

# **LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : PERANCANGAN DAN ANALISIS ALGORITMA  
Kelas : 3IA11  
Praktikum ke- : 4  
Tanggal : 14/1/2025  
Materi : BFS  
NPM : 51422161  
Nama : MUHAMMAD TARMIDZI BARIQ  
Ketua Asisten : MURAD  
Jumlah Lembar : 3



**LABORATORIUM TEKNIK INFORMATIKA**  
**UNIVERSITAS GUNADARMA**

**2025**

Buatlah kode implementasi BFS kemudian jelaskan, masing- masing praktikan tidak boleh sama.

```
1 #include <iostream>
2 #include <list>
3 #include <queue>
4 #include <vector>
5 using namespace std;
6
7 class Graph {
8     int V; // Jumlah vertex
9     vector<list<int>> adj; // Vector of adjacency lists
10
11 public:
12     // Constructor
13     Graph(int V) {
14         this->V = V;
15         adj.resize(V);
16     }
17
18     // Function untuk menambah edge ke graph
19     void addEdge(int v, int w) {
20         adj[v].push_back(w);
21     }
22
23     // BFS traversal dimulai dari vertex s
24     void BFS(int s) {
25         // Tandai semua vertex sebagai not visited
26         vector<bool> visited(V, false);
27
28         // Buat queue untuk BFS
29         queue<int> queue;
30
31         // Tandai current node sebagai visited dan masukkan ke queue
32         visited[s] = true;
33         queue.push(s);
34
35         while (!queue.empty()) {
36             // Dequeue vertex dari queue dan print
37             s = queue.front();
38             cout << s << " ";
39             queue.pop();
40
41             // Get semua adjacent vertices dari vertex yang di-dequeue
42             // Jika adjacent belum dikunjungi, maka tandai visited
43             // dan masukkan ke queue
44             for (auto adjacent : adj[s]) {
45                 if (!visited[adjacent]) {
46                     visited[adjacent] = true;
47                     queue.push(adjacent);
48                 }
49             }
50         }
51     }
52 }
```

```

48         }
49     }
50 }
51 }
52 };
53
54 int main() {
55     // Buat graph
56     Graph g(4);
57
58     // Tambah edges ke graph
59     g.addEdge(0, 1);
60     g.addEdge(0, 2);
61     g.addEdge(1, 2);
62     g.addEdge(2, 0);
63     g.addEdge(2, 3);
64     g.addEdge(3, 3);
65
66     cout << "Hasil BFS dimulai dari vertex 2:\n";
67     g.BFS(2);
68
69     return 0;
70 }

```

```

Hasil BFS dimulai dari vertex 2:
2 0 3 1

```

```

=== Code Execution Successful ===

```