

Pertemuan 12

Waktu : 135 menit

Tujuan Pembelajaran : Mahasiswa mampu menjelaskan teknik pemrograman menggunakan Sorting.

Substansi Materi : Quick Sort

Tabulasi Kegiatan Perkuliahan

No	Tahap Kegiatan	Kegiatan Pengajar	Kegiatan Mahasiswa	Media & Alat	Waktu
1	Pendahuluan	1. Membuka pertemuan 2. Mengulang materi pertemuan sebelumnya	Menyimak Bertanya	Papan Tulis	20 Menit
2	Penyajian Materi	1. Quick Sort 2. Latihan Soal	Menyimak Bertanya Menjawab Pertanyaan	Papan Tulis	80 Menit
3	Penutup	1. Menyimpulkan materi pertemuan 2. Memberikan tugas kecil 3. Menutup pertemuan	Menyimak	Papan tulis	35 Menit

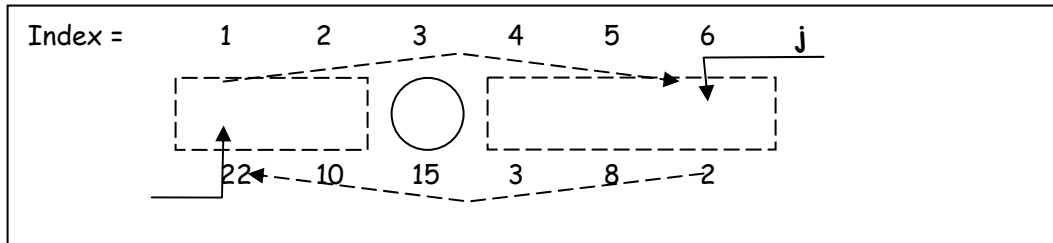
MATERI KULIAH

Quick Sort

Merupakan membandingkan suatu elemen (disebut juga pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lainnya yang lebih kecil daripada pivot tersebut terletak disebelah kirinya dan elemen-elemen lain yang lebih besar daripada pivot terletak disebelah kanannya. Dengan demikian telah terbentuk dua sublist, yang terletak di sebelah kiri dan kanan dari pivot. Lalu pada sublist kiri dan sublist kanan anggap sebuah list baru dan kerjakan proses yang sama seperti sebelumnya. Demikian seterusnya sampai tidak terdapat sublist lagi. Sehingga didalamnya terjadi sebuah proses rekursif.

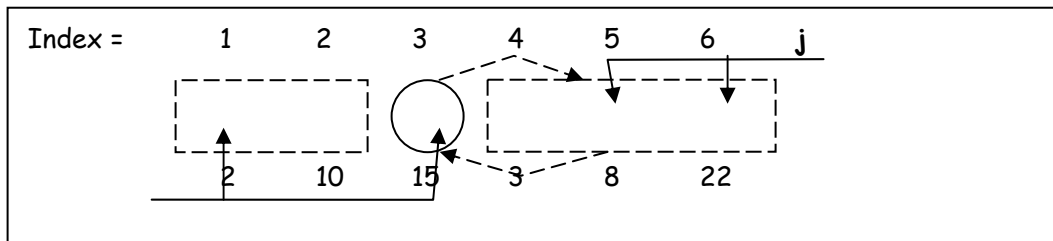
Proses :

1. Bilangan yang didalam kurung merupakan pivot
2. Persegi panjang yang digambarkan dengan garis terputus2 menunjukan sublist
3. i bergerak dari sudut kiri ke kanan sampai mendapatkan nilai yang \geq pivot
4. j bergerak dari sudut kanan ke kiri sampai menemukan nilai yang $<$ pivot



5. i berhenti pada index ke-1 karena langsung mendapatkan nilai yang $>$ dari pivot(15).
6. j berhenti pada index ke-6 karena juga langsung mendapatkan nilai yang $<$ dari pivot.
7. Karena $i < j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk oleh j sehingga urutan menjadi :

2 10 15 3 8 22



8. i berhenti pada index ke-3 (pivot) karena tidak menemukan bilangan yang > dari pivot.
9. j berhenti pada index ke-5 menunjuk pada nilai yang < dari pivot.
10. karena $i < j$ maka data yang ditunjuk oleh I (pivot) ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 10 8 3 15 22

11. Proses yang sama seperti sebelumnya dilakukan terhadap 2 buah sublist yang baru.
Sehingga menjadi :

2 3 8 10 15 22

Contoh prosedur :

“Procedure Quick Sort dengan nilai kiri sebagai pembanding (pivot)”

Procedure Asc_quick(L,R : integer); { Prosedur ascending }

Var

```

i, j : integer;
begin
  if L < R then
    begin
      i := L ; j := R+1;
      repeat
        repeat inc(i) until data[ i ] >= data [ 1 ];
        repeat dec(j) until data[ j ] <= data [ 1 ];
        if i < j then TukarData (data[i], data[j]);
      until i > j;
      TukarData(data[ 1 ], data [ j ]);
      Asc_quick(L, j-1);
    end
  end

```

```

        Asc_quick(j+1, R);
    End;
End;
```

Procedure Desc_quick(L,R : integer); { Prosedur Descending }

Var

```

    i, j : integer;
begin
    if L < R then
        begin
            i := L ; j := R+1;
            repeat
                repeat inc(i) until data[ i ] <= data [ l ];
                repeat dec(j) until data[ j ] >= data [ l ];
                if i < j then TukarData (data[i], data[j]);
            until i > j;
            TukarData(data[ l ], data [ j ]);
            Asc_quick(L, j-1);
            Asc_quick(j+1, R);
        End;
    End;
```

“Procedure Quick Sort dengan nilai tengah sebagai pembanding (pivot)”

Procedure asc_Quick(L,R : integer); { Prosedur Ascending }

Var

Mid, i, j : integer;

Begin

i := L ; j := R ; mid := data [(L+R) div 2] ;

repeat

 while data [i] < mid do inc(i);

 while data [j] > mid do dec(j);

 if i <= j then

 begin

 Change(data[i], data[j]);

 Inc(i); dec(j);

 End;

Until i > j;

If L < j then Asc_Quick(L, j);

If i < R then Asc_Quick(i, R);

End;

Procedure desc_Quick(L,R : integer); { Prosedur Descending}

Var

Mid, i, j : integer;

Begin

i := L ; j := R ; mid := data [(L+R) div 2] ;

repeat

 while data [i] > mid do inc(i);

```
while data [ j ] < mid do dec(j);  
if i <= j then  
  begin  
    Change(data[ i ], data[ j ]);  
    Inc(i); dec(j);  
  End;  
Until i > j;  
If L < j then Asc_Quick( L, j );  
If i < R then Asc_Quick(i, R);  
End;
```