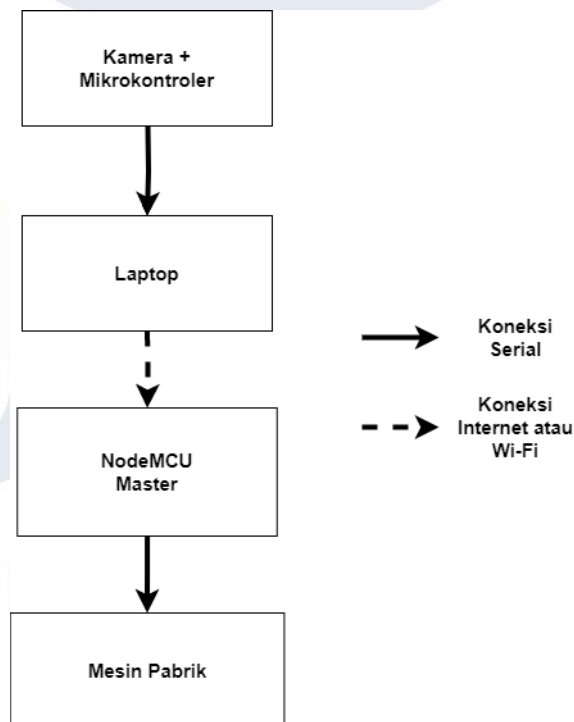


BAB II

KONSEP DESAIN & SPESIFIKASI SISTEM

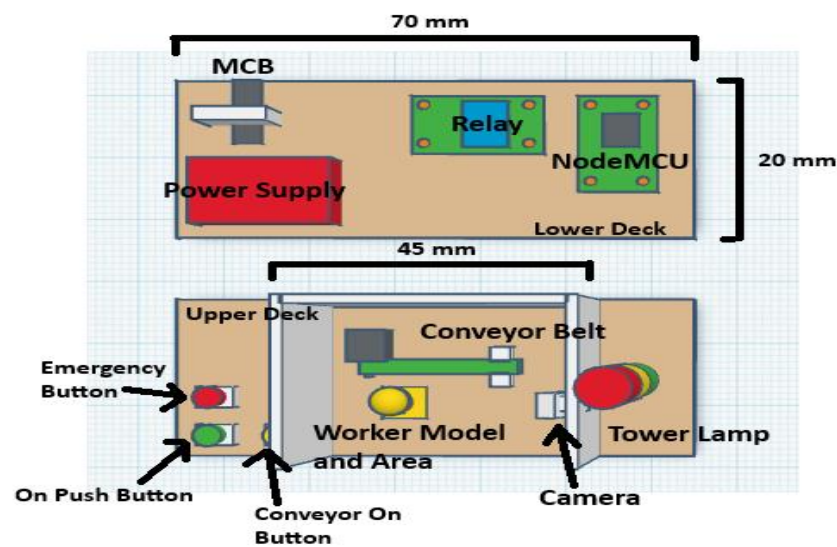
2.1 Konsep Desain Sistem

Produk ‘Implementasi Image Processing Untuk Keamanan di Lantai Pabrik’ terdiri dari sebuah *processor unit* seperti laptop, kemudian NodeMCU untuk menerima data dari laptop, dan sebuah kamera seperti ESP32Cam. Kamera yang digunakan juga memiliki kapasitas Wi-Fi untuk mengirimkan gambar digital ke laptop secara *wireless*. *Processor* menerima gambar digital kemudian mengolah gambar digital tersebut menggunakan *image processing*. Hasil dari *image processing* digunakan untuk menentukan perintah yang akan dikirimkan oleh *processor*. Perintah tersebut dikirimkan menggunakan NodeMCU yang sudah terhubung dengan *processor* menggunakan koneksi *wireless*. NodeMCU tersebutlah yang akan mengirimkan sinyal secara serial ke alat yang ada di area pekerjaan. Ilustrasi dari proses tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1 – Diagram ilustrasi proses

Untuk mengetes produk, sebuah maket yang dapat menyimulasikan area pekerjaan dalam pabrik. Maket terdiri dari beberapa komponen penting seperti manekin operator serta APD-nya berupa topi pelindung, *conveyor belt*, *industrial pilot lamp*, *emergency button*, *on switch*, dan sebuah tombol operasi. Jika digabungkan dengan produk utama, NodeMCU juga terletak pada maket untuk mengendalikan *industrial pilot lamp*, dan *conveyor belt*. Konsep dari maket beserta penjelasan isi dari maket dapat dilihat pada gambar Gambar 2.2.



Gambar 2.2– Konsep model maket

Maket dibagi menjadi dua tingkat. Tingkat pertama merupakan tingkat tempat simulasi beserta tombol-tombol dan lampu indikator. Tingkat kedua merupakan area elektrik. Area simulasi berisikan operator, area operasi, kamera, lampu indikator, sebuah *conveyor belt* yang digerakkan oleh *motor stepper*, dan tombol operasi. Lampu indikator dan *motor stepper* dikendalikan oleh NodeMCU yang terletak pada tingkat kedua. Kemudian, *power supply* pada tingkat kedua akan disambungkan dengan sumber tegangan PLN untuk memberikan daya ke semua komponen pada maket. Maket juga dilengkapi sebuah *emergency button* dan MCB sebagai pengaman. Hasil pendeteksian kamera yang dikirim melalui Wi-Fi akan diolah oleh *processor* yang kemudian mengirimkan sinyal ke maket berdasar hasil pendeteksian. Sinyal tersebut diterima oleh NodeMCU kemudian ditranslasikan menjadi sebuah perintah untuk maket.

2.2 Spesifikasi Sistem

2.2.1 Sistem Image Processing

Spesifikasi dari sistem *image processing* yang telah dibuat adalah:

1. Memiliki nilai mAP50 dan mAP50-95 sebesar 65% atau 0.65
2. Memiliki nilai F1-Score sebesar 80% atau 0.80
3. Memiliki toleransi ROI di bawah 0.5 centimeter
4. Memiliki nilai *frame per second* (FPS) di atas 0.5.

Dalam *Image Processing* atau *computer vision* secara generik, mAP dan F1-Score digunakan untuk menentukan akurasi dan presisi dari sebuah algoritma.

Mean Average Precision (mAP) adalah metrik utama untuk mengevaluasi performa model deteksi objek, yang dihitung sebagai rata-rata dari nilai *Average Precision* (AP) untuk setiap kelas objek. Perbedaan mAP50 dan mAP50-95 terletak pada *Intersection over Union* (IoU) yang digunakan untuk menentukan apakah sebuah deteksi dianggap benar.

mAP50 mengukur dengan menetapkan IoU sebesar 0.5. Sehingga sebuah deteksi dianggap valid jika area pendeteksian *overlap* dengan kotak batas objek asli adalah 50% atau lebih.

Sementara itu, mAP50-95 menghitung nilai mAP pada sepuluh batas IoU yang berbeda, mulai dari 0.5 hingga 0.95 dengan interval 0.05, lalu mengambil nilai rata-rata dari kesepuluh hasil tersebut. Oleh karena itu, mAP50-95 menilai kemampuan model untuk mendeteksi objek dan menguji lokalisasi objek dengan presisi *bounding box* yang sangat tinggi.

Skor-F1 (F1-Score) merupakan metrik evaluasi yang menggabungkan *precision* dan *recall* untuk menunjukkan performa model. Presisi mengukur seberapa akurat deteksi yang dibuat oleh model dari total deteksi yang ada, sementara *Recall* mengukur seberapa lengkap model dalam menemukan semua objek yang seharusnya terdeteksi. Skor-F1 dihitung sebagai rata-rata harmonik dari kedua nilai tersebut, sehingga memberikan bobot yang seimbang. Model hanya

akan mendapatkan skor F1 yang tinggi karena sebuah jika nilai Presisi dan *Recall*-nya sama-sama tinggi.

Nilai 65% dalam mAP tersebut bersumber dari rata-rata akurasi YOLOv11-Segmentation ketika berhadapan dengan data set Common Objects in Context (COCO) yang menghasilkan standar nilai mAP 50-95 0.65. Semakin besar nilai mAP dan F1 Score, maka semakin besar presisi dan akurasi sistem. Nilai toleransi ROI didapatkan dari besar model pekerja yang digunakan pada maket. Model pekerja memiliki panjang telapak kaki sebesar 1 cm, sehingga dapat disimpulkan pekerja sudah berada di luar area ROI ketika melewati 0.5 cm dari batas garis ROI.

2.2.2 Maket

Spesifikasi dari maket yang telah dibuat adalah:

1. Menerima sinyal dari komputer
2. Meneruskan sinyal tersebut menjadi perintah terhadap mesin yang terdapat pada maket
3. Menyalakan lampu hijau, kuning, atau merah berdasarkan sinyal dari komputer
4. Mendapatkan nilai *delay* di bawah 1 detik.

Maket yang dibuat bisa menyimulasikan kondisi pada sebuah *factory floor*, menerima sinyal dari produk utama atau prosesor, dan kemudian menjalankan perintah ke mesin dan lampu indikator berdasarkan sinyal yang diterima. Secara keseluruhan, proses tersebut harus memiliki kecepatan di bawah satu detik supaya menjadi produk yang *viable* untuk digunakan dalam rantai pabrik aslinya.

2.2.3 Standarisasi

Berdasarkan standari, berikut adalah spesifikasi sistem yang digunakan oleh proyek secara keseluruhan:

1. SNI ISO 3873, standar safety helmet sebagai APD.
2. ISO 12087, mengenai standar penggunaan arsitektur image processing.

Berdasarkan fungsi serta spesifikasi dari produk, pengguna dapat menggunakan produk selama 24 jam, atau jika mengikuti aturan *shift* dari pabrik, maka sekitar 16 jam dengan 8 jam per *shift*. *Overheating* menjadi kendala utama dalam menjalankan *processor*. Oleh karena itu, meskipun produk dapat berjalan terus menerus, setidaknya produk dimatikan jika sudah tidak ada orang lagi di area pabrik. Mengikuti *downtime* dari pabrik dengan sistem dua *shift*, maka produk memiliki sekitar 8 jam untuk beristirahat. Waktu tersebut sudah cukup untuk mengurangi risiko *overheating*. Kemudian untuk menjaga kondisi *processor*, maka sebulan sekali sebaiknya dibersihkan secara mendalam, terutama kipas ataupun *exhaust* dari mesin *processor*.

MTTF (*mean time to failure*) dari produk utama diukur dengan komponen pada produk yang mudah rusak yakni kamera. Kamera yang digunakan merupakan ESP32Cam yang sangat rentan terhadap panas. Berdasarkan manual ESP32, dianjurkan untuk tidak menggunakan ESP32Cam selama 16 jam berturut-turut. Dapat diasumsikan bahwa penggunaan maksimal berturut-turut adalah 32 jam dua kali dari anjuran jam pemakaian yang menyebabkan kerusakan fatal.

MTTR (*mean time to repair*) produk didasarkan pada perubahan kamera yang digunakan produk. Penggantian kamera hanya memerlukan pemasangan ulang kamera yang cukup mudah dilakukan, sehingga diperkirakan memerlukan *downtime* sekitar tiga puluh menit.

2.2.4 Constraint Produk

Spesifikasi produk yang didasarkan pada hambatan adalah sebagai berikut:

- Biaya material sistem dan maket tidak lebih dari 3 juta.
- Biaya produk tidak menghitung *processor* yang merupakan sebuah laptop.
- Dimensi dari maket adalah 70 x 60 x 25 cm.
- Maksimal operator yang dapat terdeteksi adalah satu orang.
- Mesin yang dikendalikan hanya satu.
- Area operasi sudah ditentukan sebelumnya dan hanya ada satu.
- APD yang digunakan operator hanyalah topi pengaman.

2.3 Metode Verifikasi Spesifikasi

Pengujian keberhasilan produk Pengujian-pengujian berikut dilaksanakan dengan menggunakan maket yang dibuat. Berikut merupakan kedua pengujian tersebut:

2.3.1 Pengujian F1 Recall

F1-Recall dapat diuji menggunakan metode manual ataupun otomatis melalui Ultralytics. Untuk percobaan ini, digunakan metode manual sebagai suatu tumpuan standar untuk percobaan mAP yang hanya bisa dilakukan secara otomatis. F1-Recall diuji berdasarkan objek yang kemudian didapatkan dengan menghitung *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN) dan *False Negatives* (FN). Dalam percobaan ini, pengukuran yang digunakan merupakan kebenaran mendeteksi operator yang sedang di area operasi (*true positives*), dan operator yang sedang tidak ada dalam operasi (*true negatives*). Rumus F1-Recall dapat dilihat pada Persamaan (1).

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (1)$$

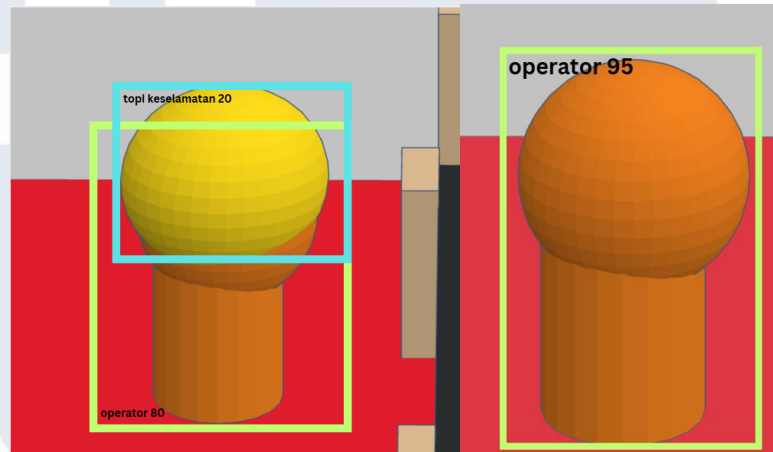
Sementara *precision* dan *recall* dapat dilihat pada Persamaan (2) dan (3);

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Untuk mendapatkan nilai TP, FP, TN dan FN, dibuatkan sebuah data set khusus yang berisikan gambar dengan kondisi berbeda. Program dijalankan terhadap data set tersebut, kemudian penguji memberikan nilai berdasarkan kondisi gambar data set yang dibandingkan dengan hasil pendeteksian. Nilai TP didapatkan ketika gambar pendeteksian berhasil mendeteksi kelas yang ada pada gambar secara akurat. Nilai FP didapatkan ketika gambar pendeteksian mendeteksi kelas pada objek yang tidak termasuk dalam kelas tersebut. Nilai TN didapatkan ketika gambar pendeteksian tidak mendeteksi objek yang tidak ada pada gambar. Nilai FN

didapatkan ketika gambar pendeteksian mendeteksi objek yang semestinya tidak ada pada gambar. Nilai F1 Score yang diharapkan dalam proyek adalah 0.8. Contoh TP dan TN dapat dilihat pada Gambar 2.3 yang menunjukkan perbandingan kondisi TP dan TN dari objek kelas topi keselamatan.



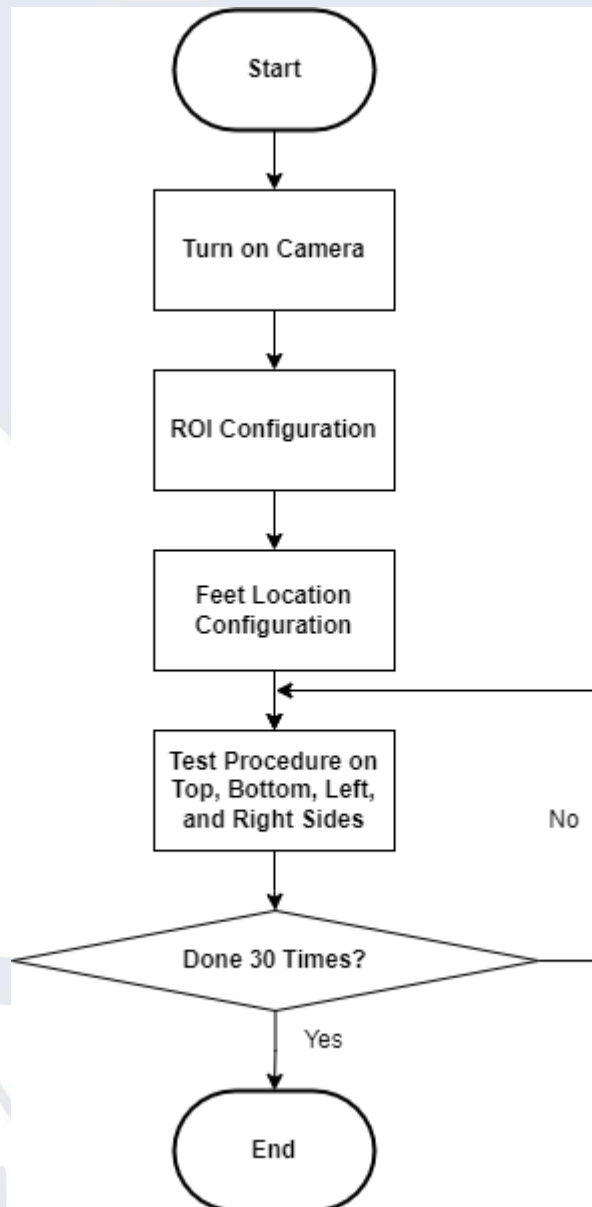
Gambar 2.3 – Ilustrasi percobaan pertama dengan kondisi *true positive* dan *true negative* dengan objek topi sebagai contoh.

2.3.2 Pengujian mAP

Nilai mAP merupakan nilai yang digunakan untuk melihat seberapa *reliable* sebuah algoritma. Pengujian mAP kemudian dilakukan dengan menyiapkan sebuah data set baru berisikan 800 gambar terbagi dari gambar TP dan TN setiap kelas yang tidak digunakan sebagai data set latihan. Data set baru tersebut juga diberikan label untuk menjadi *ground truth* sehingga model dapat mendeteksi secara internal bahwa pendeteksian benar atau salah. Pengujian mAP kemudian dilakukan menggunakan algoritma versi N, S dan M melalui fitur *val* untuk mendapatkan data yang akan digunakan untuk membandingkan metrik performa seperti mAP50, mAP50-95, dan kecepatan algoritma tersebut. Nilai mAP50 dan mAP50-95 yang menjadi target dalam pembuatan produk adalah di atas 0.65. Nilai tersebut diambil berdasarkan nilai rata-rata algoritma YOLOv11-seg terhadap data set *Common Object in Context* (COCO) yang dilakukan oleh Ultralytics dan tersedia pada dokumentasi algoritma YOLOv11.

2.3.3 Pengujian Toleransi ROI

ROI dibuat secara virtual untuk menandakan area spesial dalam *computer vision*. ROI merupakan salah satu fungsi yang muncul dari *library* OpenCV. Karena kamera berada dipasangkan dengan sudut pada tembok, maka perlu ditetapkan juga lokasi kaki dengan menggunakan fitur *point* dalam OpenCV. Gambar 2.4 merupakan gambar *flowchart* untuk proses pengecekan ROI.



Gambar 2.4 – Flowchart Proses Pengetesan Jarak

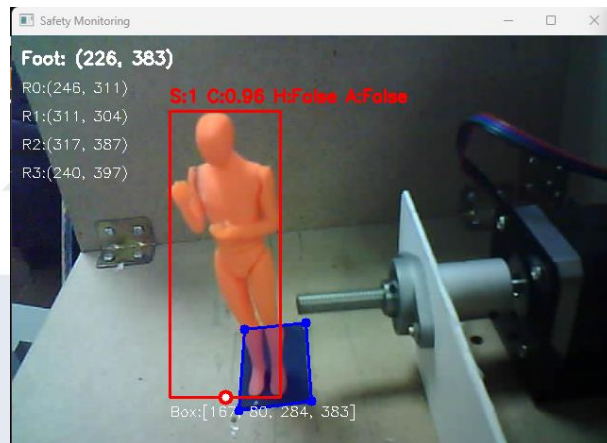
Untuk menentukan ROI pada *interface*, kode python dengan argumen tambahan `–calibrate-roi` digunakan. Kode tersebut akan memunculkan sebuah *window* tampilan kamera yang bisa digunakan untuk menentukan ROI.



Gambar 2.5 – Pemilihan Titik ROI

Pengaturan ROI dilakukan dengan cara klik pada titik area yang diinginkan seperti pada Gambar 2.5 sehingga kode dapat merekam lokasi kursor dan menentukan koordinat titik yang dipilih. Koordinat-koordinat tersebut disimpan di dalam sebuah *file* konfigurasi yang akan dipanggil oleh program jika argumen `–calibrate-roi` tidak dipanggil sehingga pengguna tidak perlu melakukan kalibrasi area ROI setelah menggunakan `–calibrate-roi`. Pemilihan koordinat ROI harus menggunakan *pattern clockwise* ataupun *counter-clockwise* seperti pada Gambar 2.5 untuk menghasilkan ROI berbentuk kotak. Hasil pembacaan setelah kalibrasi ROI dengan argumen `–debug` dapat dilihat pada Gambar 2.6.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

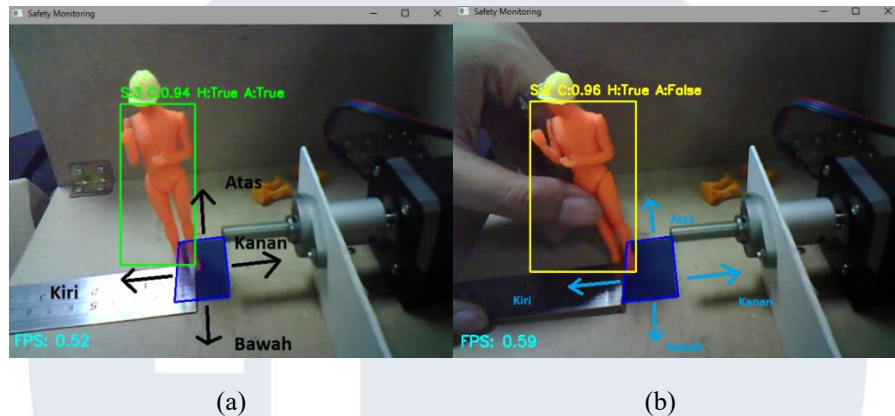


Gambar 2.6 – Hasil pembacaan Image Processing setelah kalibrasi area ROI.

Pada Gambar 2.6, R0-R3 merupakan koordinat ROI yang diambil dari pemilihan titik ROI. R0 memiliki kepanjangan *Region 0*, demikian juga R1 dan seterusnya. [Box] merupakan koordinat yang digunakan untuk menentukan lokasi pendeteksian pekerja. (Foot) merupakan koordinat letak kaki. S1 merupakan pembacaan *status* dari pekerja, 1 merupakan tidak aman sama sekali (tidak menggunakan topi pengaman dan tidak berada di area), 2 merupakan sedikit aman (menggunakan topi atau berada di area), dan 3 merupakan aman (menggunakan topi pengaman dan berada di area). C merupakan hasil pembacaan *confidence*. H merupakan pembacaan apakah pekerja menggunakan topi atau tidak. Penentu penggunaan topi oleh pekerja diatur oleh algoritma YOLO yang dilatih untuk mengetahui apakah topi sedang digunakan atau tidak. A merupakan pembacaan apakah pekerja berada di dalam area atau tidak. Penentu apakah pekerja berada di dalam area atau tidak bergantung pada koordinat letak kaki.

Pengujian dilakukan dengan cara menaruh model pekerja pada set poin tertentu yang sudah ditentukan. Poin-poin tersebut terletak di dalam area. Lebih tepatnya 4 poin yang direncanakan terletak pinggiran area. Kemudian model dari pekerja dipindahkan keluar dari titik pinggiran tersebut untuk melihat toleransi jarak kondisi pekerja yang berada di dalam area dan tidak ada di area tersebut. Pengukuran dilakukan dengan menggunakan penggaris untuk menentukan batas toleransi pendeteksian model pada area yang sudah ditentukan. Percobaan tersebut

dilakukan sebanyak 30 kali untuk melihat batas toleransi ROI. Ilustrasi dari percobaan tersebut dapat dilihat pada Gambar 2.7.



Gambar 2.7 – Contoh prosedur percobaan untuk menghitung toleransi sisi kiri dari area dari (a) titik awal pendeteksian (b) titik akhir dari pendeteksian.

Target dari nilai toleransi yang didapatkan adalah di bawah 5 milimeter. Nilai tersebut didapatkan karena besar telapak kaki model pekerja yang memiliki panjang sebesar 1 centimeter. Dalam penerapan program di dunia nyata, misalkan saja seorang pekerja memiliki besar telapak kaki sepanjang 20 centimeter, maka secara ideal, pekerja tersebut sudah terdeteksi keluar dari area ketika kakinya 10 centimeter keluar dari area tersebut.

2.3.4 Pengujian Integrasi Sistem

Pengujian dilakukan dengan cara mengaktifkan kamera dan melihat apakah komputer dapat memproses gambar yang dikirim dengan algoritma yang sudah dibuat. Pengujian melihat respons pada kapabilitas sistem untuk memberikan respons terhadap kondisi-kondisi yang mungkin terjadi menggunakan skenario maket. Skenario yang disiapkan bervariasi dalam tingkat pencahayaan di area percobaan, pose dan posisi model pekerja, penggunaan topi pekerja, dan warna topi dari pekerja. Pengujian kemudian melihat kecepatan *image processing* yang dilakukan algoritma. Kecepatan dari proses serta penampilannya akan dicatat sebanyak 30 kali untuk melihat rata-rata kecepatan proses pada produk. Pengujian ini menargetkan sistem memiliki rata-rata 0.5 FPS per detik untuk menghasilkan gambar atau tampilan yang tidak terlalu memiliki *delay*.