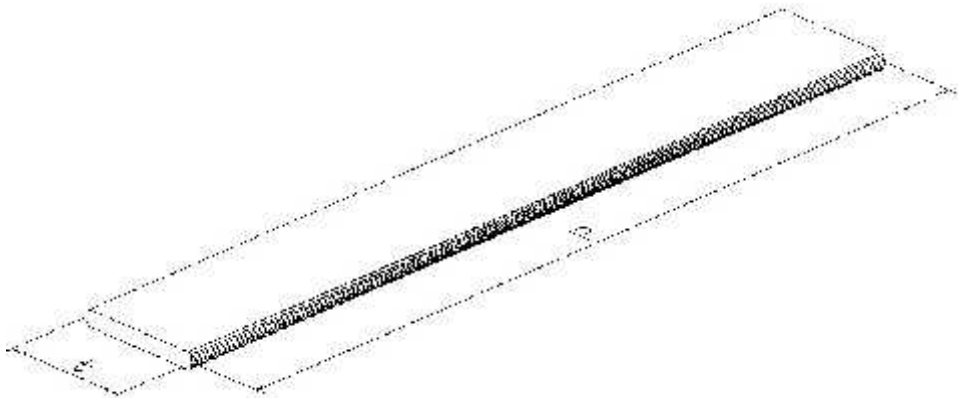


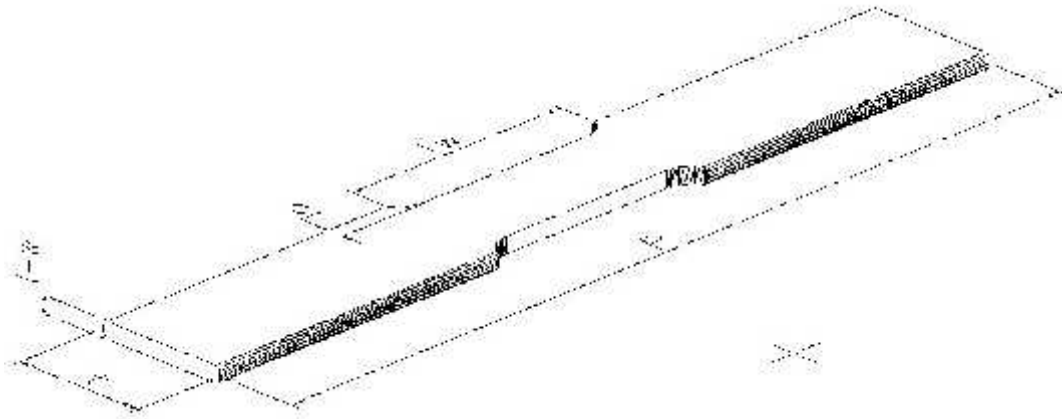
## BAB II TINJAUAN PUSTAKA

### 2.1. Definisi produk flat bar

Produk flat bar merupakan komponen dari truk yang berbentuk plate berupa lempengan baja memanjang yang berfungsi sebagai per daun. Produk ini terdapat berbagai macam tipe menurut bentuk dan ukurannya. Dalam hal ini PT. Indospring Tbk yang berperan sebagai customer mensub-orderkan (produk setengah jadi) flat bar kepada PT. IPGE dan PT.IPGE hanya memproses sebagian dari produk flat bar sampai finish menjadi produk jadi tersebut sesuai dengan spesifikasi yang diminta.



Gambar 4.1. Produk *flat-bar* sebelum di proses



Gambar 4.2. Produk *flat-bar* sesudah di proses

## 2.2. Peta Aliran Proses Produk *Flat-Bar*

Kegiatan	Simbol ASME				
	○	➔	□	◐	▽
1 Menata Flat-bar (disusun menjadi satu)	○				
2 Mengeklem Flat-bar menjadi satu	○				
3 Menunggu untuk di pindahkan (ambil kren)					
4 Memindahkan Flat-bar ke mesin Milling					
5 Merapikan Flat-bar pada cekam mesin Milling	○				
6 Memeriksa Flat-bar agar rata satu sama lain					
7 Mencekam Flat-bar	○				
8 Mengeklem Flat-bar jadi satu	○				
9 Menunggu untuk di proses (menyalakan mesin)					
10 Memproses sisi pertama	○				
11 Membersihkan Flat-bar	○				
12 Membalik Flat-bar	○				
13 Memeriksa Flat-bar agar rata satu sama lain					
14 Mencekam Flat-bar	○				
15 Mengeklem Flat-bar	○				
16 Menunggu untuk di proses (menyalakan mesin)					
17 Memproses sisi kedua	○				
18 Membersihkan Flat-bar	○				
19 Membuka cekam mesin Milling	○				
20 Memeriksa Flat-bar menggunakan JIG	○				
21 Menunggu untuk di angkat (mengambil kren)					
22 Mengangkat Flat-bar dari mesin Milling	○				
23 Meletakkan Flat-bar dan disusun kembali					

Gambar 2.2. Peta aliran proses produk *flat-bar*

### **2.3. Arti Dan Kegunaan *Linier Programming***

penggunaan sumber secara efisien atau alokasi sumber-sumber yang terbatas. Sebagian besar dari persoalan manajemen berkenaan dengan (tenaga kerja terampil, bahan mentah, modal) untuk mencapai tujuan yang diinginkan seperti penerimaan hasil penjualan yang harus maksimum, penerimaan devisa hasil ekspor nonmigas harus maksimum, jumlah biaya transpor harus minimum, lamanya waktu antrian untuk menerima pelayanan sependek mungkin, kemakmuran rakyat sebesar-besarnya.

Dalam keadaan sumber yang terbatas harus dicapai suatu hasil yang optimum. Dengan perkataan lain bagaimana caranya agar dengan masukan (input) yang serba terbatas dapat dicapai hasil kerja yaitu keluaran (output) berupa produksi barang atau jasa yang optimum. Linier programming akan memberikan banyak sekali hasil pemecahan persoalan, sebagai alternatif pengambilan tindakan, akan tetapi hanya ada satu yang optimum (maksimum atau minimum). Ingat bahwa mengambil keputusan berarti memilih alternatif, yang jelas harus alternatif yang terbaik.

Keadaan dalam praktek di mana pimpinan perusahaan bermaksud atau bertujuan untuk mencapai hasil penjualan sebesar mungkin. Logikanya harus memutuskan untuk memproduksi sebanyak-banyaknya, maka kalau semua barang tersebut laku dijual, tentu akan diperoleh jumlah hasil penjualan sebanyak-banyaknya. Tujuan benar-benar tercapai ! Akan tetapi keadaan belum tentu semulus itu, pimpinan perusahaan sebagai pembuat keputusan, ternyata akan menghadapi pembatasan-pembatasan, misalnya jumlah permintaan masyarakat tidak sebanyak yang diproduksi, sehingga barang susah dibuat. Pembatasan bukan berhenti disitu saja sebab mungkin dia menghadapi pembatasan seperti persediaan bahan mentah ternyata hanya tersedia sekian saja tidak bisa lebih, tenaga terampil yang aktif dan kreatif terbatas, jam mesin untuk memproses produksi terbatas, modal terbatas, dan permintaan masyarakat ternyata juga terbatas.

Persoalan yang timbul kemudian, bagaimana dapat mencapai hasil (output) yang optimum dengan memperhatikan input (manusia, modal, bahan

baku, waktu) yang tersedia memang terbatas. Jadi *linier programming* ialah salah satu teknik dari *Riset Operasi* untuk memecahkan persoalan optimum (maksimasi atau minimasi) dengan menggunakan persamaan dan ketidaksamaan linier dalam rangkai untuk mencapai pemecahan yang optimum dengan memperhatikan pembatasan-pembatasan yang ada.

### 2.3.1 Model *Linier Programming*

Model matematis perumusan masalah umum pengalokasian sumber daya untuk berbagai kegiatan, disebut sebagai model linier programming. Model *linier programming* ini merupakan bentuk dan susunan dari dalam menyajikan masalah-masalah yang akan dipecahkan dengan teknik linier programming. Dalam *linier programming* dikenal dua macam fungsi yaitu fungsi tujuan dan fungsi-fungsi batasan. Fungsi tujuan adalah fungsi yang menggambarkan tujuan sasaran di dalam permasalahan *linier programming* yang berkaitan dengan pengaturan secara optimal sumber daya-sumber daya, untuk memperoleh keuntungan maksimal atau biaya minimal. Pada umumnya nilai yang akan dioptimalkan dinyatakan dengan Z. Sedang fungsi batasan merupakan bentuk penyajian secara matematis batasan-batasan kapasitas yang tersedia yang akan dialokasikan secara optimal ke berbagai kegiatan.

Untuk memudahkan pembahasan model *linier programming* ini, digunakan simbol-simbol sebagai berikut :

- m : macam batasan-batasan sumber atau fasilitas yang tersedia.
- n : macam kegiatan-kegiatan yang menggunakan sumber atau fasilitas tersebut.
- i : nomor setiap macam sumber atau fasilitas yang tersedia (i = 1, 2, ....., m)
- j : nomor setiap macam kegiatan yang menggunakan sumber atau fasilitas yang tersedia (j = 1, 2, ....., n)
- $x_j$  : tingkat kegiatan ke, j. (j = 1, 2, ....., n)

$a_{ij}$  : banyaknya sumber  $i$  yang diperlukan untuk menghasilkan setiap unit keluaran (output) kegiatan  $j$  ( $i = 1, 2, \dots, m$ , dan  $j = 1, 2, \dots, n$ ).

$b_i$  : banyaknya sumber (fasilitas)  $i$  yang tersedia untuk dialokasikan ke setiap unit kegiatan ( $i = 1, 2, \dots, m$ ).

$Z$  : nilai yang dioptimalkan (maksimum atau minimum).

$C_j$  : kenaikan nilai  $Z$  apabila ada pertambahan tingkat kegiatan ( $x_j$ ) dengan satu satuan (unit) ; atau merupakan sumbangan setiap satuan keluaran kegiatan  $j$  terhadap nilai  $Z$ .

Keseluruhan simbol-simbol di atas selanjutnya disusun ke dalam bentuk tabel standar *linier programming*.

Tabel 1.1. Data untuk model linier programming

Sumber \ Kegiatan	Pemakaian sumber per unit kegiatan (keluaran)				Kapasitas sumber
	1	2	3	.....n	
1	$a_{11}$	$a_{12}$	$a_{13}$	..... $a_{1n}$	$b_1$
2	$a_{21}$	$a_{22}$	$a_{23}$	..... $a_{2n}$	$b_2$
3	$a_{31}$	$a_{32}$	$a_{33}$	..... $a_{3n}$	$b_3$
..	...	...	...	....	....
m	$a_{m1}$	$a_{m2}$	$a_{m3}$	..... $a_{mn}$	$b_m$
$\Delta Z$ pertambahan tiap unit	$C_1$	$C_2$	$C_3$	..... $C_N$	
Tingkat kegiatan	$X_1$	$X_2$	$X_3$	..... $X_n$	

Atas dasar tabel diatas kemudian dapat disusun suatu model matematis yang digunakan untuk mengemukakan suatu permasalahan *linier programming* sebagai berikut :

Formulasi model matematis dari persoalan pengalokasian sumber-sumber pada aktivitas :

Fungsi tujuan :

$$\text{Max } z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

Batasan-batasan :

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots + a_{1n} x_n \leq b_1$$

$$a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + \dots + a_{2n} x_n \leq b_2$$

$$a_{m1} x_1 + a_{m2} x_2 + a_{m3} x_3 + \dots + a_{mn} x_n \leq b_m$$

dan

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$$

Fungsi tujuan dalam *programa linier* mencerminkan atau menggambarkan tujuan yang ingin hendak dicapai dalam pemecahan suatu masalah *programa linier*. Batasan pertama mempunyai arti bahwa jumlah barang atau jasa 1 yang dihasilkan oleh kegiatan 1 dikalikan dengan kebutuhan akan sumber 1/satuan (berarti total alokasi 1 untuk kegiatan 1) ditambah dengan hasil kegiatan 2 dikalikan dengan kebutuhan tiap satuan keluaran 2 terhadap sumber 1 (dan seterusnya sampai dengan kegiatan ke – n) tidak akan melebihi jumlah (kapasitas) tersedianya sumber 1 (yang dinyatakan dengan  $b_1$ ). Hal ini berlaku pula untuk batasan-batasan lainnya sampai ke–m.

Terminologi umum untuk model *programa linier* yang diuraikan dapat diringkas sebagai berikut :

1. Fungsi yang akan dimaksimumkan :  $c_1 x_1 + c_2 x_2 + c_3 x_3 + \dots + c_n x_n$  disebut fungsi tujuan
2. Fungsi-fungsi batasan dapat dikelompokkan menjadi dua macam, yaitu
  - a. Fungsi batasan fungsional, yaitu fungsi-fungsi batasan sebanyak  $m$  (yaitu  $a_{i1} x_1 + a_{i2} x_2 + a_{i3} x_3 + \dots + a_{in} x_n$ )
  - b. Fungsi batasan non-negatif (non negative constraints) yaitu fungsi-fungsi batasan yang dinyatakan dengan  $x_i \geq 0$
3. Variabel-variabel  $x_j$  disebut sebagai decision variables
4.  $a_{ij}, b_i, c_j$  yaitu masukan-masukan (input) konstan, disebut sebagai parameter model.

Tentu saja, dalam praktek, tidak semua masalah linier programming dapat persis mengikuti model di atas. Masalah-masalah tersebut antara lain adalah :

1. Masalah minimasi, dimana seseorang dituntut untuk menentukan kombinasi (output) yang dapat minimumkan pengorbanan (misal : biaya). Dalam hal ini, tujuan dinyatakan sebagai berikut :

$$\text{Memimumkan } Z = C_1 X_1 + C_2 X_2 + C_3 X_3 + \dots\dots\dots C_n X_n$$

2. Masalah dengan fungsi batasan fungsional yang memiliki tanda matematis  $\geq$  ; sehingga apabila dirumuskan terlihat sebagai berikut :

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots\dots\dots + a_{1n} x_n \geq b_1$$

3. Masalah dengan fungsi batasan fungsional yang memiliki tanda matematis = ; sehingga bila dirumuskan sebagai berikut :

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots\dots\dots + a_{1n} x_n = b_1$$

4. Masalah tertentu, dimana fungsi batasan non-negatif tidak diperlukan ; atau dengan kata lain  $x_j$  tidak terbatas.

### 2.3.2. Asumsi Dasar *Linier Programming*

Dalam menggunakan model program linier, diperlukan beberapa asumsi sebagai berikut :

1. Asumsi kesebandingan (*proportionality*)

Kontribusi setiap variabel keputusan terhadap fungsi tujuan dan ruas kiri dari setiap pembatas adalah sebanding dengan nilai variabel keputusan itu. Jadi variabel  $X_j$  berkontribusi pada ongkos  $C_j X_j$  dan pada pembatas  $a_{ij} X_j$ . Jika  $X_j$  ditingkatkan dua kali, maka  $C_j X_j$  dan  $a_{ij} X_j$  akan meningkat dua kali.

2. Asumsi penambahan (*additivity*)

Ongkos total adalah penjumlahan dari ongkos individual, dan kontribusi total pada batas yang ke-i merupakan penjumlahan dari kontribusi individual dari individu aktivitas. Jadi berapapun nilai  $X_2$  pembuatan

sejumlah  $X_1$  suatu variabel keputusan akan selalu berkontribusi sebesar  $C_1$  terhadap fungsi tujuan. Dan berapapun harga  $X_1$  tidak akan mempengaruhi pembuatan sejumlah  $X_2$ .

3. Asumsi pembagian (*divisibility*)

Varibel keputusan dapat dibagi menjadi beberapa level fraksi (pecahan) sehingga nilai variable keputusan dibolehkan bukan integer. Jadi semua variable dapat memiliki harga berapapun asalkan real non negative.

4. Asumsi kepastian (*certainty*)

Setiap parameter, seperti koefisien fungsi tujuan, ruas kanan, koefisien teknologis, diasumsikan dapat diketahui secara pasti.

### 2.3.3. Teknik Pemecahan *Linier programming*

Dalam penyelesaian persoalan – persoalan *programa linier* dapat menggunakan dua cara yaitu dengan cara metode *grafik* dan dengan cara metode *simplek*.

#### 1. Metode *Grafik*

Secara matematis telah digambarkan bentuk dan susunan model yang berlaku dalam linier programming, yang bila diteliti nampak meliputi "kolom" dan "baris" yang teratur. Jumlah baris (menunjukkan batasan-batasan) ditentukan oleh banyaknya sumber yang akan dialokasikan ke setiap jenis kegiatan. Jumlah kolom ditentukan oleh jumlah atau macam kegiatan yang memerlukan sumber-sumber tersebut.

Metode grafik hanya dapat digunakan dalam pemecahan masalah linier programming yang ber"dimensi" :  $2 \times n$  atau  $m \times 2$ , karena keterbatasan kemampuan suatu grafik dalam "menyampaikan" sesuatu (sebenarnya grafik 3 dimensi dapat digambarkan, tetapi sangat tidak praktis). Hal ini merupakan persyaratan mutlak untuk dapat digunakannya metode grafik.



Langkah-langkah penggunaan metode grafik dapat ditunjukkan secara ringkas sebagai berikut :

- a. Menentukan fungsi tujuan dan memformulasikannya dalam bentuk matematis.
- b. Mengidentifikasi batasan-batasan yang berlaku dan memformulasikannya dalam bentuk matematis.
- c. Menggambarkan masing-masing garis fungsi batasan dalam satu sistem salib sumbu
- d. Mencari titik yang paling menguntungkan (optimal) dihubungkan dengan fungsi tujuan.

Sebelum mempraktekkan setiap langkah di atas sebaiknya terlebih dahulu diuraikan masalah yang biasanya paling kritis, yaitu menggambarkan garis-garis dari fungsi-fungsi batasan. Fungsi-fungsi batasan ini dinyatakan dalam tiga tanda, yaitu :

$\leq$  kurang dari atau sama dengan

$\geq$  lebih besar dari atau sama dengan

= sama dengan

## 2. Metode Simpleks

Suatu masalah *linier programming* hanya mengandung dua kegiatan (atau variabel-variabel keputusan) saja, maka akan dapat diselesaikan dengan *metode grafik*. Tetapi bila melibatkan lebih dari dua kegiatan maka *metode grafik* tidak dapat digunakan lagi, sehingga diperlukan *metode simplek*. *Metode simplek* merupakan suatu cara yang lazim dipakai untuk menentukan kombinasi optimal dari tiga variabel atau lebih.

Bila variabel keputusan yang dikandung tidak terlalu banyak masalah tersebut dapat diselesaikan dengan suatu *algorithm* yang biasanya sering disebut *metode simpleks tabel*. Disebut demikian karena kombinasi variabel keputusan yang optimal dicari dengan menggunakan tabel-tabel.

Langkah-langkah *metode simpleks tabel*

1. Mengubah fungsi tujuan dan batasan-batasan
2. Menyusun persamaan-persamaan di dalam tabel
  - a. Memilih kolom kunci
  - b. Memilih baris kunci
  - c. Mengubah nilai-nilai baris kunci
  - d. Mengubah nilai-nilai selain pada baris kunci
  - e. Melanjutkan perbaikan-perbaikan atau perubahan-perubahan

#### **2.4. Integer Programming (Program Bilangan Cacah)**

Programa bilangan bulat atau integer programming (IP) adalah bentuk lain dari programa linier (LP) dimana asumsi divisibilitasnya melemah atau hilang sama sekali. Bentuk ini muncul karena dalam kenyataannya tidak semua variabel keputusan dapat berupa bilangan pecahan misalnya, jika variabel keputusan yang dihadapi berkaitan dengan jumlah mesin yang diperlukan dalam suatu horison perencanaan, maka jawaban  $10/3$  mesin tidak realistis dalam konteks keputusan yang nyata. Dalam hal ini harus ditentukan, apakah akan menggunakan 3 atau 4 mesin.

Asumsi divisibilitas melemah, artinya sebagian dari nilai variabel keputusan harus berupa bilangan bulat (integer) dan sebagian lainnya boleh berupa bilangan pecahan. Persoalan IP dimana hanya sebagian dari variabel keputusannya yang harus integer disebut sebagai persoalan IP campuran.

Apabila seluruh variabel keputusan dari suatu persoalan programa linier (LP) harus berharga integer, maka persoalan tersebut disebut sebagai persoalan programa bilangan bulat (IP) murni. Dalam hal ini, asumsi divisibilitas dari LP hilang sama sekali.

Cara *simplek* adalah basis untuk penyelesaian problema *program linier* dimana disyaratkan bahwa semua variabel tak negatif. Tetapi untuk menyelesaikan problema (model) program bilangan cacah terdapat beberapa cara. Hanya saja baik program linier maupun program bilangan cacah mulai dengan ruang yang sama yaitu ruang penyelesaian layak (*feasible*). Tetapi

karena adanya persyaratan bilangan cacah bagi problem kedua yang berarti munculnya batasan tambahan menyebabkan adanya suatu pengurangan dari ruang penyelesaian layak.

Problema program bilangan cacah dapat dirumuskan sebagai berikut :

$$\text{Max} \quad : f = \sum_{j=1}^n c_j x_j$$

Dengan kendala

$$\sum_{j=1}^n a_{ij} x_j (\geq, =, \leq) b_i ; i = 1, 2, \dots, m$$

$$x_j \geq 0 \text{ semua bilangan cacah}$$

$$j = 1, 2, \dots, n$$

Dimana  $a_{ij}$ ,  $b_i$ ,  $c_j$  diketahui sebagai konstanta jika ternyata

1.  $x_j$  semua bilangan cacah, maka problema disebut problem program bilangan cacah murni (semua)
2.  $x_j$  sebagian bilangan cacah dan yang lainnya boleh tidak, maka disebut problem bilangan cacah campuran
3.  $x_j$  salah satu atau nol atau satu, problema disebut problema program nol-satu (0-1)

Cara penyelesaian :

Ada bermacam-macam pendekatan yang dapat digunakan untuk menyelesaikan problema program bilangan cacah. Beberapa diantaranya sudah menjadi metode umum untuk semua tipe dari bentuk program bilangan cacah, dan lainnya ada yang diperuntukkan untuk bentuk-bentuk khusus. Diantaranya ;

#### 1. Cara enumerasi

Metode enumerasi implisit sering kali digunakan untuk menyelesaikan persoalan program bilangan bulat nol-satu. Metode ini berangkat dari kenyataan bahwa setiap variabel harus berharga nol atau satu sehingga proses pencabangannya menjadi lebih sederhana dan penentuan fisibel-tidaknya suatu node juga menjadi lebih efisien.

## 2. Cara branch and bound

Teknik ini mencari solusi optimal dari suatu persoalan IP dengan mengenumerasi titik-titik dalam daerah fisibel dari suatu sub persoalan. Sebelum menjelaskan bagaimana cara kerja teknik ini, ada baiknya kita ingat lagi bahwa dari suatu persoalan IP, baik murni ataupun campuran, dapat diperoleh bentuk persoalan program linier relaksasi. Jika solusi dari LP relaksasi ini memiliki seluruh variabel yang berharga interger, maka solusi optimal dari persoalan LP relaksasi itu adalah juga solusi optimal dari persoalan IP.

## 3. Cara pendekatan program linier

Cara ini menyatakan bahwa problem bilangan cacah benar-benar dinyatakan sebagai problem program linier, artinya bilangan cacah tidak ikut serta dan diartikan kembali apabila jawab optimal merupakan bilangan pecahan maka dapat dilakukan dengan pembulatan baik ke bawah maupun ke atas.

## 4. Cara irisan gomory

Prosedur irisan ruang penyelesaian yang didasarkan pada intuisi dilakukan secara berulang-ulang akan menghasilkan jawaban optimal untuk program bilangan cacah. Cara irisan gomory merupakan suatu cara penyelesaian untuk menentukan jawaban optimal bagi program bilangan cacah.

### 2.5. Mesin Milling (*freis*)

Dua jenis utama pahat freis (milling cutter) adalah pahat pahat freis selubung atau mantel (slab milling cutter) dan pahat freis muka (face milling cutter), lihat gambar 2.1. Pahat freis termasuk pahat bermata potong jamak dengan jumlah mata potong sama dengan jumlah gigi freis ( $z$ ). Sesuai dengan jenis pahat yang digunakan dikenal dua macam cara yaitu mengefreis datar (slab milling) dengan sumbu putaran pahat freis selubung sejajar permukaan benda kerja, dan mengefreis tegak (face milling) dengan sumbu putaran pahat freis muka tegak lurus permukaan benda kerja. Selanjutnya mengefreis datar dibedakan menjadi dua macam cara yaitu : mengefreis naik (up milling atau konvensional milling) dan mengefreis turun (down milling).

Pahat freis dengan diameter tertentu dipasang pada poros utama (spindel) mesin freis dengan perantaraan poros pemegang (untuk pahat freis selubung) atau langsung melalui hubungan poros dan lubang konis (untuk pahat freis muka yang mempunyai poros konis). Putaran poros utama dapat dipilih sesuai dengan tingkatan putaran yang tersedia pada mesin freis. Posisi sumbu poros utama mesin freis dapat horisontal atau vertikal, tergantung pada jenis mesinnya, lihat gambar 2.2. Benda kerja yang dipasangkan pada meja dapat diatur kecepatan makannya tergantung pada harga gerak makan pergigi yang diinginkan. Besarnya kecepatan makan antara lain dipengaruhi oleh jumlah gigi pahat freis ( $z$ ). Untuk kecepatan makan yang sama maka gerak makan pergigi ( $fz$ ) menjadi berlainan bila jumlah gigi berbeda. Kedalaman potong ( $a$ ) diatur dengan cara menaikkan meja melalui roda pemutar untuk mengeser lutut pada tiang mesin freis.

Elemen- elemen dasar pada proses freis dapat ditentukan dengan memperhatikan gambar 2.3. Dalam hal ini rumus yang digunakan berlaku bagi kedua cara mengefreis, mengefreis tegak atau mengefreis datar.

#### Benda kerja

$w$  : lebar pemotongan

$l_w$  : panjang pemotongan

$a$  : kedalaman potong

#### Pahat freis

$d$  : diameter luar

$z$  : jumlah gigi

$k_t$  : sudut potong utama,

: 90 derajat untuk pahat freis selubung

#### Mesin freis

$n$  : putaran poros utama

$v_f$  : kecepatan makan

Elemen dasar proses freis adalah sebagai berikut :

1. Kecepatan potong :  $v = \frac{f \cdot d \cdot n}{1000}$  ; m/min

2. Gerak makan pergigi :  $f_z = \frac{v_f}{z_n}$  ; mm/(gigi)
3. Waktu pemotongan :  $t_c = \frac{l_t}{v_f}$  ; min
- Dimana,  $l_t = l_v + l_w + l_n$  ; mm
- $l_v \geq \sqrt{a(d-a)}$  ; untuk mengefreis datar
- $l_v \geq 0$  ; untuk mengefreis tegak
- $l_n \geq 0$  ; untuk mengefreis datar
- $l_n = \frac{d}{2}$  ; untuk mengefreis tegak
4. Kecepatan penghasilan geram
- $z = \frac{v_f \cdot a \cdot w}{1000}$  ; cm<sup>3</sup>/min

## 2.6. Optimasi Proses Pemesinan

Berdasarkan gambar teknik suatu produk atau komponen mesin beserta bentuk dan ukuran bahan yang ada, maka dapat direncanakan langkah pengerjaan dengan urutan yang paling baik (logik). Apabila jenis proses serta mesin perkakas telah ditetapkan, tindakan selanjutnya adalah menentukan jenis pahat yang akan digunakan sesuai dengan urutan atau langkah pengerjaan. Bentuk, dimensi dan jenis material pahat perlu dipilih dengan seksama, terutama geometrinya.

Tindakan sebagaimana yang disebutkan di atas mungkin dapat dipenuhi oleh seorang operator mesin perkakas yang terdidik dan berpengalaman, dimana ia sendiri harus menentukan segalanya (hal yang umum ditemukan dalam suatu bengkel, industri kecil, laboratorium, dan berbagai fasilitas penunjang atau pemeliharaan). Akan tetapi, suatu proses pembuatan tidak hanya berkaitan dengan faktor teknologi saja, melainkan berkaitan pula dengan faktor ongkos dan mungkin juga dengan faktor lain seperti kecepatan produksi demi untuk memenuhi target atau pesanan taupun seperti kecepatan produksi demi untuk

memenuhi target atau pesanan ataupun untuk mencapai keuntungan secepat mungkin. Dengan demikian, untuk perencanaan kondisi proses ini perlu ditunjang dengan data mengenai ongkos operasi, ongkos pahat, dan data pemesinan yang lengkap, yang biasanya tidak diketahui oleh seorang operator mesin. Hal ini sebenarnya memang merupakan tugas bagi seorang ahli proses (ahli teknik produksi) yang menguasai teori, memiliki data serta pengalaman yang memungkinkan penentuan kondisi proses pemesinan sesuai dengan tujuan dengan sebaik mungkin (optimum).

### 2.6.1. Komponen Waktu Produksi

Waktu untuk menghasilkan produk atau waktu yang diperlukan untuk menyelesaikan suatu pekerjaan (memotong bagian tertentu produk) dengan cara yang tertentu (menggunakan suatu jenis pahat) adalah merupakan variabel yang penting dalam rangkai penentuan kondisi pemesinan optimum. Untuk jumlah produk yang cukup besar maka secara kasar dapat ditentukan waktu pemesinan rata-rata untuk mengerjakan satu produk, yaitu dengan cara membagi seluruh waktu yang digunakan dengan jumlah produk yang dihasilkan. Akan tetapi, cara ini tidak baik untuk dilaksanakan karena tidak memberikan informasi yang jelas mengenai komponen waktu (bagian waktu total) yang berkaitan dengan setiap langkah pengerjaan. Sesuai dengan tujuan optimasi maka diinginkan pembagian waktu menurut komponennya sehingga dapat diketahui komponen waktu yang mana yang mungkin dapat diperkecil.

Secara garis besar dapat dikelompokkan 2 macam komponen waktu yaitu

- Komponen waktu yang dipengaruhi oleh variabel proses dan,
- Komponen waktu yang bebas

Untuk menghasilkan satu produk, maka diperlukan komponen-komponen waktu sebagai berikut :

a. Komponen waktu yang dipengaruhi oleh variabel proses

$$1. \quad t_c = \frac{l_t}{v_f} = \frac{l_t}{n \cdot f} \quad ; \text{ min/produk}$$

Dimana :

$t_c$  : waktu pemotongan sesungguhnya

$l_t$  : panjang pemesinan ; mm

$v_f$  : kecepatan makan ; mm/min

2.  $t_d \frac{t_c}{T}$  : min/produk

$t_d$  : waktu penggantian atau pemasangan pahat ; min

$T$  : umur pahat ; min

$\frac{t_c}{T}$  : bagian dari umur pahat yang digunakan untuk menyelesaikan satu produk (harus diusahakan lebih kecil dari 1)

b. Komponen waktu bebas (non produktif)

$$t_a = t_{LW} + t_{AT} + t_{RT} + t_{UW} + \frac{t_s}{n_t}$$

dimana :

$t_a$  : waktu non produktif ; min/produk

$t_{LW}$  : waktu pemasangan benda kerja ; min/produk

$t_{AT}$  : waktu pentiapan ; min/produk

$t_{RT}$  : waktu pengakhiran ; min/produk

$t_{UW}$  : waktu pengambilan produk ; min/produk

$\frac{t_s}{n_t}$  : bagian dari waktu penyiapan mesin beserta perlengkapannya yang dibagi rata untuk sejumlah produk yang direncanakan untuk dibuat saat itu.

Dengan demikian waktu pemesinan perproduk rata-rata adalah

$$t_m = t_a + t_c + t_d \frac{t_c}{T} \quad ; \text{ min/produk}$$



## 2.6.2. Komponen Biaya Produksi

Bagi suatu industri pemesinan adalah mutlak untuk mengetahui beberapa ongkos sebenarnya dalam pembuatan suatu produk atau komponen mesin. Dengan mengetahui harga jual produk atau harga penawaran kontrak pembuatan sejumlah produk maka dapat dibayarkan keuntungan yang akan diperoleh. Dalam kenyataan, penghitungan ongkos pembuatan tersebut tidak selalu mudah, tergantung pada ukuran perusahaan, ragam dan kompleksitas produk yang ditanganinya dan struktur penghitungan ongkos yang dianut oleh perusahaan yang bersangkutan. Ongkos pembuatan dapat ditentukan dari beberapa komponen ongkos yang membentuknya. Berbagai bentuk struktur komponen-komponen ongkos telah diajukan orang, masing-masing dengan cara pendekatan yang berbeda dengan anggapan dan penyederhanaan yang berlainan disesuaikan dengan kondisi atau ukuran perusahaan, guna mempermudah penghitungan ongkos (atau untuk tujuan lain).

Ongkos suatu produk ditentukan oleh ongkos material (bahan dasar) dan ongkos produksi yang mungkin terdiri atas gabungan beberapa langkah proses pembuatan atau pemesinan sebagaimana rumus berikut,

$$C_u = C_M + C_{plan} + C_p \quad ; \text{Rp/produk}$$

Dimana,

$C_u$  : Ongkos total ; Rp/produk

$C_M$  : Ongkos material ; Rp/produk

$C_{plan}$  : Ongkos persiapan atau perencanaan produksi

$C_p$  : Ongkos salah satu proses produksi ; Rp/produk

Ongkos material terdiri atas harga pembelian dan ongkos tak langsung yang merupakan ongkos khusus yang dibebankan bagi material yang berkaitan dengan penyimpanan (sewaktu masih berupa bahan ataupun setelah menjadi produk) dan penyiapan. Bagian gudang membebani perusahaan dengan adanya ruang atau gedung, mesin-mesin pemotong, pengangkutan dengan perhitungan atas bunga, pajak dan asuransi, pemeliharaan, serta karyawan yang menangani

bagian pergudangan. Kesemuanya itu dibagi (dengan faktor pemberat) dan dibebankan bagi masing-masing material yang ada di gudang sesuai dengan luas lantai yang diperlukan dan lama penyimpanan.

$$C_M = C_{Mo} + C_{Mi} \quad ; \text{Rp/produk}$$

Dimana :

$C_M$  : Ongkos material ; Rp/produk

$C_{Mo}$  : Harga pembelian ; Rp/produk

$C_{Mi}$  : Ongkos tak langsung ; Rp/produk

Ongkos proses produksi dapat dirinci menjadi ongkos penyiapan dan peralatan, ongkos pemesinan dan ongkos pahat, yaitu :

$$C_P = C_r + C_m + C_e \quad ; \text{Rp/produk}$$

Dimana :

$C_P$  : Ongkos produksi ; Rp/produk

$C_r$  : Ongkos penyiapan dan peralatan ; Rp/produk

$C_m$  : Ongkos pemesinan ; Rp/produk

$C_e$  : Ongkos pahat ; Rp/produk

Peralatan khusus seperti pemegang mungkin diperlukan guna memperkecil waktu non produktif (lihat waktu pemesinan perproduk rata-rata) fixture ini dibuat sesuai dengan mesin perkakas yang digunakan, langkah atau cara pemesinan dan geometri bahan (hasil proses sebelumnya). Ongkos pembuatan fixture yang biasanya mahal (dibuat khusus, toleransi geometri komponen-komponennya cukup sempit) harus diatasi dengan memperbesar jumlah produk yang akan dibuat (lot size). Ditinjau dari proses produksi, ongkos ini dianggap sebagai ongkos tetap yang tidak dipengaruhi oleh laju kecepatan produksi.

$$C_r = (C_{set} + C_{fix} + C_{pr}) / n_t \quad ; \text{Rp/produk}$$

Dimana :

$C_r$  : Ongkos persiapan dan peralatan khusus ; Rp

$C_{set}$	: Ongkos pengaturan atau setting mesin	; Rp
	: $c_m \cdot t_{set}$	
$C_{fix}$	: Ongkos perkakas-bantu-cekam	; Rp
$C_{pr}$	: Ongkos penyiapan program NC	; Rp
$n_t$	: Jumlah produk yang dibuat	; buah

Ongkos pemesinan dihitung berdasarkan waktu pemesinan perproduk dan ongkos operasi (persatuan waktu; menit) dengan demikian dipengaruhi oleh laju kecepatan produksi. Ongkos pahat perlu ditetapkan sebagai komponen ongkos yang terpisah karena mempengaruhi kaitan langsung dengan umur pahat yang merupakan variabel utama dalam proses pemesinan.

$$C_m = c_m \cdot t_m \quad ; \text{Rp/produk}$$

Dimana :

$C_m$	: Ongkos pemesinan	; Rp/produk
$c_m$	: Ongkos operasi mesin (mesin, operator, oferhead) persatuan waktu	; Rp/min
$t_m$	: Waktu pemesinan	; min/produk

$$C_e = c_e \cdot \frac{t_c}{T} \quad ; \text{Rp/produk}$$

Dimana :

$C_e$	: Ongkos pahat	; Rp/produk
$c_e$	: Ongkos pahat per mata potong	; Rp/mata potong
$\frac{t_c}{T}$	: Sebagian dari umur pahat (yang berkurang akibat pemakaiannya setiap menghasilkan setiap satu produk) merupakan rasio antara waktu pemotongan efektif $t_c$ dengan umur pahat T	; mata potong/produk.

## 2.7. Penelitian-penelitian yang relevan

Rratno dwi hartanto (ITATS 2004). Telah meneliti tentang ' perencanaan produksi gear yang optimal dengan integer programming dalam upaya memaksimalkan laba perusahaan di PT. Tjokro Putra Perkasa Surabaya' dengan hasil : selama ini sistem produksi yang digunakan pada perusahaan yaitu selalu memproduksi sesuai dengan kapasitas reguler yang ada dilakukan berdasarkan p0engalaman maupun kebiasaan dari produksi periode sebelumnya. Produksi yang telah dibuat sering terganggu karena minat konsumen untuk membeli jenis produk tertentu diluar rencana, sehingga perlu adanya penambahan jual produk untuk memenuhi permintaan konsumen. Menentukan fungsi tujuan didapat dengan cara mengurangi harga penjualan tiap produk dengan total biaya produksi tiap produk, sehingga diperoleh keuntungan tiap ppproduknya. Output yang diharapkan penulis untuk perusahaan adalah bisa membuat perencanaan jumlah produksi yang optimal dan mendapatkan keuntungan yang optimal sebelum memakai integer programming sebesar Rp. 198.692.817 setelah memakai integer programming sebesar Rp. 5.418.289 meningkat 2,8 % pertahun bila dibandingkan kondisi nyata perusahaan.