



**OTOMATISASI PEMBACAAN METER AIR SECARA *COMPUTER VISION*
UNTUK PEMANTAUAN DEBIT AIR MENGGUNAKAN YOLO V5**

LAPORAN TUGAS AKHIR



**Program Studi
S1 Teknik Komputer**

**UNIVERSITAS
Dinamika**

Oleh:

YUDA PRASETYO

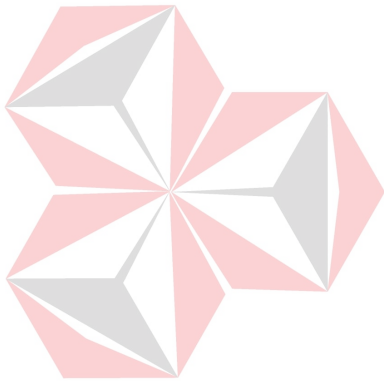
19410200025

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2025**

**OTOMATISASI PEMBACAAN METER AIR SECARA *COMPUTER VISION*
UNTUK PEMANTAUAN DEBIT AIR MENGGUNAKAN YOLO V5**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik Komputer**



Disusun Oleh:

Nama : YUDA PRASETYO

NIM : 19410200025

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2025

TUGAS AKHIR

OTOMATISASI PEMBACAAN METER AIR SECARA *COMPUTER VISION* UNTUK PEMANTAUAN DEBIT AIR MENGGUNAKAN YOLO V5

Dipersiapkan dan disusun oleh

Yuda Prasetyo

NIM: 19410200025

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada: Rabu, 20 Agustus 2025

Susunan Dewan Pembahas

Pembimbing:

I. Heri Pratikno, M.T.

NIDN.0716117302

II. Pauladie Susanto, S.Kom., M.T.

NIDN.0729047501



Digitally signed by Heri Pratikno, M.T.
DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=S1 Teknik
Komputer, email=heri@dinamika.ac.id,
c=ID
Date: 2025.09.02 14:00:28 +0700
Adobe Acrobat version: 11.0.23



cn=Pauladie Susanto, o=Universitas
Dinamika, ou=PS S1 Teknik
Komputer,
email=pauladie@dinamika.ac.id, c=ID
2025.09.03 21:13:26 +0700

Pembahas:

I. Weny Indah Kusumawati, S.Kom., M.MT.

NIDN.0721047201



cn=Weny Indah Kusumawati,
o=Undika, ou=Prodi S1 TK - FTI,
email=weny@dinamika.ac.id,
c=ID
2025.09.04 12:12:11 +0700

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana



Digitally signed by Julianto
Date: 2025.09.10 09:47:52
+07'00'

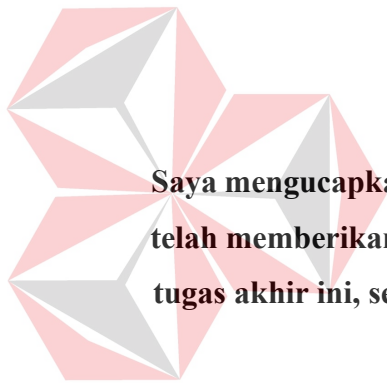
Julianto Lemantara, S.Kom., M.Eng.

NIDN.0722108601

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA



UNIVERSITAS
*“Setiap detik adalah kesempatan
untuk berubah menjadi lebih baik”*
~ Robert Downey Jr. ~



Saya mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bimbingan, dan motivasi selama proses penyusunan tugas akhir ini, sehingga akhirnya saya dapat menyelesaikannya dengan baik dan lancar.

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Yuda Prasetyo
NIM : 19410200025
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **OTOMATISASI PEMBACAAN METER AIR SECARA
COMPUTER VISION UNTUK PEMANTAUAN DEBIT
AIR MENGGUNAKAN YOLO VS**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya 11 Juli 2025



NIM: 19.41020.0025

ABSTRAK

Pembacaan meter air secara manual sering kali menimbulkan masalah seperti kesalahan pencatatan, ketidak efisienan waktu, dan kurangnya transparansi data. Penelitian ini bertujuan untuk mengembangkan sistem otomatisasi pembacaan angka pada meter air menggunakan teknologi Computer Vision dan algoritma deteksi objek YOLOv5. Dataset yang terdiri dari 2.668 gambar dianotasi menggunakan platform Roboflow dan dibagi ke dalam data latih, validasi, dan uji. Model YOLOv5 dilatih untuk mengenali angka 0–9 serta area digit secara akurat. Sistem diintegrasikan dengan protokol komunikasi MQTT, sehingga hasil pembacaan angka dapat dikirim ke aplikasi mobile IoT MQTT Panel secara real-time. Hasil pengujian menunjukkan bahwa model deteksi mencapai nilai precision sebesar 0.87, recall 0.84, dan mAP50 0.81. Sistem deteksi digit berhasil membaca angka dengan akurasi rata-rata 94%, dan akurasi pengiriman data melalui MQTT mencapai 90%. Secara keseluruhan, sistem terbukti mampu bekerja secara real-time, efisien, dan andal untuk pemantauan penggunaan air otomatis.

Kata Kunci: YOLO, visi komputer, *Water Meter*, *realtime*, *prototype*.



UNIVERSITAS
Dinamika

KATA PENGANTAR

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan Matakuliah Tugas Akhir pada Program Studi S1 Teknik Komputer Universitas Dinamika.

Dalam proses penulisan Laporan Tugas Akhir ini, penulis menerima banyak bantuan baik secara moral maupun materi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih dan penghargaan kepada:

1. Allah SWT, karena dengan rahmat dan berkat-Nya yang berlimpah penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang Tua yang selalu memberikan support selalu dalam berbagai hal sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini.
3. Bapak Julianto Lemantara, S.Kom., M.Eng., selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
4. Bapak Pauladie Susanto, S.kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer dan juga sebagai Dosen Pembimbing 2 yang telah membantu mengajarkan dan membagi ilmunya kepada penulis dalam menyelesaikan Tugas Akhir.
5. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku Dosen Pembahas yang telah memberikan banyak masukan penting sehingga Tugas Akhir ini dapat terselesaikan.
6. Bapak Heri Pratikno, M.T., selaku Dosen Pembimbing 1 yang selalu meluangkan waktunya ditengah kesibukannya untuk membimbing penulis dalam menyelesaikan Tugas Akhir.
7. Seluruh teman – teman S1 Teknik Komputer terkhusus angkatan 2019 yang telah membantu penulis berupa dukungan, semangat, dan masih banyak lagi.

Penulis menyadari bahwa penulisan laporan ini terdapat kekurangan. Oleh karena itu kritik dan saran sangat diharapkan oleh penulis untuk membantu menyempurnakan laporan ini.

Surabaya, 20 Agustus 2025

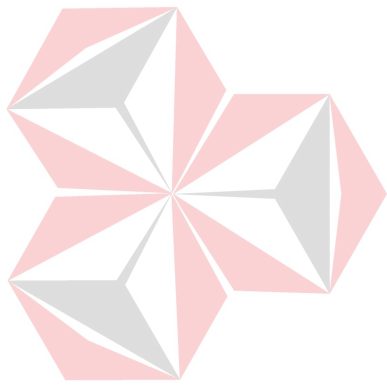
Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
BAB II LANDASAN TEORI.....	4
2.1 <i>Water Meter</i>	4
2.2 <i>Computer Vision</i>	5
2.3 Roboflow sebagai <i>Platform Dataset</i>	6
2.4 YOLO.....	7
2.4.1 Konsep Dasar YOLO.....	7
2.4.2 Arsitektur dan Cara Kerja YOLO	8
2.4.3 Alur <i>Training</i> dan Inferensi YOLOv5	8
2.4.4 Keunggulan YOLO dalam Implementasi Otomatisasi	9
2.5 Python	9
2.6 OpenCV	10
2.7 <i>Internet of Things</i>	11
2.8 MQTT	11
2.9 IoT MQTT Panel.....	13
BAB III METODOLOGI PENELITIAN	14
3.1 Studi Literatur	14
3.2 Persiapan Alat dan Bahan	14
3.2.1 Alat	15

3.2.2	Bahan	15
3.3	Pengolahan <i>Dataset</i>	16
3.3.1	Pengumpulan <i>Dataset</i>	16
3.3.2	Anotasi <i>Dataset</i>	17
3.3.3	Download <i>Dataset</i>	19
3.4	<i>Training Model</i>	20
3.5	<i>Flowchart</i>	24
3.5.1	<i>Flowchart</i> Tahapan Kerja	24
3.5.2	<i>Flowchart</i> Cara Kerja Alat.....	25
3.6	Blok Diagram	27
3.6.1	Blok Diagram Perangkat Lunak	27
3.6.2	Blok Diagram Perangkat Keras	28
3.7	Aplikasi IoT MQTT Panel	29
3.8	Implementasi Sistem	30
3.9	Pengujian Sistem.....	31
3.10	Analisis Data dan Evaluasi.....	31
BAB IV HASIL PENELITIAN		32
4.1	Validasi Hasil <i>Training</i> Data	32
4.1.1	Peralatan yang Dibutuhkan.....	32
4.1.2	Prosedur Validasi Training Data	32
4.1.3	Hasil Validasi <i>Training</i> Data.....	33
4.2	Pengujian Deteksi Digit Angka.....	37
4.2.1	Peralatan yang Dibutuhkan.....	37
4.2.2	Prosedur Deteksi Digit Angka	37
4.2.3	Hasil Deteksi Digit Angka.....	37
4.2.4	Analisa <i>Error</i>	42
4.3	Pengujian Pengiriman Data MQTT	43
4.3.1	Peralatan yang Dibutuhkan.....	43
4.3.2	Prosedur Pengiriman Data MQTT.....	44
4.3.3	Hasil Pengiriman Data MQTT.....	44
4.4	Pengujian Keseluruhan Sistem.....	46
4.4.1	Peralatan yang Dibutuhkan.....	46
4.4.2	Prosedur Pengujian Keseluruhan Sistem	46

4.4.3 Hasil Pengujian Keseluruhan Sistem.....	47
BAB V PENUTUP	49
5.1 Kesimpulan	49
5.2 Saran.....	50
DAFTAR PUSTAKA	51
LAMPIRAN	52



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 4.1 Hasil <i>training learning rate</i> tetap	33
Tabel 4.2 Hasil <i>training batch size</i> tetap	34
Tabel 4.3 Hasil <i>training epoch</i> tetap	35
Tabel 4.4 Hasil pengujian deteksi angka	40
Tabel 4.5 Rangkuman hasil pengujian deteksi angka.....	41
Tabel 4.6 Pengujian data MQTT	45
Tabel 4.7 Hasil pengujian keseluruhan sistem	47

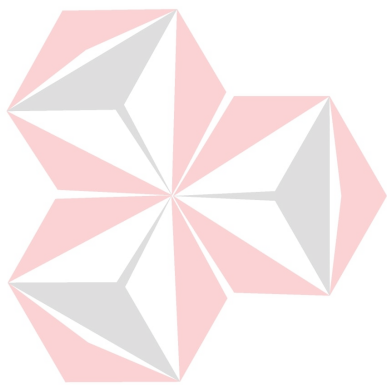


UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Water meter</i>	4
Gambar 2.2 <i>Computer Vision</i>	5
Gambar 2.3 Roboflow	6
Gambar 2.4 Konsep YOLO	7
Gambar 2.5 Bahasa pemrograman Python	9
Gambar 2.6 OpenCV Python.....	10
Gambar 2.7 IoT	11
Gambar 2.8 Protokol MQTT	12
Gambar 2.9 IoT MQTT Panel	13
Gambar 3.1 Daftar <i>dataset</i>	16
Gambar 3.2 Proses anotasi <i>dataset</i>	17
Gambar 3.3 Kelas <i>object detection</i>	18
Gambar 3.4 Proses <i>download dataset</i> YOLOv5	19
Gambar 3.5 Jumlah <i>dataset split</i>	20
Gambar 3.6 Struktur folder <i>dataset</i> YOLOv5	21
Gambar 3.7 Isi data <i>data.yaml</i>	22
Gambar 3.8 Konfigurasi parameter	22
Gambar 3.9 Proses <i>training</i>	23
Gambar 3.10 <i>Flowchart</i> tahapan kerja	24
Gambar 3.11 <i>Flowchart</i> cara kerja alat	25
Gambar 3.12 Blok diagram perangkat lunak	27
Gambar 3.13 Blok diagram perangkat keras	28
Gambar 3.14 Konfigurasi MQTT	29
Gambar 3.15 Tampilan IoT MQTT Panel	30
Gambar 4.1 Konfigurasi utama deteksi angka	37
Gambar 4.2 Pengujian <i>water meter</i> 1	39
Gambar 4.3 Pengujian <i>water meter</i>	39
Gambar 4.4 Analisa <i>error</i> digit terakhir	43
Gambar 4.5 Konfigurasi MQTT	44

Gambar 4.6 Hasil pengujian <i>monitoring water meter</i>	44
--	----



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Program <i>Training Dataset</i>	52
Lampiran 2 Program Uji Coba Gambar	53
Lampiran 3 Program <i>Webcam</i>	54
Lampiran 4 Program Seluruh Sistem	56
Lampiran 5 Hasil Pengujian	59
Lampiran 6 Form Bimbingan Tugas Akhir	61
Lampiran 7 Bukti Originalitas Tugas Akhir	62
Lampiran 8 Biodata Penulis	64



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemantauan konsumsi air menjadi salah satu aspek penting dalam pengelolaan sumber daya, baik untuk kebutuhan rumah tangga maupun industri. Sistem manual yang saat ini banyak digunakan dalam pembacaan meter air memiliki sejumlah kelemahan, seperti rentan terhadap kesalahan pencatatan, memerlukan waktu yang cukup lama, dan kurang efisien jika mencakup wilayah luas atau daerah terpencil. Masalah ini semakin signifikan mengingat kebutuhan akan efisiensi dan akurasi yang terus meningkat di era digital.

Dalam konteks pengelolaan utilitas, kesalahan dalam membaca meter air dapat berujung pada perbedaan tagihan yang signifikan, baik kelebihan maupun kekurangan, yang akhirnya menimbulkan ketidakpuasan pelanggan. Selain itu, pengolahan data yang manual juga menyulitkan integrasi informasi secara real-time untuk kebutuhan analitik atau perencanaan.

Penggunaan teknologi modern, seperti *Computer Vision* dapat menjadi solusi inovatif untuk mengatasi tantangan tersebut. Dengan memanfaatkan *Computer Vision*, angka pada meter air dapat dikenali secara otomatis melalui citra digital yang diolah menggunakan algoritma deteksi objek.

Implementasi solusi ini tidak hanya menawarkan efisiensi dalam pengelolaan sumber daya, tetapi juga meningkatkan transparansi dan keandalan data. Dalam jangka panjang, hal ini dapat mendukung pengambilan keputusan yang lebih baik, baik oleh perusahaan utilitas maupun pelanggan individu.

Berdasarkan penelitian sebelumnya oleh Risna dan Harriski Arie Pradana (2024), telah dirancang alat monitoring penggunaan air PDAM berbasis mikrokontroler Arduino Uno. Sistem tersebut menggunakan *water flow sensor*, RTC, Arduino Uno, dan LCD, di mana data dari *water flow sensor* dan RTC diproses oleh Arduino Uno untuk ditampilkan pada LCD.

Pada Tugas Akhir ini, penulis akan mengembangkan sistem berbasis *Artificial Intelligence* (AI) untuk otomatisasi pembacaan angka pada meter air menggunakan

teknologi *Computer Vision*. Sistem ini dirancang untuk mengenali angka pada meter air secara otomatis melalui citra digital.

1.2 Rumusan Masalah

Berdasarkan Rumusan Masalah yang menjadi fokus dalam penelitian ini adalah:

1. Bagaimana merancang dan mengembangkan sistem berbasis *Computer Vision* yang mampu membaca angka pada meter air secara otomatis?
2. Bagaimana mengoptimalkan akurasi pembacaan angka pada meter air?

1.3 Batasan Masalah

Dalam penelitian ini, batasan masalah ditentukan untuk memperjelas ruang lingkup proyek dan fokus pada implementasi teknis sebagai berikut:

1. Untuk melatih model deteksi objek, penelitian ini memanfaatkan *dataset* citra meter air yang dikumpulkan secara mandiri selain menggunakan *dataset* yang sudah tersedia dan relevan. Kombinasi kedua *dataset* ini diharapkan dapat meningkatkan akurasi model.
2. Deteksi angka hanya mencakup angka desimal (0–9) pada meter air.
3. Pengujian sistem dibatasi pada pembacaan angka meter air dengan berbagai kondisi pencahayaan, tetapi tidak mencakup variasi meter air non-digital atau berbasis jarum.
4. Pengujian dilakukan dengan pencahayaan yang merata atau ideal.
5. Posisi dan jarak antara kamera dan angka meter air dalam kondisi yang tetap atau statis.

1.4 Tujuan

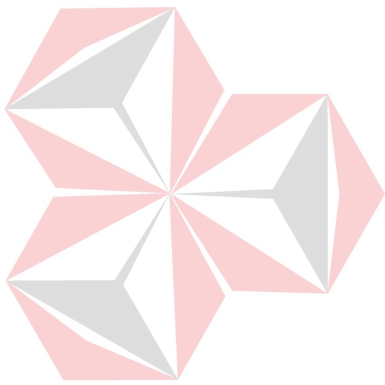
Berikut adalah tujuan dari Tugas Akhir ini:

1. Membangun sistem otomatis berbasis *Computer Vision* yang mampu mendeteksi dan membaca angka pada meter air.
2. Meningkatkan efisiensi dan akurasi dalam proses pembacaan angka.

1.5 Manfaat

Adapun manfaat yang diharapkan dari pelaksanaan tugas akhir ini adalah sebagai berikut:

1. Bagi Peneliti: Menambah wawasan dan pengalaman dalam pengembangan teknologi berbasis *Computer Vision*.
2. Bagi Industri: Memudahkan perusahaan utilitas dalam memantau penggunaan air secara efisien dan akurat melalui pengelolaan data *real-time*.
3. Bagi Masyarakat: Memberikan transparansi dan akurasi yang lebih baik dalam tagihan air serta meminimalkan potensi kesalahan pencatatan manual.
4. Secara Akademik: Berkontribusi pada pengembangan studi kasus penerapan teknologi AI dalam pemecahan masalah praktis di bidang utilitas.



UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 *Water Meter*

Water meter atau meter air adalah perangkat yang digunakan untuk mengukur volume konsumsi air oleh pelanggan, baik untuk kebutuhan rumah tangga maupun industri. Umumnya, proses pembacaan angka pada meter dilakukan secara manual oleh petugas lapangan, yang kemudian dicatat dan dilaporkan ke pihak pengelola air. Proses ini sering kali memakan waktu dan rawan terjadi kesalahan pencatatan.



Gambar 2.1 *Water meter*
(Sumber: <https://onda.id/>)

Keterbatasan sistem manual mendorong pengembangan teknologi pembacaan otomatis, agar data konsumsi air dapat dikumpulkan secara *real-time* dan akurat. Dengan otomatisasi, risiko *human error* dapat ditekan, dan efisiensi proses pencatatan serta pelaporan penggunaan air menjadi lebih baik, mendukung manajemen sumber daya air yang lebih transparan dan efektif. Oleh karena itu, pengembangan sistem pembacaan otomatis sangat penting untuk meningkatkan akurasi dan efisiensi dalam pengelolaan sumber daya air. (Yulianto, 2018)

2.2 Computer Vision

Computer Vision adalah proses pembelajaran dan menganalisa gambar ataupun video untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Dengan bahasa lebih mudahnya, *computer vision* mencoba meniru cara kerja visualisasi manusia.



Gambar 2.2 *Computer Vision*
(Sumber: <https://viso.ai>)

Computer Vision diterapkan untuk mendeteksi angka pada meter air melalui citra digital. Teknologi ini melibatkan berbagai tahapan, seperti akuisisi gambar, segmentasi objek, ekstraksi fitur, dan pengenalan pola. Salah satu pendekatan yang populer dalam *Computer Vision* adalah penggunaan model deteksi objek berbasis 5 algoritma deep learning seperti YOLO (*You Only Look Once*). Algoritma ini memungkinkan pendeteksian angka pada gambar secara cepat dan akurat dengan memanfaatkan *bounding box* yang memprediksi posisi objek pada citra.

Keunggulan utama *Computer Vision* adalah kemampuannya untuk bekerja dengan volume data yang besar dan memproses informasi secara otomatis, sehingga sangat cocok untuk aplikasi yang memerlukan efisiensi tinggi, seperti pembacaan meter air pada skala besar. (Szeliski, 2022)

2.3 Roboflow sebagai *Platform Dataset*

Roboflow adalah *platform* visi komputer yang memungkinkan pengguna membangun model visi komputer lebih cepat dan lebih akurat melalui penyediaan teknik pengumpulan data, praproses, dan pelatihan model yang lebih baik. Roboflow memungkinkan pengguna mengunggah kumpulan data khusus, menggambar anotasi, mengubah orientasi gambar, mengubah ukuran gambar, mengubah kontras gambar, dan melakukan penambahan data. Roboflow juga dapat digunakan untuk melatih model.



Gambar 2.3 Roboflow
(Sumber: <https://roboflow.com/>)

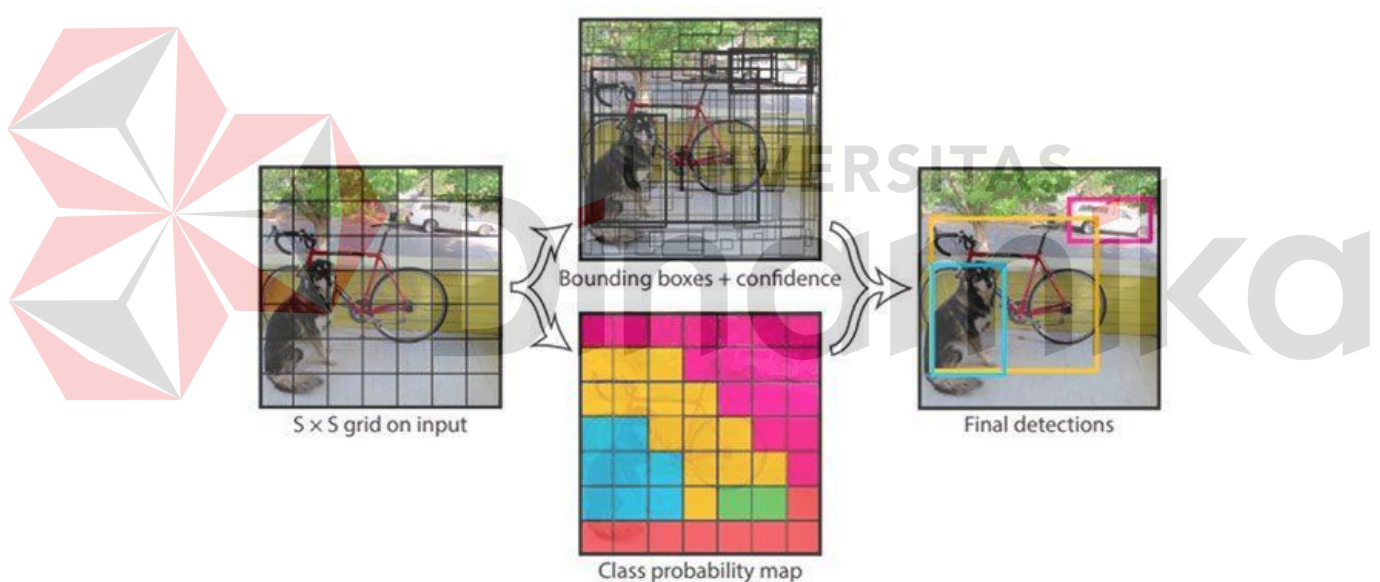
Roboflow menyediakan berbagai alat yang memudahkan proses anotasi, seperti pembuatan *bounding box* untuk angka pada meter air. Selain itu, fitur augmentasi gambar memungkinkan *dataset* diperbesar secara *virtual* dengan variasi pencahayaan, rotasi, atau *noise*, sehingga model yang dilatih menjadi lebih *robust* terhadap kondisi lingkungan yang beragam.

Dalam pengujian, dataset yang diolah melalui Roboflow akan digunakan untuk melatih model YOLOv5. Roboflow juga mendukung integrasi dengan berbagai *framework deep learning* seperti TensorFlow dan PyTorch, sehingga fleksibilitas *platform* ini sangat membantu dalam pengembangan sistem berbasis *Computer Vision*. (Brad Dwyer, 2023)

2.4 YOLO

2.4.1 Konsep Dasar YOLO

YOLO (*You Only Look Once*) adalah salah satu algoritma deteksi objek yang paling populer dalam dunia *computer vision*, khususnya karena kecepatan dan efisiensi prosesnya. Tidak seperti metode deteksi objek tradisional yang memproses gambar secara bertahap (misalnya dengan teknik *sliding window* atau *region proposal*), YOLO hanya membutuhkan satu kali pemrosesan untuk mendeteksi semua objek dalam gambar. Artinya, seluruh gambar diolah sekaligus melalui jaringan konvolusi (*Convolutional Neural Network/CNN*) dan *output*-nya langsung berupa koordinat *bounding box* dan prediksi kelas untuk setiap objek yang terdeteksi. (Ultralytics, 2023)



Gambar 2.4 Konsep YOLO
(Sumber: <https://pyimageresearch/>)

Dengan pendekatan ini, YOLO sangat cocok untuk aplikasi *real-time* seperti *video surveillance*, *autonomous driving*, hingga pembacaan meter otomatis berbasis kamera *webcam* seperti yang kamu bangun. Sistem dapat mendeteksi objek dalam waktu sangat singkat, bahkan hanya dalam hitungan milidetik, sehingga hasil deteksi dapat langsung diolah atau dikirim ke sistem *monitoring* lainnya.

2.4.2 Arsitektur dan Cara Kerja YOLO

YOLOv5 merupakan salah satu varian YOLO yang dikembangkan oleh komunitas Ultralytics dan sangat populer karena ringan, mudah digunakan, serta hasil deteksinya yang akurat. Pada dasarnya, YOLOv5 membagi gambar *input* ke dalam *grid*, lalu setiap *grid* bertanggung jawab untuk mendeteksi objek yang berada di wilayah tersebut. *Output* yang dihasilkan berupa *bounding box* (lokasi objek), *confidence score* (tingkat keyakinan deteksi), dan prediksi kelas objek.

YOLOv5 terdiri dari beberapa komponen utama:

1. *Backbone*: Bagian ini berfungsi untuk mengekstrak fitur penting dari gambar (menggunakan teknik seperti CSPDarknet53).
2. *Neck*: Merupakan bagian *intermediate* yang menggabungkan dan mengatur fitur-fitur dari *backbone* agar siap digunakan untuk deteksi. Biasanya menggunakan FPN (*Feature Pyramid Network*) atau PANet.
3. *Head*: Bagian ini bertanggung jawab menghasilkan prediksi akhir berupa *bounding box* dan label kelas dari fitur yang telah diproses sebelumnya.
4. Keunggulan YOLOv5 dibanding pendahulunya atau metode lain antara lain kecepatan inferensi tinggi, mudah di-*custom* melalui *transfer learning*, serta mendukung banyak perangkat keras (PC, Raspberry Pi, Jetson Nano, dsb).

2.4.3 Alur Training dan Inferensi YOLOv5

Training YOLOv5 diawali dengan menyiapkan *dataset* yang sudah dianotasi, yaitu gambar-gambar yang sudah diberi *bounding box* pada area objek target (misalnya angka pada meter air). Selanjutnya, model dilatih untuk mengenali pola dan fitur objek dalam gambar, sehingga saat di-inferensi, model dapat secara otomatis mendeteksi dan mengklasifikasikan objek pada gambar baru yang belum pernah dikenali sebelumnya.

Inferensi YOLOv5 berjalan sangat cepat karena seluruh proses deteksi dilakukan dalam satu kali *forward pass* pada jaringan *neural network*. Ini berbeda dengan metode lain yang mungkin harus melakukan deteksi objek berkali-kali untuk satu gambar. Setelah deteksi, *output* berupa posisi *bounding box* dan prediksi label langsung dapat digunakan untuk keperluan lebih lanjut, misalnya mengekstrak angka meter air untuk dikirim ke sistem IoT. (Chen, 2020)

2.4.4 Keunggulan YOLO dalam Implementasi Otomatisasi

Beberapa keunggulan YOLO yang membuatnya sangat ideal untuk tugas pembacaan meter air otomatis antara lain:

1. *Real-time Processing*: YOLO dapat memproses video/gambar dalam waktu nyata dengan latensi sangat rendah.
2. Akurasi Tinggi: Terutama untuk objek yang ukurannya jelas dan memiliki pola visual kuat seperti digit pada meter air.
3. *End-to-End Pipeline*: Tidak memerlukan tahapan deteksi region secara terpisah, sehingga alur kerjanya lebih efisien dan mudah diintegrasikan ke sistem lain.
4. Skalabilitas: YOLOv5 dapat digunakan di berbagai *platform*, dari PC hingga perangkat *edge* yang sumber dayanya terbatas.

2.5 Python

Python adalah bahasa pemrograman yang terkenal dengan sintaksis yang sederhana dan mudah dipelajari, sehingga sering menjadi pilihan utama dalam pengembangan aplikasi otomasi dan pemrosesan data. Python juga didukung oleh komunitas besar yang menghasilkan banyak pustaka siap pakai, seperti OpenCV untuk pengolahan citra, PyTorch untuk *deep learning*, dan Paho MQTT untuk komunikasi IoT. Kombinasi pustaka ini memungkinkan pengembang membangun sistem kompleks dengan efisien.



Gambar 2.5 Bahasa pemrograman Python
(Sumber: <https://wikipedia.org/>)

Selain itu, Python sangat kompatibel untuk integrasi berbagai perangkat keras maupun *software*. Di bidang otomatisasi, Python kerap digunakan untuk mengatur alur data dari pengambilan gambar, pemrosesan, hingga pengiriman data ke *server* atau perangkat *mobile*. Keunggulan inilah yang membuat Python sangat relevan dalam sistem pembacaan meter air otomatis berbasis *computer vision* dan IoT. (Lutz, 2021)

2.6 OpenCV

OpenCV adalah pustaka *open source* yang banyak digunakan dalam pengembangan aplikasi *computer vision* dan pemrosesan citra. Pustaka ini menyediakan berbagai macam fungsi dan algoritma yang memudahkan pengembang dalam melakukan operasi dasar seperti pembacaan gambar, konversi warna, deteksi tepi, hingga deteksi objek. OpenCV juga mendukung integrasi dengan berbagai bahasa pemrograman, salah satunya Python, sehingga sangat fleksibel dalam penggunaannya.



Gambar 2.6 OpenCV Python
(Sumber: <https://opencv.org/>)

Dalam implementasinya pada sistem otomatisasi pembacaan meter air, OpenCV berperan penting untuk menangkap gambar dari *webcam*, melakukan pra-pemrosesan seperti *cropping* atau *thresholding*, serta menampilkan hasil deteksi dalam bentuk visual *bounding box*. Integrasi OpenCV dengan algoritma deteksi objek seperti YOLOv5 semakin mempercepat proses pengolahan data secara *real-time*. (OpenCV, 2021)

2.7 *Internet of Things*

Internet of Things (IoT) adalah ekosistem perangkat fisik yang saling terhubung melalui jaringan *internet* dan mampu bertukar data secara otomatis. Perangkat-perangkat ini biasanya dilengkapi dengan sensor, aktuator, dan modul komunikasi, sehingga dapat memonitor kondisi lingkungan secara *real-time* dan mengirimkan data ke *server* atau perangkat lain untuk dianalisis lebih lanjut.



Gambar 2.7 IoT
(Sumber: <https://diskominfo.bandungkab.go.id/>)

Dalam pengelolaan air, IoT memungkinkan integrasi antara meter air, sensor, dan aplikasi *monitoring* sehingga informasi penggunaan air dapat diakses secara langsung oleh pengguna atau pihak terkait. Hal ini membuka peluang pengembangan sistem *monitoring* yang lebih cerdas, responsif, dan efisien tanpa bergantung pada pembacaan manual. (Madakam, 2015)

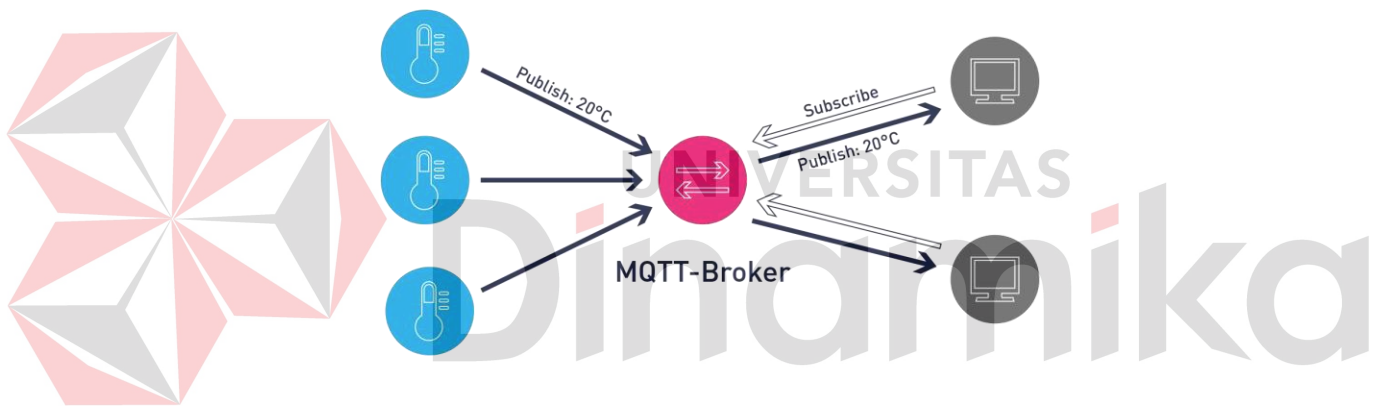
2.8 MQTT

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi berbasis *publish-subscribe* yang dirancang untuk perangkat dengan sumber daya terbatas dan jaringan yang tidak stabil. Protokol ini diperkenalkan pertama kali pada tahun 1999 dan dikembangkan agar efisien, ringan, serta mampu mengirimkan data secara andal dengan latensi rendah. MQTT sangat cocok digunakan dalam aplikasi *Internet of Things* (IoT) yang membutuhkan pengiriman data secara *real-time*, seperti sistem *monitoring* dan otomasi.

Keunikan MQTT terletak pada penggunaan model komunikasi *broker-based*, di mana setiap perangkat tidak saling terhubung langsung, melainkan melalui satu perantara yang disebut *broker*. Dengan cara ini, pengelolaan pesan menjadi lebih sederhana dan perangkat dapat beroperasi tanpa harus selalu aktif atau terhubung langsung dengan perangkat lain di jaringan.

Arsitektur MQTT terdiri dari tiga komponen utama, yaitu:

1. *Publisher*: Perangkat atau aplikasi yang mengirimkan data (pesan) ke *broker*.
2. *Broker*: *Server* yang bertugas menyalurkan pesan dari *publisher* ke *subscriber* sesuai dengan topik yang ditentukan.
3. *Subscriber*: Perangkat atau aplikasi yang berlangganan pada topik tertentu dan menerima data dari *broker* ketika ada pesan baru.



Gambar 2.8 Protokol MQTT
(Sumber: <https://paessler.com/>)

Proses kerja MQTT sangat sederhana. *Publisher* mengirimkan pesan pada suatu topik, lalu *broker* menerima dan menyalurkannya ke semua *subscriber* yang berlangganan pada topik tersebut. Model *publish-subscribe* ini memudahkan pengembangan sistem IoT yang skalabel dan fleksibel, karena perangkat dapat dengan mudah ditambah atau dihapus tanpa mempengaruhi komunikasi antar perangkat lainnya.

Pada sistem pembacaan meter air otomatis, MQTT dapat digunakan untuk mengirimkan hasil pembacaan digit meter air dari perangkat (seperti komputer yang terhubung dengan *webcam*) ke *smartphone* pengguna. Data dikirim secara berkala atau saat terjadi perubahan nilai, sehingga pengguna dapat memantau penggunaan air secara *real-time* melalui aplikasi IoT MQTT Panel. (Darwis, 2021)

2.9 IoT MQTT Panel

IoT MQTT Panel adalah aplikasi *mobile* berbasis Android yang dirancang untuk memudahkan pengguna dalam memantau, mengontrol, dan berinteraksi dengan berbagai perangkat IoT melalui protokol MQTT. Aplikasi ini berfungsi sebagai antarmuka pengguna (*user interface*) yang menghubungkan *smartphone* dengan perangkat-perangkat IoT secara *real-time*. Dengan IoT MQTT Panel, pengguna dapat membuat *dashboard* kustom berisi berbagai *widget* seperti indikator nilai, tombol kontrol, grafik historis, dan lain-lain, yang semuanya dapat dikonfigurasi untuk menampilkan atau mengirim data melalui topik MQTT tertentu. Fitur *drag-and-drop* yang dimiliki aplikasi ini membuat proses pembuatan panel pemantauan menjadi sangat intuitif, sehingga pengguna dari berbagai latar belakang teknis dapat menggunakannya dengan mudah.



Gambar 2.9 IoT MQTT Panel
(Sumber: <https://google.play/>)

Selain itu, IoT MQTT Panel mendukung berbagai jenis *payload* data, mulai dari teks, angka, hingga gambar, serta mampu menangani beberapa koneksi *broker* MQTT sekaligus.

BAB III

METODOLOGI PENELITIAN

Tujuan Metodologi penelitian ini untuk memastikan bahwa setiap tahapan dilakukan secara sistematis dan menghasilkan solusi yang sesuai dengan permasalahan yang diangkat. Tahapan-tahapan dalam penelitian ini meliputi:

3.1 Studi Literatur

Langkah pertama dalam penelitian ini adalah melakukan studi literatur untuk memahami konsep dan teknologi yang relevan, termasuk *Computer Vision*, YOLO, dan *platform* pendukung seperti Roboflow. Studi literatur ini dilakukan melalui jurnal ilmiah, buku referensi, dokumentasi teknis, dan sumber daring terpercaya.

Selain itu, penelitian sebelumnya yang sejenis juga ditinjau untuk mengidentifikasi pendekatan terbaik dan potensi masalah yang mungkin muncul. Dengan memahami teori dan praktik yang telah ada, penelitian ini dapat dirancang dengan lebih matang, serta menghasilkan inovasi yang signifikan.

3.2 Persiapan Alat dan Bahan

Dalam pelaksanaan penelitian ini, dilakukan persiapan alat dan bahan secara terencana agar sistem otomatisasi pembacaan meter air dapat diimplementasikan dan diuji dengan optimal. Pemilihan alat dan bahan disesuaikan dengan kebutuhan pengembangan sistem berbasis *computer vision* menggunakan YOLOv5 dan integrasi IoT melalui protokol MQTT. Persiapan ini meliputi perangkat keras utama, perangkat lunak pendukung, serta bahan-bahan penunjang seperti *dataset* gambar dan alat anotasi.

3.2.1 Alat

1. Kamera/*Webcam*: Digunakan untuk menangkap citra meter air secara *real-time*.
2. Komputer/Laptop: Berfungsi sebagai perangkat utama untuk pemrosesan citra dan menjalankan model YOLOv5 serta skrip Python.
3. *Smartphone* Android: Digunakan untuk memantau hasil pembacaan melalui aplikasi IoT MQTT Panel.
4. Jaringan Wi-Fi/*Router*: Mendukung konektivitas antara perangkat pemrosesan dan *smartphone*.
5. *Broker* MQTT: Bisa berupa server lokal (seperti *Mosquitto*) atau *broker cloud* untuk menyalurkan data dari perangkat ke aplikasi *mobile*. Pada tugas akhir ini akan menggunakan *broker* milik Universitas Dinamika (mqtt.dinamika.ac.id)

3.2.2 Bahan

1. *Dataset* Gambar Meter Air: Kumpulan gambar meter air yang akan digunakan untuk pelatihan dan pengujian model YOLOv5.
2. File Anotasi: Berisi informasi *bounding box* angka pada meter air dalam format yang sesuai dengan YOLO.
3. Perangkat Lunak:
 - a. Python (minimal versi 3.10)
 - b. Library: PyTorch, OpenCV, Ultralytics YOLOv5, Paho-MQTT
 - c. Aplikasi IoT MQTT Panel (Android)
4. Akun/*Server* MQTT: Diperlukan jika menggunakan *broker* MQTT berbasis *cloud*.

Dengan persiapan alat dan bahan yang matang, seluruh proses penelitian mulai dari pengambilan data, pelatihan model, hingga implementasi *monitoring* berbasis IoT dapat berjalan dengan baik dan terukur. Setiap alat dan bahan yang dipilih memiliki peran penting dalam mendukung keberhasilan sistem yang dikembangkan.

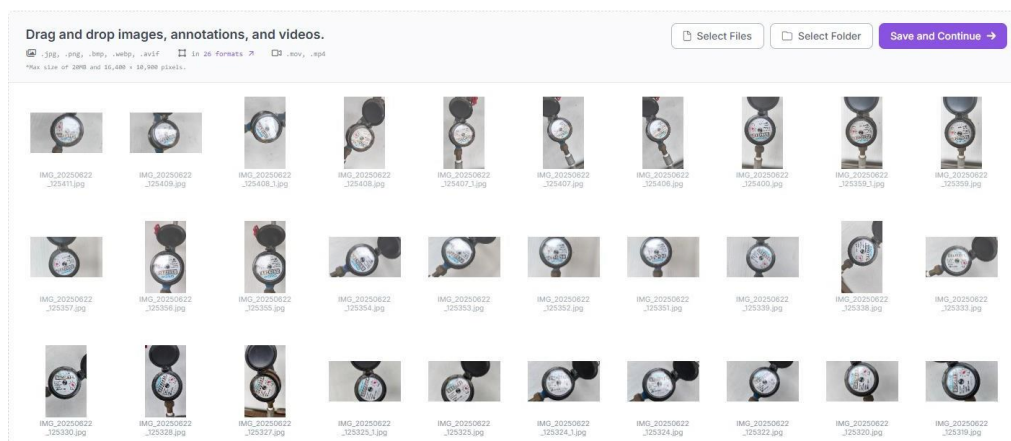
3.3 Pengolahan *Dataset*

Dataset yang digunakan dalam penelitian ini berupa gambar meter air digital dan analog yang akan diproses untuk pelatihan model. Tahapan pengumpulan *dataset* meliputi:

1. *Pengumpulan Dataset*: Mengambil gambar meter air dari berbagai sumber, seperti foto langsung menggunakan kamera atau *dataset open-source*.
2. *Anotasi Dataset*: Melakukan anotasi pada *dataset* menggunakan *bounding box* untuk menandai angka-angka pada meter air. Proses ini dilakukan menggunakan *platform* Roboflow.
3. *Download Dataset*: Setelah melakukan proses anotasi langkah selanjutnya adalah *mendownload dataset* menggunakan format khusus YOLOv5.

3.3.1 Pengumpulan *Dataset*

Pada tahap pengumpulan *dataset*, peneliti menggunakan metode kombinasi antara pengambilan gambar mandiri dan pemanfaatan *dataset* yang telah tersedia secara *daring*. Pengambilan gambar mandiri dilakukan dengan cara mendokumentasikan berbagai meter air secara langsung menggunakan kamera, sehingga diperoleh variasi kondisi pencahayaan, sudut pengambilan, dan tipe meter air yang berbeda. Gambar 3.1 menunjukkan daftar gambar *dataset water meter* yang di *upload* pada *website* Roboflow yang berjumlah total 2668 gambar.



Gambar 3.1 Daftar *dataset*

Untuk memperkaya keragaman data dan meningkatkan performa generalisasi model, pada tugas akhir ini juga ditambahkan beberapa *dataset water meter* dari sumber *online*, khususnya dari *platform* Roboflow. Dengan demikian, *dataset* yang diperoleh menjadi lebih beragam baik dari segi bentuk, jenis, maupun karakter angka pada meter air, yang penting untuk meningkatkan akurasi deteksi pada implementasi nyata.

3.3.2 Anotasi *Dataset*

Selanjutnya, pada tahap anotasi *dataset*, seluruh gambar yang telah terkumpul kemudian diberi label menggunakan fitur anotasi pada *website* Roboflow. Proses anotasi ini dilakukan dengan menandai area digit atau angka pada citra meter air menggunakan *bounding box*, sesuai kebutuhan deteksi objek pada model YOLOv5. Contoh proses anotasi pada sebuah gambar dapat dilihat pada Gambar 3.2.



Gambar 3.2 Proses anotasi *dataset*

Anotasi yang presisi sangat penting agar model dapat belajar mengenali posisi dan bentuk angka dengan tepat, sehingga menghasilkan prediksi yang akurat ketika sistem dijalankan. *Platform* Roboflow dipilih karena menyediakan antarmuka anotasi yang intuitif serta mendukung kolaborasi dan manajemen *dataset* secara efisien.



Gambar 3.3 Kelas *object detection*

Dalam proses anotasi untuk deteksi objek, pembuatan kelas-kelas ini sangat penting untuk melatih model agar dapat mengenali dan membedakan setiap objek yang relevan pada gambar. Berikut penjelasan terkait alasan dan tujuan penggunaan 11 kelas tersebut:

1. Kelas Angka 0–9

Setiap angka dari 0 hingga 9 diberikan kelas tersendiri. Tujuannya adalah agar model YOLOv5 mampu melakukan deteksi dan klasifikasi angka secara individu pada area meter air. Dengan demikian, setiap digit pada meter air dapat dikenali secara presisi sehingga memungkinkan pembacaan nilai meter secara otomatis dan akurat.

2. Kelas “dig” (Deretan Angka/Digit Area)

Selain kelas individu angka, dibuat satu kelas khusus "dig" yang digunakan untuk menandai seluruh barisan digit atau area yang berisi kumpulan angka pada meter air. Dengan adanya kelas ini, proses deteksi dapat dilakukan secara dua tahap, yaitu:

- Tahap pertama, model mendeteksi area atau kotak besar yang berisi seluruh deretan angka (*bounding box* "dig").
- Tahap kedua, di dalam area yang sudah terdeteksi tersebut, model kemudian mendeteksi dan mengenali setiap angka satu per satu (kelas 0–9).

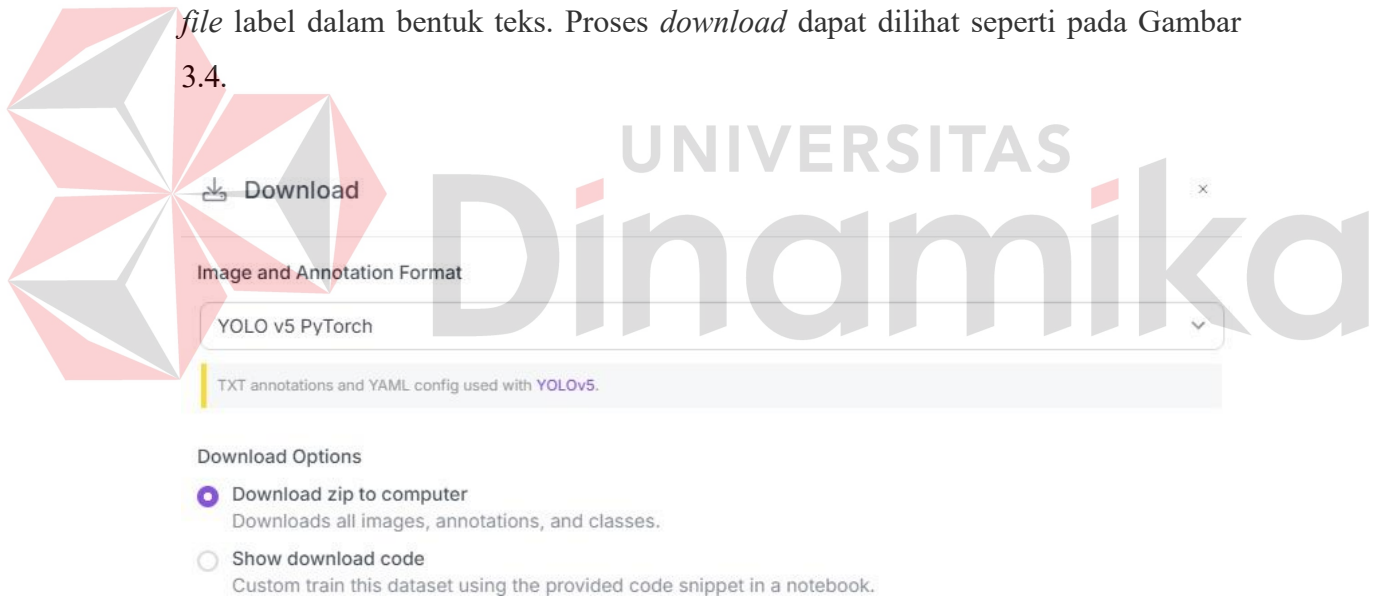
Pendekatan ini memiliki beberapa keunggulan:

- Mempersempit Area Pencarian: Dengan mendeteksi area digit terlebih dahulu, model dapat lebih fokus mendeteksi angka pada area yang relevan, sehingga mengurangi kemungkinan salah deteksi pada area gambar lain yang bukan digit.

2. Meningkatkan Akurasi: Proses dua tahap ini membantu meminimalkan deteksi palsu (*false positive*) dan meningkatkan akurasi pembacaan, karena hanya angka pada area "dig" yang diproses lebih lanjut.
3. Fleksibel untuk Meter Berbeda: Dengan anotasi seperti ini, sistem menjadi lebih adaptif untuk berbagai tipe water meter yang mungkin memiliki tata letak deretan angka yang berbeda.

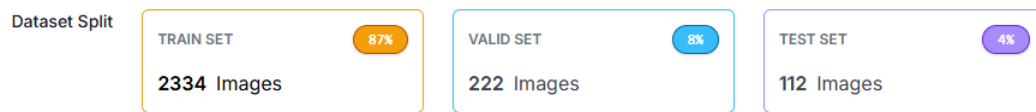
3.3.3 Download Dataset

Tahap terakhir adalah mengunduh *dataset* yang telah selesai dianotasi dalam format yang sesuai dengan kebutuhan YOLOv5. Setelah proses anotasi selesai pada Roboflow, dataset diekspor ke format YOLOv5, yang terdiri atas file gambar dan file label dalam bentuk teks. Proses *download* dapat dilihat seperti pada Gambar 3.4.



Gambar 3.4 Proses *download dataset* YOLOv5

Dari total 2668 gambar yang berhasil dikumpulkan, *dataset* tersebut kemudian dibagi menjadi tiga *subset*, yaitu 87% untuk data *train* (pelatihan), 8% untuk validasi (*validation*), dan 4% untuk pengujian (*test*). Pembagian ini dilakukan secara proporsional untuk memastikan bahwa model mendapatkan cukup banyak data untuk belajar (*training*), memiliki data tersendiri untuk evaluasi selama proses pelatihan (*validation*), serta data khusus yang benar-benar baru untuk mengukur performa model secara obyektif setelah pelatihan selesai (*test*).

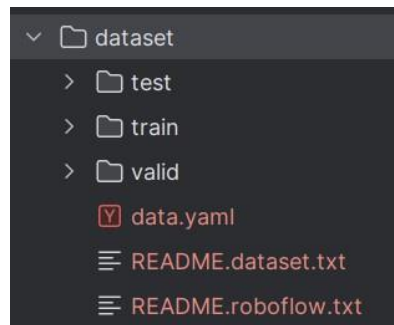


Gambar 3.5 Jumlah *dataset split*

Pembagian *dataset* ke dalam tiga *subset* ini sangat penting dalam pengembangan model berbasis *machine learning*. *Data train* digunakan agar model mampu mempelajari berbagai variasi pola pada digit meter air. Data validasi diperlukan untuk memonitor performa model selama pelatihan dan membantu menghindari *overfitting*, yaitu kondisi di mana model terlalu “hafal” data latih namun gagal mengenali data baru. Sementara itu, data test dipakai untuk menguji sejauh mana model benar-benar mampu melakukan prediksi pada data yang sama sekali belum pernah dilihat sebelumnya. Dengan proporsi *train* yang besar, model memperoleh referensi pola yang luas; sedangkan validasi dan *test* yang lebih kecil namun tetap signifikan, cukup untuk mengukur performa tanpa mengurangi jumlah data pelatihan secara drastis. Pembagian yang seimbang dan terstruktur ini merupakan standar praktik terbaik dalam *machine learning* agar hasil yang diperoleh benar-benar mewakili kemampuan sistem pada dunia nyata.

3.4 *Training Model*

Setelah proses anotasi dan ekspor dataset dari Roboflow selesai, *dataset* yang diunduh akan memiliki struktur folder yang sangat terorganisir dan sudah sesuai standar pelatihan model YOLOv5.



Gambar 3.6 Struktur folder *dataset* YOLOv5

Secara umum, di dalam folder utama *dataset* akan terdapat tiga *subfolder* utama, yaitu *train*, *val*, dan *test*, serta satu file konfigurasi utama yaitu *data.yaml*. Folder *train* berisi data gambar yang digunakan untuk proses pelatihan (*training*) model YOLOv5. Di dalamnya terdapat dua subfolder utama:

1. *images/*: berisi seluruh file gambar (biasanya berekstensi *.jpg* atau *.png*) yang digunakan sebagai data latih.
2. *labels/*: berisi file teks (*.txt*) yang merupakan hasil anotasi untuk setiap gambar. Setiap file label memiliki nama yang sama dengan file gambar yang bersesuaian, dan berisi baris-baris data dengan format YOLO, yaitu: *class x_center y_center width height*.

Folder validasi berisi data yang digunakan untuk mengevaluasi performa model selama proses *training*. Struktur subfoldernya sama dengan folder *train*. Folder *test* digunakan untuk menguji performa akhir model setelah pelatihan selesai. Data dalam folder ini tidak pernah dilihat oleh model selama pelatihan, sehingga hasil evaluasinya dapat menggambarkan kemampuan model dalam menghadapi data baru.

File *data.yaml* adalah file konfigurasi utama yang diperlukan oleh YOLOv5 untuk mengenali struktur *dataset*. File ini berisi informasi tentang *path dataset*, daftar kelas (*class names*), dan jumlah kelas.

```

train: ../dataset/train/images
val: ../dataset/val/images
test: ../dataset/test/images

nc: 11
names: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'dig']

```

Gambar 3.7 Isi data data.yaml

Setelah *dataset* siap, tahap berikutnya adalah melatih model YOLOv5 dengan menggunakan *dataset* yang telah dianotasi. Proses ini melibatkan beberapa iterasi untuk memastikan model mampu mempelajari pola angka secara efektif. Parameter seperti *learning rate*, *batch size*, dan *epoch* disesuaikan untuk mengoptimalkan performa model. Seluruh eksperimen dilakukan dengan mengganti-ganti nilai ketiga parameter tersebut sesuai kombinasi yang terdapat pada tabel pengujian. Proses pelatihan dilakukan dengan beberapa skenario, yakni:

1. Skenario pertama: *Learning rate* tetap, *batch size* dan *epoch* divariasikan.
2. Skenario kedua: *Batch size* tetap, *learning rate* dan *epoch* divariasikan.
3. Skenario ketiga: *Epoch* tetap, *learning rate* dan *batch size* divariasikan.

Setiap skenario dieksekusi dengan menjalankan *script training* YOLOv5 pada program python secara berulang, mengganti nilai parameter pada setiap percobaan. Hal ini bertujuan untuk mengetahui pengaruh masing-masing parameter terhadap hasil akurasi model.

```

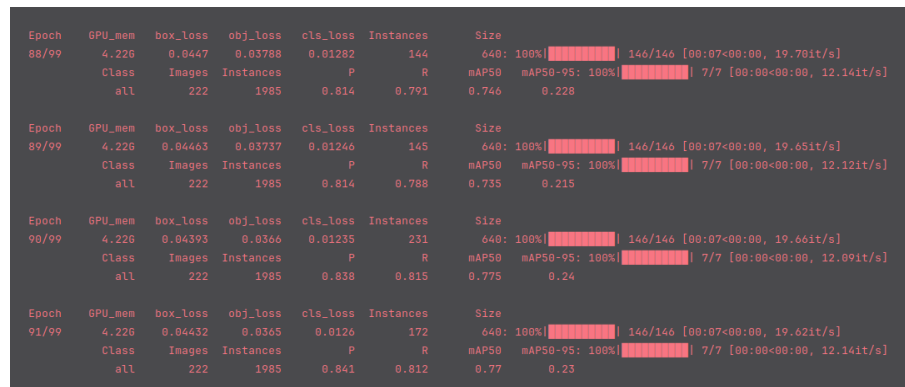
'imgsz': 640,
'batch': 15,
'epochs': 100,
'data': r'C:\TA\OCR\yolov5\dataset\data.yaml', # ganti path sesuai lokasi data.yaml
'weights': 'yolov5s.pt',
'project': r'C:\TA\OCR\yolov5\runs\train',
'name': 'custom_yolo',
'lr0': 0.001,

```

Gambar 3.8 Konfigurasi parameter

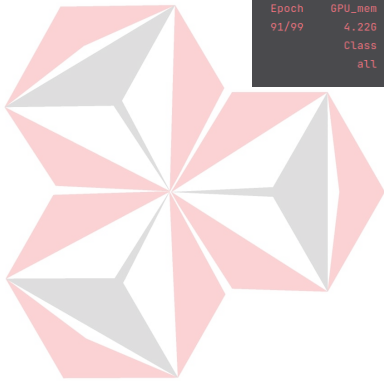
Gambar 3.8 menunjukkan cuplikan kode konfigurasi parameter *training* untuk model YOLOv5 yang digunakan dalam pelatihan deteksi digit pada meter air.

Dalam kode tersebut, ditetapkan beberapa parameter penting, antara lain *batch size*, *learning rate*, dan jumlah *epoch*. *Output* hasil *training* akan disimpan pada direktori *runs/train* dengan penamaan khusus *custom_yolo*. Konfigurasi ini dapat memastikan seluruh proses *training* berjalan sistematis dan dapat direplikasi, sehingga memudahkan dalam membandingkan performa model pada setiap eksperimen yang berbeda.



Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
88/99	4.226	0.0447	0.03788	0.01282	144	640: 100% [00:07:00:00, 19.70it/s]
Class	Images	Instances	P	R	mAP50	mAP50-95: 100% [00:00:00:00, 12.14it/s]
all	222	1985	0.814	0.791	0.746	0.228
89/99	4.226	0.04463	0.03737	0.01246	145	640: 100% [00:07:00:00, 19.65it/s]
Class	Images	Instances	P	R	mAP50	mAP50-95: 100% [00:00:00:00, 12.12it/s]
all	222	1985	0.814	0.788	0.735	0.215
90/99	4.226	0.04393	0.0366	0.01235	231	640: 100% [00:07:00:00, 19.66it/s]
Class	Images	Instances	P	R	mAP50	mAP50-95: 100% [00:00:00:00, 12.09it/s]
all	222	1985	0.838	0.815	0.775	0.24
91/99	4.226	0.04432	0.0365	0.0126	172	640: 100% [00:07:00:00, 19.62it/s]
Class	Images	Instances	P	R	mAP50	mAP50-95: 100% [00:00:00:00, 12.14it/s]
all	222	1985	0.841	0.812	0.77	0.23

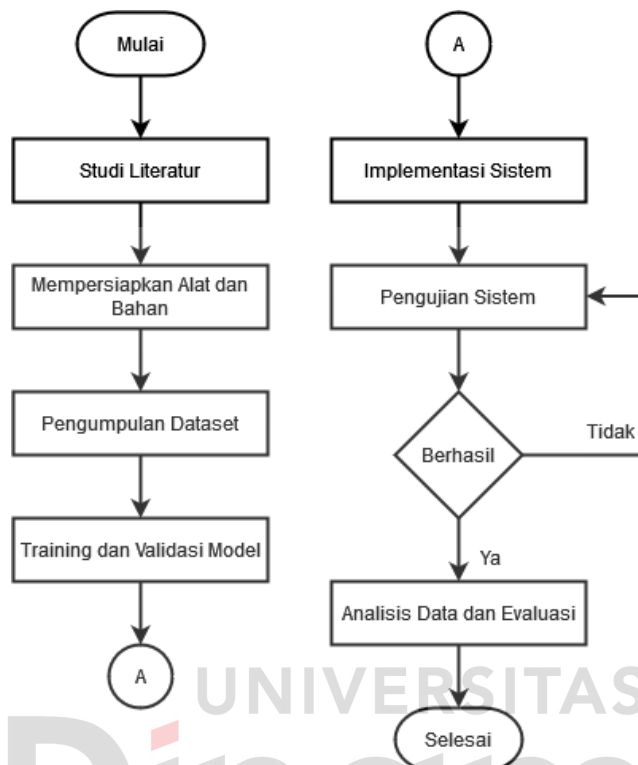
Gambar 3.9 Proses *training*



UNIVERSITAS
Dinamika

3.5 Flowchart

3.5.1 Flowchart Tahapan Kerja



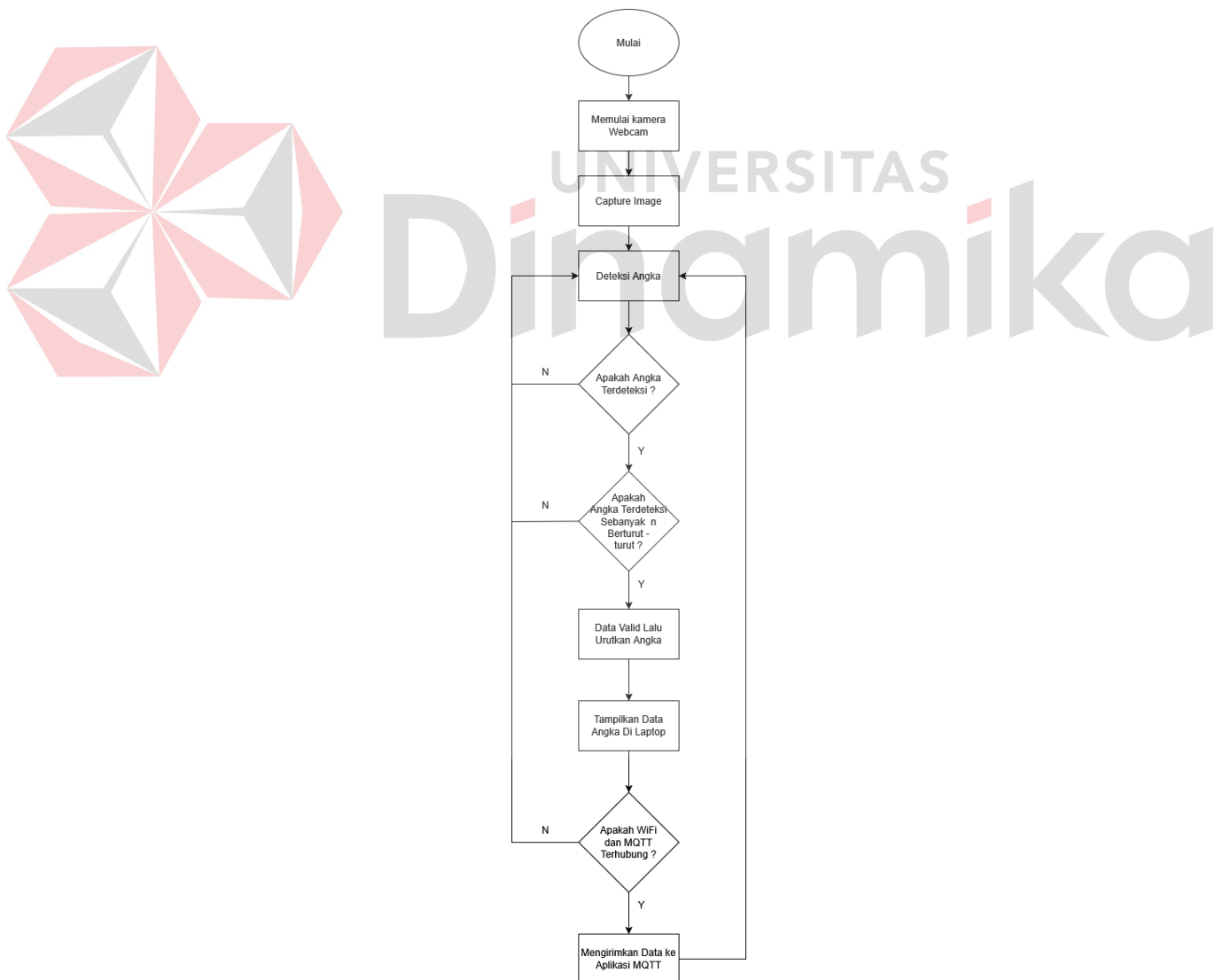
Gambar 3.10 Flowchart tahapan kerja

Flowchart pada gambar di atas menggambarkan rangkaian tahapan kerja penelitian yang sistematis dalam pengembangan sistem otomatisasi pembacaan meter air berbasis *computer vision* dan YOLOv5. Proses dimulai dari tahap studi literatur untuk mengkaji referensi dan teori yang relevan, dilanjutkan dengan persiapan alat dan bahan yang dibutuhkan, termasuk perangkat keras, perangkat lunak, serta sumber daya pendukung lainnya. Setelah semua persiapan selesai, langkah berikutnya adalah mengumpulkan *dataset* dengan memadukan pengambilan gambar secara mandiri dan pemanfaatan data dari sumber *online* agar data yang digunakan lebih variatif. *Dataset* yang telah terkumpul kemudian digunakan untuk melatih dan memvalidasi model YOLOv5 agar dapat mendeteksi digit pada meter air secara akurat.

Tahapan berikutnya adalah implementasi sistem, di mana model yang telah terlatih diintegrasikan dengan sistem komunikasi IoT, sehingga dapat digunakan

untuk pembacaan meter air secara otomatis dan pengiriman data ke perangkat *monitoring*. Setelah implementasi selesai, dilakukan pengujian sistem untuk memastikan bahwa seluruh komponen berfungsi dengan baik. Jika pada tahap pengujian sistem belum berjalan sesuai harapan, maka dilakukan evaluasi dan perbaikan hingga sistem dinyatakan berhasil. Proses penelitian diakhiri dengan analisis data hasil pengujian dan evaluasi kinerja sistem secara menyeluruh. Seluruh tahapan dalam *flowchart* ini menunjukkan pendekatan yang berurutan namun fleksibel, dengan adanya siklus evaluasi untuk menjamin keberhasilan dan kualitas akhir dari sistem yang dikembangkan.

3.5.2 Flowchart Cara Kerja Alat



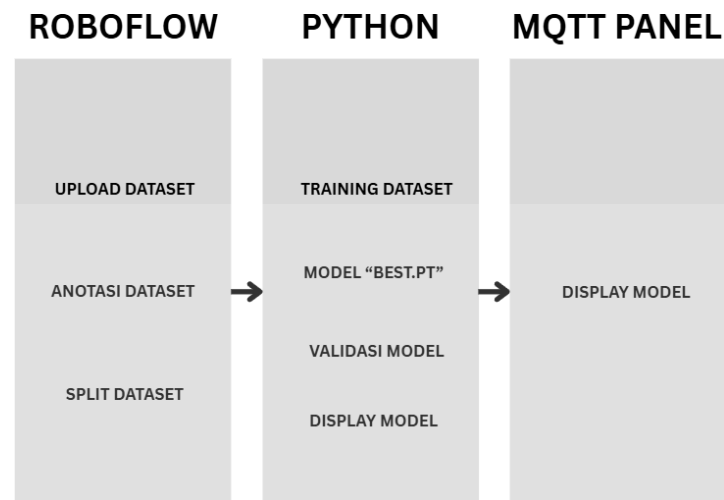
Gambar 3.11 Flowchart cara kerja alat

Flowchart pada gambar di atas menjelaskan alur proses kerja sistem pembacaan meter air otomatis yang menggunakan webcam dan algoritma deteksi angka. Proses dimulai dengan inisialisasi sistem, kemudian webcam diaktifkan untuk mengambil citra meter air secara *real-time*. Selanjutnya, sistem melakukan deteksi angka pada gambar yang diambil oleh kamera. Jika angka belum berhasil terdeteksi, proses akan kembali mengulangi tahapan deteksi hingga angka berhasil dikenali. Setelah angka berhasil terdeteksi, sistem akan melakukan verifikasi konsistensi, yaitu memastikan bahwa data angka yang terdeteksi muncul sebanyak “n” (jumlah dapat dirubah pada program python) kali secara berturut-turut. Hal ini dilakukan untuk memastikan validitas dan kestabilan data yang dideteksi, sehingga mengurangi kemungkinan *error* akibat *noise* atau deteksi palsu.

Jika data angka telah terdeteksi dengan konsisten, maka data tersebut dianggap valid dan sistem akan mengurutkan angka-angka yang terdeteksi sesuai posisinya pada meter air. Setelah itu, angka hasil pembacaan akan ditampilkan pada laptop sebagai umpan balik kepada pengguna. Secara paralel, data angka tersebut juga dikirimkan ke aplikasi MQTT, sehingga dapat dimonitor secara jarak jauh melalui perangkat lain, seperti smartphone dengan aplikasi IoT MQTT Panel. Alur ini memastikan bahwa hanya data yang benar-benar valid dan akurat yang akan ditampilkan dan dikirimkan, sehingga sistem pembacaan meter air dapat berjalan secara otomatis, efisien, dan dapat diandalkan untuk kebutuhan *monitoring* penggunaan air.

3.6 Blok Diagram

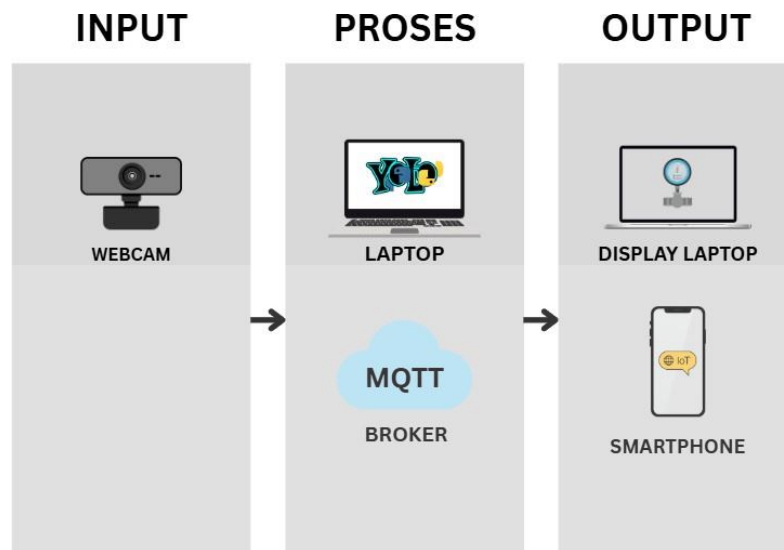
3.6.1 Blok Diagram Perangkat Lunak



Gambar 3.12 Blok diagram perangkat lunak

Gambar 3.12 menunjukkan diagram alur proses mulai dari tahap pengolahan dataset menggunakan *platform* Roboflow, dilanjutkan pelatihan dan validasi model di Python, hingga visualisasi hasil di aplikasi MQTT Panel. Pada bagian Roboflow, dataset diunggah, dianotasi, dan dibagi (*split*) menjadi data pelatihan, validasi, dan pengujian. Selanjutnya, *dataset* tersebut digunakan dalam proses *training* model YOLOv5 di Python, menghasilkan model terbaik ("*best.pt*") yang divalidasi dan hasil deteksinya divisualisasikan. Tahap akhir adalah menampilkan hasil model di aplikasi MQTT Panel sebagai bentuk pemantauan digital secara *real-time*.

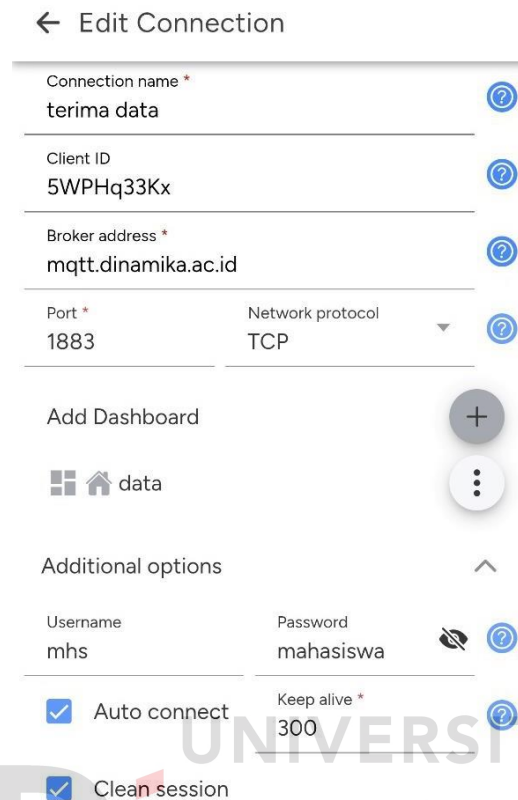
3.6.2 Blok Diagram Perangkat Keras



Gambar 3.13 Blok diagram perangkat keras

Gambar 3.13 memperlihatkan blok diagram arsitektur sistem secara keseluruhan. Alur dimulai dari input berupa *webcam* yang menangkap citra meter air, kemudian citra tersebut diproses pada laptop menggunakan model deteksi YOLOv5. Data hasil deteksi selanjutnya dikirimkan melalui *broker* MQTT. Pada sisi output, hasil pembacaan angka dapat langsung ditampilkan di layar laptop maupun diteruskan ke *smartphone* pengguna melalui aplikasi IoT berbasis MQTT untuk *monitoring* jarak jauh.

3.7 Aplikasi IoT MQTT Panel



← Edit Connection

Connection name *
terima data

Client ID
5WPHq33Kx

Broker address *
mqtt.dinamika.ac.id

Port *
1883

Network protocol
TCP

Add Dashboard

data

Additional options

Username
mhs

Password
mahasiswa

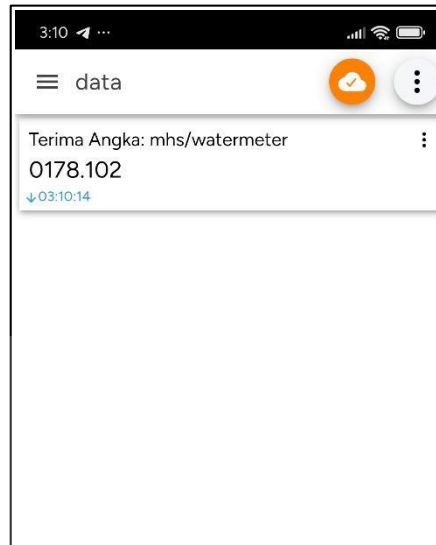
☒ Auto connect

Keep alive *
300

☒ Clean session

Gambar 3.14 Konfigurasi MQTT

Gambar ini adalah tampilan pengaturan koneksi pada aplikasi IoT MQTT Panel di *smartphone*. Pada pengaturan ini, pengguna menentukan parameter koneksi seperti nama koneksi, *Client ID*, alamat *broker* MQTT (misalnya mqtt.dinamika.ac.id), *port* yang digunakan (1883), dan protokol jaringan (TCP). Selain itu, terdapat pengaturan tambahan seperti *username*, *password*, opsi *auto connect*, *keep alive*, dan *clean session* yang semuanya digunakan untuk memastikan koneksi ke *broker* MQTT berjalan aman dan stabil sebelum data dapat diterima di aplikasi.



Gambar 3.15 Tampilan IoT MQTT Panel

Gambar 3.15 adalah tampilan penerimaan data di aplikasi MQTT Panel pada *smartphone*. Data yang diterima berupa angka hasil pembacaan meter air yang dikirim melalui topik tertentu, dalam contoh ini “mhs/watermeter”. Angka yang tampil di layar adalah hasil deteksi otomatis dari sistem, berikut waktu diterimanya data. Tampilan sederhana dan *real-time* ini memudahkan pengguna untuk memantau penggunaan air secara digital dan jarak jauh langsung dari perangkat *mobile*.

3.8 Implementasi Sistem

Setelah model dilatih, sistem mulai diimplementasikan dalam perangkat keras dan perangkat lunak. Langkah-langkah implementasi meliputi:

1. Integrasi Perangkat Keras: Menghubungkan kamera dengan laptop untuk pengambilan gambar meter air secara otomatis.
2. Integrasi Perangkat Lunak: Memasang model YOLOv5 untuk melakukan deteksi angka secara *real-time*. Setelah angka terdeteksi akan dikirimkan ke *smartphone* menggunakan komunikasi MQTT.

3.9 Pengujian Sistem

Pengujian dilakukan untuk memastikan sistem bekerja sesuai dengan spesifikasi. Tahapan ini meliputi:

1. Pengujian Akurasi Model: Mengukur tingkat akurasi deteksi angka oleh model YOLOv5 menggunakan *dataset* uji.
2. Pengujian Kecepatan Proses: Mengukur waktu yang dibutuhkan untuk melakukan deteksi angka.
3. Pengujian Stabilitas Sistem: Menguji sistem dalam kondisi operasional yang berbeda, seperti pencahayaan rendah atau gangguan konektivitas saat pengiriman data menggunakan MQTT.
4. Hasil pengujian dianalisis untuk menentukan apakah sistem memerlukan perbaikan atau sudah siap untuk diimplementasikan secara penuh.

3.10 Analisis Data dan Evaluasi

Setelah pengujian selesai, data yang dihasilkan dianalisis untuk mengevaluasi keberhasilan sistem. Analisis ini meliputi:

1. Evaluasi Akurasi dan Efisiensi Sistem: Membandingkan performa sistem dengan target yang telah ditetapkan.
2. Identifikasi Kelemahan: Mengidentifikasi potensi masalah atau kelemahan pada sistem yang memerlukan pengembangan lebih lanjut.

BAB IV

HASIL PENELITIAN

4.1 Validasi Hasil *Training* Data

Validasi hasil *training* data merupakan langkah penting untuk menilai seberapa baik model YOLOv5 yang telah dilatih mampu mengenali digit angka pada data yang belum pernah dilihat sebelumnya. Pada tahap ini, model diuji menggunakan dataset validasi yang terpisah dari data pelatihan. Proses validasi bertujuan untuk mengukur generalisasi model. Hasil validasi menjadi dasar evaluasi sebelum model diimplementasikan ke tahap pengujian sistem secara keseluruhan.

4.1.1 Peralatan yang Dibutuhkan

Peralatan dan perangkat lunak yang digunakan untuk mendukung penelitian ini, antara lain:

1. Komputer atau laptop dengan spesifikasi memadai
2. Instalasi Python dan IDE PyCharm
3. *Library* pendukung (PyTorch, YOLOv5, OpenCV)
4. *Dataset* validasi (berbeda dari *dataset training*)

4.1.2 Prosedur Validasi Training Data

Berikut adalah prosedur yang dijalankan untuk validasi training data:

1. Menyiapkan dan memuat model YOLOv5 hasil *training*.
2. Menyiapkan *dataset* validasi yang telah dianotasi sesuai format YOLOv5.
3. Melakukan proses prediksi menggunakan data validasi.
4. Menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan mAP (*mean Average Precision*).
5. Membandingkan hasil pada data training dan validasi untuk menilai generalisasi model.

4.1.3 Hasil Validasi *Training Data*

Tabel 4.1 Hasil *training learning rate* tetap

No.	Parameter Model YOLOv5					
	Learning Rate	Batch Size	Epoch	Precision	Recall	mAP50
1	0.001	10	50	0.82	0.76	0.75
		15	100	0.85	0.82	0.79
		20	150	0.87	0.84	0.81
		25	200	0.86	0.83	0.81
2	0.002	10	50	0.82	0.76	0.75
		15	100	0.85	0.82	0.79
		20	150	0.87	0.84	0.81
		25	200	0.86	0.83	0.81
3	0.003	10	50	0.82	0.76	0.75
		15	100	0.85	0.82	0.79
		20	150	0.87	0.84	0.81
		25	200	0.86	0.83	0.81
4	0.01	10	50	0.82	0.76	0.75
		15	100	0.85	0.82	0.79
		20	150	0.87	0.84	0.81
		25	200	0.86	0.83	0.81

Berikut kesimpulan yang dapat diambil dari Tabel 4.1 tentang hasil *training* YOLOv5 dengan *learning rate* tetap dan variasi *batch size* serta *epoch*:

1. Nilai *precision*, *recall*, dan mAP50 hampir sama untuk semua *learning rate* (0.001, 0.002, 0.003, dan 0.01) pada setiap kombinasi *batch size* dan *epoch*. Ini menunjukkan bahwa pada skenario ini, perubahan *learning rate* dalam rentang yang dicoba tidak memberikan pengaruh signifikan terhadap performa model.
2. Peningkatan *batch size* dan *epoch* cenderung meningkatkan hasil *precision*, *recall*, dan mAP50. Misalnya, ketika *epoch* naik dari 50 ke 200, *precision* naik dari 0.82 menjadi 0.86–0.87, *recall* dari 0.76 ke 0.83–0.84, dan mAP50 dari 0.75 ke 0.81. Hal ini menunjukkan bahwa *training* yang lebih lama (*epoch* lebih banyak) dan batch yang lebih besar membantu model belajar lebih baik.
3. Untuk semua kombinasi *learning rate*, hasil terbaik diperoleh pada *epoch* 150–200 dan *batch size* 20–25, dengan *precision* hingga 0.87, *recall* hingga 0.84, dan mAP50 hingga 0.81.

4. Tidak ada perubahan drastis (fluktuasi besar) pada metrik evaluasi di seluruh variasi parameter yang diuji, yang berarti model stabil dan tidak sensitif terhadap perubahan learning rate dalam eksperimen ini.

Tabel 4.2 Hasil *training batch size* tetap

No.	Parameter Model YOLOv5					
	Learning Rate	Batch Size	Epoch	Precision	Recall	mAP50
1	0.001	10	50	0.82	0.76	0.75
	0.002		100	0.85	0.82	0.79
	0.003		150	0.87	0.84	0.81
	0.01		200	0.86	0.83	0.81
2	0.001	15	50	0.82	0.76	0.75
	0.002		100	0.85	0.82	0.79
	0.003		150	0.87	0.84	0.81
	0.01		200	0.86	0.83	0.81
3	0.001	20	50	0.82	0.76	0.75
	0.002		100	0.85	0.82	0.79
	0.003		150	0.87	0.84	0.81
	0.01		200	0.86	0.83	0.81
4	0.001	25	50	0.82	0.76	0.75
	0.002		100	0.85	0.82	0.79
	0.003		150	0.87	0.84	0.81
	0.01		200	0.86	0.83	0.81

Berikut kesimpulan yang dapat diambil dari Tabel 4.2 Hasil *training batch size* tetap pada model YOLOv5:

1. Terlihat bahwa nilai *precision*, *recall*, dan mAP50 hampir sama untuk setiap *learning rate* (0.001, 0.002, 0.003, 0.01) dalam setiap kelompok *batch size* dan *epoch*. Artinya, variasi *learning rate* dalam rentang ini tidak terlalu mempengaruhi performa model untuk setiap *batch size* yang diuji.
2. Untuk *batch size* yang lebih besar (15, 20, 25), model cenderung menghasilkan nilai *precision*, *recall*, dan mAP50 yang sedikit lebih tinggi pada *epoch* yang lebih besar (150 dan 200), meskipun peningkatannya tidak terlalu signifikan. Ini menunjukkan bahwa penambahan *batch size* memiliki efek positif, terutama jika diikuti dengan peningkatan jumlah *epoch*.
3. Pada semua *batch size*, *precision*, *recall*, dan mAP50 meningkat seiring bertambahnya *epoch*. Misalnya, pada *batch size* 10 dan *learning rate* 0.001,

precision naik dari 0.82 (*epoch* 50) menjadi 0.87 (*epoch* 150), dan mAP50 dari 0.75 ke 0.81. Tren serupa juga terlihat pada *batch size* lainnya.

4. Model menunjukkan performa yang stabil (tidak ada penurunan atau kenaikan ekstrem pada metrik evaluasi) meskipun dilakukan perubahan nilai *learning rate* pada setiap *batch size* dan *epoch*.

Tabel 4.3 Hasil *training epoch* tetap

No.	Parameter Model YOLOv5					
	Learning Rate	Batch Size	Epoch	Precision	Recall	mAP50
1	0.001	10	50	0.82	0.76	0.75
	0.002	15		0.82	0.76	0.75
	0.003	20		0.82	0.76	0.75
	0.01	25		0.82	0.76	0.75
2	0.001	10	100	0.85	0.82	0.79
	0.002	15		0.85	0.82	0.79
	0.003	20		0.85	0.82	0.79
	0.01	25		0.85	0.82	0.79
3	0.001	10	150	0.87	0.84	0.81
	0.002	15		0.87	0.84	0.81
	0.003	20		0.87	0.84	0.81
	0.01	25		0.87	0.84	0.81
4	0.001	10	200	0.86	0.83	0.81
	0.002	15		0.86	0.83	0.81
	0.003	20		0.86	0.83	0.81
	0.01	25		0.86	0.83	0.81

Berikut kesimpulan dari Tabel 4.3 Hasil *training epoch* tetap pada model YOLOv5:

1. Dari tabel terlihat bahwa *precision*, *recall*, dan mAP50 mengalami peningkatan seiring bertambahnya jumlah *epoch*. Pada *epoch* 50, nilai *precision* hanya 0.82 dan mAP50 sebesar 0.75, sedangkan pada *epoch* 200, *precision* meningkat menjadi 0.86 dan mAP50 menjadi 0.81. Ini menunjukkan bahwa semakin lama model dilatih, semakin baik kemampuannya dalam mendeteksi digit secara akurat.
2. Untuk setiap *epoch*, peningkatan *batch size* dari 10 ke 25 juga diikuti dengan peningkatan nilai *precision*, *recall*, dan mAP50, meskipun selisihnya tidak terlalu besar. Hal ini menandakan bahwa penggunaan *batch size* yang lebih besar dapat memberikan kontribusi positif terhadap stabilitas dan performa model.
3. Variasi *learning rate* dari 0.001 hingga 0.01 tidak memberikan perubahan signifikan pada metrik evaluasi, baik pada *epoch* rendah maupun tinggi. Nilai *precision*, *recall*, dan mAP50 relatif konsisten di seluruh variasi *learning rate* yang diuji. Dengan demikian, pengaruh *learning rate* dalam rentang ini terhadap performa model relatif kecil.

Berdasarkan hasil pengujian *training data* dengan tiga pendekatan berbeda, yaitu *learning rate* tetap, *batch size* tetap, dan *epoch* tetap, dapat disimpulkan bahwa performa model YOLOv5 untuk deteksi digit angka pada *water meter* menunjukkan tren yang konsisten. Pada pengujian dengan *learning rate* tetap, variasi *batch size* dan jumlah *epoch* memberikan pengaruh yang lebih nyata terhadap peningkatan nilai *precision*, *recall*, dan mAP50, sedangkan perubahan *learning rate* pada rentang 0.001 hingga 0.01 tidak berdampak signifikan terhadap performa model. Hal serupa juga terlihat pada pengujian dengan *batch size* tetap dan *epoch* tetap, di mana penambahan jumlah *epoch* dan *batch size* secara umum menghasilkan hasil deteksi yang lebih baik. Secara keseluruhan, model mencapai nilai *precision* hingga 0.87, *recall* 0.84, dan mAP50 sebesar 0.81 pada kombinasi parameter optimal, menandakan bahwa model mampu mendeteksi digit angka dengan tingkat akurasi yang tinggi dan stabil di seluruh variasi parameter yang diuji.

4.2 Pengujian Deteksi Digit Angka

Pengujian deteksi digit angka dilakukan untuk menilai kemampuan model dalam mendeteksi dan mengenali angka pada meter air secara otomatis dari citra yang diambil oleh kamera/*webcam*. Pengujian ini bertujuan untuk mengukur tingkat akurasi dan keandalan sistem deteksi angka pada berbagai kondisi nyata, seperti variasi pencahayaan dan posisi meter air. Pengujian ini dilakukan dengan meletakkan kamera *webcam* didepan *watermeter* dengan jarak 15 cm.

4.2.1 Peralatan yang Dibutuhkan

1. *Webcam*/kamera untuk pengambilan citra meter air
2. Komputer/laptop dengan YOLOv5 terpasang
3. *Dataset* pengujian (untuk perbandingan data gambar atau video meter air)

4.2.2 Prosedur Deteksi Digit Angka

1. Menyiapkan *webcam* dan komputer dengan model deteksi YOLOv5.
2. Mengambil gambar atau video meter air sebagai data uji.
3. Menjalankan proses deteksi digit pada data uji.
4. Mencatat hasil deteksi digit angka dan *confidence level* setiap digit.
5. Membandingkan hasil deteksi dengan angka sebenarnya pada meter air.
6. Menghitung tingkat akurasi dan dokumentasi kasus *error (misdetektion)*.

4.2.3 Hasil Deteksi Digit Angka

```
# =====
# KONFIGURASI UTAMA
# =====
MODEL_PATH = r'C:\TA\OCR\yolov5\runs\train\custom_yolo3\weights\best.pt'
JUMLAH_DIGIT = 7          # Jumlah digit angka
POSISI_KOMA = 4           # Koma setelah digit ke-4
MAX_BERTURUT = 5         # Harus sama selama n frame
PREVIEW_LEBAR = 960       # Ukuran lebar tampilan window
```

Gambar 4.1 Konfigurasi utama deteksi angka

Gambar 4.1 merupakan cuplikan kode konfigurasi utama untuk program deteksi digit pada meter air berbasis YOLOv5. Setiap baris kode mendefinisikan

parameter penting yang akan digunakan saat menjalankan proses deteksi angka secara otomatis.

Penjelasan setiap parameter:

- *MODEL_PATH* adalah *path* atau lokasi *file* model YOLO (best.pt) hasil *training* yang akan digunakan untuk proses inferensi. *Path* ini harus disesuaikan dengan lokasi *file* model pada komputer pengguna. Pada pengujian ini model *training* yang digunakan adalah model *train* ke-3, yaitu model terbaik yang menghasilkan nilai *precision* hingga 0.87, *recall* 0.84, dan mAP50 sebesar 0.81.
- *JUMLAH_DIGIT* menunjukkan jumlah digit angka yang akan dideteksi, pada contoh ini sebanyak 7 digit, sesuai format angka pada meter air. Water meter yang digunakan pada pengujian ini berjumlah 7 digit angka.
- *POSISI_KOMA* menentukan posisi penempatan tanda koma pada angka hasil deteksi, di mana koma akan muncul setelah digit ke-4, sehingga hasil deteksi sesuai dengan format pembacaan meter air yang digunakan pada pengujian menggunakan 4 digit sebelum koma dan 3 digit setelah koma.
- *MAX_BERTURUT* mengatur jumlah minimum *frame* berturut-turut di mana hasil deteksi harus konsisten agar dianggap valid, misalnya harus sama selama 5 *frame* agar tidak terjadi kesalahan akibat deteksi yang tidak stabil.
- *PREVIEW_LEBAR* mengatur lebar tampilan *window* aplikasi saat menampilkan hasil deteksi secara visual, misalnya diatur ke 960 piksel.

Konfigurasi ini bertujuan agar program deteksi dapat berjalan sesuai dengan kebutuhan aplikasi dan karakteristik data meter air yang digunakan. Pengguna dapat menyesuaikan parameter-parameter ini untuk mendapatkan hasil deteksi yang optimal dan tampilan visual yang sesuai keinginan.

Gambar 4.2 Pengujian *water meter* 1Gambar 4.3 Pengujian *water meter*

Gambar 4.2 dan Gambar 4.3 merupakan tampilan hasil pengujian pembacaan digit *water meter* langsung melalui *webcam*. Dimana *bouding box* yang dihasilkan sesuai dengan digit yang tersedia pada *water meter*. Jarak 15 cm terhitung ideal untuk pengujian secara langsung menggunakan *webcam*.

Deteksi objek angka menggunakan YOLOv5 secara *default* tidak berurutan dari kiri ke kanan. Sehingga pada program python harus melakukan pengurutan data angka terlebih dahulu, pengurutan digit angka dilakukan dengan membandingkan nilai x (nilai koordinat *pixel* horizontal). Dimana digit angka akan diurutkan dari nilai x terkecil hingga x_{min} terbesar.

Tabel 4.4 Hasil pengujian deteksi angka

Perbandingan Pembacaan						Keterangan
Water Meter No.	Angka di Meteran	Data No.	Angka Hasil Pembacaan	Waktu Deteksi	FPS	
1	0178.102	1	0178.102	109.1	6.53	Valid
		2	0178.102	124.9	8.00	
		3	0178.102	109.9	9.10	
		4	0178.102	108.8	9.19	
		5	0178.102	109.3	9.15	
2	0187.943	1	0187.943	96.2	10.40	Tidak Valid
		2	0187.943	122.8	8.14	
		3	0187.943	120.1	8.32	
		4	0187.943	137.6	7.26	
		5	0187.942	125.0	8.00	
3	0260.145	1	0260.145	157.2	6.36	Valid
		2	0260.145	111.7	8.95	
		3	0260.145	102.1	9.79	
		4	0260.145	103.3	9.68	
		5	0260.145	107.3	9.32	
4	0236.317	1	0236.317	103.9	9.62	Valid
		2	0236.317	94.5	10.58	
		3	0236.317	85.7	11.68	
		4	0236.317	99.5	10.05	
		5	0236.317	111.9	8.84	
5	0000.163	1	0000.163	121.7	8.21	Tidak Valid
		2	0000.163	107.4	9.31	
		3	0000.163	117.2	8.53	
		4	0000.160	121.9	8.21	
		5	0000.160	105.9	9.45	
6	0058.251	1	0058.251	144.6	6.91	Valid
		2	0058.251	92.0	10.86	
		3	0058.251	93.8	10.66	
		4	0058.251	101.6	9.85	
		5	0058.251	99.8	10.02	
7	0177.808	1	0177.808	107.9	9.27	Valid
		2	0177.808	100.4	9.96	
		3	0177.808	91.7	10.91	
		4	0177.808	110.2	9.08	
		5	0177.808	91.5	10.93	
8	0259.777	1	0259.777	140.2	7.13	Valid
		2	0259.777	122.3	8.18	
		3	0259.777	104.0	9.61	
		4	0259.777	98.3	10.17	
		5	0259.777	92.1	98.3	

Perbandingan Pembacaan						Keterangan
Water Meter No.	Angka di Meteran	Data No.	Angka Hasil Pembacaan	Waktu Deteksi	FPS	
9	0058.251	1	0058.251	94.1	10.63	Valid
		2	0058.251	115.4	8.67	
		3	0058.251	92.3	10.83	
		4	0058.251	79.1	12.65	
		5	0058.251	101.9	9.81	
10	0177.806	1	0177.806	113.1	8.84	Valid
		2	0177.806	94.1	10.63	
		3	0177.806	91.6	10.92	
		4	0177.806	97.3	10.28	
		5	0177.806	80.7	12.39	

Berikut adalah analisis dari Tabel 4.4 Hasil pengujian deteksi angka pada 10 meteran air:

1. Meteran 1, 3, 4, 6, 7, 8, 9, dan 10:

Semua data uji pada meteran-meteran ini berhasil terdeteksi dengan benar (“Valid” untuk seluruh baris), menunjukkan sistem deteksi digit angka bekerja dengan sangat baik dan konsisten pada sebagian besar meteran air yang diuji.

2. Meteran 2 dan 5:

Pada dua meteran ini terdapat hasil “Tidak Valid” pada beberapa data uji. Hal ini menandakan bahwa pada kondisi tertentu atau karakteristik visual khusus (misal bentuk digit, pencahayaan, atau kualitas citra), model masih bisa mengalami kesalahan atau inkonsistensi deteksi.

Tabel 4.5 Rangkuman hasil pengujian deteksi angka

Water Meter No.	Data Uji	Data Valid	Akurasi (%)
1	5	5	100
2	5	4	80
3	5	5	100
4	5	5	100
5	5	3	60
6	5	5	100
7	5	5	100
8	5	5	100
9	5	5	100
10	5	5	100
Total	50	47	94

$$(1) \text{ Akurasi (\%)} = \frac{\text{Jumlah data valid}}{\text{Jumlah total data uji}} \times 100\%$$

Jumlah total data uji: $10 \text{ meteran} \times 5 = 50 \text{ data}$

Jumlah data valid: 47 data

Akurasi Total:

$$(1) \text{ Akurasi (\%)} = \frac{47}{50} \times 100\% = 94\%$$

Pengujian deteksi angka pada 10 meteran air menunjukkan performa yang sangat baik, dengan akurasi keseluruhan sebesar 94%. Pada delapan meteran air, sistem selalu berhasil mendeteksi angka dengan benar pada semua data uji, menandakan tingkat keandalan yang tinggi dan stabil pada aplikasi nyata. Kasus tidak valid hanya ditemukan pada dua meteran, sehingga dapat dijadikan acuan untuk analisis dan perbaikan lebih lanjut pada data serupa. Hasil ini membuktikan bahwa sistem deteksi digit angka berbasis YOLOv5 sudah sangat layak untuk diimplementasikan sebagai solusi otomatisasi pembacaan meter air.

4.2.4 Analisa Error

Kesalahan deteksi hanya terjadi pada meteran 2 dan 5, sisanya sempurna. Faktor utama kemungkinan berasal dari karakteristik visual angka tertentu, kualitas citra, posisi kamera, atau kondisi pencahayaan.

Pada proses pembacaan digit angka pada meter air menggunakan metode deteksi berbasis YOLOv5, sering kali terjadi *error* khususnya pada digit terakhir. Hal ini umumnya disebabkan oleh kondisi di mana angka terakhir pada counter meter air sedang berada di antara dua angka, yaitu saat roda angka sedang bergerak dan belum sepenuhnya menunjukkan satu nilai tertentu. Dalam kondisi ini, tampilan visual digit menjadi ambigu atau tidak jelas terbaca, sehingga sistem deteksi mengalami kesulitan dalam memutuskan apakah angka yang terbaca adalah nilai sebelum atau sesudahnya. Akibatnya, sistem bisa saja membaca angka terakhir dengan salah, terutama pada saat perubahan digit yang berlangsung secara perlahan.

Fenomena ini merupakan tantangan tersendiri dalam otomatisasi pembacaan meter air dan menjadi salah satu penyebab utama terjadinya hasil deteksi yang tidak valid meskipun digit-digit sebelumnya telah terbaca dengan benar. Contoh *error* pembacaan digit terakhir dapat dilihat pada Gambar 4.4.



Gambar 4.4 Analisa *error* digit terakhir

4.3 Pengujian Pengiriman Data MQTT

Pengujian pengiriman data MQTT dilakukan untuk memastikan bahwa hasil deteksi angka pada meter air dapat dikirimkan secara *real-time* dan andal dari sistem ke perangkat *monitoring* melalui protokol MQTT. Uji ini meliputi keberhasilan pengiriman, keterlambatan (*delay*), serta kesesuaian data yang diterima di aplikasi *mobile*.

4.3.1 Peralatan yang Dibutuhkan

1. Komputer/laptop sebagai *publisher* MQTT
2. *Smartphone* yang sudah terinstal aplikasi IoT MQTT Panel
3. *Broker* MQTT (bisa lokal atau *cloud*, misal *Mosquitto*, *HiveMQ*)
4. Koneksi internet atau jaringan lokal

4.3.2 Prosedur Pengiriman Data MQTT

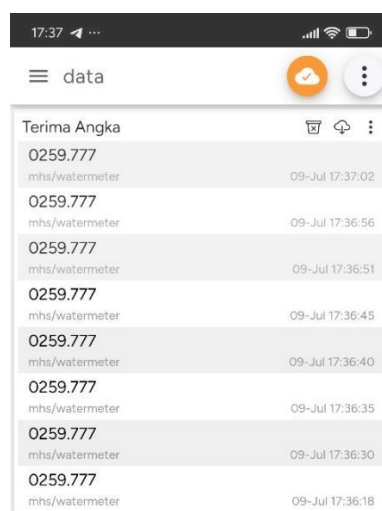
1. Mengatur koneksi antara komputer dan *broker* MQTT.
2. Menyiapkan aplikasi IoT MQTT Panel di *smartphone* dan *subscribe* ke topik yang sesuai.
3. Mengirim data hasil deteksi digit angka melalui MQTT dari komputer.
4. Memantau penerimaan data pada aplikasi *mobile* secara *real-time*.

4.3.3 Hasil Pengiriman Data MQTT

```
MQTT_BROKER = "mqtt.dinamika.ac.id"
MQTT_PORT = 1883
MQTT_USERNAME = "mhs"
MQTT_PASSWORD = "mahasiswa"
MQTT_TOPIC = "mhs/watermeter"
INTERVAL_KIRIM_MQTT = 5 # Detik, minimal waktu antar publish
```

Gambar 4.5 Konfigurasi MQTT

Gambar 4.5 menunjukkan konfigurasi dari pengujian MQTT pada program python, dimana konfigurasi ini sama dengan yang diterapkan pada aplikasi IoT MQTT Panel. Terdapat konfigurasi tambahan yaitu interval dari pengiriman data MQTT agar tidak terlalu sering mengirimkan data ke aplikasi, pada pengujian ini interval yang digunakan adalah 5 detik.



Terima Angka	Timestamp
0259.777	09-Jul 17:37:02
0259.777	09-Jul 17:36:56
0259.777	09-Jul 17:36:51
0259.777	09-Jul 17:36:45
0259.777	09-Jul 17:36:40
0259.777	09-Jul 17:36:35
0259.777	09-Jul 17:36:30
0259.777	09-Jul 17:36:18

Gambar 4.6 Hasil pengujian *monitoring water meter*

Tabel 4.6 Pengujian data MQTT

No.	Kondisi Pengiriman Data				
	Angka Hasil Pembacaan Sistem	Angka Hasil Pembacaan di Smartphone	Status Wifi / Koneksi	Keterangan	Status
1	0259.777	0259.777	Terhubung	Terkirim	Sesuai
2	0259.777	0259.777	Terhubung	Terkirim	Sesuai
3	0259.777	0259.777	Terhubung	Terkirim	Sesuai
4	0259.777	-	Wifi Mati	Tidak Terkirim	Sesuai
5	0259.777	0259.777	Terhubung	Terkirim	Sesuai
6	0259.777	0259.777	Terhubung	Terkirim	Sesuai
7	0236.317	0236.317	Terhubung	Terkirim	Sesuai
8	0236.317	0236.317	Terhubung	Terkirim	Sesuai
9	0236.317	0236.317	Terhubung	Terkirim	Sesuai
10	0236.317	0236.317	Terhubung	Terkirim	Sesuai
11	0236.317	0236.317	Terhubung	Terkirim	Sesuai
12	0236.317	0236.317	Terhubung	Terkirim	Sesuai
13	0187.943	0187.943	Terhubung	Terkirim	Sesuai
14	0187.943	0187.943	Terhubung	Terkirim	Sesuai
15	0187.943	0187.943	Terhubung	Terkirim	Sesuai
16	0187.943	0187.943	Terhubung	Terkirim	Sesuai
17	0187.943	-	Wifi Mati	Tidak Terkirim	Sesuai
18	0187.943	0187.943	Terhubung	Terkirim	Sesuai
19	0187.943	0187.943	Terhubung	Terkirim	Sesuai
20	0187.943	0187.943	Terhubung	Terkirim	Sesuai

Berikut adalah analisis Tabel 4.6 Pengujian Data MQTT. Dari 20 kali pengujian, sebanyak 18 data berhasil dikirim dan diterima (“Terkirim”), sementara 2 data gagal terkirim akibat WiFi mati.

$$(1) \text{ Akurasi Pengiriman (\%)} = \frac{20}{20} \times 100\% = 100\%$$

Semua kegagalan pengiriman terjadi karena masalah jaringan, bukan karena sistem atau aplikasi. Pengujian pengiriman data menggunakan MQTT pada sistem *monitoring* meter air menunjukkan bahwa seluruh data berhasil dikirim dan

diterima di *smartphone* selama koneksi WiFi terhubung, sementara data tidak dapat dikirim saat WiFi mati atau koneksi terputus. Namun, sistem tetap dapat berjalan secara lokal meskipun tidak ada koneksi internet, sehingga proses deteksi dan pembacaan angka di perangkat utama tetap berlangsung tanpa gangguan. Ketika koneksi jaringan kembali normal, data dapat langsung dikirim ke aplikasi *smartphone*, menandakan sistem ini memiliki ketahanan yang baik terhadap gangguan jaringan dan tetap mendukung kebutuhan *monitoring* secara andal di berbagai kondisi.

4.4 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem bertujuan untuk menguji integrasi semua komponen mulai dari akuisisi gambar, deteksi digit, pengiriman data MQTT, hingga *monitoring* melalui aplikasi *mobile*. Pengujian ini memastikan bahwa sistem dapat berjalan secara otomatis, efisien, dan handal dari awal hingga akhir alur kerja.

4.4.1 Peralatan yang Dibutuhkan

1. *Webcam*/kamera
2. Komputer/laptop dengan YOLOv5 dan MQTT *publisher*
3. *Broker* MQTT (*local/cloud*)
4. *Smartphone* dengan aplikasi IoT MQTT Panel
5. Jaringan internet atau Wi-Fi

4.4.2 Prosedur Pengujian Keseluruhan Sistem

1. Mengaktifkan seluruh perangkat dan aplikasi pendukung.
2. Melakukan akuisisi gambar meter air secara *real-time* dengan *webcam*.
3. Melakukan deteksi digit angka menggunakan model YOLOv5.
4. Mengirim hasil deteksi melalui MQTT ke *broker*.
5. Memantau hasil deteksi pada aplikasi *mobile*.
6. Mencatat performa setiap tahap, seperti waktu deteksi, waktu pengiriman, serta akurasi pembacaan.
7. Melakukan pengujian dalam beberapa kondisi berbeda (cahaya, posisi kamera, jaringan) untuk mengukur keandalan sistem secara menyeluruh.

4.4.3 Hasil Pengujian Keseluruhan Sistem

Tabel 4.7 Hasil pengujian keseluruhan sistem

Uji Coba No.	Angka Water Meter	Angka Hasil Pembacaan Sistem	Angka Hasil Pembacaan di Smartphone	Status Wifi / Koneksi	Keterangan
1	0187.943	187.943	187.943	Terhubung	Terkirim / Sesuai
2	0187.943	187.942	-	Terhubung	Tidak valid / Tidak sesuai
3	0177.806	0177.806	0177.806	Terhubung	Terkirim / Sesuai
4	0177.806	0177.806	0177.806	Terhubung	Terkirim / Sesuai
5	0236.317	0236.317	0236.317	Terhubung	Terkirim / Sesuai
6	0236.317	0236.317	0236.317	Terhubung	Terkirim / Sesuai
7	0000.163	0000.163	0000.163	Terhubung	Terkirim / Sesuai
8	0000.163	0000.163	0000.163	Terhubung	Terkirim / Sesuai
9	0058.251	0058.251	-	Tidak Terhubung	Tidak Terkirim / Sesuai
10	0058.251	0058.251	0058.251	Terhubung	Terkirim / Sesuai
11	0058.251	0058.251	0058.251	Terhubung	Terkirim / Sesuai
12	0178.102	0178.102	0178.102	Terhubung	Terkirim / Sesuai
13	0178.102	0178.102	0178.102	Terhubung	Terkirim / Sesuai
14	0178.102	0178.102	0178.102	Terhubung	Terkirim / Sesuai
15	0000.163	0000.162	-	Terhubung	Tidak Valid / Tidak Sesuai
16	0260.145	0260.145	0260.145	Terhubung	Terkirim / Sesuai
17	0260.145	0260.145	0260.145	Terhubung	Terkirim / Sesuai
18	0260.145	0260.145	0260.145	Terhubung	Terkirim / Sesuai
19	0236.317	0236.317	0236.317	Terhubung	Terkirim / Sesuai
20	0236.317	0236.317	0236.317	Terhubung	Terkirim / Sesuai

Berdasarkan Tabel 4.7, dapat dilihat bahwa secara umum sistem pembacaan angka *water meter*, deteksi oleh sistem, dan pengiriman ke aplikasi *smartphone* berjalan dengan sangat baik. Sebagian besar data uji menunjukkan hasil pembacaan yang sesuai antara angka di *water meter*, angka hasil deteksi oleh sistem, dan angka yang diterima di aplikasi *smartphone*. Hal ini ditunjukkan dengan status “Terhubung” pada koneksi WiFi dan keterangan “Terkirim” pada kolom terakhir.

Pada beberapa kasus, seperti uji coba ke-2 dan ke-15, ditemukan hasil “Tidak valid”, di mana angka hasil pembacaan sistem tidak sama persis dengan angka sebenarnya atau tidak terkirim ke *smartphone*. Pada uji coba ke-9, data tidak dapat dikirim karena status jaringan “Tidak Terhubung”, sehingga pada *smartphone* data

tidak diterima (keterangan “Tidak Terkirim”). Hal ini menunjukkan bahwa pengiriman data melalui MQTT sangat bergantung pada kondisi koneksi jaringan.

$$(1) \text{ Akurasi (\%)} = \frac{18}{20} \times 100\% = 90\%$$

Secara keseluruhan menghasilkan akurasi sebesar 85%, sistem menunjukkan reliabilitas dan akurasi yang tinggi, dengan tingkat keberhasilan pengiriman dan pembacaan yang konsisten pada mayoritas pengujian. Sistem juga memiliki toleransi terhadap gangguan jaringan, karena proses deteksi angka tetap berjalan di sisi sistem meski data tidak selalu dapat dikirim saat WiFi terputus. Hasil ini menegaskan bahwa sistem otomatisasi pembacaan meter air berbasis YOLOv5 dan MQTT yang dikembangkan layak untuk diimplementasikan dan cukup andal untuk penggunaan di lingkungan nyata.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

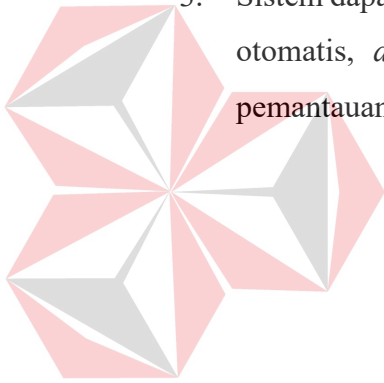
Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem otomatisasi pembacaan meter air berbasis *computer vision* menggunakan model YOLOv5 berhasil dikembangkan dan diimplementasikan dengan baik. Sistem ini mampu mendeteksi dan membaca digit angka pada meter air secara otomatis dari citra yang diambil menggunakan *webcam*, dengan tingkat akurasi deteksi yang tinggi pada berbagai kondisi pengujian.
2. Hasil *training* dan validasi menunjukkan bahwa parameter *batch size* 20 dan jumlah *epoch* 150 memberikan pengaruh paling signifikan terhadap peningkatan akurasi model, sedangkan variasi *learning rate* pada rentang 0.001 hingga 0.01 tidak memberikan dampak yang berarti. Model terbaik yang dihasilkan mencapai nilai *precision* hingga 0.87, *recall* 0.84, dan mAP50 sebesar 0.81, menandakan model optimal untuk aplikasi deteksi angka pada meter air.
3. Pada pengujian deteksi angka di 10 unit meter air dengan berbagai data uji, sistem berhasil memperoleh akurasi rata-rata sebesar 94%, dengan sebagian besar hasil deteksi sesuai dengan angka yang sebenarnya. Kasus kegagalan deteksi umumnya terjadi pada digit terakhir saat angka pada meter air sedang bergerak di antara dua nilai, yang menyebabkan ambiguitas pembacaan.
4. Sistem pengiriman data menggunakan protokol MQTT dari perangkat komputer ke aplikasi IoT MQTT Panel di *smartphone* berjalan dengan baik selama koneksi jaringan stabil. Dari hasil pengujian, akurasi pengiriman data mencapai 100%, dan semua *error* pengiriman hanya terjadi saat koneksi WiFi terputus. Sistem tetap mampu bekerja secara lokal dan akan mengirimkan data begitu koneksi kembali tersedia.
5. Pengujian integrasi seluruh sistem dari akuisisi gambar, deteksi angka, hingga pengiriman data ke aplikasi monitoring menunjukkan tingkat keberhasilan *end-to-end* sebesar 90%. Sistem terbukti dengan hasil yang baik, *real-time*, dan

dapat digunakan sebagai solusi *monitoring* penggunaan air secara otomatis di lingkungan nyata.

5.2 Saran

1. Disarankan untuk menambah jumlah dan variasi *dataset*, khususnya pada kondisi pencahayaan ekstrim, posisi miring, atau tipe meter air yang berbeda, agar model menjadi lebih *robust* dalam menghadapi berbagai situasi di lapangan.
2. Perlu dilakukan pengembangan metode *filtering* atau verifikasi tambahan khusus untuk mengatasi kasus ambiguitas pada digit terakhir yang sedang bergerak, agar tingkat *error* pembacaan dapat diminimalkan.
3. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur peringatan otomatis, *dashboard* analitik historis, atau integrasi dengan *cloud* untuk pemantauan multi-lokasi secara terpusat.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Brad Dwyer, J. G. (2023, Maret 16). *Getting Started with Roboflow*. Retrieved from roboflow.com: blog.roboflow.com/getting-started-with-roboflow/
- Chen, C. (2020). A YOLO-Based Meter Reading Recognition Algorithm for Automatic Meter Reading. *Sensors*, 20(4).
- Darwis, A. S. (2021). Monitoring dan Kontrol Otomatis Berbasis IoT dengan MQTT pada Aplikasi Mobile. *Jurnal RESTI*, 5(2).
- Lutz, M. (2021). *Programming Python (5th Edition)*. O'Reilly Media.
- Madakam, S. R. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 3(5).
- OpenCV. (2021, Februari 14). *OpenCV-Python Is Now An Official OpenCV Project*. Retrieved from OpenCV: opencv.org/blog/opencv-python-is-now-an-official-opencv-project/
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications (2nd Edition)*. Springer.
- Ultralytics. (2023, November 12). *YOLOv5 Documentation*. Retrieved from Ultralytics: docs.ultralytics.com
- Yulianto, M. &. (2018). Rancang Bangun Sistem Monitoring Konsumsi Air Berbasis IoT. *Jurnal Teknik ITS*, A137-A141.