



TUGAS AKHIR - MN 234802

**OTOMATISASI PROSES PEMBUATAN PROTOTIPE MODEL
LAMBUNG KAPAL DENGAN *3D PRINTING***

**Muhammad Rizky Agung
NRP 5018211048**

**Dosen Pembimbing
Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
Totok Yulianto, S.T., M.T.**

**DEPARTEMEN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025**



TUGAS AKHIR - MN 234802

**OTOMATISASI PROSES PEMBUATAN PROTOTIPE MODEL
LAMBUNG KAPAL DENGAN 3D PRINTING**

**Muhammad Rizky Agung
NRP 5018211048**

**Dosen Pembimbing
Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
Totok Yulianto, S.T., M.T.**

**DEPARTEMEN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025**



FINAL PROJECT - MN 234802

**AUTOMATION OF SHIP HULL PROTOTYPE
MANUFACTURING PROCESS USING 3D PRINTING
TECHNOLOGY**

**Muhammad Rizky Agung
NRP 5018211048**

Supervisor

**Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
Totok Yulianto, S.T., M.T.**

**DEPARTMENT OF NAVAL ARCHITECTURE
FACULTY OF MARINE TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025**

LEMBAR PENGESAHAN

OTOMATISASI PROSES PEMBUATAN PROTOTIPE MODEL LAMBUNG KAPAL DENGAN 3D PRINTING

TUGAS AKHIR

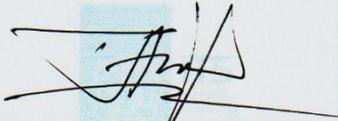
Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Sarjana Departemen Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD RIZKY AGUNG
NRP 5018211048

Disetujui oleh:

Dosen Pembimbing II



Totok Yulianto, S.T., M.T.
NIP 197007311995121001

Dosen Pembimbing I



Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
NIP 1989202011007

Mengetahui,
Kepala Departemen Teknik Perkapalan



Ir Teguh Puranto, S.T., M.T., Ph.D
NIP 199005132014041001

SURABAYA, 24 JULI 2025

LEMBAR REVISI

OTOMATISASI PROSES PEMBUATAN PROTOTIPE MODEL LAMBUNG KAPAL DENGAN *3D PRINTING*

TUGAS AKHIR

Telah direvisi sesuai dengan hasil Ujian Tugas Akhir
Tanggal 16 Juli 2025

Program Sarjana Departemen Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD RIZKY AGUNG
NRP 5018211048

Disetujui oleh Tim Penguji Ujian Tugas Akhir:

1. Prof. Ir. Achmad Zubaydi, M. Eng., Ph.D.

2. Prof. Ir. R. Sjarief Widjaja, Ph.D.



Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.

2. Totok Yulianto, S.T., M.T.



SURABAYA, 24 JULI 2025

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama Mahasiswa / NRP : Muhammad Rizky Agung / 5018211048

Departemen : Teknik Perkapalan

Dosen Pembimbing I / NIP : Dr. Eng. Ir. Yuda Apri Hermawan, S.T., M.T./
1989202011007

Dosen Pembimbing II / NIP : Totok Yulianto, S.T., M.T./ 197007311995121001

dengan ini menyatakan bahwa tugas akhir dengan judul **“OTOMATISASI PROSES PEMBUATAN PROTOTYPE MODEL LAMBUNG KAPAL DENGAN 3D PRINTING”** adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Mengetahui,
Dosen Pembimbing II



Totok Yulianto, S.T., M.T
NIP 197007311995121001

Surabaya, 24 Juli 2025
Mahasiswa,



Muhammad Rizky Agung
NRP 5018211048

Mengetahui,
Dosen Pembimbing I



Dr. Eng. Ir. Yuda Apri Hermawan, S.T., M.T
NIP 1989202011007

HALAMAN PERUNTUKAN

Dipersembahkan kepada kedua orang tua atas segala dukungan dan doanya

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas karunianya Tugas Akhir ini dapat diselesaikan dengan baik.

Pada kesempatan ini Penulis ingin mengucapkan terima kasih kepada pihak-pihak yang membantu penyelesaian Tugas Akhir ini, yaitu:

1. Bapak Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T. dan Bapak Ir. Totok Yulianto, S.T., M.T. selaku Dosen Pembimbing atas bimbingan dan motivasinya selama pengerjaan dan penyusunan Tugas Akhir ini;
2. Prof. Ir. Achmad Zubaydi, M. Eng., Ph.D. dan Prof. Ir. R. Sjarief Widjaja, Ph.D. selaku Dosen Penguji yang telah memberikan kritik dan sarannya untuk perbaikan Laporan Tugas Akhir ini;
3. Ibu Dr. Adi Novitarini Putri, S. T., M. T. selaku Kepala Laboratorium Teknologi Digital Kapal Departemen Teknik Perkapalan FTK ITS atas bantuannya selama pengerjaan Tugas Akhir ini dan atas izin pemakaian fasilitas laboratorium;
4. Seluruh rekan P61 atas bantuannya selama ini

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, sehingga kritik dan saran yang bersifat membangun sangat diharapkan. Akhir kata semoga laporan ini dapat bermanfaat bagi banyak pihak.

Surabaya, 1 Juli 2025

Muhammad Rizky Agung

OTOMATISASI PROSES PEMBUATAN PROTOTIPE MODEL LAMBUNG KAPAL DENGAN 3D PRINTING

Nama Mahasiswa : Muhammad Rizky Agung
NRP : 501821104
Departemen / Fakultas : Teknik Perkapalan / Teknologi Kelautan
Dosen Pembimbing : 1. Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
2. Totok Yulianto, S.T., M.T.

ABSTRAK

Pembuatan prototipe fisik model lambung kapal merupakan tahap krusial dalam siklus desain, namun metode konvensional menggunakan material seperti *fiberglass* memakan waktu lama dan biaya tinggi. Proses digital yang ada saat ini, mulai dari desain CAD hingga persiapan cetak 3D, masih terfragmentasi dan memerlukan banyak intervensi manual yang rentan terhadap kesalahan. Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah sistem otomatisasi yang mengintegrasikan seluruh alur kerja pembuatan prototipe lambung kapal, mulai dari file desain CAD (.step/.iges) hingga menjadi file siap cetak 3D (.gcode). Sistem ini dikembangkan menggunakan bahasa pemrograman *Python*, dengan memanfaatkan pustaka *PythonOCC* untuk memanipulasi geometri CAD dan terintegrasi dengan PrusaSlicer untuk proses *slicing* secara otomatis. Metodologi penelitian mencakup proses pembacaan file desain, solidifikasi model permukaan, penerapan skala, pengosongan bagian dalam (*hollowing*), pemotongan model menjadi beberapa blok, penambahan sambungan, hingga konversi otomatis menjadi .gcode. Seluruh fungsionalitas ini diintegrasikan dalam sebuah antarmuka pengguna grafis (GUI) berbasis PyQt5 untuk mempermudah operasional. Hasil pengujian menunjukkan bahwa sistem berhasil mengotomatiskan seluruh proses dan menghasilkan prototipe fisik dengan akurasi dimensi yang sangat tinggi, dengan simpangan di bawah 1% dari desain digital. Sistem ini terbukti mampu mengurangi waktu produksi dari 2-3 minggu menjadi kurang dari 7 hari, serta memangkas biaya produksi hingga 90% dibandingkan metode manual menggunakan *fiberglass*. Dengan demikian, sistem otomatisasi ini memberikan solusi yang efisien, cepat, dan terintegrasi untuk mempercepat siklus validasi desain di industri perkapalan.

Kata kunci: *Otomatisasi, PythonOCC, 3D Printing, Lambung Kapal, Prototipe, .gcode, CAD/CAM.*

AUTOMATION OF SHIP HULL PROTOTYPE MANUFACTURING PROCESS USING *3D PRINTING* TECHNOLOGY

Author : Muhammad Rizky Agung
Student Number : 5018211048
Department / Faculty : Naval Architecture / Marine Technology
Supervisor : 1. Dr. Eng. Ir. Yuda Apri Hermawan, S. T., M. T.
2. Totok Yulianto, S.T., M.T.

ABSTRACT

The manufacturing of physical ship hull prototypes is a crucial stage in the design cycle, but conventional methods using materials like *fiberglass* are time-consuming and costly. The current digital process, from CAD design to *3D Printing* preparation, remains fragmented and requires extensive manual intervention, which is prone to errors. This research aims to design and develop an automated system that integrates the entire workflow for creating ship hull prototypes, from a CAD design file (.step/.iges) to a ready-to-print 3D file (.gcode). The system was developed using the *Python* programming language, leveraging the *PythonOCC* library for manipulating CAD geometry and integrating with PrusaSlicer for automated slicing. The research methodology includes reading the design file, solidifying the surface model, applying scaling, hollowing out the interior, sectioning the model into several blocks, adding joints, and automatically converting it to .gcode. All these functionalities are integrated into a PyQt5-based Graphical User Interface (GUI) to simplify operation. Test results show that the system successfully automated the entire process and produced a physical prototype with high dimensional accuracy, showing less than 1% deviation from the digital design. The system is proven to reduce production time from 2-3 weeks to under 7 days, and slash production costs by up to 90% compared to the manual *fiberglass* method. Therefore, this automated system provides an efficient, fast, and integrated solution to accelerate the design validation cycle in the shipbuilding industry.

Keywords: Automation, *PythonOCC*, *3D Printing*, Ship Hull, Prototype, .gcode, CAD/CAM.

DAFTAR ISI

LEMBAR REVISI.....	vi
PERNYATAAN ORISINALITAS.....	vii
HALAMAN PERUNTUKAN.....	viii
KATA PENGANTAR.....	ix
ABSTRAK.....	x
ABSTRACT.....	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xviii
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	3
1.3. Tujuan.....	4
1.4. Batasan Masalah.....	4
1.5. Manfaat.....	4
1.6. Hipotesis.....	5
BAB 2 STUDI LITERATUR.....	7
2.1. Dasar Teori.....	7
2.1.1. Model Kapal Untuk Pengujian.....	7
2.1.2. Material Manufaktur Model Kapal dan Ketentuannya.....	8
2.1.3. <i>Computer-Aided Design</i> (CAD) dalam Desain Kapal.....	11
2.1.4. Representasi Geometri 3D dalam CAD.....	15
2.1.5. Transformasi Geometri: Scaling dan Segmentasi.....	17
2.1.6. <i>Python</i>	18
2.1.7. <i>OpenCascade</i> (OCC) dan <i>PythonOCC</i>	20
2.1.8. Anaconda CMD.....	21
2.1.9. <i>Computer-Aided Manufacturing</i> (CAM).....	21
2.1.10. <i>Fundamental 3D Printing (Additive Manufacturing)</i>	24

2.1.11. Slicer dan Prusa Slicer	27
2.1.12. G-CODE dalam <i>Computer-Aided Manufacturing</i>	28
2.2. Tinjauan Pustaka.....	32
2.2.1. Penggunaan <i>PythonOCC</i>	32
2.2.2. <i>Additive Manufacturing</i> untuk Model Teknik	33
2.2.3. Otomatisasi dalam Proses Desain dan Produksi	34
BAB 3 METODOLOGI	37
3.1. Bagan Alir.....	37
3.2. Tahap Pengerjaan.....	38
3.2.1. Identifikasi dan Perumusan Masalah	38
3.2.2. Studi Literatur	38
3.2.3. Pembuatan Konsep Kerja Program	38
3.2.4. Perancangan Program I: Solid – Slicing Approve (.STL)	38
3.2.5. Pengujian Program I	39
3.2.6. Perancangan Program II: Konfigurasi dan Slicing	39
3.2.7. Pengujian Program II	39
3.2.8. Perancangan Program III: GUI	39
3.2.9. Pengujian Program III.....	40
3.2.10. Printing dan Analisa Hasil	40
3.2.11. Pembuatan Laporan Akhir	40
BAB 4 PERANCANGAN DAN IMPLEMENTASI PROGRAM.....	41
4.1. Metode Konvensional dalam Pembuatan Model Lambung 3D Print.....	41
4.2. Perancangan Alur Otomatisasi Program.....	45
4.3. <i>Python</i> dan Anaconda CMD	46
4.4. Program I	47
4.4.1. Identifikasi Library yang Digunakan	47
4.4.2. Import dan Validasi Model	49
4.4.3. Proses Solidifikasi Model	51
4.4.4. Transformasi Skala	52
4.4.5. Proses Hollowing	53
4.4.6. Pemotongan dan Penambahan Sambungan	54
4.4.7. Ekspor File STEP dan STL.....	56
4.5. Program II.....	57

4.5.1. Perancangan Konfigurasi <i>3D Printer</i>	57
4.5.2. Identifikasi Library dan Tools	59
4.5.3. Proses Transformasi Model	60
4.5.4. Proses Slicing File STL	62
4.6. Program III: Antarmuka Pengguna Berbasis PyQt5	63
4.6.1. Identifikasi Komponen GUI	64
4.6.2. Desain Tampilan GUI	64
4.6.3. Eksekusi Proses Backend	66
4.6.4. Penanganan Kesalahan dan Validasi Input	67
BAB 5 Pengujian dan Pembahasan	69
5.1. Uji Coba Program	69
5.1.1. Pengujian Program I	69
5.1.2. Pengujian Program II	72
5.2. Pengujian Program Terintegrasi dengan GUI	77
5.3. Diagram Aktivitas User	79
5.4. Pengujian Program pada Model Lambung Lainnya	80
5.5. Pencetakan dan Perakitan Hasil Fisik	82
5.6. Perbandingan Desain dan Hasil Fisik	85
5.6.1. Perhitungan Simpangan Hidrostatik Setinggi Sarat Kapal	86
5.6.2. Perhitungan Simpangan Hidrostatik Setinggi Tinggi Kapal	91
5.7. Perbandingan Model Kapal <i>Fiberglass</i> dan Hasil Fisik	95
5.8. Analisis Biaya Produksi	97
BAB 6 KESIMPULAN DAN SARAN	101
6.1. Kesimpulan	101
6.2. Saran	101
DAFTAR PUSTAKA	103
LAMPIRAN A HASIL PENGUJIAN PROGRAM I	109
LAMPIRAN B HASIL PENGUJIAN PROGRAM II	113
LAMPIRAN C HASIL PENGUJIAN PROGRAM III (GUI)	117
LAMPIRAN D .GCODE <i>3D Printer</i>	122
BIODATA PENULIS	128

DAFTAR GAMBAR

Gambar 2. 1 Pengujian model pada kolam uji.....	7
Gambar 2. 2 Proses pembuatan kapal model.....	9
Gambar 2. 3 Contoh gambar teknik dalam 2D	12
Gambar 2. 4 Transformasi 2D ke 3D dengan Ekstrusi (sumber: stackoverflow.com).....	13
Gambar 2. 5 Transformasi 2D ke 3D dengan Revolusi (sumber: verisurf.com).....	13
Gambar 2. 6 Transformasi 2D ke 3D dengan Sweeping	14
Gambar 2. 7 Tampilan IGES surface di Autodesk Inventor.....	15
Gambar 2. 8 Perbedaan detail permukaan dari STL dan STEP.....	16
Gambar 2. 9 Transformasi dilatasi (skala).....	17
Gambar 2. 10 Kemudahan bahasa pemrograman python dibanding lainnya	19
Gambar 2. 11 Penerapan <i>pythonOCC</i> pada pembuatan <i>solid wing</i>	20
Gambar 2. 12 CNC Lathe	22
Gambar 2. 13 CNC Milling	23
Gambar 2. 14 3D <i>Printing</i> (sumber: www.ecolink.com)	24
Gambar 2. 15 Hybrid Manufacturing	24
Gambar 2. 16 Contoh 3D print skala industri.....	26
Gambar 2. 17 Proses dari STL menuju file siap print.	27
Gambar 2. 18 Contoh slicer, Orca Slicer	28
Gambar 2. 19 isi file G-code <i>CNC Milling</i>	29
Gambar 2. 20 Isi file G-code proses <i>3D printing</i>	30
Gambar 3.1 Bagan Alir Pengerjaan Tugas Akhir.....	37
Gambar 4. 1 Alur pembuatan model lambung 3D print konvensional.....	41
Gambar 4. 2 Penjelasan lebih lanjut proses split block hingga menjadi file siap print	42
Gambar 4. 3 Rancangan alur metode otomatisasi	45
Gambar 4. 4 Rancangn rinci alur program otomatisasi	45
Gambar 4. 5 Instalasi anaconda.....	46
Gambar 4. 6 Alur Perancangan Program I	47
Gambar 4. 7 Library yang digunakan pada Program I	48
Gambar 4. 8 diagram keputusan fase import file.....	50

Gambar 4. 9 Perhitungan bounding box	50
Gambar 4. 10 Opsi xmin override.....	51
Gambar 4. 11 Lambung dalam bentuk surface	51
Gambar 4. 12 Lambung yang telah solid	52
Gambar 4. 13 Diagram keputusan fase hollowing.....	53
Gambar 4. 14 Ilustrasi pemotongan per block_len	55
Gambar 4. 15 Alur Perancangan Program II	57
Gambar 4. 16 Proses penyetelan parameter	58
Gambar 4. 17 isi dari file penyetelan 3D print.....	58
Gambar 4. 18 Diagram korelasi antar program kepada program GUI	63
Gambar 4. 19 Tampilan dan kegunaan GUI	65
Gambar 5. 1 Running program	69
Gambar 5. 2 Pemilihan file	70
Gambar 5. 3 Proses running program I.....	71
Gambar 5. 4 Output tampilan running program I	71
Gambar 5. 5 Hasil running program I dalam format STL	72
Gambar 5. 6 Tampilan output program II.....	73
Gambar 5. 7 Output gambar program II	73
Gambar 5. 8 Folder output	74
Gambar 5. 9 File hasil output yang dirotasi	74
Gambar 5. 10 Pengecekan hasil rotasi	75
Gambar 5. 11 Tampilan mentah file .gcode.....	76
Gambar 5. 12 File .gcode dilihat dengan <i>software</i> bantuan slicer	76
Gambar 5. 13 Tampilan awal GUI.....	77
Gambar 5. 14 Output akhir tampilan GUI	78
Gambar 5. 15 Output akhir pemrosesan program pada folder.....	78
Gambar 5. 17 Diagram Aktivitas User pada Alur Otomatisasi Terprogram	79
Gambar 5. 18 Pemrosesan lambung planning	81
Gambar 5. 19 Tampilan G-Code variasi lambung planning hasil pemrosesan program otomatisasi	81
Gambar 5. 20 Pemrosesan lambung displasemen	81
Gambar 5. 21 Tampilan G-Code variasi lambung displasemen hasil pemrosesan program otomatisasi	82

Gambar 5. 22 Pemilihan file .gcode di 3D print.....	82
Gambar 5. 23 Proses pencetakan.....	83
Gambar 5. 24 Proses <i>3D Printing</i> Block 1 selesai	83
Gambar 5. 25 Hasil jadi block 1 transom	84
Gambar 5. 26 Pemasangan baut untuk sambungan per block	84
Gambar 5. 27 Hasil akhir setelah dirakit	85
Gambar 5. 28 Tampak depan desain dan hasil fisik	85
Gambar 5. 29 Tampak Samping desain dan hasil fisik	86
Gambar 5. 30 Tampak atas desain dan hasil fisik	86
Gambar 5. 31 Perhitungan lebar dan luas station pada WL	87
Gambar 5. 32 Perhitungan Lebar dan Luas Station pada Sta. 1	92
Gambar 5. 33 Tampak samping model kapal fiber dan hasil fisik	96
Gambar 5. 34 Tampak atas model kapal fiber dan hasil fisik	96
Gambar 5. 35 Pengukuran panjang model kapal fiber dan hasil fisik.....	96
Gambar 5. 36 Pengukuran lebar model kapal fiber dan hasil fisik.....	97
Gambar 5. 37 Penjadwalan operasional <i>3D Printer</i>	98

DAFTAR TABEL

Tabel 2. 1 Ketentuan toleransi dimensi oleh ITTC.....	11
Tabel 5. 1 Perhitungan simpangan desain dan hasil fisik	86
Tabel 5. 2 Perhitungan luasan pada station 0.....	87
Tabel 5. 3 Data luas station dan lebar pada sarat	88
Tabel 5. 4 Hasil perhitungan simpangan luas per station	88
Tabel 5. 5 Hasil perhitungan simpangan koefisien penampang per station.....	89
Tabel 5. 6 Hasil perhitungan hidrostatik.....	90
Tabel 5. 7 Hasil perhitungan V dan LCB pada hasil fisik	90
Tabel 5. 8 Hasil Perhitungan Simpangan setinggi T.....	91
Tabel 5. 9 Perhitungan Luasan setinggi H.....	92
Tabel 5. 10 Data Luasan Tiap Station setinggi H	92
Tabel 5. 11 Simpangan Luasan per Station setinggi H.....	93
Tabel 5. 12 Simpangan Koefisien Penampang per Station setinggi H	94
Tabel 5. 13 Hasil Perhitungan Hidrostatik Setinggi H	94
Tabel 5. 14 Hasil perhitungan Volume Displasemen dan LCB AP.....	95
Tabel 5. 15 Hasil perhitungan simpangan setinggi H.....	95
Tabel 5. 16 Perhitungan simpangan model kapal fiber dan hasil fisik	97
Tabel 5. 17 Lama pekerjaan dan perhitungan biaya	98
Tabel 5. 18 Analisa produksi model kapal.....	99

Halaman ini sengaja dikosongkan

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam beberapa tahun terakhir, industri perkapalan global mengalami pertumbuhan signifikan, didorong oleh meningkatnya permintaan terhadap kapal-kapal baru seperti kapal niaga, kapal khusus, hingga kapal tanpa awak (*unmanned/autonomous vessels*). Tren ini dipicu oleh meningkatnya kebutuhan logistik, eksplorasi laut, pertahanan maritim, dan efisiensi transportasi laut yang berkelanjutan. Inovasi dalam teknologi desain, digitalisasi, serta dorongan terhadap otomasi dan keberlanjutan turut mempercepat pembangunan kapal-kapal generasi baru.

Indonesia sebagai negara maritim dan juga pemilik jumlah kepulauan terbanyak di dunia dengan jumlah pulau tercatat oleh Badan Informasi Geospasial pada tahun 2024 sebanyak 17.380 pulau (Badan Informasi Geospasial, 2024) tentunya memiliki cita-cita sebagai poros maritim dunia. Dengan cita-cita tersebut, Indonesia telah memulai pembangunan “Tol Laut” dengan tujuan untuk menghubungkan antar kepulauan di Indonesia. Tentunya dalam melakukan hal ini, industri perkapalan memegang andil besar dalam penyediaan kapal-kapal yang akan digunakan dalam proyek tersebut. Tercatat oleh Kementerian Perindustrian sepanjang 2019-2021, telah dibangun sebanyak 473 unit kapal oleh galangan kapal dalam negeri dan juga pada periode selanjutnya yaitu Januari-Agustus 2022 sudah ada 363 permohonan pembangunan kapal baru di galangan dalam negeri (Kemenperin, 2022). Hal ini menunjukkan bahwa prospek industri perkapalan di Indonesia semakin menguat dan membuktikan bahwa pembangunan kapal semakin gencar dilakukan.

Dalam pembangunan kapal, pada awalnya dilakukan pendataan awal permintaan dari pemilik kapal terkait dimensi kapal, kegunaan nantinya, dan sebagainya. Setelah mendapatkan data tersebut, para engineer mulai melakukan pembuatan lambung dengan beberapa kriteria yang disesuaikan dengan permintaan pemilik. Langkah-langkah yang dilakukan dalam membuat desain dari kapal yaitu dimulai dengan pembuatan linesplan dengan memperhatikan perhitungan hambatan, *powering*, *payload*, dan sebagainya. Lalu dilakukan pembuatan *general arrangements*, perhitungan daya, pemilihan mesin dan *propeller*, perhitungan *engine propeller*

matching, dan sebagainya. Selanjutnya dalam proses pembangunan kapal baru, salah satu tahapan penting adalah pembuatan model skala kapal. Model ini digunakan untuk berbagai keperluan, mulai dari visualisasi desain awal, uji performa di laboratorium, hingga *free running model test* di lingkungan air terbuka yang mirip dengan kondisi perairan kapal akan beroperasi (ITTC, 2002). Keberadaan model yang representatif sangat penting untuk validasi performa kapal sebelum konstruksi skala penuh dilakukan.

Meskipun simulasi berbasis CFD (*Computational Fluid Dynamics*) telah berkembang pesat dalam beberapa tahun terakhir, pengujian model fisik (*experimental model test*) tetap menjadi metode yang paling dipercaya untuk validasi kinerja kapal. ITTC (International Towing tank Conference), sebagai lembaga internasional yang menetapkan standar uji kapal, menyatakan bahwa pengujian model secara fisik masih dianggap sebagai metode yang paling dapat diandalkan untuk memprediksi kebutuhan daya kapal (ITTC, 2002). Oleh karena itu, ketersediaan model skala yang akurat dan cepat menjadi sangat krusial dalam siklus desain kapal.

Saat ini, pembuatan model kapal skala kecil umumnya masih dilakukan secara manual menggunakan material seperti kayu balsa atau *fiberglass*. Proses ini melibatkan beberapa tahap seperti pembuatan cetakan, laminasi, pengeringan, hingga *finishing*, yang semuanya memerlukan keterampilan teknis tinggi serta waktu yang cukup lama. Berdasarkan studi yang dilakukan oleh Sverre Steen (Steen, 2015), proses pembuatan satu model lambung kapal dengan material *fiberglass* berukuran ± 1 meter dapat memakan waktu rata-rata 2 hingga 3 minggu, tergantung kompleksitas bentuk, teknik laminasi, dan ketersediaan cetakan. Waktu tersebut belum termasuk apabila terjadi kesalahan dalam pembuatan yang dapat memaksa proses diulang dari awal. Hal ini tentu berdampak langsung terhadap produktivitas, terutama dalam proyek yang menuntut pengujian cepat terhadap banyak varian desain.

Salah satu solusi yang kini berkembang adalah penerapan teknologi *Additive Manufacturing* atau yang lebih dikenal dengan *3D Printing*. Teknologi ini memungkinkan pencetakan objek tiga dimensi langsung dari data digital tanpa memerlukan cetakan fisik. Dalam konteks pencetakan benda 3D, *3D Printing* memiliki beberapa keunggulan, seperti penghematan waktu produksi, efisiensi biaya material, kemampuan mencetak bentuk kompleks secara presisi. Namun, dalam praktiknya, alur kerja menuju *3D Printing* masih terfragmentasi. Proses desain dilakukan di perangkat lunak CAD yang selanjutnya desain tersebut dikonversi ke file untuk proses slicing yaitu .STL, dan akhirnya diproses di *software* slicer seperti Cura

atau PrusaSlicer untuk menghasilkan file .GCODE yang berisi instruksi kepada 3D print untuk melakukan tugas percetakan. Tahapan ini sering dilakukan secara manual, tidak terintegrasi, dan rentan terhadap kesalahan teknis maupun inkonsistensi desain.

Sebagai solusi dari permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan program otomatisasi yang menyatukan proses pembuatan model lambung kapal untuk *3D Printing*, dimulai dari format .STEP hingga menjadi file .GCODE yang siap dicetak. Sistem ini dirancang menggunakan *Python* dan *PythonOCC* sebagai engine untuk membaca dan memproses model kapal, serta terintegrasi dengan PrusaSlicer untuk proses slicing. Model lambung kapal yang telah tersedia dapat diinput ke dalam program, lalu diskalakan sesuai kebutuhan dan langsung dikonversi ke format cetak 3D secara otomatis, tanpa intervensi manual. Dengan integrasi ini, proses pembuatan model fisik skala kapal menjadi lebih cepat, efisien, dan dapat direplikasi dengan mudah. Selain itu, pengguna dapat langsung memvisualisasikan desain dalam bentuk nyata melalui hasil cetak 3D, yang sangat membantu dalam proses validasi bentuk dan komunikasi desain.

Penelitian ini bertujuan untuk memberikan solusi inovatif dalam pencetakan model desain lambung kapal yang efisien, cepat, dan dapat diandalkan. Dengan menggabungkan otomatisasi alur desain-ke-produksi dan teknologi *3D Printing*, penelitian ini diharapkan memberikan kontribusi signifikan dalam pengembangan teknologi desain otomatis di bidang maritim, serta menjawab kebutuhan industri akan solusi uji model yang cepat dan rendah biaya ke pasar.

1.2. Perumusan Masalah

Adapun masalah-masalah dalam tugas akhir ini adalah sebagai berikut:

- a. Bagaimana merancang sistem otomatisasi yang mampu mengintegrasikan proses konversi dari file desain lambung kapal berbentuk surface menjadi file siap cetak *3D printing*?
- b. Bagaimana tingkat ketelitian dan kualitas hasil model lambung kapal yang dihasilkan melalui program sistem otomatisasi *3D printing* yang dikembangkan?
- c. Bagaimana efisiensi sistem otomatisasi (dari aspek biaya dan waktu) ini dapat mengurangi waktu dan biaya dalam proses pembuatan model lambung kapal jika dibandingkan dengan metode manual seperti pembuatan model berbahan *fiberglass*?

1.3. Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini sebagai berikut:

- a. Merancang sistem otomatisasi yang mampu mengintegrasikan proses dari file hasil desain lambung kapal (.step/.iges) hingga menjadi file siap cetak (.gcode) menggunakan teknologi *3D Printing*.
- b. Menganalisis tingkat ketelitian dan kualitas hasil model lambung kapal yang dihasilkan melalui proses 3D printing dari sistem otomatisasi yang dibangun.
- c. Melakukan analisis biaya pada proses pembuatan model lambung kapal skala kecil melalui pendekatan otomatisasi berbasis 3D printing dibandingkan metode konvensional seperti pembuatan manual dengan fiberglass.

1.4. Batasan Masalah

Batasan masalah dalam tugas akhir ini sebagai berikut:

- a. Jenis lambung kapal yang digunakan dibatasi untuk kapal permukaan dan *monohull*.
- b. Format file desain awal dibatasi pada format .step dan output untuk *3D Printing* hanya dalam format .gcode.
- c. Ukuran model dan konfigurasi slicing disesuaikan dengan kemampuan dan spesifikasi alat *3D Printer* yang tersedia di laboratorium, tanpa melakukan modifikasi terhadap perangkat keras.
- d. Antarmuka pengguna (GUI) yang dikembangkan bersifat dasar untuk keperluan input file, skala, dan pemilihan konfigurasi printer, bukan sistem industri atau komersial siap pakai.
- e. Penelitian ini tidak mencakup proses uji *towing tank* dan analisis performa hidrodinamika dari model hasil cetak 3D.

1.5. Manfaat

Manfaat yang akan diraih jika tugas akhir ini selesai yaitu:

- a. Membantu mempercepat proses pembuatan model lambung kapal untuk keperluan uji performa atau validasi desain awal.

- b. Menyederhanakan workflow digital dari desain hingga pencetakan 3D dengan sistem terintegrasi, sehingga mengurangi waktu dan potensi kesalahan dalam proses manual.
- c. Mengurangi ketergantungan pada proses manual yang memakan waktu dan biaya tinggi (seperti pembuatan model *fiberglass*).
- d. Menurunkan biaya prototyping model kapal dengan mengganti metode cetak dan laminasi manual menjadi manufaktur aditif berbasis filament seperti PLA.
- e. Menjadi referensi awal untuk pengembangan sistem desain otomatis pada bidang rekayasa maritim dan manufaktur digital.

1.6. Hipotesis

Program otomatisasi berbasis *Python* memungkinkan konversi model lambung kapal dari file desain berbentuk surface menjadi file siap cetak secara otomatis pada *3D Printing*.. Sistem ini diharapkan mampu mempercepat alur kerja dan menurunkan biaya dibandingkan metode konvensional berbasis fiberglass.

Halaman ini sengaja dikosongkan

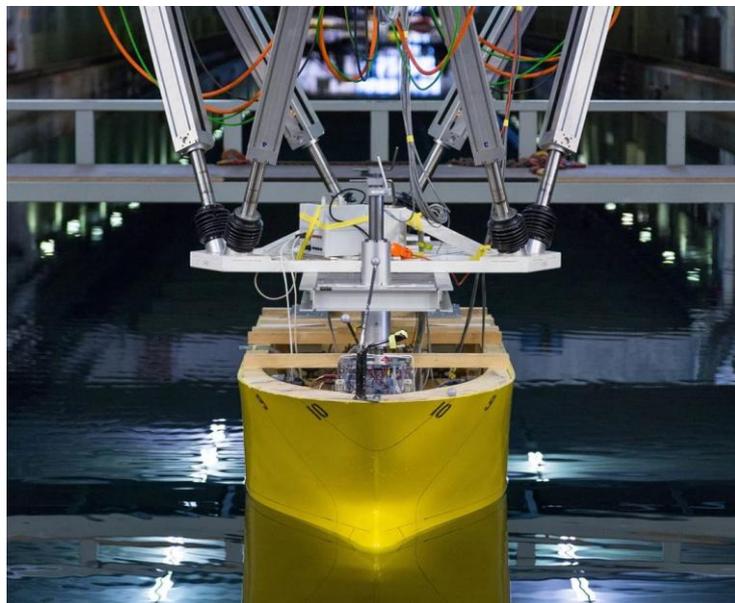
BAB 2

STUDI LITERATUR

2.1. Dasar Teori

2.1.1. Model Kapal Untuk Pengujian

Pemanfaatan model kapal dalam pengujian hidrodinamika, pada dasarnya berakar pada kemampuannya untuk mensimulasikan perilaku fisik kapal dalam berbagai kondisi laut. Tujuan utama dari model ini adalah untuk melakukan eksperimen yang dapat menjelaskan kompleksitas dinamika fluida saat berinteraksi dengan lambung kapal, sehingga para insinyur dapat mengevaluasi karakteristik performa seperti kemampuan manuver, stabilitas, dan respons hidroelastik dalam berbagai skenario operasional. Simulasi fisik seperti pada Gambar 2. 1, memungkinkan evaluasi terhadap interaksi non-linier yang terjadi pada desain kapal yang rumit dan pengaruh lingkungan, yang sering kali sulit dianalisis hanya melalui pendekatan analitik atau numerik, sebagaimana dijelaskan oleh Jiao et al. (Jiao et al., 2016).



Gambar 2. 1 Pengujian model pada kolam uji
(sumber: www.sintef.no)

Model kapal memiliki berbagai fungsi dalam pengujian hidrodinamika. Model digunakan untuk memperoleh turunan hidrodinamika penting, yaitu parameter yang mengkuantifikasi bagaimana kapal merespons gaya-gaya fluida. Pengujian model tertahan (captive model tests) dan pengujian dengan mekanisme gerak planar (Planar Motion

Mechanism/PMM) merupakan metode umum untuk mendapatkan parameter tersebut. Sebagai contoh, Ren et al. menjelaskan bahwa pengujian PMM sangat berguna dalam menganalisis medan aliran di sekitar model untuk memperoleh informasi penting terkait kemampuan manuver (Ren et al., 2020). Selain itu, pengujian dilakukan dalam berbagai kondisi, termasuk variasi bilangan Froude untuk mensimulasikan kecepatan berbeda, sehingga peneliti dapat mengevaluasi pengaruh kecepatan terhadap manuver dan stabilitas. Rameesha dan Krishnankutty menekankan bahwa karakteristik manuver dapat dipengaruhi oleh variasi bilangan Froude selama simulasi (Rameesha & Krishnankutty, 2018), sementara Liu et al. menyatakan pentingnya koefisien-koefisien ini dalam memprediksi perilaku kapal (Liu et al., 2018).

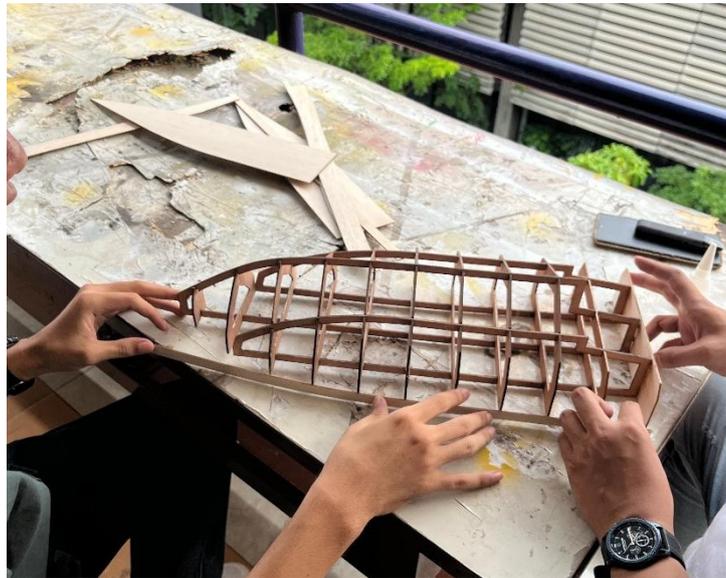
Selain itu, Computational Fluid Dynamics (CFD) memainkan peran penting sebagai pelengkap dalam pengujian model fisik. CFD memungkinkan analisis menyeluruh terhadap aliran fluida di sekitar lambung kapal dalam kondisi representatif, sehingga dapat digunakan untuk validasi dan verifikasi tambahan terhadap hasil eksperimental. Huynh dan Tran menjelaskan keunggulan metode hibrida yang menggabungkan pengujian model konvensional dengan simulasi CFD guna meningkatkan akurasi prediksi parameter hidrodinamika (Huynh & Tran, 2020). Hubungan saling melengkapi antara pendekatan eksperimental dan komputasional ini memperkuat kerangka teori dalam studi pemodelan kapal.

2.1.2. Material Manufaktur Model Kapal dan Ketentuannya

Pembuatan model kapal untuk keperluan pengujian hidrodinamika sangat bergantung pada pemilihan material yang tepat, guna merepresentasikan karakteristik fisik kapal skala penuh secara efektif sekaligus menjamin hasil pengujian yang akurat dan andal. Jenis material yang digunakan dalam pembuatan model kapal umumnya disesuaikan dengan kebutuhan spesifik dari pengujian yang dilakukan, namun pilihan yang umum mencakup fiberglass dan resin.

Fiberglass banyak dipilih sebagai material utama karena memiliki sifat mekanik yang unggul, termasuk rasio kekuatan terhadap berat yang tinggi dan ketahanan terhadap pengaruh lingkungan. Material ini sangat cocok digunakan untuk membuat model yang kuat namun ringan, yang mampu bertahan terhadap gaya tarik (towing) dan simulasi gelombang di fasilitas pengujian hidrodinamika. Selain itu, modulus elastisitas (Young's modulus) dari material yang digunakan harus lebih rendah dibandingkan baja agar dapat merepresentasikan respons elastis kapal skala penuh secara realistis, sehingga simulasi interaksi model dengan air menjadi lebih

representatif (Jiao et al., 2016). Contoh ilustrasi proses pembuatan kapal model fiberglass dapat dilihat pada Gambar 2. 2



Gambar 2. 2 Proses pembuatan kapal model bahan fiberglass

Selain fiberglass, resin juga merupakan material yang umum digunakan, terutama karena kemudahan dalam proses pencetakan (molding) serta karakteristik bobotnya yang ringan. Resin memungkinkan pembentukan bentuk lambung yang kompleks dengan tingkat detail yang tinggi, yang penting untuk menangkap perilaku hidrodinamika secara presisi. Fleksibilitas resin dalam pembuatan model memungkinkan replikasi geometri spesifik lambung kapal sesuai kebutuhan uji.

Selain fiberglass dan resin, material filamen termoplastik seperti PLA (Polylactic Acid) juga mulai digunakan dalam pembuatan model kapal skala kecil menggunakan teknologi 3D printing. Pemanfaatan metode ini memberikan fleksibilitas tinggi dalam proses desain dan produksi model, serta memungkinkan pembentukan geometri kompleks dengan lebih mudah dibanding metode tradisional. Salah satu keunggulan utama dari 3D printing adalah kemampuannya untuk memodularisasi model, memungkinkan pencetakan bagian-bagian lambung secara paralel dengan beberapa printer kecil untuk mengurangi waktu produksi (Choi et al., 2024).

Model kapal hasil cetak 3D biasanya memiliki struktur interior kosong (hollow) untuk memungkinkan penyesuaian berat atau sarat, serta menggunakan sistem perakitan berupa sambungan baut dan lem epoxy. Permukaan model yang dicetak umumnya mengalami efek tangga (staircase effect) dan membutuhkan proses pasca-pencetakan seperti dempul dan amplas agar permukaannya halus dan layak untuk pengujian towing tank. Meskipun memiliki

keterbatasan seperti ketahanan termal dan kemungkinan cacat cetak, pendekatan ini terbukti efektif untuk model uji kapal berukuran kecil dengan biaya dan waktu produksi yang lebih rendah. Validasi dari model hasil 3D printing telah dilakukan melalui pemindaian LIDAR dan uji towing tank, menunjukkan tingkat kesesuaian bentuk dan performa hidrodinamika yang tinggi terhadap model CAD asli, sesuai dengan pedoman ITTC.

Keberhasilan eksperimen ini sangat ditentukan oleh kualitas geometris dan fisik dari model, baik dalam hal akurasi bentuk maupun stabilitas dimensi. Menurut pedoman ITTC, model lambung kapal harus dibangun dengan bentuk geometris yang serupa dengan kapal sesungguhnya, khususnya pada bagian yang bersentuhan dengan permukaan air. Walaupun pada praktiknya terdapat perbedaan bahan dan metode antara model dan kapal skala penuh, deviasi dari bentuk asli harus didokumentasikan dengan jelas (ITTC, 2002). Material yang digunakan dalam pembuatan model lambung antara lain kayu, lilin (wax), foam sel tertutup berdensitas tinggi, dan plastik diperkuat serat (FRP). Umumnya, proses pembuatan dimulai dengan gambar lines plan yang dikonversi menjadi representasi numerik dalam perangkat lunak CAD, lalu dipotong menggunakan mesin CNC atau metode CAM lainnya. Tahap ini sering kali dilanjutkan dengan pemrosesan manual seperti penghalusan permukaan menggunakan amplas grit 300–400 untuk mencapai kualitas permukaan setara dengan permukaan kapal sesungguhnya.

Dalam proses manufaktur, model harus dibentuk mengikuti dimensi moulded dari kapal, dan dibagi menjadi sejumlah station dan waterline yang ditentukan secara sistematis. ITTC merekomendasikan sistem penomoran 10 atau 21 station dari *After Perpendicular* (AP) hingga *Forward Perpendicular* (FP). Waterline diberi identifikasi berdasarkan jarak vertikal dari baseline, dan baseline biasanya ditentukan sebagai bagian atas dari lunas (*keel*) atau garis sejajar dengan waterline desain untuk kapal dengan *keel* miring.

Toleransi dimensi pada model sangat ketat. Seperti yang dijelaskan pada Tabel 2. 1, untuk dimensi panjang toleransi maksimal adalah $\pm 0.05\%$ dari panjang total atau ± 1.0 mm (mana yang lebih besar). Untuk lebar dan draft, toleransi adalah ± 1.0 mm, sedangkan untuk bukaan-bukaan pada lambung, toleransi ditetapkan sebesar ± 1.0 mm juga. Toleransi ini penting untuk memastikan model dapat digunakan secara sah dalam uji hidrodinamika dengan hasil yang dapat diandalkan dan direproduksi (ITTC, 2002).

Tabel 2. 1 Ketentuan toleransi dimensi model oleh ITTC
(sumber: ITTC)

Parameter	Toleransi	Keterangan
Panjang (Length)	$\pm 0.05\%$ LoA atau ± 1.0 mm	Diambil nilai yang lebih besar antara dua batas tersebut
Lebar (Breadth)	± 1.0 mm	Diterapkan pada keseluruhan bagian melintang model
Draft (Kedalaman)	± 1.0 mm	Kedalaman dari baseline ke waterline
Bukaan-bukaan pada lambung	± 1.0 mm	Termasuk lubang bow thruster, intake, dan fitur geometri lainnya

Stabilitas geometris model seiring waktu juga menjadi perhatian penting. Model yang dibuat dari wax, misalnya, dapat mengalami perubahan dimensi akibat fluktuasi suhu. Sebuah perubahan suhu sebesar 5°C dapat menyebabkan perubahan panjang sekitar 0.15% (setara 10 mm untuk model sepanjang 7 meter). Oleh karena itu, model harus disimpan dan digunakan dalam kondisi lingkungan yang terkendali, dan setiap perubahan dimensi perlu didokumentasikan sebelum pengujian dilakukan.

Selain bentuk utama lambung, beberapa fitur tambahan (appendages) model juga harus seperti rudder, bilge *keel*, strut, dan shaft bracket juga harus dibentuk sesuai dengan bentuk eksternal desain, dengan toleransi pemasangan ± 0.5 mm terhadap posisi desain. Permukaan appendage pun harus memiliki kualitas finishing yang setara dengan lambung utama.

2.1.3. *Computer-Aided Design* (CAD) dalam Desain Kapal

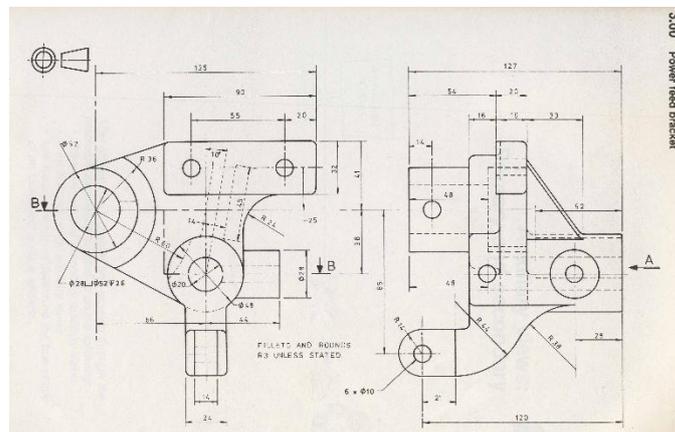
Desain Berbantuan Komputer (*Computer-Aided Design*/CAD) merupakan integrasi perangkat lunak yang canggih yang memungkinkan pengguna untuk membuat, memodifikasi, menganalisis, dan mengoptimalkan desain dalam berbagai bidang teknik dan arsitektur. Esensi dari CAD terletak pada kemampuannya dalam menyajikan simulasi visual yang akurat sebelum suatu produk diproduksi secara fisik. Dengan demikian, perbaikan dan penyesuaian dapat dilakukan pada tahap desain awal, sehingga meningkatkan efisiensi keseluruhan proses perancangan (Kolbasin & Husu, 2018; Ramadhan et al., 2019)

Fungsionalitas sistem CAD dapat dikategorikan ke dalam beberapa aspek utama, seperti pemodelan, simulasi, dan visualisasi. Pada tahap awal, perangkat lunak CAD memungkinkan perancang untuk membuat sketsa dua dimensi (2D), yang kemudian dapat dikembangkan menjadi model tiga dimensi (3D). Transisi dari 2D ke 3D ini sangat penting karena membantu memahami hubungan spasial antar komponen desain, sekaligus mengidentifikasi potensi kesalahan yang dapat muncul saat proses manufaktur (Kolbasin & Husu, 2018). Selain itu,

sistem CAD dapat menerapkan algoritma kompleks untuk melakukan optimasi struktural, sehingga penggunaan material menjadi lebih efisien sambil tetap memenuhi spesifikasi desain dalam hal kekuatan, berat, dan keberlanjutan (Temnov, 2022).

Transformasi desain dua dimensi (2D) menjadi model tiga dimensi (3D) merupakan tahapan penting dalam pengembangan produk modern. Desain 2D, yang biasanya berupa gambar teknik atau blueprint, menyediakan informasi dasar mengenai bentuk dan dimensi objek. Namun, model 3D memberikan representasi spasial yang lebih lengkap, yang memungkinkan visualisasi menyeluruh, analisis teknis, serta integrasi dengan proses manufaktur otomatis seperti Computer-Aided Manufacturing (CAM) dan pencetakan 3D (Raja & Fernandes, 2008).

1. Digitisasi Gambar 2D

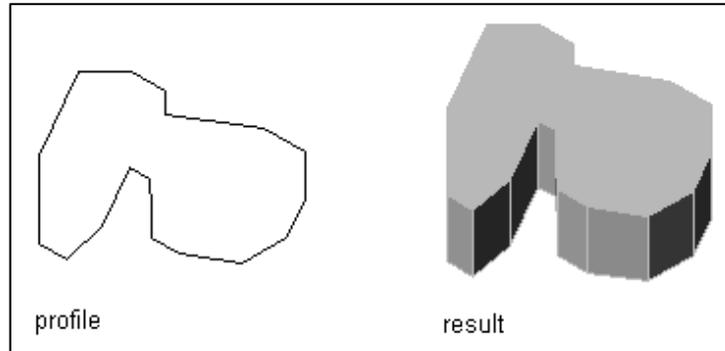


Gambar 2. 3 Contoh gambar teknik dalam 2D

Proses konversi dimulai dengan mengubah gambar kertas menjadi data digital melalui pemindaian (scanning) dan pengolahan menggunakan perangkat lunak CAD seperti pada Gambar 2. 3. Format umum yang digunakan antara lain DXF dan DWG, yang kompatibel dengan sebagian besar perangkat lunak pemodelan 3D. Metode ini memungkinkan gambar yang sudah ada untuk digunakan sebagai dasar pembuatan model 3D tanpa perlu menggambar ulang secara manual (Pipes, 2003).

2. Ekstrusi (Extrusion)

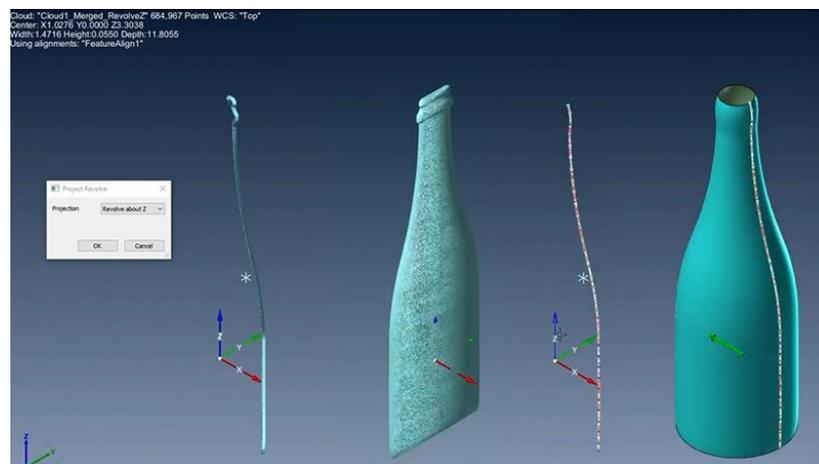
Ekstrusi adalah teknik yang paling sederhana dan umum digunakan untuk mengonversi bentuk 2D menjadi 3D. Proses pada gambar Gambar 2. 4 dilakukan dengan memberi ketebalan atau panjang tertentu pada profil 2D sehingga menjadi objek solid. Teknik ini sering digunakan dalam pembuatan komponen mekanis yang berbentuk dasar sederhana seperti pelat, balok, atau profil struktural (Zeid, 2014).



Gambar 2. 4 Transformasi 2D ke 3D dengan Ekstrusi
(sumber: stackoverflow.com)

3. Revolusi (Revolve)

Metode revolusi yang diilustrasikan pada Gambar 2. 5, digunakan untuk membentuk objek simetris dengan memutar profil 2D mengelilingi sumbu tertentu. Teknik ini efektif untuk membuat benda seperti poros, roda, katup, atau komponen berbentuk silindris. Keunggulannya adalah mampu menghasilkan bentuk kompleks yang sulit dicapai dengan ekstrusi biasa (Shih, 2018).

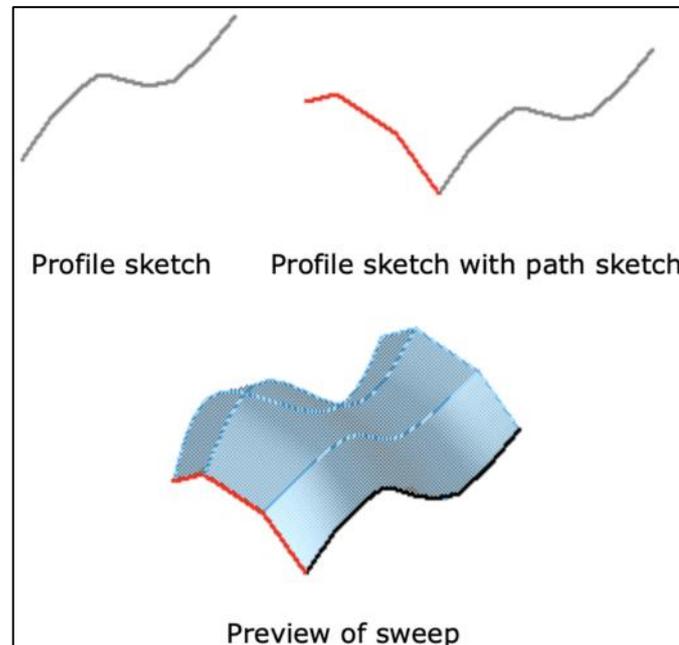


Gambar 2. 5 Transformasi 2D ke 3D dengan Revolusi
(sumber: verisurf.com)

4. Sweeping dan Lofting

Metode sweep yang diilustrasikan pada Gambar 2. 6, digunakan ketika profil 2D perlu mengikuti jalur tertentu untuk membentuk objek 3D, sedangkan loft memanfaatkan beberapa penampang 2D pada posisi berbeda yang dihubungkan menjadi satu bentuk permukaan atau solid. Kedua metode ini banyak digunakan pada desain

dengan geometri kompleks seperti bodi kendaraan, pipa berbelok, atau casing produk elektronik (Shih, 2018).



Gambar 2. 6 Transformasi 2D ke 3D dengan Sweeping
(sumber: scan2cad.com)

5. Pemodelan Parametrik dan NURBS

Pemodelan berbasis parameter memungkinkan pengguna mengontrol dimensi dan bentuk objek secara dinamis dengan mengubah nilai parameter tertentu, sehingga mempermudah iterasi desain. Selain itu, penggunaan NURBS (Non-Uniform Rational B-Spline) memungkinkan pembuatan permukaan yang halus dan akurat, yang penting dalam desain produk dengan kontur bebas seperti peralatan rumah tangga atau komponen aerodinamika (Piegl & Tiller, 1997).

6. Validasi Model 3D

Setelah konversi selesai, model 3D harus divalidasi dengan memeriksa kesesuaian dimensi dan geometri terhadap data desain awal. Validasi ini penting untuk memastikan akurasi sebelum digunakan dalam analisis atau proses manufaktur lebih lanjut (Zeid, 2014)

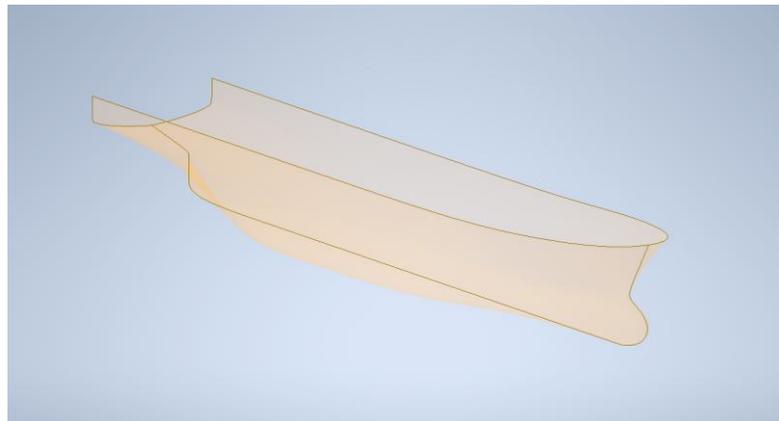
Lebih dari sekadar menggambar dan memodelkan, teknologi CAD juga memungkinkan simulasi rinci untuk mengevaluasi bagaimana desain akan berperilaku di bawah berbagai kondisi tekanan dan operasional. Hal ini sangat penting di bidang seperti rekayasa struktur, di mana pemahaman terhadap efek dinamis sangat mempengaruhi aspek keselamatan dan

fungsionalitas desain (Temnov, 2020). Otomatisasi dalam perhitungan dan pemodelan prediktif memungkinkan para insinyur dan arsitek untuk melakukan analisis menyeluruh tanpa perlu intervensi manual, yang secara signifikan mengurangi waktu iterasi dalam proses desain (Kolbasin & Husu, 2018).

Desain Berbantuan Komputer (*Computer-Aided Design/CAD*) telah secara signifikan mentransformasi proses perancangan kapal dengan meningkatkan kolaborasi, efisiensi, dan akurasi. Perkembangan CAD dalam industri perkapalan, khususnya sejak akhir abad ke-20, sangat bergantung pada kemajuan teknologi pemodelan 3D dan teknik desain parametrik. Penggunaan sistem tersebut memungkinkan representasi geometri yang kompleks, yang sangat penting dalam arsitektur kapal modern, termasuk bentuk lambung kapal dan komponen serta sistem internal yang menjamin fungsionalitas dan kepatuhan terhadap standar regulasi.

2.1.4. Representasi Geometri 3D dalam CAD

Dalam sistem desain dibantu komputer (CAD), berbagai format file digunakan untuk menyimpan dan bertukar data bentuk dan struktur objek 3D. IGES, STEP, dan STL adalah tiga format yang paling umum digunakan. IGES dan STEP dirancang untuk menyimpan data produk secara detail, termasuk bentuk, ukuran, dan hubungan antar bagian. Sementara itu, STL lebih sederhana dan hanya menyimpan bentuk permukaan objek dalam bentuk segitiga, sehingga banyak digunakan untuk keperluan *3D Printing* (Mattausch et al., 2014; Pitteri, 2020).

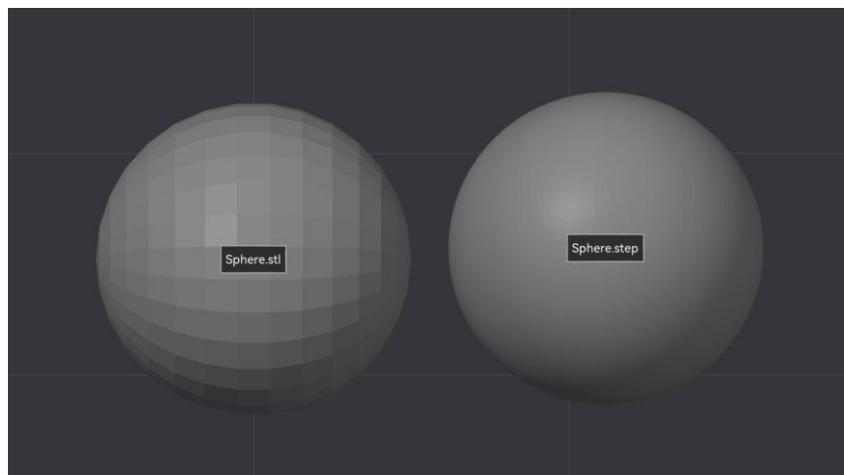


Gambar 2. 7 Tampilan IGES surface di Autodesk Inventor

IGES merupakan salah satu format tertua dalam pertukaran data CAD. Format ini mendukung representasi berbagai jenis entitas geometri seperti titik, kurva, permukaan, hingga solid. IGES banyak digunakan untuk mentransfer data antar perangkat lunak CAD yang berbeda. Contoh bentuk file IGES surface dapat dilihat pada Gambar 2. 7. Sementara itu, STEP hadir sebagai pengembangan dari IGES yang lebih modern, karena tidak hanya menyimpan

bentuk geometri tetapi juga informasi semantik dan hubungan antar komponen, sehingga cocok untuk membangun digital twin produk (Tautges, 2001).

Cara ketiga format ini menyimpan data geometri berbeda-beda. IGES menggunakan struktur hierarkis yang terdiri dari entitas geometris dan relasi antar objek (Terada & Kurumatani, 2005). STEP menggunakan pendekatan berbasis objek (object-oriented), yang memungkinkan informasi produk lebih lengkap, termasuk komponen dan hubungan antar bagian dalam suatu rakitan. Di sisi lain, STL menyederhanakan bentuk 3D menjadi kumpulan segitiga yang membentuk permukaan objek. Meskipun mudah diproses, STL tidak mendukung informasi tambahan seperti warna, bahan, atau fitur desain yang kompleks (Navangul et al., 2013) seperti pada Gambar 2. 8.

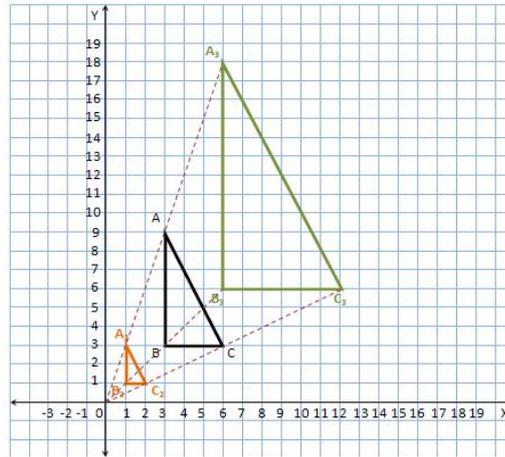


Gambar 2. 8 Perbedaan detail permukaan dari STL dan STEP
(sumber: forum.bambulab.com)

Akurasi dalam representasi geometri sangat penting agar objek yang dirancang bisa diwujudkan secara fisik dengan tepat. Ada beberapa faktor yang mempengaruhi ketelitian model CAD, seperti kesalahan pengukuran, konversi data, atau keterbatasan metode pemodelan (Turek et al., 2024). Oleh karena itu, perlu dilakukan evaluasi terhadap model CAD, misalnya melalui metode *reverse engineering* yang membandingkan model digital dengan bentuk nyata. Representasi geometri 3D sangat relevan dalam berbagai bidang, seperti desain teknik, arsitektur, hingga pencitraan medis. Seiring berkembangnya teknologi AI dan machine learning, model CAD kini juga digunakan dalam sistem otomatisasi desain dan pengenalan objek berbasis data visual. Beberapa penelitian telah menggabungkan data CAD dengan deep learning untuk meningkatkan kemampuan sistem dalam memahami bentuk dan fungsionalitas produk secara otomatis.

2.1.5. Transformasi Geometri: Scaling dan Segmentasi

Transformasi geometri sendiri merupakan teknik penting yang digunakan untuk memanipulasi bentuk, gambar, atau objek, misalnya dengan mengubah ukuran (scaling), memutar (rotation), atau menggeser posisi (translation). Dalam konteks ini, scaling dan segmentasi sering digunakan bersamaan untuk berbagai aplikasi seperti analisis gambar, pemrosesan objek, hingga pembuatan model 3D yang lebih efisien dan detail.



Gambar 2. 9 Transformasi dilatasi (skala)
(sumber: wordpress.com)

Scaling adalah proses untuk mengubah ukuran suatu objek tanpa mengubah bentuk dasarnya. Transformasi yang diperagakan dalam Gambar 2. 9 memungkinkan objek ditampilkan dalam berbagai ukuran sesuai kebutuhan, misalnya untuk menyesuaikan resolusi, melakukan normalisasi bentuk, atau memperbesar fitur tertentu agar lebih mudah dikenali (Minh & Vetterli, 2005; Sghaier & Lepage, 2016). Dalam aplikasi seperti digital watermarking, scaling digunakan agar tanda air (*watermark*) tetap bisa dikenali meskipun gambar telah diubah ukurannya (Terzija & Geisselhardt, 2006). Hal ini menunjukkan pentingnya skala dalam menjaga ketahanan data digital terhadap perubahan ukuran. Skala ini umumnya dinyatakan dalam bentuk skala panjang (λ), yang ditulis sebagai persamaan 2.1;

$$\lambda = \frac{LoA \text{ Kapal}}{LoA \text{ model}} \quad (2.1)$$

Dimana:

λ : Faktor skala

LoA : Panjang Keseluruhan (Length Overall)

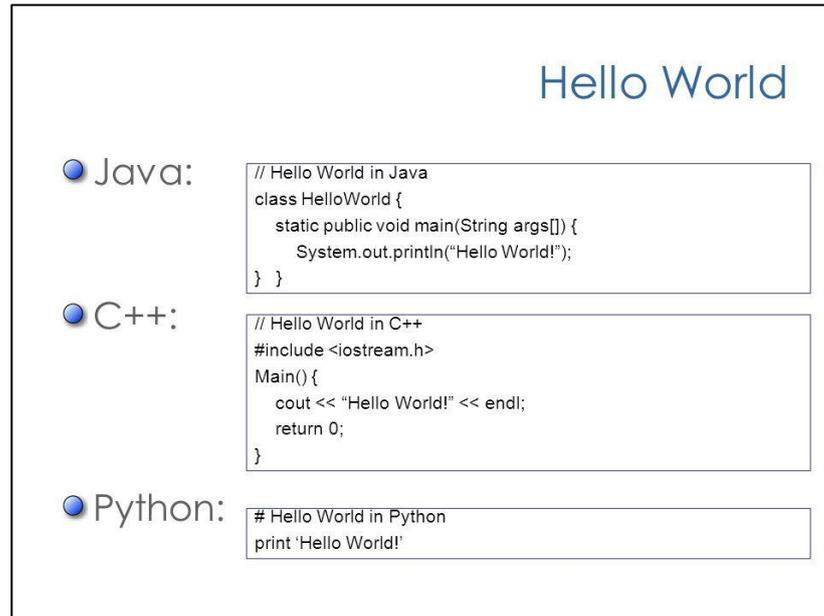
Misalnya, jika kapal aslinya sepanjang 100 meter dan modelnya 2 meter, maka skala panjangnya adalah 1:50. Pemilihan skala model harus mempertimbangkan keseimbangan antara keterbatasan fasilitas pengujian (misalnya panjang towing tank), akurasi representasi geometri kapal, dan efek skala (scale effects) terhadap hasil pengujian.

Sementara itu, segmentasi merupakan proses membagi gambar atau model menjadi beberapa bagian atau area yang lebih kecil dan bermakna. Tujuannya adalah menyederhanakan analisis dan fokus pada bagian-bagian penting, seperti fitur spesifik dalam suatu gambar. Ada beberapa metode segmentasi, seperti yang berbasis wilayah (region-based), tepi (edge-based), atau model matematis (model-based) (Nikolaidis & Pitas, 2001).. Selain itu, model deformasi (deformable models) dapat digunakan untuk mengekstrak beberapa objek sekaligus dalam berbagai tingkat detail, sehingga hasil segmentasi menjadi lebih akurat dan tahan terhadap distorsi (Fletcher et al., n.d.).

2.1.6. Python

Python merupakan bahasa pemrograman yang sangat fleksibel dan banyak digunakan dalam berbagai bidang, seperti ilmu komputer, teknik, analisis data, dan komputasi ilmiah. Filosofi desain *Python* yang menekankan pada kesederhanaan dan keterbacaan kode menjadikannya pilihan ideal bagi pemula maupun programmer berpengalaman (Zheng, 2023). Sejak dikembangkan pada akhir 1980-an, *Python* telah berkembang menjadi bahasa multiparadigma yang mendukung pendekatan berbasis objek maupun fungsional (Tasoluk & Tanrikulu, 2023). Fleksibilitas ini memungkinkan pengguna memilih pendekatan pemrograman sesuai kebutuhan proyek mereka, mulai dari pengembangan web hingga riset komputasi.

Salah satu kekuatan utama *Python* terletak pada ekosistem pustaka (library) dan kerangka kerja (framework) yang sangat luas, khususnya untuk kebutuhan komputasi ilmiah dan analisis data. Pustaka seperti NumPy dan SciPy sangat penting dalam pemrosesan data numerik dan optimasi komputasi. Sementara itu, pustaka visualisasi seperti Matplotlib dan Seaborn mempermudah pengguna dalam menyajikan data dalam bentuk grafik atau plot yang mudah dipahami. Kombinasi antara sintaks *Python* yang sederhana dengan alat numerik yang kuat telah mempermudah para peneliti dalam mengelola dan menganalisis data secara efisien.



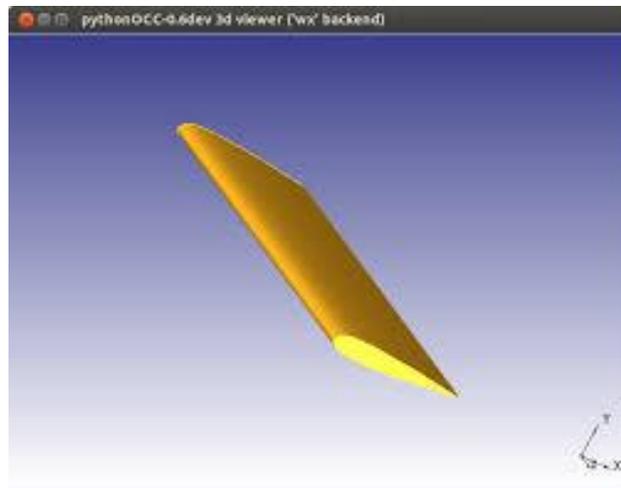
Gambar 2. 10 Kemudahan bahasa pemrograman python dibanding lainnya (sumber: www.uct.ac.za)

Python juga dikenal karena kemampuannya untuk berinteraksi dengan bahasa pemrograman lain, yang memperluas kegunaannya di berbagai bidang. Contohnya, dengan PyCUDA atau PyOpenCL, *Python* dapat memanfaatkan kekuatan GPU untuk mempercepat komputasi berskala besar (Pierro, 2013). Selain itu, *Python* juga sering dijadikan sebagai antarmuka *scripting* untuk perangkat lunak lain, menjadikannya sangat berguna dalam lingkungan multi-bahasa seperti dalam bidang neuroinformatika (Pecevski et al., 2014). Kemampuan untuk terintegrasi dengan berbagai teknologi ini sangat penting dalam alur kerja komputasi modern. Contoh kemudahan dalam pemrograman *python* dapat dilihat dalam Gambar 2. 10.

Di bidang pendidikan, *Python* juga memiliki peran besar. Sintaksnya yang mudah dipahami serta komunitasnya yang luas menjadikannya bahasa yang ideal untuk pengajaran pemrograman (Guo, 2021; Djokic-Petrovic et al., 2016). Hal ini membuat banyak lembaga pendidikan memilih *Python* sebagai bahasa pertama untuk siswa yang baru belajar coding. Perkembangan kursus online dan berbagai sumber belajar gratis juga membuat pembelajaran *Python* semakin mudah diakses oleh masyarakat umum, baik pelajar maupun profesional, terutama dalam bidang data science dan kecerdasan buatan (Zúñiga-López & Avilés-Cruz, 2020; Hota, 2022).

2.1.7. *OpenCascade (OCC) dan PythonOCC*

Dalam dunia teknik dan desain berbasis komputer, *Open CASCADE Technology (OCC)* dan *PythonOCC* menjadi dua komponen penting yang banyak digunakan untuk pemodelan 3D dan pengolahan geometri secara komputasional. OCC adalah pustaka perangkat lunak open-source yang dibuat khusus untuk menangani data CAD (*Computer-Aided Design*) 3D. Dengan OCC, pengguna bisa melakukan berbagai hal seperti membuat bentuk 3D, memvisualisasikannya, dan mengekspor atau mengimpor file CAD antar format. Sementara itu, *PythonOCC* adalah antarmuka *Python* untuk *OCC*, yang memungkinkan pengguna mengakses semua fitur OCC menggunakan bahasa pemrograman *Python* yang lebih mudah dan fleksibel (Rodríguez-Sevillano et al., 2023).



Gambar 2. 11 Penerapan *pythonOCC* pada pembuatan *solid wing*
(sumber: www.pythonocc.wordpress.com)

Penggunaan *PythonOCC* sangat membantu dalam berbagai proses pemodelan. Salah satunya adalah mengubah model 3D menjadi gambar 2D, atau melakukan manipulasi geometri seperti pemotongan, penggabungan, dan transformasi bentuk (K. Wang et al., 2022). *PythonOCC* juga bisa digunakan untuk aplikasi yang lebih kompleks seperti optimasi bentuk dan pembuatan model parametrik. Contohnya, telah diterapkan untuk merancang bentuk sayap pesawat kecil yang bisa berubah bentuk (*morphing*) secara otomatis (Rodríguez-Sevillano et al., 2023) seperti pada gambar Gambar 2. 11.

Secara keseluruhan, OCC dan *PythonOCC* memperlihatkan bagaimana keduanya menjadi alat penting dalam pengembangan teknologi desain modern. Dengan kemampuan untuk membuat, memodifikasi, dan mengotomatisasi model 3D secara komputasional, *OCC* dan *PythonOCC* membantu menjembatani proses desain tradisional dengan pendekatan digital yang lebih efisien dan inovatif.

2.1.8. Anaconda CMD

Anaconda adalah distribusi *Python* open-source yang mencakup Conda sebagai manajer paket, serta berbagai pustaka ilmiah seperti NumPy, Pandas, SciPy, dan Jupyter. Anaconda menyediakan platform lengkap untuk mengelola paket, lingkungan virtual, dan dependensi—yang sangat penting untuk menjaga reproducibility dan efisiensi dalam pemrograman ilmiah dan alur kerja otomatisasi. Anaconda CMD (Command Prompt) adalah terminal perintah yang telah dikonfigurasi khusus oleh Anaconda. Berbeda dengan command prompt standar bawaan sistem operasi, Anaconda CMD langsung terhubung dengan perintah conda, serta memiliki jalur lingkungan (environment path) dan binary *Python* yang telah dikonfigurasi secara otomatis.

Selain itu, penerapan praktis dari Anaconda dapat terlihat dalam konteks toolkit analisis dan visualisasi data, seperti CoolBox, yang menggunakan saluran Bioconda untuk manajemen paket melalui Anaconda (Xu et al., 2021). Ini menunjukkan fleksibilitas dan aksesibilitas ekosistem analisis data yang difasilitasi oleh Anaconda CLI, serta mendukung alur kerja terintegrasi bagi para peneliti dan pengembang di bidang biologi komputasi dan genomika.

Selain itu, CLI Anaconda sangat mudah digunakan. Cukup dengan beberapa perintah, pengguna dapat menginstal pustaka atau membuat lingkungan kerja sesuai kebutuhan. Ini sangat membantu, terutama bagi pemula di bidang pemrograman ilmiah, karena mengurangi kerumitan teknis dan mempercepat proses kerja. Bagi pengguna yang sudah berpengalaman, CLI ini mempercepat setup proyek dan membuat alur kerja jadi lebih efisien.

Keunggulan lainnya adalah Anaconda mendukung berbagai pustaka populer dalam machine learning dan data science, sehingga pengguna bisa langsung membuat lingkungan kerja tanpa harus khawatir soal konflik versi. Ini membuat Anaconda menjadi bagian penting dalam toolkit riset modern yang berbasis data.

Singkatnya, Anaconda CLI adalah alat penting untuk komputasi ilmiah masa kini karena mudah digunakan, cepat, fleksibel, dan mendukung berbagai pustaka penting. Peran utamanya dalam menyederhanakan manajemen lingkungan dan mendukung integrasi alat analisis data menjadikannya sangat bermanfaat bagi para peneliti dari berbagai bidang.

2.1.9. Computer-Aided Manufacturing (CAM)

Computer-Aided Manufacturing (CAM) adalah sistem teknologi yang mengintegrasikan perangkat lunak komputer dengan mesin manufaktur untuk meningkatkan efisiensi proses produksi. Integrasi ini memungkinkan pembuatan instruksi pemesinan dan

sinyal kontrol secara otomatis, sehingga mempercepat alur kerja produksi dan meningkatkan efisiensi secara keseluruhan. Contoh implementasi CAM dapat dilihat pada Gambar 2. 12. Dengan memanfaatkan teknologi CAD/CAM, produsen dapat mencapai tingkat presisi yang tinggi, mengurangi waktu produksi, dan menekan biaya pembuatan berbagai produk, termasuk komponen kapal yang kompleks (Ikubanni et al., 2022).



Gambar 2. 12 CNC Lathe
(sumber: www.fractory.com)

Hubungan antara CAM dan pendahulunya, yaitu *Computer-Aided Design (CAD)*, bersifat fundamental. CAD berfungsi sebagai langkah awal dalam proses desain, di mana model digital rinci dibuat dan kemudian diterjemahkan oleh sistem CAM menjadi instruksi manufaktur spesifik. Integrasi yang mulus ini menghasilkan alur kerja yang lebih terkoordinasi, sehingga setiap revisi pada tahap desain dapat segera diadaptasi ke tahap produksi, mengurangi potensi kesalahan dan mendorong inovasi (Boyd et al., 2023). Dalam industri perkapalan, sebagai contoh, model geometris presisi tinggi yang dibuat dengan CAD dapat diubah menjadi rangkaian perintah digital untuk mesin CNC yang digunakan dalam pemesinan lambung dan komponen kapal kompleks lainnya (Hao et al., 2012).

Salah satu keunggulan utama dari teknologi CAM terletak pada kemampuannya dalam mengotomatiskan berbagai proses melalui penggunaan sistem *Computer Numerical Control (CNC)*. Mesin CNC mengontrol alat dan peralatan berdasarkan instruksi terprogram, yang pada akhirnya meningkatkan konsistensi dan akurasi dalam proses manufaktur. Teknologi ini sangat penting dalam industri kapal, di mana pembuatan struktur besar membutuhkan tingkat ketelitian yang sangat tinggi (Ikubanni et al., 2022). Otomatisasi yang ditawarkan CAM turut mengurangi ketergantungan pada tenaga kerja manual dan meningkatkan kecepatan produksi,

memungkinkan pembuatan komponen kompleks tanpa hambatan yang umum terjadi pada proses manual (al., 2020).

Berbagai jenis CAM digunakan di berbagai sektor industri, mencerminkan keberagaman teknologi dan metodologi yang diterapkan. Jenis-jenis utama CAM meliputi:

1. *Subtractive Manufacturing*



Gambar 2. 13 CNC Milling
(sumber: www.hubs.com)

Metode CAM tradisional ini melibatkan proses penghilangan material dari blok padat (seperti logam atau plastik) untuk membentuk komponen yang diinginkan. Teknik yang digunakan mencakup milling pada Gambar 2. 13, turning, dan drilling, yang umumnya dikendalikan melalui sistem CNC (Computer Numerical Control). Misalnya, beberapa studi menunjukkan bahwa teknik subtractive mampu menjaga tingkat presisi tinggi dalam pembuatan restorasi gigi karena kemampuannya memproses material keras dengan akurat (Karatasli et al., 2011; Assiri et al., 2023).

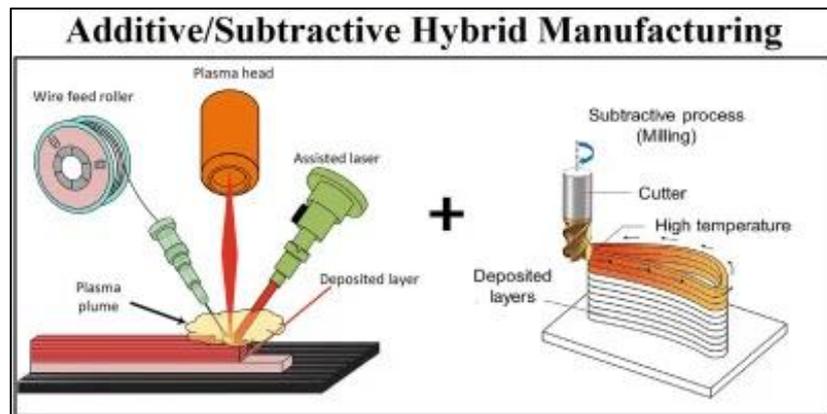
2. *Additive Manufacturing* (Pencetakan 3D)

Teknologi ini membentuk komponen secara berlapis dari model digital, menggunakan material seperti plastik, logam, dan keramik. 3D printing memungkinkan pembuatan geometri kompleks yang sulit atau tidak mungkin dicapai dengan metode subtractive. Dalam aplikasi kedokteran gigi, misalnya, additive manufacturing telah digunakan untuk membuat prosthesis khusus dan panduan bedah, menunjukkan kemajuan signifikan dalam hal presisi dan kustomisasi (Assiri et al., 2023; Fujioka-Kobayashi et al., 2024). Proses manufaktur dengan 3D printing dapat dilihat pada Gambar 2. 14.



Gambar 2. 14 3D Printing
(sumber: www.ecolink.com)

3. Hybrid Manufacturing



Gambar 2. 15 Hybrid Manufacturing
(sumber: www. blog.spatial.com)

Pendekatan ini menggabungkan proses *subtractive* dan *additive*. Contohnya pada Gambar 2. 15, suatu komponen dapat dibentuk terlebih dahulu dengan teknik additive, kemudian dilakukan penyelesaian atau detailing menggunakan metode subtractive untuk meningkatkan kualitas permukaan dan ketelitian dimensi. Sinergi antara kedua metode ini sangat berguna dalam pembuatan komponen kompleks sambil mengoptimalkan penggunaan material (Mainjot et al., 2016; Lee et al., 2008).

2.1.10. Fundamental 3D Printing (Additive Manufacturing)

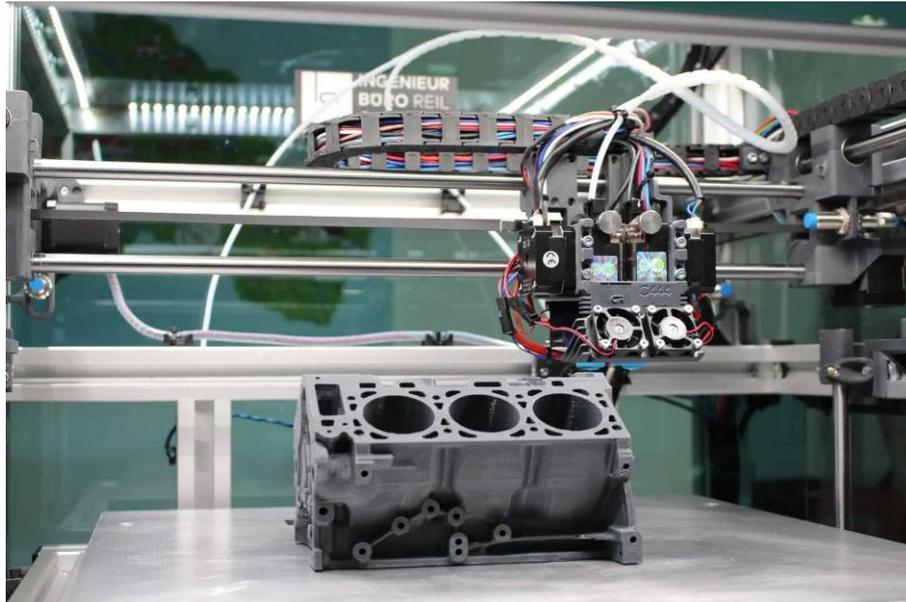
Additive Manufacturing adalah proses manufaktur di mana objek dibentuk dengan cara menambahkan material secara bertahap (lapis demi lapis), berbeda dengan metode tradisional yang mengurangi material dari blok padat. Proses ini berbasis pada model digital dari *software* CAD, yang kemudian dikonversi menjadi bentuk fisik melalui printer 3D. Kelebihan utama

AM adalah efisiensi material dan kemampuan mencetak bentuk internal rumit yang sulit atau bahkan tidak mungkin dibuat dengan metode konvensional.

Terdapat beberapa metode dalam *3D Printing*, yang dibedakan berdasarkan cara material ditambahkan. Beberapa metode umum adalah Fused Deposition Modeling (FDM), Selective Laser Sintering (SLS), dan Stereolithography (SLA). FDM adalah metode yang paling banyak digunakan karena sederhana dan murah, cocok untuk prototipe dan produksi skala kecil. SLS menggunakan sinar laser untuk mengikat partikel serbuk menjadi bentuk padat, biasanya digunakan untuk logam dan plastik yang memerlukan detail tinggi. SLA menggunakan sinar cahaya untuk mengeringkan resin cair, menghasilkan cetakan yang sangat presisi. Masing-masing metode memiliki kelebihan dan kekurangan tergantung kebutuhan material, ketelitian, dan skala produksi.

Pemilihan bahan sangat penting dalam *Additive Manufacturing* karena memengaruhi kekuatan, elastisitas, daya tahan panas, dan bahkan kompatibilitas biologis produk akhir. Bahan yang digunakan meliputi termoplastik, logam, keramik, hingga bahan biologis. Pemilihan material harus disesuaikan dengan tujuan penggunaan produk. Misalnya, untuk aplikasi dirgantara dibutuhkan material yang ringan tapi tahan terhadap suhu ekstrem (Radhika et al., 2024; Arif et al., 2023).

Teknologi AM kini digunakan di berbagai sektor seperti dirgantara, kesehatan, otomotif seperti pada Gambar 2. 16, dan produk konsumen. Di bidang dirgantara, AM membantu menciptakan komponen ringan yang efisien bahan bakar. Dalam bidang medis, *3D Printing* digunakan untuk membuat implan pasien secara spesifik dan alat bantu operasi yang disesuaikan, meningkatkan hasil perawatan pasien (Haleem et al., 2018; Wong, 2016). Di bidang industri, kecepatan prototipe dan fleksibilitas desain menjadikan AM solusi ideal untuk customization dan inovasi produk (Salmi, 2021; Gleadall et al., 2018).

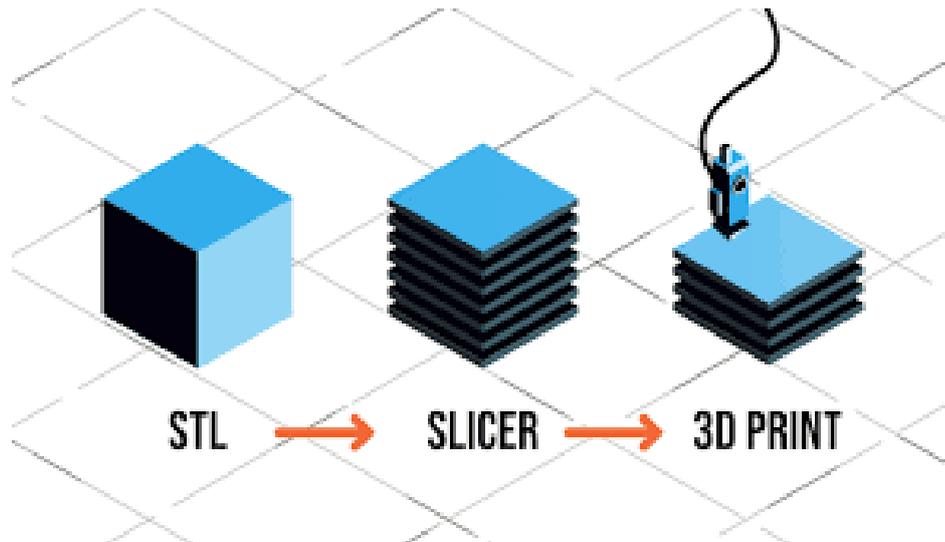


Gambar 2. 16 Contoh 3D print skala industri
(sumber: www.xometrypro.com)

Pada dasarnya, prinsip kerja 3D print dapat dijabarkan sebagai berikut:

1. Pemodelan *3D Printing* dengan format sesuai dengan aplikasi yang digunakan dalam pemodelan seperti *Solidworks*, *Inventor*, *Blender*, ataupun *Maxsurf*
2. Model diubah menjadi format *Standard Tessellation Language (STL)*, yang mengubah desain menjadi mesh berupa segitiga-segitiga kecil. Selain format STL, format AMF juga dapat digunakan agar desain dapat di print dengan 3D Print.
3. File STL kemudian diolah untuk menghasilkan lapisan-lapisan (*slices*) yang akan digunakan printer.
4. Material ditambahkan secara bertahap sesuai dengan lapisan yang dihasilkan dari proses slicing seperti pada Gambar 2. 16.
5. Model telah selesai lalu dilakukan pembersihan, penghalusan permukaan, dan pengecatan jika diperlukan.

Salah satu prinsip kerja terpenting dalam *3D Printing* yaitu *Slicing*. *Slicing* adalah proses kritis dalam *Additive Manufacturing* yang mengonversi model 3D menjadi instruksi yang dapat dibaca oleh printer (Gan, 2017). Proses pada Gambar 2. 17 dilakukan oleh perangkat lunak slicing seperti Cura, Simplify3D, atau PrusaSlicer.



Gambar 2. 17 Proses dari STL menuju file siap print.

Dalam proses *Slicing*, beberapa langkah yang dilakukan yaitu:

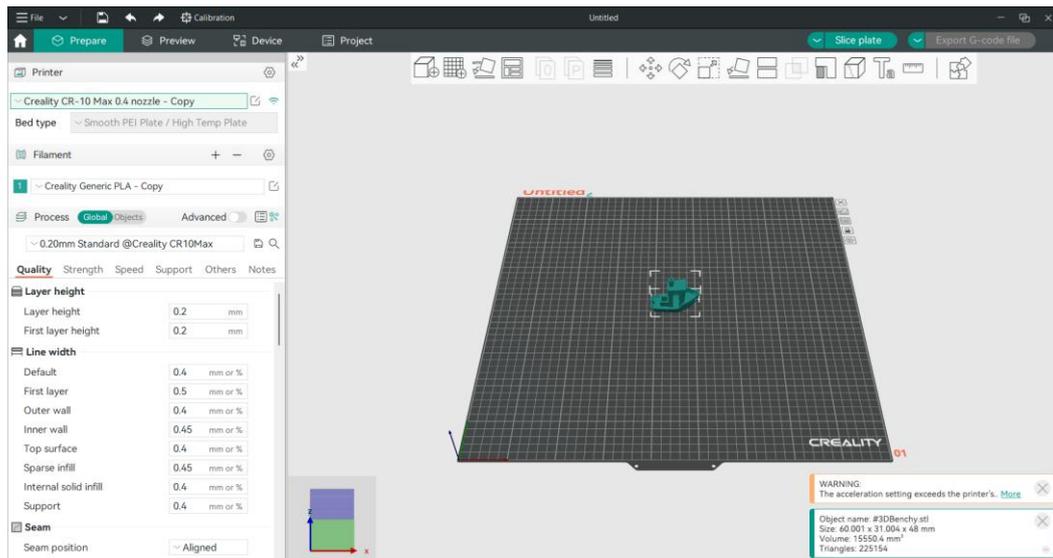
1. File model 3D dalam format STL diunggah ke *software* slicing.
2. Dilakukan pengaturan parameter-parameter yang ada dalam perangkat lunak *slicing*.

Beberapa parameter terpenting yaitu;

- a. Ketinggian lapisan
 - b. Kecepatan cetak
 - c. Isian dalam (*infill*)
 - d. Struktur penyangga
 - e. Temperatur lantai dan *nozzle*
3. Dilakukan pengaturan jalur ekstrusi material pada setiap lapisan
 4. Pembuatan kode G-Code yang nantinya akan memberikan perintah kepada *3D Printer*.

2.1.11.Slicer dan Prusa Slicer

Slicer merupakan perangkat lunak yang bertugas untuk mengonversi model 3D (biasanya dalam format STL, 3MF, atau OBJ) menjadi instruksi yang dapat dipahami oleh printer 3D. File gcode ini berisi perintah lapis demi lapis (*layer-by-layer*) mengenai gerakan nozzle, temperatur, kecepatan ekstrusi, dan parameter lain yang diperlukan dalam proses pencetakan.



Gambar 2. 18 Contoh slicer, Prusa Slicer

Slicer bekerja dengan memotong model 3D menjadi sejumlah irisan horizontal (*slice*) dan menghasilkan jalur cetak untuk setiap layer berdasarkan parameter yang ditentukan pengguna, seperti tinggi layer (*layer height*), pola pengisian (*infill*), kecepatan cetak, serta penambahan struktur penyangga (*support structure*). Oleh karena itu, proses slicing berpengaruh langsung terhadap kualitas hasil cetak, kekuatan mekanis model, serta efisiensi waktu dan konsumsi material.

Salah satu perangkat lunak slicer yang populer di kalangan pengguna printer FDM adalah PrusaSlicer pada Gambar 2. 18, yang dikembangkan oleh Prusa Research. PrusaSlicer merupakan proyek sumber terbuka yang berasal dari fork Slic3r dan telah mengalami banyak pengembangan, menjadikannya salah satu slicer paling lengkap dan fleksibel saat ini. PrusaSlicer mendukung berbagai jenis printer, material, dan strategi slicing, serta menyediakan antarmuka grafis (GUI) yang ramah pengguna dan antarmuka baris perintah (CLI) untuk automasi batch.

2.1.12. G-CODE dalam *Computer-Aided Manufacturing*

G-code, yang juga dikenal sebagai RS-274, merupakan bahasa pemrograman yang digunakan untuk mengendalikan mesin Computer Numerical Control (CNC) dan printer 3D. Bahasa ini terdiri dari serangkaian perintah yang menginstruksikan mesin mengenai arah pergerakan, alat yang digunakan, serta operasi pemesinan yang harus dilakukan berdasarkan spesifikasi desain yang ditentukan dalam model *Computer-Aided Design* (CAD). Sebagai bahasa utama dalam proses pemesinan CNC, G-code memainkan peran penting dalam otomatisasi proses manufaktur, memungkinkan produksi komponen kompleks secara konsisten

di berbagai sektor industri seperti dirgantara, otomotif, dan perkapalan (Rahman et al., 2023; Nießen & Nancarrow, 2019).

Struktur berkas G-code umumnya terdiri dari perintah-perintah berbasis kode alfanumerik seperti pada Gambar 2. 19 . Misalnya, huruf "G" menunjukkan perintah umum (general command), sementara huruf "M" merujuk pada fungsi-fungsi spesifik mesin (machine-specific commands). Setiap perintah dapat disertai dengan parameter numerik yang menjelaskan detail operasi, seperti kecepatan pemakanan (feed rate), arah gerakan alat, dan kecepatan putar spindle. Contoh perintah sederhana G-code adalah: G01 X0 Y0 F20, yang menginstruksikan mesin untuk bergerak secara linear menuju koordinat (X0, Y0) dengan kecepatan pemakanan 20 unit per menit (Rahman et al., 2022).

```
gcode

%
O1000 (Program for Milling a Circle)
G21      ; Set units to millimeters
G90      ; Absolute positioning
M06 T01  ; Select tool 1
M03 S1000 ; Start spindle at 1000 RPM (clockwise)
G00 X0 Y0 Z5 ; Rapid move to starting position (X0, Y0, Z5)
G01 Z-2 F100 ; Move tool down to Z-2 (cut depth) at 100 mm/min feed rate
G02 X20 Y0 I10 J0 F150 ; Mill a clockwise circular arc to X20, Y0, with center at (I10, J0)
G00 Z5      ; Retract tool to safe height
M05        ; Stop spindle
M09        ; Turn off coolant
M30        ; End of program
%
```

@boyiprototyping

Gambar 2. 19 isi file G-code *CNC Milling*
(sumber: www.boyiprototyping.com)

Dalam konteks manufaktur, pembuatan G-code biasanya difasilitasi melalui perangkat lunak tertentu. Model CAD yang menggambarkan desain komponen harus dikonversi ke dalam format yang kompatibel dengan mesin CNC. Proses ini umumnya dilakukan menggunakan perangkat lunak *Computer-Aided Manufacturing (CAM)*, yang menerjemahkan model CAD menjadi instruksi G-code yang sesuai. Transformasi ini sangat penting untuk memastikan bahwa operasi pemesinan selaras dengan fitur desain, toleransi, dan karakteristik material yang telah ditentukan (Nguyen et al., 2020; Kanada, 2016).

```

;Layer count: 25
;LAYER:0
M107
G0 F9000 X52.235 Y55.800 Z0.300
;TYPE:SKIRT
G1 F2340 X56.093 Y55.800 E0.18815
G1 X56.346 Y55.605 E0.20373
G1 X57.299 Y55.078 E0.25684
G1 X58.540 Y54.758 E0.31934
G1 X59.404 Y54.719 E0.36152
G1 X60.320 Y53.688 E0.42878

```

Gambar 2. 20 Isi file G-code proses 3D printing (sumber: www.researchgate.net)

G-code pada Gambar 2. 20, berfungsi sebagai bahasa dasar dalam pengoperasian mesin CNC (Computer Numerical Control) dan manufaktur aditif (pencetakan 3D), memainkan peran penting dalam mengontrol mesin masing-masing. Meskipun terdapat perbedaan signifikan dalam aplikasi dan set perintahnya, beberapa persamaan mendasar juga ditemukan dalam struktur, sintaksis, serta alur kerja antara G-code untuk CNC dan G-code untuk 3D Printing:

1. Struktur Perintah dan Sintaksis

Baik G-code pada CNC maupun 3D Printing menggunakan struktur perintah dan sintaks yang serupa, yang umumnya ditandai dengan kode alfanumerik. Perintah yang lazim digunakan dalam kedua konteks ini sering diawali dengan huruf “G” (untuk general command) atau “M” (untuk machine-specific command). Misalnya, perintah G01 umumnya menunjukkan interpolasi linier dalam operasi CNC maupun 3D Printing (Othman et al., 2021; John et al., 2023). Konsistensi sintaks ini memudahkan operator yang sudah mahir dalam satu jenis G-code untuk memahami sistem lain, sehingga mendorong pemahaman lintas aplikasi dalam pengendalian mesin (Rahman et al., 2023; Rahman et al., 2022).

2. Fungsi Dasar dan Parameter Operasional

Pada kedua proses, G-code memiliki fungsi dasar untuk mengarahkan gerakan mesin, menentukan koordinat lintasan alat, serta menjalankan berbagai perintah operasional. Sebagai contoh, perintah G0 dan G1 digunakan baik dalam CNC maupun pencetakan 3D untuk mengontrol gerakan cepat dan gerakan linier (Othman et al., 2021; Daminabo et al., 2020). Dalam pencetakan 3D, perintah-perintah tersebut mengatur pergerakan nozzle, sebagaimana dalam CNC mereka mengatur lintasan alat potong. Persamaan ini mencerminkan kesamaan mendasar

dalam perencanaan lintasan alat (toolpath planning) di kedua proses (Othman et al., 2021; John et al., 2023).

3. Pra-pemrosesan dan Slicing

Kesamaan penting lainnya terletak pada kebutuhan akan pra-pemrosesan melalui perangkat lunak slicer atau *Computer-Aided Manufacturing* (CAM). Kedua proses memerlukan model digital yang dikonversi menjadi serangkaian instruksi yang dapat dipahami mesin. Pada pencetakan 3D, file STL dikonversi menjadi G-code menggunakan perangkat lunak slicer, sedangkan pada CNC, CAM digunakan untuk mengubah model CAD menjadi G-code yang siap digunakan untuk pemesinan (Rahman et al., 2023; Daminabo et al., 2020). Proses serupa ini memastikan bahwa gerakan mesin ditentukan secara presisi demi menghasilkan produk akhir (Clifton et al., 2020; Siqueira et al., 2017).

4. Pengendalian Lintasan dan Pelapisan

Selain struktur perintah yang serupa, kedua jenis G-code juga melibatkan pengendalian lintasan, meskipun dalam konteks fisik yang berbeda. CNC machining menjalankan perintah untuk menggerakkan alat potong mengikuti lintasan yang ditentukan untuk menghilangkan material, sedangkan dalam *3D Printing*, G-code membentuk lintasan untuk menambahkan material secara berlapis. Meskipun pendekatannya berbeda, prinsip dasarnya tetap sama: kedua proses bertujuan untuk membentuk struktur tiga dimensi melalui jalur geometris yang telah ditentukan dalam G-code (Damon et al., 2020; Liu et al., 2019).

5. Fleksibilitas dan Kustomisasi

Terakhir, kedua format G-code memberikan ruang untuk kustomisasi dan optimasi berdasarkan kemampuan mesin dan kebutuhan operasional tertentu. Misalnya, pada CNC pengguna dapat mengatur kecepatan pemotongan dan kedalaman pemakanan, sementara dalam *3D Printing* pengguna dapat menyesuaikan tinggi lapisan atau laju ekstrusi. Kemampuan untuk menyempurnakan parameter ini sangat penting untuk menyesuaikan output dengan karakteristik material atau hasil yang diinginkan (Siqueira et al., 2017; Ramadhan et al., 2022).

2.2. Tinjauan Pustaka

2.2.1. Penggunaan *PythonOCC*

Penggunaan *PythonOCC*, yaitu *binding* bahasa *Python* dari Open CASCADE Technology (OCCT), semakin luas dieksplorasi di berbagai bidang, khususnya dalam pemodelan kompleks dan desain berbantuan komputer (CAD). Tinjauan pustaka ini merangkum berbagai kontribusi terkini yang menyoroti pemanfaatan *PythonOCC* dan OCCT dalam bidang rekayasa teknik, terutama pada aerodinamika, optimasi bentuk, dan pengembangan prototipe.

Salah satu aplikasi *PythonOCC* yang menonjol ditunjukkan dalam bidang teknik kedirgantaraan, khususnya pada pemodelan sayap morfing untuk *micro air vehicle* (MAV). Rodríguez-Sevillano et al. menunjukkan bahwa *PythonOCC* memungkinkan pembangkitan model parametrik tiga dimensi yang kompleks, yang mampu mengakomodasi berbagai geometri sayap melalui integrasi variabel aktuator. Fleksibilitas ini penting karena memungkinkan representasi bentuk sayap yang konvensional maupun non-standar, dengan tetap menjaga kualitas permukaan yang tinggi (Rodríguez-Sevillano et al., 2023). Kemampuan *PythonOCC* dalam memanipulasi konfigurasi geometri yang kompleks secara presisi menegaskan potensinya dalam aplikasi teknik kedirgantaraan tingkat lanjut.

Dalam konteks yang terkait, Auriemma membahas peran kernel CAD berdiferensiasi dalam proses optimasi bentuk aerodinamis berbasis gradien. Studi tersebut menggambarkan bagaimana OCCT dapat diadaptasi menggunakan pendekatan *Algorithmic Differentiation* (AD) untuk meningkatkan efisiensi dan akurasi dalam rutinitas optimasi komponen aerodinamis. Versi OCCT yang telah didiferensiasi ini terbukti instrumental dalam menyempurnakan proses desain, dan menegaskan pentingnya OCCT dalam dinamika fluida komputasional serta analisis aerodinamika (Banović et al., 2018).

Selain itu, fleksibilitas umum *PythonOCC* juga tercermin dalam perannya untuk menghasilkan dataset bagi aplikasi pembelajaran mesin. Sebagai contoh, Ma & Yang memanfaatkan *PythonOCC* untuk membuat dataset model CAD 3D dari komponen lembaran logam, yang digunakan untuk mengembangkan fitur pengenalan adaptif dalam proses manufaktur. Ini menunjukkan bahwa *PythonOCC* tidak hanya relevan dalam penggunaan CAD tradisional, tetapi juga dalam mendukung proses berbasis kecerdasan buatan, khususnya dalam interpretasi fitur geometris yang berkaitan dengan proses permesinan (Ma & Yang, 2024).

Lebih lanjut, eksplorasi integrasi OCCT dengan metode komputasi menunjukkan manfaat besar dalam mengoptimalkan alur kerja desain. Banović et al. menjelaskan integrasi

OCCT dengan solver CFD adjoint, yang memungkinkan optimasi bentuk berbasis gradien secara efektif. Studi ini menampilkan kapabilitas OCCT yang kuat dalam menghasilkan representasi geometri presisi tinggi yang penting untuk simulasi teknik lanjutan. Pendekatan ini menunjukkan transformasi mendasar dalam menghubungkan proses desain CAD dengan simulasi, dan menekankan efisiensi operasional dalam alur kerja teknik (Banović et al., 2018).

Secara keseluruhan, literatur yang ada menegaskan bahwa *PythonOCC* dan OCCT memiliki dampak signifikan di berbagai bidang. Aplikasinya mencakup desain inovatif dalam teknik kedirgantaraan, pembuatan dataset untuk kecerdasan buatan, hingga peningkatan pendekatan edukatif berbasis teknologi. Dengan perkembangan yang terus berlanjut dari OCCT dan binding *Python*-nya, *PythonOCC* menjanjikan kemajuan yang lebih lanjut dalam ranah CAD dan optimasi desain teknik.

2.2.2. Additive Manufacturing untuk Model Teknik

Salah satu aspek utama dari AM adalah kemampuannya untuk memfasilitasi pembuatan geometri kompleks yang tidak dapat dicapai melalui metode manufaktur konvensional. Jiang et al. menekankan fleksibilitas AM yang hadir dalam berbagai bentuk, seperti rapid prototyping dan layer manufacturing, yang memungkinkan fabrikasi komponen rumit secara efisien (Zhang et al., 2020). Kebebasan dalam mendesain ini tidak hanya meningkatkan kreativitas, tetapi juga menuntut perubahan dalam prinsip-prinsip rekayasa tradisional. Watschke et al. menunjukkan bahwa kapabilitas unik AM telah mendefinisikan ulang cara desain produk dikonseptualisasikan dan dieksekusi, mendorong batas-batas inovasi dalam desain teknik (Watschke et al., 2019).

Lebih lanjut, integrasi pendekatan Design for *Additive Manufacturing* (DfAM) semakin penting dalam mengoptimalkan proses produksi. Malviya et al. menjelaskan bahwa DfAM berfokus pada keterkaitan kompleks antara bentuk desain, sifat material, dan struktur geometri untuk memaksimalkan keunggulan AM. Pendekatan inovatif ini telah diadopsi oleh organisasi terkemuka seperti NASA dan General Electric untuk menyederhanakan proses manufaktur dan mengurangi berat komponen tanpa mengorbankan fungsi (Malviya et al., 2019).

Pemilihan material dalam AM juga menjadi faktor krusial, terutama untuk aplikasi di sektor berkinerja tinggi seperti dirgantara dan rekayasa biomedis. Ferro et al. mengeksplorasi manfaat AM dalam pengembangan struktur kisi ringan di industri dirgantara yang dapat meningkatkan ketahanan terhadap kelelahan (Ferro et al., 2023). Kemajuan ini dilengkapi oleh riset dalam material berkelanjutan, seperti yang ditunjukkan oleh Jeffri et al., yang meneliti

potensi biopolimer seperti polyhydroxyalkanoates untuk digunakan dalam proses AM, mendukung tren menuju produksi yang lebih ramah lingkungan (Jeffri et al., 2022). Isu dampak lingkungan AM menjadi tema yang berulang dalam literatur, seperti yang terlihat dalam studi Booyesen et al. mengenai pencetakan implan khusus, yang menyoroti rendahnya limbah material dibandingkan metode konvensional (Booyesen et al., 2019).

Optimalisasi proses dan material dalam AM juga menjadi area kajian yang semakin berkembang. Dvořák et al. memberikan wawasan tentang tantangan optimasi kompleks dalam AM, dengan menekankan perlunya pendekatan lintas disiplin untuk memaksimalkan efisiensi produksi dan meminimalkan biaya. Strategi optimasi ini menjadi sangat penting untuk mengatasi tantangan bawaan dari AM, seperti keterbatasan material dan kualitas manufaktur (Dvořák et al., 2018). Metzger et al. mencatat pentingnya desain reaktor berbasis AM yang menunjukkan bagaimana peralatan proses dapat dioptimalkan menggunakan teknologi AM, mendorong kemajuan dalam rekayasa proses kimia (Metzger et al., 2023).

Sebagai kesimpulan, studi-studi yang telah dibahas secara kolektif menegaskan manfaat multifaset dari *Additive Manufacturing* dalam desain dan aplikasi teknik. Mulai dari mendorong prinsip desain inovatif melalui DfAM hingga menghadapi tantangan keberlanjutan dan optimasi, AM merepresentasikan lompatan besar dalam teknologi manufaktur. Dengan demikian, riset dan pengembangan yang terus berlanjut dalam bidang ini membuka jalan bagi kemajuan masa depan di berbagai sektor teknik.

2.2.3. Otomatisasi dalam Proses Desain dan Produksi

Integrasi otomatisasi dalam proses desain dan produksi telah merevolusi berbagai sektor industri dengan meningkatkan produktivitas, efisiensi, dan daya saing global. Sejumlah studi terkini menunjukkan bahwa otomatisasi memainkan peran penting dalam mengoptimalkan kinerja operasional serta meminimalkan ketergantungan terhadap intervensi manual.

Dalam konteks produktivitas, Bhargavi et al. menunjukkan bahwa keberhasilan penerapan otomatisasi dalam pengujian sistem dapat meningkatkan produktivitas operasional secara signifikan (Bhargavi et al., 2024). Hal ini sejalan dengan temuan Saha yang menyatakan bahwa otomatisasi memungkinkan mekanisasi tugas-tugas berulang, sehingga mengoptimalkan efisiensi dan menurunkan biaya produksi (Saha, 2023). Alharbi turut menekankan bahwa otomatisasi industri tidak hanya berfungsi sebagai strategi efisiensi biaya, tetapi juga sebagai elemen peningkat kualitas produksi dan daya saing di pasar global (Alharbi, 2024).

Salah satu implementasi paling nyata dari otomatisasi dalam desain adalah penggunaan sistem *Computer-Aided Design* (CAD). Zahcrl menyebut bahwa penerapan CAD memungkinkan perancangan urutan perakitan secara efisien melalui representasi 3D, yang selanjutnya memfasilitasi proses produksi otomatis. Pendekatan ini tidak hanya mempercepat tahapan desain produk, tetapi juga meminimalkan kesalahan manusia dan meningkatkan kesiapan produk untuk diproduksi massal (Zahcrl, 2022). Autor juga menyoroti bahwa otomatisasi berbasis CAD berkontribusi terhadap pengurangan waktu henti produksi dan tingkat kesalahan, menjadikannya fondasi penting dalam pengembangan lini produksi otomatis yang presisi (Autor, 2015).

Namun demikian, transisi menuju sistem otomatisasi juga menghadirkan tantangan tersendiri. Wang et al. menyoroti bahwa otomatisasi memerlukan penyesuaian signifikan dalam keterampilan tenaga kerja dan dapat memicu kekhawatiran terhadap pengurangan lapangan kerja di sektor manufaktur tradisional (G. Wang et al., 2024). Hal ini juga disoroti oleh Luo, yang mengungkapkan perlunya adaptasi sumber daya manusia dalam merespons penerapan lini produksi otomatis (Luo, 2024). Dalam sektor kesehatan, Stoumpos et al. dan Ahtiainen et al. menunjukkan bahwa otomatisasi menghadirkan tantangan unik dalam menjaga keselamatan pasien dan efisiensi proses, terutama di tengah percepatan pembaruan teknologi yang sensitif terhadap kesalahan (Ahtiainen et al., 2019; Stoumpos et al., 2023).

Melihat ke depan, adopsi teknologi otomatisasi cerdas seperti Artificial Intelligence (AI) dan Robotic Process Automation (RPA) diprediksi akan semakin meningkat. Yoqubjonov et al. menunjukkan bahwa penerapan RPA mampu menyederhanakan proses bisnis dan meningkatkan efisiensi secara menyeluruh, menggambarkan arah tren global menuju solusi otomatisasi berbasis kecerdasan yang lebih terintegrasi dan adaptif (Yoqubjonov et al., 2023).

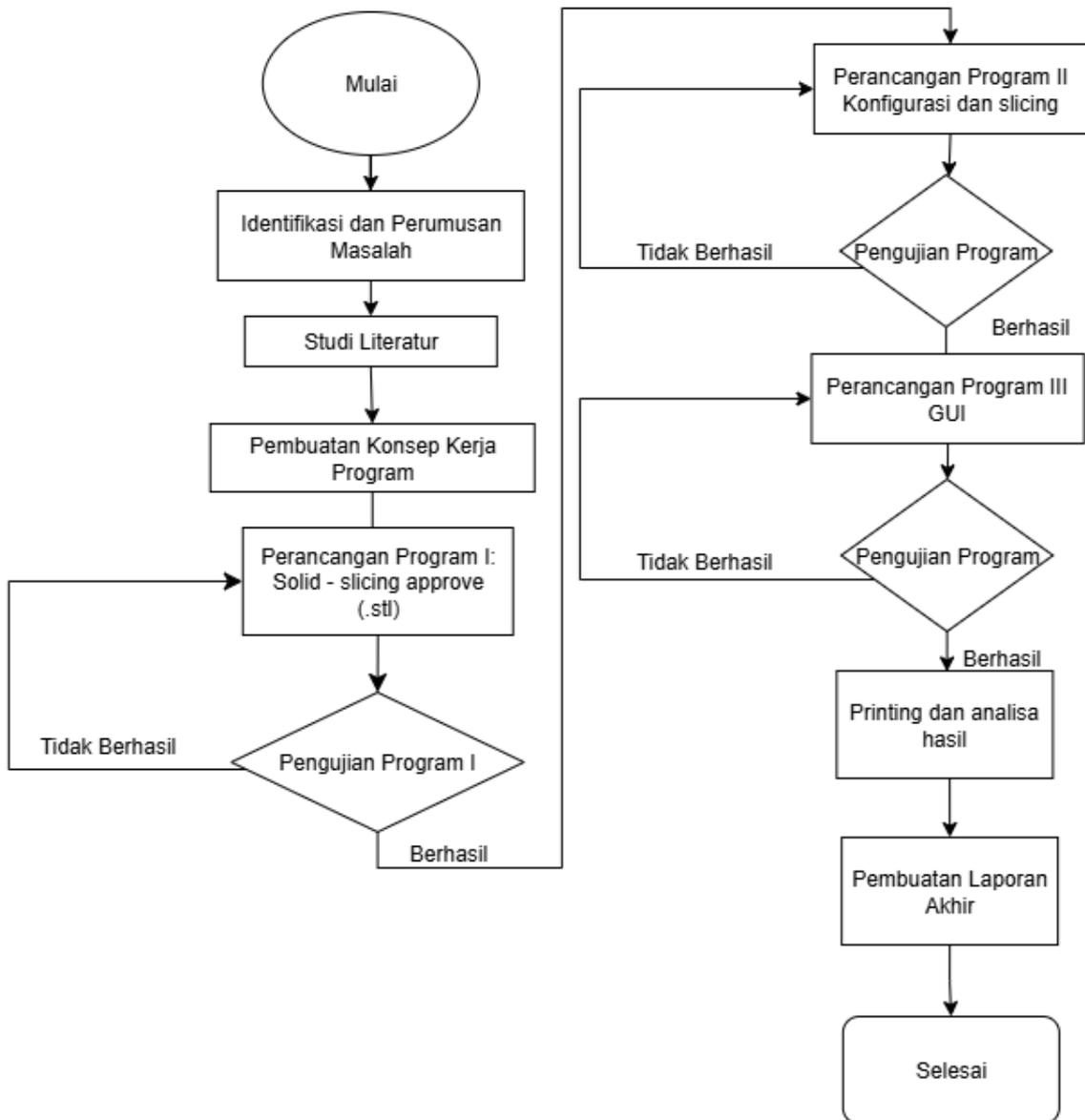
Secara keseluruhan, literatur menunjukkan bahwa otomatisasi dalam proses desain dan produksi merupakan katalisator utama untuk pencapaian produktivitas dan efisiensi operasional yang belum pernah terjadi sebelumnya. Meskipun demikian, implementasinya harus mempertimbangkan dampak terhadap tenaga kerja dan tantangan spesifik pada masing-masing sektor agar potensi penuh dari teknologi ini dapat dimanfaatkan tanpa mengorbankan kualitas dan keselamatan operasional.

Halaman ini sengaja dikosongkan

BAB 3 METODOLOGI

3.1. Bagan Alir

Secara umum metodologi dalam pengerjaan Tugas Akhir ini dapat dijabarkan dalam Gambar 3. 1



Gambar 3. 1 Bagan Alir Pengerjaan Tugas Akhir

3.2. Tahap Pengerjaan

3.2.1. Identifikasi dan Perumusan Masalah

Tahap pertama yang dilakukan dalam penyusunan tugas akhir ini adalah mengidentifikasi permasalahan nyata di lapangan, khususnya dalam proses pembuatan prototipe lambung kapal skala model. Selama ini, proses tersebut memerlukan tahapan yang panjang, banyak ketergantungan antar *software*, serta intervensi manual yang tinggi, mulai dari pemodelan, konversi format, hingga pencetakan. Permasalahan tersebut kemudian dirumuskan menjadi fokus tugas akhir, yaitu mengembangkan sistem otomatisasi berbasis *Python* untuk menghasilkan model lambung kapal siap cetak dengan alur yang terintegrasi, efisien, dan fleksibel terhadap perubahan parameter desain.

3.2.2. Studi Literatur

Untuk menunjang perancangan sistem, dilakukan studi literatur dari berbagai sumber, baik berupa jurnal ilmiah, standar teknis, dokumentasi perangkat lunak, maupun tugas akhir terdahulu. Literatur yang dikaji mencakup pemodelan lambung kapal, parameterisasi bentuk geometris, teknik *Additive Manufacturing*, dan dokumentasi teknis dari pustaka *PythonOCC*, *PyQt5*, serta *PrusaSlicer*. Studi ini penting dalam menentukan pendekatan teknis, validasi metode, serta sebagai dasar penyusunan konsep program otomatisasi yang akan dibangun.

3.2.3. Pembuatan Konsep Kerja Program

Berdasarkan hasil studi literatur dan identifikasi kebutuhan pengguna, disusunlah konsep kerja sistem otomatisasi pembuatan model lambung kapal. Konsep ini terbagi menjadi tiga bagian utama, yaitu (1) konversi model permukaan menjadi solid yang siap diolah, (2) proses pemotongan blok dan penambahan sambungan, dan (3) integrasi proses slicing dan antarmuka pengguna berbasis GUI. Setiap bagian dikembangkan sebagai *script* modular, sehingga bisa diuji secara terpisah maupun digabung dalam pipeline otomatis penuh.

3.2.4. Perancangan Program I: Solid – Slicing Approve (.STL)

Program pertama dikembangkan dengan tujuan untuk mengubah model permukaan menjadi model solid yang siap dicetak. Langkah-langkah yang dilakukan mencakup pembacaan file *STEP/IGES*, proses penyatuan permukaan (*sewing*), konversi menjadi solid (*solidify*), dan penerapan skala. Selanjutnya, dilakukan proses *hollowing* menggunakan pendekatan *voxel grid* untuk mengurangi massa material cetak. Model kemudian dipotong menjadi beberapa blok

sepanjang sumbu X, dan ditambahkan fitur sambungan berupa lubang baut M3. Setiap blok diekspor ke dalam format .stl dan .step untuk keperluan slicing dan dokumentasi teknis.

3.2.5. Pengujian Program I

Pengujian dilakukan untuk memastikan bahwa model solid hasil konversi memiliki dimensi yang benar, memiliki ketebalan yang cukup setelah hollowing, dan bahwa file STL yang dihasilkan dapat dibuka di slicer tanpa error. Selain itu, dipastikan bahwa pemotongan blok tidak menyebabkan kehilangan geometri penting dan sambungan antar blok berada di posisi yang tepat. Validasi dilakukan melalui visualisasi hasil ekspor dan pengecekan hasil slicing awal secara manual.

3.2.6. Perancangan Program II: Konfigurasi dan Slicing

Program kedua dikembangkan untuk mengotomatisasi proses slicing file STL hasil program pertama. Fitur utama dari program ini mencakup rotasi model agar posisinya optimal saat dicetak, pemanggilan PrusaSlicer melalui perintah CLI dengan konfigurasi .ini, serta pembuatan render gambar hasil model untuk dokumentasi dan validasi visual. Folder output disusun rapi dalam struktur direktori rotated/ dan hasil/ untuk menyimpan file .gcode dan .png.

3.2.7. Pengujian Program II

Program slicing diuji dengan berbagai file STL hasil pemotongan. Validasi dilakukan dengan mengecek bahwa model memiliki orientasi yang benar pada bed printer, hasil slicing sesuai dengan konfigurasi, dan file .gcode dapat digunakan langsung untuk proses pencetakan. Selain itu, hasil render PyVista diperiksa untuk memastikan bahwa bentuk geometri tidak mengalami deformasi akibat rotasi atau konversi format.

3.2.8. Perancangan Program III: GUI

Untuk menyederhanakan proses bagi pengguna akhir, dikembangkanlah antarmuka pengguna berbasis PyQt5. Antarmuka ini memungkinkan pengguna untuk memilih file model, memasukkan parameter skala, panjang blok, dan voxel size, serta memilih profil printer 3D yang digunakan. Saat tombol dijalankan, GUI akan secara otomatis memanggil program solidifikasi dan slicing secara berurutan, serta menampilkan log proses pada jendela khusus. Hal ini mempermudah pengguna yang tidak memiliki latar belakang pemrograman untuk menggunakan sistem.

3.2.9. Pengujian Program III

GUI diuji dengan berbagai kombinasi parameter untuk memastikan seluruh pipeline dapat dijalankan tanpa crash, dari pemilihan file input hingga hasil akhir berupa .gcode dan .png. Diuji juga validasi input pengguna, kesalahan input file, dan kestabilan GUI selama proses berlangsung. Keberhasilan program ditandai dengan tersedianya file hasil cetak dan render dalam waktu yang efisien dan tanpa error.

3.2.10. Printing dan Analisa Hasil

Setelah file .gcode berhasil dibuat, model dicetak menggunakan printer 3D. Hasil cetakan kemudian dianalisis dengan mengukur ulang panjang total, jumlah blok, dan posisi sambungan. Evaluasi dilakukan untuk membandingkan akurasi dimensi antara model digital dan hasil cetak fisik. Analisa ini menjadi validasi akhir terhadap efektivitas program yang telah dibuat.

3.2.11. Pembuatan Laporan Akhir

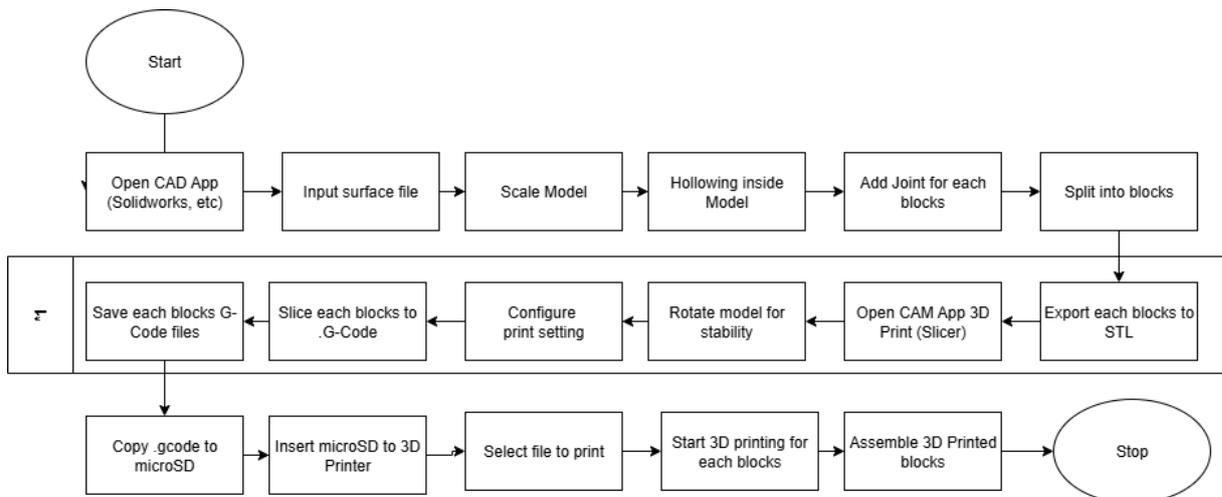
Setelah seluruh proses pengembangan dan pengujian selesai, dilakukan penyusunan laporan akhir sebagai dokumentasi lengkap dari proyek tugas akhir ini. Laporan mencakup penjelasan metodologi, penjabaran proses teknis, hasil dan evaluasi pengujian, serta kesimpulan dan saran pengembangan lebih lanjut. Laporan ini diharapkan dapat menjadi referensi dan dokumentasi berharga untuk pengembangan sistem serupa di masa depan.

BAB 4

PERANCANGAN DAN IMPLEMENTASI PROGRAM

4.1. Metode Konvensional dalam Pembuatan Model Lambung 3D Print

Sebelum merancang program otomatisasi pemrosesan 3D printing, terlebih dahulu dilakukan kajian terhadap metode konvensional yang selama ini digunakan. Metode konvensional dalam pembuatan model lambung kapal untuk keperluan pencetakan 3D memiliki alur kerja yang panjang dan penuh dengan interaksi manual. Setiap tahapan, mulai dari pengolahan awal model permukaan di perangkat lunak CAD, pengaturan skala, proses hollowing untuk mengurangi massa model, penambahan fitur sambungan antarblok, hingga pembagian model menjadi beberapa bagian kecil, membutuhkan keterampilan desain 3D dan ketelitian yang tinggi.

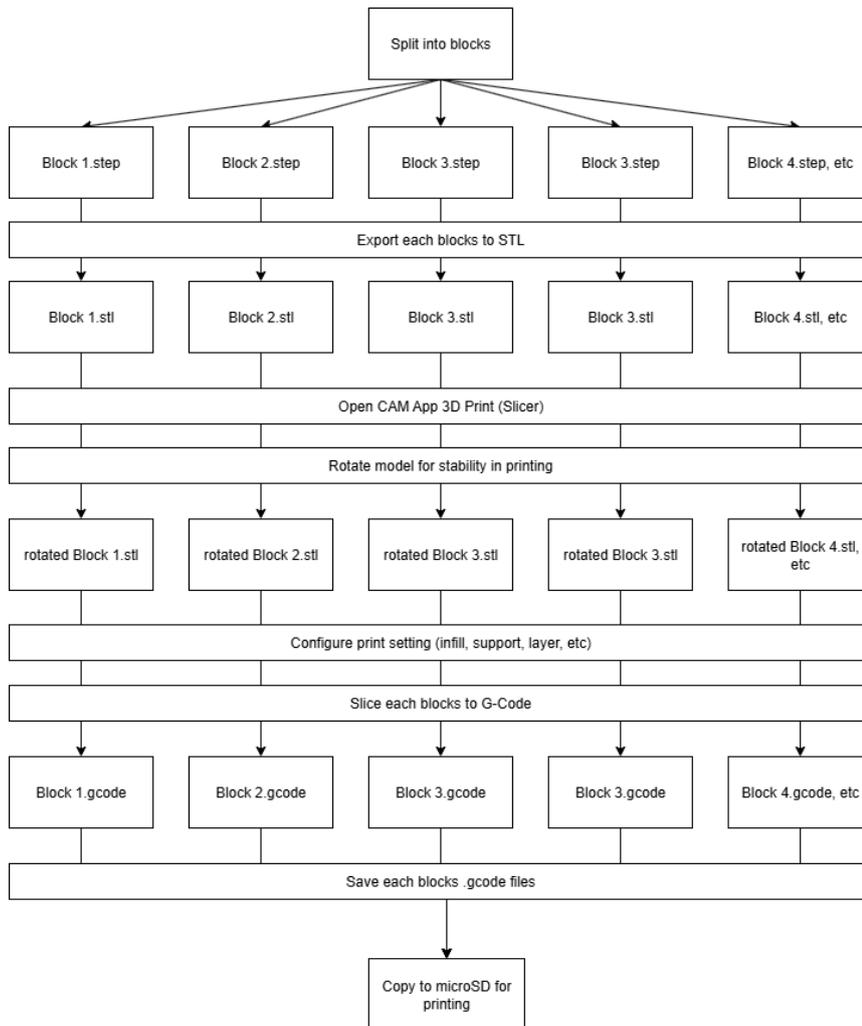


Gambar 4. 1 Alur pembuatan model lambung 3D print konvensional

Proses yang diilustrasikan pada Gambar 4. 1 dimulai dengan membuka file model permukaan lambung kapal menggunakan aplikasi CAD seperti SolidWorks atau perangkat lunak serupa. Pada tahap ini, model yang biasanya berupa permukaan (surface) harus diproses secara manual menjadi bentuk yang siap untuk pencetakan 3D.

Pada tahap scaling Model, model diperbesar atau diperkecil agar sesuai dengan ukuran printer 3D yang tersedia. Proses ini membutuhkan pengaturan parameter ukuran secara hati-hati karena akan memengaruhi dimensi hasil cetak. Selanjutnya dalam tahap hollowing model,

model harus dibuat berongga secara manual agar penggunaan material lebih hemat dan bobot model berkurang. Proses hollowing memerlukan pembuatan offset permukaan dan penghapusan volume bagian dalam, yang biasanya cukup rumit dan memerlukan keahlian CAD. Sambungan antarblok (seperti lubang baut atau sistem snap-fit) harus dirancang secara manual. Hal ini dilakukan satu per satu pada permukaan blok, memakan banyak waktu dan memerlukan ketelitian tinggi agar setiap sambungan dapat terpasang dengan baik.



Gambar 4. 2 Penjelasan lebih lanjut proses split block hingga menjadi file siap print

Kemudian dijabarkan lebih lanjut pada Gambar 4. 2, model kemudian dibagi menjadi beberapa blok agar sesuai dengan ukuran area cetak printer 3D. Setiap blok disimpan sebagai file STEP, kemudian diekspor kembali menjadi file STL satu per satu. Semakin banyak jumlah blok, semakin panjang proses ekspor yang harus dilakukan secara manual.

Setelah proses awal pemodelan meliputi pengaturan skala, pembuatan rongga (*hollowing*) untuk mengurangi bobot dan konsumsi material, serta penambahan fitur

sambungan antarblok selesai, tahapan berikutnya dalam metode konvensional adalah pembagian model menjadi beberapa blok.

Proses ini dilakukan karena ukuran volume cetak printer 3D umumnya terbatas, sehingga model lambung yang panjang harus dipisahkan menjadi beberapa bagian lebih kecil yang muat di area cetak. Dalam aplikasi CAD, proses *splitting* ini dilakukan dengan membuat bidang potong (*cutting planes*) secara manual pada posisi tertentu sesuai dengan panjang maksimum blok yang diinginkan. Setiap hasil potongan harus dicek kembali untuk memastikan tidak ada celah atau *overlap* yang akan mengganggu proses perakitan setelah pencetakan.

Setelah model terpisah menjadi beberapa blok, setiap blok disimpan dalam format STEP. Namun, proses tidak berhenti di sini karena format STEP yang berbasis permukaan/solids tidak langsung kompatibel dengan perangkat lunak *slicer*. Oleh karena itu, setiap file STEP harus diekspor kembali secara manual menjadi format STL yang lebih sesuai untuk proses slicing.

Jika model memiliki banyak blok (misalnya lebih dari 10 blok), pekerjaan ekspor ini menjadi repetitif dan memakan waktu lama. Setiap file harus:

1. Dipilih secara manual dari daftar blok.
2. Dibuka di aplikasi CAD atau konversi.
3. Ditetapkan opsi ekspor yang benar (misalnya toleransi *mesh* atau orientasi koordinat).
4. Disimpan ke folder yang sesuai dengan nama file unik.

Kesalahan kecil dalam tahap ini, seperti salah memberi nama file atau memilih opsi ekspor yang kurang tepat, dapat berdampak pada proses berikutnya di perangkat lunak *slicer*. Dengan kata lain, semakin banyak jumlah blok yang dihasilkan, semakin panjang waktu yang diperlukan dan semakin besar potensi *human error* yang muncul hanya pada tahapan ekspor file ini saja.

Setelah seluruh blok model berhasil diekspor dalam format STL, tahapan berikutnya dalam metode konvensional adalah slicing dan konfigurasi parameter pencetakan. Proses ini harus dilakukan secara individual untuk setiap blok, sehingga sangat memakan waktu terutama jika jumlah blok cukup banyak. Setiap file STL harus dibuka satu per satu dalam perangkat lunak *slicer* seperti PrusaSlicer. Model kemudian ditempatkan dan diputar agar berada pada orientasi terbaik untuk pencetakan. Orientasi yang salah dapat menyebabkan kebutuhan struktur penyangga (*support*) yang berlebihan, kualitas permukaan yang buruk, bahkan kegagalan cetak. Karena itu, penentuan orientasi memerlukan perhatian penuh dan pengalaman pengguna.

Proses ini dilakukan ulang untuk setiap blok, tanpa adanya proses otomatisasi yang mengatur orientasi secara langsung.

Setelah orientasi ditentukan, parameter pencetakan diatur secara manual. Pengguna harus menyesuaikan:

- *Infill* (kepadatan isi dalam model) agar seimbang antara kekuatan dan penggunaan material.
- *Support structure* untuk menopang bagian dengan sudut menggantung (*overhang*).
- Layer height (ketinggian lapisan) untuk mengontrol ketelitian permukaan.
- Print speed, bed temperature, dan *nozzle temperature* sesuai dengan material yang digunakan. Dan sebagainya.

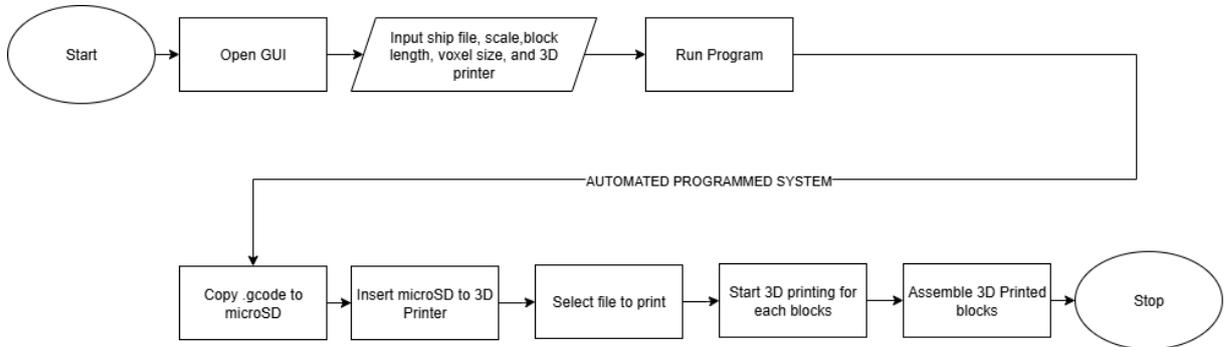
Penyesuaian ini dilakukan ulang untuk setiap blok, dan jika ada kesalahan pengaturan satu blok, maka kualitas hasil cetak akan berbeda dari blok lainnya. Tidak ada jaminan bahwa setiap blok memiliki konfigurasi parameter yang seragam karena semuanya bergantung pada ketelitian operator.

Setelah semua parameter diatur, pengguna menjalankan proses slicing untuk mengonversi model menjadi perintah G-code yang bisa dibaca oleh printer 3D. Proses ini menghasilkan satu file G-code untuk setiap blok. Jika sebuah model terbagi menjadi banyak blok, maka akan dihasilkan banyak file G-code yang harus dikelola secara manual. Selain itu, pengguna harus memastikan semua file G-code tersimpan dengan nama yang benar agar tidak tertukar. Jika ada perubahan desain, proses slicing harus diulang dari awal untuk setiap blok, yang semakin menambah waktu pengerjaan. File G-code yang dihasilkan disalin ke media microSD dan dijalankan pada printer 3D. Setiap blok dicetak satu per satu, kemudian seluruh blok hasil cetakan disatukan menjadi model lambung utuh. Proses ini membutuhkan ketelitian agar posisi setiap blok presisi dan sambungan antarblok terpasang dengan baik.

Metode konvensional ini memiliki kelemahan utama berupa banyaknya interaksi manual pada setiap tahapannya: hollowing dan pembuatan joint yang memerlukan keahlian desain, pembagian blok yang harus dilakukan satu per satu, serta proses slicing yang harus diulang-ulang untuk setiap blok. Akibatnya, waktu yang dibutuhkan menjadi sangat panjang dan risiko kesalahan manusia (*human error*) meningkat, baik dalam orientasi pencetakan, hingga pemilihan konfigurasi parameter printer saat mencetak. Metode ini jelas tidak efisien jika digunakan untuk proses yang berulang atau dengan jumlah blok yang besar.

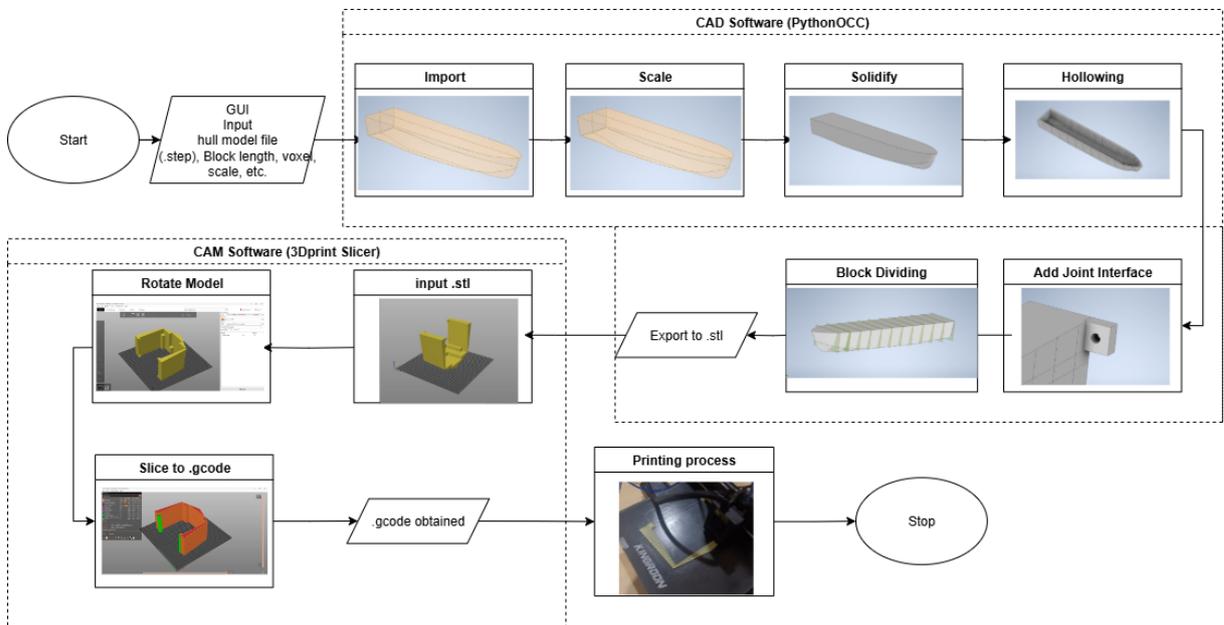
4.2. Perancangan Alur Otomatisasi Program

Setelah mengetahui bagaimana metode konvensional dari 3dprint bekerja, dilakukan perancangan akan bagaimana nantinya program otomatis akan bekerja.



Gambar 4. 3 Rancangan alur metode otomatisasi

Perancangan alur otomatisasi pada Gambar 4. 3 dilakukan untuk menjawab permasalahan yang ditemukan pada metode konvensional yang bersifat panjang, repetitif, dan membutuhkan banyak interaksi manual. Dengan pendekatan otomatisasi, setiap tahap yang sebelumnya harus dilakukan secara terpisah dapat diintegrasikan dalam satu sistem terpadu, sehingga proses dari model CAD permukaan hingga menjadi file siap cetak .gcode dapat berjalan lebih cepat, konsisten, dan minim kesalahan manusia.



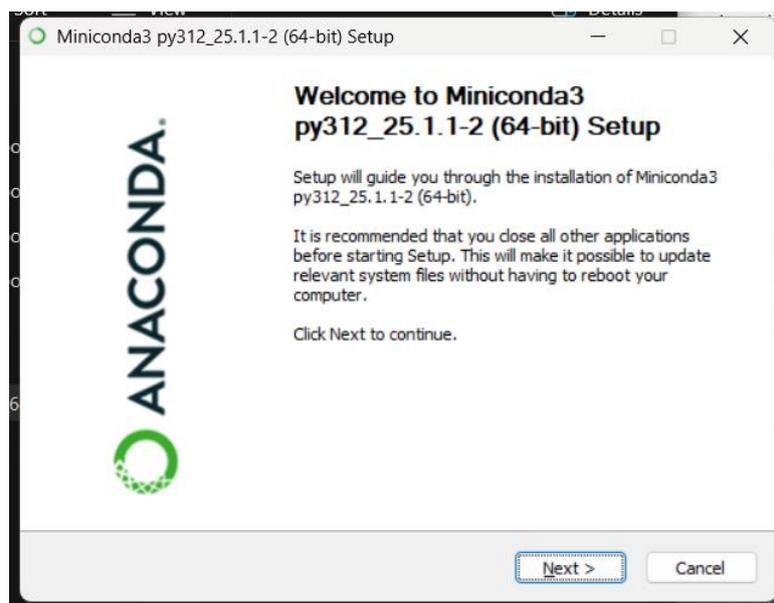
Gambar 4. 4 Rancang rinci alur program otomatisasi

Sistem pada Gambar 4. 4 ini dibangun untuk mengotomatiskan proses pembuatan prototip lambung kapal dengan menggunakan *3D Printer*, dimulai dari file CAD permukaan hingga menjadi file .gcode yang siap dicetak menggunakan printer 3D. Secara umum, sistem

ini terdiri dari dua tahapan utama, yaitu tahapan pemrosesan model menggunakan CAD *software* berbasis *PythonOCC*, dan tahapan penyusunan file siap cetak menggunakan CAM *software* slicer *3D Printing*. Untuk menyederhanakan interaksi pengguna, dikembangkan pula antarmuka grafis (GUI) berbasis PyQt5 yang mengintegrasikan seluruh tahapan ke dalam satu aplikasi terpadu.

4.3. Python dan Anaconda CMD

Sebelum menjalankan program otomatisasi pembuatan prototip lambung kapal, langkah awal yang dilakukan adalah menyiapkan lingkungan kerja berbasis *Python*. Untuk mempermudah manajemen dependensi dan menghindari konflik antar pustaka, digunakan distribusi *Python* Anaconda yang dilengkapi dengan package manager *conda*. Instalasi Anaconda dilakukan dengan mengunduh installer dari situs resmi <https://www.anaconda.com> dan mengikuti prosedur instalasi standar seperti pada Gambar 4. 5.



Gambar 4. 5 Instalasi anaconda

Setelah Anaconda terpasang, dibuat sebuah environment baru yang terpisah agar lebih terkontrol, misalnya dengan nama *kapalcadcam*, dan diaktifkan melalui perintah `conda create -n kapalcadcam` diikuti dengan `conda activate kapalcadcam`.

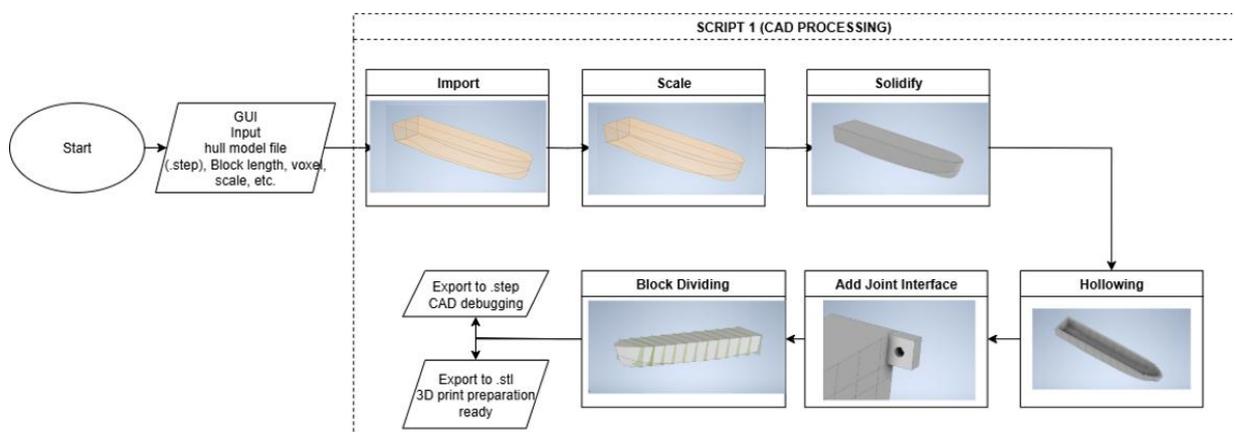
```
(base) C:\Users\User>conda create -n kapalcadcam|
```

Setelah environment aktif, dilakukan instalasi beberapa pustaka eksternal yang dibutuhkan oleh sistem. Di antaranya yaitu:

```
(base) C:\Users\User>pip install pythonocc-core
```

- *PythonOCC-core*: library utama untuk CAD modeling berbasis OpenCASCADE
- *pyqt5*: digunakan untuk GUI berbasis Qt
- *pyvista*: untuk visualisasi STL
- *joblib*: untuk parallel processing (digunakan pada proses voxel filter)
- *argparse* dan *subprocess*: library standar *Python* untuk CLI dan eksekusi program
- *matplotlib* (opsional): untuk rendering preview jika diperlukan

4.4. Program I



Gambar 4. 6 Alur Perancangan Program I

Subbab yang digambarkan pada Gambar 4. 6 mencakup seluruh tahapan pengolahan awal model lambung kapal, dimulai dari import file permukaan hingga menghasilkan file STL yang siap diproses lebih lanjut oleh slicer.

4.4.1. Identifikasi Library yang Digunakan

Langkah awal dalam proses pengembangan program adalah mengidentifikasi pustaka (library) yang relevan untuk mendukung fungsionalitas yang dibutuhkan. Program pertama ini bertujuan untuk memproses file permukaan model lambung kapal menjadi bentuk solid, melakukan transformasi skala, mengosongkan bagian dalam (hollowing), memotong model menjadi beberapa blok, menambahkan sambungan antarblok, serta menyiapkan file hasil dalam format .stl, .step, dan .gcode. Oleh karena itu, dipilihlah beberapa pustaka utama pada Gambar 4. 7.

```

import argparse
import sys
import os
import subprocess
import multiprocessing
from joblib import Parallel, delayed

from PyQt5.QtWidgets import QApplication, QDialog
from OCC.Display.backend import load_backend
load_backend("qt-pyqt5")
from OCC.Display.qtDisplay import qtViewer3d
from OCC.Extend.DataExchange import read_stl_file, write_step_file
from OCC.Core.STEPControl import STEPControl_Writer, STEPControl_AsIs, STEPControl_Reader
from OCC.Core.IGESControl import IGESControl_Reader
from OCC.Core.BRepTools import breptools_Write
from OCC.Core.BRepAlgoAPI import BRepAlgoAPI_Cut, BRepAlgoAPI_Common
from OCC.Core.BRepPrimAPI import BRepPrimAPI_MakeBox, BRepPrimAPI_MakeCylinder
from OCC.Core.BRepBuilderAPI import BRepBuilderAPI_Sewing, BRepBuilderAPI_MakeSolid, BRepBuilderAPI_Transform
from OCC.Core.Bnd import Bnd_Box
from OCC.Core.BRepBndLib import brepbndlib_Add
from OCC.Core.BRepMesh import BRepMesh_IncrementalMesh
from OCC.Core.StlAPI import StlAPI_Writer
from OCC.Core.GProp import GProp_GProps
from OCC.Core.BRepGProp import brepgprop_VolumeProperties
from OCC.Core.TopExp import TopExp_Explorer
from OCC.Core.TopAbs import TopAbs_SHELL
from OCC.Core.Quantity import Quantity_Color, Quantity_TOC_RGB
from OCC.Core.gp import gp_Pnt, gp_Dir, gp_Ax2, gp_Trsf
from OCC.Core.BOPAlgo import BOPAlgo_Builder
from OCC.Core.TopTools import TopTools_ListOfShape

```

Gambar 4. 7 Library yang digunakan pada Program I

Untuk keperluan import dan ekspor file, digunakan STEPControl_Reader dan IGESControl_Reader dari pustaka OCC.Core untuk membaca file model dari format STEP atau IGES. Proses pembacaan dilakukan menggunakan metode ReadFile() dan TransferRoots(), yang kemudian dikonversi menjadi objek TopoDS_Shape sebagai bentuk dasar dari geometri model. Hasil akhir pemrosesan diekspor ke dalam format STEP menggunakan write_step_file() atau STEPControl_Writer, ke format BREP menggunakan breptools_Write, serta ke format STL menggunakan StlAPI_Writer.

Untuk pemrosesan dan transformasi geometri, digunakan BRepBuilderAPI_Sewing untuk menjahit permukaan (shell) menjadi bentuk tertutup, dan BRepBuilderAPI_MakeSolid untuk mengubah shell tersebut menjadi bentuk solid. Transformasi skala dilakukan dengan memanfaatkan gp_Trsf sebagai transformasi matriks dan BRepBuilderAPI_Transform untuk menerapkannya pada model secara global. Untuk pembuatan fitur geometri tambahan, seperti blok pemotong dan lubang sambungan baut (bolt joint), digunakan BRepPrimAPI_MakeBox dan BRepPrimAPI_MakeCylinder. Posisi dan arah dari geometri tersebut ditentukan menggunakan gp_Pnt, gp_Dir, dan gp_Ax2.

Operasi boolean seperti pemotongan model atau pembentukan rongga (hollowing) dilakukan dengan BRepAlgoAPI_Cut dan BRepAlgoAPI_Common. Untuk penggabungan geometri dari sambungan baut ke bentuk utama secara aman, digunakan BOPAlgo_Builder dan TopTools_ListOfShape. Validasi ukuran dan volume model dilakukan dengan Bnd_Box dan brepbndlib_Add untuk mendapatkan nilai minimum dan maksimum dari koordinat model, serta

GProp_GProps dan brepprop_VolumeProperties untuk menghitung volume model dan volume tiap voxel saat proses hollowing.

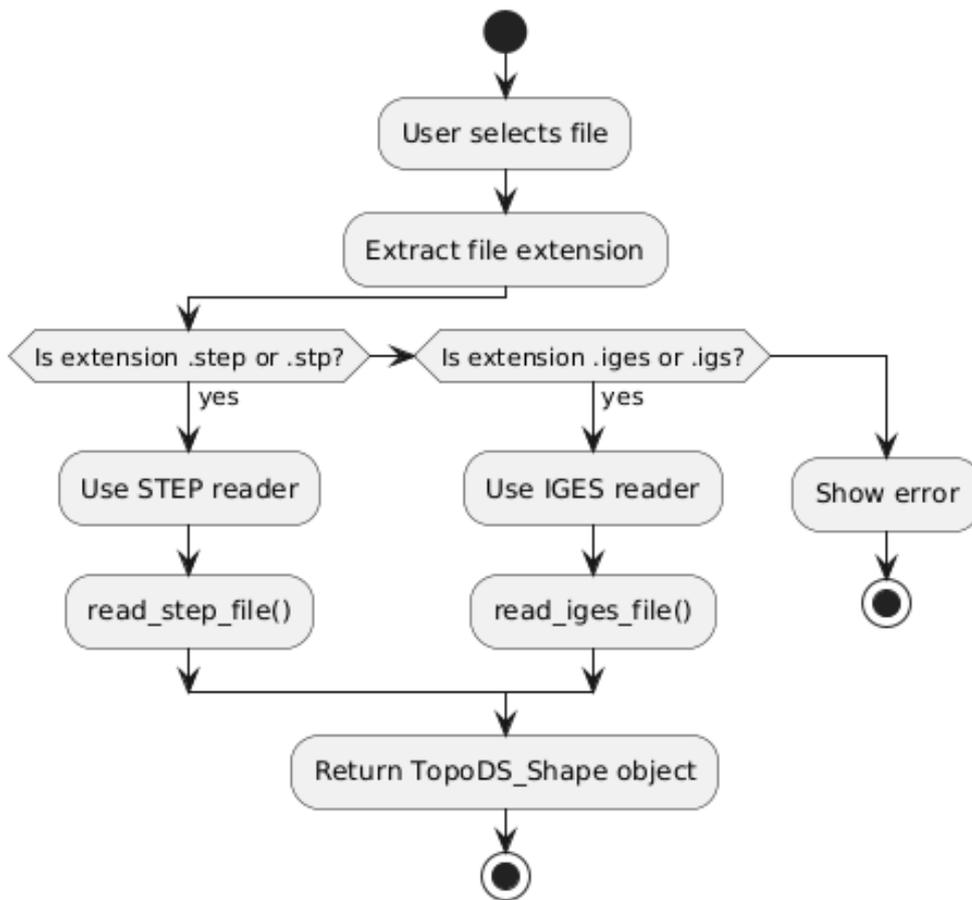
Sebelum diekspor ke STL, model dimesh menggunakan BRepMesh_IncrementalMesh agar permukaan solid dapat ditransformasikan menjadi mesh segitiga. Sementara itu, struktur topologi model dieksplorasi menggunakan TopExp_Explorer dan TopAbs_SHELL untuk mengambil bagian shell dari hasil sewing. Jika visualisasi diaktifkan, digunakan Quantity_Color dan Quantity_TOC_RGB untuk memberi warna dan pencahayaan pada tampilan. Tampilan 3D itu sendiri ditangani oleh qtViewer3d, namun hanya diaktifkan jika dibutuhkan saat debugging atau evaluasi hasil.

Untuk menjalankan proses slicing secara otomatis, digunakan modul subprocess dari pustaka standar *Python* untuk menjalankan PrusaSlicer melalui perintah CLI. Selain itu, proses voxel filtering pada tahap hollowing dipercepat dengan dukungan multiprocessing menggunakan multiprocessing dan joblib, khususnya Parallel dan delayed. Seluruh pustaka ini diorganisasikan secara modular dalam satu skrip utama, sehingga memudahkan integrasi ke dalam sistem GUI serta mendukung otomatisasi proses secara penuh dari file permukaan hingga file .gcode siap cetak.

4.4.2. Import dan Validasi Model

Proses awal dalam pemrosesan model dimulai dari pembacaan file desain 3D lambung kapal yang umumnya disediakan dalam format standar industri seperti STEP (.step, .stp) dan IGES (.iges, .igs). Untuk mengakomodasi kedua format tersebut, program dilengkapi dengan fungsi read_shape_from_file() yang akan secara otomatis mendeteksi ekstensi file dan memilih metode pembacaan yang sesuai.

Untuk file berformat STEP, digunakan pustaka STEPControl_Reader dari modul OCC.Core.STEPControl. Fungsi ReadFile() dipanggil untuk membuka file, dan dilanjutkan dengan TransferRoots() untuk mengekstrak bentuk utama dari file tersebut. Hasilnya kemudian dikonversi ke dalam objek TopoDS_Shape yang dapat diproses lebih lanjut oleh sistem. Prosedur serupa diterapkan untuk file IGES dengan menggunakan IGESControl_Reader dari modul OCC.Core.IGESControl.



Gambar 4. 8 diagram keputusan fase import file

Alur program dijabarkan dalam Gambar 4. 8. Jika proses pembacaan file gagal, program akan langsung menghentikan proses dan menampilkan pesan galat (`FileNotFoundException`) untuk menghindari kesalahan propagasi di tahap selanjutnya. Hanya file dengan ekstensi `.step`, `.stp`, `.igs`, dan `.iges` yang diperbolehkan. Dengan pendekatan ini, sistem dapat digunakan secara fleksibel pada berbagai model CAD hasil ekspor dari perangkat lunak desain lain seperti Rhino, Maxsurf, atau SolidWorks.

```

# 3. Compute bounding box
bbox = Bnd_Box(); brepbndlib_Add(hull, bbox)
xmin, ymin, zmin, xmax, ymax, zmax = bbox.Get()

print(f"[2] Bounds: X[{xmin:.1f},{xmax:.1f}] Y[{ymin:.1f},{ymax:.1f}] Z[{zmin:.1f},{zmax:.1f}]")

```

Gambar 4. 9 Perhitungan bounding box

Setelah file berhasil dimuat, model akan diproses menggunakan `Bnd_Box` dan `brepbndlib_Add()` untuk menghitung bounding box, yaitu dimensi minimum dan maksimum dari keseluruhan bentuk yang program lebih jelasnya pada Gambar 4. 9. Nilai-nilai ini akan disimpan ke dalam variabel `xmin`, `xmax`, `ymin`, `ymax`, `zmin`, dan `zmax`, dan ditampilkan pada

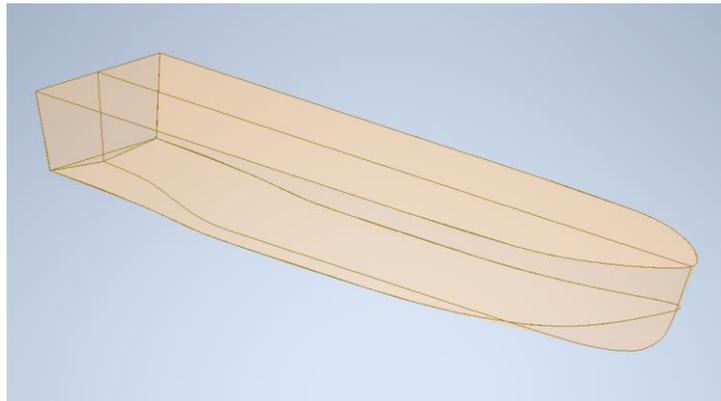
terminal sebagai referensi bagi pengguna. Informasi bounding box ini penting sebagai dasar perhitungan pada tahap-tahap selanjutnya seperti pemotongan blok, penempatan sambungan, serta pengecekan kompatibilitas dengan ukuran volume cetak printer 3D.

```
xmin_manual = 0.0
if xmin_manual is not None:
    xmin = xmin_manual
```

Gambar 4. 10 Opsi xmin override

Selain itu, pada Gambar 4. 10 pada kode terdapat opsi untuk menetapkan nilai xmin secara manual melalui variabel `xmin_manual`, untuk mengatasi kasus khusus di mana hasil bounding box tidak sesuai dengan ekspektasi akibat data geometri yang menyimpang atau mengandung titik bayangan yang tidak diperlukan (contohnya frame of reference maxsurf yang belum diatur titik AP dan FP nya). Dengan validasi ini, program memastikan bahwa model input sudah layak untuk diproses lebih lanjut pada tahap solidifikasi.

4.4.3. Proses Solidifikasi Model

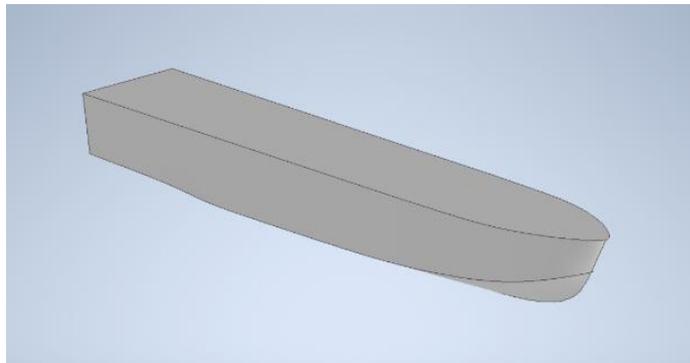


Gambar 4. 11 Lambung dalam bentuk surface

Setelah model permukaan berhasil dimuat dan divalidasi, langkah selanjutnya adalah mengubah bentuk permukaan terbuka menjadi bentuk tertutup (shell). Proses ini dilakukan menggunakan `BRepBuilderAPI_Sewing` dari pustaka `OCC.Core.BRepBuilderAPI`. Fungsi ini akan menjahit celah-celah kecil antar permukaan (face) yang belum saling terhubung yang nantinya akan menghasilkan bentuk seperti Gambar 4. 11. Shell yang dihasilkan dari proses sewing akan digunakan sebagai dasar untuk membentuk solid.

Setelah diperoleh shell yang utuh, dilakukan proses pembentukan solid menggunakan `BRepBuilderAPI_MakeSolid`. Fungsi ini akan mengemas shell ke dalam bentuk padat (solid body) yang bisa dikenakan operasi boolean selanjutnya seperti pemotongan dan hollowing.

Hasil akhir dapat dilihat pada Gambar 4. 12. Bentuk solid ini kemudian disimpan ke dalam variabel compound sebagai bentuk utama yang akan diproses terus-menerus hingga akhir.



Gambar 4. 12 Lambung yang telah solid

Tahapan ini penting karena model permukaan hasil dari *software* CAD umumnya tidak dapat langsung digunakan untuk slicing atau pencetakan 3D. Permukaan yang tidak sepenuhnya tertutup akan menyebabkan error dalam pembuatan mesh dan ekspor STL. Dengan menjahit permukaan dan membentuk solid, maka model dipastikan valid secara topologi dan dapat dikenakan operasi pemrosesan lanjutan seperti transformasi skala, hollowing, dan pemotongan blok.

4.4.4. Transformasi Skala

Setelah model berhasil dibentuk menjadi solid, dilakukan proses transformasi skala agar model sesuai dengan dimensi cetak yang diinginkan. Transformasi ini sangat penting karena file desain asli biasanya masih menggunakan ukuran sebenarnya (full scale), sementara proses pencetakan 3D hanya memungkinkan ukuran tertentu berdasarkan volume printer. Proses skala dilakukan terhadap keseluruhan model menggunakan transformasi global berbasis titik pusat (0, 0, 0) sebagai referensi. Transformasi dilakukan dengan menggunakan `gp_Trsf` dari `OCC.Core.gp`, yang merepresentasikan transformasi linier dalam ruang 3D. Transformasi ini kemudian diterapkan ke model menggunakan `BRepBuilderAPI_Transform`, dengan flag `copy=True` agar menghasilkan bentuk baru tanpa mengubah geometri asli.

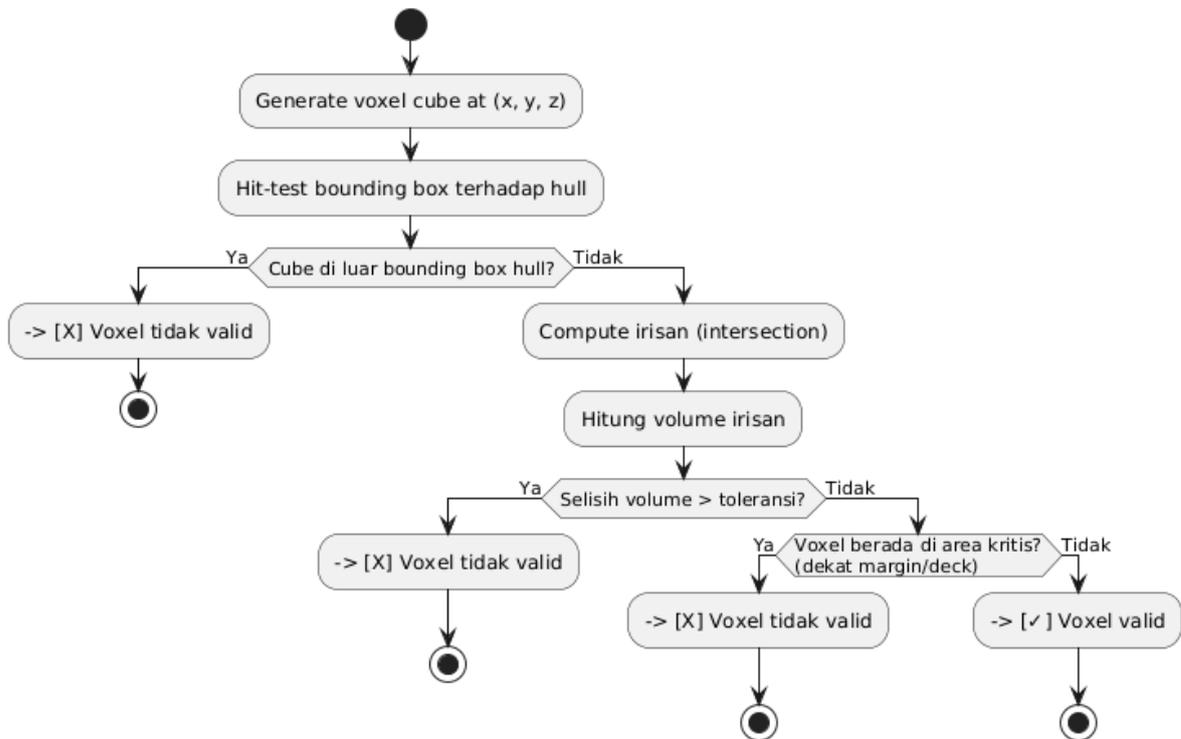
Program juga dirancang untuk efisien dengan menambahkan pengecekan terhadap nilai `SCALE_FACTOR`. Jika nilai skala adalah 1.0 (artinya tidak ingin diubah), maka proses transformasi dilewati dan model dikembalikan dalam bentuk aslinya tanpa perubahan. Hal ini mencegah pemborosan waktu dan sumber daya. Seluruh logika di atas dikemas dalam fungsi `apply_scaling(shape)` yang dipanggil tepat setelah model dibaca dari file. Fungsi ini menerima

objek TopoDS_Shape, melakukan transformasi (jika diperlukan), dan mengembalikan bentuk hasil skala.

Skala yang umum digunakan untuk prototipe adalah 1:100 atau 1:50, tergantung pada ukuran akhir model dan dimensi printer yang digunakan. Transformasi ini juga harus memperhatikan akurasi geometri, karena scaling dapat memengaruhi resolusi mesh jika tidak dikontrol.

4.4.5. Proses Hollowing

Pada program ini, hollowing dilakukan dengan pendekatan voxel-based filtering, yaitu dengan membentuk kisi 3D (voxel grid) di dalam model dan menghapus bagian interior berdasarkan kriteria tertentu. Hollowing atau pengosongan bagian dalam model bertujuan untuk mengurangi volume material yang dibutuhkan dalam proses pencetakan 3D serta mempercepat waktu produksi. Pada model lambung kapal, bagian dalam tidak perlu padat sepenuhnya, sehingga proses hollowing dapat dilakukan dengan tetap mempertahankan kekuatan struktural bagian luar.



Gambar 4. 13 Diagram keputusan fase hollowing

Proses ini dimulai dengan membentuk grid voxel yang dibatasi oleh bounding box model. Kisi voxel dibuat berdasarkan parameter VOXEL_SIZE, di mana setiap voxel

direpresentasikan sebagai kubus kecil (`BRepPrimAPI_MakeBox`) yang akan diuji keberadaannya di dalam model utama. Koordinat voxel dihitung dari titik (x_0, y_0, z_0) hingga (x_1, y_1, z_1) berdasarkan nilai bounding box yang telah ditentukan pada tahap sebelumnya. Voxel disusun dengan langkah sebesar nilai voxel (misalnya 15 mm).

Setiap voxel yang terbentuk kemudian diuji apakah berada benar-benar di dalam model dan bukan bagian dari margin atau area dekat permukaan. Pengujian dilakukan menggunakan `BRepAlgoAPI_Common` untuk mencari irisan antara voxel dan model solid. Kemudian, `brepgeom_VolumeProperties()` digunakan untuk menghitung massa (volume) irisan tersebut.

Voxel akan dianggap valid untuk dihapus jika:

- Massa irisan mendekati volume teoritis kubus voxel (`vol_cube`) dengan toleransi tertentu (`V_TOL`).
- Tidak berada dekat margin bawah atau dinding luar, agar tidak melemahkan struktur.

Jika memenuhi syarat tersebut, voxel akan dipotong dari model menggunakan `BRepAlgoAPI_Cut`.

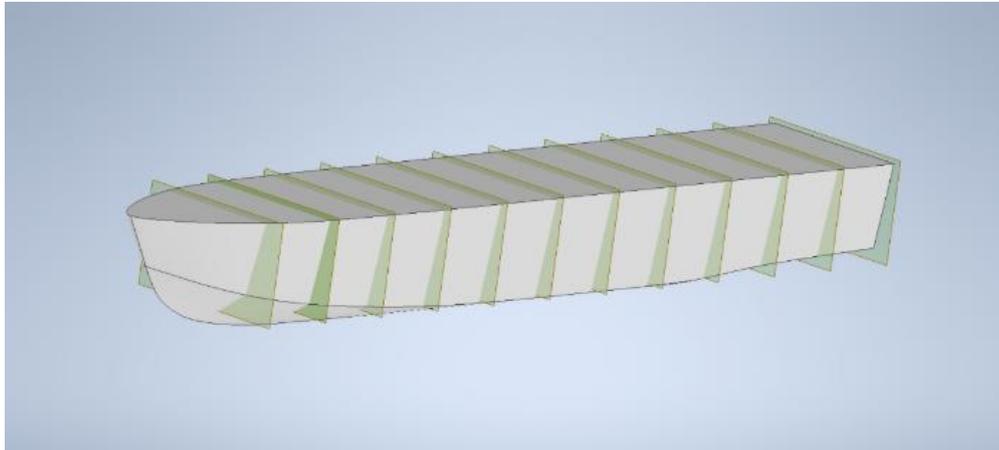
Karena jumlah voxel dapat mencapai ribuan hingga puluhan ribu, proses filtering ini dioptimalkan dengan menggunakan `Parallel` dan `delayed` dari pustaka `joblib`, yang menjalankan pengecekan voxel secara paralel sesuai jumlah core CPU. Logika keputusan fase hollowing dapat dilihat pada Gambar 4. 13. Hasilnya dikumpulkan ke dalam list valid, yang kemudian diproses satu per satu untuk dihapus dari model. Voxel yang dinyatakan valid akan dibuat dalam bentuk `BRepPrimAPI_MakeBox`, lalu dipotong dari model utama compound menggunakan `BRepAlgoAPI_Cut`. Proses ini dilakukan dalam loop, dan setiap 1000 voxel yang berhasil dihapus akan dicetak sebagai progres di terminal.

Setelah seluruh voxel interior dihapus, model akan memiliki rongga di bagian dalam yang signifikan, tetapi tetap mempertahankan ketebalan dinding luar secara struktural. Dengan pendekatan ini, proses hollowing dapat dikontrol dengan parameter `VOXEL_SIZE` dan memberikan fleksibilitas tinggi terhadap kebutuhan pengguna.

4.4.6. Pemotongan dan Penambahan Sambungan

Setelah model dikosongkan (hollow), langkah selanjutnya adalah memotong model menjadi beberapa blok agar dapat dicetak menggunakan printer 3D dengan volume terbatas.

Pemotongan seperti pada Gambar 4. 14, dilakukan sepanjang sumbu-x berdasarkan parameter BLOCK_LEN yang ditentukan pengguna. Selain itu, untuk memudahkan perakitan ulang model setelah dicetak, ditambahkan fitur sambungan antarblok berupa lubang baut (bolt joints) berbentuk kotak berlubang silinder.



Gambar 4. 14 Ilustrasi pemotongan per block_len

Pemotongan dilakukan dengan membuat kotak pemotong (BRepPrimAPI_MakeBox) sebesar dimensi blok, dari x_0 ke x_1 dengan lebar dan tinggi sesuai y_{min} – y_{max} dan z_{min} – z_{max} . Setiap blok kemudian diperoleh melalui proses BRepAlgoAPI_Common() antara model solid yang telah di-hollow dan kotak tersebut. Hasilnya adalah potongan geometri model asli yang terbatas pada satu segmen panjang BLOCK_LEN.

Sebelum pemotongan blok, program juga melakukan pemotongan horizontal di bagian atas model (deck cutting) untuk membuka permukaan atas lambung. Proses ini dilakukan dengan membentuk grid voxel horizontal di ketinggian $z_{max} - VOXEL_SIZE / 2$, lalu memotong volume tersebut dari model utama menggunakan BRepAlgoAPI_Cut. Tujuannya adalah memberikan akses atau ruang di bagian atas untuk keperluan teknis, seperti inspeksi interior model atau perakitan elektronik (jika model digunakan dalam sistem berpengerak).

Setelah pemotongan blok, program menambahkan fitur sambungan baut (bolt joint) di permukaan antarblok. Sambungan dibuat dengan membentuk kotak (BRepPrimAPI_MakeBox) di sekitar posisi sambungan (koordinat x di tengah bidang potong), dan diberi lubang silinder di tengahnya (BRepPrimAPI_MakeCylinder). Lubang diletakkan horizontal sepanjang arah-x, sehingga saat dicetak, pengguna dapat memasukkan baut atau pin antarblok secara manual untuk penyambungan.

Lokasi sambungan ditentukan pada dua titik sumbu-y (dekat y_{min} dan y_{max}) di tiap bidang potong. Dimensi kotak sambungan disesuaikan berdasarkan diameter baut (default: M3

atau 3.2 mm), yang dapat diatur pada variabel M3_DIAMETER. Gabungan antara sambungan dan model utama dilakukan dengan BOPAlgo_Builder, yang menyatukan bentuk solid utama dan sambungan baru dalam satu kesatuan.

4.4.7. Ekspor File STEP dan STL

Setelah proses pemodelan selesai, masing-masing bagian model (blok) diekspor ke dalam dua format utama, yaitu .stl dan .step. Format .stl digunakan sebagai input untuk proses slicing dan pencetakan 3D, sedangkan .step disiapkan sebagai arsip geometri parametrik yang digunakan terutama untuk kebutuhan debugging dan verifikasi oleh pengembang sistem.

Format .stl adalah format berbasis mesh segitiga yang kompatibel dengan hampir semua perangkat lunak slicer. Sebelum ekspor, permukaan tiap blok harus dimesh terlebih dahulu menggunakan BRepMesh_IncrementalMesh untuk mengubah bentuk solid menjadi representasi segitiga.

Konversi ini dikontrol oleh beberapa parameter yang menentukan kualitas dan ketelitian mesh:

a. **DEFLECTION**

Menentukan deviasi maksimum antara kurva asli dan segmen segitiga yang menggantikannya. Nilai yang lebih kecil akan menghasilkan mesh lebih halus, tetapi ukuran file STL akan lebih besar.

b. **RELATIVE**

Jika diset True, maka DEFLECTION akan dianggap sebagai persentase relatif terhadap dimensi objek. Jika False, maka dianggap absolut (dalam satuan panjang model, misalnya mm).

c. **ANGLE**

Menentukan sudut maksimum antara dua segitiga yang berbagi sisi. Semakin kecil nilainya, semakin halus transisi antar segmen.

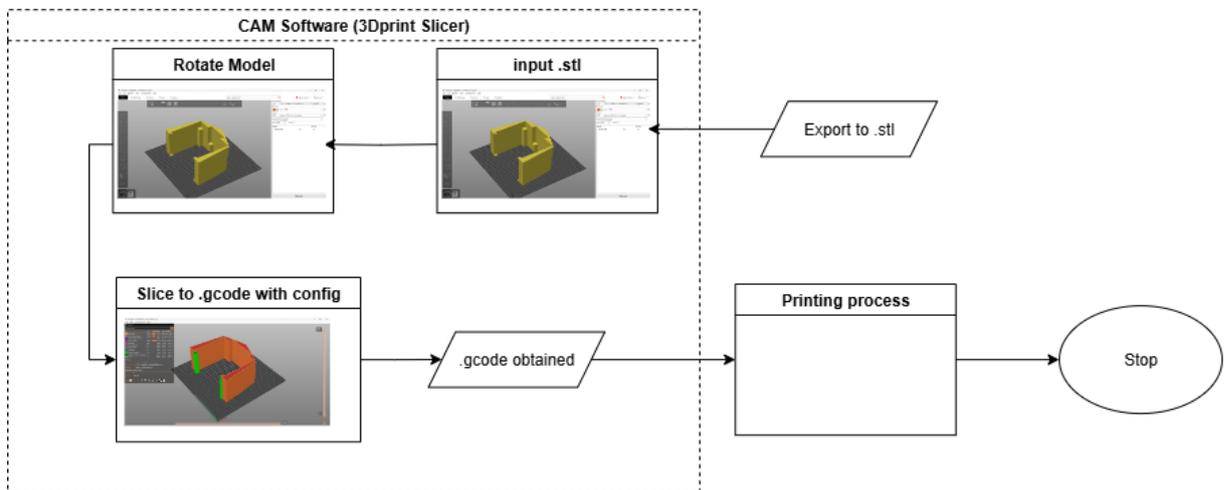
d. **PARALEL_MESH**

Menentukan apakah proses meshing dijalankan secara paralel untuk mempercepat pemrosesan pada sistem multi-core.

File STL dari tiap blok disimpan secara otomatis dalam folder output_stl/ dengan format penamaan seperti block_01.stl, block_02.stl, dan seterusnya. File inilah yang akan digunakan pada proses slicing untuk menghasilkan file .gcode.

Sebagai tambahan, setiap blok juga diekspor ke format .step menggunakan fungsi write_step_file(). Berkas STEP ini tidak digunakan langsung untuk pencetakan, namun sangat penting bagi pengembang untuk melakukan debugging bentuk akhir model, terutama setelah proses pemotongan blok dan penambahan sambungan. Format STEP memungkinkan pengguna membuka model di *software* CAD lain untuk memverifikasi ukuran, posisi joint, dan integritas model 3D secara visual dan numerik.

4.5. Program II

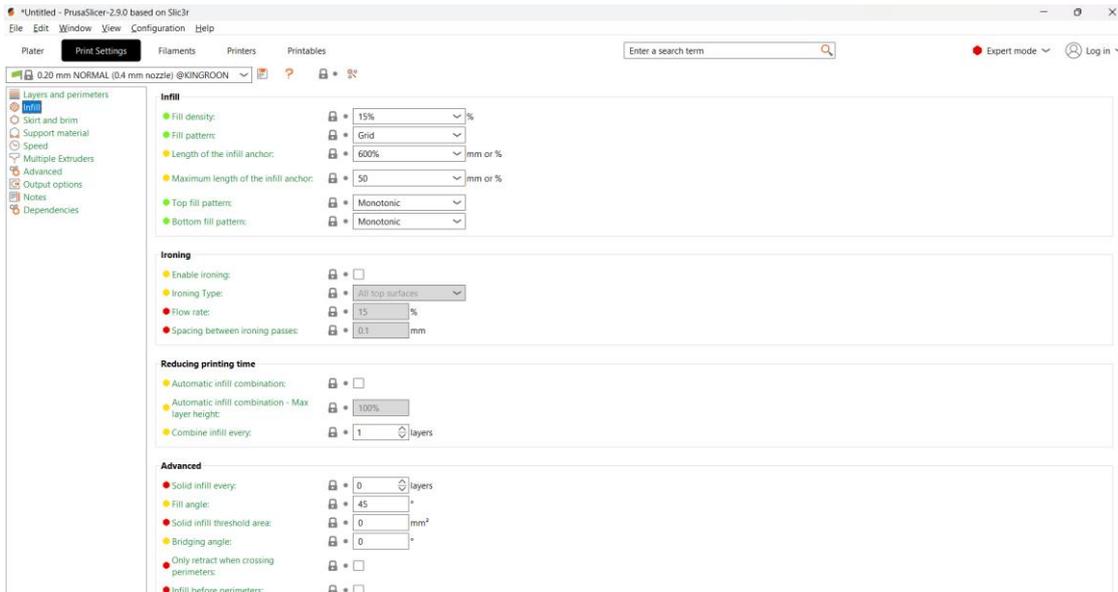


Gambar 4. 15 Alur Perancangan Program II

Pada Gambar 4. 15, subbab ini mencakup seluruh tahapan pengolahan file STL dari program sebelumnya hingga menjadi .gcode siap cetak pada *3D Printer*

4.5.1. Perancangan Konfigurasi *3D Printer*

Dalam proses pencetakan 3D, konfigurasi printer memiliki peranan penting untuk memastikan keberhasilan pencetakan serta kualitas dari model yang dihasilkan. Oleh karena itu, tahap perancangan konfigurasi dilakukan menggunakan perangkat lunak PrusaSlicer yang menyediakan berbagai parameter cetak yang dapat disesuaikan seperti pada Gambar 4. 16.



Gambar 4. 16 Proses penyetalan parameter

Beberapa parameter penting yang harus diatur dalam konfigurasi ini antara lain: ketinggian lapisan (layer height) yang mempengaruhi resolusi hasil cetak, kecepatan cetak (print speed) untuk menentukan durasi proses pencetakan, persentase isian dalam (infill) yang mempengaruhi kekuatan dan konsumsi material, struktur penyangga (support structure) untuk mendukung bagian-bagian yang menggantung, serta temperatur lantai (bed) dan nozzle, yang harus disesuaikan dengan jenis material filament yang digunakan. Penyesuaian parameter-parameter tersebut dilakukan untuk mencapai kualitas cetak yang optimal, mempercepat waktu produksi, dan meminimalkan risiko kegagalan cetak.

```

# generated by PrusaSlicer 2.9.0 on 2025-06-09 at 17:03:50 UTC
arc_fitting = disabled
autoemit_temperature_commands = 1
automatic_extrusion_widths = 0
automatic_infill_combination = 0
automatic_infill_combination_max_layer_height = 100%
avoid_crossing_curled_overhangs = 0
avoid_crossing_perimeters = 0
avoid_crossing_perimeters_max_detour = 0
bed_custom_model =
bed_custom_texture =
bed_shape = 3x3,301x3,301x301,3x301
bed_temperature = 70
before_layer_gcode = ;BEFORE_LAYER_CHANGE\nG92 E0\n;{layer_z}\n\n
between_objects_gcode =
binary_gcode = 0
bottom_fill_pattern = monotonic
bottom_solid_layers = 4
bottom_solid_min_thickness = 0
bridge_acceleration = 250
bridge_angle = 0
bridge_fan_speed = 100
bridge_flow_ratio = 0.95
bridge_speed = 50
brim_separation = 0
brim_type = outer_only
brim_width = 0
chamber_minimal_temperature = 0
chamber_temperature = 0
color_change_gcode = M600
colorprint_heights =
compatible_printers_condition_cumulative = "printer_model~/((KP3S|KP3SBLTOUCH|KP3SProS1|KP3SProS1BLTOUCH).*)/ and nozzle_diameter[0]==0.4";printer_notes~/.*PRINTER_VENDOR_KINGROON.*/
complete_objects = 0
cooling = 1
cooling_tube_length = 5
cooling_tube_retraction = 91.5
default_acceleration = 500
default_filament_profile = "Generic PLA @KINGROON"
default_print_profile = "0.16 mm OPTIMAL (0.4 mm nozzle) @KINGROON"
deretract_speed = 40

```

Gambar 4. 17 isi dari file penyetalan 3D print

Setelah seluruh parameter disesuaikan dengan kebutuhan pencetakan model lambung kapal, konfigurasi ini kemudian diekspor dalam format file .ini yang isinya dapat dilihat pada

Gambar 4. 17. File konfigurasi ini digunakan sebagai input pada tahap slicing otomatis, sehingga seluruh proses dapat berjalan konsisten dan terstandarisasi tanpa perlu melakukan pengaturan ulang secara manual pada setiap proses pencetakan.

4.5.2. Identifikasi Library dan Tools

Program kedua bertugas untuk mengotomatisasi proses konversi file .stl menjadi file .gcode yang siap dicetak dengan printer 3D. Proses ini dilakukan melalui pemanggilan *software* PrusaSlicer menggunakan command-line interface (CLI). Untuk mendukung alur ini, digunakan beberapa pustaka *Python* dan tools eksternal yang dikonfigurasi agar berjalan tanpa interaksi pengguna. Beberapa library yang digunakan dalam program ini yaitu:

1. Modul *Python*
 - a. OS
 - b. SUBPROCESS
 - c. ARGPARSE
2. Modul OCC (OpenCASCADE via *PythonOCC-core*)

Meskipun modul OCC umumnya digunakan untuk manipulasi file CAD seperti STEP/IGES, dalam *script* ini OCC digunakan untuk melakukan transformasi geometri terhadap file .stl.

- a. OCC.Extend.DataExchange

Untuk membaca model .stl ke dalam bentuk shape OCC, dan menulis ulang hasil transformasi ke file .stl baru.

- b. gp_Trnsf, gp_Ax1, gp_Pnt, gp_Dir

Modul transformasi geometri, khususnya untuk membuat transformasi rotasi. Dalam kasus ini, digunakan untuk memutar model 90° terhadap sumbu Y (arah longitudinal kapal) agar orientasinya sesuai dengan arah cetak yang diinginkan oleh pengguna.

- c. BRepBuilderAPI_Transform

Mengaplikasikan transformasi ke bentuk shape OCC yang mewakili model .stl, menghasilkan bentuk baru yang sudah dirotasi.

Dengan penggunaan OCC, transformasi rotasi dilakukan secara presisi terhadap model 3D sebelum dikirim ke proses slicing. Hal ini penting karena orientasi model sangat mempengaruhi kualitas dan keberhasilan hasil cetak.

3. PrusaSlicer CLI

PrusaSlicer CLI digunakan dalam mode command-line untuk melakukan slicing tanpa GUI. *Script* mengirimkan parameter berikut:

a. -- load

memuat file .ini sebagai konfigurasi printer.

b. -- slice

perintah utama untuk memulai slicing.

c. -- output

menentukan path output .gcode.

d. [input_stl]

file .stl hasil rotasi yang akan diproses

PrusaSlicer dijalankan per file secara otomatis dalam loop, memastikan setiap blok model diproses secara batch.

4. Visualisasi: PyVista

PyVista `pyvista.read()` dan `plotter.show(screenshot=...)` digunakan untuk membaca dan merender hasil rotasi model .stl. Gambar preview disimpan dalam bentuk .png untuk memberikan umpan balik visual mengenai orientasi dan bentuk model, terutama berguna saat mengecek kesalahan sebelum proses pencetakan dilakukan.

4.5.3. Proses Transformasi Model

Sebelum dilakukan proses slicing, setiap file .stl hasil pemotongan blok perlu ditransformasi agar memiliki orientasi yang sesuai dengan arah pencetakan pada printer 3D. Tujuan dari proses transformasi model sebelum slicing adalah untuk memastikan bahwa orientasi geometris dari model kapal telah sesuai dengan kebutuhan optimal pencetakan 3D. Dengan memutar model sebesar 90 derajat terhadap sumbu Y, program dapat mengatur posisi lambung kapal agar tegak lurus terhadap permukaan cetak, atau sebaliknya, tergantung strategi pencetakan yang dipilih. Orientasi ini sangat krusial karena dapat secara signifikan mengurangi

kebutuhan struktur penyangga (support structure), terutama pada bagian bawah lambung kapal yang biasanya memiliki kelengkungan kompleks.

Selain itu, transformasi ini juga membantu menyelaraskan arah lapisan cetak (layer) dengan bentuk alami dari model, sehingga meningkatkan kekuatan hasil cetak dan meminimalkan risiko deformasi. Dengan orientasi yang tepat sejak awal, proses slicing dapat dilakukan dengan lebih efisien, menghasilkan G-code yang lebih bersih, serta meningkatkan kualitas dan keberhasilan proses pencetakan secara keseluruhan. Langkah yang dilakukan yaitu:

1. Membaca file .stl

Program menggunakan `read_stl_file()` dari modul `OCC.Extend.DataExchange` untuk membaca dan mengubah isi file .stl menjadi bentuk shape (entitas geometri OCC) yang dapat dimanipulasi:

```
shape = read_stl_file(input_stl)
```

2. Membentuk transformasi rotasi

Untuk memutar shape, digunakan `gp_Trsf()` yang dikonfigurasi dengan sumbu putar (`gp_Ax1`) dan arah (`gp_Dir`). Dalam hal ini, rotasi dilakukan terhadap sumbu Y (0,1,0) sebesar -90° atau -1.5708 radian:

```
axis = gp_Ax1(gp_Pnt(0, 0, 0), gp_Dir(0, 1, 0))
```

```
trsf = gp_Trsf()
```

```
trsf.SetRotation(axis, -1.5708)
```

3. Menerapkan transformasi pada model

Transformasi kemudian diterapkan ke shape STL menggunakan `BRepBuilderAPI_Transform()`, menghasilkan model baru yang sudah dirotasi:

```
rotated_shape = BRepBuilderAPI_Transform(shape, trsf, True).Shape()
```

4. Menyimpan hasil rotasi

Model hasil transformasi ditulis ulang dalam bentuk file .stl baru, dengan nama yang dibedakan (`*_rotated.stl`) dan disimpan ke dalam folder `rotated/`. File ini akan digunakan sebagai input utama dalam proses slicing:

```
write_stl_file(rotated_shape, rotated_stl, linear_deflection=0.05)
```

4.5.4. Proses Slicing File STL

Setelah model .stl berhasil ditransformasi ke orientasi yang sesuai, tahap selanjutnya adalah melakukan proses slicing untuk menghasilkan file .gcode siap cetak. Proses ini dilakukan menggunakan PrusaSlicer, sebuah perangkat lunak open-source yang mendukung pemanggilan melalui command-line interface (CLI). Program memanfaatkan kemampuan ini untuk mengotomatisasi konversi dari file .stl menjadi .gcode, tanpa perlu intervensi pengguna melalui GUI.

Slicing dilakukan dalam satu loop yang memproses seluruh file .stl hasil transformasi yang berada dalam direktori rotated/. Setiap file dipasangkan dengan konfigurasi printer 3D yang telah disiapkan sebelumnya dalam bentuk file .ini, yang mencakup parameter seperti ukuran nozzle, kecepatan cetak, suhu, dan ketebalan lapisan (layer height).

Perintah slicing dikirim menggunakan subprocess.run() dengan struktur sebagai berikut:

```
command = [  
    prusa_slicer_path,  
    "--load", profile_ini,  
    "--slice",  
    "--output", output_gcode,  
    rotated_stl  
]
```

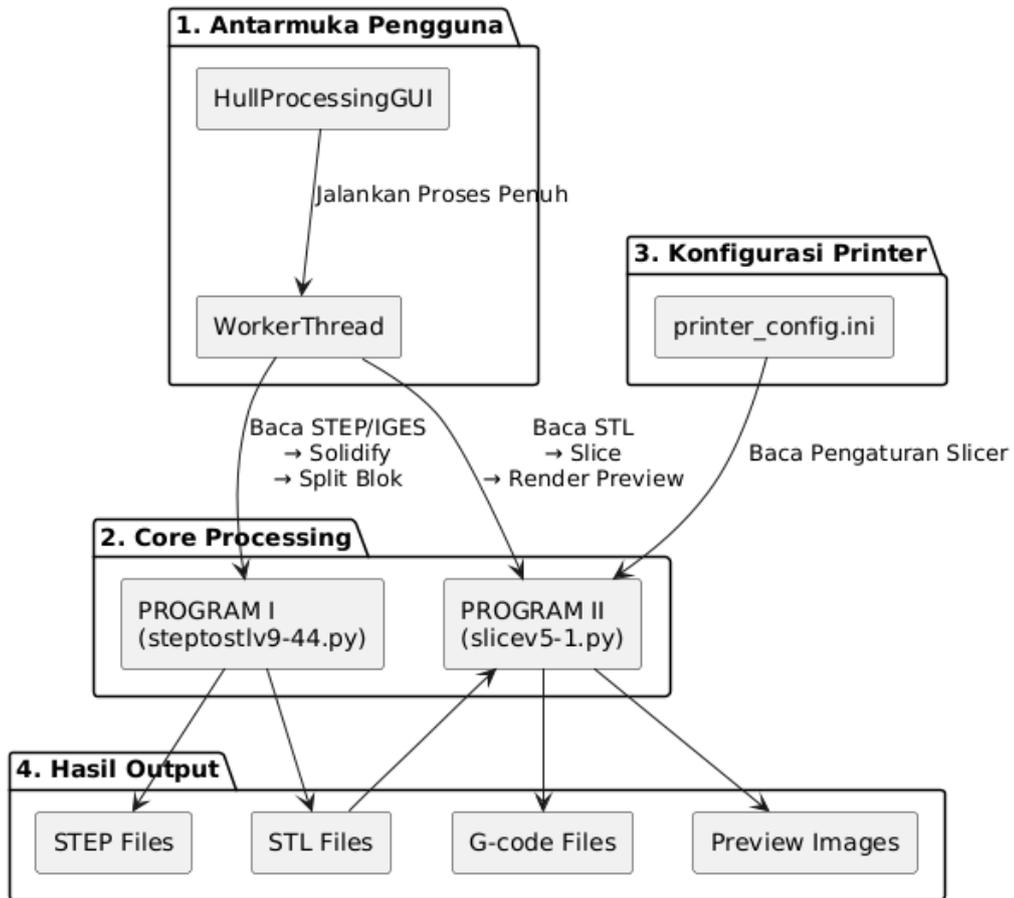
Penjelasan parameter:

- --load: Menentukan file konfigurasi printer (.ini) yang akan digunakan.
- --slice: Menandai bahwa perintah ini bertujuan untuk melakukan slicing file input.
- --output: Menentukan path dan nama file output .gcode hasil slicing.
- rotated_stl: Path ke file .stl yang akan di-slice, yakni model hasil rotasi sebelumnya.

Selama proses ini, output dari PrusaSlicer ditangkap melalui parameter capture_output=True dan dicetak ke terminal untuk kebutuhan debugging. Jika terjadi kesalahan, output error juga akan ditampilkan agar pengguna dapat segera menelusuri penyebabnya.

Slicing otomatis ini memberikan fleksibilitas tinggi dalam pemrosesan massal, serta meminimalisir risiko kesalahan manual saat proses ekspor G-code. File .gcode hasil slicing kemudian dapat langsung digunakan untuk mencetak prototipe model lambung kapal di printer 3D, sesuai dengan blok-blok pemotongan sebelumnya.

4.6. Program III: Antarmuka Pengguna Berbasis PyQt5



Gambar 4. 18 Diagram korelasi antar program kepada program GUI

Setelah seluruh proses backend dalam konversi model lambung kapal selesai disusun dalam bentuk skrip, tahap berikutnya adalah membangun antarmuka pengguna (GUI). Tujuan dari GUI ini adalah untuk menyederhanakan penggunaan sistem secara menyeluruh oleh pengguna non-programmer, serta mempercepat workflow dari input file desain hingga keluaran file .gcode yang siap dicetak. Antarmuka dibangun menggunakan pustaka PyQt5, salah satu framework GUI berbasis *Python* yang kompatibel dengan sistem desktop multiplatform. Diagram korelasi antar program dengan GUI dapat dijelaskan dalam Gambar 4. 18. Dengan tampilan yang sederhana namun fungsional, antarmuka ini memungkinkan pengguna memilih file desain awal, memasukkan parameter utama (skala, panjang blok, ukuran voxel), serta

memilih konfigurasi printer yang sesuai. Proses dijalankan dengan satu kali klik, sementara proses backend dari dua skrip sebelumnya dijalankan secara otomatis dan terpantau melalui panel log.

4.6.1. Identifikasi Komponen GUI

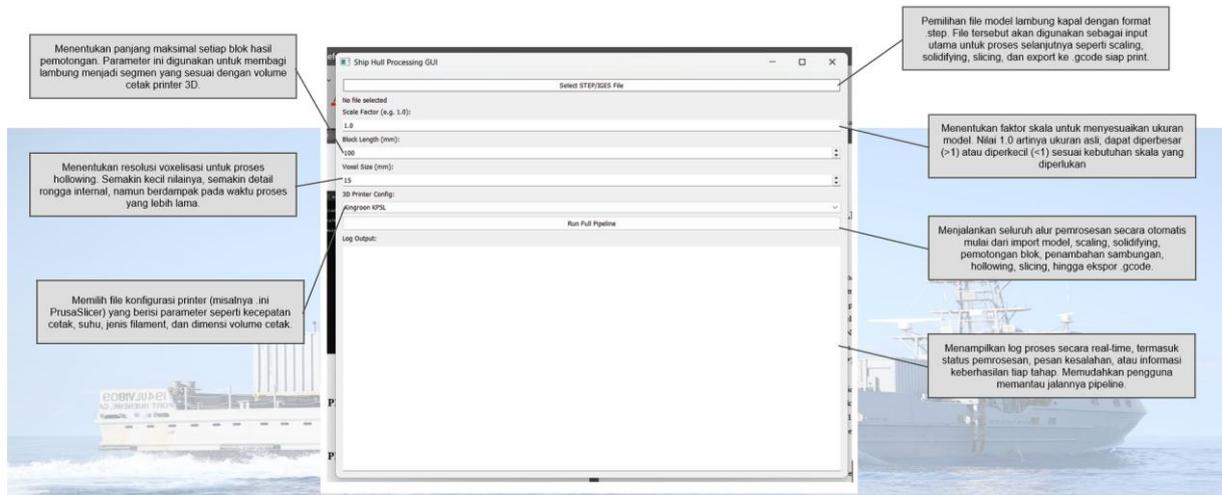
Antarmuka pengguna menggunakan komponen standar dari modul PyQt5.QtWidgets, dengan elemen-elemen berikut:

- QLabel: Menampilkan label teks untuk memberi keterangan pada kolom input (misalnya: “Input File”, “Scale”, “Block Length”, dll.).
- QLineEdit: Kolom input untuk memasukkan nilai numerik (seperti skala dan ukuran voxel).
- QPushButton: Tombol eksekusi program, pemilihan file, serta konfirmasi pemrosesan.
- QFileDialog: Digunakan untuk memilih file input STEP secara interaktif.
- QComboBox: Digunakan untuk memilih profil printer dari daftar yang tersedia (misalnya configkp5l.ini, configcr10.ini).
- QTextEdit: Panel log yang menampilkan jalannya proses, termasuk status keberhasilan atau pesan kesalahan yang terjadi selama eksekusi.

Komponen-komponen tersebut dirancang secara modular untuk memungkinkan pengembangan lebih lanjut, seperti menambahkan fitur preview visual atau pengaturan lanjutan printer di masa depan.

4.6.2. Desain Tampilan GUI

Desain tampilan antarmuka pengguna seperti pada Gambar 4. 19 dirancang dengan prinsip kesederhanaan, aksesibilitas, dan fungsionalitas penuh, sehingga pengguna dapat menjalankan keseluruhan proses hanya dengan beberapa interaksi dasar. Tampilan GUI terdiri atas beberapa elemen yang tersusun secara vertikal menggunakan layout utama bertipe QVBoxLayout, yang kemudian dikombinasikan dengan layout horizontal (QHBoxLayout) untuk pengelompokan input.



Gambar 4. 19 Tampilan dan kegunaan GUI

Secara umum, tampilan terdiri atas tiga bagian utama:

1. Bagian Input Parameter

Di bagian ini, pengguna dapat:

- Memilih file input desain dalam format .igs, .iges, .stp, atau .step melalui tombol Browse.
- Mengisi nilai skala model (contoh: 0.01 untuk skala 1:100).
- Mengisi panjang blok (block length) sesuai batas ukuran printer.
- Menentukan ukuran voxel (voxel size) untuk proses hollowing interior.
- Memilih konfigurasi printer dalam format .ini melalui dropdown.

2. Bagian Kontrol Proses

Terdapat tombol utama seperti:

- Start Processing: Menjalankan dua skrip utama secara berurutan (solidifikasi–hollow–slicing).
- Clear Log (opsional): Menghapus isi log untuk memulai sesi baru (jika disediakan).

3. Bagian Panel Log

Seluruh proses backend yang dijalankan akan mencetak log ke dalam area QTextEdit. Log ini mencakup:

- Status pembacaan file dan validasi ekstensi.

- d. Proses scaling, solidifikasi, hollowing, dan pemotongan blok.
- e. Proses rotasi model STL dan konversi ke G-code.
- f. Error atau peringatan jika terjadi kegagalan dalam pemrosesan.

Desain ini menghindari kompleksitas visual, namun tetap cukup fleksibel untuk menampung seluruh parameter penting. Struktur modular layout juga memungkinkan modifikasi atau penambahan fitur tambahan di masa depan, seperti preview 3D, penjadwalan slicing batch, atau pemilihan output path secara manual.

4.6.3. Eksekusi Proses Backend

Setelah pengguna mengisi seluruh parameter dan menekan tombol Start, GUI akan menjalankan dua skrip backend secara berurutan. Proses ini dilakukan melalui fungsi `subprocess.run()` yang dipanggil dalam thread utama, sehingga sistem dapat mengeksekusi *script* solidifikasi dan slicing tanpa memerlukan interaksi tambahan. Eksekusi otomatis ini memungkinkan seluruh pipeline dari file desain hingga file `.gcode` terjadi dalam satu klik.

Skrip pertama (`steptostlv9.py`) akan membaca file STEP/IGES, melakukan solidifikasi, scaling, pemotongan blok, dan hollowing interior menggunakan pendekatan voxelisasi. Hasilnya berupa file `.stl` per blok yang disimpan dalam direktori `output_stl/`.

Setelah proses pertama selesai, GUI secara otomatis menjalankan skrip kedua (`slicev5.py`) untuk melakukan rotasi orientasi file `.stl`, kemudian menjalankan perintah PrusaSlicer melalui command-line interface. File `.gcode` hasil pemrosesan disimpan di folder `hasil/`, sedangkan gambar pratinjau tiap blok disimpan dalam format `.png` di lokasi yang sama.

Eksekusi proses ini ditampilkan secara real-time dalam panel log (QtTextEdit) pada GUI. Informasi seperti status file, jumlah blok, ukuran voxel, waktu proses, serta pesan kesalahan akan ditampilkan secara kronologis. Hal ini membantu pengguna memantau setiap tahap dengan mudah dan segera mengetahui apabila terjadi kesalahan, seperti file yang tidak terbaca, kesalahan saat slicing, atau direktori output yang hilang.

Fungsi `run_next_program(command)` digunakan untuk menjalankan setiap tahap secara modular, dan pengguna dapat memperluasnya untuk menambahkan skrip ketiga, seperti proses pemindahan file ke SD card, simulasi G-code, atau pengolahan lanjutan.

4.6.4. Penanganan Kesalahan dan Validasi Input

Dalam pengembangan antarmuka pengguna berbasis GUI, aspek validasi input dan penanganan kesalahan menjadi komponen penting untuk menjamin program berjalan stabil dan sesuai ekspektasi pengguna. Oleh karena itu, pada sistem ini telah diterapkan beberapa mekanisme untuk mendeteksi kesalahan sejak awal dan memberikan umpan balik langsung kepada pengguna. Salah satu bentuk validasi dilakukan pada tahap awal, yaitu pengecekan format file input. Sistem secara eksplisit hanya menerima file dengan ekstensi .igs, .iges, .stp, atau .step, dan akan menolak file dengan ekstensi lain. Jika pengguna mencoba memilih file dengan format yang tidak didukung, maka pesan kesalahan akan ditampilkan di panel log dan proses tidak akan dilanjutkan.

Selanjutnya, nilai parameter seperti scale, block length, dan voxel size juga divalidasi untuk memastikan bahwa input tersebut merupakan angka yang valid dan berada dalam rentang yang masuk akal. Misalnya, jika pengguna mengosongkan kolom atau mengisi dengan karakter non-numerik, maka proses akan dihentikan sebelum skrip backend dijalankan. Penanganan kesalahan juga diterapkan saat pemanggilan proses eksternal dengan `subprocess.run()`. Output dari proses tersebut akan ditampilkan secara penuh di panel log. Dengan begitu, pengguna dapat mengetahui secara langsung apakah pemrosesan berhasil, atau jika terjadi kegagalan, di bagian mana kesalahan terjadi.

Secara umum, pendekatan ini bertujuan untuk menjadikan sistem tidak hanya otomatis, tetapi juga tangguh terhadap input yang tidak sesuai serta informatif dalam menghadapi kesalahan, sehingga pengguna dapat dengan cepat melakukan koreksi tanpa perlu melihat ke dalam kode program.

Halaman ini sengaja dikosongkan

BAB 5 PENGUJIAN DAN PEMBAHASAN

5.1. Uji Coba Program

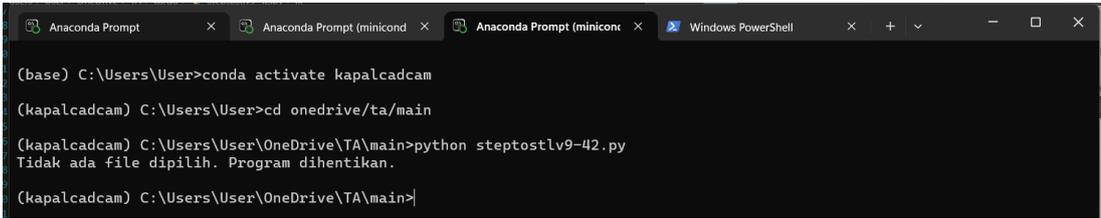
Setelah seluruh tahapan perancangan dan pengembangan program selesai dilakukan, tahap selanjutnya adalah melakukan pengujian terhadap fungsionalitas masing-masing program secara terpisah. Uji coba ini bertujuan untuk memastikan bahwa setiap bagian dari sistem baik program utama maupun program pelengkap dapat berjalan sesuai dengan fungsinya masing-masing sebelum dilakukan integrasi lebih lanjut. Pengujian dilakukan dengan menggunakan sejumlah data masukan yang telah disiapkan sebelumnya, dan hasil yang diperoleh akan diamati serta dicatat sebagai dasar untuk evaluasi dan validasi sistem secara keseluruhan.

5.1.1. Pengujian Program I

Pengujian Program I diawali dengan pengisian nilai parameter konfigurasi, yang meliputi `SCALE_FACTOR`, `BLOCK_LEN`, dan `VOXEL_SIZE`. Nilai `SCALE_FACTOR` digunakan untuk menentukan skala model terhadap ukuran aslinya, `BLOCK_LEN` merupakan panjang tiap blok pemotongan model dalam satuan milimeter, dan `VOXEL_SIZE` digunakan dalam proses filterisasi internal model untuk menghasilkan rongga atau kekosongan sesuai parameter cetak 3D.

Setelah parameter ditentukan, langkah selanjutnya adalah inialisasi environment melalui perintah `conda activate kapalcadcam` pada Anaconda Prompt, yang merupakan environment khusus berisi dependensi dan pustaka yang diperlukan oleh program. Kemudian, program dijalankan dengan perintah seperti pada Gambar 5. 1

```
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
```

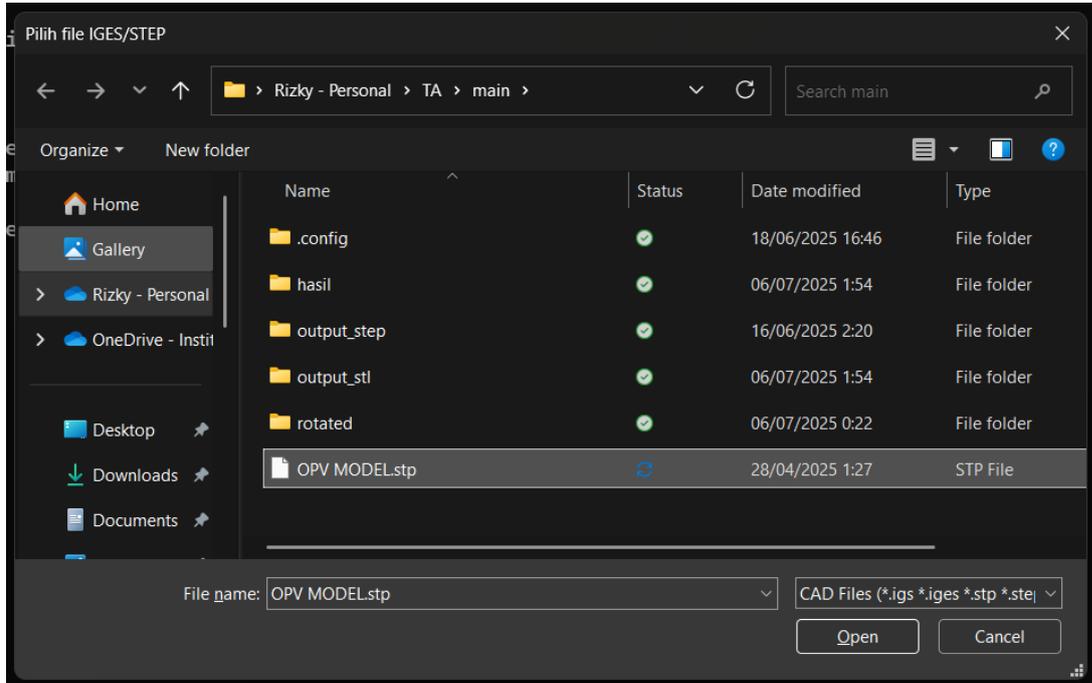


```
(base) C:\Users\User>conda activate kapalcadcam
(kapalcadcam) C:\Users\User>cd onedrive/ta/main
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.
(kapalcadcam) C:\Users\User\OneDrive\TA\main>
```

Gambar 5. 1 Running program

Selanjutnya, program ini bertanggung jawab untuk memuat file model (format STEP atau IGES), melakukan proses *solidification* atau konversi permukaan ke bentuk padat,

melakukan penyaringan voxel untuk membuat bagian dalam menjadi *hollow*, serta menghasilkan output berupa file .stl siap cetak. Apabila belum ada file yang dipilih, maka program secara otomatis menghentikan proses dan menampilkan pesan “Tidak ada file dipilih. Program dihentikan.” Hal ini menunjukkan bahwa mekanisme validasi input file berjalan dengan baik.



Gambar 5. 2 Pemilihan file

Setelah program dijalankan, akan muncul pop-up window (Gambar 5. 2) berupa jendela dialog untuk memilih file masukan. File yang dapat dipilih terbatas pada format .step dan .iges, yang merupakan format umum dalam pertukaran data model 3D CAD. Pada pengujian ini, file yang digunakan adalah OPV MODEL.stp. Setelah file dipilih, program akan menampilkan informasi batas (bounding box) dari model, termasuk nilai koordinat minimum dan maksimum pada sumbu X, Y, dan Z. Informasi ini penting untuk validasi dimensi awal dan sebagai acuan dalam proses pemotongan blok.

```
Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + - _ □ ×

(base) C:\Users\User>conda activate kapalcadcam

(kapalcadcam) C:\Users\User>cd onedrive/ta/main

(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.

(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
[1] Loaded & scaled: C:/Users/User/OneDrive/TA/main/OPV MODEL.stp
[2] Bounds: X[0.0,1255.0] Y[-105.0,105.0] Z[-0.0,165.0]
[3] Closed solid built.
[4] Voxel candidates: 360
[5] Filtering interior voxels...
|
```

Gambar 5. 3 Proses running program I

Program kemudian menjalankan serangkaian proses solidifikasi hingga ekspor block ke dalam format .stl. seperti pada Gambar 5. 3. Program selanjutnya akan memberikan peringatan apabila file output sebelumnya sudah ada, dan secara otomatis menimpa file lama. Pada akhir proses, ditampilkan informasi jumlah entitas (entities) dalam file dan konfirmasi bahwa ekspor berhasil dilakukan. Dalam contoh Gambar 5. 4, file hasil bernama output_step/block_01.step dengan 6958 entitas berhasil diekspor. Keseluruhan file ekspor dapat dilihat pada Lampiran A.

```
Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + - _ □ ×

(kapalcadcam) C:\Users\User>cd onedrive/ta/main

(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.

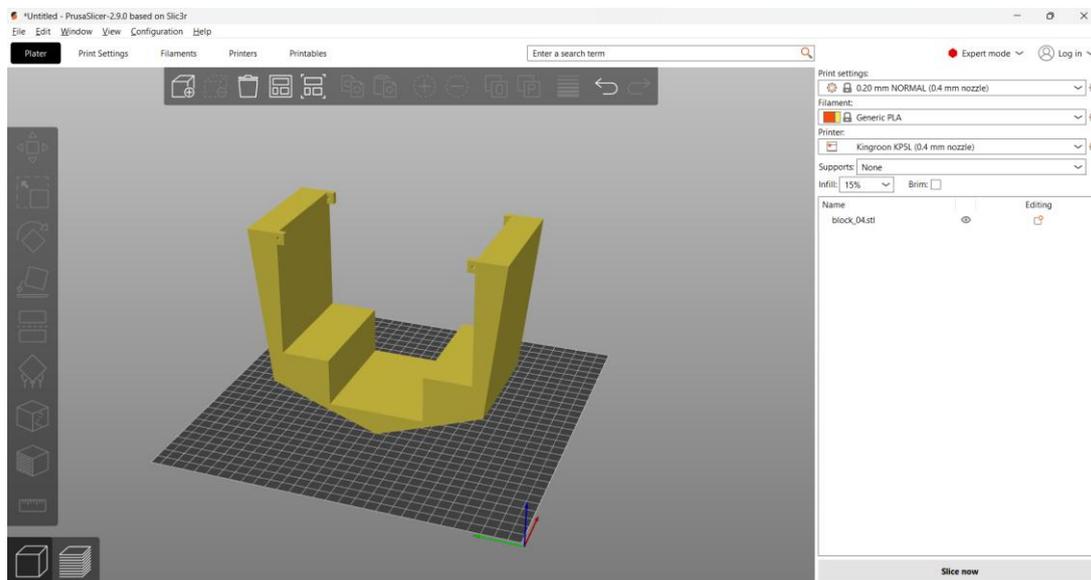
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
[1] Loaded & scaled: C:/Users/User/OneDrive/TA/main/OPV MODEL.stp
[2] Bounds: X[0.0,1255.0] Y[-105.0,105.0] Z[-0.0,165.0]
[3] Closed solid built.
[4] Voxel candidates: 360
[5] Filtering interior voxels...
    Valid voxels: 250
[5] Interior removed.
[6] Applying deck cutter...
[6] Deck cutter done.
[7] Embedding bolt joints...
[7] Bolt joints embedded.
[8] Slicing & exporting...
Warning: output_step\block_01.step file already exists and will be replaced

*****
*****      Statistics on Transfer (Write)      *****
*****
*****      Transfer Mode = 0 I.E. As Is      *****
*****      Transferring Shape, ShapeType = 0      *****
** WorkSession : Sending all data
Step File Name : output_step\block_01.step(6958 ents) Write Done
[8] Exported block_01
```

Gambar 5. 4 Output tampilan running program I

Selanjutnya, hasil file .stl diperiksa secara manual pada peninjau file .stl. Setelah proses ekspor selesai, file hasil dalam format .stl yang dihasilkan oleh program diperiksa secara

manual menggunakan perangkat lunak peninjau file .stl, seperti PrusaSlicer yang digambarkan pada Gambar 5. 5. Pemeriksaan ini bertujuan untuk memastikan bahwa seluruh fitur geometris telah terbentuk dengan benar, termasuk keutuhan bentuk lambung, kerapian pemotongan blok, serta keberadaan sambungan baut yang telah disisipkan secara otomatis oleh program.



Gambar 5. 5 Hasil running program I dalam format STL

Setelah dilakukan peninjauan terhadap seluruh blok hasil pemrosesan, evaluasi manual secara visual menunjukkan bahwa tidak terdapat kecacatan desain maupun indikasi ketidaksesuaian geometris yang dapat memengaruhi kualitas model. Setiap blok berhasil diekspor dalam kondisi tertutup sempurna, dengan tepi yang rapi dan sambungan baut yang tercetak jelas serta terletak pada posisi yang konsisten sesuai dengan rancangan.

5.1.2. Pengujian Program II

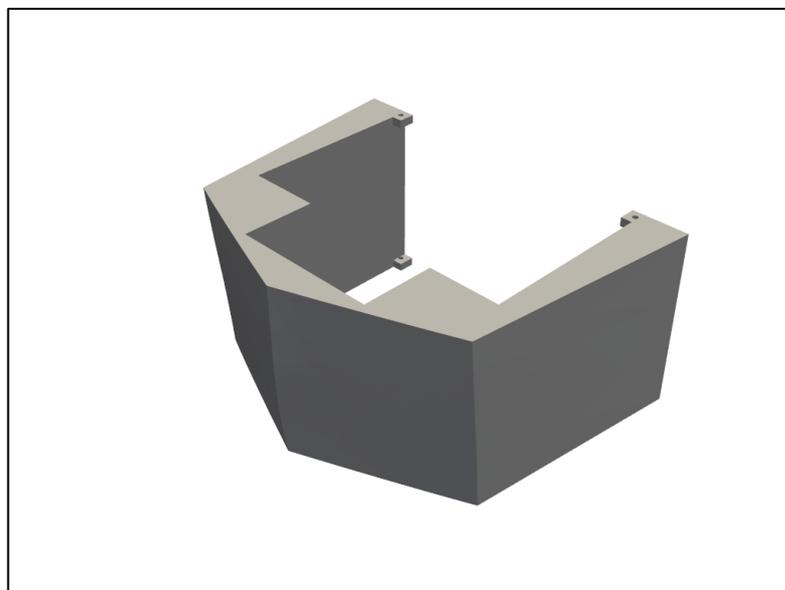
Setelah pengujian Program I selesai dilakukan dan menghasilkan file .stl dari model kapal yang telah dipotong menjadi beberapa blok, tahapan selanjutnya adalah melakukan pengujian terhadap Program II. Program ini bertanggung jawab untuk memproses file .stl yang telah dihasilkan sebelumnya, terutama dalam konteks visualisasi hasil pemotongan serta analisis lanjutan jika diperlukan.

Pada pengujian ini, tidak perlu memasukkan direktori file .stl secara manual. Hal ini dikarenakan Program II telah dirancang untuk secara otomatis membaca isi dari folder bernama output_stl, yang berada dalam direktori yang sama dengan program. Folder tersebut merupakan hasil keluaran langsung dari Program I dan menjadi input utama bagi Program II.

```
Anaconda Prompt (minicond) x Anaconda Prompt (minicond) x Anaconda Prompt (minicond) x Windows PowerShell x + -
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python slicev5.py
Rotating block_01.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_01_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_01_rotated.stl
Slicing block_01_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_01.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_01.gcode
Rendering preview of block_01_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_01.png
Rotating block_02.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_02_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_02_rotated.stl
Slicing block_02_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
```

Gambar 5. 6 Tampilan output program II

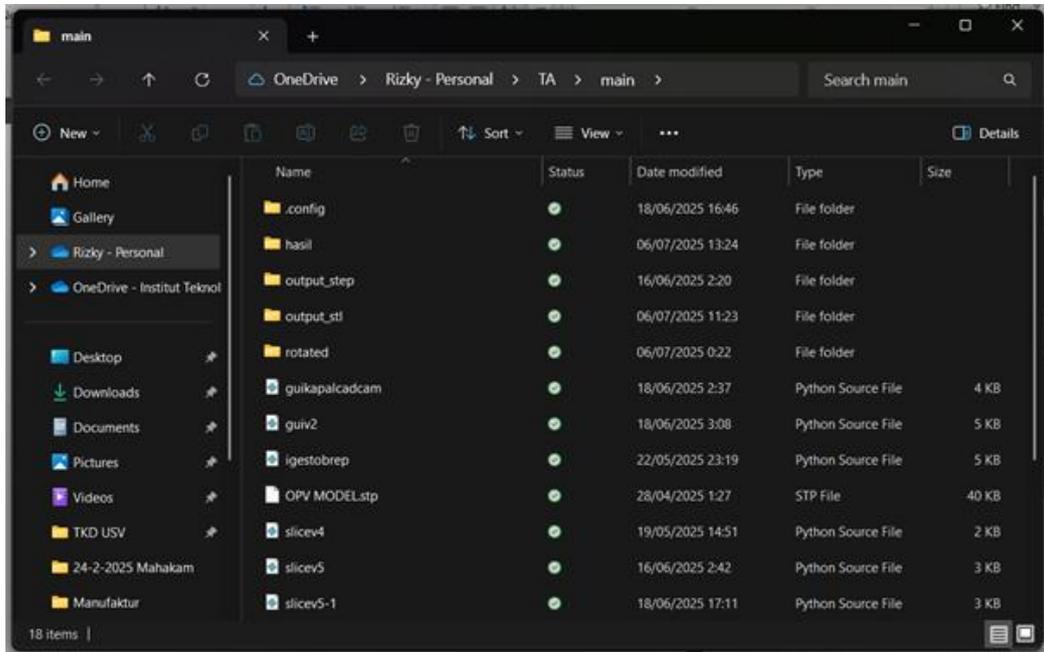
Pengujian dilakukan dengan menjalankan skrip Program II pada Anaconda Command Prompt yang sama, tanpa perlu keluar dari environment yang telah diaktifkan sebelumnya (kapalcadcam) yang digambarkan pada Gambar 5. 6. Dengan perintah *Python slicev5.py*, program dijalankan dan secara otomatis membaca seluruh file .stl yang tersedia di dalam folder output_stl.



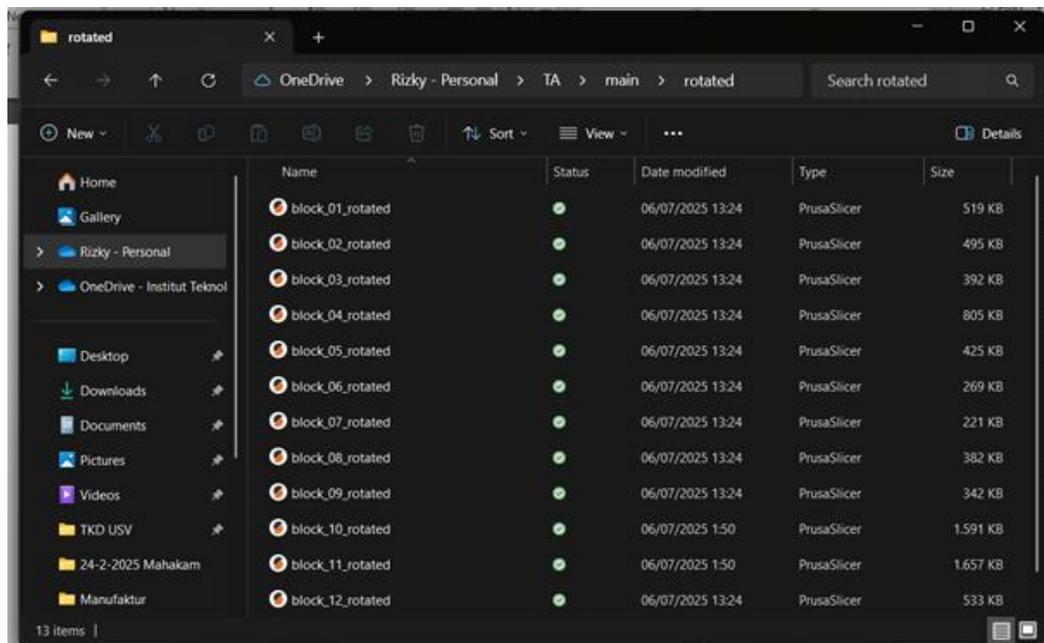
Gambar 5. 7 Output gambar program II

Selanjutnya, program menampilkan seluruh blok model secara 3D seperti pada Gambar 5. 7 menggunakan pustaka PyVista, dengan pewarnaan, irisan melintang, dan tepi yang ditampilkan sesuai dengan konfigurasi pengguna. Hasil visualisasi memberikan gambaran

menyeluruh terhadap posisi dan bentuk masing-masing blok, serta memungkinkan verifikasi ulang terhadap orientasi dan hasil slicing yang telah dilakukan.



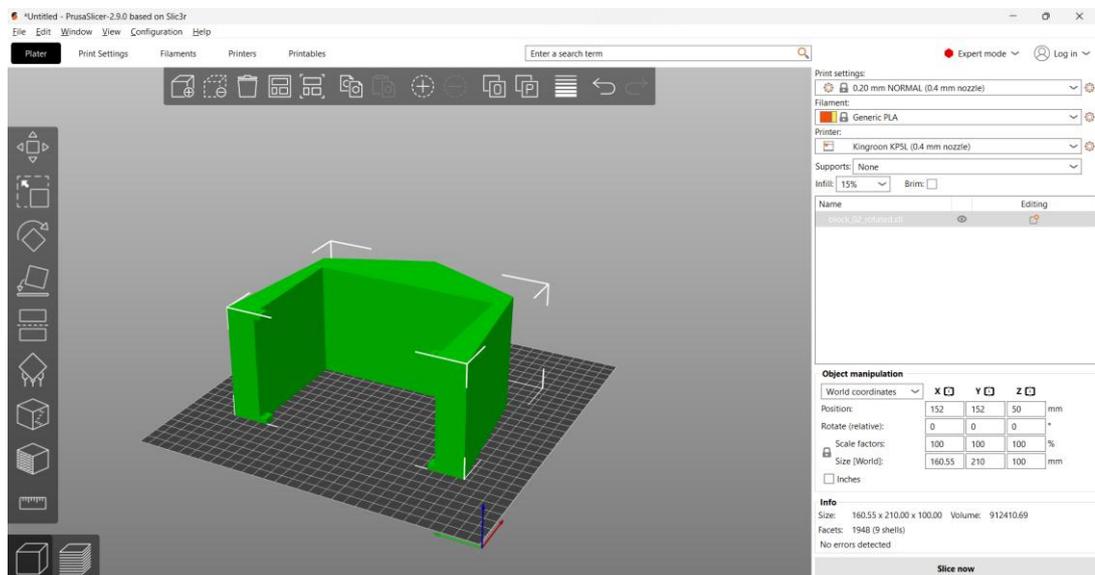
Gambar 5. 8 Folder output



Gambar 5. 9 File hasil output yang dirotasi

Setelah proses visualisasi dan pembacaan file .stl dari folder output_stl berhasil dilakukan, tahap selanjutnya adalah melakukan pengecekan manual lanjutan untuk memastikan bahwa setiap blok model benar-benar siap untuk dicetak. Pengecekan ini dilakukan dengan meninjau dua folder keluaran tambahan, yaitu folder rotated (Gambar 5. 9) dan folder hasil

(Gambar 5. 8). Folder rotated berisi file .stl yang telah mengalami proses rotasi orientasi cetak. Rotasi ini bertujuan untuk mengurangi kebutuhan struktur support, meningkatkan kekuatan lapisan cetakan pada arah memanjang kapal, serta memastikan bahwa pencetakan dapat dilakukan dalam posisi yang stabil. Setiap file dalam folder tersebut diperiksa menggunakan perangkat lunak peninjau model seperti PrusaSlicer seperti pada Gambar 5. 10 untuk memastikan bahwa hasil rotasi tidak menyebabkan deformasi bentuk dan bahwa orientasi sudah sesuai dengan kebutuhan pencetakan. Keseluruhan file ekspor program kedua dapat dilihat pada Lampiran B.



Gambar 5. 10 Pengecekan hasil rotasi

Sementara itu, folder bernama hasil merupakan hasil konversi otomatis dari file .stl menggunakan PrusaSlicer. File .gcode ini kemudian dibuka seperti pada Gambar 5. 11 dan dianalisis secara manual menggunakan fitur G-code Viewer pada PrusaSlicer seperti pada Gambar 5. 12 untuk memverifikasi bahwa jalur cetak telah terbentuk secara utuh, tidak terputus, dan lapisan-lapisan tercetak tersusun secara konsisten.

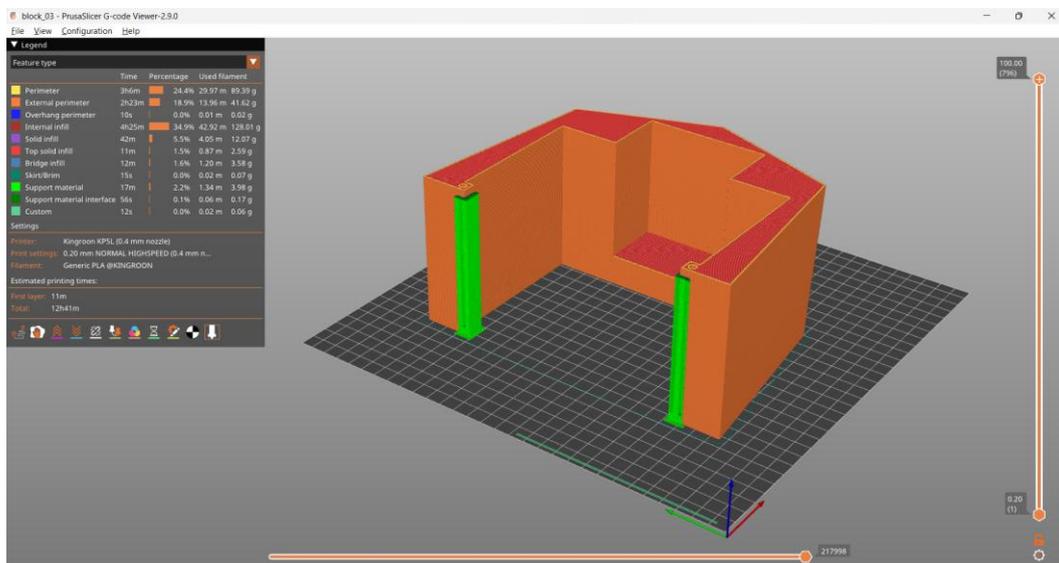
```

1 ; generated by PrusaSlicer 2.9.0 on 2025-06-21 at 11:07:28 UTC
2
3 ;
4
5 ; external perimeters extrusion width = 0.42mm
6 ; perimeters extrusion width = 0.45mm
7 ; infill extrusion width = 0.44mm
8 ; solid infill extrusion width = 0.42mm
9 ; top infill extrusion width = 0.42mm
10 ; support material extrusion width = 0.42mm
11 ; first layer extrusion width = 0.50mm
12
13
14 ; printing object block_01_rotated.stl id:0 copy 0
15 ; stop printing object block_01_rotated.stl id:0 copy 0
16
17 ;TYPE:Custom
18 G90 ; use absolute coordinates
19 M83 ; extruder relative mode
20 M140 S60 ; set final bed temp
21 M104 S150 ; set temporary nozzle temp to prevent oozing during homing and auto bed leveling
22 G4 S30 ; allow partial nozzle warmup
23 G28 ; home all axis
24 G29 ; auto bed levelling
25 G1 Z50 F240
26 G1 X7 Y10 F3000
27 M300 P300 S200 ; sound buzzer to say bed levelling is done
28 M300 P300 S0 ; pause sound so two beeps can play
29 M300 P300 S200 ; sound buzzer to say bed levelling is done
30 M104 S220 ; set final nozzle temp
31 M190 S60 ; wait for bed temp to stabilize
32 M109 S220 ; wait for nozzle temp to stabilize
33 G1 Z0.28 F240
34 G92 E0
35 G1 Y140 E10 F1500 ; prime the nozzle

```

Gambar 5. 11 Tampilan mentah file .gcode

Pemeriksaan ini juga mencakup verifikasi keberadaan struktur support jika diperlukan dan memastikan tidak ada kesalahan pemrosesan atau lintasan ekstrusi yang mencurigikan. Berdasarkan pengecekan manual terhadap kedua jenis file tersebut, dapat disimpulkan bahwa hasil keluaran program telah memenuhi kriteria kesiapan untuk tahap pencetakan fisik dengan printer 3D.

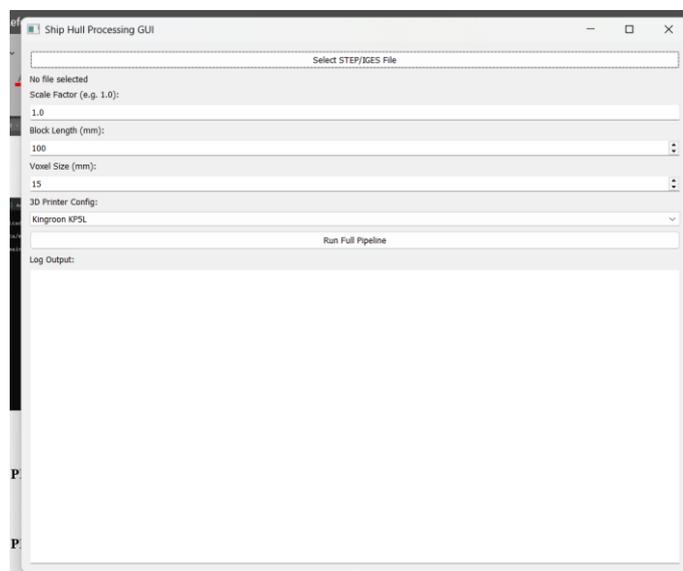


Gambar 5. 12 File .gcode dilihat dengan *software* bantuan slicer

Pengujian ini menunjukkan bahwa alur kerja otomatis dari Program I ke Program II berjalan dengan baik dan terintegrasi, meminimalkan kesalahan input dan mendukung workflow yang efisien dalam proses desain hingga produksi prototipe.

5.2. Pengujian Program Terintegrasi dengan GUI

Setelah dilakukan pengujian terhadap kedua program utama secara terpisah, tahap selanjutnya adalah menguji kinerja program dalam bentuk terintegrasi melalui antarmuka grafis pengguna (Graphical User Interface atau GUI) yang telah dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa seluruh fungsi dari Program I dan Program II dapat dijalankan secara berurutan dalam satu alur kerja terpadu yang ramah pengguna. Pengujian dilakukan dengan cara menjalankan file utama GUI berbasis PyQt5 melalui Anaconda Command Prompt, menggunakan environment yang sama (kapalcadcam) tanpa perlu menjalankan skrip program satu per satu secara manual.



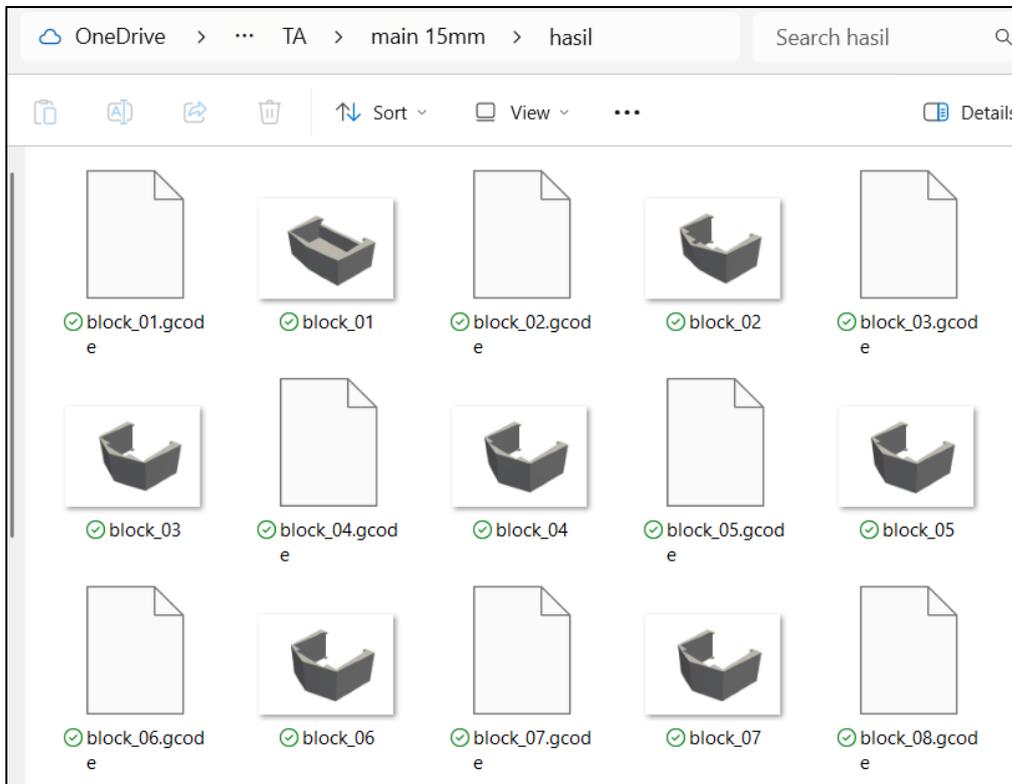
Gambar 5. 13 Tampilan awal GUI

Pada tampilan awal GUI (Gambar 5. 13), pengguna diminta untuk memilih file input dalam format `.step` atau `.iges`, serta mengisi parameter konfigurasi seperti skala model (`SCALE_FACTOR`), panjang blok (`BLOCK_LEN`), ukuran voxel (`VOXEL_SIZE`), dan konfigurasi pencetakan 3D yang digunakan. Setelah seluruh parameter dimasukkan, pengguna cukup menekan tombol “Run Full Pipeline” untuk memulai proses secara otomatis. Program kemudian menjalankan proses solidifikasi, filterisasi voxel, pemotongan dek, penyisipan sambungan baut, slicing model ke dalam blok `.stl`, hingga visualisasi hasil dengan menggunakan pustaka `PyVista`—semuanya dilakukan tanpa campur tangan langsung di terminal atau penulisan perintah manual. Output dari Log Output dapat dilihat pada Gambar 5. 14.

```
Log Output:
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Rendering preview of block_12_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_12.png
Rotating block_13.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl
Slicing block_13_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Rendering preview of block_13_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_13.png
All done.
```

Gambar 5. 14 Output akhir tampilan GUI

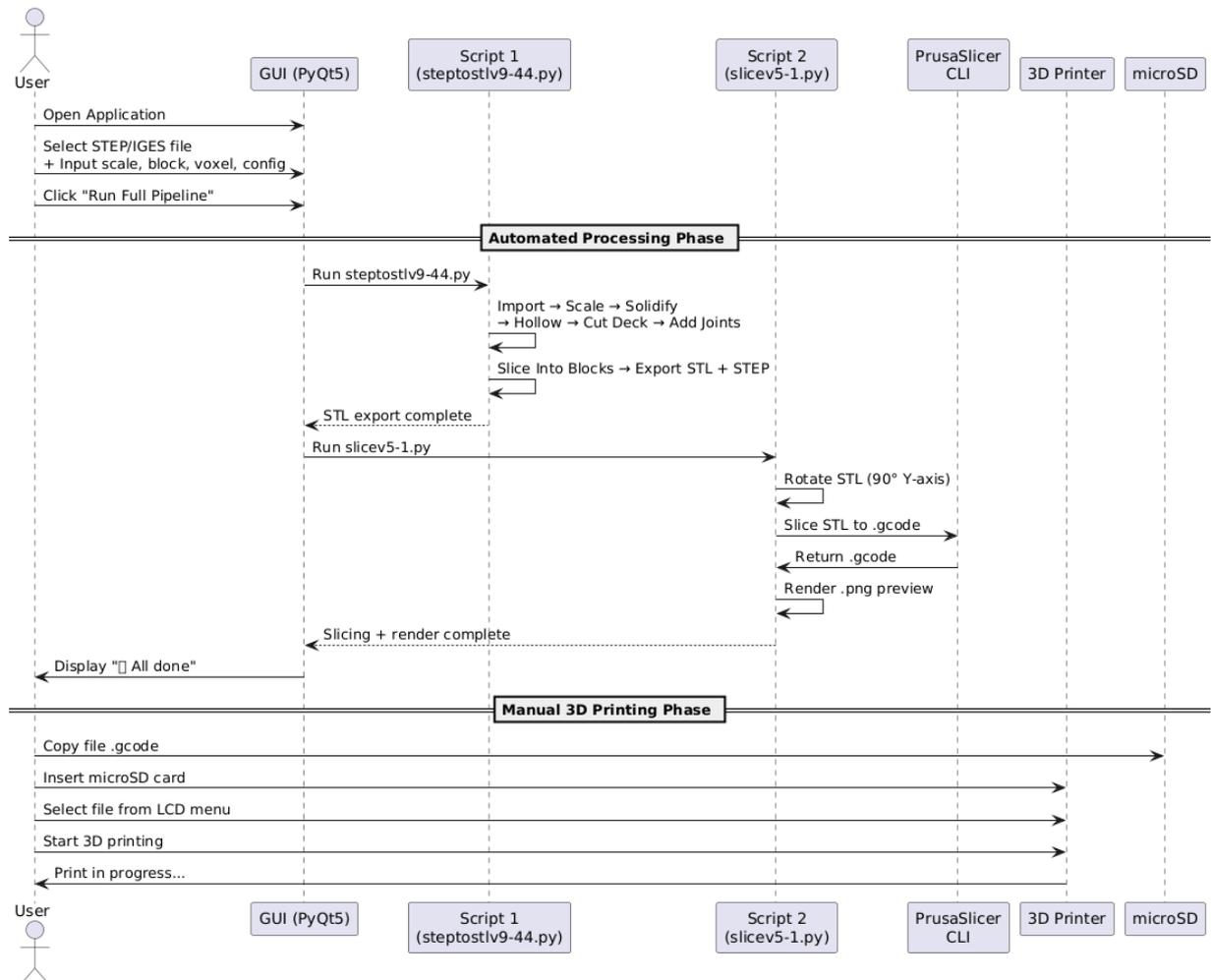
Selama pengujian, tampilan log pada GUI menampilkan progres proses secara real-time, memberikan informasi yang berguna kepada pengguna mengenai tahapan yang sedang berlangsung, waktu eksekusi, serta status keberhasilan atau kesalahan jika terjadi. Setelah proses selesai, GUI memberikan notifikasi bahwa semua langkah telah berhasil dijalankan, dan hasil dapat dicek pada folder yang telah ditentukan secara otomatis seperti pada Gambar 5. 15. Keseluruhan file ekspor pada program ketiga dapat dilihat pada Lampiran C dan Lampiran D.



Gambar 5. 15 Output akhir pemrosesan program pada folder

Pengujian ini menunjukkan bahwa integrasi antara backend (logika program) dan frontend (antarmuka GUI) telah berjalan dengan baik, dengan responsivitas dan alur kerja yang efisien. Dengan demikian, program terintegrasi ini berhasil menyederhanakan proses konversi, pemotongan, dan persiapan cetak model kapal menjadi lebih praktis, bahkan untuk pengguna yang tidak memiliki latar belakang teknis pemrograman.

5.3. Diagram Aktivitas User



Gambar 5. 16 Diagram Aktivitas User pada Alur Otomatisasi Terprogram

Diagram aktivitas pada Gambar 5. 16 menggambarkan keseluruhan alur interaksi antara pengguna (user) dan sistem, mulai dari tahap input awal hingga proses pencetakan 3D. Diagram ini disusun berdasarkan pendekatan swimlane untuk memisahkan setiap komponen yang terlibat, yaitu antarmuka pengguna (GUI), dua skrip utama (*Script 1* dan *Script 2*), PrusaSlicer CLI, perangkat printer 3D, dan media transfer file berupa microSD.

Fase pertama, yaitu Automated Processing Phase, dimulai ketika pengguna membuka aplikasi dan memasukkan file desain berformat .STEP atau .IGES, serta parameter konfigurasi

seperti skala, panjang blok, ukuran voxel, dan opsi pemotongan dek. Setelah tombol "Run Full Pipeline" ditekan, seluruh proses berikutnya berlangsung secara otomatis tanpa memerlukan intervensi manual. Proses ini mencakup konversi file menjadi bentuk solid, filterisasi voxel untuk membentuk struktur rongga, pemotongan model menjadi blok-blok modular, penambahan fitur sambungan baut, serta ekspor file hasil dalam format .STL, .STEP, .gcode, dan .png untuk keperluan visualisasi pratinjau.

Seluruh rangkaian tersebut dieksekusi secara terintegrasi antara *Script 1* (steptostlv9-44.py), *Script 2* (slicev5-1.py), dan PrusaSlicer CLI. Pengguna tidak perlu membuka file satu per satu, menavigasi ulang direktori, atau melakukan konversi secara manual di *software* terpisah. Sistem secara otomatis mengatur orientasi file STL, menjalankan slicing berdasarkan konfigurasi, dan mengembalikan file .gcode beserta pratinjau dalam bentuk .png.

Setelah semua proses selesai, sistem menampilkan notifikasi "All done", menandai bahwa seluruh tahapan digital telah selesai diproses. Selanjutnya, pengguna hanya perlu melanjutkan ke Manual *3D Printing* Phase, yang mencakup pemindahan file .gcode ke microSD, pemilihan file dari layar printer, dan memulai proses pencetakan fisik.

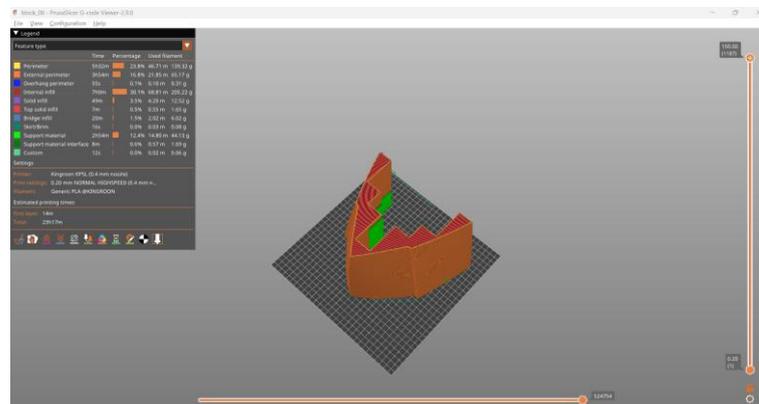
Diagram ini menunjukkan bahwa seluruh proses desain hingga siap cetak dapat dilakukan dengan satu kali klik di awal, dan seluruh pipeline pemrosesan berjalan secara otomatis. Otomatisasi ini secara signifikan mengurangi beban kerja manual, meminimalkan potensi kesalahan pengguna, serta meningkatkan efisiensi dan konsistensi dalam proses manufaktur model lambung kapal.

5.4. Pengujian Program pada Model Lambung Lainnya

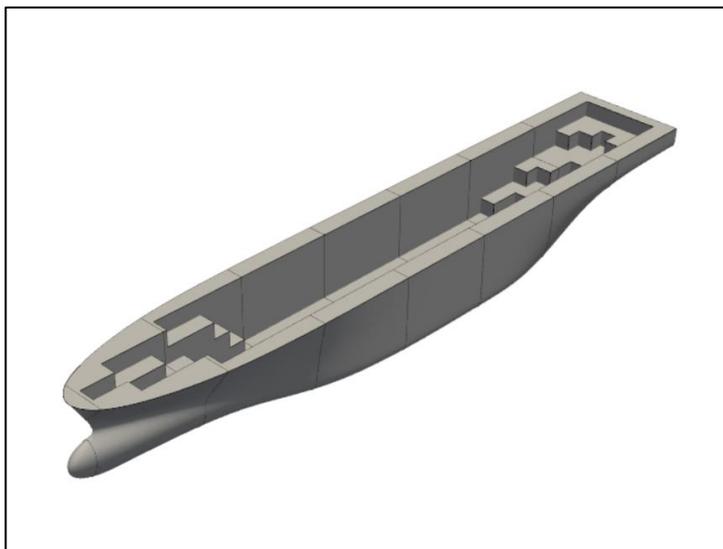
Pengujian lanjutan dilakukan terhadap dua model lambung kapal yang memiliki karakteristik geometris berbeda untuk mengevaluasi kemampuan adaptasi program terhadap variasi bentuk lambung. Model pertama merupakan lambung tipe planning yang digambarkan pada Gambar 5. 17, yang dirancang untuk kecepatan tinggi dan mengandalkan gaya angkat hidrodinamik. Sementara itu, model kedua (Gambar 5. 19) merupakan lambung tipe displacement dengan bulbous bow.



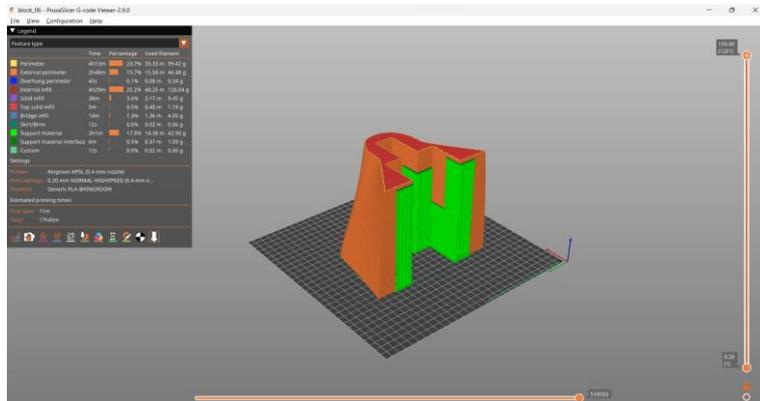
Gambar 5. 17 Pemrosesan lambung planning



Gambar 5. 18 Tampilan G-Code variasi lambung planning hasil pemrosesan program otomatisasi



Gambar 5. 19 Pemrosesan lambung displasemen



Gambar 5. 20 Tampilan G-Code variasi lambung displasemen hasil pemrosesan program otomatisasi

Kedua model diproses melalui alur program yang sama, mulai dari konversi file desain menjadi bentuk solid, proses hollowing menggunakan voxel, pembangkitan sambungan baut secara otomatis, hingga proses pemotongan menjadi blok-blok modular yang siap cetak. Meskipun bentuk kedua lambung sangat berbeda secara geometris, hasil pengujian pada Gambar 5. 18 dan Gambar 5. 20 menunjukkan bahwa seluruh proses berjalan dengan baik dan konsisten.

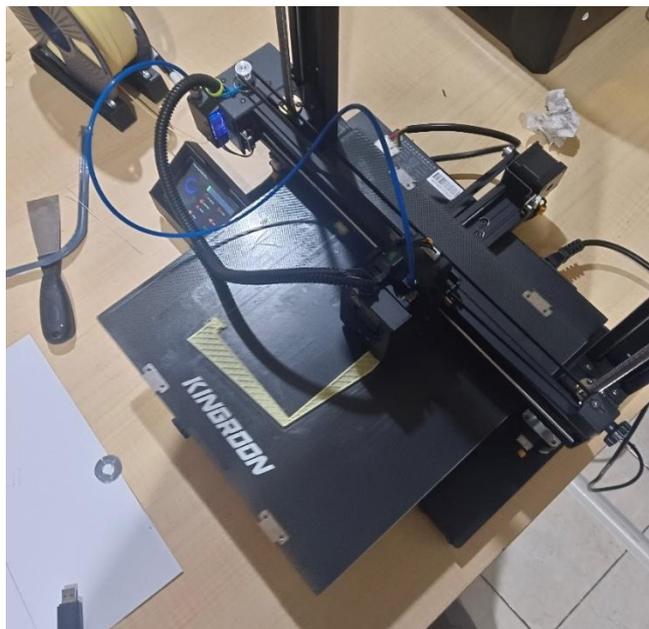
5.5. Pencetakan dan Perakitan Hasil Fisik

Setelah program berhasil menghasilkan file .gcode untuk setiap blok model, proses selanjutnya adalah mencetak hasil fisik menggunakan mesin pencetak 3D. File .gcode yang telah disiapkan sebelumnya dipindahkan dari komputer ke media penyimpanan eksternal berupa microSD card menggunakan adaptor USB. Setelah proses pemindahan selesai, microSD card kemudian dilepas dari komputer dan dipasang ke slot yang tersedia pada perangkat printer 3D.

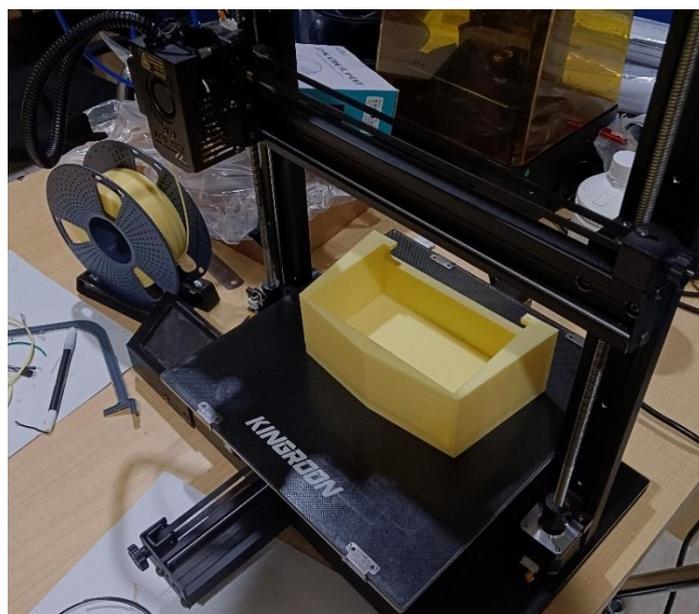


Gambar 5. 21 Pemilihan file .gcode di 3D print

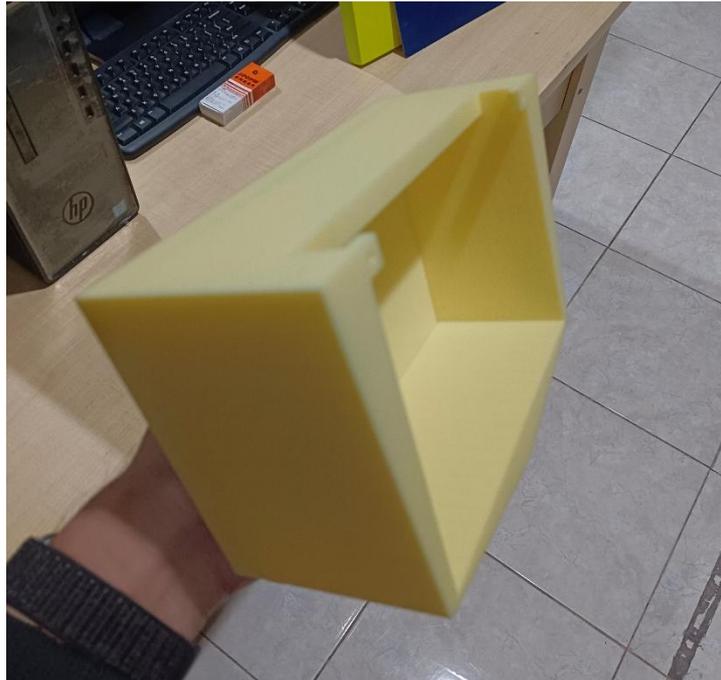
Setelah microSD card terpasang dengan baik pada mesin printer, langkah selanjutnya adalah memilih file yang akan dicetak melalui antarmuka pengguna pada layar printer seperti pada Gambar 5. 21. Proses pencetakan kemudian dimulai secara otomatis sesuai jalur yang telah ditentukan oleh file .gcode seperti pada Gambar 5. 22. Setelah satu blok selesai dicetak (Gambar 5. 23), hasil cetakan diperiksa secara visual untuk memastikan tidak terdapat cacat cetak seperti lapisan yang bergeser, kekosongan material, atau kegagalan ekstrusi. Apabila hasil cetak dinyatakan baik, proses dilanjutkan dengan pencetakan file .gcode berikutnya hingga seluruh blok selesai diproduksi.



Gambar 5. 22 Proses pencetakan

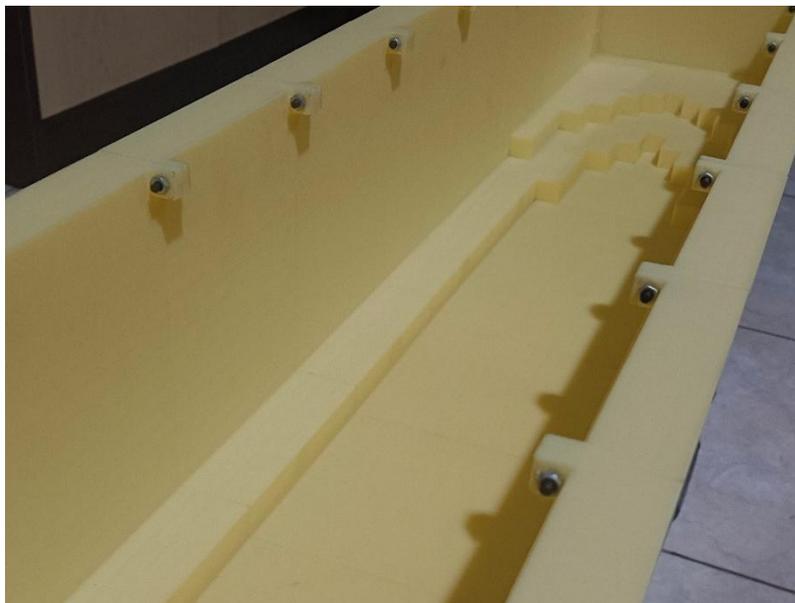


Gambar 5. 23 Proses 3D Printing Block 1 selesai



Gambar 5. 24 Hasil jadi block 1 transom

Setelah seluruh blok berhasil dicetak, tahapan akhir adalah perakitan model fisik. Setiap blok yang telah dicetak dirangkai satu per satu dan disambungkan menggunakan baut L dan mur M3 (Gambar 5. 25), sesuai dengan desain lubang sambungan yang telah disiapkan secara otomatis oleh program sebelumnya.



Gambar 5. 25 Pemasangan baut untuk sambungan per block

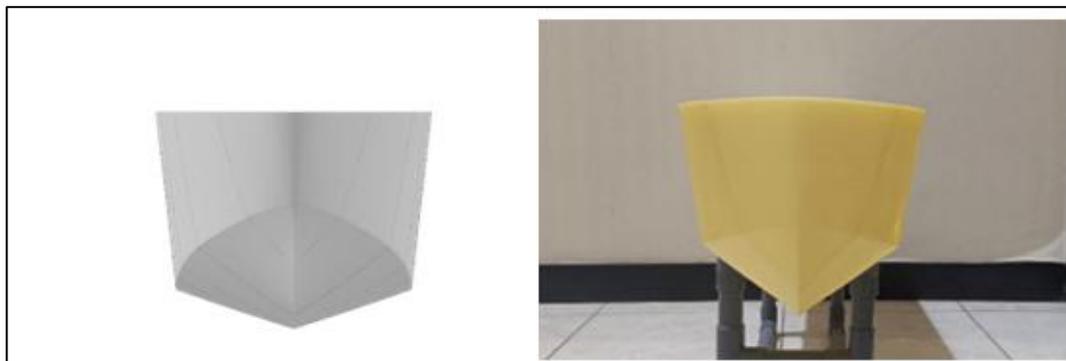
Proses perakitan ini memastikan bahwa seluruh bagian model menyatu dengan kokoh dan membentuk geometri utuh dari lambung kapal seperti yang telah dirancang dalam tahap digital. Hasil akhir dari proses perakitan dapat dilihat pada Gambar 5. 26.



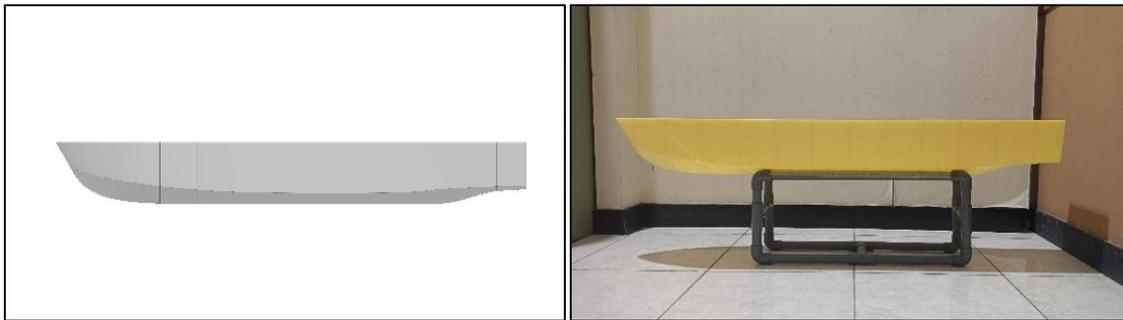
Gambar 5. 26 Hasil akhir setelah dirakit

5.6. Perbandingan Desain dan Hasil Fisik

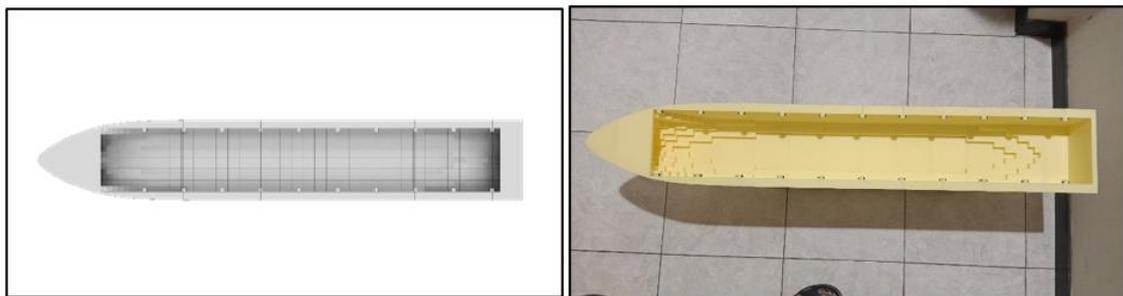
Untuk memastikan bahwa proses konversi dari desain digital ke bentuk fisik berjalan dengan akurat, dilakukan perbandingan antara model hasil desain CAD dan model hasil cetak 3D dari beberapa sudut pandang, yaitu tampak depan (Gambar 5. 27), tampak samping (Gambar 5. 28), dan tampak atas (Gambar 5. 29). Gambar sebelah kiri menunjukkan representasi desain digital, sedangkan gambar sebelah kanan memperlihatkan hasil cetak fisik yang telah dirakit.



Gambar 5. 27 Tampak depan desain dan hasil fisik



Gambar 5. 28 Tampak Samping desain dan hasil fisik



Gambar 5. 29 Tampak atas desain dan hasil fisik

Dari segi visual, tidak terlihat adanya perbedaan bentuk yang signifikan antara desain dan hasil fisik. Kontur lambung, garis tengah, dan profil bentuk keseluruhan berhasil dipertahankan dengan baik. Hasil cetak menunjukkan kualitas reproduksi geometri yang tinggi, bahkan pada bagian haluan dan buritan yang umumnya rentan terhadap distorsi akibat rotasi cetak atau keterbatasan printer.

Tabel 5. 1 Perhitungan simpangan desain dan hasil fisik

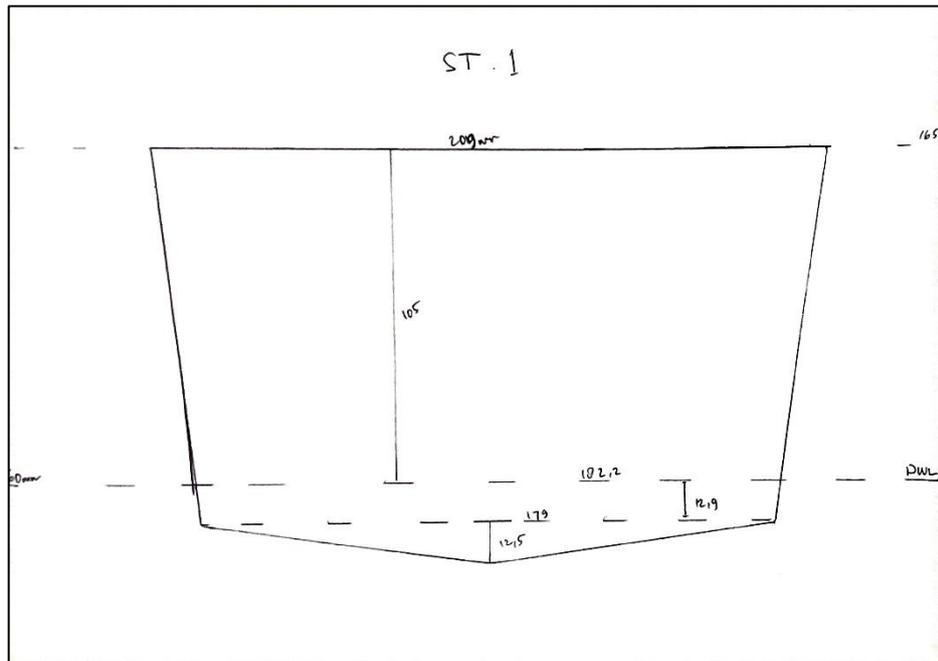
	DESIGN	3D PRINT	SIMPANGAN
LOA (mm)	1255	1254	0,08%
LWL (mm)	1192	1191	0,08%
B (mm)	210	209	0,48%
H (mm)	165	164	0,61%

Selain evaluasi visual, dilakukan pula perbandingan kuantitatif terhadap dimensi utama, yaitu Length Overall (LOA), Length Waterline (LWL), Beam (B), dan Height (H). Berdasarkan pengukuran dan perhitungan simpangan pada Tabel 5. 1, diperoleh simpangan dimensi relatif yang sangat kecil, berkisar antara 0,08% hingga 0,61%.

5.6.1. Perhitungan Simpangan Hidrostatik Setinggi Sarat Kapal

Untuk meningkatkan ketelitian dalam membandingkan hasil desain digital dengan hasil cetak fisik, dilakukan perhitungan luas penampang melintang (area station) dan koefisien penampang (midship coefficient) pada masing-masing station. Pengukuran dilakukan pada

model hasil cetak 3D dengan cara menjiplak langsung bentuk penampang setiap station pada kertas, kemudian dilakukan perhitungan luas secara manual menggunakan pendekatan bentuk geometri sederhana seperti trapesium dan segitiga (Tabel 5. 2) yang contohnya dapat dilihat pada Gambar 5. 30.



Gambar 5. 30 Perhitungan lebar dan luas station pada WL

Tabel 5. 2 Perhitungan luasan pada station 1

STATION 1			
Bentuk	Komponen	Panjang (mm)	Luas (mm ²)
Trapezium 1	a	182,2	2329,7
	b	179	
	t	12,9	
Segitiga	a	179	1118,8
	t	12,5	
Luas penampang station			3448,5

Tabel 5. 3 Data luas station dan lebar pada sarat

3D Print			Design		
Sta	A	Breadth WL	Sta	A	Breadth WL
0	0,0	0	0	0,0	0
1	3448,5	182,1	1	3433,3	182,3
2	6001,9	183,1	2	6138,3	183,3
3	7601,4	184,8	3	7752,7	185
4	8209,7	185,6	4	8170,7	185,8
5	8250,8	185,7	5	8220,6	185,9
6	8070,4	185,7	6	8221,8	185,9
7	8201,4	185,6	7	8185,3	185,8
8	8060,6	185,1	8	8039,2	185,3
9	7618,5	182,1	9	7585,1	182,3
10	6435,3	173,0	10	6427,0	173,2
11	3711,4	141,4	11	3695,1	141,6
12	144,9	21,8	12	144,3	22

Contoh perhitungan disajikan pada Tabel 5. 2 Station 1, di mana luas total penampang hasil cetak adalah 3433,3 mm², terdiri dari kombinasi trapesium dan segitiga yang membentuk profil melintang lambung. Nilai ini kemudian dibandingkan dengan data desain digital dari *software CAD* yang menunjukkan luas sebesar 3355,9 mm² (Tabel 5. 3), menghasilkan deviasi sebesar 2,26% yang dihitung pada Tabel 5. 4. Proses yang sama dilakukan pada seluruh station untuk memperoleh nilai area aktual dan koefisien penampang hasil fisik.

Tabel 5. 4 Hasil perhitungan simpangan luas per station

SIMPANGAN LUAS PENAMPANG PER STATION			
Stations	Luas Station design (mm2)	Luas Station 3D print (mm2)	Deviasi
0	0,0	0,0	0,00%
1	3433,3	3355,9	2,26%
2	6138,3	6011,9	2,06%
3	7752,7	7602,4	1,94%
4	8170,7	8044,8	1,54%
5	8220,6	8034,5	2,26%
6	8221,8	8069,4	1,85%
7	8185,3	8045,7	1,71%
8	8039,2	7922,5	1,45%
9	7585,1	7496,4	1,17%
10	6427,0	6295,7	2,04%
11	3695,1	3623,4	1,94%
12	144,3	141,6	1,87%

Perbandingan luas penampang menunjukkan bahwa mayoritas station memiliki deviasi kurang dari 2%, dengan deviasi maksimum tercatat sebesar 2,26% pada Station 2. Deviasi ini

masih berada dalam batas toleransi yang wajar untuk pencetakan 3D Print, mengingat adanya penyusutan material dan variasi toleransi pada printer 3D. Hasil ini menunjukkan bahwa bentuk penampang dari model fisik secara umum telah sesuai dengan bentuk pada desain digital.

Tabel 5. 5 Hasil perhitungan simpangan koefisien penampang per station

SIMPANGAN KOEFISIEN PENAMPANG PER STATION			
Stations	Cm Station Design	Cm Station 3D print	Deviasi
0	0,0	0,0	0,00%
1	0,3	0,3	0,55%
2	0,6	0,5	2,11%
3	0,7	0,7	1,85%
4	0,7	0,7	0,59%
5	0,7	0,7	0,48%
6	0,7	0,7	1,74%
7	0,7	0,7	0,30%
8	0,7	0,7	0,37%
9	0,7	0,7	0,55%
10	0,6	0,6	0,25%
11	0,4	0,4	0,58%
12	0,1	0,1	1,36%

Selain luas, dilakukan pula perbandingan terhadap koefisien penampang (Cm) per station, yang dihitung dari rasio antara luas penampang dengan hasil kali panjang dan lebar di waterline. Cm merupakan parameter penting dalam evaluasi bentuk lambung dari sisi hidrodinamika. Hasil perbandingan pada Tabel 5. 5 menunjukkan bahwa nilai Cm dari hasil cetak memiliki deviasi antara 0,57% hingga 2,15%, yang sekali lagi menunjukkan bahwa sistem pencetakan dan pembagian blok tidak menyebabkan penyimpangan bentuk yang signifikan.

Selain perhitungan Cm, dilakukan pula estimasi nilai volume displacement (Vdisp) dari model hasil cetak menggunakan metode luasan penampang seperti pada Tabel 5. 6. Nilai LCB dan Vdisp ini penting dalam verifikasi apakah bentuk hasil cetak tetap menjaga sifat hidrostatis dasar lambung kapal sebagaimana desain awal, dan menjadi indikator bahwa sistem pencetakan dan pemotongan blok tidak menyebabkan distorsi bentuk atau perpindahan titik daya apung yang signifikan.

Tabel 5. 6 Hasil perhitungan hidrostatis

3D Print					
Sta	A	FS	A*FS	Lengan Momen	A*FS*LM
0	0,0	1	0,00	0	0
1	3448,5	4	13793,96	100	1379396
2	6001,9	2	12003,86	200	2400772
3	7601,4	4	30405,56	300	9121668
4	8209,7	2	16419,38	400	6567752
5	8250,8	4	33003,24	500	16501621
6	8070,4	2	16140,74	600	9684444
7	8201,4	4	32805,40	700	22963780
8	8060,6	2	16121,22	800	12896978
9	7618,5	4	30474,04	900	27426640
10	6435,3	2	12870,62	1000	12870620
11	3711,4	4	14845,51	1100	16330056
12	144,9	1	144,94	1200	173923
		$\Sigma(A*FS)$	229028,47	$\Sigma(A*FS*LM)$	138317648,9

Perhitungan nilai hasil kali antara luas penampang (A) dengan pengali simpson (FS), serta hasil kali antara A, FS, dan jarak longitudinal masing-masing station (LM), dilakukan sebagai bagian dari estimasi posisi pusat daya apung (longitudinal center of buoyancy/LCB) dari model hasil cetak. Dalam hal ini, pengali simpson (FS) merepresentasikan kontribusi relatif tiap station terhadap keseluruhan volume, dan jarak LM diukur dari titik acuan (biasanya haluan kapal). Nilai-nilai hasil perkalian ini kemudian dijumlahkan untuk mendapatkan total momen terhadap titik acuan. Dari hasil rekapitulasi, diperoleh total penjumlahan $A \times FS$ sebesar 229.084,27, dan total penjumlahan $A \times FS \times LM$ sebesar 13.331.645,49. Posisi LCB kemudian dihitung dari pembagian antara total momen ($A \times FS \times LM$) dengan total $A \times FS$.

Tabel 5. 7 Hasil perhitungan V dan LCB pada hasil fisik

3D Print		
Komponen	Nilai	Satuan
Volume Displasemen	7634282,31	mm ³
LCB from AP	581,59	mm

Dengan hasil perhitungan pada Tabel 5. 6 dan Tabel 5. 7, diperoleh posisi LCB sebesar 581,95 mm dari titik acuan. Ini menunjukkan bahwa pusat daya apung longitudinal model cetak berada di lokasi yang wajar dan tidak mengalami pergeseran signifikan akibat proses segmentasi dan pencetakan. Dan selanjutnya dilakukan perhitungan simpangan dan ditemukan sebagai berikut:

Tabel 5. 8 Hasil Perhitungan Simpangan setinggi T

	DESIGN	3D PRINT	SIMPANGAN
LOA (mm)	1255	1254	0,08%
LWL (mm)	1192	1191	0,08%
B (mm)	210	209	0,48%
H (mm)	165	164	0,61%
Volume Displasemen (mm ³)	7687425,80	7634282,31	0,69%
LCB from AP	576,10	581,59	0,95%

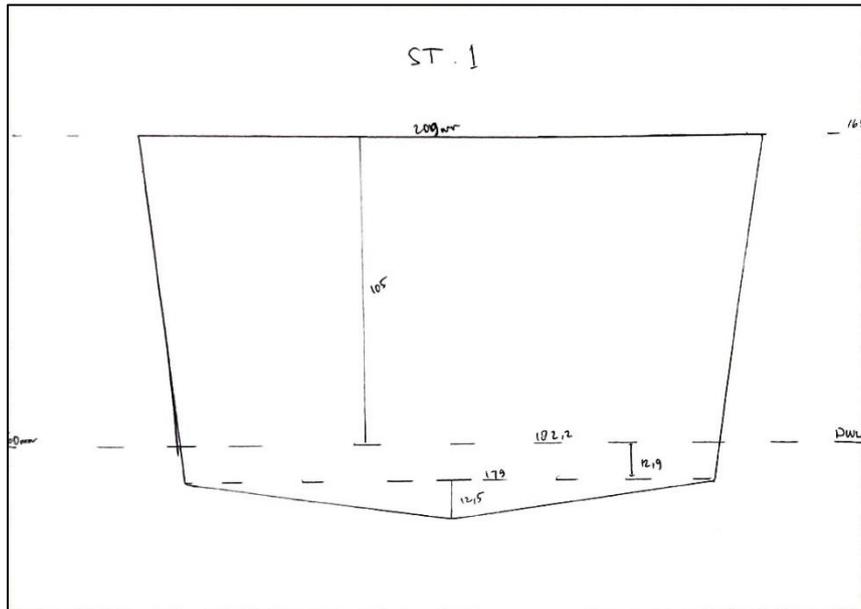
Hasil perhitungan pada Tabel 5. 8 menunjukkan bahwa model hasil cetak memiliki tingkat akurasi geometris yang sangat baik jika dibandingkan dengan desain awal. Panjang keseluruhan kapal (LOA) pada hasil cetak berbeda hanya sebesar 0,08% dari desain, panjang garis air (LWL) juga menyimpang 0,08%, lebar maksimum (B) memiliki simpangan 0,48%, dan tinggi (H) berbeda sebesar 0,61%. Nilai ini menunjukkan bahwa perbedaan dimensi yang terjadi relatif sangat kecil dan tidak memengaruhi bentuk keseluruhan model secara signifikan.

Dari sisi parameter hidrostatis, *Volume Displacement* hasil cetak tercatat sebesar 7.634.282,31 mm³, berbeda 0,69% dari nilai desain sebesar 7.687.425,80 mm³. Selain itu, posisi pusat gaya apung longitudinal (LCB) hasil cetak berada pada 581,59 mm dari titik acuan, dengan simpangan sebesar 0,95% dibandingkan dengan posisi LCB desain yang berada di 576,10 mm. Simpangan ini masih berada dalam rentang toleransi yang dapat diterima untuk model uji hidrodinamika.

Hasil ini membuktikan bahwa proses segmentasi, pencetakan 3D, dan perakitan blok model tidak menimbulkan perubahan signifikan pada bentuk maupun karakteristik hidrostatis utama dari desain aslinya. Dengan demikian, model hasil cetak layak digunakan untuk pengujian hidrodinamika dan evaluasi desain lebih lanjut tanpa memerlukan koreksi geometri tambahan.

5.6.2. Perhitungan Simpangan Hidrostatis Setinggi Tinggi Kapal

Untuk meningkatkan ketelitian dalam membandingkan hasil desain digital dengan hasil cetak fisik, dilakukan perhitungan luas penampang melintang (*area station*) dan koefisien penampang (*midship coefficient*) pada masing-masing station setinggi tinggi kapal. Pengukuran dilakukan pada model hasil cetak 3D dengan cara menjiplak langsung bentuk penampang setiap station pada kertas seperti pada Gambar 5. 31, kemudian dilakukan perhitungan luas secara manual menggunakan pendekatan bentuk geometri sederhana seperti trapesium dan segitiga.



Gambar 5. 31 Perhitungan Lebar dan Luas Station pada Sta. 1

Tabel 5. 9 Perhitungan Luasan setinggi H

STATION 1			
Bentuk	Komponen	Panjang (mm)	Luas (mm ²)
Trapezium 1	a	209	22872,6
	b	179	
	t	117,9	
Segitiga	a	179	1118,8
	t	12,5	
Luas penampang station			23991,4

Tabel 5. 10 Data Luasan Tiap Station setinggi H

3D Print (H)			Design (H)		
Sta	A	Breadth WL	Sta	A	Breadth WL
0	0,0	0	0	0,0	0
1	23991,4	209,9	1	24029,5	210
2	26733,7	209,9	2	26784,5	210
3	28446,8	209,9	3	28489,8	210
4	28890,5	209,9	4	28951,3	210
5	28938,0	209,9	5	29006,2	210
6	28948,8	209,9	6	29007,3	210
7	28918,6	209,9	7	28965,0	210
8	28723,0	209,9	8	28794,4	210
9	28024,8	208,1	9	28082,6	208,2
10	26125,3	203	10	26182,3	203,1
11	20873,6	182,3	11	20919,0	182,4
12	8904,6	117,8	12	8923,0	117,9

Hasil perhitungan luasan manual pada Tabel 5. 9 menunjukkan luas total penampang sebesar 23991,4 mm² yang terdiri atas gabungan bentuk trapesium dan segitiga yang membentuk profil melintang lambung. Nilai ini kemudian dibandingkan dengan data desain digital dari perangkat lunak CAD pada Tabel 5. 11 yang menunjukkan luas sebesar 24029,5 mm², sehingga didapatkan deviasi sebesar 0,16%.

Tabel 5. 11 Simpangan Luasan per Station setinggi H

SIMPANGAN LUAS PENAMPANG PER STATION			
Stations	Luas Station design (mm ²)	Luas Station 3D print (mm ²)	Deviasi
0	0,0	0,0	0,00%
1	24029,5	23991,4	0,16%
2	26784,5	26733,7	0,19%
3	28489,8	28446,8	0,15%
4	28951,3	28890,5	0,21%
5	29006,2	28938,0	0,24%
6	29007,3	28948,8	0,20%
7	28965,0	28918,6	0,16%
8	28794,4	28723,0	0,25%
9	28082,6	28024,8	0,21%
10	26182,3	26125,3	0,22%
11	20919,0	20873,6	0,22%
12	8923,0	8904,6	0,21%

Perhitungan serupa dilakukan pada seluruh station untuk memperoleh nilai luas penampang aktual hasil cetak. Tabel 5. 11 diatas menyajikan rekapitulasi hasil perhitungan luas penampang pada setiap station serta deviasi dibandingkan data desain digital. Hasil perbandingan menunjukkan bahwa mayoritas station memiliki deviasi yang kecil, dengan nilai deviasi maksimum sebesar 0,26% pada Station 10. Deviasi ini berada dalam batas toleransi yang wajar, mengingat adanya penyusutan material selama pencetakan serta toleransi dimensi printer 3D.

Selain itu, dilakukan pula perbandingan terhadap nilai koefisien penampang (C_m) per station pada Tabel 5. 12 yang dihitung dari rasio antara luas penampang dengan hasil kali panjang dan lebar di *waterline*. Parameter C_m merupakan salah satu indikator penting dalam mengevaluasi kesesuaian bentuk lambung dari sisi hidrodinamika. Hasil analisis pada tabel diatas menunjukkan bahwa nilai C_m hasil cetak memiliki deviasi kecil, dengan nilai maksimum sebesar 0,16%. Hal ini mengindikasikan bahwa pembagian blok dan proses pencetakan tidak menyebabkan penyimpangan bentuk melintang yang signifikan.

Tabel 5. 12 Simpangan Koefisien Penampang per Station setinggi H

SIMPANGAN KOEFISIEN PENAMPANG PER STATION			
Stations	Cm Station Design	Cm Station 3D print	Deviasi
0	0,0	0,0	0,00%
1	0,7	0,7	0,11%
2	0,8	0,8	0,14%
3	0,8	0,8	0,10%
4	0,8	0,8	0,16%
5	0,8	0,8	0,19%
6	0,8	0,8	0,15%
7	0,8	0,8	0,11%
8	0,8	0,8	0,20%
9	0,8	0,8	0,16%
10	0,8	0,8	0,17%
11	0,7	0,7	0,16%
12	0,5	0,5	0,12%

Tabel 5. 13 Hasil Perhitungan Hidrostatik Setinggi H

3D Print (H)					
Sta	A	FS	A*FS	Lengan Momen	A*FS*LM
0	0,0	1	0,00	0	0
1	23991,4	4	95965,40	100	9596540
2	26733,7	2	53467,40	200	10693480
3	28446,8	4	113787,20	300	34136160
4	28890,5	2	57781,00	400	23112400
5	28938,0	4	115752,00	500	57876000
6	28948,8	2	57897,60	600	34738560
7	28918,6	4	115674,40	700	80972080
8	28723,0	2	57446,00	800	45956800
9	28024,8	4	112099,20	900	100889280
10	26125,3	2	52250,60	1000	52250600
11	20873,6	4	83494,40	1100	91843840
12	8904,6	1	8904,60	1200	10685520
		$\Sigma(A*FS)$	924519,80	$\Sigma(A*FS*LM)$	552751260

Selain menghitung koefisien Cm, dilakukan pula estimasi nilai *Volume Displacement* (Vdisp) serta posisi pusat daya apung longitudinal (*Longitudinal Center of Buoyancy* atau LCB). Nilai ini dihitung dengan menggunakan hasil kali antara luas penampang (A) dengan pengali Simpson (FS), serta hasil kali antara A, FS, dan jarak longitudinal tiap station (LM) dari titik acuan. Pengali Simpson digunakan untuk memberikan bobot yang sesuai pada masing-masing station dalam integrasi numerik volume lambung, sedangkan LM digunakan untuk menghitung momen longitudinal total. Dari hasil rekapitulasi pada Tabel 5. 14, diperoleh total $\Sigma(A \times FS)$ sebesar 924.519,80 dan total $\Sigma(A \times FS \times LM)$ sebesar 5.527.512,60. Posisi LCB

kemudian dihitung dari pembagian antara kedua nilai tersebut sehingga didapatkan nilai LCB sebesar 575,54 mm dari titik acuan.

Tabel 5. 14 Hasil perhitungan Volume Displasemen dan LCB AP

3D Print (H)		
Komponen	Nilai	Satuan
Volume Displasemen	30817326,67	mm ³
LCB from AP	575,54	mm

Hasil perbandingan akhir pada Tabel 5. 15 menunjukkan bahwa model hasil cetak memiliki tingkat akurasi geometris yang sangat baik terhadap desain awal. Panjang keseluruhan (LOA) hasil cetak berbeda hanya 0,08%, lebar maksimum (B) memiliki simpangan 0,48%, dan tinggi (H) berbeda sebesar 0,61%. Dari sisi hidrostatis, *Volume Displacement* hasil cetak sebesar 30817326,67 mm³ memiliki simpangan 1,88% terhadap nilai desain sebesar 31408980,09 mm³, dan posisi LCB hanya berbeda 0,01% dari desain.

Tabel 5. 15 Hasil perhitungan simpangan setinggi H

	DESIGN	3D PRINT	SIMPANGAN
LOA (mm)	1255	1254	0,08%
B (mm)	210	209	0,48%
H (mm)	165	164	0,61%
Volume Displasemen (mm ³)	31408609,00	30817326,67	1,88%
LCB from AP	575,60	575,54	0,01%

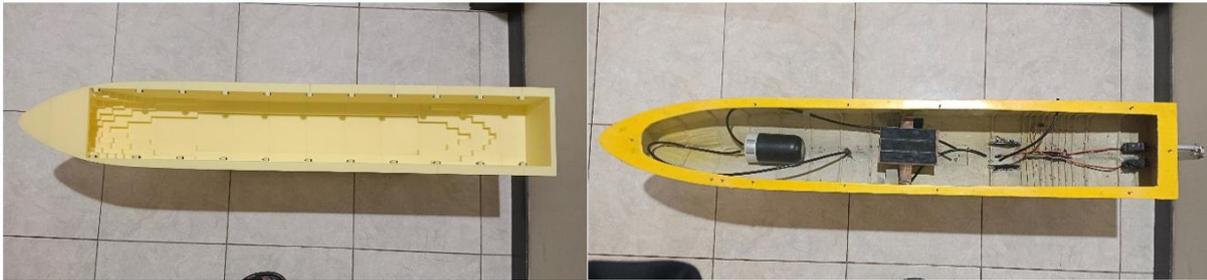
Keseluruhan hasil ini membuktikan bahwa proses segmentasi, pencetakan 3D, dan perakitan blok model tidak menyebabkan perubahan signifikan pada bentuk maupun sifat hidrostatis utama dari desain aslinya. Dengan demikian, model cetak yang dihasilkan layak digunakan untuk uji hidrodinamika dan evaluasi desain tanpa memerlukan koreksi tambahan.

5.7. Perbandingan Model Kapal *Fiberglass* dan Hasil Fisik

Sebagai bentuk validasi terhadap kualitas hasil cetak 3D, dilakukan perbandingan langsung antara model kapal hasil cetak 3D dengan model kapal sejenis yang telah dibuat menggunakan metode konvensional berbasis resin *fiberglass*. Gambar-gambar berikut menunjukkan model hasil cetak 3D yang disandingkan secara langsung dengan model *fiberglass* dari berbagai sudut: tampak samping (Gambar 5. 32), dan tampak atas (Gambar 5. 33). Dari segi visual, tidak terlihat adanya perbedaan bentuk yang signifikan antara desain dan hasil fisik. Kontur lambung, garis tengah, dan profil bentuk keseluruhan berhasil dipertahankan dengan baik.



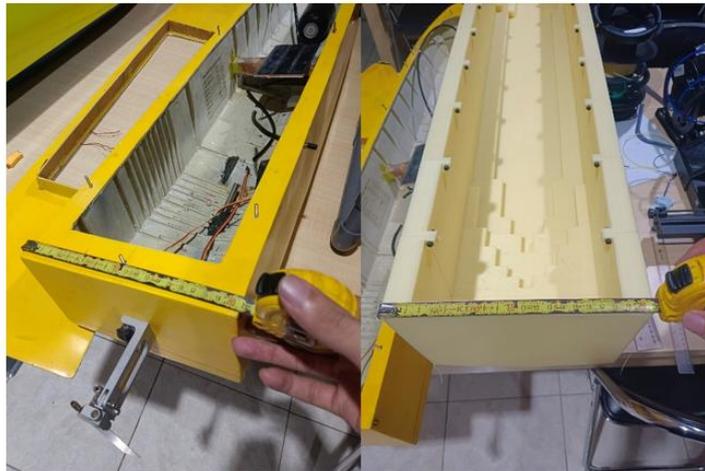
Gambar 5. 32 Tampak samping model kapal fiber dan hasil fisik



Gambar 5. 33 Tampak atas model kapal fiber dan hasil fisik



Gambar 5. 34 Pengukuran panjang model kapal fiber dan hasil fisik



Gambar 5. 35 Pengukuran lebar model kapal fiber dan hasil fisik

Analisis kuantitatif juga dilakukan terhadap dimensi utama, yaitu Length Overall (LOA) pada Gambar 5. 34, Length Waterline (LWL), Beam (B) pada Gambar 5. 35, dan Height (H), untuk mengevaluasi kesesuaian ukuran antara model *fiberglass* dan hasil cetak 3D.

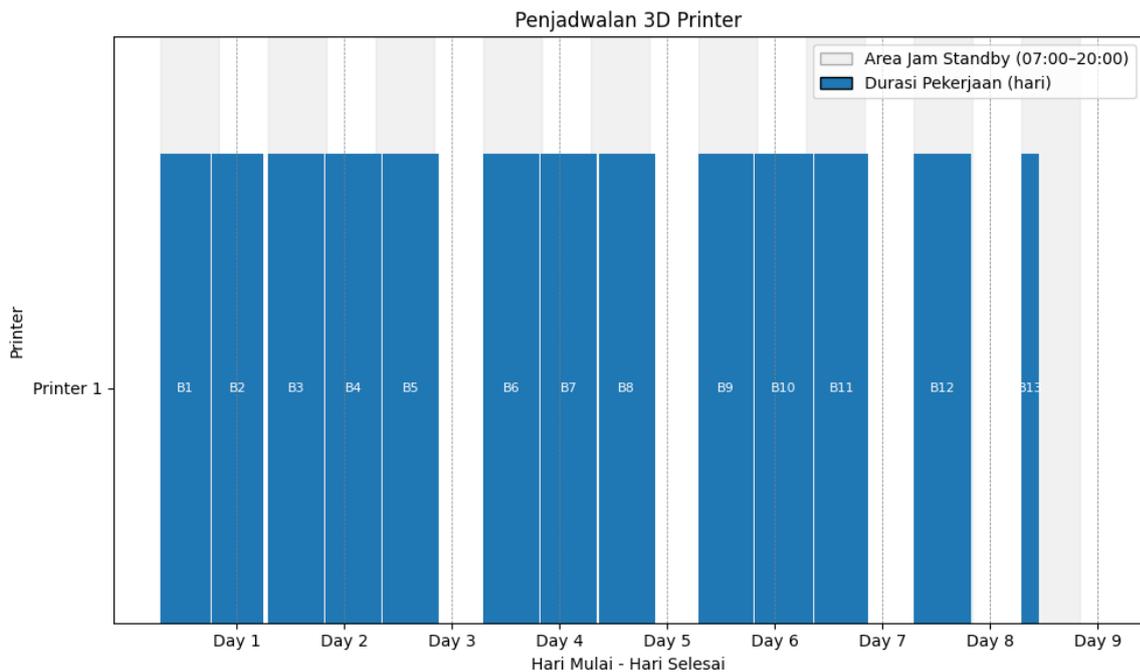
Tabel 5. 16 Perhitungan simpangan model kapal fiber dan hasil fisik

	RESIN FIBER	3D PRINT	SIMPANGAN
LOA	1253	1254	-0,080%
LWL	1192	1190	0,168%
B	208	208	0,000%
H	164	164	0,000%

Berdasarkan hasil pengukuran pada Tabel 5. 16, ditemukan bahwa simpangan antar keduanya berada dalam rentang yang sangat kecil dan dapat diterima. LOA menunjukkan selisih sebesar 0,08%, sedangkan LWL memiliki simpangan sebesar 0,168%. Sementara itu, untuk dimensi Beam dan Height, tidak ditemukan perbedaan sama sekali, dengan nilai simpangan sebesar 0,000%. Hasil ini menunjukkan bahwa model hasil cetak 3D mampu mereplikasi dimensi geometris model *fiberglass* dengan sangat akurat, bahkan dalam konteks manufaktur bebas cetakan yang sepenuhnya bergantung pada file digital dan proses otomatisasi.

5.8. Analisis Biaya Produksi

Berdasarkan hasil pengujian dan visualisasi penjadwalan pencetakan pada Gambar 5. 36 diketahui bahwa proses pencetakan seluruh blok (B1 hingga B13) menggunakan satu printer membutuhkan waktu sekitar 8 hari penuh dengan sistem kerja bertahap dan berurutan dengan spare waktu sebesar 20 menit.



Gambar 5. 36 Penjadwalan operasional 3D Printer

Dari sisi material dan biaya pada Tabel 5. 17, Tabel 5. 18 estimasi mencatat bahwa total waktu pencetakan adalah 9124 menit (atau 144 jam), dengan total berat filamen mencapai 3,4 kg. Mengingat harga filament PLA sebesar Rp155.000 per kilogram dan biaya tambahan seperti lem dan dempul sebesar Rp200.000, maka total biaya produksi dengan metode *3D Printing* ini adalah sekitar Rp820.000. Nilai ini jauh lebih rendah dibandingkan dengan metode FRP konvensional yang berkisar antara Rp5.000.000 hingga Rp15.000.000, bahkan tanpa mempertimbangkan tambahan waktu dan kebutuhan tenaga kerja terampil untuk proses laminasi *fiberglass*.

Tabel 5. 17 Lama pekerjaan dan perhitungan biaya

Block	Lama Pekerjaan (menit)	Lama Pekerjaan (Jam)	Berat filamen (g)
1	674	11:14	294,38
2	681	11:21	255,74
3	752	12:32	280,85
4	750	12:30	280,49
5	753	12:33	283,71
6	754	12:34	283,98
7	753	12:33	283,73
8	752	12:32	282,16
9	738	12:18	274,52
10	776	12:56	272,79
11	725	12:05	245,56
12	777	12:57	286,57
13	239	3:59	78,71
Total Menit	9124	Total berat (kg)	3,4
Pembulatan Berat (kg)			4
Harga PLA per 1kg			Rp 155.000,00
Total Cost			Rp 620.000,00

Tabel 5. 18 Analisa produksi model kapal

	3D Printing Automated	3D Printing Manual	Resin Fiber
Biaya Awal	Menengah (Harga 3D Print Kingroon KP5L Rp. 3.700.000)	Menengah (Harga 3D Print Kingroon KP5L Rp. 3.700.000)	Rendah
Biaya Proses Model (Software CAD, CAM dsb)	Rp 0 (PythonOCC dan PrusaSlicer open-source)	Rp 250.000 (Solidworks, estimasi lisensi per proyek)	-
Biaya Material	4 kg x Rp. 155.000,- per kg = Rp.620.000	4 kg x Rp. 155.000,- per kg = Rp.620.000	-
Biaya Tambahan (Lem, Dempul, Dextone, dsb)	Rp. 200.000	Rp. 200.000	-
Biaya Tenaga Kerja	±8 jam × Rp25.000 = Rp200.000 1 orang teknisi	3 x ±8 jam x Rp25.000 = Rp600.000 2 orang (1 teknisi ahli CAD dan 1 Teknisi 3D print)	Tinggi
Total Biaya Produksi	Rp 1.020.000	Rp 1.670.000	Rp 5.000.000 - 15.000.000
Total Waktu Produksi	Kurang lebih 11 hari(8 hari printing + 3 hari post-processing)	Kurang lebih 12 hari(1 hari CAD Processing+ 8 hari printing + 3 hari post-processing)	2 – 3 minggu
	*1 orang teknisi ahli CAD dengan waktu pekerjaan maksimal 2 hari pada rentang jam 07-15		
	**1 orang teknisi 3D print dengan estimasi hanya intervensi disaat 3D print selesai pada jam 07-20		

Analisis biaya produksi ini membandingkan tiga metode pembuatan model lambung kapal, yaitu 3D Printing Automated, 3D Printing Manual, dan Resin Fiber sebagai metode konvensional. Komponen yang dianalisis meliputi biaya awal, biaya proses model, biaya material, biaya tambahan, biaya tenaga kerja, total biaya produksi, dan total waktu produksi.

Pada komponen biaya awal, baik metode 3D Printing Automated maupun 3D Printing Manual menggunakan printer 3D Kingroon KP5L dengan harga Rp3.700.000. Metode resin fiber memiliki biaya awal yang relatif lebih rendah karena hanya memerlukan peralatan dasar seperti cetakan dan alat laminasi, meskipun membutuhkan keterampilan operator yang lebih tinggi. Biaya proses model pada metode otomatisasi adalah Rp0 karena menggunakan perangkat lunak open-source *PythonOCC* dan *PrusaSlicer* yang tidak membutuhkan biaya lisensi. Pada metode manual, diasumsikan menggunakan perangkat lunak berlisensi seperti *AutoCAD* atau *SolidWorks* dengan estimasi Rp250.000 per proyek. Metode resin fiber juga diasumsikan menggunakan perangkat lunak komersial dengan kisaran biaya yang sama.

Biaya material pada kedua metode 3D printing sama, yaitu menggunakan filament PLA seberat 4 kg dengan harga Rp155.000 per kg, sehingga total biaya material adalah Rp620.000. Pada metode resin fiber, biaya material tidak dihitung secara spesifik pada tabel karena

bergantung pada jenis resin dan serat yang digunakan, namun umumnya lebih mahal dan memerlukan bahan tambahan seperti katalis. Biaya tambahan meliputi lem, dempul, dan bahan pendukung lainnya dengan estimasi Rp200.000 untuk kedua metode 3D printing.

Perbedaan mencolok juga terlihat pada biaya tenaga kerja. Pada metode otomatisasi, keseluruhan proses dapat ditangani oleh satu teknisi dengan waktu kerja sekitar 8 jam sehingga total biaya tenaga kerja sebanyak Rp200.000 (dengan asumsi upah Rp25.000 per jam). Metode manual membutuhkan dua teknisi (teknisi CAD dan teknisi pencetak 3D) dengan waktu kerja masing-masing 8 jam, sehingga total biaya tenaga kerja meningkat menjadi Rp600.000. Pada metode resin fiber, proses pengerjaan jauh lebih kompleks dan memerlukan banyak tenaga kerja, sehingga biaya tenaga kerja dikategorikan tinggi.

Hasil keseluruhan perhitungan menunjukkan bahwa total biaya produksi metode 3D Printing Automated adalah Rp1.020.000, sementara metode 3D Printing Manual membutuhkan Rp1.670.000. Metode resin fiber jauh lebih mahal dengan kisaran Rp5.000.000–Rp15.000.000. Dari sisi waktu produksi, metode otomatisasi memerlukan 11 hari yang terdiri dari 8 hari proses pencetakan dan 3 hari post-processing. Metode manual memerlukan waktu sedikit lebih lama yaitu 12 hari karena ada tambahan 1 hari untuk pemrosesan CAD secara manual, selain 8 hari pencetakan dan 3 hari post-processing. Metode resin fiber memiliki waktu produksi paling lama, yaitu 2–3 minggu, karena harus membuat cetakan dan melakukan proses laminasi manual. Hasil analisis ini menunjukkan bahwa metode 3D Printing Automated memiliki keunggulan signifikan dibandingkan metode lainnya, dengan biaya lebih rendah, waktu produksi lebih singkat, serta kebutuhan tenaga kerja yang lebih sedikit.

BAB 6

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem otomatisasi dalam tugas akhir ini, dapat disimpulkan beberapa hal sebagai berikut:

1. Telah dikembangkan sistem otomatisasi berbasis Python yang mengintegrasikan alur kerja dari file desain lambung kapal (.step/.iges) menjadi file siap cetak (.gcode). Sistem ini mencakup konversi dan solidifikasi model, *slicing* otomatis, serta antarmuka pengguna berbasis PyQt5, sehingga mengurangi ketergantungan pada proses manual seperti pemotongan blok, pembuatan sambungan, dan konversi file. Hasil pengujian menunjukkan sistem mampu menghasilkan file .STL dan .gcode siap pakai serta memudahkan penggunaan bahkan bagi pengguna tanpa latar belakang pemrograman.
2. Hasil cetak fisik model lambung dari program otomatisasi menunjukkan tingkat akurasi geometris yang tinggi. Simpangan dimensi terhadap desain digital berkisar antara 0,08% hingga 0,95%, sedangkan perbandingan dengan model *fiberglass* menunjukkan kesesuaian dimensi hingga 0,00% pada lebar dan tinggi.
3. Analisis biaya produksi menunjukkan bahwa metode 3D Printing Automated memberikan penghematan biaya dan waktu dibandingkan metode 3D Printing Manual maupun metode konvensional berbahan Resin Fiber. Proses otomatisasi menghilangkan kebutuhan tahapan manual seperti pembagian blok, orientasi cetak, dan *slicing*, sehingga mengurangi keterlibatan teknisi tambahan serta mempercepat proses. Walaupun kebutuhan material dan biaya tambahan sama, total biaya produksi metode otomatisasi hanya sekitar Rp 1.020.000, lebih rendah dari metode manual sebesar Rp 1.670.000 dan jauh di bawah biaya pembuatan model resin fiber yang berkisar Rp 5.000.000–15.000.000. Dengan demikian, penerapan otomatisasi terbukti lebih efisien baik dari sisi biaya maupun waktu tanpa mengorbankan kualitas hasil cetak.

6.2. Saran

Berdasarkan hasil yang telah dicapai dan potensi pengembangan di masa depan, beberapa saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Perlu dilakukan pengembangan lebih lanjut pada antarmuka pengguna (GUI) untuk menambahkan fungsionalitas yang lebih canggih, seperti fitur pratinjau (preview) 3D interaktif dari model yang akan dicetak, serta penambahan opsi parameter slicing yang lebih detail agar pengguna memiliki kontrol lebih besar tanpa harus mengkonfigurasi file .ini secara manual.
2. Sistem dapat diperluas untuk mendukung jenis geometri kapal yang lebih kompleks, seperti kapal multihull (katamaran atau trimaran) atau kapal dengan struktur appendages yang rumit. Hal ini akan memerlukan pengembangan pada algoritma pemotongan blok dan penambahan sambungan agar lebih adaptif terhadap berbagai bentuk lambung.
3. Tahapan after-processing seperti pengeleman, pendempulan, dan perapian permukaan perlu direncanakan agar prototipe kokoh dan siap diuji agar performanya menyerupai bentuk akhir secara optimal.
4. Penelitian selanjutnya dapat difokuskan pada analisis performa hidrodinamika dari model hasil cetak 3D. Dengan melakukan uji *towing tank* pada prototipe yang dihasilkan, dapat divalidasi apakah kualitas permukaan dan akurasi bentuk dari hasil cetak 3D sudah memenuhi standar untuk pengujian hidrodinamika yang andal.

DAFTAR PUSTAKA

- Ahtiainen, H., Kallio, M., Airaksinen, M., & Holmström, A.-R. (2019). Safety, Time and Cost Evaluation of Automated and Semi-Automated Drug Distribution Systems in Hospitals: A Systematic Review. *European Journal of Hospital Pharmacy*, 27(5), 253–262. <https://doi.org/10.1136/ejhpharm-2018-001791>
- Alharbi, S. S. (2024). Gear Up for Development: The Automation Advantage for Sustainability in Manufacturing in the Kingdom of Saudi Arabia. *Sustainability*, 16(11), 4386. <https://doi.org/10.3390/su16114386>
- Autor, D. (2015). Why Are There Still So Many Jobs? The History and Future of Workplace Automation. *Journal of Economic Perspectives*, 29(3), 3–30. <https://doi.org/10.1257/jep.29.3.3>
- Banović, M., Mykhaskiv, O., Auriemma, S., Walther, A., Legrand, H., & Müller, J. (2018). Algorithmic Differentiation of the Open CASCADE Technology CAD Kernel and Its Coupling With an Adjoint CFD Solver. *Optimization Methods and Software*, 33(4–6), 813–828. <https://doi.org/10.1080/10556788.2018.1431235>
- Bhargavi, S. M. B., Suma, V., & Arjunan, R. V. (2024). Impact of Critical Success Factors on Productivity Gain During Automation Testing. *Journal of Wireless Mobile Networks Ubiquitous Computing and Dependable Applications*, 15(3), 92–108. <https://doi.org/10.58346/jowua.2024.i3.007>
- BIG. (2024). *SIPULAU - BIG*. <https://sipulau.big.go.id/news/11>
- Booyesen, G., Af, V. D. M., & Beer, D. d. (2019). Additive Manufacturing for Sustainable Custom-Designed Implants. *The South African Journal of Industrial Engineering*, 30(3). <https://doi.org/10.7166/30-3-2266>
- Choi, L., Choi, M., Kwon, S., Youn, D., & Song, G. (2024). Modular production of small ship models using 3D printing for model tests. *Ocean Engineering*, 302(December 2023), 117685. <https://doi.org/10.1016/j.oceaneng.2024.117685>
- Dvořák, F., Micali, M., & Mathieug, M. (2018). Planning and Scheduling in Additive Manufacturing. *Inteligencia Artificial*, 21(62), 40–52. <https://doi.org/10.4114/intartif.vol21iss62pp40-52>
- Ferro, C., Varetti, S., & Maggiore, P. (2023). Experimental Evaluation of Fatigue Strength of AlSi10Mg Lattice Structures Fabricated by AM. *Aerospace*, 10(5), 400.

<https://doi.org/10.3390/aerospace10050400>

- Gan, G. (2017). *An Introduction to Excel VBA Programming: with Applications in Finance and Insurance*.
- Huynh, V. C., & Tran, G. T. (2020). Improving the Accuracy of Ship Resistance Prediction Using Computational Fluid Dynamics Tool. *International Journal on Advanced Science Engineering and Information Technology*, 10(1), 171–177. <https://doi.org/10.18517/ijaseit.10.1.10588>
- ITTC. (2002). ITTC – Recommended Procedures and Guidelines for Model Manufacture Ship Models. 7.5-01-01-01, 1–6.
- Jeffri, N. I., Fazita, M. R. N., Leh, C. P., Hashim, R., Kassim, M. H. M., Ibrahim, M., Tay, G. S., Arai, T., Sudesh, K., & Kosugi, A. (2022). Potential of Polyhydroxyalkanoate and Nanocellulose From Oil Palm Trunk as Raw Materials for Additive Manufacturing: A Review. *Polymer Composites*, 44(1), 8–22. <https://doi.org/10.1002/pc.27034>
- Jiao, J., Ren, H., Sun, S., & Adenya, C. A. (2016). Experimental Investigation of Wave-Induced Ship Hydroelastic Vibrations by Large-Scale Model Measurement in Coastal Waves. *Shock and Vibration*, 2016, 1–14. <https://doi.org/10.1155/2016/9296783>
- Kemenperin: *Industri Perkapalan Wujudkan RI Jadi Poros Maritim Dunia*. (n.d.). Retrieved December 22, 2024, from <https://www.kemenperin.go.id/artikel/23597/Industri-Perkapalan-Wujudkan-RI-Jadi-Poros-Maritim-Dunia>
- Kolbasin, A. I., & Husu, O. (2018). Computer-Aided Design and Computer-Aided Engineering. *Matec Web of Conferences*, 170, 1115. <https://doi.org/10.1051/mateconf/201817001115>
- Liu, Y., Zou, L., Zou, Z., & Guo, H. (2018). Predictions of Ship Maneuverability Based on Virtual Captive Model Tests. *Engineering Applications of Computational Fluid Mechanics*, 12(1), 334–353. <https://doi.org/10.1080/19942060.2018.1439773>
- Luo, C. (2024). Exploring the Effects of Automated Production Lines on Traditional Manufacturing Jobs. *Highlights in Business Economics and Management*, 39, 202–206. <https://doi.org/10.54097/80edh171>
- Ma, L., & Yang, J. (2024). Adaptive Recognition of Machining Features in Sheet Metal Parts Based on a Graph Class-Incremental Learning Strategy. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-61443-2>
- Malviya, M., Berdanier, C., & Mehta, P. (2019). *An Empirical Study Linking Additive Manufacturing Design Process to Success in Manufacturability*. <https://doi.org/10.31224/osf.io/gkfp9>

- Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., & Pajarola, R. (2014). Object Detection and Classification From Large-scale Cluttered Indoor Scans. *Computer Graphics Forum*, 33(2), 11–21. <https://doi.org/10.1111/cgf.12286>
- Metzger, D. F., Klahn, C., & Dittmeyer, R. (2023). Downsizing Sustainable Aviation Fuel Production With Additive Manufacturing—An Experimental Study on a 3D Printed Reactor for Fischer-Tropsch Synthesis. *Energies*, 16(19), 6798. <https://doi.org/10.3390/en16196798>
- Minh, N., & Vetterli, M. (2005). The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *Ieee Transactions on Image Processing*, 14(12), 2091–2106. <https://doi.org/10.1109/tip.2005.859376>
- Navangul, G., Paul, R., & Anand, S. (2013). Error Minimization in Layered Manufacturing Parts by Stereolithography File Modification Using a Vertex Translation Algorithm. *Journal of Manufacturing Science and Engineering*, 135(3). <https://doi.org/10.1115/1.4024035>
- Piegl, L., & Tiller, W. (1997). *The NURBS Book*. Springer.
- Pipes, A. L. (2003). *Drawing for Designers*. Laurence King Publishing.
- Pitteri, G. (2020). *3D Object Detection and Pose Estimation of Unseen Objects in Color Images With Local Surface Embeddings*. <https://doi.org/10.48550/arxiv.2010.04075>
- Raja, V., & Fernandes, K. J. (2008). *Reverse Engineering: An Industrial Perspective*. Springer.
- Ramadhan, A., Atmadi, T., & Dinata, R. (2019). Utilization of Computer Aided Design Software as a Visual Simulation. *International Humanities and Applied Science Journal*, 2(3), 1. <https://doi.org/10.22441/ihaj.2019.v2i3.01>
- Rameesha, T. V., & Krishnankutty, P. (2018). Numerical Investigation on the Influence of Froude Number on the Maneuvering Characteristics of a Container Ship. *International Shipbuilding Progress*, 65(2), 149–185. <https://doi.org/10.3233/isp-180145>
- Ren, Z., Wang, J., & Wan, D. (2020). Investigation of the Flow Field of a Ship in Planar Motion Mechanism Tests by the Vortex Identification Method. *Journal of Marine Science and Engineering*, 8(9), 649. <https://doi.org/10.3390/jmse8090649>
- Rodríguez-Sevillano, Á. A., Casati-Calzada, M. J., Bardera-Mora, R., Nieto-Centenero, J., García, J. C. M., & Barderas, E. B. (2023). Rapid Parametric CAX Tools for Modelling Morphing Wings of Micro Air Vehicles (MAVs). *Aerospace*, 10(5), 467. <https://doi.org/10.3390/aerospace10050467>
- Saha, P. (2023). The Impact of Artificial Intelligence and Automation on Businesses Processes

- and Employment: An Analytical Study. *Pne*, 55(1).
<https://doi.org/10.48047/pne.2018.55.1.03>
- Sghaier, M. O., & Lepage, R. (2016). Road Extraction From Very High Resolution Remote Sensing Optical Images Based on Texture Analysis and Beamlet Transform. *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(5), 1946–1958.
<https://doi.org/10.1109/jstars.2015.2449296>
- Shih, R. H. (2018). *Parametric Modeling with Autodesk Inventor*. SDC Publications.
- Steen, S. (2015). *Experimental methods in marine hydrodynamics: statistical analysis*. August.
<http://astorplast.no/wp-content/uploads/AppendixStatisticalAnalysis.pdf>
- Stoumpos, A. I., Kitsios, F., & Talias, M. A. (2023). Digital Transformation in Healthcare: Technology Acceptance and Its Applications. *International Journal of Environmental Research and Public Health*, 20(4), 3407. <https://doi.org/10.3390/ijerph20043407>
- Tautges, T. (2001). CGM: A Geometry Interface for Mesh Generation, Analysis and Other Applications. *Engineering With Computers*, 17(3), 299–314.
<https://doi.org/10.1007/pl00013387>
- Temnov, V. G. (2020). Automated Calculation and Optimization of Structures of Prefabricated Buildings From Standard Elements for Static and Dynamic Effects. *Вестник Гражданских Инженеров*, 17(3), 62–71. <https://doi.org/10.23968/1999-5571-2020-17-3-62-71>
- Temnov, V. G. (2022). Structural Optimization of Forms and Structures of Architectural and Functional Spaces of Habitat Objects by Means of CAD. *Вестник Гражданских Инженеров*, 19(4), 47–57. <https://doi.org/10.23968/1999-5571-2022-19-4-47-57>
- Terada, K., & Kurumatani, M. (2005). An Integrated Procedure for Three-Dimensional Structural Analysis With the Finite Cover Method. *International Journal for Numerical Methods in Engineering*, 63(15), 2102–2123. <https://doi.org/10.1002/nme.1356>
- Terzija, N., & Geisselhardt, W. (2006). A Novel Synchronisation Approach for Digital Image Watermarking Based on Scale Invariant Feature Point Detector. 2585–2588.
<https://doi.org/10.1109/icip.2006.313015>
- Turek, P., Beżłada, W., Cierpisz, K., Dubiel, K., Frydrych, A., & Misiura, J. (2024). Analysis of the Accuracy of CAD Modeling in Engineering and Medical Industries Based on Measurement Data Using Reverse Engineering Methods. *Designs*, 8(3), 50.
<https://doi.org/10.3390/designs8030050>
- Wang, G., Zhang, Y., Yang, J., Zang, Y., & Xu, J. (2024). *Machine Tool Automation and*

Artificial Intelligence: A New Mode of Production. 56.
<https://doi.org/10.1117/12.3029019>

- Wang, K., Zheng, J., & Zhou, Z. (2022). *Neural Face Identification in a 2D Wireframe Projection of a Manifold Object*. <https://doi.org/10.48550/arxiv.2203.04229>
- Watschke, H., Kuschmitz, S., Heubach, J., Lehne, G., & Vietor, T. (2019). A Methodical Approach to Support Conceptual Design for Multi-Material Additive Manufacturing. *Proceedings of the Design Society International Conference on Engineering Design, 1(1)*, 659–668. <https://doi.org/10.1017/dsi.2019.70>
- Xu, W., Zhong, Q., Lin, D., Li, G., & Cao, G. (2021). *CoolBox: A Flexible Toolkit for Visual Analysis of Genomics Data*. <https://doi.org/10.1101/2021.04.15.439923>
- Yoqubjonov, J., Gibadullin, R. F., & Nuriev, M. (2023). Advanced Robotic Process Automation for Enterprise Efficiency. *E3s Web of Conferences, 431*, 7011. <https://doi.org/10.1051/e3sconf/202343107011>
- Zahcril, S. S. N. (2022). *Perancangan Kursi Dan Meja Laptop Yang Ergonomis Pada Sistem CAD*. <https://doi.org/10.31219/osf.io/wbr8c>
- Zeid, I. (2014). *Mastering CAD/CAM*. McGraw-Hill Education.
- Zhang, Y., Li, K., Yu, H. P., Wu, J., & Gao, B. (2020). Digital Fabrication of Removable Partial Dentures Made of Titanium Alloy and Zirconium Silicate Micro-Ceramic Using a Combination of Additive and Subtractive Manufacturing Technologies. *Rapid Prototyping Journal, 27(1)*, 93–98. <https://doi.org/10.1108/rpj-02-2020-0040>

Halaman ini sengaja dikosongkan

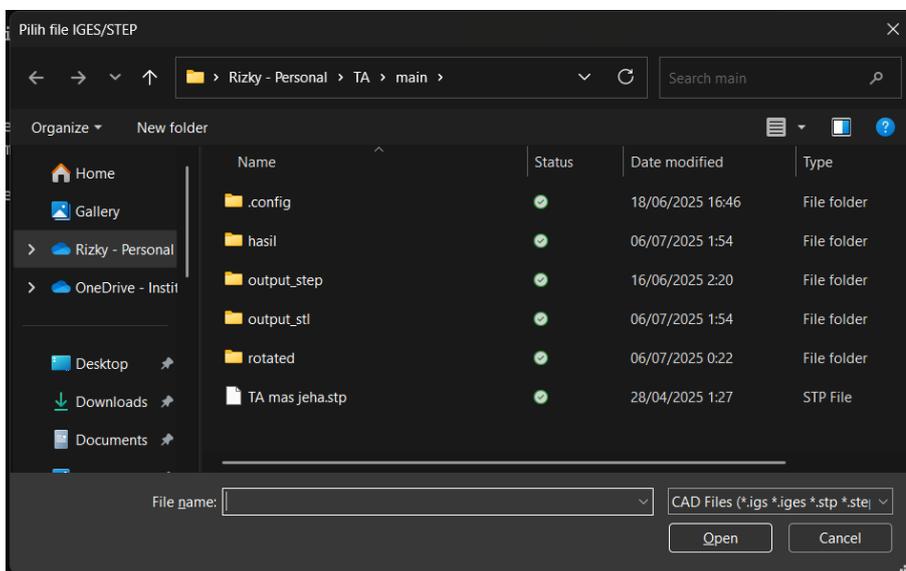
LAMPIRAN A
HASIL PENGUJIAN PROGRAM I

```

Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + -
(base) C:\Users\User>conda activate kapalcadcam
(kapalcadcam) C:\Users\User>cd onedrive/ta/main
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.
(kapalcadcam) C:\Users\User\OneDrive\TA\main>

```

Gambar LA. 1 Running program pada anaconda CMD



Gambar LA. 2 Tahap pemilihan file

```

Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + -
(base) C:\Users\User>conda activate kapalcadcam
(kapalcadcam) C:\Users\User>cd onedrive/ta/main
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
[1] Loaded & scaled: C:/Users/User/OneDrive/TA/main/OPV MODEL.stp
[2] Bounds: X[0.0,1255.0] Y[-105.0,105.0] Z[-0.0,165.0]
[3] Closed solid built.
[4] Voxel candidates: 360
[5] Filtering interior voxels...

```

Gambar LA. 3 Tampilan selanjutnya pasca pemilihan file

```

Anaconda Prompt
Anaconda Prompt (minicond)
Anaconda Prompt (minicond)
Windows PowerShell

(kapalcadcam) C:\Users\User>cd onedrive/ta/main

(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
Tidak ada file dipilih. Program dihentikan.

(kapalcadcam) C:\Users\User\OneDrive\TA\main>python steptostlv9-42.py
[1] Loaded & scaled: C:/Users/User/OneDrive/TA/main/OPV MODEL.stp
[2] Bounds: X[0.0,1255.0] Y[-105.0,105.0] Z[-0.0,165.0]
[3] Closed solid built.
[4] Voxel candidates: 360
[5] Filtering interior voxels...
    Valid voxels: 250
[5] Interior removed.
[6] Applying deck cutter...
[6] Deck cutter done.
[7] Embedding bolt joints...
[7] Bolt joints embedded.
[8] Slicing & exporting...
Warning: output_step/block_01.step file already exists and will be replaced

*****
*****      Statistics on Transfer (Write)      *****
*****
*****      Transfer Mode = 0 I.E. As Is      *****
*****      Transferring Shape, ShapeType = 0      *****
** WorkSession : Sending all data
Step File Name : output_step/block_01.step(6958 ents) Write Done
[8] Exported block_01

```

Gambar LA. 4 Tampilan cmd proses voxelisasi hingga eksporting file .stl dan .step dimulai

```

Anaconda Prompt
Anaconda Prompt (minicond)
Anaconda Prompt (minicond)
Windows PowerShell

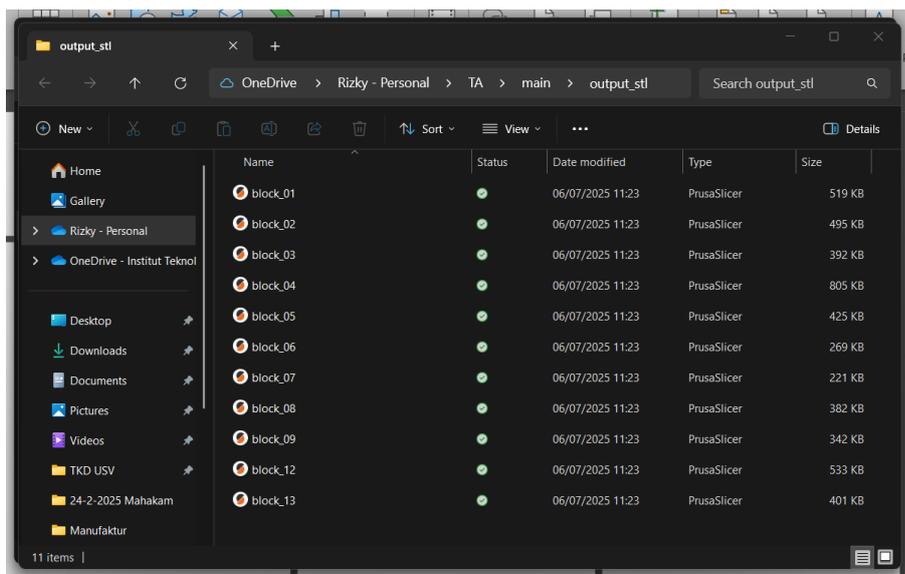
*****
*****      Statistics on Transfer (Write)      *****
*****
*****      Transfer Mode = 0 I.E. As Is      *****
*****      Transferring Shape, ShapeType = 0      *****
** WorkSession : Sending all data
Step File Name : output_step/block_11.step(50 ents) Write Done
[8] Exported block_11
Warning: output_step/block_12.step file already exists and will be replaced

*****
*****      Statistics on Transfer (Write)      *****
*****
*****      Transfer Mode = 0 I.E. As Is      *****
*****      Transferring Shape, ShapeType = 0      *****
** WorkSession : Sending all data
Step File Name : output_step/block_12.step(4696 ents) Write Done
[8] Exported block_12
Warning: output_step/block_13.step file already exists and will be replaced

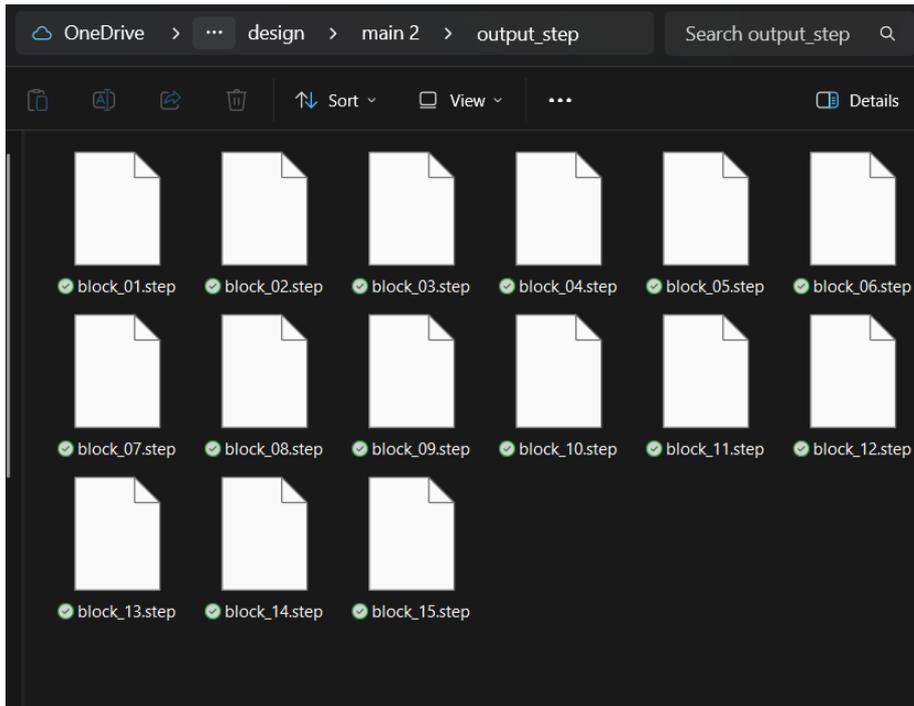
*****
*****      Statistics on Transfer (Write)      *****
*****
*****      Transfer Mode = 0 I.E. As Is      *****
*****      Transferring Shape, ShapeType = 0      *****
** WorkSession : Sending all data
Step File Name : output_step/block_13.step(1883 ents) Write Done
[8] Exported block_13

```

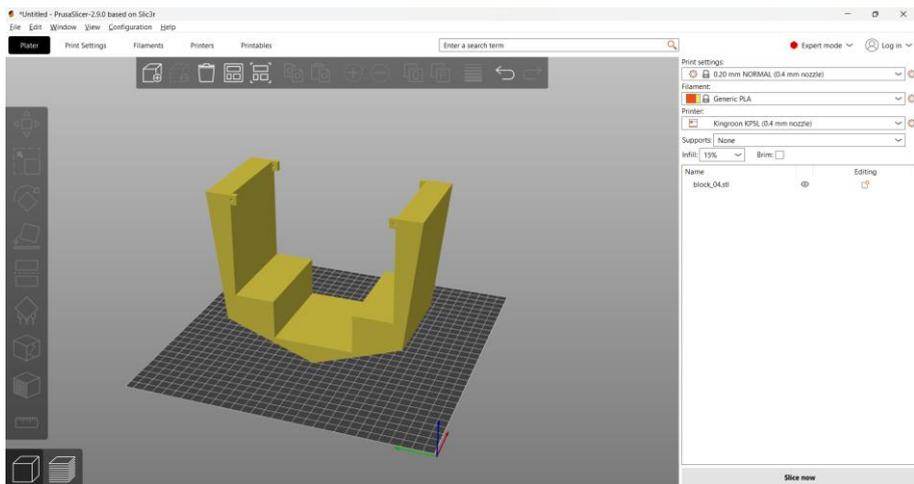
Gambar LA. 5 Tampilan cmd proses eksporting file .stl dan .step



Gambar LA. 6 file .stl hasil pemrosesan program pertama untuk tahap selanjutnya



Gambar LA. 7 file .step hasil pemrosesan program pertama untuk keperluan debugging



Gambar LA. 8 tampak file block_04.stl jika dibuka pada slicer secara manual

LAMPIRAN B
HASIL PENGUJIAN PROGRAM II

```

Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + - _ □ ×
(kapalcadcam) C:\Users\User\OneDrive\TA\main>python slicev5.py
Rotating block_01.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_01_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_01_rotated.stl
Slicing block_01_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_01.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_01.gcode
Rendering preview of block_01_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_01.png
Rotating block_02.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_02_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_02_rotated.stl
Slicing block_02_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed

```

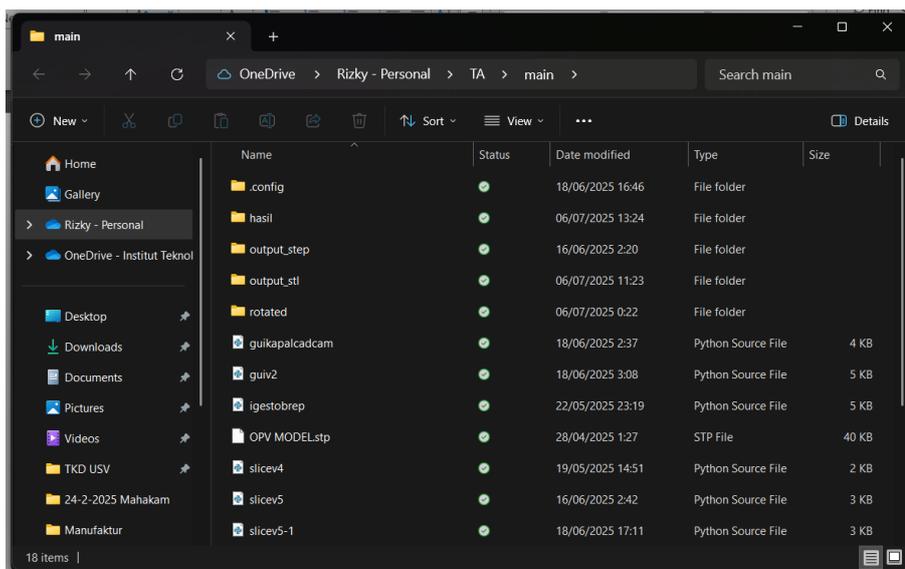
Gambar LB. 1 Proses running program kedua secara manual pada environment cmd

```

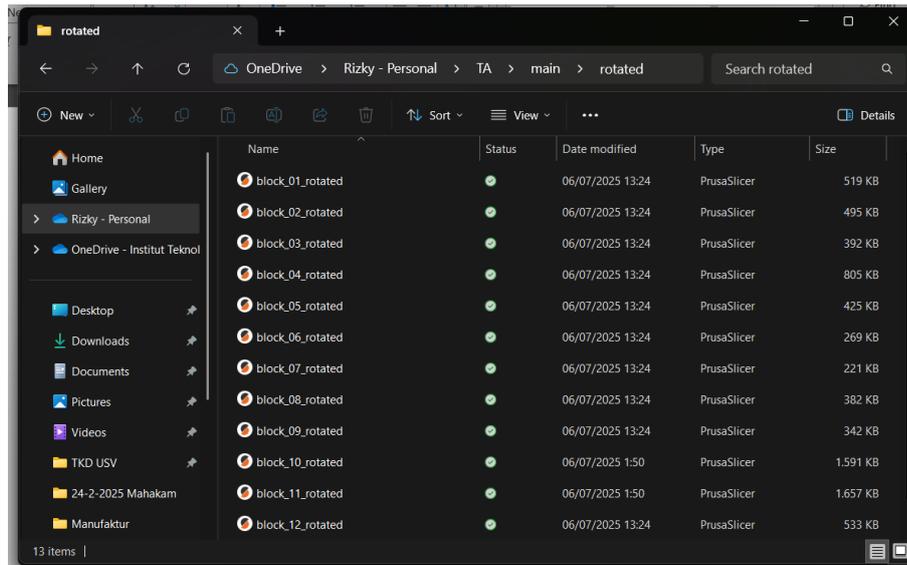
Anaconda Prompt x Anaconda Prompt (minicond x Anaconda Prompt (minicond x Windows PowerShell x + - _ □ ×
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Rendering preview of block_12_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_12.png
Rotating block_13.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl
Slicing block_13_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Rendering preview of block_13_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_13.png

```

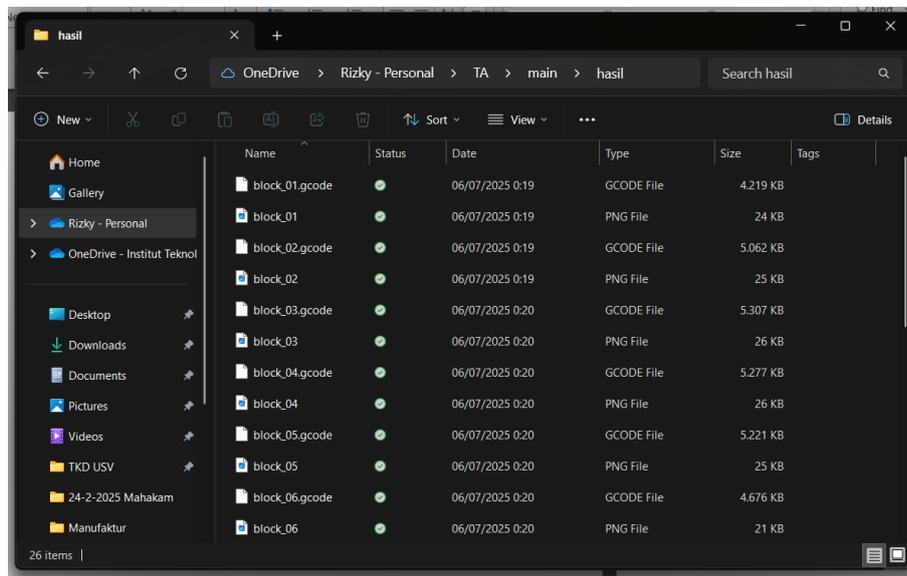
Gambar LB. 2 Tampak proses CMD pasca running program kedua selesai



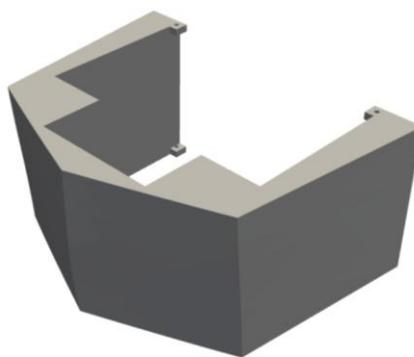
Gambar LB. 3 folder ouput program kedua (rotated dan hasil)



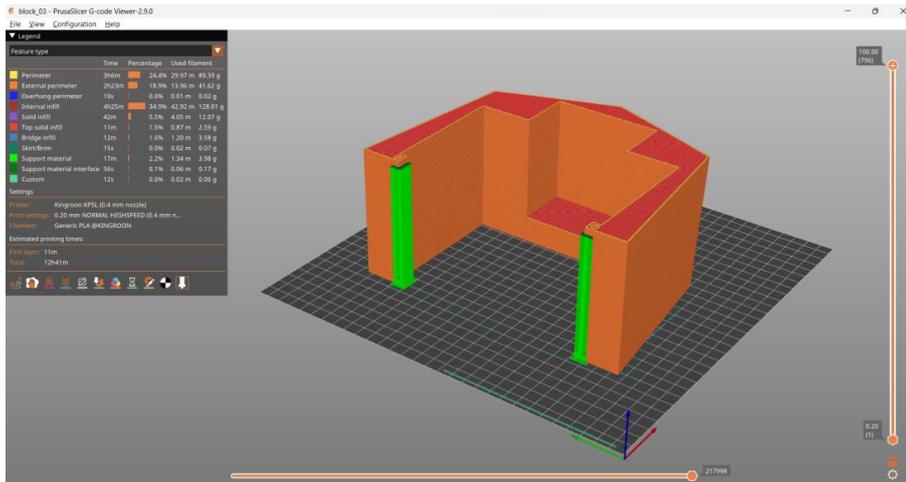
Gambar LB. 4 folder file .stl rotated



Gambar LB. 5 folder file .gcode siap print dan tampilan png masing-masing blok

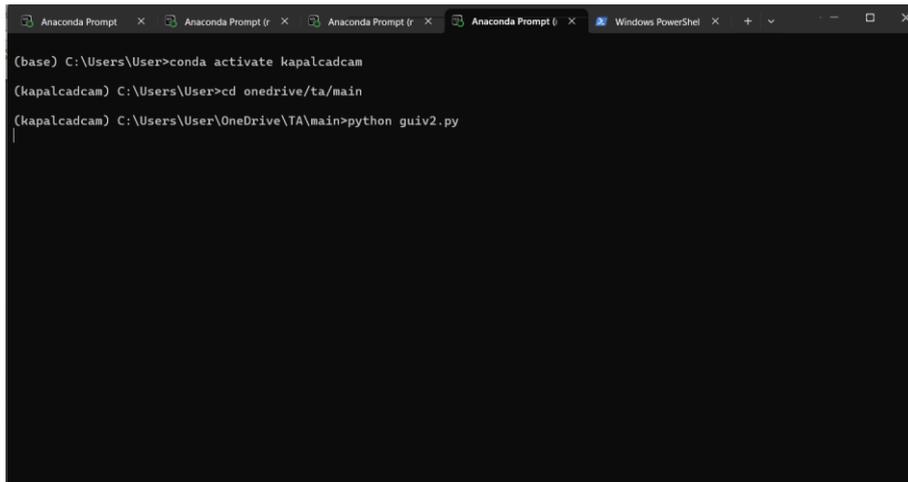


Gambar LB. 6 tampilan file png dari file block_07.stl

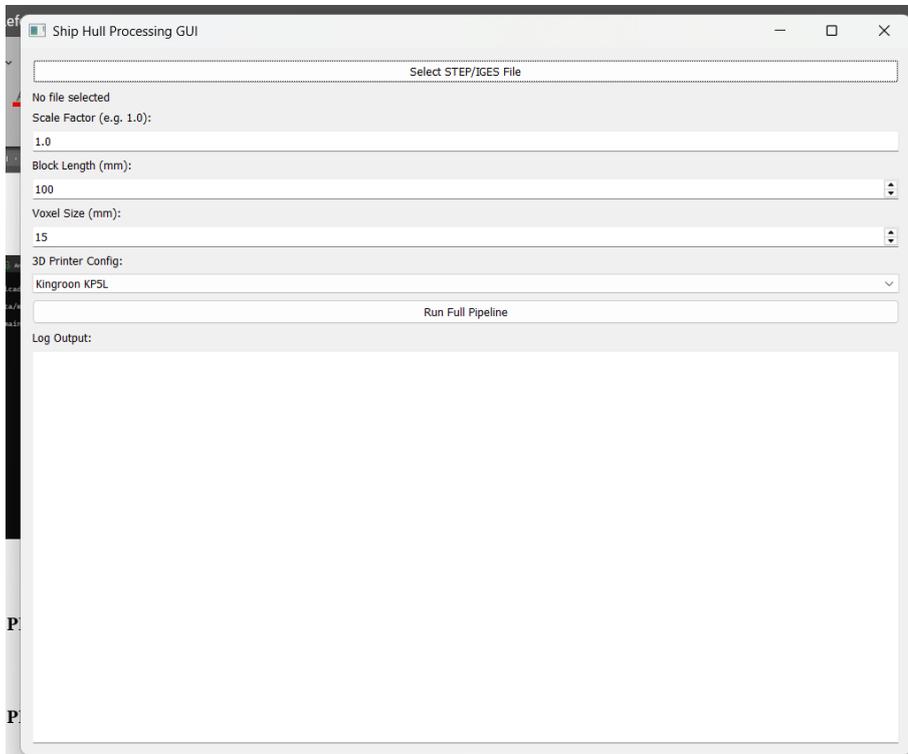


Gambar LB. 7 file .gcode jika dibuka pada .gcode viewer secara manual

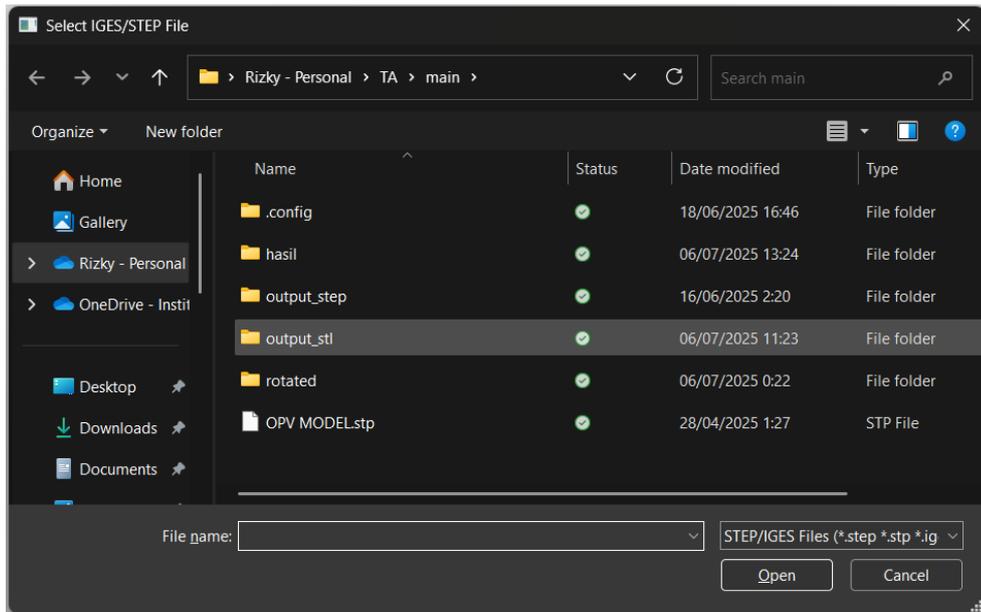
LAMPIRAN C
HASIL PENGUJIAN PROGRAM III (GUI)



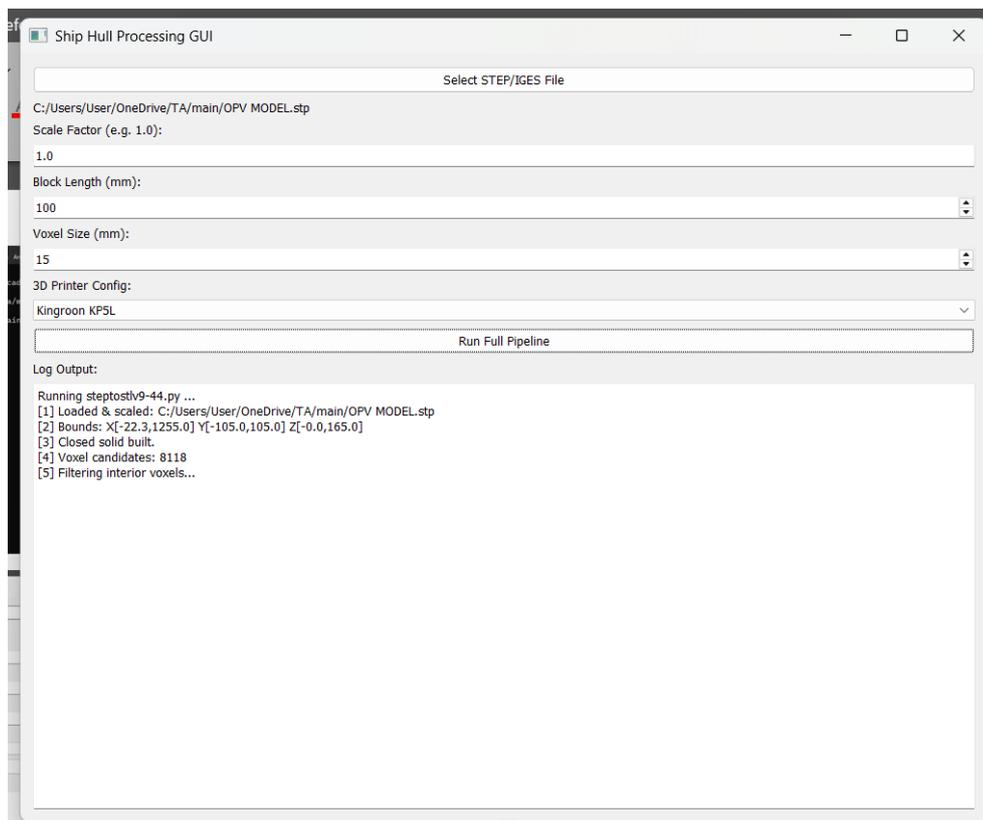
Gambar LC. 1 Running program ketiga



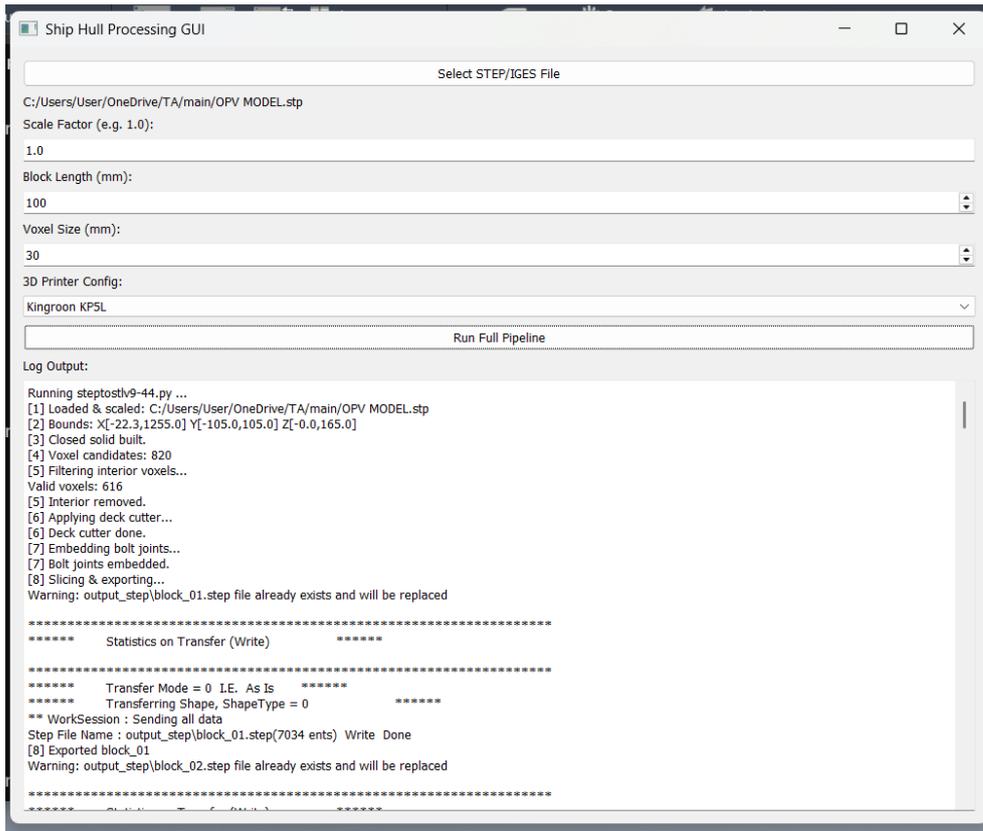
Gambar LC. 2 Tampilan awal GUI



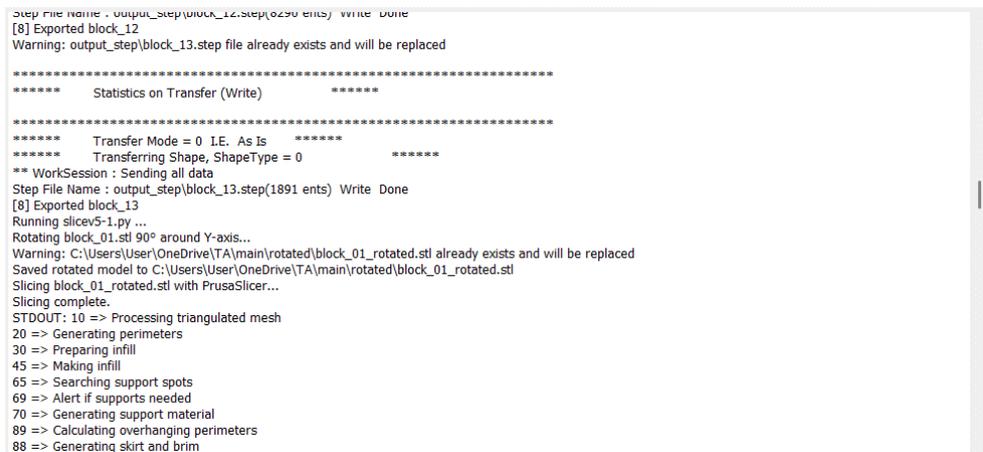
Gambar LC. 3 Tampilan ketika melakukan pemilihan file



Gambar LC. 4 Tampilan setelah melakukan setting variabel dan lakukan run



Gambar LC. 5 Tampak output cmd program I pada Log Output

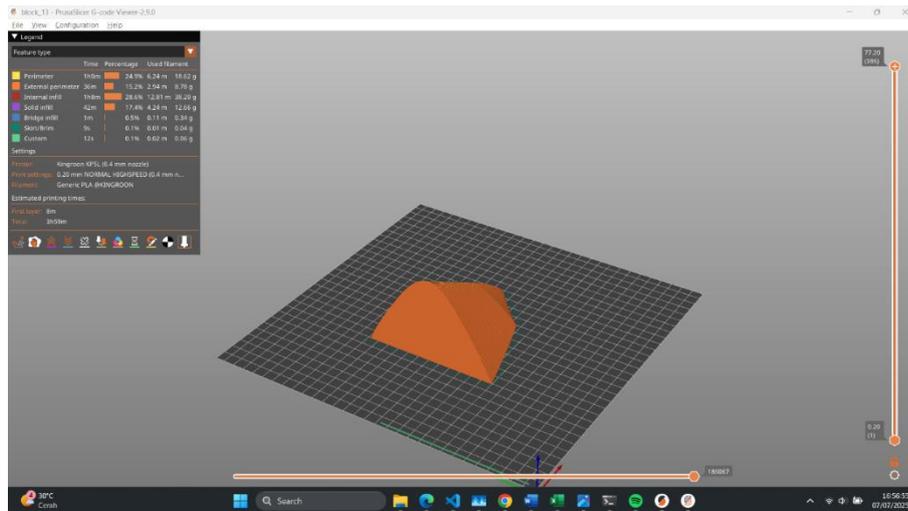


Gambar LC. 6 Tampak output cmd program II pada Log Output

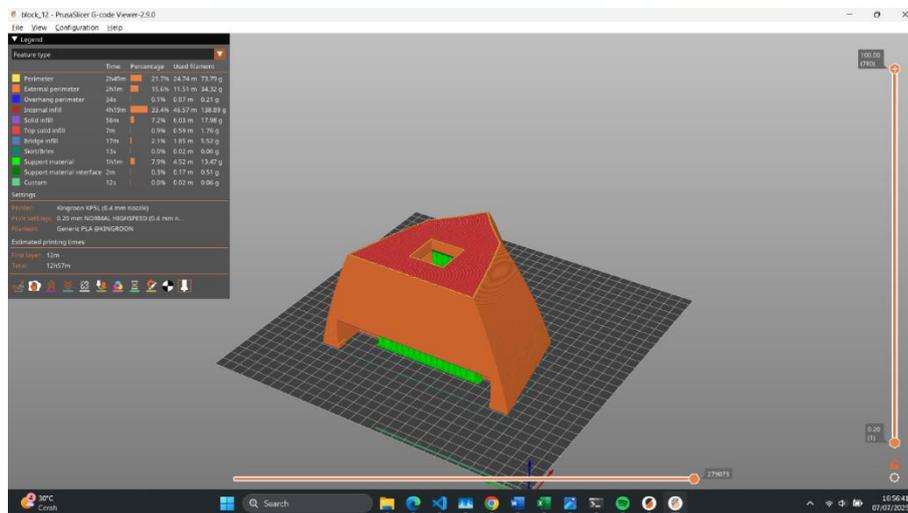
```
Log Output:
--
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_12.gcode
Rendering preview of block_12_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_12.png
Rotating block_13.stl 90° around Y-axis...
Warning: C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl already exists and will be replaced
Saved rotated model to C:\Users\User\OneDrive\TA\main\rotated\block_13_rotated.stl
Slicing block_13_rotated.stl with PrusaSlicer...
Slicing complete.
STDOUT: 10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
70 => Generating support material
89 => Calculating overhanging perimeters
88 => Generating skirt and brim
90 => Exporting G-code to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Slicing result exported to C:\Users\User\OneDrive\TA\main\hasil\block_13.gcode
Rendering preview of block_13_rotated.stl with PyVista...
Render saved to: C:\Users\User\OneDrive\TA\main\hasil\block_13.png
✅ All done.
```

Gambar LC. 7 Tampak output akhir cmd pada Log Output

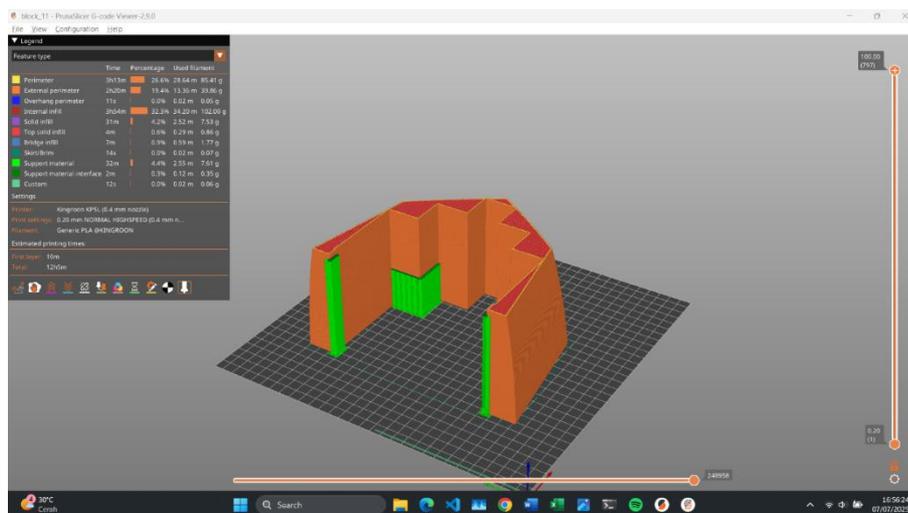
LAMPIRAN D
.GCODE 3D PRINTER



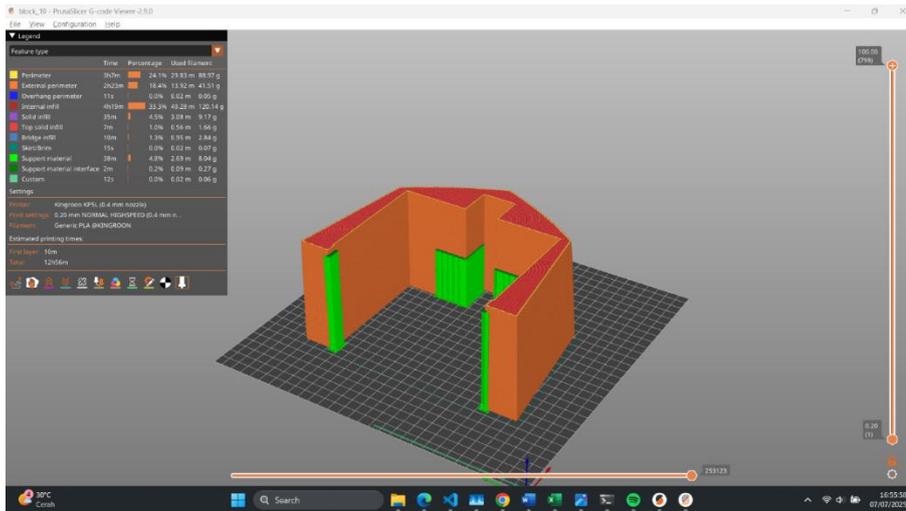
Gambar LD. 1 Tampilan file block_13.gcode dengan .gcode viewer



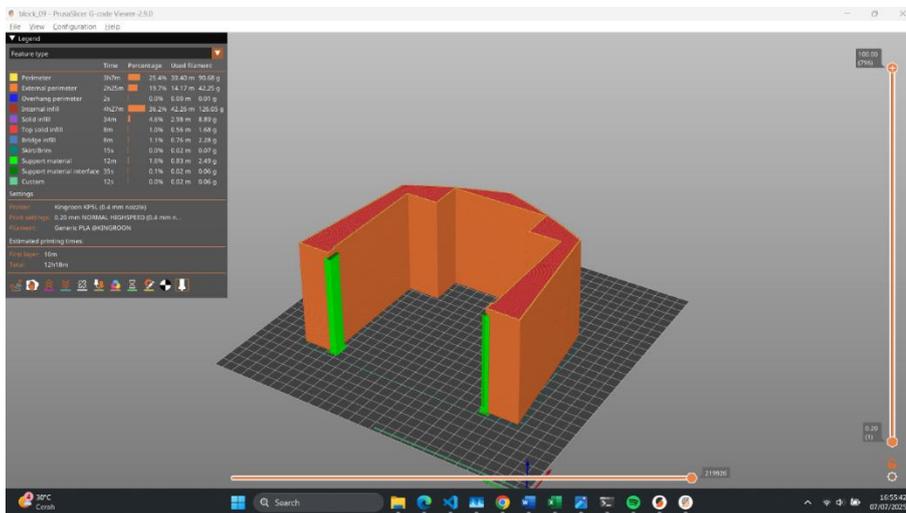
Gambar LD. 2 Tampilan file block_12.gcode dengan .gcode viewer



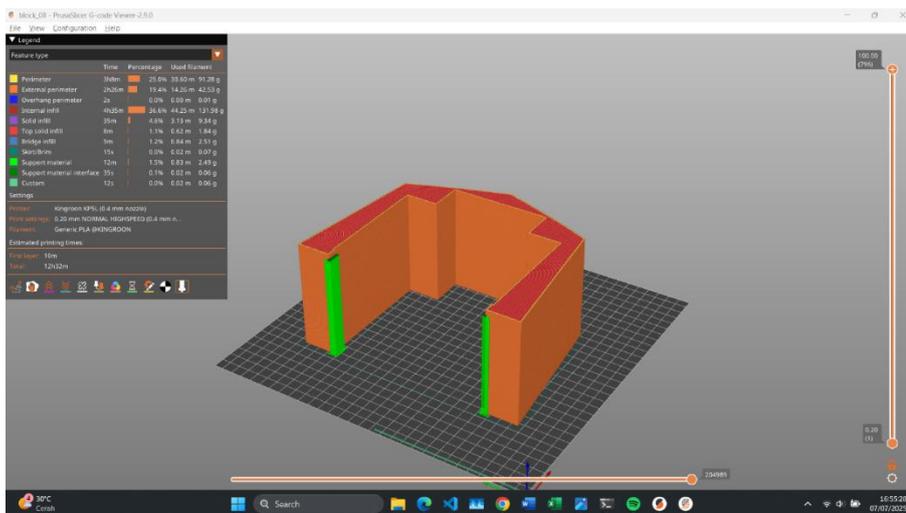
Gambar LD. 3 Tampilan file block_11.gcode dengan .gcode viewer



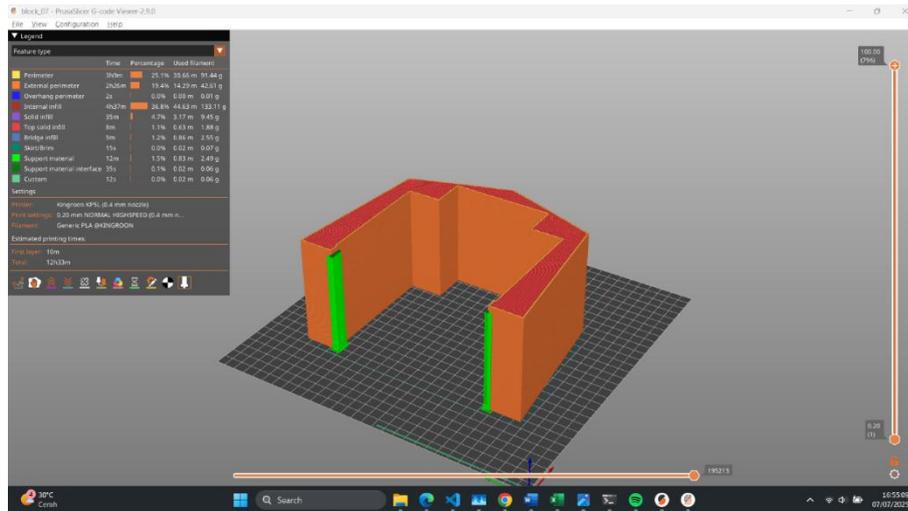
Gambar LD. 4 Tampilan file block_10.gcode dengan .gcode viewer



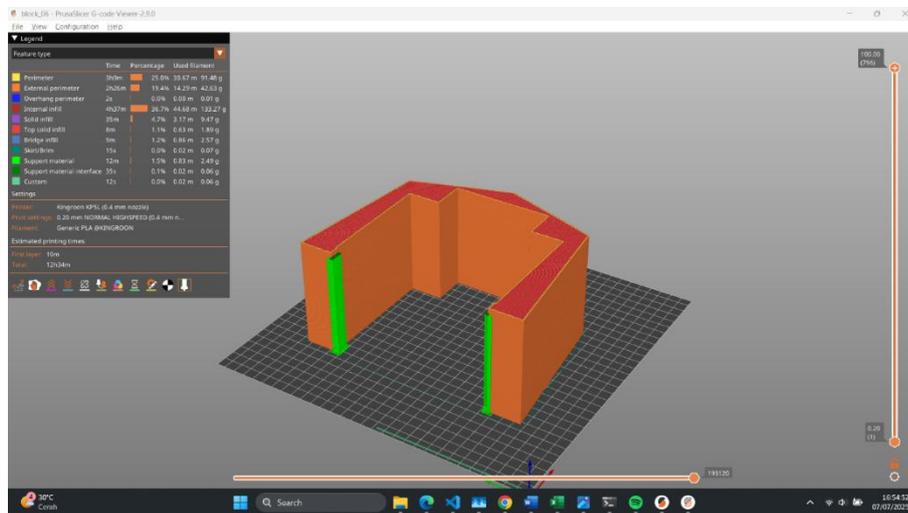
Gambar LD. 5 Tampilan file block_09.gcode dengan .gcode viewer



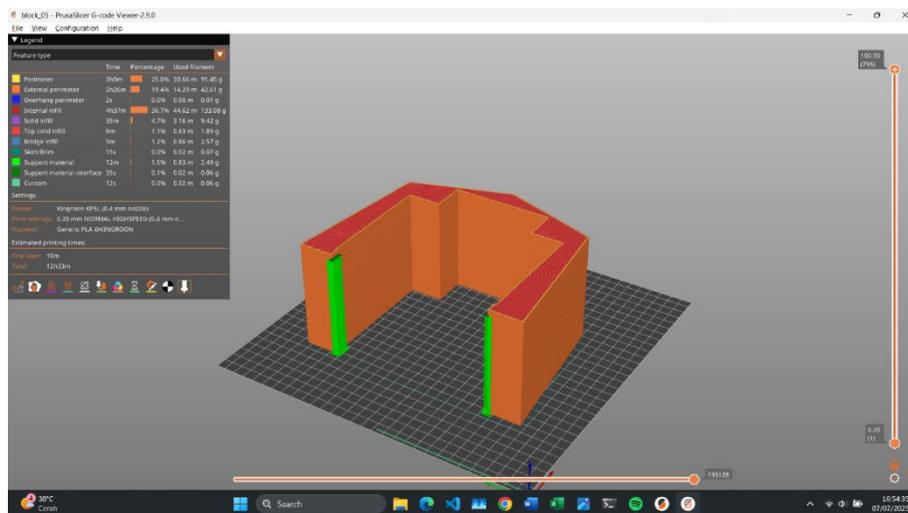
Gambar LD. 6 Tampilan file block_08.gcode dengan .gcode viewer



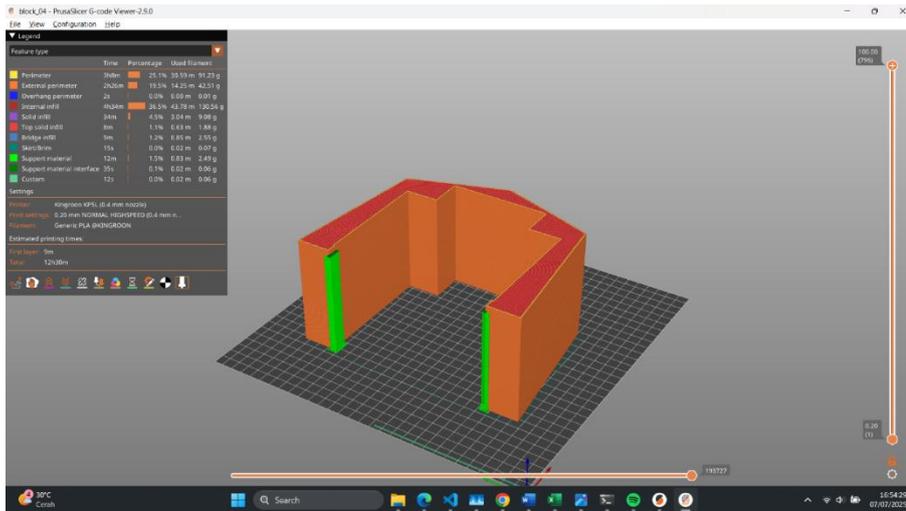
Gambar LD. 7 Tampilan file block_07.gcode dengan .gcode viewer



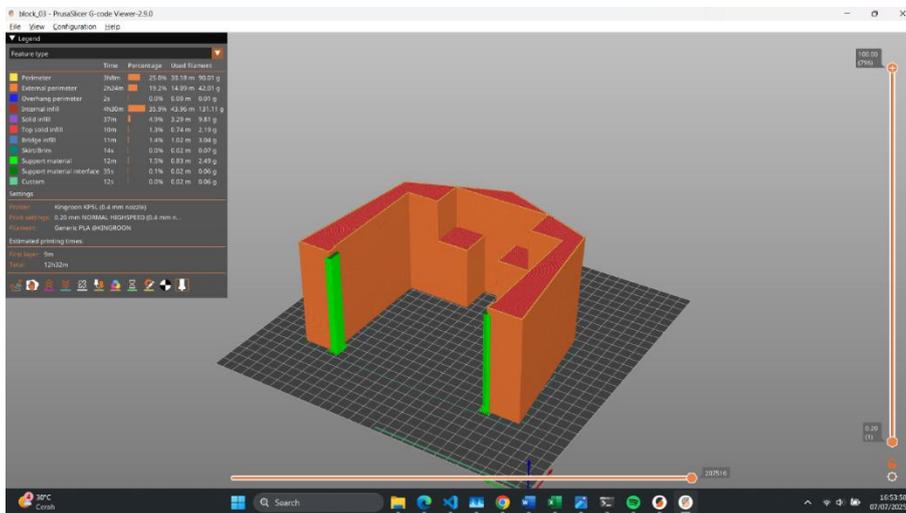
Gambar LD. 8 Tampilan file block_06.gcode dengan .gcode viewer



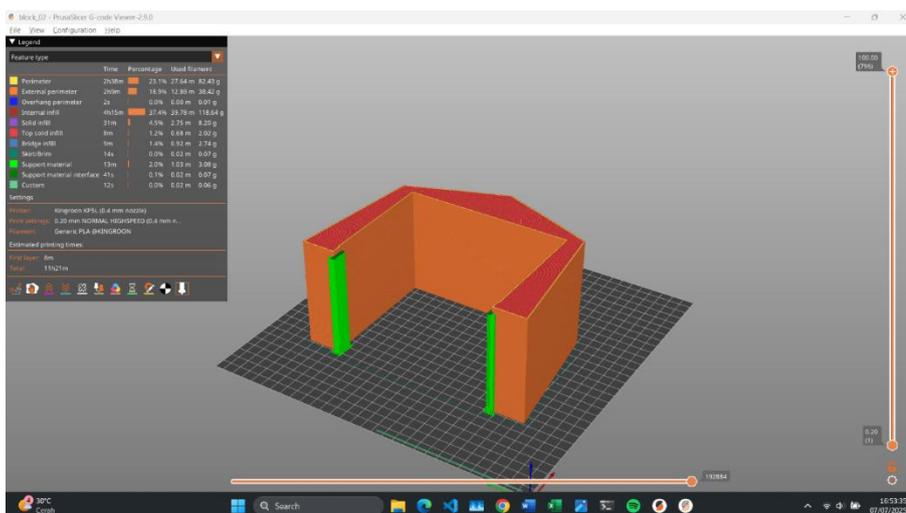
Gambar LD. 9 Tampilan file block_05.gcode dengan .gcode viewer



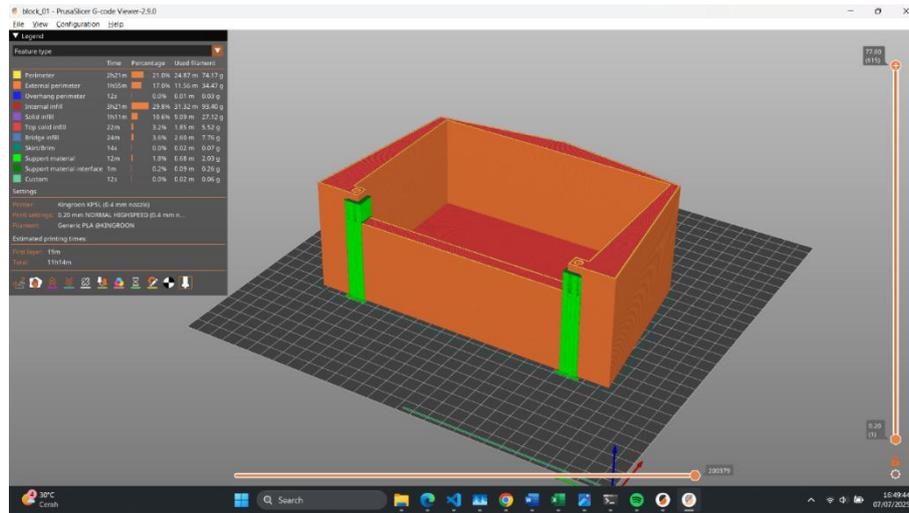
Gambar LD. 10 Tampilan file block_04.gcode dengan .gcode viewer



Gambar LD. 11 Tampilan file block_03.gcode dengan .gcode viewer



Gambar LD. 12 Tampilan file block_02.gcode dengan .gcode viewer



Gambar LD. 13 Tampilan file block_01.gcode dengan .gcode viewer

BIODATA PENULIS



Muhammad Rizky Agung, itulah nama lengkap penulis. Dilahirkan di Ponorogo pada 29 September 2002 silam, Penulis merupakan anak pertama dalam keluarga. Penulis menempuh pendidikan formal di SMA Taruna Nusantara Magelang. Setelah lulus SMA, Penulis diterima di Departemen Teknik Perkapalan FTK ITS pada tahun 2021 melalui jalur SBMPTN.

Di Departemen Teknik Perkapalan Penulis mengambil Bidang Studi Teknologi Digital Kapal. Selama masa studi di ITS, Penulis juga pernah menjadi *staff* Divisi Mekanik ICHIRO-ITS dan melanjutkan hingga menjadi *Manager* Divisi Mekanik ICHIRO-ITS 2023-2024. Selain itu, Penulis juga pernah meraih Juara II pada ajang kompetisi nasional Kontes Robot Sepakbola Humanoid Indonesia-KRI 2023 di Semarang, Jawa Tengah dan Juara I pada ajang kompetisi internasional Robocup Humanoid League 2024 di Eindhoven, Belanda

Email: 5018211048@student.its.ac.id/rzkyagng@gmail.com