

# PERENCANAAN PROSES BERBASIS FEATURE (PENERAPAN METODE EDGE GRAPH) PADA PERANCANGAN BED DAN BASE PLATE DINAMOMETER GAYA PEMOTONGAN

Insannul Kamil, Taufik, Bakri Bakar

## ABSTRAK

Untuk mengintegrasikan tahap perancangan dengan tahap manufaktur diperlukan pemanfaatan informasi hasil perancangan menjadi informasi manufaktur yang diperlukan. Hasil perancangan dalam bentuk gambar komponen pada perangkat lunak CAD (Computer Aided Design) harus diidentifikasi featurenya. Terdapat banyak metode yang dapat digunakan untuk pengenalan feature. Dalam penelitian ini dipilih metode *face edge graph*. Data hasil pengenalan feature diidentifikasi dalam beberapa tahap untuk menghasilkan suatu rencana proses manufaktur yang berisi urutan pengerjaan produk. Output rencana proses ini diimplementasikan pada pembuatan program NC (G-Code) yang akan diterapkan pada mesin CNC (Computer Numerical Control) STAMA MC-520 di Laboratorium Teknologi Produksi Fakultas Teknik Universitas Andalas.

## 1. PENDAHULUAN

Masalah yang timbul antara sistem CAD dan CAM adalah interpretasi manual tentang operasi manufaktur yang dibutuhkan untuk pembuatan suatu komponen (*part*). Dalam sistem CAD, deskripsi komponen berwujud entitas tingkat rendah seperti muka (*face*) dan rusuk (*edge*). Sedangkan yang dibutuhkan dalam proses manufaktur adalah informasi yang berhubungan dengan entitas tingkat yang lebih tinggi seperti slot, step, dan hubungan antar entitas tersebut. Dengan demikian integrasi CAD/CAM memerlukan kemampuan untuk mendapatkan informasi mengenai entitas tingkat tinggi dari data entitas tingkat rendah, atau menghubungkan geometri komponen dengan operasi manufaktur secara otomatis.

Representasi dan *reasoning* aspek geometri produk dan proses merupakan hal yang kritis dalam mengintegrasikan perancangan dan manufaktur. Pemodelan solid konvensional seperti pemodelan yang berbasis *Boundary-Representation (B-Rep)* dan *Constructive-Solid-Geometri (CSG)*, tidak memungkinkan untuk mendukung secara efisien pengembangan aplikasi dari tahap perancangan menuju tahap manufaktur.

Untuk mengurangi kesenjangan informasi tersebut, digunakan bentuk *feature* untuk menjembatani informasi geometri perancangan ke aplikasi manufaktur. Hasil perancangan dalam bentuk gambar komponen dalam perangkat lunak *Computer Aided Design (CAD)* diidentifikasi atau dikenali *feature*nya. *Feature* yang diperoleh akan digunakan untuk perencanaan proses.

Pada penelitian ini dilakukan perencanaan proses berbasis *feature* (penerapan metoda *face edge graph*) pada perancangan *bed* dan *base plate*

dinamometer gaya pemotongan dengan *feature-feature* dasar berupa *hole*, *pocket*, *slot* dan *step*.

## 2. METODE *FACE EDGE GRAPH*

Dalam pendekatan pengenalan *feature* berbasis *graph* ini, *boundary representation* dari suatu komponen ditransformasikan ke dalam *Attributed Adjacency Graph* (AAG), yaitu suatu bentuk representasi *graph* dalam bentuk matriks triangular pada komputer. AAG adalah sebuah *graph* dengan atribut yang diberikan pada setiap *arc*. Sebuah AAG didefinisikan sebagai  $G = (N, A, T)$  dimana  $N$  adalah kumpulan *node*,  $A$  adalah kumpulan *arc*, dan  $T$  adalah kumpulan atribut yang diberikan pada *arc* yang terdapat dalam  $A$ . Setiap *face* pada komponen direpresentasikan sebagai *node* dalam AAG. Perbatasan antara dua *face* ditunjukkan sebagai *arc*. Jika dua *face* yang berbatasan membentuk sudut konveks, *arc* diberi nilai 1. Sedang perbatasan konkaf diberi nilai 0 [Chang, 1990].

*Feature* suatu komponen juga dapat direpresentasikan dengan menggunakan AAG. Sebuah aturan dapat dikembangkan untuk setiap jenis *feature*. Sebelum dilakukan pencarian *sub graph* dari sebuah *feature*, dilakukan prosedur pembuangan *node*. Prosedur ini membuang hanya *node* yang mempunyai *arc* yang bernilai '1'. Prosedur ini memecah sebuah *graph* menjadi beberapa *subgraph* yang terpisah. Setiap *subgraph* berkaitan dengan sebuah *feature* atau sekumpulan *feature* yang saling berpotongan.

Sebuah *feature* dapat dikenali melalui prosedur pencocokan (*matching*). *Feature* yang saling berpotongan dapat memiliki kesamaan dalam hal *faces*; karena itu lebih banyak proses yang harus dilakukan sebelum *feature* tersebut dapat dikenali. Prosedur pemecahan *node* (*node splitting*) digunakan untuk memecah dua *feature* yang saling berpotongan.

Sebagai contoh, pada gambar 3, terdapat *slot* yang saling berpotongan. *Face* 1 digunakan oleh kedua *slot*, dan *face* 2 dan 3 sebenarnya merupakan *face* yang sama. Prosedur *node splitting* akan memecah *face* (*node*) 1 menjadi 2 *face*. Prosedur *node joining* akan menggabungkan *face* 2 dan *face* 3 menjadi *face* 2/3.

Pendekatan ini cukup efektif digunakan untuk bentuk 2 1/2 D. Strategi pembuangan *link arc* mengurangi proses pencarian yang tidak perlu. Dengan kemampuan untuk memisahkan dan menggabungkan *node*, pendekatan ini dapat mengenali perpotongan antar *feature*. Pendekatan ini tidak menggunakan informasi geometri, maka untuk mengklasifikasikan *feature* secara rinci dibutuhkan langkah lain.

## 3. Perencanaan Proses Berbasis *Feature*

Dalam perencanaan proses, hubungan antar *feature* dalam suatu komponen (*part*) sangat berpengaruh pada urutan pengerjaan, karena pengerjaan suatu *feature* dapat mengakibatkan pembentukan permukaan baru bagi pengerjaan *feature* selanjutnya [Wong & Siu, 1992]. Dengan demikian diperlukan informasi tentang ketergantungan antar *feature*.

Setiap *feature* memiliki paling sedikit satu normal, yang menunjukkan arah pendekatan *tool* untuk pengerjaan *feature* tersebut. Sehingga untuk pengerjaan suatu *part* yang terdiri dari beberapa *feature*, dapat dihasilkan banyak kemungkinan untuk menyelesaikan *part* tersebut. *Features* yang memiliki normal yang sama dapat dikerjakan dalam satu kali *setup*. Berdasarkan hal ini, maka dapat dilakukan optimasi dengan cara pengelompokan *feature* atas dasar kesamaan *setup* agar proses pengerjaan lebih cepat.

Pengerjaan suatu *feature* dapat terdiri dari beberapa operasi yang akan membutuhkan *tool* tertentu [Chang, 1990], sehingga dalam setiap *setup* dapat terjadi beberapa kali pergantian *tool* yang akan membutuhkan waktu. Hal ini dapat dikurangi dengan jalan mengelompokkan *feature* yang dapat menggunakan *tool* yang sama.

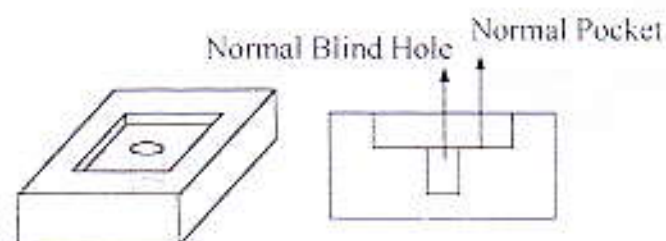
Sistem perencanaan proses yang dikembangkan merupakan suatu *rule-based system* dimana pada setiap langkah perencanaan proses diterapkan sekumpulan *rules* (aturan-aturan) untuk menghasilkan urutan pengerjaan suatu komponen. Sistem perencanaan proses berbasis *feature* ini terdiri dari tiga bagian utama, yaitu :

#### 1. Pembentukan ketergantungan *features* (*precedence feature*).

Bagian ini memeriksa jika ada ketergantungan antar *feature* dalam suatu komponen, yang dapat mempengaruhi urutan pengerjaan (hubungan *parent-child*).

Langkah-langkahnya :

- (1) Kelompokkan tiap *feature* ke dalam normal-normal yang diwakilinya.
- (2) Untuk setiap normal terwakili, urutkan *feature* berdasarkan posisi (koordinat).
- (3) Untuk setiap *feature*, periksa ketergantungannya dengan *feature* lain yang berada di permukaan *feature* tersebut.
- (4) Kelompokkan tiap *feature* ke dalam normal-normal yang diwakilinya.
- (5) Untuk setiap normal terwakili, urutkan *feature* berdasarkan posisi (koordinat).
- (6) Untuk setiap *feature*, periksa ketergantungannya dengan *feature* lain yang berada di permukaan *feature* tersebut.



Gambar Contoh Ketergantungan Antar *Feature*

Pada gambar diatas *blind hole* memiliki ketergantungan dengan *pocket*, karena *blind hole* dapat dicapai jika *pocket* sudah dikerjakan, maka permukaan *blind hole* berada pada *pocket*.

Ketergantungan : ***Pocket (Parent) → Hole (Child)***

Aturan untuk pembentukan ketergantungan *feature* adalah :

**Rule 1** JIKA permukaan *feature* A berada pada permukaan *feature* B.  
MAKA *feature* B merupakan parent dari *feature* A.

## 2. Pengurutan proses (*global sequencing*).

Bagian ini menentukan jumlah *setup* yang dibutuhkan untuk membuat suatu komponen dengan meminimasi jumlah *setup* yang diperlukan dengan normal *feature* yang dikelompokkan. Sehingga dapat ditentukan *feature* yang dapat dikerjakan dalam satu *setup*.

Pengurutan proses merupakan suatu modul untuk menentukan urutan *setup* yang harus dilakukan untuk membuat suatu komponen beserta urutan pengerjaan setiap *feature* dalam tiap *setup*. Langkah utama dalam pengurutan proses ini adalah :

- (1) Kelompokkan semua *feature* atas dasar kesamaan normal yang dimilikinya.
- (2) Pilih normal yang dapat mewakili jumlah *feature* paling banyak.
- (3) Periksa setiap *feature* pada normal tersebut, apakah hubungan ketergantungan antar *feature* (*precedence*) masih tetap terjaga.

**Rule 2** JIKA parent dari *feature* A berada pada normal yang sama.  
MAKA Kedua *feature* dikelompokkan dalam satu *setup*.

**Rule 3** JIKA *Feature* A merupakan parent dari *feature* B.  
DAN *Feature* A dan *feature* B tidak berada pada normal yang sama.  
MAKA Urutkan *setup* sesuai dengan *precedence* (*setup feature* A mendahului *setup feature* B).

- (4) Hilangkan duplikasi pengerjaan *feature*. Satu *feature* hanya boleh didekati dari satu normal. Jika masih ada *feature* yang belum terkelompok, kembali ke langkah 2.

Tujuan pengurutan ini adalah meminimasi jumlah normal yang terpilih sebagai arah *setup* dengan memaksimalkan jumlah *feature* yang dapat dikerjakan pada satu normal.

## 3. Pengurutan operasi (*final sequencing*).

Bagian ini melakukan pemilihan proses dan *tool* yang sesuai untuk pengerjaan komponen. Dengan minimasi penggantian *tool* dengan memanfaatkan orientasi *setup* dari normal yang terpilih.

Tahap ini terdiri dari tiga bagian utama, yaitu :

1. Penentuan urutan operasi yang harus dilewati setiap *feature*, dengan memanfaatkan basis data proses. Basis data proses didefinisikan untuk setiap *feature*, sebagai berikut :

Tabel 3.4 Basis Data Proses

(Berisi hubungan antara *features* dengan operasi)

Untuk setiap jenis operasi yang ada didefinisikan *tool* yang akan digunakan :

- Operasi *mill* menggunakan *end milling cutter*.
- Operasi *mill* vertikal *fillet* menggunakan *end milling cutter* dengan diameter yang sesuai atau lebih kecil dari diameter *fillet* yang dimiliki *feature*.
- Operasi *mill middle* merupakan operasi pembuangan bagian tengah khusus pada *feature* yang memiliki horizontal *fillet*, menggunakan *end milling cutter*.

**Rule 4** : JIKA Operasi = Mill

ATAU Operasi = Mill Vertikal *Fillet*

ATAU Operasi = Mill Middle

MAKA Jenis *tool* = "End Milling Cutter"

- Operasi *Mill horizontal fillet* menggunakan *ball mill cutter* dengan ukuran yang sesuai dengan diameter *fillet feature*.

**Rule 5** : JIKA Operasi = Mill Horizontal *Fillet*

MAKA Jenis *tool* = "Ball Mill Cutter"

- Operasi *Drilling* menggunakan *drill cutter*.

**Rule 6** : JIKA Operasi = Drill

MAKA Jenis *tool* = "Drill Cutter"

2. Pemilihan *tool* dilakukan dengan memanfaatkan basis data *tool*, yang berisi informasi mengenai karakteristik *tool* : jenis *tool*, panjang *tool* dan diameter *tool*. Untuk meminimasi waktu pemesinan (operasi *mill*), dipilih diameter *tool* sebesar-besarnya. Tetapi ada pembatas bagi diameter *tool* yang akan digunakan yaitu geometri *feature*. Maka dibuat *rule* dengan mendefinisikan diameter *tool* maksimum yang dapat digunakan sesuai dengan geometri *feature*.

**Rule 7** : JIKA *Feature* = Step

MAKA Diameter *tool*  $\geq 1.25 \times$  lebar *feature* .

**Rule 8** : JIKA *Feature* = Slot

MAKA Diameter *tool*  $\leq$  lebar *feature* .

**Rule 9** : JIKA *Feature* = Blind hole

MAKA Diameter *tool*  $\leq$  diameter hole

- Rule 10 :** JIKA *Feature* memiliki vertikal *fillet*  
MAKA  $Diameter\ tool < 8 * radius\ fillet$ .
- Rule 11 :** JIKA *Feature* memiliki vertikal *fillet*  
DAN  $Diameter\ tool > lebar\ feature$   
MAKA  $Diameter\ tool \leq lebar\ feature$ .
- Rule 12 :** JIKA Operasi = Mill vertikal *fillet*  
MAKA  $Diameter\ tool \leq 2 * radius\ fillet$ .
- Rule 13 :** JIKA Operasi = Mill horizontal *fillet*  
MAKA  $Diameter\ tool \leq 2 * radius\ fillet$ .
- Rule 14 :** JIKA  $Diameter\ tool > maksimum\ diameter\ tool$  yang dimiliki (terdapat dalam database)  
MAKA  $Diameter\ tool \leq maksimum\ diameter\ tool$  yang dimiliki.
- Rule 15 :** JIKA Operasi = Drill  
MAKA  $Diameter\ tool = diameter\ through\ hole$ .

### 3. Minimasi *tool* terpilih.

Agar tidak terjadi pergantian *tool* yang terlalu sering, maka diterapkan aturan-aturan (*rules*) untuk meminimasi jumlah *tool* yang terpilih untuk digunakan pada setiap *setup*. Langkah yang dilakukan adalah :

- (1) Pilih satu *setup*.
- (2) Periksa tiap diameter *tool* yang ada.
- (3) Periksa jumlah *feature* yang dapat dikerjakan dengan menggunakan *tool* tersebut.
- (4) Pilih *tool* dengan diameter yang dapat mengerjakan jumlah *feature* paling banyak.
- (5) Lakukan langkah 2 – 4 sampai untuk semua *feature* terpilih *tool*.
- (6) Lakukan langkah 1 – 5 untuk *setup* selanjutnya.

Aturan untuk menentukan apakah suatu *feature* dapat dikerjakan dengan diameter *tool* tertentu (yang sedang diperiksa) adalah :

- Rule 16 :** JIKA *Feature* = Step  
DAN  $Diameter \geq diameter\ tool\ feature$   
MAKA  $Diameter\ tool = diameter$
- Rule 17 :** JIKA *Feature* = Step  
DAN  $Diameter < diameter\ tool\ feature$

DAN Lebar *feature* / Diameter  $\leq 4$  (pembatas tambahan)

DAN Diameter / tinggi *feature*  $> 1/6$  (pembatas tambahan)

MAKA Diameter *tool* = diameter

#### 4. Pengurutan

Langkah-langkah dalam pengurutan ini, adalah :

- (1) Kelompokkan *feature* atas dasar kesamaan *tool* yang sudah terpilih.
- (2) Periksa hubungan *parent-child* dalam setiap kelompok, dan kemudian perbaiki urutan pergantian *tool*.

## 4 HASIL DAN ANALISIS

### 4.1 Analisis Model Pengenalan *Feature*

#### Graph komponen

*Graph* komponen adalah *graph* yang merepresentasikan komponen yang akan dikenali. Aturan yang digunakan dalam pembuatan *graph* komponen adalah:

- Setiap muka (*face*),  $f_i$  dari sebuah komponen direpresentasikan dengan sebuah simpul (*node*) dalam *graph* G.
- Setiap *node* memiliki atribut yang menunjukkan tipe *face*.
- Untuk setiap rusuk (*edge*),  $e_j$  dari sebuah komponen, yang berada pada dua *face*  $f_i$  dan  $f_s$ , ada sebuah *link arc* yang menghubungkan kedua *node* tersebut dalam G.
- Setiap *link* memiliki atribut yaitu nilai yang menunjukkan sudut pertemuan kedua *face* pada tengah *edge* yang diwakilinya, dan atribut yang menunjukkan tipe *edge*.
- Untuk setiap *node*, jika *face* yang diwakilinya memiliki lebih dari 1 *loop*, maka:  
Jika (atribut tipe *face* bukan 1) atau (atribut tipe *face* = 1 dan *node* memiliki *link* yang mewakili *edge* bukan lingkaran penuh).  
Maka *node* dipecah menjadi beberapa *node*, masing-masing memiliki kelompok *link* yang mewakili satu *loop*.

#### Graph *feature* standar

Aturan pembuatan *graph* untuk setiap *feature* standar, sama seperti pembuatan *graph* komponen, tetapi hanya untuk *face-face* yang membatasi *feature* standar tersebut. Ada pengecualian yaitu untuk *feature hole*, dimana *hole* berujung dimasukkan juga dalam *graph*. Hal ini dilakukan untuk mendapatkan data *edge* pada ujung *hole*, karena tanpa data ini informasi mengenai *feature hole* tidak dapat diperoleh kembali.

Sesuai dengan ruang lingkup penelitian, *feature* standar terbatas pada *feature* yang telah didefinisikan, yaitu *pocket*, *step*, *slot*, dan *hole* tanpa melibatkan bentuk gabungan. Hal ini bukan berarti bentuk gabungan tidak dapat

dikenali sama sekali, melainkan bentuk tersebut dapat dikenali dengan cara mendefinisikannya sebagai bentuk standar yang baru.

#### **Proses pencarian graph feature standar**

Dengan adanya *graph* komponen dan *graph feature* standar maka proses pengenalan *feature* adalah proses mencari *graph feature* standar di dalam *graph* komponen. Aturan proses pencarian adalah sebagai berikut :

- Di antara 2 buah *graph feature* standar A dan B, pencarian *graph* A harus mendahului *graph* B jika B merupakan *sub-graph* dari A.
- *Feature* standar ditemukan jika pada *graph* komponen terdapat *sub-graph* yang memiliki *feature* dan atribut yang sama seperti *graph feature* standar.
- Setiap *node* dan *link* pada *graph* komponen hanya dapat menjadi bagian dari satu *feature* standar saja.

#### **4.2 Analisis Model Perencanaan Proses Berbasis Feature**

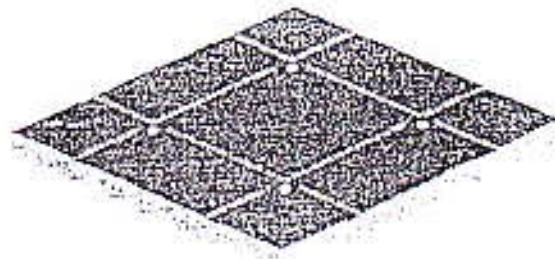
Model desain produk yang dikerjakan mengacu pada ketersediaan 18 *feature-ID* baik secara terpisah maupun secara gabungan (*compound feature*). Dalam suatu model *part* acuan (referensi) *feature* yang dipunyai berpedomankan pada salah satu titik pandang yang berhubungan dengan sumbu koordinat. Acuan ini akan menentukan posisi *feature* yang tersedia dalam komponen (*part*) tersebut.

Koordinat *feature-ID* yang dikenali didefinisikan sebagai batas atas dan batas bawah dari setiap sumbu koordinat acuan. Damayanti (1995) menyatakan ketergantungan antara dua *feature* dibentuk dengan cara memeriksa batas-batas kedua *feature*, jika ada batas yang memiliki nilai yang sama maka kedua *feature* memiliki ketergantungan satu sama lain. Semua *feature* dikenali sebagai kandidat *child*, dan proses pencarian *parent* berdasarkan konsep *upper* dan *lower feature*. Jika *upper feature-ID* A berhubungan dengan *lower* dari *feature-ID* B, maka *feature-ID* B merupakan *parent* dan *feature-ID* A sebagai *child*. Konsep *parent-child* ini mendasari pemikiran untuk melakukan pengurutan proses pengerjaan pemesinan.



### 4.3 Implementasi dan Analisis Komponen

- Analisis Komponen 1 : Meja Atas (*Bed*)



Komponen ini memiliki karakteristik khusus yaitu pada posisi dari *feature* standar yang berhimpit. Keadaan ini ditunjukkan oleh kedua buah lubang pada setiap ujung dari komponen. Hal ini menjadi khusus karena bentuk gabungan yang tidak mengubah *loop* yang ada pada masing-masing *feature*.

Hasil keluaran model pengenalan *feature* untuk komponen ini menunjukkan bahwa dua buah volume potong berbentuk silinder di setiap ujung komponen didefinisikan sebagai 1 *blind hole* dan 1 *through hole*. Untuk mengetahui keadaan bentuk yang berhimpit ini, perencanaan proses harus melakukan perhitungan dari data geometri *feature-feature* standard hasil pengenalan.

Hasil keluaran model perencanaan proses berbasis *feature* menunjukkan komponen ini memiliki 20 buah *feature* yang terdiri atas 4 buah *blind hole planar base*, 4 buah *through hole* dan 12 buah *slot*. *Feature-feature* yang terdapat pada komponen ini memiliki 5 buah normal, sehingga proses pengerjaan komponen ini meliputi 5 kali *setup*. Operasi-operasi yang akan dilakukan terdiri atas *Mill* dan *Drill* dengan *tool* usulan sesuai dengan diameter lubang dan lebar *slot*.

- Analisis Komponen 2 : Meja Bawah (*Base plate*)



Pengenalan *feature* pada komponen ini hanya melibatkan bentuk-bentuk *independent*. *Feature-feature* standar yang terdapat pada komponen ini adalah *through hole* dan *slot no fillet*. Hasil keluaran model pengenalan *feature* menunjukkan bahwa komponen ini memiliki 8 buah *feature* yang terdiri atas 4 buah *slot no fillet* dan 4 buah *through hole*.

Hasil keluaran model perencanaan proses berbasis *feature* pada komponen ini menunjukkan bahwa *feature-feature* yang terdapat pada komponen ini memiliki 2 normal. Sedangkan pada tahap pengurutan proses digunakan 1 normal yang mewakili seluruh *feature* yang ada. Operasi-operasi yang dilakukan dalam pengerjaan komponen ini terdiri atas Mill dan Drill, dengan menggunakan 2 jenis pahat yaitu *end milling cutter* diameter 20 dan *drill cutter* diameter 10.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Implementasi dan analisis terhadap komponen uji memberikan beberapa kesimpulan antara lain :

- Perencanaan proses yang dikembangkan dikhususkan bagi komponen berbentuk prismatic dengan material awal (*blank material*) berupa *block*.
- Rencana proses standar yang dihasilkan terbatas pada *features* standar yang dapat dikenali dari proses pengenalan *feature*.
- Pengembangan data *feature* standar membuka peluang besar dalam meningkatkan kemampuan pengenalan bentuk.
- Dua *feature* standar yang berhimpit dapat dikenali jika *loop* pada masing-masing *feature* standard tidak berubah.
- Data output hasil dari proses pengenalan masih harus diolah lebih lanjut untuk beberapa proses tertentu dalam perencanaan proses.
- Model CAPP yang dikembangkan masih menggunakan aktivitas pemasukan data input secara manual.

### 5.2 Saran

Pengembangan selanjutnya dapat berupa :

- Pengembangan cara lain untuk memperoleh informasi mengenai *features* dari suatu gambar rancangan dengan menggunakan suatu *interface* CAD otomatis yang menggunakan pendekatan *design by features*.
- Pengembangan perencanaan proses untuk bentuk-bentuk prismatic.
- Pengembangan perencanaan proses yang menggunakan suatu *interface* CAD yang otomatis dan pembuatan program komputasi untuk pemasukan data input.
- Pengembangan perencanaan proses yang menghasilkan urutan pengerjaan yang dilengkapi dengan berbagai parameter pemesinan dan suatu program NC.

## DAFTAR KEPUSTAKAAN

- Anton, Howard, Aljabar Linier Elementer, Edisi Kelima, Penerbit Erlangga, Jakarta, 1995.
- Bedworth, David D., Henderson, Mark R., & Wolfe, Philip M., *Computer Integrated Design and Manufacturing*, Mc Graw-Hill International Edition, 1991.
- Case, K., & Gao, J., *Feature Technology : an Overview*, International Journal of Computer Integrated Manufacturing, 1993.
- Chang, Tien-Chien, *Expert Process Planning for Manufacturing*, Addison-Wesley Publishing Company Inc., USA, 1990.
- Chang, Tien-Chien, Wysk, Richard A., & Wang, Hsu-Pin, *Computer Aided Manufacturing*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1991.
- D. Renaldy, Pengembangan Perangkat Lunak Pengenalan Bentuk Gambar Rancangan pada AutoCAD, Thesis, 1997.
- Groover, M., *Automation Production System and Computer Aided Manufacturing*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1989.
- Hoshi, Tetsunaro, *Developmental Research on CAD/CAM Technology for Future Automation of Job Shop Mechanical Component Fabrication*, Toyohashi University of Technology (TUT), Toyohashi, Japan, 1994.
- Jablonski, S., Reinwald, B., & Ruf, T., *Integration of Process Planning and Job Shop Scheduling for Dynamic and Adaptive Manufacturing Control*, Departemen of Computer Science VI (Database Systems), University of Erlangen-Nuernberg, Erlangen, F.R.G., 1990.
- Kusiak, A., *Intelligent Manufacturing Systems*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1990.
- Marefat, Michael, & Kashyap, R. L., *Automatic Construction of Process Plans from Solid Model Representations*, IEEE Transactions on Systems, Man and Cybernetics, 1992.
- McMahon, Chris, & Browne, Jimmie, *CAD/CAM From Principles to Practice*, Addison-Wesley Publishing Company, 1993.
- Sasmita, Adju, Drs., *AutoCAD 13 for Windows*, PT. Elex Media Komputindo, Jakarta, 1999.
- Salomons, O.W., *Feature Recognition*, Thesis, 1998.
- Subagya, Studi Awal *Computer Aided Design*, Teknologi Kelompok, dan Perencanaan Kebutuhan Material, Tugas Akhir, Teknik Industri FTI-ITB, Bandung, 1990.
- Toha, Isa Setiyah, Ariastuti, Retna, Pengenalan *Feature* Komponen Menggunakan Model *Face Edge Graph*, Proceeding Seminar CAD/CAM, ITB Bandung, 1995.
- Toha, Isa Setiyah, Damayanti, Dina, Perencanaan Proses Berbasis *Features*, Proceeding Seminar CAD/CAM, ITB Bandung, 1995.
- Tolaram, Nanik, Pemrograman dengan Turbo C++, Penerbit Andi Offset, Yogyakarta, 1991.

Zakaria, Endang Ahmad, Perancangan dan Pengembangan Prototype Perencanaan  
Proses Berbasis Komputer dengan Metoda *Variant*, Tugas Akhir, Teknik  
Industri FTI-ITB, Bandung, 1993.  
Zeid, Ibrahim, CAD/CAM Theory and Practice, Mc. Graw-Hill Inc., 1995.