

Contoh Praktikum Algoritma dan Struktur Data



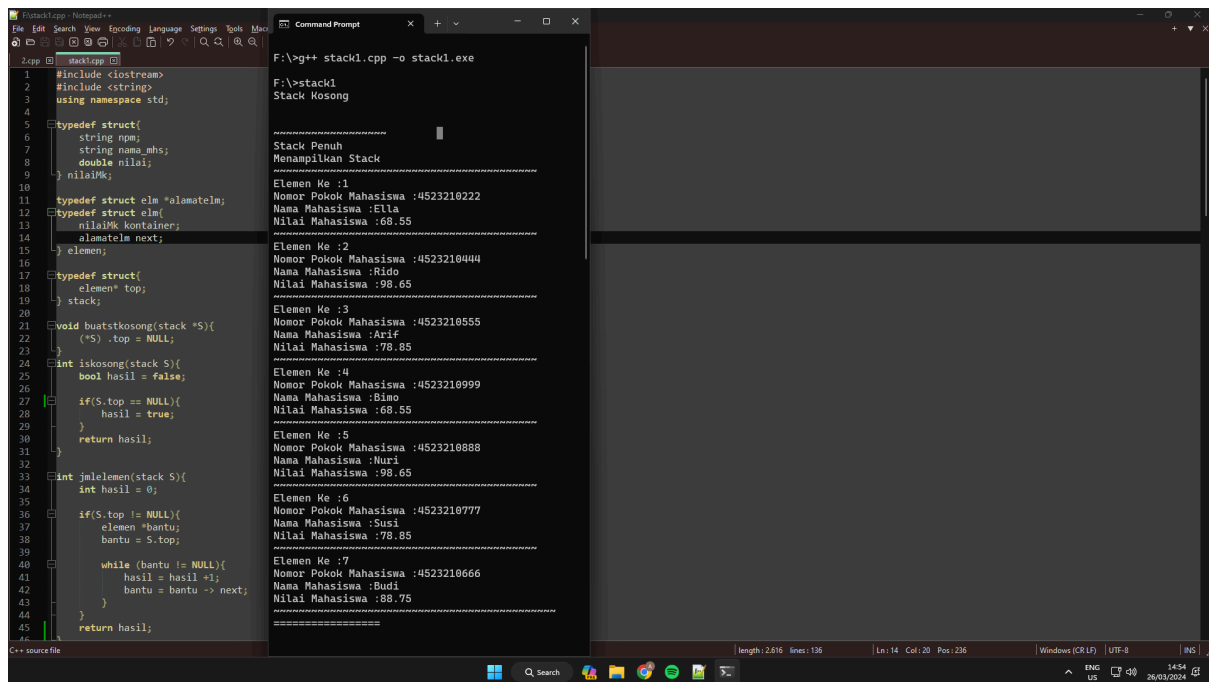
Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen Pengajar :

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

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

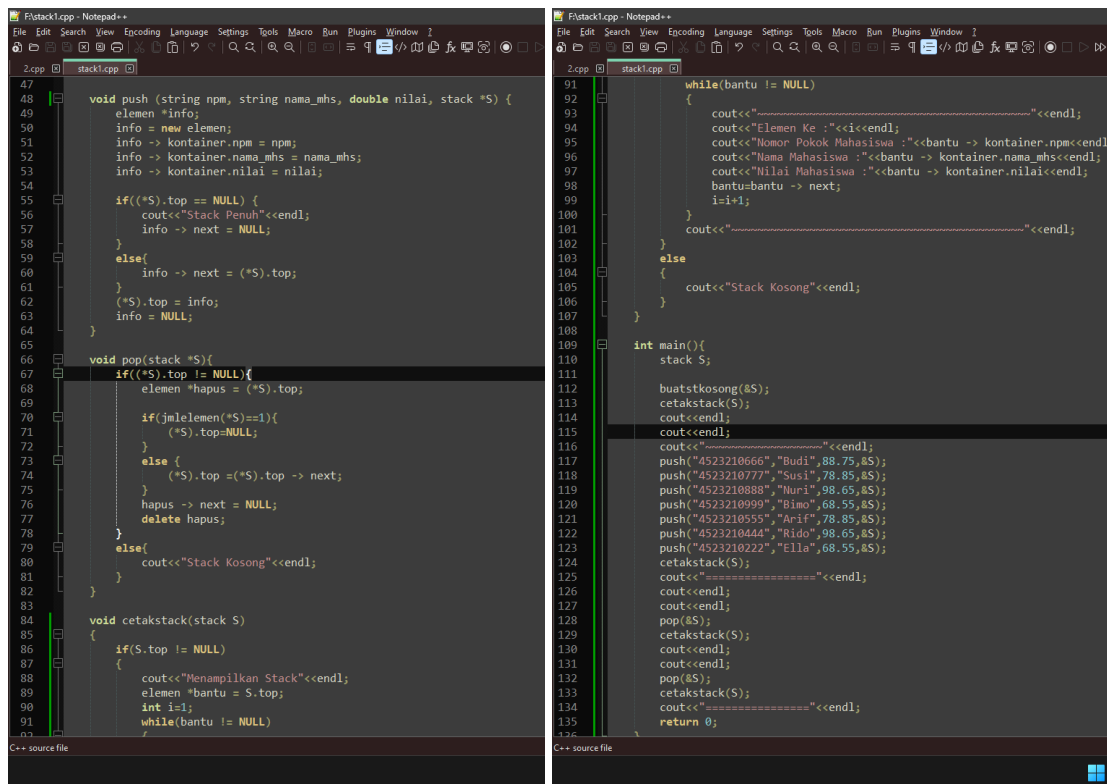
Contoh 1



```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 typedef struct{
6     string npm;
7     string nama_mhs;
8     double nilai;
9     struct tms *next;
10 } tms;
11
12 typedef struct elm {
13     tms *container;
14     struct elm *next;
15 } elm;
16
17 typedef struct{
18     elm *top;
19 } stack;
20
21 void buatstkosong(stack *S){
22     (*S).top = NULL;
23 }
24
25 int iskosong(stack S){
26     bool hasil = false;
27     if(S.top == NULL){
28         hasil = true;
29     }
30     return hasil;
31 }
32
33 int inlelemen(stack S){
34     int hasil = 0;
35     if(S.top != NULL){
36         elm *bantu;
37         bantu = S.top;
38         while (bantu != NULL){
39             hasil = hasil + 1;
40             bantu = bantu -> next;
41         }
42     }
43     return hasil;
44 }
```

```
F:\>g++ stack1.cpp -o stack1.exe
F:\>stack1
Stack Kosong

Stack Penuh
Menampilkan Stack
Elemen Ke :1
Nomor Pokok Mahasiswa :4523210222
Nama Mahasiswa :Ella
Nilai Mahasiswa :68.55
Elemen Ke :2
Nomor Pokok Mahasiswa :4523210444
Nama Mahasiswa :Rido
Nilai Mahasiswa :98.65
Elemen Ke :3
Nomor Pokok Mahasiswa :4523210555
Nama Mahasiswa :Arif
Nilai Mahasiswa :78.85
Elemen Ke :4
Nomor Pokok Mahasiswa :4523210999
Nama Mahasiswa :Bimo
Nilai Mahasiswa :68.55
Elemen Ke :5
Nomor Pokok Mahasiswa :4523210888
Nama Mahasiswa :Nuri
Nilai Mahasiswa :98.65
Elemen Ke :6
Nomor Pokok Mahasiswa :4523210777
Nama Mahasiswa :Susi
Nilai Mahasiswa :78.85
Elemen Ke :7
Nomor Pokok Mahasiswa :4523210666
Nama Mahasiswa :Budi
Nilai Mahasiswa :88.75
=====
```



```
47 void push (string npm, string nama_mhs, double nilai, stack *S) {
48     struct tms *info;
49     info = new tms;
50     info -> container.npm = npm;
51     info -> container.nama_mhs = nama_mhs;
52     info -> container.nilai = nilai;
53     if((*S).top == NULL) {
54         cout<<"Stack Penuh"<<endl;
55         info -> next = NULL;
56     }
57     else{
58         info -> next = (*S).top;
59     }
60     (*S).top = info;
61     info = NULL;
62 }
63
64 void pop(stack *S){
65     if((*S).top != NULL){
66         struct tms *hapus = (*S).top;
67         if(jmllelemen(*S)==1){
68             (*S).top=NULL;
69         }
70         else {
71             (*S).top = (*S).top -> next;
72             hapus -> next = NULL;
73             delete hapus;
74         }
75     }
76     cout<<"Stack Kosong"<<endl;
77 }
78
79 void cetakstack(stack S)
80 {
81     if(S.top != NULL)
82     {
83         cout<<"Menampilkan Stack"<<endl;
84         struct tms *bantu = S.top;
85         int i=1;
86         while(bantu != NULL)
87         {
88             cout<<"Elemen Ke : "<<i<<endl;
89             cout<<"Nomor Pokok Mahasiswa : "<<bantu -> container.npm<<endl;
90             cout<<"Nama Mahasiswa : "<<bantu -> container.nama_mhs<<endl;
91             cout<<"Nilai Mahasiswa : "<<bantu -> container.nilai<<endl;
92             bantu=bantu -> next;
93             i=i+1;
94         }
95         cout<<"Stack Kosong"<<endl;
96     }
97 }
98
99 int main(){
100     stack S;
101     buatstkosong(&S);
102     cetakstack(S);
103     cout<<endl;
104     cout<<"===== "<<endl;
105     push("4523210666","Budi",88.75,&S);
106     push("4523210777","Susi",78.85,&S);
107     push("4523210888","Nuri",98.65,&S);
108     push("4523210999","Bimo",68.55,&S);
109     push("4523210555","Arif",78.85,&S);
110     push("4523210444","Rido",98.65,&S);
111     push("4523210222","Ella",68.55,&S);
112     cetakstack(S);
113     cout<<"===== "<<endl;
114     cout<<endl;
115     cout<<endl;
116     pop(&S);
117     cetakstack(S);
118     cout<<endl;
119     pop(&S);
120     cetakstack(S);
121     cout<<"===== "<<endl;
122     return 0;
123 }
```

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[i]})
 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. Kalo 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 Kalo 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 struktur
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","Ari",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

Contoh2

```

1 #include <stdio.h>
2 #include <iostream>
3 #include <stdlib.h>
4 #include <string.h>
5 using namespace std;
6
7 void buat_simpul(int X);
8 void awal();
9 void cetak();
10 void insertkanan();
11 struct node{
12     int info;
13     struct node *link;
14 }typedef struct node simpul;
15 simpul *P, *first, *last, *Q;
16 int X;
17
18 int main(){
19     int A[7] = {22,28,7,11,66,63,10};
20     int i;
21     first = NULL;
22     i=0;
23     X=A[i];
24     buat_simpul(X);
25     awal();
26     for(i=1;i<=6;i++)
27     {
28         X=A[i];
29         buat_simpul(X);
30         insertkanan();
31         cout<<"Nilai Sebelum Dilakukan Insert Kanan : ";
32         cetak();
33         X=100;
34         buat_simpul(X);
35         insertkanan();
36         cout<<endl;
37         cout<<"Nilai Setelah Dilakukan Insert Kanan : ";
38         cetak();
39         cout<<endl;
40         ~cin.get();
41     }
42 void buat_simpul(int X){
43     P=(simpul*) malloc(sizeof(simpul));
44     if(P!=NULL){
45         P->info=X;
46         ~
47     }
48     ~
49     ~
50     ~
51 void awal(){
52     first=P;
53     last=P;
54     P->link=NULL;
55 }
56 void insertkanan(){
57     last->link=P;
58     last=P;
59     P->link=NULL;
60 }
61 void cetak(){
62     int X;
63     Q=first;
64     while(Q!=NULL){
65         X=Q->info;
66         cout<<" "<<X;
67         Q=Q->link;
68     }
69 }

```

```

F:\>g++ cnthtprak3-2.cpp -o 1
F:\>1
Nilai Sebelum Dilakukan Insert Kanan : 22 28 7 11 66 63 10
Nilai Setelah Dilakukan Insert Kanan : 22 28 7 11 66 63 10 100

```

Pseudocode

Kamus/Deklarasi Variabel
A[7], i = int

Algoritma/Deskripsi
A[7] = {22,28,7,11,66,63,10}
first = NULL
i = 0
X=A[i];
buat_simpul(X);
awal();
for(i=1;i<=6;i++)
 X=A[i]
 buat_simpul(X)
 insertkanan()
endfor
print ("cetak()")
X = 100
buat_simpul(X)
insertkanan()
print ("Cetak()")

Kamus/Deklarasi Variabel fungsi buat_simpul
X = int

Algoritma/Deskripsi Fungsi buatsimpul(X)
P=(simpul*) malloc(sizeof(simpul))
if(P!=NULL)
 P -> info =X
else
 print ("Pembuatan Simpul Tidak Berhasil")
exit(1)

Kamus/Deklarasi Variabel fungsi awal
-

Algoritma/Deskripsi Fungsi buatsimpul(X)
 first=P
 last=P
 P -> link=NULL

Kamus/Deklarasi Variabel Fungsi insertkanan
-

Algoritma/Deskripsi Fungsi insertkanan
 last -> link = P
 last=P
 P->link=NULL

Kamus/Deklarasi Variabel Cetak
X=int

Algoritma/Deskripsi Fungsi Cetak
Q=first;
 while(Q!=NULL)
 X=Q->info
 print ("X")
Q=Q->link

Algoritma :

1. Membuat Fungsi Utama
2. $A[7] = \{22, 28, 7, 11, 66, 63, 10\}$
3. $I = 0$
4. $X = A[I]$
5. Memanggil fungsi buat_simpul(X)
6. Memanggil fungsi awal
7. Selama ($I=1$), maka kerjakan baris 10 s.d 13
8. $X = A[I]$
9. Memanggil fungsi buat_simpul(X)
10. Memanggil fungsi insertkanan
11. Mencetak/Menampilkan Nilai Variabel Dari Fungsi cetak
12. $X=100$
13. Memanggil fungsi buat_simpul(X)
14. Memanggil fungsi insertkanan
15. Mencetak/Menampilkan Nilai Variabel Dari Fungsi cetak
16. Membuat fungsi buat_simpul(X)
17. $P = (\text{simpul}^*)$
18. Jika ($P \neq \text{NULL}$) maka kerjakan baris 26 s.d 28 kalau tidak baris 29
19. $P \rightarrow \text{info} = X$
20. Mencetak/Menampilkan "Pembuatan Simpul Tidak Berhasil"
21. $\text{exit}(1)$
22. Membuat fungsi awal
23. $\text{first} = P$
24. $\text{last} = P$
25. $P \rightarrow \text{link} = \text{NULL}$
26. Membuat fungsi insertkanan
27. $\text{last} \rightarrow \text{link} = P$
28. $\text{last} = P$
29. $P \rightarrow \text{link} = \text{NULL}$
30. Membuat fungsi cetak
31. $Q = \text{first}$
32. Selama ($Q \neq \text{NULL}$) maka kerjakan baris 46 s.d 48
33. $X = Q \rightarrow \text{info}$
34. Mencetak/Menampilkan Nilai Variabel X
35. $Q = Q \rightarrow \text{Link}$