OPTIMASI PEMAKAIAN BAHAN BAKU DENGAN ALGORITMA PROGRAM DINAMIS SEKUENSIAL

Dion Jogi Parlinggoman / 13509045

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha No. 10, Bandung 40132
13509045@std.stei.itb.ac.id

ABSTRAK

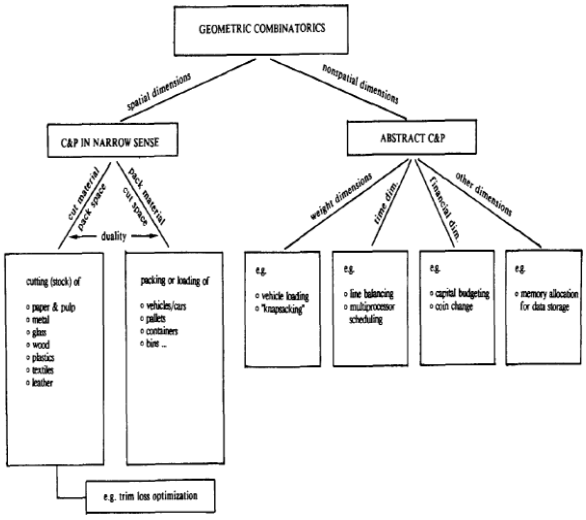
Pengaturan produksi merupakan usaha-usaha pengelolaan secara optimal penggunaan sumber daya yang terbatas (atau sering disebut faktor-faktor produksi), seperti tenaga kerja, mesin-mesin, peralatan, bahan mentah, dan sebagainya – dalam proses pengubahan bahan mentah menjadi berbagai produk dan jasa. Para manajer produksi dan operasional berusaha untuk memperhitungkan dan mengarahkan berbagai masukan (*input*) sumber daya yang digunakan agar dapat memproduksi sesuatu sebagai keluaran (*output*) dalam jumlah, kualitas, harga, waktu dan tempat tertentu sesuai dengan permintaan dari konsumen dan standar yang ditetapkan perusahaan. Pemakaian bahan baku yang tidak terkontrol dan sia-sia akan sangat mempengaruhi kualitas proses dalam produksi, produk tersebut, dan layanan dari perusahaan industri manufaktur. Oleh karena itu, sangat dibutuhkan sebuah solusi untuk memecahkan "cutting-stock problem" dengan hasil yang memuaskan dan optimal.
Pada makalah ini dijelaskan penerapan sebuah solusi paralel dari metode *sequential dynamic programming* untuk memecahkan masalah 2D knapsack (*cutting-stock problem*) karena metode ini dapat melakukan optimasi dengan meminimalkan sisa pemotongan / pemakaian bahan baku dengan baik.

Kata kunci: Optimasi bahan baku manufaktur, *cutting stock problem*, dan *Knapsack Problem*

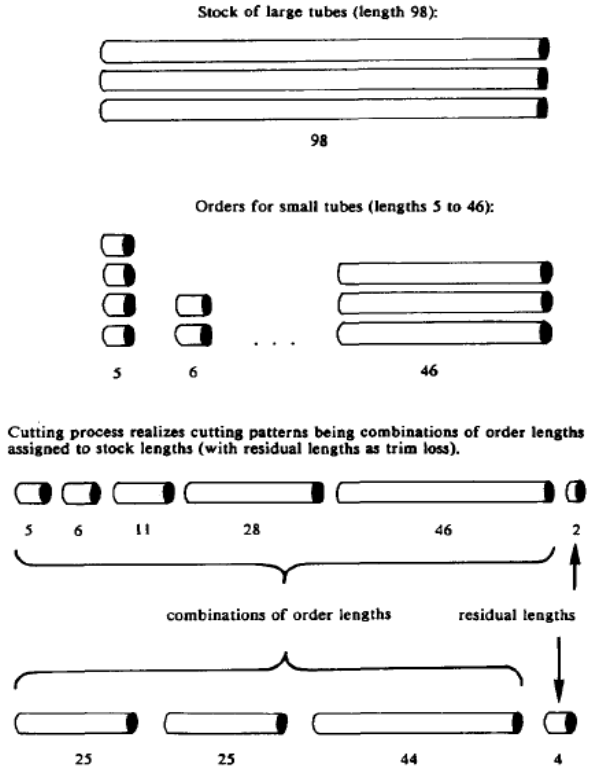
I. PENDAHULUAN

Pengendalian Aktivitas Produksi (*Production Activity Control* = PAC) bertujuan untuk mempertahankan keseimbangan antara sumber-sumber daya manufaktur yang tersedia dan permintaan total. Fungsi dari PAC, yang sering disebut sebagai:

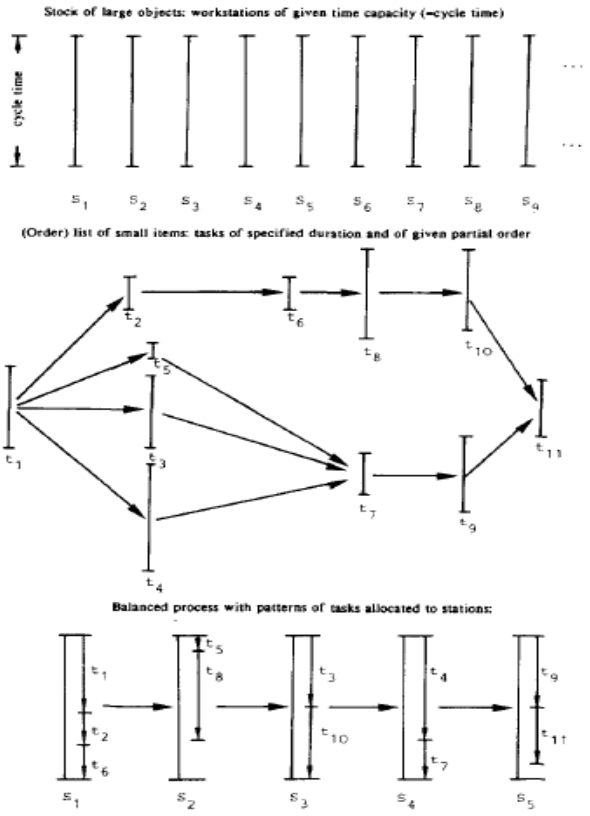
shop floor control (SFC) adalah melakukan aktivitas-aktivitas sebagaimana telah direncanakan, melaporkan hasil-hasil operasi, dan memperbaiki atau merevisi rencana-rencana yang diperlukan untuk mencapai hasil yang diinginkan. PAC melakukan umpan-balik melalui laporan pengukuran barang produksi dan pengaturan sumber daya yang aktual dan membandingkannya dengan rencana-rencana yang telah dipertimbangkan terlebih dahulu.
Dengan demikian, PAC merupakan komponen yang penting dari *close-loop* MRP. MRP adalah *manufacturing resource planning*. MRP adalah mekanisme untuk menghitung material yang dibutuhkan, waktu ketika material dibutuhkan, dan jumlah material yang dibutuhkan. MRP sering dikaitkan dengan Enterprise Resource Planning. PAC mencakup aktivitas-aktivitas keseluruhan dari *shop-floor scheduling and control* yang biasa disebut sebagai *shop-floor control* yang secara keseluruhan merupakan bagian dalam PAC, serta sebagian aktivitas-aktivitas penjadwalan dan tindak lanjut terhadap pemasok (*supplier scheduling and follow-up*) yang merupakan bagian terbesar dalam PAC.



Gambar 1.1 Masalah dalam perhitungan kombinatorial.



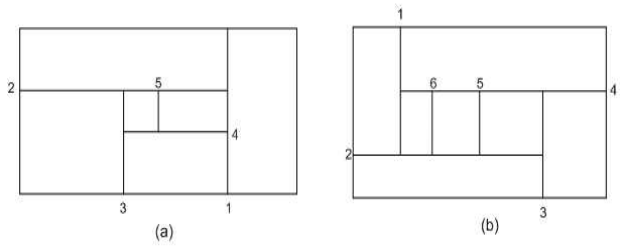
Gambar 1.2 Masalah *cutting stock problem.*



Gambar 1.3 Masalah CSP dimodelkan dengan garis.

Makalah ini membahas suatu permasalahan yang sering dijumpai dalam bidang industri, seperti industri pengolahan kaca, industri pengolahan kulit, dan industri pembuatan kapal, dalam meminimumkan kebutuhan bahan baku. Bahan baku yang tersedia umumnya berbentuk persegi yang harus dipotong-potong menjadi bentuk-bentuk yang lebih kecil yang disebut *final*. Setiap *final* mempunyai jumlah permintaan (*demand*) tertentu. Masalah pemotongan bahan baku ini dikenal dengan sebutan masalah *cutting stock*. Jika *final* yang ada berbentuk persegi panjang, maka masalah *cutting stock* tersebut disebut masalah *cutting stock* dua dimensi.

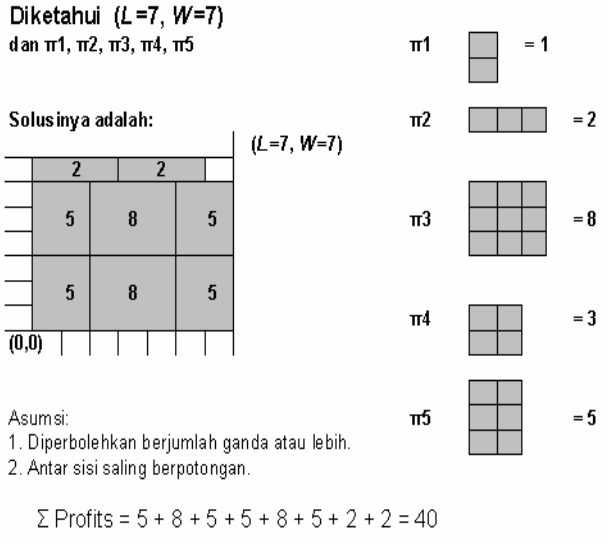
Dalam pengelolaan bahan baku pada industri manufaktur sering terjadi proses pemotongan bahan baku menjadi beberapa bagian untuk diproses lebih lanjut menjadi barang yang dibutuhkan konsumen. Saat ini, di sebagian besar industry manufaktur, masalah pemotongan bahan baku masih dikerjakan secara manual hanya mengandalkan pengalaman operator dalam menentukan pola pemotongan (*cutting pattern*). Hasil pemotongan dengan cara manual ini tidak efisien karena menghasilkan sisa bahan baku yang banyak. Padahal, industri manufaktur harus menghasilkan barang dengan harga jual yang bersaing dengan keuntungan yang sebesar-besarnya. Karena mahalnya harga bahan baku, maka perlu dicari suatu solusi masalah *cutting stock*. Solusi ini nantinya diharapkan dapat mengurangi biaya produksi.



Gambar 1.4
(a) Tipe Pemotongan *Guillotine*.
(b) Tipe Pemotongan *Non Guillotine.*

Hal ini sering menjadi faktor penting yang mendorong manusia untuk melakukan optimasi dalam sisa pemotongan bahan baku. Permasalahan optimasi sisa pemotongan dikenal sebagai masalah knapsack. Sebuah masalah knapsack memerlukan proses pencarian sebuah subset dari kumpulan objek dengan tujuan yaitu memaksimumkan jumlah dari keuntungan objek yang diinginkan dan tidak boleh melebihi ukuran dari knapsack atau jika diterapkan pada industri manufaktur, yakni rencana dalam industri itu sendiri. Sejumlah masalah muncul dalam bidang ilmu komputer dan riset operasional, yaitu aplikasi ”*cutting stock problem*”. Masalah 2D sangat menarik untuk dipelajari lebih jauh.

Masalah knapsack 0-1 sudah dapat diselesaikan secara efisien dengan *linier systolic array*.
Beberapa algoritma knapsack 2D memperhatikan sejumlah batasan masalah yang dikenal, dan masalah-masalah ini akan mudah untuk diselesaikan dengan menambahkan batasan-batasan atau menggunakan *approximation method*. Masalah yang dianalisis di sini termasuk dalam kelompok NP, dimana solusi terbaik hasil komputasi secara sekuensial adalah $O(LW(n+L+W))$, dimana n adalah jumlah objek, dan L dan W adalah dimensi dari knapsack. *Running time* untuk algoritma ini adalah *pseudo-polynomial* karena dalam kaitan dengan ukuran input, kapasitas knapsack dikodekan hanya dalam $\log2(L) + \log2(W)$ bit.



Gambar 1.5 Sebuah solusi untuk masalah knapsack sederhana.

Pada perusahaan yang memproduksi perabotan yang menggunakan bahan baku berupa kayu lapis untuk membuat perlengkapan rumah tangga dan kantor seperti meja, kursi, dan almari akan menghadapi suatu kejadian untuk merencanakan pemotongan bahan baku kayu lapis. Selain itu, perusahaan juga melayani pemesanan barang sesuai dengan keinginan konsumen, seperti bentuk dan ukuran barang yang dipesan. Permasalahan-permasalahan yang sering terjadi adalah:

- Kesalahan-kesalahan pemotongan bahan baku yang menyebabkan kerugian bagi pihak perusahaan.
- Kekeliruan perhitungan dalam proses perencanaan produksi, yakni pemotongan bahan baku mengakibatkan pemakaian bahan baku yang tidak optimal dan keuntungan berkurang.

II. DASAR TEORI

Program dinamis adalah suatu teknik matematis yang biasanya digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Dalam hal ini program dinamis menyediakan prosedur sistematis untuk menentukan kombinasi keputusan yang optimal. Tujuan utama model ini ialah untuk mempermudah penyelesaian persoalan optimasi yang mempunyai karakteristik tertentu.
Tidak seperti pemrograman linier, tidak ada bentuk matematis standar untuk perumusan pemrograman dinamis. Akan tetapi, pemrograman dinamis adalah pendekatan umum untuk pemecahan masalah dan persamaan tertentu yang digunakan di dalamnya harus dibentuk sesuai dengan situasi masalah yang dihadapi.
Istilah yang biasa digunakan antara lain:

1. Tahap adalah bagian persoalan yang mengandung *decision variable*.
2. Alternatif, pada setiap tahap terdapat *decision variable* dan fungsi tujuan yang menentukan besarnya nilai setiap alternatif.
3. Status menunjukkan kaitan satu tahap dengan tahap lainnya, diatur sedemikian rupa sehingga setiap tahap dapat dioptimisasikan secara terpisah sehingga hasil optimasi layak untuk seluruh persoalan.

III. PENYELESAIAN MASALAH

3.1 ANALISIS MASALAH

Masalah Knapsack 2D

Masalah *knapsack 2D* merupakan masalah pengisian daerah berdimensi (L, W) dengan n buah persegi dengan ukuran (li, wi) dimana i = 1, 2, ..., n. Profit adalah suatu nilai yang positif, π1, π1, ..., πn yang berkaitan dengan tiap persegi. Dengan parameter ini, keuntungan maksimum dari π1z1, π1z2, ..., πnzn dihitung dan zi adalah suatu nilai positif dimana knapsack dibagi dalam zi bentuk persegi i, yang mempunyai ukuran (li, wi).
Masalah pemotongan ini hanya mengizinkan terjadinya proses rekursi sisi per sisi, sehingga semua pemotongan dibuat tegak lurus dari satu sisi persegi terhadap yang lainnya. Objek benda dapat mempunyai orientasi yang sudah tepat atau dapat diputar 90 derajat. Tambahan n objek dengan dimensi (wi, li) dengan keuntungan π1 ditambahkan ketika proses rotasi memungkinkan. Salah satu algoritma untuk menyelesaikan masalah ini adalah dengan menerapkan *dynamic programming*.
Fungsi knapsack $F(x, y)$, didapatkan dari teknik *dynamic programming*, dilakukan proses perhitungan sehingga untuk lokasi (x, y), $F(x, y)$ adalah keuntungan terbesar

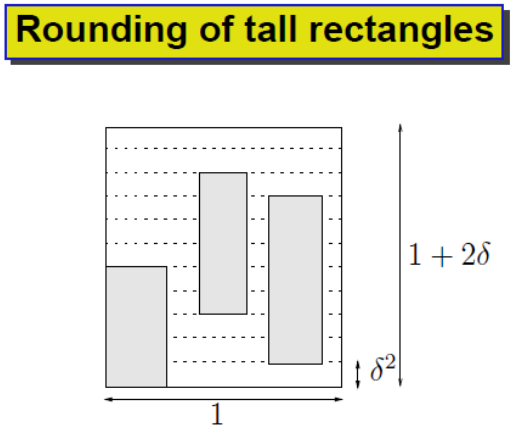
yang diperoleh dari persegi yang dibuat dari sisi *X* dan *Y* dan titik (*x*, *y*). Hal ini memenuhi pertidaksamaan matamatis berikut ini yang berhubungan dengan batasan pemotongan yang dapat dilakukan.

$$0 \le x \le L,$$
$$0 \le y \le W,$$
$$F(x, y) \ge 0,$$
$$F(x1 + x2, y) \ge F(x1, y) + F(x2, y),$$
$$F(x, y1 + y2) \ge F(x, y1) + F(x, y2),$$
$$F(li, wi) \ge \pi i, (i = 1, 2, \ldots, n)$$

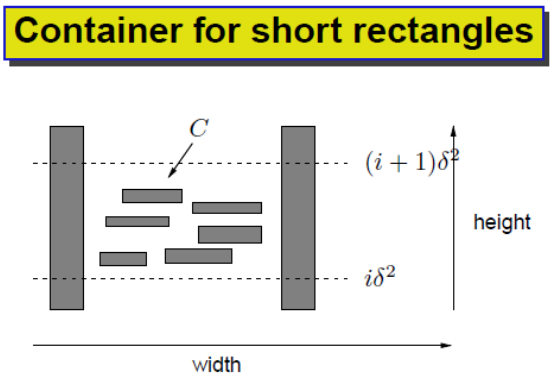
Untuk memaksimalkan penggunaan bahan baku, algoritma bersarang dibutuhkan dan algoritma tersebut diharapkan mampu mengecek suatu daerah yang 'gelap' atau belum terisi, untuk masalah pemotongan bahan adalah daerah yang belum digunakan/dipotong.

Ide untuk masalah 2D *knapsack problem*:

1. Hilangkan anggota himpunan potongan yang bernilai murah.
2. Potong bahan baku dengan bentuk memanjang atau memendek dan lebar atau sempit.
3. Gerakkan masing-masing potongan secara vertikal.



Gambar 2.1 Pemosisian potongan bahan baku.



Gambar 2.2 Peletakan potongan yang pendek dan lebar pada *container*

Ide lebih lanjut adalah:

1. Gabungkan potongan yang pendek dan lebar dalam satu *container*.
2. Tentukan nilai K secara ketinggian dengan keuntungan terbesar.
3. Lakukan perhitungan untuk ketinggian K dan *container*.
4. Lakukan secara *linier programming* untuk melakukan penempatan potongan.

3.2 ANALISIS ALGORITMA SEQUENTIAL DYNAMIC PROGRAMMING

Sequential Dynamic Programming adalah suatu teknik matematis yang biasanya digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Tujuan utama model ini adalah untuk mempermudah penyelesaiann persoalan optimasi yang mempunyai karakteristik tertentu.
Ide dasar *dynamic programming* ini adalah membagi persoalan menjadi beberapa bagian yang lebih kecil sehingga memudahkan penyelesaiannya. Akan tetapi, berbeda dengan *linear programming*, pada persoalan *dynamic programming* ini tidak ada formulasi matematis yang standar. Karena itu, persamaan-persamaan yang terpilih untuk digunakan harus dikembangkan agar dapat memenuhi masing-masing situasi yang dihadapi. Dengan demikian, maka antara persoalan yang satu dengan persoalan lainnya dapat mempunyai struktur penyelesaian persoalan yang berbeda. Keuntungan dari penggunaan *dynamic programming* adalah dapat memperoleh solusi dari suatu masalah tanpa adanya *exponential running time*.
Setiap objek berbentuk kotak akan diatur posisinya, dan sebuah tabel akan mencatat pemotongan terbaik untuk tiap lokasi. Setelah itu, semua isi tabel akan dibaca untuk menghitung jumlah panjang dan lebar dari obyek untuk membuat konfigurasi dari obyek yang menghasilkan keuntungan maksimum. Algoritma dari *sequential dynamic programming* dapat dilihat pada Segmen 1. Berikut ini adalah persamaan matematis relasi yang digunakan untuk mendapatkan algoritma *sequential*.

$$F0\ (x, y) = \max\ \{0, \pi j \mid lj \le x\ \Lambda\ wj \le y\}$$
$$Fk\ (x, y) = \max\ \{\ Fk\text{-}1\ (x, y), Fk\text{-}1\ (x1, y) + Fk\text{-}1\ (x2, y),$$
$$Fk\text{-}1\ (x, y1) + Fk\text{-}1\ (x, y2)\ \}$$
$$0 < x1 \le x2,\ x1 + x2 \le x,\ 0 < y1 \le y2,\ y1 + y2 \le y.$$

Langkah-langkah penyelesaian *cutting stock problem* dengan *dynamic programming* adalah:
Pada step 1 algoritma, semua lokasi dalam knapsack diinisialisasi dengan nilai 0.
Pada step 2 ini, setiap obyek mulai diperhatikan, meletakkan nilai keuntungan tertinggi pada semua lokasi dimana memungkinkan akan dilakukan.

Pada step 3 ini dilakukan pembacaan isi tabel 2D dari baris yang paling rendah ke baris yang paling tinggi, menjumlahkan semua kemungkinan kombinasi yang dapat dilakukan baik secara vertikal atau horisontal dan menyimpan dua obyek yang mempunyai jumlah keuntungan tertinggi. Pada segmen 1 ditunjukkan bagaimana nilai $F(x, y)$ dihitung secara iterasi. Karena hanya potongan persegi yang digunakan, solusi parsial pada posisi (i, j) dipotong secara paralel menjadi dua bagian, x dan y. Karena simetris, hanya potongan x, mulai dari $x = 0$ sampai $i/2$, dan potongan y mulai dari 0 sampai $j/2$ yang dipertimbangkan untuk setiap (i, j). Algoritma *sequential* ini mempunyai waktu proses $O(LW(n+L+W))$, dimana n adalah jumlah obyek, dan L, W menyatakan dimensi dari knapsack.
Secara sederhana,

```
For i = 0 to L                          // Step 1
    For j = 0 to w
        F[ i,j ] = 0
For i = 0 to n                          // Step 2
    For j = 0 to L
        For j = 0 to w
            If (Ik <= j and Wk <= j and πk > F[ i,j ] )
                F[ i,j ] = πk
For i = 0 to L                          // Step 3
    For j = 0 to w
        For k = 0 to [ i/2 ]
            Sum = F[ k,j ] + F[ i-k,j ]
            If ( sum > F[ i,j ] )
                F[ i,j ] = sum
        For k = 0 to [ j/2 ]
            Sum = F[ k,i ] + F[ i,j-k ]
            If ( sum > F[ i,j ] )
                F[ i,j ] = sum
```

IV. KESIMPULAN

Cutting Stock Problem dikategorikan ke dalam *NP-hard problems.* Permasalahan yang dibahas pada makalah ini dapat dikategorikan sebagai *cutting stock problem.* Waktu proses yang diperlukan untuk model sequential adalah $O(LW(n + L + W))$.

V. SARAN

Penulis memberi saran agar dalam penggunaan algoritma program dinamis sekuensial dapat menggunakan proses secara parallel. Karena untuk mengurangi *running time*, dapat dilakukan proses secara parallel. Dan untuk menjamin bahwa kita akan mendapatkan solusi optimal, perlu dibangun model integer programming yang akan digunakan untuk mencari jumlah bahan baku tambahan yang dibutuhkan untuk memenuhi kekurangan permintaan. Masalah ini juga perlu diselesaikan dengan memperhitungkan masalah bilangan bulat. Jika segala hal yang dipaparkan sebelumnya telah digabungkan dengan algoritma program dinamis sekuensial yang telah dijelaskan diatas, masalah pemotongan bahan baku akan dapat diselesaikan dengan optimal dan dapat digunakan dengan baik.

VI. REFERENSI

[1] Hopper, E.; Turton, B., *A Review of The Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems.pdf*

[2] Bortfeldt, Andreas; Winter, Tobias, *A Genetic Algorithm for the Two-Dimensional Knapsack Problem with Rectangular Pieces*, May 2008.

[3] Jansen, Klaus, *Approximation Algorithms for the 2-Dimensional Packing Problem – Universitat Kiel* .

[4] Dyckhoff, Harald, *A Typology of Cutting And Packing Problems*, Aachen.

[5] Ulm, Darrell; Baker, Johnnie; Scherger, Michael, *Solving 2-D Knapsack Problem Using a Hybrid Data-Parallel/Control Style of Computing*, USA.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2011

Dion Jogi Parlinggoman
13509045