



KERJA PRAKTIK - IF184801

**FITUR EXTRACTION UNTUK Mencari Kecepatan
Relatif Berdasarkan Data Dascham Kendaraan**

Teknik Informatika - Kampus ITS

Sukolilo Surabaya

Periode: 1 Oktober – 31 Desember 2024

Oleh:

Akmal Nafis

5025211216

Pembimbing Jurusan

Ilham Gurrat Adillion, S.Kom, M.Eng.

Pembimbing Lapangan

Dr. Eng. Muhamad Hilmi Muchtar Aditya Pradana, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025



KERJA PRAKTIK - IF184801

**Fitur Extraction untuk Mencari Kecepatan Relatif
Berdasarkan Data Dashcam Kendaraan**

Teknik Informatika - Kampus ITS

Sukolilo Surabaya

Periode: 1 Oktober 2024 – 31 Desember 2024

Oleh:

Akmal Nafis

5025211216

Pembimbing Jurusan

Ilham Gurrat Adillion, S.Kom, M.Eng.

Pembimbing Lapangan

Dr. Eng. Muhamad Hilmil Muchtar Aditya Pradana, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR KODE SUMBER	xi
LEMBAR PENGESAHAN	xiii
KATA PENGANTAR	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah	3
1.5. Lokasi dan Waktu Kerja Praktik	3
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur	4
1.6.3. Analisis dan Perancangan Sistem	4
1.6.4. Pengujian dan Evaluasi	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan	4
1.7.2. Bab II Profil Perusahaan	4
1.7.3. Bab III Tinjauan Pustaka	4

1.7.4.	Bab IV Perancangan & Implementasi sistem	5
1.7.5.	Bab V Pengujian dan Evaluasi	5
1.7.6.	Bab VI Kesimpulan dan Saran	5
BAB II PROFIL PERUSAHAAN		7
2.1.	Profil Departemen Teknik Informatika ITS	7
2.2.	Lokasi	7
BAB III TINJAUAN PUSTAKA		9
3.1.	Optical Flow	9
3.2.	Deeplab V3+	10
3.3.	SerNet-Former	11
3.4.	Propainter	12
3.5.	RAFT	13
BAB IV PERANCANGAN DAN IMPLEMENTASI		15
4.1.	Rancangan Sistem	15
4.2.	Implementasi Sistem	15
4.2.1.	Metodologi	15
4.2.2.	Implementasi SerNet-Former	17
4.2.3.	Implementasi Propainter	18
4.2.4.	Implementasi Raft	22
BAB V PENGUJIAN DAN EVALUASI		25
5.1	Tujuan Pengujian	25
5.2	Kriteria Pengujian	25
5.3	Skenario Pengujian	25

5.4	Hasil Pengujian	26
BAB VI KESIMPULAN DAN SARAN		28
6.1	Kesimpulan	28
6.2	Saran	29
DAFTAR PUSTAKA		31

DAFTAR GAMBAR

Gambar 3. 1 Diagram arsitektur DeepLabV3+ yang memanfaatkan atrous convolution dan <i>atrous spatial pyramid pooling</i> (ASPP) dalam encoder (Wang et al., 2022).....	11
Gambar 3.2 Arsitektur Propainter yang didalamnya terdapat proses mask-aware propagation [6].....	13
Gambar 3.3 Arsitektur RAFT untuk estimasi optical flow berdasarkan paper RAFT [7]	14
Gambar 4.1 Diagram sistem yang dibangun.....	15
Gambar 5.1 Hasil pengujian sistem untuk implementasi inpainting pada optical flow	27

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Pseudocode 4.1 pipeline sistem utama	17
Pseudocode 4.2 Proses inti pada Propainter	19
Pseudocode 4.3 Fungsi untuk menyimpan lokasi objek menjadi binary mask	19
Pseudocode 4.4 Binary dilatation untuk memperluas masking.....	21
Pseudocode 4.5 extrapolation Propainter	22
Pseudocode 4.6 Fungsi demo pada RAFT untuk memproses dua frame menjadi optical flow	24

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Fitur Extraction Untuk Mencari Kecepatan Relatif
Berdasarkan Data Dashcam Kendaraan**

Oleh:

Akmal Nafis

5025211216

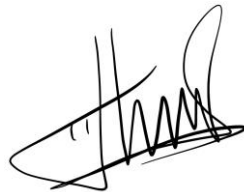
Disetujui oleh Pembimbing Kerja Praktik:

1. Ilham Gurrat Adillion, S.Kom,
M.Eng.,



(Pembimbing Departemen)

2. Dr. Eng. Muhamad Hilmi
Muchtar Aditya Pradana, S.Kom,
M.Sc.
NIP. 199205162024061001



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Fitur Extraction untuk Mencari Kecepatan Relatif Berdasarkan Data Dashcam Kendaraan

Nama Mahasiswa 1 : Akmal Nafis
NRP : 5025211216
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Ilham Gurrat Adillion, S.Kom,
M.Eng
Pembimbing Lapangan : Dr. Eng. Muhamad Hilmil Muchtar
Aditya Pradana, S.Kom., M.Sc

ABSTRAK

Estimasi kecepatan kendaraan merupakan komponen fundamental dalam pengembangan sistem kendaraan otonom yang mempengaruhi kualitas pengambilan keputusan pada berbagai situasi dinamis. Data kecepatan yang akurat sangat essential untuk mendukung navigasi yang aman dan efisien. Meskipun sensor kecepatan konvensional dapat menyediakan informasi langsung, ketergantungan pada sensor tunggal berpotensi meningkatkan risiko kegagalan sistem.

Penelitian ini mengimplementasikan *pipeline* terintegrasi yang menggabungkan sebagai fitur ekstraksi untuk estimasi menggunakan *optical flow*. Hasil implementasi menunjukkan bahwa sistem yang dikembangkan berhasil mengintegrasikan ketiga komponen dalam satu pipeline yang berfungsi efektif pada data video dashcam, dengan SerNet-Former memberikan segmentasi yang memuaskan, ProPainter melakukan *inpainting* yang sukses, dan RAFT menghasilkan estimasi *optical flow* yang lebih bersih dan konsisten

Kata Kunci : Estimasi Kecepatan, Segmentasi Citra, Inpainting, Optical Flow, SerNet-Former, ProPainter, RAFT

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Fitur *Extraction* untuk Mencari Kecepatan Relatif Berdasarkan Data Dashcam Kendaraan.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Selaku dosen pembimbing departemen kerja praktik bapak Ilham Gurrat Adillion, S.Kom, M.Eng.
3. Bapak Dr. Eng. Muhamad Hilmil Muchtar Aditya Pradana, S.Kom., M.Sc. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 29 Januari 2025

Akmal Nafis

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Belakangan ini, data yang menggabungkan berbagai jenis informasi seperti gambar, video, dan data sensor telah menjadi sangat penting dalam pengembangan teknologi transportasi, terutama untuk kendaraan yang bisa mengemudi sendiri. Keunggulan data multimodal adalah ketika satu sensor rusak atau bermasalah, sensor lainnya masih bisa bekerja sebagai cadangan, sehingga sistem tetap dapat berfungsi dengan baik [1].

Masalahnya, mendapatkan data lengkap tentang kecelakaan kendaraan itu sangat sulit. Kita tidak mungkin sengaja menabrakkan mobil hanya untuk mengumpulkan data, dan rekaman dashcam yang ada biasanya tidak dilengkapi dengan informasi sensor yang lengkap. Karena itulah, penelitian ini mencoba mengembangkan cara untuk memperkirakan kecepatan kendaraan dari video dashcam saja, yang nantinya bisa melengkapi data yang dibutuhkan untuk mengembangkan mobil otonom dan menganalisis kecelakaan lalu lintas.

Ada dua cara utama yang biasa digunakan untuk memperkirakan kecepatan kendaraan: metode tradisional dan metode deep learning. Metode tradisional seperti optical flow dengan mutual information [2] bekerja dengan menganalisis pergerakan objek dalam video frame demi frame. Sayangnya, metode ini sering tidak akurat ketika pencahayaan buruk, mudah terganggu oleh noise, dan sangat sensitif terhadap perubahan cahaya. Selain itu,

kalau kameranya goyang atau sudut pandangnya berubah, hasilnya juga bisa tidak tepat.

Penelitian ini mengusulkan pendekatan baru dengan menggabungkan segmentasi menggunakan SERNet-Former, Propainter, dan optical flow RAFT untuk mendapatkan citra *optical flow* yang lebih baik.

1.2. Tujuan

1. Mengembangkan metodologi baru dalam computer vision yang menggabungkan segmentasi dan optical flow
2. Memberikan solusi untuk masalah estimasi kecepatan dari perspektif kamera bergerak (*dashcam*)
3. Memperkaya literatur tentang pengolahan data single-modal untuk aplikasi transportasi

1.3. Manfaat

Penelitian tentang *inpainting* untuk *optical flow* pada *dashcam* ini memberikan manfaat praktis yang sangat terasa dalam kehidupan sehari-hari, terutama untuk investigasi kecelakaan lalu lintas yang selama ini mengandalkan kesaksian subjektif. Dengan teknologi ini, rekaman *dashcam* dapat dianalisis untuk mengetahui kecepatan kendaraan saat kejadian secara objektif, membantu penegakan hukum dan klaim asuransi. Selain itu, industri otomotif dapat mengembangkan sistem keselamatan yang lebih terjangkau dengan memanfaatkan *dashcam* yang sudah ada daripada menambah sensor mahal, sementara data kecepatan dari banyak *dashcam* juga bisa digunakan untuk mengoptimalkan arus lalu lintas dan mengurangi kemacetan di perkotaan. Secara jangka panjang, penelitian ini mendorong kemandirian teknologi dalam negeri di bidang transportasi otonom.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan *pipeline* terintegrasi yang menggabungkan SERnet-Former untuk segmentasi, ProPainter untuk *inpainting* dan RAFT untuk *optical flow* dalam memproses data *dashcam* kendaraan?
2. Bagaimana mengoptimalkan urutan pemrosesan dan transfer data antar modul?
3. Bagaimana mengintegrasikan output segmentasi SERNet-Former sebagai input untuk ProPainter dan memastikan hasil *inpainting* yang optimal untuk perhitungan RAFT *optical flow*?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut :

Lokasi : Teknik Informatika ITS

Alamat : Gedung Teknik Informatika ITS – Kampus ITS
Sukolilo, Surabaya, 60117.

Waktu : 1 Oktober – 31 Desember 2024

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Tahap ini diperlukan untuk memahami kebutuhan dari pengimplementasian penggunaan segmentasi dan *inpainting* pada citra *optical flow*. Diskusi ini mencakup mengenai apa jenis kendaraan yang akan menjadi *inpainting*.

1.6.2. Studi Literatur

Setelah memahami kebutuhan proyek, kami mempelajari penelitian-penelitian terdahulu terkait penggunaan segmentasi pada citra dashcam kendaraan.

1.6.3. Analisis dan Perancangan Sistem

Sistem dibuat berdasarkan arahan yang telah diberikan pembimbing lapangan KP. Sistem dibangun mengikuti arsitektur sistem yang disesuaikan.

1.6.4. Pengujian dan Evaluasi

Setelah proses pengembangan sistem dilakukan terdapat evaluasi untuk memastikan bahwa program berjalan dengan baik dan memiliki hasil yang konsisten.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Departemen Teknik Informatika ITS mulai dari profil, lokasi perusahaan.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Perancangan & Implementasi sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.6. Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Departemen Teknik Informatika ITS

Departemen Teknik Informatika (DTI) merupakan salah satu departemen besar yang diakui secara nasional dan internasional di Institut Teknologi Sepuluh Nopember (ITS). Sejak tahun 2020, DTI menjadi bagian dari Fakultas Teknologi Elektro dan Informatika Cerdas (FT-EIC), menawarkan program studi mulai dari S1 hingga S3.

Departemen ini memiliki delapan bidang minat utama: Algoritma dan Pemrograman (AP), Arsitektur dan Jaringan Komputer (AJK), Grafika, Interaksi, dan Game (GIGa), Komputasi Berbasis Jaringan (KBJ), Komputasi Cerdas dan Visi (KCV), Manajemen Cerdas Informasi (MCI), Pemodelan dan Komputasi Terapan (PKT), serta Rekayasa Perangkat Lunak (RPL).

2.2. Lokasi

Departemen Teknik Informatika - Kampus ITS Sukolilo, Surabaya, Jawa Timur.

[Halaman ini sengaja dikosongkan]

BAB III TINJAUAN PUSTAKA

3.1. Optical Flow

Optical flow adalah teknik dalam komputer visi yang digunakan untuk menganalisis pergerakan objek dalam urutan gambar atau video. Metode ini bekerja dengan mempelajari perubahan intensitas piksel antara dua frame berurutan, yang kemudian digunakan untuk memperkirakan arah serta kecepatan gerakan. Dengan memanfaatkan perbedaan visual yang muncul akibat perpindahan objek, *optical flow* mampu menghasilkan pola gerakan berupa medan vektor, yang sering disebut sebagai medan *optical flow*.

Dalam persamaan (3.1), I_x , I_y , dan I_t merepresentasikan gradien intensitas gambar masing-masing pada sumbu x , y , dan waktu (t) yang berfungsi untuk menggambarkan perubahan intensitas piksel dalam citra. Vektor $u(x, y)$ dan $v(x, y)$ menunjukkan komponen dari *optical flow* pada koordinat (x, y) , yang merepresentasikan arah serta kecepatan pergerakan objek dalam gambar. Selain itu, terdapat konstanta regulasi yang bertugas mengontrol keseimbangan antara *brightness constancy constraint* dan tingkat kehalusan pergerakan (*smoothness constraint*). Sedangkan ∇u dan ∇v mengacu pada gradien spasial dari komponen horizontal dan vertikal *optical flow* yang menggambarkan perubahan posisi [3].

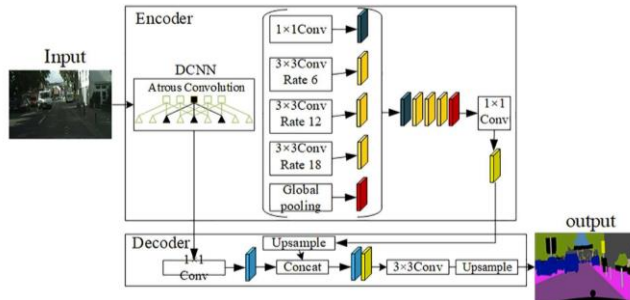
$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] dx dy \quad (3.1)$$

Dalam prosesnya, teknik ini banyak diterapkan dalam berbagai bidang seperti pemantauan lalu lintas, deteksi objek bergerak, dan analisis dinamika gerakan dalam video. *Optical flow* dapat membantu dalam pengembangan sistem pengawasan berbasis komputer, pemodelan interaksi antara kendaraan, dan bahkan dalam penerapan teknologi kecerdasan buatan untuk kendaraan otonom. Dengan mengamati pola gerakan dari satu frame ke frame berikutnya, sistem dapat mengenali perilaku objek di dalam adegan serta memprediksi pergerakan yang akan terjadi.

3.2. Deeplab V3+

DeepLab v3+ adalah salah satu model *deep learning* untuk tugas *semantic segmentation* sebagai penyempurnaan dari arsitektur sebelumnya, DeepLab v3. Seperti dalam Gambar 3.1 model ini menggabungkan *Atrous Spatial Pyramid Pooling (ASPP)* untuk menangkap informasi *multi-scale* dengan stuktur *encoder-decoder* yang berguna memperbaiki segmentasi objek agar menghasilkan segmentasi yang detail. Dengan menggunakan *dilated atrous convolutions*, DeepLab v3+ mampu memperluas jangkauan segmentasi tanpa kehilangan *resolusi spasial*, sehingga meningkatkan akurasi segmentasi. Selain itu, arsitektur ini mengintegrasikan modul *Xception* dan *separable convolution* untuk meningkatkan efisiensi komputasi. DeepLab v3+ telah terbukti unggul dalam berbagai dataset seperti PASCAL VOC dan Cityscapes, menjadikannya

pilihan populer dalam aplikasi computer vision seperti otonom kendaraan dan analisis citra medis.



Gambar 3. 1 Diagram arsitektur DeepLabV3+ yang memanfaatkan atrous convolution dan atrous spatial pyramid pooling (ASPP) dalam encoder [4].

3.3. SerNet-Former

SERNet-Former merupakan inovasi terbaru dalam teknologi semantic segmentation yang dikembangkan dengan fokus pada efisiensi komputasi sambil tetap mampu mengintegrasikan data dari berbagai bagian citra secara optimal. Arsitektur yang digunakan adalah encoder-decoder berbasis Efficient-ResNet. Pada bagian encoder, terdapat komponen *attention-boosting gates* (AbGs) dan *attention-boosting modules* (AbMs) yang berperan dalam memaksimalkan penangkapan informasi krusial yang sering diabaikan oleh sistem jaringan konvensional.

Model SERNet-Former juga memiliki fitur *attention-fusion networks* (AfNs) yang bertugas menggabungkan kembali data spasial dari output encoder selama tahap upsampling [5]. Kemampuan model ini

dalam menggabungkan informasi dari beragam tingkat skala dan kedalaman secara efisien bertujuan untuk meningkatkan akurasi prediksi setiap piksel. Evaluasi yang dilakukan menggunakan dataset Cityscapes membuktikan bahwa SERNet-Former dapat meraih tingkat akurasi yang tinggi secara menyeluruh. Pada intinya, SERNet-Former menghadirkan pendekatan yang efisien dalam bidang segmentasi citra.

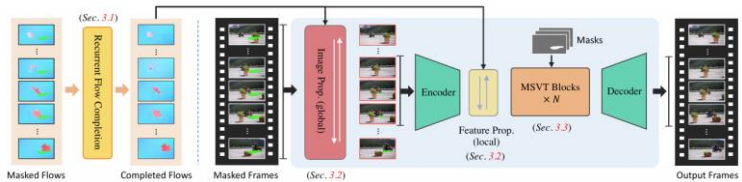
Kelebihan pokok dari SERNet-Former adalah kapasitasnya dalam menciptakan keseimbangan antara kinerja dan efisiensi komputasi, sehingga sangat cocok untuk implementasi real-time seperti pada sistem kendaraan otonom, pencitraan medis, dan bidang robotika. Berbeda dengan model transformer konvensional yang membutuhkan kapasitas komputasi besar, SERNet-Former menghadirkan struktur yang lebih efisien tanpa menurunkan mutu segmentasi. Melalui perpaduan inovatif antara elemen *attention-boosting* dan *attention-fusion networks*, model ini tidak hanya mengoptimalkan akurasi namun juga mempercepat proses komputasi secara substansial.

3.4. Propainter

ProPainter adalah sebuah teknik canggih untuk memperbaiki bagian yang rusak atau hilang dalam video, yang diperkenalkan di konferensi ICCV 2023. Metode ini dirancang khusus untuk menghasilkan video yang lebih natural dan konsisten dengan memadukan dua pendekatan

utama yaitu *propagation* dan pemrosesan menggunakan *Transformer*.

Dibandingkan metode sebelumnya, ProPainter lebih baik dalam menyebarkan informasi dari frame-frame sekitarnya berkat teknik *mask-aware propagation* seperti pada Gambar 3.2. yang secara otomatis menyesuaikan cara mengisi bagian yang hilang berdasarkan gerakan objek dan latar belakang. Selain itu, dengan memanfaatkan *Transformer*, metode ini mampu memahami hubungan antar bagian video secara lebih menyeluruh, sehingga hasil perbaikannya terlihat lebih halus saat menangani area yang hilang besar atau gerakan yang rumit .



Gambar 3.2 Arsitektur Propainter yang didalamnya terdapat proses *mask-aware propagation* [6].

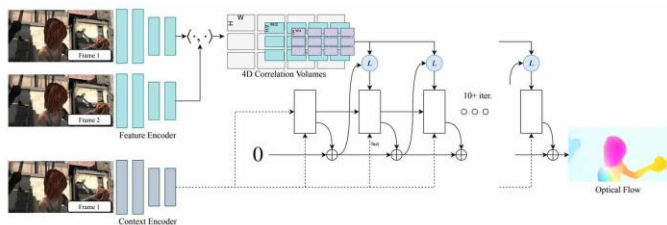
ProPainter telah diuji pada berbagai dataset video dan terbukti lebih unggul dibanding teknik-teknik sebelumnya, membuatnya sangat berguna untuk aplikasi seperti menghapus objek yang tidak diinginkan dari video atau gambar.

3.5. RAFT

RAFT (*Recurrent All-Pairs Field Transforms*) adalah sebuah arsitektur *deep learning* untuk estimasi *optical flow* yang dirancang untuk memprediksi

pergerakan piksel antar-frame video secara akurat dan efisien. Berbeda dengan metode tradisional yang bergantung pada pendekatan konvolusional atau *one-shot prediction*, RAFT menggunakan mekanisme rekursif berbasis recurrent network untuk secara iteratif memperbaiki prediksi aliran optik, dimulai dari estimasi kasar hingga konvergensi ke hasil yang presisi [7].

Arsitektur ini memanfaatkan korelasi 4D semua-pasangan (*all-pairs correlation volume*) untuk menangkap hubungan antar-piksel, lalu secara bertahap menyempurnakannya melalui mekanisme berbasis GRU (*Gated Recurrent Unit*) sesuai pada Gambar 3.3. Pendekatan ini membuat RAFT sangat tahan terhadap gerakan cepat, tekstur rendah, dan perubahan cahaya, sekaligus mempertahankan efisiensi komputasi. RAFT telah menunjukkan kinerja terdepan pada benchmark seperti Sintel dan KITTI, menjadikannya pilihan utama untuk aplikasi seperti *motion tracking*, *video enhancement*, dan *autonomous driving*.

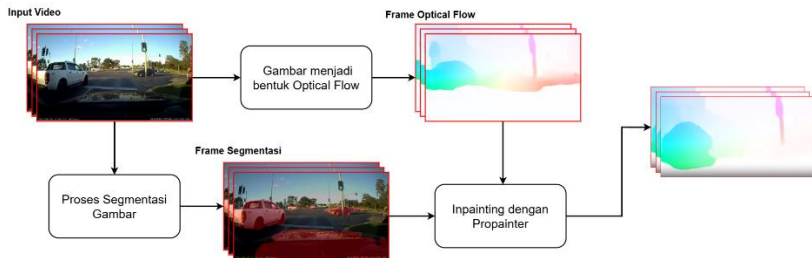


Gambar 3.3 Arsitektur RAFT untuk estimasi optical flow berdasarkan paper RAFT [7]

BAB IV PERANCANGAN DAN IMPLEMENTASI

4.1. Rancangan Sistem

Sistem ini dirancang dengan arsitektur modular yang mengintegrasikan tiga komponen utama: SerNet-Former untuk segmentasi otomatis, ProPainter untuk proses *inpainting*, dan RAFT untuk estimasi *optical flow*.



Gambar 4.1 Diagram sistem yang dibangun

4.2. Implementasi Sistem

Implementasi sistem ini mengintegrasikan tiga komponen utama: SerNet-Former untuk segmentasi objek, ProPainter untuk proses *inpainting*, dan RAFT untuk estimasi *optical flow*. Setiap komponen bekerja secara berurutan dalam *pipeline* yang terstruktur untuk menghasilkan video dengan kualitas tinggi.

4.2.1. Metodologi

Implementasi mengikuti alur kerja *pipeline* yang sistematis, dimulai dari ekstraksi *frame* video hingga rekonstruksi video final. Proses ini dirancang untuk

menangani *multiple* video secara otomatis dengan pembersihan *temporary* files untuk efisiensi storage.

```
def main():
    video_files = list_all_mp4_files(INPUT_FOLDER)
    if video_files is empty:
        print("No videos found.")
        return

    for video in video_files:
        video_path = join(INPUT_FOLDER, video)
        video_name =
        get_name_without_extension(video)

        # 1. Extract frames from video
        extract_frames(video_path,
        OUTPUT_FRAMES_FOLDER)

        # 2. Prepare options for inpainting and RAFT
        opts = prepare_propainter_options()
        raft_opts = prepare_raft_options()

        # 3. Run segmentation on each frame
        for frame in
        sorted_frames(OUTPUT_FRAMES_FOLDER):
            run_sernet_segmentation(frame)

        # 4. Run ProPainter for inpainting
        run_inpainting(opts)

        # 5. Compute optical flow between inpainted
        frames
        for each_pair in
        consecutive_frames(OUTPUT_FRAMES_PROPAINT
        ER):
```

```

run_raft_demo(raft_opts, pair[0], pair[1])

# 6. Save frames and .npy with duplication

save_frames_with_duplicates(OUTPUT_FLOW_FRAMES_FOLDER,
FLOW_IMAGE_FOLDER/video_name, video_name)

# 7. Cleanup temporary folders
clean_folder(OUTPUT_FRAMES_FOLDER)

clean_folder(OUTPUT_FRAMES_PROPAINTER)
clean_folder(MASK_FOLDER)

# 8. Create final flow video

create_video_from_frames(OUTPUT_FLOW_FRAMES_FOLDER, OUTPUT_FLOW_VIDEO)

clean_folder(OUTPUT_FLOW_FRAMES_FOLDER)

```

Pseudocode 4.1 pipeline sistem utama

Pipeline utama mendemonstrasikan integrasi ketiga model dalam satu alur kerja yang efisien. Setiap langkah dirancang untuk mempertahankan konsistensi temporal dan kualitas visual sepanjang proses transformasi video.

4.2.2. Implementasi SerNet-Former

SerNet-Former berfungsi sebagai modul segmentasi otomatis yang menghasilkan *mask* untuk objek target dalam setiap *frame* video. Model ini mendeteksi dan membuat *binary mask* dari objek yang akan dihilangkan, sehingga

ProPainter dapat melakukan *inpainting* pada wilayah yang tepat.

Pseudocode ini menunjukkan proses segmentasi otomatis yang menghasilkan *mask binary* untuk setiap *frame*. Output *mask* ini kemudian digunakan oleh ProPainter sebagai wilayah yang akan di-*inpaint*, memungkinkan penghapusan objek yang konsisten sepanjang video.

4.2.3. Implementasi Propainter

ProPainter menangani proses *inpainting* utama dengan dukungan untuk berbagai format input dan *preprocessing*. Implementasi mencakup fungsi-fungsi utilitas untuk pembacaan *frame*, pemrosesan *mask*, dan *extrapolation* untuk video *outpainting*.

```
def propainter process(args):
    gt_flows_bi = computeRAFT_flow(frames, fixRAFT,
    args.raft_iter)
    if use_half:
        frames, flow_masks, masks_dilated =
        convert_half_precision(frames, flow_masks,
        masks_dilated)
        gt_flows_bi =
        convert_half_precision(gt_flows_bi)
        fix_flow_complete, model =
        to_half(fix_flow_complete), to_half(model)

        pred_flows_bi = complete_flow(gt_flows_bi,
        flow_masks, fix_flow_complete,
        args.subvideo_length)
        masked_frames = frames * (1 - masks_dilated)
```

```

        updated_frames, updated_masks =
image_propagation(frames, masked_frames,
pred_flows_bi, masks_dilated, model,
args.subvideo_length)

    comp_frames = transformer_inference(frames_inp,
updated_frames, masks_dilated, updated_masks,
pred_flows_bi, model, args)
    save_frames_and_video(comp_frames,
masked_frame_for_save, save_root, out_size, fps)

```

Pseudocode 4.2 Proses inti pada Propainter

```

def binary_mask(mask, th=0.1):
    mask[mask>th] = 1
    mask[mask<=th] = 0
    return mask

```

Pseudocode 4.3 Fungsi untuk menyimpan lokasi objek menjadi binary mask

Binary mask memastikan region yang akan di-*inpaint* terdefinisi dengan jelas, dengan *threshold* yang dapat disesuaikan berdasarkan karakteristik dataset.

```

def read_mask(mpath, length, size, flow_mask_dilates=8,
mask_dilates=5):
    masks_img = []
    masks_dilated = []
    flow_masks = []

    # Baca satu gambar atau beberapa dari folder
    if mpath.endswith(('jpg', 'jpeg', 'png', 'JPG', 'JPEG',
'PNG')):
        with Image.open(mpath) as img:
            masks_img.append(img.copy())
    else:
        mnames = sorted(os.listdir(mpath))

```

```

for mp in mnames:
    with Image.open(os.path.join(mpath, mp)) as
img:
    masks_img.append(img.copy())

for mask_img in masks_img:
    if size is not None:
        mask_img = mask_img.resize(size,
Image.NEAREST)
    mask_img = np.array(mask_img.convert('L'))

    # Dilatasi mask khusus untuk optical flow
    if flow_mask_dilates > 0:
        flow_mask_img =
scipy.ndimage.binary_dilation(mask_img,
iterations=flow_mask_dilates).astype(np.uint8)
    else:
        flow_mask_img = mask_img.astype(np.uint8)

flow_masks.append(Image.fromarray(flow_mask_img
* 255))

    # Dilatasi umum
    if mask_dilates > 0:
        mask_img =
scipy.ndimage.binary_dilation(mask_img,
iterations=mask_dilates).astype(np.uint8)
    else:
        mask_img = mask_img.astype(np.uint8)

masks_dilated.append(Image.fromarray(mask_img *
255))

if len(masks_img) == 1:
    flow_masks *= length

```

```

    masks_dilated *= length

    return flow_masks, masks_dilated

```

Pseudocode 4.4 Binary dilatation untuk memperluas masking

Proses pembacaan *mask* mendukung *single image* dan *multiple masks*, dengan dilatasi yang berbeda.

```

def extrapolation(video_ori, scale):
    nFrame = len(video_ori)
    imgW, imgH = video_ori[0].size

    # Hitung ukuran baru dengan skala (misal: [1.2, 1.5])
    imgH_extr = int(scale[0] * imgH)
    imgW_extr = int(scale[1] * imgW)
    imgH_extr -= imgH_extr % 8
    imgW_extr -= imgW_extr % 8

    H_start = int((imgH_extr - imgH) / 2)
    W_start = int((imgW_extr - imgW) / 2)

    frames = []
    for v in video_ori:
        frame = np.zeros((imgH_extr, imgW_extr, 3),
dtype=np.uint8)
        frame[H_start: H_start + imgH, W_start: W_start
+ imgW, :] = v
        frames.append(Image.fromarray(frame))

    masks_dilated = []
    flow_masks = []

    dilate_h = 4 if H_start > 10 else 0
    dilate_w = 4 if W_start > 10 else 0

```

```

    mask = np.ones((imgH_extr, imgW_extr),
dtype=np.uint8)
    mask[H_start + dilate_h: H_start + imgH - dilate_h,
        W_start + dilate_w: W_start + imgW - dilate_w]
= 0
    flow_masks.append(Image.fromarray(mask * 255))

    mask[H_start: H_start + imgH, W_start: W_start +
imgW] = 0
    masks_dilated.append(Image.fromarray(mask *
255))

    flow_masks *= nFrame
    masks_dilated *= nFrame

    return frames, flow_masks, masks_dilated,
(imgW_extr, imgH_extr)

```

Pseudocode 4.5 extrapolation ProPainter

Fungsi *extrapolation* menangani kasus video *outpainting* dengan perluasan canvas dan pembuatan *mask* untuk wilayah yang akan di-*extend*. Berikut ini merupakan *pipeline* utama ProPainter mengintegrasikan semua komponen dalam alur kerja yang komprehensif, dari *preprocessing* hingga output final.

4.2.4. Implementasi Raft

RAFT (Recurrent All-Pairs Field Transforms) berperan penting dalam estimasi optical flow antar frame yang akurat. Implementasi ini menghasilkan visualisasi flow dan data numerik.

```

def demo(raft_opts, img1_path, img2_path, output_dir):
    # Load the RAFT model
    model = torch.nn.DataParallel(RAFT(raft_opts))
    model.load_state_dict(torch.load(raft_opts.model))
    model = model.module
    model.to(DEVICE)
    model.eval()

    os.makedirs(output_dir, exist_ok=True) # Ensure the
    output directory exists

    with torch.no_grad():
        # Load images
        image1 = load_image(imfile1)
        image2 = load_image(imfile2)

        # Pad images
        padder = InputPadder(image1.shape)
        image1, image2 = padder.pad(image1, image2)

        # Compute optical flow
        flow_low, flow_up = model(image1, image2,
            iters=20, test_mode=True)

        # Generate filename for saving the flow image
        flow_filename = os.path.join(output_dir,
            f'flow_{os.path.basename(imfile1)}_{os.path.basename(
            imfile2)}.jpg')
        npy_filename = os.path.join(output_dir,
            f'flow_{os.path.basename(imfile1)}_{os.path.basename(
            imfile2)}.npz')
        # Visualize and save the flow image
        viz(flow_up, save_path=flow_filename,
            npy_path=npy_filename)

```

Pseudocode 4.6 Fungsi demo pada RAFT untuk memproses dua frame menjadi optical flow

Fungsi demo menangani komputasi *optical flow* antara dua frame dengan output berupa visualisasi dan data numerik untuk pemrosesan selanjutnya. Fungsi visualisasi mengkonversi *tensor flow* menjadi representasi visual yang intuitif dan menyimpan data numerik untuk analisis lebih lanjut. Implementasi RAFT ini menyediakan pondasi yang solid untuk estimasi pergerakan yang akurat.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap sistem yang telah dikembangkan, meliputi metodologi pengujian, kriteria evaluasi, dan analisis hasil untuk memastikan kualitas dan performa sistem.

5.1 Tujuan Pengujian

Pengujian dilakukan untuk memvalidasi integrasi ketiga komponen utama (SerNet-Former, ProPainter, dan RAFT) dan mengidentifikasi kekurangan yang perlu diperbaiki.

5.2 Kriteria Pengujian

Penilaian keberhasilan pengujian dilakukan berdasarkan pencapaian hasil yang telah ditentukan, sebagai berikut:

- a. Fungsi berjalan tanpa bug.
- b. Output *optical flow* memiliki kualitas yang baik.
- c. Identifikasi area yang memerlukan optimasi untuk pengembangan lebih lanjut

5.3 Skenario Pengujian

Adapun Skenario Pengujian dilakukan dengan beberapa skenario, seperti berikut

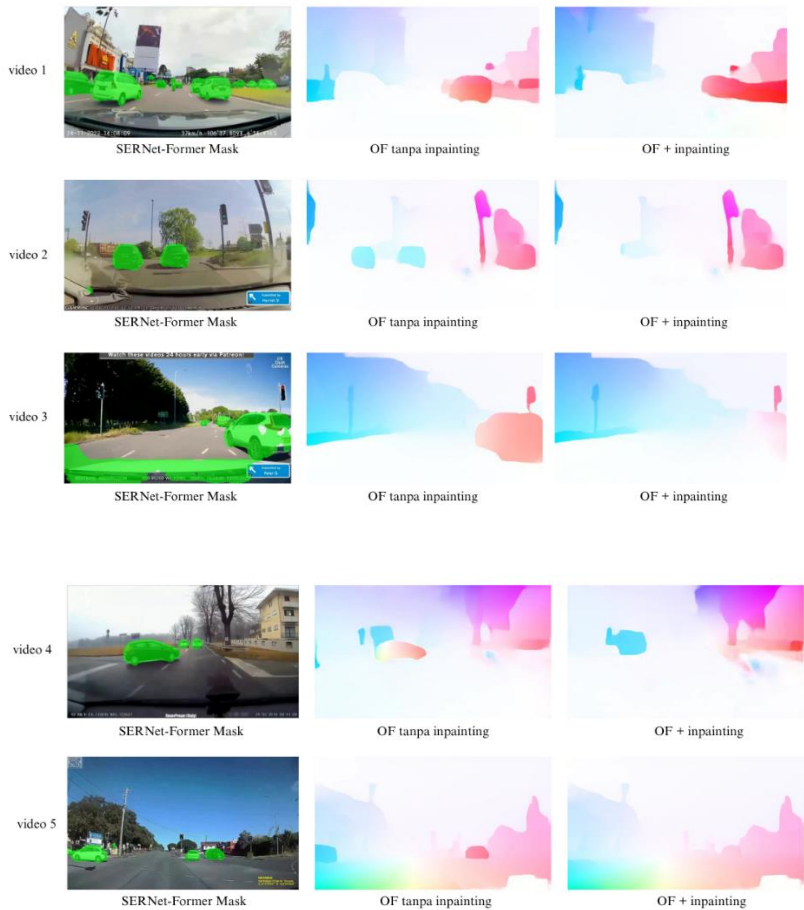
- a. Analisis flow pattern pada video dengan objek diam.
- b. Deteksi *inpainting* di area *boundary frame*.
- c. Verifikasi keberhasilan *inpainting* dalam menghapus objek bergerak.

5.4 Hasil Pengujian

Berdasarkan pengujian yang dilakukan terhadap sistem terintegrasi yang menggabungkan SerNet-Former, ProPainter, dan RAFT, diperoleh hasil evaluasi yang menyeluruh. Secara keseluruhan, sistem berhasil menjalankan fungsi utamanya tanpa mengalami bug atau error fatal selama proses pengujian. Integrasi ketiga komponen utama menunjukkan kompatibilitas yang baik.

Hasil komponen masking menggunakan SerNet-Former menunjukkan kinerja yang baik dalam deteksi dan segmentasi objek target yakni mobil. Sistem mampu mengidentifikasi objek dengan akurasi tinggi pada kondisi normal dan menghasilkan mask yang berkualitas baik. Namun, ditemukan beberapa keterbatasan pada area boundary atau batas objek, dimana hasil segmentasi menjadi kurang optimal. Terdapat fenomena gap kecil pada mask yang dihasilkan, terutama pada objek dengan tepi yang kompleks atau tidak jelas. Hal ini mengindikasikan perlunya optimasi lebih lanjut pada algoritma post-processing untuk memperbaiki kualitas mask di area boundary.

Komponen *inpainting* menggunakan ProPainter menunjukkan kemampuan yang baik dalam merekonstruksi background dan menghilangkan objek yang tidak diinginkan dari video. Pada Gambar 5.1 Proses *inpainting* berjalan dengan lancar pada kondisi standar dan mampu mempertahankan konsistensi temporal yang memuaskan. Namun, performa sistem mengalami penurunan ketika berhadapan dengan video yang memiliki objek bergerak dengan kecepatan tinggi. Selain itu, kehadiran bayangan objek juga menyebabkan hasil *inpainting* menjadi kurang optimal, dimana sistem kesulitan dalam membedakan antara objek utama dan bayangan yang ditimbulkannya.



Gambar 5.1 Hasil pengujian sistem untuk implementasi inpainting pada optical flow

Berdasarkan hasil pengujian yang telah dilakukan, sistem berhasil memenuhi kriteria pengujian yang telah ditetapkan. Sistem berjalan tanpa bug, *output optical flow* memiliki kualitas yang baik, dan area-area yang disegmentasi dapat terhapus dengan

baik. Meskipun demikian, beberapa keterbatasan yang ditemukan memerlukan perhatian khusus untuk pengembangan lebih lanjut.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dalam implementasi *inpainting* untuk optical flow pada data *dashcam* kendaraan sebagai fitur *extraction* dengan mengintegrasikan SerNet-Former, ProPainter, dan RAFT, dapat ditarik beberapa kesimpulan penting. Sistem yang dikembangkan berhasil menggabungkan tiga komponen utama dalam satu *pipeline* yang terintegrasi dan berfungsi dengan baik pada data video *dashcam*. SerNet-Former mampu melakukan deteksi dan segmentasi objek dalam video *dashcam* dengan hasil yang memuaskan, ProPainter berhasil melakukan proses *inpainting* untuk menghilangkan objek yang dapat mengganggu analisis optical flow, dan RAFT memberikan estimasi optical flow yang lebih bersih dan konsisten setelah proses *inpainting* dilakukan.

Secara keseluruhan, penelitian ini dapat dikembangkan menjadi aplikasi estimasi kecepatan dalam sistem transportasi otonom. Sistem yang dikembangkan dapat menjadi fondasi untuk pengembangan sistem Advanced Driver Assistance Systems (ADAS) atau *autonomous vehicle* yang memerlukan estimasi kecepatan kendaraan secara akurat dan real-time. Dengan perbaikan pada area-area yang telah diidentifikasi, sistem ini memiliki potensi untuk diimplementasikan dalam berbagai aplikasi transportasi.

6.2 Saran

Penelitian ini menunjukkan bahwa implementasi inpainting pada *optical flow* dari data dashcam masih memiliki beberapa tantangan, terutama pada area tepi *mask*, kondisi visual kompleks, dan kebutuhan pemrosesan real-time. Beberapa saran untuk pengembangan lebih lanjut dari hasil kerja praktik ini adalah sebagai berikut:

1. Masalah Tepi Masking:
Perlu teknik post-processing untuk menghaluskan tepi mask agar hasil inpainting lebih akurat.
2. Robust terhadap Kondisi Dashcam:
Kembangkan algoritma yang mampu menangani objek cepat, bayangan, refleksi, dan cahaya tidak stabil.
3. Optimasi untuk Real-Time:
Gunakan proses paralel dan optimasi algoritma agar sistem bisa berjalan cepat dan responsif saat digunakan langsung.
4. Dataset Dashcam yang Beragam:
Evaluasi harus menggunakan dataset dengan kondisi cuaca, waktu, dan jalan berbeda untuk meningkatkan akurasi hasil.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] H. Caesar *et al.*, “nuScenes: A multimodal dataset for autonomous driving,” Mar. 2019, [Online]. Available: <http://arxiv.org/abs/1903.11027>
- [2] L. Capito, K. Redmill, and U. Ozguner, “Optical Flow based Visual Potential Field for Autonomous Driving,” May 2020, [Online]. Available: <http://arxiv.org/abs/2006.00168>
- [3] A. Alfano, L. Maiano, L. Papa, and I. Amerini, “Estimating optical flow: A comprehensive review of the state of the art,” *Computer Vision and Image Understanding*, vol. 249, p. 104160, Dec. 2024, doi: 10.1016/j.cviu.2024.104160.
- [4] Y. Wang, C. Wang, H. Wu, and P. Chen, “An improved Deeplabv3+ semantic segmentation algorithm with multiple loss constraints,” *PLoS One*, vol. 17, no. 1 January, Jan. 2022, doi: 10.1371/journal.pone.0261582.
- [5] S. Erisen, “SERNet-Former: Semantic Segmentation by Efficient Residual Network with Attention-Boosting Gates and Attention-Fusion Networks,” Jan. 2024, doi: 10.48550/arXiv.2401.15741.
- [6] S. Zhou, C. Li, K. C. K. Chan, and C. C. Loy, “ProPainter: Improving Propagation and Transformer for Video Inpainting,” Sep. 2023, [Online]. Available: <http://arxiv.org/abs/2309.03897>
- [7] Z. Teed and J. Deng, “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow,” Mar. 2020, [Online]. Available: <http://arxiv.org/abs/2003.12039>

BIODATA PENULIS

Nama : Akmal Nafis
Tempat, Tanggal Lahir : Bangkalan, 25 Juli 2003
Jenis Kelamin : Laki-laki
Telepon : +6282139259137
Email : akmalnafis25@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 8 (Delapan)