

BAB IV ANALISIS DAN HASIL

4.1 Pengujian Komponen

Pada bab ini dilakukan pengujian terhadap sistem yang telah dirancang untuk mengetahui kinerja masing-masing komponen serta keseluruhan sistem dalam mendukung proses inspeksi visual. Pengujian dilakukan secara bertahap dimulai dari pengujian komponen, pengujian sistem, hingga pengujian pengolahan citra.

4.1.1 Pengujian Kamera USB

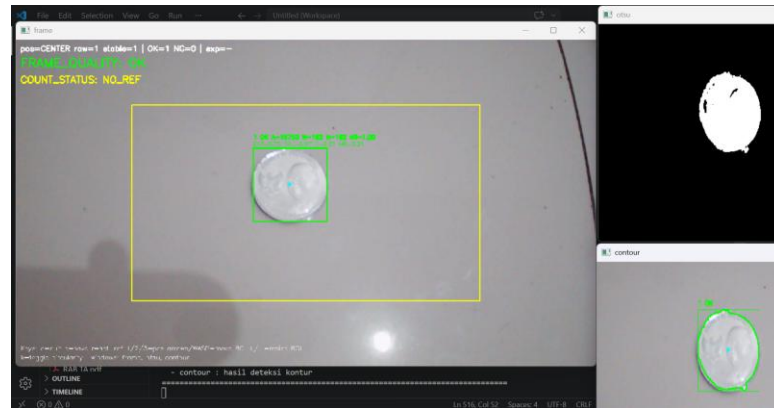
Pengujian dilakukan pada komponen kamera USB yang berfungsi sebagai perangkat utama dalam proses akuisisi citra objek pada sistem inspeksi visual. Pengujian ini bertujuan untuk memastikan bahwa kamera dapat terhubung dengan sistem, mampu menangkap citra objek, serta menghasilkan kualitas citra yang memadai untuk proses pengolahan citra selanjutnya.

Metode pengujian dilakukan dengan menghubungkan kamera USB ke laptop, kemudian menampilkan hasil tangkapan citra secara *real-time* menggunakan program berbasis *OpenCV*. Pengujian dilakukan dengan menghubungkan kamera USB ke laptop, kemudian menampilkan hasil tangkapan citra secara *real-time* menggunakan program berbasis *OpenCV*. Pengujian meliputi koneksi kamera dengan sistem, pengambilan citra objek, serta kualitas citra yang dihasilkan sebagaimana ditunjukkan pada **gambar 4.1** dan **gambar 4.2**.



Gambar 4. 1 Koneksi Kamera USB dengan Sistem

Koneksi kamera USB yang telah berhasil kemudian dilanjutkan dengan pengujian akuisisi dan deteksi objek, seperti ditunjukkan pada **gambar 4.2**.



Gambar 4. 2 Hasil Akuisisi dan Deteksi Objek Menggunakan Kamera USB

Berdasarkan hasil pengujian yang ditunjukkan pada **gambar 4.1**, kamera USB berhasil terhubung dengan sistem dan mampu menampilkan citra objek secara *real-time*. Selain itu, citra yang dihasilkan juga dapat diproses menggunakan metode *thresholding* Otsu dan deteksi kontur, sehingga objek dapat teridentifikasi dengan jelas. Untuk mengetahui hasil pengujian secara lebih rinci, dilakukan pengujian terhadap beberapa parameter yang disajikan dalam **tabel 4.1**.

Tabel 4. 1 Hasil Pengujian Kamera USB

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Koneksi kamera dengan sistem	Kamera berhasil terhubung dan dapat diakses oleh sistem.	Berhasil	Kamera dapat digunakan sebagai perangkat akuisisi citra.
2	Pengambilan citra objek	Objek berhasil ditangkap dan ditampilkan pada layar monitoring.	Berhasil	Citra objek dapat ditampilkan secara <i>real-time</i> .

3	Tampilan <i>real-time</i>	Tampilan kamera berjalan stabil selama pengujian.	Berhasil	Sistem dapat menampilkan citra objek secara kontinu. Citra dapat digunakan untuk proses
4	Kualitas citra	Objek terlihat jelas pada area inspeksi.	Baik	grayscale, <i>thresholding</i> Otsu, dan deteksi kontur.
5	Kesesuaian citra untuk pengolahan citra	Citra berhasil diproses pada tahap pengolahan citra berikutnya.	Sesuai	Hasil kamera mendukung proses inspeksi visual.

Berdasarkan hasil pengujian yang telah dilakukan, kamera USB berhasil terhubung dengan sistem yang ditunjukkan dengan munculnya tampilan citra pada layar. Pada pengujian pengambilan citra objek, kamera mampu menangkap objek dengan baik dan menampilkan citra secara *real-time*.

Selain itu, citra yang dihasilkan memiliki kualitas yang cukup baik, ditunjukkan dengan tingkat pencahayaan yang memadai dan objek yang terlihat jelas. Hal ini memungkinkan citra untuk diproses lebih lanjut menggunakan metode *thresholding* Otsu dan deteksi kontur, sehingga objek dapat teridentifikasi dengan baik.

4.1.2 Pengujian Motor *Stepper* dan Driver TB6600

Pengujian motor *stepper* dan driver TB6600 dilakukan untuk memastikan bahwa aktuator dapat bergerak sesuai dengan perintah yang diberikan oleh sistem serta driver mampu mengendalikan pergerakan motor dengan baik. Pengujian meliputi pergerakan motor ke posisi kiri, tengah, dan kanan, kontrol arah putaran, respon terhadap perintah, koneksi driver dengan Arduino, serta stabilitas sistem selama operasi, hasil pengujian ditunjukkan pada **gambar 4.3** hingga **gambar 4.7**.



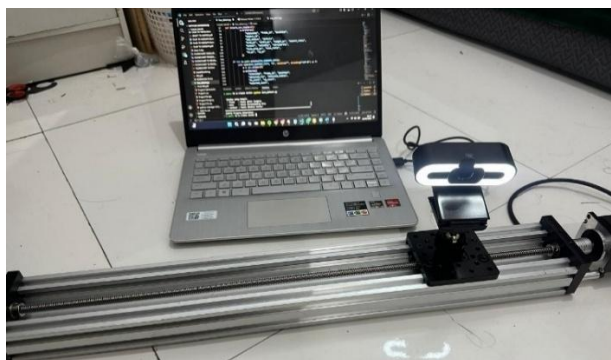
Gambar 4. 3 Pergerakan Motor *Stepper* ke Posisi Kiri

Selanjutnya, motor *stepper* bergerak ke posisi tengah seperti ditunjukkan pada **gambar 4.4**.



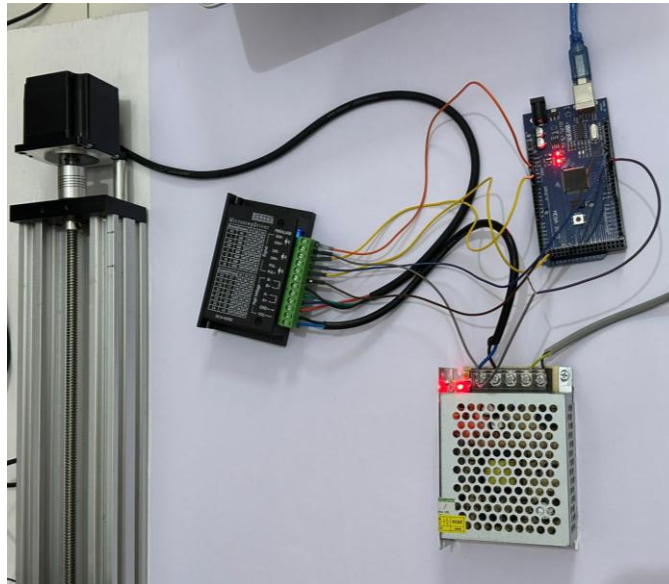
Gambar 4. 4 Pergerakan Motor *Stepper* ke Posisi Tengah

Kemudian, motor *stepper* bergerak ke posisi kanan seperti ditunjukkan pada **gambar 4.5**.



Gambar 4. 5 Pergerakan Motor *Stepper* ke Posisi Kanan

Driver TB6600 dan wiring sistem yang digunakan ditunjukkan pada **Gambar 4.6.**



Gambar 4. 6 Driver TB6600 dan Wiring Sistem

Setelah rangkaian terpasang, dilakukan pengujian respon perintah dan kontrol motor. Hasil pengujian ditunjukkan pada **gambar 4.7.**

```

Output  Serial Monitor X
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM3')

CMD=LEFT
STATUS=BERGERAK_KIRI
POS=Kiri
DIST=98
CMD=LEFT
STATUS=BERGERAK_KIRI
STATUS=SUDAH_DI_KIRI
POS=Kiri
DIST=142
CMD=SETCENTER
STATUS=SETCENTER_OK
POS=Tengah
CMD=RIGHT
STATUS=BERGERAK_KANAN
  
```

Gambar 4. 7 Pengujian Respon Perintah dan Kontrol Motor

Berdasarkan hasil pengujian yang ditunjukkan pada **gambar 4.7**, motor *stepper* dapat bergerak ke posisi kiri, tengah, dan kanan sesuai dengan jumlah langkah (step) yang telah ditentukan. Selain itu, driver TB6600 mampu mengendalikan arah pergerakan motor dan merespon perintah dari sistem dengan baik. Hasil pengujian selengkapnya disajikan pada **tabel 4.2.**

Tabel 4. 2 Hasil Pengujian Motor *Stepper* dan Driver TB6600

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Koneksi driver TB6600 dengan Arduino	Driver berhasil terhubung dengan Arduino melalui pin STEP, DIR, dan ENA.	Berhasil	Koneksi antar komponen berjalan dengan baik.
2	Pergerakan motor ke posisi kiri	Motor bergerak ke posisi kiri sesuai perintah sistem.	Berhasil	Pergerakan digunakan untuk membawa kamera menuju sisi kiri area inspeksi.
3	Pergerakan motor ke posisi tengah	Motor berhasil kembali ke posisi tengah setelah bergerak dari posisi kiri atau kanan.	Berhasil	Posisi tengah digunakan sebagai titik acuan awal dan posisi standby sistem.
4	Pergerakan motor ke posisi kanan	Motor bergerak ke posisi kanan sesuai perintah sistem.	Berhasil	Pergerakan digunakan untuk membawa kamera menuju sisi kanan area inspeksi.
5	Kontrol arah putaran motor	Arah putaran motor sesuai dengan perintah yang diberikan.	Berfungsi	Perubahan arah dikendalikan melalui sinyal DIR pada driver TB6600.

6	Respon motor terhadap perintah	Motor dapat bergerak setelah menerima perintah dari sistem.	Baik	Perintah dapat diberikan melalui antarmuka sistem maupun tombol kontrol.
7	Kestabilan gerak motor	Motor bergerak stabil pada saat pengujian posisi kamera.	Stabil	Gerakan motor mampu mendukung perpindahan kamera pada mekanisme linear actuator.
8	Stabilitas driver TB6600	Driver bekerja stabil selama proses pergerakan motor.	Stabil	Driver TB6600 mampu mengendalikan motor <i>stepper</i> selama operasi sistem.

Berdasarkan **tabel 4.2**, motor *stepper* NEMA23 dan driver TB6600 dapat bekerja dengan baik sebagai sistem penggerak kamera. Driver TB6600 berhasil menerima sinyal kontrol dari Arduino Mega 2560 melalui pin STEP, DIR, dan ENA. Motor *stepper* dapat bergerak menuju posisi kiri, kembali ke posisi tengah, dan bergerak menuju posisi kanan sesuai perintah sistem.

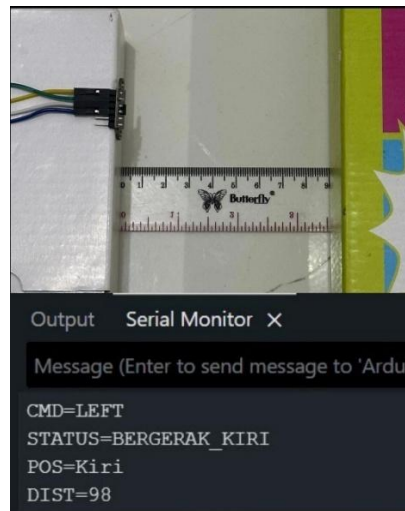
Hasil pengujian juga menunjukkan bahwa arah putaran motor dapat berubah sesuai perintah kiri dan kanan. Motor mampu merespon perintah dari sistem dengan baik, baik melalui komunikasi serial maupun tombol kontrol. Selama proses pengujian, pergerakan motor berjalan stabil sehingga dapat digunakan untuk menggerakkan kamera pada mekanisme linear actuator. Dengan demikian, motor *stepper* NEMA23 dan driver TB6600 dinilai sesuai untuk digunakan sebagai penggerak posisi kamera pada sistem inspeksi visual.

4.1.3 Pengujian Sensor VL53L0X

Pengujian sensor jarak VL53L0X dilakukan untuk mengetahui tingkat akurasi pembacaan jarak terhadap objek. Sensor ini bekerja dengan prinsip *Time of*

Flight (ToF), yaitu mengukur waktu tempuh cahaya inframerah dari sensor ke objek dan kembali lagi ke sensor.

Metode pengujian dilakukan dengan membandingkan nilai jarak yang dibaca sensor terhadap jarak sebenarnya yang diukur menggunakan penggaris. Pengujian dilakukan pada tiga titik posisi, yaitu kiri, tengah, dan kanan, sebagaimana ditunjukkan pada **gambar 4.8** untuk posisi kiri.



Gambar 4. 8 Pengujian Sensor VL53L0X Posisi Kiri

Selanjutnya, dilakukan pengujian sensor VL53L0X pada posisi tengah seperti ditunjukkan pada **gambar 4.9**.



Gambar 4. 9 Pengujian Sensor VL53L0X Posisi Tengah

Kemudian, dilakukan pengujian sensor VL53L0X pada posisi kanan seperti ditunjukkan pada **gambar 4.10**.



Gambar 4. 10 Pengujian Sensor VL53L0X Posisi Kanan

Program yang digunakan untuk pembacaan sensor jarak ditunjukkan pada gambar 4.11.

```
int jarak = sensor.readRangeContinuousMillimeters();
```

Gambar 4. 11 Potongan program sensor jarak

Potongan program tersebut digunakan untuk membaca nilai jarak dari sensor VL53L0X dalam satuan milimeter. Sensor melakukan pengukuran secara kontinu, kemudian nilai hasil pembacaan dikirimkan ke mikrokontroler untuk diproses lebih lanjut dan ditampilkan melalui serial monitor maupun LCD, nilai pembacaan sensor dalam satuan milimeter kemudian dikonversi ke sentimeter untuk mempermudah proses analisis. Hasil pengujian sensor jarak ditunjukkan pada tabel 4.3.

Tabel 4. 3 Hasil Pengujian Sensor Jarak VL53L0X

No	Posisi Kamera	Jarak Asli (cm)	Jarak Sensor (cm)	Selisih Jarak (cm)	Error (%)	Keterangan
1	Kiri	9,0	9,8	0,8	8,89%	Sensor mampu membaca jarak objek pada posisi kiri dengan selisih pembacaan 0,8 cm.

						Sensor mampu membaca jarak objek pada posisi tengah, namun terdapat selisih pembacaan sebesar 1,5 cm.
2	Tengah	10,0	11,5	1,5	15,00%	
						Sensor mampu membaca jarak objek pada posisi kanan, namun selisih pembacaan lebih besar dibandingkan posisi lainnya.
3	Kanan	12,0	14,0	2,0	16,67%	

Berdasarkan hasil pengujian, sensor VL53L0X mampu mendeteksi jarak objek dengan baik, namun terdapat selisih antara nilai pembacaan sensor dan nilai referensi. Pada jarak 9,0 cm, sensor membaca sebesar 9,8 cm dengan *error* sebesar 0,8 cm atau 8,89%. Pada jarak 10,0 cm, *error* meningkat menjadi 1,5 cm atau 15%, dan pada jarak 12,0 cm *error* mencapai 2,0 cm atau 16,67%. Nilai *error* terbesar terdapat pada posisi kanan, yaitu sebesar 16,67%. Perbedaan pembacaan ini dapat dipengaruhi oleh sudut pantulan sensor terhadap permukaan objek, posisi objek, pencahayaan sekitar, atau kestabilan jarak antara sensor dan objek. Meskipun terdapat selisih pembacaan, sensor tetap dapat digunakan untuk menampilkan informasi jarak objek pada sistem.

Hal ini menunjukkan bahwa sensor memiliki kecenderungan menghasilkan nilai pembacaan yang lebih besar dibandingkan nilai sebenarnya. Selain itu, dapat diamati bahwa nilai *error* meningkat seiring dengan bertambahnya jarak pengukuran. Perbedaan hasil pengukuran ini dapat disebabkan oleh beberapa faktor seperti posisi pemasangan sensor yang tidak tegak lurus terhadap objek, permukaan objek yang mempengaruhi pantulan cahaya serta keterbatasan akurasi sensor pada jarak tertentu.

Berdasarkan pengujian yang telah dilakukan, sensor VL53L0X dapat digunakan untuk mengukur jarak pada sistem inspeksi visual yang dirancang. Meskipun terdapat *error* pada hasil pembacaan, sensor masih dapat memberikan informasi jarak yang cukup baik untuk kebutuhan sistem, dengan kecenderungan *error* yang meningkat pada jarak yang lebih jauh.

4.1.4 Pengujian LCD I2C

Pengujian LCD dilakukan untuk memastikan LCD dapat menampilkan informasi dari sistem. Metode pengujian dilakukan dengan menghubungkan LCD I2C ke mikrokontroler Arduino Mega 2560, kemudian sistem dijalankan untuk menampilkan informasi status pergerakan motor dan posisi kamera. Pengujian dilakukan untuk memastikan LCD mampu menampilkan data secara jelas dan dapat memperbarui informasi sesuai kondisi sistem secara *real-time* seperti ditunjukkan pada **gambar 4.12**.



Gambar 4. 12 Pengujian LCD I2C

Berdasarkan **gambar 4.12**, LCD I2C berhasil menyala dan menampilkan informasi dari sistem. Tampilan pada LCD menunjukkan bahwa komunikasi antara LCD I2C dan Arduino Mega 2560 dapat berjalan dengan baik. Informasi yang ditampilkan pada LCD digunakan sebagai indikator pendukung untuk mengetahui status sistem, seperti kondisi pergerakan motor dan posisi kamera. Untuk mengetahui hasil pengujian LCD I2C secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter pengujian yang disajikan pada **tabel 4.4**.

Tabel 4. 4 Hasil Pengujian LCD I2C

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Koneksi LCD I2C dengan Arduino	LCD berhasil terhubung dan menyala saat sistem dijalankan.	Berhasil	Koneksi LCD I2C dengan Arduino berjalan dengan baik.
2	Tampilan teks pada LCD	Teks dapat ditampilkan pada layar LCD dan mudah dibaca.	Berhasil	LCD dapat menampilkan informasi seperti status sistem dan posisi kamera.
3	Update data pada LCD	Informasi posisi dan status dapat berubah mengikuti proses sistem.	Berhasil	LCD mampu memperbarui tampilan saat sistem berpindah status.
4	Tampilan status posisi kamera	Posisi kamera berhasil ditampilkan sesuai proses pergerakan sistem.	Berhasil	Tampilan LCD membantu pengguna mengetahui posisi kamera saat sistem berjalan.
5	Kestabilan tampilan LCD	Tampilan LCD tidak mengalami gangguan selama pengujian.	Stabil	LCD dapat digunakan sebagai media informasi pendukung pada sistem inspeksi visual.

Berdasarkan **tabel 4.4**, LCD I2C 16×2 dapat bekerja dengan baik sebagai media tampilan informasi pada sistem inspeksi visual. LCD berhasil terhubung

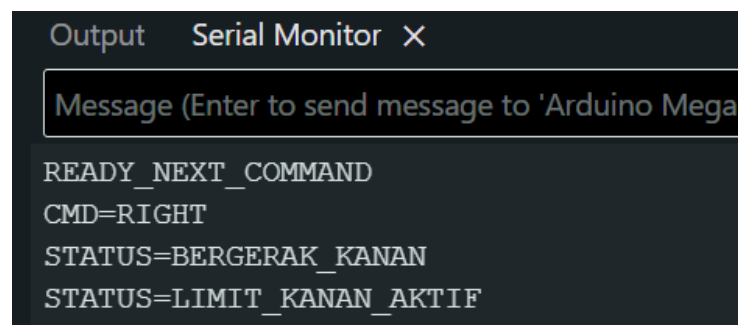
dengan Arduino melalui komunikasi I2C dan mampu menampilkan teks dengan jelas. Informasi yang ditampilkan pada LCD dapat diperbarui sesuai kondisi sistem, seperti status proses dan posisi kamera.

Hasil pengujian menunjukkan bahwa LCD mampu menampilkan informasi secara stabil selama sistem dijalankan. Dengan demikian, LCD I2C dinilai sesuai untuk digunakan sebagai indikator tampilan pendukung yang membantu pengguna mengetahui kondisi sistem secara langsung.

4.1.5 Pengujian Limit Switch

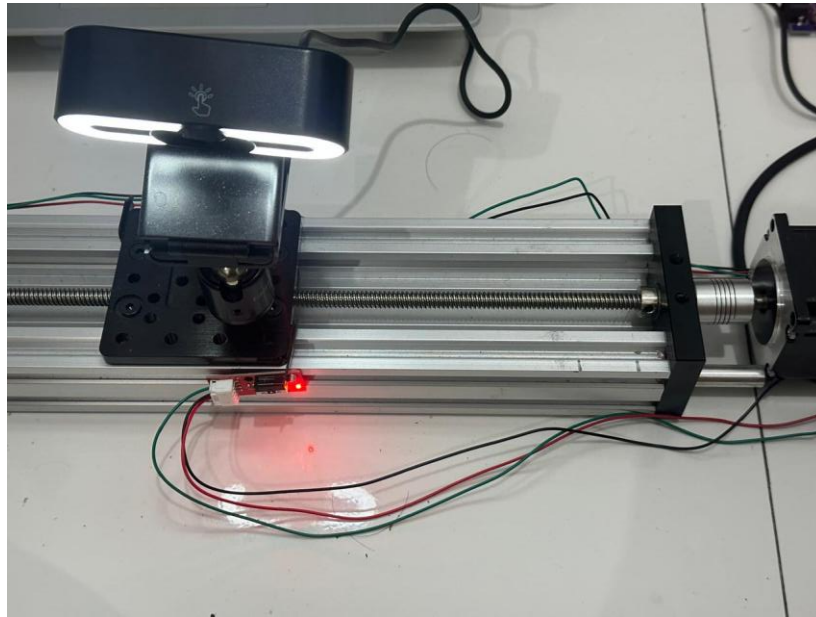
Pengujian limit switch dilakukan untuk memastikan bahwa sistem mampu mendeteksi batas pergerakan carriage kamera pada rel C-Beam serta menghentikan motor *stepper* secara otomatis ketika mencapai posisi batas yang telah ditentukan. Pengujian ini bertujuan untuk mencegah terjadinya benturan mekanis pada ujung rel dan memastikan mekanisme pergerakan kamera dapat beroperasi dengan aman dan terkontrol.

Metode pengujian dilakukan dengan menghubungkan limit switch kiri dan limit switch kanan ke Arduino Mega 2560. Selanjutnya carriage kamera digerakkan menuju posisi kiri dan kanan menggunakan perintah yang dikirim melalui sistem. Ketika carriage kamera menyentuh limit switch, Arduino akan mendeteksi perubahan kondisi input dan mengirimkan informasi status ke Serial Monitor. Pada saat yang sama, motor *stepper* akan dihentikan secara otomatis sebagai respons terhadap aktivasi limit switch. Hasil pengujian limit switch kanan ditunjukkan pada **gambar 4.13**.



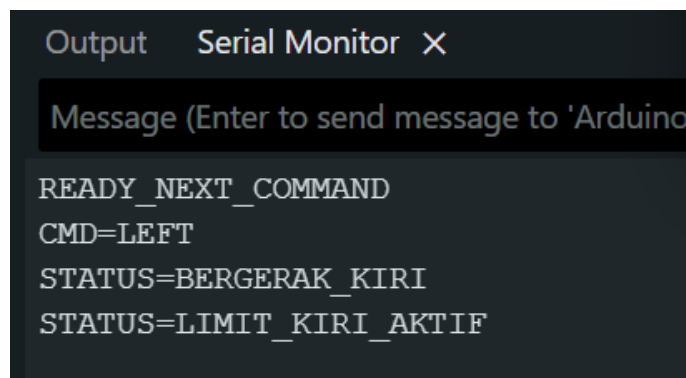
Gambar 4. 13 Tampilan Status Limit Switch kanan pada Serial Monitor

Selanjutnya, dilakukan pengujian limit switch kanan seperti ditunjukkan pada **Gambar 4.14**.



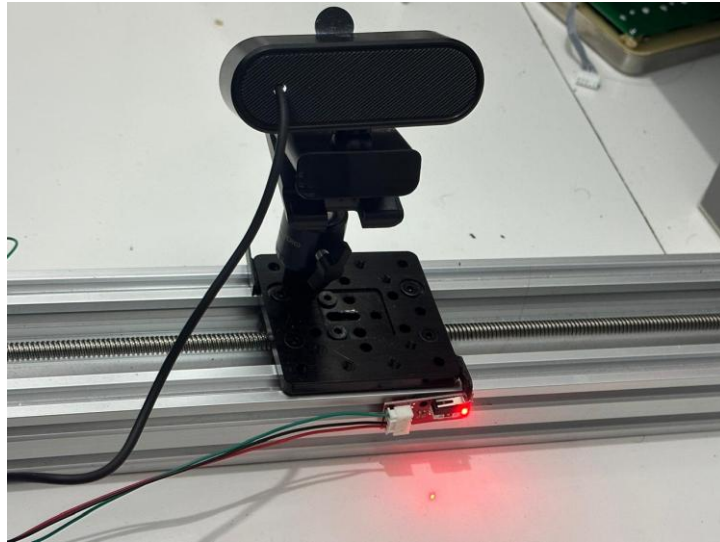
Gambar 4. 14 Pengujian Limit Switch Kanan

Tampilan status limit switch kiri pada *Serial Monitor* ditunjukkan pada **gambar 4.15**.



Gambar 4. 15 Tampilan Status Limit Switch kiri pada Serial Monitor

Kemudian, dilakukan pengujian limit switch kiri seperti ditunjukkan pada **gambar 4.16**.



Gambar 4. 16 Pengujian Limit Switch Kiri

Berdasarkan hasil pengujian yang ditunjukkan pada **gambar 4.14** dan **gambar 4.16** limit switch berhasil mendeteksi ketika aktuator mencapai batas pergerakan yang telah ditentukan. Aktivasi limit switch ditandai dengan munculnya informasi status pada status serial monitor berupa pesan “STATUS=LIMIT_KANAN_AKTIF” atau “STATUS=LIMIT_KIRI_AKTIF” sesuai dengan limit switch yang tertekan.

Selain ditunjukkan oleh informasi pada Serial Monitor, aktivasi limit switch juga dapat diamati melalui indikator LED pada modul sensor yang menyala ketika aktuator menyentuh tuas limit switch. Setelah limit switch aktif, sistem secara otomatis menghentikan pergerakan motor *stepper* sehingga aktuator tidak bergerak melewati batas area kerja yang telah ditentukan. Untuk mengetahui hasil pengujian secara lebih rinci, dilakukan pengujian terhadap beberapa parameter yang disajikan pada **tabel 4.5**.

Tabel 4. 5 Hasil Pengujian Limit Switch

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Koneksi limit switch kiri dengan sistem	Status limit switch kiri berhasil terbaca oleh Arduino.	Berhasil	Limit switch kiri dapat digunakan sebagai

				pendeteksi batas pergerakan kamera.
2	Koneksi limit switch kanan dengan sistem	Status limit switch kanan berhasil terbaca oleh Arduino.	Berhasil	Limit switch kanan dapat digunakan sebagai pendeteksi batas pergerakan kamera.
3	Deteksi batas pergerakan carriage kamera	Limit switch aktif ketika mekanisme carriage menyentuh batas kiri atau kanan.	Berhasil	Limit switch mampu mendeteksi posisi batas pada mekanisme linear actuator.
4	Penghentian motor <i>stepper</i> otomatis	Motor berhenti saat limit switch tertekan pada batas pergerakan.	Berhasil	Limit switch berfungsi sebagai pengaman agar carriage kamera tidak bergerak melewati batas lintasan.
5	Tampilan status pada serial monitor	Status limit switch dapat ditampilkan pada serial monitor.	Berhasil	Informasi status digunakan untuk memantau kondisi limit switch selama pengujian.
6	Indikator LED limit switch	LED menyala ketika limit switch dalam kondisi aktif.	Berhasil	Indikator LED membantu memastikan kondisi limit switch secara visual.

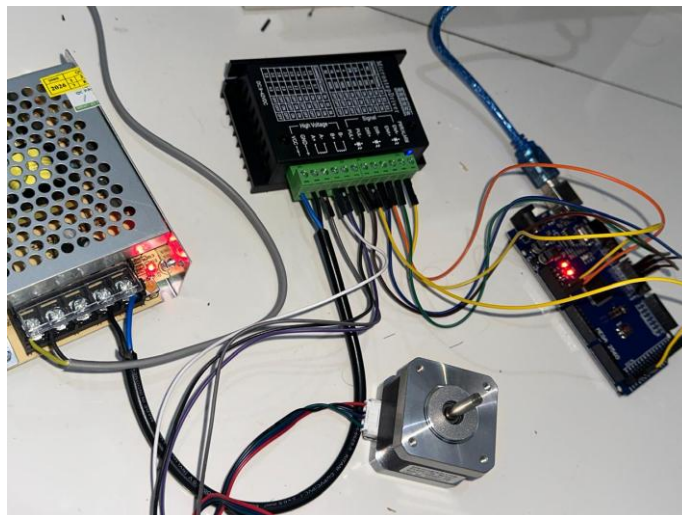
Berdasarkan hasil pengujian yang ditunjukkan pada **tabel 4.5**, kedua limit switch yang digunakan pada sistem mampu bekerja dengan baik sesuai dengan fungsi yang telah dirancang. Sistem dapat mendeteksi aktivasi limit switch kiri maupun kanan ketika carriage kamera mencapai batas pergerakan pada rel C-Beam. Selain itu, sistem juga mampu menampilkan status aktivasi pada serial monitor serta menghentikan motor *stepper* secara otomatis saat limit switch tertekan.

Hasil pengujian ini menunjukkan bahwa limit switch telah berfungsi dengan baik sebagai sistem pengaman pada mekanisme pergerakan kamera. Dengan adanya limit switch, pergerakan carriage kamera pada rel C-Beam dapat dikendalikan dengan lebih aman dan terkontrol sehingga dapat meminimalkan risiko benturan maupun kerusakan mekanis akibat pergerakan yang melebihi batas area kerja yang telah ditentukan.

4.1.6 Pengujian Catu Daya

Pengujian catu daya dilakukan untuk memastikan power supply mampu menyuplai tegangan yang dibutuhkan oleh sistem inspeksi visual. Catu daya digunakan untuk memberikan sumber tegangan pada driver motor *stepper*, Arduino Mega 2560, sensor, dan komponen pendukung lainnya agar sistem dapat bekerja dengan stabil.

Metode pengujian dilakukan dengan menghubungkan power supply 12V ke driver TB6600 dan sistem kontrol. Pengujian dilakukan dengan menyalakan sistem kemudian mengamati kondisi kerja komponen seperti Arduino, driver motor, LCD, dan sensor saat sistem beroperasi sebagaimana ditunjukkan pada **gambar 4.17**.



Gambar 4. 17 Pengujian Power Supply pada Sistem

Berdasarkan hasil pengujian yang ditunjukkan pada **gambar 4.17** power supply 12V berhasil menyuplai daya ke driver TB6600, Arduino Mega 2560, dan motor *stepper*. Sistem dapat menyala dengan baik yang ditandai dengan aktifnya

indikator LED pada driver dan mikrokontroler. Untuk mengetahui hasil pengujian secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter yang disajikan pada **tabel 4.6**.

Tabel 4. 6 Hasil Pengujian Catu Daya

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Tegangan output power supply	Tegangan output terbaca sebesar 12V.	Sesuai	Tegangan output sesuai dengan kebutuhan sistem utama.
2	Suplai daya ke driver TB6600	Driver TB6600 berhasil menyala dan dapat mengendalikan motor <i>stepper</i> .	Berhasil	Driver menerima suplai daya dengan baik dari power supply 12V.
3	Suplai daya ke motor <i>stepper</i>	Motor <i>stepper</i> dapat bergerak sesuai perintah sistem.	Berhasil	Catu daya mampu mendukung kerja motor <i>stepper</i> melalui driver TB6600.
4	Suplai daya ke rangkaian kontrol	Arduino Mega 2560 dan rangkaian kontrol dapat menjalankan program sistem.	Berfungsi	Arduino bekerja sebagai pengendali sistem dan membaca input/output yang digunakan.
5	Suplai daya ke komponen pendukung	Komponen pendukung dapat menyala dan berfungsi saat sistem dijalankan.	Berhasil	Catu daya mendukung kebutuhan perangkat pendukung melalui rangkaian suplai yang digunakan.

6	Stabilitas sistem	Sistem dapat bekerja stabil selama proses pengujian.	Stabil	Tidak ditemukan gangguan suplai daya selama pengujian berlangsung.
---	-------------------	--	--------	--

Berdasarkan **tabel 4.6**, catu daya mampu menyuplai kebutuhan tegangan sistem dengan baik selama proses pengujian berlangsung. Power supply 12V dapat memberikan sumber daya ke driver TB6600 sehingga motor *stepper* NEMA23 dapat dikendalikan sesuai perintah sistem. Selain itu, rangkaian kontrol dan komponen pendukung seperti LCD, sensor, dan indikator juga dapat bekerja selama sistem dijalankan.

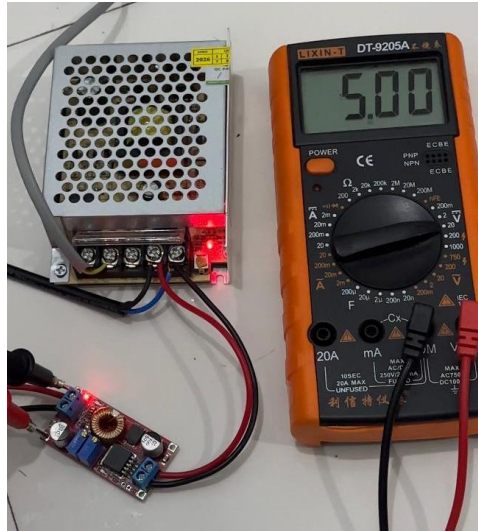
Hasil pengujian menunjukkan bahwa sistem dapat beroperasi secara stabil tanpa gangguan seperti tegangan tidak stabil atau sistem mati secara tiba-tiba. Hal ini menunjukkan bahwa catu daya yang digunakan telah mampu mendukung kebutuhan perangkat keras pada sistem inspeksi visual. Dengan demikian, sistem catu daya dinilai mampu mendukung proses kerja perangkat keras secara stabil dan dapat digunakan pada tahap integrasi sistem selanjutnya.

4.1.7 Step Down Buck Converter XL4015

Pengujian step down DC-DC XL4015 dilakukan untuk memastikan modul dapat menurunkan tegangan keluaran dari power supply 12 VDC menjadi tegangan 5 VDC yang stabil sebelum digunakan sebagai sumber daya untuk Arduino Mega 2560, LCD I2C, sensor VL53L0X, dan komponen pendukung lainnya. Pengujian ini bertujuan untuk memastikan tegangan keluaran yang dihasilkan sesuai dengan kebutuhan sistem sehingga seluruh komponen dapat beroperasi dengan aman dan stabil.

Metode pengujian dilakukan dengan menghubungkan terminal input modul XL4015 ke power supply 12 VDC. Selanjutnya tegangan keluaran modul diukur menggunakan multimeter digital pada terminal output modul. Pengaturan tegangan dilakukan dengan memutar trimpot pengatur tegangan hingga diperoleh tegangan

keluaran sebesar 5 VDC. Hasil pengujian step down DC-DC XL4015 ditunjukkan pada **gambar 4.18**.



Gambar 4. 18 Hasil Buck Converter

Berdasarkan hasil pengujian yang ditunjukkan pada **gambar 4.18** modul XL4015 mampu menurunkan tegangan input sebesar 12 VDC menjadi tegangan keluaran sebesar 5 VDC sesuai dengan kebutuhan sistem. Tegangan keluaran yang dihasilkan relatif stabil dan berada dalam rentang kerja yang aman untuk digunakan sebagai sumber daya Arduino Mega 2560 dan perangkat pendukung lainnya. Untuk mengetahui hasil pengujian secara lebih rinci, dilakukan pengukuran terhadap parameter tegangan masukan dan tegangan keluaran yang disajikan pada **tabel 4.7**.

Tabel 4. 7 Hasil Pengujian Step Down Buck Converter XL4015

No	Parameter Pengujian	Hasil Pengamatan	Status	Keterangan
1	Tegangan masukan	Tegangan input terbaca sebesar 12VDC.	Sesuai	Tegangan masukan berasal dari power supply utama sistem.

2	Tegangan keluaran	Tegangan output terbaca sebesar 5VDC.	Sesuai	Tegangan keluaran sesuai dengan kebutuhan komponen 5V pada sistem.
3	Stabilitas tegangan keluaran	Tegangan keluaran tetap stabil dan tidak mengalami fluktuasi signifikan.	Stabil	Modul dapat memberikan suplai 5V secara stabil selama pengujian.
4	Suplai ke komponen pendukung	Komponen pendukung dapat menyala dan bekerja dengan baik.	Berhasil	XL4015 digunakan sebagai sumber tegangan 5V untuk komponen pendukung sistem.
5	Kelayakan penggunaan pada sistem	Sistem dapat berjalan tanpa gangguan pada bagian komponen 5V.	Layak	Modul XL4015 sesuai digunakan sebagai penurun tegangan dari 12VDC ke 5VDC.

Berdasarkan **tabel 4.7**, modul step down buck converter XL4015 mampu bekerja dengan baik dalam menurunkan tegangan dari 12VDC menjadi 5VDC. Tegangan keluaran yang dihasilkan sesuai dengan kebutuhan komponen pendukung pada sistem inspeksi visual, seperti LCD I2C, sensor VL53L0X, limit switch module, dan rangkaian 5V lainnya.

Hasil pengujian menunjukkan bahwa tegangan keluaran XL4015 stabil selama sistem dijalankan dan tidak mengalami fluktuasi yang signifikan. Dengan demikian, modul XL4015 dinilai layak digunakan sebagai penurun tegangan dari power supply 12VDC ke 5VDC untuk mendukung kerja komponen elektronika pada sistem inspeksi visual.

4.2 Pengujian Aplikasi Kontrol Delphi

Pengujian aplikasi kontrol berbasis Delphi dilakukan untuk memastikan bahwa sistem yang telah dirancang mampu berfungsi sebagai antarmuka utama dalam mengendalikan seluruh proses inspeksi visual. Aplikasi ini berperan sebagai penghubung antara pengguna dengan perangkat keras, khususnya mikrokontroler dan sistem mekanik, melalui komunikasi serial.

Selain sebagai pengendali, aplikasi Delphi juga berfungsi untuk menampilkan informasi kondisi sistem secara *real-time*, seperti status koneksi, posisi kamera, serta hasil pengukuran dan klasifikasi objek. Oleh karena itu, pengujian dilakukan secara menyeluruh terhadap setiap komponen pada aplikasi guna memastikan sistem dapat bekerja secara terintegrasi.

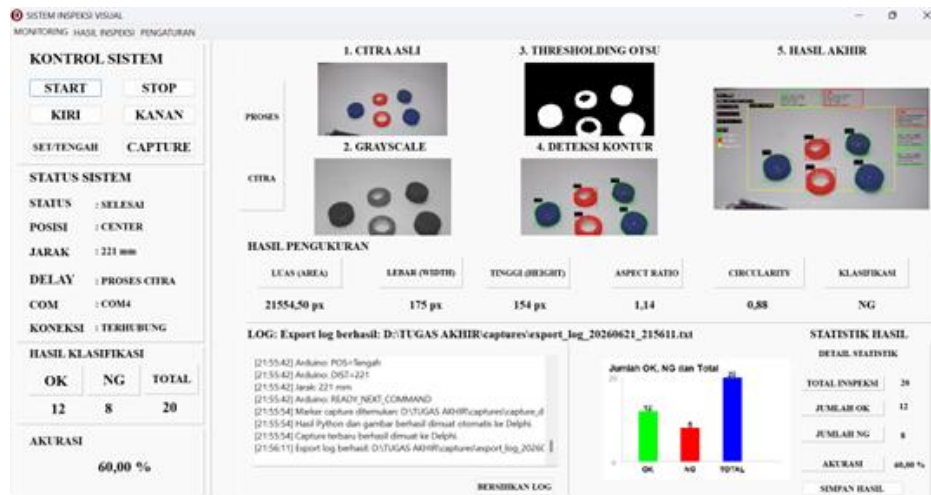
Pengujian yang dilakukan meliputi tampilan antarmuka aplikasi, fungsi tombol kontrol, komunikasi serial dengan mikrokontroler, serta kemampuan sistem dalam menampilkan data hasil inspeksi.

4.2.1 Tampilan Aplikasi Delphi

Aplikasi kontrol sistem inspeksi visual dirancang menggunakan Delphi sebagai antarmuka utama yang digunakan untuk mengendalikan proses inspeksi serta memantau kondisi sistem secara keseluruhan. Antarmuka ini dibuat dalam bentuk beberapa tab untuk memudahkan pengguna dalam mengakses fungsi-fungsi yang tersedia.

Secara umum, aplikasi terdiri dari tiga bagian utama, yaitu tab Monitoring, tab Hasil Inspeksi, dan tab Pengaturan. Setiap tab memiliki fungsi yang berbeda namun saling terintegrasi dalam mendukung proses inspeksi visual.

Tampilan utama aplikasi pada tab Monitoring ditunjukkan pada **gambar 4.19**. Pada tampilan ini, pengguna dapat mengontrol jalannya sistem inspeksi serta memantau kondisi sistem secara *real-time*.



Gambar 4. 19 Tampilan Monitoring Aplikasi Kontrol Delphi

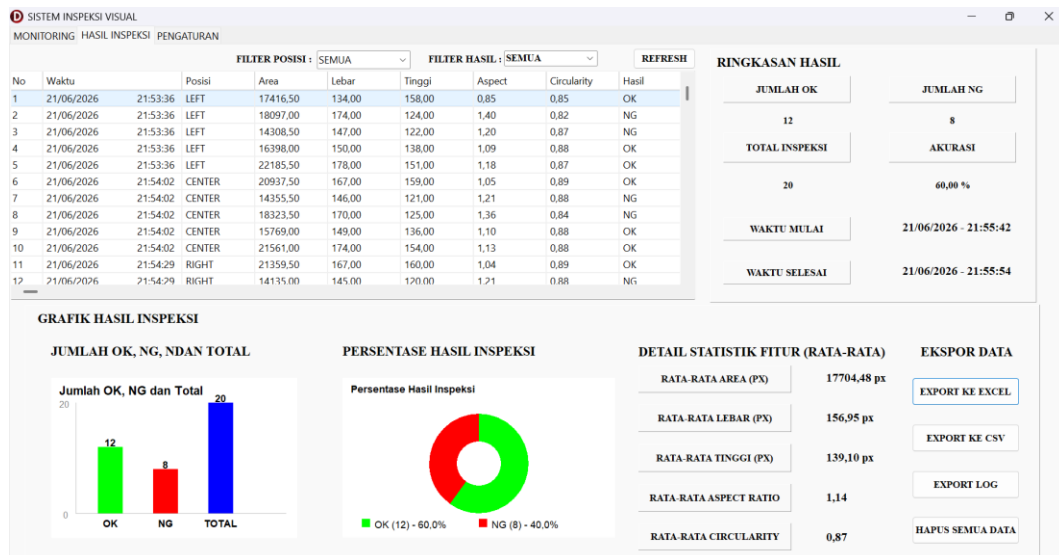
Tampilan tab Monitoring pada **gambar 4.19** merupakan bagian utama yang digunakan untuk mengendalikan dan memantau kerja sistem secara *real-time*. Pada tab ini, pengguna dapat menjalankan sistem inspeksi, mengatur pergerakan kamera, serta melihat status sistem selama proses pengujian berlangsung.

Pada bagian kiri terdapat panel kontrol sistem yang berisi tombol-tombol utama seperti Start, Stop, Kiri, Tengah, Kanan, dan Capture. Tombol Start digunakan untuk menjalankan proses inspeksi otomatis, sedangkan tombol Stop digunakan untuk menghentikan proses sistem. Tombol Kiri, Tengah, dan Kanan digunakan untuk menggerakkan kamera secara manual menuju posisi tertentu, sedangkan tombol Capture digunakan untuk mengambil citra objek secara manual.

Di bagian tengah terdapat area proses citra yang menampilkan tahapan pengolahan citra, mulai dari citra asli, grayscale, *thresholding* Otsu, deteksi kontur, hingga hasil akhir klasifikasi objek. Tampilan ini memberikan gambaran visual mengenai alur pengolahan citra yang dilakukan oleh sistem.

Selain itu, terdapat bagian status sistem yang menampilkan informasi seperti status sistem, posisi kamera, jarak objek, serta status koneksi dengan mikrokontroler. Informasi ini digunakan untuk memastikan bahwa sistem berada dalam kondisi normal selama proses inspeksi berlangsung.

Pada bagian bawah, ditampilkan hasil pengukuran dan statistik yang meliputi luas objek, dimensi objek, *aspect ratio*, *circularity*, serta hasil klasifikasi objek OK atau NG. Selain itu, aplikasi juga menampilkan jumlah objek yang terdeteksi dan data hasil inspeksi yang digunakan untuk memantau performa sistem. Selanjutnya tampilan tab Hasil Inspeksi ditunjukkan pada **gambar 4.20**.



Gambar 4. 20 Tampilan Hasil Inspeksi Aplikasi Kontrol Delphi

Tampilan tab Hasil Inspeksi pada **gambar 4.20** berfungsi untuk menampilkan data hasil inspeksi secara lebih rinci dalam bentuk tabel. Pada tab ini, data hasil inspeksi ditampilkan berdasarkan parameter hasil ekstraksi fitur objek, seperti area, lebar, tinggi, *aspect ratio*, *circularity*, dan hasil klasifikasi objek.

Data hasil inspeksi disusun dalam bentuk tabel agar pengguna lebih mudah membaca dan menganalisis hasil pengujian. Selain itu, terdapat bagian ringkasan hasil yang menampilkan jumlah objek OK, jumlah objek NG, total inspeksi, serta nilai akurasi sistem. Informasi tersebut digunakan untuk mengevaluasi hasil inspeksi secara keseluruhan.

Pada bagian bawah, ditampilkan grafik atau ringkasan statistik hasil inspeksi yang dapat membantu pengguna melihat kecenderungan hasil pengujian. Dengan adanya tab Hasil Inspeksi, pengguna dapat melakukan pemantauan dan

analisis hasil inspeksi secara lebih terstruktur dan sistematis. Selanjutnya, tampilan pengaturan sistem yang digunakan ditunjukkan pada **gambar 4.21**.

SISTEM INSPEKSI VISUAL
MONITORING HASIL INSPEKSI PENGATURAN

1. KOMUNIKASI

COM PORT	COM4	CONNECT DISCONNECT
BAUDRATE	9600	
PARITY	None	
DATA BITS	8	
STOP BITS	1	

2. MOTOR DAN GERAK

STEP KIRI	10000	JUMLAH STEP POSISI KIRI
STEP KANAN	10000	JUMLAH STEP POSISI KANAN
DELAY INSPEKSI	Menunggu Python Selesai	WAKTU JEDA INSPEKSI
KECEPATAN MOTOR	Normal	KECEPATAN PERGERAKAN MOTOR
MODE GERAK	Otomatis 3 Posisi	MODE INSPEKSI SISTEM

3. PENGOLAHAN CITRA

THRESHOLD METHOD	OTSU	METODE THRESHOLDING CITRA
MIN. AREA (PX)	1000 px	LUAS MINIMUM OBJEK VALID
CIRCULARITY MIN	0.70	BATAS CIRCULARITY MINIMUM (OK)
CIRCULARITY MAX	1.00	BATAS CIRCULARITY MAKSIMUM (OK)
ASPECT RATIO MAX	1.50	BATAS ASPECT RATIO MAKSIMUM
FILTER NOISE (PX)	3 px	UKURAN FILTER NOISE (MEDIAN BLUR)

4. PENYIMPANAN

FOLDER HASIL GAMBAR	D:\TUGAS AKHIR\captures\	BROWSE
FORMAT GAMBAR	JPG	FORMAT PENYIMPANAN HASIL
AUTO SAVE	☒ ON	SIMPAN GAMBAR OTOMATIS
NAMA FILE	Tanggal_Waktu_Posisi	FORMAT PENAMAAN FILE
SIMPAN LOG	☒ ON	SIMPAN LOG SISTEM KE FILE

SIMPAN PENGATURAN RESET KE DEFAULT

Gambar 4. 21 Tampilan Pengaturan Aplikasi Kontrol Delphi

Tampilan tab Pengaturan pada **gambar 4.21** digunakan untuk melihat parameter sistem yang digunakan sesuai kebutuhan pengujian. Pada bagian ini, terdapat pengaturan komunikasi serial yang digunakan untuk menghubungkan aplikasi Delphi dengan Arduino Mega 2560. Pengaturan tersebut diperlukan agar aplikasi dapat mengirim perintah dan menerima data dari mikrokontroler.

Selain itu, terdapat informasi pengaturan motor dan gerak yang digunakan untuk menyesuaikan pergerakan kamera, seperti jumlah langkah motor, arah gerak, dan delay inspeksi. Parameter ini berfungsi untuk menampilkan perpindahan kamera pada posisi kiri, tengah, dan kanan agar sesuai dengan kebutuhan sistem.

Pada bagian pengolahan citra, terdapat informasi parameter yang berkaitan dengan proses deteksi objek, seperti metode *thresholding*, batas area minimum objek, *circularity*, dan parameter pendukung lainnya. Pengaturan ini digunakan untuk menyesuaikan proses pengolahan citra agar hasil deteksi dan klasifikasi objek dapat berjalan lebih optimal.

Selain itu, terdapat juga pengaturan penyimpanan data, seperti lokasi penyimpanan hasil gambar, format file, serta opsi penyimpanan otomatis. Dengan

adanya tab Pengaturan, sistem menjadi lebih fleksibel karena pengguna dapat menyesuaikan parameter pengujian sesuai kebutuhan. Fungsi tampilan aplikasi delphi dimuat dalam bentuk sajian tabel untuk mengetahui hasil dari aplikasinya yang disajikan pada **tabel 4.8**.

Tabel 4. 8 Fungsi Tampilan Aplikasi Delphi

No	Bagian Tampilan	Fungsi Utama	Keterangan
1	Tab Monitoring	Mengontrol dan memantau proses inspeksi secara <i>real-time</i> .	Digunakan untuk menjalankan sistem, menggerakkan kamera, melakukan capture, dan melihat status sistem.
2	Tab Hasil Inspeksi	Menampilkan data hasil inspeksi objek.	Berisi tabel hasil fitur objek, hasil klasifikasi OK dan NG, total inspeksi, serta ringkasan hasil pengujian.
3	Tab Pengaturan	Mengatur parameter sistem.	Digunakan untuk mengatur komunikasi serial, parameter motor, parameter pengolahan citra, dan penyimpanan data.
4	Area Proses Citra	Menampilkan tahapan pengolahan citra.	Menampilkan citra asli, grayscale, <i>thresholding</i> Otsu, deteksi kontur, dan hasil akhir klasifikasi.
5	Panel Kontrol	Memberikan perintah ke sistem.	Berisi tombol Start, Stop, Kiri, Tengah, Kanan, dan Capture.

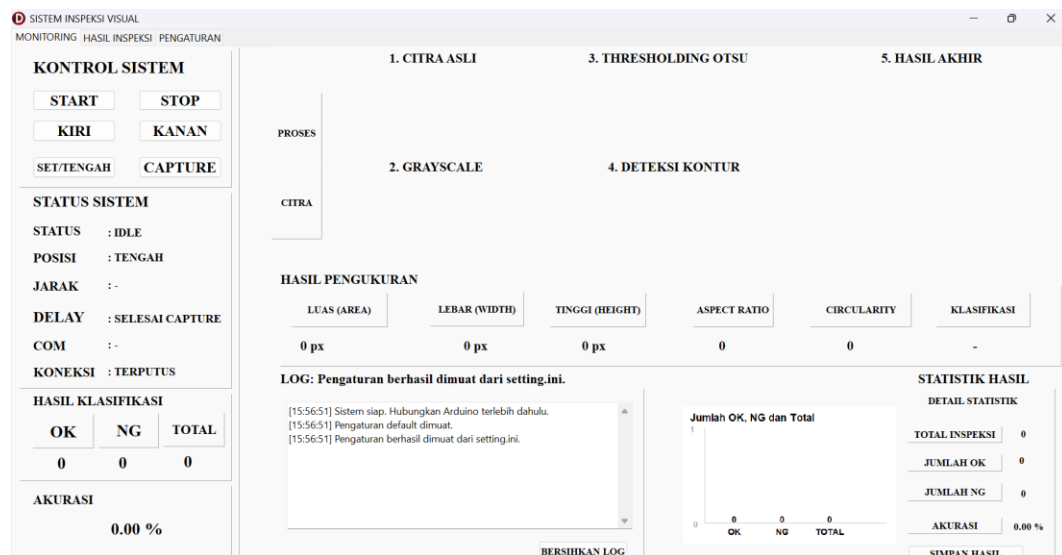
Secara keseluruhan, aplikasi Delphi berfungsi sebagai pusat kontrol dan monitoring sistem inspeksi visual. Aplikasi ini dapat mengirim perintah ke Arduino, menampilkan status sistem, memantau posisi kamera, menampilkan hasil

pengolahan citra, serta menyajikan data hasil klasifikasi objek. Dengan demikian, tampilan aplikasi Delphi dinilai mampu mendukung proses pengoperasian sistem inspeksi visual secara lebih mudah dan terstruktur.

4.2.2 Pengujian Fungsi Tombol Aplikasi Delphi

Pengujian fungsi tombol pada aplikasi Delphi dilakukan untuk memastikan bahwa setiap tombol kontrol dapat berfungsi dengan baik sesuai dengan perintah yang diberikan. Pengujian ini juga dilakukan untuk mengetahui apakah perintah yang dikirimkan dari aplikasi Delphi dapat diterima dan dijalankan oleh sistem melalui komunikasi serial dengan Arduino Mega 2560.

Tombol pada aplikasi Delphi digunakan untuk mengendalikan proses inspeksi, mengatur pergerakan kamera, melakukan pengambilan citra, serta menghentikan proses sistem apabila diperlukan. Tampilan pengujian tombol pada aplikasi Delphi ditunjukkan pada **gambar 4.22**.



Gambar 4. 22 Pengujian tombol sistem pada aplikasi Delphi saat proses Start

Berdasarkan **gambar 4.22**, aplikasi Delphi menyediakan beberapa tombol kontrol utama, yaitu Start, Stop, SETCENTER/Homing, Kiri, Tengah, Kanan, dan Capture. Setiap tombol memiliki fungsi tertentu dalam mengendalikan sistem inspeksi visual, baik untuk menjalankan proses otomatis, mengatur posisi kamera, maupun melakukan pengambilan citra objek.

Selanjutnya dilakukan pengujian terhadap masing-masing tombol untuk mengetahui apakah sistem dapat merespon perintah yang diberikan dengan baik. Hasil pengujian fungsi tombol ditunjukkan pada **tabel 4.9**.

Tabel 4. 9 Hasil Pengujian Fungsi Tombol Aplikasi Delphi

No	Tombol	Fungsi	Hasil Pengujian	Keterangan
1	Start	Memulai proses inspeksi otomatis	Berhasil	Sistem mulai berjalan dan proses inspeksi dapat dilakukan.
2	Stop	Menghentikan proses sistem	Berhasil	Proses sistem dapat dihentikan melalui aplikasi Delphi.
3	SETCENTER /Homing	Menetapkan atau mengembalikan posisi awal kamera	Berhasil	Posisi tengah berhasil digunakan sebagai titik acuan sistem.
4	Kiri	Menggerakkan kamera ke posisi kiri	Berhasil	Kamera dapat bergeser ke posisi kiri.
5	Tengah	Menggerakkan kamera ke posisi tengah	Berhasil	Kamera dapat kembali ke posisi tengah.
6	Kanan	Menggerakkan kamera ke posisi kanan	Berhasil	Kamera dapat bergeser ke posisi kanan.
7	Capture	Mengambil citra objek	Berhasil	Citra objek berhasil diambil dan dapat

digunakan untuk
proses pengolahan
citra.

Berdasarkan **tabel 4.9**, seluruh tombol pada aplikasi Delphi dapat berfungsi dengan baik sesuai dengan perintah yang diberikan. Tombol Start berhasil menjalankan proses inspeksi otomatis, sedangkan tombol Stop dapat menghentikan proses sistem saat dibutuhkan.

Tombol SETCENTER/Homing digunakan untuk menetapkan atau mengembalikan posisi kamera ke titik tengah sebagai posisi acuan awal sistem. Posisi ini penting agar sistem memiliki titik referensi sebelum kamera bergerak ke posisi kiri dan kanan. Sementara itu, tombol Kiri, Tengah, dan Kanan digunakan untuk mengatur posisi kamera secara manual sesuai kebutuhan pengujian.

Pada tombol Capture, sistem berhasil melakukan pengambilan citra objek pada area inspeksi. Citra yang diambil kemudian dapat digunakan pada tahap pengolahan citra berikutnya. Secara keseluruhan, hasil pengujian menunjukkan bahwa tombol kontrol pada aplikasi Delphi telah mampu menjalankan fungsi utama sistem dan dapat berkomunikasi dengan mikrokontroler melalui komunikasi serial.

4.3 Pengujian Pengolahan Citra

Pengujian pengolahan citra dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses pengambilan dan pengolahan citra objek geometris lingkaran menggunakan metode pengolahan citra digital berbasis *OpenCV*. Tahapan pengolahan citra pada sistem ini meliputi proses akuisisi citra, konversi grayscale, *thresholding* menggunakan metode Otsu, deteksi kontur, hingga proses ekstraksi fitur objek yang digunakan dalam proses klasifikasi objek.

Pengujian dilakukan menggunakan kamera USB sebagai media akuisisi citra yang dipasang pada mekanisme pergerakan kamera berbasis motor *stepper*. Kamera digunakan untuk menangkap citra objek secara *real-time* pada area inspeksi yang telah ditentukan. Selanjutnya citra diproses menggunakan program berbasis Python

OpenCV sehingga sistem mampu mengenali bentuk objek berdasarkan parameter geometris tertentu.

Pada tahap pengolahan citra, sistem melakukan beberapa proses secara berurutan dimulai dari pengambilan citra asli objek, proses konversi citra menjadi grayscale, proses segmentasi objek menggunakan *thresholding* Otsu, deteksi kontur objek, hingga menghasilkan data fitur geometris objek seperti luas area, dimensi objek, *aspect ratio*, dan *circularity*. Tahapan proses pengolahan citra dilakukan untuk mempermudah sistem dalam mengenali bentuk objek dan menentukan hasil klasifikasi objek secara otomatis.

Pengujian pengolahan citra dilakukan untuk memastikan setiap tahapan proses dapat berjalan dengan baik dan menghasilkan citra yang sesuai untuk proses inspeksi visual. Hasil pengujian pada setiap tahapan pengolahan citra dijelaskan pada subbab berikut.

4.3.1 Hasil Akuisisi Citra

Akuisisi citra merupakan tahap awal pada sistem inspeksi visual yang berfungsi untuk mengambil gambar objek menggunakan kamera USB sebagai input utama pada proses pengolahan citra. Pada tahap ini, kamera menangkap citra objek geometris lingkaran secara *real-time* pada area inspeksi yang telah ditentukan. Hasil citra yang diperoleh kemudian digunakan sebagai data masukan pada proses pengolahan citra selanjutnya.

Pengujian akuisisi citra dilakukan untuk memastikan kamera mampu menangkap objek dengan baik, menghasilkan citra yang cukup jelas, serta dapat menampilkan objek secara *real-time* pada sistem. Selain itu, pengujian ini juga bertujuan untuk memastikan posisi objek masih dapat terdeteksi dengan baik meskipun kamera bergerak pada beberapa posisi pengamatan.

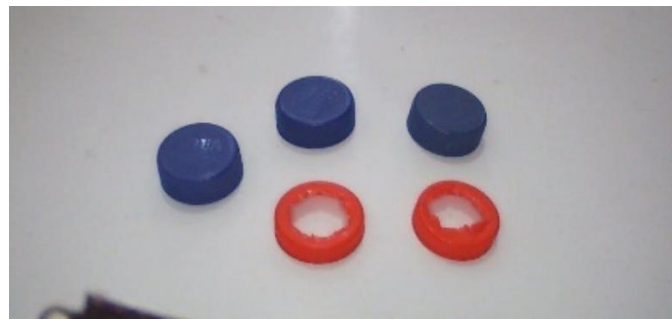
Metode pengujian dilakukan dengan menghubungkan kamera USB ke laptop kemudian menjalankan program berbasis Python *OpenCV* untuk menampilkan hasil tangkapan citra secara langsung. Kamera diarahkan pada objek geometris lingkaran yang berada pada area inspeksi, kemudian sistem melakukan proses

pengambilan gambar secara kontinu. Hasil pengujian akuisisi citra ditunjukkan pada **gambar 4.23** untuk posisi kiri.



Gambar 4. 23 Hasil Akuisisi Citra Pada Posisi Kamera Kiri

Selanjutnya, hasil akuisisi citra pada posisi kamera tengah ditunjukkan pada **gambar 4.24**.



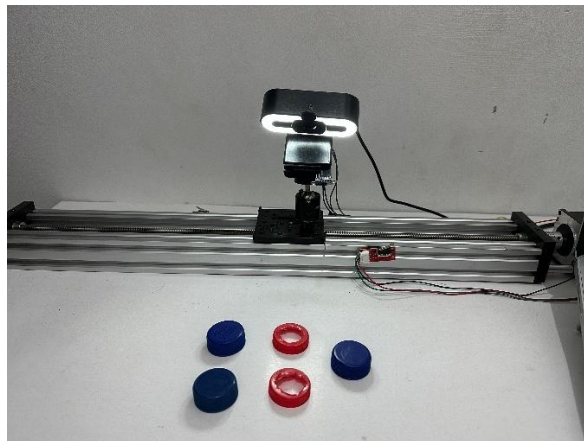
Gambar 4. 24 Hasil Akuisisi Citra Pada Posisi Kamera Tengah

Kemudian, hasil akuisisi citra pada posisi kamera kanan ditunjukkan pada **gambar 4.25**.



Gambar 4. 25 Hasil Akuisisi Citra Pada Posisi Kamera Kanan

Posisi kamera selama proses akuisisi citra ditunjukkan pada **gambar 4.26**.



Gambar 4. 26 Posisi Kamera Pada Proses Akuisisi Citra

Berdasarkan **gambar 4.23** hingga **gambar 4.25**, kamera USB berhasil menangkap citra objek dengan baik dan mampu menampilkan objek secara *real-time* pada sistem. Objek lingkaran dapat terlihat dengan cukup jelas pada area inspeksi sehingga citra yang dihasilkan dapat digunakan pada tahap pengolahan citra selanjutnya. Selain itu, sistem juga mampu mempertahankan kualitas tampilan citra selama proses pengambilan gambar berlangsung.

Untuk mengetahui hasil pengujian akuisisi citra secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter pengujian yang disajikan pada **tabel 4.10**.

Tabel 4. 10 Hasil Pengujian Akuisisi Citra

No	Posisi Kamera	Jumlah		Koneksi Kamera	Tampilan <i>Real-time</i>	Kualitas Citra	Keterangan
		Objek per Capture					
1	Kiri	5	Berhasil	Berhasil	Baik		Kamera berhasil mengambil citra objek pada posisi kiri. Objek masih terlihat jelas pada area inspeksi dan dapat digunakan

						untuk proses pengolahan citra.
						Kamera berhasil mengambil citra objek pada posisi tengah. Posisi ini menghasilkan tampilan objek yang lebih stabil karena kamera berada lebih sejajar terhadap area inspeksi.
2	Tengah	5	Berhasil	Berhasil	Baik	Kamera berhasil mengambil citra objek pada posisi kanan. Objek masih dapat ditampilkan pada area inspeksi meskipun terdapat sedikit perbedaan sudut pandang kamera.
3	Kanan	5	Berhasil	Berhasil	Baik	

Berdasarkan **tabel 4.10**, proses akuisisi citra dapat berjalan dengan baik pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Pada setiap posisi, kamera berhasil melakukan capture sebanyak lima kali dengan jumlah lima objek pada setiap capture. Kamera USB juga berhasil terhubung dengan sistem, menampilkan

citra secara *real-time*, serta menampilkan area inspeksi sebagai batas pengamatan objek.

Hasil pengujian menunjukkan bahwa citra yang diperoleh memiliki kualitas yang cukup baik untuk digunakan pada proses pengolahan citra berikutnya. Posisi tengah menghasilkan tampilan citra yang lebih stabil karena kamera berada pada posisi yang lebih sejajar terhadap objek. Sementara itu, posisi kiri dan kanan masih dapat digunakan untuk proses pengolahan citra meskipun terdapat perbedaan sudut pandang pengambilan gambar.

4.3.2 Hasil Grayscale

Proses grayscale merupakan tahap awal pengolahan citra yang dilakukan setelah proses akuisisi citra objek. Pada tahap ini, citra berwarna (RGB) dikonversi menjadi citra keabuan (grayscale) untuk menyederhanakan informasi citra sebelum dilakukan proses segmentasi dan deteksi objek. Konversi grayscale dilakukan dengan mengubah nilai warna pada setiap piksel menjadi tingkat intensitas abu-abu sehingga proses pengolahan citra dapat dilakukan dengan lebih mudah dan efisien.

Penggunaan citra grayscale dilakukan agar proses pengolahan citra menjadi lebih sederhana dan efisien. Dengan mengubah citra menjadi grayscale, sistem tidak lagi memproses tiga kanal warna RGB, tetapi hanya menggunakan satu kanal intensitas. Hal ini memudahkan sistem dalam membedakan objek dengan latar belakang berdasarkan perbedaan nilai intensitas cahaya.

Metode pengujian dilakukan dengan memproses citra hasil akuisisi menggunakan program Python *OpenCV*. Citra hasil capture dari kamera USB pada posisi kiri, tengah, dan kanan dikonversi menjadi citra grayscale. Hasil grayscale tersebut kemudian digunakan sebagai masukan untuk proses *thresholding* Otsu. Hasil konversi citra grayscale ditunjukkan pada **gambar 4.27** untuk posisi kamera kiri.



Gambar 4. 27 Hasil Konversi Citra Grayscale Pada Posisi Kamera Kiri

Selanjutnya, hasil konversi citra *grayscale* pada posisi kamera tengah ditunjukkan pada **gambar 4.28**.



Gambar 4. 28 Hasil Konversi Citra Grayscale Pada Posisi Kamera Tengah

Kemudian, hasil konversi citra *grayscale* pada posisi kamera kanan ditunjukkan pada **gambar 4.29**.



Gambar 4. 29 Hasil Konversi Citra Grayscale Pada Posisi Kamera Kanan

Berdasarkan **gambar 4.27** hingga **gambar 4.29**, citra objek berhasil dikonversi dari citra berwarna menjadi citra grayscale. Pada hasil grayscale, objek dan latar belakang terlihat dalam bentuk tingkat keabuan sehingga perbedaan

intensitas cahaya antar objek menjadi lebih jelas. Proses ini membantu sistem dalam melakukan segmentasi objek pada tahap pengolahan citra berikutnya.

Selain itu, hasil grayscale menunjukkan bahwa bentuk objek masih dapat dikenali dengan baik meskipun informasi warna telah dihilangkan. Hal ini penting untuk memastikan proses *thresholding* dan deteksi kontur dapat dilakukan secara optimal tanpa dipengaruhi variasi warna pada objek.

Untuk mengetahui hasil pengujian grayscale secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter yang disajikan pada **tabel 4.11**.

Tabel 4. 11 Hasil Pengujian Grayscale

No	Posisi Kamera	Jumlah Objek per Capture	Proses Grayscale	Kejelasan Bentuk Objek	Perbedaan Objek dan Latar Belakang	Keterangan
1	Kiri	5	Berhasil	Baik	Jelas	Citra pada posisi kiri berhasil dikonversi menjadi grayscale dan objek masih dapat dikenali dengan baik.
2	Tengah	5	Berhasil	Baik	Jelas	Citra pada posisi tengah menghasilkan tampilan grayscale yang stabil

						dan objek terlihat jelas pada area inspeksi.
						Citra pada posisi kanan berhasil dikonversi menjadi grayscale, meskipun terdapat sedikit perbedaan intensitas akibat sudut pengambilan gambar.
3	Kanan	5	Berhasil	Baik	Cukup jelas	

Berdasarkan **tabel 4.11**, proses grayscale berhasil dilakukan pada seluruh posisi kamera, yaitu kiri, tengah, dan kanan. Setiap posisi kamera memiliki lima data capture dengan jumlah lima objek pada setiap capture. Hasil grayscale menunjukkan bahwa bentuk objek masih dapat dikenali dengan baik, sehingga informasi bentuk objek tidak hilang meskipun citra telah diubah dari citra berwarna menjadi citra keabuan.

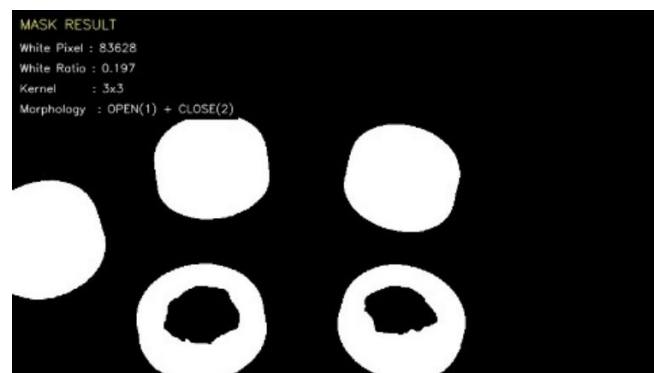
Perbedaan intensitas antara objek dan latar belakang juga masih terlihat pada citra grayscale. Hal ini menunjukkan bahwa citra grayscale telah memenuhi kebutuhan sistem sebagai masukan untuk proses segmentasi citra. Dengan demikian, hasil grayscale dapat digunakan pada tahap *thresholding* Otsu untuk memisahkan objek dari latar belakang sebelum dilakukan proses deteksi kontur.

4.3.3 Hasil *thresholding* Otsu

Thresholding Otsu merupakan proses segmentasi citra yang digunakan untuk memisahkan objek dengan latar belakang berdasarkan nilai intensitas piksel secara otomatis. Metode Otsu bekerja dengan menentukan nilai threshold optimal sehingga objek dapat dipisahkan dari background dalam bentuk citra biner hitam dan putih. Pada sistem inspeksi visual ini, metode *thresholding* Otsu digunakan untuk mempermudah proses deteksi bentuk objek sebelum dilakukan proses deteksi kontur.

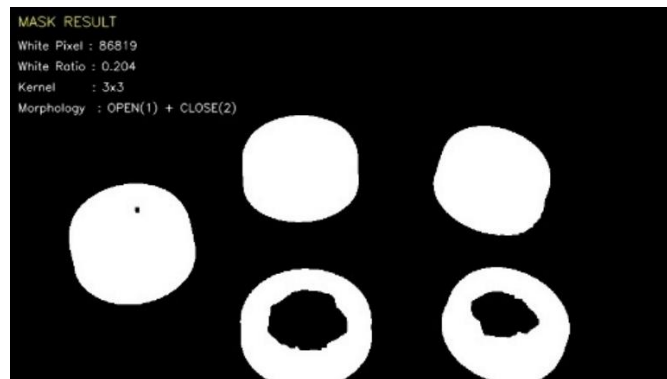
Proses *thresholding* dilakukan setelah citra berhasil dikonversi menjadi grayscale. Pada tahap ini, sistem memproses citra keabuan dan menentukan nilai threshold optimal menggunakan metode Otsu. Hasil dari proses ini berupa citra biner, yaitu citra yang hanya memiliki dua nilai piksel, hitam dan putih. Objek ditampilkan sebagai area putih, sedangkan latar belakang ditampilkan sebagai area hitam.

Metode pengujian dilakukan dengan memproses citra grayscale menggunakan program berbasis Python *OpenCV*. Sistem secara otomatis melakukan segmentasi objek menggunakan metode *thresholding* Otsu kemudian menampilkan hasil citra biner pada sistem. Hasil proses *thresholding* Otsu ditunjukkan pada **gambar 4.30** untuk posisi kiri.



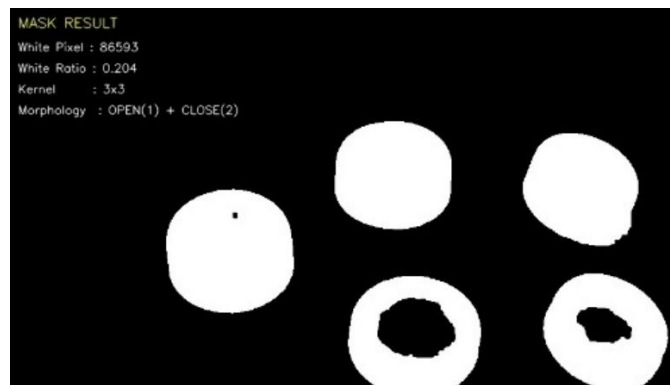
Gambar 4. 30 Hasil *thresholding* Otsu Pada Posisi Kamera Kiri

Selanjutnya, hasil *thresholding* Otsu pada posisi kamera tengah ditunjukkan pada **gambar 4.31**.



Gambar 4. 31 Hasil *thresholding* Otsu Pada Posisi Kamera Tengah

Kemudian, hasil *thresholding* Otsu pada posisi kamera kanan ditunjukkan pada **gambar 4.32**.



Gambar 4. 32 Hasil *thresholding* Otsu Pada Posisi Kamera Kanan

Berdasarkan **gambar 4.30** hingga **gambar 4.32**, metode *thresholding* Otsu berhasil memisahkan objek dengan latar belakang sehingga objek terlihat lebih jelas dalam bentuk citra biner. Objek ditampilkan berwarna putih sedangkan background ditampilkan berwarna hitam. Hasil segmentasi menunjukkan bahwa bentuk objek dapat dikenali dengan baik sehingga mempermudah proses deteksi kontur pada tahap berikutnya.

Selain itu, proses *thresholding* juga mampu mengurangi pengaruh warna dan detail yang tidak diperlukan pada citra asli. Dengan adanya proses segmentasi ini, sistem dapat lebih fokus dalam mendeteksi area objek utama yang akan digunakan dalam proses ekstraksi fitur geometris dan klasifikasi objek.

Untuk mengetahui hasil pengujian *thresholding* Otsu secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter yang disajikan pada **tabel 4.12**.

Tabel 4. 12 Hasil Pengujian *Thresholding* Otsu

No	Posisi Kamera	Jumlah Objek per Capture	Proses <i>Thresholding</i> Otsu	Pemisahan Objek dan Latar Belakang	Kejelasan Bentuk Objek	Keterangan
1	Kiri	5	Berhasil	Berhasil	Baik	Sistem berhasil menghasilkan citra biner pada posisi kiri. Objek dapat dipisahkan dari latar belakang dan dapat digunakan untuk proses deteksi kontur.
2	Tengah	5	Berhasil	Berhasil	Baik	Sistem berhasil melakukan segmentasi pada posisi tengah. Bentuk objek terlihat lebih stabil karena posisi kamera lebih sejajar terhadap area inspeksi.
3	Kanan	5	Berhasil	Berhasil	Cukup baik	Sistem berhasil menghasilkan

citra biner pada
posisi kanan,
meskipun
terdapat sedikit
perbedaan hasil
segmentasi
akibat sudut
pengambilan
gambar.

Berdasarkan **tabel 4.12**, proses *thresholding* Otsu berhasil dilakukan pada seluruh posisi kamera, yaitu kiri, tengah, dan kanan. Pada setiap posisi kamera, sistem melakukan pengujian terhadap lima data capture dengan lima objek pada setiap capture. Hasil *thresholding* menunjukkan bahwa objek dapat dipisahkan dari latar belakang sehingga bentuk objek dapat dikenali dalam bentuk citra biner.

Hasil segmentasi pada posisi tengah terlihat lebih stabil karena kamera berada lebih sejajar terhadap area inspeksi. Sementara itu, posisi kiri dan kanan masih dapat menghasilkan citra biner yang sesuai untuk proses deteksi kontur, meskipun terdapat sedikit perbedaan bentuk segmentasi akibat sudut pengambilan gambar. Selain menghasilkan citra biner, sistem juga menampilkan informasi pendukung seperti jumlah white pixel, white ratio, ukuran kernel, serta proses morphology yang digunakan untuk memperbaiki hasil segmentasi objek. Proses morphology berupa operasi *opening* dan *closing* digunakan untuk mengurangi noise dan memperjelas bentuk objek hasil *thresholding*. Dengan demikian, metode *thresholding* Otsu dinilai sesuai digunakan sebagai tahap segmentasi pada sistem inspeksi visual objek geometris berbentuk lingkaran.

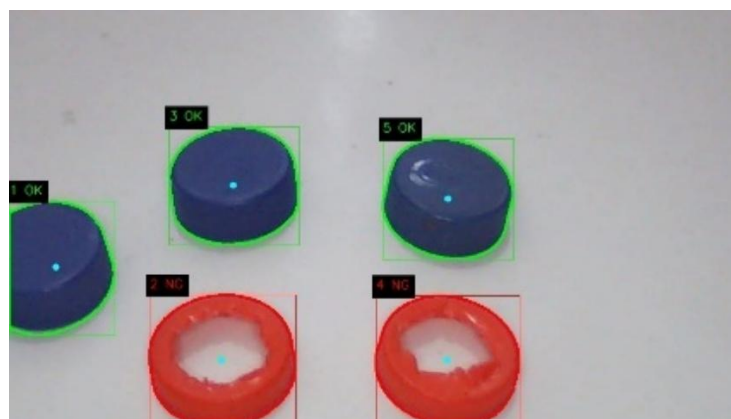
4.3.4 Hasil Deteksi Kontur

Deteksi kontur merupakan proses pengolahan citra yang digunakan untuk mengenali bentuk dan batas tepi objek berdasarkan hasil segmentasi citra sebelumnya. Pada sistem inspeksi visual ini, proses deteksi kontur digunakan untuk

mengidentifikasi objek geometris lingkaran serta menentukan batas area objek yang akan digunakan dalam proses ekstraksi fitur dan klasifikasi objek.

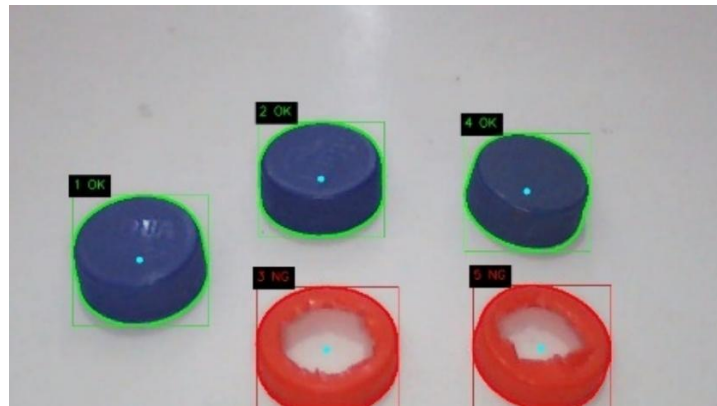
Proses deteksi kontur dilakukan setelah citra berhasil melalui tahap *thresholding* Otsu. Pada tahap ini, sistem mencari area objek yang memiliki perbedaan intensitas dengan latar belakang kemudian membentuk garis kontur pada bagian tepi objek. Selain mendeteksi bentuk objek, sistem juga menampilkan informasi posisi objek, titik tengah objek (*centroid*), bounding box, serta hasil klasifikasi objek berdasarkan parameter geometris yang telah ditentukan.

Metode pengujian dilakukan menggunakan program berbasis Python *OpenCV* dengan memanfaatkan fungsi deteksi kontur pada citra hasil *thresholding*. Sistem kemudian menampilkan hasil deteksi berupa garis kontur objek, area pembatas objek, serta informasi hasil klasifikasi objek secara *real-time*. Hasil deteksi kontur ditunjukkan pada **gambar 4.33** hingga **gambar 4.35**.



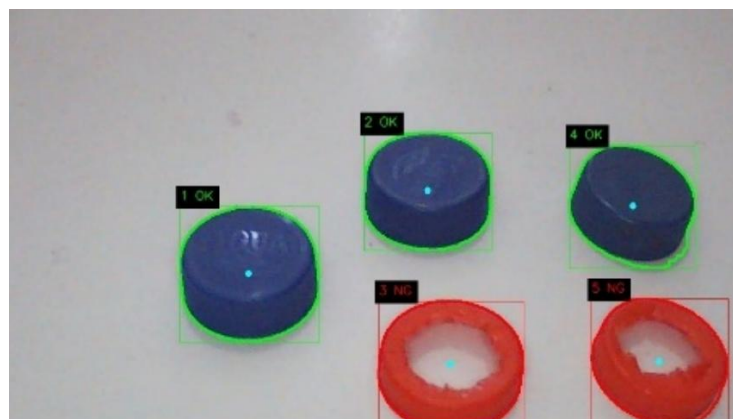
Gambar 4. 33 Hasil Deteksi Kontur Pada Posisi Kamera Kiri

Selanjutnya, hasil deteksi kontur pada posisi kamera tengah ditunjukkan pada **gambar 4.34**.



Gambar 4. 34 Hasil Deteksi Kontur Pada Posisi Kamera Tengah

Kemudian, hasil deteksi kontur pada posisi kamera kanan ditunjukkan pada **gambar 4.35**.



Gambar 4. 35 Hasil Deteksi Kontur Pada Posisi Kamera Kanan

Berdasarkan **gambar 4.33** hingga **gambar 4.35**, sistem berhasil mendeteksi kontur pada objek yang berada pada area inspeksi. Objek dengan bentuk lingkaran normal ditandai dengan garis kontur dan bounding box berwarna hijau serta diklasifikasikan sebagai objek OK. Sementara itu, objek yang memiliki lubang atau bentuk tidak sesuai ditandai dengan warna merah dan diklasifikasikan sebagai objek NG.

Selain itu, sistem juga berhasil menampilkan titik pusat objek (*centroid*) yang digunakan untuk membantu proses identifikasi posisi objek pada citra. Hasil deteksi menunjukkan bahwa proses kontur mampu membedakan objek normal dan objek cacat berdasarkan bentuk geometris objek yang terdeteksi.

Untuk mengetahui hasil pengujian deteksi kontur secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter yang disajikan pada **tabel 4.13**.

Tabel 4. 13 Hasil Pengujian Deteksi Kontur

No	Posisi Kamera	Jumlah Objek per Capture	Proses Deteksi Kontur	Deteksi <i>Centroid</i>	Bounding Box	Keterangan
1	Kiri	5	Berhasil	Berhasil	Berhasil	Sistem berhasil mendeteksi kontur objek pada posisi kiri. Garis kontur, <i>centroid</i> , dan bounding box dapat ditampilkan pada area inspeksi.
2	Tengah	5	Berhasil	Berhasil	Berhasil	Sistem berhasil mendeteksi kontur objek pada posisi tengah. Hasil deteksi terlihat lebih stabil karena kamera berada lebih sejajar terhadap area inspeksi.
3	Kanan	5	Berhasil	Berhasil	Berhasil	Sistem berhasil mendeteksi kontur objek pada posisi kanan, meskipun terdapat sedikit perbedaan tampilan akibat sudut pengambilan gambar.

Berdasarkan **tabel 4.13**, proses deteksi kontur berhasil dilakukan pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Pada setiap posisi kamera, sistem melakukan lima kali capture dengan jumlah lima objek pada setiap capture. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan garis kontur, *centroid*, dan bounding box pada objek yang berada di area inspeksi.

Proses deteksi kontur juga mampu membantu sistem dalam membedakan objek OK dan NG berdasarkan bentuk geometris objek yang terdeteksi. Hasil kontur yang diperoleh kemudian digunakan sebagai dasar dalam proses ekstraksi fitur, seperti area, *aspect ratio*, *circularity*, *solidity*, *extent*, dan *hole ratio*. Dengan demikian, proses deteksi kontur dinilai berhasil mendukung tahap ekstraksi fitur dan klasifikasi objek pada sistem inspeksi visual.

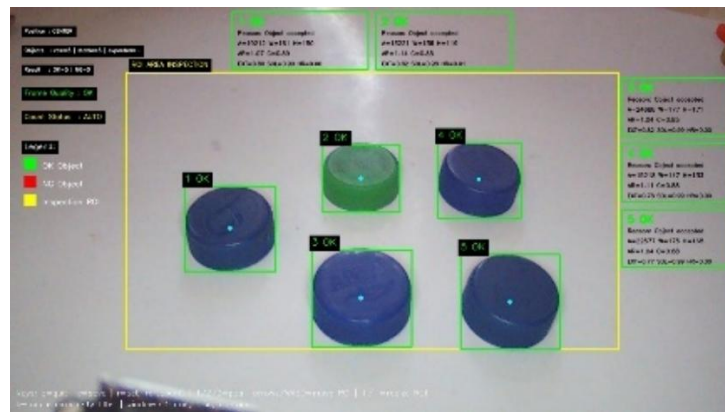
4.3.5 Hasil Pengolahan Citra Keseluruhan

Hasil pengolahan citra merupakan tahap akhir pada proses pengolahan citra yang menggabungkan seluruh tahapan sebelumnya, mulai dari akuisisi citra, konversi grayscale, *thresholding* Otsu, deteksi kontur, hingga proses klasifikasi objek. Pada tahap ini, sistem menampilkan hasil identifikasi objek secara lengkap beserta informasi parameter geometris objek yang digunakan dalam proses inspeksi visual.

Proses pengolahan citra dilakukan secara otomatis menggunakan program berbasis Python *OpenCV*. Sistem melakukan deteksi objek pada area inspeksi (Region of Interest/ROI), kemudian menghitung beberapa parameter geometris seperti luas area objek (*Area*), lebar objek (*Width*), tinggi objek (*Height*), *aspect ratio* (AR), *circularity* (C), *solidity* (SOL), *extent* (EXT), dan *hole ratio* (HR). Parameter tersebut digunakan sebagai dasar dalam menentukan klasifikasi objek apakah termasuk kategori OK atau NG.

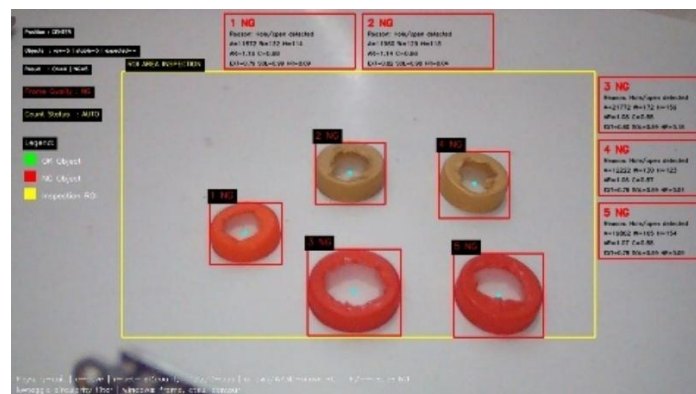
Metode pengujian dilakukan dengan menempatkan beberapa objek pada area inspeksi, kemudian sistem melakukan proses pengolahan citra secara *real-time*. Sistem selanjutnya menampilkan hasil deteksi objek, bounding box, nilai parameter geometris, jumlah objek terdeteksi, jumlah objek OK dan NG, serta hasil klasifikasi

objek pada layar monitoring. Hasil pengolahan citra keseluruhan ditunjukkan pada **gambar 4.36** sampai **gambar 4.40**.



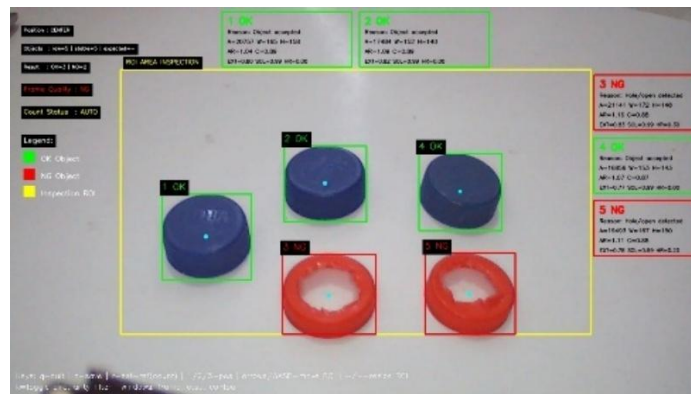
Gambar 4. 36 Hasil pengolahan Citra Keseluruhan Pada Siklus 1 Dengan Kondisi Semua Objek OK

Selanjutnya, hasil pengolahan citra keseluruhan pada siklus 2 dengan kondisi semua objek NG ditunjukkan pada **gambar 4.37**.



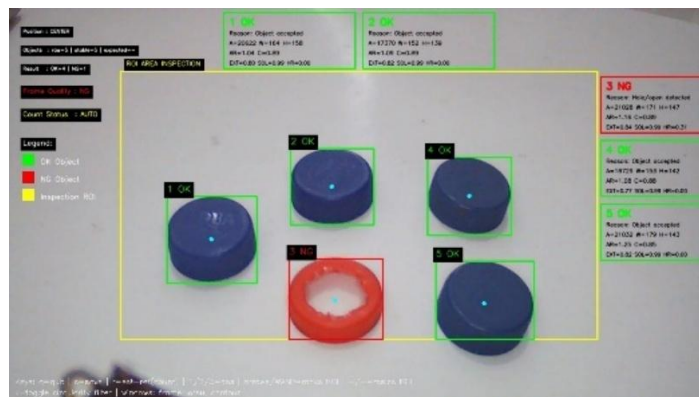
Gambar 4. 37 Hasil Pengolahan Citra Keseluruhan Pada Siklus 2 Dengan Kondisi Semua Objek NG

Kemudian, hasil pengolahan citra keseluruhan pada siklus 3 dengan kondisi objek campuran, yaitu 3 objek OK dan 2 objek NG, ditunjukkan pada **gambar 4.38**.



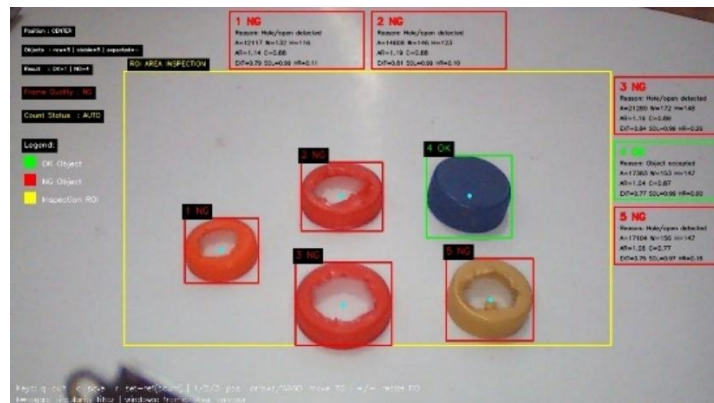
Gambar 4. 38 Hasil Pengolahan Citra Keseluruhan Pada Siklus 3 Dengan Kondisi Objek Campuran 3 OK Dan 2 NG

Hasil pengolahan citra keseluruhan pada siklus 4 dengan kondisi mayoritas objek OK ditunjukkan pada **gambar 4.39**.



Gambar 4. 39 Hasil Pengolahan Citra Keseluruhan Pada Siklus 4 Dengan Kondisi Mayoritas Objek OK

Selanjutnya, hasil pengolahan citra keseluruhan pada siklus 5 dengan kondisi mayoritas objek NG ditunjukkan pada **gambar 4.40**.



Gambar 4. 40 Hasil Pengolahan Citra Keseluruhan Pada Siklus 5 Dengan Kondisi Mayoritas Objek NG

Berdasarkan hasil pengolahan citra keseluruhan, sistem berhasil melakukan deteksi objek yang berada pada area inspeksi. Objek dengan bentuk geometris yang sesuai ditandai menggunakan bounding box berwarna hijau dan diklasifikasikan sebagai objek OK, sedangkan objek yang memiliki lubang atau bentuk tidak sesuai ditandai menggunakan bounding box berwarna merah dan diklasifikasikan sebagai objek NG.

Selain menampilkan hasil klasifikasi objek, sistem juga berhasil menampilkan berbagai parameter hasil ekstraksi fitur objek seperti area, dimensi objek, *aspect ratio*, *circularity*, *solidity*, *extent*, dan *hole ratio* secara *real-time*. Sistem juga menampilkan jumlah objek yang terdeteksi, jumlah objek OK dan NG, status kualitas frame, serta area inspeksi (ROI) yang digunakan selama proses pengolahan citra berlangsung.

Hasil pengolahan citra menunjukkan bahwa sistem mampu melakukan proses identifikasi objek secara otomatis dengan baik. Objek yang memiliki bentuk mendekati lingkaran sempurna dapat dikenali sebagai objek OK, sedangkan objek yang memiliki cacat atau lubang dapat dideteksi sebagai objek NG. Dengan demikian, proses pengolahan citra pada sistem inspeksi visual dinilai mampu mendukung proses deteksi dan klasifikasi objek geometris lingkaran secara *real-time*.

Untuk mengetahui hasil pengolahan citra secara lebih rinci, dilakukan pengamatan terhadap beberapa parameter yang disajikan pada **tabel 4.14**.

Tabel 4. 14 Hasil Pengolahan Citra Keseluruhan

No	Siklus	Kondisi Objek	Jumlah Objek	Objek Terdeteksi	Prediksi OK	Prediksi NG	Keterangan
1	1	Semua OK	15	15	14	1	Sistem berhasil mengolah citra pada tiga posisi kamera. Terdapat satu objek OK yang terbaca sebagai NG.
2	2	Semua NG	15	14	0	14	Sistem berhasil mengolah citra objek NG, namun terdapat satu objek yang tidak terdeteksi pada salah satu posisi kamera.

3	3	Campuran OK dan NG	15	15	9	6	Sistem berhasil mengolah citra objek campuran dan menampilka n hasil klasifikasi OK dan NG.
4	4	Mayoritas OK	15	15	12	3	Sistem berhasil mengolah citra dengan kondisi mayoritas objek OK.
5	5	Mayoritas NG	15	15	4	11	Sistem berhasil mengolah citra dengan kondisi mayoritas objek NG. Terdapat satu objek NG yang terbaca sebagai OK.

Berdasarkan **tabel 4.14**, sistem mampu melakukan seluruh tahapan pengolahan citra secara otomatis mulai dari akuisisi citra hingga proses klasifikasi objek. Pada setiap siklus, sistem melakukan pengambilan citra pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Setiap capture berisi lima objek, sehingga setiap siklus memiliki 15 data objek.

Hasil pengolahan citra menunjukkan bahwa sistem dapat menampilkan objek yang terdeteksi, bounding box, parameter geometris, serta hasil klasifikasi OK dan NG. Pada sebagian besar pengujian, objek dapat terdeteksi dengan baik dan hasil klasifikasi dapat ditampilkan secara *real-time*. Namun, masih terdapat beberapa kondisi yang menyebabkan kesalahan hasil, seperti objek OK terbaca sebagai NG, objek NG terbaca sebagai OK, dan satu objek NG tidak terdeteksi pada salah satu pengujian.

Secara keseluruhan, hasil pengolahan citra menunjukkan bahwa sistem mampu mendukung proses inspeksi visual objek geometris berbentuk lingkaran. Tahapan akuisisi citra, grayscale, *thresholding* Otsu, deteksi kontur, ekstraksi fitur, dan klasifikasi objek dapat berjalan secara berurutan sehingga sistem dapat membedakan objek OK dan NG berdasarkan parameter geometris yang dihasilkan.

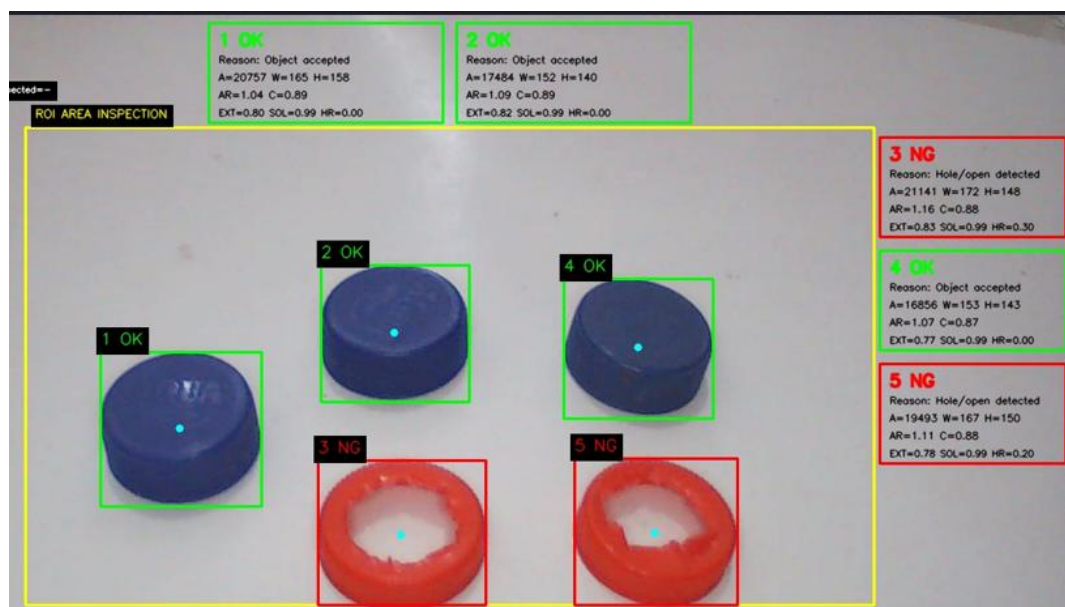
4.4 Pengujian Ekstraksi Fitur Objek

Pengujian ekstraksi fitur objek dilakukan untuk mengetahui kemampuan sistem dalam memperoleh parameter geometris objek dari hasil proses pengolahan citra. Proses ekstraksi fitur merupakan tahapan lanjutan setelah sistem berhasil melakukan *thresholding* Otsu dan deteksi kontur objek. Pada tahap ini, sistem melakukan pengambilan informasi geometris objek yang nantinya digunakan sebagai dasar dalam proses klasifikasi objek OK dan NG.

Fitur geometris yang diamati pada penelitian ini meliputi luas area objek (Area), lebar objek (*Width*), tinggi objek (*Height*), *aspect ratio* (AR), dan *circularity* (C). Selain parameter utama tersebut, sistem juga menampilkan parameter pendukung seperti *solidity* (SOL), *extent* (EXT), dan *hole ratio* (HR) untuk membantu proses identifikasi bentuk objek secara lebih detail.

Luas area objek digunakan untuk mengetahui ukuran area objek yang terdeteksi pada citra. Parameter *width* dan *height* digunakan untuk mengetahui dimensi objek berdasarkan bounding box hasil deteksi kontur. *Aspect ratio* digunakan untuk mengetahui perbandingan antara lebar dan tinggi objek, sedangkan *circularity* digunakan untuk mengetahui tingkat kemiripan bentuk objek terhadap lingkaran. Nilai *circularity* yang mendekati 1 menunjukkan bahwa objek memiliki bentuk yang semakin menyerupai lingkaran. Parameter *solidity* digunakan untuk melihat tingkat kepadatan bentuk objek, *extent* digunakan untuk mengetahui perbandingan area objek terhadap area bounding box, sedangkan *hole ratio* digunakan untuk mendeteksi adanya lubang atau bagian terbuka pada objek.

Metode pengujian dilakukan dengan menempatkan beberapa objek geometris pada area inspeksi kemudian sistem melakukan proses pengolahan citra secara otomatis menggunakan program berbasis Python *OpenCV*. Setelah proses deteksi kontur selesai dilakukan, sistem secara otomatis menghitung parameter geometris dari masing-masing objek yang terdeteksi kemudian menampilkan hasil ekstraksi fitur pada layar monitoring secara *real-time*. Hasil ekstraksi fitur objek ditunjukkan pada **gambar 4.41**.



Gambar 4. 41 Hasil Ekstraksi Fitur Objek Pada Sistem Inspeksi Visual

Selanjutnya, detail hasil ekstraksi fitur objek pada sistem inspeksi visual ditunjukkan pada **gambar 4.42**.

1 OK Reason: Object accepted A=20757 W=165 H=158 AR=1.04 C=0.89 EXT=0.80 SOL=0.99 HR=0.00	2 OK Reason: Object accepted A=17484 W=152 H=140 AR=1.09 C=0.89 EXT=0.82 SOL=0.99 HR=0.00
--	--

Gambar 4. 42 Detail Hasil Ekstraksi Fitur Objek Pada Sistem Inspeksi Visual

Data hasil ekstraksi fitur objek ditampilkan pada **gambar 4.43**.

3 NG Reason: Hole/open detected A=21141 W=172 H=148 AR=1.16 C=0.88 EXT=0.83 SOL=0.99 HR=0.30	4 OK Reason: Object accepted A=16856 W=153 H=143 AR=1.07 C=0.87 EXT=0.77 SOL=0.99 HR=0.00	5 NG Reason: Hole/open detected A=19493 W=167 H=150 AR=1.11 C=0.88 EXT=0.78 SOL=0.99 HR=0.20
---	--	---

Gambar 4. 43 Tampilan Data Hasil Ekstraksi Fitur Objek

Berdasarkan **gambar 4.41**, sistem berhasil menampilkan hasil ekstraksi fitur dari setiap objek yang terdeteksi pada area inspeksi. Sistem menampilkan parameter geometris objek seperti luas area, lebar objek, tinggi objek, *aspect ratio*, *circularity*, *solidity*, *extent*, serta *hole ratio* secara otomatis. Selain itu, sistem juga berhasil memberikan hasil klasifikasi objek berdasarkan parameter geometris yang diperoleh.

Objek yang memiliki bentuk mendekati lingkaran sempurna ditandai dengan bounding box berwarna hijau dan diklasifikasikan sebagai objek OK. Sementara

itu, objek yang memiliki bentuk tidak sesuai atau memiliki lubang pada bagian tengah objek ditandai dengan bounding box berwarna merah dan diklasifikasikan sebagai objek NG.

Untuk mengetahui hasil ekstraksi fitur objek secara lebih rinci, dilakukan pengamatan terhadap parameter geometris objek. Data yang ditampilkan pada **tabel 4.15** merupakan contoh hasil ekstraksi fitur pada satu siklus pengujian yang terdiri dari tiga posisi kamera, yaitu kiri, tengah, dan kanan. Setiap posisi kamera memiliki lima objek uji, sehingga jumlah data pada tabel ini adalah 15 data objek.

Tabel 4. 15 Hasil Ekstraksi Fitur Objek pada Siklus Pengujian Siklus 3 Objek 3 OK 2 NG

No	Posisi Kamera	Area	Lebar	Tinggi	Aspect	Circularity	Hasil Sistem	Keterangan
1	Kiri	16346,5	125	159	0,79	0,83	OK	Sesuai
2	Kiri	21336	173	152	1,14	0,89	NG	Sesuai
3	Kiri	17392	154	140	1,1	0,89	OK	Sesuai
4	Kiri	20382	170	151	1,13	0,88	NG	Sesuai
5	Kiri	17411	155	143	1,08	0,88	OK	Sesuai
6	Tengah	20757	165	158	1,04	0,89	OK	Sesuai
7	Tengah	17484,5	152	140	1,09	0,89	OK	Sesuai
8	Tengah	21141,5	172	148	1,16	0,88	NG	Sesuai
9	Tengah	16856,5	153	143	1,07	0,87	OK	Sesuai
10	Tengah	19493	167	150	1,11	0,88	NG	Sesuai
11	Kanan	21236	165	160	1,03	0,89	OK	Sesuai
12	Kanan	17328,5	151	139	1,09	0,89	OK	Sesuai
13	Kanan	20538,5	172	144	1,19	0,88	NG	Sesuai

14	Kanan	16395	149	145	1,03	0,85	OK	Sesuai
15	Kanan	18234	162	147	1,1	0,87	NG	Sesuai

Berdasarkan **tabel 4.15**, sistem berhasil melakukan ekstraksi fitur geometris objek pada siklus pengujian ke-3. Setiap objek yang terdeteksi menghasilkan nilai area, lebar, tinggi, *aspect ratio*, dan *circularity*. Nilai-nilai tersebut menunjukkan karakteristik bentuk objek yang digunakan sebagai dasar dalam proses klasifikasi objek OK dan NG.

Pada tabel tersebut, objek yang diklasifikasikan sebagai OK memiliki nilai *aspect ratio* yang masih berada dalam rentang parameter sistem serta nilai *circularity* yang mendekati bentuk lingkaran. Sementara itu, objek yang diklasifikasikan sebagai NG tetap dapat terdeteksi dan diekstraksi fiturnya, tetapi memiliki karakteristik tertentu yang menunjukkan ketidaksesuaian terhadap parameter sistem, seperti adanya lubang atau bentuk yang tidak sesuai.

Hasil pengujian pada siklus ke-3 menunjukkan bahwa sistem mampu melakukan ekstraksi fitur pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Setiap posisi kamera menghasilkan lima data objek, sehingga hasil ekstraksi fitur dapat digunakan untuk melihat perbedaan karakteristik objek pada setiap posisi pengamatan.

Selain tabel detail hasil ekstraksi fitur, dilakukan juga rekapitulasi hasil ekstraksi fitur pada seluruh siklus pengujian. Rekapitulasi ini digunakan untuk mengetahui jumlah objek yang berhasil terdeteksi dan jumlah objek yang diklasifikasikan sebagai OK maupun NG pada setiap posisi kamera. Hasil rekapitulasi ekstraksi fitur objek ditunjukkan pada **tabel 4.16**.

Tabel 4. 16 Rekapitulasi Hasil Ekstraksi Fitur Objek

No	Siklus	Posisi Kamera	Kondisi Objek	Jumlah Objek	Objek Terdeteksi	Objek OK	Objek NG	Status Ekstraksi	Keterangan
1	1	Kiri	Objek 5 OK	5	5	5	0	Berhasil	Seluruh fitur berhasil diekstraksi.
2	1	Tengah	Objek 5 OK	5	5	5	0	Berhasil	Seluruh fitur berhasil diekstraksi.
3	1	Kanan	Objek 5 OK	5	5	4	1	Tidak Berhasil	Sistem mengalami salah detect
4	2	Kiri	Objek 5 NG	5	5	0	5	Berhasil	Objek cacat berhasil diekstraksi fiturnya.
5	2	Tengah	Objek 5 NG	5	5	0	5	Berhasil	Objek cacat berhasil diekstraksi fiturnya.
6	2	Kanan	Objek 5 NG	5	4	0	4	Tidak Berhasil	Objek tidak terdeteksi
7	3	Kiri	Objek 3 OK dan 2 NG	5	5	3	2	Berhasil	Fitur objek OK dan NG

									berhasil diekstraksi.
8	3	Tengah	Objek 3 OK dan 2 NG	5	5	3	2	Berhasil	Fitur objek OK dan NG berhasil diekstraksi.
9	3	Kanan	Objek 3 OK dan 2 NG	5	5	3	2	Berhasil	Fitur objek OK dan NG berhasil diekstraksi.
10	4	Kiri	Mayoritas OK	5	5	4	1	Berhasil	Fitur objek mayoritas OK berhasil diekstraksi.
11	4	Tengah	Mayoritas OK	5	5	4	1	Berhasil	Fitur objek mayoritas OK berhasil diekstraksi.
12	4	Kanan	Mayoritas OK	5	5	4	1	Berhasil	Fitur objek mayoritas OK berhasil diekstraksi.
13	5	Kiri	Mayoritas NG	5	5	2	3	Tidak Berhasil	Sistem salah mendeteksi

14	5	Tengah	Mayoritas NG	5	5	1	4	Berhasil	Fitur objek mayoritas NG berhasil diekstraksi. Fitur objek mayoritas NG berhasil diekstraksi.
15	5	Kanan	Mayoritas NG	5	5	1	4	Berhasil	

Berdasarkan **tabel 4.16**, proses ekstraksi fitur dilakukan pada lima siklus pengujian dengan tiga posisi kamera pada setiap siklus. Setiap posisi kamera memiliki lima objek uji, sehingga total data objek yang digunakan adalah 75 data objek. Hasil rekapitulasi menunjukkan bahwa sebagian besar objek dapat terdeteksi dan diekstraksi fiturnya dengan baik oleh sistem.

Pada siklus 1 posisi kanan, seluruh objek berhasil terdeteksi dan diekstraksi fiturnya, tetapi terdapat satu objek OK yang diklasifikasikan sebagai NG. Pada siklus 2 posisi kanan, terdapat satu objek NG yang tidak terdeteksi sehingga fitur objek tersebut tidak dapat diekstraksi. Pada siklus 5 posisi kiri, seluruh objek berhasil terdeteksi, tetapi terdapat satu objek NG yang diklasifikasikan sebagai OK. Kondisi tersebut dapat dipengaruhi oleh sudut pandang kamera, pencahayaan, posisi objek, atau hasil segmentasi citra yang kurang sempurna.

Secara umum, hasil rekapitulasi menunjukkan bahwa sistem mampu melakukan ekstraksi fitur pada setiap kondisi objek, baik objek OK, objek NG, objek campuran, mayoritas OK, maupun mayoritas NG. Data hasil ekstraksi fitur tersebut kemudian digunakan sebagai dasar dalam proses klasifikasi objek pada tahap berikutnya.

Parameter hasil ekstraksi fitur yang digunakan dalam proses klasifikasi ditentukan berdasarkan program Python yang digunakan pada sistem. Parameter

tersebut menjadi acuan sistem dalam menentukan apakah objek termasuk kategori OK atau NG. Parameter klasifikasi objek OK dan NG berdasarkan program Python ditunjukkan pada **tabel 4.17**.

Tabel 4. 17 Parameter Klasifikasi Objek OK dan NG Berdasarkan Program Python

No	Parameter	Batas Objek OK	Kondisi Objek NG	Keterangan
1	Area	≥ 1500 piksel	Objek dengan area < 1500 piksel tidak diproses sebagai objek valid	Area digunakan sebagai batas minimum agar noise atau objek kecil tidak ikut terbaca sebagai objek uji.
2	Lebar dan Tinggi	Lebar ≥ 25 piksel dan tinggi ≥ 25 piksel	Objek dengan lebar atau tinggi < 25 piksel tidak diproses sebagai objek valid	Digunakan untuk menyaring objek yang terlalu kecil atau noise pada citra.
3	<i>Aspect ratio</i>	0,45 sampai 1,80	$< 0,45$ atau $> 1,80$	Menunjukkan perbandingan lebar dan tinggi objek. Objek di luar batas tersebut dianggap memiliki proporsi bentuk yang tidak sesuai.
4	<i>Extent</i>	$\geq 0,28$	$< 0,28$	Menunjukkan perbandingan area objek terhadap bounding box. Nilai <i>extent</i> rendah menunjukkan area isi objek terlalu kecil.
5	<i>Solidity</i>	$\geq 0,40$	$< 0,40$	Menunjukkan kepadatan kontur objek terhadap convex hull. Nilai

				<i>solidity</i> rendah menunjukkan kontur tidak padat atau bentuk tidak utuh.
6	<i>Circularity</i>	Tidak digunakan sebagai penentu utama pada kondisi default	Digunakan apabila fitur <i>circularity</i> diaktifkan, dengan batas 0,15 sampai 1,00	Pada program, <i>USE_CIRCULARITY</i> bernilai False, sehingga <i>circularity</i> hanya ditampilkan sebagai nilai fitur dan tidak langsung menentukan OK/NG.
7	<i>Hole Area</i>	< 250 piksel	≥ 250 piksel dan <i>hole ratio</i> $\geq 0,01$	Digunakan untuk mendeteksi adanya lubang atau bagian terbuka pada objek.
8	<i>Hole Ratio</i>	< 0,01 atau <i>hole area</i> < 250 piksel	$\geq 0,01$ dan <i>hole area</i> ≥ 250 piksel	Objek dikategorikan memiliki defect <i>HOLE/OPEN</i> apabila luas <i>hole</i> dan rasio <i>hole</i> melewati batas program.
9	Kontur Objek	Kontur memenuhi parameter area, ukuran, <i>aspect ratio</i> , <i>extent</i> , dan <i>solidity</i>	Kontur tidak memenuhi salah satu parameter utama	Kontur digunakan sebagai dasar ekstraksi fitur dan penentuan status objek.
10	Status Klasifikasi	Tidak memiliki defect	Memiliki satu atau lebih defect	Objek diklasifikasikan OK apabila tidak ditemukan defect. Objek diklasifikasikan NG apabila terdapat defect seperti <i>HOLE/OPEN</i> , <i>LOW_EXTENT</i> , <i>LOW_SOLIDITY</i> , <i>LOW_CIRC</i> , atau <i>BAD_AR</i> .

Berdasarkan **tabel 4.17**, klasifikasi objek pada sistem dilakukan dengan membandingkan nilai fitur geometris objek terhadap batas parameter yang terdapat pada program Python. Objek dikategorikan sebagai OK apabila memenuhi parameter sistem dan tidak memiliki defect. Sebaliknya, objek dikategorikan sebagai NG apabila terdapat indikasi defect seperti *hole/open*, *aspect ratio* di luar batas, nilai *extent* rendah, atau nilai *solidity* rendah.

Parameter *hole area* dan *hole ratio* menjadi bagian penting dalam mendeteksi objek yang memiliki lubang atau bagian terbuka. Jika objek memiliki *hole area* minimal 250 px dan *hole ratio* minimal 0,01, maka objek akan dikategorikan sebagai NG. Selain itu, parameter *aspect ratio*, *extent*, dan *solidity* digunakan untuk membantu sistem mengenali bentuk objek yang tidak sesuai dengan karakteristik objek uji.

Pada program yang digunakan, *circularity* ditampilkan sebagai nilai fitur, tetapi tidak digunakan sebagai penentu utama pada kondisi default karena `USE_CIRCULARITY` bernilai False. Dengan demikian, *circularity* berfungsi sebagai parameter pendukung untuk melihat tingkat kemiripan bentuk objek terhadap lingkaran, sedangkan keputusan utama klasifikasi dipengaruhi oleh defect yang terdeteksi berdasarkan parameter program.

Dengan demikian, proses ekstraksi fitur dinilai berhasil mendukung proses klasifikasi objek pada sistem inspeksi visual. Data hasil ekstraksi fitur dapat digunakan sebagai dasar dalam menentukan objek OK dan NG secara otomatis berdasarkan parameter geometris yang telah ditentukan pada program Python.

4.5 Pengujian Klasifikasi dan Pengukuran Diameter Objek

Pengujian pengukuran dan klasifikasi objek dilakukan untuk mengetahui kemampuan sistem dalam mengidentifikasi objek ke dalam kategori OK dan NG berdasarkan hasil ekstraksi fitur geometris objek. Bagian ini digunakan untuk membandingkan kondisi aktual objek dengan hasil klasifikasi yang diberikan oleh sistem. Nilai fitur objek tidak ditampilkan kembali secara rinci pada bagian ini karena telah dibahas pada bagian 4.4.

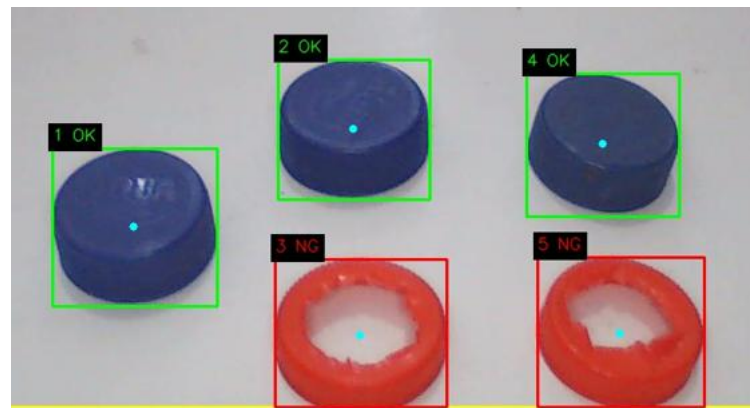
Proses klasifikasi dilakukan setelah sistem berhasil melakukan akuisisi citra, konversi grayscale, *thresholding* Otsu, deteksi kontur, dan ekstraksi fitur objek. Parameter hasil ekstraksi fitur yang digunakan dalam proses klasifikasi meliputi area, *aspect ratio*, *circularity*, *solidity*, *extent*, dan *hole ratio*. Objek diklasifikasikan sebagai OK apabila memenuhi parameter sistem dan tidak memiliki defect, sedangkan objek diklasifikasikan sebagai NG apabila memiliki bentuk tidak sesuai, terdapat lubang atau *hole/open*, atau memiliki nilai parameter yang berada di luar batas yang telah ditentukan.

Objek dengan bentuk mendekati lingkaran sempurna dan tidak memiliki cacat akan diklasifikasikan sebagai objek OK. Sementara itu, objek yang memiliki bentuk tidak sesuai, terdapat lubang (*hole/open*), atau memiliki nilai parameter geometris di luar batas yang telah ditentukan akan diklasifikasikan sebagai objek NG.

Metode pengujian dilakukan dengan menempatkan beberapa objek pada area inspeksi kemudian sistem melakukan proses pengolahan citra secara otomatis menggunakan program berbasis Python *OpenCV*. Sistem selanjutnya menghitung parameter geometris objek dan melakukan proses klasifikasi secara *real-time* berdasarkan nilai parameter yang diperoleh. Pengujian dilakukan sebanyak lima siklus. Pada setiap siklus, sistem melakukan pengambilan citra pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Setiap capture berisi lima objek uji, sehingga total data objek yang digunakan dalam pengujian klasifikasi adalah:

$$5 \text{ siklus} \times 3 \text{ posisi kamera} \times 5 \text{ objek} = 75 \text{ data objek.}$$

Hasil klasifikasi objek ditunjukkan pada **gambar 4.44**. dan **gambar 4.45** untuk hasil klasifikasi objek pada program python.



Gambar 4.44 Hasil Klasifikasi Objek Pada Siklus Ke-3 Dengan Kondisi Campuran 3 Objek OK Dan 2 Objek NG

Berdasarkan **Gambar 4.44**, sistem berhasil melakukan klasifikasi objek pada kondisi campuran OK dan NG. Objek yang diklasifikasikan sebagai OK ditandai dengan bounding box berwarna hijau, sedangkan objek yang diklasifikasikan sebagai NG ditandai dengan bounding box berwarna merah. Hasil tersebut menunjukkan bahwa sistem mampu membedakan objek OK dan NG berdasarkan parameter geometris yang diperoleh dari proses ekstraksi fitur.

RAW=5;STABLE=5;OK=3;NG=2;

Gambar 4.45 Tampilan Hasil Klasifikasi Objek OK Dan NG Pada Program Python *Opencv*

Pada **gambar 4.45**, sistem juga menampilkan hasil klasifikasi melalui program Python *OpenCV*. Tampilan tersebut memperlihatkan bahwa sistem dapat menampilkan status objek OK dan NG secara *real-time* pada layar monitoring.

4.5.1 Pengujian Klasifikasi Objek Berdasarkan Kondisi Aktual

Sebelum dilakukan evaluasi terhadap hasil klasifikasi sistem, kondisi aktual setiap objek terlebih dahulu ditentukan berdasarkan pengamatan langsung terhadap kondisi fisik objek. Penentuan kondisi aktual dilakukan dengan mengamati bentuk objek, keberadaan cacat, serta kesesuaian objek terhadap karakteristik objek normal.

Pada penelitian ini, kondisi aktual digunakan sebagai acuan (*ground truth*) untuk membandingkan hasil klasifikasi yang diberikan oleh sistem. Objek dengan

kondisi fisik normal, berbentuk lingkaran utuh, dan tidak memiliki kerusakan dikategorikan sebagai OK. Sedangkan objek yang memiliki lubang, perubahan bentuk, atau tidak memenuhi karakteristik lingkaran ideal dikategorikan sebagai NG. Hasil pengujian klasifikasi berdasarkan kondisi aktual dan hasil prediksi sistem ditunjukkan pada **tabel 4.18**.

Tabel 4. 18 Data Pengujian Klasifikasi Objek

No	Siklus	Posisi Kamera	Kondisi Objek	Aktual OK	Aktual NG	Prediksi OK	Prediksi NG	Benar	Salah	Keterangan
1	1	Kiri	Semua OK	5	0	5	0	5	0	Sesuai
2	1	Tengah	Semua OK	5	0	5	0	5	0	Sesuai
3	1	Kanan	Semua OK	5	0	4	1	4	1	1 Objek OK terbaca NG
4	2	Kiri	Semua NG	0	5	0	5	5	0	Sesuai
5	2	Tengah	Semua NG	0	5	0	5	5	0	Sesuai
6	2	Kanan	Semua NG	0	5	0	4	4	1	1 Objek NG tidak terbaca
7	3	Kiri	Campuran 3 OK dan 2 NG	3	2	3	2	5	0	Sesuai
8	3	Tengah	Campuran 3 OK dan 2 NG	3	2	3	2	5	0	Sesuai

			Campuran							
9	3	Kanan	3 OK dan 2 NG	3	2	3	2	5	0	Sesuai
10	4	Kiri	Mayoritas OK	4	1	4	1	5	0	Sesuai
11	4	Tengah	Mayoritas OK	4	1	4	1	5	0	Sesuai
12	4	Kanan	Mayoritas OK	4	1	4	1	5	0	Sesuai
13	5	Kiri	Mayoritas NG	1	4	2	3	4	1	1 Objek NG terbaca OK
14	5	Tengah	Mayoritas NG	1	4	1	4	5	0	Sesuai
15	5	Kanan	Mayoritas NG	1	4	1	4	5	0	Sesuai

Berdasarkan **tabel 4.18**, pengujian dilakukan terhadap 75 data objek yang terdiri dari objek OK dan NG. Hasil pengujian menunjukkan bahwa sistem mampu melakukan klasifikasi terhadap sebagian besar objek sesuai dengan kondisi aktual. Dari keseluruhan data pengujian, diperoleh sebanyak 72 data yang berhasil diklasifikasikan dengan benar dan 3 data mengalami kesalahan klasifikasi.

Kesalahan klasifikasi terjadi pada beberapa kondisi, yaitu pada siklus pertama posisi kamera kanan terdapat satu objek OK yang terbaca sebagai NG, pada siklus kedua posisi kamera kanan terdapat satu objek NG yang tidak berhasil terdeteksi, serta pada siklus kelima posisi kamera kiri terdapat satu objek NG yang terbaca sebagai OK.

Kesalahan tersebut dapat disebabkan oleh beberapa faktor seperti perubahan posisi objek terhadap kamera, variasi pencahayaan, hasil segmentasi citra yang

kurang sempurna, maupun bentuk objek yang mengalami deformasi sehingga memiliki karakteristik geometris yang mendekati objek normal.

4.5.2 Evaluasi Hasil Klasifikasi Sistem

Untuk mengetahui tingkat kesesuaian hasil klasifikasi sistem terhadap kondisi aktual objek, dilakukan perbandingan antara hasil klasifikasi sistem dengan kondisi aktual objek. Hasil perbandingan kondisi aktual dan hasil klasifikasi sistem ditunjukkan pada **tabel 4.19**.

Tabel 4. 19 Perbandingan Kondisi Aktual dan Hasil Klasifikasi Sistem

No	Kondisi Aktual	Jumlah Data Aktual	Hasil Sistem Sesuai	Hasil Sistem Tidak Sesuai	Tingkat Kesesuaian	Keterangan
1	OK	39	38	1	97,44%	Terdapat 1 objek OK yang terbaca sebagai NG akibat perbedaan pencahayaan dan segmentasi.
2	NG	36	34	2	94,44%	Terdapat 1 objek NG yang tidak terdeteksi dan 1 objek NG terbaca sebagai OK.
3	Total	75	72	3	96,00%	Sistem mampu mengklasifikasikan sebagian besar objek sesuai kondisi aktual.

Berdasarkan **tabel 4.19**, dari total 75 data objek yang diuji, sistem berhasil mengklasifikasikan 72 data dengan benar dan terdapat 3 data yang mengalami

kesalahan klasifikasi. Pada kondisi aktual OK, sistem berhasil mengenali 38 dari 39 objek dengan benar dengan tingkat kesesuaian sebesar 97,44%. Sementara itu, pada kondisi aktual NG, sistem berhasil mengenali 34 dari 36 objek dengan tingkat kesesuaian sebesar 94,44%.

Secara keseluruhan, tingkat keberhasilan sistem dalam melakukan klasifikasi objek mencapai 96,00%. Hasil tersebut menunjukkan bahwa metode thresholding Otsu, deteksi kontur, dan ekstraksi fitur geometris mampu digunakan untuk membedakan objek OK dan NG berdasarkan karakteristik bentuk objek. Lalu dilakukan evaluasi menggunakan confusion matrix yang dimana karena penelitian berfokus pada pendeteksian objek cacat, maka kelas NG digunakan sebagai kelas positif (*positive class*), sedangkan kelas OK sebagai kelas negatif (*negative class*). Hasil ditunjukkan pada **tabel 4.20**.

Tabel 4. 20 Confusion Matrix Hasil Klasifikasi Objek

Kondisi Aktual / Prediksi Sistem	Prediksi OK	Prediksi NG	Total Aktual
Aktual OK	38	1	39
Aktual NG	2	34	36
Total Prediksi	40	35	75

Berdasarkan **tabel 4.34** diperoleh nilai TP sebesar 34, TN sebesar 38, FP sebesar 1 dan FN sebesar 2, nilai selanjutnya digunakan untuk menghitung accuracy berdasarkan persamaan (2.1), precision berdasarkan persamaan (2.2), recall berdasarkan persamaan (2.3), dan F1-Score berdasarkan persamaan (2.4). Hasil perhitungan evaluasi klasifikasi objek ditunjukkan pada **tabel 4.21**.

Tabel 4. 21 Hasil Evaluasi Klasifikasi Objek

No	Parameter Evaluasi	Persamaan	Rumus	Perhitungan	Hasil
1	Accuracy	Persamaan (2.1)	$\frac{(TP + TN)}{Total} \times 100\%$	$(34 + 38) / 75 \times 100\%$	96,00%
2	Precision	Persamaan (2.2)	$TP / (TP + FP) \times 100\%$	$34 / (34 + 1) \times 100\%$	97,14%
3	Recall	Persamaan (2.3)	$TP / (TP + FN) \times 100\%$	$34 / (34 + 2) \times 100\%$	94,44%
4	F1-Score	Persamaan (2.4)	$\frac{2 \times Precision \times Recall}{(Precision + Recall)}$	$2 \times 97,14\% \times 94,44\% / (97,14\% + 94,44\%)$	95,77%

Berdasarkan hasil evaluasi klasifikasi objek, sistem memperoleh nilai accuracy sebesar 96,00%. Nilai tersebut menunjukkan bahwa sistem mampu mengklasifikasikan sebagian besar objek sesuai dengan kondisi aktualnya. Dari total 75 data objek, terdapat 72 data yang berhasil diklasifikasikan dengan benar dan 3 data yang tidak sesuai.

Nilai precision sebesar 97,14% menunjukkan bahwa sebagian besar objek yang diprediksi sebagai NG benar-benar merupakan objek NG. Hal ini menunjukkan bahwa sistem memiliki kemampuan yang baik dalam menghindari kesalahan klasifikasi objek OK menjadi NG. Sementara itu, nilai recall sebesar 94,44% menunjukkan bahwa sistem mampu mendeteksi sebagian besar objek NG dari seluruh objek NG yang diuji, termasuk objek dengan kondisi lubang maupun deformasi bentuk.

Nilai F1-score sebesar 95,77% menunjukkan keseimbangan antara precision dan recall. Dengan nilai tersebut, sistem dapat dikatakan memiliki performa klasifikasi yang baik dalam membedakan objek OK dan NG. Meskipun masih terdapat beberapa kesalahan klasifikasi, hasil pengujian menunjukkan bahwa metode klasifikasi berbasis parameter geometris mampu mendukung proses inspeksi visual objek geometris berbentuk lingkaran secara otomatis dan *real-time*.

Dengan demikian, sistem klasifikasi objek pada penelitian ini dinilai berhasil mendukung proses inspeksi visual karena mampu membedakan objek OK dan NG berdasarkan hasil ekstraksi fitur geometris dan parameter klasifikasi yang telah ditentukan pada program Python.

4.5.3 Pengujian Pengukuran Diameter Objek Menggunakan Jangka Sorong

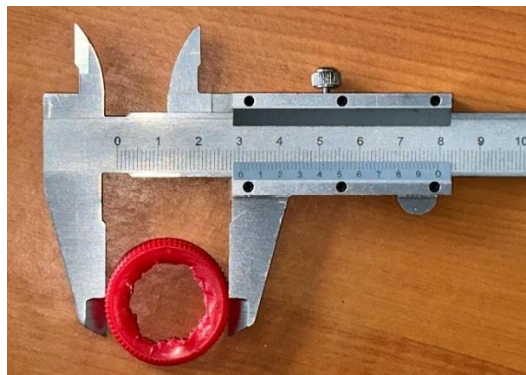
Setelah dilakukan evaluasi terhadap performa klasifikasi, selanjutnya dilakukan pengujian kemampuan sistem dalam mengukur diameter objek melakukan klasifikasi objek, untuk mengetahui kemampuan dalam melakukan pengukuran diameter objek pengukuran sistem kemudian dibandingkan dengan hasil pengukuran manual menggunakan jangka sorong sebagai alat ukur referensi.

Pengukuran menggunakan jangka sorong dilakukan untuk memperoleh nilai diameter aktual objek yang digunakan sebagai pembanding terhadap hasil pengukuran sistem. Parameter yang dibandingkan meliputi diameter luar objek, diameter lubang pada objek NG, serta selisih hasil pengukuran antara sistem dan pengukuran manual. Hasil pengukuran manual menggunakan jangka sorong ditunjukkan pada **gambar 4.46** hingga **gambar 4.48**.



Gambar 4. 46 Pengukuran Manual Objek OK Menggunakan Jangka Sorong

Selanjutnya, pengukuran manual objek NG menggunakan jangka sorong ditunjukkan pada **gambar 4.47**.



Gambar 4. 47 Pengukuran Manual Objek NG Menggunakan Jangka Sorong

Kemudian, pengukuran manual objek NG dengan kondisi deformasi menggunakan jangka sorong ditunjukkan pada **gambar 4.48**.



Gambar 4. 48 Pengukuran Manual Objek NG (Deformasi) Menggunakan Jangka Sorong

Berdasarkan hasil pengukuran tersebut diperoleh data diameter aktual objek seperti ditunjukkan pada **tabel 4.22**.

Tabel 4. 22 Data Pengukuran Manual Objek Uji Menggunakan Jangka Sorong

No	Jenis Objek	Diameter Luar (mm)	Diameter Lubang (mm)	Kondisi Aktual	Keterangan
1	Tutup botol normal 1	30,00	-	OK	Objek berbentuk lingkaran utuh tanpa cacat.
2	Tutup botol normal 2	28,00	-	OK	Tidak terdapat lubang atau kerusakan pada objek.
3	Tutup botol berlubang 1	29,00	17,00	NG	Terdapat lubang pada bagian tengah objek.
4	Tutup botol berlubang 2	26,00	15,00	NG	Bagian tengah objek mengalami cacat (<i>hole/open</i>).
5	Tutup botol deformasi (gepeng)	35,00	-	NG	Bentuk tidak simetris, tidak membentuk lingkaran sempurna.

Berdasarkan **tabel 4.36**, objek uji yang digunakan terdiri dari tutup botol dengan kondisi normal, berlubang, dan mengalami deformasi bentuk. Pengukuran

dilakukan menggunakan jangka sorong untuk memperoleh nilai diameter luar serta mendeteksi adanya perubahan struktur fisik pada objek.

Objek dengan kondisi normal memiliki diameter yang relatif seragam dan tidak menunjukkan perbedaan signifikan antara bentuk satu dengan lainnya. Hal ini menunjukkan bahwa objek masih berada dalam kondisi geometris yang stabil dan sesuai karakteristik lingkaran, sehingga dikategorikan sebagai OK.

Sementara itu, objek dengan kondisi NG terdiri dari dua kategori, yaitu objek berlubang dan objek yang mengalami deformasi bentuk. Pada objek berlubang, terdapat bagian tengah yang tidak utuh sehingga menghasilkan nilai diameter lubang yang dapat diukur menggunakan jangka sorong. Sedangkan pada objek deformasi, tidak ditemukan lubang, namun terjadi perubahan bentuk menjadi tidak simetris (oval/gepeng) yang menyebabkan ketidaksesuaian terhadap bentuk lingkaran ideal. Selanjutnya hasil pengukuran jangka sorong dibandingkan dengan hasil pengukuran sistem seperti ditunjukkan pada **tabel 4.23**.

Tabel 4. 23 Perbandingan Hasil Pengukuran Diameter Menggunakan Jangka Sorong dan Sistem

No	Jenis Objek	Kondisi Manual	Kondisi Sistem	Diameter Manual (mm)	Diameter Sistem (mm)	Selisih (mm)	Error (%)
1	Tutup botol normal 1	OK	OK	30	30	0,00	0,00%
2	Tutup botol normal 2	OK	OK	28	28	0,00	0,00%
3	Tutup botol berlubang 1	NG	NG	29	29,81	0,81	2,79%
4	Tutup botol berlubang 2	NG	OK	26	28,26	2,26	8,69%
5	Tutup botol deformasi (gepeng)	NG	NG	35	29,61	5,39	15,40%

Berdasarkan **tabel 4.38**, hasil pengukuran diameter menggunakan sistem menunjukkan nilai yang mendekati hasil pengukuran manual. Pada objek normal, sistem mampu menghasilkan nilai pengukuran yang hampir sama dengan hasil pengukuran jangka sorong, sedangkan pada objek berlubang dan objek deformasi terdapat selisih pengukuran yang lebih besar. Perbedaan tersebut dipengaruhi oleh proses segmentasi citra, deteksi kontur, dan konversi ukuran piksel menjadi satuan milimeter. Untuk gambaran terhadap hasil manual dengan sistem dapat dilihat pada grafik **gambar 4.49**.



Gambar 4. 49 Grafik Perbandingan Diameter Manual dan Sistem

Untuk memperjelas hasil perbandingan, data pengukuran divisualisasikan dalam bentuk grafik seperti ditunjukkan pada **gambar 4.49**. Grafik menunjukkan bahwa hasil pengukuran sistem mengikuti pola yang hampir sama dengan hasil pengukuran manual. Perbedaan terbesar terjadi pada objek deformasi, sedangkan objek normal memiliki nilai pengukuran yang hampir identik. Rata-rata persentase *error* sebesar 8,96% menunjukkan bahwa sistem mampu melakukan pengukuran diameter dengan tingkat ketelitian yang baik.

4.6 Pengujian Variasi Posisi Kamera

Pengujian variasi posisi kamera dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses inspeksi visual pada beberapa posisi pengambilan citra. Pengujian ini bertujuan untuk memastikan bahwa kamera dapat bergerak menuju posisi kiri, tengah, dan kanan sesuai perintah yang diberikan, serta mampu melakukan proses pengambilan citra pada setiap posisi tersebut.

Pada sistem inspeksi visual ini, pergerakan kamera dikendalikan menggunakan motor *stepper* NEMA23 yang digerakkan melalui driver TB6600 dan dikontrol oleh Arduino Mega 2560. Kamera dipasang pada mekanisme linear actuator sehingga dapat berpindah posisi secara horizontal. Posisi kamera yang digunakan dalam pengujian ini terdiri dari tiga titik, yaitu posisi kiri, posisi tengah, dan posisi kanan.

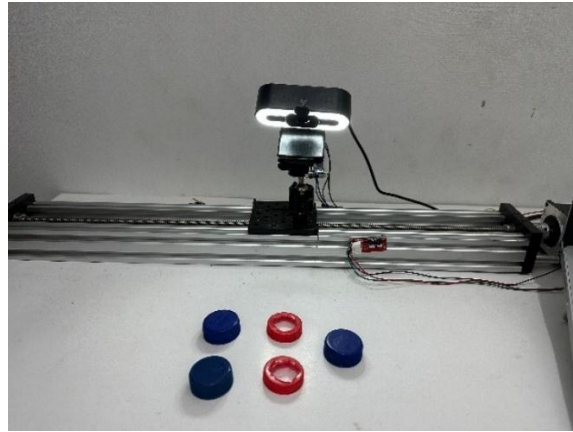
Sebelum pengujian dilakukan, kamera terlebih dahulu ditempatkan pada posisi tengah sebagai titik acuan awal. Titik tengah tersebut ditetapkan melalui perintah SETCENTER pada sistem. Setelah posisi tengah ditentukan, sistem dapat melakukan pergerakan kamera secara manual maupun otomatis menuju posisi kiri, tengah, dan kanan. Pengujian ini dilakukan untuk memastikan bahwa setiap posisi kamera dapat digunakan sebagai titik pengambilan citra objek.

Pengujian variasi posisi kamera dilakukan dengan menjalankan sistem pada mode otomatis. Pada mode ini, kamera akan bergerak dari posisi tengah menuju posisi kiri, kemudian sistem melakukan pengambilan citra dan pengolahan citra. Setelah proses pengambilan citra selesai, kamera bergerak kembali ke posisi tengah untuk melakukan proses inspeksi pada posisi tengah. Selanjutnya, kamera bergerak menuju posisi kanan dan melakukan proses inspeksi pada posisi tersebut. Setelah seluruh proses selesai, kamera akan kembali ke posisi tengah sebagai posisi standby. Hasil posisi kamera pada proses pengujian variasi posisi kamera ditunjukkan pada **gambar 4.50** sampai **gambar 4.52**.



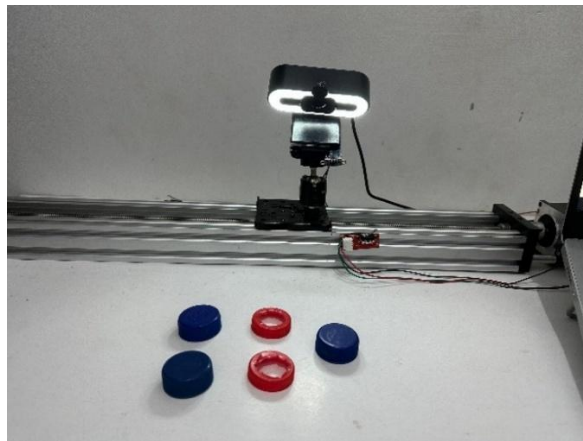
Gambar 4. 50 Posisi Kamera Pada Sisi Kiri Area Inspeksi

Selanjutnya, posisi kamera pada bagian tengah area inspeksi ditunjukkan pada **gambar 4.51**.



Gambar 4. 51 Posisi Kamera Pada Bagian Tengah Area Inspeksi

Kemudian, posisi kamera pada sisi kanan area inspeksi ditunjukkan pada **gambar 4.52**.



Gambar 4. 52 Posisi Kamera Pada Sisi Kanan Area Inspeksi

Berdasarkan **gambar 4.50** hingga **gambar 4.52**, sistem berhasil menggerakkan kamera menuju tiga posisi pengambilan citra, yaitu kiri, tengah, dan kanan. Setiap posisi kamera dapat digunakan untuk melakukan proses akuisisi citra objek. Pergerakan kamera dari satu posisi ke posisi lainnya dapat berjalan sesuai urutan yang telah diprogram, sehingga sistem mampu melakukan inspeksi objek pada beberapa area pengamatan.

Selain pengujian pergerakan kamera, sistem juga diuji untuk memastikan bahwa hasil pengolahan citra pada setiap posisi dapat ditampilkan pada antarmuka Delphi. Hasil pengolahan citra pada masing-masing posisi kamera ditunjukkan pada **gambar 4.53**.

STATUS SISTEM	
STATUS	: MENUNGGU_CAPTURE
POSISI	: TENGAH
JARAK	: 222 mm
DELAY	: PROSES CITRA
COM	: COM4
KONEKSI	: TERHUBUNG

Gambar 4. 53 Tampilan Posisi Kamera Pada Delphi

Data yang digunakan pada pengujian variasi posisi kamera berasal dari lima siklus pengujian. Pada setiap posisi kamera, sistem melakukan lima kali capture dengan lima objek pada setiap capture. Dengan demikian, setiap posisi kamera memiliki jumlah data sebagai berikut:

$$5 \text{ capture} \times 5 \text{ objek} = 25 \text{ data objek}$$

Untuk hasil pengujian variasi posisi kamera secara lebih rinci, dilakukan pencatatan hasil pengujian pada masing-masing posisi. Data hasil pengujian variasi posisi kamera disajikan pada **tabel 4.24**.

Tabel 4. 24 Hasil Pengujian Variasi Posisi Kamera

No	Posisi Kamera	Jumlah Capture	Jumlah Objek	Jarak Rata-Rata	Benar	Salah	Akurasi Posisi	Keterangan
1	Kiri	5	25	220 mm	24	1	96,00%	Kamera berhasil bergerak dan mengambil citra pada posisi kiri. Terdapat satu kesalahan

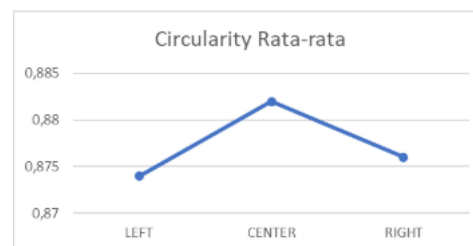
								klasifikasi pada objek yang memiliki kontur kurang jelas.
								Posisi tengah menghasilkan deteksi paling stabil karena posisi kamera lebih sejajar terhadap objek. Kamera berhasil bergerak dan mengambil citra pada posisi kanan, namun
2	Tengah	5	25	225 mm	25	0	100%	terdapat beberapa kesalahan klasifikasi akibat sudut pandang, pencahayaan, atau hasil segmentasi objek.
3	Kanan	5	25	230 mm	23	2	92,00%	

Berdasarkan **tabel 4.39**, sistem berhasil melakukan pergerakan kamera ke tiga posisi, yaitu kiri, tengah, dan kanan. Pada setiap posisi kamera, sistem melakukan lima kali capture dengan jumlah lima objek pada setiap capture, sehingga setiap posisi memiliki 25 data objek. Hasil pengujian menunjukkan bahwa seluruh pergerakan kamera berhasil dilakukan dengan tingkat keberhasilan 5/5 pada setiap posisi.

Posisi tengah memperoleh hasil paling stabil dengan nilai akurasi 100%. Hal ini terjadi karena posisi kamera berada lebih sejajar terhadap area inspeksi sehingga objek dapat terlihat lebih jelas dan hasil segmentasi citra lebih stabil. Posisi kiri memperoleh akurasi sebesar 96,00% dengan satu kesalahan klasifikasi.

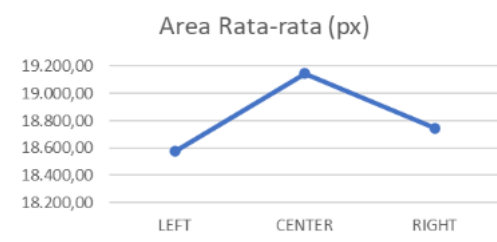
Sementara itu, posisi kanan memperoleh akurasi sebesar 92,00% dengan dua kesalahan klasifikasi.

Untuk memberikan gambaran yang lebih jelas mengenai pengaruh variasi posisi kamera terhadap hasil ekstraksi fitur, dilakukan visualisasi nilai rata-rata circularity, area, dan aspect ratio pada setiap posisi kamera. Grafik hasil pengujian circularity ditunjukkan pada **gambar 4.54**.



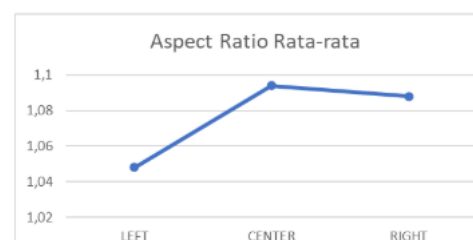
Gambar 4. 54 Grafik Rata-rata Circularity pada Setiap Posisi Kamera

Selanjutnya, grafik rata-rata *area* pada setiap posisi kamera ditunjukkan pada **gambar 4.55**.



Gambar 4. 55 Grafik Rata-rata Area pada Setiap Posisi Kamera

Kemudian, grafik rata-rata *aspect ratio* pada setiap posisi kamera ditunjukkan pada **gambar 4.56**.



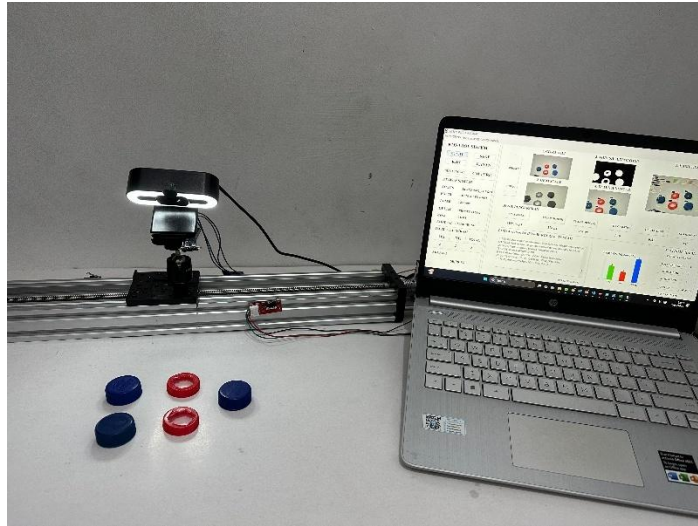
Gambar 4. 56 Grafik Rata-rata Aspect Ratio pada Setiap Posisi Kamera

Perbedaan hasil pada setiap posisi dapat dipengaruhi oleh sudut pandang kamera, pencahayaan, posisi objek, dan hasil segmentasi citra. Secara keseluruhan sistem mampu melakukan proses inspeksi visual pada tiga posisi kamera. Dengan demikian, pengujian variasi posisi kamera menunjukkan bahwa mekanisme pergerakan kamera dan proses pengambilan citra dapat berjalan sesuai fungsi yang dirancang.

4.7 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan untuk mengetahui apakah seluruh bagian sistem inspeksi visual dapat bekerja secara terintegrasi sesuai dengan rancangan yang telah dibuat. Pengujian ini mencakup kerja sama antara perangkat keras dan perangkat lunak, yaitu kamera USB, motor *stepper* NEMA23, driver TB6600, Arduino Mega 2560, sensor jarak VL53L0X, limit switch, push button, program Python *OpenCV*, serta antarmuka Delphi.

Pada pengujian ini, sistem diuji mulai dari proses koneksi antara Delphi dan Arduino, penentuan posisi awal kamera, pergerakan kamera menuju tiga posisi inspeksi, proses akuisisi citra, pengolahan citra, ekstraksi fitur objek, klasifikasi objek, hingga penampilan hasil inspeksi pada antarmuka Delphi. Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan inspeksi visual secara otomatis pada posisi kiri, tengah, dan kanan. Hasil keseluruhan sistem ditunjukkan pada **gambar 4.57**.



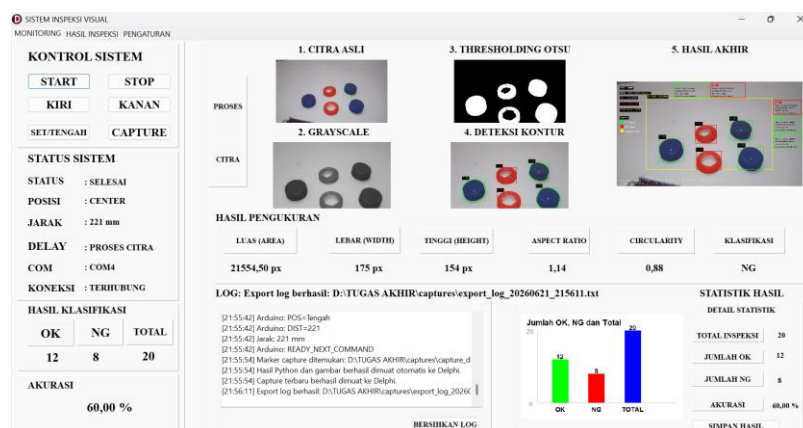
Gambar 4. 57 Perangkat Sistem Inspeksi Visual Secara Keseluruhan

Berdasarkan **gambar 4.57**, sistem terdiri dari mekanisme linear actuator sebagai jalur pergerakan kamera, kamera USB sebagai perangkat akuisisi citra, motor *stepper* sebagai penggerak posisi kamera, serta laptop yang digunakan untuk menjalankan antarmuka Delphi dan program pengolahan citra berbasis Python *OpenCV*. Objek uji diletakkan pada area inspeksi sehingga dapat ditangkap oleh kamera pada saat proses pengujian berlangsung.

Pengujian dimulai dengan menghubungkan Arduino Mega 2560 ke antarmuka Delphi melalui komunikasi serial. Setelah koneksi berhasil dilakukan, posisi tengah kamera ditetapkan menggunakan perintah SETCENTER. Posisi tengah digunakan sebagai acuan awal sistem sebelum kamera bergerak menuju posisi kiri dan kanan. Setelah posisi tengah ditentukan, sistem dijalankan menggunakan perintah START untuk memulai proses inspeksi otomatis.

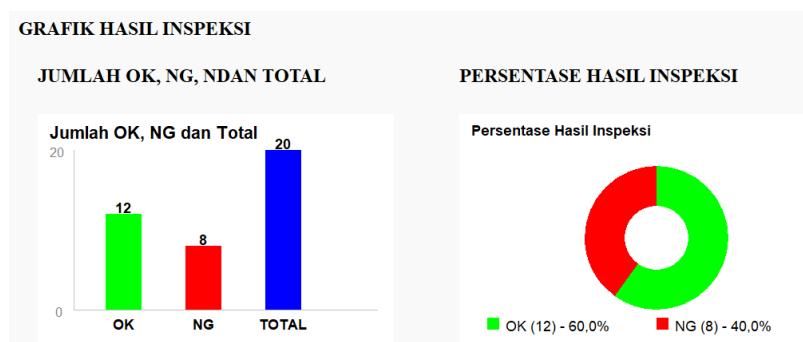
Satu siklus pengujian keseluruhan sistem terdiri dari beberapa tahapan, yaitu SETCENTER, START, kamera bergerak ke posisi kiri, capture dan proses citra, kamera bergerak ke posisi tengah, capture dan proses citra, kamera bergerak ke posisi kanan, capture dan proses citra, kemudian kamera kembali ke posisi tengah sebagai posisi standby. Setelah proses tersebut selesai, hasil klasifikasi ditampilkan pada antarmuka Delphi.

Pada mode otomatis, kamera bergerak menuju posisi kiri terlebih dahulu, kemudian sistem melakukan pengambilan citra dan menjalankan proses pengolahan citra. Setelah proses pengolahan citra selesai, kamera bergerak menuju posisi tengah untuk melakukan inspeksi berikutnya. Selanjutnya, kamera bergerak menuju posisi kanan dan melakukan proses inspeksi pada posisi tersebut. Setelah seluruh proses selesai, kamera kembali ke posisi tengah sebagai posisi standby. Hasil pengujian keseluruhan sistem pada antarmuka Delphi ditunjukkan pada **gambar 4.58** hingga **gambar 4.59**.



Gambar 4. 58 Tampilan Aplikasi Delphi Saat Sistem Melakukan Monitoring Dan Kontrol

Selanjutnya, tampilan hasil akhir klasifikasi objek OK dan NG ditunjukkan pada **gambar 4.59**.

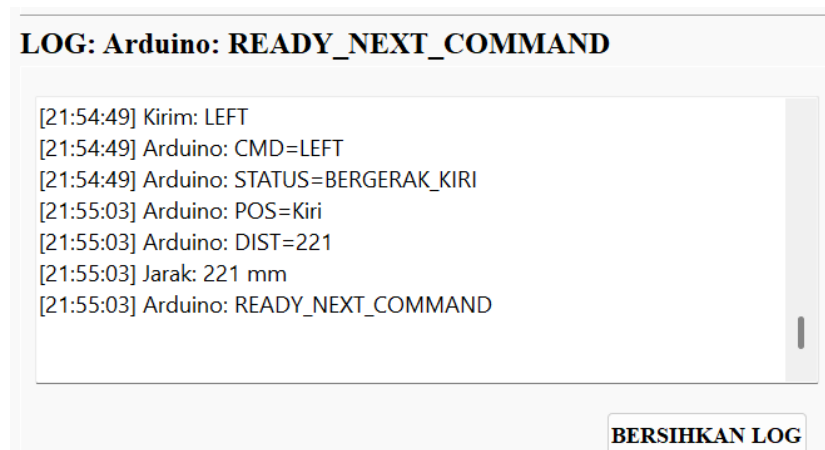


Gambar 4. 59 Tampilan Hasil Akhir Klasifikasi Objek OK Dan NG

Berdasarkan **gambar 4.58**, sistem berhasil menampilkan hasil pengolahan citra secara lengkap pada antarmuka Delphi. Hasil yang ditampilkan meliputi citra asli, citra grayscale, hasil *thresholding* Otsu, hasil deteksi kontur, dan hasil akhir

klasifikasi objek. Selain itu, sistem juga menampilkan informasi status sistem, posisi kamera, jarak objek, hasil klasifikasi OK dan NG, total inspeksi, serta grafik hasil inspeksi.

Pada pengujian yang ditampilkan, sistem mampu mendeteksi beberapa objek pada area inspeksi dan mengklasifikasikan objek ke dalam kategori OK dan NG. Objek yang memenuhi parameter bentuk lingkaran diklasifikasikan sebagai OK, sedangkan objek yang tidak sesuai dengan parameter klasifikasi dikategorikan sebagai NG. Hasil pengujian juga menunjukkan bahwa data hasil inspeksi dapat ditampilkan secara langsung pada antarmuka Delphi. Serta hasil bukti log proses pada delphi yang tertera pada **gambar 4.60**.



Gambar 4. 60 Bukti Komunikasi Dan Log Proses Sistem Pada Antarmuka Delphi

Berdasarkan log proses pada **gambar 4.60**, antarmuka Delphi dapat menampilkan informasi proses sistem mulai dari koneksi, pergerakan kamera, proses capture, hingga hasil klasifikasi. Log tersebut menjadi bukti bahwa perangkat lunak dan perangkat keras dapat saling terhubung selama proses pengujian berlangsung.

Untuk mengetahui hasil pengujian keseluruhan sistem secara lebih rinci, dilakukan pengujian sebanyak lima siklus. Setiap siklus terdiri dari tiga kali capture, yaitu pada posisi kiri, tengah, dan kanan. Setiap capture berisi lima objek, sehingga setiap siklus memiliki 15 data objek. Hasil pengujian keseluruhan sistem per siklus disajikan pada **tabel 4.25**.

Tabel 4. 25 Hasil Pengujian Keseluruhan Sistem per Siklus

No	Siklus	SETCENTER	Jumlah Capture	Objek Terdeteksi	Benar	Salah	Akurasi Siklus	Keterangan
1	1	Berhasil	3/3	15/15	14	1	93,33%	Sistem berhasil melakukan capture pada posisi kiri, tengah, dan kanan. Terdapat 1 objek OK terbaca NG.
2	2	Berhasil	3/3	15/15	14	1	93,33%	Sistem berhasil melakukan pengujian objek NG. Terdapat 1 objek NG tidak terbaca.
3	3	Berhasil	3/3	15/15	15	0	100%	Sistem berhasil mengklasifikasikan objek campuran OK dan NG.
4	4	Berhasil	3/3	15/15	15	0	100%	Sistem berhasil melakukan pengujian mayoritas objek OK.
5	5	Berhasil	3/3	15/15	14	1	93,33%	Sistem berhasil melakukan pengujian mayoritas objek NG. Terdapat 1 objek NG terbaca OK.

Berdasarkan **tabel 4.25**, sistem berhasil menjalankan pengujian keseluruhan sebanyak lima siklus. Pada setiap siklus, sistem mampu melakukan proses SETCENTER, menjalankan perintah START, menggerakkan kamera ke tiga posisi inspeksi, melakukan capture citra, memproses citra, dan menampilkan hasil klasifikasi objek.

Pada siklus 1, sistem memperoleh akurasi sebesar 93,33% karena terdapat satu objek OK yang terbaca sebagai NG. Pada siklus 2, sistem memperoleh akurasi sebesar 93,33% karena terdapat satu objek NG yang tidak terdeteksi. Pada siklus 3 dan siklus 4, sistem memperoleh akurasi sebesar 100% karena seluruh objek berhasil diklasifikasikan sesuai dengan kondisi aktual. Pada siklus 5, sistem memperoleh akurasi sebesar 93,33% karena terdapat satu objek NG yang terbaca sebagai OK.

Hasil tersebut menunjukkan bahwa sistem dapat bekerja secara otomatis dari awal sampai akhir. Meskipun masih terdapat beberapa kesalahan pada kondisi tertentu, sistem tetap mampu menjalankan proses inspeksi visual secara terintegrasi pada tiga posisi kamera.

Selain pengujian per siklus, dilakukan juga rekapitulasi tahapan kerja sistem untuk mengetahui keberhasilan setiap bagian sistem. Rekapitulasi tahapan pengujian keseluruhan sistem ditunjukkan pada **tabel 4.26**.

Tabel 4. 26 Rekapitulasi Tahapan Pengujian Keseluruhan Sistem

No	Tahapan Pengujian	Jumlah Pengujian	Hasil	Keterangan
1	Koneksi Delphi dengan Arduino	5	Berhasil	Sistem dapat terhubung melalui komunikasi serial.
2	Penentuan posisi awal SETCENTER	5	Berhasil	Posisi tengah dapat ditetapkan sebagai titik awal sistem.

	Pergerakan			Motor <i>stepper</i>
3	kamera ke posisi kiri	5	Berhasil	menggerakkan kamera menuju posisi kiri.
4	Akuisisi citra posisi kiri	5	Berhasil	Kamera berhasil mengambil citra pada posisi kiri.
5	Pergerakan kamera ke posisi tengah	5	Berhasil	Kamera bergerak menuju posisi tengah.
6	Akuisisi citra posisi tengah	5	Berhasil	Kamera berhasil mengambil citra pada posisi tengah.
7	Pergerakan kamera ke posisi kanan	5	Berhasil	Motor <i>stepper</i> menggerakkan kamera menuju posisi kanan.
8	Akuisisi citra posisi kanan	5	Berhasil	Kamera berhasil mengambil citra pada posisi kanan.
9	Pengolahan citra	15 capture	Berhasil	Sistem menghasilkan citra grayscale, <i>thresholding</i> Otsu, dan deteksi kontur.
10	Ekstraksi fitur objek	75 objek	Berhasil	Sistem menghasilkan fitur area, <i>aspect ratio</i> , <i>circularity</i> , <i>solidity</i> , <i>extent</i> , dan <i>hole ratio</i> .
11	Klasifikasi objek	75 objek	Berhasil	Sistem menampilkan hasil klasifikasi OK dan NG.
12	Indikator hasil	5 siklus	Berhasil	Lampu indikator OK/NG menyala sesuai hasil klasifikasi sistem.
13	Log proses pada Delphi	5 siklus	Berhasil	Aplikasi menampilkan log proses koneksi, pergerakan

				kamera, capture, dan hasil klasifikasi.
14	Tombol Stop dan Emergency	5	Berhasil	Sistem dapat dihentikan melalui tombol Stop dan Emergency.

Berdasarkan **tabel 4.26**, seluruh tahapan pengujian sistem dapat berjalan sesuai dengan rancangan. Sistem dapat menghubungkan perangkat keras dan perangkat lunak secara terintegrasi. Arduino Mega 2560 berfungsi sebagai pengendali motor *stepper* dan pembaca sensor, sedangkan Delphi digunakan sebagai antarmuka monitoring dan kontrol sistem. Program Python *OpenCV* berfungsi untuk melakukan pengolahan citra dan klasifikasi objek.

Hasil pengujian menunjukkan bahwa sistem mampu melakukan inspeksi visual secara otomatis pada tiga posisi kamera, yaitu kiri, tengah, dan kanan. Setiap posisi kamera dapat digunakan untuk proses pengambilan citra, kemudian citra tersebut diproses untuk menghasilkan fitur geometris dan hasil klasifikasi objek. Hasil akhir pengujian ditampilkan pada antarmuka Delphi sehingga pengguna dapat memantau status sistem, posisi kamera, hasil citra, serta data klasifikasi secara langsung.

Dengan demikian, pengujian keseluruhan sistem menunjukkan bahwa sistem inspeksi visual berbasis metode *thresholding* Otsu dan deteksi kontur dapat bekerja sesuai dengan tujuan perancangan. Sistem mampu mengintegrasikan kendali motor *stepper*, akuisisi citra, pengolahan citra, ekstraksi fitur, klasifikasi objek, indikator hasil, dan tampilan hasil inspeksi dalam satu sistem yang saling terhubung.