

**IMPLEMENTASI YOLOv4 DENGAN *FILTER PRE-PROCESSING*
GAMBAR UNTUK OPTIMALISASI SISTEM MONITORING KEAMANAN
RUMAH BERBASIS ESP32-CAM**



MUH SYAIFULLAH TAMSIR

D041201048



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**

**IMPLEMENTASI YOLOv4 DENGAN *FILTER PRE-PROCESSING*
GAMBAR UNTUK OPTIMALISASI SISTEM MONITORING KEAMANAN
RUMAH BERBASIS ESP32-CAM**

MUH. SYAIFULLAH TAMSIR

D041201048

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

program studi teknik elektro

Program Studi Teknik Elektro

Teknik Elektro

Fakultas Teknik

Universitas Hasanuddin

2024

LEMBAR PENGESAHAN SKRIPSI

**IMPLEMENTASI YOLOv4 DENGAN *FILTER PRE-PROCESSING* GAMBAR
UNTUK OPTIMALISASI SISTEM MONITORING KEAMANAN RUMAH
BERBASIS ESP32-CAM**

Disusun dan diajukan oleh

Muhammad Syaifullah Tamsir

D041201048

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Elektro
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 13 November 2024
Dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,



Azran Budi Arief, S.T., M.T
NIP. 198902012019031007

Ketua Program Studi,



Prof. Dr.-Ing./Ir. Faizal A Samman, M.T., IPU., ASEAN.Eng. ACPE.
NIP. 197506052002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Muhammad Syaifullah Tamsir

NIM : D041201048

Program Studi : Teknik Elektro

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

IMPLEMENTASI YOLOv4 DENGAN *FILTER PRE-PROCESSING* GAMBAR UNTUK OPTIMALISASI SISTEM MONITORING KEAMANAN RUMAH BERBASIS ESP32-CAM

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 13 November 2024

nyatakan



Muhammad Syaifullah Tamsir

D041201048

UCAPAN TERIMAKASIH

Penelitian yang saya lakukan dapat terlaksana dengan sukses dan diskripsi ini dapat terampung atas bimbingan, diskusi dan arahan Bapak Azran Budi Arief, S.T., M.T. selaku pembimbing, Bapak Prof. Dr. Ir. Andani Achmad, M.T. selaku penguji 1 dan Bapak Ir. Samuel Panggalo, M.T selaku penguji 2. Saya mengucapkan banyak terima kasih kepada mereka.

Ucapan terimakasih juga saya ucapkan kepada Bapak/Ibu Dosen yang telah memberikan ilmu pengetahuan yang sangat berharga serta seluruh Staff Departemen Teknik Elektro yang telah membantu memfasilitasi penulis selama menempuh Pendidikan di kampus ini.

Kepada kedua orang tua, saudara dan semua keluarga saya ucapkan terima kasih atas segala macam bentuk dukungan yang telah mereka berikan kepada saya selama menempuh perkuliahan dari awal sampai selesai. Saya juga mengucapkan terima kasih kepada teman teman dari teknik elektro angkatan 2020 yang saling memberi semangat dan berbagi suka duka dari awal perkuliahan hingga saat ini.

Penulis

Muh. Syaifullah Tamsir

ABSTRAK

MUH. SYAIFULLAH TAMSIR. **Implementasi YOLOv4 Dengan *Filter Pre-processing* Gambar Untuk Optimalisasi Sistem Monitoring Keamanan Rumah Berbasis ESP32-CAM** (dibimbing oleh Azran Budi Arief, S.T., M.T.)

Penelitian ini bertujuan untuk mengoptimalkan sistem keamanan rumah berbasis ESP32-CAM dengan mengintegrasikan YOLOv4 dan teknik *pre-processing* gambar (*Gaussian filter*, *Median filter*, *Histogram Equalization*, CLAHE) agar meningkatkan kemampuan deteksi objek dalam penggunaannya. Pengujian dilakukan dengan membandingkan kinerja YOLOv4 dengan dan tanpa *pre-processing* dalam hal *precision*, *recall*, dan *F1-score*, serta mengukur kualitas gambar melalui SNR, PSNR, MSE, dan SSIM. Sistem diuji secara *realtime* di berbagai kondisi (pagi, siang, malam) dan terintegrasi dengan aplikasi Telegram untuk notifikasi. Hasil menunjukkan bahwa YOLOv4 dengan *pre-processing* memiliki *precision* yang lebih tinggi (0.98) dibandingkan tanpa *pre-processing* (0.9425), meski *recall* lebih rendah (0.551). *Pre-processing* meningkatkan stabilitas deteksi, terutama dalam kondisi pencahayaan bervariasi, namun mengurangi *recall* dan *F1-score*. Pada pengujian notifikasi *realtime*, YOLOv4 dengan *pre-processing* memberikan waktu pengiriman yang lebih cepat, terutama pada jarak dekat hingga menengah (0.8-1.1 detik). PIR sensor terbukti kurang andal karena waktu pengirimannya lebih lambat (2-4 detik). Kesimpulannya, integrasi YOLOv4 dan *pre-processing* gambar secara signifikan meningkatkan performa deteksi dan efisiensi sistem keamanan rumah berbasis ESP32-CAM, terutama dalam memberikan notifikasi yang lebih cepat dan akurat pada berbagai kondisi pencahayaan, menjadikannya solusi yang lebih efektif untuk sistem keamanan modern.

Kata Kunci: YOLOv4, *Pre-processing* Gambar, ESP32-CAM

ABSTRACT

MUH. SYAIFULLAH TAMSIR. ***Implementation of YOLOv4 with Image Pre-processing Filters for Optimizing Home Security Monitoring System Based on ESP32-CAM*** (supervised by Azran Budi Arief, S.T., M.T.)

This research aims to optimize a home security system based on ESP32-CAM by integrating YOLOv4 and image pre-processing techniques (Gaussian filter, Median filter, Histogram Equalization, CLAHE) to enhance object detection capabilities. The testing involved comparing YOLOv4 performance with and without pre-processing in terms of precision, recall, and F1-score, as well as evaluating image quality using SNR, PSNR, MSE, and SSIM. The system was tested in real-time under various conditions (morning, afternoon, night) and integrated with the Telegram application for notifications. The results showed that YOLOv4 with pre-processing achieved higher precision (0.98) compared to without pre-processing (0.9425), although recall was lower (0.551). Pre-processing improved detection stability, especially under varying lighting conditions, but reduced recall and F1-score. In real-time notification testing, YOLOv4 with pre-processing delivered faster notification times, particularly at close to medium distances (0.8–1.1 seconds). The PIR sensor was found to be less reliable due to slower notification times (2–4 seconds). In conclusion, the integration of YOLOv4 and image pre-processing significantly enhances the detection performance and efficiency of the ESP32-CAM-based home security system, particularly in providing faster and more accurate notifications under diverse lighting conditions, making it a more effective solution for modern security systems.

Keywords: YOLOv4, Image Pre-processing , ESP32-CAM

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN PENGAJUAN.....	ii
HALAMAN PENGESAHAN PENGESAHAN	iii
PERNYATAAN KEASLIAN SKRIPSI	iv
UCAPAN TERIMAKASIH	v
ABSTRAK	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Ruang Lingkup	4
1.6 Studi Tedahulu	5
1.7 Teori.....	8
1.7.1 <i>Internet of Things</i> (IoT).....	8
1.7.2 Sistem Monitoring Keamanan Rumah.....	8
1.7.3 ESP32-CAM	9
1.7.4 <i>You Only Look Once</i> (YOLO).....	10
1.7.5 <i>Visual Studio Code</i>	12
1.7.6 Arduino IDE	12
1.7.7 Telegram	13
1.7.8 <i>Pre-processing</i> Gambar	14
1.7.9 <i>Metriks Performance</i> YOLO	15
1.7.10Kualitas Gambar.....	16
BAB II METODE PENELITIAN.....	19
2.1 Waktu dan Lokasi Penelitian	19
2.2 Variabel Penelitian/Perancangan	19
2.3 Bahan Uji dan Alat.....	20

2.4 Skema Sistem Kerja.....	22
2.5 Teknik Pengumpulan Data	26
2.6 Teknik Analisis Data.....	28
BAB III HASIL DAN PEMBAHASAN	29
3.1 Hasil Perancangan Alat.....	29
3.2 Metriks <i>Performance</i> Antara YOLOv4 Tanpa <i>Pre-processing</i> Gambar dan YOLOv4 Dengan <i>Pre-processing</i> Gambar.....	29
3.3 Kualitas Gambar Dengan <i>Pre-processing</i> Gambar	32
3.4 Pemakaian Secara <i>Realtime</i>	36
3.4.1 Pengujian Pemakaian Secara <i>Realtime</i> Dalam Ruangan	38
3.4.2 Pengujian Pemakaian Secara <i>Realtime</i> Diluar Ruang (Pagi).....	39
3.4.3 Pengujian Pemakaian Secara <i>Realtime</i> Diluar Ruang (Siang-Sore).....	40
3.4.4 Pengujian Pemakaian Secara <i>Realtime</i> Diluar Ruang (Malam)	41
BAB IV KESIMPULAN	43
4.1 Kesimpulan.....	43
4.2 Saran.....	43
DAFTAR PUSTAKA.....	44
LAMPIRAN.....	47

DAFTAR GAMBAR

Gambar 1 Bagaimana IoT bekerja	8
Gambar 2 Konfigurasi sistem monitoring rumah	9
Gambar 3 ESP32-CAM.....	10
Gambar 4 Arsitektur <i>You Only Look Once</i> (YOLO).....	11
Gambar 5 Logo <i>visual studio code</i>	12
Gambar 6 Logo software arduino IDE.....	13
Gambar 7 Logo aplikasi telegram	13
Gambar 8 Confusion matriks	16
Gambar 9 Diagram alir penelitian	22
Gambar 10 Blok sistem alat.....	23
Gambar 11 Skematik perancangan ESP32-CAM	23
Gambar 12 Ilustrasi rancangan hasil akhir alat	24
Gambar 13 Blok sistem pengolahan citra	25
Gambar 14 Diagram alir sistem	25
Gambar 15 Hasil akhir rancangan alat.....	29
Gambar 16 Gambar hasil uji <i>metriks performance</i> tanpa <i>filter pre-processing</i>	30
Gambar 17 Gambar hasil uji <i>metriks performance</i> dengan <i>filter pre-processing</i>	30
Gambar 18 Gambar <i>original</i> tanpa <i>filter</i>	32
Gambar 19 Gambar dengan <i>gaussian filter</i>	32
Gambar 20 Gambar dengan <i>median filter</i>	33
Gambar 21 Gambar dengan penerapan <i>histogram equalization</i>	33
Gambar 22 Gambar akhir setelah proses CLAHE	34
Gambar 23 Gambar pengujian kualitas gambar dari YOLOv4 dengan <i>filter pre-processing</i> dan tanpa <i>filter pre-processing</i>	34
Gambar 24 Gambar pengujian kalibrasi kamera ESP32-CAM	37

DAFTAR TABEL

Tabel 1 Spesifikasi ESP32-CAM.....	10
Tabel 2 Perangkat keras	20
Tabel 3 Perangkat lunak	21
Tabel 4 Perbandingan hasil <i>metriks performance</i>	31
Tabel 5 Perbandingan hasil kualitas gambar	35
Tabel 6 Hasil kalibrasi kamera ESP32-CAM.....	37
Tabel 7 Hasil pengujian secara <i>realtime</i> skenario dalam ruangan.....	38
Tabel 8 Hasil pengujian secara <i>realtime</i> skenario diluar ruangan (pagi).....	39
Tabel 9 Hasil pengujian secara <i>realtime</i> skenario diluar ruangan (siang-sore).....	40
Tabel 10 Hasil pengujian secara <i>realtime</i> skenario diluar ruangan (malam).....	41

DAFTAR LAMPIRAN

Lampiran 1 Skrip program pengukuran metrik <i>performance</i>	47
Lampiran 2 Skrip program parameter kualitas gambar	55
Lampiran 3 Skrip program untuk menjalankan pengujian secara <i>realtime</i>	58
Lampiran 4 Dokumentasi pengujian secara <i>realtime</i>	62

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah merupakan tempat berlindung yang melindungi penghuninya dari kondisi cuaca, ancaman luar, dan memberikan kenyamanan serta privasi. Selain sebagai tempat tinggal, rumah juga merupakan pusat kehidupan keluarga, tempat di mana orang beristirahat, berkumpul, dan menjalani aktivitas sehari-hari. Rumah memiliki peran penting dalam menciptakan rasa aman dan stabilitas bagi penghuninya. Keberadaan rumah yang aman dan nyaman tidak hanya berdampak pada kesehatan fisik tetapi juga kesejahteraan mental dan emosional. Dalam konteks masyarakat modern, rumah menjadi simbol pencapaian dan investasi jangka panjang, yang mana setiap individu berupaya untuk menjaga dan melindunginya dari berbagai ancaman dan gangguan eksternal.

Pada zaman ini keamanan rumah merupakan aspek yang diperhatikan oleh masyarakat karena berbagai faktor yang dapat mempengaruhi kehidupan sehari-hari dan keselamatan individu. Salah satu alasan utama keamanan rumah menjadi penting yaitu peningkatan tingkat kejahatan pencurian dan perampokan yang marak terjadi. Pencurian rumah merupakan salah satu jenis kejahatan properti yang paling sering dilaporkan dengan ribuan kasus yang dilaporkan setiap tahun di berbagai negara, terutama pada kawasan perkotaan yang mempunyai kepadatan penduduk. Kawasan dengan kepadatan penduduk yang tinggi kerap menghadapi tantangan keamanan yang lebih besar dibandingkan Kawasan dengan kepadatan penduduk yang rendah seperti daerah pedesaan dimana interaksi yang lebih erat dan pengawasan komunitas dapat mengurangi resiko terjadinya kejahatan.

Menurut laporan dari Databoks.katadata.co.id (2023), Kepolisian Republik Indonesia (Polri) melaporkan ada 137.419 kasus kejahatan yang terjadi di Indonesia selama periode Januari-April 2023, yang meningkat 30,7% dibandingkan dengan 105.133 kasus pada periode yang sama tahun lalu. Mayoritas kasus kejahatan yang terjadi tahun ini adalah pencurian dengan pemberatan (curat), sebanyak 30.019 kasus. Berdasarkan UU Nomor 1 Tahun 2023 tentang Kitab Undang-Undang Hukum Pidana (KUHP), curat adalah pencurian yang dilakukan dalam keadaan tertentu sehingga hukumannya lebih berat, termasuk pencurian benda suci keagamaan, benda purbakala, ternak, barang sumber mata pencaharian utama, pencurian saat terjadi bencana atau kecelakaan, pencurian di malam hari dalam rumah atau pekarangan tertutup, pencurian dengan cara merusak atau memalsukan, dan pencurian secara bersama-sama dan bersekutu. Sepanjang Januari-April 2023, 10 kasus kejahatan yang paling banyak terjadi adalah pencurian dengan pemberatan (30.019 kasus), pencurian biasa (20.043 kasus), penipuan (6.425 kasus), penganiayaan (6.374 kasus), narkoba (5.287 kasus), penggelapan asal-usul (3.516 kasus), curanmor roda dua (3.136 kasus), pencurian dengan kekerasan (3.124 kasus), pengeroyokan (1.953 kasus), dan penggelapan (7 kasus).

Selain itu, nilai properti dan barang-barang di dalam rumah yang semakin meningkat menjadikan rumah sebagai target potensial bagi pelaku kejahatan. Penelitian menunjukkan bahwa rumah dengan barang berharga seperti peralatan elektronik, perhiasan, dan uang tunai lebih rentan terhadap pencurian. Kemajuan teknologi juga mempengaruhi dinamika keamanan rumah. Sistem keamanan yang canggih seperti kamera pengawas dan alarm memang dapat meningkatkan perlindungan, namun teknologi ini juga dapat dieksploitasi oleh penjahat melalui serangan siber dan pemantauan elektronik.

Sistem keamanan rumah saat ini menghadapi beberapa masalah spesifik yang mengurangi efektivitasnya dalam melindungi penghuni dan properti. Keterbatasan sistem keamanan konvensional dan kebutuhan akan solusi yang lebih canggih dan terintegrasi menjadi semakin jelas dengan perkembangan teknologi dan peningkatan kompleksitas ancaman keamanan. Sistem keamanan konvensional, seperti alarm dan kamera CCTV pasif, sering kali hanya memberi peringatan setelah kejadian kejahatan terjadi. Banyak sistem tradisional juga tidak terintegrasi dengan baik, mengandalkan intervensi manusia yang dapat membuat kesalahan, serta rentan terhadap sabotase.

Oleh karena itu, diperlukan solusi keamanan yang lebih canggih seperti sistem berbasis IoT, pemantauan dan respons *realtime*, automasi dan kendali jarak jauh, serta peningkatan keamanan data. Sistem berbasis IoT dapat menghubungkan berbagai perangkat keamanan ke dalam satu jaringan yang terintegrasi, memungkinkan pengawasan yang lebih komprehensif dan respons yang lebih cepat. Teknologi canggih memungkinkan pemantauan dan analisis *realtime*, sementara automasi dan kendali jarak jauh memberikan fleksibilitas dalam mengontrol keamanan rumah. Selain itu, peningkatan keamanan data dan enkripsi diperlukan untuk melindungi dari ancaman siber. Solusi-solusi ini memungkinkan perlindungan yang lebih efektif dan respons yang lebih cepat terhadap ancaman, meningkatkan keselamatan dan ketenangan pikiran penghuni rumah.

Dalam menghadapi tantangan keamanan rumah modern, penggunaan teknologi canggih menjadi suatu keharusan. Salah satu solusi yang dapat diimplementasikan adalah penggunaan ESP32-CAM yang dikombinasikan dengan PIR *motion* sensor. Kombinasi ini menawarkan cara efektif untuk mendeteksi gerakan dan memantau aktivitas di sekitar rumah. Namun, sistem ini memiliki keterbatasan yang perlu dioptimalkan untuk meningkatkan akurasi dan keandalannya.

PIR (*Passive Infrared*) *motion* sensor adalah perangkat yang mendeteksi gerakan berdasarkan perubahan panas inframerah dalam lingkungannya. PIR *motion* sensor biasanya digunakan untuk mendeteksi keberadaan manusia atau hewan di area tertentu.

Kombinasi ESP32-CAM dan PIR *motion* sensor memungkinkan sistem keamanan rumah yang dapat mendeteksi gerakan dan merekam video atau mengambil gambar ketika ada aktivitas mencurigakan. Namun, sistem ini memiliki

beberapa keterbatasan, seperti jarak yang terbatas, deteksi gerakan yang terlalu sensitif atau tidak akurat yang dapat menyebabkan peringatan yang tidak berarti.

Untuk meningkatkan akurasi dan keandalan sistem, diperlukan beberapa langkah optimalisasi. Salah satu cara untuk mengatasi masalah deteksi gerakan pada PIR *motion* sensor adalah dengan mengintegrasikan algoritma YOLOv4 (*You Only Look Once* version 4) dan menggunakan beberapa teknik *filter pre-processing* gambar.

YOLOv4 adalah algoritma deteksi objek yang sangat efisien dan akurat. Algoritma ini dapat digunakan untuk menganalisis gambar yang diambil oleh ESP32-CAM dan menentukan apakah gerakan yang terdeteksi oleh PIR sensor disebabkan oleh manusia atau objek lain. Dengan mengintegrasikan YOLOv4, sistem dapat mengurangi jumlah peringatan yang disebabkan oleh gerakan non-manusia seperti hewan peliharaan atau bayangan pohon.

Pertama yaitu *gaussian filter* menggunakan *filter* ini untuk mengurangi *noise* pada gambar yang diambil oleh ESP32-CAM, sehingga meningkatkan kualitas deteksi objek. *Median filter*, *filter* ini membantu dalam menghilangkan *noise salt and pepper* yang sering muncul pada gambar digital, membuat deteksi objek menjadi lebih akurat. *Histogram equalization*, teknik ini dapat digunakan untuk meningkatkan kontras gambar, yang sangat membantu dalam kondisi pencahayaan yang buruk. CLAHE (*Contrast Limited Adaptive Histogram Equalization*) merupakan metode yang bekerja lebih baik pada gambar dengan kontras yang sangat rendah atau sangat tinggi.

Sebagai bagian dari optimalisasi sistem keamanan, salah satu fitur penting yang dapat ditambahkan adalah notifikasi *realtime* melalui aplikasi *messaging* seperti Telegram. Fitur ini memungkinkan pemilik rumah menerima peringatan segera dan gambar objek yang terdeteksi langsung ke perangkat mobile mereka.

Berdasarkan uraian tersebut, peneliti tertarik melakukan penelitian dengan judul "Implementasi YOLOv4 Dengan Teknik *pre-processing* Gambar Untuk Optimalisasi Sistem Monitoring Keamanan Rumah Berbasis ESP32-CAM". Penelitian ini diharapkan dapat memberikan solusi keamanan rumah yang lebih efektif dan efisien. Inovasi ini tidak hanya meningkatkan keamanan rumah, tetapi juga memberikan rasa aman bagi pemilik rumah. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan teknologi keamanan berbasis IoT dan kecerdasan buatan, serta membuka peluang untuk penelitian lebih lanjut di masa depan.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang yang telah dipaparkan, rumusan masalah yang muncul yaitu:

1. Bagaimana perbandingan metrik *performance* dari deteksi objek yang dihasilkan oleh penggunaan *filter Gaussian, Median, Histogram Equalization*,

dan CLAHE dibandingkan dengan sistem deteksi objek tanpa menggunakan *filter*?

2. Bagaimana pengaruh integrasi YOLOv4 dan teknik *pre-processing* gambar terhadap performa sistem keamanan rumah dalam kondisi pencahayaan yang berbeda-beda, seperti pagi, siang dan malam hari?
3. Bagaimana penerapan notifikasi *realtime* melalui aplikasi *messaging* seperti Telegram terhadap peringatan yang diberikan oleh sistem pengawasan rumah berbasis ESP32-CAM yang telah dioptimalkan dengan YOLOv4 dan teknik *pre-processing* gambar?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Untuk membandingkan *metrics performance* YOLOv4 yang dihasilkan oleh penggunaan *filter gaussian*, *Median*, *Histogram Equalization*, dan CLAHE dibandingkan dengan sistem yang mendeteksi gerakan tanpa menggunakan *filter*.
2. Untuk melihat pengaruh integrasi YOLOv4 dan teknik *pre-processing* gambar terhadap performa sistem keamanan rumah dalam berbagai kondisi pencahayaan, seperti pagi hari, siang-sore hari, dan malam hari.
3. Untuk mengevaluasi bagaimana penerapan notifikasi *realtime* melalui aplikasi *messaging* seperti Telegram terhadap peringatan yang diberikan oleh sistem pengawasan rumah berbasis ESP32-CAM yang telah dioptimalkan dengan YOLOv4 dan teknik *pre-processing* gambar.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan berbagai manfaat bagi peningkatan keandalan sistem pengawasan rumah melalui penerapan teknik *pre-processing* gambar seperti *filter Gaussian*, *Median*, *Histogram Equalization*, dan CLAHE untuk mengurangi *noise* untuk meningkatkan kualitas deteksi objek ESP32-CAM. Selain itu, penelitian ini bertujuan untuk mengembangkan teknologi deteksi objek yang lebih akurat dan responsif dengan menggunakan YOLOv4, serta menyediakan notifikasi *realtime* kepada pemilik rumah melalui aplikasi, yang akan meningkatkan keselamatan dan keamanan rumah. Penggunaan teknologi IoT dan ESP32-CAM yang ekonomis dalam penelitian ini juga menunjukkan bagaimana solusi keamanan rumah yang efektif dan terjangkau dapat diimplementasikan, memberikan kemudahan monitoring bagi pengguna dan kontribusi terhadap pengembangan sistem visi komputer.

1.5 Ruang Lingkup

Batasan masalah dalam penelitian ini yaitu:

1. Penelitian ini dibatasi pada pengujian sistem keamanan di lingkungan rumah tangga saja. Sistem tidak diuji dalam skala yang lebih besar, fokusnya adalah pada aplikasi keamanan rumah untuk individu atau keluarga.

2. Penelitian ini tidak mencakup aspek keamanan data secara mendalam, data yang diambil melalui kamera ESP32-CAM oleh *Server* dan dari *Server* ke aplikasi telegram tidak dienkripsi secara khusus dalam penelitian ini.
3. Penelitian ini berfokus pada penggunaan ESP32-CAM dan laptop sebagai *Server* lokal sebagai perangkat keras utama dan tidak mencakup perangkat keras tambahan seperti sistem alarm, kunci pintu otomatis ataupun sensor lainnya yang mungkin digunakan pada sistem yang lebih kompleks.
4. Algoritma YOLO yang digunakan adalah algoritma yolo dengan versi 4, untuk perbandingan digunakan algoritma YOLOv4 dengan *filter Pre-processing* gambar dengan algoritma YOLOv4 tanpa *Pre-processing* gambar.
5. Teknik *pre-processing* gambar yang diterapkan adalah terbatas pada *Gaussian Blur*, *Median filtering*, *Histogram Equalization*, dan CLAHE. Teknik lain yang tidak disebutkan tidak akan dieksplorasi atau dievaluasi dalam penelitian ini.

1.6 Studi Tedahulu

1. Venna, F. C. Perbandingan IoT pada Sensor Kinect, Sensor PIR, dan RFID dalam Sistem Keamanan Rumah (2022). Jurnal ini membahas perbandingan penggunaan tiga jenis sensor-sensor kinect, Sensor PIR (*Passive Infrared*), dan RFID (*Radio Frequency Identification*) dalam sistem keamanan rumah berbasis *Internet of Things* (IoT). Penelitian bertujuan untuk mengevaluasi keefektifan, efisiensi, dan keandalan masing-masing sensor dalam mendeteksi dan merespons ancaman keamanan di lingkungan rumah. Hasil penelitian menunjukkan bahwa setiap sensor memiliki keunggulan dan kelemahan masing-masing. Pilihan sensor yang terbaik tergantung pada kebutuhan spesifik dan kondisi lingkungan dari sistem keamanan rumah yang diterapkan. Sensor Kinect unggul dalam deteksi yang akurat, Sensor PIR dalam efisiensi energi, dan RFID dalam aplikasi akses kontrol.
2. Siswanto, S. Prototype Smart Home dengan Konsep IoT (*Internet of Things*) Berbasis NodeMCU dan Telegram (2020). Jurnal ini membahas tentang pengembangan prototipe smart home dengan konsep *Internet of Things* (IoT) yang berbasis pada NodeMCU dan aplikasi Telegram. Penelitian ini bertujuan untuk menciptakan sistem rumah cerdas yang dapat dikendalikan dan dipantau dari jarak jauh menggunakan teknologi IoT. NodeMCU, sebuah mikrokontroler berbasis ESP8266 yang memiliki kemampuan Wi-Fi, digunakan sebagai pusat pengendali utama dalam sistem ini. Sistem ini memungkinkan berbagai perangkat rumah tangga untuk terhubung dan dikendalikan melalui internet. Hasil pengujian menunjukkan bahwa prototipe ini berfungsi dengan baik, di mana perangkat dapat merespons perintah dengan cepat dan notifikasi dapat diterima secara *realtime*. Penggunaan NodeMCU dan Telegram terbukti efektif dalam mengimplementasikan sistem smart home yang terjangkau dan mudah digunakan.
3. Saputra, F. A. Prototipe Sistem Keamanan Ruang *Server* Otomatis Menggunakan ESP32-CAM dan Algoritma *You Only Look Once* (YOLO) (2022). Jurnal ini membahas pengembangan prototipe sistem keamanan ruang *Server* otomatis

menggunakan ESP32-CAM dan algoritma *You Only Look Once* (YOLO). Penelitian ini bertujuan untuk menciptakan sistem yang mampu mendeteksi dan mengidentifikasi ancaman keamanan di ruang *Server* secara otomatis dan *realtime*. ESP32-CAM, sebuah modul kamera berbasis mikrokontroler ESP32 dengan kemampuan konektivitas Wi-Fi, digunakan sebagai perangkat utama untuk menangkap gambar dan video. Hasil pengujian menunjukkan bahwa prototipe ini mampu mendeteksi objek dengan baik dan memberikan notifikasi secara cepat dan akurat. Implementasi ESP32-CAM dan algoritma YOLO dalam sistem keamanan ruang *Server* ini terbukti efektif dalam meningkatkan keamanan dengan memungkinkan pemantauan otomatis dan *realtime*.

4. Yoga, I. P. S. Pendeteksi Jumlah Orang pada Sistem Bangunan Pintar Menggunakan Algoritma *You Only Look Once* (2023). Jurnal ini membahas tentang pengembangan sistem pendeteksi jumlah orang dalam bangunan pintar menggunakan algoritma *You Only Look Once* (YOLO). Penelitian ini bertujuan untuk menciptakan sistem yang mampu secara otomatis mendeteksi dan menghitung jumlah orang yang berada di dalam sebuah bangunan, guna meningkatkan efisiensi dan keamanan sistem bangunan pintar. Algoritma YOLO, yang dikenal karena kecepatannya dan akurasi dalam deteksi objek, digunakan untuk menganalisis gambar dan video yang diambil oleh kamera yang dipasang di dalam bangunan. Sistem ini dirancang untuk mengidentifikasi individu dan menghitung jumlahnya secara *realtime*, serta mengirimkan data tersebut ke pusat pengendalian bangunan pintar. Dengan informasi ini, pengelola bangunan dapat mengoptimalkan penggunaan energi, mengatur sistem ventilasi, serta meningkatkan keamanan dengan memantau kepadatan orang di berbagai area bangunan.
5. Zuanita Syifaul Jannah Implementasi Algoritma YOLO (*You Only Look Once*) Untuk Deteksi Rias Adat Nusantara (2022). Jurnal ini membahas implementasi algoritma *You Only Look Once* (YOLO) untuk mendeteksi rias adat Nusantara. Penelitian ini bertujuan untuk mengembangkan sistem yang mampu mengidentifikasi dan mendeteksi berbagai jenis rias adat yang ada di Indonesia menggunakan teknologi deteksi objek berbasis YOLO. Algoritma YOLO dipilih karena kecepatannya dan akurasi dalam mendeteksi objek dalam gambar atau video, sehingga cocok untuk aplikasi yang membutuhkan pengenalan visual secara *realtime*. Dalam penelitian ini, gambar-gambar rias adat Nusantara dikumpulkan dan digunakan sebagai data pelatihan untuk mengembangkan model deteksi berbasis YOLO. Sistem ini kemudian diuji untuk melihat kemampuannya dalam mengenali berbagai jenis rias adat dari gambar yang diberikan. Hasil pengujian menunjukkan bahwa sistem ini mampu mendeteksi rias adat dengan tingkat akurasi yang tinggi, menunjukkan bahwa YOLO efektif dalam mengenali karakteristik unik dari setiap rias adat yang dianalisis.
6. Kamil, F. Pengolahan Citra Digital Menggunakan Metode YOLO untuk Mendeteksi Kualitas Dari Biji Kopi Berbasis Android (2023). Jurnal ini membahas pengolahan citra digital menggunakan metode *You Only Look Once* (YOLO) untuk mendeteksi kualitas biji kopi berbasis android. Penelitian ini bertujuan untuk mengembangkan aplikasi yang mampu mengidentifikasi dan menilai kualitas biji

kopi secara otomatis melalui gambar yang diambil menggunakan perangkat android. Algoritma YOLO, yang dikenal karena kemampuannya mendeteksi objek dengan cepat dan akurat, digunakan untuk menganalisis citra biji kopi dan mengklasifikasikannya berdasarkan kualitas. Proses pengembangan aplikasi melibatkan pengumpulan dataset gambar biji kopi dengan berbagai tingkat kualitas, yang kemudian digunakan untuk melatih model deteksi berbasis YOLO. Aplikasi ini dirancang agar pengguna dapat mengambil foto biji kopi menggunakan kamera ponsel mereka, dan sistem akan secara otomatis mendeteksi dan memberikan penilaian kualitas biji kopi tersebut.

7. Noviyanti, D. S. Penerapan Metode *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *Gaussian filter* dalam Perbaikan Kualitas Citra Fundus Retina (2020). Jurnal ini membahas penerapan dua teknik *pre-processing* gambar, yaitu *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *Gaussian filter*, untuk meningkatkan kualitas citra fundus retina. Tujuannya adalah untuk mengoptimalkan kualitas gambar medis yang sering digunakan dalam diagnosis penyakit mata. Penelitian ini menggunakan metode CLAHE untuk meningkatkan kontras lokal pada citra fundus retina dan *Gaussian filter* untuk menghaluskan gambar serta mengurangi *noise*. Proses melibatkan beberapa langkah, mulai dari akuisisi citra fundus, penerapan *Gaussian filter* untuk mengurangi *noise*, dilanjutkan dengan CLAHE untuk meningkatkan kontras, hingga evaluasi hasil dengan parameter kualitas gambar yang sesuai. Hasil penelitian menunjukkan bahwa kombinasi CLAHE dan *Gaussian filter* secara signifikan meningkatkan kualitas citra fundus retina. Peningkatan kontras membantu dalam memperjelas detail-detail penting dalam citra, sedangkan pengurangan *noise* melalui *Gaussian filter* membuat citra lebih bersih dan lebih mudah untuk dianalisis oleh ahli medis. Evaluasi berdasarkan parameter kualitas gambar menunjukkan adanya perbaikan yang berarti dibandingkan dengan citra asli tanpa *Pre-processing*.
8. Nabuasa, Y. N. Perbandingan Metode *Histogram Equalization* dan Specification pada Citra Abu-Abu (2019). Jurnal ini membahas perbandingan antara dua metode peningkatan kualitas gambar, yaitu *Histogram Equalization* dan Histogram Specification, pada citra abu-abu. Fokusnya adalah pada evaluasi efektivitas kedua metode tersebut dalam meningkatkan kontras gambar abu-abu. Penelitian ini menggunakan metode *Histogram Equalization* untuk mendistribusikan intensitas piksel secara merata dan metode Histogram Specification untuk mengubah distribusi histogram gambar target sesuai dengan histogram gambar referensi. Citra abu-abu yang digunakan dalam penelitian ini diproses menggunakan kedua metode tersebut, dan hasilnya dibandingkan berdasarkan parameter kualitas gambar seperti kontras, histogram, dan persepsi visual. Hasil penelitian menunjukkan bahwa kedua metode mampu meningkatkan kontras gambar abu-abu, tetapi dengan karakteristik yang berbeda. *Histogram Equalization* cenderung meningkatkan kontras secara keseluruhan namun dapat menyebabkan kehilangan detail di area dengan intensitas tinggi atau rendah. Sementara itu, Histogram Specification memberikan hasil yang lebih terkontrol dengan distribusi kontras yang sesuai dengan gambar referensi, menjaga detail

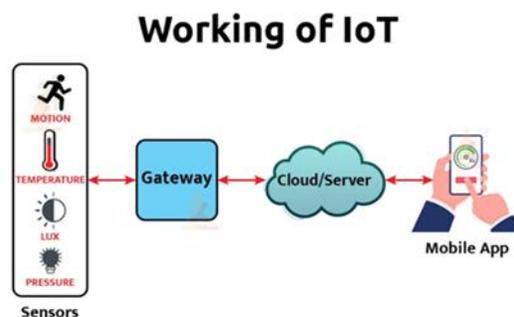
gambar lebih baik. Evaluasi persepsi visual menunjukkan bahwa Histogram Specification lebih disukai karena memberikan hasil yang lebih alami dan terjaga.

1.7 Teori

1.7.1 *Internet of Things (IoT)*

Internet of Things (IoT) adalah konsep dan teknologi yang memungkinkan berbagai perangkat fisik terhubung ke internet, berkomunikasi, dan berbagi data satu sama lain. Perangkat ini bisa berupa segala sesuatu yang memiliki sensor atau aktuator, seperti peralatan rumah tangga, kendaraan, kamera CCTV, sensor pemantauan lingkungan, perangkat medis, dan banyak lagi. Dengan konektivitas ini, perangkat tidak hanya bisa mengumpulkan dan mengirim data tetapi juga dapat menerima instruksi dan beroperasi secara otomatis atau semi-otomatis berdasarkan informasi yang mereka terima.

Konsep IoT bertujuan untuk membuat internet semakin berkembang dan meluas dengan memungkinkan akses dan interaksi yang mudah dengan beragam perangkat seperti peralatan rumah tangga, kamera CCTV, sensor pemantauan, aktuator, display, kendaraan, dan sebagainya. IoT akan mendorong pengembangan sejumlah aplikasi yang memanfaatkan jumlah dan variasi data yang berpotensi besar yang dihasilkan oleh objek-objek tersebut untuk memberikan layanan baru kepada warga negara, perusahaan, dan administrasi publik (Wilianto dan Kurniawan, 2018).



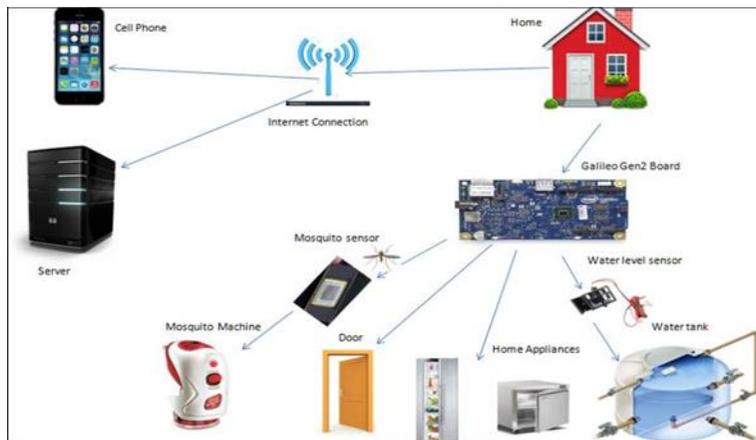
Gambar 1 Bagaimana IoT bekerja

Tantangan utama dalam IoT adalah menghubungkan dunia fisik dengan dunia informasi, memproses data yang diperoleh dari peralatan elektronik melalui antarmuka antara pengguna dan peralatan. Sensor mengumpulkan data fisik mentah dari skenario *realtime* dan mengubahnya ke dalam format yang dapat dipahami oleh mesin sehingga mudah dipertukarkan antara berbagai format data (*Thing*) (Junaidi, A. 2015).

1.7.2 **Sistem Monitoring Keamanan Rumah**

Sistem monitoring keamanan rumah adalah kumpulan teknologi, perangkat elektronik, dan sistem informasi yang dirancang untuk meningkatkan keamanan dan pengawasan rumah secara keseluruhan. Tujuannya adalah untuk memberikan

perlindungan terhadap intrusi, kebakaran, atau keadaan darurat lainnya dengan mendeteksi, memberi peringatan, dan mengelola respons terhadap ancaman potensial. Sistem ini dapat terdiri dari berbagai komponen, termasuk sensor pintu dan jendela, sensor gerakan, kamera keamanan, alarm, dan perangkat kontrol yang terhubung ke internet.



Gambar 2 Konfigurasi sistem monitoring rumah

Sistem monitoring keamanan rumah telah menjadi fokus perhatian yang signifikan dalam era IoT. Penerapan *Internet of Things* (IoT) memungkinkan integrasi yang lebih dalam antara perangkat elektronik, sensor, dan sistem informasi, membawa dampak besar dalam berbagai aspek kehidupan, termasuk keamanan rumah. Dalam konteks ini, sensor-sensor yang terhubung ke internet memainkan peran kunci dalam meningkatkan keamanan rumah. Sensor pintu, jendela, dan gerakan yang terkoneksi ke jaringan dapat mendeteksi aktivitas yang mencurigakan dan memberikan pemberitahuan kepada pemilik rumah secara *realtime*. Lebih dari itu, sistem IoT juga memungkinkan respons otomatis terhadap situasi darurat, seperti memicu alarm atau mengaktifkan kamera keamanan untuk merekam kejadian yang mencurigakan.

1.7.3 ESP32-CAM

ESP32-CAM merupakan mikrokontroler yang memiliki fasilitas tambahan seperti bluetooth, wifi, kamera, bahkan slot microSD. ESP32-CAM biasanya digunakan untuk proyek-proyek IoT (*Internet of Things*) yang membutuhkan fitur kamera. Modul ESP32-CAM memiliki lebih sedikit pin I/O karena sebagian besar pin sudah digunakan untuk fungsi kamera dan slot kartu microSD. Karena tidak memiliki port USB khusus, kita membutuhkan USB TTL untuk memprogram modul ini atau dapat menambahkan modul *downloader* khusus untuk ESP32-CAM (Aji. 2023).



Gambar 3 ESP32-CAM

Tabel 1 Spesifikasi ESP32-CAM

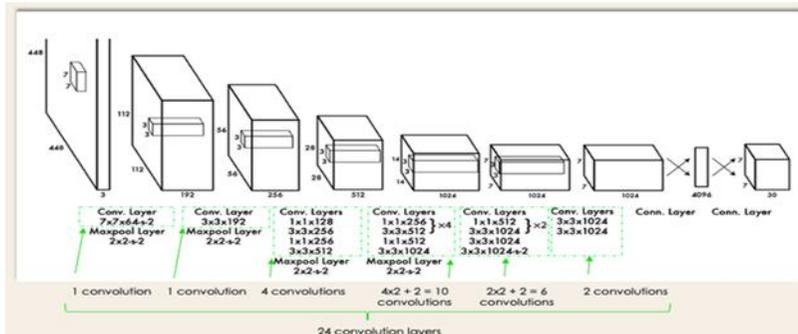
Spesifikasi ESP32-CAM	Deskripsi
Mikrokontroler	ESP32 (dual-core)
Kamera	OV2640 2MP
Antarmuka Jaringan	WiFi 802.11 b/g/n, Bluetooth
Slot Kartu Memori	MicroSD
Tegangan Operasi	5V
Konsumsi Daya	~180mA saat menyiarkan video
Resolusi Video Maksimum	1600x1200 piksel
Dukungan Perangkat Lunak	Arduino IDE, ESP-IDF
Fitur Tambahan	Deteksi Gerakan, Pengenalan Wajah

1.7.4 You Only Look Once (YOLO)

YOLO (*You Only Look Once*) adalah algoritma deteksi objek yang dirancang untuk deteksi *realtime* yang cepat dan akurat. Dikembangkan oleh Joseph Redmon dan timnya, YOLO mengubah pendekatan tradisional dalam deteksi objek dengan memproses gambar input hanya sekali, menghasilkan prediksi *bounding box* dan kelas objek secara langsung dalam satu langkah.

YOLO menggunakan pendekatan jaringan saraf tiruan (JST) untuk mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas serta probabilitas untuk setiap wilayah. Kotak-kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi. YOLO memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*, karena memproses seluruh citra saat pengujian dan memberikan prediksi secara global pada citra. YOLO menerapkan arsitektur yang mirip dengan jaringan saraf konvolusional (*Convolutional Neural Networks*). YOLO hanya menggunakan lapisan konvolusi dan lapisan *pooling*. Lapisan konvolusi terakhir disesuaikan dengan jumlah kelas dan jumlah kotak prediksi yang diinginkan (Gerald dan Lubis, 2024)

Menurut Datacamp.com (2020) Arsitektur YOLO mirip dengan GoogleNet. Seperti yang diilustrasikan di bawah ini, terdapat total 24 lapisan konvolusi, empat lapisan *max-pooling*, dan dua lapisan *fully connected*.



Gambar 4 Arsitektur *You Only Look Once* (YOLO)

Menurut Datacamp.com (2020) arsitektur bekerja sebagai berikut:

1. Mengubah ukuran gambar input menjadi 448x448 sebelum melewati jaringan konvolusi.
2. Konvolusi 1x1 diterapkan terlebih dahulu untuk mengurangi jumlah saluran, yang kemudian diikuti oleh konvolusi 3x3 untuk menghasilkan keluaran kuboidal.
3. Fungsi aktivasi yang digunakan di dalamnya adalah ReLU, kecuali untuk lapisan terakhir yang menggunakan fungsi aktivasi linear.
4. Beberapa teknik tambahan, seperti normalisasi *batch* dan *dropout*, masing-masing meragukan model dan mencegah *overfitting*.

YOLO bekerja dengan membagi gambar input menjadi sebuah grid dan kemudian untuk setiap *grid cell*, algoritma ini memprediksi *bounding box* (kotak pembatas) serta probabilitas kelas objek yang mungkin ada dalam *bounding box* tersebut (Datacamp.com, 2020).

1. Pembagian gambar menjadi *grid*: gambar dibagi menjadi *grid* dengan ukuran $S \times S$. Misalnya, jika $S = 7$, maka gambar dibagi menjadi 7×7 *grid cells*.
2. Prediksi *bounding box* dan kelas objek: setiap *grid cell* memprediksi beberapa *bounding boxes* (kotak pembatas) dan kepercayaan (*confidence score*) dari kotak tersebut. Kepercayaan ini mencerminkan seberapa besar kemungkinan adanya objek di dalam kotak serta akurasi dari kotak tersebut. Selain itu, setiap *grid cell* juga memprediksi probabilitas kelas untuk masing-masing objek.
3. *Confidence score* dihitung sebagai produk dari probabilitas bahwa terdapat objek di dalam kotak dan *Intersection over Union* (IoU) antara kotak yang diprediksi dan kotak sebenarnya (*ground truth*).
 - a) Tujuan dari IoU (sebuah nilai antara 0 dan 1) adalah untuk meniadakan kotak-kotak *grid* tersebut hanya menyimpan yang relevan.

- b) Pengguna menentukan ambang batas seleksi IoU-nya, yang misalnya bisa 0,5.
 - c) Kemudian YOLO menghitung IoU dari setiap sel *grid* yang merupakan luas irisan dibagi oleh luas gabungan.
 - d) Algoritma YOLO akan mengabaikan prediksi dari sel *grid* yang memiliki $\text{IoU} \leq$ ambang batas dan mempertimbangkan yang memiliki $\text{IoU} >$ ambang batas.
4. *Non-Maximum Suppression*: Untuk mengurangi duplikasi deteksi objek yang sama, YOLO menggunakan teknik *Non-Maximum Suppression* (NMS). NMS mempertahankan kotak dengan *confidence score* tertinggi dan menghilangkan kotak lainnya yang memiliki IoU tinggi dengan kotak tersebut.

1.7.5 Visual Studio Code

Visual Studio Code (VSCode) adalah Integrated Development Environment (IDE) yang sangat populer dan multifungsi. Dikembangkan oleh Microsoft, VSCode tersedia untuk berbagai sistem operasi seperti Windows, Linux, dan Mac. Fungsi utama dari VSCode adalah sebagai editor kode sumber yang mendukung berbagai bahasa pemrograman, termasuk Python, JavaScript, C++, dan lainnya. Kelebihannya terletak pada kemudahan penggunaan, dukungan ekstensi, dan fitur seperti debugging, Git integration, serta IntelliSense, yang membantu mempercepat proses coding dengan auto-completion yang cerdas.



Gambar 5 Logo *visual studio code*

Dalam konteks pengembangan *Internet of Things* (IoT), VSCode digunakan untuk memfasilitasi pemrograman perangkat yang terhubung ke jaringan, seperti perangkat berbasis ESP32-CAM. Dalam penelitian, VSCode membantu memudahkan proses pengkodean dan simulasi perangkat IoT melalui antarmuka yang ringan namun kuat, serta dukungan terhadap berbagai plugin yang relevan untuk pengembangan IoT (Romzi. M, 2020).

1.7.6 Arduino IDE

Arduino Integrated Development Environment (IDE) adalah perangkat lunak open-source yang digunakan untuk menulis, mengedit, dan mengunggah kode ke papan mikrokontroler Arduino. Dikembangkan oleh tim Arduino, IDE ini bertujuan untuk menyediakan antarmuka pemrograman yang mudah diakses bagi pengembang pemula dan profesional. Sejak diperkenalkan pada tahun 2005,

Arduino IDE telah menjadi alat yang sangat populer dalam komunitas pembuat, pengembang hobi, dan profesional untuk mengembangkan proyek-proyek elektronik dan IoT.

Aplikasi Arduino IDE berguna untuk mengedit, membuat, dan mengunggah program ke board yang ditentukan, serta untuk menulis kode program. Aplikasi ini berfungsi sebagai software untuk membuat program dan mensimulasikan perilaku rangkaian elektronika, baik analog maupun digital (Kamal. 2023).



Gambar 6 Logo software arduino IDE

Salah satu kelebihan utama Arduino IDE adalah kemudahan penggunaannya. Dengan antarmuka yang sederhana dan alat-alat bantu yang komprehensif, pengguna dapat dengan cepat memulai pengembangan proyek tanpa perlu menghabiskan banyak waktu untuk mempelajari perangkat lunak yang kompleks. Selain itu, dukungan untuk berbagai jenis papan Arduino memungkinkan fleksibilitas dalam pengembangan proyek, mulai dari prototipe sederhana hingga sistem yang lebih kompleks. Pustaka bawaan yang kaya juga membantu pengguna untuk dengan mudah menambahkan fungsi tambahan tanpa perlu menulis kode dari awal.

1.7.7 Telegram

Telegram adalah aplikasi pesan instan berbasis cloud yang dikenal karena fokusnya pada kecepatan dan keamanan. Didirikan oleh Pavel Durov dan Nikolai Durov, Telegram diluncurkan pada tahun 2013. Aplikasi ini menawarkan berbagai fitur canggih yang membuatnya populer di kalangan pengguna di seluruh dunia.



Gambar 7 Logo aplikasi telegram

Telegram dirancang sebagai aplikasi multiplatform yang tersedia untuk berbagai sistem operasi seperti Android, iOS, Windows, macOS, dan Linux. Keberadaan aplikasi desktop yang kuat menjadikannya pilihan ideal bagi pengguna yang sering bekerja di komputer. Telegram juga menyediakan API terbuka yang memungkinkan pengembang untuk menciptakan bot dan aplikasi pihak ketiga.

API terbuka Telegram memungkinkan pengembang untuk menciptakan bot yang dapat menjalankan berbagai fungsi, seperti memberikan berita terbaru, memproses pembayaran, atau menyediakan layanan pelanggan. Bot ini memperkaya pengalaman pengguna dan meningkatkan fungsionalitas aplikasi.

1.7.8 *Pre-processing* Gambar

Dalam dunia pemrosesan citra, *pre-processing* merupakan langkah esensial untuk memastikan bahwa data gambar yang digunakan dalam sistem deteksi objek sudah siap dan optimal. *pre-processing* gambar memainkan peran penting dalam mempersiapkan data visual yang lebih berkualitas untuk analisis lebih lanjut. Pada penelitian ini, digunakan empat teknik *pre-processing* gambar: *Gaussian filter*, *Median filter*, *Histogram Equalization*, dan CLAHE (*Contrast Limited Adaptive Histogram Equalization*). Masing-masing teknik ini memiliki fungsi spesifik dalam mereduksi *noise*, meningkatkan kualitas kontras, dan memastikan bahwa gambar yang dihasilkan dari kamera ESP32-CAM lebih optimal untuk pendeteksian objek oleh model YOLOv4.

Filter gaussian adalah *filter* perata linier yang memilih bobot berdasarkan bentuk fungsi Gaussian. Baik dalam domain spasial maupun domain frekuensi, *filter* perata Gaussian merupakan jenis *filter* low-pass yang efektif, terutama untuk menghilangkan derau yang mengikuti distribusi normal. (Wang, M. 2014).

Teknik ini bekerja dengan menerapkan konvolusi menggunakan kernel Gaussian, yang menghitung nilai rata-rata piksel berdasarkan distribusi gaussian di sekitarnya. Hasilnya adalah gambar yang lebih lembut dengan transisi piksel yang lebih halus. *Gaussian filter* sangat berguna untuk mengurangi efek *noise* pada gambar yang dapat mengganggu deteksi objek, sehingga meningkatkan kinerja algoritma deteksi.

Median filter adalah metode *filter* yang digunakan dalam pemrosesan citra untuk mengurangi *noise* dengan cara mengambil nilai median dari piksel-piksel di sekitarnya. *Filter* ini membantu menghaluskan citra dengan menghilangkan outlier atau nilai ekstrem yang dapat mempengaruhi kualitas citra (Noviyanti, D. S. 2020).

Median filter adalah metode non-linear yang digunakan untuk menghilangkan *noise* dari gambar digital, terutama *noise* garam dan merica. Teknik ini menggantikan setiap piksel dalam gambar dengan nilai median dari tetangganya dalam jendela persegi. *Median filter* sangat efektif dalam membersihkan *noise* tanpa mengaburkan tepi objek, sehingga gambar tetap tajam dan jelas. Teknik ini sering digunakan dalam aplikasi pengolahan gambar medis dan astronomi.

Peningkatan Gambar adalah proses memperbaiki kualitas visual gambar sehingga hasilnya lebih sesuai daripada gambar aslinya. Peningkatan gambar digunakan untuk memperbaiki kontras gambar yang memiliki luminansi rendah. Peningkatan gambar menghilangkan distorsi dari gambar dan meningkatkan kualitasnya. *Histogram equalization* adalah salah satu teknik peningkatan. *Histogram equalization* digunakan untuk meningkatkan kontras gambar berwarna (Kapoor, K. 2015).

Metode ini biasanya meningkatkan kontras global dari banyak gambar, terutama ketika data yang dapat digunakan dari gambar diwakili oleh nilai kontras yang berdekatan. Melalui penyesuaian ini, intensitas dapat didistribusikan lebih baik pada histogram. Ini memungkinkan area dengan kontras lokal rendah untuk mendapatkan kontras yang lebih tinggi. *Histogram equalization* mencapai hal ini dengan menyebarkan nilai intensitas yang paling sering muncul secara efektif (Dorothy, R. 2015).

Histogram Equalization memiliki manfaat untuk meningkatkan *global contrast* dari citra, sehingga informasi yang terdapat pada citra menjadi lebih jelas terlihat. Metode ini juga dapat meratakan nilai intensitas citra sesuai dengan histogramnya, sehingga citra memiliki penyebaran histogram yang merata (Nabuasa, Y. N. 2019).

Contrast Limited Adaptive Histogram Equalization (CLAHE) adalah metode untuk meningkatkan kontras visual dengan memperkuat kontras lokal gambar. Gambar yang dihasilkan dari peningkatan kontras menggunakan CLAHE memiliki kelemahan, yaitu derau pada gambar juga ikut diproses. Untuk mengatasi kekurangan ini, *filter gaussian* diterapkan untuk mengurangi derau pada gambar hasil CLAHE (Furqan, M. 2022).

CLAHE adalah varian dari *histogram equalization* yang bekerja pada bagian-bagian kecil dari gambar dan membatasi peningkatan kontras untuk mencegah amplifikasi *noise*. Teknik ini membagi gambar menjadi blok-blok kecil dan menerapkan *histogram equalization* dengan batasan tertentu (*clip limit*) pada setiap blok. Hasil dari setiap blok kemudian digabungkan menggunakan interpolasi bilinear. CLAHE meningkatkan kontras lokal tanpa meningkatkan *noise* secara signifikan, membuatnya sangat berguna dalam pemrosesan gambar medis dan citra satelit.

1.7.9 Metriks Performance YOLO

Confusion matrix digunakan untuk mengevaluasi performa dari sebuah model machine learning. *Confusion matrix* adalah matriks yang menampilkan prediksi dari klasifikasi aktual dan klasifikasi yang diprediksi. Terdapat empat klasifikasi dalam confusion matrix yaitu *True Negative* (TN), *True Positive* (TP), *False Negative* (FN), dan *False Positive* (FP) yang berasal dari nilai aktual dan nilai prediksi. Definisi dari confusion matrix dimana TP (*True Positive*) adalah jumlah sampel positif yang diklasifikasikan dengan benar, TN (*True Negative*) adalah jumlah sampel negatif yang diklasifikasikan dengan benar, FP (*False Positive*) adalah jumlah sampel negatif yang salah diklasifikasikan sebagai positif, FN (*False Negative*) adalah jumlah sampel positif yang salah diklasifikasikan sebagai negatif. Ilustrasi dari

confusion matrix dapat dilihat pada gambar dibawah. Performa model dapat dihitung dengan menggunakan *precision*, *recall*, dan *F1-score* yang berasal dari *confusion matrix* (Sary, I. P. 2023).

	Actual	
Prediction	TP	FP
	FN	TN

Gambar 8 Confusion matriks

Presisi adalah rasio TP terhadap jumlah total data positif yang diprediksi. Pada bagian penyebut, terdapat variabel FP sebagai pembagi.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Di sisi lain, *recall* didefinisikan sebagai rasio TP terhadap jumlah total contoh yang benar-benar positif. Penyebutnya termasuk FN sebagai pembagi.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Ketika *recall* sangat tinggi, presisi akan sangat rendah, dan sebaliknya. Ada hubungan *trade-off* antara presisi dan *recall*. Hubungan *trade-off* ini menyiratkan bahwa jumlah kedua variabel ini sama dengan 1. Harmonisasi rata-rata antara *precision* dan *recall* disebut dengan *F1-score*. Maka nilai terbaik untuk *F1-score* adalah 1.0, sedangkan nilai terburuknya adalah 0.0.

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

1.7.10 Kualitas Gambar

Dua jenis metode evaluasi digunakan untuk penilaian kualitas gambar: evaluasi subjektif dan objektif. Evaluasi subjektif, meskipun akurat, bersifat merepotkan, memakan waktu, dan mahal. Oleh karena itu, upaya yang signifikan telah dilakukan untuk mengembangkan metrik kualitas gambar objektif. SNR, PSNR, MSE, dan SSIM merupakan beberapa ukuran kualitas gambar objektif yang paling sering digunakan.

Merupakan suatu citra (gambar) analog dengan ukuran panjang kali lebar, dapat digitalisasi dengan mengambil sampling berupa matriks berukuran m kali n, dengan m adalah jumlah sampling untuk panjang, dan n adalah jumlah sampling untuk lebar. SNR (*Signal to Noise Ratio*) adalah proses dari perbandingan antara signal yang telah dikirim terhadap *noise* maupun proses hasil. Pada signal ini digunakan untuk mengetahui besarnya pengaruh redaman signal terhadap signal yang ditransmisikan (Prastya, D. Z. E., 2020).

Untuk menghitung SNR kita memakai persamaan berikut:

$$SNR = 10 \cdot \log_{10} \left(\frac{\sum_{m,n} I_{m,n}^2}{\sum_{m,n} (I_{m,n} - \overline{I_{m,n}})^2} \right) \quad (4)$$

Nilai SNR yang baik untuk kualitas gambar umumnya bergantung pada aplikasi dan jenis gambar. Secara umum:

1. SNR di atas 40 dB biasanya dianggap sebagai kualitas gambar yang sangat baik.
2. SNR sekitar 20-30 dB menunjukkan kualitas gambar yang cukup baik.
3. SNR di bawah 20 dB seringkali dianggap sebagai kualitas yang kurang baik atau buruk karena lebih banyak *noise* yang mengganggu gambar.

Peak Signal to Noise Ratio (PSNR) adalah ukuran matematis untuk menilai kualitas gambar berdasarkan perbedaan piksel antara dua gambar. Metrik ini banyak digunakan untuk mengevaluasi kualitas gambar terkompresi dibandingkan dengan gambar aslinya. PSNR dihitung menggunakan *Mean Squared Error* (MSE) antara gambar asli dan gambar yang terdegradasi, dan biasanya dinyatakan dalam satuan desibel (dB). Untuk menghitung PSNR kita memakai persamaan berikut:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (5)$$

MAX adalah nilai piksel maksimum yang mungkin (255 untuk gambar 8-bit). MSE adalah kesalahan kuadrat rata-rata antara gambar asli dan gambar terdistorsi. PSNR umumnya digunakan dalam teknik kompresi gambar dan video. Metrik ini menilai efektivitas algoritma kompresi dengan mengukur seberapa baik gambar yang terdegradasi mendekati gambar aslinya. Namun, PSNR tidak selalu berkorelasi dengan persepsi manusia terhadap kualitas gambar. Selain itu, PSNR sensitif terhadap perubahan kecil pada nilai piksel dan tidak mampu menangkap perbedaan persepsi yang lebih halus secara akurat (Al Najjar. Y, 2024).

Dalam kompresi gambar dan video, nilai PSNR bervariasi dari 30 hingga 50 dB untuk representasi data 8-bit dan dari 60 hingga 80 dB untuk data 16-bit. Dalam transmisi nirkabel, rentang kerugian kualitas yang dapat diterima adalah sekitar 20 - 25 dB (Sara. U, 2019).

MSE (*Mean Square Error*) adalah salah satu metode statistik untuk mengukur seberapa mirip data yang dibandingkan. Peningkatan kualitas citra dapat diukur secara kuantitatif menggunakan MSE (*Mean Square Error*), yaitu sigma dari jumlah error antara citra hasil peningkatan kualitas dan citra asli. MSE dapat dihitung menggunakan persamaan berikut:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (f_a(i,j) - f_b(i,j))^2 \quad (6)$$

Dimana M dan N adalah ukuran panjang dan lebar citra, $f_a(i,j)$ adalah intensitas citra di titik (i,j) citra asli, dan $f_b(i,j)$ adalah intensitas citra di titik (i,j) setelah kualitas ditingkatkan (Kusuma, I. W. A. W. 2020).

SSIM (*Structural Similarity Index*) adalah metrik yang banyak digunakan untuk mengevaluasi kesamaan antara dua gambar. Metrik ini menilai kesamaan struktural dengan membandingkan luminansi, kontras, dan struktur. Untuk menghitung SSIM kita memakai persamaan berikut:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (7)$$

Indeks SSIM yang bernilai 1 menunjukkan kesesuaian ideal antara dua gambar, sementara indeks SSIM yang lebih rendah dari 1 menunjukkan ketidaksesuaian antara dua gambar (Peng, J. 2020).

BAB II METODE PENELITIAN

2.1 Waktu dan Lokasi Penelitian

Penelitian ini dilakukan di lingkungan rumah pribadi. Penelitian ini dilaksanakan mulai Juli 2024 sampai dengan September 2024, sedangkan penulisan tugas akhir dimulai pada Agustus 2024 sampai dengan November 2024.

2.2 Variabel Penelitian/Perancangan

Penelitian ini bertujuan untuk mengembangkan sistem pengawasan berbasis ESP32-CAM yang menggunakan algoritma deteksi objek YOLOv4. Sistem ini menerapkan teknik *Pre-processing* gambar seperti *Gaussian Blur*, *Median filtering*, *Histogram Equalization*, dan CLAHE untuk meningkatkan kualitas deteksi objek yang dihasilkan. Selain itu, sistem ini terintegrasi dengan aplikasi Telegram untuk menampilkan hasil deteksi dan memberikan notifikasi *realtime* kepada pengguna.

1. Variabel Independen

Variabel independen dalam penelitian ini adalah teknik *pre-processing* gambar yang diterapkan pada gambar yang diambil oleh ESP32-CAM. Teknik-teknik tersebut meliputi *Gaussian filter*, *Median filter*, *Histogram Equalization*, dan CLAHE. *Gaussian filter* digunakan untuk menghaluskan gambar dan mengurangi *noise* global, memberikan gambar yang lebih bersih. *Median filter* berfungsi untuk menghilangkan *noise* tipe *salt and pepper*, menjaga kejelasan tepi objek. *Histogram Equalization* diterapkan untuk meningkatkan kontras global gambar, membuat detail lebih terlihat. Sementara itu, CLAHE (*Contrast Limited Adaptive Histogram Equalization*) digunakan untuk meningkatkan kontras lokal tanpa memperkuat *noise*, memberikan gambar dengan kontras yang seimbang. Penggunaan teknik-teknik ini bertujuan untuk mengoptimalkan kualitas gambar sebelum diterapkan algoritma deteksi objek YOLOv4.

2. Variabel Dependen

Variabel dependen dalam penelitian ini mencakup kualitas gambar dan akurasi deteksi objek. Kualitas gambar dievaluasi melalui beberapa parameter: *Signal to Noise Ratio* (SNR) yang mengukur tingkat sinyal terhadap *noise* dalam gambar, menunjukkan kejernihan gambar; *Peak Signal to Noise Ratio* (PSNR) yang mengukur kualitas rekonstruksi gambar dengan membandingkan gambar asli dan gambar yang telah diproses, menunjukkan seberapa baik gambar diproses; *Mean Squared Error* (MSE) yang mengukur perbedaan rata-rata kuadrat antara gambar asli dan gambar yang telah diproses, menilai kesalahan pemrosesan; dan *Structural Similarity Index* (SSIM) yang mengukur kesamaan struktural antara gambar asli dan gambar yang telah diproses, menilai kualitas visual gambar. Akurasi deteksi objek dinilai melalui *precision* yang mengukur ketepatan deteksi objek oleh

YOLOv4, *recall* yang mengukur kemampuan YOLOv4 dalam mendeteksi semua objek yang ada dan *F1-score* yang mengukur keseimbangan antara *precision* dan *recall*. Parameter-parameter ini memberikan gambaran komprehensif tentang kinerja sistem pengawasan berbasis ESP32-CAM dan YOLOv4 dalam mendeteksi objek dengan kualitas gambar yang ditingkatkan.

2.3 Bahan Uji dan Alat

Berikut adalah tabel yang menjelaskan detail perangkat keras dan perangkat lunak yang digunakan.

Tabel 2 Perangkat keras

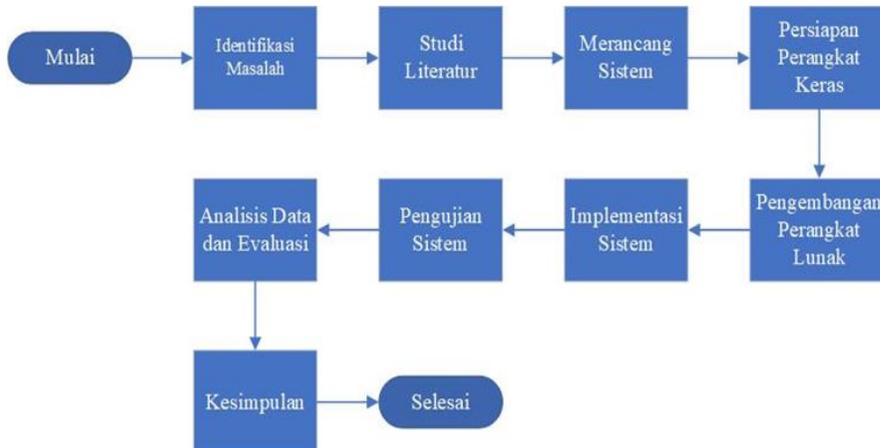
Kategori	Nama Alat	Deskripsi Fungsi
ESP32-CAM	Modul kamera ESP32-CAM	Modul mikroprosesor dengan kamera 2MP untuk menangkap gambar dan mengirimkan data gambar ke <i>Server</i> .
PIR Sensor	HC-SR501	Sensor yang mendeteksi gerakan dan memicu ESP32-CAM untuk menangkap gambar ketika ada gerakan terdeteksi.
<i>Server</i>	Laptop/PC dengan <i>Visual Studio Code</i>	Digunakan untuk menjalankan skrip pemrosesan gambar dan deteksi objek YOLOv4.
<i>Power Supply</i>	5V <i>Power Supply</i>	Menyediakan daya yang diperlukan untuk ESP32-CAM dan sensor PIR.
Kabel USB	Kabel USB <i>Type-A to Micro-B</i>	Menghubungkan ESP32-CAM ke laptop/PC untuk pemrograman dan komunikasi data.
<i>Wi-Fi</i>	<i>Router Wi-Fi</i>	<i>Router</i> untuk menghubungkan ESP32-CAM dengan laptop/PC melalui jaringan Wi-Fi.

Tabel 3 Perangkat lunak

Kategori	Nama Alat	Deskripsi Fungsi
Firmware ESP32-CAM	PlatformIO/Arduino IDE	IDE dan ekosistem untuk pengembangan firmware ESP32-CAM, memungkinkan pengembangan dan upload kode ke perangkat.
Algoritma Deteksi	YOLOv4 (<i>You Only Look Once</i>)	Algoritma deteksi objek yang cepat dan efisien untuk mendeteksi objek dalam gambar yang telah diproses.
Bahasa Pemrograman	Python	Digunakan untuk menulis skrip pemrosesan gambar, deteksi objek, dan integrasi dengan Telegram.
Perpustakaan OpenCV	OpenCV	Library untuk pemrosesan gambar, yang digunakan untuk menerapkan berbagai teknik <i>pre-processing</i> gambar.
Perpustakaan Scikit-Image	Scikit-Image	Library untuk analisis dan pemrosesan gambar, digunakan untuk menghitung metrik seperti SSIM.
Perpustakaan Tkinter	Tkinter	Library GUI untuk membangun antarmuka pengguna yang memungkinkan pemuatan gambar dan tampilan hasil analisis.
Telegram	Telegram	Platform IoT untuk menghubungkan ESP32-CAM & local <i>Server</i> dengan smartphone dan menampilkan hasil deteksi serta notifikasi.
<i>Visual Studio Code</i>	<i>Visual Studio Code</i>	Editor kode sumber yang digunakan untuk mengembangkan skrip pemrosesan gambar dan deteksi objek dengan Python.

2.4 Skema Sistem Kerja

Sistem ini dirancang dengan mempertimbangkan berbagai aspek dimana proses perancangannya meliputi serangkaian langkah sistematis sebagai berikut:



Gambar 9 Diagram alir penelitian

1. Identifikasi Masalah

Dalam penelitian ini, masalah yang diidentifikasi adalah kebutuhan akan sistem pengawasan yang efisien dan efektif yang mampu mendeteksi gerakan dan mengidentifikasi objek dalam area pengawasan secara *realtime*. Selain itu, sistem harus mampu mengirimkan notifikasi ke pengguna melalui aplikasi mobile ketika ada gerakan yang terdeteksi.

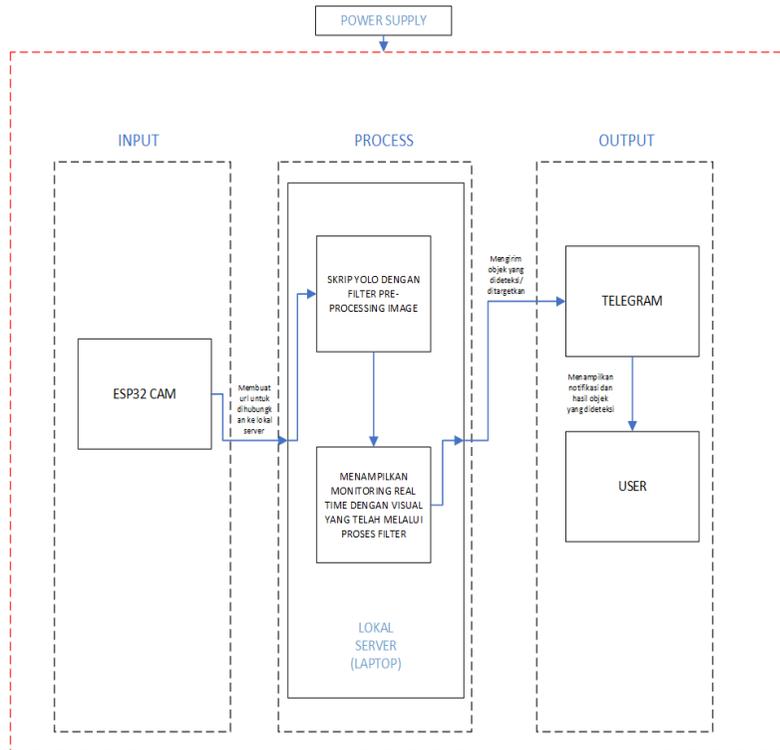
2. Studi Literatur

Setelah mengidentifikasi masalah, langkah berikutnya adalah melakukan studi literatur. Peneliti mengumpulkan informasi dari berbagai sumber, seperti jurnal ilmiah, buku, dan artikel, yang relevan dengan topik sistem pengawasan, teknologi ESP32-CAM, sensor PIR, algoritma deteksi objek YOLOv4, dan teknik *pre-processing* gambar (*Gaussian, Median, Histogram Equalization*, dan *CLAHE*). Studi literatur ini membantu peneliti memahami perkembangan terkini dan teknologi yang tersedia serta metode terbaik yang dapat digunakan untuk mengatasi masalah yang diidentifikasi.

3. Merancang Sistem

Berdasarkan hasil studi literatur, peneliti kemudian merancang sistem pengawasan yang akan dikembangkan. Desain sistem mencakup arsitektur keseluruhan, spesifikasi perangkat keras dan perangkat lunak yang akan digunakan, serta alur kerja sistem. Desain ini juga mencakup diagram alir

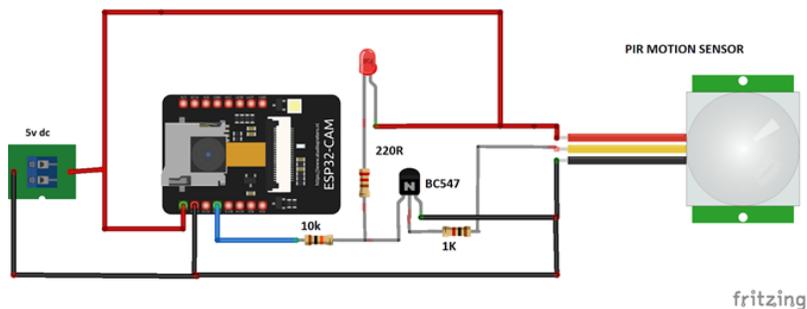
yang menggambarkan bagaimana sistem akan bekerja, mulai dari deteksi gerakan hingga pengiriman notifikasi ke aplikasi mobile.



Gambar 10 Blok sistem alat

4. Persiapan Perangkat Keras

Selanjutnya menyiapkan semua perangkat keras yang diperlukan untuk implementasi sistem. Perangkat keras yang disiapkan meliputi modul ESP32-CAM, sensor PIR, *power supply*, kabel USB, *board downloader* ESP32-CAM-MB, *router* Wi-Fi, dan *smartphone* dengan aplikasi Telegram. Setiap perangkat keras diuji untuk memastikan berfungsi dengan baik sebelum diintegrasikan ke dalam sistem.



Gambar 11 Skematik perancangan ESP32-CAM



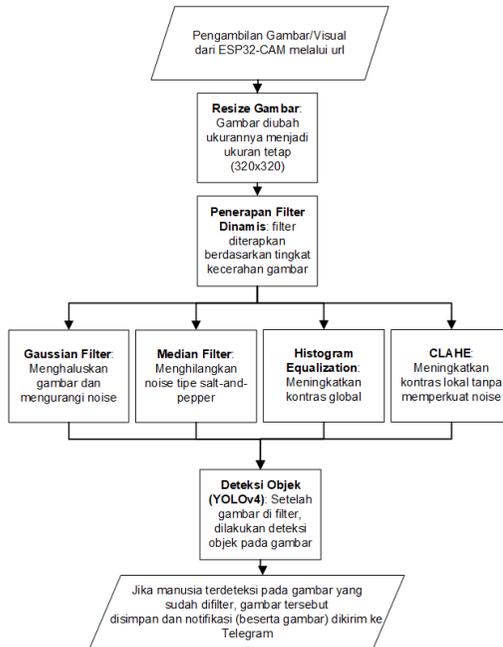
Gambar 12 Ilustrasi rancangan hasil akhir alat

5. Pengembangan Perangkat Lunak

Berikutnya adalah mengembangkan perangkat lunak yang diperlukan untuk menjalankan sistem. Pengembangan perangkat lunak mencakup penulisan kode untuk *firmware* ESP32-CAM, skrip pemrosesan gambar dan deteksi objek menggunakan YOLOv4, serta integrasi dengan aplikasi Telegram. Perangkat lunak dikembangkan menggunakan berbagai alat dan platform, termasuk Arduino IDE atau PlatformIO untuk *firmware*, Python untuk pemrosesan gambar, dan OpenCV serta Scikit-Image untuk teknik *Pre-processing*.

6. Implementasi Sistem

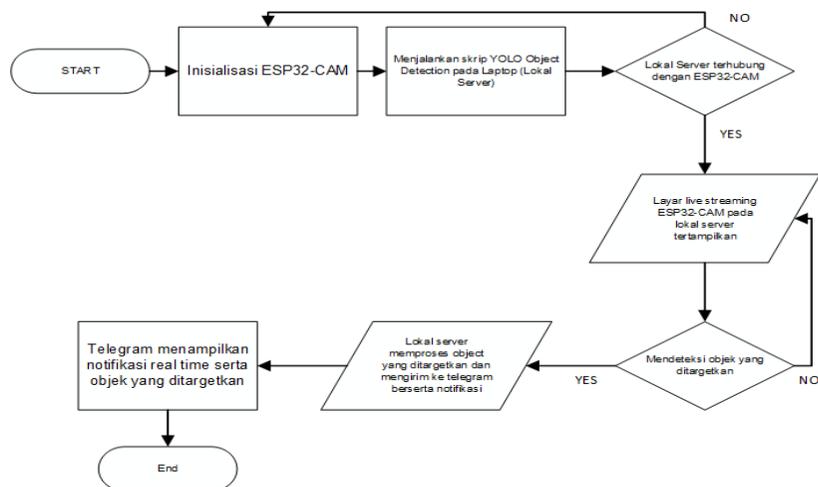
Pada tahap ini, perangkat keras dan perangkat lunak yang telah dikembangkan diintegrasikan menjadi satu sistem yang utuh. ESP32-CAM diprogram untuk menangkap gambar ketika skrip YOLO *Object Detection* pada *local server* mendeteksi gerakan. Gambar yang diambil diproses lebih lanjut menggunakan algoritma YOLOv4 dan teknik *pre-processing*. Hasil deteksi objek dikirim ke aplikasi Telegram untuk ditampilkan kepada pengguna.



Gambar 13 Blok sistem pengolahan citra

7. Pengujian Sistem

Setelah sistem diimplementasikan, langkah berikutnya adalah menguji sistem untuk memastikan bahwa semua komponen bekerja dengan baik dan sesuai dengan spesifikasi yang telah ditentukan. Pengujian dilakukan untuk mengevaluasi kinerja sistem, termasuk keakuratan deteksi objek, kualitas gambar setelah *pre-processing*, dan kecepatan respon sistem. Pengujian dilakukan dalam berbagai kondisi untuk memastikan sistem dapat diandalkan.



Gambar 14 Diagram alir sistem

8. Analisis Data dan Evaluasi

Data yang diperoleh selama pengujian dianalisis untuk mengevaluasi kinerja sistem. Parameter yang dianalisis meliputi *Signal to Noise Ratio* (SNR), *Peak Signal to Noise Ratio* (PSNR), *Mean Squared Error* (MSE), *Structural Similarity Index* (SSIM), *precision*, *recall* dan *F1-score*. Hasil analisis digunakan untuk mengevaluasi efektivitas teknik *pre-processing* dan keakuratan deteksi objek oleh YOLOv4 dibandingkan dengan tidak memakai teknik *pre-processing*.

9. Kesimpulan

Merumuskan kesimpulan setelah data diperoleh dan dianalisis.

2.5 Teknik Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui beberapa tahap yang melibatkan studi literatur, pengambilan gambar, penerapan teknik *pre-processing*, deteksi objek, dan pengukuran berbagai parameter kualitas gambar serta performa deteksi. Berikut adalah penjelasan langkah-langkah pengumpulan data:

1. Studi Literatur

Metode studi literatur melibatkan beberapa langkah utama. Pertama, perlu mengidentifikasi sumber informasi yang relevan, seperti buku, jurnal ilmiah, artikel, dan dokumen resmi yang terkait dengan topik penelitian, seperti penggunaan Algoritma YOLO dalam mendeteksi objek dan penerapannya pada alat seperti ESP32-CAM. Setelah menemukan sumber informasi yang relevan, perlu menyusun informasi tersebut menjadi ringkasan sistematis yang sesuai dengan struktur penelitian. Selanjutnya, harus menganalisis informasi yang telah dirangkum untuk mendukung argumen dan temuan penelitian, serta mengidentifikasi pola yang muncul dari berbagai sumber yang telah ditelusuri. Setelah itu menyusun daftar pustaka yang mencantumkan semua sumber yang digunakan dalam studi literatur, dengan format yang sesuai dengan pedoman penulisan yang digunakan, seperti APA, MLA, atau IEEE.

2. Pengambilan Gambar

Gambar diambil menggunakan modul ESP32-CAM yang membuat URL untuk menghubungkan fungsi kamera ESP32-CAM. URL kemudian dimasukkan ke skrip YOLO Object Detection yang berada pada *local server* untuk penerapan *filter pre-processing* gambar.

3. Penerapan Teknik *Pre-processing*

Gambar yang diambil oleh ESP32-CAM kemudian diproses menggunakan berbagai teknik *pre-processing*.

- 1) *Gaussian filter*: Menghaluskan gambar untuk mengurangi *noise*.
- 2) *Median filter*: Menghilangkan *noise* tipe *salt and pepper* tanpa mengaburkan tepi objek.
- 3) *Histogram Equalization*: Meningkatkan kontras global gambar.
- 4) CLAHE (*Contrast Limited Adaptive Histogram Equalization*): Meningkatkan kontras lokal tanpa memperkuat *noise*.

4. Deteksi Objek

Setelah gambar diproses dengan teknik *pre-processing*, algoritma YOLOv4 digunakan untuk mendeteksi objek dalam gambar. Proses deteksi ini menghasilkan *bounding box*, label objek, dan *confidence score* untuk setiap objek yang terdeteksi.

5. Pengukuran Parameter Kualitas Gambar dan Akurasi Deteksi Objek

Untuk mengevaluasi efektivitas teknik *pre-processing*, beberapa parameter kualitas gambar diukur:

- 1) *Signal to Noise Ratio* (SNR): Mengukur tingkat sinyal terhadap tingkat *noise* dalam gambar.
 - 2) *Peak Signal to Noise Ratio* (PSNR): Mengukur kualitas rekonstruksi gambar dengan membandingkan gambar asli dan gambar yang telah diproses.
 - 3) *Mean Squared Error* (MSE): Mengukur perbedaan rata-rata kuadrat antara gambar asli dan gambar yang telah diproses.
 - 4) *Structural Similarity Index* (SSIM): Mengukur kesamaan struktural antara gambar asli dan gambar yang telah diproses.
- #### 6. Pengukuran Metriks *Performance* Deteksi Objek YOLOv4

Untuk mengevaluasi kinerja deteksi objek oleh YOLOv4, beberapa parameter akurasi deteksi diukur:

- 1) *Precision*: Mengukur ketepatan deteksi objek.
 - 2) *Recall*: Mengukur kemampuan YOLOv4 dalam mendeteksi semua objek yang ada.
 - 3) *F1-score*: Mengukur keseimbangan antara *precision* dan *recall*.
- #### 7. Pengujian Secara *Realtime* dalam Berbagai Kondisi

Pengujian dilakukan secara *realtime* dalam berbagai kondisi pencahayaan di dalam dan diluar ruangan untuk memastikan sistem bekerja dengan baik di berbagai situasi.

8. Analisis Data

Data yang dikumpulkan dianalisis untuk mengevaluasi efektivitas teknik *pre-processing* gambar dan kinerja deteksi objek. Analisis ini mencakup perbandingan nilai SNR, PSNR, MSE, dan SSIM sebelum dan setelah penerapan *pre-processing*, serta analisis hasil deteksi objek.

9. Integrasi dengan Aplikasi Telegram

Hasil deteksi objek serta pesan notifikasi dikirim ke aplikasi Telegram untuk ditampilkan kepada pengguna. Data yang ditampilkan mencakup gambar objek yang terdeteksi.

2.6 Teknik Analisis Data

Teknik analisis data dalam penelitian ini menggunakan analisis deskriptif dan statistik sederhana. Analisis deskriptif digunakan untuk memberikan gambaran umum tentang performa sistem, seperti waktu pemrosesan dan kualitas gambar. Analisis akurasi deteksi dilakukan untuk mengevaluasi kinerja algoritma YOLO yang dimodifikasi. Analisis statistik sederhana, seperti perhitungan *precision*, *recall* dan *F1-score*. Program komputer yang digunakan dalam analisis ini adalah Python, dengan *library* NumPy untuk analisis data, serta beberapa *library* lain yang menunjang untuk visualisasi hasil analisis. Pemilihan Python didasarkan pada kemampuannya yang kuat dan fleksibel, serta dukungan banyak *library* yang mendukung analisis data dan visualisasi. Analisis data dilakukan dengan menggabungkan hasil pengukuran langsung, observasi, dan pengujian sistem untuk memberikan gambaran komprehensif tentang kinerja dan reliabilitas sistem.