

**LAPORAN UJIAN AKHIR SEMESTER
DESAIN DAN ANALISIS ALGORITMA**

TUGAS 2



DISUSUN OLEH

Muhammad Defaroyan

M0519059

**PROGRAM INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
UNIVERSITAS SEBELAS MARET
2020**

1. Soal Tugas #1

Diberikan permasalahan 0/1 Knapsack sebagai berikut :

Terdapat 4 obyek dengan bobot (W_i) dan Profit (P_i) masing-masing :

W_1	2	P_1	2000
W_2	5	P_2	3000
W_3	10	P_3	5000
W_4	5	P_4	100

Sedangkan kapasitas Knapsack (K) = 16

Tugas mahasiswa adalah :

- a. Tuliskan algoritma brute force untuk menyelesaikan masalah tersebut
 1. Menghitung seberapa banyak kombinasi yang dapat dihasilkan dari data yang ada
 2. Membuat loop sebanyak kemungkinan yang dapat terjadi
 3. Membuat sebuah array yang berisikan kombinasi yang dapat dibuat. Array ini berupa data bit
 4. Array bit yang sebelumnya dibuat, diubah sesuai dengan nilai bit yang ada dengan nilai 1 yang berarti objek akan diambil, dan 0 yang berarti objek tidak akan diambil
 5. Dari bit yang sudah diubah nilainya maka akan dibandingkan dengan kandidat kombinasi terbanyak pada waktu yang berkaitan, perbandingan ini ditujukan untuk mencari nilai tertinggi
 6. Hasil dari brute force knapsack ditampilkan pada console dalam bentuk tabel
- b. Tuliskan program (source code) dengan bahasa pemrograman yang anda kuasai berikut hasil running program untuk menyelesaikan masalah tersebut
 - Source code
Source Code dibuat menggunakan bahasa pemrograman javascript kemudian dijalankan menggunakan nodeJs.

```

const w =[2,5,10,5];
const p =[2000,3000,5000,100]
const knapsack = 16;
var bit = [0,0,0,0]
var bestW = 0;
var bestP=0;
var bestComb;
var loc;
var batas = w.length
//Loop sebanyak kombinasi yang mungkin dibuat
for (var i = 0; i < Math.pow(2,batas); i++){
    var j = 0;
    tW = 0
    tP = 0
    //membuat kombinasi berupa nilai nilai bit
    while (bit[j] !=0 && j<batas-1){
        bit[j] = 0;
        j = j+1;
    }
    bit[j] = 1;
    //console.log('\tIter ' + i)
    //console.log(bit);
    //Mengubah tiap bit menjadi nilai nilai yang ada
    for (var k = 0; k<batas;k++){
        if(bit[k] == 1){
            tW = tW + w[k]
            tP = tP + p[k]
        }
    }
    //Membandingkan tiap bit untuk mencari nilai tertinggi
    if (tP > bestP && tW < knapsack){
        loc = null;
        bestP = tP
        bestW = tW
        for(var l=0;l<bit.length;l++){
            if(bit[l] == 1){
                var pos = l+1
                loc = loc + pos + ' ';
            }
        }
    }
}

function Result(weight,price,lokasi){
    this.Barang_Ke = lokasi
    this.Total_Berat = weight;
    this.Total_Harga = price;
}

console.log("Hasil dari Brute Force :")
console.table(new Result(bestW,bestP,loc))

```

- Hasil running program

E:\Kuliah\Sem 3\DAA\UAS\1>node bruteforce.js
Hasil dari Brute Force :

(index)	Values
Barang_Ke	'2 3 '
Total_Berat	15
Total_Harga	8000

2. Soal Tugas #2

Diberikan permasalahan sederhana, yakni perpangkatan : a^n

Tugas mahasiswa adalah menyelesaikan permasalahan tersebut dengan menggunakan 3 algoritma yang berbeda yaitu :

a. Algoritma brute force

- Pseudo code

```
function exponentBF(a : real, pangkat : integer) = real
Deklarasi
    i : integer
    res : real
Algoritma
    res = 1
    for i = 0 to pangkat do
        res = res * a
    endfor
    return res
```

b. Algoritma decrease & conquer

- Pseudo code

```
function Dec(a : real, pangkat : integer) = real
Algoritma
    if pangkat == 0
        return 1
    else
        return a*Dec(a,pangkat-1)
    endif
```

c. Algoritma divide & conquer

- Pseudo code

```

function Div(a : real, pangkat : integer) = real
Deklarasi
    setengah : integer
Algoritma
    if pangkat == 0
        return 1
    else if pangkat == 1
        return a
    endif

    setengah = floor(pangkat/2)
    return Div(a, setengah) * Div(a, pangkat-setengah)

```

Untuk setiap algoritma, tuliskan *source code* dan running-time saat program dijalankan dengan nilai n yang signifikan

- **Source Code**

Code yang dibuat menggunakan bahasa pemrograman javascript kemudian dijalankan menggunakan nodeJs. Ketiga metode diatas dibuat dalam satu file yang sama untuk mempermudah melakukan perbandingan

```

const { stdin, stdout } = process;

//Fungsi untuk membuat input dari stdin dan stdout
function input(question) {
  return new Promise((resolve, reject) => {
    stdin.resume();
    stdout.write(question);
    stdin.on('data', data => resolve(data.toString().trim()));
    stdin.on('error', err => reject(err));
  });
}

//Fungsi untuk algoritma bruteforce
function bruteforce(a, pangkat){
  var hasil
  hasil = 1
  for(var i = 0; i<pangkat; i++){
    hasil = hasil * a;
  }
  return(hasil);
}

//Fungsi untuk algoritma decrease and conquer
function decrease(a, pangkat){
  if(pangkat == 0){
    return 1;
  }else{
    return decrease(a,pangkat-1) * a
  }
}

//Fungsi untuk algoritma divide and conquer
function divide(a, pangkat){
  if(pangkat == 0){return 1}
  else if (pangkat == 1){return a}

  var setengah = Math.floor(pangkat/2)
  return (divide(a,setengah)) * (divide(a,pangkat-setengah))
}

//fungsi pembuatan objek yang kemudian akan di print sebagai tabel
function Result(bf,dec,div){
  this.Waktu_Brute_Force = bf
  this.Waktu_Decrease_Conquer = dec;
  this.Waktu_Divide_Conquer = div;
}

//fungsi main
async function main() {
  try{
    //Input user untuk nilai a dan pangkat
    var a = await input('Masukkan Nilai Konstanta = ')
    var pangkat = await input('Masukkan nilai pangkat = ')
  }
}

```

```

//Timer pemanggilan fungsi bruteforce
const bf0 = process.hrtime();
var brute = bruteforce(a,pangkat)
const bf1 = process.hrtime(bf0);
//console.log(brute)

//Timer pemanggilan fungsi decrease and conquer
const dec0 = process.hrtime()
var decreas = decrease(a,pangkat)
const dec1 = process.hrtime(dec0);
//console.log(decreas)

//Timer pemanggilan fungsi divie and conquer
const div0 = process.hrtime();
var div = divide(a,pangkat)
const div1 = process.hrtime(div0);
//console.log(div)

//Output waktu yang digunakan bernilai microsecond
console.table(new Result(bf1[1]/ 1000,
                        dec1[1]/ 1000,
                        div1[1]/ 1000))
}
catch(err) {
    console.log("There's an error!");
    console.log(err);
}
process.exit();
}

//Program dijalankan dengan memanggil fungsi main
main()

```

- Hasil Running time

Input		Output								
a	n	Running Time								
2	10	<div>E:\Kuliah\Sem 3\DAA\UAS\2>node two.js Masukkan Nilai Konstanta = 2 Masukkan nilai pangkat = 10</div> <table><thead><tr><th>(index)</th><th>values</th></tr></thead><tbody><tr><td>Waktu_Brute_Force</td><td>57.2</td></tr><tr><td>Waktu_Decrease_Conquer</td><td>36.001</td></tr><tr><td>Waktu_Divide_Conquer</td><td>58.799</td></tr></tbody></table>	(index)	values	Waktu_Brute_Force	57.2	Waktu_Decrease_Conquer	36.001	Waktu_Divide_Conquer	58.799
(index)	values									
Waktu_Brute_Force	57.2									
Waktu_Decrease_Conquer	36.001									
Waktu_Divide_Conquer	58.799									

2	100	E:\Kuliah\Sem 3\DAA\UAS\2>node two.js Masukkan Nilai Konstanta = 2 Masukkan nilai pangkat = 100 <table><tr><th>(index)</th><th>Values</th></tr><tr><td>Waktu_Brute_Force</td><td>73.6</td></tr><tr><td>Waktu_Decrease_Conquer</td><td>51.5</td></tr><tr><td>Waktu_Divide_Conquer</td><td>90.1</td></tr></table>	(index)	Values	Waktu_Brute_Force	73.6	Waktu_Decrease_Conquer	51.5	Waktu_Divide_Conquer	90.1
(index)	Values									
Waktu_Brute_Force	73.6									
Waktu_Decrease_Conquer	51.5									
Waktu_Divide_Conquer	90.1									
2	1000	E:\Kuliah\Sem 3\DAA\UAS\2>node two.js Masukkan Nilai Konstanta = 2 Masukkan nilai pangkat = 1000 <table><tr><th>(index)</th><th>Values</th></tr><tr><td>Waktu_Brute_Force</td><td>134.299</td></tr><tr><td>Waktu_Decrease_Conquer</td><td>173.5</td></tr><tr><td>Waktu_Divide_Conquer</td><td>321.399</td></tr></table>	(index)	Values	Waktu_Brute_Force	134.299	Waktu_Decrease_Conquer	173.5	Waktu_Divide_Conquer	321.399
(index)	Values									
Waktu_Brute_Force	134.299									
Waktu_Decrease_Conquer	173.5									
Waktu_Divide_Conquer	321.399									
2	10000	E:\Kuliah\Sem 3\DAA\UAS\2>node two.js Masukkan Nilai Konstanta = 2 Masukkan nilai pangkat = 10000 <table><tr><th>(index)</th><th>Values</th></tr><tr><td>Waktu_Brute_Force</td><td>1681.399</td></tr><tr><td>Waktu_Decrease_Conquer</td><td>1686.8</td></tr><tr><td>Waktu_Divide_Conquer</td><td>3494</td></tr></table>	(index)	Values	Waktu_Brute_Force	1681.399	Waktu_Decrease_Conquer	1686.8	Waktu_Divide_Conquer	3494
(index)	Values									
Waktu_Brute_Force	1681.399									
Waktu_Decrease_Conquer	1686.8									
Waktu_Divide_Conquer	3494									