

# Pemisahan Background dan Foreground pada Video Menggunakan Low-Rank Approximation Berbasis Singular Value Decomposition (SVD)

Muhammad Haris Putra Sulastianto<sup>1,2</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>[13524053@itb.ac.id](mailto:13524053@itb.ac.id) <sup>2</sup>[mharispuras.work@gmail.com](mailto:mharispuras.work@gmail.com)

**Abstract**—Pada video dengan kamera statis, tampilan background umumnya tidak banyak berubah dari satu frame ke frame berikutnya, sedangkan objek yang bergerak hanya muncul pada sebagian kecil area gambar. Sifat ini dapat dimanfaatkan untuk memisahkan background dan foreground menggunakan pendekatan aljabar linier. Pada makalah ini, digunakan metode *low-rank approximation* berbasis *Singular Value Decomposition* (SVD) untuk memodelkan background sebagai komponen ber-rank rendah dan foreground sebagai selisih antara video asli dan hasil aproksimasi background. Prosesnya dilakukan dengan mengambil beberapa *singular value* terbesar digunakan untuk membentuk aproksimasi low-rank yang merepresentasikan background, sementara selisih antara data asli dan aproksimasi low-rank menghasilkan komponen foreground. Metode ini mampu memisahkan background dan objek bergerak secara cukup jelas dengan langkah komputasi yang relatif mudah dengan beberapa syarat teknis video yaitu kamera statis, background stabil, foreground jarang, dan cahaya stabil. Namun, karena foreground dimodelkan sebagai komponen residual akibatnya foreground yang dihasilkan tidak mempertahankan warna asli objek, melainkan hanya menunjukkan bagian yang mengalami perubahan.

**Keywords**—Low-rank approximation, singular value decomposition (SVD), matrix decomposition, matrix rank, background modeling, foreground detection

## I. PENDAHULUAN

Video yang direkam menggunakan kamera statis memiliki karakteristik berupa background yang relatif konsisten dari satu frame ke frame berikutnya, sementara perubahan visual umumnya disebabkan oleh keberadaan objek bergerak. Perbedaan sifat antara bagian yang statis dan bagian yang berubah ini dapat dianalisis secara matematis melalui representasi video dalam bentuk matriks. Dengan memandang setiap frame sebagai vektor dan seluruh rangkaian video sebagai kumpulan vektor dalam ruang berdimensi tinggi, permasalahan pemisahan background dan foreground dapat diformulasikan menggunakan konsep-konsep aljabar linier.

Dalam aljabar linier, struktur data yang bersifat konsisten dan redundan sering kali dapat direpresentasikan oleh subruang berdimensi lebih rendah. Pada konteks video dengan kamera statis, kesamaan antar-frame menyebabkan matriks representasi video memiliki *rank* efektif yang relatif kecil. Struktur ber-rank rendah ini merepresentasikan pola dominan

yang muncul secara stabil sepanjang waktu, yang dalam hal ini berkaitan dengan background video. Sebaliknya, perubahan lokal yang muncul akibat objek bergerak tidak mengikuti pola dominan tersebut dapat dipandang sebagai komponen residual yang dalam hal ini berkaitan dengan foreground video.

Salah satu teknik dekomposisi matriks yang mampu menggkap struktur ber-rank rendah adalah *Singular Value Decomposition* (SVD). Melalui SVD, suatu matriks dapat diuraikan menjadi komponen-komponen ortogonal yang diurutkan berdasarkan kontribusinya terhadap struktur keseluruhan matriks. Dengan mempertahankan sejumlah *singular value* terbesar, diperoleh aproksimasi matriks ber-rank rendah yang merepresentasikan struktur dominan data, yang pada video berkaitan dengan background. Selisih antara matriks asli dan hasil aproksimasi tersebut kemudian memuat variasi yang lebih kecil dan berkaitan dengan foreground.

## II. DASAR TEORI

### A. Representasi Video dalam Bentuk Matriks

Video digital merupakan rangkaian frame yang direkam secara berurutan dalam selang waktu tertentu. Setiap frame pada video dapat dipandang sebagai sebuah citra digital yang tersusun atas piksel-piksel dengan nilai intensitas tertentu. Untuk keperluan analisis matematis, setiap frame video dapat direpresentasikan sebagai sebuah vektor dengan cara meratakan (flatten) matriks piksel menjadi satu dimensi.

Misalkan sebuah video terdiri dari  $T$  frame dan setiap frame memiliki ukuran  $H \times W$  piksel. Dengan melakukan proses flatten pada setiap frame, diperoleh vektor berdimensi  $D = H \times W$ . Seluruh frame kemudian disusun sebagai baris dalam sebuah matriks sehingga diperoleh matriks representasi video

$$X \in \mathbb{R}^{T \times D},$$

di mana setiap baris merepresentasikan satu frame video dan setiap kolom merepresentasikan posisi piksel yang sama pada seluruh frame.

### B. Rank Matriks dan Aproksimasi Rank Rendah

Rank suatu matriks menyatakan jumlah maksimum baris atau kolom yang saling bebas secara linier. Dalam konteks data

berdimensi tinggi, sering dijumpai bahwa meskipun ukuran matriks besar, informasi yang terkandung di dalamnya bersifat redundan sehingga dapat direpresentasikan secara efektif oleh subruang berdimensi lebih rendah.

Pada video dengan kamera statis, kesamaan visual antar-frame menyebabkan baris-baris pada matriks representasi video memiliki korelasi yang tinggi. Akibatnya, matriks tersebut memiliki *rank* efektif yang relatif rendah. Matriks dengan karakteristik seperti ini disebut sebagai matriks ber-rank rendah (*low-rank matrix*).

Aproksimasi rank rendah bertujuan untuk mendekati matriks asli dengan matriks lain yang memiliki rank lebih kecil, namun tetap mempertahankan struktur utama data.

### C. Singular Value Decomposition

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

Gambar 1. SVD: Dekomposisi Matriks

Sumber: [Link Sumber](#)

[1], [2] *Singular Value Decomposition* (SVD) merupakan salah satu teknik dekomposisi matriks yang umum digunakan dalam aljabar linier. SVD menyatakan bahwa setiap matriks  $X \in \mathbb{R}^{m \times n}$  dapat diuraikan menjadi

$$X = U\Sigma V^T,$$

di mana:

- $U \in \mathbb{R}^{m \times m}$  adalah matriks ortogonal yang kolom-kolomnya disebut vektor singular kiri,
- $\Sigma \in \mathbb{R}^{m \times n}$  adalah matriks diagonal yang berisi nilai-nilai singular (*singular values*),
- $V \in \mathbb{R}^{n \times n}$  adalah matriks ortogonal yang kolom-kolomnya disebut vektor singular kanan.

Nilai singular pada matriks  $\Sigma$  disusun secara menurun dan menunjukkan tingkat kontribusi masing-masing komponen terhadap struktur keseluruhan matriks. Komponen dengan nilai singular besar merepresentasikan pola dominan, sedangkan nilai singular kecil berkaitan dengan detail atau variasi yang lebih lemah.

### D. Truncated SVD and Aproksimasi Rank Rendah

Dalam praktik, tidak semua komponen hasil SVD diperlukan. Dengan mempertahankan hanya  $k$  nilai singular terbesar

beserta vektor singular yang terkait, diperoleh dekomposisi SVD terpotong (*truncated SVD*) sebagai berikut:

$$X \approx U_k \Sigma_k V_k^T,$$

di mana  $k < \min(m, n)$ . Aproksimasi ini merupakan aproksimasi rank rendah terbaik terhadap matriks  $X$  dalam arti kuadrat kesalahan minimum.

Komponen  $U_k \Sigma_k V_k^T$  merepresentasikan struktur utama data, sementara selisih antara matriks asli dan aproksimasi tersebut merepresentasikan komponen yang tidak dapat dijelaskan oleh struktur dominan.

### E. Pemisahan Background dan Foreground pada Video

Pada video dengan kamera statis, background muncul secara konsisten pada hampir seluruh frame, sehingga dapat dimodelkan sebagai komponen ber-rank rendah. Dengan menerapkan truncated SVD pada matriks representasi video, komponen low-rank yang diperoleh digunakan sebagai estimasi background video.

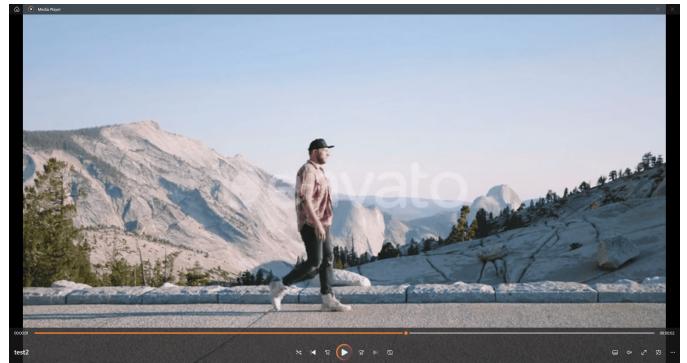
Foreground diperoleh sebagai selisih antara matriks video asli dan hasil aproksimasi low-rank, yaitu

$$X_{fg} = X - X_{bg},$$

dengan  $X_{bg} = U_k \Sigma_k V_k^T$ . Karena foreground diperoleh sebagai hasil selisih terhadap background, warna foreground yang dihasilkan tidak akan sama dengan warna pada video aslinya, melainkan hanya menunjukkan bagian-bagian frame yang mengalami perubahan.

## III. IMPLEMENTASI

### A. Pra-pemrosesan Video



Gambar 2. Contoh Video yang Memenuhi Kriteria

Sumber: <https://www.youtube.com/watch?v=84IYjtCflvY>

Tahap awal implementasi dimulai dengan menyiapkan data video yang direkam menggunakan kamera statis. Video diasumsikan memiliki background yang relatif stabil, pencahayaan yang tidak berubah secara signifikan, serta objek bergerak yang tidak muncul dalam waktu lama, tidak berukuran besar, dan tidak mendominasi area citra. Setiap video kemudian diekstraksi menjadi sejumlah frame dengan interval waktu tertentu.

```

1 while True:
2     ret, frame = cap.read()
3     if not ret:
4         break
5     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
6     frames.append(gray)

```

Gambar 3. Code: Konversi Grayscale

Untuk menyederhanakan pemrosesan dan mengurangi kompleksitas data, setiap frame dikonversi ke dalam citra *grayscale*. Seluruh frame yang digunakan berasal dari satu sumber video dan memiliki ukuran yang sama, sehingga setiap piksel merepresentasikan posisi spasial yang konsisten pada seluruh frame.

#### B. Pembentukan Matriks Representasi Video

```

1 frames = np.array(frames)      # (T, H, W)
2 T, H, W = frames.shape
3
4 X = frames.reshape(T, H * W)   # (T, HW)
5 X = X.astype(np.float64)

```

Gambar 4. Code: *Flatten* Matriks Frame

Setiap frame hasil pra-pemrosesan direpresentasikan sebagai sebuah vektor dengan cara meratakan (*flatten*) matriks piksel menjadi vektor satu dimensi. Jika ukuran setiap frame adalah  $H \times W$ , maka diperoleh vektor berdimensi  $D = H \times W$ .

Seluruh vektor frame kemudian disusun sebagai baris dalam sebuah matriks sehingga membentuk matriks representasi video

$$X \in \mathbb{R}^{T \times D},$$

di mana  $T$  menyatakan jumlah frame dan setiap baris merepresentasikan satu frame video. Dengan representasi ini, perubahan visual antar-frame dapat dianalisis secara matematis menggunakan operasi aljabar linier.

#### C. Penerapan Singular Value Decomposition

Setelah matriks representasi video  $X$  terbentuk, dilakukan dekomposisi menggunakan *Singular Value Decomposition* sehingga diperoleh

$$X = U \Sigma V^T.$$

Matriks  $\Sigma$  berisi nilai-nilai singular yang diurutkan secara menurun. Nilai singular terbesar menunjukkan komponen yang

```

1 U, S, VT = np.linalg.svd(X, full_matrices=False)

```

Gambar 5. Code: Operasi SVD pada Matriks Video

paling dominan dalam data, sedangkan nilai singular kecil merepresentasikan variasi yang lebih lemah.

Matriks  $U$  dan  $V$  merupakan matriks ortogonal yang masing-masing memuat vektor singular kiri dan vektor singular kanan. Pada representasi matriks video  $X \in \mathbb{R}^{T \times D}$ , baris-baris matriks  $X$  merepresentasikan frame video, sedangkan kolom-kolomnya merepresentasikan piksel.

Matriks  $U$  menggambarkan keterkaitan antar-frame dalam domain waktu, di mana setiap kolom  $U$  menunjukkan bagaimana suatu komponen utama muncul sepanjang rangkaian frame. Sementara itu, matriks  $V$  memuat pola-pola spasial pada domain piksel, di mana setiap kolom  $V$  merepresentasikan distribusi nilai piksel yang membentuk struktur visual dominan pada video.

#### D. Aproksimasi Rank Rendah untuk Background

```

1 k = 1
2
3 U_k = U[:, :k]
4 S_k = S[:k]
5 VT_k = VT[:k, :]
6
7 X_bg = U_k @ np.diag(S_k) @ VT_k
8

```

Gambar 6. Code: Aproksimasi *low-rank* Komponen SVD untuk background

Untuk memperoleh komponen background, digunakan aproksimasi rank rendah dengan mempertahankan  $k$  nilai singular terbesar beserta vektor singular yang terkait. Aproksimasi ini dinyatakan sebagai

$$X_{bg} = U_k \Sigma_k V_k^T,$$

dengan  $k < \min(T, D)$ . Matriks  $X_{bg}$  merepresentasikan struktur dominan yang muncul secara konsisten sepanjang waktu, sehingga dipandang sebagai estimasi background video.

Setiap baris dari  $X_{bg}$  kemudian dapat dibentuk kembali menjadi frame dua dimensi untuk memperoleh citra background pada masing-masing waktu.

Pada implementasi ini, nilai  $k$  dipilih sebesar 1 untuk menangkap struktur dominan background, dengan asumsi bahwa background video bersifat sangat stabil sepanjang waktu.

#### E. Ekstraksi Foreground sebagai Komponen Residual

```

1 X_fg = X - X_bg
2
3 background = X_bg.reshape(T, H, W)
4 foreground = X_fg.reshape(T, H, W)

```

Gambar 7. Code: Komponen Residual sebagai Foreground

```

1 for i in range(T):
2     bg_frame = np.clip(background[i], 0, 255).astype(np.uint8)
3     fg_frame = np.clip(np.abs(foreground[i]), 0, 255).astype(np.uint8)
4
5     out_bg.write(bg_frame)
6     out_fg.write(fg_frame)

```

Gambar 8. Code: Membentuk Kembali Matriks menjadi Video

Foreground diperoleh dengan menghitung selisih antara matriks video asli dan hasil aproksimasi rank rendah, yaitu

$$X_{fg} = X - X_{bg}.$$

Komponen  $X_{fg}$  memuat perubahan lokal yang tidak dapat dijelaskan oleh struktur ber-rank rendah. Setelah dibentuk kembali ke ukuran frame asli, komponen ini menunjukkan area-area yang mengalami perubahan antar-frame, yang berkaitan dengan keberadaan objek bergerak.

Karena foreground diperoleh sebagai hasil selisih terhadap background, warna foreground yang dihasilkan tidak akan sama dengan warna pada video aslinya, melainkan hanya menyoroti bagian-bagian frame yang mengalami perubahan.

#### F. Alur Implementasi

Secara ringkas, tahapan implementasi pemisahan background dan foreground pada video dapat dirangkum sebagai berikut:

- 1) Mengekstraksi frame dari video input.
- 2) Melakukan pra-pemrosesan berupa konversi grayscale dan penyesuaian ukuran frame.

- 3) Membentuk matriks representasi video dari seluruh frame.
- 4) Menerapkan *Singular Value Decomposition* pada matriks video.
- 5) Membentuk aproksimasi rank rendah sebagai background.
- 6) Menghitung residual untuk memperoleh foreground.

#### G. Contoh Alur Implementasi Sederhana

Untuk memperjelas alur implementasi pemisahan background dan foreground menggunakan aproksimasi rank rendah berbasis SVD, diberikan contoh sederhana dengan ukuran data yang kecil.

Misalkan sebuah video terdiri dari  $T = 3$  frame, dan setiap frame memiliki ukuran  $2 \times 2$  piksel. Nilai intensitas piksel pada masing-masing frame ditunjukkan sebagai berikut.

Frame pertama (background saja):

$$F_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Frame kedua (terdapat objek bergerak):

$$F_2 = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$$

Frame ketiga (background kembali):

$$F_3 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

- 1) Pembentukan Matriks Video: Setiap frame diratakan (*flatten*) menjadi vektor satu dimensi. Dengan menyusun setiap frame sebagai baris, diperoleh matriks representasi video

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 4}.$$

Matriks  $X$  memiliki  $T = 3$  baris yang merepresentasikan frame dan  $D = 4$  kolom yang merepresentasikan piksel.

- 2) Dekomposisi dan Aproksimasi Rank Rendah:  
Selanjutnya, dilakukan *Singular Value Decomposition* terhadap matriks  $X$  sehingga diperoleh

$$X = U\Sigma V^T.$$

Dengan mempertahankan satu nilai singular terbesar ( $k = 1$ ), diperoleh aproksimasi rank rendah

$$X_{bg} = U_1 \Sigma_1 V_1^T,$$

yang merepresentasikan struktur dominan data. Dalam konteks video, komponen ini dipandang sebagai estimasi background karena muncul secara konsisten pada seluruh frame.

Hasil aproksimasi ini secara konseptual mendekati

$$X_{bg} \approx \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

### 3) Ekstraksi Foreground:

Foreground diperoleh dengan menghitung selisih antara matriks asli dan hasil aproksimasi rank rendah, yaitu

$$X_{fg} = X - X_{bg}.$$

Secara konseptual, diperoleh

$$X_{fg} \approx \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Komponen ini menunjukkan perubahan lokal yang terjadi pada frame kedua, yang berkaitan dengan keberadaan objek bergerak.

### 4) Pembentukan Kembali Frame:

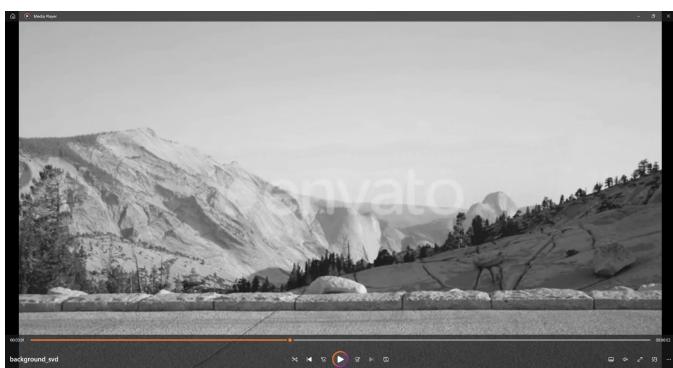
Matriks  $X_{bg}$  dan  $X_{fg}$  kemudian dibentuk kembali ke ukuran frame semula, yaitu  $2 \times 2$ , sehingga diperoleh urutan frame background dan foreground. Dengan demikian, aproksimasi rank rendah digunakan untuk memodelkan background, sedangkan residual digunakan untuk mengekstraksi foreground.

## IV. HASIL DAN PEMBAHASAN

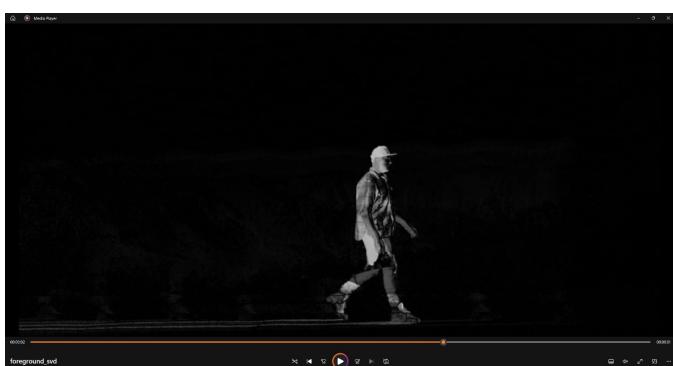
### A. Data Uji

Percobaan dilakukan menggunakan sebuah video yang direkam dengan kamera statis. (seperti ditunjukkan pada Gambar III-A)

### B. Hasil Pemisahan Background dan Foreground



Gambar 9. Hasil Background pada Data



Gambar 10. Hasil Foreground pada Data

Hasil pemisahan menunjukkan bahwa background yang diperoleh bersifat relatif stabil dan konsisten antar-frame, sedangkan foreground menampilkan area-area yang mengalami perubahan. Hal ini menunjukkan bahwa metode yang digunakan mampu memisahkan struktur dominan video dari variasi lokal yang disebabkan oleh objek bergerak.

### C. Analisis Hasil

Berdasarkan hasil percobaan, dapat diamati bahwa pendekatan aproksimasi rank rendah efektif dalam menangkap karakteristik background pada video dengan kamera statis. Background yang dihasilkan tampak lebih halus dibandingkan frame asli, yang mengindikasikan bahwa variasi kecil dan noise turut tereduksi melalui pemilihan rank rendah.

Foreground yang diperoleh berupa citra residual, sehingga tidak mempertahankan warna asli objek pada video. Foreground lebih menonjolkan perubahan intensitas piksel antar-frame, sesuai dengan pemodelan foreground sebagai komponen residual dalam pendekatan ini.

### D. Pengaruh Pemilihan Nilai $k$

Pemilihan nilai  $k$  berpengaruh terhadap kualitas estimasi background. Pada percobaan ini digunakan nilai  $k = 1$  untuk menangkap struktur dominan background dengan asumsi bahwa background video bersifat sangat stabil. Nilai  $k$  yang kecil menghasilkan background yang sederhana, namun dapat mengabaikan variasi background yang lebih kompleks.

Apabila nilai  $k$  diperbesar, variasi background tambahan dapat ikut terakomodasi, namun berpotensi memasukkan sebagian informasi foreground ke dalam komponen background. Oleh karena itu, pemilihan nilai  $k$  perlu disesuaikan dengan karakteristik video yang digunakan.

### E. Keterbatasan Metode

Metode pemisahan background dan foreground berbasis aproksimasi rank rendah memiliki beberapa keterbatasan. Pendekatan ini mengasumsikan kamera statis, background yang relatif stabil, serta objek bergerak yang tidak muncul dalam waktu lama dan tidak mendominasi area citra. Selain itu, perubahan pencahayaan yang signifikan dapat mempengaruhi kualitas hasil pemisahan.

Meskipun demikian, metode ini tetap memberikan gambaran yang jelas mengenai penerapan konsep aljabar linier, khususnya *Singular Value Decomposition* dan aproksimasi rank rendah, dalam permasalahan pemrosesan video.

## V. KESIMPULAN

Pada makalah ini telah dibahas pemisahan background dan foreground pada video menggunakan pendekatan *low-rank approximation* berbasis *Singular Value Decomposition* (SVD). Dengan merepresentasikan video sebagai matriks yang baris-barisnya merepresentasikan frame dan kolom-kolomnya merepresentasikan piksel, struktur dominan video dapat dianalisis menggunakan konsep aljabar linier.

Hasil percobaan menunjukkan bahwa aproksimasi rank rendah dengan mempertahankan sejumlah kecil nilai singular

terbesar mampu memodelkan background video secara efektif pada kondisi kamera statis dan background yang relatif stabil. Foreground diperoleh sebagai komponen residual yang menyoroti area-area perubahan antar-frame, sehingga keberadaan objek bergerak dapat diidentifikasi meskipun warna aslinya tidak dipertahankan.

Pemilihan nilai  $k$  berpengaruh terhadap kualitas hasil pemisahan. Nilai  $k$  yang kecil menghasilkan background yang sederhana dan stabil, namun dapat mengabaikan variasi background yang lebih kompleks. Oleh karena itu, pemilihan nilai  $k$  perlu disesuaikan dengan karakteristik video yang digunakan.

Secara keseluruhan, pendekatan ini menunjukkan bahwa konsep dasar aljabar linier, khususnya SVD dan aproksimasi rank rendah, dapat diterapkan secara langsung untuk menyelesaikan permasalahan pemrosesan video sederhana. Ke depan, pendekatan ini dapat dikembangkan lebih lanjut dengan mempertimbangkan pemilihan nilai  $k$  yang adaptif atau penanganan video dengan background dinamis.



Muhammad Haris Putra Sulastianto

## VI. LAMPIRAN

Github: <https://github.com/mhps-null/makalah-algeo.git>

Video: <https://youtu.be/gWIvOMXuv2k>

## VII. PENUTUP

Tak lupa ucapan terima kasih penulis berikan kepada Tuhan Yang Maha Esa, Institut Teknologi Bandung, Prodi Teknik Informatika, dan dosen mata kuliah IF2123 Aljabar Linier dan Geometri, Bapak Rinaldi Munir. Diharapkan makalah ini dapat bermanfaat bagi pembaca.

## REFERENSI

- [1] R. Munir, “Singular value decomposition (svd) (bagian 1).” <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-21-Singular-value-decomposition-Bagian1-2025.pdf>, 2025. Lecture notes, accessed Dec. 4, 2025.
- [2] R. Munir, “Singular value decomposition (svd) (bagian 2).” <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-22-Singular-value-decomposition-Bagian2-2025.pdf>, 2025. Lecture notes, accessed Dec. 4, 2025.
- [3] 3Blue1Brown, “Essence of linear algebra.” [https://youtube.com/playlist?list=PLZHQQbOWTQDPD3MizzM2xVFitgF8hE\\_ab](https://youtube.com/playlist?list=PLZHQQbOWTQDPD3MizzM2xVFitgF8hE_ab), 2025. YouTube playlist, accessed Dec. 10, 2025.
- [4] sudgylacmoe, “A swift introduction to geometric algebra.” [https://youtu.be/60z\\_hpEAfD8](https://youtu.be/60z_hpEAfD8), 2020. YouTube video, accessed Dec. 10, 2025.
- [5] V. Kernel, “Svd visualized, singular value decomposition explained — see matrix , chapter 3.” <https://www.youtube.com/watch?v=vSczTbgc8Rc>, 2022. YouTube video, accessed Dec. 12, 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, 24 Desember 2025