

SUBSTRAKSI BACKGROUND DAN DETEKSI BAYANGAN PADA CITRA GRAYSCALE SEKUENSIAL

Binarahandra R.E.⁽¹⁾
namakubinar@yahoo.com

Widi Hapsari⁽²⁾
widi@ukdw.ac.id

Junius Karel T.⁽³⁾
karel@ukdw.ac.id

Abstraksi

Ekstraksi objek pada suatu citra grayscale dengan background yang tidak tetap merupakan hal yang rumit sehingga membutuhkan pendekatan dengan computer vision. Penelitian ini menerapkan proses substraksi background dan deteksi bayangan untuk mendapatkan bentuk dari objek yang diinginkan.

Proses substraksi bayangan membutuhkan background model yang diperoleh dari pengolah citra sekuensial input dari pengguna. Tujuan dari background model ini adalah untuk menentukan pengelompokan piksel sebagai background atau sebagai objek. Proses deteksi bayangan akan mengkalkulasikan nilai piksel objek dengan background model menggunakan normalized cross-correlation (NCC) untuk melihat apakah nilai objek adalah suatu penguatan nilai dari background model yang menandakan piksel tersebut adalah bayangan. Penelitian ini akan mengimplementasikan metode tersebut dan menganalisa hasil serta pengaruh input pengguna untuk mendapatkan hasil substraksi background dan deteksi bayangan yang optimal.

Kata kunci : *computer vision, substraksi background, deteksi bayangan*

1. Pendahuluan

Perkembangan teknologi telah membantu memudahkan manusia dan *computer vision* merupakan salah satu contohnya. Salah satu permasalahan yang dapat dipecahkan oleh *computer vision* adalah dalam hal ekstraksi objek dari suatu citra dengan cara mengeliminasi *background*.

Proses ekstraksi objek terutama dengan *background* yang dinamis atau sering disebut *adaptive background* membutuhkan proses *training*. Proses *training* tersebut membutuhkan beberapa sampel citra untuk diolah yakni berupa citra dari *background* yang diambil secara sekuensial pada sudut yang sama. Pengambilan citra secara sekuensial ditujukan untuk melihat variasi perubahan *background* yang mungkin terjadi. Dalam

¹ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

² Teknik Informatika, Fakultas Teknologi Informasi Univeristas Kristen Duta Wacana

³ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

prosesnya citra akan dikonversikan ke dalam model warna *grayscale*. Hal tersebut ditujukan untuk memudahkan proses *training* karena nilai yang diproses dalam setiap pikselnya hanya satu nilai.

Permasalahan yang acap kali muncul setelah proses ekstraksi objek adalah bayangan seringkali diklasifikasikan sebagai objek. Menurut Grest, Frahm, dan Koch (2003), *normalized cross-correlation (NCC)* dapat digunakan untuk mengenali kandidat bayangan. *NCC* memiliki karakteristik untuk mengenali versi skala dari suatu citra. Kandidat bayangan yang terpilih akan diperbaiki lagi dengan proses perhitungan statistik lokal dari ratio piksel.

Permasalahan yang dibahas adalah bagaimana sistem dapat mengekstraksi bentuk objek dari suatu citra dan memisahkan bayangan dari objek hasil ekstraksi citra.

2. Landasan Teori

2.1. Konversi Grayscale

Ada beberapa metode untuk mengkonversi suatu citra warna *RGB* ke *grayscale*. Salah satu cara yang paling umum dipakai adalah rumus:

$$\text{Grayscale} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad [1]$$

Kanal *Red* direpresentasikan dengan R, kanal *Green* dengan G dan kanal *Blue* dengan B.

2.2. Pembuatan *Background Model*

Background model diperoleh dari pemrosesan beberapa citra sekuensial hasil ekstraksi dari file video. Penggunaan citra sekuensial ditujukan untuk mendapatkan nilai toleransi terhadap perubahan pencahayaan yang mungkin terjadi pada saat pengambilan *background*.

Dari hasil ekstraksi video didapatkan N buah citra sekuensial. Citra yang berasal dari hasil ekstraksi yang berupa citra *RGB* kemudian dikonversi menjadi citra *grayscale*. Nilai piksel dari citra-citra tersebut akan dipindahkan ke dalam *array* V yang merupakan *array* 3 dimensi.

Array V digambarkan memiliki koordinat i , j dan k , dimana i dan j merepresentasikan kolom dan baris dari satu citra dan k merepresentasikan urutan citra sekuensial. Jadi $V^k(i,j)$ adalah nilai dari piksel (i,j) dari citra ke- k pada V .

Pada seluruh nilai piksel (i,j) pada V dilakukan kalkulasi untuk mendapatkan nilai standar deviasi $\sigma(i,j)$ dan mean intensitas piksel $\lambda(i,j)$.

Ditentukan V^z merupakan citra ke- k dari *array* V atau V^k , jika setiap nilai (i,j) pada V^k memenuhi persamaan:

$$|V^k(i, j) - \lambda(i, j)| \leq x\sigma(i, j), \quad [2]$$

Dimana menurut Jaques Jr, Jung dan Musse (2005) nilai dari x adalah 2. Bila ada salah satu nilai (i, j) pada V^k yang tidak memenuhi persamaan [2], maka V^z tidak memenuhi syarat menjadi V^z . V^z inilah yang akan dipergunakan dalam proses pembuatan *background model*.

Background model yang terbentuk dari *array V* yang kita lambangkan dengan $B(i, j)$, terdiri dari 3 komponen nilai yakni $[m(i, j), n(i, j), d(i, j)]$ yang digambarkan sebagai berikut:

$$\begin{bmatrix} m(i, j) \\ n(i, j) \\ d(i, j) \end{bmatrix} = \begin{bmatrix} \min_z V^z(i, j) \\ \max_z V^z(i, j) \\ \max_z |V^z(i, j) - V^{z-1}(i, j)| \end{bmatrix}, \quad [3]$$

$m(i, j)$ = nilai minimum dari seluruh citra (i, j) pada *array V^z*

$n(i, j)$ = nilai maksimum dari seluruh citra (i, j) pada *array V^z*

$d(i, j)$ = nilai maksimum absolut dari selisih nilai piksel (i, j) pada citra yang berurutan pada *array V^z*

2.3. Substraksi Background

Dilambangkan *input* citra yang ingin disubstraksi dan diidentifikasi bayangannya adalah $I(i, j)$. Citra $I(i, j)$ yang ideal adalah citra *background* yang ditambahkan dengan objek baru. Untuk mendapatkan citra $I(i, j)$ yang ideal, citra $I(i, j)$ dapat diambil bersamaan dengan pengambilan video untuk pembuatan *background model*. Tujuan dari pengambilan yang sama untuk mencegah perubahan posisi pada *background*. Citra $I(i, j)$ ini juga dikonversikan menjadi *grayscale* terlebih dahulu sebelum diproses.

Proses substraksi citra $I(i, j)$ dibandingkan dengan $B(i, j)$. Piksel (i, j) pada citra $I(i, j)$ diklasifikasikan sebagai objek jika memenuhi:

$$I(i, j) > (m(i, j) - k\mu) \text{ dan } I(i, j) < (n(i, j) + k\mu) \quad [4]$$

k = nilai parameter tetap

μ = median dari seluruh nilai $d(i, j)$ dari persamaan [2]

2.4. Identifikasi Kandidat Bayangan

$B(i, j)$ adalah *background model* yang terbentuk dari proses *temporal median filter* dan $I(i, j)$ adalah citra yang ingin disubstraksi. Untuk setiap piksel $I(i, j)$ yang diklasifikasikan sebagai objek pada persamaan (3), dibuat *template T_{ij}*. T_{ij} merupakan sekumpulan piksel berbentuk persegi berukuran $(2N+1) \times (2N+1)$ dengan piksel (i, j) sebagai pusatnya.

Piksel yang membentuk T_{ij} adalah $I(i+n, j+m)$, dimana $-N \leq n \leq N, -N \leq m \leq N$.
Dapat dikatakan T_{ij} bersesuaian dengan piksel tetangga (i,j) .

Nilai NCC dari *template* T_{ij} dan citra B untuk piksel (i,j) adalah:

$$NCC(i, j) = \frac{ER(i, j)}{E_B(i, j)E_{Tij}} \quad [5]$$

Dimana:

$$ER(i, j) = \sum_{n=-N}^N \sum_{m=-N}^N B(i+n, j+m) T_{ij}(n, m),$$

$$E_B(i, j) = \sqrt{\sum_{n=-N}^N \sum_{m=-N}^N B(i+n, j+m)^2}, \text{ dan} \quad [6]$$

$$E_{Tij} = \sqrt{\sum_{n=-N}^N \sum_{m=-N}^N T_{ij}(n, m)^2}.$$

Untuk piksel (i,j) dalam regional bayangan, nilai NCC dalam *neighboring region* harus bernilai mendekati satu dan nilai E_{Tij} dalam regional ini harus lebih rendah dari $E_B(i, j)$ yang merupakan regional yang korespondensif dengan citra *background*. Nilai piksel (i,j) pada citra $I(i,j)$ ditetapkan sebagai kandidat bayangan jika:

$$NCC(i, j) > L_{NCC} \text{ dan } E_{Tij} < E_B(i, j) \quad [7]$$

2.5. Perbaikan Bayangan

Proses perbaikan bayangan diperlukan untuk mengatasi kekurangan dari algoritma NCC . Kekurangan tersebut adalah salah identifikasi objek bergerak sebagai bayangan. Proses perbaikan bayangan menggunakan piksel pada citra $I(i,j)$ yang teridentifikasi sebagai kandidat bayangan pada persamaan [7].

Tujuan dari tahap ini adalah dengan memverifikasi jika rasio $I(i,j)/B(i,j)$ dengan piksel tetangganya disekitar piksel kandidat bayangan adalah konstan, dengan memperhitungkan standar deviasi dari $I(i,j)/B(i,j)$ dengan piksel tetangga.

Ditentukan regional R berukuran $(2M+1) \times (2M+1)$ piksel (nilai $M = 1$) dengan pusatnya adalah piksel kandidat bayangan, dan kandidat diklasifikasikan sebagai bayangan jika:

$$std_R \left(\frac{I(i, j)}{B(i, j)} \right) < L_{std} \text{ dan } L_{low} \leq \left(\frac{I(i, j)}{B(i, j)} \right) < 1, \quad [8]$$

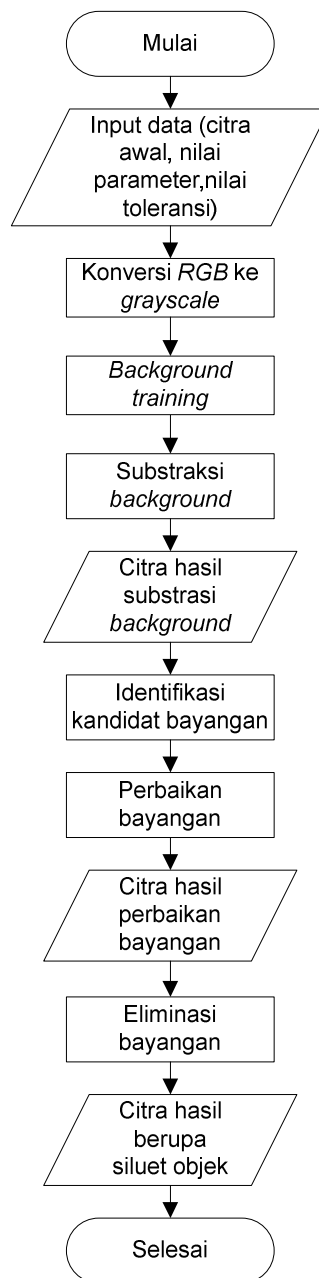
$std_R \left(\frac{I(i, j)}{B(i, j)} \right)$ = standar deviasi dari $I(i,j)/B(i,j)$ pada regional R

L_{std} = nilai *threshold* yang mengotrol maksimum standar deviasi pada piksel tetangga yang dianalisis.

L_{low} = nilai *threshold* yang mencegah pengklasifikasian yang salah dari obyek gelap dengan piksel berintensitas rendah sebagai bayangan.

3. Perancangan Sistem

Secara garis besar terdapat 5 buah proses utama dalam sistem. Proses tersebut adalah konversi grayscale, *background training*, substraksi *background*, identifikasi kandidat bayangan dan perbaikan bayangan. *Input* yang diterima sistem adalah citra awal, nilai parameter, dan nilai toleransi. *Output* yang dihasilkan sistem adalah citra siluet dari bentuk objek. Gambar 1. merupakan diagram alir proses sistem secara umum.

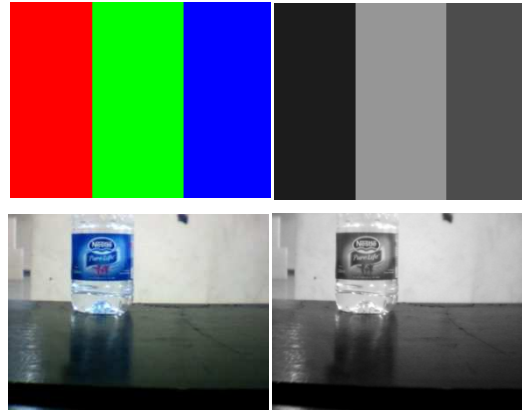


Gambar 1. Diagram Alir Sistem

4. Analisis

4.1. Analisa Proses Konversi Grayscale

Proses konversi ini berjalan dengan cukup baik. Dari citra hasil yang diperoleh tidak terdapat permasalahan yang dapat mengganggu proses selanjutnya. Dibawah ini adalah contoh citra hasil dari proses konversi *grayscale*.



Gambar 1.a. (kiri atas), 2.b. (kanan atas), 2.c. (kiri bawah), 2.d. (kanan bawah)
Citra sebelum dan sesudah konversi

4.2. Analisa *Background Training*

4.2.1. Analisa Nilai Toleransi Terhadap Pemilihan Citra Sekuensial

Nilai toleransi yang dimaksud disini adalah nilai x dari persamaan [2] yakni $|V^k(i, j) - \lambda(i, j)| \leq x\sigma(i, j)$, dimana nilai toleransi x yang disarankan dalam *W4* (2000) adalah 2. Hal tersebut untuk memastikan bahwa *background model* yang terbentuk dapat melakukan substraksi dengan baik.

Namun dari hasil percobaan ternyata nilai x yang disarankan tidak meloloskan semua citra sekuensial yang terpilih dan menyebabkan proses berikutnya tidak dapat berjalan. Hal tersebut terjadi dikarenakan video yang diekstrak menjadi citra sekuensial menggunakan webcam berkualitas rendah sehingga terdapat *noise* pada citra hasil ekstraksi.

Berdasarkan hal tersebut maka nilai toleransi dinaikkan namun dengan tetap memastikan bahwa citra sekuensial yang diproses merupakan citra *background* yang valid. Berikut adalah contoh citra *background* yang diinginkan dan yang tidak diinginkan.



Gambar 2.a. (ujung kiri), 3.b. (tengah kiri) , 3.c. (tengah kanan), 3.d. (ujung kanan)
Contoh citra *background* diinginkan dan tidak diinginkan

Gambar 3.a. dan 3.c. menunjukkan citra *background* yang diinginkan dimana objek-objek pada *background* berada pada posisi tetap atau tidak bergerak. Sedangkan gambar 3.b. dan 3.d. merupakan contoh citra *background* yang tidak diinginkan atau pengganggu karena terdapat objek yang bergerak. Objek bergerak ini yang akan mengacaukan *background model* yang terbentuk.

Menurut Jaques Jr, Jung dan Musse (2005), 100 buah citra sekuensial yang benar dapat membentuk *background model* yang baik. Berikut adalah tabel hasil percobaan dengan berbagai variasi nilai toleransi berdasar gambar 3.a. dan 3.c. Citra benar adalah citra *background* yang diinginkan dan citra pengganggu adalah citra yang tidak diinginkan.

Tabel 1. Hasil Pengujian Toleransi

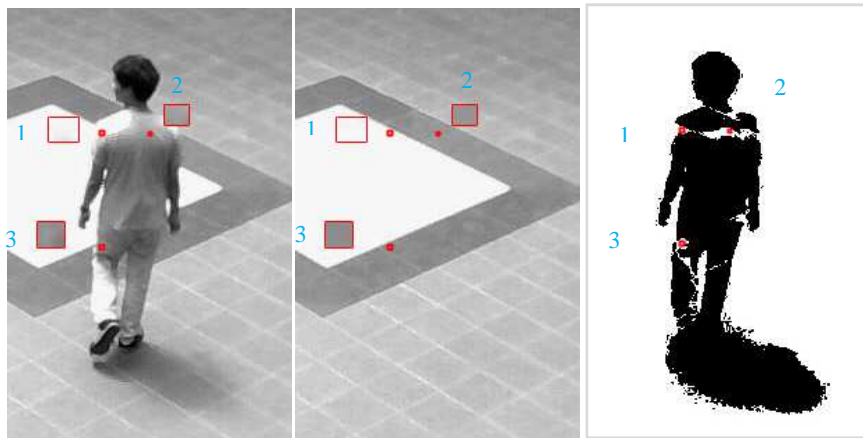
| No | Tol. | Jumlah Citra benar | Jumlah Citra pengganggu | Gambar 3.a | | Gambar 3.c | |
|----|------|--------------------|-------------------------|-------------------|------------------|-------------------|------------------|
| | | | | dengan pengganggu | tanpa pengganggu | dengan pengganggu | tanpa pengganggu |
| 1 | 2 | 100 | 10 | 0 | 0 | 0 | 0 |
| 2 | 3 | 100 | 10 | 0 | 0 | 0 | 0 |
| 3 | 4 | 100 | 10 | 0 | 0 | 0 | 0 |
| 4 | 5 | 100 | 10 | 10 | 2 | 0 | 0 |
| 5 | 5.5 | 100 | 10 | 29 | 19 | 0 | 0 |
| 6 | 6 | 100 | 10 | 45 | 51 | 0 | 0 |
| 7 | 6.5 | 100 | 10 | 69 | 58 | 0 | 0 |
| 8 | 7 | 100 | 10 | 74 | 61 | 0 | 2 |
| 9 | 7.5 | 100 | 10 | 89 | 87 | 72 | 66 |
| 10 | 8 | 100 | 10 | 90 | 89 | 72 | 66 |
| 11 | 8.5 | 100 | 10 | 90 | 89 | 72 | 67 |
| 12 | 9 | 100 | 10 | 90 | 89 | 73 | 67 |
| 13 | 9.5 | 100 | 10 | 90 | 90 | 73 | 67 |
| 14 | 10 | 100 | 10 | 91 | 90 | 73 | 67 |
| 15 | 10.5 | 100 | 10 | 96 | 100 | 73 | 100 |
| 16 | 11 | 100 | 10 | 110 | 100 | 110 | 100 |

Dari tabel di atas toleransi dengan jumlah citra terpilih dibawah atau sama dengan 2 dianggap toleransi yang dapat digunakan dalam proses. Nilai toleransi 7.5 hingga 10 memiliki jumlah citra terpilih tidak jauh berbeda. Dengan pertimbangan bahwa nilai toleransi awal adalah 2, maka diambil nilai toleransi yang tidak terlalu jauh dari toleransi awal namun memiliki jumlah citra terpilih yang cukup banyak. Sehingga ditentukan bahwa nilai toleransi yang dipakai pada sistem ini adalah 7.5.

Dengan didapatkan citra terpilih, maka dengan menggunakan persamaan [3] terbentuklah *background model* yang terdiri dari 3 komponen nilai yakni nilai minimum $m(i,j)$, nilai maksimum $n(i,j)$ dan nilai maksimum absolut dari selisih nilai piksel (i,j) pada citra yang berurutan $d(i,j)$ pada *array* V^z .

4.2.2. Analisa Substraksi *Background*

Citra hasil yang diharapkan dari proses substraksi *background* adalah bentuk siluet gabungan dari objek dan bayangan. Namun berdasar dari hasil percobaan terdapat beberapa bagian dari objek yang tidak ditandai sebagai objek.



Gambar 3. Piksel-piksel objek yang tidak dikenali sebagai objek

Gambar 4 menunjukkan bahwa titik 1, 2 dan 3 yang sesungguhnya adalah bagian dari objek ternyata berdasarkan hasil substraksi, titik-titik piksel tersebut tidak ditandai sebagai bagian dari objek. Berdasarkan rumus persamaan [4] dimana piksel pada foreground akan dikenali sebagai objek jika memenuhi rumus:

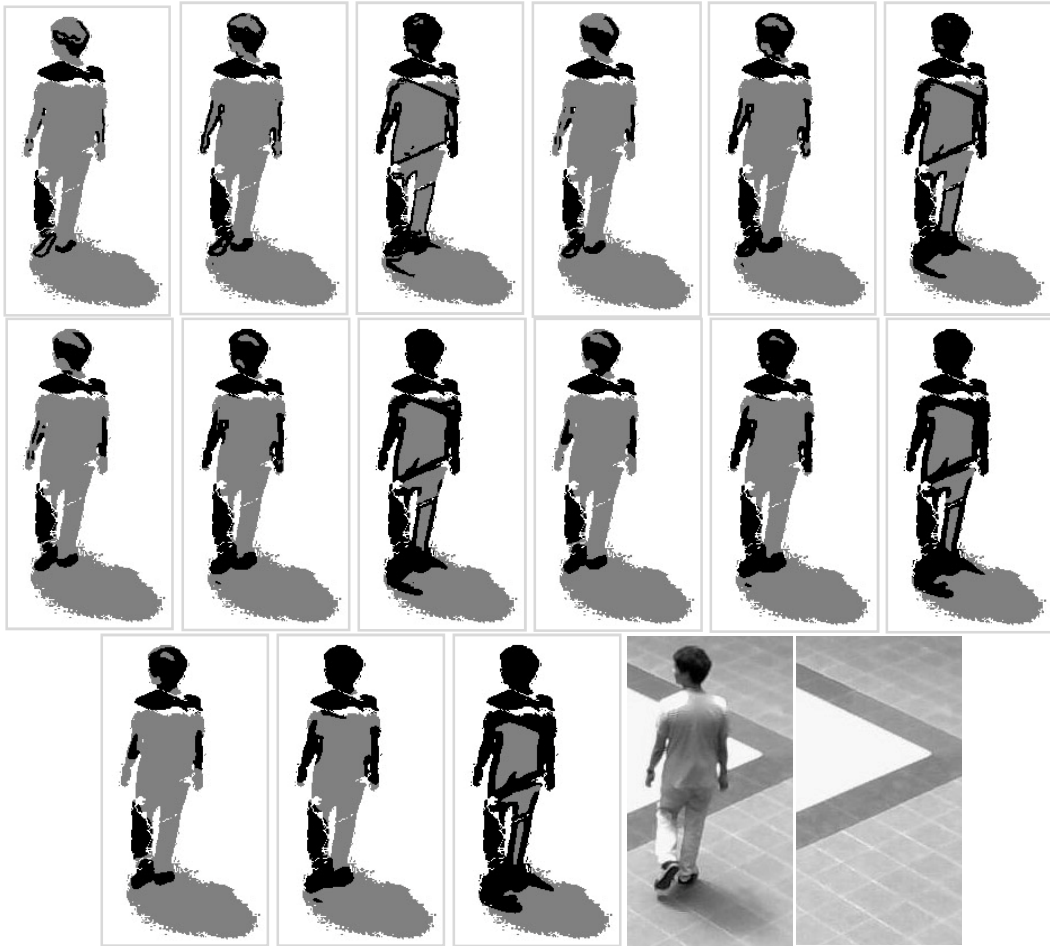
$$I(i, j) > (m(i, j) - k\mu) \text{ dan } I(i, j) < (n(i, j) + k\mu)$$

Nilai k adalah parameter tetap dan μ adalah median untuk seluruh nilai $d(i,j)$. Titik-titik diatas setelah dianalisis ternyata tidak memenuhi persyaratan rumus di atas.

4.3. Analisa Identifikasi Bayangan

4.3.1. Analisa Ukuran *Neighbour* dan L_{NCC} Terhadap Identifikasi Bayangan

Menurut Jaques Jr, Jung dan Musse (2005), nilai L_{NCC} dan nilai ukuran *neighbour* yang disarankan adalah 0,95 dan 4. Maka nilai yang akan diujicobakan adalah nilai-nilai yang tidak terlalu jauh dari nilai yang disarankan pada penelitian sebelumnya. Berikut adalah gambar hasil percobaan dengan berbagai kombinasi nilai L_{NCC} dan ukuran *neighbour*.



Gambar 4.a, 5.b, 5.c, 5.d, 5.e, 5.f (baris pertama dari kiri ke kanan),
5.g, 5.h, 5.i, 5.j, 5.k, 5.l (baris kedua dari kiri ke kanan),
5.m, 5.n, 5.o, 5.p, 5.q (baris ketiga dari kiri ke kanan)
 Hasil Identifikasi Bayangan dengan berbagai toleransi
Gambar 5.p sebagai gambar objek dan
Gambar 5.q sebagai sampel dari *background*.

Pada gambar 5.a, 5.d, 5.g, 5.j dan 5.m yang memiliki nilai L_{NCC} 0.90 menghasilkan gambar yang kurang memuaskan. Tampak pada bagian tangan kiri dari objek yang seharusnya teridentifikasi sebagai objek justru dikenali sebagai bayangan. Pada gambar 5.b, 5.e, 5.h, 5.k dan 5.n yang memiliki nilai L_{NCC} 0.95 menghasilkan hasil yang dapat diterima karena mampu mengenali bagian bayangan dan objek dengan baik. Pada gambar 5.c, 5.e, 5.i, 5.l dan 5.o yang memiliki nilai L_{NCC} 0.99 menghasilkan kurang memuaskan. Pada bagian bayangan dibawah kaki terdapat bayangan yang justru dikenali sebagai objek. Dari pengamatan diatas maka asumsi ukuran nilai L_{NCC} yang memiliki kemampuan mengenali bayangan dan objek dengan baik adalah 0.95.

Dengan asumsi nilai L_{NCC} yang digunakan 0.95, akan diperbandingkan gambar 5.b, 5.e, 5.h, 5.k dan 5.n untuk melihat ukuran *neighbour* yang dapat mengenali bayangan dengan baik. Pada gambar 5.b dan 5.e pada bagian tangan dan bagian telapak kaki kiri

terdapat sebagian yang dikenali sebagai bayangan, sedangkan pada gambar 5.h, 5.k dan 5.n bagian tersebut tetap dikenali sebagai objek.

Pada gambar 5.b, 5.e dan 5.h pada bagian bayangan dibawah kaki kiri dapat dikenali sebagai bayangan bayangan, sedangkan pada gambar 5.k dan 5.n bagian tersebut justru dikenali sebagai objek. Dengan memperhatikan hasil pengamatan maka gambar 5.h dengan nilai *neighbour* 4 memberikan hasil yang cukup baik dalam mengenali bayangan dan objek.

Dengan demikian nilai L_{NCC} dan ukuran *neighbour* yang dianggap dapat mengenali objek dan bayangan dengan baik adalah 4 dan 0.95.

4.4. Analisa Perbaikan Bayangan

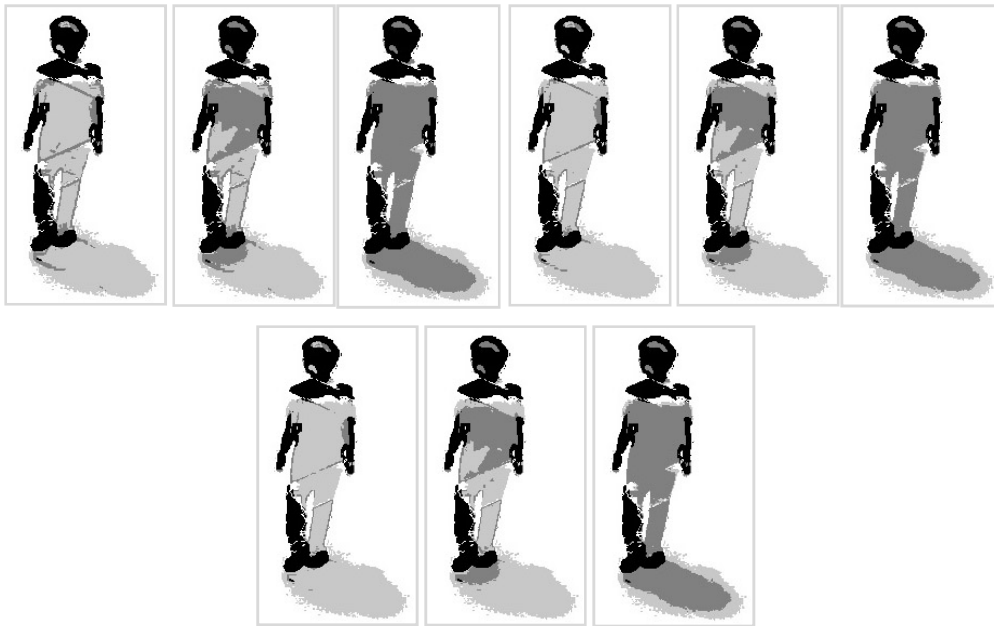
Proses perbaikan bayangan menggunakan metode dengan memverifikasi jika rasio $I(i,j)/B(i,j)$ dengan piksel tetangganya disekitar piksel bayangan hasil identifikasi bayangan adalah konstan, dengan memperhitungkan standar deviasi dari $I(i,j)/B(i,j)$ dengan piksel tetangga. Metode ini untuk ditujukan untuk mengatasi kesalahan pengenalan objek bergerak sebagai bayangan. Pada proses perbaikan bayangan kembali mempergunakan *background model* hasil *temporal median filter*. Hasil dari proses ini dipengaruhi oleh nilai toleransi pembatas L_{std} dan L_{low} .

4.4.1. Analisa Nilai Toleransi L_{std} dan L_{low} Terhadap Perbaikan Bayangan

Menurut Jaques Jr, Jung dan Musse (2005), nilai L_{std} dan L_{low} yang disarankan adalah 0,05 dan 0.5. Maka nilai yang akan diujicobakan adalah nilai-nilai yang tidak terlalu jauh dari nilai yang disarankan pada penelitian sebelumnya. Berikut adalah gambar hasil percobaan dengan berbagai kombinasi nilai L_{std} dan L_{low} .



Gambar 5 a, 6.b, 6.c, 6.d, 6.e, 6.f
(baris pertama dari kiri ke kanan),
Hasil Perbaikan Bayangan dengan berbagai toleransi



Gambar 6.g, 6.h, 6.i, 6.j, 6.k, 6.l (baris kedua dari kiri ke kanan),
Gambar 6.m, 6.n, 6.o (baris ketiga dari kiri ke kanan)
 Hasil Perbaikan Bayangan dengan berbagai toleransi

Hasil dari perbaikan bayangan adalah mendefinisikan ulang piksel objek yang ditandai sebagai bayangan pada proses identifikasi bayanga. Metode menggunakan L_{std} dan L_{low} sebenarnya ditujukan untuk mengatasi kesalahan pengenalan objek bergerak sebagai bayangan. Namun pada hasil percobaan metode ini dapat digunakan untuk pendefinisian ulang bayangan.

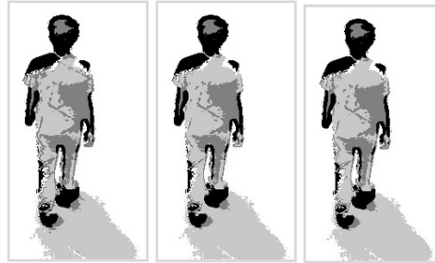
Pada gambar 6.a, 6.d, 6.g, 6.j dan 6.m yang memiliki nilai L_{low} 0.03 menghasilkan hasil yang kurang memuaskan karena menghasilakn area terpilih yang kurang akurat.

Pada gambar 6.b, 6.e, 6.h, 6.k dan 6.n yang memiliki nilai L_{low} 0.05 hanya satu gambar saja yang menghasilkan hasil yang kurang memuaskan yakni gambar 6.a dimana pada area bayangan dibagian bawah kaki sebagian besar didefinisikan ulang sebagai objek. Selebihnya pada gambar 6.e, 6.h, 6.k dan 6.n area bayangan kaki sebagian besar masih ditandai sebagai bayangan dan pada bagian punggung sebagian sudah didefinisikan ulang sebagai objek.

Pada gambar 6.c, 6.f, 6.i, 6.l dan 6.o yang memiliki nilai L_{low} 0.08 memberikan hasil yang kurang memuaskan dimana pada area bayangan dibagian bawah kaki sebagian besar didefinisikan ulang sebagai objek. Dari pengamatan diatas maka asumsi ukuran nilai L_{low} yang memiliki kemampuan mendefinisikan ulang bayangan dan objek dengan baik adalah 0.5.

Dengan asumsi nilai L_{low} yang digunakan 0.5, akan diperbandingkan gambar 6.b, 6.e, 6.h, 6.k dan 6.n untuk melihat nilai L_{std} yang dapat mendefinisikan ulang bayangan dengan baik. Pada gambar 6.b tampak bahwa sebagian besar area dibawah kaki

didefinisikan ulang sebagai objek dimana seharusnya area tersebut merupakan bayangan. Dari perbandingan gambar 6.e, 6.h, 6.k dan 6.n, ternyata gambar 6.h, 6.k dan 6.n memiliki area hasil perbaikan bayangan yang kurang lebih sama. Untuk meneliti toleransi nilai L_{std} yang baik, dilakukan pengambilan sampel gambar yang berbeda.



Gambar 6.a, 7.b dan 7.c
Hasil Perbaikan Bayangan dengan untuk L_{std} yang berbeda

Pada gambar 7.a, 7.b dan 7.c memiliki nilai L_{std} 0.05, 0.08 dan 0.1 dengan nilai L_{low} 0.5. Dari ketiga gambar tersebut ternyata gambar 7.a memiliki area terdefinisi ulang sebagai objek di bagian punggung lebih banyak dari gambar 7.b dan 7.c. Pada bagian bawah kaki area yang didefinisikan ulang pada semua gambar 7 tidak jauh berbeda.

Dengan demikian nilai L_{std} yang dianggap cukup bisa mendefinisikan ulang bayangan dan objek dengan baik adalah 0.05.

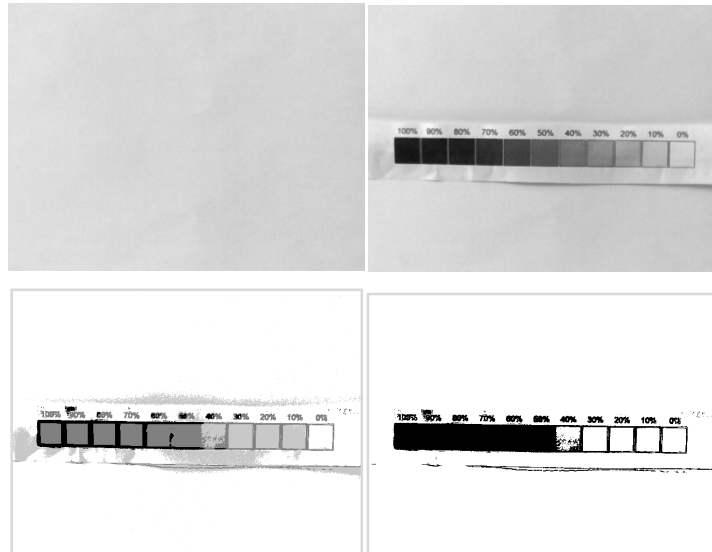
4.4.2. Analisa Keseluruhan Proses Substraksi Background dan Deteksi Bayangan

Pada hasil pengamatan sebelumnya didapatkan bahwa nilai toleransi *background* adalah 7.5, ukuran neighbour adalah 4, toleransi L_{NCC} adalah 0.95, toleransi L_{low} adalah 0.5 dan toleransi L_{std} adalah 0.05. Berikut adalah beberapa contoh hasil akhir bentuk siluet dari objek yang telah dipisahkan dari background dan bayangan.



Gambar 7. Hasil Akhir Substraksi *Background* dan Deteksi Bayangan

Dari hasil akhir objek diatas ternyata didapat bahwa terdapat area objek yang tidak berwarna hitam. Untuk meneliti alasan terjadinya hal tersebut dilakukan deteksi terhadap objek warna dari putih hingga warna hitam.



Gambar 8.a, 9.b, 9.c dan 9.d
Hasil Akhir Substraksi *Background* dan Deteksi Bayangan

Gambar 9.a adalah sampel dari background dan 9.b adalah citra berobjek sebuah kartu gradasi warna. Kartu warna tersebut terdiri dari 11 warna dimulai dari putih, hitam 10%, hitam 20%, hitam 30%, hitam 40%, hitam 50%, hitam 60%, hitam 70%, hitam 80%, hitam 90%, dan hitam 100%. Gambar 9.c adalah gambar hasil substraksi dan deteksi bayangan sedangkan gambar 9.d hasil akhir dari objek yang terdeteksi. Berikut adalah tabel data nilai piksel pada area kartu warna.

Tabel Error! No text of specified style in document.2. Tabel Perbandingan
Data Piksel Kartu Warna

| No | Warna | Koordinat | PO | TMF | SB | IB | PB |
|----|-----------|------------|-----|-----|-------------|-------------|-------------|
| 1 | Putih | (271)(565) | 202 | 211 | bukan objek | bukan objek | bukan objek |
| 2 | hitam 10% | (271)(514) | 181 | 215 | objek | bayangan | bukan objek |
| 3 | hitam 20% | (271)(469) | 147 | 214 | objek | bayangan | bukan objek |
| 4 | hitam 30% | (271)(420) | 132 | 215 | objek | bayangan | bukan objek |

Tabel Error! No text of specified style in document.3. Tabel Perbandingan Data Piksel Kartu Warna (lanjutan)

| | | | | | | | |
|----|---------------|------------|-----|-----|-------|----------|-------------|
| 5 | hitam 40% | (271)(367) | 112 | 215 | objek | bayangan | bukan objek |
| 6 | hitam 40% | (271)(373) | 108 | 217 | objek | bayangan | Objek |
| 7 | hitam 50% | (271)(319) | 82 | 221 | objek | bayangan | objek |
| 8 | hitam 60% | (271)(274) | 65 | 218 | objek | bayangan | objek |
| 9 | hitam 70% | (271)(222) | 45 | 219 | objek | bayangan | objek |
| 10 | hitam 80% | (271)(171) | 40 | 221 | objek | bayangan | objek |
| 11 | hitam 90% | (271)(122) | 29 | 224 | objek | bayangan | objek |
| 12 | hitam 100% | (271)(75) | 18 | 220 | objek | bayangan | objek |

PO adalah piksel objek, TMF adalah nilai *background model* hasil *temporal median filter*, SB adalah label hasil substraksi *background*, IB adalah label hasil identifikasi bayangan dan PB adalah label hasil perbaikan bayangan.

Dari gambar 9 dan tabel 2 didapatkan objek dengan warna putih, hitam 10% dan hitam 30% tidak dikenali sebagai objek. Sedangkan hitam 50% hingga hitam 100% dikenali sebagai objek. Pada area hitam 40% ternyata tidak semua bagian dari area tersebut dikenali sebagai objek.

Berdasarkan pada pengamatan diatas disimpulkan bahwa objek yang memiliki warna dengan tingkat gradasi *grayscale* terhadap citra *background* pada kisaran dibawah 40% tidak akan dikenali sebagai objek.

5. Kesimpulan

Kesimpulan dari keseluruhan proses adalah :

- 1) Sistem ini dapat mendeteksi bentuk objek dengan metode substraksi *background* dan deteksi bayangan.
- 2) Parameter yang mempengaruhi kerja sistem adalah ukuran *neighbour*, nilai toleransi L_{NCC} , nilai toleransi L_{std} dan nilai toleransi L_{low} .

- 3) Semakin banyak citra sekuensial yang dipergunakan dalam pembuatan background model, semakin baik pula hasil dari proses dari *substraksi background*.
- 4) Ukuran *neighbour* yang semakin besar pada proses deteksi bayangan akan menghasilkan area bayangan yang lebih yang sempit.
- 5) Nilai toleransi L_{NCC} yang semakin besar pada proses deteksi bayangan akan menghasilkan area bayangan yang lebih yang sempit.
- 6) Nilai toleransi L_{std} yang semakin besar pada proses perbaikan bayangan akan menghasilkan area bayangan yang didefinisikan ulang sebagai objek yang lebih yang sempit.
- 7) Nilai toleransi L_{low} yang semakin besar pada proses perbaikan bayangan akan menghasilkan area bayangan yang didefinisikan ulang sebagai objek yang lebih yang luas.
- 8) Perbedaan tingkat gradasi warna *foreground* terhadap *background* yang akan dikenali sebagai objek adalah kisaran 40%.

Daftar Pustaka

- Castleman K.R. (1996). *Digital Image Processing*. New Jersey: Prentice Hall
- Forsyth, D.A., dan Ponce, J. (2003). *Computer Vision A Modern Approach*, New Jersey: Prentice Hall.
- Grest D., Frahm J.-M., and Koch R. (2003). *A color similarity measure for robust shadow removal in real time*. In *Vision, Modeling and Visualization*, pages 253–260
- Haritaoglu, D. Harwood, and L. Davis. (2004). *W4: Realtiesurveillance of people and their activities*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):809–830.
- Jacques Jr, Julio Cezar Silveira., Jung, Cl´audio Rosito., Musse, Soraia Raupp. (2005). *Background Subtraction and Shadow Detection in Grayscale Video Sequences*, University of Vale do Rio dos Sinos.
- Lo B.P.L. and Velastin S.A.. (2001). *Automatic congestion detection system for underground platforms*. Proc. ISIMP2001, pp. 158-161.
- MathWork. *rgb2gray : Convert RGB image or colormap to grayscale*, diakses tanggal 6 Maret 2010 di www.mathworks.com/help/toolbox/images/ref/rgb2gray.html
- Piccardi, Massimo. (2004). *Background subtraction techniques: a review*. IEEE International Conference on Systems, Man and Cybernetics.