

# DETEKSI KERUSAKAN CONVEYOR BELT REAL TIME BERBASIS YOLOV11 EDGETPU PADA RASPBERRY PI 5

Muhammad Nur Bijak Bestari\*<sup>1</sup>, Mulki Indana Zulfa<sup>1</sup>

<sup>1</sup>Program Studi Teknik Elektro, Universitas Jenderal Soedirman, Indonesia

\*e-mail: [muhammad.bestari@mhs.unsoed.ac.id](mailto:muhammad.bestari@mhs.unsoed.ac.id)

## Abstrak

Conveyor belt merupakan komponen vital dalam industri semen yang berfungsi sebagai media transportasi material secara kontinu. Kerusakan pada conveyor belt seperti *tear*, *hole*, *patch work*, dan *puncture* yang tidak terdeteksi sejak dini dapat menyebabkan *downtime* produksi, peningkatan biaya perawatan, serta risiko keselamatan kerja. Metode inspeksi visual manual yang umum digunakan memiliki keterbatasan dari sisi waktu, ketelitian, dan subjektivitas operator. Penelitian ini bertujuan untuk mengembangkan sistem deteksi kerusakan conveyor belt secara *real time* berbasis *deep learning* yang dapat dijalankan pada perangkat *edge* dengan sumber daya terbatas. Sistem dikembangkan menggunakan model YOLOv11 yang dilatih dengan dataset gabungan hasil inspeksi lapangan, dataset publik, serta data hasil pengambilan otomatis, dengan total lebih dari 3.000 citra hasil augmentasi. Model terlatih dikonversi ke format TensorFlow Lite dan dikompilasi untuk Google Coral EdgeTPU agar dapat diimplementasikan pada Raspberry Pi 5 yang terhubung dengan kamera IP RTSP. Hasil pelatihan menunjukkan performa model dengan nilai *precision* sebesar 0,742, *recall* 0,730, *mAP@50* 0,764, dan *mAP@50–95* 0,553. Implementasi sistem di lingkungan industri menunjukkan kinerja inferensi yang stabil dengan latensi rendah serta kemampuan deteksi yang baik, khususnya pada kelas *patch work*. Sistem yang dikembangkan mampu melakukan pemantauan kondisi conveyor belt secara otomatis dan berkelanjutan, sehingga berpotensi mendukung pengurangan *downtime* dan peningkatan efisiensi pemeliharaan di lingkungan industri.

**Kata kunci:** conveyor belt; EdgeTPU; Google Coral; Raspberry Pi 5; YOLOv11

## Abstract

Conveyor belts play a critical role in the cement industry as continuous material transportation systems. Undetected damage such as *tear*, *hole*, *patch work*, and *puncture* can lead to production *downtime*, increased maintenance costs, and safety risks. Conventional manual visual inspections are limited in terms of time efficiency, accuracy, and operator subjectivity. This study aims to develop a real-time conveyor belt damage detection system based on deep learning that can be deployed on resource-constrained edge devices. The proposed system employs a YOLOv11 model trained on a combined dataset consisting of field inspection data, public datasets, and automatically collected images, resulting in more than 3,000 augmented training samples. The trained model is converted into TensorFlow Lite format and compiled for Google Coral EdgeTPU to enable efficient deployment on a Raspberry Pi 5 integrated with an RTSP-based IP camera. Training results demonstrate promising performance, achieving a precision of 0.742, recall of 0.730, *mAP@50* of 0.764, and *mAP@50–95* of 0.553. Implementation in an industrial environment shows stable real-time inference with low latency and reliable detection performance, particularly for the *patch work* class. The developed system enables continuous and automated conveyor belt condition monitoring, offering potential benefits in reducing *downtime* and improving maintenance efficiency in industrial applications.

**Keywords:** conveyor belt; EdgeTPU; Google Coral; Raspberry Pi 5; YOLOv11

## 1. PENDAHULUAN

Conveyor belt merupakan komponen penting dalam sistem transportasi material pada industri manufaktur dan pertambangan yang beroperasi secara kontinu. Kondisi conveyor belt yang tidak terpantau dengan baik dapat menyebabkan penurunan kinerja sistem, peningkatan biaya perawatan, serta gangguan terhadap keselamatan kerja. Ketidakpastian dalam penilaian kondisi conveyor belt, khususnya apabila hanya mengandalkan indikator berbasis waktu dan inspeksi manual, dapat menurunkan akurasi dalam pengambilan keputusan pemeliharaan [1]. Pada lingkungan industri, inspeksi kondisi conveyor belt umumnya masih dilakukan secara visual oleh operator dengan interval tertentu. Metode ini memiliki keterbatasan dalam mendeteksi kerusakan awal seperti *tear*, *hole*, *patch work*, dan *puncture*, terutama ketika kerusakan berukuran kecil atau berada pada area yang sulit dijangkau. Keterbatasan tersebut dapat menyebabkan kerusakan berkembang menjadi kegagalan yang lebih serius sebelum tindakan perbaikan dilakukan [2].

Perkembangan teknologi *machine vision* dan *deep learning* memungkinkan penerapan sistem inspeksi otomatis berbasis citra digital untuk mendeteksi kerusakan conveyor belt secara lebih akurat dan konsisten. Pendekatan *object detection* berbasis *convolutional neural network* (CNN), khususnya keluarga algoritma YOLO,

telah banyak digunakan dalam aplikasi industri karena kemampuannya melakukan deteksi objek secara cepat dan *real time* [3] [4]

Berbagai penelitian sebelumnya telah mengkaji metode deteksi kerusakan conveyor belt dengan pendekatan yang beragam. Guo *et al.* [2] melakukan kajian komprehensif terhadap metode deteksi robekan conveyor belt, mencakup metode berbasis sensor, *X-ray*, dan visi komputer. Meskipun metode non-destruktif seperti *X-ray* dan sensor magnetik mampu mendeteksi kerusakan internal, pendekatan tersebut membutuhkan perangkat khusus dengan biaya tinggi serta instalasi kompleks, sehingga kurang praktis untuk implementasi luas di lingkungan industri. Metode berbasis visi komputer dinilai lebih fleksibel, namun tantangan utama masih terletak pada keandalan sistem dan efisiensi komputasi. Pendekatan berbasis pembelajaran mendalam mulai banyak digunakan untuk meningkatkan akurasi deteksi. Liu *et al.* [3] mengusulkan model YOLO-STOD berbasis YOLOv5 dengan penambahan mekanisme atensi untuk meningkatkan deteksi robekan berukuran kecil pada lingkungan tambang. Hu *et al.* [4] mengembangkan metode YOLO yang dimodifikasi dengan struktur ringan untuk mencapai keseimbangan antara kecepatan dan akurasi deteksi secara *real time*. Meskipun hasil yang diperoleh menunjukkan performa yang tinggi, sebagian besar penelitian tersebut masih berfokus pada pengembangan arsitektur model dan pengujian di lingkungan komputasi dengan sumber daya relatif besar, tanpa pembahasan mendalam terkait kesiapan implementasi pada perangkat *edge* dengan keterbatasan komputasi dan konsumsi daya. Di sisi lain, pendekatan pemantauan kondisi conveyor belt berbasis indikator waktu dan sistem diagnostik non-visual masih memiliki tingkat ketidakpastian yang cukup tinggi. Rzeszowska *et al.* [1] menunjukkan bahwa metode penilaian kondisi conveyor belt berbasis indikator waktu dan data historis memiliki keterbatasan dalam mendeteksi kerusakan aktual secara langsung, sehingga berpotensi menunda tindakan pemeliharaan. Hal ini memperkuat kebutuhan akan sistem deteksi berbasis visual yang mampu memberikan informasi kondisi conveyor belt secara langsung dan *real time*[5].

Berdasarkan tinjauan tersebut, masih terdapat research gap pada penerapan sistem deteksi kerusakan conveyor belt berbasis pembelajaran mendalam yang tidak hanya berfokus pada peningkatan akurasi model, tetapi juga mempertimbangkan aspek implementasi nyata pada perangkat *edge* dengan sumber daya terbatas. Oleh karena itu, penelitian ini mengusulkan sistem deteksi kerusakan conveyor belt *real time* berbasis YOLOv11 yang diimplementasikan pada Raspberry Pi 5 dengan akselerasi Google Coral EdgeTPU. Pendekatan ini bertujuan untuk menjembatani kesenjangan antara performa deteksi yang tinggi dan kesiapan implementasi sistem di lingkungan industri secara langsung. Struktur penulisan paper ini disusun sebagai berikut. Bagian kedua menjelaskan metode penelitian yang mencakup arsitektur sistem, dataset, model deteksi, dan metrik evaluasi. Bagian ketiga menyajikan hasil penelitian berupa performa pelatihan model dan hasil inferensi *real time*. Bagian keempat membahas analisis hasil serta implikasi penerapan sistem di lingkungan industri. Bagian kelima menyimpulkan hasil penelitian dan menyampaikan arah pengembangan selanjutnya.

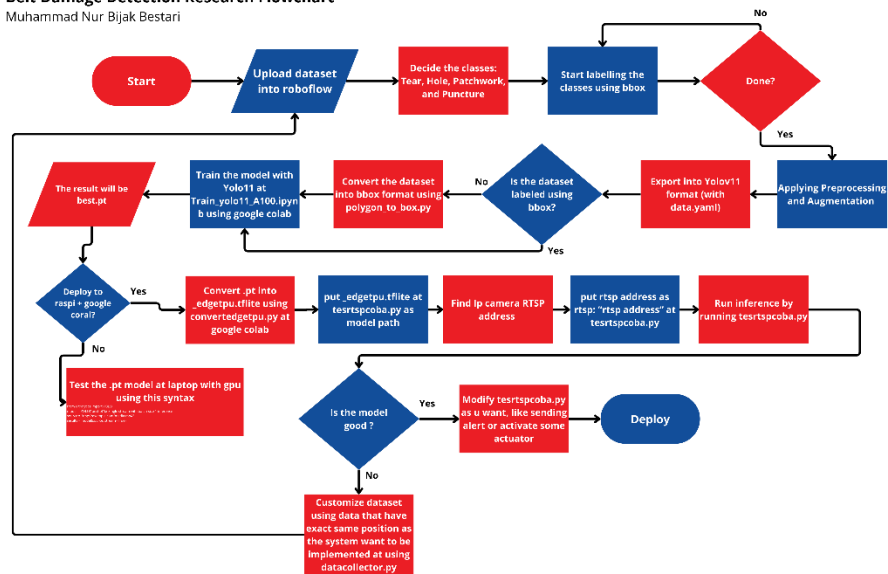
## 2. METODE

Metode penelitian pada studi ini disusun untuk mengembangkan dan mengevaluasi sistem deteksi kerusakan conveyor belt berbasis *deep learning* yang dapat beroperasi secara *real time* pada perangkat *edge*. Tahapan penelitian meliputi perancangan sistem, pengumpulan dan pengolahan dataset, pelatihan model deteksi, konversi model untuk inferensi *edge*, serta pengujian kinerja sistem di lingkungan industri perusahaan semen terkemuka di Jawa Barat.

### 2.1 Alur Metodologi Penelitian

Alur metodologi penelitian diawali dengan proses pengumpulan dataset yang berasal dari inspeksi lapangan, dataset publik, serta data hasil pengambilan otomatis. Dataset tersebut kemudian dianotasi dan diproses sebelum digunakan dalam tahap pelatihan model YOLOv11. Model terlatih selanjutnya dikonversi ke format yang kompatibel dengan Google Coral EdgeTPU dan diimplementasikan pada Raspberry Pi 5 untuk pengujian inferensi *real time*[6] [7]. Tahapan akhir penelitian mencakup evaluasi performa sistem berdasarkan metrik akurasi dan kecepatan inferensi seperti yang tertera pada Gambar 1.

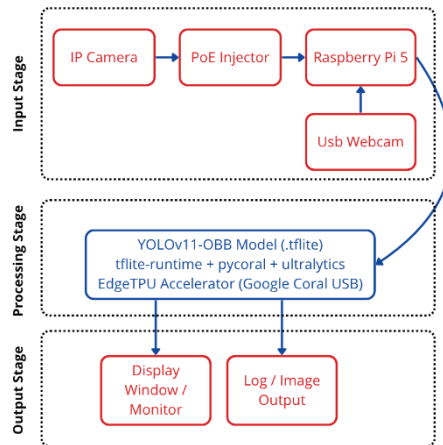
**Belt Damage Detection Research Flowchart**  
Muhammad Nur Bijak Bestari



Gambar 1 Flowchart Metodologi Penelitian

## 2.2 Arsitektur Sistem

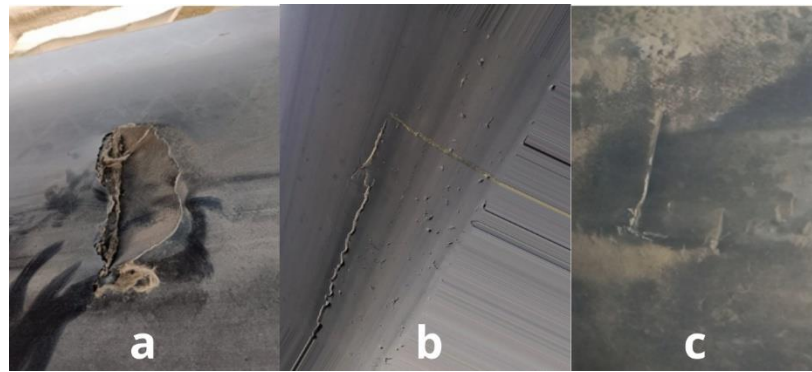
Arsitektur sistem dirancang seperti pada Gambar 2, berbasis *edge computing*, di mana proses inferensi dilakukan langsung pada perangkat Raspberry Pi 5 tanpa ketergantungan pada server eksternal. Sistem menggunakan kamera IP berbasis RTSP sebagai sumber data video yang mengamati permukaan conveyor belt secara kontinu. Data video diproses oleh model YOLOv11 yang telah dikonversi ke format TensorFlow Lite dan dikompilasi untuk Google Coral EdgeTPU. Hasil deteksi ditampilkan dalam bentuk *bouding box* dan label kelas kerusakan secara *real time*.



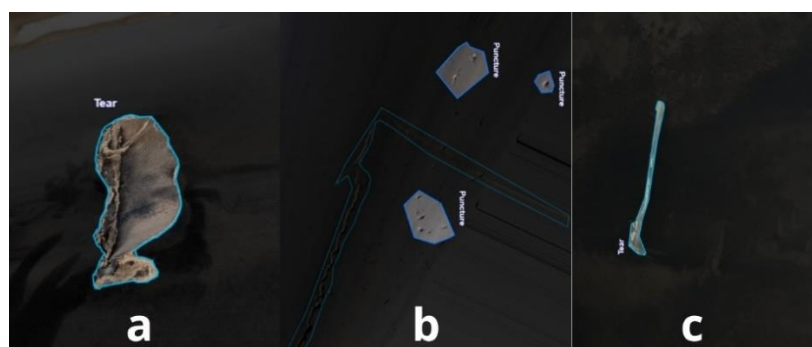
Gambar 2 Diagram Blok Arsitektur Sistem Deteksi Kerusakan Conveyor Belt

## 2.3 Dataset dan Proses Anotasi

Dataset yang digunakan dalam penelitian ini terdiri dari tiga sumber utama, yaitu data hasil inspeksi lapangan di Perusahaan semen terkemuka di Jawa Barat, dataset publik *conveyor belt damage*, serta data hasil pengambilan otomatis menggunakan program *data collector*. Dataset mencakup empat kelas kerusakan, yaitu *tear*, *hole*, *patch work*, dan *puncture*. Seluruh data dianotasi menggunakan format *bouding box* berorientasi untuk mendukung proses pelatihan model deteksi objek[8]. Dataset yang masih raw seperti pada Gambar 3 dilakukan proses labeling terlebih dahulu, hasilnya ada pada Gambar 4. Proses augmentasi data diterapkan untuk meningkatkan variasi dataset dan mengurangi risiko *overfitting* selama pelatihan model.



Gambar 3 Dataset Raw pada conveyor belt: (a) dataset inspeksi langsung, (b) dataset roboflow, (c) dataset sample dari datacollector.py



Gambar 4 Dataset yang sudah di label dengan segmentasi pada conveyor belt: (a) dataset inspeksi langsung, (b) dataset roboflow, (c) dataset sample dari datacollector.py

## 2.4 Model Deteksi YOLOv11

Model deteksi yang digunakan pada penelitian ini adalah YOLOv11 tipe *nano* (YOLOv11n) yang dipilih karena memiliki arsitektur ringan dan sesuai untuk implementasi pada perangkat *edge*[9]. Model ini mampu melakukan deteksi objek secara satu tahap (*single-stage detection*) sehingga mendukung kebutuhan inferensi *real time*. Pemilihan YOLOv11 didasarkan pada kemampuannya dalam meningkatkan keseimbangan antara akurasi dan kecepatan inferensi dibandingkan generasi YOLO sebelumnya, sebagaimana ditunjukkan pada penelitian deteksi kerusakan conveyor belt berbasis YOLO [3] [4].

## 2.5 Konversi Model dan Implementasi EdgeTPU

Model YOLOv11 yang telah dilatih pada lingkungan *training* dikonversi ke format TensorFlow Lite untuk mendukung inferensi pada perangkat *edge*. Selanjutnya, model TFLite dikompilasi menggunakan *EdgeTPU Compiler* agar dapat dijalankan pada Google Coral EdgeTPU. Proses ini bertujuan untuk mempercepat inferensi dengan memanfaatkan akselerasi perangkat keras, sehingga sistem dapat mencapai performa *real time* dengan latensi rendah[10]. Implementasi sistem dilakukan pada Raspberry Pi 5 yang terhubung langsung dengan kamera IP RTSP.

## 2.6 Parameter Pelatihan dan Pengujian

Pelatihan model dilakukan menggunakan dataset yang telah dianotasi dengan pembagian data latih dan data validasi. Parameter pelatihan meliputi ukuran *input image*, jumlah *epoch*, *batch size*, dan *learning rate* yang disesuaikan dengan kapasitas perangkat pelatihan. Pengujian inferensi dilakukan pada beberapa resolusi input untuk mengevaluasi pengaruh resolusi terhadap kecepatan dan stabilitas sistem pada perangkat *edge*[11].

## 2.7 Metrik Evaluasi

Evaluasi performa sistem dilakukan menggunakan metrik *precision*, *recall*, dan *mean Average Precision (mAP)*, yang umum digunakan pada penelitian deteksi objek [3] [4] [11]. Selain itu, kinerja sistem juga dievaluasi berdasarkan *frame per second (FPS)*, latensi inferensi, serta kestabilan sistem selama pengoperasian berkelanjutan.

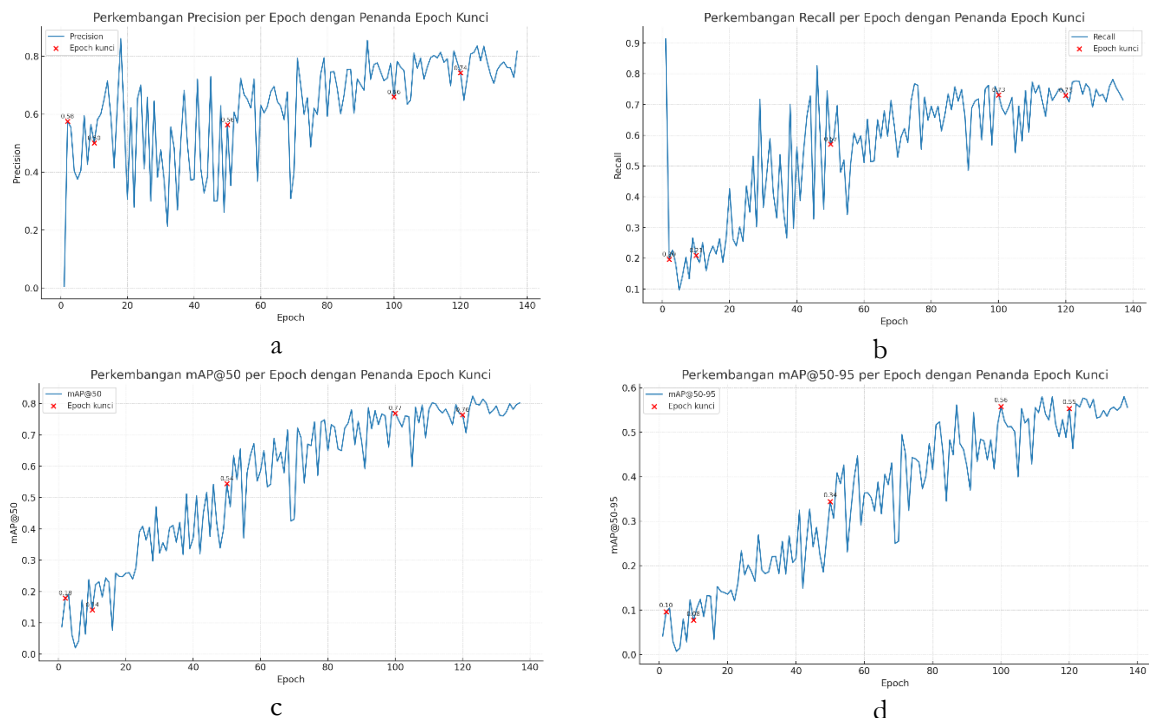
Pendekatan evaluasi ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya keseimbangan antara akurasi dan kecepatan inferensi dalam sistem deteksi kerusakan conveyor belt berbasis *machine vision* [5].

### 3. HASIL PENELITIAN

Bagian ini menyajikan hasil penelitian yang diperoleh dari proses pelatihan model, implementasi sistem pada perangkat *edge*, serta pengujian inferensi *real time* di lingkungan industri. Penyajian hasil dilakukan dalam bentuk grafik, tabel, dan dokumentasi visual sistem untuk memberikan gambaran kinerja model dan sistem secara menyeluruh[12].

#### 3.1 Hasil Pelatihan Model YOLOv11

Proses pelatihan model YOLOv11 dilakukan menggunakan dataset yang telah dianotasi dan melalui tahapan augmentasi data. Selama proses pelatihan, performa model dipantau menggunakan metrik precision, recall, dan mean Average Precision (mAP). Hasil pelatihan menunjukkan bahwa model mengalami peningkatan performa secara bertahap seiring bertambahnya jumlah epoch, yang mengindikasikan bahwa proses pembelajaran berjalan dengan baik dan stabil[13].

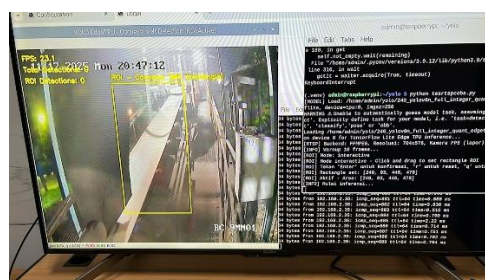


Gambar 5 Grafik Performa (a) Precision, (b) Recall, (c) mAP50, (d) Map50-95 Pelatihan Model YOLOv11

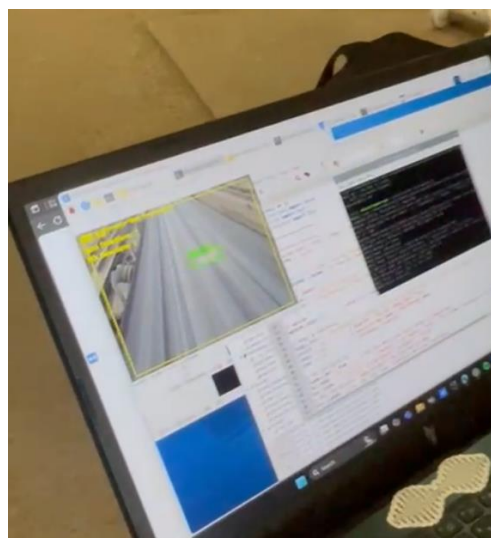
Grafik pada Gambar 5 menunjukkan bahwa nilai *precision* dan *recall* meningkat secara konsisten hingga mencapai kondisi konvergen. Nilai *mAP@50* yang diperoleh menunjukkan bahwa model memiliki kemampuan deteksi yang baik terhadap kelas kerusakan conveyor belt yang digunakan dalam penelitian ini[14].

#### 3.2 Hasil Inferensi Sistem di Lingkungan Industri

Pengujian inferensi dilakukan dengan mengimplementasikan model YOLOv11 yang telah dikonversi ke format TensorFlow Lite dan dikompilasi untuk Google Coral EdgeTPU pada Raspberry Pi 5. Sistem dihubungkan dengan kamera IP berbasis RTSP untuk melakukan pemantauan permukaan conveyor belt secara *real time* di lingkungan industri perusahaan semen terkemuka di Jawa Barat. Berdasarkan hasil inferensi yang ditampilkan pada Gambar 6 dan 7, sistem mampu mendeteksi kerusakan conveyor belt secara *real time* dengan tingkat kejelasan visual yang baik. *Bounding box* yang dihasilkan menunjukkan bahwa model dapat mengidentifikasi area kerusakan secara tepat, khususnya pada kelas *patch work* yang memiliki karakteristik visual paling dominan[8].



Gambar 6 Antarmuka Sistem



Gambar 7 Sistem Mendeteksi Patchwork

### 3.3 Kinerja Sistem Inferensi

Kinerja sistem dievaluasi berdasarkan kecepatan inferensi, latensi, serta stabilitas sistem selama pengoperasian berkelanjutan[15]. Pengujian dilakukan pada beberapa konfigurasi resolusi *input image* untuk mengetahui pengaruh resolusi terhadap performa sistem pada perangkat *edge*. Hasil pada Tabel 1 menunjukkan bahwa penggunaan model TFLite yang dikompilasi untuk EdgeTPU memberikan performa inferensi tertinggi dibandingkan format lainnya, khususnya pada resolusi *input*  $320 \times 320$  piksel. Sistem mampu mencapai kecepatan inferensi hingga puluhan FPS dengan latensi rendah, sehingga memenuhi kebutuhan pemantauan *real time* di lingkungan industri.

Tabel 1. Hasil Pengujian Kinerja Inferensi Sistem

Format Model	280px	320px	640px	Keterangan
PyTorch (.pt)	~20 FPS	~10 FPS	~3–7 FPS	FPS tinggi pada resolusi kecil, namun deteksi tidak muncul dengan baik; pada 640 px mulai muncul <i>bounding box</i> .
ONNX (CPU)	-	-	~20 FPS	Diuji dengan webcam dan dataset umum; deteksi objek manusia muncul



Format Model	280px	320px	640px	Keterangan
TFLite + Coral EdgeTPU	-	~30–40 FPS	~10 FPS	dengan label yang benar.  FPS tertinggi di resolusi 320 px, dengan hasil deteksi <i>Patch Work</i> cukup akurat.

### 3.4 Stabilitas Operasional Sistem

Selain kecepatan inferensi, sistem juga diuji dari sisi stabilitas operasional selama penggunaan berkelanjutan. Parameter yang diamati terlampir pada Tabel 2 meliputi suhu perangkat Raspberry Pi 5 dan Google Coral EdgeTPU, serta kestabilan aliran video RTSP selama proses inferensi berlangsung. Pengujian menunjukkan bahwa sistem dapat beroperasi secara stabil tanpa mengalami *frame drop* signifikan maupun peningkatan suhu yang berlebihan. Hal ini menunjukkan bahwa konfigurasi sistem yang digunakan telah sesuai untuk implementasi jangka panjang di lingkungan industri.

Tabel 2. Parameter Operasional Sistem Selama Pengujian

Parameter	Nilai	Keterangan
FPS tertinggi (EdgeTPU, 320 px)	30–40 FPS	RTSP substream, deteksi <i>Patch Work</i> stabil
FPS terendah (PyTorch, 640 px)	3–7 FPS	Inferensi lambat, <i>bounding box</i> mulai muncul
FPS (ONNX, 640 px)	~20 FPS	Diuji menggunakan webcam dan dataset umum
Latensi rata-rata	~70–90 ms	Antara input frame dan hasil output
Suhu RPi5	~55 ± 5 °C	Stabil dalam panel tertutup
Suhu Coral USB	~65 °C	Stabil selama inferensi berkelanjutan
Kestabilan RTSP	Stabil	Tidak ada <i>drop frame</i> yang signifikan

## 4. DISKUSI

Hasil penelitian menunjukkan bahwa sistem deteksi kerusakan conveyor belt berbasis YOLOv11 yang diimplementasikan pada perangkat *edge* mampu melakukan inferensi secara *real time* dengan performa stabil di lingkungan industri. Pencapaian ini menegaskan bahwa pendekatan *edge computing* dapat menjadi solusi efektif untuk mengatasi keterbatasan inspeksi manual dan sistem berbasis *server* dalam pemantauan kondisi conveyor belt secara berkelanjutan. Model YOLOv11 yang digunakan mampu mengidentifikasi beberapa jenis kerusakan conveyor belt dengan baik, khususnya pada kelas *patch work* yang memiliki karakteristik visual paling jelas. Hasil ini sejalan dengan temuan pada penelitian sebelumnya yang menyatakan bahwa metode berbasis YOLO efektif dalam mendeteksi kerusakan conveyor belt pada kondisi industri nyata [3] [4]. Dibandingkan dengan pendekatan deteksi berbasis indikator waktu yang memiliki tingkat ketidakpastian tinggi [1], sistem berbasis *machine vision* yang dikembangkan dalam penelitian ini memberikan hasil yang lebih konsisten karena langsung menganalisis kondisi visual permukaan belt.

Penggunaan Google Coral EdgeTPU terbukti mampu meningkatkan kecepatan inferensi secara signifikan dibandingkan eksekusi model tanpa akselerator perangkat keras. Hal ini mendukung penerapan sistem pada perangkat dengan sumber daya terbatas seperti Raspberry Pi 5, tanpa mengorbankan kebutuhan

*real time*. Temuan ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya efisiensi komputasi dalam sistem deteksi kerusakan conveyor belt berbasis visi komputer [5]. Selain itu, pemrosesan langsung di perangkat *edge* mengurangi ketergantungan terhadap koneksi jaringan, yang menjadi kendala utama pada sistem berbasis *cloud* di lingkungan industri. Meskipun demikian, hasil penelitian juga menunjukkan adanya keterbatasan sistem, terutama pada deteksi kerusakan dengan ukuran kecil atau kontras rendah terhadap latar belakang conveyor belt. Kondisi pencahayaan yang tidak seragam dan variasi tekstur permukaan belt dapat mempengaruhi keandalan deteksi, sebagaimana juga dilaporkan pada penelitian deteksi robekan conveyor belt di lingkungan pertambangan [2]. Keterbatasan ini menunjukkan bahwa kualitas dan keberagaman dataset memiliki peran penting dalam meningkatkan kemampuan generalisasi model.

Sistem yang dikembangkan telah menunjukkan potensi untuk digunakan sebagai alat pemantauan awal (*early detection*) kondisi conveyor belt pada industri semen skala besar di wilayah Jawa Barat. Integrasi sistem dengan kamera IP berbasis RTSP memungkinkan pemantauan otomatis tanpa mengganggu proses produksi yang sedang berjalan. Namun, untuk implementasi pada skala yang lebih luas, diperlukan pengembangan lanjutan, seperti integrasi dengan sistem peringatan dini atau sistem pemeliharaan terpusat agar hasil deteksi dapat dimanfaatkan secara optimal dalam pengambilan keputusan pemeliharaan. Secara keseluruhan, hasil diskusi ini menegaskan bahwa sistem deteksi kerusakan conveyor belt berbasis YOLOv11 pada perangkat *edge* merupakan pendekatan yang relevan dan aplikatif untuk lingkungan industri. Dibandingkan penelitian terdahulu yang berfokus pada peningkatan arsitektur model atau akurasi semata [3] [4] penelitian ini menekankan keseimbangan antara akurasi, kecepatan inferensi, dan kesiapan implementasi di lapangan, sehingga memberikan kontribusi praktis dalam penerapan *edge AI* untuk pemantauan kondisi conveyor belt.

## 5. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan sistem deteksi kerusakan conveyor belt *real time* berbasis YOLOv11 pada Raspberry Pi 5 dengan akselerasi Google Coral EdgeTPU untuk mendukung pemantauan kondisi conveyor belt di lingkungan industri. Sistem yang dikembangkan mampu melakukan inferensi secara stabil dengan latensi rendah serta menunjukkan kinerja deteksi yang baik terhadap berbagai jenis kerusakan, khususnya *patch work*, sehingga menjawab tujuan penelitian dalam menghadirkan solusi inspeksi otomatis yang efisien dan aplikatif. Integrasi model deteksi dengan kamera IP berbasis RTSP memungkinkan pemantauan berkelanjutan tanpa mengganggu proses operasional, sementara pemanfaatan akselerator EdgeTPU meningkatkan kecepatan inferensi pada perangkat dengan sumber daya terbatas. Sebagai pengembangan lanjutan, sistem ini dapat ditingkatkan melalui penambahan dan diversifikasi dataset untuk memperbaiki kemampuan generalisasi model pada berbagai kondisi pencahayaan dan jenis kerusakan, serta integrasi dengan sistem peringatan dini atau manajemen pemeliharaan terpusat agar hasil deteksi dapat dimanfaatkan secara langsung dalam pengambilan keputusan pemeliharaan di lingkungan industri. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa pendekatan *edge AI* yang diusulkan mampu menjembatani kebutuhan antara akurasi deteksi dan kesiapan implementasi sistem di lapangan.

## UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada institusi industri tempat pelaksanaan kerja praktik atas izin dan dukungan yang diberikan selama proses pengambilan data penelitian. Penulis juga mengucapkan terima kasih kepada institusi akademik yang telah memberikan bimbingan dan dukungan selama penelitian ini berlangsung.

## DAFTAR PUSTAKA

- [1] A. Rzeszowska, L. Jurdziak, R. Blażej, and P. Lewandowicz, "Analysis of Uncertainty in Conveyor Belt Condition Assessment Using Time-Based Indicators," *Applied Sciences*, vol. 15, no. 14, p. 7939, Jul. 2025, doi: 10.3390/app15147939.
- [2] X. Guo, X. Liu, H. Zhou, R. Stanislawski, G. Królczyk, and Z. Li, "Belt Tear Detection for Coal Mining Conveyors," *Micromachines (Basel)*, vol. 13, no. 3, p. 449, Mar. 2022, doi: 10.3390/mi13030449.



- [3] W. Liu, Q. Tao, N. Wang, W. Xiao, and C. Pan, "YOLO-STOD: an industrial conveyor belt tear detection model based on Yolov5 algorithm," *Sci Rep*, vol. 15, no. 1, p. 1659, Jan. 2025, doi: 10.1038/s41598-024-83619-6.
- [4] Y. Hu, Y. Zhang, J. Shen, and G. Wang, "Real-time conveyor belt damage detection method based on improved YOLO," *Engineering Research Express*, vol. 7, no. 2, p. 025513, Jun. 2025, doi: 10.1088/2631-8695/adc8fd.
- [5] X. Guo *et al.*, "Machine vision based damage detection for conveyor belt safety using Fusion knowledge distillation," *Alexandria Engineering Journal*, vol. 71, pp. 161–172, May 2023, doi: 10.1016/j.aej.2023.03.034.
- [6] R. Prokscha, M. Schneider, and A. Höß, "Efficient Edge Deployment Demonstrated on YOLOv5 and Coral Edge TPU," in *Industrial Artificial Intelligence Technologies and Applications*, New York: River Publishers, 2023, pp. 141–155. doi: 10.1201/9781003377382-11.
- [7] A. Ghosh, S. A. Al Mahmud, T. I. R. Uday, and D. Md. Farid, "Assistive Technology for Visually Impaired using Tensor Flow Object Detection in Raspberry Pi and Coral USB Accelerator," in *2020 IEEE Region 10 Symposium (TENSYP)*, IEEE, 2020, pp. 186–189. doi: 10.1109/TENSYP50017.2020.9230630.
- [8] J. Murrugarra-Llerena, L. N. Kirsten, and C. R. Jung, "Can We Trust Bounding Box Annotations for Object Detection?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2022, pp. 4813–4822.
- [9] H. Feng, G. Mu, S. Zhong, P. Zhang, and T. Yuan, "Benchmark Analysis of YOLO Performance on Edge Intelligence Devices," *Cryptography*, vol. 6, no. 2, p. 16, Apr. 2022, doi: 10.3390/cryptography6020016.
- [10] K.-F. Chen and D.-Y. Hong, "Rewriting Deep Learning Models for Maximizing Edge TPU Utilization," in *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, Jan. 2023, pp. 657–664. doi: 10.1109/ICPADS56603.2022.00091.
- [11] V. J. Reddi *et al.*, "MLPerf Inference Benchmark," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, IEEE, May 2020, pp. 446–459. doi: 10.1109/ISCA45697.2020.00045.
- [12] J. Meyer, "Performance with tables and graphs: Effects of training and a visual search model," *Ergonomics*, vol. 43, no. 11, pp. 1840–1865, 2000.
- [13] S. Afaq and S. Rao, "Significance of epochs on training a neural network," *Int. J. Sci. Technol. Res.*, vol. 9, no. 06, pp. 485–488, 2020.
- [14] M. Zhang, "An Improved Fire Detection Algorithm Based on YOLOv8 Integrated with DGICnv, FourBranchAttention and GSIOU," *HighTech and Innovation Journal*, vol. 5, no. 3, pp. 677–689, Sep. 2024, doi: 10.28991/HIJ-2024-05-03-09.
- [15] S. S. Ogden and T. Guo, "MDINFERENCE: Balancing Inference Accuracy and Latency for Mobile Applications," in *2020 IEEE International Conference on Cloud Engineering (IC2E)*, IEEE, Apr. 2020, pp. 28–39. doi: 10.1109/IC2E48712.2020.00010.