**JOBSHEET 15**

**Graph**



**Name**

Sherly Lutfi Azkiah Sulistyawati

**NIM**

2341720241

**Class**

1I

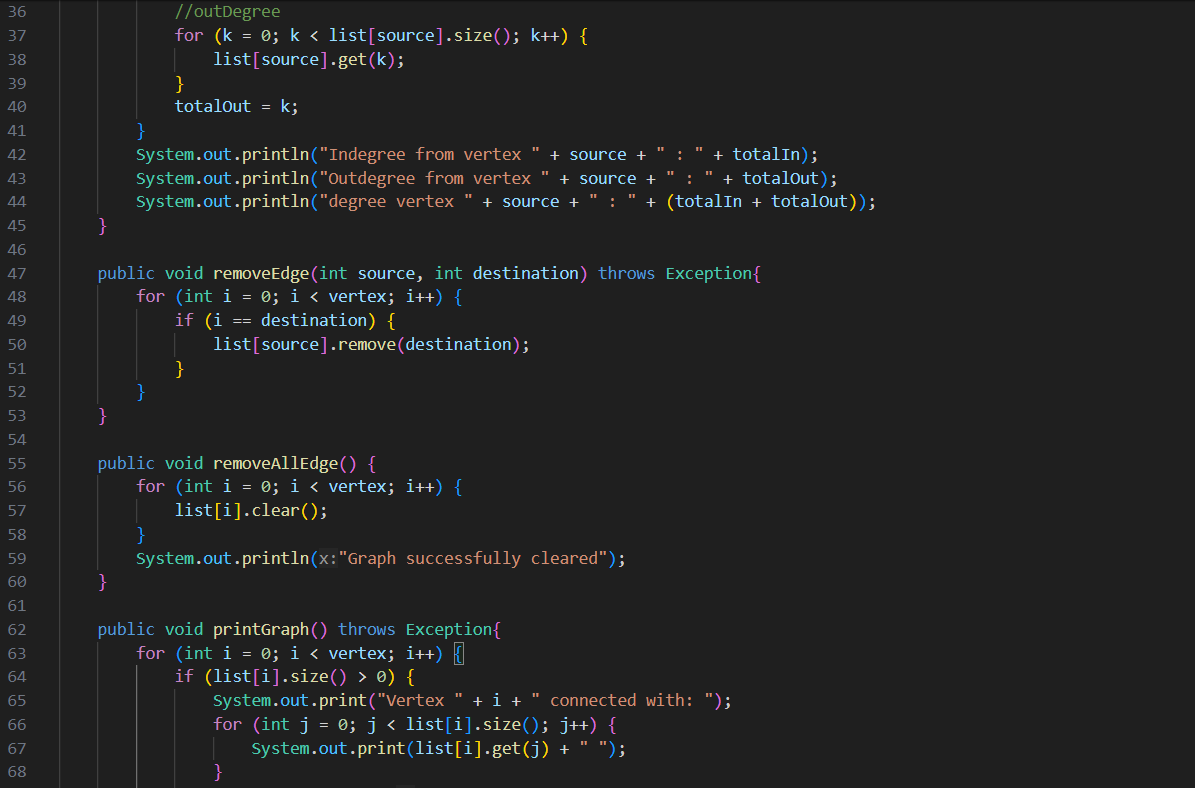
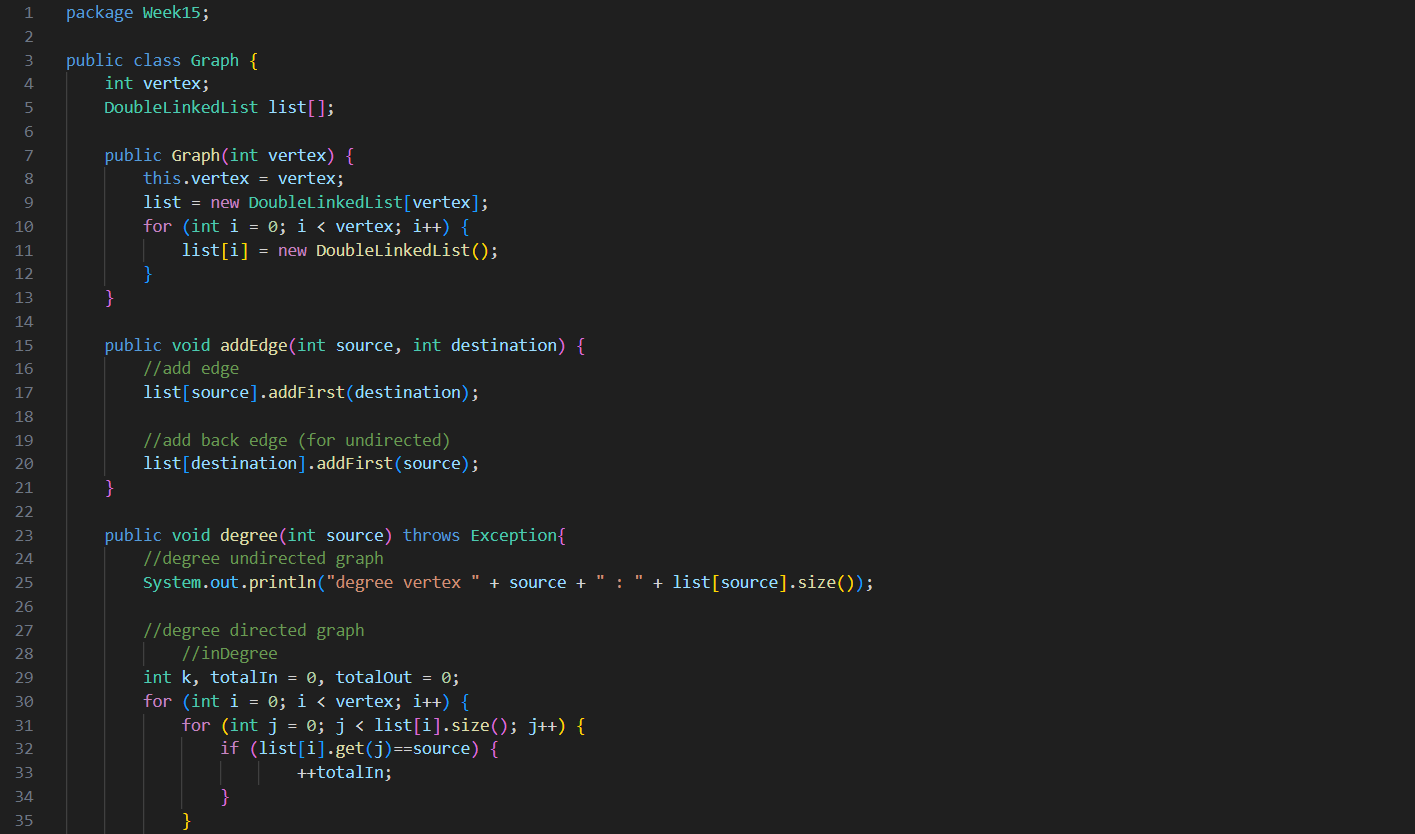
**Major**

Information Technology

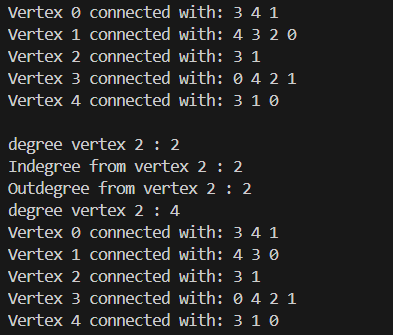
**Study Program**

D4 Informatics Engineering

**Practicum 1**

****

****

****

**Question**

1. Mention 3 kinds of algorithm that uses Graph fundamental, what’s the use of those ?

* Dijkstra's Algorithm: Used for finding the shortest paths between nodes in a graph, which is useful in routing and navigation systems.
* Depth-First Search (DFS): Used for traversing or searching tree or graph data structures. Applications include pathfinding, topological sorting, and solving puzzles.
* Breadth-First Search (BFS): Used for traversing or searching tree or graph data structures. Applications include finding the shortest path in an unweighted graph and peer-to-peer networks.

1. In class Graph, there is an array with LinkedList data type, LinkedList list[]. What’s the aim of this?

* The LinkedList list[] array is used to store the adjacency lists for each vertex in the graph. Each element of the array corresponds to a vertex, and the linked list at each index stores the vertices adjacent to that vertex. This structure allows efficient representation and traversal of adjacency relationships in the graph.

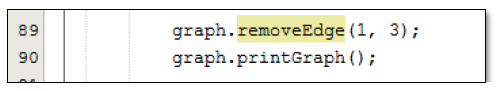
1. What is the reason of calling method **addFirst()** to add data, instead of calling other add methods in Linked list when using method addEdge in **class Graph?**

* The addFirst() method is used to add an edge at the beginning of the linked list to ensure that the most recently added edges are easily accessible. This can simplify some operations, such as ensuring the most recent edges are processed first, which can be helpful in certain graph algorithms.

1. How do we detect prev pointer when we are about to remove an edge of a graph?

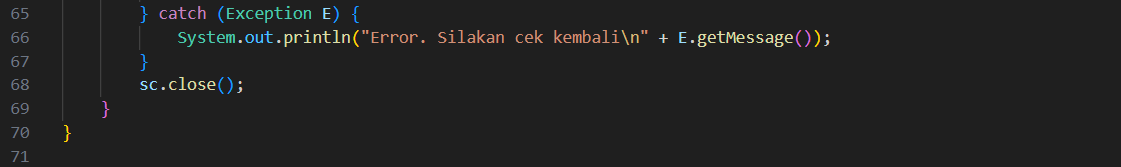
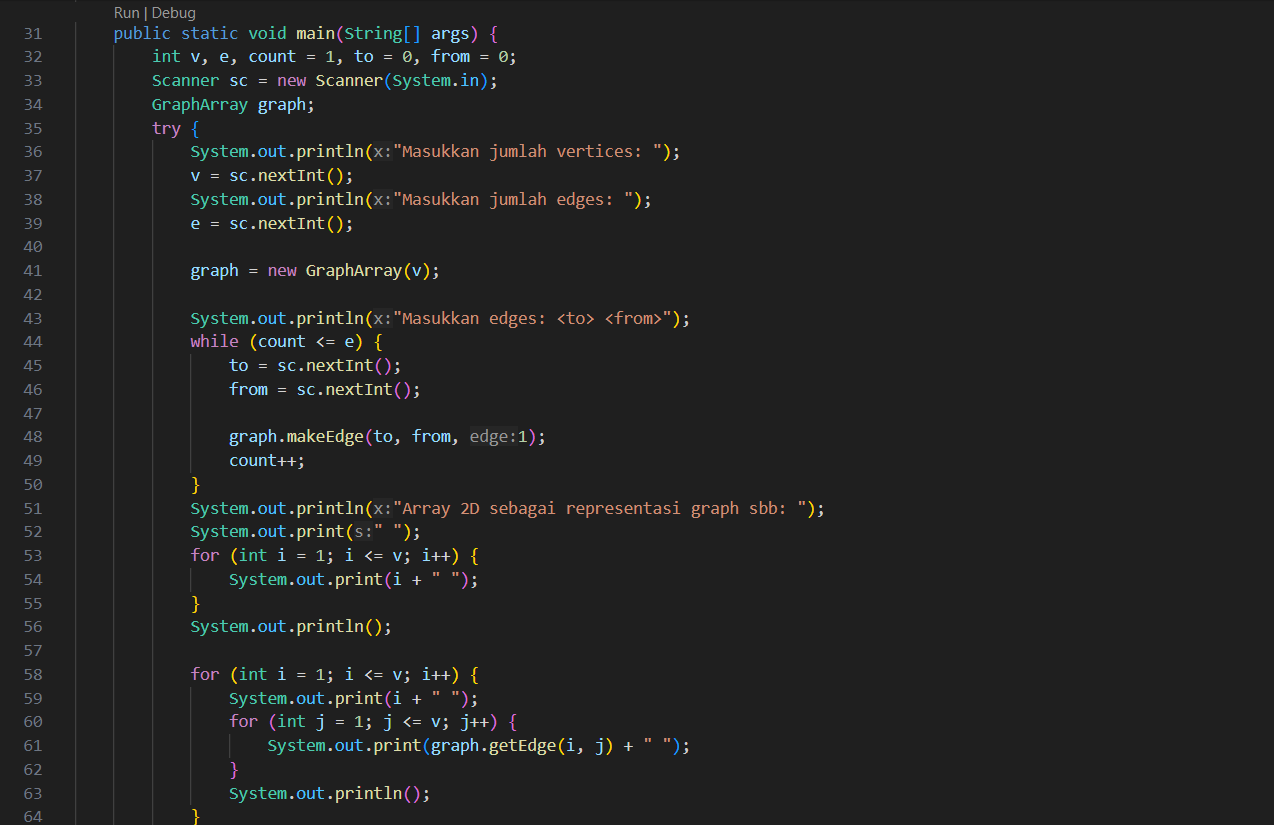
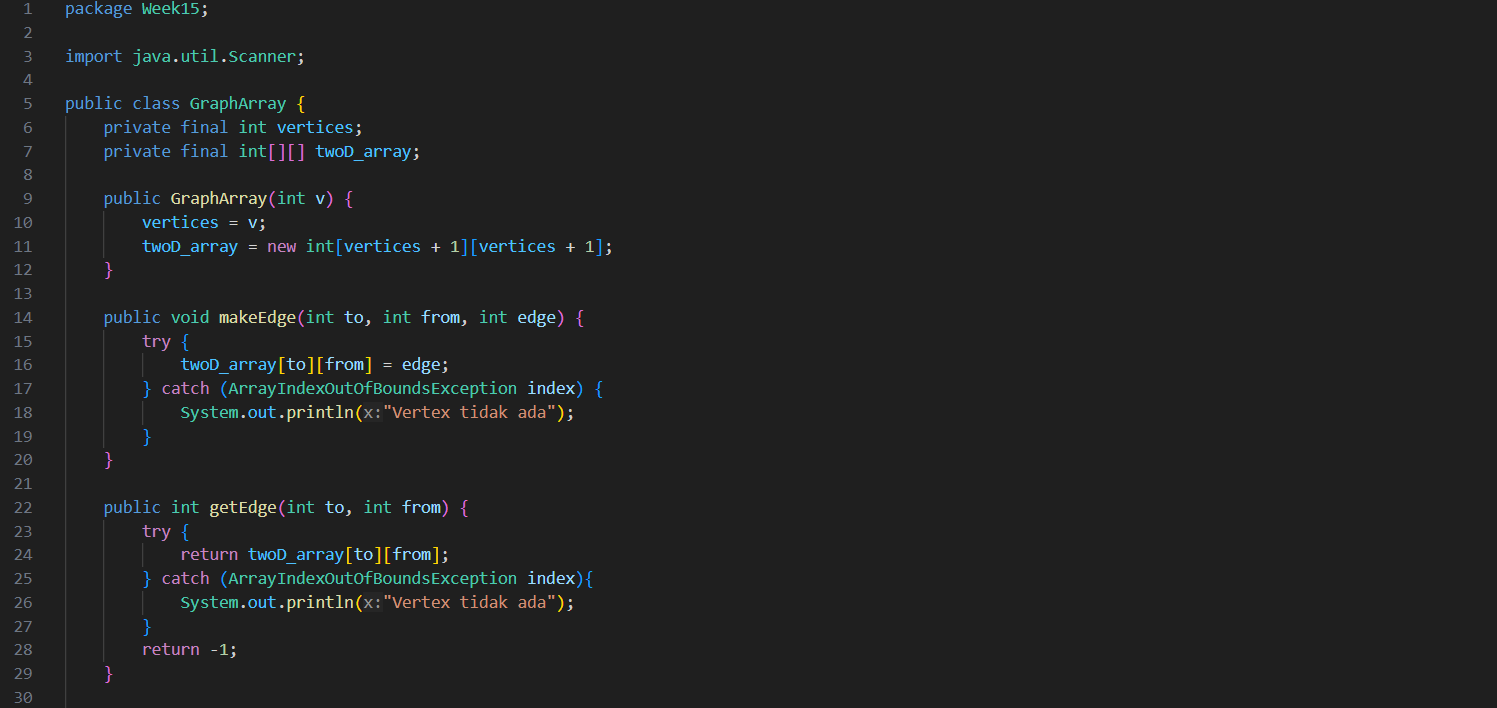
* We can traverse the linked list starting from the head, keeping track of the current node and its previous node. When the node to be removed is found, the prev pointer will be pointing to the node before it, allowing us to update the next pointer of the previous node to bypass the node to be removed.

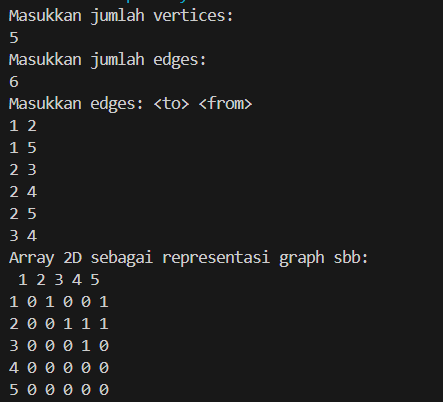
1. Why in practicum 1.2, the 12th step is to remove path that is not the first path to produce the wrong output? What’s the solution?



* Because the linked list traversal may not correctly update the pointers, leading to broken links in the list. The solution is to ensure that the linked list traversal and edge removal are handled correctly by properly updating the prev and next pointers to maintain the integrity of the list.

**Practicum 2**

****

****

**Question**

1. What is the degree difference between directed and undirected graphs?

* In a directed graph, the degree of a vertex is split into in-degree (the number of edges coming into the vertex) and out-degree (the number of edges going out of the vertex).
* In an undirected graph, the degree of a vertex is simply the number of edges connected to it, as there is no direction associated with the edges.

1. In the graph implementation using adjacency matrix. Why does the number of vertices have to be added to 1 in the following array index?

* The number of vertices may need to be incremented by 1 to account for the zero-based indexing of arrays in programming. This ensures that the array can represent all vertices correctly, including the last vertex, without running into index out-of-bounds errors.

1. What is the use of the **getEdge()** method?

* The getEdge() method is used to retrieve and display the edges of the graph. It helps in visualizing the graph structure by providing information on which vertices are connected by edges.

1. What kind of graph were implemented on practicum 1.3 ?

* Practicum 1.3 likely implemented a graph using an adjacency matrix representation, which involves using a 2-dimensional array to represent the connections between vertices.

1. Why does the main method use try-catch Exception?

* The main method uses a try-catch block to handle potential exceptions that may occur during the execution of the program. This can include exceptions such as array index out-of-bounds, null pointer exceptions, or other runtime errors. The try-catch block ensures that the program can handle these errors gracefully without crashing.

**Assignment**

1. Convert the path in 1.2 as an input !
2. Add method **graphType** with boolean as its return type to differentiate which graph is *directed* or *undirected graph*. Then update all the method that relates to **graphType()** (only runs thestatement based on the graph type) in practicum 1.2
3. Modify method **removeEdge()** in practicum 1.2 so that it won’t give the wrong path other than the initial path as an output !
4. Convert vertex’s data type in the graph of practicum 1.2. and 1.3 from integer to generic data type so that it can accepts all basic data type in Java programming language! For example, if the initial vertex are 0,1,2,3, dst. Then the next will be in form of region name, like Malang, Surabaya, Gresik, Bandung, dst.