

Laporan Pengembangan Sistem Pencarian Rute Pengiriman Barang Menggunakan Algoritma Uninformed Search



Disusun Oleh:

Muhammad Rafi Rajendra	2341720158
Kamila Habiba Putri Ananta	2341720175
Kibar Mustofa	2341720034

**Jurusan Teknologi Informasi
Politeknik Negeri Malang**

BFS

Breadth-First Search (BFS) adalah algoritma pencarian yang menjelajahi graf atau pohon level demi level, dimulai dari node akar (root). Algoritma ini menggunakan struktur data queue (antrian) untuk menyimpan node yang akan dieksplorasi, memastikan bahwa semua node pada level saat ini dikunjungi sebelum berpindah ke level berikutnya. Keunggulan utama BFS adalah kemampuannya menemukan jalur terpendek (optimal) dalam graf tidak berbobot karena sifatnya yang sistematis. Namun, BFS memerlukan memori yang besar karena harus menyimpan semua node yang belum dieksplorasi, terutama pada graf dengan kedalaman besar. Contoh aplikasi BFS adalah pencarian rute terpendek dalam peta atau algoritma crawling website.

DFS

Depth-First Search (DFS) adalah algoritma pencarian yang menjelajahi graf atau pohon dengan mendalami setiap cabang sejauh mungkin sebelum backtracking (mundur). Algoritma ini menggunakan struktur data stack (tumpukan) atau rekursi untuk mengeksplorasi node. DFS cenderung lebih efisien dalam penggunaan memori dibanding BFS karena hanya menyimpan satu jalur pada satu waktu. Namun, DFS tidak menjamin menemukan jalur terpendek dan bisa terjebak dalam loop jika graf memiliki siklus tanpa implementasi pengecekan node yang sudah dikunjungi. DFS sering digunakan untuk menyelesaikan masalah seperti puzzle, deteksi siklus, atau topological sorting.

UCS

Uniform Cost Search (UCS) adalah algoritma pencarian yang mengoptimalkan total biaya (cost) dari node awal ke node tujuan. UCS merupakan generalisasi dari BFS untuk graf berbobot dan menggunakan priority queue (heap) untuk mengeksplorasi node dengan biaya terkecil terlebih dahulu. Algoritma ini menjamin solusi optimal karena memprioritaskan node dengan biaya kumulatif terendah. Kelemahan UCS adalah kompleksitas memori yang tinggi, mirip dengan BFS, karena harus menyimpan semua node yang mungkin dieksplorasi. UCS sering dipakai dalam aplikasi seperti navigasi rute terpendek dengan bobot (misalnya, algoritma Dijkstra).

DLS

Depth-Limited Search (DLS) adalah varian dari DFS yang membatasi kedalaman pencarian hingga batas tertentu (max_depth). Algoritma ini dirancang untuk menghindari kelemahan DFS yang bisa terjebak dalam pencarian tak terbatas dengan mengabaikan node di luar kedalaman yang ditentukan. Jika solusi berada di dalam batas kedalaman, DLS akan menemukannya; jika tidak, pencarian gagal. DLS menjadi dasar untuk algoritma Iterative Deepening Depth-First Search (IDDFS) yang menggabungkan kelebihan BFS dan DFS. Contoh penggunaan DLS adalah dalam game AI atau pencarian di tree dengan kedalaman terbatas.

Jurnal:

PT Indah Logistik International Express merupakan perusahaan yang bergerak pada bidang jasa layanan pengiriman barang (yang mana dalam hal ini merupakan gudang pusat di Bekasi). Permasalahan yang ditemukan yakni pengiriman barang yang dilakukan tidak sesuai dengan kapasitas kendaraan yang digunakan. Masalah yang lain adalah rute yang telah dilaksanakan pengirimannya belum pernah dievaluasi terkait apakah rute tersebut adalah rute terbaik dalam hal kecepatan pengiriman, ketepatan waktu dan efisiensi biaya. Tujuan penelitian ini adalah untuk mendapatkan rute pengiriman barang yang optimal. Metode yang digunakan pada penelitian ini adalah metode Saving Matrix yang bertujuan agar pengiriman barang yang sesuai pesanan konsumen dapat dilakukan dengan cara yang efektif dan efisien. Hasil penelitian menunjukkan bahwa rute yang efektif berdasarkan perhitungan metode Saving Matrix adalah sebanyak 6 rute dengan jarak terpendek yaitu sebesar 212,4 km dan penghematan sebesar 65,5 km.

Keterangan:

A1 : Tujuan Pengiriman Jatiasih
A2 : Tujuan Pengiriman Pondok Gede
A3 : Tujuan Pengiriman Jati Sampurna
A4 : Tujuan Pengiriman Bantargebang
A5 : Tujuan Pengiriman Mustika Jaya
A6 : Tujuan Pengiriman Rawa Lumbu
A7 : Tujuan Pengiriman Bekasi Timur
A8 : Tujuan Pengiriman Bekasi Selatan
A9 : Tujuan Pengiriman Bekasi Barat
A10 : Tujuan Pengiriman Bekasi Utara
A11 : Tujuan Pengiriman Tambun Selatan
A12 : Tujuan Pengiriman Tambun Utara
A13 : Tujuan Pengiriman Cibitung

penghematan jarak:

Rangking	Gabungan rute	Saving (km)
1	A12-A13	24.5
2	A1-A3	23.8
3	A2-A3	21.6
4	A11-A13	20.8
5	A2-A9	20.7
6	A1-A2	17.3
7	A3-A4	16.9
8	A3-A9	16.4

9	A10-A12	16.3
10	A8-A9	16.3
11	A2-A13	16.1
12	A9-A13	16
13	A2-A8	14.7
14	A7-A12	14.1
15	A9-A10	13.7
16	A9-A12	13.5
17	A1-A4	12.5
18	A5-A13	12.5
19	A3-A8	12.4
20	A10-A13	12.4
21	A7-A13	12
22	A1-A9	11.5
23	A7-A9	11.4
24	A11-A12	11.4
25	A7-A10	10.1
26	A4-A13	9.8
27	A2-A10	9.4
28	A9-A11	9.3
29	A7-A11	9.3
30	A2-A4	9.2
31	A1-A8	9.1
32	A8-A10	8
33	A2-A12	7.6
34	A2-A7	6.6
35	A3-A12	6.6

36	A8-A12	6.6
37	A6-A9	6.5
38	A3-A5	6.4
39	A4-A9	6.4
40	A3-A6	6.3
41	A10-A11	6.3
42	A4-A10	6.1
43	A7-A8	5.9
44	A8-A13	5.9
45	A3-A10	5.4
46	A4-A7	5.1
47	A8-A11	4.9
48	A2-A6	4.8
49	A4-A5	4.7
50	A2-A11	4.4
51	A3-A11	4.3
52	A5-A11	4.3
53	A4-A8	4.1
54	A6-A8	3.7
55	A6-A12	3.4
56	A5-A9	3.3
57	A4-A11	3.2
58	A5-A12	3.2
59	A6-A13	3.2
60	A4-A6	3
61	A1-A6	2.9
62	A1-A2	2.8

63	A1-A7	2.8
64	A1-A12	2.7
65	A6-A7	2.6
66	A1-A10	2.5
67	A1-A5	2.3
68	A3-A7	2.3
69	A6-A10	2.2
70	A6-A11	2.1
71	A1-A11	1.6
72	A5-A7	1.4
73	A4-A12	1.4
74	A3-A13	1
75	A5-A6	0.5
76	A5-A8	0.4
77	A5-A10	0.3
78	A2-A5	0

link jurnal: <https://journal.unj.ac.id/unj/index.php/logistik/article/download/28949/13598>

Output Program

PENCARIAN JALUR OPTIMAL PENGIRIMAN
DENGAN VISUALISASI DAN OPTIMASI

Titik yang tersedia: ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'A11', 'A12', 'A13']

Masukkan titik awal (atau 'exit' untuk keluar): A2

Masukkan titik tujuan (pisahkan dengan koma, contoh: A2,A5,A8)

Tujuan: A1, A3

Gunakan batas waktu operasional? (y/n): n

Pilih metode pencarian:

1. Breadth-First Search (BFS)
2. Depth-First Search (DFS)
3. Uniform Cost Search (UCS)
4. Depth-Limited Search (DLS)
5. Bandingkan Semua Algoritma
6. Ganti titik awal/tujuan
7. Keluar

Masukkan pilihan (1-7):

--- Hasil BFS ---

HASIL PENCARIAN RUTE

Rute: A2 -> A1 -> A3

Total jarak: 16.20 km

Estimasi waktu perjalanan: 1 jam 32 menit

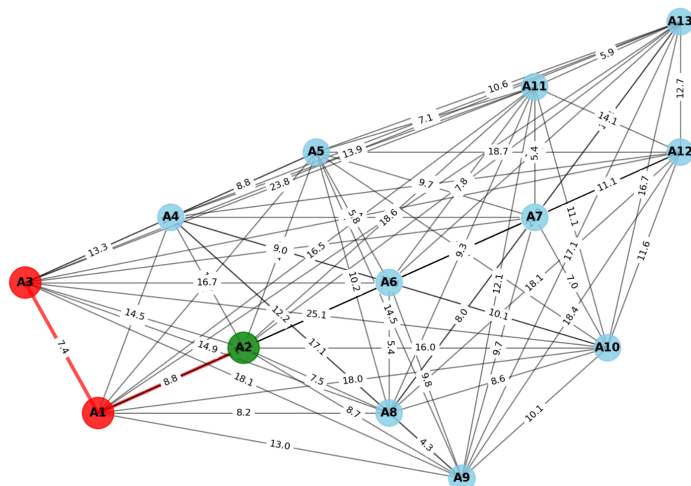
Jumlah node yang dikunjungi: 5

Waktu komputasi pencarian: 0.0008 detik

Apakah ingin menyimpan visualisasi? (y/n):

Figure 1

Visualisasi Graf Jaringan Pengiriman



--- Hasil DFS ---

=====

HASIL PENCARIAN RUTE

=====

Rute: A2 -> A12 -> A3 -> A13 -> A1 -> A13 -> A3

Total jarak: 189.70 km

Estimasi waktu perjalanan: 8 jam 19 menit

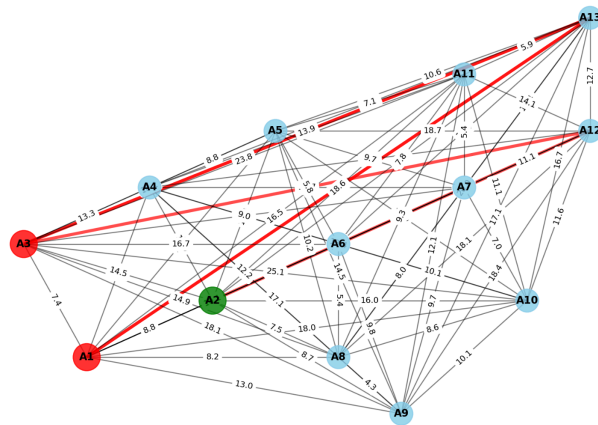
Jumlah node yang dikunjungi: 9

Waktu komputasi pencarian: 0.0000 detik

Apakah ingin menyimpan visualisasi? (y/n):

Figure 1

Visualisasi Graf Jaringan Pengiriman



--- Hasil UCS ---

=====

HASIL PENCARIAN RUTE

=====

Rute: A2 -> A1 -> A3

Total jarak: 16.20 km

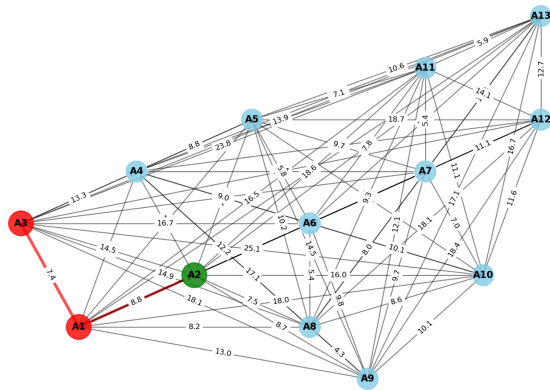
Estimasi waktu perjalanan: 1 jam 32 menit

Jumlah node yang dikunjungi: 6

Waktu komputasi pencarian: 0.0000 detik

Apakah ingin menyimpan visualisasi? (y/n):

Visualisasi Graf Jaringan Pengiriman



--- Hasil DLS (kedalaman 5) ---

=====

HASIL PENCARIAN RUTE

=====

Rute: A2 -> A12 -> A3 -> A13 -> A1 -> A13 -> A3

Total jarak: 189.70 km

Estimasi waktu perjalanan: 8 jam 19 menit

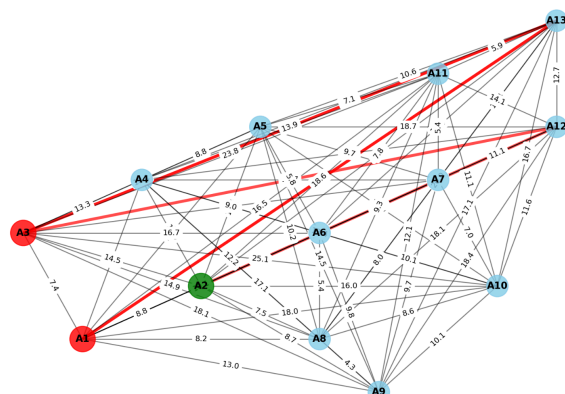
Jumlah node yang dikunjungi: 9

Waktu komputasi pencarian: 5.8023 detik

Apakah ingin menyimpan visualisasi? (y/n): n

Coba metode lain untuk rute yang sama? (y/n):

Visualisasi Graf Jaringan Pengiriman



Analisis Performa:

1. BFS:

- Waktu
Cenderung lebih lambat untuk graf besar karena ekspansi level demi level.
- Node Dikunjungi
Lebih banyak karena mengeksplorasi semua tetangga.
- Jalur
Optimal untuk graf tidak berbobot (jalur terpendek).

2. DFS:

- Waktu
Cepat jika solusi dekat root, tetapi bisa sangat lambat jika solusi di depth besar.
- Node Dikunjungi
Biasanya lebih sedikit jika solusi tidak dalam.
- Jalur
Tidak optimal (mungkin menemukan jalur panjang).

3. UCS:

- Waktu
Lebih lambat karena menggunakan priority queue.
- Node Dikunjungi
Bergantung pada struktur bobot.
- Jalur
Optimal untuk graf berbobot (jalur dengan cost terendah).

4. DLS:

- Waktu
Lebih cepat dari DFS jika depth limit sesuai.
- Node Dikunjungi
Terbatas oleh maksimal kedalaman.
- Jalur
Tidak optimal jika solusi di luar maksimal kedalaman.