Tugas:

Buatlah program tentang algoritma Algoritma Breadth First Search (BFS) dalam bahasa C. Lalu, jelaskan terkait algoritma tersebut dan bagaimana prinsip queue digunakan dalam algoritma tersebut!

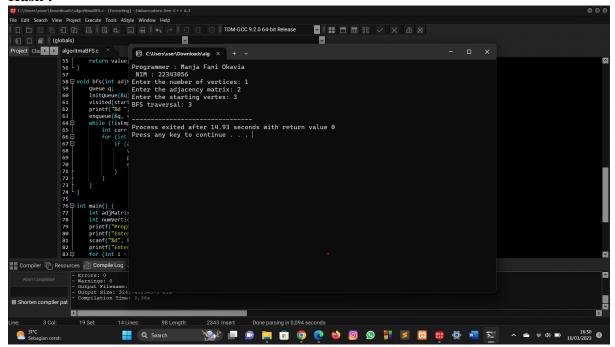
Source code:

```
Nama File : algoritmaBFS.c
            : Manja Fani Oktavia
           : 22343056
  Nim
*/
#include <stdio.h>
#include <stdlib.h>
#define MAX QUEUE SIZE 100
#define MAX VERTEX NUM 100
typedef struct {
  int data[MAX_QUEUE_SIZE];
  int front, rear;
} Queue;
void initQueue(Queue *q) {
  q->front = -1;
  q->rear = -1;
}
int isEmpty(Queue *q) {
  return q->front == -1;
}
int isFull(Queue *q) {
  return q->rear == MAX QUEUE SIZE - 1;
}
void enqueue(Queue *q, int value) {
  if (isFull(q)) {
    printf("Queue overflow\n");
    exit(1);
  if (isEmpty(q)) {
    q->front = 0;
  q->rear++;
  q->data[q->rear] = value;
```

```
}
int dequeue(Queue *q) {
  if (isEmpty(q)) {
    printf("Queue underflow\n");
    exit(1);
  int value = q->data[q->front];
  if (q->front == q->rear) {
    q->front = -1;
    q->rear = -1;
  } else {
    q->front++;
  return value;
}
void bfs(int adjMatrix[MAX VERTEX NUM][MAX VERTEX NUM], int
visited[MAX VERTEX NUM], int start, int numVertices) {
  Queue q;
  initQueue(&q);
  visited[start] = 1;
  printf("%d ", start);
  enqueue(&q, start);
  while (!isEmpty(&q)) {
    int curr = dequeue(&q);
    for (int i = 0; i < numVertices; i++) {
       if (adjMatrix[curr][i] == 1 &&!visited[i]) {
         visited[i] = 1;
         printf("%d ", i);
         enqueue(&q, i);
       }
    }
int main() {
  int adjMatrix[MAX VERTEX NUM][MAX VERTEX NUM],
visited[MAX_VERTEX_NUM];
  int numVertices, start;
  printf("Programmer: Manja Fani Okavia\n NIM: 22343056\n");
  printf("Enter the number of vertices: ");
  scanf("%d", &numVertices);
  printf("Enter the adjacency matrix: ");
  for (int i = 0; i < numVertices; i++) {
    for (int j = 0; j < \text{numVertices}; j++) {
```

```
scanf("%d", &adjMatrix[i][j]);
}
printf("Enter the starting vertex: ");
scanf("%d", &start);
for (int i = 0; i < numVertices; i++) {
    visited[i] = 0;
}
printf("BFS traversal: ");
bfs(adjMatrix, visited, start, numVertices);
printf("\n");
return 0;
}</pre>
```

Hasil:



Penjelasan singkat tentang algoritma BFS:

Algoritma BFS adalah salah satu algoritma penelusuran grafik (graph traversal) yang digunakan untuk mengunjungi setiap simpul pada grafik atau pohon secara sistematis, dengan mengunjungi semua simpul yang terhubung pada level yang sama sebelum melanjutkan ke simpul yang terhubung pada level berikutnya. Algoritma ini mengimplementasikan konsep pengunjungan terlebih dahulu pada level yang lebih rendah sebelum pengunjungan pada level yang lebih tinggi (level-order traversal).

Prinsip queue digunakan dalam algoritma BFS untuk memastikan bahwa simpul yang ditelusuri pada level yang sama diproses secara urut (FIFO - First In First Out). Pada setiap langkah BFS, simpul ditambahkan ke dalam antrian atau queue dan setelah pengunjungan simpul tersebut, simpul tersebut dikeluarkan dari queue dan simpul terhubung pada level

berikutnya dimasukkan ke dalam queue. Dalam implementasi program di atas, queue diwakili oleh array queue[], front dan rear digunakan sebagai indeks yang menunjukkan posisi pertama dan terakhir pada queue, serta queue diinisialisasi dengan nilai -1.