

# Improved Breadth First search untuk optimasi pencarian jalur angkutan umum

Suprihatin

Sistem Informasi Universitas Ahmad Dahlan Yogyakarta Indonesia

e-mail: suprihatin@uad.ac.id

## Abstrak

Orang secara umum kesulitan untuk menentukan jalur angkutan, karena untuk menuju ke satu tujuan banyak alternatif jalur yang harus dilalui. Penelitian ini bertujuan untuk model pencarian rute jalur bus alternatif yang lebih cepat menghasilkan rute yang harus dilalui. Metode yang digunakan pada penelitian ini adalah Improved Breadth first search dengan cara memodifikasi BFS sehingga ditingkatkan unjuk kerjanya dalam menghasilkan penyelesaian pencarian rute. Metode improved BFS pada dasarnya sama seperti BFS melakukan pencarian level demi level berhenti jika ditemukan titik finish semu. Hasil penelitian menunjukkan bahwa improved BFS dapat meningkatkan unjuk kerja metode BFS. Penelitian dapat menjadi model untuk diaplikasikan pada kasus pencarian rute yang optimum lainnya.

**Kata Kunci:** BFS, pencarian rute, optimasi, improved BFS

## 1. Pendahuluan

Transportasi merupakan layanan jasa yang berguna untuk memindahkan atau membawa orang maupun barang dari satu tempat menuju ke tempat lain [1]. Transportasi adalah sarana fasilitas yang sering digunakan oleh masyarakat untuk menunjang segala aktivitasnya yang tidak dapat dipisahkan dari kehidupan sehari-harinya [2]. Transportasi umum merupakan kebutuhan bagi masyarakat kota. Transportasi membutuhkan informasi jalur kendaraan [3]. Jalur-jalur tersebut saling berpotongan di suatu halte atau terminal sehingga membentuk jaringan transportasi [4]. Seorang penumpang bepergian dari suatu tempat ke tempat lain dapat menaiki suatu jalur kendaraan dengan berpindah ke jalur kendaraan yang lain [5]. Bukan persoalan yang mudah bagi penumpang untuk dapat melakukan perpindahan jalur kendaraan sehingga akan sampai di tempat yang dituju [6]. Rute yang dibentuk oleh jalur kendaraan merupakan kumpulan halte atau terminal yang berurutan, dapat membentuk sirkuit euler graf yaitu dari satu halte/terminal akan kembali ke halte/terminal itu lagi [7].

Jaringan transportasi pada dasarnya adalah graff berarah dan berbobot dengan titik-titiknya adalah halte/terminal [8], garis adalah dua titik yang berurutan pada jalurnya, bobot

adalah jalurnya. Ada dua metode pencarian buta untuk mendapatkan penyelesaian pencarian jalur yaitu Depth First Search (DFS) [9] dan Breadth First Search (BFS) [10]. Proses DFS akan menelusuri node dari level terendah ke level selanjutnya untuk mendapatkan node yang dicari, jika tidak mendapatkan akan melakukan backtracking ke level sebelumnya [11]. Metode BFS pencarian dengan mengunjungi node-node pada level terendah sampai ditemukan node yang dicari jika tidak ditemukan akan dilanjutkan ke level selanjutnya [12]. DFS menghasilkan node-node yang lebih banyak daripada BFS, karena alasan inilah dipilih BFS untuk penelitian ini [13].

BFS membutuhkan waktu yang lama karena harus menyusuri level demi level. Permasalahan akan bertambah jika metode BFS diterapkan pada jalur transportasi yang terdiri dari banyak halte dan jalur [14]. Metode ini akan dimodifikasi dengan mengganti titik finish menjadi titik finish semu [15]. Titik finish semu adalah titik yang dapat mengantarkan ke titik finish dengan jalur tertentu [16].

## 2. Metode Penelitian

Graff adalah kumpulan titik-titik dan garis-garis yang menghubungkan [17]. Suatu path adalah kumpulan titik-titik yang berurutan tanpa titik yang sama [18]. Algoritma 1 berikut ini adalah

pseudocode BFS [19] untuk mencari path dari titik Start sampai titik Finish.

**Algoritma 1. Pseudocode BFS**

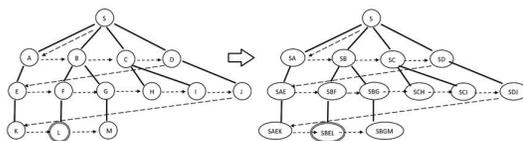
```

1  var
2  path : list of Node;
3  found : boolean;
4  Begin
5  queue ← null;
6  GetIntoQueue(Start);
7  queue position ← head of queue ;
8  found ← false;
9  while (not found) do
10  Begin
11  path ← element of queue position
12  found ← (last element of path = Finish);
13  if not found then
14  Begin
15  node ← the last node of the path
16  for all points: k
17  Begin
18  exist ← move(node,k);
19  if (exist and not in(k,path)) then
20  Begin
21  newpath ← path + k;
22  GetIntoQueue(newpath);
23  end;
24  end;
25  End;
26  queue position ← queue position next
27  end;
28  end;

```

Algoritma dimulai dengan antrian kosong pada baris 5, baris 6 memasukan titik start ke antrian. Posisi antrian pada awal antrian seperti ditunjukkan pada baris 7. Baris 8 inialisasi ketemu dengan harga false. Baris 9 perulangan yaitu selagi belum ketemu lakukan baris 10 sampai baris 27. Baris 11 menunjukan path yang dicari diisi dengan elemen posisi antrian. Jika jalur tersebut elemen terakhir adalah titik finish maka jalur yang dicari telah ketemu ditunjukkan pada baris 12. Jika tidak ketemu maka jalur diperdalam pencariannya, yang ditunjukkan pada baris 14 sampai dengan baris 25. Baris 15 adalah mengambil node dari elemen terakhir path. Baris 16 memberlakukan untuk semua titik-titik k yaitu jika ada penerus k dan k tidak berada dalam path maka baris 21 membuat path baru dengan menambahkan elemen k ke path tersebut dan baris 22 memasukan path baru ke antrian.

Gambar 1 dibawah ini contoh yang menunjukan proses pencarian metode BFS:



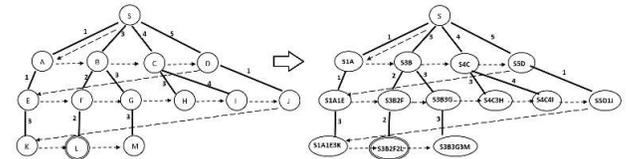
**Gambar 1. Pohon BFS dan Pohon Path**

Gambar 1 diatas dimulai dengan antrian kosong baris 5 pada pseudocode BFS, baris 6 antrian berisi string S, posisi antrian pada elemen S. Titik S bukan finish, titik A, B, C, dan D penerus titik S, maka String S dikembangkan menjadi SA, SB, SC, dan SD. Tambahkan hasil pengembangan ke antrian sehingga menjadi S, SA, SB, SC, SD. Untuk selanjutnya setiap selesai dikembangkan akan masuk ke antrian.

Posisi antrian maju ke elemen selanjutnya SA. Titik A pada SA bukan finish, titik E merupakan penerus titik A. maka elemen SA dikembangkan menjadi SAE dan masuk antrian. Posisi antrian maju ke elemen SB, Titik B bukan finish, titik F dan G penerus titik B. Elemen SB dikembangkan menjadi SBF dan SBG. Posisi antrian maju ke elemen SC. Titik C bukan finish, titik H dan I penerus titik C. Elemen SC dikembangkan menjadi SCH, SCI. Posisi antrian maju ke elemen SD, Titik D bukan finish, titik J penerus titik D. Elemen SD dikembangkan menjadi SDJ. Posisi antrian maju ke elemen SAEK, Titik K bukan finish, titik L tidak mempunyai penerus. Posisi antrian maju ke elemen SBEL, titik L finish proses selesai, sehingga hasilnya adalah SBEL.

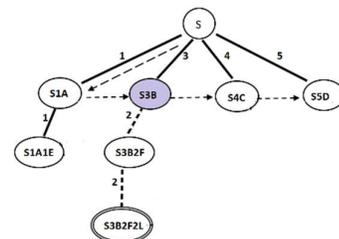
**3. Hasil dan Pembahasan**

Jalur angkutan merupakan graf berbobot, sebagai misal pohon pencarian seperti gambar 2 di bawah ini:



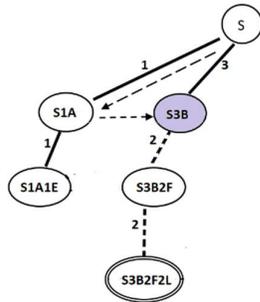
**Gambar 2. Pencarian BFS graff berbobot dan Pohon Path**

Proses pencarian BFS ditambahkan dengan bobotnya sehingga hasil pencariannya adalah S1A1E3K. Proses BFS membutuhkan waktu yang lama, metode ini akan diperbaiki dengan memotong node-node pada graff pencarian jika ditemukan node finish semu (yaitu node yang mengantarkan ke node finish dengan suatu jalur tertentu) [20]. Gambar di bawah menunjukan node B merupakan titik finish semu karena ada jalur 2 yang mengantarkan sampai titik finish (L). Gambar 3 berikut merupakan pohon pencarian BFS menambahkan titik finish semu B:



**Gambar 3. Pencarian BFS perbaruan jalur angkutan**

Proses pencarian akan diperbaiki lagi dengan menghentikan pengembangan path jika ditemukan titik finish semu, seperti pada gambar 4 berikut:



**Gambar 4.** Pencarian BFS perbaruan kedua jalur angkutan

Perbaikan terakhir pada gambar 4 diatas akan digunakan dalam menyelesaikan permasalahan pencarian jalur ini.

### Implementasi

Untuk mempermudah dalam implementasi akan dibentuk algoritma dan fungsi-fungsi yang menyertainya.

#### A. Algoritma

Berikut ini adalah pseudocode improve BFS sebagai implemtasi dari gambar 4 diatas bertujuan untuk mempercepat proses BFS:

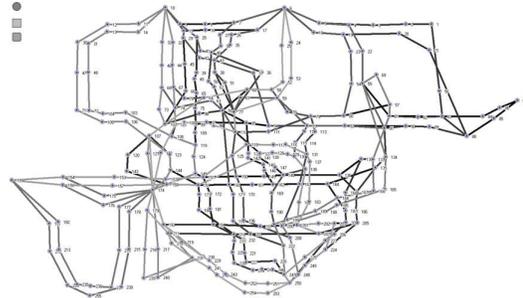
#### Algoritma 2. Pseudocode ImproveBFS

```

1  var
2  path : list of Node;
3  found : boolean;
4  stop : boolean;
5  Begin
6  queue ← null;
7  stop ← false;
8  GetIntoQueue(Start);
9  queue position ← head of queue ;
10 found ← false;
11 while (not found) and (not stop ) do
12   Begin
13   path ← element of queue position
14   found ← (speudofinish(last element of path));
15   if not found then
16     Begin
17     node ← the last node of the path
18     for all points: k and not stop
19       Begin
20         exist ← move(node,j,k);
21         if (exist and not in(k,path)) then
22           Begin
23             newpath ← path + j + k;
24             GetIntoQueue(newpath);
25             stop = speudofinish(k);
26           end;
27         end;
28   End;
29   If(stop) then
30     Begin
31       Jump to last queue position ;
32     found ← true;
33     end else
34       queue position ← queue position next
35   end;
36 end;
```

Algoritma 2 merupakan perbaruan BFS yaitu dengan menambahkan variabel stop untuk menghentikan proses pencarian. Variabel stop diinisialisasi dengan false pada baris 7. Perulangan while diperbarui dengan tidak ketemu dan tidak stop pada baris 11. Variabel ketemu diisi dengan titik finish semu pada baris 14. Baris 18 ada tambahan keadaan dan tidak stop. Baris 25 pemberian harga stop titik finish

semu. Baris 29-33 adalah jika ketemu titik finish semu kerjakan lompat ke akhir antrian dan telah ketemu jalurnya pada path terakhir antrian, jika tidak maka antrian maju ke antrian selanjutnya. Objek Penelitian akan dilakukan dengan jalur bus yang membutuhkan banyak node dan banyak jalur. Sebagai node akan dipilih tipe byte(char), dan jalur bertipe string untuk mempermudah implementasi dalam algoritma atau pemrogramnya. Akan diambil suatu jalur transportasi sebagai berikut:



**Gambar 5.** Graff Implentasi jalur angkutan

#### B. Fungsi Pindah

Jalur diwakili dengan string yaitu jika diketahui string s maka dapat dibentuk fungsi pindah(a,i,b), dimana i merupakan nomor jalurnya a dan b adalah karakter pada s yang berurutan. Elemen a lebih dahulu daripada elemen b. Jika digambarkan sebagai berikut:

Jalur i: -----ab-----

Jika diketahui titik a dan dicari titik mana saja sesudahnya maka tidak semua jalur mengandung fungsi pindah dari titik a ketitik lainnya hanya sebagian saja sebagai misal diketahui jalur-jalur sebagai berikut:

Jalur 1: -----ab-----

Jalur 2: -----ac-----

---

Jalur 3: -----b-----

Jalur 4: -----ax-----

Maka dapat diambil fungsi pindah sebagai berikut: pindah(a,1,b), pindah(a,2,c), pindah(a,4,x).

#### C. Fungsi Titik finish semu

Jika diketahui jalur-jalur yang membuat cycle sebagai berikut:

Jalur 1: abcdefghijka

Jalur 2: defgbacjklmnopd

Sebagai misal pencarian sampai titik c dan finishnya adalah k, maka c disebut titik finish semu karena dengan jalur 1 dari titik c bisa diantar ke titik k dengan titik-titik yang dilalui: cdefghijk, dengan jalur 2 titik-titiknya adalah cjk. Ada lebih dari satu maka akan diambil yang panjangnya minimal yaitu: cjk.

Sebagai misal pencarian sampai titik k dan finishnya adalah c maka untuk mempermudah dapat digambarkan jalurnya sebagai berikut:

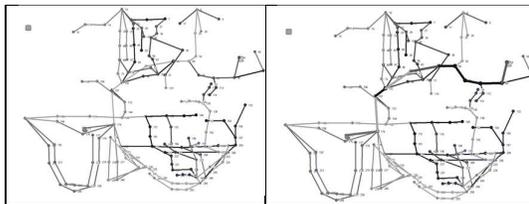
Jalur 1: abcdefghijkabcdefghijka

Jalur 2: defgbacjklmnopdefgbacjklmnopd

Sehingga titik-titik yang dilalui: dengan jalur 1: kabc, dengan jalur 2: klimnopdefgbac, ada lebih dari satu pilihan maka diambil yang minimal. Metode ini yang akan dipakai untuk mencari titik finish semu.

#### D. Percobaan Algoritma Improve BFS

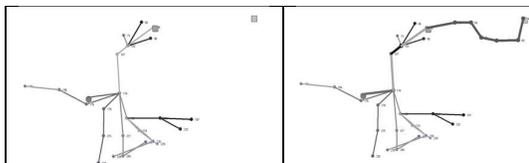
Sebagai misal start dan finish dipilih dan ditunjukkan dengan lingkaran sebagai start dan bujur sangkar sebagai titik finishnya. Pohon BFS dan hasil pencariannya adalah seperti gambar 6 sebagai berikut:



**Gambar 6.** Pohon BFS jalur angkutan dan hasil pencariannya

Gambar 6 adalah pembentukan pohon BFS dari graff gambar 5 dengan start 175 titik finish 54. Berdasarkan percobaan membutuhkan waktu: 27 detik 564 milidetik. Rutenya adalah 175 =11=> 174 =17=> 107 =2=> 102 =17=> 64 =9=> 60 =2=> 58 =2=> 78 =2=> 79 =2=> 80 =9=> 54 artinya dari titik 175 naik jalur 11 turun di 174 naik jalur 17 turun di 107 naik jalur 2 turun di 102 dan seterusnya berakhir di titik 54.

Mempergunakan Improve BFS dengan titik start yang sama 175 dan titik finish yang sama juga 54, maka pohon dan hasil pencariannya terbentuk seperti gambar 7 dibawah ini:



**Gambar 7.** Pohon Improve BFS jalur angkutan dan hasil pencariannya

Gambar 7 banyak titik-titik yang terhapus akan mempercepat proses pencarian, berdasarkan percobaan membutuhkan waktu: 0 detik 171 milidetik. Rutenya adalah 175 =11=> 174 =17=> 107 =2=> 102 =17=> 64. Titik 64 adalah titik finish semu yang digambarkan dengan titik persegi bulat. Setelah sampai ke titik finish semu ini dilanjutkan naik jalur 9 berhenti jika sudah sampai tujuan titik 54, dengan titik-titik yang

dilalui yaitu: 64=>60=>58=>78=>79=>80=>54

#### 4. Kesimpulan

Hasil perbandingan metode BFS dan Improve BFS menunjukkan metode Improve memberikan unjuk kerja lebih baik yaitu jauh lebih cepat menghasilkan hasil pencarian.

#### Saran

Variabel titik/node bertipe byte, maka terbatas hanya 256 titik saja, untuk selanjutnya dapat diterapkan untuk variabel yang lebih dari byte misalkan 2 byte atau yang lebih besar lagi.

#### Referensi

- [1] H. Y. Wono, M. Angela, and M. I. Reinald, "Pengaruh Kualitas Pelayanan dan Kepuasan Konsumen Terhadap Loyalitas Konsumen CV Saga Selaras Pratama," *CHANNEL J. Komun.*, vol. 8, no. 1, p. 51, 2020, doi: 10.12928/channel.v8i1.15119.
- [2] Nova, D. D. Rita, and N. Widiastuti, "Pembentukan Karakter Mandiri Anak Melalui Kegiatan Naik Transportasi Umum," *Comm-Edu (Community Educ. Journal)*, vol. 02, no. 02, pp. 113–118, 2019.
- [3] M. Agustien, E. Buchari, and dkk, "Sosialisasi Pelayanan Teman Bus Sebagai Upaya Meningkatkan Minat Masyarakat Meningkatkan Layanan Angkutan Umum Di Kota Palembang," *Community J. Pengabd.*, vol. 4, no. 1, pp. 29–38, 2022.
- [4] T. ANDRIANSYAH, "Evaluasi Konfigurasi Ruang Jalur Pejalan Kaki Di Lingkungan Universitas Brawijaya, Kota Malang," pp. 1–29, 2019.
- [5] S. M. Hafran, M. T. Syarkawi, I. Syafei, I. Munsyir, and S. Saleh, "Analisis Kinerja Angkutan Umum BMA (Studi Kasus Rute Pinrang – Makassar PP)," *PENA Tek. J. Ilm. Ilmu-Ilmu Tek.*, vol. 4, no. 2, p. 111, 2021, doi: 10.51557/pt\_jiit.v4i2.590.
- [6] H. Sukmarini and S. Bahri, "Kajian Penataan Jalur Pejalan Kaki Koridor Stasiun Tanah Abang Jakarta Pusat," *Plano Krisna*, vol. 17, no. 2, pp. 80–103, 2021.
- [7] Z. Buako, L. Yahya, and N. Achmad, "Aplikasi Algoritma Floyd-Warshall Dengan Pendekatan," vol. 9, no. 2, pp. 62–70, 2021.
- [8] M. A. Widiawaty, G. P. Pramulatsih, and V. Pebriani, "Jaringan Transportasi Dan Pengembangan Destinasi Pariwisata Di Kota Cirebon," *Travel Ind. student forum*, vol. 7, no. 1, pp. 127–148, 2019, [Online].

- Available:  
[https://www.researchgate.net/publication/269107473\\_What\\_is\\_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~reynal/Civilwars\\_12December2010.pdf%0Ahttps://think-asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625](https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~reynal/Civilwars_12December2010.pdf%0Ahttps://think-asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625).
- [9] O. Riansanti, M. Ihsan, and D. Suhaimi, "Connectivity algorithm with depth first search (DFS) on simple graphs," *J. Phys. Conf. Ser.*, vol. 948, no. 1, 2018, doi: 10.1088/1742-6596/948/1/012065.
- [10] A. Gaihre, Z. Wu, F. Yao, and H. Liu, "XBFS: EXploring runtime optimizations for Breadth-First Search on GPUs," *HPDC 2019- Proc. 28th Int. Symp. High-Performance Parallel Distrib. Comput.*, pp. 121–131, 2019, doi: 10.1145/3307681.3326606.
- [11] F. Pribadi and R. Munir, "Penerapan Pewarnaan Simpul Graf dengan Algoritma Welch Powell dan Algoritma Depth First Search pada Distribusi Giliran Main Angklung," *J. CoreIT J. Has. Penelit. Ilmu Komput. dan Teknol. Inf.*, vol. 7, no. 2, p. 58, 2021, doi: 10.24014/coreit.v7i2.15338.
- [12] N. Ratama, T. T, and Munawaroh, *Kecerdasan Buatan*, no. 1. 2021.
- [13] R. F. SYIHABUDDIN, "Implementasi Algoritma a-Star Dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang," vol. 4, no. 1, pp. 1–23, 2021.
- [14] F. Kurniawan, "Aplikasi Rekomendasi Jalur Bus Trans Jogja Menggunakan Google Maps Api Dan Google Places Api," *Tugas Akhir Jur. Tek. Inform. Fak. Teknol. Ind. Univ. Islam Indones.*, 2018.
- [15] R. F. Maulana, R. Anggoro, and S. Djanali, "Pengembangan Greedy Perimeter Stateless Routing (Gpsr) Dengan Konsep Overlay Network Pada Vanets," *JUTI J. Ilm. Teknol. Inf.*, vol. 18, no. 1, p. 29, 2020, doi: 10.12962/j24068535.v18i1.a944.
- [16] T. Rahmawati, Timothy John Pattiasina, and Michael Samuel Syaranamual, "Sistem Informasi Geografis Jalur Transportasi Angkutan Umum Berbasis Web: Studi Kasus Terminal Mardika Kota Ambon," *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 180–192, 2022, doi: 10.24002/konstelasi.v2i1.5629.
- [17] R. M. Barahama, C. E. J. C. Montolalu, and R. Tumilaar, "Eksentrisitas Digraf pada Graf Gir Menggunakan Algoritma Breadth First Search," *d'Cartesian J. Mat. dan Apl.*, vol. 10, no. 1, pp. 31–36, 2021, [Online]. Available: <https://ejournal.unsrat.ac.id/index.php/decartesian/article/view/32995>.
- [18] L. Safira, P. Harsadi, and S. Harjanto, "Penerapan Navmesh Dengan Algoritma A Star Pathfinding Pada Game Edukasi 3d Go Green," *J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, p. 17, 2021, doi: 10.30646/tikomsin.v9i1.540.
- [19] F. Lourence Tobing and F. Adline Twince Tobing, "Analisis Perbandingan Algoritma Dfs, Bfs Dan Dijkstra Untuk Menentukan Rute Terpendek Pada Peta Geografis," *J. Widya*, vol. 3, no. 1, pp. 59–67, 2022, [Online]. Available: <https://jurnal.amikwidyaloka.ac.id/index.php/awl>.
- [20] H. Sulaiman, Y. Yuliani, E. Fitri, N. Herlinawati, and S. Watmah, "Algoritma Dijkstra untuk Pendistribusian Carica Nida Food Wonosobo," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 2, p. 203, 2020,