

Identifikasi Obyek Bergerak Berbasis Segmentasi Warna RGB

Risna Julia Sari¹, Achmad Fanany Onnilita Gaffar², Arief Bramanto Wicaksono Putra³
Teknologi Informasi, Teknik Informatika Multimedia, Politeknik Negeri Samarinda, Kalimantan Timur
Email: julia.cabbia20@gmail.com, onnygaffar212@gmail.com, ariefbram@gmail.com

Abstract—Penelitian ini menerapkan metode segmentasi warna berbasis komponen RGB untuk mengidentifikasi obyek bergerak. Metode kuantisasi berbasis *multi-thresholding* yang diterapkan di tiap komponen RGB digunakan untuk memisahkan pola obyek dalam citra. Teknik *adjustment* digunakan untuk mengatur *contrast* dan *brightness* dari citra yang sudah terkuantisasi. Segmentasi warna RGB dilakukan dengan memperkurangkan antar komponen R, G dan B. Terdapat 6 (enam) kemungkinan segmentasi warna yaitu segmentasi R-G, R-B, G-R, G-B, B-R dan B-G. Dari hasil percobaan yang dilakukan, diperoleh segmentasi terbaik untuk mendeteksi obyek bergerak adalah segmentasi R-B untuk dominan *frame* warna *red*, segmentasi G-B untuk dominan *frame* warna *green*, dan segmentasi B-R untuk dominan *frame* warna *blue*.

Keywords—Akuisisi; Segmentation; Multi-Thresholding; Quantitation; Adjustment; RGB

I. PENDAHULUAN

Manusia memiliki kemampuan penglihatan yang sangat kompleks sehingga manusia dapat melakukan proses perekaman, pendeteksian warna maupun obyek hanya dengan mengingit saat pertama kali melihat. Obyek berupa citra dilanjutkan ke otak untuk dianalisa sehingga manusia dapat mengerti obyek apa yang terlihat dalam pandangan penglihatannya. Kemampuan penglihatan manusia seperti yang disebutkan di atas, sangatlah berbeda dengan apa yang dilakukan oleh sistem penglihatan komputer atau biasa disebut computer vision dengan menggunakan sensor kamera. Pada sistem penglihatan komputer, hasil yang dilihat oleh sensor tidak dapat diterjemahkan secara langsung. Oleh karena itu, dibutuhkan proses pengolahan citra terlebih dahulu.

Obyek dibagi dalam 2 (dua) macam yaitu obyek bergerak dan tidak bergerak. Yang dimaksud dengan obyek bergerak adalah ketika letak obyek berubah dari posisi sebelumnya. Dari perpindahan obyek tersebut terdapat perbedaan citra melalui pengamatan dengan menggunakan kamera[1].

Penggunaan kamera sebagai alat akuisisi banyak digunakan diberbagai bidang seperti Video Surveillance dan monitoring lalu lintas pada sistem keamanan. Setiap pergerakan obyek akan menghasilkan perbedaan frame dari citra yang dihasilkan. Perbedaan frame dapat digunakan sebagai salah satu cara untuk mendeteksi suatu obyek yang bergerak[2]. Terdapat berbagai macam teknik dan metode untuk mendeteksi suatu obyek bergerak selain menggunakan perbedaan frame (Frame Difference). Beberapa diantaranya adalah Optical Flow,

Background Substraction, Gaussian Mixture Model dan Hidden Markov Model.

Di dalam penelitian ini akan dilakukan pelacakan obyek bergerak dengan menggunakan metode Segmentasi warna berbasis true color (RGB). Sebagai perangkat akuisisinya digunakan Camera pi yang terintegrasi dalam bareboard jenis Raspberry. Pada prinsipnya, pelacakan obyek bergerak menggunakan metode segmentasi warna RGB didasarkan pada perbedaan antar komponen R, G, dan B yang dilakukan secara real-time.

II. METODE

A. Multi-Thresholding

Thresholding adalah proses pembagian *gray level* (level keabuan) berdasarkan nilai *threshold* yang ditentukan. Jika ditetapkan 1 (satu) nilai *threshold* maka akan terdapat 2 (dua) bagian interval pada *gray level*. Semua piksel yang memiliki nilai intensitas di bawah nilai *threshold* akan diberi nilai 0 (nol), dan sebaliknya akan diberi nilai 1 (satu) hal ini dapat dilakukan dengan menggunakan Pers.(1).

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (1)$$

Metode yang dapat digunakan dalam proses *thresholding* adalah metode OTSU. Metode OTSU digunakan dalam melakukan penyeleksian nilai *threshold* berdasarkan nilai histogram citra abu-abu. Proses penyeleksian menggunakan pendekatan statistika[3]. Nilai *thresholding* dipilih dari nilai warna keabuan yang terletak pada interval histogram. Nilai ambang yang akan dicari dari suatu citra *gray level* dinyatakan dengan K. Nilai K berkisar antara 1 sampai dengan L, dengan nilai L = 255. Probabilitas setiap piksel pada level ke i dapat dinyatakan sebagai proses normalisasi histogram dengan menggunakan Pers.(2).

$$p_i = \frac{n_i}{N} \quad (2)$$

Dengan, n_i = menyatakan jumlah pixel pada level ke i; N = menyatakan total jumlah pixel pada citra.

Dalam metode OTSU nilai *threshold* pada Pers.(1) ditentukan dengan memaksimalkan *between class variance* dengan menggunakan Pers.(3) [4].

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \quad (3)$$

Dimana:

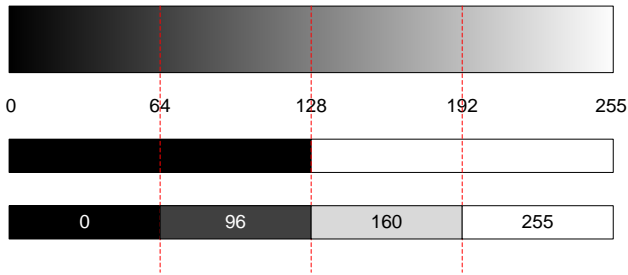
$$\omega_0 = \sum_{q=0}^{k-1} p_q(r_q)$$

$$\omega_1 = \sum_{q=k}^{L-1} p_q(r_q)$$

$$\mu_0 = \sum_{q=0}^{k-1} qp_q(r_q) / \omega_0$$

$$\mu_1 = \sum_{q=k}^{L-1} qp_q(r_q) / \omega_1$$

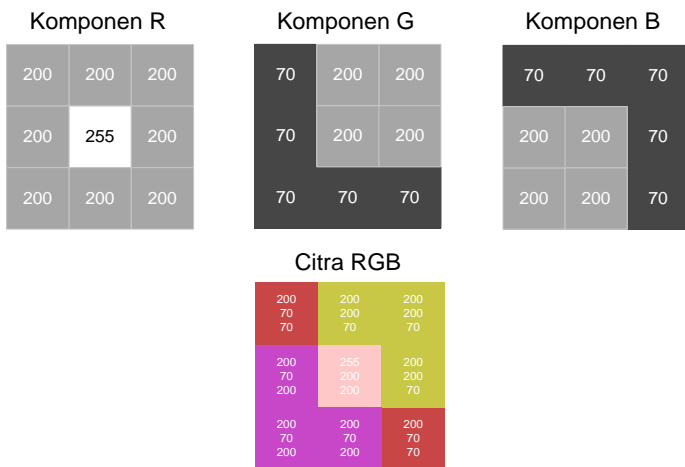
Multi-thresholding adalah proses thresholding pada gray level dengan jumlah threshold lebih dari 1 (satu). Sebagai contoh ditunjukkan dalam Gambar 1.



Gambar 1. Multi-thresholding dengan 3 (tiga) nilai threshold

B. Segmentasi Warna

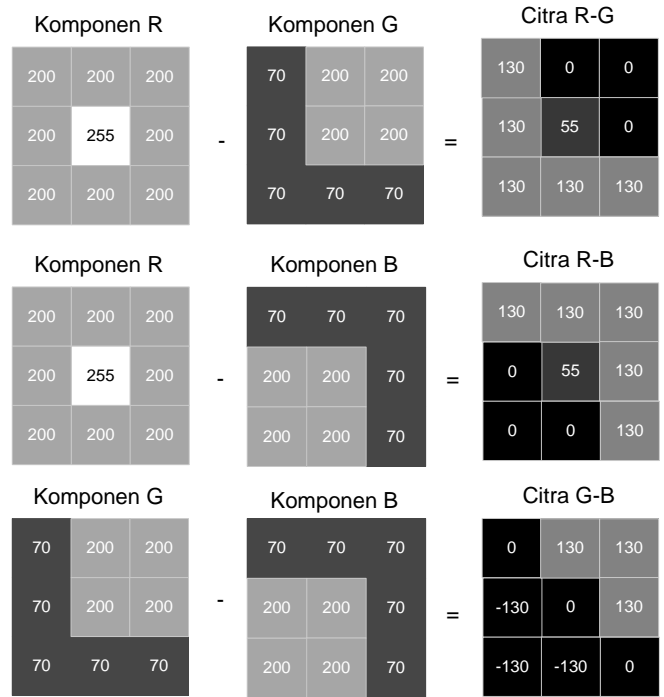
Citra RGB merupakan kombinasi dari 3 (tiga) komponen citra gray (R,G, dan B) seperti ditunjukkan dalam Gambar 2.



Gambar 2. Representasi Citra RGB

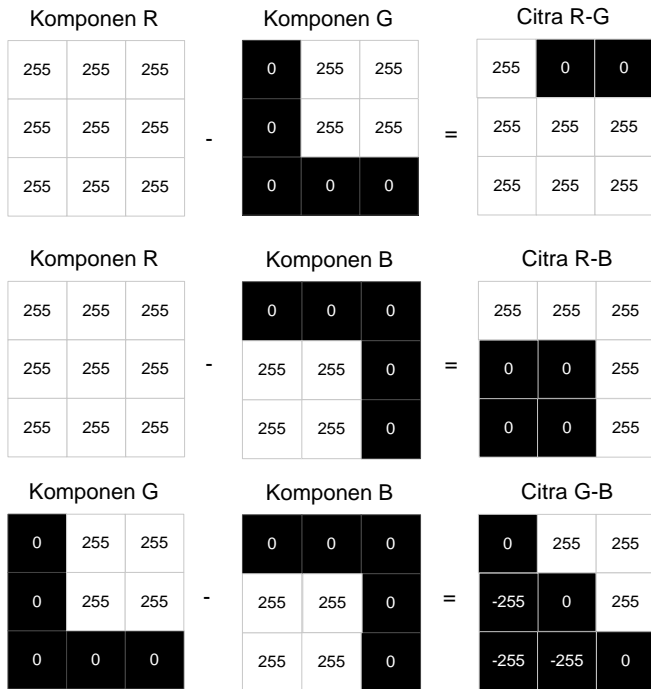
Citra RGB yang ditunjukkan dalam Gambar 2 jika dilakukan segmentasi warna secara langsung tanpa *thresholding* maka

akan menghasilkan citra seperti yang ditunjukkan dalam Gambar 3.



Gambar 3. Segmentasi tanpa *Multi-thresholding*

Tampak dari Gambar 3 menghasilkan citra segmentasi yang masih mengandung citra keabuan. Dalam hal ini obyek yang diharapkan teridentifikasi tidak begitu tampak. Jika dilakukan *Multi-thresholding* pada Gambar 2 dengan menggunakan *Multi-thresholding* seperti dalam Gambar 1 maka akan diperoleh citra seperti yang ditunjukkan dalam Gambar 4.



Gambar 4. Segmentasi menggunakan *Multi-Thresholding*

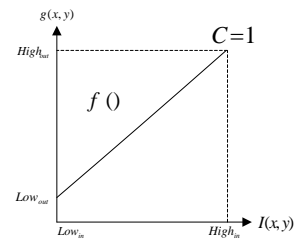
C. Adjustment

Dalam kondisi tertentu, biasanya hasil dari tahap pre-processing adalah berupa citra, terdapat beberapa citra dengan contrast (ketajaman) dan brightness (intensitas terang citra) yang rendah. Untuk meningkatkan kualitas visualisasi citra diperlukan proses adjustment.

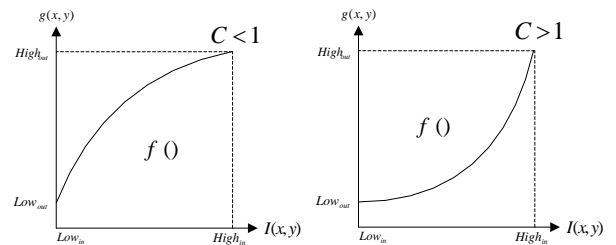
Citra dengan kontras yang rendah atau biasa disebut Low-contrast image biasanya dihasilkan dari pencahayaan yang buruk, kurangnya jangkauan perangkat akuisitor citra, atau karena kesalahan pengaturan lensa akuisitor pada saat proses akuisisi. Teknik contrast-stretching digunakan untuk meningkatkan jangkauan dinamis gray level. Pada prinsipnya teknik ini memetakan gray level dari image input ke gray level tertentu yang dinyatakan dengan Pers. (4).

$$g(x, y) = f(I(x, y)) \tag{4}$$

Dimana $f()$ adalah fungsi yang memetakan gray level image $I(x, y)$ menjadi $g(x, y)$. Secara garis besar ditunjukkan dalam Gambar 5.



Gambar A
Citra dengan Normal-Contrast



Gambar B
Citra dengan High-Contrast

Gambar C
Citra dengan Low-Contrast

Gambar 5. Pemetaan *contrast* citra

Gambar 5 merepresentasikan fungsi linear $f()$ yang memetakan $[Low_{in} High_{in}]$ dari image input ke $[Low_{out} High_{out}]$ image output. Fungsi pemetaan linear tersebut dinyatakan dengan Pers. (5).

$$Low_{in} = \min(I(x, y))$$

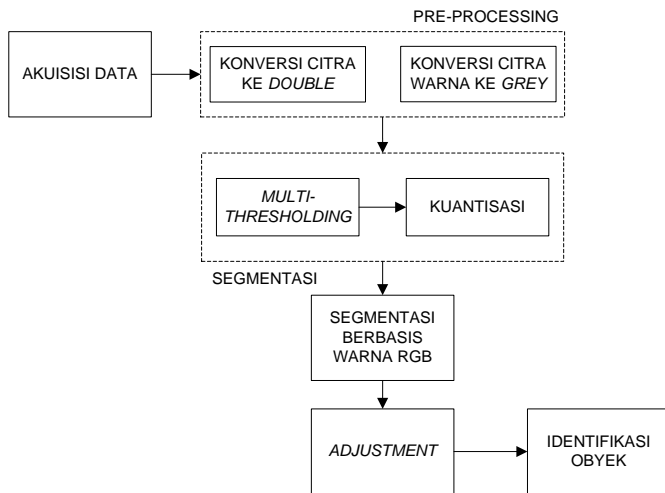
$$High_{in} = \max(I(x, y))$$

$$g(x, y) = f(I(x, y)) = Low_{out} + (High_{out} - Low_{out}) \cdot \frac{(I(x, y) - Low_{in})}{(High_{in} - Low_{in})} \tag{5}$$

Proses Adjustment memiliki satu parameter untuk mengatur brightness yaitu gamma. Namun pada percobaan ini tidak menggunakan pengaturan brightness

D. Skema Penelitian

Metode penelitian ini menggunakan 6 (enam) tahapan umum yang ditunjukkan dalam Gambar 7 sebagai metode penelitian.



Gambar 6. Metode Penelitian

III. HASIL DAN PEMBAHASAN

Pada penelitian ini, tool yang digunakan untuk pemrograman adalah MATLAB yang telah terintegrasi dengan perangkat Raspberry sebagai alat akuisisi. Secara garis besar, penelitian ini terdiri dari 5 (lima) tahapan umum, yaitu Akuisisi Data, Pre-Processing, Segmentasi, Adjustment dan Identifikasi Obyek seperti yang ditunjukkan dalam Gambar 7.

A. Akuisisi Data

Akuisisi data dilakukan dengan menggunakan Camera pi yang terintegrasi dalam board jenis Raspberry. Pada prinsipnya, video terdiri dari sejumlah frame berbasis citra RGB dalam durasi waktu tertentu. Di dalam penelitian ini, video dibangun berdasarkan jumlah frame yang di-capture oleh camera pi sebanyak 46 frame. Contoh hasil akuisisi frame ditunjukkan dalam Gambar 8.



Gambar 7. Hasil Akuisisi Menggunakan Camera Pi

Dari hasil akuisisi yang ditunjukkan dalam Gambar 8, tampak bahwa bagian obyek yang berupa pola pakaian merepresentasikan komponen warna RGB.

B. Pre-processing

Tahapan *pre-processing* terbagi menjadi 2 (dua) proses yaitu konversi citra ke tipe data *double* dan konversi citra ke tipe data *gray*. Beberapa tahap yang dilakukan adalah sebagai berikut:

- Data citra yang tersimpan di dalam matriks pertama kali adalah tipe uint8. Perhitungan aritmatika akan lebih mudah dilakukan jika data bertipe double.
- Citra *grayscale* merupakan citra yang hanya memiliki 1 (satu) kanal pada setiap pikselnya. Nilai *grayscale* merupakan nilai akumulasi dari komponen RGB. Warna yang dimiliki dari tipe *grayscale* adalah dari hitam, keabuan dan putih. Pada dasarnya *grayscale* adalah ukuran derajat keabuan disetiap piksel yang dapat dihitung dengan Pers. (6).

$$\text{Gray} = (0,2989 * R) + (0,5870 * G) + (0,1140 * B) \quad (6)$$

Hasil konversi citra ke *grayscale* dapat dilihat dalam Gambar 8.



Gambar 8. Konversi RGB to Gray

C. Segmentasi

Dalam proses segmentasi warna, terdiri dari dua tahapan utama, yaitu Proses *Multi-Thresholding* dan Kuantisasi.

1) Multi-Thresholding

Pada penelitian ini digunakan 3 (tiga) nilai *threshold* pada masing-masing komponen R, G dan B. Nilai *threshold* yang digunakan sesuai nilai probabilitas piksel tiap komponen RGB pada setiap *frame* berdasarkan Pers.(2). Tabel I menunjukkan nilai *multi-threshold* berdasarkan *frame* yang ditunjukkan dalam Gambar 9.

TABLE I. NILAI MULTI-THRESHOLD

Frame	Multi-Threshold		
	Komp R	Komp G	Komp B
Red 1	51	42	41
	98	109	103
	147	163	153
Red 2	70	54	60
	129	133	134

	189	202	199
Red 3	41	34	26
	82	91	74
	121	138	116
Green 1	38	59	32
	85	105	66
	122	132	91
Green 2	46	82	38
	106	136	87
	154	169	117
Green 3	33	54	34
	77	103	73
	113	131	102
Blue 1	76	80	49
	158	142	107
	217	194	146
Blue 2	50	54	30
	111	101	68
	158	143	94
Blue 3	36	54	31
	89	100	66
	118	126	94

2) Kuantisasi

Proses Kuantisasi adalah mengubah nilai piksel tiap *frame* ke nilai *threshold* terdekat. *Frame Red Gray 2*, *Frame Green Gray 2* dan *Frame Blue Gray 1* akan digunakan sebagai visualisasi percobaan untuk proses kuantisasi. Hasil kuantisasi ditunjukkan dalam Gambar 9.



Gambar 9. Hasil Kuantisasi

3) Segmentasi Warna

Untuk mengidentifikasi obyek bergerak pada penelitian ini dilakukan proses segmentasi terhadap komponen R, G dan B. Segmentasi dilakukan dengan cara memperkurangkan antar komponen RGB. Hasil dari pengurangan kemudian diubah ke dalam bentuk biner untuk memisahkan antara bagian *background* dan *foreground*. Beberapa kombinasi segmentasi warna berbasis komponen RGB dibagi menjadi 6 (enam) kemungkinan yaitu:

- Komponen *Red – Green* (Segmentasi R-G)

- Komponen *Red – Blue* (Segmentasi R-B)
- Komponen *Green – Red* (Segmentasi G-R)
- Komponen *Green – Blue* (Segmentasi G-B)
- Komponen *Blue – Red* (Segmentasi B-R)
- Komponen *Blue – Green* (Segmentasi B-G)

Hasil dari segmentasi warna berdasarkan 6 kemungkinan seperti tersebut di atas ditunjukkan dalam Gambar 1.

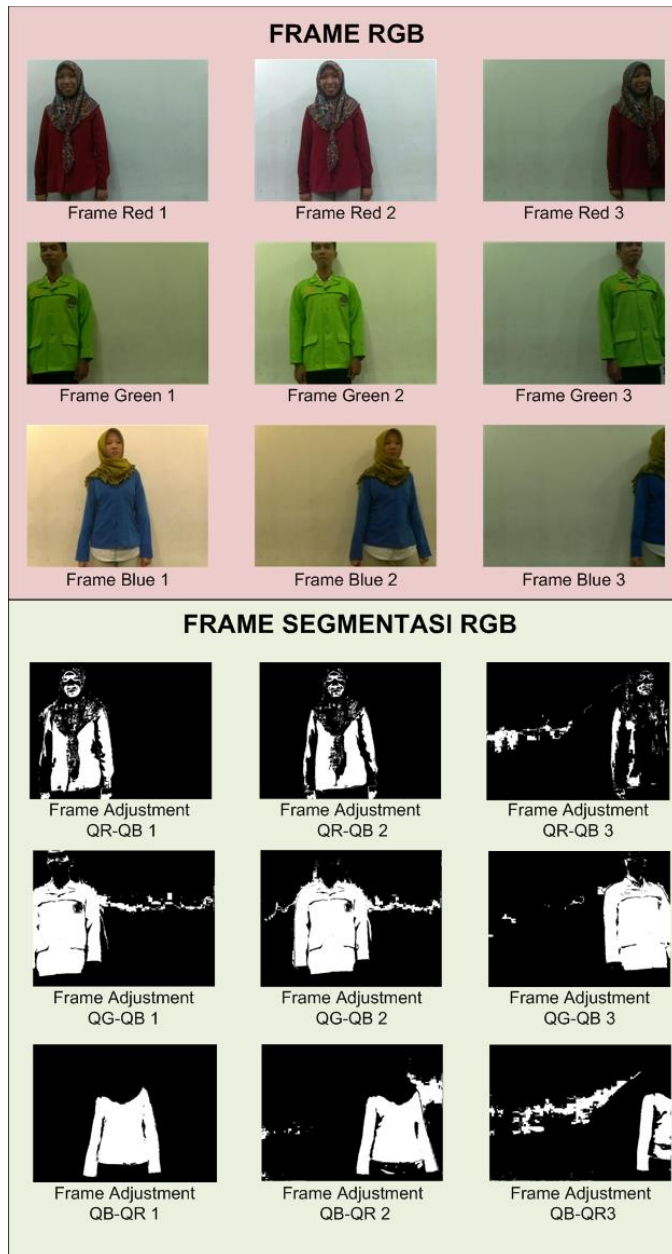
D. Adjustment

Untuk meningkatkan hasil segmentasi warna yang ditunjukkan pada Gambar 11, maka perlu dilakukan proses *adjustment* untuk mengatur nilai *contrast* (ketajaman) dan *brightness* (intensitas terang citra) pada sebuah citra dengan menggunakan Pers.(4) dan Pers.(5). Hasil dari proses *adjustment* ditunjukkan dalam Gambar 12.

E. Identifikasi Obyek

Hasil akuisisi sejumlah 46 *frame* kemudian diproses berdasarkan tahap-tahap seperti yang ditunjukkan dalam Gambar 7.

Berdasarkan Gambar 8, identifikasi segmentasi R-B digunakan untuk dominan *frame* warna *red*, segmentasi G-B untuk dominan *frame* warna *green*, dan segmentasi B-R untuk dominan *frame* warna *blue*. Hasil identifikasi ditunjukkan dalam Gambar 10.



Gambar 10. Identifikasi segmentasi frame

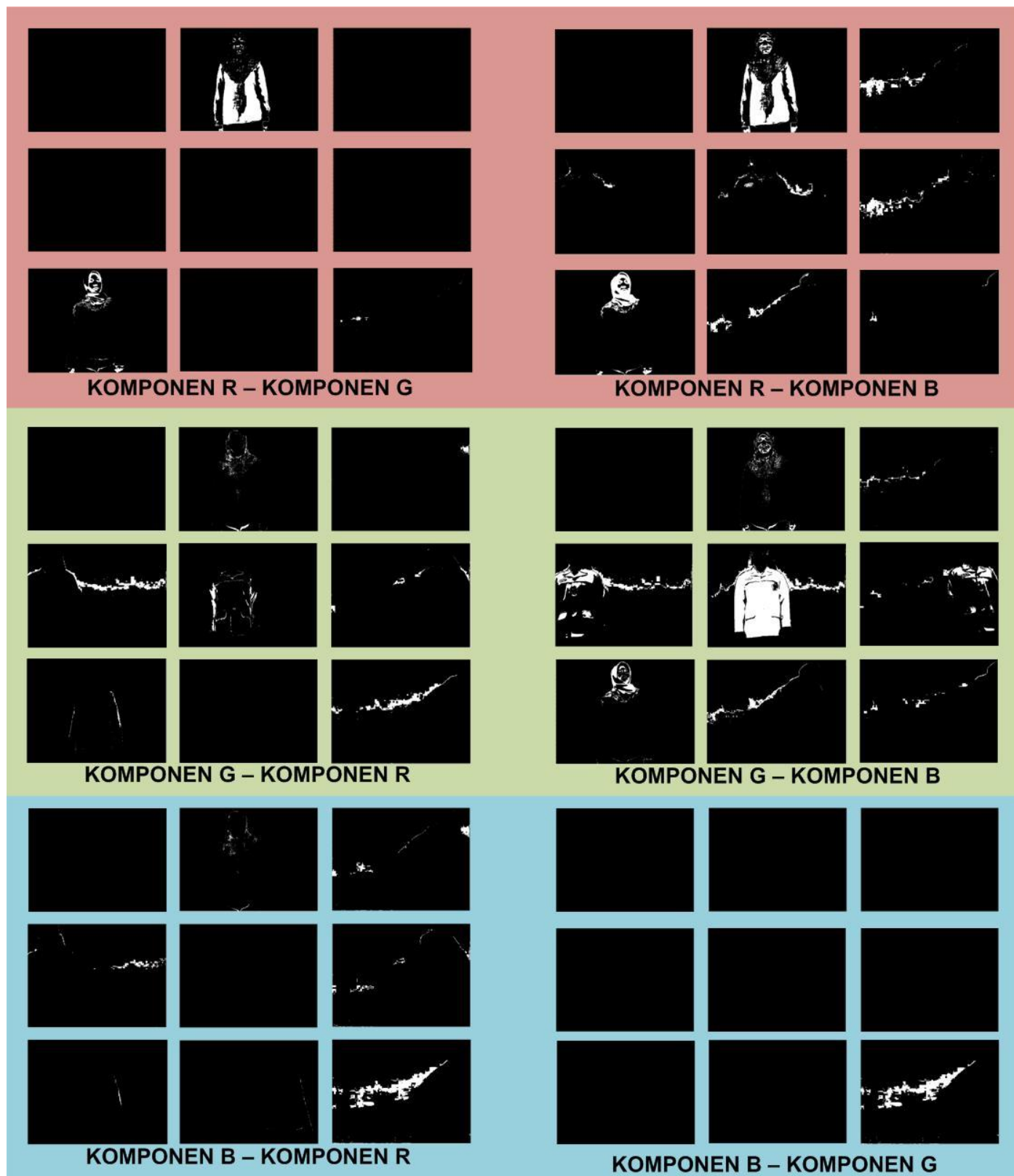
IV. KESIMPULAN

Dari hasil percobaan segmentasi dengan cara memperkurangkan antar komponen RGB diperoleh hasil segmentasi terbaik untuk *frame* dominan warna *red* adalah segmentasi R-B, untuk *frame* dominan warna *green* adalah segmentasi G-B, sedangkan untuk *frame* dominan warna *blue* adalah segmentasi B-R.

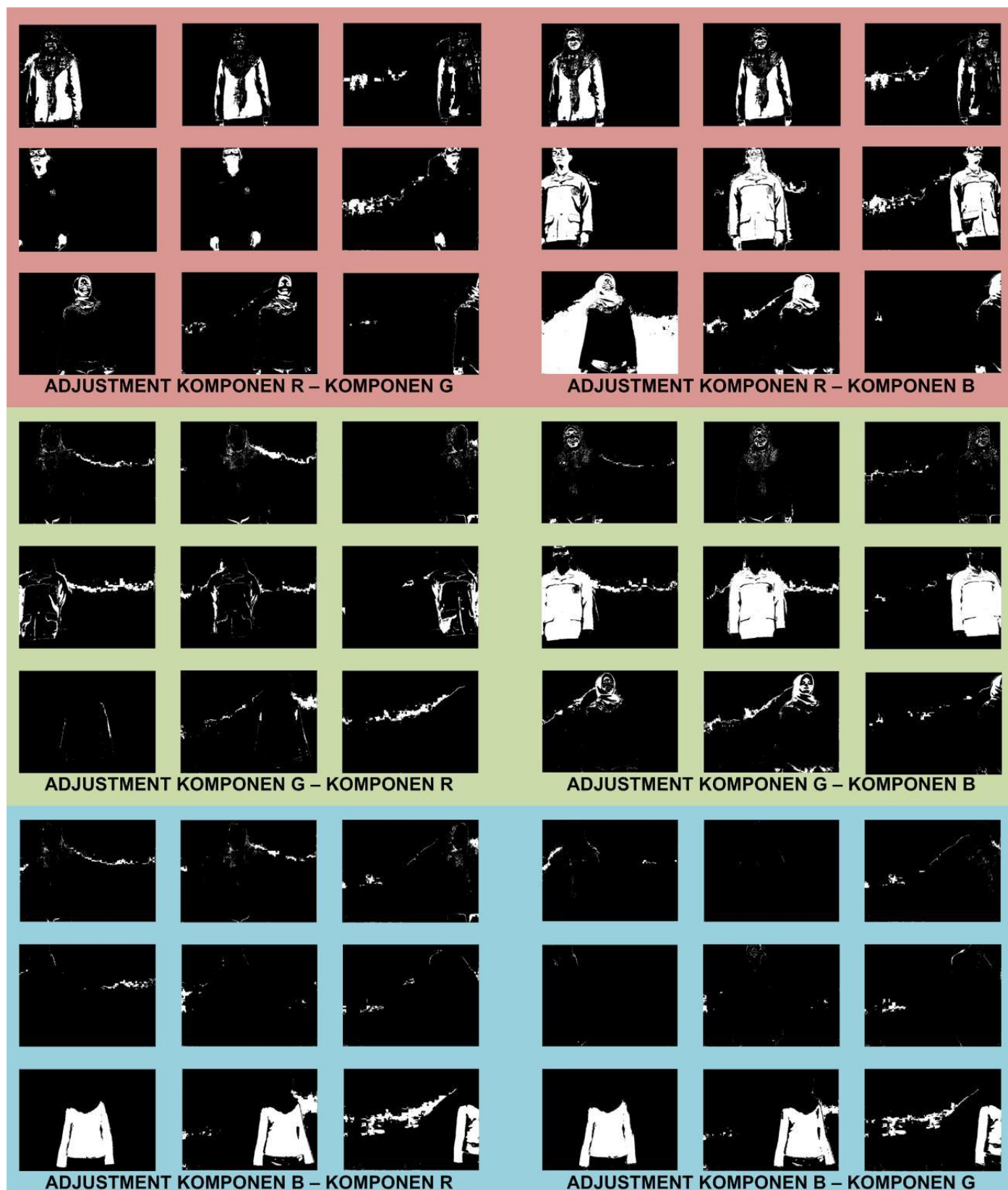
Segmentasi warna RGB bergantung pada kuantisasi hasil *multi-thresholding* yang berfungsi untuk memisahkan pola obyek dalam citra. Semakin banyak jumlah *threshold* yang digunakan maka akan semakin banyak pula kemungkinan pola obyek yang dapat dideteksi. Teknik *adjustment* diperlukan untuk meningkatkan kualitas visualisasi citra hasil segmentasi.

REFERENCES

- [1] H. Gani, A. Lawi, and I. S. Areni, "PELACAKAN CIRI OBYEK BERGERAK PADA SISTEM WAKTU NYATA vol. 3, no. 2, pp. 174–181, 2014.
- [2] N. Singla, "Motion Detection Based on Frame Difference Method," *Int. J. Inf. Comput. Technol.*, vol. 4, no. 15, pp. 1559–1565, 2014.
- [3] D. Putra, "Binerisasi citra tangan dengan metode otsu," *Tekno. Elektro*, vol. 3, no. 2, pp. 11–13, 2004.
- [4] P. S. Liao, T. S. Chen, and P. C. Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*, vol. 17, no. 5, pp. 713–727, 2001.
- [5] I. The Mathworks, *Fuzzy Logic Toolbox™ User 's Guide*. 2014.



Gambar 11. Hasil Segmentasi Warna



Gambar 12. Hasil Adjustment