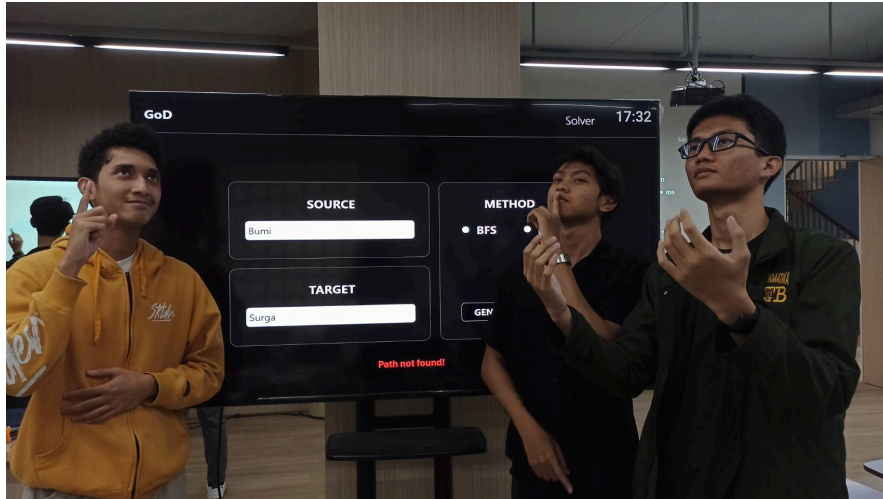


Tugas Besar 2 IF2211 Strategi Algoritma
Semester II tahun 2023/2024
**Pemanfaatan Algoritma IDS dan BFS dalam Permainan
WikiRace**



Dosen Pengampu:

Ir. Rila Mandala, M.Eng., Ph.D.

Monterico Adrian, S.T., M.T.

Kelas K3 - Kelompok GoD:

Muhammad Zaki	13522136
---------------	----------

Ahmad Rafi Maliki	13522137
-------------------	----------

Muhammad Dzaki Arta	13522149
---------------------	----------

**Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung**

Daftar Isi

Daftar Isi.....	2
BAB I Deskripsi Tugas.....	3
BAB II Landasan Teori.....	4
2.1. Algoritma.....	4
2.1.1. Penjelajahan Graf.....	4
2.1.2. Iterative Deepening Search (IDS).....	5
2.1.3. Breadth First Search (BFS).....	7
2.2. Aplikasi Web.....	8
BAB III Analisis Pemecahan Masalah.....	9
3.1. Langkah Pemecahan Masalah.....	9
3.1.1. Pemetaan Algoritma IDS.....	9
3.1.2. Pemetaan Algoritma BFS.....	10
3.2. Arsitektur dan Fungsionalitas Web.....	11
3.3. Ilustrasi Kasus.....	12
BAB IV Implementasi dan Pengujian.....	14
4.1. Spesifikasi Teknis.....	14
4.1.1. Struktur Data.....	14
4.1.2. Fungsi.....	15
4.1.3. Prosedur.....	16
4.2. Cara Menggunakan Program.....	16
4.2.1. Cara Instalasi Program.....	16
4.2.2. Cara Menggunakan Program.....	17
4.3. Hasil Pengujian.....	20
4.4. Analisis Hasil Pengujian.....	27
BAB V Kesimpulan.....	28
5.1. Kesimpulan.....	28
5.2. Saran dan Refleksi.....	28
Lampiran.....	29
Daftar Pustaka.....	30

BAB I

Deskripsi Tugas

WikiRace atau Wiki Game adalah permainan yang melibatkan Wikipedia, sebuah ensiklopedia daring gratis yang dikelola oleh berbagai relawan di dunia, dimana pemain mulai pada suatu artikel Wikipedia dan harus menelusuri artikel-artikel lain pada Wikipedia (dengan mengeklik tautan di dalam setiap artikel) untuk menuju suatu artikel lain yang telah ditentukan sebelumnya dalam waktu paling singkat atau klik (artikel) paling sedikit.



Gambar 1. Ilustrasi Graf WikiRace

(Sumber: https://miro.medium.com/v2/resize:fit:1400/1*jxmEbVn2FFWybZsIicJCWQ.png)

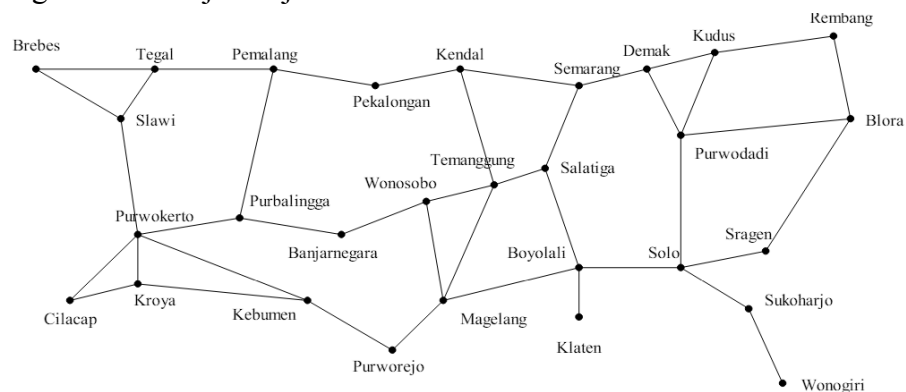
BAB II

Landasan Teori

2.1. Algoritma

2.1.1. Penjelajahan Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.



Gambar 2.1.1.1 Graf yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah.

Penjelajahan graf mengunjungi simpul simpul di dalam graf dengan cara yang sistematis. Penjelajahan graf ada 2 yaitu:

1. Pencarian melebar (Breadth First Search/ BFS)
2. Pencarian mendalam (Depth First Search/DFS)

Graf dapat merupakan representasi persoalan. Penjelajahan graf artinya melakukan pencarian solusi persoalan yang direpresentasikan dengan graf



Gambar 2.1.1 Social Graf

Algoritma pencarian solusi berbasis graf yaitu:

1. Tanpa Informasi (uninformed/blind search)
 - Tidak ada informasi tambahan yang disediakan
 - Contoh: DFS, BFS, Depth Limited Search, Iterative Deepening Search, Uniform, Cost Search
2. Dengan informasi (informed Search)
 - Pencarian solusi berbasis heuristik
 - Mengetahui non-goal state yang “lebih menjanjikan” daripada yang lain
 - Contoh: Best First Search, A*

Dalam pencarian proses solusi terdapat 2 representasi graf yaitu :

1. Graf statis, adalah graf yang sudah terbentuk sebelum proses pencarian dilakukan. Graf direpresentasikan sebagai struktur data
2. Graf dinamis, adalah graf yang terbentuk saat proses pencarian dilakukan. Graf tidak tersedia sebelum pencarian, graf dibangun selama pencarian solusi

2.1.2. Iterative Deepening Search (IDS)

Iterative Deepening Search (IDS) merupakan sebuah algoritma pencarian yang menggabungkan strategi Depth-First Search (DFS) dengan pendekatan peningkatan kedalaman secara bertahap. Algoritma ini dimulai dengan mengatur kedalaman pencarian pada nilai awal, kemudian melakukan serangkaian pencarian dengan batasan kedalaman yang semakin dalam pada setiap iterasinya.

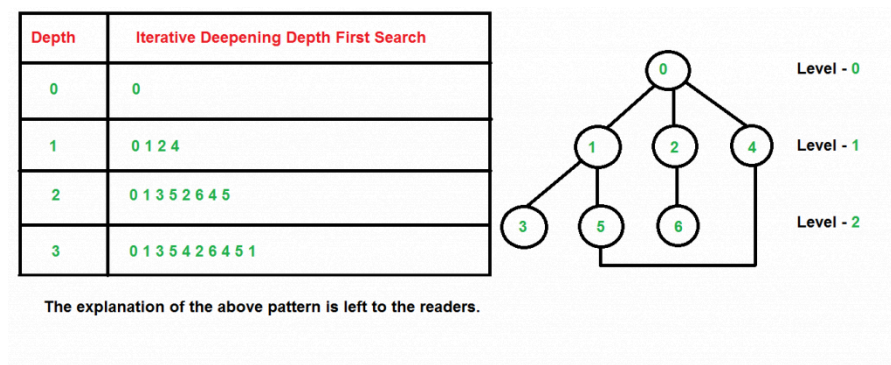
Pada setiap iterasi, IDS menggunakan Depth-Limited Search (DLS) untuk menjelajahi simpul-simpul dalam ruang pencarian hingga kedalaman tertentu. Jika pencarian pada kedalaman tertentu tidak menemukan solusi, IDS meningkatkan kedalaman dan melanjutkan pencarian pada kedalaman yang lebih dalam. Proses ini terus berlanjut hingga solusi ditemukan atau kedalaman maksimum yang ditentukan tercapai. IDS memberikan keuntungan dalam efisiensi penggunaan memori karena hanya perlu menyimpan informasi pencarian untuk satu kedalaman pada satu waktu, sementara masih dapat

mengeksplorasi ruang pencarian secara lebih luas daripada DFS, algoritma dari IDS itu sendiri adalah :

```

Depth ← 0
Iterate
    result ← DLS(problem,depth)
stop: result ≠ cutoff
    depth depth+1
→ result

```



Gambar 2.1.2 Visualisasi IDS

Sumber : <https://www.geeksforgeeks.org>

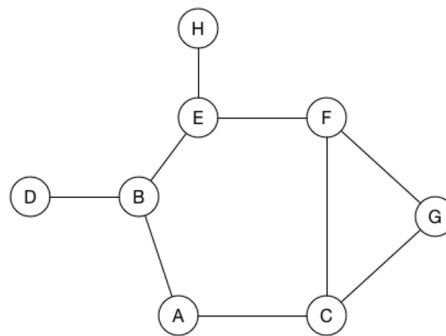
Dilihat dari gambar 2.1.2.1 saat mencapai batas kedalaman tertentu dan belum menemukan solusi, algoritma ini kembali ke simpul awal (root) dan melanjutkan pencarian dari sana dengan menambah kedalaman lagi. Hal ini dapat dianggap sebagai salah satu kelebihan dan keunikan dari algoritma ini, karena dapat mengeksplorasi ruang pencarian secara lebih dalam daripada BFS dalam beberapa kasus.

Salah satu contoh penerapannya yang umum adalah dalam permainan seperti permainan catur, di mana algoritma ini dapat digunakan untuk mencari langkah-langkah terbaik dengan mempertimbangkan serangkaian langkah di masa depan sampai kedalaman tertentu. Meskipun algoritma ini mungkin membutuhkan lebih banyak sumber daya komputasi daripada BFS karena penambahan kedalaman yang berulang, namun dapat memberikan solusi yang lebih optimal dalam beberapa kasus.

2.1.3. Breadth First Search (BFS)

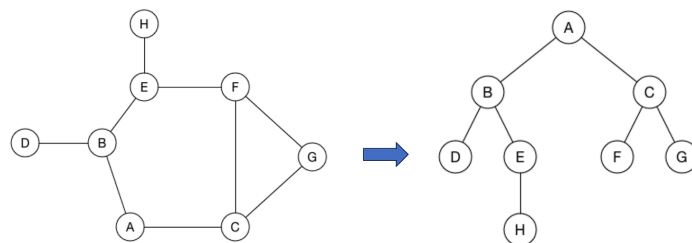
BFS merupakan traversal. dengan algoritma yaitu:

1. Kunjungi simpul v
2. Kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu
3. Kunjungi simpul yang belum dikunjungi yang tadi bertetangga dengan simpul simpul yang tadi dikunjungi, demikian seterusnya.



Gambar 2.1.3 contoh graf BFS

Dapat dilihat dari gambar diatas penjelajahan BFS yang dimulai dari A akan mencari solusi dengan node yang bertetangga dengan A dahulu, yaitu ke B kemudian ke C, Kemudian kita cek node yang bertetangga dengan B dan C dan seterusnya. Maka penjelajahan grafnya akan menjadi seperti $A \rightarrow A,B,C,D,E,F,G,H$.



Gambar 2.1.4 Contoh graf BFS menjadi Pohon BFS

Urutan simpul simpul yang dikunjungi secara BFS adalah $A \rightarrow A,B,C,D,E,F,G,H$.

2.2. Aplikasi Web

Aplikasi web (web app) merupakan program aplikasi yang dijalankan di server jarak jauh dan dapat diakses melalui internet melalui antarmuka browser. Definisi layanan web merujuk pada aplikasi web dan merupakan bagian penting dari banyak situs web modern. Pengembangan aplikasi web menawarkan berbagai manfaat dan digunakan oleh berbagai entitas, mulai dari organisasi besar hingga individu, dengan tujuan yang beragam.

Pengembang merancang aplikasi web untuk menjangkau berbagai pengguna dan kebutuhan. Organisasi menggunakan aplikasi web untuk mengelola data, berkomunikasi dengan anggota tim secara efisien, dan menyediakan layanan kepada pelanggan secara online. Di sisi lain, individu menggunakan aplikasi web untuk berbagai keperluan, seperti berbelanja online, mengakses layanan perbankan, atau mengatur jadwal.

Salah satu keunggulan utama aplikasi web adalah aksesibilitasnya yang luas. Pengguna dapat mengakses aplikasi web ini tanpa perlu mengunduh atau menginstal perangkat lunak tambahan, cukup dengan membuka browser pada perangkat mereka. Hal ini membuat aplikasi web menjadi solusi yang fleksibel dan dapat diakses dari mana saja, selama terhubung ke internet.

BAB III

Analisis Pemecahan Masalah

3.1. Langkah Pemecahan Masalah

Permasalahan yang akan diselesaikan dalam tugas besar 2 ini adalah mencari path dari suatu link wikipedia ke wikipedia yang lain. Untuk melakukan hal tersebut kita perlu melakukan scraping pada website wikipedia untuk mendapatkan link yang ada di halaman tersebut. Untuk memecahkan masalah itu kita menggunakan 2 pendekatan algoritma yaitu IDS dan juga BFS untuk BFS menggunakan metode iteratif dan struktur data queue sedangkan DFS menggunakan metode rekursif dan struktur data stack. Saat sedang melakukan pencarian dengan algoritma BFS dan DFS.

Program juga diimplementasikan melalui website untuk melakukan visualisasi pencarian path. Dalam website ini meminta input dari user untuk memilih lokasi awal dan juga lokasi akhir atau target dari pencarian. Nantinya setelah program selesai mengeksekusi program akan mengeluarkan output berupa waktu eksekusi, rute, jumlah simpul yang di cek, dan juga path.

3.1.1. Pemetaan Algoritma IDS

Proses pencarian dimulai dengan artikel sumber yang ditentukan, dan dari sana, algoritma secara rekursif memeriksa setiap link yang terkait dengan artikel tersebut.

Ketika sebuah link ditemukan, algoritma memeriksa apakah link tersebut mengarah ke artikel tujuan yang ditentukan. Jika iya, pencarian selesai dan jalur dari artikel sumber ke artikel tujuan ditemukan. Namun, jika link tersebut tidak mengarah ke artikel tujuan, algoritma memperdalam pencarian dengan menurunkan kedalaman pencarian dan melakukan rekursi pada link yang baru ditemukan. Proses ini terus berlanjut hingga entah jalur ditemukan atau kedalaman maksimum pencarian tercapai.

Selama proses pencarian, algoritma menggunakan struktur data stack untuk melacak simpul-simpul yang akan dieksplorasi berikutnya. Ketika menemukan sebuah artikel yang tidak sesuai dengan target, algoritma menggunakan backtracking untuk kembali ke simpul sebelumnya dan mencoba link yang lain untuk melanjutkan pencarian.

Dengan pendekatan ini, algoritma IDS dapat menjelajahi ruang pencarian dengan efisien dan menemukan jalur yang memenuhi kriteria pencarian dengan membatasi kedalaman pencarian sesuai yang ditentukan.

3.1.2. Pemetaan Algoritma BFS

Proses pencarian dimulai dengan artikel sumber yang ditentukan, dan dari sana, algoritma secara iteratif memeriksa setiap link yang terkait dengan artikel tersebut.

Selama proses pencarian, algoritma mengekskansi simpul-simpul yang ada dalam queue secara bertahap. Setiap iterasi, algoritma mengambil sejumlah artikel dari queue untuk di-scraping secara paralel. Artikel-artikel ini kemudian di-scraping untuk mendapatkan link-link terkait.

Setelah mendapatkan link-link terkait, algoritma memeriksa apakah artikel tujuan sudah ditemukan di antara link-link tersebut. Jika ya, pencarian selesai dan jalur dari artikel sumber ke artikel tujuan ditemukan. Jika tidak, algoritma memperbarui struktur data parent dan queue dengan informasi baru yang didapat dari hasil scraping.

Proses ini berlanjut hingga artikel tujuan ditemukan atau tidak ada lagi artikel yang tersisa dalam queue untuk dieksplorasi. Saat pencarian selesai, algoritma melakukan backtracking untuk membangun jalur dari artikel tujuan ke artikel sumber berdasarkan informasi parent yang telah tercatat.

3.2. Arsitektur dan Fungsionalitas Web

3.2.1. Front-end

Web ini dibangun dengan menggunakan *framework* [React](#) dan [Tailwind](#) yang ditulis menggunakan bahasa JavaScript sisi front-end nya. Selain itu, web ini juga menggunakan [Axios](#) sebagai HTTP client untuk melakukan fetching data dari server backend. Secara lokal, front-end dijalankan pada alamat <http://localhost:5173/>.

3.2.2. Back-end

Server back-end dibuat menggunakan bahasa [Go](#) dengan *framework* [Gin](#) sebagai web frameworknya. Selain itu digunakan juga [GoColly](#) sebagai *framework* untuk scrapping. Secara lokal, server back-end dijalankan pada alamat <http://localhost:8080/>. Server back-end memiliki 3 endpoint yaitu <http://localhost:8080/api/status> untuk mengecek response, <http://localhost:8080/api/BFS> untuk melakukan penjelajahan wiki race dengan algoritma BFS dan juga <http://localhost:8080/api/IDS> untuk melakukan penjelajahan wiki race dengan algoritma IDS.

3.2.3. Fungsionalitas

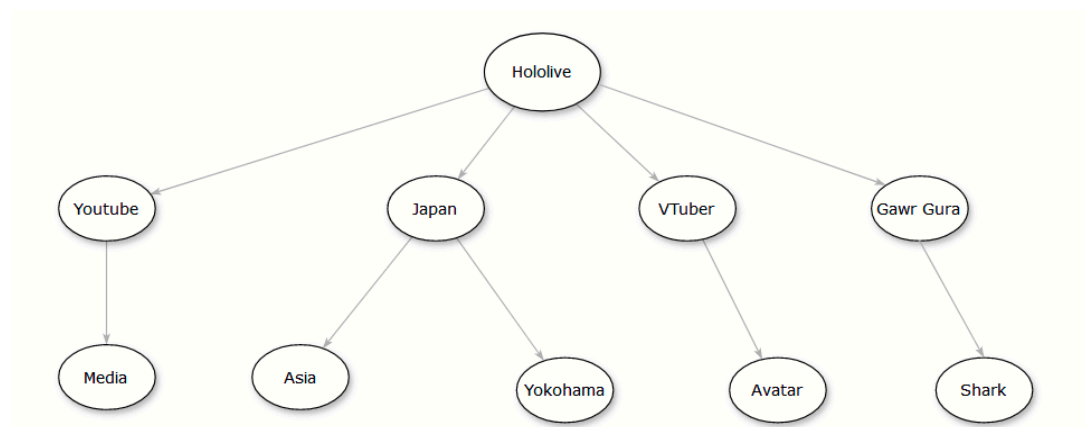
Web ini dapat digunakan untuk melakukan penjelajahan path pada permainan wiki race berdasarkan masukan dari user.

Web ini dapat menerima masukan <source wiki> sebagai sumber penjelajahan dan juga <target wiki> sebagai target penjelajahan permainan wiki race. Web juga menyediakan <opsi> penjelajahan berdasarkan algoritma BFS atau IDS yang berupa *bulletin check*.

Setelah pengguna memasukkan ketiga masukan tersebut. Pengguna dapat menekan tombol “Generate Path” untuk melakukan pencarian. Setelah pencarian selesai akan ditampilkan waktu yang dibutuhkan, halaman wikipedia yang dijelajahi, serta jalur penjelajahan dari *source* ke *target*.

3.3. Ilustrasi Kasus

Untuk memperjelas analisis pemecahan masalah, kami akan melakukan ilustrasi kasus dengan penyelesaiannya. Misal pada kasus dimana kami ingin mencari path dari Hololive ke Shark dengan graf berbentuk sebagai berikut :



Gambar 3.3.1 Ilustrasi Kasus

Untuk lebih jelas akan kami gunakan tabel untuk mengilustrasikan pencarian menggunakan BFS dan juga IDS

TABEL BFS

Head	Queue
Hololive	Youtube,Japan,VTuber,Gawr Gura
Youtube	Japan,Vtuber,Gawr Gura, Media
Japan	Vtuber, Gawr, Gura, Media, Asia, Yokohama
VTuber	Gawr Gura, Media, Asia, Yokohama,Avatar
Gawr Gura	Media, Asia, Yokohama,Avatar, Shark
Media	Asia, Yokohama,Avatar, Shark
Asia	Yokohama,Avatar, Shark
Yokohama	Avatar, Shark
Avatar	Shark
Shark	

TABEL IDS

Head	Stack
Depth 0	
Hololive	
Depth 1	
Youtube	Hololive
Japan	Hololive
VTuber	Hololive
Gawr Gura	Hololive
Depth 2	
Media	Hololive, Yotube
Asia	Hololive, Japan
Yokohama	Hololive, Japan
Avatar	Hololive, VTuber
Shark	Hololive, Gawr Gura

BAB IV

Implementasi dan Pengujian

4.1. Spesifikasi Teknis

4.1.1. Struktur Data

Dalam program ini kami menggunakan beberapa struktur data dengan tipe sendiri untuk memudahkan dalam mengelola data tersebut

```
type Wiki struct {
    Title string
    URL    string
}

type WikiTitle struct {
    Title string
    Parent string
}

type Duration struct {
    Hours    int
    Minutes  int
    Seconds  int
    Milliseconds int
}
```

1. Wiki: Tipe data Wiki adalah sebuah struktur yang digunakan untuk merepresentasikan informasi tentang suatu artikel Wikipedia. Setiap objek Wiki memiliki dua atribut, yaitu Title yang merupakan judul artikel dan URL yang merupakan URL artikel tersebut. Struktur ini berguna untuk menyimpan informasi terkait artikel Wikipedia dalam format yang mudah diakses dan diproses dalam program.
2. WikiTitle: Tipe data WikiTitle adalah struktur yang mirip dengan tipe data Wiki, namun hanya menyimpan informasi tentang judul artikel dan parent dari artikel tersebut. Atribut Title digunakan untuk menyimpan judul artikel, sedangkan atribut Parent digunakan untuk menyimpan judul artikel yang menjadi parent dari artikel tersebut. Tipe data ini berguna dalam konteks algoritma pencarian jalur atau struktur data yang memerlukan informasi tentang hubungan antar artikel.
3. Duration: Tipe data Duration adalah sebuah struktur yang digunakan untuk merepresentasikan durasi waktu dalam format jam, menit, detik, dan milidetik. Setiap objek Duration memiliki empat atribut, yaitu Hours, Minutes, Seconds, dan Milliseconds, yang masing-masing merepresentasikan bagian-bagian dari durasi waktu tersebut. Struktur ini berguna untuk merepresentasikan durasi

waktu dalam pemrograman, misalnya untuk mengukur waktu eksekusi suatu operasi atau proses.

4.1.2. Fungsi

Nama	Penjelasan
FormatDuration(duration time.Duration)	Digunakan untuk menghasilkan format waktu yang sesuai, fungsi ini mengembalikan sebuah string
IsValidWiki(title string)	Digunakan untuk memvalidasi apakah link akan dituju merupakan link yang valid atau tidak, fungsi ini mengembalikan sebuah boolean
ConstructURL(title string)	Digunakan untuk membangun URL dari title, fungsi ini mengembalikan string
Scrap(urls []string)	Digunakan untuk mengambil link yang sesuai dari url wikipedia, fungsi ini mengembalikan sebuah array of WikiTitle yang merupakan kumpulan dari link yang sudah diambil dan juga boolean yang akan bernilai true jika tidak ada kesalahan dan false jika terjadi kesalahan
EntryPoint(source_wiki, target_wiki string)	Digunakan untuk awal mula masuk ke fungsi IDS maupun BFS, fungsi ini mengembalikan hasil dari pencarian tersebut berupa path yaitu array of Wiki, durasi pencarian yang bertipe string, dan juga banyak link yang dicari.
IDS(sourceTitle string, targetURL string, depth int, path []utils.Wiki)	Digunakan untuk melakukan pencari menggunakan algoritma IDS dan mengembalikan boolean found untuk cek apakah sudah ketemu atau belum, lalu path bertipe array of wiki, banyak link yang dicari, dan juga boolean err untuk cek apakah ada kesalahan atau tidak
BREADTH_FIRST_SEARCH(sour ce, target string)	Digunakan untuk menggunakan algoritma BFS dan mengembalikan path bertipe array of wiki, banyak link yang dicari, dan juga boolean

	untuk cek apakah terjadi error atau tidak
--	---

4.1.3. Prosedur

Nama	Penjelasan
printPath(path []utils.Wiki)	Digunakan untuk menampilkan path dalam terminal

4.2. Cara Menggunakan Program

4.2.1. Cara Instalasi Program

1. Clone Repository github GoD

```
git clone https://github.com/rafimaliki/Tubes2_GoD
```

2. Masuk ke root *directory*

```
cd Tubes2_GoD
```

3. setup **BACKEND**

```
cd src/backend
go run main.go
```

“go run main.go” disini buat nyalain backend nya
Jika berhasil di run maka akan ada prompt dan muncul “Listening and serving HTTP on :8080” pada terminal

4. setup **FRONTEND**

Buka terminal baru di root *directory* repository

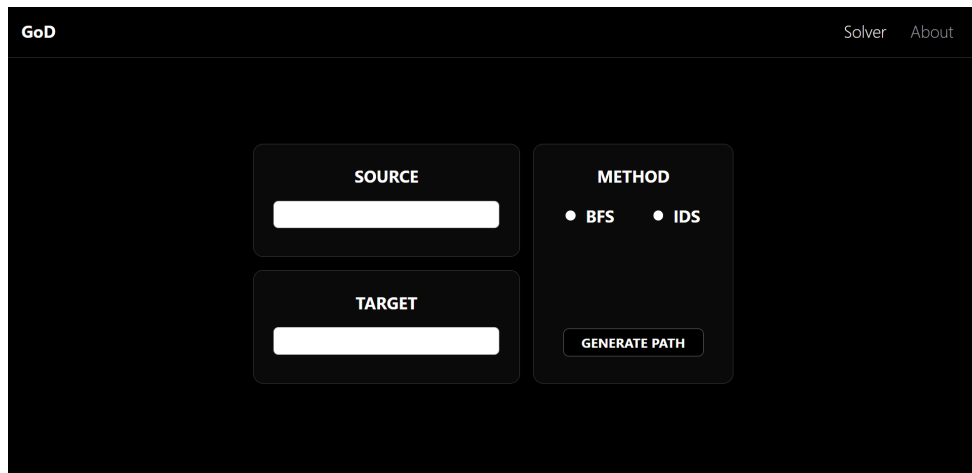
```
cd src/frontend
npm i
npm run dev
```

“npm i” disini buat install node module (ini cuman sekali aja di awal)
“npm run dev” disini buat jalanin front end

akses front-end di <http://localhost:5173/> pada browser

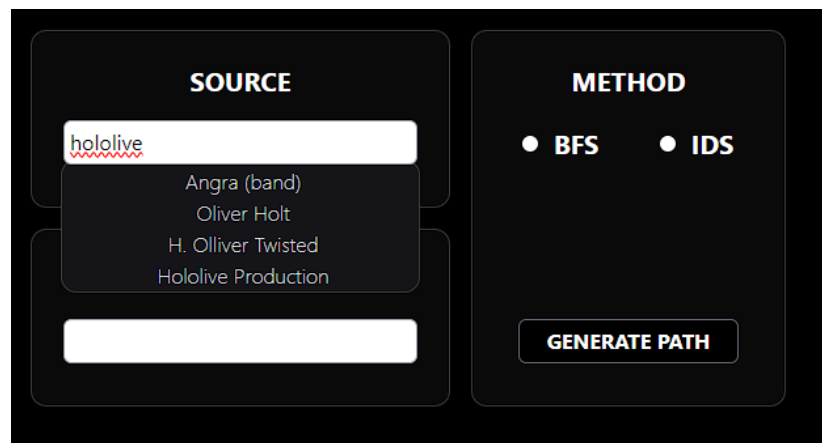
4.2.2 Cara Menggunakan Program

Setelah program dijalankan maka website akan mempunyai tampilan



Gambar 4.2.1 Tampilan Website Wikirace GoD

1. Masukan Website wikipedia awal pada bagian source. Jika masukan tidak spesifik maka program akan menampilkan website yang mirip dengan masukan



Gambar 4.2.2 Masukan website wikipedia awal

2. Masukan target website wikipedia pada bagian target. Jika masukan tidak spesifik maka program akan menampilkan website yang mirip dengan masukan

The interface consists of three main sections: SOURCE, TARGET, and METHOD. The SOURCE section has a text input field containing 'Hololive Production'. The TARGET section has a text input field containing 'shar' and a dropdown menu showing suggestions: 'Sharia', 'Shark', 'Shar', 'Shah Rukh Khan', and 'SharePoint'. The METHOD section has two radio buttons, 'BFS' and 'IDS', both of which are currently unselected. A 'GENERATE PATH' button is located below the radio buttons.

Gambar 4.2.3 Masukan target website pada

3. Pilih metode apa untuk menjalankan program pada bagian Method. Ketika dipilih maka akan ada bulat hitam menandakan method tersebut sudah terpilih

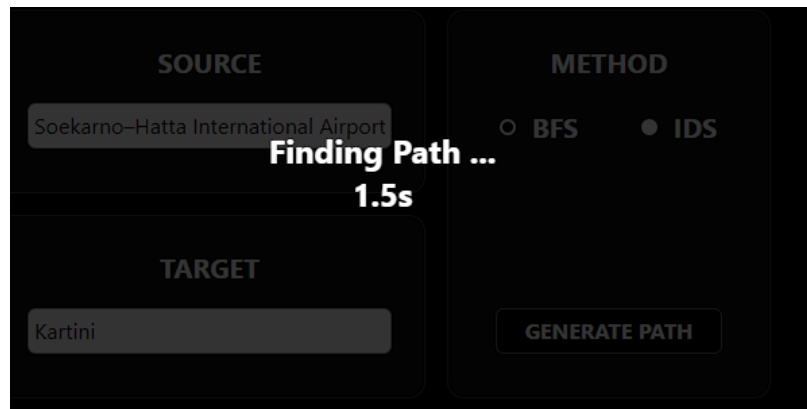
The interface is similar to the previous one, but the 'TARGET' field now contains 'Shark'. The 'METHOD' section still shows unselected radio buttons for 'BFS' and 'IDS'. The 'GENERATE PATH' button remains visible.

Gambar 4.2.4 Sebelum memilih method

In this state, the 'BFS' radio button in the METHOD section is now selected, indicated by a filled circle. The 'IDS' radio button remains unselected. The 'GENERATE PATH' button is still present.

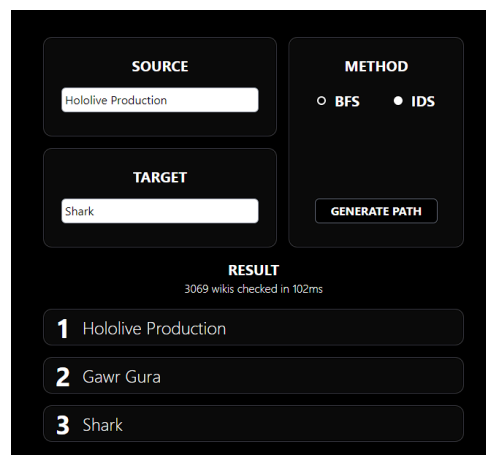
Gambar 4.2.5 Sesudah memilih method (method BFS)

4. Untuk Mencari solusi maka kita harus memencet tombol “GENERATE PATH”.



Gambar 4.2.6 Tampilan program saat mencari solusi

5. Setelah program selesai mencari solusi. Maka akan terdapat jumlah website wikipedia yang dijelajahi, durasi waktu penjelajahan, dan *path* hasil solusi.



Gambar 4.2.7 Tampilan website setelah program mencari solusi

4.3. Hasil Pengujian

4.3.1. Degree 1

4.3.1.1. Joko Widodo → Indonesia

BFS

The screenshot shows a web interface for a Breadth-First Search (BFS) algorithm. It has a dark theme. On the left, there are two input fields: 'SOURCE' with 'Joko Widodo' and 'TARGET' with 'Indonesia'. On the right, there is a 'METHOD' section with two radio buttons: 'BFS' (selected) and 'IDS'. Below the inputs is a 'GENERATE PATH' button. The 'RESULT' section shows the path: '1 Joko Widodo' and '2 Indonesia'. Below the path, it says '472 wikis checked in 20ms'.

Method	Source	Target	Result
BFS	Joko Widodo	Indonesia	1 Joko Widodo 2 Indonesia 472 wikis checked in 20ms

IDS

The screenshot shows a web interface for an Iterative Deepening Search (IDS) algorithm. It has a dark theme. On the left, there are two input fields: 'SOURCE' with 'Joko Widodo' and 'TARGET' with 'Indonesia'. On the right, there is a 'METHOD' section with two radio buttons: 'BFS' (selected) and 'IDS' (selected). Below the inputs is a 'GENERATE PATH' button. The 'RESULT' section shows the path: '1 Joko Widodo' and '2 Indonesia'. Below the path, it says '1152 wikis checked in 865ms'.

Method	Source	Target	Result
BFS	Joko Widodo	Indonesia	1 Joko Widodo 2 Indonesia 1152 wikis checked in 865ms

4.3.1.2. Hololive Production → Youtube

BFS

SOURCE

TARGET

METHOD
☐ BFS ☒ IDS

GENERATE PATH

RESULT
20 wikis checked in 6ms

1 Hololive Production

2 YouTube

IDS

SOURCE

TARGET

METHOD
☒ BFS ☐ IDS

GENERATE PATH

RESULT
135 wikis checked in 80ms

1 Hololive Production

2 YouTube

4.3.2. Degree 2

4.3.2.1. Government → Corruption

BFS

The screenshot shows a web interface for a Breadth-First Search (BFS) algorithm. It has a dark theme. On the left, there are two input fields: 'SOURCE' with the value 'Government' and 'TARGET' with the value 'Corruption'. On the right, there is a 'METHOD' section with two radio buttons: 'BFS' (selected) and 'IDS'. Below the radio buttons is a 'GENERATE PATH' button. At the bottom, the 'RESULT' section shows '1481 wikis checked in 224ms' and a list of three items: '1 Government', '2 World Governance Index', and '3 Corruption'.

Source	Method	Target	Result
Government	BFS	Corruption	1481 wikis checked in 224ms

- 1 Government
- 2 World Governance Index
- 3 Corruption

IDS

The screenshot shows a web interface for an Iterative Deepening Search (IDS) algorithm. It has a dark theme. On the left, there are two input fields: 'SOURCE' with the value 'Government' and 'TARGET' with the value 'Corruption'. On the right, there is a 'METHOD' section with two radio buttons: 'BFS' (selected) and 'IDS'. Below the radio buttons is a 'GENERATE PATH' button. At the bottom, the 'RESULT' section shows '1623 wikis checked in 455ms' and a list of three items: '1 Government', '2 World_Governance_Index', and '3 Corruption'.

Source	Method	Target	Result
Government	BFS	Corruption	1623 wikis checked in 455ms

- 1 Government
- 2 World_Governance_Index
- 3 Corruption

4.3.2.2. Hololive Production → Shark

BFS

SOURCE

TARGET

METHOD
☐ BFS ☒ IDS

GENERATE PATH

RESULT
8103 wikis checked in 832ms

- 1 Hololive Production
- 2 Gawr Gura
- 3 Shark

IDS

SOURCE

TARGET

METHOD
☒ BFS ☐ IDS

GENERATE PATH

RESULT
238 wikis checked in 668ms

- 1 Hololive Production
- 2 Gawr_Gura
- 3 Shark

4.3.3. Degree 3

4.3.3.1. Hololive Production → Chikamatsu Monzaemon

BFS

The screenshot shows a web interface for a search algorithm. It has four main sections: SOURCE, METHOD, TARGET, and RESULT. The SOURCE field contains 'Hololive Production' and the TARGET field contains 'Chikamatsu Monzaemon'. The METHOD section has two radio buttons: 'BFS' (selected) and 'IDS'. A 'GENERATE PATH' button is located below the METHOD section. The RESULT section displays the search progress: '222423 wikis checked in 48s 423ms'. Below this, a list of four items is shown, each with a number and a name: 1 Hololive Production, 2 Japanese language, 3 Early Modern Japanese, and 4 Chikamatsu Monzaemon.

Source	Method	Target	Result
Hololive Production	<input checked="" type="radio"/> BFS <input type="radio"/> IDS	Chikamatsu Monzaemon	222423 wikis checked in 48s 423ms
1 Hololive Production			
2 Japanese language			
3 Early Modern Japanese			
4 Chikamatsu Monzaemon			

IDS

The screenshot shows a web interface for a search algorithm. It has four main sections: SOURCE, METHOD, TARGET, and RESULT. The SOURCE field contains 'Hololive Production' and the TARGET field contains 'Chikamatsu Monzaemon'. The METHOD section has two radio buttons: 'BFS' (selected) and 'IDS' (unselected). A 'GENERATE PATH' button is located below the METHOD section. The RESULT section displays the search progress: '1636 wikis checked in 20s 216ms'. Below this, a list of four items is shown, each with a number and a name: 1 Hololive Production, 2 Minato,_Tokyo, 3 Amagasaki, and 4 Chikamatsu_Monzaemon.

Source	Method	Target	Result
Hololive Production	<input checked="" type="radio"/> BFS <input type="radio"/> IDS	Chikamatsu Monzaemon	1636 wikis checked in 20s 216ms
1 Hololive Production			
2 Minato,_Tokyo			
3 Amagasaki			
4 Chikamatsu_Monzaemon			

4.3.3.2. Joko Widodo → King Zhou of Shang

BFS

SOURCE
Joko Widodo

TARGET
King Zhou of Shang

METHOD
☐ BFS ☒ IDS
GENERATE PATH

RESULT
464502 wikis checked in 1m 7s 79ms

- 1 Joko Widodo
- 2 Given name
- 3 Chinese name
- 4 King Zhou of Shang

IDS

SOURCE
Joko Widodo

TARGET
King Zhou of Shang

METHOD
☒ BFS ☐ IDS
GENERATE PATH

RESULT
2255 wikis checked in 1m 18s 217ms

- 1 Joko Widodo
- 2 Indonesian_names
- 3 Chinese_name
- 4 King_Zhou_of_Shang

4.3.4. Degree 4

4.3.4.1. Ivar Nosy → Shark

BFS

The screenshot shows a web application interface for finding a path between two entities. It has a dark theme. The 'SOURCE' field contains 'Ivar Nosy' and the 'TARGET' field contains 'Shark'. The 'METHOD' section has 'BFS' selected. A 'GENERATE PATH' button is visible. The 'RESULT' section shows the path: 1 Ivar Nosy, 2 Haakon IV of Norway, 3 Tunis, 4 Tertiary industry, 5 Shark. It also indicates that 82346 wikis were checked in 13s 750ms.

SOURCE	TARGET	METHOD
Ivar Nosy	Shark	BFS

RESULT
82346 wikis checked in 13s 750ms

- 1 Ivar Nosy
- 2 Haakon IV of Norway
- 3 Tunis
- 4 Tertiary industry
- 5 Shark

4.3.4.2. Klára Langhoffer-Pétervári → Portuguese Roller Hockey First Division

BFS

The screenshot shows the same web application interface as above, but with 'Klára Langhoffer-Pétervári' in the 'SOURCE' field and 'Portuguese Roller Hockey First Division' in the 'TARGET' field. The 'METHOD' is still 'BFS'. The 'GENERATE PATH' button is visible. The 'RESULT' section shows the path: 1 Klára Langhoffer-Pétervári, 2 Sports Reference, 3 International Olympic Committee, 4 World Skate, 5 Portuguese Roller Hockey First Division. It also indicates that 477980 wikis were checked in 1m 883ms.

SOURCE	TARGET	METHOD
Klára Langhoffer-Pétervári	Portuguese Roller Hockey First Division	BFS

RESULT
477980 wikis checked in 1m 883ms

- 1 Klára Langhoffer-Pétervári
- 2 Sports Reference
- 3 International Olympic Committee
- 4 World Skate
- 5 Portuguese Roller Hockey First Division

4.4. Analisis Hasil Pengujian

Dapat dilihat dari hasil pengujian bahwa algoritma BFS lebih cepat dibanding algoritma IDS hal ini dapat dikarenakan sifat dari BFS yaitu mempelajari semua simpul pada depth yang sama terlebih dahulu sebelum lanjut ke depth selanjutnya. Karena biasanya solusi wiki ditemukan pada depth yang awal maka BFS cenderung lebih optimal dibanding IDS. Sedangkan IDS dapat menghabiskan banyak waktu untuk menjelajahi depth yang lebih dalam secara iteratif. Hal ini bertentangan dengan sifat wikirace yang seperti pohon yang memiliki banyak cabang sehingga solusinya cenderung berada pada depth yang lebih rendah.

Dalam hal pencarian pada kedalaman yang ke 4 IDS kami membutuhkan waktu yang terlalu lama untuk mencari, hal ini mungkin disebabkan oleh algoritma IDS yang hanya bisa memungkinkan satu pencarian dalam satu waktu, kelompok kami memutuskan untuk tidak menggunakan concurrency atau menggunakan go routine untuk menjaga struktur atau isi dalam stack merupakan benar benar sesuai dengan IDS yang diajarkan di kelas yaitu tidak mendahului node tetangga lain sebelum node awal selesai dicari atau selesai dijalankan.

Dari hasil yang diperoleh kita bisa melihat bahwa algoritma pencarian menggunakan BFS mampu mencapai degree 4 dengan cepat. Hal ini disebabkan oleh pemanfaatan teknik konkurensi (concurrency) yang mempercepat proses pencarian. Dengan menggunakan konkurensi, algoritma dapat mengirimkan beberapa permintaan tautan sekaligus untuk diambil informasinya (scraping), sehingga tidak perlu menunggu satu node selesai dijalankan sebelum memulai yang lain. Dengan demikian, konkurensi memungkinkan algoritma BFS untuk secara efisien menjelajahi graf yang lebih dalam dengan cepat, meningkatkan efektivitas pencarian menuju degree.

Dari pengujian yang dilakukan kelompok kami dapat dengan cepat mencari apabila sudah dilakukan caching terlebih dahulu. Caching diperlukan agar tidak perlu lagi meminta request pada wikipedia yang menurut kami itu salah satu faktor yang menyebabkan lamanya pencarian. Dengan adanya caching kita hanya perlu melihat data yang sudah ada jadi tidak perlu lagi mengambil data dari wikipedia.

BAB V

Kesimpulan

5.1. Kesimpulan

BFS merupakan algoritma yang lebih optimal dibanding IDS dalam mencari solusi Wikirace. Hal ini dikarenakan sifat wikirace sendiri seperti pohon yang memiliki banyak cabang sehingga solusi akan cenderung lebih berada pada depth yang lebih rendah. BFS mencari solusi di depth yang sama terlebih dahulu sebelum lanjut ke depth selanjutnya maka BFS akan menjadi algoritma yang lebih optimal dibanding dengan IDS yang mencari solusi pada depth yang lebih dalam dahulu secara iteratif.

5.2. Saran dan Refleksi

Dalam pengerjaan tugas besar ini, terdapat beberapa kendala seperti memilih package mana yang baik untuk melakukan scraping, bagaimana agar terhindar dari timeout oleh wikipedia karena melakukan banyak *request* dalam waktu yang berdekatan, penggunaan bahasa pemrograman Go yang belum pernah diajarkan. Namun kelompok kami berusaha melakukan eksplorasi sebanyak mungkin dan mendalaminya sehingga akhirnya tugas ini dengan lancar.

Sebagai improve dimasa depan mungkin kami akan menggunakan metode caching yang lebih teroptimisasi, agar tidak membutuhkan banyak ruang pada *storage*, dan juga menggunakan metode scraping yang lebih efisien untuk mengoptimalkan waktu pencarian.

Lampiran

Link Repository : https://github.com/rafimaliki/Tubes2_GoD

Daftar Pustaka

- Munir, Rinaldi & Maulidevi, Nur Ulfa (2021). *Breadth / Depth First Search- Bag1*. Diakses pada 27 April 2024 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/BFS-DFS-2021-Bag1-2024.pdf>
- Munir, Rinaldi & Maulidevi, Nur Ulfa (2021). *Breadth / Depth First Search- Bag2*. Diakses pada 27 April 2024 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/BFS-DFS-2021-Bag2-2024.pdf>
- techtarge (2023). *What is web application?*. Diakses pada 27 April 2024 dari <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>