

Lampiran 1

Contoh Soal dan Pembahasan

Ujian Teori Pemrograman Pascal

58 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

SOAL-SOAL DEKLARASI

1. Manakah yang mendeklarasikan tipe enumerasi dengan tepat?

- a. Type a=integer;
- b. Type a=1..300;
- c. Type a=(baik, jelek, buruk);
- d. Type a=[baik, jelek, buruk];
- e. Type a=baik, jelek, buruk;

Jawab:

- c. Type a=(baik, jelek, buruk);

2. Tipe di bawah ini mana yang tidak dapat melakukan operasi aritmatika?

- a. integer
- b. byte
- c. real
- d. boolean
- e. word

Jawab:

- d. boolean

3. Deklarasi prosedur manakah yang dibenarkan?

- a. procedure hapus;
- b. procedure hapus(s:string);
- c. procedure hapus(var s:string);
- d. procedure hapus(s:string):boolean;
- e. procedure hapus(var data);

Jawab:

- d. procedure hapus(s:string):boolean;

Pembahasan:

Untuk penulisan prosedur, tidak diperbolehkan adanya nilai kembali. Sedangkan parameter tanpa tipe

data (pada opsi e), dapat dibenarkan.

Konsep Dasar Pemrograman Prosedural 59

Tim Olimpiade Komputer Indonesia

4. Deklarasi function manakah yang tidak diizinkan?

- a. Function density(x:real):real;
- b. Function density(b:byte):byte;
- c. Function density(var s:string):real;
- d. Function density(var data):byte;
- e. Function density;

Jawab:

- e. Function density;

Pembahasan:

Untuk penulisan function, harus ada nilai kembali.

5. Tipe variabel ekspresi manakah yang tidak dapat ditampilkan dengan procedure Writeln?

- a. Type T=Integer;
- b. Type T=String;
- c. Type C=Char;
- d. Type T=(Small, Medium, Large)

e. Semua valid

Jawab:

d. Type T=(Small, Medium, Large)

Pembahasan:

Opsi d adalah tipe data enumerasi. Tipe data enumerasi tidak dapat ditampilkan dengan perintah Writeln.

6. Dengan deklarasi berikut:

```
Type warna=(merah,kuning,hijau,biru,hitam,putih,jingga);
```

```
Var w:warna;
```

Perintah mana yang salah?

a. If w in [warna] then writeln('ada');

b. w:=merah;

w:=w + kuning;

c. w:=[merah];

d. w:=hijau;

dec(w);

e. w:='Merah';

60 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Jawab:

d. w:=hijau;

dec(w);

Pembahasan:

Variabel w adalah variabel yang mempunyai tipe data enumerasi yang merupakan salah satu tipe data

ordinal. Karena merupakan tipe data ordinal, maka variabel w dapat dioperasikan dengan fungsi atau

prosedur seperti ORD, DEC, INC, PRED, dan SUCC.

7. Pada deklarasi di atas, jika variabel W1 berisi [merah,kuning,hijau] dan variabel w2 berisi [merah,kuning,hitam] maka, jika diberikan statemen $W3 := W1 + W2$, W3 akan berisi:

a. [merah,kuning,hijau,hitam]

b. [merah,kuning,hijau,merah,kuning,hitam]

c. [hijau,hitam]

d. [merah,kuning,merah,kuning,hijau, hitam]

e. [merah,kuning]

Jawab:

a. [merah,kuning,hijau,hitam]

Pembahasan:

Operator + pada tipe data himpunan adalah gabungan atau union.

8. Jika diberikan statemen $W3 := W1 - W2$, W3 akan berisi:

a. [merah,kuning,hijau,hitam]

b. [merah,kuning,hijau,merah,kuning,hitam]

c. [hijau]

d. [merah,kuning,merah,kuning,hijau, hitam]

e. [merah,kuning]

Jawab:

c. [hijau]

Pembahasan:

Operator - pada tipe data himpunan adalah operator *difference*.

Konsep Dasar Pemrograman Prosedural 61

Tim Olimpiade Komputer Indonesia

9. Jika diberikan statemen $W3 := W1 * W2$, W3 akan berisi:

a. [merah,kuning,hijau,hitam]

b. [merah,kuning,hijau,merah,kuning,hitam]

c. [hijau,hitam]

d. [merah,kuning,merah,kuning,hijau, hitam]

e. [merah,kuning]

Jawab:

c. [hijau,hitam]

Pembahasan:

Operator * pada tipe data himpunan adalah operator irisan.

SOAL-SOAL INPUT / OUTPUT

10. Perhatikan potongan program berikut ini :

```
Begin  
writeln((10 shr 1) shl 2);  
end.
```

Apa yang dihasilkan oleh program diatas...

a. 18

b. 19

c. 20

d. 21

e. 22

Jawab:

c. 20

Pembahasan:

Operator SHR adalah operasi pergeseran bit ke kanan dan operasi shl adalah operasi pergeseran bit ke kiri.

$10 \text{ shr } 1 = 5$ ($1010 \text{ shr } 1 = 0101 = 5$)

$5 \text{ shl } 2 = 20$ ($0101 \text{ shl } 2 = 10100 = 20$)

62 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

SOAL-SOAL STRUKTUR KONTROL

11. Bagaimana keluaran program di bawah ini?

```
Var  
I:integer;  
Begin  
I:=2;  
Case I of  
1,3,5,7,9:writeln('Ganjil');  
2:writeln('Prima genap');  
0..10:writeln('Normal');  
else writeln('Tidak normal');  
end;  
end;
```

a. Prima genap

b. Normal

c. Prima genap

Normal

d. Normal

Prima genap

e. Prima genap

Tidak normal

Jawab:

a. Prima genap

Pembahasan:

Struktur kendali case akan segera keluar untuk menjalankan statement berikutnya setelah menemukan

nilai yang tepat.

Perhatikan program di bawah ini:

```
var I,j,k:integer;  
L:byte;  
begin  
i:=3;  
j:=4;  
k:=32;  
L:=0;
```

```
{If - 1 }
if i + j and k = 0 then
writeln('Betul')
else
```

Konsep Dasar Pemrograman Prosedural 63

Tim Olimpiade Komputer Indonesia

```
writeln('Salah');
{If - 2 }
if (i = 2) and (j < i) or (k > i) then
writeln('Betul')
else
writeln('Salah');
{If - 3}
if not L in [1..120] then
writeln('Betul')
else
writeln('Salah');
end.
```

Program diatas berisi tiga perintah if then else yang saling tidak berkaitan, masing-masing IF diberi

nama IF - 1, IF - 2, IF - 3.

12. Perintah if manakah yang tidak dibenarkan:

- If - 1
- If - 2
- If - 3
- If - 1 dan if - 2
- Tidak ada if yang salah

Jawab:

- Tidak ada if yang salah

Pembahasan:

Pada If - 1, ekspresi i+j and k adalah ekspresi matematika dengan urutan pengerjaan j and k kemudian

ditambahkan dengan i. Ini merupakan ekspresi yang valid dalam bahasa Pascal

Pada If - 3, ekspresi Not L akan dioperasikan terlebih dulu. Ini juga merupakan ekspresi yang valid

dalam bahasa Pascal.

13. Pada program di atas, if mana yang menghasilkan output "Betul"?

- If - 1
- If - 2
- If - 3
- If - 1 dan if - 2
- Tidak ada if yang menghasilkan "Betul"

Jawab:

64 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

b. If - 2

Pembahasan:

Urutan pengerjaan operator AND dan OR adalah AND akan dievaluasi terlebih dulu. Pada kondisi

pertama, (i = 2) and (j < i) akan menghasilkan nilai FALSE, namun pada saat dievaluasi dengan

menggunakan kondisi OR, yaitu (k>i), akan menghasilkan TRUE, sehingga yang dicetak adalah "Betul"

SOAL-SOAL PERULANGAN

14. Perhatikan penggalan program berikut ni :

```
const
Data: array [1..3,1..3] of char=
((('1','1','2'),('2','2','4'),('4','4','8')));
var i, j : byte;
begin
for i:= 1 to 3 do
```

```

begin
for j:=3 downto 1 do
write(Data[i,j]):
writeln;
end;
end.

```

Apa keluaran program di atas ?

- a. 112
224
448
- b. '1''1''2'
'2''2''4'
'4''4''8'
- c. 211
422
844
- d. '2''1''1'
'4''2''2'
'8''4''4'

Konsep Dasar Pemrograman Prosedural 65
Tim Olimpiade Komputer Indonesia

- e. 124
124
248

Jawab:

- c. 211
422
844

15. Perhatikan program dibawah ini :

```

type data=set of char;
var setchar:data;
s:string;
i:integer;
begin
setchar:=[];
readln(s);
for i:=1 to length(s) do
begin
if not(s[i] in setchar) then
begin
setchar:=setchar+[s[i]];
write(s[i]);
end;
end;
writeln;
end.

```

Output dari program di atas jika input 'To be or Not To be that is the question' adalah

- a. 'To berNthaisqun.'
- b. 'To berNhaisqu`
- c. 'to@bernhaisquN'
- d. 'T N.'
- e. 'OBERTHAISQUN'

Jawab:

- a. 'To berNthaisqun.'

Pembahasan:

Yang perlu diperhatikan adalah bahwa tidak ada anggota yang sama dalam sebuah set (himpunan).

66 Konsep Dasar Pemrograman Prosedural
Tim Olimpiade Komputer Indonesia

16. Gunakan program berikut untuk menjawab pertanyaan :

```

type data=set of byte;
var setint:data;
i:integer;

```

```

begin
setint:=1;
setint:=setint+3;
setint:=5;
for i:=1 to 5 do
begin
if (i in setint) then continue else setint:=i;
end;
end.

```

Output dari program di atas adalah:

- a. [1, 2, 3, 4, 5]
- b. [1, 3, 5]
- c. [5]
- d. [1, 3]
- e. []

Jawab:

- c. [5]

Pembahasan:

Statement di bawah ini

```

setint:=1;
setint:=setint+3;
setint:=5;

```

Akan membuat setint berisi [5] saja. Pada statement berikutnya:

```

for i:=1 to 5 do
begin
if (i in setint) then continue else setint:=i;
end;

```

Akan membuat setint berisi nilai terakhir dari i yaitu 5.

SOAL-SOAL PROSEDUR DAN FUNGSI

17. Perhatikan program berikut :

```
var s:string;
```

Konsep Dasar Pemrograman Prosedural 67

Tim Olimpiade Komputer Indonesia

```

begin
s:='TOKI GO GET GOLD!';
delete(s,1,length(s)-12);
writeln(s);
end.

```

Apa keluaran program di atas ?

- a. GO GET GOLD!
- b. GO GET GOLD!
- c. GET GOLD!
- d. TOKI GO GET
- e. TOKI GO GE

Jawab:

- a. GO GET GOLD!

Pembahasan:

Procedure delete:

Deklarasi : procedure Delete(var S: String; Index: Integer; Count:Integer);

Keterangan : procedure delete akan menghapus S sebanyak count karakter, dimulai dari posisi Index.

Function length:

Deklarasi : Function Length (S : String) : Integer;

Keterangan : Length menghasilkan panjang dari S, bernilai antara 0 sampai dengan 255. Jika S tidak berisi apa-apa maka akan menghasilkan 0.

Statement delete(s,1,length(s)-12) akan menghapus s dari posisi 1 sebanyak panjang s, yaitu 17-12 = 5.

Sehingga yang dihapus adalah karakter 'TOKI ' dan s akan bernilai GO GET GOLD!

18. Perhatikan penggalan program berikut :

```

var i,k: integer;
begin
i:=5; k:=0;

```

```
k:=trunc(sqrt(i))+1;  
writeln(k);  
end.
```

Apa keluaran program di atas ?

- a. 3
- b. 2.24

68 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

- c. 2

- d. 0

- e. program tidak dapat dijalankan

Jawab:

- a. 3

Pembahasan:

Fungsi sqrt :

Deklarasi : Function Sqrt (X : Real) : Real;

Keterangan : menghasilkan akar pangkat dua dari x, di mana x harus positif

Fungsi trunc:

Deklarasi : Function Trunc (X : Real) : Longint;

Keterangan : menghasilkan bilangan bulat dari X, akan selalu lebih kecil atau sama dengan X.

Sqrt(5) akan menghasilkan 2.23

Trunc(2.23) akan menghasilkan 2

Sehingga k:=trunc(sqrt(i))+1; akan menghasilkan 3

19. Mengacu pada program berikut :

```
var  
A,B:string;  
C:string[10];  
begin  
A:='TOKI MEMANG';  
B:='HEBAT';  
C:=A+B;  
if (Pos(B)>0) then  
begin  
writeln('A');  
end else  
writeln('B');  
end.
```

Apa yang terjadi jika program di atas di jalankan...

- a. Huruf 'A' tercetak
- b. Huruf 'B' tercetak
- c. Tidak dapat dipastikan
- d. Terjadi error
- e. Tidak bisa di compile

Konsep Dasar Pemrograman Prosedural 69

Tim Olimpiade Komputer Indonesia

Jawab:

- e. Tidak bisa di compile

Pembahasan:

Kesalahan pertama yang akan ditemui program adalah pada function pos.

Deklarasi:Function Pos (Substr : String; S : String) : Integer;

Keterangan : function pos akan menghasilkan urutan atau posisi substr di S. Jika tidak ditemukan,

maka akan menghasilkan 0.

Pada program function pos hanya terdiri dari 1 parameter saja sehingga program tidak akan dapat dijalankan.

20. Perhatikan potongan program berikut :

```
begin  
writeln(round(frac(3.7)));  
end.
```

Apa keluaran program di atas ?

a. 0 b. 1 c. 2 d. 3 e. 4

Jawab:

b. 1

Pembahasan:

Fungsi frac (lihat pembahasan di atas)

Fungsi round

Deklarasi: `Function Round (X : Real) : Longint;`

Keterangan : membulatkan bilangan X, yang mungkin lebih besar atau lebih kecil dari X.

Frac(3.7) akan menghasilkan 0.7

Round(0.7) akan menghasilkan 1

21. Diketahui deklarasi fungsi dan variabel sebagai berikut:

```
var St: String;  
procedure Sulap(var S: String);  
begin  
  if S = 'Kecil' then S := 'kecil' else
```

70 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

```
  if S = 'Besar' then S := 'BESAR';  
end;
```

Di antara potongan program berikut, manakah yang salah?

a. `St := Chr(60);`

`Sulap(St);`

b. `St := 'KECIL';`

`Sulap(St);`

c. `St := Chr(45) + Chr(65);`

`Sulap(St);`

d. `Sulap('Besar');`

e. Semua ekspresi di atas benar

Jawab:

a. `St := Chr(60);`

`Sulap(St);`

Pembahasan:

Sebuah variabel string tidak dapat diberikan nilai bertipe data character.

SOAL-SOAL OPERASI FILE

22. Perintah mana yang tidak boleh digunakan untuk file bertipe text?

a. Assign

b. Reset

c. EOF

d. FilePos

e. Semua boleh digunakan untuk Text

Jawab:

d. FilePos

Pembahasan:

Perintah FilePos adalah perintah untuk mengetahui posisi file pointer (penunjuk file), dan hanya dapat

dioperasikan untuk file bertipe bukan text.

Konsep Dasar Pemrograman Prosedural 71

Tim Olimpiade Komputer Indonesia

Gunakan program berikut ini untuk menjawab soal di bawah ini:

```
program Uji;  
var T: Text;  
i, j, k: integer;  
begin  
  Assign(T, 'INPUT.TXT');  
  Reset(T);  
  Readln(T, i, j, k);  
  writeln(i, ' ', j, ' ', k);
```



```

Readln(T,i);
Readln(T,j);
Writeln(i,' ',j);
Close(T);
End.

```

23. Misalkan file INPUT.TXT berisi baris-baris sebagai berikut:

```

3 1 4 9
5 2 6
8 7
0

```

Bagaimanakah output dari program tersebut?

a. 3 1 4 9
5 2 6
8 7

b. 3 1 4
9 5

c. 3 1 4
5 2

d. 3 1 4
5 8

e. Terjadi runtime error karena isi file INPUT.TXT tidak sesuai untuk program ini.

Jawab:

d. 3 1 4
5 8

72 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Pembahasan:

Perintah Readln akan melakukan pembacaan di baris berikutnya. Perintah Readln pertama akan melakukan pembacaan pada file baris pertama, perintah Readln berikutnya melakukan pembacaan

pada baris ke dua dan perintah Readln terakhir melakukan pembacaan pada baris ketiga.

SOAL-SOAL KASUS / MEMBACA PROGRAM

Program berikut ini dipakai untuk menjawab dua soal di bawah ini

```

var Bil:Integer;
procedure Find(B:Integer;I:Integer);
var J,R:Integer;
begin
R:=Round(sqrt(B));
J:=2;
while (J<=R) and (B Mod J<>0) do
inc(J);
if J<=R then
begin
write(J,'*');
Find(B div J, I+1);
end
else if I>0 then
writeln(B,'=',Bil)
else
writeln('Bilangan Prima!');
end;
begin
write('Masukkan bilangan : ');
Readln(Bil);
Find(Bil,0);
end.

```

24. Bagaimana output program di atas bila inputnya 42?

a. $7 * 3 * 2 = 42$

b. Bilangan prima

c. =42

d. $2 * 3 * 7 = 42$

e. Salah semaa

Jawab:

Konsep Dasar Pemrograman Prosedural 73

Tim Olimpiade Komputer Indonesia

a. $7 * 3 * 2 = 42$

25. Bagaimana output program di atas bila, inputnya 23?

a. = 23

b. Bilangan pima

c. $23 * 1 = 23$

d. = 23 Bilangan prima!

e. Salah semua

Jawab:

b. Bilangan prima

Joni, petugas statistik yang baru saja belajar Pascal. Mencoba membuat program perata-rata sebagai berikut

```
var Amatan:array[5] of integer;
Jumlah:Integer;
RataRata:Integer;
I:Integer;
begin
for I:=1 to 5 do
begin
Write('Amatan ke-',I,' : ');
Readln(Amatan[I]);
end;
Jumlah:=0;
For I:=1 to 5 do
begin
Jumlah:=Jumlah+Amatan[I];
RataRata:=Jumlah/5;
Writeln('Jumlah = ',Jumlah);
Writeln('Rata-rata = ',RataRata);
Readln;
end.
```

Gunakan program yang dibuat oleh Joni ini untuk menjawab soal-soal berikut.

26. Ketika si Joni mencoba menjalankan program tersebut, ternyata, compiler menunjukkan sebuah pesan

kesalahan yang membuat: ia kebingungan. Tahukah Anda kesalahan pertama yang dibuat Joni?

a. Judul program (program Statistik) terlalu panjang, maksimum 8 karakter (misalnya:

program

Stat)

74 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

b. Procedure Readln (pada baris terakhir program sebelum end.) tidak boleh dipanggil tanpa parameter. Jadi seharusnya: Readln(I);

c. Statement for dengan variabel sama tidak boleh diulangi dua kali. Seharusnya dideklarasikan variabel lain, misalnya var I: Integer untuk for yang kedua

d. Deklarasi array salah, semestinya: var Amatan: array[1..5] of Integer;

e. Nama variabel seperti RataRata tidak valid, seharusnya Ratarata

Jawab:

a. Deklarasi array salah, semestinya: var Amatan: array[1..5] of Integer;

Pembahasan:

Deklarasi dari array adalah:

type
identifier=array[tipe_indeks] of tipe_data
di mana tipe_indeks adalah tipe data ordinal.

27. Setelah Anda memberi saran demikian, ternyata Joni masih belum bisa meng-compile programnya. Apa

sebabnya?

a. setiap variabel harus dideklarasikan dengan keyword var sendiri-sendiri.

Misalnya:

```
var Jumlah: Integer;
var RataRata: Integer;
Var I: Integer;
```

b. Variabel RataRata tidak harus bertipe Real

c. Semua variabel, kecuali I seharusnya adalah Real, tidak boleh Integer

d. Pemisah antara parameter dalam Write dan Writeln harus titik koma, bukan koma, Misalnya

```
Writeln('Jumlah = ',Jumlah);
```

e. Semua alasan di atas salah

Jawab:

c. Semua variabel, kecuali I seharusnya adalah Real, tidak boleh Integer

Pembahasan:

Dalam program diberikan instruksi $RataRata := Jumlah / 5$ yang berarti variabel RataRata harus bertipe

Real. Karena operator / hanya dikenal oleh variabel yang bertipe real.

Konsep Dasar Pemrograman Prosedural 75

Tim Olimpiade Komputer Indonesia

28. Joni mengganti operator "/" dengan "div" pada baris ke-15 program tersebut. Apa akibatnya?

a. program tidak mau di-compile karena. operator div tidak dapat digunakan di situ

b. nilai rata-ratanya menjadi 5

c. nilai rata-ratanya menjadi 6

d. nilai rata-ratanya menjadi 0

e. nilai rata-ratanya menjadi 2

Jawab:

b. nilai rata-ratanya menjadi 5

Pembahasan:

Perintah div adalah operator pembagian yang menghasilkan pembulatan ke bawah.

Gunakan program berikut ini untuk menjawab beberapa soal selanjutnya:

```
uses crt;
var j:array['A'..'Z'] of Byte;
c:char;
Kal:string;
procedure HH(S:String);
var i:integer; {baris-6}
m:char;
begin
  for i:= 1 to length(S) do
  begin
    m:=S[i]; {baris-11}
    if m in ['A'..'Z'] then {baris-12}
      inc(j[i]);
    end;
  end;
  begin
    for c:='A' to 'Z' do j[c]:=0;
    Kal:='PASAR';
    HH(Kal);
    for c:='A' to 'Z' do
      if j[c]>0 then write(c,j[c],' ');
    writeln;
    Kal:='RAYA';
    HH(Kal);
    for c:='Z' downto 'A' do
      if j[c]>0 then write(c,j[c],' ');
    writeln;
```

76 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

end.

29. Bila terdapat kesalahan yang menyebabkan program sama sekali tidak dapat dijalankan sebutkan pada

baris berapa, dan bagaimana perbaikannya?

a. Kesalahan semacam ini tidak ada

b. Baris 12, seharusnya ditulis

```
If [m] in ['A'..'Z'] then
```

c. Baris 6 seharusnya ditulis

```
var i: Char;
```

d. Baris 13, seharusnya ditulis

```
Inc(J[m]);
```

e. Index array hanya boleh berupa angka. Jadi deklarasi variabel seharusnya ditulis:

```
const A = 1; Z = 26;
```

```
var J: array[A..Z] of Byte;
```

```
c: Byte;
```

```
Kal: String;
```

dan semua konstanta karakter dalam perintah for harus diganti, misalnya: for c:= A to Z do dan seterusnya

Jawab:

b. Baris 13, seharusnya ditulis

```
inc(J[m]);
```

Pembahasan:

Variabel J adalah variabel dengan tipe data array yang mempunyai indeks ['A'..'Z']. Dalam program

diberikan indeks berupa bilangan bulat, yaitu i. Hal ini akan menghasilkan pesan kesalahan type mismatch.

30. Dengan perbaikan seperti nomor sebelumnya (kalau ada), maka program bisa dijalankan.

Apakah hasil

dari program tersebut?

a. A2 P1 R1 S1

A4 P1 R2 S1 Y1

b. A2 P1 R1 S1

Y1 R1 A2

c. A2 P1 R1 S1

Y1 R2 A4

d. P1 A2 S1 R1

Y1 A4 R2

e. A2 P1 R1 S1

Konsep Dasar Pemrograman Prosedural 77

Tim Olimpiade Komputer Indonesia

Y1 S1 R2 P1 A4

Jawab:

e. A2 P1 R1 S1

Y1 S1 R2 P1 A4

31. Tindakan apakah yang dilakukan oleh subrutin HH ketika dipanggil oleh baris 19 program di atas,

dengan string S berisi kata "PASAR"?

a. Menghitung frekuensi kemunculan huruf-huruf alfabet dan menyimpannya dalam array J

b. Mengumpulkan huruf-huruf alfabet yang muncul lebih dari satu kali ke dalam array J

c. Mencatat letak setiap huruf alfabet ke dalam array J

d. Menentukan huruf yang paling sering dan paling jarang muncul dalam array J

e. Mengurutkan huruf-huruf menurut urutan alfabet dari yang terkecil sampai yang terbesar.

Jawab:

a. menghitung frekuensi kemunculan huruf-huruf alfabet dan menyimpannya dalam array J

78 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Lampiran 2

Contoh Soal dan Pembahasan

Ujian Praktek Pemrograman Pascal

Contoh 1.

Cetak Angka (*Soal ini pernah diberikan pada Practice Session OSN 2003, Balikpapan*)

Ketentuan umum :

Nama Program : CETAK.PAS

Batas Run-time : 1 detik / test case

Nama File Masukan : CETAK.IN

Nama File Keluaran : CETAK.OUT

Tulis sebuah program yang membaca sebuah bilangan bulat n dan mencetak bilangan bulat dari 1 sampai dengan n^2 .

FORMAT MASUKAN (Nama File: CETAK.IN)

Masukan terdiri dari sebuah bilangan bulat n ($1 \leq n \leq 10$).

CONTOH MASUKAN

3

FORMAT KELUARAN (Nama File: CETAK.OUT)

Keluaran terdiri dari n baris. Baris ke- i ($1 \leq i \leq n$) berisi $(2i - 1)$ bilangan bulat, dengan tiap bilangan bulat dipisahkan oleh sebuah spasi. Setiap bilangan bulat terurut menurun sehingga setiap bilangan bulat yang tercetak selalu lebih besar daripada bilangan bulat di sebelah kanannya (bila ada) dan selalu lebih besar daripada bilangan-bilangan bulat pada baris-baris di bawahnya (bila ada).

CONTOH KELUARAN

9

8 7 6

5 4 3 2 1

PEMBAHASAN DAN SOLUSI

Oleh : Ilham Winata Kurnia, Kontestan IOI 2002 Yong-In, Korea

Dalam soal ini, kita diminta untuk menampilkan bilangan 1 sampai dengan N^2 secara terurut menurun.

Selain itu, setiap barisnya, kita perlu menampilkan sejumlah bilangan (bila diperhatikan, pada baris ke- i , ada 2

$\ast i - 1$ bilangan yang ditampilkan, dan ini adalah barisan bilangan ganjil). Jadi, kita cukup melakukan iterasi

hingga ada tepat $2 \ast i - 1$ bilangan yang tercetak pada baris ke- i (dengan menggunakan perintah `write()`).

Setelah itu baru kita cetak pengganti barisnya (dengan menggunakan perintah `writeln()`).

Solusi :

```
program CetakAngka(input, output);
Const InFile = 'CETAK.IN';
OutFile = 'CETAK.OUT';
MAXN = 10;
var i, j, k, n : integer;
begin
  assign(input, InFile);
  assign(output, OutFile);
  reset(input);
  rewrite(output);
  readln(n);
```

```

j := n * n; k := j;
for i := 1 to n do
begin
write(j); dec(j);
while k - j >= (2 * i) - 1 do begin write(' ', j); dec(j); end;
writeln;
end;
close(input);
close(output);
end.

```

82 Konsep Dasar Pemrograman Prosedural
Tim Olimpiade Komputer Indonesia

Contoh 2

Menghitung Perulangan

Ketentuan umum :

Nama Program : HITUNG.PAS

Batas Run-time : 1 detik / test case

Nama File Masukan : HITUNG.IN

Nama File Keluaran : HITUNG.OUT

Diberikan dua buah untaian huruf (*string*), hitung berapa kali string kedua muncul sebagai bagian dari string

pertama. Asumsikan bahwa tiap kemunculan dari string kedua pada string pertama boleh saling menimpa

(*overlap*). Panjang string pertama maksimal 10000, sementara panjang string kedua maksimal 200.

String

didefinisikan sebagai untaian karakter-karakter dengan kode ASCII 32 – 127 yang dibatasi oleh karakterkarakter

dengan kode ASCII yang tidak termasuk dalam jangkauan 32 – 127 tersebut.

FORMAT MASUKAN (Nama File: HITUNG.IN)

Masukan terdiri dari dua baris. Baris pertama berisi string pertama, sementara baris kedua berisi string kedua.

CONTOH MASUKAN

abcdefabcghiabcabcjklmnlabcw

abc

FORMAT KELUARAN (Nama File: HITUNG.OUT)

Keluaran hanya terdiri dari sebuah baris berisi sebuah bilangan bulat yang menyatakan banyak kemunculan

string kedua pada string pertama.

CONTOH KELUARAN

5

Konsep Dasar Pemrograman Prosedural 83

Tim Olimpiade Komputer Indonesia

PEMBAHASAN

Oleh : Ilham Winata Kurnia, Kontestan IOI 2002, Yong In – Korea

Inti dari “Menghitung Perulangan” tidak lain adalah berulang kali menyimulasikan perintah *find*, yang kita

sering temui pada *software-software word processing* seperti Notepad, Edit, Star Office, dan sebagainya. Ada

beberapa cara yang dapat digunakan untuk melakukan hal ini. Akan tetapi, untuk tingkat ini, hanya satu

cara yang akan dibahas di sini, yaitu dengan *Brute Force*.

Permasalahan ini bisa dibagi menjadi 2 bagian: membaca input dan mencari keberadaan substring. Ada satu

hal yang menyebabkan membaca input menjadi bagian tersendiri, yaitu panjang string pertama melebihi

255. Oleh sebab itu, kita tidak dapat serta merta melakukan pembacaan dengan menggunakan `readln`

(walaupun nanti kita akan lihat sebuah pengecualian).

Salah satu cara untuk mengatasi hal ini adalah dengan mendefinisikan sebuah tipe variabel sendiri yang

berupa `array[1..10000] of char`. Untuk membaca sebuah baris pada masukan, maka kita baca karakter demi

karakter sampai kita menemui karakter penanda akhir baris. Di Linux, karakter penanda akhir baris adalah

ASCII #10 alias *newline character*, sementara pada Windows, karakter penanda akhir baris adalah rentetan

ASCII #13 atau *linefeed character* dan ASCII #10.

Seperti yang dikatakan di atas, ada beberapa cara yang dapat dilakukan untuk menentukan keberadaan

substring pada sebuah string, tapi kita akan hanya membahas satu saja. Cara *brute force* mengiterasi semua

karakter pada string dan membandingkan setiap karakter pada bagian string yang sedang diiterasi dengan

karakter-karakter dari substring. Bila setiap perbandingannya adalah benar, maka substring tersebut ada

bagian dari string. Kita hanya cukup menghitung berapa kali perbandingan tersebut benar untuk mengetahui berapa kali kemunculan substring pada string tersebut.

SOLUSI

```
program MenghitungPerulangan(input, output);
Const InFile = 'HITUNG.IN';
OutFile = 'HITUNG.OUT';
MAXA = 10000;
MAXB = 200;
var a : array[1..MAXA + 1] of char;
b : array[1..MAXB + 1] of char;
cta, ctb, ans : integer;
i, j : integer;
cek : boolean;

84 Konsep Dasar Pemrograman Prosedural
Tim Olimpiade Komputer Indonesia
begin
assign(input, InFile);
assign(output, OutFile);
reset(input);
rewrite(output);
cta := 0; ctb := 1; ans := 0;
while not eoln do
begin
inc(cta);
read(a[cta]);
end;
{ read input }
b[ctb] := #13;
while (b[ctb] = #10) or (b[ctb] = #13) do read(b[ctb]);
while not eoln and not eof do
begin
inc(ctb);
read(b[ctb]);
end;
{ find substring locations }
for i := 1 to cta - ctb + 1 do
begin
cek := true;
for j := 1 to ctb do
if a[i + j - 1] <> b[j] then begin cek := false; break; end;
if cek then inc(ans);
```

```

end;
writeln(ans);
close(input);
close(output);
end.

```

Adapun solusi lain yang lebih mudah adalah untuk menggunakan tipe yang diberikan oleh Free Pascal. Dengan menggunakan tipe ini, kita tinggal menggunakan perintah `pos`, `copy`, dan `length`. Hal ini dimungkinkan karena dapat menyimpan untaian karakter dengan jumlah karakter hampir tak terhingga, tapi tetap mempertahankan sifat-sifat yang dimiliki oleh tipe variabel. Dengan demikian, kita

cukup melakukan satu iterasi saja, yaitu iterasi untuk mencari posisi dimana ada kemunculan substring pada

string. Walaupun demikian, waktu yang dibutuhkan oleh program ini lebih kurang sama dengan program

di atas karena perintah sendiri melakukan iterasi untuk mencari posisi substring. Di bawah ini adalah

contoh source codenya:

Konsep Dasar Pemrograman Prosedural 85

Tim Olimpiade Komputer Indonesia

program MenghitungPerulanganAlternatif(input, output);

var a, b : ANSIStrng;

la, i, j, ans : integer;

begin

assign(input, 'HITUNG.IN');

assign(output, 'HITUNG.OUT');

reset(input);

rewrite(output);

readln(a);

readln(b);

la := length(a);

i := 1; j := 1; ans := 0;

{ cari substring sebanyak-banyaknya }

while j <> 0 do

begin

j := pos(b, copy(a, i, la));

i := i + j;

if j <> 0 then inc(ans);

end;

writeln(ans);

close(input);

close(output);

end.

Seperti halnya soal "Cari Maksimum", ada 10 test case yang digunakan. Berikut rinciannya:

No. Panjang

String 1

Panjang

String 2

Keterangan

1 1 1 Tidak ada kemunculan

2 1 2 Tidak ada kemunculan

3 495 1 Mengetes pembacaan input, dengan string 2 adalah sebuah spasi

4 10000 2 String 1 berisi hanya 3 jenis karakter

5 100 3 String 1 berisi hanya 3 jenis karakter

6 500 5 String 1 berisi hanya 2 jenis karakter, sementara string 2 berisi hanya sebuah jenis karakter

7 1520 20 String 1 berisi hanya 4 jenis karakter random dan disisipkan sebuah kemunculan dari string 2

8 10000 200 Mengetes kasus terbesar. String 1 dan 2 hanya terdiri dari sebuah jenis karakter

9 4517 40 Sama seperti 6, tapi string 2 berisi 2 buah jenis karakter

10 8188 100 Sama seperti 1, tapi string 2 juga berisi 3 jenis karakter

86 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Soal 3.

Tempat Tidur Bebek

Nama File : TIDUR.PAS

Pak Dengklek punya N ($1 \leq N \leq 2500$) bebek yang tidur di sebuah kandang besar dengan K kamar yang

dinomori 0 sampai dengan $K-1$. Bebek ke- i dinomori secara unik dengan nomor S_i ($1 \leq S_i \leq 1000000$). Setiap

bebek tahu di mana untuk tidur karena dia tidur di kamar nomor $S_i \bmod K$. Tentu saja, para bebek tidak mau

membagi tempatnya untuk tidur.

Diberikan sebuah himpunan bebek dan nomornya, tentukan nilai minimum K sedemikian sehingga tidak

ada 2 bebek yang tidur di kamar yang sama.

FORMAT MASUKAN (TIDUR.IN)

Baris 1 : Sebuah bilangan bulat untuk N

Baris 2.. $N+1$: Sebuah bilangan bulat yang menyatakan nomor bebek

CONTOH MASUKAN

5
4
6
9
10
13

FORMAT KELUARAN (TIDUR.OUT)

Sebuah baris dengan nilai minimum K pada baris tersebut.

CONTOH KELUARAN

8

Konsep Dasar Pemrograman Prosedural 87

Tim Olimpiade Komputer Indonesia

PEMBAHASAN

Oleh : Ilham Winata Kurnia, Anggota TOKI 2002, Yong-In - Korea Selatan

Soal ini merupakan terjemahan dari soal berjudul "Cows in Beds" yang pernah dikeluarkan oleh United States of America Computing Olympiad (USACO) dalam salah satu kontesnya. Berikut adalah

solusi dari salah satu pesertanya, Bruce Merry, yang berhasil melewati tes data yang dibuat oleh USACO (tes datanya lebih menantang daripada yang digunakan untuk seleksi TOKI).

Cara yang paling terlihat untuk menyelesaikan masalah ini adalah untuk melakukan loop melalui semua

kemungkinan nilai K dan pada setiap iterasi, kita cari di mana setiap bebek tidur sambil menggunakan

sebuah *array of boolean* untuk mengecek adanya perulangan (yaitu, ada dua bebek dalam sebuah kamar).

Untuk mempercepat, kita dapat memulai iterasi dari N karena tidak mungkin kita dapat menempatkan N

bebek di kurang dari N kamar tanpa adanya yang membagi tempat tidurnya (tampak jelas, tapi para

matematikawan berpikir bahwa ini cukup penting untuk disebut *pigeon hole principle* atau prinsip burung dara).

Cara yang lebih ambisius adalah untuk menandai (dengan menggunakan sebuah *array of boolean*) semua nilai

dari K yang tidak dapat digunakan dan carilah nilai pertama yang benar. Nilai-nilai K yang tidak dapat

digunakan adalah faktor dari $|S_i - S_j|$ untuk semua nilai (i, j) . Sayangnya, ada banyak sekali pasangan (i, j) ,

dan mencari faktor-faktor adalah sangat lambat. Mungkin kita bisa membuat beberapa trik yang memanfaatkan fakta bahwa ada **banyak** faktor yang diulang, bahkan nilai selisih yang berulang.

Namun,

dengan batas memori 16MB kita akan kesulitan untuk menyeimbangkan antara batas waktu dan memori,

dan selain itu kita masih melakukan banyak sekali pembagian.

Pendekatan yang Bruce Merry gunakan adalah untuk menandai $|S_i - S_j|$ sebagai nilai K yang tidak dapat

digunakan, untuk setiap pasang (i, j) , lalu menggunakan algoritma yang pertama tapi dengan melewati nilai-nilai

tersebut. Karena nilai dari $|S_i - S_j|$ adalah dalam rentang 0..1000000, maka memori bukan suatu masalah.

Selain itu, karena tidak ada operasi pembagian atau perkalian, maka cara ini relatif cepat. Ide dasarnya

adalah bahwa untuk tes data yang besar (tentu saja, tes-tes ini adalah tujuan dari optimasi kita) kita perlu

menghilangkan sebanyak mungkin nilai K sehingga mengecek nilai-nilai yang tersisa dengan cara pertama

adalah jauh lebih cepat.

Salah satu optimasi lain adalah untuk juga menandai nilai $|S_i - S_j| / 2$ dimana $|S_i - S_j|$ adalah bilangan genap.

Ini menjamin bahwa semua nilai K antara 333334 dan 1000000 ditandai karena semua faktor lain dari $|S_i - S_j|$

88 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

harus paling besar $1000000/3=333333$. Dengan kata lain, begitu nilai $K > 333333$, maka kita cukup melakukan

sebuah tes penuh pada nilai K berikutnya yang tidak ditandai, karena nilai tersebut adalah jawabannya.

SOURCE CODE (Oleh : Bruce Merry)

```
program cows_in_bed;
const
  inname = 'TIDUR.IN';
  outname = 'TIDUR.OUT';
var
  N : integer;
  K : longint;
  Si : array[1..5000] of longint;
  used : array[0..999999] of boolean;
  nogood : array[1..1000000] of boolean;
procedure readin;
var
  f : text;
  i : integer;
begin
  assign(f, inname);
  reset(f);
  readln(f, N);
  for i := 1 to N do
    readln(f, Si[i]);
  close(f);
end;
procedure makenogood;
var
  i, j : integer;
begin
```

```

fillchar(nogood, sizeof(nogood), 0);
for i := 1 to N - 1 do
  for j := i + 1 to N do
    nogood[abs(Si[i] - Si[j])] := true;
  end;
procedure solve;
var
  i : integer;
  cur : longint;
  flag : boolean;
begin
  K := N - 1;
  flag := false;
  repeat
    inc(K);
    if nogood[K] then continue;
    flag := true;
    fillchar(used, sizeof(used[0]) * K, 0);
    for i := 1 to N do

```

Konsep Dasar Pemrograman Prosedural 89

Tim Olimpiade Komputer Indonesia

```

begin
  cur := Si[i] mod K;
  if used[cur] then
    begin
      flag := false;
      break;
    end;
  used[cur] := true;
end;
until flag;
end;
procedure writeout;
var
  f : text;
begin
  assign(f, outname);
  rewrite(f);
  writeln(f, K);
  close(f);
end;
begin
  readin;
  makenogood;
  solve;
  writeout;
end.

```

90 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Contoh 4.

Spiral Huruf

Huruf-huruf yang disusun dalam konfigurasi di bawah ini adalah huruf alfabitis biasa.

Tulislah program

untuk mencetak spiral huruf dengan ukuran N ($1 \leq N \leq 5$, dan N ganjil) searah jarum jam. Spiral bisa dibuat

dari tengah, atau dari salah satu pojok (tergantung dari inputnya). Data masukan adalah N dan A. A = 1

artinya tengah, A = 2 artinya pojok kiri atas, A = 3 artinya pojok kanan atas, A = 4 artinya pojok kanan

bawah, A = 5 artinya pojok kiri bawah.

FORMAT MASUKAN

Data input disimpan dalam text file SPIRAL.IN. File ini berisi nilai N dan A.

FORMAT KELUARAN

Simpan hasil tampilan spiral huruf pada file SPIRAL.OUT.

CONTOH MASUKAN

3 4

CONTOH KELUARAN

E F G

D I H

C B A

PEMBAHASAN

Dalam masalah ini, kita diinginkan untuk membuat sebuah program yang dapat menghasilkan sebuah spiral

huruf dalam bentuk mirip matriks $N \times N$ dengan cara yang sudah ditentukan (A). Pengertian spiral huruf

dapat langsung dimengerti dengan melihat contoh keluaran. Ada sebuah frase kunci yang amat penting

untuk menyelesaikan masalah ini, yaitu “searah jarum jam”.

Sebetulnya, soal ini tidak jelas karena tidak memberikan keterangan bagaimana spiral harus dibentuk bila A

= 1. Ada 4 kemungkinan, yaitu bergerak ke atas, kiri, bawah dan kanan setelah mengisi kotak paling tengah.

Konsep Dasar Pemrograman Prosedural 91

Tim Olimpiade Komputer Indonesia

Oleh karena itu, mari kita asumsikan bahwa setelah mengisi kotak yang ditengah maka kita bergerak ke

sebelah kanan. Misal, untuk input 3 1, maka outputnya akan menjadi

G H I

F A B

E D C

SOLUSI 1

Bila kita melihat batas N, maka kita dapat simpulkan bahwa N hanya dapat bernilai 1, 3 dan 5.

Karena untuk

setiap N hanya ada 5 kemungkinan nilai A, maka total hanya ada 15 kemungkinan input.

Mengingat jumlah

yang begitu kecil ini, maka kita bisa mengerjakan semuanya secara manual, kemudian memasukkannya ke

dalam program secara manual. Dengan menggunakan perintah if-then-else, langsung cetak solusinya yang

telah kita masukkan secara manual ke dalam program.

SOLUSI 2

Cara yang lain untuk menyelesaikan masalah ini adalah untuk membuat program yang menyimulasikan

pergerakan posisi huruf. Bayangkanlah output itu terletak pada suatu system koordinat kartesius. Posisi

pojok kiri bawah memiliki koordinat (1, 1) dan posisi pojok kanan atas memiliki koordinat (N, N). Dalam

contoh di atas, C berkoordinat (1, 1) dan G (3, 3). Kemudian, setiap huruf disimpan dalam sebuah tabel 2

dimensi dengan besar $N \times N$.

Kita misalkan koordinat-x dengan variabel x dan koordinat-y dengan variabel y . Bila kita ingin bergerak ke

atas, maka x kita tambah dengan 1; turun, x kita kurang dengan 1; kanan, y kita tambah dengan 1; kiri, y kita

kurang dengan 1. Karena dalam soal dikehendaki untuk mencetak spiralnya searah jarum jam, maka marilah

kita buat aturan arah gerak sebagai berikut: kanan-bawah-kiri-atas-kanan-.... Yang sekarang jadi masalah

adalah kapan kita harus berubah arah gerak.

Dengan mencoba-coba, maka kita dapat aturan sebagai berikut:

- Bila $A = 1$ maka:

(x, y) = koordinat tengah tabel

Tabel[x, y] = "A"

Arah gerak = atas

92 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

BILA masih ada kotak kosong LAKUKAN

JIKA bisa bergerak dengan Arah gerak berikutnya MAKA

ubah Arah gerak menjadi Arah gerak berikutnya

JIKA TIDAK MAKA

bergerak sesuai dengan Arah gerak yang sekarang

Tabel[x, y] = huruf selanjutnya

- Bila $A = 2$ maka:

(x, y) = koordinat pojok kiri atas

Tabel[x, y] = "A"

Arah gerak = kanan

BILA masih ada kotak kosong LAKUKAN

JIKA bisa bergerak dengan Arah gerak saat ini MAKA

bergerak sesuai dengan Arah gerak saat ini

Tabel[x, y] = huruf selanjutnya

JIKA TIDAK MAKA

Arah gerak = Arah gerak berikutnya

- Langkah penyelesaian untuk $A = 2$ berlaku juga untuk $A = 3$, $A = 4$, dan $A = 5$. Hanya saja, posisi

awal (x, y) diubah sesuai dengan yang tertulis di soal dan arah gerak pertama kali juga disesuaikan.

Misal, untuk $A = 5$, maka posisi awal (x, y) adalah koordinat pojok kiri bawah dan Arah gerak = atas.

Konsep Dasar Pemrograman Prosedural 93

Tim Olimpiade Komputer Indonesia

Contoh 5.

Pembagian Panjang

Untuk setiap integer [bilangan bulat] yang diberikan, cetaklah hasil pembagian integer jika sebuah integer

masukan dibagi dengan 13. Sisa pembagian tidak perlu di tulis. Jika hasil pembagian adalah 0, maka tuliskan

saja nol sebagai outputnya.

FORMAT MASUKAN

Terdiri atas sejumlah baris integer yang minimal terdiri atas 1 digit dan paling panjang 50 digit tanpa spasi

sebelum dan sesudah integer di setiap barisnya .

FORMAT KELUARAN

Hasil pembagian integer dari data input dengan 13.

CONTOH

MASUKAN KELUARAN

0 0

12 0

13 1

14 1

25 1

26 2

262626 20202

13131313131313131313131313 1010101010101010101010101010101

PEMBAHASAN

Apa yang dituntut dari soal ini cukup jelas yaitu untuk mencetak hasil bulat dari suatu bilangan bulat yang

dibagi dengan 13. Akan tetapi ada dua hal yang menjadi masalah. Pertama, panjang bilangan bulat bisa

mencapai 50 digit. Kedua, tidak tertulis apakah input selalu merupakan bilangan bulat positif atau bisa juga

94 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

merupakan bilangan bulat negatif. Untuk masalah yang kedua, amat mudah untuk dipecahkan dengan 1

perintah if. Akan tetapi, untuk menyelesaikan masalah yang pertama butuh lebih dari itu.

Berikut ini kita

asumsikan bahwa input telah diubah ke bilangan bulat positif.

Dengan adanya petunjuk bahwa panjang input bisa mencapai 50 digit, maka kita tidak bisa menyimpan

input dalam suatu variabel bilangan biasa. Longint hanya bisa menampung maksimum 10 digit bilangan

bulat. Comp (dalam Turbo Pascal) atau Int64 (dalam Free Pascal) hanya sanggup menampung 19 digit

bilangan bulat. Mungkin ada yang beranggapan bahwa variabel-variabel *floating point* seperti Real (dalam

Pascal) atau Double dapat menampung bilangan bulat tersebut. Hal itu memang benar, tapi yang disimpan

hanya bisa sampai 23 digit pertama beserta tingkat eksponennya dalam basis 10 (sistem desimal). Bila

pembagian dilakukan, maka hasilnya hanya akurat sampai digit ke 22. Dengan demikian, kita harus

memikirkan suatu teknik penyimpanan dengan menggunakan tipe variabel dasar yang tersedia.

Adalah suatu yang alami bila kita berpikir untuk menggunakan array. Misalkan input kita simpan dalam

array dari integer dengan panjang 50. Setiap digit dari input kita simpan dalam 1 elemen.

Sekarang timbul

masalah baru, yaitu bagaimana cara mendapatkan setiap digit dari input. Ada beberapa cara.

Salah satunya

adalah untuk membaca tiap baris sebagai sebuah string. Setelah itu, setiap karakter dari string tersebut

dikonversi ke nilai digit menggunakan nilai ASCII dari karakter tersebut. Kemudian, panjang string itu

menjadi panjang bilangan.

Setelah kita mendapatkan input tersebut dalam array, maka kita tinggal melakukan pembagian panjang

dengan cara sesuai dengan yang telah diajarkan sewaktu kita masih duduk di sekolah dasar.

Berikut ini kode

semu (*pseudocode*) langkah-langkah untuk memecahkan soal ini. Sekedar catatan: “\” adalah pembagian

yang hanya menghasilkan bilangan bulatnya saja.

Bil = jawab = array dari integer dengan panjang 50.

```

Pan Bil = Panjang dari Bil
Pan jawab = Panjang dari jawab = 0
Ubah input ke dalam array Bil
Sisa = 0
DARI I = 1 SAMPAI DENGAN Pan Bil LAKUKAN
Sisa = Sisa x 10 + Bil[I]
Pan jawab = Pan jawab + 1
jawab[Pan jawab] = Sisa \ 13

```

Konsep Dasar Pemrograman Prosedural 95

Tim Olimpiade Komputer Indonesia

Sebuah catatan kecil, untuk mencetak jawabannya, masih diperlukan ketelitian karena dalam ada kasus di

mana jawabannya = 0 dan kasus di mana jawab[1] = 0 dan jawab[2] = 0.

96 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Contoh 6.

Warisan *(Sumber : CEOI 1995)*

Dua orang bersaudara, Pak Dingklik dan Pak Dengklek ingin membagi beberapa barang warisan yang

didapat dari almarhum orang tuanya. Setiap barang yang ada, harus diberikan kepada Pak Dingklik dan Pak

Dengklek. Hanya saja, masalahnya setiap barang warisan tersebut tidak dapat dibagi menjadi dua. Supaya

adil, masing-masing warisan tersebut diberikan sebuah nilai dalam bentuk bilangan bulat yang positif.

Misalkan A dan B melambangkan total nilai dari keseluruhan warisan yang diterima oleh Pak Dingklik dan

Pak Dengklek. Tujuan yang hendak dicapai adalah meminimalkan selisih dan A dan B. Buat sebuah

program untuk menghitung nilai A dan B.

FORMAT MASUKAN

Baris pertama dari file WARIS.IN terdiri dari N, jumlah dari warisan yang ada ($1 \leq N \leq 100$).

Baris

berikutnya berisi N bilangan bulat positif, yang merupakan nilai dari setiap warisan. Tiap nilai ≤ 200 .

FORMAT KELUARAN

Tulis pada file WARIS.OUT nilai A dan B ($A < B$)

CONTOH MASUKAN

7

28 7 11 8 9 7 27

CONTOH KELUARAN

48 49

Konsep Dasar Pemrograman Prosedural 97

Tim Olimpiade Komputer Indonesia

PEMBAHASAN

(Oleh Windra Swastika, Mantan peserta seleksi TOKI)

Jika masalah ini diselesaikan secara brute force (dengan mencoba semua kemungkinan yang ada), maka akan

mempunyai kompleksitas sebesar $O(2^N)$. Jika $N=100$, maka komputasi akan sangat memakan waktu yang

besar. Untuk itu penyelesaian secara brute force adalah tidak disarankan untuk soal ini.

Pendekatan yang dilakukan adalah secara pemrograman dinamis (Dynamic Programming /DP).

Konsep DP

adalah menyelesaikan masalah dengan membagi menjadi sub masalah. Dari sub masalah yang terkecil (yang

lebih mudah dicari nilai optimalnya), proses akan berlanjut untuk memecahkan masalah di

atasnya (dengan

memanfaatkan nilai optimal dari masalah sebelumnya). Nilai-nilai optimal tadi disimpan dalam sebuah

tabel. DP menukar space dengan time!

Untuk soal ini, sebuah tabel yang mempunyai kombinasi penjumlahan untuk $j=1, j=2, j=3 \dots j=N$.

$TABEL[j] = 0$ jika $j=0$

$TABEL[j] = 1..N$ artinya warisan yang terakhir yang diberikan untuk mendapatkan subtotal j

$TABEL[j] = 101$ artinya subtotal j tidak bisa dihasilkan

Misalkan W adalah array untuk menyimpan nilai dari setiap warisan. Nilai dari subtotal j akan ditambahkan

untuk mendapatkan subtotal baru ($j+W[I]$) dengan syarat: $TABEL[j]<i$ (nilai j adalah hasil dari subtotal yang

pernah dicapai sebelumnya).

SOLUSI

```
{
=====
-- WARIS.PAS --
=====
Windra Swastika
}
const
finput='waris.in';
foutput='waris.out';
kosong=101;
var
tabel:array [0..10000] of byte;
warisan:array [1..100] of byte;
f:text;
i:byte;

98 Konsep Dasar Pemrograman Prosedural
Tim Olimpiade Komputer Indonesia
j:word;
n:byte;
sum:word;
begin
assign(f,finput);
reset(f);
readln(f,n);
for i:=1 to n do
begin
read(f,warisan[i]);
inc(sum,g[i]);
end;
close(f);
fillchar(tabel,sizeof(tabel),kosong);
tabel[0]:=0; {nilai awal untuk tabel[0]}
for i:=1 to n do
for j:=0 to sum shr 1-warisan[i] do
if (tabel[j]<i) and (tabel[j+warisan[i]]=kosong) then
tabel[j+warisan[i]]:=i;
assign(f,foutput);
rewrite(f);
for i:=sum shr 1 downto 0 do {cukup mulai dari setengah sum}
if tabel[i]<>kosong then
begin
writeln(i,' ',sum-i);
break;
end;
close(f);
end.

Konsep Dasar Pemrograman Prosedural 99
```


Contoh 7.

Firetruck (Sumber : ACM 1991 - Problem A)

Departemen pemadam kebakaran kota Arjosari bekerja sama dengan dinas transportasi lokal untuk mengolah peta yang menunjukkan status jalan-jalan pada kota Arjosari. Sialnya, jalan-jalan di kota itu pada hari tertentu harus ditutup karena adanya pawai mingguan. Akibatnya, para petugas pemadam kebakaran harus mencari rute jalan yang paling pendek jika hendak memadamkan api pada jalan tertentu di kota

Arjosari.

Jika pada suatu saat, departemen pemadam kebakaran menerima laporan terjadinya kebakaran pada jalan

tertentu, maka pihak departemen pemadam kebakaran segera meminta daftar jalan-jalan yang dapat dilalui

ke dinas transportasi. Dinas transportasi mengirimkan semua jalan yang pada saat itu bisa dilalui. Buat

sebuah program untuk mencari semua rute yang diawali dari kantor pusat departemen pemadam kebakaran

menuju jalan yang dituju.

FORMAT MASUKAN (fire.in)

Baris pertama berisi bilangan bulat yang merupakan jalan terdekat dari pusat api yang harus dituju

($2 \leq N < 21$).

Baris-baris berikutnya merupakan pasangan bilangan bulat kurang dari 21 (dipisahkan dengan spasi) yang

menunjukkan jalan yang dapat dilalui. Baris terakhir pada file input adalah pasangan bilangan 0 0. Misalkan

pasangan jalan 4 7 ada pada file input, artinya bahwa jalan 4 dan 7 dapat dilalui, begitu juga sebaliknya.

FORMAT KELUARAN (fire.out)

Semua kemungkinan rute yang dapat menuju ke N yang dimulai dari 1.

Setiap baris berisi urutan jalan yang harus ditempuh diawali dari 1 dan diakhiri di N. Agar perjalanan

petugas pemadam kebakaran efisien, truk pemadam kebakaran tidak boleh menempuh jalan yang pernah

dilalui lebih dari sekali.

100 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

CONTOH MASUKAN

6

1 2

1 3

3 4

3 5

4 6

5 6

2 3

2 4

0 0

CONTOH KELUARAN

1 2 3 4 6
1 2 3 5 6
1 2 4 3 5 6
1 2 4 6
1 3 2 4 6
1 3 4 6
1 3 5 6

Soal ini termasuk soal-soal yang menggunakan dasar teori graf. Pada teori graf istilah dikenal verteks dan edge. Yang dimaksud dengan verteks adalah titik dan edge adalah garis. Sebuah graf dinotasikan dengan

$e=[u,v]$, yang berarti bahwa edge e berawal pada verteks u dan berakhir pada verteks v .

Pada Firetruck, terdapat maksimal 21 verteks. Sehingga contoh pada file input dapat dinotasikan $e1=[1,2]$,

yang artinya edge $e1$ berawal pada verteks 1 dan berakhir pada verteks 2. Dari file input, kita bisa

mendapatkan sebuah graf seperti pada gambar berikut:

Konsep Dasar Pemrograman Prosedural 101

Tim Olimpiade Komputer Indonesia

Dengan diagram seperti di atas, kita dapat membuat sebuah struktur data yang berisi jalur untuk masing-masing

verteks. Dalam teori graf, jalur tersebut disebut dengan matriks jalur (*path matrix*).

```
const
max_route=100
type
tarrstreet=array[1..max_route,1..max_route] of boolean
var
arrstreet:tarrstreet;
```

Variabel `arrstreet` adalah sebuah matriks jalur (array dengan tipe boolean) yang menunjukkan adanya jalur

dari verteks ke verteks. Dengan struktur data tersebut serta menggunakan teknik pencarian DFS (Depth First

Search), maka dapat dengan mudah dibuat prosedur penelusuran dari verteks awal (1) menuju verteks

tertentu.

```
const
finput='fire.in';
foutput='fire.out';
max_route=100;
{
=====
-- FIRE.PAS --
=====
Windra Swastika
}
```

102 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

```
type
tstreet=1..21;
troute=array[0..max_route] of tstreet;
tarrstreet=array[1..max_route,1..max_route] of boolean;
var
fi,fo:text;
map:tarrstreet;
used=array[1..max_route] of boolean;
destination:tstreet;
i,j:byte;
N:byte;
route:troute;
procedure seekroute(depth:byte;start:tstreet);
```

```

var
i:byte;
begin
if start=destination then
begin
for i:=0 to depth-1 do
write(fo,route[i],' ');
writeln(fo);
end else
for i:=1 to N do
begin
if map[start,i] and not used[i] then
begin
used[start]:=true;
route[depth]:=i;
seekroute(depth+1,i);
used[start]:=false;
end;
end;
end;
begin
assign(fi,'fire.in');
reset(fi);
readln(fi,destination);
fillchar(map,sizeof(tarrstreet),false);
fillchar(used,sizeof(used),false);
fillchar(route,sizeof(troute),0);
N:=0; {number of street}
repeat
readln(fi,i,j);
if (i>=0) and (j>=0) then
begin
map[i,j]:=true;
map[j,i]:=true;
inc(N);
end;

```

Konsep Dasar Pemrograman Prosedural 103

Tim Olimpiade Komputer Indonesia

```

until (i=0) and (j=0);
close(fi);
assign(fo,'fire.out');
rewrite(fo);
route[0]:=1; {start from street #1}
used[1]:=true; {street #1 already used}
seekroute(1,1);
close(fo);
end.

```

104 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Index

A

algoritma · 2, 7, 49, 80

Algoritma · 7

append · 44, 45

Array · 39, 40

ASCII · 16, 75, 76, 87

Assign · 44, 47, 64, 65

Assignment · 11, 44

B

bahasa assembly · 2

Bahasa assembly · 2

bahasa mesin · 2, 3, 4

bahasa Pascal · 4, 5, 9, 10, 11, 16,

18, 20, 57

biner · 14, 19, 43

Boolean · 13, 18, 19

bug · 4

C

Case · 24, 56

case of · 25

Character · 40

compile time · 38

D

debugger · 1, 4

deklaratif · 1

E

Editor · 4

eksekusi · 1, 3, 4, 27, 38

Eksekusi · 2

EKSEKUSI · 1

Ekspresi · 8, 15, 18, 21

Elemen · 7

Enum · 37

Enumerasi · 37

eof · 46, 77

F

faktorial · 49, 50

false · 18, 19, 28, 46, 77, 81, 82, 95

File · 8, 43, 44, 45, 46, 73, 75, 79, 83

FOR · 27

FreePascal · 5, 13, 18, 22

Fungsi · 18, 22, 31, 34, 46, 49, 62, 63

fungsional · 1

G

global · 9, 33

H

hardware · 2, 4

hardware) · 4

himpunan · 38, 39, 54, 55, 59, 79

I

IDE · 4

Identifier · 10

if - then · 23

if - then - else · 23

Indeks loop · 27

input · 8, 13, 32, 44, 45, 46, 47, 59,

74, 76, 77, 78, 83, 84, 86, 87, 92,

93

Input · 7, 8, 13

Instruksi · 8, 23, 24

Integer · 13, 14, 15, 19, 37, 53, 61,

63, 66, 67, 68

Integrated Development

Environment · 4

interpreter · 1, 2, 3, 4

106 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

Interpreter · 2, 3, 4

intersection · 39

Invariant Loop · 29

iteratif · 50

K

Karakter · 13, 15, 16, 43
keyword · 9, 68
kompilator · 1, 2, 3, 4, 5, 9, 10, 18,
22, 28, 38, 40
Kompilator · 3, 4, 5
Konstanta · 8, 12
Konversi · 21, 22

L

lokal · 33, 92
loop · 14, 27, 28, 29, 40, 41, 46, 50,
80

M

mekanisme · 1
MEKANISME · 1
MOD · 20
Moduler · 8

N

NESTED LOOP · 28

O

objek · 1
operand · 8, 15
operator · 8, 15, 19, 54, 55, 58, 68,
69
ord · 16, 40
output · 8, 13, 45, 46, 57, 65, 66, 74,
76, 77, 78, 84
Output · 7, 8, 13, 59, 60

P

paradigma · 1, 2, 7
Paradigma · 1, 2
PARADIGMA · 1
parameter · 32, 33, 34, 45, 46, 52,
63, 67, 68
Parameter · 32, 33
pemrograman · 1, 5, 7, 8, 11, 12, 13,
15, 31, 34, 38, 90, 100
PEMROGRAMAN · 1, 7
precedence · 15
program · 1, 2, 3, 4, 5, 8, 9, 10, 11,
12, 17, 24, 27, 29, 31, 33, 34, 38,
43, 44, 45, 47, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68,
69, 70, 73, 74, 76, 77, 78, 81, 83,
84, 89, 92
PROGRAM · 1, 9, 66
programmer · 2, 4
prosedur · 10, 13, 22, 31, 32, 33, 34,
44, 45, 49, 52, 54, 94
prosedural · 1, 2, 7, 8, 12, 31, 38
proses · 4, 8, 16, 19, 21, 44, 90
Proses · 4, 7, 8, 44

R

read · 13, 14, 39, 44, 46, 77, 91
readln · 13, 14, 18, 37, 38, 39, 44, 45,
46, 47, 59, 74, 76, 77, 78, 81, 95
real · 11, 14, 19, 20, 21, 22, 37, 39,
52, 53, 68

Real · 13, 19, 20, 62, 63, 68, 87
reference · 34
rekurens · 49, 50
Rekurens · 8
REKURENS · 49
rekursif · 49, 50
REPEAT - UNTIL · 28
representasi · 1, 12, 14, 17, 19, 21
reset · 44, 45, 47, 74, 77, 78, 81, 91, 95
rewrite · 44, 45, 47, 74, 77, 78, 82, 91, 96
run time · 38

S

Scope · 33
seksadesimal · 38
sekuensial · 2, 8
SET · 38
sintaks · 2, 40
software · 4, 76, 100
STDIN · 46
STDOUT · 46
String · 13, 17, 40, 53, 61, 63, 69, 70, 75, 78
Struktur Data · 8
subprogram · 31
Subtipe · 37, 38

Konsep Dasar Pemrograman Prosedural 107
Tim Olimpiade Komputer Indonesia

T

Tabel · 39, 40, 84, 85
teks · 4, 43, 44, 45
Tipe · 7, 12, 13, 14, 16, 18, 19, 20, 21, 37, 38, 39, 52, 53
TOKI · 31, 43, 46, 47, 60, 61, 62, 80, 90, 100
true · 18, 19, 28, 46, 77, 81, 82, 95, 96

U

unary · 15
underscore · 10
UNICODE · 16, 17
union · 39, 54

V

valid · 10, 38, 53, 57, 67
variabel · 10, 11, 12, 13, 14, 15, 17, 20, 33, 34, 38, 39, 43, 44, 53, 54, 63, 64, 67, 68, 70, 76, 77, 84, 87
Variabel · 8, 11, 13, 20, 33, 35, 54, 68, 70, 94

W

WHILE - DO · 27
write · 13, 32, 39, 45, 46, 58, 59, 69, 74, 95
writeln · 10, 12, 13, 16, 17, 18, 20, 21, 23, 24, 27, 29, 31, 32, 33, 37, 39, 40, 45, 46, 47, 53, 56, 58, 59,

60, 61, 63, 69, 74, 77, 78, 82, 95

108 Konsep Dasar Pemrograman Prosedural

Tim Olimpiade Komputer Indonesia

BIODATA SINGKAT PENULIS

Yohanes Nugoroho, Sarjana Teknik Informatika lulus tahun 2002 dari Departemen Informatika Institut Teknologi Bandung saat ini bekerja sebagai teaching assistant di Departemen Teknik Informatika ITB dan merupakan salah satu pembina di TOKI Biro ITB. Saat ini juga sedang menyelesaikan studi lanjutan S2-nya di institusi yang sama.

Selain menguasai hampir semua bahasa pemrograman, penulis adalah salah satu Pengajar/Pembina di Tim Olimpiade Komputer Indonesia, yang handal. Penulis juga aktif pada kegiatan Opensource di Indonesia dan banyak terlibat dalam proyek-proyek pembangunan software., serta aktif menghasilkan berbagai macam tulisan/artikel tentang teknologi informasi.