

Mata Kuliah : Dasar-dasar Pemrograman(Praaktek)
Kode Mata Kuliah : KBTI4104
Jumlah SKS : 4 SKS
Nama Dosen : Ani Rahmani
Minggu ke : 7
Tanggal : 30 Oktober 2015
Jadwal : Jumat (08.10 – 11.30)

Macam-macam Algoritma Sorting

Algoritma Sorting atau pengurutan digunakan untuk mengolah data dan mengurutkannya secara *ascending* atau *descending* sesuai kebutuhan. Algoritma Sorting ada banyak macamnya, tapi yang akan dibahas hanya beberapa, yaitu Bubble sort, Selection Sort, Insertion Sort, dan Quick Sort.

Contoh : Mengurutkan secara *Ascending*

A	4	3	6	3	10	23	5	7	8
---	---	---	---	---	----	----	---	---	---

a. Bubble Sort

Bubble sort dilakukan dengan membandingkan masing-masing item dalam suatu list secara berpasangan, menukar item dilakukan sampai akhir list secara berurutan.

6 5 3 1 8 7 2 4

Berikut contoh algoritma Bubble Sort dengan menggunakan data array A diatas

Algoritma

Begin

| i,j : integer

| for i $\leftarrow$ 1 to 8 do

```

|   |   for j  $\leftarrow$  i + 1 to 9 do
|   |       |   if (A[ i ] > A[ j ])
|   |       |       |   then A[ i ]  $\leftarrow$  A[ i ] + A[ j ]
|   |       |       |       A[ j ]  $\leftarrow$  A[ i ] - A[ j ]   // proses swaping
|   |       |       |       A[ i ]  $\leftarrow$  A[ i ] - A[ j ]
|   |       |   endif
|   |   endfor
|   endfor
End

```

Bahasa C

```

#include <stdio.h>

int main(){

    int i,j;

    for(i=0;i<8;i++){

        for(j=i+1;j<9;j++){

            if(A[i]>A[j]){

                A[i]=A[i]+A[j];

                A[j]=A[i]-A[j];

                A[i]=A[i]-A[j];

            }

        }

    }

    return 0;

}

```

b. Selection Sort

Cara kerja Selection Sort yaitu memilih elemen dengan nilai paling rendah dan menukar elemen yang terpilih dengan elemen ke-i. nilai dari i dimulai dari 1 ke n adalah jumlah total elemen dikurangi 1.

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7

Berikut contoh algoritma Selection Sort

Algoritma

Begin

```
| i,j,k,Min : integer
| for i  $\leftarrow$  1 to 8 do
|   | Min  $\leftarrow$  A[ i ]
|   | k  $\leftarrow$  i
|   | for j  $\leftarrow$  i to 9 do
|       | if (Min > A[ j ])
```

```

|   |   |   | then Min  $\leftarrow$  A[ j ]
|   |   |   |       k  $\leftarrow$  j
|   |   |   | endif
|   |   | endfor
|   |   | if (k  $\neq$  i )
|   |   |   | then A[ i ]  $\leftarrow$  A[ i ] + A[ k ]
|   |   |   |       A[ k ]  $\leftarrow$  A[ i ] - A[ k ]
|   |   |   |       A[ i ]  $\leftarrow$  A[ i ] - A[ k ]
|   |   |   | endif
|   |   | endfor
| End

```

Bahasa C

```
#include <stdio.h>
```

```
int main(){
```

```
    int i,j,k,Min;
```

```
    for(i=0;i<8;i++){
```

```
        Min=A[i];
```

```
        k=i;
```

```
        for(j=i;j<9;j++){
```

```
            if(Min>A[j]){
```

```
                Min=A[j];
```

```
                k=j;
```

```
            }
```

```
        }
```

```
        if(k!=i){
```

```
A[i]=A[i]+A[k];  
  
A[k]=A[i]-A[k];  
  
A[i]=A[i]-A[k];  
  
}  
  
}  
  
return 0;  
  
}
```

c. Insertion Sort

Cara kerja insertion sort sebagaimana namanya. Pertama-tama, dilakukan proses iterasi, dimana di setiap iterasi insertion sort memindahkan nilai elemen, kemudian menyisipkannya berulang-ulang sampai ketempat yang tepat. Begitu seterusnya dilakukan. Dari proses iterasi, seperti biasa, terbentuklah bagian yang telah di-sorting dan bagian yang belum.

6 5 3 1 8 7 2 4

Berikut contoh algoritma insertion sort.

Algoritma

Begin

| c,d : integer

| for c $\leftarrow$ 2 to 9 do

```

|   | d ← c
|   | while (d>1 AND A[d]<A[d-1]) do
|   |   | A[d] ← A[d]+ A[d-1]
|   |   | A[d-1] ← A[d] – A[d-1]
|   |   | A[d] ← A[d] – A[d-1]
|   |   | d ← d - 1
|   | endwhile
| endfor
end

```

Bahasa C

```

#include <stdio.h>

int main()
{
    int c, d;

    for (c = 1 ; c <= 8; c++) {

        d = c;

        while ( d > 0 && array[d] < array[d-1]) {

            A[d] =A[d]+a[d-1];

            A[d-1]=A[d]-A[d-1];

            A[d] =A[d]-A[d-1];

            d--;

        }

    }

    return 0;
}

```

```
}
```

d. Quick Sort

Quick sort merupakan metode pengurutan dengan algoritma berdasarkan pola divide-and-conquer.

Algoritma ini hanya memiliki 2 langkah sebagai berikut :

- Divide = bisa dikatakan Memilah rangkaian data menjadi dua sub-rangkaian $A[p...q-1]$ dan $A[q+1...r]$ dimana setiap elemen $A[p...q-1]$ adalah kurang dari atau sama dengan $A[q]$ dan setiap elemen pada $A[q+1...r]$ adalah lebih besar atau sama dengan elemen pada $A[q]$. $A[q]$ disebut sebagai elemen pivot. Perhitungan pada elemen q merupakan salah satu bagian dari prosedur pemisahan.
- Conquer = dengan cara Mengurutkan elemen pada sub-rangkaian secara rekursif. Pada algoritma quicksort, langkah "kombinasi" tidak di lakukan karena telah terjadi pengurutan elemen – elemen pada sub-array

berikut contoh quick sort :

6 5 3 1 8 7 2 4

Algoritma	Modul yang dipanggil
<u>Begin</u> quicksort(A,0,8) <u>end</u>	<u>Procedure</u> quicksort(A[:int,kiri:int,kanan:int) tmp,i,j,pivot : int $i \leftarrow \text{kiri}$ $j \leftarrow \text{kanan}$ $\text{pivot} \leftarrow A[(\text{kiri}+\text{kanan})\text{div } 2 + 1]$ <u>while</u> (i<=j) <u>while</u> (A[i + 1]<pivot)

	<pre> i ← i + 1 endwhile while (A[j+1]>pivot) j ← j - 1 endwhile if (i<=j) then tmp ← A[i + 1] A[i + 1] ← A[j + 1] A[j + 1] ← tmp i ← i + 1 j ← j - 1 endif endwhile if (kiri < j) then quicksort(A,kiri,j) endif if (i < kanan) then quicksort(A,i,kanan) endif end </pre>
Bahasa C #include <stdio.h> void quicksort(int*A,int kiri,int kanan);	<pre> void quicksort(int*A,int kiri,int kanan){ int tmp,i=kiri,j=kanan; int pivot=A[(kiri+kanan)/2]; while(i<=j){ </pre>

```
int main(){
```

```
    quicksort(A,0,8);
```

```
    return 0;
```

```
}
```

```
while(A[i]<pivot){
```

```
    i++;
```

```
}
```

```
while(A[j]>pivot){
```

```
    j--;
```

```
}
```

```
if(i<=j){
```

```
    tmp=A[i];
```

```
    A[i]=A[j];
```

```
    A[j]=tmp;
```

```
    i++;
```

```
    j--;
```

```
}
```

```
}
```

```
if(kiri<j){
```

```
    quicksort(A,0,8);
```

```
}
```

```
if(i<kanan){
```

```
    quicksort(A,i,kanan);
```

```
}
```

```
}
```