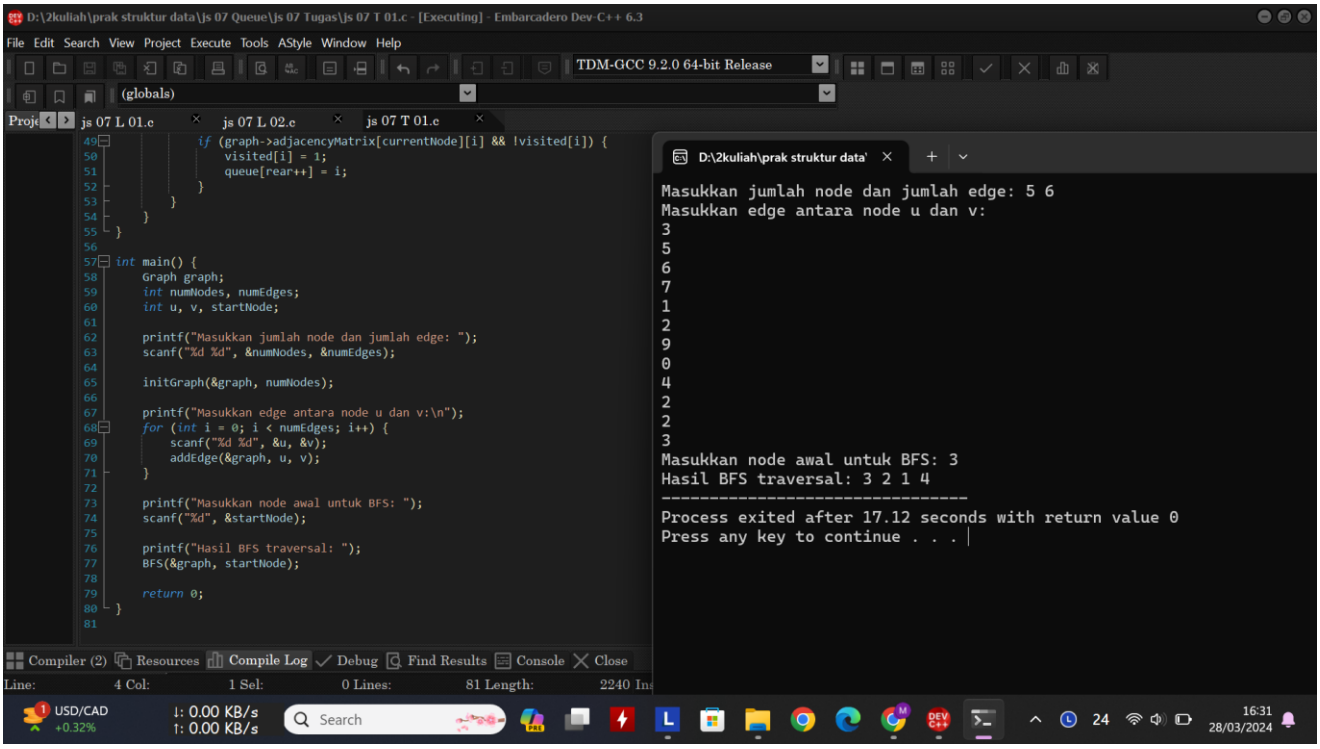
Tugas Job Sheet 7 Queue_Praktikum Struktur data
Muhamad Akbar Fauzan_23343075

```
// js 07 T 01.c
// created by 23343075_muhamad akbar fauzan
// program tentang algoritma Algoritma Breadth First Search (BFS)

#include <stdio.h>
#include <stdlib.h>

#define MAX_NODES 100

// Struktur untuk merepresentasikan graf
typedef struct {
   int numNodes;
   int adjacencyMatrix[MAX_NODES][MAX_NODES];
} Graph;

// Inisialisasi graf dengan jumlah node
void initGraph(Graph *graph, int numNodes) {
   graph->numNodes = numNodes;
   for (int i = 0; i < numNodes; i++) {
      for (int j = 0; j < numNodes; j++) {
         graph->adjacencyMatrix[i][j] = 0;
      }
   }
}

// Menambahkan edge antara node u dan v
void addEdge(Graph *graph, int u, int v) {
   graph->adjacencyMatrix[u][v] = 1;
   graph->adjacencyMatrix[v][u] = 1; // karena graf tidak berarah
}

// Breadth First Search
void BFS(Graph *graph, int startNode) {
   int visited[MAX_NODES] = {0}; // Array untuk menandai node yang sudah dikunjungi
   int queue[MAX_NODES]; // Queue untuk menyimpan node yang akan dikunjungi
   int front = 0, rear = 0; // Variabel untuk mengatur front dan rear queue

   // Mulai dari node awal
   visited[startNode] = 1;
   queue[rear++] = startNode;

   // Selama queue tidak kosong, lakukan BFS
   while (front < rear) {
```

```
        int currentNode = queue[front++];
        printf("%d ", currentNode);

        // Tambahkan semua tetangga yang belum dikunjungi ke dalam queue
        for (int i = 0; i < graph->numNodes; i++) {
            if (graph->adjacencyMatrix[currentNode][i] && !visited[i]) {
                visited[i] = 1;
                queue[rear++] = i;
            }
        }
    }
}

int main() {
    Graph graph;
    int numNodes, numEdges;
    int u, v, startNode;

    printf("Masukkan jumlah node dan jumlah edge: ");
    scanf("%d %d", &numNodes, &numEdges);

    initGraph(&graph, numNodes);

    printf("Masukkan edge antara node u dan v:\n");
    for (int i = 0; i < numEdges; i++) {
        scanf("%d %d", &u, &v);
        addEdge(&graph, u, v);
    }

    printf("Masukkan node awal untuk BFS: ");
    scanf("%d", &startNode);

    printf("Hasil BFS traversal: ");
    BFS(&graph, startNode);

    return 0;
}
```



Penjelasan:

Algoritma Breadth First Search (BFS) adalah algoritma pencarian yang digunakan untuk melakukan penelusuran pada graf atau struktur data lain yang berupa pohon. Algoritma ini dimulai dari sebuah node awal (start node) dan menelusuri semua tetangga dari node tersebut terlebih dahulu sebelum melanjutkan ke tetangga-tetangga yang lebih jauh.

Prinsip queue digunakan dalam algoritma BFS untuk menyimpan node-node yang akan dikunjungi. Node-node ini disimpan dalam sebuah queue dan dikeluarkan sesuai dengan prinsip FIFO (First In First Out). Dengan menggunakan queue, BFS memastikan bahwa node-node akan dikunjungi secara berurutan berdasarkan kedalaman dari node awal. Jika sebuah node telah dikunjungi, maka node tersebut dimasukkan ke dalam queue dan kemudian dikeluarkan dari queue setelah semua tetangganya telah dikunjungi. Hal ini memastikan bahwa semua node pada kedalaman tertentu dikunjungi sebelum melanjutkan ke kedalaman selanjutnya.