

LAPORAN TUGAS 01
DESAIN DAN ANALISIS ALGORITMA

Judul: Greedy



DISUSUN OLEH
ILHAM NUR ROMDONI M0520038

PROGRAM INFORMATIKA
FAKULTAS MIPA
UNIVERSITAS SEBELAS MARET
2021

TUGAS 01 : 5 Oktober 2021

Diberikan persoalan *Integer Knapsack* dengan data properti objek sebagai berikut :

PROPERTI OBJEK			
i	W_i	P_i	P_i / W_i
1	200	80	0.4
2	100	70	0.7
3	90	36	0.4
4	40	8	0.2
5	20	20	1.0
6	10	4	0.4

Fungsi kendala :

Kapasitas *Knapsack* $K = 100$.

1. Tugas anda adalah melakukan **CODING** untuk menyelesaikan persoalan di atas dengan menggunakan pendekatan :
 - a. Greedy by Weight
 - b. Greedy by Profit
 - c. Greedy by Density

```
#include <iostream>
using namespace std;

void ByWeight(int num, float weight[], float profit[], float density[], float kapasitas) {

    float x[num], tp = 0, temp;
    int i, j, u = kapasitas;

    for (i = 0; i < num; i++) {
        for (j = i + 1; j < num; j++) {
            if (weight[i] > weight[j]) {
                temp = weight[j];
                weight[j] = weight[i];
                weight[i] = temp;

                temp = profit[j];
                profit[j] = profit[i];
                profit[i] = temp;

                temp = density[j];
                density[j] = density[i];
                density[i] = temp;
            }
        }
    }
}
```

```

    cout << "Weight setiap item adalah: ";
    for (i = 0; i < num; i++) {
        if (weight[i] > u){
            x[i] = 0;
            break;
        }else {
            x[i] = 1;
            tp = tp + profit[i];
            u = u - weight[i];
            cout << weight[i] << " ";
        }
    }
    cout << "\nWeight total adalah: " << kapasitas - u;
    cout << "\nProfit total adalah: " << tp;
}

void ByProfit(int num, float weight[], float profit[], float density[], float kapasitas) {

    float x[num], tp = 0, temp;
    int i, j, u = kapasitas;

    for (i = 0; i < num; i++) {
        for (j = i + 1; j < num; j++) {
            if (profit[i] < profit[j]) {
                temp = profit[j];
                profit[j] = profit[i];
                profit[i] = temp;

                temp = weight[j];
                weight[j] = weight[i];
                weight[i] = temp;

                temp = density[j];
                density[j] = density[i];
                density[i] = temp;
            }
        }
    }

    cout << "Weight setiap item adalah: ";
    for (i = 0; i < num; i++) {
        if (weight[i] > u){
            x[i] = 0;
            continue;
        }else {
            x[i] = 1;
            tp = tp + profit[i];
            u = u - weight[i];
            cout << weight[i] << " ";
        }
    }
    cout << "\nWeight total adalah: " << kapasitas - u;
    cout << "\nProfit total adalah: " << tp;
}

void ByDensity(int num, float weight[], float profit[], float density[], float kapasitas) {

    float x[num], tp = 0, temp;
    int i, j, u = kapasitas;

    for (i = 0; i < num; i++) {
        for (j = i + 1; j < num; j++) {
            if (density[i] < density[j]) {
                temp = density[j];
                density[j] = density[i];
                density[i] = temp;

                temp = weight[j];
                weight[j] = weight[i];
                weight[i] = temp;

                temp = profit[j];
                profit[j] = profit[i];
                profit[i] = temp;
            }
        }
    }

```

```

        }
    }
}

cout << "Weight setiap item adalah: ";
for (i = 0; i < num; i++) {
    if (weight[i] > u){
        x[i] = 0;
        continue;
    }else {
        x[i] = 1;
        tp = tp + profit[i];
        u = u - weight[i];
        cout << weight[i] << " ";
    }
}
cout << "\nWeight total adalah: " << kapasitas - u;
cout << "\nProfit total adalah: " << tp;
}

int main(){
    float weight[] = {200, 100, 90, 40, 20, 10};
    float profit[] = {80, 70, 36, 8, 20, 4};
    float kapasitas = 100;
    int num = sizeof(profit) / sizeof(profit[0]);
    int i, j;
    float density[num];

    for (i = 0; i < num; i++) {
        density[i] = profit[i] / weight[i];
    }

    cout << "Greedy by Weight: \n";
    ByWeight(num, weight, profit, density, kapasitas);
    cout << endl;
    cout << endl;
    cout << "Greedy by Profit: \n";
    ByProfit(num, weight, profit, density, kapasitas);
    cout << endl;
    cout << endl;
    cout << "Greedy by Density: \n";
    ByDensity(num, weight, profit, density, kapasitas);
    cout << endl;
    cout << endl;
    return(0);
}

```

OUTPUT

```

Select D:\Documents\College\3rd Semester\Algorithm Design and Analyse\Integer Knapsack with Gr...
Greedy by Weight:
Weight setiap item adalah: 10 20 40
Weight total adalah: 70
Profit total adalah: 32

Greedy by Profit:
Weight setiap item adalah: 100
Weight total adalah: 100
Profit total adalah: 70

Greedy by Density:
Weight setiap item adalah: 20 10 40
Weight total adalah: 70
Profit total adalah: 32

-----
Process exited after 0.06556 seconds with return value 0
Press any key to continue . . .

```

2. Apakah Pendekatan **Greedy** yang anda implementasikan dalam kasus di atas memberikan solusi optimal?

PROPERTI OBJEK				<i>Greedy by</i>			Solusi Optimal
i	W_i	P_i	P_i / W_i	<i>profit</i>	<i>weight</i>	<i>density</i>	
1	200	80	0.4	0	0	0	0
2	100	70	0.7	1	0	0	1
3	90	36	0.4	0	0	0	0
4	40	8	0.2	0	1	1	0
5	20	20	1.0	0	1	1	0
6	10	4	0.4	0	1	1	0
Total weight				100	70	70	70
Total profit				70	32	32	32

Solusi optimal didapatkan dengan memaksimalkan profit tetapi tetap memperhatikan weight agar tidak melebihi kapasitas. Dilihat dari implementasi kasus di atas, pendekatan greedy by profit memberikan solusi optimal tidak dengan greedy by weight dan density. Tetapi dalam kasus lain yang pernah saya temui, pendekatan greedy by profit terkadang juga tidak dapat memberikan solusi optimal.