

PROPOSAL SKRIPSI

**SIMULASI PERENCANAAN LONG HORIZON MANIPULATION TASK
FRANKA KITCHEN MENGGUNAKAN FLOWPOLICY**



**AHMAD DAFFA ABIYYU
NIM. 163221055**

**PROGRAM SARJANA
TEKNIK ROBOTIKA DAN KECERDASAN BUATAN
DEPARTEMEN TEKNIK
FAKULTAS TEKNOLOGI MAJU DAN MULTIDISIPLIN
UNIVERSITAS AIRLANGGA
2026**

LEMBAR PENGESAHAN

SIMULASI PERENCANAAN LONG HORIZON MANIPULATION TASK FRANKA KITCHEN MENGGUNAKAN FLOWPOLICY

Nama : Ahmad Daffa Abiyyu
NIM : 163221055
Tanggal Seminar Proposal :

Surabaya,

Pembimbing I

Pembimbing II

Yutika Amelia Effendi, S.Kom.,
M.Kom., Ph.D
NIP. 199404142018083201

Ir. Asif Ali Zamzami, S.ST., M.Sc.,
Ph.D., IPM
NIP. 199207222022103101

Mengetahui,
Koordinator Program Studi
S1 Teknik Robotika dan Kecerdasan Buatan

Ir. Asif Ali Zamzami, S.ST., M.Sc., Ph.D., IPM
NIP. 199207222022103101

PERNYATAAN ORISINALITAS SKRIPSI

Saya, Ahmad Daffa Abiyyu, NIM. 163221055, penulis Skripsi yang berjudul
**SIMULASI PERENCANAAN LONG HORIZON MANIPULATION TASK
FRANKA KITCHEN MENGGUNAKAN FLOWPOLICY**

menyatakan bahwa:

1. Skripsi ini adalah asli dan benar-benar hasil karya saya sendiri, bukan hasil karya pihak lain dengan mengatasnamakan saya, bukan merupakan hasil tiruan atau jiplakan (plagiarism) dari karya pihak lain, dan/atau bukan tulisan yang dibuat dengan bantuan kecerdasan buatan.
2. Skripsi ini belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Airlangga, maupun di perguruan tinggi lainnya.
3. Dalam Skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dengan disebutkan nama pengarang dan dicantumkan dalam daftar kepustakaan.

Pernyataan ini saya buat dengan sebenar-benarnya, dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis Skripsi ini, serta sanksi-sanksi lainnya sesuai dengan norma dan peraturan yang berlaku di Universitas Airlangga.

Surabaya, xx Januari 20xx

Tanda tangan di atas materai

Ahmad Daffa Abiyyu
NIM. 163221055

KETENTUAN PENGGUNAAN SKRIPSI

Ketentuan hak cipta bagi skripsi yang tidak dipublikasikan, terdaftar, tersedia, serta terbuka untuk umum di Perpustakaan Universitas Airlangga, dimiliki penulis dengan mengikuti aturan HKI yang berlaku di Universitas Airlangga. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan dengan seizin penulis dan harus disitasi sesuai dengan kaidah ilmiah. Memperbanyak atau menerbitkan sebagian atau seluruh skripsi haruslah seizin Penulis.

Sitasi Skripsi ini dapat ditulis sebagai berikut:

Abiyyu, A.D (2026). Simulasi Perencanaan Long Horizon Manipualtion Task Franka Kitchen Menggunakan FlowPolicy. Skripsi. Universitas Airlangga.

Abiyyu, A.D (2026). Simulation of Long-Horizon Manipulation Task Planning in the Franka Kitchen Using FlowPolicy. Undergraduate Thesis. Skripsi. Universitas Airlangga.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang melimpahkan rahmat dan tuntutanNya sehingga skripsi dengan judul “Simulasi Perencanaan Long Horizon Manipulation Task Franka Kitchen Menggunakan FlowPolicy” dapat terselesaikan. Tidak lupa pula ucapan terima kasih dan penghargaan setinggi-tingginya penulis sampaikan kepada:

1. Kedua orang tua penulis yang selalu mendoakan dan memberikan dukungan kepada penulis.
2. Yutika Amelia Effendi, S.Kom., M.Kom., Ph.D sebagai dosen pembimbing I yang senantiasa memberi bimbingan dan arahan dalam penulisan skripsi.
3. Ir. Asif Ali Zamzami, S.ST., M.Sc., Ph.D., IPM sebagai dosen pembimbing II yang senantiasa memberi bimbingan dan arahan dalam penulisan skripsi.
4. Pasangan saya Yulan Fadilla Savitri dengan segala kebajikan dan kesabarannya sehingga penulis dapat menyelesaikan laporan skripsi
5. Teman-teman program studi Teknik Robotika dan Kecerdasan Buatan Angkatan 2022 yang telah memberikan banyak apresiasi, bantuan, dan dukungan dalam pengerjaan skripsi ini.
6. Semua pihak yang telah memberi bantuan dan dorongan agar penyusunan skripsi ini dapat diselesaikan.

Penulis sadar bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis menerima kritik dan saran yang bersifat membangun.

Surabaya, xx Januari 20xx

Tanda tangan di atas materai

Ahmad Daffa Abiyyu

NIM. 163221055

DAFTAR ISI

HALAMAN COVER.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS SKRIPSI.....	iii
KETENTUAN PENGGUNAAN SKRIPSI.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Pengertian Imitation Learning.....	7
2.1.1 Perkembangan Penelitian Imitation Learning.....	7
2.2 Trajectory Planning	14
2.3 Rancangan FlowPolicy	15
2.3.1 Flow Matching	16
2.3.2 Consistency Model	18

2.3.3	Consistency Flow Matching	20
2.4	Pengertian Robot Manipulator	23
2.5	Deskripsi Lingkungan Franka Kitchen.....	24
2.5.1	Franka Emika Panda Robot	24
2.5.2	Konfigurasi Gerak Robot Franka Emika Panda Pada Franka Kitchen	26
2.5.3	Konfigurasi Franka Kitchen.....	27
BAB III	METODE PENELITIAN	28
3.1	Lokasi dan Waktu Penelitian	28
3.2	Bahan dan Alat	28
3.3	Variabel Penelitian	29
3.4	Prosedur Penelitian	29
3.4.1	Jadwal Penelitian	32
3.4.2	Preprocessing Dataset Lingkungan Franka Kitchen	33
3.4.3	Pelatihan dan Pengujian FlowPolicy pada Franka Kitchen	34
3.4.4	Hyperparameter Fine Tuning FlowPolicy	39
3.5	Evaluasi	40
3.5.1	Success Rate	41
3.5.2	Average Completion Length.....	42
3.5.3	Inference Latemcy	42
3.5.4	Failure Mode Analysis.....	43
DAFTAR PUSTAKA.....		45
LAMPIRAN.....		47

DAFTAR TABEL

Tabel 2.1.1 Perkembangan Penelitian Imitation Learning	10
Tabel 2.5.1 Denavit-Hartenberg Robot Franka Emika Panda.....	25
Tabel 2.5.2 Konfigurasi Lingkungan Franka Kitchen dengan Aksi Robot Franka Emika Panda.....	26
Tabel 2.5.3 Ukuran Benda dan Parameter Keberhasilan di Lingkungan Franka Emika Kitchen.....	27
Tabel 3.4.1 Jadwal Penelitian.....	32
Tabel 3.4.2 Hyperparameter FlowPolicy.....	34
Tabel 3.4.3 Ruang Lingkup Hyperparameter FlowPolicy.....	39

DAFTAR GAMBAR

Gambar 2.3.1 Perbandingan Score Matching dan Flow Matching dengan untuk Konversi Distribusi Gaussian ke Distribusi Target.....	17
Gambar 2.3.2 Proses Pelatihan Consistency Model	19
Gambar 2.3.3 Proses Pelatihan Multi Segment Consistency Flow Matching.....	21
Gambar 2.5.1 Konfigurasi Sendi dan Kerangka Koordinat Franka Emika Panda	25
Gambar 3.4.1 Skema Prosedur Penelitian	30
Gambar 3.4.2 Tahap Preprocessing Dataset Franka Kitchen	33
Gambar 3.4.3 Pelatihan FlowPolicy pada Lingkungan Franka Kitchen	36
Gambar 3.4.4 Pengujian FlowPolicy pada Lingkungan Franka Kitchen	38

DAFTAR LAMPIRAN

Lampiran 1 Lingkungan Franka Kitchen	46
---	-----------

BAB I

PENDAHULUAN

1.1 Latar Belakang

Long horizon task dalam domain robotika, khususnya pada robot manipulator, didefinisikan sebagai tugas yang memerlukan penyelesaian sub-tugas secara berurutan dalam rentang waktu yang panjang. Tantangan utama dalam tugas *long horizon* ini adalah menjaga konsistensi aksi yang tepat dari awal hingga akhir tanpa mengakumulasi kesalahan. Tidak hanya itu, terkadang sub-tugas yang dilakukan secara berurutan dilaksanakan dalam lingkungan dengan benda-benda yang sensitif disekitarnya, maka toleransi error dalam robot manipulator harus sangat kecil (Ren et al., 2022). Guna mengatasi hal tersebut, metode *imitation learning* hadir sebagai pendekatan yang memungkinkan robot mempelajari kebijakan kontrol secara langsung dengan meniru pola gerakan dari demonstrasi ahli (Chenu et al., 2022). Penerapan *imitation learning* pada *long horizon task* memungkinkan robot melaksanakan tugas dengan kesalahan yang lebih terkontrol berbeda dengan metode pembelajaran lain yang mengandalkan eksplorasi mandiri. Berdasarkan hal tersebut, *imitation learning* sering digunakan pada *long horizon task* terutama pada lingkungan yang memiliki benda sensitif disekitarnya. Penerapan *imitation learning* banyak ditemukan pada pabrik-pabrik manufaktur yang memerlukan tingkat presisi tinggi pada saat merakit komponen dan pabrik lain yang membutuhkan kualitas produk yang terkontrol (Chu et al., 2025; Wang et al., 2025).

Penerapan *imitation learning* dalam pabrik-pabrik terletak pada kebutuhannya untuk menyelesaikan rangkaian sub-tugas yang saling bergantung dan memiliki kompleksitas lintasan yang bervariasi. Salah satu lingkungan *benchmark* yang sering digunakan untuk merepresentasikan permasalahan tersebut adalah Franka Kitchen (Gupta et al., 2019). Lingkungan Franka Kitchen mensimulasikan robot manipulator Franka Emika Panda untuk melakukan empat aktivitas dapur yaitu membuka pintu kabinet, memutar knob kompor, membuka microwave, dan menaruh teko di kompor. Lingkungan ini menuntut robot untuk mengeksekusi perencanaan trayektori yang presisi dalam jangka waktu panjang dan

menjaga konsistensi antar sub-tugas agar kegagalan pada satu tahap tidak merusak keseluruhan rangkaian tugas. Berdasarkan hal tersebut, lingkungan simulasi franka kitchen dapat dimanfaatkan sebagai *benchmark* untuk mengevaluasi performa model *imitation learning* secara terkontrol sebelum diterapkan pada lingkungan lain yang lebih kompleks atau pada sistem robot nyata.

Model *imitation learning* pada lingkungan franka kitchen dievaluasi berdasarkan kemampuan model dalam menyelesaikan empat sub-tugas secara sekuensial. Hasil evaluasi menunjukkan bahwa model konvensional seperti LSTM-GMM mampu menyelesaikan dua sub-tugas secara sekuensial dengan tingkat keberhasilan sebesar 90%, namun performa model menurun menjadi 34% saat mengerjakan sub-tugas ketiga dan keempat. Pola penurunan serupa juga terlihat pada model *Implicit Behavior Cloning* (IBC) dan *Behavior Transformer* (BeT) yang menunjukkan tingkat keberhasilan tinggi pada dua sub-tugas dan menurun ketika melakukan sub-tugas ketiga dan keempat. Berdasarkan data tersebut, dapat disimpulkan bahwa model *imitation learning* sebelumnya mengalami kesulitan dalam mempertahankan performa yang stabil setelah menyelesaikan lebih dari dua sub-tugas dalam lingkungan franka kitchen (Chi et al., 2025).

Keterbatasan pada model *imitation learning* sebelumnya mendorong peneliti untuk mengeksplorasi pendekatan model *imitation learning* yang berbasis *generative artificial intelligence* terbaru yaitu *diffusion* bernama *diffusion policy*. Menurut penelitian Chi, model ini mencapai tingkat keberhasilan 100% dalam empat sub-tugas yang dilakukan secara sekuensial (Chi et al., 2025). Namun demikian *diffusion policy* secara inheren menangkap kekurangan model *diffusion* yaitu arsitekturnya yang tidak efisien memori (Avrahami et al., 2023). Dalam lingkungan *long horizon task* khususnya franka kitchen, kekurangan ini menyebabkan akumulasi latensi pada setiap langkah aksi sehingga robot menjadi kurang responsif dan tidak mampu menyesuaikan gerakan dengan perubahan kondisi secara *real-time* apabila suatu trajektori terdapat kesalahan. Beberapa pendekatan telah diusulkan guna mengatasi kekurangan tersebut, seperti distilasi model untuk mereduksi kompleksitas jaringan. Salah satunya adalah *Consistency policy* yang menggunakan *Consistency Trajectory Model* (CTM) untuk

mengeksekusi aksi hanya dalam satu langkah inferensi. Namun pendekatan ini mengorbankan stabilitas *training* dan biaya training yang lebih mahal (Wolf et al., 2025). Selain itu, terdapat penelitian yang menggunakan teknik *pruning* pada jaringan *diffusion*, tetapi hasil model *diffusion policy* turun 30% sehingga model ini tidak terlalu efektif pada lingkungan franka kitchen (Wu et al., 2025). Terlepas dari model *diffusion*, terdapat pendekatan lain yang mengusulkan untuk mengubah model *diffusion* menjadi *flow matching*.

Flow matching adalah teknik pemodelan generatif yang mampu memetakan suatu distribusi awal yang bersifat acak ke distribusi target tertentu melalui pemodelan aliran probabilitas, tanpa memerlukan proses *forward diffusion* secara berulang. Salah satu pendekatan *flow matching* terbaru yang diterapkan di bidang robotika adalah FlowPolicy. FlowPolicy adalah model *policy* untuk robot yang dibuat dari *consistency flow matching* dengan PointNet *encoder* untuk mengekstrak informasi yang ada di lingkungan. Namun pada implementasinya model FlowPolicy sejauh ini hanya diuji pada lingkungan *short horizon* (Q. Zhang et al., 2025). Belum adanya studi mengenai kinerja FlowPolicy pada skenario *long horizon* menyisakan pertanyaan besar mengenai apakah efektivitas model ini dapat dipertahankan ketika dihadapkan pada ketergantungan tugas sekuensial yang panjang. Berdasarkan hal tersebut, penelitian ini dilakukan untuk mengevaluasi FlowPolicy pada lingkungan franka kitchen guna menguji kapabilitas model berbasis *state* tersebut dalam menyelesaikan lingkungan yang memerlukan beberapa sub-tugas berurutan.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang, penulis mengangkat beberapa permasalahan, sebagai berikut:

1. Bagaimana algoritma FlowPolicy yang dimodifikasi menjadi model berbasis *state* dapat diimplementasikan untuk menyelesaikan tugas manipulasi di lingkungan simulasi franka kitchen?
2. Bagaimana *success rate* model FlowPolicy berbasis *state* dalam menyelesaikan skenario *long horizon task* pada lingkungan franka kitchen?

3. Bagaimana *trade-off* internal antara *inference latency* dan *success rate* pada implementasi FlowPolicy tersebut ketika dilakukan variasi *hyperparameter*?

1.3 Tujuan Penelitian

Berikut ini adalah tujuan dari penelitian yang diangkat oleh penulis.

1. Mengimplementasikan dan mensimulasikan algoritma FlowPolicy yang telah dimodifikasi menjadi model berbasis *state* untuk menyelesaikan tugas manipulasi di lingkungan simulasi *franka kitchen*.
2. Mengevaluasi *success rate* dari model FlowPolicy berbasis *state* tersebut dalam menyelesaikan skenario *long horizon task* di lingkungan *franka kitchen*.
3. Menganalisis *trade-off* internal *success rate* dan *inference latency* dari penerapan FlowPolicy berdasarkan penyesuaian konfigurasi *hyperparameter* pada lingkungan *franka kitchen*.

1.4 Manfaat Penelitian

Penelitian yang dihasilkan oleh penulis diharapkan dapat memberikan beberapa kontribusi akademik, antara lain sebagai berikut:

1. Penelitian ini memberikan wawasan mendalam mengenai efektivitas serta internal *trade-off* antara *inference latency* dan *success rate* pada penerapan algoritma FlowPolicy untuk *long horizon manipulation task*.
2. Penelitian ini menjadi rujukan literatur bagi penelitian selanjutnya mengenai modifikasi dan evaluasi model *imitation learning* berbasis *state* pada lingkungan simulasi *franka kitchen*.
3. Penelitian ini membuka peluang pengembangan dan eksplorasi lebih lanjut terkait pemanfaatan model *flow matching* dalam sistem kontrol robot manipulator untuk tugas sekuensial yang panjang.

1.5 Batasan Masalah

Batasan masalah yang diangkat penulis pada penelitian ini antara lain sebagai berikut:

1. Penelitian ini berfokus pada *feasibility* algoritma FlowPolicy dalam lingkungan

simulasi franka kitchen dan tidak diimplementasikan ke lingkungan nyata.

2. Penelitian ini tidak menggunakan PointNet *encoder* FlowPolicy dikarenakan lingkungan franka kitchen adalah *state-based*.
3. Skenario *long horizon* dalam penelitian ini didefinisikan secara kuantitatif sebagai penyelesaian empat sub-tugas secara sekuensial yaitu membuka *microwave*, memutar *knob* kompor, menaruh teko di kompor, dan membuka cabinet.
4. Penelitian ini bersifat evaluasi internal terhadap FlowPolicy dan tidak melakukan komparasi secara langsung atau reimplementasi dengan model *diffusion policy* atau model *imitation learning* lainnya pada perangkat keras yang sama. Performa dari studi sebelumnya hanya digunakan sebagai referensi diskusi, bukan *baseline* komparasi absolut.
5. Robot yang digunakan adalah model simulasi franka robot yang telah disediakan dalam lingkungan simulasi franka kitchen.
6. Pelatihan dan pengujian model FlowPolicy dari penelitian ini dilakukan pada perangkat laptop dengan spesifikasi VRAM Nvidia GTX 1080 dengan memori 8 Gb.
7. Penelitian ini hanya menggunakan dataset kitchen-Complete-v2 yang berisi demonstrasi robot di lingkungan franka kitchen yang telah dibuat di *library gymnasium*.

1.6 Kontribusi Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, penelitian ini memberikan beberapa kontribusi ilmiah sebagai berikut:

1. Melakukan adaptasi pada kerangka kerja FlowPolicy asli dengan menghilangkan *visual encoder* sehingga model beroperasi murni secara *state-based* untuk menyesuaikan dengan karakteristik dataset Franka Kitchen.
2. Memberi evaluasi internal yang menguji dan menganalisis efektivitas FlowPolicy pada skenario manipulasi *long horizon* di lingkungan franka kitchen.
3. Menyediakan analisis sensitivitas *hyperparameter* terhadap stabilitas masa

pelatihan dan pengujian serta analisis pola kegagalan saat robot mengeksekusi tugas dengan dependensi yang panjang.

4. Memberikan analisis kuantitatif terkait *trade-off* antara *inference latency* dan *success rate* dari metode FlowPolicy yang diakibatkan oleh perubahan *hyperparameter* pada lingkungan franka kitchen.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Imitation Learning

Imitation learning merupakan pendekatan pembelajaran yang bertujuan untuk mengoptimalkan model agar dapat meniru perilaku hasil demonstrasi dan merepresentasikannya dalam bentuk aturan atau *policy*. Jika model dapat meniru demonstrasi dengan sempurna, maka hal ini memungkinkan otomatisasi terhadap pekerjaan yang dilakukan oleh demonstrator dalam suatu lingkungan (Oba & Ukita, 2023).

Metode dasar dalam *imitation learning* yang mengoptimalkan model dengan meminimalkan kesalahan prediksi aksi pada setiap langkah waktu secara berdasarkan data pelatihan yang telah dikumpulkan sebelumnya dikenal sebagai *behavior cloning*. (Moraes et al., 2024). Pada awal perkembangannya *behavior cloning* dapat memetakan *state* dengan aksi, tetapi model ini rentan terhadap akumulasi kesalahan karena proses optimasinya dilakukan pada tingkat aksi tunggal di setiap langkah waktu, bukan pada urutan aksi secara keseluruhan (Mehta et al., 2025). Keterbatasan tersebut mendorong perkembangan berbagai pendekatan lanjutan dalam *imitation learning* yang bertujuan meningkatkan stabilitas *policy* terhadap pergeseran distribusi seperti pada pembahasan subbab berikut.

2.1.1 Perkembangan Penelitian Imitation Learning

Pendekatan awal dalam *imitation learning* yaitu *behavior cloning* memformulasikan pemetaan dari observasi ke aksi sebagai masalah *supervised regression* (Chi et al., 2025). Meskipun pendekatan ini mampu mengerjakan tugas manipulasi sederhana dengan model yang dengan tingkat kompleksitas rendah. Pendekatan ini memiliki keterbatasan dimana model tidak dapat diterapkan pada lingkungan *testing* yang belum dipelajari.

Berbagai penelitian telah mengusulkan penggunaan arsitektur model yang lebih kompleks guna mengatasi keterbatasan tersebut seperti IBC yang memanfaatkan pendekatan *energy-based model* untuk mengevaluasi kualitas aksi

yang dihasilkan. Selain itu terdapat model berbasis jaringan saraf yang lebih kompleks seperti LSTM-GMM yang menggabungkan *Long Short-Term Memory* (LSTM) untuk memodelkan dependensi temporal dan *Gaussian Mixture Model* (GMM) untuk merepresentasikan variasi atau multimodalitas aksi. Pendekatan lain adalah BeT yang menggunakan arsitektur *transformer* dengan mekanisme *self-attention* yang mampu menangkap ketergantungan jangka panjang. Meskipun demikian, sebagian besar pendekatan *imitation learning* tersebut masih menunjukkan keterbatasan dalam menyelesaikan tugas manipulasi *long horizon tasks*. Model-model tersebut umumnya efektif pada skenario *short-horizon* namun kurang optimal dalam merepresentasikan distribusi aksi yang kompleks dan multimodal secara konsisten dalam rangkaian tugas berurutan sehingga performanya cenderung menurun pada lingkungan dengan dependensi temporal yang lebih panjang.

Guna mengatasi keterbatasan dalam merepresentasikan distribusi aksi yang kompleks tersebut, *diffusion policy* memperkenalkan penggunaan *Denoising Diffusion Probabilistic Models* (DDPM) untuk membuat suatu *policy* yang dapat merepresentasikan distribusi aksi yang kompleks. Pendekatan ini merepresentasikan *policy* sebagai proses *denoising* bertahap yang memungkinkan model untuk menangani banyak kemungkinan aksi secara lebih baik dan menghasilkan gerakan yang lebih stabil dibandingkan metode regresi konvensional (Chi et al., 2025).

Penelitian mengenai *diffusion policy* selanjutnya dikembangkan menjadi 3D *Diffusion Policy* (DP3) dengan mengintegrasikan persepsi tiga dimensi berbasis *point cloud* ke dalam kerangka kerja *diffusion*. Dengan memanfaatkan representasi 3D yang invarian terhadap perubahan sudut pandang, model ini terbukti meningkatkan kemampuan generalisasi robot secara signifikan dalam berbagai tugas manipulasi. Namun, tantangan utama yang masih tersisa dari keluarga model *diffusion* adalah latensi inferensi yang tinggi akibat proses *denoising* iteratif yang membutuhkan banyak langkah komputasi (Ze et al., 2024).

Berbagai upaya telah dilakukan untuk mempercepat proses inferensi ini. *consistency policy* mengusulkan metode *consistency distillation* untuk memangkas

proses *diffusion* menjadi beberapa langkah saja dan mempercepat inferensi. Sejalan dengan itu, LightDP berfokus pada efisiensi komputasi di perangkat *mobile* melalui kompresi jaringan dan pengurangan langkah sampling. Meskipun model-model ini berhasil meningkatkan kecepatan, seringkali terdapat *trade-off* yang besar antara kecepatan inferensi dengan kualitas generasi *policy* model. *Consistency policy* memiliki keterbatasan karena metode *consistency distillation* terkadang tidak stabil pada *epoch* awal, terutama pada lingkungan yang menuntut penyelesaian beberapa sub-tugas sehingga pelatihan pada *epoch* selanjutnya menghasilkan aksi yang tidak tepat dan menyimpang dari perilaku demonstrasi yang pada akhirnya menurunkan kinerja penyelesaian tugas secara keseluruhan. Selain itu, LightDP, meskipun mampu mempercepat proses inferensi, hanya mencapai tingkat keberhasilan sebesar 70% pada percobaan di lingkungan *franka kitchen*. (Prasad et al., 2024; Wu et al., 2025).

Sebagai pendekatan alternatif untuk mengeksplorasi hambatan tersebut, penelitian ini mengadopsi FlowPolicy sebuah metode baru yang pada arsitektur aslinya berbasis *consistency flow matching* dan *point cloud encoder*. Secara teoretis, berbeda dengan metode *diffusion* yang umumnya memerlukan banyak langkah inferensi atau proses distilasi yang rumit, FlowPolicy dirancang dengan tujuan agar model dapat menghasilkan aksi secara lebih efisien melalui metode *flow matching*. Oleh karena itu, metode ini menjadi alternatif yang menarik untuk dievaluasi kinerjanya. Tabel 2.1 merangkum beberapa perkembangan penelitian *imitation learning* yang relevan dengan penelitian ini.

Tabel 2.1.1 Perkembangan Penelitian *Imitation Learning*

No	Penelitian	Metode yang digunakan	Hasil Penelitian	Keunggulan	Kekurangan
1	Behavior Cloning (Zare et al., 2024a)	<i>Multi Layer Perceptron</i> (MLP) untuk memetakan <i>state</i> dengan aksi	Pendekatan simpel yang pertama diterapkan di robotika dan Efisien memori	Tingkat kompleksitas model yang rendah dan tidak menghabiskan banyak memori	Hasil <i>training</i> yang seringkali tidak dapat diterapkan pada lingkungan <i>testing</i>
2	IBC: Implicit Behavior Cloning (Florence et al., 2022)	<i>Energy-based model</i> untuk memetakan <i>state</i> ke aksi	Peningkatan model <i>behavior cloning</i> dengan <i>loss function</i> berupa berbasis <i>energy-based model</i> yang bisa diterapkan pada lingkungan <i>testing</i>	<i>Success rate</i> yang hampir sempurna pada dua sub-tugas awal franka kitchen	<i>Success rate</i> turun 30% pada sub-tugas ketiga dan turun 60% keempat

3	LSTM-GMM (Mandlekar et al., 2021)	<i>Long-Short Term Memory (LSTM)</i> dengan output <i>policy</i> berupa <i>Gaussian Mixture Model (GMM)</i>	Peningkatan model <i>implicit behavior cloning</i> pada franka kitchen sebesar 10%	<i>Success rate</i> yang hampir sempurna pada dua sub-tugas awal franka kitchen	<i>Success rate</i> turun 18% pada sub-tugas ketiga dan turun 54% keempat
4	BeT: Behavior Transformer (Shafiullah et al., 2022)	<i>Transformer model</i> yang digunakan untuk memetakan aksi dengan <i>conditioning</i> berupa observasi	Peningkatan dari model LSTM-GMM dan IBC dalam lingkungan <i>long horizon task</i> franka kitchen sebesar 3%	<i>Success rate</i> yang hampir sempurna pada dua sub-tugas awal franka kitchen	<i>Success rate</i> turun 23% pada sub-tugas ketiga dan turun 38% keempat
5	Diffusion Policy: Visuomotor policy learning via action diffusion (Chi et al., 2025)	DDPM dengan Unet <i>encoder</i> .	Peningkatan <i>success rate</i> dari model <i>imitation learning</i> konvensional.	<i>Success rate</i> 100% dalam empat sub-tugas yang dikerjakan pada lingkungan simulasi franka	<i>Inference latency</i> yang lambat

				kitchen.	
6	3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations (Ze et al., 2024)	DDPM dengan MLP point encoder.	Peningkatan <i>success rate</i> pada <i>short horizon manipulation task</i> yang berbasis <i>vision</i> dibanding <i>diffusion policy</i> .	Peningkatan <i>success rate</i> sebesar 20% pada <i>short horizon manipulation task</i> yang berbasis <i>vision</i> .	<i>Inference latency</i> yang lambat dan pengujian simulasi hanya diterapkan pada lingkungan yang <i>short horizon</i> .
7	Consistency Policy Accelerated Visuomotor Policies via Consistency Distillation (Prasad et al., 2024)	<i>Denoising Diffusion Probabilistic Models</i> didistilisasi menjadi MLP dengan consistency distillation dan digabung dengan Unet encoder.	Peningkatan <i>inference latency</i> pada dibanding <i>diffusion policy</i> .	Peningkatan <i>inference latency</i> sebesar 99% persen lebih cepat dari pada <i>diffusion policy</i>	Tidak stabil pada fase <i>training</i> terutama pada <i>epoch</i> awal
8	<i>On-Device Diffusion</i>	<i>Denoising Diffusion</i>	Peningkatan <i>inference</i>	Peningkatan	Rata-rata <i>success</i>

	Transformer Policy for Efficient Robot Manipulation (Wu et al., 2025)	<i>Probabilistic Models</i> yang dipruning dan digabung dengan Unet <i>encoder</i> .	<i>latency</i> pada dibanding <i>diffusion policy</i>	<i>inference latency</i> sebesar 98,9% persen lebih cepat dari pada <i>diffusion</i> <i>policy</i>	<i>rate</i> menurun menjadi 70%
--	--	---	--	--	------------------------------------

Berdasarkan ringkasan pada Tabel 2.1.1, penelitian di bidang *imitation learning* menunjukkan perkembangan dari pendekatan sederhana seperti *behavior cloning* hingga model yang lebih kompleks seperti *diffusion policy*. Pada tahap awal, *imitation learning* umumnya hanya mampu diterapkan pada lingkungan yang identik dengan data demonstrasi dan cenderung gagal ketika dihadapkan pada kondisi baru yang belum pernah dipelajari. Seiring dengan perkembangan arsitektur *neural network* yang semakin kompleks, keterbatasan tersebut dapat diatasi melalui penggunaan arsitektur yang dapat memetakan distribusi aksi secara lebih dinamis (Zare et al., 2024b).

Di sisi lain, kebutuhan industri manufaktur yang menuntut tingkat presisi tinggi dan batas toleransi kesalahan yang rendah pada proses otomatisasi produksi barang, mendorong penerapan *imitation learning* untuk menangani tugas dengan tahapan yang panjang dan aman (S. Zhang et al., 2024). Walaupun berbagai pendekatan terbaru telah meningkatkan kemampuan representasi dan kualitas prediksi aksi, masih terdapat tantangan berupa *trade-off* performa dengan kecepatan yang belum memuaskan dari model terbaru (Chi et al., 2025). Model dengan akurasi tinggi umumnya memiliki kompleksitas komputasi yang besar sehingga belum sepenuhnya memenuhi kebutuhan sistem produksi yang memerlukan respons cepat dan stabil secara *real-time*.

Dengan demikian, penelitian ini berupaya mengeksplorasi alternatif pemodelan menggunakan metode *flow matching* bernama FlowPolicy yang secara teoretis dirancang untuk lebih efisien secara komputasi dan memori. Melalui pendekatan ini, penelitian ini bertujuan untuk mengevaluasi kinerja model manipulasi robot pada lingkungan franka kitchen, khususnya dalam menganalisis *trade-off* antara *success rate* dan *inference latency*.

2.2 Trajectory Planning

Trajectory planning pada robot manipulator adalah proses penentuan lintasan dan profil gerak yang harus diikuti oleh *end-effector* atau setiap *joint* robot dari kondisi awal menuju kondisi tujuan secara optimal dan aman. Proses ini mempertimbangkan berbagai kendala, seperti batasan kinematika dan dinamika

robot, batas kecepatan dan percepatan joint, serta keberadaan rintangan di lingkungan kerja. Selain itu, *trajectory planning* juga bertujuan untuk mengoptimalkan kriteria tertentu, misalnya meminimalkan waktu tempuh, energi, atau kesalahan pelacakan, sehingga gerakan robot menjadi halus, efisien, dan stabil (Kramer & Bertram, 2022; Wen & Pagilla, 2023).

Secara umum, *trajectory planning* memiliki dua metode yaitu *position control* dan *velocity control*. Dalam konteks penelitian ini, dataset franka emika panda menyediakan data aksi-aksi dengan satuan kecepatan maka penelitian ini menggunakan *velocity control* dengan teknik optimasi FlowPolicy untuk menyelesaikan tugas franka kitchen.

Dalam konteks *velocity control*, fokus utama *trajectory planning* bergeser dari sekadar pencapaian posisi akhir menuju pengaturan kecepatan *joint* atau *end-effector* sepanjang lintasan yang direncanakan. Pendekatan berbasis *velocity control* menekankan perancangan profil kecepatan yang kontinu dan terbatas sehingga setiap *joint* bergerak dalam batas kecepatan maksimum yang diizinkan serta menghindari perubahan kecepatan yang mendadak. Hal ini penting untuk mengurangi getaran dan meningkatkan akurasi pelacakan lintasan. Pada level kontrol, *trajectory planning* menyediakan referensi kecepatan yang kemudian dilacak oleh pengendali kecepatan, sehingga sistem mampu merespons gangguan secara lebih cepat dibandingkan kontrol posisi murni (Pajak, 2024; Xiong et al., 2024).

2.3 Rancangan FlowPolicy

FlowPolicy merupakan metode pembelajaran *imitation learning* yang dirancang untuk menghasilkan kebijakan manipulasi robot secara cepat dan andal. Metode ini memanfaatkan pendekatan *consistency flow matching* yang dikombinasikan dengan representasi visual 3D berupa *point cloud* dan *state encoder* untuk mempelajari pemetaan dari observasi lingkungan ke aksi robot.

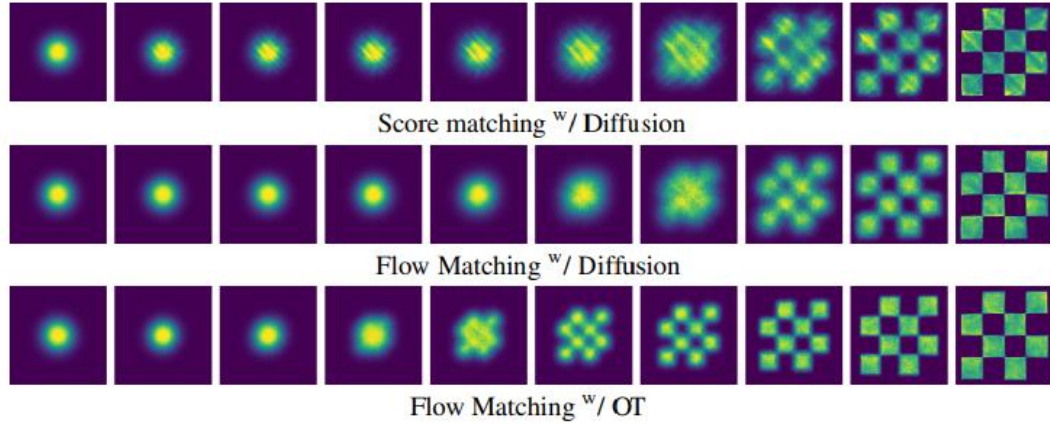
Berbeda dengan metode *policy learning* berbasis *diffusion* yang umumnya memerlukan banyak langkah inferensi, FlowPolicy mendefinisikan alur yang lebih ringkas dengan cara menghapus *forward process* serta memangkas proses *denoising*

iteratif, sehingga memungkinkan proses inferensi yang lebih cepat. Meskipun pendekatan ini secara teoretis meningkatkan efisiensi komputasi, peningkatan kecepatan inferensi tidak selalu berbanding lurus dengan kualitas kebijakan yang dihasilkan, terutama pada *long horizon task*. Pada skenario tugas sekuensial yang panjang, kompresi langkah inferensi berisiko mengorbankan stabilitas pergerakan. (Chi et al., 2025; Q. Zhang et al., 2025).

Model *flow matching* yang digunakan oleh FlowPolicy dirancang berdasarkan integrasi antara *flow matching* dan *consistency model* untuk menciptakan *consistency flow matching*. Metode ini digunakan untuk mencocokkan distribusi awal berupa *noise* dengan distribusi target dari trayektori demonstrasi. Pada arsitektur aslinya, FlowPolicy menggunakan *vision encoder* yang diadopsi dari PointNet++ untuk memproses data visual 3D berupa *point cloud*. Namun, dikarenakan penelitian ini berfokus pada pengujian di lingkungan *long horizon* *franka kitchen* yang menyediakan data observasi langsung berwujud *state*, seperti posisi dan kecepatan *joint robot*. Arsitektur asli FlowPolicy dimodifikasi secara spesifik menjadi model *state based* FlowPolicy. Modifikasi ini dilakukan dengan menghilangkan *vision encoder* PointNet++ dan menggantinya dengan *state encoder*. Pendekatan adaptasi dari *vision based* menjadi *state based* ini merupakan prosedur standar dalam pengujian algoritma *imitation learning* untuk mengevaluasi kemampuan inti model dalam memetakan dinamika aksi tanpa terbebani oleh kompleksitas komputasi pemrosesan visual. Pendekatan serupa juga telah divalidasi dan digunakan pada varian *state based* dari algoritma *diffusion policy* (Chi et al., 2025).

2.3.1 Flow Matching

Flow matching merupakan konsep dasar yang digunakan pada FlowPolicy dimana model generatif yang mampu memetakan suatu distribusi awal yang bersifat acak ke distribusi target tertentu melalui pemodelan aliran probabilitas, seperti ditunjukkan pada Gambar 2.3.1 berikut



Gambar 2.3.1 Perbandingan *Score Matching* dan *Flow Matching* dengan untuk Konversi Distribusi *Gaussian* ke Distribusi Target

Model ini merupakan pengembangan dari model *diffusion*, di mana proses pembelajaran tidak lagi bergantung pada penambahan *noise* secara berulang untuk membentuk distribusi acak dari data. Sebaliknya, *flow matching* secara langsung mendefinisikan dan mempelajari medan vektor yang menggambarkan transformasi dari distribusi acak menuju distribusi data. Dengan menghilangkan proses *forward diffusion* yang bersifat redundan, pendekatan ini mampu mempercepat proses inferensi sekaligus mempertahankan kualitas generasi data (Q. Zhang et al., 2025).

Flow matching menerima input medan vektor yang dapat menghasilkan distribusi probabilitas *ground truth* data dan medan vektor yang membuat distribusi normal. Flow matching mencoba mempelajari medan vektor *ground truth* dengan cara mengoperasikan medan vektor distribusi normal agar distribusi awal (p_0) sesuai dengan distribusi *ground truth* (p_1). Pengoperasian ini dilakukan menggunakan suatu *generator* berisi *artificial neural network* dan dapat di rumuskan sebagai berikut (Q. Zhang et al., 2025).

$$\begin{cases} \frac{d\xi_x(t)}{dt} = v_\theta(t, \xi_x(t)), \\ \xi_x(0) = x, \end{cases} \quad (1)$$

Persamaan (1) bertujuan untuk mempelajari medan vektor $v_\theta(t, \xi_x(t))$ sehingga medan vektor $\xi_x(t)$ mampu mentransformasikan distribusi acak awal secara kontinu menjadi distribusi target. Lalu mengoreksi hasil menggunakan

Persamaan (2) *loss function* berikut (Q. Zhang et al., 2025).

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, p_t} [\|v_\theta(t, x_t) - u(t, x_t)\|_2^2] \quad (2)$$

Namun dikarenakan medan vektor target yang ingin dicapai pada *loss function* tersebut tidak diketahui secara eksplisit, proses komputasi medan vektor secara keseluruhan menjadi kompleks dan memerlukan biaya komputasi yang tinggi. Oleh sebab itu, paper ini mengusulkan pendekatan yang lebih efisien yaitu dengan tidak menghitung medan vektor secara menyeluruh, melainkan hanya menggunakan satu sampel data sebagai variabel kondisional. Melalui pendekatan ini, pergerakan satu sampel tersebut secara implisit akan mendorong pergeseran seluruh distribusi probabilitas menuju distribusi target. Berdasarkan hal tersebut, *loss function* Persamaan (2) dapat ditulis ulang sebagai berikut (Q. Zhang et al., 2025).

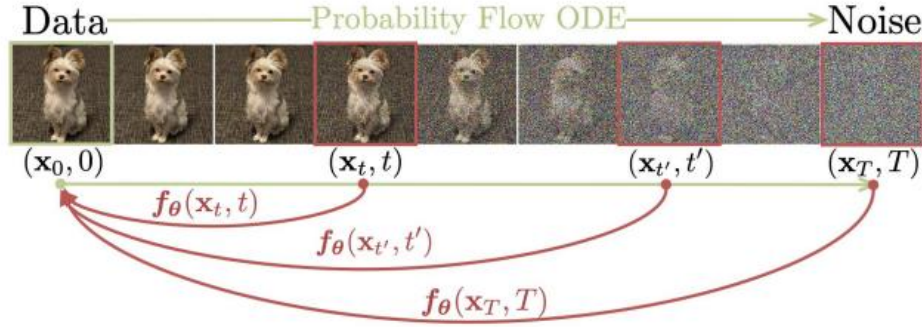
$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, p_t} [\|v_\theta(t, x_t) - u(t, x_t|x_1)\|_2^2] \quad (3)$$

Dalam ranah robotika, teknik *flow matching* ini dapat melampaui model *diffusion* terutama dalam hal efisiensi komputasi dan kualitas respons waktu nyata yang krusial bagi sistem otonom. Berbeda dengan model *diffusion* yang bergantung pada proses *denoising* iteratif yang panjang dan sering kali menghambat kecepatan pengambilan keputusan robot, *flow matching* menawarkan pendekatan yang memodelkan medan vektor deterministik untuk memetakan distribusi awal ke target melalui jalur yang lebih optimal secara matematis. Mekanisme ini memungkinkan penggunaan penyelesain persamaan diferensial biasa atau *ordinary differential equation* dengan langkah evaluasi yang jauh lebih sedikit, sehingga robot dapat menghasilkan aksi yang responsif dengan latensi minimal tanpa mengorbankan akurasi (Fang et al., 2025).

2.3.2 Consistency Model

Consistency model sendiri merupakan teknik optimasi *diffusion model*

dimana proses *denoising* atau *reverse diffusion process* tidak lagi dilakukan secara sekuensial seperti pada Gambar 2.3.2 berikut.



Gambar 2.3.2 Proses Pelatihan *Consistency Model*

Consistency model sendiri merupakan teknik optimasi *diffusion model* dimana proses *denoising* atau *reverse diffusion process* tidak lagi dilakukan secara sekuensial seperti pada Gambar 2.3.2 berikut. Setiap *timestep* dari proses *consistency model* dimana data merupakan kombinasi dari noise dan gambar asli akan langsung diprediksi dengan data ground truth sehingga model dapat memprediksi dari timestep manapun. (Song et al., 2023).

Penerapannya di ranah robotika visuomotor diwujudkan melalui metode yang disebut *consistency policy* yang dirancang untuk mengatasi latensi tinggi pada *diffusion policy* akibat proses inferensi iteratif yang panjang. Dalam kerangka kerja ini, *consistency policy* dilatih dengan cara mendistilasi *teacher policy* berbasis *diffusion* yang telah dilatih sebelumnya, di mana *student model* belajar untuk memetakan observasi langsung ke rangkaian aksi yang valid dalam satu langkah inferensi saja, atau melalui beberapa langkah (Prasad et al., 2024).

Pendekatan ini mengadaptasi arsitektur CTM dengan beberapa modifikasi kunci agar sesuai untuk kontrol robot, seperti penggunaan varians sampel awal yang dikurangi $\mathcal{N}(0, 1/T^2)$ untuk menjaga prediksi tetap berada dalam distribusi data demonstrasi ahli. Hasil eksperimen menunjukkan bahwa metode ini mampu mempercepat waktu inferensi dibandingkan metode *diffusion* standar seperti DDPM atau *Denoising Diffusion Implicit Model* (DDIM) pada perangkat keras dengan daya komputasi terbatas, tetapi dengan kekurangan fase *training* yang terkadang tidak stabil (Prasad et al., 2024).

Selain *consistency policy* yang berbasis pada *diffusion policy* dan CTM, *consistency model* juga bisa diterapkan pada *flow matching model*. Implementasi *consistency model* dengan *flow matching model* ini dinamakan sebagai *consistency flow matching*.

2.3.3 Consistency Flow Matching

Umumnya, *flow matching model* memprediksi medan vektor dalam setiap *timestep* sehingga medan vektor pada *timestep* pertama akan berbeda dengan *timestep* yang kedua. Cara memprediksi medan vektor ini tidak efisien apabila digunakan pada distribusi yang kompleks sehingga beberapa penelitian telah muncul untuk mengembangkan metode garis *flow* yang lurus dan optimal digunakan pada distribusi yang kompleks.

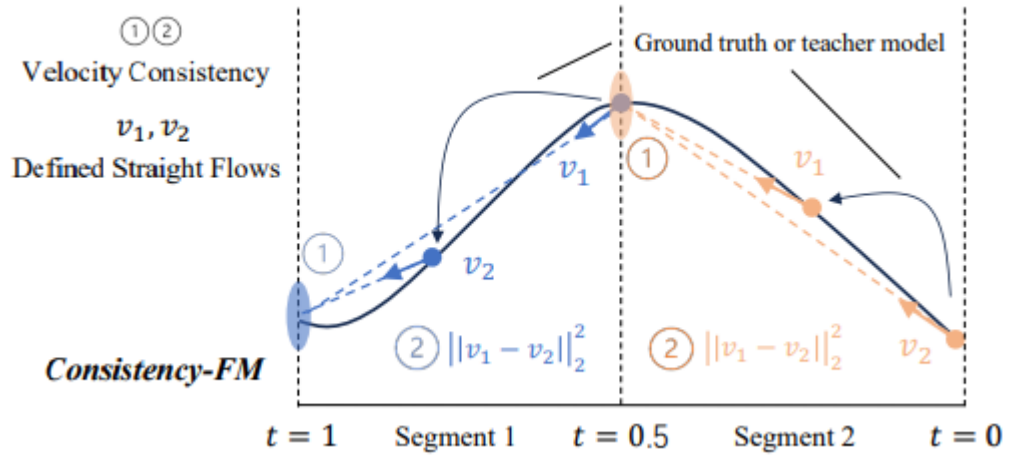
Salah satu metode yang digunakan untuk membuat garis *flow* yang lebih lurus ini adalah *consistency flow matching*. Metode *consistency flow matching* didasarkan pada Persamaan (1). Lalu, untuk mencocokkan medan vektor yang diprediksi dengan medan vektor data *ground truth*, perhitungan *loss function* digunakan dengan cara merumuskan ulang persamaan (3) menjadi berikut.

$$\begin{aligned}\mathcal{L}_\theta &= \mathbb{E}_{t \sim \mathcal{U}} \mathbb{E}_{x_t, x_{t+\Delta t}} \|f_\theta(t, x_t) - f_{\theta^-}(t + \Delta t, x_{t+\Delta t})\|_2^2 + \alpha \|v_\theta(t, x_t) \\ &\quad - v_{\theta^-}(t + \Delta t, x_{t+\Delta t})\|_2^2, \\ f_\theta(t, x_t) &= x_t + (1 - t) * v_\theta(t, x_t)\end{aligned}\tag{4}$$

Pada Persamaan (4), \mathcal{U} adalah *uniform distribution* pada setiap *timestep* $[0, 1 - \Delta t]$, α adalah bilangan skalar positif, Δt adalah *hyperparameter* interval waktu, θ melambangkan bobot model, dan θ^- adalah rata-rata berjalan dari nilai θ sebelumnya yang dihitung dengan *exponential moving average*. *Loss function* pada Persamaan (4) berkerja dengan menghitung perbedaan antara prediksi data asli pada waktu t dengan prediksi data asli pada waktu $t + \Delta t$, lalu menghitung perbedaan medan vektor pada waktu t dengan prediksi medan vektor pada waktu $t + \Delta t$. *Loss function* ini tidak secara langsung mencocokkan prediksi medan vektor dengan medan vektor *ground truth* melainkan mencocokkan prediksi aksi dengan aksi

ground truth dengan cara memasukkannya ke dalam model prediksi medan vektor lalu menegakkan konsistensi temporal antara prediksi pada waktu t dan $t + \Delta t$ agar *flow* yang dipelajari bersifat stabil dan berubah secara halus sepanjang waktu.

Dalam implementasinya pada distribusi probabilitas yang kompleks, *consistency flow matching* menyediakan pendekatan *multi-segment* yang membagi lintasan linear menjadi beberapa segmen dengan cara membagi interval waktu $[0,1]$ menjadi K segmen yang sama panjang. Metode ini mempelajari medan vektor $v_{\theta}^i(t, x_t)$ pada setiap segmen i yang di optimasi dari waktu i/K sampai $(i + 1)/K$.



Gambar 2.3.3 Proses Pelatihan *Multi Segment Consistency Flow Matching*

Gambar 2.3.3 merupakan visualisasi proses pelatihan *multi segment consistency flow matching* yang dimulai dengan membagi segmen dari waktu $[0,1]$ dengan rumus berikut.

$$[0,1] = \bigcup_{i=0}^{K-1} \left[\frac{i}{K}, \frac{i+1}{K} \right] \quad (5)$$

Perhitungan ini digunakan agar setiap segmen dalam fase *training* memiliki panjang segmen yang sama yaitu $1/K$. Lalu, untuk setiap segmen medan vektor akan diprediksi dengan Persamaan (4) dan dilatih menggunakan *loss function* Persamaan (6) berikut.

$$\begin{aligned}
\mathcal{L}_\theta &= \mathbb{E}_{t \sim \mathcal{U}^i} \lambda^i \mathbb{E}_{x_t, x_{t+\Delta t}} \left\| f_\theta^i(t, x_t) - f_\theta^i(t + \Delta t, x_{t+\Delta t}) \right\|_2^2 + \\
&\quad \alpha \left\| v_\theta^i(t, x_t) - v_\theta^i(t + \Delta t, x_{t+\Delta t}) \right\|_2^2, \\
f_\theta^i(x) &= x + \left(\frac{i+1}{K} - 1 \right) \cdot v_\theta^i(t, x)
\end{aligned} \tag{6}$$

Berdasarkan perhitungan Persamaan (6) diatas, \mathcal{U}^i adalah uniform distribution pada dari waktu i/K sampai $(i+1)/K$. Langkah pergerakan setiap segmen dikendalikan oleh bilangan positif yang sangat kecil Δt . x dan x_t bersama-sama mengikuti pola distribusi yang telah ditentukan. Alur dan medan vektor setiap segmen dideskripsikan oleh $v_\theta^i(t, x_t)$ yang bergantung pada parameter θ yaitu bobot pelatihan dan waktu t . Selain itu, λ^i adalah faktor pembobot positif yang digunakan untuk menyeimbangkan kepentingan dari berbagai bagian, terutama karena medan vektor di bagian tengah biasanya lebih sulit untuk dilatih.

Saat inferensi, model *multi segment consistency flow matching* bisa menghasilkan sampel data asli dengan cara mengambil input distribusi normal, lalu mengikuti hasil *training* dan mengevaluasi model agar distribusi data sama dengan distribusi data asli. Teknik iteratif sampling juga bisa di gunakan pada pendekatan *multi-segment* dengan metode euler dalam setiap segmen. Dengan demikian model dapat menghasilkan data yang mirip atau sama dengan data asli dengan waktu yang efisien. Rumus persamaan inferensi *multi segment consistency flow matching* dapat di formulasikan sebagai berikut.

$$x_{i/k} = x_{(i-1)/K} + \frac{1}{K} * v_\theta^i((i-1)/K, x_{(i-1)/K}), i = 1, 2, \dots, K-1 \tag{7}$$

$$x_{t+\Delta t} = x_t + \Delta t * v_\theta^i(t, x_t), t \in [i/K, (i+1)/K - \Delta t] \tag{8}$$

Persamaan (7) dan (8) mendeskripsikan proses inferensi pada *multi-segment consistency flow matching*, di mana transformasi dari keadaan awal yaitu noise menuju keadaan target dilakukan melalui lintasan yang dibagi menjadi beberapa segmen waktu. Persamaan (7) memodelkan pembaruan keadaan secara diskrit pada

batas antar segmen dengan menggeser medan vektor sebelumnya searah medan vektor v_{θ}^i yang dipelajari pada segmen tersebut dengan ukuran langkah waktu sebesar $1/K$. Sementara itu, Persamaan (8) menjelaskan dinamika kontinu di dalam satu segmen, di mana medan vektor berevolusi mengikuti integrasi euler dari medan vektor yang sama dengan langkah waktu kecil Δt . Dengan pendekatan ini, lintasan keseluruhan terbentuk sebagai rangkaian garis lurus lokal atau disebut sebagai *piecewise-linear* yang menjaga konsistensi vektor pada tiap segmen, sehingga memungkinkan proses inferensi yang efisien dan stabil tanpa memerlukan banyak langkah sampling seperti pada metode diffusion.

2.4 Pengertian Robot Manipulator

Robot manipulator merupakan sistem elektromekanis yang andal dan mampu memberikan solusi efektif untuk berbagai aplikasi industri terkini seperti pengambilan, penempatan, pengemasan, pengecatan, dan pengelasan. Secara khusus, robot manipulator dengan derajat kebebasan banyak digunakan pada seluruh tahapan proses produksi, termasuk sistem otomasi karena kemampuannya dalam melakukan operasi industri secara efektif tanpa memerlukan keterlibatan manusia.

Salah satu tantangan robot manipulator yang ada di industri adalah kapabilitas mereka dalam melakukan serangkaian tugas dengan tingkat keberhasilan yang tinggi dalam rentang waktu tertentu. Berdasarkan tantangan ini, banyak penelitian dan pendekatan telah dikembangkan untuk meningkatkan kemampuan robot dalam memahami lingkungan kerja, merencanakan trajektori gerakan, serta menyesuaikan perilaku mereka terhadap variasi kondisi operasional. Berbagai pendekatan tersebut mencakup metode berbasis pemodelan matematis, optimasi lintasan, hingga pemanfaatan teknik pembelajaran mesin seperti *imitation learning* dan *reinforcement learning*. Adanya pendekatan-pendekatan ini, diharapkan robot manipulator tidak hanya mampu bekerja secara presisi dan efisien, tetapi juga adaptif terhadap perubahan lingkungan dan kompleksitas tugas di dunia industri (Luo et al., 2025).

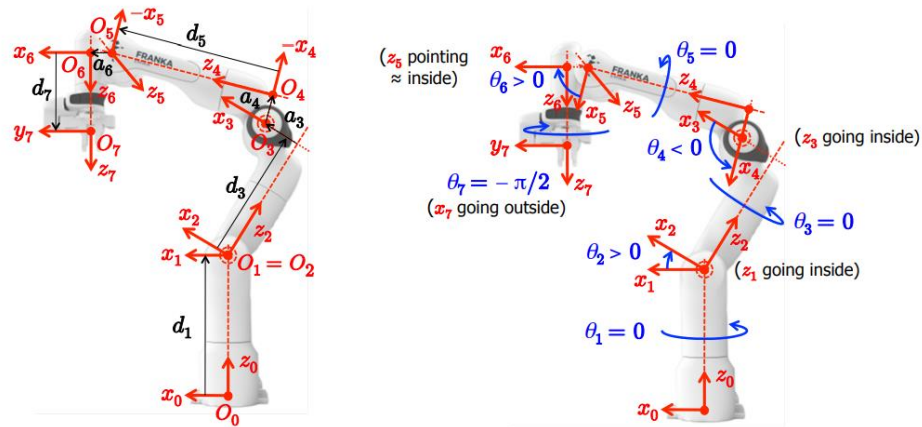
2.5 Deskripsi Lingkungan Franka Kitchen

Franka kitchen merupakan salah satu lingkungan *benchmark* paling populer untuk mengevaluasi kemampuan agen dalam menyelesaikan urutan manipulasi kompleks pada domain robotik nyata dan simulasi (Chi et al., 2025). Lingkungan ini pertama kali diperkenalkan dalam D4RL dan dirancang untuk memodelkan berbagai skenario manipulasi dapur yang melibatkan beberapa objek dengan tugas yang berbeda-beda, seperti membuka pintu kabinet, memutar knob kompor, membuka *microwave*, dan menaruh teko di kompor (Li et al., 2024).

Lingkungan ini menyediakan dataset yang berisi observasi dan aksi berisi kecepatan sudut yang kompatibel dengan franka emika panda yaitu robot manipulator 7-DoF yang dikembangkan dengan konfigurasi sistem gerak berbasis kinematika differensial dan mesin fisika MuJoCo. Selain itu, lingkungan ini juga menyediakan konfigurasi lingkungan yang berisi target tugas franka kitchen untuk benda seperti *microwave*, teko, kompor, dan lemari kabinet.

2.5.1 Franka Emika Panda Robot

Franka emika robot, khususnya franka emika panda, merupakan sebuah lengan robot dengan 7 derajat kebebasan (7-DOF) yang banyak digunakan sebagai platform standar dalam penelitian robotika modern. Robot ini dirancang dengan fokus pada interaksi manusia–robot yang aman, presisi tinggi, serta fleksibilitas kinematic (Haddadin, 2024). Konfigurasi setiap sendi pada robot Franka dapat dilihat pada Gambar 2.5.1.



Gambar 2.5.1 Konfigurasi Sendi dan Kerangka Koordinat Franka Emika Panda

Dari konfigurasi tersebut, dapat disusun tabel parameter Denavit-Hartenberg (DH) yang merepresentasikan hubungan geometrik antar sendi.

Tabel 2.5.1 Denavit-Hartenberg Robot Franka Emika Panda

i	α_i	a_i	d_i	θ_i
1	$\frac{\pi}{2}$	0	$d_1 > 0$	q_1
2	$-\frac{\pi}{2}$	0	0	q_2
3	$\frac{\pi}{2}$	$a_3 > 0$	$d_3 > 0$	q_3
4	$-\frac{\pi}{2}$	$a_4 < 0$	0	q_4
5	$\frac{\pi}{2}$	0	$d_5 > 0$	q_5
6	$\frac{\pi}{2}$	$a_6 > 0$	0	q_6
7	0	0	$d_7 > 0$	q_7

Tabel 2.5.1 DH diatas mendeskripsikan konfigurasi kinematika robot melalui empat parameter utama yaitu panjang link (a_i), sudut putar (i), offset link (d_i), dan sudut sendi (θ_i) untuk setiap sendi yang menjadi dasar dalam perhitungan kinematika maju maupun mundur robot (Haddadin, 2024). Dengan menggunakan

keempat parameter ini, posisi dan orientasi setiap *joint* dapat dinyatakan dalam bentuk matriks transformasi homogen yang memungkinkan perhitungan posisi *end-effector* secara sistematis maupun penentuan konfigurasi sendi yang diperlukan untuk mencapai pose tertentu. Konfigurasi kinematika tersebut kemudian dibangun di atas mesin fisika MuJoCo yang dikendalikan menggunakan kontrol berbasis kecepatan sendi dan linear.

2.5.2 Konfigurasi Gerak Robot Franka Emika Panda Pada Franka Kitchen

MuJoCo menggunakan pendekatan *generalized coordinates* untuk merepresentasikan sistem dan mengimplementasikan algoritma *recursive dynamics* seperti komputasi *forward dynamics* yang cepat dan stabil (Todorov et al., 2012). Pendekatan ini mampu memodelkan robot dengan struktur kinematika kompleks di lingkungan sekitarnya, termasuk efek gravitasi, gesekan, dan tumbukan.

Hubungan antara informasi kontrol robot dan keadaan lingkungan sekitarnya direpresentasikan secara langsung dalam vektor state yang terdapat pada data observasi. Seperti pada Tabel 2.5.2 berikut, vektor mencakup informasi konfigurasi robot seperti posisi sendi serta kondisi objek yang terlibat dalam tugas manipulasi.

Tabel 2.5.2 Konfigurasi Lingkungan Franka Kitchen dengan Aksi Robot Franka Emika Panda

Franka Kitchen	Konfigurasi
Dimensi Input Observasi	60 (Box(-1.0, 1.0, (9,), float32))
Dimensi Input Aksi	9 (Box(-1.0, 1.0, (9,), float32))
Sub-task	4 (list[str])

Dengan demikian, setiap aksi yang dihasilkan model dalam bentuk kecepatan sendi akan menyebabkan perubahan pada konfigurasi robot dan objek sehingga menghasilkan nilai observasi baru pada langkah waktu berikutnya. Perubahan tersebut direkam sebagai transisi *state* yang menjadi dasar proses pembelajaran model. Melalui rangkaian transisi ini, model mempelajari hubungan antara kondisi awal, aksi yang diberikan, dan kondisi akhir yang dihasilkan. Apabila pada suatu *state* tertentu robot berhasil memanipulasi objek sesuai dengan tujuan sub-tugas yang ditetapkan, maka tugas tersebut dinyatakan berhasil. Lingkungan ini

menyediakan mekanisme evaluasi berbasis penyelesaian sub-tugas yang memungkinkan pengukuran tingkat keberhasilan kebijakan secara terstruktur dalam rangkaian tugas manipulasi berurutan (Gupta et al., 2019).

2.5.3 Konfigurasi Franka Kitchen

Dalam setiap sub-tugas di lingkungan franka kitchen, terdapat berbagai objek berartikulasi dengan karakteristik dinamika yang berbeda-beda sehingga menuntut pemodelan manipulasi yang presisi. Setiap objek didefinisikan sebagai *body* dalam MuJoCo dengan satu atau lebih *joint* yang menentukan cara mereka dapat dimanipulasi oleh robot. Dalam konteks *long horizon task*, keberhasilan tugas diukur berdasarkan pencapaian konfigurasi target pada sendi-sendi objek ini, seperti apakah pintu *microwave* sudah terbuka pada sudut tertentu atau apakah teko sudah berada di koordinat yang diinginkan (Gupta et al., 2019).

Objek-objek dalam dapur dirancang untuk memberikan tantangan manipulasi yang beragam, mulai dari gerakan rotasi sederhana hingga interaksi dengan objek bebas. Berikut adalah Tabel 2.4.3 yang merinci dimensi dan target operasional untuk berbagai objek yang ada di dalam lingkungan franka kitchen gymnasium.

Tabel 2.5.3 Ukuran Benda dan Parameter Keberhasilan di Lingkungan Franka Emika Kitchen

Objek	Jenis Pergerakan	Tingkat keberhasilan	Estimasi Dimensi Visual (P x L x T)
Microwave	Hinge (Engsel)	Bukaan: 0,37 Radian	0,5m x 0,4m x 0,3m
Kettle (Teko)	Free (6-DoF)	Posisi: [-0,23, 0,75, 1,62]	Tinggi: 0,22m, Diameter: 0,16m
Slide Cabinet	Slide (Translasional)	Pergeseran: 0,37 Meter	0,4m x 0,02m x 0,6m (Pintu)
Stove Knobs	Hinge (Rotasi)	Aktivasi: ~[-0,9] Rad/m	Diameter: 0,05m, Tebal: 0,02m

Teko merupakan objek yang paling menarik secara teknis karena

menggunakan *free joint* yang memungkinkannya untuk dipindahkan secara bebas di ruang 3D. Di sisi lain, lemari dan laci menggunakan batasan linier atau rotasi yang membatasi ruang gerakan mereka, memberikan tantangan dalam hal arah tarikan oleh *griper* robot. Ketinggian meja dapur diatur secara konsisten pada koordinat z tertentu untuk memastikan bahwa robot dapat melakukan tugas-tugas manipulasi tanpa harus beroperasi di batas jangkauan kinematikanya (Gupta et al., 2019).

BAB III

METODE PENELITIAN

3.1 Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Teknik Robotika dan Kecerdasan Buatan, Gedung Nano, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga. Uji coba model robot ini akan dilakukan melalui simulasi menggunakan *library* gymnasium untuk menguji performa robot. Penelitian ini dilakukan selama 12 bulan, dimulai dari bulan Maret 2026 sampai dengan bulan Juni 2026.

3.2 Bahan dan Alat

Adapun bahan dan alat *software* yang dibutuhkan dalam pelaksanaan penelitian ini, meliputi

1. Franka Emika Robot dengan spesifikasi sebagai berikut:
 - a. Derajat kebebasan (DoF): 7-DoF lengan dan 2-DoF dari *gripper*.
 - b. Jangkauan Maksimum: 855 mm
 - c. Payload Maksimum: 3 kg
 - d. Berat Arm: 17.8 kg
 - e. Batas Kecepatan *Joint*:
 - J1 – J4: 150 °/s
 - J5 – J7: 180 °/s
 - J8 – J9 (*end effector*): 2 m/s
2. Laptop dengan kartu grafis Geforce GTX 1080 Ti video memori sebesar 8 Gb,

Processor i7, RAM 16 Gb, serta *Operating System* (OS) Linux Ubuntu 20.04.

3. Dataset demonstrasi dari Minari yang berisi *action space* dan *observation space* dalam 4209 *step* dan 19 episodes yang berisi tugas untuk menaruh pot, membuka kabinet, membuka mikrowave, dan menyalakan kompor.
4. Environment franka kitchen menggunakan *library* gymnasium yang menyediakan spesifikasi lingkungan franka kitchen beserta robot manipulator yang digunakan.

3.3 Variabel Penelitian

Berikut variabel yang digunakan dalam penelitian ini, antara lain:

1. Variabel bebas: Parameter FlowPolicy meliputi *seed*, *epoch*, *batch size*, *learning rate*, jumlah *hidden dimension*, *time embedding dimension*, *epsilon*, jumlah segmen, *delta t*, *alpha*, *observation horizon*, dan *action horizon*
2. Variabel terikat: Hasil pelatihan FlowPolicy seperti aksi yang dihasilkan, *state* yang diterima dan dihasilkan, *success rate* model, dan *inference latency*.
3. Variabel kontrol: Lingkungan simulasi franka kitchen dengan dataset demonstrasi yang digunakan dan jumlah percobaan yang dilakukan.

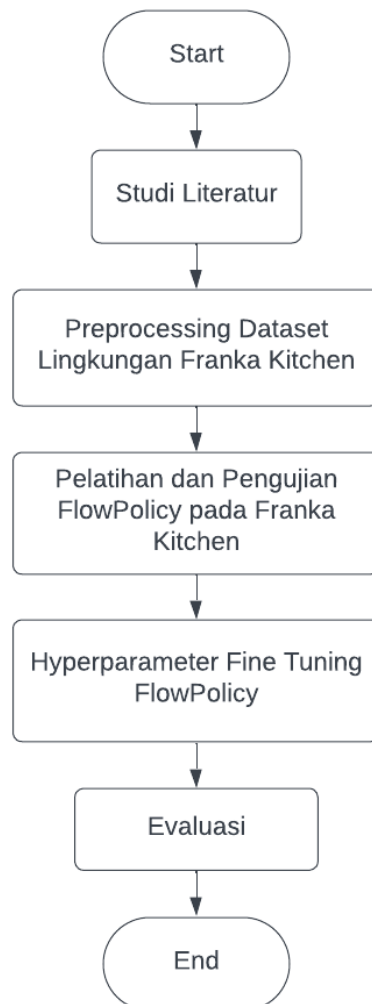
3.4 Prosedur Penelitian

Penelitian ini dilaksanakan melalui lima tahapan utama yang sistematis. Tahap pertama dimulai dengan studi literatur mendalam mengenai arsitektur FlowPolicy yaitu *flow matching model* dan *encoder state* franka kitchen. Tahap kedua penelitian ini adalah pengambilan dan *preprocessing* dataset demonstrasi dan konfigurasi lingkungan simulasi franka kitchen menggunakan *library* gymnasium. Setelah tahap *preprocessing* dilakukan, pelatihan model FlowPolicy dapat dilakukan dengan konfigurasi parameter awal yang mengacu pada literatur sebagai model acuan. Hasil training tersebut disimpan dan dapat digunakan untuk menguji model pada lingkungan franka kitchen.

Tahap berikutnya difokuskan pada menemukan variasi variabel bebas FlowPolicy yang cocok untuk lingkungan franka kitchen. Setiap konfigurasi dilatih dan diuji pada skenario tugas manipulasi yang sama dengan jumlah percobaan yang

konsisten. Evaluasi kinerja dilakukan berdasarkan metrik *success rate* dan *inference latency* pada setiap sub-tugas manipulasi untuk menganalisis *trade-off* antara performa dan efisiensi komputasi dari pendekatan yang diusulkan.

Hasil evaluasi selanjutnya dibandingkan dengan model *diffusion policy* yang diperoleh dari literatur sebagai referensi performa. Analisis ini digunakan untuk menarik kesimpulan mengenai efektivitas FlowPolicy dari model *imitation learning* yang sebelumnya diterapkan dalam lingkungan *franka kitchen*. Diagram alur prosedur penelitian secara lengkap disajikan pada Gambar 3.1.



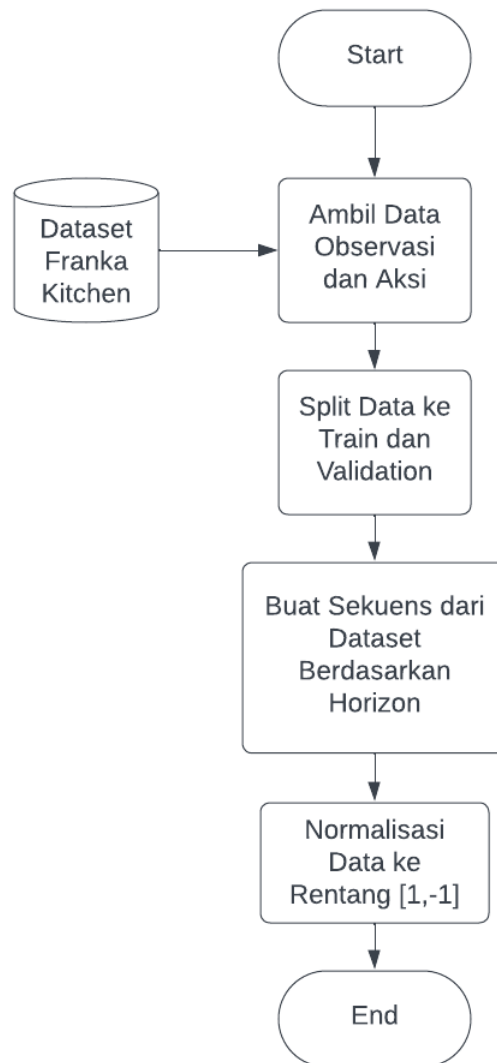
Gambar 3.4.1 Skema Prosedur Penelitian

3.4.1 Jadwal Penelitian

Tabel 3.4.1 Jadwal Penelitian

	Juni 2025 – Desember 2025	Januari 2026	Februari 2026	Maret 2026	April 2026	Mei 2026	Juni 2026
Studi Literatur							
Preprocessing Dataset							
Pelatihan dan Pengujian FlowPolicy pada Franka Kitchen							
Hyperparameter Fine Tuning FlowPolicy							
Evaluasi							

3.4.2 Preprocessing Dataset Lingkungan Franka Kitchen



Gambar 3.4.2 Tahap *Preprocessing Dataset* Franka Kitchen

Preprocessing dataset pada lingkungan franka kitchen dilakukan berdasarkan alur yang ditunjukkan pada Gambar 3.4.2. Mengetahui dataset demonstrasi awal yaitu kitchen-complete-v2 memiliki jumlah data yang sedikit yaitu 19 episode dengan total 4209 *timesteps*. Proses *preprocessing* dataset ini dirancang secara khusus untuk mencegah model *overfitting*. Langkah pertama dimulai dengan mengekstraksi informasi *state* lingkungan serta perintah aksi dalam ruang kecepatan sendi robot.

Guna memaksimalkan jumlah sampel dari data awal yang terbatas, data

tersebut kemudian disusun menjadi sekuens menggunakan teknik *sliding window* dengan *stride* 1. Pembentukan sekuens yang saling tumpang tindih berdasarkan panjang *horizon* ini bertujuan untuk memungkinkan model mempelajari dependensi temporal antar *state*, sekaligus melipatgandakan jumlah sampel yang dapat dipelajari (Mrowca et al., 2022). Selain itu, teknik augmentasi data berupa penambahan *gaussian noise* berskala kecil pada vektor *state* juga diterapkan untuk melatih ketahanan model terhadap pergeseran distribusi (Yang et al., 2026).

Setelah melalui tahap operasi tersebut, dataset dipisahkan menjadi dua bagian dengan rasio 80% untuk data latih dan 20% untuk data validasi. Pembagian ini bertujuan untuk memastikan bahwa proses evaluasi model dilakukan secara objektif terhadap data yang tidak pernah dipelajari selama masa pelatihan. Kumpulan data yang telah diproses dan dipisahkan ini kemudian akan digunakan sebagai *input* utama pada tahap pelatihan arsitektur *state based FlowPolicy*.

3.4.3 Pelatihan dan Pengujian FlowPolicy pada Franka Kitchen

Fase pelatihan pertama diawali dengan inisialisasi *hyperparameter* di Tabel 3.4.2 berikut yang digunakan pada model *encoder FlowPolicy* dan *flow network* berbasis *consistency flow matching*.

Tabel 3.4.2 Hyperparameter FlowPolicy

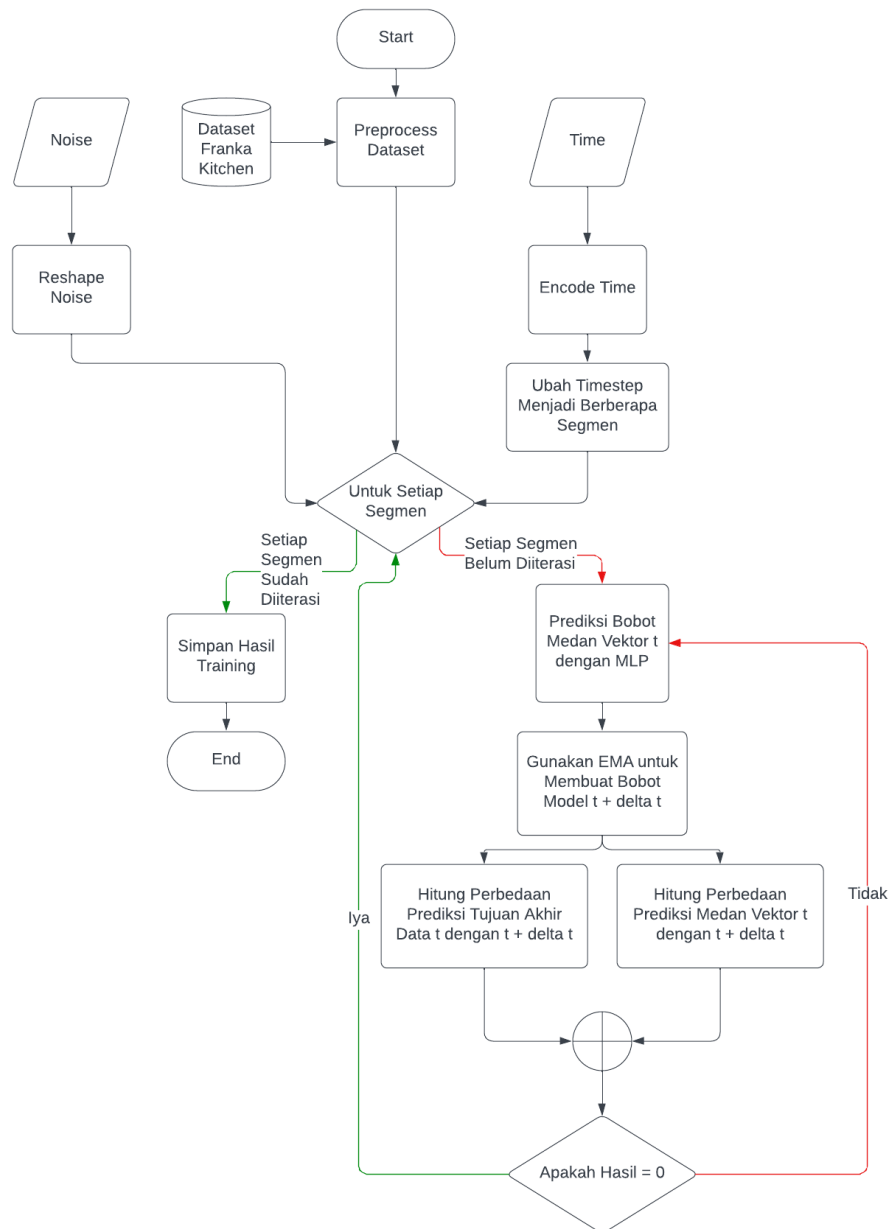
FlowPolicy	Konfigurasi
<i>Epoch</i>	500
<i>Learning Rate</i>	1e-4
<i>Batch Size</i>	128
<i>Seed</i>	100
Jumlah <i>hidden dimension</i>	512
<i>Time embedding dimension</i>	256
Jumlah Segmen	3
<i>Epsilon</i>	0.01
<i>Delta</i>	1
<i>Delta t</i>	0.02

<i>Action Horizon</i>	2
<i>Observation Horizon</i>	4

Hyperparameter ini akan disesuaikan dan dievaluasi untuk memperoleh konfigurasi yang paling optimal dalam lingkungan franka kitchen pada tahap selanjutnya.

Selain itu, proses pelatihan maupun pengujian model tidak dilakukan dalam sekali percobaan, tetapi eksperimen akan diulang sebanyak 100 kali menggunakan *random seed* yang berbeda. Pengendalian *seed* ini diterapkan secara menyeluruh pada semua aspek komputasi, seperti inisialisasi bobot awal *neural network*, pengacakan susunan data *batch* selama proses pelatihan, hingga penentuan *reset state* awal pada lingkungan simulasi franka kitchen Pendekatan ini merupakan standar evaluasi empiris untuk memvalidasi bahwa performa dan konvergensi dari arsitektur *state based* FlowPolicy bersifat konsisten serta tidak bergantung pada satu konfigurasi acak tertentu.

Setelah pemrosesan dataset dan inisiasi *hyperparameter*, proses pelatihan dimulai mengikuti Gambar 3.4.3 berikut.



Gambar 3.4.3 Pelatihan FlowPolicy pada Lingkungan Franka Kitchen

Tahap pertama proses pelatihan ini adalah mengambil sampel *random noise* yang mengikuti distribusi *gaussian*. *Noise* ini kemudian diubah bentuknya agar sesuai dengan dimensi *state* aksi yang didapat dari dataset yang telah di proses. Sementara itu, variabel waktu juga diproses melalui tahap *encoding* sehingga dapat direpresentasikan sebagai vektor fitur yang informatif bagi jaringan saraf. Nilai waktu tersebut selanjutnya dibagi ke dalam beberapa segmen untuk mendukung

mekanisme pelatihan berbasis *multi segment consistency flow matching*.

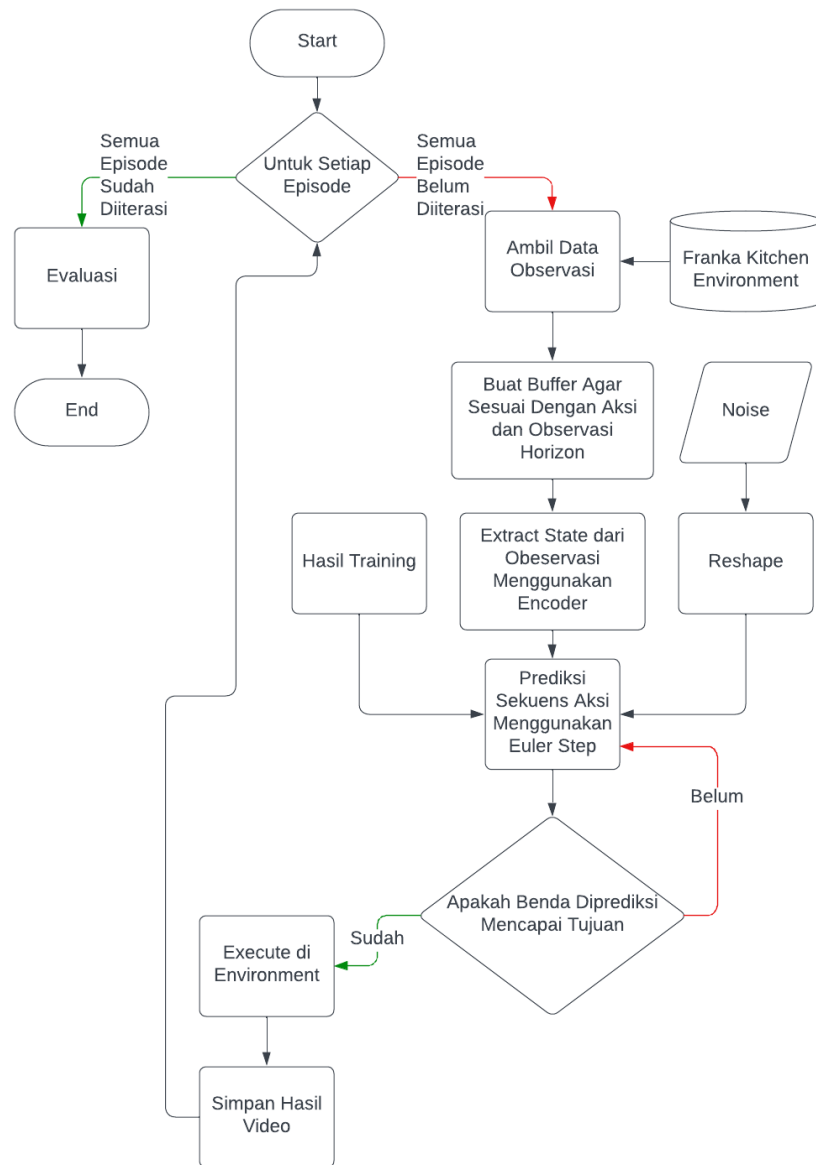
Pada setiap segmen, FlowPolicy dilatih untuk memprediksi medan vektor sesuai dengan Persamaan (1), dengan mengganti variabel umum x menjadi aksi a_t yang pada tahap awal berasal dari *noise*.

$$\begin{cases} \frac{d\xi_{a_t}(t)}{dt} = v_\theta(t, \xi_{a_t}(t)), \\ \xi_{a_t}(0) = a_t, \end{cases} \quad (9)$$

Persamaan (9) ini menyatakan bahwa model mempelajari medan vektor v_θ yang mengarahkan transformasi bertahap dari *noise* menuju distribusi aksi demonstrasi. Dengan kata lain, model menyesuaikan arah perubahan aksi sepanjang waktu sehingga aliran yang terbentuk mengikuti pola data. Untuk memastikan konsistensi antar waktu dalam setiap segmen, medan vektor yang diprediksi kemudian dioptimalkan menggunakan *multi segment loss function* pada Persamaan (6) yang disesuaikan dengan variabel kondisi (t, a_t, s) .

$$\begin{aligned} \mathcal{L}_\theta &= \mathbb{E}_{t \sim \mathcal{U}^i} \lambda^i \mathbb{E}_{x_t, x_{t+\Delta t}} \left\| f_\theta^i(t, a_t, s) - f_\theta^i(t + \Delta t, a_{t+\Delta t}, s) \right\|_2^2 + \\ &\quad \alpha \left\| v_\theta^i(t, a_t, s) - v_\theta^i(t + \Delta t, a_{t+\Delta t}, s) \right\|_2^2, \\ f_\theta^i(t, a_t, s) &= a_t + \left(\frac{i+1}{K} - 1 \right) \cdot v_\theta^i(t, a_t, s) \end{aligned} \quad (10)$$

Melalui Persamaan (10) ini, model tidak hanya didorong untuk menghasilkan aliran aksi yang mendekati target, tetapi juga menjaga konsistensi medan vektor antara waktu t dan $t + \Delta t$ dalam setiap segmen. Proses ini diulang pada seluruh segmen hingga parameter model terkonvergensi dan mampu memetakan *noise* awal menjadi aksi yang selaras dengan distribusi demonstrasi. Hasil dari fase *training* kemudian disimpan dalam format file *.pt* yang berisi parameter model terlatih dan selanjutnya digunakan untuk pengujian pada lingkungan *franka kitchen* yang tidak termasuk dalam data pelatihan, seperti yang diilustrasikan pada Gambar 3.4.4 berikut.



Gambar 3.4.4 Pengujian FlowPolicy pada Lingkungan Franka Kitchen

Pada tahap pengujian, model dijalankan dalam beberapa episode untuk mengevaluasi tingkat keberhasilan penyelesaian tugas serta latensi inferensi yang dihasilkan. Evaluasi ini bertujuan untuk mengukur kemampuan generalisasi model terhadap kondisi lingkungan baru sekaligus efisiensi komputasinya saat menghasilkan aksi.

Tahap awal pengujian FlowPolicy dilakukan dengan melakukan *encoding*

terhadap *state* yang diperoleh langsung dari lingkungan *franka kitchen*. *State* ini berperan sebagai kondisi *s* yang mengondisikan prediksi aksi. Selanjutnya, dibuat sampel *noise* acak yang kemudian disesuaikan dimensinya agar kompatibel dengan representasi aksi hasil *encoder*. Dengan memanfaatkan parameter hasil pelatihan, model kemudian memetakan distribusi *noise gaussian* menuju distribusi aksi target melalui proses integrasi bertahap berdasarkan medan vektor yang telah dipelajari. Proses ini dapat dirumuskan dengan Persamaan (11) dan (12) berikut yang di modifikasi dari Persamaan (7) dan (8).

$$a_{i/k} = a_{(i-1)/K} + \frac{1}{K} * v_{\theta}^i((i-1)/K, a_{(i-1)/K}, s), i = 1, 2, \dots K-1 \quad (11)$$

Persamaan (11) tersebut menunjukkan pembaruan aksi secara bertahap pada setiap segmen ke-*i*. Secara umum, integrasi numerik juga dapat dinyatakan sebagai.

$$a_{t+\Delta t} = a_t + \Delta t * v_{\theta}^i(t, a_t, s), t \in [i/K, (i+1)/K - \Delta t] \quad (12)$$

Persamaan (12) merepresentasikan metode integrasi euler untuk menghitung aksi pada waktu berikutnya berdasarkan medan vektor yang diprediksi model. Melalui mekanisme tersebut, *noise* awal ditransformasikan secara bertahap menjadi aksi yang mengikuti distribusi demonstrasi sehingga model mampu menghasilkan aksi yang dapat dievaluasi secara langsung pada lingkungan simulasi.

3.4.4 Hyperparameter Fine Tuning FlowPolicy

Dalam setiap sesi pelatihan dan pengujian, nilai-nilai *hyperparameter* pada Tabel 3.4.3 diubah-ubah secara acak guna mengeksplorasi kombinasi yang memberikan performa optimal.

Tabel 3.4.3 Ruang Lingkup Hyperparameter FlowPolicy

FlowPolicy	Konfigurasi
<i>Epoch</i>	[100, 500, 1000, 1500]
<i>Learning Rate</i>	[1e-4, 1e-5, 5e-4]

<i>Batch Size</i>	[64, 128, 256]
<i>Seed</i>	100
Jumlah <i>hidden dimension</i>	[256, 512, 1024]
<i>Time embedding dimension</i>	[128, 256, 512]
Jumlah Segmen	[1,2,3,4,5]
<i>Epsilon</i>	[0.01, 0.0001, 0.005]
<i>Delta</i>	[1, 0.001, 0.1]
<i>Delta t</i>	[0.01, 0.001, 0.1]
<i>Action Horizon</i>	[2, 4, 8, 16]
<i>Observation Horizon</i>	[4, 8, 16, 32]

Hyperparameter fine tuning pada FlowPolicy bertujuan untuk menemukan konfigurasi terbaik yang dapat digunakan dalam menyelesaikan tugas di lingkungan franka kitchen. Pendekatan ini memungkinkan identifikasi parameter yang paling sesuai serta menghindari bias dari pemilihan nilai secara manual.

Proses penyesuaian *hyperparameter* pada arsitektur *state based* FlowPolicy diimplementasikan menggunakan metode pencarian sistematis berupa *Random Search*. Eksperimen untuk pencarian kombinasi ini akan dieksekusi sebanyak 30 percobaan dengan mengeksplorasi ruang pencarian pada batasan nilai yang telah ditetapkan. Kriteria utama dalam penentuan konfigurasi terbaik didasarkan pada model yang mampu mencapai *success rate* tertinggi pada tahap validasi tanpa menunjukkan gejala ketidakstabilan pelatihan seperti *exploding gradient*. Selain untuk mencari konfigurasi yang optimal, proses evaluasi ini juga dirancang untuk melakukan analisis sensitivitas. Analisis tersebut bertujuan untuk mengidentifikasi *hyperparameter* yang paling berpengaruh secara signifikan terhadap performa model, terutama pada parameter jumlah *inference step* dan jumlah segmen pada *flow matching*.

3.5 Evaluasi

Evaluasi kinerja sistem *state based* FlowPolicy pada simulasi robot franka

emika panda dilakukan untuk menilai efektivitas dan efisiensi model dalam menyelesaikan tugas-tugas manipulasi sekuensial pada lingkungan franka kitchen. Evaluasi ini difokuskan pada empat metrik utama, yaitu *success rate*, *average completion length*, *inference latency*, dan *failure mode analysis*. Guna memastikan keandalan dan stabilitas eksperimen, seluruh proses evaluasi pada metrik kuantitatif dieksekusi secara berulang menggunakan 100 random seed yang berbeda. Oleh karena itu, hasil pengujian tidak dilaporkan sebagai nilai tunggal, melainkan dihitung berdasarkan agregasi dari 100 eksperimen independen tersebut dan disajikan dalam format nilai rata-rata beserta simpangan bakunya. Pendekatan pelaporan ini diterapkan untuk memvalidasi bahwa performa model terbebas dari bias inisialisasi awal yang bersifat acak

3.5.1 Success Rate

Success rate digunakan untuk mengukur kemampuan robot dalam menyelesaikan rangkaian sub-tugas manipulasi secara berurutan. Sesuai dengan standar pengujian *benchmark* pada lingkungan franka kitchen, batas maksimum langkah per episode ditetapkan secara ketat sebanyak 280 langkah (Gupta et al., 2019). Untuk menguji kemampuan generalisasi model terhadap variasi kondisi awal lingkungan, status tata letak objek pada awal setiap episode akan sesuai dengan batas variansi yang disediakan oleh *library* gymnasium. *Success rate* didefinisikan sebagai rasio antara jumlah percobaan yang berhasil terhadap jumlah total percobaan yang dilakukan, yang dirumuskan sebagai berikut.

$$\text{success rate} = \frac{N_{\text{success}}}{N_{\text{total}}} \times 100\% \quad (13)$$

Dimana N_{success} menyatakan jumlah episode yang berhasil diselesaikan dan N_{total} merupakan total episode evaluasi. Metrik ini mencerminkan reliabilitas kebijakan dalam menghasilkan aksi yang konsisten untuk menyelesaikan tugas manipulasi.

Selain mengukur keberhasilan total, metrik ini mencatat persentase

keberhasilan robot dalam menyelesaikan tahapan tugas. Hal ini bertujuan untuk menganalisis pada tahap mana kegagalan paling sering terjadi.

$$\text{success rate}_k = \frac{N_{\text{success}_k}}{N_{\text{total}}} \times 100\% \quad (14)$$

Disini k merupakan variabel yang mempresentasikan jumlah sub-tugas yang berhasil diselesaikan secara berurutan.

3.5.2 Average Completion Length

Selain tingkat keberhasilan, penelitian ini juga mengevaluasi efisiensi temporal kebijakan melalui metrik rata-rata waktu penyelesaian semua sub-tugas. Metrik ini mengukur rata-rata *timesteps* yang dibutuhkan robot untuk menyelesaikan seluruh rangkaian tugas secara tuntas dalam episode yang berhasil. Nilai L_{avg} yang lebih rendah mengindikasikan bahwa FlowPolicy mampu menghasilkan lintasan yang lebih langsung dan efisien tanpa gerakan redundan. Rumus perhitungan didefinisikan sebagai berikut.

$$L_{\text{avg}} = \frac{1}{N_{\text{success}}} \sum_{i=1}^{N_{\text{success}}} T_i \quad (15)$$

Pada Persamaan (15) ini L_{avg} merepresentasikan rata-rata panjang penyelesaian, N_{success} adalah jumlah total episode yang berhasil diselesaikan, dan T_i menyatakan jumlah langkah waktu yang dikonsumsi secara spesifik pada episode berhasil ke- i .

3.5.3 Inference Latency

Inference latency atau latensi inferensi merupakan metrik evaluasi yang digunakan untuk mengukur lama waktu yang dibutuhkan model dalam melakukan proses inferensi atau pengujian pada setiap iterasi. Pada Gambar 3.4.4, proses pengukuran latensi dihitung menggunakan formula waktu akhir dikurangi waktu

awal pada setiap prediksi sekuens aksi atau pada setiap iterasi model.

Selain hal itu, guna menghilangkan bias waktu akibat proses pemanasan perangkat keras atau GPU *warm-up*, metode *dummy pass* diterapkan sebelum evaluasi utama dimulai. Proses ini dilakukan dengan menerima angka tensor acak yang memiliki dimensi serupa dengan observasi *state* ke dalam model sebanyak puluhan iterasi. Hal ini memastikan inisialisasi CUDA dan alokasi memori VRAM telah mencapai kondisi stabil saat pencatatan waktu sebenarnya dilakukan.

Secara kuantitatif, *inference latency* (Lat) dapat dinyatakan sebagai selisih antara waktu akhir pengujian dan waktu awal pengujian, seperti yang diformulasikan pada persamaan

$$\text{Lat} = t_{\text{akhir pengujian}} - t_{\text{awal pengujian}} \quad (17)$$

Apabila hasil nilai latensi pada Persamaan (17) menunjukkan nilai yang rendah maka model mampu merespons perubahan lingkungan dengan cepat, sedangkan sebaliknya apabila hasil nilai latensi tinggi maka model berpotensi menyebabkan keterlambatan aksi yang dapat mengganggu kelancaran eksekusi tugas.

3.5.4 Failure Mode Analysis

Failure mode analysis adalah metode evaluasi kualitatif yang digunakan untuk mengetahui penyebab kegagalan suatu sub-tugas secara mendetail. Tipe kegagalan bisa seperti benturan dengan lingkungan sekitar atau antar bagian tubuh robot, kegagalan penggengaman di mana objek terlepas dari *gripper* atau gagal diraih, serta kegagalan penyelesaian tugas seperti sudut bukaan pintu *microwave* yang tidak mencapai target radian yang ditentukan atau penempatan objek yang meleset dari koordinat tujuan. Analisis ini bertujuan untuk mengidentifikasi kelemahan spesifik pada perilaku robot yang dihasilkan oleh FlowPolicy sehingga dapat memberikan wawasan mengenai aspek gerakan mana yang memerlukan perbaikan lebih lanjut.

DAFTAR PUSTAKA

Automatic citation updates are disabled. To see the bibliography, click Refresh in the Zotero tab.

LAMPIRAN

Lampiran 1 Lingkungan Franka Kitchen

