

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Perancangan dan Analisis Algoritma
Kelas : 3IA13
Praktikum ke- : 4
Tanggal : 8 November 2022
Materi : Metode Brute and Force
NPM : 51420249
Nama : Tiara Puspita
Ketua Asisten : Nadia Cantika Putri
Jumlah Lembar : 7

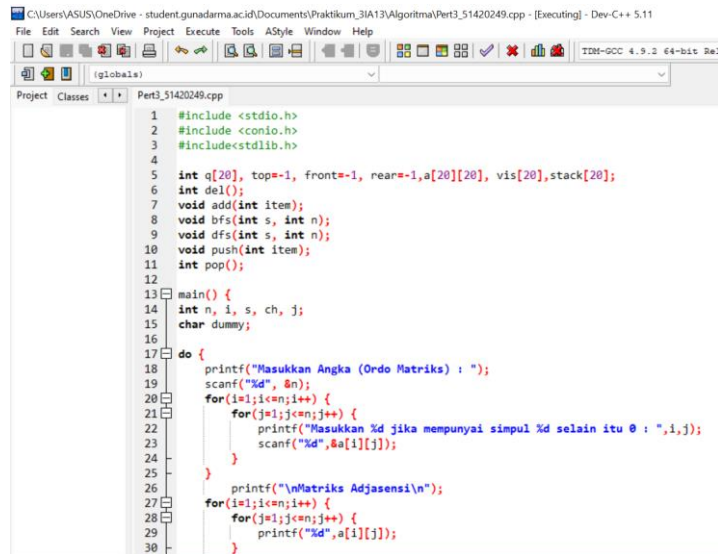


LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

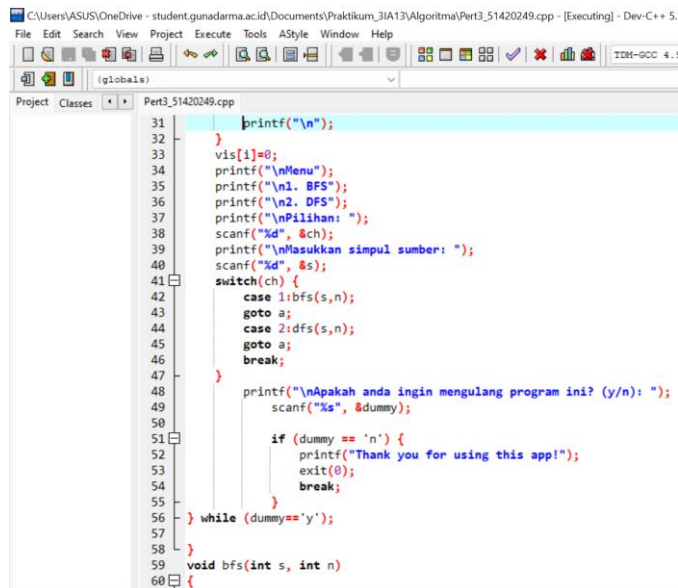
2022

Listing Program



```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4
5 int a[20], top=-1, front=-1, rear=-1, a[20][20], vis[20], stack[20];
6 int del();
7 void add(int item);
8 void bfs(int s, int n);
9 void dfs(int s, int n);
10 void push(int item);
11 int pop();
12
13 main() {
14     int n, i, s, ch, j;
15     char dummy;
16
17     do {
18         printf("Masukkan Angka (Ordo Matriks) : ");
19         scanf("%d", &n);
20         for(i=1; i<=n; i++) {
21             for(j=1; j<=n; j++) {
22                 printf("Masukkan %d jika mempunyai simpul %d selain itu 0 : ", i, j);
23                 scanf("%d", &a[i][j]);
24             }
25         }
26         printf("\nMatriks Adjasensi\n");
27         for(i=1; i<=n; i++) {
28             for(j=1; j<=n; j++) {
29                 printf("%d", a[i][j]);
30             }
31         }
32     } while (dummy != 'y');
```

Gambar 1. Coding split



```
31         printf("\n");
32     }
33     vis[i]=0;
34     printf("\nMenu");
35     printf("\n1. BFS");
36     printf("\n2. DFS");
37     printf("\nPilihan: ");
38     scanf("%d", &ch);
39     printf("\nMasukkan simpul sumber: ");
40     scanf("%d", &s);
41     switch(ch) {
42         case 1: bfs(s, n);
43             goto a;
44         case 2: dfs(s, n);
45             goto a;
46         break;
47     }
48     printf("\nApakah anda ingin mengulang program ini? (y/n): ");
49     scanf("%s", &dummy);
50
51     if (dummy == 'n') {
52         printf("Thank you for using this app!");
53         exit(0);
54         break;
55     }
56 } while (dummy == 'y');
57
58 void bfs(int s, int n)
59 {
60 }
```

Gambar 2 Coding split

```

61 int p,i;
62 add(s);
63 vis[s]=1;
64 p=del();
65 if(p!=0)
66 printf("%d",p);
67 while(p!=0)
68 {
69 for(i=1;i<=n;i++)
70 if((a[p][i] !=0) && (vis[i]==0))
71 {
72 add(i);
73 vis[i]=1;
74 }
75 p=del();
76 if(p!=0)
77 printf("%d",p);
78 }
79 for(i=1;i<=n;i++)
80 if(vis[i]==0)
81 bfs(i,n);
82 }
83 void add(int item)
84 {
85 if(rear==19)
86 printf("antrian full");
87 else
88 if(rear==-1)
89 {
90 q[++rear]=item;

```

Gambar 3 Coding split

```

91 front++;
92 }
93 else
94 q[++rear]=item;
95 }
96 int del()
97 {
98 int k;
99 if((front>rear)|| (front==-1))
100 return(0);
101 else
102 {
103 k=q[front++];
104 return(k);
105 }
106 }
107 void dfs(int s, int n)
108 {
109 int i,k;
110 push(s);
111 vis[s]=1;
112 k=pop();
113 if(k!=0)
114 printf("%d", k);
115 while(k!=0)
116 {
117 for(i=1;i<=n;i++)
118 if((a[k][i] !=0)&&(vis[i]==0))
119 {
120 push(i);
121 vis[i]=1;
122 }
123 k=pop();
124 if(k!=0)
125 printf("%d", k);
126 }
127 for(i=1;i<=n;i++)
128 if(vis[i]==0)
129 dfs(i,n);
130 }
131 void push(int item)
132 {
133 if(top==19)
134 printf("stack overflow");
135 else
136 stack[++top]=item;
137 }
138 int pop(){
139 int k;
140 if(top==-1)
141 return(0);
142 else{
143 k=stack[top--];
144 return(k);
145 }}

```

Gambar 4 Coding split

```
C:\Users\ASUS\OneDrive - student.gunadarma.ac.id\Documents\Praktikum_3IA13\Algoritma\Pert3_51420249.exe
Masukkan Angka (Ordo Matriks) : 3
Masukkan 1 jika mempunyai simpul 1 selain itu 0 : 1
Masukkan 1 jika mempunyai simpul 2 selain itu 0 : 0
Masukkan 1 jika mempunyai simpul 3 selain itu 0 : 0
Masukkan 2 jika mempunyai simpul 1 selain itu 0 : 0
Masukkan 2 jika mempunyai simpul 2 selain itu 0 : 1
Masukkan 2 jika mempunyai simpul 3 selain itu 0 : 0
Masukkan 3 jika mempunyai simpul 1 selain itu 0 : 0
Masukkan 3 jika mempunyai simpul 2 selain itu 0 : 0
Masukkan 3 jika mempunyai simpul 3 selain itu 0 : 1

Matriks Adjasensi
100
010
001

Menu
1. BFS
2. DFS
Pilihan: 1

Masukkan simpul sumber: 4
4123
```

Gambar 5 Output Program

Algoritma Traversal

Penggunaan algoritma traversal digunakan untuk program menelusuri tree. DFS atau Depth First search yang berarti menelusuri secara mendalam dimulai dari paling kiri sedangkan BFS atau Breadth First Search yaitu menelusuri tree secara melebar.