

## **GAUSSIAN MIXTURE MODEL UNTUK PENGHITUNGAN TINGKAT KEBERSIHAN SUNGAI BERBASIS PENGOLAHAN CITRA**

**Bayu Charisma Putra<sup>1</sup>, Yunita Nur Afifah<sup>2</sup>**

Teknik Informatika, Fakultas Teknik  
Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia  
e-mail : <sup>1</sup>bayu\_charisma@dosen.umaha.ac.id, <sup>2</sup>yunitanurafifah02@gmail.com

Diterima: 18 April 2018. Disetujui : 15 Juni 2018. Dipublikasikan : 29 Juni 2018



©2018 –TESJ Fakultas Teknik Universitas Maarif Hasyim Latif. Ini adalah artikel dengan akses terbuka di bawah lisensi CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

### **ABSTRAK**

Banjir adalah salah satu permasalahan yang paling sering terjadi pada hampir setiap musim hujan. Salah satu penyebab banjir adalah terhambatnya aliran sungai akibat sampah yang menumpuk, sehingga air sungai yang seharusnya mengalir ke hilir akan meluber ke pemukiman. Hal tersebut tentunya harus mendapat perhatian lebih dari dinas terkait, salah satunya adalah dengan menggali informasi sungai mana saja yang menjadi prioritas berdasarkan kondisi kebersihannya. Seiring berkembangnya era digital, informasi tersebut dapat diperoleh dengan menggunakan sebuah alat. Dalam jurnal ini akan dibahas bagaimana memperoleh informasi tingkat kebersihan sungai dengan metode Gaussian Mixture Model (GMM) berbasis pengolahan citra. Untuk mencapai tujuan tersebut diperlukan beberapa tahap. Tahap pertama adalah mendeteksi sampah yang ada pada sungai dengan proses *background subtraction* menggunakan metode GMM. *Output* dari proses *background subtraction* adalah kumpulan gambar yang hanya memiliki dua intensitas cahaya yang bertujuan untuk membedakan *background* dan *foreground*. Masing-masing gambar tersebut selanjutnya di-filter menggunakan metode *Median Filter* untuk mengurangi gangguan pada gambar (*noise*) dan lebih mempertajam hasil deteksi. Terakhir adalah penghitungan tingkat kebersihan sungai berdasarkan prosentase sampah pada sungai di dalam ROI (*Region of Interest*). Selanjutnya perhitungan tersebut disimulasikan dengan aplikasi berbasis desktop dengan bahasa pemrograman JAVA.

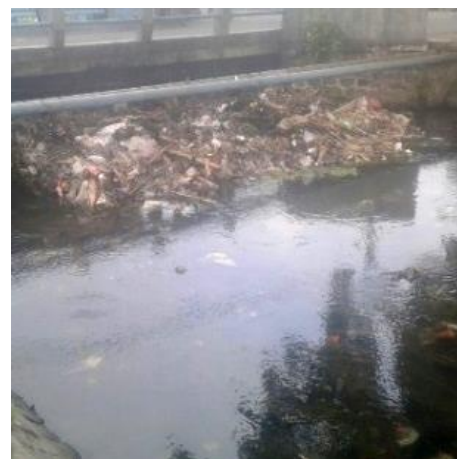
**Kata kunci:** banjir, deteksi sampah, gaussian mixture model, median filter, pengolahan citra, java

### **PENDAHULUAN**

Hampir setiap musim penghujan di Indonesia mempunyai masalah yaitu banjir. Salah satu faktor banjir adalah terhambatnya aliran air sungai akibat dari banyaknya sampah yang menumpuk seperti yang ditampilkan pada Gambar

1. Dengan mengetahui informasi dasar seperti tingkat kebersihan sungai berdasarkan banyaknya sampah di sungai, dinas terkait dapat dengan sigap mengatasinya sehingga faktor tersebut dapat diminimalisir.

Pesatnya perkembangan teknologi di semua aspek seperti saat ini sangat mendukung



Gambar 1. Aliran air sungai yang terhambat



Gambar 2. Proses Background Subtraction

penyelesaian masalah secara komputasi. Dengan menggunakan data *input* berupa video air sungai yang mengalir, penghitungan tingkat kebersihan sungai dengan menggunakan pengolahan citra dapat menjadi pilihan yang akan menghemat tenaga, waktu, dan biaya. Video akan dijadikan sebagai sumber untuk memperoleh obyek yang bergerak dengan algoritma *Background Subtraction*.

*Background Subtraction* merupakan salah satu algoritma yang dapat memproses serangkaian citra pada video digital untuk mendapatkan obyek *foreground* dari model *background* yang telah ditentukan sebelumnya. Obyek *foreground* adalah segala sesuatu yang dideteksi dari hasil pengurangan citra pada waktu  $t$  dengan model *background*. Hasil tersebut dapat digunakan untuk mendeteksi gerak objek (Bharti, 2013).

Rosenberg C melakukan perbandingan dari 7 metode *Background Subtraction*; *Basic*, *Min Max Inter-Frame Difference*, *One Gaussian(1-G)*, *Eigen Backgrounds (Eigen)*, *Kernel Density Estimation (KDE)*, *Codebook (CBRGB)*, dan *Gaussian Mixture Model (GMM)*. Dari penelitian tersebut diperoleh kesimpulan bahwa metode GMM memiliki tingkat akurasi yang tinggi dengan berbagai kondisi *input* video; static, multimodal, maupun *noise background* serta memiliki waktu komputasi dan kebutuhan memori yang lebih efisien dibandingkan metode yang lain (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2010).

Berdasarkan paparan diatas, dalam jurnal ini membahas tentang bagaimana cara menghitung tingkat kebersihan dengan menggunakan GMM berbasis pengolahan citra digital. Tahap pertama yang dilakukan adalah deteksi sampah yang mengalir pada sungai. Untuk lebih meningkatkan tingkat akurasi deteksi, akan dilakukan *filtering* dengan menggunakan metode *Median Filter*. Terakhir adalah penghitungan tingkat kebersihan sungai berdasarkan prosentase sampah dengan sungai pada ROI (*Region of Interest*).

## METODE PENELITIAN

### *Background Subtraction*

*Background Subtraction* adalah proses untuk mendapatkan *foreground object* (obyek yang bergerak) dari serangkaian citra. *Foreground* diperoleh dengan melakukan pengurangan *frame* pada waktu  $t$  dengan *background* model dari video, kemudian hasil pengurangan tersebut dibandingkan dengan suatu nilai ambang batas (*threshold*) tertentu. Contoh proses *background subtraction* diberikan pada Gambar 2. Persamaan (1) dan persamaan (2), adalah representasi dari penjelasan tersebut dalam bentuk matematis.

$$|I(x, y, t) - B(x, y, t)| = d(x, y, t) \quad (1)$$

$$f(x, y, t) = \begin{cases} 1, & \text{Jika } d(x, y, t) \geq \tau, \\ 0, & \text{lainnya} \end{cases} \quad (2)$$

Dengan  $I(x, y, t)$  merupakan intensitas dari citra  $(x, y)$  pada *frame*  $t$ ,  $B(x, y, t)$  merupakan intensitas dari *background* citra  $(x, y)$  pada *frame*  $t$ ,  $d(x, y, t)$  merupakan hasil pengurangan intensitas dari *background* citra terhadap intensitas dari citra pada *frame*  $t$ ,  $\tau$  merupakan ambang batas atau *threshold*, dan  $f(x, y, t)$  adalah intensitas dari *foreground* citra.

### *Gaussian Mixture Model*

*Gaussian Mixture Model* (GMM) merupakan salah satu metode yang paling baik untuk mendapatkan *foreground object* (*Background Subtraction*). GMM adalah tipe model kepadatan yang terdiri dari banyak komponen berupa fungsi gaussian. Komponen fungsi gaussian tersebut terdiri dari *weight* yang berbeda untuk menghasilkan beberapa model kepadatan. Model dari *Gaussian Mixture* terbentuk dari warna-warna pada piksel berdasarkan waktu. Model tersebut terbagi menjadi 2 bagian, model yang mencerminkan *foreground* yang dalam penelitian ini adalah sampah dan model yang mencerminkan *background*.

Menurut Amaluddin dkk (Amaluddin, Muslim, & Naba, 2015) Jumlah model GMM yang digunakan mempengaruhi jumlah model *background*. Semakin sedikit jumlah model GMM yang digunakan maka akan semakin sedikit pula model *background* yang dimiliki suatu piksel. GMM memproses tiap piksel pada citra, baik citra

tersebut berupa skalar (citra *grayscale*) maupun vektor (citra berwarna).

Menurut Amaluddin dkk (Amaluddin et al., 2015) terdapat beberapa tahapan dalam pemilihan distribusi yang mencerminkan *background*. Tahapan-tahapan tersebut adalah :

1. Pencocokan *input* terhadap distribusi.

Pada proses ini *input* yang berupa intensitas dari warna dibandingkan dengan semua distribusi, kemudian dipilih distribusi yang paling sesuai. Suatu intensitas piksel *input* dikatakan sesuai dengan suatu distribusi jika nilai intensitas piksel tersebut masuk dalam *range* 2.5 simpangan baku dari distribusi tersebut. Untuk pencocokan *input* digunakan persamaan (3).

$$\mu_k - (2.5 \times \sigma_k) < X_t < \mu_k + (2.5 \times \sigma_k) \quad (3)$$

2. Memperbarui parameter

Pada proses ini dilakukan pembaharuan beberapa parameter GMM, dimana parameter tersebut akan digunakan untuk mengolah intensitas piksel atau *input* selanjutnya. Nilai yang di-*update* terdiri dari *weight*, *mean*, dan *standard deviation*. Nilai *weight* di-*update* menggunakan persamaan (4). Nilai *mean* di-*update* menggunakan persamaan (6) dengan  $\rho$  diberikan pada persamaan (5). Nilai *standard deviation* di-*update* menggunakan persamaan (7). Persamaan-persamaan tersebut akan diberlakukan pada masing-masing piksel (*i, j*).

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

$$\rho = \frac{\alpha}{\omega_{k,t}} \quad (5)$$

$$\mu_{k,t} = (1 - \rho)\mu_{k,t-1} + \rho X_t \quad (6)$$

$$\sigma_{k,t}^2 = (1 - \rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})(X_t - \mu_{k,t}) \quad (7)$$

Dimana  $\omega_{k,t}$  adalah bobot dari gaussian ke-*k* pada *frame t*,  $\alpha$  adalah *learning rate*,  $M_{k,t}$  adalah sebuah parameter yang mempunyai nilai 0 untuk model yang tidak cocok dan 1 untuk model yang cocok,  $\mu_{k,t}$  adalah *mean* dari gaussian ke-*k* pada *frame t*, dan  $\sigma_{k,t}$  adalah simpangan baku dari gaussian ke-*k* pada *frame t*. Setelah parameter bobot di-*update*, terdapat tahap normalisasi dari bobot, yang bertujuan agar total dari bobot semua distribusi tepat 1. Sementara *mean* dan simpangan baku di-*update* hanya jika ada nilai piksel yang cocok dengan distribusi tersebut.

3. Pemilihan Distribusi *Background*

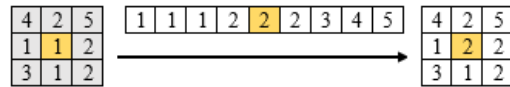
Tujuan dari proses ini adalah memilih model *Gaussian* yang paling menyerupai *background*. Proses pertama model *Gaussian* diurutkan berdasarkan  $\frac{\omega}{\sigma^2}$ . Distribusi yang paling menyerupai *background* berada di urutan paling atas dan yang paling tidak menyerupai *background* ada di bawah yang nantinya

digantikan oleh distribusi yang lain. Untuk memilih *B* distribusi pertama yang dijadikan distribusi *background* digunakan persamaan (8), dengan *T* adalah nilai ambang batas yang telah ditentukan sebelumnya.

$$B = \operatorname{argmin}_b \left( \sum_{k=1}^b \omega_k > T \right) \quad (8)$$

**Median Filter**

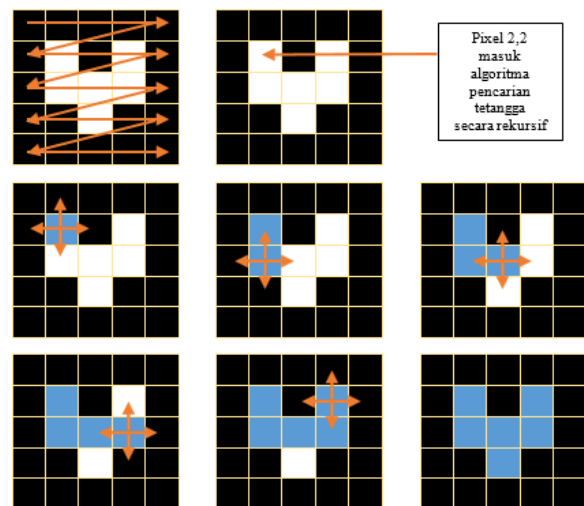
Prinsip kerja dari metode median *filter* adalah dengan cara mengganti nilai intensitas dari tiap piksel dengan nilai median dari piksel-piksel tetangganya. Pola jangkauan piksel-piksel tetangga disebut dengan *window* yang mana telah ditentukan sebelumnya. Nilai median dihitung dengan melakukan *sorting* pada bagian *window* dan piksel itu sendiri. Hasil *output* dari median *filter* adalah *frame* dengan intensitas yang diperoleh dari nilai-nilai median tersebut. Proses tersebut dapat digambarkan pada Gambar 3.



Gambar 3. Proses median filter

**Object Detection**

Pada penelitian ini, algoritma deteksi obyek yang digunakan adalah *pixel connected* yaitu dengan memeriksa semua piksel tetangganya di dalam area ROI yang sebelumnya telah diinputkan. Terdapat dua proses pada deteksi obyek yaitu mencari salah satu piksel dari suatu obyek yang bergerak dan mencari seluruh piksel dari obyek tersebut secara rekursif. Gambar 4 menjelaskan tentang proses tersebut.

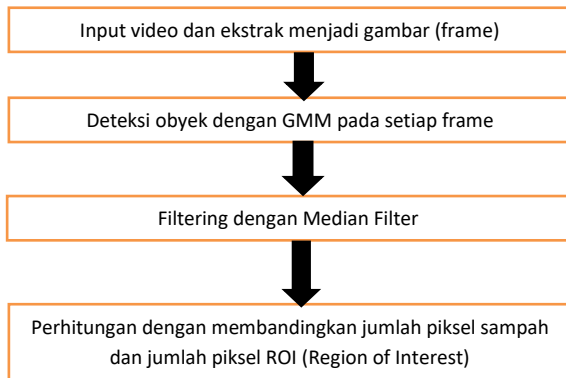


Gambar 4. Proses deteksi obyek

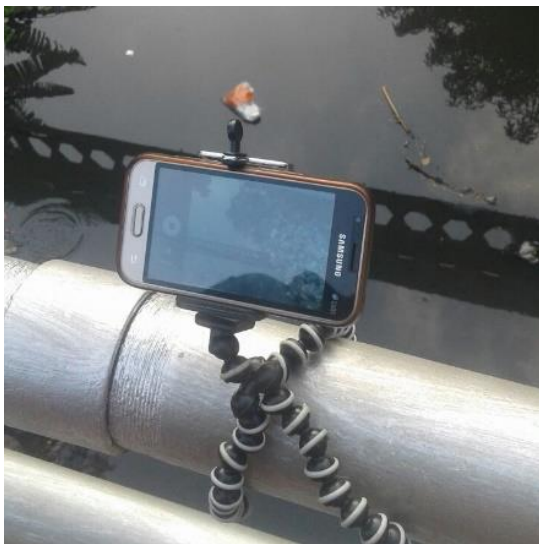
**HASIL DAN PEMBAHASAN**

Berdasarkan metode penelitian yang dijelaskan sebelumnya, untuk mendapatkan *output*

yaitu tingkat kebersihan sungai, akan dilakukan simulasi menggunakan aplikasi yang telah dibuat dengan tahapan seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Diagram proses perhitungan tingkat kebersihan sungai.



Gambar 6. Cara pengambilan data *input*, yaitu melalui kamera *smartphone*.

Data berupa video diambil dari atas jembatan. Alat untuk mengambil video terdiri dari sebuah *smartphone* dan sebuah *octopus* sebagai kaki. *Smartphone* dikaitkan pada *octopus* dan *octopus* dikaitkan pada pegangan jembatan. Pengambilan data dapat dilihat pada Gambar 6.

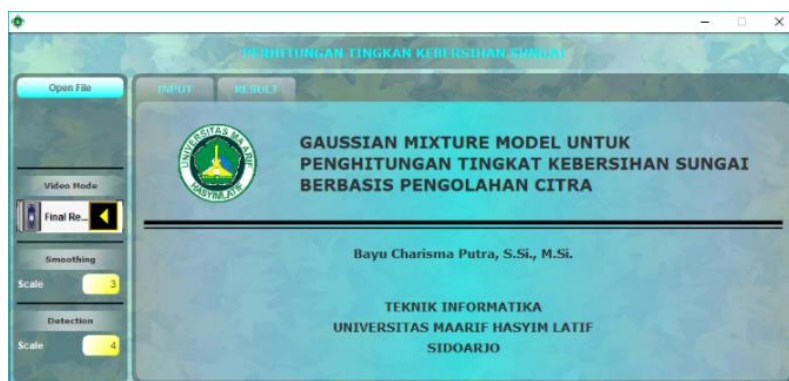
Dalam pengambilan data, kamera yang digunakan harus dalam posisi diam atau tidak bergerak. Hal tersebut dikarenakan proses *background subtraction* membutuhkan posisi piksel *background* yang sama pada setiap *frame* dari video. Kamera juga harus mencakup keseluruhan lebar sungai untuk memperoleh tingkat akurasi yang layak.

Simulasi dilakukan dengan bantuan sebuah aplikasi *desktop* yang dibuat dengan bahasa pemrograman JAVA. Aplikasi juga dibantu oleh sebuah *library* bernama OpenCV yang berfungsi untuk proses pengolahan citra pada JAVA. Tampilan muka pada aplikasi dapat dilihat pada Gambar 7.

Tahap pertama, tentukan ROI terlebih dahulu. Hal ini bertujuan untuk memilih daerah sampel (hanya sungai, bukan daratan) dan mempersempit piksel yang akan diproses sehingga akan mempersingkat waktu komputasi. Input ROI disediakan di tab input pada aplikasi. Pada tab input disediakan 6 parameter input untuk menentukan bentuk ROI. Tab input ditunjukkan pada Gambar 8.

Setelah ROI ditentukan, selanjutnya data berupa video diekstrak menjadi potongan-potongan *frame*. Kemudian setiap *frame* dideteksi keberadaan sampahnya dengan menggunakan metode *background subtraction* yang menjadikan intensitas warna pada setiap piksel dari *frame* sebagai sampel distribusinya. Model *background* dari video akan terus di-*update* dengan menggunakan metode Gaussian Mixture Model. Sehingga terdapat kemungkinan *background* pada waktu  $t$  tidak sama dengan *background* pada waktu  $t + 1$ . *Foreground* (obyek berupa sampah) kemudian akan diperoleh dari proses *background subtraction* dengan membandingkan setiap *frame* pada waktu  $t$  dengan *background* pada waktu  $t$ . *Output* dari proses *background subtraction* adalah gambar atau *frame* dengan 2 jenis warna seperti yang ditunjukkan pada Gambar 2, yaitu hitam untuk *background* dan putih untuk *foreground*.

Setelah obyek berupa sampah diperoleh, maka *filtering* perlu dilakukan. Hal tersebut untuk membantu menghilangkan *noise* pada *frame*. *Noise*



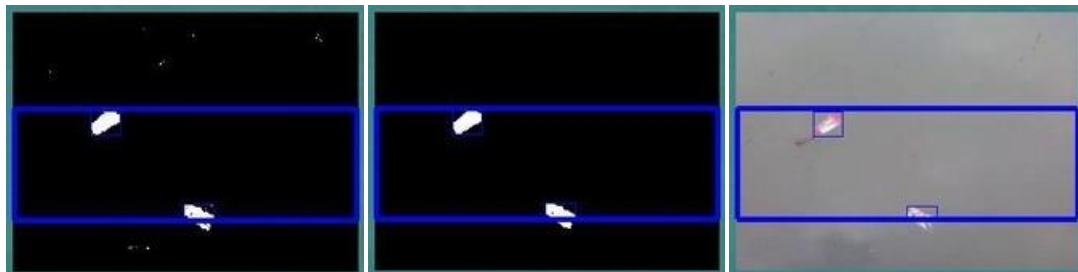
Gambar 7. Tampilan muka



Gambar 8. Tampilan Input ROI



Gambar 9. Tampilan proses



Gambar 10. Output dari tiap proses : Background Subtraction (kiri); Median Filter (tengah); Object Detection (kanan)

yang dimaksud adalah gangguan pada gambar hasil dari proses *background subtraction*, Seperti air hujan atau serangga terbang. Hasil dari proses *filtering* berupa *frame* yang telah dihilangkan *noise*-nya, Sehingga secara visual akan tampak lebih halus.

Setiap *frame* hasil *output* dari proses *filtering* akan dihitung tingkat kebersihannya berdasarkan sampah yang terdeteksi pada area ROI. Satuan dari penghitungan tersebut adalah piksel. Sehingga untuk menghitung tingkat kebersihan perlu diketahui terlebih dahulu jumlah piksel sampah di dalam ROI dan jumlah piksel ROI. Jumlah piksel sampah dan jumlah piksel ROI diperoleh dengan algoritma deteksi obyek seperti yang dijelaskan pada pembahasan sebelumnya. Tampilan *interface* pada aplikasi dari tahapan perhitungan tingkat kebersihan sungai ditampilkan pada Gambar 9. Sedangkan tampilan *interface* pada

masing-masing tahapan ditampilkan pada Gambar 10.

Perhitungan tingkat kebersihan tiap *frame* dapat diperoleh dengan menggunakan persamaan (9). Sedangkan tingkat kebersihan keseluruhan diperoleh dari rata-rata tingkat kebersihan seluruh *frame* dari video input.

$$\text{Tingkat kebersihan} = \frac{\text{Jumlah piksel sampah}}{\text{Jumlah piksel ROI}} \times 100\% \quad (9)$$

Data untuk simulasi diambil di tiga sungai kecil yang berbeda, yaitu sungai megare, sungai bungurasih, dan sungai bambe. Dengan menggunakan aplikasi yang telah dibuat data tersebut diinputkan satu persatu kemudian dilihat hasilnya. Hasil dari simulasi ialah

- Sungai Bambe 99.27%
- Sungai Bungurasih 99.81%
- Sungai Megare 99.8%

Tentunya simulasi tersebut tidak 100% akurat dikarenakan banyak yang masih belum bisa

ter-cover aplikasi, diantaranya adalah sampah yang bergerak di kedalaman, sampah yang warnanya menyerupai sungai dan sampah yang berhenti dan tidak mengalir.

### PENUTUP

Pada hasil dari simulasi yang diambil di tiga sungai, yaitu sungai megare, sungai bungurasih, dan sungai bambe, menunjukkan bahwa sungai di kawasan Universitas Maarif Hasyim Latif masih terdapat sampah yang mengalir di sungai.

Jurnal ini masih bisa dikembangkan lagi karena masih banyak yang belum bisa tercover pada simulasi dari aplikasi seperti yaitu mendeteksi sampah yang berhenti, sampah yang bergerak di dalam air dan perubahan warna yang disebabkan air yang tercemar.

### DAFTAR PUSTAKA

- Amaluddin, F., Muslim, M. A., & Naba, A. (2015). Klasifikasi Kendaraan Menggunakan Gaussian Mixture Model (GMM) dan Fuzzy Cluster K Means (FCM). *Jurnal EECCIS*, 9(1), 19-24.
- Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., & Rosenberger, C. (2010). Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3), 33003.
- Bharti, T. T. (2013). Background subtraction technique review. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(3), 166-168.
- Hatta, M., & Susrama, I. G. (2017). Counting Sperma Aktif Menggunakan Metode Otsu Threshold dan Local Adaptive Threshold. *Teknika : Engineering and Sains Journal*, 1(1), 47-54.
- Hatta, M., Susrama, I. G., Purnama, I. K. E., & Hariadi, M. (2016). Cacah Spermatozoa Menggunakan Background Segmentation dan Boundary Detection. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, 11(1), 67-74.