

Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur *Xception*

Ridol Liusman Gaho¹, Irsan Taufik Ali², Esa Prakasa³

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Riau, Kampus Bina Widya, Km. 12,5 Simpang Baru, Pekanbaru, Riau, Indonesia

³Badan Riset dan Inovasi Nasional, Gedung B.J. Habibie, Jl. M.H. Thamrin No. 8, Jakarta Pusat 10340
E-mail: ridol.liusman5927@student.unri.ac.id¹, irsan.ali@lecturer.unri.ac.id², esa.prakasa@lipi.go.id³

Abstrack – Highways are the main infrastructure for land transportation. The better the condition of a highway, the better the speed and safety for drivers. One of the main causes of accidents on highways is due to the road conditions being unsuitable for use because of damage. Therefore, monitoring and maintaining the surface condition of roads is very important. The quality check of highways is generally done manually, a method that requires significant time and effort. The vast number of roads and the manual checks that consume a lot of time and money become obstacles in road maintenance. Therefore, the system "Highway Surface Quality Classification Using CNN Method Based on Xception Architecture" was developed as an alternative to perform surface quality checks on highways. This method uses deep learning CNN with Xception transfer learning architecture. Xception was chosen because it has a complex yet efficient architecture in terms of time usage and high accuracy for image classification, producing accurate models with short training times. Furthermore, several previous studies have shown that Xception outperforms several other architectures. The use of deep learning in classifying highway surface damage is expected to speed up and simplify the process of monitoring road surface conditions. The model is created using a dataset with 4 classes based on the level of damage released by the Ministry of Public Works and Public Housing (PUPR). The highest test results showed a model accuracy of 90.11% and 90% for validation.

Keywords - Road, Damaged, Xception.

Intisari – Jalan raya merupakan prasarana utama untuk transportasi darat, semakin baik kondisi suatu jalan raya semakin baik pula kecepatan dan keselamatan pengendara yang melintas. Salah satu penyebab utama dari kecelakaan yang terjadi di jalan raya adalah karena kondisi jalan yang tidak layak digunakan di karenakan kondisi jalan yang rusak. Oleh karena hal tersebut pengawasan dan perawatan kondisi permukaan jalan sangat penting dilakukan. Pengecekan kualitas jalan raya umumnya dilakukan secara manual, cara ini memakan waktu dan tenaga yang cukup besar. Banyaknya jumlah jalan serta pengecekan manual yang memakan biaya dan waktu yang banyak menjadi penghambat dalam perawatan jalan. Oleh karena itu, dikembangkan sistem "Klasifikasi Kualitas Perukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur *Xception*" sebagai salah satu alternatif untuk melakukan pengecekan kualitas permukaan jalan raya. Metode ini menggunakan *deep learning CNN* dengan arsitektur *transfer learning Xception*, pemilihan *Xception* dipilih karena mempunyai arsitektur yang kompleks namun efisien dalam penggunaan waktunya dan memiliki akurasi yang tinggi untuk melakukan klasifikasi gambar, menghasilkan model akurat dengan waktu pelatihan singkat. Selain itu beberapa penelitian yang telah dilakukan sebelumnya juga menunjukkan bahwa *Xception* mengungguli beberapa arsitektur lainnya. Penggunaan *deep learning* dalam klasifikasi kerusakan permukaan jalan raya di harapkan dapat mempercepat dan mempermudah proses pengawasan kondisi permukaan jalan. Model dibuat menggunakan dataset dengan pembagian 4 kelas berdasarkan pada tingkat kerusakan yang rilis oleh Kementrian PUPR. Hasil pengujian tertinggi menunjukkan akurasi model 90,11% dan 90% untuk pengujian.

Kata Kunci – Jalan, Kerusakan, *Xception*.

I. PENDAHULUAN

Jalan raya adalah lintasan untuk melewati lalu lintas dari satu tempat ke tempat lain, baik berupa jalur yang diperkeras maupun tidak, serta dapat dilewati oleh benda dan makhluk [1]. Transportasi merupakan sarana utama dalam pergerakan manusia, dengan transportasi darat menjadi yang paling sering digunakan, baik itu kendaraan pribadi maupun umum. Kecepatan dan keselamatan transportasi sangat dipengaruhi oleh kondisi jalan raya. Kondisi jalan yang baik tidak hanya meningkatkan kecepatan, tetapi juga keselamatan pengendara. Oleh karena itu, inspeksi kelayakan jalan sangat penting, terutama kualitas permukaannya. Selain faktor keamanan dan kenyamanan pengguna jalan, kerusakan jalan juga dapat mempengaruhi pertumbuhan ekonomi suatu daerah. Kerusakan jalan dapat menghambat transportasi barang dan jasa, memperlambat mobilitas, dan mengurangi efisiensi logistik [2].

Menurut data yang dirilis oleh Badan Pusat Statistik, Negara Indonesia memiliki jalan baik dengan presentase 42,59% sedangkan sisanya dalam kondisi sedang dan rusak [3]. Untuk lebih detailnya dapat dilihat pada Tabel I di bawah ini.

TABEL I
KONDISI JALAN DI INDONESIA TAHUN 2021 (KM)

Kondisi Jalan	Tingkat Kewenangan			Total	Presentase(%)
	Nasional	Provinsi	Kabupaten		
Baik	16.790	28.996	186.858	232.644	42,59%
Sedang	26.378	12.840	99.956	139.174	25,48%
Rusak	2.646	6.330	78.478	87.454	16,01%
Rusak Berat	1.203	6.385	79.256	86.844	15,90%
Total	47.017	54.551	444.548	546.116	

Menurut jurnal ”Survey Pemeliharaan Jalan” yang di terbitkan oleh kementerian PUPR tingkat kerusakan permukaan perkerasan pada permukaan beraspal dapat dilihat pada Tabel II dibawah ini [4].

TABEL II
TINGKAT KERUSAKAN PERMUKAAN PERKERASAN

Kondisi Jalan	Keterangan
Baik	Permukaan perkerasan yang benar-benar rata, tidak ada gelombang, dan tidak ada kerusakan permukaan
Sedang	Permukaan perkerasan sudah mulai bergelombang, dan sudah ada sedikit kerusakan permukaan dan penambalan
Rusak Ringan	Jalan dengan permukaan perkerasan bergelombang yang sudah mulai mengganggu kenyamanan berkendara
Rusak Berat	Jalan dengan kerusakan permukaan berupa lubang-lubang yang disertai dengan kerusakan pondasi dan lapis-lapis lain dibawahnya seperti lobang-lobang yang dalam, ambles, sungkur, dsb yang cukup besar.

Metode manual untuk pengecekan jalan memakan waktu dan tenaga. Metode manual dilakukan dengan cara menyusuri jalan, mengambil foto kerusakan menggunakan kamera, mengukur luas area yang rusak, menentukan tingkat kerusakan berdasarkan jenis kerusakan jalan, kemudian menghitungnya dan mencatatnya dalam bentuk laporan. Untuk mengatasi ini, machine learning dapat digunakan untuk klasifikasi kualitas permukaan jalan, mengurangi biaya dan waktu [5]. Ekstraksi tekstur dengan CNN (Convolutional Neural Network) adalah metode yang efektif untuk mengenali permukaan jalan yang mulus, retak, atau berlubang. CNN,

yang meniru sistem pengenalan citra pada otak manusia, terbukti signifikan dalam pengenalan citra[6].

Setelah ekstraksi tekstur, data diolah menggunakan machine learning untuk mengklasifikasikan gambar jalan ke dalam kategori kelayakan. Python sering digunakan dalam pembuatan model machine learning karena kemudahannya dan dukungan library seperti TensorFlow.

Jumlah data yang banyak biasanya menghasilkan model yang lebih baik, namun transfer learning dapat digunakan jika data terbatas. Transfer learning menggunakan model yang telah dilatih sebelumnya (pre-trained), seperti Xception, yang memiliki keunggulan dalam efisiensi dan akurasi.

Berdasarkan dari penguraian diatas, penggunaan *machine learning* dapat dijadikan alternatif dalam melakukan klasifikasi kerusakan jalan raya. Hal ini bertujuan untuk meningkatkan efektifitas dan efisiensi sehingga mengurangi waktu dan biaya yang diperlukan dalam proses pengecekan jalan. Penelitian ini bertujuan untuk mengembangkan model *machine learning* yang dapat mengklasifikasikan kualitas permukaan jalan raya dengan kelas baik, sedang, rusak ringan, dan rusak berat menggunakan arsitektur *Xception*.

II. SIGNIFIKANSI STUDI

Pada signifikansi studi ini akan dijelaskan Studi Literatur terkait berupa Penelitian Terdahulu, *Dataset*, *Pre-Processing Data*, *Transfer Learning*, Metode Penelitian, dan Simulasi Pelatihan.

A. Studi Literatur

Penelitian pertama dilakukan oleh Tim Puguh B. Prakoso, Utami Sylvia Lestari, dan Yuslena Sari melakukan penelitian pada tahun 2019 berjudul "Deteksi Keretakan Permukaan Perkerasan Lentur Jalan Raya (Studi Kasus: Tanah Lunak Di Banjarmasin)". Penelitian ini bertujuan menciptakan sistem cerdas untuk mendeteksi kecacatan jalan di Kota Banjarmasin dengan dasar tanah gambut. Dataset dikumpulkan menggunakan kamera Canon 550D dengan jarak 1 meter. Pre-processing dilakukan melalui segmentasi, thresholding, dan morfologi. Ekstraksi fitur dilakukan menggunakan Matlab, kemudian diklasifikasikan dengan metode Neural Network. Model yang dihasilkan memiliki akurasi 60% serta recall dan precision 50% [7].

Penelitian dengan judul "Identifikasi Kerusakan Jalan Menggunakan Metode Deep Learning (DL) Model Convolutional Neural Networks (CNN)" oleh Yoga Triardhana, Bandi Sasmito, dan Firman Hadi pada tahun 2021 menggunakan CNN dan YOLO untuk identifikasi kerusakan jalan di Semarang. Deteksi otomatis ini lebih efisien dibanding pemantauan manual, dengan akurasi keseluruhan 85.34% dan kappa accuracy 82.36% [5].

Nurfiah, Yan Rianto, dan Dwiza Riana melakukan penelitian pada tahun 2021 dengan judul "Identifikasi Tingkat Kerusakan Jalan Raya Menggunakan Thresholding Dan K-Means". Penelitian ini menggunakan Thresholding dan K-Means untuk mengidentifikasi kerusakan jalan raya, terutama lubang, dengan menghitung diameter lubang jalannya pada citra grayscale hasil pre-processing. Meskipun semua gambar diinput dapat disegmentasikan, hanya 21 dari 27 sampel yang berhasil disegmentasikan dengan baik [8].

Pada tahun 2022, tim Arif Riyandi, Tony Widodo, dan Shofwatul Uyun melakukan penelitian berjudul "Klasifikasi Pada Citra Jalan Rusak Menggunakan Metode Convolutional Neural Network". Penelitian ini bertujuan mencari metode terbaik dalam pre-processing citra menggunakan grayscale, filter mean, filter media, dan filter balance, diikuti oleh algoritma CNN untuk klasifikasi jalan rusak. Dataset yang digunakan berasal dari Kaggle dengan kata kunci "road damage dataset". Metode CNN dibandingkan dengan berbagai pre-processing untuk menentukan yang terbaik. Model terbaik menggunakan pre-processing grayscale, mencapai akurasi training 88% dan akurasi validation 99% [9].

Penelitian terakhir oleh Alvin Eka Putra, Mohammad Farid Naufal, dan Vincentius Riandaru Prasetyo pada tahun 2023 berjudul “Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning” membandingkan akurasi pre-trained model Xception, DenseNet201, MobileNetV2, VGG16, VGG19, dan ResNet50 untuk klasifikasi 10 jenis rempah. Xception memiliki akurasi terbaik 97%, sementara MobileNetV2 menunjukkan waktu komputasi terbaik 28.11 detik [10].

Rangkuman dari penelitian terdahulu dapat dilihat pada Tabel III berikut :

TABEL III
PENELITIAN TERDAHULU

No	Judul	Tahun, penulis	Metode	Hasil/kesimpulan
1	Deteksi Keretakan Permukaan Perkerasan Lentur Jalan Raya (Studi Kasus: Tanah Lunak di Banjarmasin)	2019, Puguh B. Prakoso , Utami Sylvia Lestari, Yuslena Sari	Neural Network	Penelitian ini menghasilkan sebuah model dengan hasil dari klasifikasi untuk deteksi jalan adalah 60% untuk hasil akurasi dan 50% untuk recall dan precision.
2	Identifikasi Kerusakan Jalan Menggunakan Metode Deep Learning (DL) Model Convolutional Neural Networks (CNN)	2021, Yoga Triardhana, Bandi Sasmito, Firman Hadi	CNN	Model CNN yang dihasilkan bermanfaat untuk identifikasi kerusakan jalan dari video maupun gambar, mampu mengenali kelas kerusakan seperti D00 (retak memanjang), D01 (sambungan konstruksi memanjang), D10 (retak melintang), D11 (sambungan konstruksi melintang), D20 (retak tidak beraturan), D40 (lubang dan benjolan), D43 (zebra cross pudar), dan D44 (marka jalan pudar). Hasil titik persebaran kerusakan jalan dibandingkan dengan validasi manual menggunakan GPS smartphone memiliki nilai RMSE 7,82 meter. Model ini mencapai nilai overall accuracy 85,34% dan kappa accuracy 82,36%.
3	Identifikasi Tingkat Kerusakan Jalan Raya Menggunakan Thresholding Dan K-Means	2021, Nurfiyah , Yan Rianto , Dwiza Riana	Thresholding Dan K-Means	Algoritma Thresholding dan K-means, dengan region 1, dapat diterapkan untuk melakukan segmentasi citra jalan yang mengalami lubang. Setelah berhasil disegmentasikan, citra dapat diukur diameternya. Namun, dari 27 citra yang digunakan dalam penelitian ini, hanya 21 yang berhasil disegmentasikan dengan baik. Dalam konteks identifikasi tingkat kerusakan pada citra jalan raya, algoritma Thresholding dan K-Means berguna jika segmentasi citra dapat dilakukan secara efektif.
4	Klasifikasi Pada Citra Jalan Rusak Menggunakan Metode Convolutional Neural Network	2022, Arif Riyandi, Tony Widodo, Shofwatul Uyun	CNN	Optimizer terbaik adalah "adam", dengan learning rate optimal sebesar 0.001 dan loss terbaik adalah "Categorical Crossentropy". Epoch terbaik adalah 250. Hasil pre-processing terbaik menggunakan metode grayscale, dengan akurasi training mencapai 88% dan akurasi validation mencapai 99%.
5	Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning	2023, Alvin Eka Putra, Mohammad Farid Naufal, dan Vincentius Riandaru Prasetyo	CNN	Akurasi terbaik diperoleh oleh Xception pada variasi 3 dengan nilai 97%, sedangkan akurasi terburuk oleh ResNet50. Komputasi terbaik dicapai oleh MobileNetV2 pada variasi 2 dengan waktu 28,11 detik, sedangkan komputasi terburuk oleh VGG19 dengan waktu 79,9379 detik.

B. Dataset

Dataset adalah objek yang merepresentasikan data dan relasinya di memory. *Dataset* merupakan kumpulan objek dan atributnya [11]. *Dataset* dapat berbentuk angka, teks, gambar, suara, atau kombinasi dari beberapa jenis data tersebut. Penelitian ini menggunakan data berupa gambar dengan format JPG yang diambil secara langsung dari jalanan di Universitas Riau dan sekitarnya. Setelah dilakukan pengambilan data, berhasil didapatkan total 419 citra gambar. Rasio pembagian dataset untuk data latih dan data uji adalah 80:20, dengan demikian di dapatkan 328 gambar untuk data latih dan 91 gambar untuk data uji. Detail yang digunakan pada penelitian ini ditunjukkan seperti pada Tabel IV.

TABEL IV
DETAIL PENGGUNAAN DATASET

No	Nama Kelas	Jumlah	Data Latih	Data Uji
1	Baik	87	64	23
2	Sedang	152	122	30
3	Rusak Ringan	110	88	22
4	Rusak Berat	70	55	15

Adapun tampilan dari *dataset* yang digunakan pada penelitian ini disajikan di Gambar 1.



Gambar 1. Cuplikan tampilan *dataset* penelitian

Pada Gambar 1 ditampilkan contoh daripada citra yang terdapat pada dataset yang digunakan. Mulai dari gambar paling kiri adalah jalan dengan citra baik, sedang, rusak ringan, dan rusak berat.

C. Pre-Processing

Setelah dataset dikumpulkan, selanjutnya dataset memasuki tahapan *pre-processing*. *Pre-processing* di citra digital adalah proses persiapan atau pengolahan awal citra digital sebelum masuk ke tahap pengolahan selanjutnya. Pada penelitian ini dataset terlebih dahulu dibagi dengan rasio 80:20 didalam direktori yang berbeda. Selanjutnya membuat sebuah array untuk menampung dataset setelah dilakukan *pre-processing*. Dataset akan melalui proses *resize* dan *pre-processing input Xception* sebelum dimasukkan kedalam array yang telah dipersiapkan sebelumnya. *Pre-processing input Xception* akan melakukan normalisasi piksel, perubahan format data, pengurangan nilai rata-rata piksel setiap channel gambar, sedangkan ukuran *resize* adalah 299x299. Kode lengkap untuk proses ini dapat dilihat pada Gambar 2 berikut.

```

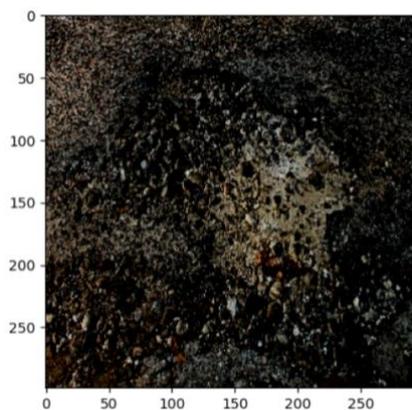
#Pembuatan array untuk data yang ada
IMG_WIDTH = 299
IMG_HEIGHT = 299
IMG_CHANNELS = 3
SAMPLES = 328 #Data Train
SAMPLES2 = 91 #Data Valid
CATEGORIES = 4 #Category
X_train = np.zeros((SAMPLES, IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS), dtype='float32')
y_train = np.zeros((SAMPLES, CATEGORIES), dtype='int32')
X_val = np.zeros((SAMPLES2, IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS), dtype='float32')
y_val = np.zeros((SAMPLES2, CATEGORIES), dtype='int32')

#Pre-processing data train
i = 0
for dirname, _, filenames in os.walk('/content/drive/MyDrive/Dataset v3/Train'):
    for filename in tqdm(filenames):
        _path = os.path.join(dirname, filename)
        _img = load_img(_path, color_mode = "rgb", target_size=(IMG_HEIGHT, IMG_WIDTH))
        _img = np.array(_img, dtype = 'float32')
        _img = preprocess_input(_img)
        X_train[i,:,:,:3] = _img
        _label = dirname.split('/')[-1]
        _index = label_code[_label]
        y_train[i,_index] = 1
        i += 1

#Pre-processing data valid
i = 0
for dirname, _, filenames in os.walk('/content/drive/MyDrive/Dataset v3/Valid'):
    for filename in tqdm(filenames):
        _path = os.path.join(dirname, filename)
        _img = load_img(_path, color_mode = "rgb", target_size=(IMG_HEIGHT, IMG_WIDTH))
        _img = np.array(_img, dtype = 'float32')
        _img = preprocess_input(_img)
        X_val[i,:,:,:3] = _img
        _label = dirname.split('/')[-1]
        _index = label_code[_label]
        y_val[i,_index] = 1
        i += 1
    
```

Gambar 2. Bilah Program Pre-Processing

Pada Gambar 2 ditampilkan bilan program untuk proses *pre-processing*. Proses awal dimulai dengan pembuatan array untuk menampung dataset, kemudian dilakukan fungsi perulangan terhadap data latih dan uji secara terpisah untuk melakukan *pre-processing*. Adapun tampilan dari dataset yang telah melewati proses *pre-processing* disajikan pada Gambar 4 berikut.



Gambar 3. Citra Gambar Setelah Proses Pre-Processing

Pada Gambar 3 ditampilkan contoh citra yang telah melalui proses *pre-processing*. Citra yang telah melalui *pre-processing* akan memiliki ukuran 299x299 dengan pixel yang telah dinormalisasikan

D. Transfer Learning

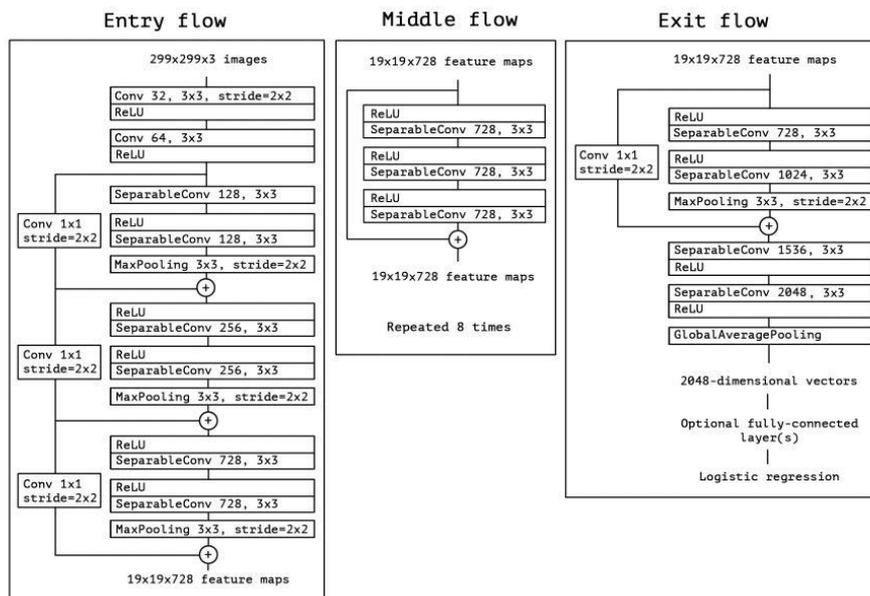
Dataset yang digunakan dalam penelitian ini terbilang sedikit sehingga membutuhkan transfer learning. Transfer learning adalah suatu metode dalam machine learning di mana

sebuah model yang telah dibuat dan dilatih untuk suatu tujuan digunakan kembali untuk tujuan baru atau tujuan lainnya [12]. Metode ini sangat berguna ketika data yang tersedia untuk pelatihan model baru tidak cukup banyak atau tidak cukup beragam untuk memungkinkan pelatihan model dari awal (scratch) dengan performa yang optimal. Transfer learning memungkinkan pemanfaatan model yang telah dilatih pada dataset besar dan beragam, seperti ImageNet, sehingga model tersebut sudah memiliki kemampuan untuk mengenali pola dasar dalam data.

Terdapat banyak pre-trained model yang dapat digunakan dalam olah citra gambar. Beberapa di antaranya adalah VGG19, yang dikenal karena arsitekturnya yang sederhana namun efektif, dan Inceptionv3, yang menawarkan kombinasi layer yang lebih kompleks untuk menangkap berbagai tingkat fitur dalam gambar. Xception, di sisi lain, menggunakan depthwise separable convolutions untuk mengurangi jumlah parameter dan meningkatkan efisiensi komputasi. Dengan memanfaatkan pre-trained model ini, penelitian dapat dilakukan lebih efisien dan hasil yang diperoleh dapat lebih akurat meskipun dataset yang digunakan terbatas.

E. Xception

Pada penelitian ini digunakan pre-trained model *Xception* dalam proses transfer learningnya. Model *Xception* memiliki ukuran yang ringan dan dalam beberapa penelitian di dapati lebih baik dari beberapa pre-trained model lainnya. Arsitektur dari Model *Xception* ini merupakan *stack linear* yang terdiri dari *depthwise separable convolution layers* yang dilengkapi dengan *residual connections* [13]. Arsitektur *Xception* terbagi menjadi tiga bagian: entry flow, middle flow, dan exit flow. Entry flow menerima citra input, kemudian middle flow menerapkan Separable Convolution, sebuah variasi dari Depthwise Separable Convolution, dengan ReLU sebagai fungsi aktivasi pada setiap lapisan konvolusi dan fully connected. Pada tahap terakhir, metode Logistic Regression digunakan. Pada bagian Exit Flow, *Xception* dimodifikasi untuk menerapkan Transfer Learning dengan membuat model kepala kustom. Modifikasi ini melibatkan penghapusan kepala model *Xception* bawaan dan penggantian dengan kepala model kustom. Arsitektur dari *Xception* disajikan dalam Gambar 4 berikut.



Gambar 4. Arsitektur Xception

Pada Gambar 4 ditampilkan arsitektur dari *pre-trained* model *Xception*. Arsitektur terdiri dari tiga lapisan yaitu *entry flow*, *middle flow*, dan *exit flow*. Model yang dilatih akan melewati ketiga lapisan tersebut sebelum akhirnya memasuki *layer* berikutnya.

F. Metode Penelitian Eksperimental

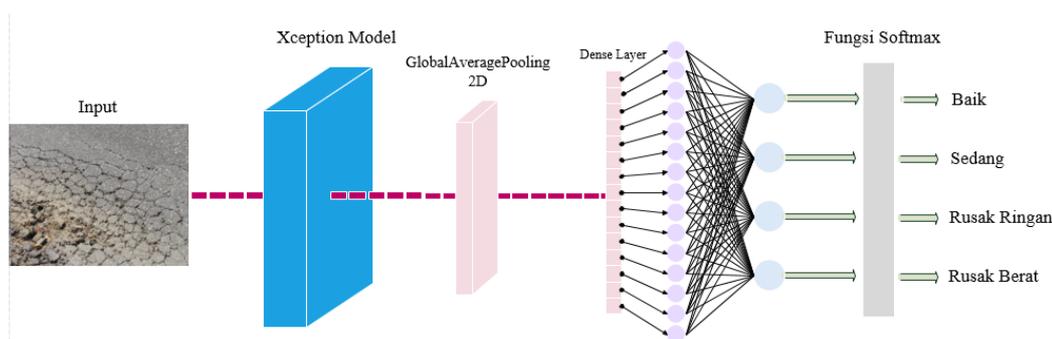
Metode penelitian yang digunakan didalam penelitian ini ialah metode eksperimental. Eksperimen adalah penelitian ilmiah di mana peneliti memanipulasi dan mengontrol satu atau lebih variabel bebas serta mengamati variabel terikat untuk menemukan perubahan yang terjadi akibat manipulasi variabel bebas tersebut. Pada penelitian ini variabel bebas yang digunakan ialah *epoch* dan *batch size*, sedangkan untuk variabel terikat dapat dilihat pada Tabel V berikut.

TABEL V
VARIABLE TERIKAT

Parameter	Tipe
Dense Activation	ReLU
Optimizer	SGD
Loss Function	Categorical, Cross Entropy
Metrics	Accuracy
Output	Softmax

G. Pemodelan

Setelah dataset terkumpul, langkah berikutnya adalah membangun arsitektur model yang akan dilatih. Model ini dibangun dengan menggunakan variabel terikat yang telah ditentukan pada Tabel V. Proses pembangunan model ini penting untuk memastikan bahwa model mampu mempelajari dan mengenali pola-pola dalam data dengan baik. Model yang dirancang memiliki empat lapisan utama selain lapisan input dan output. *Layer* yang terdapat pada model adalah *base model Xception*, *GlobalAveragePooling2D*, *Dense 264 Layer Activation ReLU*, dan terakhir *Dense 4 Layer Activation Softmax*. Visualisasi arsitektur model yang dibangun dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur Model

Kode lengkap untuk membuat arsitektur model dapat dilihat pada Gambar 6 berikut.

```
#Import weights Xception No Top
weights = Path('/content/drive/MyDrive/Xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5')
#Import Xception Pre-Trained Model
base_model = tf.keras.applications.Xception(weights=weights, input_shape=(299,299, 3), include_top=False)
#Model
inputs = tf.keras.Input(shape=(299, 299, 3))
x = base_model(inputs)
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(256, activation = 'relu')(x)
outputs = tf.keras.layers.Dense(4, activation = 'softmax')(x)
model = tf.keras.Model(inputs, outputs)
```

Gambar 5. Bilah Program Pembuatan Model

Pada Gambar 5 dan Gambar 6 diperlihatkan arsitektur dan kode dari model yang dibangun. Model terdiri dari empat *layer* diluar dari *layer* input. Dataset yang telah dipersiapkan akan melewati *layer-layer* tersebut selama proses pelatihan agar dapat mempelajari data dengan baik.

H. Simulasi Pelatihan

Setelah model dibuat, tahapan selanjutnya ialah melatih model dengan menggunakan dataset yang telah di sediakan. Hal ini bertujuan agar didapatkan model yang dapat mengklasifikasikan kualitas jalan raya dengan baik. Model dilatih sebanyak 4 percobaan dengan *hyperparameter* yang dibandingkan adalah Batch Size dan Epoch. Perbandingan pada setiap percobaan dapat dilihat pada Tabel VI.

TABEL VI
PERBANDINGAN PARAMETER PERCOBAAN

Percobaan	Batch Size	Epoch
1	32	40
2	32	60
3	64	40
4	64	60

III. HASIL DAN PEMBAHASAN

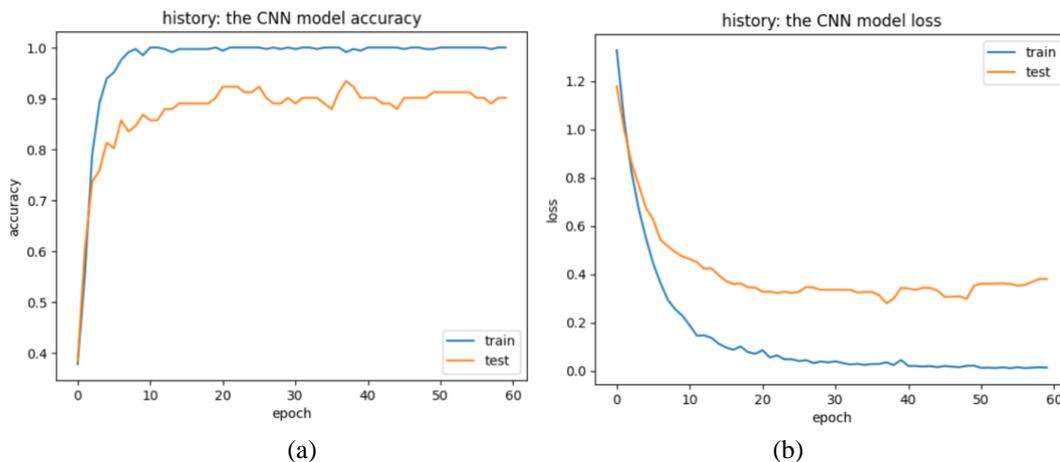
A. Pelatihan Model

Pada penelitian ini model dilatih sebanyak 4 kali, hal ini dilakukan untuk mendapatkan hasil akurasi terbaik yang kemudian akan digunakan dalam proses validasi sistem. Pada empat kali percobaan tersebut variable *epoch* dan *batch size* menjadi *hyperparameter* yang dimanipulasi untuk melihat pengaruhnya terhadap akurasi. Penggunaan *hyperparameter* ini dikarenakan semakin lama dan banyak pelatihan yang dilakukan terhadap suatu model dapat mempengaruhi hasil dari model tersebut [14]. Kombinasi *epoch* dan *batch size* dapat mempengaruhi efisiensi komputasi, stabilitas pembaruan bobot, dan hasil akhir model. Hasil dari proses pelatihan dapat dilihat pada Tabel VII berikut.

TABEL VII
HASIL PERCOBAAN PELATIHAN

Percobaan	Epoch	Batch Size	Accuracy		Loss	
			Training	Validation	Training	Validation
1	40	32	0.9970	0.8571	0.0185	0.4521
2	60	32	1.000	0.8681	0.0092	0.4177
3	40	64	1.000	0.8462	0.0279	0.3198
4	60	64	1.000	0.9011	0.0130	0.3802

Pada Tabel VII dapat dilihat bahwa setiap fase skema percobaan memiliki hasil yang berbeda-beda dimana percobaan dengan batch size 64 cenderung memiliki hasil yang lebih baik. Percobaan dengan nilai terbaik ialah percobaan ke-empat dengan hasil akurasi 90%, namun untuk *validation loss* terbaik di dapatkan pada percobaan ke-tiga dengan nilai *loss* sebesar 0.3198. Untuk grafik *accuracy* dan *loss* dari model terbaik dapat dilihat pada Gambar 6.



Gambar 6. (a) Grafik Accuracy (b) Grafik Loss pada Percobaan ke-empat

Pada Gambar 6 model dilatih dengan besaran *epoch* 60, *batch size* 64. Pada percobaan latih ini didapatkan *validation accuracy* 93% pada *epoch* 38 namun turun ke 90% pada *epoch* 60. Sedangkan pada Gambar 6 (b) diperlihatkan grafik dari loss yang mana pada epoch 60 didapatkan nilai 0.0130 untuk *loss* dan 0.3802 pada *validation loss*. Pada percobaan latih ini akurasi yang didapatkan sudah cukup baik sampai menyentuh 90%. Melihat hasil akurasi penelitian terdahulu pada Tabel III, akurasi model ini sudah cukup baik yang mana lebih tinggi dari penelitian yang dilakukan oleh Puguh [7] dan Yoga [5], namun masih terjadi *overfitting* pada model. *Overfitting* adalah kondisi di mana model pembelajaran mesin terlalu menyesuaikan diri dengan data pelatihan, sehingga tidak dapat menggeneralisasi dengan baik pada data baru. Ini terjadi ketika model terlalu kompleks dan menangkap noise atau fluktuasi acak sebagai pola yang sebenarnya.

B. Hasil Pengujian

Setelah didapatkan model dengan hasil terbaik, model tersebut kemudian masuk kedalam tahapan pengujian. Pada tahapan ini model akan di uji menggunakan data yang berasal dari luar dataset. Data uji yang digunakan dalam penelitian ini berjumlah 40 citra yang mana terdiri dari 10 citra untuk setiap kelas klasifikasi yang ada. Pengujian dilakukan dengan mengload citra kedalam model, kemudian digunakan fungsi prediksi dari *tensorflow* yang akan menghasilkan kelas yang sesuai dengan prediksi sistem.

Proses klasifikasi di mulai dari memasukkan data yang kemudian diproses didalam model *Xception* di dalam *layer* ini terdapat 132 *layers*, selanjutnya di proses dengan *GlobalAveragePooling2D* & *Dense Layer ReLu* dan akhirnya didapatkanlah hasil klasifikasinya menggunakan fungsi *Softmax*. Nilai tertinggi yang dihasilkan dari proses *Softmax* akan dianggap sebagai hasil klasifikasi yang telah dilakukan. Hasil dari pengujian model ini kemudian di sajikan dalam tabel *confussion matrix*. *Confussion matrix* adalah metode untuk mengukur kinerja klasifikasi dengan membandingkan hasil klasifikasi sistem dengan hasil yang seharusnya [15]. Hasil *confussion matrix* dapat dilihat pada Tabel VIII berikut.

TABEL VIII
CONFUSION MATRIX

Matrix	Prediksi Kelas				
	Baik	Sedang	Rusak Ringan	Rusak Berat	
Kelas Asli	Baik	9	1	0	0
	Sedang	2	8	0	0
	Rusak Ringan	0	0	10	0
	Rusak Berat	0	0	1	9

Dari Tabel VIII dapat diketahui hasil pengujian yang telah dilakukan dapat memprediksi dengan benar sebanyak 36 citra dari 40 citra jalan raya. Adapun hasil akhir akurasi dari pengujian yang telah dilakukan terhadap model ialah dengan jumlah prediksi data benar sebanyak 36 dari 40 data uji yakni 90%.

IV. KESIMPULAN

Berdasarkan hasil pelatihan dan pengujian yang dilakukan pada penelitian ini, dapat disimpulkan bahwa tujuan penelitian ini adalah untuk menerapkan teknologi *Machine Learning* dalam *klasifikasi* kualitas jalan raya berdasarkan kualitas permukaannya dengan menggunakan transfer learning arsitektur *Xception*. Didapatkan akurasi terbaik pada percobaan pelatihan keempat dengan nilai akurasi sebesar 90%, namun masih terjadi *overfitting* pada model. *Overfitting* terjadi dikarenakan kurangnya variasi dalam data pelatihan yang mana dataset yang digunakan terbatas pada data yang diambil di daerah Universitas Riau. Telah dicoba pelatihan dengan menggunakan *dropout* dan *augmentation* tidak memberikan dampak yang signifikan terhadap *overfitting* pada dataset yang digunakan pada model ini. *Hyperparameter Epoch* dan *Batch Size* pada fase training dinilai berpengaruh terhadap hasil akurasi dari proses pelatihan. Validasi dilakukan terhadap model dengan akurasi terbaik, dan dari proses pengujian tersebut didapati bahwa model berhasil memprediksi dengan benar sebanyak 36 gambar dari 40 data uji yang diberikan, dengan persentase keberhasilan akhir sebesar 90%. Model ini dapat digunakan membantu mengurangi waktu dan biaya yang dibutuhkan dalam proses pengecekan kualitas jalan raya sehingga proses pengecekan dan perawatan jalan dapat lebih efektif dan efisien.

REFERENSI

- [1] Sitohang O., Hatoguan E., et al. Perancangan Jalan Raya Antar Kota (Rural Road). Edisi Pertama. Bandung. 2022:1.
- [2] Nabawi, I., Feriska, Y., & Diantoro, W. Analisis Dampak Kerusakan Jalan terhadap Pengguna Jalan dan Lingkungan di Ruas Jalan Pebatan-Rengaspendawa Brebes Impact Analysis of Road Damage on Road Users and the Environment on Jalan Pebatan-Rengaspendawa Brebes. *Infratech Building Journal (IJB)*, 2(1), 28–34, 2021.
- [3] Badan Pusat Statistik. Statistik Transportasi Darat 2021. BPS RI, Jakarta, 2022.
- [4] Kementerian PUPR, Survey Pemeliharaan Jalan. Kementerian PUPR RI, Jakarta.
- [5] Yoga Triardhana, Bandi Sasmito, F. H.. Identifikasi Kerusakan Jalan Menggunakan Metode Deep Learning (DI) Model Convolutional Neural Networks (Cnn). *Jurnal Geodesi Undip, DI*, 1–8, 2020.
- [6] Peryanto, A., Yudhana, A., & Umar, R. Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, 4(1), 45–51, 2020.
- [7] Prakoso, P. B., Lestari, U. S., & Sari, Y. Deteksi Keretakan Permukaan Perkerasan Lentur Jalan Raya (Studi Kasus : Tanah Lunak Di Banjarmasin). *Seminar Nasional Lingkungan Lahan Basah*, 4(April), 247–251, 2019.
- [8] Rianto, Y., Riana, D., & Nusa Mandiri, S. Identifikasi Tingkat Kerusakan Jalan Raya Menggunakan Thresholding Dan K-Means Identification of Road Damage Using Thresholding and K-Means. *CSRID Journal*, 13(1), 3906287, 2021. <https://doi.org/https://www.doi.org/10.22303>
- [9] Riyandi, A., Widodo, T., & Uyun, S. Classification of Damaged Road Images Using the Convolutional Neural Network Method. *Telematika*, 19(2), 147, 2022. <https://doi.org/10.31315/telematika.v19i2.6460>

- [10] Putra, A. E., Naufal, M. F., & Prasetyo, V. R. Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 9(1), 12, 2023. <https://doi.org/10.26418/jp.v9i1.58186>
- [11] Yahya, & Mahpuz . Penggunaan Algoritma K-Means Untuk Menganalisis Pelanggan Potensial Pada Dealer SPS Motor Honda Lombok Timur Nusa Tenggara Barat. *Jurnal Informatika Dan Teknologi*, 2(2), 109–118, 2019.
- [12] Zhuang, F. et al. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, Jan. 2021, DOI: 10.1109/JPROC.2020.3004555.
- [13] Imanuel, A., & Setiabudi, D. H. Penerapan Convolutional Neural Network dengan Pre-Trained Model Xception untuk Meningkatkan Akurasi dalam Mengidentifikasi Jenis Ras Kucing. *Jurnal INFRA*, 10(2), 247–253, 2022.
- [14] Pratama, A. Deteksi Kerusakan Jalan Menggunakan Convolutional Neural Network Berbasis SSD EfficientNet. Universitas Kristen Satya Wacana, Jawa Tengah, 2020.
- [15] Zubair, A. Klasifikasi Ketertarikan Mahasiswa Pembelajaran Berbasis Video Business Intelligence Menggunakan Adaboost. *Senasif, September*, 3050–3058, 2022.

UCAPAN TERIMA KASIH

Terima kasih yang tulus kepada semua pihak yang memberi dukungan dengan penuh serta kerjasama yang sangat berperan penting dalam kesuksesan pelaksanaan penelitian ini.