

Tugas Praktikum Algoritma dan Struktur Data



Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen:

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

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

psad2-03

The screenshot shows a C++ program in Visual Studio and its execution in a Command Prompt. The program defines a struct for a student and a main function that takes input and prints the student's information and class grade based on their average score (ipk).

```

1 #include <iostream>
2 using namespace std;
3
4 struct mahasiswa {
5     public:
6         char npm[9];
7         char nama[30];
8         float ipk;
9 };
10
11 int main() {
12     mahasiswa mhs;
13     cout << "Nomor Pokok Mahasiswa = "; cin >> mhs.npm;
14     cout << "Nama Mahasiswa = "; cin >> mhs.nama;
15     cout << "Induk Prestasi Kumulatif = "; cin >> mhs.ipk;
16     cout << endl;
17
18     string kelas_nilai;
19     if (mhs.ipk >= 3.5) {
20         kelas_nilai = "A (Sangat Baik)";
21     }
22     else if (mhs.ipk >= 3.0) {
23         kelas_nilai = "B (Baik)";
24     }
25     else if (mhs.ipk >= 2.5) {
26         kelas_nilai = "C (Cukup)";
27     }
28     else if (mhs.ipk >= 2.0) {
29         kelas_nilai = "D (Kurang)";
30     }
31     else {
32         kelas_nilai = "E (Gagal)";
33     }
34
35     cout << endl;
36     cout << "-----" << endl;
37     cout << "Data Anda " << endl;
38     cout << "-----" << endl;
39     cout << "Nomor Pokok Mahasiswa = " << mhs.npm << endl;
40     cout << "Nama Mahasiswa = " << mhs.nama << endl;
41     cout << "Induk Prestasi Kumulatif = " << mhs.ipk << " (" << kelas_nilai << ")" << endl;
42     cin.get();
43 }

```

The Command Prompt shows the execution of the program with the following output:

```

F:\>g++ psad2-03.cpp -o psad2-03
F:\>psad2-03
Nomor Pokok Mahasiswa = 125
Nama Mahasiswa = Agil
Induk Prestasi Kumulatif = 3.5

-----
Data Anda
-----
Nomor Pokok Mahasiswa = 125
Nama Mahasiswa = Agil
Induk Prestasi Kumulatif = 3.5 (A (Sangat Baik))
F:\>

```

Pseudocode

Kamus/Deklarasi Variabel

npm[9] : char
nama[30] : char
ipk : float
kelas_nilai : string

Algoritma/Deskripsi

```

mahasiswa mhs
input (npm)
input (nama)
input (ipk)
if (mhs.ipk >= 3.5)
    kelas_nilai = "A (Sangat Baik)"
else
if (mhs.ipk >= 3.0)
    kelas_nilai = "B (Baik)"
else
if (mhs.ipk >= 2.5)
    kelas_nilai = "C (Cukup)"
else
if (mhs.ipk >= 2.0)
    kelas_nilai = "D (Kurang)"
else
    kelas_nilai = "E (Gagal)"
print (mhs.npm)
print (mhs.nama)
print (mhs.ipk, kelas_nilai)

```

Algoritma

1. Deklarasi class (class mahasiswa {npm[9], nama[30],ipk})
2. Mendefinisikan objek (mahasiswa mhs)
3. Masukkan anggota struktur (mhs.npm)
4. Masukkan anggota struktur (mhs.nama)
5. Masukkan anggota struktur (mhs.ipk)
6. Jika (mhs.ipk >= 3.5), kerjakan baris 8
7. kelas_nilai = A (Sangat Baik)
8. Jika (mhs.ipk >= 3.0), kerjakan baris 12
9. kelas_nilai = B (Baik)
10. Jika (mhs.ipk >= 2.5), kerjakan baris 16
11. kelas_nilai = C (Kurang)
12. Jika (mhs.ipk >= 2.0), kerjakan baris 20
13. kelas_nilai = D (Kurang)
14. Mencetak/Menampilkan nilai mhs.npm
15. Mencetak/Menampilkan nilai mhs.nama
16. Mencetak/Menampilkan nilai mhs.ipk
17. selesai

psad2-10

```
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 } NilaiMK;
10
11 typedef struct {
12     int top;
13     NilaiMK Dat[10];
14 } stack;
15
16 void buatSTkosong(stack *S) {
17     (*S).top = -1;
18 }
19
20 bool isKosong(stack S) {
21     return (S.top == -1);
22 }
23
24 bool isPenuh(stack S) {
25     return (S.top == 9);
26 }
27
28 void push(stack *S) {
29     if (isPenuh(*S)) {
30         cout << "Stack Penuh" << endl;
31         return;
32     }
33
34     string npm, nama_mhs;
35     double nilai;
36
37     cout << "Masukkan Nomor Pokok Mahasiswa: ";
38     cin >> npm;
39     cout << "Masukkan Nama Mahasiswa: ";
40     cin.ignore();
41     getline(cin, nama_mhs);
42     cout << "Masukkan Nilai Mahasiswa: ";
43     cin >> nilai;
44
45     if (isKosong(*S)) {
46         (*S).top = 0;
47     } else {
48         (*S).top++;
49     }
50
51     (*S).Dat[(*S).top].npm = npm;
52     (*S).Dat[(*S).top].nama_mhs = nama_mhs;
53     (*S).Dat[(*S).top].nilai = nilai;
54 }
55
56 void pop(stack *S) {
57     if (isKosong(*S)) {
58         cout << "Stack Kosong" << endl;
59         return;
60     }
61
62     (*S).top--;
63 }
64
65 void cetakstack(stack S) {
66     int i;
67     if (isKosong(S)) {
68         cout << "Stack Kosong" << endl;
69         return;
70     }
71
72     cout << "Menampilkan Stack" << endl;
73     for (i = S.top; i >= 0; i--) {
74         cout << "-----" << endl;
75         cout << endl;
76         cout << "Elemen Ke : " << i << endl;
77         cout << "Nomor Pokok Mahasiswa : " << S.Dat[i].npm << endl;
78         cout << "Nama Mahasiswa : " << S.Dat[i].nama_mhs << endl;
79         cout << "Nilai Mahasiswa : " << S.Dat[i].nilai << endl;
80         cout << "-----" << endl;
81     }
82 }
83
84 int main() {
85     stack S;
86
87     buatSTkosong(&S);
88     cetakstack(S);
89     cout << endl;
90     cout << endl;
91     cout << "-----" << endl;
92     push(&S);
93     push(&S);
94     push(&S);
95     push(&S);
96     cetakstack(S);
97     cout << "-----" << endl;
98     cout << endl;
99     cout << endl;
100    pop(&S);
101    cetakstack(S);
102    cout << endl;
103    cout << endl;
104    pop(&S);
105    cetakstack(S);
106    cout << "-----" << endl;
107    return 0;
108 }
```

```
83
84 int main() {
85     stack S;
86
87     buatSTkosong(&S);
88     cetakstack(S);
89     cout << endl;
90     cout << endl;
91     cout << "-----" << endl;
92     push(&S);
93     push(&S);
94     push(&S);
95     push(&S);
96     cetakstack(S);
97     cout << "-----" << endl;
98     cout << endl;
99     cout << endl;
100    pop(&S);
101    cetakstack(S);
102    cout << endl;
103    cout << endl;
104    pop(&S);
105    cetakstack(S);
106    cout << "-----" << endl;
107    return 0;
108 }
```

```
Command Prompt

F:\>1
Stack Kosong

Masukkan Nomor Pokok Mahasiswa: 125
Masukkan Nama Mahasiswa: Agil
Masukkan Nilai Mahasiswa: 3.00
Masukkan Nomor Pokok Mahasiswa: 126
Masukkan Nama Mahasiswa: Deri
Masukkan Nilai Mahasiswa: 3.50
Masukkan Nomor Pokok Mahasiswa: 127
Masukkan Nama Mahasiswa: Syah
Masukkan Nilai Mahasiswa: 3.75
Masukkan Nomor Pokok Mahasiswa: 128
Masukkan Nama Mahasiswa: Hasan
Masukkan Nilai Mahasiswa: 4.00
Menampilkan Stack

Elemen Ke : 3
Nomor Pokok Mahasiswa :128
Nama Mahasiswa :Hasan
Nilai Mahasiswa :4

Elemen Ke : 2
Nomor Pokok Mahasiswa :127
Nama Mahasiswa :Syah
Nilai Mahasiswa :3.75

Elemen Ke : 1
Nomor Pokok Mahasiswa :126
Nama Mahasiswa :Deri
Nilai Mahasiswa :3.5

Elemen Ke : 0
Nomor Pokok Mahasiswa :125
Nama Mahasiswa :Agil
Nilai Mahasiswa :3

=====

Menampilkan Stack
```

```
Command Prompt

Menampilkan Stack

Elemen Ke : 2
Nomor Pokok Mahasiswa :127
Nama Mahasiswa :Syah
Nilai Mahasiswa :3.75

Elemen Ke : 1
Nomor Pokok Mahasiswa :126
Nama Mahasiswa :Deri
Nilai Mahasiswa :3.5

Elemen Ke : 0
Nomor Pokok Mahasiswa :125
Nama Mahasiswa :Agil
Nilai Mahasiswa :3

Menampilkan Stack

Elemen Ke : 1
Nomor Pokok Mahasiswa :126
Nama Mahasiswa :Deri
Nilai Mahasiswa :3.5

Elemen Ke : 0
Nomor Pokok Mahasiswa :125
Nama Mahasiswa :Agil
Nilai Mahasiswa :3

=====

F:\>
```

Pseudocode

Kamus/Deklarasi Variabel fungsi buatSTKosong
-

Algoritma/Dekripsi fungsi buatSTKosong
(*S).top = -1

Kamus/Deklarasi Variabel fungsi isKosong
hasil = boolean

Algoritma/Deskripsi fungsi isKosong(stack S)
if(S.top == -1)
 hasil = true
endif
print(hasil)

Kamus/Deklarasi Variabel fungsi isPenuh
hasil = boolean

Algoritma/Deskripsi fungsi isPenuh(stack S)
if(S.top == 9)
 hasil = true
endif
print(hasil)

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

Algoritma/Deskripsi fungsi push stack(*S)
push(npm,nama_mhs,nilai,stack*S)
if(isPenuh(*S))
 print("Stack Penuh")
return
input (npm)
input (nama_mhs)
input (nilai

```
if(isKosong(*S))
    (*S).top = 0
else
    (*S).top++
endif
(*S).Dat[(*)S).top].npm=npm
(*S).Dat[(*)S).top].nama_mhs=nama_mhs
(*S).Dat[(*)S).top].nilai=nilai
endif
```

Kamus/Deklarasi Variabel fungsi pop
-

Algoritma/Dekripsi fungsi pop(stack *S)
if((isKosong (*S))
 print ("stack kosong")
endif
(*S).top--

Kamus/Deklarasi Variabel fungsi cetakstack
i : int

Algoritma/Dekripsi fungsi cetakstack(stack *S)
if(isKosong(S))
 print ("stack kosong")
for(i=S.top;i>=0;i--)
 print(i, S.Dat[i].npm, S.Dat[i].nama_mhs, S.Dat[i].nilai)
endfor
endif

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 (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","Sari",98.65,&S);
```

```
push("4523210999","Bimo",68.55,&S);
```

```
CetakStack ; Pop(&S) ; CetakStack(S) ; Pop(&S) ; Cetak
```

Algoritma :

```
1.  Membuat fungsi buatSTKosong (stack *S)
2.  (*S).top = -1
3.  Membuat fungsi isKosong (stack S)
4.  ( S.top = -1 )
5.  Membuat fungsi isPenuh (stack S)
6.  Jika (S.top = 9)
7.  Membuat fungsi Push (stack *S)
8.  Jika (isPenuh (*S) ), maka kerjakan baris 14 kalau tidak
    baris 15
9.  Menampilkan/mencetak "Stack Penuh"
10. Memasukkan anggota karakter npm
11. Memasukkan anggota karakter nama_mhs
12. Memasukkan anggota karakter nilai
13. Jika (isKosong(*S)) maka kerjakan baris 29 s.d 31
14. (*S).top = 0
15. (*S).top++
16. (*S).Dat[(S).top].npm = npm
17. (*S).Dat[(S).top].nama_mhs = nama_mhs
18. (*S).Dat[(S).top].nilai = nilai
19. Membuat fungsi pop(stack *S)
20. Jika ((S).top ), maka kerjakan baris 41 s.d. 42 Kalo tidak
    baris 43
21. Menampilkan/mencetak "Stack Kosong"
22. (*S).top--
23. Jika (S) top-1, maka kerjakan basis 29
24. (*S).top = (*S) top = -1
25. Membuat fungsi CetakStack(stack S)
26. Jika (isKosong(S)) maka kerjakan baris 51 s.d 52
27. Menampilkan/Mencetak "Stack Kosong"
28. Menampilkan/Mencetak "Menampilkan Stack"
29. Selama (i=S.top) maka kerjakan baris 57 s.d 64
30. Menampilkan/mencetak (i)
31. Menampilkan/mencetak (SDat[i].npm
32. Menampilkan/mencetak (S.Dat[i].)nama_mhs)
33. Menampilkan/mencetak (SDat[i].nilai
```

```
34. Memanggil fungsi main
35. Memanggil fungsi StKosong (&S)
36. Memanggil fungsi cetakstack (S)
37. Memasukkan isi elemen stack (&S),
38. Memasukkan isi elemen stack (&S)
39. Memasukkan isi elemen stack (&S),
40. Memasukkan isi elemen stack (&S),
41. Memanggil fungsi CetakStack (S)
42. Memanggil fungsi Pop (&S)
43. Memanggil fungsi CetakStack (S)
44. Memanggil fungsi Pop (&S)
45. Memanggil fungsi CetakStack (S)
46. Memanggil fungsi Pop (&S)
47. Mencetak (0)
48. Selesai
```