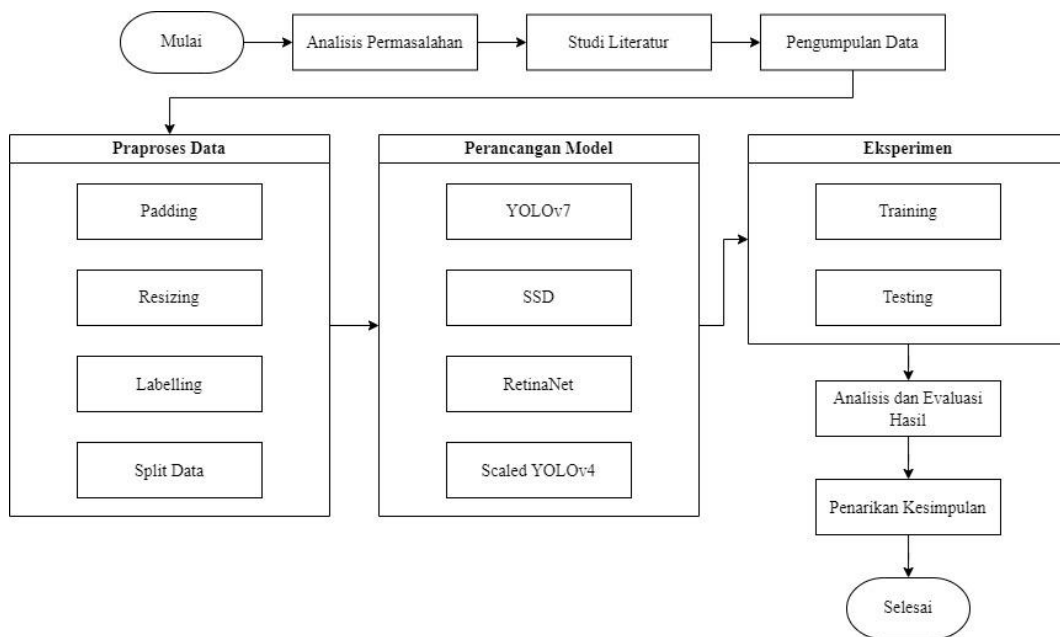


BAB 3 METODOLOGI PENELITIAN

3.1 Desain Penelitian

Pada penelitian ini terdapat beberapa tahapan pengerjaan. Tahapan-tahapan ini dimulai dari merumuskan masalah, studi literatur, pengumpulan data, implementasi, evaluasi, dan tahapan terakhir yaitu penarikan kesimpulan.



Gambar 3.1 Desain Penelitian

Adapun penjelasan dari desain penelitian pada Gambar 3.1 adalah sebagai berikut.

3.1.1 Analisis Permasalahan

Analisis permasalahan merupakan proses pencarian rumusan masalah yang nantinya digunakan sebagai acuan bagaimana penelitian deteksi objek kerusakan jalan ini dilakukan dan bagaimana implementasi metode YOLOv7 terhadap deteksi objek kerusakan jalan ini, dimana nantinya jawaban dari pertanyaan-pertanyaan rumusan masalah inilah yang akan menjadi hasil penelitian.

3.1.2 Studi Literatur

Studi literatur merupakan serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca dan mencatat, serta mengolah bahan penelitian. Studi literatur yang dikumpulkan dan dipelajari adalah mengenai topik *computer vision, object detection, machine learning, artificial neural network, deep learning, convolutional neural network, you only look once (YOLO), Single Shot Detector (SSD), RetinaNet*.

3.1.3 Pengumpulan Data

Pengumpulan data ini akan digunakan sebagai dataset dalam penelitian deteksi objek kerusakan pada permukaan jalan, kemudian dataset ini akan diproses melalui tahap *pre-processing*, lalu dibagi menjadi dua bagian yaitu data *train-set* dan data *test-set* untuk menguji model yang sudah dibuat menggunakan metode *random sampling*.

Data yang digunakan pada penelitian ini merupakan data yang diambil dari Dinas Bina Marga dan Penataan Ruang (DBMPR) Provinsi Jawa Barat. Data ini berisikan gambar-gambar kerusakan pada permukaan jalan provinsi yang menjadi kewenangan DBMPR Provinsi Jawa Barat yang dimana kerusakan yang ada dalam gambar tersebut dikategorikan menjadi 2 jenis kerusakan yaitu lubang dan retak.

3.1.4 Praproses Data

Untuk tahap praproses data yang akan dilakukan pada penelitian ini yaitu mulai dari *padding* gambar agar ukuran panjang dan lebar dari gambar menjadi sama, kemudian dilanjutkan dengan *resizing* agar ukuran gambar tidak terlalu besar dan tidak terlalu kecil, lalu dilanjutkan dengan *labelling* objek kerusakan yang ada pada gambar dengan cara membuat kotak pembatas atau biasa disebut dengan *bounding box* pada gambar dan juga memberikan kategori atau kelas terhadap kotak pembatas tersebut, terakhir dilakukan proses *split* data untuk membagi data menjadi data *train* yang digunakan pada saat pelatihan model dan data *test* yang digunakan pada saat evaluasi model.

3.1.5 Perancangan Model

Pada tahap ini akan dilakukan perancangan model yang digunakan pada penelitian ini. Model ini akan dirancang berdasarkan studi literatur yang telah dipelajari sebelumnya dari berbagai sumber pustaka. Dalam tahapan perancangan model ini juga akan dilakukan praproses pada data yang telah dikumpulkan sebelumnya. Kemudian dilanjutkan dengan pelatihan model menggunakan data yang telah diproses untuk mendeteksi objek kerusakan pada permukaan jalan.

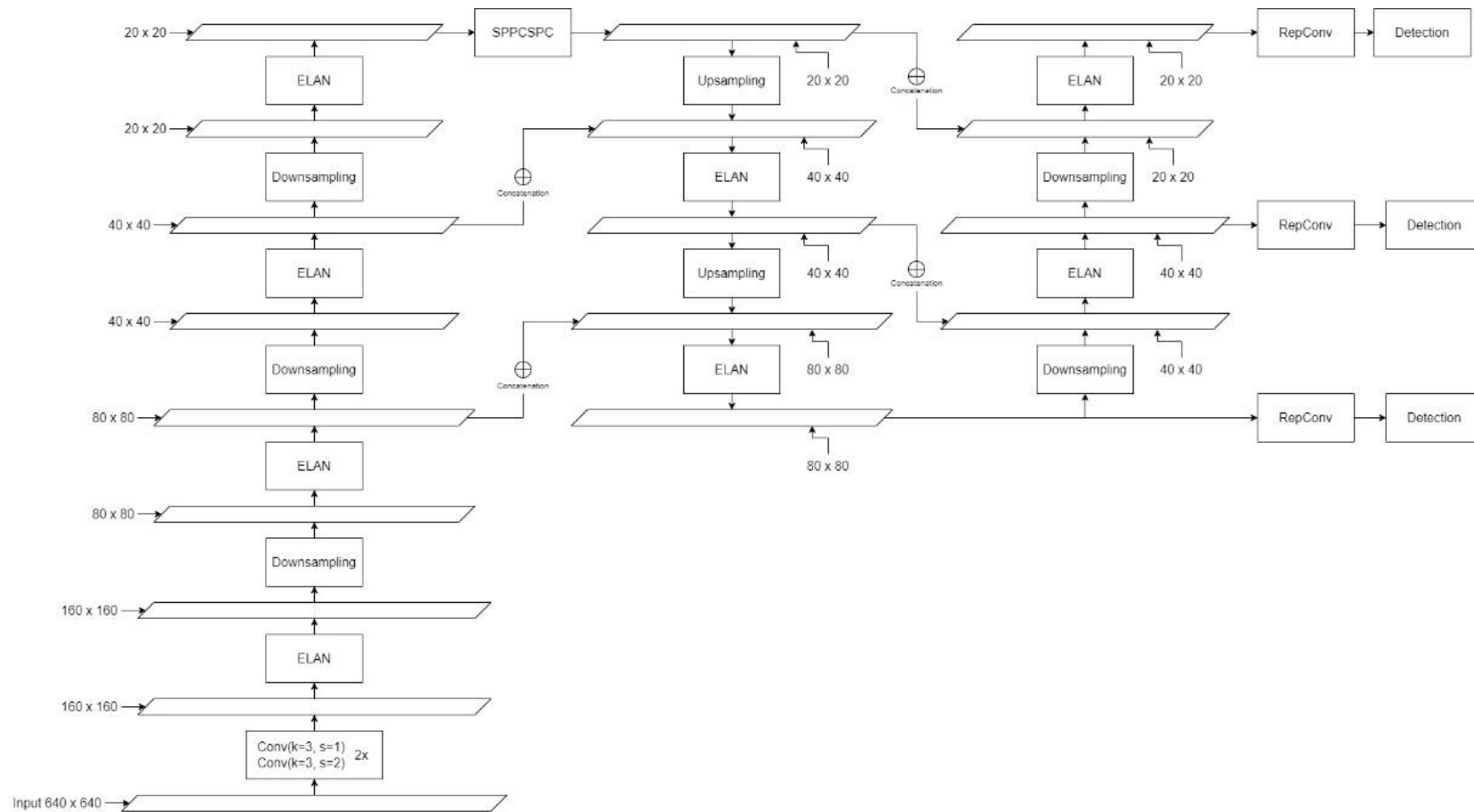
Dalam penelitian ini, akan dibangun model pendeteksi kerusakan jalan menggunakan metode YOLOv7 dan juga akan dibangun beberapa model pendeteksi objek lainnya menggunakan metode terdahulu seperti *Single Shot Detector* (SSD), RetinaNet, dan juga Scaled YOLOv4.

a) YOLOv7

Model YOLOv7 yang akan digunakan dalam perancangan dan pelatihan pada penelitian ini menggunakan model yang telah dirancang sebelumnya menggunakan *framework* PyTorch oleh pencipta aslinya yaitu Chien-Yao Wang yang dapat diakses pada <https://github.com/WongKinYiu/yolov7>. Pelatihan model YOLOv7 ini digunakan teknik *transfer learning* dengan cara *fine tuning* dari bobot inisiasi dari *pretrained weight* YOLOv7 “yolov7_training.pt” yang telah tersedia pada *repository*. Arsitektur YOLOv7 yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Arsitektur YOLOv7 yang digunakan pada penelitian

Lapisan	Deskripsi
<i>Input</i>	640x640 <i>pixel</i>
<i>Backbone</i>	Tumpukan blok komputasi ELAN dan <i>Convolution</i>
<i>Neck</i>	SPPCSPC + PAN + RepConv
<i>Head</i>	YOLO <i>Head</i> + <i>Implicit Knowledge</i>



Gambar 3.2 Arsitektur YOLOv7 yang digunakan pada penelitian

Ade Prianto, 2023
**PERBANDINGAN KINERJA METODE
 YOLO V7, SSD, RETINANET, DAN SCALED YOLO V4
 UNTUK DETEKSI OBJEK KERUSAKAN PADA PERMUKAAN JALAN**
 Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

b) *Single Shot Detector (SSD)*

Model SSD yang akan digunakan dalam perancangan dan pelatihan pada penelitian ini menggunakan model yang telah tersedia pada *framework* Tensorflow Object Detection API yang dapat diakses pada *repository* GitHub https://github.com/tensorflow/models/tree/master/research/object_detection.

Arsitektur SSD yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2 Arsitektur SSD yang digunakan pada penelitian

Lapisan	Deskripsi
<i>Input</i>	300x300 <i>pixel</i>
<i>Backbone</i>	MobileNetV2
<i>Head</i>	SSD <i>Head</i>

c) *RetinaNet*

Model RetinaNet yang akan digunakan dalam perancangan dan pelatihan pada penelitian ini menggunakan model yang telah tersedia pada *framework* Tensorflow Object Detection API yang dapat diakses pada *repository* GitHub https://github.com/tensorflow/models/tree/master/research/object_detection.

Arsitektur RetinaNet yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Arsitektur RetinaNet yang digunakan pada penelitian

Lapisan	Deskripsi
<i>Input</i>	640x640 <i>pixel</i>
<i>Backbone</i>	ResNet50
<i>Neck</i>	FPN
<i>Head</i>	RetinaNet <i>Head</i>

d) *Scaled YOLOv4*

Model Scaled YOLOv4 yang akan digunakan dalam perancangan dan pelatihan pada penelitian ini menggunakan model yang telah dirancang

sebelumnya menggunakan framework PyTorch oleh pencipta aslinya yaitu Chien-Yao Wang yang dapat diakses pada <https://github.com/WongKinYiu/ScaledYOLOv4>. Arsitektur Scaled YOLOv4 yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4 Arsitektur Scaled YOLOv4 yang digunakan pada penelitian

Lapisan	Deskripsi
<i>Input</i>	640x640 <i>pixel</i>
<i>Backbone</i>	CSPDarknet-53
<i>Neck</i>	SPPCSPC + PAN
<i>Head</i>	YOLO <i>Head</i>

3.1.6 Eksperimen

Pada tahap eksperimen ini, akan dilakukan pelatihan model YOLOv7, SSD, RetinaNet, dan Scaled YOLOv4 dengan beberapa skenario penggunaan *hyperparameter* yang berbeda-beda. Hasil pelatihan model-model ini nantinya akan dievaluasi kinerjanya menggunakan metrik evaluasi *average precision* (AP), *precision*, *recall*, *F1-Score*, dan waktu komputasi yang dibutuhkan.

3.1.7 Analisis dan Evaluasi Hasil

Pada tahap ini dilakukan analisis dan evaluasi terhadap hasil dari eksperimen yang telah dilakukan sebelumnya. Model deteksi objek kerusakan pada permukaan jalan menggunakan metode YOLOv7 akan dievaluasi dan dibandingkan kinerjanya dengan model SSD, RetinaNet, dan Scaled YOLOv4.

Untuk mengevaluasi hasil deteksi pada permasalahan penelitian ini, digunakan suatu pengukuran yang dinamakan *average presicion* (AP). Selain AP, pada penelitian ini juga kecepatan deteksi setiap model akan diukur dan dievaluasi dalam bentuk *millisecond*. Juga model-model ini akan dinilai dari segi *precision*, *recall*, dan *F1-score* nya dengan persamaan sebagai berikut:

Persamaan *precision*

$$Precision = \frac{TP}{TP + FP} \quad (3.1)$$

Persamaan *recall*

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

Persamaan *F1-score*

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.3)$$

Persamaan *average precision*

$$AP = \sum_{i=0}^n [Recall_{i+1} - Recall_i] * P_{inter}(Recall_{i+1}) \quad (3.4)$$

3.1.8 Penarikan Kesimpulan

Penarikan kesimpulan diambil berdasarkan hasil dari analisis pada tahap evaluasi. Kesimpulan ini dibuat bertujuan untuk dapat melihat kinerja model yang telah dibuat. Penarikan kesimpulan juga mengacu pada rumusan masalah yang dibuat tentang bagaimana penelitian deteksi objek kerusakan jalan menggunakan algoritma YOLOv7 ini dilakukan, dan bagaimana performanya.

3.2 Lingkungan Komputasi

Lingkungan komputasi pada penelitian ini adalah seperangkat komputer dan beberapa perangkat lunak pendukung. Pada penelitian ini, spesifikasi perangkat keras (*hardware*) komputer sebagai berikut:

- a) Perangkat keras yang digunakan untuk pengembangan (Laptop)
 - CPU Intel® Core™ i7-8750H 6 Cores
 - RAM 16 GB
 - HDD 1 TB
 - SSD 512 GB
 - GPU NVIDIA GeForce GTX 1050 Ti

b) Perangkat keras yang digunakan untuk pelatihan dan pengujian model deteksi objek kerusakan (Google Colaboratory)

- CPU Intel(R) Xeon(R) CPU @ 2.20GHz
- RAM 16 GB
- GPU NVIDIA Tesla V100 dengan VRAM 16 GB

Adapun perangkat lunak (*software*) yang digunakan dalam penelitian ini yaitu:

- Sistem operasi Microsoft Windows 10 Home 64-bit
- Jupyter Notebook
- Python 3.7
- Google Colab
- PyTorch
- Tensorflow