teknologi informasi menjadi kebutuhan yang tidak dapat dipisahkan dalam menjalankan roda bisnis di semua instansi. Kebutuhan akan teknologi informasi yang dirasakan ini tidak hanya digunakan sebagai pendukung proses saja, namun sudah menjadi bagian utama yang harus berjalan dengan baik. Seperti halnya CV. Labatrans yang membutuhkan teknologi tersebut agar mampu terus bersaing dalam menjalankan proses bisnisnya di bidang pengiriman barang, serta dapat membantu meningkatkan kepercayaan pengguna jasa pengiriman barang.

CV. Labatrans merupakan salah satu perusahaan swasta yang berdiri sejak tahun 2009, bergerak di bidang jasa pengiriman/ekspedisi khususnya pengiriman barang melalui

darat, dimana area pengirimannya meliputi kota Bandung dan Pekalongan. Dalam menjalankan proses bisnisnya, perusahaan ini mengalami beberapa kendala yang seringkali malah dianggap menjadi hal yang biasa terjadi, namun ini berdampak pada kerugian perusahaan yang cukup besar. Salah satu kendala yang dihadapi oleh perusahaan yaitu kurangnya kontrol dalam melakukan muat barang. Setiap menerima barang dari konsumen, staf perusahaan hanya mencatat data pengirim dan penerima dari barang yang akan dikirim tanpa mengetahui kapasitas dari kendaraan yang akan mengirim, karena memang staf disini tidak bertanggung jawab atas hal tersebut. Masalah terjadi saat melakukan muat barang sebanyak mungkin ke dalam kendaraan hingga terkadang melebihi kapasitas muatan kendaraan. Apabila kendaraan sudah penuh, maka barang akan dikirim menggunakan kendaraan lain di hari berikutnya, hal ini menyebabkan penumpukan barang di gudang dan yang menjadi masalah utama konsumen akan kecewa karena barang yang dikirim tidak diterima tepat pada waktu yang dijanjikan. Masalah lain terjadi ketika konsumen menanyakan keberadaan barang yang mereka kirim, dimana mereka ingin mengetahui sudah atau sedang berada dimana barang mereka karena alasan tertentu seperti barang atau makanan mudah rusak. Dan biasanya staf dari perusahaan tidak dapat memberikan jawaban karena keterbatasan informasi yang dimiliki.

Oleh karena itu, CV. Labatrans memerlukan suatu sistem yang berguna untuk menghitung kapasitas muatan barang pada kendaraan yang akan mengirim untuk menghindari kerugian yang dapat berakibat fatal bagi perusahaan seperti rusaknya barang karena penumpukan barang di gudang dan komplain dari konsumen yang kecewa karena barang yang dikirim tidak diterima pada waktu yang dijanjikan. Selain itu akan dibuat sistem yang dapat memberikan informasi status proses pengiriman barang ketika konsumen ingin mengetahui keberadaan barangnya.

Harapan dari penelitian ini adalah untuk membantu

meningkatkan kinerja dan kualitas pelayanan perusahaan sehingga menciptakan kepercayaan dalam pelayanan jasa pengiriman yang diberikan CV. Labatrans.

II. KAJIAN TEORI

A. Optimasi

Optimasi merupakan bagian yang tidak terpisahkan dari rekayasa proses desain. Fokus dari optimasi adalah menemukan solusi yang optimal pada masalah desain melalui pertimbangan sistematis dari alternatif yang diberikan untuk kepuasan sumber daya dan kendala biaya. Banyak masalah teknik yang terbuka dan kompleks. Tujuan optimasi secara keseluruhan untuk meminimalkan biaya, memaksimalkan keuntungan, untuk merampingkan kegiatan produksi, meningkatkan efisiensi proses, dan lain-lain. Dalam menentukan solusi yang optimal diperlukan pertimbangan yang cermat dari beberapa alternatif yang sering dibandingkan pada beberapa kriteria [1].

B. Knapsack Problem

Knapsack merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak objek dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari n objek (1, 2, 3, ..., n) dimana setiap objek memiliki bobot (w_i) dan nilai profit (p_i) dengan memperhatikan juga kapasitas dari media penyimpanan sebesar (M).

Masalah *knapsack* (*knapsack problem*) merupakan sebuah persoalan yang menarik. Dalam dunia nyata permasalahan *knapsack* ini sering sekali digunakan pada bidang jasa pengangkutan barang seperti pengangkutan peti kemas dalam sebuah media pengangkut. Dalam usaha tersebut, diinginkan keuntungan yang maksimal untuk mengangkut barang yang ada dengan tidak melebihi kapasitas yang ada. Berdasarkan persoalan tersebut, diharapkan ada suatu solusi yang secara otomatis dapat mengatasi persoalan itu.

Knapsack sendiri terdiri dari beberapa persoalan, yaitu:

1. *Knapsack 0-1 (integer knapsack)*

Objek yang dimasukkan ke dalam media penyimpanan, dimana dimensinya harus dimasukkan semua atau tidak sama sekali.

2. *Bounded Knapsack*

Objek yang dimasukkan ke dalam media penyimpanan, dimana dimensinya bisa dimasukkan sebagian atau seluruhnya.

3. *Unbounded knapsack*

Jumlah objek yang dimasukkan ke dalam media penyimpanan yang macamnya tidak terbatas [2].

Pada penelitian ini menggunakan persoalan *Knapsack 0-1* atau *integer knapsack*. Pada persoalan *integer knapsack*, barang yang diangkut dimensinya harus diangkut seluruhnya atau tidak sama sekali. Permasalahan *integer*

knapsack mempunyai solusi persoalan yang dinyatakan sebagai himpunan:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

Yang dalam hal ini, $x_i = 1$ jika benda ke- i dimasukkan ke dalam media penyimpanan atau $x_i = 0$ jika benda ke- i tidak dimasukkan ke dalam media penyimpanan. Karena itulah persoalan ini dinamakan 0-1 *knapsack*. Sebagai contoh, $X = \{1, 0, 0, 1\}$ adalah sebuah solusi yang ditemukan, maka benda ke-1 dan ke-4 dimasukkan ke dalam media penyimpanan dan benda ke-2 dan ke-3 tidak dimasukkan ke dalam media penyimpanan [3].

Secara matematis, persoalan *integer knapsack* yang dapat dirumuskan sebagai berikut:

Fungsi tujuan Maks/Min

$$Z = \sum_{i=1}^{ii} p_{i \cdot x_i}$$

Kendala

$$z = \sum_{i=1}^{ii} w_{i \cdot x_i} \leq M$$

dengan $x_i = 0$ atau 1 , $i = 1, 2, \dots, n$

keterangan:

Z = nilai optimum dari fungsi tujuan

z = kendala fungsi tujuan

p_i = keuntungan barang- i , dengan $i = 1, 2, \dots, n$

w_i = berat (weight) barang, dengan $i = 1, 2, \dots, n$

M = kapasitas media penyimpanan (*knapsack*)

x_i = banyaknya barang jenis ke- i

C. Algoritma Dynamic Programming

Dynamic Programming merupakan metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Pada penyelesaian persoalan dengan metode *dynamic programming* terdapat beberapa persoalan, antara lain:

1. Terdapat sejumlah berhingga pilihan yang mungkin.
2. Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya.
3. Menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Penerapan algoritma *dynamic programming* rekursif maju (*forward recursion*) pada persoalan *integer knapsack* adalah sebagai berikut:

1. Tahap (k) adalah proses memasukkan objek ke dalam M .
2. Status (y) menyatakan kapasitas muat M yang tersisa setelah memasukkan objek pada tahap sebelumnya.

- a. Dari tahap ke-1, masukkan objek ke-1 ke dalam M untuk setiap satuan kapasitas M sampai batas kapasitas maksimumnya. Karena kapasitas M adalah bilangan bulat, maka pendekatan ini praktis.
- b. Misalkan ketika memasukkan objek pada tahap k , kapasitas muat M sekarang adalah $y - w_k$.
- c. Untuk mengisi kapasitas sisanya, dilakukan dengan menerapkan prinsip optimalitas yang mengacu pada nilai optimum dari tahap sebelumnya untuk kapasitas sisa $y - w_k$ yaitu $f_{k-1}(y - w_k)$.
- d. Selanjutnya membandingkan nilai keuntungan dari objek pada tahap k (yaitu p_k) plus nilai $f_{k-1}(y - w_k)$ dengan keuntungan pengisian hanya $k - 1$ macam objek $f_{k-1}(y)$.
- e. Jika $p_k + f_{k-1}(y - w_k)$ lebih kecil dari $f_{k-1}(y)$, maka objek yang ke- k tidak dimasukkan kedalam M , tetapi jika lebih besar, maka objek yang ke- k dimasukkan.

Relasi rekurens untuk persoalan ini adalah:

$$f_0(y) = 0, y = 0, 1, 2, \dots, M$$

$$f_k(y) = -\infty, y < 0$$

$$f_k(y) = \max\{f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}, k = 1, 2, \dots, n$$

yang dalam hal ini,

$f_k(y)$ = keuntungan optimum dari persoalan *integer knapsack* pada tahap k untuk kapasitas M sebesar y .

$f_0(y) = 0$ adalah nilai dari persoalan *knapsack* kosong dengan kapasitas y .

$f_k(y) = -\infty$ adalah nilai dari persoalan *knapsack* untuk kapasitas *negative*. Solusi optimum dari persoalan *integer knapsack* adalah $f_n(M)$ [4].

Berikut penyelesaian masalah 0-1 *knapsack* dengan menggunakan algoritma *dynamic programming*. Dari data pada Tabel I, dapat dicari solusi dengan beberapa tahap, yaitu:

TABEL I
DATA BOBOT DAN KEUNTUNGAN BARANG ($n = 3$)

Barang Ke-1	Wi	Pi
1	2	65
2	3	80
3	1	30

Kapasitas Media = ($M=5$)

1. Tahap 1

$$f_1(y) = \max\{f_0(y), p_1 + f_0(y - w_1)\}$$

$$= \max\{f_0(y), 65 + f_0(y - 2)\}$$

TABEL II
TAHAP 1 PENCARIAN SOLUSI 0-1 *KNAPSACK*

Y	Solusi Optimum			
	$f_1(y)$	$80 + f_1(y - 3)$	$f_2(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$80 + (-\infty) = -\infty$	65	(1, 0, 0)
3	65	$80 + 0 = 80$	80	(0, 1, 0)
4	65	$80 + 0 = 80$	80	(0, 1, 0)
5	65	$80 + 65 = 145$	145	(1, 1, 0)

2. Tahap 2

$$f_2(y) = \max\{f_1(y), p_2 + f_1(y - w_2)\}$$

$$= \max\{f_1(y), 80 + f_1(y - 3)\}$$

TABEL III
TAHAP 2 PENCARIAN SOLUSI 0-1 *KNAPSACK*

Y	Solusi Optimum			
	$f_1(y)$	$80 + f_1(y - 3)$	$f_2(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$80 + (-\infty) = -\infty$	65	(1, 0, 0)
3	65	$80 + 0 = 80$	80	(0, 1, 0)
4	65	$80 + 0 = 80$	80	(0, 1, 0)
5	65	$80 + 65 = 145$	145	(1, 1, 0)

3. Tahap 3

$$f_3(y) = \max\{f_2(y), p_3 + f_2(y - w_3)\}$$

$$= \max\{f_2(y), 30 + f_2(y - 1)\}$$

TABEL IV
TAHAP 3 PENCARIAN SOLUSI 0-1 *KNAPSACK*

Y	Solusi Optimum			
	$f_2(y)$	$30 + f_2(y - 1)$	$f_3(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$30 + 0 = 30$	30	(0, 0, 1)
2	65	$30 + 0 = 30$	65	(1, 0, 0)
3	80	$30 + 65 = 95$	95	(1, 0, 1)
4	80	$30 + 80 = 110$	110	(0, 1, 1)
5	145	$30 + 80 = 110$	145	(1, 1, 0)

4. Tabel V merupakan ringkasan hasil perhitungan dari tabel tahap 1 sampai tahap 3.

TABEL V
RINGKASAN HASIL PERHITUNGAN ALGORITMA DYNAMIC PROGRAMMING

Jenis Barang	Kapasitas Media					
	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	65	65	65	65
2	0	0	65	80	80	145
3	0	30	65	95	110	145

Solusi optimum $X = (1, 1, 0)$ dengan $\sum p = f = 145$.

D. Tracking (Pelacakan)

Arti *tracking* dalam bahasa Inggris adalah pelacakan. Secara signifikan dalam proses pengiriman barang *tracking* merupakan pelacakan jalur yang ditempuh suatu barang hingga akhirnya sampai ke tujuan.

Tracking berasal dari kata *Track* yaitu jalur. Sedangkan pengertian *tracking* dalam pengiriman barang adalah posisi dimana barang, asal dan tujuan pengiriman dan *route* yang ditempuh.

Tracking adalah suatu proses pencatatan interval perjalanan barang dari tempat asal ke tempat tujuan oleh perusahaan pengangkutan [5].

E. Sistem Informasi

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu [7].

Dari definisi dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu:

- Setiap sistem terdiri dari berbagai unsur.
- Unsur-unsur tersebut merupakan bagian yang tidak terpisahkan dari sistem yang bersangkutan.
- Unsur-unsur didalam sistem tersebut bekerja sama untuk mencapai tujuan sistem.
- Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Terdapat banyak definisi mengenai sistem informasi, beberapa diantaranya adalah sebagai berikut:

- Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [7].
- Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai [8].

F. Proses Bisnis

Proses bisnis adalah sejumlah aktivitas yang mengubah sejumlah *input* menjadi sejumlah *output* (barang dan jasa) untuk orang lain atau proses yang menggunakan orang dan alat [9].

Di dalam proses bisnis terdapat empat aliran entitas yang harus dikelola dengan baik, yaitu:

- Aliran produk (*Flow of goods*).
- Aliran informasi (*Flow of information*).
- Aliran uang (*Flow of money*).
- Aliran dokumen (*Flow of documents*).

Cara pemodelan proses bisnis dapat dirancang dengan cara-cara sebagai berikut [9]:

- Menentukan tujuan, ruang lingkup, dan batasan.
- Pemahaman dan memetakan proses yang berjalan.
- Mengukur kinerja proses.
- Menentukan akar masalah (*root cause*).
- Mengidentifikasi perbaikan proses.
- Implementasi perbaikan proses bisnis

G. Flowchart

Flowchart adalah gambar simbol-simbol yang digunakan untuk menggambarkan urutan proses atau instruksi-instruksi yang terjadi di dalam suatu program komputer secara sistematis dan logis [10]. Ada lima macam bagan alir:

- Bagan Alir Sistem (*System Flowchart*)
- Bagan Alir Dokumen (*Document Flowchart*)
- Bagan Alir Skematik (*Schematic Flowchart*)
- Bagan Alir Program (*Program Flowchart*)
- Bagan Alir Proses (*Process Flowchart*)

Adapun jenis-jenis *flowchart* dapat dibagi menjadi dua, yaitu:

- Program Flowchart*
Simbol-simbol yang menggambarkan proses secara rinci dan detail antara instruksi yang satu dengan instruksi yang lainnya di dalam suatu program komputer yang bersifat secara logik.
- System Flowchart*
Simbol yang menggambarkan urutan prosedur secara detail dalam suatu sistem komputerisasi dan bersifat fisik.

H. Data Flow Diagram

Data Flow Diagram (DFD) adalah teknik grafik yang menggambarkan aliran informasi dan perubahannya yang diperlakukan sebagai pergerakan data dari masukan hingga keluaran [11].

DFD menyediakan dua tujuan, yaitu:

- Menyediakan indikasi bagaimana data di transformasi selama mereka bergerak di dalam sistem.
- Menggambarkan fungsi (dan subfungsi) yang merubah aliran data. DFD menyediakan aliran informasi tambahan yang digunakan selama analisis dari informasi domain dan disajikan sebagai dasar

B. Batasan Optimalisasi Sistem

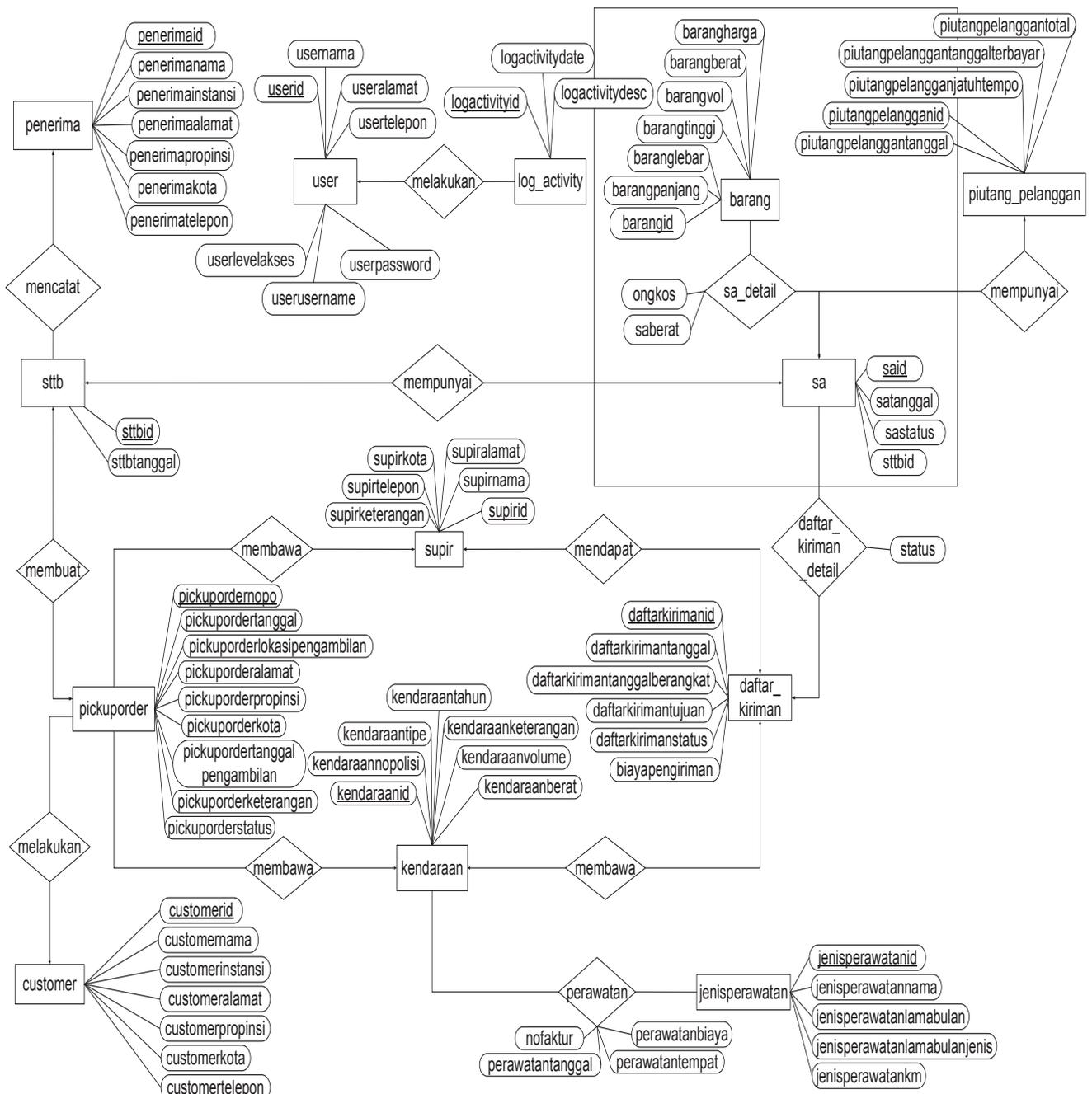
Batasan yang digunakan dalam proses optimalisasi adalah sebagai berikut:

- Ukuran barang yang diterima bersifat bulat positif.
- Setiap barang telah dibungkus dengan wadah solid berbentuk persegi (kubus) atau persegi panjang (balok).
- Ukuran minimal barang adalah 1cm × 1cm × 1cm dan berat minimum barang adalah 1 kg.

- Ukuran maksimal barang adalah 400cm × 400cm × 400cm dan berat maksimal barang adalah 1000kg.

C. Rancangan data

Pemodelan data dalam pengembangan aplikasi ini menggunakan diagram *Entity Relationship* (ERD), dimana merupakan suatu model yang menggambarkan susunan (*layout*) penyimpanan data dari sebuah sistem [6]. Gambar 2 merupakan gambar hasil perancangan ERD pada CV. Labtrans.



Gambar 2. Entity Relationship Diagram

D. Data Flow Diagram (DFD)

Aliran data pada CV. Labatrans digambarkan dengan DFD. Terdapat dua pengguna pada aplikasi yaitu bagian administrasi dan bagian *owner* (pemilik). Berikut penjelasannya:

1. Proses *login* sistem merupakan proses yang harus dilakukan semua pengguna untuk mengetahui proses mana saja yang dapat digunakan oleh setiap pengguna.
2. Proses operasional, proses operasional gudang, proses transportasi, proses keuangan dan proses keuangan hanya dapat digunakan oleh bagian administrasi.
3. Proses laporan hanya dapat digunakan oleh bagian *owner*.

Pada DFD *Level 1* terdapat 7 proses yang akan dijelaskan pada Gambar 3.

Detil aliran data pada DFD level 1 adalah sebagai berikut:

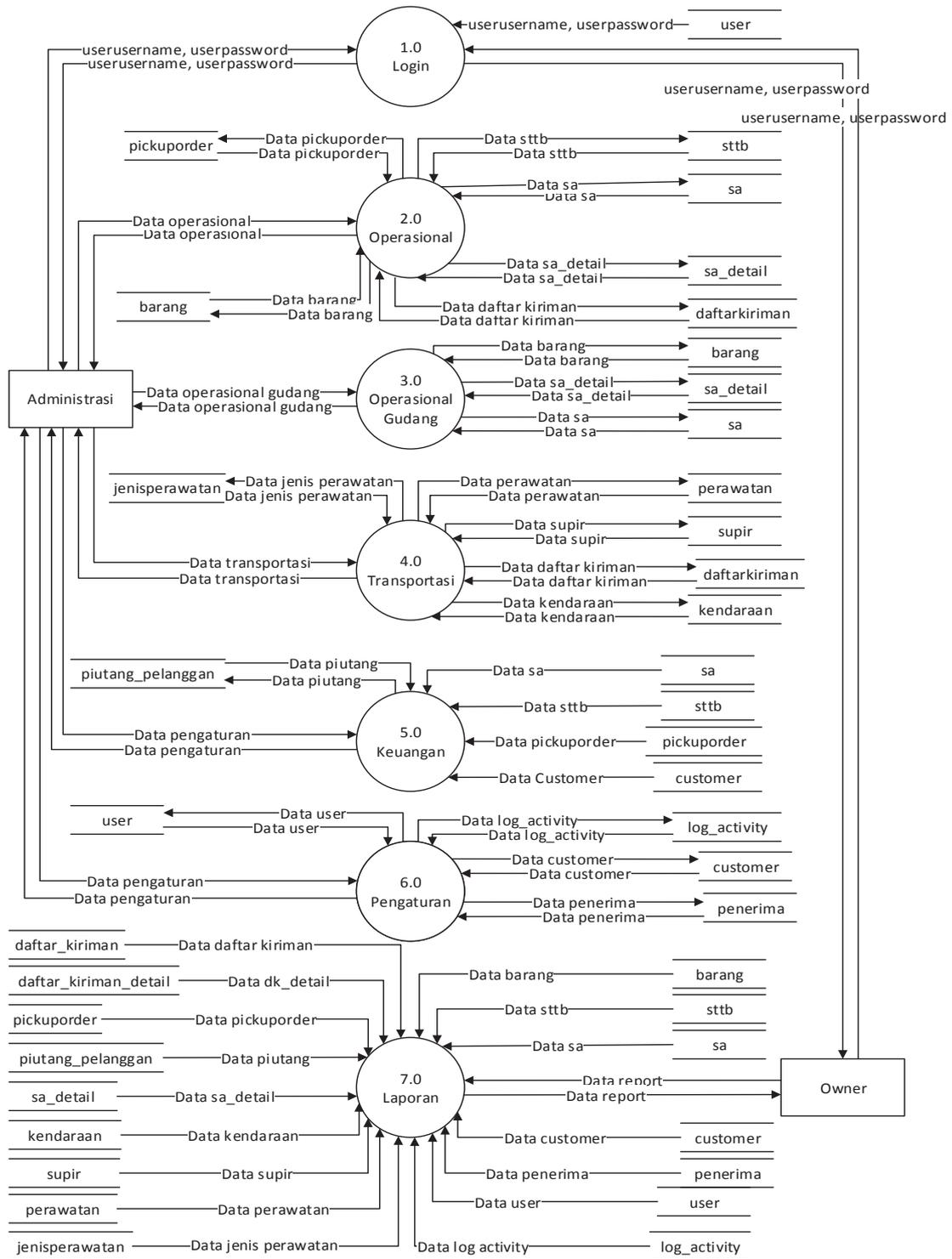
- Data user: *userid*, *username*, *useralamat*, *usertelepon*, *userlevelakses*, *userusername*, *userpassword*, *userstatus*
- Data log activity: *logactivityid*, *logactivitydate*, *logactivitydesc*
- Data pickuporder: *pickuporderid*, *pickupordertanggal*, *pickuporderinstansi*, *pickuporder telepon*, *pickuporderloksipengambilan*, *pickuporderalamat*, *pickuporderpropinsi*, *pickuporderkota*, *pickupordertanggalpengambilan*, *pickuporderketerangan*, *pickuporderstatus*
- Data sttb: *sttbid*, *sttbtanggal*
- Data sa: *said*, *satanggal*, *sastatus*
- Data sa_detail: *saberat*, *ongkos*
- Data barang: *barangid*, *barangnama*, *barangpanjang*, *baranglebar*, *barangtinggi*, *barangvol*, *barangberat*, *barangharga*
- Data daftar kiriman: *daftarkirimanid*, *daftarkirimantanggal*, *daftarkirimantanggalberangkat*, *daftarkirimantujuan*, *daftarkirimanstatus*, *biayapengiriman*
- Data dk_detail: *status*
- Data perawatan: *nofaktur*, *perawatantanggal*, *perawatantempat*, *perawatanbiaya*
- Data jenis perawatan: *jenisperawatanid*, *jenisperawatannama*, *jenisperawatanlamabulan*, *jenisperawatanlamabulanjenis*, *jenisperawatankm*

- Data supir: *supirid*, *supirnama*, *supiralamat*, *supirpropinsi*, *supirkota*, *supirtelepon*, *supirketerangan*
- Data kendaraan: *kendaraaid*, *kendaraannopolisi*, *kendaraantipe*, *kendaraantahun*, *kendaraanketerangan*, *kendaraanvolume*, *kendaraanberat*
- Data piutang: *piutangpelangganid*, *piutangpelanggan tanggal*, *piutangpelangganjatuh tempo*, *piutangpelanggan tanggalterbayar*, *piutangpelanggan total*
- Data customer: *customerid*, *customernama*, *customerinstansi*, *customeralamat*, *customerpropinsi*, *customerkota*, *customer telepon*
- Data penerima: *penerimaid*, *penerima nama*, *penerimainstansi*, *penerima alamat*, *penerima propinsi*, *penerima kota*, *penerima telepon*

E. Batasan Aplikasi

Batasan aplikasi yang dibahas dan dijabarkan pada penelitian ini yaitu:

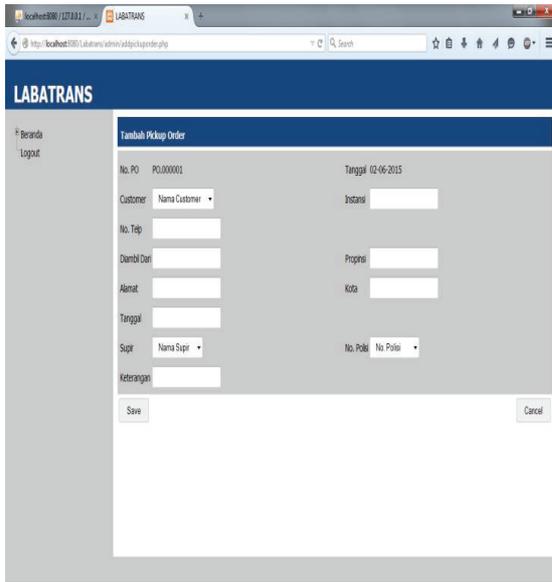
- a. Pembuatan aplikasi ini berdasarkan pengolahan data manual dari CV. Labatrans.
- b. Aplikasi akan diimplementasikan dengan sistem aplikasi berbasis *website*.
- c. Aplikasi ini dapat diakses oleh *owner* (pemilik perusahaan) dan admin (*staff* perusahaan)
- d. Fitur-fitur yang dapat diakses oleh *owner* (pemilik perusahaan) adalah semua fitur laporan yang ada pada aplikasi.
- e. Fitur-fitur yang dapat diakses oleh admin (*staff* perusahaan) adalah:
 - Pembuatan surat pengambilan barang (*Pickup Order*).
 - Pembuatan Surat Tanda Terima Barang (STTB).
 - Pembuatan Surat Angkutan (SA).
 - Pembuatan daftar data pengiriman barang per kendaraan (Daftar Kiriman).
 - Membuat status *pickup order* (fitur ini digunakan untuk mengetahui apakah sudah dilakukan pengambilan barang).
 - Membuat fitur *tracking* (pembuatan fitur *tracking* akan dibuat menyerupai status pengiriman JNE).
 - Membuat fitur barang dalam gudang.
 - Membuat fitur optimalisasi barang.
 - Membuat fitur *log activity user*.



Gambar 3. Data Flow Diagram Level 1

IV. HASIL PENELITIAN

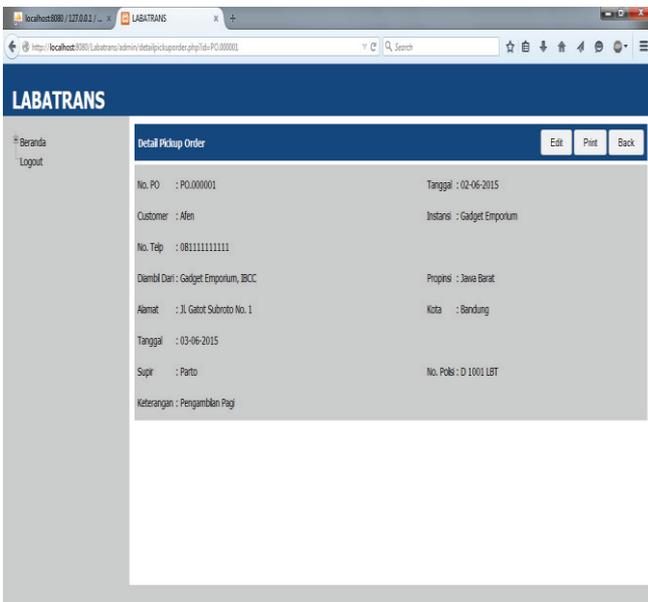
A. Halaman Tambah Pickup Order



Gambar 4 Halaman Tambah Pickup Order

Gambar 4 adalah tampilan tambah *pickup order* yang digunakan oleh admin saat pertama kali menambahkan data *pickup order* baru. Admin mengisi *textbox* dan memilih data dari *dropdownlist* yang telah disediakan. Tombol “Save” dipilih untuk menyimpan data *pickup order* baru sedangkan untuk membatalkan penambahan data admin memilih tombol “Cancel”.

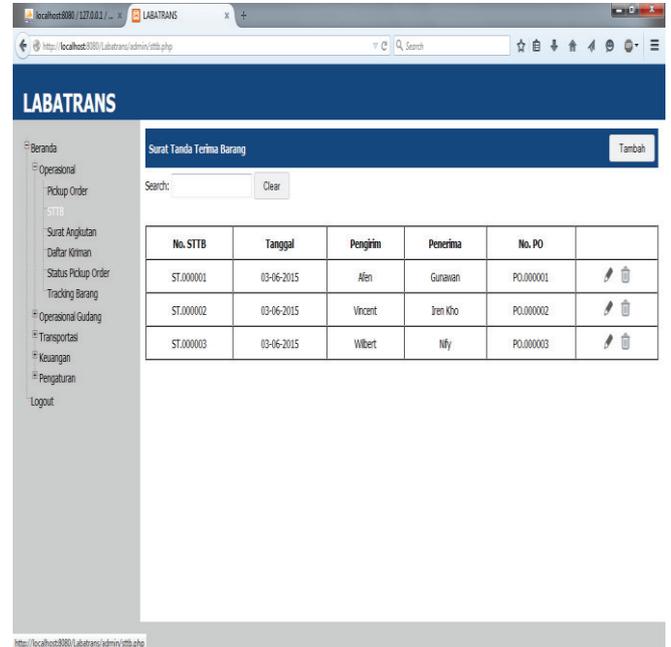
B. Halaman Detail Pickup Order



Gambar 5 Halaman Detail Pickup Order

Gambar 5 berisi label-label yang menampilkan data *pickup order* secara detail, terdapat tombol “Edit” untuk mengubah data *pickup order*, tombol “Print” untuk mencetak surat *pickup order*, dan tombol “Back” dapat digunakan untuk kembali ke menu *pickup order*.

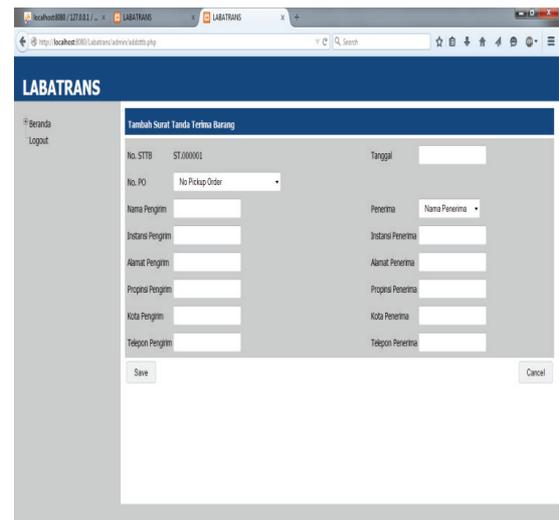
C. Halaman Menu Surat Tanda Terima Barang (STTB)



Gambar 6 Halaman Menu STTB

Gambar 6 adalah tampilan menu Surat Tanda Terima Barang (STTB) yang merupakan kelanjutan dari halaman tambah *Pickup Order* (PO). Halaman ini digunakan pengguna untuk melihat data-data STTB.

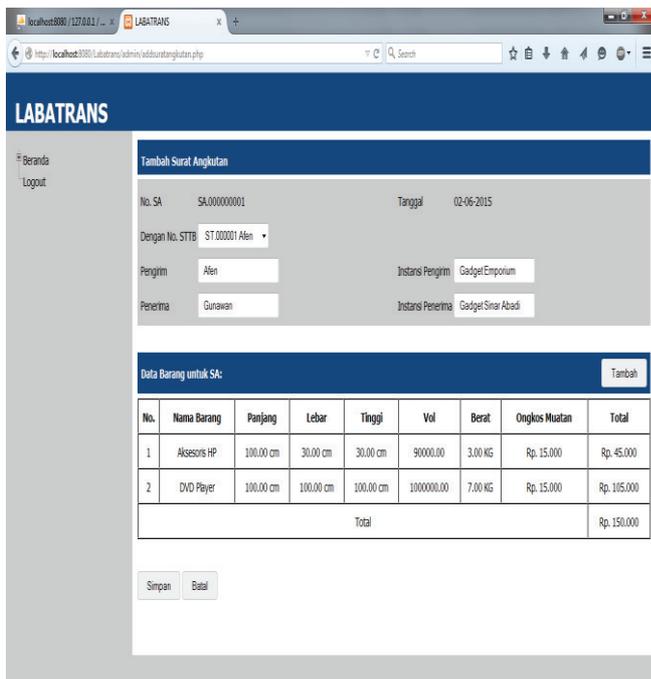
D. Halaman Tambah Surat Tanda Terima Barang (STTB)



Gambar 7 Halaman Tambah STTB

Gambar 7 adalah tampilan tambah Surat Tanda Terima Barang (STTB) apabila admin menekan tombol “Tambah”. Admin akan mengisi tanggal untuk pembuatan STTB, memilih data dari *dropdownlist* No. PO, kemudian memilih data *dropdownlist* penerima. Selanjutnya sistem akan menampilkan data identitas pengirim dan penerima secara otomatis. Setelah itu admin akan menekan tombol “Save” untuk menyimpan data STTB baru atau tombol “Cancel” apabila admin ingin membatalkan penambahan data STTB.

E. Halaman Tambah Surat Angkutan (SA)

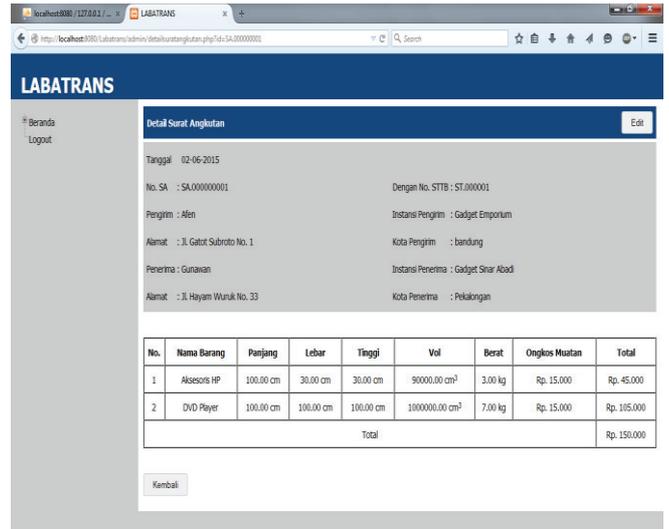


Gambar 8 Halaman Tambah Surat Angkutan

Pada Gambar 8 merupakan lanjutan dari proses tambah STTB, dimana admin dapat melakukan penambahan data Surat Angkutan baru. Pada halaman ini admin hanya memilih No. STTB, lalu sistem menampilkan data identitas pengirim dan penerima secara otomatis. Kemudian tombol “Tambah” dipilih admin untuk memasukkan data barang berdasarkan No. SA. Admin dapat memilih tombol “Simpan” untuk menyimpan data SA baru berdasarkan No. STTB yang telah dipilih. Tombol “Batal” dipilih ketika pengguna ingin membatalkan penambahan data SA.

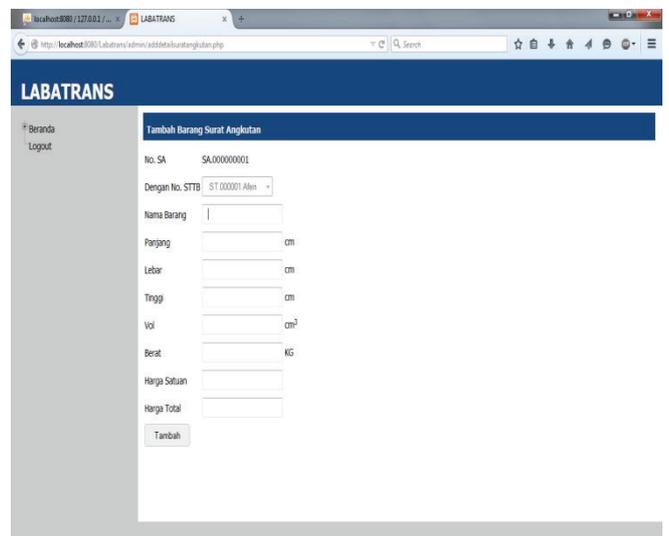
F. Halaman Detail Surat Angkutan (SA)

Pada Gambar 9 berisi label dan tabel yang menampilkan data SA dan data barang berdasarkan No. SA secara detail. Terdapat tombol “Edit” untuk mengubah data SA. Tombol “Kembali” dipilih jika admin ingin kembali ke menu Surat Angkutan.



Gambar 9 Halaman Detail Surat Angkutan

G. Halaman Tambah Barang

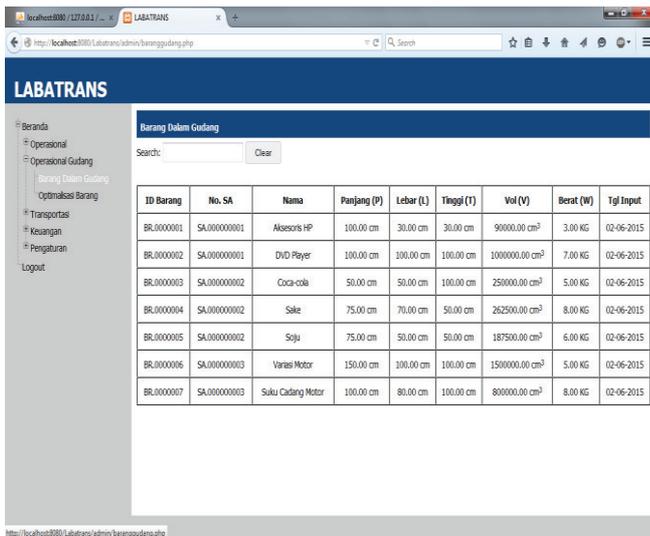


Gambar 10 Halaman Tambah Barang

Gambar 10 merupakan lanjutan dari proses tambah Surat Angkutan, dimana data barang yang akan dimuat ke dalam kendaraan akan dimasukkan detail nya di halaman ini.

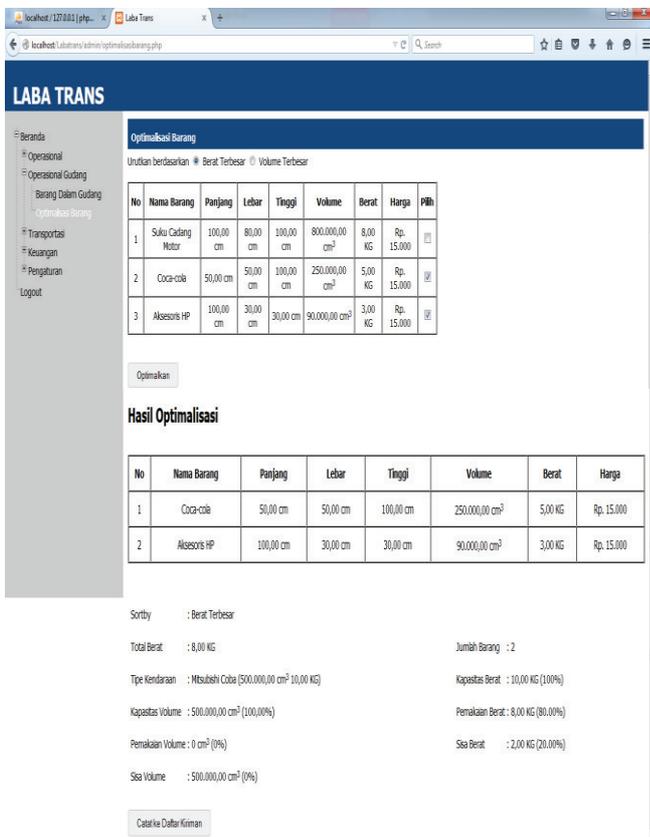
H. Halaman Menu Barang Dalam Gudang

Tampilan menu barang dalam gudang pada gambar 11 digunakan oleh admin untuk melihat barang-barang yang ada didalam gudang berdasarkan No. SA.



Gambar 11 Halaman Menu Barang Dalam Gudang

I. Halaman Optimalisasi Barang



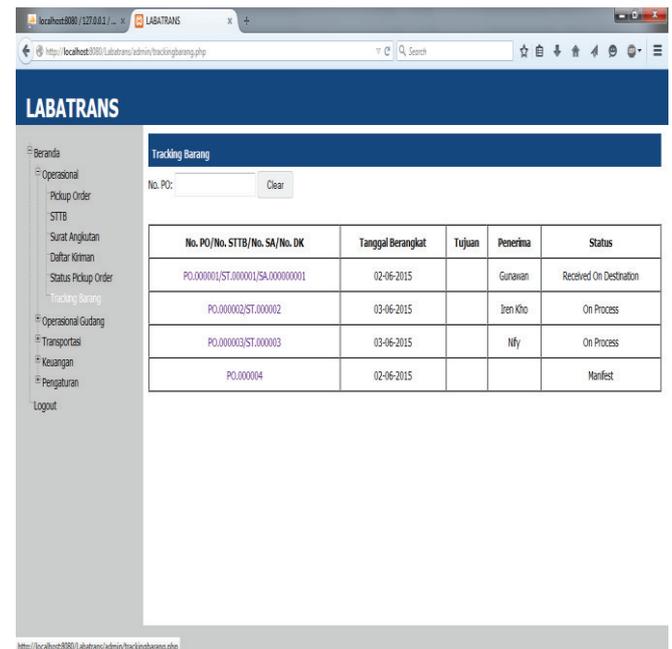
Gambar 12 Halaman Optimalisasi Barang

Gambar 12 merupakan halaman optimalisasi barang yang merupakan kelanjutan dari memasukan halaman Surat Angkutan. Di halaman ini admin memilih barang mana saja yang akan dimuat kedalam kendaraan, bisa berdasarkan barang dengan berat terbesar atau volume terbesar. Setelah itu admin menekan tombol “Optimalkan” sehingga sistem

akan melakukan proses optimalisasi untuk menentukan kapasitas kendaraan mana yang sesuai dengan barang yang telah dipilih agar ketika barang akan dimuat kedalam kendaraan tidak melebihi kapasitas kendaraan.

Detil hasil dari optimalisasi yaitu: urutan berdasarkan total berat, tipe kendaraan, kapasitas volume, pemakaian volume, sisa volume, jumlah barang, kapasitas berat, pemakaian berat, dan sisa berat. Setelah barang dimuat ke dalam kendaraan, selanjutnya admin dapat menyimpan datanya, dan sistem akan melakukan perhitungan optimalisasi yang baru apabila terdapat barang baru yang akan dimuat lagi ke dalam kendaraan.

J. Halaman Tracking Barang

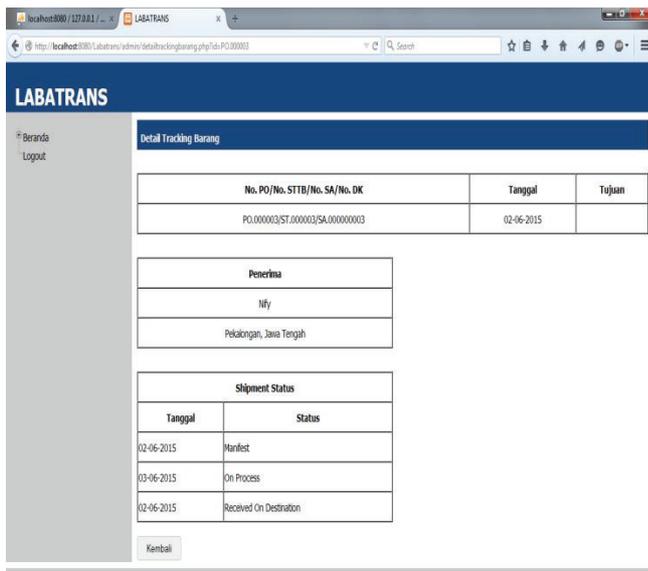


Gambar 13 Halaman Tracking Barang

Gambar 13 digunakan untuk melihat keberadaan data barang yang dikirim pada CV. Labatrans. Status yang bisa didapat oleh konsumen yang ingin melihat yaitu: apakah barang sudah diterima, sudah diproses, atau sudah terkirim. Admin dapat memilih data dari kolom “No. PO” atau “No. STTB” atau “No. SA” atau “No. DK” untuk mengetahui detail barang yang dipilih.

K. Halaman Detail Tacking Barang

Gambar 14 merupakan lanjutan dari halaman Tracking Barang. Halaman ini menampilkan detil barang lengkap beserta status saat admin memilih barang yang hendak diketahui keberadaannya.



Gambar 14. Halaman Detail Tracking Barang

V. SIMPULAN

A. Simpulan

Berdasarkan hasil analisa dan perancangan, dapat disimpulkan bahwa:

1. Aplikasi membuat sistem optimalisasi dan *tracking* pada pengiriman barang secara *desktop*.
2. Aplikasi ini membuat sistem optimalisasi barang untuk CV. Labatrans agar pada saat memuat barang tidak terjadi *overload*.
3. Aplikasi ini menjadi informasi bagi konsumen ketika konsumen ingin mengetahui keberadaan barang secara *offline*.

B. Saran

Beberapa saran yang dapat digunakan sebagai hasil pembahasan dan pengembangan, yaitu:

1. Aplikasi ini dapat dikembangkan dengan penambahan fitur penyusunan atau pengepakan barang kedalam kontainer kendaraan secara 2 dimensi atau 3 dimensi.
2. Aplikasi ini dapat dikembangkan dengan penambahan fitur *pickup order* secara *online*.
3. Aplikasi ini dapat dikembangkan dengan penambahan fitur *tracking* barang secara *online*.
4. Aplikasi dapat ditambah fitur statistika atau grafik sesuai kebutuhan CV. Labatrans.

DAFTAR PUSTAKA

- [1] I. Kamran, *Fundamental Engineering Optimization Methods*, 1st ed., Kamran Iqbal & bookboon.com, 2013.
- [2] T. T. Dimiyati and A. Dimiyati, *Operations Research: Model-model Pengambilan Keputusan*, Bandung: Sinar Baru Agesindo, 2006.
- [3] T. H. A, *Riset Operasi*, Jakarta: Binarupa Aksara, 2005.
- [4] H. F. S and J. L. Gerald, *Introduction to Operations Research Eight Edition Jilid II*, Jakarta: Andi, 2005.
- [5] T. Rumapea, *Kamus Lengkap Perdagangan Internasional*, Jakarta: Gramedia Pustaka Utama, 2008
- [6] Fathansyah, *Basis Data*, Bandung: Informatika, 2007
- [7] H. M. Jogiyanto, *Analisa dan Desain Sistem Informasi (Pendekatan Terstruktur)*, Yogyakarta: Andi, 2010.
- [8] J. A. O'Brien, *Pengantar Sistem Informasi*, Edisi 12, Jakarta: Salemba Empat, 2005.
- [9] R. E. Indrajit & R. Djokopranoto, *Konsep Manajemen Supply Chain*, Jakarta: PT. Grasindo, 2012.
- [10] Sugiyono, *Pemrograman Terstruktur untuk Pelajar dan Mahasiswa*, Kuningan: Panji Gumilang Press, 2005.
- [11] R. S. Pressman, *Rekayasa Perangkat Lunak (Pendekatan Praktisi)*, Yogyakarta: Andi, 2012.
- [12] A. Kristanto, *Perancangan Sistem Informasi*, Yogyakarta: Gava Media, 2008