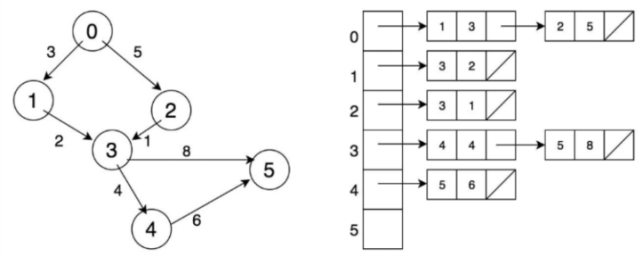
Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

MODUL 13 PRAKTIKUM STRUKTUR DATA

1. Buatlah program untuk untuk graph berarah dan berbobot dan representasinya berikut ini:



Penugasan13.1.c

Untuk membuat program tersebut, maka cukup memodifikasi program **modul13.c.** Bagian yang dimodifikasi antara lain:

- Inputan pada fungsi **main( )** disesuaikan untuk graph diatas sehingga dapat mengakomodir pembuatan adjacency list yang sesuai dengan permintaan soal. Karena terdapat node yang tidak menunjuk ke node lain, maka dest untuk node tersebut di inputkan nilai -1.
- Untuk mengakomodasi tampilan node yang tidak menunjuk ke node lain, maka pada fungsi **createGraph** ( ) diberikan kondisi apabila dest = -1, maka edge tersebut diabaikan atau tidak dimasukkan ke dalam representasi adjacency list.
- Selanjutnya pada fungsi **printGraph ( )**, ketika node tidak menujuk kemanapun, maka hanya ditampilkan bagian src (source) saja, tanpa ada dest dan weigth.

Berikut contoh outputnya sesuai dengan representasi graf pada soal:

```
~~~Representasi Graf dengan Adjacency List~~~

0 -> 1 (3)      0 -> 2 (5)
1 -> 3 (2)
2 -> 3 (1)
3 -> 5 (8)      3 -> 4 (4)
4 -> 5 (5)
5


...Program finished with exit code 0
Press ENTER to exit console.
```

2. Buat program untuk contoh representasi graph tak berarah.

Contoh yang digunakan adalah bentuk graf yang sama persis dengan graf pada soal 1, hanya saja dibuat menjadi graf tak berarah. Sehingga hanya perlu memodifikasi program pada **penugasan13.1.c**. Bagian yang dimodifikasi antara lain:

- Untuk membuat representasi graf tak berarah, kita perlu menambahkan edge yang saling terhubung, yaitu edge dari simpul A ke simpul B dan edge dari simpul B ke simpul A, sehingga pada fungsi createGraph( ) perlu dibuat node baru yang saling menunjuk, yaitu dari src ke dest dan dest ke src.
- Untuk mengurutkan destinasi (dest) yang ditunjuk secara menaik maka ditambahkan perulangan dengan loop while. Hal ini bertujuan untuk membandingkan nilai dest dari node current dengan nilai dest dari node baru (newNode). Jika nilai dest dari node current kurang dari nilai dest dari newNode, kita pindahkan prev dan current ke node selanjutnya dalam adjacency list sampai menemukan posisi yang tepat.
- Setelah keluar dari loop, kita menentukan posisi yang tepat untuk menyisipkan newNode dalam urutan menaik di adjacency list pada simpul src. Jika prev masih NULL, artinya newNode harus ditempatkan di awal adjacency list, sehingga kita menghubungkan newNode ke graph →head[src]. Jika prev tidak NULL, artinya newNode harus disisipkan setelah prev, sehingga kita menghubungkan newNode antara prev dan current.

Berikut output dari program tersebut:

```
~~~Representasi Graf dengan Adjacency List~~~

0 - 1 (3)       0 - 2 (5)
1 - 0 (3)       1 - 3 (2)
2 - 0 (5)       2 - 3 (1)
3 - 1 (2)       3 - 2 (1)       3 - 4 (4)       3 - 5 (8)
4 - 3 (4)       4 - 5 (5)
5 - 3 (8)       5 - 4 (5)


...Program finished with exit code 0
Press ENTER to exit console.
```

Catatan : Program penugasan *terlampir*