

PENERAPAN ALGORITMA DIJKSTRA UNTUK MENCARI RUTE TERPENDEK KE KAMPUS

TUGAS KELOMPOK DAA:

NAYLA PUTRI CAHYA RAMADANI – 2024071020

DAVA PANGESTU PUTERA – 2023071029

1. Masalah yang Terjadi

Bayangkan mahasiswa UPJ Bintaro yang setiap hari pergi ke kampus.

- Kalau berangkat tanpa tahu rute mana yang paling dekat → boros waktu, boros bensin/ongkos, sering telat sampai kampus.
- Akibatnya, waktu belajar berkurang, biaya transportasi membengkak, dan sering kena macet di jalan yang sebenarnya lebih jauh.

Padahal sebenarnya ada beberapa pilihan jalan dari rumah ke kampus. Tapi masalahnya: sulit menentukan rute mana yang paling pendek, lewat jalan utama atau lewat jalan alternatif?

2. Solusi: Sistem Pencari Rute Terpendek dengan Algoritma Dijkstra

- Sistem ini bisa memberi tahu: "Kalau mau sampai kampus dengan jarak paling pendek, pakai rute ini!"
- Algoritma mengambil data dari peta jalan di sekitar Bintaro, mengukur jarak antar titik penting (perempatan, stasiun, mall, kampus).

3. Prinsip Divide and Conquer dalam Implementasi

Masalah besar (mencari rute terpendek di seluruh area Tangerang Selatan) dipecah jadi masalah kecil (rute terpendek per zona).

1. Divide (Bagi):

o Perjalanan dibagi menjadi beberapa zona (contoh: dari rumah di Tangerang Selatan ke Kampus UPJ):

- Zona 1: Rumah → Area Ciputat/Pamulang
- Zona 2: Area Ciputat → Area Bintaro (Stasiun/Plaza)
- Zona 3: Area Bintaro → Kampus UPJ

o Tiap zona punya beberapa titik penting:

- Zona 1: Rumah, Perempatan Ciputat, Stasiun Jurang Mangu
- Zona 2: Perempatan, Stasiun, Bintaro Plaza
- Zona 3: Bintaro Plaza, Kampus UPJ

2. Conquer (Taklukkan):

o Di tiap zona, algoritma Dijkstra menghitung jarak terpendek.

Contoh perhitungan Zona 1 (Rumah → Area Ciputat/Stasiun):

Titik yang bisa dilalui: Rumah (A), Ciputat (B), Stasiun Jurang Mangu (C)

Jarak antar titik:

- A → B: 3 km
- A → C: 4 km
- B → C: 2 km

Hasil Dijkstra Zona 1:

- Rumah → Ciputat: 3 km
- Rumah → Stasiun (lewat Ciputat): $3+2 = 5$ km
- Rumah → Stasiun (langsung): 4 km

Pilihan terpendek ke Stasiun: langsung = 4 km

o Proses yang sama dilakukan untuk Zona 2 dan Zona 3

3. Combine (Gabungkan):

o Semua hasil dari tiap zona digabung.

o Dari sini, sistem menghasilkan:

o Rute lengkap dengan jarak terpendek

o Total jarak keseluruhan

Contoh output gabungan:

ZONA	RUTE	JARAK
Zona 1	Rumah → Stasiun Jurang Mangu	4 km
Zona 2	Stasiun → Plaza Bintaro	3 km
Zona 3	Plaza Bintaro → Kampus UPJ	2 km
Zona 4	Rumah → Stasiun → Plaza → Kampus	9 km

4. Contoh Sederhana

Bayangkan kita mau ke warung makan dari kelas:

• Kalau jalan tanpa mikir → bisa jadi kita keliling sekolah dulu, padahal ada jalan pintas.

• Tapi kalau:

o Bagi perjalanan: dari kelas ke kantin, dari kantin ke gerbang, dari gerbang ke warung.

- o Tiap bagian diukur jaraknya (berapa meter tiap jalur).
- o Setelah itu, pilih yang total jaraknya paling pendek.
- Mirip dengan Divide and Conquer → lebih cepat tahu jalan mana yang paling dekat.

5. Algoritma yang Dipakai (dan disederhanakan)

- Membagi perjalanan jadi beberapa zona → seperti saat kita bagi tugas kelompok supaya lebih cepat selesai.
- Menjalankan Dijkstra per zona untuk cari jarak terpendek → mirip ketika kita pilih jalan mana yang paling dekat waktu pulang sekolah.
- Menggabungkan semua zona jadi satu rute lengkap → seperti ketua kelompok yang menyatukan hasil kerja semua anggota jadi satu laporan utuh.

6. Manfaat untuk Mahasiswa UPJ

- Hemat waktu dan biaya transportasi – tidak perlu coba-coba rute yang lebih jauh.
- Tidak telat ke kampus – sudah tahu rute terpendek sebelum berangkat.
- Bisa dikembangkan jadi aplikasi – mahasiswa tinggal input lokasi rumah, sistem langsung kasih tau rute terpendek ke kampus.
- Mengurangi kemacetan – kalau banyak orang pakai rute optimal, jalan tidak terlalu penuh di satu titik.

PENERAPAN ALGORITMA DIJKSTRA DI KEHIDUPAN NYATA

Algoritma Dijkstra sudah banyak dipakai di berbagai sistem:

1. Aplikasi Peta Digital (Google Maps, Waze)
 - o Masalah: mencari rute terpendek dari rumah ke tujuan.
 - o Divide: area dibagi jadi beberapa zona jalan.
 - o Conquer: tiap zona dihitung jaraknya.
 - o Combine: sistem gabungkan semua zona dan kasih tau rute terpendek.
2. Aplikasi Ojek Online (Gojek, Grab)
 - o Masalah: mencari driver terdekat dan rute tercepat.
 - o Divide: kota dibagi jadi grid area kecil.
 - o Conquer: cari driver terdekat di area itu dan hitung rute dengan Dijkstra.
 - o Combine: sistem kasih tau driver mana yang paling dekat dengan rute optimal.
3. Sistem Navigasi GPS
 - o Masalah: mengarahkan kendaraan ke tujuan dengan jarak minimal.
 - o Divide: peta dibagi per segmen jalan.
 - o Conquer: hitung jarak per segmen.
 - o Combine: tampilkan rute lengkap di layar GPS.

KESIMPULAN

Algoritma Dijkstra dengan pendekatan Divide and Conquer adalah solusi yang tepat untuk mencari rute terpendek karena:

- Efisien – tidak perlu hitung semua kemungkinan rute sekaligus, cukup per zona
- Akurat – selalu menemukan rute dengan jarak paling pendek
- Praktis – mudah diimplementasikan dalam aplikasi mobile atau web
- Terbukti – sudah digunakan di Google Maps, Gojek, GPS, dan berbagai aplikasi navigasi lain
- Bermanfaat – membantu mahasiswa hemat waktu dan biaya transportasi ke kampus