

Implementasi Algoritma BFS dan DFS dalam Pencarian Rute Angkutan Umum di Bandung Utara

Almeiza Arvin Muzaki (13519066)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519066@std.stei.itb.ac.id

Abstract—Angkutan Kota (Angkot) merupakan alternatif moda transportasi di Kota Bandung yang cukup berpotensi untuk diberdayakan lebih lanjut. Sistem rute absolut yang diberlakukan pada angkot sebagai “jurusan”. Angkot-angkot berbagai jurusan ini banyak memiliki irisan rute sehingga terdapat banyak kombinasi rute perjalanan yang dapat dilakukan dengan angkot. Selain itu, rute angkot dapat direpresentasikan dalam bentuk graf, sedangkan graf dapat diperlakukan dengan operasi pencarian rute. Dengan kata lain, persoalan pemilihan jurusan angkot dalam perjalanan dapat dipecahkan dengan persoalan pencarian rute pada graf. Makalah ini membahas implementasi dari algoritma BFS dan DFS dalam pencarian rute angkot melalui pencarian rute graf angkot.

Keywords—Algoritma, Angkutan, Kota, Angkot, BFS, DFS, Rute

I. PENDAHULUAN

Dalam keseharian manusia yang begitu padat, manusia terus mengupayakan moda transportasi yang memadai guna memenuhi kebutuhan mobilitas. Salah satu transportasi tersebut adalah angkutan umum (angkot). Setelah sekian lama beroperasi, mengingat beberapa keunggulan penggunaan angkot, moda transportasi ini tetap memikat banyak kalangan.

Angkot berjalan menggunakan sistem rute. Satu rute terdiri dari beberapa jalan. Rute yang dominan dilalui oleh banyak orang dirumuskan menjadi sebuah “jurusan”. Semua angkot yang berjurusan sama akan hampir selalu sama setiap waktunya.

Dengan diberlakukannya sistem rute pada angkot, terpecahkanlah masalah angkutan berkapasitas tinggi, tetapi timbul masalah baru perihal keterbacaan dan kebermanfaatan rute. Beberapa orang tidak paham bagaimana menyesuaikan rute angkot dengan rute perjalanan yang mereka kehendaki. “Bagaimana cara saya mencapai lokasi A dari lokasi B?”. “Dimana saja kah saya harus berganti jurusan angkot?”. “Dua pertanyaan tersebut adalah dua pertanyaan yang paling sering terdengar dalam dunia perangkotan.

Jika ditinjau kembali, kita akan dapat melihat beberapa kesamaan rute angkot dengan persoalan rute pada graf. Cara mencapai lokasi B dari lokasi A ekuivalen dengan mencari jalur antara titik B dari titik A pada graf. Dimana saja kah letak

pergantian rute yang perlu dilakukan dapat disaksikan dari hasil pencarian jalur dari tahap sebelumnya. Dengan demikian, persoalan rute angkot ini dapat diselesaikan dengan persoalan graf. Tentunya persoalan pengambilan informasi dari graf juga tidak lepas dari algoritma yang menaunginya.

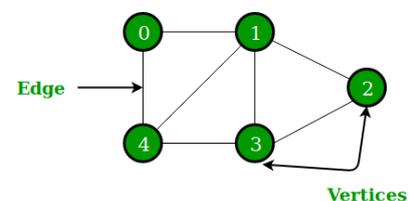
Algoritma pemrosesan graf yang akan diuji pada studi kasus ini adalah algoritma BFS dan DFS. Kedua algoritma ini cukup sering dipertandingkan satu sama lain, mengingat kedua algoritma ini saling mengisi satu sama lain. Jika BFS melakukan pencarian jalur secara “melebar” satu persatu, DFS melakukan pencarian jalur secara “mendalam”.

Pada makalah ini, penulis akan menguraikan implementasi dan hasil penggunaan algoritma BFS dan DFS pada persoalan pencarian rute angkot, khususnya pada wilayah Bandung Utara.

II. DASAR TEORI

A. Graf

Graf merupakan struktur data yang dinotasikan sebagai $G = (V, E)$, dengan $V = \{v_1, v_2, \dots, v_n\}$, himpunan tidak kosong dari beberapa simpul, dan $E = \{e_1, e_2, \dots, e_n\}$, himpunan sisi yang menghubungkan dua buah simpul.



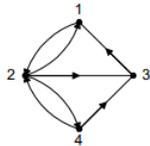
Gambar 2.A.1. Struktur data graf sederhana (geeksforgeeks.com)

Pada Gambar 2.A.1, dapat diperoleh informasi $V = \{0,1,2,3,4\}$ dan $E = \{(0,4), (0,1), (3,4), (1,4), (1,3), (1,2), (1,4)\}$.

Graf dapat dikelompokkan menurut keberadaan gelang dan sisi ganda. Gelang merupakan edge yang menghubungkan suatu vertice dengan dirinya sendiri sehingga terbentuk sebuah loop edge, sedangkan sisi ganda merupakan sisi berjumlah dua atau lebih yang mengikat 2 vertices sekaligus.

1. Graf Sederhana, tidak mengandung gelang maupun sisi ganda seperti Gambar 2.A.1.
2. Graf Tak Sederhana, mengandung sisi ganda atau gelang
 - 2.1. Graf Ganda, mengandung sisi ganda
 - 2.2. Graf Semu, mengandung gelang

Selain itu, terdapat graf yang mengandung orientasi arah yang disebut sebagai graf berarah. Graf ini setiap sisinya diberikan arah dalam bentuk panah dari suatu vertice ke vertice lainnya.



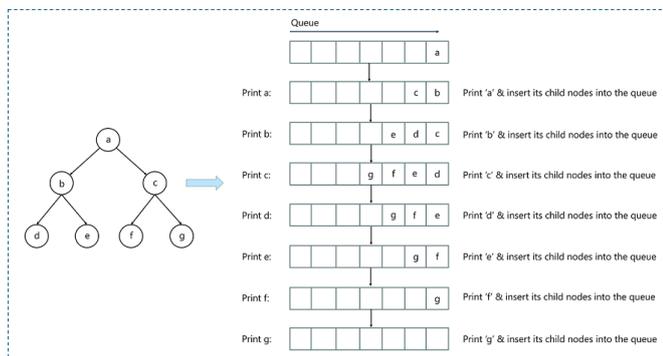
Gambar 2.A.2. Ilustrasi graf berarah (informatika.stei.itb.ac.id)

B. Breadth First Search

Algoritma Breadth First Search (BFS) termasuk algoritma traversal graf yang artinya mengunjungi vertices-vertices graf secara sistematis. Algoritma ini juga termasuk algoritma tanpa informasi yang artinya tidak ada informasi tambahan yang terlibat dalam sistematisa penelusuran vertices.

BFS merupakan algoritma dengan pencarian traversal melebar. Pengecekan jalur dilakukan dengan membangkitkan seluruh vertice tetangga terlebih dahulu, dilanjutkan dengan pengunjangan tetangga-tetangga tadi untuk dilakukan proses yang sama, demikian seterusnya. Algoritma tersebut adalah sebagai berikut:

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



Gambar 2.B.1. Ilustrasi langkah-langkah BFS (edureka.co)

Beberapa struktur data yang diperlukan dalam pemrosesan BFS diantaranya:

1. Matriks ketetanggaan $A = [a_{ij}]$ yang berukuran $n \times n$, dengan $a_{ij} = 1$ jika vertice i dan j saling bertetangga dan $a_{ij} = 0$ jika vertice i dan j tidak saling bertetangga.
2. Queue untuk menyimpan seluruh simpul yang telah dikunjungi.
3. Array of Boolean. True jika vertice i sudah dikunjungi dan False jika vertice i belum dikunjungi.

```

Procedure BFS (input v : integer) {
input : simpul awal
output : semua simpul yang dikunjungi
}

Deklarasi
w : integer
q : queue
procedure MakeQueue(input/output q : queue){ membuat queue kosong}
procedure InsertQueue(input/output q: queue, input v : integer)
{memasukkan v ke dalam q dari posisi terbelakang}
procedure DelQueue (input/output q: queue, input v : integer) {menghapus v dari q}

Algoritma
MakeQueue(q)
Output(v)
Dikunjungi[v] <- True

InsertQueue(q,v)

While not EmptyQueue(q) do
  DelQueue(q,v)
  For every w neighbour of v do
    If not dikunjungi[w]then
      Write(w)
      InsertQueue(q,w)
      Dikunjungi[w] <- true
  
```

C. Depth First Search

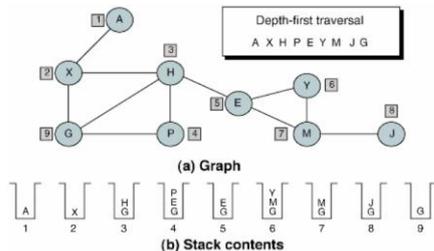
Sebagaimana BFS, DFS juga merupakan algoritma pencarian traversal graf dan tanpa informasi.

Akan tetapi, berkebalikan dengan BFS, DFS merupakan algoritma traversal mendalam. Tetangga yang dibangkitkan hanyalah satu dan dipanggil kembali tetangga dari tetangganya (tetangga level berikutnya).

Algoritma tersebut adalah sebagai berikut:

1. Kunjungi vertice v
2. Kunjungi vertice w yang bertetangga dengan v .
3. Ulangi DFS mulai dari vertice w .

- Ketika mencapai vertice u sedemikian sehingga semua vertice yang bertetangga dengannya telah dikunjungi, pencarian diruntut-balik (backtrack) ke simpul terakhir yang dikunjungi sebelumnya dan mempunyai simpul w yang belum dikunjungi.
- Pencarian berakhir bila tidak ada lagi simpul yang belum dikunjungi yang dapat dicapai dari simpul yang telah dikunjungi.



Gambar 2.C.1. Ilustrasi langkah-langkah DFS (stackoverflow.com)

```

Procedure DFS (input v:integer){
Input : v, vertice awal kunjungan
Output : Semua simpul yang dikunjungi
}
Deklarasi
W : integer
Algoritma
Write(v)
Dikunjungi[v] <-true
W traversal [1..n]
If A[v,w] = 1 then
If not dikunjungi[w] then
DFS(w)

```

D. Perbandingan Algoritma BFS dan DFS

Berdasarkan notasi algoritma serta ilustrasi yang telah dijabarkan sebelumnya, terdapat beberapa perbedaan karakteristik antara BFS dan DFS.

- Performa BFS akan lebih baik apabila vertice target berada tidak jauh dari vertice awal, sedangkan performa DFS akan lebih baik apabila vertice target dan vertice awal saling berjauhan.
- BFS lebih lambat dan lebih kompleks dibandingkan DFS, tetapi menjamin perolehan jalur terpendek.
- DFS lebih cepat dibandingkan BFS, tetapi jumlah vertice yang dikunjungi akan jauh lebih banyak dibandingkan BFS dan tidak menjamin akan diperolehnya jalur terpendek.
- Kompleksitas BFS dan DFS sama-sama bernilai $O(V+E)$.

III. PEMBAHASAN

A. Implementasi

Pemilihan wilayah Bandung Utara disebabkan oleh tingginya kegiatan lalu lintas yang terjadi sepanjang harinya yang cukup tinggi dibandingkan wilayah Bandung lainnya.

Setiap angkot memiliki dua rute yang saling berkebalikan, tetapi belum tentu sama persis, mengingat adanya beberapa jalan searah di Bandung Utara.

Pendataan rute ini juga memperhatikan beberapa aspek. Pertama, hanya jalanan yang berada di wilayah Bandung Utara saja yang dilibatkan sebagai rute. Kedua, hanya jurusan-jurusan yang relatif mengangkut banyak penumpang saja yang diikutsertakan sebab ada beberapa jurusan yang sepi penumpang seperti Bumi Panyileukan-Sekemirung.

Setelah dilakukan pendataan, hasil rekap jurusan serta rute seluruh angkot yang melalui jalanan wilayah Bandung Utara dalam representasi array adalah sebagai berikut:

(Pembacaan : angkot bergerak dari $V_1, V_2 \dots$ hingga V_n)

1. Abdul Muis-Cicaheum

["Supratman", "Katamso", "Pahlawan", "Cikutra", "Surapati"]

["Surapati", "Cikutra", "Pahlawan", "Katamso", "Supratman"]

2. Abdul Muis-Ledeng

["Rivai", "Cipaganti", "Setiabudi", "Karang Sari", "Sukajadi", "Setiabudi"]

["Setiabudi", "Cihampelas", "Wastu Kencana", "Riau", "Purnawarman"]

3. Cicaheum-Ledeng

["Surapati", "Pahlawan", "Katamso", "Diponegoro", "Sulanjana", "Tamansari", "Siliwangi", "Cihampelas", "Cipaganti", "Setiabudi"]

["Setiabudi", "Cihampelas", "Siliwangi", "Sumur Bandung", "Tamansari", "Diponegoro", "Supratman", "Katamso", "Pahlawan", "Surapati"]

4. Cicaheum-Ciroyom

["Surapati", "Panatayuda", "Dipati Ukur", "Simpang Dago", "Sumur Bandung", "Tamansari", "Siliwangi", "Cihampelas", "Eyckman", "Pasir Kaliki", "Pajajaran"]

["Pajajaran", "Pasir Kaliki", "Eyckman", "Cipaganti", "Setiabudi", "Cihampelas", "Siliwangi", "Simpang Dago", "Dipati Ukur", "Panatayuda", "Surapati"]

5. Stasiun Hall-Dago

["Wastu Kencana", "Riau", "Dago"]

["Dago", "Riau", "Wastu Kencana"]

6. Stasiun Hall-Sadang Serang

["Supratman", "Katamso", "Pahlawan", "Cikutra Barat", "Sadang Serang"]

["Sadang Serang", "Cikutra Barat", "Pahlawan", "Katamso", "Supratman"]

7. Stasiun Hall - Ciumbuleuit

["Riau", "Merdeka", "Wastu Kencana", "Pajajaran", "Cihampelas", "Rivai", "Cipaganti", "Setiabudi", "Ciumbuleuit"]

["Ciumbuleuit", "Cihampelas", "Wastu Kencana", "Pajajaran", "Cicendo"]

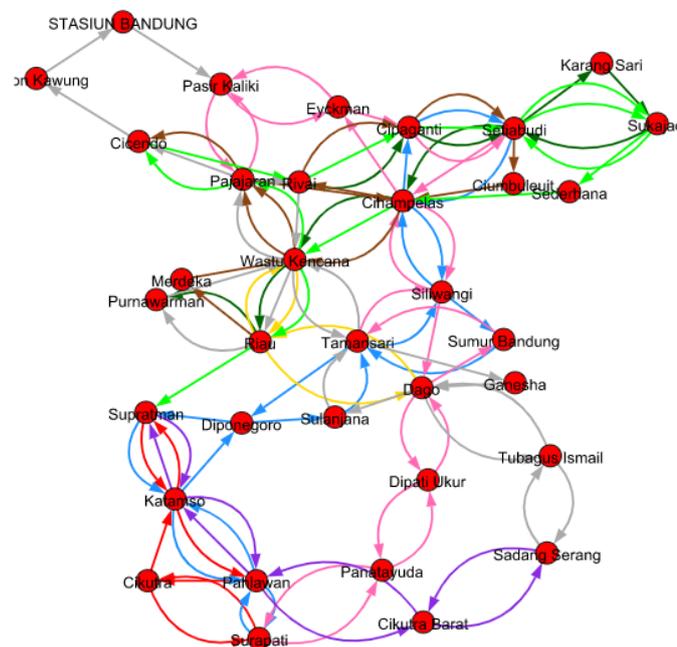
8. Margahayu Raya - Ledeng

["Wastu Kencana", "Pajajaran", "Cicendo", "Rivai", "Cipaganti", "Setiabudi", "Sukajadi", "Setiabudi"]
 ["Setiabudi", "Sukajadi", "Sederhana", "Cihampelas", "Wastu Kencana", "Riau", "Supratman"]

9. Sadang Serang - Caringin

["Pajajaran", "Rivai", "Wastu Kencana", "Tamansari", "Ganesha", "Dago", "Tubagus Ismail", "Sadang Serang"]
 ["Sadang Serang", "Tubagus Ismail", "Simpang Dago", "Tamansari", "Wastu Kencana", "Riau", "Purnawarman", "Wastu Kencana", "Pajajaran", "Cicendo", "Kebon Kawung", "STASIUN BANDUNG", "Pasir Kaliki"]

Berikutnya, akan dibangun graf menggunakan library igraph (graph visualizer) dan matplotlib (graph figure). Dasarnya adalah menjadikan setiap jalan sebagai vertice dan menjadikan hubungan jalan yang dilalui suatu jurusan angkot sebagai edge. Hasil visualisasi dengan konsep tersebut menjadi sebagai berikut:



Gambar 3.A.1. Graf gabungan rute angkot di Bandung Utara (pribadi)

Gambar dengan ukuran besar dapat dilihat pada tautan: <http://tiny.cc/grafangkotbdgutara>

Setiap satu jurusan diwakili oleh sebuah warna.

1. Abdul Muis-Cicaheum berwarna merah.
2. Abdul Muis-Ledeng berwarna hijau tua.
3. Cicaheum-Ledeng berwarna biru.

4. Cicaheum-Ciroyom berwarna pink.
5. Stasiun Hall-Dago berwarna emas kekuningan.
6. Stasiun Hall-Sadang Serang berwarna ungu.
7. Stasiun Hall-Ciumbuleuit berwarna coklat.
8. Margahayu Raya – Ledeng berwarna hijau muda.
9. Sadang Serang – Caringin berwarna abu-abu.

B. Pengujian

Akan dilakukan empat kali input sample dalam bentuk vertice asal dan vertice tujuan. Akan dilihat pula bagaimana perbedaan dan persamaan hasil pencarian yang terjadi.

1. V asal = "Pahlawan", V tujuan = "Dago"

```
Masukkan simpul asal : Pahlawan
Masukkan simpul tujuan : Dago
BFS : ['Pahlawan', 'Surapati', 'Panatayuda', 'Dipati Ukur', 'Dago']
DFS : ['Pahlawan', 'Cikutra', 'Surapati', 'Panatayuda', 'Dipati Ukur', 'Dago']
```

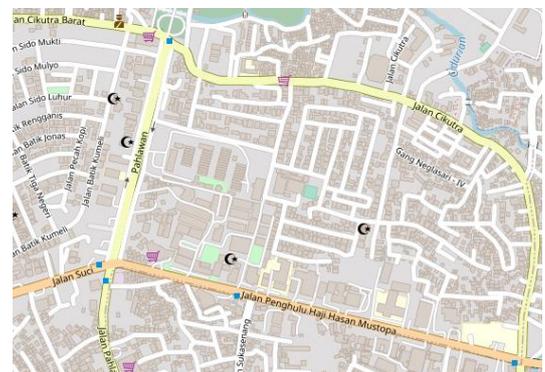
Rute ini merupakan rute bergerak dari arah utara timur ke utara barat, cukup dekat dan tidak memerlukan banyak pergantian angkot. Hasil pencarian jalur dari BFS dan DFS pun hampir serupa, hanya terdapat perbedaan satu jalan saja.

Untuk BFS, diperoleh jurusan angkot : Cicaheum-Ledeng dan Cicaheum-Ciroyom.

Untuk DFS, diperoleh jurusan angkot : AbdulMuis-Cicaheum dan Cicaheum-Ciroyom.

Jika ditinjau dari maps, jalan "Cikutra" sebelum "Surapati" hanyalah sebuah jalan memutar yang tidak efektif. Mengingat ini adalah rute dari timur ke barat, seharusnya dari "Pahlawan", kita bisa langsung mengambil "Surapati", daripada harus memutar lewat "Cikutra".

Keterangan : Jalan Suci = Jalan P.H.H.Mustopa = Surapati



Gambar 3.B.1. Peta asli Jalan Pahlawan, Bandung

2. V asal = "Wastu Kencana", V tujuan = "Setiabudi"

Rute ini adalah rute yang hanya melibatkan daerah utara barat dan merupakan rute dengan jarak menengah.

```

Masukkan simpul asal : Wastu Kencana
Masukkan simpul tujuan : Setiabudi
BFS : ['Wastu Kencana', 'Pajajaran', 'Cihampelas', 'Cipaganti', 'Setiabudi']
DFS : ['Wastu Kencana', 'Riau', 'Purnawarman', 'Dago', 'Sumur Bandung', 'Tamansari', 'Siliwangi', 'Cihampelas', 'Cipaganti', 'Setiabudi']

```

Untuk BFS, diperoleh jurusan angkot : StasiunHall-Ciumbuleuit dan Cicaheum-Ledeng.

Untuk DFS, diperoleh jurusan angkot : StasiunHall-Dago, Cicaheum-Ciroyom, Cicaheum-Ledeng.

Jika ditinjau dari maps, kedua rute yang dihasilkan ini merupakan pemilihan jalan yang sangat berbeda. Hasil pencarian BFS yang cenderung lebih dekat merupakan akses menuju "Setiabudi" via "Pajajaran", sedangkan hasil pencarian DFS yang terlihat lebih jauh merupakan akses menuju "Setiabudi" via "Siliwangi". Dengan demikian, jalur ini bisa saling menggantikan satu sama lain ketika terjadi hal-hal yang tidak terduga.

3. V asal = "Sadang Serang", V tujuan = "Siliwangi"

Rute ini merupakan rute yang cukup dekat dari utara tengah menuju utara barat.

```

Masukkan simpul asal : Sadang Serang
Masukkan simpul tujuan : Siliwangi
BFS : ['Sadang Serang', 'Tubagus Ismail', 'Dago', 'Sulanjana', 'Tamansari', 'Siliwangi']
DFS : ['Sadang Serang', 'Cikutra Barat', 'Pahlawan', 'Cikutra', 'Surapati', 'Panatayuda', 'Dipati Ukur', 'Dago', 'Sumur Bandung', 'Tamansari', 'Siliwangi']

```

Untuk BFS, diperoleh jurusan angkot : SadangSerang-Caringin dan Cicaheum-Ciroyom.

Untuk DFS, diperoleh jurusan angkot : StasiunHall-SadangSerang, AbdulMuis-Cicaheum dan Cicaheum-Ciroyom.

Jika ditinjau dari maps, kedua rute yang dihasilkan benar-benar berbeda.

Rute yang dihasilkan BFS merupakan rute terdekat yang langsung menuju tujuan. Akan tetapi, rute yang dihasilkan DFS merupakan rute memutar yang menyusuri daerah utara tengah, utara barat, kembali ke utara tengah, barulah menuju utara barat.

Pada gambar berikut, hasil pencarian BFS ditunjukkan dengan warna biru dan hasil pencarian DFS ditunjukkan dengan warna kuning.



Gambar 3.B.3. Perbandingan jalur hasil BFS dan DFS

Terdapat satu hal yang menarik dari sample ini. Setelah keluar dari "Tubagus Ismail" dan berada pada jalan "Dago", mengapa jalur biru BFS tidak

mengambil rute langsung belok saja seperti jalur kuning DFS?

Hal ini kembali lagi kepada prinsip awal BFS dan DFS. Kedua algoritma ini merupakan pencarian tanpa informasi. Pengambilan keputusan BFS dalam sample ini sama sekali tidak melihat "jarak jalan yang diambil".

Jumlah jalan yang sedikit belum tentu berjarak pendek dan jumlah jalan banyak belum tentu berjarak panjang. Akan tetapi, jumlah jalan sesedikit mungkin, dalam studi ini, akan mengarah pada jarak tempuh yang cenderung memendek, meskipun mungkin bukan yang terpendek.

4. V asal = "Surapati", V tujuan = "Setiabudi"

Rute ini merupakan salah satu rute perjalanan terpanjang, yakni dari utara timur ke utara barat.

```

Masukkan simpul asal : Surapati
Masukkan simpul tujuan : Setiabudi
BFS : ['Surapati', 'Pahlawan', 'Katamsa', 'Diponegoro', 'Sulanjana', 'Tamansari', 'Siliwangi', 'Cihampelas', 'Cipaganti', 'Setiabudi']
DFS : ['Surapati', 'Cikutra', 'Pahlawan', 'Katamsa', 'Supratman', 'Diponegoro', 'Sulanjana', 'Tamansari', 'Siliwangi', 'Cihampelas', 'Wastu Kencana', 'Riau', 'Purnawarman', 'Dago', 'Sumur Bandung', 'Dipati Ukur', 'Panatayuda', 'Tubagus Ismail', 'Sadang Serang', 'Cikutra Barat', 'Herdeka', 'Pajajaran', 'Pasir Kaliki', 'Eyckman', 'Cipaganti', 'Setiabudi']

```

Untuk BFS, diperoleh jurusan angkot : Cicaheum-Ledeng.

Untuk DFS, diperoleh jurusan angkot : AbdulMuis-Cicaheum, Cicaheum-Ledeng, AbdulMuis-Ledeng, Cicaheum-Ciroyom.

Untuk rute panjang ini, BFS berhasil menemukan solusi, meskipun kedalaman pencariannya cukup besar.

Di sample ini, mulai terlihat bahwa algoritma DFS tidak cocok untuk persoalan graf rute angkot ini. Timbul suatu lintasan sirkuler yang tidak efektif. "Cikutra" dan "Cikutra Barat" merupakan jalan yang saling bersebelahan. "Cikutra" yang muncul di awal pencarian diikuti dengan kemunculan "Cikutra Barat" di akhir pencarian.

IV. KESIMPULAN

Setelah dilakukan pengamatan terhadap beberapa sample uji, terlihat bahwa hasil yang diberikan oleh BFS dan DFS, keduanya bisa saling beririsan seperti pada sample pertama. Hasil lainnya yang dapat muncul yaitu saling melengkapi, seperti pada sample kedua. Dua buah hasil yang benar-benar berbeda dapat dipertimbangkan penggunaannya berdasarkan situasi dan kondisi yang terjadi di perkotaan sesungguhnya.

Kesimpulan pengamatan yang kedua adalah mengenai keefektifan pemanfaatan BFS dan DFS yang terjadi. Algoritma BFS menghasilkan rute yang selalu lebih pendek dibandingkan algoritma DFS pada semua sample. Hal ini sesuai dengan sifat BFS yang penelusurannya melebar sehingga menjamin ditemukannya rute terpendek dalam sebuah graf komplit. Sebaliknya, DFS tidak menunjukkan performa yang cemerlang untuk persoalan pencarian rute angkot ini. Hal ini disebabkan karena pembangkitan tetangga pada DFS baru akan terjadi ketika menemui jalan buntu, sedangkan pada graf rute angkot ini tidak ada jalan buntu sama sekali (graf sirkuler). Inilah juga

menyebabkan mengapa hasil algoritma DFS yang muncul cenderung banyak berputar-putar/tidak efektif.

Hal terakhir yaitu kinerja dari BFS dan DFS yang terjadi, pada akhirnya sesuai hipotesis. BFS yang menjamin rute terpendek dapat memberikan hasil-hasil yang cukup efektif, meskipun tidak yang paling efektif seperti pada sample 3. Dengan demikian, dapat disimpulkan bahwa algoritma BFS lebih baik dalam studi kasus pencarian rute angkot ini.

Akhir kata, semoga semua pengamatan yang telah dilakukan ini bisa bermanfaat bagi kemajuan dunia transportasi, khususnya transportasi berbasis rute seperti angkot dan bis, serta menjadi pedoman pemilihan rute bagi seluruh penduduk Indonesia pengguna transportasi berbasis rute.

VIDEO LINK AT YOUTUBE

<https://youtu.be/qhTIdQWyz84>

UCAPAN TERIMAKASIH

Penulis menghaturkan syukur sebesar-besarnya kepada Allah SWT. karena atas berbagai rahmat dan karuniaNya-lah penulis dapat menyelesaikan karya tulis ini. Penulis juga menyampaikan banyak terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc, Bapak Dr. Ir. Rinaldi Munir, Bapak Ir. Rila Mandala, M.Eng, Ph.D., dan Bapak Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D. sebagai pembimbing saya dalam mata kuliah IF2211 Strategi Algoritma. Tentunya saya juga mengucapkan terimakasih pada seluruh keluarga, teman dan rekan yang senantiasa mendukung saya dalam menyukseskan perkuliahan dan karya ilmiah ini.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag1.pdf>
- [2] University Of California San Diego, <https://cseweb.ucsd.edu/classes/sp15/cse100-ab/lectures/Lec12-DFSAndBFS.pdf>
- [3] Glorio Masala, Emil, dkk. "Perbandingan Algoritma Breadth First Search dan Depth First Search pada Game Mummy Maze Deluxe"

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2021



Almeiza Arvin Muzaki
13519066