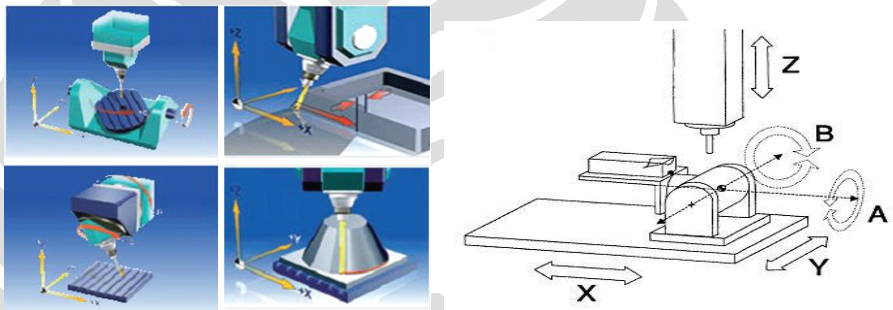
BAB II
TEORI DASAR

2.1. Proses Pemesinan Multi-Axis

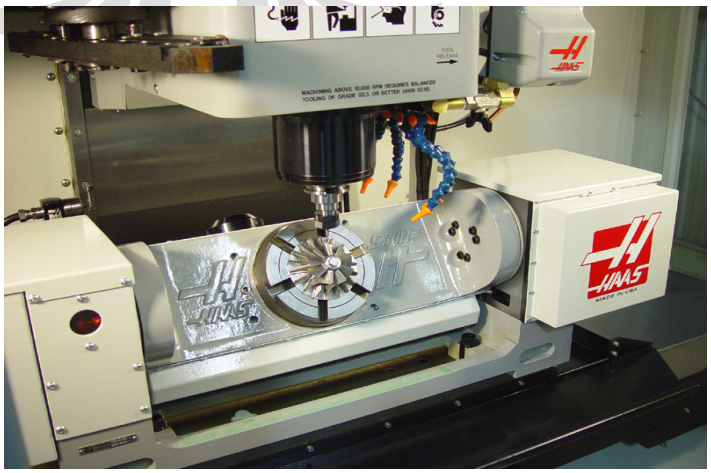
Proses pemesinan *multi-axis* didefinisikan sebagai proses pemesinan yang dilakukan dengan mesin *frais*/*milling* (CNC) dengan pergerakan lima-axis (5-axis), atau biasa disebut pemesinan 5-axis. Pada pemesinan tersebut, pahat selain dapat melakukan tiga gerakan linear (3-axis) terhadap sumbu X, sumbu Y dan Z, dapat juga melakukan tambahan dua jenis pergerakan rotasi yang bergantung dari konfigurasi mesin 5-axis [5]. Bentuk gerakan linier maupun rotasi seperti terlihat pada gambar 2.1. dibawah ini.



(a) (b)

Gambar 2.1. (a) Bentuk konfigurasi proses pemesinan 5-aksis [10]
(b) konfigurasi gerakan linier dan rotasi pemesinan 5-aksis. [7]

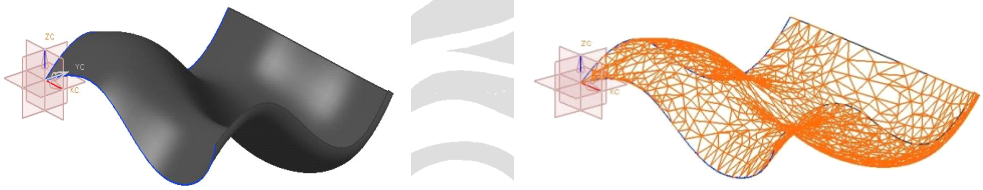
Contoh dari mesin 5-aksis terlihat pada gambar dibawah



Gambar 2.2. Mesin 5-Aksis [11]

2.2. Model Faset

Model-Faset 3D merupakan model berbasis triangular mesh. Ada dua cara untuk memperoleh model faset, yaitu : (1) dibuat langsung dari *points*-cloud yang diperoleh dari rekayasa balik (*reverse* engineering); dan (2) triangulasi (pemfasetan) dari model solid/parametric. Triangulasi meshing merupakan salah satu proses pemodelan berbasis elemen hingga (*Finite Element Model)* dimana sebuah model akan digenerasi menjadi elemen kecil-kecil yang berbentuk segitiga. Contoh dari model faset yang dibuat dari model permukaan parametrik seperti terlihat pada gambar dibawah.



(a) (b)

Gambar 2.3. (a) Model surface parametric (b) Model faset

2.2.1. Struktur-Data Model-Faset 3D [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 memiliki dua data utama, *list index vertex* dan *list index segitiga*. Proses pembuatan list ini didasarkan pada urutan segitiga pada file *Stereolithography* (STL) dari model faset 3D. Adapun bentuk data dari STL File dapat dilihat sebagai berikut

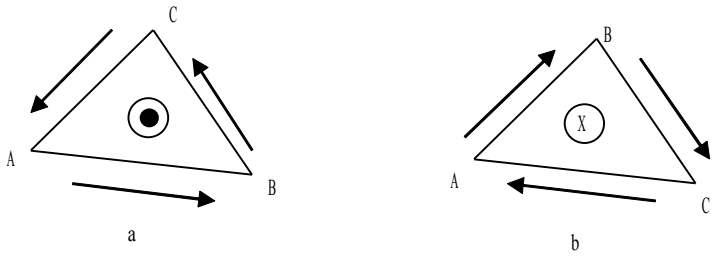
```
solid
  facet normal +7.6212244E-02 +0.0000000E+00 -9.9709161E-01
    outer loop
      vertex   -1.8025210E+01 -2.5000000E+01 +1.5441596E+01
```

```
    vertex   -2.5735497E+01 -2.5000000E+01 +1.4852264E+01
    vertex   -2.5735497E+01 +0.0000000E+00 +1.4852264E+01
   endloop
  endfacet
  facet normal +1.4307332E-01 +0.0000000E+00 -9.8971209E-01
   outer loop
    vertex   -2.5735497E+01 -2.5000000E+01 +1.4852264E+01
    vertex   -3.0509943E+01 -2.5000000E+01 +1.4162067E+01
    vertex   -3.0509943E+01 +0.0000000E+00 +1.4162067E+01
   endloop
  endfacet
 endsolid
```

Fungsi dibentuknya sebuah segitiga dari tiga titik adalah : (1) membentuk bidang segitga (face), dan (2) menentukan vektor normal pada bidang tersebut.

2.2.2. Penghitungan Vektor Normal [6]

Vektor normal sangat diperlukan dalam proses pemesinan 5-axis karena dalam proses pemotongan material, pergerakan pahat dalam proses tersebut meliputi posisi tegak berdasar sumbu xyz dan posisi pahat membentuk sudut terhadap bidang xyz, sesuai dengan permukaan bidang. Vektor normal merupakan hasil perkalaian *cross product* antara 2 vektor pembentuk sisi segitiga. Arah vektor normal tergantung pada arah putaran vektor dari ketiga vertex yang membentuknya dengan mengikuti kaidah tangan kanan. Jika putaran searah dengan jarum jam (*clockwise*), maka vektor normal akan menuju bidang. Sebaliknya, jika putaran berlawanan arah jarum jam (*counter clockwise*), maka vektor normal keluar dari bidang. Adapun visualisasi pembentukan vektor normal dapat dilihat pada gambar dibawah. Pada Gambar a, urutan pembentukan vertex berlawanan arah jarum jam (*counter clockwise*), maka cross product antara vektor-vektor yang menghubungkan vertex tersebut akan menghasilkan sebuah vektor normal yang arahnya keluar dari bidang, sedangkan pada Gambar b, urutan pembentukan vertex searah jarum jam (*clockwise*), maka cross product antara vektor-vektor yang menghubungkan vertex akan menghasilkan sebuah vektor normal yang arahnya masuk ke bidang



Gambar 2.4. Vektor Normal Segitiga [6]

Pada proses pembuatan lintasan pahat, maka cc-point akan muncul pada :

a. Pada vertex segitiga.
b. Pada sisi segitiga.

a. Jika cc-point terletak pada vertex segitiga, maka ada dua metode penghitungan vektor normal, yaitu *Resultan Normal* dan *Resultan Normal Berbobot*.

   i) Resultan Normal. Vektor normal sebuah cc-point sama dengan vektor normal vertex yang letaknya sama dengan cc-point tersebut. Dalam hal ini perhitungan dilakukan dengan menghitung resultan vektor normal beberapa segitiga yang mengelilingi vertex tersebut, atau dengan kata lain resultan vektor normal dari segitiga-segitiga yang salah satu vertexnya adalah vertex yang bersangkutan. Persamaan yang digunakan untuk metode ini adalah :

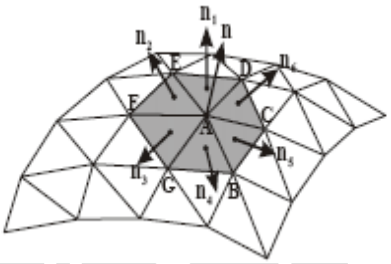
$$n = \frac{\sum_i n_i}{\left|\sum_i n_i\right|}$$

   Di mana n adalah vektor normal vertex (*cc-point*), dan n_i adalah vektor normal segitiga ke-i yang mengelilingi vertex tersebut.

   ii) Resultan Normal Berbobot. Pada metode ini, luas setiap segitiga juga diperhitungkan bersama dengan vektor normal segitiga yang mengelilingi vertex yang bersangkutan. Persamaan yang digunakan untuk metode ini adalah :

$$n = \frac{\sum_i \frac{a_i}{A} n_i}{\left| \sum_i \frac{a_i}{A} n_i \right|}$$

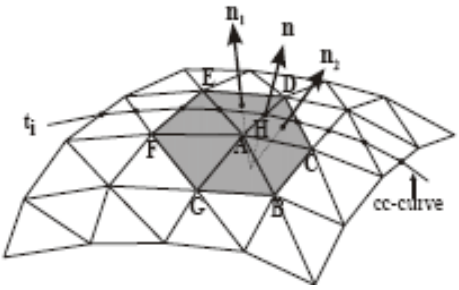
Di mana a_i adalah luas segitiga ke-i yang mengelilingi vertex yang bersangkutan, dan A adalah luasan total dari seluruh segitiga tetangga.



Gambar 2.5. Vektor normal cc-point [6]

b. Jika cc-point terletak pada sisi segitiga, maka ada dua metode penghitungan vektor normal, yaitu *Resultan Normal Bidang* dan *Interpolasi Normal Vertex*.

i) Resultan Normal Bidang. Menghitung vektor normal cc-point yang jatuh pada sebuah edge berdasarkan resultan vektor normal segitiga yang memiliki afiliasi langsung dengan edge tersebut.



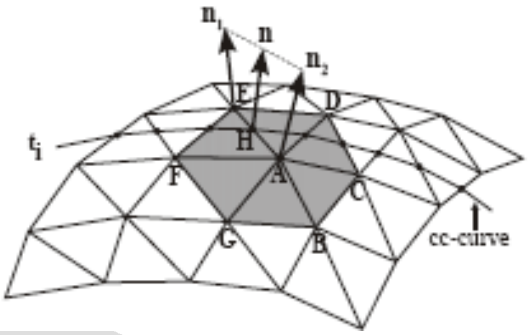
Gambar 2.6. Vektor normal cc-point
pada perhitungan resultan normal bidang [6]

Vektor normal ini dapat dihitung dengan persamaan

$$n = \frac{\left(n_j + n_k\right)}{\left|\left(n_j + n_k\right)\right|}$$

Dimana n_j dan n_k adalah vektor normal dari segitiga-segitiga yang bersisian dengan edge tempat *cc-point* berada (n_1 dan n_2 pada Gambar 18).

ii) Interpolasi Normal Vertex. Vektor normal cc-point dihitung berdasarkan interpolasi linear vektor normal dua vertex penyusun edge, tempat cc-point yang bersangkutan berada.



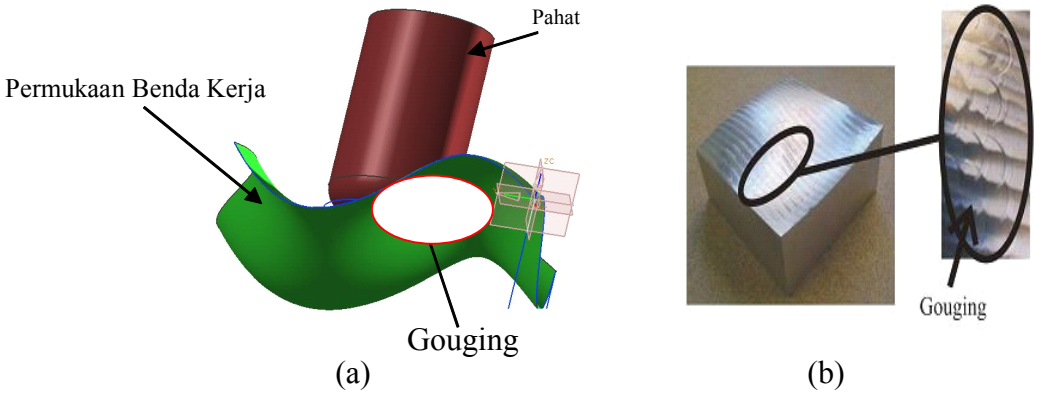
Gambar 2.7. Vektor normal cc-point
Pada Interpolasi Normal Vertex [6]

Berdasarkan Gambar , vektor normal n(s) di mana $s \in [0,1]$ (s adalah perbandingan antara panjang sisi EH dengan sisi EA) pada sisi EA dapat dihitung sebagai berikut,

$$n = \frac{(n_j + n_k)}{\left|(n_j + n_k)\right|}$$

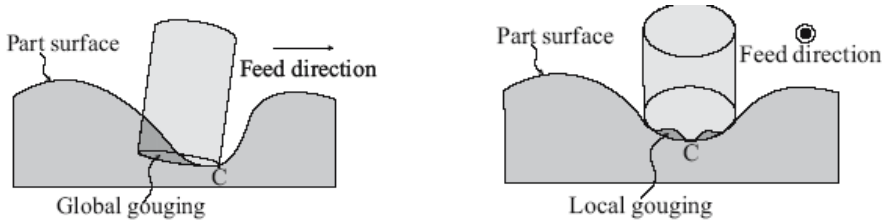
2.3. Interferensi (*Gouging*)

Interferensi (*Gouging*) adalah fenomena dimana pahat memotong material lebih banyak dari yang dispesifikasikan. Interferensi tersebut dapat menyebabkan cacat pada produk akhir dari suatu proses pemesinan. Fenomena keduanya ((a) *gouging* dan (b) cacat yang dihasilkan) tersebut dapat dilihat pada gambar 2.8 dibawah ini.



Gambar 2.8. (a) Interferensi pada model Parametrik (b) Cacat produk akibat gouging [6]

Berdasar lokasi terjadinya gouging, maka secara umum gouging diklasifikasikan menjadi dua yaitu : (1) Gouging yang terjadi pada *rear-bottom side* dari pahat; dan (2) Gouging yang letaknya dekat dengan CC-point, seperti terlihat pada gambar 2.9.

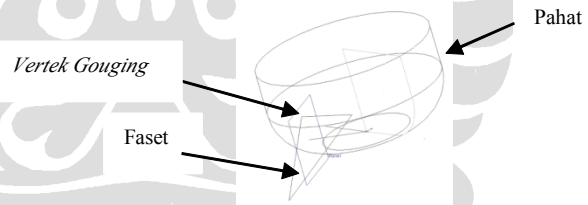


Gambar 2.9. Gouging Berdasar Lokasi [6]

2.3.1. Jenis Interferensi Pada Model Faset

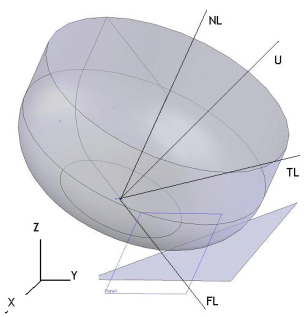
Dari hasil identifikasi, ada tiga jenis interferensi (*gouging*) yang mungkin terjadi pada pemesinan muka. Tiga jenis interferensi yang dimaksud adalah :

1. Muka pahat (*tool bottom*) interferensi terhadap vertex dari segitiga (gambar 2.10).



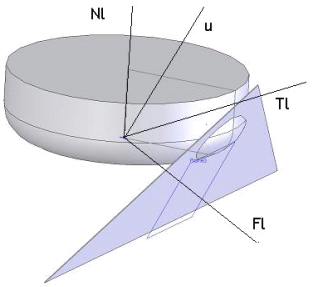
Gambar 2.10. Interferensi Pada Vertex

2. Muka pahat interferensi terhadap sisi segitiga



Gambar 2.11. Interferensi Pada Edge

3. Muka pahat interferensi terhadap bidang segitiga



Gambar 2.12. Interferensi Pada Face

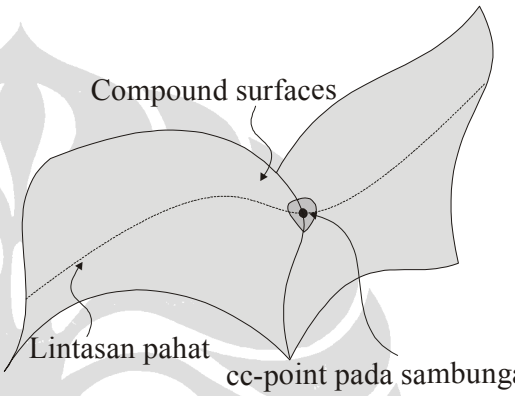
2.3.2. Metode Eliminasi Gouging [5]

Sampai saat ini, telah banyak metode yang diusulkan untuk menanggulangi permasalahan gouging, antara lain yang terkenal adalah : **a)** metode *curvature matching* [Marciniak 1987] [Jensen and Anderson 1993] [Mullins et. al. 1993] [Kruth and Klewais 1994] [Bedi et. al. 1997] [Lee 1997, 1998] [Wang and Tang 1999] [Rao and Sarma 2000] [Sarma 2000] [Yoon et. al. 2003] [Tournier dan Duc 2004], [Jaganathan dan Lin 2005], [Quinsat dan Sabourin 2007], **b)** metode *Z-map* atau *G-buffer* [Saito and Takahashi 1991] [Choi dan Jerard 1998], dan **c)** metode *point spreading* (penyebaran point) [Gray et. al. 2003]. [Kiswanto, 2007].

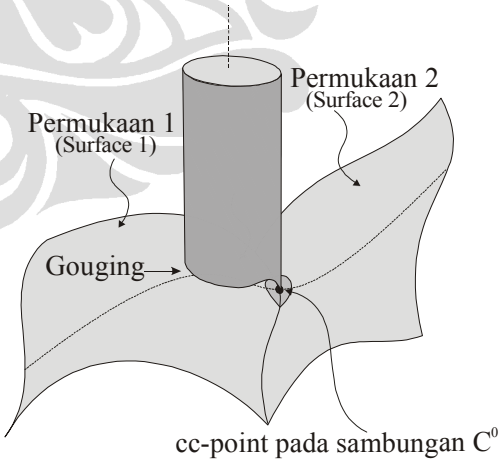
- **Metode *curvature matching***

  Pada metode *curvature matching*, kurvatur lokal (*local curvature*) atau kelengkungan lokal dari sebuah model permukaan 3D yang di lalui oleh pahat (sepanjang *tool path*) dianalisa dan dibandingkan dengan kurvatur (*curvature*) dari pahat yang digunakan. Metode ini awalnya dikembangkan oleh Marciniak [Marciniak 1987]. Kemudian diperbaiki dan dipopulerkan oleh Kruth [Kruth dan Klewais 1994]. Metode ini kemudian dengan cukup intensif banyak disampaikan oleh Y. S. Lee [Lee 1997, 1998, 1998b]. Pada prinsipnya metode ini, kurvatur lokal dari model permukaan 3D dari produk (dies) pada *cutter contact point* (*cc-point*) di proyeksikan dua arah yaitu pada bidang yang paralel dan tegak lurus arah pemotongan (arah gerak pahat) dibandingkan dengan kurvatur dari pahat. CC-point adalah titik kontak

antara model permukaan pahat dan model permukaan 3D produk (dies) disetiap titik pusat pahat (CL-point). Kondisi bebas *gouging* didapat bila kurvatur dari model pahat lebih besar (~ kelengkungan radius lebih kecil) dibanding kurvatur dari model 3D produk (~ kelengkungan radius lebih besar) di setiap cc-point sepanjang lintasan pahat. Namun, dengan metode ini *gouging* masih mungkin terjadi yaitu pada sambungan antar model permukaan dengan tingkat kontinuitas C^0 dan C^1 seperti terlihat pada Gambar 10.



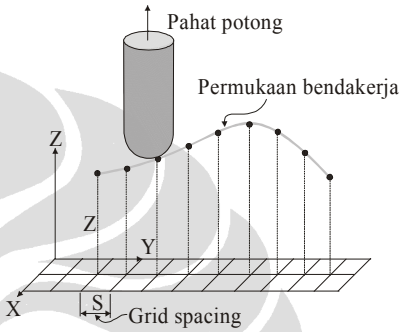
Gambar 2.13. CC-point terletak di sambungan dua permukaan majemuk (compound surface) yang memiliki kontinuitas C^0[5]



Gambar 2.14. Pahat tidak gouging dengan permukaan 2 disaat gouging permukaan 1[5]

- **Metode *Z-map***

Prinsip dasar metode ini adalah dengan membuat lintasan pahat berdasarkan serangkaian posisi pahat pada lokasi diskrit (*grid spacing*) dengan interval yang kecil (Gambar 12) [Li 1993] [Jerard dan Choi 1998] [Baek 2006]. Namun, dengan metode ini, untuk mencegah terjadinya *gouging* di satu posisi pahat diperlukan kepadatan *spasi grid* yang sangat tinggi (sampai dengan 500.000 untuk keakurasian 0.05 mm) sehingga menghasilkan perhitungan yang sangat intensif yang harus dilakukan disetiap posisi pahat.



Gambar 2.15. Metode Z-map dengan pemakaian grid spasi [5]

- **Metode *point spreading***

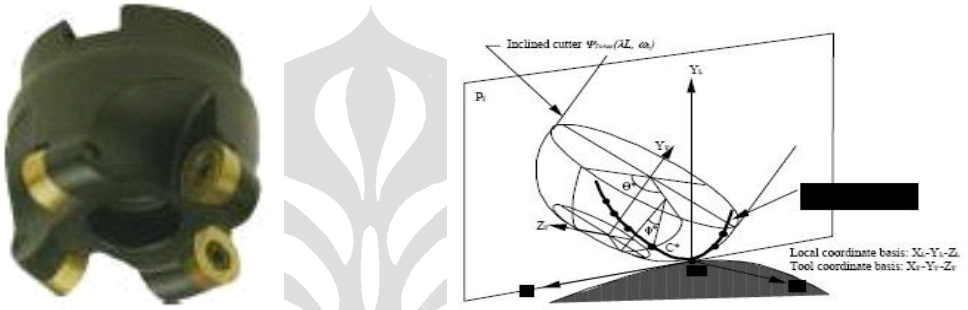
Point spreading atau penyebaran titik merupakan salah satu metode yang paling konvensional dan banyak digunakan sistem-CAM saat ini. Dengan metode ini sekumpulan titik di distribusikan (disebar) pada model permukaan disekitar cc-point [Gray et. al. 2003]. Dalam hal ini penyebaran titik proporsional dengan toleransi pemesinan yang dispesifikasikan. Untuk dapat mendeteksi kemungkinan terjadinya *gouging*, maka dibutuhkan *kepadatan titik* yang sangat tinggi dan *jarak antar titik* tersebut harus diatur sedemikian rupa sehingga keakurasian pemesinan dapat dijamin sesuai dengan spesifikasi desain [Austin et. al. 1996] [Lee 1997]. Pada prakteknya, karena kompleksitas model permukaan produk, kedua hal ini sering gagal di capai sehingga *gouging* dapat terjadi (Gambar 13). Selain dari pada itu, karena metode ini membutuhkan sebaran titik dalam jumlah yang sangat banyak, maka, seperti 2

metode sebelumnya, membutuhkan struktur data yang kompleks dan komputasi yang sangat intensif.

2.4. Persamaan-Persamaan Matematika Yang Relevan

2.4.1. Pahat Toroid (*Filleted-EndMill*)

Pahat toroid merupakan pahat end-mills yang memiliki radius kelengkungan r pada sisi ujung bagian bawah pahat, seperti terlihat pada gambar 2.16.



(a) **(b)**

Gambar 2.16. (a) Pahat Toroid [9] (b) Model Pahat Toroid [9]

Permukaan pahat Toroid secara matematis dapat dituliskan dengan bentuk sebagai berikut [9] :

$$\varphi(\theta, \emptyset, \alpha, \beta) = \begin{pmatrix} (\alpha R_1 + R_2 \sin\emptyset)\cos\theta \\ (\alpha R_1 + R_2 \sin\emptyset)\sin\theta \\ R_2(1-\cos\emptyset) + \beta H \end{pmatrix}$$

2.4.2. Sistem Koordinat

Perhitungan interfensi antara model pahat dengan model faset dilakukan dalam sistem koordinat lokal. Untuk membentuk sistem koordinat lokal ke global atau sebaliknya, dibutuhkan suatu matrix transformasi, yaitu matrix *lokal_to_global* dan matrix *global_to_lokal*. Isi dari matrix tersebut adalah vektor sumbu x, y, dan z dari sistem koordinat lokal. Berikut adalah bentuk matrix transformasi tersebut,

$$\mathbf{M} = \begin{bmatrix} n_x & n_y & n_z & 0 \\ o_x & o_y & o_z & 0 \\ a_x & a_y & a_z & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

dimana n adalah vektor sumbu-x lokal, o adalah vektor sumbu-y lokal, a adalah vektor sumbu-z lokal, dan t adalah lokasi titik pada sistem koordinat global yang akan menjadi titik pusat sistem koordinat lokal. Jika suatu titik berada pada posisi r (r_x,r_y,r_z) pada sistem koordinat global, dan titik yang sama berada pada posisi p (px,py,pz) pada sistem koordinat lokal, maka hal tersebut dapat ditulis dalam persamaan,

$$[r \quad 1] = [p \quad 1]M$$

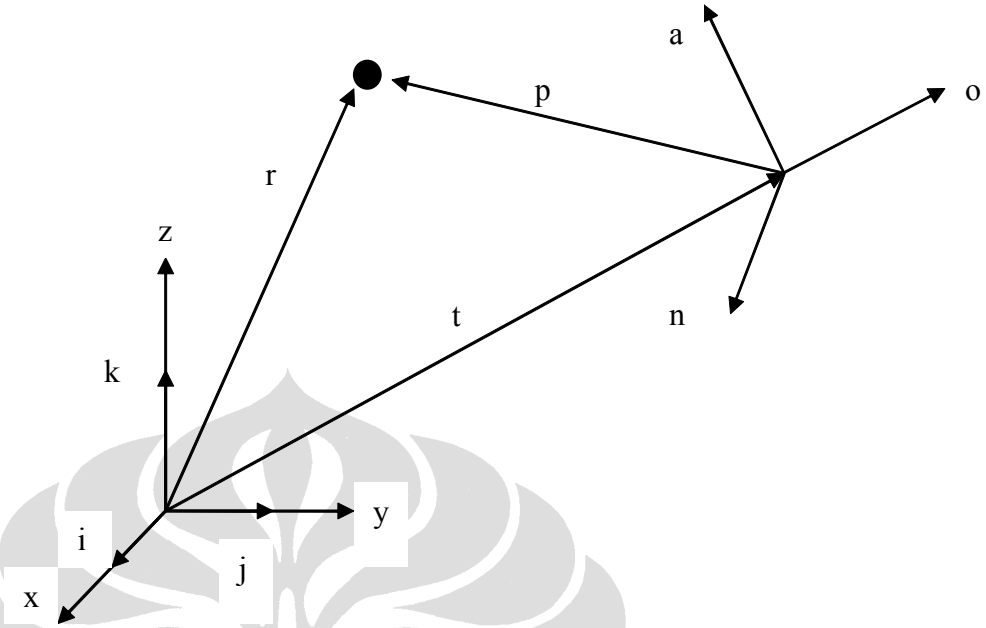
Persamaan diatas digunakan untuk mencari lokasi titik pada sistem koordinat global dari lokasi yang telah diketahui pada sistem koordinat lokal. Pada kasus sebaliknya, mencari lokasi titik pada sistem koordinat lokal yang belum diketahui dari sistem koordinat global dapat digunakan persamaan,

$$[p \quad 1] = [r \quad 1]M^{-1}$$

Dalam keadaan orthogonal, inverse matrix M, yaitu matrix M^{-1} bisa didapat dengan mudah [CHO98],

$$\mathbf{M} = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ -n \bullet t & -o \bullet t & -a \bullet t & 1 \end{bmatrix}$$

Persamaan-persamaan tersebut dapat digambarkan dalam bidang tiga dimensi



Gambar 2.17 Orintasi Titik ke SKL[1]

Sesuai dengan persamaan matrix transformasi dan gambar di atas, persamaan matrix yang di bentuk dalam sistem-CAM ini untuk permasalahan lokal ke global adalah :

$$\begin{bmatrix} r_x & r_y & r_z & 1 \end{bmatrix} = \begin{bmatrix} p_u & p_v & p_w & 1 \end{bmatrix} \begin{bmatrix} f_x & f_y & f_z & 0 \\ l_x & l_y & l_z & 0 \\ n_x & n_y & n_z & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

Kemudian untuk persamaan dari global ke lokal, adalah sebagai berikut,

$$\begin{bmatrix} r_x & r_y & r_z & 1 \end{bmatrix} \begin{bmatrix} f_x & l_x & n_x & 0 \\ f_y & l_y & n_y & 0 \\ f_z & l_z & n_z & 0 \\ -f \bullet t & -o \bullet t & -a \bullet t & 1 \end{bmatrix} = \begin{bmatrix} p_u & p_v & p_w & 1 \end{bmatrix}$$

Langkah-langkah yang dilakukan untuk mendapatkan vektor-vektor NFL adalah sebagai berikut,

i. Cari vektor N yang merupakan vektor normal bidang pada sistem koordinat lokal. Vektor ini akan menjadi sumbu z pada sistem koordinat lokal.
ii. Vektor F adalah vektor yang akan menjadi sumbu x pada sistem koordinat lokal. Vektor ini sejajar dengan lintasan pahat pada bidang 2D, namun arahnya berlawanan.
iii. Dari informasi yang didapat pada langkah kedua, nilai vektor F dapat dihitung berdasarkan syarat-syarat berikut,
    a. Sejajar sumbu y negatif dalam bidang 2D
    b. Panjang vektor adalah 1
    c. Membentuk sudut 90^o terhadap vektor N
iv. Tentukan vektor L yang merupakan perkalian silang (*cross product*) antara vektor N dan vektor F.
v. Vektor-vektor NFL yang telah didapat, bersama dengan letak cc-point dalam sistem koordinat global, akan mengisi elemen matrix transformasi