**LAPORAN PRAKTIKUM**

**STRUKTUR DATA DAN ALGORITMA**

**Judul: Analisis Asimtotik**



**DISUSUN OLEH**

**Ilham Nur Romdoni M0520038**

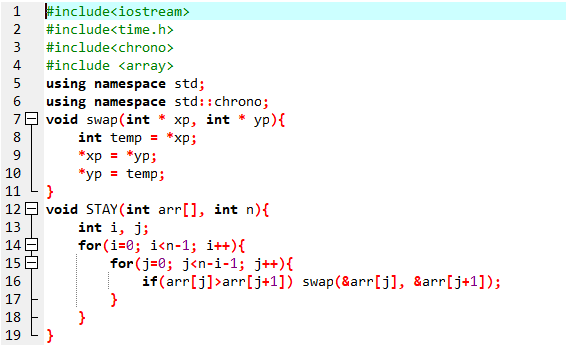
**PROGRAM STUDI INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS SEBELAS MARET**

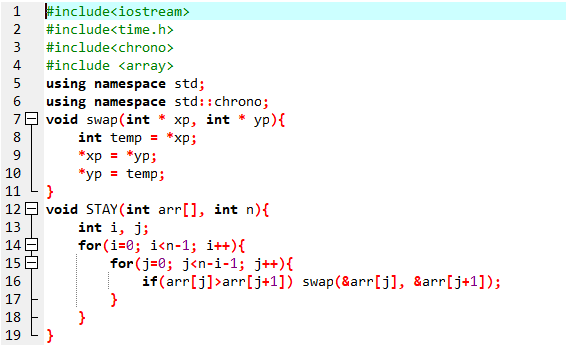
**2021**

1. **algoritma-1.cpp**



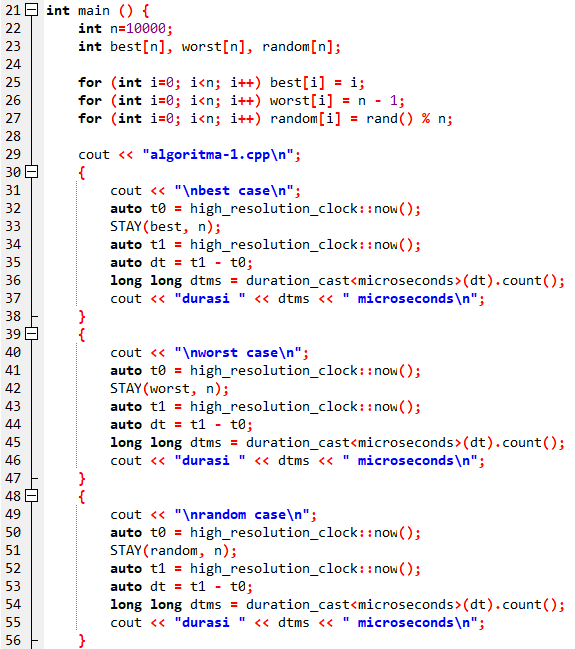
**algoritma-1.cpp** menggunakan fungsi swap yang digunakan untuk menukarkan dua buah nilai. Pada tipe data integer, dibuat sebuah *pointer*. Pointer ini hanya bisa menunjuk ke variabel yang dideklarasikan. Terdapat juga fungsi STAY yang merupakan algoritma *bubble* *sort* untuk mengurutkan angka-angka dengan menggunakan fungsi swap. Algoritma *bubble* *sort* diaplikasikan dengan cara membandingkan elemen ke satu dengan elemen selanjutnya.

1. Big-O

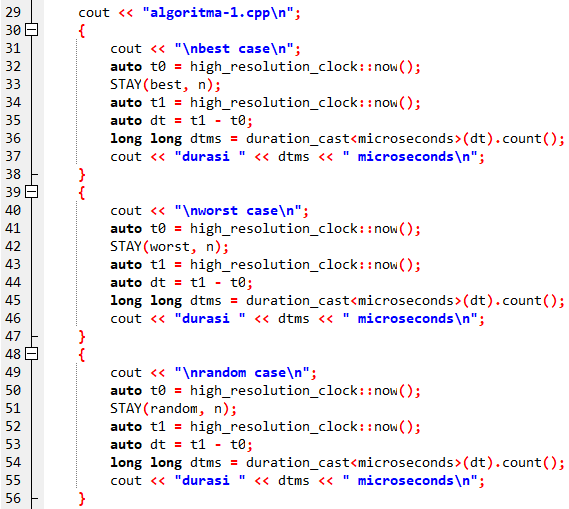


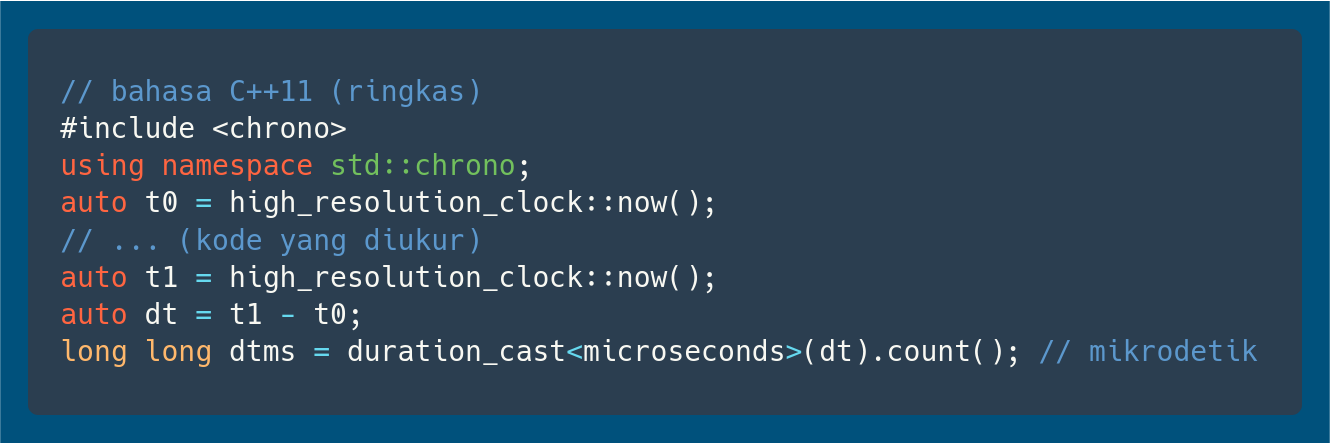
*Loop* pada baris ke 13 dan 14 masing-masing memiliki Big O = O(n) dan operasi lainnya adalah O(1).. Jadi, Big O-nya adalah O(n.n)= O(n2)

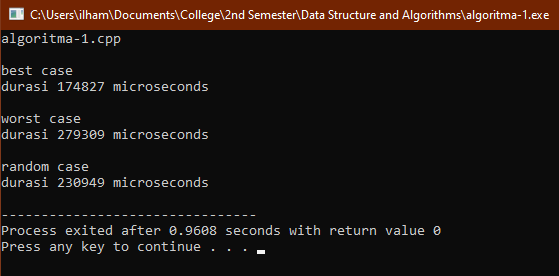
1. Analisa Test Case



Pada baris ke-21, merupakan fungsi main yang akan dijalankan pertama kali saat mendeklarasikan *variable* dan *array*, kemudian untuk *best* *case* yang akan digunakan data dengan urutan asli, untuk *worst* *case* menggunakan data yang urutannya terbalik, sedangkan *random* *case* menggunakan data acak untuk membandingkan.

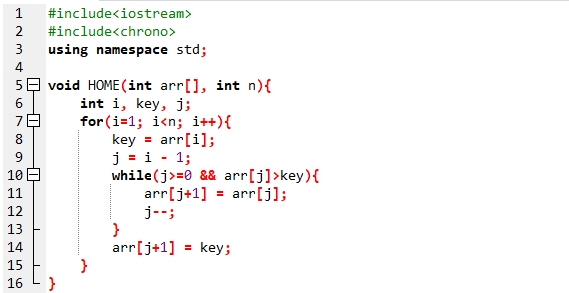


 Baris ke-30 merupakan perintah pengukuran waktu yang memilik format di atas. Kode dijalankan dengan cara mencari selisih waktu selesai dari waktu mulai dan waktu selesai, sehingga didapatkan durasi waktu berjalannya program tersebut dalam satuan *microsecond*.



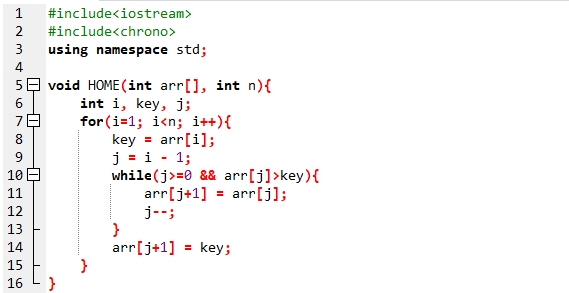
*Compiler* pemrograman menampilkan hasil sesuai dengan jalannya perintah yang dioperasikan dalam algoritma dan komponen masukan pada kode yang diukur.

1. *Best* *Case*: O(n)
2. *Worst* *Case*: O(n2)
3. *Random* *Case*: O(n2)
4. **algoritma-2.cpp**



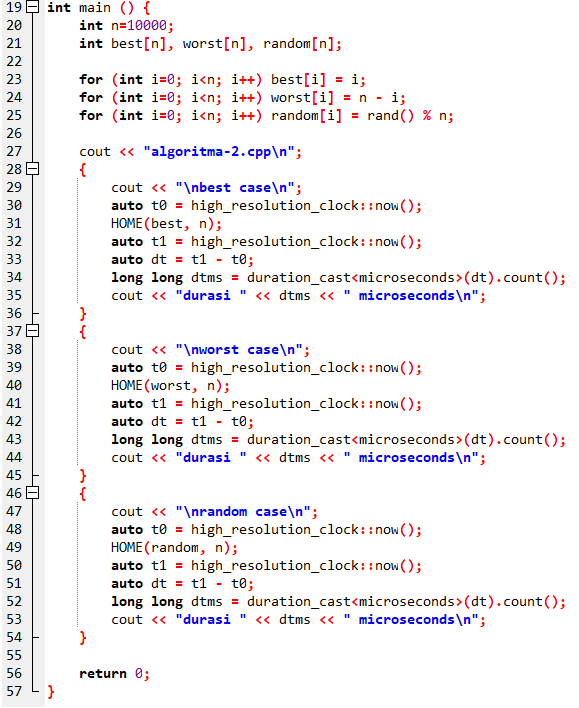
**algoritma-2.cpp** menggunakan fungsi HOME yang merupakan algoritma *insertion* *sort* untuk mengurutkan angka-angka dengan akan membandingkan dua elemen data pertama, mengurutkannya, kemudian mengecek elemen data berikutnya satu persatu dan membandingkannya dengan elemen data yang telah diurutkan.

1. Big-O

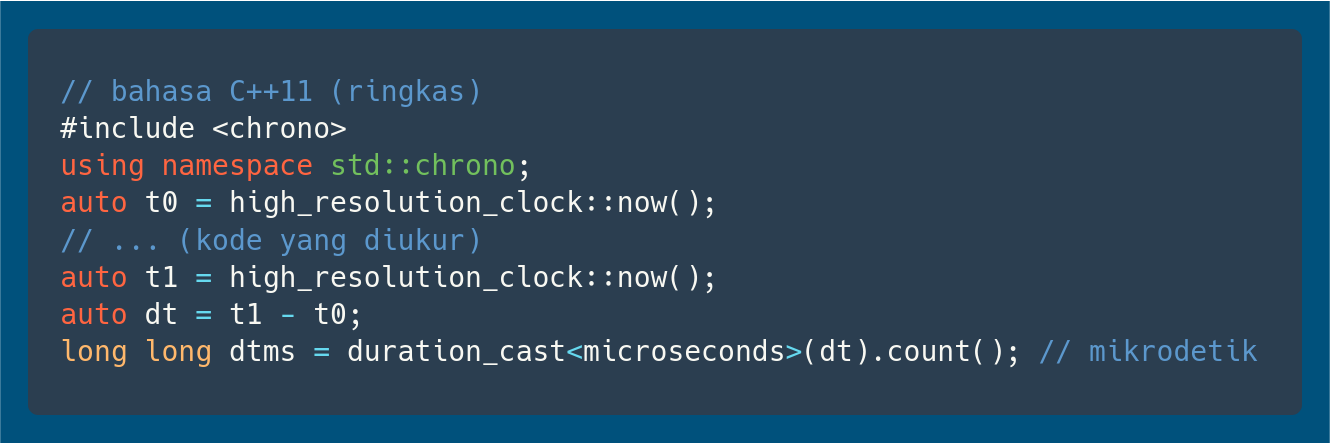


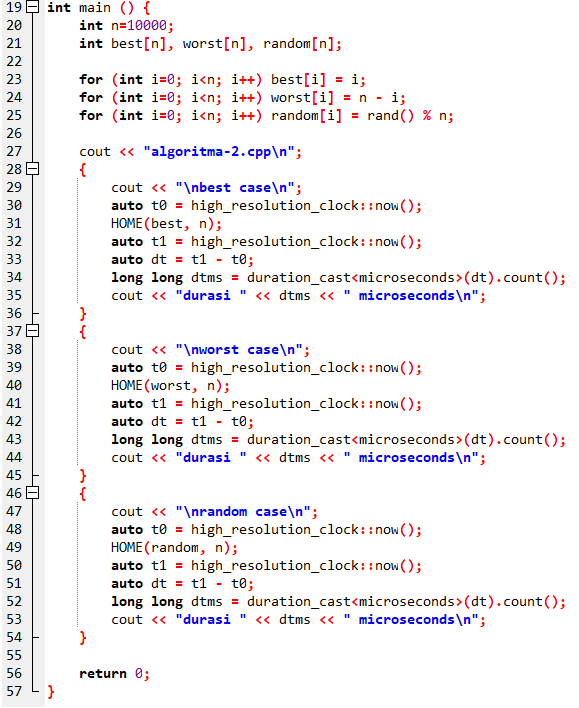
*Loop* pada baris ke-7 dan 10 masing-masing memiliki Big O = O(n) dan operasi lainnya adalah O(1). Jadi, Big O-nya adalah O(n.n)= O(n2)

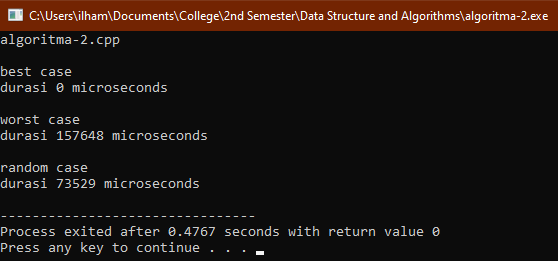
1. Analisa Test Case



Pada baris ke-19, merupakan fungsi main yang akan dijalankan pertama kali saat mendeklarasikan *variable* dan *array*, kemudian untuk *best* *case* yang akan digunakan data dengan urutan asli, untuk *worst* *case* menggunakan data yang urutannya terbalik, sedangkan *random* *case* menggunakan data acak untuk membandingkan.

 Baris ke-27 merupakan perintah pengukuran waktu yang memilik format di atas. Kode dijalankan dengan cara mencari selisih waktu selesai dari waktu mulai dan waktu selesai, sehingga didapatkan durasi waktu berjalannya program tersebut dalam satuan *microsecond*.





*Compiler* pemrograman menampilkan hasil sesuai dengan jalannya perintah yang dioperasikan dalam algoritma dan komponen masukan pada kode yang diukur.

1. *Best* *Case*: O(n)
2. *Worst* *Case*: O(n2)
3. *Random* *Case*: O(n2)
4. **algoritma-3.cpp**

****

**algoritma-3.cpp** menggunakan eraseAT merupakan pernyataan deklarasi *function* yang memberi tahu *compiler* akan berapa tipe data yang dibutuhkan untuk parameter, dan tipe data pada *return value* dari fungsi tersebut.

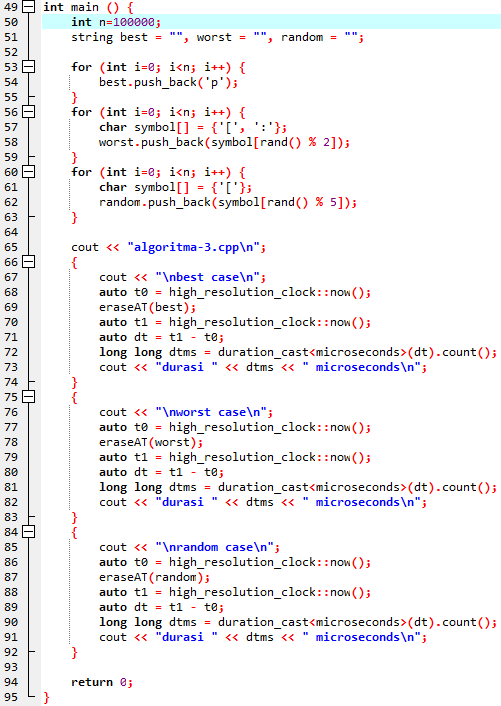
Fungsi if akan mengeksekusi jika suatu kondisi terpenuhi jika tidak maka disinilah fungsi dari else tersebut. Fungsi else akan melakukan perintah *looping*.

1. Big-O

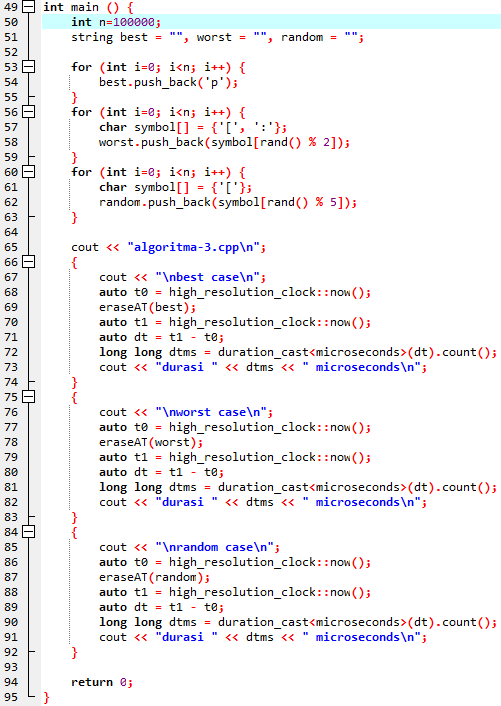
****

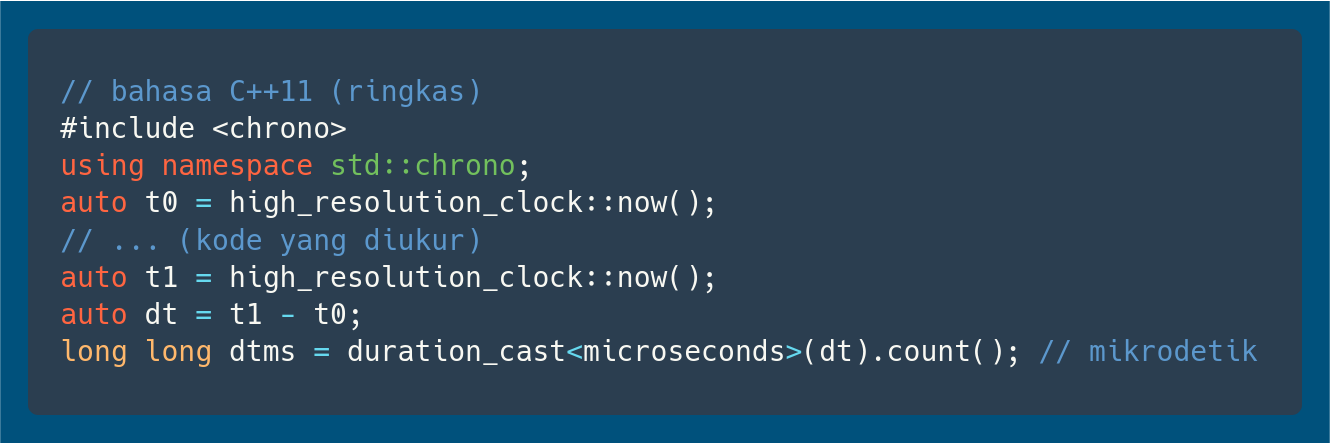
O(n).O(1).O(1)= O(n)

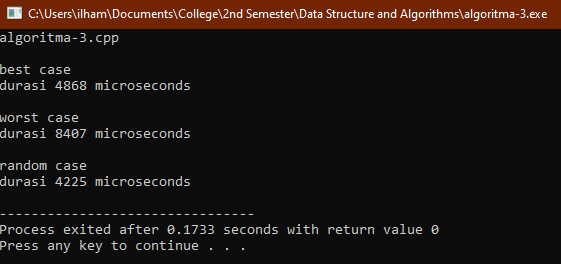
1. Analisa Test Case



Pada baris ke-49, merupakan fungsi main yang akan dijalankan pertama kali saat mendeklarasikan *variable*. Untuk *best* *case* yang akan digunakan tipe data *string* yang tidak berisi salah satu dari ‘[‘, ‘:’, ‘|’, atau ‘]’, untuk *worst* *case* menggunakan tipe data *string* yang hanya berisi salah satu dari ‘[‘ atau ‘:’, sedangkan random case menggunakan tipe data *string* yang hanya berisi salah satu dari dari ‘[‘, ‘:’, ‘|’, ‘]’, atau huruf. Setiap kondisi akan melakukan fungsi *loop*.



 Baris ke-65 merupakan perintah pengukuran waktu yang memilik format di atas. Kode dijalankan dengan cara mencari selisih waktu selesai dari waktu mulai dan waktu selesai, sehingga didapatkan durasi waktu berjalannya program tersebut dalam satuan *microsecond*.



*Compiler* pemrograman menampilkan hasil sesuai dengan jalannya perintah yang dioperasikan dalam algoritma dan komponen masukan pada kode yang diukur.

1. *Best* *Case*: O(n)
2. *Worst* *Case*: O(n)
3. *Random* *Case*: O(n)