

Jobsheet 12

Praktikum Algoritma & Struktur Data



Rafi Ody Prasetyo
(2341720180)

D-IV Teknik Informatika
Politeknik Negeri Malang
Semester 2
2024

Percobaan 1

1. SS Hasil Percobaan

- Run langkah 14

```
InDegree dari Gedung A: 0
OutDegree dari Gedung A: 2
Degree dari Gedung A: 2
Gedung A terhubung dengan
C (100m),
B (50m),

Gedung B terhubung dengan
D (70m),

Gedung C terhubung dengan
D (40m),

Gedung D terhubung dengan
E (60m),

Gedung E terhubung dengan
F (80m),
```

- Run langkah 17

```
Gedung A terhubung dengan
C (100m),
B (50m),

Gedung C terhubung dengan
D (40m),

Gedung D terhubung dengan
E (60m),

Gedung E terhubung dengan
F (80m),
```

2. Github

<https://github.com/rafiody16/Praktikum-Algoritma-dan-Struktur-Data--smt-2-/tree/main/Jobsheet%2012/Percobaan1>

Pertanyaan Percobaan 1

1. Perbaiki kode program Anda apabila terdapat error atau hasil kompilasi kode tidak sesuai!

Jawab:

```
CodeRunner: Run | Explain | Generate Javadoc | X
public void remove(int index) {
    Node21 current = head;
    while (current != null) {
        if (current.data == index) {
            if (current.prev != null) {
                current.prev.next = current.next;
            } else {
                head = current.next;
            }
            if (current.next != null) {
                current.next.prev = current.prev;
            }
            break;
        }
        current = current.next;
    }
    size--;
}
```

Menambahkan size — pada method remove agar langkah 17 dapat dijalankan.

Sebelum ditambahkan size – apabila kode dirun akan muncul error seperti berikut.

```
InDegree dari Gedung A: 0
OutDegree dari Gedung A: 2
Degree dari Gedung A: 2
Gedung A terhubung dengan
C (100m),
B (50m),

Gedung B terhubung dengan
Exception in thread "main" java.lang.NullPointerException: Cannot read field "data" because "tmp" is null
    at DoubleLinkedList21.get(DoubleLinkedList21.java:162)
    at Graph21.printGraph(Graph21.java:62)
    at GraphMain21.main(GraphMain21.java:14)
PS D:\tugas_kuliah\Semester 2\Praktikum Algoritma dan Struktur Data (smt 2)> ^C
PS D:\tugas_kuliah\Semester 2\Praktikum Algoritma dan Struktur Data (smt 2)>
```

2. Pada class Graph, terdapat atribut list[] bertipe DoubleLinkedList. Sebutkan tujuan pembuatan variabel tersebut!

Jawab:

Untuk melakukan pembuatan array list[] yang akan terhubung dengan DoubleLinkedList.

3. Jelaskan alur kerja dari method removeEdge!

Jawab:

```
public void removeEdge(int asal, int tujuan) {
    for (int i = 0; i < vertex; i++) {
        if (i == tujuan) {
            list[asal].remove(tujuan);
        }
    }
}
```

Pertama program akan melakukan pencarian dengan perulangan, apabila perulangan i mencapai nilai sama dengan nilai tujuan, program akan menghapus nilai yang dituju sesuai dengan tempat asal pada array listnya.

4. Apakah alasan pemanggilan method `addFirst()` untuk menambahkan data, bukan method `add` jenis lain saat digunakan pada method `addEdge` pada class `Graph`?

Jawab:

Dengan menggunakan `addFirst()`, penambahan edge ke dalam graf menjadi lebih cepat dan efisien, memastikan bahwa operasi graf dapat dilakukan dengan kompleksitas waktu yang minimal dan pengelolaan memori yang efektif.

5. Modifikasi kode program sehingga dapat dilakukan pengecekan apakah terdapat jalur antara suatu node dengan node lainnya, seperti contoh berikut (Anda dapat memanfaatkan `Scanner`).

```
Masukkan gedung asal: 2
Masukkan gedung tujuan: 3
Gedung C dan D bertetangga

Masukkan gedung asal: 2
Masukkan gedung tujuan: 5
Gedung C dan F tidak bertetangga
```

Jawab:

Kode:

- Graph

```
public boolean checkInConnection(int asal, int tujuan) {
    for (int i = 0; i < list[asal].size(); i++) {
        if (list[asal].get(i) == tujuan) {
            return true;
        }
    }
    return false;
}

public void checkPath(int asal, int tujuan) {
    if (checkInConnection(asal, tujuan)) {
        System.out.println("Gedung " + (char) ('A' + asal) + " dan " + (char) ('A' + tujuan) + " bertetangga");
    } else {
        System.out.println("Gedung " + (char) ('A' + asal) + " dan " + (char) ('A' + tujuan) + " tidak bertetangga");
    }
}
```

- Main

```
while (true) {
    System.out.print("Masukkan gedung asal: ");
    int gdAsal = Integer.parseInt(sc.nextLine());
    if (gdAsal < 0) {
        break;
    }
    System.out.print("Masukkan gedung tujuan: ");
    int gdTujuan = Integer.parseInt(sc.nextLine());
    if (gdTujuan < 0) {
        break;
    }
    gedung.checkPath(gdAsal, gdTujuan);
    System.out.println();
}
```

Output:

```
Masukkan gedung asal: 2
Masukkan gedung tujuan: 3
Gedung C dan D bertetangga

Masukkan gedung asal: 2
Masukkan gedung tujuan: 5
Gedung C dan F tidak bertetangga
```

Percobaan 2

1. SS Hasil Percobaan

```
Gedung A:
Gedung A (0m), Gedung B (50m), Gedung C (0m), Gedung D (0m),
Gedung B:
Gedung A (60m), Gedung B (0m), Gedung C (70m), Gedung D (0m),
Gedung C:
Gedung A (0m), Gedung B (80m), Gedung C (0m), Gedung D (40m),
Gedung D:
Gedung A (90m), Gedung B (0m), Gedung C (0m), Gedung D (0m),
Hasil setelah penghapusan edge
Gedung A:
Gedung A (0m), Gedung B (50m), Gedung C (0m), Gedung D (0m),
Gedung B:
Gedung A (60m), Gedung B (0m), Gedung C (70m), Gedung D (0m),
Gedung C:
Gedung A (0m), Gedung B (0m), Gedung C (0m), Gedung D (40m),
Gedung D:
Gedung A (90m), Gedung B (0m), Gedung C (0m), Gedung D (0m),
```

2. Github

<https://github.com/rafiody16/Praktikum-Algoritma-dan-Struktur-Data--smt-2-/tree/main/Jobsheet%2012/Percobaan2>

Pertanyaan Percobaan 2

1. Perbaiki kode program Anda apabila terdapat error atau hasil kompilasi kode tidak sesuai!

Jawab:

Merubah nilai dari matriks[asal][tujuan] yang sebelumnya -1 menjadi 0 agar hasil kompilasi sesuai.

```
public void removeEdge(int asal, int tujuan) {  
    matriks[asal][tujuan] = 0;  
}
```

2. Apa jenis graph yang digunakan pada Percobaan 2?

Jawab:

Pada percobaan 2 terdapat jenis directed graph yang ditandai dengan method makeEdge yang dibuat dari satu vertex asal ke vertex tujuan. Dan ada juga weighted graph karena setiap edge memiliki parameter jarak dalam method makeEdge.

3. Apa maksud dari dua baris kode berikut?

```
gdg.makeEdge(1, 2, 70);  
gdg.makeEdge(2, 1, 80);
```

Jawab:

Baris kode tersebut digunakan untuk mengisi nilai asal, tujuan, dan jarak pada method makeEdge.

4. Modifikasi kode program sehingga terdapat method untuk menghitung degree, termasuk inDegree dan outDegree!

Jawab:

Kode:

- MatriksGraph

```
public void degree(int asal) {  
    int k, totalIn = 0, totalOut = 0;  
    //inDegree  
    for (int j = 0; j < vertex; j++) {  
        if (matriks[j][asal] > 0) {  
            ++totalIn;  
        }  
    }  
    //outDegree  
    for (k = 0; k < matriks[asal].length; k++) {  
        if (matriks[asal][k] > 0) {  
            k++;  
        }  
    }  
    totalOut = k;  
    System.out.println("InDegree dari Gedung " + (char) ('A' + asal) + ": " + totalIn);  
    System.out.println("OutDegree dari Gedung " + (char) ('A' + asal) + ": " + totalOut);  
    System.out.println("Degree dari Gedung " + (char) ('A' + asal) + ": " + (totalIn + totalOut));  
}
```

- MainGraph

```
for (int i = 0; i < gdg.vertex; i++) {  
    System.out.println((char) ('A' + i));  
    gdg.degree(i);  
}
```

Output:

```
A  
InDegree dari Gedung A: 2  
OutDegree dari Gedung A: 4  
Degree dari Gedung A: 6  
B  
InDegree dari Gedung B: 1  
OutDegree dari Gedung B: 4  
Degree dari Gedung B: 5  
C  
InDegree dari Gedung C: 1  
OutDegree dari Gedung C: 5  
Degree dari Gedung C: 6  
D  
InDegree dari Gedung D: 1  
OutDegree dari Gedung D: 4  
Degree dari Gedung D: 5
```