PENUGASAN BIG O NOTATION

MATA KULIAH STRUKTUR DATA

DISUSUN OLEH :

NAMA : DAMAYANTI EKA PUTRI

NPM : 22552011210

DOSEN :

M.Reksa Ariansyah, S.Kom., M.Kom

PROGRAM STUDI

TEKNIK INFORMATIKA RM 22A

SEKOLAH TINGGI TEKNOLOGI BANDUNG

TAHUN AJARAN 2022/2023

BIG O NOTATION

Big O Notation yaitu sebuah cara atau metode untuk melakukan analisa terhadap sebuah algoritma pemrograman terhadap suatu eksekusi.

CONTOH BIG O NOTATION

1. **PENGULANGAN DALAM PENGULANGAN : O(N^2)**
   Biasanya disebut juga Quadratic yaitu pengulangan didalam pengulangan. Quadratic Time adalah ketika runtime dari fungsi kita adalah sebesar n^2, dimana n adalah jumlah input dari fungsi tersebut.
   Contoh :

```
for (let i = 0; i < hotels.length; i++) {
  for (let j = 0; j < hotels.length; j++) {
    // kode untuk membandingkan satu harga dengan harga lainnya...
  }
}
```

2. **PENGULANGAN DALAM SEBUAH SET : O(N)**
   Biasanya disebut juga sebagai Linear, yaitu mencari harga tertinggi dan terendah, yang dilakukan hanya dalam 1 kali pengulangan.
   Linear Time adalah ketika runtime dari fungsi kita berbanding lurus dengan jumlah input yang diberikan.
   Contoh :

```
  for (let i = 0; i < hotels.length; i++) {
          // cari harga paling kecil...
  // cari harga paling besar...
}
```

3. **HANYA SATU OPERASI : O(1)**
   Biasanya disebut juga dengan Constant yaitu asumsi yang sudah diurut berdasarkan harga, lalu tinggal mencari elemen pertama dan terakhir.
   Constant Time artinya banyaknya input yang diberikan kepada sebuah algoritma, tidak akan mempengaruhi waktu proses (*runtime*) dari algoritma tersebut.
   Contoh :

```
  const hotels = [
          { price: 78, brand: "Sheraton Senggigi Beach Resort" },
          { price: 180, brand: "Hotel Tugu Lombok" },
          ...
          ...
          { price: 317, brand: "The Oberio" }
]
```

4. **2. O(log n) — Logarithmic Time**
   Logarithmic Time artinya ketika kita memberikan input sebesar n terhadap sebuah fungsi, jumlah tahapan yang dilakukan oleh fungsi tersebut berkurang berdasarkan suatu faktor.
5. **5. (2^n) —Exponential Time**
   Exponential Time biasanya digunakan dalam situasi dimana kita tidak terlalu tahu

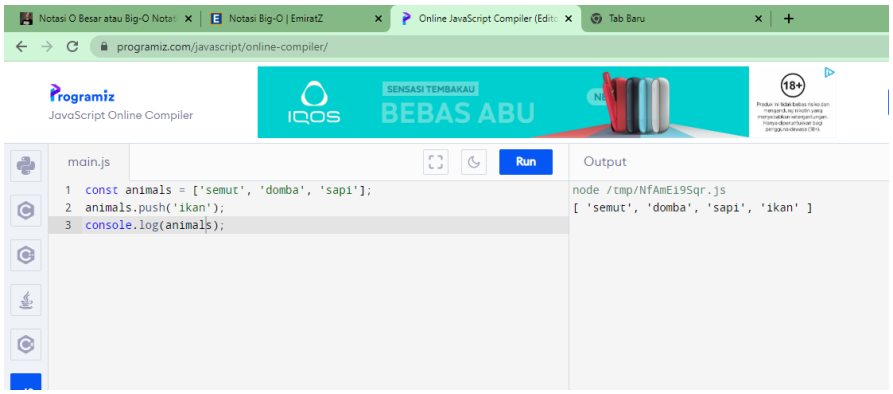
terhadap permasalahan yang dihadapi, sehingga mengharuskan kita mencoba setiap kombinasi dan permutasi dari semua kemungkinan.

ADA BEBERAPA CONTOH BIG O DARI FUNGSI

1. **Array.push()**

   Push() merupakan sebuah metode untuk menambahkan item baru kedalam sebuah array. Item yang ditambahkan akan berada diakhir array tersebut. Contoh :

```
const animals = ['ants', 'goats', 'cows'];
animals.push('fish');
console.log(animals); // ['ants', 'goats', 'cows', 'fish']
```

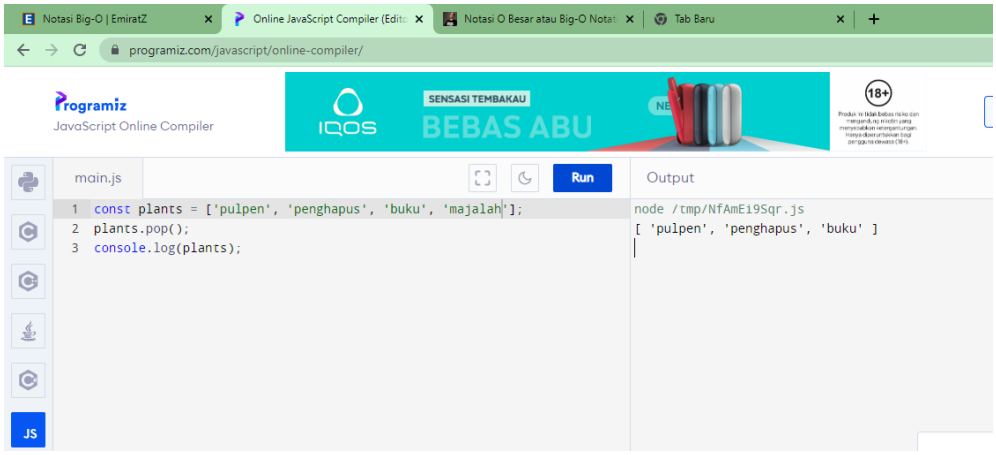


   Karena metode push() tidak peduli dengan seberapa banyak atau sedikit jumlah item yang ada, artinya operasi yang berjalan tetap sama, maka metode push() ini dapat diwakilkan dengan notasi O(1) atau konstan.

2. **Arrap.pop()**

   pop() merupakan sebuah metode yang mengambil item terakhir dari array sehingga jumlah item yang ada di array akan berkurang satu. Contoh :

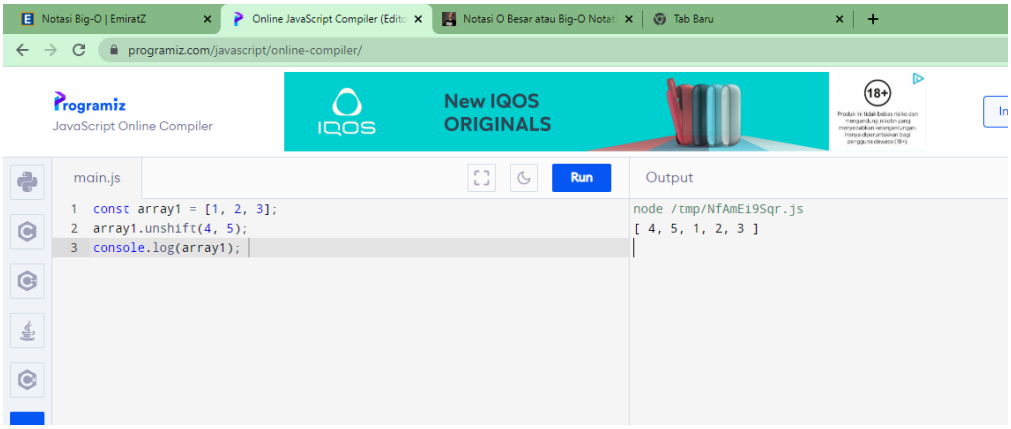
```
const plants = ['broccoli', 'cauliflower', 'cabbage', 'tomato'];
plants.pop();
console.log(plants); // ["broccoli", "cauliflower", "cabbage"]
```



   Mirip seperti metode push() diatas, metode pop() juga tidak mempermasalahkan jumlah item yang ada, artinya operasi yang berjalan tetap sama, maka metode pop() ini juga dapat diwakilkan dengan notasi O(1) atau konstan.

3. **Array.unshift()**
unshift() adalah sebuah metode untuk menambahkan satu atau beberapa item ke bagian awal dari sebuah array.
Contoh :

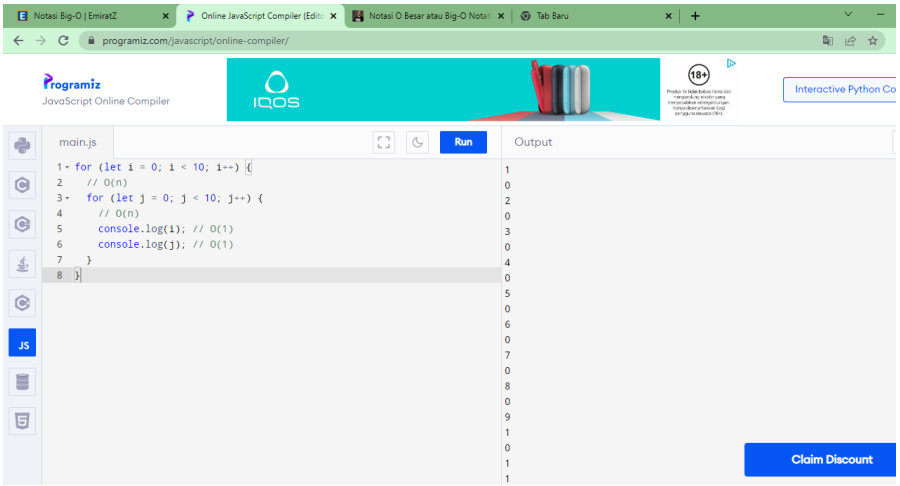
```
const array1 = [1, 2, 3];
array1.unshift(4, 5);
console.log(array1); // [4, 5, 1, 2, 3]
```



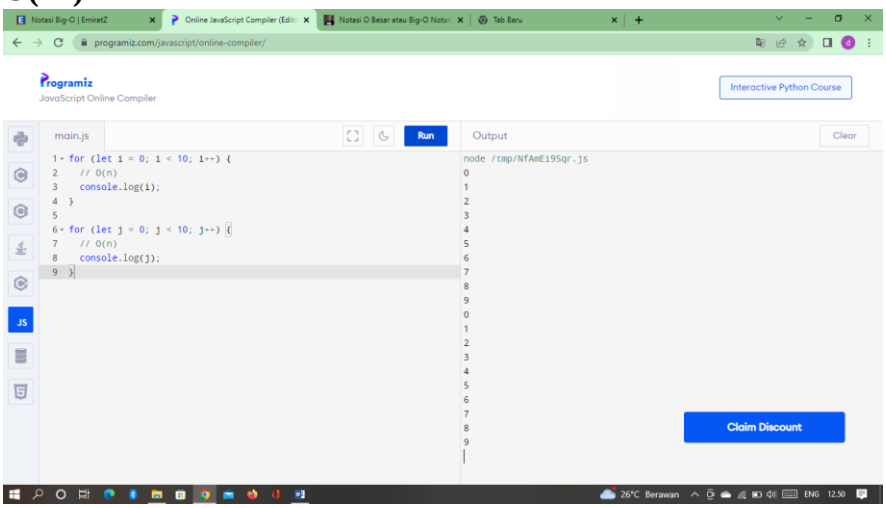

```
function unshift(arr, newItem) {
  let newArr = [];
  newArr[0] = newItem;
  for (let i = 1; i < arr.length + 1; i++) {
    newArr[i] = arr[i - 1];
  }
  return newArr;
}
```

Hal yang menambah kompleksitas adalah ketika harus mengubah indeks dari array karena akan menempatkan item baru di indeks ke-0. Secara otomatis indeks akan bergeser sebanyak satu langkah. Dan karena itu kita menggunakan pengulangan for hingga menjadikan operasi unshift() dapat diwakilkan oleh notasi linear atau O(n).

4. **Menghitung total kompleksitas kode**
terdapat pengulangan bersarang, maka operasi yang digunakan adalah perkalian. Artinya hasil dari pengulangan bersarang tersebut adalah: O(n) * O(n) = O(n2).



5. **O(2n)**



dilambangkan sebagai O(n) ditambah dengan O(n) sehingga menjadi O(2n).

```
for (let i = 0; i < 10; i++) {
  // O(n)
  console.log(i);
}

for (let j = 0; j < 10; j++) {
  // O(n)
  console.log(j);
}
```

6. **O (Log n)**

```
dvdrental=# EXPLAIN ANALYZE SELECT * FROM film WHERE title='Academy Dinosaur';
EXPLAIN ANALYZE SELECT * FROM film WHERE title='Academy Dinosaur';
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Seq Scan on film  (cost=0.00..66.50 rows=1 width=384) (actual time=0.031..0.505 rows=1 loops=1)
   Filter: ((title)::text = 'Academy Dinosaur'::text)
   Rows Removed by Filter: 999
 Planning time: 1.558 ms
 Execution time: 0.605 ms
(5 rows)
```

Tanpa Index

(Sebelum menggunakan index)

```
dvdrental=# CREATE INDEX idx_title on film(title);
CREATE INDEX idx_title on film(title);
CREATE INDEX
dvdrental=# EXPLAIN ANALYZE SELECT * FROM film WHERE title='Academy Dinosaur';
EXPLAIN ANALYZE SELECT * FROM film WHERE title='Academy Dinosaur';
WARNING: terminal is not fully functional
-  (press RETURN)
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Index Scan using idx_title on film  (cost=0.28..8.29 rows=1 width=384) (actual time=0.068..0.069 rows=1
loop   Index Cond: ((title)::text = 'Academy Dinosaur'::text)
 Planning time: 0.502 ms
 Execution time: 0.109 ms
(4 rows)
```

Dengan Index

(sesudah menggunakan index)

Ketika sudah menggunakan index maka dari itu bisa disebut sebagai O(Log n)

KESIMPULAN

O(1)	**Menjalankan sebuah perintah**
O(1)	**Mendapatkan sebuah item dari array, objek atau variabel**
O(log n)	**Pengulangan yang berkurang setengahnya setiap iterasi**
O(n2)	**Pengulangan dalam pengulangan**
O(n3)	**Pengulangan dalam pengulangan dalam pengulangan**