

**SISTEM PENDETEKSI KERUSAKAN PADA BANGUNAN RUMAH  
MENGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO)**

**LAPORAN TUGAS AKHIR**

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh gelar sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung



**DISUSUN OLEH :**

**AKHIDATUL WARDA**

**NIM 32602000012**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS ISLAM SULTAN AGUNG**

**SEMARANG**

**2024**

***FINAL PROJECT***

***DAMAGE DETECTION SYSTEM IN HOUSE BUILDINGS USING THE  
YOU LOOK ONCE (YOLO) ALGORITHM***

*Proposed to complete the requirement to obtain a bachelor's degree (S1)  
at Informatics Engineering Department Of Industrial Technology Faculty  
Sultan Agung Islamic University*



*Arranged By :*

**AKHIDATUL WARDA**

**NIM 32602000012**

***MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG***

**2024**

## LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul “ Sistem Pendeteksi Kerusakan Pada Bangunan Rumah Menggunakan Algoritma *You Only Look Once (YOLO)* ”

ini disusun oleh :

Nama : Akhidatul Warda

NIM : 32602000012

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari : Kamis

Tanggal : 05 September 2024

Mengesahkan,

Pembimbing I

Pembimbing II

  
Sam Farisa Chaerul Maviana, ST, M.Kom

NIK. 210615046

  
Dedy Kurniadi, ST, M.Kom

NIK. 210615048

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Sultan Agung

  
Moch Taufik, ST, MIT

NIK. 210604034

## LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul “**Sistem Pendeteksi Kerusakan Pada Bangunan Rumah Menggunakan Algoritma *You Only Look Once (YOLO)***” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :


Hari : *Rabu*

Tanggal : *04 September 2024*

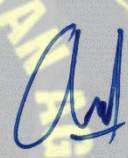
### TIM PENGUJI

Ketua Penguji

Anggota Penguji

  
Moch. Taufik, ST, MIT

NIK. 210604034

  
Andi Risnyah, ST, M.Kom

NIK. 210616053

UNISSULA

جامعة سلطان ابي جعفر الإسلامية

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Akhidatul Warda

NIM : 32602000012

Judul Tugas Akhir : SISTEM PENDETEKSI KERUSAKAN PADA BANGUNAN RUMAH MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 04 September 2024

Yang menyatakan,

  
Akhidatul Warda



## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Akhidatul Warda

NIM : 32602000012

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :  
SISTEM PENDETEKSI KERUSAKAN PADA BANGUNAN RUMAH  
MENGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO)

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 04 September 2024

Yang menyatakan,

  
  
Akhidatul Warda

## KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayahnya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul “Sistem Pendeteksi Kerusakan Pada Bangunan Rumah Menggunakan Algoritma *You Only Look Once* (YOLO)” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang. Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis. Oleh karena itu, saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H., yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana S.T., M.T., IPU., ASEAN.Eng.
3. Dosen Pembimbing I, Bapak Sam Farisa Chaerul Haviana, ST. Kom yang telah membimbing dan memberikan banyak nasehat dan saran.
4. Dosen Pembimbing II, Bapak Dedy Kurniadi, ST.M.Kom yang telah membimbing dan memberikan banyak nasehat dan saran.
5. Orang tua dan keluarga penulis yang telah mendukung, mendoakan dan mengizinkan untuk menyelesaikan laporan ini.
6. Kepada teman – teman dan semua pihak yang tidak dapat penulis sebutkan satu persatu atas doa, dukungan serta bantuannya.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 23 Agustus 2024

Akhidatul Warda

## DAFTAR ISI

<b>COVER .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN PEMBIMBING .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN PENGUJI.....</b>	<b>iv</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR .....</b>	<b>v</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>ABSTRAK .....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah .....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI .....</b>	<b>5</b>
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori.....	8
2.2.1 Kerusakan Pada Bangunan Rumah .....	8
2.2.2 <i>Object detection</i> .....	10
2.2.3 <i>Convolution Neural Networks</i> .....	11
2.2.4 <i>You Only Look Once (YOLO)v7</i> .....	14
2.2.5 <i>PyTorch</i> .....	17
<b>BAB III METODE PENELITIAN .....</b>	<b>18</b>
3.1 Studi Literatur.....	18
3.2 Metode Penelitian.....	18
3.2.1 Pengumpulan Data .....	19
3.2.2 <i>Pre-processing Data</i> .....	20



3.2.3	<i>Split Data</i> .....	21
3.2.4	Anotasi Data.....	22
3.2.5	Pelatihan Model .....	23
3.2.6	Evaluasi Model.....	24
3.3	<i>Deployment</i> model.....	25
<b>BAB IV HASIL DAN ANALISIS PENELITIAN .....</b>		<b>26</b>
4.1	Hasil.....	26
4.1.1	Hasil Pelatihan Model.....	26
4.1.2	Tampilan aplikasi .....	28
4.1.3	Pengujian Sistem.....	31
4.2	Analisis .....	38
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>40</b>
5.1	Kesimpulan.....	40
5.2	Saran.....	41
<b>DAFTAR PUSTAKA</b>		



## DAFTAR GAMBAR

Gambar 2. 1 <i>Wall crack</i> .....	9
Gambar 2. 2 <i>Damp stain</i> .....	10
Gambar 2. 3 Lapisan CNN (Rahmawati dkk., 2024).....	11
Gambar 2. 4 proses <i>convolution layer</i> (Dahman, 2021) .....	12
Gambar 2. 5 konsep <i>pooling layer</i> .....	13
Gambar 2. 6 <i>Fully connected</i> .....	14
Gambar 2. 7 arsitektur yolov7 (Mostafa dkk., 2023).....	15
Gambar 3. 1 <i>Flowchart</i> alur penelitian .....	18
Gambar 3. 2 Data gambar dan anotasi <i>damp stain</i> .....	19
Gambar 3. 3 Data gambar dan anotasi <i>wall crack</i> .....	19
Gambar 3. 4 Proses Anotasi Data .....	22
Gambar 3. 5 <i>Flowchart</i> pelatihan model.....	23
Gambar 3. 6 Alur sistem .....	25
Gambar 4. 1 Matrik <i>training</i> dan evaluasi .....	26
Gambar 4. 2 Hasil akhir pelatihan.....	27
Gambar 4. 3 <i>F1 score</i> .....	27
Gambar 4. 4 Tampilan aplikasi pada PC.....	28
Gambar 4. 5 Tampilan aplikasi pada Ponsel.....	29
Gambar 4. 6 <i>upload</i> gambar .....	29
Gambar 4. 7 <i>upload</i> gambar pada hp .....	30
Gambar 4. 8 Hasil <i>predict wall crack</i> .....	30
Gambar 4. 9 hasil <i>predict damp stain</i> .....	31

## DAFTAR TABEL

Tabel 2. 1 Tabel tinjauan pustaka .....	7
Tabel 3. 1 Tabel <i>split data</i> .....	21
Tabel 4. 1 Hasil pelatihan model.....	27
Tabel 4. 2 Pengujian <i>damp stain</i> .....	32
Tabel 4. 3 Pengujian <i>wall crack</i> .....	35



## ABSTRAK

Kerusakan pada bangunan, seperti *damp stain* (noda bekas air) dan *wall crack* (retakan dinding), merupakan jenis kerusakan yang biasa terjadi pada bangunan rumah. Penelitian ini bertujuan untuk mengimplementasikan sistem deteksi otomatis menggunakan model *You Only Look Once* (YOLO)v7 yang dapat mengidentifikasi dan mengklasifikasikan dua jenis kerusakan ini dari gambar. Pelatihan model menggunakan *dataset* berjumlah 820 gambar yang terdiri dari dua kelas yaitu *damp stain* dan *wall crack*, memperoleh hasil nilai *precision* sebesar 0.681 dan *recall* sebesar 0.619 dengan nilai *mAP@0.5* sebesar 0.681. Hasil tersebut menunjukkan bahwa model memiliki performa yang cukup baik dalam deteksi objek pada *threshold* IoU 0.5. Namun, nilai *mAP@0.5:0.95* yang lebih rendah pada 0.311 mengindikasikan penurunan efektivitas deteksi pada berbagai level *threshold*. Analisis menunjukkan bahwa model dapat mendeteksi kerusakan sesuai dengan jenisnya, akan tetapi dari beberapa percobaan yang telah dilakukan model juga mendeteksi objek yang seharusnya tidak terdeteksi atau masih terdapat *noise*, hal ini teridentifikasi adanya *overfitting* pada model, dimana model cenderung terlalu sensitif terhadap pola tertentu pada data latih sehingga menghasilkan prediksi yang kurang akurat pada data uji. Fluktuasi pada *validation loss* juga menunjukkan adanya ketidakstabilan performa model dalam menghadapi data yang bervariasi.

**Kata Kunci** : Deteksi, *You Only Look Once*, *Damp stain*, *Wall Crack*

## ABSTRACT

*Damage to buildings, such as damp stains and wall cracks, is a common type of damage that occurs in residential structures. This research aims to implement an automatic detection system using the You Only Look Once (YOLO)v7 model that can identify and classify these two types of damage from images. The model training using a dataset of 820 images consisting of two classes, namely damp stain and wall crack, achieved a precision value of 0.681 and a recall of 0.619, with a mAP@0.5 value of 0.681. These results indicate that the model performs quite well in object detection at an IoU threshold of 0.5. However, the lower mAP@0.5:0.95 value of 0.311 suggests a decrease in detection effectiveness at various threshold levels. The analysis shows that the model can detect damage according to its type; however, from several experiments that have been conducted, the model also detects objects that should not be detected or still contains noise. This indicates the presence of overfitting in the model, where the model tends to be overly sensitive to certain patterns in the training data, resulting in less accurate predictions on the test data. Fluctuations in validation loss also indicate instability in the model's performance when facing varied data.*

**Keywords** : *Detection, You Only Look Once, Damp stain, Wall Crack.*

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Pada saat ini, teknologi telah menjadi bagian yang tidak terpisahkan dari berbagai komponen kehidupan. Seperti dalam pemeliharaan bangunan rumah yang merupakan aspek penting karena tidak hanya menjadi kualitas bangunan tetapi juga menjamin keamanan dan kenyamanan penghuninya. Kerusakan yang biasa dijumpai pada tembok dan langit – langit rumah seperti retakan dan noda lembab, dapat diidentifikasi adanya masalah setruktural yang dapat mempengaruhi kualitas bangunan. Pentingnya mengidentifikasi kerusakan lebih dini sehingga secepatnya dapat dilakukan perbaikan guna mencegah kerusakan lebih lanjut serta memastikan kondisi bangunan rumah tetap baik dan terjaga.

Retakan pada tembok merupakan salah satu jenis kerusakan struktural yang umumnya terjadi pada bangunan. Retakan terjadi dapat terjadi karena berbagai faktor, seperti tanah yang berada dibawah bangunan yang bermasalah, penggunaan bahan material pembuatan bangunan dengan kualitas rendah, atau perubahan suhu yang ekstrim. Dengan adanya retakan dapat dicurigai adanya kerusakan yang lebih serius, seperti amblesnya fondasi bangunan atau struktur bangunan yang tidak stabil. *Damp stain* atau biasa dikenal dengan noda lembab yang biasa terdapat pada tembok dan langit – langit rumah indikasi adanya kelembapan pada bangunan, masalah ini muncul diakibatkan kebocoran air seperti kebocoran pipa atau kebocoran atap mengakibatkan air meresap pada material bangunan sehingga meninggalkan noda, kemudian uap air yang berubah jadi embun pada permukaan tembok, kelembapan dari lingkungan seperti kelembapan tanah, air tanah yang meresap melalui fondasi merambat ke tembok dapat mengakibatkan *damp stain* (Khatab dkk., 2022). Tembok rumah yang lembab karena mengandung air tidak hanya mempengaruhi tampilan bangunan rumah tetapi juga dapat mengakibatkan adanya jamur dan lumut, serta kerusakan kualitas bangunan.

Maka dari itu pentingnya untuk identifikasi serta penanganan lebih awal untuk menjaga kualitas bangunan serta menjaga kesehatan lingkungan (Chundakus Habsya dkk., 2020). Teknologi modern seperti deteksi kerusakan pada bangunan rumah secara otomatis diperlukannya sistem dengan kuliatas yang baik, dengan menggunakan *You Only Look Once* (YOLO), YOLO yang merupakan algoritma dari *Convolution Neural Networks* (CNN), CNN sendiri adalah salah satu metode dari deep learning yang telah banyak digunakan untuk berbagai proyek deteksi objek (Sistem, 2021), sebagai salah satu algoritma dari CNN, YOLO telah terbukti sangat efisien dalam melakukan pendeteksian objek pada gambar, YOLO sangat cocok untuk digunakan dalam sistem pendeteksi kerusakan pada bangunan rumah, karena dapat mendeteksi objek dengan cepat dan akurat serta memberikan hasil yang terbaik.(Arif dkk., 2023).

Dengan sistem deteksi otomatis ini, proses pemeliharaan bangunan rumah dapat dilakukan dengan lebih efektif. Identifikasi kerusakan bangunan berupa retakan dan noda lembab dapat dilakukan dengan cepat dan akurat, sehingga menghemat waktu dan energi yang dikeluarkan untuk pemeriksaan secara manual serta dapat melakukan penanganan lebih awal sebelum terjadi kerusakan yang lebih serius. Berharap dengan sistem ini dapat meningkatkan efesiensi pemeriharaan bangunan, mengurangi risiko kerusakan pada struktural yang fatal, serta dapat memastikan keadaan lingkungan yang sehat, aman dan nyaman bagi penghuninya.

## 1.2 Perumusan Masalah

Bagaimana algoritma *You Only Look Once* (YOLO) dapat melakukan deteksi kerusakan pada bangunan rumah secara akurat ?

### 1.3 Pembatasan Masalah

Berikut batasan masalah dari penulisan proposal tugas akhir :

1. Hanya dapat mendeteksi dua jenis kerusakan pada bangunan rumah, yaitu retakan pada tembok serta noda lembab pada tembok dan langit – langit. Untuk kerusakan lain seperti kerusakan pada struktur bangunan atau fondasi, lantai dan lainnya tidak termasuk dalam sistem.
2. Dapat mendeteksi kerusakan pada bangunan didalam maupun diluar ruangan.
3. Hanya dapat mendeteksi gambar dengan pencahayaan yang baik. Kondisi pencahayaan yang buruk tidak dipertimbangkan pada pengembangan sistem ini.
4. Pada penelitian ini hanya dapat mendeteksi objek yang berbentuk gambar saja, sistem tidak mendukung untuk melakukan deteksi objek secara real time ataupun video.
5. Menggunakan metode CNN dengan algoritma YOLO versi 7.

### 1.4 Tujuan

Tujuan dari tugas akhir ini adalah bagaimana algoritma *You Only Look Once* (YOLO) dapat mendeteksi gambar berupa kerusakan pada bangunan rumah.

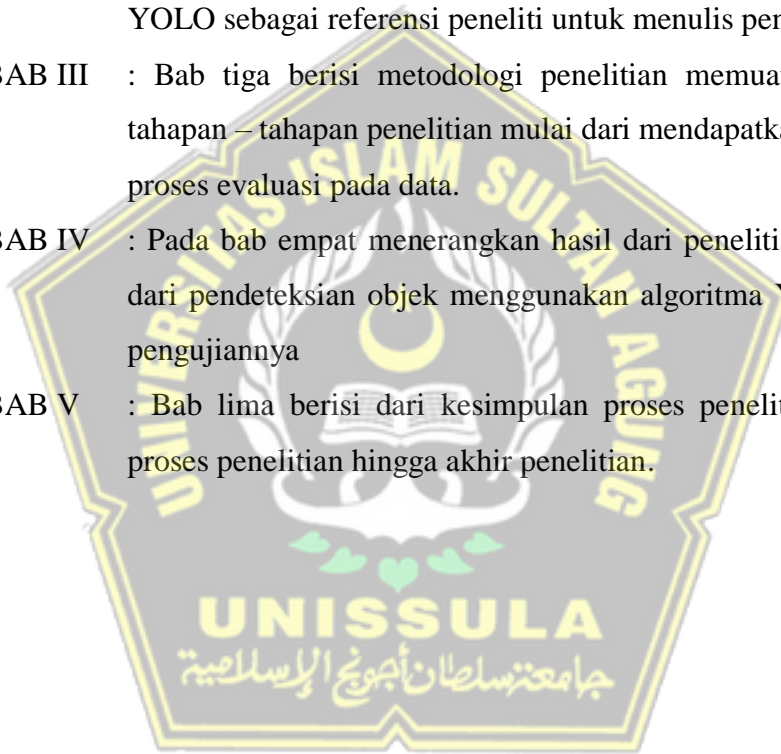
### 1.5 Manfaat

Algoritma *You Only Look Once* (YOLO) dapat dimanfaatkan untuk mendeteksi gambar kerusakan pada bangunan rumah secara otomatis.

## 1.6 Sistematika Penulisan

Pada penyusunan laporan tugas akhir ini digunakan sistematika penulisan sebagai berikut :

- BAB I : Pada bab satu menjelaskan mengenai latar belakang pemilihan judul, perumusan masalah, pembatasan masalah, tujuan penelitian, serta sistematika penulisan
- BAB II : Bab dua berisi tentang penjelasan pada penelitian – penelitian sebelumnya dan dasar teori yang berhubungan dengan algoritma YOLO sebagai referensi peneliti untuk menulis penelitian ini.
- BAB III : Bab tiga berisi metodologi penelitian memuat proses serta tahapan – tahapan penelitian mulai dari mendapatkan data hingga proses evaluasi pada data.
- BAB IV : Pada bab empat menerangkan hasil dari penelitian yaitu, hasil dari pendeteksian objek menggunakan algoritma YOLO beserta pengujiannya
- BAB V : Bab lima berisi dari kesimpulan proses penelitian dari awal proses penelitian hingga akhir penelitian.





## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Untuk membuat sistem *object detection* ada beberapa metode yang dapat digunakan, salah satunya *Color Blob Detection*. Seperti pada penelitian yang dilakukan oleh (Adianto dkk., 2024) mengimplementasikan metode *color blob detection* untuk melakukan *object detection* pada daun sawi, kemudian hasilnya akan diukur dengan memanfaatkan confusion matrix yang digunakan untuk menguji akurasi pada sistem. Saat melakukan pengujian sistem dengan cara menguji pada jarak 20 cm pada saat mendeteksi daun sawi mendapatkan hasil mencapai 50%. Dalam perhitungan akurasi, *precision*, *recall*, dan *F1 score* menampilkan gambaran model hingga mencapai akurasi 66,6%, *Precision* 71%, *recall* 62,5% serta *F1 score* 67%.

Selain itu ada juga penelitian lainnya dari (Fajriansyah dkk., 2023) pada penelitian ini menggunakan metode RetinaNet yang digunakan untuk otomatisasi deteksi pada penyakit tumbuhan. Pada penelitian ini juga menggunakan sistem segmentasi citra, yaitu analisis *bold* serta metode pemodelan warna, untuk menyiapkan data *training* dan data *testing*. Dengan menggunakan data *training* dan data *testing* jaringan saraf tiruan akan dilatih agar dapat memahami pola dan fitur yang ada pada citra penyakit tumbuhan. Kemudian, dataset citra tanaman yang berjumlah dua kelas yaitu sehat dan terinfeksi penyakit, dengan menggunakan dataset tersebut kinerja sistem akan diuji, sehingga akan menghasilkan sistem otomatisasi deteksi pada penyakit tumbuhan.

YOLO merupakan salah satu algoritma dari metode CNN yang juga dapat mendeteksi objek berupa gambar ataupun video. Salah satu penelitian yang memanfaatkan algoritma YOLO adalah (Yasen dkk., 2023), dalam penelitian ini metode yolo digunakan untuk mendeteksi hama dan penyakit yang ada pada daun cabai. Berikut empat kelas yang digunakan pada penelitian ini diantaranya daun cabai yang terkena penyakit kuning, hama thrips

bercak daun, serta daun yang sehat. Kelas akan diuji menggunakan model yolo yang menghasilkan *training* data sebesar 73% dengan *epoch* 150. Untuk nilai dari *precision* memperoleh 77,4%, kemudian untuk nilai *recall* 67,1%, serta mAP mendapatkan 75,1%.

Penelitian lain yang menggunakan YOLO sebagai algoritmanya adalah penelitian dari (Sarosa dan Muna, 2021), penelitian ini berisi tentang penerapan algoritma YOLO untuk mendeteksi korban bencana alam dengan menggunakan pengolahan citra yang bertujuan untuk membantu tim SAR dalam melakukan tugas evakuasi korban bencana. Ada dua macam algoritma yang dipakai pada penelitian ini yaitu YOLOv3 dan juga YOLOv3 Tiny, dengan 200 gambar sebagai dataset yang dibagi menjadi 100 gambar sebagai data *training* dan 100 gambar sebagai data *testing*. Kemudian diuji dan menghasilkan mencapai F1 *Score* memperoleh 95,3% dari algoritma YOLOv3.

Kemudian penelitian dari (Asni dkk., 2021) pada penelitian ini untuk mengetahui kemampuannya dalam melakukan deteksi algoritma YOLO akan diuji menggunakan empat jenis kendaraan diantaranya motor, mobil, bus dan truck dengan menggunakan 15 data *testing* yang kemudian akan dilakukan pengujian. Hasil dari pengujian menghasilkan akurasi 98,667%, dengan mendeteksi dari 64 gambar dari total keseluruhan 66 gambar yang berasal dari 15 sampel, dan terdapat satu sampel memperoleh akurasi sebesar 82%, dengan rata-rata waktu komputasi pada saat melakukan pendeteksian objek 3,067% detik.

Menurut (Kamil, 2023) metode YOLO dapat melakukan pendeteksian objek secara real time serta dapat menciptakan hasil yang lebih akurat dibanding dengan metode lain. Pada penelitian ini melakukan *object detection* pada kualitas dari biji kopi. Untuk percobaan deteksi dilakukan sebanyak dua kali yaitu, dengan percobaan sesuai pencahayaan (low, normal, dan high) serta percobaan sesuai dengan jarak (jarak 5 cm, 25 cm, dan 50 cm). Untuk hasil dari *testing* dari pencahayaan mendapatkan akurasi sebesar 83,33% dan 66,66% untuk hasil dari jarak. Dalam 12 percobaan,

menghasilkan 9 yang mampu mengidentifikasi biji kopi dengan akurat, dan 3 yang tidak dapat terdeteksi. Tingkat akurasi pada deteksi pada kualitas biji kopi memiliki nilai rata – rata 75%.

Tabel 2. 1 Tabel tinjauan pustaka

No	Peneliti	Model	Hasil
1.	(Adianto dkk., 2024)	Color Blob	Dalam perhitungan akurasi, <i>precision</i> , <i>recall</i> , dan <i>F1 score</i> menampilkan gambaran model hingga mencapai akurasi 66,6%, <i>Precision</i> 71%, <i>recall</i> 62,5% serta <i>F1 score</i> 67%.
2.	(Fajriansyah dkk., 2023)	RetinaNet	Metode RetinaNet dalam pengolahan citra sebagai teknik pengenalan objek yang digunakan untuk mendeteksi serta mengklasifikasi objek.
3.	(Yasen dkk., 2023)	YOLO	Untuk nilai dari <i>precision</i> memperoleh 77,4%, kemudian untuk nilai <i>recall</i> 67,1%, serta <i>mAP</i> mendapatkan 75,1%.
4.	(Sarosa dan Muna, 2021)	YOLOv3	dengan 200 gambar sebagai dataset yang dibagi menjadi 100 gambar sebagai data <i>training</i> dan 100 gambar sebagai data <i>testing</i> . Kemudian diuji dan menghasilkan mencapai <i>F1 Score</i> memperoleh 95,3%.
5.	(Asni dkk., 2021)	YOLOv1	Hasil dari pengujian menghasilkan akurasi 98,667%, dengan rata – rata waktu komputasi pada saat melakukan pendeteksian objek 3,067% detik.
6.	(Kamil, 2023)	YOLO	Untuk hasil dari <i>testing</i> dari pencahayaan mendapatkan akurasi sebesar 83,33% dan 66,66% untuk hasil dari jarak. Tingkat akurasi pada deteksi pada kualitas biji kopi memiliki nilai rata – rata 75%.

No	Peneliti	Model	Hasil
7.	(Guntara, 2023)	YOLOv7	Mengenai deteksi atap bangunan berbasis citra udara menggunakan algoritma yolov7 menunjukkan hasil dari pengujian model mendapatkan skor F1 mencapai 80% dengan pengujian 200 gambar.
8.	(Nayottama dan dkk., 2023)	YOLOv7	Pada penelitian ini menggunakan model yolov7, pada pelatihan model menghasilkan nilai rata – rata sebesar 80%.
9.	(Passa dkk., 2023)	YOLOv7	Model yang dihasilkan pada penelitian deteksi tumor otak mencapai mAP sebesar 93,2%, <i>Precision</i> 91,4%, serta F1-Score 90,8%.
10.	(Giarida dkk., 2024)	YOLO	Untuk keseluruhan interval kecepatan, pada penelitian ini mendapatkan akurasi rata – rata sebesar 96,05%, kemudian untuk nilai eror 3,93%, serta nilai presesi sebesar 98,98%.

## 2.2 Dasar Teori

### 2.2.1 Kerusakan Pada Bangunan Rumah

Kerusakan bangunan rumah merupakan masalah biasa terjadi sehingga dapat memengaruhi kualitas struktur dan juga tampilan suatu bangunan. Retakan dinding serta noda bekas air (*damp stain*) pada dinding dan langit – langit rumah kerusakan yang biasa dijumpai serta menjadi fokus pada penelitian ini, kedua jenis kerusakan ini memiliki ciri yang berbeda juga terjadinya kerusakan ini mempunyai sebab yang beragam.

#### 1. *Wall crack* (retakan dinding)

Pecahan atau celah yang biasa terdapat pada permukaan dinding atau biasa dikenal dengan retakan tembok dibagi menjadi dua jenis yaitu keretakan struktural dan keretakan non – struktural. Retakan juga disebabkan oleh berbagai faktor, yaitu keadaan tanah yang tidak setabil

membuat pergerakan fondasi pada bangunan yang mengakibatkan tegangan pada tembok sehingga menghasilkan retakan, kelembapan lingkungan mengakibatkan material pada dinding menguai serta menyusut sehingga memicu terjadinya retakan, bahan bangunan yang digunakan dengan kualitas rendah, kemudian faktor lingkungan, alam seperti gempa bumi, dan masih banyak faktor lainnya (Arianto dan Zuliarso, 2023).



Gambar 2. 1 Wall crack

Gambar 2.1 menampilkan salah satu contoh kerusakan bangunan berupa retakan tembok merukan data yang digunakan dalam penelitian ini.

2. *Damp stains* (noda bekas air)

Kemudian adanya pola bekas air yang terdapat pada tembok dan langit-langit bangunan sebagai penanda adanya kebocoran, seperti kebocoran pipa atau saluran air, kebocoran atap. Faktor lainnya adalah pada lingkungan seperti lingkungan yang lembab mengakibatkan kondensasi pada permukaan tembok. Apabila hal tersebut terjadi secara terus menerus tidak hanya mengakibatkan menimbulkan pola bekas air tetapi juga dapat mengakibatkan tumbuhnya jamur yang berbahaya bagi penghuni rumah (Maharani dkk., 2023).



Gambar 2. 2 *Damp stain*

Gambar 2.2 menampilkan kerusakan bangunan berupa *damp stain* (noda bekas air), dalam kelas ini pola kerusakannya yang beragam dan tidak terpatok dalam satu bentuk pola saja.

### 2.2.2 *Object detection*

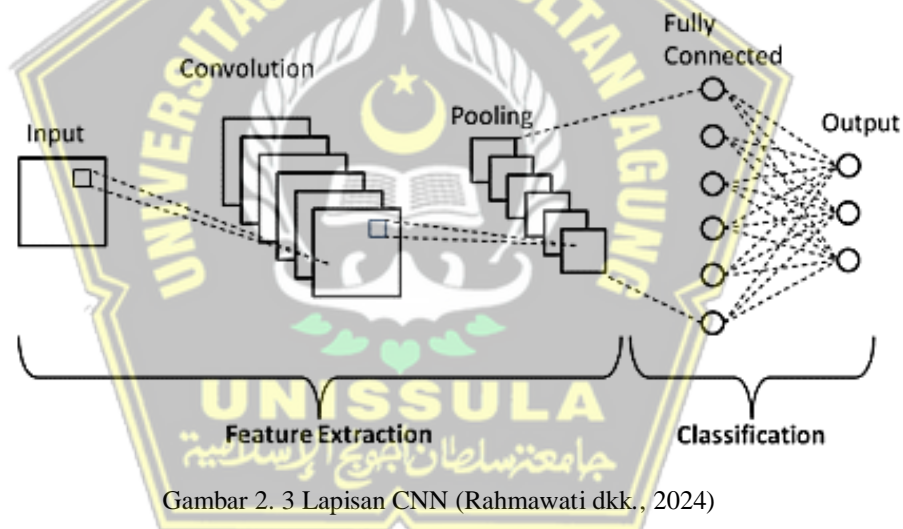
*Object detection* merupakan *task* penting dalam bidang *computer vision* yang digunakan untuk mengidentifikasi serta membatasi setiap bagian – bagian dari objek dengan membuat kotak pembatas (*bounding box*) pada sekitar objek yang terkumpul dalam kategori atau kelas tertentu yang berupa gambar ataupun video, sehingga dapat mengetahui objek mana yang akan dideteksi. *Object detection* tidak hanya digunakan untuk mengidentifikasi objek, tetapi juga digunakan untuk menetapkan letak relatif dari kelas serta objek – objek tertentu yang berupa gambar ataupun video. Maka dari itu, *object detection* dapat memungkinkan untuk melakukan pemisahan, pengklasifikasian, serta menemukan objek – objek secara otomatis yang berupa gambar ataupun video. (Alfarizi dkk., 2023).

*Object detection* digunakan untuk mengidentifikasi serta menetapkan letak objek – objek tertentu yang terdapat pada gambar ataupun video. Sistem kerja dari *object detection* dengan memperoleh *input* gambar dan menghasilkan *output* dengan menunjukkan hasil dari klasifikasi objek pada gambar, kemudian menentukan letak dari objek menggunakan *bounding box*. *object detection* biasanya digunakan untuk beberapa jenis objek seperti, deteksi pada manusia, deteksi pada kendaraan, serta deteksi pada bangunan.

Dengan seiring perkembangan teknologi, *object detection* semakin banyak digunakan di berbagai bidang, termasuk dalam bidang properti (Tiyar dan Fudholi, 2021).

### 2.2.3 Convolution Neural Networks

*Convolution Neural Networks* (CNN) adalah salah satu metode dari *deep learning* yang memungkinkan melakukan proses pembelajaran mandiri dalam melakukan pengenalan objek, ekstrasi fitur, serta klasifikasi objek. CNN merupakan metode yang dikembangkan dari arsitektur *Multi Layer Perceptron* (MLP) yang digunakan untuk mengolah data yang berbentuk gambar (Husna dkk., 2022). Untuk sistem kerja, CNN mempunyai tiga lapisan yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Untuk setiap *layer* mempunyai tugasnya masing – masing.

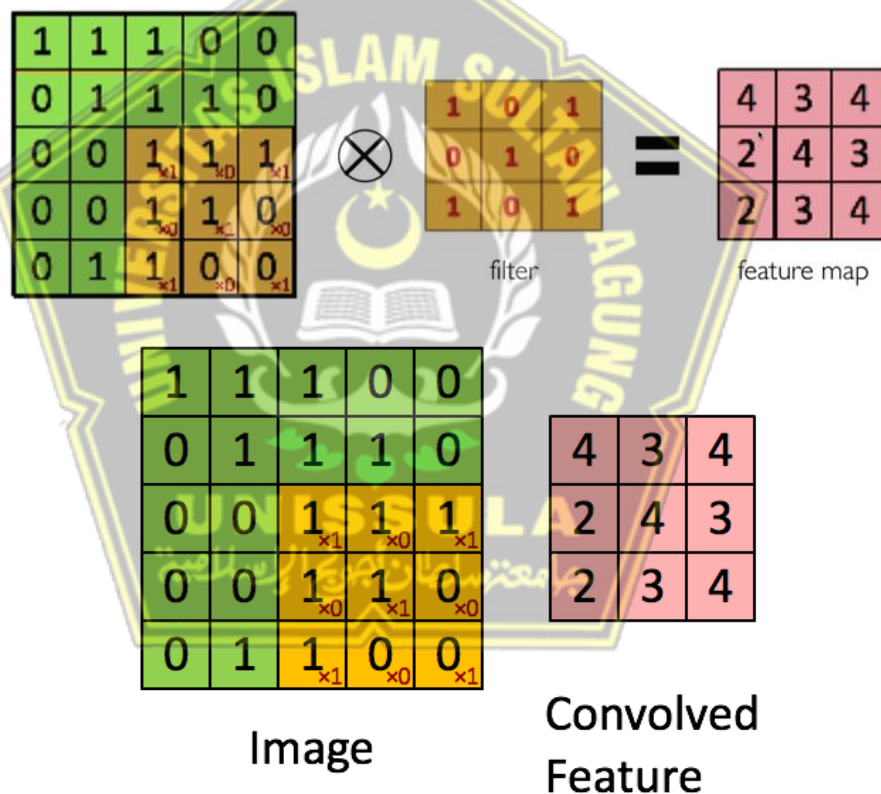


Gambar 2. 3 Lapisan CNN (Rahmawati dkk., 2024)

CNN dibagi menjadi dua komponen, yang pertama ada komponen ekstrasi fitur yang berfungsi untuk mengidentifikasi fitur – fitur sederhana yang ada dalam gambar yang terdiri dari *convolution layer* dan *pooling layer*, kemudian yang kedua ada komponen klasifikasi terdiri dari *fully connected layer* yang berfungsi untuk mengklasifikasi fitur yang telah dihasilkan oleh komponen komponen ekstrasi (Alwanda dkk., 2020).

### 1. Convolution Layer

*Convolution layer* merupakan lapisan utama CNN yang berfungsi untuk melakukan operasi konvolusi pada *output layer* sebelumnya. Tujuan dilakukannya operasi konvolusi pada data citra adalah untuk mengekstraksi fitur dari *input* citra sehingga menghasilkan transformasi linier yang diperoleh dari data yang telah di input sesuai dengan informasi yang telah tersedia pada data. Bobot pada *layer* menentukan *kernel* konvolusi yang digunakan, *kernel* konvolusi dapat dipelajari melalui data yang telah di *input* pada CNN, sehingga CNN dapat memahami struktur data yang telah di input.



Gambar 2. 4 proses *convolution layer* (Dahman, 2021)

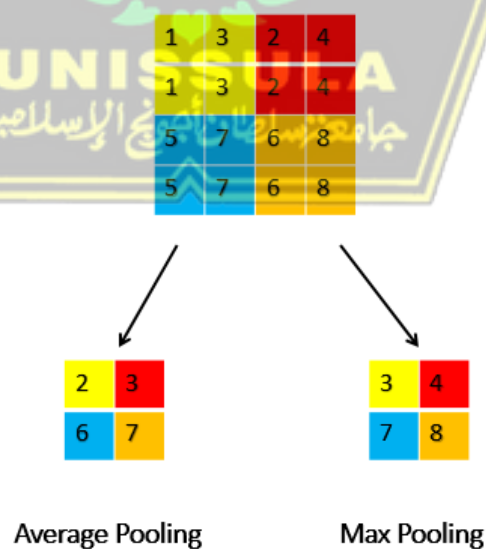
Pada proses konvolusi, dimulai dengan menginputkan gambar yang ingin di proses, kemudian melalui proses filter/kernel merupakan matriks filter yang umumnya berukuran lebih kecil dari pada ukuran gambar input, seperti 3x3, 5x5, atau 7x7. Filter ini akan diplikasikan pada gambar input yang berfungsi untuk mengekstraksi fitur – fitur tertentu, seperti tepi,



tekstur ataupun pola. Selanjutnya melalui proses konvolusi, yang merupakan proses pergeseran filter di seluruh gambar, setiap pergeseran nilai yang ada pada filter dikalikan dengan nilai piksel yang ada pada gambar kemudian hasilnya dijumlahkan sehingga menghasilkan suatu nilai dalam output gambar. Proses ini dilakukan berulang pada seluruh bagian pada gambar, dan dari proses tersebut menghasilkan gambar output yang disebut dengan *feature map* atau *activation map*.

## 2. Pooling Layer

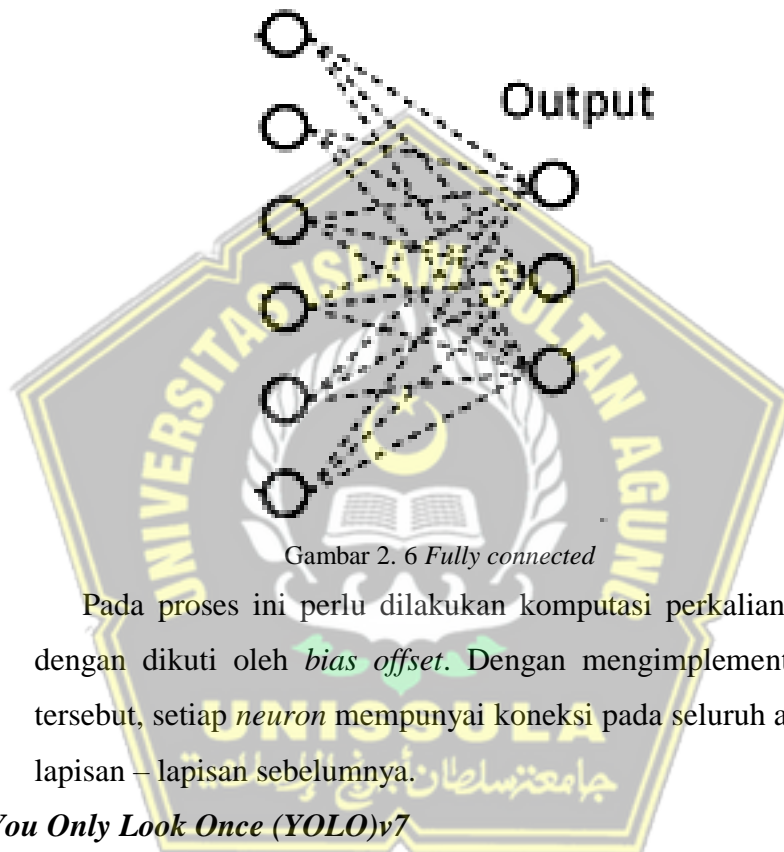
*Pooling layer* berfungsi untuk mengurangi dimensi yang ada pada data citra matriks yang di hasilkan oleh *convolution layer* sehingga dapat mempercepat proses komputasi, membantu mengatasi terjadinya *overfitting*, serta mencegah terjadinya kompleksitas pada model. Tak hanya itu, *pooling layer* juga digunakan untuk mengekstraksi fitur yang terdapat dalam domain sehingga pada proses pelatihan model menjadi lebih efektif. Terdapat dua jenis *pooling layer*, yaitu *max pooling* yang digunakan untuk mengembalikan nilai *max* pada gambar yang dijangkau oleh *kernel* dan *average pooling* bertugas untuk mengembalikan nilai rata-rata dari gambar yang dijangkau oleh *kernel* ( Syaifuddin, 2022).



Gambar 2. 5 konsep *pooling layer*

### 3. *Fully Connected Layer*

*Fully connected layer* digunakan untuk melakukan transformasi pada dimensi data, sehingga data dapat diklasifikasi secara linier. Masukan dari *fully connected layer* merupakan hasil dari dua proses lapisan sebelumnya yang berupa *feature map* karena proses *fully connected layer* dilakukan setelah proses *convolution layer* dan juga *pooling layer*.



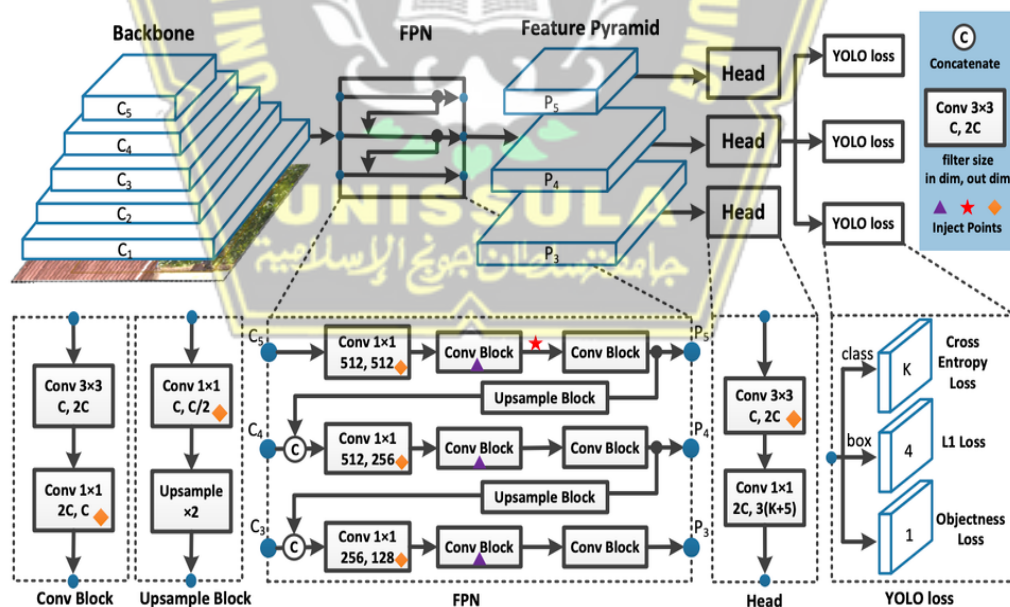
Gambar 2. 6 *Fully connected*

Pada proses ini perlu dilakukan komputasi perkalian pada matriks dengan diikuti oleh *bias offset*. Dengan mengimplementasikan proses tersebut, setiap *neuron* mempunyai koneksi pada seluruh aktivitas dalam lapisan – lapisan sebelumnya.

#### 2.2.4 *You Only Look Once (YOLO)v7*

*You Only Look Once* atau yang biasa disingkat dengan YOLO merupakan algoritma pendeteksi objek secara *real - time* yang dikembangkan oleh Joseph Redmon, Santosh DivVala, Ross Girshich, serta Ali Farhadi pada tahun 2016. Algoritma YOLO menggunakan metode CNN yang berfungsi sebagai pendeteksi objek pada gambar ataupun video yang cepat dan akurat (Andi dkk., 2024).

YOLOv7 merupakan versi terbaru dari algoritma YOLO yang mempunyai karakter kecepatan pada deteksi, kemudian mempunyai presisi yang tinggi, serta dapat dengan mudah melakukan pelatihan data dan juga mudah untuk menerapkannya (Hasan, 2023). Algoritma YOLO merupakan *detector* objek dengan satu tahap. Dalam sistem kerjanya YOLO dapat memprediksi lokasi serta kelas pada objek dengan kotak pembatas (*bounding box*) yang ada pada gambar (Guntara, 2023). YOLOv7 mempunyai peningkatan yang signifikan dari versi sebelumnya, perubahan utama dari YOLO yaitu menggunakan resolusi  $640 \times 640 \text{ pixel}$ , lebih tinggi dibandingkan dengan versi sebelumnya, dengan meningkatkannya hasil dari resolusi ini memungkinkan untuk YOLOv7 mampu melakukan pendeteksian objek dengan lebih akurat. Selanjutnya YOLOv7 juga mempunyai peningkatan pada kecepatan pemrosesan gambar, dengan mencapai 155 *frame per detik* (FPS), dengan itu membuat YOLOv7 menjadi algoritma yang pas untuk sistem pendeteksi objek yang akurat (Shelomo dkk., 2023).

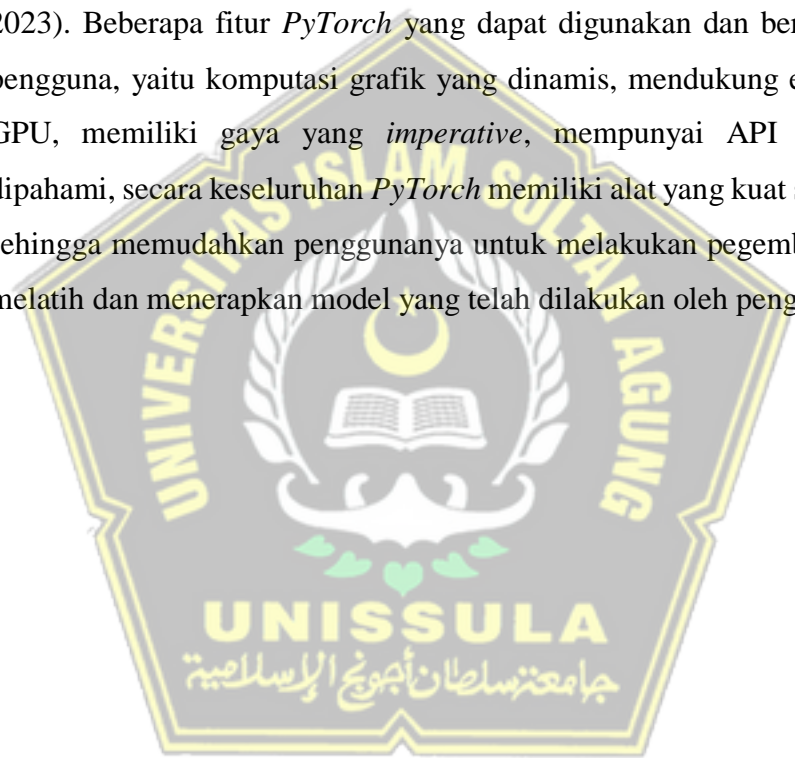


Gambar 2. 7 arsitektur *yolov7* (Mostafa dkk., 2023).

Pada gambar 2.7 menampilkan bagaimana arsitektur yolov7 bekerja, diawali dengan *backbone* yaitu mempersiapkan representasi fitur dasar dari gambar inputan, dengan setiap layer memperoleh informasi yang semakin kompleks. C1, C2, C3, C4, C5 yang merupakan layer – layer konvolusi yang membentuk *backbone* jaringan yang berfungsi untuk mengekstrak fitur dari gambar input. Fungsi dari proses *backbone conv* blok merupakan blok yang berisi operasi konvolusi yang digunakan untuk memproses fitur, sedangkan *upsample block* berfungsi untuk mengembalikan resolusi yang lebih tinggi dari *feature maps*, untuk aturan kernel 3x3 digunakan untuk membantu deteksi fitur serta kernel 1x1 berfungsi untuk mengubah dimensi fitur tanpa ada perubahan resolusi spasial. FPN (*Feature Pyramid Network*) berfungsi sebagai penghubung informasi dari berbagai skala resolusi sebagai peningkatan kemampuan deteksi objek pada berbagai ukuran. P3, P4, P5 merupakan feature map dari berbagai skala yang dihasilkan oleh FPN. *Head* atau bisa disebut *dense prediction* menghasilkan prediksi akhir yang terdiri dari skor kepercayaan prediksi, probabilitas kelas, serta vektor memuat titik tengah dari *bounding box* dengan memprediksi titik tengah, tinggi, serta lebar. *Conv 3x3*, *conv 1x1* merupakan lapisan konvolusi dalam *head* yang digunakan untuk memproses fitur gabungan sebagai prediksi akhir, sehingga menghasilkan output prediksi meliputi objek, *bounding box*, serta *confidence score*. YOLO *loss* yang berfungsi untuk menghitung kesalahan prediksi serta menyesuaikan parameter selama proses pelatihan jaringan, pertama komponen *loss* berfungsi untuk melatih jaringan, *class loss* menggunakan *cross entropy loss* berfungsi untuk klasifikasi objek, *box loss* menggunakan *L1 loss* berfungsi untuk menghitung kesalahan pada objek, *objectness loss* berfungsi memprediksi ada atau tidaknya objek dalam gambar. *Concatenate* berfungsi untuk menggabungkan berbagai fitur dari layer atau skala lainnya sehingga menghasilkan representasi yang lebih beragam. *Inject points* memperkenalkan berbagai informasi dari resolusi sehingga dapat meningkatkan ketepatan prediksi.

### 2.2.5 PyTorch

*PyTorch* merupakan *library* yang biasa digunakan untuk berbagai macam tugas dalam bidang *Machine Learning*, *Deep Learning* dan juga *Natural Language Processing* (NLP) yang dikembangkan oleh Facebook's AI Research lab (FAIR) pada tahun 2016. Pada dasarnya *PyTorch* dikembangkan menggunakan bahasa c++, walaupun begitu *PyTorch* juga mempunyai *frontend* API yang berbasis *python* sehingga pengguna dimudahkan pada saat menggunakan *library* tersebut (Qodryantha dkk., 2023). Beberapa fitur *PyTorch* yang dapat digunakan dan bermanfaat bagi pengguna, yaitu komputasi grafik yang dinamis, mendukung eksekusi pada GPU, memiliki gaya yang *imperative*, mempunyai API yang mudah dipahami, secara keseluruhan *PyTorch* memiliki alat yang kuat serta fleksibel sehingga memudahkan penggunaannya untuk melakukan pengembangan serta melatih dan menerapkan model yang telah dilakukan oleh pengguna.



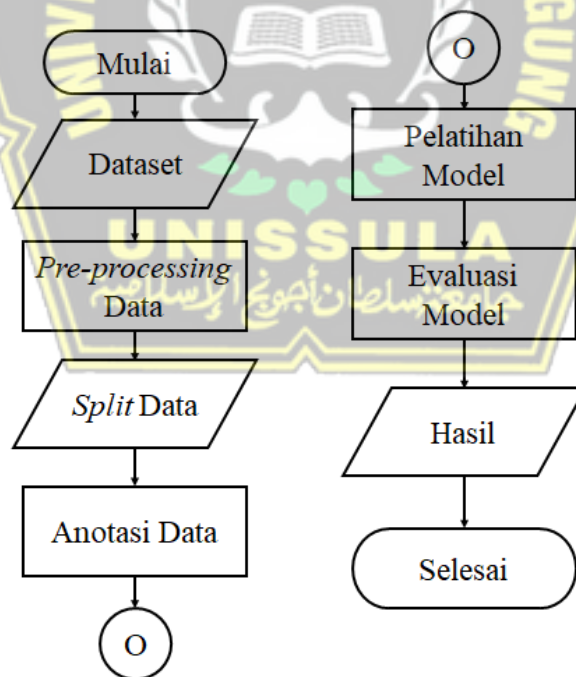
## BAB III METODE PENELITIAN

### 3.1 Studi Literatur

Pada penelitian ini melakukan tinjauan dari berbagai macam jurnal, artikel dan skripsi terdahulu, juga berbagai situs web yang berada di internet. Tujuannya untuk peneliti mempelajari teori tentang *object detection* , *Convolution Neural Networks* (CNN), Algoritma *You Only Look Once* (YOLO)v7.

### 3.2 Metode Penelitian

Pada penelitian ini menggunakan algoritma *You Only Look Once* (YOLO) versi 7 yang nantinya akan melakukan deteksi objek kerusakan pada rumah yang berupa noda bekas air (*damp stain*) pada tembok dan langit – langit serta retakan tembok (*wall crack*). Berikut langkah – langkah yang dilakukan dari penelitian ini :

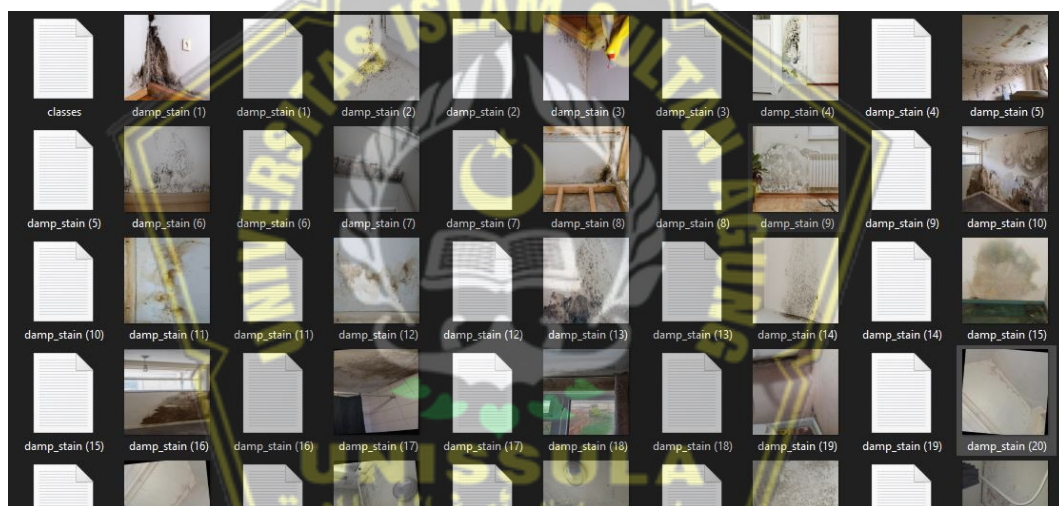


Gambar 3. 1 *Flowchart* alur penelitian

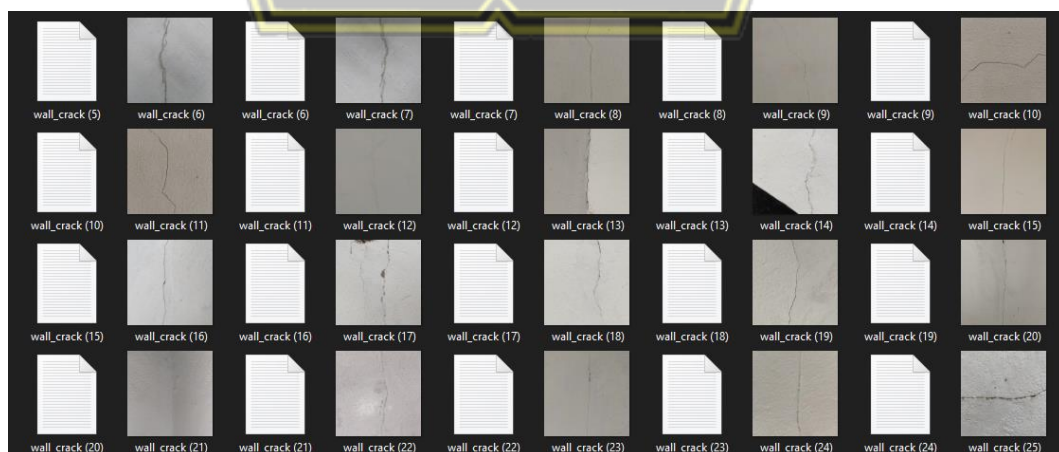
### 3.2.1 Pengumpulan Data

Pada penelitian ini, proses pengumpulan data (*data collecting*) yang berupa gambar diperoleh dari dua sumber utama, yaitu pertama dari gambar didapat dari pemotretan mandiri yang dilakukan oleh peneliti dengan mencari data gambar di beberapa bangunan rumah, kedua gambar diperoleh dari *platform roboflow*, dari kedua sumber tersebut gambar – gambar diseleksi dan kemudian diambil yang dibutuhkan lalu digabungkan menjadi satu setelah itu dirubah menjadi format .zip dan disimpan dalam *google drive* pada link dibawah ini :

[https://drive.google.com/file/d/1viHOgY26cflGq8V\\_kLvZdTCHZuB35Qr/view?usp=sharing](https://drive.google.com/file/d/1viHOgY26cflGq8V_kLvZdTCHZuB35Qr/view?usp=sharing)



Gambar 3. 2 Data gambar dan anotasi *damp stain*



Gambar 3. 3 Data gambar dan anotasi *wall crack*

Pada gambar 3.2 dan 3.3 merupakan kumpulan gambar beserta hasil anotasinya yang nantinya akan dilakukan pengujian model pada penelitian ini, gambar dibagi menjadi dua kelas yaitu kelas *damp\_stain* (noda bekas air) dan *wall\_crack* (retakan dinding).

### 3.2.2 Pre-processing Data

*Pre-processing* merupakan langkah awal untuk melakukan pengolahan data sebelum dilakukan proses pemodelan yang bertujuan untuk meningkatkan kualitas gambar dengan cara menghilangkan *damp stains*serta hal lain yang tidak diinginkan dapat mengurangi kualitas visual, sehingga menghasilkan citra dengan kualitas yang lebih baik. Pada penelitian ini *pre-processing* dilakukan secara manual. Berikut tahapan melakukan *pre-processing* data :

#### a. *Croppig data*

*Croppig* data merupakan proses bagian pada gambar yang kiranya tidak digunakan akan di potong atau di *crop* dengan tujuan untuk memfokuskan area yang ingin dideteksi, dengan dilakukannya *cropping*, salah satu langkah untuk meningkatkan akurasi karena model lebih fokus pada objek yang akan dideteksi.

#### b. *Rezise gambar*

*Rezise* gambar proses perubahan ukuran pada gambar baik dalam tinggi (*height*) maupun lebar (*width*) gambar, sehingga semua gambar menjadi satu ukuran, YOLOv7 menggunakan input gambar dengan ukuran 640 x 640 piksel. Tujuan dilakukannya *resize* gambar adalah menyesuaikan dengan kebutuhan dalam melakukan pelatihan mode, mempertahankan rasio pada gambar agar proporsi gambar tetap konsisten, dapat meningkatkan efesiensi dalam pelatihan model dengan melakukan *resize* gambar dapat mengurangi beban pada komputasi pada saat dilakukannya pelatihan model.



### c. Augmentasi data

Augmentasi data digunakan untuk peningkatan jumlah dan variasi data. Tujuan dilakukan augmentasi data adalah meningkatkan kemampuan model dalam mengenali objek yang terdeteksi, mengurangi *overfitting* karena terdapat berbagai macam pola yang dipelajari oleh model, kemudian agar model dapat mengenali pola yang lebih umum pada data. Pada penelitian ini transformasi augmentasi yang digunakan adalah sebagai berikut :

1. Rotasi (*rotation range*), yaitu dengan cara memutar gambar secara acak sehingga model dapat mempelajari objek meskipun dengan posisi gambar yang berubah
2. *Flipping*, mengubah posisi objek pada objek dengan cara membalik gambar secara *vertical* maupun *horizontal* sehingga mendapatkan gambar dengan variasi baru.
3. Pengubahan warna, kecerahan, saturasi warna, ataupun kontras pada gambar dengan tujuan model dapat mengenali objek dengan pencahayaan yang beragam.
4. *Zoom range* dengan cara memperbesar ataupun memperkecil gambar sehingga model dapat mempelajari objek dengan fitur yang relevan.

### 3.2.3 Split Data

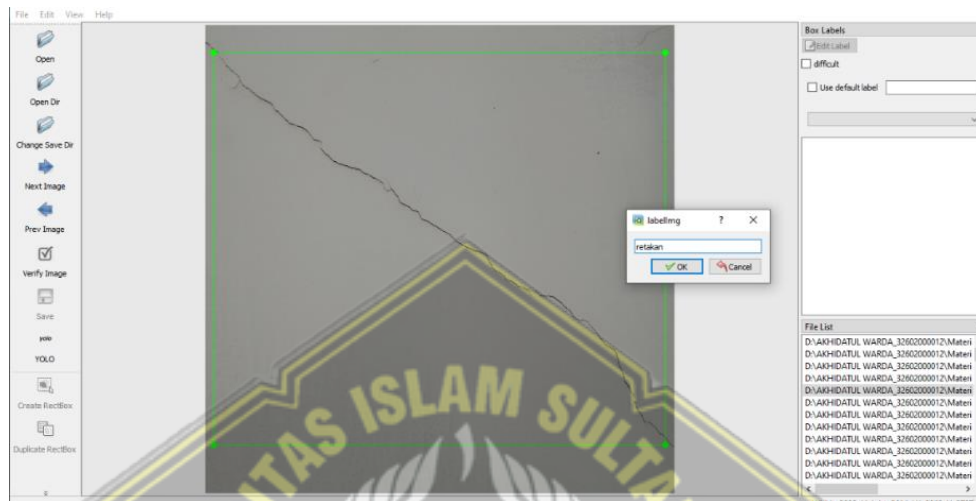
Setelah semua gambar melalui semua proses *pre-processing*, kemudian gambar akan di split data atau dibagi menjadi tiga data, yaitu data *training*, data *validation* dan data *testing*. Dengan pembagian seperti pada tabel 3.1 berikut.

Tabel 3. 1 Tabel *split* data

Nama kelas	<i>Damp stain</i>	<i>Wall crack</i>
Data <i>Training</i>	300	300
Data <i>Validation</i>	90	90
Data <i>Testing</i>	20	20
Total	410	410

### 3.2.4 Anotasi Data

Anotasi data merupakan proses pelabelan menggunakan kotak pembatas (*bounding box*) yang ada pada sekitar gambar. Pada penelitian ini alat yang digunakan untuk melakukan proses anotasi data adalah labelImg.



Gambar 3. 4 Proses Anotasi Data

Pada gambar 3.4 merupakan tampilan labelImg pilih menu open dir untuk membuka direktori berisi gambar yang akan dilakukan anotasi, kemudian klik “*creat rectbox*” untuk mulai membuat *bounding box* pada objek yang ingin dideteksi, setelah membuat *bounding box* jendela pop up muncul dan masukan nama kelas objek, terakhir simpan hasil anotasi, terdapat beberapa format anoatasi gambar seperti pascal, createML serta YOLO, dalam proses ini dipilih format YOLO karena sesuai dengan model YOLOv7. Proses anotasi ini mengubah informasi objek dalam gambar menjadi representasi numerik yang dapat diproses oleh model YOLO, file anotasi YOLO biasa dalam format ‘txt’ yang berisi satu baris angka dengan keterangan *Id class* angka yang menandai sebagai kelas suatu objek, *x* center koordinat *x* pusat *bounding box* yang dinormalisasi dalam rentang [0,1], *y* center koordinat *y* pusat *bounding box* yang dinormalisasi dalam rentang [0,1], kemudian menentu *width* atau lebar *bounding box* yang dinormalisasi dalam rentang [0,1], terakhir *height* atau tinggi *bounding box* yang dinormalisasi dalam rentang [0,1]. Contoh suatu gambar mempunyai ukuran 800x600 piksel, dengan objek berada pada titik koordinat (400,300) dan lebar

240 piksel serta 160 piksel. Dengan format YOLO, koordinat akan di normalisasikan sebagai berikut :

$$x \text{ center} = 400/800 = 0.5$$

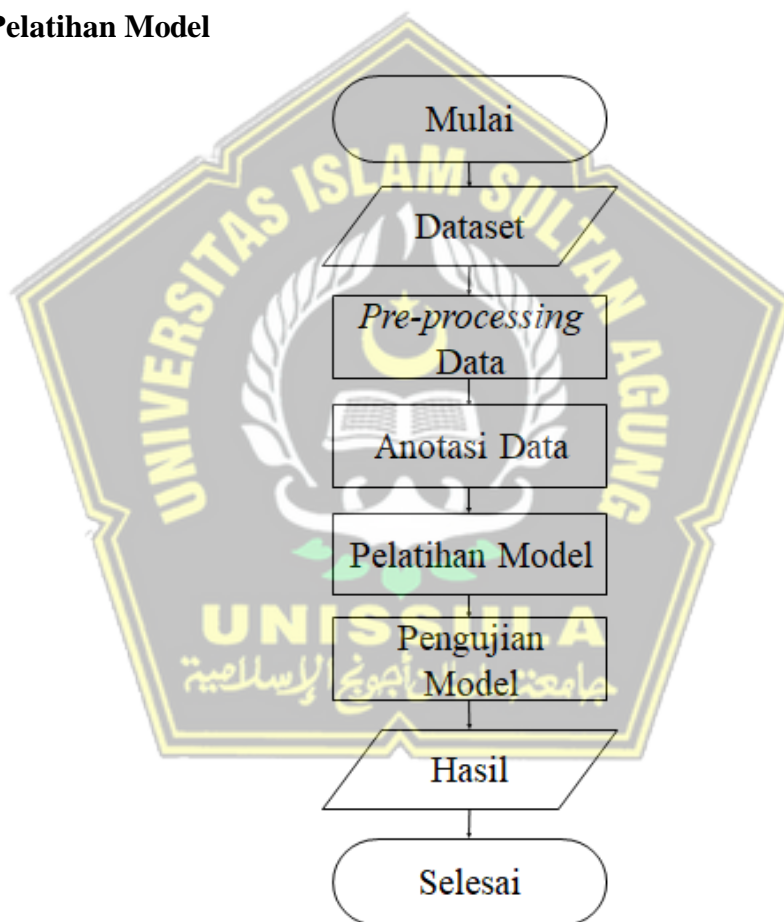
$$y \text{ center} = 300/600 = 0.5$$

$$\text{width} = 240/800 = 0.3$$

$$\text{height} = 160/600 = 0.267$$

Sehingga menghasilkan representasi numerik berupa 0 0.5 0.3 0.267 yang kemudian dapat diproses oleh mesin.

### 3.2.5 Pelatihan Model



Gambar 3. 5 Flowchart pelatihan model

Setelah dilakukan semua proses dari pengumpulan data, kemudian data dilakukan *pre-processing* untuk mempersiapkan data agar dapat diproses pada saat pelatihan model, kemudian dilanjut anotasi data guna memberi label pada data dan sampai di tahap pelatihan model .Pada proses pelatihan model untuk langkah awal yang dilakukan mempersiapkan dataset yang telah

melalui proses *pre-processing* data, kemudian mempersiapkan lingkungan dengan mengunduh program YOLOv7 yang berasal dari GitHub serta mengunduh semua dependensi yang diperlukan yang berisi beberapa library seperti *PyTorch* dan *torchvision*, selanjutnya melakukan inisialisasi model dengan mengunduh pretrained yolo7 untuk melakukan inisiasi weights yang berfungsi untuk mengatur nilai bobot awal sebelum dilakukannya pelatihan, selanjutnya proses mengatur parameter dengan mengatur jumlah workers, batch size, *epoch*, dan mengatur ukuran gambar yang akan diinput dalam proses *training*. Terakhir proses *training* dilakukan menggunakan GPU sehingga menentukan penggunaan GPU, melakukan prediksi atau *forward pass*, menghitung *loss* model, serta melakukan *backward pass* dan *update* bobot. Setelah proses *training* kemudian menghasilkan model yang akan digunakan dalam tahap pengujian model.

### 3.2.6 Evaluasi Model

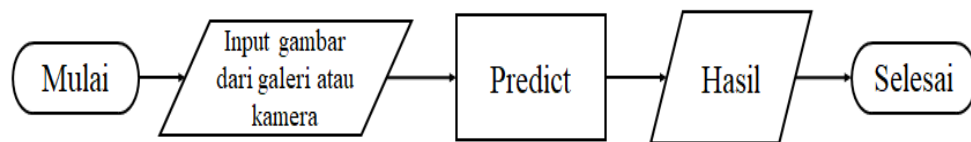
Evaluasi model merupakan proses untuk melihat kombinasi terbaik pada model dalam melakukan prediksi pada dua jenis kerusakan dengan hasil yang tepat. Berikut matrik yang digunakan dalam evaluasi model yolo7 :

1. Matrik Evaluasi
  - a. *Precision* (presisi), digunakan untuk mengukur seberapa banyaknya prediksi positif model yang sesuai dengan objek yang terdeteksi.
  - b. *Recall* (daya ingat), berfungsi untuk mengukur banyaknya objek yang ada pada gambar, sehingga dapat dikatakan model telah berhasil mendeteksi objek.
  - c. *F1 score*, merupakan matrik evaluasi yang digunakan untuk menyeimbangkan antara *precision* dan *recall*.
2. *Mean Average Precision* (mAP), mengukur seberapa baik model dalam mendeteksi objek pada keseluruhan dengan pengukuran presisi rata – rata diberbagai tingkatan pada *recall*, semakin tinggi nilai Map, maka dikatakan semakin baik juga model dalam mendeteksi.
3. *Intersection over union* (IoU), berfungsi untuk mengukur *bounding box* dalam melakukan prediksi sesuai dengan bounding yang sebenarnya.

Semakin tinggi nilai dari IoU maka dapat dikatakan semakin tepat juga model dalam melakukan prediksi.

### 3.3 Deployment model

Berikut langkah – langkah berjalan model pada aplikasi :



Gambar 3. 6 Alur sistem

*Flowchart* diatas menjelaskan alur dari sistem pada aplikasi web, dengan keterangan pertama dengan input gambar dari galeri atau dengan tangkap gambar langsung dengan kamera, kemudian gambar yang sudah diinput akan diprediksi dengan model yang telah dilatih sebelumnya, setelah itu sistem akan menampilkan gambar jenis kerusakan pada bangunan rumah berupa *damp stain* (noda bekas air) atau *damp stain* (retakan dinding) yang telah diprediksi dengan menandai objek jenis kerusakan dengan *bounding box* (kotak pembatas).

## BAB IV HASIL DAN ANALISIS PENELITIAN

### 4.1 Hasil

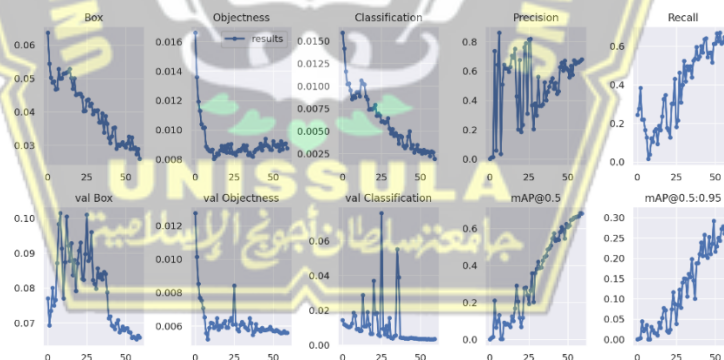
Merujuk pada gambar 3.5 alur pelatihan model dengan melakukan beberapa tahapan untuk mendapatkan hasil model yang akan digunakan dalam deteksi kerusakan bangunan rumah berupa *damp stain* (noda bekas air) dan *wall crack* (retakan dinding).

#### 4.1.1 Hasil Pelatihan Model

Proses pelatihan model menggunakan parameter yang telah ditentukan sebelumnya, *batch size*, *epoch*, dan *resize* gambar. Hasil dari proses pelatihan model dapat dilihat pada grafik evaluasi matriks dibawah ini, yang menunjukkan nilai *loss*, *precision*, *racall*, dan mAP yang berfungsi untuk menilai seberapa baik kinerja model.

##### 1. Matriks Evaluasi

Gambar dibawah ini merupakan matrix hasil dari pelatihan model



Gambar 4. 1 Matrik *training* dan evaluasi

Pada gambar 4.1 menampilkan fungsi *loss* mengalami penurunan pada proses pelatihan, akan tetapi masih ada fluktuasi yang cukup besar pada proses evaluasi yang dapat menimbulkan masalah seperti data yang tidak stabil atau mengakibatkan *overfitting*.

Pada tabel 4.1 diatas menampilkan matrik evaluasi hasil dari pelatihan model, dengan nilai *precision* 0.681, *recall* 0.619, mAP 0.681 dan mAP@ 0.5:0.95 bernilai 0.311.

Tabel 4. 1 Hasil pelatihan model

<b>Matrik Evaluasi</b>	<b>Nilai akhir</b>
<i>Precision</i>	0.681
<i>Recall</i>	0.619
mAP@ 0.5	0.681
mAP@ 0.5:0.95	0.311

Hasil nilai tersebut dapat dilihat dari hasil akhir pelatihan yang dapat dilihat dari gambar 4.2

```

Epoch  gpu_mem  box   obj   cls  total  labels  img_size
59/59   15.3G  0.02591 0.008753 0.001914 0.03658  20      640: 100% 38/38 [00:43:00:00, 1.14s/it]
Class  Images  Labels  P      R      mAP@.5  mAP@.5:.95: 100% 6/6 [00:06:00:00, 1.08s/it]
all    180     210     0.681  0.619  0.681   0.311
damp_stain  180     119     0.583  0.429  0.521   0.235
wall_crack  180     91      0.773  0.809  0.841   0.387
60 epochs completed in 0.947 hours.

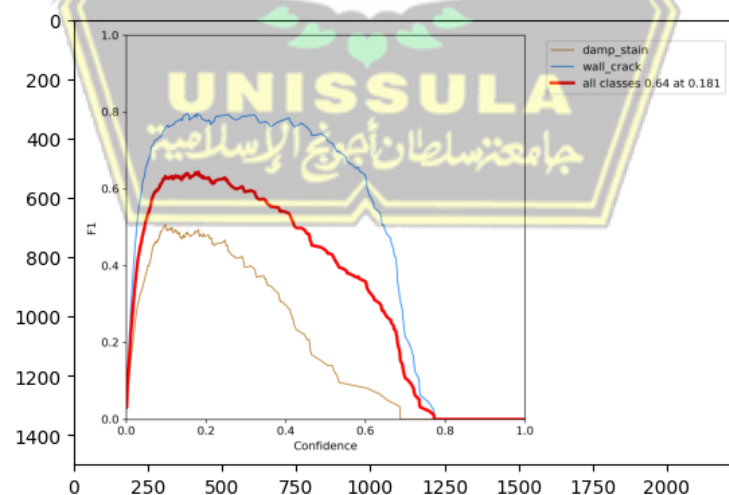
Optimizer stripped from runs/train/yolov7-custom/weights/last.pt, 142.1MB
Optimizer stripped from runs/train/yolov7-custom/weights/best.pt, 142.1MB

```

Gambar 4. 2 Hasil akhir pelatihan

## 2. F1 Score

Gambar dibawah ini adalah *matrix F1 score* hasil dari evaluasi model.



Gambar 4. 3 F1 score

Gambar 4.3 merupakan hasil *F1 score* hasil dari evaluasi model yolov7, yang menampilkan bahwa nilai dari *F1 score* sebesar 0.64 dari semua kelas.

Berikut hasil perhitungan *F1 score* :

$$F1 \text{ score} = 2x \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$F1 \text{ score} = 2x \frac{0.681 \times 0.619}{0.681 + 0.619}$$

$$F1 \text{ score} = 2x \frac{0.421}{1.3}$$

$$F1 \text{ score} = 2x 0.324$$

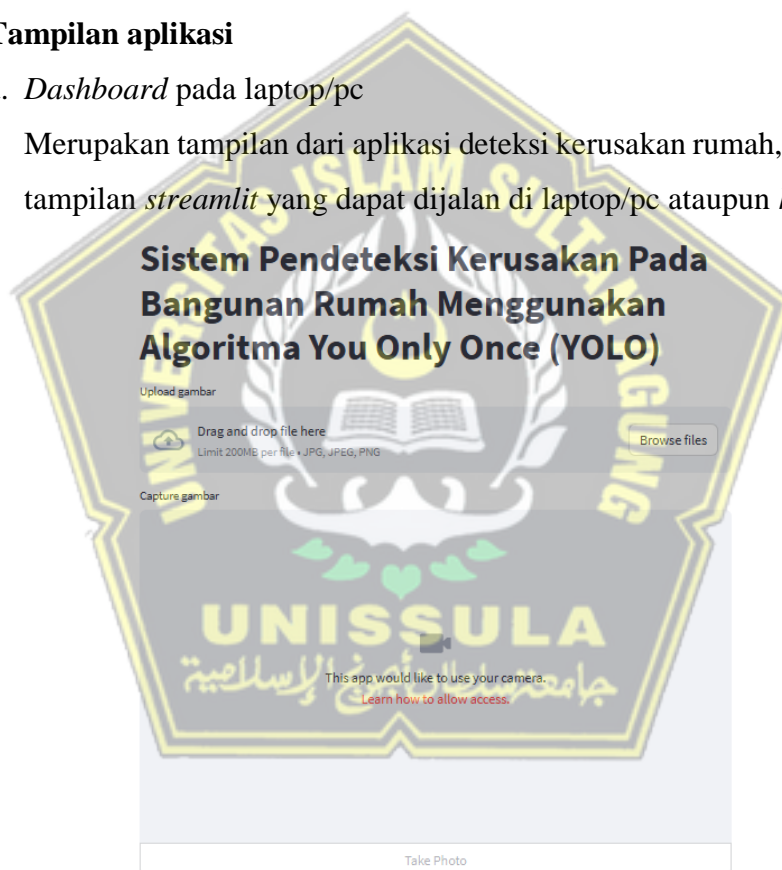
$$F1 \text{ score} = 2x 0.324$$

$$F1 \text{ score} = 0.64$$

#### 4.1.2 Tampilan aplikasi

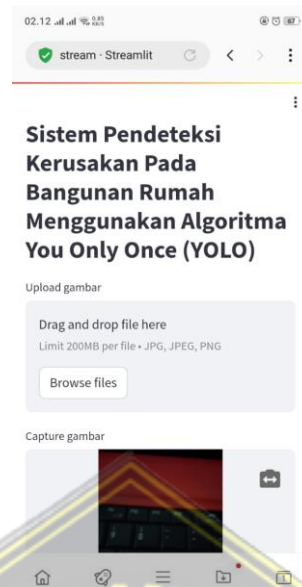
##### a. *Dashboard* pada laptop/pc

Merupakan tampilan dari aplikasi deteksi kerusakan rumah, menggunakan tampilan *streamlit* yang dapat di jalan di laptop/pc ataupun *handpone*.



Gambar 4. 4 Tampilan aplikasi pada PC



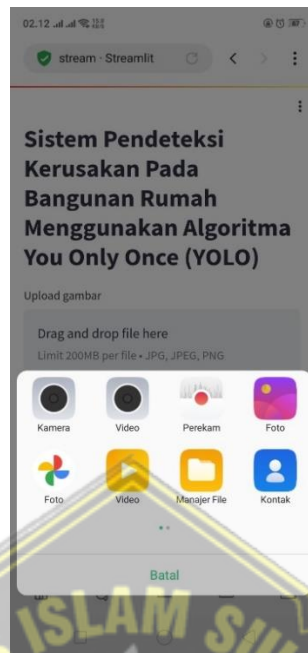


Gambar 4. 5 Tampilan aplikasi pada Ponsel

Pada gambar 4.4 dan 4.5 menampilkan halaman sistem deteksi kerusakan menggunakan. *Bstreamlitrowse files* digunakan untuk *upload* gambar yang ada pada penyimpanan perangkat, *capture* gambar digunakan untuk mengambil gambar secara langsung melalui kamera.



Gambar 4. 6 *upload* gambar



Gambar 4. 7 *upload* gambar pada hp

Gambar 4.6 dan gambar 4.7 menampilkan data gambar yang ada diperangkat yang akan dilakukan pemrosesan data dan kemudian gambar mana yang akan dilakukan prediksi.

b. *Predict*

Setelah gambar *diupload*, langkah selanjutnya gambar akan diprediksi oleh sistem yang kemudian menampilkan hasil prediksi apakah gambar termasuk dalam *damp stain* atau *wall crack*.



Hasil deteksi: wall\_crack 0.52

Gambar 4. 8 Hasil *predict wall crack*

Pada gambar 4.8 menampilkan hasil prediksi dengan jenis *damp stain*, gambar yang diunggah jenis *damp stain* sehingga menunjukkan model dapat mendeteksi data dengan tepat dengan menampilkan akurasi 0.52.



Gambar 4. 9 hasil *predict damp stain*

Pada gambar 4.9 menampilkan hasil prediksi dengan jenis *damp stain*, gambar yang diunggah jenis *damp stain* sehingga menunjukkan model dapat mendeteksi data dengan tepat dengan menampilkan akurasi 0.13.




#### 4.1.3 Pengujian Sistem



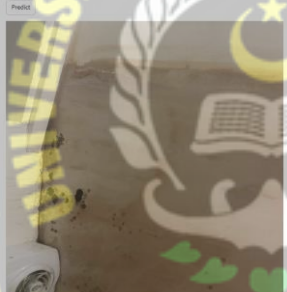
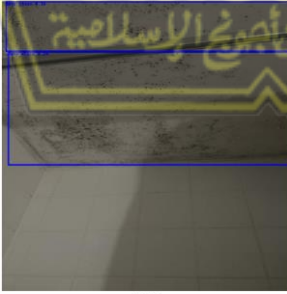
Berikut pengujian yang dilakukan menggunakan data *testing* dan mendapatkan hasil yang beragam :

##### a. Pengujian *damp stain*

Tabel 4.2 merupakan hasil pengujian dari kelas *damp stain* dengan menggunakan 10 gambar uji dan menghasilkan berbagai macam *output*.

Tabel 4. 2 Pengujian *damp stain*

No	Deteksi <i>damp stain</i>	Hasil deteksi	Akurasi
1.	 <p>Hasil deteksi: damp_stain 0.44, damp_stain 0.26</p>	Berhasil	<i>damp stain</i> 0.44 <i>damp stain</i> 0.26
2.	 <p>Hasil deteksi: damp_stain 0.16</p>	Berhasil	<i>damp stain</i> 0.16
3.	 <p>Hasil deteksi: damp_stain 0.24, damp_stain 0.11</p>	Berhasil	<i>damp stain</i> 0.24 <i>damp stain</i> 0.11
4.	 <p>Hasil deteksi: damp_stain 0.22, wall_crack 0.18, damp_stain 0.15</p>	Berhasil tetapi terdapat <i>noise</i>	<i>damp stain</i> 0.22 <i>noise</i> : <i>wall crack</i> 0.18 <i>damp stain</i> 0.15

No	Deteksi <i>damp stain</i>	Hasil deteksi	Akurasi
5.	 <p>Hasil deteksi damp_stain:0.33, damp_stain:0.29, wall_crack:0.22</p>	Bersahil tetapi terdapat <i>noise</i>	<i>damp stain</i> 0.33 <i>wall crack</i> 0.12 <i>noise</i> : <i>damp stain</i> 0.29
6.	 <p>Hasil deteksi wall_crack:0.36, damp_stain:0.24, damp_stain:0.17, damp_stain:0.14</p>	Berhasil tetapi terdapat <i>noise</i>	<i>damp stain</i> 0.36 <i>wall crack</i> 0.36 <i>damp stain</i> 0.17 <i>noise</i> : <i>damp stain</i> 0.14
7.	 <p>Tidak ada objek terdeteksi</p>	Tidak berhasil	-
8.	 <p>Hasil deteksi damp_stain:0.24, damp_stain:0.16</p>	Berhasil	<i>damp stain</i> 0.24 <i>damp stain</i> 0.16

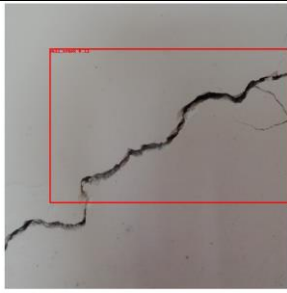


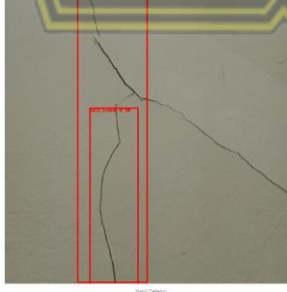
No	Deteksi <i>damp stain</i>	Hasil deteksi	Akurasi
9.		Berhasil	<i>damp stain</i> 0.11 <i>damp stain</i> 0.10
10.		Berhasil tetapi terdapat <i>noise</i>	<i>damp stain</i> 0.15 <i>wall crack</i> 0.13 <i>noise</i> : <i>damp stain</i> 0.19

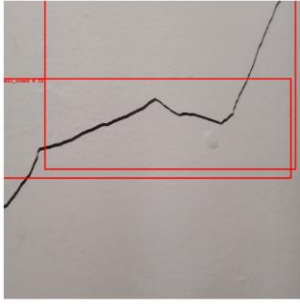

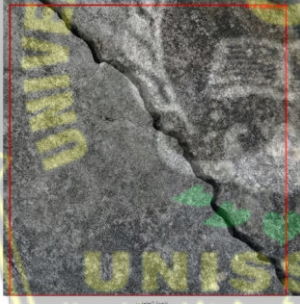
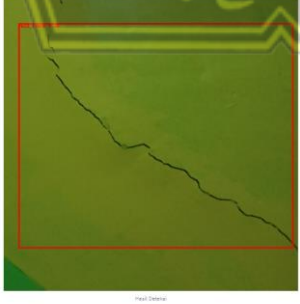
Tabel 4.2 merupakan hasil dari percobaan dari 10 gambar pada kelas *damp stain*, dari pengujian tersebut mendapat hasil sebagian besar model berhasil dalam mendeteksi *damp stain*, akan tetapi ada beberapa gambar percobaan yang terdapat *noise* berupa *damp stain* dan ada juga *noise wall crack* yang mendakan bahwa objek yang terdeteksi tidak semesti ada atau tidak semestinya terdeteksi oleh model, namun dengan begitu model sudah cukup baik dalam mendeteksi kerusakan pada bangunan.

b. Pengujian *Wall crack*



Tabel 4.3 merupakan hasil pengujian dari kelas *wall crack* dengan menggunakan 10 gambar uji dan juga hasil, berhasil atau tidaknya model mendeteksi serta terdapat akurasi.

Tabel 4. 3 Pengujian *wall crack*

No	Deteksi <i>wall crack</i>	Hasil deteksi	Akurasi
1.		Berhasil	<i>Wall crack</i> 0.11
2.		Berhasil	<i>Wall crack</i> 0.37
3.		Berhasil tetapi terdapat <i>noise</i>	<i>wall crack</i> 0.65 <i>noise</i> : <i>damp stain</i> 0.19
4.		Berhasil	<i>wall crack</i> 0.43 <i>wall crack</i> 0.10

No	Deteksi <i>wall crack</i>	Hasil deteksi	Akurasi
5.	 <p>Hasil deteksi: wall_crack 0.12, wall_crack 0.11</p>	Berhasil	<i>wall crack</i> 0.12 <i>wall crack</i> 0.11
6.	 <p>Hasil deteksi: wall_crack 0.41, wall_crack 0.38</p>	Berhasil	<i>wall crack</i> 0.41 <i>wall crack</i> 0.38
7.	 <p>Hasil deteksi: wall_crack 0.33</p>	Berhasil	<i>wall crack</i> 0.33
8.	 <p>Hasil deteksi: wall_crack 0.26</p>	Berhasil	<i>wall crack</i> 0.26



No	Deteksi <i>wall crack</i>	Hasil deteksi	Akurasi
9.		Berhasil	<i>wall crack</i> 0.24 <i>wall crack</i> 0.20
10.		Berhasil tetapi terdapat <i>noise</i>	<i>wall crack</i> 0.12 <i>wall crack</i> 0.11 <i>noise</i> : <i>damp stain</i> 0.10

Tabel 4.3 merupakan hasil dari percobaan dari 10 gambar pada kelas *wall crack*, dari pengujian tersebut mendapat hasil sebagian besar model berhasil dalam mendeksi *wall crack*, akan tetapi ada dua gambar percobaan yang terdapat *noise* dengan mendeteksi adanya *damp stain* yang menandakan bahwa objek yang terdeteksi tidak semestinya ada atau tidak semestinya terdeteksi, untuk selebihnya model mendeteksi dengan baik pada kelas *wall crack*.

Pada kedua percobaan tersebut dapat disimpulkan bahwa model dapat melakukan deteksi objek pada gambar dengan cukup baik, sebagian besar model berhasil dalam melakukan deteksi sesuai dengan kelas atau sesuai dengan jenis kerusakan, akan tetapi masih ada beberapa *noise* pada percobaan gambar yang terdapat dalam *damp stain*, model mendeteksi kelas *wall crack* dan juga mendeteksi *damp stain* model tidak seharusnya mendeteksi objek tersebut tetapi malah terdeteksi oleh model, kemudian untuk percobaan pada kerusakan *wall crack* terdapat dua percobaan yang menghasilkan *noise* yaitu model mendeteksi kerusakan *damp stain* yang seharusnya tidak ada kerusakan tersebut. Dengan hal tersebut dapat diindikasikan adanya *overfitting* pada model.

## 4.2 Analisis

Dari hasil evaluasi model dengan parameter *batch size* 16, *epoch* 60 mendapatkan hasil matrik evaluasi yang menunjukkan model dalam mendeteksi kerusakan pada bangunan rumah. Pada model yang telah dilatih mendapat nilai *precision* sebesar 0.681, nilai *recall* 0.619, dengan nilai – nilai tersebut menunjukkan bahwa model mempunyai nilai dalam mendeteksi kerusakan cukup akurat. Selain itu, pada model ini berhasil mendapatkan nilai mAP mencapai 0.681, dengan ini menandakan bahwa model mempunyai performa yang cukup baik dalam mendeteksi objek pada gambar. Akan tetapi, untuk mAP@0:5:0.95 mendapatkan 0.311 menandakan performa model menurun saat evaluasi pada berbagai *threshold* IoU. Walaupun begitu, model masih bisa mendeteksi objek dengan baik pada kerusakan bangunan rumah dengan dua jenis kerusakan yaitu *damp stain* dan *wall crack*.

Pada pelatihan model ini terdapat fluktuasi pada matrik *validation loss* diidentifikasi model mengalami *overfitting*. Dengan lonjakan *validation loss* menandakan bahwa model tidak selalu dapat mendeteksi dengan baik pada data yang belum pernah dilihat sebelumnya. Selain itu, nilai mAP@0:5:0.95 yang relatif rendah menandakan bahwa model masih kurang efektif dalam mendeteksi objek pada gambar dengan berbagai tingkatan *threshold* IoU, seperti mengidentifikasi objek kecil pada gambar dengan fitur yang hampir sama dengan *background*. Selain itu, fluktuasi yang terjadi pada *validation loss* menunjukkan bahwa kinerja model tidak stabil ketika dihadapkan pada data yang berbeda. Hal ini mungkin menunjukkan bahwa model mengalami kesulitan dalam mengenali pola dari data pelatihan hingga data validasi, sehingga mengurangi akurasi deteksi pada data yang lebih beragam.

Dari kedua percobaan yang telah dilakukan dengan 10 gambar kerusakan *damp stain* dan 10 gambar *wall crack* dapat disimpulkan bahwa model mampu mendeteksi objek pada citra dengan cukup baik terutama dalam mengidentifikasi kelas atau jenis kerusakan yang sesuai. Meskipun sebagian besar deteksi yang dilakukan oleh model tersebut akurat, namun terdapat beberapa *noise* terutama pada citra yang mengandung *damp stain*. Pada beberapa kasus, model salah mendeteksi kelas *wall crack* selain mendeteksi *damp stain*, padahal objek tersebut seharusnya tidak ada. Selain itu pada percobaan deteksi *wall crack* ditemukan dua kasus dimana model salah mendeteksi adanya kerusakan *damp stain* yang sebenarnya tidak ada. Hal ini mengindikasikan terjadinya *overfitting* pada model, dimana model cenderung terlalu sensitif terhadap pola tertentu pada data latih sehingga menghasilkan prediksi yang kurang akurat pada data uji.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari pelatihan model YOLOv7 yang dikembangkan untuk mendeteksi kerusakan bangunan rumah berupa *damp stain* dan *wall crack* memperlihatkan hasil yang cukup baik, akan tetapi masih ada beberapa tantangan yang perlu dihadapi. Berdasarkan hasil evaluasi dengan parameter *batsize* 16 dan *epoch* 60 menunjukkan kinerja model yang cukup baik dalam melakukan deteksi kerusakan pada bangunan rumah berupa *damp stain* dan *wall crack*. Dengan nilai *precision* sebesar 0.681 dan *recall* 0.619, model ini dapat mengidentifikasi kerusakan pada bangunan rumah. Untuk nilai *mean Average Precision* (mAP) mempunyai nilai 0.681 mengidentifikasi bahwa model mempunyai performa yang cukup baik dalam mendeteksi objek.

Akan tetapi performa model menurun pada proses evaluasi di berbagai *threshold* IoU yang dapat dilihat pada nilai mAP@0.5:0.95 mendapatkan nilai sebesar 0.311. Pada model ini juga ada terdapat fluktuasi pada matrik *validation loss* diidentifikasi model mengalami *overfitting*. Meskipun begitu, model ini tetap efektif dalam mendeteksi kerusakan, terutama pada kelas *wall crack*, secara keseluruhan model ini sudah cukup baik dalam melakukan deteksi kerusakan bangunan rumah meskipun masih ada beberapa yang perlu diperbaiki atau diperlukan guna meningkatkan performa model.

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, untuk penelitian selanjutnya peneliti menyarankan :

1. Pada penelitian ini mendapatkan akurasi sebesar 68%, untuk penelitian selanjutnya dapat menggunakan model lain yang dapat menghasilkan akurasi yang lebih baik.
2. Penelitian selanjutnya dapat menambah jumlah data serta meningkatkan variasi dalam data sehingga dapat menghindari terjadinya *overfitting*.
3. Dapat menambahkan fitur secara *real time* menggunakan kamera, sehingga *user* dapat mendeteksi kerusakan bangunan rumah secara langsung.



## DAFTAR PUSTAKA

- Adianto, W. dkk. (2024) "Implementasi Metode Color Blob Detection Pada Objek Daun Sawi," *JCOSIS (Journal Computer Science and Information Syetem*, hal. 20–26.
- Alwanda, M.R. dkk. (2020) "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *Jurnal Algoritme*, 1(1), hal. 45–56. Tersedia pada: <https://doi.org/10.35957/algoritme.v1i1.434>.
- Andi, I. dkk. (2024) "Mask Detection Using the YOLO (You Only Look Once) Method," *Media Informasi Teknologi*, 1(1), hal. 1–12.
- Arianto, B.I. dan Zuliarso, E. (2023) "Implementasi Metode YOLO pada Deteksi Pakaian Keselamatan yang Lengkap di Proyek Kontruksi," *Journal of Multidisciplinary Research and Development*, 6(1), hal. 56–63. Tersedia pada: <https://creativecommons.org/licenses/by/4.0/>.
- Arif, M.F. dkk. (2023) "Deteksi Kendaraan Dengan Metode YOLO," *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, 2(1), hal. 20–27. Tersedia pada: <https://jurnalmahasiswa.com/index.php/aidanspk>.
- Asni, A.B. dkk. (2021) "Penerapan Metode Yolo Object Detection V1 Terhadap Proses Pendeteksian Jenis Kendaraan Di Parkiran," 6(1).
- Chundakus Habsya dkk. (2020) "Pelatihan Perbaikan Kerusakan Rumah Tinggal Di Dusun Klolok, Gawan, Colomadu, Karanganyar," *Jurnal Universitas Sebelas Maret*, (2), hal. 337–348. Tersedia pada: <https://jurnal.uns.ac.id/aktivita/article/download/59602/34838>.
- Dahman, D. (2021) "Convolutional Neural Network LeCun mengenalkan versi awal CNN yaitu LeNet (berasal dari nama LeCun) yang berhasil mengenali karakter tulisan tangan," *medium.com [Preprint]*. Tersedia pada: <https://medium.com/sysinfo/convolutional-neural-network-1c8c1d7e0707>.
- Fajriansyah, D. dkk. (2023) "Automatisasi Deteksi Penyakit Tumbuhan Menggunakan Metode RetinaNet," *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, 1(1), hal. 48–53. Tersedia pada:

<https://jurnalmahasiswa.com/index.php/aidanspk>.

Gelar Guntara, R. (2023) “Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), hal. 55–60. Tersedia pada: <https://doi.org/10.47233/jteksis.v5i1.750>.

Giarida, Z. dkk. (2024) “Perancangan Sistem Pengukur Kecepatan Kendaraan Berbasis Kamera Menggunakan Algoritma YOLO,” 2(3), hal. 655–662.

Guntara, R.G. (2023) “Deteksi Atap Bangunan Berbasis Citra Udara Menggunakan Google Colab dan Algoritma Deep Learning YOLOv7,” 2(1), hal. 9–18.

Hasan, N.F. (2023) “Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7,” 12(D1), hal. 30–35.

Kamil (2023) “Pengolahan Citra Digital Menggunakan Metode Yolo Untuk Mendeteksi Kualitas Dari Biji Kopi Berbasis Android,” *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, 1(1), hal. 120–125. Tersedia pada: <https://jurnalmahasiswa.com/index.php/aidanspk>.

Khatab, U. dkk. (2022) “Klasifikasi Tanah Di Lima Kecamatan Kota Payakumbuh Dengan Sistem Aashto,” *Jurnal Rekayasa*, 12(2), hal. 164–174. Tersedia pada: <https://doi.org/10.37037/jrftsp.v12i2.146>.

Maharani, A.A. dkk. (2023) “ISSN: 2459-9727 Prosiding Seminar Nasional Teknik Sipil 2023 Teknik Sipil Fakultas Teknik Universitas Muhammadiyah Surakarta.”

Mostafa, M. dkk. (2023) “Automatic Vehicle Classification and Speed Tracking,” *Proceedings of the 2nd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2023*, (May), hal. 972–977. Tersedia pada: <https://doi.org/10.1109/ICAAIC56838.2023.10140935>.

Muhammad Syaifuddin, N.U.R. (2022) “Sistem Deteksi Penyakit Mata Pterigium Berbasis Android Menggunakan Metode Convolutional Neural Network.” Universitas Islam Sultan Agung Semarang.

Nafis Alfarizi, D. dkk. (2023) “Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis,” *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, 1(1), hal. 54–63. Tersedia pada:

<https://jurnalmahasiswa.com/index.php/aidanspk>.

- Nayottama, A. dan dkk. (2023) "Sistem Keamanan pada Peternakan Sapi," 12(2).
- Nihayatul Husna, I. dkk. (2022) "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," Seminar Nasional Forte Regional, 7, hal. 1–6.
- Passa, R.S. dkk. (2023) "Deteksi Tumor Otak Pada Magnetic Resonance Imaging Menggunakan Yolov7," Jurnal Ilmiah Matrik, 25(2), Hal. 116–121. Tersedia Pada: <https://doi.org/10.33557/JurnalMatrik.V25i2.2404>.
- Qodryantha, A. dkk. (2023) "Pembangunan Sistem Pengawasan Cerdas Dengan Visualisasi 3D Development Of Smart Surveillance System With 3D Visualization," 9(1), hal. 1–16. Tersedia pada: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/19508?btwaf=91705735>.
- Rahmawati, A.A. dkk. (2024) "Classification of Javanese Nglegena Script Using Complexvalued Neural Network," JIKO (Jurnal Informatika dan Komputer), 7(1), hal. 30–35. Tersedia pada: <https://doi.org/10.33387/jiko.v7i1.7808>.
- Sarosa, M. dan Muna, N. (2021) "Implementasi Algoritma You Only Look Once ( Yolo ) Untuk Implementation of You Only Look Once ( Yolo ) Algorithm for," Jurnal Teknologi Informasi dan Ilmu Komputer, 8(4), hal. 787–792. Tersedia pada: <https://doi.org/10.25126/jtiik.202184407>.
- Shelomo, J. dkk. (2023) "Perbandingan Keandalan dan Efisiensi antara YOLOv7 dan YOLOv8 untuk Deteksi Jenis Kendaraan Bermotor di Jalan Tol," hal. 1–11.
- Sistem, R. (2021) "Jurnal resti," 5(10), hal. 624–630.
- Tiyar, R.I. dan Fudholi, D.H. (2021) "Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek," Petir, 14(2), hal. 258–268. Tersedia pada: <https://doi.org/10.33322/petir.v14i2.1350>.
- Yasen, N.M. dkk. (2023) "Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode Deep Learning," Elektron : Jurnal Ilmiah, 15, hal. 63–71. Tersedia pada: <https://doi.org/10.30630/eji.0.0.397>.