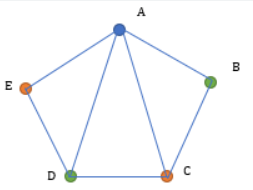
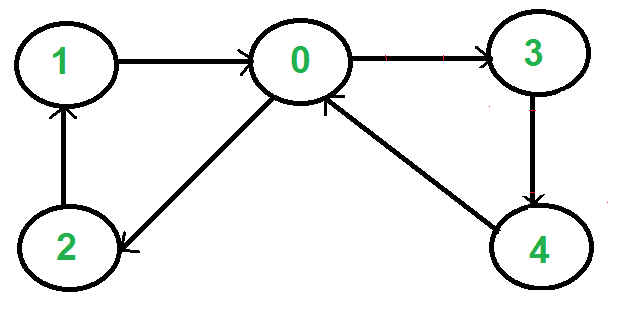
**BAB 8.GRAF**

**8.1 Definisi Graf**   
 Graf adalah himpunan benda-benda yang disebut "simpul" (vertex atau node) yang terhubung oleh "sisi" (edge) atau "busur" (arc). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan "vertex” / “simpul") yang dihubungkan oleh garis-garis (melambangkan "edge" / "Sisi”) atau garis berpanah (melambangkan "busur"). Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama. Sisi yang demikian dinamakan "gelang" (loop)   
 Graf dibagi menjadi berapa macam yaitu :   
 **1.Undirected Graf**



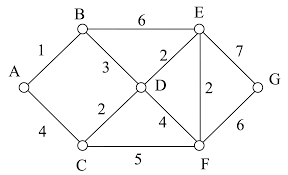
Sumber : https://www.foldertips.com/tik/graph-pohon/

**2.Directed Graf**



Sumber : https://www.geeksforgeeks.org/what-is-directed-graph-directed-graph-meaning/

**3.Weighted Graf**



Sumber : https://webwork.moravian.edu/100.2/sec\_shortPaths.html

**8.2 Operasi pada Graf**

|  |  |
| --- | --- |
| Add Vertex | Menambah Total Simpul |
| Delete Vertex | Menghapus Simpul |
| Add Edge | Menambah Masing masing garis Antara simpul |
| Delete Edge | Menghapus salah satu garis antara simpul |
| SearchPath | Mencari Jalur antara simpul satu dengan yang lain |

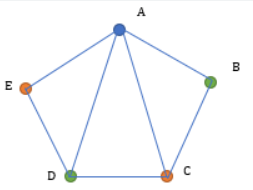
**8.3 Konsep Implementasi Graf pada Adjency Matriks**

Adjacency matrix atau matriks ketetanggaan adalah salah satu representasi graf yang paling umum. Representasi ini menggunakan matriks untuk merepresentasikan hubungan antar simpul pada sebuah graf.

Dalam adjacency matrix, setiap baris dan kolom merepresentasikan sebuah simpul pada graf, dan nilai pada matriks menunjukkan apakah ada sisi yang menghubungkan dua simpul tersebut atau tidak. Untuk undirected dan directed graf Jika ada sisi yang menghubungkan dua simpul, maka nilai pada matriks adalah 1, dan jika tidak, maka nilai pada matriks adalah 0.namun untuk weighted graph nilai pada matriks untuk kedua simpul diisi dengan nilai beban dari edge atau garis

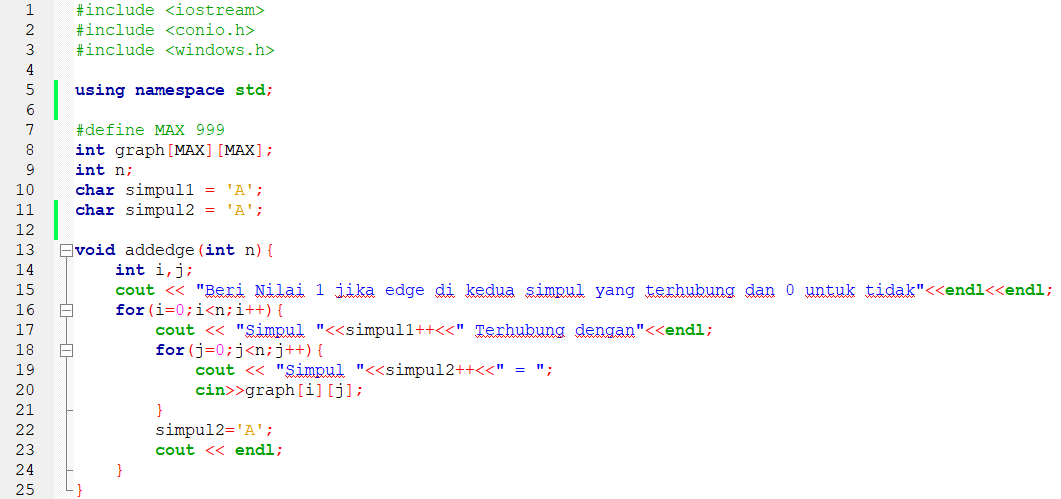
**Contoh 1 :**

**Buat Program untuk merepresentasikan graph diatas dengan konsep adjacency matriks**

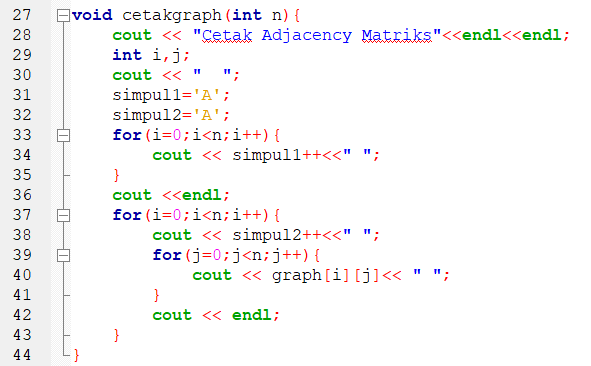


Sumber : https://www.foldertips.com/tik/graph-pohon/

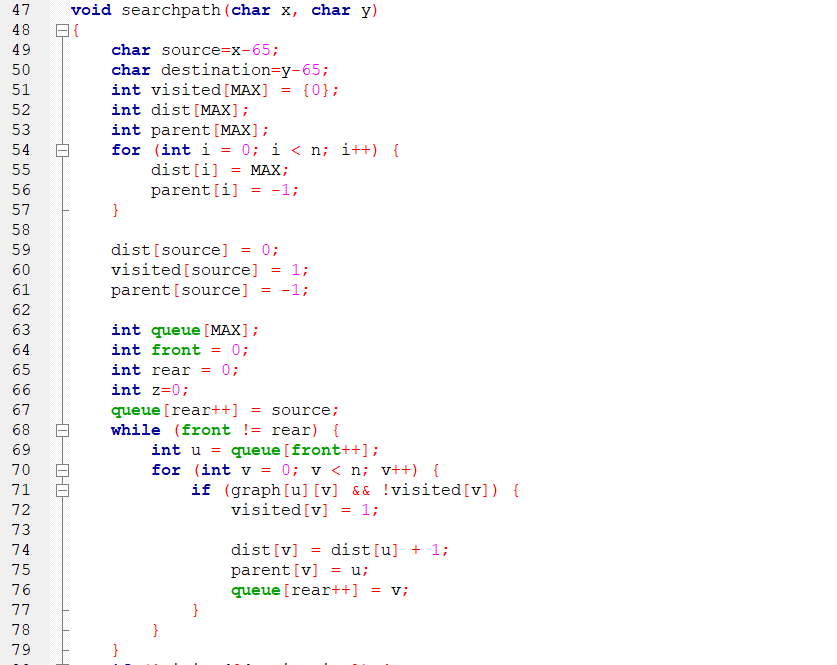
Pertama kita buat fungsi untuk menambahkan sisi (edge)

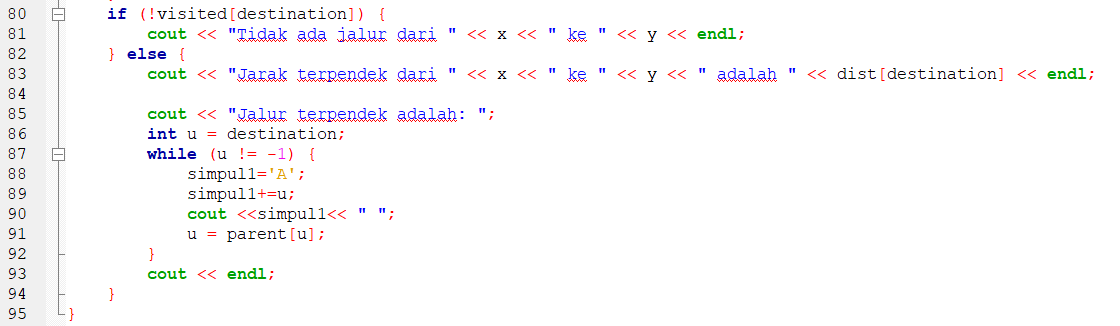


Kemudian buat fungsi untuk menampilkan graph dalam bentuk matriks

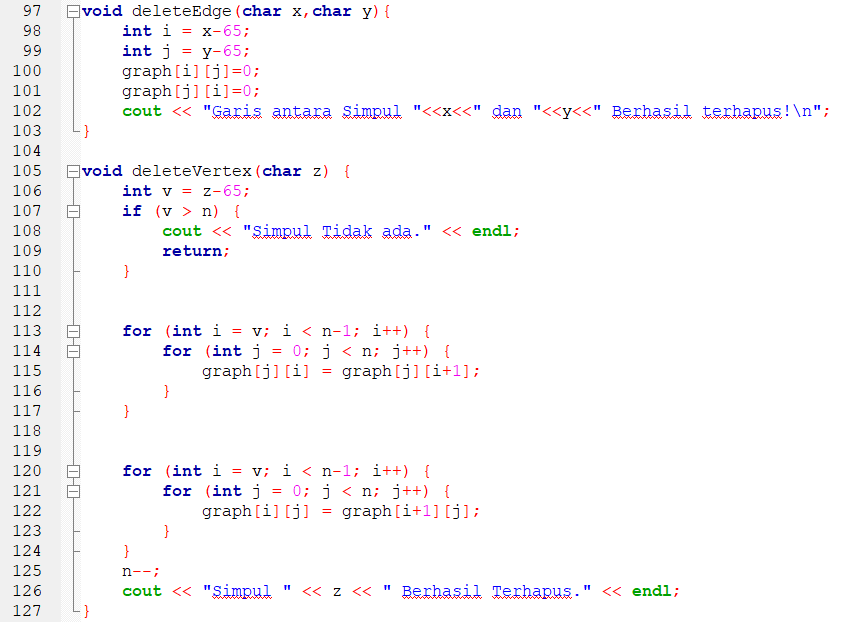


Lalu buat fungsi untuk mencari jalur antar simpul

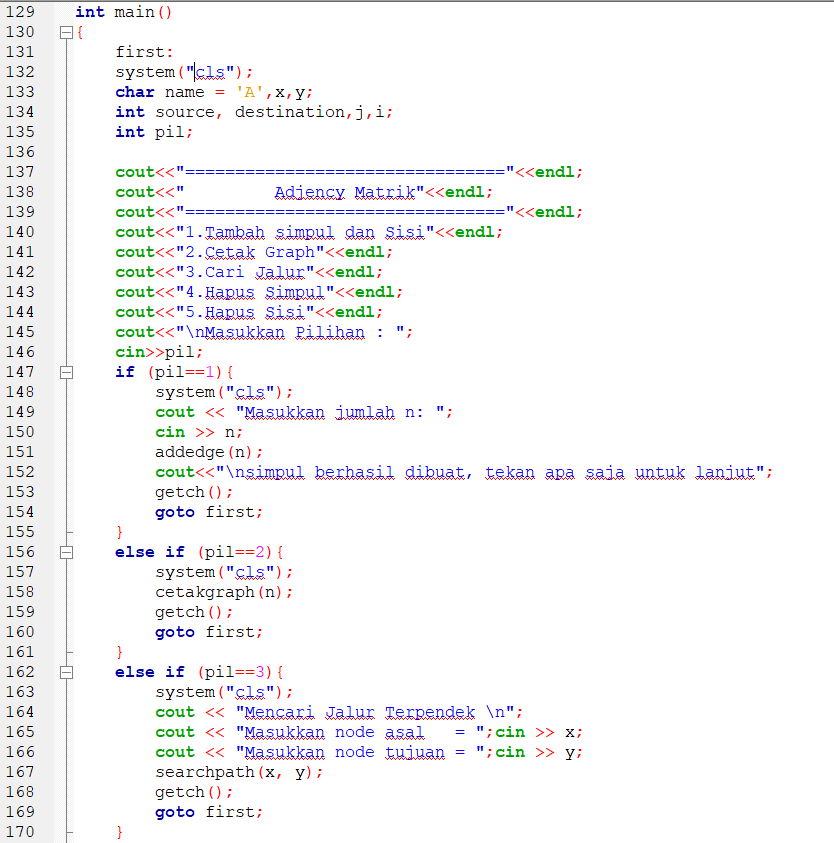


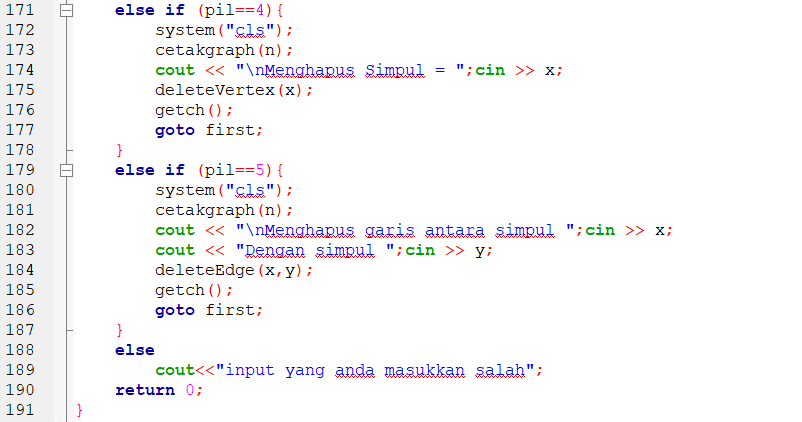


Yang terakhir buat fungsi untuk menghapus sisi dan simpul

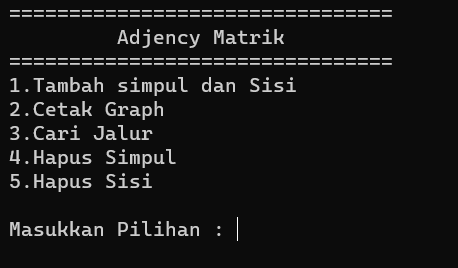


Lalu kita akan panggil semua fungsi yang tadi dibuat pada fungsi utama.

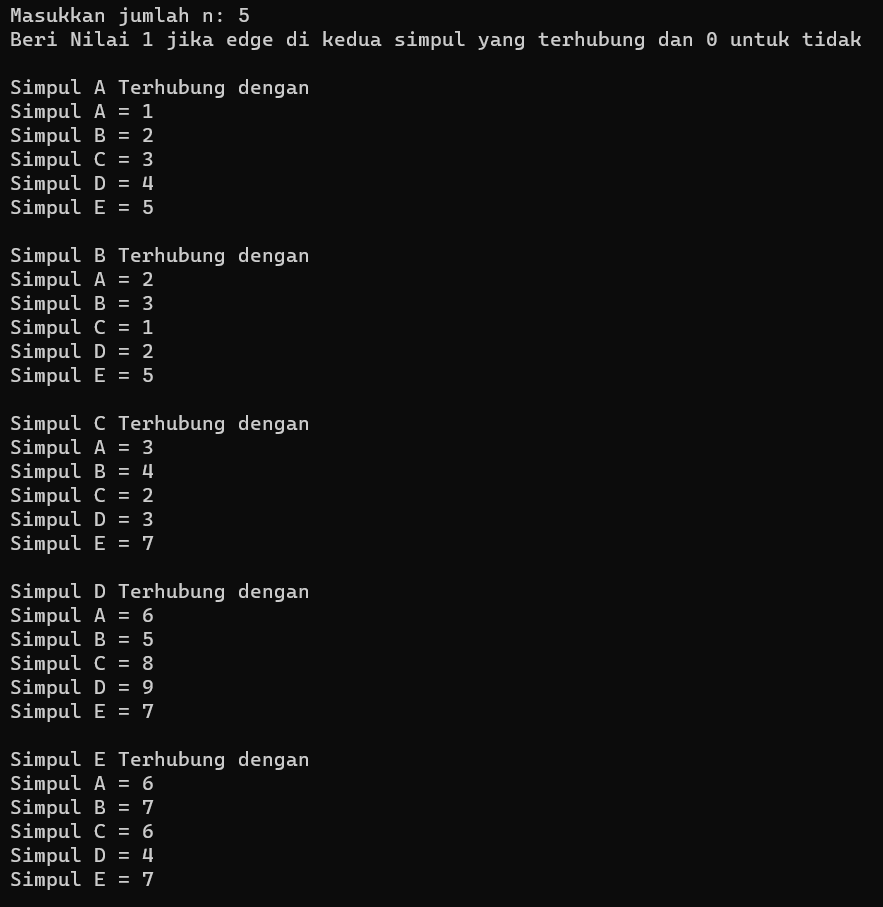




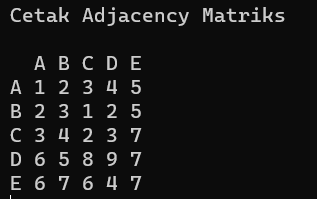
Berikut adalah output dari program Ketika di run



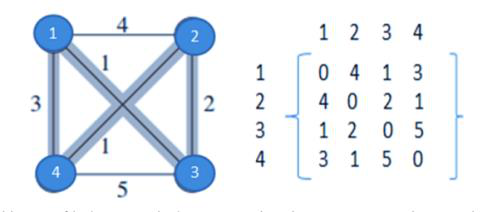
Tambah simpul dan sisi



Cetak Graph



LATIHAN:



Di atas kita dapat melihat grafik dan matriks biaya yang lengkap yang mencakup jarak antara masing-masing desa. Kita dapat mengamati bahwa matriks biaya adalah simetris yang berarti jarak antara desa 2 hingga 3 sama dengan jarak antara desa 3 hingga 2. Masalah di sini adalah penjual keliling ingin mengetahui turnya dengan biaya minimum. Katakanlah Tukang post berada dititik 1 dia berkeliling semua titik lainnya yang belom dikunjungi dan kembali ke titik awal dengan biaya minimum

Buatlah program dengan c++ dengan mengimplementasikan adjacency matriks dan beberapa algoritma untuk menemukan solusi dari Tukang Pos,untuk titik awal berupa inputnya

Contoh Outputnya

