BAB II

PROSES PEMESINAN DENGAN *CAD CAM*

Langkah awal masuk kedalam basis program sistem CAM adalah mengindentifikasikan *project* / benda kerja dengan struktur kode dalam sistem koordinat, dimana pola yanng digunakan adalah *implementasi* dari struktur pengerjaan didalam ruang 3 dimensi dengan batasan volume tertentu yang selanjutnya akan dibahas sebagai *bounding box* (kotak *bounded*/simpan) dan koordinat didalamnya (*bounding box)* tersebut diimplementasikan kepada ukuran benda nyata yang diskalakan dalam point koordinat system. Beberapa langkah yang dilakukan dalam pemesinan dengan *Sistem CAM* diantaranya harus mampu :

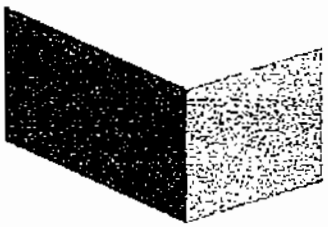
a. Mengerti struktur model faset dalam 3 dimensi obyek
b. Metode identifikasi *feature* (didalamya model faset dari *software* CAD dalam format STL file)
c. Informasi dari *normal vector* (*normal vector* dari poligon dalam kumpulan 3 vertek yang terkandung dalam stl tersebut)
d. Indentifikasi *volume feature* (*Solid*/ruang padatan atau *surface*/hanya permukaan) untuk proses lebih lanjut.

II.1. Struktur Model Faset dan 3 Dimensi Obyek

Struktur data obyek yang dibuat dalam ruang 3 dimensi telah dibuat dengan menggunakan *software* CAD (*Computer Aided Design*) dimana ragam dan jenis perumpamaan benda dalam ruang nyata (*reality)* bisa dibuat dalam *software* tersebut. Dalam system yang telah dibakukan didunia internasional model *representatif* untuk ruang 3 dimensi banyak digunakan dengan model faset, dimana obyek dibuat berdasarkan kumpulan titik yang membentuk sekumpulan seperti awan (*points clouds*) dan dihubungkan susunannya kedalam dengan segitiga (*triangulasi*)[5]. Representasi pada obyek yang akan dijadikan obyek penelitian dalam makalah ini penulis dapatkan dari *software* CAD, disini penulis menggunakan salah satu *software* sebagai *software* CAD-nya dan merupakan sarana untuk mendapat obyek 3D dan *representasi* output berupa STL file dari

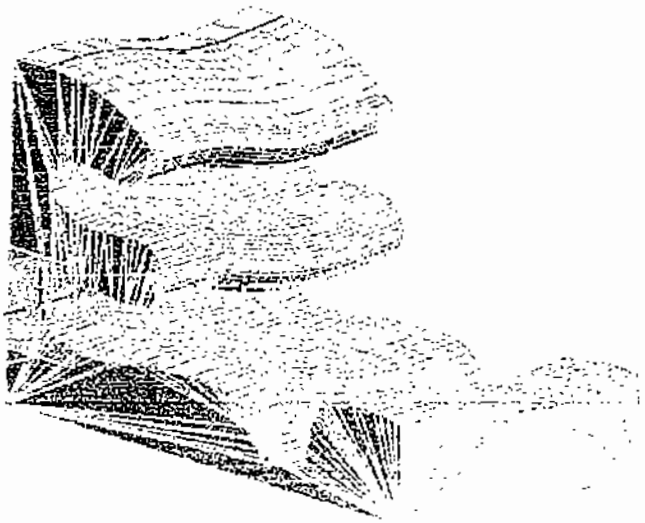
obyek tersebut. *Representasi* Obyek tersebut diilustrasikan dalam gambar dibawah (gambar II.1 dan gambar II.2) ini :

(a)



(b)

Gambar II.1. Obyek Gambar yang dibuat dalam Software Uni-Graphich (a)Complex model (b)Simple Model

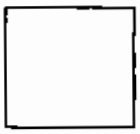
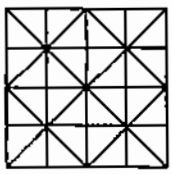


Gambar II.2. Representasi obyek Model dalam STL file

II.2. Metode Untuk Mengindentifikasikan *Feature*, Model Faset Dan File STL

Sistem CAM merupakan kumpulan perintah dari orientasi pahat yang telah dihasilkan dari pembacaan file STL yang dihasilkan oleh *software* CAD yang direpresentasikan dalam bentuk *parametric surface* dan *solid model* [5].

Model-Faset 3D merupakan model berbasis triangular mesh. Ada dua cara untuk memperoleh model faset, yaitu : (1) dibuat langsung dari *points-clouds* yang diperoleh dari rekayasa balik (*reverse* engineering) dan biasanya diperoleh dari pembacaan pada obyek nyata menggunakan alat pembaca (*scanner 3D*) yang direpresentasikan pada *bounding BOX*/ Kotak/ruang Pembatas); dan (2) *triangulasi* (pem-*faset*-an) dari model *solid/parametric* [5]. *Triangulasi meshing* merupakan diskritisasi sebuah model 3D menjadi segitiga-segitiga yang menyusun model tersebut. Segitiga tersebut memiliki informasi berupa verteks dan *normal vector* [4]. Adapun visualisasinya dapat dilihat dalam gambar 4

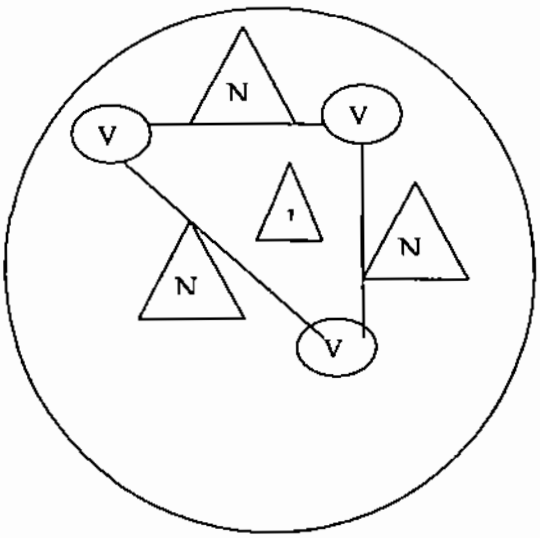
a)  (b) 

Gambar II.3 Model Triangular Mesh 2D[2]

Banyak data yang digunakan untuk merepresentasikan model-faset 3D. Namun, struktur data yang dikenal paling efisien dengan kebutuhan memori yang tidak besar serta mudah dibaca adalah Struktur Data Lawson. Kelemahan struktur data ini adalah tidak dapat digunakan untuk lintasan pahat, sehingga struktur data yang digunakan untuk struktur data ini adalah struktur data Lawson yang telah dimodifikasi [2].

Data struktur yang dibuat terdiri dari dua data utama, *list index vertex* dan *list index segitiga*. Proses pembuatan list ini didasarkan pada urutan segitiga pada file

STL-nya dari model faset 3D. Adapun gambaran dari model struktur data adalah sebagai gambar berikut.

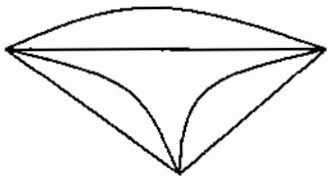


Gambar II.4 Model Struktur Data 1

Dari model facet tersebut di harapkan keakuarasian dapat di jaga [2]. Perhitungan keakurasian di dapat berdasar:

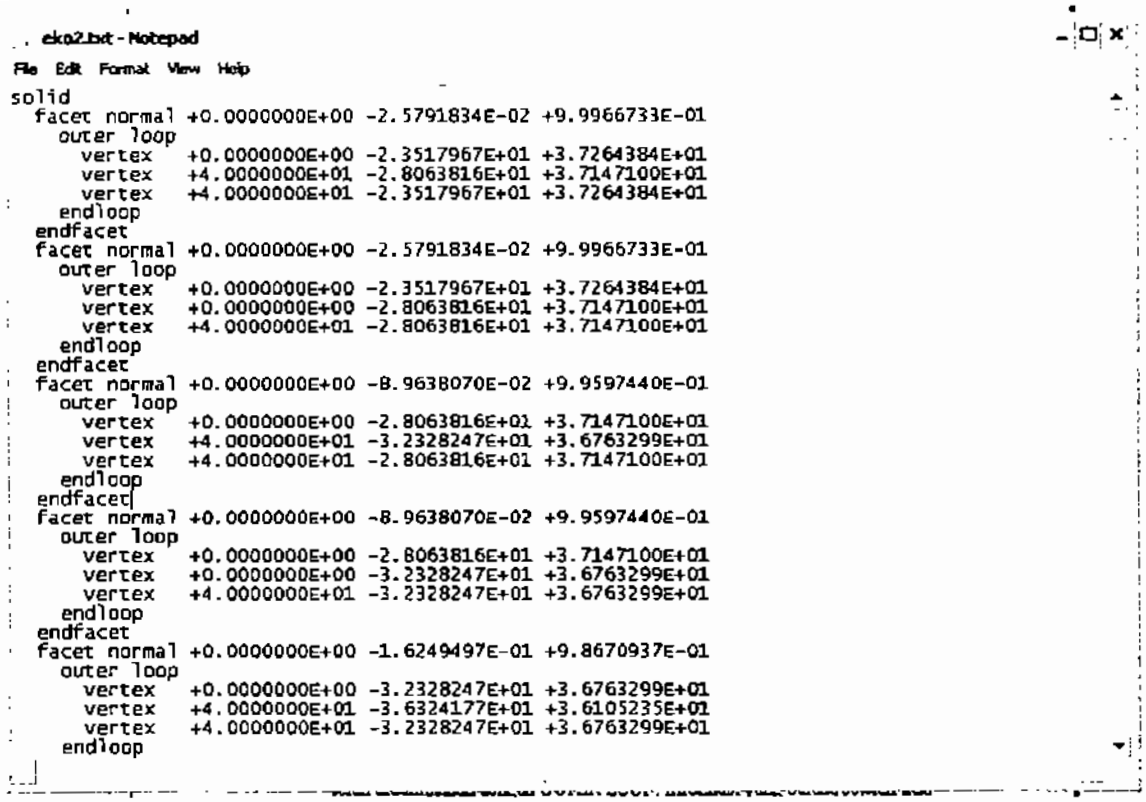
1. mencari jarak terjauh dari permukaan model *solid parametric* terhadap segitiga
2. jarak permukaan *solid parametric* ke sisi segitiga.

hal ini dapat di gambarkan sebagai:



Gambar II.5. Referensi Toleransi Triangulasi

Data yang diperoleh dari solid model dalam sistem CAD hanya berupa STL file yang kemudian akan diolah untuk menjadi perintah pada Sistem CAM. Format STL ini adalah yang paling umum digunakan karena lebih mudah dibaca dan dimengerti, serta dapat dibuka di *text* editor (Untuk format STL yang menggunakan mode ASCII). Gambar berikut adalah contoh isi dari file berekstensi stl (ASCII).



eko2.txt - Notepad

File Edit Format View Help

```
solid
  facet normal +0.0000000E+00 -2.5791834E-02 +9.9966733E-01
    outer loop
      vertex   +0.0000000E+00 -2.3517967E+01 +3.7264384E+01
      vertex   +4.0000000E+01 -2.8063816E+01 +3.7147100E+01
      vertex   +4.0000000E+01 -2.3517967E+01 +3.7264384E+01
    endloop
  endfacet
  facet normal +0.0000000E+00 -2.5791834E-02 +9.9966733E-01
    outer loop
      vertex   +0.0000000E+00 -2.3517967E+01 +3.7264384E+01
      vertex   +0.0000000E+00 -2.8063816E+01 +3.7147100E+01
      vertex   +4.0000000E+01 -2.8063816E+01 +3.7147100E+01
    endloop
  endfacet
  facet normal +0.0000000E+00 -8.9638070E-02 +9.9597440E-01
    outer loop
      vertex   +0.0000000E+00 -2.8063816E+01 +3.7147100E+01
      vertex   +4.0000000E+01 -3.2328247E+01 +3.6763299E+01
      vertex   +4.0000000E+01 -2.8063816E+01 +3.7147100E+01
    endloop
  endfacet
  facet normal +0.0000000E+00 -8.9638070E-02 +9.9597440E-01
    outer loop
      vertex   +0.0000000E+00 -2.8063816E+01 +3.7147100E+01
      vertex   +0.0000000E+00 -3.2328247E+01 +3.6763299E+01
      vertex   +4.0000000E+01 -3.2328247E+01 +3.6763299E+01
    endloop
  endfacet
  facet normal +0.0000000E+00 -1.6249497E-01 +9.8670937E-01
    outer loop
      vertex   +0.0000000E+00 -3.2328247E+01 +3.6763299E+01
      vertex   +4.0000000E+01 -3.6324177E+01 +3.6105235E+01
      vertex   +4.0000000E+01 -3.2328247E+01 +3.6763299E+01
    endloop
```

Gambar II.6. Contoh isi file STL (ASCII, bukan binary)

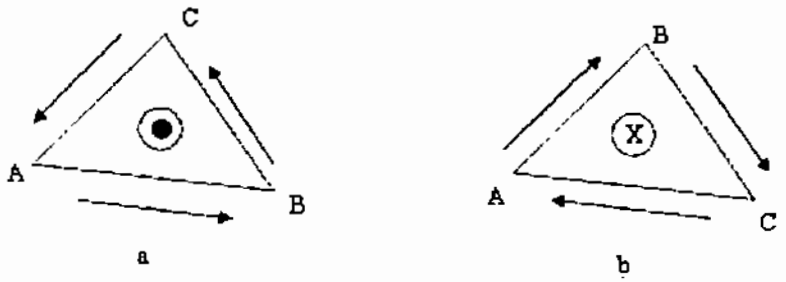
Dilihat diatas bahwa susunan dari sebuah STL file terdiri atas 3 koordinat titik dan bergabung menjadi garis (*vertex*) dan memiliki facet normal terhadap permukaan bidangnya. Berikut adalah penjelasan mengenai isi dari file stl di atas,

1. Kata *SOLID* menandakan dimulainya penggambaran atau penyimpanan model faset hingga ditutup dengan kata *ENDSOLID*.
2. Kata *FACET NORMAL* menandakan bahwa akan dibangun sebuah permukaan yang berbentuk segitiga dengan nilai *normal vector* berada pada kata setelah kata *FACET NORMAL*, hingga bertemu dengan kata *ENDFACET* yang berarti sebuah permukaan segitiga telah terbentuk, beserta informasi urutan dan letak verteks, serta *normal vector* dari segitiga tersebut.
3. Kata *OUTER LOOP* menandakan dimulainya loop dari koordinat verteks-verteks yang membangun segitiga hingga bertemu dengan kata *ENDLOOP*.

kata *VERTEX* merupakan verteks penyusun sebuah segitiga yang sebelumnya telah didefinisikan dengan *OUTER LOOP*. Informasi yang berada setelah kata *VERTEX* adalah posisi verteks pada sistem koordinat 3D.

II.3. Informasi Vektor Normal

Arah *normal vector* segitiga akan sesuai dengan urutan pengambilan verteks dalam mencari *normal vector* mengikuti kaidah tangan kanan. Jika putaran searah dengan jarum jam (*clockwise*), maka *normal vector* akan menuju bidang. Sebaliknya, jika putaran berlawanan arah jarum jam (*counter clockwise*), maka *normal vector* keluar dari bidang, untuk gambaran yang lebih jelas dapat dilihat pada Gambar II.7.



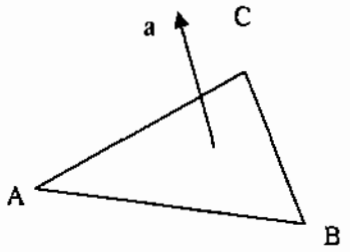
Gambar II.7. Arah Normal vector Segitiga [5]

Arah *normal vector* bergantung pada urutan verteks membentuk sebuah segitiga. Pada Gambar II.7 kiri, urutan pembentukan verteks berlawanan arah jarum jam (*counter clockwise*), maka perkalian silang (*cross product*) antara vektor-vektor yang menghubungkan verteks tersebut akan menghasilkan sebuah *normal vector* yang arahnya keluar dari bidang, sedangkan pada Gambar II.7 kanan, urutan pembentukan verteks searah jarum jam (*clockwise*), maka perkalian silang (*cross product*) antara vektor-vektor yang menghubungkan verteks akan menghasilkan sebuah *normal vector* yang arahnya masuk ke bidang.

II.4. Arah *Normal Vector*

Pembentukan bidang datar berfungsi untuk menghubungkan titik-titik penyusun bidang dan membentuk sebuah permukaan. Bentuk segitiga adalah bentuk yang paling sesuai untuk fungsi tersebut sebab pada ruang tiga dimensi, bentuk yang memiliki 3 buah *vertex* ini selalu konsisten pada penempatan titik, arah bidang, dan *normal vector* bidang. Bidang datar lain selain segitiga, memiliki kemungkinan terjadi ketidak-konsistenan antara arah bidang dengan penempatan

titik, sebab sangat mungkin terjadi bidang tersebut memiliki lebih dari satu *normal vector* untuk sebuah bidang datar akibat dari letak titik-titik penyusunnya yang tidak konsisten. Sebagai contoh, misalnya pada sebuah ruang 3 dimensi terdapat 4 buah titik yang akan disusun menjadi sebuah bidang datar segiempat beraturan.



Gambar II.8 Arah Normal vector

1.a (Gambar II.8) adalah arah *normal vector* pada sebuah segitiga dalam ruang 3 dimensi selalu konsisten. *Normal vector* pada sebuah bidang yang bukan segitiga (memiliki lebih dari empat *vertex*), dapat berjumlah lebih dari satu bergantung pada titik-titik yang dipilih sebagai acuan dalam membuat *normal vector* .

II.5. Persamaan garis linier

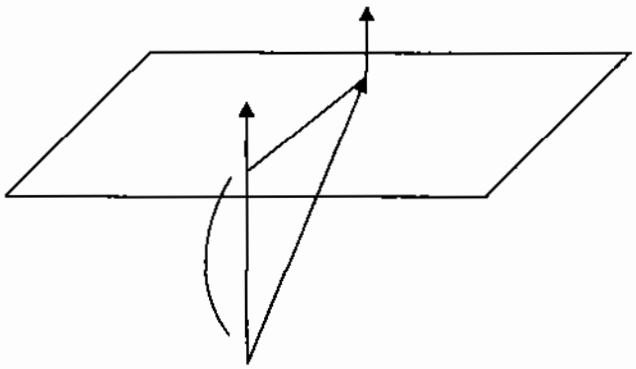
$$Ax + by = c \qquad \text{(II.1)}$$

Persamaan garis perhitungan vektor [5]:

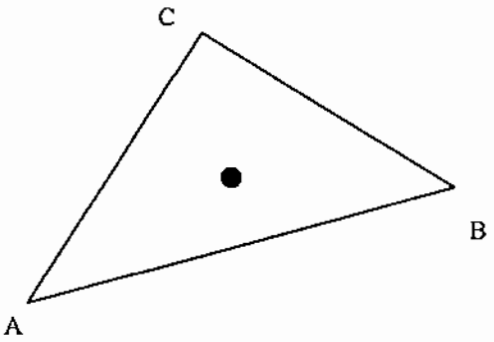
$$fx + gy + h = 0 \qquad \text{(II.2)}$$

dimana:

1. F, g , h adalah konstanta perhitungan
2. x dan y merupakan titik koordinat yang akan dilakukan perhitungan.



Gambar II.9. Sumbu normal vector



Gambar II.10. pemeriksaan titik apakah ada didalam atau diluar garis

Persamaan diatas tersebut merupakan acuan untuk perhitungan mencari perpotongan garis terhadap bidang segitiga.

Titik perpotongan berada di luar atau didalam dapat direpresentasikan dengan nilai :

$$fx + gy + h > 0 \quad \text{(II.3)}$$

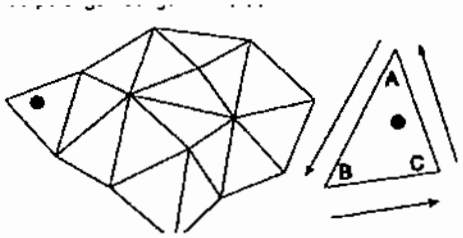
dimana:

$$f = -(y_2 - y_1), g = (x_2 - x_1) \text{ dan } h = (x_1 y_2 - x_2 y_1) \quad \text{(II.4)}$$

Pengecekan dilakukan dalam sistem koordinat x, y dan dilakukan pada garis AB, BC, dan CA secara berurutan. Persamaan garis di atas menghubungkan dua titik sehingga dan hasilnya dapat menentukan letak dari titik tersebut.

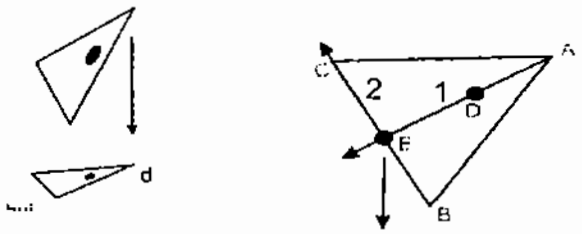
II.6. Penentuan Tinggi Titik dalam Bidang

Pertama kali yang dilakukan dalam mencari segitiga pendeteksian terhadap arah jalan *tool* adalah sebagai berikut:



Gambar II.11 titik yang memotong segitiga

Segitiga di proyeksikan ke bidang x, y dan meniadakan sumbu z



Gambar II.12.a proyeksi segitiga pada bidang x.y

Dengan menggunakan persamaan garis sebagai berikut:

$$t = ax + by + c \qquad \text{(II.5)}$$

Pemeriksaan dilakukan pada setiap *edge* dengan aturan segitiga *clock wise*.setelah titik x,y yang di kehendaki telah di dapat maka kita harus menentukan tinggi fature benda kerja dengan rumusan:

$$\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} + n\begin{pmatrix} X_D - X_1 \\ Y_D - Y_1 \end{pmatrix} = \begin{pmatrix} X_B \\ Y_B \end{pmatrix} + m\begin{pmatrix} X_C - X_B \\ Y_C - Y_B \end{pmatrix} \qquad \text{(II.6)}$$

Nilai m dan n dapat di tentukan dengan cara eliminasi:

E = A + n(D-A) atau

E= B + m(C-B) (II.7)

Vector BC = n x *Vector* BE

Nilai n didapat, selanjutnya z dari titik E dapat dicari dengan :

$$\begin{pmatrix} Xe \\ Ye \\ Ze \end{pmatrix} = \begin{pmatrix} Xb \\ Yb \\ Zb \end{pmatrix} + n \begin{pmatrix} Xc - Xb \\ Yc - Yb \\ Zc - Zb \end{pmatrix} \qquad \text{(II.8)}$$

Apabila Titik jatuh pada bidang segitiga maka perhitungan tersebut dilanjutkan dengan memanfaatkan nilai x dan y pada titik yang didapat diatas (terakhir dilakukan pembacaan)

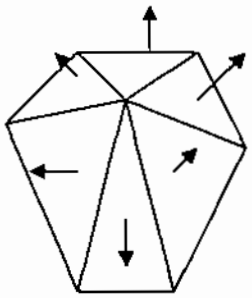
Dari persamaan Zd = Za + n(Ze-Za), sehingga didapat persamaan :

$$\begin{pmatrix} Xd \\ Yd \\ Zd \end{pmatrix} = \begin{pmatrix} Xa \\ Ya \\ Za \end{pmatrix} + n \begin{pmatrix} Xe - Xa \\ Ye - Ya \\ Ze - Za \end{pmatrix} \qquad \text{(II.9)}$$

Sehingga nilai z pada titik D dapat diketahui.

II.7. Pengembangan Metode Penghitungan *Normal vector* pada CC-point

Pada model faset 3D dan metode pembuatan *cc-point* untuk lintasan pahat yang dikembangkan, *cc-point* dapat terletak pada *vertex* dari segitiga dan pada sisi segitiga. Dua metode penghitungan *normal vector* dari cc-point yang terletak pada *vertex* segitiga dari pada model faset yang coba dikembangkan yaitu yang disebut Resultan Normal, Resultan Normal Berbobot dan interpolasi normal vertek[2]. Ketiga metode ini adalah sebagai berikut :



Gambar II.13a Normal vector [2]

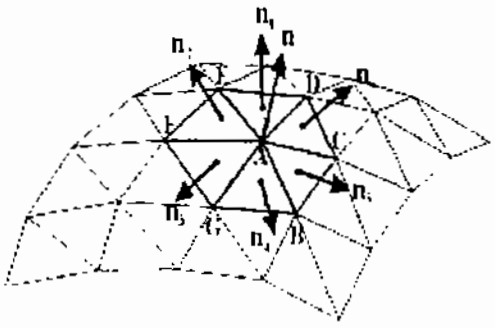
a) **Resultan Normal** [5][11]: Pada metode ini, *normal vector* n dari cc-point yang terletak pada sebuah *vertex*, yang dikelilingi oleh rangkaian segitiga yang mengelilingi *vertex* tersebut, diperoleh dengan menghitung resultan masing-masing *normal vector* segitiga yang mengelilingi *vertex* tersebut seperti terlihat pada Gambar 6. Untuk hal ini *normal vector vertex* (cc-point) dihitung dengan persamaan sederhana berikut :

$$n = \frac{\sum_i n_i}{\left[\sum_i n_i\right]} \qquad \text{(II.10)}$$

Dimana : n_i adalah *normal vector* ke-i dari seluruh segitiga tetangga yang mengelilingi *vertex*.

b) Resultan normal berbobot

Pada metode ini, *normal vector* pada sebuah cc-point (pada sebuah *vertex*) dihitung dengan mempertimbangkan seluruh luasan segitiga tetangga yang mengelilingi *vertex* tersebut.



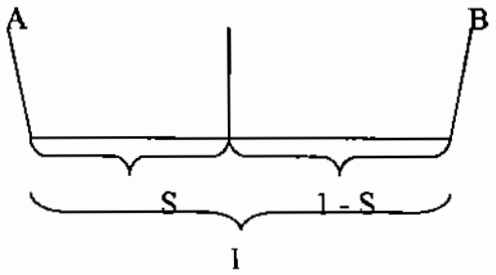
Gambar II.13.b diatas merupakan normal vector dari cc point yang jatuh pada vertek yang dikelilingi serangkaian segitiga.[5]

$$n = \frac{\sum \frac{a_i}{A} n_i}{\left[\sum \frac{a_i}{A} n_i\right]} \qquad \text{(II.11)}$$

Dimana n_i adalah *normal vector* dari segitiga tetangga ke-i (a_i) yaitu factor robot dengan a_i sebagai luasan segitiga ke-i dan A adalah total penjumlahan luasan dari seluruh segitiga tetangga. Ini bisa diperhitungkan bila kondisi pada segitiga tetangga punya nilai *normal vector* yang berlainan atau bila antara *vertex* satu dengan lanilnya tersebut hampir sama atau memenuhi batas toleransi tertentu, dapat dikatakan sekumpulan segitiga tersebut membentuk bidang datar.[5]

c) Interpolasi normal Vertek [5]

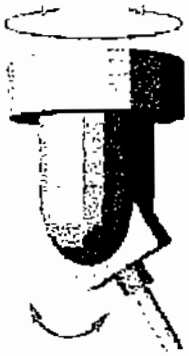
Perhitungan dilakukan bila *normal vector* jatuh pada sisi segitiga (*edge*) dimana diperhtungkan interpolasi antara 2 titik pada *edge* tersebut dengan melihat jaraknya dan melakukan interpolasi linier terhadap 2 vertek penyusun sisi segitiga tersebut.'



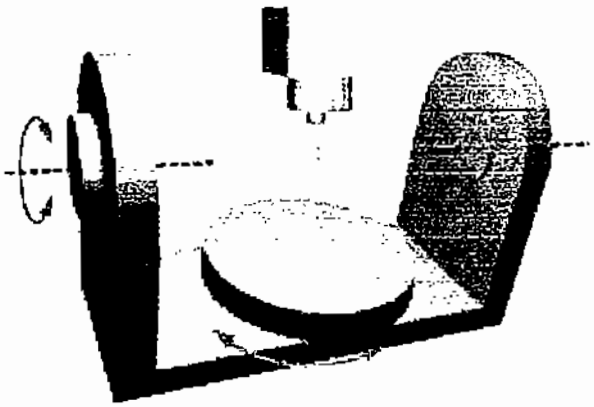
Gambar II.13.c interpolasi Normal Vertek

II.8. Jenis Dan Type Pemesinan 5 Axis

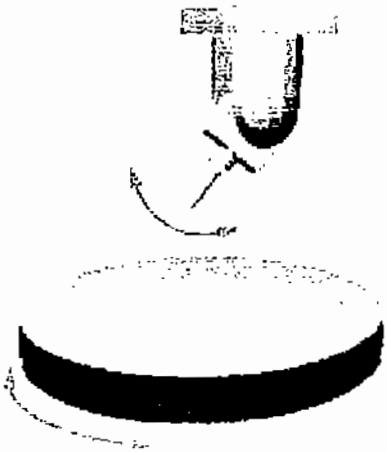
Pemesinan 5 axis yang ada saat ini ada beberapa jenis, dilihat dari cara kerjanya dan faktor gaya pemakanan yang terjadi diantaranya [9]:



Gambar. II.14a. Orientasi kepala holder[9]

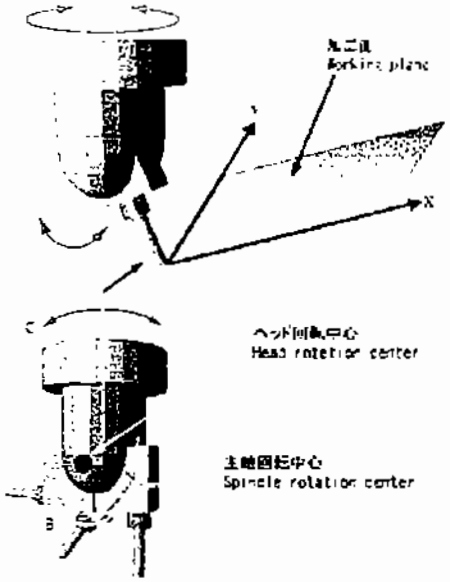


Gambar II.14b. orientasi meja berputar dengan sumbu holder tetap[9]

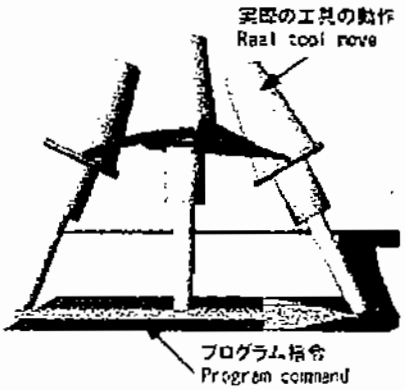


Gambar II.14c. orientasi meja dan holder (orientasi keduanya) [9]

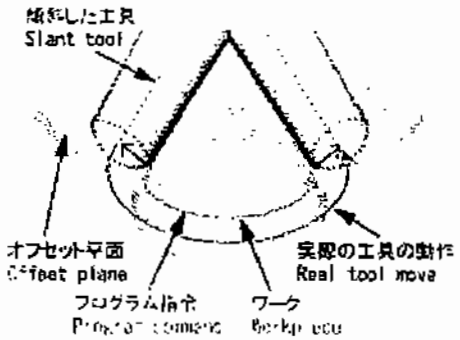
Dari carakerjanya :



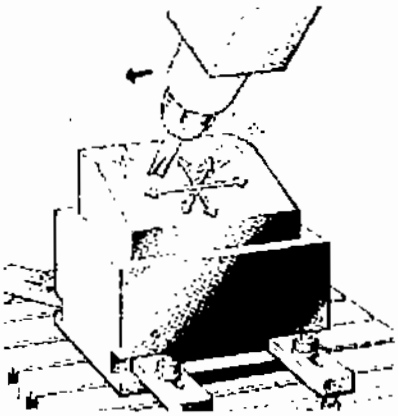
Gambar II.15a. tegak lurus terhadap bidang datar[9]



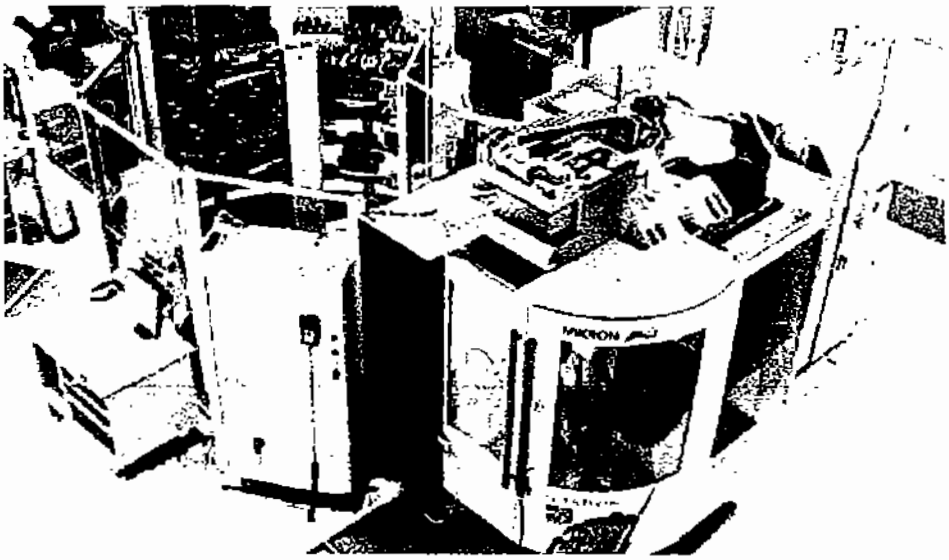
Gambar II.15b. tegak lurus dan berousat pada pahat[9]



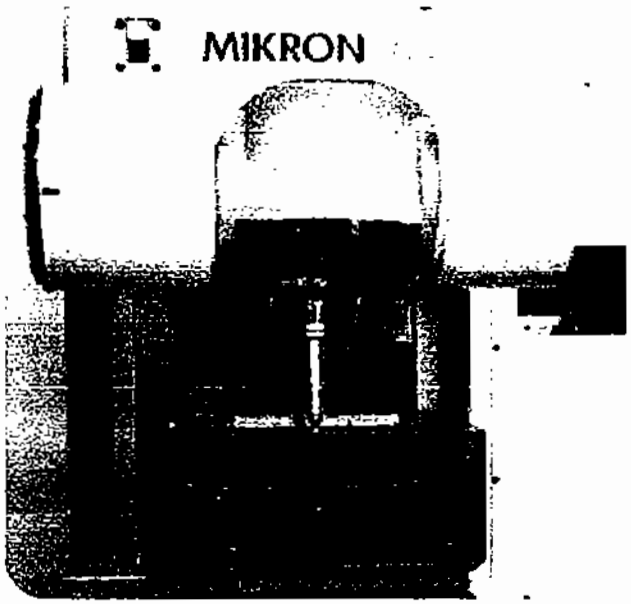
gambar II.15c. Tiga dimensi pemakan pada punggung pahat. [9]



Gambar II.15d. Tiga dimensi berpusat pada handle pemakanan[9]



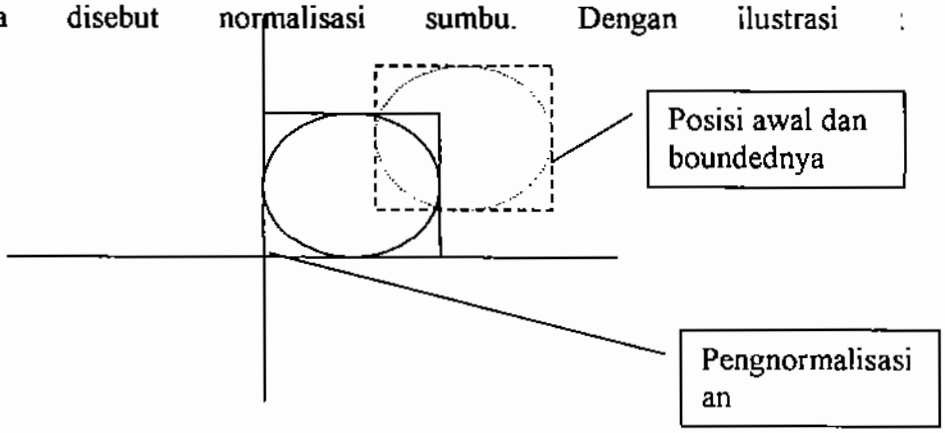
Gambar II.16a.Contoh mesin 5-axis[9]



Gambar II.16.b Contoh mesin 5 axis. [9]

II.9. Normalisasi Sumbu

Melakukan deteksi ini dengan menormalisasi bidang dari koordinat obyek , maksudnya memposisi kedudukan awal dari pembacaan pada bagian sisi atau sudut dari obyek kerja dengan memposisikannya pada bidang koordinat pusat X=0,Y=0 dan Z=0, tetapi ini tidaklah mutlak, asalkan posisi dari benda verja tersebut bisa diperhiungkan posisi awalnya (X, dan Y saja yang punya nilai 0), perhitungan sudah dapat dilakukan. Langkah kerja untuk pergeseran obyek pada koordinat ruang yang mudah dibaca ini biasa disebut normalisasi sumbu. Dengan ilustrasi :



Gambar II.17. Normalisasi koordinat