BAB II

MATERI PENUNJANG

2.1. Permutasi.

Definisi 1 :

Suatu permutasi r unsur, yang diambil dari n unsur yang berlainan adalah penempatan r unsur itu dalam satu urutan ($r \le n$).

Dan ditulis dalam rumus sebagai berikut :

$$ {}_nP_r = n\,(n-1)\,(n-2)\ldots(n-r+1). $$

$$ = \frac{n\,!}{(n-r)!} $$

$$ {}_nP_n = n\,(n-1)\,(n-2)\ldots\ldots1 $$

$$ = n! $$

Bukti untuk rumus diatas adalah sebagai berikut :

Ada n unsur dan diambil r unsur dari n unsur tersebut, maka

pada kotak ke 1 ada n pilihan

pada kotak ke 2 ada (n – 1) pilihan

pada kotak ke r ada (n + 1 – r) pilihan

Karena untuk setiap pilihan digabungkan maka ada $n(n-1)(n-2)\ldots\ldots\ldots(n+1-r)$ permutasi samadengan

$$n(n-1)(n-2)\ldots\ldots(n+1-r)\cdot\frac{(n-r)(n-r-1)\ldots\ldots 1}{(n-r)(n-r-1)\ldots\ldots 1}$$

$$= \frac{n\,!}{(n-r)\,!} \quad \text{terbukti}$$

Untuk r = n akan didapat

$$_nP_n = \frac{n\,!}{(n-n)\,!}$$

$$= \frac{n\,!}{0\,!} = \frac{n\,!}{1} = n\,! \quad \text{terbukti}$$

Contoh 1 :

Terdapat tiga unsur a,b,c tentukan permutasi dan banyaknya permutasi tiga dari tiga unsur dan dua dari tiga unsur.

Jawab : $_3P_3 = 3\,!$

$= 3.2.1 = 6$

$_3P_2 = \frac{3\,!}{(3-2)\,!}$

$= 3.2.1 = 6$

Untuk tiga dari tiga unsur.

(a,b,c), (a,c,b), (b,a,c),

(b,c,a), (c,a,b), (c,b,a).

Untuk dua dari tiga unsur.

(a,b), (a,c), (b,c), (b,a), (c,a), (c,b).

2.2. Fungsi MOD.

Fungsi ini digunakan untuk menghitung sisa pembagian dari dua buah operand.

Bentuk umum :

```
Arg 1 MOD Arg 2
```

dengan Arg 1 dan Arg 2 adalah operand-operand yang akan dioperasikan.

Contoh :

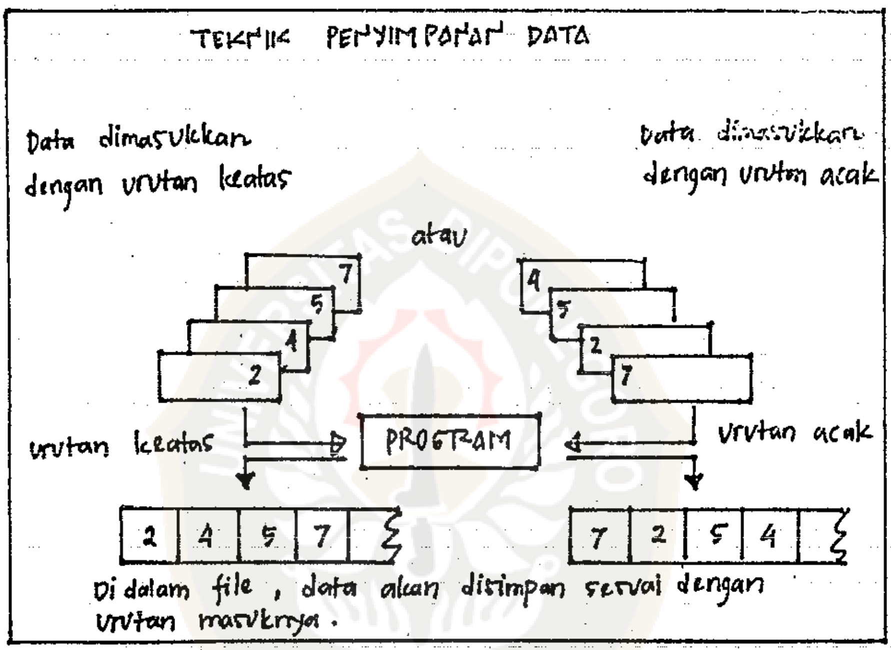
```
10  x% = 15  :  y% = 4
20  Sisa % = x% MOD y%
30  PRINT "Sisa pembagian dari "; x%;" Dibagi  ";  y%
    ;" = "; Sisa %

Run
Sisa pembagian dari 15 dibagi 4 = 3
```

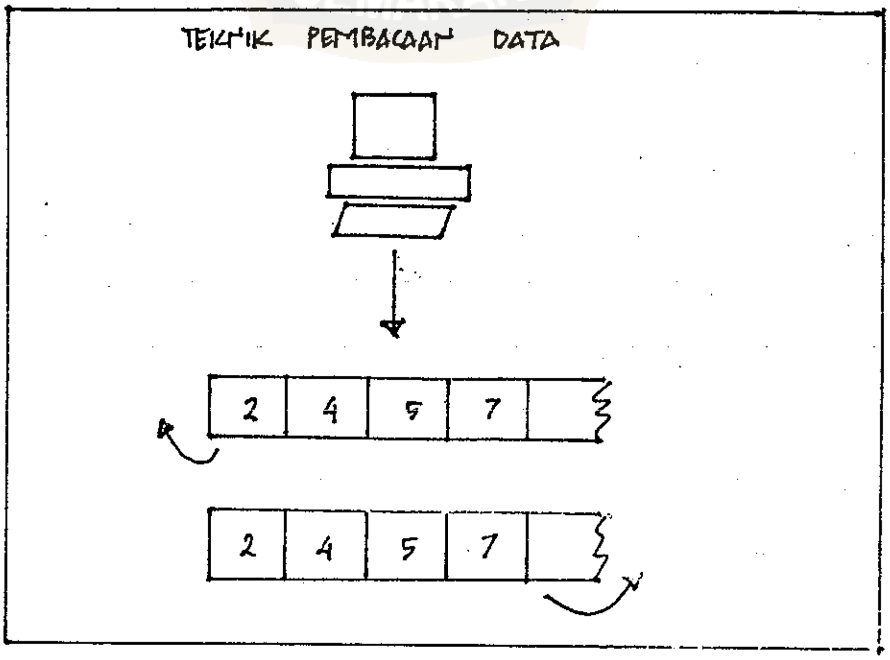
2.3. Teknik pembacaan data pada file data .

File data dibagi dua yaitu :

1 .File data yang berurutan.

Pada file data yang berurutan, teknik pembacaan datanya adalah bahwa data yang pertama masuk akan disimpan pada record pertama, data kedua pada record kedua, dan pada data terakhir akan disimpan pula pada record yang terakhir, karena pada file data yang berurutan tidak mengenal adanya nomor kunci record. Sehingga dapat dikatakan bahwa cara penyimpanan data yang cara penyimpanan data yang dilakukan, akan berdasar urutan masuk dari data yang bersangkutan. Untuk lebih jelasnya lihat gambar mengenai teknik penyimpanan dan pembacaan pada file data yang berurutan.



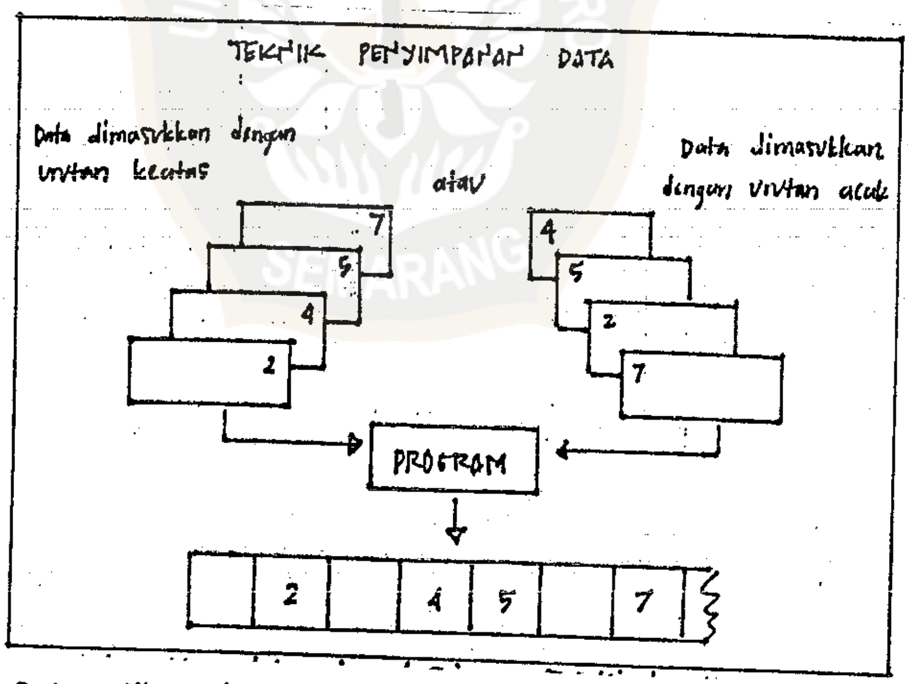


Yang pertama data akan dibaca satu demi satu dengan dimulai data pada record pertama (FIFO). Yang kedua data akan dibaca satu demi satu dengan dimulai data pada record terakhir (LIFO).

2. File data yang acak.

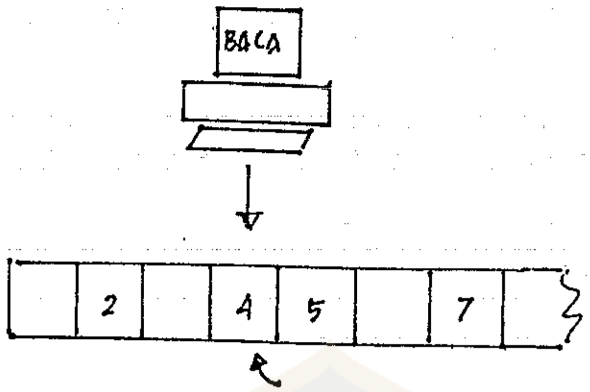
Pada file data secara acak mengenal adanya nomor kunci record, maka data yang tersimpan pada record nomor tujuh (dan sampai hal ini, record nomor satu sampai dengan nomor enam akan kosong), kemudian data kedua dengan nomor kunci record dua,akan ditempatkan pada record yang kedua dan seterusnya. Sehingga dengan demikian, seandainya hanya punya satu record saja dengan nomor kunci record 1000 dan ingin agar komputer melakukan pembacaan data dengan teknik yang berurutan, maka komputer juga harus membaca isi dari record-record sebelumnya dan sesudahnya, dikatakan sesudahnya, artinya komputer akan tetap melakukan pembacaan pula pada record-record sesudah record yang ke 1000 tersebut . Karena perintah yang diterima oleh komputer adalah : BACA SELURUH DATA YANG ADA DIDALAM FILE , akibatnya pembacaan yang dilakukan oleh komputer, tidak mungkin bisa berhenti hanya sampai pada record yang ke

1000 saja. Komputer akan terus bekerja, sehingga semua record yang ada didalam file akan habis terbaca, maka pembacaan record memakan waktu yang sangat lama. Seperti diketahui, pada file data yang acak ini bisa dilakukan pembacaan dengan dua cara, yaitu pembacaan sacara langsung, ataupun secara berurutan. Untuk lebih jelasnya lihat gambar mengenai teknik penyimpanan dan pembacaan pada file data yang acak.



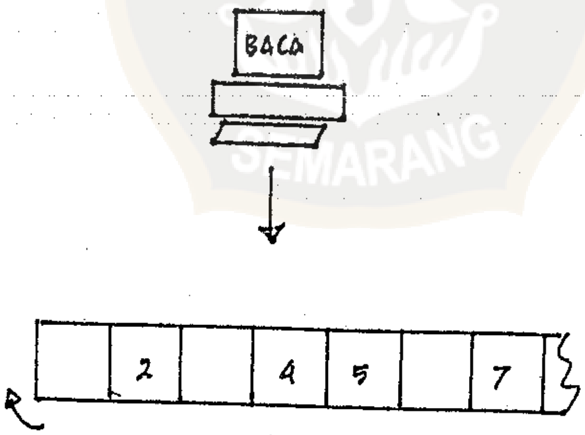
Dalam file, data akan tersimpan sesuai dengan nomor kunci record.

TEKNIK PEMBACAAN DATA SECARA ACAK.



Data akan dibaca secara langsung berdasar nomor kunci record.

TEKNIK PEMBACAAN DATA SECARA BERURUTAN.



Data akan dibaca secara urut dengan dimulai data pada record pertama.

2.4. Barisan.

Definisi 1 :

Barisan adalah kumpulan bilangan $a_1, a_2, \ldots.$ dalam urutan tertentu dan yang dibentuk menurut suatu aturan. Setiap bilangan dalam berisan disebut suku ; a_n disebut suku ke n. Jika banyaknya suku berhingga, maka berisan disebut barisan hingga dan jika banyaknya suku tak hingga disebut barisan tak hingga. $a_1, a_2, a_3, \ldots.$ ditulis secara pendek $\{a_n\}$

Contoh 1 :

$3, 7, 11, 15, \ldots.$ adalah barisan tak hingga. suku ke n ditentuksn $a_n = 4n - 1$, $n = 1,2,3,\ldots$

Definisi 2 :

Kumpulan titik bilangan nyata dikatakan terbatas, jika untuk semua titik x dalam kumpulan itu, ada konstanta a dan b sehingga $a \leq x \leq b$

Barisan terbatas.

Barisan $\{a_n\}$ dikatakan terbatas disebelah atas jika ada konstanta A sehingga berlaku $\{a_n\} \leq A$ untuk $n = 1,2,\ldots$ A disebut suatu batas atas.

Jika semua $\{a_n\} \geq B$ dikatakan barisan $\{a_n\}$

terbatas disebelah bawah. B disebut suatu batas bawah.

Jika $B \le a_n \le A$, barisan dikatakan terbatas.

Barisan Monoton.

Barisan dikatakan naik monoton, jika berlaku $a_{n+1} \ge a_n$ untuk $n = 1,2,\ldots..$

jika berlaku $a_{n+1} > a_n$ $(n = 1,2,\ldots)$, barisan disebut naik monoton murni.

Barisan $\{a_n\}$ disebut turun monoton jika $a_{n+1} \le a_n$ $(n = 1,2,3,\ldots)$

Contoh 2 :

1. Barisan $2, 1\frac{1}{2}, 1\frac{1}{3}, 1\frac{1}{4}, \ldots$adalah terbatas atas ,monoton turun, barisan juga turun monoton murni.
2. $\{(-1)^n\}$ adalah terbatas, tetapi tak turun monoton dan juga tak naik monoton.

Teorema :

Barisan yang naik monoton dan terbatas di sebelah atas mempunyai limit.

Sebelum bukti diberikan, perhatikan axioma yang berikut. KUmpulan bilangan nyata yang tak kosong dan terbatas disebelah atas, selalu mempunyai batas atas terkecil. Demikian pula kumpulan bilangan nyata yang tak kosong dan terbatas di sebelah bawah, mempunyai batas bawah terbesar.

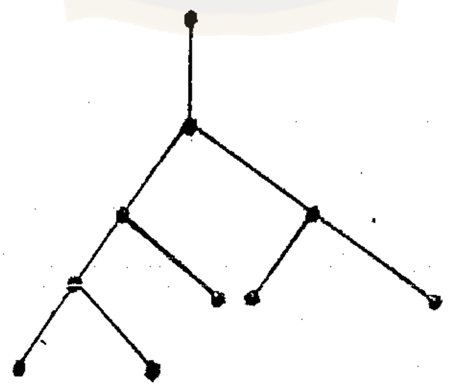
Bukti :

Karena $\{ a_n \}$ terbatas disebelah atas, maka ada batas atas terkecil. Sebut saja L. Ambillah>0, maka $L - \varepsilon$ bukanlah batas atas. Oleh karena itu ada a_N (untuk suatu N) sehingga $a_N > L - \varepsilon$ Karena barisan monoton naik $a_N, a_{N+1}, \ldots$. semua lebih besar daripada $L - \varepsilon$ dan semua $a_n \leq L$. Jadi $L - \varepsilon < a_n \leq L$ untuk $n \geq N$ atau $L - \varepsilon < a_n < L + \varepsilon$ untuk $n \geq N$. Dengan demikian L merupakan limit barisan itu.

2.5. Tree

Definisi 28 :

Sebuah pohon (tree) adalah graph terhubung (connected garph) tanpa sirkuit.



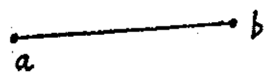
Teorema 4.:

Ada satu dan hanya satu path diantara titik-titik dalam sebuah pohon T.

Bukti :

Jika T adalah graph terhubung, maka harus ada sedikitnya satu path diantarapasangan titik-titik dalam T. Andaikan diantara dua titik a dan b pada T ada dua path yang berbeda, union dari dua path tersebut membentuk suatu sirkuit dan menurut Definisi 28. maka T bukan merupakan pohon. Pengandaian salah yang benar adalah ada satu dan hanya satu path antara pasangan titik-titik dalam T.

Contoh : Dimisalkan ada dua titik a dan b maka menurut teorema 4 harus ada satu dan hanya satu path diantara titik a dan b.



Teorema 5 :

Jika dalam Graph G ada satu dan hanya satu path diantara pasangan titik-titiknya , maka G adalah pohon (tree)

Bukti :

Andaikan dalam graph G ada sebuah sirkuit maka ada sedikitnya satu pasang titik a dan b sedemikian hingga ada dua path yang berbeda antara a dan b. Kontradiksi dengan G mempunyai satu dan hanya satu path diantara pasangan titik-titiknya. Pengandaian salah, yang benar adalah G tidak mempunyai sirkuit sehingga menurut Definisi 28 G adalah pohon (tree).

Teorema 6 :

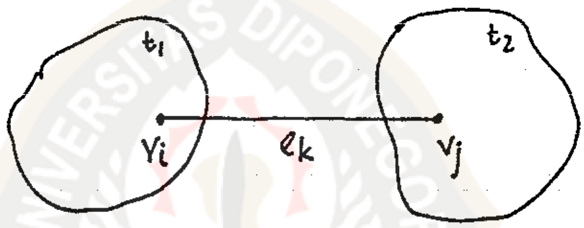
Sebuah pohon dengan n titik mempunyai n-1 garis.

Bukti :

Teorema akan dibuktikan dengan menginduksi jumlah titiknya. jelas terlihat bahwa teorema benar untuk n = 1,2,3,.... Sekarang diandaikan teorema berlaku untuk seluruh pohon dengan jumlah titik n - 1 buah. akan dibuktikan teorema benar untuk jumlah titik n buah.

Pandang sebuah pohon T dengan n titik, e_k adalah sebuah garis graph dengan titik akhir v_i dan v_j. Sesuai dengan teorema 4, maka tidak ada path lain diantara v_i dan v_j

kecuali e_k. Karena itu penghapusan e_k dari T akan memutus pohon ini, sehingga T-e_k terdiri dari tepat dua komponen. Kedua pohon itu t_1 dan t_2 mempunyai titik lebih sedikit dari n. Ambil t_1 terdiri dari n − 1 titik dan t_2 terdiri dari satu titik, sehingga T-e_k memuat n − 2 garis graph (untuk n titik). Dari sini T tepat mempunayi n − 1 garis graf.



Teorema 7 :

Sebuah graph dengan n titik,n − 1 garis graph dan tidak punya sirkuit adalah terhubung.

Bukti :

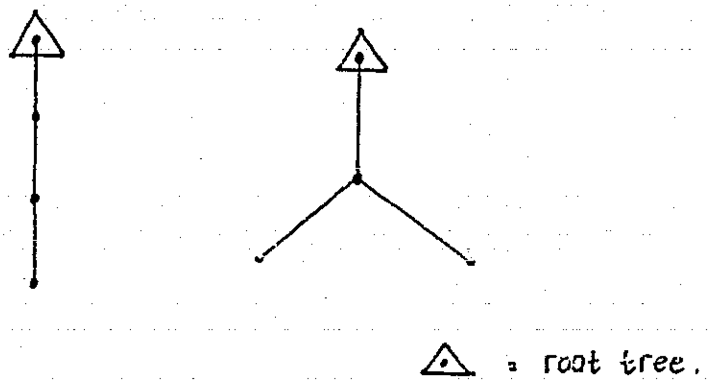
Andaikan ada graph G tanpa sirkuit yang mempunyai n titik dan n − 1 garis graph yang tak terhubung. Dalam hal ini G akan memuat dua atau lebih komponen bukan sirkuit. Ambil G memuat dua komponen g_1 (V_1,E_1) dan g_2 (V_2, E_2), tambahkan sebuah garis graph e diantara V_1 dan V_2 pada G, maka penambahan e tidak akan menghasilkan sirkuit , sehingga G U e adalah graph terhubung yang tidak mempunyai sirkuit. Menurut Definisi 28 G U e adalah pohon dengan n titik dan n garis graph. Kontradiksi dengan teorema 6 yang menyatakan bahwa suatu pohon dengan n titik mempunyai n - 1 garis graph. Jadi pengandaian salah , yang benar adalah tidak ada graph G tanpa sirkuit yang mempunyai n titik dan n - 1 garis graph yang tidak terhubung.

Akibat dari teorema 4 sampai dengan 7, dapat disimpulkan 5 Definisi pohon yang berbeda tapi equivalen. Suatu graph dengan n titik dikatakan pohon jika :

1. G adalah graph terhubung dan tidak mempunyai sirkuit atau.
2. G adalah graph terhubung dan mempunyai n - 1 garis graph atau.
3. G adalah graph yang tidak mempunyai sirkuit dan mempunyai n - 1 garis graph atau.
4. Ada tepat satu path diantara pasangan titik-titik dalam graph G atau.
5. G adalah graph terhubung minimal.

Dalam tree, ada satu titik yang berbeda fungsinya dengan titik yang lain, yang disebut root tree. Dari root tree inilah semua pengiriman dikeluarkan (dikirim).

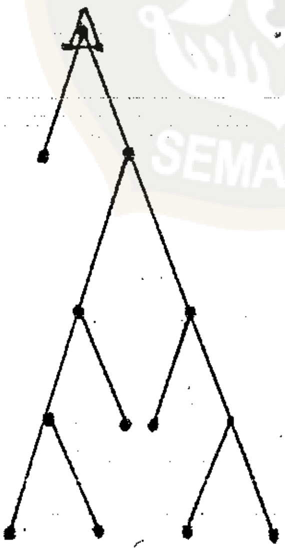
Contoh :



Binary root tree adalah root tree khusus dan tree khusus adalah binary tree dimana setiap binary tree hanya mempunyai dua anak (dua pendant vertices).

Untuk lebih jelasnya binary tree adalah tree dengan tiga atau lebih titik.

Contoh :



Gambar 3.12.

Dari gambar terlihat bahwa gambar 3.12. terdiri dari tujuh pendant vertices , dan terdiri dari tiga belas titik serta satu titik sebagai binary root tree yaitu titik yang paling atas.