



## Konversi Citra RGB Ke Grayscale Menggunakan OpenCV

Ibnu Suhada<sup>1</sup>, Ade Ermawati<sup>1\*</sup>, Muhammad Paris Rifandie Maulidhan<sup>1</sup>, Perani Rosyani<sup>2</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Kota Tangerang Selatan, Indonesia

Email : <sup>1\*</sup>[akuadeermawati@gmail.com](mailto:akuadeermawati@gmail.com), <sup>2</sup>[dosen0083@unpam.ac.id](mailto:dosen0083@unpam.ac.id)

(\* : coressponding author)

**Abstrak** - Program ini bertujuan untuk melakukan konversi citra RGB (Red, Green, Blue) menjadi citra grayscale menggunakan library OpenCV pada Python. Proses konversi dilakukan dengan memanggil fungsi cv2.cvtColor() yang tersedia pada OpenCV. Citra grayscale yang dihasilkan kemudian disimpan ke dalam file dengan nama yang sama menggunakan fungsi cv2.imwrite(). Program juga menampilkan citra asli dan citra grayscale menggunakan fungsi cv2.imshow() sebelum akhirnya menutup semua jendela yang terbuka.

**Kata Kunci :** Pengolahan Citra Digital, Efisiensi Komputasi, OpenCV, RGB

**Abstract** - This program aims to convert RGB (Red, Green, Blue) images into grayscale images using the OpenCV library in Python. The conversion process is carried out by calling the cv2.cvtColor() function available in OpenCV. The resulting grayscale image is then saved into a file with the same name using the cv2.imwrite() function. The program also displays the original image and a grayscale image using the cv2.imshow() function before finally closing all open windows.

**Keywords:** Digital Image Processing, Computing Efficiency, OpenCV, RGB

## 1. PENDAHULUAN

Pengolahan citra digital (*digital image processing*) merupakan salah satu cabang ilmu komputer yang memiliki banyak aplikasi dalam berbagai bidang, seperti pengenalan pola, analisis citra medis, dan pengolahan video. Salah satu langkah awal yang sering dilakukan dalam pengolahan citra adalah konversi citra dari ruang warna RGB (Red, Green, Blue) ke citra grayscale. Citra grayscale, yang hanya memiliki satu komponen warna (*intensitas*), memiliki beberapa keuntungan dibandingkan dengan citra RGB. Antara lain, ukuran file yang lebih kecil dan proses komputasi yang lebih efisien. Hal ini karena citra grayscale hanya memerlukan satu nilai untuk setiap piksel, berbeda dengan citra RGB yang membutuhkan tiga nilai (R, G, dan B) untuk setiap piksel.

Konversi citra RGB ke grayscale dapat dilakukan dengan berbagai metode, salah satunya adalah dengan menghitung rata-rata dari ketiga komponen warna (R, G, dan B) atau menggunakan formula tertentu yang memberikan bobot berbeda untuk masing-masing komponen. Penggunaan formula yang tepat dapat menghasilkan citra grayscale yang sesuai dengan persepsi visual manusia.

Dalam penelitian ini, kami mengembangkan sebuah program yang mengimplementasikan konversi citra RGB ke grayscale menggunakan library OpenCV pada Python. OpenCV (Open Source Computer Vision Library) adalah salah satu library open-source yang banyak digunakan dalam pengolahan citra digital dan computer vision. Fungsi-fungsi yang disediakan oleh OpenCV, seperti cv2.imread(), cv2.cvtColor(), dan cv2.imwrite(), memudahkan proses konversi citra RGB ke grayscale. Tujuan dari penelitian ini adalah untuk memahami dan mengimplementasikan proses konversi citra RGB ke grayscale menggunakan OpenCV pada Python. Hasil dari program ini diharapkan dapat menjadi dasar untuk pengembangan aplikasi pengolahan citra digital lebih lanjut.

## 2. DASAR TEORI

### 2.1 Citra Digital

Citra digital adalah representasi visual dari objek atau scene dalam bentuk numerik yang dapat diproses oleh komputer. Citra digital terdiri dari kumpulan piksel yang masing-masing memiliki nilai intensitas tertentu.



## 2.2 Ruang Warna RGB

RGB (Red, Green, Blue) adalah salah satu ruang warna yang umum digunakan untuk merepresentasikan citra digital. Setiap piksel dalam citra RGB memiliki tiga komponen: nilai merah (R), nilai hijau (G), dan nilai biru (B). Kombinasi nilai R, G, dan B dapat menghasilkan berbagai macam warna.

## 2.3 Citra Grayscale

Citra grayscale adalah citra digital yang hanya memiliki satu komponen warna, yaitu intensitas. Nilai intensitas pada citra grayscale biasanya berkisar antara 0 (hitam) hingga 255 (putih). Konversi dari citra RGB ke citra grayscale dapat dilakukan dengan menghitung rata-rata dari ketiga komponen warna (R, G, dan B) atau dengan menggunakan formula tertentu.

## 2.4 Konversi Citra RGB ke Grayscale

Salah satu formula yang umum digunakan untuk konversi citra RGB ke grayscale adalah:

$$\text{Grayscale} = 0.299 * R + 0.587 * G + 0.114 * B$$

Formula ini memberikan bobot yang berbeda untuk setiap komponen warna, sesuai dengan sensitivitas mata manusia terhadap warna-warna tersebut.

## 2.5 OpenCV (*Open Source Computer Vision Library*)

OpenCV adalah library open-source yang digunakan untuk pengolahan citra digital dan computer vision. OpenCV menyediakan fungsi-fungsi yang memudahkan dalam melakukan operasi pada citra digital, seperti membaca, menulis, dan mengkonversi citra. Fungsi cv2.cvtColor() pada OpenCV dapat digunakan untuk mengkonversi citra RGB menjadi citra grayscale.

# 3. METODE

## 3.1 Persiapan Lingkungan Pengembangan

Memastikan Python dan library OpenCV telah terinstal dengan benar pada sistem. Menyiapkan file citra RGB yang akan dikonversi, dalam contoh ini adalah file 'ibnu.jpeg'.

## 3.2 Membaca Citra RGB dari File

Menggunakan fungsi cv2.imread() dari OpenCV untuk membaca file citra RGB 'ibnu.jpeg'. Fungsi cv2.imread() akan mengembalikan objek citra dalam bentuk NumPy array.

## 3.3 Konversi Citra RGB ke Grayscale

Menggunakan fungsi cv2.cvtColor() dari OpenCV untuk mengonversi citra RGB menjadi citra grayscale. Fungsi cv2.cvtColor() memerlukan dua argumen: objek citra dan kode konversi warna, dalam hal ini cv2.COLOR\_BGR2GRAY. Hasil konversi disimpan dalam variabel gray\_image.

## 3.4 Menyimpan Citra Grayscale ke File

Menggunakan fungsi cv2.imwrite() dari OpenCV untuk menyimpan citra grayscale ke file dengan nama yang sama, yaitu 'ibnu.jpeg'.

## 3.5 Menampilkan Citra Asli dan Citra Grayscale

Menggunakan fungsi cv2.imshow() dari OpenCV untuk menampilkan citra asli (RGB) dan citra grayscale. Fungsi cv2.imshow() memerlukan dua argumen: judul jendela dan objek citra.

## 3.6 Menunggu Input Pengguna dan Menutup Jendela

Menggunakan fungsi cv2.waitKey(0) untuk menunggu hingga pengguna menekan sembarang tombol. Menggunakan fungsi cv2.destroyAllWindows() untuk menutup semua jendela yang telah dibuka.



## 4. UJI COBA DAN EVALUASI

### 4.1 Uji Program Konversi Citra RGB Ke Grayscale

Untuk menguji program konversi citra RGB ke grayscale, dilakukan beberapa skenario uji coba sebagai berikut:

- a. Uji Coba 1: Konversi Citra RGB Sederhana
  - 1) Input: File citra RGB 'ibnu.jpeg'
  - 2) Harapan: Program dapat membaca citra RGB, mengonversinya ke grayscale, menyimpan citra grayscale, dan menampilkan kedua citra.
  - 3) Hasil: Program berhasil menjalankan semua tahapan sesuai harapan. Citra grayscale yang dihasilkan terlihat sesuai dengan persepsi visual.
- b. Uji Coba 2: Konversi Citra RGB Kompleks
  - 1) Input: File citra RGB dengan ukuran besar, tekstur kompleks, dan variasi warna yang beragam.
  - 2) Harapan: Program dapat menangani citra RGB yang lebih kompleks dan menghasilkan citra grayscale yang akurat.
  - 3) Hasil: Program dapat mengonversi citra RGB kompleks dengan baik. Citra grayscale yang dihasilkan tetap mempertahankan fitur-fitur penting dari citra asli.
- c. Uji Coba 3: Penanganan Kesalahan
  - 1) Input: File citra RGB yang tidak ditemukan atau tidak dapat dibuka.
  - 2) Harapan: Program dapat menangani kesalahan dengan memberikan pesan error yang jelas.
  - 3) Hasil: Program belum menangani kesalahan dengan baik. Jika file citra tidak ditemukan, program akan langsung crash dan menghasilkan pesan error yang tidak informatif bagi pengguna.

### 4.2 Evaluasi

Secara keseluruhan, program konversi citra RGB ke grayscale menggunakan OpenCV telah berjalan dengan baik. Program dapat membaca citra RGB, mengonversi ke grayscale, menyimpan citra grayscale, dan menampilkan kedua citra dengan benar. Citra grayscale yang dihasilkan juga sesuai dengan persepsi visual manusia.

Namun, program masih memiliki kekurangan dalam hal penanganan kesalahan. Jika terjadi kesalahan, misalnya file citra tidak ditemukan, program belum memberikan pesan error yang jelas. Hal ini perlu diperbaiki untuk meningkatkan keandalan program.

Selain itu, program juga belum menyediakan fitur-fitur tambahan, seperti opsi untuk mengatur parameter konversi warna atau melakukan operasi lain pada citra. Pengembangan lebih lanjut dapat dilakukan untuk memperkaya fungsionalitas program.

## 5. KESIMPULAN

### 5.1 Kesimpulan

Program konversi citra RGB ke grayscale menggunakan OpenCV yang telah dikembangkan berhasil menjalankan proses konversi citra dengan baik. Program dapat membaca citra RGB dari file, mengonversi ke citra grayscale, menyimpan citra grayscale, dan menampilkan citra asli dan citra grayscale.

Dasar teori yang mendasari program ini mencakup pemahaman tentang citra digital, ruang warna RGB, citra grayscale, serta prinsip konversi citra RGB ke grayscale. Penggunaan library



OpenCV pada Python memudahkan implementasi program dengan menyediakan fungsi-fungsi yang diperlukan, seperti cv2.imread(), cv2.cvtColor(), cv2.imwrite(), dan cv2.imshow().

Hasil uji coba menunjukkan program dapat menangani citra RGB sederhana maupun kompleks dengan baik. Citra grayscale yang dihasilkan sesuai dengan persepsi visual manusia. Namun, program masih memiliki kekurangan dalam hal penanganan kesalahan, seperti ketika file citra tidak ditemukan.

Secara keseluruhan, program ini dapat digunakan sebagai dasar untuk pengembangan aplikasi pengolahan citra digital lebih lanjut. Pengembangan lebih lanjut dapat dilakukan untuk menambahkan fitur-fitur tambahan, seperti opsi pengaturan parameter konversi warna, serta peningkatan kemampuan penanganan kesalahan.

## **5.2 Saran**

### a. Peningkatan Penanganan Kesalahan

Program saat ini belum menangani kesalahan dengan baik, seperti ketika file citra tidak ditemukan. Disarankan untuk menambahkan blok try-except yang dapat menangkap dan menampilkan pesan error yang jelas bagi pengguna.

### b. Penambahan Fitur Kustomisasi

Program saat ini hanya melakukan konversi citra RGB ke grayscale menggunakan formula baku. Disarankan untuk menambahkan opsi kustomisasi, seperti kemampuan untuk mengatur parameter konversi warna atau melakukan operasi lain pada citra (misalnya, penyesuaian kontras, perbaikan kualitas citra, dll.).

### c. Integrasi dengan Aplikasi atau Sistem yang Lebih Besar

Program ini dapat diintegrasikan dengan aplikasi atau sistem pengolahan citra digital yang lebih besar, seperti aplikasi pengenalan pola, analisis citra medis, atau aplikasi computer vision lainnya. Integrasi ini dapat meningkatkan manfaat dan kegunaan program dalam lingkup yang lebih luas.

### d. Pengujian yang Lebih Komprehensif

Uji coba yang dilakukan saat ini masih terbatas pada skenario sederhana dan citra kompleks. Disarankan untuk melakukan pengujian yang lebih komprehensif, termasuk pengujian pada dataset citra yang lebih beragam, serta pengujian pada sistem operasi dan perangkat yang berbeda.

### e. Dokumentasi dan Kemudahan Penggunaan

Untuk memudahkan penggunaan dan pengembangan lebih lanjut, program sebaiknya dilengkapi dengan dokumentasi yang jelas, termasuk petunjuk penggunaan, contoh kode, dan penjelasan API yang digunakan. Hal ini akan membantu pengembang lain yang ingin memanfaatkan atau memperluas program ini.

## **DAFTAR PUSTAKA**

- OpenCV-Python Tutorials. (n.d.). Accessed on: Jun 27, 2024. [Online]. Available: <https://opencv-python-tutorial.readthedocs.io/en/latest/>
- Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.
- Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Prentice Hall.
- Jain, A. K. (1989). Fundamentals of Digital Image Processing. Prentice Hall.
- Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.
- Shapiro, L. G., & Stockman, G. C. (2001). Computer Vision. Prentice Hall.
- Pratt, W. K. (2007). Digital Image Processing: PIKS Inside (4th ed.). Wiley-Interscience.
- Parker, J. R. (2010). Algorithms for Image Processing and Computer Vision (2nd ed.). Wiley.
- Sonka, M., Hlavac, V., & Boyle, R. (2014). Image Processing, Analysis, and Machine Vision (4th ed.). Cengage Learning.
- Forsyth, D. A., & Ponce, J. (2011). Computer Vision: A Modern Approach (2nd ed.). Pearson.



**JRIIN : Jurnal Riset Informatika dan Inovasi**

**Volume 2, No. 3, Agustus Tahun 2024**

**ISSN 3025-0919 (media online)**

**Hal 390-394**

Burger, W., & Burge, M. J. (2016). Principles of Digital Image Processing: Core Algorithms. Springer.  
Ballard, D. H., & Brown, C. M. (1982). Computer Vision. Prentice Hall.