

***TRAJECTORY OPTIMIZATION GERAKAN DOBOT COLLABORATIVE PADA
DOBOT MG400 DAN E6 DENGAN PENDEKATAN EVOLUTIONARY ALGORITHM***

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia**



Nama : Muhammad Dava Aji
No. Mahasiswa : 22522125

**PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2026**

PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Jakarta, 25 Juni 2026



METERAI
TEMPEL

F6D25AOX082737048

Muhammad Dava Aji

22522125

SURAT BUKTI PENELITIAN



FAKULTAS
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4110, 4100
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor : 293/Ka.lab SIMANTI/20/Lab.SIMANTI/VI/2026

Assalamu'alaikum Warohmatullahi Wabarokaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Muhammad Dava Aji

Nim : 22522125

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul **"TRAJECTORY OPTIMIZATION GERAKAN DOBOT COLLABORATIVE PADA DOBOT MG400 DAN E6 DENGAN PENDEKATAN EVALUTIONARY ALGORITHM"**.

Mulai penelitian 9 Maret 2026 sampai 10 Juni 2026

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warohmatullahi Wabarokaatuh

Yogyakarta, 11 Juni 2026

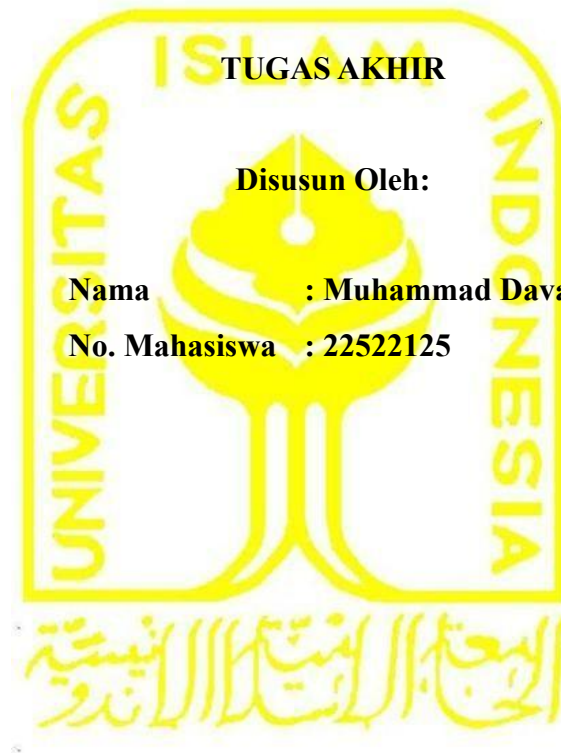
Kepala Laboratorium

Sistem Manufaktur Terintegrasi


Putri Dwi Annisa, S.T., M.Sc.

LEMBAR PENGESAHAN PEMBIMBING

***TRAJECTORY OPTIMIZATION GERAKAN DOBOT COLLABORATIVE PADA
DOBOT MG400 DAN E6 DENGAN PENDEKATAN *EVOLUTIONARY ALGORITHM****



Yogyakarta

Dosen Pembimbing

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM

**TRAJECTORY OPTIMIZATION GERAKAN DOBOT COLLABORATIVE PADA
DOBOT MG400 DAN E6 DENGAN PENDEKATAN *EVOLUTIONARY ALGORITHM***

TUGAS AKHIR

Disusun Oleh :

Nama : Muhammad Dava Aji

No. Mahasiswa : 22522125

**Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Teknologi Industri**

Universitas Islam Indonesia

Yogyakarta, 6 - Juli – 2026

Tim Penguji

Ir. Muhammad Ridwan Andi Purnomo,
S.T., M.Sc., Ph.D., IPM

Ketua

Dr. Agus Mansur, S.T., M.Eng.Sc

Anggota I

Wahyudhi Sutrisno, S.T., M.M., M.T

Anggota II

Mengetahui,

Ketua Program Studi Teknik Industri Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM
NIK. 015220101

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil ‘alamin, segala puji bagi Allah atas limpahan rahmat dan kekuatan-Nya hingga penelitian ini dapat terselesaikan dengan sebaik-baiknya. Penelitian ini kupersembahkan dengan tulus kepada kedua orang tuaku tercinta, yang dengan cinta, doa, dan kesabaran tak henti menjadi cahaya dalam setiap langkah perjuanganku. Juga untuk saudara-saudaraku tersayang, yang selalu menjadi penguat dan teman dalam perjalanan panjang ini. Semoga penelitian ini menjadi awal dari baktiku yang terus tumbuh untuk keluarga yang selalu percaya dan mendukungku.

MOTTO

"Karena sesungguhnya bersama kesulitan itu ada kemudahan."

(QS. Al-Insyirah: 6)

"Kamu Itu Gak Bodoh, Kamu Hanya Butuh Membaca Lebih."

(dr.Tirta)

"Dunia yang membuatmu lelah lantas mengapa tuhan yang kau jauhi."

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh,

Alhamdulillahirabbil'alamiin, segala puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul "*Trajectory Optimization Gerakan Dobot Collaborative Pada Dobot Mg400 Dan E6 Dengan Pendekatan Evolutionary Algorithm*", sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia. Shalawat serta salam tak lupa penulis haturkan kepada Nabi Muhammad SAW yang telah membawa cahaya ilmu pengetahuan bagi umat manusia.

Tugas Akhir ini bukan sekadar rangkaian algoritma, data, atau metode. Ia adalah saksi dari malam-malam panjang yang sunyi, dari doa-doa lirih yang tak terdengar, dari lembar-lembar revisi yang kadang membuat hati ingin menyerah. Ini adalah tentang keberanian untuk melangkah, bahkan ketika langkah terasa berat. Tentang keyakinan, bahwa meski tak selalu mudah, semua perjuangan akan menemukan ujungnya. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. Sri Kusumadewi, S.Si.,M.T. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc. selaku Ketua Jurusan Teknik Industri, Universitas Islam Indonesia.
3. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. selaku Ketua Program Studi Teknik Industri Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia dan selaku dosen pembimbing tugas akhir yang telah memberikan ilmu, arahan, koreksi, dan semangat dalam setiap tahapan penyusunan Tugas Akhir ini.
4. Orang tua penulis serta saudara-saudari saya yang selalu hadir dalam setiap ketikan Tugas Akhir ini, selalu memberikan doa yang tulus, dan selalu memberikan dukungan dengan baik dan halus.
5. Bapak Heri susilo yang memberikan saya tempat yang teduh untuk menimba ilmu di lab siman, diberikan arahan, dan terus diberikan pelajaran yang berharga.
6. Bule yang sudah selalu meminjamkan monitor komputer kepada saya, Prabu dan teman teman jamaah yang selalu mengajak saya nongkrong untuk refresh pikiran.
7. Farah, Khansa dan yang lainnya yang selalu mengerjakan skripsi bersama.
8. Teman-teman saya yang tidak bisa saya ucapkan satu persatu, terima kasih udah memberikan canda tawa, dan ruang yang hangat untuk menghibur saya dikala mengerjakan Tugas Akhir.

Akhir kata, penulis menyadari sepenuhnya bahwa karya ini belumlah sempurna. Namun semoga apa yang ditulis, dihitung, dan direnungkan dalam lembaran ini dapat memberi manfaat bagi dunia akademik maupun praktisi industri yang sedang mencari efisiensi dalam distribusi.

ABSTRAK

Perkembangan industri manufaktur menuju era Industri 4.0 dan 5.0 mendorong penggunaan robot kolaboratif (*cobot*) sebagai solusi otomasi yang fleksibel dan efisien. Dalam sistem produksi *make-to-stock*, efisiensi waktu siklus (*cycle time*) menjadi indikator keberhasilan utama karena penghematan setiap detik akan terakumulasi secara signifikan dalam skala operasional penuh. Penelitian ini bertujuan untuk menentukan posisi koordinat optimal dari tiga titik kritis dalam sistem *Dobot Collaborative*, yaitu *input point*, *collaboration point*, dan *output point*, menggunakan pendekatan *Evolutionary Algorithm* (EA) guna mereduksi *cycle time* secara terukur. Objek penelitian adalah sistem *Dobot Collaborative* di Laboratorium Sistem Manufaktur Terintegrasi Universitas Islam Indonesia, yang terdiri dari Dobot E6 (6-DOF) dengan *mechanical gripper* dan Dobot MG400 (4-DOF) dengan *suction cup*. Optimasi dilakukan melalui fitur *Solver* pada Microsoft Excel dengan fungsi objektif minimasi total jarak tempuh *end-effector*, yang dibatasi oleh parameter kinematika fisik masing-masing robot. Hasil penelitian menunjukkan bahwa penerapan EA berhasil mereduksi total jarak tempuh Dobot E6 sebesar 17,70% dan Dobot MG400 sebesar 3,48%. *Cycle time* Dobot E6 berkurang dari 7,25 detik menjadi 7,15 detik, sedangkan Dobot MG400 berkurang dari 5,00 detik menjadi 4,90 detik. Secara gabungan, total waktu siklus sistem *Dobot Collaborative* berkurang dari 12,25 detik menjadi 12,05 detik, setara reduksi 0,20 detik (1,63%) per siklus. Dalam skala operasional 8 jam kerja per hari, penghematan ini menghasilkan tambahan sekitar 39 siklus produksi atau peningkatan kapasitas sebesar $\pm 1,66\%$ tanpa penambahan perangkat keras apapun. Penelitian ini membuktikan bahwa optimasi koordinat berbasis *Evolutionary Algorithm* dapat menghasilkan *continuous improvement* yang terukur dan signifikan secara kumulatif pada sistem produksi *make-to-stock*, sejalan dengan prinsip *Kaizen* dalam manufaktur modern.

Kata Kunci: *trajectory optimization, evolutionary algorithm, Dobot Collaborative, cycle time, make-to-stock, robot kolaboratif, continuous improvement, Dobot MG400, Dobot E6*

DAFTAR ISI

PERNYATAAN KEASLIAN	ii
SURAT BUKTI PENELITIAN	iii
LEMBAR PENGESAHAN PEMBIMBING	iv
HALAMAN PERSEMBAHAN	vi
MOTTO	vii
KATA PENGANTAR	viii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA	7
2.1 Kajian Literatur	7
2.2 Landasan Teori	13
2.2.1. <i>Trajectory Optimization</i>	13
2.2.2. <i>Dobot Collaborative</i>	13
2.2.3. Dobot E6	13
2.2.4. Dobot MG400	14
2.2.5. <i>End-Effector</i>	15
2.2.6. <i>Suction Cup</i>	15
2.2.7. <i>Gripper</i>	16
2.2.8. Evolutionary Algorithm	17
BAB III METODE PENELITIAN	19
3.1 Objek dan Lokasi Penelitian	19
3.2 Metode Pengumpulan Data	19
3.2.1. Formulasi Fungsi Objektif Utama (<i>Multi-Objective</i>)	20
3.2.2. Penjabaran Komponen Fungsi Objektif	20
3.2.3. Batasan Kinematis (<i>Kinematic Constraints</i>)	21
3.2.4. Perhitungan Jarak Lintasan dan Validasi <i>Cycle time</i>	21
3.2.5. Justifikasi Pemilihan Microsoft Excel Solver sebagai Alat Optimasi	23
3.1. Alur Penelitian	24
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA	28
4.1. Pengumpulan Data Pengujian	28
4.1.1. <i>Dobot Collaborative</i>	28
4.1.2. Formulasi Masalah	29
4.1.3. Data penelitian	30
4.2. Pengolahan Data	31
4.2.1 <i>Trajectory Optimization</i>	31
4.2.2 Evolutionary Algorithm	40
4.2.3 <i>Trajectory Optimization</i> (usulan)	43
BAB V PEMBAHASAN	52

5.1.	Analisis Hasil	52
5.1.1.	Perbandingan Total Jarak Tempuh	52
5.1.2.	Perbandingan <i>Cycle time Real</i>	53
5.1.3.	Perbandingan Waktu Gerak Terukur	54
5.1.4.	Perbandingan Kecepatan Rata-rata	54
5.1.5.	Perbandingan Total Waktu.....	55
5.1.6.	Rekapitulasi Hasil Pembahasan	56
5.1.7.	Analisis Dampak Ekonomis	57
BAB VI	PENUTUP.....	60
6.1.	Kesimpulan.....	60
6.2.	Saran.....	61
DAFTAR PUSTAKA		62
LAMPIRAN		A1

DAFTAR TABEL

Tabel 2. 1 State of art.....	9
Tabel 4. 1 data titik koordinat setiap point	30
Tabel 4. 2 Data waktu.....	30
Tabel 4. 3 hasil validasi awalan.....	39
Tabel 4. 4 Variabel aktual dan batasan	40
Tabel 4. 5 Nilai Tetap	41
Tabel 4. 6 Tabel variabel	41
Tabel 4. 7 Formula jarak 6 segmen	41
Tabel 4. 8 Perbandingan Hasil.....	43
Tabel 4. 9 Validasi Usulan.....	50
Tabel 4. 10 Tabel Perbandingan validasi Usulan dan Awalan	51
Tabel 5. 1 Perbandingan Total Jarak Tempuh.....	52
Tabel 5. 2 Perbandingan <i>Cycle time</i> Real.....	53
Tabel 5. 3 Perbandingan Waktu Gerak Terukur	54
Tabel 5. 4 Perbandingan Kecepatan Rata-rata.....	54
Tabel 5. 5 Perbandingan Total Waktu Siklus Gabungan	55

DAFTAR GAMBAR

Gambar 2. 1 Dobot E6.....	14
Gambar 2. 2 Dobot MG400.....	15
Gambar 2. 3 Section Cup	16
Gambar 2. 4 Gripper.....	16
Gambar 3. 1 Alur Penelitian.....	25
Gambar 4. 1 <i>Dobot Collaborative</i>	31
Gambar 4. 2 3D Modelling <i>Dobot Collaborative</i>	31
Gambar 4. 3 Segmen 1 P1→ P13.....	32
Gambar 4. 4 Segmen 2 P13→ P12.....	33
Gambar 4. 5 Segmen 3 P2→ P3.....	33
Gambar 4. 6 Segmen 4 P3→ P4.....	34
Gambar 4. 7 Segmen 5 P4→ P18.....	34
Gambar 4. 8 Segmen 5 P18→ P6.....	35
Gambar 4. 9 Segmen 1 P3→ P6.....	37
Gambar 4. 10 Segmen 2 P6→ P4.....	37
Gambar 4. 11 Segmen 3 P4→ P6.....	38
Gambar 4. 12 Segmen 3 P6→ P5.....	38
Gambar 4. 13 Konfigurasi excel	43
Gambar 4. 14 Segmen 1 P1→ P13.....	44
Gambar 4. 15 Segmen 2 P13→ P12.....	44
Gambar 4. 16 Segmen 3 P2→ P3.....	45
Gambar 4. 17 Segmen 4 P3→ P4.....	45
Gambar 4. 18 Segmen 5 P4→ P18.....	46
Gambar 4. 19 Segmen 5 P18→ P6.....	46
Gambar 4. 20 Segmen 1 P3→ P6.....	48
Gambar 4. 21 Segmen 2 P6→ P4.....	48
Gambar 4. 22 Segmen 3 P4→ P6.....	49
Gambar 4. 23 Segmen 3 P6→ P5.....	49

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan sektor manufaktur global saat ini tengah mengalami transformasi fundamental menuju era Industri 4.0 dan transisi awal menuju Industri 5.0. Paradigma ini ditandai dengan pergeseran dari sistem produksi massal yang kaku menuju sistem manufaktur yang jauh lebih fleksibel, di mana tuntutan terhadap efisiensi, kecepatan, dan kemampuan adaptasi sistem produksi menjadi semakin krusial. Salah satu teknologi kunci yang memungkinkan fleksibilitas tingkat tinggi ini adalah penggunaan Robot Kolaboratif atau *Cobot* (*Collaborative Robot*)(Hentout et al., 2019). Berbeda dengan robot industri konvensional yang harus bekerja di dalam area pengaman terisolasi, *cobot* dirancang dengan sensor keamanan terintegrasi yang memungkinkannya bekerja berdampingan dengan operator manusia dalam ruang kerja bersama.

Dalam konteks penerapannya, robot kolaboratif paling banyak digunakan pada sistem produksi *make-to-stock*, yakni sistem di mana produk yang sama diproduksi secara massal dalam siklus kerja yang identik dan berulang. Berbeda dengan sistem *make-to-order* yang mengutamakan fleksibilitas pemrograman, tantangan utama pada sistem *make-to-stock* adalah bagaimana memaksimalkan produktivitas dengan cara meminimalkan waktu siklus (*cycle time*) setiap pergerakan robot. Implementasi *cobot* dalam lini produksi berulang ini tidak serta merta menjamin peningkatan produktivitas apabila koordinat titik pergerakan dan tata letak area kerjanya tidak dikonfigurasi secara optimal.

Dalam sistem *make-to-stock*, produk yang identik diproduksi secara berulang dalam siklus kerja yang konstan, sehingga setiap detik yang dapat dihemat dari satu siklus akan terakumulasi menjadi penghematan yang sangat besar dalam skala operasional penuh. Oleh karena itu, indikator keberhasilan utama dalam sistem *make-to-stock* adalah berkurangnya *cycle time*, yakni waktu yang dibutuhkan untuk menyelesaikan satu siklus produksi secara penuh, mulai dari pengambilan bahan baku hingga penempatan produk jadi ke *output tray*. Reduksi *cycle time* secara langsung meningkatkan kapasitas produksi per satuan waktu tanpa memerlukan penambahan sumber daya fisik maupun investasi perangkat keras baru. Pencapaian reduksi *cycle time* inilah yang menjadi bukti konkret dari keberhasilan optimasi pada sistem *make-to-stock*, dan sekaligus menjadi titik tolak bagi penerapan perbaikan yang lebih lanjut dan berkelanjutan dalam lini produksi tersebut(Lorenz et al., 2021).

Dalam *continuous improvement* yang menjadi inti filosofi manufaktur modern, setiap elemen dalam lini produksitermasuk profil pergerakan robotharus dievaluasi dan ditingkatkan secara berkelanjutan. Prinsip ini mengajarkan bahwa tidak ada konfigurasi yang bersifat final; selalu ada potensi perbaikan yang tersembunyi di balik setiap parameter operasional yang tampak sudah berjalan. Semangat perbaikan berkelanjutan inilah yang pada akhirnya mendorong berkembangnya ekosistem perangkat *cobot* yang semakin canggih dan mudah diimplementasikan dalam berbagai skala industri.

Konsep *continuous improvement*, atau dalam bahasa Jepang dikenal sebagai *Kaizen* (改善), merupakan filosofi manajemen yang meyakini bahwa peningkatan kecil dan berkelanjutan pada setiap aspek proses bisnis akan menghasilkan perbaikan yang signifikan secara kumulatif dalam jangka panjang. Kata *Kaizen* terdiri dari dua kata bahasa Jepang: *kai* yang berarti “perubahan” dan *zen* yang berarti “baik”, sehingga secara harfiah bermakna “perubahan menuju yang lebih baik secara terus-menerus”. Filosofi ini pertama kali dipopulerkan oleh Toyota melalui *Toyota Production System* (TPS) dan telah menjadi fondasi utama metodologi *lean manufacturing* modern. Dalam praktiknya, *continuous improvement* diimplementasikan melalui siklus *Plan-Do-Check-Act* (PDCA), di mana setiap perubahan direncanakan, diuji, dievaluasi, dan kemudian distandardisasi apabila terbukti memberikan perbaikan yang terukur. Prinsip ini berlandaskan pada keyakinan bahwa tidak ada sistem atau proses yang bersifat final dan sempurnatermasuk sistem pergerakan robot kolaboratifsehingga selalu terbuka peluang perbaikan yang dapat dieksploitasi melalui pendekatan algoritmik yang terstruktur dan berbasis data(Katerina Kareska, 2024).

Selaras dengan *continuous improvement* tersebut, dalam ekosistem robot kolaboratif yang terus berkembang, merek Dobot telah muncul sebagai salah satu penyedia terkemuka perangkat *cobot* yang dirancang khusus untuk lingkungan pendidikan, penelitian, dan industri skala kecil hingga menengah. Platform Dobot menawarkan kemudahan pemrograman melalui antarmuka visual maupun bahasa pemrograman standar, sehingga menjadikannya pilihan yang sangat relevan untuk studi rekayasa industri dan pengembangan sistem otomasi modern. Untuk mencapai hal tersebut diperlukan optimalisasi platform dobot agar mendapatkan efektifitas dan efisiensi dalam penggunaanya.

Oleh karena itu, salah satu metode yang dapat digunakan untuk optimalisasi dari platform dobot adalah dengan optimalisasi lintasan. Optimalisasi lintasan atau *trajectory optimization* merupakan metode ilmiah yang digunakan untuk menentukan posisi koordinat titik gerak robot

secara optimal. Berbeda dengan perencanaan jalur (*path planning*) yang hanya mendefinisikan rute geometris dari satu titik ke titik lain, perencanaan lintasan melibatkan penetapan hukum gerak yang secara eksplisit mengikat waktu, kecepatan, percepatan, dan sentakan (*jerk*) pada setiap segmen rute tersebut (Gasparetto & Zanotto, 2007a). Dalam konteks penelitian ini, optimasi lintasan tidak hanya bertujuan mempersingkat total jarak tempuh, tetapi juga memastikan bahwa profil gerak yang dihasilkan tetap memenuhi batasan kinematika fisik robot dan tidak menimbulkan getaran berlebihan yang dapat mengganggu kestabilan genggamannya *end-effector*.

Untuk mengoptimalkan posisi ketiga titik tersebut secara sistematis dan ilmiah, penelitian ini menggunakan pendekatan *Evolutionary Algorithm* (EA) sebagai metode optimasi utama. Penting untuk dipahami mengapa masalah ini tidak dapat diselesaikan dengan pendekatan *trial and error* atau metode konvensional berbasis gradien. Setiap titik memiliki tiga komponen koordinat (X, Y, Z) sehingga terdapat minimal sembilan variabel yang dioptimasi sekaligus, dengan setiap perubahan berdampak pada total jarak tempuh melalui perhitungan jarak Euclidean tiga dimensi yang bersifat *non-linear* dan non-konveks. Ruang solusi yang terbentuk sangat besar: pendekatan *trial and error* tidak dapat menjamin ditemukannya solusi optimal global, sementara metode berbasis gradien rentan terjebak di solusi lokal. EA dirancang khusus untuk mengatasi tantangan ini tanpa membutuhkan informasi turunan fungsi, EA menjelajahi ruang solusi secara *global* melalui mekanisme evolusi biologis (seleksi, mutasi, persilangan) dan secara alami menghindari jebakan solusi lokal. Keunggulan EA untuk masalah optimasi lintasan robot berderajat kebebasan tinggi. Yang membedakan penelitian ini dari sekadar *trial and error* adalah bahwa seluruh proses diformulasikan sebagai *model matematis*: fungsi objektif yang secara eksplisit mendefinisikan *cycle time* sebagai fungsi dari koordinat ketiga titik, dilengkapi batasan kinematik yang terukur. Dengan model ini, proses pencarian solusi menjadi terstruktur, dapat direproduksi, dan hasilnya dapat diverifikasi melalui pengukuran aktual di lapangan. Implementasi EA melalui fitur *Solver* pada Microsoft Excel menjadikan pendekatan ini transparan dan dapat direplikasi tanpa perangkat lunak khusus sebuah keunggulan praktis yang memperkuat validitas ilmiah penelitian ini (Garriz & Domingo, 2022).

Pada penelitian ini, kedua robot tersebut dikonfigurasi sebagai satu sistem *Dobot Collaborative* yang bekerja secara terkoordinasi di Laboratorium Sistem Manufaktur Universitas Islam Indonesia (UII). Dobot E6 berperan mengambil objek dari *input tray* yang dilengkapi dengan *mechanical gripper*, kemudian menyerahkannya ke *collaboration point*.

Selanjutnya, Dobot MG400 yang dilengkapi dengan *end-effector* tipe *suction cup* (sistem vakum) mengambil objek dari *collaboration point* tersebut dan memindahkannya ke *output tray*. Siklus kerja kolaboratif ini berulang terus-menerus dalam simulasi lini produksi *make-to-stock*.

Secara teknis, permasalahan yang diteliti adalah penentuan posisi koordinat optimal dalam sistem *Dobot Collaborative*: (1) *input point* titik tempat Dobot E6 mencengkeram objek dari *input tray*; (2) *collaboration point* titik di ruang 3D tempat kedua robot berhadapan untuk serah terima objek; dan (3) *output point* titik tempat Dobot MG400 meletakkan objek ke *output tray*. Ketiga titik ini masing-masing memiliki tiga derajat kebebasan posisi (X, Y, Z): perubahan kecil pada salah satu koordinat akan berdampak langsung pada total jarak tempuh seluruh lintasan dan, dengan demikian, pada *cycle time* waktu yang dibutuhkan untuk menyelesaikan satu siklus kerja penuh dari pengambilan hingga penempatan objek. *Cycle time* inilah yang menjadi satu-satunya indikator keberhasilan optimasi yaitu seberapa besar reduksi waktu per siklus yang dapat diraih melalui pergeseran koordinat ke posisi yang secara paling efisien. Kondisi aktual di laboratorium menunjukkan bahwa konfigurasi saat ini ditetapkan berdasarkan pertimbangan tata letak fisik semata bukan berdasarkan perhitungan yang mempertimbangkan minimasi *cycle time* sehingga masih menyimpan potensi reduksi yang belum optimal.

Observasi awal yang dilakukan di laboratorium membuktikan bahwa pergeseran koordinat ketiga titik tersebut secara terukur dapat mempersingkat waktu siklus gerak sistem *Dobot Collaborative*. Pengujian ini menunjukkan bahwa optimasi salah satu posisi diantara posisi *input point*, *collaboration point*, dan *output point* berhasil mengurangi *cycle time* sebesar sekitar 0,25 detik per siklus. Temuan ini menunjukkan bahwa jalur dobot saat ini masih belum optimal.

Penghematan 0,25 detik per siklus dapat dipahami secara konteksnya, bahwa dalam sistem *make-to-stock* yang sesungguhnya. Apabila sistem *Dobot Collaborative* beroperasi selama 8 jam kerja per hari dengan *cycle time* dari *input* sampai *output* sebesar 12,50 detik per barang, maka total siklus per hari berjumlah sekitar 2.400 siklus (28.800 detik ÷ 12,50 detik/siklus). Penghematan 0,25 detik per siklus berarti menghemat sekitar 600 detik atau 10 menit per hari. Dalam skala operasional nyata dengan kecepatan dan volume yang jauh lebih tinggi, penghematan ini bersifat sangat signifikan dan berdampak langsung pada peningkatan kapasitas produksi tanpa penambahan sumber daya.

Oleh karena itu, penelitian ini berfokus pada penerapan *EA* untuk mengoptimalkan posisi tiga titik (*input point*, *collaboration point*, dan *output point*) pada sistem *Dobot Collaborative* di Laboratorium Sistem Manufaktur UII, dalam rangka mewujudkan *continuous improvement* yang terukur pada sistem produksi *make-to-stock*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menentukan posisi optimal tiga titik (*input point*, *collaboration point*, dan *output point*) pada sistem *Dobot Collaborative* menggunakan *Evolutionary Algorithm* dalam rangka mereduksi *cycle time*?
2. Berapa besar reduksi *cycle time* yang dapat dicapai melalui optimasi sistem *Dobot Collaborative* dibandingkan dengan kondisi setelah dikurangi dengan 0,25 detik (12,25 detik)?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menentukan posisi koordinat optimal pada sistem *Dobot Collaborative* menggunakan *Evolutionary Algorithm*.
2. Mengukur dan menganalisis perbandingan *cycle time* sistem *Dobot Collaborative* sebelum (dikurangi dengan 0,25 detik) dan sesudah optimasi.

1.4 Batasan Masalah

Berikut merupakan Batasan masalah dari penelitian ini:

1. Objek penelitian adalah sistem *Dobot Collaborative* yang terdiri dari Dobot MG400 (4-Axis) dengan *end-effector* tipe *suction cup* dan Dobot E6 (6-Axis) dengan *end-effector* tipe *mechanical gripper* di Laboratorium Sistem Manufaktur Terintegrasi Teknik Industri UII.
2. Variabel optimasi difokuskan poin koordinat Dobot E6 dan MG400
3. Algoritma optimasi yang digunakan terbatas pada pendekatan *Evolutionary Algorithm* melalui *solver* yang tersedia pada perangkat lunak Microsoft Excel.
4. Fungsi objektif optimasi berfokus pada minimasi total *cycle time* sistem *Dobot Collaborative* secara keseluruhan dalam skenario produksi *make-to-stock*.
5. Parameter geometris (dimensi area kerja, jangkauan maksimal lengan, dan zona aman kedua robot) digunakan sebagai batasan (*constraints*) dalam formulasi optimasi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi yang bermanfaat bagi berbagai pihak, antara lain:

1. Bagi sistem yang diteliti
 - a. Memberikan rekomendasi teknis berupa koordinat titik gerak yang optimal (*input point*, *collaboration point*, *output point*) untuk dikonfigurasi ulang pada sistem *Dobot Collaborative*, sehingga aset laboratorium dapat beroperasi dengan *cycle time* yang lebih efisien.
 - b. Hasil penelitian dapat dijadikan acuan perbaikan untuk sistem yang diteliti.
2. Bagi Peneliti
 - a. Sebagai sarana implementasi ilmu pengetahuan yang diperoleh selama perkuliahan, khususnya terkait mata kuliah Otomasi Industri, Robotika, dan Riset Operasi (*Optimization*).
 - b. Meningkatkan kompetensi teknis dalam pemrograman robot kolaboratif dan penerapan *Evolutionary Algorithm* untuk penyelesaian masalah rekayasa nyata berbasis pendekatan *continuous improvement*.
 - c. Melatih kemampuan analisis kuantitatif dalam mengukur dan membuktikan dampak nyata dari optimasi terhadap *cycle time* sistem produksi.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Kajian literatur ini diharapkan dapat memberikan dasar yang kuat bagi pengujian dan analisis dalam rangka mencapai hasil penelitian yang optimal melalui referensi dari penelitian terdahulu dengan topik dan metode yang relevan melalui identifikasi *research gap* tiap referensi yang digunakan. Berikut merupakan kajian literatur yang akan disajikan dalam kajian induktif:

Penelitian mengenai optimasi lintasan (*trajectory optimization*) pada robot industri dan kolaboratif telah berkembang pesat untuk mengatasi keterbatasan metode konvensional dalam hal efisiensi waktu dan konsumsi energi. Garriz dan Domingo (2022) menegaskan pentingnya optimasi lintasan tidak hanya dari sisi performa gerak tetapi juga efisiensi energi dalam industri manufaktur, yang menjadi landasan utama urgensi penelitian ini. Hal ini didukung oleh Akbari et al. (2023) yang membuktikan bahwa penerapan algoritma evolusioner (*Evolutionary Algorithms*) mampu secara simultan mengurangi konsumsi energi dan waktu eksekusi pada lengan robot. Pendekatan *multi-objective* ini sangat krusial, sebagaimana dipaparkan oleh Qin et al. (2022), yang mengembangkan algoritma untuk menyeimbangkan berbagai variabel performa robot industri agar mencapai produktivitas maksimal.

Dalam konteks algoritma yang digunakan, *Genetic Algorithm* (GA) sebagai bagian dari algoritma evolusioner masih menjadi metode yang dominan dan terbukti andal. Zanchettin et al. (2022) secara spesifik meneliti optimasi lintasan pada robot kolaboratif menggunakan simulasi berbasis GA, yang sangat relevan dengan objek penelitian Dobot kolaboratif. Efektivitas GA juga ditunjukkan oleh X. Liu et al. (2022) dalam lingkungan *Digital Twin*, yang memungkinkan prediksi gerak yang presisi. Selain itu, Perera et al. (2021) memanfaatkan GA untuk menyelesaikan masalah *Inverse Kinematics*, yang merupakan fondasi dasar dalam menggerakkan manipulator robot ke posisi target. Aplikasi algoritma genetika juga sempat dikaji oleh Yaping Lu (2023) untuk meningkatkan kecerdasan lintasan pada proses gerinda (*grinding*), meskipun tantangan dalam implementasinya tetap ada.

Selain GA, berbagai varian algoritma metaheuristik baru yang terinspirasi dari alam telah bermunculan untuk menangani kompleksitas kinematika robot, khususnya untuk robot dengan derajat kebebasan tinggi (seperti Dobot E6 yang 6-Axis). Mousa et al. (2024) mengusulkan *Whale-Based Algorithm* untuk lengan robot 6-Axis, yang menawarkan solusi alternatif untuk masalah non-linear. Sementara itu, J. Liu et al. (2023) memperkenalkan *Multiverse Algorithm*

untuk optimasi lintasan, menunjukkan bahwa eksplorasi ruang solusi dapat dilakukan dengan berbagai pendekatan stokastik. Penelitian terbaru dari Sánchez-Sosa dan Chavero-Navarrete (2025) bahkan mengembangkan desain optimal sel robotik menggunakan pendekatan ganda untuk ruang dan waktu berbasis algoritma evolusioner, memperluas cakupan optimasi ke tata letak kerja. Juříček et al. (2023) dalam tinjauan literturnya menyimpulkan bahwa teknik komputasi evolusioner memang menjadi solusi *state-of-the-art* untuk masalah perencanaan jalur (*path planning*) yang kompleks.

Aspek kehalusan gerak (*smoothness*) atau minimasi *jerk* (sentakan) yang menjadi fokus latar belakang penelitian ini juga telah banyak dibahas menggunakan algoritma terbaru. Wei et al. (2024) menggunakan *Improved Carnivorous Plant Algorithm* untuk menghasilkan lintasan yang mulus (*smooth*) dan optimal secara waktu, yang langsung menjawab masalah keausan mekanis pada robot. Serupa dengan itu, Ji dan Yu (2025) menerapkan *Improved Snake Optimization Algorithm* untuk perencanaan lintasan yang optimal terhadap dampak waktu dan guncangan (*time-impact optimal*), yang sangat relevan untuk aplikasi penanganan material sensitif seperti *wafer*.

Di sisi lain, terdapat pula pengembangan metode berbasis *Reinforcement Learning* sebagai pembanding atau pelengkap metode evolusioner. Xu et al. (2025) menggunakan algoritma TD3 yang ditingkatkan untuk optimasi lintasan multi-objektif, sementara Dong et al. (2024) menggabungkan *Deep Reinforcement Learning* untuk mengurutkan tugas sekaligus merencanakan lintasan robot. Meskipun metode pembelajaran mendalam ini ada, pendekatan *Evolutionary Algorithm* yang diusulkan dalam penelitian ini tetap memiliki relevansi tinggi karena kemampuannya yang kuat dalam pencarian global dan fleksibilitasnya terhadap batasan kinodinamik pada robot kolaboratif seperti Dobot MG400 dan E6

Tabel 2. 1 *State of art*

No	Penulis	Judul	Obyek (Industri)			Metode					
			Manufaktur	Kesehatan	Industry Lainnya	<i>Evolutionary Algorithm</i>	<i>Plant Algorithm</i>	<i>Trajectory Optimization</i>	<i>Genetic Algorithm</i>	<i>Multiverse Algorithm</i>	<i>Optimization Algorithm</i>
1	(Sánchez-Sosa & Chavero-Navarrete, 2025)	<i>Evolutionary Algorithms for the Optimal Design of Robotic Cells: A Dual Approximation for Space and Time</i>	√			√					
2	(Xu et al., 2025)	<i>Multi-objective trajectory optimization method for industrial robots based on improved TD3 algorithm</i>	√			√					
3	(X. Liu et al., 2022)	<i>Genetic Algorithm-Based Trajectory Optimization for Digital Twin Robots</i>	√			√					
4	(Wei et al., 2024)	<i>Smooth and Time-Optimal Trajectory Planning for Robots Using Improved Carnivorous Plant Algorithm</i>	√				√				

No	Penulis	Judul	Obyek (Industri)			Metode					
			Manufaktur	Kesehatan	Industry Lainnya	<i>Evolutionary Algorithm</i>	<i>Plant Algorithm</i>	<i>Trajectory Optimization</i>	<i>Genetic Algorithm</i>	<i>Multiverse Algorithm</i>	<i>Optimization Algorithm</i>
5	(Qin et al., 2022)	<i>A Multi-Objective Trajectory Optimization Algorithm For Industrial Robot</i>	√					√			
6	(Yaping Lu, 2023)	<i>Retracted: Improved Genetic Algorithm for Intelligent Grinding Trajectory of Industrial Robot Sensor</i>	√			√					
7	(Garriz & Domingo, 2022)	<i>Trajectory Optimization in Terms of Energy and Performance of an Industrial Robot in the Manufacturing Industry</i>	√					√			
8	(Dong et al., 2024)	<i>Optimizing Robotic Task Sequencing and Trajectory Planning on the Basis of Deep Reinforcement Learning</i>	√					√	√		

No	Penulis	Judul	Obyek (Industri)			Metode					
			Manufaktur	Kesehatan	Industry Lainnya	<i>Evolutionary Algorithm</i>	<i>Plant Algorithm</i>	<i>Trajectory Optimization</i>	<i>Genetic Algorithm</i>	<i>Multiverse Algorithm</i>	<i>Optimization Algorithm</i>
9	(Akbari et al., 2023)	<i>Movement Optimization of Robotic Arms for Energy and Time Reduction using Evolutionary Algorithms</i>	√			√					
10	(Zanchettin et al., 2022)	<i>Trajectory Optimisation In Collaborative Robotics Based On Simulations And Genetic Algorithms</i>						√	√		
11	(Juříček et al., 2023)	<i>Evolutionary Computation Techniques for Path Planning Problems in Industrial Robotics: A State-of-the-Art Review</i>	√	√		√					
12	(Perera et al., 2021)	<i>Optimising Robot Manipulator Inverse Kinematics with Genetic Algorithm</i>	√			√					

No	Penulis	Judul	Obyek (Industri)			Metode					
			Manufaktur	Kesehatan	Industry Lainnya	<i>Evolutionary Algorithm</i>	<i>Plant Algorithm</i>	<i>Trajectory Optimization</i>	<i>Genetic Algorithm</i>	<i>Multiverse Algorithm</i>	<i>Optimization Algorithm</i>
13	(J. Liu et al., 2023)	<i>Robotic arm trajectory optimization based on multiverse algorithm</i>	√							√	
14	(Mousa et al., 2024)	<i>Whale-Based Trajectory Optimization Algorithm for 6 DOF Robotic Arm</i>	√						√		
15	(Ji & Yu, 2025)	<i>Time-Impact Optimal Trajectory Planning for Wafer-Handling Robotic Arms Based on the Improved Snake Optimization Algorithm</i>			√						√
16	(Muhammad Dava Aji, 2026)	<i>Trajectory Optimization Gerakan Dobot Collaborative Pada Dobot Mg400 Dan E6 Dengan Pendekatan Evolutionary Algorithm</i>	√			√		√			

2.2 Landasan Teori

2.2.1. *Trajectory Optimization*

Trajectory Optimization berbeda dengan perencanaan jalur (*path planning*); lintasan mencakup hukum gerak yang mengikat waktu. Metode konvensional sering menggunakan interpolasi polinomial orde tinggi (kubik atau kuintik) atau *B-Splines* untuk memastikan kontinuitas posisi, kecepatan, dan percepatan. Namun, tantangan utama dalam optimasi lintasan modern adalah menyelesaikan masalah *Multi-Objective Optimization* (MOO), yaitu menyeimbangkan tiga variabel yang saling bertentangan: meminimalkan waktu tempuh (*execution time*) untuk produktivitas, meminimalkan sentakan (*jerk*) untuk kehalusan gerak dan keawetan mesin, serta meminimalkan konsumsi energi. Fungsi objektif biasanya dirumuskan sebagai penjumlahan terbobot (*weighted sum*) dari ketiga variabel tersebut, di mana bobot disesuaikan dengan jenis *end-effector* yang digunakan (Gasparetto & Zanotto, 2007a).

2.2.2. *Dobot Collaborative*

Robot kolaboratif, atau yang populer dengan istilah *cobot*, merepresentasikan evolusi signifikan dalam teknologi otomasi industri yang dirancang untuk bekerja berdampingan dengan manusia dalam ruang kerja bersama (*shared workspace*). Berbeda dengan robot industri konvensional yang beroperasi secara terisolasi di dalam pagar pengaman (*safety fences*) untuk mencegah risiko cedera pada pekerja, *cobot* dibangun dengan filosofi interaksi fisik yang aman. Jika robot tradisional dirancang untuk kecepatan tinggi dan beban berat dengan isolasi total, *cobot* dirancang untuk fleksibilitas, kemudahan pemrograman, dan kemampuan mendeteksi keberadaan manusia. Pergeseran ini mendukung konsep Industri 4.0 dan 5.0, di mana keunggulan kognitif manusia digabungkan dengan presisi dan daya tahan robot untuk menciptakan sistem manufaktur yang lebih adaptif dan efisien (Hentout et al., 2019).

2.2.3. *Dobot E6*

Dobot E6 diklasifikasikan sebagai *desktop collaborative robot* dengan konfigurasi lengan artikulasi 6-Derajat Kebebasan (*6-Degree of Freedom* atau 6-Axis). Berbeda dengan robot 4-Axis yang terbatas pada gerakan planar dan vertikal, arsitektur 6-Axis memberikan kemampuan spasial penuh yang menyerupai lengan manusia, memungkinkan *end-effector* untuk mencapai posisi koordinat (x, y, z) dengan orientasi sudut (α, β, γ) atau *roll, pitch, yaw* yang bebas. Secara teoritis, struktur ini memberikan fleksibilitas tinggi dalam menghindari rintangan (*obstacle avoidance*) dan mengatasi masalah singularitas kinematik. Menurut teori robotika, manipulator 6-Axis adalah standar minimum untuk melakukan tugas manipulasi objek

yang kompleks di ruang tiga dimensi secara sembarang, menjadikannya objek ideal untuk studi optimasi lintasan yang melibatkan orientasi sudut yang presisi (Zhang et al., 2022).



Gambar 2. 1 Dobot E6

2.2.4. Dobot MG400

Dobot MG400 diklasifikasikan sebagai *desktop robotic arm* dengan konfigurasi 4-Derajat Kebebasan (4-Axis). Dalam teori robotika, manipulator 4-Axis dirancang khusus untuk efisiensi tinggi pada tugas pemindahan material (*pick-and-place*) dalam bidang kerja horizontal, menyerupai karakteristik robot SCARA (*Selective Compliance Assembly Robot Arm*) namun dengan struktur lengan artikulasi. Secara kinematis, pengurangan derajat kebebasan dari 6 menjadi 4 membatasi orientasi *end-effector* robot; robot ini dapat memposisikan benda pada koordinat (x, y, z) namun rotasinya terbatas pada sumbu vertikal (yaw atau rotasi sumbu Z). Keterbatasan ini menyederhanakan perhitungan kinematika balik (*inverse kinematics*) dibandingkan robot 6-Axis, namun menuntut perencanaan lintasan yang presisi agar robot tidak mengalami singularitas saat mencoba menjangkau area kerja di batas jangkauan maksimumnya. Analisis parameter Denavit-Hartenberg (D-H) tetap menjadi standar fundamental untuk memodelkan kerangka koordinat setiap sendi pada robot jenis ini guna memastikan akurasi posisi (Shi et al., 2011).



Gambar 2. 2 Dobot MG400

2.2.5. *End-Effector*

End-Effector atau sering disebut sebagai *End-of-Arm Tooling* (EOAT) didefinisikan sebagai perangkat yang dipasang pada flensa terakhir (*mounting flange*) dari lengan robot untuk berinteraksi langsung dengan lingkungan atau benda kerja. Jika lengan robot berfungsi sebagai sistem pemosisian (*positioning device*) yang mengantarkan alat ke koordinat tertentu, maka *end-effector* berfungsi sebagai pelaksana tugas utama, baik itu memegang (*grasping*), memproses (seperti pengelasan atau pengecatan), atau merakit. Efektivitas keseluruhan dari sel robotik tidak hanya ditentukan oleh kepresisian lengan robot, tetapi sangat bergantung pada desain dan kesesuaian *end-effector* dengan karakteristik fisik objek yang dimanipulasi. Kegagalan pada *end-effector* berarti kegagalan total pada siklus kerja, terlepas dari seberapa canggih algoritma kontrol yang digunakan pada lengan robot (Thrun & Leonard, 2008).

2.2.6. *Suction Cup*

End-effector tipe *suction cup* atau *vacuum gripper* merupakan perangkat pencengkeram yang paling umum digunakan dalam aplikasi *pick-and-place* industri karena kesederhanaan desain dan fleksibilitasnya dalam menangani objek dengan permukaan datar. Secara prinsip, mekanisme ini bekerja dengan menciptakan perbedaan tekanan udara antara ruang di dalam *cup* (cawan) dengan tekanan atmosfer lingkungan. Pada robot kolaboratif skala kecil seperti Dobot MG400, kevakuman ini umumnya dihasilkan melalui generator vakum berbasis *Venturi effect* atau pompa diafragma listrik mini. Kualitas cengkeraman sangat bergantung pada integritas segel (*seal*) antara bibir karet *cup* dengan permukaan benda kerja; kebocoran udara sekecil apa pun dapat menurunkan perbedaan tekanan secara drastis, yang berakibat pada hilangnya gaya angkat (Mantriota, 2007).



Gambar 2. 3 Section Cup

2.2.7. *Gripper*

Gripper merupakan jenis *end-effector* yang paling menyerupai fungsi tangan manusia, umumnya terdiri dari dua atau lebih jari (*fingers/jaws*) yang digerakkan oleh aktuator listrik atau pneumatik. Dalam aplikasi robot kolaboratif seperti Dobot E6, tipe yang paling umum digunakan adalah *two-finger parallel gripper*. Secara teoritis, mekanisme cengkeraman pada *gripper* mekanik diklasifikasikan menjadi dua mode utama: *Force Closure* dan *Form Closure*. *Form Closure* terjadi ketika bentuk jari melingkupi objek sehingga objek terkunci secara geometris dan tidak mungkin lepas tanpa merusak *gripper*. Namun, pada penanganan benda datar atau kotak yang umum di industri, mode yang lebih sering terjadi adalah *Force Closure* (atau *Friction Grasp*). Dalam mode ini, kestabilan objek sepenuhnya bergantung pada gaya normal yang diberikan oleh jari *gripper* dan koefisien gesek statis antara permukaan jari dan benda. Kegagalan cengkeraman terjadi jika gaya luar (akibat gravitasi atau inersia gerak robot) melebihi gaya gesek tersebut (Fakhri et al., 2023).



Gambar 2. 4 Gripper

2.2.8. Evolutionary Algorithm

Evolutionary Algorithm (EA) merupakan suatu penyelesaian masalah optimasi yang didasari pada teori Darwin tentang kemampuan alami manusia untuk beradaptasi dengan alam. Sebuah populasi individu dalam suatu lingkungan dengan keterbatasan sumber daya, yang dimana bersaing untuk mendapatkan sumber daya tersebut, menghasilkan seleksi alam, yang menghasilkan populasi yang terdiri dari individu yang paling mampu beradaptasi (Ruswan Suwarman, 2021).

Metode *Evolutionary Algorithm* (EA) adalah suatu metode dalam solver Microsoft Excel yang dapat digunakan untuk menemukan solusi model optimal. Metode ini didasarkan pada filosofi evolusi makhluk hidup. Dibawah ini merupakan tiga keunggulan dari *Evolutionary Algorithm* dibandingkan dengan metode metaheuristik (Yulianto et al., 2020).

1. Kompleksitas fungsi objektif tidak mempengaruhi *Evolutionary Algorithm* selama fungsi tersebut dapat dievaluasi untuk masalah uji coba yang diberikan.
2. Kompleksitas kendala yang ada tidak mempengaruhi *Evolutionary Algorithm* secara signifikan.
3. *Evolutionary solver* sendiri tidak terbatas pada Lokasi optimum suatu hasil dan akan terus bekerja untuk mencapai hasil yang diinginkan.

2.2.8. Make To Stock (MTS)

Make-to-Stock (MTS) merupakan salah satu strategi produksi fundamental dalam manajemen manufaktur, di mana produk diproduksi secara massal berdasarkan proyeksi permintaan (*demand forecast*) sebelum adanya pesanan aktual dari pelanggan. Dalam sistem MTS, barang jadi disimpan sebagai stok dan siap dikirimkan begitu pesanan masuk, sehingga lead time pengiriman ke pelanggan menjadi sangat singkat. Strategi ini sangat efektif untuk produk dengan pola permintaan yang stabil dan dapat diprediksi, di mana efisiensi kapasitas produksi dapat dimaksimalkan melalui produksi batch yang berulang

Karakteristik utama sistem *make-to-stock* adalah penggunaan pola produksi *push*, di mana pasokan bahan baku dan proses produksi “didorong” melalui lini produksi berdasarkan rencana yang dibuat di awal periode, bukan berdasarkan permintaan aktual secara real-time. Konsekuensinya, sistem ini sangat bergantung pada akurasi perencanaan dan penjadwalan produksi. Dua kunci utama efisiensi dalam MTS adalah: (1) akurasi *demand forecasting* untuk menentukan jumlah yang harus diproduksi, dan (2) optimasi operasional lini produksi untuk memastikan setiap unit diproduksi dengan efisiensi waktu dan biaya tertinggi (Beemsterboer et al., 2017).

2.2.9. *Continuous improvement*

Continuous improvement, atau dalam bahasa Jepang dikenal sebagai *Kaizen* (改善), adalah filosofi manajemen yang menyatakan bahwa peningkatan kecil dan berkelanjutan pada setiap aspek proses bisnis akan menghasilkan perbaikan yang signifikan secara kumulatif dalam jangka panjang. Kata *Kaizen* terdiri dari dua kata bahasa Jepang: *kai* yang berarti “perubahan” dan *zen* yang berarti “baik”, sehingga secara harfiah bermakna “perubahan menuju yang lebih baik secara terus-menerus”. Pendekatan ini pertama kali dipopulerkan oleh Toyota melalui *Toyota Production System* (TPS) dan telah menjadi fondasi utama metodologi *lean manufacturing* modern.

Prinsip *continuous improvement* berlandaskan pada keyakinan bahwa tidak ada sistem atau proses yang bersifat final dan sempurnaselalu ada peluang perbaikan yang tersembunyi di balik setiap parameter operasional yang tampak telah berjalan dengan baik. Filosofi ini menolak mentalitas “sudah cukup baik” dan mendorong evaluasi serta eksperimentasi yang bersifat saintifik dan terstruktur terhadap setiap elemen proses. Dalam praktiknya, *continuous improvement* diimplementasikan melalui siklus *Plan-Do-Check-Act* (PDCA), di mana setiap perubahan direncanakan, diuji, dievaluasi, dan kemudian distandardisasi apabila terbukti memberikan perbaikan yang terukur (Katerina Kareska, 2024).

BAB III

METODE PENELITIAN

3.1 Objek dan Lokasi Penelitian

Objek penelitian merupakan sasaran ilmiah yang digunakan sebagai tempat pengumpulan data dengan tujuan penelitian yang bersifat objektif, valid, dan reliabel. Objek pada penelitian optimasi jalur dan jarak *Dobot Collaborative* berbasis *trajectory optimization* dengan kombinasi *evolutionary algorithm* untuk meningkatkan efisiensi waktu dan jarak dari *Dobot Collaborative* E6 dan MG400 yang ada di lab sistem manufaktur Teknik Industri UII.

3.2 Metode Pengumpulan Data

Dalam melakukan penelitian ini, peneliti perlu untuk memperhatikan jenis data yang digunakan, hal ini bertujuan untuk memudahkan pada pengambilan data yang akan digunakan. Hanya 1 jenis data yang diperlukan untuk penulis dalam melakukan penelitian ini, yaitu data primer:

1. Data Primer

Data primer merupakan data yang dikumpulkan secara langsung dari objek penelitian melalui alat bantu pengukur atau alat pengambilan data langsung pada subjek. Data primer biasanya diambil melalui observasi dimana peneliti melakukan observasi pada *Dobot Collaborative* E6 dan MG400 dengan mengambil poin koordinat *Dobot Collaborative*, *Cycle time*, waktu tunggu gripper dan vakum, serta modbus

3.3 *Trajectory Optimization*

Dalam ilmu robotika, pergerakan manipulator tidak hanya sebatas menentukan rute geometris dari titik awal ke titik tujuan. Terdapat perbedaan mendasar antara perencanaan jalur (*path planning*) dan perencanaan lintasan (*trajectory planning*); lintasan mencakup hukum gerak yang mengikat waktu. Dengan kata lain, lintasan (*trajectory*) adalah deskripsi terperinci mengenai fungsi waktu dari posisi, kecepatan, percepatan, dan sentakan/jerk untuk setiap sendi robot. Jika perencanaan jalur hanya berfokus pada pembentukan kurva spasial yang bebas dari tabrakan, perencanaan lintasan bertugas mendefinisikan "bagaimana" dan "seberapa cepat" robot tersebut menelusuri kurva spasialnya dari waktu ke waktu. Untuk memastikan robot beroperasi dengan aman dan mulus, metode konvensional sering menggunakan interpolasi polinomial orde tinggi (seperti kubik atau kuintik) atau kurva B-Splines guna memastikan kontinuitas posisi, kecepatan, dan percepatan. Namun, untuk mendapatkan performa yang maksimal, penentuan fungsi waktu ini berkembang menjadi sebuah masalah optimisasi yang

kompleks (Gasparetto & Zanotto, 2007). Berdasarkan batasan masalah yang Anda tulis, parameter optimasi difokuskan pada *Multi-Objective Optimization* (MOO) yang mencakup minimasi waktu tempuh, minimasi sentakan (*jerk*), dan efisiensi energi. Pendekatan untuk menyelesaikan masalah ini adalah dengan menggabungkan ketiga variabel tersebut menggunakan fungsi penjumlahan terbobot (*weighted sum*).

3.2.1. Formulasi Fungsi Objektif Utama (Multi-Objective)

Dalam optimasi lintasan, robot harus bergerak dari titik awal ke titik akhir dengan mengoptimalkan beberapa kriteria sekaligus. Karena kriteria ini sering kali saling bertentangan (misalnya, bergerak cepat akan meningkatkan *jerk* dan energi), fungsi objektif total (F_{obj}) dirumuskan menggunakan metode *weighted sum*:

$$FF_{obj} = w_1 \cdot f_T + w_2 \cdot f_J + w_e \cdot f_E$$

Keterangan:

- w_1, w_2, w_3 = Bobot (penalti) untuk masing-masing kriteria. Bobot ini disesuaikan dengan karakteristik end-effector. Misalnya, untuk MG400 dengan *suction cup*, w_2 (bobot *jerk*) akan diberikan nilai lebih besar agar vakum tidak terlepas.
- f_T = Fungsi objektif waktu (*Time*).
- f_J = Fungsi objektif sentakan (*Jerk*).
- f_E = Fungsi objektif energi/upaya (*Energy*).

3.2.2. Penjabaran Komponen Fungsi Objektif

a. Minimasi Waktu Tempuh (f_T)

Waktu tempuh adalah total waktu yang dibutuhkan robot untuk menyelesaikan lintasan dari titik awal hingga titik akhir.

$$f_T = \sum_{i=1}^N \Delta t_i$$

Keterangan: N adalah jumlah segmen lintasan, dan Δt_i adalah durasi waktu pada setiap segmen.

b. Minimasi Sentakan / Jerk (f_J)

Jerk adalah turunan ketiga dari posisi terhadap waktu, atau laju perubahan percepatan. Meminimalkan *jerk* akan menghasilkan gerakan yang halus dan mengurangi getaran mekanis pada sendi (*joint*) Dobot MG400 dan E6.

$$f_J = \int_0^{t_f} \sum_{j=1}^D (\ddot{q}_j(t))^2 dt$$

Keterangan: D adalah derajat kebebasan robot (4 untuk MG400, 6 untuk E6), t_f adalah waktu total, dan $\ddot{q}_j(t)$ adalah nilai *jerk* pada sendi ke- j .

c. Minimasi Energi (f_E)

Konsumsi energi secara kinematis sering kali didekati dengan meminimalkan kuadrat percepatan, karena percepatan berbanding lurus dengan torsi dan arus motor yang dibutuhkan.

$$f_E = \int_0^{t_f} \sum_{j=1}^D (\ddot{q}_j(t))^2 dt$$

Keterangan: $\ddot{q}_j(t)$ adalah percepatan pada sendi ke- j .

3.2.3. Batasan Kinematis (Kinematic Constraints)

Algoritma evolusioner yang Anda gunakan nantinya harus mencari solusi lintasan yang tidak melanggar batas maksimal fisik dari motor servo Dobot. Batasan ini dirumuskan sebagai:

- Batas Kecepatan: $|\dot{q}_j(t)| \leq V_{maxj}$
- Batas Percepatan: $|\ddot{q}_j(t)| \leq A_{maxj}$
- Batas Jerk: $|\dddot{q}_j(t)| \leq J_{maxj}$

3.2.4. Perhitungan Jarak Lintasan dan Validasi *Cycle time*

Untuk memvalidasi hasil optimasi lintasan, digunakan beberapa rumus dasar perhitungan kinematika. Rumus-rumus ini diterapkan secara konsisten pada data koordinat aktual Dobot E6 dan Dobot MG400 guna membuktikan bahwa reduksi *cycle time* yang diperoleh bersumber dari perpendekan jarak lintasan dan bukan dari perubahan parameter kecepatan robot ((Garriz & Domingo, 2022))

a. Formula Jarak Euclidean Tiga Dimensi

Jarak antara dua titik koordinat dalam ruang tiga dimensi dihitung menggunakan rumus jarak Euclidean. Rumus ini menjadi dasar perhitungan panjang setiap segmen lintasan robot dari titik awal ke titik tujuan (Gaspardo & Zanotto, 2007a):

$$d_i = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

Keterangan: d_i = jarak tempuh segmen ke- i (mm); $\Delta x = x_2 - x_1$; $\Delta y = y_2 - y_1$; $\Delta z = z_2 - z_1$ (selisih koordinat pada sumbu X, Y, dan Z antara titik awal dan titik akhir setiap segmen).

b. Total Jarak Tempuh

Total jarak tempuh end-effector dalam satu siklus gerakan diperoleh dengan menjumlahkan seluruh jarak setiap segmen lintasan (Akbari et al., 2023):

$$d_{total} = \sum_{i=1}^n d_i$$

Keterangan: d_{total} = total jarak tempuh end-effector dalam satu siklus (mm); n = jumlah segmen lintasan; d_i = jarak tempuh pada segmen ke- i .

c. Waktu Gerak Aktual

Cycle time terukur di lapangan mencakup waktu tunggu (wait time) dari instruksi tunda pada program robot, seperti waktu aktivasi gripper atau vakum, yang tidak termasuk dalam waktu gerak aktual. Waktu gerak aktual diperoleh sebagai:

$$t_{gerak} = t_{cycle} - \sum t_{wait}$$

Keterangan: t_{gerak} = waktu gerak aktual robot (detik); t_{cycle} = *cycle time* terukur di lapangan (detik); t_{wait} = total waktu tunggu instruksi tunda pada program (detik).

d. Kecepatan Rata-rata Gerak

Kecepatan rata-rata gerak robot diperoleh dengan membagi total jarak tempuh terhadap waktu gerak aktual. Nilai ini digunakan untuk memvalidasi konsistensi antara hasil perhitungan jarak lintasan dengan *cycle time* aktual yang terukur di lapangan:

$$v_{rata-rata} = \frac{d_{total}}{t_{gerak}}$$

Keterangan: $v_{rata-rata}$ = kecepatan rata-rata gerak end-effector (mm/detik); d_{total} = total jarak tempuh (mm); t_{gerak} = waktu gerak aktual (detik).

e. Prediksi dan Validasi *Cycle time*

Setelah kecepatan rata-rata gerak diperoleh, dapat dihitung prediksi *cycle time* berdasarkan total jarak tempuh lintasan yang baru. Jika prediksi mendekati *cycle time* aktual, maka perhitungan jarak lintasan dinyatakan valid dan kecepatan robot dianggap konstan (Wei et al., 2024).

Formula prediksi *cycle time* adalah:

$$CT_{prediksi} = \frac{d_{total}}{v_{rata-rata}} + \sum t_{wait}$$

Keterangan: $CT_{prediksi}$ = prediksi *cycle time* berdasarkan jarak lintasan yang teroptimasi (detik); d_{total} = total jarak tempuh lintasan usulan (mm); $v_{rata-rata}$ = kecepatan rata-rata hasil validasi awal (mm/detik); t_{wait} = total waktu tunggu (detik). Apabila $CT_{prediksi} \approx CT_{aktual}$, maka model lintasan dan asumsi kecepatan konstan dinyatakan valid dan dapat digunakan untuk mengestimasi dampak optimasi koordinat terhadap *cycle time* secara andal.

3.2.5. Justifikasi Pemilihan Microsoft Excel Solver sebagai Alat Optimasi

Penyelesaian permasalahan optimasi lintasan berbasis *Evolutionary Algorithm* pada dasarnya dapat dilakukan melalui berbagai perangkat lunak, mulai dari *toolbox* khusus seperti *Global Optimization Toolbox* pada MATLAB, *library* berbasis Python seperti DEAP dan PyGAD, hingga *solver* bawaan pada perangkat lunak *spreadsheet* seperti Microsoft Excel. Sebagian besar penelitian terdahulu menggunakan pendekatan pemrograman khusus dengan algoritma metaheuristik tingkat lanjut, seperti TD3, *Improved Carnivorous Plant Algorithm*, maupun *Whale-Based Algorithm*, yang seluruhnya diimplementasikan melalui bahasa pemrograman atau perangkat lunak simulasi tertentu. Berbeda dengan pendekatan tersebut, penelitian ini secara sengaja memilih fitur *Evolutionary Solver* pada Microsoft Excel sebagai alat optimasi utama.

Pemilihan ini didasari beberapa pertimbangan praktis. Pertama, dari sisi aksesibilitas dan biaya, Microsoft Excel merupakan perangkat lunak yang telah tersedia secara luas di lingkungan akademik, tanpa memerlukan lisensi tambahan sebagaimana dibutuhkan oleh MATLAB, sehingga metode penelitian ini dapat direplikasi oleh peneliti lain, laboratorium pendidikan, maupun pelaku industri skala kecil dan menengah tanpa hambatan biaya perangkat lunak. Kedua, dari sisi transparansi, formula perhitungan jarak Euclidean, fungsi *objective* (*cycle time*), serta batasan (*constraints*) pada Excel Solver ditampilkan secara langsung pada sel-sel lembar kerja, sehingga proses optimasi bersifat transparan dan dapat ditelusuri (*traceable*) secara manual berbeda dengan implementasi pada bahasa pemrograman seperti Python atau MATLAB yang membutuhkan penelusuran kode program secara terpisah dari representasi data numeriknya. Ketiga, dari sisi kesesuaian dengan karakteristik data, seluruh data koordinat, *cycle time*, dan waktu tunggu yang dikumpulkan secara alami berbentuk tabular, sehingga pengolahannya lebih efisien dilakukan langsung pada lingkungan *spreadsheet* dibandingkan harus melakukan proses *import* atau konversi data ke lingkungan pemrograman lain (Rady, 2024).

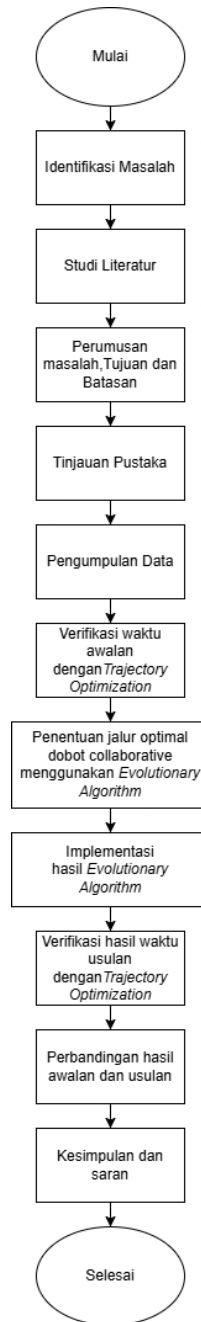
Selain pertimbangan praktis tersebut, pemilihan *Evolutionary Solver* juga didasari kesesuaiannya dengan karakteristik permasalahan penelitian *Solver* pada Microsoft Excel memiliki tiga keunggulan utama dibandingkan metode metaheuristik lain tidak terpengaruh oleh kompleksitas fungsi objektif selama fungsi tersebut dapat dievaluasi, tidak terpengaruh secara signifikan oleh kompleksitas kendala. Ketiga karakteristik ini relevan dengan permasalahan penelitian, yaitu optimasi sembilan variabel koordinat kontinu dengan fungsi

objektif yang bersifat non-linear dan non-konveks (perhitungan jarak Euclidean tiga dimensi) yang tidak memerlukan informasi turunan (gradien) untuk diselesaikan, sehingga metode berbasis gradien maupun pendekatan *trial and error* konvensional kurang sesuai untuk permasalahan ini.

Pemilihan Microsoft Excel Solver bukan dilandasi asumsi bahwa perangkat lunak ini lebih unggul secara algoritmik dibandingkan MATLAB atau Python kedua alternatif tersebut justru menawarkan kontrol parameter algoritma (seperti *population size* dan *mutation rate*) serta kapabilitas komputasi yang lebih tinggi, karena Excel Solver pada dasarnya bersifat *black-box* dengan kontrol parameter yang terbatas. Pemilihan ini murni didasari kesesuaian dengan konteks dan tujuan praktis penelitian, yaitu menghasilkan metode optimasi lintasan yang aksesibel, transparan, dan mudah direplikasi oleh laboratorium pendidikan maupun industri skala kecil-menengah tanpa memerlukan investasi perangkat lunak maupun keahlian pemrograman tingkat lanjut.

3.1. Alur Penelitian

Tujuan dari diagram penelitian ini adalah untuk membuat alur penelitian yang akan memudahkan peneliti dalam melakukan penelitian.



Gambar 3. 1 Alur Penelitian

Berdasarkan diagram alur diatas, berikut merupakan penjelasan dari alur penelitian di atas, sebagai berikut:

1. Mulai

Pada tahap awal ini, dilakukan observasi masalah yang pada *Dobot Collaborative* di Lab Sistem Manufaktur Teknik Industri UII yaitu masalah pada gerakan *Dobot Collaborative*.

2. Identifikasi Masalah

Masalah ini disebabkan karena kurang optimalnya jalur lintasan dobot yang sekarang sehingga masih perlu dioptimalkan.

3. Studi Literatur

Mengumpulkan referensi dan penelitian dari berbagai artikel, jurnal, dan dokumen ilmiah yang membahas tentang masalah di atas, metode optimasi gerakan, terutama untuk *Trajectory Optimization* dan *Evolutionary Algorithm* (EA).

4. Perumusan Masalah, Tujuan, dan Batasan

Pada bagian ini, masalah utama penelitian ini adalah bagaimana mengoptimalkan jalur antar *Dobot Collaborative* efisiensi dari gerakan dobot yang akan mereduksi dari *cycle time*.

5. Tinjauan Pustaka

Ada dua pendekatan yang dilakukan pada tahap ini, yaitu kajian induktif dan kajian deduktif. Kajian induktif mengacu pada temuan penelitian sebelumnya untuk mengidentifikasi perbedaan. Kajian deduktif merujuk pada teori-teori yang mendasari ilmu pengetahuan, seperti *Trajectory Optimization*, dan *Evolutionary Algorithm*.

6. Pengumpulan Data

Data yang dikumpulkan akan digunakan untuk membuat model matematis dan simulasi algoritma. Data ini termasuk luas area *Dobot Collaborative*, *Cycle time* yang gerakan antar dobot, jarak antar *Dobot Collaborative*, jarak maksimal gerakan dobot.

7. Verifikasi waktu awalan dengan *Trajectory Optimization*

Dari data koordinat aktual yang telah dikumpulkan, dilakukan verifikasi *cycle time* awalan menggunakan pendekatan *Trajectory Optimization*. Perhitungan jarak lintasan dan *cycle time* dilakukan berdasarkan formula jarak Euclidean tiga dimensi untuk memastikan bahwa nilai *cycle time* awalan tervalidasi sebelum dilakukan optimasi.

8. Penentuan jalur optimal *dobot collaborative* menggunakan *Evolutionary Algorithm*

Berdasarkan data yang telah terverifikasi, pada tahap ini digunakan *Evolutionary Algorithm* melalui fitur *Solver* pada Microsoft Excel untuk menentukan koordinat optimal ketiga titik kritis (*input point*, *collaboration point*, dan *output point*) pada sistem *Dobot Collaborative* yang dapat meminimalkan total jarak tempuh dan *cycle time*.

9. Implementasi hasil *Evolutionary Algorithm*

Koordinat usulan yang dihasilkan oleh *Evolutionary Algorithm* diimplementasikan pada sistem *Dobot Collaborative*. Jalur baru dikonfigurasi sesuai koordinat optimal yang diperoleh dari hasil optimasi, kemudian sistem dijalankan untuk memperoleh data *cycle time* usulan.

10. Verifikasi hasil waktu usulan dengan *Trajectory Optimization*

Setelah implementasi dilakukan, *cycle time* hasil usulan diverifikasi kembali menggunakan pendekatan *Trajectory Optimization* guna memastikan bahwa reduksi *cycle time* yang diperoleh memang bersumber dari perpendekan jarak lintasan dan bukan dari perubahan parameter kecepatan robot.

11. Perbandingan hasil awalan dan usulan

Setelah didapatkan waktu *Cycle time* awalan dan usulan, maka akan dilakukan perbandingan hasil *Cycle time* dari kedua hasil tersebut mana yang memiliki waktu yang lebih kecil, Sehingga *Cycle time* yang memiliki hasil lebih kecil akan terpilih.

12. Kesimpulan dan Saran

Pada tahap terakhir ini, hasil penelitian diuraikan secara keseluruhan, jawaban terhadap rumusan masalah diberikan, dan saran diberikan yang akan digunakan sebagai usulan perbaikan *Dobot Collaborative*. Selain itu saran diberikan tentang bagaimana melanjutkan penelitian lanjutan.

13. Selesai

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1. Pengumpulan Data Pengujian

Pada bagian ini, dilakukan pengumpulan data berdasarkan metode yang telah dijelaskan sebelumnya. Data yang diperoleh berasal dari hasil pengukuran yang dilakukan secara langsung pada *Dobot Collaborative*.

4.1.1. *Dobot Collaborative*

Penelitian ini menggunakan dua jenis lengan robot dengan karakteristik kinematika yang berbeda, yaitu Dobot MG400 dan Dobot Magician E6, untuk memfasilitasi pengujian algoritma optimasi lintasan secara komprehensif. Penggunaan dua arsitektur robot ini di dalam satu stasiun kerja memberikan landasan eksperimen yang kuat untuk memvalidasi kinerja komputasi dalam menyelesaikan permasalahan *inverse kinematics* pada tingkat kompleksitas yang bervariasi.

Dobot MG400 merupakan robot industri tingkat *desktop* dengan konfigurasi empat derajat kebebasan (4-DOF). Lengan robot ini memiliki kapasitas beban maksimal (*payload*) sebesar 750 gram, radius jangkauan kerja (*reach*) mencapai 440 mm, dan tingkat akurasi pengulangan (*repeatability*) yang sangat presisi di angka ± 0.05 mm. Dengan kecepatan rotasi maksimal mencapai $300^\circ/\text{s}$ pada setiap sendinya, MG400 sangat ideal untuk mengevaluasi parameter fungsi objektif yang berfokus pada efisiensi waktu tempuh dan minimasi jarak absolut (*path length*). Dalam konteks pengujian algoritma evolusioner, arsitektur 4-DOF ini memungkinkan proses pencarian solusi spasial yang lebih efisien karena beban komputasinya relatif lebih ringan, sekaligus memastikan bahwa hasil rancangan lintasan dapat dieksekusi secara aktual tanpa deviasi yang signifikan.

Sebagai variabel komparasi dalam penelitian, stasiun kerja ini juga dilengkapi dengan Dobot Magician E6, sebuah robot kolaboratif (*cobot*) berukuran ringkas yang menawarkan kendali spasial secara menyeluruh melalui mekanisme enam derajat kebebasan (6-DOF). Magician E6 memiliki kapasitas beban yang identik dengan MG400 yaitu 750 gram, namun dengan jangkauan kerja yang sedikit lebih luas mencapai 450 mm, akurasi pengulangan ± 0.1 mm, dan kecepatan sendi maksimal $120^\circ/\text{s}$. Konfigurasi 6-DOF pada *cobot* ini memungkinkan pengendalian orientasi *end-effector* secara utuh (*pitch, roll, yaw*), yang sangat krusial untuk mengevaluasi ketangguhan algoritma saat dihadapkan pada kendala perencanaan lintasan bebas tabrakan (*collision-free path planning*) di sekitar rintangan pada area kerja. Pemecahan *inverse kinematics* pada struktur 6-DOF yang secara matematis lebih kompleks ini akan memberikan

validasi tingkat tinggi terhadap kualitas konvergensi dari algoritma optimasi yang dikembangkan.

4.1.2. Formulasi Masalah

Pada sistem otomasi yang melibatkan lengan robot kolaboratif, penyelesaian masalah dalam pengoptimalan lintasan pergerakan (*trajectory optimization*) dilakukan dengan tujuan untuk meningkatkan efisiensi produksi operasional. Seringkali, *cycle time* atau waktu siklus yang didapatkan di lapangan belum mencapai titik paling minimal yang ditargetkan oleh sistem. Waktu siklus yang tidak bisa maksimal ini dipengaruhi oleh penggunaan koordinat dan jalur pergerakan standar (*default*) antar stasiun kerja yang belum dioptimasi untuk pergerakan spasial terpendek. Idealnya, perpindahan *end-effector* dari satu titik ke titik lain harus dilakukan seefisien mungkin, tetapi yang terjadi di lapangan pergerakan tersebut masih menghasilkan lintasan yang berlebih atau tidak presisi, sehingga *cycle time* menjadi lebih lama. Adapun beberapa hal yang menjadi batasan (*constraints*) yang tidak boleh dilanggar ketika robot beroperasi, yaitu batasan jangkauan kerja maksimum (*workspace limit*), batas sudut rotasi tiap sendi (*joint angle limits*), serta batas kecepatan mekanis maksimum mesin (seperti kecepatan 300°/s untuk MG400 dan 120°/s untuk Magician E6). Oleh karena itu, dibutuhkannya pengoptimalan koordinat posisi lintasan, agar lengan robot dapat mencapai titik target dengan pergerakan paling ringkas tanpa melebihi batasan kinematika tersebut. Penentuan koordinat yang optimal akan dilihat dari perbandingan *cycle time* yang dihasilkan, sehingga minimasi waktu tempuh spasial yang dilakukan akan berdampak langsung pada peningkatan efisiensi sistem kerja robot kolaboratif. Pada penelitian ini terdapat titik awal pergerakan (*home position*) dan beberapa titik tujuan spasial (*target points*), di mana posisi titik referensi pada ruang direpresentasikan dalam koordinat kartesian (x, y, z).

Berdasarkan masalah yang ada pada penelitian di atas, maka permasalahan termasuk dalam jenis kasus *Kinematic Path Planning* dan *Inverse Kinematics Optimization*, yaitu masalah pergerakan distribusi spasial yang memperhitungkan batasan derajat kebebasan (DOF) dan ruang kerja maksimal sebuah lengan robot. Lalu masuk juga ke dalam *Time-Optimal Trajectory Planning*, karena proses pergerakan difokuskan pada penyelesaian target perpindahan dalam durasi paling singkat. Untuk mengeksekusi perhitungan tersebut, algoritma yang digunakan adalah *Evolutionary Algorithm*, yang di mana berfokus pada pencarian solusi ruang gerak terbaik untuk menentukan set titik koordinat paling optimal dengan melihat dari *cycle time* yang tersingkat sebelum dan sesudah optimasi.

4.1.3. Data penelitian

Data yang digunakan untuk penelitian Optimasi Gerakan dobot collaboration yang menggunakan Trajectory Optimization dan *Evolutionary Algorithm* meliputi data titik koordinat tiap point X,Y,Z , *cycle time*, dan waktu tunggu(Gripper, Vakum, Dan. Koordinat yang digunakan pada data berfungsi untuk membuat alur tempuh yang lebih singkat supaya dapat meminimalkan *cycle time* dan cycle yang dibutuhkan untuk menghitung validasi kecepatan dobot. *Cycle time Dobot Collaborative* yaitu 12,25 detik, Berikut tabel 4.1 merupakan data titik koordinat setiap point yang digunakan.

Tabel 4. 1 data titik koordinat setiap point

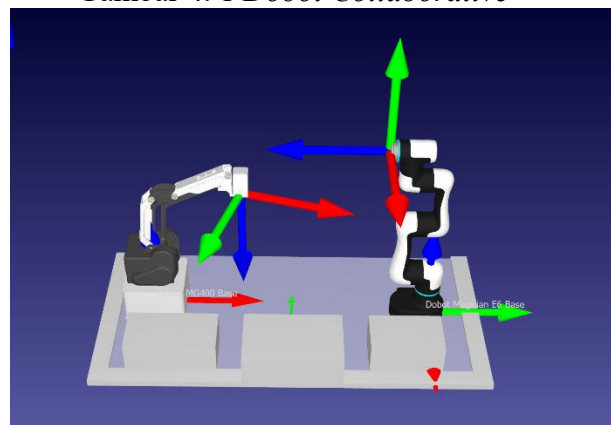
Titik	x (mm)	y (mm)	z (mm)
Dobot E6			
P1(start Point)	0	-255,138	288,986
P13 (Transit)	273,803	-34,938	303,206
P2 (Object1Before)	277,228	-101,740	238,634
P3 (Object1Take)	274,285	-163,159	233,125
P4 (Object1Up)	274,367	-165,573	257,699
P6 (DropPoint)	0,706	-419,448	158,206
P18 (Before Drop)	0	-255,138	288,986
Dobot MG400			
P3(start point)	-0.658	257.05	107.96
P4(Take Object)	-21.10	404.60	11.378
P5(Place object)	256.66	166.00	39.612
P6(Before Take Object)	-19.22	370.71	77.801

Tabel 4. 2 Data waktu

Data	Waktu(Detik)
<i>Cycle time</i>	12,25 detik
Waktu tunggu Gripper	0.5 detik
Waktu tunggu Vakum	1,45 <i>detik</i>
Waktu tunggu Modbus	0,29 detik



Gambar 4. 1 *Dobot Collaborative*



Gambar 4. 2 3D Modelling *Dobot Collaborative*

4.2. Pengolahan Data

Setelah data terkumpul, langkah selanjutnya yaitu melakukan pengolahan data sesuai dengan tahapan yang telah dijelaskan sebelumnya. Proses ini bertujuan untuk memperoleh informasi yang lebih bermakna dari data mentah yang diperoleh. Pengolahan data dilakukan dengan beberapa metode, meliputi *Trajectory optimization* dan *Evolutionary Algorithm*. Perhitungan dilakukan dengan bantuan *software excel*.

4.2.1 *Trajectory Optimization*

Berdasarkan hasil pengumpulan data yang dilakukan, maka didapatkan bahwa terdapat 11 point koordinat yang ada, dengan 7 point dari dobot e6 dan 4 point dobot MG400 pada dobot collaboration di laboratorium sistem manufaktur.

1. Perhitungan E6

Perhitungan pada Dobot Magician E6 dilakukan untuk memperoleh jarak tempuh total end-effector serta memvalidasi kecepatan gerak aktual robot, dengan mengacu pada urutan gerakan

yang telah diprogramkan pada sistem kendali robot. Urutan gerakan tersebut diperoleh dari kode program Object1, DI(3), sebagaimana berikut.

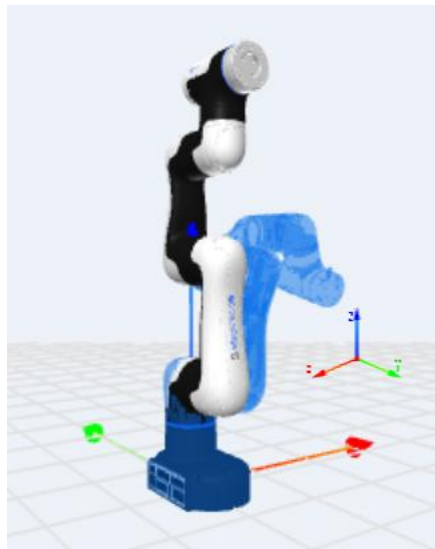
Urutan Gerakan dari Kode (Object1, DI(3)):

P1 → MovJ(P13) → MovJ(P2) → MovJ(P3) → Wait(250ms) → DO(9,1) → Wait(250ms) → MovL(P4) → MovJ(P18) → MovJ(P6) → [Wait → handover MG400]

Koordinat dari point.json (E6):

Pada tahap ini, jarak antar dua titik koordinat dihitung dengan menggunakan rumus jarak Euclidean dalam ruang tiga dimensi, yaitu akar kuadrat dari penjumlahan kuadrat selisih posisi pada sumbu x, y, dan z (Δx , Δy , Δz) antara titik awal dan titik akhir setiap segmen. Perhitungan dilakukan secara berurutan untuk seluruh segmen lintasan sesuai urutan gerakan robot, sehingga diperoleh jarak tempuh (d) pada masing-masing segmen sebagai berikut.

- **Segmen 1 P1 → P13**



Gambar 4. 3 Segmen 1 P1 → P13

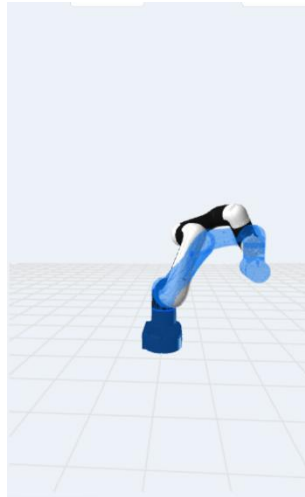
$$\Delta x = 0,000 - 273,803 = 273,803$$

$$\Delta y = 0 - (-34,938) = 34,938$$

$$\Delta z = 0 - 303,2059 = 303,2059$$

$$\begin{aligned} d_1 &= \sqrt{(273,803)^2 + (34,938)^2 + (303,2059)^2} \\ &= \sqrt{74968,08 + 1.220,66 + 91.933,81} = \sqrt{168.122,55} = 410,027 \text{ mm} \end{aligned}$$

- **Segmen 2 P13 → P2**



Gambar 4. 4 Segmen 2 P13→ P12

$$\Delta x = 277,228 - 273,803 = 3,425$$

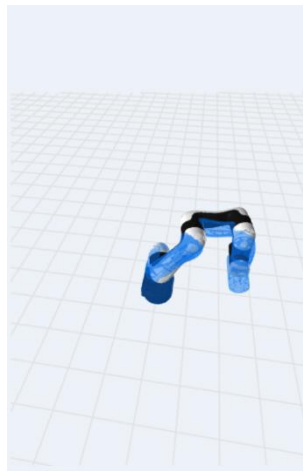
$$\Delta y = -101,740 - (-34,938) = -101,740 + 34,938 = -66,802$$

$$\Delta z = 238,634 - 303,206 = -64,572$$

$$d_2 = \sqrt{(3,425)^2 + (-66,802)^2 + (-64,572)^2}$$

$$= \sqrt{11,73 + 4462,51 + 4169,54} = \sqrt{8643,78} = 92,97 \text{ mm}$$

• **Segmen 3 P2 → P3**



Gambar 4. 5 Segmen 3 P2→ P3

$$\Delta x = 274,285 - 277,228 = -2,943$$

$$\Delta y = -163,159 - (-101,740) = -163,159 + 101,740 = -61,419$$

$$\Delta z = 233,125 - 238,634 = -5,509$$

$$d_3 = \sqrt{(-2,943)^2 + (-61,419)^2 + (-5,509)^2}$$

$$= \sqrt{8,66 + 3772,29 + 30,35} = \sqrt{3811,30} = 61,74 \text{ mm}$$

• **Segmen 4 P3 → P4 (MovL)**



Gambar 4. 6 Segmen 4 P3→ P4

$$\Delta x = 274,367 - 274,285 = 0,082$$

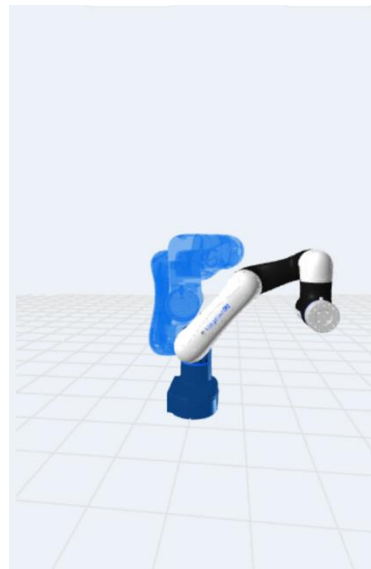
$$\Delta y = -165,573 - (-163,159) = -165,573 + 163,159 = -2,414$$

$$\Delta z = 257,699 - 233,125 = 24,574$$

$$d_4 = \sqrt{(0,082)^2 + (-2,414)^2 + (24,574)^2}$$

$$= \sqrt{0,007 + 5,827 + 603,88} = \sqrt{609,71} = 24,69 \text{ mm}$$

• **Segmen 5 P4 → P18**



Gambar 4. 7 Segmen 5 P4→ P18

$$\Delta x = 0,000 - 274,367 = -274,367$$

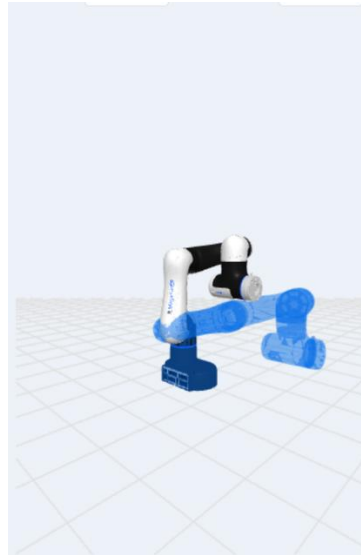
$$\Delta y = -255,138 - (-165,573) = -255,138 + 165,573 = -89,565$$

$$\Delta z = 288,986 - 257,699 = 31,287$$

$$d_5 = \sqrt{(-274,367)^2 + (-89,565)^2 + (31,287)^2}$$

$$= \sqrt{75277,26 + 8021,89 + 978,88} = \sqrt{84278,03} = 290,30 \text{ mm}$$

• Segmen 6 P18 → P6



Gambar 4. 8 Segmen 5 P18→ P6

$$\Delta x = 0,706 - 0,000 = 0,706$$

$$\Delta y = -419,448 - (-255,138) = -419,448 + 255,138 = -164,310$$

$$\Delta z = 158,206 - 288,986 = -130,780$$

$$d_6 = \sqrt{(0,706)^2 + (-164,310)^2 + (-130,780)^2}$$

$$= \sqrt{0,499 + 26997,77 + 17103,41} = \sqrt{44101,68} = 210,00 \text{ mm}$$

Setelah jarak setiap segmen diperoleh pada langkah sebelumnya, selanjutnya seluruh jarak tersebut dijumlahkan untuk mendapatkan total jarak tempuh (d_{total}) end-effector dalam satu siklus gerakan. Selain itu, dihitung pula total waktu tunggu (wait time) yang berasal dari instruksi tunda pada program, seperti waktu tunggu gripper, yang turut memengaruhi *cycle time* namun tidak termasuk dalam waktu gerak aktual robot.

$$d_{total,E6} = 410,027 + 92,97 + 61,74 + 24,69 + 290,30 + 210,00 = 1089,727 \text{ mm}$$

Waktu yang dibutuhkan: 7.25 detik (termasuk Waktu tunggu Griper)

Waktu tunggu (gripper saja):

$$t_{wait} = 250 + 250 = 500 \text{ ms} = 0.5 \text{ detik}$$

Kecepatan rata-rata gerak robot diperoleh dengan membagi total jarak tempuh terhadap waktu gerak aktual (t_{gerak}), yaitu *cycle time* real yang tercatat pada saat pengujian

dikurangi dengan total waktu tunggu. Perhitungan ini bertujuan untuk memvalidasi kesesuaian antara hasil perhitungan jarak lintasan dengan waktu siklus aktual yang terukur di lapangan.

Dari data real skripsi: *cycle time* E6 = 7,25 detik

$$t_{gerak} = 7,25 - 0,5 = 6,75 \text{ detik}$$

$$v_{rata-rata,E6} = \frac{1089,727}{6,05} = 161,441 \text{ mm/detik}$$

2. PERHITUNGAN MG400

Perhitungan dengan metode yang sama juga dilakukan pada Dobot MG400 untuk memperoleh jarak tempuh total dan kecepatan rata-rata gerak end-effector, berdasarkan urutan gerakan yang diprogramkan pada kode Object1, Coil 0, sebagaimana berikut.

Urutan Gerakan dari Kode (Object1, Coil 0):

MovJ(P6) → MovL(P4) → DO(9,1) → MovJ(P6) → Jump(P5) → DO(9,0)

Koordinat dari point.json (MG400):

Sebagaimana pada perhitungan Dobot E6, jarak setiap segmen lintasan Dobot MG400 dihitung menggunakan rumus jarak Euclidean berdasarkan selisih koordinat x, y, dan z antar titik pada tiap segmen, sehingga diperoleh hasil sebagai berikut.

- **Segmen 1 P3 → P6(Movl)**

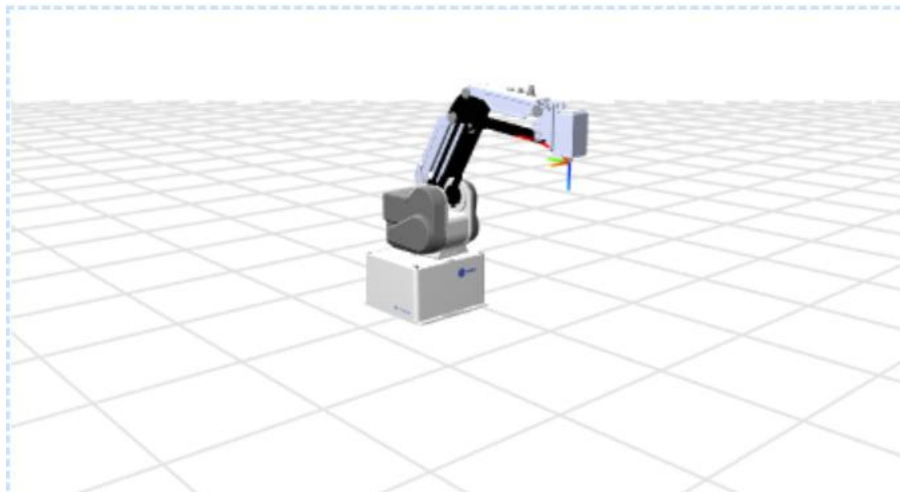
$$\Delta x = -19,22 - (-0,658) = -18,562$$

$$\Delta y = 370,71 - 257,05 = 113,66$$

$$\Delta z = 77,801 - 107,96 = -326,214$$

$$d_1 = \sqrt{(-18,562)^2 + (113,66)^2 + (-326,214)^2}$$

$$= \sqrt{344,547844 + 12918,5956 + 106.415,573796} = \sqrt{119.678,71724} = 345,94$$



Gambar 4. 9 Segmen 1 P3→ P6

- **Segmen 2 (P6 → P4) [MovL Linear]:**

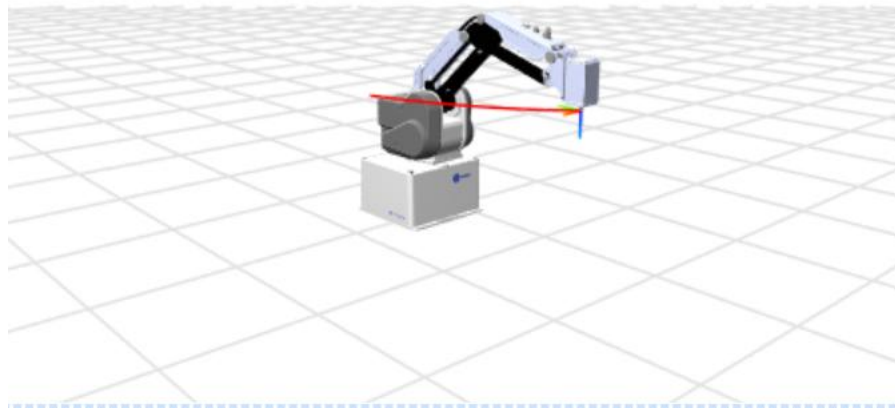
$$\Delta x = -21,10 - (-19,22) = 1,88$$

$$\Delta y = 404,60 - 370,71 = 33,89$$

$$\Delta z = 11,378 - 77,801 = -66,423$$

$$d_2 = \sqrt{(1,88)^2 + (33,89)^2 + (-66,423)^2}$$

$$= \sqrt{3,5344 + 1.148,5321 + 4.412,014929} = \sqrt{5.564,081429} = 74,59 \text{ mm}$$



Gambar 4. 10 Segmen 2 P6→ P4

- **Segmen 3 P4 → P6(Movl)**

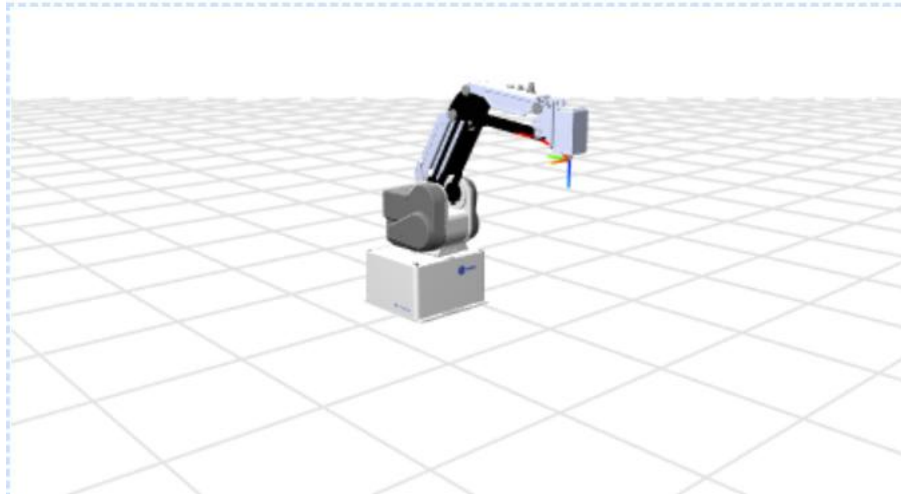
$$\Delta x = -19,22 - (-21,10) = -1,88$$

$$\Delta y = 370,71 - 404,60 = -33,89$$

$$\Delta z = 77,801 - 11,378 = 66,423$$

$$d_2 = \sqrt{(1,88)^2 + (33,89)^2 + (-66,423)^2}$$

$$= \sqrt{3,5344 + 1.148,5321 + 4.412,014929} = \sqrt{5.564,081429} = 74,59 \text{ mm}$$



Gambar 4. 11 Segmen 3 P4→ P6

- **Segmen 4 (P6 → P5) [Jump]:**

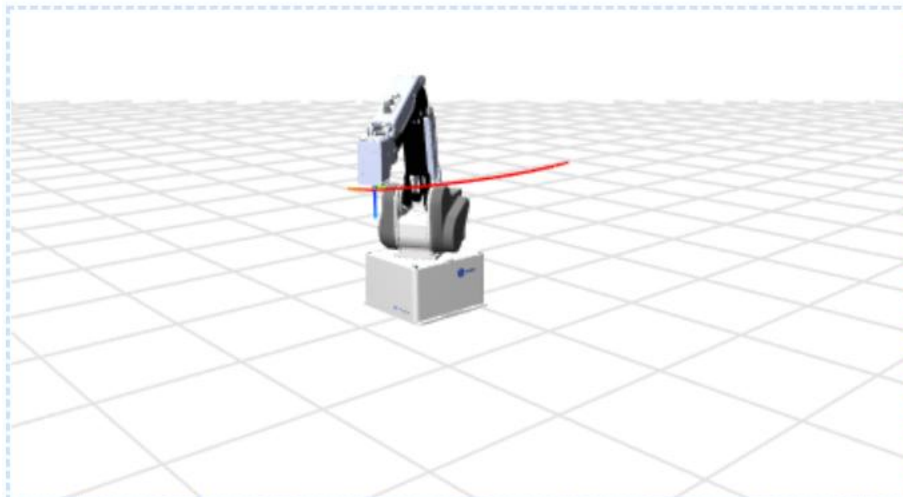
$$\Delta x = 256,66 - (-19,22) = 275,88$$

$$\Delta y = 166 - 370,71 = -204,71$$

$$\Delta z = 39,612 - 77,801 = -38,189$$

$$d_4 = \sqrt{(275,88)^2 + (-204,71)^2 + (-38,189)^2}$$

$$= \sqrt{76.109,7744 + 41.906,1841 + 1.458,399721} = \sqrt{119.474,35} = 345,65 \text{ mm}$$



Gambar 4. 12 Segmen 3 P6→ P5

Selanjutnya, seluruh jarak segmen pada Dobot MG400 dijumlahkan untuk memperoleh total jarak tempuh end-effector. Total waktu tunggu juga dihitung dengan mempertimbangkan waktu tunda yang terjadi pada proses pengambilan objek menggunakan vakum (vacuum gripper) serta waktu komunikasi modbus pada sistem kendali robot.

$$d_{total, MG400} = 345,94 + 74,59 + 74,59 + 345,65 = 840,77 \text{ mm}$$

Waktu tunggu vakum: 1,45 detik

Waktu tunggu modbus: 0,29 detik

Dengan cara yang sama seperti perhitungan pada Dobot E6, kecepatan rata-rata gerak Dobot MG400 diperoleh dari pembagian total jarak tempuh terhadap waktu gerak aktual, yaitu *cycle time* real dikurangi total waktu tunggu, untuk memvalidasi kesesuaian hasil perhitungan dengan data pengujian aktual.

Dari data real *cycle time* MG400 = 5 detik

$$t_{gerak} = 5.0 - 0.29 - 1.45 = 3.26 \text{ detik}$$

$$v_{rata-rata, MG400} = \frac{840.77}{3.26} = 257.9 \text{ mm/detik}$$

3. Rekapitulasi Validasi

Hasil perhitungan jarak tempuh, waktu tunggu, *cycle time*, serta kecepatan rata-rata pada Dobot E6 dan Dobot MG400 selanjutnya direkapitulasi ke dalam tabel berikut sebagai dasar validasi awal sebelum dilakukan optimasi lintasan.

Tabel 4. 3 hasil validasi awalan

Parameter	E6	MG400
Kecepatan setting	SpeedFactor(100%)	SpeedJ(80%)
Total jarak tempuh	1089,86 mm	840,77 mm
Total waktu tunggu	0.5 detik	1,74 detik
<i>Cycle time</i> real	7,25	5 detik
Waktu gerak terukur	6.75	3,26 detik
Kecepatan rata-rata	161,441 mm/s	257,9 mm/s
Kecepatan di 100%	226.5 mm/s	552.3 mm/s

Untuk memastikan keakuratan hasil perhitungan, dilakukan verifikasi ulang (cross-check) dengan menghitung kembali prediksi *cycle time* menggunakan kecepatan rata-rata yang telah diperoleh sebelumnya. Apabila hasil prediksi *cycle time* tersebut sesuai dengan *cycle time* real hasil pengujian, maka perhitungan jarak dan kecepatan yang dilakukan dinyatakan valid.

E6 prediksi *cycle time* jika kecepatan = 161,441 mm/s:

$$t_{prediksi} = \frac{1089.727}{161.441} + 0.5 = 6.75 + 0.5 = 7.25 \text{ detik}$$

MG400 prediksi *cycle time* jika kecepatan = 257,9 mm/s:

$$t_{prediksi} = \frac{840,77}{257,9} + 0.34 = 3.66 + 0.34 = 5.0 \text{ detik } \checkmark$$

4.2.2 Evolutionary Algorithm

Dalam penelitian ini, Evolutionary Algorithm dimanfaatkan sebagai alat perhitungan untuk menentukan koordinat yang optimal yang akan digunakan sebagai usulan koordinat terbaru untuk mendapatkan nilai *cycle time* yang optimal. Untuk mendapatkan nilai koordinat yang optimal diperlukan point, sumbu(X,Y,Z) Batas bawah serta atas, dan nilai actual dari setiap point.

1. Menentukan variabel yang diubah dan batasan

Pada perhitungan yang dilakukan ini, peneliti menggunakan rumus Euclidean, maka diperlukan perubahan dari data Lat-Long menjadi data X-Y, hal ini dikarenakan Eucclidean hanya valid di bidang datar, sedangkan data Lat-Long berada pada permukaan melengkung. Jadi jika menggunakan rumus Haversine maka tidak memerlukan perubahan datanya. Perubahan data ini dibantu dengan GeoTekno.com. pada gambar 4. 4 merupakan perubahan data yang dilakukan untuk menjadi data X-Y.

Tabel 4. 4 Variabel aktual dan batasan

Titik	Sumbu	Batas Bawah	Batas Atas	Nilai Awal (Aktual)
P13	X	273.803	277.228	273.803
P13	Y	-101,74	-34,938	-34,938
P13	Z	238.634	303.206	303.206
P2	X	274.285	277.228	277.228
P2	Y	-163,159	-101,74	-101,74
P2	Z	233.125	238.634	238.634
P18	X	0	0,706	0
P18	Y	-419,448	-255,138	-255,138
P18	Z	158.206	288.986	288.986
P6	Y	-419,448	-255,138	-419,448
p6	Z	288	158	158

2. Menentukan nilai tetap

Sebelum menentukan variabel yang akan dioptimasi, terlebih dahulu ditentukan titik-titik koordinat yang nilainya bersifat tetap (tidak diubah) selama proses optimasi berlangsung, karena posisi tersebut berkaitan langsung dengan proses pengambilan dan penempatan objek

(pick and place) yang tidak dapat diubah tanpa mengubah fungsi kerja robot. Berikut merupakan titik-titik beserta nilai tetap yang digunakan.

Tabel 4. 5 Nilai Tetap			
Nilai Tetap			
Label	X	Y	Z
P3 (Object1Take) TETAP	274	-163,159	233
P4 (Object1Up) TETAP	274	-165,573	258
P6 X TETAP	0,706	-	-

3. Membuat tabel sel variabel

Selanjutnya dibuat tabel sel variabel yang berfungsi sebagai referensi sel pada lembar kerja Excel yang nilainya akan diubah oleh Solver selama proses pencarian solusi optimal. Setiap sel variabel merepresentasikan satu koordinat (X, Y, atau Z) dari titik-titik yang telah ditentukan batas bawah dan batas atasnya pada tahap sebelumnya.

Tabel 4. 6 Tabel variabel		
Sel Variabel		
Sel	Label	Nilai Awal
C3	X P13	274
C4	Y P13	-101
C5	Z P13	238
C6	X P2	274
C7	Y P2	-134,9256336
C8	Z P2	235
C9	X P18	0,706
C10	Y P18	-331,5923129
C11	Z P18	288
C12	Z P6	200
C13	Y-P6	-400

4. Membuat formula jarak 6 segmen

Setelah sel variabel ditentukan, dibuat formula perhitungan jarak untuk keenam segmen lintasan menggunakan rumus jarak Euclidean berdasarkan referensi sel variabel yang telah dibuat. Formula ini akan memperbarui nilai jarak setiap segmen secara otomatis seiring dengan perubahan nilai koordinat yang dilakukan oleh Solver pada proses optimasi.

Tabel 4. 7 Formula jarak 6 segmen		
Segmen	Formula	Hasil
Segmen 1	$\text{SQRT}((P13(x)-P1(x))^2+(P13(y)-P1(y))^2+(P13(z)-P1(z))^2)$	410,0276
Segmen 2	$\text{SQRT}((P2(x)-P13(x))^2+(P2(y)-P13(y))^2+(P2(z)-P13(z))^2)$	92,97194

Segmen	Formula	Hasil
Segmen 3	$\text{SQRT}((P3(x)-P2(x))^2+(P3(y)-P2(y))^2+(P3(z)-P2(z))^2)$	61,73576
Segmen 4	$\text{SQRT}((P4(x)-P3(x))^2+(P4(y)-P3(y))^2+(P4(z)-P3(z))^2)$	24,69242
Segmen 5	$\text{SQRT}((P18(x)-P4(x))^2+(P18(y)-P4(y))^2+(P18(z)-P4(z))^2)$	290,3068
Segmen 6	$\text{SQRT}((P6(x)-P18(x))^2+(P6(y)-P18(y))^2+(P6(z)-P18(z))^2)$	210,1324

5. Membuat formula *Objective (cycle time)*

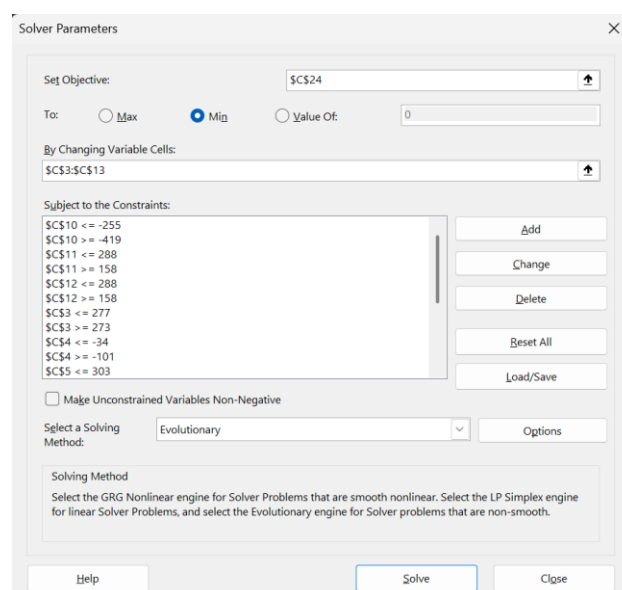
Formula *objective* (fungsi tujuan) dibuat dengan menjumlahkan seluruh jarak pada keenam segmen untuk memperoleh total jarak tempuh, kemudian nilai tersebut dibagi dengan kecepatan rata-rata gerak robot untuk memperoleh estimasi *cycle time*. Fungsi tujuan inilah yang selanjutnya akan diminimalkan oleh Solver melalui proses pencarian nilai koordinat yang optimal.

Total jarak = segmen 1 + segmen 2 + segmen 3 + segmen 4 + segmen 5 + segmen 6

Cycle time total = (total jarak / 161,441)

6. Membuat Konfigurasi Solver

Konfigurasi Solver dilakukan dengan menetapkan sel formula *cycle time (objective)* sebagai target yang akan diminimalkan (set *objective* to min), sel-sel variabel yang telah dibuat sebagai sel yang nilainya diubah (by changing variable cells), serta menambahkan batasan (constraints) berupa rentang batas bawah dan batas atas pada setiap variabel sesuai dengan tabel sebelumnya. Metode penyelesaian yang digunakan adalah Evolutionary, yang sesuai untuk permasalahan optimasi non-linear dengan banyak variabel dan batasan.



Gambar 4. 13 Konfigurasi excel

7. Membuat perbandingan hasil

Setelah proses solving selesai dijalankan, Solver akan menghasilkan nilai-nilai koordinat baru pada sel variabel yang telah memenuhi fungsi tujuan optimal. Nilai-nilai tersebut kemudian dibandingkan dengan nilai awal (aktual) sebelum optimasi untuk melihat perubahan koordinat yang terjadi, sebagaimana ditampilkan pada tabel berikut.

Tabel 4. 8 Perbandingan Hasil

Sel	Label	Nilai Awal	Nilai Solver
C3	X P13	274	274
C4	Y P13	-34,938	-101
C5	Z P13	303	238
C6	X P2	277	274
C7	Y P2	-101,74	-134,926
C8	Z P2	239	235
C9	X P18	0	0,706
C10	Y P18	-255,138	-331,592
C11	Z P18	289	288
C12	Z P6	158	200
C13	Y-P6	-419,448	-400

4.2.3 Trajectory Optimization(usulan)

Setelah diperoleh nilai koordinat hasil optimasi dari Solver, selanjutnya dilakukan validasi teoritis terhadap koordinat usulan tersebut dengan melakukan kembali perhitungan jarak tempuh dan *cycle time* menggunakan metode yang sama seperti pada perhitungan awal. Validasi ini dilakukan untuk membuktikan bahwa koordinat hasil optimasi mampu menghasilkan *cycle time* yang lebih singkat dibandingkan dengan *cycle time* pada kondisi awal.

1. Perhitungan E6

Perhitungan validasi pada Dobot E6 dilakukan dengan menggunakan urutan gerakan yang sama seperti perhitungan sebelumnya, namun dengan koordinat titik yang telah diperbarui berdasarkan hasil optimasi Solver, sebagaimana berikut.

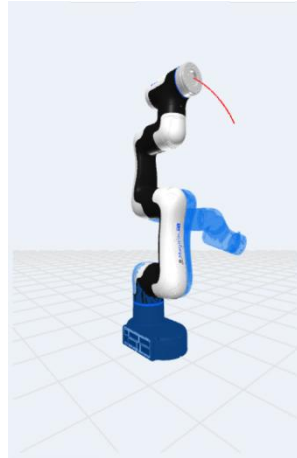
Urutan Gerakan dari Kode (Object1, DI(3)):

P1 → MovJ(P13) → MovJ(P2) → MovJ(P3) → Wait(250ms) → DO(9,1) → Wait(250ms) → MovL(P4) → MovJ(P18) → MovJ(P6) → [Wait(2200ms) → handover MG400]

Koordinat dari point.json (E6):

Jarak setiap segmen dihitung kembali menggunakan rumus Euclidean yang sama, dengan menggunakan koordinat titik hasil optimasi (nilai solver) sebagai data perhitungan, sehingga diperoleh hasil sebagai berikut.

- **Segmen 1 P1 → P13**



Gambar 4. 14 Segmen 1 P1→ P13

$$\Delta x = 0,000 - 274,28 = 274,28$$

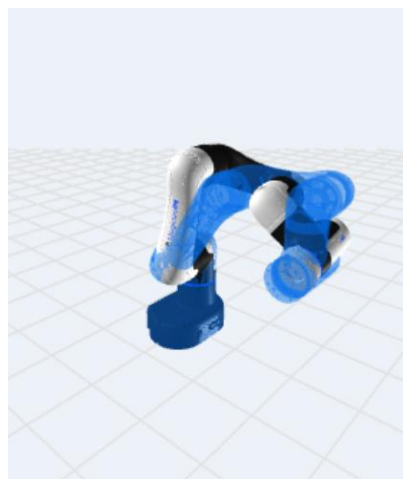
$$\Delta y = 0 - (-101) = 101$$

$$\Delta z = 0 - 238 = -238$$

$$d_1 = \sqrt{(-274,28)^2 + (101)^2 + (-238)^2}$$

$$= \sqrt{75.229,51 + 10.201 + 56.644} = \sqrt{142077,28} = 376,93 \text{ mm}$$

- **Segmen 2 P13 → P2**



Gambar 4. 15 Segmen 2 P13→ P12

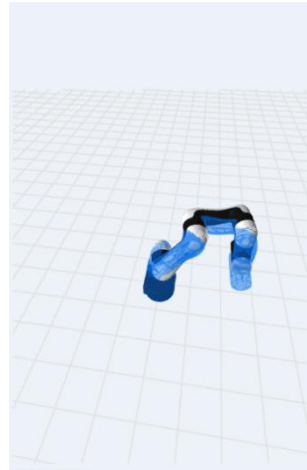
$$\Delta x = 274,28 - 274,28 = 0$$

$$\Delta y = -134,92 - (-101) = -33,92$$

$$\Delta z = 235 - 238 = -3$$

$$\begin{aligned} d_2 &= \sqrt{(0)^2 + (-33,92)^2 + (-3)^2} \\ &= \sqrt{0 + 1.150,56 + 9} = \sqrt{1.159,56} = 34,02\text{mm} \end{aligned}$$

- **Segmen 3 P2 → P3**



Gambar 4. 16 Segmen 3 P2→ P3

$$\Delta x = 274,28 - 274,28 = 0$$

$$\Delta y = -163,159 - (-134,92) = -163,159 + 134,92 = -28,239$$

$$\Delta z = 233,125 - 235 = -1,875$$

$$\begin{aligned} d_3 &= \sqrt{(0)^2 + (-28,239)^2 + (-1,875)^2} \\ &= \sqrt{0 + 797,44 + 3,51} = \sqrt{802,026} = 28,32 \text{ mm} \end{aligned}$$

- **Segmen 4 P3 → P4 (MovL)**



Gambar 4. 17 Segmen 4 P3→ P4

$$\Delta x = 274,367 - 274,285 = 0,082$$

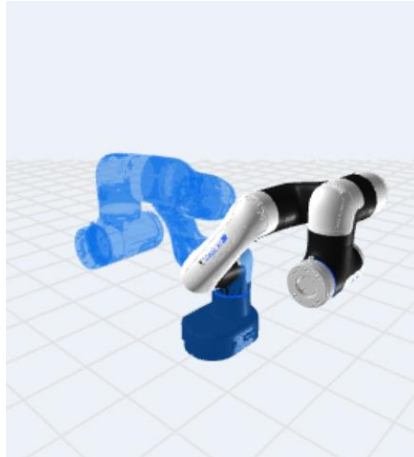
$$\Delta y = -165,573 - (-163,159) = -165,573 + 163,159 = -2,414$$

$$\Delta z = 257,699 - 233,125 = 24,574$$

$$d_4 = \sqrt{(0,082)^2 + (-2,414)^2 + (24,574)^2}$$

$$= \sqrt{0,007 + 5,827 + 603,88} = \sqrt{609,71} = 24,69 \text{ mm}$$

- **Segmen 5 P4 → P18**



Gambar 4. 18 Segmen 5 P4→ P18

$$\Delta x = 0,706 - 274,367 = -273,661$$

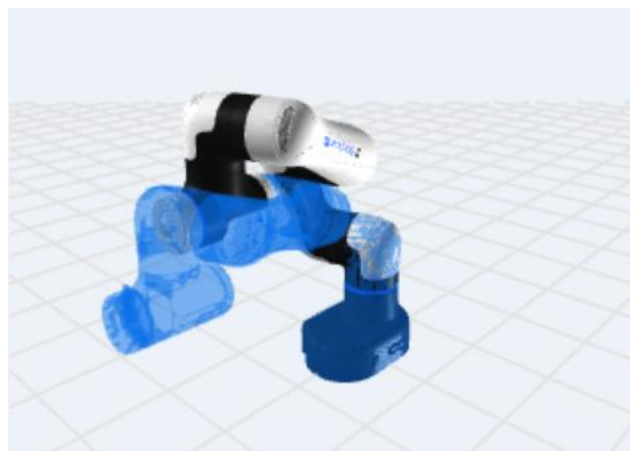
$$\Delta y = -331,59 - (-165,573) = -331,59 + 165,573 = -166,017$$

$$\Delta z = 288 - 257,699 = 30,30$$

$$d_5 = \sqrt{(-273,661)^2 + (-166,017)^2 + (30,30)^2}$$

$$= \sqrt{74.890,34 + 27.561,64 + 918,09} = \sqrt{103370,90} = 321,51 \text{ mm}$$

- **Segmen 6 P18 → P6**



Gambar 4. 19 Segmen 5 P18→ P6

$$\Delta x = 0,706 - 0,706 = 0$$

$$\Delta y = -400 - (-331,59) = -400 + 331,59 = -68,41$$

$$\Delta z = 200 - 288 = -88$$

$$d_6 = \sqrt{(0)^2 + (-68,41)^2 + (-88)^2}$$

$$= \sqrt{0 + 4.679,92 + 7.744} = \sqrt{12423,61} = 111,46 \text{ mm}$$

Seluruh jarak segmen hasil perhitungan dengan koordinat usulan kemudian dijumlahkan untuk memperoleh total jarak tempuh end-effector, dengan memperhitungkan pula total waktu tunggu yang sama seperti pada perhitungan awal.

$$d_{total,E6} = 376,93 + 34,02 + 28,32 + 24,69 + 321,51 + 111,46 = 896,94 \text{ mm}$$

Waktu yang dibutuhkan: 7.15 detik (termasuk Waktu tunggu Griper)

Waktu tunggu (gripper saja):

$$t_{wait} = 250 + 250 = 500 \text{ ms} = 0.5 \text{ detik}$$

Kecepatan rata-rata gerak robot dengan koordinat usulan dihitung dengan cara yang sama, yaitu membagi total jarak tempuh dengan waktu gerak aktual, untuk selanjutnya digunakan sebagai dasar verifikasi *cycle time* hasil optimasi.

Dari data real skripsi: *cycle time* E6 = 7,15 detik

$$t_{gerak} = 7,15 - 0,5 = 6,65 \text{ detik}$$

$$v_{rata-rata,E6} = \frac{896,94}{6,65} = 134,87 \text{ mm/detik}$$

2. Perhitungan Mg400

Perhitungan validasi pada Dobot MG400 juga dilakukan dengan urutan gerakan yang sama seperti sebelumnya, dengan menggunakan koordinat titik hasil optimasi sebagai data perhitungan, sebagaimana berikut.

Urutan Gerakan dari Kode (Object1, Coil 0):

MovJ(P6) → MovL(P4) → DO(9,1) → MovJ(P6) → Jump(P5) → DO(9,0)

Dengan metode perhitungan jarak yang sama, jarak setiap segmen lintasan Dobot MG400 dihitung kembali menggunakan koordinat hasil optimasi, sehingga diperoleh hasil sebagai berikut.

- **Segmen 1 P3 → P6(Movl)**

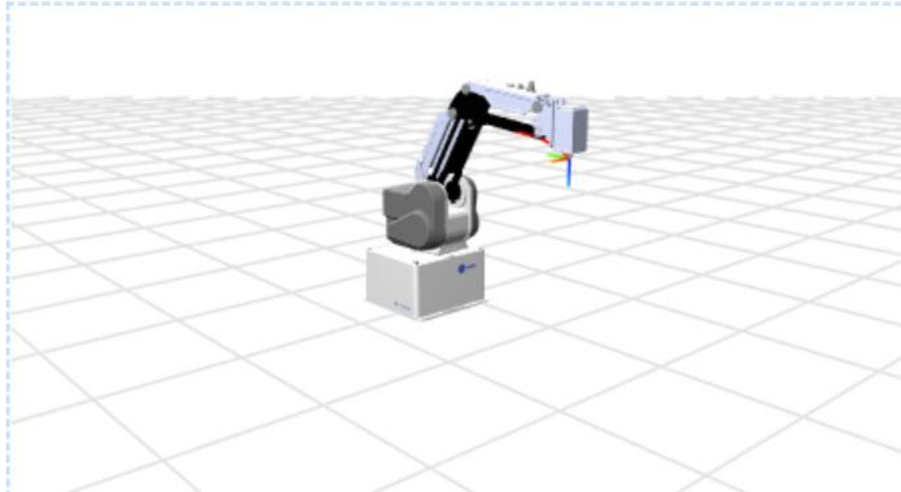
$$\Delta x = -19,22 - (-0,658) = -18,562$$

$$\Delta y = 370,71 - 257,05 = 113,66$$

$$\Delta z = 77,801 - 107,96 = -326,214$$

$$d_1 = \sqrt{(-18,562)^2 + (113,66)^2 + (-326,214)^2}$$

$$= \sqrt{344,547844 + 12918,5956 + 106.415,573796} = \sqrt{119.678,71724} = 345,94$$



Gambar 4. 20 Segmen 1 P3 → P6

- **Segmen 2 P6 → P4) [MovL Linear]:**

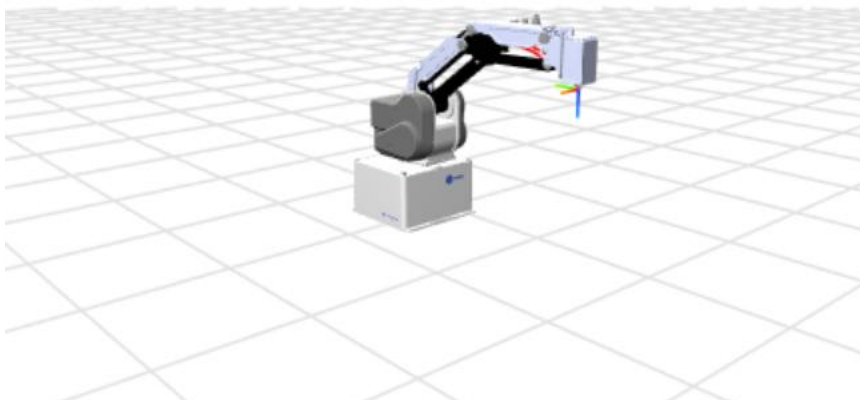
$$\Delta x = -21,2 - (-19,22) = -1,98$$

$$\Delta y = 424,47 - 370,71 = 53,76$$

$$\Delta z = 51,33 - 77,801 = -26,471$$

$$d_2 = \sqrt{(-1,98)^2 + (53,76)^2 + (-26,471)^2}$$

$$= \sqrt{3,92 + 2.890,13 + 700,71} = \sqrt{3.594,76} = 59,95 \text{ mm}$$



Gambar 4. 21 Segmen 2 P6 → P4

- **Segmen 3 P4 → P6(Movl)**

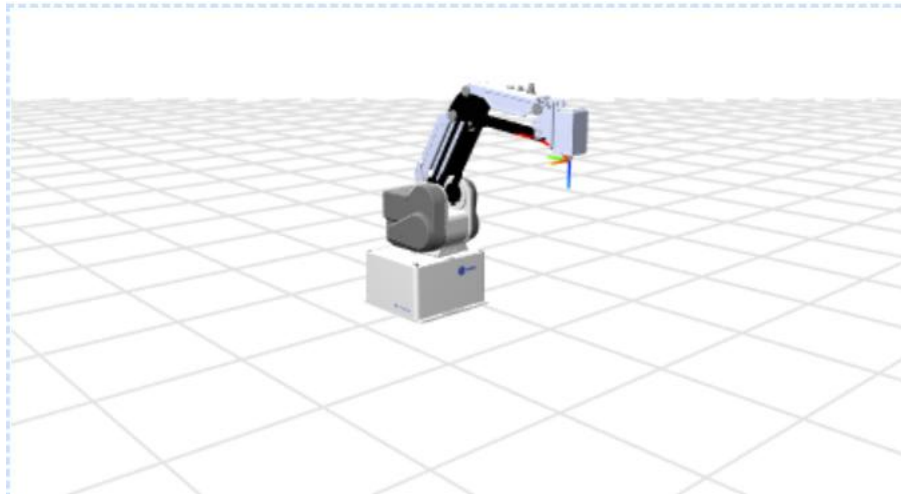
$$\Delta x = -19,22 - (-21,2) = 1,98$$

$$\Delta y = 370,71 - 404,60 = -53,76$$

$$\Delta z = 77,801 - 51,33 = 26,471$$

$$d_2 = \sqrt{(1,98)^2 + (-53,76)^2 + (26,471)^2}$$

$$= \sqrt{3,92 + 2.890,13 + 700,71} = \sqrt{3.594,76} = 59,95 \text{ mm}$$



Gambar 4. 22 Segmen 3 P4 → P6

- **Segmen 4 (P6 → P5) [Jump]:**

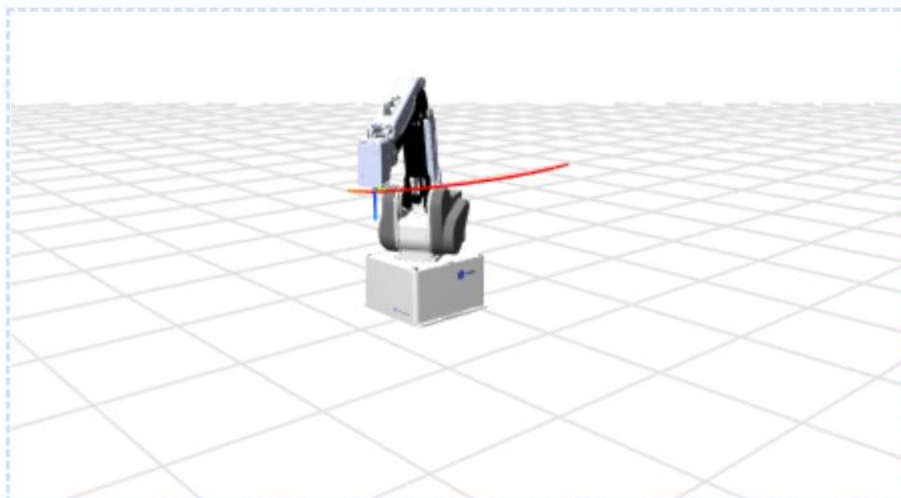
$$\Delta x = 256,66 - (-19,22) = 275,88$$

$$\Delta y = 166 - 370,71 = -204,71$$

$$\Delta z = 39,612 - 77,801 = -38,189$$

$$d_4 = \sqrt{(275,88)^2 + (-204,71)^2 + (-38,189)^2}$$

$$= \sqrt{76.109,7744 + 41.906,1841 + 1.458,399721} = \sqrt{119.474,35} = 345,65 \text{ mm}$$



Gambar 4. 23 Segmen 3 P6 → P5

Total jarak tempuh Dobot MG400 dengan koordinat usulan diperoleh dengan menjumlahkan seluruh jarak segmen, dengan memperhitungkan waktu tunggu vakum dan waktu komunikasi modbus yang sama seperti pada perhitungan sebelumnya.

$$d_{total, MG400} = 345,94 + 59,95 + 59,95 + 345,65 = 811,49 \text{ mm}$$

Waktu tunggu vakum: 1,45 detik

Waktu tunggu modbus: 0,29 detik

Kecepatan rata-rata gerak Dobot MG400 dengan koordinat usulan dihitung dengan cara yang sama seperti sebelumnya, untuk selanjutnya dibandingkan dengan hasil perhitungan awal guna melihat pengaruh optimasi terhadap kecepatan gerak robot.

Dari data real *cycle time* MG400 = 4,90 detik

$$t_{gerak} = 4,90 - 0,29 - 1,45 = 3,16 \text{ detik}$$

$$v_{rata-rata, MG400} = \frac{811,49}{3,16} = 256,80 \text{ mm/detik}$$

3. Rekapitulasi Validasi

Hasil perhitungan jarak tempuh, waktu tunggu, *cycle time*, serta kecepatan rata-rata pada Dobot E6 dan Dobot MG400 dengan menggunakan koordinat hasil optimasi (usulan) selanjutnya direkapitulasi pada tabel berikut, sebagai pembandingan terhadap hasil rekapitulasi sebelum dilakukan optimasi.

Tabel 4. 9 Validasi Usulan

Parameter	E6	MG400
Kecepatan setting	SpeedFactor(100%)	SpeedJ(80%)
Total jarak tempuh	896,94 mm	811,49 mm
Total waktu tunggu	0.5 detik	1.74 detik
<i>Cycle time</i> real	7,15 detik	4,90 detik
Waktu gerak terukur	66,5 detik	3,16 detik
Kecepatan rata-rata	134,87 mm/s	256,80 mm/s
Kecepatan di 100%	226.5 mm/s	552.3 mm/s

Verifikasi ulang juga dilakukan terhadap hasil perhitungan dengan koordinat usulan, dengan menghitung kembali prediksi *cycle time* menggunakan kecepatan rata-rata hasil optimasi. Kesesuaian antara hasil prediksi dengan *cycle time* yang ditetapkan menunjukkan bahwa perhitungan koordinat hasil optimasi telah valid dan dapat diterapkan.

E6 prediksi *cycle time* jika kecepatan = 134,87 mm/s:

$$t_{prediksi} = \frac{896,94}{134,87} + 0.5 = 66,5 + 0.5 = 7,15 \text{ detik}$$

MG400 prediksi *cycle time* jika kecepatan = 256,80mm/s:

$$t_{prediksi} = \frac{811,49}{256,80} + 1.74 = 3,16 + 1.74 = 4,90 \text{ detik } \checkmark$$

Tabel 4. 10 Tabel Perbandingan validasi Usulan dan Awalan

Parameter	E6(awalan)	MG400(awalan)	E6(usulan)	MG400(usulan)
Kecepatan setting	SpeedFactor(100%)	SpeedJ(80%)	SpeedFactor(100%)	SpeedJ(80%)
Total jarak tempuh	1089,86 mm	840,77 mm	896,94 mm	811,49 mm
Total waktu tunggu	0.5 detik	1,74 detik	0.5 detik	1.74 detik
<i>Cycle time</i> real	7,25	5 detik	7,15 detik	4,90 detik
Waktu gerak terukur	6.75	3,26 detik	66,5 detik	3,16 detik
Kecepatan rata-rata	161,441 mm/s	257,9 mm/s	134,87 mm/s	256,80 mm/s
Kecepatan di 100%	226.5 mm/s	552.3 mm/s	226.5 mm/s	552.3 mm/s
Total Waktu	12,25 Detik		12,05 Detik	

BAB V

PEMBAHASAN

5.1. Analisis Hasil

Pada bagian ini akan membahas secara mendalam hasil perhitungan dan validasi yang telah diselesaikan sebelumnya, dengan fokus pada perbandingan kinerja gerak Dobot Magician E6 dan Dobot MG400 sebelum (awalan) dan setelah (usulan) dilakukan optimasi lintasan menggunakan pendekatan *Evolutionary Algorithm*. Pembahasan disusun berdasarkan lima parameter utama yang menjadi indikator keberhasilan optimasi, yaitu total jarak tempuh, *cycle time* real, waktu gerak terukur, kecepatan rata-rata, dan total waktu siklus produksi gabungan. Seluruh data yang dianalisis pada bab ini mengacu pada hasil rekapitulasi yang telah disajikan.

5.1.1. Perbandingan Total Jarak Tempuh

Total jarak tempuh merupakan panjang lintasan *end-effector* yang ditempuh oleh masing-masing *Dobot Collaborative* dalam menyelesaikan satu siklus kerja penuh, mulai dari titik awal (*home position*) hingga seluruh titik tujuan (*target points*) yang telah ditentukan. Parameter ini menjadi indikator paling fundamental dari keberhasilan optimasi lintasan, karena perubahan koordinat hasil *Evolutionary Algorithm* akan tercermin secara langsung pada besaran jarak yang dihasilkan.

Tabel 5. 1 Perbandingan Total Jarak Tempuh

Robot	Awalan	Usulan	Selisih	Penurunan (%)
Dobot E6	1089,86 mm	896,94 mm	192,92 mm	17,70%
Dobot MG400	840,77 mm	811,49 mm	29,28 mm	3,48%

Berdasarkan Tabel 5.1, kedua dobot mengalami penurunan total jarak tempuh setelah koordinat usulan diterapkan. Pada Dobot E6, total jarak tempuh berkurang dari 1089,86 mm menjadi 896,94 mm, atau setara dengan penurunan sebesar 192,92 mm (17,70%). Penurunan ini tergolong signifikan dan mengindikasikan bahwa lintasan *default* pada Dobot E6 masih menyimpan banyak gerakan yang kurang efisien, terutama pada segmen-segmen yang melibatkan perubahan orientasi *end-effector* akibat konfigurasi enam derajat kebebasan (6-DOF) yang dimilikinya, sehingga ruang solusi yang dapat dieksplorasi oleh algoritma evolusioner menjadi lebih luas.

Berbeda dengan Dobot E6, Dobot MG400 dengan konfigurasi empat derajat kebebasan (4-DOF) hanya mengalami penurunan jarak tempuh dari 840,77 mm menjadi 811,49 mm, yaitu sebesar 29,28 mm atau 3,48%. Nilai penurunan yang lebih kecil ini menunjukkan bahwa

lintasan *default* pada MG400 sebenarnya sudah relatif efisien, karena ruang gerak yang lebih terbatas (jumlah DOF lebih sedikit) membuat ruang optimasi koordinat yang tersedia bagi algoritma evolusioner menjadi lebih kecil dibandingkan dengan E6. Perbedaan besaran penurunan ini mengindikasikan bahwa kompleksitas kinematika robot turut memengaruhi besarnya potensi optimasi yang dapat dicapai robot dengan derajat kebebasan yang lebih tinggi memiliki ruang solusi yang lebih luas untuk dieksplorasi, sehingga peluang ditemukannya lintasan yang lebih singkat menjadi lebih besar.

5.1.2. Perbandingan *Cycle time Real*

Cycle time real merupakan parameter kunci keberhasilan sistem produksi *make-to-stock*, karena setiap detik yang berhasil dihemat dalam satu siklus akan terakumulasi menjadi penghematan waktu produksi yang besar pada skala operasional penuh.

Tabel 5. 2 Perbandingan *Cycle time Real*

Robot	Awalan	Usulan	Selisih	Penurunan (%)
Dobot E6	7,25 detik	7,15 detik	0,10 detik	1,38%
Dobot MG400	5,00 detik	4,90 detik	0,10 detik	2,00%

Pada Dobot E6, *cycle time real* menurun dari 7,25 detik menjadi 7,15 detik, atau berkurang 0,10 detik (1,38%). Pada Dobot MG400, *cycle time real* menurun dari 5,00 detik menjadi 4,90 detik, atau berkurang 0,10 detik (2,00%). Meskipun secara nominal penurunan pada kedua dobot bernilai sama, yaitu 0,10 detik, secara proporsional penurunan pada MG400 (2,00%) sedikit lebih besar dibandingkan E6 (1,38%) karena nilai *cycle time* dasar MG400 memang lebih singkat dibandingkan E6, sehingga selisih waktu yang sama akan memberikan dampak persentase yang lebih besar.

Penurunan *cycle time* yang konsisten pada kedua dobot ini menjadi bukti bahwa optimasi koordinat hasil *Evolutionary Algorithm* berhasil diterjemahkan menjadi perbaikan kinerja aktual di lapangan, bukan hanya pada simulasi maupun perhitungan teoretis semata, sebagaimana telah dikonfirmasi melalui proses verifikasi ulang (*cross-check*) pada Bab IV. Sekalipun nilai pengurangan 0,10 detik tampak kecil apabila dilihat dari satu siklus saja, prinsip *continuous improvement* (*Kaizen*) menegaskan bahwa perbaikan kecil yang konsisten pada setiap siklus produksi akan menghasilkan dampak kumulatif yang besar dalam jangka panjang, terutama pada sistem *make-to-stock* yang memproduksi produk identik secara berulang.

5.1.3. Perbandingan Waktu Gerak Terukur

Waktu gerak terukur (*tgerak*) merupakan komponen waktu yang merepresentasikan durasi pergerakan fisik murni *end-effector* dobot, yang diperoleh dengan mengurangi *cycle time* real dengan total waktu tunggu (waktu tunggu *vacuum gripper* dan waktu komunikasi *modbus* antar-*controller*). Parameter ini penting untuk diisolasi karena waktu tunggu bersifat konstan dan ditentukan oleh karakteristik *hardware* serta protokol komunikasi, sehingga tidak dipengaruhi oleh optimasi koordinat lintasan.

Tabel 5. 3 Perbandingan Waktu Gerak Terukur

Robot	Awalan	Usulan	Selisih	Penurunan (%)
Dobot E6	6,75 detik	6,65 detik	0,10 detik	1,48%
Dobot MG400	3,26 detik	3,16 detik	0,10 detik	3,07%

Pada Dobot E6, waktu gerak terukur menurun dari 6,75 detik menjadi 6,65 detik (berkurang 0,10 detik atau 1,48%). Pada Dobot MG400, waktu gerak terukur menurun dari 3,26 detik menjadi 3,16 detik (berkurang 0,10 detik atau 3,07%). Nilai penurunan waktu gerak terukur pada kedua dobot identik dengan nilai penurunan *cycle time* real, karena waktu tunggu (0,5 detik untuk E6 dan 1,74 detik untuk MG400) bersifat konstan dan tidak berubah antara skema awalan maupun usulan. Hal ini mengonfirmasi bahwa seluruh perbaikan *cycle time* yang terjadi murni bersumber dari efisiensi pergerakan fisik robot hasil optimasi lintasan, bukan dari faktor lain seperti perubahan waktu tunggu sistem.

5.1.4. Perbandingan Kecepatan Rata-rata

Kecepatan rata-rata (*vrata-rata*) dihitung dari rasio total jarak tempuh terhadap waktu gerak terukur ($v = d/t$), sehingga nilainya dipengaruhi secara simultan oleh perubahan pada kedua parameter tersebut.

Tabel 5. 4 Perbandingan Kecepatan Rata-rata

Robot	Awalan	Usulan	Selisih	Perubahan (%)
Dobot E6	161,441 mm/detik	134,87 mm/detik	-26,571 mm/detik	-16,46%
Dobot MG400	257,90 mm/detik	256,80 mm/detik	-1,10 mm/detik	-0,43%

Berbeda dengan ketiga parameter sebelumnya yang seluruhnya menunjukkan tren penurunan sebagai bentuk perbaikan, parameter kecepatan rata-rata justru menunjukkan tren yang menurun pada kedua dobot. Pada Dobot E6, kecepatan rata-rata turun dari 161,441 mm/detik menjadi 134,87 mm/detik, atau berkurang 26,571 mm/detik (16,46%). Pada Dobot

MG400, kecepatan rata-rata turun tipis dari 257,90 mm/detik menjadi 256,80 mm/detik, atau berkurang 1,10 mm/detik (0,43%).

Penurunan kecepatan rata-rata ini bukan merupakan indikasi penurunan kinerja, melainkan konsekuensi matematis dari optimasi lintasan itu sendiri. Pada Dobot E6, total jarak tempuh berkurang signifikan sebesar 17,70% (subbab 5.1), sementara waktu gerak terukur hanya berkurang 1,48% (subbab 5.3). Karena kecepatan rata-rata dihitung dari rasio jarak terhadap waktu, penurunan jarak yang jauh lebih besar dibandingkan penurunan waktu menyebabkan nilai rasio tersebut ikut menurun, meskipun *cycle time* keseluruhan tetap mengalami perbaikan. Pola yang sama, meskipun dengan magnitudo lebih kecil, juga terjadi pada Dobot MG400.

Apabila dibandingkan dengan kecepatan maksimum pada *setting* 100% (226,5 mm/detik untuk E6 dan 552,3 mm/detik untuk MG400), tingkat utilisasi kecepatan E6 menurun dari 71,28% menjadi 59,55%, sedangkan utilisasi kecepatan MG400 relatif stabil dari 46,69% menjadi 46,49%. Temuan ini menegaskan bahwa indikator utama keberhasilan optimasi pada penelitian ini bukan terletak pada peningkatan kecepatan rata-rata, melainkan pada minimasi total jarak tempuh dan *cycle time* secara keseluruhan.

5.1.5. Perbandingan Total Waktu

Total waktu pada Tabel 4.8 merepresentasikan akumulasi *cycle time* real Dobot E6 dan Dobot MG400 dalam satu siklus produksi penuh yang melibatkan proses *handover* objek antara kedua *Dobot Collaborative* pada satu stasiun kerja.

Tabel 5. 5 Perbandingan Total Waktu Siklus Gabungan

Skema	Total Waktu Siklus	Selisih
Awalan	12,25 detik	
Usulan	12,05 detik	0,20 detik (1,63%)

Total waktu siklus gabungan menurun dari 12,25 detik pada skema awalan menjadi 12,05 detik pada skema usulan, atau berkurang sebesar 0,20 detik (1,63%). Penurunan ini merupakan akumulasi langsung dari perbaikan *cycle time* real pada masing-masing dobot (subbab 5.2), yaitu 0,10 detik dari Dobot E6 dan 0,10 detik dari Dobot MG400.

Meskipun penurunan 0,20 detik per siklus tampak kecil apabila dilihat dari satu kali produksi saja, dampaknya menjadi signifikan apabila diakumulasikan dalam skala operasional yang berkelanjutan, sesuai dengan karakteristik sistem *make-to-stock* yang. Sebagai ilustrasi, apabila *Dobot Collaborative* dioperasikan secara kontinu selama 8 jam kerja (28.800 detik) per hari, maka skema awalan mampu menyelesaikan $28.800 \div 12,25 \approx 2.351$ unit dalam siklus produksi, sedangkan skema usulan mampu menyelesaikan $28.800 \div 12,05 \approx 2.390$ unit dalam

siklus produksi. Selisih tersebut setara dengan 39 unit dalam siklus produksi tambahan per hari kerja, atau peningkatan kapasitas produksi harian sebesar kurang lebih 1,66%, tanpa memerlukan penambahan satu pun sumber daya fisik maupun investasi *hardware* baru. Temuan ini sejalan dengan prinsip yang dikemukakan oleh Lorenz et al. (2021) bahwa reduksi *cycle time* pada sistem *make-to-stock* berkontribusi langsung terhadap peningkatan kapasitas produksi.

5.1.6. Rekapitulasi Hasil Pembahasan

Untuk memberikan gambaran yang lebih ringkas, hasil pembahasan kelima parameter di atas direkapitulasi dalam bentuk persentase perubahan berikut.

Tabel 5.6 Rekapitulasi Persentase Perubahan Tiap Parameter

Parameter	Dobot E6	Dobot MG400
Total Jarak Tempuh	-17,70%	-3,48%
<i>Cycle time</i> Real	-1,38%	-2,00%
Waktu Gerak Terukur	-1,48%	-3,07%
Kecepatan Rata-rata	-16,46%	-0,43%
Total Waktu Siklus Gabungan (E6 + MG400)	-1,63%	

Secara keseluruhan, hasil pembahasan pada kelima parameter tersebut menunjukkan bahwa penerapan *Evolutionary Algorithm* dalam *trajectory optimization* berhasil memperbaiki kinerja gerak Dobot E6 dan Dobot MG400 secara konsisten. Tiga parameter utama, yaitu total jarak tempuh, *cycle time* real, dan waktu gerak terukur, mengalami penurunan pada kedua dobot, yang menunjukkan keberhasilan optimasi lintasan. Kecepatan rata-rata yang menurun bukan merupakan kegagalan optimasi, melainkan konsekuensi logis dari penurunan jarak tempuh yang lebih besar dibandingkan penurunan waktu gerak. Adapun total waktu siklus gabungan yang menurun dari 12,25 detik menjadi 12,05 detik mengonfirmasi bahwa optimasi pada masing-masing dobot secara individual mampu memberikan kontribusi nyata terhadap efisiensi sistem kerja kolaboratif secara keseluruhan, sehingga dapat dikatakan bahwa tujuan penelitian sebagaimana dirumuskan pada Bab I telah tercapai.

Selain melakukan optimasi pergeseran koordinat menggunakan *Evolutionary Algorithm*, reduksi *cycle time* dan total jarak tempuh pada Dobot Magician E6 secara teoritis dapat dimaksimalkan lebih lanjut melalui modifikasi struktur lintasan, yaitu dengan menghilangkan titik-titik transit atau *via points*. Berdasarkan konfigurasi program awal, pergerakan lengan robot E6 melibatkan beberapa titik perantara, yaitu P13 (Transit) dan P2 (Object1Before) pada fase pengambilan objek, serta P18 (Before Drop) pada fase peletakan objek.

Penghapusan ketiga titik tersebut dari urutan program akan memaksa robot untuk melakukan pergerakan langsung (*direct path*) dari titik awal (P1) langsung menuju titik pengambilan (P3), dan dari titik angkat (P4) langsung menuju titik peletakan (P6). Secara analisis kinematis dan geometris, penghilangan titik-titik transit ini akan mempercepat *cycle time* dan memangkas jarak tempuh total

5.1.7. Analisis Dampak Ekonomis

Selain dampak teknis berupa reduksi *cycle time* dan peningkatan kapasitas produksi, hasil optimasi *Evolutionary Algorithm* pada sistem *Dobot Collaborative* ini juga dapat diterjemahkan ke dalam estimasi dampak ekonomis, sebagai gambaran nilai tambah yang dapat diperoleh apabila hasil penelitian ini diterapkan pada skala operasional produksi yang sesungguhnya. *Cycle time* sendiri telah lama diposisikan sebagai salah satu indikator ekonomis dan teknis utama dalam menilai kinerja proses produksi pada industri manufaktur, sehingga setiap satuan waktu yang berhasil direduksi berpotensi diterjemahkan secara langsung menjadi nilai tambah operasional maupun finansial (Taifa & Vhora, 2019).

a. Dasar Perhitungan

Berdasarkan hasil sebelumnya, penerapan koordinat usulan menghasilkan tambahan kapasitas produksi sebesar 39 siklus per hari kerja (8 jam), atau peningkatan kapasitas sebesar $\pm 1,66\%$ tanpa penambahan sumber daya fisik maupun investasi *hardware* baru. Nilai ini yang menjadi dasar estimasi dampak ekonomis.

b. Formulasi Estimasi Dampak Ekonomis

Estimasi potensi tambahan keuntungan pada penelitian ini didekati menggunakan konsep margin kontribusi (*contribution margin*), yaitu selisih antara harga jual dan biaya variabel produksi per unit. Pendekatan ini dipilih karena secara klasik telah digunakan sebagai kerangka analisis untuk menerjemahkan pemanfaatan kapasitas produksi tambahan menjadi nilai ekonomis tanpa perlu melibatkan perhitungan biaya tetap yang pada dasarnya tidak berubah akibat optimasi lintasan (Horngren, 1967). Kerangka ini juga selaras dengan konsep *kaizen costing*, yang memandang perbaikan proses secara bertahap termasuk perbaikan teknis seperti optimasi lintasan robot pada penelitian ini sebagai sumber penghematan biaya dan peningkatan margin yang terakumulasi dari waktu ke waktu tanpa mengubah struktur biaya produksi secara mendasar (Modarress et al., 2005).

Estimasi potensi tambahan keuntungan dirumuskan sebagai berikut:

$$\text{Potensi Tambahan Keuntungan (Rp/hari)} = \Delta \text{Unit per hari} \times \text{Margin Kontribusi per Unit (Rp/unit)}$$

Keterangan: Δ Unit per hari = 39 unit; Margin Kontribusi per Unit = selisih antara harga jual dan biaya variabel produksi per unit produk yang diproses oleh sistem *Dobot Collaborative*. Untuk proyeksi bulanan dan tahunan:

$$\text{Potensi Tambahan Keuntungan (Rp/bulan)} = \text{Potensi Tambahan Keuntungan (Rp/hari)} \times \text{Hari Kerja Efektif per Bulan}$$

$$\text{Potensi Tambahan Keuntungan (Rp/tahun)} = \text{Potensi Tambahan Keuntungan (Rp/bulan)} \times 12$$

c. Efisiensi Tanpa Investasi Modal (Zero Capital Expenditure)

Hal yang menjadikan hasil optimasi ini bernilai secara ekonomis bukan hanya dari sisi nominal tambahan keuntungan, melainkan dari sifatnya yang tidak memerlukan investasi modal (*capital expenditure*) sama sekali. Peningkatan kapasitas produksi sebesar 1,66% ini diperoleh murni dari pergeseran koordinat titik gerak robot yang sudah tersedia, tanpa penambahan unit robot baru, tanpa penambahan tenaga kerja, dan tanpa penambahan jam operasional.

Temuan ini sejalan dengan sejumlah studi kasus pada industri manufaktur yang membuktikan bahwa perbaikan proses tanpa investasi modal mampu menghasilkan peningkatan produktivitas yang substansial. Sebagai contoh, penataan ulang tata letak stasiun kerja tanpa satu pun tambahan modal terbukti mampu meningkatkan produktivitas hingga 71% pada proses fabrikasi *cross arm* tiang listrik (Degu, 2024). Bukti serupa juga ditunjukkan pada penerapan prinsip *Kaizen* di salah satu fasilitas Toyota, di mana optimasi proses kerja menggunakan fasilitas yang sudah tersedia berhasil meningkatkan output tahunan sebesar 13% sekaligus menghasilkan penghematan biaya yang signifikan tanpa investasi fasilitas baru (Abdilmouti, 2015).

Kedua temuan tersebut memperkuat argumen bahwa peningkatan kapasitas sebesar 1,66% pada sistem *Dobot Collaborative*, meski tampak kecil secara nominal, sepenuhnya bersumber dari efisiensi proses dan bukan dari penambahan sumber daya. Sebagai perbandingan, apabila perusahaan ingin mencapai peningkatan kapasitas yang setara melalui pendekatan konvensional seperti penambahan *shift* kerja, penambahan tenaga kerja, atau investasi unit robot tambahan dibutuhkan biaya investasi dan/atau biaya operasional tambahan yang jauh lebih besar dibandingkan biaya implementasi optimasi lintasan ini, yang secara praktis hanya membutuhkan waktu analisis dan konfigurasi ulang program robot. Dengan demikian, pendekatan ini dapat dikategorikan sebagai perbaikan dengan tingkat pengembalian investasi (*Return on Investment/ROI*) yang sangat tinggi, karena manfaat yang diperoleh tidak

diikuti biaya modal yang signifikan sejalan dengan prinsip *Kaizen* yang menekankan perbaikan berkelanjutan dengan sumber daya yang efisien.

d. Keterbatasan Analisis

Perlu digarisbawahi bahwa estimasi dampak ekonomis pada subbab ini bersifat ilustratif, mengingat penelitian dilakukan pada skala laboratorium dan belum menggunakan data biaya produksi maupun harga jual produk aktual dari sebuah lini produksi komersial. Untuk penelitian atau implementasi lanjutan pada industri yang sesungguhnya menerapkan sistem *make-to-stock* dengan *Dobot Collaborative*, disarankan menggunakan data biaya dan margin kontribusi riil perusahaan agar diperoleh perhitungan dampak ekonomis yang lebih akurat dan dapat dipertanggungjawabkan secara finansial.

BAB VI PENUTUP

6.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. *Evolutionary Algorithm* yang dijalankan melalui fitur Solver pada Microsoft Excel berhasil menentukan posisi koordinat optimal untuk tiga titik kritis dalam sistem *Dobot Collaborative*, yaitu *input point*, *collaboration point*, dan *output point*. Proses optimasi menghasilkan pergeseran koordinat pada sejumlah titik referensi, khususnya pada titik P13 (Transit), P2 (Object1Before), P18 (Before Drop), dan P6 (DropPoint) untuk Dobot E6, serta P4 pada Dobot MG400. Koordinat usulan yang dihasilkan tetap berada dalam rentang batas kinematika yang telah ditetapkan (*joint angle limits*, *workspace limit*, dan batas kecepatan mekanis), sehingga layak untuk diimplementasikan secara aktual tanpa risiko pelanggaran batasan fisik robot. Besarnya perubahan koordinat yang ditemukan pada Dobot E6 (penurunan jarak tempuh 17,70%) dibandingkan Dobot MG400 (3,48%) mengonfirmasi bahwa robot dengan derajat kebebasan lebih tinggi (6-DOF vs. 4-DOF) memiliki ruang solusi yang lebih luas untuk dieksplorasi, sehingga potensi optimasi yang dapat diraih pun lebih besar.
2. Optimasi koordinat menggunakan *Evolutionary Algorithm* berhasil mereduksi *cycle time* pada kedua robot secara konsisten. Dobot E6 mengalami penurunan *cycle time* dari 7,25 detik menjadi 7,15 detik (0,10 detik / 1,38%), sedangkan Dobot MG400 mengalami penurunan dari 5,00 detik menjadi 4,90 detik (↓0,10 detik / 2,00%). Secara gabungan, total waktu siklus sistem *Dobot Collaborative* berkurang dari 12,25 detik menjadi 12,05 detik, atau setara dengan reduksi 0,20 detik (1,63%) per siklus. Apabila diproyeksikan ke dalam skala operasional penuh 8 jam kerja per hari, penghematan ini menghasilkan 39 unit dalam siklus produksi tambahan per hari, atau peningkatan kapasitas produksi sebesar $\pm 1,66\%$ tanpa memerlukan penambahan perangkat keras maupun sumber daya fisik apapun. Hasil ini membuktikan bahwa pendekatan *Evolutionary Algorithm* dalam *trajectory optimization* mampu menghasilkan *continuous improvement* yang terukur dan signifikan secara kumulatif pada sistem produksi *make-to-stock*.

6.2. Saran

Berdasarkan hasil penelitian dan kesimpulan yang telah disampaikan, maka beberapa saran dapat diajukan sebagai berikut:

1. Bagi Sistem yang diteliti

- Koordinat hasil optimasi yang telah tervalidasi dapat segera dikonfigurasi ulang pada sistem *Dobot Collaborative* di laboratorium, menggantikan konfigurasi *default* saat ini, guna memperoleh efisiensi *cycle time* sebesar 1,63% secara nyata dalam setiap sesi demonstrasi maupun praktikum.
- Berdasarkan temuan pada Bab V, pergerakan Dobot E6 masih melibatkan titik-titik perantara (P13, P2, P18) yang secara kinematis dapat dieliminasi agar robot bergerak secara *direct path*. Laboratorium disarankan untuk mengevaluasi dan menghapus titik-titik transit tersebut dari kode program guna memaksimalkan reduksi *cycle time* lebih lanjut di luar pencapaian yang telah diraih penelitian ini.

2. Bagi Penelitian selanjutnya

- Penelitian ini membatasi optimasi pada sejumlah titik tertentu dengan nilai tetap pada beberapa titik *pick-and-place*. Penelitian selanjutnya disarankan untuk mempertimbangkan optimasi secara menyeluruh terhadap seluruh titik koordinat, termasuk *home position*, guna menggali potensi reduksi *cycle time* yang lebih maksimal.
- Untuk mengukur tingkat keoptimalan solusi yang dihasilkan, penelitian berikutnya dapat membandingkan kinerja *Evolutionary Algorithm* dengan algoritma metaheuristik lain seperti *Genetic Algorithm* (GA), *Whale Optimization Algorithm*, atau *Particle Swarm Optimization* (PSO) pada objek dan fungsi objektif yang sama, sehingga diperoleh rekomendasi algoritma terbaik untuk kasus *Dobot Collaborative* sejenis.

DAFTAR PUSTAKA

- Abdulmouti, H. (2015). *The Role of Kaizen (Continuous Improvement) in Improving Companies' Performance: A Case Study*.
- Akbari, A., Mozaffari, S., Singh, R., Ahmadi, M., & Alirezaee, S. (2023). *Movement Optimization of Robotic Arms for Energy and Time Reduction using Evolutionary Algorithms*.
- Beemsterboer, B., Land, M., & Teunter, R. (2017). Flexible lot sizing in hybrid make-to-order/make-to-stock production planning. *European Journal of Operational Research*, 260(3), 1014–1023. <https://doi.org/10.1016/j.ejor.2017.01.015>
- Degu, Y. M. (2024). Enhancing productivity through work study - A case of electric power pole cross arm fabrication. *Heliyon*, 10(12). <https://doi.org/10.1016/j.heliyon.2024.e32868>
- Dong, X., Wan, G., Zeng, P., Song, C., & Cui, S. (2024). Optimizing Robotic Task Sequencing and Trajectory Planning on the Basis of Deep Reinforcement Learning. *Biomimetics*, 9(1). <https://doi.org/10.3390/biomimetics9010010>
- Fakhri, O., Youssef, G., Nacy, S., Jameel Al-Tamimi, A. N., Hussein, O., Abbood, W. T., Abdullah, O. I., & AL-Karkhi, N. K. (2023). A Systematic Approach to Stable Grasping of a Two-Finger Robotic Hand Activated by Jamming of Granular Media. *Electronics (Switzerland)*, 12(8). <https://doi.org/10.3390/electronics12081902>
- Garriz, C., & Domingo, R. (2022). Trajectory Optimization in Terms of Energy and Performance of an Industrial Robot in the Manufacturing Industry. *Sensors*, 22(19). <https://doi.org/10.3390/s22197538>
- Gasparetto, A., & Zanutto, V. (2007a). A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, 42(4), 455–471. <https://doi.org/10.1016/j.mechmachtheory.2006.04.002>
- Gasparetto, A., & Zanutto, V. (2007b). A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, 42(4), 455–471. <https://doi.org/10.1016/j.mechmachtheory.2006.04.002>
- Hentout, A., Aouache, M., Maoudj, A., & Akli, I. (2019). Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. *Advanced Robotics*, 33(15–16), 764–799. <https://doi.org/10.1080/01691864.2019.1636714>
- Ji, Y., & Yu, J. (2025). Time-Impact Optimal Trajectory Planning for Wafer-Handling Robotic Arms Based on the Improved Snake Optimization Algorithm. *Sensors*, 25(6). <https://doi.org/10.3390/s25061872>
- Juříček, M., Parák, R., & Kůdela, J. (2023). Evolutionary Computation Techniques for Path Planning Problems in Industrial Robotics: A State-of-the-Art Review. In *Computation* (Vol. 11, Number 12). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/computation11120245>
- Katerina Kareska. (2024). *“Implementing KAIZEN for Continuous Improvement in Manufacturing Processes: A Quantitative Analysis of Efficiency and Productivity Gains”*
 Assistant prof.d-r Katerina Kareska.
[https://doi.org/Kareska,PhDKaterina,ImplementingKAIZENforContinuousImprovementinManufacturingProcesses:AQuantitativeAnalysisofEfficiencyandProductivityGains\(May28,2024\).AvailableatSSRN:https://ssrn.com/abstract=4844986orhttp://dx.doi.org/10.2139/ssrn.4844986](https://doi.org/Kareska,PhDKaterina,ImplementingKAIZENforContinuousImprovementinManufacturingProcesses:AQuantitativeAnalysisofEfficiencyandProductivityGains(May28,2024).AvailableatSSRN:https://ssrn.com/abstract=4844986orhttp://dx.doi.org/10.2139/ssrn.4844986)

- Liu, J., Wang, H., Li, X., Chen, K., & Li, C. (2023). Robotic arm trajectory optimization based on multiverse algorithm. *Mathematical Biosciences and Engineering*, 20(2), 2776–2792. <https://doi.org/10.3934/mbe.2023130>
- Liu, X., Jiang, D., Tao, B., Jiang, G., Sun, Y., Kong, J., Tong, X., Zhao, G., & Chen, B. (2022). Genetic Algorithm-Based Trajectory Optimization for Digital Twin Robots. *Frontiers in Bioengineering and Biotechnology*, 9. <https://doi.org/10.3389/fbioe.2021.793782>
- Lorenz, R., Senoner, J., Sihni, W., & Netland, T. (2021). Using process mining to improve productivity in make-to-stock manufacturing. *International Journal of Production Research*, 59(16), 4869–4880. <https://doi.org/10.1080/00207543.2021.1906460>
- Mantriota, G. (2007). Theoretical model of the grasp with vacuum gripper. *Mechanism and Machine Theory*, 42(1), 2–17. <https://doi.org/10.1016/j.mechmachtheory.2006.03.003>
- Modarress, B., Ansari, A., & Lockwood, D. L. (2005). Kaizen costing for lean manufacturing: A case study. *International Journal of Production Research*, 43(9), 1751–1760. <https://doi.org/10.1080/00207540500034174>
- Mousa, M. A. A., Elgohr, A. T., & Khater, H. A. (2024). Whale-Based Trajectory Optimization Algorithm for 6 DOF Robotic Arm. *Annals of Emerging Technologies in Computing*, 8(4), 99–114. <https://doi.org/10.33166/AETiC.2024.04.005>
- Perera, K. B. A. D. C. T. D., Makavita, C. D., & Dias, M. U. B. (2021). *Optimising Robot Manipulator Inverse Kinematics with Genetic Algorithm*.
- Qin, L., Ying-qi, J., Yu-feng, H., Gang, L., & Chuang, Y. (2022). A multi-objective trajectory optimization algorithm for industrial robot. *Chinese Journal of Engineering Design*, 29(2), 187–195. <https://doi.org/10.3785/j.issn.1006-754X.2022.00.011>
- Rady, M. (2024). Application of evolutionary algorithms to design economic flat slab buildings based on the intended function. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-58763-8>
- Ruswan Suwarman, H. (2021). Evaluasi Penerapan Evolutionary Algorithm Untuk Pemecahan Traveling Salesman Problem. *Sistemik (Jurnal Ilmiah Nasional Bidang Ilmu Teknik)*, 9(1), 28–34.
- Sánchez-Sosa, R. A., & Chavero-Navarrete, E. (2025). Evolutionary Algorithms for the Optimal Design of Robotic Cells: A Dual Approximation for Space and Time. *Applied Sciences (Switzerland)*, 15(15). <https://doi.org/10.3390/app15158455>
- Shi, L., Jiang, Y., & Jing, Z. (2011). A kinematic simulation environment for a 4-DOF SCARA robot. *Advanced Materials Research*, 338, 766–769. <https://doi.org/10.4028/www.scientific.net/AMR.338.766>
- Taifa, I. W. R., & Vhora, T. N. (2019). Cycle time reduction for productivity improvement in the manufacturing industry. *JIEMS Journal of Industrial Engineering and Management Studies*, 6(2), 147–164. <https://doi.org/10.22116/JIEMS.2019.93495>
- Thrun, S., & Leonard, J. J. (2008). *Simultaneous 37. Simultaneous Localization and Mapping*. https://doi.org/10.1007/978-3-540-30301-5_38
- Wei, B., Liu, C., Zhang, X., Zheng, K., Cao, Z., & Chen, Z. (2024). Smooth and Time-Optimal Trajectory Planning for Robots Using Improved Carnivorous Plant Algorithm. *Machines*, 12(11). <https://doi.org/10.3390/machines12110802>
- Xu, Y., Kong, S., Kong, X., & Ying, Z. (2025). Multi-objective trajectory optimization method for industrial robots based on improved TD3 algorithm. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-25949-7>
- Yaping Lu. (2023). Retracted: Improved Genetic Algorithm for Intelligent Grinding Trajectory of Industrial Robot Sensor. In *Journal of Sensors* (Vol. 2023). Hindawi Limited. <https://doi.org/10.1155/2023/9820943>

- Yulianto, S., Gede, I., Widyadana, A., & Sepadyati, N. (2020). *Optimasi Rute Pengiriman Menggunakan* (Vol. 10, Number 1).
- Zanchettin, A. M., Messeri, C., Cristantielli, D., & Rocco, P. (2022). Trajectory optimisation in collaborative robotics based on simulations and genetic algorithms. *International Journal of Intelligent Robotics and Applications*, 6(4), 707–723. <https://doi.org/10.1007/s41315-022-00240-4>
- Zhang, H., Xia, Q., Sun, J., & Zhao, Q. (2022). A Fully Geometric Approach for Inverse Kinematics of a Six-Degree-of-Freedom Robot Arm. *Journal of Physics: Conference Series*, 2338(1). <https://doi.org/10.1088/1742-6596/2338/1/012089>

B-Interface Point Dobot E6

RobotStudio Pro 4.3

Magician EG Settings Log Dobot Application Help document

LogicTA Open Save

Programming Points

Mode: Joint Linear Run to

Delete Cover + Add point

NO.	Point	Alias	User	Tool	X	Y	Z	Rx	Ry	Rz
1	P1	StartPoint	0	0	0.0000	-120.8000	615.2000	-90.0000	0.0000	180.0000
2	P2	Object1Before	0	0	277.2279	-101.7401	216.6344	-80.0571	3.8775	178.9808
3	P3	Object1Take	0	0	214.2855	-163.1389	233.1245	-89.9793	4.1415	-178.0519
4	P4	Object1Up	0	0	274.3674	-105.5731	257.6988	-88.0353	4.1445	-178.0519
5	P5	BeforeDrop	0	0	-0.0000	-255.1383	139.3196	-89.7598	0.0229	-179.9999
6	P6	DropPoint	0	0	0.7061	-419.4481	158.2059	-87.9909	3.8630	-179.7943
7	P7	Object2Before	0	0	341.9578	-122.8613	235.1091	-91.3770	3.6178	-178.1333
8	P8	Object2Take	0	0	345.7039	-161.7770	226.7704	-88.2817	3.1824	-178.7084
9	P9	Object2Up	0	0	346.7334	-160.4290	281.3199	-87.7441	4.1372	-179.1404
10	P10	Object3Before	0	0	273.3241	45.9887	189.9162	-91.2266	4.0558	179.5231
11	P11	Object3Take	0	0	277.3308	-89.6680	185.0879	-91.2157	3.1871	179.7544
12	P12	Object3Up	0	0	274.5145	-106.5777	277.3489	-94.5792	3.1092	178.7883

Jog Monitor Speed(100%) Safety posture

User coord: 0 Tool coord: 0

Joint Coordinate system

Continuous jog Step: 0.01 0.1 0.5 1

J1- -75.4688° J1+
J2- -14.5365° J2+
J3- -9.5720° J3+
J4- -102.8296° J4+
J5- -76.5475° J5+
J6- 183.3256° J6+

29PC Carah Search 14:16 02/06/2026

RobotStudio Pro 4.3

Magician EG Settings Log Dobot Application Help document

LogicTA Utama Open Save

Programming Points

Mode: Joint Linear Run to

Delete Cover + Add point

9	P9	Object2Up	0	0	346.7334	-160.4290	281.3199	-87.7441	4.1372	-179.1404
10	P10	Object3Before	0	0	273.3241	45.9887	189.9162	-91.2266	4.0558	179.5231
11	P11	Object3Take	0	0	277.3308	-89.6680	185.0879	-91.2157	3.1871	179.7544
12	P12	Object3Up	0	0	274.5145	-106.5777	277.3489	-94.5792	3.1092	178.7883
13	P13	0	0	0	273.8032	-94.9376	303.2099	-93.0028	0.5164	179.4026
14	P14	Object4Before	0	0	347.8282	36.1891	199.0360	-93.5981	3.3118	179.4903
15	P15	Object4Take	0	0	348.1330	-89.0673	186.9648	-92.3328	3.4908	179.0022
16	P16	Object4Up	0	0	340.8671	94.9954	238.8051	-92.0242	1.7538	-179.1149
17	P17	Object4Move	0	0	235.4115	-114.7387	410.5243	-99.0897	45.6940	177.0928
18	P18	2BeforeDrop	0	0	-0.0000	-255.1383	288.9863	-89.7598	0.0229	-179.9999
19	P19	Object5Take	0	0	273.9565	-12.7207	143.1507	-91.0802	4.4039	-178.4867
20	P20	Object5Before	0	0	268.9058	32.2191	150.4590	-92.7217	6.5551	-179.6401
21	P21	Object5Up	0	0	272.7028	-13.1760	195.8047	-89.2235	4.1787	-179.1538

Jog Monitor Speed(100%) Safety posture

User coord: 0 Tool coord: 0

Joint Coordinate system

Continuous jog Step: 0.01 0.1 0.5 1

J1- -0.7294° J1+
J2- -2.1187° J2+
J3- -2.6179° J3+
J4- -9.9167° J4+
J5- -9.8286° J5+
J6- 10.9059° J6+

Sangat lembap Sekarang Search 15:00 02/06/2026

B-Interface Point Dobot MG400

DobotStudio Pro 2.8.1 x86_64 stable.202402021647

LOGICTAAA

MG400 192.168.1.6 Connected

Safe LI Online

Global Speed(72%)

File Save Undo Recovery Debug

Points RunTo mode MovJ Import Export

Control

Events

Control

Operators

String

Custom

Move

Posture

IO

Modbus

TCP

Name	Alias	X	Y	Z	R	User	Tool
InitialPose		350	0	0	0	0	0
P2		7.2655	267.43	166.63	160	0	0
P3	StartPoint	-0.658	257.05	107.96	160	0	0
P4	TakeObject	-21.2	424.47	51.33	160	0	0
P5	Object5	256.66	166.00	39.612	160	0	0
P6	BeforeTak...	-19.22	370.71	77.801	160	0	0
P7	Object6	329.08	164.62	39.612	160	0	0
P9	Object3	256.80	92.152	39.612	160	0	0
P10	Object4	325.23	89.379	39.612	160	0	0
P11	Phase1Done	210.42	244.89	106.74	160	0	0
P13	Object5Take	257.27	166.31	33.600	160	0	0

Add

User Frame 0 Tool Frame 0

Mode Jog Step

Step Value 0.1 1 5 10

X -0.66 Y 257.06 Z 107.97 R 160.00

J1 90.15 J2 -8.97 J3 3.64 J4 69.85

13:18 10/06/2024