

BAB III

ALGORITHMMA DAN BILANGAN ACAK

3.1 PENGERTIAN ALGORITHMMA

Untuk memperoleh hasil proses komputer sesuai dengan yang dikehendaki, kepada komputer haruslah diberikan se-rangkaian perintah (intruksi) yang dinamakan program.

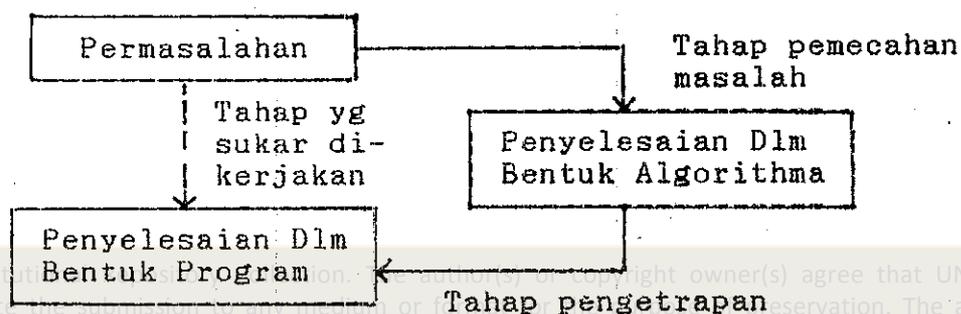
Oleh sebab itu penulisan program harus jelas, sehingga tidak menimbulkan bermacam-macam penafsiran.

Meskipun sebuah kalimat seperti "hitunglah rata-rata dari nilai test ini" tampaknya sudah menjelaskan perintah per-hitungan yang dikehendaki, sebenarnya kalimat ini jauh dari tepat bila dituliskan dalam bentuk program.

Hal ini disebabkan banyak hal-hal penting yang terabaikan sehingga dapat menimbulkan bermacam-macam penafsiran, seperti misalnya dimanakah nilai test itu, berapa jumlahnya, apakah yang tidak hadir juga diperhitungkan.

Penulisan program merupakan penulisan tahap pemecahan masalah dan tahap pengetrapan pemecahan masalah dalam suatu bahasa pemrograman yang dikehendaki.

Agar penulisan program dapat dilakukan lebih mudah, maka kedua tahap tersebut dapat dipisahkan dan dikerjakan sendiri-sendiri (lihat gambar 3.1).



Gambar 3.1

Dalam tahap pemecahan masalah, pembahasan dititik beratkan pada rancangan algoritma dari persoalan yang akan diselesaikan. Bila telah diperoleh suatu algoritma, maka dapat dilakukan pengetrapan pemecahan masalah dari bentuk algoritma kedalam bahasa pemrograman.

Sebuah algoritma dapat didefinisikan sebagai suatu rangkaian langkah-langkah terurut untuk memecahkan suatu persoalan tertentu.

Untuk membuat suatu algoritma ada beberapa sifat yang penting dan harus diperhatikan.

Sifat-sifat tersebut adalah :

Pertama, langkah-langkah yang ada pada algoritma harus diberikan dengan jelas.

Kedua, pembuatan algoritma harus efektif, yang berarti algoritma tersebut harus memecahkan masalah yang ada dalam sejumlah langkah-langkah yang terbatas.

Suatu algoritma dibuat untuk manusia, sedangkan program dibuat untuk komputer.

Suatu algoritma dapat dituliskan dalam dua cara, yaitu :

1. Dituliskan dengan menggunakan kata-kata.
2. Dituliskan dengan simbol-simbol tertentu dan kata-kata.

Penulisan dengan cara ini dinamakan penulisan flowchart (diagram alur).

Setelah suatu algoritma dibuat, katakan dalam deretan langkah-langkah, algoritma tersebut akan diterapkan pada bahasa pemrograman yang dikehendaki.

Selanjutnya untuk menentukan algoritma yang dibuat itu betul atau salah, dapat dilakukan dengan memeriksa hasil yang diperoleh dari program yang diberikan kepada komputer. Bila hasilnya sesuai dengan hasil yang diharapkan, maka algoritma tersebut benar.

3.1.1 FORMAT UMUM PENULISAN ALGORITHMMA DENGAN KATA-KATA

Untuk menuliskan sebuah algoritma dengan kata-kata digunakan format sebagai berikut :

ALGORITMA NAMA , perincian dari keseluruhan langkah-langkah yang akan dikerjakan.

Langkah 0. [.....] Pernyataan

Langkah 1. [.....] Pernyataan

Langkah 2. [.....] Pernyataan

Langkah K. [.....] Pernyataan

Langkah K+1. [.....] Pernyataan

Langkah N. [.....] Pernyataan

KETERANGAN

a) NOMOR LANGKAH

Nomor langkah pada suatu algoritma digunakan untuk menunjukkan urutan langkah-langkah yang harus dikerjakan. Nomor langkah dituliskan dengan "langkah i", dengan i bulat non negatif dan dituliskan secara berurutan dari 0. Untuk menuliskan nomor langkah dimulai dari kiri. Nomor langkah yang dimulai satu atau dua langkah dari tepi kiri, digunakan untuk membedakan tingkat dari logikanya.

b) TANDA KURUNG / [.....]

Tanda kurung [.....] dituliskan setelah "langkah i", berfungsi untuk menyatakan dengan singkat dan jelas tujuan dari "langkah i".

c) PERNYATAAN

Setelah tanda kurung [....], selanjutnya dituliskan pernyataan-pernyataan yang berjumlah satu atau lebih. Pernyataan yang satu dengan yang lain dipisahkan titik koma dan berakhir dengan titik.

Dalam hal ini pernyataan diartikan sebagai perintah yang akan dikerjakan komputer.

3.2 BILANGAN ACAK

Pandang bilangan bulat r_1, r_2, r_3, \dots , yang diperoleh dari suatu pemilihan yang dilakukan secara acak dengan pengembalian dari himpunan bilangan bulat $\{1, 2, 3, 4, \dots, N\}$. Jika $N=2$ pemilihannya dapat dilakukan dengan melemparkan sebuah mata uang yang seimbang. Jika $N=6$ pemilihannya dapat dilakukan dengan melemparkan sebuah dadu yang seimbang. Dalam hal ini r_1 atau r_2 atau r_3 dinamakan bilangan acak.

Oleh karena terpilihnya suatu bilangan acak pada suatu pemilihan tidak dipengaruhi oleh terpilihnya bilangan acak pada pemilihan sebelumnya, maka kejadian-kejadian yang menghasilkan bilangan acak mempunyai hubungan independent (bebas) (sesuai 2.1.2).

3.2.1 PEMBUATAN BILANGAN ACAK DENGAN KOMPUTER

Untuk menghasilkan bilangan acak dengan komputer, dapat dilakukan dengan menggunakan suatu fungsi yang bernama

RND (Random).

Fungsi RND (Random) adalah fungsi dalam bahasa BASIC

yang digunakan untuk membuat bilangan acak.

RND menghasilkan sebuah bilangan acak yang bernilai antara 0 dan 1. Untuk menghasilkan bilangan acak, RND ditulis dalam suatu program. Dengan menuliskan RND berarti memberi intruksi pada komputer untuk melakukan suatu perhitungan agar diperoleh sebuah bilangan acak yang bernilai antara 0 dan 1. Setiap kali RND dijumpai dalam program, suatu bilangan yang berlainan akan dihasilkan. Bilangan ini akan menjadi bilangan berikutnya. Akan tetapi setiap kali program dijalankan, sekumpulan bilangan yang bernilai antara 0 dan 1 akan dihasilkan dengan urutan yang tetap. Agar diperoleh berbeda, dapat digunakan pernyataan RANDOMIZE yang ditulis sebelum pernyataan RND. Dengan RANDOMIZE, komputer meminta suatu bilangan yang dapat berharga dari -32768 s/d 32767.

Bilangan-bilangan tersebut dinamakan benih (Random number seed) bagi fungsi RND (Random).

Benih dimaksudkan sebagai suatu harga untuk memulai perhitungan dalam menghasilkan bilangan acak.

Memberikan suatu bilangan yang dapat berharga dari -32768 s/d 32767 berarti memberi intruksi kepada komputer agar melakukan suatu perhitungan untuk menghasilkan bilangan acak dengan menggunakan harga yang diberikan.

Sebagai contoh, akan diberikan suatu program yang dapat menghasilkan sejumlah bilangan acak yang bernilai antara 0 dan 1, dengan dan tanpa menggunakan RANDOMIZE.

```

10 FOR N=1 TO 5
20     PRINT RND;
30 NEXT N
40 END
RUN
.7151002 .683111 .4821425 .9992938 .6465093
RUN
.7151002 .683111 .4821425 .9992938 .6465093

```

```

10 RANDOMIZE
20 FOR N =1 TO 5
30     PRINT RND;
40 NEXT N
50 END
RUN
Random number seed (-32768 to 32767)? 357
.7751215 .2061109 .839107 .1153508 .7257672
RUN
Random number seed (-32768 to 32767)? 809
.5793818 .7698927 .1594805 .1776677 .5388165

```

Seperti telah diketahui, RND menghasilkan bilangan acak yang bernilai antara 0 dan 1. Untuk menghasilkan bilangan acak bulat yang dapat berharga dari A s/d B, pernyataan RND dapat diganti dengan $INT((B-A+1)*RND) + A$. Sebagai gambaran, akan diberikan contoh program yang menghasilkan sejumlah bilangan acak bulat yang dapat berharga dari 1 s/d 40. Dalam hal ini dibedakan dengan dan tanpa pernyataan RANDOMIZE.

```

5 RANDOMIZE
10 A=1 : B=40
20 FOR I=1 TO 10
30     X=INT((B-A+1)*RND)+A
40     PRINT X;
50 NEXT I
60 END
RUN
Random number seed (-32768 to 32767)? 320
 11  31  8  2  39  36  2  31  1  40
RUN
Random number seed (-32768 to 32767)? 2
 27  22 35  3  35  7  8  13 17  31

```

```

10 A=1 : B=40
20 FOR I=1 TO 10
30     X=INT((B-A+1)*RND)+A
40     PRINT X;
50 NEXT I
60 END

```

RUN									
28	28	20	40	26	6	15	24	6	38
RUN									
29	28	20	40	26	6	15	24	6	38

Bilangan acak yang dihasilkan fungsi RND dengan dan tanpa pernyataan RANDOMIZE ini, akan digunakan untuk pemilihan sampel yang akan dibicarakan pada BAB IV.

