

Praktik Kecerdasan Buatan

Algoritma Searching

Searching Algorithm

-

-

-

DFS

BFS

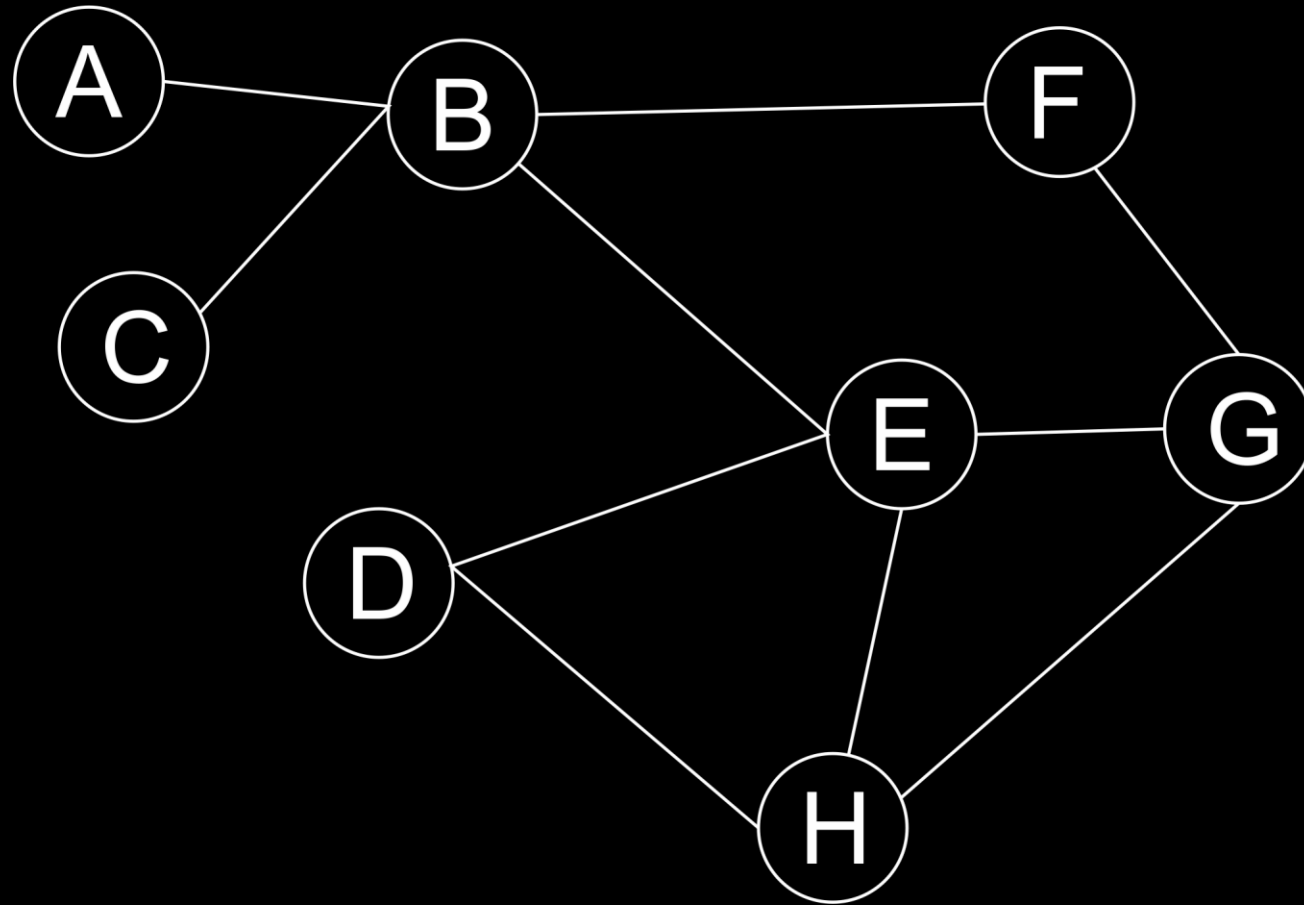
DJIKSTRA

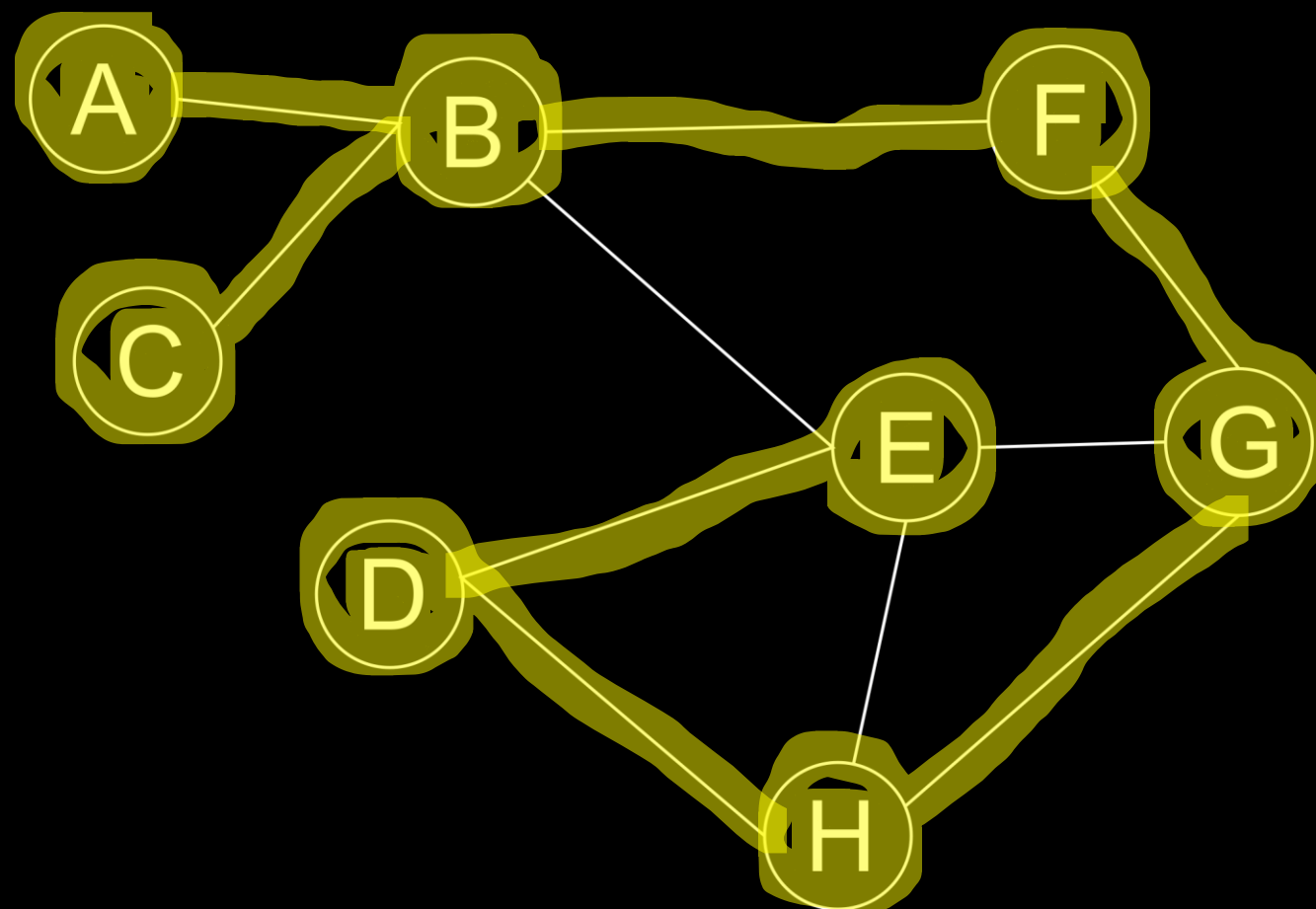
-

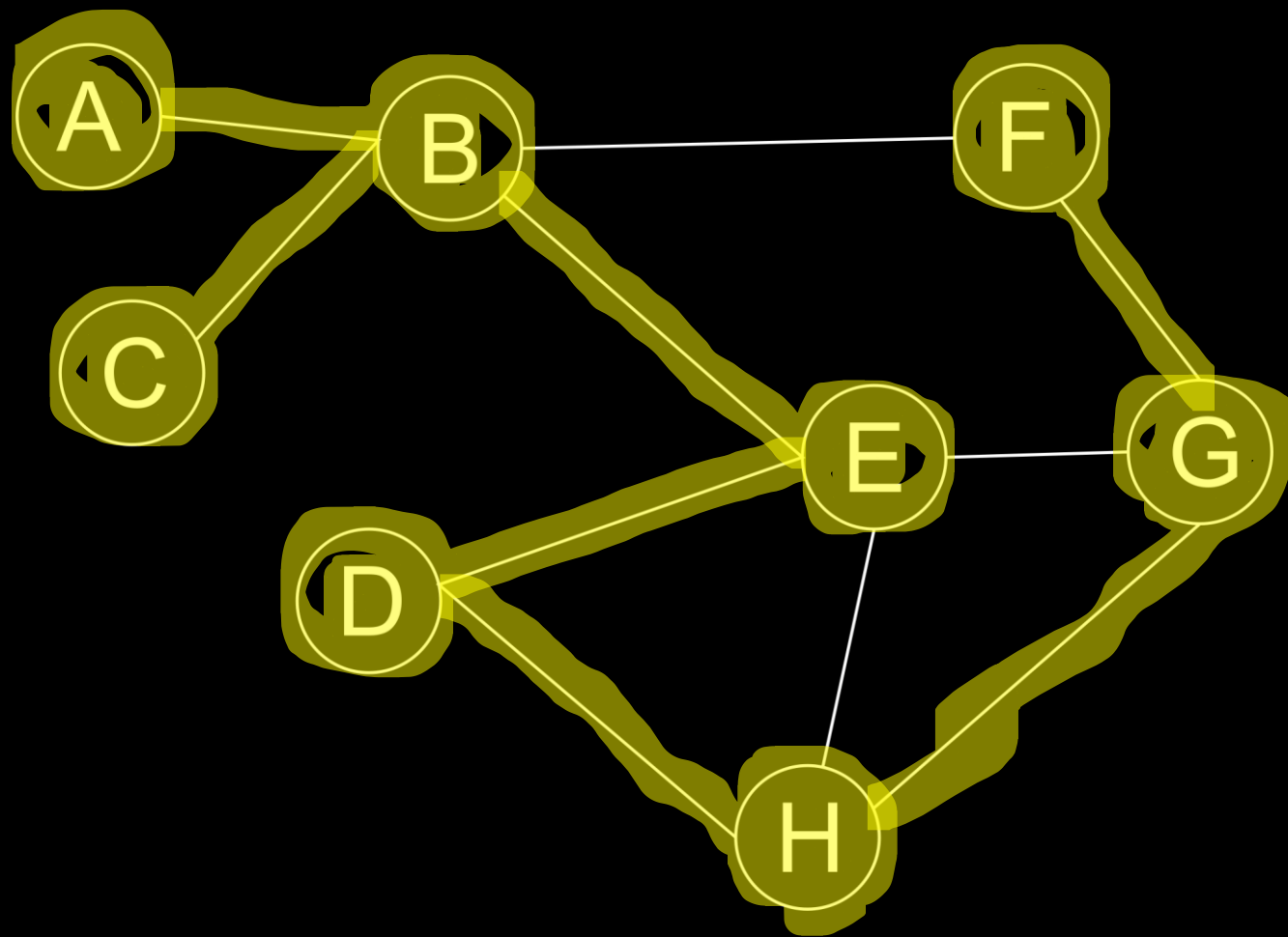
-

-

Minggu Lalu... (DFS)







stack/tumpukan



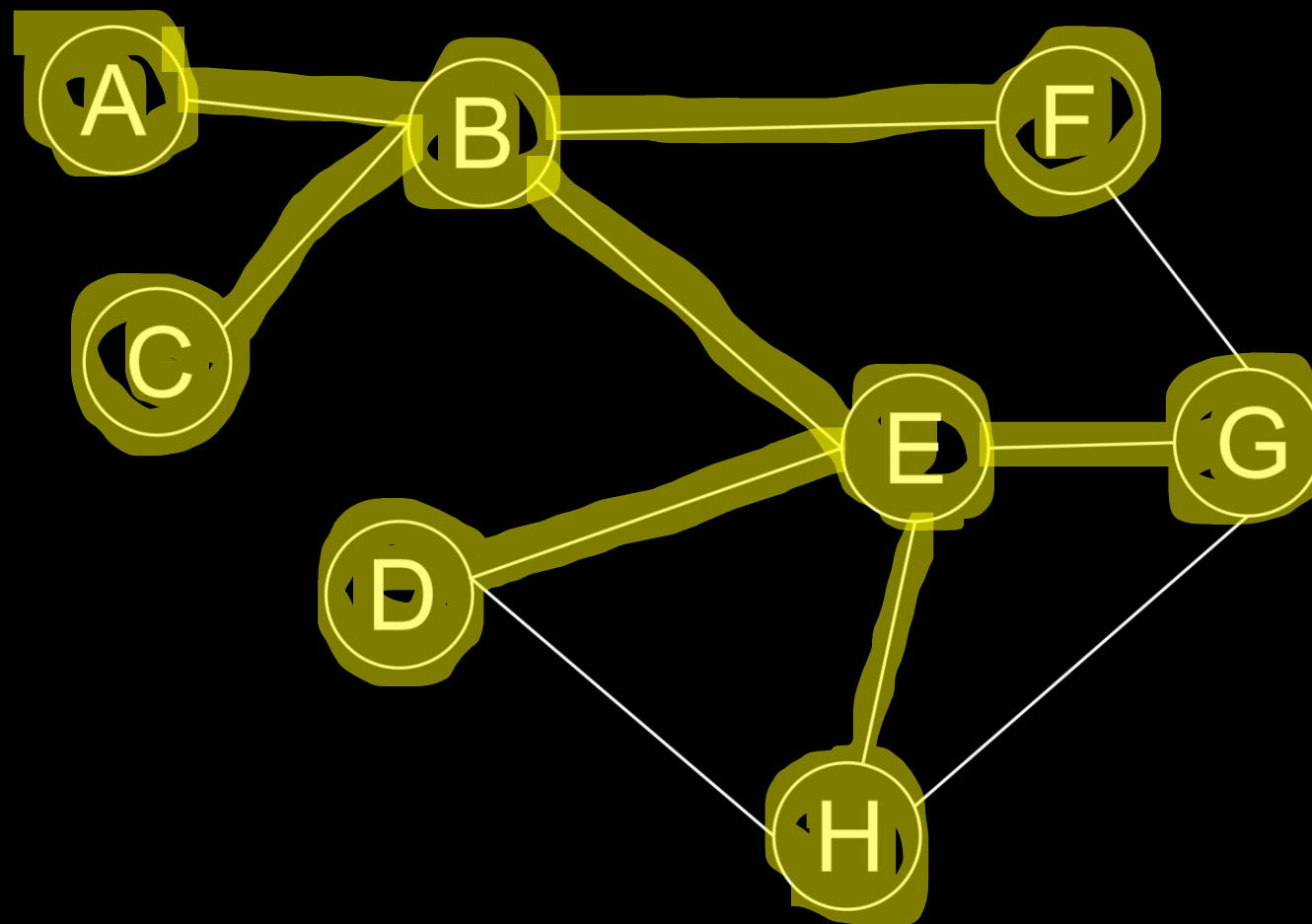
ambil yang
terakhir untuk
dikunjungi

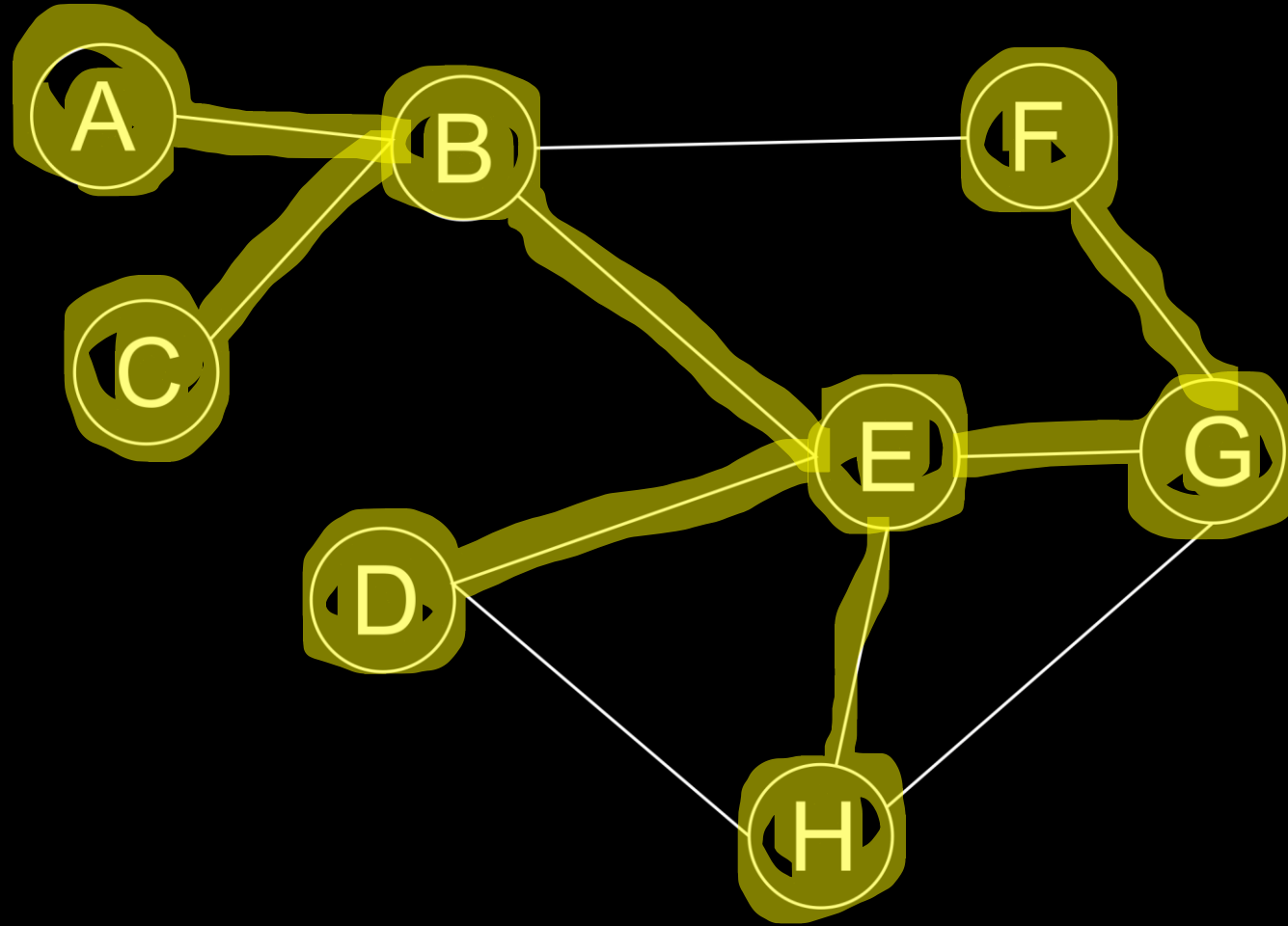
Pada DFS, kita dapat menggunakan konsep rekursi atau pun stack/tumpukan.

Dan perlu diingat bahwa **stack/tumpukan** mengambil data dari yang paling terakhir!

BFS (Breadth First Search)

- Breadth-First Search (BFS) adalah **algoritma pencarian** yang digunakan untuk **mengeksplorasi atau mencari informasi dalam struktur data** seperti grafik atau pohon secara level demi level, mulai dari simpul awal (atau akar) dan melanjutkan dengan mengeksplorasi semua simpul yang bertetangga pada tingkat yang sama sebelum melanjutkan ke tingkat berikutnya. Dengan kata lain, **BFS bekerja dengan cara "menyebar" terlebih dahulu sebelum "mendalam."**
- Contoh sederhana: Bayangkan Anda mencoba menemukan jalan terpendek dari titik A ke titik B di peta jalan kota. Anda akan mulai dengan mencari di sekitar lokasi titik A, kemudian memeriksa area di sekitar itu, dan seterusnya, hingga Anda menemukan titik B. Ini adalah cara BFS berfungsi, dengan fokus pada penjelajahan secara "lebar" terlebih dahulu sebelum memeriksa tingkat lebih dalam dari struktur data.





queue/antrian



ambil yang
paling awal untuk
dikunjungi

Pada BFS, kita dapat menggunakan konsep queue/antrian

Dan perlu diingat bahwa queue/antrian
mengambil data dari yang paling awal!

Perbandingan DFS dan BFS

Perbedaan utama antara Breadth-First Search (BFS) dan Depth-First Search (DFS) terletak pada cara mereka menjelajahi atau mencari dalam struktur data seperti grafik atau pohon.

Perbandingan DFS dan BFS (lanjutan)

Cara Penjelajahan:

- **BFS:** BFS menjelajahi atau mencari **secara level demi level**, dimulai dari simpul awal dan mengeksplorasi semua simpul yang bertetangga **pada tingkat yang sama sebelum bergerak ke tingkat berikutnya.**
- **DFS:** DFS menjelajahi atau mencari dengan cara **mengeksplorasi sejauh mungkin di sepanjang satu cabang (branch) sebelum kembali dan menjelajahi cabang-cabang lain.** Ini berarti DFS akan "mendalam" ke dalam sebuah cabang sebelum menjelajahi cabang lainnya.

Perbandingan DFS dan BFS (lanjutan)

Urutan Penemuan/Searching:

- **BFS:** BFS umumnya menemukan solusi atau simpul tujuan terdekat terlebih dahulu. Dalam konteks mencari jalur terpendek, BFS menjamin bahwa solusi yang pertama kali ditemukan adalah jalur terpendek.
- **DFS:** DFS tidak menjamin bahwa solusi yang pertama kali ditemukan adalah jalur terpendek. Solusi yang ditemukan pertama kali mungkin bukan yang terpendek, tergantung pada struktur grafik dan urutan pencarian.

Perbandingan DFS dan BFS (lanjutan)

Struktur Data:

- **BFS:** BFS biasanya diimplementasikan dengan menggunakan **antrian (queue)** sebagai struktur data, sehingga simpul yang pertama kali dimasukkan ke dalam antrian adalah yang pertama kali dieksplorasi.
- **DFS:** DFS biasanya diimplementasikan dengan menggunakan **tumpukan (stack) atau rekursi** sebagai struktur data, sehingga simpul yang terakhir dimasukkan adalah yang pertama kali dieksplorasi.

Perbandingan DFS dan BFS (lanjutan)

Pemakaian Memori:

- **BFS:** Biasanya menggunakan lebih banyak memori dibandingkan dengan DFS karena harus menyimpan semua simpul pada tingkat yang sama dalam antrian.
- **DFS:** Biasanya lebih hemat memori dibandingkan dengan BFS karena hanya perlu menyimpan simpul pada satu cabang sekaligus.

Perbandingan DFS dan BFS (lanjutan)

- **Keuntungan Aplikasi:**
- **BFS:** Digunakan ketika Anda ingin menemukan jalur terpendek atau mencari solusi yang terdekat, seperti dalam masalah pencarian jarak terpendek.
- **DFS:** Digunakan dalam situasi di mana Anda ingin menjelajahi seluruh cabang dalam sebuah struktur, seperti dalam pencarian jalur, penelusuran grafik, atau topological sorting.

Pilihan antara BFS dan DFS tergantung pada jenis masalah yang akan kita selesaikan dan persyaratan spesifik yang Anda miliki, termasuk apakah Anda ingin menemukan jalur terpendek atau hanya menjelajahi seluruh struktur.

Praktik DFS dan BFS

Tentukan jalur terpendek dari A ke G menggunakan DFS dan BFS.

Program bisa dilihat di:

<https://github.com/taufiqmus/Praktik-Kecerdasan-Buatan/tree/main/Searching>

Bandingkan jika dilakukan prosedur manual (tanpa program), apakah sama dengan hasil dari program? Buatlah analisisnya!

