

# Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python

**Ngakan Nyoman Pandika Pinata, I Made Sukarsa, Ni Kadek Dwi Rusjyanthi**

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana  
Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: [pandikapinata.sch@gmail.com](mailto:pandikapinata.sch@gmail.com), [sukarsa@unud.ac.id](mailto:sukarsa@unud.ac.id), [dwi.rusjyanthi@unud.ac.id](mailto:dwi.rusjyanthi@unud.ac.id)

## Abstrak

Tingginya pertumbuhan penduduk di Indonesia menyebabkan kepemilikan kendaraan bermotor pribadi semakin tinggi yang mempengaruhi meningkatnya kemacetan dan juga kecelakaan lalu lintas. Peramalan angka kecelakaan lalu lintas dilakukan pada penelitian ini sebagai salah satu upaya yang dapat dimanfaatkan sebagai dasar tindakan antisipasi terkait peningkatan angka kecelakaan lalu lintas. Penelitian ini bertujuan untuk meramalkan kecelakaan lalu lintas menurut akibatnya menggunakan Xtreme Gradient Boosting (XGBoost) dengan bahasa pemrograman Python. Data yang digunakan dalam penelitian ini yaitu data dari Badan Pusat Statistik Provinsi Bali dengan periode dari Tahun 1996 sampai dengan Tahun 2019 dalam selang waktu tahunan. Hasil peramalan diukur menggunakan RMSE (Root Mean Square Error). Penerapan XGBoost untuk meramalkan data kecelakaan lalu lintas menurut akibatnya, menunjukkan model XGBoost memiliki performa yang sangat baik pada dua kategori yaitu kategori jumlah orang meninggal akibat kecelakaan dengan nilai RMSE 4,92 dan jumlah orang yang mengalami luka berat dengan nilai RMSE 4,11. Nilai RMSE model XGBoost untuk kategori jumlah kejadian kecelakaan lalu lintas yaitu sebesar 21,69 dan kategori orang yang mengalami luka ringan akibat kecelakaan yaitu sebesar 77,24.

**Kata kunci:** Kecelakaan Lalu Lintas, Peramalan, XGBoost, Xtreme Gradient Boosting

## Abstract

The high population growth in Indonesia causes individual transportation ownership increase which affecting congestion and traffic accidents. Predicting the number of traffic accidents is carried out in this study as an effort that can be used as a basis for anticipating an increase in the number of traffic accidents. This study aims to predict traffic accidents according to their consequences using Xtreme Gradient Boosting (XGBoost) with Python programming language. The data used in this study are data from the Bali Province Central Bureau of Statistics with a period from 1996 to 2019 in annual intervals. Forecasting results are measured using RMSE (Root Mean Square Error). The application of XGBoost to predict traffic accident data according to their consequences shows that the XGBoost model has an excellent performance in two categories, namely the category of the number of people who died due to accidents with an RMSE value of 4.92 and the number of people who suffered serious injuries with an RMSE value of 4.11. The RMSE value of the XGBoost model for the category of the number of traffic accidents is 21.69 and the category of people who have suffered minor injuries due to accidents is 77.24.

**Keywords:** Traffic Accidents, Forecasting, XGBoost, Xtreme Gradient Boosting

## 1. Pendahuluan

Kecelakaan lalu lintas di Indonesia terus meningkat seiring dengan meningkatnya kendaraan bermotor yang mengakibatkan kerugian materi, luka ringan, luka berat sampai dengan meninggal [1][2]. Faktor penyebab kecelakaan lalu lintas diantaranya pengemudi, kendaraan dan prasarana maupun lingkungan. Berdasarkan data Badan Pusat Statistik (BPS), Provinsi Bali juga mengalami hal yang sama yaitu kepemilikan kendaraan bermotor pribadi semakin tinggi yang disebabkan oleh bertambahnya penduduk Provinsi Bali sehingga tidak menutup kemungkinan bertambahnya angka kecelakaan lalu lintas yang terjadi [3]. Upaya yang dapat dilakukan untuk mengurangi angka kecelakaan lalu lintas yaitu melihat proyeksi kedepan

atau kecenderungan pola sehingga dapat digunakan mencanangkan strategi ataupun kebijakan agar dapat meminimalisir terjadinya kecelakaan lalu lintas. Proyeksi dapat dilakukan dengan melakukan peramalan pada data runtun atau *time-series*.

Peramalan dapat dilakukan dengan berbagai metode baik secara statistik atau menggunakan machine learning. Penelitian mengenai peramalan kecelakaan lalu lintas yang berfokus pada jumlah kematian di Jawa Timur dengan metode *winter exponential smoothing* sudah dilakukan sebelumnya, dimana metode ini masuk dalam kategori metode statistik. Hasil dari metode *winter exponential smoothing* didapat MAPE (*Mean Absolute Percentage Error*) sebesar 8,10% [4]. Peramalan mengenai kecelakaan lalu lintas juga dilakukan dengan metode ARIMA untuk Kota Pontianak dengan model terbaik yang dihasilkan yaitu ARIMA (0,1,1) dengan MAPE 14,88% [5]. Penelitian mengenai jumlah kecelakaan lalu lintas Provinsi Bali sudah dilakukan sebelumnya, dimana penelitian tersebut menggunakan metode Support Vector Regression (SVR) dengan permasalahan regresi. Metode SVM menggunakan kernel Polynomial menghasilkan nilai Persentase MAPE sebesar 7,95% [6]. Penggunaan *neural network* dalam melakukan peramalan juga sudah dilakukan sebelumnya, dimana menggunakan metode multilayer perceptron untuk melakukan peramalan jumlah tersangka penyalahgunaan narkoba menggunakan data dari BNN (Badan Narkotika Nasional) Provinsi Bali dengan periode Tahun 2007 sampai dengan Tahun 2018. Persentase error yang dihasilkan sebesar 3,7% [7]. Peramalan mengenai kedatangan wisatawan asing menggunakan metode *neural network* dan *single moving average* menunjukkan metode *neural network* menghasilkan nilai *error* yang lebih kecil [8]. Peramalan curah hujan juga dilakukan sebelumnya menggunakan *neural network* yang dibandingkan dengan *support vector regression*. Metode *neural network* menghasilkan nilai MAPE yang lebih kecil sebesar 0,85% dibandingkan SVR sebesar 1,17% [9].

Penelitian terkait lainnya, yaitu penggunaan metode XGBoost (Xtreme Gradient Boosting) dalam meramalkan beban listrik pada *smart building*, dari beberapa model yang sudah diuji dengan metode ARIMA, SARIMA, XGBoost, Random Forest dan Long Short-Term Memory didapat model XGBoost sebagai model terbaik dengan sMAPE (*symmetric Mean Absolute Percentage Error*) sebesar 4,33 [10]. Metode XGBoost disarankan digunakan daripada metode SVM (*Support Vector Machine*) dari segi keakuratan, stabilitas, dan efisiensi dalam meramal radiasi matahari global (H) studi kasus China menggunakan data suhu dan curah hujan [11]. Penelitian ini berfokus pada peramalan pada jumlah kejadian, jumlah orang meninggal dunia, jumlah orang yang mengalami luka ringan, luka berat pada setiap tahunnya berdasarkan data BPS Provinsi Bali. Peramalan menggunakan Xtreme Gradient Boosting dan diimplementasikan pada bahasa pemrograman Python. Python dipilih karena dilengkapi *library* seperti Pandas, Plotly, Numpy, scikit-learn dan XGBoost yang memudahkan dalam pengolahan data. Penelitian ini diharapkan dapat memberikan rekomendasi kepada pihak terkait untuk mencanangkan strategi guna mengurangi kecelakaan lalu lintas yang terjadi pada Provinsi Bali.

## 2. Metodologi Penelitian

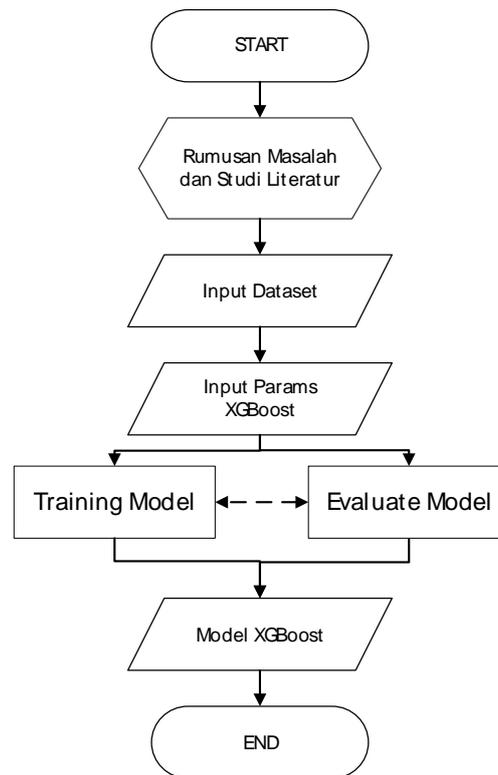
### 2.1 Data

Data yang digunakan untuk melakukan peramalan adalah data jumlah kecelakaan lalu lintas menurut akibatnya dari Tahun 1996 sampai dengan Tahun 2019, yang didapat dari Website Badan Pusat Statistik (BPS) Provinsi Bali [3]. Data yang diperoleh sebanyak 24 data dengan selang waktu tahunan dari Tahun 1996 sampai dengan 2019. Beberapa data yang digunakan dapat dilihat pada Tabel 1.

Tabel 1. Data kecelakaan lalu lintas di Bali tahun 1996 sampai 2019

Tahun	Kejadian	Meninggal Dunia	Luka Berat	Luka Ringan
1996	638	408	385	459
1997	1084	473	631	725
1998	729	540	359	443
1999	677	493	387	425
2000	643	462	399	475

## 2.2 Gambaran Umum



Gambar 1. Alur penelitian

Tahap awal dalam melakukan penelitian ini yaitu dengan menentukan rumusan masalah dan studi literatur. Rumusan masalah yang diambil yaitu bagaimana cara melakukan peramalan pada angka kecelakaan lalu lintas menurut akibatnya dengan studi kasus Provinsi Bali. Peneliti selanjutnya melakukan studi literatur, ditemukan metode yang cukup baik dalam melakukan prediksi yaitu dengan metode Xtreme Gradient Boosting Decision Tree (XGBoost) [10] [11]. Tahap berikutnya yaitu memuat dataset dan *input* nilai parameter model XGBoost. *Training* dan *evaluate* model dilakukan setelah memasukkan nilai params. *Training* dan *evaluate* dikerjakan secara bersamaan. Hasil akhir yaitu model XGBoost terbaik untuk masing-masing kategori berdasarkan data jumlah kecelakaan lalu lintas menurut akibatnya.

## 3. Kajian Pustaka

### 3.1 XGBoost

XGBoost ditemukan pada Tahun 2016 oleh Tianqi Chen [12], dimana metode XGBoost merupakan pengembangan dari algoritma GBDT (*Gradient Boosting Decision Tree*) yang sebelumnya ditemukan oleh Friedman [13]. XGBoost merupakan pustaka *machine learning* yang dapat digunakan untuk melakukan prediksi maupun klasifikasi, sama seperti metode lainnya XGBoost juga dapat diterapkan pada berbagai bidang seperti pendidikan [14], kesehatan [15], pemerintahan [6] dan lain-lain. XGBoost dalam prosesnya memerlukan beberapa parameter sebagai acuan diantaranya adalah sebagai berikut [16]:

- 1) `colsample_bytree` merupakan parameter untuk memilih banyak *sample* kolom yang akan digunakan, *default* 1 yang berarti keseluruhan kolom. *Range* parameter dari 0 sampai 1.
- 2) `eta` merupakan parameter *learning rate* yang berfungsi untuk mencegah model mengalami *overfitting*. *Range* parameter dari 0 sampai 1.
- 3) `gamma` adalah parameter untuk menentukan pemangkasan dari node pada pohon yang dibuat. Semakin besar `gamma` semakin konservatif model yang dibangun. *Range* parameter dari 0 sampai tak terhingga.

- 4) `max_depth` merupakan parameter untuk menentukan kedalaman pohon yang akan dibangun, default 6. *Range* parameter dari 0 sampai tak terhingga.
- 5) `min_child_weight` merupakan parameter untuk menentukan batasan berat minimum yang dimiliki oleh sebuah node. *Range* parameter dari 0 sampai tak terhingga.
- 6) `subsample` adalah parameter untuk memilih banyak *sample* baris data yang akan digunakan, *default* 1 yang berarti keseluruhan baris data. *Range* dari 0 sampai 1.
- 7) `objective` merupakan parameter yang berfungsi untuk menentukan tujuan dari model yang dibangun seperti regresi ataupun klasifikasi.
- 8) `eval_metric` adalah parameter untuk memilih ukuran evaluasi yang digunakan terdapat banyak ukuran evaluasi seperti RMSLE, RMSE, MAE, MAPE, AUC dan lain-lain.

### 3.2 Python

Python adalah bahasa pemrograman tingkat tinggi. Python dibuat oleh Guido van Rossum di Centrum Wiskunde & Informatica (CWI), Belanda dan pertama kali dirilis pada Tahun 1991. Python dapat dipergunakan untuk proyek skala kecil ataupun besar [17]. Python saat ini sudah mencapai versi 3.x dan dapat digunakan untuk berbagai kebutuhan seperti *web development*, *GUI development*, *scientific*, *software development*, dan *system administration* [18].

### 3.3 Persiapan XGBoost di Python

Library XGBoost dalam Kernel berbasis *cloud service* biasanya sudah terinstal secara *default*, namun dalam *local machine computer* harus memasangnya terlebih dahulu. Sintaks untuk memasang pustaka XGBoost dapat dilihat pada Kode Program 1.

```
pip install xgboost
```

Kode Program 1. Instalasi pustaka XGBoost

XGBoost apabila sudah terpasang pada *local machine computer* atau Kernel berbasis *cloud service* dapat digunakan dengan cara mengimpornya terlebih dahulu seperti Kode Program 2.

```
import xgboost as xgb
```

Kode Program 2. *Import* pustaka XGBoost

### 3.4 Ukuran Evaluasi

Ukuran evaluasi yang digunakan dalam penelitian ini yaitu RMSE (*Root Mean Square Error*). RMSE merupakan akar kuadrat dari MSE (*Mean Square Error*) yang dimana dalam penerapannya MSE berpengaruh signifikan dalam dataset yang memiliki *outlier*. Nilai RMSE dapat dicari dengan menggunakan rumus pada Persamaan 1.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(A_i - y_i)^2}{n}} \quad (1)$$

Penjelasan mengenai Persamaan 1 adalah sebagai berikut:

$A_i$  = nilai aktual pada data ke- $i$

$y_i$  = nilai prediksi pada data ke- $i$

$n$  = jumlah data

## 4. Hasil dan Pembahasan

Implementasi dilakukan menggunakan bahasa pemrograman Python 3 dengan spesifikasi hardware processor Intel Core i3-5005U, RAM sebesar 8GB, SSD 250GB dan GPU NVIDIA GeForce 930M. Langkah pertama yaitu memuat data. Memuat data dilakukan menggunakan pustaka Pandas. Sintaks dalam memuat data dapat dilihat pada Kode Program 3.

```
kecelakaan_df = pd.read_excel('Kecelakaan.xlsx', index_col='Tahun')
```

### Kode Program 3. Memuat data menggunakan Pandas

Kode Program 3 adalah langkah untuk memuat data menggunakan pustaka Pandas, dikarenakan data memiliki format *excel* maka fungsi yang digunakan adalah `read_excel`. Fungsi `read_excel` didalamnya terdapat parameter `index_col` yang berfungsi memilih kolom yang nantinya dijadikan *index* dalam *DataFrame*. Data yang sudah dimuat selanjutnya dilakukan visualisasi yang dapat dilihat pada Gambar 2.



Gambar 2. Visualisasi kecelakaan lalu lintas menurut akibat kecelakaan

Berdasarkan visualisasi pada Gambar 2 diketahui pola “Kejadian” kecelakaan dengan “Luka Ringan” memiliki pola yang serupa kemudian, kecenderungan orang yang mengalami luka berat akibat kecelakaan memiliki kecenderungan menurun dari Tahun 2011 sampai dengan Tahun 2019. Orang yang meninggal akibat kecelakaan lalu lintas di Bali dalam 5 tahun terakhir berkisar dari 400 sampai dengan 500 orang setiap tahunnya. Tahap berikutnya yaitu membuat *DataFrame* baru sesuai dengan kolom akibat kecelakaan lalu lintas.

```
kejadian_df = kecelakaan_df[['Kejadian']].copy()
meninggal_df = kecelakaan_df[['Meninggal Dunia']].copy()
berat_df = kecelakaan_df[['Luka Berat']].copy()
ringan_df = kecelakaan_df[['Luka Ringan']].copy()
```

### Kode Program 4. Membuat *DataFrame* baru

Kode Program 4 merupakan sintaks membuat *DataFrame* baru. *DataFrame* baru yang dibuat diantaranya adalah `kejadian_df` yang memuat jumlah kejadian kecelakaan lalu lintas, `meninggal_df` yang memuat jumlah orang yang meninggal akibat kecelakaan lalu lintas, `berat_df` yang memuat jumlah orang yang mengalami luka berat akibat kecelakaan lalu lintas dan `ringan_df` yang memuat jumlah orang yang mengalami luka ringan akibat kecelakaan lalu lintas.

Implementasi berikutnya yaitu melakukan *preprocessing* pada setiap *DataFrame* yang sudah dibuat sebelumnya dengan merubah tipe data atribut `Tahun` dari *string* menjadi *date(year)*. Data yang sudah melalui *preprocessing* selanjutnya dibagi menjadi data latih dan data uji. **Data latih** sebanyak **20 data** dari Tahun 1996 sampai Tahun 2015. **Data uji** sebanyak **4 data** dengan periode dari Tahun 2016 sampai dengan Tahun 2019. Tahap selanjutnya yaitu membuat data *matrix* untuk masukan model *XGBoost*. Data *Matrix* atau yang disingkat menjadi *DMatrix* merupakan struktur data yang digunakan oleh *XGBoost*. Sintaks mengubah data menjadi *DMatrix* dapat dilihat pada Kode Program 5.

```
xgb.DMatrix(atribut_data, label=target_data)
#Contoh
```

```
dtrain = xgb.DMatrix(atribut_data_latih, label=target_data_latih)
dtest = xgb.DMatrix(atribut_data_uji, label=target_data_uji)
```

Kode Program 5. Transformasi data menjadi DMatrix

Langkah berikutnya yaitu menginisiasi model XGBoost dengan memasukkan beberapa parameter seperti eta, max\_depth, min\_child\_weight, subsample, colsample\_bytree, dan gamma. Kode Program dalam inisiasi model XGBoost dapat dilihat pada Kode Program 6.

```
params = {
    'colsample_bytree': float (range 0-1),
    'eta': float,
    'gamma': float,
    'max_depth': int,
    'min_child_weight': float,
    'subsample': float (range 0-1),
    'objective': 'reg:squarederror',
    'eval_metric': 'rmse'
}

model = xgb.train(
    params,
    data_latih,
    num_boost_round=int,
    evals=[(data_uji, "Test")],
    early_stopping_rounds=int
)
```

Kode Program 6. *Blueprint* model XGBoost

Kode Program 6 merupakan *blueprint* dalam inisiasi model XGBoost. Nilai-nilai pada variabel params dicoba terus-menerus dengan berbagai nilai sampai mendapatkan suatu model dengan nilai *error* yang kecil. Variabel model mempunyai beberapa atribut seperti params yaitu parameter yang sudah diinisiasi sebelumnya, kemudian data\_latih merupakan data yang dilatih untuk dapat memprediksi pada data uji, num\_boost\_round merupakan banyak boosting yang dilakukan, atribut evals untuk menunjuk data uji yang digunakan dan early\_stopping\_rounds merupakan atribut untuk melakukan pemberhentian *boosting* jika nilai *error* pada suatu *tree* tidak mengalami perubahan yang signifikan.

*Training* model dilakukan dengan 20 baris data latih yang sudah dikonversikan sebelumnya menjadi DMatrix dan tahap evaluasi model langsung dapat dilakukan karena sudah mengisi nilai pada atribut evals. Hasil evaluasi model dengan beberapa parameter pada model “Kejadian” dapat dilihat pada Tabel 2.

Tabel 2. Hasil beberapa model “Kejadian”

colsample_bytree	eta	gamma	max_detph	min_child_weight	subsample	score
0,95	0,175	2,10	8	2,0	0,90	70,13
0,85	0,275	6,90	2	4,0	0,80	56,14
0,95	0,300	2,65	8	2,0	0,60	50,80
0,75	0,2	3,65	8	3,0	0,65	21,69

Tabel 2 merupakan beberapa hasil pencarian model “Kejadian” dengan beberapa nilai parameter yang berbeda. Seluruh parameter sangat berpengaruh terhadap hasil model. Langkah berikutnya setelah ditemukan parameter yang tepat, maka dilakukan *training* untuk dapat memprediksi angka kecelakaan pada data uji. Prediksi pada data uji dapat dilakukan dengan sintaks seperti Kode Program 7.

```
model.predict(dtest)
```

Kode Program 7. *Import* pustaka XGBoost

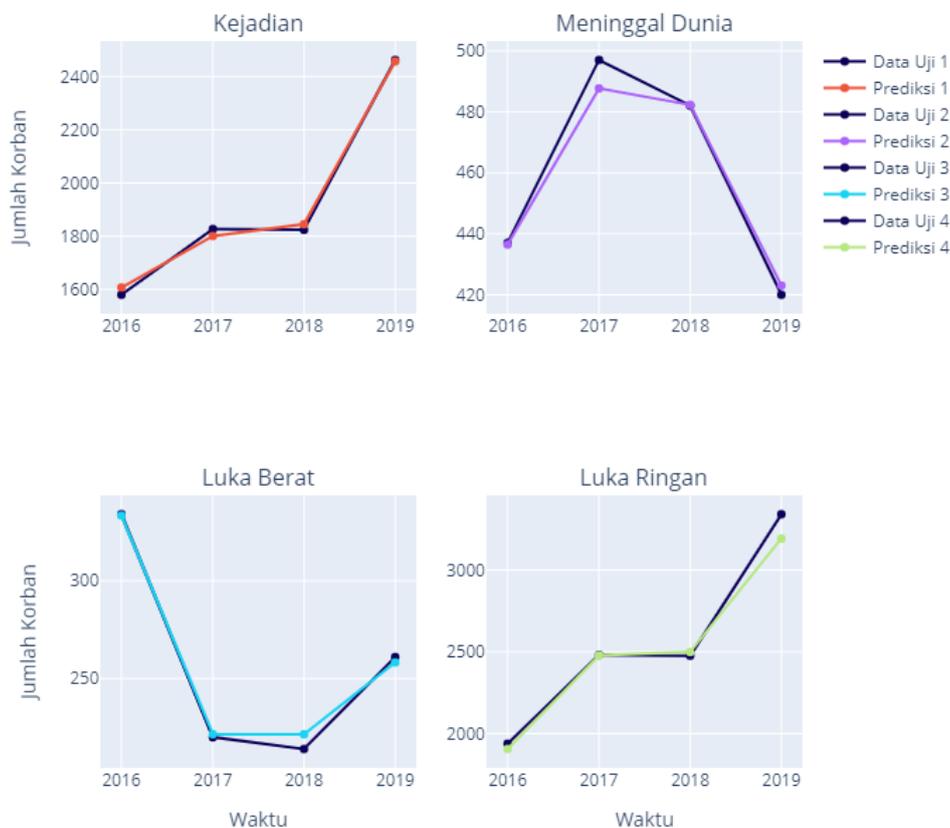
Kode Program 7 merupakan sintaks untuk melakukan prediksi pada data uji dengan model yang sudah dilatih sebelumnya. Parameter dalam fungsi *predict* harus berupa DMatrix. Model terbaik untuk masing-masing kategori kecelakaan menurut akibatnya dapat dilihat pada Tabel 3.

Tabel 3. Pengelompokan kategori MAPE

Model	RMSE
Model "Kejadian"	21,69
Model "Meninggal Dunia"	4,92
Model "Luka Berat"	4,11
Model "Luka Ringan"	77,24

Hasil dari semua model XGBoost pada masing-masing kategori kecelakaan menurut akibatnya memiliki RMSE yang cukup rendah, terutama pada Model "Meninggal Dunia" dan Model "Luka Berat". Nilai RMSE yang rendah menunjukkan model dapat mengikuti pola data latih dan dapat memprediksi pola selanjutnya yang terdapat pada data uji. Model dengan akurasi tertinggi dimiliki oleh model jumlah orang mengalami luka berat akibat kecelakaan lalu lintas dengan nilai RMSE 4,11. Model yang memiliki akurasi terendah yaitu model jumlah orang mengalami luka ringan akibat kecelakaan lalu lintas dengan nilai RMSE 77,24.

Hasil semua Model XGBoost



Gambar 3. Hasil peramalan pada data kecelakaan lalu lintas

Gambar 3 merupakan visualisasi perbandingan antara data aktual dengan hasil prediksi pada masing-masing model. Data aktual ditunjukkan garis berwarna biru tua (gelap) sedangkan hasil prediksi ditunjukkan garis yang berwarna merah, ungu, biru muda, dan hijau. Data uji pada masing-masing model sebanyak 4 data dari Tahun 2016 sampai 2019. Berdasarkan Gambar 3 diketahui masing-masing model dapat mengikuti pola pada setiap kategori kecelakaan lalu lintas.

## 5. Kesimpulan

Berdasarkan hasil penelitian mengenai penerapan XGBoost (Xtreme Gradient Boosting) untuk meramalkan jumlah kejadian, jumlah orang meninggal dunia, jumlah orang yang mengalami luka ringan, dan luka berat pada setiap tahunnya menghasilkan nilai *error* yang cukup rendah. Nilai *error* dari 4 kategori masing-masing yaitu RMSE 21,69 untuk model peramalan jumlah kejadian, RMSE 4,92 untuk model peramalan jumlah orang meninggal dunia, RMSE 4,11 untuk model peramalan jumlah orang yang mengalami luka berat dan RMSE 77,24 untuk model peramalan jumlah orang yang mengalami luka ringan.

## Daftar Pustaka

- [1] BPS. "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi 2016-2018." 2018.
- [2] BPS. "Jumlah Kendaraan Bermotor (Unit), 2016-2018." 2018.
- [3] BPS. "Banyaknya Kecelakaan Lalu Lintas Menurut Akibat Kecelakaan di Bali. 1996-2019." 2020.
- [4] P. R. Nuharianti. "Peramalan Jumlah Kematian Akibat Kecelakaan Lalu Lintas di Jawa Timur dengan Metode Winter Exponential Smoothing." *Jurnal Ilmiah Kesehatan Media Husada*. 2016; 5(2): 167-177.
- [5] C. Y. Ismayanti dan D. Kusnandar. "Verifikasi Model Arima Pada Peramalan Jumlah Kecelakaan Lalu Lintas Kota Pontianak Menggunakan." *Buletin Ilmiah Matematika Statistika dan Terapannya*. 2019; 08(3): 421-428.
- [6] N. Putu, R. Apriyanti, I. K. Gede, D. Putra, dan I. M. S. Putra. "Peramalan Jumlah Kecelakaan Lalu Lintas Menggunakan Metode Support Vector Regression." *Jurnal Ilmiah Menara Penelitian Akademika Teknologi Informasi*. 2020; 8(2): 72-80.
- [7] P. Githa Pratiwi, I. Ketut Gede Darma Putra, dan D. Purnami Singgih Putri. "Peramalan Jumlah Tersangka Penyalahgunaan Narkoba Menggunakan Metode Multilayer Perceptron." *Jurnal Ilmiah Menara Penelitian Akademika Teknologi Informasi*. 2019; 7(2): 143-150.
- [8] W. O. Vihikan, I. K. G. D. Putra, dan I. P. A. Dharmadi. "Foreign Tourist Arrivals Forecasting using Recurrent Neural Network Backpropagation Through Time." *Telkomnika (Telecommunication Computing Electronics and Control)*. 2017; 15(3): 1257-1264.
- [9] I. A. P. Utami dan P. W. Putra, I Ketut Gede Darma Buana. "Rainfall Prediction with Neural Network Method and Support Vector Regression." *Water Energy International*. 2019; 62r(6): 51-55.
- [10] S. Hadri, Y. Naitmalek, M. Najib, M. Bakhouya, Y. Fakhri, dan M. Elaroussi. "A Comparative Study of Predictive Approaches For Load Forecasting In Smart Buildings." *Procedia Computer Science*. 2019; 160: 173-180.
- [11] J. Fan *et al.*, "Comparison Of Support Vector Machine And Extreme Gradient Boosting For Predicting Daily Global Solar Radiation Using Temperature And Precipitation In Humid Subtropical Climates: A Case Study In China." *Energy Conversion and Management*. 2018; 164: 102-111.
- [12] T. Chen dan C. Guestrin, "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco. 2016; 785-794.
- [13] J. H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*. 2001; 29(5): 1189-1232.
- [14] R. R. Waliyansyah dan N. D. Saputro. "Forecasting New Student Candidates using the Random Forest Method." *Lontar Komputer Jurnal Ilmiah Teknologi Informasi*. 2020; 11(1): 44-56.
- [15] M. Mirqotussa'adah, M. A. Muslim, E. Sugiharti, B. Prasetyo, dan S. Alimah. "Penerapan

- Dizcretization dan Teknik Bagging untuk Meningkatkan Akurasi Klasifikasi Berbasis Ensemble pada Algoritma C4.5 dalam Mendiagnosa Diabetes.” *Lontar Komputer Jurnal Ilmiah Teknologi Informasi.*, 2017; 8(2): 135-143.
- [16] M. Ayub. “Proses Data Mining dalam Sistem Pembelajaran Berbantuan Komputer.” *Jurnal Sistem Informasi.* 2007; 2(1): 21–30.
- [17] XGBoost Developers. “XGBoost Parameters.” Tersedia pada: <https://xgboost.readthedocs.io/en/latest/parameter.html>, diakses 25 Juni 2020.
- [18] Python Software Foundation. “Python.” Tersedia pada: <https://www.python.org/>, diakses 25 Juni 2020.