

**ANALISIS KARAKTERISTIK METODE SORTING DALAM
IMPLEMENTASI KONSEP *EXTREME POINT-BASED*
HEURISTIC PADA PROSES MUAT**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**FADHILLAH MEIZI RAMADHAN
0606077106**



**UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2010**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar**

Nama : Fadhillah Meizi Ramadhan

NPM : 0606077106

Tanda tangan :

Tanggal : Juni 2010



PERSETUJUAN

Skripsi dengan judul:

“Analisis Karakteristik Metode Sorting Dalam Implementasi Konsep Extreme Point-Based Heuristic Pada Proses Muat”

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Industri Departemen Teknik Industri Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian skripsi

Depok, 22 Juni 2010

Pembimbing Skripsi

(Armand Omar Moeis ST M.Sc)

NIP 04 06 05 0072

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Fadhillah Meizi Ramadhan
NPM : 0606077106
Program Studi : Teknik Industri
Judul Skripsi : Analisis Karakteristik Metode Sorting Dalam Implementasi Konsep *Extreme Point-Based Heuristic* Pada Proses Muat

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Armand Omar Moeis, ST., M.Sc. ()

Penguji : Arian Dhini, ST.,MT. ()

Penguji : Ir. Amar Rachman, MEIM. ()

Penguji : Ir. Yadrifil, MSc. ()

Ditetapkan di : Depok

Tanggal :

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Fadhillah Meizi Ramadhan

NPM : 0606077106

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

Analisis Karakteristik Metode Sorting Dalam Implementasi Konsep *Extreme Point-Based Heuristic* Pada Proses Muat

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2010

Yang Menyatakan

(Fadhillah Meizi Ramadhan)

UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT yang Mahakuasa atas segala ridho, karunia, dan rahmat-Nya sehingga skripsi ini dapat selesai dengan baik. Penulisan skripsi ini dilakukan dalam rangka melengkapi salah satu persyaratan untuk menyelesaikan Program Pendidikan Sarjana Teknik Industri, Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini, saya menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. Ir. T. Yuri M. Zagloel, MEng.Sc, selaku Ketua Departemen Teknik Industri Universitas Indonesia, yang telah memberikan banyak bimbingan bagi para mahasiswanya.
2. Bapak Armand Omar Moeis ST M.Sc selaku dosen pembimbing skripsi, untuk segala bimbingan, bantuan, arahan, dukungan, serta ide-idenya yang menenangkan hati
3. Bapak Komarudin, ST., M.Eng., yang secara luar biasa memberikan banyak sekali ide, saran, bantuan dan pengarahan hingga tetes darah penghabisan.
4. Seluruh dosen Departemen Teknik Industri yang telah memberikan ilmu-ilmu yang sangat berharga kepada penulis
5. Bu Har, Mbak Triana, Mbak Willy, Mas Mursyid, Mas Latief, Mas Iwan dan seluruh karyawan Departemen Teknik Industri atas semua bantuannya kepada penulis
6. Mama,(Alm) Papa yang sangat-sangat luar biasa mendidik dan membesarkan anak ganteng (namanya juga turunan) ini menjadi (mudah-mudahan) luar biasa juga. Ka pipit, Bang Ojan, Ka Ani, Ela yang tidak pernah hentinya meramaikan serta mendukung hidup saya (cailaaa), saya panjatkan beribu-ribu doa dan terima kasih (apa coba).
7. Lucy dan Febi, sahabat-sahabat fasilkom yang pantang menyerah tetap berusaha membuat bocah yang dulunya tidak bisa baca tulis ini menjadi bisa sedikit membaca.

8. Disa yang telah banyak membantu baik secara teknis, non-teknis, non-fat, or low sugar. Arigatoouuu!
 9. Yudianto, Fajri, Fathur, Dezto yang telah berhasil menyambung nasib saya dan memberikan dukungan baik financial maupun material (penggaris butterfly, bolpen, dll) dan mohon penulis dihubungi lewat email lamanya (yang udah expired) jika ada hutang-piutang yang masih tertinggal. Hal ini juga berlaku untuk landra dan andre, bocah-bocah kape yang terlihat lebih sibuk dari saya.
 10. Sisca, Steven, Christie, Satria rekan seperjuangan dalam skripsi sejak awal hingga akhir.
 11. Tim Futsal TI dan Teknik yang telah banyak menginspirasi penulis untuk bermain di liga inggris. Penulis belajar bahwa setiap inspirasi yang saya dapatkan memang perlu direvisi.
 12. Lelaki bimbang yang selalu galau tanpa pencerahan berarti selama hidupnya. Mudah-mudahan kegalauan dapat menyebar layaknya kegalauan FDI terhadap LT.
 13. Seluruh teman-teman angkatan 2006 Teknik Industri atas kebersamaan dan keceriaan bersama. Kalian adalah harta yang tak ternilai dalam hidup.
 14. Semua pihak yang terlibat dan telah membantu penulis sehingga skripsi ini dapat terselesaikan dengan baik.
- Akhirnya, penulis berharap agar skripsi ini bisa memberikan inspirasi dan manfaat bagi semua pihak yang membacanya dan bagi pengembangan ilmu pengetahuan.

Depok, 22 Juni 2010

Penulis

ABSTRAK

Nama : Fadhillah Meizi Ramadhan
Program Studi : Teknik Industri
Judul : Analisis Karakteristik Metode Sorting Dalam Implementasi Konsep *Extreme Point-Based Heuristic* Pada Proses Muat

Extreme point-based heuristic (EPBH) merupakan metode heuristik yang memiliki performa lebih baik dibandingkan dengan metode heuristik yang lain dalam menyelesaikan masalah *bin packing* atau pemuatan barang. Performa proses pemuatan dari konsep heuristik juga sangat bergantung pada pemilihan barang yang akan dimuat terlebih dahulu. Umumnya, operator mengurutkan barang yang akan dimuat berdasarkan pemilihan yang kasat mata. Penelitian ini bertujuan untuk melihat karakteristik serta performa metode-metode *sorting* yang dapat dilakukan secara manual oleh operator pada metode EPBH pada kelas-kelas barang tertentu. Hasil yang didapat memperlihatkan metode *sorting* berdasarkan *volume* memiliki performa terbaik.

Kata kunci:

Extreme Point-based Heuristic, Karakteristik Metode *sorting*, *Bin Packing Problem*.

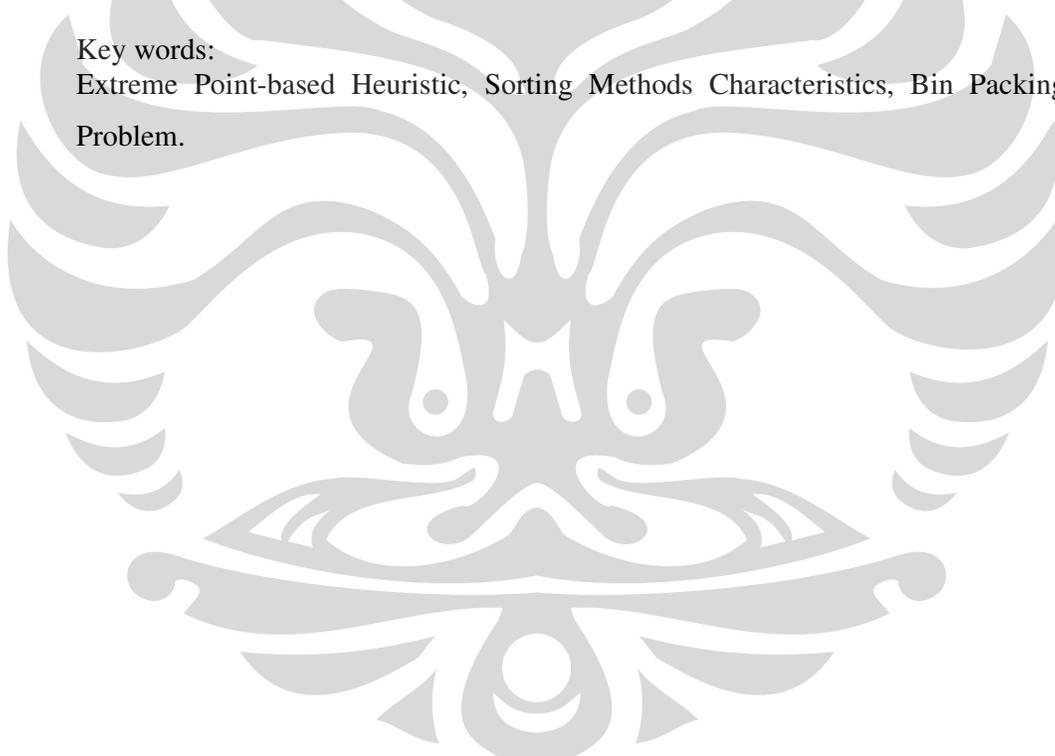
ABSTRACT

Name : Fadhillah Meizi Ramadhan
Study Program: Industrial Engineering
Title : Characteristic Analysis of Sorting Method in Extreme Point-Based Heuristic Implementation in Loading Process

Ekstreme point-based heuristic (EPBH) is a heuristic method which experimentally perform better result in solving bin packing problem compared to another existing heuristic method. Heuristic concept performance also depends on sorting method which chooses the first good to be loaded. The common sorting methods by operator in loading process rely on his plain view. The objective of the research is to observe common sorting methods characteristics in EPBH method in various classes of goods. The result shows that volume based sorting method yielded the best result at any class of goods.

Key words:

Extreme Point-based Heuristic, Sorting Methods Characteristics, Bin Packing Problem.



DAFTAR ISI

Halaman Pernyataan Orisinalitas	ii
Persetujuan	iii
Halaman Pengesahan	iv
Halaman Pernyataan Persetujuan Publikasi	v
Ucapan Terima Kasih	vi
Abstrak	viii
Abstract	ix
Daftar Isi	x
Daftar Tabel	xiii
Daftar Gambar	xiv
Daftar Lampiran	xv
BAB 1 Pendahuluan	1
Latar Belakang Permasalahan	1
1.1. Diagram Keterkaitan Masalah	3
1.2. Perumusan Masalah	4
1.3. Tujuan Penelitian	4
1.4. Batasan Permasalahan	4
1.5. Metodologi Penelitian	4
1.6. Sistematika Penulisan	7
BAB 2 Dasar Teori	9
2.1. Masalah Proses Muat	9
2.2. Pembuatan Barang dalam Kontainer 3 Dimensi	10
2.3. Konsep <i>Extreme Points</i>	12
2.3.1. Prinsip Umum <i>Extreme Points</i>	12
2.3.2. Algoritma Umum <i>Extreme Points</i>	12
2.3.3. Bangun Heuristik <i>Extreme Points</i>	14
2.4. Metode <i>Sorting</i>	15
BAB 3 Pengumpulan dan Pengolahan Data	17
3.1. Pembangkitan Data	17
3.2. Perancangan Perangkat Lunak	19
3.2.1. Pengurutan Barang	21
3.2.2. Uji Kecocokan EP	22
3.2.3. Memperbaharui Daftar EP	24
3.2.4. Pengurutan EP	28
3.2.5. Pengujian Algoritma secara Manual	29
3.2.6. Uji Validitas Program	30
3.2.6.1. Uji Umum	30
3.2.6.2. Uji Proyeksi	32
3.2.6.2.1. Proyeksi Sejajar Sumbu X	32
3.2.6.2.2. Proyeksi Sejajar Sumbu Y	34
3.2.6.2.3. Proyeksi Sejajar Sumbu Z	35
3.3. Pengolahan Data	36
BAB 4 Pembahasan	38

4.1. Pembahasan Hasil Pengolahan Data	38
4.1.1.Pembahasan <i>Class 1</i>	38
4.1.2.Pembahasan <i>Class 2</i>	38
4.1.3.Pembahasan <i>Class 3</i>	41
4.1.4.Pembahasan <i>Class 4</i>	43
4.1.5.Pembahasan <i>Class 5</i>	44
4.1.6.Pembahasan <i>Class 6</i>	46
4.1.7.Pembahasan Hasil <i>Class 7</i>	47
4.1.8.Pembahasan Hasil <i>Class 8</i>	49
4.2. Evaluasi Hasil	50
4.2.1.Evaluasi Keseluruhan	50
4.2.2.Evaluasi Metode Pengurutan	51
4.2.2.1.Peforma Metode <i>No Sort</i>	53
4.2.2.2.Peforma Metode <i>Width</i>	52
4.2.2.3.Peforma Metode <i>Length</i>	52
4.2.2.4.Peforma Metode <i>Height</i>	52
4.2.2.5.Peforma Metode <i>Base Area</i>	53
4.2.2.6.Peforma Metode <i>Volume</i>	53
4.2.2.7.Peforma Metode <i>Base Diagonal</i>	53
4.2.2.8.Peforma Metode <i>Diagonal</i>	54
BAB 5 Kesimpulan	55
Daftar Referensi	56
Lampiran	58

DAFTAR TABEL

Tabel 4.1	Hasil Pengolahan Data Kelas 1	38
Tabel 4.2	Peforma Metode Tiap Kriteria Ukuran Kelas 1.....	39
Tabel 4.3	Hasil Pengolahan Data Kelas 2	40
Tabel 4.4	Peforma Metode Tiap Kriteria Ukuran Kelas 2.....	41
Tabel 4.5	Hasil Pengolahan Data Kelas 3	41
Tabel 4.6	Peforma Metode Tiap Kriteria Ukuran Kelas 3.....	42
Tabel 4.7	Hasil Pengolahan Data Kelas 4	43
Tabel 4.8	Peforma Metode Tiap Kriteria Ukuran Kelas 4.....	44
Tabel 4.9	Hasil Pengolahan Data Kelas 5	44
Tabel 4.10	Peforma Metode Tiap Kriteria Ukuran Kelas 5.....	45
Tabel 4.11	Hasil Pengolahan Data Kelas 6	46
Tabel 4.12	Peforma Metode Tiap Kriteria Ukuran Kelas 6.....	47
Tabel 4.13	Hasil Pengolahan Data Kelas 7	47
Tabel 4.14	Peforma Metode Tiap Kriteria Ukuran Kelas 7.....	48
Tabel 4.15	Hasil Pengolahan Data Kelas 8	49
Tabel 4.16	Peforma Metode Tiap Kriteria Ukuran Kelas 8	50
Tabel 4.17	Peringkat Peforma Metode <i>Sorting</i> pada Tiap Kelas.....	51
Tabel 4.18	Hasil Uji Keseluruhan.....	51



DAFTAR GAMBAR

Gambar 1.1	Diagram Keterkaitan Masalah	3
Gambar 1.2	Diagram Alir Penelitian	5
Gambar 2.1	<i>Corner Point</i> pada 3 dan 2 Dimensi	11
Gambar 2.2	Proyeksi <i>Extreme Points</i>	14
Gambar 3.1	Diagram alir algoritma <i>extreme point based heuristic FFD</i>	20
Gambar 3.2	Diagram Alir Uji Kecocokan EP	23
Gambar 3.3	Diagram Alir Memperbaharui Daftar EP	25
Gambar 3.4	Contoh Proyeksi Titik pada Bidang	26
Gambar 3.5	Diagram Alir Pembangkitan EP dari Proyeksi	28
Gambar 3.6	Hasil Perhitungan Uji Validitas Manual	29
Gambar 3.7	Hasil Visualisasi Uji Validitas Manual	30
Gambar 3.8	Uji Validitas Program	31
Gambar 3.9	Hasil Uji Validitas Umum	32
Gambar 3.10	Uji Validitas Proyeksi X	32
Gambar 3.11	Hasil Uji Validitas Proyeksi X	33
Gambar 3.12	Visualisasi Hasil Uji Validitas Proyeksi X	33
Gambar 3.13	Uji Validitas Proyeksi Y	34
Gambar 3.14	Hasil Uji Validitas Proyeksi Y	34
Gambar 3.15	Uji Validitas Proyeksi Z	35
Gambar 3.16	Hasil Uji Validitas Proyeksi Z	35
Gambar 3.17	Contoh Keluaran Pengolahan Data	36
Gambar 4.1	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 1	39
Gambar 4.2	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 2	40
Gambar 4.3	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 3	42
Gambar 4.4	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 4	43
Gambar 4.5	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 5	45
Gambar 4.6	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 6	46
Gambar 4.7	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 7	48
Gambar 4.8	Grafik Jumlah Barang vs Jumlah <i>Bin</i> Kelas 8	49

DAFTAR LAMPIRAN

Lampiran. Data Generation..... 58



BAB 1

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Dalam menjalankan bisnis produksi ataupun distribusi logistik, efisiensi dalam kegiatan operasional menjadi penentu kekuatan dari sebuah perusahaan. Dalam praktek sehari-hari, perusahaan tidak hanya meminimumkan biaya-biaya yang umum seperti transportasi, penggudangan, buruh, dan *overhead* akan tetapi juga mengoptimalkan penggunaan alat bantu operasional mereka. *Material handling* merupakan istilah bagi alat angkut yang biasa digunakan dalam kegiatan operasional perusahaan distribusi logistik. Optimasi dari penggunaan alat angkut ini berkorelasi langsung dengan kinerja dari divisi operasional serta efisiensi biaya transportasi untuk mengangkut alat *material handling* tersebut. Masalah utama dalam hal ini adalah ketersediaan dari alat *material handling* yang dibutuhkan untuk setiap kegiatan distribusi barang ke salah satu lokasi tujuan. Ketersediaan disini bisa diartikan sebagai efektifitas penempatan alat *material handling* di tiap lokasinya jika dilihat dari perencanaan operasionalnya. Untuk itu dibutuhkan program yang dapat membantu dalam perhitungan jumlah alat *material handling* yang dibutuhkan pada setiap proses muat pada tiap lokasi sehingga dapat meminimalisir keadaan dimana terjadi kekurangan alat angkut yang berimplikasi terhadap jumlah alat transportasi yang dibutuhkan untuk mengangkut alat *material handling* tersebut.

Problem proses muat seperti ini dikenal di literatur sebagai *Bin Packing Problem* (BPP) atau sering disebut sebagai *Container Loading Problem* (CLP). Mengingat barang-barang (paket) yang dimuat di dalam alat *material handling* (dalam hal ini *bin*) bisa sangat bervariasi baik dalam ukuran maupun bentuknya, diasumsikan bahwa bentuk dari paket yang didistribusikan berbentuk *retangular box* (balok atau kubus) untuk memperkecil ruang lingkup penelitian. Optimasi yang akan dilakukan diharapkan menghasilkan suatu solusi susunan dari paket yang mempertimbangkan kestabilan paket serta menja agar tidak ada paket yang melayang atau pun tumpang tindih (*overlapping*).

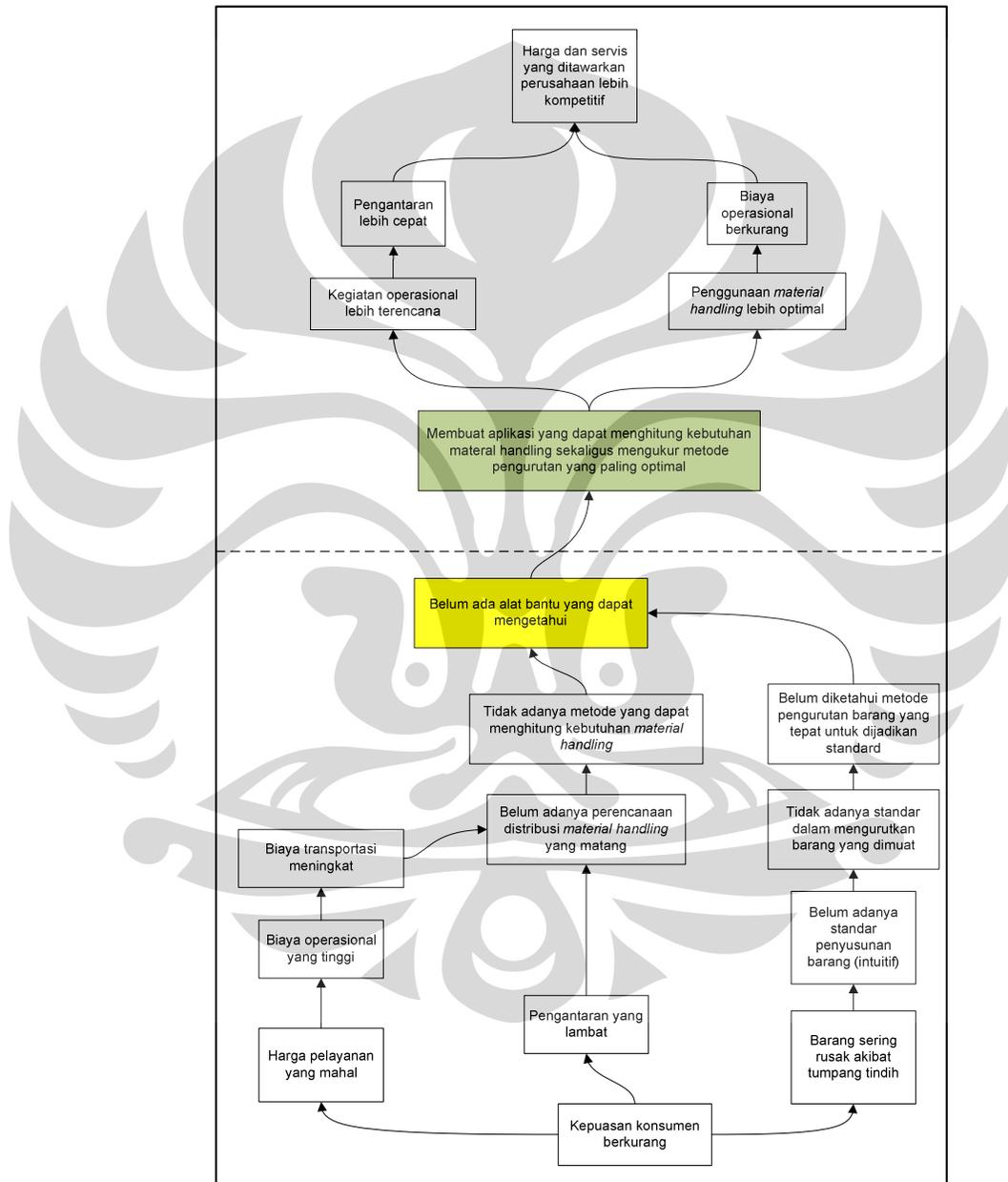
Selama ini pertimbangan dalam proses muat hanya didasarkan kepada perbandingan volume, berat serta paket yang lebih dahulu tiba ke lokasi untuk dimuat/diletakkan ke alat angkut (*cargotainer/pallet*) yang bersifat intuitif atau lebih bergantung kepada pengalaman semata. Untuk itu, dibutuhkan model standar yang meningkatkan utilitas dari pemakaian ruang dari *bin* serta memudahkan dalam perhitungan kebutuhan *bin* dalam tiap proses muat.

Metode heuristik yang tepat untuk menghilangkan sifat intuitif dalam proses muat. Konsep heuristik memiliki aturan yang jelas dalam proses muat sehingga dapat diimplementasikan dalam proses muat sebenarnya. Inti dari model heuristik yang akan diimplementasikan adalah meminimalisir jumlah pemakaian *bin* serta menghindari kerusakan paket akibat proses muat yang salah. Dalam penelitian ini diambil konsep *extreme point-based heuristic* (EPBH) dimana, selain merupakan konsep heuristik, EPBH juga memiliki keunggulan dalam efisiensi penggunaan *bin*.

Banyaknya *bin* yang terpakai dalam suatu proses muat tidak hanya bergantung pada metode peletakkannya tetapi juga tergantung dari cara memilih barang yang akan dimuat lebih awal dibanding barang lainnya. Untuk itu, dalam penelitian ini akan dibahas apa metode *sorting* yang sesuai untuk kasus-kasus masalah pemuatan tertentu. Konsep *extreme point-based heuristic* yang juga merupakan pengembangan dari konsep heuristik sebelumnya yaitu *corner point* membawakan 6 metode *sorting* barang yang dianggap memiliki performa yang terbaik yaitu : *volume-height*, *height-volume*, *area-height*, *height-area*, *clustered area-height*, dan *clustered height-area*. Akan tetapi, metode-metode tersebut akan sulit jika dilakukan pada proses pemuatan manual, dimana proses *sorting* dilakukan oleh operator sehingga parameter pembandingnya harus kasat mata. Untuk itu, penelitian ini akan membawakan 8 metode *sorting* yang lebih mudah digunakan pada proses muat manual yaitu pengurutan berdasarkan barang yang lebih dahulu masuk, panjang barang, lebar barang, tinggi barang, luas alas barang, volume barang, diagonal alas barang, serta diagonal luas barang. Dari metode-metode nantinya akan dilihat metode apakah yang paling efektif dalam implementasi konsep EPBH pada kasus-kasus yang ada.

1.2 Diagram Keterkaitan Masalah

Berikut ini merupakan diagram keterkaitan masalah dari penelitian ini. Diagram ini memberikan informasi mengenai latar belakang permasalahan, perumusan masalah, dan tujuan serta solusi yang diharapkan dari penelitian ini.



Gambar 1.1 Diagram Keterkaitan Masalah

1.3 Perumusan Masalah

Masalah umum yang diutarakan pada latar belakang masalah diatas adalah efisiensi penggunaan *bin*, sehingga rumusan masalah yang dapat membawa penelitian ini kepada solusi yang diinginkan adalah: “*Bagaimana menentukan metode sorting yang efektif dalam implementasi konsep EPBH pada kasus-kasus tertentu*”

1.4 Tujuan Penelitian

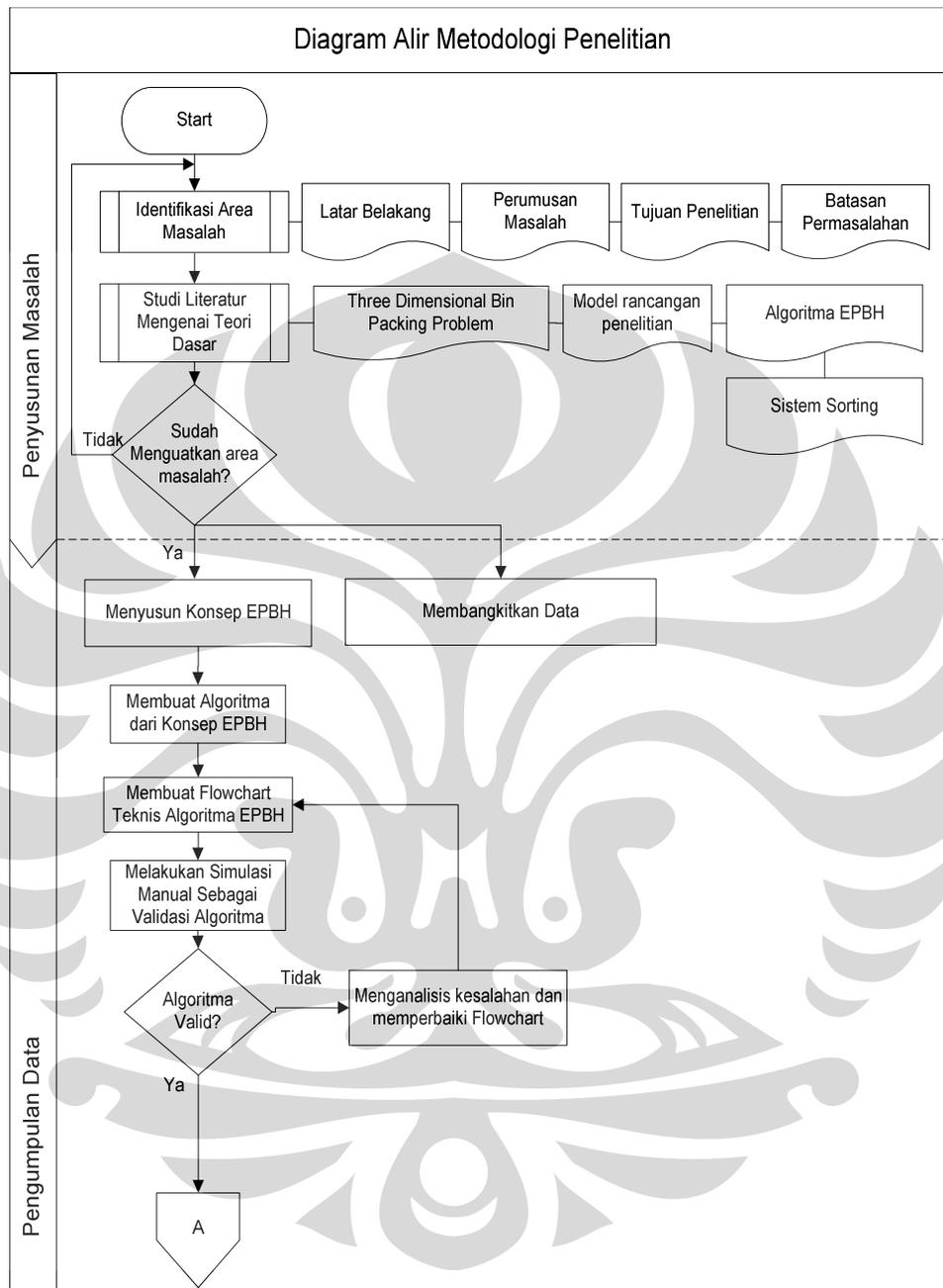
Efektifitas dari metode *extreme point-based heuristic* ini sebenarnya sangat bergantung dari bagaimana mengurutkan barang yang akan dimuat dalam *container* sehingga, agar performa dari bangun heuristik ini baik, dibutuhkan suatu penelitian yang bertujuan untuk mencari metode *sorting* yang tepat dalam usaha pencarian solusi untuk beberapa kombinasi barang yang memiliki kriteria tertentu. Untuk itu, penelitian ini bertujuan untuk melihat metode *sorting* yang terbaik dalam penyelesaian proses muat untuk kelas-kelas serta ukuran jumlah barang tertentu.

1.5 Batasan Permasalahan

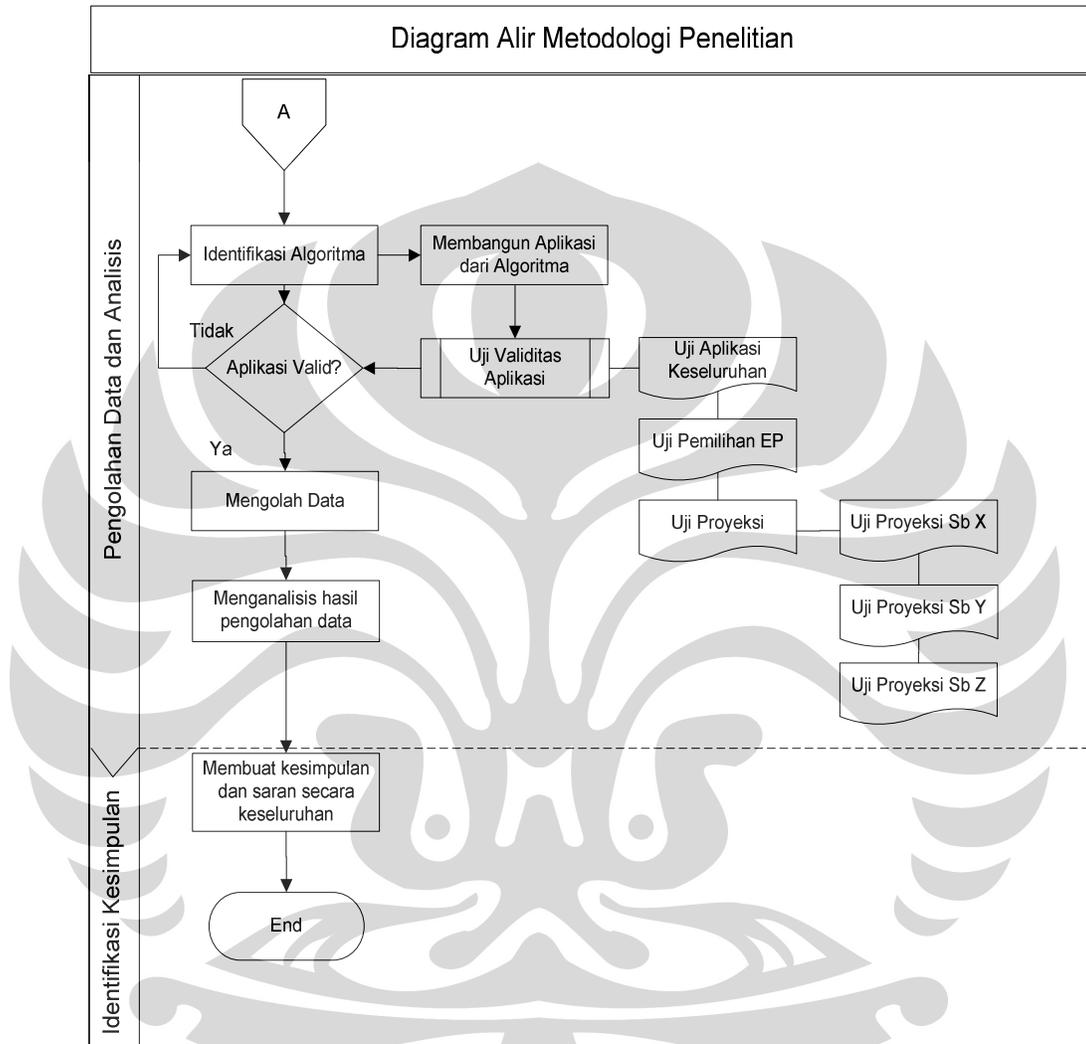
Dalam menentukan lingkup dari penelitian diperlukan batasan-batasan untuk membantu terfokusnya penelitian. Batasan-batasan tersebut adalah: *material handling* yang akan dipakai untuk memasukkan paket atau barang berukuran tetap dan *rectangular*, barang yang di muat kedalam *material handling* juga diasumsikan berbentuk *rectangular* (balok atau kubus), barang yang di muat tidak dapat diputar, serta jumlah *material handling* yang tersedia dianggap tidak terbatas untuk mencegah perluasan masalah yang dapat menyentuh kepada perencanaan distribusi *material handling* tersebut. Karena masalah ini masuk pada area *bin-packing problem* dimana konsep ini lebih berfokus kepada penggunaan ruang dari *bin* yang membuat variabel berat tidak diindahkan sehingga diasumsikan daya tahan kotak dianggap tinggi sehingga dapat menampung beban berat dan dapat ditumpuk-tumpuk tanpa batas.

1.6 Metodologi Penelitian

Metodologi dalam penelitian ini dapat dilihat dalam diagram berikut:



Gambar 1.2 Diagram Alir Metode Penelitian



Gambar 1.2 Diagram Alir Metode Penelitian (Sambungan)

1. Pemilihan Topik Penelitian

Permasalahan perusahaan distribusi logistik seperti yang dipaparkan di latar belakang mendorong untuk ditelitinya model yang tepat untuk mengakomodasi permasalahan perhitungan jumlah kebutuhan *material handling* pada proses muat untuk ruang 3 dimensi.

2. Studi Literatur

Mempelajari teori-teori yang membawa pendekatan berbeda yang berkaitan dengan *three dimensional bin packing problem* dalam literatur dan kemudian mencari pendekatan yang sesuai dengan keadaan yang diinginkan.

3. Pembangkitan data

Menggunakan metode pembangkitan data yang digunakan pada penelitian sebelumnya untuk mendapatkan data yang dapat diolah sesuai dengan tujuan penelitian ini.

4. Perancangan Perangkat Lunak

Menterjemahkan konsep ke algoritma-algoritma teknis yang akan diterjemahkan menjadi bahasa program dan menghasilkan suatu program yang memudahkan proses pengolahan data. Melakukan validasi terhadap algoritma-algoritma tersebut secara manual, serta melakukan validasi terhadap program secara keseluruhan.

5. Pengolahan Data

Dalam tahap ini data yang ada disimulasikan dengan menggunakan aplikasi yang telah diimplementasikan dengan model yang telah dikembangkan.

6. Analisa Hasil

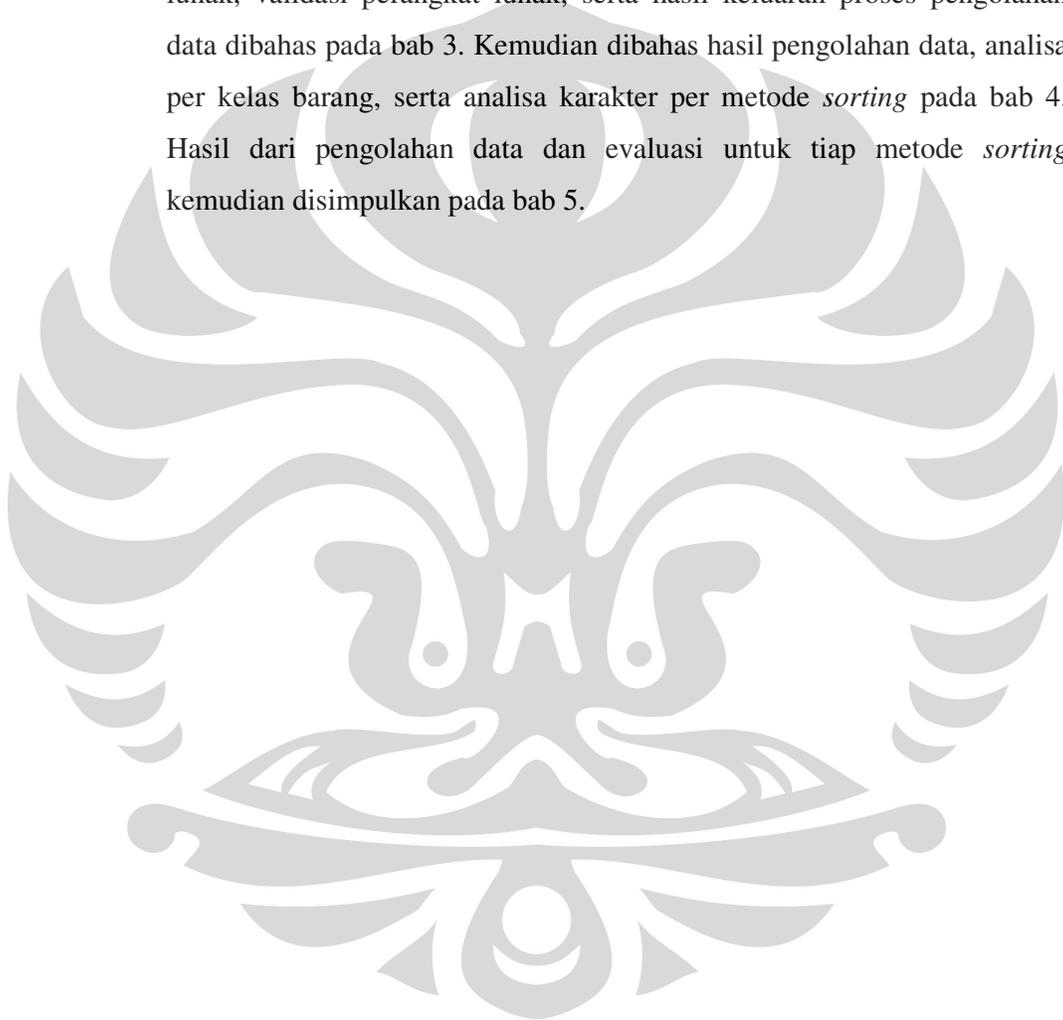
Tahap analisa hasil ini dibagi menjadi 2:

1. Analisa hasil berdasarkan kelas, untuk melihat kecenderungan hasil dari kelas-kelas tertentu
2. Analisa per metode *sorting*, untuk melihat keunggulan-keunggulan metode *sorting* yang ada dan mengetahui metode *sorting* yang tepat dalam penyelesaian masalah.

1.7 SISTEMATIKA PENULISAN

Skripsi ini terdiri dari beberapa bab, yaitu:

Bab 1 berisi latar belakang masalah, tujuan dan manfaat penelitian, permasalahan, ruang lingkup permasalahan, metodologi penelitian, serta sistematika penyusunan skripsi. Lalu bab 2 membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan menjadi acuan dalam pembuatan skripsi ini. Untuk pembahasan bagaimana proses pembangkitan data, algoritma-algoritma yang membangun perangkat lunak, validasi perangkat lunak, serta hasil keluaran proses pengolahan data dibahas pada bab 3. Kemudian dibahas hasil pengolahan data, analisa per kelas barang, serta analisa karakter per metode *sorting* pada bab 4. Hasil dari pengolahan data dan evaluasi untuk tiap metode *sorting* kemudian disimpulkan pada bab 5.



BAB 2 DASAR TEORI

2.1 Masalah Proses Muat

Proses muat secara umum merupakan proses dimana sejumlah barang dimuat kedalam suatu *material handling* ataupun *container*. Proses pemuatan barang ke dalam *container* tersebut merupakan aktifitas dasar dalam industri manufaktur serta distribusi logistic. Masalah proses muat ini sudah lama menjadi perbincangan di kalangan ilmuwan riset operasional. Banyak penciptaan serta pengembangan konsep-konsep yang telah diperkenalkan untuk meningkatkan efisiensi serta efektifitas proses muat. Masalah pemuatan barang-barang ke dalam *container* atau biasa disebut *3D packing problems* ini memiliki beberapa sub-bidang studi tergantung hambatan-hambatan yang dimiliki setiap masalah. Klasifikasi masalah pemuatan barang secara umum adalah sebagai berikut :

1. *Bin-Packing Problems (3D-BPP)*

Masalah dalam *3D-BPP* adalah ketika diberikan sejumlah barang yang berbentuk *rectangular* dengan ukuran bervariasi yang akan dimuat kedalam *bin* berbentuk *rectangular* dengan ukuran tetap. Asumsi dalam proses pemuatannya adalah barang tidak boleh dirotasi sehingga barang dimuat bersisian secara paralel dengan dinding dari kontainernya. Masalah ini diklasifikasikan sebagai NP-Hard seperti yang dijelaskan oleh Martello (2000).

2. *Knapsack Loading Problems (3D-KLP)*

Knapsack Loading Problems bertujuan untuk memaksimalkan keuntungan dimana tiap barang yang dimuat memiliki nilai profit tersendiri.

3. *Container Packing Problems (3D-CPP)*

Barang-barang dimuat kedalam suatu container dimana tujuan akhirnya adalah meminimalisir penggunaan ruang didalam container . Salah satu penemuan yang sudah umum di bidang ini adalah “*Strip Packing*” dimana container dianggap memiliki tinggi yang tidak terbatas sehingga tujuan dari solusinya

adalah memperkecil tinggi tumpukan untuk mengurangi penggunaan ruang (volume container).

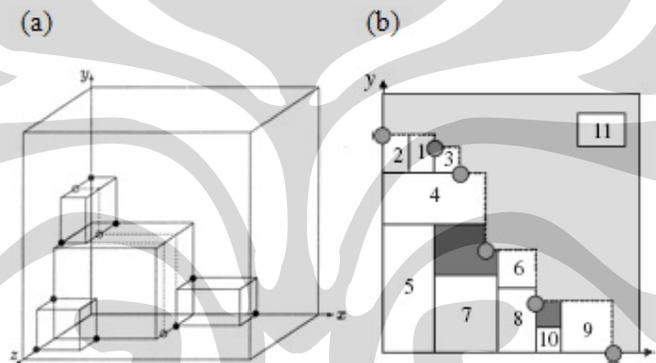
Extreme Point-Based Heuristic (EP-BH) merupakan pendekatan untuk masalah *3D-BPP* dimana bangun heuristik ini merupakan pendekatan yang berbasis pada titik peletakan barang. Dalam menentukan titik peletakan barang, performa dari metode heuristik sangat dipengaruhi oleh pemilihan dari aturan peletakan. Aturan-aturan yang berlaku di EP-BH bertujuan untuk menemukan titik yang paling tepat dalam peletakan barang dapat mengatasi hambatan-hambatan tambahan seperti perbaikan posisi dari barang yang sudah dimuat.

2.2 Pemuatan Barang Dalam Kontainer 3 Dimensi

Penelitian mengenai *Bin-Packing Problem* pada awalnya berfokus pada pengembangan formulasi untuk *2D-Packings*. Sebuah pendekatan yang biasa dipakai dalam masalah *2D-Packing* yaitu *shelf method* dimana pendekatan ini merupakan kombinasi beberapa prosedur yang di desain untuk masalah *mono-dimensional* (1D-BP) (Chung 1982, Berkey dan Wang 1987). Barang-barang di urutkan berdasarkan ukurannya kemudian dimuat kedalam “*shelves*” yang berukuran sama dengan lebar kotaknya . Kemudian agar dapat mengakomodasi masalah *2D-Packing*, “*shelves*” yang ada dimasukkan ke dalam container dengan pendekatan 1D-BP dimana ukuran dari barang yang akan dimuat sama dengan panjang dari *shelf* dan ukuran dari kontainer berdimensi 1 dianggap sama dengan panjang dari kontainer berdimensi 2. Pendekatan yang berbasis *shelf* ini kemudian dikembangkan untuk menyelesaikan masalah *3D-Packing* dengan membuat *2D-shelf* lalu disusun kedalam container tiga-dimensi, kemudian tinggi barang-barang maksimal yang dimuat sama dengan tinggi *shelf*, pendekatan kemudian dikenal sebagai *wall-building* (George dan Robinson 1980, Pisinger 2002).

Martello (2000) mengeluarkan metode yang berbasis *corner point*. *Corner point* ini merupakan titik ujung suatu benda dimana ujung kiri-bawah-belakang sebuah benda dapat diletakkan dalam pemuatan kedalam kontainer. *Corner point* ini adalah metode yang menggunakan perpotongan dari garis-garis yang dimiliki barang-barang yang paling ujung (untuk 2D) dan perpanjangan garis benda ke

benda lain atau dinding kontainer (lihat gambar 2.1 (a)). Martello (2000) juga mendesain sebuah *branch and bound algorithm* untuk melihat apakah sejumlah barang dapat di muat dalam kontainer atau tidak. Den Boef (2005) kemudian menunjukkan bahwa metode algoritma *corner point* yang dibuat Martello (2000) melewati cara-cara muat yang seharusnya ada sehingga melemahkan performa algoritmanya. Martello (2007) menanggapi penelitian Den Boef (2005) dengan meluncurkan versi baru dari algoritma *corner point* yang juga diikuti pengembangan dari algoritma *branch and bound* yang kemudian menjadi metode pencarian solusi pertama untuk *3D-Packing*.



Gambar 2.1 *Corner points* pada 3 dan 2 dimensi

(Sumber: *Extreme point-based heuristic for Three-dimensional bin packing*, 2008)

Martello pada tahun 2000 juga mengeluarkan metode *lower bound* untuk 3D-BP dimana metode ini merupakan metode pengembangan dari Fekete dan Schepers (1997) yang mengembangkan kegunaan metode *dual-feasible functions* untuk masalah 2D-BP dengan bertolak kepada metode yang sama untuk 1D-BP yang diperkenalkan oleh Johnson (1973). Pengembangan *lower bound* juga dilakukan oleh Boschetti (2004) yang kemudian memperkenalkan metode *dual-feasible functions* yang terbaru hingga saat penelitian ini ditulis.

Sebuah algoritma *tabu search* untuk masalah 2D-BP dikemukakan oleh Lodi (1999a). Algoritma ini berisi 2 bangun heuristik sederhana dalam memuat barang-barang ke dalam *bin* dan *tabu search* berguna untuk mengontrol

pergerakan barang diantara *bin-bin* yang ada dimana barang yang berada di *bin* yang paling berpotensi untuk kosong dipindahkan ke *bin* yang lebih penuh. Kemudian Faroe (2003) memperkenalkan metode heuristik *guided local search (GLS)* untuk 3D-BP. GLS menggunakan algoritma dimana perolehan jumlah kebutuhan *bin* didapat dengan prosedur heuristik yang “tamak” dimana tujuannya adalah memperkecil jumlah *bin* dan disaat yang sama GLS mencari posisi muat yang layak. Proses GLS selesai setelah batas waktu yang ditentukan habis atau solusi yang dihasilkan sudah cocok dengan *lower bound* sebelum komputasi dimulai.

Lodi (2002b) memperkenalkan sebuah heuristik yang berbasis *shelf* yang baru untuk 3D-BP yang dinamakan *Height first-Area second (HA)*. Algoritma ini memiliki 2 cara dalam pencapaian solusinya, yang pertama barang-barang dibagi menjadi beberapa kelompok menurut tinggi mereka, kemudian baru dimuat kedalam *bin* dengan menggunakan algoritma *branch and bound* yg dibuat Martello dan Toth (1990) untuk 1D-BP. Yang kedua, mengurutkan barang dikelompokkan berdasarkan luas bidangnya dan kemudian dimuat seperti cara yang pertama. Kedua cara ini kemudian dibandingkan hasilnya dan yang terbaik diambil untuk dijadikan solusi akhir. Konsep HA ini merupakan bangun heuristik yang memiliki hasil terbaik dalam pengujian dalam contoh-contoh masalah.

2.3 Konsep *Extreme Points*

2.3.1 Prinsip Umum *Extreme point*

Konsep *extreme points* (EPs) adalah konsep yang diperkenalkan Crainic (2008) yang juga merupakan pengembangan dari konsep *corner points*. Konsep heuristik ini adalah konsep yang memuat benda-benda berbentuk *rectangular* ke dalam *bin dan* bertujuan untuk meminimumkan jumlah *bin*, memaksimalkan utilisasi ruang, serta efisiensi dalam usaha komputasinya. Efisiensi dari usaha komputasi didapat dari aturan-aturan dari konsep EPs yang dibuat untuk efisiensi proses peletakan serta menghindari peristiwa perbaikan posisi dari barang yang sudah dimuat untuk pelebaran ruang lingkup pencarian solusi.

2.3.2 Algoritma Umum *Extreme point*

Konsep EPs adalah ketika dimana sebuah barang k dengan ukuran w_k , d_k , h_k dimuat kedalam sebuah *bin* dan ujung kiri bawah belakangnya diletakkan diposisi (x_k, y_k, z_k) kemudian menghasilkan beberapa titik kandidat peletakkan, EPs, untuk kemudian dipilih sebagai kandidat penempatan barang yang akan dimuat berikutnya. EPs yang dihasilkan adalah hasil proyeksi titik-titik yang berkoordinat (x_k+w_k, y_k, z_k) , (x_k, y_k+d_k, z_k) , dan (x_k, y_k, z_k+h_k) ke sumbu-sumbu orthogonal dari container (lihat gambar 2.2).

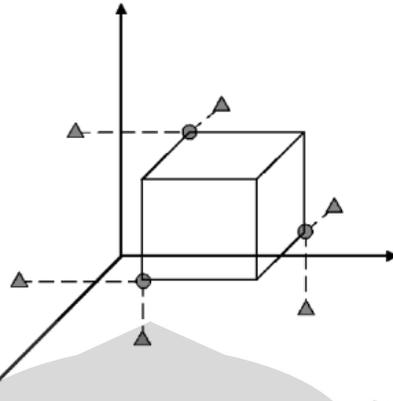
Peristiwa proyeksi diatas terangkum dalam algoritma sebagai berikut:

- Jika kontainer dalam kondisi kosong, maka benda diletakkan pada titik $(0,0,0)$ yang kemudian langsung menghasilkan 3 titik kandidat baru di koordinat $(w_k, 0, 0)$, $(0, d_k, 0, 0)$, dan $(0, 0, z_k)$.
- Jika container tidak dalam keadaan kosong, kemudian barang diletakkan di titik (x_k, y_k, z_k) , maka titik peletakan baru dihasilkan dari melalui proyeksi seperti berikut:
 1. Titik (x_k+w_k, y_k, z_k) diproyeksikan ke sumbu Y dan Z dari *bin*,
 2. Titik (x_k, y_k+d_k, z_k) diproyeksikan ke sumbu X dan Z *bin*, serta
 3. Titik (x_k, y_k, z_k+h_k) diproyeksikan ke sumbu X dan Y *bin*.

Titik tersebut di proyeksikan ke semua barang yang terletak antara item k dengan dinding dari *bin* sesuai dengan arah sumbu proyeksinya.

- Jika ada lebih dari satu barang yang dapat diproyeksikan dari titik proyeksi diatas maka dipilih proyeksi terhadap barang yang terdekat sebagai EP baru.

Proses proyeksi ini terus berlangsung selama proses muat berlangsung sehingga semakin banyak barang yang dimuat semakin banyak EP yang dihasilkan sehingga EP yang ada akan diperbaharui dengan menambah EP yang baru dihasilkan serta menghapus EP yang sudah terpakai. EP yang dipakai dalam penempatan benda adalah EP dimana ketika barang k diletakkan tidak akan beririsan dengan barang-barang sebelumnya.



Gambar 2.2 Proyeksi *extreme points*

(Sumber: *Extreme point-based heuristic for Three-dimensional bin packing*, 2008)

2.3.3 Bangun Heuristik *Extreme points*

Prosedur heuristik yang dikembangkan dalam konsep EPs ini ada 2, yaitu *first fit decreasing* (FFD) dan *best fit decreasing* (BFD). Prosedur FFD dan BFD merupakan bangun heuristik yang biasa digunakan untuk masalah 1D-BP. Kedua prosedur ini berbeda dari bagaimana barang dimuat. Prosedur FFD memuat barang-barang dari satu *bin* ke *bin* yang lain dengan melihat dimana barang tersebut dapat didimuat. Sedangkan BFD lebih melihat dimana *bin* yang paling baik (memiliki *merit function* terbaik) dalam memuat barang yang akan masuk. Pembahasan lebih lanjut untuk kedua prosedur tersebut dijelaskan sebagai berikut:

I. *First fit decreasing*

Algoritma dari *extreme point first fit decreasing* (EP-FFD) awalnya menguji apakah dimensi dari barang yang dimasukkan dapat diakomodasi atau tidak oleh suatu *bin*, jika barang tidak dapat diletakkan pada *bin* tersebut, maka barang akan dimasukkan pada *bin* berikutnya. Untuk menentukan apakah barang tersebut dapat diakomodasi oleh suatu *bin* atau tidak adalah dengan melihat EPnya. Syarat EP yang dapat dipakai adalah ketika ujung kiri bawah belakang dari barang diletakkan di posisi tersebut dan

barang itu tidak beririsan dengan benda-benda lain yang sebelumnya sudah dimuat.

II. *Best fit decreasing*

Prosedur untuk BFD hampir sama dengan FFD, algoritma awal dari BFD juga menguji kesesuaian dari EP apakah cocok untuk dijadikan titik peletakan barang berikutnya atau tidak. EP yang sesuai menurut prosedur BFD adalah EP yang memiliki *merit function* terbaik dari *bin-bin* yang ada. Pada prosedur ini, untuk tiap EP yang dapat mengakomodasi barang yang akan dimuat, tidak beririsan dengan benda lain, juga langsung dihitung *merit function*nya. EP yang memiliki nilai *merit function* terbaik yang dipilih sebagai titik peletakan berikutnya.

2.4 Metode Sorting

Efisiensi yang dihasilkan dari *extreme point-based heuristic* dipengaruhi oleh metode *sorting* yang digunakan. Penelitian Crainic (2008) menyebutkan bahwa ada 6 metode *sorting* yang memiliki hasil yang paling memuaskan, yaitu:

- *Volume-Height*: Barang-barang yang akan dimuat diurutkan berdasarkan volume ($w_i \times d_i \times h_i$) yang terbesar, kemudian jika ada barang-barang yang memiliki nilai ruang sama maka akan diurutkan berdasarkan tingginya (h_i) yang terbesar.
- *Height-Volume*: Barang-barang yang akan dimuat diurutkan berdasarkan tinggi h_i yang terbesar, kemudian jika ada barang-barang yang memiliki tinggi sama maka akan diurutkan berdasarkan volumenya ($w_i \times d_i \times h_i$) yang terbesar.
- *Area-Height*: Barang-barang yang akan dimuat diurutkan berdasarkan luas bidang ($w_i \times d_i$) yang terbesar, kemudian jika ada barang-barang yang memiliki luas bidang sama maka akan diurutkan berdasarkan tinggi (h_i) yang terbesar.

- *Height-Area*: Barang-barang yang akan dimuat diurutkan berdasarkan tinggi (h_i) yang terbesar, kemudian jika ada barang-barang yang memiliki tinggi sama maka akan di urutkan berdasarkan luas bidang ($w_i \times d_i$) yang terbesar
- *Clustered Area-Height*: Karena jarang ada 2 barang yang memiliki luas bidang yang sama, maka aturan sorting yang kedua (*Height* untuk kriteria *Area-Height*) jarang digunakan. Untuk membuat aturan *sorting* ini lebih efektif, maka dibuat interval luas bidang yang membentuk kelas-kelas untuk kelompok barang yang luas areanya masuk kedalam kelas tersebut. Kemudian dari kelas yang ada barang-barang diurutkan berdasarkan tinggi (h_i) yang terbesar. Rumus untuk menentukan interval tiap kelas dapat dilihat sebagai berikut:

$$A_{j,\delta} = \left[\frac{(j-1) \times WD}{100} \delta, \frac{j \times WD}{100} \delta \right] \quad (2.1)$$

dengan $W \times D$ adalah luas *bin* dan $\delta \in [1, 100]$.

- *Clustered Height-Area*: Aturan *sorting* ini merupakan kebalikan dari kriteria *sorting* sebelumnya. Kelas-kelas dibuat berdasarkan interval dari tingginya, kemudian barang-barang yang berada pada kelas yang sama diurutkan berdasarkan luas bidang ($w_i \times d_i$) yang terbesar. Rumus untuk menentukan interval tiap kelas pada kriteria *sorting* ini dapat dilihat sebagai berikut:

$$h_{j,\delta} = \left[\frac{(j-1) \times H}{100} \delta, \frac{j \times H}{100} \delta \right] \quad (2.2)$$

dengan $W \times D$ adalah luas *bin* dan $\delta \in [1, 100]$.

BAB 3 PENGUMPULAN DAN PENGOLAHAN DATA

3.1 Pembangkitan Data

Dalam penelitian ini, data-data yang akan diolah merupakan data-data yang dibangkitkan dengan metode *data generation* yang dibuat oleh Crainic (2008) yang dikembangkan dari contoh-contoh masalah yang dibuat oleh Martello (2000). Dalam pembangkitan data ini, diperoleh 8 kelas data dengan tipe data yang berbeda-beda. Untuk kelas 1 sampai 4 ukuran *bin* $W = D = H = 100$, dan terdapat 5 tipe variasi barang yang dipakai pada kelas 1 sampai 5:

- Type 1: w_j diacak secara uniform $\left[1, \frac{1}{2}W\right]$, h_j diacak secara uniform $\left[\frac{2}{3}H, H\right]$, d_j diacak secara uniform $\left[\frac{2}{3}D, D\right]$;
- Type 2: w_j diacak secara uniform $\left[\frac{2}{3}W, W\right]$, h_j diacak secara uniform $\left[1, \frac{1}{2}H\right]$, d_j diacak secara uniform $\left[\frac{2}{3}D, D\right]$;
- Type 3: w_j diacak secara uniform $\left[\frac{2}{3}W, W\right]$, h_j diacak secara uniform $\left[\frac{2}{3}H, H\right]$, d_j diacak secara uniform $\left[1, \frac{1}{2}D\right]$;
- Type 4: w_j diacak secara uniform $\left[\frac{1}{2}W, W\right]$, h_j diacak secara uniform $\left[\frac{1}{2}H, H\right]$, d_j diacak secara uniform $\left[\frac{1}{2}D, D\right]$;
- Type 5: w_j diacak secara uniform $\left[1, \frac{1}{2}W\right]$, h_j diacak secara uniform $\left[1, \frac{1}{2}H\right]$, d_j diacak secara uniform $\left[1, \frac{1}{2}D\right]$;

Untuk tiap kelas, kombinasi barang-barangnya adalah sebagai berikut:

- *Class 1*: Probabilitas keluar barang dari data tipe 1 sebesar 60%, tipe 2, 3, 4, 5 masing-masing 10%;
- *Class 2*: Probabilitas keluar barang dari data tipe 2 sebesar 60%, tipe 1, 3, 4, 5 masing-masing 10%;

- *Class 3*: Probabilitas keluar barang dari data tipe 3 sebesar 60%, tipe 1, 2, 4, 5 masing-masing 10%;
- *Class 4*: Probabilitas keluar barang dari data tipe 4 sebesar 60%, tipe 1, 2, 3, 5 masing-masing 10%;
- *Class 5*: Probabilitas keluar barang dari data tipe 5 sebesar 60%, tipe 1, 2, 3, 4 masing-masing 10%;
- *Class 6*: w_j, h_j , dan d_j diacak secara uniform $[1,10]$ dan $W = D = H = 10$;
- *Class 7*: w_j, h_j , dan d_j diacak secara uniform $[1,35]$ dan $W = D = H = 40$;
- *Class 8*: w_j, h_j , dan d_j diacak secara uniform $[1,100]$ dan $W = D = H = 100$;

Tiap-tiap kelas ini memiliki ciri-ciri variasi tersendiri dimana:

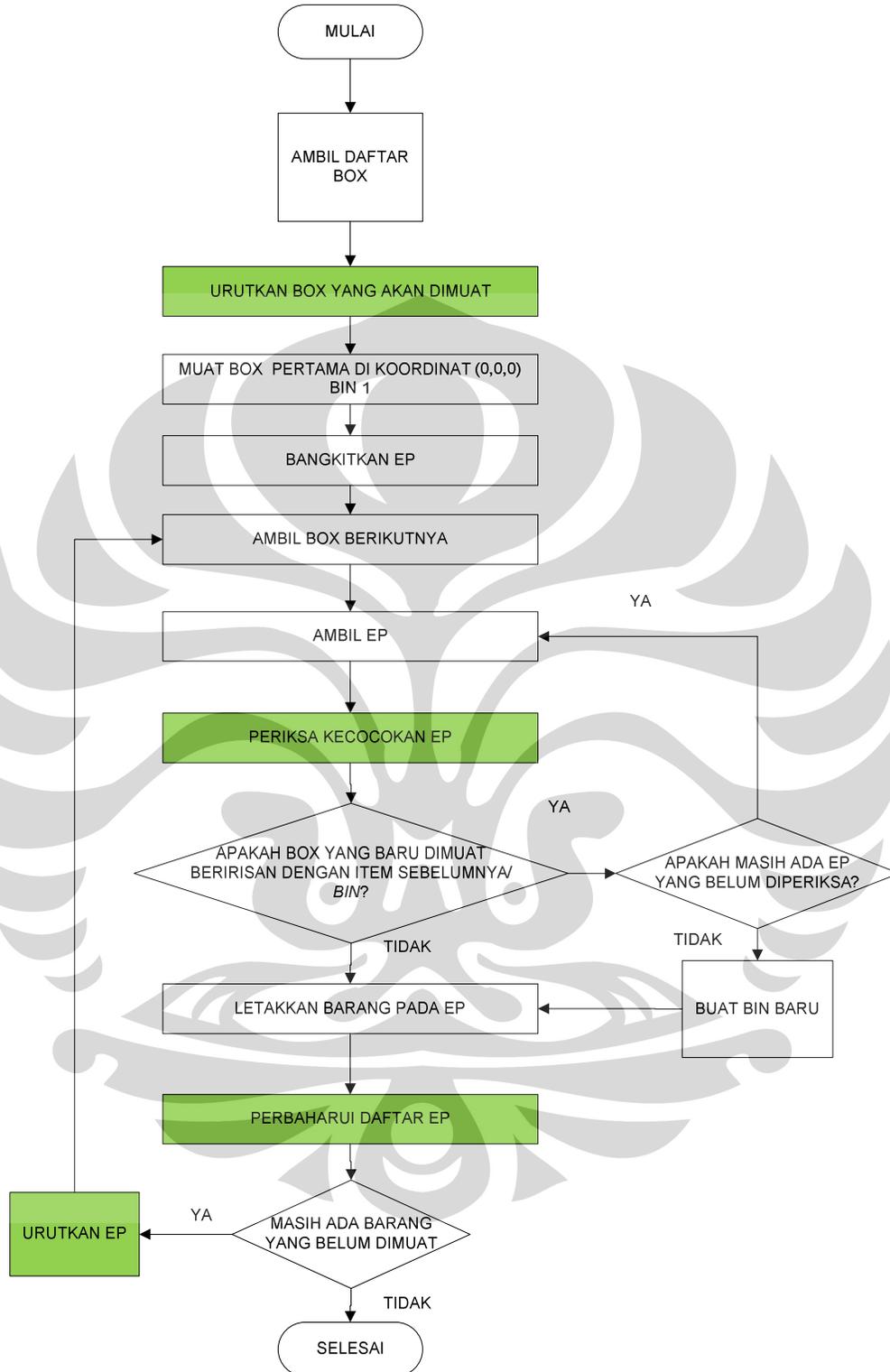
- *Class 1*: Mayoritas barangnya memiliki luas sisi bidang yang memanjang sejajar sumbu Y yang besar (mendekati ukuran *bin*)
- *Class 2*: Mayoritas barangnya memiliki luas alas bidang yang besar (mendekati ukuran *bin*)
- *Class 3*: Mayoritas barangnya memiliki luas sisi bidang yang memanjang sejajar sumbu X yang besar (mendekati ukuran *bin*)
- *Class 4*: Mayoritas barangnya berukuran sedang
- *Class 5*: Mayoritas barang berukuran kecil
- *Class 6*: Ukuran box besar relative terhadap ukuran *bin*
- *Class 7*: Ukuran box relative sedang terhadap ukuran *bin*
- *Class 8*: Ukuran Box bervariasi dari yang sangat kecil hingga besar (mendekati ukuran *bin*)

Dari kelas-kelas barang diatas, dibangkitkan 200 data yang akan diproses pada pengolahan data. Pembangkitan data dilakukan dengan menggunakan Microsoft Excel 2007. Data yang sudah dibangkitkan dapat dilihat pada bagian Lampiran.

3.2 Perancangan Perangkat Lunak

Konsep *extreme point-based heuristic first-fit decreasing* (EPBH FFD) merupakan konsep yang termasuk paling baru dalam penyelesaian masalah *bin-packing* sehingga perangkat lunak yang mengimplementasikan algoritma heuristik ini juga sulit ditemukan. Untuk itu, perlu adanya model teknis yang dapat membantu dalam perancangan perangkat lunak untuk konsep ini.

Berdasarkan penjelasan mengenai konsep umum dari metoda EPBH, maka algoritma dari EPBH dapat dilihat dari diagram alir algoritma EPBH (gambar 3.1). Diagram alir pada gambar 3.1 menjelaskan bahwa langkah pertama dari algoritma ini adalah mendapatkan daftar *input* barang yang akan dimuat. Setelah *input* didapatkan kemudian *box* (kotak barang) diurutkan sesuai metode *sorting* yang akan diimplemetasikan. Kemudian ambil barang urutan teratas yang akan dimasukkan ke dalam *bin*, lalu periksa apakah beririsan dengan *bin*-nya atau tidak. Jika barang tidak beririsan, maka dimasukkan ke dalam *bin* pada koordinat EP awal (0,0,0). Barang tersebut kemudian menghasilkan *extreme point-extreme point* (EP) baru sebagai titik kandidat peletakkan berikutnya. EP-EP yang dihasilkan dari peletakkan pertama kemudian diurutkan berdasarkan nilai z,y,x terkecil. Kemudian EP yang teratas akan dicoba kelayakannya untuk barang yang selanjutnya. Uji kelayakan dari EP ini akan dijabarkan pada penjelasan berikutnya. Jika EP yang teratas tidak layak, maka algoritma akan melanjutkan ke EP selanjutnya dan melakukan uji kelayakan dari EP tersebut. Apabila ditemukan EP yang layak, maka algoritma akan meletakkan titik pojok-kiri-bawah-belakang dari barang yang akan dimuat ke EP tersebut dan langsung membuat EP baru akibat masuknya barang baru. Akan tetapi jika sama sekali tidak ditemukan EP yang layak untuk barang yang akan dimuat, maka algoritma otomatis menghasilkan *bin* baru untuk memuat barang ini, dan kemudian memulai proses untuk barang baru berikutnya untuk dimuat ke *bin* yang lama.



Gambar 3.1 Diagram alir algoritma *extreme point based heuristic FFD*

Dalam diagram alir pada gambar 3.1 juga terdapat 4 proses penting (berwarna hijau) dalam implementasi algoritma secara teknis, yaitu:

- Pengurutan Box
- Uji Kecocokan EP
- Memperbaharui daftar EP
- Pengurutan EP

3.2.1 Pengurutan Barang

Sesuai dengan tujuannya, penelitian ini ingin melihat karakter dari tiap metode pengurutan (*sorting*) dari barang-barang yang akan dimuat untuk tiap kelas barang-barang. Seperti yang dibahas pada bab 2 dimana Crainic (2008) memperlihatkan mengajukan 6 sistem *sorting* yang menunjukkan hasil terbaik dalam eksperimennya. Akan tetapi metode *sorting* jarang diaplikasikan pada proses muat manual. Untuk itu, peneliti akan melihat pengaruh dari kelas-kelas barang terhadap 8 metode *sorting* yang biasa dipakai dalam proses muat manual. Delapan proses pengurutan yang akan dibahas adalah:

1. *No Sort* (Tanpa Pengurutan)

Barang-barang dimasukkan sesuai dengan dengan list barang awal. Sehingga pengurutan mengikuti urutan yang sudah terbentuk dari daftar barang.

2. *Width* (Lebar)

Barang-barang dari daftar awal kemudian diurutkan berdasarkan nilai lebar yang terbesar. Lebar disini adalah sisi benda yang memanjang kearah sumbu X dari *bin*.

3. *Length* (Panjang)

Barang-barang dari daftar awal kemudian diurutkan berdasarkan nilai panjang yang terbesar. Panjang disini adalah sisi benda yang memanjang kearah sumbu Y dari *bin*.

4. *Height* (tinggi)

Barang-barang dari daftar awal kemudian diurutkan berdasarkan nilai tinggi yang terbesar. Tinggi disini adalah sisi benda yang memanjang kearah sumbu Z dari *bin*.

5. Base Area (Luas Bidang)

Metode pengurutan ini mengurutkan barang-barang yang akan masuk berdasarkan luas bidang ($w \times d$) nya. Semua barang yang masuk dihitung luasnya terlebih dahulu kemudian diurutkan berdasarkan luas yang terbesar ke terkecil.

6. Volume

Metode pengurutan ini mengurutkan barang-barang yang akan masuk berdasarkan volume ($w \times d \times h$) nya. Semua barang yang masuk dihitung volumenya terlebih dahulu kemudian diurutkan berdasarkan volume yang terbesar ke terkecil.

7. Base Diagonal (Diagonal Alas)

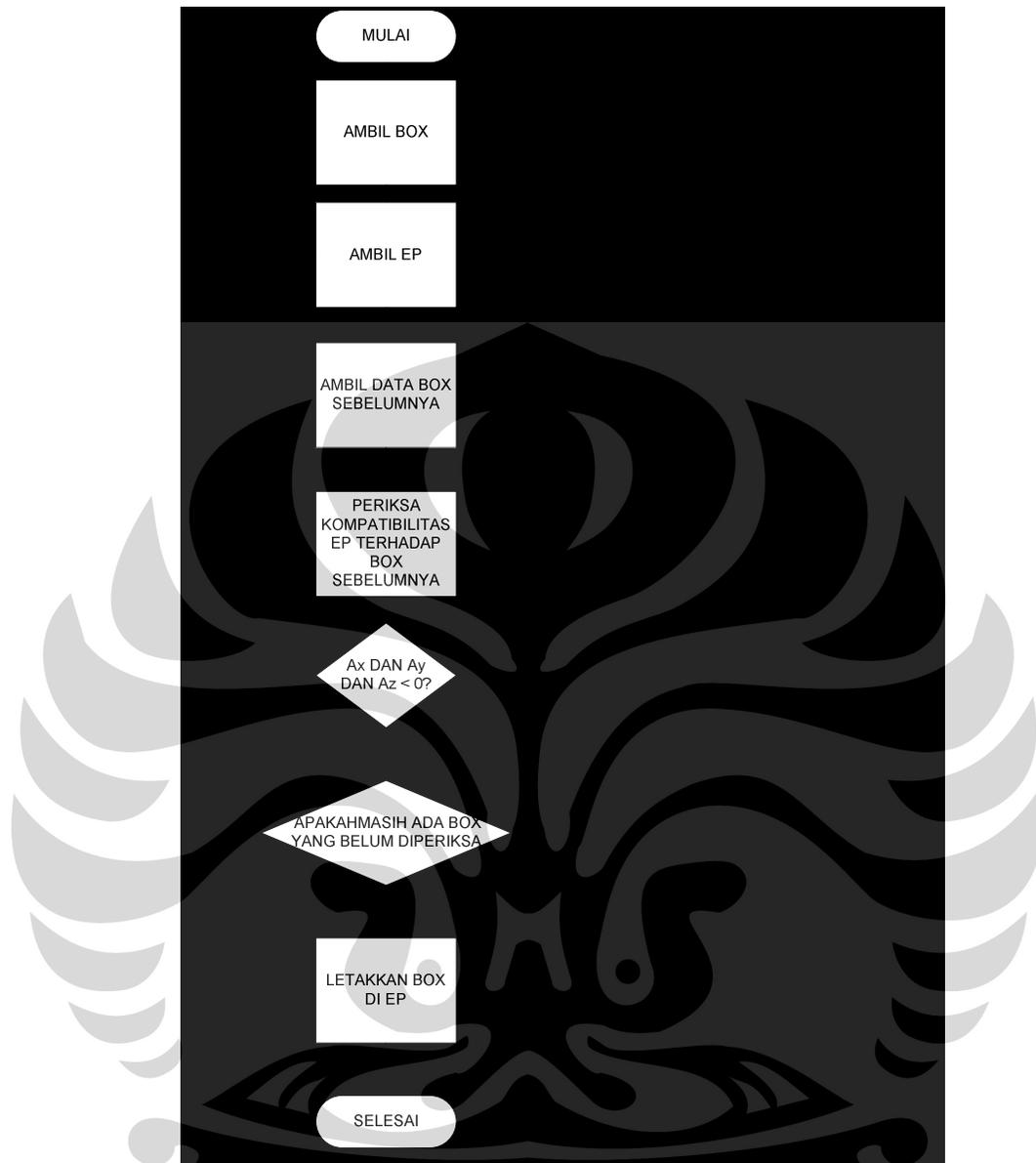
Metode pengurutan ini mengurutkan barang-barang yang akan masuk berdasarkan diagonal alas ($\sqrt{w^2 + d^2}$) nya. Semua barang yang masuk dihitung diagonal alas nya terlebih dahulu kemudian diurutkan berdasarkan diagonal alas yang terbesar ke terkecil.

8. Diagonal

Metode pengurutan ini mengurutkan barang-barang yang akan masuk berdasarkan diagonal ruang ($\sqrt{w^2 + d^2 + h^2}$) nya. Semua barang yang masuk dihitung diagonal ruang nya terlebih dahulu kemudian diurutkan berdasarkan diagonal ruang yang terbesar ke terkecil.

3.2.2 Uji Kecocokan EP

Sebuah titik ekstrim dianggap cocok jika ketika EP tersebut dipakai pada suatu barang, maka barang tersebut tidak akan beririsan dengan barang-barang sebelumnya ataupun *bin* tempat barang tersebut akan dimuat. Proses uji kecocokan EP digambarkan dalam diagram alir uji kecocokan EP Gambar 3.2. Pada diagram alir uji kecocokan EP (Gambar 3.2) dimana (x_i, y_i, z_i) merupakan EP yang dipakai oleh benda i , dijelaskan langkah-langkah dalam



Gambar 3.2 Diagram Alir Uji Kecocokan EP

menguji apakah sebuah EP cocok untuk dipakai oleh barang yang akan dimuat berikutnya atau tidak. Langkah awalnya adalah mengambil data dimensi (panjang, lebar, tinggi) barang berikutnya. Kemudian diambil EP urutan teratas dari daftar EP, lalu EP tersebut digunakan untuk barang berikutnya. Kemudian EP dan barang baru tersebut di uji apakah beririsan dengan EP dan benda sebelumnya atau tidak. Proses ini menggunakan fungsi A_x , A_y , dan A_z untuk melihat apakah suatu benda beririsan dengan benda lain pada sumbu tertentu (A_x untuk sumbu x ,

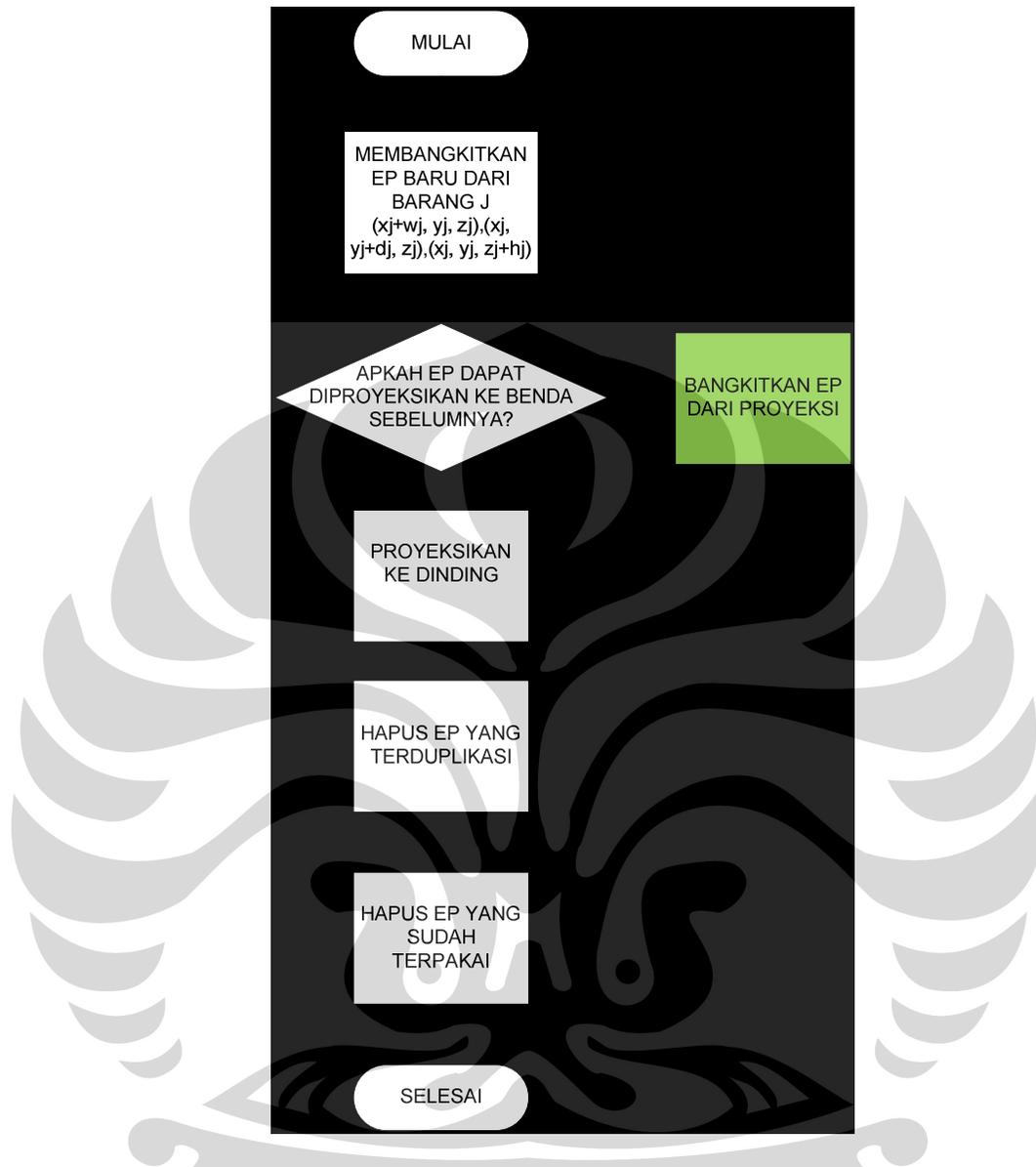
Ay untuk sumbu y, dan Az untuk sumbu z). Benda dianggap beririsan di salah satu sumbu koordinat jika nilai A untuk setiap sumbu lebih kecil daripada 0. Karena *box* atau barang yang ditinjau berdimensi 3, maka benda tersebut dikatakan beririsan dengan benda lain jika benda tersebut beririsan di tiga sumbu koordinat ($A_x, A_y, \text{ dan } A_z < 0$).

3.2.3 Memperbaharui daftar EP

Seiring bertambahnya barang yang dimuat ke dalam *bin*, jumlah *extreme point* yang tersedia meningkat. Setiap terjadinya penambahan barang ke dalam *bin* paling banyak tercipta 6 titik kandidat peletakan baru untuk peletakan barang berikutnya. Seperti algoritma-algoritma lain, proses memperbaharui daftar ini juga memiliki *time complexity*. Jika sebuah sekumpulan barang yang akan dimuat I , dan sub-kumpulan barang $I_j \subseteq I$ telah dimuat ke dalam *bin j*, maka ketika ada suatu barang baru yang dimuat ke *bin j* akan terbentuk 6 EP baru dimana masuknya 6 EP baru membutuhkan operasi sebanyak $6 * \ln(|I_j|)$ sehingga waktu yang dibutuhkan untuk proses muat di *bin j* adalah sebesar $O(|I_j| + 6 * \ln(|I_j|)) = O(|I_j|)$. Proses memperbaharui daftar EP dapat dilihat pada diagram alir gambar 3.3.

Pada diagram alir memperbaharui daftar EP (gambar 3.3) dijelaskan algoritma dalam memperbaharui daftar EP. Langkah awal algoritma ini adalah dengan membangkitkan 3 *extreme point* baru ketika barang j dimasukkan yaitu $(x_j + w_j, y_j, z_j)$, $(x_j, y_j + d_j, z_j)$, dan $(x_j, y_j, z_j + h_j)$. Kemudian titik-titik tersebut dicek apakah dapat diproyeksikan ke barang-barang sebelumnya atau tidak. Jika tidak, maka titik tersebut langsung diproyeksikan ke dinding dari *bin* tersebut. Titik Proyeksi yang dihasilkan ke dinding adalah:

- $(x_j + w_j, 0, z_j), (x_j + w_j, y_j, 0)$ jika EP yang berada pada pojok-kiri-bawah-depan barang masing-masing dapat diproyeksikan sejajar sumbu X dan sumbu Z
- $(0, y_j + d_j, z_j), (x_j, y_j + d_j, 0)$ jika EP yang berada pada pojok-kanan-bawah-belakang barang masing-masing dapat diproyeksikan sejajar sumbu Y dan sumbu Z



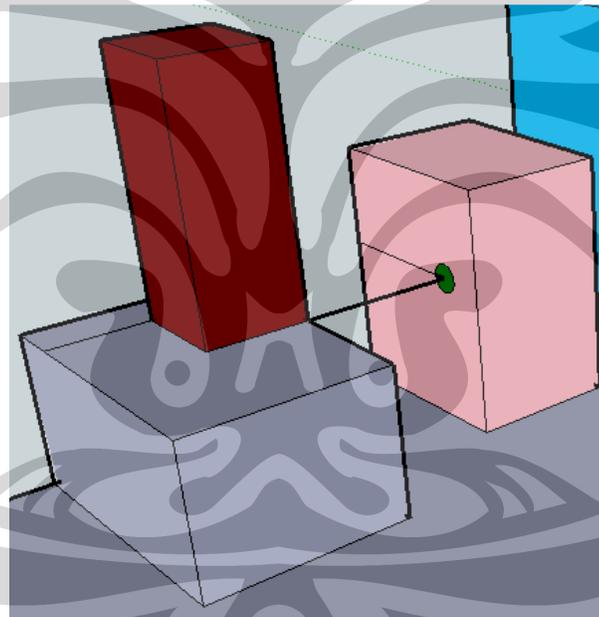
Gambar 3.3 Diagram Alir Memperbaharui Daftar EP

- $(0, yj, zj + hj), (xj, 0, zj + hj)$ jika EP yang berada pada pojok-kiri-atas-belakang barang masing-masing dapat diproyeksikan sejajar sumbu Y dan sumbu X (lihat gambar 2.2).

Jika dapat diproyeksikan, titik hasil proyeksi tersebut akan menjadi EP-EP baru untuk dimasukkan ke dalam daftar EP yang baru. Setelah semua EP yang baru dibangkitkan akibat barang j masuk, algoritma menghapus EP yang berganda

sehingga tidak ada duplikasi lain untuk setiap EP, kemudian EP yang sudah ditempati (x_j, y_j, z_j) dihapus dari daftar EP terbaru.

Dalam algoritma *update EP list* (memperbaharui daftar EP) terdapat algoritma yang juga perlu di kembangkan secara teknis, yaitu proses pembangkitan EP dari proyeksi. Proses ini memiliki algoritma tersendiri seperti yang dapat dilihat dalam diagram alir pembangkitan EP dari proyeksi (gambar 3.5). Pada algoritma ini, metode yang digunakan untuk mencari titik proyeksi dari titik tersebut adalah dengan cara mencari benda yang letaknya dilalui garis yang diproyeksikan dari EP-EP barang yang baru masuk. Syarat agar benda tersebut dilalui oleh garis proyeksi adalah titik asal proyeksinya ada didalam area bidang barang sebelumnya (gambar 3.4).



Gambar 3.4 Contoh Proyeksi Titik Pada Bidang

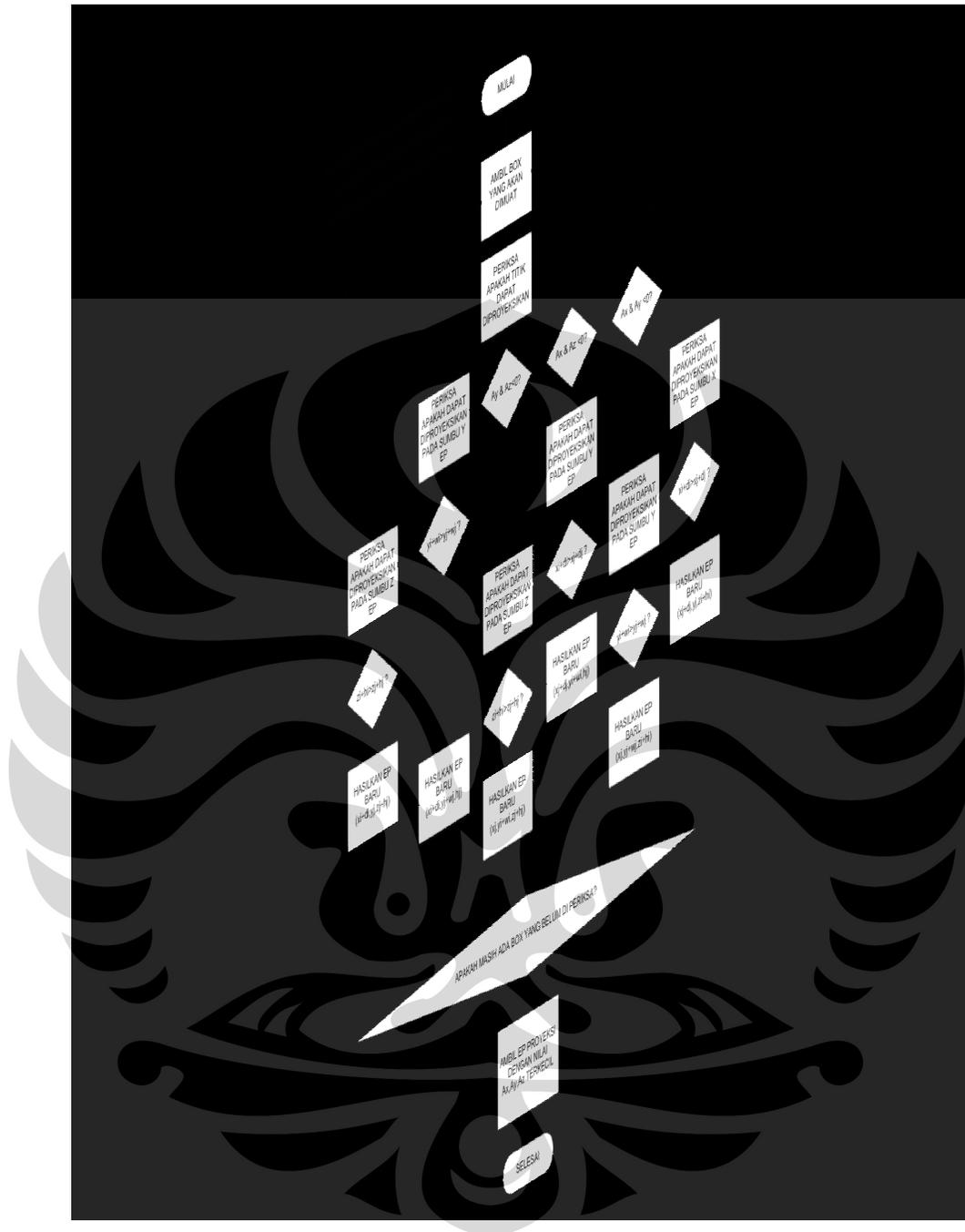
Pada gambar 3.5 terlihat bahwa EP benda berwarna coklat yang ada pada pojokkan-bawah-belakang benda diproyeksikan sejajar sumbu X. Disini terlihat bahwa titik tersebut berada dalam area bidang dari benda dibelakangnya (merah muda), walaupun terpisah dengan jarak pada sumbu X, sehingga titik tersebut dapat diproyeksikan. Untuk mengimplementasikan hal itu, diagram alir pembangkitan EP dari proyeksi (gambar 3.5) menjelaskan bahwa barang yang

baru harus beraturan dengan benda sebelumnya di 2 sumbu koordinat dan titik ekstrimnya tidak lebih besar dari titik ekstrim barang sebelumnya, yang pembagiannya adalah sebagai berikut:

- Jika A_y dan $A_z < 0$, maka titik tersebut dapat diproyeksikan sejajar dengan sumbu X. Titik yang diproyeksikan sejajar dengan sumbu X adalah titik yang berada di sumbu Z bidang (pojok-kiri-atas-belakang) dan Y bidang (pojok-kanan-bawah-belakang).
- Jika A_x dan $A_z < 0$, maka titik tersebut dapat diproyeksikan sejajar dengan sumbu Y. Titik yang diproyeksikan sejajar dengan sumbu Y adalah titik yang berada di sumbu X bidang (pojok-kiri-bawah-depan) dan Z bidang (pojok-kiri-atas-belakang).
- Jika A_x dan $A_y < 0$, maka titik tersebut dapat diproyeksikan sejajar dengan sumbu Z. Titik yang diproyeksikan sejajar dengan sumbu Z adalah titik yang berada di sumbu X bidang (pojok-kiri-bawah-depan) dan Y bidang (pojok-kanan-bawah-belakang).

Kemudian agar bisa diproyeksikan, koordinat EP pada bidang yang baru tidak boleh lebih besar nilainya dibandingkan dengan EP barang yang dibelakangnya sesuai dengan sumbu bidang dimana EP tersebut terletak. Contoh: EP yang diproyeksikan searah sumbu X (gambar 3.4) oleh barang yang berwarna coklat ke barang berwarna merah muda adalah EP yang berada pada sumbu Y bidang. Jika koordinat EP yang diproyeksikan (130,100,20), maka apabila nilai EP pada sumbu Y bidang berwarna merah jambu lebih kecil daripada 100, proyeksi (gambar 3.4) tidak dapat dilakukan.

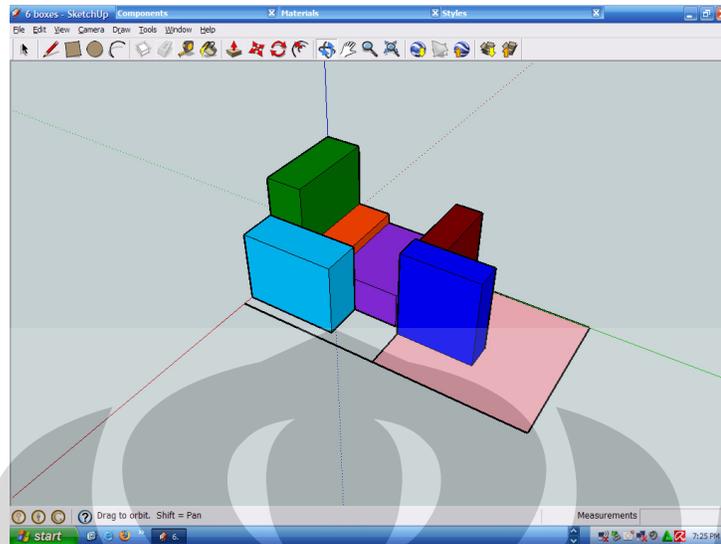
Algoritma pada diagram alir pembangkitan EP dari proyeksi (gambar 3.5) akan dioperasikan untuk setiap barang yang sudah dimuat terlebih dahulu. Jika ada banyak barang yang berpotensi untuk terjadinya proyeksi pada suatu sumbu koordinat, maka algoritma memilih benda yang terdekat dengan titik proyeksi.



Gambar 3.5 Diagram Alir Pembangkitan EP Dari Proyeksi

3.2.4 Pengurutan EP

Setelah daftar EP diperbaharui, maka list dari EP tersebut harus diurutkan kembali sesuai algoritma dari EPBH-FFD. Pengurutan dilakukan dengan



Gambar 3.7 Hasil Visualisasi Uji Validitas Manual

Setelah diuji secara manual, algoritma-algoritma yang ada diatas kemudian diimplementasikan menjadi sebuah aplikasi berbasis C++ untuk mempermudah dalam proses pengolahan data.

3.2.6 Uji Validitas Program

Data yang sudah dibangkitkan kemudian diolah menggunakan program berbasis C++. Untuk langkah awal, kita lakukan uji validitas untuk program ini. Salah satu metode pengujian adalah membandingkan hasil pengerjaan manual yang menggunakan program Microsoft Excel 2007 dengan hasil komputasi dari program. Uji validitas dibagi menjadi 2 langkah pengujian:

1. Uji Umum
2. Uji Proyeksi

3.2.6.1 Uji Umum

Pada proses ini akan akan dibandingkan hasil komputasi dengan hasil uji manual yang sudah dilakukan sebelumnya. Seperti pada gambar 3.8, data-data barang serta *bin* yang dipakai pada uji manual dimasukkan ke dalam program, kemudian program dijalankan. Karena uji coba manual dilakukan tanpa

pengurutan barang, maka hasil dari program yang diambil adalah hasil *sort by number* seperti gambar 3.9.



Gambar 3.8 Uji Validitas Umum

Hasil program pada gambar 3.9 memperlihatkan hasil yang sama pada uji manual, pengurutan dalam pemilihan EP, persyaratan dalam membuat *bin* baru juga telah sesuai dengan algoritma yang dibahas sebelumnya. Hanya saja belum diketahui apakah program ini dapat melakukan operasi yang berlaku untuk kasus-kasus tertentu yaitu proyeksi. Untuk itu, uji validitas akan dilanjutkan ke uji proyeksi.

```

uji validitas program 6 barang - Notepad
File Edit Format View Help
Heuristics for Bin Packing Problem

Sorting Heuristic: Sorted By Number
Number Of Bins needed: 2
Minimum Utilization: 0.1142257778
Solution: 1 2 3 4 5 6
No, width, Length, Height, Bin, X-Origin, Y-Origin, Z-Origin
1, 99, 75, 49, 1, 0, 0, 0
2, 39, 97, 76, 1, 99, 0, 0
3, 98, 38, 68, 1, 0, 75, 0
4, 98, 38, 68, 1, 0, 0, 49
5, 28, 80, 87, 2, 0, 0, 0
6, 26, 78, 94, 2, 28, 0, 0

```

Gambar 3.9 Hasil Uji Validitas Umum

3.2.6.2 Uji Proyeksi

Dalam uji proyeksi ini, kita akan menguji kemampuan program dalam proyeksi yang sejajar sumbu X, sumbu Y, serta sumbu Z.

3.2.6.2.1 Proyeksi Sejajar Sumbu X

Uji validitas ini memasukkan 4 barang secara berurutan kedalam bin dengan ukuran 20x50x20. Proses dari uji ini dapat dilihat dari gambar 3.10 dan 3.11.

Gambar 3.10 Uji Validitas Proyeksi X

Digambar 3.10 terlihat ada 4 barang yang dimuat, dan hasilnya terlihat seperti pada gambar 3.11 berikut ini,

```

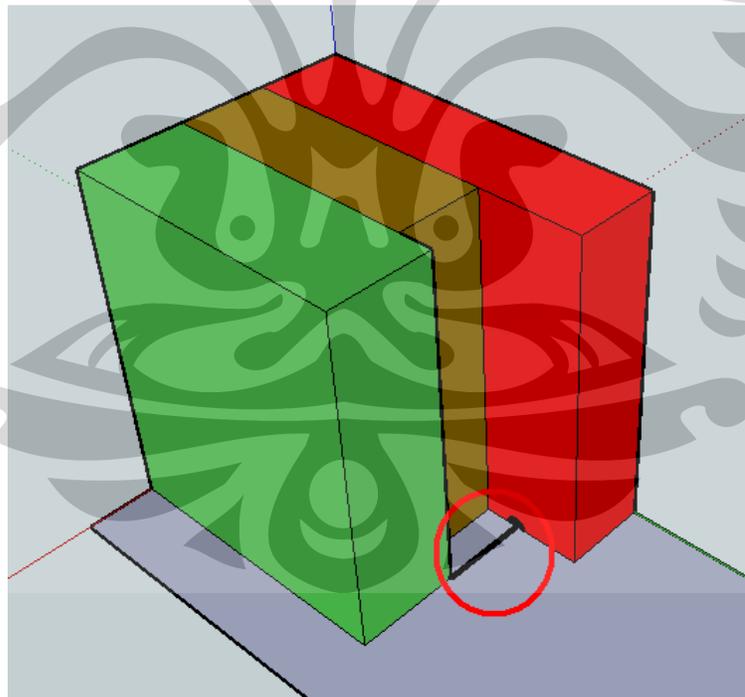
tes proyeksi x - Notepad
File Edit Format View Help
20x50x20
Heuristics for Bin Packing Problem

Sorting Heuristic: Sorted By Number
Number of Bins needed: 1
Minimum Utilization: 0.357
Solution: 1 2 3 4
No, width, Length, Height, Bin, x-origin, y-origin, z-origin
1, 5, 18, 20, 1, 0, 0, 0
2, 5, 15, 20, 1, 5, 0, 0
3, 6, 17, 20, 1, 10, 0, 0
4, 6, 15, 20, 1, 5, 17, 0

```

Gambar 3.11 Hasil Uji Validitas Proyeksi X

Disini terlihat bahwa item nomor 4 diletakkan pada titik hasil proyeksi barang nomor 3 ke nomor 2 seperti yang ditampilkan gambar 3.12.

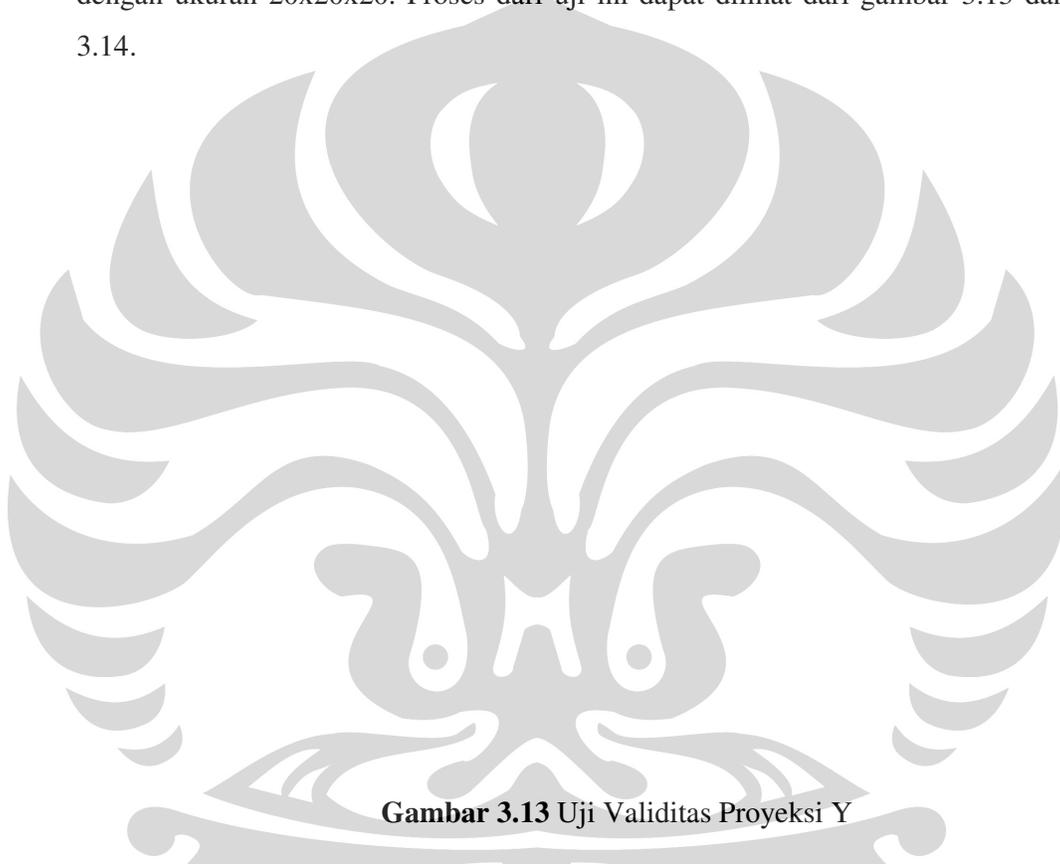


Gambar 3.12 Visualisasi Hasil Uji Validitas Proyeksi X

Pada gambar 3.12 terlihat bahwa EP yang berada pada sumbu Y bidang barang (10,17,0) diproyeksikan sejajar dengan sumbu X ke barang pertama. Maka titik hasil proyeksi tersebut (5,17,0) menjadi EP untuk peletakkan barang nomor 4.

3.2.6.2.2 Proyeksi Sejajar Sumbu Y

Uji validitas ini memasukkan 4 barang secara berurutan kedalam bin dengan ukuran 20x20x20. Proses dari uji ini dapat dilihat dari gambar 3.13 dan 3.14.



Gambar 3.13 Uji Validitas Proyeksi Y

```
tes proyeksi y - Notepad
File Edit Format View Help
Heuristics for Bin Packing Problem

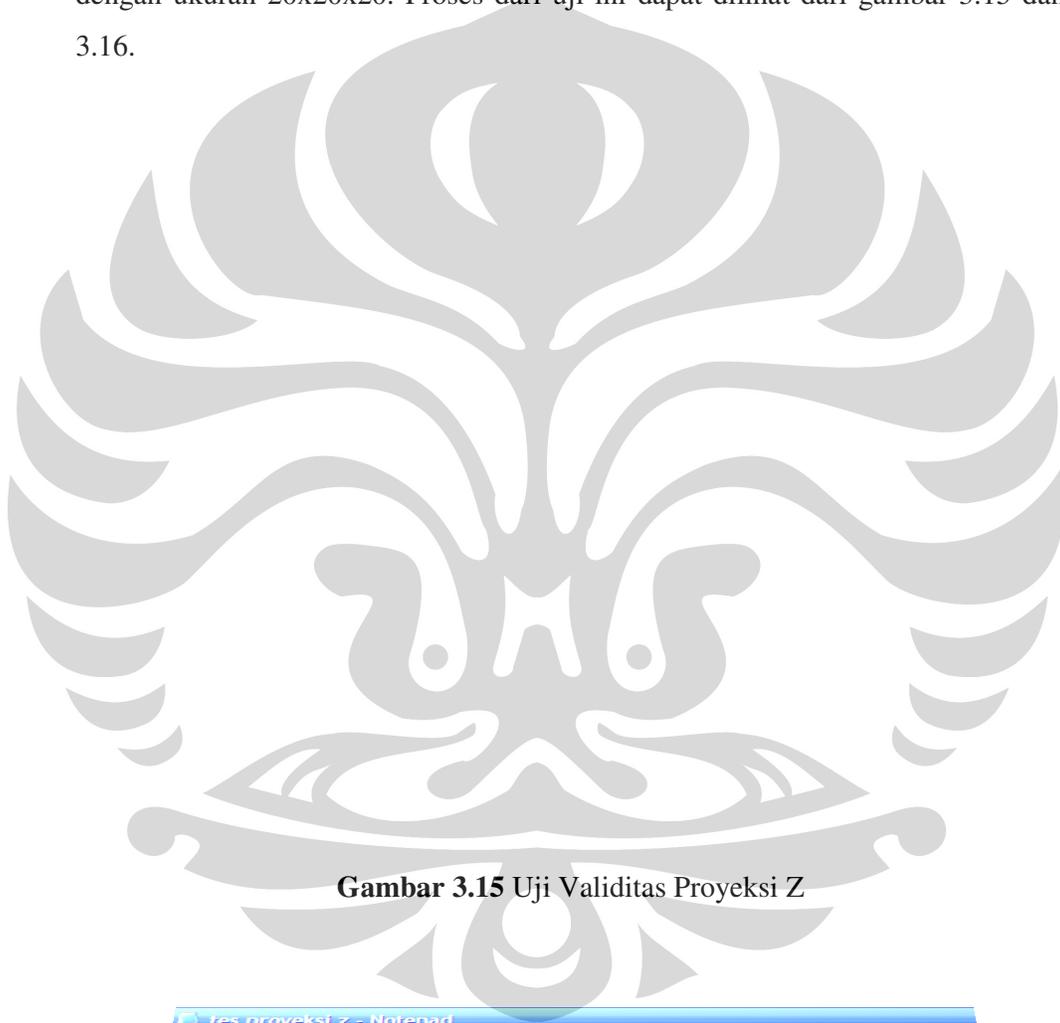
Sorting Heuristic: sorted By Number
Number Of Bins needed: 1
Minimum Utilization: 0.69
Solution: 1 2 3 4
No, width, Length, Height, Bin, X-Origin, Y-Origin, Z-Origin
1, 12, 5, 20, 1, 0, 0, 0
2, 14, 6, 10, 1, 0, 5, 0
3, 18, 8, 20, 1, 0, 11, 0
4, 2, 15, 20, 1, 18, 0, 0
```

Gambar 3.14 Hasil Uji Validitas Proyeksi Y

Gambar 3.14 memperlihatkan bahwa EP yang dipakai untuk meletakkan barang nomor 4 (18,0,0) merupakan hasil proyeksi EP yang dihasilkan barang nomor 3 (18,11,0) yang ditarik sejajar sumbu X ke barang pertama.

3.2.6.2.3 Proyeksi Sejajar Sumbu Z

Uji validitas ini memasukkan 3 barang secara berurutan kedalam bin dengan ukuran 20x20x20. Proses dari uji ini dapat dilihat dari gambar 3.15 dan 3.16.



Gambar 3.15 Uji Validitas Proyeksi Z

```

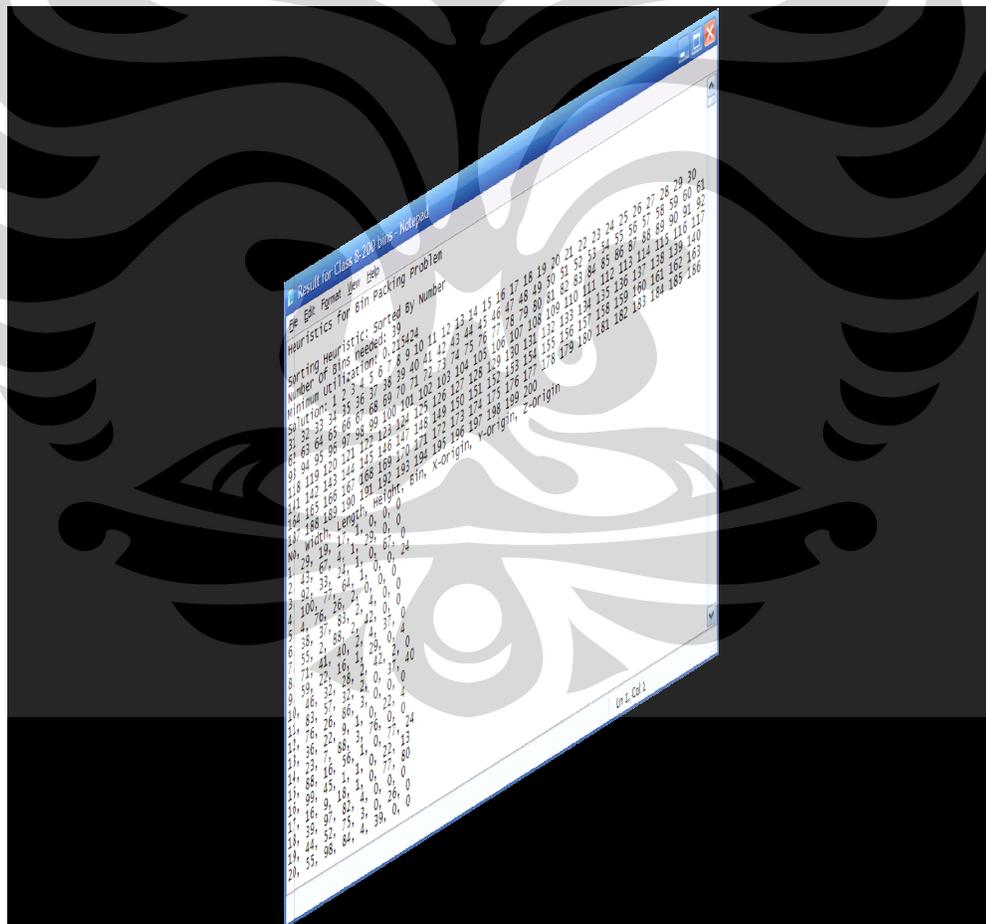
tes proyeksi z - Notepad
File Edit Format View Help
Heuristics for Bin Packing Problem
Sorting Heuristic: Sorted By Number
Number Of Bins needed: 1
Minimum Utilization: 0.55
Solution: 1 2 3
No. width, length, Height, Bin, x-origin, y-origin, z-origin
1, 20, 10, 10, 1, 0, 0, 0
2, 10, 15, 10, 1, 0, 0, 10
3, 10, 5, 18, 1, 0, 15, 0
  
```

Gambar 3.16 Hasil Uji Validitas Proyeksi Z

Gambar 3.16 memperlihatkan bahwa EP yang dipakai untuk meletakkan barang nomor 3 (0,15,0) merupakan hasil proyeksi EP yang dihasilkan barang nomor 2 (0,15,10) yang ditarik sejajar sumbu Z ke alas *bin*.

3.3 Pengolahan Data

Pada proses pengolahan data, data-data yang telah dihasilkan pada proses data generation kemudian diolah dengan program yang telah dibangun berdasarkan algoritma EPDH-FFD. Data yang dihasilkan dari metode *data generation* adalah sebanyak 200 data tiap kelasnya. Data-data ini kemudian diolah per kelas dengan kelipatan 20 data untuk tiap pengolahannya. Untuk melihat salah satu hasil keluaran dari pengolahan data dapat dilihat pada gambar 3.17.



Gambar 3.17 Contoh Keluaran Pengolahan Data

Keluaran dari pengolahan data adalah jumlah *bin* yang diperlukan pada proses muat, nilai utilisasi terkecil dari seluruh *bin*, solusi atau urutan pemuatan, dan koordinat peletakan dari barang tersebut. Akan tetapi, untuk memenuhi tujuan penelitian ini maka informasi yang dijadikan pembanding adalah jumlah *bin* yang dibutuhkan. Setelah semua data diolah, kemudian hasil pengolahan dikumpulkan dan dikelompokkan berdasarkan kelasnya. Hasil-hasil pengolahan data yang telah dielompokkan per kelas dapat dilihat pada bab pembahasan.





BAB 4 PEMBAHASAN

4.1 Pembahasan Hasil Pengolahan Data

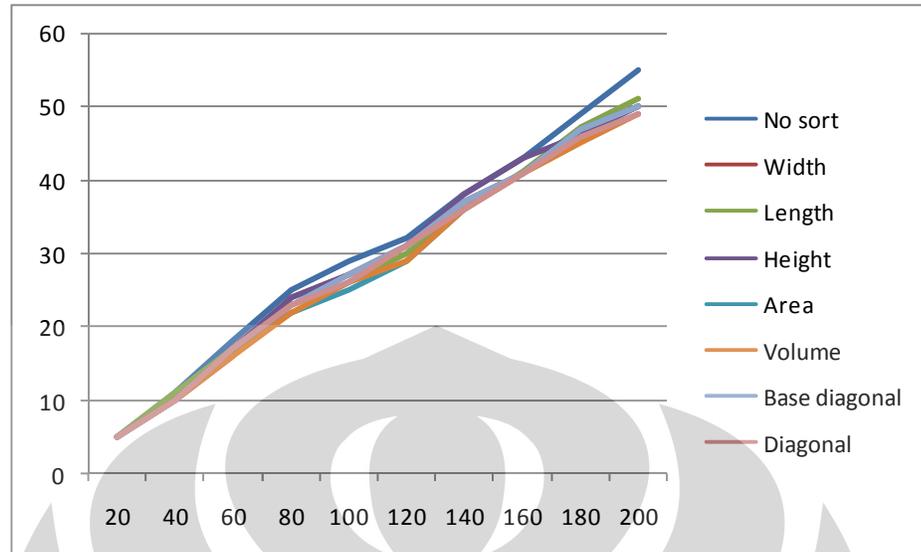
Hasil Pengolahan data pada bab sebelumnya kemudian di kelompokkan menjadi 8 kelas. Dari kedelapan kelas ini, yang juga mewakili kelas-kelas dari sampel barang, akan dibahas metode pengurutan apa yang terbaik. Pembahasan untuk tiap kelas ini juga dibagi menjadi 4, yaitu pembahasan untuk jumlah barang sedikit ($\text{jumlah barang} \leq 60$), barang dengan jumlah sedang ($60 < \text{jumlah barang} \leq 100$) dan jumlah banyak ($100 < \text{jumlah barang} \leq 200$) serta pembahasan keseluruhan.

4.1.1 Pembahasan *Class 1*

Hasil pengolahan data untuk kelas barang nomor 1 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.1 Hasil pengolahan data kelas 1

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	5	11	18	25	29	32	38	43	49	55	305
Width	5	10	16	24	27	31	37	41	47	50	288
Length	5	11	17	23	26	30	36	41	47	51	287
Height	5	10	17	24	27	31	38	43	46	50	291
Area	5	10	17	22	25	29	36	41	47	50	282
Volume	5	10	16	22	26	29	36	41	45	49	279
Base											
diagonal	5	10	17	23	27	31	37	41	47	50	288
Diagonal	5	10	17	23	26	31	36	41	46	49	284



Gambar 4.1 Grafik jumlah barang vs jumlah *bin* kelas 8

Dari gambar 4.1 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 1. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 279, berbeda 3 *bin* dari metode pengurutan berbasis luas alas barang (282).

Tabel 4.2 Peforma Metode Tiap Kriteria Ukuran Kelas 1

Method	Amount		
	Small	Medium	Large
No sort	34	54	217
Width	31	51	206
Length	33	49	205
Height	32	51	208
Area	32	47	203
Volume	31	48	200
Base diagonal	32	50	206
Diagonal	32	49	203

Pada table 4.2, terlihat bahwa tiap ukuran set barang baik jika dimuat dengan metode *sorting* yang berbeda-beda. Untuk kelas dengan ukuran jumlah yang kecil, metode *width* dan *volume* unggul, *base area* unggul di ukuran jumlah

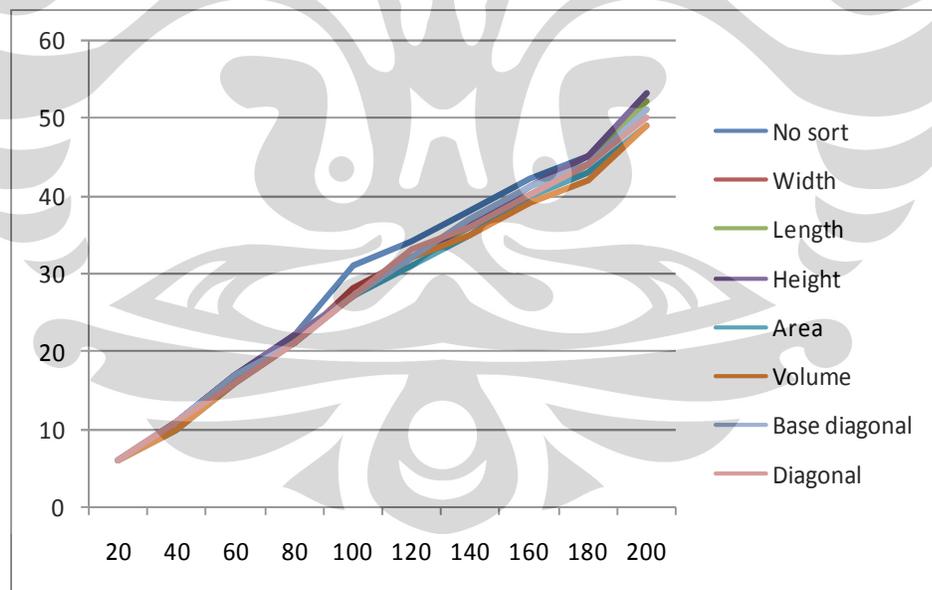
sedang, dan *volume* menunjukkan performa yang baik pada ukuran jumlah yang besar.

4.1.2 Pembahasan *Class 2*

Hasil pengolahan data untuk kelas barang nomor 2 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.3 Hasil pengolahan data kelas 2

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	6	11	17	22	31	34	38	42	45	53	299
Width	6	11	17	21	28	32	36	40	44	51	286
Length	6	10	16	22	27	32	36	40	44	52	285
Height	6	11	17	22	27	32	36	41	45	53	290
Area	6	10	16	21	27	31	35	40	43	49	278
Volume	6	10	16	21	27	32	35	39	42	49	277
Base diagonal	6	11	17	21	27	32	37	41	44	51	287
Diagonal	6	11	16	21	27	33	36	40	44	50	284



Gambar 4.2 Grafik jumlah barang vs jumlah *bin* kelas 2

Dari gambar 4.2 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 2. Jumlah total *bin* yang

dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 277, berbeda 1 *bin* dari metode pengurutan berbasis luas alas barang (278).

Tabel 4.4 Peforma Metode Tiap Kriteria Ukuran Kelas 2

Method	Amount		
	Small	Medium	Large
No sort	34	53	212
Width	34	49	203
Length	32	49	204
Height	34	49	207
Area	32	48	198
Volume	32	48	197
Base diagonal	34	48	205
Diagonal	33	48	203

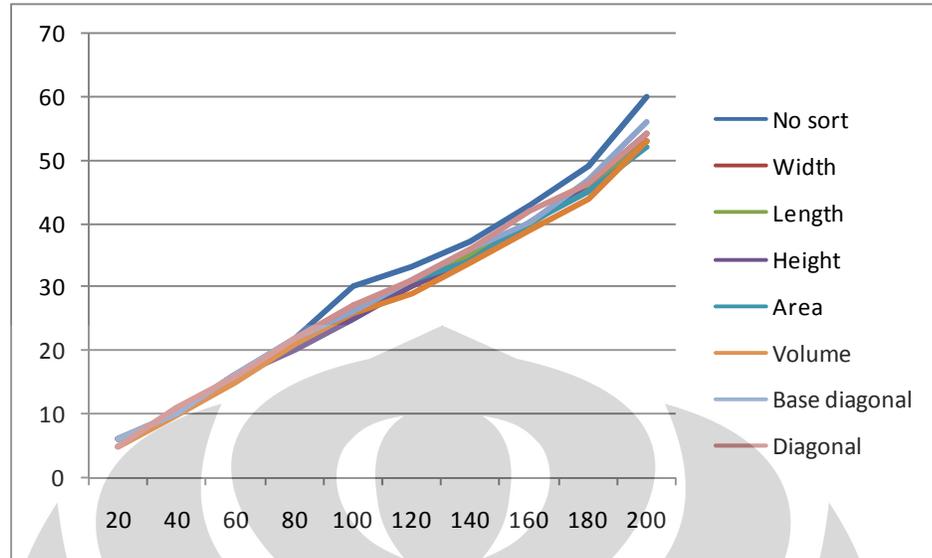
Table 4.4 menunjukkan bahwa hampir semua metode *sorting* memiliki performa baik pada set ukuran kecil dan sedang barang-barang kelas 2, sedangkan untuk set ukuran besar metode *volume* memiliki performa cukup menonjol bersama *base area*.

4.1.3 Pembahasan *Class 3*

Hasil pengolahan data untuk kelas barang nomor 3 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.5 Hasil pengolahan data kelas 3

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	6	10	16	22	30	33	37	43	49	60	306
Width	6	10	16	21	27	30	35	40	46	53	284
Length	6	10	15	21	26	30	35	40	46	53	282
Height	5	10	16	20	25	30	34	40	46	54	280
Area	6	10	15	21	26	31	34	40	45	52	280
Volume	5	10	15	21	26	29	34	39	44	53	276
Base diagonal	6	10	16	22	26	31	36	40	47	56	290
Diagonal	5	11	16	22	27	31	36	42	46	54	290



Gambar 4.3 Grafik jumlah barang vs jumlah *bin* kelas 3

Dari gambar 4.3 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 3. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 276, berbeda 4 *bin* dari metode pengurutan berbasis luas alas dan tinggi barang (280).

Tabel 4.6 Performa Metode Tiap Kriteria Ukuran Kelas 3

Method	Amount		
	Small	Medium	Large
No sort	32	52	222
Width	32	48	204
Length	31	47	204
Height	31	45	204
Area	31	47	202
Volume	30	47	199
Base diagonal	32	48	210
Diagonal	32	49	209

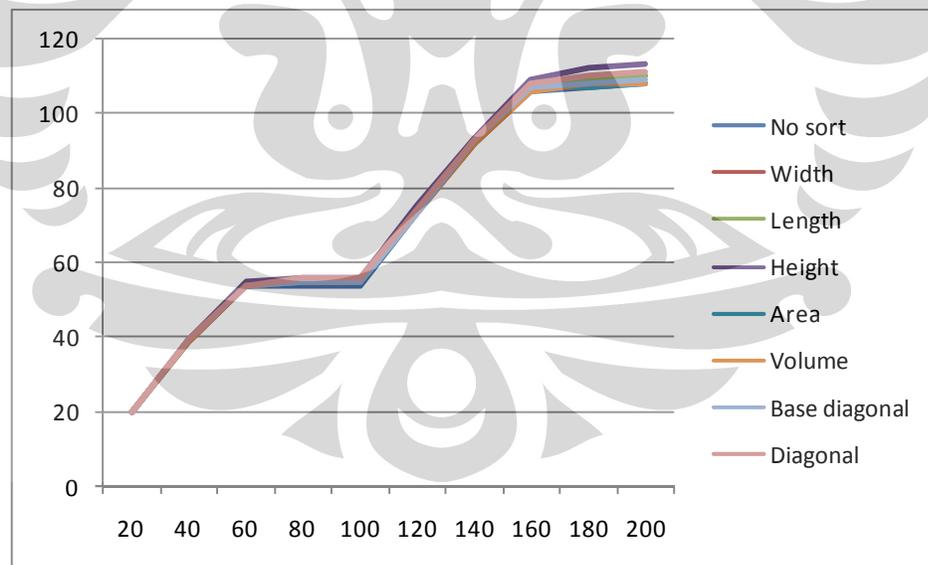
Pada kelas 3, set ukuran kecil lebih efektif jika menggunakan metode *volume*, sedangkan untuk set ukuran sedang metode *width* lebih unggul, dan pada jumlah barang yang besar *volume* lagi-lagi menunjukkan performa yang baik.

4.1.4 Pembahasan *Class 4*

Hasil pengolahan data untuk kelas barang nomor 4 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.7 Hasil pengolahan data kelas 4

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	20	39	54	54	54	73	92	106	107	108	707
Width	20	39	54	55	55	74	93	107	109	110	716
Length	20	39	55	55	55	74	93	108	109	110	718
Height	20	39	55	56	56	75	93	109	112	113	728
Area	20	39	54	55	55	73	93	107	107	108	711
Volume	20	39	54	55	55	73	92	106	108	108	710
Base diagonal	20	39	54	55	55	73	93	107	108	109	713
Diagonal	20	39	54	56	56	74	93	108	110	111	721



Gambar 4.4 Grafik jumlah barang vs jumlah *bin* kelas 4

Dari gambar 4.4 terlihat bahwa metode pengurutan berdasarkan barang yang terlebih dahulu masuk memiliki performa yang terbaik untuk kumpulan barang

kelas 4. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 707, berbeda 1 *bin* dari metode pengurutan berbasis luas alas (710).

Tabel 4.8 Peforma Metode Tiap Kriteria Ukuran Kelas 4

Method	Amount		
	Small	Medium	Large
No sort	113	108	486
Width	113	110	493
Length	114	110	494
Height	114	112	502
Area	113	110	488
Volume	113	110	487
Base diagonal	113	110	490
Diagonal	113	112	496

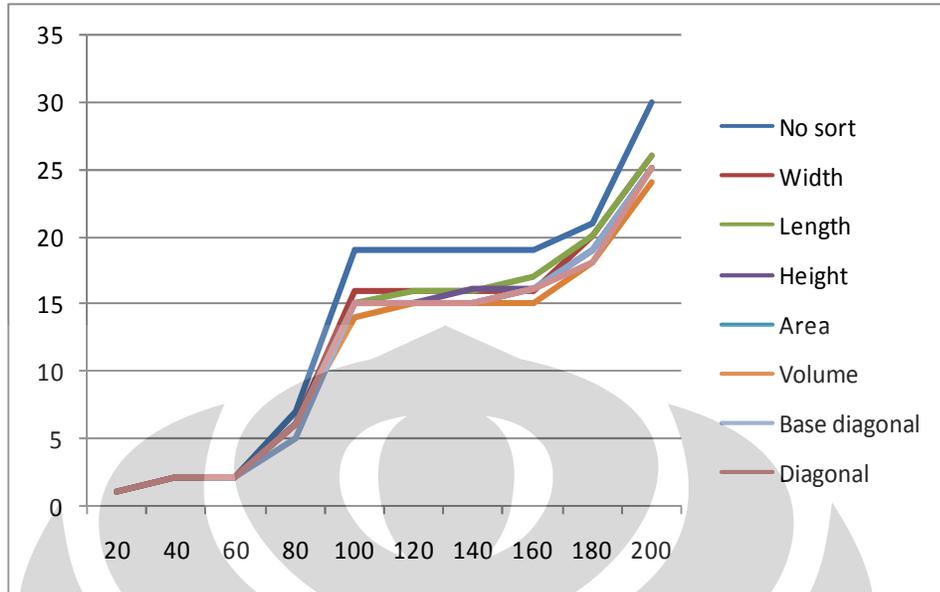
Pada kelas 4, rata-rata metode *sorting* memiliki peforma yang sama pada set ukuran kecil dan sedang. Pada set ukuran besar, metode *volume* dan *base area* memiliki nilai yang terbaik.

4.1.5 Pembahasan *Class5*

Hasil pengolahan data untuk kelas barang nomor 5 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.9 Hasil pengolahan data kelas 5

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	1	2	2	7	19	19	19	19	21	30	139
Width	1	2	2	6	16	16	16	16	20	26	121
Length	1	2	2	6	15	16	16	17	20	26	121
Height	1	2	2	6	15	15	16	16	19	25	117
Area	1	2	2	6	15	15	15	16	19	25	116
Volume	1	2	2	6	14	15	15	15	18	24	112
Base diagonal	1	2	2	5	15	15	15	16	19	25	115
Diagonal	1	2	2	6	15	15	15	16	18	25	115



Gambar 4.5 Grafik jumlah barang vs jumlah *bin* kelas 5

Dari gambar 4.5 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 5. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 112, berbeda 3 *bin* dari metode pengurutan berbasis diagonal alas dan diagonal ruang (115).

Tabel 4.10 Performa Metode Tiap Kriteria Ukuran Kelas 5

Method	Amount		
	Small	Medium	Large
No sort	5	26	108
Width	5	22	94
Length	5	21	95
Height	5	21	91
Area	5	21	90
Volume	5	20	87
Base diagonal	5	20	90
Diagonal	5	21	89

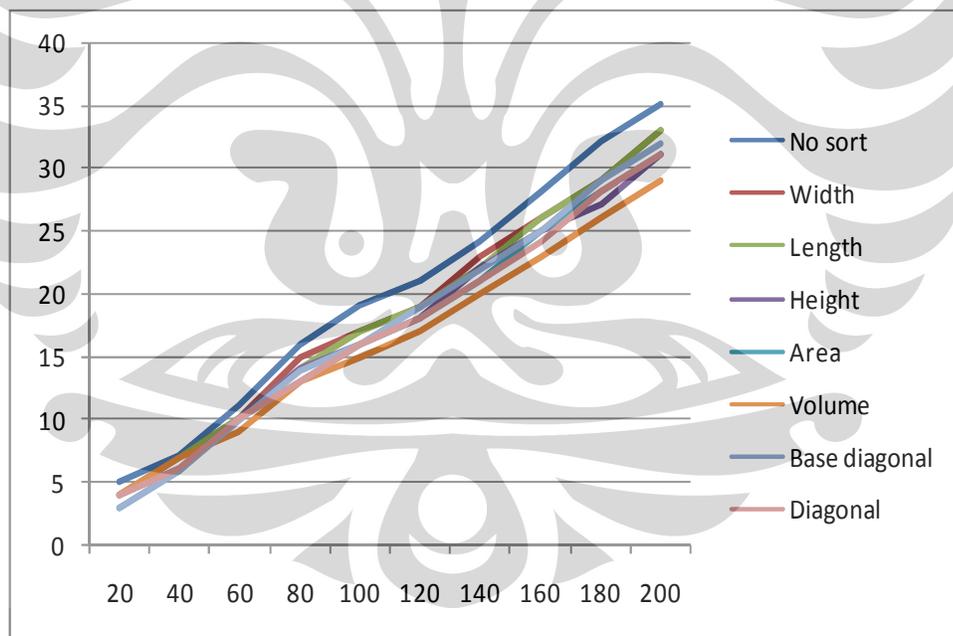
Kecuali metode *no sort*, hampir semua metode efektif dalam menangani barang-barang kelas 5 untuk ukuran jumlah kecil dan sedang. Sedangkan *volume* tetap menguasai barang-barang ukuran jumlah besar dalam hal performa.

4.1.6 Pembahasan *Class 6*

Hasil pengolahan data untuk kelas barang nomor 6 ($W = D = H = 10$) adalah sebagai berikut:

Tabel 4.11 Hasil pengolahan data kelas 6

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	5	7	11	16	19	21	24	28	32	35	198
Width	4	7	10	15	17	19	23	26	29	33	183
Length	4	7	10	14	17	19	22	26	29	33	181
Height	4	6	10	14	16	18	22	25	27	31	173
Area	3	6	10	13	16	18	21	25	28	31	171
Volume	4	7	9	13	15	17	20	23	26	29	163
Base diagonal	3	6	10	14	16	19	22	25	29	32	176
Diagonal	4	6	10	13	16	18	21	24	28	31	171



Gambar 4.6 Grafik jumlah barang vs jumlah *bin* kelas 6

Dari gambar 4.6, terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 6. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar

163, berbeda 8 *bin* dari metode pengurutan berbasis area dan diagonal ruang (171).

Tabel 4.12 Peforma Metode Tiap Kriteria Ukuran Kelas 6

Method	Amount		
	Small	Medium	Large
No sort	23	35	140
Width	21	32	130
Length	21	31	129
Height	20	30	123
Area	19	29	123
Volume	20	28	115
Base diagonal	19	30	127
Diagonal	20	29	122

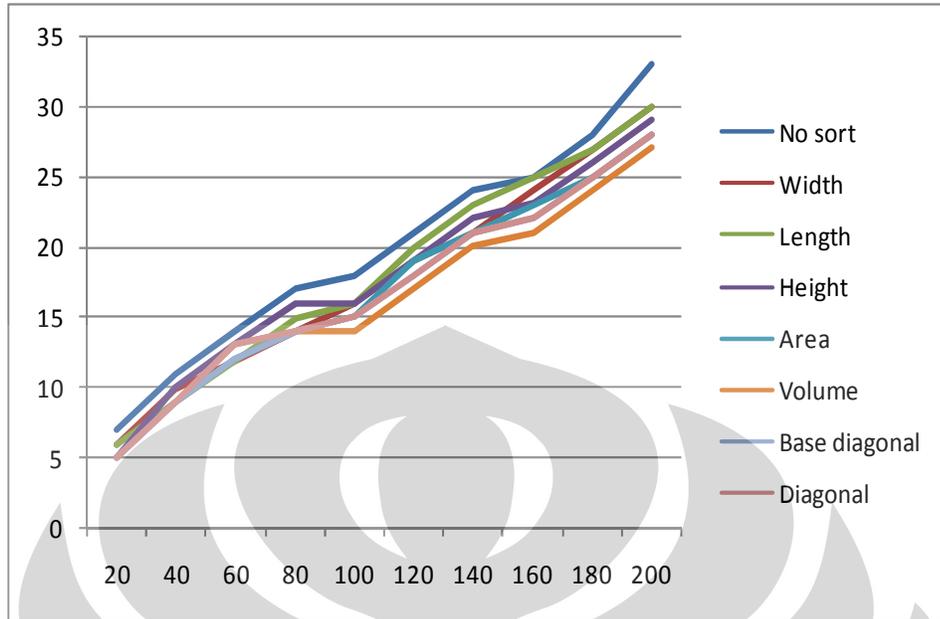
Untuk kelas 6, metode *base area* dan *base diagonal* memiliki peforma yang baik pada ukuran jumlah kecil dan sedang. Ukuran jumlah besar lebih efektif jika diselesaikan dengan metode *volume*.

4.1.7 Pembahasan Hasil *Class 7*

Hasil pengolahan data untuk kelas barang nomor 7 ($W = D = H = 40$) adalah sebagai berikut:

Tabel 4.13 Hasil pengolahan data kelas 7

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	7	11	14	17	18	21	24	25	28	33	198
Width	6	10	12	14	16	19	21	24	27	30	179
Length	6	9	12	15	16	20	23	25	27	30	183
Height	5	10	13	16	16	19	22	23	26	29	179
Area	5	9	12	14	15	19	21	23	25	28	171
Volume	5	9	13	14	14	17	20	21	24	27	164
Base diagonal	5	9	12	14	15	18	21	22	25	28	169
Diagonal	5	9	13	14	15	18	21	22	25	28	170



Gambar 4.7 Grafik jumlah barang vs jumlah *bin* kelas 7

Dari gambar 4.7 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 7. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 164, berbeda 5 *bin* dari metode pengurutan berbasis diagonal alas (169).

Tabel 4.14 Peforma Metode Tiap Kriteria Ukuran Kelas 7

Method	Amount		
	Small	Medium	Large
No sort	32	35	131
Width	28	30	121
Length	27	31	125
Height	28	32	119
Area	26	29	116
Volume	27	28	109
Base diagonal	26	29	114
Diagonal	27	29	114

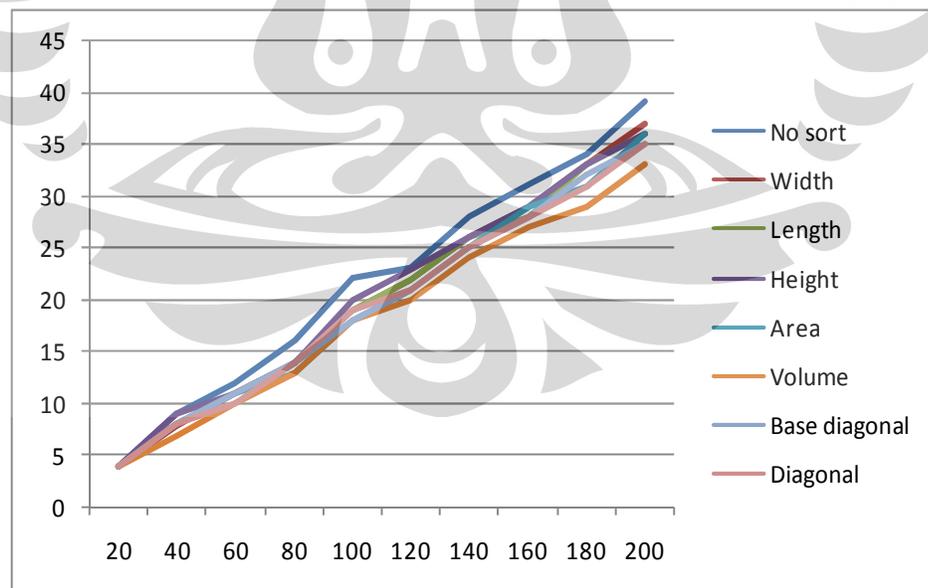
Pada table 4.14 terlihat bahwa untuk ukuran jumlah kecil metode *base area* dan *base diagonal* memiliki peforma yang baik. Untuk ukuran jumlah besar, *volume* tetap terbaik.

4.1.8 Pembahasan Hasil Class 8

Hasil pengolahan data untuk kelas barang nomor 8 ($W = D = H = 100$) adalah sebagai berikut:

Tabel 4.15 Hasil pengolahan data kelas 8

Sorting method	n										TOTAL
	20	40	60	80	100	120	140	160	180	200	
No sort	4	9	12	16	22	23	28	31	34	39	218
Width	4	8	11	14	19	22	26	29	33	37	203
Length	4	8	11	14	19	22	26	28	33	36	201
Height	4	9	11	14	20	23	26	29	33	36	205
Area	4	8	11	13	18	21	25	29	31	36	196
Volume	4	7	10	13	18	20	24	27	29	33	185
Base diagonal	4	8	11	14	18	21	25	28	32	35	196
Diagonal	4	8	10	14	19	21	25	28	31	35	195



Gambar 4.8 Grafik jumlah barang vs jumlah *bin* kelas 8

Dari gambar 4.8 terlihat bahwa metode pengurutan berdasarkan *volume* memiliki performa yang terbaik untuk kumpulan barang kelas 8. Jumlah total *bin* yang dibutuhkan untuk 10 kasus jumlah barang yakni 20-200 (kelipatan 20) sebesar 185, berbeda 10 *bin* dari metode pengurutan berbasis diagonal ruang (195).

Tabel 4.16 Peforma Metode Tiap Kriteria Ukuran Kelas 8

Method	Amount		
	Small	Medium	Large
No sort	25	38	155
Width	23	33	147
Length	23	33	145
Height	24	34	147
Area	23	31	142
Volume	21	31	133
Base diagonal	23	32	141
Diagonal	22	33	140

Pada kelas ini, metode *volume* mendominasi dengan menunjukkan performa yang sama baiknya untuk ukuran jumlah kecil, sedang, dan besar. Lalu diikuti *base area* yang memiliki performa sama hanya untuk ukuran jumlah sedang.

4.2 Evaluasi Hasil

4.2.1 Evaluasi Keseluruhan

Dari hasil-hasil yang telah dibahas, terlihat bahwa metode pengurutan berbasis *volume* memperlihatkan performa yang paling baik dibanding 7 metode pengurutan lainnya. Keunggulan dari metode *sorting* ini dapat terlihat pada seluruh kelas sampel barang dimana *bin* yang terbentuk dari metode ini paling sedikit dibanding yang lain. Hanya saja pada implementasi di dunia nyata, tidak mudah untuk membandingkan *volume* 2 buah benda jika perbedaan *volumenya* tidak ekstrim. Untuk itu kita juga melihat metode *sorting* yang memiliki hasil terbaik kedua untuk melihat solusi alternatif untuk tiap kelas.

Tabel 4.17 Peringkat peforma metode *sorting* pada tiap kelas

Classification	1st stand	2nd stand	gap [bin(s)]
Class 1	volume	base area	3
Class 2	volume	base area	1
Class 3	volume	base area, height	4
Class 4	volume	base area	1
Class 5	volume	base diagonal, diagonal	3
Class 6	volume	base area, diagonal	8
Class 7	volume	base diagonal	5
Class 8	volume	diagonal	1

Pada table 4.8 terlihat metode pengurutan berbasis luas alas (*base area*) mendominasi 2^{nd} stand, dimana metode ini meraih peringkat kedua pada kelas 1, 2, 3, 4, dan 6. Hal ini juga dengan karakteristik dari barang-barang pada kelas-kelas tersebut dimana terlihat bahwa metode *sorting* berbasis luas alas ini tidak mendominasi kelas-kelas yang mayoritas barangnya kecil-kecil relative terhadap *bin*, sebaliknya menunjukkan peforma yang baik pada barang-barang yang relative besar. Bahkan untuk kelas 2, metode *sorting* berbasis luas alas ini hanya berselisih tipis dengan metode *volume*. Maksimalnya peforma metode *base area* pada kelas 2 disebabkan mayoritas luas alas barang-barang pada kelas tersebut besar relatif luas *bin*.

Tabel 4.18 Hasil uji keseluruhan

Class	Amount of Generated Bin							
	No sort	Width	Length	Height	Area	Volume	Base diagonal	Diagonal
1	305	288	287	291	282	279	288	284
2	299	286	285	290	278	277	287	284
3	306	284	282	280	280	276	290	290
4	707	716	718	728	711	710	713	721
5	139	121	121	117	116	112	115	115
6	198	183	181	173	171	163	176	171
7	198	179	183	179	171	164	169	170
8	218	203	201	205	196	185	196	195
Grand Total	2370	2260	2258	2263	2205	2166	2234	2230

4.2.2 Evaluasi Metode Pengurutan

Evaluasi hasil keseluruhan juga membutuhkan evaluasi spesifik pada tiap-tiap metode *sorting*. Oleh karena itu, dilakukan pembahasan serta evaluasi performa dari tiap-tiap metode yang di tinjau. Evaluasi dari tiap-tiap metode *sorting* tersebut adalah sebagai berikut:

4.2.2.1 Peforma Metode *No Sort*

Metode *no sort* merupakan perwakilan dari metode yang biasa dilakukan di dunia nyata dimana barang yang pertama kali dimuat adalah barang yang pertama masuk. Performa dari metode ini, seperti yang sudah diperkirakan, paling rendah dibandingkan metode lain di hampir tiap kelas sehingga dapat dibilang tidak memiliki keunggulan apa-apa kecuali kecepatan dalam proses muat (tidak masuk dalam factor yang ditinjau).

4.2.2.2 Peforma Metode *Width*

Metode *Width* atau metode pengurutan berbasis lebar barang merupakan metode *sorting* yang tidak menunjukkan performa terlalu baik, dimana berselisih lebih dari 8 *bin* pada tiap kelas terhadap metode *sorting* terbaik pada kelas tersebut. Kelemahan pada metode ini adalah pada ketidak-seimbangan proporsi barang dimana pemilihan barang berdasarkan satu dimensi dapat menyebabkan banyaknya ruang kosong tidak terpakai terutama sepanjang sumbu X dan Z *bin* karena dimensi kecuali *width* pemilihannya tidak diatur.

4.2.2.3 Peforma Metode *Length*

Performa metode ini juga sama dengan 2 metode lainnya, metode ini juga tidak menunjukkan keunggulan di salah satu kelas. Hal ini mungkin juga dikarenakan tidak adanya kelas yang cocok dengan karakter metode *sorting* ini. Kelemahan pada metode ini adalah pada ketidak-seimbangan proporsi barang dimana pemilihan barang berdasarkan satu dimensi dapat menyebabkan banyaknya ruang kosong tidak terpakai terutama sepanjang sumbu Y dan Z *bin*.

4.2.2.4 Peforma Metode *Height*

Metode ini lebih banyak di posisi bawah performanya dibandingkan metode lain setelah *no sort*. Akan tetapi, metode ini dianggap cukup baik pada kelas 6 dimana barang-barangnya di random dengan dimensi yang intervalnya cukup besar relative ukuran *bin*. Sama seperti metode *length* dan *width* metode ini juga menyebabkan pemilihan luas dari barang tidak teratur sehingga luas area dari *bin* tidak dimanfaatkan dengan baik. Hal ini sekaligus memperkecil utilisasi ruangan dari *bin* tersebut.

4.2.2.5 Peforma Metode *Base Area*

Peforma dari metode ini cukup baik dimana metode *sorting* ini menguasai *2nd stand* dimana merupakan metode terbaik kedua setelah metode *volume*. Hanya saja peforma dari metode ini relatif lemah dalam menangani kasus pada kelas 5, dimana pada kelas ini barang-barang yang dimuat berukuran kecil relatif *bin*-nya. Metode *base area* ini cukup baik performanya karena seimbang proporsi dimensi alas barang yang dipilih. Luas area *bin* dimanfaatkan dengan baik dan ini membuat proses muat lebih optimal.

4.2.2.6 Peforma Metode *Volume*

Metode yang memiliki peforma terbaik, hanya kalah oleh peforma *no sort* pada barang-barang kelas 4 dimana berselisih 3 *bin* pada hasil keseluruhan. Metode ini rata-rata pada tiap kelas kurang baik atau biasa saja apabila jumlah barang <60 tetapi sangat baik pada jumlah diatasnya. Secara keseluruhan, metode ini merupakan metode yang terbaik untuk diimplementasikan dibanding metode lainnya. Ini disebabkan metode ini dapat memanfaatkan ruangan *bin* dengan baik, sehingga optimasi dari penggunaan *bin* juga meningkat.

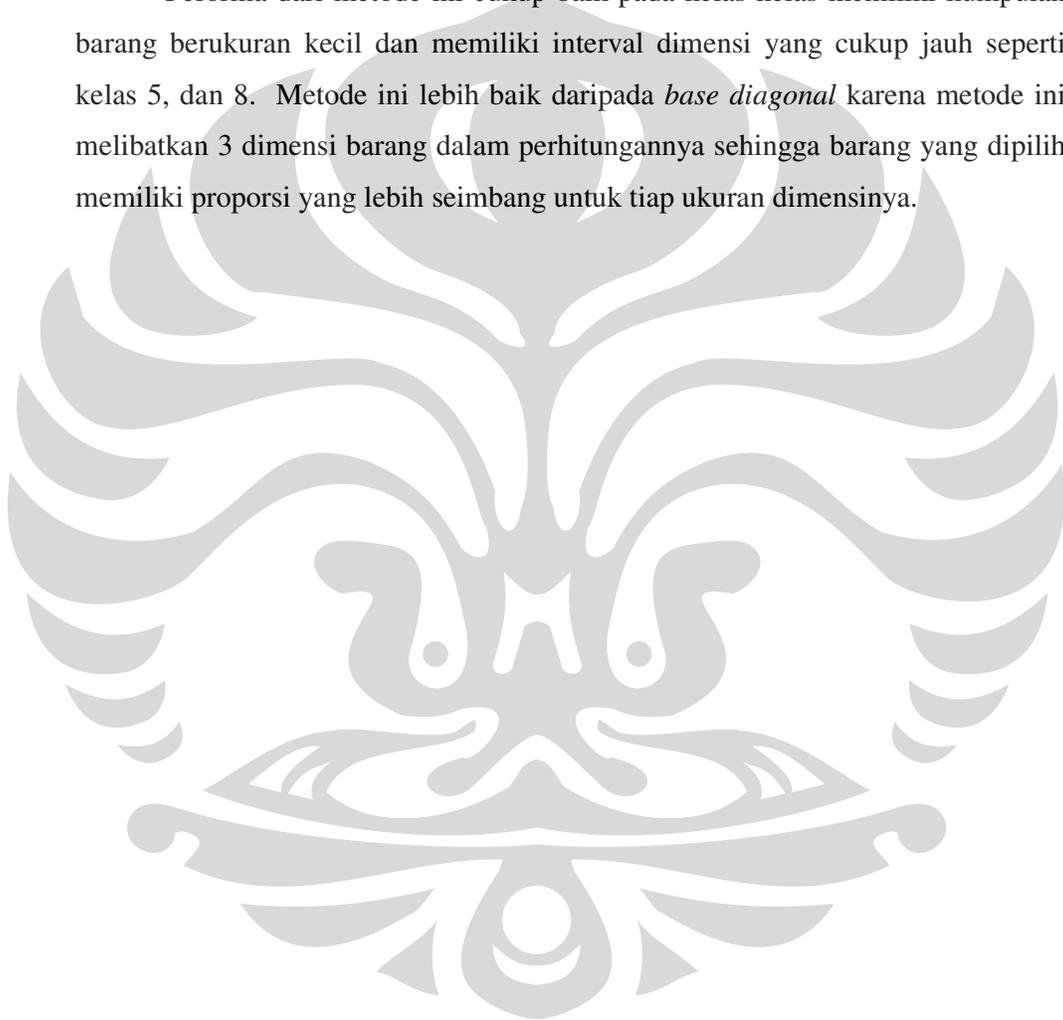
4.2.2.7 Peforma Metode *Base Diagonal*

Hasil yang cukup baik ditunjukkan oleh metode ini pada kelas 5 dan 7. Ini memperlihatkan bahwa metode ini memiliki peforma yang cukup baik pada barang-barang yang relatif kecil. Metode ini tidak begitu baik performanya untuk keseluruhan kelas karena metode ini lebih memilih barang dengan perbandingan antara ukuran *width* dan *length* yang besar. Ini membuat metode ini kurang efektif

karena proses muatnya lebih memilih barang yang proporsi alasnya kurang seimbang, sehingga metode ini sangat lemah terutama jika dihadapkan dengan barang-barang dengan perbandingan dimensi panjang dan lebar yang cukup jauh (misalnya barang tipe 3).

4.2.2.8 Peforma Metode *Diagonal*

Peforma dari metode ini cukup baik pada kelas-kelas memiliki kumpulan barang berukuran kecil dan memiliki interval dimensi yang cukup jauh seperti kelas 5, dan 8. Metode ini lebih baik daripada *base diagonal* karena metode ini melibatkan 3 dimensi barang dalam perhitungannya sehingga barang yang dipilih memiliki proporsi yang lebih seimbang untuk tiap ukuran dimensinya.



BAB 5 KESIMPULAN

5.1 Kesimpulan

1. Dalam proses muat manual menggunakan konsep EPBH secara umum (table 4.18), metode *sorting* berdasarkan *volume* memiliki performa yang paling baik karena membutuhkan *bin* paling sedikit pada hasil simulasi konsep EPBH.
2. Pada ukuran jumlah barang yang besar, metode *sorting* berdasarkan *volume* merupakan yang terbaik untuk tiap kelas barang.
3. Untuk ukuran jumlah sedang dan kecil, metode *sorting* berdasarkan luas area unggul dibanding metode-metode *sorting* lainnya.
4. Proporsi dari tiap dimensi barang berpengaruh kepada *utilisasi* ruangan *bin*.

5.2 Saran

1. Perlu adanya studi lanjutan mengenai bagaimana cara mengukur *volume* barang agar EPBH dapat secara efektif diimplementasikan dalam proses muat manual.
2. Untuk mengukur jumlah kebutuhan *material handling* pada proses muat manual pada keadaan sekarang, perlu adanya studi mengenai bagaimana melihat kecenderungan operator dalam melakukan proses *sorting* dalam suatu proses muat.

DAFTAR REFERENSI

- Berkey, J. O., P. Y. Wang. 1987. Two dimensional finite bin packing algorithms. *J. Oper. Res. Soc.* **38** 423–429.
- Chung, F. K. R., M. R. Garey, D. S. Johnson. 1982. On packing twodimensional bins. *SIAM—J. Algebraic Discrete Methods* **3** 66–76.
- Crainic, T. G., G. Perboli, R. Tadei. 2008. Ektreme point-based heuristic for three dimensional bin packing. *INFORMS J. Comput.* **20** 368-384.
- Den Boef, E., J. Korst, S. Martello, D. Pisinger, D. Vigo. 2005. Erratum to the “three-dimensional bin packing problem”: Robotpackable and orthogonal variants of packing problems. *Oper. Res.* **53** 735–736.
- George, J. A., D. F. Robinson. 1980. A heuristic for packing boxes into a container. *Comput. Oper. Res.* **7** 147–156.
- Boschetti, M. A. 2004. New lower bounds for the finite three-dimensional bin packing problem. *Discrete Appl. Math.* **140** 241–258.
- Faroe, O., D. Pisinger, M. Zachariassen. 2003. Guided local search for the three-dimensional bin packing problem. *INFORMS J. Comput.* **15** 267–283.
- Johnson, D. S. 1973. Near-optimal bin packing algorithms. Ph.D. thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge.
- Lim, A., B. Rodrigues, Y. Yang. 2005. 3-D container packing heuristics. *Applied Int.* **22** 125-134
- Lodi, A., S. Martello, D. Vigo. 1999a. Approximation algorithms for the oriented two-dimensional bin packing problem. *Eur. J. Opre. Res.* **112** 158–166.
- Lodi, A., S. Martello, D. Vigo. 2002b. Heuristic algorithms for the three-dimensional bin packing problem. *Eur. J. Oper. Res.* **141** 410–420.
- Lodi, A., S. Martello, D. Vigo. 2004. TSpack: A unified tabu search code for multi-dimensional bin packing problems. *Ann. Oper. Res.* **131** 203–213.
- Martello, S., P. Toth. 1990. *Knapsack Problems—Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, UK.
- Martello, S., D. Pisinger, D. Vigo, E. den Boef, J. Korst. 2007. Algorithms for general and robot-packable variants of the three-dimensional bin packing problem. *ACM Trans. Math. Software* **33** 7–19.
- Martello, S., D. Vigo. 1998. Exact solution of the finite two dimensional bin packing problem. *Oper. Res.* **44** 388-399.

Martello, S., D. Pisinger, D. Vigo. 2000. The three dimensional bin packing problem. *Oper. Res.* **48** 256-267.

Monaci, M., P. Toth. 2006. A set-covering based heuristic approach for bin-packing problems. *INFORMS J. Comput.* **18** 71-85.

Pisinger, D. 2002. Heuristics for the container loading problem. *Eur. J. Oper. Res.* **141** 382-392.



Kelas 1-4

	class 1			class 2			class 3			class 4		
	w	d	h	w	d	h	w	d	h	w	d	h
1	27	71	77	86	74	48	99	44	77	55	64	53
2	5	93	76	75	99	5	80	43	68	80	54	93
3	2	76	87	99	82	42	80	43	68	99	92	60
4	19	76	74	91	84	35	89	8	90	89	56	79
5	17	72	83	91	84	35	89	8	90	89	56	79
6	10	88	70	96	84	2	84	20	90	89	80	89
7	10	69	75	94	27	92	78	38	100	99	97	62
8	10	69	75	94	27	92	94	17	99	52	63	75
9	32	71	84	96	14	78	94	17	99	68	90	76
10	21	73	95	68	16	85	90	31	92	68	90	76
11	50	94	90	77	27	72	90	31	92	53	60	60
12	20	82	77	77	27	72	77	8	83	53	60	60
13	10	70	71	77	27	72	83	95	18	65	52	85
14	4	75	82	96	11	67	74	80	17	73	67	59
15	4	75	82	96	11	67	70	96	1	52	80	84
16	35	86	76	96	6	73	93	91	16	55	78	98
17	31	82	88	85	43	80	94	100	20	76	73	60
18	31	82	88	94	41	71	82	77	13	76	73	60
19	31	82	88	86	33	81	98	88	27	94	56	59
20	49	80	69	89	21	72	91	94	25	71	76	51
21	30	70	93	74	45	92	91	94	25	50	73	77
22	39	82	82	96	49	67	71	83	46	59	65	70
23	23	78	75	75	6	70	94	85	44	78	78	96
24	13	68	81	83	15	85	94	85	44	81	93	86
25	22	71	69	77	11	92	76	95	16	81	93	86
26	41	68	91	68	12	78	84	96	40	90	59	76
27	27	97	83	68	12	78	84	96	40	85	91	59
28	11	95	100	70	11	78	91	73	6	75	99	72
29	11	95	100	71	37	88	91	73	6	58	67	71
30	45	83	88	99	27	69	72	68	42	68	95	68
31	15	68	95	80	4	68	84	69	6	82	52	52
32	38	94	75	80	4	68	84	90	14	56	77	87
33	4	97	84	93	40	83	99	69	16	56	77	87
34	43	86	80	99	33	78	69	74	1	93	63	70
35	46	68	71	85	36	75	69	74	1	65	99	63
36	8	95	96	99	14	82	88	100	8	50	77	75
37	36	84	98	70	26	95	79	94	41	50	77	75
38	17	90	80	70	26	95	94	80	30	79	53	82
39	49	69	75	100	33	100	73	100	28	57	96	81
40	30	70	93	68	29	78	94	72	15	57	64	63
41	35	67	85	88	19	95	78	68	18	57	64	63
42	19	99	88	88	19	95	86	70	49	64	69	78
43	47	72	93	78	47	69	69	93	20	67	50	67
44	36	89	71	78	47	69	76	94	46	59	88	63
45	39	68	83	78	47	69	83	94	13	97	91	66
46	10	68	83	71	27	86	91	69	3	97	91	66
47	50	74	92	71	27	86	99	94	12	74	81	66
48	49	69	73	82	39	77	79	100	7	94	97	95
49	23	97	96	82	39	77	67	72	46	94	97	95
50	30	100	97	69	26	83	98	80	16	69	60	55
51	20	90	68	74	36	73	67	53	74	72	84	62
52	95	86	27	79	13	78	67	53	74	86	62	59
53	97	93	1	79	13	78	43	83	92	88	70	82
54	86	74	48	79	13	78	21	93	80	77	57	79
55	78	68	44	79	50	96	45	95	68	77	57	79
56	79	71	33	79	50	96	4	71	92	34	91	83
57	76	14	95	96	7	76	4	71	92	43	83	92
58	75	30	83	93	15	91	25	100	91	28	88	98
59	77	16	77	68	20	73	25	100	91	45	95	68
60	95	45	94	73	23	93	42	88	96	25	100	91

Lampiran : Data Generation (Lanjutan)

	class 1			class 2			class 3			class 4		
	w	d	h	w	d	h	w	d	h	w	d	h
61	99	26	80	73	23	93	31	89	99	48	72	69
62	93	15	72	27	71	77	38	100	93	37	68	98
63	86	22	80	39	82	82	38	100	93	17	87	82
64	68	36	77	35	67	85	33	99	89	18	73	85
65	80	43	68	29	84	99	39	80	85	50	99	85
66	89	8	90	5	98	83	44	71	74	10	86	94
67	90	31	92	27	77	92	29	85	91	22	80	70
68	73	5	95	2	75	79	10	82	82	22	80	70
69	77	8	83	8	70	86	18	73	85	70	11	78
70	79	71	37	78	38	100	10	86	94	71	37	88
71	100	86	5	72	47	70	2	97	83	93	40	83
72	68	96	36	77	8	83	2	97	83	79	4	91
73	67	89	33	85	83	12	32	69	82	99	33	78
74	71	71	43	69	90	9	70	11	78	69	37	90
75	95	83	37	82	77	13	100	26	99	99	14	82
76	84	85	24	86	99	2	100	33	69	100	33	100
77	72	99	25	67	96	5	82	39	77	79	50	96
78	89	80	89	72	68	42	79	13	78	90	5	74
79	87	70	69	89	80	89	79	50	96	96	20	91
80	71	72	78	71	58	68	79	50	96	73	23	93
81	62	84	83	52	63	75	90	5	74	99	77	30
82	100	78	87	81	93	62	68	20	73	87	78	1
83	86	81	90	94	84	51	71	58	68	87	78	1
84	83	91	84	94	84	51	80	92	53	78	74	39
85	19	28	33	94	84	51	97	62	100	97	90	24
86	18	19	28	52	80	84	92	59	73	69	76	25
87	13	37	35	50	73	77	53	55	93	78	68	18
88	13	37	35	81	93	86	88	84	97	69	93	20
89	40	38	20	90	59	76	76	73	60	94	79	16
90	43	4	19	100	64	59	94	56	59	83	94	13
91	18	8	25	19	28	33	50	21	27	94	69	32
92	6	38	35	50	21	27	13	37	35	79	100	7
93	37	44	22	27	50	7	43	4	19	69	77	10
94	27	48	46	18	19	28	18	37	22	96	68	10
95	12	41	32	4	37	16	35	34	23	6	32	21
96	45	29	27	18	37	22	22	25	22	37	2	3
97	16	11	15	41	10	20	47	42	13	27	26	12
98	27	37	12	9	16	30	3	50	46	24	33	9
99	33	5	26	9	16	30	5	28	24	24	33	9
100	40	9	18	9	16	30	19	32	38	40	20	50
101	34	75	72	99	82	42	80	43	68	55	64	53
102	27	71	98	91	84	35	95	15	70	89	80	89
103	28	96	82	96	78	29	94	17	99	80	92	53
104	5	93	76	96	14	78	74	80	17	50	95	89
105	5	93	76	70	16	89	70	96	1	65	52	85
106	39	82	82	70	16	89	80	89	37	78	78	96
107	30	94	84	88	38	89	80	89	37	96	70	53
108	26	89	68	84	38	78	80	89	37	85	91	59
109	26	89	68	71	37	88	93	91	16	85	91	59
110	26	89	68	71	37	88	73	90	3	56	50	85
111	29	84	99	71	37	88	72	68	42	56	50	85
112	18	67	100	99	27	69	72	68	42	96	94	94
113	8	99	78	94	14	81	99	69	16	96	94	94
114	8	99	78	94	14	81	67	82	28	76	70	65
115	5	98	83	92	12	69	81	86	7	79	53	82
116	12	93	82	94	10	82	81	86	7	79	85	84
117	34	96	90	85	36	75	87	78	1	93	51	92
118	27	77	92	70	26	95	91	83	45	59	88	63
119	8	70	86	80	33	70	69	93	20	96	74	61
120	12	73	100	77	48	86	90	75	29	72	84	62

Lampiran : Data Generation (Lanjutan)

121	34	96	97	79	13	78	83	94	13	70	97	89
122	36	92	99	78	68	44	98	80	16	83	80	74
123	36	92	99	90	94	44	82	44	73	89	76	80
124	48	76	71	79	71	33	84	20	90	64	87	50
125	48	76	71	94	27	92	84	20	90	69	62	61
126	48	76	71	89	8	81	73	90	3	85	89	87
127	1	89	87	89	8	81	71	98	26	85	89	87
128	16	85	85	96	11	67	71	98	26	80	70	56
129	50	89	87	72	6	70	87	89	39	53	55	93
130	24	100	91	89	8	74	94	72	15	80	59	71
131	5	88	99	70	11	78	91	83	45	55	78	98
132	50	94	74	99	33	78	91	83	45	75	99	72
133	50	94	74	86	19	68	91	83	45	58	67	71
134	50	94	74	87	5	92	69	93	20	80	98	69
135	5	87	100	69	37	90	94	79	16	56	77	87
136	43	83	92	82	35	70	83	94	13	50	90	60
137	19	72	67	89	8	70	71	80	31	69	85	94
138	45	95	68	82	43	81	75	83	7	88	100	68
139	4	71	92	88	93	4	68	36	77	88	100	68
140	18	82	79	96	78	29	73	5	95	86	62	59
141	25	100	91	96	14	78	89	89	23	64	87	50
142	41	74	72	96	11	67	80	72	16	91	82	69
143	41	74	72	74	7	82	93	91	16	94	84	51
144	9	90	69	94	41	71	98	88	27	53	60	60
145	9	90	69	88	38	89	98	88	27	52	80	84
146	7	81	96	84	38	78	71	83	46	71	76	51
147	23	92	68	83	21	79	94	85	44	75	99	72
148	48	83	92	83	15	85	94	85	44	95	81	75
149	31	89	99	83	15	85	71	98	26	82	52	52
150	38	100	93	83	15	85	73	100	28	99	84	73
151	17	87	82	80	4	68	73	100	28	99	79	69
152	39	80	85	70	26	95	78	74	39	79	53	82
153	29	85	91	70	26	95	69	76	25	57	64	63
154	10	82	82	88	19	95	82	91	18	79	85	84
155	22	80	70	79	13	83	86	70	49	72	84	62
156	99	82	42	69	36	94	69	93	20	5	93	76
157	68	16	85	89	8	70	76	94	46	34	96	90
158	77	27	72	74	36	73	90	75	29	2	75	79
159	77	27	72	82	43	81	91	76	25	36	93	90
160	72	48	70	5	88	99	78	75	13	36	92	99
161	72	48	70	42	96	69	87	99	24	16	85	85
162	85	39	73	47	70	71	42	69	88	50	89	87
163	96	49	67	47	70	71	5	93	76	31	97	87
164	96	49	67	2	74	96	30	94	84	70	11	78
165	75	6	70	28	88	98	18	67	100	80	4	68
166	83	15	85	5	92	68	5	98	83	80	4	68
167	68	12	78	19	72	67	5	98	83	93	40	83
168	69	74	1	19	72	67	12	93	82	94	14	81
169	73	100	28	18	82	79	48	76	71	92	12	69
170	69	76	25	18	82	79	31	97	87	94	10	82
171	85	74	25	41	74	72	67	48	80	99	33	78
172	76	94	46	10	82	82	93	40	83	69	37	90
173	79	100	7	10	86	94	92	12	69	100	33	69
174	69	77	10	82	44	73	100	26	99	100	33	69
175	80	98	21	70	96	1	99	33	78	82	35	70
176	87	99	24	86	78	37	82	35	70	80	33	70
177	51	64	79	98	88	27	87	39	95	77	48	86
178	99	92	60	73	90	3	73	23	93	69	21	77
179	67	53	74	76	95	16	55	64	53	87	39	95
180	87	70	69	84	69	6	89	80	89	87	39	95

Lampiran : Data Generation (Lanjutan)

	class 1			class 2			class 3			class 4		
	w	d	h	w	d	h	w	d	h	w	d	h
181	77	56	100	99	84	73	89	80	89	79	50	96
182	53	60	60	99	84	73	52	63	75	73	23	93
183	52	80	84	96	94	94	52	63	75	80	43	68
184	52	80	84	64	69	78	92	59	73	82	44	73
185	60	90	66	93	51	92	92	59	73	78	38	100
186	78	78	96	97	91	66	53	55	93	94	100	20
187	2	19	10	97	91	66	95	65	53	94	100	20
188	2	43	37	72	84	62	60	90	66	93	90	26
189	37	2	3	83	80	74	50	73	77	93	90	26
190	27	3	17	86	62	59	50	73	77	93	90	26
191	24	33	9	19	28	33	83	84	77	86	78	37
192	43	10	11	4	13	4	59	65	70	98	88	27
193	34	38	29	27	50	7	28	36	16	84	69	6
194	17	42	9	13	37	35	19	13	25	28	36	16
195	18	19	21	33	34	44	47	28	26	21	47	4
196	34	3	5	36	26	31	34	38	29	19	41	24
197	35	22	8	29	12	43	2	11	37	19	41	24
198	25	39	41	50	12	49	1	36	40	2	43	37
199	36	17	35	42	19	47	40	20	50	28	22	1
200	9	6	48	19	32	38	46	47	39	36	17	35



Lampiran : Data Generation (Lanjutan)

Kelas 5-8

	class 5			class 6			class 7			class 8		
	w	d	h	w	d	h	w	d	h	w	d	h
1	19	28	33	9	4	1	12	15	20	29	19	17
2	29	4	45	7	1	6	29	22	29	43	67	4
3	31	41	9	7	2	3	6	30	17	92	33	24
4	50	21	27	7	2	1	22	12	10	100	77	64
5	27	50	7	2	3	9	7	21	26	4	76	26
6	27	50	7	5	1	10	20	20	34	38	37	83
7	37	34	12	2	4	5	35	20	15	55	2	88
8	37	34	12	9	1	9	30	14	13	71	41	40
9	37	34	12	5	4	9	20	33	16	59	22	16
10	12	12	25	7	2	1	19	7	16	46	32	28
11	18	19	28	9	6	10	2	33	7	83	57	32
12	13	37	35	4	10	5	22	19	31	76	26	86
13	6	47	41	2	1	7	14	11	35	36	22	9
14	43	4	19	3	7	1	21	35	19	23	7	88
15	43	4	19	10	5	3	31	32	4	88	16	56
16	43	4	19	6	5	8	21	32	23	99	45	1
17	33	34	44	10	3	7	31	29	31	16	9	18
18	40	17	50	5	6	2	9	27	5	39	97	82
19	4	37	16	6	10	7	19	20	19	44	52	75
20	30	17	14	10	4	1	14	33	17	55	98	84
21	41	10	20	6	4	6	4	32	11	32	80	62
22	20	22	38	1	6	4	25	10	23	20	84	27
23	48	7	27	10	5	8	21	2	14	92	46	63
24	29	16	48	1	10	6	35	6	15	6	3	91
25	10	5	12	1	4	6	14	19	34	1	83	55
26	29	12	43	1	9	7	5	2	16	63	23	94
27	29	12	43	8	6	1	19	17	11	29	77	23
28	44	27	29	5	8	1	16	11	27	34	61	69
29	48	24	27	6	7	5	6	33	35	57	46	51
30	20	38	19	2	9	8	26	35	35	52	42	83
31	3	50	46	10	9	2	27	35	22	32	53	88
32	5	28	24	8	10	1	23	6	30	70	18	5
33	30	19	38	7	3	3	12	9	10	62	19	45
34	19	32	38	2	8	9	32	32	7	97	48	83
35	9	49	33	6	3	1	7	35	30	82	60	23
36	9	49	33	1	5	7	23	31	23	71	5	29
37	50	37	43	5	3	8	8	15	32	72	85	48
38	50	37	43	8	3	10	20	23	27	94	64	66
39	25	14	9	9	1	10	9	17	19	25	22	63
40	25	14	9	3	3	2	33	12	14	3	40	50
41	2	19	10	3	10	3	15	9	11	97	3	46
42	48	25	32	7	3	8	26	13	1	63	56	86
43	23	8	39	3	1	3	20	25	31	97	55	27
44	27	24	11	3	6	2	30	33	9	82	50	89
45	27	24	11	10	5	8	16	31	16	1	42	67
46	27	24	11	6	5	1	15	12	29	59	32	91
47	27	24	11	4	4	5	2	20	28	54	73	8
48	11	31	24	8	1	10	32	34	34	78	23	38
49	37	2	3	6	9	7	17	5	27	44	52	26
50	37	2	3	7	4	6	35	26	23	66	18	96
51	17	26	22	6	3	8	33	35	27	47	43	36
52	16	4	47	8	3	5	20	15	12	4	43	61
53	19	13	25	1	1	7	16	1	28	67	92	48
54	26	34	5	2	4	1	18	3	25	66	87	1
55	43	10	11	2	3	9	22	18	1	99	4	21
56	43	10	11	10	2	6	20	17	7	1	90	28
57	17	24	15	2	10	6	4	8	31	22	84	70
58	27	50	27	2	10	9	6	12	10	100	30	68
59	2	39	35	5	2	5	24	4	11	19	41	82
60	2	39	35	10	8	3	6	11	7	36	50	81

Lampiran : Data Generation (Lanjutan)

	class 5 WDH=100x100x100			class 6 WDH=10x10x10			class 7 WDH=40x40x40			class 8 WDH=100x100x100		
	w	d	h	w	d	h	w	d	h	w	d	h
61	1	36	40	5	3	6	5	14	17	55	65	61
62	40	20	50	1	10	10	2	29	3	32	47	46
63	25	39	41	10	3	3	35	12	20	19	61	6
64	36	17	35	3	6	3	15	32	7	15	66	11
65	46	47	39	5	7	8	25	9	18	8	24	63
66	30	94	84	10	7	9	3	1	32	4	29	31
67	27	77	92	1	10	2	20	24	29	80	56	66
68	36	92	99	1	1	10	10	15	5	17	42	31
69	1	89	87	7	1	5	33	15	4	92	10	99
70	50	89	87	7	7	7	26	8	10	24	47	36
71	6	89	91	9	5	8	33	30	20	55	56	22
72	31	97	87	9	7	3	25	21	3	98	92	27
73	12	93	84	2	8	7	29	21	27	82	96	40
74	95	86	27	10	3	2	33	32	20	74	14	80
75	95	86	27	6	8	10	24	8	12	98	11	13
76	96	84	2	3	3	1	9	6	16	89	86	71
77	96	14	78	4	8	9	6	27	2	86	68	17
78	68	16	85	9	3	5	11	12	2	44	79	5
79	84	38	78	8	3	6	19	27	13	44	86	35
80	84	38	78	5	1	2	11	5	33	83	64	18
81	94	13	90	3	8	2	33	14	12	56	1	74
82	69	74	1	5	9	3	15	29	28	12	19	16
83	99	77	30	9	3	10	24	18	27	5	32	4
84	78	82	3	1	3	10	4	27	34	88	24	75
85	96	77	44	2	7	1	32	8	29	23	95	72
86	92	99	38	6	4	2	29	3	3	19	88	24
87	90	75	29	8	7	10	1	1	23	7	51	1
88	73	91	23	4	2	2	22	5	21	36	82	90
89	98	80	16	4	4	7	16	1	32	57	19	31
90	75	99	72	6	3	7	24	8	23	68	74	80
91	96	94	94	3	5	7	24	33	9	88	73	66
92	50	77	75	5	8	8	22	6	26	74	98	13
93	57	64	63	5	5	3	14	12	31	10	81	77
94	79	85	84	9	10	3	2	13	17	52	68	78
95	97	93	73	4	3	8	16	34	13	34	85	68
96	50	90	60	4	5	1	17	33	22	25	27	15
97	72	84	62	10	6	2	4	28	1	81	90	93
98	72	84	62	3	3	3	21	2	21	70	52	67
99	89	76	80	3	5	5	9	25	23	25	51	28
100	77	57	79	10	6	2	17	9	10	58	15	57
101	46	47	39	3	8	2	1	8	12	38	61	59
102	19	28	33	4	3	8	28	10	31	49	40	3
103	4	13	4	3	10	9	27	16	8	28	75	100
104	4	13	4	6	3	6	14	9	19	62	45	7
105	49	28	41	3	2	1	28	31	22	40	74	15
106	37	34	12	1	10	2	16	27	34	93	57	59
107	18	19	28	9	4	3	24	18	31	82	31	61
108	43	4	19	2	3	2	28	7	4	1	65	74
109	33	34	44	5	1	1	30	35	34	72	67	10
110	20	22	38	7	4	9	4	4	14	78	15	5
111	47	17	4	3	6	6	23	6	20	44	20	87
112	3	18	28	8	1	3	7	29	20	67	57	63
113	46	34	35	9	4	2	27	32	5	74	34	55
114	10	5	12	2	10	9	11	15	32	86	42	96
115	42	50	31	1	6	4	13	30	33	61	82	8
116	42	50	31	1	7	7	35	33	15	40	3	78
117	5	8	44	4	10	2	4	23	6	42	58	52
118	5	8	44	10	5	1	24	33	16	3	47	24
119	10	43	17	5	10	10	15	11	7	19	47	52
120	12	2	28	3	4	1	33	6	8	44	68	26

Lampiran : Data Generation (Lanjutan)

	class 5 WDH=100x100x100			class 6 WDH=10x10x10			class 7 WDH=40x40x40			class 8 WDH=100x100x100		
	w	d	h	w	d	h	w	d	h	w	d	h
121	22	25	22	5	7	2	9	33	10	61	87	99
122	47	42	13	1	2	3	17	25	21	27	68	34
123	44	27	29	1	3	3	33	25	23	11	67	44
124	44	27	29	7	7	7	8	1	16	83	94	37
125	44	27	29	2	5	10	25	14	30	18	68	62
126	24	50	34	7	4	2	30	18	16	70	88	29
127	24	50	34	1	6	8	14	14	1	31	21	98
128	48	24	27	4	2	7	30	28	25	69	88	19
129	6	32	21	3	9	7	16	11	27	53	80	96
130	23	39	10	1	4	9	19	3	25	60	85	26
131	19	41	24	10	9	3	20	4	21	7	40	30
132	2	19	10	10	3	7	20	10	10	23	18	12
133	3	30	12	8	8	9	15	33	21	55	57	57
134	48	25	32	3	1	6	8	21	24	32	80	60
135	28	14	7	5	8	5	27	15	18	9	20	20
136	27	24	11	4	10	1	17	11	28	72	68	70
137	27	24	11	3	2	3	6	21	27	49	44	36
138	11	31	24	10	8	1	19	33	12	78	83	27
139	7	25	35	5	7	10	32	1	19	11	37	76
140	3	43	23	1	7	4	29	6	34	29	99	58
141	43	37	32	3	2	3	13	17	10	89	38	100
142	19	13	25	10	5	8	17	15	14	77	27	100
143	27	3	17	7	10	2	15	17	1	45	30	21
144	27	26	12	10	1	6	10	25	34	79	6	16
145	27	26	12	4	4	6	33	20	2	80	19	45
146	27	26	12	9	7	9	31	13	13	96	9	83
147	32	49	18	10	9	1	22	14	16	62	40	98
148	32	49	18	3	5	2	29	3	17	25	20	86
149	43	10	11	6	3	2	1	30	19	11	66	40
150	17	24	15	5	8	2	29	32	14	49	95	54
151	17	24	15	3	10	10	19	23	31	7	22	82
152	34	38	29	4	7	6	29	26	4	10	79	24
153	18	19	21	9	2	10	24	23	9	30	89	38
154	28	22	1	3	2	6	19	16	3	26	77	62
155	34	3	5	5	10	3	12	12	35	84	92	11
156	34	3	5	10	9	3	35	23	3	20	49	95
157	1	36	40	9	1	2	10	12	8	86	92	65
158	40	20	50	6	6	7	14	35	3	48	41	22
159	25	39	41	5	4	6	4	15	32	22	16	82
160	36	17	35	10	3	1	18	21	32	85	32	98
161	40	2	3	2	9	2	2	12	15	23	68	56
162	9	6	48	3	5	1	26	13	32	77	29	68
163	50	94	74	6	4	2	15	23	8	71	20	57
164	27	93	90	5	3	1	33	33	18	44	56	74
165	19	72	67	6	9	9	6	26	18	99	69	25
166	4	71	92	9	6	7	30	13	33	60	40	25
167	43	88	87	2	9	8	30	34	11	11	47	53
168	9	90	69	7	7	4	31	22	21	54	26	57
169	23	92	68	3	8	9	12	34	35	84	92	16
170	2	82	92	3	1	9	35	4	10	5	22	23
171	39	80	85	5	1	6	16	18	8	64	27	26
172	10	82	82	2	1	9	4	13	28	77	86	100
173	10	82	82	6	6	6	17	19	33	9	35	80
174	9	69	93	1	9	6	23	1	5	27	35	61
175	2	97	83	5	1	9	10	16	22	14	89	95
176	95	86	27	1	4	3	9	32	24	28	27	85
177	75	99	5	3	7	2	19	25	12	62	67	54
178	78	68	44	7	5	10	22	12	22	56	28	35
179	96	78	29	4	9	5	11	29	33	59	93	16
180	89	8	81	1	6	8	12	6	9	62	52	73