

BAB IX

INPUT DAN OUTPUT

INPUT

Input merupakan sesuatu data yang dimasukkan ke dalam program untuk diproses. Dalam hal ini, hanya akan dibahas input yang diperoleh dari entri data melalui keyboard.

Perintah Pascal untuk menyatakan input adalah:

```
readln(variabel);
```

Melalui perintah di atas, data yang dientrikan melalui keyboard akan diassign ke dalam suatu variabel. Setelah data dientrikan, user harus menekan tombol ENTER (RETURN)

Contoh:

```
Program ContohInput;  
Var  
    Bil : integer;  
Begin  
    Write('Masukkan sebarang bilangan bulat : ');  
    Readln(bil);  
    Write('Bilangan yang Anda masukkan adalah : ',bil);  
End.
```

Misalkan ketika program di atas dijalankan kita masukkan bilangan 6, maka tampilannya:

```
Masukkan sebarang bilangan bulat : 6      [enter]  
Bilangan yang Anda masukkan adalah : 6
```

Catatan:

- Untuk input berupa bilangan real yang memuat desimal, digunakan tanda titik yang menyatakan koma, misal 345.12. Selain itu dapat pula dituliskan dalam bentuk format scientific, misal 1E-5 (sama dengan 0.00001) atau 3.5E+4 (sama dengan 35000)
- Sedangkan input yang berupa karakter tidak perlu diapit dengan tanda petik, demikian pula untuk tipe data boolean.

OUTPUT

Setelah data diinput dan selanjutnya diproses, selanjutnya harus ditampilkan sebagai output supaya user dapat mengetahui solusi dari permasalahan. Untuk menampilkan data ke layar (output), terdapat dua perintah dalam Pascal, yaitu

```
Write(argumen);  
Writeln(argumen);
```

Perbedaan dari kedua perintah di atas adalah sbb:

Apabila digunakan `write`, maka argumen yang ditulis akan ditampilkan ke layar. Setelah itu kursor akan berada di sebelah kanan dari argumen yang telah ditulis tadi. Sehingga apabila setelah `write` tersebut terdapat `write` atau `writeln` lagi, maka argumen yang baru akan ditulis di sebelah kanan dari argumen yang lama.

Sedangkan apabila digunakan `writeln`, maka argumen yang ditulis akan ditampilkan ke layar. Setelah itu kursor akan berada di bawah argumen yang telah ditulis tadi. Sehingga apabila setelah `write` tersebut terdapat `write` atau `writeln` lagi, maka argumen yang baru akan ditulis di bawah dari argumen yang lama.

Catatan:

Argumen dapat berupa variabel, string, operasi, maupun function.

Contoh:

```
a := 4;      {misalkan a dan b adalah integer}
b := 5;
write('Ini contoh write');
write('Ini contoh write');
writeln('Ini contoh writeln');
writeln('Ini contoh writeln');
writeln(a);
write(a+b);
writeln(sqr(a+b));
writeln(a,' ditambah ',b,' adalah ', a+b);
```

output dari potongan program di atas adalah

```
Ini contoh writeIni contoh writeIni contoh writeln
Ini contoh writeln
4
981
4 ditambah 5 adalah 9
```

FORMATTING OUTPUT

Tampilan output dapat diatur jaraknya. Untuk tipe data karakter/string serta integer dapat menambahkan sintaks berikut ini dalam `write/writeln`.

`value/variabel : lebarfield`

dengan `value/variabel` merupakan nilai atau isi dari variabel yang akan ditampilkan ke layar. Sedangkan `lebarfield` merupakan bilangan positif yang menunjukkan lebar space untuk nilai yang akan ditampilkan.

Contoh:

```
a := 10;
write('Ini adalah isi dari a :',a:5);
```

Hasil outputnya adalah

Ini adalah isi dari a : 10

Nilai 5 berarti terdapat 5 space untuk meletakkan 10 (cara pengisian adalah rata kanan), sehingga terdapat sisa 3 buah space di depan 10. Hal yang sama juga berlaku untuk nilai yang berupa karakter maupun string.

Bagaimana dengan nilai yang berupa real? Bagaimana cara mengatur jumlah digit di belakang koma? Berikut ini adalah sintaks untuk ditambahkan dalam write/writeln.

Value/variabel : lebarfield : jumlahdigit

Contoh:

```
a := 10.2345562;  
write('Ini adalah isi dari a : ',a:7:2);
```

Outputnya adalah

Ini adalah isi dari a : 10.23

Perhatikan tampilan di atas. Masih terdapat jarak 2 spasi di depan 10.23. Hal ini disebabkan untuk menuliskan 10.23 diperlukan 5 space, sehingga dari 7 space yang ada, hanya digunakan 5 space saja.