

LAPORAN STUDI KASUS ANALISIS DAN DESAIN ALGORITMA



Disusun Oleh :

- | | |
|--------------------------|-------------|
| 1. Anggito Suryo Baskoro | (E41191560) |
| 2. Achmad Aji Wicaksono | (E41191405) |
| 3. Alvin Eko Cahyo F | (E41191422) |
| 4. Danang Ardianto | (E41191453) |
| 5. Atala Ilyasa Pratama | (E41191649) |

TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2020

1. Uraian Masalah

Sebuah Toko Buku akan mengadakan evaluasi penjualan buku dan pembelian stok buku berdasarkan sisa stok buku yang paling sedikit. Proses pembelian stok buku hanya dilakukan pada 10 judul buku yang jumlah sisa stoknya paling sedikit walaupun trend penjualan untuk buku tersebut tidak tinggi. Pembelian stok buku hanya ditambahkan sebanyak 5 item per-judul buku. Berikut data stok buku yang tersisa:

NO	JUDUL	STOK
1	The 7 Habits of Highly Effective People	2
2	How to Win Friends and Influence People	4
3	Think and Grow Rich	5
4	Awaken The Giant Within	1
5	As a Man Thinketh	7
6	The Greatest Salesman in The World	6
7	Don't Sweat the Small Stuff	5
8	Drive	3
9	The Power of Positive Thinking	3
10	Man's Search for Meaning	2
11	Make Your Bed : Little Things That Can Change Your Life... And Maybe the World	2
12	Alchemist	1

2. Penyelesaian Algoritma Masalah

a. Pilihan Algoritma

Algoritma yang digunakan yaitu algoritma Quick Sort

b. Penjelasan Alasan Algoritma Quick Sort dipilih

Algoritma Quick Sort merupakan algoritma penyortiran suatu data dengan cepat, sehingga suatu permasalahan diatas cocok menggunakan algoritma quick sort dikarenakan data diatas membutuhkan suatu cara penyortiran ascending agar tahu berapa banyak judul buku yang perlu ditambahkan stoknya.

c. Pseudocode Implementasi Algoritma Quick Sort Terhadap Studi Kasus

- Step 1 : Panjang data diperiksa, jika panjang data lebih dari 10 maka jalankan Step 2, jika tidak maka langsung ke Step 3

If (A.length > 10)

- Step 2 : Urutkan data secara ascending menggunakan quicksort

```
procedure QuickSort(input/output A : TabelInt, input i,j: integer)
{ Mengurutkan tabel A[i..j] dengan algoritma Quick Sort.
  Masukan: Tabel A[i..j] yang sudah terdefinisi elemen-elemennya.
  Keluaran: Tabel A[i..j] yang terurut menaik.
}
Deklarasi
  k : integer

Algoritma:
  if i < j then          { Ukuran(A) > 1 }
    Partisi(A, i, j, k)  { Dipartisi pada indeks k }
    QuickSort(A, i, k)   { Urut A[i..k] dengan Quick Sort }
    QuickSort(A, k+1, j) { Urut A[k+1..j] dengan Quick Sort }
  endif
```

```
procedure Partisi(input/output A : TabelInt, input i, j : integer,
  output q : integer)
{ Membagi tabel A[i..j] menjadi upatabel A[i..q] dan A[q+1..j]
  Masukan: Tabel A[i..j] yang sudah terdefinisi harganya.
  Keluaran upatabel A[i..q] dan upatabel A[q+1..j] sedemikian sehingga
  elemen tabel A[i..q] lebih kecil dari elemen tabel A[q+1..j]
}
Deklarasi
  pivot, temp : integer

Algoritma:
  pivot ← A[(i + j) div 2]  { pivot = elemen tengah }
  p ← i
  q ← j
  repeat
    while A[p] < pivot do
      p ← p + 1
    endwhile
    { A[p] ≥ pivot }

    while A[q] > pivot do
      q ← q - 1
    endwhile
    { A[q] ≤ pivot }

    if p ≤ q then
      { pertukarkan A[p] dengan A[q] }
      temp ← A[p]
      A[p] ← A[q]
      A[q] ← temp

      { tentukan awal pemindaian berikutnya }
      p ← p + 1
      q ← q - 1
    endif
  until p > q
```

- Step 3 : Tambahkan 5 item pada stok buku 10 data pertama

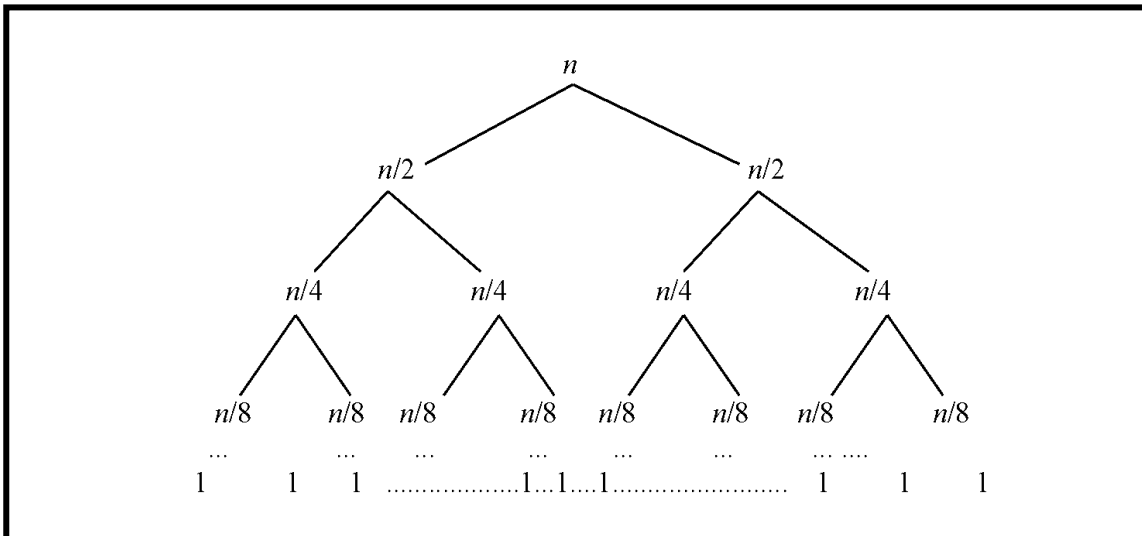
```
x = 0  
  
if (A.length > 10)  
    x = 10  
else  
    x = A.length  
endif  
  
for (i = 1; i <= x; i++) do  
    A[i] = A[i] + 5  
endfor
```

3. Nilai Big-O

- Jika n data lebih dari 10, nilai Big-O quicksort

a. Kasus Terbaik (Best Case)

Kasus terbaik terjadi bila pivot adalah elemen median sedemikian sehingga kedua upatabel berukuran relatif sama setiap kali pemartisian



$$T(n) = \begin{cases} a & , n = 1 \\ 2T(n/2) + cn & , n > 1 \end{cases}$$

Rekursif quicksort

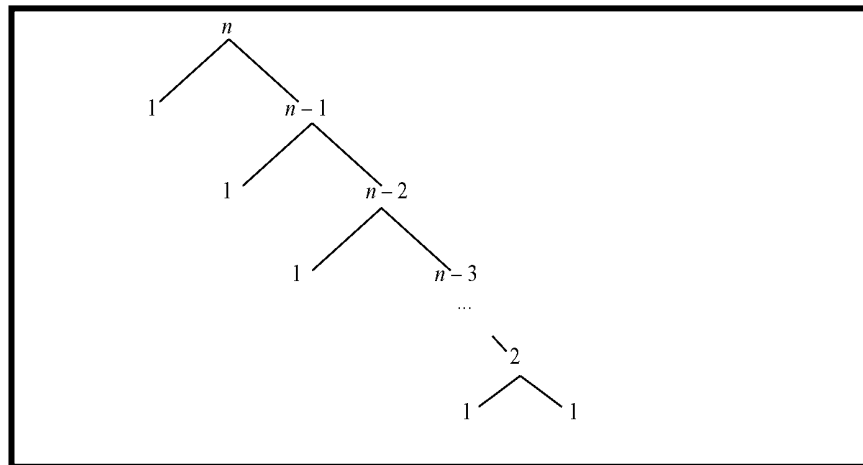
Fungsi partisi

Penyelesaian (seperti pada *Merge Sort*):

$$T(n) = 2T(n/2) + cn = na + cn^2 \log n = O(n^2 \log n).$$

b. Kasus Terburuk (Worst Case)

Kasus ini terjadi bila pada setiap partisi pivot selalu elemen maksimum (atau elemen minimum) tabel. Dan kasus jika tabel sudah terurut menaik/menurun



Kompleksitas waktu pengurutan:

$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$

Rekursif quicksort

Fungsi partisi

Penyelesaian (seperti pada *Insertion Sort*):

$$T(n) = T(n-1) + cn = O(n^2).$$

- Jika n data kurang dari 10, nilai Big-O perulangan *for*

$$T(n) = n$$

$$O = n$$

4. Kesimpulan

Permasalahan diatas dapat diselesaikan dengan beberapa langkah, langkah pertama yaitu memeriksa data apakah panjang data lebih dari 10, jika lebih dari 10 maka perlu melakukan pengurutan data, jika panjang data kurang dari 10 maka langsung ke langkah selanjutnya yaitu penambahan nilai data

Mengurutkan data dari data yang terkecil ke data terbesar menggunakan algoritma sorting, disini kami menggunakan algoritma quicksort karena algoritma ini dinilai cepat dalam pengurutan data

Setelah data terurut, kemudian masuk ke langkah selanjutnya yaitu menambahkan masing-masing 5 nilai pada 10 data pertama