**Running Time & Big O Notation**

Apa itu Running time :

* Kita selalu dihadapkan pada kebutuhan untuk mencaari algoritma yang paling efisien dalam hal waktu dan ukuran
* Berapa banyak waktu yang bis akita hemat menggunakan binary search
* Jumlah pengecekan maksimum akan sama dengan jumlah elemen dalam list tersebut atau linear time
* Ketika enggunakan binary search, maka Ketika dihadapkan 100 item maka akan membutuhkan paling banyak 7 kali pengecekan.
* Binary search akan menerapkan log time

Big O Notation :

* Big O notasi adalah notasi khusus untuk mendeskripsikan seberapa cepat suatu algoritma
* Sebaiknya kita dapat membandingkan algoritma satu dengan lainnya
* Bagaimana implementasi Big O notation dalam membandingkan beberapa algoritma
* Binar search jauh lebih cepat untuk dijalankan, tetapi simple search jauh lebih muda untuk diimplementasikan sehingga lebih kecil kemungkinan untuk ditemukannya sebuah bug di aplikasinya.
* Seiring meningkatnya jumlah item, peningkatan waktu proses binary search jauh lebih kecil.
* Tidak cukup untuk hanya mengetahui suatu algoritma untuk berjalan. Kita utuh mencari tau berapa running time meningkat seiring dengan peningkatan ukurannya. Oleh karena itu Big O notation ini meenjadi lebih relevan dalam membandngkan algoritma

Penerapan Big O Notaion :

* Big O notation mengekspresikan kecepatan suatu algoritma
* Penlisan Big O notation dalam bentuk *O(n)*
* Big O notation tidak mengekspresikan kecepatan algoritma dalam satuan detik. Tetapi menggunakan bagaimana kecepatan suatu algoritma berkembang

Daftar Big O Notation yang umum ditemui. (Tercepat sampai terlambat) :

* O(Log n) atau log time. Contoh : Binary search.
* O(n), atau linear time. Contoh : Simple search.
* O(n\*log n). Contoh : quicksort
* O(n2). Contoh: Selection sort
* O(n!)