

BUCKET SORT

Muhammad Akbar Faiz (1303184019)

Pratama Azmi Atmajaya(1303180096)

Afiliasi *gatau ini apa

Alamat *harus banget apa

Email : akbarfaiz@student.telkomuniversity.ac.id ,pramzz@student.telkomuniversity.ac.id

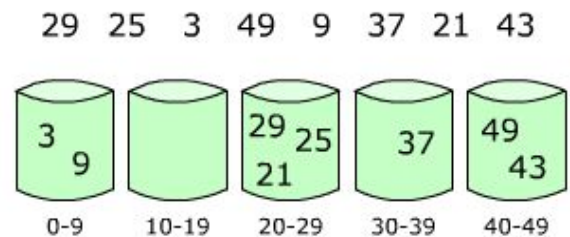
PENDAHULUAN

Pengurutan data dalam struktur data sangat penting untuk data yang bertipe data numerik ataupun karakter sehingga digunakan secara luas dalam aplikasi dimana data tersebut tersusun dengan pola tertentu sehingga tersusun secara teratur berdasarkan aturan tertentu. Menurut Book-shelf, definisi algoritma pengurutan adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dari tiap-tiap elemen. Ada beberapa keuntungan dari pengurutan ini memudahkan dan mempercepat pencarian serta pengecekan elemen dari suatu data. Pengurutan data erat kaitannya dengan pencarian data dan pengelolaan data, dimana dalam kedua hal tersebut terdapat beberapa proses dalam memilah-milah data untuk kebutuhan tertentu. Pencarian data dan pengelolaan data yang efektif tidak dapat terlepas dari faktor keterurutan data yang baik. Apabila data-data yang ada tidak diolah dengan tepat maka akan tercecer dan sulit dipergunakan untuk kepentingan tertentu. Oleh sebab itu, dibutuhkan suatu algoritma yang dapat mengurutkan data sesuai aturan tertentu agar efektif dan efisien. Dengan algoritma yang tepat, data yang dimiliki tidak akan berantakan apalagi sampai hilang ataupun lenyap. Tidak dipungkiri bahwa harga dari sebuah data sangat mahal terutama jika data tersebut penting dan bersifat rahasia. Dalam makalah ini akan membahas penggunaan algoritma Bucket sort dalam mengurutkan elemen-elemen dalam suatu data. Bucket sort ini sendiri jika diartikan ke dalam bahasa Indonesia per kata akan menjadi urut dan ember dimana algoritma ini akan memilah elemen ke dalam ember tertentu sesuai aturan yang ditetapkan.

Kata kunci : Pengurutan, data, elemen, susun.

I. PENJELASAN PROGRAM

Algoritma yang akan digunakan untuk mengurutkan data adalah *Bucket sort*. Di dalam algoritma *Bucket sort* array dari elemen dialokasikan ke dalam beberapa ember. Sesuai namanya, setiap ember akan menampung beberapa elemen yang mirip atau serupa dan ember-ember tersebut akan diurutkan satu sama lain menggunakan algoritma pengurutan lainnya ataupun dengan menerapkan pengurutan ember secara rekursif.



Gambar 1. Contoh Bucket sort kepada suatu data

II. STRATEGI ALGORITMA

Ada beberapa hal yang akan diperkirakan dalam algoritma *Bucket sort*. Hal-hal ini akan dipertimbangkan dalam strategi algoritma.

2.1 Kompleksitas Waktu

Dalam suatu algoritma pasti ada kompleksitas waktu yang terdiri dari 3 kemungkinan.

Tabel 1 Kompleksitas Waktu *Bucket sort*

Worst	$O(n^2)$	Kasus terburuk terjadi jika elemen dalam ember berada dalam urutan terbalik
Average	$O(n)$	Ini terjadi ketika elemen didistribusikan secara acak dalam array. Bahkan jika elemen tidak didistribusikan

		secara seragam, bucket sort berjalan dalam waktu linear.
Best	$O(n+k)$	ketika elemen didistribusikan secara seragam dalam ember dengan jumlah elemen yang hampir sama di setiap ember. Kompleksitas menjadi lebih baik ketika elemen di dalam setiap ember sudah diurutkan.

2.2 Kompleksitas Ukuran

Kompleksitas ukuran *Bucket sort* adalah $O(n^2)$. Karena kami membuat ember dengan panjang yang sama dan setiap ember memiliki sub ember dengan panjang yang sama.

III. FUNGSIONAL PROGRAM

3.1 Penjelasan Pseudocode

Langkah-langkah pada pengurutan data dengan algoritma *Bucket sort* adalah sebagai berikut.

1. Membuat suatu inisial array yaitu ember kosong sebanyak jumlah array yang diberikan.
2. Memasukan setiap objek pada array yang ingin diurutkan ke dalam ember yang sesuai.
3. Melakukan pengurutan pada setiap objek yang ada di tiap ember, disini menggunakan *Insertion sort*.

3.2 Pseudocode

```
def bucket_sort(nums):
    bucket = []
    for i in range(0, len(nums)):
        bucket.append([])

    for i in range(0, len(nums)):
        bucketLocation = int(nums[i] * len(nums) )
        bucket[bucketLocation].append(nums[i])

    # Using Insertion Sort
    for i in range(0, len(bucket)):
        for j in range(1, len(bucket[i])):

            current = bucket[i][j]
            prev = j - 1
            while prev >= 0 and current < bucket[i][prev]:
                bucket[i][prev+1] = bucket[i][prev]
                prev = prev - 1
            bucket[i][prev+1] = current

    index = 0
    for i in range(0, len(bucket)):
        for j in range(0, len(bucket[i])):
            nums[index] = bucket[i][j]
            index = index + 1
    print("Sorted Elements : {}".format(nums))
```

IV. SCREEN SHOOT OUTPUT PROGRAM

V. REFERENSI

- [1] Parta, <https://kreatip.id/materi/implementasi-algoritma/sorting>. Diakses tanggal 9 Desember 2020.
- [2] Neeraj Mishra, <https://www.thecrazyprogrammer.com/2017/02/bucket-sort-in-c.html>. Diakses tanggal 9 Desember 2020.
- [3] Prashant Yadav, [Bucket Sort Algorithm | LearnersBucket](#). Diakses tanggal 15 Desember 2020.