**SPESIFIKASI TUGAS BESAR**

**ANALISIS KOMPLEKSITAS ALGORITMA**

**BUBBLE SORT, INSERTION SORT, DAN BUCKET SORT**

## 

**Aldira Fazri – 1301204564**

**Abdul Halim Ichwan – 1301204478**

**Firhan Maulana – 1301208566**

Program Studi S1 Informatika

Universitas Telkom

Jl. Telekomunikasi No.1, Terusan Buahbatu, Bojongsoang, Bandung

e-mail: ald.faz14@gmail.com, halim.ichwan@gmail.com,firhan.m@gmail.com

### PENDAHULUAN

**Sorting (pengurutan) adalah proses mengurutkan n buah bilangan dalam urut menaik (*ascending*)  atau  urut  menurun  (*descending*). Proses  mengurutkan 2  buah bilangan  merupakan  proses membandingkan  bilangan  tersebut  dengan  operator  pembanding yang  kemudian  dilakukan  proses  tukar  posisi  (*exchange  position*)  sesuai  dengan  operator pembandingnya  apakah  bilangan  tersebut  di  urut  secara *ascending*  atau  diurutkan  secara *descending*. Proses   tersebut berlanjut sampai dengan n buah bilangan dapat terurut sempurna. Algoritma yang baik adalah algoritma yang efisien. Efisiensi suatu algoritma diukur dengan waktu dan  memori yang dibutuhkan untuk mengeksekusi algoritma. Algoritma yang efisien adalah algoritma yang meminimalkan penggunaan waktu dan ruang. Setiap algoritma sorting memberikan hasil yang diinginkan, yaitu membuat bilangan menjadi urut dengan menggunakan metode yang berbeda-beda dan juga memiliki kelebihan dan kekurangan masing-masing. Berikut penjelasan mengenai algoritma yang akan dibahas dalam laporan ini. *Insertion Sort* merupakan sebuah teknik pengurutan dengan cara membandingkan dan mengurutkan dua data pertama pada *array*, kemudian membandingkan data para *array* berikutnya apakah sudah berada di tempat semestinya. Algoritma *insertion sort* seperti proses pengurutan kartu yang berada di tangan kita. Algoritma ini dapat mengurutkan data dari besar ke kecil (*Ascending*) dan kecil ke besar (*Descending*). Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algoritma ini adalah Ο() di mana n adalah jumlah item. Misalnya diaplikasikan pada data. Jika data yang anda ingin kelola sudah ada, pengurutan dimulai dengan mengambil satu data kemudian membandingkannya dengan data-data yang ada didepannya. Misal data yang diambil ternyata sudah memenuhi syarat perbandingan, sehingga data yang diambil tersebut akan diletakan di depan data yang dibandingkan, selanjutnya lihat data-data yang dibandingkan akan bergeser mundur. *Bucket sort* adalah algoritma tipe perbandingan yang menetapkan elemen list yang ingin kita urutkan dalam bucket, atau bins. Isi bucket ini kemudian diurutkan dengan algoritma lain. Setelah pengurutan isi bucket ditambahkan, membentuk koleksi yang telah diurutkan. *Bubble Sort* adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan/penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.Algoritma bubble sort ini memiliki kelebihan dan kekurangan. Karena kelebihannya, metode ini adalah cara termudah untuk mengurutkan data. Algoritma bubble sort tidak hanya sederhana, tetapi juga mudah untuk dipahami. Di sisi lain, sisi negatifnya adalah efisiensi. Bubble sort adalah metode pengurutan yang tidak efisien karena pengurutan data yang sangat besar  sangat lambat. Juga, bahkan jika data diurutkan dengan baik, jumlah iterasi tetap sama.Dalam tugas besar Analisis Kompleksitas Algoritma semester ini, kami mendapatkan topik Insertion Sort dan Merge Sort untuk mencari C(n), T(n) , kelas efisiensinya, dan perbedaan running time antara Bubble Sort dengan algoritma sorting Insertion Sort dan Merge Sort. Dalam memvisualisasi dari algoritma tersebut kami menggunakan Bahasa pemrograman python untuk menampilkan simulasi pengurutan yang ditugaskan.**

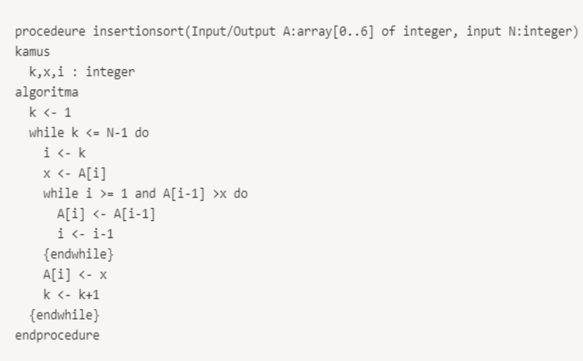
**Kata kunci:** insertion sort, bubble sort, bucket sort, python

array, grafik.

### 1. PENJELASAN PROGRAM

1. Definisi dan pseudocode
2. **Insertion** **Sort**

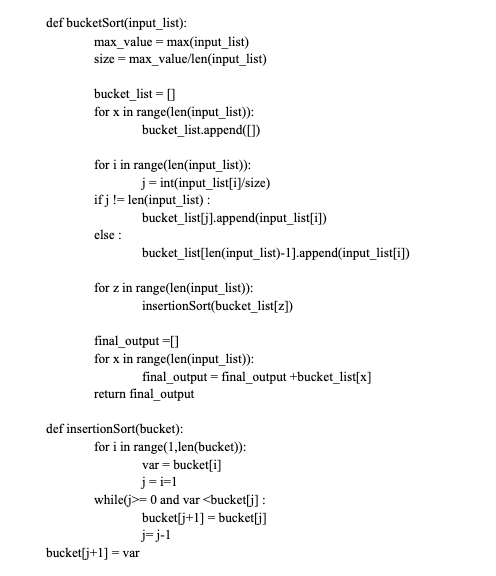
Insertion sort adalah algoritma pengurutan dengan membandingkan yang kemudian diurutkan dari kedua data pertama dalam array, lalu membandingkan apakah data dalam array selanjuatnya dalam posisi yang telah benar. Insertion sort dapat dianalogikan Ketika kita mengurutkan kartu yang kita pegang. Algoritma ini dapat mengurutkan secara ascending maupun descending. Algoritma ini tidak cocok untuk data set dengan jumlah yang besar.



**Gambar 1. Pseudocode Insertion Sort**

1. **Bucket Sort**

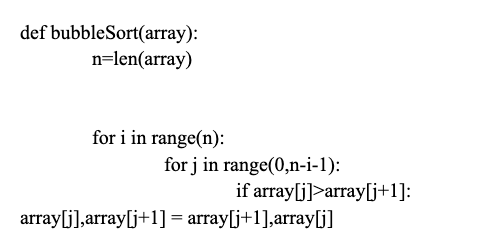
Bucket sort adalah algoritma tipe perbandingan yang menetapkan elemen list yang ingin kita urutkan dalam bucket, atau bins. Isi bucket ini kemudian diurutkan dengan algoritma lain. Setelah pengurutan isi bucket ditambahkan, membentuk koleksi yang telah diurutkan.

****

**Gambar 2. Pseudocode Bucket Sort**

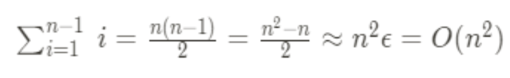
1. **Bubble Sort**

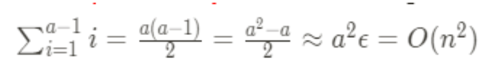
Bubble Sort adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan/penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.

****

**Gambar 3. Pseudocode Bubble Sort**

1. **C(n), T(n), dan Kelas Efisiensi**
2. **Insertion Sort**

**Worst case :**

**Average case :**

**Best case :**



**Kelas efisiensi :**

Kelas efisinesi dari insertion sort adalah Kuadratik

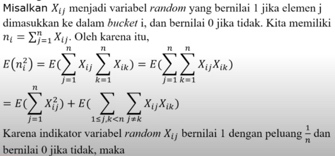
1. **Bucket Sort**

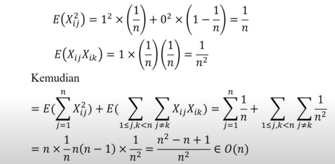
**T(n) :**

* **Worst case**



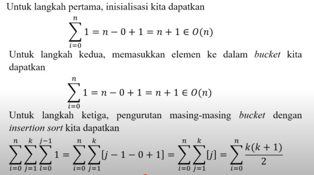
* **Average case**

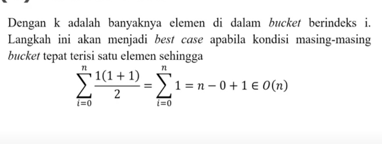




**Gambar 4. T(n) Bucket Sort**

* **C(n)**





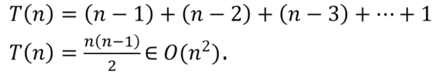
**Gambar 5. C(n) Bucket Sort**

1. **Bubble Sort**

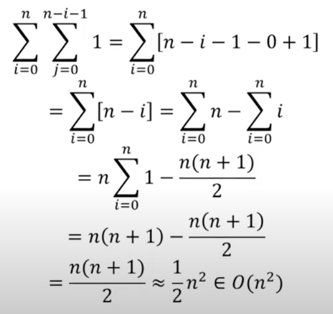
* **T(n)**

****

* **Worst Case**

****

* **C(n)**

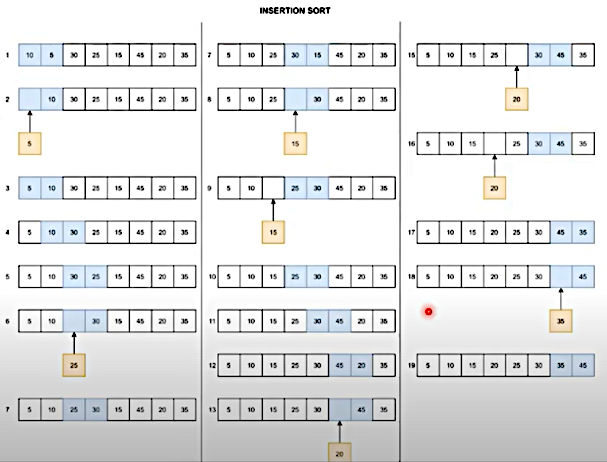
****

**Gambar 6. C(n) Bubble Sort**

### II. STRATEGI ALGORITMA

1. **Insertion Sort**

### Contoh pengurutan



**Gambar 7. Pengurutan Insertion Sort**

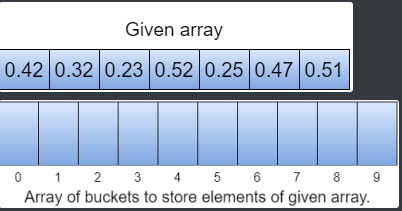
* 1. **Running Time**

**Tabel 1 Tabel perhitungan Insertion Sort**

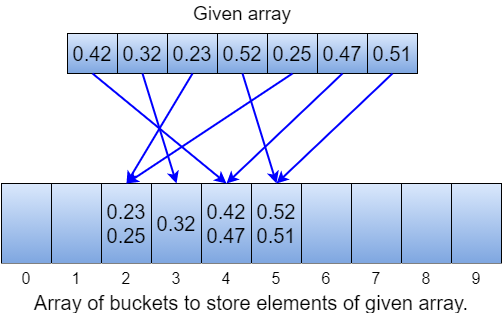
|  |  |
| --- | --- |
| **Input Size(n)** | **Hasil Running Time /ms** |
| **10** | **0.0** |
| **100** | **0.0** |
| **1000** | **0.0300** |
| **10000** | **2.8459** |
| **100000** | **307.33** |

1. **Bucket Sort**

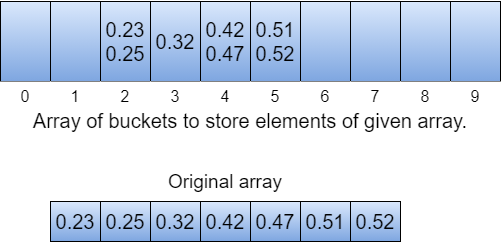
### Contoh pengurutan



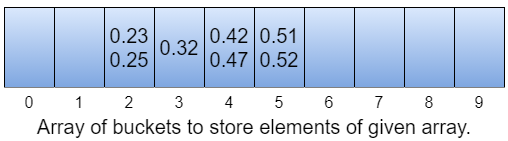
**Gambar 8. Pengurutan Bucket Sort**



**Gambar 9. Pengurutan Bucket Sort**



**Gambar 10. Pengurutan Bucket Sort**



**Gambar 11. Pengurutan Bucket Sort**

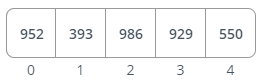
* 1. **Running Time**

**Tabel 2 Tabel perhitungan Bucket Sort**

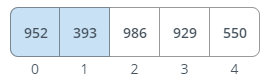
|  |  |
| --- | --- |
| **Input Size(n)** | **Hasil Running Time /ms** |
| **10** | **0.0** |
| **100** | **0.0010** |
| **1000** | **0.0300** |
| **10000** | **2.8459** |
| **100000** | **43.108** |

1. **Bubble Sort**

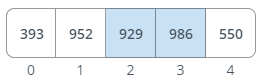
### Contoh pengurutan

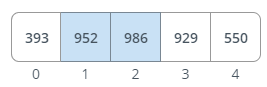


**Gambar 12. Pengurutan Bubble Sort**

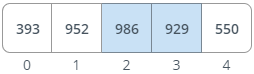


**Gambar 13. Pengurutan Bubble Sort**

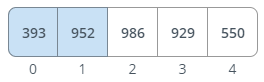


**Gambar 14. Pengurutan Bubble Sort** 

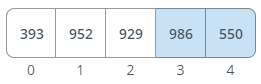
**Gambar 15. Pengurutan Bubble Sort**

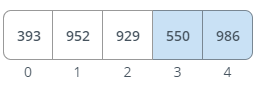


**Gambar 16. Pengurutan Bubble Sort**

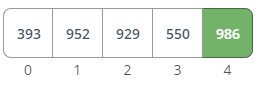


**Gambar 17. Pengurutan Bubble Sort**



**Gambar 18. Pengurutan Bubble Sort** 

**Gambar 19. Pengurutan Bubble Sort**



**Gambar 20. Pengurutan Bubble Sort**

* 1. **Running Time**

**Tabel 3 Tabel perhitungan Bubble Sort**

|  |  |
| --- | --- |
| **Input Size(n)** | **Hasil Running Time /ms** |
| **10** | **0.0** |
| **100** | **0.0010** |
| **1000** | **0.0702** |
| **10000** | **7.1116** |
| **100000** | **937.43** |

**III. FUNGSIONALITAS PROGRAM**

* def create\_random\_list (size)

Untuk membuat data list secara random.

* def bubbleSort(array)

Algoritma bubble sort pengurutan paling sederhana yang bekerja dengan menukar elemen yang berdekatan berulang kali jika urutannya salah.

* def bucketSort(input\_list)

Algoritma bucket sort

1. Buat n bucket kosong (atau daftar).
2. Lakukan hal itu untuk setiap elemen array arr[i].
3. Masukkan arr[i] ke dalam ember[n\*array[i]]
4. Urutkan bucket individu menggunakan insertion sort.

Gabungkan semua bucket yang diurutkan.

* def insertionSort(bucket)

Algoritma insertion sort

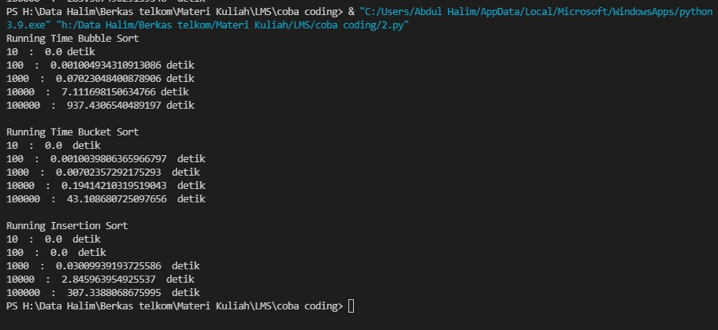
Untuk mengurutkan array berukuran n dalam urutan membesar.

* print("Running Time Bubble Sort")

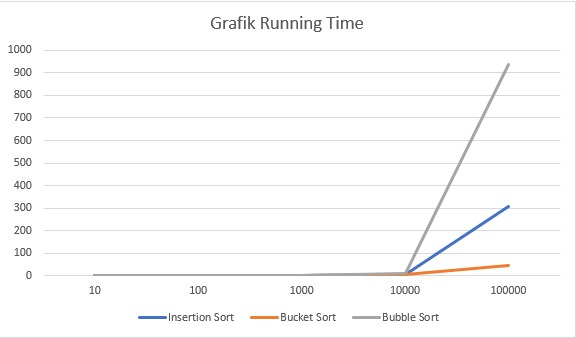
for size in ns

Untuk menghitung running time menggunakan bubble sort.

1. **SCREEN SHOOT OUTPUT**



**Gambar 21. Output program**

****

**Gambar 22. Grafik Hasil Running Time Program**

1. **KESIMPULAN**

Hasil dari kita men sort data menggunakan 3 metode, yaitu bubble sort, bucket sort dan insertion sort. Bucket sort yang paling cepat dalam memproses data secara random karena algoritma nya yang stabil, sedangkan pada bubble sort memiliki algoritma yang simple dan mudah diubah menjadi kode sangat cocok untuk pengurutan data dengan elemen kecil telah terurut, dan untuk insertion sort yaitu Implementasi yang sederhana, paling efisien untuk data berukuran kecil.

**V. REFERENSI**

[1] Rheinadi, R. (2009). Analisis Algoritma Bubble Sort. Bandung, Jawa Barat, Indonesia

[2] Khalid, M.J (t.thn.). Bucket sort in phyton. Diambil kembaliari stack Abuse : <https://stackabuse.com/bucket-sort-un-python>