Algoritma dan Struktur Data

Jose Sitanggang

July 2019

1 | Array Dinamis

1.1 Pendahuluan

Array Dinamis merupakan sebuah *array* yang ukurannya dapat bertambah atau berkurang mengikuti jumlah elemen yang akan disimpan ke *array*. Karena Array Dinamis juga adalah *array* maka besar runing time untuk operasi akses atau *update* juga konstan atau $O(1)$. Sedikit berbeda dengan *array*, konsep penambahan elemen di Array Dinamis mirip dengan penambahan elemen di struktur data Stack, yang mana elemen di tambahkan di akhir *array* (operasi ini disebut *push*).

Pada implementasinya, untuk mendapatkan ukuran *array* yang dinamis dilakukan dengan cara membuat *array* baru berukuran **dua kali** ukuran *array* semula ketika pada operasi *push* berikutnya tidak ada ruang kosong di *array* sebelumnya.

1.2 Cara Kerja Array Dinamis

Ketika membuat Array Dinamis ada dua atribut yang harus ditentukan lebih dahulu yaitu *capacity* dan $size$. *capacity* merupakan banyak elemen maksimum yang dapat ditampung Array Dinamis dan $size$ merupakan banyak elemen yang sudah ditampung di Array Dinamis.

Misalkan diberikan Array Dinamis A yang pada awalnya kosong, kemudian akan dilakukan penambahan N buah elemen ke A. Untuk mempermudah, pada implementasinya saat A kosong maka nilai $capacity = 1$ dan $size = 0$, yang artinya A hanya boleh menampung hanya satu elemen pada saat ini.

Akan ada dua kemudian yang terjadi saat operasi *push* dilakukan, yaitu:

1. Kondisi $size < capacity$ artinya operasi *push* dapat dengan mudah dilakukan, yaitu dengan menambahkan elemen di $A[size]$ kemudian $size = size + 1$.

2. Kondisi $size = capacity$ artinya tidak ada ruang kosong lagi untuk menambahkan elemen baru. Disinilah dilakukan operasi melipatgandakan ukuran A, Idenya :

   (a) Membuat *array* baru A' berukuran **dua kali** A.

   (b) Menyalin semua elemen dari A ke A'.

   (c) Menghapus A dan mengubah alamat referensi A ke A' (Konsep pointer).

1.3 Analisis Kompleksitas Array Dinamis

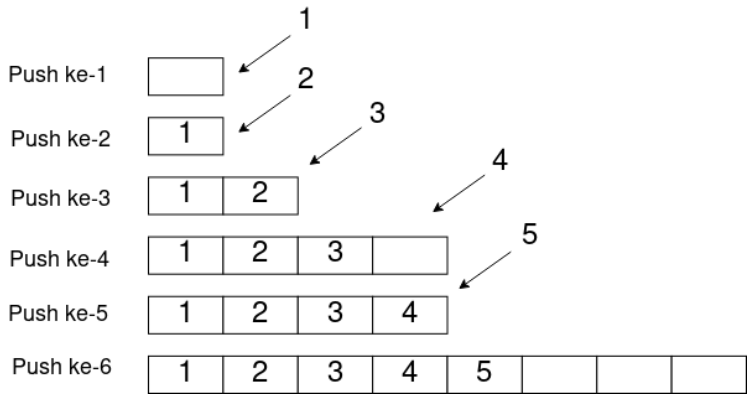

Figure 1.1: Gambar ilustrasi operasi *push*.

Seperti yang diketahui saat melakukan operasi *push* ada dua kondisi yang terjadi, yaitu:

1. Ketika $size < capacity$. Pada kondisi ini operasi *push* dilakukan pada $O(1)$ karena dengan *size* dapat langsung menentukan dimana posisi elemen baru akan diletakkan.

2. Ketika $size = capacity$. Pada kondisi ini memerlukan operasi menyalin elemen ke A', sehingga operasi *push* dilakukan pada $O(k + 1)$ dengan k adalah banyak elemen yang disalin ke A' dan 1 untuk operasi *push* setelah penyalinan.

Sekilas terlihat bahwa operasi *push* pada Array Dinamis akan mengkonsumsi waktu yang banyak. Namun, pada kenyataannya operasi *push* dapat dilakukan cukup baik yaitu pada $O(1)$. Bagaimana bisa ? Karena untuk menyatakan kompleksitas tidak selalu dilihat dari kemungkinan terburuk, tertapi perlu diperhatikan seluruh operasi yang dilakukan (*Amortized Analysis*).

Pada kasus ini ada dua kondisi operasi *push*, oleh karena itu akan dilihat nilai rata-rata dari kedua kondisi tersebut. Gambar diatas adalah merupakan ilustrasi operasi *push* pada A, terlihat bahwa proses lipatganda ukuran A berada di operasi *push* ke-2 *dan* 4 dan jika diteruskan mengikuti pola *push* ke-$2^1, 2^2, 2^3, ..., 2^P$ dengan P adalah banyaknya lipatganda yang terjadi sehingga $2^P = N$ yang artinya sudah cukup ruang untuk menyimpan N buah elemen ke A.

$$\begin{aligned} 2^P &= N \\ \log_2\ 2^P &= \log_2\ N \\ P &= \lfloor \log_2\ N \rfloor \end{aligned}$$

Pada setiap operasi *push* ke-M diperlukan penyalinan elemen sebanyak $\frac{M}{2}$, dengan $M = 2^1, 2^2, .., 2^P$ sehingga total penyalinan yang dilakukan adalah:

$$\begin{aligned} total &= \frac{2^1}{2} + \frac{2^2}{2} + \frac{2^3}{2} .. + \frac{2^P}{2} \\ &= 2^0 + 2^1 + 2^2 + ... + 2^{P-1} \end{aligned}$$

dengan menggunakan rumus deret geometri $S_n = \frac{a(r^n-1)}{r-1}$ dengan $a = 1, r = 2, n = P = \lfloor \log_2\ N \rfloor$

$$\begin{aligned} total &= \frac{(2^{(\log_2\ N)} - 1)}{2-1} \\ &= \frac{N-1}{1} \\ total &= N-1 \end{aligned}$$

dengan mengetahui total penyalinan elemen $total = N - 1$, Seperti yang diketahui sebelumnya untuk melakukan operasi *push* pada kondisi $size = capacity$ adalah $O(k+1)$ dan k adalah banyak elemen yang disalin maka dapat diperoleh total operasi di kondisi $size = capacity$ adalah $O_2 = N + P - 1$ sedangkan untuk kondisi $size < capacity$ adalah $O_1 = N - 1$. Sehingga nilai rata-rata $O_{1,2}$

adalah :

$$\begin{aligned} O_{1,2} &= \left\lceil \frac{O_1 + O_2}{N} \right\rceil \\ &= \left\lceil \frac{(N-P) + (N+P-1)}{N} \right\rceil \\ &= \left\lceil \frac{2N-1}{N} \right\rceil \\ O_{1,2} &= 2 \end{aligned}$$

Dengan *Amortized Analysis* disimpulkan bahwa operasi *push* pada Array Dinamis dapat dilakukan pada kompleksitas konstan yaitu $O(2)$ atau bisa dituliskan $O(1)$.