

# Penerapan Algoritma Greedy dan *Brute Force* untuk Mengoptimalkan Jumlah Penggunaan Triplek Bekisting dalam Proses Konstruksi Bangunan

Cornelius Yan Mintareja, 13516113<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13516113@std.stei.itb.ac.id, onel.raja@gmail.com

**Abstrak** – Pemotongan triplek bekisting merupakan hal yang sering dilakukan dalam proses konstruksi bangunan. Namun demikian, para perancang konstruksi bangunan biasanya tidak terlalu memikirkan secara rinci hal tersebut. Mereka biasanya merasa cukup menggunakan metode yang sederhana dalam melakukan pemotongan triplek tersebut. Padahal, dengan melakukan optimasi pada cara pemotongan triplek yang tepat sesuai kebutuhan tersebut, kita dapat menghemat jumlah triplek yang digunakan secara cukup besar. Salah satu cara melakukan optimasi jumlah penggunaan triplek bekisting tersebut adalah dengan algoritma greedy dan brute force. Dengan melakukan optimasi tersebut, seorang perancang konstruksi bangunan akan dapat menghemat jumlah penggunaan triplek dan mengurangi anggaran proses konstruksi bangunan secara keseluruhan.

**Kata kunci** — Triplek, Bekisting, Greedy, Brute Force, Optimal.

## I. PENDAHULUAN

Bekisting adalah konstruksi sementara yang berupa cetakan untuk menentukan bentuk dari konstruksi beton pada saat beton masih segar. Dalam proses konstruksi bangunan, bekisting merupakan salah satu hal yang sangat penting pada tahap pengecoran beton. Terdapat berbagai jenis bekisting yang biasanya digunakan dalam konstruksi bangunan, diantaranya bekisting sistem dan bekisting konvensional. Hingga sekarang, bekisting konvensional lah yang lebih umum digunakan karena harganya yang relatif lebih murah dan sederhana dibandingkan bekisting sistem yang modern namun mahal, sulit didapat, serta membutuhkan keahlian dan peralatan berat. Ketika menggunakan bekisting konvensional, maka triplek lah yang biasanya dipilih sebagai bahan dari bekisting tersebut.

Untuk membuat bekisting konvensional tersebut, triplek perlu dipotong-potong sesuai dengan rancangan yang telah diperhitungkan. Hal-hal yang diperhitungkan diantaranya panjang, lebar, dan tinggi dari bekisting. Dalam proses pemotongan triplek tersebut, terdapat suatu masalah utama. Setiap jenis triplek memiliki ukuran selembar yang tetap untuk panjang dan lebarnya, yaitu 244 x 122 cm. Dikarenakan kebutuhan yang diperlukan (>20 meter) jauh lebih besar daripada selembar triplek, maka diperlukan cukup banyak lembaran triplek pada setiap pembuatan bekisting. Berbeda dengan ukuran dasar triplek awal yang tetap, ukuran bekisting

yang diinginkan tentunya berubah-ubah sesuai kebutuhan. Oleh karena itu, diperlukan perhitungan-perhitungan yang berbeda pula untuk setiap pembuatan bekisting.

Pada setiap saat proses pemotongan triplek, akan terdapat bagian sisa yang tidak digunakan karena kombinasi panjang & lebar bekisting yang diperlukan tidak dapat menepati 244 x 122 cm. Namun demikian, para perancang konstruksi bangunan biasanya tidak terlalu mempedulikan dan menghargai sisa triplek yang terbuang tersebut. Mereka biasanya tidak memiliki waktu untuk menghitung kombinasi potongan optimal untuk meminimalkan sisa triplek yang terbuang tersebut dan lebih memilih mengira-ngira saja jumlah triplek yang dibutuhkan, menambahkan batas error, dan membeli berlebih dari jumlah perhitungan tersebut. Padahal, harga triplek tersebut tidak murah, sehingga semakin banyak sisa triplek yang tidak terpakai, semakin banyak pula anggaran yang disia-siakan. Dari masalah itulah, perlu dibuat sebuah program penghitung kombinasi pemotongan optimal agar sisa triplek yang terbuang pun minimal, mengurangi biaya dari proyek konstruksi secara keseluruhan.

Untuk menyelesaikan persoalan optimasi tersebut, dapat digunakan algoritma greedy serta sedikit unsur *brute force*. Dengan menganggap meminimalkan sisa triplek pada proses pemotongan sebagai fungsi obyektif algoritma greedy, maka persoalan optimasi jumlah penggunaan triplek bekisting ini dapat diselesaikan.

## II. LANDASAN TEORI

### A. Algoritma Greedy

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Persoalan optimasi adalah persoalan mencari solusi optimum. Ada dua persoalan optimasi yaitu maksimasi dan minimasi.

Prinsip greedy adalah “take what you can get now!”, yang berarti algoritma ini membentuk solusi langkah per langkah namun pada setiap langkah terdapat banyak pilihan yang perlu dievaluasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang sudah diambil pada suatu langkah tidak dapat

diubah/diperhitungkan kembali dalam menentukan pilihan selanjutnya.

Pada setiap langkah, kita membuat pilihan optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Elemen-elemen algoritma greedy yaitu :

1. Himpunan kandidat,  $C$ .  
Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi,  $S$ .  
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi  
Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal.
4. Fungsi kelayakan (*feasible*)  
Fungsi yang memeriksa apakah suatu kandidat yang telah terpilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar batasan/kendala yang ada.
5. Fungsi obyektif  
Fungsi yang memaksimalkan atau meminimumkan nilai solusi.

Dengan kata lain algoritma greedy melibatkan pencarian sebuah himpunan bagian,  $S$ , dari himpunan kandidat  $C$  yang dalam hal ini,  $S$  harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan  $S$  dioptimasi oleh fungsi obyektif. [1]

Beberapa persoalan klasik yang dapat diselesaikan dengan algoritma greedy yaitu persoalan penukaran uang (*coin change*) untuk kasus spesial, pengimbangan berat (*load balancing*), dan penutup interval (*interval covering*). Algoritma greedy juga digunakan dalam penyelesaian algoritma Kruskal dan Prim untuk menyelesaikan persoalan pohon rentang minimum (*minimum spanning tree*).

## B. Algoritma Brute Force

*Brute force* merupakan suatu metode pemecahan masalah yang paling sederhana dan paling mudah untuk diimplementasikan. Istilah lain yang sering dianggap sama yaitu *exhaustive search* dan *complete search*. *Brute force* merupakan suatu pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah. Hal tersebut biasanya didasarkan langsung pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma ini memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas (*obvious way*). [1]

Algoritma *brute force* melakukan penelusuran terhadap seluruh kemungkinan jawaban untuk mendapatkan solusi yang diinginkan. Dalam pencarian tersebut, kita dapat melakukan pemangkasan (yaitu, memilih untuk tidak ditelusuri) pada bagian-bagian dalam pencarian jika kita telah menentukan

bahwa bagian-bagian tersebut tidak memiliki kemungkinan untuk mengandung solusi yang dibutuhkan.

Dari cara tersebut, dapat terlihat bahwa algoritma *brute force* membutuhkan jumlah langkah yang besar dalam penyelesaiannya, terutama bila masalah yang dipecahkan berukuran besar (dalam hal ini ukuran masukannya), sehingga algoritma *brute force* terlihat tidak mangkus dan tidak cerdas. Untuk masalah yang ukurannya kecil, kesederhanaan *brute force* biasanya lebih diperhitungkan daripada ketidakmangkusannya.

Meskipun telah disebutkan bahwa algoritma *brute force* tidak mangkus, algoritma ini dapat diterapkan pada sebagian besar masalah. Sulit untuk mencari suatu masalah yang tidak dapat dipecahkan dengan algoritma *brute force*. Bahkan, terdapat masalah yang hanya dapat dipecahkan secara *brute force*. Algoritma *brute force* juga seringkali lebih mudah diimplementasikan daripada algoritma yang lebih canggih. Dikarenakan kesederhanaannya tersebut, terkadang algoritma *brute force* dapat menjadi lebih mangkus ditinjau dari segi implementasinya.

## C. Bekisting

Bekisting (*formwork*) adalah konstruksi bersifat sementara yang merupakan cetakan untuk menentukan bentuk dari konstruksi beton pada saat beton masih segar. Menurut Stephens (1985), bekisting adalah cetakan sementara yang digunakan untuk menahan beton selama beton dituang dan dibentuk sesuai dengan bentuk yang diinginkan. Dikarenakan berfungsi sebagai cetakan sementara, bekisting akan dilepas kembali apabila beton yang dituang telah mencapai kekuatan yang cukup. Meskipun bersifat sementara, pembuatan bekisting harus benar dan tepat supaya diperoleh bangunan yang berkualitas.

Hingga saat ini, terdapat beberapa jenis bekisting beserta kelebihanannya masing-masing, yaitu [3] :

### 1. Bekisting konvensional

Bekisting konvensional merupakan jenis bekisting yang pertama kali dikenal. Bekisting konvensional atau tradisional ini hanya mengandalkan triplek dan kayu atau papan. Jenis papan yang dipakai biasanya adalah papan yang tahan kelembaban. Bekisting konvensional ini terbilang murah bila dibandingkan dengan pengadaan atau penyewaan bekisting modern.

### 2. Bekisting Knock Down

Bekisting *knock down* merupakan salah satu jenis bekisting yang baru / modern. Sistem bekisting *knock down* ini menggunakan bahan besi *hollow* dan plat baja. Penggunaan material tersebut tentunya akan menghasilkan bentuk yang lebih presisi jika dibandingkan dengan penggunaan triplek dan papan pada sistem bekisting konvensional.

### 3. Bekisting Fiberglass

Jenis bekisting modern yang terakhir ialah bekisting *fiberglass*. Bekisting yang terbuat dari bahan *fiberglass* ini tahan terhadap air, sehingga sangat cocok dipakai pada konstruksi di bawah tanah. Selain itu, bahan

fiberglass tidak mudah berkarat, ramah lingkungan, ringan, mudah dibersihkan, mudah dipasang, dan juga mudah dilepas.

Menurut *Blake (1975)*, ada beberapa aspek yang harus diperhatikan pada pemakaian bekisting dalam suatu pekerjaan konstruksi beton. Aspek-aspek tersebut ialah :

1. Aspek kualitas

Aspek pertama adalah kualitas bekisting yang akan digunakan harus tepat dan layak serta sesuai dengan bentuk pekerjaan struktur yang akan dikerjakan. Permukaan bekisting yang akan digunakan harus rata sehingga hasil permukaan beton baik.

2. Aspek keamanan

Aspek kedua adalah keamanan bagi pekerja konstruksi tersebut. Oleh karena itu, bekisting harus cukup kuat menahan beton agar beton tidak runtuh dan mendatangkan bahaya bagi pekerja sekitarnya.

3. Aspek biaya

Aspek yang ketiga adalah biaya pemakaian bekisting yang harus direncanakan seekonomis mungkin.



**Gambar 1** – Bekisting konvensional dengan bahan Triplek yang telah diisi dengan beton [4]

D. Triplek

*Plywood*, atau yang dalam bahasa Indonesia biasanya disebut kayu lapis, adalah produk olahan kayu yang terdiri dari lembaran-lembaran kayu yang direkatkan menjadi satu dengan tekanan tinggi menggunakan lem khusus. *Plywood* yang terdiri dari tiga lapisan disebut dengan triplek, sedangkan yang terdiri dari lebih banyak lapisan disebut multiplek. Triplek dan multiplek dapat digunakan dalam banyak hal. Triplek yang tebal dapat digunakan untuk bekisting, selain itu dapat juga sebagai bahan membuat *furniture*, material *plafond*, partisi dinding sementara, dan daun pintu. [5]

Sama seperti bahan material lainnya, triplek juga memiliki jenis dan ukuran yang beragam. Setiap jenis triplek tersebut memiliki karakteristik yang berbeda-beda sehingga jenis barang/material yang ingin dibuat akan menentukan jenis triplek yang digunakan. Ada beberapa jenis triplek atau multiplek yang beredar di pasaran saat ini, diantaranya yaitu triplek *soft wood*, triplek *hard wood*, triplek *partikel board*, triplek *blockboard*,

triplek MDF (*Medium Density Fiberboard*), dan masih banyak triplek-triplek jenis lainnya lagi. Karena keragaman jenis dan bahan tersebut, masing-masing jenis triplek juga memiliki harga yang berbeda-beda.

Dalam kategori jenis bekisting, triplek termasuk kedalam jenis bekisting konvensional. Semakin baik jenis triplek yang digunakan, maka akan semakin baik pula hasil cetakan beton yang dihasilkan. Oleh karena itu, salah satu jenis triplek baik yang cocok digunakan untuk bekisting adalah triplek jenis *soft wood*. Ukuran triplek *soft wood* yang beredar di pasaran memiliki ukuran yang selalu sama, yaitu 122 x 244 cm. Triplek jenis ini memiliki bahan yang lebih lunak dan biasanya berbahan dasar kayu albasia/jinging, dimana populasinya banyak ditemukan di daerah Jawa Barat. Berikut adalah daftar harga triplek Albasia ukuran 122x244 cm yang biasanya digunakan untuk membuat bekisting :

Albasia 122 x 244cm (berdasarkan ketebalan)	
3mm	Rp 52.000
4mm	Rp 82.000
8mm	Rp 87.000
9mm	Rp 124.000
12mm	Rp 155.000
15mm	Rp 176.000
18mm	Rp 167.500

**Gambar 2** – Daftar harga triplek Albasia berukuran 122x244 cm berdasarkan ketebalannya pada Mei 2018 [6]

III. ANALISIS PEMECAHAN MASALAH

A. Spesifikasi Kondisi Masalah

Berikut adalah beberapa spesifikasi dari masalah yang akan dipecahkan pada makalah ini :

1. Bekisting terbuat dari triplek konvensional berjenis Albasia
2. Ukuran triplek yang digunakan selalu tetap, yaitu 122 x 244 cm
3. Ketebalan triplek yang dipilih adalah 9 mm
4. Tinggi balok 42 cm
5. Lebar balok 16 cm
6. Panjang balok 70 m
7. Dikarenakan sebuah bekisting terdiri dari 2 sisi tinggi (sedangkan sisi lebar hanya dibutuhkan 1 saja karena bagian atasnya terbuka), sehingga panjang triplek berukuran tinggi balok yang dibutuhkan adalah 2 x 70 = 140 m (triplek berukuran lebar balok tetap 70 m)
8. Harga sel lembar triplek Albasia 122 x 244 cm dengan tebal 9 mm adalah Rp 124.000

Spesifikasi tersebut tentu saja hanya berupa sebuah contoh kasus. Pada kenyataannya, tinggi, lebar, dan panjang balok sangat beragam. Jenis bekisting beserta ketebalannya pun dapat

berbeda-beda. Namun demikian, jika menggunakan bekisting triplek, maka hampir semuanya berukuran 122 x 244 cm.



**Gambar 4** – Bekisting terdiri dari 2 sisi tinggi dan 1 sisi lebar [7]

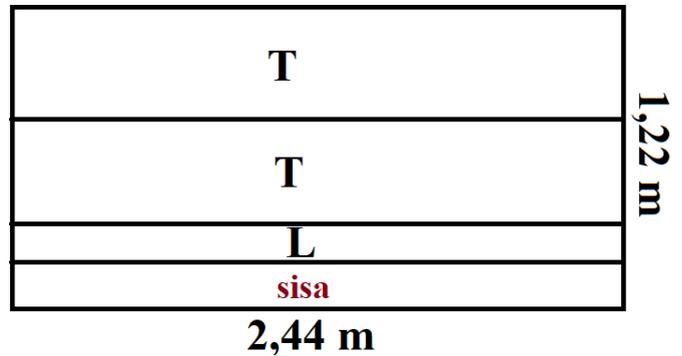
**B. Pemecahan dengan Cara Sederhana**

Pada umumnya, para pihak perancang proyek konstruksi bangunan tidak terlalu memperhatikan secara rinci masalah pemotongan triplek bekisting. Mereka tidak terlalu mepedulikan dan menghargai sisa triplek yang terbuang tersebut. Mereka biasanya tidak memiliki waktu untuk menghitung kombinasi potongan optimal untuk meminimalkan sisa triplek yang terbuang tersebut dan lebih memilih mengira-ngira saja jumlah triplek yang dibutuhkan, menambahkan batas error, dan membeli berlebih dari jumlah perhitungan tersebut. Oleh karena itu, mereka hanya menggunakan cara yang sangat sederhana untuk merancang cara pemotongan triplek & menghitung jumlah triplek yang diperlukan. Berikut langkah-langkah yang biasanya mereka lakukan :

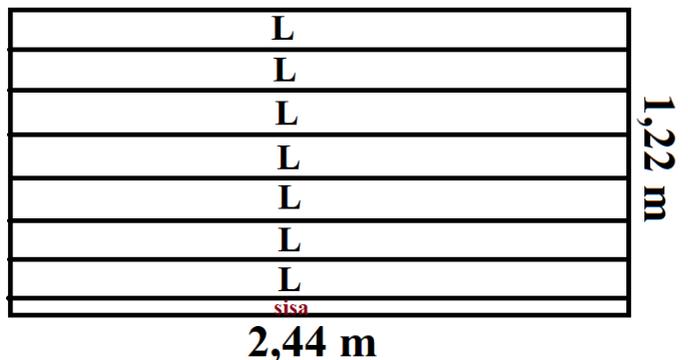
(T untuk bagian tinggi balok, L untuk bagian lebar balok)

1. Hitung total T yang dapat dibuat pada selembar triplek dengan cara menghitung  $\left\lfloor \frac{TinggiTriplek}{TinggiBalok} \right\rfloor$ . Pada kasus ini,  $\left\lfloor \frac{112cm}{42cm} \right\rfloor = 2 T$
2. Gambarkan sketsa seluruh potongan tinggi tersebut pada triplek. Jika masih ada sisa, cek apakah sisa tersebut dapat diisi L. Total L yang dapat diisi yaitu  $\left\lfloor \frac{(TinggiTriplek - \left\lfloor \frac{TinggiTriplek}{TinggiBalok} \right\rfloor * TinggiBalok)}{LebarBalok} \right\rfloor$ . Dalam kasus ini,  $\left\lfloor \frac{(112cm - [112cm/42cm] * 42cm)}{16cm} \right\rfloor = 1 L$
3. Buatlah bentuk tersebut secara terus menerus hingga total panjang T yang didapat (LebarTriplek \* Jumlah T \* Jumlah Triplek) sudah mencapai PanjangBalok. Pada kasus ini yaitu  $(2,44 * 2 * 30) = 146,4 > 140$ . Jadi, total triplek yang sudah diperlukan yaitu 30 lembar.
4. Hitung juga total panjang L yang telah didapat jika pada pola sebelumnya terdapat potongan L juga. Kemudian, hitung total L yang dapat dibuat pada selembar triplek jenis selanjutnya dengan cara menghitung  $\left\lfloor \frac{TinggiTriplek}{LebarBalok} \right\rfloor$ . Pada kasus ini,  $\left\lfloor \frac{112cm}{16cm} \right\rfloor = 7 L$

5. Buatlah bentuk tersebut secara terus menerus hingga total panjang L yang didapat (LebarTriplek \* Jumlah L \* Jumlah Triplek) sudah mencapai PanjangBalok yang tersisa. Pada kasus ini, sudah tidak perlu membuat triplek dengan jenis tersebut karena jumlah L yang dibutuhkan sudah melebihi kebutuhan ( $30 * 2,44 > 70$ ). Oleh karena itu, total triplek yang diperlukan untuk membuat jenis ini ada 0 lembar.
6. Hitung total triplek yang dibutuhkan. Pada kasus ini yaitu  $30 \text{ lembar} + 0 \text{ lembar} = 30 \text{ lembar triplek}$ .



**Gambar 4** – Triplek jenis pertama, berisi 2 T dan 1 L, dibuat sebanyak 30 lembar



**Gambar 5** – Triplek jenis kedua, berisi 7 L, namun sudah tidak perlu dibuat

Cara ini disebut sederhana karena merupakan cara yang tidak melakukan perhitungan optimasi terlebih dahulu. Orang-orang biasanya akan secara spontan melakukan perhitungan dengan cara tersebut. Bahkan, terkadang ada beberapa perancang konstruksi bangunan yang menggunakan cara yang lebih sederhana dari cara tersebut dan menambahkan batas error sehingga perlu menambah jumlah triplek yang dibutuhkan.

Pemecahan masalah secara sederhana tersebut sebenarnya sudah mengandung prinsip greedy. Namun demikian, terdapat suatu *kecacatan* pada pemecahannya, yaitu pada bagian pemilihan fungsi seleksinya. Pembahasan lebih lanjut mengenai pemecahan masalah secara greedy akan dibahas pada sub bab selanjutnya.

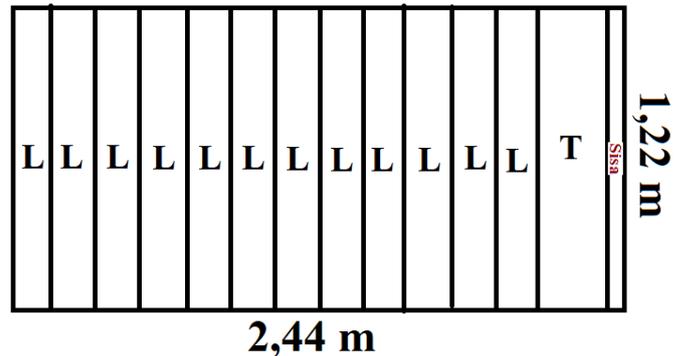
C. Pemecahan dengan cara greedy

Permasalahan triplek bekisting ini dapat dipecahkan dengan algoritma greedy karena terdapat solusi langkah per langkah yang tepat untuk menyelesaikan solusi secara global. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Berikut penerapan elemen-elemen greedy pada permasalahan optimasi triplek bekisting ini :

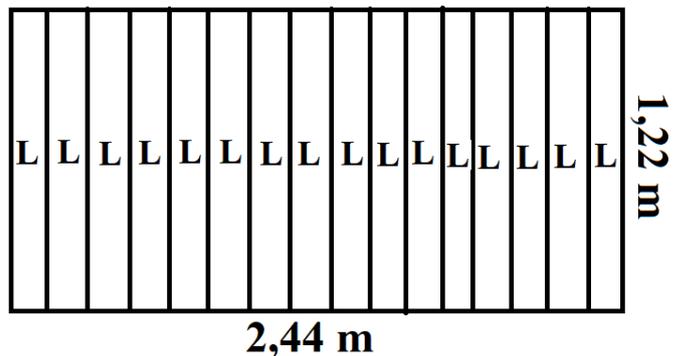
1. Himpunan kandidat, C  
Himpunan kandidat dalam permasalahan ini yaitu kombinasi-kombinasi dari pemotongan triplek bekisting dengan berbagai pola/jenis. Pola/jenis yang dihasilkan dapat lebih dari 1, tidak mesti hanya sebuah pola tetap saja.
2. Himpunan solusi, S  
Himpunan solusi dalam permasalahan ini yaitu kombinasi-kombinasi potongan triplek bekisting yang total potongan tinggi balok (T) dan potongan lebar balok (L) nya sudah lebih besar sama dengan dari panjang balok yang dibutuhkan.
3. Fungsi seleksi  
Fungsi seleksi dalam permasalahan ini tujuan utamanya adalah mencari kombinasi pemotongan triplek yang paling optimum. Fungsi seleksi ini juga dibagi menjadi 3 kasus, yaitu :
  - i. Kasus jika jumlah potongan tinggi masih tersisa cukup banyak (tidak cukup dipenuhi hanya dengan 1 lembar triplek). Cara menyelesaikan masalah kasus pertama ini yaitu dengan metode *brute force*. Metode *brute force* tersebut akan dijelaskan lebih lanjut pada sub bab berikutnya.
  - ii. Kasus jika jumlah potongan tinggi tersisa tinggal sedikit lagi (cukup hanya dengan 1 lembar triplek atau JumlahTriplek2 = 1). Cara menyelesaikan masalah kasus kedua ini yaitu dengan mengisikan jumlah sisa T ( $JumlahSisaT = [(PanjangTriplek - \{JumlahTriplekJenisSebelumnya * TinggiBalok\}) / TinggiTriplek]$ ) yang masih kurang ke karton terakhir tersebut. Dalam kasus ini,  $JumlahSisaT = [(70m - \{23 * 0,42m\}) / 1,22m] = 1$  T (cara mendapatkan nilai 23 dapat dilihat pada sub bab selanjutnya). Setelah mengisi T pada triplek tersebut, isilah sisa bagian triplek dengan L sebanyak-banyaknya, atau jika dengan rumus  $JumlahLperTriplek = [(TinggiKarton - JumlahSisaT * TinggiBalok) / LebarBalok]$ . Dalam kasus ini,  $JumlahLperTriplek = [(1,22 - 1 * 0,42) / 0,16] = 12$  L.
  - iii. Kasus jika jumlah potongan tinggi sudah cukup dan jumlah potongan lebar masih kurang. Cara menyelesaikan masalah kasus

ketiga ini yaitu dengan membuat triplek berisi L semaksimal mungkin. Jumlah L yang dapat diisi yaitu  $\lfloor \frac{PanjangTriplek}{LebarBalok} \rfloor$ . Pada kasus ini,  $\lfloor \frac{224cm}{16cm} \rfloor = 16$  L. Jumlah triplek jenis ini yang dibutuhkan bergantung terhadap sisa L yang diperlukan setelah melalui kasus i dan ii. Dalam kasus ini, karena jumlah L yang diperlukan sudah dipenuhi pada tahap i dan ii ( $(23 * 1,22 * 2 + 1 * 1,22 * 12) = 70,76m > 70m$ ), maka tidak perlu dibuat triplek berjenis ini lagi.

4. Fungsi kelayakan  
Suatu kandidat disebut layak, apabila jumlah panjang potongan tinggi dan jumlah panjang potongan lebar sudah melebihi permintaan yaitu sepanjang 2 kali panjang balok untuk T dan 1 kali panjang balok untuk L.
5. Fungsi obyektif  
Fungsi obyektif dari permasalahan ini adalah mencari solusi yang memiliki jumlah triplek bekisting dibutuhkan paling minimal.



Gambar 5 – Triplek jenis kedua, berisi 1 T dan 12 L, dibuat sebanyak 1 lembar

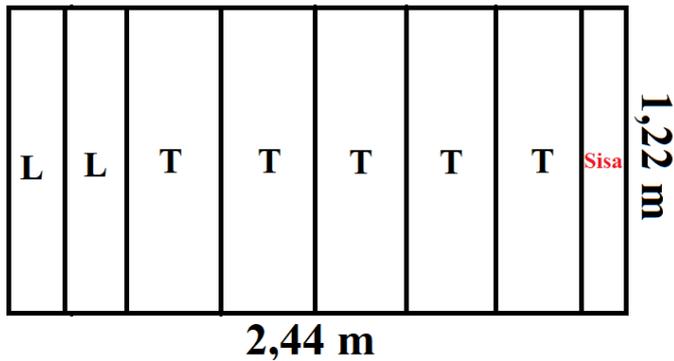


Gambar 5 – Triplek jenis ketiga, berisi 16 L, namun sudah tidak perlu dibuat

D. Pemecahan dengan cara *brute force*

Seperti yang telah disampaikan pada sub bab selanjutnya, metode *brute force* dalam permasalahan triplek bekisting ini merupakan salah satu bagian dari langkah fungsi seleksi algoritma greedy. *Brute force* dilakukan untuk mencari pola triplek optimal ketika T dan L masih sama-sama belum memenuhi kebutuhan. Dalam permasalahan ini, suatu pola disebut optimal jika dalam sebuah triplek tersebut terdapat bentuk T yang lebih banyak dari L namun memiliki luas sisa yang paling sedikit. Suatu pola jenis ini perlu memiliki T karena jumlah T 2 kali lebih banyak daripada L, sehingga jika triplek hanya mengandung L saja, maka besar kemungkinan sisa triplek jenis lain akan mengandung luas sisa yang jauh lebih besar sehingga total luas sisa keseluruhan menjadi lebih besar pula dan algoritma greedy menjadi tidak optimal.

Untuk mendapatkan pola dengan syarat tersebut, dilakukan *exhaustive search* pada triplek untuk mencari kombinasi isi T dan L yang tepat. *Exhaustive search* juga tidak hanya dilakukan untuk orientasi triplek vertikal, melainkan horizontal juga. Beberapa kandidat yang ditemukan yaitu TTTT horizontal, TTLLLL horizontal, TTL vertikal, dan sebagainya. Dari seluruh kombinasi yang ditemukan tersebut, ditemukan suatu pola yang paling optimal dan sesuai dengan syarat pada paragraf sebelumnya, yaitu TTTTLL horizontal (5 T dan 1 L). Jumlah triplek berpola tersebut yang dibutuhkan sebanyak  $\lfloor \frac{\text{PanjangBalokT}}{\text{JumlahT} * \text{TinggiTriplek}} \rfloor$ , yang dalam kasus ini sebanyak  $\lfloor \frac{140m}{(5 * 1,22m)} \rfloor = 23$  lembar triplek.



Gambar 6 – Triplek jenis pertama, berisi 5 T dan 2 L secara horizontal, dibuat sebanyak 23 lembar

IV. ANALISIS HASIL PENERAPAN ALGORITMA GREEDY DAN *BRUTE FORCE*

A. Analisis Keunggulan Cara Greedy dan *Brute Force* Dibandingkan Cara Sederhana

Tujuan utama dari optimasi jumlah triplek bekisting ini ialah ingin mengeluarkan jumlah triplek yang minimum. Selain faktor ramah lingkungan, penggunaan triplek yang lebih sedikit juga dapat menghemat anggaran yang tidak sedikit.

Pada contoh kasus di bab sebelumnya, total triplek yang dibutuhkan dengan cara sederhana adalah 30 lembar, sedangkan total triplek yang dibutuhkan dengan cara greedy dan *brute force* adalah 24 lembar. Oleh karena itu, jumlah triplek yang dihemat menjadi  $30 - 24 = 6$  lembar ! Dengan menggunakan harga 1 lembar triplek Albasia 122 x 244 cm ketebalan 9mm adalah Rp 124.000, maka dengan cara greedy dan *brute force* tersebut pada satu kasus ini saja, seorang perancang konstruksi bangunan dapat menghemat anggaran sebanyak  $6 * Rp 124.000 = Rp 744.000$  !

Total selisih harga, baru total sisa ceritain kenapa bisa beda jauh gitu

B. Analisis Perbedaan Jumlah Total Triplek Cara Greedy dan *Brute Force* dengan Cara Sederhana

Penyebab utama perbedaan jumlah triplek dengan kedua cara berbeda tersebut sangat besar adalah jumlah luas sisa tak terpakai yang dihasilkan pada setiap pemotongan karton. Cara sederhana akan selalu menghasilkan luas sisa yang lebih banyak dibandingkan dengan cara greedy dan *brute force* yang selalu optimal.

Pada kasus ini, jumlah luas sisa dari cara sederhana yaitu :

$$\{(122 - (2 * 42 + 1 * 16)) * 244\} * 30 = 16,104 m^2$$

Sedangkan jumlah luas sisa dari cara greedy dan *brute force* yaitu :

$$\begin{aligned} &\{(244 - (5 * 42 + 2 * 16)) * 122\} * 30 \\ &+ \{(244 - (1 * 42 + 12 * 16)) * 122\} * 1 \\ &= 0,854 m^2 \end{aligned}$$

Dari hasil perhitungan tersebut, terlihat bahwa jumlah luas triplek yang tidak terpakai dari cara sederhana lebih banyak seluas  $16,104 - 0,854 = 15,25 m^2$  ! Oleh karena itu, tidak aneh jika jumlah triplek yang dibutuhkan dengan cara sederhana lebih banyak 6 lembar dibandingkan dengan cara greedy dan *brute force*.

## V. KESIMPULAN

Proses pemotongan triplek bekisting merupakan hal yang biasanya tidak terlalu dipikirkan oleh para perancang konstruksi bangunan dan hanya menggunakan cara yang sederhana saja. Padahal, dengan melakukan optimasi pada cara pemotongan triplek yang tepat sesuai dengan kebutuhan, kita dapat menghemat jumlah triplek yang digunakan secara cukup besar. Salah satu cara melakukan optimasi jumlah penggunaan triplek bekisting tersebut adalah dengan algoritma greedy dan *brute force*. Selisih jumlah triplek yang digunakan tersebut dapat berbeda cukup jauh karena perbedaan jumlah luas sisa triplek yang tidak terpakai antara cara sederhana dengan cara greedy dan *brute force* pun terlihat sangat besar. Dengan mengambil suatu contoh kasus pembuatan balok bekisting dengan panjang 70 meter saja, penggunaan algoritma greedy dan *brute force* sudah dapat menghemat anggaran sekitar Rp 744.000.

## VI. UCAPAN TERIMA KASIH

Dengan mengucapkan puji dan syukur kepada Tuhan yang Maha Esa atas segala karunia dan anugrah-Nya, sehingga makalah dengan judul “Penerapan Algoritma Greedy dan *Brute Force* untuk Mengoptimalkan Jumlah Penggunaan Triplek Bekisting dalam Proses Konstruksi Bangunan” ini dapat terselesaikan tepat pada waktunya. Makalah ini dimaksudkan untuk memenuhi Tugas Makalah IF2211 Strategi Algoritma, Sem. II Tahun 2017/2018. Tak lupa saya juga mengucapkan terima kasih kepada ibu Masayu Leylia K., ibu Nur Ulfa M., serta Bapak Rinaldi Munir sebagai dosen-dosen dari mata kuliah Strategi Algoritma, atas bimbingan dan ilmu yang sudah diberikannya hingga makalah ini dapat diselesaikan.

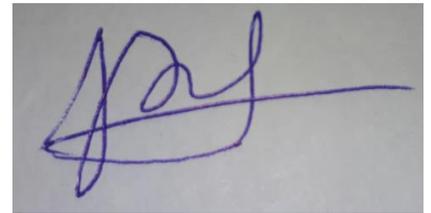
## SUMBER

- [1] Munir, Rinaldi. 2007. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung: Institut Teknologi Bandung.
- [2] <http://www.vedcmalang.com/pppptkboemlg/index.php/menutama/departemen-bangunan-30/1498-ubr> diakses tanggal 6 Mei 2018, pukul 19.00.
- [3] <http://strong-indonesia.com/artikel/jenis-bekisting-pekerjaan-beton/> diakses tanggal 6 Mei 2018, pukul 19.00.
- [4] <https://www.ilmutekniksipil.com/bekisting/bekisting-konvensional> diakses tanggal 6 Mei 2018, pukul 19.50
- [5] <https://www.mediabangunan.com/2013/05/daftar-harga-tripleks-dan-multipleks.html> diakses tanggal 6 Mei 2018, pukul 20.00
- [6] <http://hargaper.com/harga-triplek-terbaru.html> diakses tanggal 6 Mei 2018, pukul 20.00
- [7] <https://www.bangunrumah.name/pengertian-dan-jenis-bekisting/> diakses tanggal 6 Mei 2018, pukul 21.00

## 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, 7 Mei 2018



Cornelius Yan Mintareja – 13516113

E:\TUBES\Bekisting.exe

```
Input tinggi balok (dalam cm) : 42
Input lebar balok (dalam cm) : 16
Input panjang balok (dalam m) : 70

Jumlah bekisting balok minimal yang optimal yaitu 24 dengan cara Horizontal
Berikut adalah rincian pemotongannya ( T untuk bagian Tinggi, L untuk bagian Lebar )

Bentuk ini sebanyak 23 : L L T T T T T
Bentuk ini sebanyak 1 : L L L L L L L L L L L L T
Apakah ingin menghitung lagi ? ( y / n ) : _
```

Gambar 7 – Lampiran *screenshot* program optimasi bekisting dengan contoh kasus seperti diatas menggunakan algoritma greedy dan *brute force*