



# Implementasi 3D Bin Packing Problem Menggunakan Algoritma Tabu Search

Edhy Poerwandono<sup>1\*</sup>, Sultan Faqih Fiddin<sup>2</sup>

<sup>1,2</sup>Sistem Informasi, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, Jakarta Timur, Indonesia

Email Penulis Korespondensi: <sup>1</sup>eddypurwandana@gmail.com

**Abstrak**—Ruang penyimpanan merupakan salah satu tantangan dalam industri logistik, terutama dalam pengemasan dan pengiriman barang pada kontainer. Perusahaan logistik umumnya membutuhkan solusi yang efektif untuk mengoptimalkan ruang yang tersedia pada kontainer. Terdapat banyak metode dan pendekatan untuk masalah ini, salah satunya menggunakan algoritma tabu search yang dapat digunakan untuk memecahkan masalah ini dan menggunakan metode penelitian kuantitatif. Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma tabu search dan menampilkan hasilnya yang akan menggunakan gambar 3 dimensi. Hasil dari penelitian ini menunjukkan bahwa metode tabu search dapat mengoptimalkan penggunaan ruang penyimpanan dan dengan menampilkan gambar 3 dimensi, penerapan di lapangan akan sangat mudah. Dari hasil pengujian, algoritma tabu search menunjukkan kinerja yang lebih baik dalam hal optimasi ruang penyimpanan dan efisiensi biaya. Oleh karena itu, perusahaan logistik kargo dapat mempertimbangkan penggunaan metode tabu search untuk meningkatkan efisiensi dan produktivitas memuat barang ke dalam kontainer..

**Kata Kunci:** Tabu Search, ThreeJS, 3D, Bin packing problem, Container loading

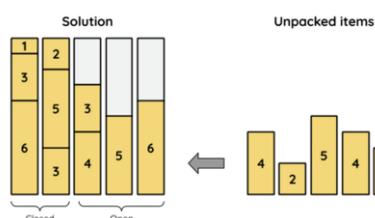
**Abstract**—Storage space is one of the challenges in the logistics industry, especially in packaging and shipping goods in containers. Logistics companies generally require effective solutions to optimize the available space in containers. There are many methods and approaches to this problem, one of which is using the tabu search algorithm which can be used to solve this problem. The purpose of this research is to implement the tabu search method and present its results using 3-dimensional images. The results of this study show that the tabu search method can optimize the use of storage space and by presenting 3- dimensional images, the implementation in the field will be very easy. From the test results, the tabu search and method showed better performance in terms of storage space optimization and cost efficiency. Therefore, cargo logistics companies can consider using the tabu search method to improve efficiency and productivity in loading goods into containers.

**Keywords:** Tabu Search, ThreeJS, 3D, Bin packing problem, Container loading

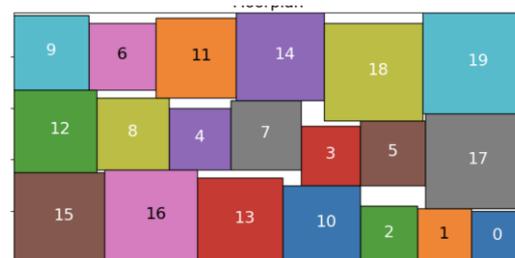
## I. PENDAHULUAN

Menyusun bin adalah sebuah masalah yang sering terjadi dalam berbagai situasi di industri, salah satu tujuannya adalah untuk meminimalkan biaya transportasi dan penyimpanan dengan cara memadatkan [1] sebanyak mungkin wadah ke tempat penyimpanan [2]. Dalam industri pengiriman barang dan transportasi, berbagai kontainer dengan ukuran berbeda digunakan. Peti kemas standar dengan panjang 20 atau 40 kaki digunakan untuk transportasi laut, sementara berbagai jenis peti kemas digunakan untuk transportasi darat atau udara [3] dan ada 3 ukuran kontainer yang telah memenuhi persyaratan teknis International Organization for Standardization (ISO) yaitu panjang 20 kaki, 40 kaki dan 45 kaki.

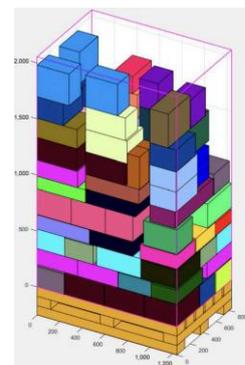
Masalah ini telah banyak diteliti oleh para ahli dan ada banyak pendekatan [1] yang dilakukan agar dapat mengeluarkan hasil dari perhitungan formulasi berdasarkan algoritma tertentu yang mana outputnya terbagi menjadi 3 tipe, yaitu gambar 1 dimensional, 2 dimensional dan 3 dimensional.



Gambar 1. Gambar 1 Dimensi



Gambar 2. Gambar 2 Dimensi



Gambar 3. Gambar 3 Dimensi



Menentukan algoritma yang tepat dalam kasus ini yang harus dilakukan yaitu menganalisa data bin yang akan dimasukkan ke kontainer, karena tidak ada algoritma yang telah dipublikasikan yang dapat memberikan solusi mutlak untuk 3D bin packing problem [4]. Wäscher et al. (2007) mengusulkan bahwa masalah pemotongan dan pengepakan dapat diklasifikasikan berdasarkan beberapa faktor seperti jumlah dimensi yang terlibat, jenis barang yang akan dikemas (besar atau kecil), dan tujuan optimasi [5].

Untuk mengatasi masalah tersebut, salah satu solusi yang dapat digunakan adalah menggunakan algoritma Tabu Search luarannya berupa gambar 3D, dengan menampilkan gambar 3D dapat terlihat jelas tata letak dan penempatan seluruh barang yang akan dimasukkan ke kontainer, dengan tujuan dapat meminimalisir kontainer yang digunakan dan efisiensi penggunaan ruang pada kontainer.

## II. METODE PENELITIAN

Metodologi penelitian adalah suatu cara atau teknik untuk mendapatkan informasi dan sumber data yang akan digunakan dalam penelitian. Informasi atau data ini bisa dalam bentuk apa saja seperti, jurnal, artikel, tesis, buku, koran dan sebagainya.

Metode Kualitatif merupakan metode pendekatan yang digunakan pada penelitian ini yang digunakan untuk memahami dan menjelaskan secara mendalam.

Proses penelitian kualitatif melibatkan pengumpulan data melalui metode seperti wawancara, observasi, studi arsip, atau analisis dokumen. Data yang dikumpulkan kemudian dianalisis melalui proses seperti pemilahan, pengelompokan, dan pencarian tema atau pola yang muncul.

### 1. Identifikasi Masalah

Masalah ini memiliki beberapa faktor penting yang harus diperhatikan dalam proses mengatur dan menata barang di dalam kontainer, berikut adalah poin-poin penting yang teridentifikasi sebagai masalah pada penelitian ini:

#### A. Pemanfaatan Ruang

Tantangan pertama dan yang paling utama adalah memaksimalkan pemanfaatan ruang yang terbatas dalam kontainer. Masalah ini melibatkan pemilihan dan penataan yang efisien dari berbagai jenis barang dengan berbagai ukuran, bentuk, dan berat agar dapat mengisi setiap ruang kosong dengan optimal. Tujuan utamanya adalah untuk mengurangi ruang yang tidak terpakai dan meminimalkan jumlah kontainer yang digunakan.

#### B. Optimalisasi Waktu dan Biaya

Efisiensi waktu dan biaya adalah faktor penting dalam proses muat barang ke dalam kontainer. Masalah ini melibatkan penentuan urutan pengisian kontainer, penataan barang dengan cara yang paling efisien, dan pengelolaan aliran barang yang efektif

agar dapat mengurangi waktu dan biaya yang terkait dengan proses muat barang.

### C. Barang Tidak Umum

Pada saat observasi, peneliti melihat langsung bahwa barang-barang yang akan dikirim memiliki ukuran, bentuk atau karakteristik khusus yang mempengaruhi cara penataan. Masalah ini melibatkan penyesuaian strategi penataan untuk memastikan barang khusus dapat dimuat dengan aman dan efisien

### 2. Pengumpulan Data

Pengumpulan data adalah metode yang dipakai untuk mengumpulkan data-data penelitian. Artinya, dalam menulis maupun membuat karya ilmiah, penulis harus menentukan pengumpulan data yang sesuai dan tepat. Pengumpulan data dilakukan dengan cara: Observasi, Studi Literatur dan Wawancara.

### 3. Pengolahan Data

Langkah pertama dalam mempersiapkan data adalah mengatasi dan membersihkan data yang nilainya hilang atau kosong atau yang biasa disebut *missing data*. *Missing data* yang dimaksud tidak sama dengan data yang bernilai 0 atau yang lainnya [6], melainkan nilainya yang hilang atau kosong yang bisa disebut juga *null* jika dalam *database*. Dalam kasus ini *missing data* akan dihapus agar memudahkan penggunaan data di proses selanjutnya, namun dalam beberapa kasus lainnya *missing data* dapat diabaikan atau dapat diganti dengan nilai lainnya. *Filter* data merupakan bagian dari persiapan data yang mana proses ini juga diperlukan agar data yang digunakan sebagai objek penelitian adalah data yang akurat. Dalam penelitian ini *filter* data yang dimaksud adalah menyeleksi data yang duplikat atau data yang sama dengan data lainnya [6], data duplikat juga dapat mengganggu akurasi data, sehingga diperlukan proses untuk menghapus data yang duplikat.

### 4. Pendekatan Algoritma

Implementasi metodologi untuk menyelesaikan masalah penataan barang di kontainer, para ahli sebelumnya mendefinisikan dua kategori solusi: *placement & improvement heuristics and the exact methods* [7].

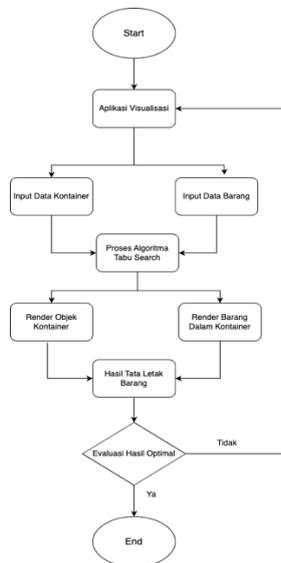
Tabu search merupakan bagian dari *improvement heuristics* yang dapat menyelesaikan 3D *packing problem* karena tabu search salah satu *metaheuristics* yang sangat populer [8] dalam hal *combinatorial optimization* dan populer juga untuk digunakan pada masalah pemuatan barang di kontainer. Teknik tabu



search meningkatkan kinerja metode ini dengan mengizinkan pelanggaran aturan dasar [7]. Pada setiap langkah, kita diizinkan untuk memilih solusi yang kurang baik jika tidak ada peningkatan yang dapat dicapai, dengan demikian peneliti akan menggunakan tabu search [5] sebagai pendekatan algoritma yang akan diterapkan pada perancangan aplikasi pemuatan barang dalam kontainer.

5. Rancangan Pengujian

Rancangan pengujian dibentuk bertujuan untuk bisa memvalidasi kualitas dan efektivitas aplikasi 3D pemuatan kontainer yang menggunakan algoritma Tabu Search. Dengan melibatkan variasi kasus uji dan melakukan perbandingan dengan solusi optimal atau metode lain, pengujian ini dapat memberikan informasi yang berguna tentang kemampuan aplikasi dalam menghasilkan tata letak objek yang optimal di dalam kontainer.



Gambar 5. Rancangan Pengujian

III. HASIL DAN PEMBAHASAN

Untuk menghasilkan *output* yang maksimal pada penelitian ini, maka akan diimplementasikan visualisasi 3D pada aplikasi berbasis web yang akan dibangun menggunakan NodeJS, JavaScript, jQuery, HTML5 dan CSS.

Membangun aplikasi web tersebut akan menjadi 2 bagian utama yaitu bagian *backend* dan *frontend* yang akan fokus kepada tugasnya masing-masing. Ini memungkinkan pengembang untuk fokus pada aspek yang sesuai dengan keahlian mereka, sehingga menghasilkan aplikasi yang efisien, skalabel, dan memiliki antarmuka yang menarik bagi pengguna.

HTML, CSS, jQuery dalam hal ini akan bertugas sebagai *frontend* yang mana bertujuan untuk menampilkan *user interface* berupa *form input*, objek 3D kontainer, objek 3D kotak barang dalam kontainer. Menampilkan objek 3D

pada aplikasi web membutuhkan *library* tambahan yaitu ThreeJS dan GASP.

NodeJS mengambil peran sebagai *backend* yang berfungsi untuk mengelola proses algoritma Tabu Search dan beberapa proses lainnya. Node.js umumnya digunakan sebagai *backend* dari aplikasi web *full stack* karena para pengembang dapat menggunakan bahasa yang sama (JavaScript/TypeScript) baik untuk bagian *backend* maupun *frontend*. Selain itu, NodeJS memungkinkan permintaan ke server yang dikelolanya bersifat non-blocking [14].

Agar *Backend* dan *Frontend* dapat berkomunikasi data, pada hal ini peneliti menggunakan Web Worker untuk saling melempar data, sehingga komunikasi datanya lebih mudah.

Web Worker kerap digunakan untuk berkomunikasi data karena teknologi ini salah satu teknologi yang umumnya ada pada browser, dan Web Worker ini akan berjalan di *background* yang mana tidak akan mengganggu yang berjalan di halaman browser.

3.1 Pembuatan Web Visualisasi

3.1.1 Pondasi Layout

Membangun web visualisasi 3d akan dimulai dengan bagian front end terlebih dahulu. *Front end* berfokus untuk menampilkan visual tampilan yang akan dilihat oleh pengguna dibangun menggunakan HTML5, CSS dan ThreeJS yang didalamnya hanya akan ada satu halaman yang akan mencakup semua kebutuhan visualisasi 3d.

Fondasi *layout* dibangun dengan HTML5 yang dapat menampilkan visual dan menampung semua komponen ThreeJS dan form. Untuk *me-render* ThreeJS maka diperlukan sebuah komponen HTML5 div dengan ditambahkan atribut *class* dan *id* yang unik agar dapat diinisiasi oleh ThreeJS melalui *id* unik tersebut, berikut kodenya:

```
<div class="threeD-container" id="3D-container"></div>
```

Selanjutnya ThreeJS akan dipanggil dan diinisiasi melalui file baru dengan nama *main.js* yang berisikan kode:

```
import * as THREE from 'three';
```

```
const ThreeDContainer = document.getElementById("3D-container");
```

3.1.2 Membuat Scene

ThreeJS memerlukan 3 elemen penting agar dapat menampilkan konten objek 3d, yaitu: *scene*, *camera* dan (*three.js*, 2023). *Camera* yang digunakan merupakan *camera PerspectiveCamera*, memiliki 4 atribut utama yaitu:

- 1) *Field of View* merupakan sejauh mana adegan yang dapat terlihat pada layar pada setiap momen tertentu. Nilainya diukur dalam derajat.
- 2) Aspek Rasio umumnya akan diisi dengan nilai  $width\ screen / height\ screen$



- 3) *Near* dan *far clipping plane* merupakan objek yang berada lebih jauh dari kamera daripada nilai '*far*' atau lebih dekat dari '*near*' tidak akan dirender

Setelah camera maka langkah selanjutnya adalah *renderer*, agar setup scene dan camera dapat ditampilkan pada halaman web dengan menambahkan konfigurasi *setPixelRatio* dan *setSize* yang berfungsi untuk menandakan width dan height pada area mana yang akan dirender, dalam kasus ini akan menggunakan parameter *devicePixelRatio*, *clientWidth*, *clientHeight* berdasarkan ukuran jendela browser (three.js, 2023).

```
const scene = new THREE.Scene();
```

```
camera = new THREE.PerspectiveCamera(75,
window.innerWidth / window.innerHeight, 1, 10000);
```

```
renderer = new THREE.WebGLRenderer({antialias:
true, alpha: true});
```

```
renderer.setPixelRatio(window.devicePixelRatio);
```

```
renderer.setSize(ThreeDContainer.clientWidth,
ThreeDContainer.clientHeight);
```

```
renderer.render(scene, camera);
```

### 3.1.3 Membuat Form

Pada web ini terdapat 2 form yang menjadi bagian penting dalam implementasi 3d *bin packing problem*. Nilai dari 2 *form* tersebut akan dikirim ke *backend* untuk diproses oleh algoritma tabu search melalui web worker pada saat tombol *generate* ditekan yang ada pada halaman web.

- 1) Form ukuran kontainer yang dapat dipilih oleh pengguna dengan 3 opsi, yaitu: 20 kaki, 40 kaki, 45 kaki
- 2) Form *input* barang yang terdiri dari *label* barang, *length* barang, *width* barang, *height* barang, *weight* barang dan *quantity* barang.

Gambar 6. Pilihan ukuran kontainer

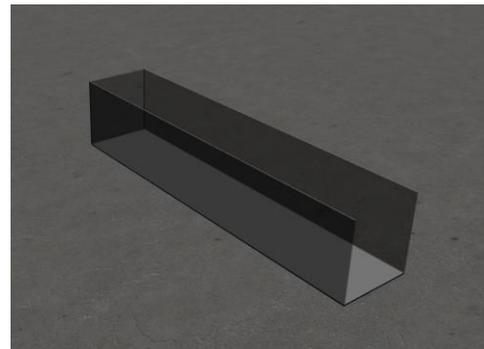
Gambar 7. Form *input* barang

### 3.1.4 Loading 3D Kontainer

Pada saat ukuran telah dipilih maka pengguna diharuskan untuk klik tombol *select / update*, hal itu dilakukan agar ThreeJS dapat menghasilkan 3d objek. Objek yang dihasilkan berupa 3d kontainer yang digunakan sebagai dasar utama agar data barang yang diinput dapat diletakkan pada 3d objek kontainer tersebut.

ThreeJS dapat membuat 3d kontainer menggunakan beberapa fungsi:

- 1) *Group*, berfungsi agar dapat menggabungkan beberapa objek sekaligus
- 2) *BoxGeometry* merupakan suatu representasi geometris dari sebuah kubus dengan ukuran 'lebar', 'tinggi', dan 'kedalaman' tertentu. Ketika dibuat, kubus tersebut berada di tengah-tengah koordinat, dengan setiap sisinya sejajar dengan salah satu sumbu koordinat
- 3) *MeshLambertMaterial* merupakan material menggunakan model Lambertian non-fisik untuk menghitung reflektansi. Ini dapat dengan baik mensimulasikan beberapa permukaan
- 4) *Mesh*, merupakan fungsi untuk menggambarkan objek berbasis mesh poligon segitiga.



Gambar 8. 3D Objek Kontainer

### 3.1.5 Koneksi ke Backend

*Backend* dapat memperoleh data yang akan diproses melalui Web Worker, yang mana web worker ini merupakan fitur yang diperkenalkan dari HTML5. Web Worker digunakan dalam penelitian ini karena implementasinya mudah dilakukan dan fundamentalnya dapat dipahami oleh pemula. Data yang dikirimkan ke backend melalui web worker yaitu: *label* barang, *length* barang, *width* barang, *height* barang, *weight* barang dan *quantity* barang.

Pada penelitian ini web worker berfungsi untuk mengirimkan data dimensi barang yang diperoleh melalui elemen *form* yang telah dibuat pada proses 3.1.3. Langkah yang dilakukan menggunakan web worker yaitu:

- 1) Buat fungsi *onmessage* pada *file external javascript backend* yang akan dipanggil oleh *client* (HTML5) dengan menuliskan kode di dalam fungsi tersebut disesuaikan dengan kebutuhan dan pada akhir fungsi tersebut tambahkan kode fungsi *postMessage* diikuti dengan parameter yang akan dikembalikan ke *client*



- 2) Inisiasi web worker pada sisi *client* (HTML5) dan memanggil *file external javascript* yang dibuat pada Langkah sebelumnya dengan menuliskan kode:

```
const worker = new Worker('worker.js', { type:
    "module" });
```

- 3) Setelah inisiasi web worker telah siap untuk mengirimkan data kepada *backend* dengan menggunakan kode *worker.postMessage* diikuti dengan parameter data yang akan dikirim ke *backend*
- 4) Siapkan fungsi pada *client* untuk menerima umpan balik data dari *backend* dengan menuliskan fungsi *worker.onmessage*, yang didalamnya dapat ditulis dengan berbagai tujuan dan kebutuhan setelah menerima umpan balik data.

### 3.1.6 Proses Algoritma Tabu Search

Tahap ini merupakan tahap yang paling penting pada penelitian ini, yang mana dalam proses ini menentukan skema tata letak barang dalam kontainer dan menentukan posisi terbaik untuk masing-masing barangnya.

Implementasi dari algoritma Tabu Search untuk mencari koordinat terbaik dalam menyelesaikan permasalahan *bin packing problem* pada sebuah 3D objek kontainer. Algoritma ini bertujuan untuk mencari solusi optimal dengan meminimalkan ruang yang tidak terpakai dalam objek 3D.

Proses Tabu Search dilakukan dengan mencoba menempatkan setiap barang pada setiap iterasi dengan mengacak urutan barang. Selanjutnya program akan mencoba menempatkan barang-barang tersebut pada setiap posisi yang memungkinkan dalam kontainer 3D, dan mencatat posisi terbaik yang menghasilkan skor (volume yang digunakan dalam kontainer) terkecil.

Berikut adalah langkah – langkah yang dilakukan oleh algoritma tabu search sehingga mendapatkan koordinat terbaik pada setiap barangnya.

- 1) Tabu search menerima data tentang dimensi dan jumlah barang yang akan diatur di dalam objek kontainer 3D, serta batasan-batasan yang harus dipenuhi seperti berat objek dan dimensi setiap barang
- 2) Inisialisasi langkah pertama dengan membuat posisi awal secara acak pada setiap barangnya dalam objek 3D yang berupa koordinat x, y dan z [8] [9].
- 3) Sebelum menentukan barang pada posisi tertentu, program akan memeriksa apakah barang tersebut *overlap* satu sama lain. Hal ini dilakukan agar setiap barang yang telah ditempatkan pada kontainer untuk memastikan tidak *overlap* [10].
- 4) Buat suatu tabu *list* untuk menyimpan langkah-langkah yang harus dihindari dalam iterasi berikutnya agar menghindari *stuck* di solusi lokal [8].
- 5) Lakukan iterasi pencarian dengan mengulang langkah-langkah berikut:
  - a. Evaluasi solusi saat ini dengan menghitung jumlah ruang yang tidak terpakai dalam objek kontainer 3D [9].

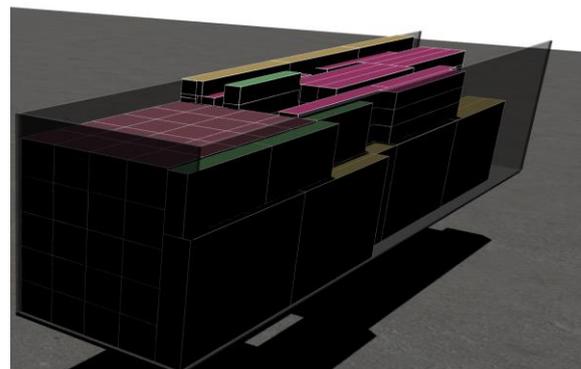
- b. Pilih *neighbor* terbaik dari solusi saat ini dengan memindahkan atau memutar satu atau beberapa barang dalam objek kontainer 3D.
  - c. Periksa apakah langkah ini ada dalam tabu *list* atau tidak, jika ada, cari *neighbor* terbaik lainnya [9].
  - d. Perbarui tabu *list* dengan memasukkan langkah yang baru saja diambil ke dalam tabu *list* [9].
- 6) Mengulangi langkah 4 selama batas iterasi atau kondisi berhenti terpenuhi [8] [9].
  - 7) Hasil dari algoritma tabu search merupakan koordinat terbaik untuk setiap barang dalam objek kontainer 3D, sehingga ruang yang tidak terpakai dapat diminimalkan [8].

### 3.1.7 Loading Hasil 3D Tata Letak Barang

Pada saat tabu search telah selesai menentukan posisi terbaik dari setiap barangnya, maka dari *backend* harus memanggil fungsi *postMessage* pada web worker agar data yang sudah ditentukan oleh tabu search dapat dikirim ke *frontend*. Data yang dikirim merupakan data hasil proses tabu search yang dilakukan pada poin 3.1.6 berupa koordinat untuk setiap barangnya dengan atribut x, y dan z.

*Frontend* akan menerima data yang dikirim oleh *backend* melalui web worker dan akan dilanjutkan dengan proses *rendering* 3D objek oleh ThreeJS, objek 3D ini akan merepresentasikan hasil *bin packing problem* dengan posisi terbaik yang telah ditemukan.

Proses pembentukan objek 3D barang dapat dilakukan melalui *BoxGeometry*, *MeshLambertMaterial*, *Mesh* dan untuk menentukan posisi koordinatnya dapat menggunakan fungsi *position.set()* pada kelas *Mesh*. Lalu pada setiap barangnya dapat ditambahkan ke *scene* yang telah dibuat sebelumnya pada poin 3.1.2



Gambar 9. 3D Objek Kontainer dan Tata Letak Barang

### 3.2 ThreeJS

ThreeJS adalah library JavaScript untuk membuat game dan aplikasi 3D. Three.js menggunakan WebGL yang mana merupakan API JavaScript untuk *render* grafik 2D dan 3D [11] interaktif yang *native* browser tanpa memerlukan plugin browser tambahan. Pada dasarnya kita dapat menggunakan API WebGL secara langsung untuk menampilkan model 3D. Ada beberapa *high-level library* untuk *rendering* 3D selain Three.js, diantaranya adalah PhiloGL, Scene.js, dan GLGE.



### 3.3 WebGL

WebGL, yang biasa dikenal juga sebagai *Web Graphics Library*, adalah teknologi web terbaru yang memungkinkan tampilan grafis 3D *real-time* di web menggunakan JavaScript, tanpa memerlukan tambahan plug-in seperti beberapa alat interaktif lainnya. Sebagai contoh, VRML (*Virtual Reality Modeling Language*) memerlukan pemasangan *plug-in* Cortona untuk memvisualisasikan objek 3D. Sebaliknya, WebGL berasal dari OpenGL dan menyediakan API khusus untuk grafis 3D, terintegrasi dengan mulus dalam konteks HTML melalui elemen kanvas HTML5 [12] dan WebGL memungkinkan pengembangan halaman website dengan struktur 3D yang kompleks dan bahkan dapat membangun permainan web 3D [13].

### 3.4 Web Worker

Web Worker merupakan fitur yang diperkenalkan dari HTML5 yang dapat mengizinkan developer untuk mengeksekusi *script* di latar belakang dan memisahkan dari pekekseskusan *thread* utama [14]. Fitur ini memudahkan developer untuk menjalankan *script* tanpa mengganggu aktivitas utama *user-interface*. Web Worker digunakan untuk melakukan sebuah proses yang tidak perlu dilihat oleh pengguna secara langsung yang biasanya dimanfaatkan untuk komunikasi antara *user-interface* dan *backend*.

## IV. KESIMPULAN

*Bin packing problem* merupakan kategori *NP-Hard* yang mana sulit untuk menentukan solusi mutlakanya, maka dari itu peneliti mengimplementasikan algoritma tabu search dalam konteks *container loading* dengan harapan dapat menambah ilmu dan referensi bagi peneliti lain.

Penelitian ini dimudahkan oleh ThreeJS yang memiliki peran penting pada penelitian untuk memudahkan proses rendering 3D objek.

Pada penelitian ini tabu search berhasil menemukan koordinat terbaiknya pada objek 3d kontainer, namun untuk menentukan akurasi tabu search pada konteks ini diperlukan penelitian lebih lanjut.

## UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

## V. REFERENSI

- [1] U. Khairuddin, N. A. Z. M. Razi, M. S. Z. Abidin, and R. Yusof, "Smart Packing Simulator for 3D Packing Problem Using Genetic Algorithm," in *Journal of Physics: Conference Series*, 2020, vol. 1447, no. 1. doi: 10.1088/1742-6596/1447/1/012041.
- [2] J. Kang and S. Park, "Algorithms for the variable sized bin packing problem," *Eur. J. Oper. Res.*, vol. 147, no. 2, pp. 365–372, 2003, doi: 10.1016/S0377-2217(02)00247-3.
- [3] X. Li, Z. Zhao, and K. Zhang, "A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins," *IIE Annu. Conf. Expo 2014*, no. March, pp. 2039–2048, 2014.
- [4] S. Martello, D. Pisinger, and D. Vigo, "Three-dimensional bin packing problem," *Oper. Res.*, vol. 48, no. 2, pp. 256–267, 2000, doi: 10.1287/opre.48.2.256.12386.
- [5] X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowland, "A comparative review of 3D container loading algorithms," *Int. Trans. Oper. Res.*, vol. 23, no. 1–2, pp. 287–320, 2016, doi: 10.1111/itor.12094.
- [6] S. Edition, *Data Mining for the Masses*.
- [7] J. Evers, *3D PACKING ALGORITHM AS TOOL FOR*. 2022.
- [8] P. Leon, R. Cueva, M. Tupia, and G. Paiva Dias, "A Taboo-Search Algorithm for 3D-Binpacking Problem in Containers," *Adv. Intell. Syst. Comput.*, vol. 930, pp. 229–240, 2019, doi: 10.1007/978-3-030-16181-1\_22.
- [9] G. Scheithauer, *Introduction to Cutting and Packing Optimization*, vol. 263. 2018. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-64403-5>
- [10] F. Gzara, S. Elhedhli, and B. C. Yildiz, "The Pallet Loading Problem: Three-dimensional bin packing with practical constraints," *Eur. J. Oper. Res.*, vol. 287, no. 3, pp. 1062–1074, 2020, doi: 10.1016/j.ejor.2020.04.053.
- [11] Y. Wang, Y. Li, Y. Tan, Q. Zhang, and J. Zheng, "Large Scale Network Topology Visualization System Based on ThreeJS," no. Icaita, pp. 152–155, 2016, doi: 10.2991/icaita-16.2016.39.
- [12] R. K. Moloo, S. Pudaruth, M. Ramodhin, and R. B. Rozbully, "A 3D Virtual Tour of the University of Mauritius using WebGL," *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 2891–2894, 2016, doi: 10.1109/ICEEOT.2016.7755226.
- [13] J. Chen and J. Chen, "Research on point aggregation algorithm based on WEBGL," *2019 IEEE 3rd Int. Conf. Electron. Inf. Technol. Comput. Eng. EITCE 2019*, pp. 1243–1246, 2019, doi: 10.1109/EITCE47263.2019.9095035.
- [14] Z. Wang, H. Deng, L. Hu, and X. Zhu, "HTML5 web worker transparent offloading method for web applications," *Int. Conf. Commun. Technol. Proceedings, ICCT*, vol. 2019-October, pp. 1319–1323, 2019, doi: 10.1109/ICCT.2018.8600046.