

Perbandingan Algoritma *Greedy* dengan Algoritma *Dynamic Programming* dalam Stabilisasi Video Digital

Muhammad Izzat Jundy - 13523092

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: izzatjundy04@gmail.com, 13523092@std.stei.itb.ac.id

Abstrak—Goyangan pada video yang direkam menggunakan perangkat portabel seperti *smartphone* menurunkan kualitas visual dan kenyamanan menonton. Stabilisasi Video Digital (SVD) menawarkan solusi pascaproduksi untuk mengatasi masalah ini. Makalah ini menyajikan analisis perbandingan kinerja antara algoritma *greedy* yang bersifat lokal dan reaktif dengan pendekatan *dynamic programming* (DP) yang bersifat global untuk tugas SVD. Kedua algoritma diimplementasikan dan diuji pada dua skenario video—goyangan frekuensi tinggi dan panning halus—dengan metrik evaluasi meliputi waktu eksekusi, persentase kehilangan area gambar akibat pemangkasan, dan analisis kualitas visual. Hasil eksperimen menunjukkan bahwa algoritma *greedy* secara komputasi lebih cepat, tetapi gagal total dalam menangani gerakan kamera yang disengaja dan menghasilkan *drifting* yang parah. Sebaliknya, pendekatan DP, meskipun secara signifikan lebih lambat, secara konsisten menghasilkan video yang jauh lebih mulus, alami, dan andal di kedua skenario pengujian. Penelitian ini menyimpulkan bahwa untuk hasil stabilisasi berkualitas tinggi, pendekatan global seperti DP adalah strategi yang superior, sedangkan pendekatan *greedy* murni terbukti tidak cukup sangkil untuk aplikasi praktis.

Kata kunci—stabilisasi video digital; *dynamic programming*; algoritma *greedy*; analisis kinerja; pengolahan citra digital

I. PENDAHULUAN

Meningkatnya penggunaan perangkat perekam video digital seperti kamera pada *smartphone* dan kamera *mirrorless*, memunculkan permasalahan baru berupa kurang stabilnya visual dari video-video yang dihasilkan oleh perangkat-perangkat tersebut. Video yang kurang stabil, atau dalam dunia profesional biasa disebut dengan *shaky* (goyang), relatif kurang nyaman untuk ditonton. Oleh sebab itu, diperlukan solusi berupa stabilisasi video digital (SVD) yang dapat menstabilkan video-video tersebut agar terhindar dari *shaky*.

Stabilisasi video digital (SVD) dapat direalisasikan menggunakan berbagai algoritma. Dua algoritma yang menarik adalah *greedy*, yaitu algoritma yang mencari solusi optimal lokal, dan *dynamic programming*, yaitu algoritma yang mencari solusi optimal global. Dalam penelitian ini, akan ditelisik bagaimana perbandingan penggunaan dua algoritma tersebut dalam SVD. Perbandingan akan dinilai berdasarkan bagaimana kualitas stabilisasi video yang dihasilkan dan seberapa tingkat

efisiensi waktu komputasi. Kontribusi utama dari penelitian ini akan berfokus pada perbandingan kuantitatif dan kualitatif dari kedua algoritma tersebut dalam penerapannya pada SVD.

II. DASAR TEORI

A. Stabilisasi Video Digital

Stabilisasi video digital (SVD) adalah sekumpulan teknik pengolahan citra yang bertujuan untuk mengurangi atau menghilangkan goyangan (*shakiness*) yang tidak diinginkan dari sebuah rekaman video. Goyangan ini umumnya disebabkan oleh pergerakan tangan saat memegang kamera, getaran lokasi pemegang kamera, atau faktor lingkungan lainnya. Tujuan utama dari SVD adalah untuk menghasilkan video yang lebih halus dan nyaman untuk ditonton tanpa memerlukan perangkat keras stabilisasi khusus seperti gimbal. Proses SVD secara umum dapat dibagi menjadi dua fase utama, yaitu estimasi gerakan dan kompensasi gerakan.

1) Estimasi Gerakan (*Motion Estimation*)

Fase pertama adalah mengidentifikasi dan mengukur gerakan kamera yang tidak diinginkan antar dua frame video yang bersebelahan. Salah satu pendekatan yang paling umum adalah menggunakan metode berbasis fitur (*feature-based*). Proses tersebut dilakukan dengan langkah-langkah sebagai berikut.

a) Deteksi Fitur

Algoritma mendeteksi titik-titik yang tampak unik (fitur) pada frame pertama, seperti sudut-sudut objek atau area dengan tekstur yang kaya. Contoh algoritma pendeteksi fitur adalah *Good Features to Track* (GFTT).

b) Pelacakan Fitur (*Feature Tracking*)

Kemudian, sistem melacak posisi dari fitur-fitur tersebut pada frame berikutnya. Algoritma seperti *Lucas-Kanade Optical Flow* sering digunakan pada tahap ini karena efisiensinya.

c) Kalkulasi Transformasi

Berdasarkan pergeseran posisi dari sekumpulan fitur yang berhasil dilacak dari tahap sebelumnya, sebuah model

transformasi matematis dihitung. Model yang banyak digunakan adalah transformasi *affine*, yang dapat merepresentasikan tiga jenis gerakan utama kamera, yaitu translasi pada sumbu X, translasi pada sumbu Y, dan rotasi pada sumbu Z. Hasil dari fase ini adalah serangkaian data transformasi yang merepresentasikan "jalur goyangan" dari video.

2) Kompensasi Gerakan (Motion Compensation)

Setelah mendapatkan "jalur goyangan" dari video, dilakukan penghalusan dengan mengoreksi "jalur goyangan" tersebut. Ini dilakukan dengan menerapkan transformasi geometris kebalikan pada setiap frame. Operasi warpAffine pada pustaka seperti OpenCV digunakan untuk menerapkan matriks transformasi (yang berisi informasi translasi dan rotasi) pada sebuah gambar. Proses ini secara efektif menggeser dan memutar setiap *frame* untuk meniadakan goyangan sehingga video dapat terlihat lebih stabil. Meskipun proses ini dapat memunculkan tepian hitam di sekitar *frame*, hal ini dapat diatasi dengan melakukan pemangkasan gambar atau *zoom*.

B. Strategi Penghalusan Jalur Gerakan

Inti dari perbandingan dalam penelitian ini terletak pada bagian ini, yaitu tentang bagaimana cara mengubah "jalur goyang" menjadi "jalur mulus". Dua strategi algoritma yang dianalisis adalah *greedy* dan *dynamic programming*.

1) Greedy

Algoritma *greedy* adalah algoritma yang membuat pilihan berdasarkan yang tampaknya terbaik pada setiap langkah (optimal lokal), tanpa mempertimbangkan gambaran keseluruhan. Dalam konteks SVD, algoritma *greedy* yang murni akan beroperasi sebagai berikut.

- Tanpa membaca "jalur goyangan" terlebih dahulu.
- Pada setiap *frame* f , algoritma hanya melihat perubahan posisi yang terjadi pada *frame* $f - 1$.
- Keputusan koreksi yang diambil hanya bertujuan untuk membuat transisi dari *frame* $f - 1$ ke *frame* f menjadi semulus mungkin.
- Algoritma ini tidak peduli dengan posisi *frame-frame* sebelum $f - 1$.

Dengan hanya mempertimbangkan *frame* yang terdekat saja, algoritma *greedy* sangat cepat secara komputasi. Namun, risikonya, algoritma ini memiliki kelemahan fatal, yaitu rentan terhadap akumulasi galat. Kesalahan pembuatan keputusan koreksi yang kecil pada setiap *frame* bisa saja menumpuk dan terus menumpuk seiring dengan pergantian *frame*, menyebabkan posisi pusat gambar secara perlahan bergeser dari posisi seharusnya.

2) Dynamic Programming

Algoritma *dynamic programming* (DP) adalah algoritma yang digunakan untuk menyelesaikan masalah optimisasi kompleks dengan memecah masalah-masalah tersebut menjadi submasalah-submasalah yang lebih sederhana dan saling tumpang tindih (*overlapping subproblems*). Prinsip dasarnya

adalah *optimal substructure*, solusi optimal untuk masalah utama dapat dibangun dari solusi optimal submasalahnya.

Secara teoretis, stabilisasi video dapat dimodelkan sebagai masalah pencarian jalur terpendek dalam sebuah graf. Setiap *frame* memiliki beberapa kemungkinan keadaan yang dianggap stabil, dan DP bertugas untuk mencari rangkaian keadaan (jalur) dari awal hingga akhir video yang memiliki total "biaya goyangan" paling minimal. Dengan mengevaluasi semua kemungkinan jalur secara efisien, DP mampu menemukan solusi optimal global.

Dalam implementasi praktis, algoritma ini seringkali disederhanakan (diaproksimasi) untuk efisiensi komputasi. Salah satu teknik yang banyak digunakan adalah dengan menghitung lintasan kumulatif ("jalur goyangan") dari seluruh gerakan kamera terlebih dahulu. Kemudian, lintasan ini dihaluskan menggunakan *filter* seperti *moving average*. Proses penghalusan lintasan global ini menangkap esensi dari DP—yaitu mempertimbangkan keseluruhan lintasan untuk membuat keputusan—tanpa harus membangun graf keadaan yang masif secara eksplisit. Pendekatan inilah yang akan diimplementasikan dan dianalisis dalam penelitian ini sebagai representasi dari strategi DP.

III. METODOLOGI PENELITIAN

Bab ini merincikan metodologi yang digunakan untuk mengimplementasikan dan menguji kedua strategi algoritma stabilisasi video. Perancangan sistem sengaja dibedakan agar secara jelas merefleksikan perbedaan filosofis antara pendekatan lokal yang reaktif (*greedy*) dan pendekatan global yang proaktif (*dynamic programming*).

A. Alur Kerja Sistem

Proses stabilisasi video dalam penelitian ini dirancang sebagai sebuah *pipeline* pemrosesan data yang terstruktur. Tujuannya adalah untuk mengubah data video mentah yang tidak stabil menjadi video keluaran yang mulus secara visual. Alur kerja ini dibagi menjadi tiga tahapan utama sebagai berikut.

1) Estimasi Gerakan

Tahap pertama adalah analisis video secara menyeluruh untuk membangun pemahaman kuantitatif mengenai gerakan kamera. Sistem akan "menonton" video *frame* demi *frame*, mengidentifikasi titik-titik fitur yang konsisten, dan melacak pergeserannya. Hasil dari tahap ini bukanlah video, melainkan sebuah data sekuensial yang merepresentasikan "jalur goyangan" mentah dari kamera. "Jalur goyangan" ini mencatat setiap pergeseran dan rotasi yang terjadi antar dua *frame* yang bersebelahan.

2) Penghalusan Jalur

Data "jalur goyangan" ini kemudian menjadi masukan bagi tahap kedua, yang merupakan inti dari perbandingan algoritma dalam penelitian ini. Sistem akan mengolah jejak goyangan tersebut menggunakan salah satu dari dua strategi algoritma (*greedy* atau DP) untuk menghasilkan serangkaian instruksi koreksi yang baru dan lebih mulus.

3) Aplikasi Transformasi

Tahap terakhir adalah sintesis, yaitu pembuatan video final. Sistem akan membaca kembali video asli frame demi frame, dan untuk setiap frame, sistem akan menerapkan instruksi koreksi yang sesuai dengan yang telah dihasilkan pada tahap kedua.

B. Implementasi Strategi Penghalusan

Perbedaan antara kedua algoritma yang digunakan terletak pada bagaimana cara memandang dan menggunakan data gerakan ("jalur goyangan") untuk membuat keputusan.

1) Greedy (Lokal-Reaktif)

Implementasi algoritma *greedy* dirancang untuk merepresentasikan konsep dasarnya yang bekerja secara lokal dan tanpa mempertimbangkan keseluruhan jalur. Logika intinya tidak memerlukan pengetahuan tentang seluruh "jalur goyangan", melainkan bekerja secara reaktif *frame* demi *frame*.

Prosesnya bekerja sebagai berikut: untuk setiap *frame* f , sistem pertama-tama menghitung data goyangan sesaat yang terjadi dari *frame* sebelumnya ($f - 1$). Berdasarkan hanya pada data sesaat ini, sebuah keputusan langsung dibuat dengan membandingkannya terhadap ambang batas (*threshold*) yang telah ditentukan. Apabila magnitudo gerakan melampaui batas toleransi, maka perubahan posisi antar kedua *frame* tersebut dianggap sebagai getaran yang tidak diinginkan dan sebuah instruksi koreksi akan dibuat untuk menetralkannya. Sebaliknya, jika gerakan dianggap minor, sistem akan mengabaikannya. Keputusan koreksi untuk sebuah *frame* diakumulasikan dari koreksi *frame* sebelumnya, menciptakan sebuah rantai dependensi lokal.

2) Dynamic Programming (Global-Proaktif)

Strategi algoritma *dynamic programming* (DP) pada penelitian ini menghasilkan solusi terbaik yang bersifat global. Langkah pertama yang dilakukan dalam implementasi algoritma ini adalah dengan menganalisis keseluruhan video untuk memetakan "jalur goyangan" dari awal video sampai akhir. Representasi konkretnya adalah untuk setiap *frame*, terdapat translasi pada sumbu X, translasi pada sumbu Y, dan rotasi pada sumbu Z.

Setelah mendapatkan gambaran jelas lintasan yang ada, proses penghalusan dilakukan. Tujuan tahap proses ini adalah untuk merencanakan dan "menggambar" satu "jalur ideal" baru yang paling mulus secara keseluruhan di atas lintasan yang goyang tersebut. Setelah "jalur ideal" ini selesai direncanakan, instruksi koreksi untuk setiap *frame* dihitung berdasarkan seberapa jauh posisi aslinya di "jalur goyang" menyimpang dari posisi seharusnya di "jalur ideal". Dengan demikian, setiap keputusan koreksi dibuat berdasarkan konteks gerakan video secara keseluruhan.

C. Skenario Pengujian

Untuk membandingkan kedua metode secara adil, dilakukan pengujian dengan skenario yang terkontrol dengan detail sebagai berikut.

1) Dataset Video

Pengujian menggunakan dua klip video berdurasi sekitar 10 detik dengan resolusi 1920x1080 piksel dan 30 FPS. Video A berisi rekaman *still* (diam) dengan goyangan berfrekuensi tinggi. Video B berisi gerakan kamera *panning* yang disengaja dengan goyangan minor.

2) Metrik Evaluasi

Aspek yang dibandingkan antara hasil dari algoritma *greedy* dan *dynamic programming* adalah sebagai berikut.

a) Waktu Eksekusi

Diukur dalam satuan milidetik, mencakup keseluruhan proses dari tepat setelah menerima video hingga sebelum video mulai dibuat berdasarkan "jalur ideal".

b) Gambar Hilang dari Hasil Pemangkasan Gambar yang Diperlukan

Pada proses pergeseran setiap *frame* mengikuti "jalur ideal" yang sudah ditentukan, akan terdapat bagian di kanan, kiri, atas, atau bawah gambar yang menjadi hilang, biasanya menjadi hitam. Untuk mengatasi permasalahan tersebut, diperlukan pemangkasan gambar dengan cara *zoom* menyesuaikan dengan jarak paling panjang antara tepi *frame* dengan gambar asli dari keseluruhan video hasil stabilisasi (panjang pemangkasan minimal). Algoritma pemangkasan gambar tidak akan dibahas dalam makalah ini. Namun, variabel yang digunakan sebagai ukuran pemangkasan itu lah yang akan digunakan sebagai parameter.

Akan diperoleh panjang pemangkasan minimal untuk horizontal dan vertikal dari stabilisasi yang dilakukan, yang kemudian disesuaikan dengan rasio *frame* (dalam penelitian ini 16:9) dengan *pseudocode* sebagai berikut.

```
IF crop_vertikal * 16 < crop_horizontal * 9
  THEN
    crop_horizontal = ceil(crop_horizontal/16) * 16
    crop_vertikal = crop_horizontal * 9 / 16
ELSE
  THEN
    crop_vertikal = ceil(crop_vertikal/9) * 9
    crop_horizontal = crop_vertikal * 16 / 9
```

Dari perhitungan tersebut, diperoleh besar pemangkasan yang diperlukan. Kemudian, dilakukan perhitungan sebagai berikut.

$$\text{Area Hilang (\%)} = \left(1 - \frac{\text{Area Hasil Crop}}{\text{Area Gambar Asli}}\right) \times 100\%$$

Dengan $\text{Area Hasil Crop} = (1920 - 2 \times \text{crop}_{\text{horizontal}}) \times (1080 - 2 \times \text{crop}_{\text{vertikal}})$ dan $\text{Area Gambar Asli} = 1920 \times 1080$.

c) Kualitas Visual

Analisis kualitatif yang dilakukan dengan membandingkan video hasil dari kedua metode secara berdampingan. Fokus analisis adalah pada kehalusan gerakan, ada atau tidaknya fenomena *drifting* (pergeseran pusat gambar), dan bagaimana setiap algoritma menangani gerakan yang disengaja.

3) Lingkungan Pengujian

Seluruh eksperimen dijalankan pada sistem dengan spesifikasi prosesor Intel Core Ultra 5 125H, RAM 16 GB,

menggunakan Python 3.12.10 dan pustaka OpenCV versi 4.9.0.80.

IV. HASIL DAN PEMBAHASAN

Bab ini menyajikan dan menganalisis hasil dari eksperimen yang telah dilakukan sesuai dengan metodologi yang diuraikan pada Bab III. Sebelum menyajikan hasil, penting untuk melakukan pemilihan parameter yang digunakan dalam pengujian. Parameter-parameter ini dipilih melalui serangkaian eksperimen awal untuk menemukan keseimbangan yang representatif antara kualitas stabilisasi dan area hilang yang mungkin timbul.

Parameter pengujian yang ditetapkan adalah *smoothing radius* = 30 untuk algoritma *dynamic programming* dan *translation threshold* = 1.5 untuk metode *greedy*.

- Pemilihan *smoothing radius* = 30: Pada video 30 FPS, radius 30 berarti algoritma mempertimbangkan sebuah *frame-frame* sekitar satu detik ke masa lalu dan satu detik ke masa depan untuk menghaluskan lintasan pada setiap *frame*. Ukuran jendela dua detik ini dianggap cukup lebar untuk secara efektif meredam goyangan berfrekuensi tinggi (getaran tangan atau lainnya), tetapi cukup sempit untuk tidak sepenuhnya menghilangkan gerakan kamera yang disengaja (seperti *panning*).
- Pemilihan *translation threshold* = 1.5: Nilai 1.5 piksel dipilih sebagai ambang batas untuk pergeseran. Gerakan di bawah 1.5 piksel antar frame seringkali tidak terlalu terlihat oleh mata manusia dan dapat dianggap sebagai noise sensor atau getaran mikro. Dengan menetapkan threshold ini, algoritma *greedy* diinstruksikan untuk mengabaikan goyangan minor dan hanya bereaksi pada guncangan yang lebih signifikan, mencegah koreksi berlebihan yang justru dapat menciptakan getaran baru.

A. Hasil Kuantitatif

Hasil dari kedua video dan kedua algoritma menghasilkan hasil kuantitatif yang cukup berbeda.

1) Pengujian pada Video A (Goyangan Tinggi)

Video A, video *still* (diam) berdurasi 10 detik yang frekuensi goyangan tinggi, digunakan untuk menguji kemampuan algoritma dalam menstabilkan video dengan frekuensi goyangan yang tinggi. Berikut hasil dari pengujian pada Video A.

TABLE I. TABEL HASIL PERCOBAAN PADA VIDEO A

Metode	Parameter	Waktu Eksekusi (milidetik)	Area Hilang (%)
<i>Greedy</i>	Translation threshold = 1.5	20.53	24.89%
<i>Dynamic Programming</i>	Smooth radius = 30	13201.20	12.89%

2) Pengujian pada Video B (Panning Halus)

Video B, video *panning* berdurasi 7 detik yang frekuensi goyagannya rendah, digunakan untuk menguji kemampuan algoritma dalam menstabilkan video dengan goyangan yang disengaja. Berikut hasil dari pengujian pada Video B.

TABLE II. TABEL HASIL PERCOBAAN PADA VIDEO B

Metode	Parameter	Waktu Eksekusi (milidetik)	Area Hilang (%)
<i>Greedy</i>	Translation threshold = 1.5	14.03	73.31%
<i>Dynamic Programming</i>	Smooth radius = 30	9877.12	12.89%

Dari kedua sampel, dapat dilihat bahwa algoritma *greedy*, meskipun lebih cepat, menghasilkan area hilang yang lebih besar dari DP. Bahkan, pada video *panning*, dia menghilangkan sebesar 73.31%, sekitar 6 kali dari DP.

B. Hasil Kualitatif

Pada Video A, hasil dari algoritma DP jauh lebih terasa halus dibanding hasil dari algoritma, juga video aslinya. Meskipun tidak sepenuhnya terasa “mengambang”, tetapi hasil dari algoritma DP cukup memuaskan dan layak tonton. Video *greedy* masih sangat terasa goyagannya. Dibandingkan video aslinya, hasilnya tidak signifikan berbeda.

Pada Video B, hasil dari algoritma DP dan *greedy* sama-sama terasa halus. Akan tetapi, hasil dari algoritma *greedy* tidak mengamini gerakan yang disengaja. Lebih lanjut, *panning* yang dilakukan ke samping kanan di video, tidak lantas membuat objek yang ada bergeser ke kiri (berlawanan arah *panning* kamera), melainkan tetap diam, dan akhirnya video terpotong sebanyak *panning*-an dari kamera. Sementara hasil dari algoritma DP, mengikuti gerakan kamera yang disengaja dan menstabilkan goyangan minor, menyebabkan videonya sangat nyaman untuk ditonton.

C. Pembahasan

Hasil kuantitatif dan kualitatif yang disajikan menunjukkan perbedaan fundamental dalam cara kedua algoritma memandang dan menyelesaikan masalah stabilisasi, yang mengarah pada trade-off yang jelas antara efisiensi dan kualitas.

1) Analisis Efisiensi Komputasi vs. Kualitas Global

Data waktu eksekusi secara konsisten menunjukkan bahwa algoritma *greedy* memiliki kinerja ribuan kali lebih cepat daripada pendekatan DP. Hal ini disebabkan oleh perbedaan kompleksitas fundamental. Algoritma *greedy* hanya melakukan satu kali kalkulasi lokal untuk setiap frame ($O(N)$), sementara pendekatan DP memerlukan beberapa pass atas seluruh data: estimasi gerakan global, kalkulasi lintasan kumulatif, penghalusan lintasan, dan akhirnya aplikasi transformasi.

Namun, kecepatan komputasi *greedy* harus dibayar mahal dengan kualitas hasil yang rendah. Seperti yang ditunjukkan pada hasil kualitatif Video A, kemampuannya untuk meredam goyangan sangat terbatas. Sebaliknya, waktu proses DP yang lama diinvestasikan untuk perencanaan global, yang terbukti menghasilkan video dengan tingkat kehalusan yang jauh lebih superior.

2) Kegagalan Algoritma Lokal pada Gerakan Disengaja

Hasil pada Video B adalah demonstrasi klasik dari kelemahan inheren sebuah algoritma lokal. Algoritma *greedy*, yang tidak memiliki "memori" atau pandangan global, menginterpretasikan gerakan panning yang konsisten sebagai serangkaian "goyangan" besar yang harus dilawan. Dengan terus-menerus menerapkan koreksi yang berlawanan dengan arah gerak kamera, ia secara efektif "mengunci" latar belakang gambar pada posisi awalnya. Akibatnya adalah kehilangan area gambar yang katastropis (73.31%), di mana sebagian besar frame terpotong karena kamera terus bergerak menjauh dari area yang "dikunci" tersebut.

Sebaliknya, DP mampu mengenali panning sebagai sebuah lintasan yang mulus dan berbiaya rendah. Ia tidak melawannya, melainkan menghaluskan getaran-getaran kecil di sepanjang lintasan tersebut, sehingga berhasil mempertahankan intensi sinematik dari gerakan kamera.

3) Interpretasi Metrik Area Hilang

Hasil yang menarik terlihat pada Video A, di mana *greedy* justru kehilangan lebih banyak area gambar (24.89%) dibandingkan DP (12.89%). Ini mungkin tampak berlawanan dengan intuisi, namun dapat dijelaskan oleh sifat reaktif *greedy*. Pada video dengan goyangan frekuensi tinggi, *greedy* terus-menerus melakukan koreksi bolak-balik secara ekstrem untuk setiap getaran yang melebihi threshold. Akumulasi dari koreksi-koreksi yang tidak terkoordinasi ini ternyata menghasilkan rentang gerakan total yang lebih liar daripada jalur optimal yang lebih terpusat yang ditemukan oleh DP. DP, meskipun harus melakukan zoom untuk mengakomodasi jalur mulusnya, menemukan jalur yang lebih "konservatif" secara keseluruhan.

4) Implikasi Praktis

Berdasarkan analisis ini, pendekatan *greedy* murni terbukti tidak cocok untuk stabilisasi video yang andal karena ketidakmampuannya membedakan goyangan dan gerakan disengaja, serta masalah drifting yang parah. Pendekatan DP, meskipun lambat, merupakan strategi yang jauh lebih superior dan robust untuk menghasilkan video berkualitas profesional yang stabil dan alami.

V. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan dan menganalisis perbandingan dua strategi algoritma yang berbeda secara fundamental, yaitu *greedy* dan *dynamic programming*, untuk tugas stabilisasi video digital. Hasil eksperimen secara keseluruhan menunjukkan adanya trade-off yang mendasar antara efisiensi komputasi dan kualitas hasil akhir. Algoritma *greedy*, dengan sifatnya yang lokal dan reaktif, terbukti lebih cepat dalam pemrosesan, tetapi gagal menghasilkan video yang andal. Kelemahan utamanya adalah ketidakmampuannya membedakan antara goyangan minor dan gerakan kamera yang disengaja, yang menyebabkan *drifting* yang parah pada skenario *panning* dan kualitas peredaman goyangan yang tidak signifikan pada video dengan getaran tinggi. Sebaliknya,

pendekatan *dynamic programming* (DP) yang bersifat global, meskipun secara signifikan lebih lambat, secara konsisten mampu menghasilkan video yang jauh lebih mulus, mempertahankan intensi perekam pada gerakan yang disengaja, dan secara efektif menekan berbagai jenis goyangan. Bahkan dalam beberapa kasus, pendekatan DP yang lebih terencana menghasilkan kehilangan area gambar yang lebih kecil dibandingkan akumulasi galat dari koreksi reaktif *greedy*. Dengan demikian, penelitian ini menyimpulkan bahwa untuk mencapai hasil stabilisasi berkualitas tinggi yang dapat diandalkan untuk aplikasi praktis, strategi dengan analisis global seperti *dynamic programming* adalah pendekatan yang sangkil.

Berdasarkan temuan ini, beberapa arah untuk penelitian selanjutnya dapat dieksplorasi. Salah satu area yang menjanjikan adalah pengembangan algoritma hibrida yang dapat memanfaatkan kecepatan *greedy* pada segmen video yang relatif tenang dan secara cerdas beralih ke pendekatan DP saat gerakan kompleks terdeteksi. Selain itu, optimisasi pada implementasi DP, seperti penggunaan fungsi biaya yang lebih adaptif atau pemrosesan paralel, dapat menjadi fokus untuk mengurangi waktu komputasi tanpa mengorbankan kualitas secara signifikan, sehingga menjembatani kesenjangan kinerja antara kedua pendekatan tersebut.

REFERENSI

- [1] Aggarwal, S. (2021, October 18). Video Stabilization Using Point Feature Matching in OpenCV. Learn OpenCV. Diakses pada 25 Juni 2025, dari <https://learnopencv.com/video-stabilization-using-point-feature-matching-in-opencv/>
- [2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). The MIT Press.
- [3] Janno, J., & Käambre, M. (2018). Real-time video stabilization for mobile phones. In 2018 11th International Conference on Electrical and Electronics Engineering (ELECO) (hlm. 161-165). IEEE. <https://doi.org/10.23919/ELECO.2018.8604245>
- [4] Matsushita, Y., Ofek, E., Ge, W., Tang, X., & Shum, H. Y. (2006). Full-frame video stabilization with motion inpainting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(7), 1150–1163. <https://doi.org/10.1109/TPAMI.2006.141>
- [5] OpenCV. (n.d.). Geometric Image Transformations. OpenCV 4.9.0 documentation. Diakses pada 24 Juni 2025, dari https://docs.opencv.org/4.x/d4/d61/tutorial_basic_geometric_transformations.html

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025



Muhammad Izzat Jundy - 13523092