

## Contoh Praktikum Algoritma dan Struktur Data



Nama : Agil Deriansyah Hasan  
Nim : 4522210125

Dosen Pengajar :

Dra.SRI REZEKI CANDRA NURSARI,M.Kom  
Prak. Algoritma dan Struktur Data - I

**S1-Teknik Informatika  
Fakultas Teknik  
Universitas Pancasila 2023/2024**

## cnthprak12-1

```

1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4
5  int cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];
6  int main(){
7      int m;
8      cout<<"Masukkan Vertex = "; cin>> n;
9      cout<<"Masukkan Busur = "; cin>> m;
10     cout<<"Busur"<<endl;
11     for(k=1;k<=m;k++){
12         cin >>i>>j;
13         cost[i][j]=1; }
14     cout<<"Inisialisasi Vertex = ";
15     cin>>v;
16     cout<<"DFS Order Vertex = ";
17     cout<<v<<" ";
18     visited[v]=1;
19     k=1;
20     while(k<n){
21         for(j=n;j>=1;j--){
22             if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1 ) {
23                 visit[j]=1;
24                 stk[top]=j;
25                 top++;}
26             v=stk[--top];
27             cout<<v<<" ";
28             k++;
29             visit[v]=0;
30             visited[v]=1;
31             cin.get();
32     }
33     return 0;}

```

```

F:\>g++ cnthprak12-1.cpp -o 1
F:\>1
Masukkan Vertex = 4
Masukkan Busur =4
Busur
1 2
1 3
2 4
3 4
Inisialisasi Vertex = 1
DFS Order Vertex =1 2 4 3
F:\>

```

### Pseudocode :

Kamus/Deklarasi Variabel fungsi main  
cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10,  
m = int

Algoritma/Deskripsi fungsi utama

input n

input m

for(k=1;k<=m;k++)

    cin >>i>>j

    cost[i][j]=1

endfor

input v

print v

visited[v]=1

k=1

while(k<n)

    for(j=n;j>=1;j--)

        if(cost[v][j]!=0 && visited[j]!=1&&visit[j]!=1 )

            visit[j]=1

            stk[top]=j

            top++

        endif

        v=stk[--top]

        print v

        k++

    visit[v]=0

    visited[v]=1

endwhile

### Algoritma :

1. Memasukkan/Menginput Nilai n
2. Memasukkan/Menginput Nilai m
3. Selama (k=1) maka kerjakan baris 4s.d6
4. Memasukkan/Menginput Nilai i
5. Memasukkan/Menginput Nilai j
6. cost[i][j]=1
7. Memasukkan/Menginput Nilai v
8. Mencetak/Menampilkan Nilai v
9. visited[v]=1
10. k=1
11. Selama(k<n) maka kerjakan baris 12 s.d 21
12. Selama (j=n)
13. Jika(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1) maka kerjakan baris 14s.d16
14. visit[j]=1
15. stk[top]=j
16. top++
17. v=stk[--top]
18. Mencetak/Menampilkan Nilai v
19. k++
20. visit[v]=0
21. visited[v]=1
22. Selesai

## cnthprak12-2

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
7  void bfs(int v){
8      for(i=1;i<=n;i++){
9          if(a[v][i] && !visited[i])
10             q[++r]=i;
11             if(f<=r){
12                 visited[q[f]]=1;
13                 bfs(q[f++]);
14             }
15     }
16
17     int main(){
18         int v;
19         cout<<"Masukkan Matriks atau Vertex : "; cin >> n;
20         for(i=1;i<=n;i++){
21             q[i]=0;
22             visited[i]=0;
23             cout<<"Masukkan Data Graph Dalam Bentuk Matriks [nxn] : "<<endl;
24             for(i=1;i<=n;i++){
25                 for(j=1;j<=n;j++){
26                     cin >> a[i][j];
27                 }
28             }
29             cout<<"Masukkan Vertex Awal : "; cin >> v;
30             bfs(v);
31             cout<<" Node Yang Dapat Dicapai : ";
32             for(i=1;i<=n;i++){
33                 if(visited[i])
34                     cout<<i<<" ";
35                 else
36                 {
37                     cout<<"BFS - Tidak Semua Node Dapat Dijangkau";
38                     break;
39                 }
40             }
41             cout<<endl;
42             cin.get();
43     }

```

```

F:\>g++ cnthprak12-2.cpp -o 1
F:\>1
Masukkan Matriks atau Vertex : 4
Masukkan Data Graph Dalam Bentuk Matriks [nxn] :

1 1 1 1
0 1 0 0
0 0 1 0
0 0 0 1
Masukkan Vertex Awal : 1
Node Yang Dapat Dicapai : 1 2 3 4
F:\>

```

### Pseudocode :

Kamus/Deklarasi Variabel fungsi bfs  
a,q,visited,n,i,j,f,r,v = int

Algoritma/Deskripsi fungsi bfs(v)

```

for(i=1;i<=n;i++)
    if(a[v][i] && !visited[i])
        q[++r]=i
        if(f<=r)
            visited[q[f]]=1
            bfs(q[f++])
        endif
    endif
endfor

```

Kamus/Deklarasi Variabel fungsi utama  
v =int

Algoritma/Deskripsi fungsi utama

```

input n
for(i=1;i<=n;i++)
    q[i]=0
    visited[i]=0
endfor
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        input a[i][j]
    endfor
endfor
input v
bfs(v)
for(i=1;i<=n;i++)
    if(visited[i])
        print i
    else
        print ("BFS - Tidak Semua Node Dapat Dijangkau")
        break
    endif
endfor

```

### Algoritma :

1. Membuat fungsi bfs(v)
2. Selama (i=1)
3. Jika (a[v][i] && !visited[i])
4. q[++r]=i
5. Jika (f<=r) maka kerjakan baris 6 s.d 7
6. visited[q[f]]=1
7. bfs(q[f++])
8. Membuat fungsi utama
9. Menginput/Memasukkan Nilai n
10. Selama (i=1) maka kerjakan baris 11 s.d 13
11. q[i]=0
12. visited[i]=0
13. i++
14. Selama (i=1) maka kerjakan baris 15 s.d 17
15. Selama (j=1) maka kerjakan baris 16
16. j++
17. i++
18. Menginput/Memasukkan Nilai a[i][j]
19. Menginput/Memasukkan Nilai v
20. Memanggil Fungsi bfs[v]
21. Selama (i=1) maka kerjakan baris 22 s.d 23
22. Jika (visited[i])
23. Mencetak/Menampilkan Nilai i
24. break
25. Selesai

## cnthprak12-3

```
cnthprak12-1.cpp | cnthprak12-2.cpp | cnthprak12-3.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
7  int del();
8  void add(int item);
9  void bfs(int s, int n);
10 void dfs(int s, int n);
11 void push(int item);
12 int pop();
13 int main(){
14     int n,i,s,cn,j;
15     cout<<"Masukkan Data : "; cin >> n;
16     for(i=1;i<=n;i++){
17         for(j=1;j<=n;j++){
18             cout<<"Masukkan"<<i<<"Data Jika Mempunyai Simpul"<<j<<"Selain itu ";
19             cin>>a[i][j];
20         }
21     }
22     cout<<"Matrik Adjacency"<<endl;
23     for(i=1;i<=n;i++){
24         for(j=1;j<=n;j++){
25             cout<<a[i][j]<<" ";
26         }
27         cout<<endl;
28     }
29     for(i=1;i<=n;i++){
30         vis[i]=0;
31         cout<<"Menu"<<endl;
32         cout<<"~~~~~"<<endl;
33         cout<<"1. BFS - Breadth First Search"<<endl;
34         cout<<"2. DFS - Depth First Search"<<endl;
35         cout<<"~~~~~"<<endl;
36         cout<<"    Pilihan"<<endl;
37         cout<<"Pilih :"; cin>> cn;
38         cout<<"~~~~~"<<endl;
39         cout<<"Masukkan Simpul Sumber :";cin >> s;
40         switch(cn){
41             case 1:bfs(s,n);
42             case 2:dfs(s,n);
43             case 3:break;
44         }
45         return(0);
46     }
47 }
48
```

```
cnthprak12-1.cpp | cnthprak12-2.cpp | cnthprak12-3.cpp
49 void bfs(int s, int n){
50     int p,i;
51     add(s);
52     vis[s]=1;
53     p=del();
54     if(p!=0)
55         cout<<p<<" ";
56     while(p != 0){
57         for(i=1;i<=n;i++){
58             if((a[p][i]!=0)&&(vis[i]==0)){
59                 add(i);
60                 vis[i]=1;
61             }
62             p=del();
63             if(p!=0)
64                 cout<<p<<" ";
65         }
66         for(i=1;i<=n;i++)
67             if(vis[i]==0)
68                 bfs(i,n);
69     }
70 }
71 void add(int item){
72     if(rear==19){
73         cout<<"Antrian Penuh"<<endl;
74         cout<<"~~~~~"<<endl;
75     }
76     else
77         if(rear==-1){
78             q[++rear]=item;
79             front++;
80         }
81     else
82         q[++rear]=item;
83 }
84
85 int del(){
86     int k;
87     if((front>rear)|| (front==-1))
88         return(0);
89     else{
90         k=q[front++];
91         return(k);
92     }
93 }
94
```

```

93     }
94
95     void dfs(int s, int n){
96         int i,k;
97         push(s);
98         vis[s]=1;
99         k=pop();
100        if(k!=0);
101        cout<<k<<" ";
102        while(k!=0){
103            for(i=1;i<=n;i++){
104                if((a[k][i]!=0)&&(vis[i]==0)){
105                    push(i);
106                    vis[i]=1;
107                }
108                k=pop();
109                if(k!=0)
110                    cout<<k<<" ";
111            }
112            for(i=1;i<=n;i++){
113                if(vis[i]==0)
114                    dfs(i,n);
115            }
116        }
117        void push(int item){
118            if(top==19)
119                cout<<"Stack Overflow"<<endl;
120            else
121                stack[++top]=item;
122        }
123
124        int pop(){
125            int k;
126            if(top==-1)
127                return(0);
128            else{
129                k=stack[top--];
130                return(k);
131            }
132        }

```

F:\>g++ cnthprak12-3.cpp -o 1

```

F:\>1
Masukkan Data : 4
Masukkan1Data Jika Mempunyai Simpul1Selain itu 1
Masukkan1Data Jika Mempunyai Simpul2Selain itu 0
Masukkan1Data Jika Mempunyai Simpul3Selain itu 0
Masukkan1Data Jika Mempunyai Simpul4Selain itu 1
Masukkan2Data Jika Mempunyai Simpul1Selain itu 0
Masukkan2Data Jika Mempunyai Simpul2Selain itu 0
Masukkan2Data Jika Mempunyai Simpul3Selain itu 0
Masukkan2Data Jika Mempunyai Simpul4Selain itu 0
Masukkan3Data Jika Mempunyai Simpul1Selain itu 0
Masukkan3Data Jika Mempunyai Simpul2Selain itu 1
Masukkan3Data Jika Mempunyai Simpul3Selain itu 1
Masukkan3Data Jika Mempunyai Simpul4Selain itu 1
Masukkan4Data Jika Mempunyai Simpul1Selain itu 0
Masukkan4Data Jika Mempunyai Simpul2Selain itu 0
Masukkan4Data Jika Mempunyai Simpul3Selain itu 0
Masukkan4Data Jika Mempunyai Simpul4Selain itu 0
Matrik Adjacency
1 0 0 1
0 0 0 0
0 1 1 1
0 0 0 0
Menu
~~~~~
1. BFS - Breadth First Search
2. DFS - Depth First Search
~~~~~
Pilihan
Pilih :1
~~~~~
Masukkan Simpul Sumber :1
14231

```

### Pseudocode :

Kamus/Deklarasi Variabel fungsi utama  
n,i,s,cn,j = int

Algoritma/Deskripsi fungsi utama

```
input n
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        print i; print j
        input a[i][j]
    endfor
endfor
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        print a[i][j]
    endfor
endfor
for(i=1;i<=n;i++)
    vis[i]=0
    print cn
    print s
    switch(cn)
        case 1:bfs(s,n)
        case 2:dfs(s,n)
        case 3:break
    endswitch
return(0)
endfor
```

Kamus/Deklarasi Variabel fungsi bfs  
s,n,p,i = int

Algoritma/Deskripsi fungsi bfs(s,n)

```
add(s)
vis[s]=1
p=del()
if(p!=0)
    print p
    while (p != 0)
        for(i=1;i<=n;i++)
            if((a[p][i]!=0)&&(vis[i]==0))
                add(i)
                vis[i]=1
            endif
        p=del()
        if(p!=0)
            print p
        endwhile
        for(i=1;i<=n;i++)
            if(vis[i]==0)
                bfs(i,n)
            endif
        endfor
    endwhile
endwhile
```

Kamus/Deklarasi Variabel fungsi add  
item, rear, q, front = int

Algoritma/Deskripsi fungsi add(item)  
if (rear == 19)  
    Print ("Antrian Penuh")  
else  
    if(rear== -1\_  
        q[++rear]=item  
        front  
    else  
        q[++rear]=item  
    endif  
endif

Kamus/Deklarasi Variabel fungsi del  
k, rear, front, q = int

Algoritma/Deskripsi fungsi del  
if((front>rear)|| (front== -1))  
    return(0)  
else  
    k=q[front++]  
    return(k)  
endif

Kamus/Deklarasi Variabel fungsi dfs  
s,n,i,k,vis = int

Algoritma/Deskripsi fungsi dfs(s,n)  
push(s)  
vis[s]=1  
k=pop()  
if (k!=0)  
    print k  
    while (k !=0)  
        for(i=1;i<=n;i++)  
            if((a[k][i]!=0)&&(vis[i]==0))  
                push(i)  
                vis[i]=1  
        endif  
        k=pop()  
        if(k!=0)  
            print k  
    endwhile  
    for(i=1;i<=n;i++)  
        if(vis[i]==0)  
            dfs(i,n)

Kamus/Deklarasi Variabel fungsi push  
item, top =int

Algoritma/Deskripsi fungsi push(item)  
if(top==19)  
    print ("Stack Overflow")  
else  
    stack[++top]=item  
endif

Kamus/Deklarasi Variabel fungsi pop  
k, stack = int

Algoritma/Deskripsi fungsi pop  
if (top== -1)  
    return 0  
else  
    k=stack[top--]  
    return k  
endif



### Algoritma :

1. Membuat fungsi utama
2. Menginput/Memasukkan Nilai n
3. Selama(i=1) maka kerjakan baris 4 s.d 9
4. Selama (j=1) maka kerjakan baris 5 s.d 8
5. Mencetak/Menampilkan Nilai i
6. Mencetak/Menampilkan Nilai j
7. Menginput/Memasukkan Nilai a[i][j]
8. j++
9. i++
10. Selama (i=1) maka kerjakan baris 11 s.d 14
11. Selama (j=1) maka kerjakan baris 12 s.d 13
12. Mencetak/Menampilkan Nilai a[i][j]
13. j++
14. i++
15. Selama (i=1) maka kerjakan baris 16 s.d 24
16. vis[i]=0
17. Menginput/Memasukkan Nilai cn
18. Menginput/Memasukkan Nilai s
19. Switch(cn)
20. case 1:bfs(s,n)
21. case 2:dfs(s,n)
22. case 3:break
23. return 0
24. i++
25. Membuat fungsi bfs(s,n)
26. Memanggil fungsi add(s)
27. vis[s]=1
28. p=Memanggil fungsi del

29. Jika (p != 0)
30. Mencetak/Menampilkan Nilai p
31. Selama (p != 0) maka kerjakan baris 32 s.d 38
32. selama (i=1)
33. Jika ((a[p][i]!=0)&&(vis[i]==0)) maka kerjakan baris 34 s.d 35
34. Memanggil fungsi add(i)
35. vis[i] = 1
36. p=del()
37. Jika (p!=0)
38. Mencetak/Menampilkan Nilai p
39. Selama (i=1)
40. Jika (vis[i]=0)
41. Memanggil fungsi bfs(i,n)
42. Membuat fungsi add(item)
43. Jika (rear==19) maka kerjakan baris 44 kalau tidak baris 45 s.d
44. Mencetak ("Antrian Penuh")
45. Jika (rear==-1) maka kerjakan baris 46 s.d 47 kalau tidak baris 48
46. q[++rea]=item
47. front++
48. q[++rea]=item
49. Membuat fungsi del
50. Jika ((front>rear)||((front==1)))
51. return 0
52. k=q[front++]
53. return k

```
54. Membuat fungsi dfs(s,n)
55. Memanggil fungsi push(s)
56. vis[s]=1
57. Jika (k != 0)
58. Mencetak/Menampilkan Nilai k
59. Selama (k != 0) maka kerjakan baris 60 s.d 63
60. Selama (i=1)
61. Jika ((a[k][i]!=0)&&(vis[i]==0))
62. push(i)
63. vis[i]=1
64. k=pop
65. Jika(k!=0)
66. Mencetak/Menampilkan Nilai k
67. Selama (i=1)
68. Jika (vis[i]==0)
69. Memanggil fungsi dfs(i,n)
70. Membuat fungsi push(item)
71. Jika (top==19)
72. Mencetak ("Stack Overflow")
73. Stack[++top]=item
74. Membuat fungsi pop
75. Jika(top ==-1)
76. return 0
77. k = stack[top-]
78. return k
```