***Pengertian***

*Knapsack* merupakan optimasi pengangkutan barang atau disebut juga optimasi kombinatorial. *Knapsack problem*adalah suatu masalah bagaimana cara menentukan pemilihan barang dari sekumpulan barang di mana setiap barang tersebut mempunyai berat dan profit masing – masing, sehingga dari pemilihan barang tersebut didapatkan profit yang maksimum.

***Tujuan***

Tujuan *Knapsack problem* untuk mendapatkan keuntungan yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media transportasi tersebut.

***Jenis-Jenis Knapsack Problem***

Terdapat beberapa variasi Knapsack problem:

* *0/1 Knapsack problem*

Setiap barang hanya tersedia 1 unit, take it or leave it.

***/0 Knapsack***

**Persoalan**: Diberikan *n* buah objek dan sebuah *knapsack* dengan kapasitas bobot *W*. Setiap objek  memiliki properti bobot (*weigth*) *wi* dan keuntungan(profit) *pi*.  Objektif persoalan adalah memilih memilih objek-objek yang dimasukkan ke dalam*knapsack* sedemikian sehingga memaksimumkan keuntungan. Total bobot objek yang dimasukkan ke dalam *knapsack* tidak boleh melebihi kapasitas *knapsack*.  (Aplikasi: masalah pengangkutan barang)

Persoalan 0/1 *Knapsack* dapat kita pandang sebagai mencari himpunan bagian (*subset*) dari keseluruhan objek yang muat ke dalam *knapsack* dan memberikan total keuntungan terbesar.

Algoritma *exhaustive search* untuk persoalan 0/1 *Knapsack* ini adalah:

1. Enumerasikan (*list*) semua himpunan bagian dari himpunan dengan n objek.
2. Hitung (evaluasi) total keuntungan dari setiap himpunan bagian dari langkah 1.
3. Pilih himpunan bagian yang memberikan total keuntungan terbesar.

**Contoh**:

Tinjau persoalan 0/1 *Knapsack* dengan *n* = 4. Misalkan objek-objek tersebut kita beri nomor 1, 2, 3, dan 4. Properti setiap objek *i* dan kapasitas *knapsack* adalah sebagai  berikut

*w*1 = 2;    *p*1 = 20

*w*2 = 5;    *p*1 = 30

*w*3 = 10;  *p*1 = 50

*w*4 = 5;    *p*1 = 10

Kapasitas *knapsack* *W* = 16

**Penyelesaian dengan metode *exhaustive search***

*Exhaustive search* adalah teknik pencarian solusi secara solusi *brute force* untuk masalah yang melibatkan pencarian elemen dengan sifat khusus, biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. Meskipun algoritma *exhaustive* secara teoritis menghasilkan solusi, namun waktu atau sumber daya yang dibutuhkan dalam pencarian solusinya sangat besar.

Langkah-langkah pencarian solusi 0/1 *Knapsack* secara *exhaustive search* dirangkum dalam tabel di bawah ini:

[](http://blog.ub.ac.id/ayrie/files/2012/02/tabel-knapsack.png)

Himpunan bagian objek yang memberikan keuntungan maksimum adalah {2, 3} dengan total keuntungan adalah 80. Solusi persoalan 0/1 *Knapsack* di atas adalah *X* = {0, 1, 1, 0}. Berapa banyak himpunan bagian dari sebuah himpunan dengan *n*elemen? Jawabnya adalah 2*n*. Waktu untuk menghitung total bobot objek yang dipilih = *O*(*n*). Sehingga, Kompleksitas algoritma *exhaustive search* untuk persoalan 0/1 *Knapsack* adalah

**[http://blog.ub.ac.id/ayrie/files/2012/02/rumus-knapsack-150x44.png](http://blog.ub.ac.id/ayrie/files/2012/02/rumus-knapsack.png)**

0/1 *Knapsack* adalah contoh persoalan yang mempunyai kompleksitas algoritma eksponensial. 0/1 *Knapsack* digolongkan sebagai persoalan NP (*Non-deterministic Polynomial*), karena tidak mungkin dapat ditemukan algoritma polinomial untuk memecahkannya.