

BAB III

METODOLOGI PENELITIAN

3.1 Analisis Masalah

Analisis masalah dilakukan untuk mengidentifikasi permasalahan yang menjadi dasar penelitian. Berdasarkan studi literatur, identifikasi partikel mikroplastik masih banyak dilakukan secara manual menggunakan mikroskop atau metode laboratorium seperti FTIR dan Raman Spectroscopy. Meskipun akurat, metode tersebut memerlukan biaya yang tinggi, waktu yang lama, serta berpotensi menimbulkan human error.

Sebagai solusi, penelitian ini menerapkan algoritma Contour Detection berbasis *Computer Vision* dan *Digital Image Processing*. Algoritma ini dipilih karena mampu mendeteksi batas objek secara otomatis sehingga proses identifikasi partikel mikroplastik menjadi lebih cepat, efisien, dan mengurangi kesalahan pengamatan.

Sebelum proses deteksi dilakukan, citra terlebih dahulu melalui tahap *preprocessing* yang meliputi *resize*, konversi ke *grayscale*, *Gaussian Blur*, *thresholding*, serta operasi morfologi (*opening* dan *closing*). Tahapan ini bertujuan meningkatkan kualitas citra sehingga algoritma Contour Detection dapat mendeteksi partikel mikroplastik dengan lebih optimal.

3.2 Metode Pengumpulan Data

Metode pengumpulan data pada penelitian ini menggunakan citra partikel mikroplastik dari Zenodo, Kaggle, GitHub, DeepParticle Dataset — Figshare, dan Laboratorium FMIPA Biologi Universitas Bengkulu, dengan citra laboratorium digunakan sebagai data uji sistem. Tahapan pengumpulan data meliputi studi

literatur, pengumpulan dataset, seleksi dataset, dokumentasi data, verifikasi data, dan *ground truth*.

3.2.1 Studi Literatur

Tahap studi literatur dilakukan dengan mengkaji penelitian terdahulu yang berkaitan dengan identifikasi partikel mikroplastik, pengolahan citra digital (*image processing*), *computer vision*, OpenCV, dan algoritma *Contour Detection*. Referensi yang dipelajari meliputi penelitian Meilany Syabrina Daulay mengenai mikroplastik pada air minum, Richard Szeliski tentang *digital image processing*, Gary Bradski dan Adrian Kaehler mengenai OpenCV, serta Adrian Rosebrock terkait penerapan *Contour Detection*. Studi literatur ini digunakan sebagai landasan dalam menentukan metode yang diterapkan pada penelitian.

3.2.2 Pengumpulan Dataset Citra Mikroplastik

Dataset diperoleh dari beberapa sumber, yaitu Zenodo, Kaggle, GitHub, DeepParticle Dataset — Figshare dan Laboratorium FMIPA Biologi Universitas Bengkulu. Data yang diperoleh berupa citra digital partikel mikroplastik dengan format dan karakteristik yang bervariasi. Dataset dari Zenodo, Kaggle, dan GitHub digunakan sebagai data pengembangan sistem, sedangkan citra hasil pengamatan mikroskop dari Laboratorium FMIPA Biologi Universitas Bengkulu digunakan sebagai data uji (*testing*) untuk mengevaluasi kinerja sistem identifikasi partikel mikroplastik. Rincian dataset yang digunakan dapat dilihat pada Tabel 3.1

Tabel 3.1 Pengumpulan Dataset Citra Mikroplastik

Sumber Dataset	Jumlah Citra	Format Data	Ukuran File	Resolusi	Keterangan
Zenodo	413	PNG, JPG	±100 KB–5 MB	±512 × 512 – 2048 × 2048 piksel	Dataset citra digital partikel mikroplastik yang digunakan sebagai data pengembangan sistem.
Kaggle	99	PNG, JPG	±100 KB–3 MB	±512 × 512 – 1920 × 1080 piksel	Dataset citra digital partikel mikroplastik sebagai data pendukung pengembangan sistem.
GitHub	459	PNG, JPG	±100 KB–2 MB	±512 × 512 – 1920 × 1080 piksel	Dataset citra digital partikel mikroplastik untuk pengembangan, latih dan validasi sistem.
DeepParticle Dataset — Figshare	400	JNG, JPG	±100 KB–2 MB	±512 × 512 – 1920 × 1080 piksel	Data latih, Citra partikel mikroplastik (pellet, foam)
Laboratorium FMIPA Biologi Universitas Bengkulu	52	JPG, PNG	±500 KB–10 MB	±1280 × 720 – 4032 × 3024 piksel	Citra hasil pengamatan mikroskop yang digunakan sebagai data uji (testing) untuk mengevaluasi kemampuan sistem dalam mendeteksi

Sumber Dataset	Jumlah Citra	Format Data	Ukuran File	Resolusi	Keterangan
					partikel mikroplastik.

3.2.3 Seleksi Dataset

Dataset yang telah diperoleh kemudian diseleksi berdasarkan beberapa kriteria agar sesuai dengan kebutuhan penelitian. Proses seleksi dilakukan dengan memperhatikan kualitas citra, ketajaman gambar, format file, serta kejelasan objek mikroplastik. Citra yang tidak memenuhi kriteria penelitian tidak digunakan dalam proses analisis.

3.2.4 Ground Truth

Ground truth merupakan data acuan yang digunakan untuk mengevaluasi hasil deteksi partikel mikroplastik menggunakan algoritma Contour Detection. Pada penelitian ini, ground truth diperoleh melalui penghitungan jumlah partikel secara manual pada setiap citra berdasarkan pengamatan visual. Hasil penghitungan tersebut kemudian dibandingkan dengan hasil deteksi sistem untuk mengetahui tingkat kesesuaian deteksi partikel.

Table 3.2 Perbandingan Ground Truth dengan Hasil Contour Detection

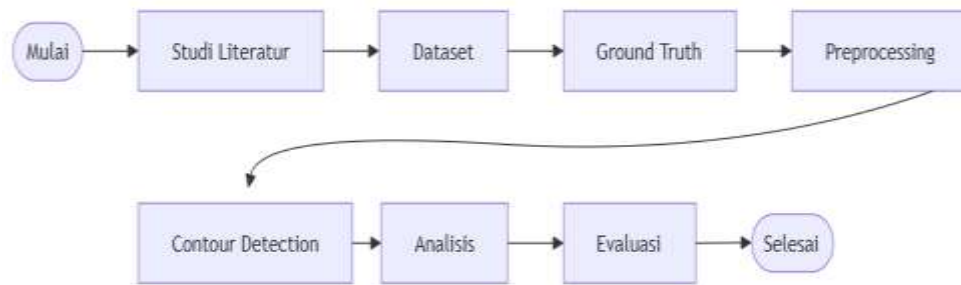
No	Nama Citra	Ground Truth	Hasil Contour	Selisih
----	------------	--------------	---------------	---------

		(Manual)	Detection	
1	Parikel Mikroplastik			3
2	Partikel Mikroplastik			8

Tabel di atas merupakan contoh format yang digunakan untuk membandingkan jumlah partikel hasil pengamatan manual (ground truth) dengan hasil deteksi menggunakan algoritma Contour Detection. Selisih antara kedua nilai tersebut digunakan sebagai dasar evaluasi kinerja sistem.

3.3 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan alur kerja sistem identifikasi partikel mikroplastik menggunakan algoritma *Contour Detection* berbasis *Digital Image Processing*. Sistem dirancang agar mampu mengolah dataset citra mikroplastik secara otomatis mulai dari proses masukan (*input*) hingga menghasilkan keluaran (*output*) berupa informasi jumlah partikel mikroplastik beserta visualisasi hasil deteksi.



Gambar 3.1 Diagram gambar alur sistem

3.3.1 Objek Penelitian

Objek penelitian pada penelitian ini adalah citra digital partikel mikroplastik yang digunakan sebagai data masukan (*input*) dalam proses identifikasi menggunakan algoritma Contour Detection. Citra yang digunakan merupakan data sekunder yang diperoleh dari dataset penelitian terdahulu dan hasil pengamatan mikroskop terhadap partikel mikroplastik.

Partikel mikroplastik dipilih sebagai objek penelitian karena memiliki bentuk dan ukuran yang beragam sehingga memerlukan metode yang mampu mendeteksi batas objek secara akurat. Variasi bentuk partikel serta keberadaan noise pada citra menjadi tantangan dalam proses identifikasi, sehingga diperlukan teknik pengolahan citra digital untuk meningkatkan kualitas citra sebelum proses deteksi dilakukan.

Dalam penelitian ini, objek penelitian memiliki karakteristik Sebagai berikut:

- a. Objek berupa citra digital partikel mikroplastik.
- b. Citra berasal dari dataset hasil penelitian terdahulu.
- c. Citra berisi satu atau lebih partikel mikroplastik.

- d. Partikel mikroplastik memiliki bentuk dan ukuran yang bervariasi.

Seluruh citra kemudian diproses melalui tahapan *preprocessing* yang meliputi *resize*, konversi ke *grayscale*, *Gaussian Blur*, *thresholding*, serta operasi morfologi (*opening* dan *closing*) sebelum dilakukan proses identifikasi menggunakan algoritma Contour Detection.



Gambar 3.2 Objek Yang Diteliti

3.3.2 Preprocessing Citra

Preprocessing merupakan tahapan awal dalam pengolahan citra yang bertujuan untuk meningkatkan kualitas citra sebelum dilakukan proses identifikasi menggunakan algoritma Contour Detection. Tahapan ini dilakukan agar objek partikel mikroplastik dapat dikenali dengan lebih baik serta mengurangi gangguan (*noise*) yang dapat memengaruhi hasil deteksi. Pada penelitian ini, proses preprocessing terdiri atas beberapa tahapan, yaitu *resize*, konversi ke *grayscale*, *Gaussian Blur*, *thresholding*, serta operasi morfologi (*opening* dan *closing*).

a. *Resize* Citra

Tahap *resize* dilakukan untuk menyeragamkan ukuran seluruh citra yang digunakan dalam penelitian. Penyamaan ukuran citra bertujuan untuk mempermudah proses pengolahan serta meningkatkan efisiensi komputasi tanpa men-

gubah karakteristik utama objek partikel mikroplastik.

$$I'(x',y') = I(x,y)$$

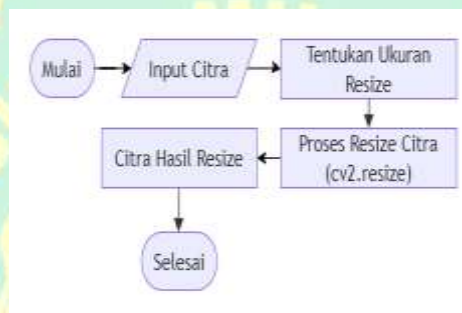
Keterangan:

$I(x,y)$ = citra asli.

$I'(x',y')$ = citra hasil *resize*.

x, y = koordinat piksel citra asli.

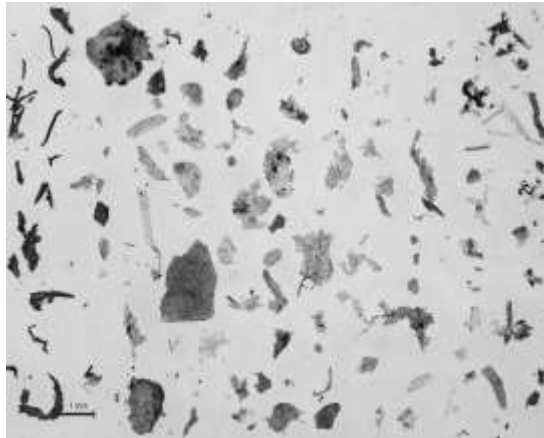
x', y' = koordinat piksel citra hasil *resize*.



Gambar3.3. Diagram gambar alur Resize Citra.

b. Konversi ke Grayscale

Setelah proses *resize*, citra berwarna diubah menjadi citra *grayscale*. Proses ini dilakukan untuk menyederhanakan informasi citra dari tiga kanal warna (RGB) menjadi satu kanal intensitas sehingga mempermudah proses pengolahan pada tahapan berikutnya.



Gambar 3.4. Citra Partikel Mikroplastik Graycale

Konversi citra RGB ke grayscale dilakukan menggunakan persamaan sebagai berikut:

$$\text{Gray} = (0,299 \times R) + (0,587 \times G) + (0,114 \times B)$$

Keterangan:

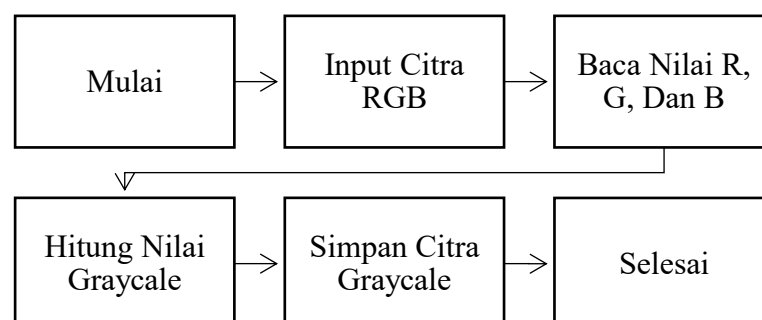
Gray = nilai intensitas citra grayscale

R = nilai kanal merah (Red)

G = nilai kanal hijau (Green)

B = nilai kanal biru (Blue)

Persamaan tersebut memberikan bobot yang berbeda pada setiap kanal warna sesuai dengan sensitivitas mata manusia terhadap warna. Kanal hijau memiliki kontribusi terbesar terhadap tingkat kecerahan, diikuti oleh kanal merah dan kanal biru.



Gambar 3.5. Diagram gambar Alir Proses Grayscale Conversionc.

c. Gaussian Blur

Tahap *Gaussian Blur* bertujuan untuk mengurangi *noise* atau gangguan pada citra dengan cara menghaluskan intensitas piksel. Proses ini membantu menghasilkan citra yang lebih bersih sehingga batas objek mikroplastik menjadi lebih mudah dikenali pada tahap deteksi.



Gambar 3.6. Citra Partikel Mikroplastik Gaussian Blur

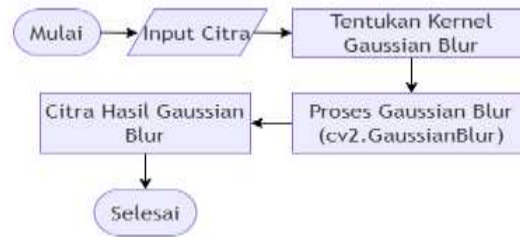
$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Keterangan:

$G(x, y)$ = fungsi Gaussian.

x, y = koordinat piksel.

σ = standar deviasi Gaussian.



Gambar 3.7. Diagram gambar Alir Proses *Gaussian Blur*

d. Thresholding

Thresholding merupakan salah satu tahapan penting dalam proses pengolahan citra yang bertujuan untuk memisahkan objek ikan dari latar belakang. Pada tahap ini, citra hasil filtering yang masih berupa citra grayscale diubah menjadi citra biner (binary image), yaitu citra yang hanya memiliki dua nilai intensitas, yaitu hitam (0) dan putih (255). Proses ini memudahkan sistem dalam mengenali batas objek sehingga proses pencarian kontur dapat dilakukan dengan lebih akurat.

Pada penelitian ini digunakan metode Binary Threshold yang bekerja dengan membandingkan nilai intensitas setiap piksel terhadap suatu nilai ambang (threshold). Apabila nilai intensitas piksel lebih besar dari nilai ambang, maka piksel tersebut akan diubah menjadi warna putih (255). Sebaliknya, apabila nilai intensitas lebih kecil atau sama dengan nilai ambang, maka piksel akan diubah menjadi warna hitam (0).

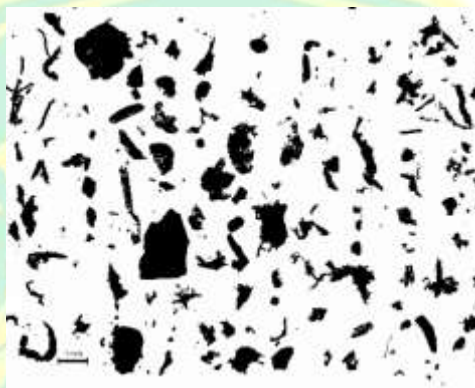
Secara matematis, proses thresholding dapat dinyatakan sebagai berikut.

$$g(x,y) = \begin{cases} 255, & \text{Jika } f(x,y) > T \\ 0, & \text{Jika } f(x,y) \leq T \end{cases}$$

Keterangan:

1. $g(x,y)$ = nilai piksel citra biner.
2. $f(x,y)$ = nilai intensitas piksel pada citra grayscale.
3. T = nilai ambang (threshold).

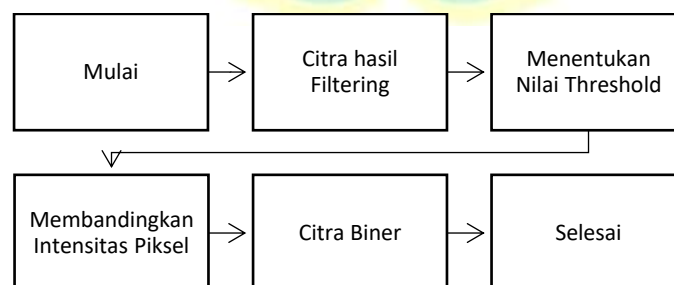
Pada implementasinya, proses thresholding dilakukan menggunakan pustaka OpenCV dengan fungsi `threshold()`. Fungsi ini mengubah citra grayscale menjadi citra biner berdasarkan nilai ambang yang telah ditentukan.



Gambar 3.8. Citra Partikel Mikroplastik Thresholding

Keterangan:

1. blur merupakan citra hasil proses filtering.
2. 127 adalah nilai ambang (threshold).
3. 255 merupakan nilai maksimum piksel putih.
4. `cv2.THRESH_BINARY` menunjukkan metode threshold



Gambar 3.9. Diagram gambar Alir Proses Thresholding

e. Operasi Morfologi (Opening dan Closing)

Tahap terakhir pada *preprocessing* adalah operasi morfologi yang terdiri atas *opening* dan *closing*. Operasi *opening* digunakan untuk menghilangkan objek kecil atau *noise* yang tidak diinginkan, sedangkan *closing* bertujuan untuk menutup celah atau lubang kecil pada objek sehingga bentuk partikel mikroplastik menjadi lebih utuh. Hasil dari tahapan ini menghasilkan citra yang lebih bersih dan siap diproses menggunakan algoritma Contour Detection.

Persamaan Opening:

$$A \circ B = (A \ominus B) \oplus B$$

Persamaan Closing:

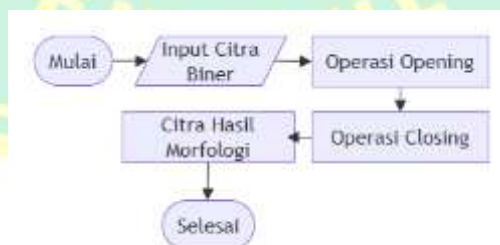
$$A \bullet B = (A \oplus B) \ominus B$$

Keterangan:

A = citra biner.

B = *structuring element*.

$\bullet$ = operasi *closing*.



Gambar 3.10. Diagram gambar Alur Operasi Morfologi (Opening dan Closing)

3.3.3 Contour detection

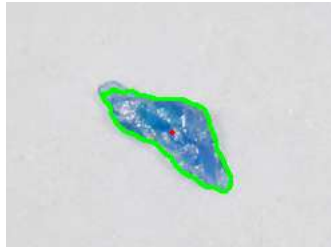
Setelah citra melalui tahapan *preprocessing*, proses selanjutnya adalah identifikasi objek menggunakan algoritma Contour Detection. Algoritma ini digunakan untuk mendeteksi batas (*boundary*) atau kontur dari setiap partikel mikroplastik yang terdapat pada citra biner. Kontur merupakan kumpulan titik yang membentuk garis luar suatu objek sehingga dapat digunakan untuk membedakan objek dari latar belakang.

Pada penelitian ini, proses *Contour Detection* dilakukan menggunakan pustaka OpenCV melalui fungsi `findContours()`. Fungsi tersebut bekerja dengan menelusuri piksel-piksel objek pada citra biner hasil *thresholding* dan operasi morfologi untuk memperoleh koordinat kontur setiap partikel mikroplastik. Setelah kontur berhasil ditemukan, sistem akan menggambarkan batas objek pada citra menggunakan fungsi `drawContours()` sehingga partikel mikroplastik dapat divisualisasikan dengan lebih jelas.



Gambar 3.11. Diagram gambar Alir Proses *Contour Detection*

Pada implementasinya, proses pencarian kontur dilakukan menggunakan fungsi `findContours()` pada pustaka OpenCV. Fungsi tersebut digunakan untuk memperoleh seluruh kontur yang terdapat pada citra biner.



Gambar 3.12. Gambar Partikel Berhasil di Beri Contur

3.3.4 Filtering Partikel

Setelah proses Contour Detection selesai dilakukan, sistem akan menghasilkan sejumlah kontur yang merepresentasikan seluruh objek yang terdapat pada citra. Namun, tidak semua kontur yang terdeteksi merupakan partikel mikroplastik karena masih terdapat objek lain, seperti *noise*, bayangan, maupun objek yang berasal dari latar belakang citra. Oleh karena itu, dilakukan proses filtering partikel untuk menyaring kontur sehingga hanya objek yang memenuhi kriteria sebagai partikel mikroplastik yang diproses pada tahap selanjutnya.

Pada penelitian ini, proses penyaringan dilakukan berdasarkan luas kontur (*Contour Area*) yang diperoleh dari setiap objek hasil deteksi. Luas kontur digunakan sebagai parameter untuk membedakan objek yang relevan dengan objek yang berukuran terlalu kecil maupun terlalu besar. Objek dengan luas di bawah batas minimum (*minimum area threshold*) dianggap sebagai *noise* sehingga tidak dilibatkan dalam proses analisis, sedangkan objek yang memenuhi nilai ambang akan dipertahankan sebagai kandidat partikel mikroplastik.

Perhitungan luas kontur dilakukan menggunakan fungsi `cv.contourArea()` pada pustaka OpenCV. Secara matematis, luas suatu kontur dapat dihitung menggunakan persamaan berikut.

$$A = \frac{1}{2} \left| \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \right|$$

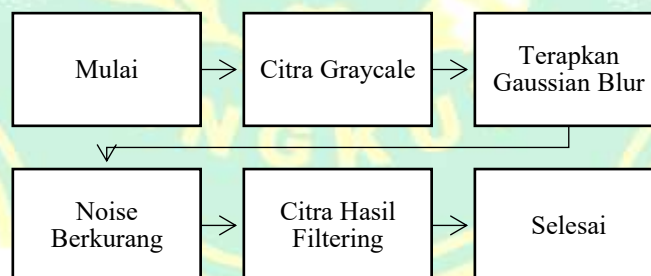
Keterangan:

A = luas kontur (piksel²).

x_i, y_i = koordinat titik ke- i pada kontur.

n = jumlah titik penyusun kontur.

Setelah proses penyaringan selesai dilakukan, sistem hanya menyimpan kontur yang memenuhi nilai ambang luas (*area threshold*) sebagai objek valid. Hasil penyaringan tersebut kemudian digunakan pada tahap analisis partikel untuk memperoleh informasi mengenai jumlah partikel serta karakteristik objek yang berhasil diidentifikasi.



Gambar 3.13. Diagram gambar Alir Proses Filtering Partiker

3.3.5 Analisis Partikel

Setelah proses *Filtering Partikel* selesai dilakukan, sistem akan melakukan analisis terhadap setiap partikel mikroplastik yang berhasil teridentifikasi.

Tahap ini bertujuan untuk memperoleh informasi mengenai karakteristik objek yang telah memenuhi kriteria penyaringan. Analisis dilakukan terhadap seluruh kontur yang telah lolos proses *filtering*, sehingga objek yang dianalisis merupakan partikel mikroplastik yang telah dipisahkan dari *noise* maupun objek lain yang tidak relevan.

Ringkasan Statistik	
Jumlah Partikel:	1
Rata-rata Luas (px ²):	ax: 3476.0)
Rata-rata Diameter (px):	66.527
Waktu Proses (detik):	0.082

Gambar 3.14. Hasil Pengukuran Jumlah, Area, Diameter

Pada penelitian ini, proses analisis dilakukan berdasarkan informasi yang diperoleh dari kontur hasil deteksi menggunakan algoritma Contour Detection. Parameter yang dianalisis meliputi jumlah partikel dan luas partikel (Contour Area). Kedua parameter tersebut digunakan untuk memberikan informasi mengenai karakteristik partikel mikroplastik yang terdapat pada citra hasil pengamatan.

a. Perhitungan Jumlah Partikel

Jumlah partikel diperoleh dengan menghitung seluruh kontur yang telah memenuhi proses *Filtering Partikel*. Setiap kontur yang lolos proses penyaringan dianggap sebagai satu partikel mikroplastik.

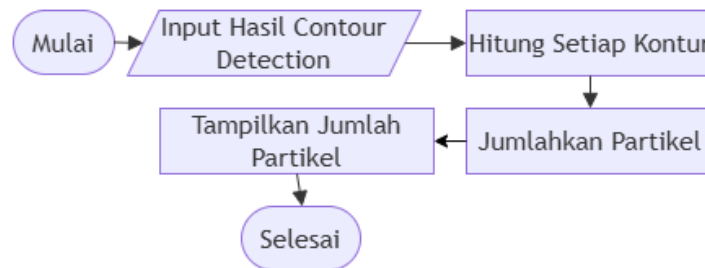
Persamaan jumlah partikel dapat dituliskan sebagai berikut.

$$N = \sum_{i=1}^n 1$$

Keterangan:

N = jumlah partikel yang terdeteksi.

n = jumlah kontur yang memenuhi proses penyaringan.



Gambar 3.15. Diagram gambar Alir Perhitungan Jumlah Partikel

b. Perhitungan Luas Partikel (Contour Area)

Luas partikel digunakan untuk mengetahui ukuran setiap objek mikroplastik berdasarkan daerah yang dibatasi oleh kontur. Nilai luas diperoleh menggunakan fungsi `cv.contourArea()` pada pustaka OpenCV.

Persamaan luas kontur adalah sebagai berikut.

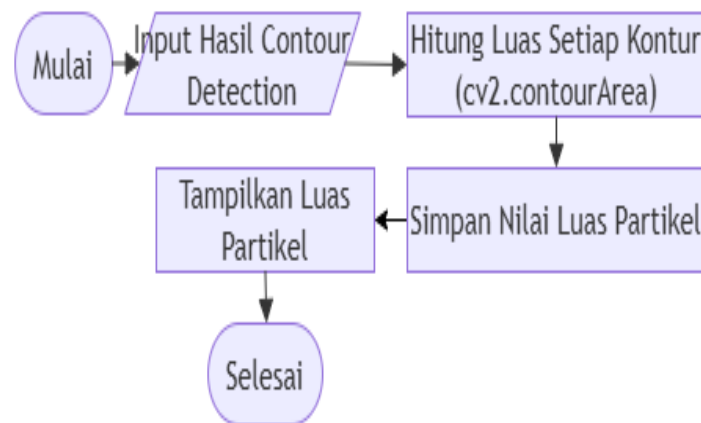
$$A = \frac{1}{2} \left| \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \right|$$

Keterangan:

A = luas kontur (piksel²).

x_i, y_i = koordinat titik ke- i pada kontur.

n = jumlah titik penyusun kontur.



Gambar 3.16. Diagram gambar Alir Perhitungan Luas Partikel

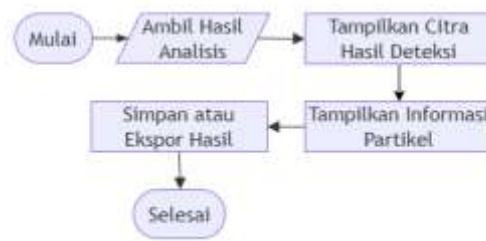
Setelah proses analisis selesai dilakukan, sistem menghasilkan informasi berupa jumlah partikel yang berhasil diidentifikasi serta luas masing-masing partikel. Informasi tersebut kemudian ditampilkan pada antarmuka aplikasi sebagai hasil identifikasi partikel mikroplastik.

3.3.6 Output Siste

Output sistem merupakan hasil akhir dari proses pengolahan citra yang meliputi *preprocessing*, *Contour Detection*, *Filtering Partikel*, dan analisis partikel. Hasil tersebut ditampilkan melalui antarmuka aplikasi dalam bentuk visualisasi citra dan informasi hasil identifikasi.

Visualisasi yang dihasilkan berupa citra yang telah diberi kontur pada setiap partikel mikroplastik yang berhasil dideteksi. Selain itu, sistem menampilkan informasi jumlah partikel dan luas (*Contour Area*) setiap partikel yang memenuhi kriteria penyaringan.

Output yang dihasilkan diharapkan dapat membantu proses identifikasi partikel mikroplastik secara lebih cepat, mudah, dan objektif dibandingkan pengamatan secara manual.



Gambar 3.17. Diagram gambar Alir ouput Sistem

3.4 Perancangan Antarmuka

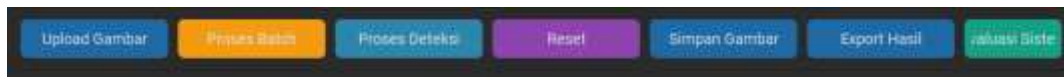
Perancangan antarmuka (*User Interface*) merupakan tahap perancangan tampilan aplikasi agar sistem mudah digunakan dalam proses identifikasi partikel mikroplastik. Antarmuka dirancang sederhana dan informatif, meliputi halaman beranda, halaman identifikasi citra, dan halaman hasil identifikasi.

3.4.1 Halaman Beranda

Halaman beranda merupakan tampilan awal yang muncul ketika aplikasi dijalankan. Halaman ini berfungsi sebagai media informasi mengenai aplikasi identifikasi partikel mikroplastik menggunakan algoritma **Contour Detection**. Selain itu, halaman beranda juga menyediakan tombol untuk memulai proses identifikasi citra.

Pada halaman ini ditampilkan beberapa komponen utama, yaitu:

- Judul aplikasi.
- Deskripsi singkat aplikasi.
- Tombol Mulai Identifikasi.



Gambar 3.19. Halaman Identifikasi

3.4.3 Halaman Hasil Identifikasi

Halaman hasil identifikasi menampilkan citra yang telah diberi kontur pada partikel mikroplastik yang berhasil dideteksi. Selain itu, sistem juga menyajikan informasi jumlah partikel dan luas (*Contour Area*) sebagai hasil analisis.

Komponen yang terdapat pada halaman ini meliputi:

- Citra hasil identifikasi.
- Jumlah partikel yang terdeteksi.
- Informasi luas partikel.
- Tombol **Simpan Hasil**
- Tombol **Kembali**.



Gambar 3.20. Halaman Hasil Identifikasi

3.5 Implementasi Aplikasi

Implementasi aplikasi merupakan tahap penerapan hasil perancangan sistem ke dalam bentuk aplikasi identifikasi partikel mikroplastik menggunakan algoritma *Contour Detection*. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dengan pustaka OpenCV sebagai *library* utama dalam pengolahan citra digital.

3.5.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan dalam penelitian ini berfungsi sebagai media untuk menjalankan aplikasi identifikasi partikel mikroplastik. Spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 3.12.

Tabel 3.3. Spesifikasi Perangkat Keras

No	Komponen	Spesifikasi
1	Prosesor	
2	RAM	
3	Penyimpanan	
4	Sistem Operasi	

3.5.2 Perangkat Lunak (Software)

Perangkat lunak yang digunakan pada penelitian ini berfungsi sebagai pendukung dalam proses pengembangan aplikasi, pengolahan citra, serta implementasi algoritma **Contour Detection**. Adapun perangkat lunak yang digunakan .

Table 3.4. Perangkat Lunak Yang di gunakan

No	Perangkat Lunak	Fungsi
1	Windows 11	Sistem operasi
2	Python	Bahasa pemrograman
3	OpenCV	Pengolahan citra dan Contour Detection
4	NumPy	Pengolahan data numerik
5	Visual Studio Code	Editor kode program
6	Matplotlib	Visualisasi citra
7	Git (<i>opsional</i>)	Version control

3.5.3 Implementasi Program

Implementasi program merupakan proses penerapan algoritma Contour Detection ke dalam aplikasi menggunakan bahasa pemrograman Python. Proses implementasi meliputi pembacaan citra, preprocessing, pendeteksian kontur, filtering partikel, analisis partikel, dan penyajian hasil identifikasi. OpenCV digunakan untuk pengolahan citra, sedangkan NumPy mendukung proses komputasi numerik.

3.6 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan aplikasi identifikasi partikel mikroplastik menggunakan algoritma Contour Detection berjalan sesuai dengan perancangan. Pengujian mencakup seluruh proses, mulai dari pemilihan citra, *preprocessing*, pendeteksian kontur, *filtering* partikel, analisis partikel, hingga penyajian hasil identifikasi.

Metode pengujian yang digunakan adalah *Black Box Testing*, yaitu pengujian berdasarkan masukan (*input*) dan keluaran (*output*) tanpa memperhatikan proses internal program. Pengujian dilakukan menggunakan beberapa citra mikroplastik untuk memastikan setiap fungsi aplikasi berjalan dengan baik.

Selain pengujian fungsional, dilakukan evaluasi hasil identifikasi dengan membandingkan hasil deteksi sistem terhadap pengamatan visual pada citra. Perbandingan ini bertujuan untuk menilai kesesuaian hasil deteksi partikel mikroplastik yang dihasilkan sistem.

3.6.1 Pengujian Black Box

Pengujian **Black Box** dilakukan dengan menguji seluruh fungsi utama aplikasi tanpa memperhatikan struktur kode program. Pengujian difokuskan pada kesesuaian antara masukan yang diberikan dengan keluaran yang dihasilkan oleh sistem. Suatu fungsi dinyatakan berhasil apabila sistem mampu menghasilkan keluaran sesuai dengan yang diharapkan.

Adapun skenario pengujian yang dilakukan pada penelitian ini dapat dilihat pada Tabel 14

Table 3.5. Skenario Pengujian Black Box

No.	Fitur yang Diuji	Input	Output yang Diharapkan
1.	Memilih citra	Pengguna memilih citra mikroplastik	Citra berhasil ditampilkan pada aplikasi
2.	Preprocessing	Citra diproses	Sistem menghasilkan

No.	Fitur yang Diuji	Input	Output yang Diharapkan
	citra		citra hasil <i>preprocessing</i>
3.	Contour Detection	Citra hasil <i>pre-processing</i>	Sistem mendeteksi kontur partikel mikroplastik
4.	Filtering Partikel	Kontur hasil deteksi	Sistem menyaring objek yang tidak memenuhi kriteria
5.	Analisis Partikel	Kontur hasil <i>filtering</i>	Sistem menampilkan jumlah partikel dan luas partikel
6.	Output Sistem	Seluruh proses selesai	Hasil identifikasi ditampilkan pada aplikasi

3.6.2 Evaluasi Hasil Identifikasi

Evaluasi hasil identifikasi dilakukan dengan membandingkan hasil deteksi sistem dengan objek partikel mikroplastik pada citra. Evaluasi ini bertujuan untuk mengetahui kemampuan algoritma Contour Detection dalam mengidentifikasi partikel mikroplastik.

1. kemampuan sistem mendeteksi partikel mikroplastik;

2. kemampuan sistem membedakan partikel mikroplastik dengan *noise* setelah proses *filtering*;
3. kesesuaian jumlah partikel yang terdeteksi dengan hasil pengamatan visual; dan
4. kemampuan sistem menampilkan jumlah partikel dan luas (*Contour Area*) partikel.

