

Deteksi Objek Bergerak pada Video dan Penerapan
Metode *Gaussian Mixture Model*



Oleh :

Dr. Moch Arief Soeleman, M.Kom

Dr. Muljono, S.Si., M.Kom

Muslih, M.Kom

Karis Widiatmoko, S.Si., M. Kom

Ricardus Anggi P. M.CS

PENERBIT PUSTAKA MAGISTER

SEMARANG 2019

ISBN 978-623-7140-22-1

Deteksi Objek Bergerak pada Video dan Penerapan Metode Gaussian Mixture Model

Penulis :

Dr. Moch. Arief Soeleman, M.Kom
Dr. Muljono, M.Kom
Muslih, M.Kom
Karis Widiatmoko, S.Si., M.Kom
Ricardus Anggi P, M.CS

**PENERBIT PUSTAKA MAGISTER
SEMARANG 2019**

Bahan Ajar

“Deteksi Objek Bergerak pada Video dan Penerapan Metode Gaussian Mixture Model”

Penulis :

Dr. Moch. Arief Soeleman, M.Kom

Dr. Muljono, M.Kom

Muslih, M.Kom

Karis Widiatmoko, S.Si., M.Kom

Ricardus Anggi P, M.CS

ISBN 978-623-7140-22-1



Hak Cipta @ M. Arief Soeleman, Muljono, Muslih, Karis W, Ricardus A

Cetakan Pertama, 2019

Diterbitkan pertama kali oleh

Penerbit Pustaka Magister

Jalan Pucangsari Timur IV/19 Pucanggading

Semarang

KATA PENGANTAR

Buku ini disusun sebagai pelengkap buku ajar pada mata kuliah visi computer untuk melatih mahasiswa dapat Menyusun project pada bidang computer visi dan penerapannya. Materi pada buku ini dimulai dengan mendeteksi objek bergerak pada video, kemudian penerapan threshold, segmentasi dan identifikasi gerak jatuh manusia. Semua penerapan pada kasus dimateri ajar ini menggunakan tools komputasi Matlab, tidak menutup kemungkinan dapat dikerjakan dengan tools lainnya.

Buku ini berisi tentang penerapan sistem cerdas berbasis video pengamatan yang dapat mendeteksi secara otomatis pada objek yang terpantau oleh kamera baik didalam ruangan seperti kantor, rumah maupun diluar ruangan. Pada buku ini dibahas beberapa penerapan beberapa metode untuk pelacakan objek, pengenalan objek manusia yang terjatuh maupun peningkatan kinerja metode algoritma.

Penulis berharap agar buku ini dapat memberikan tambahan pengetahuan bagi pembaca dan mahasiswa khususnya dalam melakukan pengembangan projek di bidang visi computer.

Semarang, Nopember 2019

Tim Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR .	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vi
BAB 1. Deteksi Objek Manusia Berbasis Kamera	1
1.1 Latar Belakang	1
1.2 Mengapa Perlu Deteksi	3
BAB 2. Metode Gaussian Mixture Model pada Threshold Deteksi Objek Bergerak	7 9
2.1 Deteksi Objek Bergerak	9
2.2 Penentuan Ambang Batas deteksi objek bergerak	9
2.3 Otsu Algoritma	11
2.4 Gaussian Mixture Model	14
2.5 Gray Threshold	21
BAB 3. Segmentasi Objek Manusia	21
3.1 Segmentasi objek Bergerak	21
3.2 Pemrosesan awal	22
3.3 Background subtraction	23
3.4 Klasterisasi objek bergerak dengan K-Means	26
3.5 Tahap Setelah pemrosesan	26
3.6 Eksperimen dan Pengujian	27
3.7 Segmentasi dengan FCM Clustering	28
... 3.8 Evaluasi Pengujian	33
BAB 4. Identifikasi Jatuh Manusia	33
4.1 Ekstraksi Fitur	34

4.2	Ekstraksi Fitur GLCM	38
4.3	Neural Network untuk Klasifikasi	44
4.4	Evaluasi Klasifikasi	46
BAB 5. Implementasi		47
5.1	Proses Identifikasi orang Jatuh	48
5.2	Fitur	49
DAFTAR PUSTAKA.		50

DAFTAR GAMBAR

Gambar 1.1	Metode dan Tahapan.....	6
Gambar 2.1	Alur Deteksi Objek Bergerak	8
Gambar 2.2	Frame Sequence Dataset	13
Gambar 2.3	Threshold objek Terdeteksi	14
Gambar 2.4	Hasil Treshold Gray	15
Gambar 2.5	Grafik MSE	18
Gambar 2.6	Grafik PSNR	18
Gambar 3.1	Alur Segmentasi	22
Gambar 3.2	Dataset Manusia.....	27
Gambar 3.3	Hasil Segmentasi.....	30
Gambar 3.4	Grafik MSE Segementasi.....	32
Gambar 3.5	Grafik PSNR Segmentasi	32
Gambar 4.1	Pelacakan Objek Manusia	40
Gambar 4.2	Proses Deteksi orang jatuh	42
Gambar 5.1	Identifikasi orang jatuh	47

BAB 1

Deteksi Objek Manusia Berbasis Kamera

1.1. Latar Belakang

Pesatnya pertumbuhan dan perkembangan dari teknologi digital serta ketersediaan pada peralatan berbasis video capture seperti kamera digital, telepon bergerak dengan kamera, mendorong terjadinya peningkatan secara cepat dalam perangkat penyimpanan, jaringan. Seperti yang telah disebutkan sebelumnya, manfaat pemakaian peralatan tersebut bahwa penggunaan kamera sirkuit tertutup (CCTV) untuk video pengamatan sangat penting bagi area publik maupun area privat telah meningkat sangat tajam. Dengan adanya peningkatan tersebut, maka diperlukan sebuah manajemen yang efektif dalam pengelolaan data video yang berukuran sangat besar dan terus bertambah. Ini menjadi sebuah masalah yang sangat signifikan dan sulit untuk mengurutkan, data secara manual yang didalamnya secara manual menandai makna pada deteksi, di dalam pencarian secara cepat.

Manusia merupakan bagian yang paling penting sebagai subjek yang melakukan pengamatan dalam berbagai bentuk situasi. Manusia biasa melakukan deskripsi dari bentuk yang tampak, seperti ukuran, perilaku atau aksi. Sebagai contoh, untuk mengenali manusia yang berjalan atau terjatuh, maka deteksi akan melakukan analisis terhadap semantik yang sedang terjadi pada perubahan gerakan manusia tersebut. Deteksi dari manusia yang bergerak pada tingkatan yang berbeda dapat membantu memahami semantik dari pergerakan yang dilakukan oleh manusia yang teridentifikasi.

Pemrosesan video melalui kamera pengamatan atau CCTV sangat menarik dan penting. Di dalamnya terdapat beberapa hal mulai dari resolusi yang rendah, kecepatan frame, kemudian variasi dari lingkungan seperti pencahayaan, perubahan setiap saat, adanya *noise* dan variasi arah pengambilan objek gambar.

Studi yang mempelajari tentang video deteksi pada sekumpulan data set berukuran besar dari konsep semantik untuk data video merupakan harapan untuk menghubungkan perbedaan yang ada (Hauptman, 2005), (Hauptman, et al., 2007), (Wang, et al., 2007), (Naphade & Smith, 2004). Menurut Hauptmann pada (Hauptman, et al., 2007) *semantic gap* antara low-level fitur dan informasi ke user dimasukkan ke dalam 2 kelompok gap, pertama pemetaan low-level fitur ke tingkat menengah konsep semantic, kedua pemetaan konsep tersebut ke dalam kebutuhan user. Deteksi atau pelabelan memastikan proses untuk menyelesaikan ke pendekatan low-level.

Deteksi video secara otomatis juga di kenal dengan nama "*video concept detection*" (Naphade & Smith, 2004), "*video semantic analysis*" (Snoek, et al., 2005) atau "*high level feature extraction*" (Kraaij & Over, 2005). Umumnya seperti diketahui bahwa metode pelabelan dalam gambar dapat dilakukan secara otomatis atau semi-otomatis. Penelitian terakhir, metode-metode dalam membantu untuk proses otomatis menghasilkan sekumpulan dari semantik pelabelan berbasis gambar pada konten visual diantaranya (Blei & Jordan, 2003), (Carneiro, et al., 2007), (Chan, et al., 2006), (Duyugulu, et al., 2002) metode ini pertama melakukan ekstraksi fitur aras rendah pada citra dan membuat sebuah matematika model untuk menghubungkan aras rendah pada isi citra sehingga disebut metode *content based image tagging* (CBIT)

Deteksi objek manusia terjatuh menggunakan kamera secara otomatis dapat dilakukan melalui ekstraksi beberapa fitur dari sebuah frame gambar. Dari pemetaan fitur tersebut akan dilakukan sebagai kunci pelabelan teks. Secara umum pemanfaatan fitur dapat dikelompokkan pada warna, tekstur dan bentuk.

Beberapa tahun terakhir, deteksi pada video mengalami kemajuan cukup signifikan. Banyak usulan untuk pendekatan dalam video deteksi, antara lain seperti semi pembelajaran terbimbing (Wang, et al., 2006), (Chapelle, et al., 2006), *active learning* (Naphade & Smith, 2004), (Boutell, et al., 2004) *multi-instance learning* (Dietterich, et al., 1997) dan *multi-label learning* (Meng, et al., 2009), semua metode telah berhasil meningkatkan kinerja dalam area masing-masing. Pada metode pendekatan terbimbing, model dari konsep semantik dibangun melalui sekumpulan pelatihan penamaan dan ketika ada penamaan sampel baru dapat diprediksi melalui model pembelajar. Sedangkan pada semi-pembelajaran terbimbing dan *active learning* merupakan dua pendekatan untuk mengatasi kesulitan pada data pelatihan yang tidak mencukupi pada model terbimbing. Metode *semi-supervised learning* mengeksploitasi tak berlabel data dengan asumsi yang pasti dan memiliki keyakinan untuk membangun model yang lebih akurat dibanding yang sudah dicapai pada model supervised.

2.1 Mengapa Perlu Deteksi

Mendeteksi objek pada video, khususnya perubahan objek manusia yang dari berjalan normal kemudian terjatuh sehingga dapat dikenali oleh mesin computer merupakan bagian penelitian yang sangat penting. Berbagai pendekatan untuk mengenali aksi pergerakan manusia pada video antara lain pendekatan berbasis bentuk, penampilan, skala, sudut pandang dan pose. Berdasarkan pada

penelitian sebelumnya khususnya pada identifikasi pada aksi manusia telah mencapai keberhasilan yang signifikan pada domain *spatio temporal* (Bobick A & Davis J, 2001). Sejumlah besar penelitian sebelumnya telah membahas pertanyaan tentang kategorisasi tindakan manusia dan analisis gerak. Salah satu bidang pekerjaan didasarkan pada perhitungan korelasi antara volume data video. Peneliti terdahulu pada penelitian (Bobick A & Davis J, 2001) menggunakan histori pergerakan citra yang mendeteksi gerakan dan bentuk untuk merepresentasikan gerakan. Mereka telah memperkenalkan gerakan deskripsi global pada energi citra dan gambar sejarah gerakan, yang digunakan sebagai template yang dapat dicocokkan dengan model yang tersimpan dari aksi yang dikenali. Metode yang diterapkan tergantung pada proses pengurangan latar belakang dan demikian tidak melakukan toleransi pergerakan kamera dan latar belakang dinamik.

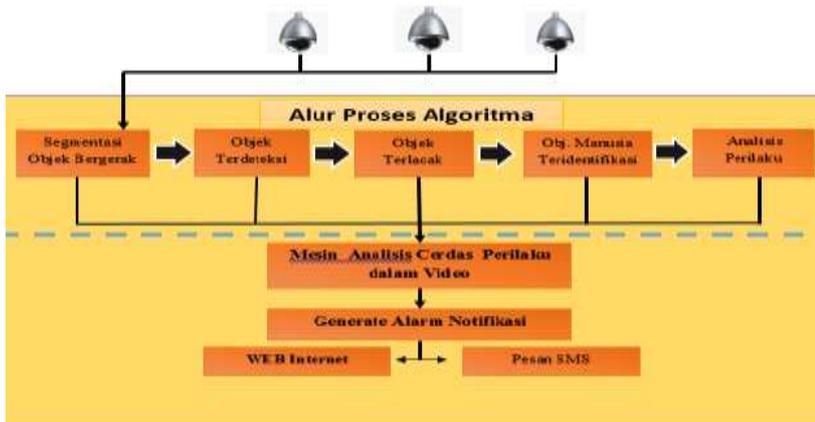
Dalam penelitian lain (Blank, et al., 2005) menunjukkan tindakan representasi sebagai ruang-waktu bentuk dan ekstrak fitur ruang-waktu untuk pengenalan tindakan, seperti ruang-waktu dinamika *saliency*, aksi lokal, membentuk struktur dan orientasi. Demikian pula, pendekatan ini bergantung pada pembatasan latar belakang statis yang memungkinkan mereka untuk mensegmen latar depan dengan menggunakan latar belakang pengurangan. Peneliti lainnya pada (Effros, et al., 2003) melakukan pengenalan tindakan dengan melakukan pengukuran hubungan antara *optical flow* dari video beresolusi rendah. Metode yang diusulkan terdiri dari beberapa tahapan, pertama melakukan proses segmentasi dan menstabilkan setiap objek manusia di dalam frame berurutan, serta adanya

intervensi manusia lebih lanjut untuk membubuhi keterangan atau label tindakan di setiap volume spasial-temporal yang dihasilkan.

Dalam penelitian yang berbeda pada (Imran, 2007) mengusulkan korelasi berbasis perilaku untuk menghitung kesamaan antara volume ruang-waktu yang memungkinkan untuk menemukan perilaku dan aksi dinamis yang memiliki kesamaan. Metode yang diusulkan oleh penulis membutuhkan penentuan sebuah template yang memiliki queri tindakan akan berkorelasi dengan video dalam database. Pada setiap piksel gradien ruang-waktu dari video yang sesuai dengan tampungan harus dihitung dan diringkas dalam sebuah matrik. Eigen dari matrik yang dihasilkan digunakan untuk menghitung kesamaan antara dua tampungan spasial temporal. Metode yang dilakukan tersebut memerlukan perhitungan yang signifikan karena korelasi prosedur antar setiap tampungan dalam urutan pengujian dan database video.

Pendekatan lain yang populer adalah untuk pelacakan bagian tubuh kemudian menggunakan lintasan gerak yang diperoleh untuk melakukan pengenalan tindakan. Peneliti lainnya dalam (Ramanan & Forsyth, 2007) melakukan pendekatan pada pengenalan tindakan dengan terlebih dahulu melacak manusia pada video dengan menggunakan prosedur struktur bergambar. Kemudian konfigurasi tubuh 3D dijadikan referensi dan dibandingkan sehingga menjadi pustaka gerakan 3D. Algoritma yang di implementasikan untuk memberikan pelabelan pada urutan pengujian, tetapi sangat tergantung pada hasil pelacakan. Estimasi dari 3D menimbulkan kemungkinan dalam pengenalan secara signifikan terhadap kesalahan karena sangat kesulitan dalam memecahkan problem ambiguitas.

Problem dari semantik deteksi video berkaitan sangat erat terhadap problem dasar pada kategori visual, seperti klasifikasi dan pengenalan secara spesifik kelas dari objek. Dari semua problem tersebut, problem dari semantik video deteksi yang belum dilakukan dalam penelitian adalah pembahasan untuk mengenali objek saat objek bergerak berubah secara tiba-tiba. Sebagai fokus utama dalam deteksi video objek bahwa kamera *surveillance* pada dasarnya hanya bisa merekam dan tidak bisa mengenali semantik dalam video. Sehingga hanya manusia yang bisa memahami semantik tersebut, dengan bantuan pembelajaran mesin atau dengan kecerdasan buatan maka kamera pengamatan akan menjadi cerdas sehingga mampu mengenali semantik dari video tersebut. Untuk mencapai kondisi yang efektif dari perubahan yang tampak maka fitur-fitur objek harus dapat dikenali dalam deteksi video objek. Dibawah ini urutan metode yang dilakukan untuk proses deteksi objek manusia khususnya terjatuh berbasis kamera CCTV. Adapun ilustrasinya dapat ditunjukkan sebagai berikut :



Gambar 1.1. Metode dan Tahapan

BAB 2

Metode Gaussian Mixture Model pada Threshold Deteksi Objek Bergerak

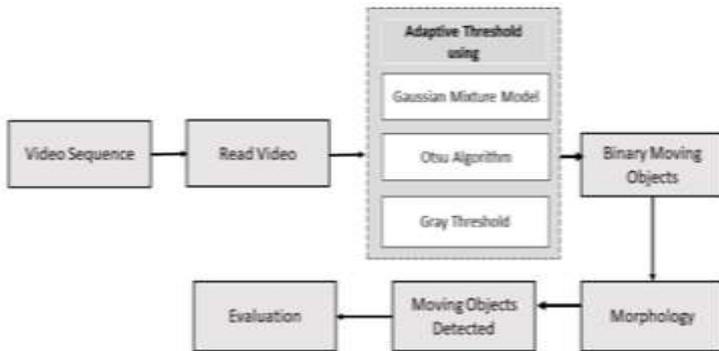
2.1 Deteksi Objek Bergerak

Dalam kamera system pengamatan secara otomatis, kamera dan peralatan sensor umumnya digunakan untuk melakukan pengawasan terhadap aktifitas sebuah lokasi atau tempat dengan tujuan memahami kejadian yang terjadi secara otomatis pada suatu lokasi. Pemahaman kejadian otomatis diharapkan menjadi fungsi sebagai pendeteksi aktifitas yang tidak wajar dan sebagai pengawasan keamanan. Sistem yang ada saat ini memiliki volume yang berukuran sangat besar pada video deteksi objek.

Deteksi objek bergerak pada video secara otomatis (Azeem, 2010) diharapkan akan menjadikan pencapaian secara efisien dan tercapainya deteksi otomatis. Deteksi secara otomatis dari sebuah kejadian akan dapat mengurangi lebar pita kanal yang diperlukan untuk memastikan proses pada saat penyimpanan dapat ditentukan sesuai yang diperlukan untuk disimpan atau tidak. Pemahaman pada level yang lebih tinggi dari sebuah kejadian terdiri dari level yang lebih rendah dari tugas visi komputer dapat dibentuk seperti penggunaan deteksi dari pergerakan yang tidak biasanya, pelacakan target, pelabelan pada bagian objek dan pemahaman pada interaksi antara manusia. Untuk merealisasikan beberapa pekerjaan tersebut, sangatlah penting membangun sebuah representasi dari sebuah *background* dan

membandingkan terhadap video *frame* baru atau berikutnya sebagai representasinya. Proses ini dinamakan dengan *background subtraction*.

Pada tahapan awal untuk melakukan indentifikasi terhadap objek bergerak dimulai proses untuk melakukan deteksi terhadap obyek bergerak digunakan pendekatan berbasis *background subtraction* (Elhabian, et al., 2008). Pendekatan ini merupakan proses untuk mendeteksi pergerakan atau perbedaan yang signifikan di dalam *video frame* (Piccardi, 2004) . Untuk algoritma *background subtraction* diimplementasikan metode *frame difference*. Metode *frame difference* merupakan perbedaan antara dua *frame* secara berurutan dalam sebuah *video* dan sebuah *reference background frame* untuk ekstrak daerah bergerak dari sebuah citra yang diciptakan sebuah citra berbeda. Adapun proposed metode berbasis algoritma gaussian mixture model sebagai berikut :



Gambar 2.1. Model Alur dari Deteksi dan *Threshold* Objek Bergerak

2.2. Penentuan Ambang batas deteksi objek bergerak

Penentuan ambang batas merupakan salah satu metode paling sederhana untuk memperoleh segmentasi gambar yang tajam (Otsu,

1979). Proses dari ambang batas menghasilkan gambar biner, sehingga piksel objek atau latar depan memiliki nilai 1 dan latar belakang memiliki nilai 0, dengan demikian objek akan muncul secara konsisten lebih cerah atau lebih gelap dari latar belakang. Dalam kondisi seperti itu, suatu objek gambar biner dapat diperoleh dengan *thresholding* gambar tingkat keabuan. Untuk mengekstrak objek dari latar belakang adalah dengan memilih nilai T ambang batas yang memisahkan mode ini, maka setiap titik (x, y) , dimana $f(x, y) \geq T$ disebut titik objek, jika tidak, titik ini disebut titik latar belakang. Persamaan *thresholding* $g(x, y)$ didefinisikan sebagai berikut:

$$g(x, y) = \begin{cases} 1; & \text{if } f(x, y) \geq T \\ 0; & \text{if } f(x, y) < T \end{cases} \quad (2.1)$$

2.3 Otsu Algorithm

Penentuan ambang batas metode Otsu merupakan metode yang ditemukan (Otsu, 1979) untuk segmentasi berdasarkan pada histogram citra, histogram ini menunjukkan nilai intensitas dari tiap piksel pada citra dalam dimensi 1. Sumbu x menyatakan level intensitas yang berbeda, sedangkan sumbu Y menyatakan jumlah piksel yang memiliki nilai intensitas tersebut Threshold dari algoritma Otsu inialisasi menggunakan parameter T . Nilai bentangan dari T berkisar antara 1 dan L , dimana nilai $L = 255$, probabilitas dari tiap piksel di dalam level ke- i dapat di jelaskan pada persamaan (2.2) :

$$p_i = n_i / N \quad (2.2)$$

n_i adalah angka dari pixel di dalam i^{th} level dan N merupakan banyaknya piksel. Rata-rata tingkat keabuan dari sebuah citra digunakan persamaan (2.3) :

$$\mu_T = \sum_{i=1}^{L-1} i \times p_i \quad (2.3)$$

Untuk *threshold* tunggal, Otsu membagi piksel ke dalam dua kelas $C_1 = \{0, 1, \dots, T\}$ dan $C_2 = \{T+1, T+2, \dots, L-1\}$. N menunjukkan probabilitas dari kelas dapat di komputasi dengan persamaan (2.4) :

$$\omega_1(T) = \sum_{i=1}^T p_i \quad \omega_2(t) = \sum_{i=1}^{L-1} p_i \quad (3.4)$$

$$\mu_1(T) = \sum_{i=1}^T \frac{i \times p_i}{\omega_1(T)} \quad \mu_2(T) = \sum_{i=1}^{L-1} \frac{i \times p_i}{\omega_2(T)} \quad (3.5)$$

Kemudian nilai dari T dapat di komputasi menggunakan persamaan (2.6).

$$T^* = \max_{1 \leq k < L} \alpha_B^2(T) \quad (2.6)$$

dengan persamaan (2.7) :

$$\alpha_B^2(T) = \omega_1(T) (\mu_1(T) - \mu_T)^2 + \omega_2(T) (\mu_2(T) - \mu_T)^2 \quad (2.7)$$

Adapun eksperimen yang dilakukan pada dataset menggunakan metode *background subtraction* untuk mendapatkan objek *foreground*.

2.4. Gaussian Mixture Models

Algoritma klusterisasi dapat dipertimbangkan sebagai teknik untuk klasifikasi piksel sebagai *foreground* dan *background*. Algoritma klusterisasi dapat dibagi ke dalam klusterisasi keras dan lunak. Klusterisasi keras diartikan sebuah objek keanggotaanya hanya pada

satu kluster, sedangkan klusterisasi lunak keanggotaanya dapat berada pada beberapa *cluster*.

Stauffer et.al dalam penelitiannya mengusulkan metode *Gaussian Mixture Model* (GMM) sebagai sebuah tipe *density model* yang terdiri dari komponen fungsi Gaussian. Komponen fungsi ini terdiri dari *weight* yang berbeda untuk menghasilkan *multi-model density*. Metode gaussian mixture model yang diusulkan Grimson sangat efisien dalam memisahkan antara latar depan dan latar belakang dari input frame sequence.

Jumlah model GMM yang digunakan mempengaruhi jumlah model *background*. Semakin besar jumlah model GMM yang dipakai semakin banyak model *background* yang dimiliki oleh suatu piksel. Terdapat beberapa tahap proses untuk metode ini yaitu tahap pencocokan input terhadap distribusi dan tahap pemilihan distribusi yang mencerminkan *background*. Pada GMM tiap piksel Di dalam tahap pencocokan terdapat tahap *update* parameter. Adapun model persamaan GMM sebagai berikut :

$$P(X_t) = \sum_{i=1}^K \omega_{it} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.8)$$

Parameter K pada persamaan (2.8) merupakan angka dari distribusi, sedangkan μ merupakan nilai rata-rata dari Gaussian pada satuan waktu t, dan Σ merupakan matrik kovarian pada i threshold pada gaussian dan ω merupakan bobot. Persamaan berikut merupakan fungsi Gaussian probabilitas density.

$$\begin{aligned} & \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} \Sigma^{1/2}} \exp^{\frac{1}{2}(x_t - \mu_t)^T \Sigma^{-1}(x_t - \mu_t)} \end{aligned} \quad (2.9)$$

Parameter n merupakan dimensi distrusi gaussian, $n=1$ ketika model background merupakan image gray, \sum_{it} merupakan citra warna dan $n=3$ merupakan citra RGB.

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha M_{(i,t)} \quad (2.10)$$

Ketika parameter M bernilai 1 maka match, dan sebaliknya bernilai 0. Nilai parameter μ dan α terupdated dengan persamaan berikut :

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho X_{i,t} \quad (2.11)$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^2 \quad (2.12)$$

$$\rho = \alpha\eta(X_t, \mu_{i,t}, \sigma_{i,t}) \quad (2.13)$$

$$B = \arg \min_b (\sum_{k=1}^b w_k > c_f) \quad (2.14)$$

Piksel-piksel yang merupakan probabilistic cocok terhadap model gaussian, apabila cocok terhadap latar belakang maka akan terklasifikasi piksel latar belakang.

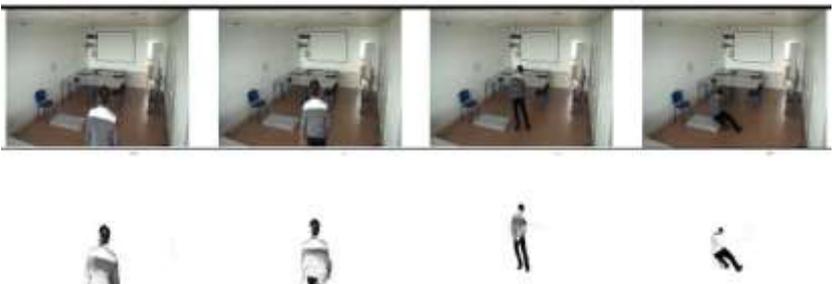
Menggunakan persamaan (2.10) pada tahap pembaruan parameter, nilai-nilai parameter GMM digunakan untuk memproses input berikutnya. Nilai yang diperbarui terdiri dari bobot, rata-rata, dan varian. Nilai berat diperbarui setiap saat. Setelah nilai bobot dinormalisasi, berat total semua distribusi tidak lebih dari 1. Kemudian nilai rata-rata distribusi diperbarui setiap kali ada nilai piksel yang cocok dengan nilai distribusi.

Menggunakan persamaan (2.12) dan (2.13) nilai standar deviasi suatu distribusi diperbarui setiap kali ada nilai piksel yang cocok dengan distribusi itu. Langkah selanjutnya dengan persamaan (2.14) piksel yang probabilistik cocok dengan model Gaussian, jika mereka bertentangan dengan latar belakang, mereka akan diklasifikasikan sebagai piksel latar belakang dan lawan mereka sebagai latar depan.



Gambar 2.2. Frame sequence pada dataset Video

Pada usulan metode yang dilakukan terdiri atas beberapa langkah untuk melakukan proses threshold pada objek bergerak seperti yang di ilustrasikan pada gambar 1. Pada tahap awal dimulai pemecahan frame pada video dataset, kemudian proses background subtraction untuk mendapatkan foreground, langkah berikutnya threshold pada objek bergerak dengan Gaussian Mixture Models yang menghasilkan binary moving objects pada frame yang dilanjutkan proses morphology hingga objek bergerak terdeteksi.





Gambar 2.3. Threshold objek Manusia Terdeteksi Bergerak dalam Gedung.

2.5. Gray Threshold

Citra frame biner [20] dikonstruksi dengan cara mengirik melalui skala abu-abu atau gambar berwarna dengan mengelilingi nilai piksel menjadi 1 atau 0 bergantung pada apakah mereka berada di atas atau di bawah nilai ambang batas. Ini biasa untuk memisahkan atau membagi objek bagian dalam gambar berdasarkan nilai piksel. Ambang operasi dasar beroperasi pada gambar sebagai berikut:

```

if  $Z(r,s) > \text{threshold}$ 
     $Z(r,s) = 1$ 
else
     $Z(r,s) = 0$ 
End
  
```

Sesuai skala frame gambar bergerak abu-abu, bahwa piksel memiliki substansi nilai intensitas tunggal, ambang tunggal harus dipilih, dan untuk gambar berwarna ambang terpisah dapat ditentukan pada setiap kedalaman berbeda. Adapun hasil percobaan menggunakan gray threshold dapat ditunjukkan sebagai berikut :



Gambar 2.4 Hasil threshold gray pada video frame

Pada hasil percobaan tampak bahwa hasil threshold terlihat sangat tidak sempurna objeknya, cenderung ke piksel yang tidak terlihat jelas.

Untuk mengukur kinerja dari metode Gaussian Mixture Model dalam penentuan ambang batas pada frame sequence dengan menggunakan *Mean Square Error* (MSE) (Spagnolo, et al., 2006) dan *Peak Signal to Noise Ratio* (PSNR) (Spagnolo, et al., 2006). Metode MSE dan PSNR bekerja dengan membandingkan antara objek yang dihasilkan dengan objek uji atau *ground truth*. Nilai MSE menunjukkan tingkat error dari proses penentuan ambang batas objek terdeteksi, angka yang didapatkan semakin kecil menunjukkan semakin baik. Sedangkan nilai angka PSNR yang besar menunjukkan metode penentuan ambang batas pada deteksi lebih baik. Adapun persamaan MSE dan PSNR yang diacu sebagai berikut :

$$MSE(X, Y) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [X(i, j) - Y(i, j)] \quad (2.15)$$

$$PSNR(X, Y) = 10 \cdot \log_{10} \left(\frac{\max^2}{MSE(X, Y)} \right) \quad (2.16)$$

Dari hasil eksperimen penentuan ambang batas objek bergerak pada dataset Le2i "Laboratoire Electronique, Informatique et Image" video surveillance dapat ditunjukkan dalam Tabel 2.1. Adapun hasil keseluruhan pada masing-masing MSE dapat ditunjukkan pada gambar Grafik 2.5. Pada gambar Grafik 2.6, menunjukkan bahwa deteksi terhadap objek bergerak yang berukuran sangat kecil metode ambang batas OTSU cenderung rendah dalam mendeteksi objek bergerak, sehingga hasil pada grafik menunjukkan posisi stagnan. Sedangkan pada metode GMM menunjukkan kinerja yang lebih baik meski objek berukuran sangat kecil.

Di bagian ini, hasil dari algoritma eksperimental yang telah diterapkan pada ambang batas objek bergerak untuk deteksi objek disajikan. Dataset yang digunakan terdiri dari 200 urutan bingkai. Untuk menilai praktik ambang adaptif kami, kami menggunakan dataset secara realistis dengan izin untuk Le2i "Laboratoire Electronique, Informatique et Image" status pengawasan video dengan menggunakan kamera tanpa pendamping. Video memiliki frame rate 25 per detik, dan dengan resolusi 320 x 240 piksel. Data video dijelaskan dengan kesulitan utama dari urutan bingkai yang masuk akal yang dapat ditemukan pada manusia maupun di ruang kantor langsung. Urutan bingkai terdiri dari fluktuasi pencahayaan, dan kesulitan yang berbeda seolah-olah oklusi atau latar belakang membingungkan dan bertekstur. Para aktor membawa aktivitas kebiasaan normal tertentu. Dataset terdiri dari sejumlah frame. Untuk tujuan penilaian, dengan informasi tambahan yang mewakili kebenaran dasar posisi deteksi dalam urutan gambar. Kemudian, setiap frame dari setiap video diberi label: pelokalan tubuh secara otomatis ditugaskan dengan kotak pembatas. Eksperimen menggunakan MATLAB 2016b dan berjalan pada CPU dengan prosesor

i5-8265U dengan RAM LPDDR3 8 GB. Berikut ini ditunjukkan kerangka hasil percobaan :

Tabel 2.1 Hasil MSE dan PSNR Metode Treshold

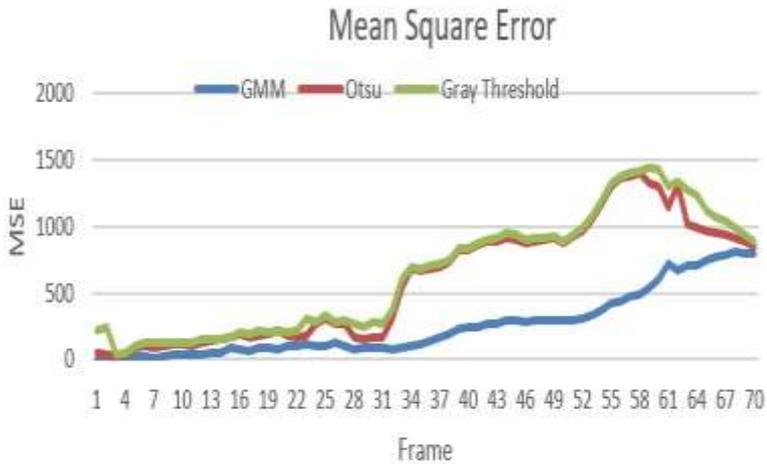
Evaluation	GMM Threshold	Otsu Threshold	Gray Threshold
MSE	257.18	595.36	645.39
PSNR	24.71	20.66	19.35

Hasil thresholding dalam bergerak manusia dalam dataset video, Gambar. 3 dan Gambar. 4 menunjukkan MSE dan PSNR ambang adaptif dalam deteksi manusia, berturut-turut. Diagram hasil menunjukkan pencapaian terbaik GMM (MSE = 89,11, PSNR = 28,63) pada frame 33. Rata-rata, MSE GMM adalah 257,18, Otsu adalah 595,36 dan Gray Threshold adalah 645,38. Juga, PSNR GMM adalah 24,71, Otsu adalah 20,66 dan Gray Threshold adalah 19,36.

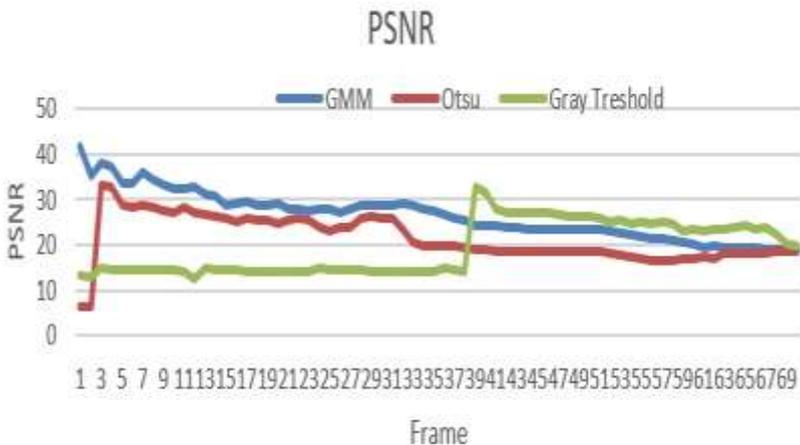
Berdasarkan hasil pengujian bahwa semakin rendah nilai Mean Square Error (MSE), proses thresholding menemukan objek dalam bergerak pada frame video semakin baik hasilnya atau memiliki tingkat kesalahan yang rendah, sedangkan jika tanda dari Peak Signal menjadi Noise Ratio (PSNR) lebih tinggi semakin baik kualitas ambang batas objek bergerak dalam bingkai video. Kemudian dapat ditunjukkan pada gambar 3 grafik yang menunjukkan tingkat keberhasilan proses ambang objek bergerak.

Berdasarkan hasil pengujian bahwa semakin rendah nilai *Mean Square Error* (MSE) maka proses thresholding deteksi objek bergerak pada frame video semakin baik hasilnya atau memiliki tingkat kesalahan yang rendah, sedangkan jika nilai *Peak Signal to Noise Ratio* (PSNR) yang semakin tinggi maka semakin baik kualitas threshold objek bergerak

pada video frame tersebut. Maka dapat dilihat pada gambar grafik 4 yang menunjukkan tingkat keberhasilan dari proses threshold objek bergerak .



Gambar 2.5. Grafik MSE GMM, OTSU dan Gray Threshold pada Dataset Deteksi Manusia



Gambar 2.6. Grafik PSNR GMM, OTSU dan Gray Threshold pada Dataset Deteksi Manusia

Adapun hasil dari kinerja PSNR pada eksperimen dataset deteksi manusia di tunjukkan pada gambar 2.6. Secara keseluruhan kinerja dari PSNR GMM lebih baik dibanding kinerja dari OTSU dan gray treshold. Kemungkinan alasan metode ambang batas Otsu memiliki kinerja yang lebih rendah dibandingkan dengan GMM adalah karena Otsu tidak menggunakan koherensi spasial dan struktur objek apa pun. Juga, ambang batas Otsu masih menggunakan asumsi dalam menggunakan kelas biner, yang mempartisi nilai histogram abu-abu ke dalam kelas biner. Karena dataset yang dapat dilihat pada gambar 3, itu menunjukkan bahwa video berisi lebih dari dua kelas segmen. Jadi, dalam bingkai nomor 179, ambang batas Otsu membagi tempat tidur dan orang menjadi satu karena warna tempat tidur dan warna pakaian memiliki warna abu-abu yang sama.

Pada analisis eksperimen terhadap tiga metode pada threshold ada kemungkinan alasan metode ambang batas Otsu memiliki kinerja yang lebih rendah dibandingkan dengan GMM adalah karena Otsu tidak menggunakan koherensi spasial dan struktur objek apa pun. Juga, ambang batas Otsu masih menggunakan asumsi dalam menggunakan kelas biner, yang mempartisi nilai histogram abu-abu ke dalam kelas biner. Karena dataset yang dapat dilihat pada gambar 3, itu menunjukkan bahwa video berisi lebih dari dua kelas segmen. Jadi, dalam bingkai nomor 179, ambang batas Otsu membagi tempat tidur dan orang menjadi satu karena warna tempat tidur dan warna pakaian memiliki warna abu-abu yang sama.

Kemungkinan alasan ambang Abu-abu berkinerja buruk dibandingkan dengan GMM adalah ambang Abu-abu gagal mendeteksi piksel non-tepi.

Dapat disimpulkan pada masalah threshold ini memberikan teknologi ambang adaptif pada menetapkan ambang batas objek bergerak terutama dalam deteksi manusia. Hasilnya menunjukkan bahwa pencapaian suatu objek yang bergerak dalam deteksi video menggunakan Gaussian Mixture Model lebih baik daripada dengan algoritma Otsu standar dan ambang batas abu-abu. Nilai rata-rata MSE adalah 257,18 dB lebih kecil dari yang lain dan nilai PSNR adalah 24,71 dB lebih tinggi dari Otsu dan ambang batas abu-abu.

BAB 3

SEGMENTASI OBJEK MANUSIA

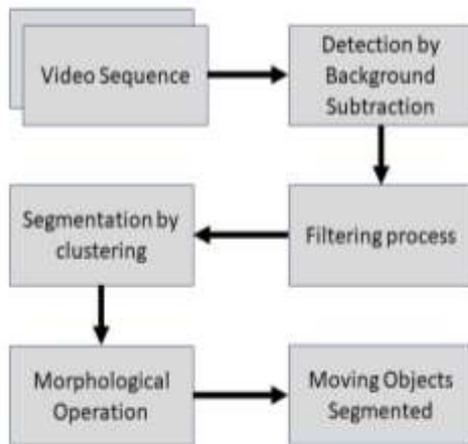
3.1. Segmentasi Objek Bergerak

Pada tahapan ini, setelah proses sebelumnya berupa deteksi pergerakan objek manusia dapat dideteksi, berikutnya merupakan tahapan proses untuk melakukan segmentasi objek bergerak. Metode yang digunakan pada proses segmentasi merupakan metode klusterisasi, antara lain K-means, Fuzzy C-Means dan Self Organizing Map. Ketiga metode tersebut merupakan metode klusterisasi yang banyak digunakan pada beberapa literatur dalam pemelitan (Jaffar, et al., 2009), (Lucia & Alfredo, 2008) dan (Chen, et al., 2008). Pada ketiga algoritma klusterisasi tersebut akan dilakukan analisis terhadap kinerja berupa rata-rata kesalahan segmentasi, kecepatan proses dan nilai dari PSNR dari proses segmentasi. Seperti yang telah dilakukan pada proses sebelumnya, bahwa proses dasar dari semua proses klusterisasi untuk segmentasi akan menggunakan basis *background subtraction* (Delp & Ka Ki, 2010). Kemudian operasi morfologi dilakukan pada beberapa proses untuk mendapatkan hasil yang optimal terhadap objek yang disegmentasi. Kemudian setelah tahapan morfologi akan ditentukan penentuan *region of interest* sebagai batas pemilihan area *ground truth* untuk pengujian hasil segmentasi objek manusia yang bergerak.

3.2. Pemrosesan Awal

Frame gambar hasil ekstraksi mempunyai ruang warna RGB. Setiap gambar akan dilakukan proses inialisasi, proses ini akan memberikan standarisasi dimensi gambar akan di-normalisasi menjadi 240 x 352. Kemudian *frame* gambar hasil masukan di ekstrak menjadi 3 bagian R,

G, dan B. Sebelum proses inisialisasi dilakukan, akan tentukan dulu *frame* gambar yang menjadi model *background*. *Frame* gambar pertama akan digunakan sebagai model *background awal*. Tahap *preprocessing* ini dilakukan sebagai persiapan dalam pendeteksian *foreground* menggunakan *background subtraction*. Adapun urutan proses dari segmentasi menggunakan pendekatan klusterisasi dapat di tunjukkan pada Gambar 3.1 *sebagai berikut* :



Gambar 3.1. Alur Proses Segmentasi dengan Klusterisasi

3.3. *Background Subtraction*

Pada tahap *background subtraction* berisi beberapa proses. Dimulai dengan pencarian nilai terbaik dari model *background* dengan *frame* masukan. Proses ini dimulai dengan menghitung jarak kedua citra. Digunakan fungsi jarak. yang kami tentukan berdasarkan rumus perhitungan jarak *Euclidean* sebagai berikut :

$$d(a, b) = [(a_R - b_R)^2 + (a_G - b_G)^2 + (a_B - b_B)^2]^{1/2} \quad (3.1)$$

Nilai piksel pada citra hasil pengurangan akan dibandingkan dengan nilai *threshold* yang telah ditetapkan. Piksel akan masuk kategori terbaik jika nilainya lebih kecil atau sama dengan nilai *threshold*. Piksel yang masuk kategori terbaik akan ditetapkan sebagai Background dan diberi nilai 0, sedang kondisi lainnya akan masuk sebagai *foreground* diberi nilai 1. Sesuai dengan algoritma klasterisasi yang diusulkan akan dijelaskan sesuai dengan metode klasterisasi masing-masing.

3.4. Klasterisasi Objek Bergerak dengan K-Means

Klaster merupakan kesatuan nilai-nilai dalam jarak tertentu pada kepadatan suatu daerah (relatif besar) dibandingkan dengan kepadatan nilai-nilai daerah sekitarnya. Teknik klasterisasi bermanfaat untuk segmentasi citra dan klasifikasi data yang belum diolah untuk menciptakan kelas-kelas. Warna diwakili dalam vektor 3 dimensi dari nilai titiknya. Masing-Masing komponen warna dihadirkan dalam warna merah, hijau dan biru (RGB). Perlu diketahui bahwa penggunaan penyajian ini, jika dua garis vektor adalah saling berdekatan, warna akan ditampilkan serupa, rata-rata dari dua garis vektor, jika warna yang akan ditampilkan sangat berbeda, maka akan diambil jalan tengah dengan menghadirkan suatu warna secara kasar dari warna aslinya. Acuan ini juga ketika rata-rata berbagai garis vektor RGB. Adapun cara-cara lain yang tidak membatasi untuk menghadirkan suatu warna dengan garis vektor 3 dimensi. Berikut untuk detilnya dari penjelasan di atas, bagaimana pilihan penampilan warna mempengaruhi hasil proses klasterisasi:

1. Langkah awal menetapkan data set dari algoritma yang akan digunakan (K-Means), yaitu dengan melakukan pengambilan nilai acak dari k.
2. Kemudian, nilai RGB dari tiap pixel diciptakan, dan menghasilkan nilai baru dalam 3-vektor.
3. Algoritma K-Means diterapkan pada dataset, menetapkan klusterisasi pusat k. Algoritma KMeans akan menetapkan adirkan k warna untuk menggambarkan frame tersebut.
4. Tiap-Tiap piksel citra dikonversi dalam suatu garis vektor RGB, dan ditampilkan menggunakan rata-rata dari kelompok warna yang dihasilkan.

Salah satu metode dalam klusterisasi adalah metode K-means, metode ini merupakan algoritma klusterisasi yang paling banyak digunakan dengan pertimbangan kecepatan prosesnya.

Langkah-langkah metode K-means dalam penentuan data cluster sebagai berikut (Chen , et al., 2008):

1. Tentukan k sebagai jumlah kluster yang di bentuk. Untuk menentukan banyaknya kluster k dilakukan dengan beberapa pertimbangan seperti pertimbangan teoritis dan konseptual yang mungkin diusulkan untuk menentukan berapa banyak kluster. Menentukan jumlah kluster (k), tetapkan pusat kluster sembarang dari objek-objek yang tersedia sebanyak k kluster, kemudian untuk menghitung pusat kluster ke-i berikutnya, digunakan rumus sebagai berikut :

$$v = \frac{\sum_{i=1}^n x_i}{n} \quad (3.2)$$

Adapun nilai parameter masing – masing :

v = pusat kluster

x_1 = objek ke-i

n = banyaknya objek/jumlah objek yang menjadi anggota cluster $i= 1,2,3\dots n$

2. Menghitung jarak setiap data ke pusat kluster, contohnya untuk menghitung jarak antara objek dengan centroid menggunakan *Euclidian*

$$D(X, Y) = \sqrt{\sum (X_i - Y_i)^2} \quad (3.3)$$

Parameter X_i = Objek x ke-i, Y_i = obyek y ke-i

3. Mengelompokan data ke dalam kluster yang dengan jarak yang paling pendek
4. Menghitung pusat kluster yang baru
5. Mengulangi langkah 2 sampai dengan 4 hingga sudah tidak ada lagi data yang berpindah ke kluster yang lain. Pengecekan konvergensi dilakukan dengan membandingkan matriks *group assignment* pada iterasi sebelumnya dengan matrik *group assignment* pada iterasi yang sedang berjalan. Jika hasilnya sama maka analisis algoritma kluster *k-means* sudah konvergen, tetapi jika berbeda maka belum konvergen sehingga perlu dilakukan iterasi berikutnya.

Pada langkah-langkah penentuan kluster metode K-means diatas, menghitung penentuan jarak setiap data ke pusat cluster sangatlah penting dimana keakuratan dalam penentuan kluster sangat dipengaruhi oleh penentuan jarak tersebut.

Algoritma ini dimulai dengan penyekatan masukan menunjuk ke dalam tetapan k secara acak. Kemudian melakukan kalkulasi rata-rata titik, atau pusat luasan, dari tiap data. Hal ini mengakibatkan suatu sekat baru dengan menghubungkan masing-masing dengan pusat luasan yang terdekat. Kemudian pusat luasan dihitung kembali untuk kluster yang baru, dan algoritma yang diulangi dua langkah sampai pemusatan,

yang mana diperoleh ketika poin-poin tidak lagi berpindah kluster atau sebagai alternatif pusat luasan adalah tidak lagi diubah.

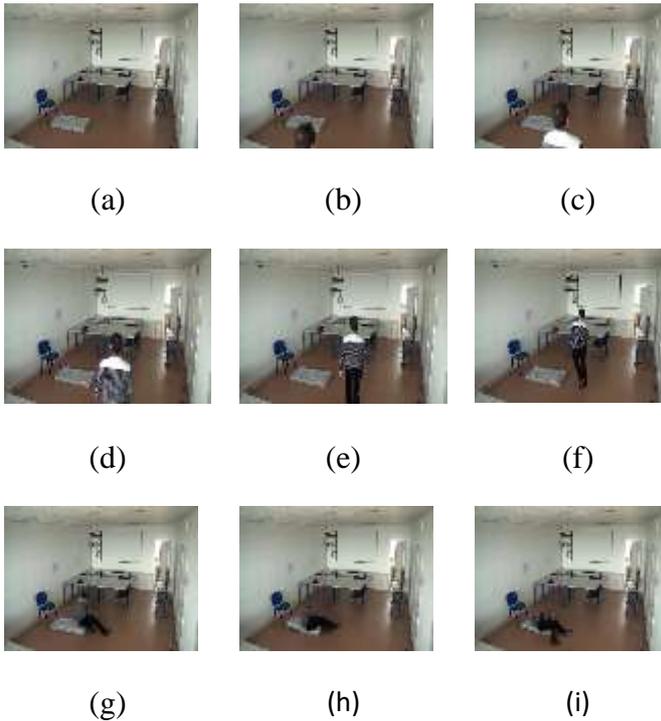
Walaupun algoritma harus selalu memusat, tidak ada batas pada banyaknya iterasi yang berulang-ulang sesuai yang diperlukan. Suatu implementasi dapat dipilih untuk menghentikan algoritma itu setelah suatu jumlah iterasi tertentu. Pemusatan tidaklah menjamin untuk menghasilkan suatu hasil yang maksimal secara menyeluruh.

3.5. Tahap Setelah Pemrosesan

Tahapan ini diaplikasikan teknik Morfologi dan filtering untuk memperbaiki hasil segmentasi dari *Background Subtraction*. Teknik yang dapat digunakan dalam *post-processing* seperti Morfologi (*Closing, Opening, Dilation, Erosion*), Median filter. Tahap terakhir adalah deteksi objek dan penandaan objek (foreground). Diaplikasikan algoritma *connected component* yang dilanjutkan dengan teknik *bounding box* yang akan memberi batas persegi objek yang terdeteksi pada frame.

3.6. Eksperimen dan Pengujian

Pada tahapan ini menjelaskan tentang teknik pengujian yang digunakan. Pelacakan objek dilakukan menggunakan dua metode yaitu *background Subtraction* standar dan *background subtraction* berbasis klusterisasi. Proses eksperimen dan pengujian menggunakan tiga dataset yang telah disiapkan dari tahap sebelumnya. Tahap ini akan dijalankan deteksi objek menggunakan *background subtraction* standar dan berbasis som pada setiap dataset. Setiap dataset berisi 100 frame. Hasil daeteksi akan diukur dengan membandingkan dengan citra *ground truth*. Dilakukan pengaturan parameter unutm mendapatkan kinerja dan akurasi terbaik.



Gambar 3.2. Dataset Manusia dalam Gerak

3.7 Segmentasi dengan FCM Clustering

Algoritma fuzzy c-means merupakan metode klasterisasi lunak yang menghasilkan sebuah matrik yang terhubung. Di dalamnya terdapat piksel yang terkoneksi dengan bebas terhadap semua klaster. Fuzzy C-means berusaha untuk mengurangi kesalahan dengan menghitung square error dengan persamaan sebagai berikut :

$$SSE = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m \leq \infty \quad (3.2)$$

Parameter u_{ij} merepresentasikan keanggotaan dari piksel x_i ke dalam j^{th} kluster, c_j dan j^{th} merupakan pusat kluster.

$$\sum_{i=1}^c u_{ij} = 1, 1 \leq j \leq n \quad (3.3)$$

$$u_{ij} \geq 0, 1 \leq i \leq c, 1 \leq j \leq n \quad (3.4)$$

$$\sum_{i=1}^n u_{ij} = 1, 1 \leq i \leq c \quad (3.5)$$

Algoritma FCM terdiri atas beberapa tahapan seperti berikut

1. Masukkan data dari citra atau video
2. Pilih jumlah kluster dengan nilai ε ($\varepsilon > 0$).
3. Hitung partisi matrik dengan persamaan (3.6)

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (3.6)$$

4. Lakukan perbaikan pusat kluster dengan persamaan (3.7).

$$c_j = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (3.7)$$

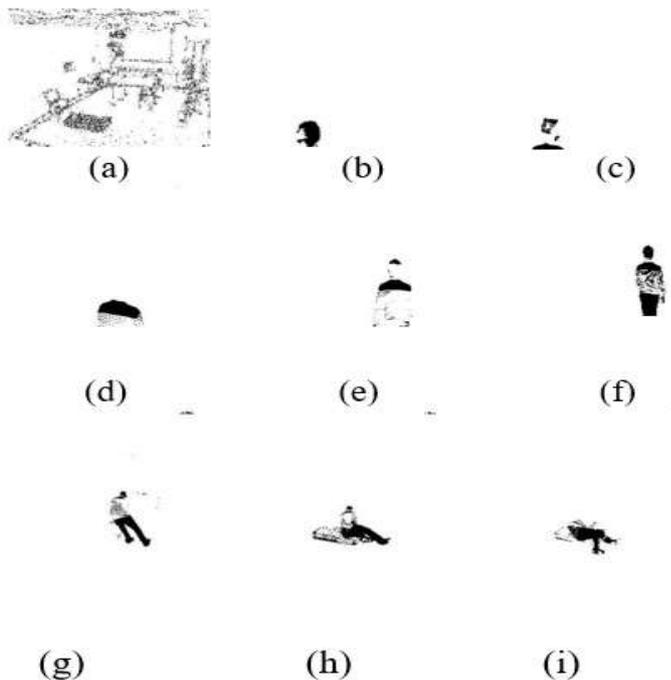
5. Lakukan perulangan proses 3 dan 4 hingga $\|c^k - c^{k+1}\| < \varepsilon$.

3.8 Evaluasi Pengujian

Analisis cluster memiliki ide dasar yaitu jumlah ruang antara evaluasi dua titik atau afinitas. Jarak adalah pengukur ekstensi antara dua titik objek; dimensi data adalah standar pengukuran; kedekatan

MSE menunjukkan semakin kecil tingkat kesalahan terhadap sehingga semakin baik hasil capaiannya. Adapun hasil dari keseluruhan eksperimen dari dataset diatas dapat ditunjukkan dalam grafik yang menunjukkan performance masing-masing hasil uji coba metode klastering yang berbeda-beda.

Adapun hasil dari keseluruhan eksperimen dari dataset diatas dapat ditunjukkan dalam grafik yang menunjukkan performance masing-masing hasil uji coba metode klastering yang berbeda-beda. Adapun hasil segmentasi pada objek bergerak dapat ditunjukkan pada gambar berikut ini :



Gambar 3.3 Hasil Segmentasi K-means

Adapun hasil dari penggunaan distance pada masing-masing hasil dapat ditunjukkan pada tabel berikut :

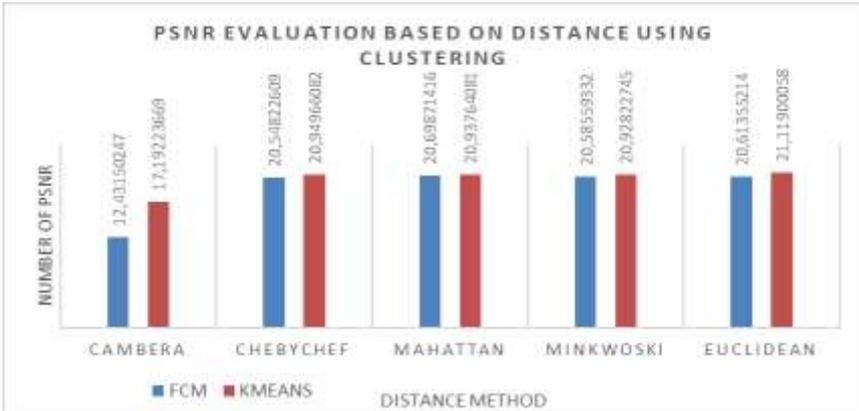
Tabel 3.1 Hasil MSE Segmentasi K-Means dengan Variasi Distance

Distance	Clustering	
	K-Means	FCM
Cambara	2273.698	4061.64
Chebychev	815.77	981.088
Manhattan	746.114	810.862
Minkwoski	747.391	905.75
Euclidean	722.099	876.301

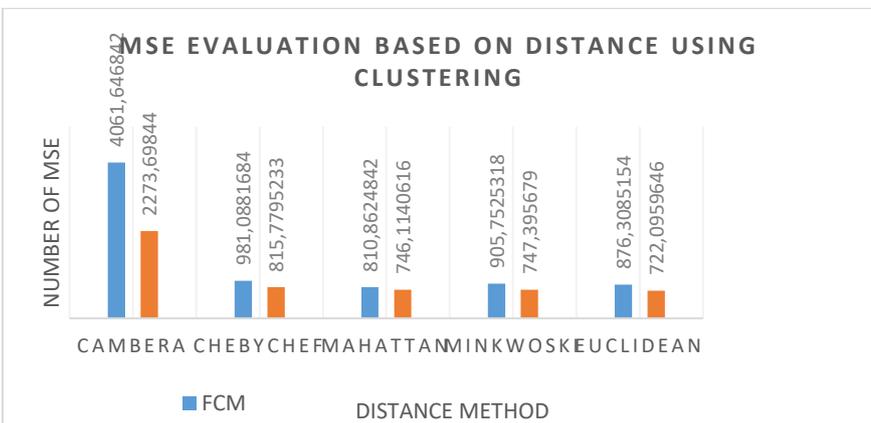
Tabel 3.2 Hasil PSNR Segmentasi K-Means dengan Variasi Distance

Distance	Clustering	
	K-Means	FCM
Cambara	17.192	12.431
Chebychev	20.949	20.548
Manhattan	20.937	20.698
Minkwoski	20.928	20.588
Euclidean	21.119	20.613

perbedaan yang diharapkan dalam frame sequence algoritma K-means yang diusulkan bekerja dengan baik. Kami melakukan evaluasi dengan melakukan MSE dalam metode yang dimaksudkan pada proses segmentasi untuk pengelompokan. Adapun hasil secara keseluruhan pada percobaan diatas dapat ditunjukkan pada gambar grafik dibawah pada gambar 3.4 dan gambar 3.5 ini sebagai berikut :



Gambar 3.4. Grafik PSNR K-means, FCM



Gambar 3.5. Grafik MSE K-means, FCM

Adapun hasil dari PSNR merupakan kebalikan dari nilai MSE, semakin tinggi nilai PSNR maka semakin baik hasil performance kinerja algoritma tersebut. Dari data Tabel 3.1 dapat dilihat bahwa kinerja dari algoritma Kmeans memiliki angka PSNR yang lebih tinggi dibanding dua algoritma klasterisasi FCM.

BAB 4

IDENTIFIKASI JATUH MANUSIA

Pada tahapan ini untuk melakukan proses identifikasi pada objek yang akan dikenali melalui mesin kamera berbasis CCTV adalah melakukan ekstraksi fitur terhadap objek yang telah disegmentasi pada bahasan sebelumnya.

4.1 Ekstraksi Fitur

Ekstraksi fitur mengubah data gambar mentah menjadi data numerik yang berisi karakteristik gambar. Karakteristik gambar adalah komponen penting dari persepsi visual manusia, yang diperoleh berdasarkan warna, bentuk, dan tekstur. Salah satu metode yang dapat digunakan untuk melakukan ini adalah dengan menggunakan grey-level co -acence matrix (GLCM) Metode GLCM mengubah gambar menjadi angka berdasarkan karakteristik teksturnya, yang menggunakan matriks yang terdiri dari k jumlah baris dan kolom l pada tingkat abu-abu. GLCM tidak tergantung pada variasi ukuran data gambar. Oleh karena itu, metode ini cocok untuk teknik ekstraksi fitur. Selain itu, GLCM mampu menunjukkan varietas fitur masing-masing gerakan manusia dalam bentuk angka sesuai dengan apa yang dilihat oleh para ahli.

GLCM mewakili hubungan antara piksel tetangga dalam gambar. Arah orientasi dan jarak spasial yang telah ditentukan memberikan nilai

matriks baru dalam bentuk matriks GLCM. Arah orientasi dalam komputasi GLCM adalah 00, 45, 90, dan 135. Hasil dari matriks GLCM adalah matriks kejadian bersama dan dapat digunakan untuk mendapatkan fitur momen-detik sudut, kontras, korelasi, homogenitas, dan entropi. Kelima fitur diperoleh dari gambar pada tingkat abu-abu 8-bit dari satu sudut orientasi sedangkan setiap lapisan dalam gambar skala abu-abu digunakan untuk mendapatkan fitur yang berbeda.

4.2. Ekstraksi Fitur Matrik Kookurensi Aras Keabuan (GLCM)

Pada tahapan ini, penulis menggunakan *Gray Level Co-occurrence Matrix (GLCM)* tekstur fitur untuk mengekstraksi fitur. GLCM yang merupakan tabulasi dari frekuensi atau seberapa sering perbedaan kombinasi dari ketajaman piksel (*gray level*) terjadi dalam sebuah *frame*. Komputasi dari dari image *frame* berbasis matematika ditunjukkan dalam persamaan (4.1), parameter $p(i, j)$ merupakan nilai (i, j) -th sebagai masukan dalam kookurensi matrik ternormalisasi, N menyatakan dimensi dari kookurensi matrik (angka level keabuan) dan $p_x(i)$ dan $p_y(i)$ merupakan margin probabilitas dengan persamaan berikut :

$$p_x(i) = \sum_{j=1}^N p(i, j), p_y(j) = \sum_{i=1}^N p(i, j) \quad (4.1)$$

μ merupakan nilai rata-rata dari μ_x dan μ_y :

1. *Energy*, merupakan fitur untuk mengukur konsentrasi pasangan intensitas pada matrik *co-occurrence*, semakin tinggi intensitas suatu image frame, maka nilai energinya semakin besar, yang

mempunyai fungsi untuk mengukur konsentrasi pasangan level citra pada matrik *co-occurrence*.

$$f_1 = \sum_i \sum_j P(i, j)^2 \quad (4.2)$$

2. *Sum of Square*

Sum of Square merupakan refleksi *degree* dari kejelasan dan kedalaman tekstur. Semakin dalam tekstur dari sebuah citra, maka nilai semakin besar IMnya. Ke-dalaman dari sebuah citra direfleksikan dengan sebuah efek blur nilai dari pasangan piksel merupakan perbedaan intensitas yang lebih besar dalam nilai IM, dan diagonalnya semakin besar pada GLCM.

$$f_2 = \sum_{i=0} \sum_{j=0} (i - j)^2 P(i, j) \quad (4.3)$$

3. *Entropy*

Entropi mengindikasikan kompleksitas citra. Semakin tinggi nilai entropi, semakin kompleks citra tersebut. Perlu diketahui, entropi dan energi berkecenderungan berkebalikan. Entropi juga merepresentasikan jumlah informasi yang terkandung di dalam sebaran data.

$$f_3 = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i, j) \log P(i, j) \quad (4.4)$$

4. *Correlation*

Korelasi dapat mengukur kemiripan dari sebuah elemen arah baris atau kolom dalam objek gambar pada GLCM. Adapun persamaannya sebagai berikut :

$$f_4 = -\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{(i - \mu_x)(j - \mu_y)}{\sqrt{\sigma_x \sigma_y}} p_{ij} \quad (4.5)$$

5. *Angular second moments*

$$f_5 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p_{ij}^2 \quad (4.6)$$

6. *Clusshade*

$$f_6 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i + j - \mu_x - \mu_y)^3 p_{ij} \quad (4.7)$$

7. *Cluspro*

$$f_7 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i + j - \mu_x - \mu_y)^4 p_{ij} \quad (4.8)$$

8. *Max pro:*

$$f_8 = \max(p_{ij}) \quad (4.9)$$

9. *Dissimilarity*

$$f_9 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i - j| p_{ij} \quad (4.10)$$

10. *Autocorrelation*

$$f_{10} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (ij) p_{ij} \quad (4.11)$$

11. *Inertia:*

$$f_{11} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j) p_{ij} \quad (4.12)$$

12. *Difference Entropy:*

$$f_{12} = \sum_{i=0}^{N-1} p_{x+y}(i) \log(p_{x+y}(i)) \quad (4.13)$$

13. *Sum Entropy:*

$$f_{13} = \sum_{i=0}^{2N-2} p_{x+y}(i) \log(p_{x+y}(i)) \quad (4.14)$$

14. *Sum Average* :

$$f_{14} = \sum_{i=0}^{2N-2} ip_{x+y}(i) \quad (4.15)$$

15. *Sum Variance* :

$$f_{15} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - \mu)^2 p_{ij} \quad (4.16)$$

16. INV

$$f_{16} = \sum_{i=0}^{N-1} \frac{p_{ij}}{1 + (i - j)} \quad (4.17)$$

17. IDN

$$f_{17} = \sum_{i=0}^{N-1} \frac{p_{ij}}{1 + |i - j|^2 / N} \quad (4.18)$$

18. *Inverse Difference Moment* :

$$f_{18} = \sum_{i=0}^{N-1} \frac{p_{ij}}{1 + (i - j)^2 / N^2} \quad (4.19)$$

Oleh karena itu, ketika kombinasi dari semua lapisan gambar dan sudut orientasi digunakan, ia menghasilkan elemen fitur 80 dari segmen dan sudut yang berbeda. Unsur-unsur terdiri dari 20 karakteristik lapisan R, G, dan B, sedangkan 20 sisanya diperoleh dari gambar skala abu-abu. Setiap gambar menghasilkan 80 fitur. Sebagai pembandingan dilakukan ekstraksi terhadap beberapa fitur seperti HOG, LBP, SFTA, STC dan STT. Berikut hasil ekstraksi dan pelatihan :

Tabel 4.1. Hasil Ekstraksi dan Validitas Pengenalan Fitur Pada Momentum Neural Network (BPNN)

Training Cycle	Learning Rate	Momentum	Fitur Objek	0,100	0,200	0,300	0,400	0,500	0,600	0,700	0,800	0,900	1,000
10	0,10	0,1	Zernike	0,686	0,681	0,678	0,667	0,681	0,677	0,678	0,678	0,681	0,682
10	0,10	0,4	HOG	0,767	0,724	0,736	0,769	0,761	0,764	0,758	0,749	0,754	0,754
10	0,10	1,0	INHU	0,646	0,646	0,646	0,646	0,645	0,645	0,646	0,645	0,645	0,646
10	0,10	0,2	LBP	0,618	0,665	0,637	0,573	0,643	0,629	0,659	0,650	0,661	0,655
10	0,10	0,3	GLCM	0,783	0,777	0,787	0,781	0,779	0,769	0,768	0,745	0,684	0,646
10	0,10	0,4	SFTA	0,854	0,859	0,855	0,860	0,857	0,852	0,857	0,858	0,859	0,858
10	0,10	0,1	STC	0,707	0,704	0,703	0,704	0,702	0,705	0,703	0,701	0,705	0,706
10	0,10	0,8	STT	0,687	0,678	0,687	0,691	0,688	0,687	0,685	0,691	0,685	0,688

4.3 Neural Network (BPNN) untuk Klasifikasi

Metode back-propagation propagation network (BPNN) yang kami gunakan untuk menguji hasil peningkatan gambar dengan mengklasifikasikan setiap gambar yang terdiri dari berbagai posisi gerakan berdasarkan objek terekstraksi. BPNN adalah algoritma pembelajaran terawasi, yang meliputi fase pelatihan dan fase pengujian. BPNN membutuhkan beberapa parameter seperti jumlah lapisan, nilai bobot, nilai bias, laju pembelajaran, momentum nilai, siklus pelatihan, dan fungsi gangguan. Metode BPNN dibagi menjadi tiga lapisan, yaitu lapisan input, lapisan tersembunyi, dan lapisan keluaran. Di lapisan input, data berada dalam tipe multivariat. Data yang digunakan dalam lapisan input diperoleh dari proses fitur dalam bentuk matriks dengan ukuran data N dikalikan dengan fitur m .

Matriks diproses dalam lapisan tersembunyi seperti yang ditunjukkan pada Persamaan (4.20), di mana lapisan tersembunyi

mengaktifkan fungsi sigmoid. Total neuron yang digunakan diperoleh dari rata-rata jumlah total atribut dan jumlah kelas yang kemudian ditambahkan ke 14. Hasil perhitungan proses lapisan tersembunyi adalah hasil di lapisan output (lihat Persamaan. (4.21)) . Hasil dari lapisan keluaran kemudian dibandingkan dengan label asli. Kesesuaian hasil yang dianalisis menunjukkan tingkat akurasi atau nilai objektif metode BPNN (lihat Persamaan (4.22)). Dalam proses tersebut, kesalahan prediksi digunakan untuk memperbarui setiap bobot saat ini, seperti yang ditunjukkan pada Persamaan. (4.23) dan Persamaan (4.24). Metode ini memperbarui setiap bobotnya berdasarkan Persamaan (4.25). Metode BPNN berhenti ketika pengulangan sama dengan jumlah siklus pelatihan atau tingkat kesalahan atau tujuan yang diharapkan.

$$z_i = b_i + \sum_{j=0}^n x_j v_{ij} \quad (4.20)$$

$$y_i = b_0 + \sum_{j=0}^n x_j w_{ij} \quad (4.21)$$

$$\delta_y = (t_k - z_k) f'(y) \quad (4.22)$$

$$\Delta v_j = \sum_{j=1}^m \delta_y w_{ij} \quad (4.23)$$

$$\Delta w = a \delta_y z \quad (4.24)$$

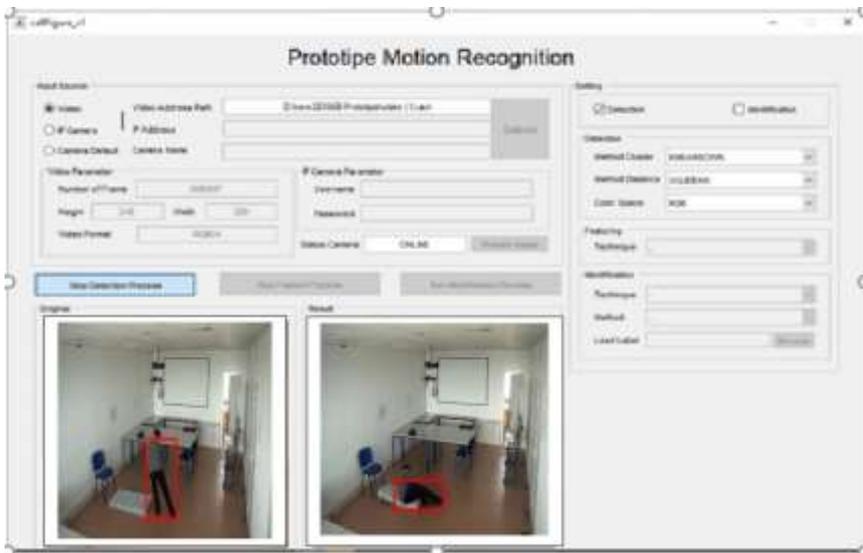
$$w_{new} = w_{old} + \Delta w \quad (4.25)$$

$$v_{new} = v_{old} + \Delta v \quad (4.26)$$

Evaluasi kinerja dari metode klasifikasi dilakukan dengan menghitung akurasi dan Kappa. Akurasi didefinisikan sebagai klasifikasi yang benar dari semua data yang diperoleh. Nilai akurasi diperoleh menggunakan Persamaan (4.27) dengan t dan n sebagai masing-masing jumlah data sampel diklasifikasikan dengan benar, dan n sebagai jumlah total data sampel.

$$accuracy = t/n \times 100 \quad (4.27)$$

Adapun model dari deteksi objek manusia pada frame video yang diujikan seperti gambar berikut:



Gambar 4.1. Pelacakan Objek pada Video

Adapun hasil pengujian BPNN pada learning rate berbagai fitur yang diekstraksi dapat ditunjukkan pada tabel 3 tersebut dibawah ini.

Tabel 4.2 Hasil Pelatihan data dengan BPNN

Training Cycle	Learning Rate	Momentum	Fitur Objek										
10	0,10	0,1	Zernike	0,686	0,674	0,677	0,649	0,675	0,660	0,662	0,662	0,655	0,646
10	0,10	0,4	HOG	0,719	0,652	0,635	0,646	0,646	0,630	0,646	0,654	0,649	0,650
10	0,10	1,0	INHU	0,646	0,646	0,646	0,646	0,587	0,646	0,558	0,646	0,646	0,617
10	0,30	0,2	LBP	0,623	0,602	0,637	0,605	0,580	0,478	0,475	0,476	0,410	0,384
10	0,10	0,3	GLCM	0,780	0,774	0,760	0,704	0,691	0,661	0,685	0,643	0,625	0,616
10	0,20	0,4	SFTA	0,854	0,854	0,807	0,774	0,768	0,742	0,740	0,738	0,731	0,726
10	0,10	0,1	STC	0,707	0,701	0,706	0,702	0,678	0,701	0,687	0,673	0,665	0,678
10	0,10	0,8	STT	0,680	0,661	0,640	0,638	0,629	0,661	0,657	0,652	0,645	0,645

Tabel 4.3 Hasil Ekstraksi dan Validitas Pengenalan Fitur Pada Training Cycle Neural Network (BPNN)

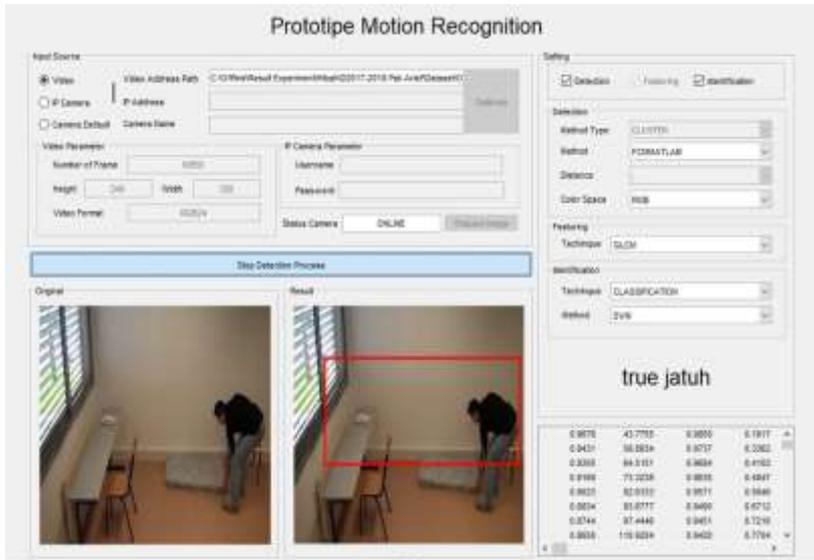
Training Cycle	Learning Rate	Momentum	Fitur Objek	10	20	30	40	50	60	70	80	90	100
100	0,10	0,1	Zernike	0,662	0,663	0,667	0,664	0,674	0,676	0,673	0,678	0,680	0,682
10	0,10	0,4	HOG	0,760	0,751	0,719	0,742	0,742	0,721	0,730	0,740	0,728	0,736
10	0,10	1,0	INHU	0,646	0,645	0,617	0,646	0,646	0,646	0,646	0,646	0,646	0,646
30	0,30	0,2	LBP	0,587	0,634	0,643	0,604	0,630	0,606	0,636	0,610	0,606	0,620
80	0,10	0,3	GLCM	0,735	0,761	0,769	0,774	0,771	0,773	0,779	0,780	0,774	0,780
100	0,20	0,4	SFTA	0,824	0,826	0,835	0,840	0,840	0,837	0,843	0,837	0,845	0,852
100	0,10	0,1	STC	0,664	0,670	0,686	0,689	0,691	0,700	0,696	0,698	0,703	0,706
100	0,10	0,8	STT	0,671	0,662	0,667	0,675	0,679	0,671	0,678	0,677	0,672	0,681

Kinerja metode BPPN yang diterapkan pada dataset asli ditunjukkan pada Tabel 4.2 hingga 4.3. Tabel 4.3 menunjukkan bahwa peningkatan jumlah siklus pelatihan memengaruhi kinerja akurasi. Ada beberapa nilai kinerja dalam menunjukkan peningkatan, tetapi masih di bawah rata-rata yang dihasilkan dari setiap percobaan.

Pada tahapan penelitian ini proses untuk melakukan ekstraksi fitur pada objek bergerak dalam hal ini manusia, klasifikasi dan testing terhadap objek bergerak dilakukan untuk mengenali gerakan jatuh objek manusia

Dari hasil fitur pada GLCM akan dilakukan klasifikasi dengan menggunakan LVQ.

BPPN yang digunakan pada tahapan ini untuk melakukan klasifikasi terhadap fitur GLCM yang telah diekstraksi. Dalam NN data yang ada pada fitur GLCM dianggap sebagai vektor input. Data dapat dinotasikan sebagai $V_i \in \mathbf{R}^d$ dengan nilai $i = 1, 2, \dots, n$, dimana nilai n merupakan banyaknya data. Dari data-data yang ada tersebut dilakukan pelatihan sesuai dengan pola jatuh manusia yang sesuai. Untuk memudahkan memberikan pelabelan pola-pola jatuh manusia maka setiap data dilabelkan dengan model berikut $X_i \in \{1, 2, 3, \dots, z\}$ dengan nilai $i = 1, 2, 3, \dots, n$, dimana nilai n menunjukkan banyaknya data dan z merupakan banyaknya pola yang dilakukan pelatihan. Pada tahapan identifikasi pada posisi manusia dalam keadaan terjatuh atau normal berdiri.



Gambar 4.2 Proses Deteksi orang Jatuh

Adapun algoritma dari Neural network dapat dijelaskan sebagai berikut

:

1. Tetapkan nilai bobot (w), maksimum epoch (MaxEpoch), estimasi nilai error minimum (Eps) dan Learning rate (α)
2. Nilai masukkan :
 - a. Input : $v(z,n)$;
 - b. Target : $T(1,n)$
3. Tentukan kondisi awal:
 - a. Epoch = 0;
 - b. Err = 1;
4. Lakukan proses jika (Epoch < MaxEpoch) atau (Learning rate > Eps)
 - a. Epoch = epoch + 1;

- b. Kerjakan untuk nilai $i = 1$ sampai ke n
- Tentukan nilai J sehingga $\|v - w_j\|$ bernilai minimum dinyatakan C_j
 - Lakukan perbaikan pada nilai w_j dengan ketentuan :
 - Jika nilai $T = C_j$ maka $w_j(\text{baru}) + \alpha [v - w_j(\text{lama})]$
 - Jika nilai $T \neq C_j$ maka $w_j(\text{baru}) = w_j(\text{lama}) - \alpha [v - w_j(\text{lama})]$
 - Kurangi nilai dari α

Keterangan notasi:

- v vektor pelatihan (v_1, v_2, \dots, v_n)
- T kategori bernilai benar untuk vektor pelatihan
- w_j vektor bobot unit output j (w_{ij}, \dots, w_{nj})
- C_j nilai kategori yang diwakili dengan unit output j
- $\|v - w_j\|$ nilai jarak Euclidean antara vektor input dan vektor bobot untuk unit output j

4.5. Evaluasi

Untuk melakukan pengujian keberhasilan dari performa hasil klasifikasi NN pada model yang diusulkan, kami menggunakan matrik konfusi sebagai representasi. Dengan menggunakan matrik konfusi hasil akurasi dari klasifikasi yang diperoleh dapat dilakukan perhitungan dengan persamaan (4.28) sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.28}$$

Adapun penjelasan dari empat kondisi pada konfusi matrik dapat dijelaskan sebagai berikut :

- True Positif (TP) merupakan sebuah nilai yang bernilai positif dimana hasil klasifikasi menunjukkan nilai kebenaran positif.
- False Negative (FN) menunjukkan nilai dimana jika hasil prediksi menunjukkan nilai salah
- True Negative (TN) merupakan sebuah nilai negatif yang diklasifikasikan tepat sebagai nilai negatif

Tabel 4.4. Matrik Konfusi

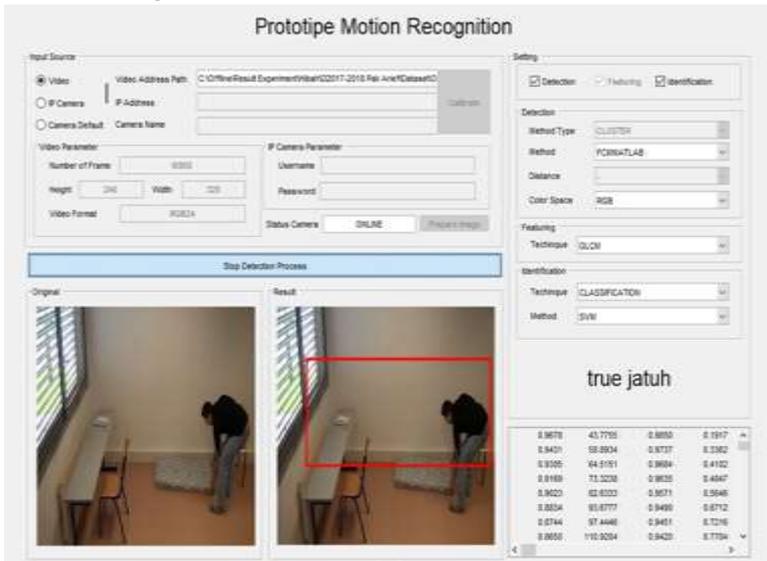
Prediksi	Pengamatan	
	Aktual Positif	Aktual negatif
Positive	TP	FP
Negative	FN	TN

- False Negative (FP) merupakan hasil prediksi menunjukkan negative
- dan di hitung sebagai False Positif. Dalam eksperimen, dataset dibagi menjadi *training* dan *testing* dengan *split ratio* 70:30. Sebanyak 70% dari data akan dijadikan sebagai training dan 30% digunakan sebagai *testing*.
- Dalam tabel tersebut menghasilkan akurasi sebesar 87.77%. Angka tersebut didapat dari total jumlah TP dan TN dibagi dengan jumlah data yang diujikan. Nilai TP dan TN tersebut tersusun diagonal sesuai dengan kategorinya masing-masing.

BAB 5

Implementasi

Dari paparan diatas bahwa pada penelitian untuk melakukan deteksi orang jatuh berbasis kamera CCTV didapatkan hasil analisis sebagai berikut:



Gambar 5.1 Identifikasi orang Jatuh

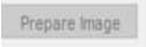
- Bahwa gaussian mixture model dapat menjadi pendekatan dalam melakukan deteksi objek khususnya dalam penentuan ambang batas atau threshold objek terdeteksi *background subtraction* dapat digunakan untuk melakukan deteksi pada pergerakan objek bergerak. Dengan penggunaan GMM adaptive secara adaptif dapat dilakukan sehingga objek yang terdeteksi pada beberapa kondisi cahaya, dapat didapatkan hasil yang maksimal. Adapun sebagai pembanding untuk penentuan ambang batas

dilakukan perbandingan dengan otsu dan gray threshold algoritma. Secara empiris nilai erorr (MSE) dan PSNR menunjukkan bahwa GMM memiliki nilai yang lebih baik dari segi tingkat erorr yang kecil dibanding otsu dan gray threshold.

- b. Bahwa untuk melakukan segmentasi objek manusia khususnya untuk proses identifikasi apakah objek terjatuh atau dalam kondisi normal, telah berhasil dilakukan dengan pendekatan klatsterisasi menggunakan k-means dan fuzzy c-means. K-means memiliki kecepatan proses dibanding dengan fuzzy c-means. Pada proses segmentasi dilakukan pengujian terhadap jarak yang ditambahkan pada k-means dan fuzzy c-means. Dari hasil empiris nilai mse dan psnr hasil segmentasi objek manusia menunjukkan k-means lebih baik dibanding fuzzy pada beberapa pengujian distance yang berbeda.
- c. Pada hasil ekstraksi fitur terhadap objek manusia yang telah disegmentasi menggunakan GLCM, ada fitur-fitur yang menjadi referensi untuk data pelatihan menggunakan Neural Network Untuk melakukan pengujian terhadap kinerja fitur glcm pada proses pengenalan orang jatuh dilakukan ekstraksi beberapa fitur lainnya seperti LPB, HOG, SFTA dan lainnya. Dengan bantuan pendekatan supervised learning menggunakan metode *Neural Network* (NN) hasil keakurasian berdasarkan label yang telah di lakukan didapatkan metode *Neural Network* menghasilkan akurasi sebesar 87.77%.

Angka sebesar 87.77% masih bisa ditingkatkan lagi, sehingga proses identifikasi untuk orang jatuh berbasis kamera akan lebih akurat.

5.2 Pengoperasian

	<p>Pada menu option grup ini, untuk memilih sumber data yang akan diamati, ketika pilihan ini memilih option Video maka akan ditampilkan alamat lengkap video tersebut.</p>
	<p>Tombol run detection, akan menjadi enabled ketika proses calibrate telah diset, sehingga proses pengurangan latar depan dengan latar belakang dapat dieksekusi dengan benar. Jika tidak memilih ini maka tidak diproses.</p>
	<p>Tombol chek box ini untuk memilih proses yang dijalankan, maka klik option Detection.</p>
	<p>Tombol kombo box ini, khususnya pada method cluster digunakan untuk memilih metode klasterisasi objek bergerak pada video. Ada 3 metode yang dapat dipilih salah satunya k-means.</p>
	<p>Tombol ini digunakan untuk menginisialisasi citra yan telah siap dijalankan untuk deteksi.</p>
	<p>Pada pilihan ini akan otomatis menampilkan alamat dataset berasal, setelah memilih video dan ditampilkan pada kotak original.</p>

DAFTAR PUSTAKA

- Amer, A.**, 2002. New binary morphological operations for effective low-cost boundary detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(2).
- Azeem, M.**, 2010. Modified background subtraction algorithm for motion detection in surveillance systems. *Journal of American Arabic Academy for Science and Technology*, 1(2), pp. 112-123.
- Barrat, S. & Tabbone, S.**, 2008. *Classification and Automatic annotation extension of images using Bayesian network*. s.l., s.n.
- Blank, M., Gorelick, L., Shechtman, E. & Irani, M.**, 2005. *Actions as space time-shapes*. s.l., s.n.
- Blei, D. & Jordan, M., 2003. *Modeling Annotated Data*. s.l., s.n.
- Bobick A & Davis J**, 2001. The recognition of human movement using temporal template. *Pattern Analysis and Machine Intelligent, IEEE*, 23(3), pp. 257-267.
- Boutell, M., Luo, J. & Shen , X.**, 2004. Learning multi-label scene classification [J]. *Pattern Recognition*, 2004, 37: 1757-1771.. *Pattern Recognition*, Volume 37, pp. 1757-1771.
- Bovic, A.**, 1998. *The hand book of image and video processing*. s.l.:Academic Press.
- Carneiro, G., Chan, A., Moreno, P. & Vasconcelos, N.**, 2007. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3), pp. 394-410.
- Cha & Hyuk, S.**, 2007. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods In Applied Sciences*, 1(4), pp. 300-307.

- Chan, A., Moreno, P. & Vasconcelous, N., 2006.** *Using Statistics to Search and Annotate Pictures: An Evaluation of Semantic Image Annotation and Retrieval on Large Databases.* s.l., s.n.
- Chapelle, O., Zien, A. & Scholkopf, B., 2006.** *Semi-Supervised Learning.* s.l.:MIT Press.
- Chen , T. W., Chen, Y. L. & Chien, S. Y., 2008.** *Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space.* s.l., s.n.
- Chen, S. C. & Zhang, D. Q., 2004.** Robust image segmentation using FCM with spatial con-straints based on new kernel-induced distance measure. *IEEE Trans. Syst. ManCybern*, 34(4), pp. 1907-1916.
- Ciro D'Elia, D., Simona, R., Maurizio , A. & Brun, 2014.** SAR Image Classification Through Information-Theoretic Textural Features MRF Segmentation, and Object-Oriented Learning Vector Quantization. *IEEE Journal of selected Topics in applied earth observations and remote sensing* , 7(4), pp. 1116-1126.
- Collins, R. et al., 2000.** *System for Video Surveillance and Monitoring,* s.l.: Robotics Institute, Carneige Mellon University.
- Delp, E. & Ka Ki, N., 2010.** *Object Tracking initialization using automatic moving object detection.* s.l., s.n.
- Dollar, P., Rabaud, V., Cottrel, G. & Belongie, S., 2005.** *Behaviour Recognition via Sparse Spatio-Temporal Features.* s.l., s.n.
- Effros, A., Alexander , B., Mori, G. & Malik, J., 2003.** *Recognizing Action at a Distance.* Nice, France, s.n.
- Fathi A & Mori G, 2008.** *Action recognition by learning mid-level motion features.* s.l., s.n.
- Hauptman, A., 2005.** *Lessons for the future from a decade of informedia video analysis research.* Singapore, s.n., pp. 1-10.

- Hauptman, A. et al.**, 2007. Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news. *IEEE Trans. Multimedia*, vol. 9, no. 5,, pp. 958-966.
- Hosseini, M. & Moghadam, M.**, 2012. Fuzzy rule-based reasoning approach for event detection and annotation of broadcast soccer video. *Appl. Soft. Computer* .
- Imran, J. N.**, 2007. Learning self-similarities for action recognition using conditional random fields. *InTechopen*, pp. 1-15.
- Irani, M. et al.**, 1996 . Efficient representations of video sequences and their applications,. *Signal Process Image Communication*, 8(4), pp. 327-351.
- Liu, H., Jeng, B., Yih, J. & Yu, Y.**, 2009. *Fuzzy C- means algorithm based on standard mahalanobis distance*. Huangshen , P.R. China., s.n.
- Liu, J., Luo, J. & Shah, M.**, 2009. *Recognizing Realistic Actions from Videos in the Wild*. s.l., s.n.
- Li, Y. et al.**, 2009. *A framework for video event detection using weighed SVM classifiers*. s.l., s.n.
- Lucia, M. & Alfredo , P.**, 2008. A Self-Organizing Approach to Background for Visual Surveillance Applications. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 17(7), pp. 1168-1177.
- Meng, W. et al.**, 2009. Unified Video Annotation via Multigraph Learning. *IEEE Transaction On Circuits and Systems For Video Technology*,, 19(5).
- Naphade, M. & Smith, J.**, 2004. *On the detection of semantic concepts at TRECVID*. New York, s.n.
- Otsu, N.**, 1979. A threshold selection method from gray-level histograms. *IEEE Trans.Syst. Man Cybern.*, 9(1), pp. 62-66.

Palma, D., Ascenso, J. & Pereira, F., 2004. *Automatic text extraction in digital video based on Motion analysis.* Porto, s.n.

Ramanan & Forsyth, 2007. Recognizing Human and their activities.

Robert, H. M., K, S. & Its'Hak, D., 1973. Texture Features for Image Classification. *IEEE Transactions On Systems, Man and Cybernetics*, 6(3), pp. 610-621.

Shao-Yi, C., Kai, C. W., Hsiang, T. Y. & Chen, H.-Y., 2013. Video object segmentation and tracking framework with improved threshold decision and diffusion distance. *IEEE Trans. On Circuits and Systems For Video Technology*, 23(6), pp. 921-934.

Soeleman, M. A., Hariadi, M. & Purnomo, M. H., 2012. *Adaptive threshold for background subtraction in moving object detection using fuzzy c-means.* Philippine, s.n.

Spagnolo, P., Orazio, T., Distante & A, M. L., 2006. Robust foreground segmentation from color video sequence using background subtraction with multiple threshold. *Journal Image and Vision*, Volume 24, pp. 441-423.

Stauffer, C. & W.E.L, G., 2000. Learning Pattern of Activity using Real Time Tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 747-757.

Wang , M. et al., 2009. Unified Video annotation via Multigraph Learning. *IEEE Trans. Circuits Syst. Video Tech. , 19(5).*

Wang, D., Zhang, B., Li, X. & Li, J., 2007. *Video search in concept subspace: A text-like paradigm.* Amsterdam, The Netherlands, s.n., pp. 603-610.

Wang, M. et al., 2006. *Automatic Video Annotation by Semi-Supervised Learning with Kernel Density Estimation.* Santa Barbara, California, USA, s.n.

ISBN 978-623-7140-22-1



9 786237 140221