LAPORAN PRAKTIKUM STRUKTUR DATA



OLEH :

ALIFFIA HUMAIRAH

NIM : 2311531004

DOSEN PENGAMPU :

DR. WAHYUDI,S.T,MT.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

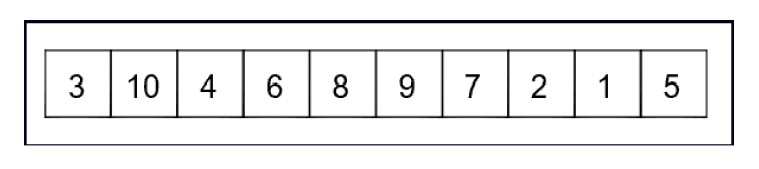
1. TUJUAN

Mahasiswa mampu memahami penggunaan insertion sort, selection sort, dan buble sort.

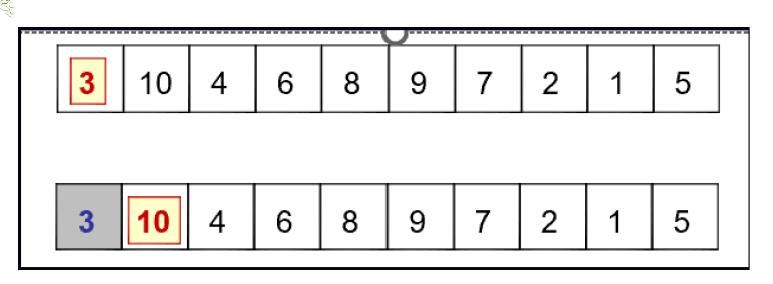
1. KAJIAN TEORI
2. **Insertion Sort**

Insertion sort atau metode penyisipan bertujuan untuk menjadikan bagian sisi kiri array terurutkan sampai dengan seluruh array berhasil diurutkan. Metode ini mengurutkan bilangan-bilangan yang telah dibaca dan berikutnya secara berulang akan menyisipkan bilangan-bilangan dalam array yang belum terbaca ke sisi kiri array yang telah terurut.

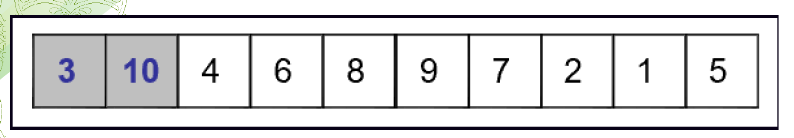
Contoh:



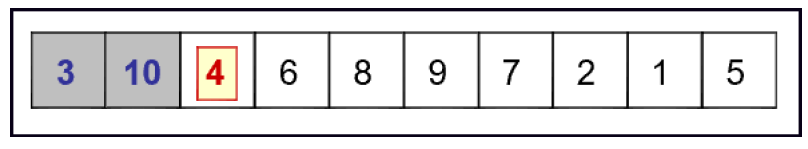
Bilangan paling kiri (3) bisa dikatakan telah terurut secara relatif terhadap dirinya sendiri.



Cek, untuk melihat apakah bilangan kedua (10) lebih kecil dari pada yang pertama (3). Jika ya, tukarkan kedua bilangan ini. Namun, kali ini kita tidak perlu melakukan penukaran.

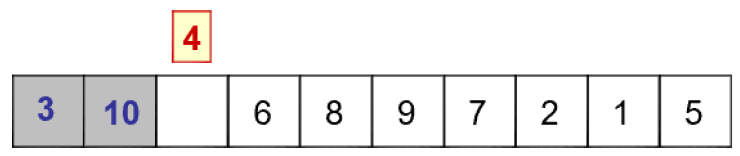


Bagian biru / abu-abu (dua bilangan pertama) sekarang dalam keadaan terurut secara relatif.

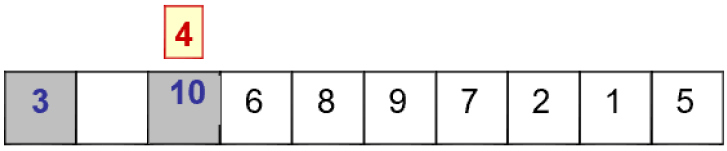


Berikutnya, kita perlu menyisipkan bilangan ketiga (4) ke dalam bagian biru / abu-abu tetap dalam keadaan terurut secara relatif.

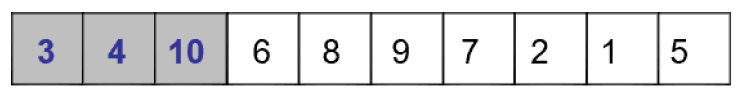
Pertama: ambil bilangan ketiga (4)



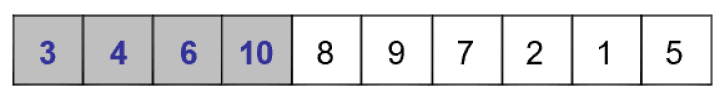
Kedua: geser bilangan kedua (10) sehingga ada ruang untuk disisipi.



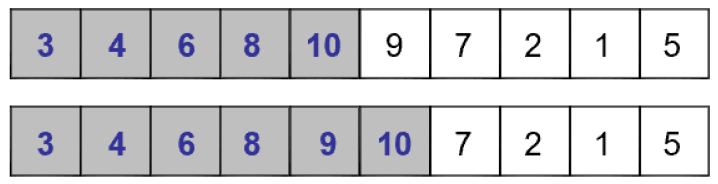
Ketiga: sisipkan bilangan 4 ke posisi yang tepat

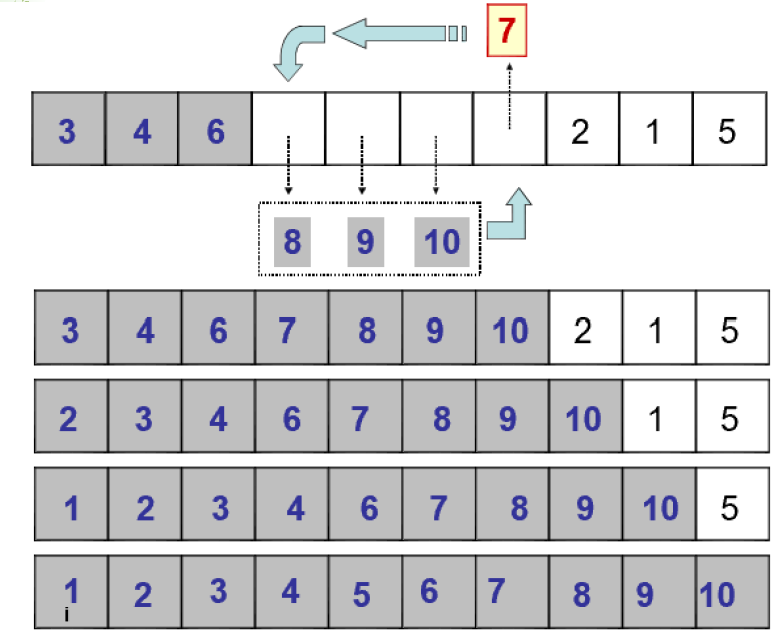


Sekarang, tiga bilangan pertama sudah terurut secara relatif dan kita sisipkan bilangan keempat kepada tiga bilangan pertama tersebut. Setelah penyisipan, empat bilangan pertama haruslah dalam keadaan terurut secara relatif.



Ulangi proses tersebut sampai bilangan terakhir disisipkan.

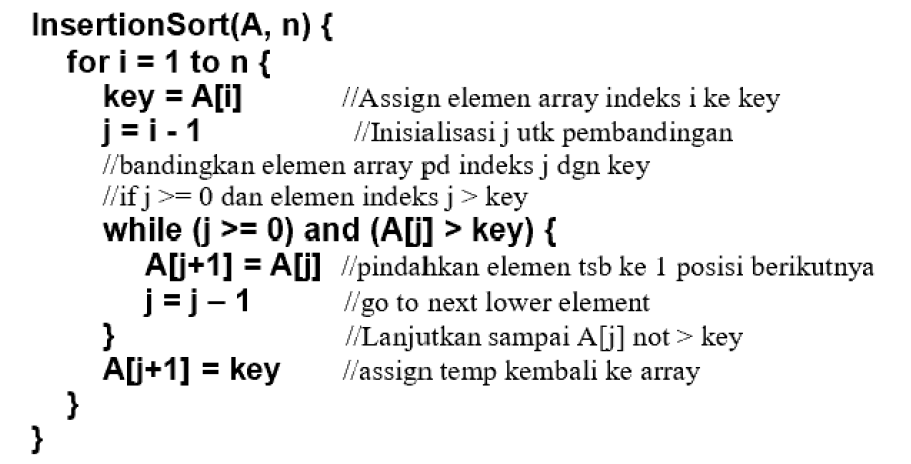




**ALGORITMA METODE PENYISIPAN**

1. i 1
2. Selama (I < n) kerjakan baris 3 sampai dengan 9
3. Key A[i]
4. J i – 1
5. Selama j >= 0 dan (A[j] > key)0 kerjakan baris 6 dan 7
6. A[j + 1] A[j]
7. J j – 1
8. A[j+1] key
9. i I +1

**PSEUDOCODE**



**INSERTION SORT ANALYSIS**

Running time bukan hanya bergantung pada ukuran array, namun juga pada susunan isinya

Best case:

* Array sudah dalam keadaan terurut naik
* Loop terdalam tidak pernah dieksekusi
* Jumlah pergeseran : 2(n-1)
* Jumlah pembandingan key (C) : (n-1)

Worst case:

* Array dalam urutan kebalikannya
* Loop terdalam dieksekusi debanyak p-1 kali, untuk p= 2,3,...,n
* Jumlah pergeseran:

2(n-1) + (1 + 2 +...+n-1) = 2(n-1) + n\*(n-1)/2

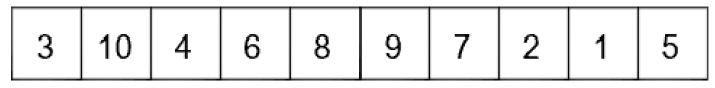
* Jumlah pembandingan key:

(1+2+...+n-1) = n\*(n-1)/2

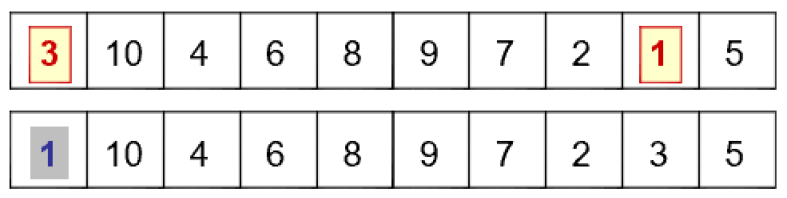
1. **Selection Sort**

Selection sort atau metode seleksi

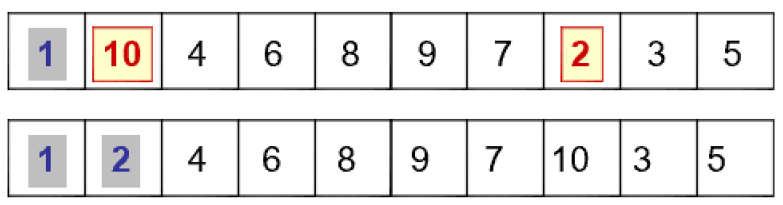
Contoh:

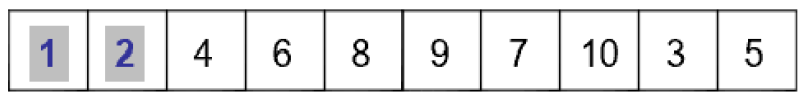


Cek seluruh elemen array, tentukan nilai terkecil (1) dan tukarkan posisinya dengan posisi nilai yang tersimpan pada posisi pertama dari array (3)



Temukan nilai terkecil kedua (2), dan tukarkan posisinya dengan nilai yang berada pada posisi kedua (10).





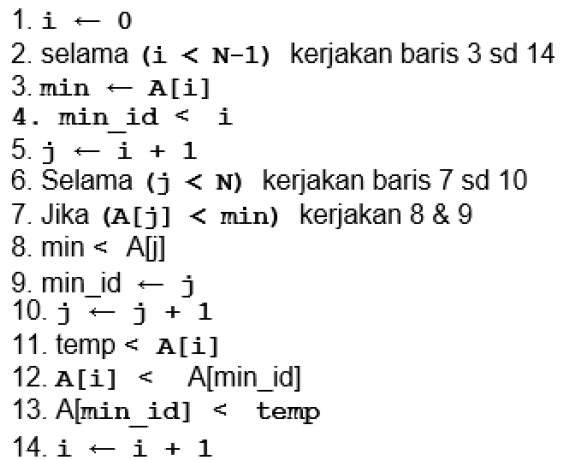
Dua elemen biru pertama tidak akan berubah lagi sebab mereka sudah merupakan nilai terkecil pertama dan kedua dalam array tersebut.

Sekarang, ulangi proses ”pilih dan tukar”

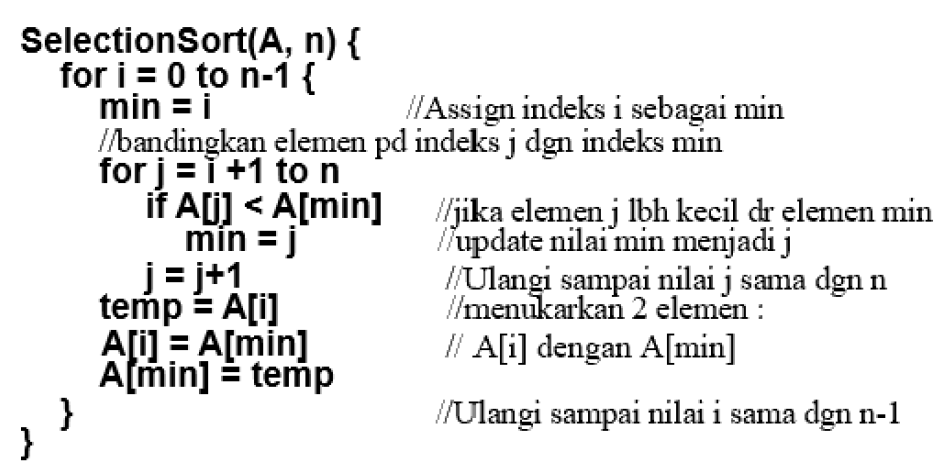




**ALGORITMA METODE SELEKSI**



**PSEUDOCODE**



**SELECTION SORT ANALYSIS**

Secara umum, yang dilakukan dalam metode seleksi adalah pembandingan key (elemen pada posisi min) serta penukaran elemen. Sehingga untuk menganalisis metode ini harus dihitung jumlah pembandingannya serta jumlah oenukaran elemennya.

Pada algoritma diatas, loop for terluar dilakukan sebanyak n-1 kali

Pada setiap iterasi, dilakukan satu kali penukaran elemen, sehingga:

* total penukaran/swap = n-1
* total pergeseran = 3\* n-1

(pada setiap penukaran terjadi 3x pergeseran)

Jumlah pembandingan pada metode ini adalah = 1+2+..+n-1 = n\*(n-1)/2

Dalam metode ini, jumlah pembandingan untuk best case dan worst casenya sama

Memindahkan dari kanan ke kiri, meletakkan elemen ke posisi finalnya tanpa merevisi lagi posisis tersebut

Menghabiskan sebagian besar waktu untuk mencari elemen terkecil pada sisi array yang belum terurut

1. **Buble Sort**

Buble sort atau meetode gelembung disebut dengan metode penukaran (exchange sort) adalah metode yang mengurutkan data dengan elem, kemudian melakukan penukaran bila perlu

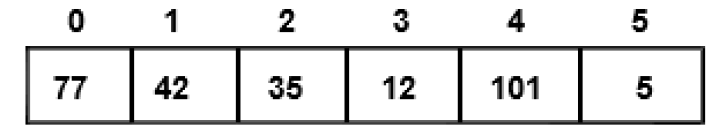
Metode ini mudah dipahami dan diprogram, tetapi bila dibandingkan dengan metode lain yang kita pelajari, metode ini merupakan metode yang paling tidak efisien.

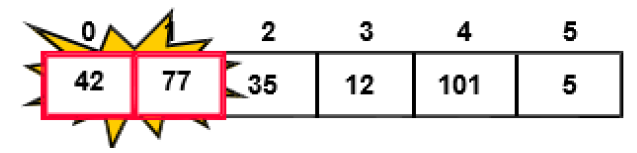
**”Bubbling Up”**

Mengecek sekumpulan elemen

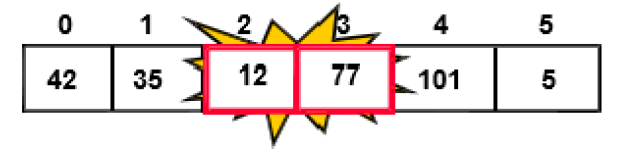
Memindahkannya dari posisi awal ke akhir

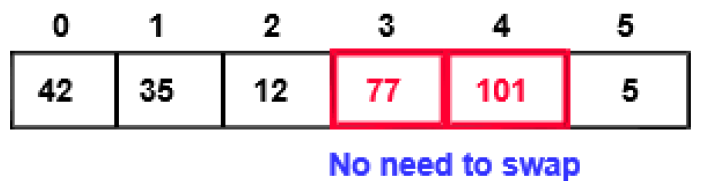
”menggelembungkan” nilai terbesar ke bagisn akhir menggunakan metode pembandingan sepasang dan penukaran.

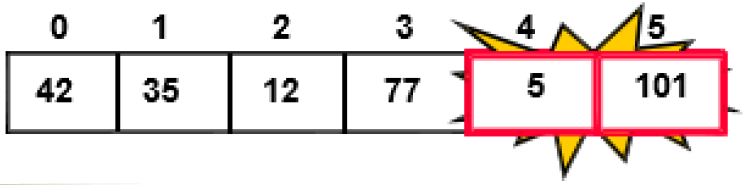








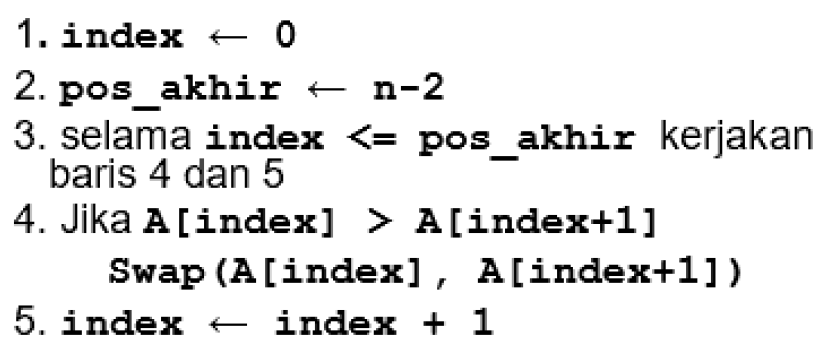




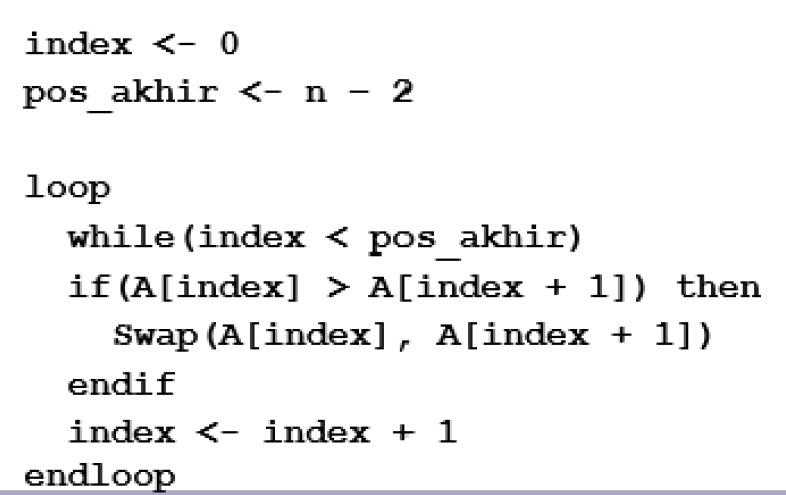


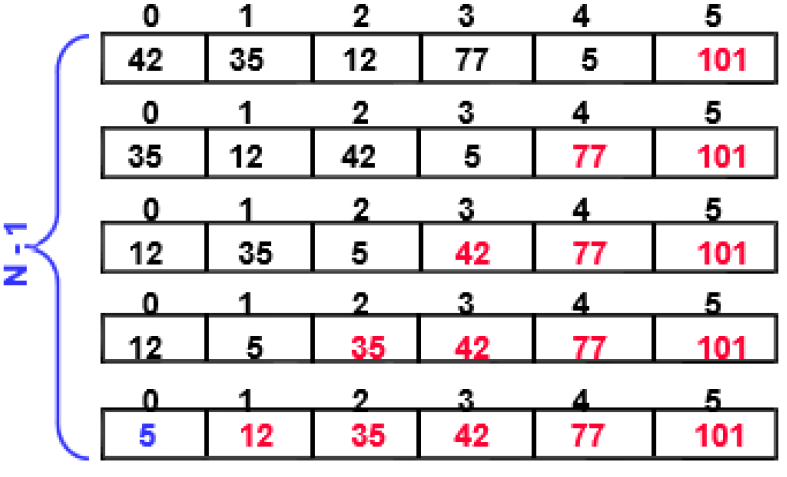
**ALGORITMA METODE BUBBLE SORT**

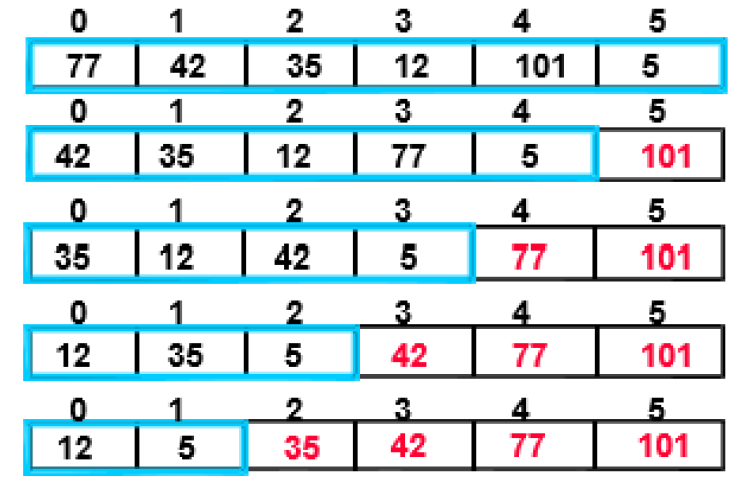
**Untuk satu kali iterasi**



**Versi 2**







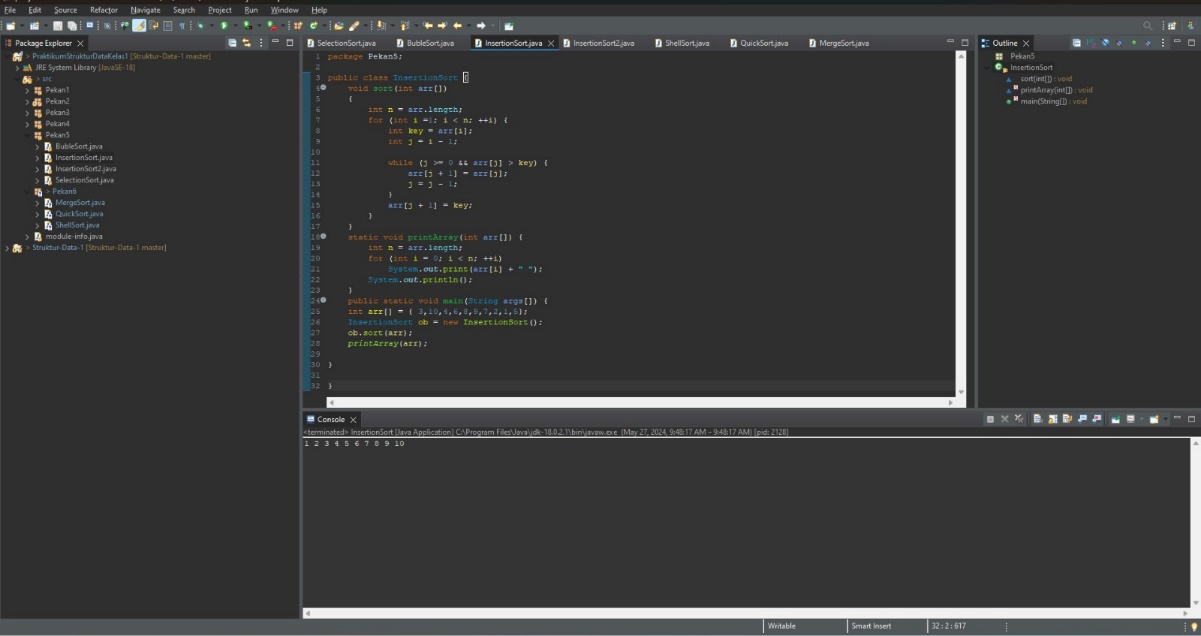
**BUBBLE SORT ANALYSIS**

* Best Case
* Array sudah dalam keadaan trurut naik
* Jumlah pembandingan key (C) : n-1
* Ju mlah swap = 0
* Ji+umlah pergeseran (M) : 0
* Worst Case
* Array dalam urutan kebalikannya
* Jumlah pembandingan key (C): (n-1) + (n-2) +…+1 = n\*(n-1)/2
* Jumlah swap = (n-1) + (n-2) +…+1 = n\*(n-1)/2
* Jumlah pergeseran (M) : 3\*n\*(n-1)/2

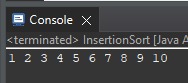
Bubble sort mirip dengan selection, setiap kali proses “Bubble Up” alan memilih maksimum dari elemen yang ada pada sisi unsorted

1. LANGKAH KERJA
2. Codingan untuk class InsertionSort

Input:

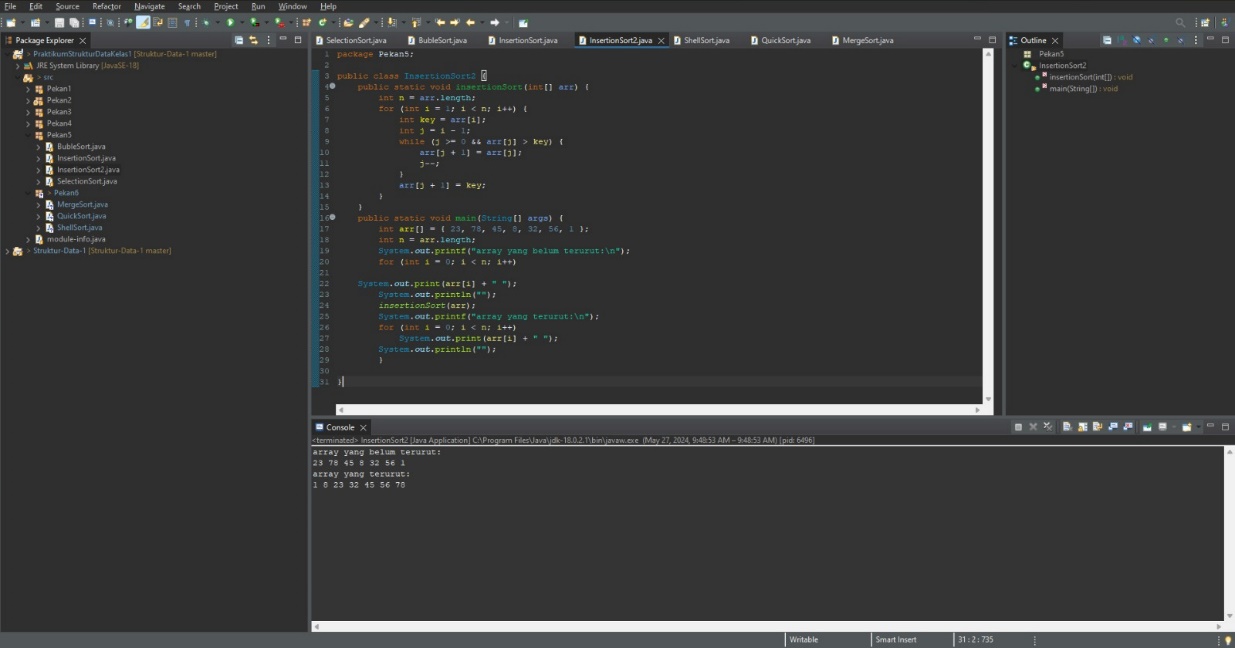


Output:

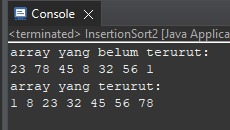


1. Codingan untuk class InsertionSort2

Input:

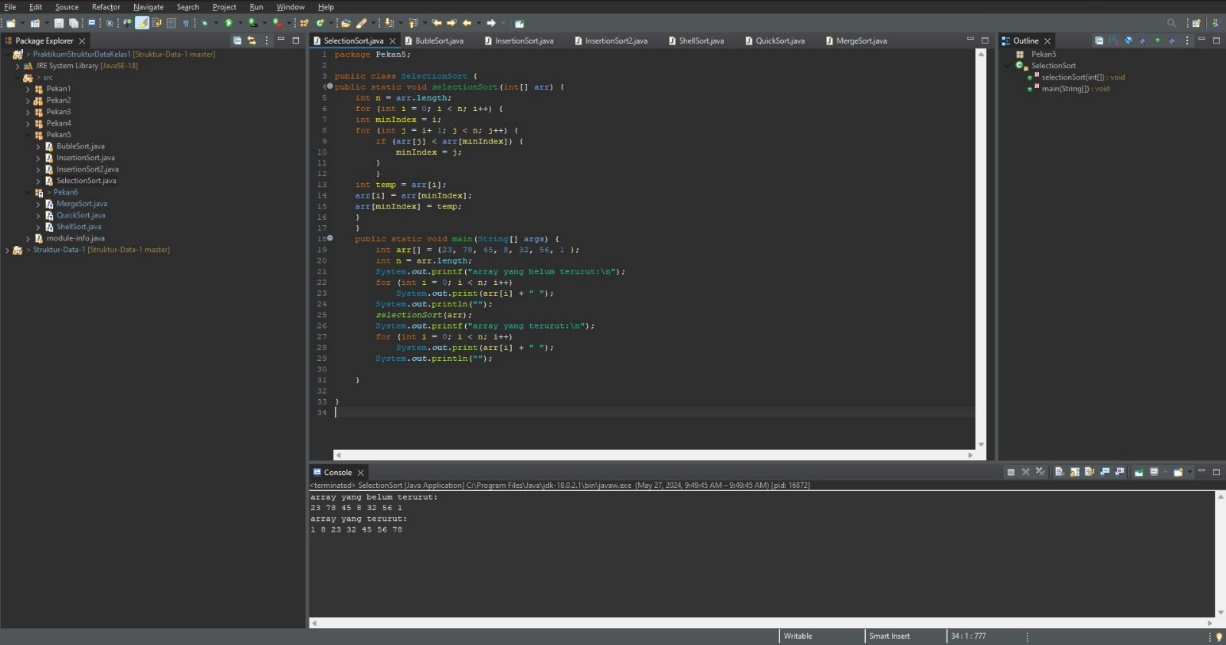


Output:

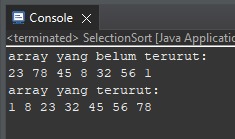


1. Codingan untuk class SelectionSort

Input:

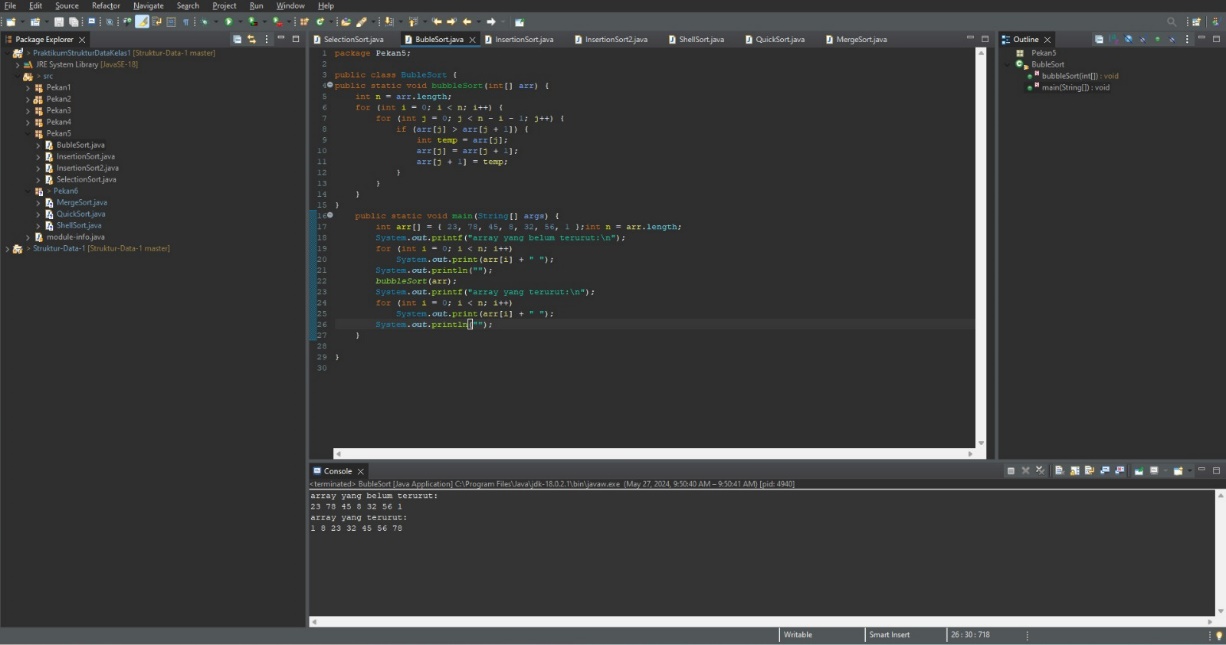


Output:

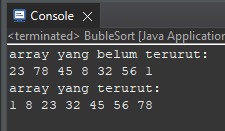


1. Codingan untuk class BubleSort

Input:



Output:



1. KESIMPULAN

Insertion sort adalah metode penyisihan

Selection sort adalah metode seleksi

Bubble sort adalah metode gelembung