

Penerapan Sistem Persamaan Linier untuk Pemrosesan Gambar dan Video dalam Efek Visual Film

Jessica Allen -13523059
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
135232059@mahasiswa.itb.ac.id

Abstrak— Pemrosesan gambar dan video memiliki peran penting dalam menciptakan efek visual yang memukau pada film. Salah satu fondasi matematis dalam proses ini adalah Sistem Persamaan Linier. Sistem ini diaplikasikan dalam berbagai teknik seperti transformasi geometris, konvolusi, dan rekonstruksi citra untuk menghasilkan efek visual seperti blur, deteksi tepi, hingga simulasi fluida. Dengan memanfaatkan matriks dan algoritma berbasis linier, kompleksitas algoritma dapat dioptimalkan untuk menghasilkan visual yang realistis. Makalah ini membahas penerapan sistem persamaan linier dalam pemrosesan gambar dan video, memberikan landasan teori, implementasi praktis, hingga contoh aplikasi dalam film terkenal.

Kata Kunci— Pemrosesan Gambar, Sistem Persamaan Linier, Efek Visual, Industri Film

I. PENDAHULUAN

Pada masa kini, menonton film telah menjadi aktivitas yang sangat mudah diakses oleh berbagai kalangan. Dahulu, proses pembuatan dan penayangan film jauh lebih kompleks karena bergantung pada teknologi yang lebih sederhana. Film harus direkam langsung menggunakan kamera analog dan diputar dengan roll film secara manual. Meskipun membutuhkan upaya yang besar, orang-orang tetap membuat film karena media ini telah lama menjadi sarana hiburan yang penting.

Seiring perkembangan teknologi, proses pembuatan film kini mengalami transformasi besar. Dunia perfilman sangat didukung oleh teknologi modern yang memungkinkan para kreator memproduksi karya dengan berbagai efek visual yang memukau. Efek-efek visual ini tidak hanya memperindah film tetapi juga menciptakan pengalaman yang terlihat mustahil untuk direalisasikan secara langsung. Dengan bantuan teknologi, produser dapat menghadirkan efek-efek tersebut dengan lebih mudah dan efisien.

Efek visual pada gambar dan video memiliki peran penting dalam menarik perhatian dan minat penonton, sehingga meningkatkan antusiasme terhadap film yang ditonton. Namun, algoritma yang digunakan untuk menghasilkan efek-efek visual tersebut sering kali terlihat sangat kompleks. Meski demikian, algoritma kompleks ini umumnya dibangun berdasarkan konsep-konsep dasar

yang sederhana.

Salah satu konsep dasar yang menjadi fondasi berbagai algoritma pemrosesan gambar dan video adalah Sistem Persamaan Linier. Konsep ini, meskipun sederhana, dapat diaplikasikan dalam berbagai proses seperti pemrosesan warna, efek blur, deteksi tepi (*edge detection*), penajaman gambar (*sharpening*), *morphing*, *warping*, dan lain sebagainya. Makalah ini akan membahas bagaimana Sistem Persamaan Linier dapat diterapkan dalam pemrosesan gambar dan video untuk menciptakan efek visual dalam film.

II. LANDASAN TEORI

A. Sistem Persamaan Linier

1) Definisi dan Konsep Sistem Persamaan Linier

Sistem Persamaan Linier adalah Kumpulan dari dua atau lebih persamaan linier yang memiliki sejumlah variabel yang saling berkaitan. Kaitan di antara variabel-variabel ini adalah secara linier.

Persamaan linier adalah persamaan yang merepresentasikan hubungan linier antara variabel-variabel dan setiap variabel memiliki pangkat tertinggi sebesar satu. Secara matematis, system persamaan linier dapat dinyatakan dalam bentuk umum sebagai berikut.

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

⋮

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

Keterangan:

- a_{ij} adalah koefisien konstanta yang mengalikan variabel
- x_j adalah variabel yang ingin ditemukan nilainya
- b_i adalah konstanta pada sisi kanan persamaan
- m adalah jumlah persamaan

- n adalah jumlah variable

Solusi sistem persamaan linier dapat memiliki berbagai bentuk, yaitu sebagai berikut.

a) Satu solusi unik

Sebuah sistem persamaan linier dikatakan memiliki satu buah solusi unik ketika terdapat tepat satu nilai untuk setiap variabel yang sedang dicari. Dengan kata lain, tidak terdapat jawaban lain yang dapat memenuhi persamaan tersebut selain solusi unik yang didefinisikan.

b) Tak hingga jumlah solusi

Sebuah sistem persamaan linier dikatakan memiliki jumlah solusi yang tidak terhingga banyaknya ketika terdapat banyak kombinasi nilai variabel yang memenuhi sistem persamaan linier tersebut. Dengan kata lain, tidak hanya satu solusi yang dapat menjadi jawaban atas sistem persamaan linier yang didefinisikan.

c) Tidak ada solusi

Sebuah sistem persamaan linier dikatakan tidak memiliki solusi ketika persamaan-persamaan yang terdapat pada sistem persamaan tersebut saling bertentangan. Hal ini menyebabkan tidak adanya solusi yang dapat memenuhi seluruh persamaan tersebut.

2) Representasi Matriks Sistem Persamaan Linier

Sistem persamaan linier dapat direpresentasikan dalam bentuk matriks. Bentuk matriks ini lebih kompak dan sering sekali digunakan untuk memroses berbagai data. Sistem persamaan linier dengan jumlah persamaan sebanyak m dan dengan variabel sebanyak n dapat dinyatakan sebagai

$$Ax = b$$

Keterangan:

- A adalah matriks koefisien yang berukuran $m \times n$
- x adalah vektor kolom variabel
- b adalah vektor kolom konstanta

3) Metode Penyelesaian Sistem Persamaan Linier

Terdapat berbagai cara yang dapat digunakan untuk menyelesaikan persoalan sistem persamaan linier. Beberapa di antaranya adalah sebagai berikut.

a) Metode Substitusi

Metode substitusi adalah salah satu cara penyelesaian sistem persamaan linier dengan mengisolasi satu buah variabel dalam satu persamaan dan kemudian menggantikan variabel tersebut dalam persamaan lainnya.

Langkah-langkah rinci untuk melakukan metode substitusi adalah sebagai berikut.

- Memilih salah satu persamaan dan mengisolasi salah satu variabel
- Mensubstitusi nilai variabel yang diisolasi ke dalam persamaan lain untuk mengurangi jumlah variabel yang ada pada persamaan tersebut
- Mengulang langkah kedua hingga sistem menjadi cukup sederhana untuk diselesaikan secara langsung
- Terakhir, substitusi kembali nilai variabel yang diperoleh ke dalam persamaan sebelumnya untuk memperoleh solusi yang lengkap

b) Metode Eliminasi

Metode eliminasi adalah teknik untuk menghilangkan salah satu variabel dengan menjumlahkan atau mengurangi persamaan sehingga variabel tersebut dieliminasi.

Langkah-langkah penyelesaian dengan menggunakan metode eliminasi adalah sebagai berikut.

- Mengalikan salah satu atau lebih persamaan dengan sebuah konstanta untuk menyamakan koefisien salah satu variabel
- Menambahkan atau mengurangi persamaan sehingga salah satu variabel hilang.
- Menyelesaikan sistem yang tersisa dengan menggunakan langkah di atas atau dengan metode substitusi.
- Mensubstitusi nilai variabel ke dalam persamaan lainnya untuk memperoleh solusi yang lengkap

c) Metode Matriks

Metode matriks dalam penyelesaian sistem persamaan linier meliputi metode Eliminasi Gauss dan Invers Matriks.

1. Eliminasi Gauss

Eliminasi Gauss merupakan salah satu metode penyelesaian sistem persamaan linier dalam bentuk matriks yang bertujuan untuk mengubah matriks koefisien ke dalam bentuk eselon baris, yaitu matriks segitiga atas, dengan memanfaatkan operasi baris elementer.

Operasi baris elementer adalah tiga jenis manipulasi yang dapat dilakukan pada baris matriks untuk menyelesaikan sistem persamaan linier atau mengubah matriks ke bentuk yang lebih sederhana. Operasi ini tidak mengubah solusi dari sistem

persamaan, yang menjadikannya metode yang sangat sesuai untuk memanipulasikan matriks sistem persamaan linier. Tiga jenis operasi baris elementer adalah sebagai berikut.

a) Penukaran baris

Penukaran baris dalam operasi baris elementer berarti menukarkan posisi dua baris dalam sebuah matriks. Contohnya adalah sebagai berikut.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$$

b) Perkalian baris dengan skalar

Perkalian baris dengan skalar meliputi mengalikan seluruh elemen suatu baris pada matriks dengan konstanta yang bukan nol. Contohnya adalah sebagai berikut, ketika baris pertama pada matriks dikalikan 2.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix}$$

c) Penjumlahan atau pengurangan antar-baris

Penjumlahan atau pengurangan antar-baris berarti menambahkan atau mengurangi kelipatan suatu baris kepada baris lain. Contohnya adalah sebagai berikut, dengan baris kedua dikurangi dengan 3 kali baris pertama.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$$

Langkah-langkah untuk menyelesaikan sistem persamaan linier dengan menggunakan metode matriks dalam eliminasi Gauss adalah sebagai berikut.

- Representasikan sistem persamaan linier $Ax = b$ dalam bentuk augmented matriks yang berbentuk $[A | b]$.
- Lakukan operasi baris elementer yang meliputi penukaran, penjumlahan, atau perkalian untuk membuat elemen-elemen di bawah diagonal utama matriks menjadi nol.

- Terakhir, gunakan substitusi mundur untuk menyelesaikan nilai variabel.

2. Invers Matriks

Jika matriks A dalam sistem persamaan linier $Ax = b$ memiliki sebuah matriks invers, maka solusi dari sistem persamaan linier dapat langsung dihitung dengan menggunakan persamaan berikut.

$$x = A^{-1}b$$

d) Metode Iteratif

Metode iteratif digunakan ketika sistem persamaan linier sudah terlalu besar atau kompleks untuk diselesaikan secara langsung. Solusi dihitung secara bertahap dengan pendekatan berturut-turut hingga konvergen. Beberapa jenis penyelesaian sistem persamaan linier metode iteratif yaitu sebagai berikut.

1. Metode Jacobi

Dalam metode Jacobi, setiap variabel dihitung secara independent berdasarkan nilai dari iterasi sebelumnya. Untuk menggunakan metode ini, matriks A terlebih dahulu dipecah menjadi tiga bagian, yaitu D (matriks diagonal A), L (matriks segitiga bawah tanpa elemen diagonal), dan U (matriks segitiga atas tanpa elemen diagonal).

Iterasi dilakukan dengan menggunakan rumus sebagai berikut.

$$x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)})$$

2. Metode Gauss-Seidel

Berbeda dengan Jacobi, metode Gauss-Seidel memperbarui nilai variabel segera setelah dihitung dalam iterasi. Metode ini mempercepat konvergensi karena nilai yang baru diperbarui dapat langsung digunakan dalam perhitungan selanjutnya.

Iterasi dilakukan dengan menggunakan rumus sebagai berikut.

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij}x_j^{(k+1)} - \sum_{j > i} a_{ij}x_j^{(k)} \right)$$

B. Pemrosesan Gambar dan Video

1) Definisi Pemrosesan Gambar

Pemrosesan gambar adalah serangkaian teknik, algoritma, dan operasi matematis yang diterapkan pada citra digital untuk menghasilkan gambar yang lebih informatif, menarik, atau siap digunakan untuk aplikasi tertentu.

2) Tujuan Pemrosesan Gambar

Pemrosesan gambar dilakukan untuk sejumlah alasan, beberapa di antaranya adalah sebagai berikut.

- Meningkatkan kualitas gambar, seperti perbaikan kontras, penghapusan noise, ataupun penajaman tepi (*sharpening*).
- Mengidentifikasi fitur, seperti deteksi tepi, segmentasi objek, atau pengenalan pola dalam gambar.
- Menghasilkan transformasi estetis atau teknis untuk menghasilkan efek visual, seperti efek blur, morphing, atau warping yang sering digunakan dalam industri perfilman.

3) Citra Digital

Citra digital merupakan representasi gambar dalam bentuk matriks nilai piksel, di mana setiap piksel merepresentasikan intensitas warna pada koordinat tertentu. Berikut merupakan contoh citra digital.

- Gambar *grayscale*, dimana piksel direpresentasikan dengan nilai intensitas dari 0 (hitam) hingga 255 (putih).
- Gambar berwarna, dimana setiap piksel memiliki tiga saluran warna utama (RGB, yaitu merah, hijau, dan biru), dengan masing-masing nilai intensitas bervariasi dari 0 hingga 255 juga.

4) Pemrosesan Video

Video merupakan rangkaian gambar digital (*frame*) yang diproses dalam domain waktu. Setiap *frame* diperlakukan sebagai gambar individu, tetapi dengan mempertimbangkan kesinambungan temporal antar-*frame*.

Beberapa contoh pemrosesan video adalah sebagai berikut.

- Peningkatan kualitas video, seperti mengurangi noise, menyesuaikan kecerahan, atau menstabilkan gerakan.
- Memberikan efek temporal, seperti *slow motion*, efek waktu terhenti, ataupun transisi.
- Menghitung pergerakan antar-*frame* untuk memahami gerakan kamera atau objek dalam video.

Dalam efek visual film, pemrosesan video digunakan untuk menciptakan ilusi gerakan, menggabungkan elemen CGI (*Computer Generated Imagery*, teknologi yang digunakan untuk menciptakan gambar, animasi, ataupun efek visual yang sepenuhnya dihasilkan oleh komputer), ataupun mengubah adegan secara dramatis.

C. Efek Visual dalam Film

Efek Visual (VFX) adalah proses pembuatan atau manipulasi elemen visual yang tidak dapat difilmkan secara langsung atau sulit untuk dilakukan dalam dunia

nyata. VFX digunakan untuk menciptakan adegan atau elemen visual yang spektakuler tetapi terlihat realistis dengan menggabungkan elemen yang diambil langsung dengan elemen yang dihasilkan oleh komputer.

Terdapat banyak jenis efek visual dalam film, beberapa di antaranya adalah sebagai berikut.

- Efek CGI (*Computer-Generated Imagery*), yang meliputi pembuatan elemen digital seperti karakter, pemandangan, atau objek tiga dimensi yang terlihat realistis.
- Efek praktis dengan *Digital Enhancement*, yang meliputi penggabungan efek fisik yang nyata dengan sentuhan digital untuk menyempurnakan visualnya.
- Chroma Keying*, atau lebih sering dikenal dengan *Green Screen*, yang meliputi penggantian latar belakang hijau atau biru dengan gambar lain menggunakan Teknik digital.
- Motion Capture*, yang menggunakan sensor untuk merekam gerakan aktor yang kemudian diterjemahkan menjadi karakter digital.
- Simulasi Efek Alam, yang berfungsi untuk membuat elemen seperti air, api, asap, atau lainnya secara digital.

D. Aplikasi Sistem Persamaan Linear dalam Pemrosesan Gambar/Video

1) Transformasi Geometri

Transformasi geometri melibatkan penggunaan matriks linier untuk melakukan operasi seperti rotasi, translasi, skala, dan transformasi perspektif pada citra digital. Operasi ini memungkinkan manipulasi posisi dan orientasi gambar secara matematis. Contoh transformasi matriks adalah sebagai berikut.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Dengan (x,y) adalah koordinat awal, (x', y') adalah koordinat setelah transformasi, dan matriks 3×3 yang digunakan menentukan jenis transformasi yang diterapkan.

2) Konvolusi

Konvolusi adalah operasi matematika yang menggabungkan dua fungsi untuk menghasilkan fungsi ketiga. Dalam pemrosesan citra, konvolusi digunakan untuk menerapkan berbagai efek seperti *blur*, *edge detection*, dan *sharpening* dengan mengalikan matriks kernel (filter) dengan matriks citra. Rumus dasar konvolusi 2D antara matriks gambar $I(x, y)$ dan kernel $K(i, j)$ diberikan oleh rumus berikut.

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k K(i, j) \cdot I(x + i, y + j)$$

Keterangan:

- $I(x, y)$ adalah nilai piksel pada koordinat (x, y) dalam gambar asli.

- $K(i, j)$ adalah nilai elemen kernel pada koordinat (i, j) .
- $I(x, y)$ adalah nilai piksel pada koordinat (x, y) dalam gambar hasil.
- k adalah setengah ukuran kernel (untuk kernel 3×3 , $k = 1$)

3) Optical Flow

Optical flow adalah teknik untuk menghitung vektor kecepatan yang merepresentasikan pergerakan piksel dari satu *frame* ke *frame* berikutnya dalam video. Sistem persamaan linier digunakan untuk menentukan arah dan kecepatan gerakan dalam analisis optical flow.

Dengan berasumsi bahwa intensitas piksel tetap konstan meskipun ada pergerakan dalam frame, yang menyebabkan piksel pada posisi tertentu di frame pertama $I(x, y, t)$ akan berpindah ke posisi baru $I(x + \Delta x, y + \Delta y, t + \Delta t)$ pada frame berikutnya. Perubahan intensitas piksel ini dapat diwakili dengan persamaan Taylor.

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t$$

Jika intensitas piksel konstan, maka

$$I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t) = 0$$

Kemudian, akan diperoleh

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

Keterangan:

- $\partial I / \partial x$ dan $\partial I / \partial y$ adalah gradien intensitas dalam arah x dan y
- $\partial I / \partial t$ adalah perubahan intensitas piksel antar-frame
- v_x dan v_y adalah komponen vector kecepatan piksel dalam arah x dan y

Persamaan di atas dapat direpresentasikan dalam bentuk matriks linier untuk Kumpulan piksel, yaitu sebagai berikut.

$$Gv = -e$$

Untuk setiap piksel, persamaannya adalah sebagai berikut

$$\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\frac{\partial I}{\partial t}$$

Terdapat dua jenis metode yang dapat digunakan untuk mendapatkan solusi, yaitu Lucas-Kanade Method dan juga Horn-Schunck Method.

a) Lucas-Kanade Method

Menggunakan asumsi bahwa kecepatan piksel di area kecil ($N \times N$) adalah sama, maka persamaan linier untuk banyak piksel pada area tersebut adalah sebagai berikut.

$$A^T A v = A^T b$$

b) Horn-Schunck Method

Metode ini menambahkan regularisasi untuk membuat solusi menjadi lebih halus,

dengan memanfaatkan parameter regularisasi (λ).

$$E = \int \left(\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} \right)^2 dx dy + \lambda \int (|\nabla v_x|^2 + |\nabla v_y|^2) dx dy$$

4) Rekonstruksi Citra

Rekonstruksi citra melibatkan penggunaan sistem persamaan linier untuk memperbaiki atau membangun kembali citra yang rusak atau tidak lengkap.

Citra digital direpresentasikan sebagai matriks I dengan nilai piksel $I(x, y)$ di setiap koordinat. Citra yang rusak dapat direpresentasikan dengan elemen matriks yang hilang atau termodifikasi, yaitu sebagai berikut.

$$I_{\text{rusak}}(x, y) = \begin{cases} I(x, y), & \text{jika data tersedia} \\ ?, & \text{jika data hilang} \end{cases}$$

Contoh penerapan sederhana adalah dengan memanfaatkan teknik sederhana seperti interpolasi, inpainting, atau dekomposisi spektral sebagai sistem persamaan linier untuk memodelkan hubungan antara piksel yang diketahui dan piksel yang hilang. Misalnya, jika kita memiliki citra 3×3 dengan piksel sebagai berikut.

$$I = \begin{bmatrix} 1 & 2 & ? \\ 4 & ? & 6 \\ ? & 8 & 9 \end{bmatrix}$$

Piksel yang hilang dapat diperkirakan dengan menggunakan rata-rata piksel tetangganya, yang merupakan contoh sederhana penggunaan interpolasi linier.

$$I(1, 3) = \frac{I(1, 2) + I(2, 3)}{2} = \frac{2 + 6}{2} = 4$$

E. Aljabar Linier dalam Pemrosesan Data Digital

1) Transformasi Warna

Transformasi warna digunakan untuk mengonversi gambar dari satu model warna ke model lainnya atau untuk melakukan manipulasi warna tertentu. Operasi ini melibatkan matriks transformasi berbasis sistem persamaan linier.

a) Model Warna

1. RGB (Red, Green, Blue), yang merupakan representasi gambar dalam tiga saluran warna utama.
2. Grayscale, yaitu gambar dengan satu saluran intensitas (hitam-putih) yang dapat dihitung sebagai kombinasi dari RGB, yaitu sebagai berikut.

$$Y = 0.299R + 0.587G + 0.114B$$

3. YUV/YCbCr, yaitu model warna yang memisahkan luminansi (Y) dan

krominansi (U dan V) untuk efisiensi kompresi video.

- b) Transformasi Warna dengan Matriks
Konversi antar-model warna dapat direpresentasikan dalam matriks sebagai berikut.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2) Singular Value Decomposition (SVD)

SVD adalah dekomposisi matriks yang digunakan untuk menganalisis struktur data, termasuk kompresi dan rekonstruksi gambar. Setiap matriks A berukuran $m \times n$ dapat diuraikan menjadi:

$$A = U\Sigma V^T$$

Keterangan:

- U adalah matriks orthogonal ($m \times m$)
- Σ adalah matriks diagonal ($m \times n$) yang berisi nilai singular
- V^T adalah matriks orthogonal ($n \times n$)

SVD dapat diaplikasikan dalam kompresi gambar, dengan menyimpan hanya beberapa nilai singular terbesar untuk merepresentasikan gambar. Matriks Σ dengan nilai singular yang lebih kecil dibuang untuk mengurangi ukuran data.

Selain itu, SVD juga dapat digunakan untuk rekonstruksi citra dengan menggunakan nilai singular utama untuk membangun kembali gambar dengan detail yang cukup baik, meski tidak sempurna.

3) Fourier Transform

Fourier Transform (FT) digunakan untuk menganalisis sinyal (gambar atau audio) dalam domain frekuensi. Ini berhubungan erat dengan sistem linier karena FT mendekomposisi sinyal menjadi kombinasi linier fungsi sinusoidal.

FT dapat diaplikasikan dalam filtering dengan memisahkan komponen frekuensi rendah (blur) atau tinggi (edge detection) pada gambar. Selain itu, FT juga dapat digunakan dalam kompresi data dengan menggunakan representasi frekuensi untuk mengabaikan komponen yang kurang penting.

$$F(k) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i k x} dx$$

Keterangan:

- $F(k)$ adalah representasi sinyal dalam domain frekuensi
- $f(x)$ adalah sinyal dalam domain waktu atau spasial
- k adalah frekuensi

III. IMPLEMENTASI

A. Kompleksitas Algoritma dan Implementasi

1) Efisiensi Komputasi

Sistem Persamaan Linier digunakan dalam pemrosesan gambar dan video karena efisiensinya dalam menyelesaikan masalah besar dengan cepat.

Komputasi berbasis matriks juga sangat cocok untuk algoritma pemrosesan gambar/video karena banyaknya operasi seperti transformasi geometri, konvolusi, atau filtering dapat dioptimalkan dengan operasi linier.

2) Penerapan dalam Software VFX

Software seperti Adobe After Effects, Blender, dan Maya menggunakan algoritma berbasis sistem linier untuk melakukan berbagai hal, seperti menghitung transformasi objek (rotasi, translasi, skala), simulasi fisik (asap, air, deformasi), ataupun efek visual lainnya seperti blur, sharpening, atau deteksi tepi.

B. Studi Kasus

1) Simulasi Asap atau Air

Simulasi asap atau air dalam film menggunakan sistem persamaan linier untuk menyelesaikan persamaan Navier-Stokes, yang menggambarkan perilaku fluida. Persamaan ini digunakan untuk menghasilkan simulasi fisik yang realistis.

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}$$

Keterangan:

- \mathbf{v} adalah vektor kecepatan fluida
- t adalah waktu
- ρ adalah massa jenis fluida
- p adalah tekanan fluida
- ν adalah viskositas fluida
- \mathbf{f} adalah gaya eksternal

Untuk menyelesaikan persamaan ini, domain fluida dibagi menjadi grid. Sistem persamaan linier digunakan untuk menghitung kecepatan fluida pada setiap titik grid serta distribusi tekanan untuk memastikan fluida bersifat inkompresibel. Solusi persamaan dilakukan dengan menggunakan metode numerik seperti Eliminasi Gauss atau Metode Iteratif Jacobi.

Contoh penggunaan simulasi asap atau air ini adalah pada film Avatar (2009), ketika simulasi air dan asap memanfaatkan solusi Navier-Stokes untuk menciptakan efek fluida yang realistis. Selain itu, dalam film animasi Frozen II, air simulasi juga digunakan untuk menciptakan interaksi antara karakter dan lingkungan.

2) Deformasi Objek

Deformasi objek melibatkan perubahan bentuk objek secara dinamis berdasarkan

pengaruh gaya atau interaksi dengan lingkungan. Deformasi objek sering dimodelkan menggunakan Finite Element Method (FEM) atau pendekatan berbasis matriks untuk menghitung perubahan bentuk. Deformasi objek elastis dapat dihitung menggunakan persamaan linier sebagai berikut.

$$K\mathbf{u} = \mathbf{f}$$

Keterangan:

- K adalah Matriks kekakuan (stiffness matrix) yang merepresentasikan sifat elastis material.
- U adalah Vektor perpindahan titik-titik pada objek.
- F adalah Vektor gaya eksternal yang diterapkan pada objek.

Contoh aplikasi deformasi objek adalah dalam skinning karakter, untuk membuat karakter lebih halus. Biasanya, skinning ini digunakan dalam film animasi seperti Zootopia atau Toy Story. Selain itu, deformasi juga digunakan dalam adegan tabrakan mobil atau kehancuran bangunan dalam film aksi seperti Fast & Furious.

C. Implementasi Program Sederhana

1) Blur

Untuk menunjukkan implementasi program sederhana Sistem Persamaan Linier dalam pemrosesan gambar, akan digunakan beberapa pustaka dari python dalam aplikasinya. Berikut adalah pustaka yang digunakan.

```
import numpy as np
from scipy.ndimage import convolve
import matplotlib.pyplot as plt
from skimage import io, color
```

Gambar 1. Library Program Blur

Setelah itu, dibaca file gambar yang ingin diproses dan didefinisikan kernel yang akan digunakan untuk memroses gambar. Kernel efek blur dapat diubah sesuai dengan intensitas blur yang diinginkan.

```
# Membaca gambar berwarna
image = io.imread('image.jpg')

# Kernel untuk efek blur
kernel = np.ones((10, 10)) / 100
```

Gambar 2. Potongan Program Blur (1)

Proses gambar (konvolusi) dengan kernel pada setiap saluran.

```
# Fungsi untuk menerapkan konvolusi pada setiap saluran
def apply_blur(image, kernel):
    blurred_channels = []
    for i in range(3): # Proses masing-masing RGB
        blurred_channel = convolve2d(image[:, :, i],
                                     kernel, mode='same', boundary='symm')
        blurred_channels.append(blurred_channel)
    return np.stack(blurred_channels, axis=-1)

# Terapkan blur pada gambar
blurred_image = apply_blur(image, kernel)
```

Gambar 3. Potongan Program Blur (2)

Didefinisikan penampilan hasil gambar blur dan original.

```
# Menampilkan hasil
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image)

plt.subplot(1, 2, 2)
plt.title("Blurred Image")
plt.imshow(blurred_image.astype('uint8'))
plt.show()
```

Gambar 4. Potongan Program Blur (3)

2) Deteksi Tepi

Dalam implementasi program sederhana deteksi tepi, digunakan beberapa pustaka dari python dalam aplikasinya. Berikut adalah pustaka yang digunakan.

Selanjutnya adalah pembacaan gambar serta pendefinisian kernel sobel untuk deteksi tepi. Setelah itu, diterapkanlah deteksi tepi pada gambar, setelah didefinisikan fungsi untuk melakukannya. Terakhir, ditampilkanlah hasil deteksi tepi gambar.

Dalam aplikasinya, dapat deteksi tepi gambar dapat dilakukan pada gambar berwarna maupun gambar grayscale. Implementasi keduanya berbeda, yang akan ditunjukkan pada gambar di bawah.

Gambar Berwarna:

```
# Membaca gambar
image = io.imread('image.jpg')

# Kernel Sobel untuk deteksi tepi
sobel_x = np.array([[ -1,  0,  1],
                    [ -2,  0,  2],
                    [ -1,  0,  1]])

sobel_y = np.array([[ -1, -2, -1],
                    [  0,  0,  0],
                    [  1,  2,  1]])
```

Gambar 5. Potongan Program Deteksi Tepi Berwarna (1)

```
# Fungsi untuk menerapkan deteksi tepi
def detect_edges(image, sobel_x, sobel_y):
    edges_channels = []
    for i in range(3): # Proses masing-masing RGB
        edges_x = convolve(image[:, :, i], sobel_x)
        edges_y = convolve(image[:, :, i], sobel_y)
        edges = np.hypot(edges_x, edges_y)
        edges_channels.append(edges)
    return np.stack(edges_channels, axis=-1)

# Terapkan deteksi tepi
edges_image = detect_edges(image, sobel_x, sobel_y)
```

Gambar 6. Potongan Program Deteksi Tepi Berwarna (2)

```
# Menampilkan hasil
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image)

plt.subplot(1, 2, 2)
plt.title("Edge Detection")
plt.imshow(edges_image.astype('uint8'))
plt.show()
```

Gambar 7. Potongan Program Deteksi Tepi Berwarna (3)

Gambar Grayscale:

```
# Membaca gambar
image = color.rgb2gray(io.imread('image.jpg'))

# Kernel Sobel untuk deteksi tepi
sobel_x = np.array([[ -1, 0, 1],
                    [ -2, 0, 2],
                    [ -1, 0, 1]])

sobel_y = np.array([[ -1, -2, -1],
                    [ 0, 0, 0],
                    [ 1, 2, 1]])
```

Gambar 8. Potongan Program Deteksi Tepi Grayscale (1)

```
# Konvolusi gambar dengan kernel Sobel
edges_x = convolve(image, sobel_x)
edges_y = convolve(image, sobel_y)
edges = np.hypot(edges_x, edges_y)
```

Gambar 9. Potongan Program Deteksi Tepi Grayscale (2)

```
# Menampilkan hasil
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image, cmap='gray')

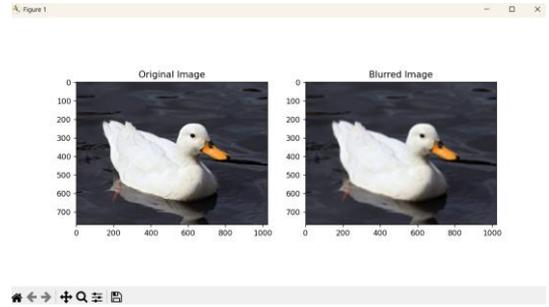
plt.subplot(1, 2, 2)
plt.title("Edge Detection")
plt.imshow(edges, cmap='gray')
plt.show()
```

Gambar 10. Potongan Program Deteksi Tepi Grayscale (3)

IV. PERCOBAAN

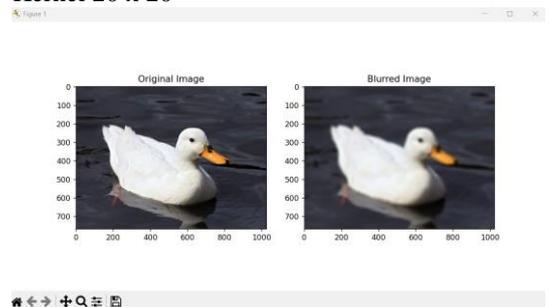
A. Hasil Percobaan Program Blur

- 1) Kernel 10 x 10



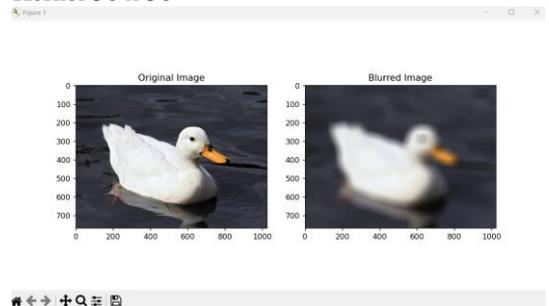
Gambar 11. Hasil Program Blur (1)

- 2) Kernel 20 x 20



Gambar 12. Hasil Program Blur (2)

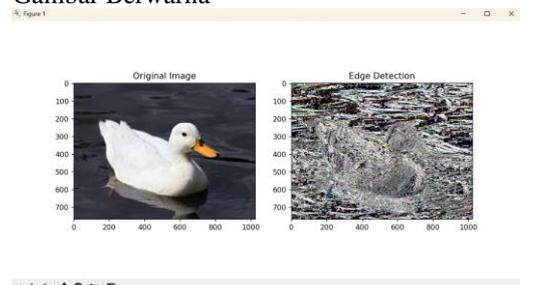
- 3) Kernel 50 x 50



Gambar 13. Hasil Program Blur (3)

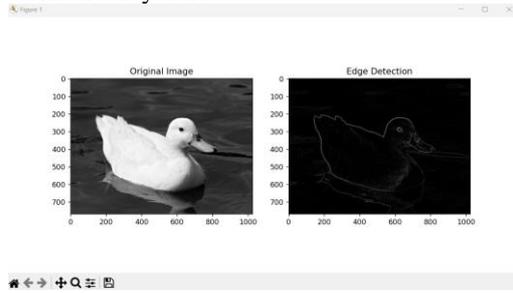
B. Hasil Percobaan Program Deteksi Tepi

- 1) Gambar Berwarna



Gambar 14. Hasil Program Deteksi Tepi (1)

2) Gambar Grayscale



Gambar 15. Hasil Program Blur (2)

V. KESIMPULAN

Sistem Persamaan Linier menjadi fondasi penting dalam berbagai algoritma pemrosesan gambar dan video. Melalui transformasi geometris, konvolusi, dan rekonstruksi citra, teknik ini dapat menciptakan efek visual yang tidak hanya estetis tetapi juga fungsional, seperti simulasi fluida atau deformasi objek. Implementasi dalam software VFX seperti Adobe After Effects dan Blender menunjukkan bagaimana sistem ini diterapkan secara praktis. Dengan efisiensi komputasi dan fleksibilitasnya, Sistem Persamaan Linier terus menjadi dasar dalam inovasi visual di industri perfilman.

VI. APRESIASI

Penulis ingin mengucapkan terima kasih sebesar-besarnya kepada seluruh dosen IF2123 Aljabar Linier dan Geometri, khususnya kepada Ir. Rila Mandala, M.Eng., Ph.D. sebagai dosen pengajar Kelas 1 IF2123, atas dukungan dan kesabarannya dalam mengajar mahasiswa/i. Melalui makalah ini, saya belajar lebih dalam mengenai penerapan Sistem Persamaan Linier dalam kaitannya dengan bidang yang saya minati. Saya juga mengucapkan terima kasih yang spesial kepada Dr. Ir. Rinaldi, M.T, yang telah menyediakan wadah pembelajaran bagi para mahasiswa/i Teknik Informatika melalui websitenya. Website tersebut telah membantu beribu mahasiswa dalam memahami mata kuliah yang ada di ITB, tidak luput juga membantu mereka dalam berkarya.

REFERENSI

- [1] R. Munir, "Aljabar Linier dan Geometri - ITB, 2024-2025," Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi/munir/AljabarGeometri/algeo.htm>. Diakses pada December 31, 2024.
- [2] Sukarna, M. Abdy, dan Rahmat, "Perbandingan Metode Iterasi Jacobi dan Metode Iterasi Gauss-Seidel dalam Menyelesaikan Sistem Persamaan Linear Fuzzy," *Journal of Mathematics, Computations, and Statistics*, vol. 2, no. 1, April 2019. [Online]. Available: <https://ojs.unm.ac.id/jmathcos/article/view/12447>. Diakses pada December 31, 2024.
- [3] M. Rifqi, "Pengertian dan Teknik Dasar Image Processing," Rifqi Mulyawan Blog, 2022. [Online]. Available:

<https://rifqimulyawan.com/blog/pengertian-image-processing/>.

Diakses pada December 31, 2024.

- [4] "Teknologi Pemrosesan Video: Konsep dan Aplikasi," *Gadgetzku*, 2023. [Online]. Available: <https://www.gadgetzku.com/teknologi-pemrosesan-video/>. Diakses pada December 31, 2024.
- [5] IDS Education, "Istilah dalam Visual Effect (VFX)," *IDS Education Blog*, 2023. [Online]. Available: <https://idseducation.com/istilah-dalam-visual-effect-vfx/>. Diakses pada December Januari 1, 2025.
- [6] P2DPT UMA, "Transformasi Sinematik: Bagaimana Efek Visual Mengubah Industri Film," *Pusat Pengembangan dan Penelitian Teknologi Universitas Medan Area*, 2024. [Online]. Available: <https://p2dpt.uma.ac.id/2024/09/12/transformasi-sinematik-bagaimana-efek-visual-mengubah-industri-film/>. Diakses pada Januari 1, 2025.
- [7] Lighttups, "How Visual Effects Work in Film," *Lighttups*, 2023. [Online]. Available: <https://id.lighttups.io/how-visual-effects-work-film/>. Diakses pada Januari 1, 2025.
- [8] H. Setiawan, "Implementasi Aljabar Linier dalam Pengolahan Citra Digital," *Jurnal Dunia Ilmu*, vol. 4, no. 9, 2024. [Online]. Available: <https://duniailmu.org/index.php/repo/article/view/528>. Diakses pada Januari 1, 2025.
- [9] "Rekonstruksi Citra: Proses Pembentukan Gambar pada CT Scan," *123dok*. [Online]. Available: <https://text-id.123dok.com/document/nq7652vy6-rekonstruksi-citra-proses-pembentukan-gambar-pada-ct-scan.html>. Diakses pada Januari 1, 2025.
- [10] C. Hapsoro dan A. Srigutomo, "Pemodelan Aliran Fluida 2-D Pada Kasus Aliran Permukaan Menggunakan Metode Beda Hingga," *ResearchGate*, 2018. [Online]. Available: https://www.researchgate.net/publication/329388603_2-D_Fluid_Surface_Flow_Modeling_using_Finite-Difference_Method. Diakses pada Januari 1, 2025.

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, 2 Januari 2024

Jessica Allen 13523059