



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB 3

METODE PENELITIAN

3.1 Gambaran Umum Objek Penelitian

3.1.1 Sejarah Perusahaan

PT. Kiat Ananda adalah sebuah perusahaan yang berpengalaman dan handal yang menawarkan jasa manajemen untuk produk makanan beku dan produk lainnya. Didirikan di Jakarta pada tahun 1997, PT. Kiat Ananda berusaha menyediakan solusi rantai manajemen pasokan yang terintegrasi dan layanan konsultasi, dari proses membeli barang-barang untuk pemasok (impor atau lokal) dari pabriknya, pengiriman barang dari pemasok untuk klien, serta memiliki ruang penyimpanan dingin dan distributor barang dari klien ke distributor dan pengecer yang mereka miliki.

Selama bertahun-tahun, PT. Kiat Ananda dioperasikan oleh orang-orang yang berpengalaman dan kompeten dibidangnya masing-masing. Perusahaan juga menawarkan sistem jasa konsultasi *cold supply chain* dengan dukungan dari gudang yang memiliki berpendingin modern, sistem manajemen gudang yang profesional, dan armada truk serta truk yang memiliki pendingin yang dapat diandalkan.

Karena PT. Kiat Ananda murni terlibat dalam *supply chain management* dan tidak memproduksi, PT. Kiat Ananda dapat memberikan layanan profesional yang independen tanpa konflik kepentingan. Dengan

daftar proyek-proyek yang sukses, PT. Kiat Ananda telah dipercaya oleh berbagai klien multinasional sejak berdiri.

Alasan peneliti memilih perusahaan ini adalah kemudahan dalam pengumpulan serta pengambilan data di perusahaan tersebut.

3.1.2 Visi dan Misi Perusahaan

Visi

- Untuk menjadi penyedia terbaik *cold supply chain management* di Indonesia

Misi

- Memberikan layanan dan solusi terbaik bagi pelanggan sesuai dengan kebutuhan.
- Memberikan kesempatan kesempatan kepada karyawan yang berkualitas dan terampil dengan dukungan dari sistem yang handal dan sarana kerja yang lengkap untuk menjamin keberhasilan proyek.
- Sumber daya terbaik akan menjamin kelangsungan pertumbuhan perusahaan serta kesejahteraan stakeholder.

3.2 Penelitian Sebelumnya

Dalam mengembangkan perencanaan pencarian ruang kosong kontainer ini peneliti menemukan 3 penelitian sejenis yaitu Optimasi Penataan Silinder Dalam Kontainer Dengan Algoritma Genetika oleh Novita Wulan Sari dari Kampus ITS Keputih Sukolilo, Surabaya.

Unsur penting dalam memuat barang didalam kontainer adalah jenis dari barang serta ukuran dari barang dan kontainer itu sendiri

dengan tujuan mengisi ruang kosong didalam kontainer secara maksimal dengan ketentuan yang telah ditetapkan. Apabila isi kontainer memiliki banyak ruang kosong, maka ada barang yang tertinggal dan tidak bisa dimasukkan ke dalam kontainer sehingga perhitungan/*planning* yang telah ditentukan menjadi meleset dan menyebabkan untuk melakukan pemesanan kembali kontainer yang seharusnya tidak diperlukan.

Dari penelitian diatas diketahui bahwa Novita meneliti optimasi barang yang berbentuk silinder untuk masuk ke dalam kontainer. Penelitian tersebut tidak jauh berbeda dengan yang sedang dilakukan oleh peneliti yang membedakan adalah bentuk barang yang diujikan.

Kemudian peneliti menemukan penelitian lainnya yang berjudul Optimasi Pola Penyusunan Barang Dalam Ruang Tiga Dimensi Menggunakan Metode Genetic Algorithm oleh Kartika Gunadi dan Irwan Kristanto Julistiono dari Universitas Kristen Petra.

Dari penelitian diatas peneliti mengetahui bahwa mereka membuat pencarian ruang kosong tetapi tidak memakai batasan masalah dan tidak ditempatkan kedalam kontainer melainkan dapat menentukan sendiri luas ruangan yang akan diujikan. Hal tersebut dapat memicu berbagai hasil yang banyak karena tidak memiliki batasan luas ruangan dan barang tersebut.

Penelitian selanjutnya dengan judul An Efficient Algorithm for 3D Rectangular Box Packing oleh M.Zahid Gurbuz dari Yildiz Technical University.

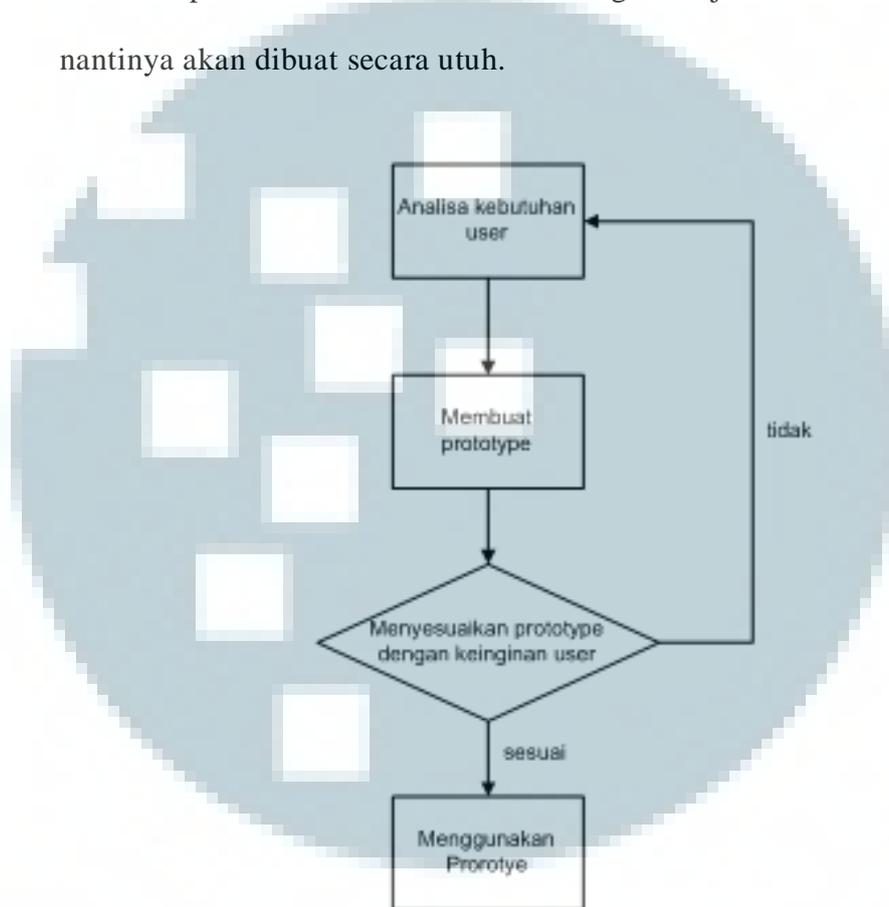
Berdasarkan penelitian diatas peneliti mengetahui bahwa untuk melakukan meminimalisir ruang kosong dengan menentukan barang yang besar terlebih dahulu untuk dimasukkan pada posisi yang pertama dan kemudian berlanjut sampai ke ukuran yang paling kecil. Metode algoritma ini disebut *Largest Area Fit-First (LAFF)*. Peneliti menggunakan metode ini dengan catatan peneliti mengembangkan model simulasi yang tidak dibuat dari penelitian diatas. Penelitian ini hanya menghitung berapa barang yang bisa masuk ke dalam kontainer tetapi tidak melakukan simulasi barang apa saja yang masuk.

3.3 Metode Penelitian

Perancangan optimasi desain pencarian ruang kosong dalam kontainer dengan metode *LAFF*. Dengan menggunakan metode tersebut kita dapat menentukan diameter dari barang berupa *box* dan luas kontainer yang dapat kita tentukan sendiri. Berdasarkan hasil inputan tersebut maka akan menampilkan berapa banyak *box* yang dapat dimuat dalam satu kontainer.

Alasan peneliti memilih metode ini adalah dengan melakukan input berupa diameter *box* dan kontainer metode tersebut dapat menentukan barang-barang yang memiliki diameter paling besar dapat dimasukkan pertama kali dan akan mencari ruang kosong sesuai dengan ukuran barang sampai yang paling kecil yang dimuat sehingga hal tersebut sesuai dengan tujuan penulis dan untuk meminimalisir ruang kosong. Selain metode algoritma *LAFF* peneliti juga menggunakan metode *prototype*. Alasan peneliti menggunakan metode *prototype*

adalah pelanggan dan pembuat dapat berperan langsung dalam mendesain sistem dan pengembangan sistem. Model yang dibuat berbentuk sementara sehingga jika ada tambahan/kekurangan/tidak cocok dapat dibenarkan dan dibuat lagi menjadi versi final yang nantinya akan dibuat secara utuh.



Gambar 3.1 Model *prototype*

U M M N

Tabel 3.1 Perbandingan *Prototype* vs *Spiral*

	<i>Prototype</i>	<i>Spiral</i>
Kelebihan	Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan	Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar.
	Lebih menghemat waktu dalam pengembangan system	Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi permasalahan yang serius.
	Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya	Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja iteratif.
Kekurangan	Banyak ketidaksesuaian pada bentuk <i>prototype</i>	Memerlukan tenaga ahli untuk memperkirakan resiko, dan harus mengandalkannya supaya sukses.
	<i>Prototype</i> yang disetujui oleh client harus dikembangkan tanpa ada tambahan data dari client	Butuh waktu lama untuk menerapkan paradigma ini menuju kepastian yang absolute.

Tabel 3.2 Perbandingan *LAFF* vs Algoritma Genetika

<i>LAFF</i>	Algoritma Genetika
Hanya memerlukan perhitungan panjang, lebar dan tinggi dari <i>box</i>	Mebutuhkan perhitungan yang rumit seperti menghitung <i>fitness cost</i>
Menentukan letak barang dengan mengisi ruang kosong pada bagian sisi luar kontainer terlebih dahulu dan mengisi sisi bawah lalu sisi samping	Menentukan letak barang dengan menggunakan koordinat dari hasil perhitungan dan bersifat random
Waktu proses tidak memakan banyak waktu	Waktu proses memakan banyak waktu lebih lama

3.4 Variable Penelitian

a. Independen (dimensi barang)

Variabel independen yang digunakan dalam penelitian ini adalah dimensi barang yang bersifat fleksibel dengan ukuran yang dapat dimasukkan sesuai keinginan *user*.

b. Dependen (luas kontainer)

Variabel dependen yang digunakan dalam penelitian ini adalah luas didalam kontainer tersebut karena luas tidak akan berubah-ubah karena luas dalam kontainer bersifat statis dan dimensi kontainer yang telah ditentukan yaitu sebesar 20 *feet*.

3.5 Teknik Pengumpulan Data

a. Wawancara

Pada metode ini, peneliti melakukan wawancara langsung kepada Bapak Mave selaku *Head Planner* yang bertugas membuat perencanaan muatan barang ke dalam kontainer dan peneliti mendapatkan informasi atau data-data seperti frekuensi keluar masuk kontainer dan perkiraan barang yang dimasukkan kedalam suatu kontainer.

b. Observasi

Pada metode ini, peneliti melakukan observasi ke perusahaan PT. Kiat Ananda Solusindo yang berada di Jalan Raya Narogong KM 12,5, Pangkalan 5 No 7, Cileungsi, Jawa Barat untuk melihat secara langsung alur dari bagaimana *Head Planner* melakukan perencanaan dan melihat secara langsung penempatan barang ke dalam kontainer.

3.6 Teknik Simulasi Data

Untuk mengetahui jalannya proses simulasi penempatan barang diruang kosong kontainer, diperlukan tampilan serta perhitungan dari luas kontainer dan luas barang agar dapat menghasilkan suatu simulasi yang sesuai dengan *user input*.

Untuk mengecek hasil pencarian ruang kosong untuk diisi barang, ditampilkan simulasi grafik 2D dengan variabel yang dimasukkan.

BAB 4

ANALISIS DAN HASIL PENELITIAN

4.1 Analisa Kebutuhan User

Pada tahap ini penulis dimulai dengan pengumpulan kebutuhan. Penulis melakukan analisis kebutuhan dan mengidentifikasi segala kebutuhan.

4.1.1 Analisa Sistem yang Berjalan

Sistem yang berjalan saat ini di PT. Kiat Ananda masih menggunakan sistem manual yaitu membuat gambar berupa kontainer dan barang-barang yang nantinya akan dimasukkan sesuai gambar.

4.1.2 Identifikasi Masalah

Berdasarkan hasil analisis pada sistem berlangsung, terdapat masalah yang dihadapi yaitu adanya ruang kosong yang tidak dapat dimaksimalkan karena pengepakan barang ke dalam kontainer tidak sesuai dengan gambar yang dibuat tidak sesuai ukuran aslinya.

4.1.3 Usulan Penyelesaian Masalah

Dengan melihat permasalahan yang ada maka di butuhkan suatu sistem yang dapat membantu mengatasi masalah ruang kosong yang dapat diminimalisir.

Adapun aktor yang terlibat pada sistem ini adalah *Inventory*, *Planner* dan *Checker*. *Inventory* adalah orang yang dapat mengecek lokasi penyimpanan, mengecek jumlah barang dan menghitung ruang kosong secara manual dilokasi tersebut. *Planner* adalah orang melakukan pengecekan jumlah barang *by sistem* dan *follow up* case baik pengiriman maupun penerimaan. *Checker* adalah orang yang memeriksa, menghitung, mencocokkan barang dengan dokumen dan membuat berita acara apabila ada barang yang janggal. Serta melakukan pemuatan barang.

Pengumpulan kebutuhan dapat dijelaskan dalam bentuk aktivitas-aktivitas *use case* berikut :



Gambar 4.1 Use case diagram

4.2 Membuat *Prototype*

Setelah mendapatkan serta mengetahui seluruh kebutuhan yang diperlukan dalam pengembangan sistem, maka dilakukan tahap pembuatan *prototype*. Tahap pertama yaitu mendesain *user interface* dari kontainer dan

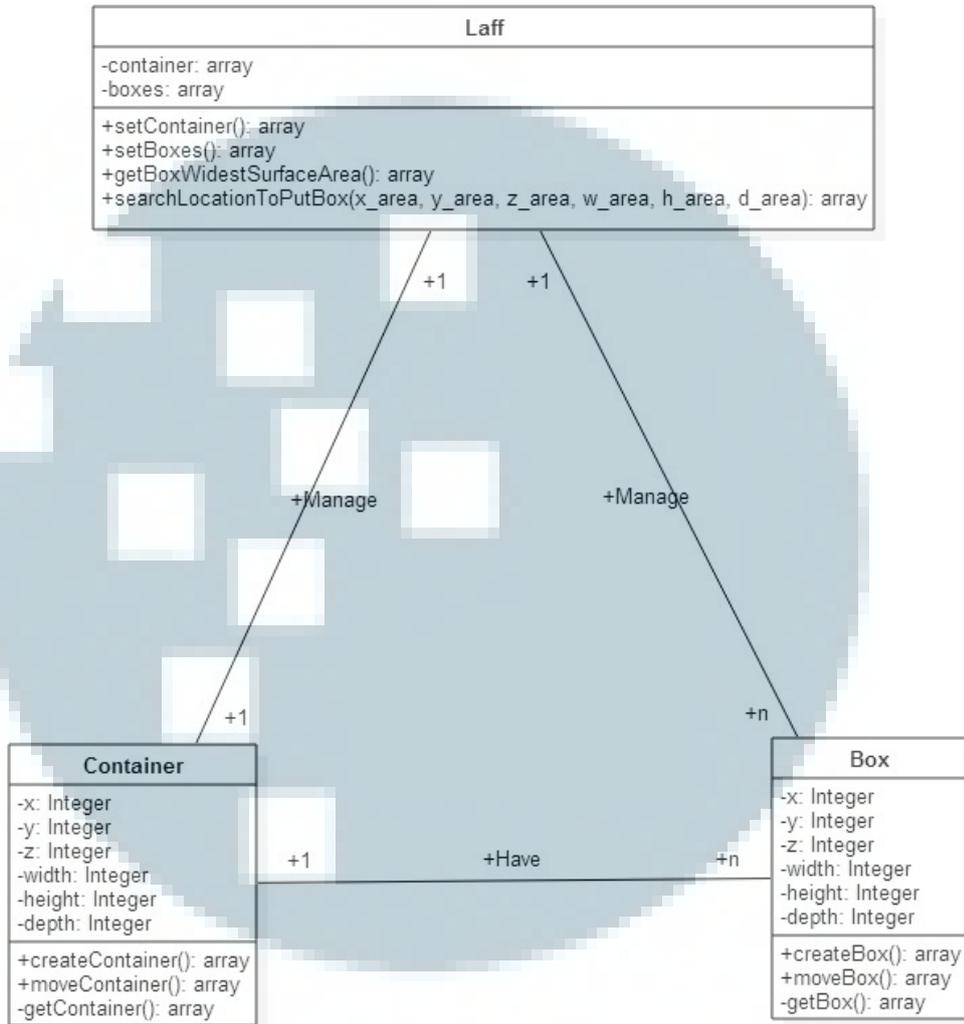
box yang akan ditambahkan di sistem. Didalam tahap ini *user* dapat menambahkan ukuran kontainer dan ukuran *box* dengan bervariasi dengan menggunakan perbandingan skala dikarenakan ukuran laptop yang tidak terlalu besar sehingga tidak dapat menampilkan ukuran pada aslinya. Tahap selanjutnya adalah membuat simulasi penempatan barang. Dalam tahap ini setelah *user* menambahkan ukuran kontainer dan *box* maka ketika diklik tombol *calculate* di web tersebut akan menampilkan gambar kontainer dengan barang-barang yang telah disusun oleh algoritma *LAFF*.

4.2.1 *Class Diagram*

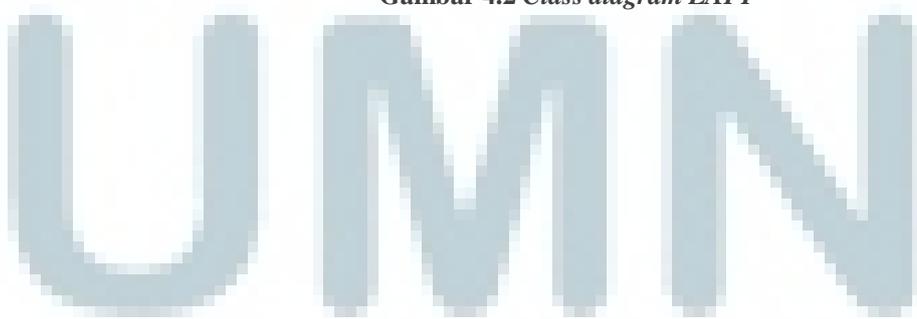
Dalam aplikasi ini terdapat input dan fungsi dari masing-masing pihak penggunanya. Berikut adalah gambaran class diagram yang akan dibuat:

U M N

Class Diagram Container Dan Box Dengan Laff



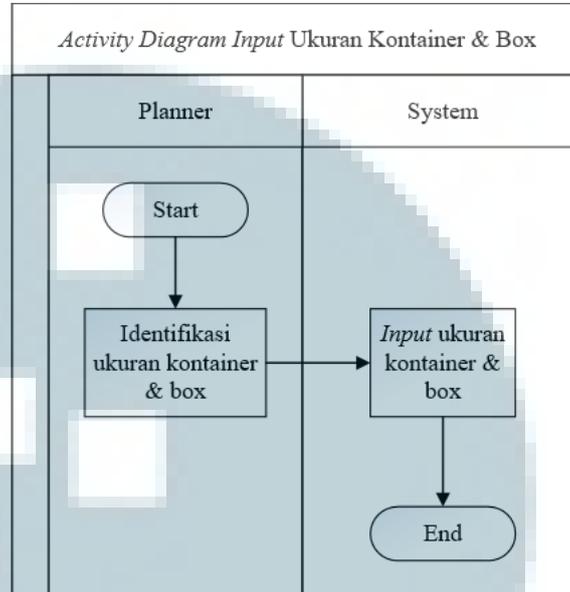
Gambar 4.2 Class diagram LAFF



4.2.2 Activity Diagram

a. Activity Diagram Input Ukuran Kontainer Dan Box

Tabel 4.1 *Input ukuran kontainer dan box*

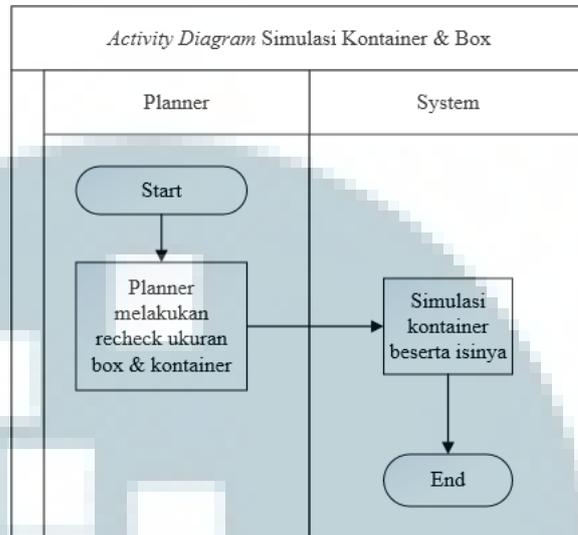


Dalam *activity diagram* diatas ini menjelaskan urutan aktifitas yang dilakukan oleh *Planner*. Adapun langkah-langkah yang harus dilakukan oleh *Planner* adalah mengidentifikasi ukuran barang dan kontainer kemudian melakukan *input* ukuran kontainer dan box di *website*.

U
M
M
N

b. Activity Diagram Simulasi Kontainer Dan Box

Tabel 4.2 Simulasi kontainer dan box

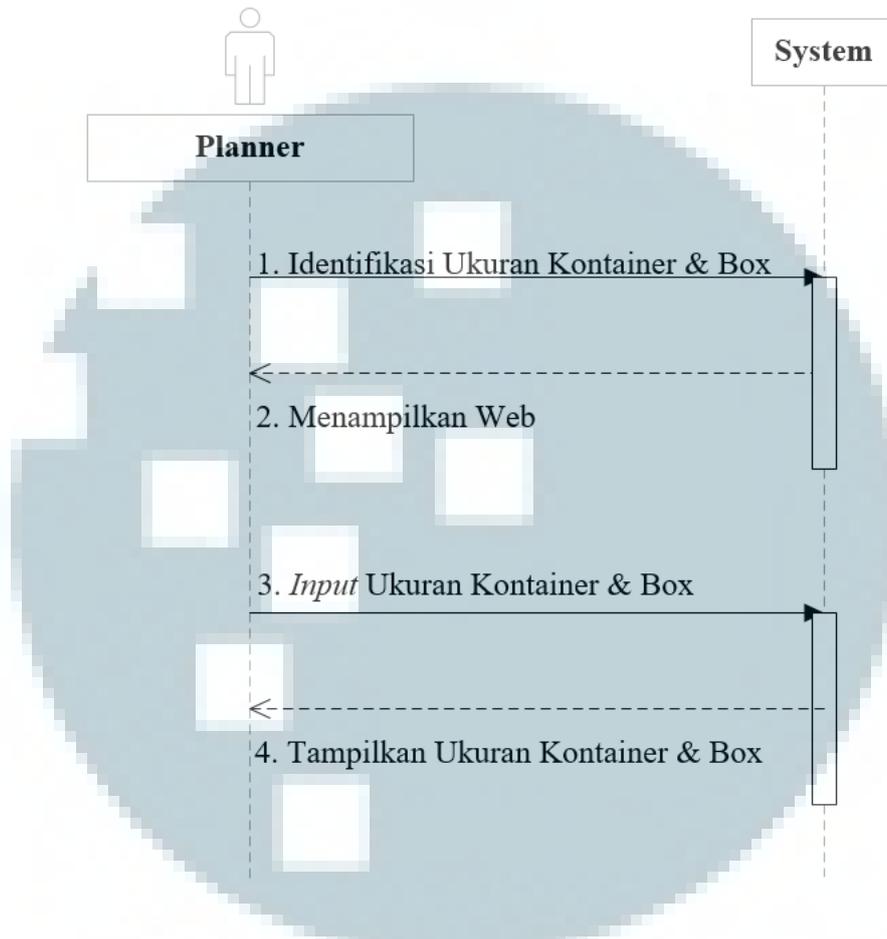


Dalam melakukan simulasi kontainer, *Planner* akan melakukan pengecekan ulang terhadap ukuran barang dan kontainer. Kemudian sistem akan melakukan simulasi ketika *Planner* melakukan klik *button calculate*.

U
M
M
N

4.2.3 Sequence Diagram

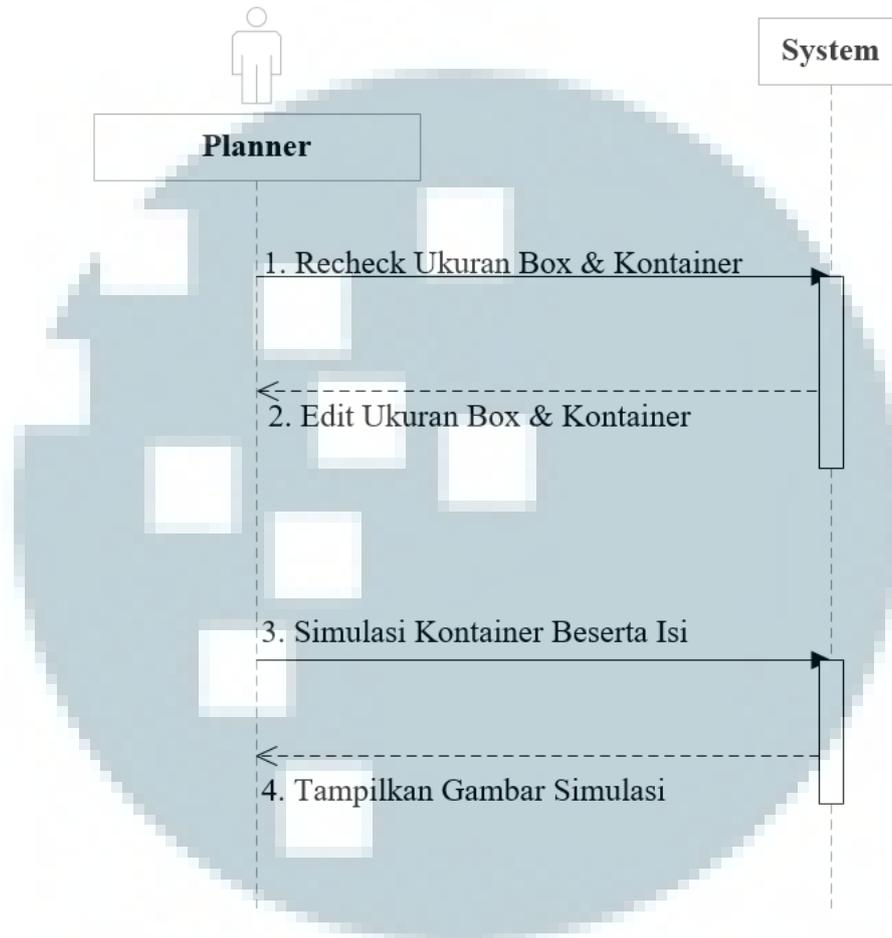
a. Sequence Diagram Planner



Gambar 4.3 Sequence diagram planner dengan sistem

Keterangan gambar diatas: ketika ada perintah untuk melakukan simulasi kontainer maka *Planner* akan melakukan identifikasi ukuran dari barang dan kontainer. Setelah itu sistem menampilkan *web* untuk melakukan simulasi dan *input* ukuran barang serta kontainer. Selanjutnya masukkan ukuran barang dan kontainer dan sistem akan menampilkan ukuran kontainer dan barang

setelah itu web akan mengkonfirmasi ukuran barang dan kontainer.



Gambar 4.4 *Sequence diagram planner dengan sistem bagian 2*

Keterangan gambar diatas: setelah melakukan *input* ukuran barang dan kontainer, *Planner* melakukan pengecekan kembali mengenai ukuran barang dan kontainer dan jika terjadi kesalahan maka sistem akan menampilkan layar edit untuk mengganti ukuran barang. Setelah itu dapat dilakukan simulasi oleh *Planner* dan sistem akan menampilkan hasil gambar simulasi tersebut.

4.2.4 Desain Interface

1. Design User Interface Perhitungan Ruang Kosong

Keterangan gambar dibawah: gambaran ui yang ditampilkan pada web terdapat *field* kosong yang harus diisi, yaitu: ukuran kontainer dan ukuran barang. Setelah data tersebut diisi, dapat di klik *button calculate* yang nantinya akan menampilkan output berupa tampilan simulasi gambar yang telah disesuaikan ukuran kontainer dan ukuran barangnya.

The screenshot shows a web interface titled "Box Dimension". It features a 3D wireframe diagram of a box on the left. To the right, there are two rows of input fields: "Box Container" and "Box Item". Each row has three input fields labeled "Width (cm)", "Height (cm)", and "Depth (cm)", with "x" symbols between them. A blue "Add Box" button is located to the right of the "Box Item" row. Below these fields is a table with three columns: "Item Name", "Width x Height x Depth", and "Action". At the bottom of the form, there are two buttons: a green "Calculate" button and an orange "Reset" button. The interface is set against a blue header and footer. A large, semi-transparent "UMN" watermark is overlaid on the entire image.

Gambar 4.5 Design user interface perhitungan ruang kosong

2. Proses Perencanaan Ruang Kosong

Ketika mengakses web www.bungkus.ml maka akan ditampilkan gambaran seperti gambar 4.5. Terdapat keterangan

box container yaitu diameter dari kontainer yang telah ditentukan dan di isi secara manual. Selain itu terdapat keterangan *box item* yaitu diameter dari luas barang/*box* dan dapat di tambah sesuai dengan keinginan *user*.

The screenshot shows a web application interface for calculating box dimensions. It features a blue header with the title 'Box Dimension' and a trash icon. Below the header, there is a 3D wireframe diagram of a box. The main input area contains two rows of dimension fields: 'Box Container' with Width (cm) 2, Height (cm) 2, and Depth (cm) 2; and 'Box Item' with Width (cm) 1, Height (cm) 1, and Depth (cm) 1. An 'Add Box' button is located to the right of the 'Box Item' fields. Below the input fields is a table with the following data:

Order Item Name	Width x Height x Depth	Action
A	1 cm x 1 cm x 1 cm	 
B	1 cm x 1 cm x 1 cm	 

At the bottom of the input section, there are two buttons: 'Calculate' (green) and 'Reset' (orange). Below this is a blue bar with a calculator icon and the text 'Calculate Results'.

Gambar 4.6 Tampilan ketika *box* dan kontainer di *input*

Dibawah ini adalah gambar ketika melakukan *testing* web dan hasil simulasi dari yang telah di *test*. Akan ada keterangan *Packed box* berisi total *box* yang dimasukkan, total volume dari kontainer, dan sisa ruang kosong dari kontainer. Selain itu ada keterangan *remaining box* berisi sisa *box* yang tidak dapat dimasukkan ke dalam kontainer dan total volume dari *box* tersebut. Warna hijau adalah warna dari *box* yang akan

dimasukkan, dan warna merah adalah warna dari kontainer tersebut.

Box Dimension

Width (cm) Height (cm) Depth (cm)

Box Container 2 x 2 x 2

Box Item Width (cm) x Height (cm) x Depth (cm) **Add Box**

Order Item Name	Width x Height x Depth	Action
A	1 cm x 1 cm x 1 cm	
B	1 cm x 1 cm x 1 cm	
C	1 cm x 1 cm x 1 cm	
D	1 cm x 1 cm x 1 cm	

Calculate Results

Packed Boxes
Packed boxes : 4
Total Volume : 8 cm³
Wasted Space : 4 cm³

Level : 0

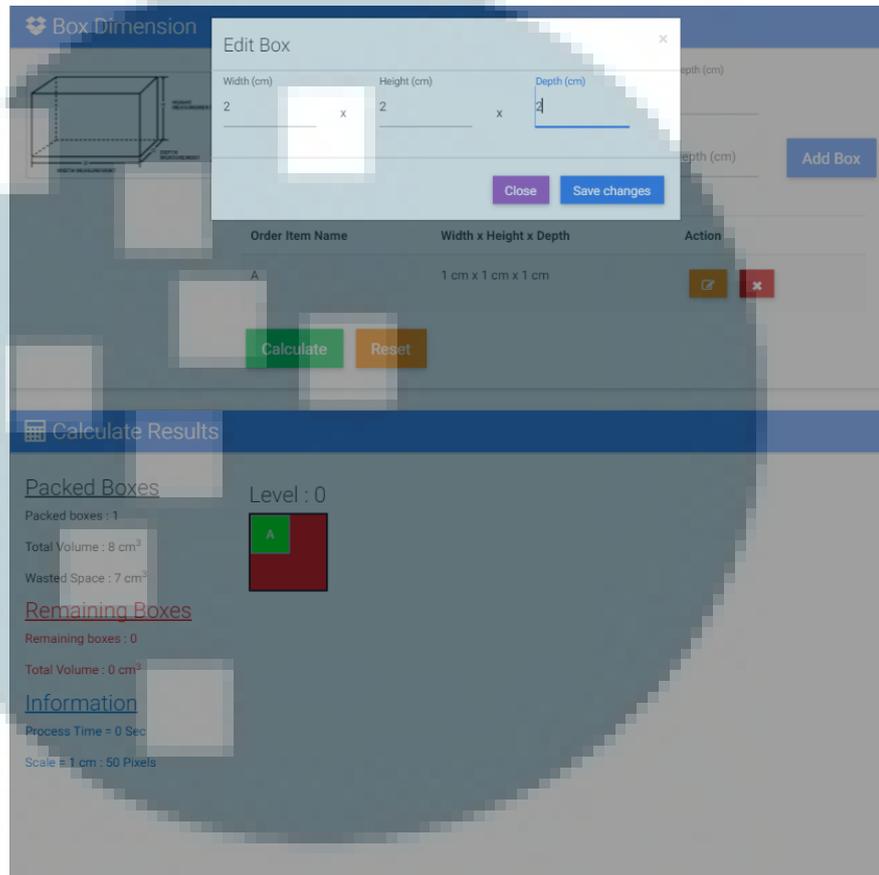
A	C
B	D

Remaining Boxes
Remaining boxes : 0
Total Volume : 0 cm³

Information
Process Time = 0 Sec
Scale = 1 cm : 50 Pixels

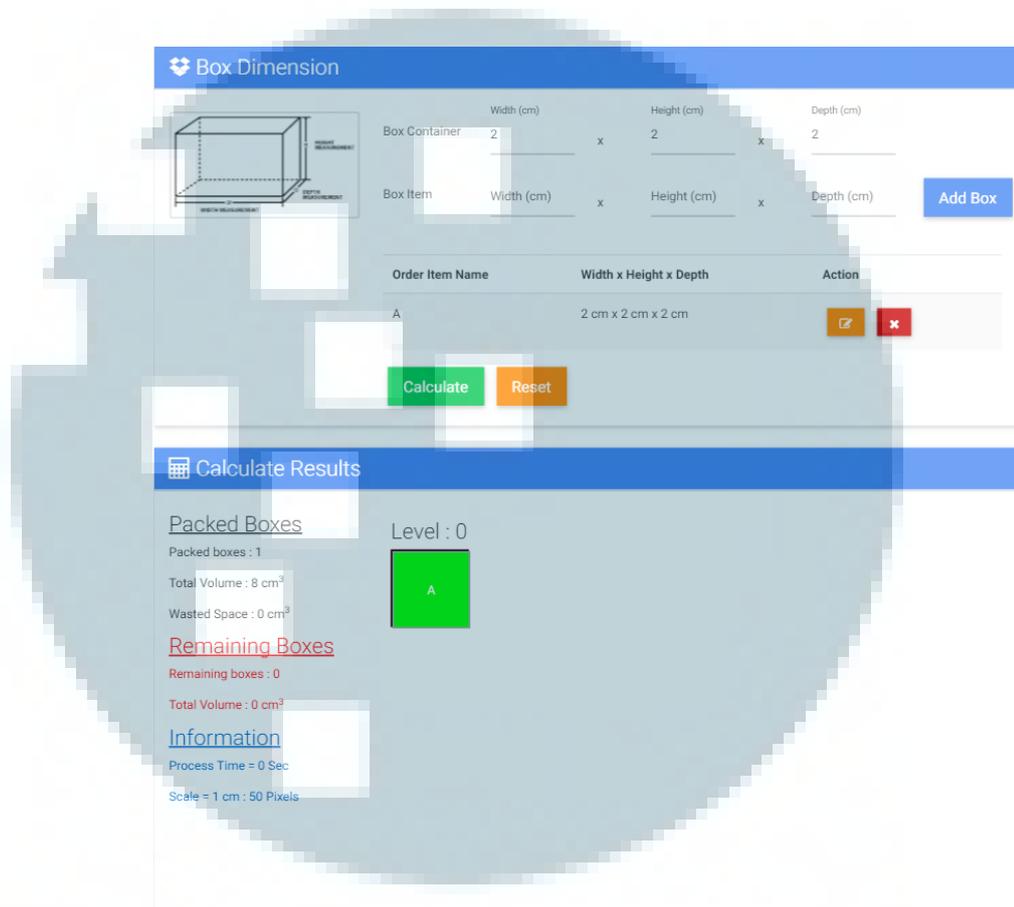
Gambar 4.7 Hasil test

Ketika tombol *action edit* yang terdapat pada keterangan di gambar 4.7 dibagian sebelah kiri dari *icon X* di tekan maka akan muncul suatu *pop-up* yang berisi *user* dapat melakukan perbaikan/*edit* diameter dari *box* tersebut. Kemudian klik *save changes* untuk menyimpan hasil *edit* dan keterangan *Width, Height, Depth Box* akan berubah.



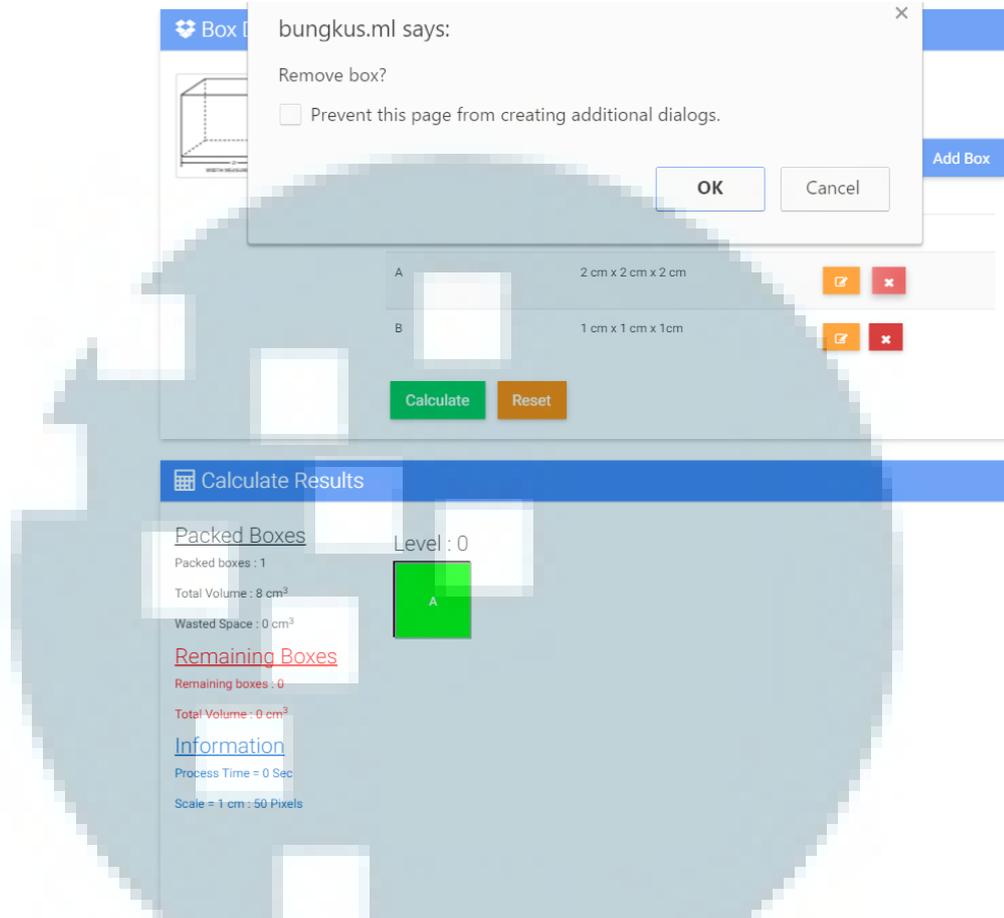
Gambar 4.8 Edit box

Ketika telah dilakukan klik *save changes* hal itu tidak merubah pada gambar simulasi sebelum ukuran *box* atau kontainer diubah. Diperlukan untuk melakukan klik tombol *calculate* lagi untuk merubah gambar simulasi yang sudah di *edit* seperti gambar dibawah ini.



Gambar 4.9 Hasil edit box

Disebelah tombol *action edit* terdapat tombol *action delete* dengan *icon* berbentuk X yang digunakan untuk melakukan penghapusan barang ketika ada barang yang salah dimasukkan tetapi berjumlah sedikit. Ketika melakukan klik pada *icon X* akan muncul suatu *pop-up* yang gunanya untuk memastikan apakah barang tersebut benar-benar dihapus atau tidak.



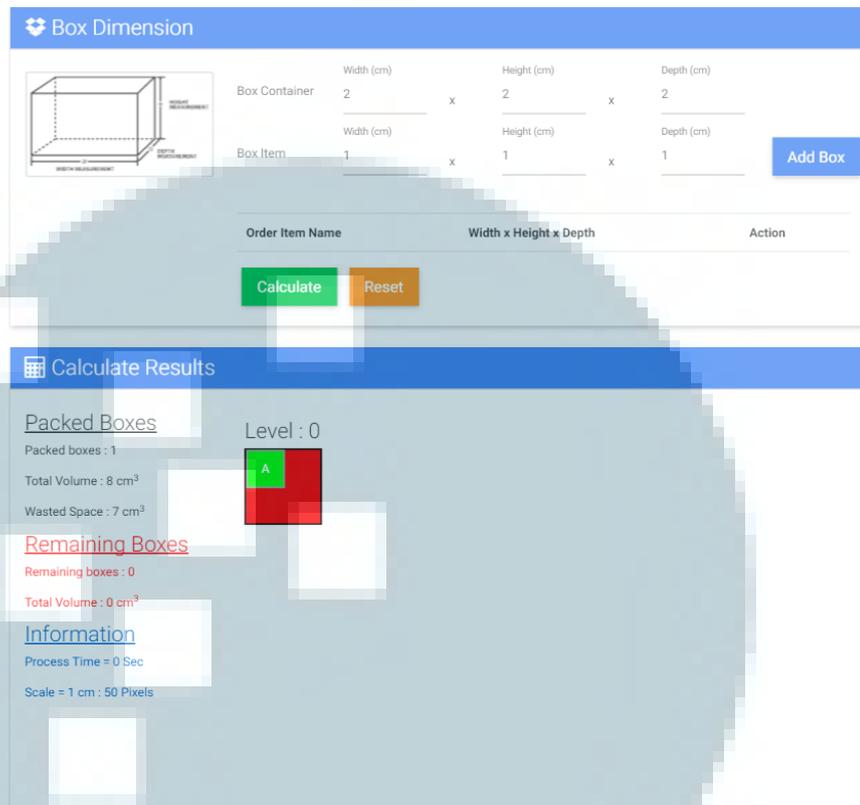
Gambar 4.10 Ketika *icon x* diklik

Setelah melakukan penghapusan urutan barang akan menjadi naik 1 abjad dari huruf sebelumnya.

U M N

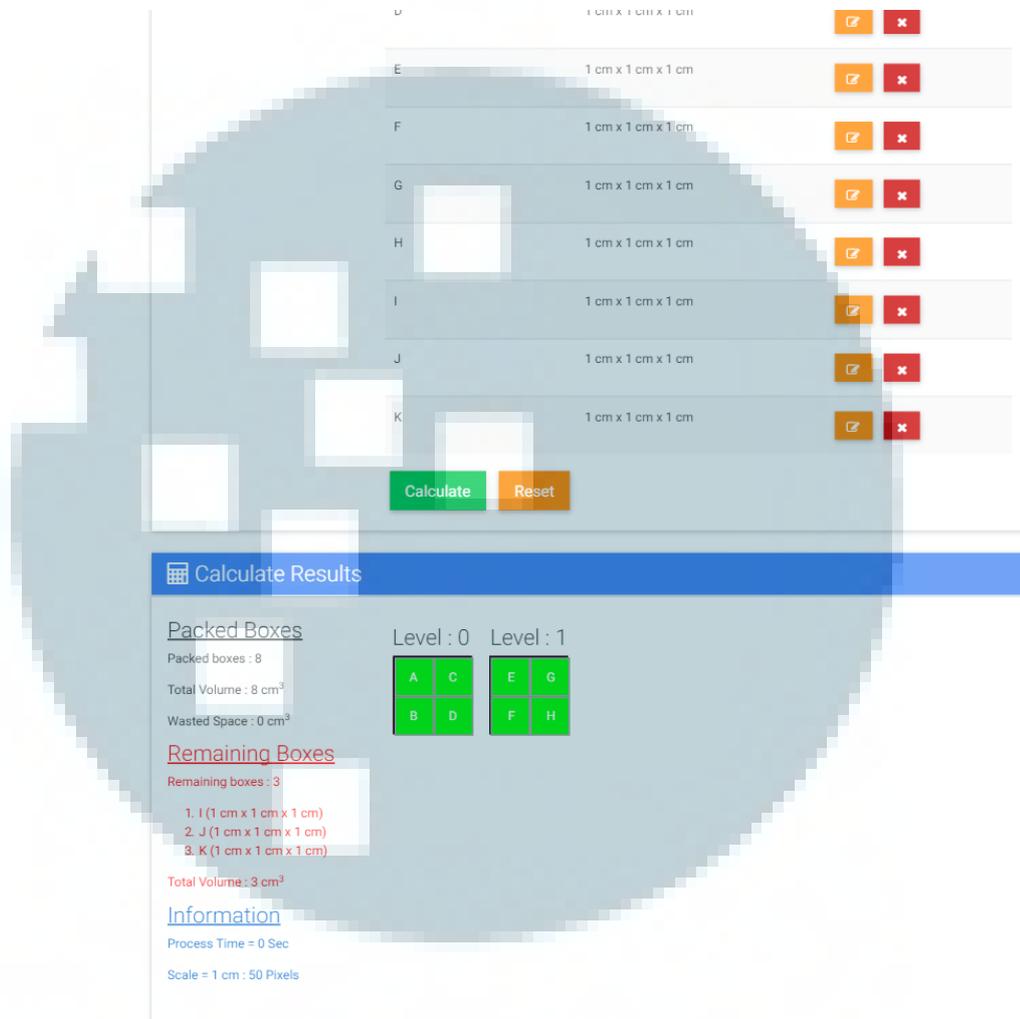
Gambar 4.11 Ketika berhasil dihapus barangnya

Dari UI diatas terdapat tombol *Reset* yang akan menghapus semua *order* barang yang pernah dimasukkan. Ketika di klik maka akan langsung menghapus tanpa menampilkan suatu *pop-up*.



Gambar 4.12 Ketika tombol *reset* diklik

Dalam melakukan perhitungan dan simulasi ini diameter *box* tidak boleh lebih besar daripada diameter kontainer. Jika terjadi ukuran barang lebih besar di banding kontainer maka akan ada barang-barang yang tidak dapat dimasukkan ke dalam simulasi yang nantinya akan diberi keterangan di bagian *remaining box* dengan keterangan berwarna merah seperti gambar dibawah ini.



Gambar 4.13 Hasil test diameter *box* lebih besar dari kontainer

4.2.5 Coding

Salah satu *coding* untuk membuat tampilan `bungkus.ml` adalah untuk menentukan letak barang. Gambar dibawah ini menunjukkan *coding* untuk meletakkan barang tersebut.

```

BOX_CHECK ← Boxes[X]

PBC ← PANJANG_BOX_CHECK

LBC ← LEBAR_BOX_CHECK

TBC ← TINGGI_BOX_CHECK

PC ← PANJANG_CONTAINER
LC ← LEBAR_CONTAINER
TC ← TINGGI_CONTAINER

IF PC <= PBC AND <= LBC AND TC <= TBC THEN
    IF BOX_PILIHAN <> NULL THEN
        LUAS_BOX_CHECK ← PBC * LBC
        LUAS_BOX_PILIHAN ← PANJANG_BOX_PILIHAN
        * LEBAR_BOX_PILIHAN
        IF LUAS_BOX_CHECK > LUAS_BOX_PILIHAN
        THEN
            BOX_PILIHAN = BOX_CHECK
        ELSE
            IF LUAS_BOX_CHECK = LUAS_BOX_PILIHAN
            THEN
                IF TBC < TINGGI_BOX_PILIHAN THEN
                    BOX_PILIHAN = BOX_CHECK
                ENDIF
            END IF
        ENDIF
    ENDIF
ENDIF

```

```
ELSE
    BOX_PILIHAN = Boxes[X]
ENDIF
X ← X + 1
ENDWHILE
```

4.3 Menggunakan *Prototype*

Pada tahap ini, dilakukan pengujian *prototype*, sebelumnya penulis menganalisis spesifikasi kebutuhan komputer/laptop yang dapat digunakan untuk menerapkan aplikasi ini, pengujian aplikasi pada laptop dengan syarat-syarat yang telah terpenuhi yaitu, laptop Lenovo, untuk melakukan kinerja aplikasi pada laptop tersebut.

4.3.1 Spesifikasi Laptop/Komputer yang digunakan

Laptop/Komputer yang digunakan untuk menjalankan aplikasi ini harus memiliki spesifikasi minimal sebagai berikut:

- a. Mempunyai *browser*
- b. Mendukung kinerja php

Dalam mengimplementasikan aplikasi yang dilakukan oleh penulis, laptop yang digunakan adalah Lenovo Z40. Laptop tersebut telah memenuhi kebutuhan spesifikasi minimal laptop yang digunakan yang telah disebutkan diatas.

4.3.2 Pengujian aplikasi pada laptop Lenovo Z40

Pada bagian ini akan dibahas pengujian terhadap perancangan aplikasi minimalisir ruang kosong. Pengujian aplikasi dilakukan dengan melakukan *blind test*. *Blind test* adalah serangkaian tes yang dilakukan secara acak hingga mengeluarkan hasil.

4.4 Testing

Pada tahap ini, peneliti melakukan *User Acceptance Testing* (UAT) dari sistem yang telah dibuat oleh peneliti. UAT ini sendiri dilakukan oleh *Head Planner* yang telah melakukan *testing* melalui sistem yang telah dibuat oleh peneliti dan dapat diakses pada www.bungkus.ml. Setelah penulis melakukan *testing* keseluruhan sistem, dapat dinyatakan bahwa sistem simulasi dapat berjalan dengan lancar dengan menampilkan gambar dan dapat ditambahkan ukuran *box* serta kontainer tersebut. UAT yang dilakukan pada tanggal 12 Juli 2016, memberikan kesimpulan bahwa sistem yang sudah dibuat oleh peneliti dapat digunakan secara keseluruhan sesuai dengan tujuan peneliti.

Selain itu penulis juga melakukan test sebanyak 100x untuk melihat probabilitas errornya. Test dilakukan dengan menentukan ukuran barang secara random dan total barang yang dimasukkan juga random. Hasilnya berupa seberapa banyak ruang kosong (*waste space*) yang akan dibandingkan dengan 2 penelitian sebelumnya yang dibuat oleh Novita dan

M. Gurbuz serta hasil sisa ruang kosong dari *Head Planner*. Hasil dari test seperti pada gambar dibawah ini.

Tabel 4.3 Hasil Testing

Ukuran kontainer = 7x3x3		63			
No	Total Box	Total Volume Box	Waste(%)	Perhitungan Waktu	Hasil Pengurangan Volume
1	9	62	1.59	0.1	1
2	6	54	14.29	0.1	9
3	8	63	0.00	0.1	0
4	3	13	79.37	0.1	50
5	4	40	36.51	0.1	23
6	7	58	7.94	0.1	5
7	5	42	33.33	0.1	21
8	9	58	7.94	0.1	5
9	5	63	0.00	0.1	0
10	5	53	15.87	0.1	10
11	4	52	17.46	0.1	11
12	6	62	1.59	0.1	1
13	5	34	46.03	0.1	29
14	6	53	15.87	0.1	10
15	4	56	11.11	0.1	7
16	7	51	19.05	0.1	12
17	6	66	-4.76	0.1	-3
18	10	70	-11.11	0.1	-7
19	4	46	26.98	0.1	17
20	4	60	4.76	0.1	3

Pada gambar diatas terlihat hasil *waste(%)* untuk mendapatkan hasil perhitungan sisa ruang kosong berupa % dapat dihitung dengan cara $(\text{volume kontainer} - \text{total volume box}) / \text{volume kontainer} * 100$. Dari 100 kali testing diketahui bahwa rata-rata *waste(%)* adalah 22.30 sedangkan 77.70 sudah terisi penuh dalam kontainer dan rata-rata dari waktu yang dibutuhkan adalah 0.1 detik dengan menggunakan *processor i7-4510U* dan memiliki 12.35 GFlops. Jika dibandingkan dengan penelitian Novita rata-rata *waste(%)* adalah 38,71 dan waktu yang dibutuhkan adalah 16.32 detik dengan menggunakan *processor i3-3217U* yang memiliki 7.83GFlops. Pada penelitian M. Gurbuz diketahui bahwa rata-rata *waste(%)* adalah 30.11 dan

waktu yang dibutuhkan adalah 0.16 detik dan menggunakan *processor* Intel Core Duo dengan 4.52GFlops. Dengan menggunakan ratio dari FLOPS, penulis dapat melakukan perkiraan waktu yang dimiliki oleh Novita sebesar 10.3 dan milik M. Gurbuz sebesar 0.06 detik dengan menggunakan *processor* i7-4510U, sehingga disimpulkan bahwa penelitian ini lebih cepat dalam melakukan pencarian ruang kosong dibandingkan dengan penelitian lainnya.

4.5 Hasil Diskusi

Hasil yang didapat dari penelitian ini adalah *Planner* dapat memaksimalkan ruang yang ada dikontainer sehingga semakin banyak barang yang dapat ditampung di satu kontainer.

Tabel 4.4 Hasil diskusi

Sebelum	Sesudah
<i>Planner</i> melakukan perkiraan ruang kosong hanya dengan menggunakan gambar yang tidak sesuai dengan ukuran asli kontainer sehingga terdapat banyak ruang kosong dalam satu kontainer.	<i>Planner</i> dapat mensimulasikan barang-barang sesuai dengan ukuran yang ditentukan dan kontainer yang telah ditentukan.
Melakukan secara manual sehingga rentan terjadinya kesalahan.	Melakukan secara otomatis dengan tampilan yang telah dibuat dan meminimalisir kesalahan karena komputer lebih detail.

Dapat mengganti posisi barang dari horizontal ke vertical begitu juga dengan kontainer	Sistem tidak dapat mengganti posisi barang dan kontainer
--	--

4.6 *Maintenance*

Pemeliharaan atau *maintenance* sistem secara berkala sangat perlu dilakukan agar keseluruhan bisnis proses dapat berjalan sesuai yang diinginkan, terutama dalam hal melakukan tampilan gambar simulasi.

Langkah-langkah *maintenance* terhadap sistem ini adalah sebagai berikut:

- Melakukan pengecekan secara berkala terhadap sistem untuk meminimalisir kesalahan seperti tampilan simulasi yang barang a dan b yang menyatu.
- Menambah masa durasi hosting agar sistem tetap berjalan dan dapat diakses oleh *Head Planner*.

UMMN