

PEMBUATAN PERANGKAT LUNAK CAM UNTUK MESIN CNC FREIS 3 SUMBU PADA AutoCAD

Sigit Yoewono Martowibowo⁽¹⁾ dan Wildan Rojuli Daa'i⁽²⁾

⁽¹⁾Laboratorium Proses Pemesinan, Departemen Teknik Mesin FTI-ITB
Jl. Ganesa 10, Bandung 40132

⁽²⁾PT. LG Innotek Indonesia, Bekasi International Industrial Estate,
Blok C-8 No.12&12A, Lippo Cikarang, Bekasi 17550

Ringkasan

Penelitian ini bertujuan untuk membuat perangkat lunak CAM untuk mesin freis CNC 3 sumbu pada AutoCAD release 2000 atau yang lebih baru menggunakan bahasa pemrograman Visual Basic for Application (VBA). Proses pemesinan dilakukan pada bidang X-Y menggunakan pahat endmill. Masukan yang diperlukan oleh perangkat lunak adalah gambar 3-D yang merupakan model yang akan dimesin. Keluaran perangkat lunak adalah file ASCII yang merupakan data lintasan pahat dalam bentuk G-Code. Perangkat lunak juga menyediakan fasilitas untuk mendekati bentuk spline dan elips dengan bentuk garis yang saling berhubungan dengan tetap memperhatikan batas toleransi geometrik profil garis.

Abstract

This study is aimed to produce CAM software run under AutoCAD 2000 or latest version using Visual Basic for Application (VBA) program. Machining processes are carried out using endmill cutter at X-Y plane. Input of this software is 3-D drawings of the model. The output of this software are G-Code of cutting tool path. The software also provides facilities to approach spline and ellips with connected lines but still considering geometric tolerances of line profile.

Keyword: AutoCAD, CAD/CAM, NC Program, G-Code, Spline, VBA, CNC milling machine

1. PENDAHULUAN

Perkembangan sangat pesat dalam bidang teknik terjadi seiring dengan waktu. Matematika, komputer, analisis numerik, dan keahlian desain merupakan persyaratan yang harus dimiliki oleh seorang desainer. Selama ± 30 tahun terakhir, pemanfaatan teknologi dapat terasa, mulai dari penggunaan tabel matematika dalam perhitungan kemudian meningkat menggunakan kalkulator dan PC (*Personal Computer*). Pada 15 tahun terakhir, PC mulai dimanfaatkan dalam desain grafik yang membuat pengaruh besar pada proses desain dan manufaktur [1].

CAD (*Computer Aided Design*) merupakan perangkat lunak yang dimanfaatkan untuk mempermudah proses desain, sedangkan CAM (*Computer Aided Manufacturing*) merupakan perangkat lunak untuk mempermudah proses manufaktur. Sebagian perangkat lunak hanya mengkhususkan diri dalam proses desain, sebagai contoh AutoCAD dan sebagian lagi merupakan perangkat lunak yang mengintegrasikan CAD/CAM di dalam sebuah program, sebagai contoh : ProE, MasterCAM, UniGraphic, dsb. Bila pengguna AutoCAD akan melakukan proses manufaktur maka ia harus beralih ke perangkat lunak lain yang memiliki fasilitas CAM.

Dalam pengoperasian mesin CNC (*Computer based Numerical Control*), diperlukan suatu *part program*

(sering juga disebut *NC program/program NC*). Program NC merupakan sejumlah urutan perintah logik yang dibuat untuk suatu jenis mesin perkakas NC dalam rangka pembuatan suatu komponen mesin/peralatan [2]. Tata cara penulisan program NC dilakukan sesuai dengan standar yang ditetapkan oleh ISO, sehingga program NC sering disebut sebagai program dalam kode G (*G-Code*). Pemrograman NC dapat dilakukan dalam tiga cara, yaitu:

1. Pemrograman manual (*manual programming*) yang dilakukan pada mesin NC (*on line, machine tool level, manual NC input*).
2. Pemrograman dengan komputer bantu (*computer assisted programming*) yang dilakukan pada komputer lain (*off line*).
3. Pemrograman terintegrasi dalam sistem CAD/CAM yang memanfaatkan basis data geometrik (*internal geometrical data base*) yang dihasilkan oleh perancang produk dengan sistem CAD.

Pemrograman manual umumnya digunakan pada pembuatan produk dengan geometri relatif sederhana. Cara ini memungkinkan terjadinya banyak kesalahan dan memerlukan waktu pengerjaan yang cukup lama. Pemakaian komputer bantu pada pemrograman, umumnya dilakukan untuk produk dengan geometri yang lebih kompleks. Cara ini menggunakan komputer untuk mendefinisikan gerakan pahat sepanjang elemen geometrik produk berdasarkan gambar teknik. Cara

kedua ini dapat mempersingkat waktu pembuatan program, tetapi memerlukan pemrogram yang ahli untuk menginterpretasikan bentuk geometri produk ke dalam lintasan pahat yang diperlukan. Cara yang paling baik adalah dengan menggunakan sistem CAD/CAM yang mana sistem ini mengintegrasikan proses desain dan proses manufaktur.

Dengan memberikan fasilitas CAM pada perangkat lunak AutoCAD maka diharapkan proses produksi dapat berjalan secara berkesinambungan yang pada akhirnya akan meningkatkan produktivitas.

Penelitian ini bertujuan untuk menambahkan fasilitas CAM pada perangkat lunak AutoCAD 2000 (atau *release* yang lebih baru). Fasilitas CAM ini dapat menghasilkan program NC (dalam bentuk *G-Code/G-ISO*) dengan masukan berupa geometri produk 3-D padat (dalam bentuk gambar AutoCAD, *.dwg), dengan sifat-sifat sebagai berikut :

1. Data yang dihasilkan dapat diintegrasikan dengan perangkat lunak CAM yang lain. Hal ini terjadi karena program NC yang dihasilkan ditulis dalam bentuk *file* ASCII, yang mana setiap perangkat lunak CAM memiliki kemampuan untuk membaca data *file* ASCII.
2. Bersifat *user friendly*. Untuk mengoperasikannya tidak diperlukan waktu yang lama (dengan menganggap pengguna telah mahir menggunakan AutoCAD).
3. Merupakan program yang terintegrasi di dalam AutoCAD.
4. Memiliki kemampuan untuk melakukan pendekatan bentuk tak beraturan (*spline*) dan elips.

2. MESIN PERKAKAS NC

Mesin perkakas NC serupa dengan mesin perkakas konvensional, terdiri atas berbagai jenis sesuai dengan proses yang bisa dilaksanakan. Perbedaan utama, yaitu mesin perkakas NC telah dilengkapi dengan *controller* yang mengatur gerakan mesin perkakas NC.

Dibandingkan mesin perkakas konvensional, konstruksi mesin perkakas NC secara umum lebih baik dengan transmisi daya yang kompak dan pemakaian elemen penggerak serta pembimbing yang lebih teliti. Mesin perkakas NC juga memiliki keunggulan karena lebih teliti (*accurate*), lebih tepat (*precision*), mampu membuat beragam bentuk produk, dan lebih produktif.

❑ **Ketelitian.** Mesin perkakas NC memiliki kemampuan untuk menghasilkan produk dengan ukuran teliti serta daerah toleransi geometrik yang sempit. Hal ini disebabkan karena konstruksi mesin perkakas NC yang lebih baik, sistem pendeteksian posisi yang teliti, dan memiliki sistem pengkompensasi kesalahan posisi dibandingkan dengan mesin perkakas konvensional.

❑ **Ketepatan.** Memiliki arti bahwa pengerjaan oleh mesin perkakas NC dapat diulang tanpa kesalahan

sesuai dengan program NC yang dibuat. Operator yang terampil dalam hal ini tidak diperlukan tetapi keahlian pemrogram yang diperlukan.

❑ **Kompleksitas.** Mesin perkakas NC memiliki kemampuan untuk menghasilkan produk dengan kerumitan geometri produk yang tinggi.

❑ **Produktif.** Mengingat pergerakan mesin perkakas NC dilakukan secara otomatis berdasarkan program yang telah dibuat sehingga hal ini akan meningkatkan waktu produksi secara keseluruhan. Penggunaan ATC (*Automatic Tool Changer*) dan APT (*Automatic Pallette Changer*) pada mesin perkakas NC juga berguna untuk mempercepat waktu pergantian dan *set-up* benda kerja.

3. LINGKUNGAN AutoCAD [3]

3.1 Elemen Gambar

Elemen gambar yang disediakan oleh AutoCAD untuk membuat suatu model gambar disebut sebagai entiti (*entity*). Entiti tersebut antara lain : titik (*point*), garis (*line*), busur (*arc*), persegi panjang (*rectangular*), lingkaran (*circle*), elips (*ellipse*), *polyline*, *spline*, *3Dface*, *3Dsolid*, *region*, dsb. Entiti ini yang membentuk suatu gambar dalam AutoCAD.

Setiap elemen gambar memiliki properti (*properties*) tertentu. Dengan menentukan properti-properti ini kita dapat membuat suatu entiti dalam AutoCAD. Misalnya, apabila dalam pembuatan entiti garis maka perlu ditentukan titik awal (*startpoint*) dan titik akhir (*endpoint*) dari garis tersebut. Titik awal dan titik akhir tersebut merupakan properti yang dimiliki oleh entiti garis. Untuk membuat sebuah garis sederhana cukup didefinisikan properti titik awal dan titik akhir garis. Modifikasi lebih lanjut dapat dilakukan dengan mengubah properti lainnya seperti warna garis, tebal garis, *layer*, dsb. Sebagian properti elemen gambar tidak dapat diubah dan secara otomatis telah diatur oleh AutoCAD.

3.2 Pemodelan dalam AutoCAD

Pembuatan model dari obyek 3-D pada bidang representasi 2-D dapat dilakukan oleh AutoCAD. Pemodelan menggunakan obyek 3-D lebih sulit dan membutuhkan waktu lama dibandingkan dengan pemodelan obyek 2-D, akan tetapi pemodelan ini memiliki beberapa keunggulan, yaitu :

- Pilihan pandangan yang lebih luas.
- Pandangan 2-D dapat dihasilkan secara otomatis berdasarkan 3-D.
- Memberikan pandangan obyek yang lebih realistik (nyata).
- Kemudahan membuat animasi (gambar bergerak).
- Kemudahan melakukan perhitungan teknik.

Jenis pemodelan yang dapat dilakukan oleh AutoCAD terdiri atas 3 bentuk, yaitu : *wireframe*, *surface* dan *solid*. Masing-masing pemodelan memiliki teknik pembuatan dan *editing* tersendiri.

a. *Wireframe modelling*

Model *wireframe* adalah penggambaran obyek 3-D menggunakan rangka. Model *wireframe* hanya terdiri atas titik, garis, dan kurva yang menunjukkan sudut-sudut dalam entiti model.



Gambar 1 *Wireframe modelling* [3]

b. *Surface modelling*

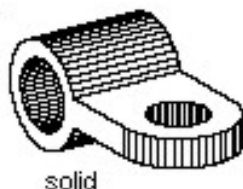
Pada model *surface* pendefinisian obyek 3-D selain dilakukan dengan titik, garis, dan kurva juga menambahkan permukaan di antara garis/kurva yang membentuk area tertutup (*close area*). Permukaan tersebut didefinisikan menggunakan *polygonal mesh* yang merupakan bidang pendekatan.



Gambar 2 *Surface modelling* [3]

c. *Solid modelling*

Pemodelan obyek 3-D *solid* dilakukan dengan membentuk benda padat. Obyek 3-D dapat dibuat menggunakan bentuk-bentuk dasar 3-D, seperti : *boxes, cones, cylinders, spheres, wedges* dan *tori (donuts)*. Bentuk-bentuk dasar tersebut dapat dikombinasikan untuk membuat benda padat yang kompleks dengan cara penggabungan, pengurangan atau perpotongan. Entiti akan memiliki properti seperti massa, volume, radius gravitasi, dsb. Obyek *solid* dapat dibuat dengan operasi “*extrude*” ataupun dengan operasi “*revolve*” terhadap obyek 2-D berbentuk kurva tertutup.

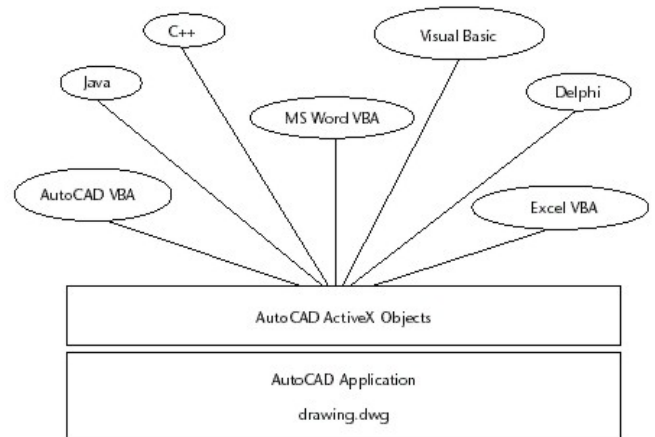


Gambar 3 *Solid modelling* [3]

3.3 AutoCAD ActiveX

AutoCAD menyediakan suatu fasilitas untuk melakukan manipulasi obyek-obyek dalam lingkungan AutoCAD dengan menggunakan bahasa pemrograman, baik dari dalam maupun dari luar lingkungan

AutoCAD. Mekanisme ini terjadi dengan cara mengekspos obyek AutoCAD ke luar lingkungan AutoCAD melalui perantaraan fasilitas *ActiveX*. Suatu obyek yang telah terekspos berarti akan dapat diakses oleh bahasa pemrograman ataupun aplikasi lain, seperti Microsoft Word atau Microsoft Excel melalui *Visual Basic for Application (VBA)*.



Gambar 4 AutoCAD *ActiveX* [3]

Penerapan teknologi *ActiveX* pada AutoCAD memberikan dua manfaat, yaitu :

1. AutoCAD dapat diakses oleh berbagai bahasa pemrograman. Sebelum diterapkannya metode *ActiveX* ini, AutoCAD hanya dapat diakses oleh bahasa pemrograman AutoLISP dan C++.
2. Mempermudah pertukaran data antar aplikasi berbasis window.

Obyek adalah bagian utama dalam aplikasi *ActiveX*. Setiap obyek yang terekspos merupakan representasi sebuah bagian dalam aplikasi AutoCAD. Jenis obyek yang dapat diekspos sangat beragam, diantaranya :

- ❖ *Graphical objects*; seperti garis, busur, teks, dan dimensi.
- ❖ *Style setting*; seperti jenis garis dan jenis dimensi.
- ❖ *Organizational structure*; seperti *layer, groups*, dan *blocks*.
- ❖ *Drawing display*; seperti *view* dan *viewports*.
- ❖ *The drawing dan AutoCAD application*.

3.4 Visual Basic Application (VBA)

VBA adalah pemrograman berorientasi obyek yang dirancang untuk memberikan kemampuan serupa dengan Visual Basic. Berbeda dengan Visual Basic, VBA hanya dapat bekerja dalam lingkungan tempat ia didefinisikan. Sebagai contoh, Microsoft Word VBA hanya dapat bekerja dalam lingkungan Microsoft Word, AutoCAD VBA hanya dapat bekerja dalam lingkungan AutoCAD, dsb. Untuk menjalankan AutoCAD VBA maka perangkat lunak AutoCAD harus dijalankan terlebih dahulu. Aplikasi VBA dibandingkan terhadap Visual Basic akan memberikan keuntungan berupa eksekusi program yang lebih cepat. Implementasi VBA dalam AutoCAD memberikan keuntungan sebagai berikut :

1. VBA beroperasi sejalan dengan AutoCAD. Hal ini memberikan manfaat berupa eksekusi program yang lebih cepat.
2. Fasilitas *feedback* yang cepat. Dalam hal ini pemrogram VBA secara langsung dapat melihat perubahan yang terjadi dalam lingkungan AutoCAD.
3. Fleksibilitas dalam implementasi *project* di VBA dengan adanya pilihan *standalone* atau *embedded project*.

Pengiriman pesan dari VBA ke AutoCAD dilakukan melalui perantara (*interface*) *ActiveX automation*. Koneksi antara AutoCAD, *ActiveX automation*, dan VBA tidak hanya memberikan fasilitas untuk mengubah obyek pada AutoCAD, tetapi dapat juga digunakan untuk melakukan pengiriman dan penerimaan data dengan aplikasi yang lain.

Tiga elemen utama pada pemrograman AutoCAD VBA. Pertama, lingkungan AutoCAD yang terdiri atas bermacam-macam obyek yaitu entiti, data, dan *command*. Karena AutoCAD dirancang sebagai aplikasi *open-architecture* dengan berbagai *interface*, maka pemahaman menyeluruh tentang obyek AutoCAD sangat diperlukan. Kedua, AutoCAD *ActiveX* yang menghasilkan komunikasi antara VBA dengan obyek dalam AutoCAD. Ketiga, VBA *programming*, yang memiliki *objects*, *keywords* dan *constants* tersendiri. VBA *programming* akan menjalankan, mengontrol, dan memeriksa program.

Secara umum tiap obyek di dalam AutoCAD memiliki atribut yang menyertainya, yaitu *properties*, *methods*, dan *events*.

❑ **Properties**

Merupakan karakteristik yang dimiliki oleh obyek. Sebagai contoh, obyek garis akan memiliki properti warna, panjang, titik awal, dan titik akhir. Properti merupakan sifat yang menyertai obyek, sebagian properti ditentukan oleh pengguna AutoCAD seperti jenis *layer* dan tebal garis. Sebagian lainnya ditentukan atau dihasilkan secara otomatis oleh AutoCAD seperti properti *objectID*, *length*, *delta*, dsb.

❑ **Methods**

Operasi *methods* disediakan oleh AutoCAD untuk melakukan perubahan atau penambahan obyek AutoCAD, seperti metode *AddCircle* yang digunakan untuk membuat lingkaran (*circle*) pada obyek *modelspace*, *paperspace*, atau *blocks*. Masukan yang dibutuhkan untuk masing-masing *methods* berbeda-beda sesuai dengan obyeknya. Untuk setiap *methods* hanya terdapat satu cara untuk menjalankannya. Misalnya dalam metode *AddCircle* (metode yang dipakai untuk membuat entiti lingkaran) kita diharuskan memberikan masukan berupa koordinat titik pusat (*centerpoint*) dan radius lingkaran.

❑ **Events**

Events adalah pesan yang dikirim oleh AutoCAD untuk memberikan informasi kepada pengguna tentang suatu kejadian di dalam lingkungan

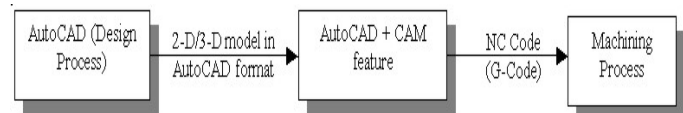
AutoCAD. Misalnya, ketika membuka *drawing* baru maka *Event BeginOpen* terjadi. *Event* ini mengandung nama gambar yang sedang dibuka.

4. PERANCANGAN PROGRAM CAM

Penelitian ini bertujuan menghasilkan fasilitas CAM pada AutoCAD untuk mesin freis CNC 3 sumbu (*CNC milling machine 3-axes*) dengan pengontrol Fanuc 6MB dan diberi nama G-Mill 2002. Mengingat luasnya ruang lingkup permasalahan maka penelitian ini dibatasi beberapa faktor sebagai berikut :

- ❖ Pahat freis dibatasi hanya untuk jenis pahat *endmill*.
- ❖ Bidang pemotongan dilaksanakan pada bidang X-Y.
- ❖ Pemotongan dilakukan dengan *single pass*.

Fasilitas CAM yang akan dibuat bekerja di dalam lingkungan AutoCAD. Masukan yang dibutuhkan adalah gambar AutoCAD dalam bentuk 3-D *solid* serta parameter proses pemesinan yang dipilih.



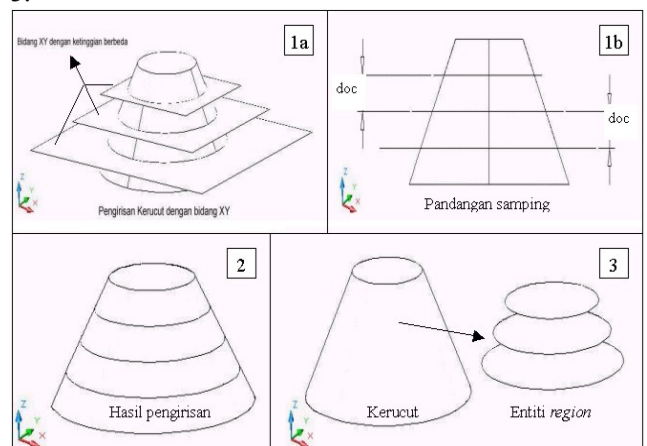
Gambar 5 Aliran pemrosesan data

Ide dasar pada penelitian ini adalah membuat entiti garis dan busur dalam AutoCAD berdasarkan gambar 3-D, yang mana koordinat dan data entiti tersebut akan diolah menjadi data lintasan pahat dalam bentuk *G-Code*. Kemudian diperlukan proses lebih lanjut untuk menghasilkan entiti representasi lintasan pahat karena data gambar 3-D merupakan gambar obyek hasil pemesinan dan bukan gambar lintasan pahat.

4.1. Tahap Pembentukan Lintasan

4.1.1 Section

Sebagai contoh dipilih bentuk geometri yang akan dimesin adalah kerucut. Proses *section* akan mengiris bentuk kerucut ini dengan beberapa bidang X-Y dengan ketinggian sesuai dengan *doc* (*depth of cut*) yang telah ditentukan pengguna, lihat Gb.(6)-1a dan 1b. Hasil pengirisan kerucut menghasilkan entiti *region* yang memiliki bentuk kontur terluar sama dengan kerucut pada ketinggian bidang X-Y, lihat Gb.(6)-2 dan 3.



Gambar 6 Proses pengirisan (*section*)

Entiti *region* hasil operasi *section* kemudian di-*explode* menjadi entiti tunggal. Hasil *explode* dapat terdiri atas entiti garis, busur, elips atau *spline*. Entiti garis dan busur tidak memerlukan pengolahan lebih lanjut, sedangkan entiti elips dan *spline* perlu didekati dengan metode entiti busur atau entiti garis.

4.1.2 Pendekatan *spline* dan elips

Entiti *spline* termasuk kurva sintetik, yaitu bentuk geometri yang tidak memiliki representasi dalam persamaan matematik. Entiti ini dibutuhkan untuk membentuk geometri yang tidak beraturan. Karena keterbatasan dalam pemrograman *G-Code* yang tidak dapat melakukan pergerakan *spline* dan hanya dibatasi oleh pergerakan linear dan pergerakan sirkular, maka dibutuhkan suatu metode yang dapat mendekati geometri *spline* dengan geometri garis atau busur.

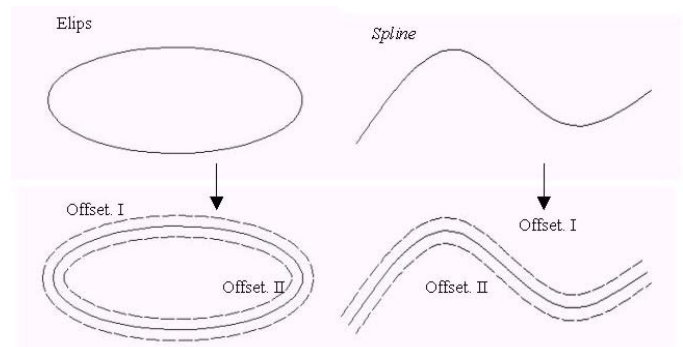
Pergerakan pahat relatif terhadap benda kerja pada mesin perkakas NC tidak dapat mengikuti geometri berbentuk elips. Entiti elips merupakan entiti analitik yang dapat dinyatakan dalam persamaan matematik. Persamaan matematik elips berbeda dengan persamaan matematik untuk garis atau busur. Hal ini menyebabkan entiti elips tidak dapat diubah menjadi beberapa entiti garis atau busur. Oleh karena itu, diperlukan metode pendekatan geometri elips dengan geometri garis atau busur.

Untuk membuat sebuah entiti busur diperlukan 4 buah data, yaitu : sudut awal (*start angle*), sudut akhir (*end angle*), koordinat titik pusat, dan radius busur. Untuk membuat sebuah entiti garis diperlukan 2 buah data, yaitu : koordinat titik awal dan koordinat titik akhir. Metode pendekatan entiti elips/*spline* menggunakan entiti busur tentunya akan lebih kompleks dibandingkan terhadap pendekatan menggunakan entiti garis. Oleh karena itu, dalam penelitian ini metode pendekatan elips/*spline* dilakukan dengan menggunakan entiti garis.

Dengan memperhatikan toleransi profil garis maka entiti elips/*spline* dapat didekati dengan beberapa entiti garis. Entiti garis yang dihasilkan harus terletak di dalam daerah toleransi profil garis entiti *spline* dan elips. Pendekatan dilakukan dengan cara membagi entiti *spline* atau elips menjadi beberapa bagian menggunakan bantuan lingkaran pada radius yang telah ditentukan.

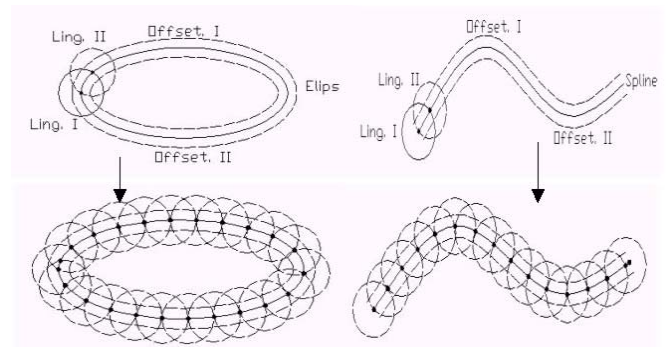
Tahapan metode pendekatan entiti elips/*spline* adalah sebagai berikut :

1. Meng-*offset* entiti elips/*spline* dengan jarak *offset* $\frac{1}{2}$ dari besar daerah toleransi. Peng-*offset*-an dilakukan dua kali pada arah yang berbeda (lihat Gb.7). Hasil operasi akan menjadikan entiti elips/*spline* diapit oleh entiti hasil *offset*.
2. Penggambaran lingkaran bantu (Ling. I) dengan radius lingkaran yang telah ditentukan dan titik pusat (*center*) lingkaran berada pada titik awal



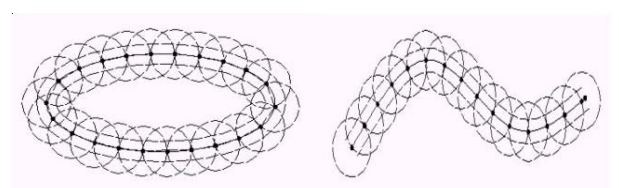
Gambar 7 Offset elips/*spline* sebesar $\frac{1}{2}$ nilai toleransi profil garis

(untuk elips) atau pada titik *vertex(0)* (untuk *spline*). Perpotongan (*intersection*) Ling. I dengan elips/*spline* akan menghasilkan satu atau dua buah titik perpotongan. Salah satu titik perpotongan tersebut akan dijadikan titik pusat lingkaran berikutnya (Ling. II) dengan jarak radius yang sama dengan radius Ling. I. Proses ini terus dilakukan sampai lingkaran bantu membagi keseluruhan entiti elips/*spline*. Apabila Ling. I tidak memotong entiti elips/*spline*, berarti radius lingkaran bantu yang dipilih terlalu besar sehingga perlu didefinisikan radius lingkaran bantu yang lebih kecil. Pada program G-Mill 2002, pendefinisian ulang radius lingkaran bantu dilakukan secara otomatis.



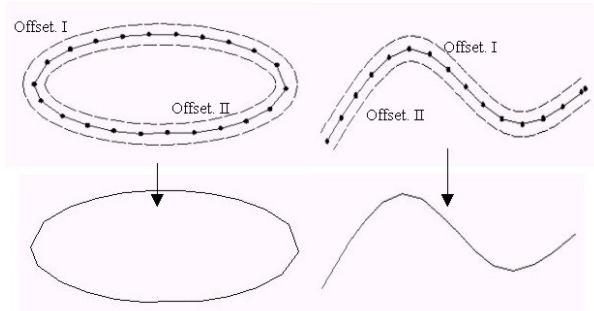
Gambar 8 Penggambaran lingkaran bantu

3. Interseksi lingkaran bantu dengan entiti elips/*spline* akan menghasilkan titik perpotongan. Antara koordinat titik perpotongan yang berdekatan akan dihubungkan dengan entiti garis. Apabila garis yang dibentuk ternyata memotong *Offset I* atau *Offset II*, maka proses pendekatan entiti elips/*spline* akan diulang dengan mendefinisikan radius lingkaran bantu yang lebih kecil. Pada program G-Mill 2002, pendefinisian ulang radius lingkaran bantu dilakukan secara otomatis.



Gambar 9 Penggambaran garis melalui titik perpotongan

4. Menghapus entiti lingkaran bantu, *Offset I*, *Offset II*, dan entiti elips/*spline*, sehingga dihasilkan beberapa entiti garis yang mengikuti kontur entiti elips/*spline* dan tetap berada di dalam daerah toleransi profil garis.

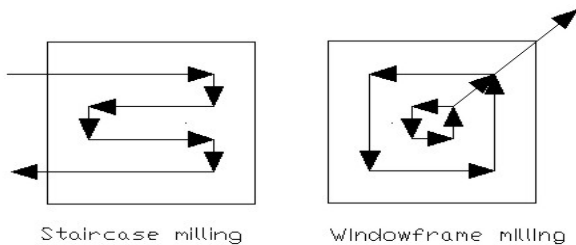


Gambar 10 Entiti garis hasil pendekatan elips/*spline*

Setelah seluruh elips dan *spline* diubah menjadi entiti garis maka di dalam lingkungan AutoCAD hanya terdapat entiti garis, busur dan lingkaran saja.

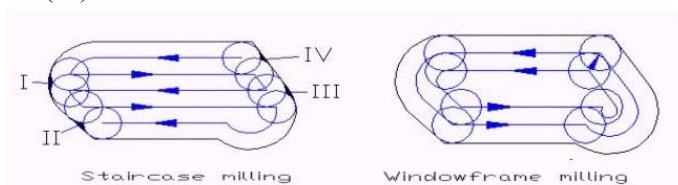
4.1.3 Offset rongga

Berdasarkan pada pergerakan mata pahat, maka proses pemesinan secara umum dapat dibedakan menjadi *staircase* dan *windowframe*. *Staircase milling* biasa juga disebut *wafering* atau *sweeping*, melakukan proses pemesinan dengan cara mata pahat bergerak secara paralel terhadap permukaan benda kerja yang dihasilkan sehingga seluruh permukaan selesai diproses. Pada pergerakan *windowframe* (biasa juga disebut *bull's eye milling*), proses pemesinan dilakukan dengan cara mata pahat bergerak spiral [4].



Gambar 11 *Staircase milling* dan *windowframe milling*

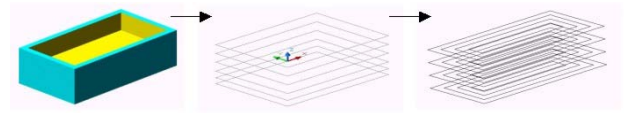
Offset digunakan untuk menghasilkan lintasan pahat *windowframe*. Obyek yang di-*offset* adalah entiti lingkaran atau entiti *polyline* (entiti garis saling berhubungan). Pemilihan metode *windowframe* daripada metode *staircase* dapat dijelaskan pada Gb.(12).



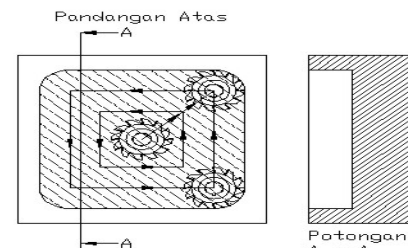
Gambar 12 Perbandingan *staircase milling* dan *windowframe milling*

Pada lintasan *staircase milling* terlihat daerah yang tidak terlewati pahat (tidak terpotong) yaitu bagian I, II, III, dan IV sedangkan pada lintasan *windowframe milling* seluruh daerah terlewati pahat (terpotong).

Peng-*offset*-an dilakukan hingga dihasilkan entiti hasil *offset* yang memiliki luas area lebih kecil dibandingkan terhadap luas daerah obyek yang di-*offset*. Selanjutnya, obyek hasil *offset* akan di-*offset* lagi dengan jarak *offset* yang sama dan akan menghasilkan obyek yang mempunyai luas daerah lebih kecil. Operasi ini terus dilakukan (*looping*) sampai operasi *offset* tidak dapat dilakukan lagi. Operasi di atas diulang untuk setiap kedalaman potong yang ditentukan.



Gambar 13 Operasi *offset*

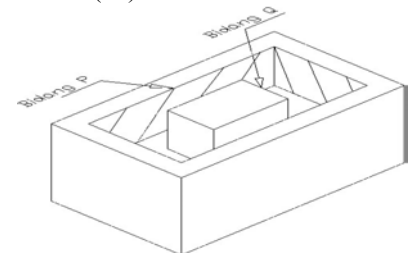


Gambar 14 Lintasan *windowframe* yang dihasilkan

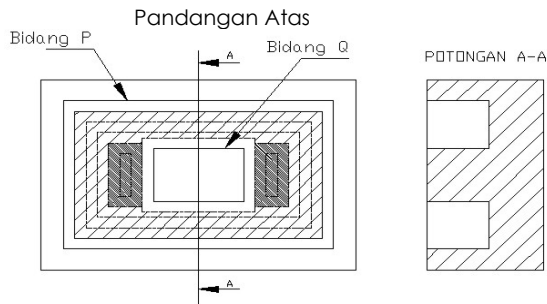
4.1.4 Offset rongga dengan pulau

Untuk bentuk geometri rongga dengan pulau di tengah, maka dibutuhkan operasi *offset* khusus. Operasi ini merupakan modifikasi dari operasi *offset* rongga dengan memperhatikan keberadaan pulau. Operasi *offset* pertama dilakukan dengan cara melakukan *looping offset* terhadap entiti *polyline* atau lingkaran terluar yang memiliki luas area terbesar. *Offset* dilakukan untuk menghasilkan obyek dengan luas daerah yang lebih kecil. Operasi *offset* ini dilakukan terus sampai obyek hasil operasi *offset* memotong/berinterseksi dengan obyek pulau.

Dari obyek yang memotong pulau dibuat sebuah *close entity* dengan menggunakan operasi pengurangan (*subtract*). Hasil *offset* yang memotong pulau akan dikenakan operasi *subtract* terhadap entiti pulau. Hasil operasi *subtract* adalah berupa *open entity*, yang kemudian diubah menjadi *close entity*. *Close entity* yang dihasilkan akan dikenakan operasi *offset* dengan jarak *offset* yang sama dan arah peng-*offset*-an yang menghasilkan luas daerah obyek yang semakin kecil. Entiti hasil operasi *offset* menjadi acuan dalam pembuatan lintasan pahat. Untuk lebih jelasnya dapat dilihat pada Gb. (16).



Gambar 15 Fitur rongga tengah dengan pulau



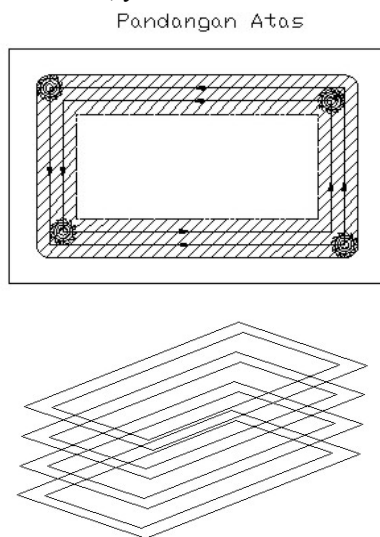
Gambar 16 Lintasan *windowframe* untuk rongga pulau

Keterangan : bidang P dan Q merupakan “bidang pemesian” yaitu bidang yang dihasilkan dan merupakan hasil akhir dari pemesian.

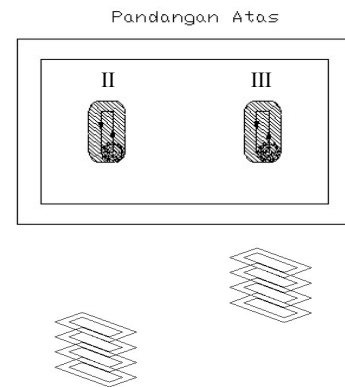
Fitur yang dipilih sebagai contoh adalah bentuk rongga dengan pulau ditengah. Kemampuan perangkat lunak G-Mill 2002 tidak terbatas pada bentuk geometri tersebut, dapat pula membuat fitur dengan bentuk geometri tak beraturan.

Tahapan pembuatan lintasan pahat untuk bentuk rongga dengan pulau adalah sebagai berikut :

1. Pembuatan lintasan pahat untuk rongga (sampai sebelum lintasan pahat memotong daerah pulau). Daerah yang diarsir merupakan daerah yang dilewati oleh lintasan pahat (Gb.17). Lintasan pahat digambarkan dengan garis putus-putus. Lintasan ini dapat dinamakan *windowframe* I. Masih terdapatnya bagian dari fitur yang tidak terlewati oleh lintasan pahat dibutuhkan beberapa lintasan pahat baru untuk membuang daerah “material sisa” (Gb.17).
2. Pembuatan lintasan pahat pada daerah “material sisa”. Pada daerah “material sisa” dibuat *windowframe* baru (bagian diarsir) yang melingkupi seluruh daerah sisa material (Gb.18). Pada fitur tersebut dibuat 2 *windowframe* tambahan, yaitu II dan III.



Gambar 17 Lintasan *windowframe* I



Gambar 18 Lintasan pahat untuk “material sisa”

Pada pembuatan program *G-Code*, setiap *windowframe* akan diproses secara terpisah. Hal ini berarti pemesian dilakukan secara bertahap mulai dari *windowframe* awal (*windowframe* I) dan dilanjutkan ke *windowframe* untuk pemotongan daerah “material sisa” (*windowframe* II dan III). Pahat akan bergerak ke bidang *retract* setelah menyelesaikan satu *windowframe* kemudian menyelesaikan *windowframe* berikutnya. Pada setiap *windowframe*, pahat akan bergerak mulai dari pusat sampai ke sisi terluar *windowframe*.

4.2. Lintasan *Positioning* dan *Retract*

4.2.1 Pemosisian (*positioning*)

Gerakan pemosisian diperlukan untuk menempatkan pahat pada titik awal pemotongan di setiap kedalaman potong tertentu. Perlu diperhatikan bahwa metode pemosisian ini harus dilakukan dalam waktu yang sesingkat mungkin dan menempuh jarak yang sependek mungkin. Namun demikian, selama pergerakan pahat relatif terhadap benda kerja, faktor keamanan tetap harus diperhatikan. Faktor keamanan yang perlu diperhatikan dalam hal ini adalah tabrakan (*coalition*) antara pahat dengan benda kerja, pahat dengan meja dan menjaga agar pergerakan pahat masih berada di dalam ruang kerja mesin perkakas NC.

Gerakan pemosisian dilakukan dalam tiga tahap sebagai berikut :

1. Pergerakan pahat menuju titik yang memiliki koordinat sumbu X dan Y sama dengan titik awal pemotongan. Gerakan ini dilakukan pada bidang *retract* (pada nilai Z yang telah ditentukan oleh pengguna). Pergerakan dilakukan dengan kode G00 (*rapid feed*).
2. Pergerakan pahat searah sumbu Z negatif (searah sumbu pahat), sampai pahat berada 2 mm di atas titik awal pemotongan. Gerakan dilakukan dengan kode G00 (*rapid feed*).
3. Pergerakan pahat searah sumbu Z negatif (searah sumbu pahat), sampai pahat berada di titik awal pemotongan. Gerakan dilakukan dengan kode G01 (*linear interpolation*) dengan menggunakan harga kecepatan makan tertentu.

4.2.2 *Retract*

Gerakan *retract* adalah pergerakan pahat menuju bidang *retract* (*retract plane*). Bidang *retract* didefinisikan sebagai bidang yang menjadi tempat posisi pahat setiap akhir proses pemotongan. Kode yang digunakan adalah G00 (*rapid feed*), karena pada gerakan menuju bidang *retract* pahat tidak melakukan proses pemotongan. Pergerakan pahat ke bidang *retract* juga berfungsi untuk menghindari tabrakan antara pahat dengan benda kerja, sehingga bidang *retract* harus terletak di luar benda kerja yang akan dimesin. Pada penelitian ini bidang *retract* merupakan bidang X-Y memiliki koordinat sumbu Z tertentu (harga Z ditentukan pengguna).

5. GENERATING

Proses *generating* akan membaca informasi entiti dalam AutoCAD dan mengubahnya menjadi informasi lintasan pahat dalam *G-Code*. Pengaturan parameter pemesinan, pahat, benda kerja, dan mesin perkakas NC ke dalam bentuk *G-Code* juga dilakukan di dalam modul *generating*. Proses ini dilakukan sepenuhnya oleh G-Mill 2002 dengan data pemesinan dari pengguna. Tampilan perangkat lunak G-Mill 2002 diperlihatkan pada Gb.(19). Hasil proses ini adalah file *G-Code* dalam bentuk ASCII (*texts file*) dan siap untuk dihubungkan dengan mesin CNC.

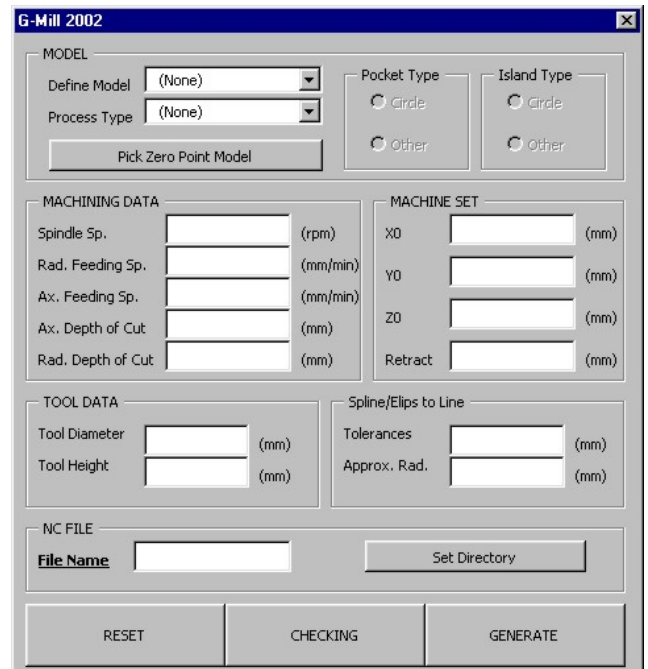
Pengujian perangkat lunak G-Mill 2002 dilakukan menggunakan perangkat lunak simulasi SURFCAM *Verify*. SURFCAM *Verify* mensimulasikan proses pemesinan pada mesin perkakas NC. Hasil pengujian menunjukkan hasil yang baik tanpa kesalahan pemesinan. Diharapkan di masa mendatang pengujian langsung pada mesin freis dapat dilakukan untuk dibandingkan dengan simulasi yang telah dilakukan.

6. KESIMPULAN

Penelitian ini telah menghasilkan :

1. Perangkat lunak CAM (G-Mill 2002) yang terintegrasi di dalam perangkat lunak AutoCAD, sehingga proses desain menggunakan CAD dapat diintegrasikan dengan proses manufaktur.
2. Tampilan G-Mill 2002 dibuat sesederhana mungkin dengan menampilkan sebuah *window* untuk seluruh proses manufaktur, sehingga perangkat lunak ini bersifat *user friendly*.
3. G-Mill 2002 dapat melakukan pendekatan geometri *spline* dan elips dengan geometri garis yang memenuhi daerah toleransi profil garis.
4. G-Mill 2002 beroperasi untuk mesin freis CNC 3 sumbu dengan keluaran berupa file *G-Code*.

Perangkat lunak CAM ini dirancang sebagai perangkat lunak yang terintegrasi dalam AutoCAD dan diprogram berdasarkan sistem modul, sehingga membuka kesempatan pengembangan lebih lanjut. Diharapkan perangkat lunak ini dapat menjadi pilihan bagi pengguna yang memerlukan sistem CAD/CAM sederhana (mudah dipelajari).



Gambar 19 Tampilan perangkat lunak CAM pada AutoCAD (G-Mill 2002)

7. DAFTAR PUSTAKA

1. Ibrahim, Zein, *CAD/CAM Theory and Practice*, Mc-Graw Hill, Inc, New-York, 1991.
2. Rochim, Taufiq, *Pemrograman NC*, ITB, Bandung, 2002.
3. Autodesk, *AutoCAD 2000 Help*, United States of America, 1992.
4. Chang, T.C, Wysk, R.A, Wang, H.P., *Computer Aided Manufacturing*, Prentice Hall Inc, New Jersey, 1998.
5. Wildan Rojuli Daa'i, *Pembuatan Perangkat Lunak CAM untuk Mesin CNC Freis 3 sumbu menggunakan AutoCAD 2000*, Tugas Sarjana, Departemen Teknik Mesin ITB, 2003.

