

BAB III

PERANCANGAN SISTEM

3.1 Analisis Sistem

Untuk dapat merancang suatu sistem informasi, diperlukan sebuah data-data yang lengkap seputar suatu bagian dari organisasi atau perusahaan tersebut dalam hal ini adalah sistem penentuan jumlah peti kemas yang dimuat pada kapal laut pada PT. Anugerah Dwi Sukses. Data-data tersebut didapatkan dari hasil identifikasi permasalahan melalui wawancara dan observasi pada beberapa pihak terkait seputar pekerjaan yang biasa dilakukan.

Data-data yang terkumpul dari hasil wawancara dan observasi tersebut kemudian dibuat sebuah *document flow* untuk dapat mendeskripsikan alur dari proses bisnis dan *job desk* masing-masing *entity*. Adapun *Document flow* yang dimaksud adalah sebagai berikut :

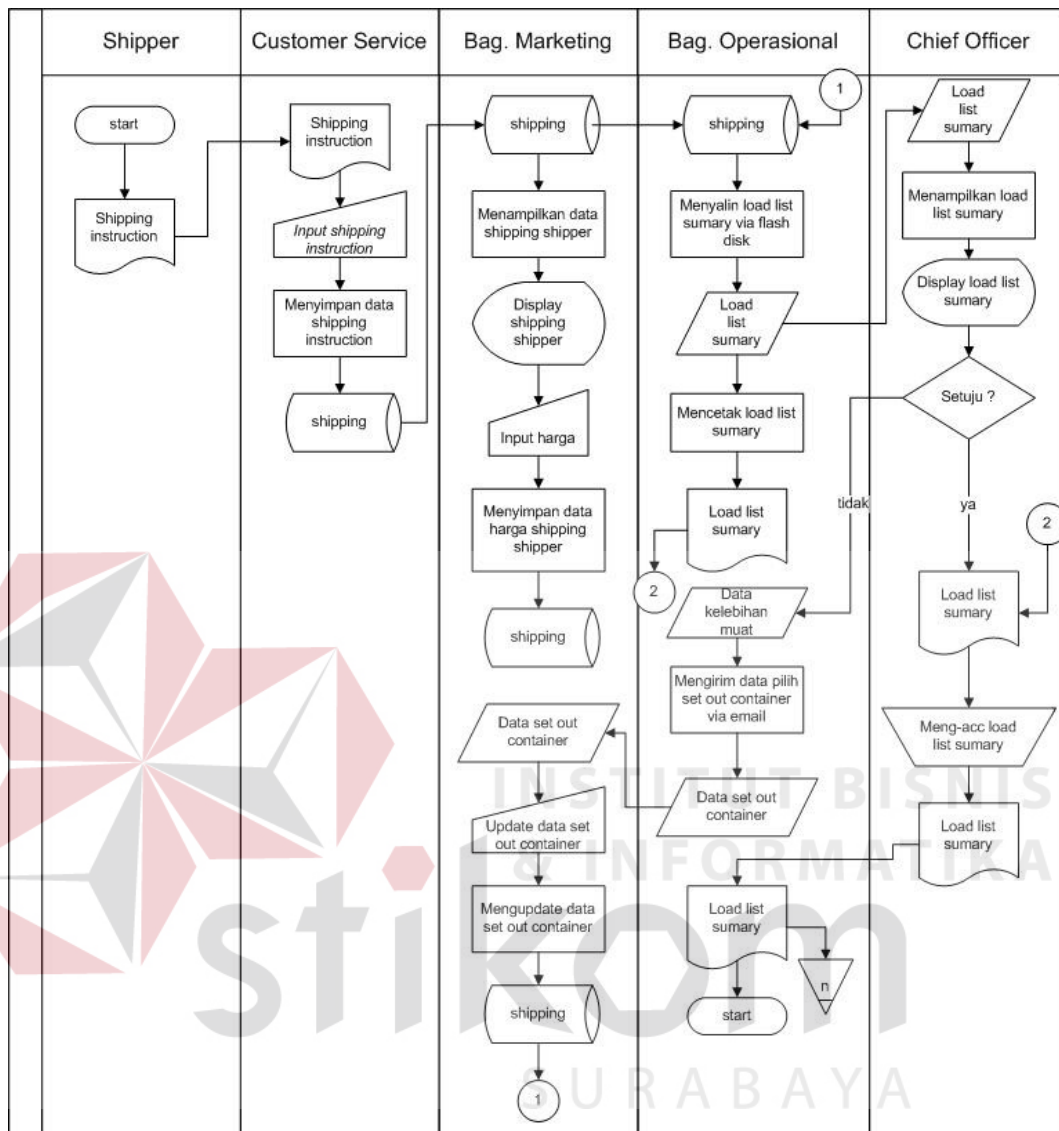
3.1.1 System Flow Lama

Gambar 3.1 menjelaskan bahwa *entity shipper* menyerahkan *shipping instruction* kepada *Customer Service* pelayaran, kemudian pihak *Customer Service* menyimpan data *shipping instruction* ke dalam sistem. Pada *entity* bagian *marketing* dapat melakukan *display* data transaksi *shipping*, dan dapat melakukan *input* data harga pada transaksi tersebut untuk disimpan ke dalam sistem. Pada *entity* bagian Operasional bertugas untuk mengirimkan data *load list* kepada *chief officer* dengan cara melakukan *copy file* dari sistem ke dalam *flash disk*. Data

tersebut kemudian dilihat oleh *chief officer* untuk dilakukan verifikasi mengenai pemuatan peti kemas pada kapal laut. Apabila data *load list* tersebut tidak sesuai dengan prosedur muat *safety* kapal laut maka *chief officer* akan memberitahukan informasi mengenai kelebihan muatan ke pada bagian operasional. bagian operasional akan mengirimkan data kelebihan muat tersebut kepada bagian *marketing* via *e-mail*.

Bagian *marketing* akan menerima data kelebihan tersebut dan melakukan *set out* peti kemas dengan berat yang mendekati dengan kelebihan tersebut. Setelah selesai melakukan *set out* akan muncul data *load list* yang baru di dalam sistem. Kemudian bagian *marketing* akan mengirim pesan via *e-mail* kepada bagian Operasional bahwa data kelebihan muat sudah teratasi.

Bagian Operasional akan menyampaikan data *load list* yang baru kepada *chief officer* dengan menggunakan *flash disk*. Kemudian *chief officer* melakukan verifikasi lagi berdasarkan *safety* muat peti kemas terhadap kapal laut, apabila setuju maka bagian operasional akan membuat *load list summary* untuk ditandatangani oleh *chief officer* sebagai bukti bahwa data peti kemas tersebut telah disetujui untuk dimuat ke kapal laut.

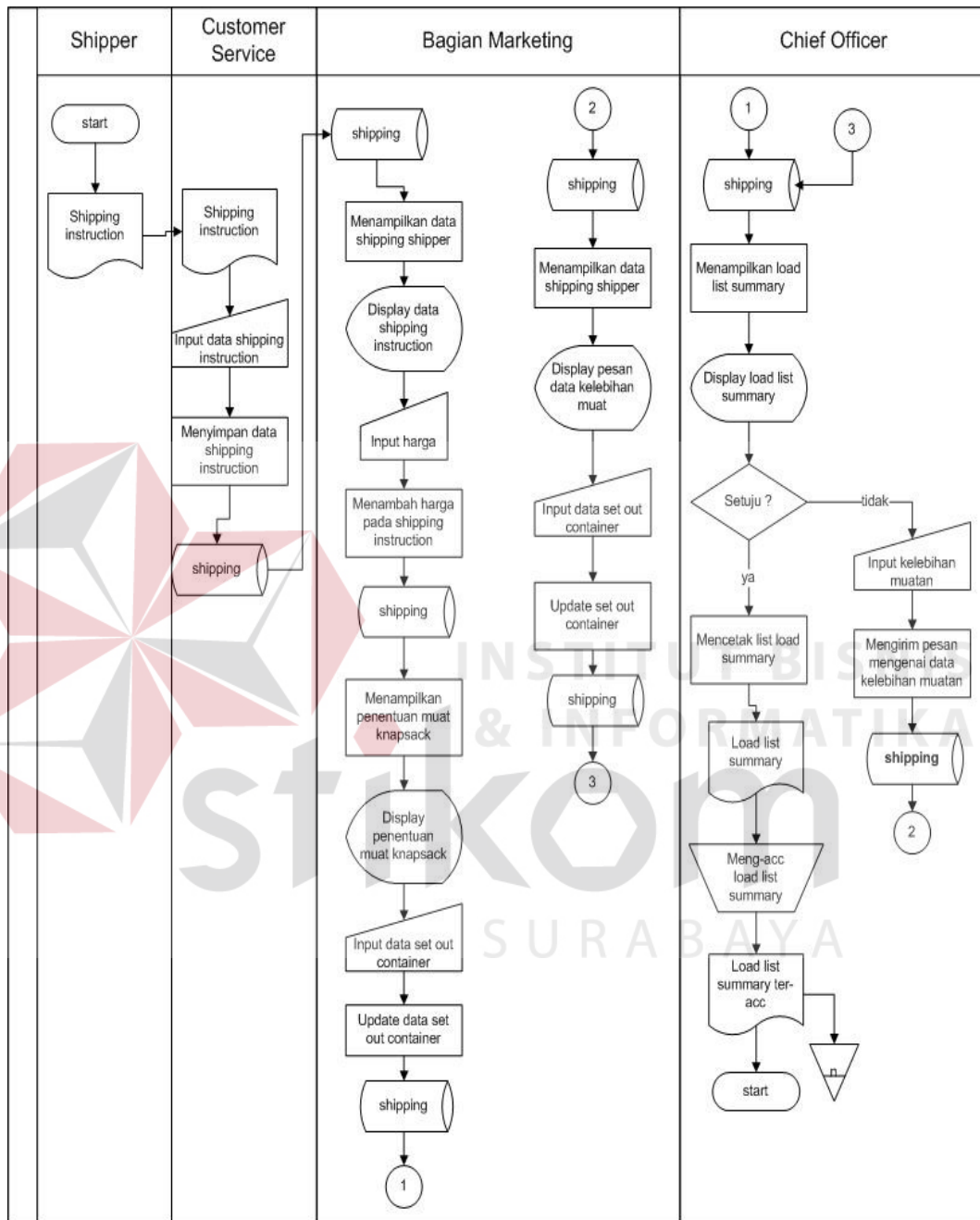


Gambar 3.1 System Flow Lama Penentuan Jumlah Peti Kemas yang Dimuat Pada Kapal Laut

3.2 Perancangan Sistem

Setelah analisis sistem yang dijelaskan dengan *document flow* selesai dibuat, maka langkah selanjutnya adalah merancang sistem sesuai dengan topik bahasan. Dalam rancang sistem ini akan terbagi beberapa bagian yaitu *System Flow*, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, *Struktur Database*, *Desain input/output*, *Rencana Evaluasi Sistem*.

3.2.1 System Flow Rekomendasi



Gambar 3.2 System Flow Penentuan Jumlah Peti Kemasan yang dimuat Pada Kapal Laut Rekomendasi

Gambar 3.2 menjelaskan bahwa *entity shipper* menyerahkan *shipping instruction* kepada *Customer Service* pelayaran, kemudian pihak *Customer*

Service menyimpan data *shipping instruction* tersebut ke dalam sistem. Pada *entity* bagian *marketing* dapat melakukan *display* data transaksi *shipping*, dan dapat melakukan *input* data harga pada transaksi tersebut untuk disimpan ke dalam sistem. Bagian *marketing* dapat melakukan penentuan muat peti kemas dengan menggunakan metode *knapsack*. Hasil dari pada perhitungan tersebut sebagai tolak ukur untuk melakukan *set out* peti kemas.

Setelah bagian *marketing* melakukan *set out* peti kemas maka sistem akan melakukan *update* data. *chief officer* dapat menampilkan data dari sistem kemudian melakukan verifikasi data muat tersebut berdasarkan prosedur *safety* muat peti kemas pada kapal laut. Apabila *chief officer* tidak menyetujui data tersebut maka *chief officer* dapat mengirim pesan berupa data kelebihan muat kepada bagian *marketing*.

Setelah pesan sudah dikirim kepada bagian *marketing* sistem akan memberikan notifikasi kepada bagian *marketing* bahwa terdapat pesan dari *chief officer*. bagian *marketing* dapat membuka pesan tersebut yang berisikan data kelebihan muat. bagian *marketing* melakukan *set out* berdasarkan dari data kelebihan yang diterima. Setelah melakukan *set out* maka sistem akan melakukan *update* data. bagian *marketing* yang telah melakukan *set out* dapat mengirimkan pesan kepada *chief officer* sebagai tanda bahwa data telah dilakukan perubahan berdasarkan permintaan *chief officer*. Setelah dilakukan pengiriman pesan yang dilakukan oleh bagian *marketing* terhadap *chief officer*, maka dari sistem akan memberikan notifikasi pesan yang telah diterima. *chief officer* dapat membuka pesan dan melakukan verifikasi terhadap data yang telah dilakukan perubahan.

Jika setuju maka *chief officer* akan melakukan cetak *load list summary* dan melakukan tanda tangan terhadap laporan tersebut dan dilakukan pengarsipan sesuai dengan tanggal tersebut.

3.2.2 Data Flow Diagram (DFD)

Setelah dibuat desain *system flow*, langkah selanjutnya yaitu membuat desain DFD untuk mengetahui proses-proses apa saja yang dibutuhkan untuk dapat memenuhi kebutuhan sistem informasi penentuan jumlah muat peti kemas pada kapal laut dengan metode *knapsack*.

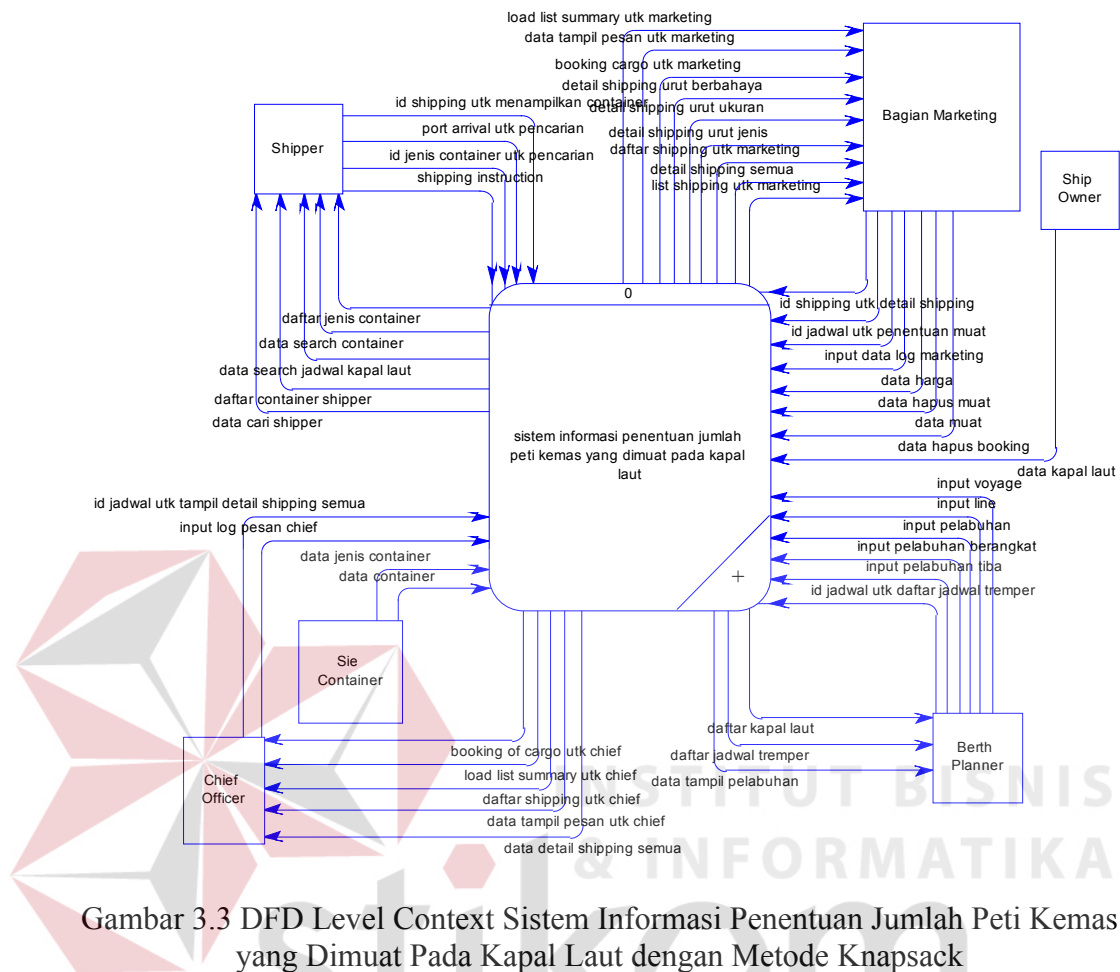
1. DFD Level Context Sistem Informasi Penentuan Jumlah Peti Kemas yang dimuat pada Kapal Laut dengan Metode Knapsack

Gambar 3.3, merupakan DFD level *context* gambar tersebut terdapat 6 *entity* yang terlibat dalam *input/output* data ke dan dari sistem. *External entity* tersebut adalah *external entity shipper*, *external entity* bagian *marketing*, *external entity ship owner*, *external entity chief officer*, *external entity sie container*, dan *external entity Berth Planner*. Adapun *external entity shipper* memasukkan data ke sistem adalah, *id shipping* untuk menampilkan *container*, *port arrival* untuk pencarian, *id jenis container* untuk pencarian, dan *shipping instruction*.

Ouput yang diperoleh *external entity shipper* dari sistem adalah daftar jenis *container*, data *search container*, data cari jadwal kapal laut, daftar *container shipper*, dan data cari *shipper*. *External entity* bagian *marketing* melakukan *input* data ke sistem adalah, *id shipping* untuk detail *shipping*, *id jadwal* untuk penentuan muat, data *input log marketing*, data harga, data hapus muat, data muat,

dan data hapus *booking*. *Output* yang diperoleh *external entity* untuk bagian *marketing* dari sistem adalah *load list summary* untuk bagian *marketing*, data tampil pesan untuk bagian *marketing*, *booking of cargo* untuk bagian *marketing*, data detail *shipping*urut barang berbahaya, data detail *shipping*urut ukuran peti kemas, data detail *shipping*urut jenis peti kemas, daftar *shipping* untuk bagian *marketing*, data detail *shipping* untuk bagian *marketing*, dan data *list shipping* untuk bagian *marketing*. *External entity ship owner* hanya menginputkan data kapal laut ke sistem, dan tidak terdapat *output* dari sistem ke *external entity ship owner*. Adapun *external entity sie container* memasukkan data ke sistem adalah data jenis *container* dan data *container*. Tidak terdapat *output* dari sistem untuk *external entity sie container*.

External entity chief officer memasukkan data ke sistem adalah *id* jadwal untuk menampilkan detail *shipping* dan data *input log* pesan. *Output* yang diterima dari sistem kepada *external entity chief officer* adalah data *booking of cargo* untuk *chief officer*, data *load list summary* untuk *chief officer*, daftar *shipping* untuk *chief officer*, data tampil pesan untuk *chief officer*, data detail *shipping* untuk *chief officer*. *External entity Berth Planner* memasukkan data ke sistem adalah data *input voyage*, data *input line* kapal laut, data *input* pelabuhan, data *input* pelabuhan berangkat, data *input* pelabuhan tiba, dan *id* jadwal untuk menampilkan daftar jadwal kapal laut berjenis *tremper*.



Gambar 3.3 DFD Level Context Sistem Informasi Penentuan Jumlah Peti Kemas yang Dimuat Pada Kapal Laut dengan Metode Knapsack

2. DFD Level 0 Sistem Informasi Penentuan Jumlah Peti Kemas yang Dimuat Pada Kapal Laut dengan Metode Knapsack

Gambar 3.4, merupakan DFD level 0 dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Pada gambar tersebut terdapat 6 *entity* yang terlibat dalam *input/output* data ke dan dari sistem. *External entity* tersebut adalah *external entity shipper*, *external entity bagian marketing*, *external entity ship owner*, *external entity chief officer*, *external entity sie container*, dan *external entity Berth Planner*. Proses *mastering container*, proses melakukan konfirmasi *shipping instruction*, proses memberikan

uang tambang, dan proses *mastering* jadwal kapal laut. Pada gambar terdapat pula *data store* yang berjumlah 14 *data store* yaitu, *data store container*, *data store jenis container*, *data store shipping container*, *data store notify party*, *data store consignee*, *data store shipper*, *shipping*, *data store log pesan*, *data store pelabuhan*, *data store pelabuhan tiba*, *data store pelabuhan berangkat*, *data store voyage kapal*, *data store line kapal*, dan *data store kapal laut*.

Pada proses *mastering container* terdapat *external entity* *sie container* yang memberikan 2 *input* ke proses yaitu data jenis *container* dan data *container*. Terdapat pula data *input* dari 2 *data store* ke proses yaitu *data store container* dan *data store jenis container* dengan masing-masing data *output* jenis *container* dan *container*. Dari proses *mastering container* sendiri mengeluarkan 3 *output* yaitu data *input shipping container* yang ditujukan untuk proses melakukan konfirmasi *shipping instruction*, data *input container* yang ditujukan untuk *data store container*, dan data *input jenis container* yang ditujukan pada *data store jenis container*.

Proses melakukan konfirmasi *shipping instruction* terdapat *external entity* *shipper* yang memberikan 4 *input* ke proses tersebut yaitu *id shipping* untuk menampilkan *container*, data *port arrival* untuk pencarian, *id jenis container* untuk pencarian, dan *shipping instruction*. Terdapat data *input* dari 5 *data store* ke proses yaitu data *output shipping*, data *output shipper*, data *output consignee*, data *output notify party*, dan data *output shipping container*. Terdapat pula 2 proses yang memasukkan ke proses melakukan konfirmasi *shipping instruction* yaitu proses *mastering container* dan proses *mastering* jadwal kapal laut, dengan data

input masing-masing adalah data *input shipping container* dan data *input* jadwal kapal laut untuk *shipping*. Dari proses melakukan konfirmasi *shipping instruction* sendiri mengeluarkan *output* 4 terhadap *data store* yaitu *data store*, *shipping container*, *data store notify party*, *data store notify party*, *data store consignee*, *data store shipper*, dan *data store shipping* dengan masing-masing *output shipping*, *shipper*, *consignee*, *notify party*, dan *shipping container*. Satu *output* terakhir diberikan kepada proses memberikan uang tambang yaitu data ambil *shipping* untuk *freight*. Proses memberikan uang tambang terdapat 2 *external entity* yang memasukkan data ke proses yaitu *external entity* bagian *marketing* dan *external entity chief officer*.

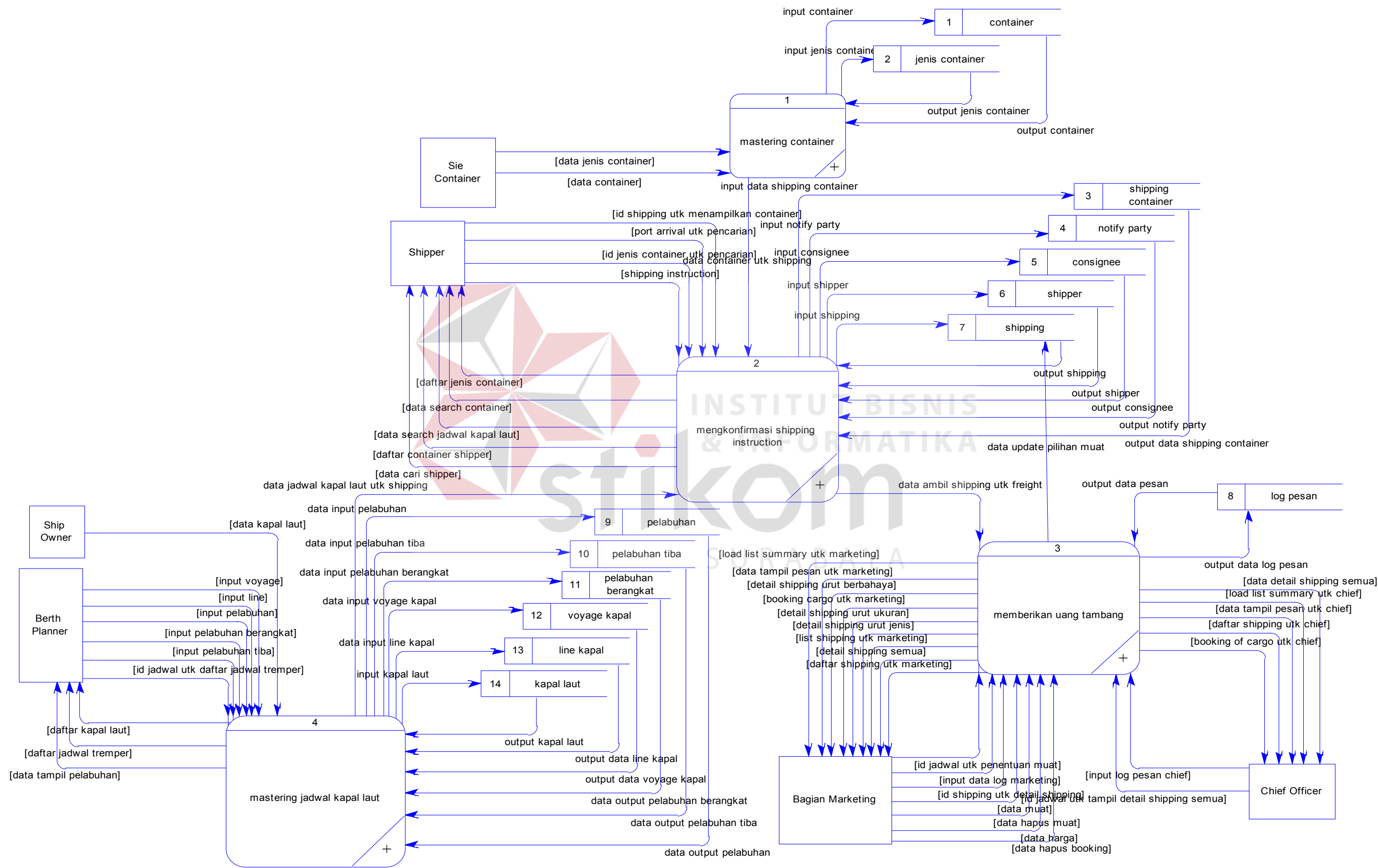
Adapun *external entity* bagian *marketing* memasukkan beberapa data diantaranya adalah *id* jadwal untuk penentuan muat, data *input log* pesan *marketing*, *id shipping* untuk detail *shipping*, data muat, data hapus muat, data harga, dan data hapus *booking*. *External entity chief officer* memasukkan beberapa data ke proses diantaranya adalah *input log* pesan *chief* dan *id shipping* untuk menampilkan detail *shipping*. *Data store log* pesan memasukkan data *output log* pesan ke proses dan proses melakukan konfirmasi *shipping instruction* memasukkan data ambil *shipping* untuk *freight* ke proses.

Proses memberikan uang tambang memberikan beberapa *input* ke *external entity* bagian *marketing* diantaranya adalah data *load list summary* untuk *marketing*, data tampil pesan untuk *marketing*, detail *shipping*urut barang berbahaya, *booking of cargo* untuk *marketing*, detail *shipping*urut ukuran peti kemas, detail *shipping*urut jenis peti kemas, *list shipping* untuk *marketing*, data

detail *shipping* untuk *marketing*, dan daftar *shipping* untuk *marketing*. Adapun proses memberikan uang tambang memberikan beberapa *input* ke *external entity chief officer* diantaranya adalah data detail *shipping* untuk *chief*, *load list summary* untuk *chief*, data tampil pesan untuk *chief*, daftar *shipping* untuk *chief*, dan *booking of cargo* untuk *chief*.

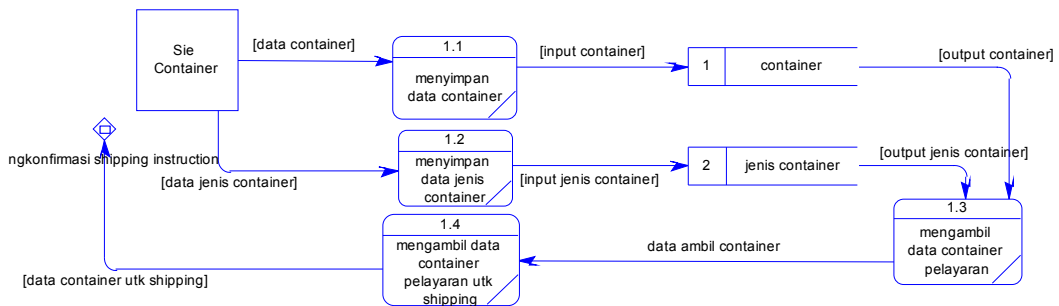
Proses memberikan uang tambang juga memberikan data *output shipping* ke *data store shipping* dan data *output log* pesan ke *data store log* pesan. Proses *mastering* jadwal kapal laut terdapat 2 *external entity* yang memberikan data ke proses yaitu *external entity bert planner* dan *external entity ship owner*. Pada *external entity bert planner* memberikan beberapa data ke sistem diantaranya adalah data *input voyage* kapal, data *input line* kapal, data *input* pelabuhan, data *input* pelabuhan berangkat, data *input* pelabuhan tiba, dan *id* jadwal untuk daftar jadwal kapal laut jenis *tremper*.

Pada *external entity ship owner* memasukkan data kapal laut ke proses. Terdapat 6 *data store* yang memasukkan data ke sistem yaitu *data store* pelabuhan, *data store* pelabuhan tiba, *data store* pelabuhan berangkat, *data store voyage* kapal, *data store line* kapal, dan *data store* kapal laut. Pada proses *mastering* jadwal kapal laut sendiri memasukkan beberapa data ke *external entity Berth Planner* diantaranya adalah data daftar kapal laut, data daftar jadwal kapal laut, data tampil pelabuhan.



Gambar 3.4 DFD Level 0 Sistem Informasi Penentuan Jumlah Peti Kemas yang Dimuat Pada Kapal Laut dengan Metode Knapsack

3. DFD Level 1 Mastering Container



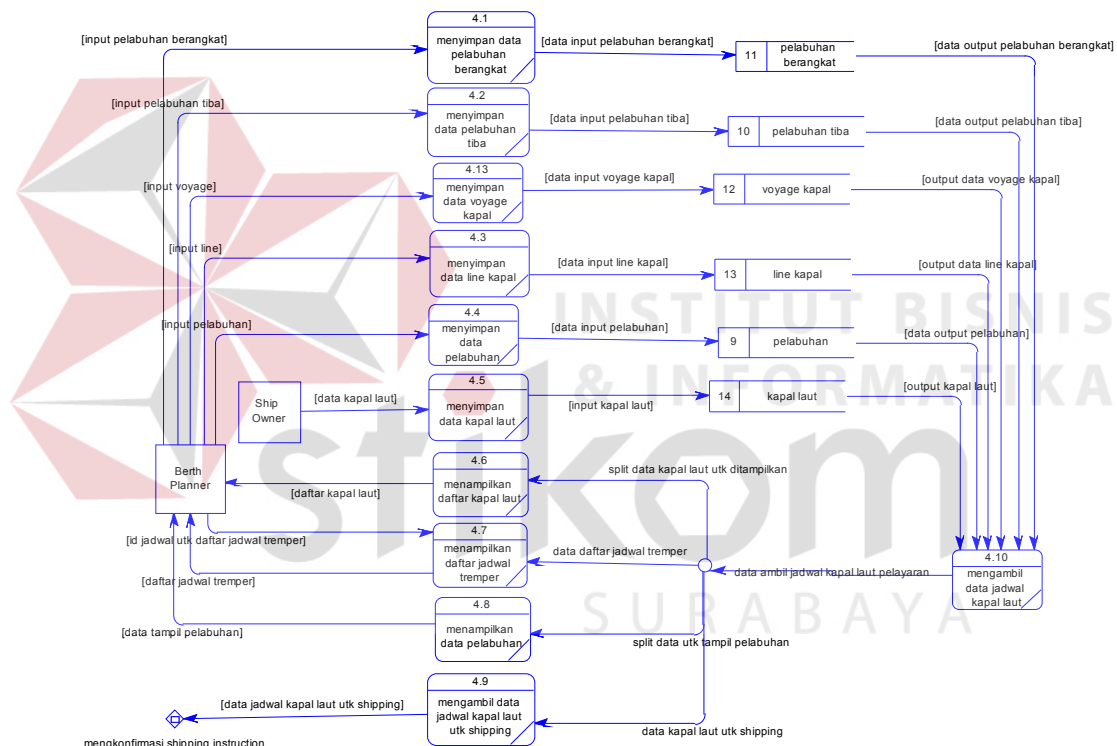
Gambar 3.5 DFD Level 1 Mastering Container

Gambar 3.5, merupakan DFD level 1 hasil *decompose* proses *mastering container* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut. Terdapat *external entity sie container* dan 4 proses yaitu proses menyimpan data *container*, proses menyimpan data jenis *container*, proses mengambil data *container* pelayaran, dan proses mengambil data *container* pelayaran untuk *shipping*. Terdapat juga 2 *data store* yaitu *data store container* dan *data store jenis container*. Pada proses menyimpan data *container*, terdapat data *input* dari *external entity sie container* yaitu data *container* dan mengeluarkan *output* data *input container* yang ditujukan ke *data store container*.

Pada proses menyimpan data jenis *container*, terdapat data *input* dari *external entity sie container* yaitu data jenis *container* dan mengeluarkan *output* data *input jenis container* yang ditujukan ke *data store jenis container*. Proses mengambil data *container* menerima data *input* dari *data store container* dan jenis *container* masing-masing data adalah data *output container* dan data *output jenis container*. Proses mengambil data *container* pelayaran mengeluarkan *output* data ambil *container* yang ditujukan pada proses mengambil data *container* pelayaran

untuk *shipping* yang kemudian data menghasilkan *output* data *container* untuk *shipping* yang ditujukan pada proses melakukan konfirmasi *shipping instruction*. Demikian penjelasan dari DFD level 1 proses *mastering container* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*.

4. DFD Level 1 Mastering Jadwal Kapal Laut



Gambar 3.6 DFD Level 1 Mastering Jadwal Kapal Laut

Gambar 3.6, merupakan DFD level 1 hasil *decompose* proses *mastering* jadwal kapal laut dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut. Terdapat 2 *external entity* yaitu *external entity Berth Planner* dan *external entity ship owner*. Terdapat 9 proses yaitu proses menyimpan data pelabuhan berangkat, proses menyimpan data pelabuhan tiba, proses menyimpan

data *voyage* kapal, proses menyimpan data *line* kapal, proses menyimpan data *line* kapal, proses menyimpan data pelabuhan, proses menampilkan daftar kapal laut, proses menampilkan daftar jadwal *tremper*, proses menampilkan data pelabuhan, proses mengambil data jadwal kapal laut, dan proses mengambil data jadwal kapal laut. Terdapat juga 6 *data store* yaitu *data store* pelabuhan berangkat, *data store* pelabuhan tiba, *data store voyage* kapal, *data store line* kapal, *data store* pelabuhan, dan *data store* kapal laut. Pada proses menyimpan data pelabuhan berangkat, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input* pelabuhan berangkat yang ditujukan pada *data store* pelabuhan berangkat.

Pada proses menyimpan data pelabuhan tiba, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input* pelabuhan tiba yang ditujukan pada *data store* pelabuhan tiba. Proses menyimpan data *voyage* kapal, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input voyage* kapal yang ditujukan pada *data store voyage* kapal. Proses menyimpan data *line* kapal, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input line* kapal yang ditujukan pada *data store line* kapal.

Proses menyimpan data pelabuhan, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input* pelabuhan yang ditujukan pada *data store* pelabuhan. Proses menyimpan data kapal laut, terdapat data *input* dari *external entity Berth Planner* dan mengeluarkan *output* data *input* kapal laut yang ditujukan pada *data store* kapal laut. Proses menampilkan daftar kapal laut,

terdapat data *input split* kapal laut untuk ditampilkan dari proses mengambil data jadwal kapal laut. Proses tersebut menghasilkan data daftar kapal laut yang ditujukan untuk *external entity Berth Planner*. Proses menampilkan daftar jadwal *tremper*, terdapat data *input split* daftar jadwal *tremper* dari proses mengambil data jadwal kapal laut. Proses tersebut menghasilkan data daftar kapal laut yang ditujukan untuk *external entity Berth Planner*.

Proses menampilkan data pelabuhan, terdapat data *input split* untuk pelabuhan dari proses mengambil data jadwal kapal laut. Proses tersebut menghasilkan data daftar kapal laut yang ditujukan untuk *external entity Berth Planner*. Proses mengambil data jadwal kapal laut untuk *shipping*, terdapat data *input split* kapal laut untuk *shipping* dari proses mengambil data jadwal kapal laut. Proses tersebut menghasilkan data daftar kapal laut yang ditujukan untuk *external entity Berth Planner*.

Proses mengambil data jadwal kapal laut, terdapat data *input* dari *data store* pelabuhan berangkat, *data store* pelabuhan tiba, *data store voyage* kapal, *data store line* kapal, *data store* pelabuhan, *data store* kapal laut yang masing-masing memberikan data *output* pelabuhan berangkat, data *output* pelabuhan tiba, data *output voyage* kapal laut, data *output line* kapal, data *output* pelabuhan, dan data *output* kapal laut ke dalam proses. Proses ini mengeluarkan *output* data ambil jadwal kapal laut pelayaran yang kemudian dilakukan *split* dan diberikan pada proses menampilkan daftar kapal laut, proses menampilkan daftar jadwal *tremper*, proses menampilkan data pelabuhan, dan proses mengambil data jadwal kapal laut untuk *shipping*.

5. DFD Level 1 Memberikan Uang Tambang

Gambar 3.7, merupakan DFD level 1 hasil *decompose* proses memberikan uang tambang dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut. Terdapat 2 *external entity* pada proses ini yaitu *external entity* bagian *marketing* dan *external entity* *chief officer*. Terdapat 16 proses pada proses memberikan uang tambang yaitu proses menampilkan detail *shipping*urut barang berbahaya, proses menampilkan detail *shipping*urut jenis peti kemas, proses menampilkan detail *shipping*urut ukuran peti kemas, proses menampilkan detail *shipping*urut semua berdasarkan *id shipping*, proses menampilkan *booking of cargo*, proses menampilkan pesan, proses menampilkan daftar *shipping*, proses menampilkan *load list summary*, proses menampilkan *load list*, proses menampilkan detail *shipping*urut semua berdasarkan *id* jadwal, proses memasukkan data log pesan, proses memilih muat peti kemas, proses menghapus *list booking of cargo*, dan proses mengambil data *shipping* untuk *freight* pelayaran. Terdapat juga 2 *data store* dalam proses ini yaitu *data store* log pesan dan *data store shipping*.

Proses menampilkan detail *shipping*urut barang berbahaya mempunyai beberapa data *input* diantaranya adalah data *input split id shipping* untuk detail *shipping*urut barang berbahaya dari *external entity* bagian *marketing* dan data *input split* data detail *shipping*urut barang berbahaya dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* dari proses ini adalah detail *shipping*urut barang berbahaya yang ditujukan pada *external entity* bagian *marketing*. Proses menampilkan detail *shipping*urut jenis peti kemas mempunyai

beberapa data *input* diantaranya adalah data *input split id shipping* untuk detail *shipping*urut jenis peti kemas dari *external entity* bagian *marketing* dan data *input split* data *shipping* untuk detail *shipping*urut jenis peti kemas dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* dari proses ini adalah detail *shipping*urut jenis peti kemas untuk *external entity* bagian *marketing*.

Proses menampilkan detail *shipping*urut ukuran peti kemas mempunyai beberapa data *input* diantaranya adalah data *input split id shipping* untuk detail *shipping*urut ukuran peti kemas dari *external entity* bagian *marketing* dan data *input split* data *shipping* untuk detail *shipping*urut jenis peti kemas dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* dari proses ini adalah detail *shipping*urut ukuran peti kemas yang ditujukan untuk *external entity* bagian *marketing*.

Proses menampilkan detail *shipping*urut semua berdasarkan *id shipping* mempunyai beberapa data *input* diantaranya adalah data *input split id shipping* untuk detail *shipping*urut semua berdasarkan *id shipping* dan data *input split* data *shipping* untuk detail *shipping*urut semua berdasarkan *id shipping* dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* yang dihasilkan oleh sistem ini adalah data *output* detail *shipping*urut semua berdasarkan *id shipping* yang ditujukan untuk bagian *marketing*.

Proses menampilkan *booking of cargo* mempunyai data *input split* data *shipping* untuk *booking of cargo* dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* yang dihasilkan oleh proses ini adalah data *output*

loading of cargo yang ditujukan untuk *external entity* bagian *marketing* dan *external entity chief officer*. Proses menampilkan pesan mempunyai beberapa data *input* yaitu data *input log* pesan dari *data store log* pesan dan data *input split id* jadwal untuk pesan dari *external entity* bagian *marketing*. Data *output* yang dihasilkan proses ini adalah data *output log* pesan yang ditujukan untuk *external entity* bagian *marketing* dan *external entity chief officer*.

Proses menampilkan daftar *shipping* mempunyai beberapa data *input* diantaranya adalah data *input split id* jadwal untuk daftar *shipping* dari *external entity* bagian *marketing* dan data *input split* data untuk daftar *shipping* dari proses mengambil data untuk daftar *shipping*. Data *output* yang dihasilkan proses ini adalah data *output* daftar *shipping* yang ditujukan untuk *external entity* bagian *marketing* dan *external entity chief officer*.

Proses menampilkan *load list summary* mempunyai beberapa data *input* diantaranya adalah data *input split id* jadwal untuk *load list summary* dari *external entity* bagian *marketing* dan data *input split* data *shipping* untuk *load list summary* dari proses mengambil data *shipping* untuk *load list summary*. Data *output* yang dihasilkan proses ini adalah data *load list summary* yang ditujukan untuk *external entity* bagian *marketing* dan *external entity chief officer*. Proses menampilkan *load list* mempunyai beberapa data *input* diantaranya adalah data *input split id* jadwal untuk *load list* dari *external entity* bagian *marketing* dan data *input split shipping* untuk *load list* dari proses mengambil data *shipping* untuk *freight* pelayaran. Data *output* yang dihasilkan oleh proses ini adalah data *output load list* yang ditujukan untuk *external entity* bagian *marketing*.

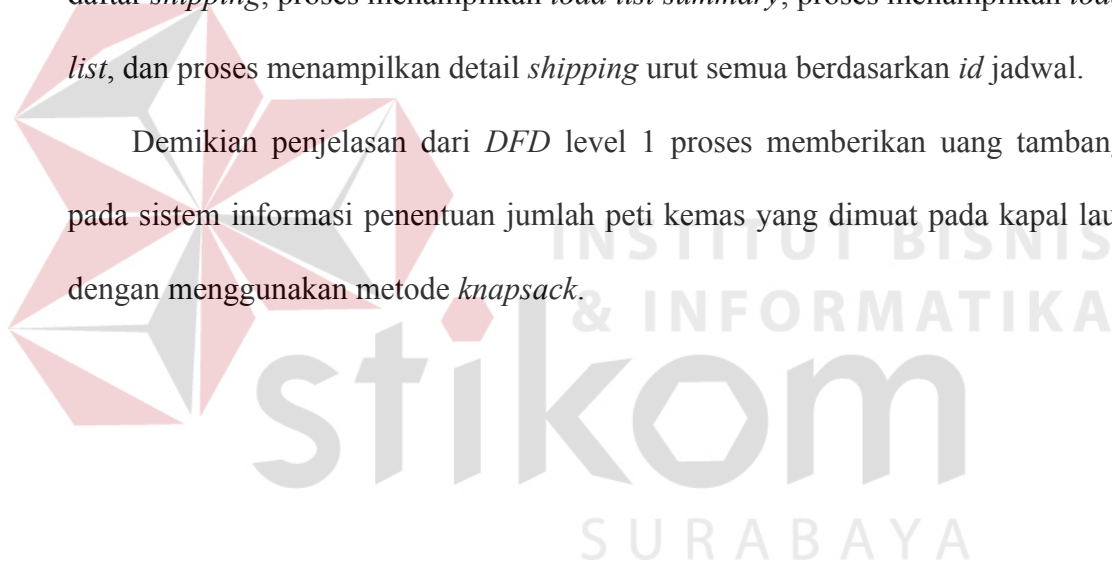
Proses menampilkan data detail *shipping*urut semua berdasarkan *id* jadwal mempunyai beberapa data *input* diantaranya adalah data *input split id* jadwal untuk detail *shipping*urut semua berdasarkan *id* jadwal dari *external entity chief officer* dan data *input split* untuk detail *shipping*urut semua berdasarkan *id* jadwal. Data *output* yang dihasilkan oleh proses ini adalah data *output* detail *shipping*urut semua berdasarkan *id* jadwal yang ditujukan untuk *external entity chief officer*. Proses memasukkan data log pesan mempunyai data *input* log pesan dari *external entity* bagian *marketing* dan *external entity chief officer*.

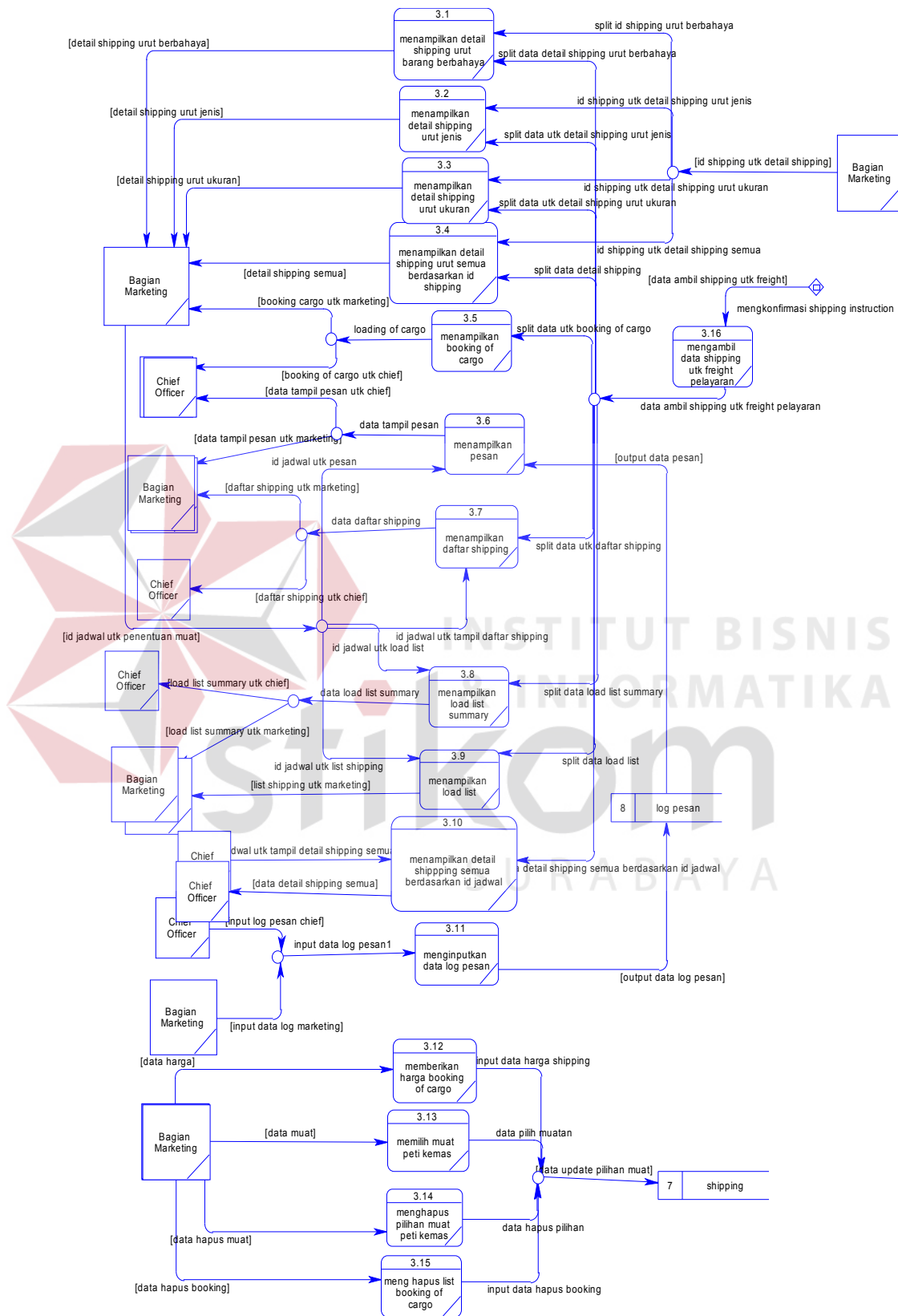
Data *output* yang dihasilkan adalah data *output log* pesan yang ditujukan untuk data store *log* pesan. Proses memberikan harga pada *booking of cargo* mempunyai data *input* harga dari *external entity* bagian *marketing*. Data *output* yang dihasilkan pada proses ini adalah data *output* harga *shipping* yang ditujukan untuk data store *shipping*. Proses memilih muat peti kemas mempunyai data *input* muat peti kemas yang diberikan oleh *external entity* bagian *marketing*. Data *output* yang dihasilkan proses ini adalah data *output* pilih muat yang ditujukan untuk data store *shipping*. Proses menghapus pilihan muat peti kemas mempunyai data *input* hapus muat yang diberikan oleh *external entity* bagian *marketing*.

Data *output* yang dihasilkan proses ini adalah data *output* hapus pilihan yang ditujukan untuk data store *shipping*. Proses menghapus *list booking of cargo* mempunyai data *input* booking yang diberikan oleh *external entity* bagian *marketing*. Data *output* yang dihasilkan proses ini adalah data *output* hapus booking yang ditujukan untuk data store *shipping*. Proses mengambil data *shipping* untuk *freight* pelayaran mempunyai data *input* ambil *shipping* untuk

freight dari proses melakukan konfirmasi *shipping instruction*. Data *output* yang dihasilkan proses ini adalah data *shipping* untuk freight pelayaran yang akan dilakukan *split* ke masing-masing proses yaitu proses menampilkan detail *shipping*urut barang berbahaya, proses menampilkan detail *shipping*urut jenis peti kemas, proses menampilkan detail *shipping*urut ukuran peti kemas, proses menampilkan detail *shipping*urut semua berdasarkan *id shipping*, proses menampilkan *booking of cargo*, proses menampilkan pesan, proses menampilkan daftar *shipping*, proses menampilkan *load list summary*, proses menampilkan *load list*, dan proses menampilkan detail *shipping*urut semua berdasarkan *id* jadwal.

Demikian penjelasan dari *DFD* level 1 proses memberikan uang tambang pada sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*.





Gambar 3.7 DFD Level 1 Memberikan Uang Tambang

6. DFD Level 1 Mengonfirmasi Shipping Instruction

Gambar 3.8, merupakan DFD level 1 hasil decompose proses melakukan konfirmasi *shipping instruction* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Terdapat *external entity shipper* yang memberikan data *input* dari dan ke proses ini. Terdapat 12 proses pada DFD level 1 ini diantaranya adalah proses menyimpan data *shipper*, proses menyimpan data *consignee*, proses menyimpan data *notify party*, proses menyimpan data *container shipper*, proses menampilkan data cari *shipper*, proses menampilkan daftar jenis *container*, proses menampilkan data cari *container*, proses menampilkan data cari jadwal kapal laut, proses menampilkan daftar *container shipper*, proses mengambil data *shipping* untuk *freight*, dan proses mengambil data transaksi *shipping*. Terdapat 5 data store pada proses ini yaitu data store *shipper*, data store *consignee*, data store *shipping*, data store *notify party*, dan data store *shipping container*.

External entity shipper memberikan data *input shipping instruction* yang kemudian dibagikan pada masing-masing proses yaitu proses menyimpan data *shipper*, proses menyimpan data *consignee*, proses menyimpan data *notify party*, proses menyimpan data *shipping container*, dan proses menyimpan data *shipping*. Proses menyimpan data *shipper* mempunyai data *input shipper* yang diberikan *external entity shipper*. Data *output* dari proses ini adalah data *output shipper* yang ditujukan pada data store *shipper*. Proses menyimpan data *consignee* mempunyai data *input consignee* yang diberikan *external entity shipper*. Data *output* dari proses ini adalah data *output consignee* yang dtujukan pada data store

consignee. Proses menyimpan data *notify party* mempunyai data *input notify party* yang diberikan *external entity shipper*. Data *output* dari proses ini adalah data *output notify party* yang ditujukan pada data store *notify party*. Proses menyimpan data *shipping* mempunyai data *input shipping* yang diberikan pada *external entity shipper*. Data *output* dari proses ini adalah data *output shipping* yang ditujukan pada data store *shipping*.

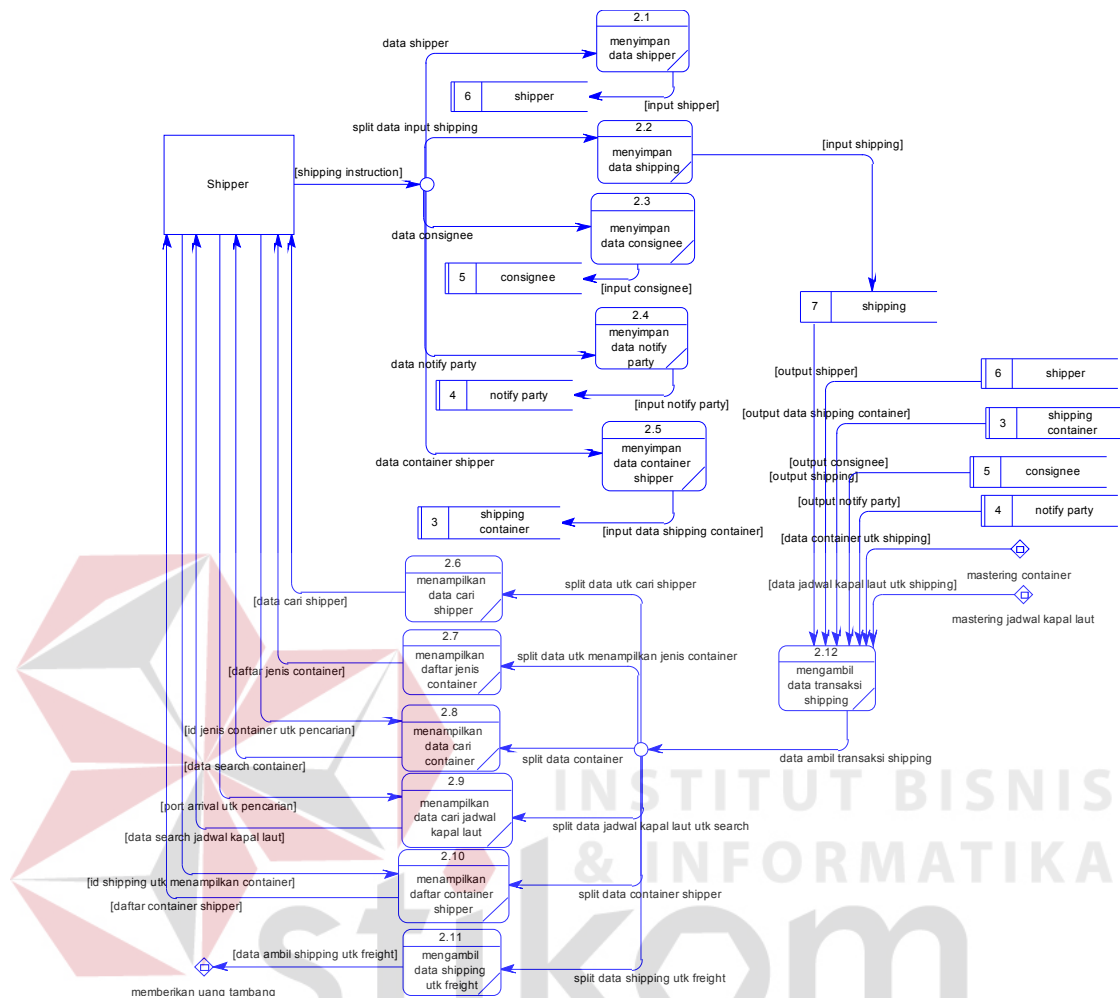
Proses menyimpan data *shipping container* mempunyai data *input shipping container* yang diberikan pada *external entity shipper*. Data *output* dari proses ini adalah data *output shipping container* yang ditujukan pada data store *shipping container*. Proses menampilkan data cari *shipper* mempunyai data *input split data shipping* untuk cari *shipper* dari proses mengambil data transaksi *shipping*. Data *output* dari proses ini adalah data *output cari shipper* yang ditujukan untuk *external entity shipper*. Proses menampilkan data jenis *container* mempunyai data *input split data shipping* untuk jenis *container* dari proses mengambil data transaksi *shipping*. Data *output* dari proses ini adalah data *output jenis container* yang ditujukan untuk *external entity shipper*.

Proses menampilkan data cari *container* mempunyai beberapa data *input* diantaranya adalah data *input cari container* dari *external entity shipper* dan data *input split shipping* untuk cari *container* dari proses mengambil data transaksi *shipping*. Data *output* yang dihasilkan proses ini adalah data *output cari container* yang ditujukan untuk *external entity shipper*. Proses menampilkan data cari jadwal kapal laut mempunyai beberapa data *input* diantaranya adalah data *input cari jadwal kapal laut* dari *external entity shipper* dan data *input split shipping*

untuk cari jadwal kapal laut dari proses mengambil data *shipping* untuk cari jadwal kapal laut. Data *output* dari proses ini adalah data *output* cari jadwal kapal laut yang ditujukan untuk *external entity shipper*.

Proses menampilkan data *container shipper* mempunyai beberapa data *input* diantaranya adalah data *input id shipping* untuk *container shipper* yang diberikan *external entity shipper* dan data *input split shipping* untuk *container shipper* yang diberikan oleh proses mengambil data transaksi *shipping*. Data *output* dari proses ini adalah data *output container shipper* yang ditujukan untuk *external entity shipper*. Proses mengambil data *shipping* untuk *freight* mempunyai data *input* berupa data *split shipping* untuk *freight* dari proses mengambil data *shipping*. Data *output* yang dihasilkan oleh proses ini adalah data *output shipping* untuk *freight* yang ditujukan untuk proses memberikan uang tambang.

Proses mengambil data transaksi *shipping* mempunyai beberapa data *input* diantaranya adalah data *input shipper* dari data store *shipper*, data *input container* dari data store *container*, data *input notify party* dari data store *notify party*, data *input shipping container* dari data store *shipping container*, dan data *input shipping* dari data store *shipping*. Data *output* dari proses ini adalah data *output* ambil transaksi *shipping* yang akan dibagikan ke masing-masing proses yaitu proses menampilkan data cari *shipper*, proses menampilkan daftar jenis *container*, proses menampilkan data cari *container*, proses menampilkan data cari jadwal kapal laut, proses menampilkan daftar *container shipper*, dan proses mengambil data *shipping* untuk *freight*.

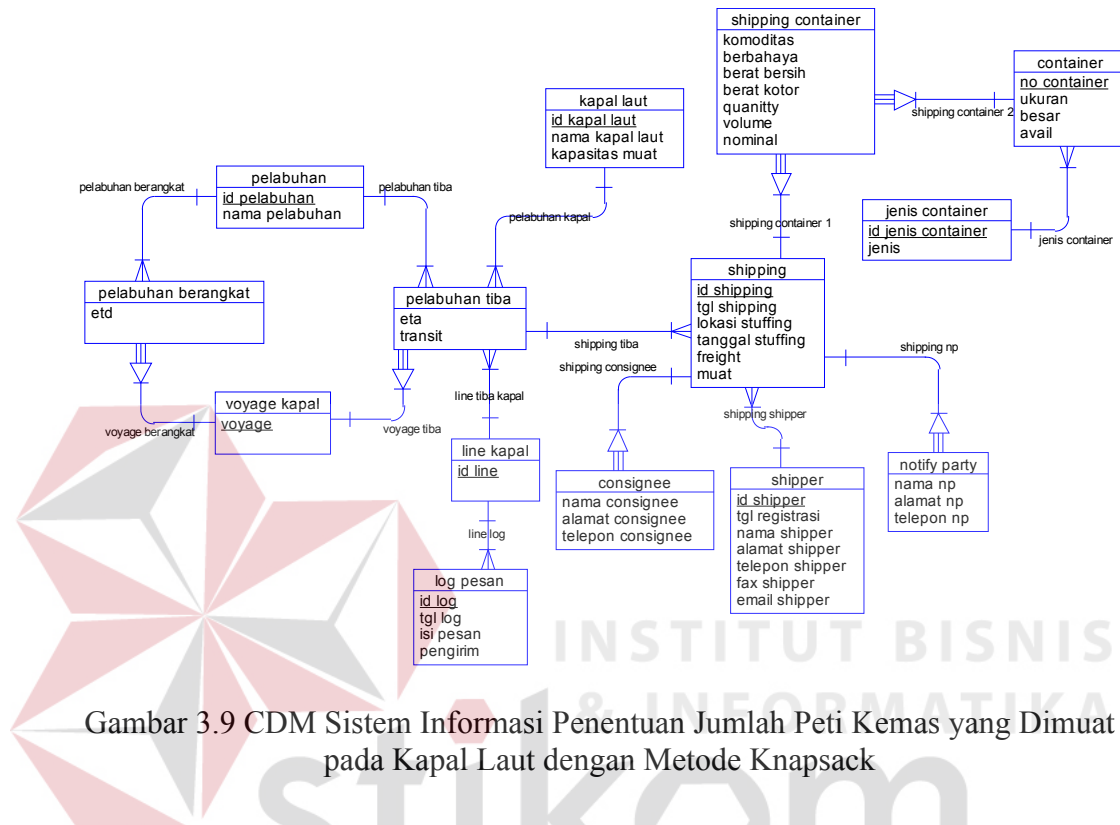


Gambar 3.8 DFD Level 1 Melakukan Konfirmasi Shipping Instruction

3.2.3 Entity Relationship Diagram (ERD)

Setelah dilakukannya analisis proses-proses yang dibutuhkan oleh sistem, maka langkah selanjutnya adalah melakukan perancangan *database* sebagai tempat penyimpanan data-data yang diinputkan oleh *user* ke dalam sistem. Adapun tahap-tahapan dalam perancangan *database* adalah dimulai dari *Conceptual Data Model* (CDM), kemudian *Physical Data Model* (PDM), lalu menjadi *query sql* terstruktur yang siap untuk diimplementasikan ke dalam sistem.

1. Conceptual Model Diagram (CDM) Sistem Informasi Penentuan Jumlah Peti Kemas yang Dimuat pada Kapal Laut dengan Metode Knapsack



Gambar 3.9 CDM Sistem Informasi Penentuan Jumlah Peti Kemas yang Dimuat pada Kapal Laut dengan Metode Knapsack

Gambar 3.9, merupakan CDM dari *database* anugerah_ds dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Terdapat 14 tabel dalam CDM ini yaitu tabel pelabuhan, tabel pelabuhan berangkat, tabel pelabuhan tiba, tabel *voyage* kapal, tabel kapal laut, tabel *line* kapal, tabel *log* pesan, tabel *shipping*, tabel *consignee*, tabel *notify party*, tabel *shipper*, tabel *shipping container*, tabel *container*, dan tabel jenis *container*. Tabel pelabuhan berangkat mempunyai relasi *many to one* terhadap tabel pelabuhan, dengan nama relasi pelabuhan berangkat. Tabel pelabuhan berangkat juga mempunyai relasi *many to one* dengan tipe *dependent* terhadap tabel *voyage* kapal laut, dengan nama relasi *voyage* berangkat. Pelabuhan

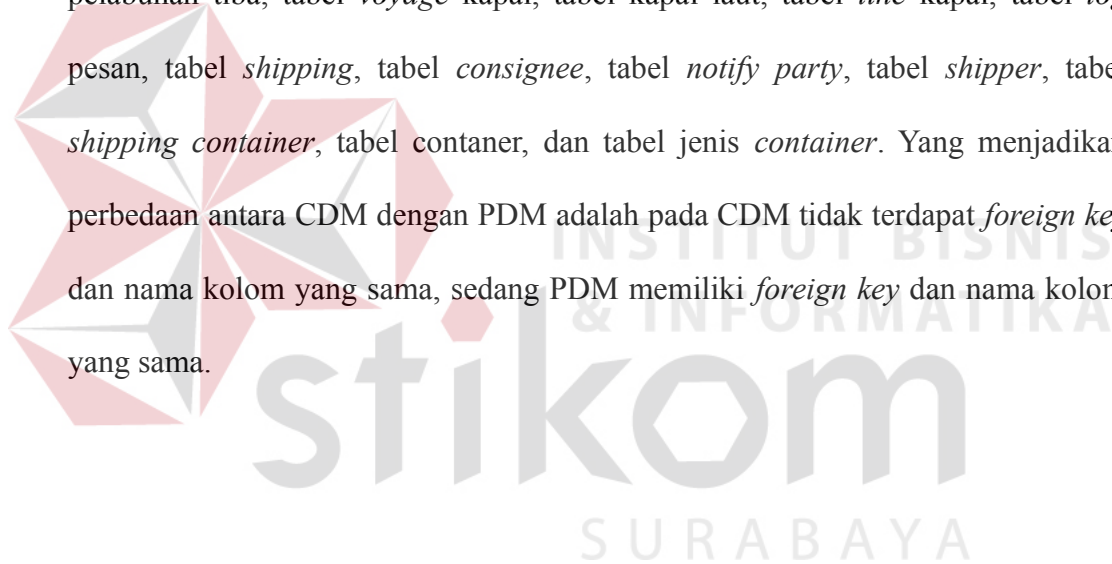
tiba mempunyai relasi *many to one* terhadap tabel pelabuhan, dengan nama relasi pelabuhan tiba. Pelabuhan tiba juga mempunyai relasi *many to one* bertipe *dependent* terhadap tabel *voyage* kapal, dengan nama relasi *voyage* tiba. Pelabuhan tiba mempunyai relasi *many to one* terhadap tabel kapal laut, dengan nama relasi pelabuhan kapal. Tabel pelabuhan tiba mempunyai relasi *many to one* terhadap tabel *line* kapal, dengan nama relasi *line* tiba kapal.

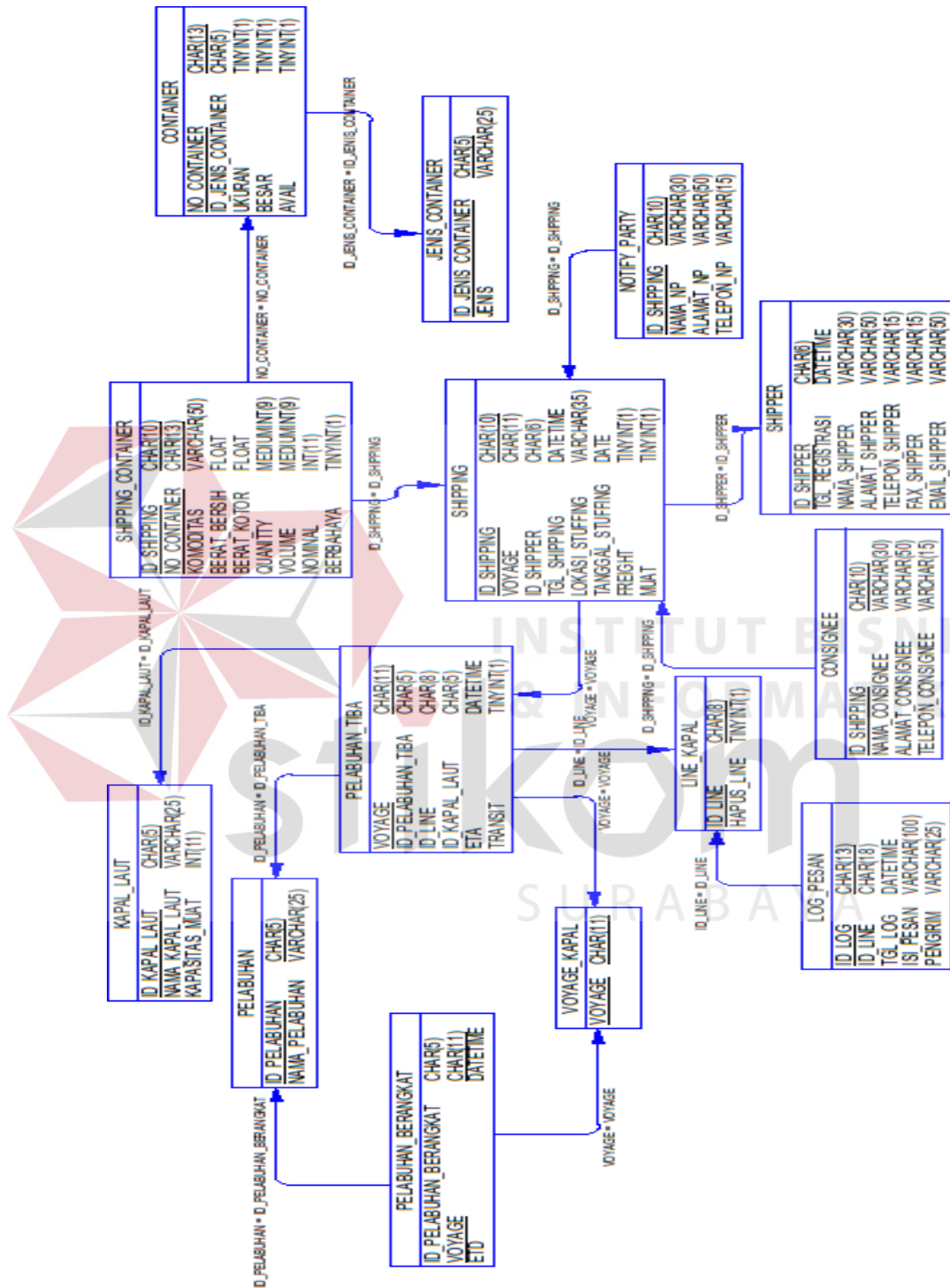
Tabel *log* pesan mempunyai relasi *many to one* terhadap tabel *line* kapal, dengan nama relasi *line log*. Tabel *shipping* mempunyai relasi *many to one* terhadap tabel pelabuhan tiba, dengan nama relasi *line* tiba kapal. Tabel *shipping* juga mempunyai relasi *many to one* terhadap tabel *shipper*, dengan nama relasi *shipper shipper*. Tabel *consignee* mempunyai relasi *many to one* bertipe *dependent* terhadap tabel *shipping*, dengan nama relasi *shipping consignee*.

Tabel *notify party* mempunyai relasi *many to one* bertipe *dependent* terhadap tabel *shipping*, dengan nama relasi *shipping np*. Tabel *shipping container* mempunyai relasi *many to one* bertipe *dependent* terhadap tabel *shipping*, dengan nama relasi *shipping container 1*. Tabel *container* mempunyai relasi *many to one* terhadap tabel jenis *container*, dengan nama relasi jenis *container*. Demikian penjelasan dari *cdm* mengenai relasi antar tabel pada sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*.

2. Physical Data Model (PDM) Sistem Informasi Penentuan Jumlah Peti Kemas yang dimuat Pada Kapal Laut dengan Metode Knapsack

Gambar 3.10, merupakan hasil generate CDM ke PDM, dan tabel-tabel tersebut sudah siap untuk diimplementasikan ke dalam sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan *knapsack*. Jumlah tabel dari CDM ke PDM adalah sama yaitu berjumlah 14 buah, yaitu diantaranya adalah tabel pelabuhan, tabel pelabuhan berangkat, tabel pelabuhan tiba, tabel *voyage* kapal, tabel kapal laut, tabel *line* kapal, tabel *log* pesan, tabel *shipping*, tabel *consignee*, tabel *notify party*, tabel *shipper*, tabel *shipping container*, tabel *container*, dan tabel jenis *container*. Yang menjadikan perbedaan antara CDM dengan PDM adalah pada CDM tidak terdapat *foreign key* dan nama kolom yang sama, sedang PDM memiliki *foreign key* dan nama kolom yang sama.





Gambar 3.10 PDM Sistem Informasi Penentuan Jumlah Peti Kemas yang dimuat Pada Kapal Laut dengan Metode Knapsack

3.2.4 Database Structure

1. PELABUHAN

Fungsi : menyimpan data pelabuhan

Primary key : ID_PELABUHAN

Foreign key : -

Tabel 3.1 PELABUHAN

No.	Column	Type	Size	Key
1	ID_PELABUHAN	CHAR	5	<i>Primary key</i>
2	NAMA_PELABUHAN	VARHCHAR	25	-

2. VOYAGE_KAPAL

Fungsi : menyimpan data *voyage* kapal laut

Primary key : VOYAGE

Foreign key : -

Tabel 3.2 VOYAGE_KAPAL

No.	Column	Type	Size	Key
1	VOYAGE	CHAR	11	<i>Primary key</i>

3. PELABUHAN_BERANGKAT

Fungsi : menyimpan data pelabuhan berangkat kapal laut

Primary key : VOYAGE

Foreign key : ID_PELABUHAN_BERANGKAT

Tabel 3.3 PELABUHAN_BERANGKAT

No.	Column	Type	Size	Key
1	VOYAGE	CHAR	11	<i>Primary key</i>
2	ID_PELABUHANG_BERANGKAT	CHAR	5	<i>Foreign key</i>
3	E.T.D	DATETIME	-	-

4. KAPAL_LAUT

Fungsi : menyimpan data kapal laut

Primary key : ID_KAPAL_LAUT

Foreign key : -

Tabel 3.4 KAPAL LAUT

No.	Column	Type	Size	Key
1	ID_KAPAL_LAUT	CHAR	5	<i>Primary key</i>
2	NAMA_KAPAL_LAUT	VARCHAR	25	-
3	KAPASITAS_MUAT	INT	11	-

5. LINE_KAPAL

Fungsi : menyimpan data jalur pelabuhan keberangkatan kapal laut

Primary key : ID_LINE

Foreign key : -

Tabel 3.5 LINE_KAPAL

No.	Column	Type	Size	Key
1	ID_LINE	CHAR	8	<i>Primary key</i>
2	HAPUS_LINE	TINYINT	1	-

6. PELABUHAN_TIBA

Fungsi : menyimpan data pelabuhan tiba kapal laut

Primary key : VOYAGE

Foreign key : ID_PELABUHAN, ID_LINE, ID_KAPAL_LAUT

Tabel 3.6 PELABUHAN TIBA

No.	Column	Type	Size	Key
1	VOYAGE	CHAR	11	Primary key
2	ID_PELABUHAN_TIBA	CHAR	5	Foreign key
3	ID_LINE	CHAR	8	Foreign key
4	ID_KAPAL_LAUT	CHAR	5	Foreign key
5	ETA	DATETIME	-	-
6	TRANSIT	TINYINT	1	-

7. LOG_PESAN

Fungsi : menyimpan data *log* pesan.

Primary key : ID_LOG_PESAN

Foreign key : ID_LINE

Tabel 3.7 LOG PESAN

No.	Column	Type	Size	Key
1	ID_LOG_PESAN	CHAR	13	Primary key
2	ID_LINE	CHAR	8	Foreign key
3	TGL_LOG	DATETIME	-	
4	ISI_PESAN	VARCHAR	100	
5	PENGIRIM	VARCHAR	25	

8. SHIPPER

Fungsi : menyimpan data *shipper*

Primary key : ID_SHIPPER

Foreign key : -

Tabel 3.8 SHIPPER

No.	Column	Type	Size	Key
1	ID_SHIPPER	CHAR	6	Primary key
2	TGL_REGISTRASI	DATETIME	-	
3	NAMA_SHIPPER	VARCHAR	30	
4	ALAMAT_SHIPPER	VARCHAR	50	
5	TELEPON_SHIPPER	VARCHAR	15	
6	FAX_SHIPPER	VARCHAR	15	
7	EMAIL_SHIPPER	VARCHAR	50	

9. SHIPPING

Fungsi : menyimpan data transaksi *shipping*

Primary key : ID_SHIPPING

Foreign key : VOYAGE, ID_SHIPPER

TABEL 3.9 SHIPPING

No.	Column	Type	Size	Key
1	ID_SHIPPING	CHAR	10	Primary key
2	VOYAGE	CHAR	11	Foreign key
3	ID_SHIPPER	CHAR	6	Foreign key
4	TGL_SHIPPING	DATETIME	-	-
5	LOKASI STUFFING	VARCHAR	35	-
6	TANGGAL_STUFFING	DATE	-	-
7	FREIGHT	TINYINT	1	-
8	MUAT	TINYINT	1	-

10. CONSIGNEE

Fungsi : menyimpan data *consignee*

Primary key : ID_SHIPPING

Foreign key : -

Tabel 3.3.10 CONSIGNEE

No.	Column	Type	Size	Key
1	ID_SHIPPING	CHAR	10	Primary key
2	NAMA_CONSIGNEE	VARCHAR	30	-
3	ALAMAT_CONSIGNEE	VARCHAR	50	-
4	TELEPON_CONSIGNEE	VARCHAR	15	-

11. NOTIFY_PARTY

Fungsi : menyimpan data *notify party*

Primary key : ID_SHIPPING

Foreign key : -

Tabel 3.11 NOTIFY_PARTY

No.	Column	Type	Size	Key
1	ID_SHIPPING	CHAR	10	Primary key
2	NAMA_NP	VARCHAR	30	-
3	ALAMAT_NP	VARCHAR	50	-
4	TELEPON_NP	VARCHAR	15	-

12. JENIS_CONTAINER

Fungsi : menyimpan data jenis *container*.

Primary key : ID_JENIS_CONTAINER

Foreign key : -

Tabel 3.12 JENIS_CONTAINER

No.	Column	Type	Size	Key
1	ID_JENIS_CONTAINER	CHAR	5	Primary key
2	JENIS	VARCHAR	25	-

13. CONTAINER

Fungsi : menyimpan data *container*.

Primary key : NO_CONTAINER

Foreign key : ID_JENIS_CONTAINER

Tabel 3.13 CONTAINER

No.	Column	Type	Size	Key
1	NO_CONTAINER	CHAR	13	<i>Primary key</i>
2	ID_JENIS_CONTAINER	CHAR	5	<i>Foreign key</i>
3	UKURAN	TINYINT	1	-
4	BESAR	TINYINT	1	-
5	AVAIL	TINYINT	1	-

14. SHIPPING_CONTAINER

Fungsi : menyimpan data *shipping container*

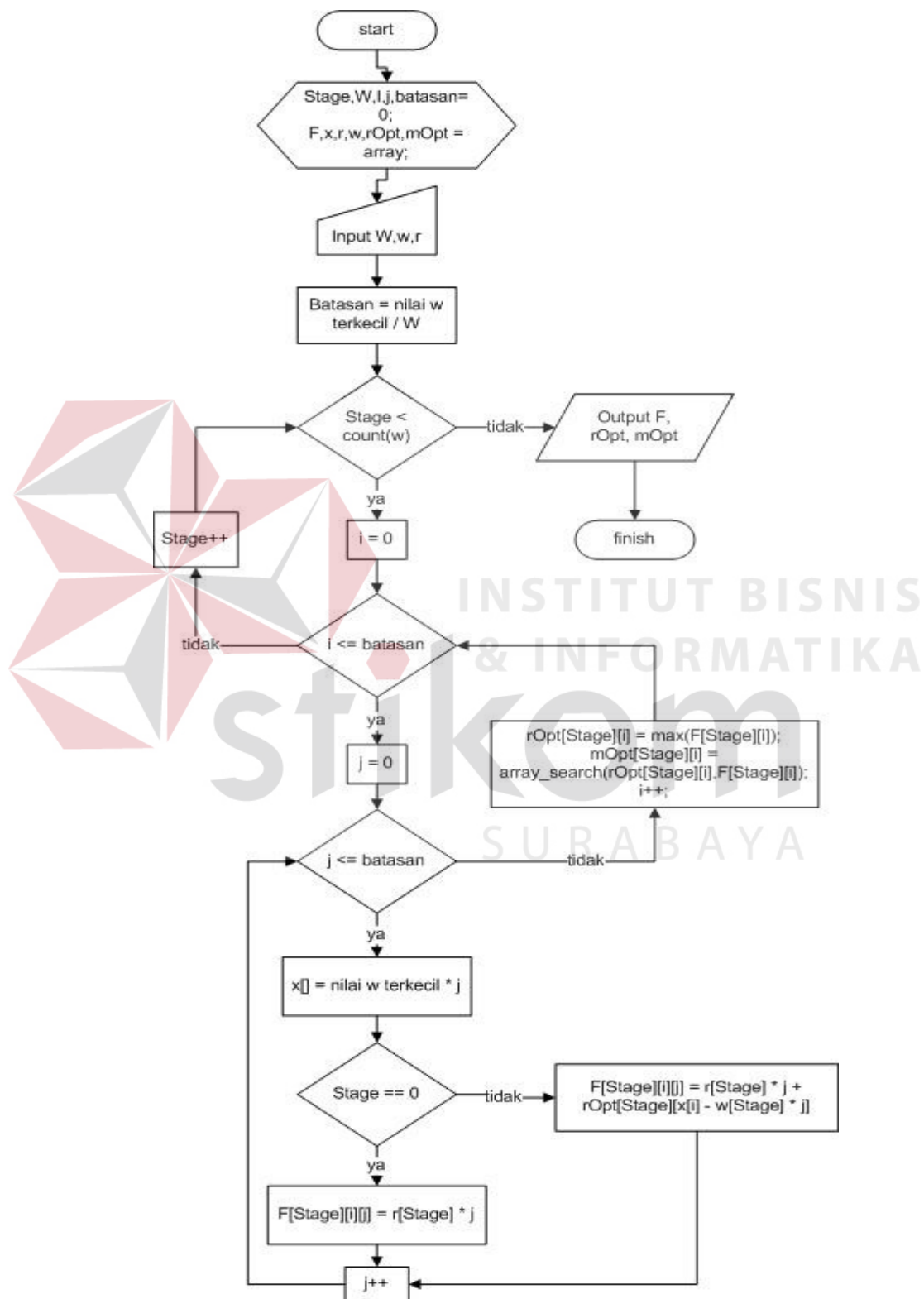
Primary key : ID_SHIPPING, NO_CONTAINER

Foreign key : -

Tabel 3.14 SHIPPING_CONTAINER

No.	Column	Type	Size	Key
1	ID_SHIPPING	CHAR	10	<i>Primary key</i>
2	NO_CONTAINER	CHAR	13	<i>Primary key</i>
3	KOMODITAS	VARCHAR	50	-
4	BERAT_BERSIH	FLOAT	-	-
5	BERAT_KOTOR	FLOAT	-	-
6	QUANTITY	MEDIUMINT	9	-
7	VOLUME	MEDIUMINT	9	-
8	NOMINAL	INT	11	-
9	BERBAHAYA	TINYINT	1	-

3.2.5 Flowchart Perhitungan Knapsack



Gambar 3.11 Flowchart Perhitungan Knapsack

Gambar 3.11 merupakan alur proses perhitungan *knapsack* untuk menentukan berat optimal yang dapat dimuat pada suatu *container* (wadah) dengan kapasitas yang terbatas. Pertama-tama menginisialisasi variabel W , i , dan j bernilai 0, kemudian menginisialisasi variabel F , x , r , w , $rOpt$, dan $mOpt$ bertipe *array*. Langkah selanjutnya memasukkan nilai pada variabel W , w , r dengan ketentuan W mewakili batas kapasitas muat suatu barang, w mewakili berat suatu *item*, dan r mewakili nilai atau harga suatu *item*.

Lalu memasuki proses pencarian berat terkecil dari suatu *item* untuk dikalikan dengan kapasitas dan dimasukkan pada variabel batasan. Terdapat 3 buah *looping* (perulangan) yaitu *looping* untuk *stage*, *looping* untuk alokasi jumlah muat barang, dan *looping* untuk alokasi berat muat. *Looping* untuk *stage* dilakukan sampai variabel *stage* hampir mendekati dari jumlah *item* barang yang dimasukkan. *Looping* untuk alokasi jumlah muat barang dan *looping* untuk alokasi berat muat dilakukan sampai nilai dari variabel i dan j sama dengan nilai dari variabel batasan.

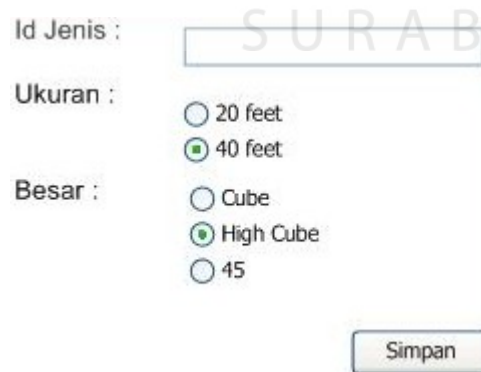
Langkah selanjutnya sampai pada proses perhitungan *knapsack* yang disajikan dalam 2 tahap, tahap pertama mempunyai syarat apabila variabel *stage* bernilai nol maka jalankan rumus sebagai berikut: $F[Stage][i][j] = r[Stage] * j$. Jika variabel *stage* tidak bernilai nol maka jalankan rumus sebagai berikut: $F[Stage][i][j] = r[Stage] * j + rOpt[Stage][x[i] - w[Stage] * j]$. Langkah yang terakhir dilakukan adalah mengambil nilai pendapatan optimal dan jumlah muat optimal dari variabel F . Nilai pendapatan optimal didapat dengan cara mencari nilai pada variabel $F[stage][i]$ yang terbesar dan ditampung pada variabel $rOpt$.

Sedangkan nilai dari jumlah muat optimal didapatkan dari pencarian variabel $rOpt[Stage][i]$ ke variabel $F[Stage][i]$ yang hasilnya ditampung dalam variabel $mOpt[Stage][i]$. Jika seluruh *stage* telah dijalankan maka hasil dari perhitungan *knapsack* dapat dilihat pada variabel F , $rOpt$, dan $mOpt$.

3.2.6 Desain Input/Output

Setelah membuat rancangan *database*, langkah berikutnya adalah mendesain *input/output*. Dalam mendesain *input/output* haruslah terdapat kesesuaian antara yang disimpan oleh *user* dan yang ditampilkan oleh sistem, dengan kolom-kolom suatu tabel yang terdapat dalam *database* yang telah dirancang, sehingga terjadi keselarasan. Adapun desain *input/output* yang harus terdapat dalam sistem penentuan jumlah peti kemas yang dimuat pada kapal laut dengan metode *knapsack* adalah sebagai berikut :

1. Desain Input Peti Kemas



The image shows a web form for entering container data. It has three main sections: 'Id Jenis' with a text input field, 'Ukuran' with two radio button options ('20 feet' and '40 feet', where '40 feet' is selected), and 'Besarnya' with three radio button options ('Cube', 'High Cube', and '45', where 'High Cube' is selected). A 'Simpan' button is located at the bottom right of the form.

Gambar 3.12 Desain Input Peti Kemas

Gambar 3.12 merupakan desain *input* peti kemas yang mempunyai fungsi untuk menyimpan data peti kemas ke dalam sistem. Data *input* peti kemas yang

dimasukkan ke dalam sistem antara lain adalah *id* jenis peti kemas, ukuran peti kemas yang mempunyai beberapa pilihan diantaranya adalah 20 *feet* dan 40 *feet*, serta besar peti kemas yang mempunyai pilihan *cube*, *high cube* dan 45.

2. Desain Input Jadwal Keberangkatan Kapal Laut

Id Kapal :

ETA : tgl / bln / thn jam : Mnt

ETD : tgl / bln / thn jam : Mnt

Pelabuhan Tiba :

Pelabuhan Berangkat :

Gambar 3.13 Desain Input Jadwal Keberangkatan Kapal Laut

Gambar 3.13 merupakan desain *input* jadwal keberangkatan kapal laut yang berfungsi untuk menyimpan data jadwal keberangkatan kapal laut ke dalam sistem. Data *input* jadwal keberangkatan kapal laut yang dimasukkan ke dalam sistem antara lain adalah *id* kapal laut, E.T.A (*Estimated Time Arrival*), E.T.D (*Estimated Time Departure*), pelabuhan tiba, dan pelabuhan berangkat. Desain *input* ini diperuntukkan untuk hak akses sebagai administrator untuk menambah data pada master jadwal keberangkatan kapal laut.

3. Desain Input Jenis Peti Kemas

Gambar 3.14 merupakan desain *input* jenis peti kemas yang mempunyai fungsi untuk menyimpan data jenis peti kemas ke dalam sistem. Data *input* jenis

peti kemas yang dimasukkan ke dalam sistem antara lain adalah jenis peti kemas. Desain *input* ini diperuntukkan untuk hak akses administrator agar dapat menambah data pada *master jenis container*.



Jenis :

Simpan

Gambar 3.14 Desain Input Jenis Container

4. Desain Input Kapal Laut

Gambar 3.15 merupakan desain *input* kapal laut yang mempunyai fungsi untuk menyimpan data kapal laut ke dalam sistem. Data *input* kapal laut yang dimasukkan ke dalam sistem antara lain adalah nama kapal laut dan kapasitas muat kapal laut dengan tipe *integer* dengan satuan ton. Desain *input* ini diperuntukkan untuk hak akses sebagai administrator untuk menambah data master kapal laut.



Nama :

Kapasitas : Ton

Simpan

Gambar 3.15 Desain Input Kapal Laut

5. Desain Input Pelabuhan

Gambar 3.16 merupakan desain *input* pelabuhan yang berfungsi untuk menyimpan data pelabuhan ke dalam sistem. Data *input* pelabuhan yang dimasukkan ke dalam sistem antara lain adalah nama pelabuhan.

Desain *input* ini diperuntukkan untuk hak akses sebagai administrator untuk dapat menambah data pada master pelabuhan.

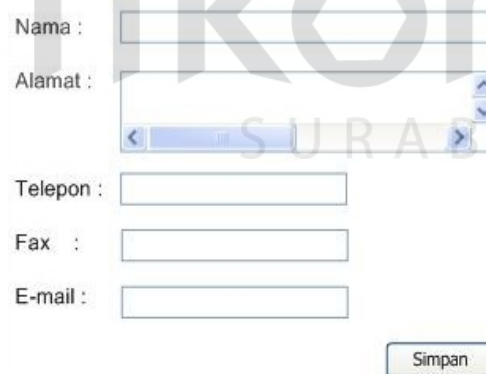


Nama Pelabuhan :

Gambar 3.16 Desain Input Pelabuhan

6. Desain Input Shipper

Gambar 3.17 merupakan desain *input shipper* yang mempunyai fungsi untuk menyimpan data *shipper* ke dalam sistem. Data *input shipper* yang dimasukkan ke dalam sistem antara lain adalah nama, alamat rumah, nomor telepon, nomor *fax*, dan alamat *e-mail*. Desain input ini diperuntukkan untuk hak akses sebagai administrator untuk dapat menambah data pada master *shipper*.



Nama :

Alamat :

Telepon :

Fax :

E-mail :

Gambar 3.17 Desain Input Shipper

7. Desain Input Detail Shipping

Gambar 3.18 merupakan desain *input detail shipping* yang mempunyai fungsi untuk menyimpan data detail *shipping* ke dalam sistem. Data *input detail shipping*

yang dimasukkan ke dalam sistem antara lain adalah nomor *container*, komoditas, berat bersih dengan tipe *float* dengan satuan ton, berat kotor dengan tipe *float* dengan satuan ton, *quantity* dengan tipe *integer* dengan satuan *PCS (Packages)*, volume bertipe *integer* dengan satuan m^3 , dan berbahaya yang mempunyai pilihan ya dan tidak.

Id Shipping :

No. Container :

Komoditas :

Berat Bersih : ton

Berat Kotor : ton

Quantity : Pcs

Volume : m3

Berbahaya : Ya Tidak

INSTITUT BISNIS INFORMATIKA
stikom
SURABAYA

Gambar 3.18 Desain Input Detail Shipping

8. Desain Input Shipping

Gambar 3.19 merupakan desain *input shipping* yang mempunyai fungsi untuk menyimpan data *shipping*. Data *input shipping* yang dimasukkan ke dalam sistem antara lain adalah *id shipper*, nama *consignee*, alamat rumah *consignee*, nomor telepon *consignee*, nama *notify party*, alamat rumah *notify party*, nomor telepon *notify party*, nomor *voyage*, tanggal stuffing, lokasi stuffing, dan *freight* yang mempunyai pilihan berupa *prepaid* dan *payable*.

The image shows a web form for shipping input, divided into four main sections:

- Shipper:** A single text input field labeled "Id Shipper".
- Consignee:** Fields for "Nama", "Alamat" (with a multi-line address editor), and "Telepon".
- Notify Party:** A checkbox "Same as consignee", followed by "Nama", "Alamat" (with a multi-line address editor), and "Telepon".
- Input shipping:** Fields for "Voyage", "tgl. stuffing:" (with separate boxes for day, month, and year), "Lokasi stuffing:" (with a multi-line address editor), and "Freight:" (with radio buttons for "prepaid" and "payable").

A "Simpan" button is located at the bottom right of the form. A large watermark for "STIKOM SURABAYA" is overlaid on the form.

Gambar 3.19 Desain Input Shipping

9. Desain Output Peti Kemas

Gambar 3.20 merupakan desain *output* peti kemas. Desain ini mempunyai fungsi untuk menampilkan data peti kemas. Data yang ditampilkan pada desain ini adalah nomor *container*, jenis peti kemas, ukuran peti kemas, dan besar peti

kemas. Desain *output* ini diperuntukkan untuk hak akses sebagai *customer service* untuk dapat mengisi data pada *input shipping instruction*.

No. Container	Jenis	Ukuran	High Cube
Link			

Gambar 3.20 Desain Output Peti Kemas

10. Desain Output Booking of Cargo

Id jadwal	Notify	Kapal laut	ETA	ETD	Port Arrival	Port Departure
Link	*					

Gambar 3.21 Desain Output Booking of Cargo

Gambar 3.21 merupakan desain *output booking of cargo*. Desain ini mempunyai fungsi untuk menampilkan data *booking of cargo*. Data *output booking of cargo* terdiri dari *id* jadwal keberangkatan kapal laut, nama kapal laut, ETA, E.T.D, pelabuhan tiba, dan pelabuhan berangkat. Desain output ini diperuntukkan untuk bagian *marketing* dan *chief officer* untuk dapat menuju desain *output shipping container*.

11. Desain Output Shipper

Id Shipper	Nama	Alamat	Telepon
link			

Gambar 3.22 Desain Output Shipper

Gambar 3.22 merupakan desain *output shipper*. Desain mempunyai fungsi untuk menampilkan data *shipper*. Data *output shipper* terdiri dari *id shipper*, nama *shipper*, alamat *shipper*, dan nomor telepon *shipper*. Desain *output* ini diperuntukkan untuk hak akses sebagai *customer service* untuk dapat mengisi data *shipping instructions*.

12. Desain Output Jadwal Keberangkatan Kapal Laut

Nama Kapal :

E.T.A	E.T.D	Pelabuhan Tiba	Pelabuhan Berangkat

Gambar 3.23 Desain Output Jadwal Keberangkatan Kapal Laut

Gambar 3.23 merupakan desain *output* jadwal keberangkatan kapal laut. Desain ini mempunyai fungsi untuk menampilkan data jadwal keberangkatan kapal laut. Data *output* jadwal keberangkatan kapal laut terdiri dari nama kapal laut, E.T.A, E.T.D, pelabuhan tiba, pelabuhan berangkat. Desain *output* ini diperuntukkan untuk hak akses sebagai administrator untuk dapat mengisi data jadwal keberangkatan kapal laut.

13. Desain Output Muat Peti Kemas

No	Id Shipping	Jml. Cont.	Berat	Pendapatan
	link			
Total :		0 Item	0 Ton	Rp. 0

Gambar 3.24 Desain Output Muat Peti Kemas

Gambar 3.24 merupakan desain *output* muat peti kemas. Desain ini mempunyai fungsi untuk menampilkan data muat peti kemas. Data *output* muat peti kemas terdiri dari *id shipping*, jumlah peti kemas, berat peti kemas, dan pendapatan *shipping*. Desain *output* ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data uang tambang *shipping*.

14. Desain Output Detail Shipping

No	No. Container	Komoditas	Ukuran	Besar	Berbahaya	Quantity	Volume	Nominal	Berat
Total :							0 m3	Rp. 0,00	0 Ton

Gambar 3.25 Desain Output Detail Shipping

Gambar 3.25 merupakan desain *output* detail *shipping*. Desain *output* detail *shipping* mempunyai fungsi untuk menampilkan detail *shipping* kepada *user*. Data *output* detail *shipping* terdiri dari nomor *container*, komoditas, ukuran peti kemas, besar peti kemas, barang berbahaya, *quantity*, volume, nominal dan berat peti kemas. Desain *output* ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data *input* uang tambang.

15. Desain Output Shipping

Id Shipping	No. Voyage	Kapal Laut	Freight	Tujuan	Jml. Container
Link					

Gambar 3.26 Desain Output Shipping

Gambar 3.26 merupakan desain *output shipping*. Desain *output shipping* mempunyai fungsi untuk menampilkan data *output shipping* kepada *user*. Data *output shipping* terdiri dari *id shipping*, nomor *voyage*, nama kapal laut, *freight*, tujuan pelabuhan kapal laut, jumlah muat peti kemas pada kapal laut. Desain output ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data transaksi uang tambang.

16. Desain Output Peti Kemas Berdasarkan Urut Barang Berbahaya

Gambar 3.27 merupakan desain *output* peti kemas berdasarkan urut barang berbahaya. Desain *output* peti kemas urut barang berbahaya mempunyai fungsi untuk menampilkan data peti kemas berdasarkan urut barang berbahaya kepada *user*. Data *output* peti kemas berdasarkan urut barang berbahaya terdiri dari barang berbahaya, jumlah muat peti kemas pada kapal laut, nominal *shipping*, dan berat muat peti kemas pada kapal laut. Desain *output* ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data transaksi uang tambang dan juga *chief officer* untuk dapat mengetahui data *order shipping*.

No	Berbahaya	Jumlah	Nominal	Berat

Total :	0 Item	Rp. 0,00	0 Ton
---------	--------	----------	-------

Gambar 3.27 Desain Output Peti Kemas Berdasarkan Urut Barang Berbahaya

17. Desain Output Peti Kemas Berdasarkan Urut Jenis

Gambar 3.28 merupakan desain *output* peti kemas berdasarkan urutan jenis. Desain *output* peti kemas berdasarkan urutan jenis mempunyai fungsi untuk menampilkan data peti kemas berdasarkan urutan jenis kepada *user*. Data *output* peti kemas berdasarkan urutan jenis terdiri dari jenis peti kemas, jumlah muat peti kemas, nominal *shipping*, dan berat muat peti kemas pada kapal laut. Desain *output* ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data transaksi uang tambang dan *chief officer* untuk dapat mengetahui data *order shipping*.

No	Jenis	Jumlah	Nominal	Berat
Total :		0 Item	Rp. 0,00	0 Ton

Gambar 3.28 Desain Output Peti Kemas Berdasarkan Urut Jenis

18. Desain Output Peti Kemas Berdasarkan Urut Ukuran

Gambar 3.28 merupakan desain *output* peti kemas berdasarkan Urut Ukuran. Desain *output* peti kemas berdasarkan urutan ukuran mempunyai fungsi untuk menampilkan data peti kemas berdasarkan urutan ukuran kepada *user*. Data *output* peti kemas berdasarkan urutan ukuran terdiri dari ukuran peti kemas, besar peti kemas, jumlah muat peti kemas pada kapal laut, nominal *shipping*, dan berat muat peti kemas pada kapal laut. Desain *output* ini diperuntukkan untuk hak akses sebagai bagian *marketing* untuk dapat mengisi data uang tambang dan *chief officer* dalam mengetahui data *order shipping*.

No	Ukuran	High Cube	Jumlah	Nominal	Berat
Total :			0 item	Rp. 0,00	0 Ton

Gambar 3.29 Desain Output Peti Kemas Berdasarkan Urut Ukuran

3.2.7 Rencana Evaluasi

Pada Tabel 3.15, merupakan rencana evaluasi, dalam melakukan testing sistem. Apakah sistem yang telah dibuat telah menjalankan prosedur sesuai dengan kebutuhan, jika tidak maka pengembang harus memperbaiki kesalahan tersebut. Untuk itu dibutuhkan *black box testing* untuk dapat menguji sistem yang telah dibuat berupa tabel-tabel.

Tabel 3.15 Rencana Evaluasi

No	Tujuan	Input	Output Harapan	Output Program	Hasil
1	Menyimpan data kapal laut	<i>Id</i> kapal laut='KRU01', nama kapal laut='Krakatau',kapasitas muat=750	Data kapal laut tersimpan		
2	Menyimpan data pelabuhan	<i>Id</i> pelabuhan='SUA01', nama pelabuhan='Surabaya'	Data pelabuhan tersimpan		
3	Menyimpan data pelabuhan tiba	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan tiba='JAA01', <i>id</i> line='L1302001', <i>id</i> kapal='KRU01', <i>eta</i> ='2013-02-10 06:00', <i>transit</i> ='0'	Data pelabuhan tiba tersimpan		
4	Menyimpan data pelabuhan berangkat	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan berangkat='SUA01', <i>E.T.D</i> ='2013-02-10 09:00'	Data pelabuhan berangkat tersimpan		
5	Menyimpan data <i>line</i> kapal	<i>Id line</i> ='L1302001'	Data <i>line</i> kapal tersimpan		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
6	Menyimpan data <i>voyage</i> kapal	<i>voyage</i> ='SU0JA130201'	Data <i>voyage</i> kapal tersimpan		
7	Menyimpan data jenis peti kemas	<i>Id</i> jenis peti kemas='DRY01', <i>jenis</i> ='dry'	Data jenis peti kemas tersimpan		
8	Menyimpan data peti kemas	No <i>container</i> ='DRY0120C00001' <i>id</i> jenis <i>container</i> ='DRY01', <i>ukuran</i> =0, <i>besar</i> =0	Data peti kemas tersimpan		
9	Menyimpan data <i>consignee</i>	<i>Id shipping</i> ='13GDU00001', <i>nama consignee</i> ='takashii mura', <i>alamat consignee</i> ='jl. Takoyaki 13, dumai', <i>telepon consignee</i> ='01374394702'	Data <i>consignee</i> tersimpan		
10	Menyimpan data <i>notify party</i>	<i>Id shipping</i> ='13GDU00001' <i>nama np</i> ='takashii mura', <i>alamat consignee</i> ='jl. Takoyaki 13, dumai', <i>telepon consignee</i> ='013743947'	Data <i>notify party</i> tersimpan		
11	Menyimpan data <i>shipper</i>	<i>Id shipper</i> ='D13001', <i>tgl registrasi</i> ='2013-01-01 09:00', <i>nama shipper</i> ='Dihan woro', <i>alamat shipper</i> ='jl. Manggis 10, Mojokerto', <i>telepon shipper</i> ='03170238920', <i>fax shipper</i> ='2987594', <i>email shipper</i> ='dihan@gmail.com'	Data <i>shipper</i> tersimpan		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
12	Menyimpan data <i>container shipper</i>	<i>Id shipping</i> ='13GDU00001', no <i>container</i> ='DRY0120C00001', komoditas='furniture', barang berbahaya=0, berat bersih='18.15', berat kotor='18.20', <i>quantity</i> ='120', <i>volume</i> ='30000', <i>nominal</i> ='0'	Data container shipper tersimpan		
13	Menyimpan data <i>shipping</i>	<i>Id shipping</i> ='13GDU00001', <i>voyage</i> ='JA0DU130201', <i>id shipper</i> ='G13001', tgl <i>shipping</i> ='2013-01-05 07:00' lokasi='jl. Jambu 6, Mojokerto', tanggal stuffing='2013-02-05', <i>freight</i> =0	Data shipping tersimpan		
14	Menyimpan data log pesan	<i>Id pesan</i> ='G1302001', log <i>line</i> ='L1302001', <i>id tgl</i> log='2013-01-01 09:00', isi pesan='muat peti kemas', pengirim='chief'	Data log pesan tersimpan		
15	Menyimpan data uang tambah	<i>Id shipping</i> ='13GDU00001', no <i>container</i> ='DRY0120C00001' <i>nominal</i> ='10000000'	Data uang tambah tersimpan		
16	Menghapus <i>list booking of cargo</i>	<i>Id line</i> ='L1302001'	Id line 'L1302001' terhapus		
17	Memilih muat peti kemas	<i>Id shipping</i> ='13GDU00001'	Id shipping '13GDU00001' terpilih		
18	Menghapus pilihan muat peti kemas	<i>Id shipping</i> ='13GDU00001'	Id shipping '13GDU00001' terhapus		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
19	Menampilkan data cari shipper	Nama <i>shipper</i> ='woro'	<i>Id shipper</i> ='D13001', tgl registrasi='2013-01-01 09:00', nama <i>shipper</i> ='Dihan woro', alamat <i>shipper</i> ='Jl. Manggis 10, Mojokerto', telepon <i>shipper</i> ='03170238920'		
20	Menampilkan daftar jadwal kapal laut	-	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan tiba='JAA01', <i>id line</i> ='L1302001', <i>id</i> kapal='KRU01', <i>eta</i> ='2013-02-10 06:00', <i>transit</i> ='0', <i>id</i> pelabuhan berangkat='SUA01', E.T.D='2013-02-10 09:00',nama kapal laut='K Krakatau',kapa sitas muat=750		
21	Menampilkan daftar jenis peti kemas	-	<i>Id</i> jenis peti kemas='DRY01', <i>jenis</i> ='dry'		
22	Menampilkan data cari peti kemas	<i>Id jenis</i> ="", <i>ukuran</i> ="", <i>besar</i> =""	No <i>container</i> ='DRY0120C00001' <i>id jenis container</i> ='DRY01', <i>avail</i> =0, <i>ukuran</i> ='20 feet', <i>jenis</i> ='dry', <i>besar</i> ='C'		
23	Menampilkan data cari jadwal kapal laut	Pelabuhan tiba="", E.T.D="" <i>transit</i> =""	<i>voyage</i> ='SU0JA130201', <i>id line</i> ='L1302001', <i>eta</i> ='10-02-2013 06:00',E.T.D='10-02-2013 12:00', <i>transit</i> =TREM PER, pelabuhan berangkat='Surabaya'		
24	Menampilkan daftar <i>container shipper</i>	<i>Id shipping</i> ='13GDU00001'	<i>Id shipping</i> ='13GDU00001', <i>no container</i> ='DRY0120C00001', <i>ukuran</i> ='20 feet', <i>besar</i> ='C', <i>jenis</i> ='dry'		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
25	Menampilkan detail <i>shipping</i> berdasarkan urutan jenis	<i>Id shipping</i> ='13GDU00001'	<i>Id shipping</i> ='13GDU00001', jenis='dry', jumlah=1, berat=18.20, nominal=10000000		
26	Menampilkan detail <i>shipping</i> berdasarkan urutan ukuran	<i>Id shipping</i> ='13GDU00001'	<i>Id shipping</i> ='13GDU00001' ukuran='20 feet' besar='C', jumlah=1, berat=18.20, nominal=10000000		
27	Menampilkan detail <i>shipping</i> urutan semua berdasarkan <i>id shipping</i>	<i>Id shipping</i> ='13GDU00001'	<i>Id shipping</i> ='13GDU00001' nominal=10000000, no <i>container</i> ='DRY0120C00001' komoditas='furniture', berat=18.20, <i>quantity</i> ='120', volume='30000', berbahaya='tidak', ukurn='20 feet', besar='C', jenis='dry', nominal='10000'		
28	Menampilkan detail <i>shipping</i> urutan semua berdasarkan <i>id line</i>	<i>Id line</i> ='L1302001'	<i>Id line</i> ='L1302001', No <i>container</i> ='DRY0120C00001', komoditas='furniture', berat=18.20, quntity=120, volume=30000, berbahaya='tidak', ukuran='20 feet', besar='C', jenis='dry'		
29	Menampilkan detail <i>shipping</i> urutan barang berbahaya	<i>Id shipping</i> ='13GDU00001'	berbahaya='tidak', jumlah=1, berat=18.20, nominal=10000		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
30	Menampilkan <i>booking of cargo</i>	-	Pelabuhan tiba='jakarta', <i>voyage</i> ='SU0JA130201', eta='10-02-2013 06:00', E.T.D='10-02-2013 12:00', hapus <i>line</i> =0, nama kapal laut='K Krakatau', pelabuhan berangkat='surabaya', <i>id line</i> ='L1302001'		
31	Menampilkan daftar <i>shipping</i>	<i>Id line</i> ='L1302001'	Pelabuhan tiba='jakarta', <i>voyage</i> ='SU0JA130201', eta='10-02-2013', hapus <i>line</i> =1, nama kapal laut='krakakatau', kapasitas muat=750, E.T.D='10-02-2013 12:00', <i>id line</i> ='1302001'		
32	Menampilkan <i>load list</i>	<i>Id line</i> ='L1302001'	Pelabuhan tiba='Jakarta', <i>voyage</i> ='SU0JA130201', <i>id line</i> ='L1302001', <i>freight</i> ='prepaid', <i>id shipping</i> ='13GDU00001', muat=0, nominal=10000000, jumlah=1, berat=18.20		
33	Menampilkan <i>load list summary</i>	<i>Id line</i> ='L1302001'	Pelabuhan tiba='Jakarta', <i>voyage</i> ='SU0JA130201', <i>id line</i> ='L1302001', <i>freight</i> ='prepaid', <i>id shipping</i> ='13GDU00001', muat=0, nominal=10000000, jumlah=1, berat=18.20		

No	Tujuan	Input	Output Harapan	Output Program	Hasil
34	Menampilkan pesan	<i>Id line='L1302001'</i>	<i>Id log='G1302001', id line='L1302001', tanggal log='01-01-2013 09:00', isi pesan='muat peti kemas', pengirim='chief'</i>		
35	Menghitung <i>knapsack</i>	berat={22,23,28,24,31,25,27,32}, nominal={1000000,3000000,5700000,5200000,5900000,5300000,5400000,6000000}	berat={28},nominal={5700000},jumlah={3}		



BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi Sistem

Setelah perancangan sistem selesai dibuat pada bab III, maka pada bab ini akan mengimplementasikan rancangan sistem tersebut berupa *screenshot* atau potongan-potongan gambar dari aplikasi yang telah dibuat. Berikut ini adalah *screenshot* aplikasi sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut pada metode *knapsack* yang akan dijelaskan layaknya seperti buku manual, bagaimana langkah-langkah menjalankan sistem informasi ini :

1. Halaman Depan Web



Gambar 4.1 Halaman Depan Web

Gambar 4.1 merupakan halaman depan *web* sistem informasi penentuan jumlah muat peti kemas pada kapal laut dengan menggunakan metode *knapsack*.

Halaman ini merupakan halaman pembuka yang berisi mengenai informasi PT. Anugerah Dwi Sukses selaku perusahaan *liner*. Informasi tersebut meliputi alamat, nomor telepon perusahaan, jasa yang diberikan perusahaan dan lain-lain. Halaman ini juga sekaligus sebagai halaman penghubung dengan *feature-feature* yang terkait dengan sistem seperti *maintenance data master*, transaksi *shipping*, penentuan jumlah peti kemas yang dimuat pada kapal laut dengan metode *knapsack*, dan mencetak laporan *load list summary* untuk *chief officer*. Halaman ini tidak memiliki hak akses, sehingga siapa pun dapat mengakses halaman ini.

2. Halaman Login



The image shows a login page for a system titled "SISTEM INFORMASI PENENTUAN JUMLAH PETI KEMAS YANG DIMUAT PADA KAPAL LAUT METODE KNAPSACK (studi kasus PT. Anugerah Dwi Sukses)". The page has a red header with a ship icon and a yellow navigation bar with a "HOME" link. Below the header is a "Login" section with input fields for "Username" and "Password", and a "Login" button. A large "STIKOM SURABAYA" watermark is visible in the background.

Gambar 4.2 Halaman Login

Gambar 4.2 merupakan halaman *login* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Halaman ini berfungsi untuk membagi hak akses masing-masing *entity* untuk membatasi *feature-feature* yang tersedia dalam sistem, yang bertujuan untuk meminimalkan risiko penyelewengan data. Adapun *entity* yang berhak untuk melakukan *login* adalah *entity customer service*, *entity bagian marketing*, *entity*

chief officer, dan *entity administrator*. Masing-masing *entity* mempunyai *feature* yang berlainan. Hal ini ditujukan untuk menjaga keamanan data pada sistem. Hak akses sistem ini masih dalam per jabatan dalam organisasi dan belum perorangan. Hal ini telah dijelaskan pada *sub* bab pembatasan masalah sistem.

3. Halaman Transaksi Order Shipping

Input Order Shipping

Input Data Shipper

Id Shipper :

Input Data Consignee

Nama Consignee :

Alamat Consignee :

Telepon Consignee :

Input Data Notify Party

Same as Consignee

Nama Notify Party :

Alamat Notify Party :

Telepon Notify Party :

Input Data Order Shipping

Voyage :

Tanggal Stuffing :

Lokasi Stuffing :

Freight : Prepaid
 Payable

Container

Gambar 4.3 Halaman transaksi Order Shipping

Gambar 4.3 merupakan halaman transaksi *order shipping* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Fungsi dari halaman ini adalah untuk menyimpan data transaksi *order shipping* yang bersumber dari *shipper*. Seorang shipper akan memberikan shipping instruction pada customer service untuk diinputkan ke dalam sistem. Untuk dapat mengisi data shipping instruction ini, seorang shipper harus sudah teregistrasi pada sistem ini. Seorang shipper diketahui sudah teregistrasi atau belum pada sistem dengan mengetahui id shipper yang bersangkutan. Apabila shipper yang bersangkutan belum teregistrasi maka seorang shipper diwajibkan untuk melakukan registrasi terlebih dahulu. Adapun hak akses yang diperuntukkan untuk halaman input ini adalah untuk *customer service*.

4. Halaman Cari Data Shipper

Cari Data Shipper

Nama Shipper :

[registrasi](#)

Id Shipper	Nama	Alamat	Telepon
D13001	Dihan woro	jl. manggis 10, Mojokerto	03170238920
N13001	Nurul Huda	jl. jambu 6, Mojokerto	036870238920
M13001	Muryono	jl. wonokromo 24, Surabaya	036870238920
M13002	Momo san	jl. ikan gabus 22, Surabaya	036870238920
K13001	Komang	jl. ikan gabus 21, Surabaya	036870238920
A13001	Ayu Anugraini	jl. ikan mungsing w/38, surabaya	03170238920
G13001	Gwen Stefani	jl. jambu 6, Singaraja Bali	036870238920

Gambar 4.4 Halaman Cari Data Shipper

Gambar 4.4 merupakan halaman cari data *shipper* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut. Halaman ini memiliki fungsi untuk membantu *entity customer service* dalam mencari *id shipper* yang harus dimasukkan pada halaman transaksi *order shipping*. Halaman ini hanya dapat di akses oleh *entity customer service*. Halaman ini akan muncul apabila *customer service* menekan *text box id shipper* pada halaman transaksi *order shipping*. Pada halaman ini *customer service* memasukkan nama *shipper* yang bersangkutan, maka sistem akan mencari nama *shipper* dalam *database*. Jika nama *shipper* telah terdaftar pada *database*, maka dalam *list shipper* yang ditampilkan oleh sistem, akan muncul data *shipper* secara lengkap. Jika pada *list shipper* tidak terdapat nama *shipper* yang dimaksud, maka *shipper* tersebut dipastikan belum mendaftar pada sistem, sehingga *customer service* harus melakukan registrasi pada *shipper* tersebut dengan menekan *link* registrasi yang telah tersedia pada halaman tersebut. Pada *list shipper* yang telah ditampilkan oleh sistem, *customer service* dapat menekan *id shipper* yang berupa *link* untuk menghubungkan ke halaman transaksi *order shipping*, sehingga *id shipper* pada halaman cari *shipper* akan berpindah ke halaman transaksi *order shipping*.

5. Halaman Cari Data Jadwal Kapal Laut

Gambar 4.5 merupakan halaman cari data jadwal kapal laut dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut. Halaman ini memiliki fungsi untuk membantu *entity shipper* dalam mencari jadwal keberangkatan kapal laut dalam melakukan *input* data nomor *voyage* pada halaman transaksi *order shipping*. Halaman ini hanya dapat di akses oleh *entity*

customer service. Dalam mencari jadwal keberangkatan kapal laut, terdapat beberapa data *input* yang harus dimasukkan oleh *customer service* diantaranya adalah tujuan keberangkatan kapal laut berupa nama kota yang terdapat di Indonesia, tanggal keberangkatan kapal laut, dan jenis keberangkatan kapal laut apakah itu *tremper* atau *liner*.

Jenis keberangkatan *tremper* memiliki arti bahwa keberangkatan kapal laut yang bersangkutan memiliki satu jalur untuk berlabuh dan jenis keberangkatan *liner* memiliki arti bahwa keberangkatan kapal laut memiliki lebih dari satu jalur berlabuh. Pada *list* keberangkatan kapal laut yang ditampilkan oleh sistem, maka *customer service* dapat menekan *link* nomor *voyage* yang diinginkan oleh *shipper*, maka nomor *voyage* tersebut akan berpindah dari halaman cari jadwal keberangkatan kapal laut menuju halaman transaksi *order shipping*.

Cari Data Jadwal Kapal Laut

Tujuan :

Tgl. Berangkat :

Transit : Liner
 Tremper

Voyage	Kapal Laut	E.T.A	E.T.D	Port Departure	Port Arrival	Transit
SU0JA130201	Krakatau	10-02-2013 06:00:00	10-02-2013 00:00:00	Surabaya	Jakarta	TREMPER
SE0JA130201	MerbaBUK01	13-02-2013 05:00:00	10-02-2013 00:00:00	Semarang	Jakarta	TREMPER

Gambar 4.5 Halaman Cari Data Jadwal Kapal Laut

6. Halaman Booking of Cargo

Gambar 4.6 merupakan halaman *booking of cargo* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan metode *knapsack*. Halaman ini memiliki fungsi untuk menampilkan seluruh jadwal keberangkatan kapal laut yang telah dibuat oleh *entity berth planner*. Halaman ini hanya dapat di akses oleh *entity* bagian *marketing*. Pada halaman ini bagian *marketing* dapat melakukan hapus pada *booking of cargo*, apabila terdapat jadwal keberangkatan kapal laut yang telah kadaluwarsa. Pada halaman ini juga terdapat *feature* notifikasi pesan yang belum terbaca yang dikirim oleh *chief officer*. Apabila terdapat jadwal keberangkatan kapal laut yang baru, biasanya akan ditempatkan pada *list* yang paling atas, maka bagian *marketing* dapat menekan *link id line* yang bersangkutan, maka sistem akan menghubungkan ke halaman daftar *order shipping*.



List Booking Of Cargo

Id Line	Surat	Nama Kapal	ETA	ETD	Port Arrival	Port Departure	
L1302001	surat	Krakatau	10-02-2013 10:00:00	10-02-2013 00:00:00	Dumai	Jakarta	Hapus
L1302002	surat	MerbaBUK01	13-02-2013 06:00:00	12-04-2013 00:00:00	Samarinda	Jakarta	Hapus
L1304001	surat	Kerinci	12-04-2013 12:00:00	12-04-2013 00:00:00	Samarinda	Jamurang	Hapus
L1305001	surat	Kilimanjaro	13-05-2013 11:00:00	13-05-2013 00:00:00	Morong	Samarinda	Hapus
L1306001	surat	Himalaya	16-06-2013 12:00:00	10-02-2013 00:00:00	Jakarta	Surabaya	Hapus
L1306002	surat	Bromo	17-06-2013 13:00:00	10-02-2013 00:00:00	Batam	Surabaya	Hapus

Gambar 4.6 Halaman Booking of Cargo

7. Halaman Order Shipping

Gambar 4.7 merupakan halaman daftar *order shipping* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Fungsi dari halaman ini adalah menampilkan seluruh data transaksi *order shipping* berdasarkan jadwal yang dipilih oleh bagian *marketing*. Halaman ini akan menampilkan data kapal laut, daftar jadwal *voyage* kapal laut, dan *list order shipping*. Kolom pendapatan pada *list shipping* merupakan harga jasa yang diberikan pihak perusahaan pada *shipper*. Apabila kolom pendapatan bernilai nol, maka bagian *marketing* harus memberikan harga pada *order shipping* yang bersangkutan berdasarkan pada data-data yang telah tersedia pada halaman tersebut. Bagian *marketing* akan menekan *link* pada kolom *id shipping* yang bersangkutan, maka sistem akan menghubungkan ke halaman *input harga shipping*. Adapun penjelasan lebih detail mengenai halaman *input harga shipping* akan dijelaskan di bawah ini. Halaman ini diperuntukkan untuk hak akses sebagai bagian *marketing* sebagai langkah awal menuju halaman detail *shipping* untuk memberikan uang tumbang pada masing-masing *order shipping*.

Data Kapal Laut

Id Line : L1302001
 Nama Kapal : Krakatau
 Jenis Voyage : LINER

Jadwal Voyage Kapal Laut

No. Voyage	E.T.A	E.T.D	Port Arrival	Port Departure
SU0JA130201	10-02-2013 06:00:00	10-02-2013 00:00:00	Jakarta	Surabaya
SU0JA130201	10-02-2013 06:00:00	13-02-2013 00:00:00	Jakarta	Surabaya
JA0DU130201	10-02-2013 10:00:00	10-02-2013 00:00:00	Dumai	Jakarta

List Shipping

No	Id Shipping	No. Voyage	Freight	Jml.	Tujuan	Pendapatan
1	13ADU00001	JA0DU130201	PREPAID	4	Dumai	121000000
2	13KJA00001	SU0JA130201	PREPAID	5	Jakarta	222000000
3	13KSA00001	BA0SA130201	PREPAID	2	Sabang	20000000
4	13MSA00001	BA0SA130201	PREPAID	5	Sabang	56000000
5	13NJA00001	SU0JA130201	PREPAID	3	Jakarta	124000000
6	13NJA00002	SU0JA130201	PREPAID	3	Jakarta	111000000
7	13NJA00003	SU0JA130201	PREPAID	4	Jakarta	46000000

Gambar 4.7 Halaman Order Shipping

8. Halaman Input Harga Shipping

Gambar 4.8 merupakan halaman *input* harga *shipping* dari sistem informasi penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Halaman ini memiliki fungsi untuk memberikan harga pada transaksi *order shipping*. Harga yang diberikan harus berdasarkan dengan data-data yang telah tersedia pada sistem. Data dari harga yang diberikan akan disimpan ke *database* oleh sistem. Setelah semua transaksi *order shipping* telah

diberikan harga yang pantas, maka bagian *marketing* dapat kembali ke halaman sebelumnya yaitu halaman daftar *order shipping*. Pada halaman *order shipping* apabila pada *list shipping* kolom pendapatan sudah tidak bernilai nol, maka bagian *marketing* dapat ke halaman penentuan jumlah peti kemas yang dimuat pada kapal laut dengan metode *knapsack*. Adapun penjelasan lebih detail mengenai halaman penentuan jumlah peti kemas yang dimuat pada kapal laut dengan metode *knapsack* dapat dilihat di bawah ini.



No	No. Cont.	Komoditas	Ukuran	Besar	Berbahaya	Quantity	Volume	Berat	Nominal
1	DRY0120C00006	besi tua	20 FEET	C	TIDAK	120	30000	19.80	10000000
2	OPP0120C00008	Furniture	20 FEET	C	TIDAK	120	30000	19.80	10000000
3	OPP0140HC0007	batok kelapa	40 FEET	H.C	TIDAK	256	66500	30.97	23000000
4	OPP0140C00008	bijih besi	40 FEET	C	TIDAK	301	68100	31.01	78000000

Gambar 4.8 Halaman Input Harga Shipping

9. Halaman Penentuan Jumlah Peti Kemas yang dimuat pada Kapal Laut dengan Menggunakan Metode Knapsack

Gambar 4.9 merupakan halaman penentuan jumlah peti kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Halaman ini memiliki fungsi untuk mengoptimalkan kapasitas muat berat peti kemas pada kapal laut dengan satuan ton dan juga mengoptimalkan pendapatan yang diperoleh dari transaksi *order shipping* dengan menggunakan metode *knapsack*. Halaman ini hanya dapat di akses oleh *entity* bagian *marketing*. Halaman ini menampilkan data

kapal laut, hasil perhitungan *knapsack*, *list shipping*, dan *list muat shipping*. Data yang ditampilkan pada halaman ini adalah data kapal laut, data hasil perhitungan *knapsack*, dan data pilih muat peti kemas pada kapal laut.

Data Kapal Laut

Id Line : L1302001
 Nama Kapal : Krakatau
 Kapasitas muat : 550 Ton

Hasil Perhitungan Knapsack

No	ID SI	Berat Optimum	Pendapatan Optimum
1	13NJA00002	83 Kg	Rp. 111.000.000
2	13NJA00001	92 Kg	Rp. 124.000.000
3	13MSA00001	106 Kg	Rp. 56.000.000
4	13KJA00001	147 Kg	Rp. 222.000.000
5	13ADU00001	102 Kg	Rp. 121.000.000
Total :		529 Kg	Rp. 634.000.000

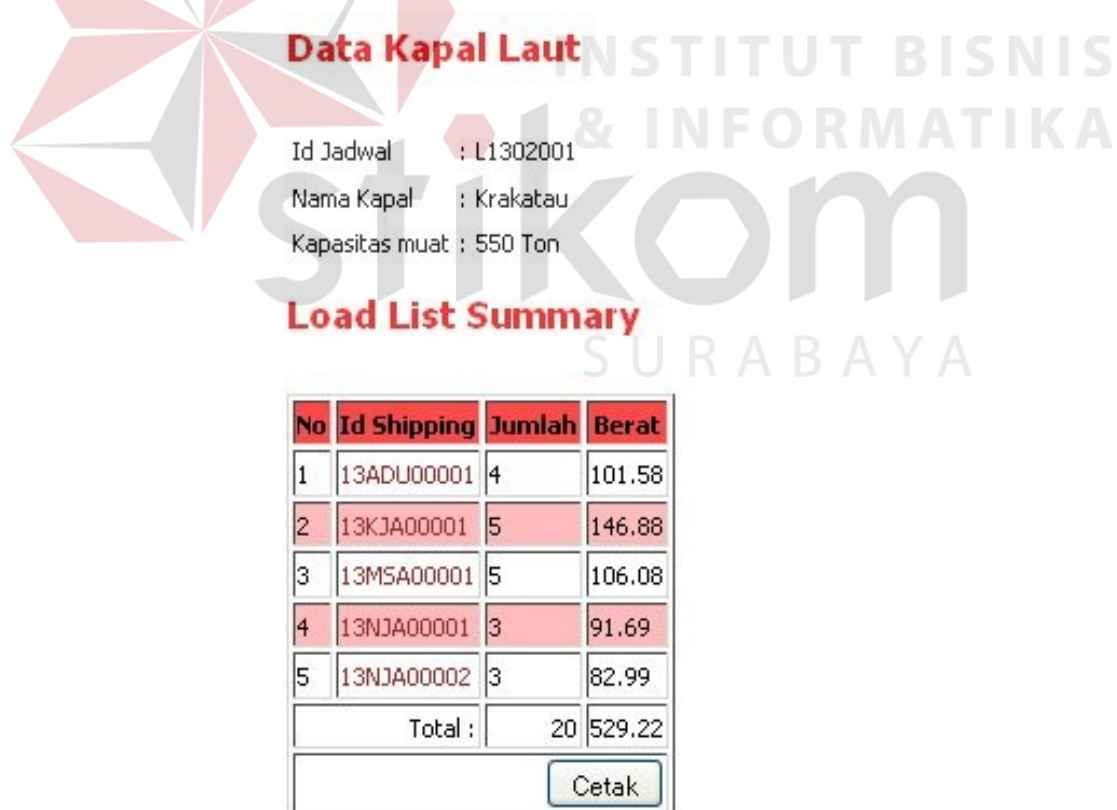
List Shipping

<input checked="" type="checkbox"/>	No	Id Shipping	jml. Cont.	Berat	Nominal
<input checked="" type="checkbox"/>	1	13ADU00001	4	101.58	121.000.000,00
<input checked="" type="checkbox"/>	2	13GDU00001	7	147.32	76.000.000,00
<input checked="" type="checkbox"/>	3	13KJA00001	5	146.88	222.000.000,00
<input checked="" type="checkbox"/>	4	13KSA00001	2	39.60	20.000.000,00
<input checked="" type="checkbox"/>	5	13MSA00001	5	106.08	56.000.000,00
<input checked="" type="checkbox"/>	6	13NJA00001	3	91.69	124.000.000,00
<input checked="" type="checkbox"/>	7	13NJA00002	3	82.99	111.000.000,00
<input checked="" type="checkbox"/>	8	13NJA00003	4	88.24	46.000.000,00
Total :			33	804.38	776.000.000,00

Gambar 4.9 Halaman Penentuan Jumlah Peti Kemas yang dimuat pada Kapal Laut dengan Menggunakan Metode Knapsack

10. Halaman Load List Summary

Gambar 4.10 merupakan halaman *load list summary* dari sistem informasi penentuan jumlah kemas yang dimuat pada kapal laut dengan menggunakan metode *knapsack*. Halaman ini memiliki fungsi untuk menampilkan data peti kemas yang sudah optimal untuk dimuat ke kapal laut. Halaman ini hanya dapat diakses oleh *entity chief officer*. Apabila terdapat kelebihan berat dari peti kemas dan masih harus dikurangi, maka *chief officer* dapat mengirimkan pesan mengenai *set out* peti kemas ke bagian *marketing* dengan *feature* kirim pesan yang telah tersedia. Jika pada *list load summary* sudah optimal, maka *chief officer* dapat mencetak *load list summary*.



Data Kapal Laut

Id Jadwal : L1302001
 Nama Kapal : Krakatau
 Kapasitas muat : 550 Ton

Load List Summary

No	Id Shipping	Jumlah	Berat
1	13ADU00001	4	101.58
2	13KJA00001	5	146.88
3	13MSA00001	5	106.08
4	13NJA00001	3	91.69
5	13NJA00002	3	82.99
Total :		20	529.22

Cetak

Gambar 4.10 Halaman Load List Summary

11. Halaman Cetak Load List Summary

Gambar 4.11 merupakan halaman cetak *load list summary* dari sistem informasi penentuan jumlah peti kemas pada kapal laut dengan menggunakan metode *knapsack*. Halaman ini mempunyai fungsi untuk mencetak *load list summary* sebagai bukti fisik bahwa penentuan jumlah peti kemas yang dimuat pada kapal laut telah disetujui dan dapat diimplementasikan secara *real*.

Laporan Pemuatan Peti Kemas pada Kapal Laut
PT. Anugerah Dwi Sukses

Tanggal : 05-02-2013 12:34:44

Data Kapal Laut

Id Jadwal : L1302001
Nama Kapal : Krakatau
Kapasitas muat : 550 Ton

Detail Container

No	No. Cont.	Ukuran	Besar	Berbahaya	Jenis	Quantity	Volume	Berat
1	DRY0120C00005	20 FEET	CUBE	TIDAK	dry	120	30.000	18.20
2	SOP0140C00001	40 FEET	CUBE	TIDAK	soft top	353	68.100	30.99
3	SOP0140450001	40 FEET	45	TIDAK	soft top	354	68.100	32.03
4	SOP0140450002	40 FEET	45	YA	soft top	213	68.100	32.03

Gambar 4.11 Halaman Cetak Load List Summary

4.2 Evaluasi Sistem

Setelah Aplikasi yang telah dirancang telah selesai dibuat, maka untuk langkah terakhir sebelum aplikasi ini diimplementasikan terhadap perusahaan ada baiknya dilakukan tahap evaluasi sistem dengan menggunakan metode *black box testing*. Dengan tahap ini diharapkan dapat meminimalisir kesalahan dalam sistem sebelum sampai ke tangan *end user*, dalam hal ini pihak perusahaan. Adapun hasil evaluasi sistem dapat dilihat pada tabel 4.1.

Tabel 4.1 Evaluasi Sistem

No	Tujuan	Input	Output Harapan	Output Program	Hasil
1	Menyimpan data kapal laut	<i>Id</i> kapal laut='KRU01', nama kapal laut='K Krakatau', kapasitas muat=750	Data kapal laut tersimpan	Data kapal laut tersimpan	sukses
2	Menyimpan data pelabuhan	<i>Id</i> pelabuhan='SUA01', nama pelabuhan='Surabaya'	Data pelabuhan tersimpan	Data pelabuhan tersimpan	sukses
3	Menyimpan data pelabuhan tiba	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan tiba='JAA01', <i>id</i> <i>line</i> ='L1302001', <i>id</i> kapal='KRU01', <i>eta</i> ='2013-02-10 06:00', <i>transit</i> ='0'	Data pelabuhan tiba tersimpan	Data pelabuhan tiba tersimpan	sukses
4	Menyimpan data pelabuhan berangkat	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan berangkat='SUA01', E.T.D='2013-02-10 09:00'	Data pelabuhan berangkat tersimpan	Data pelabuhan berangkat tersimpan	sukses
5	Menyimpan data kapal	<i>Id line</i> ='L1302001'	Data line kapal tersimpan	Data line kapal tersimpan	sukses
6	Menyimpan data kapal	<i>voyage</i> ='SU0JA130201'	Data voyage kapal tersimpan	Data voyage kapal tersimpan	sukses
7	Menyimpan data jenis peti kemas	<i>Id</i> jenis peti kemas='DRY01', <i>jenis</i> ='dry'	Data jenis peti kemas tersimpan	Data jenis peti kemas tersimpan	sukses
8	Menyimpan data peti kemas	No <i>container</i> ='DRY0120C00001' <i>id</i> jenis <i>container</i> ='DRY01', <i>ukuran</i> =0, <i>besar</i> =0	Data peti kemas tersimpan	Data peti kemas tersimpan	sukses
9	Menyimpan data <i>consignee</i>	<i>Id</i> <i>shipping</i> ='13GDU00001', nama <i>consignee</i> ='takashii mura', alamat <i>consignee</i> ='Jl. Takoyaki 13, dumai', telepon <i>consignee</i> ='01374394702'	Data <i>consignee</i> tersimpan	Data <i>consignee</i> tersimpan	sukses

No	Tujuan	Input	Output Harapan	Output Program	Hasil
10	Menyimpan data <i>notify party</i>	<i>Id shipping='13GDU00001'</i> nama np='takashii mura', alamat <i>consignee='jl. Takoyaki 13, dumai'</i> , telepon <i>consignee='01374394702'</i>	Data notify party tersimpan	Data notify party tersimpan	sukses
11	Menyimpan data <i>shipper</i>	<i>Id shipper='D13001'</i> , tgl registrasi='2013-01-01 09:00', nama <i>shipper='Dihan woro'</i> , alamat <i>shipper='jl. Manggis 10, Mojokerto'</i> , telepon <i>shipper='03170238920'</i> , fax <i>shipper='2987594'</i> , email <i>shipper='dihan@gmail.com'</i>	Data shipper tersimpan	Data shipper tersimpan	sukses
12	Menyimpan data <i>container shipper</i>	<i>Id shipping='13GDU00001'</i> , no <i>container='DRY0120C00001'</i> , komoditas='furniture', barang berbahaya=0, berat bersih='18.15', berat kotor='18.20', <i>quantity='120'</i> , <i>volume='30000'</i> , <i>nominal='0'</i>	Data container shipper tersimpan	Data container shipper tersimpan	sukses
13	Menyimpan data <i>shipping</i>	<i>Id shipping='13GDU00001'</i> , <i>voyage='JA0DU130201'</i> , <i>id shipper='G13001'</i> , tgl <i>shipping='2013-01-05 07:00'</i> lokasi='jl. Jambu 6, Mojokerto', tanggal <i>stuffing='2013-02-05'</i> , <i>freight=0</i>	Data shipping tersimpan	Data shipping tersimpan	sukses

No	Tujuan	Input	Output Harapan	Output Program	Hasil
14	Menyimpan data log pesan	<i>Id</i> log pesan='G1302001', <i>id line</i> ='L1302001', tgl log='2013-01-01 09:00', isi pesan='muat peti kemas', pengirim='chief'	Data log pesan tersimpan	Data log pesan tersimpan	sukses
15	Menyimpan data uang tambang	<i>Id shipping</i> ='13GDU00001' no container='DRY0120C00001' nominal='10000000'	Data uang tambang tersimpan	Data uang tambang tersimpan	sukses
16	Menghapus <i>list booking of cargo</i>	<i>Id line</i> ='L1302001'	Id line 'L1302001' terhapus	Id line 'L1302001' terhapus	sukses
17	Memilih muat peti kemas	<i>Id shipping</i> ='13GDU00001'	Id shipping '13GDU00001' terpilih	Id shipping '13GDU00001' terpilih	sukses
18	Menghapus pilihan muat peti kemas	<i>Id shipping</i> ='13GDU00001'	Id shipping '13GDU00001' terhapus	Id shipping '13GDU00001' terhapus	sukses
19	Menampilkan data cari <i>shipper</i>	Nama <i>shipper</i> ='woro'	<i>Id shipper</i> ='D13001', tgl registrasi='2013-01-01 09:00', nama <i>shipper</i> ='Dihan woro', alamat <i>shipper</i> ='Jl. Manggis 10, Mojokerto', telepon <i>shipper</i> ='03170238920'	<i>Id shipper</i> ='D13001', tgl registrasi='2013-01-01 09:00', nama <i>shipper</i> ='Dihan woro', alamat <i>shipper</i> ='Jl. Manggis 10, Mojokerto', telepon <i>shipper</i> ='03170238920'	sukses
20	Menampilkan daftar jadwal kapal laut	-	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan tiba='JAA01', <i>id line</i> ='L1302001', <i>id kapal</i> ='KRU01', <i>eta</i> ='2013-02-10 06:00', <i>transit</i> ='0', <i>id</i> pelabuhan berangkat='SUA01', E.T.D='2013-02-10 09:00', nama kapal laut='Krakatau', kapasitas muat=750	<i>voyage</i> ='SU0JA130201', <i>id</i> pelabuhan tiba='JAA01', <i>id line</i> ='L1302001', <i>id kapal</i> ='KRU01', <i>eta</i> ='2013-02-10 06:00', <i>transit</i> ='0', <i>id</i> pelabuhan berangkat='SUA01', E.T.D='2013-02-10 09:00', nama kapal laut='Krakatau', kapasitas muat=750	sukses

No	Tujuan	Input	Output Harapan	Output Program	Hasil
21	Menampilkan daftar jenis peti kemas	-	<i>Id</i> jenis peti kemas='DRY01', jenis='dry'	<i>Id</i> jenis peti kemas='DRY01', jenis='dry'	sukses
22	Menampilkan data cari peti kemas	<i>Id</i> jenis=",ukuran=",besar="	No container='DRY0120 C00001' <i>id</i> jenis container='DRY01', avail=0,ukuran='20 feet', jenis='dry', besar='C'	No container='DRY0120 C00001' <i>id</i> jenis container='DRY01', avail=0,ukuran='20feet', jenis='dry', besar='C'	sukses
23	Menampilkan data cari jadwal kapal laut	Pelabuhan tiba=", E.T.D=" transit="	voyage='SU0JA1302 01', <i>id</i> line='L1302001',eta='10-02-2013 06:00',E.T.D='10-02-2013 12:00',transit=TREMPER, pelabuhan berangkat='Surabaya'	voyage='SU0JA1302 01', <i>id</i> line='L1302001',eta='10-02-2013 06:00',E.T.D='10-02-2013 12:00',transit=TREMPER, pelabuhan berangkat='Surabaya'	sukses
24	Menampilkan daftar container shipper	<i>Id</i> shipping='13GDU00001'	<i>Id</i> shipping='13GDU00001', no container='DRY0120 C00001', ukuran='20 feet', besar='C', jenis='dry'	<i>Id</i> shipping='13GDU00001', no container='DRY0120 C00001', ukuran='20 feet', besar='C', jenis='dry'	sukses
25	Menampilkan detail shipping berdasarkan urutan jenis	<i>Id</i> shipping='13GDU00001'	<i>Id</i> shipping='13GDU00001', jenis='dry', jumlah=1, berat=18.20, nominal=10000000	<i>Id</i> shipping='13GDU00001', jenis='dry', jumlah=1, berat=18.20, nominal=10000000	sukses
26	Menampilkan detail shipping berdasarkan urutan ukuran	<i>Id</i> shipping='13GDU00001'	<i>Id</i> shipping='13GDU00001' ukuran='20 feet' besar='C', jumlah=1, berat=18.20, nominal=10000000	<i>Id</i> shipping='13GDU00001' ukuran='20 feet' besar='C', jumlah=1, berat=18.20, nominal=10000000	sukses

No	Tujuan	Input	Output Harapan	Output Program	Hasil
27	Menampilkan detail <i>shipping</i> urut semua berdasarkan <i>id shipping</i>	<i>Id shipping</i> ='13GDU0001'	<i>Id shipping</i> ='13GDU0001' nominal=10000000, no <i>container</i> ='DRY0120C00001' komoditas='furniture', berat=18.20, <i>quantity</i> ='120', volume='30000', berbahaya='tidak', ukuran='20 feet', besar='C', jenis='dry', nominal='10000'	<i>Id shipping</i> ='13GDU0001' nominal=10000000, no <i>container</i> ='DRY0120C00001' komoditas='furniture', berat=18.20, <i>quantity</i> ='120', volume='30000', berbahaya='tidak', ukuran='20 feet', besar='C', jenis='dry', nominal='10000'	sukses
28	Menampilkan detail <i>shipping</i> urut semua berdasarkan <i>id line</i>	<i>Id line</i> ='L1302001'	<i>Id line</i> ='L1302001', No <i>container</i> ='DRY0120C00001', komoditas='furniture', berat=18.20, quantity=120, volume=30000, berbahaya='tidak', ukuran='20 feet', besar='C', jenis='dry'	<i>Id line</i> ='L1302001', No <i>container</i> ='DRY0120C00001', komoditas='furniture', berat=18.20, quantity=120, volume=30000, berbahaya='tidak', ukuran='20 feet', besar='C', jenis='dry'	sukses
29	Menampilkan detail <i>shipping</i> urut barang berbahaya	<i>Id shipping</i> ='13GDU0001'	berbahaya='tidak', jumlah=1, berat=18.20, nominal=10000	berbahaya='tidak', jumlah=1, berat=18.20, nominal=10000	sukses
30	Menampilkan <i>booking of cargo</i>	-	Pelabuhan tiba='jakarta', <i>voyage</i> ='SU0JA130201', eta='10-02-2013 06:00', E.T.D='10-02-2013 12:00', hapus <i>line</i> =0, nama kapal laut='K Krakatau', pelabuhan berangkat='surabaya', <i>id line</i> ='L1302001'	Pelabuhan tiba='jakarta', <i>voyage</i> ='SU0JA130201', eta='10-02-2013 06:00', E.T.D='10-02-2013 12:00', hapus <i>line</i> =0, nama kapal laut='K Krakatau', pelabuhan berangkat='surabaya', <i>id line</i> ='L1302001'	sukses

No	Tujuan	Input	Output Harapan	Output Program	Hasil
31	Menampilkan daftar <i>shipping</i>	<i>Id line='L1302001'</i>	Pelabuhan tiba='jakarta', <i>voyage='SU0JA130201'</i> , eta='10-02-2013', hapus <i>line=1</i> , nama kapal laut='krakakatau', kapasitas muat=750, E.T.D='10-02-2013 12:00', <i>id line='1302001'</i>	Pelabuhan tiba='jakarta', <i>voyage='SU0JA130201'</i> , eta='10-02-2013', hapus <i>line=1</i> , nama kapal laut='krakakatau', kapasitas muat=750, E.T.D='10-02-2013 12:00', <i>id line='1302001'</i>	sukses
32	Menampilkan <i>load list</i>	<i>Id line='L1302001'</i>	Pelabuhan tiba='Jakarta', <i>voyage='SU0JA130201'</i> , <i>id line='L1302001'</i> , <i>freight='prepaid'</i> , <i>id shipping='13GDU00001'</i> , muat=0, nominal=10000000, jumlah=1, berat=18.20	Pelabuhan tiba='Jakarta', <i>voyage='SU0JA130201'</i> , <i>id line='L1302001'</i> , <i>freight='prepaid'</i> , <i>id shipping='13GDU00001'</i> , muat=0, nominal=10000000, jumlah=1, berat=18.20	sukses
33	Menampilkan <i>load list summary</i>	<i>Id line='L1302001'</i>	Pelabuhan tiba='Jakarta', <i>voyage='SU0JA130201'</i> , <i>id line='L1302001'</i> , <i>freight='prepaid'</i> , <i>id shipping='13GDU00001'</i> , muat=0, nominal=10000000, jumlah=1, berat=18.20	Pelabuhan tiba='Jakarta', <i>voyage='SU0JA130201'</i> , <i>id line='L1302001'</i> , <i>freight='prepaid'</i> , <i>id shipping='13GDU00001'</i> , muat=0, nominal=10000000, jumlah=1, berat=18.20	sukses
34	Menampilkan pesan	<i>Id line='L1302001'</i>	<i>Id log='G1302001'</i> , <i>id line='L1302001'</i> , tanggal log='01-01-2013 09:00', isi pesan='muat peti kemas', pengirim='chief'	<i>Id log='G1302001'</i> , <i>id line='L1302001'</i> , tanggal log='01-01-2013 09:00', isi pesan='muat peti kemas', pengirim='chief'	sukses
35	Menghitung <i>knapsack</i>	berat={22,23,28,24,31,25,27,32}, nominal={10000000,30000000,57000000,52000000,59000000,53000000,54000000,60000000}	berat={28},nominal={57000000},jumlah={3}	berat={28},nominal={57000000},jumlah={3}	sukses

4.3 Analisis Evaluasi

Berdasarkan pada hasil evaluasi yang telah dilakukan di atas, maka didapatkan analisis- analisis sebagai berikut :

1. *Black box testing* pada fungsi dasar sistem

Beberapa tes yang telah dilakukan untuk menguji fungsi-fungsi dasar yang terdapat pada sistem penentuan jumlah peti kemas yang dimuat pada kapal laut telah berjalan sesuai dengan tujuan dan kebutuhan perusahaan. Hal tersebut telah terbukti dengan dibuatnya Evaluasi sistem pada tabel 4,1, dari nomor *testing* 1 sampai dengan nomor *testing* 34. Hasil evaluasi sistem dengan teknik *black box testing* pada fungsi dasar sistem tersebut ditunjukkan pada hasil akhir bernilai “sukses”.

2. Perhitungan penentuan jumlah peti kemas yang dimuat pada kapal laut memakai metode *knapsack*

Dilakukan sebuah uji coba sederhana dalam menentukan jumlah peti kemas yang dimuat pada kapal laut tanpa menggunakan metode *knapsack* secara manual untuk menguji apakah aplikasi yang dibangun telah menunjukkan informasi yang akurat kepada *user*. Adapun contoh kasusnya adalah sebagai berikut :

Terdapat data transaksi *shipping* pada PT. Anugerah Dwi Sukses dengan beberapa peti kemas dengan masing-masing berat adalah 22, 23, 28, 24, 31, 25, 27 dan 32 (dalam satuan ton). Pendapatan jasa muat peti kemas yang diterima oleh perusahaan tersebut masing-masing adalah 10.000.000, 30.000.000, 57.000.000, 52.000.000, 59.000.000, 53.000.000, 54.000.000, dan 60.000.000 (dalam satuan rupiah). Peti kemas tersebut diangkut dengan sebuah kapal laut yang kapasitas 100

ton. Adapun data transaksi *shipping* dapat dilihat pada tabel 4.2.

Tabel 4.2 Data *Shipping*

Item	Berat (Ton)	Pendapatan (Rupiah)
1	22	10.000.000
2	23	30.000.000
3	28	57.000.000
4	24	52.000.000
5	31	59.000.000
6	25	53.000.000
7	27	54.000.000
8	32	60.000.000
Total	212	375.000.000

Dari data *shipping* pada tabel di atas terlihat bahwa total berat dari keseluruhan peti kemas adalah 212 ton dan pendapatan total adalah Rp 375.000.000. Hal tersebut menunjukkan bahwa peti kemas yang akan dimuat pada kapal laut harus dilakukan seleksi terlebih dahulu agar dapat dimuat pada kapal laut yang kapasitas hanya 100 ton. Untuk dapat membuktikan bahwa hasil *output* program pada perhitungan *knapsack* pada evaluasi sistem tabel 4.3 dengan nomor *testing* 35 bernilai benar, maka akan dibuat sebuah perhitungan manual sebagai perbandingan bahwa *output* perhitungan *knapsack* pada sistem dengan perhitungan manual adalah sama. Adapun perhitungan manual *knapsack* sebagai berikut :

Perhitungan *knapsack stage* ke-0 adalah sebagai berikut:

$$f_0(x_0) = \max_{\substack{m_0=0,1,2,3 \\ x_0=0,22,44,66,88}} \{10000000 \cdot m_0\}, \text{ untuk lebih jelasnya dapat dilihat}$$

pada tabel 4.3 sebagai berikut :

Tabel 4.3 Perhitungan *knapsack stage* ke-0

x_0	$10000000 \cdot m_0$					<i>Optimum Solution</i>	
	$m_0 = 0$	$m_0 = 1$	$m_0 = 2$	$m_0 = 3$	$m_0 = 4$	$f_0(x_0)$	m_0
0	0	-	-	-	-	0	0
22	0	10000000	-	-	-	10000000	1
44	0	10000000	20000000	-	-	20000000	2
66	0	10000000	20000000	30000000	-	30000000	3
88	0	10000000	20000000	30000000	40000000	40000000	4

Perhitungan *knapsack stage* ke-1 adalah sebagai berikut :

$$f_1(x_1) = \max_{\substack{m_1=0,1,2,3 \\ x_1=0,44,66,88}} \{30000000 \cdot m_1 + f_1(x_1 - 23 \cdot m_1)\} . \text{ Untuk lebih jelasnya}$$

dapat dilihat pada tabel 4.4 sebagai berikut :

Tabel 4.4 Perhitungan *knapsack stage* ke-1

x_1	$30000000 \cdot m_1 + f_1(x_1 - 23 \cdot m_1)$				<i>Optimum Solution</i>	
	$m_1 = 0$	$m_1 = 1$	$m_1 = 2$	$m_1 = 3$	$f_1(x_1)$	m_1
0	0	-	-	-	0	0
22	10000000	-	-	-	10000000	0
44	20000000	30000000	-	-	30000000	1
66	30000000	40000000	60000000	-	60000000	2
88	40000000	50000000	70000000	90000000	90000000	3

Perhitungan *knapsack stage* ke-2 adalah sebagai berikut :

$$f_2(x_2) = \max_{\substack{m_2=0,1,2,3 \\ x_2=0,44,66,88}} \{52000000 \cdot m_2 + f_2(x_2 - 24 \cdot m_2)\} . \quad \text{Adapun}$$

perhitungannya dapat dilihat pada tabel 4.5 sebagai berikut :

Tabel 4.5 Perhitungan knapsack *stage* ke-2

x_2	$52000000 \cdot m_2 + f_2(x_2 - 24 \cdot m_2)$				<i>Optimum Solution</i>	
	$m_2 = 0$	$m_2 = 1$	$m_2 = 2$	$m_2 = 3$	$f_2(x_2)$	m_2
0	0	-	-	-	0	0
22	10000000	-	-	-	10000000	0
44	30000000	52000000	-	-	52000000	1
66	60000000	62000000	104000000	-	104000000	2
88	90000000	82000000	114000000	156000000	156000000	3

Perhitungan *knapsack stage* ke-3 adalah sebagai berikut :

$$f_3(x_3) = \max_{m_3=0,1,2,3} \left\{ 53000000 \cdot m_3 + f_3(x_3 - 25 \cdot m_3) \right\}, \text{ Untuk lebih jelasnya } x_3 = 0, 44, 66, 88$$

dapat dilihat pada tabel 4.6 sebagai berikut :

Tabel 4.6 Perhitungan *knapsack stage* ke-3

x_3	$53000000 \cdot m_3 + f_3(x_3 - 25 \cdot m_3)$				<i>Optimum Solution</i>	
	$m_3 = 0$	$m_3 = 1$	$m_3 = 2$	$m_3 = 3$	$f_3(x_3)$	m_3
0	0	-	-	-	0	0
22	10000000	-	-	-	10000000	0
44	52000000	53000000	-	-	53000000	1
66	104000000	63000000	106000000	-	106000000	2
88	156000000	105000000	116000000	159000000	159000000	3

Perhitungan *knapsack stage* ke-4 adalah sebagai berikut :

$$f_4(x_4) = \max_{m_4=0,1,2,3} \left\{ 54000000 \cdot m_4 + f_4(x_4 - 27 \cdot m_4) \right\}, \text{ untuk lebih jelasnya } x_4 = 0, 44, 66, 88$$

dapat dilihat pada tabel 4.7 sebagai berikut :

Tabel 4.7 Perhitungan *knapsack stage* ke-4

x_4	$54000000 \cdot m_4 + f_4(x_4 - 27 \cdot m_4)$				<i>Optimum Solution</i>	
	$m_4 = 0$	$m_4 = 1$	$m_4 = 2$	$m_4 = 3$	$f_4(x_4)$	m_4
0	0	-	-	-	0	0
22	10000000	-	-	-	10000000	0
44	53000000	54000000	-	-	54000000	1
66	106000000	64000000	108000000	-	108000000	2
88	159000000	107000000	118000000	162000000	162000000	3

Perhitungan *knapsack stage* ke-5 adalah sebagai berikut :

$$f_5(x_5) = \max_{m_5=0,1,2,3} \left\{ 57000000 \cdot m_5 + f_5(x_5 - 28 \cdot m_5) \right\}, \text{ untuk lebih jelasnya } x_5 = 0, 44, 66, 88$$

dapat dilihat pada tabel 4.8 sebagai berikut :

Tabel 4.8 Perhitungan *knapsack stage* ke-5

x_5	$57000000 \cdot m_5 + f_5(x_5 - 28 \cdot m_5)$				<i>Optimum Solution</i>	
	$m_5 = 0$	$m_5 = 1$	$m_5 = 2$	$m_5 = 3$	$f_5(x_5)$	m_5
0	0	-	-	-	0	0
22	10000000	-	-	-	10000000	0
44	54000000	57000000	-	-	57000000	1
66	108000000	67000000	114000000	-	114000000	2
88	162000000	111000000	124000000	171000000	171000000	3

Perhitungan *knapsack stage* ke-6 adalah sebagai berikut :

$$f_6(x_6) = \max_{m_6=0,1,2} \left\{ 59000000 \cdot m_6 + f_6(x_6 - 31 \cdot m_6) \right\}, \text{ untuk lebih jelasnya } x_6 = 0, 44, 66$$

dapat dilihat pada tabel 4.9 sebagai berikut :

Tabel 4.9 Perhitungan *knapsack stage* ke-6

x_6	$59000000 \cdot m_6 + f_6(x_6 - 31 \cdot m_6)$			<i>Optimum Solution</i>	
	$m_6 = 0$	$m_6 = 1$	$m_6 = 2$	$f_6(x_6)$	m_6
0	0	-	-	0	0
22	10000000	-	-	10000000	0
44	57000000	59000000	-	59000000	1
66	114000000	69000000	118000000	118000000	2
88	171000000	116000000	128000000	171000000	0

Perhitungan *knapsack stage* ke-7 adalah sebagai berikut :

$$f_7(x_7) = \max_{m_7=0,1,2} \left\{ 60000000 \cdot m_7 + f_7(x_7 - 32 \cdot m_7) \right\}, \text{ untuk lebih jelasnya } x_7 = 0, 44, 66$$

dapat dilihat pada tabel 4.10 sebagai berikut :

Tabel 4.10 Perhitungan *knapsack stage* ke-7

x_7	$60000000 \cdot m_7 + f_7(x_7 - 32 \cdot m_7)$			<i>Optimum Solution</i>	
	$m_7 = 0$	$m_7 = 1$	$m_7 = 2$	$f_7(x_7)$	m_7
0	0	-	-	0	0
22	10000000	-	-	10000000	0
44	59000000	60000000	-	60000000	1
66	118000000	70000000	120000000	120000000	2
88	171000000	119000000	130000000	171000000	0

Setelah dilakukan perhitungan *knapsack* seperti di atas, maka hasil pendapatan optimum yang didapat dari transaksi *order* peti kemas tersebut dapat mencapai Rp. 171.000.000 dan berat optimum peti kemas yang dapat dimuat pada kapal laut adalah 84 ton, dengan mengangkut peti kemas dengan berat 28 ton dan pendapatan sebesar Rp. 57.000.000 sebanyak 3 *item*.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil analisis masalah pada PT. Anugerah Dwi Sukses sampai pada perancangan sistem, kemudian implementasi dan evaluasi sistem, dapat disimpulkan sebagai berikut :

1. Aplikasi yang telah dibangun beserta semua fungsi-fungsi dasarnya telah sesuai dengan kebutuhan dan tujuan perusahaan. Fungsi dasar tersebut meliputi fungsi menyimpan data *shipper*, menyimpan data *consignee*, menyimpan data *order shipping* dan lain-lain. Fungsi-fungsi tersebut turut andil bagian di dalam perhitungan knapsack untuk menentukan berat optimal peti kemas yang dimuat pada kapal laut.
2. Metode *knapsack* yang terimplentasikan ke dalam sistem yang dibangun, telah mampu memberikan informasi berupa berat optimal peti kemas yang dimuat pada kapal laut kepada *chief officer*.

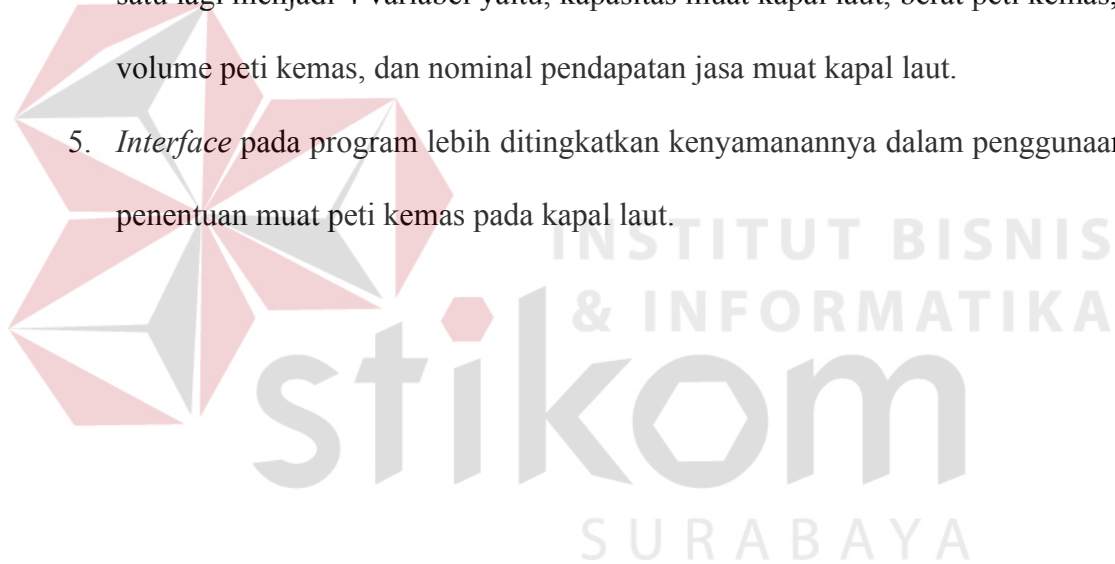
5.2 Saran

Adapun beberapa saran untuk para developer atau mahasiswa yang ingin melanjutkan atau ingin mengembangkan sistem ini supaya menjadi lebih berguna bagi perusahaan pengguna adalah sebagai berikut :

1. Terdapat metode dalam melakukan penjadwalan pada kapal laut.
2. Terdapat perhitungan secara otomatis dalam memberikan nominal pendapatan

muat peti kemas pada kapal laut berdasarkan dari data transaksi *shipping*, serta mata uang yang digunakan pada sistem dapat bervariasi.

3. Skala pengembangan sistem dapat diperluas, seperti pengadaan bidang-bidang tertentu dalam perusahaan seperti bidang bongkar peti kemas pada kapal laut, bidang keuangan, bidang administrasi, bidang kepegawaian, bidang penjadwalan kapal laut dan lain-lain.
4. Variabel yang digunakan untuk menentukan jumlah muat peti kemas ditambah satu lagi menjadi 4 variabel yaitu, kapasitas muat kapal laut, berat peti kemas, volume peti kemas, dan nominal pendapatan jasa muat kapal laut.
5. *Interface* pada program lebih ditingkatkan kenyamanannya dalam penggunaan penentuan muat peti kemas pada kapal laut.



DAFTAR PUSTAKA

Jogiyanto, M., Hartono. 2005, *Analisis dan Desain Sistem Informasi : pendekatan Terstruktur dan Praktek Aplikasi Bisnis*, Andi Offset, Yogyakarta.

Kendall & Kendall, 2003, *Analisis dan Perancangan Sistem, Jilid 1*, Person Education Asia Pte. Ltd. Dan PT Prenhallindo, Jakarta.

Romeo, 2003, *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM, Surabaya.

Sudjarmiko, 2001, *Pokok Pelayaran Niaga*, PT. Toko Gunung Agung, Jakarta.

Taha, Hamdy, A., 2007, *Research Operation*, Prentice Hall, New York.

Undang-Undang Republik Indonesia Nomor 17 Tahun 2008 Tentang Pelayaran



LAMPIRAN

Lampiran 1. Biodata Penulis

BIODATA PENULIS

Nama : Nyoman Tri Dharma yasa

Alamat : Jl. Ikan Mungsing V/38
Surabaya – Jawa Timur

Tempat/ Tgl. Lahir : Surabaya, 04 September 1988

Email : slowmer@gmail.com

Contact Person : 03170238920

Riwayat Pendidikan :

1994 – 2000, SDN Perak Barat I Surabaya

2000 – 2003, SMPN 7 Surabaya

2003 – 2006, SMAN 7 Surabaya

2006 – 2013, S1-SI STIKOM Surabaya



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA

Lampiran 2. Listing Program

Halaman Perhitungan Knapsack

```

<?php
    session_start();
?>
<html>
<head><title>Perhitungan Knapsack</title>
</head>
<body>
<form method=POST action=DetailPerhitungan.php>
<?php
require_once '../Control.php';
require_once '../TabelKnapsack1.php';

$ctrl = new Control();

$idLine = $_SESSION['txtIdLine'];

if(isset($idLine))
{
    //menampilkan daftar shipping (3.3)
    $ctrl->getKon()->buka();
    $temp = $ctrl->getKon()->show($ctrl->getQuery()->daftarSi($idLine));
    $ctrl->getKon()->tutup();

    //menampilkan grup berat peti kemas asc (3.4)
    $ctrl->getKon()->buka();
    $temp1 = $ctrl->getKon()->show($ctrl->getQuery()->grupContainer1($idLine));
    $ctrl->getKon()->tutup();
}
else
{
    echo "tidak dapat menampilkan daftar shipping, karena id line kosong!";
}

$kapasitas = $temp[0]['KAPASITAS_MUAT'];

foreach($temp1 as $hasil)
{
    $berat[] = $hasil['BERAT'];
    $pendapatan[] = $hasil['NOMINAL'];
}

$knap = new TabelKnapsack1();
$knap->output1($kapasitas,$berat,$pendapatan);
?>
</form>
</body>
</html>

```

Query.php

```
<?php
class Query
{
    //menghapus jadwal kapal laut (4.4)
    function hapusJadwal($idLine)
    {
        $temp = "call hapusjadwal('$idLine')";
        return $temp;
    }

    //menampilkan booking of cargo (3.2)
    function booking()
    {
        $temp = "call booking()";
        return $temp;
    }

    //memberikan harga shipping (2.8)
    function inputHargaSi($id, $no, $nominal)
    {
        $temp = "call inpuhargasi('$id','$no','$nominal')";
        return $temp;
    }

    //menampilkan load list (3.6)
    function loadList($idShipping)
    {
        $temp = "call loadlist('$idShipping')";
        return $temp;
    }

    //menampilkan detail shipping (3.5)
    function detailSi($idLine,$muat)
    {
        $temp = "call detailshipping('$idLine','$muat')";
        return $temp;
    }

    //menampilkan daftar shipping (3.3)
    function daftarSi($idLine)
    {
        $temp = "call daftarshipping('$idLine')";
        return $temp;
    }

    //menampilkan grup berat peti kemas asc (3.4)
    function grupContainer($idLine)
    {
        $temp = "call grupcontainer('$idLine')";
        return $temp;
    }

    //menampilkan grup berat peti kemas asc (3.4)
    function grupContainer1($idLine)
```

```

{
    $temp = "call grupcontainer1('$idLine');";
    return $temp;
}

//menampilkan data peti kemas belum sudah terpilih (3.1)
function containerPilihan($idLine,$nominal,$muat,$berat)
{
    $temp = "SELECT D.ID_LINE, A.ID_SHIPPING, A.NO_CONTAINER,
floor(IF(B.UKURAN=0 AND B.BESAR=0,1.6,IF(B.UKURAN=0 AND
B.BESAR=1,2.32,IF(B.UKURAN=0 AND B.BESAR=2,2.4,IF(B.UKURAN=1 AND
B.BESAR=0,3.41,IF(B.UKURAN=1 AND B.BESAR=1,3.9,IF(B.UKURAN=1 AND
B.BESAR=2,4.45,0))))))+A.BERAT_KOTOR) 'BERAT', A.NOMINAL, A.MUAT FROM
SHIPPING_CONTAINER A JOIN CONTAINER B ON A.NO_CONTAINER =
B.NO_CONTAINER JOIN SHIPPING C ON C.ID_SHIPPING = A.ID_SHIPPING JOIN
PELABUHAN_TIBA D ON D.VOYAGE = C.VOYAGE WHERE D.ID_LINE = '$idLine' AND
A.NOMINAL like '$nominal%' and A.MUAT = '$muat' having BERAT like '$berat%';";
    return $temp;
}

//menghapus dan memilih muat peti kemas (2.14)
function hapusMuat($idSi, $noContainer, $muat)
{
    $temp = "call hapusmuat('$idSi','$noContainer','$muat');";
    return $temp;
}
?>
}
}

Control.php
<?php
require_once 'Koneksi.php';
require_once 'Query.php';

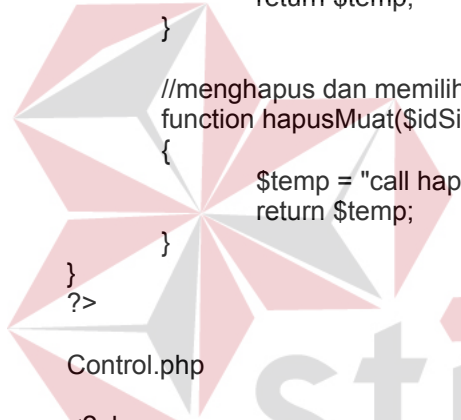
class Control
{
    private $koneksi;
    private $query;

    function Control()
    {
        $this->koneksi = new Koneksi();
        $this->query = new Query();
    }

    function getKon()
    {
        return $this->koneksi;
    }

    function getQuery()
    {
        return $this->query;
    }
}
}

```



?>

Koneksi.php

```

<?php
class Koneksi
{
    private $hostname;
    private $database;
    private $username;
    private $password;
    private $db;

    public function Koneksi()
    {
        $this->hostname = 'localhost';
        $this->username = 'root';
        $this->database = 'ANUGERAH_REVISI';
        $this->password = 'slowmer';
    }

    public function setHostname($hostname)
    {
        $this->hostname = $hostname;
    }

    public function setUsername($username)
    {
        $this->username = $username;
    }

    public function setPassword($password)
    {
        $this->password = $password;
    }

    public function setDatabase($database)
    {
        $this->database = $database;
    }

    public function buka()
    {
        $this->db = @new MySQLi($this->hostname,$this->username,$this->password,$this->database);
        $this->db->connect_errno and die("<h3 style='color:red'>Anda tidak mempunyai hak akses!</h3><br><a href='../Beranda.php'>Kembali</a>");
    }

    public function eksekusi($query)
    {
        $this->db->query($query) or die("<h3 style='color:red'>gagal disimpan</h3>");
        //echo $query . "<br>";
    }
}

```

```

public function show($query)
{
    $result = $this->db->query($query) or die("<h3 style='color:red'>gagal
menampilkan</h3>");
    while($row = $result->fetch_assoc()) {
        $temp[] = $row;
    }
    if(empty($temp))
    {
        $temp = "";
    }
    return $temp;
}

```

```

public function tutup()
{
    mysqli_close($this->db);
}
}
?>

```

KnapsackTaha2.php

```

<?php
class KnapsackTaha2
{

```

```

    function getKecil($w)
    {
        $kecil = $w[0];
        return $kecil;
    }

```

```

    function getBatasan($W,$kecil)
    {
        $batasan = $W / $kecil;
        return $batasan;
    }

```

```

    function getAlokW($batasan, $w,$alokM)
    {
        for($i=0;$i<=$batasan;$i++)
        {
            $alokW[] = $w * $alokM[$i];
        }
        return $alokW;
    }

```

```

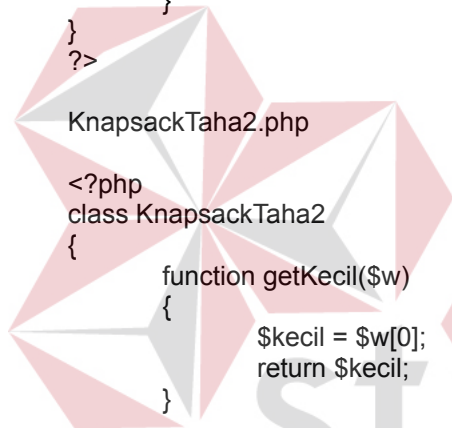
    function getAlokWKecil($alokW)
    {
        $alokWKecil = $alokW[1];
        return $alokWKecil;
    }

```

```

    function getAlokR($batasan, $r, $alokM)
    {
        for($i=0;$i<=$batasan;$i++)

```



```

    {
        $alokR[] = $r * $alokM[$i];
    }
    return $alokR;
}

function getAlokM($batasan)
{
    for($i=0;$i<=$batasan;$i++)
    {
        $alokM[] = $i;
    }
    return $alokM;
}

function getAlokMKecil($alokM)
{
    $alokMKecil = $alokM[1];
    return $alokMKecil;
}

function getX($kecil,$batasan)
{
    for($i=0;$i<=$batasan;$i++)
    {
        $x[] = $kecil * $i;
    }
    return $x;
}

function getAlokW1($alokW,$x,$batasan)
{
    $i=0;
    $j=0;

    while($j<=$batasan)
    {
        if($i<=$batasan)
        {
            if($x[$i]>=$alokW[$j])
            {
                $alokW1[] = $x[$i];
                $j++;
                $i++;
            }
            else
            {
                $i++;
            }
        }
        else
        {
            $alokW1[] = -1111111;
            $j++;
        }
    }
    return $alokW1;
}

```

```
}

```

```
function getNilai($alokW,$alokW1,$alokR,$kecil,$batasannya,$x,$fx)

```

```
{
    for($j=0;$j<=$batasannya;$j++)
    {
        for($i=0;$i<=$batasannya;$i++)
        {
            if($alokW[$i]==0)
            {
                $a = $x[$j] / $kecil;
                $nilai[$j][$i] = $fx[$a];
            }elseif($alokW[$i] > $x[$j])
            {
                $nilai[$j][$i] = -1111111;
            }else
            {
                if($nilai[($j-1)][$i] == -1111111)
                {
                    $a = 0;
                    $nilai[$j][$i] = $alokR[$i] + $fx[$a];
                }else
                {
                    $a = ($x[$j] - $alokW1[$i]) / $kecil;
                    $nilai[$j][$i] = $alokR[$i] + $fx[$a];
                }
            }
        }
    }
    return $nilai;
}

```

```
function getFx($batasannya)

```

```
{
    for($i=0;$i<=$batasannya;$i++)
    {
        $fx[] = 0;
    }

    return $fx;
}

```

```
function getFx1($nilai, $batasannya)

```

```
{
    for($i=0;$i<=$batasannya;$i++)
    {
        $fx[] = max($nilai[$i]);
    }
    return $fx;
}

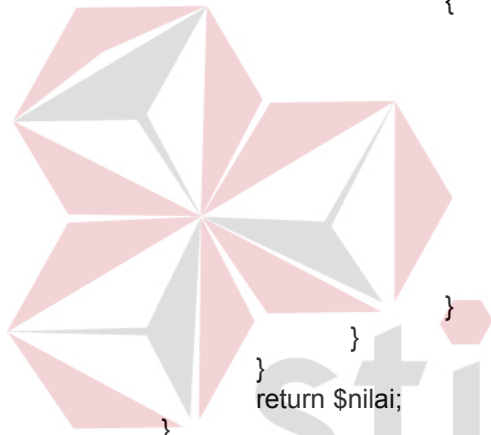
```

```
function getM($fx,$batasannya,$nilai)

```

```
{
    for($i=0;$i<=$batasannya;$i++)
    {

```



INDONESIA
& INFORMATIKA
stikom
SURABAYA

```

        $m[] = array_search($fx[$i], $nilai[$i]);
    }
    return $m;
}
}
?>

```

TabelKnapsack1.php

```

<?php
require_once "KnapsackTaha2.php";

class TabelKnapsack1 extends KnapsackTaha2
{
    function output1($W,$w,$r)
    {
        $banyak = count($w);
        $i=0;
        array_multisort($w,$r);
        $kecil = KnapsackTaha2::getKecil($w);
        $batasan = KnapsackTaha2::getBatasan($W,$kecil);
        $alokM = KnapsackTaha2::getAlokM($batasan);
        $fx = KnapsackTaha2::getFx($batasan);
        $fx1[] = $fx;
        $m[] = $fx;
        $x = KnapsackTaha2::getX($kecil,$batasan);

        while($i < $banyak)
        {
            $alokR[] = KnapsackTaha2::getAlokR($batasan,$r[$i],$alokM);
            $alokW[] = KnapsackTaha2::getAlokW($batasan,$w[$i],$alokM);
            $alokW1[] = KnapsackTaha2::getAlokW1($alokW[$i],$x,
$batasan);

            $nilai[] = KnapsackTaha2::getNilai($alokW[$i],$alokW1[$i],
$alokR[$i],$kecil,$batasan,$x,$fx1[$i]);
            $fx1[] = KnapsackTaha2::getFx1($nilai[$i],$batasan);
            $m[] = KnapsackTaha2::getM($fx1[($i+1)],$batasan,$nilai[$i]);
            $i++;
        }

        TabelKnapsack1::tabelProses($batasan,$banyak,$alokM,$alokR,$alokW,
$alokW1,$x,$fx1,$m,$nilai);
    }
}

```

```

}
function output2($W,$w,$r)
{
    $banyak = count($w);
    $i=0;
    array_multisort($w,$r);
    $kecil = KnapsackTaha2::getKecil($w);
    $batasan = KnapsackTaha2::getBatasan($W,$kecil);
    $alokM = KnapsackTaha2::getAlokM($batasan);
    $fx = KnapsackTaha2::getFx($batasan);
    $fx1[] = $fx;
    $m[] = $fx;
    $x = KnapsackTaha2::getX($kecil,$batasan);
    while($i < $banyak)
    {
        $alokR[] = KnapsackTaha2::getAlokR($batasan,$r[$i],$alokM);
        $alokW[] = KnapsackTaha2::getAlokW($batasan,$w[$i],$alokM);
        $alokW1[] = KnapsackTaha2::getAlokW1($alokW[$i],$x,
$batasan);
        $nilai[] = KnapsackTaha2::getNilai($alokW[$i],$alokW1[$i],
$alokR[$i],$kecil,$batasan,$x,$fx1[$i]);
        $fx1[] = KnapsackTaha2::getFx1($nilai[$i],$batasan);
        $m[] = KnapsackTaha2::getM($fx1[$i+1],$batasan,$nilai[$i]);
        $i++;
    }
    TabelKnapsack1::tabelAnalisis($kecil,$banyak,$batasan,$x,$m,
$fx1,$nilai,$alokR,$alokW,$alokW1);
}
function tabellInput($W,$w,$r,$banyak)
{
    echo "<h4>Kapasitas : $W</h4>";
    echo "<h2>Inputan</h2>";
    echo "<table border=1>";
    echo "<th>No.</th>";
    echo "<th>Berat</th>";
    echo "<th>Pendapatan</th>";
    for($i=0;$i<$banyak;$i++)
    {

```

```

        echo "<tr align=right><td>" . ($i+1) . "</td>";
        echo "<td>" . $w[$i] . " ton</td>";
        echo "<td>Rp. " . number_format($r[$i],0,"",'') . "</td>";
        echo "</tr>";
    }
    echo "<tr align=right style='font-weight:bold;'><td>Total : </td><td>" .
array_sum($w) . " ton</td><td>Rp. " . number_format(array_sum($r),0,"",'') .
"</td></tr></table>";
}

function tabelProses($batasan,$banyak,$alokM,$alokR,$alokW,$alokW1,$x,$fx,
$m,$nilai)
{
    echo "<h2>Proses Hitung Knapsack</h2>";
    echo "<table border=1>";
    // untuk menampilkan stage
    for($i=0;$i<$banyak;$i++)
    {
        echo "<tr align=center><td colspan=" . ($batasan + 2) . ">stage
ke-$i</td></tr>";
        echo "<tr><td>m$i : </td>";
        // buat horizontal untuk m ke-i
        foreach($alokM as $stampilan)
        {
            echo "<td>" . $stampilan . "</td>";
        }
        echo "</tr>";

        // buat horizontal untuk r ke-i
        echo "<tr><td>r$i . m$i : </td>";
        foreach($alokR[$i] as $stampilan)
        {
            echo "<td>" . $stampilan . "</td>";
        }
        echo "</tr>";

        // buat horizontal untuk w ke-i
        echo "<tr><td>w$i . m$i : </td>";
        foreach($alokW[$i] as $stampilan)
        {
            echo "<td>" . $stampilan . "</td>";
        }
        echo "</tr>";

        // buat horizontal untuk w ke-i
        echo "<tr><td>w#i . m$i : </td>";
        foreach($alokW1[$i] as $stampilan)
        {
            echo "<td>" . $stampilan . "</td>";
        }
        echo "</tr>";

        // menampilkan data vertikal
        for($k=0;$k<=$batasan;$k++)

```

```

        {
            echo "<tr><td>x$i : " . $x[$k] . "</td>";

            // menampilkan nilai
            for($l=0;$l<=$batasan;$l++)
            {
                echo "<td>" . $nilai[$l][$k][$l] . "</td>";
            }
            echo "<td style=font-weight:bold;>" . $fx[$(i+1)][$k] .

"</td>";

            echo "<td style=font-weight:bold;>" . $m[$(i+1)][$k] .

"</td>";

            echo "</tr>";
        }
    }
    echo "</table>";
}

function tabelAnalisis($kecil,$banyak,$batasan,$x,$m,$fx,$nilai,$alokR,$alokW,
$alokW1)
{
    echo "<h3>Analisis Sensitifitas</h3>";

    echo "Alokasi kapasitas kapal laut (x) satuan Ton : <select name=txtX>";
    for($i=$batasan;$i>=0;$i--)
    {
        if(isset($_POST['txtX']) && $x[$i] == $_POST['txtX'])
        {
            echo "<option selected>" . $x[$i] . "</option>";
        }else
        {
            echo "<option>" . $x[$i] . "</option>";
        }
    }

    echo "</select> <input type=submit value=Pilih>";
    echo "<br>";

    if(isset($_POST['txtX']))
    {
        $a = array_search($_POST['txtX'],$x);
        TabelKnapsack1::step2($kecil,$banyak,$a,$m,$x,$fx,$nilai,
$alokR,$alokW,$alokW1);
    }else
    {
        TabelKnapsack1::step2($kecil,$banyak,$batasan,$m,$x,$fx,
$nilai,$alokR,$alokW,$alokW1);
    }
}

function tabelOutput($brt,$pend,$jmlOpt,$pendOpt)
{
    echo "<table border=1>";
    echo "<th bgcolor=ff4848>Berat Optimal</th>";
    echo "<th bgcolor=ff4848>Pendapatan Optimal</th>";
}

```



```

echo "<th bgcolor=ff4848>Jumlah Muat</th>";
echo "<th bgcolor=ff4848>Akum. berat Opt.</th>";
echo "<th bgcolor=ff4848>Akum. Pend. Opt.</th>";
$cnt = count($brt);
$color=1;

for($i = 0; $i < $cnt; $i++)
{
    if($color==1)
    {
        echo "<tr align=right>";
        echo "<td> . $brt[$i] . " Ton</td>";
        echo "<td>Rp. " . number_format($pend[$i],0,".") .

" Ton</td>";

        echo "<td> . $jmlOpt[$i] . " item</td>";
        echo "<td style=font-weight:bold> . $jmlOpt[$i] * $brt[$i] .

number_format($pendOpt[$i],0,".") . "</td>";
        echo "<tr>";

        $color++;
    }else
    {
        echo "<tr align=right>";
        echo "<td> . $brt[$i] . " Ton</td>";
        echo "<td>Rp. " . number_format($pend[$i],0,".") .

" Ton</td>";

        echo "<td> . $jmlOpt[$i] . " item</td>";
        echo "<td style=font-weight:bold> . $jmlOpt[$i] * $brt[$i] .

echo "<td style=font-weight:bold>Rp. " .
number_format($pendOpt[$i],0,".") . "</td>";

        $color--;
    }
}
}

```

```

function step1($banyak,$batasan,$m)
{
    $a = $banyak;

    while($a > 0)
    {
        if($m[$a][$batasan] <= 0)
        {
            $a--;
        }else
        {
            break;
        }
    }
    $a--;
    return $a;
}

```

```

}

function step3($d,$a,$fx)
{
    $f = array_search($d,$fx[$a]);
    return $f;
}

function step20($a,$batasan,$alokR,$fx)
{
    $b = array_search($fx[($a+1)][$batasan],$alokR[$a]);
    return $b;
}

function getPendOpt($a,$batasan,$fx)
{
    $pendOpt = $fx[($a+1)][$batasan];
    return $pendOpt;
}

function getJmlOpt($a,$batasan,$m)
{
    $jmlOpt = $m[($a+1)][$batasan];
    return $jmlOpt;
}

function getPendapatan($a,$alokR)
{
    $pend = $alokR[$a][1];
    return $pend;
}

function getBerat($a,$alokW)
{
    $brt = $alokW[$a][1];
    return $brt;
}

function step21($a,$batasan,$nilai,$fx)
{
    $c = array_search($fx[($a+1)][$batasan],$nilai[$a][$batasan]);
    return $c;
}

function step22($batasan,$a,$x,$m,$alokW1,$kecil)
{
    $c = $m[($a+1)][$batasan];
    $d = floor(($x[$batasan] - $alokW1[$a][$c]) / $kecil);

    return $d;
}

function step2($kecil,$banyak,$batasan,$m,$x,$fx,$nilai,$alokR,$alokW,
$alokW1)

```

```

{
    $a = $banyak;
    $c = $batasan;

    while($a>=0)
    {
        $a = TabelKnapsack1::step1($a,$c,$m);

        $b = TabelKnapsack1::step20($a,$c,$alokR,$fx);

        if(empty($b))
        {
            $pendOpt[] = TabelKnapsack1::getPendOpt($a,$c,$fx);
            $jmlOpt[] = TabelKnapsack1::getJmlOpt($a,$c,$m);
            $brt[] = TabelKnapsack1::getBerat($a,$alokW);
            $pend[] = TabelKnapsack1::getPendapatan($a,$alokR);

            $c = TabelKnapsack1::step22($c,$a,$x,$m,
                $alokW1,$kecil);
            $a--;
        }else
        {
            $pendOpt[] = TabelKnapsack1::getPendOpt($a,$c,$fx);
            $jmlOpt[] = TabelKnapsack1::getJmlOpt($a,$c,$m);
            $brt[] = TabelKnapsack1::getBerat($a,$alokW);
            $pend[] = TabelKnapsack1::getPendapatan($a,$alokR);

            break;
        }
    }
    TabelKnapsack1::tabelOutput($brt,$pend,$jmlOpt,$pendOpt);
    /*
    echo "total pendapatan : ";
    print_r($pendOpt);
    echo "<br>jumlah optimal : ";
    print_r($jmlOpt);
    echo "<br>berat optimal : ";
    print_r($brt);
    echo "<br>pendapatan optimal : ";
    print_r($pend);*/
    //echo "terdapat variabel optimal pada stage ke-$a, baris ke-$b";
}
?>

```