

# **Laporan Tugas Besar 2**

## **Strategi Algoritma**



### **Kelompok 44**

### **Bayu Sang Alkemis**

Anggota Kelompok:  
Orvin Andika I A (13523017)  
Joel Hotlan H S (13523025)  
Fajar Kurniawan (13523027)

**Program Studi Teknik Informatika**  
**Institut Teknologi Bandung**

# Daftar Isi

<b>Daftar Isi.....</b>	<b>2</b>
<b>Daftar Gambar.....</b>	<b>4</b>
<b>Daftar Tabel.....</b>	<b>5</b>
<b>Bab 1.....</b>	<b>6</b>
<b>Deskripsi Tugas.....</b>	<b>6</b>
<b>Bab 2.....</b>	<b>8</b>
<b>Landasan Teori.....</b>	<b>8</b>
2.1. Penjelajahan Graf.....	8
2.2. Algoritma Breadth First Search.....	8
2.3. Algoritma Depth First Search.....	9
2.4. Docker.....	9
2.5. Deployment.....	9
2.6. Pembangunan Aplikasi Web.....	10
<b>Bab 3.....</b>	<b>11</b>
<b>Analisis Pemecahan Masalah.....</b>	<b>11</b>
3.1. Langkah-Langkah Pemecahan Masalah.....	11
3.2. Pemetaan Masalah.....	11
3.3. Fitur Fungsional dan Arsitektur Web.....	11
3.4. Contoh Ilustrasi Kasus.....	12
3.4.1. BFS.....	12
3.4.2. DFS.....	12
<b>Bab 4.....</b>	<b>13</b>
<b>Implementasi dan Pengujian.....</b>	<b>13</b>
4.1. Spesifikasi Teknis Program.....	13
4.1.1. Struktur Data.....	13
4.1.2. Fungsi dan Prosedur.....	13
4.2. Tata Cara Penggunaan Program.....	15
4.2.1. Interface.....	15
4.2.2. Cara Menjalankan Program.....	15
4.2.3. Cara Penggunaan Program.....	15
4.3. Hasil Pengujian.....	16
4.3.1. Hasil Pengujian Breadth First Search.....	16
4.3.2. Hasil Pengujian Depth First Search.....	19
4.4. Analisis Hasil Pengujian.....	23
<b>Bab 5.....</b>	<b>24</b>
<b>Kesimpulan.....</b>	<b>24</b>
5.1. Kesimpulan.....	24
5.2. Saran.....	24
5.3. Refleksi.....	24

<b>Lampiran.....</b>	<b>25</b>
Tautan Repository Github.....	25
Tautan Video.....	25
<b>Daftar Pustaka.....</b>	<b>26</b>

## Daftar Gambar

Gambar 1.1 Little Alchemy 2.....	6
Gambar 1.2 Elemen dasar pada Little Alchemy 2.....	7
Gambar 2.2.1 Ilustrasi BFS.....	9
Gambar 2.3.1 Ilustrasi DFS.....	9
Gambar 3.3.1 Elemen Brick dan Resepnya.....	12
Tabel 4.1.2.1 Fungsi dan Prosedur Program.....	13
Gambar 4.1.1 Antarmuka Program.....	15
Gambar 4.3.1.1 Hasil Pencarian Single Recipe Brick dengan BFS.....	16
Gambar 4.3.1.2 Hasil Pencarian Multiple Recipe 5 Brick dengan DFS.....	16
Gambar 4.3.1.3 Hasil Pencarian Single Recipe Hero dengan BFS.....	17
Gambar 4.3.1.4 Hasil Pencarian Multiple Recipe 2 Hero Recipe 1 dengan BFS.....	17
Gambar 4.3.1.5 Hasil Pencarian Multiple Recipe 2 Hero Recipe 2 dengan BFS.....	17
Gambar 4.3.1.6 Hasil Pencarian Single Recipe Angler dengan BFS.....	17
Gambar 4.3.1.7 Hasil Pencarian Multiple Recipe 2 Angler Recipe 1 dengan BFS.....	18
Gambar 4.3.1.8 Hasil Pencarian Multiple Recipe 2 Angler Recipe 2 dengan BFS.....	18
Gambar 4.3.1.9 Hasil Pencarian Single Recipe Bread dengan BFS.....	19
Gambar 4.3.1.10 Hasil Pencarian Multiple Recipe 2 Bread Recipe 1 dengan BFS.....	19
Gambar 4.3.1.11 Hasil Pencarian Multiple Recipe 2 Bread Recipe 2 dengan BFS.....	19
Gambar 4.3.2.1 Hasil Pencarian Single Recipe Brick dengan DFS.....	20
Gambar 4.3.2.2 Hasil Pencarian Multiple Recipe 5 Brick dengan DFS.....	20
Gambar 4.3.2.3 Hasil Pencarian Single Recipe Hero dengan DFS.....	20
Gambar 4.3.2.4 Hasil Pencarian Multiple Recipe 2 Hero Recipe 1 dengan DFS.....	21
Gambar 4.3.2.5 Hasil Pencarian Multiple Recipe 2 Hero Recipe 2 dengan DFS.....	21
Gambar 4.3.2.6 Hasil Pencarian Single Recipe Angler dengan DFS.....	21
Gambar 4.3.2.7 Hasil Pencarian Multiple Recipe 2 Angler Recipe 1 dengan DFS.....	22
Gambar 4.3.2.8 Hasil Pencarian Multiple Recipe 2 Angler Recipe 2 dengan DFS.....	22
Gambar 4.3.2.9 Hasil Pencarian Single Recipe Bread dengan DFS.....	22
Gambar 4.3.2.10 Hasil Pencarian Multiple Recipe 2 Bread Recipe 1 dengan DFS.....	23
Gambar 4.3.2.11 Hasil Pencarian Multiple Recipe 2 Bread Recipe 2 dengan DFS.....	23

## **Daftar Tabel**

Tabel 4.1.2.1 Fungsi dan Prosedur Program.....	14
--	----

## **Bab 1**

### **Deskripsi Tugas**





Gambar 1.2 Elemen dasar pada Little Alchemy 2

2. Elemen turunan

Terdapat 720 elemen turunan yang dibagi menjadi beberapa *tier* tergantung tingkat kesulitan dan banyak langkah yang harus dilakukan. Setiap elemen turunan memiliki *recipe* yang terdiri atas elemen lainnya atau elemen itu sendiri.

3. *Combine Mechanism*

Untuk mendapatkan elemen turunan pemain dapat melakukan *combine* antara 2 elemen untuk menghasilkan elemen baru. Elemen turunan yang telah didapatkan dapat digunakan kembali oleh pemain untuk membentuk elemen lainnya.

## Bab 2

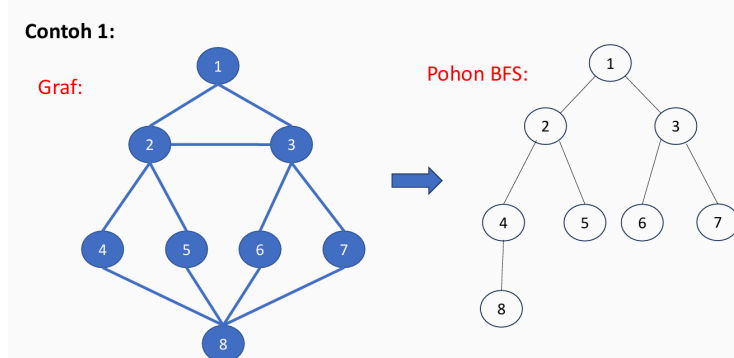
# Landasan Teori

### 2.1. Penjelajahan Graf

Penjelajahan graf adalah metode sistematis untuk mengunjungi simpul-simpul yang ada dalam graf. Penjelajahan graf juga bisa dianggap sebagai pencarian solusi persoalan yang direpresentasikan dengan graf. Terdapat dua jenis pendekatan dalam penjelajahan graf. Jenis pertama adalah uninformed search. Uninformed search atau blind search adalah metode pencarian yang tidak menggunakan pengetahuan tambahan selain struktur graf itu sendiri. Algoritma BFS dan DFS termasuk ke dalam jenis ini. Kelebihan dari jenis ini adalah jenis ini tidak membutuhkan heuristik apa pun sehingga sangat sederhana. Sementara itu, kekurangannya adalah jenis ini tidak cocok untuk graf yang besar karena akan melakukan pencarian yang tidak perlu. Jenis kedua adalah informed search. Informed search adalah proses pencarian menggunakan heuristik untuk mengarahkan pencarian ke tujuan. Algoritma Greedy Best First Search dan A\* termasuk ke dalam jenis ini. Kelebihan jenis ini adalah proses pencariannya lebih efisien. Sementara itu, kekurangannya dari jenis ini adalah diperlukannya fungsi heuristik yang baik agar pencarian efisien, tetapi pemilihan heuristik terkadang adalah hal yang rumit.

### 2.2. Algoritma Breadth First Search

Cara kerja algoritma ini dimulai dengan mengunjungi simpul awal. Kemudian kunjungi semua simpul yang bertetangga dengan simpul awal. Setelah itu, kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang sudah dikunjungi. Hal ini diteruskan hingga simpul tujuan ditemukan. Algoritma ini memanfaatkan beberapa struktur data. Pertama, algoritma ini memanfaatkan struktur data matriks ketetanggaan yang berukuran  $n \times n$  dengan  $n$  adalah jumlah simpul. Kedua, algoritma ini menggunakan queue untuk menyimpan simpul yang telah dikunjungi. Terakhir, algoritma ini menggunakan tabel boolean yang berisi true jika simpul sudah dikunjungi. Berikut adalah ilustrasi cara kerja algoritma BFS.

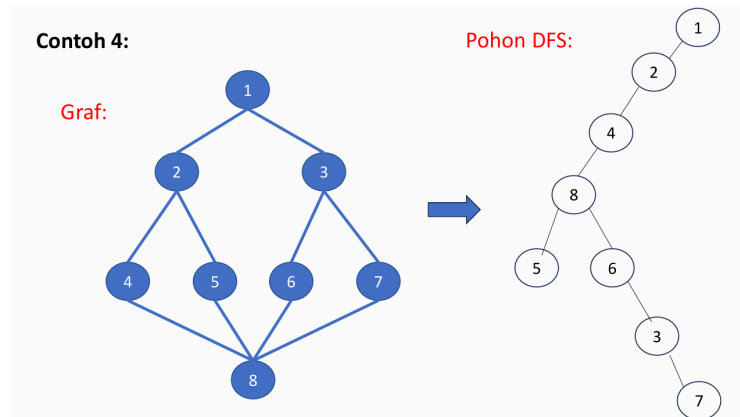




Gambar 2.2.1 Ilustrasi BFS

## 2.3. Algoritma Depth First Search

Cara kerja algoritma ini dimulai dengan mengunjungi simpul awal, misalnya v. Kemudian algoritma akan mengunjungi simpul w yang bertetangga dengan v. Setelah itu, algoritma DFS diulang dari simpul w. Setelah algoritma sampai pada simpul u yang semua tetangganya telah dikunjungi, pencarian akan kembali ke simpul sebelumnya yang masih memiliki tetangga yang belum dikunjungi. Pencarian berakhir saat tidak ada lagi simpul yang belum dikunjungi dan dapat dicapai dari simpul yang sudah dikunjungi. Berikut ilustrasi cara kerja algoritma DFS.



Gambar 2.3.1 Ilustrasi DFS

## 2.4. Docker

Docker adalah software open-source yang digunakan untuk meluncurkan (deploy) aplikasi di dalam container virtual. Dengan container virtual ini (containerization), aplikasi bisa dijalankan secara terisolasi di environment yang kompleks sehingga tidak menimbulkan masalah pada environment lainnya. Docker bekerja dengan menciptakan sebuah ruang isolasi untuk meluncurkan aplikasi atau layanan. Ruang isolasi ini disebut Container, seperti ‘wadah’ yang akan menampung suatu benda agar tidak tumpah ke area lain. Docker container adalah sebuah paket software yang berisi semua dependensi yang diperlukan untuk menjalankan aplikasi tertentu. Kelebihan dari Docker adalah portabilitasnya. Docker memungkinkan pengguna untuk membuat atau menginstall aplikasi kompleks dan memastikan aplikasi tersebut dapat berjalan. Sementara itu, kekurangan Docker adalah kecepatannya yang masih lebih rendah dari server fisik.

## 2.5. Deployment

Deployment adalah proses penerapan atau penempatan suatu aplikasi, sistem, solusi, atau teknologi ke dalam lingkungan operasional yang sesungguhnya setelah melalui tahap pengembangan (*development*). Proses *deployment* melibatkan penempatan aplikasi atau

sistem ke dalam lingkungan produksi atau operasional yang biasanya berbeda dari lingkungan pengembangan atau pengujian. Tujuan utama dari *deployment* adalah membuat aplikasi atau sistem tersedia untuk pengguna akhir (*end-user*) sehingga mereka dapat memanfaatkannya sesuai dengan kebutuhan.

## 2.6. Pembangunan Aplikasi Web

Aplikasi web ini dibangun untuk membantu orang untuk membuat elemen tertentu dalam permainan Little Alchemy 2. Aplikasi ini akan mengirimkan resep elemen yang ditekan oleh pengguna. Pencarian resep dilakukan dengan memanfaatkan dua algoritma, yaitu Breadth First Search dan Depth First Search. Pengguna dapat memilih algoritma yang ingin digunakan dalam resep. Selain itu, aplikasi ini juga memiliki dua mode, yaitu single recipe dan multiple recipe. Dalam mode multiple recipe, pengguna dapat memasukkan jumlah resep maksimal yang diinginkan. Fitur lain yang ada dalam aplikasi web ini adalah pagination dan search untuk menangani banyaknya elemen yang ada dalam permainan Little Alchemy 2.

Aplikasi web ini dibangun menggunakan Reactjs untuk bagian frontend dan Golang untuk bagian backend. Golang menawarkan kemampuan untuk menangani beban berat tanpa mengurangi kecepatan respon aplikasi. Hal ini dibutuhkan baik dalam scraping data resep dari web Little Alchemy 2 maupun pencarian resep menggunakan BFS atau DFS.

## **Bab 3**

### **Analisis Pemecahan Masalah**

#### **3.1. Langkah-Langkah Pemecahan Masalah**

Permasalahan dalam Little Alchemy 2 adalah mencari kombinasi beberapa elemen yang dapat menghasilkan elemen tertentu. Dengan web ini, pengguna harus memilih elemen mana yang ingin dicari resepnya. Kemudian, pengguna memilih algoritma mana yang ingin dipakai. Algoritma yang tersedia adalah algoritma BFS dan DFS. Selain itu, pengguna juga harus memilih apakah ingin menggunakan mode single recipe atau multiple recipe.

Frontend akan mengirimkan request kepada backend berisi elemen, mode, dan algoritma. Backend kemudian akan mencari resep yang diinginkan sesuai request. Setelah pencarian selesai backend akan mengirim data kembali ke frontend. Frontend akan menggunakan data tersebut untuk membangun sebuah tree, menampilkan simpul yang dikunjungi, dan waktu yang digunakan dalam pencarian.

#### **3.2. Pemetaan Masalah**

Berikut adalah pemetaan masalah Little Alchemy 2 berdasarkan pemecahan masalah traversal graf:

- a. Simpul : Semua elemen dalam permainan Little Alchemy 2
- b. Sisi : Suatu elemen dapat dibentuk dari elemen lain sehingga membentuk sisi.
- c. Simpul awal : Elemen yang ingin dicari resepnya
- d. Simpul akhir : Elemen dasar

Terdapat 720 elemen dalam permainan Little Alchemy 2 yang dibagi menjadi tier-tier yang berkorelasi dengan tingkat kerumitan pembuatannya. Karena tier tertinggi adalah 15, dapat terlihat bahwa kombinasi resep yang ada sangatlah banyak.

#### **3.3. Fitur Fungsional dan Arsitektur Web**

Aplikasi web ini dirancang untuk membantu pemain Little Alchemy 2 untuk mencari resep dari suatu elemen yang ingin ia buat. Aplikasi web ini memiliki beberapa fitur untuk memudahkan penggunaannya. Pertama, web ini bisa melakukan scraping seluruh resep dari elemen yang ada di permainan Little Alchemy 2. Aplikasi web ini juga bisa menerima input elemen hanya dengan menekan ikon dari elemen tersebut. Terdapat fitur search untuk memudahkan pengguna mencari elemen yang dia inginkan. Untuk mode dan algoritma bisa dipilih dengan menekan checkbox dari pilihan yang diinginkan. Setelah menerima input yang dibutuhkan, aplikasi web ini bisa melakukan pencarian resep elemen dengan algoritma BFS dan DFS. Aplikasi ini juga mendukung visualisasi resep dalam bentuk tree.

Aplikasi web ini dibangun dengan mengintegrasikan frontend sebagai antarmuka untuk user dan backend yang digunakan untuk mencari solusi. Pada sisi frontend, aplikasi ini menggunakan ReactJs untuk memberikan antarmuka yang interaktif pada pengguna.

Pengguna dapat mengakses bagian frontend dan memberikan data yang diperlukan untuk dikirimkan ke backend. Sementara itu, bagian backend dari aplikasi ini dibangun menggunakan Golang. Bagian backend akan merespon permintaan dari frontend dan mengirimkan elemen yang dicari ke frontend dalam bentuk json.

### 3.4. Contoh Ilustrasi Kasus

Untuk memperjelas analisis pemecahan masalah, kami akan memberikan sebuah ilustrasi kasus dan pemecahannya. Kasus yang digunakan adalah kasus dengan elemen yang ingin dicari adalah Brick. Karena Brick adalah elemen tier 2, ia hanya bisa dibentuk dengan elemen pada tier 1 atau di bawahnya. Oleh karena itu, resep yang valid hanyalah Mud + Fire.

Element	Recipes
 Brick	<ul style="list-style-type: none"> <li>•  Mud +  Fire</li> <li>•  Mud +  Sun</li> <li>•  Clay +  Fire</li> <li>•  Clay +  Sun</li> <li>•  Clay +  Stone</li> </ul>

Gambar 3.3.1 Elemen Brick dan Resepnya

#### 3.4.1. BFS

Pada iterasi ke-0, program akan mengunjungi elemen yang dicari yaitu Brick. Pada iterasi ke-1, program akan mengunjungi seluruh elemen dalam resep Brick, dalam konteks ini hanya Mud dan Fire. Karena Fire merupakan elemen dasar, maka pencarian dari cabang Fire berhenti di situ. Pada iterasi ke-2, program akan mengunjungi seluruh elemen dalam resep Mud. Resep Mud adalah Earth dan Water. Karena kedua elemen merupakan elemen dasar, pencarian BFS berhenti.

#### 3.4.2. DFS

Pada iterasi ke-0, program akan mengunjungi elemen yang dicari yaitu Brick. Pada iterasi ke-1, program akan mengunjungi elemen pertama dalam resep Brick, yaitu Mud. Pada iterasi ke-2, program akan mengunjungi elemen pertama dalam resep Mud, yaitu Earth. Karena Earth adalah elemen dasar, pada iterasi ke-3, program akan mengunjungi elemen kedua dalam resep Mud, yaitu Water. Karena Water adalah elemen dasar, pada iterasi ke-4, program akan mengunjungi elemen kedua dalam resep Brick, yaitu Fire. Karena Fire adalah elemen dasar, pencarian DFS dihentikan.

## Bab 4

# Implementasi dan Pengujian

### 4.1. Spesifikasi Teknis Program

#### 4.1.1. Struktur Data

Terdapat beberapa struktur data yang digunakan dalam pembangun aplikasi ini. Struktur data tersebut digunakan baik di bagian frontend maupun bagian backend.

Struktur data pertama adalah RecipeRequest. Struktur data ini digunakan untuk merepresentasikan input dari pengguna yang berisi elemen, algoritma, dan jumlah resep maksimalnya.

Struktur data kedua adalah RecipeResponse. Struktur data ini digunakan untuk merepresentasikan hasil pencarian yang dilakukan oleh bagian backend. Struktur data ini berisi resep-resep, waktu yang diperlukan, dan jumlah simpul yang dikunjungi. Struktur data ini yang akan dikirim ke frontend untuk menampilkan hasil.

Struktur data ketiga adalah Recipe. Struktur data ini merepresentasikan resep dari tiap elemen. Struktur data ini berisi tier elemen dan resepnya. Struktur data ini digunakan baik pada algoritma BFS maupun DFS.

#### 4.1.2. Fungsi dan Prosedur

Tabel 4.1.2.1 Fungsi dan Prosedur Program

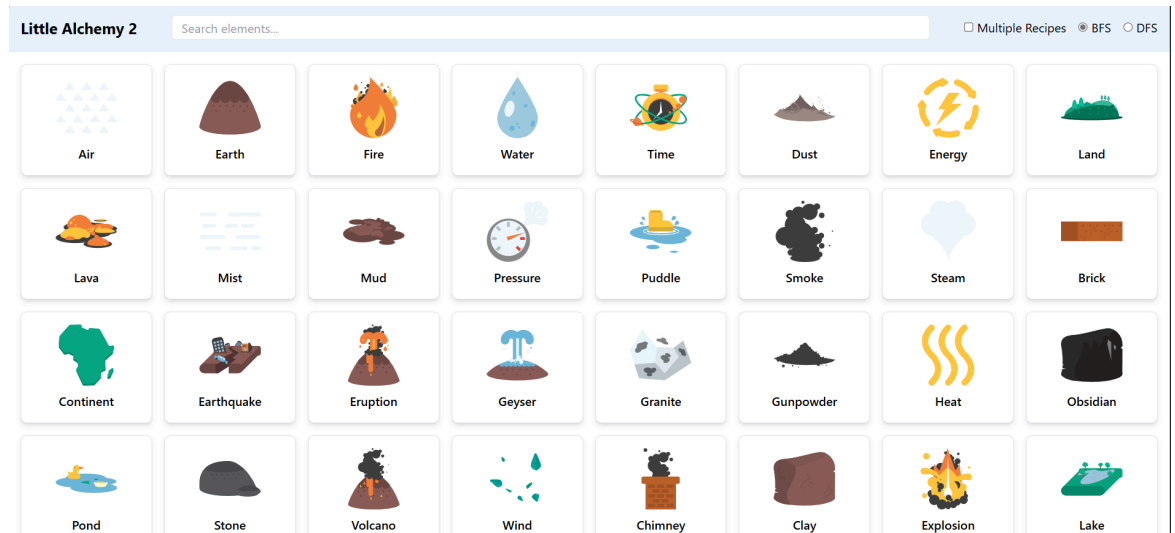
Fungsi / Prosedur	Deskripsi
ElemenList({onSelect, search})	Fungsi ini digunakan untuk menampilkan seluruh elemen yang ada pada permainan Little Alchemy 2
generateRecipeTree(element, RecipeMap)	Fungsi ini digunakan untuk menyusun tree dari resep yang dikirim backend
RecipeTree({element, MaxRecipes, searchMode})	Fungsi ini digunakan untuk mengirim request ke backend, menerima response, dan menampilkan hasil
TopBar()	Fungsi ini digunakan untuk menampilkan topbar dan menerima input mode dan algoritma
App()	Fungsi ini yang menggabungkan semua elemen frontend dan menampilkannya
enableCORS()	Prosedur ini digunakan untuk

	mengatur konfigurasi CORS
exploreRecipe()	Fungsi ini yang akan menjalankan pencarian sesuai dengan input pengguna
handleRecipe()	Fungsi ini yang akan mengubah hasil pencarian menjadi struktur data RecipeResponse
main()	Fungsi utama backend
DFS()	Fungsi ini menyediakan antarmuka umum untuk pencarian DFS
BFS()	Fungsi ini menyediakan antarmuka umum untuk pencarian BFS
CompleteScrapRecipe()	Fungsi ini digunakan untuk melakukan scraping resep dari website Little Alchemy
ScrapeToJsonComplete()	Fungsi ini menyediakan antarmuka umum untuk scraping
CleanRecipes()	Fungsi ini menghapus resep yang memiliki elemen yang tidak ada
filterRecipes()	Fungsi ini menghapus resep yang memiliki elemen dengan tier lebih tinggi
removeAllInvalidRecipes()	Fungsi ini menghapus resep dengan elemen yang tidak memiliki resep dan bukan elemen dasar
readRecipeJson()	Fungsi ini membaca hasil scraping dan menyimpannya dalam map[string]Recipe
isRecipeComplete()	Fungsi ini mengecek apakah resep sudah memiliki semua komponen yang diperlukan
searchDFS()	Fungsi ini yang melakukan pencarian resep dengan algoritma DFS
searchBFS()	Fungsi ini yang melakukan pencarian resep dengan algoritma BFS

## 4.2. Tata Cara Penggunaan Program

### 4.2.1. Interface

Berikut adalah interface dari aplikasi web kami.



Gambar 4.1.1 Antarmuka Program

### 4.2.2. Cara Menjalankan Program

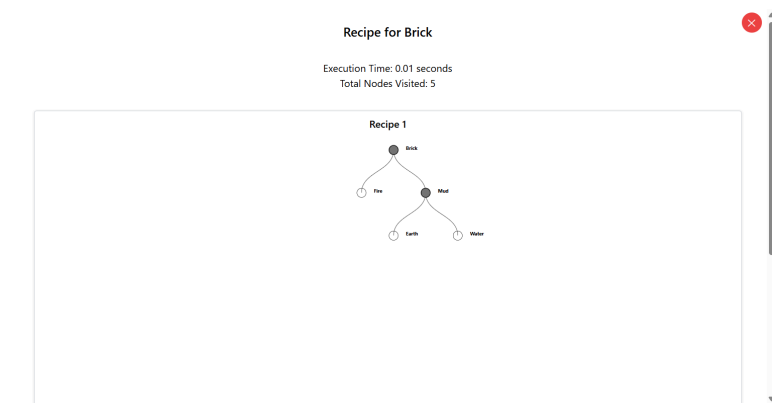
Program bisa dijalankan dengan mengakses alamat berikut pada perangkat yang tersambung ke internet <https://recipe-finder-production-6d64.up.railway.app/>

### 4.2.3. Cara Penggunaan Program

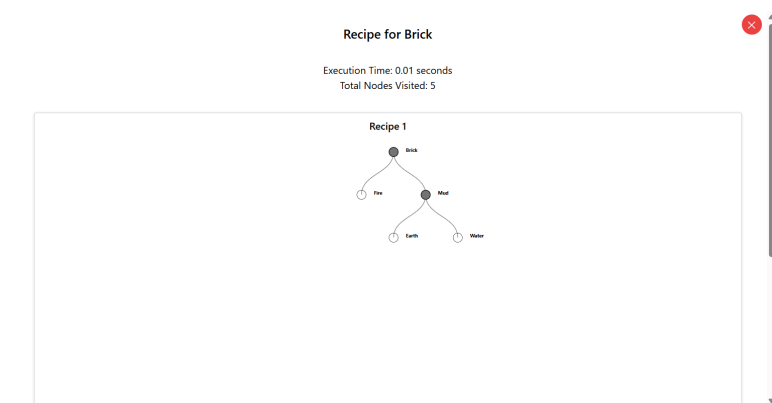
- Pilih mode yang ingin digunakan (single atau multiple recipe)
- Jika multiple recipe, masukkan jumlah resep maksimal
- Pilih algoritma yang ingin digunakan (BFS atau DFS)
- Pilih elemen yang ingin dicari resepnya (bisa memanfaatkan fitur pencarian)
- Tunggu hingga hasil ditampilkan dalam bentuk pohon biner.

## 4.3. Hasil Pengujian

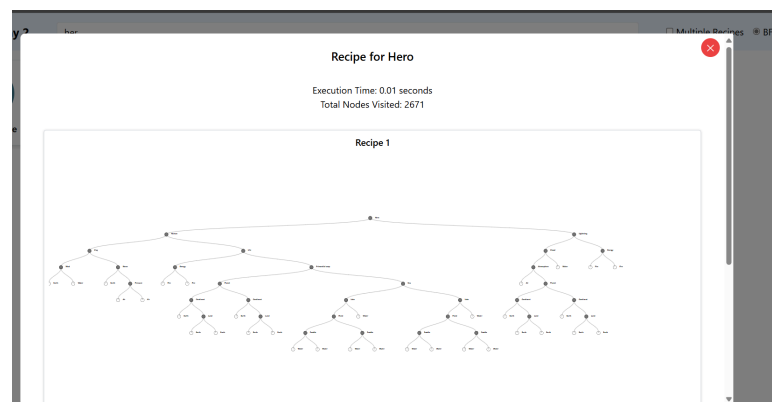
### 4.3.1. Hasil Pengujian Breadth First Search



Gambar 4.3.1.1 Hasil Pencarian Single Recipe Brick dengan BFS

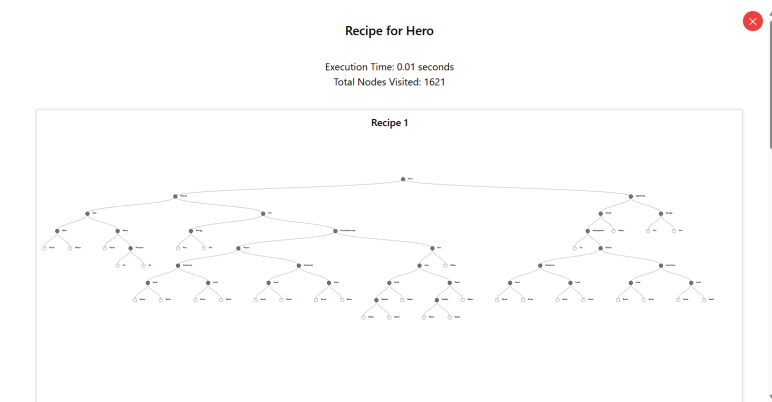


Gambar 4.3.1.2 Hasil Pencarian Multiple Recipe 5 Brick dengan DFS

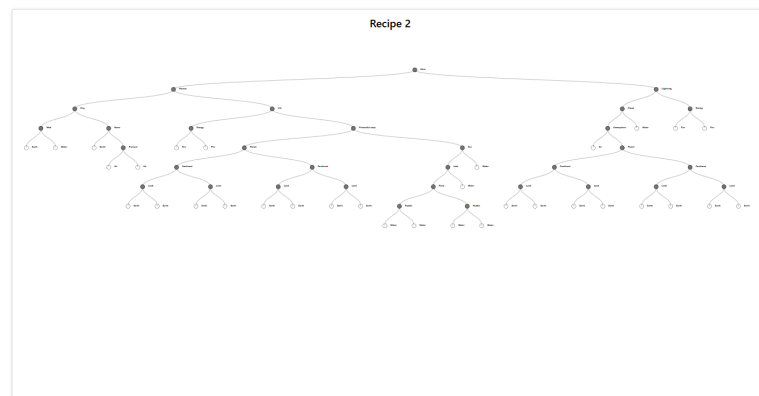




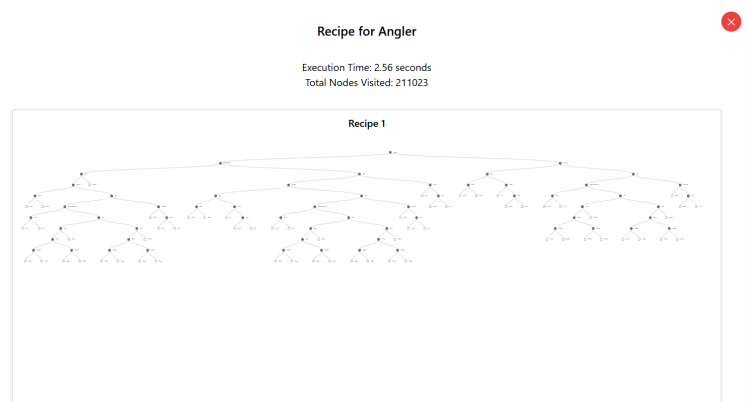
Gambar 4.3.1.3 Hasil Pencarian Single Recipe Hero dengan BFS



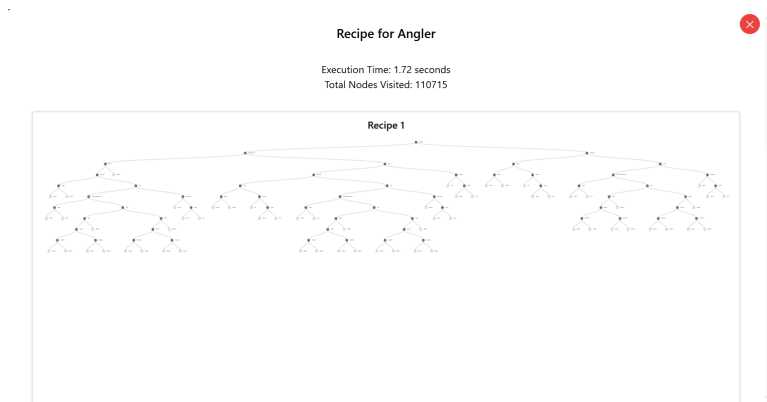
Gambar 4.3.1.4 Hasil Pencarian Multiple Recipe 2 Hero Recipe 1 dengan BFS



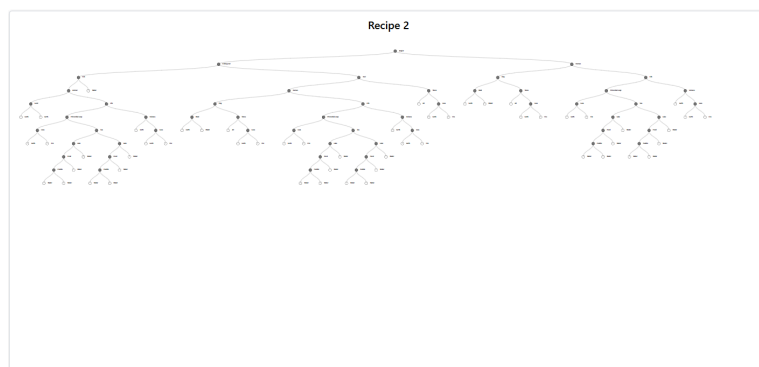
Gambar 4.3.1.5 Hasil Pencarian Multiple Recipe 2 Hero Recipe 2 dengan BFS



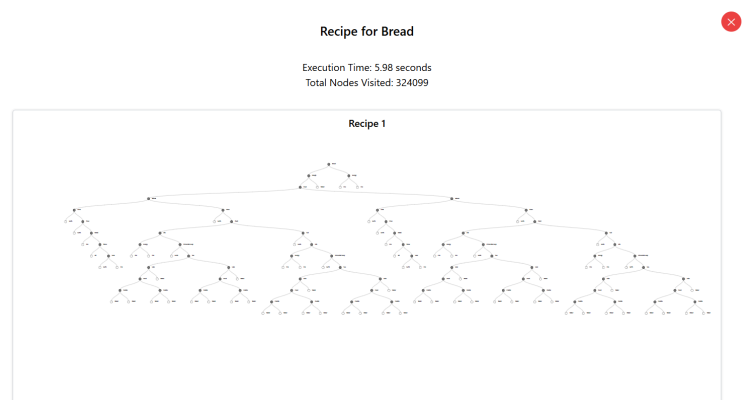
Gambar 4.3.1.6 Hasil Pencarian Single Recipe Angler dengan BFS



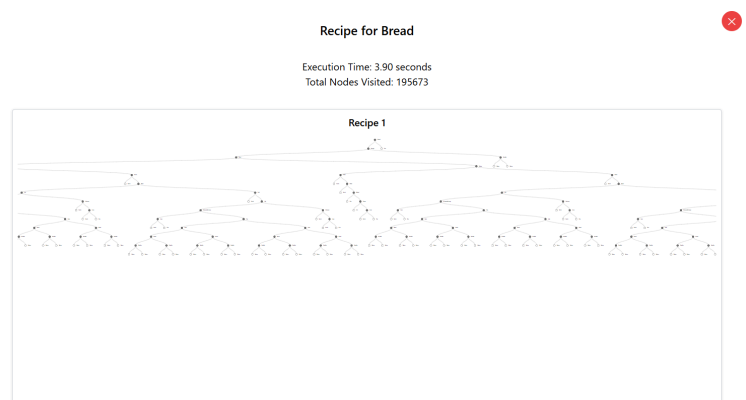
Gambar 4.3.1.7 Hasil Pencarian Multiple Recipe 2 Angler Recipe 1 dengan BFS



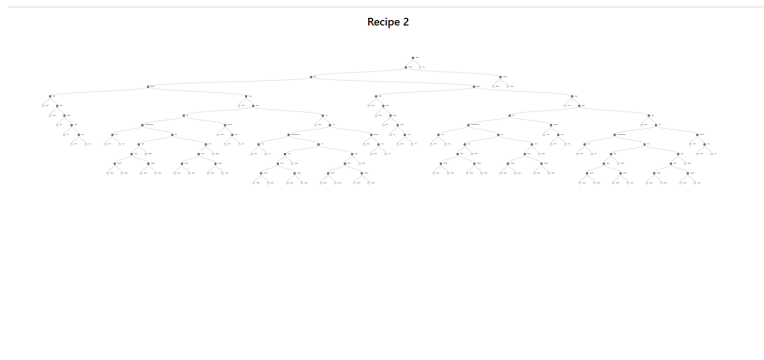
Gambar 4.3.1.8 Hasil Pencarian Multiple Recipe 2 Angler Recipe 2 dengan BFS



Gambar 4.3.1.9 Hasil Pencarian Single Recipe Bread dengan BFS

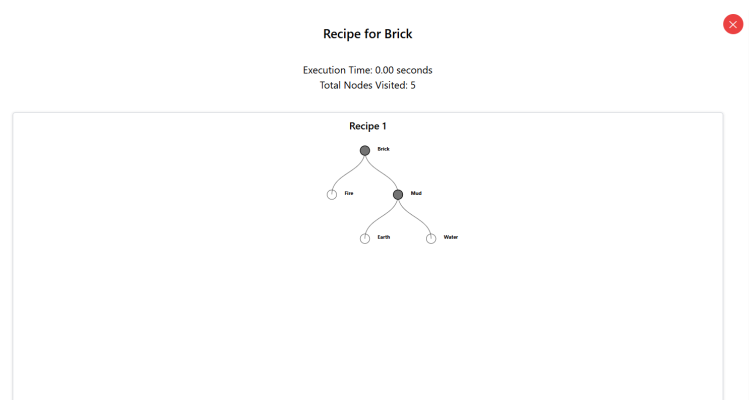


Gambar 4.3.1.10 Hasil Pencarian Multiple Recipe 2 Bread Recipe 1 dengan BFS

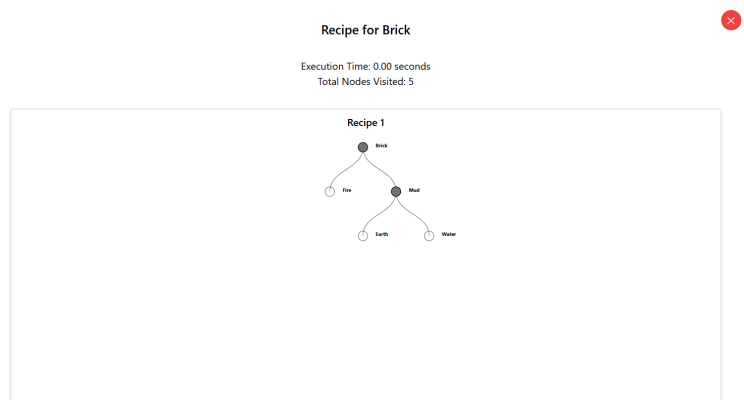


Gambar 4.3.1.11 Hasil Pencarian Multiple Recipe 2 Bread Recipe 2 dengan BFS

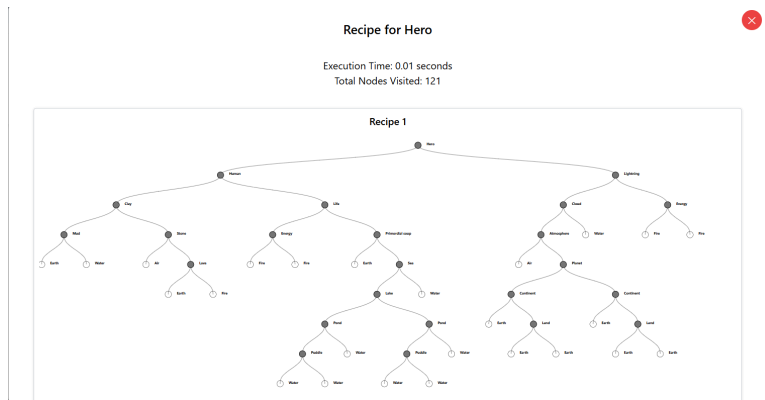
#### 4.3.2. Hasil Pengujian Depth First Search



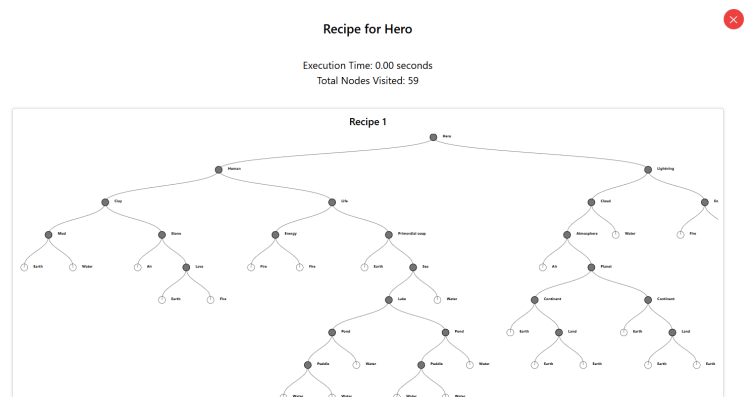
Gambar 4.3.2.1 Hasil Pencarian Single Recipe Brick dengan DFS



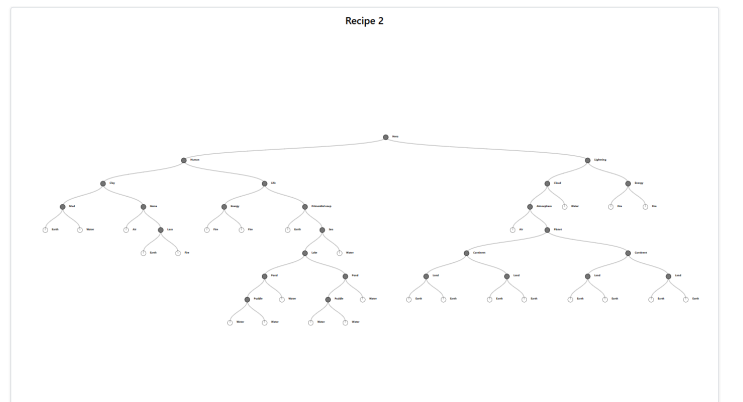
Gambar 4.3.2.2 Hasil Pencarian Multiple Recipe 5 Brick dengan DFS



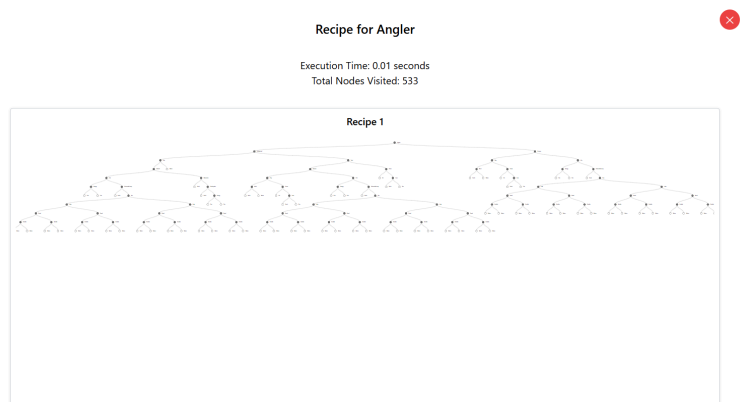
Gambar 4.3.2.3 Hasil Pencarian Single Recipe Hero dengan DFS



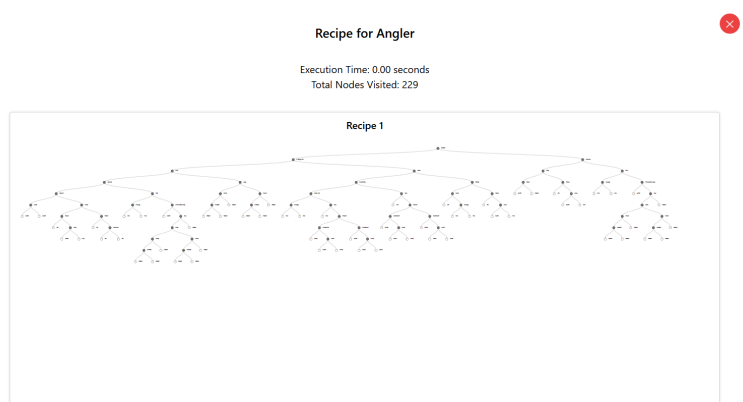
Gambar 4.3.2.4 Hasil Pencarian Multiple Recipe 2 Hero Recipe 1 dengan DFS



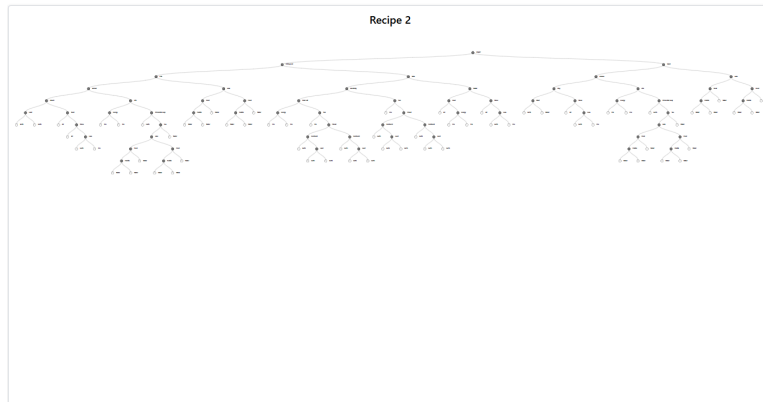
Gambar 4.3.2.5 Hasil Pencarian Multiple Recipe 2 Hero Recipe 2 dengan DFS



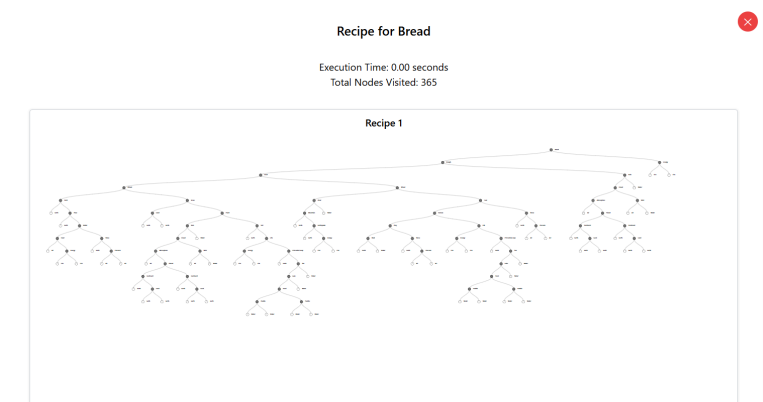
Gambar 4.3.2.6 Hasil Pencarian Single Recipe Angler dengan DFS



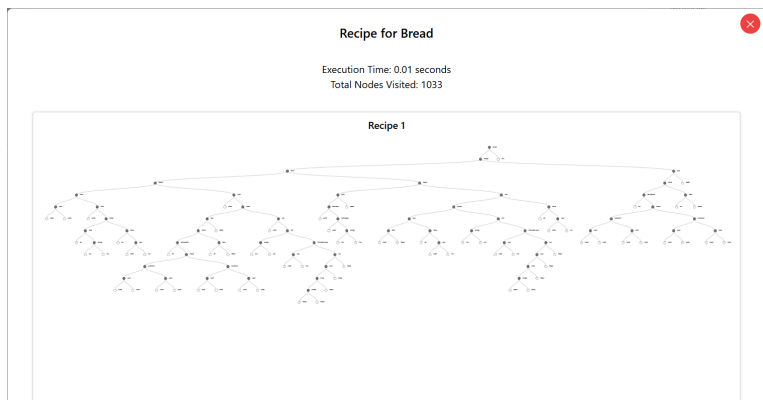
Gambar 4.3.2.7 Hasil Pencarian Multiple Recipe 2 Angler Recipe 1 dengan DFS



Gambar 4.3.2.8 Hasil Pencarian Multiple Recipe 2 Angler Recipe 2 dengan DFS



Gambar 4.3.2.9 Hasil Pencarian Single Recipe Bread dengan DFS



## Simpulan Hasil Pengujian

Berdasarkan hasil pengujian, kedua algoritma dapat memberikan resep yang valid dari elemen target ke elemen dasar sebagai leaf. Akan tetapi, karakteristik pencarian kedua algoritma sangat berbeda. Breadth First Search melakukan pencarian secara melebar, memprioritaskan kombinasi dari resep-resep target terlebih dahulu. Hal ini memberikan kekonsistenan urutan eksplorasi. Depth First Search melakukan pencarian secara mendalam, sehingga bisa lebih cepat untuk mencapai elemen dasar.

# Bab 5

## Kesimpulan

### 5.1. Kesimpulan

Pada pengerjaan tugas besar ini, telah berhasil dibangun aplikasi web yang dapat mencari resep elemen tertentu di permainan Little Alchemy 2. Program ini memanfaatkan ReactJs untuk antarmuka pengguna dan algoritma BFS atau DFS untuk pencarian resep. Pada antarmuka pengguna, tersedia fitur pagination dan search untuk memudahkan pengguna mencari resep yang diinginkan. Algoritma pencarian resep juga terbilang efektif karena mampu memberikan resep dengan waktu yang relatif cepat.

### 5.2. Saran

Saran untuk pengembangan aplikasi adalah peningkatan pada tampilan antarmuka. Antarmuka saat ini masih sangat sederhana. Selain itu, optimasi juga bisa dilakukan pada algoritma pencarian. Algoritma lain seperti bidirectional juga bisa ditambahkan. Selain itu, bisa ditambahkan fitur live update untuk meningkatkan pengalaman pengguna.

Pendalaman pemahaman multithreading pada Go juga diperlukan. Multithreading pada Go sangatlah kuat karena bisa melakukan beberapa pekerjaan tanpa mengurangi kecepatan respon aplikasi. Akan tetapi, jika tidak berhati-hati dapat menyebabkan race condition yang akan mengganggu keberjalanan program

### 5.3. Refleksi

Pengerjaan tugas besar ini memberikan pengalaman baru terhadap teknik scraping dan konkurensi dalam pemrograman. Tugas besar ini juga menambah pengalaman dalam pengembangan aplikasi web. Selain hard skill, pengerjaan tugas besar ini juga memberikan pelajaran tentang pembagian tugas dan kerja sama.

Pengerjaan tugas besar ini juga memberikan pelajaran agar tidak menunda-nunda pekerjaan. Pekerjaan yang mungkin terlihat mudah akan tetap sulit jika kita kehabisan waktu. Selain itu, banyaknya tugas lain juga membuat kita bisa meningkatkan kemampuan manajemen waktu dalam pengerjaan tugas besar ini.




# Lampiran

## Tautan Repository Github

[orvin14/Tubes2\\_BayuSangAlkemis](#)

## Tautan Video

 Tubes Stima 2 - Bayu Sang Alkemis

No	Poin	Ya	Tidak
1	Aplikasi dapat dijalankan.	V	
2	Aplikasi dapat memperoleh data <i>recipe</i> melalui scraping.	V	
3	Algoritma <i>Depth First Search</i> dan <i>Breadth First Search</i> dapat menemukan <i>recipe</i> elemen dengan benar.	V	
4	Aplikasi dapat menampilkan visualisasi <i>recipe</i> elemen yang dicari sesuai dengan spesifikasi.	V	
5	Aplikasi mengimplementasikan multithreading.	V	
6	Membuat laporan sesuai dengan spesifikasi.	V	
7	Membuat bonus video dan diunggah pada Youtube.	V	
8	Membuat bonus algoritma pencarian <i>Bidirectional</i> .		V
9	Membuat bonus <i>Live Update</i> .		V
10	Aplikasi di- <i>containerize</i> dengan Docker.	V	
11	Aplikasi di- <i>deploy</i> dan dapat diakses melalui internet.	V	

# Daftar Pustaka

- Rinaldi Munir. 2024. Breadth/Depth First Search (BFS/DFS) (Bagian 1)  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/BFS-DFS-2021-Bag1-2024.pdf> Rinaldi
- Munir. 2021. Breadth/Depth First Search (BFS/DFS) (Bagian 2)  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag2.pdf>
- Hostinger. Apa itu Docker? Diakses pada 10 Mei 2025  
[Apa Itu Docker? Pengertian Docker, Cara Kerja, dan Fungsinya](#)
- Dicoding. Deployment: Pengertian, Tujuan, dan Jenis. Diakses pada 10 Mei 2025.  
[Deployment: Pengertian, Tujuan, dan Jenis-jenisnya](#)