

1. Tentukan fungsi `bubbleSort` yang mengambil satu parameter, `ray` (array).
2. Inisialisasi variabel `z` dengan panjang array `ray`.
3. Inisialisasi variabel `swapped` sebagai `False` untuk menentukan pertukaran elemen.
4. Lakukan loop untuk memilah sebanyak `z-1` kali.
 - a. Lakukan loop lagi untuk memilah elemen pada setiap langkah iterasi pertama.
 - i. Periksa apakah `ray[j]` lebih besar dari `ray[j+1]`.
 - ii. Jika ya, setel `swapped` menjadi `True` dan tukar elemen `ray[j]` dan `ray[j+1]`.
5. Cetak array yang telah diurutkan.

Selection Sort:

```plaintext

1. Inisialisasi array `a` dengan beberapa nilai.
2. Lakukan loop untuk memilah setiap nilai pada array `a`.
  - a. Tentukan indeks minimum (`min\_idx`) sebagai `i`.
  - b. Lakukan loop lagi untuk memilah melalui indeks `i+1` hingga akhir array.
    - i. Periksa apakah `a[min\_idx]` lebih besar dari `a[j]`.
    - ii. Jika ya, perbarui `min\_idx` dengan nilai `j`.
  - c. Tukar nilai antara `a[min\_idx]` dan `a[i]`.
3. Cetak array yang telah diurutkan.

#### ### Insertion Sort:

```plaintext

1. Tentukan fungsi `insertionSort` yang mengambil satu parameter, `ray` (array).
2. Lakukan loop untuk memilah semua elemen dalam array, dimulai dari elemen kedua (`i=1`).
 - a. Simpan elemen ke-*i* dalam variabel `key`.
 - b. Inisialisasi variabel `j` sebagai `i-1`.
 - c. Lakukan loop while selama `j` tidak kurang dari 0 dan `key` lebih kecil dari `ray[j]`.
 - i. Geser elemen ke kanan sampai menemukan posisi yang tepat.
 - d. Setel `ray[j + 1]` menjadi `key`.

3. Cetak array yang telah diurutkan.

Catatan:

- Setiap metode memiliki implementasi dan algoritma sendiri-sendiri.
- Pengujian dilakukan dengan beberapa nilai array pada setiap metode pengurutan.
- Setiap array yang telah diurutkan dicetak setelah proses pengurutan.