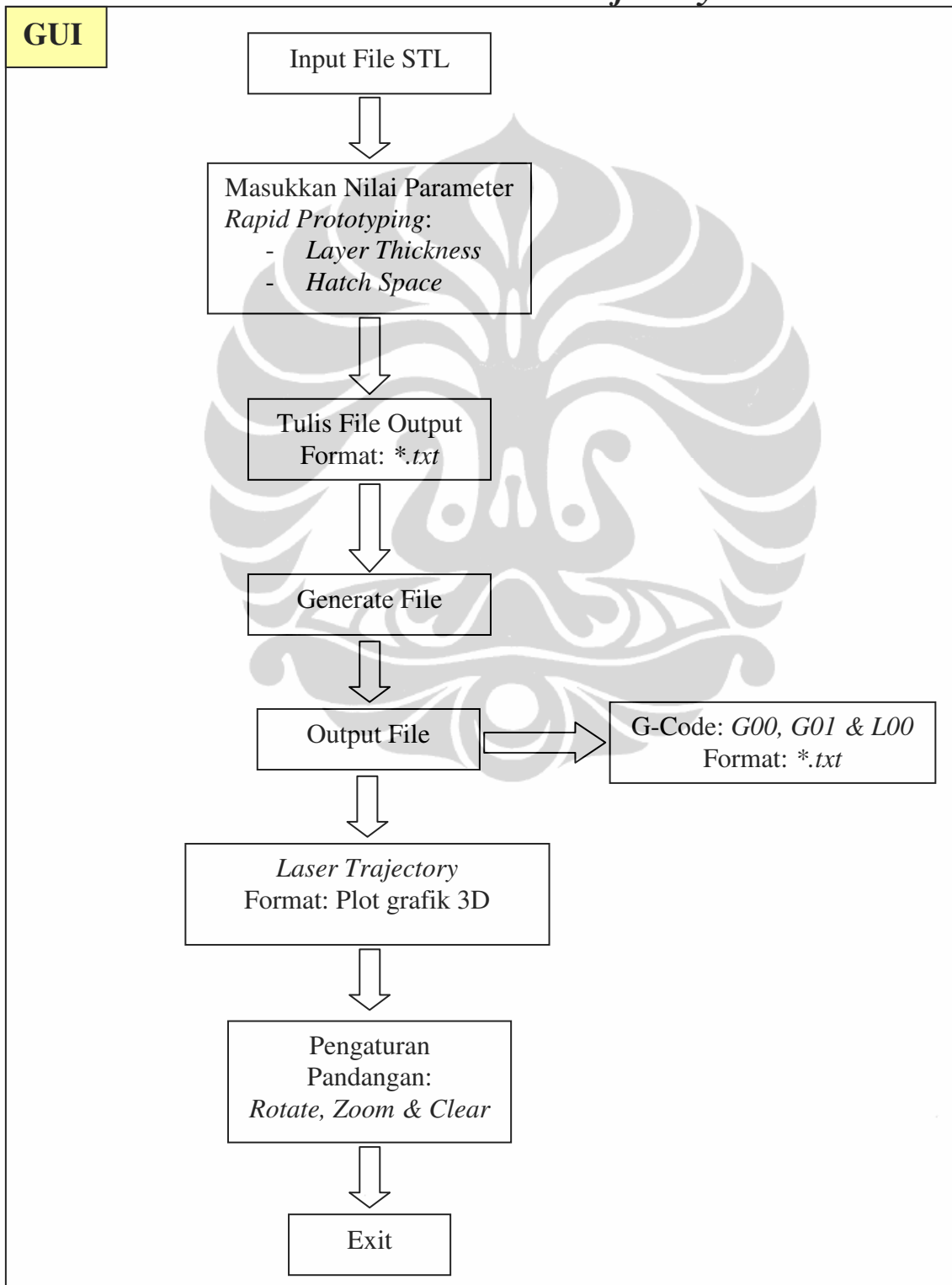


- i. Membuat lintasan untuk setiap layer. Lintasan dibuat dengan terlebih dahulu menentukan titik x sesuai dengan *hatch space* yang telah ditentukan sebelumnya.
- j. Mengurutkan titik potong berdasarkan arah sumbu-y.
- k. Menghubungkan titik-titik lintasan yang telah diurutkan dengan fungsi plot grafik.
- l. Mengeluarkan file output dalam bentuk *.txt*, yang berisi kode mesin L00, G00, dan G01 di ikuti koordinat lintasannya.



BAB IV PENGUJIAN SIMULASI DAN ANALISA HASIL

4.1 GUI Proses Pembuatan *Laser Trajectory*

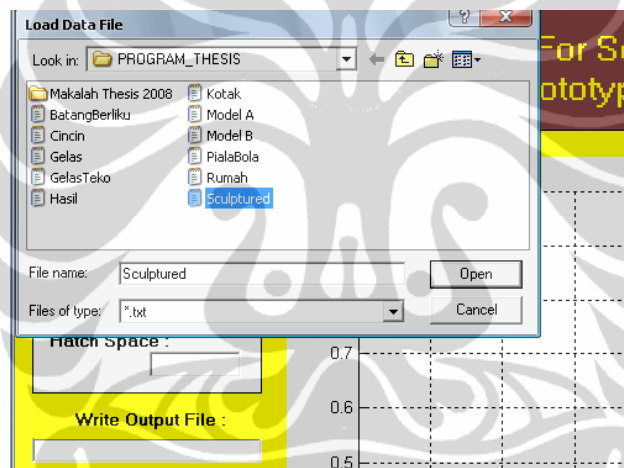


Gambar IV.1 : Algoritma pada GUI

Seperti yang sudah dijelaskan sebelumnya bahwa dalam proses pembuatan di Software pemrograman, ada dua file yang dibuat, yaitu file berisi kode program dan file GUI (*graphic user interface*) sebagai media komunikasi yang menarik. Berikut penjelasan cara kerja dari GUI pada algoritma gambar IV.1 dan tampilan GUI gambar IV.2.

- **Input File STL**

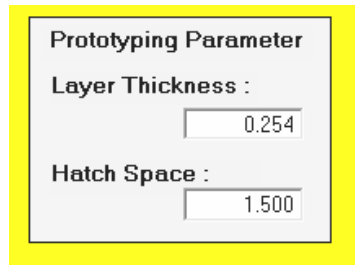
Memasukkan file STL dengan extension .txt kedalam fungsi “generate”. File yang dipilih akan ditampilkan pada display sesuai dengan nama dan tipe file.



- **Prototyping Parameter**

Pada bagian ini terdiri dari dua parameter, yaitu:

- **Layer Thickness:** merupakan parameter ketebalan *layer* bidang potong sumbu-z. Besar kecilnya *layer thickness* tergantung jarak antar layer yang diinginkan. Jarak ini disesuaikan dengan kemampuan solidifikasi raw material ketika terkena pancaran laser.
- **Hatch Space:** merupakan parameter jarak spasi pada sumbu-x untuk setiap lapisan. Besar kecilnya hatch space juga tergantung pada kemampuan solidifikasi raw material ketika terkena pancaran laser, dan tergantung pada ukuran focus laser beam.



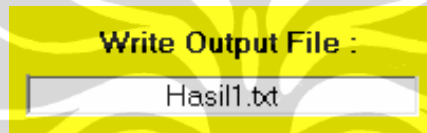
Prototyping Parameter

Layer Thickness :

Hatch Space :

- **Write Output File**

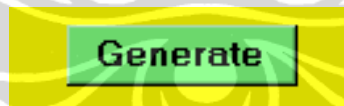
Pada bagian ini terdapat edit text yang harus diisi untuk nama file output. File output ditulis dengan format *nama_file.txt*. File ini digunakan sebagai tempat untuk menaruh file koordinat hasil generate.



Write Output File :

- **Generate**

Merupakan tombol fungsi perintah untuk melakukan perhitungan input file yang sudah dipilih. Hasil generate terdiri dari dua bentuk, yaitu berupa grafik yang berupa lintasan laser, dan berupa format text yang masuk ke file output.



Generate

- **Laser Trajectory**

Merupakan tombol untuk melihat lintasan laser hasil dari generate file. Tampilan hasil akan digambarkan pada grafik dengan garis warna hitam dan merah. Garis warna biru merupakan lintasan dengan pancaran *laser on*, dan warna garis merah merupakan lintasan dengan pancaran *laser off*.



Laser Trajectory

- **Rotate**


Merupakan tombol untuk merubah tampilan grafik dengan memutar dan memilih sesuai posisi pandangan yang diinginkan.



Rotate

- **Zoom**

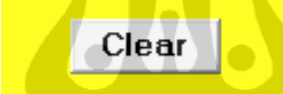
Merupakan tombol untuk memperjelas tampilan grafik sesuai posisi yang diinginkan.



Zoom

- **Clear**

Merupakan tombol untuk me-reset tampilan ke posisi awal.



Clear

- **Open Output File**

Merupakan tombol untuk membuka file hasil generate. File ini berisi G-Code (L00, G00, & G01). Kode G00 dan G01 diikuti dengan koordinat posisi dari titik lintasan.

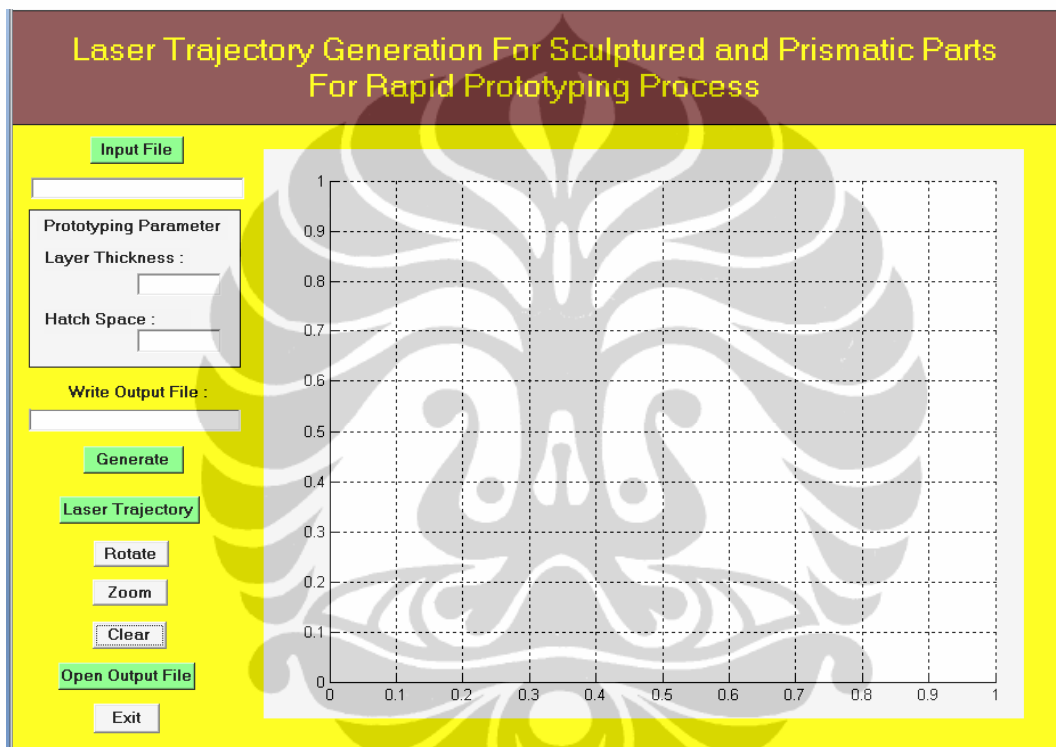
- L00 merupakan tanda posisi laser pada layer dengan kondisi laser off. L00 1 merupakan posisi laser di layer 1, L00 2 merupakan posisi laser di layer 2, dan seterusnya sampai layer terakhir.
- G00 merupakan gerak linier laser ke posisi awal atau posisi yang dituju dengan kondisi laser *off*.
- G01 merupakan gerak linier laser ke posisi yang dituju dengan kondisi laser *on*.



Open Output File

- ***Exit***

Merupakan tombol perintah untuk keluar dari menu.



Gambar IV.2 : Tampilan GUI Program

4.2 Simulasi Program

Simulasi program dilakukan dengan menggunakan komputer yang memiliki spesifikasi,

Processor : Intel Pentium Dual Core 3.20 GHz.

Memory : 512 MB

Skenario simulasi dilakukan dengan tiga model prismatic dan tiga model berkontur dengan nilai parameter *slice interval* dan *hatch space* berbeda.

Model 1

Nama File :

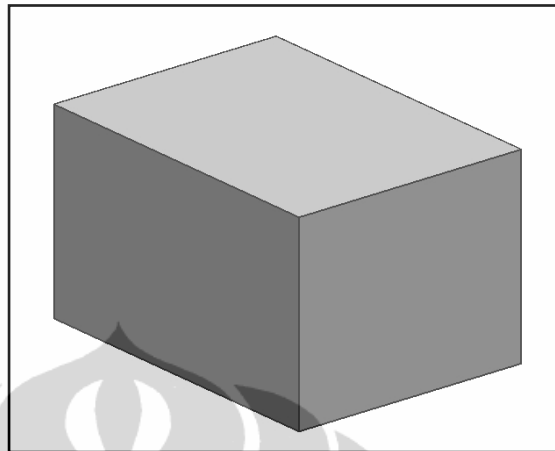
Model_1.txt

Ukuran Model :

P x L x T

40 x 25 x 12 mm

Ukuran file STL : 4 KB



Model 2

Nama File :

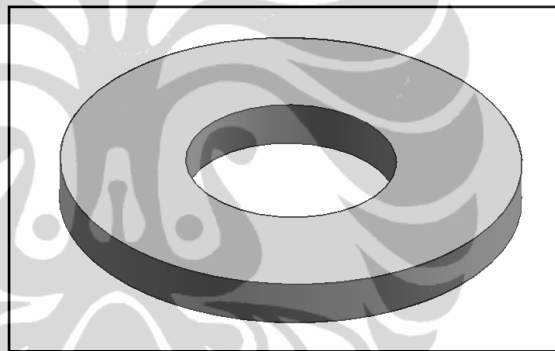
Model_2.txt

Ukuran Model :

Do x T

80 x 6 mm

Ukuran file STL : 107 KB



Model 3

Nama File :

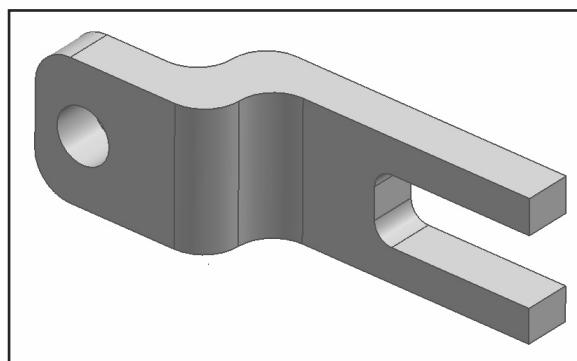
Model_3.txt

Ukuran Model :

P x L x T

80 x 20 x 25 mm

Ukuran file STL : 71 KB



Gambar IV.3 : Model prismatic untuk file input

Model 4

Nama File :

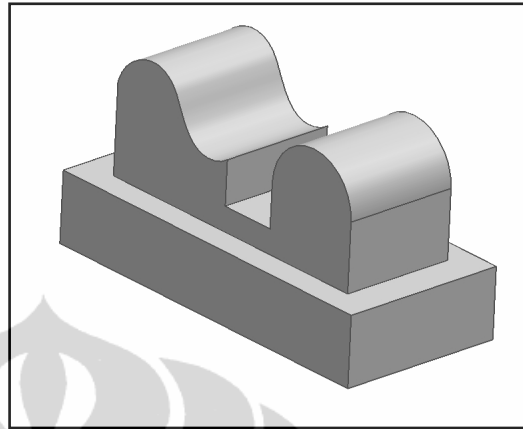
Model_4.txt

Ukuran Model :

P x L x T

60 x 50 x 35 mm

Ukuran file STL : 45 KB



Model 5

Nama File :

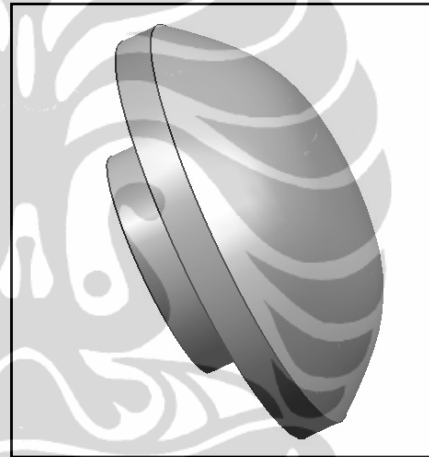
Model_5.txt

Ukuran Model :

Do x T

30 x 20 mm

Ukuran file STL : 405 KB



Model 6

Nama File :

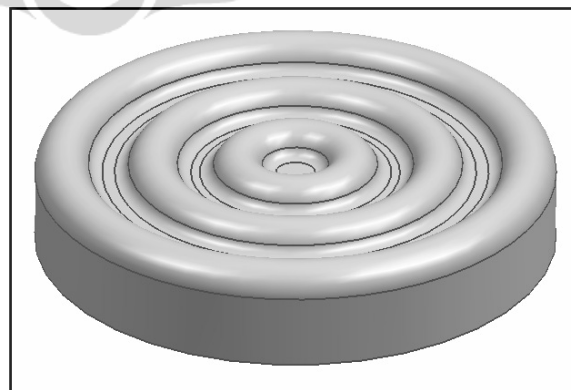
Model_6.txt

Ukuran Model :

Do x T

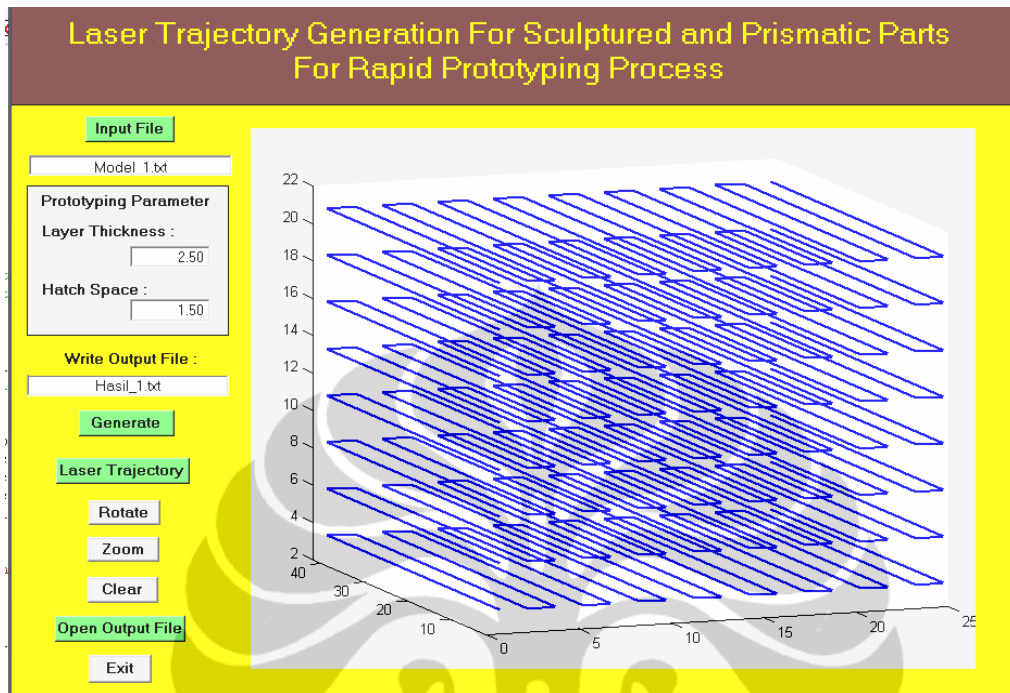
40 x 12 mm

Ukuran file STL : 1,555 KB

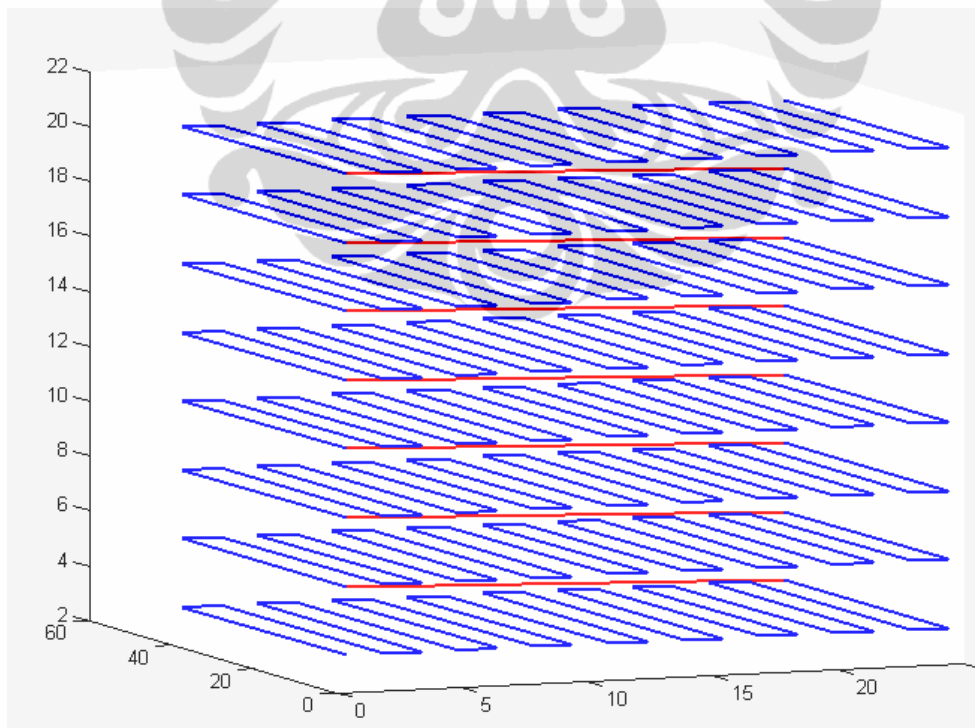


Gambar 4.4 : Model berkontur untuk file input

4.3 Hasil Simulasi Model Prismatik



(a). Hasil generate



(b). Laser trajectory

Gambar IV.5 Hasil simulasi model 1

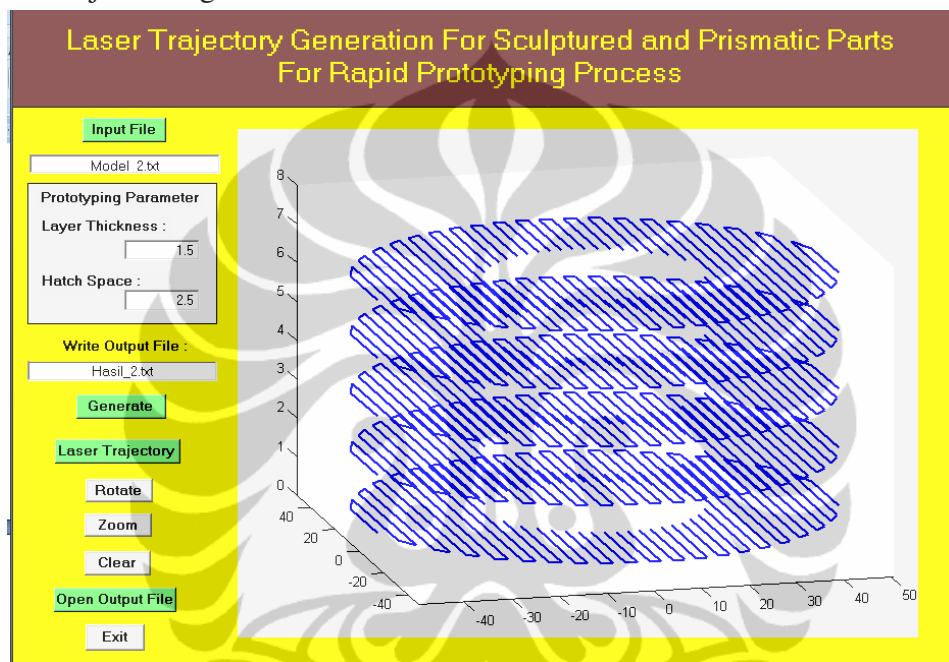
Pada simulasi Model 1, parameter *layer thickness* 2.50 mm, dan *hatch space* 1.50 mm, program bekerja membuat lintasan pada model selama 2 s dengan hasil yang baik. Hasil dari gambar IV.5 menunjukkan lintasan laser dengan arah *directional parallel*. Ada dua warna yang terdapat pada lintasan laser, warna biru menunjukkan lintasan laser untuk setiap layer dengan pacaran *laser on*, sedangkan warna biru merupakan gerak lintasan laser menuju ke posisi layer selanjutnya dengan kondisi *laser off*. Jumlah layer yang terbentuk berjumlah 8 layer, jumlah ini berdasarkan dari jarak pada sumbu-z dibagi dengan *layer thickness*. Semakin besar nilai *layer thickness* yang dimasukkan maka akan mengurangi jumlah layer. Dengan kata lain jumlah layer akan bergantung pada nilai *layer thickness* yang dimasukkan. Jumlah lintasan untuk masing-masing layer yang terbentuk berjumlah 17 lintasan, jumlah ini bergantung pada jarak sumbu-x dibagi dengan *hatch space*. Semakin kecil nilai *hatch space* maka akan menambah jumlah lintasan masing-masing layer.

L00 1 G00 0.750000:2.000000:3.250000 G01 0.750000:42.000000:3.250000 G01 2.250000:42.000000:3.250000 G01 2.250000:2.000000:3.250000 G01 3.750000:2.000000:3.250000 G01 3.750000:42.000000:3.250000 G01 5.250000:42.000000:3.250000	LOO 5 G00 0.750000:2.000000:13.250000 G01 0.750000:42.000000:13.250000 G01 2.250000:42.000000:13.250000 G01 2.250000:2.000000:13.250000
LOO 2 G00 0.750000:2.000000:5.750000 G01 0.750000:42.000000:5.750000 G01 2.250000:42.000000:5.750000 G01 2.250000:2.000000:5.750000	LOO 6 G00 0.750000:2.000000:15.750000 G01 0.750000:42.000000:15.750000 G01 2.250000:42.000000:15.750000 G01 2.250000:2.000000:15.750000
LOO 3 G00 0.750000:2.000000:8.250000 G01 0.750000:42.000000:8.250000 G01 2.250000:42.000000:8.250000 G01 2.250000:2.000000:8.250000	LOO 7 G00 0.750000:2.000000:18.250000 G01 0.750000:42.000000:18.250000 G01 2.250000:42.000000:18.250000 G01 2.250000:2.000000:18.250000
LOO 4 G00 0.750000:2.000000:10.750000 G01 0.750000:42.000000:10.750000 G01 2.250000:42.000000:10.750000 G01 2.250000:2.000000:10.750000	LOO 8 G00 0.750000:2.000000:20.750000 G01 0.750000:42.000000:20.750000 G01 2.250000:42.000000:20.750000 G01 2.250000:2.000000:20.750000 G01 24.750000:2.000000:20.750000 G01 24.750000:42.000000:20.750000

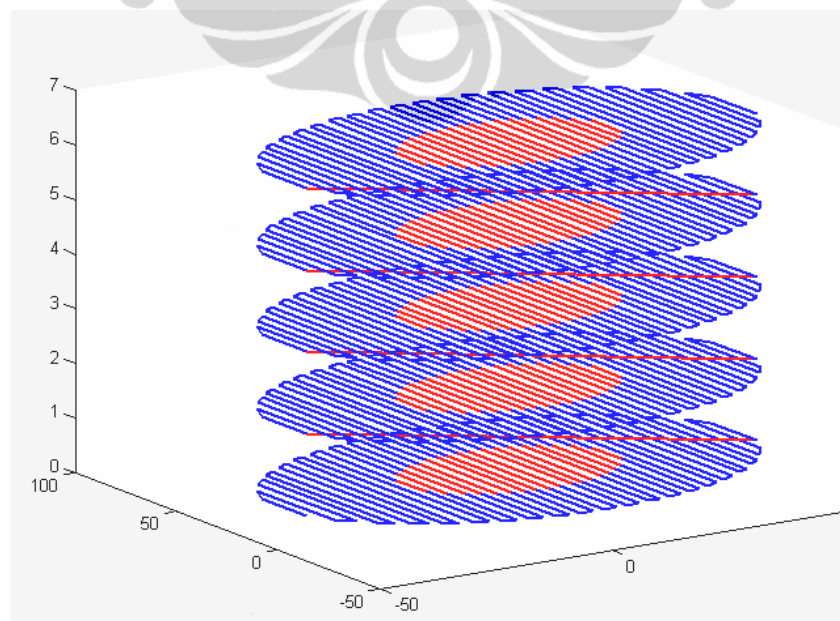
Gambar IV.6 : File output model 1 dengan kode mesin

Seperti yang telah dikemukakan sebelumnya bahwa program yang dihasilkan berupa plot lintasan laser dan file output yang berisi *G-Code* untuk

mesin yang diikuti nilai koordinat. Pada gambar IV.6 menunjukkan file output untuk model ini, file yang lengkap dapat dilihat pada *Lampiran 2a*. Didalam file tersebut berisi kode L00 sekaligus menunjukkan tanda lokasi layer ke berapa, diikuti G00 yang berarti gerak laser menuju koordinat tujuan atau layer berikutnya dengan kondisi *laser off*. Sedangkan G01 merupakan gerak laser menuju koordinat tujuan dengan kondisi *laser on*.



(a). Hasil generate



(b). Laser trajectory

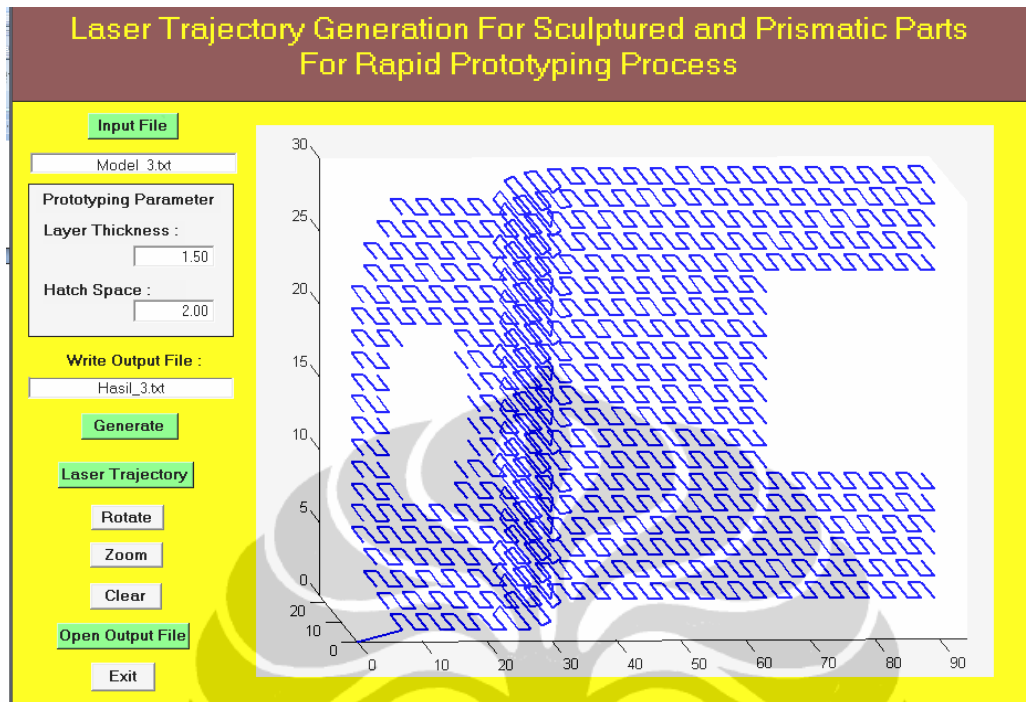
Gambar IV.7 : Hasil simulasi model 2

Pada gambar IV.7 merupakan hasil program pada model yang berbentuk seperti cincin. Proses *generate* dilakukan dengan nilai *layer thicknes* 1.50 mm, dan *hatch space* 2.50 mm. Lintasan laser terbentuk pada model setelah proses berjalan 10 s, dan hasilnya cukup baik walaupun model 2 ini terbentuk dari kurva lingkaran. Jumlah layer sebanyak 5 merupakan hasil bagi dari jarak sumbu-z terhadap nilai *layer thickness*. Sedangkan jumlah lintasan sebanyak 40 merupakan hasil dari pembagian diameter lingkaran dengan *hatch space*.

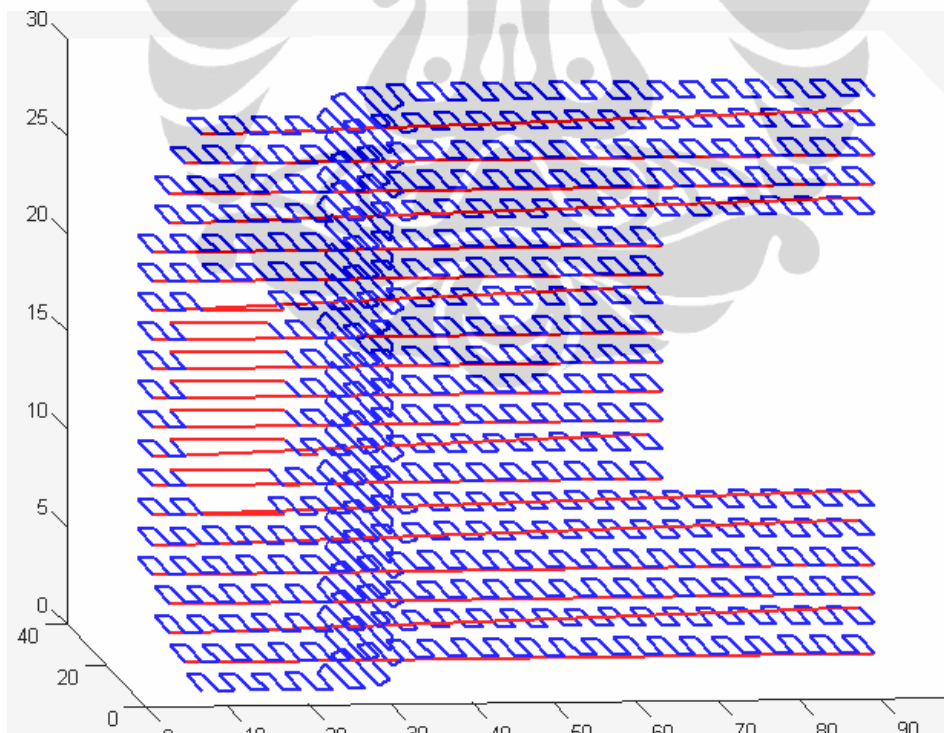
Dari plot yang dihasilkan terdapat dua warna garis. Garis biru merupakan garis lintasan dengan kondisi *laser on*, dan garis merah merupakan garis lintasan dengan *laser off*. Karena model 2 ini merupakan bentuknya cincin yang tengah-tengahnya berlubang, maka hasil *laser trajectory* pada bagian lubang ditengah akan terbentuk garis warna merah, karena bagian tersebut dinyatakan dengan kondisi *laser off*. File output untuk model 2 dapat dilihat pada **Lampiran 2b**.

Model lain seperti hasil *generate* pada gambar IV.8 memperlihatkan *laser trajectory* model 3 dengan nilai *layer thickness* 1.50 mm, dan *hatch space* 2.00 mm. Waktu proses pembuatan model ini sekitar 10 s.

Pada gambar IV.8 terlihat bahwa lintasan laser untuk model 3 terlihat teratur sesuai algoritma. Pada bagian berlubang terlihat garis merah yang berarti lintasan tersebut tidak ada pancaran laser, tetapi hanya sebatas lintasan saja. Sama seperti sebelumnya bahwa jumlah layer bergantung pada nilai *layer thickness* yang dimasukkan, semakin besar ketebalannya maka semakin sedikit layernya. Begitu juga dengan jumlah lintasan untuk tiap-tiap layer tergantung pada nilai *hatch space* yang dimasukkan. File output untuk model 3 dapat dilihat pada **Lampiran 2c**.



(a). Hasil generate



(b). Laser trajectory

Gambar IV.8 : Hasil simulasi model 3

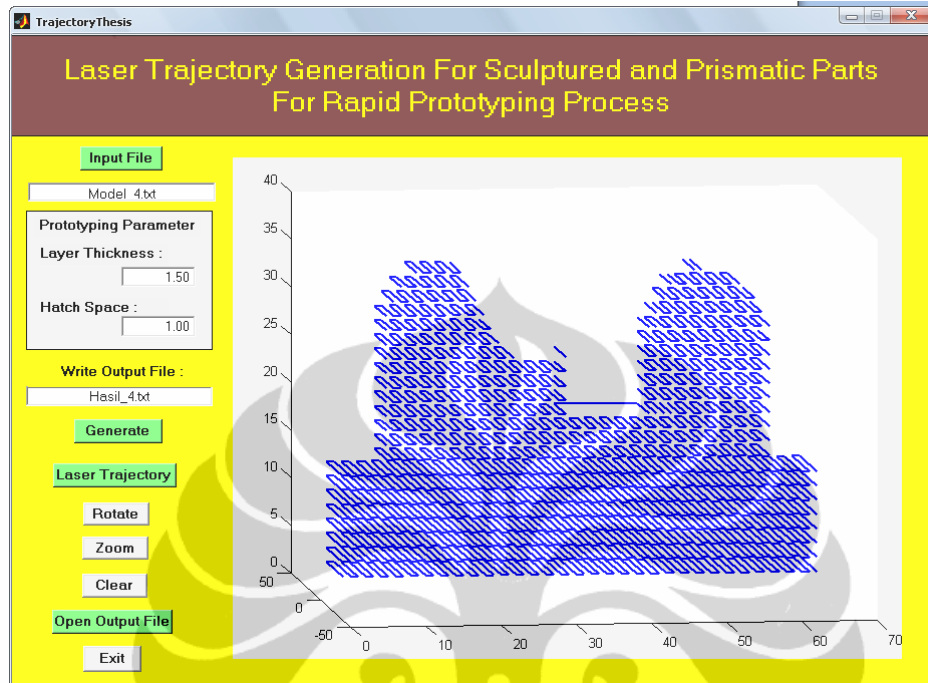
Dengan merubah nilai *layer thickness* dan *hatch space* dari model 3 tersebut dapat diperoleh hubungan antara nilai parameter dengan waktu *generate* model.

Tabel IV.1 Hubungan nilai parameter terhadap waktu generate trajectory model 3 (80 x 20 x 25 mm)

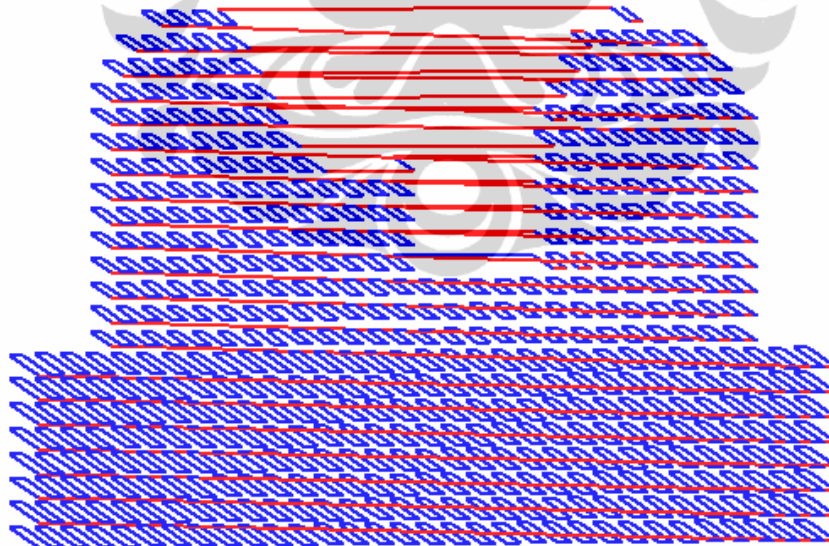
Pengujian	Parameter Prototyping		Jumlah Layer	Waktu generate
	Layer Thickness	Hatch Space		
Simulasi 1	2.00 mm	2.00 mm	15	7 s
Simulasi 2	1.50 mm	1.50 mm	20	9 s
Simulasi 3	1.00 mm	1.00 mm	30	15 s
Simulasi 4	0.50 mm	0.50 mm	60	78 s
Simulasi 5	0.25 mm	0.25 mm	120	1436 s

Dari tabel IV.1 menunjukkan adanya hubungan antara nilai parameter prototyping dengan jumlah layer dan waktu *generate* model. Semakin kecil nilai parameter yang dimasukkan maka semakin banyak jumlah layer yang dihasilkan dan semakin besar waktu yang dibutuhkan untuk membuat model tersebut.

4.4 Hasil Simulasi Model Berkontur



(a). Hasil generate



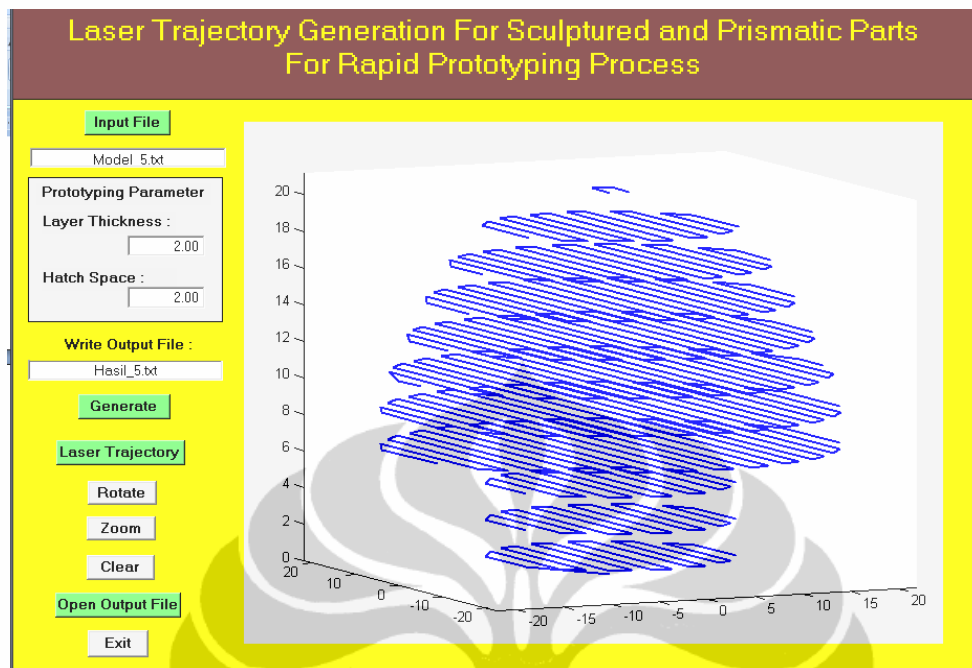
(b). Laser trajectory

Gambar IV.9 : Hasil simulasi model 4

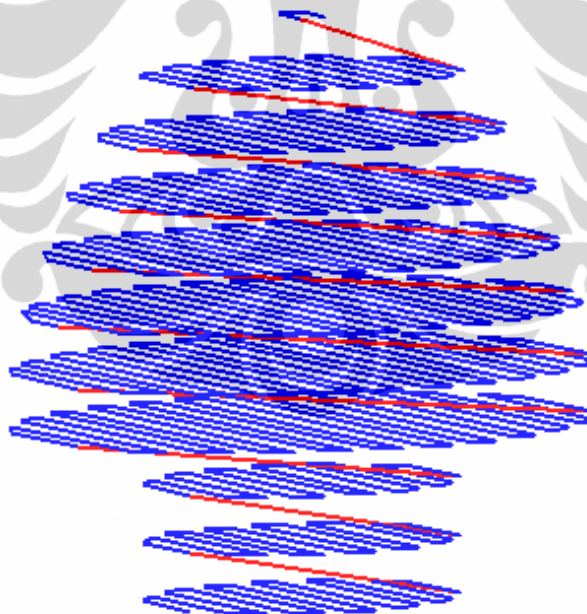
Hasil gambar IV.9 menunjukkan lintasan laser memiliki pola yang teratur walaupun pada benda berkontur. Seperti pada model prisma, jumlah layer tergantung dari nilai *layer thickness* yang dimasukkan, sedangkan jumlah lintasan untuk tiap-tiap layer tergantung nilai *hatch space*. Pada bagian berkontur akan nampak seperti efek bertangga (*stair step effect*) dari setiap layer. Efek ini dapat menyebabkan berkurangnya kualitas permukaan produk yang dihasilkan. Pengaruh ini dapat dikurangi dengan memperkecil nilai *layer thickness* yang dimasukkan. Tetapi akibatnya akan memperlama proses pembuatan karena semakin banyak jumlah layer yang harus di buat lintasannya.

Waktu yang diperlukan untuk proses pembuatan model ini sekitar 6 s, dengan kondisi nilai parameter *layer thickness* 1.50 mm dan *hatch space* 1.00 mm. Nilai ini terlalu besar dan sangat-sangat tidak akurat jika dibuat dalam *prototyping*. Nilai yang dimasukkan ini sebatas untuk simulasi dan pemodelan agar mudah dilihat lintasannya. File output untuk model 4 dapat dilihat pada **Lampiran 2d**.

Model 5 berikut dilakukan dengan kondisi *layer thickness* 2.00 mm, dan *hatch space* 2.00 mm. Dalam gambar IV.10 terlihat jelas bahwa lintasan ini dapat terbentuk dengan baik untuk setiap layernya. Proses generate untuk kondisi ini adalah sekitar 31 s.



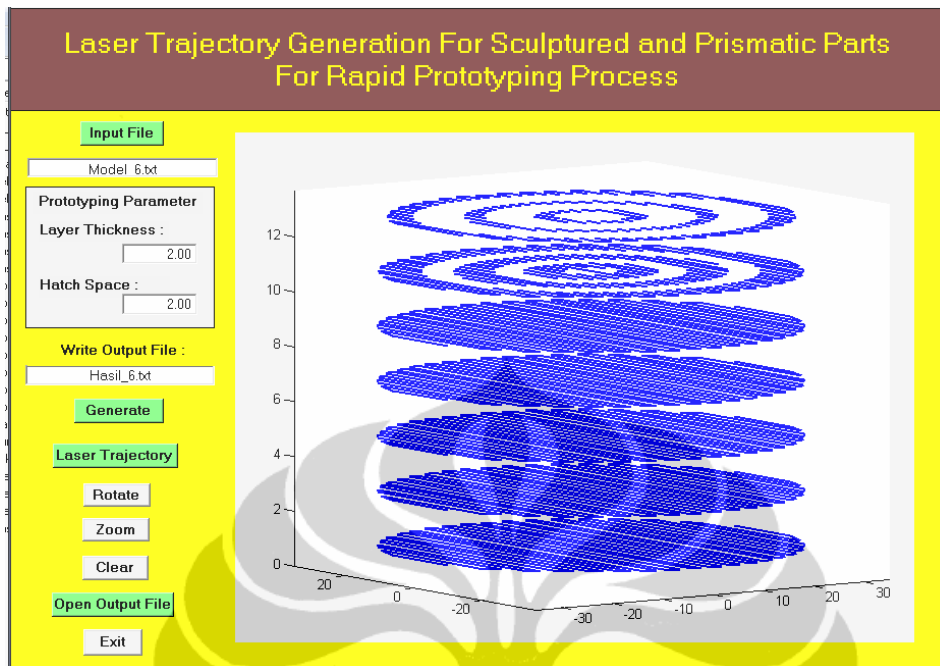
(a). Hasil generate



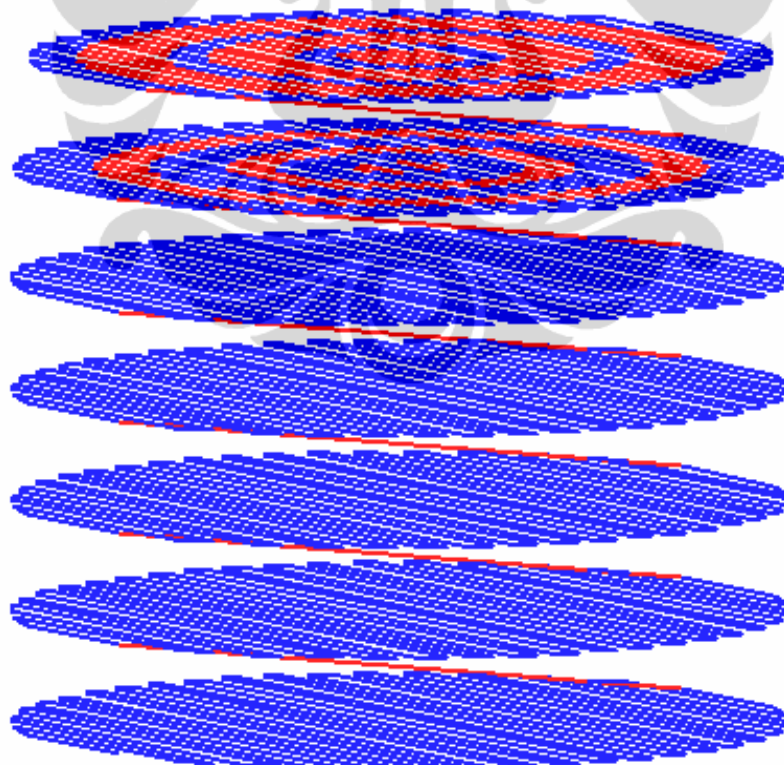
(b). Laser trajectory

Gambar IV.10 : Hasil simulasi model 5

Ada sebelas layer terbentuk dengan kondisi ini. Jumlah ini tergantung dari nilai layer thickness dan hatch space yang dimasukkan. File output untuk model 5 dapat dilihat pada *Lampiran 2e*.



(a). Hasil generate



(b). Laser trajectory

Gambar IV.11 : Hasil simulasi model 6

Model 6 pada gambar IV.11 memperlihatkan *laser trajectory* dengan nilai *layer thickness* 2.00 mm, dan *hatch space* 2.00 mm. Waktu proses pembuatan model ini sekitar 92 s.

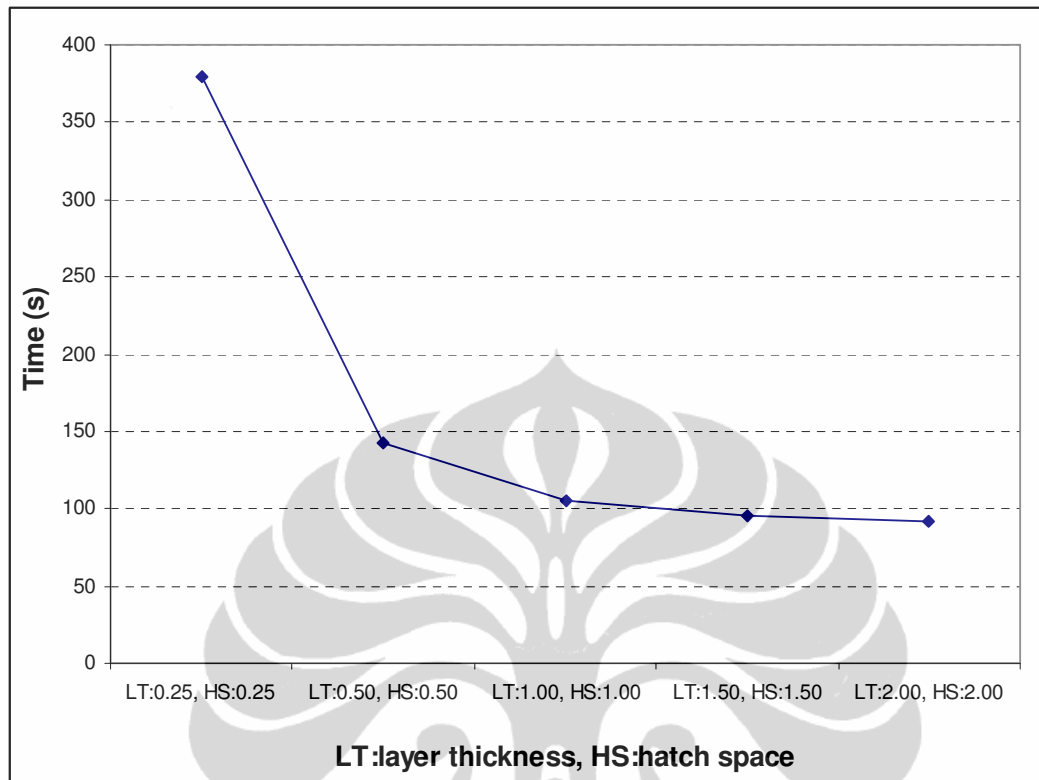
Bentuk permukaan atas dari model ini sebenarnya bergelombang, tetapi karena bentuk grafik di software pemrograman ini tidak diatur sehingga seakan-akan seperti ada celah ditengah. Namun bentuk lintasan yang ada sudah tampak jelas dan mengikuti pola yang sesuai dengan algoritma. File output untuk model 6 dapat dilihat pada *Lampiran 2f*.

Lintasan setiap layer terdiri dari garis-garis lurus dengan jarak antar garis disesuaikan dengan nilai hatch space. Lintasan tersebut dibatasi oleh lingkaran yang terbentuk dari titik-titik hasil perpotongan model *facet* terhadap bidang-z. Dengan merubah nilai *layer thickness* dan *hatch space* dari model 6 tersebut dapat diperoleh hubungan antara nilai parameter dengan waktu generate trajectory model.

Tabel IV.2 Hubungan nilai parameter terhadap waktu generate trajectory model 6 (40 x 20 mm)

Pengujian	Parameter Prototyping		Jumlah Layer	Waktu generate
	Layer Thickness	Hatch Space		
Simulasi 1	2.00 mm	2.00 mm	7	92 s
Simulasi 2	1.50 mm	1.50 mm	9	95 s
Simulasi 3	1.00 mm	1.00 mm	14	105 s
Simulasi 4	0.50 mm	0.50 mm	28	142 s
Simulasi 5	0.25 mm	0.25 mm	56	380 s

Dari tabel IV.2 menunjukkan adanya hubungan antara nilai parameter *prototyping* dengan jumlah layer dan waktu *generate* model berkontur. Semakin kecil nilai parameter yang dimasukkan maka semakin banyak jumlah layer yang dihasilkan dan semakin besar waktu yang dibutuhkan untuk membuat model tersebut. Pada gambar IV.12 terlihat grafik hubungan antara parameter *prototyping* terhadap waktu generate. Hubungan ini tidak linier terutama pada nilai *layer thickness* dan *hatch space* yang kecil terlihat waktu yang semakin lama.



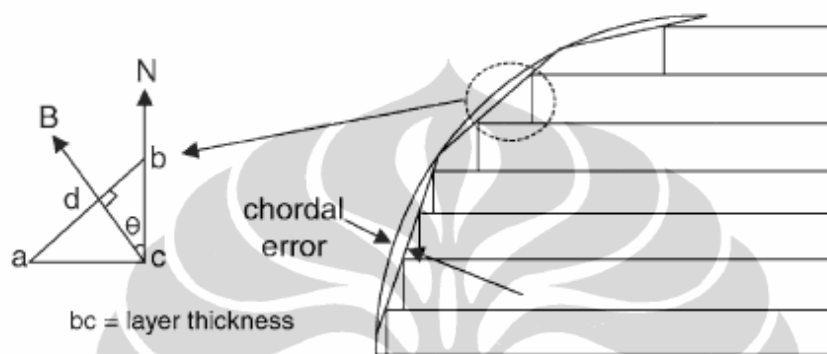
Gambar IV.12 : Grafik hubungan antara parameter prototyping terhadap waktu.

4.5 Pengaruh Paramater Proses Terhadap Keakuratan

Keakuratan permukaan dapat dijelaskan sebagai deviasi geometri dari model CAD sebelumnya terhadap produk. Kerugian keakuratan merupakan *error* yang diperoleh dari tiga tahapan proses, yaitu : (1) *pre-process errors*, (2) *process planning errors*; dan (3) *post-process errors*.

Pre-process errors merupakan *error* yang tidak bisa dipisahkan dari representasi model produk sistim CAD untuk tujuan pertukaran data. Permukaan model produk biasanya direpresentasikan dalam format STL, yang berisi sekumpulan *facet* segitiga. *Error* ini akibat dari nilai *chordal error* pada saat *exporting* file ke format STL. Biasanya nilai ini diset sebelum triangulasi model CAD dilakukan. Semakin kecil nilai *chordal* maka semakin baik permukaan yang terfasetisasi. Penelitian mengenai pengaruh *error* ini belum dilakukan pada saat ini, tetapi dari tesis ini bisa didapatkan gambaran mengenai pengaruh *error* awal untuk penelitian lebih lanjut.

Process planning error, error ini terbentuk pada bagian yang mengalami “*stair-step effect*”, seperti ditunjukkan dalam gambar IV.12, dan pengaruhnya akan menjadi lebih jelas dan nyata pada permukaan berkontur. *Error* ini dapat dihitung dengan tinggi *cusp*, yaitu jarak antara yang diharapkan dengan pendekatan permukaan pada tiap *facet*.



Gambar IV.13 : *Stairs-step effect terhadap ketebalan layer* [6]

Dengan gambar IV.13, *process planning error* bisa ditentukan dengan menghitung nilai *Cusp height* sebagai berikut [6],

$$\text{Cusp height } (h_c) = bc \cos \theta \quad (IV.1)$$

$$\text{Maximum cusp height (MCH)} = \max (h_c) \quad (IV.2)$$

$$\text{Average cusp height (ACH)} = \frac{\sum_{i=1}^{N_f} h_{ci}}{N_f} \quad (IV.3)$$

Jadi error pada saat perencanaan proses dapat dihitung dari nilai rata-rata *cusp height* yang mencerminkan pertengahan deviasi linier dari semua *facet*, dan nilai error maksimum terjadi saat *cusp height* maksimum. Untuk itu penelitian mengenai nilai *error* ini bisa dilakukan pada penelitian lebih lanjut, karena titik potong untuk setiap *z* dari algoritma ini telah diketahui.

Post-process error termasuk error proses, *shrinkage* dan *warpage*. *Error* proses terutama pada mekanisme pergerakan (*deliver*) laser dan sudut dorongan (*induced angle*) dengan permukaan produk yang dibuat. *Error shrinkage* terutama pada solidifikasi produk yang dibuat. Untuk proses *rapid prototyping* yang

menggunakan energi panas untuk solidifikasi/sinter material, seperti SLS, *prototype* cenderung untuk menyusut setelah mendingin. *Warpage* adalah bentuk lain ketidakakuratan yang disebabkan oleh distribusi yang tidak seimbang dari energi panas dan resultan gaya pengikatan. *Error* ini jelas belum bisa dihitung jika belum diimplementasikan ke mesin. Diharapkan dengan adanya pengembangan ini berlanjut ke pembuatan mesin *rapid prototyping* untuk diperoleh hasil yang sempurna dari rangkaian penelitian.

