

**LAPORAN TUGAS BESAR**  
**TF 4012 PENGUKURAN BERBASIS CITRA**

**PENGOLAHAN CITRA UNTUK PEMOTRETAN**  
**MODE MALAM**



**Disusun Oleh:**

<b>Eraraya Ricardo Muten</b>	<b>13316082</b>	<b>Konseptor, Programmer, Fotografer Penulis Makalah</b>
<b>P.C Astuto Santosa</b>	<b>13316085</b>	<b>Konseptor, Komentator, Fotografer, Penulis Makalah</b>
<b>Ferio Brahmana</b>	<b>13317032</b>	<b>Konseptor, Komentator, Penulis Makalah</b>

**PROGRAM STUDI TEKNIK FISIKA**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**INSTITUT TEKNOLOGI BANDUNG**

**2020**

## I. Latar Belakang

Salah satu tanda kemajuan zaman, khususnya dalam bidang pengolahan citra, adalah munculnya teknologi pemotretan mode malam (*night mode*). Seperti yang telah diketahui, dengan cahaya yang sedikit citra yang terbentuk lewat pemotretan tidak akan maksimal kualitasnya karena sensor-sensor kamera tidak bisa bekerja dengan apik. Namun, dengan hadirnya mode malam, seseorang bisa mendapatkan kualitas citra yang sama baiknya dengan pemotretan pada lingkungan dengan cahaya yang cukup. Hal ini dapat dicapai berkat adanya kemajuan pada proses pengolahan citra menggunakan proses matematika dan komputasi.

Jika diperjelas, secara fisis, masalah pada perolehan citra yang didapat pada lingkungan sedikit, sebagian besar, didapat dari perangkat keras alat optik. Umumnya, pada pengambilan citra di lingkungan gelap, pengamat perlu meningkatkan sensitivitas dari sensor (nilai ISO) dan waktu pengambilan gambar. Kedua efek tersebut secara simultan akan menyebabkan sensor-sensor berlaku tidak wajar sehingga ada gambar-gambar terang yang tidak diinginkan pada citra (baik dari naiknya efek derau, ada anomali sensor, maupun tertangkapnya cahaya-cahaya *artificial*).

Lalu, karena perlunya biaya tinggi untuk meningkatkan kualitas perangkat keras, ada beberapa upaya untuk melakukan pemrosesan peningkatan kualitas citra lewat perangkat lunak saja. Metode yang berorientasi pada pengolahan sinyal dan komputasi ini tengah berkembang pesat mulai dari pengolahan sinyal sederhana (aljabar linear, *filtering*, transformasi geometri, dan sebagainya) sampai pengolahan menggunakan metode statistik maju (pembelajaran mesin, *deep learning*, kecerdasan buatan, dan lain-lain). Oleh karena itu, berdasarkan pengetahuan penulis terkait dengan pengolahan citra baik yang berasal dari kuliah maupun sumber lainnya, penulis ingin mencoba untuk ikut memberikan kontribusi terkait pengolahan citra untuk mode malam ini.

## II. Metodologi

Untuk memperoleh pengolahan mode malam lewat perangkat lunak, perlu dipahami terlebih dahulu bahwa efek derau akan memunculkan efek derau *gaussian* pada citra seperti pada **Gambar 1** (seolah-olah ada butir pasir pada citra). Efek gangguan *gaussian* tersebut akan selalu ada pada pengambilan gambar yang mengandalkan sedikit cahaya. Selain gangguan tersebut, tentunya citra juga akan sulit nampak jelas (baik tidak terlihat maupun *blur* karena gelap) oleh mata telanjang karena tidak memiliki cahaya yang cukup. Terakhir, untuk kepentingan pengolahan mode malam, pengambilan gambar secara berkali-kali dalam waktu sangat singkat (*burst shots*) juga diperlukan. Sayangnya, sering kali terjadi pergeseran/gangguan akan kamera sehingga citra

yang diambil akan mengalami, setidaknya, sedikit pergeseran. Oleh karena itu, tiga hal utama yang perlu dikejar dari algoritma pengolahan mode malam adalah pemrosesan untuk menyesuaikan keselarasan dari tiap hasil *burst shots*, menggunakan citra-citra hasil tangkapan untuk menghilangkan derau *gaussian*, dan mampu memberikan citra yang tampak lebih jelas (lebih terang secara tampak maupun lebih presisi representasi objeknya).



**Gambar 1.** Citra Hasil Tangkapan pada Lingkungan dengan Pencahayaan Rendah

Dalam mencapai tujuan tersebut, algoritma mode malam yang diajukan perlu untuk melaksanakan beberapa tahap. Tahap pertama perlu ada perubahan kontras untuk menghasilkan citra yang lebih terang dan/atau seragam. Berikutnya, seperti yang telah disebutkan sebelumnya, citra yang diambil sering kali mengalami pergeseran antar satu sama lain sehingga perlu mengajukan algoritma yang bisa mendeteksi pergeseran dan mengkompensasi pergeseran ini (tahap ini biasa disebut sebagai tahap *image registration*). Algoritma *high dynamic range* (HDR) digunakan untuk meningkatkan *brightness* dari citra tanpa mengorbankan *shadow* dan *detail*. Terakhir, untuk mengurangi derau merica dan garam, algoritma filter *Gaussian* digunakan pada citra. Setiap metode akan dijelaskan lebih rinci dalam subbab berikutnya. Penulis menggunakan kerangka kerja pemrograman OpenCV [1] dan Scikit-Image [2] yang berbasis bahasa pemrograman Python.

## **2.1. Perubahan Format Citra (Perubahan Kanal Warna) dan *Histogram Equalization***

Pertama, citra hasil *burst short* seluruhnya perlu disesuaikan kontrasnya sebelum dilewatkan pada pengolahan lainnya untuk mendapatkan gambar yang cukup terang dan jelas. Dalam mewujudkan proses perubahan kontras ini, penulis lalu melewatkan citra kepada algoritma *histogram equalization* yang berfungsi untuk melinearisasi histogram kumulatif dari citra. Namun, sebelum melewatkan citra pada algoritma *histogram equalization*, citra berwarna yang ditangkap (RGB) terlebih dahulu perlu untuk diubah menjadi citra skala keabuan (*greyscale*) sebab prinsip kerja *histogram equalization* hanya akan bekerja pada skala keabuan.

Jadi, pertama, penulis melewati citra pada algoritma untuk merubah citra RGB ke *grayscale* dengan persamaan matematik seperti pada persamaan (1). Untuk merubah format RGB ke *grayscale*, pada dasarnya perlu dipahami terlebih dahulu bahwa persepsi manusia dalam menangkap warna merah, hijau, dan biru tidaklah sama. Mata manusia membobot tiap warna (akibat perbedaan panjang gelombang ataupun alasan psikologis) secara berbeda. Oleh karena itu, koefisien pada representasi warna RGB pada intensitas di skala keabuan tidak seragam, tetapi sesuai pada persamaan (1) sebagai model konversi yang digunakan oleh PAL dan NTSC berdasarkan ITU-R Recommendation BT.601 [3][4].

$$GS(x, y) = 0.299 \cdot R(x, y) + 0.587 \cdot G(x, y) + 0.114 \cdot B(x, y) \quad (1)$$

Lalu, setelah dilewatkan pada *converter* peta warna, citra dalam skala keabuan akan dilewatkan pada algoritma *histogram equalization*. Algoritma ini dibuat sendiri oleh penulis dan dibangun berdasarkan dasar yang terdapat pada referensi kuliah [5]. Formula algoritma tersebut dapat direpresentasikan dengan persamaan matematis seperti pada persamaan (2). Seperti yang disebutkan sebelumnya, citra yang dilewatkan pada algoritma ini akan berubah kontrasnya dan dapat dilihat pada **Gambar 2**.

$$h(v) = \text{round} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right) \quad (2)$$

dengan  $v$  merupakan intensitas suatu piksel pada citra masukan,  $h(v)$  adalah fungsi yang akan memetakan piksel citra masukan  $v$  ke piksel citra keluaran,  $cdf$  merupakan fungsi histogram kumulatif citra masukan,  $L - 1$  merupakan batas atas intensitas piksel (255).



**Gambar 2.** Citra yang telah Diubah Menjadi Keabuan dan Melewati Algoritma *Histogram Equalization*

## 2.2. Transformasi Fourier dan Korelasi

Selanjutnya, setelah tiap gambar hasil *burst shots* ditala kontrasnya lewat *histogram equalization*, seluruh gambar perlu untuk disesuaikan secara geometri. Sebab, seluruh citra

tersebut diperlukan untuk HDR dan keseluruhannya sebisa mungkin simetris. Oleh karena itu, dengan sulitnya menghindari tidak adanya pergeseran pada pengambilan citra oleh pemotret, perlu pengolahan lewat perangkat lunak untuk memastikan bentuk simetris dari tiap foto.

Dalam melakukan deteksi pergeseran, penulis menggunakan prinsip pengolahan sinyal citra, spesifiknya, korelasi antar dua citra. Pertama, perlu dipilih satu citra yang digunakan sebagai citra referensi ( $g_a$ ). Citra berikutnya ( $g_b$ ) perlu untuk dilihat seberapa jauh pergeserannya dibandingkan dengan citra referensi. Namun, pekerjaan ini akan sulit jika ditinjau pada domain ruang – yang diwakilkan oleh intensitas pada tiap pixel dalam ruang dua dimensi. Karena itu, pergeseran citra akan dilihat berdasarkan korelasi antar citra dalam domain frekuensi dengan menggunakan prinsip Transformasi Fourier 2-Dimensi.

Hasil Transformasi Fourier dari kedua citra,  $\mathbf{G}_a = \mathcal{F}\{g_a\}$ ,  $\mathbf{G}_b = \mathcal{F}\{g_b\}$ , kemudian digunakan untuk menghitung *cross-power spectrum*, yaitu mengalikan konjugat kompleks dari  $\mathbf{G}_b$  dengan  $\mathbf{G}_a$  kemudian menormalisasi hasil perkaliannya. Perkalian dan normalisasi ini dilakukan secara *elementwise*, atau secara matematis untuk elemen pada koordinat (x, y) dapat dituliskan sebagai berikut

$$R_{xy} = \frac{G_{a,xy} \cdot G_{b,xy}^*}{|G_{a,xy} \cdot G_{b,xy}^*|} \quad (3)$$

Setelah perkalian ini dilakukan untuk setiap elemen, matriks  $R$  yang diperoleh kemudian ditransformasi Fourier balik,  $r = \mathcal{F}^{-1}\{R\}$ , untuk mendapatkan matriks *cross-correlation* ternormalisasi  $r$ . Pergeseran citra kedua ( $g_b$ ) terhadap citra referensi ( $g_a$ ), didapatkan dari koordinat piksel pada matriks  $r$  dengan nilai tertinggi,  $(\Delta x, \Delta y) = \arg \max_{(x,y)} \{r\}$ . Secara ilustratif dapat dilihat pada **Gambar 3**.



**Gambar 3.** Citra Referensi untuk Korelasi, Citra Offset, dan Hasil *Cross-correlation*-nya

### 2.3. Transformasi Geometri Citra

Berdasarkan hasil Transformasi Fourier dan korelasi, nilai pergeseran yang diperlukan telah didapatkan. Berikutnya, yang diperlukan adalah metode untuk menggeser citra-citra tersebut. Dengan asumsi bahwa pergeseran yang terjadi hanya pada komponen dua dimensi secara linear, pergeseran yang diperlukan hanya transformasi linear sejauh nilai pergeseran yang telah diperoleh. Representasi matematis dari transformasi ini dapat dilihat pada persamaan (4). Untuk algoritma yang digunakan, penulis merujuk pada algoritma *Euclidean transformation* yang dapat diakses secara umum [5]

$$g(x'_i, y'_i) = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = g(x_i, y_i) + T[\Delta x, \Delta y] = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (4)$$

yang mana  $g(x_i, y_i)$  adalah citra tinjauan,  $T[\Delta x, \Delta y]$  adalah matriks transformasi yang menggeser nilai pixel dari tiap komponen ke -  $i$  citra sebesar  $\Delta x$  dan  $\Delta y$  untuk tiap komponen  $x$  dan  $y$  dari citra. Hasil citra yang telah ditransformasi dinotasikan dengan  $g(x'_i, y'_i)$ .

Citra hasil *histogram equalization* hanya digunakan untuk mendapatkan nilai  $\Delta x$  dan  $\Delta y$  pada tahap Transformasi Fourier dan Korelasi. Citra yang kemudian digunakan dalam tahap ini adalah citra RGB asli yang tidak melalui tahap konversi ke skala keabuan dan *histogram equalization*.



**Gambar 4.** Citra Referensi untuk Transformasi Linear, Citra Offset, dan Citra Offset yang telah ditransformasi

## 2.4. HDR (*High Dynamic Range*)

HDR adalah teknik pada pencitraan atau fotografi untuk mendapatkan hasil dengan rentang luminositas dinamis yang jauh lebih besar dibandingkan teknik fotografi standar. Pada dasarnya mata manusia dapat menyesuaikan dengan berbagai kondisi cahaya, tapi sebagian besar perangkat pencitraan terbatas menggunakan 8-bit untuk tiap kanal, yang seringkali menyebabkan kondisi *overexposed* atau *underexposed*.

Penulis menggunakan teknik *Exposure Fusion* yang diusulkan oleh Mertens et al [6]. Citra keluaran  $O$  merupakan penjumlahan terbobot dari  $N$  citra masukan  $I_k$  (dengan *exposure time* yang berbeda - beda) dengan persamaan sebagai berikut

$$O_{xy} = \sum_{k=1}^N \hat{W}_{xy,k} I_{xy,k} \quad (5)$$

$$\hat{W}_{xy,k} = \left[ \sum_{k'=1}^N W_{xy,k'} \right]^{-1} W_{xy,k} \quad (6)$$

$$W_{xy,k} = (C_{xy,k})^{\omega_C} \times (S_{xy,k})^{\omega_S} \times (E_{xy,k})^{\omega_E} \quad (7)$$

$$F(x, y) = \left( -\frac{(i_{xy} - 0.5)^2}{2\sigma^2} \right) \quad (8)$$

dengan  $(x, y)$  adalah koordinat piksel,  $\omega_C$ ,  $\omega_S$ ,  $\omega_E$  adalah konstanta pembobotan (pada makalah ini digunakan  $\omega_C = 0.8$ ,  $\omega_S = 0.8$ ,  $\omega_E = 0.5$ ),  $C$  adalah matriks nilai absolut hasil filter Laplacian pada citra masukan yang telah dikonversi ke skala keabuan,  $S$  adalah matriks yang tiap elemennya merupakan standar deviasi dari ketiga kanal warna untuk tiap piksel dengan koordinat yang sama dengan elemen tersebut, dan  $E$  adalah matriks yang mengukur seberapa baik *exposure* dari citra. Tiap elemen dengan koordinat  $(x, y)$  pada matriks  $E$  ini dihitung dengan perkalian tiga fungsi Gaussian pada persamaan (8) (satu untuk tiap kanal), dengan  $\sigma = 0.2$  dan  $i_{xy}$  adalah intensitas ternormalisasi ([0, 1]) pada koordinat  $(x, y)$ . Hasil proses ini dapat dilihat pada Bab III.

### 2.5. Filter Gaussian

Setelah melalui HDR, citra dilewatkan lagi pada sebuah filter *Gaussian* dengan  $\sigma = 1.0$  untuk mengurangi derau merica dan garam akibat *sensor burning* ketika pengambilan gambar dengan ISO yang cukup tinggi dan ketidak sempurnaan proses *image registration*. Hasil proses ini dapat dilihat pada Bab III.

## III. Hasil dan Analisis

Pencahayaan luar dan dalam ruangan umumnya berbeda sehingga untuk pengujian algoritma digunakan dua objek foto berbeda. Yang pertama adalah objek kebun (luar ruangan) dan objek mainan mobil (dalam ruangan).

### 3.1. Kebun

Seluruh foto untuk objek ini diambil dengan kamera telepon genggam Samsung Galaxy Note9. Citra masukan yang digunakan diambil menggunakan mode manual dengan ISO 200, f/1.5, 4.3 mm, dan tiga nilai *exposure time* yang berbeda yaitu 2 detik, 1/2 detik, dan 1/4 detik. Ketiga citra masukan ini diproses dengan algoritma mode malam sebagaimana dijabarkan pada Bab II,

kemudian citra keluarannya (**Gambar 6**) dibandingkan dengan citra hasil mode *automatic* yang disediakan Samsung (**Gambar 5**).



**Gambar 5.** Hasil Mode Automatic



**Gambar 6.** Hasil Pemrosesan Algoritma Mode Malam

- Detail dari objek – objek seperti rumput dan daun lebih baik kualitasnya pada citra hasil algoritma mode malam (kanan). Citra hasil mode *automatic* terlihat lebih berderau. Citra hasil mode malam terlihat jauh lebih sedikit deraunya namun sedikit lebih halus.



**Gambar 7.** Objek Rumput Mode Automatic



**Gambar 8.** Objek Rumput Algoritma Mode Malam

- Terdapat bagian pohon yang sangat gelap dan tidak terlihat pada citra hasil mode *automatic* (kiri), sedangkan bagian pohon ini dapat terlihat dengan jelas pada citra hasil algoritma mode malam (kanan).





**Gambar 9.** Objek Pohon Mode *Automatic*



**Gambar 10.** Objek Pohon Algoritma Mode Malam

- Pada bagian *highlight*, citra hasil algoritma malam (kanan) masih berhasil menangkap tulisan huruf “Y” yang merupakan penanda blok rumah tetangga. Sedangkan pada citra hasil mode *automatic*, detail ini hilang sepenuhnya.



**Gambar 11.** Objek Huruf Y Mode *Automatic*



**Gambar 12.** Objek Huruf Y Algoritma Mode Malam

Dari poin – poin ini, terlihat kalau algoritma mode malam yang digunakan berhasil mempertahankan *dynamic range* pada citra, bagian terang tidak mengalami *overexposed* dan bagian gelap tidak mengalami *underexposed*. Ini berkat penggabungan beberapa citra dengan *exposure time* yang berbeda pada tahap HDR. *Image registration* yang dilakukan oleh algoritma mode malam juga sangat baik dalam mengatasi pergeseran translasi akibat getaran tangan ketika mengambil beberapa citra. Hal ini terlihat dari tidak berbayangnya citra hasil algoritma mode malam meskipun merupakan citra hasil penggabungan beberapa citra berbeda. Tahap filter *Gaussian* dalam algoritma mode malam juga berhasil mengatasi derau pada citra, meskipun kompensasinya adalah citra menjadi sedikit lebih halus.

### **3.2. Mainan Mobil**

Objek mobil mainan ini diambil dengan menggunakan kamera *mirrorless* Sony a6000. Pengaturan untuk ISO dan *aperture* atau kondisi bukaan berlaku tetap untuk seluruh sesi pengambilan citra, yaitu 10.000 dan f/6.3. Kondisi yang berubah dalam setiap pengambilan citra masukan untuk algoritma mode malam adalah *exposure time* yaitu sebesar 1/10 detik, 1/40 detik, 1/160 detik. Pencahayaan dalam ruangan cenderung *uniform* sehingga tidak banyak *highlight* maupun *shadow*.



**Gambar 13.** Hasil Pemrosesan Algoritma Mode Malam

Jika dibandingkan dengan keadaan yang mendekati sesungguhnya (mata telanjang) pada **Gambar 1**, terlihat algoritma mode malam memberikan peningkatan kualitas yang sangat baik. Objek menjadi jelas terlihat dan derau jauh berkurang. Sedikit getaran translasi yang terjadi saat pengambilan citra juga teratasi dengan baik, terlihat dari objek mainan mobil yang tidak berbayang/blur.

#### IV. Kesimpulan

Algoritma mode malam dengan serangkaian tahapan yang diajukan pada makalah ini telah berhasil meningkatkan kualitas citra jika dibandingkan dengan mode *automatic*. Tahapan *image registration* (Transformasi Fourier, Korelasi, dan Transformasi Geometri) berhasil mengatasi getaran translasi yang terjadi saat proses pengambilan citra. Algoritma *Exposure Fusion* pada tahap HDR berhasil mempertahankan detail pada *highlight* dan *shadow*. Tahap pemfilteran dengan filter *Gaussian* berhasil mengurangi derau dengan cukup signifikan meski citra menjadi sedikit lebih halus.

#### V. Referensi

- [1] OpenCV. Open Source Computer Vision Library 2015.
- [2] Van Der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. Scikit-image: Image processing in python. PeerJ 2014;2014. <https://doi.org/10.7717/peerj.453>.
- [3] BT.601: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. ITU; n.d.
- [4] Palus H. Representations of colour images in different colour spaces. Colour Image Process. Handb., Springer US; 1998, hal. 67–90. [https://doi.org/10.1007/978-1-4615-5779-1\\_4](https://doi.org/10.1007/978-1-4615-5779-1_4).
- [5] Solomon C. Fundamentals of Digital Image Processing A Practical Approach with Examples in Matlab. Chichester UK: John Wiley & Sons Ltd; 2010.
- [6] Mertens T, Kautz J, Reeth F Van. Exposure Fusion. 15th Pacific Conf Comput Graph Appl 2007:382–90.