

### Teori Graf

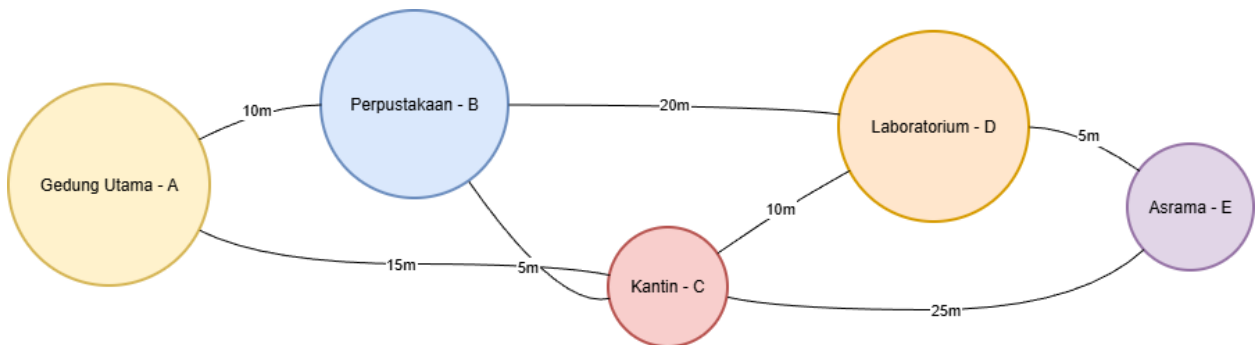
<b>Nama</b>	<b>:Muhammad Subhan Fajriansyah</b>
<b>NIM</b>	<b>:3312511004</b>
<b>Kelas</b>	<b>: Malam A</b>

#### Tugas Studi Kasus

- I. Sebuah kampus vokasi memiliki sejumlah gedung yang saling terhubung melalui jalan setapak, koridor, dan jalur kendaraan. Pihak kampus berencana membangun system navigasi digital yang mampu:
  - Menampilkan jalur antar Gedung
  - Menentukan rute terpendek antar lokasi,
  - Mengevaluasi rute patroli keamanan kampus
  - Menentukan rute kunjungan ke seluruh gedung tanpa pengulangan.
- a. Jelaskan bagaimana studi kasus kampus di atas dapat dimodelkan dalam bentuk graf.  
= Studi kasus ini dimodelkan sebagai graf  $G = (V, E)$ , di mana kampus direpresentasikan sebagai sekumpulan entitas diskrit (gedung) yang saling terhubung.
- b. Tentukan apa yang menjadi simpul (vertex) dan sisi (edge) dalam graf tersebut.
  - Simpul (Vertex): Merepresentasikan gedung atau lokasi spesifik (contoh: Gedung Utama, Perpustakaan, kantin, laboratorium dan asrama).
  - Sisi (Edge): Merepresentasikan jalur penghubung fisik seperti jalan setapak, koridor, atau jalur kendaraan.
- c. Jelaskan apakah graf yang terbentuk termasuk graf berarah atau tidak berarah, serta alasannya.  
= Model ini dapat menggunakan Graf Berarah jika terdapat jalur kendaraan satu arah atau koridor yang membatasi arah jalan. Namun, untuk jalur pejalan kaki dua arah, dapat dimodelkan sebagai graf tak berarah. Dalam sistem navigasi modern, graf berarah lebih tepat karena mampu mengakomodasi batasan arah akses.
- d. Jelaskan jenis graf yang paling sesuai untuk memodelkan sistem navigasi kampus tersebut.  
= Jenis graf yang paling sesuai adalah Graf Berbobot (Weighted Graph) yang Berarah. Bobot pada sisi merepresentasikan jarak fisik (meter) atau waktu tempuh antar gedung, yang merupakan parameter krusial bagi Algoritma Dijkstra dalam menentukan lintasan terpendek.

- e. Sebutkan minimal dua terminologi dasar graf yang relevan dalam kasus ini dan jelaskan maknanya dalam konteks kampus.
1. Lintasan Terpendek (Shortest Path): Barisan sisi dengan total bobot minimum dari simpul asal ke simpul tujuan, digunakan untuk navigasi antar lokasi.
  2. Adjacent (Bertetangga): Dua gedung dikatakan bertetangga jika terhubung langsung oleh satu jalur tanpa melewati gedung lain.
  3. Path (Lintasan): Urutan jalur yang dilalui dari gedung asal ke gedung tujuan.
  4. Degree (Derajat): Jumlah jalur yang terhubung ke sebuah gedung; berguna untuk menganalisis titik pertemuan tersibuk di kampus.
  5. Sirkuit Hamilton: Lintasan tertutup yang melalui tiap simpul (gedung) tepat satu kali, relevan untuk optimasi rute kunjungan seluruh gedung tanpa pengulangan.
  6. Sirkuit Euler: Lintasan yang melalui setiap sisi (jalur) tepat satu kali, relevan untuk mengevaluasi efisiensi rute patroli keamanan agar seluruh jalan terpantau.

II. Buatlah graf dengan memisalkan jumlah gedung dan jalur penghubung tiap Gedung beserta jaraknya.



- $V$  (Gedung): {A: Gedung Utama, B: Perpustakaan, C: Kantin, D: Laboratorium, E: Asrama}
- $E$  (Jalur & Jarak):
  - $(A - B) = 10 \text{ m}$
  - $(A - C) = 15 \text{ m}$
  - $(B - C) = 5 \text{ m}$
  - $(B - D) = 20 \text{ m}$
  - $(C - D) = 10 \text{ m}$
  - $(D - E) = 5 \text{ m}$
  - $(C - E) = 25 \text{ m}$

- a. Tentukan apakah graf memiliki lintasan Euler atau sirkuit Euler. Kemudian jelaskan implikasi hasil tersebut terhadap rute patroli keamanan kampus.

- Analisis:
    - Gedung A: derajat 2
    - Gedung B: derajat 3 (Ganjil)
    - Gedung C: derajat 4
    - Gedung D: derajat 3 (Ganjil)
    - Gedung E: derajat 2
  - Jawaban: Karena terdapat tepat dua simpul berderajat ganjil (B dan D), maka graf ini memiliki Lintasan Euler, tetapi tidak memiliki Sirkuit Euler.
  - Implikasi: Petugas keamanan bisa melakukan patroli melewati setiap jalan tepat satu kali, namun mereka tidak bisa mengakhiri patroli di gedung tempat mereka memulai (titik awal  $\neq$  titik akhir).
- b. Gunakan algoritma Dijkstra dan tampilkan Langkah perhitungannya dalam menentukan lintasan terpendek beserta total jaraknya.
- =
- Algoritma Dijkstra (Rute Terpendek A ke D):
- Langkah 1: Inisialisasi jarak  $A=0$ , lainnya  $\infty$ .
- Langkah 2: Dari A, tetangga adalah B(10) dan C(15). Pilih B karena terkecil.
- Langkah 3: Dari B, cek tetangga: C ( $10+5=15$ , sama) dan D ( $10+20=30$ ).
- Langkah 4: Cek jalur lewat C:  $A \rightarrow C \rightarrow D$  memiliki total jarak  $15 + 10 = 25$ .
- Hasil: Lintasan terpendek dari A ke D adalah  $A \rightarrow C \rightarrow D$  dengan total jarak 25 meter. (Lebih pendek daripada lewat B yang jaraknya 30m).

### Tugas Baca

- Terminologi graf (terhubung, Upagraf (Subgraf), Komplemen upagraf, Upagraf merentang, Cut Set, Graf Berbobot).
- Beberapa Graf Sederhana Khusus.
- Representasi Graf.

### Rangkuman Tugas baca :

- Terminologi Graf Lanjutan

Berdasarkan pemahaman saya, terdapat beberapa istilah teknis yang krusial dalam struktur graf:

- Graf Terhubung (*Connected Graph*): Kondisi di mana setiap simpul dalam graf memiliki jalur untuk mencapai simpul lainnya. Tidak ada bagian yang terisolasi.
- Upagraf (*Subgraf*): Bagian atau pecahan dari sebuah graf utama. Jika kita mengambil beberapa titik dan garis dari graf asli, itu disebut subgraf.
- Komplemen Upagraf: Kumpulan sisi (jalur) yang ada di graf utama tetapi tidak disertakan dalam upagraf yang kita pilih tadi.
- Upagraf Merentang (*Spanning Subgraph*): Subgraf yang menyertakan seluruh titik (simpul) dari graf asli, namun tidak harus menyertakan semua jalurnya.
- Cut Set: Himpunan sisi yang sangat krusial; jika sisi-sisi ini dihapus, graf yang tadinya terhubung akan langsung terputus menjadi dua bagian terpisah.
- Graf Berbobot (*Weighted Graph*): Graf yang setiap sisinya memiliki nilai numerik. Dalam konteks navigasi kampus (seperti studi kasus hal. 22), bobot ini bisa merepresentasikan jarak meter atau waktu tempuh antar gedung.

- Beberapa Graf Sederhana Khusus

- Graf Lengkap ( $K_n$ ): Graf yang sangat padat karena setiap simpul terhubung langsung ke semua simpul lainnya.
- Graf Lingkaran ( $C_n$ ): Graf yang bentuknya menyerupai cincin atau sirkuit tertutup sederhana.
- Graf Teratur (*Regular Graph*): Graf yang "adil", di mana setiap titiknya memiliki jumlah derajat (koneksi) yang sama persis.

- Graf Bipartit: Graf unik di mana simpul-simpulnya dibagi menjadi dua kelompok. Aturannya: sisi hanya boleh menghubungkan simpul dari kelompok A ke kelompok B, tidak boleh ada koneksi sesama anggota kelompok.

c. Representasi Graf dalam Pemrograman

Untuk mengimplementasikan graf ke dalam kode program (misalnya menggunakan Python atau C++), ada tiga cara utama:

1. Matriks Ketetanggaan (*Adjacency Matrix*): Menggunakan tabel dua dimensi. Sangat cepat untuk mengecek apakah dua simpul terhubung, tapi memakan banyak memori jika grafnya besar namun jalurnya sedikit.
2. Matriks Bersisian (*Incidence Matrix*): Matriks yang menunjukkan hubungan antara titik simpul dengan garis sisinya.
3. Senarai Ketetanggaan (*Adjacency List*): Cara yang paling efisien memori. Setiap simpul memiliki daftar (list) siapa saja tetangga langsungnya. Ini biasanya yang paling sering dipakai dalam algoritma rute terpendek.