

FISH MOVEMENT TRACKING MENGGUNAKAN METODE GAUSSIAN MIXTURE MODELS (GMM) DAN KALMAN FILTER

Hafizhun Alim¹, Mulyono², Muhammad Eka Suryana³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta, Jakarta Timur, Indonesia

hafizhun.alim93@gmail.com¹, mulyono@unj.ac.id², eka-suryana@unj.ac.id³

Abstrak— Potensi industri perikanan Indonesia merupakan yang terbesar di dunia, baik perikanan tangkap maupun perikanan budidaya. Salah satu masalah yang sering dihadapi oleh pelaku usaha budidaya ikan adalah pada saat proses penghitungan ikan yang masih menggunakan cara-cara manual. Penelitian ini bertujuan untuk melakukan proses penghitungan ikan dengan lebih mudah dan efisien. Dalam penelitian ini dilakukan proses *tracking* objek ikan menggunakan *Kalman Filter* yang dibantu oleh GMM sebagai metode deteksinya. Keluaran yang didapat menunjukkan bahwa sistem mampu melakukan pelacakan dengan *error* yang kecil, serta menghasilkan rata-rata jumlah objek yang mendekati rata-rata jumlah objek sebenarnya pada video kategori latar belakang sederhana.

Kata Kunci— *Object Detection Fish Movement Tracking, Ikan, Object Tracking, Peternakan Ikan.*

I. PENDAHULUAN

Potensi industri perikanan Indonesia merupakan yang terbesar di dunia, baik perikanan tangkap maupun perikanan budidaya. Di Indonesia sendiri, Badan Pusat Statistik (BPS) mencatat ada lebih dari 1,5 juta usaha budidaya ikan dengan total produksi mencapai 16 juta ton di tahun 2016. Dengan proporsi konsumsi protein yang berasal dari ikan/udang/cumi/kerang mencapai 13,27 persen pada tahun 2019, masih di atas konsumsi daging sebesar 6,54 persen serta konsumsi telur/susu sebesar 5,50 persen [1].

Salah satu masalah yang sering dihadapi oleh pelaku usaha budidaya ikan adalah pada saat proses penghitungan yang masih menggunakan cara-cara tradisional seperti menghitung satu-perstau atau menimbang berat wadahnya. Kedua cara tersebut tentu sangat tidak efisien dan efektif. Penghitungan ikan secara manual dapat merusak fisik ikan dan sangat tidak hemat waktu, sementara penghitungan ikan dengan menggunakan berat tidaklah selalu akurat [2].

Dari permasalahan di atas, diperlukan sebuah alat yang dapat menghitung jumlah ikan secara otomatis. Affandy [3] melakukan penelitian terhadap proses penghitungan ikan, yang membuat sebuah *prototype* alat penghitung ikan otomatis dengan menggunakan sensor *infra-red*, yaitu dengan cara mengalirkan ikan secara satu persatu melalui sensor tersebut, yang keluarannya kemudian ditampilkan melalui layar LED. Dari penelitian tersebut, didapatkan hasil akhir yang dapat

diterima, dengan tingkat kepercayaan sebesar 95%. Efisiensi waktu juga menjadi lebih baik, untuk setiap 100 ekor bibit ikan mas, dapat dihitung dalam waktu 100 detik yang semula 190 detik.

Al-Amri [2] juga melakukan penelitian yang sama. perbedaannya hanya di sensor yang digunakan, yaitu sensor *proximity*. Dari hasil uji coba didapatkan hasil yang baik, dengan persentase error sebesar 4,07%. Pengukuran waktu juga tercatat mengalami perubahan yang cukup signifikan dengan perbandingan waktu, 20 menit per 1000 bibit ikan dengan cara manual, dan 228 detik per 1000 bibit ikan dengan menggunakan alat tersebut.

Penghitungan ikan menggunakan alat fisik seperti di atas bukanlah tanpa kekurangan. Beberapa kekurangan tersebut antara lain adalah, proses penghitungan dilakukan dalam kondisi *closed and controlled environment*, kemudian sensor yang masih sensitif terhadap perubahan lingkungan, dan juga biaya perawatan alat yang perlu dikeluarkan jika terdapat sensor yang rusak. Maka dari itu, dibutuhkan sebuah sistem yang lebih fleksibel, mudah digunakan, *robust* terhadap perubahan lingkungan, serta membutuhkan biaya yang sangat minim ataupun tidak sama sekali (*zero-cost*).

Dari permasalahan di atas, penulis mengusulkan untuk melakukan proses penghitungan ikan melalui proses pelacakan pergerakan objek ikan (*fish movement tracking*) menggunakan metode *Kalman Filter* dengan GMM sebagai metode deteksi objeknya. Hasil yang diharapkan adalah sistem mampu melakukan pelacakan lebih dari satu objek sekaligus secara akurat dan juga menghasilkan jumlah rata-rata objek yang mendekati jumlah rata-rata sebenarnya.

II. KAJIAN PUSTAKA

A. Background Subtraction Menggunakan Gaussian Mixture Models (GMM)

Background Subtraction adalah metode yang digunakan untuk mendeteksi objek bergerak dalam video dengan cara membandingkan setiap frame dengan sebuah (*reference frame*) atau *background model*. *Background modeling* adalah tahap yang penting dari *Background Subtraction*. Objektif utama dari proses *background modeling* adalah untuk mendapatkan model

latar belakang yang berisikan informasi berupa bagian statis dari sebuah *scene*, atau segala sesuatu yang dapat dikatakan sebagai latar belakang. Dibalik kemudahan implementasinya, kelemahan dari metode tradisional ini adalah sistem tidak dapat mengatasi perubahan yang terjadi pada latar belakang selama video berlangsung.

Gaussian Mixture Models (GMM) dapat digunakan untuk memodelkan latar belakang pada teknik *Background Subtraction*. Dalam penelitian [4] sebuah piksel dimodelkan sebagai *mixture of gaussian*. Berdasarkan *evidence* dan varians dari masing-masing distribusi *Gaussian*, dapat ditentukan distribusi mana yang masuk ke dalam kelas warna latar belakang (*background color*) dan juga kelas warna latar depan (*foreground color*). Tahapan *Background Subtraction* menggunakan GMM dapat dijabarkan sebagai berikut.

1. Pertama, setiap masukkan piksel baru dibandingkan dengan elemen-elemen K distribusi *Gaussian*. Jika nilai dari sebuah piksel berada pada 2,5 standar deviasi sebuah distribusi *Gaussian*, maka nilai piksel tersebut dapat dikatakan *match* dengan distribusi terkait.
2. Kedua, jika terdapat distribusi yang *match*, maka *update* nilai dari *weight*, *mean*, dan *variance* distribusi *Gaussian* untuk menyesuaikan dengan nilai piksel baru tersebut. Jika tidak ada distribusi yang *match*, maka buat distribusi baru yang menggunakan nilai piksel baru sebagai rata-ratanya, nilai varians awal yang besar, dan juga bobot yang rendah. Distribusi yang paling kecil kemungkinannya akan digantikan oleh distribusi baru jika $K \text{ Gaussians} > K$
3. Ketiga, tentukan distribusi *Gaussian* mana yang termasuk ke dalam kelas warna latar belakang atau *background model*. Nilai *threshold T* diaplikasikan terhadap variable *weight / variance*.
4. Terakhir, distribusi yang tidak termasuk kedalam kelas warna latar belakang adalah distribusi latar depan.

Formula dari *Gaussian Mixture Models* dapat didefinisikan sebagai berikut

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

Dimana K adalah jumlah distribusi *Gaussian*, $\omega_{i,t}$ adalah estimasi dari *evidence / weight / bobot* (proporsi data yang direpresentasikan oleh *Gaussian* yang bersangkutan), $\mu_{i,t}$ adalah nilai rata-rata dari *Gaussian* ke- i pada waktu t , $\Sigma_{i,t}$ adalah kovarians matriks *Gaussian* ke- i pada waktu t , dan η adalah fungsi probabilitas *Gaussian*

$$\eta(X_t, \mu_t, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \quad (2)$$

Dengan kovarians matriks

$$\sum_{i,t} = \sigma_k^2 I \quad (3)$$

Jika terdapat *match*, maka distribusi parameter (ω, μ, σ) dari distribusi *Gaussian* diperbarui sebagai berikut

$$\omega_{k,t} = (1 - \alpha) \omega_{k,t-1} + \alpha (M_{k,t}) \quad (4)$$

$$\mu_t = (1 - \rho) \mu_t - 1 + \rho X_t \quad (5)$$

$$\sigma_t^2 = (1 - \rho) \sigma_t^2 - 1 + \rho (X_t - \mu_t)^T (X_t - \mu_t) \quad (6)$$

Dimana α adalah learning rate, $M_{k,t}$ bernilai 1 untuk distribusi yang "*matched*", dan 0 untuk distribusi yang lainnya. Lalu

$$\rho = \alpha \eta(X_t | \mu_t, \sigma_t) \quad (7)$$

B. Operasi Morfologi

Operasi morfologi adalah sebuah operasi yang diaplikasikan terhadap bentuk dan struktur sebuah objek di dalam citra. Operasi ini bertujuan untuk menghapus *noise* secara lebih agresif. Digunakan dua jenis operasi morfologi yaitu, erosi dan dilasi.

Erosi membuat objek pada citra menjadi lebih kecil. Erosi dapat digunakan untuk menghilangkan *noise* pada citra dan juga memisahkan objek satu dengan yang lainnya. Operasi dilasi membuat objek pada citra bertambah ukurannya menjadi lebih besar. Dilasi sangat berguna 19 untuk menutup lubang pada objek atau menggabungkan bagian-bagian objek yang terpisah. Operasi erosi yang diikuti oleh dilasi disebut sebagai *opening* [5].

C. Contour Tracing

Contour tracing atau *border following* dalam pengolahan citra digital adalah proses menghasilkan tepi suatu objek pada citra. *Border following* telah dipelajari secara mendalam karena metode ini memiliki jangkauan pengaplikasian yang sangat luas, seperti pengenalan objek, analisis citra, deteksi objek, dan kompresi data citra [6].

Dalam penelitiannya [6], Suzuki menawarkan dua buah algoritma *border following*. Algoritme pertama adalah salah satu algoritme yang pertama kali mendefinisikan hubungan hierarki antar tepi (*border*). Selain itu, algoritme ini juga dapat membedakan tepi luar (*outer border*) dan tepi lubang (*hole border*) di sebuah objek. Secara garis besar, langkah-langkah *contour tracing* adalah sebagai berikut.

1. Pindai piksel dari kiri ke kanan, dan dari atas ke bawah sampai *piksel* $\neq 0$. Lalu tentukan apakah piksel tersebut merupakan *starting point* tepi luar atau tepi lubang.
2. Tentukan *parent border* dari piksel tersebut.
3. Lakukan operasi *border following* dengan piksel terkait sebagai *starting point*-nya.

4. Jika *pointer* sudah kembali ke posisi semula, lanjutkan pemindaian piksel sampai piksel terakhir.

Selain algoritme di atas, Suzuki juga mengusulkan algoritme *border following* yang hanya akan menghasilkan tepi terluarnya saja.

D. Tracking Menggunakan Kalman Filter

1. Definisi

Secara matematis, Kalman Filter adalah sebuah algoritme estimator yang dapat melakukan prediksi dan koreksi terhadap keadaan sebuah proses linear [7]. Model matematis yang akurat sangat diperlukan untuk mendapatkan hasil yang optimal. Kalman filter digunakan untuk melakukan estimasi suatu keadaan x dalam sistem linear. Model dari proses (*process model*) yang mendefinisikan evolusi suatu keadaan dari waktu $k-1$ sampai waktu k adalah

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1} \quad (8)$$

Kemudian model yang menggambarkan hubungan antara keadaan dan observasi pada waktu k adalah

$$z_k = Hx_k + v_k \quad (9)$$

Dimana w_{k-1} adalah vektor *noise* dari proses dan v_{k-1} adalah vektor *noise* dari observasi, dan

$$x_k = \begin{bmatrix} p_k & q_k & l_k & h_k & v_{p,k} & v_{q,k} \end{bmatrix}^T \quad (10)$$

$$z_k = \begin{bmatrix} p_k & q_k & l_k & h_k \end{bmatrix}^T \quad (11)$$

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (13)$$

Variabel p_k dan q_k merepresentasikan nilai horizontal dan vertikal koordinat titik tengah, l_k dan h_k merepresentasikan setengah dari lebar dan tinggi *bounding box*, serta $v_{p,k}$ dan $v_{q,k}$ merepresentasikan kecepatan dari p_k dan q_k .

2. Algoritme Kalman Filter

Setelah melalui proses pendefinisian model proses dan model pengukuran di atas, *Kalman Filter* kemudian dapat digunakan untuk memperkirakan lokasi serta ukuran sebuah objek pada *frame* selanjutnya. *Kalman Filter* terdiri dari dua

tahap, yaitu *predict* dan *update*. Dalam literasi lain, tahapan ini juga biasa disebut sebagai *propagation* dan *correction*. Tahapan *predict* dan *update* dapat diringkas ke dalam persamaan berikut

Predict

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1} \quad (14)$$

$$P_k^- = FP_{k-1}^+F^T + Q \quad (15)$$

Update

$$\tilde{y}_k = z_k - H\hat{x}_k^- \quad (16)$$

$$K_k = P_k^- H^T (R + HP_k^- H^T)^{-1} \quad (17)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y}_k \quad (18)$$

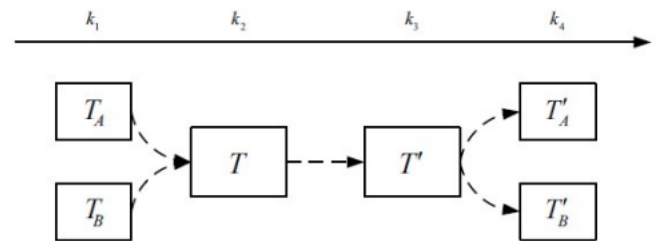
$$P_k^+ = (I - K_k H) P_k^- \quad (19)$$

Dari persamaan di atas, operator hat ($\hat{\cdot}$) menandakan sebuah estimasi dari sebuah variabel. Dengan kata lain, \hat{x} adalah estimasi dari x . Superskrip $-$ dan $+$ masing-masing menyatakan estimasi yang diprediksi (*prior*) dan diperbarui (*posterior*).

Untuk memprediksi estimasi dari sebuah keadaan \hat{x}_k^- dibutuhkan estimasi keadaan \hat{x}_{k-1}^+ dari iterasi *Kalman Filter* sebelumnya. Variabel P adalah matriks kovarians dari keadaan x . Matriks kovarians ini digunakan untuk membantu *Kalman Gain* K_k dalam menentukan bobot terhadap nilai yang diprediksi atau nilai yang diukur. Residual pengukuran \tilde{y}_k adalah selisih antara nilai yang diobservasi z_k , dengan nilai yang diestimasi $H\hat{x}_k^-$. Q dan R adalah matriks kovarians.

3. Decision of Occlusion

Occlusion adalah sebuah peristiwa dimana dua atau lebih objek saling bertumpuk satu sama lain. Hal ini tentu dapat berpengaruh terhadap performa sistem jika tidak ditangani dengan baik. *Occlusion* pada dasarnya mencakup dua masalah, yaitu masalah *merge* dan *split*. Menyatuanya dua atau lebih objek disebut sebagai proses *merge*. Sebaliknya, terpisahnya objek yang sebelumnya menyatu disebut sebagai proses *split* [8].



GAMBAR 1. PROSES MERGE DAN SPLIT

4. Asosiasi Data

Untuk mendapatkan hasil pelacakan objek yang akurat, proses asosiasi data perlu dilakukan dengan menggunakan jarak antara keadaan objek yang diestimasi dengan hasil observasi serta luas dari objek tersebut sebagai *cost function*. Secara garis besar, *cost function* ini digunakan untuk mengklasifikasikan objek hasil dari prediksi *Kalman Filter* dengan objek hasil dari observasi (deteksi). Pertama, jarak antara dua titik pusat didefinisikan sebagai

$$D_k(i, j) = \frac{|\sqrt{(p_{k-1}^j - p_k^i)^2 + (q_{k-1}^j - q_k^i)^2}|}{\max|(p_{k-1}^j - p_k^i)^2 + (q_{k-1}^j - q_k^i)^2|} \quad (20)$$

Dimana p_{k-1}^j dan q_{k-1}^j adalah nilai titik pusat dari koordinat x dan y yang didapatkan dari keadaan \hat{x}_{k-1}^+ dalam iterasi ke j^{th} . Lalu p_k^i dan q_k^i adalah nilai titik pusat dari koordinat x dan y yang didapatkan dari observasi z_k dalam iterasi ke i^{th} . Variabel $i = 1, \dots, m$ dan $j = 1, \dots, n$ dimana m adalah jumlah objek yang terdeteksi dari hasil observasi pada frame k dan n adalah jumlah objek yang dihasilkan oleh estimasi *Kalman Filter* pada frame $k - 1$. Kedua, luas antara dua objek didefinisikan sebagai

$$A_k(i, j) = \frac{|A_k^i - A_{k-1}^j|}{\max|A_k^i - A_{k-1}^j|} \quad (21)$$

Dimana $A_k^i = 4l_k^i h_k^i$ dan $A_{k-1}^j = 4l_{k-1}^j h_{k-1}^j$ yang merepresentasikan *bounding box*. Semakin kecil nilai fungsi di atas, semakin mirip deskripsi bentuk dari dua objek tersebut. Dengan mengombinasikan persamaan 20 dan persamaan 21, didefinisikan *cost function* sebagai berikut.

$$C_k(i, j) = \alpha D_k(i, j) + \beta A_k(i, j) \quad (22)$$

Dimana $\alpha + \beta = 1$, parameter ini dapat diatur secara eksperimental, dalam hal ini digunakan $\alpha = 0.8$ dan $\beta = 0.2$

III. METODE PENELITIAN

Secara garis besar, metode penelitian yang dilakukan terdiri dari tiga tahap yaitu, pengumpulan dan pemrosesan data, kemudian dilanjutkan dengan pelacakan oleh sistem, dan terakhir evaluasi hasil.

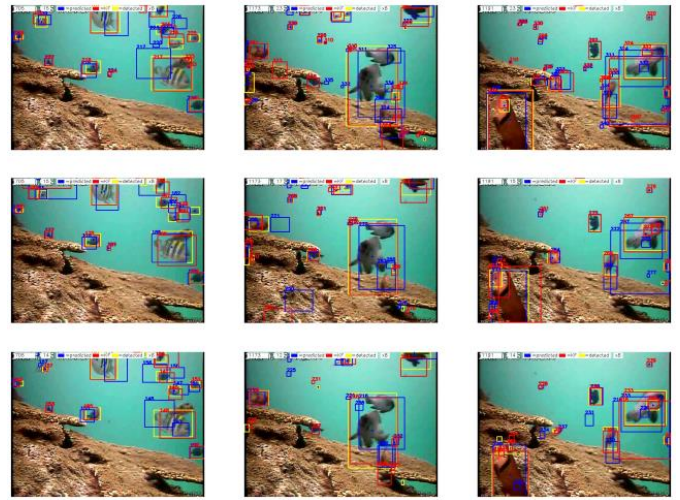
Berdasarkan jumlah ikan dan keadaan latar belakang, terdapat empat skenario pengujian yang akan dilakukan pada penelitian ini. Keempat skenario tersebut adalah

1. Pengujian terhadap video berisikan satu ikan dengan latar belakang sederhana.
2. Pengujian terhadap video berisikan satu ikan dengan latar belakang kompleks.
3. Pengujian terhadap video berisikan dua atau lebih ikan (*multiple object*) dengan latar belakang sederhana.
4. Pengujian terhadap video berisikan dua atau lebih ikan (*multiple object*) dengan latar belakang kompleks

Terakhir, dilakukan proses evaluasi terhadap hasil pelacakan sistem. Dibandingkan rata-rata jumlah objek sebenarnya dengan rata-rata jumlah objek yang berhasil dilacak (*counting*). Kemudian dihitung juga *error* masing-masing koordinat titik tengah objek yang dilacak dengan koordinat titik tengah objek hasil deteksi (*tracking*).

IV. HASIL DAN PEMBAHASAN

Pada tahap ini, masing-masing video diuji performa *tracking* serta *counting*-nya. Diambil data uji setiap 10 *frame* sekali selama video berlangsung. Kemudian dihitung *RMSE* koordinat piksel objek (x, y) dari seluruh data uji untuk masing-masing kategori pengujian. Secara garis besar, hasil yang diharapkan adalah nilai *RMSE* yang kecil dimana nilai 0 menandakan model yang akurat. Selain itu dihitung juga jumlah objek rata-rata seluruh data uji hasil dari prediksi KF.



GAMBAR 2. SAMPEL FRAME HASIL KELUARAN TRACKING

1. Hasil Penghitungan Ikan

Tabel 1 memperlihatkan skor *counting* sistem dengan parameter sistem terbaik. Variabel jumlah rata-rata didapatkan dari hasil menghitung rata-rata jumlah *track* setiap 10 *frame* sekali selama video berlangsung.

TABEL 1. HASIL PERFORMA *COUNTING* SISTEM DENGAN PARAMETER TERBAIK

| Indeks | Rata-rata Jumlah | Rata-rata Jumlah Sebenarnya | Error |
|--------|------------------|-----------------------------|-------|
| 9908 | 6,5 | 1 | 84,62 |
| 9866 | 6,4 | 1 | 84,38 |
| gt_124 | 14,2 | 12,7 | 10,56 |
| gt_116 | 3,4 | 4,2 | 23,53 |

Pada video indeks 9908 (objek tunggal latar belakang sederhana), *counting error* yang dihasilkan berada diangka yang sangat besar yaitu 84,62%. Angka yang tinggi tersebut didapatkan karena beberapa faktor. Pertama, tidak optimalnya *dataset* yang digunakan untuk percobaan objek tunggal latar

belakang sederhana. Latar belakang video indeks 9908 masih banyak mengandung *noise* bergerak yang dapat mengganggu proses deteksi objek seperti iluminasi cahaya, dan partikel-partikel kecil yang terbawa oleh air. Kedua, kumpulan *noise* tersebut pada akhirnya ikut terbaca sebagai objek bergerak oleh proses deteksi objek menggunakan GMM, dikarenakan GMM belum dapat membedakan objek bergerak ikan dengan objek bergerak lainnya. Kedua hal inilah yang paling berdampak pada buruknya akurasi yang dihasilkan.

Video indeks 9866 juga mengalami hal yang serupa seperti video indeks 9908. Pada percobaan video objek tunggal latar belakang kompleks didapatkan *counting error* yang sangat besar yaitu sekitar 84,38%. Hal ini dikarenakan proses objek deteksi menggunakan GMM yang belum bisa membedakan objek bergerak ikan dengan objek bergerak lainnya pada video indeks 9866 yang sudah sangat *noisy*.

Pada indeks video selanjutnya, yaitu video indeks gt_124 yang masuk ke dalam kategori video objek lebih dari satu dan latar belakang sederhana, didapatkan *counting error* sebesar 10,56%. Error yang kecil ini didapat karena video *dataset* yang digunakan sudah cukup optimal. Pada pengujian ini dapat kita lihat bagaimana video dengan *noise* yang sangat minim dapat menghasilkan keluaran yang lebih baik dari pada video dengan *noise* yang cukup banyak (9908, 9866).

Video indeks gt_116 adalah video kategori objek lebih dari satu dan latar belakang kompleks. Jika diperhatikan, *counting error* yang dihasilkan lebih besar dibandingkan video indeks gt_124, yaitu sebesar 23,53%. Seperti pada pengujian tiga kategori video sebelumnya, video yang mengandung banyak *noise* tentu akan menghasilkan keluaran yang lebih buruk dari pada video dengan sedikit *noise*.

2. Hasil Pelacakan Ikan

Tabel 2 memperlihatkan skor *tracking* sistem dengan parameter sistem terbaik. Variabel *RMSE* dihitung dengan cara membandingkan koordinat titik tengah objek yang berhasil dideteksi oleh metode sebelumnya dengan koordinat titik tengah objek yang diprediksi oleh *Kalman Filter*.

TABEL 2. HASIL PERFORMA *TRACKING* SISTEM DENGAN PARAMETER TERBAIK

| Indeks | RMSEX | RMSEY | Total RMSE |
|--------|-------|-------|------------|
| 9908 | 10,8 | 10,6 | 15,1 |
| 9866 | 11,8 | 10,5 | 15,8 |
| gt_124 | 10,4 | 8,2 | 13,6 |
| gt_116 | 7,8 | 6,9 | 10,6 |

Untuk video indeks 9908 dan 9866, rata-rata nilai *RMSE_x* (koordinat titik tengah *x*) antara koordinat piksel sesungguhnya dan koordinat piksel yang diprediksi adalah 11,3. Nilai ini menunjukkan hasil yang cukup baik mengingat nilai piksel maksimal koordinat *x* (lebar citra) adalah 720 piksel. Begitu juga dengan koordinat titik tengah *y*, didapat rata-rata nilai

RMSE_y sebesar 10,55, dimana nilai piksel maksimal koordinat *y* (panjang citra) adalah 480 piksel.

Video indeks gt_124 menghasilkan nilai terbaik *RMSE_x* = 10,4 dan *RMSE_y* = 8,2, dimana nilai piksel maksimal koordinat *x* (lebar citra) adalah 640 piksel dan nilai piksel maksimal koordinat titik *y* (panjang citra) adalah 480 piksel. Pada video indeks gt_116, nilai terbaik *RMSE_x* dan *RMSE_y* yang dihasilkan adalah 7,8 dan 6,9, dimana nilai piksel maksimal koordinat *x* (lebar citra) adalah 320 piksel dan nilai piksel maksimal koordinat *y* (panjang citra) adalah 240 piksel.

Sistem berhasil melakukan pelacakan dengan nilai *RMSE* yang cukup baik untuk masing-masing video. Nilai *RMSE* merepresentasikan seberapa jauh rata-rata *offside* kedua koordinat titik tengah tersebut. Semakin nilainya mendekati 0, maka dapat dikatakan bahwa hasil prediksi oleh *Kalman Filter* semakin akurat.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari perancangan, implementasi dan uji coba sistem pelacakan ikan dengan metode *GMM* dan *Kalman Filter*, diperoleh kesimpulan sebagai berikut:

1. Sistem mampu melakukan pelacakan dengan lebih baik seiring dengan bertambahnya jumlah frame yang diproses.
2. Sistem mampu melakukan pelacakan dengan cukup baik menggunakan parameter structuring element ukuran 7×13 dan downsampling sebesar $\times 8$.
3. Pada dataset video indeks 9908 dan 9866 (objek tunggal), sistem pada akhirnya dapat menghasilkan rata-rata jumlah objek yang jauh dari rata-rata sebenarnya. Hal ini dikarenakan dataset yang digunakan kurang optimal pada video indeks 9908, serta gagalnya proses deteksi dan noise removal pada video indeks 9866.
4. Pada dataset video indeks gt_124 dan gt_116 (objek lebih dari satu), sistem pada akhirnya dapat menghasilkan rata-rata jumlah objek yang mendekati rata-rata sebenarnya.

B. Saran

Adapun beberapa saran yang dapat penulis sampaikan untuk penelitian selanjutnya adalah:

1. Penelitian ini belum dapat membedakan objek ikan dengan objek bergerak lainnya. Maka dari itu, perlu adanya pengembangan lanjutan yang menyempurnakan proses deteksi objek dengan metode yang lebih kompleks seperti Neural-Network, sehingga sistem dapat membedakan objek ikan dengan objek lainnya dengan baik.

2. Proses asosiasi data dapat ditingkatkan cost function-nya. Selain ukuran dan posisi, fungsi tersebut dapat ditambahkan variabel lain seperti warna.
3. Penelitian ini dapat digunakan untuk menguji objek selain ikan.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, "Konsumsi Kalori dan Protein Penduduk Indonesia dan Provinsi," 2016. <https://www.bps.go.id/publication/2020/06/29/78ae644b0d37f4e1329d522f/konsumsi-kalori-dan-protein-penduduk-indonesia-dan-provinsi-september-2019.html> (diakses 21 September 2021).
- [2] C. F. Al-Amri, "Rancang Bangun Fish Counter untuk Menghitung Bibit Ikan Lele," Universitas Islam Indonesia, Yogyakarta, 2020.
- [3] A. P. Affandy, "Model Alat Penghitung Benih Ikan Air Tawar Ikan Mas (Fish Counter)," Universitas Padjadjaran, Jatinagor, 2016.
- [4] C. Stauffer dan W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.
- [5] A. M. Khan, "Morphological Operations for Image Processing: Understanding and its Applications," 2013. [Daring]. Tersedia pada: <https://www.researchgate.net/publication/272484795>
- [6] S. Suzuki, "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics, and Image Processing*, vol. 30, hlm. 32–46, 1985.
- [7] R. Chavan dan S. R. Gengaje, "Multiple Object Detection using GMM Technique and Tracking using Kalman Filter," 2017.
- [8] X. Li, K. Wang, dan W. Wang, *IEEE International Conference on Information and Automation (ICIA), 2010 Harbin, China, 20-23 June 2010*. 2010.
- [9] J.-M. Jeong, T.-S. Yoon, dan J.-B. Park, "Kalman Filter Based Multiple Objects Detection-Tracking Algorithm Robust to Occlusion," *SICE Annual Conference*, 2014.