

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**

Contoh 1

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 typedef struct{
6     string npw;
7     string nama_ahs;
8     double nilai;
9 }tdata;
10
11 typedef struct elm{
12     tdata data;
13     elm* next;
14 }tlist;
15
16 typedef struct{
17     tlist* first;
18     tlist* last;
19 }tqueue;
20
21 void buatqueue(tqueue* Q){
22     (*Q).first=NULL;
23     (*Q).last=NULL;
24 }
25
26 int isEmpty(tqueue Q){
27     bool hasil=false;
28     if(Q.first==NULL){
29         hasil=true;
30     }
31     return hasil;
32 }
33
34 int jmlElemen(tqueue Q){
35     int hasil=0;
36     if(Q.first!=NULL){
37         tlist* bantu=Q.first;
38         while(bantu!=NULL){
39             hasil++;
40             bantu=bantu->next;
41         }
42     }
43     return hasil;
44 }
45
46 void del(tqueue Q){
47     if(Q.first!=NULL){
48         tlist* bantu=Q.first;
49         while(bantu!=NULL){
50             tlist* bantu2=bantu->next;
51             delete bantu;
52             bantu=bantu2;
53         }
54         bantu=NULL;
55     }
56     Q.first=NULL;
57     Q.last=NULL;
58 }
59
60 void add(string npw, string nama_ahs, double nilai, tqueue* Q){
61     tlist* info = new tlist;
62     info->data.npw = npw;
63     info->data.nama_ahs = nama_ahs;
64     info->data.nilai = nilai;
65     info->next=NULL;
66
67     if(Q.first==NULL){
68         (*Q).first=info;
69     }
70     else{
71         (*Q).last->next=info;
72     }
73     (*Q).last=info;
74     info->next=NULL;
75 }
76
77 void del(tqueue Q){
78     if(Q.first!=NULL){
79         tlist* bantu=Q.first;
80         while(bantu!=NULL){
81             tlist* bantu2=bantu->next;
82             delete bantu;
83             bantu=bantu2;
84         }
85         bantu=NULL;
86     }
87     Q.first=NULL;
88     Q.last=NULL;
89 }
90
91 void cetakqueue(tqueue Q){
92     if(Q.first==NULL){
93         cout<<"Queue Kosong"<<endl;
94     }
95     else{
96         tlist* bantu=Q.first;
97         while(bantu!=NULL){
98             cout<<"Elemen Ke : "<<bantu->data.npw<<endl;
99             cout<<"Nomor Pokok Mahasiswa : "<<bantu->data.nama_ahs<<endl;
100             cout<<"Nama Mahasiswa : "<<bantu->data.nilai<<endl;
101             bantu=bantu->next;
102         }
103     }
104 }
105
106 int main(){
107     tqueue Q;
108     buatqueue(&Q);
109     cetakqueue(Q);
110     cout<<endl;
111     cout<<endl;
112     cout<<endl;
113     cout<<endl;
114     add("452210666","Budi",88.75,&Q);
115     add("452210777","Susi",78.85,&Q);
116     add("452210888","Nuri",98.65,&Q);
117     add("452210999","Bimo",68.55,&Q);
118     add("452210555","Arif",78.85,&Q);
119     add("452210444","Rido",98.65,&Q);
120     add("452210222","Ella",68.55,&Q);
121     cetakqueue(Q);
122     cout<<endl;
123     cout<<endl;
124     del(&Q);
125     cetakqueue(Q);
126     cout<<endl;
127     cout<<endl;
128     del(&Q);
129     cetakqueue(Q);
130     cout<<endl;
131     return 0;
132 }

```

Pseudocode

Kamus/Deklarasi Variabel fungsi buatSTKosong

-

Algoritma/Dekripsi fungsi buatSTKosong(stack *S)
(*S).top = NULL

Kamus/Deklarasi Variabel fungsi isKosong
hasil = bool

Algoritma/Dekripsi fungsi isKosong(stack S)
hasil = false
if(S.top==NULL)
 hasil=true
endif
return hasil

Kamus/Deklarasi Variabel fungsi jmlelemen
hasil = int

Algoritma/Dekripsi fungsi jmlelemen(stack S)
hasil = 0
if(S.top != NULL)
 elemen *bantu
 bantu = S.top
 while(bantu != NULL)
 hasil=hasil+1
 bantu=bantu->next
 endwhile
endif
return hasil

Kamus/Deklarasi Variabel fungsi push
npm,nama_mhs = char
nilai = double

Algoritma/Dekripsi fungsi push (npm, nama_mhs, nilai, stack *S)
elemen *info

```
info = new elemen
info -> kontainer.npm = npm
info -> kontainer.nama_mhs = nama_mhs
info -> kontainer.nilai = nilai
if((*S).top == NULL)
    print("Stack Penuh")
info -> next = NULL

else
    info->next=(*S).top
endif
(*S).top = info
info = NULL
```

Kamus/Deklarasi Variabel fungsi pop

-

Algoritma/Dekripsi fungsi pop(stack *S)
if((*S).top != NULL)
 elemen *hapus = (*S).top
 if(jmlelemen(*S)==1)
 (*S).top=NULL
 else
 (*S).top =(*S).top -> next
 hapus -> next = NULL
 delete hapus
 else
 print("Stack Kosong")

Kamus/Deklarasi Variabel fungsi cetakstack
i = int

Algoritma/Dekripsi fungsi cetakstack(stack S)
if(S.top != NULL)
 print("Menampilkan Stack")
 elemen *bantu = S.top
 while(bantu != NULL)
 print(i)
 print(bantu->kontainer.npm)
 print(bantu->kontainer.nama_mhs)
 print(bantu->kontainer.nilai)
 bantu=bantu->next
 else
 print("Stack Kosong")

Kamus/Deklarasi Variabel fungsi
npm, nama_mhs : char
nilai : double
top : int

Algoritma/Dekripsi fungsi
typedef struktur (struct(npm,nama_mhs,nilai))
 type NilaiMK
typedef struktur(kontainer, next)
 type elemen
typedef struktur (struct{top, NilaiMk Dat[j]})
 type stack
 stack S
buatST(&S) ; CetakStack(S)
 push("4523210666","Budi",88.75,&S);
 push("4523210777","Susi",78.85,&S);
 push("4523210888","Nuri",98.65,&S);
 push("4523210999","Bimo",68.55,&S);
 push("4523210555","Arif",78.85,&S);
 push("4523210444","Rido",98.65,&S);
CetakStack ; Pop(&S) ; CetakStack(S) ; Pop(&S) ; Cetakstack(S)
print (0)

Algoritma :

1. Membuat fungsi buatSTKosong (stack *S)
2. (*S).top = NULL
3. Membuat fungsi isKosong (stack S)
4. hasil = false
5. Jika (S.top==NULL), maka kerjakan baris 6 s.d 7
6. Hasil = true
7. Memanggil fungsi hasil
8. Membuat fungsi jmlelemen (stack S)
9. hasil = 0
10. Jika (S.top = NULL), maka kerjakan baris 16 s.d 18
11. elemen *bantu
12. bantu = S.top
13. Jika(bantu != NULL) maka kerjakan baris 11 s.d 12
14. hasil = hasil +1;
15. bantu = bantu -> next
16. Memanggil fungsi hasil
17. Membuat fungsi Push (npm, nama_mhs, nilai, stack *S)
18. elemen *info
19. info = new elemen
20. info -> kontainer.npm = npm
21. info -> kontainer.nama_mhs = nama_mhs
22. info -> kontainer.nilai = nilai
23. Jika ((*S).top == NULL), maka kerjakan baris 15 kalau tidak baris 16
24. Menampilkan/mencetak "Stack Penuh"
25. info -> next = NULL
26. Membuat fungsi pop(stack *S)
27. Jika ((*S).top == NULL), maka kerjakan baris 27 s.d. Kalau tidak baris 28
28. elemen *hapus =(*S).top
29. Jika (jmlelemn(S)==1, maka kerjakan basis 29
30. (*S).top=NULL
31. (*S).top =(*S).top -> next;
32. hapus -> next = NULL
33. delete hapus
34. Menampilkan/Mencetak "Stack Kosong"

35. Membuat fungsi CetakStack(stack S)
36. Jika(S.top !=NULL)
37. Menampilkan/mencetak "Menampilkan Stock"
38. i = 1
39. Selama (bantu !=NULL), maka kerjakan bors 35 s.d 39 Kalau tidak baris 40
40. Menampilkan/mencetak nilai variabel i
41. Menampilkan/mencetak nilai variabel bantu->kontainer.npm
42. Menampilkan/mencetak nilai variabel bantu->kontainer.nama_mhs
43. Menampilkan/mencetak nilai variabel bantu->kontainer.nilai
44. bantu=bantu->next
45. i=i+1
46. Menampilkan/mencetak ("Stack Kosong")
47. Membuat fungsi utama
48. Membuat type struktur (npm, nama mhs, nilai)
49. Membuat objek NilaiMk dari type struktur
50. Membuat type struktur(kontainer, next)
51. Membuat objek elemen dari type struktur
43. Membuat Type struktur (top)
44. Membuat objek stack dari objek NilaiMK
45. Membuat objek stack dari type strukfur
46. objek S dari objek stack
47. Memanggil fungsi CetakStack (S)
48. Memasukkan isi elemen stack("4523210666","Budi",88.75,&S);
49. Memasukkan isi elemen stack("4523210777","Susi",78.85,&S);
50. Memasukkan isi elemen stack("4523210888","Nuri",98.65,&S);
51. Memasukkan isi elemen stack("4523210999","Bimo",68.55,&S);
52. Memasukkan isi elemen stack("4523210555","Arif",78.85,&S);
53. Memasukkan isi elemen stack("4523210444","Rido",98.65,&S);
54. Memasukkan isi elemen stack("4523210222","Ella",68.55,&S);
- 55.. Memanggil fungsi CetakStack (S)
56. Memanggil fungsi Pop (&S)
57. Memanggil fungsi CetakStack (S)
58. Memanggil fungsi Pop (&S)
59. Memanggil fungsi CetakStack (S)
60. Memanggil fungsi Pop (&S)
61. Selesai