Prediksi Total Transaksi Menggunakan *Extreme Gradient Boosting (XGBoost)*

**Mohamad Akbar Wisnu Nadyanto\*1, Desy Komalasari2**

1,2Program Studi Ilmu Komputer; STIMIK ESQ Business School

Jl. Tb. Simatupang, Kav. 1 Cilandak Timur, Jakarta 11560

e-mail: \*1m.akbar.w@students.esqbs.ac.id, 2desy.komalasari@esqbs.ac.id

***Abstrak:****Saat ini, jumlah data tumbuh secara eksponensial. Seiring dengan hal tersebut, data dapat diolah menjadi suatu informasi yang bermanfaat menggunakan ilmu statistika terutama dalam bidang bisnis. Dengan pertumbuhan data di era digital seperti sekarang, pengolahan data tidak lagi hanya dilakukan dengan pendekatan ilmu statistika saja tetapi dapat digabungkan dengan pendekatan teknologi machine learning. Hal ini dikarenakan dengan meningkatnya volume data maka pengolahan data menggunakan statistika memerlukan waktu yang cukup lama dan rentan terhadap kesalahan masukan ataupun perhitungan sehingga berdampak pada hasil. Untuk itu machine learning hadir sebagai otomatisasi dalam mengolah data yang juga menggunakan teori statistika dalam membangun model matematika. Dalam penelitian ini akan menggunakan teknik prediksi regresi pada transaksi penjualan dengan pendekatan machine learning yang berperan sebagai otomatisasi data. Beberapa penelitian tentang prediksi regresi pada transaksi penjualan sampai saat ini masih sering diteliti dengan melibatkan machine learning. Oleh karena itu, pada penelitian ini akan dilakukan prediksi regresi total transaksi dengan melibatkan machine learning dan menggunakan metode XGBoost. Kinerja dari XGBoost dapat tergolong sangat baik, hal ini dibuktikan dalam pengujian model yang telah dilakukan. Hasil evaluasi model menunjukkan nilai MAE pada data latih dan data uji masing-masing menghasilkan nilai sebesar 0.1353 dan 0.1308. Sedangkan evaluasi menggunakan pada data latih dan data uji masing-masing menunjukkan nilai yang sama yaitu 0,99.*

***Kata Kunci:*** *Prediksi, Penjualan, Machine Learning, XGBoost.*

# pendahuluan

Pada saat ini, jumlah data tumbuh secara eksponensial. Seiring dengan hal tersebut, data dapat diolah menjadi suatu informasi yang bermanfaat menggunakan ilmu statistika terutama dalam bidang bisnis. Dengan pertumbuhan data di era digital seperti sekarang, pengolahan data tidak lagi hanya dilakukan dengan pendekatan ilmu statistika saja tetapi dapat digabungkan dengan pendekatan teknologi *machine learning*. Hal ini dikarenakan dengan meningkatnya volume data maka pengolahan data menggunakan statistika memerlukan waktu yang cukup lama dan rentan terhadap kesalahan masukan ataupun perhitungan sehingga berdampak pada hasil. Untuk itu *machine learning* hadir sebagai otomatisasi dalam mengolah data yang juga menggunakan teori statistika dalam membangun model matematika [1]. Menurut Rajula dkk., [2] perbedaan utama antara pendekatan statistika dan *machine learning* adalah statistika bertujuan untuk menyimpulkan hubungan antar variabel, sementara *machine learning* fokus dalam membuat model prediksi. Kelebihan *machine learning* yaitu memiliki fleksibilitas yang besar dan bebas dari asumsi, sedangkan statistika bergantung pada asumsi yang kuat seperti jenis distribusi kesalahan, parameter dalam prediktor linear, dan proporsional. *Machine learning* juga memiliki kemampuan untuk menganalisis berbagai tipe data dan mengintegrasikannya ke dalam prediksi. Salah satu teknik yang banyak dilakukan dalam *machine learning* adalah prediksi.

Prediksi adalah perkiraan apa yang akan terjadi di masa yang akan datang. Prediksi didapat berdasarkan pengetahuan dan pengalaman di masa lalu [3]. Dalam penerapannya, prediksi dibagi menjadi dua yaitu klasifikasi dan regresi. Klasifikasi adalah pengelompokkan data yang digunakan untuk variabel target diskrit. Hanya beberapa jenis kemungkinan nilai target yang diperoleh dan tidak ada nilai deret waktu yang harus diperoleh untuk mendapatkan nilai akhir. Sedangkan regresi mirip dengan klasifikasi, perbedaannya adalah regresi tidak dapat mencari pola yang dijabarkan sebagai kelas dan regresi digunakan untuk variabel target kontinu. Dalam regresi, nilai deret waktu harus dihitung untuk mendapat nilai akhir dan ada nilai kontinu yang harus dihitung untuk mendapatkan nilai target akhir yang diinginkan [4].

Seiring berkembangnya jumlah data dan teknologi yang sangat pesat, pendekatan *machine learning* banyak digunakan dalam bidang bisnis terutama prediksi transaksi penjualan [5]. Dengan melakukan prediksi regresi pada transaksi penjualan, maka dapat membantu perusahaan membuat rencana penjualan yang lebih baik sehingga dapat menentukan langkah yang tepat dalam meningkatkan transaksi penjualan tersebut. Oleh karena itu, perlu dilakukan pendekatan *machine learning* guna untuk otomatisasi data agar pengolahan data dalam jumlah yang besar dapat lebih cepat dan menghasilkan hasil prediksi yang akurat.

*Machine Learning* atau pembelajaran mesin adalah pemrograman komputer untuk mengoptimalkan kriteria kinerja menggunakan contoh data atau pengalaman masa lalu [1]. Pemodelan *machine learning* ditentukan dengan beberapa parameter dan pembelajaran adalah eksekusi program komputer untuk mengoptimalkan parameter model menggunakan data pelatihan atau pengalaman masa lalu. Model dalam *machine learning* dapat bersifat prediksi untuk membuat prediksi di masa depan atau deskriptif untuk memperoleh pengetahuan dari data atau keduanya. Dalam kasus ini, *machine learning* yang dilakukan adalah prediksi regresi dengan menggunakan metode *machine learning*. Prediksi yang dilakukan yaitu regresi dengan memprediksi data numerik yang bersifat kontinu dan mengevaluasi model yang telah dibuat tersebut. Salah satu metode *machine learning* yang cukup populer digunakan dalam penelitian prediksi regresi adalah metode *XGBoost*.

*eXtreme Gradient Boosting* atau lebih sering dikenal dengan singkatan *XGBoost* adalah suatu model algoritma *machine learning* yang merupakan implementasi pohon regresi dan memiliki aturan keputusan yang sama dengan *decision tree*. Dalam pohon regresi, setiap simpul merepresentasikan nilai untuk pengujian atribut dan setiap daunnya dengan skor merepresentasikan keputusan [6]. *XGBoost* juga termasuk pemecah model linear yang efisien dan algoritma pembelajaran pohon, algoritma ini juga dapat digunakan untuk berbagai fungsi tujuan, termasuk regresi, klasifikasi, dan peringkat. Keunggulan *XGBoost* diantaranya dapat melakukan komputasi paralel secara otomatis sehingga secara general lebih cepat 10 kali dari *gradient boosting machines* (GBM), *XGBoost* juga mendukung kustomisasi fungsi objektif, fungsi evaluasi, dan memiliki performa yang lebih baik pada beberapa *dataset* yang berbeda (T. Chen dan He, 2017). Dengan alasan itulah *XGBoost* merupakan salah satu algoritma *machine learning* yang popular dan banyak digunakan dalam penelitian prediksi klasifikasi ataupun regresi.

Penelitian terhadap prediksi regresi transaksi atau penjualan telah banyak dilakukan, diantaranya penelitian berjudul “*Machine Learning Model for Sales Forecasting by Using XGBoost*” [6]. Penelitian ini menggunakan metode *XGBoost* dan tujuan penelitian ini adalah untuk mengetahui performa model algoritma *XGBoost*. Hasil penelitian tersebut menunjukkan perhitungan RMSSE (*Root Mean Squared Scaled Error*) diperoleh hasil model *XGBoost* terendah sebesar 0,655, regresi linear memperoleh hasil sebesar 0,783, 19,5% lebih tinggi dari model *XGBoost*, dan model *ridge regression* mendapatkan hasil sebesar 0,774, juga 13,6% lebih tinggi dari model *XGBoost*.

Penelitian selanjutnya berjudul “*Walmart Sales Forecasting using XGBoost Algorithm and Feature Engineering*” [5]. Penelitian ini menggunakan metode *XGBoost* dan tujuan penelitian ini adalah untuk mengetahui performa model algoritma *XGBoost*. Hasil penelitian tersebut menunjukkan perhitungan RMSSE (*Root Mean Squared Scaled Error*) diperoleh hasil model *XGBoost* terendah sebesar 0,652, regresi logistik menghasilkan nilai sebesar 0,793, 0,141 lebih besar dari model *XGBoost*, dan model *ridge regression* menghasilkan nilai sebesar 0,765, 0,113 lebih besar dari model *XGBoost*.

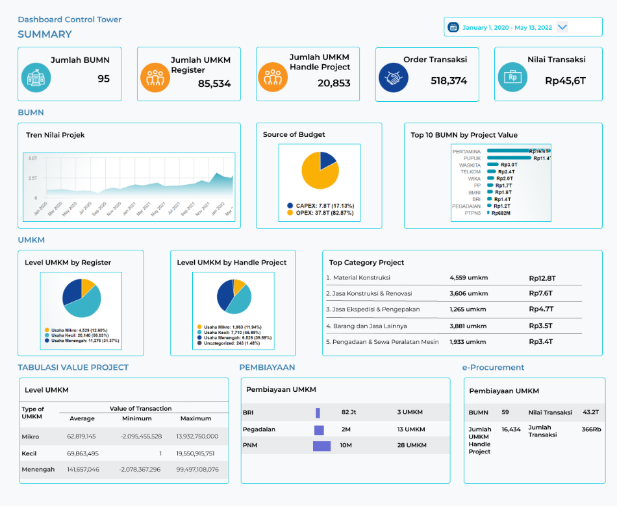
Berdasarkan penjelasan dan beberapa referensi penelitian terdahulu yang telah dijabarkan, *XGBoost* memiliki nilai error yang paling rendah dibanding dengan metode *machine learning* lainnya. Oleh karena itu, pada penelitian ini akan menerapkan konsep *machine learning* menggunakan metode *XGBoost* untuk melakukan analisis model prediksi regresi total transaksi. Data yang digunakan adalah data transaksi perusahaan XYZ pada *platform* Pasar Digital Usaha Mikro, Kecil, Menengah (PADI UMKM). Penerapan prediksi yang dilakukan dalam penelitian ini yaitu prediksi regresi yang berfokus pada evaluasi pengujian model sehingga diharapkan mampu mendapatkan model terbaik untuk prediksi regresi total transaksi.

# landasan teori

## Pasar digital usaha mikro, kecil, menengah (padi umkm)

Pasar Digital Usaha Mikro, Kecil dan Menengah (PADI UMKM) merupakan *digital platform* yang diinisiasi oleh Kementerian BUMN dan bersinergi dengan BUMN, Kementerian UMKM dan juga Lembaga Kebijakan Pengadaan Jasa Pemerintah. PADI UMKM juga dibuat dengan tujuan untuk membuka akses pasar yang lebih luas bagi UMKM untuk pengadaan barang dan jasa pemerintah, khususnya BUMN. Pengadaan Barang Dan Jasa Pemerintah Di Marketplace PaDi UMKM. (2022). Diakses pada Mei 13, 2022, dari PADI UMKM: https://padiumkm.id/. PADI UMKM juga diinisiasi untuk memonitor kontribusi BUMN terhadap UMKM di seluruh Indonesia. Monitorisasi meliputi pengumpulan data *seller* UMKM, data transaksi BUMN dengan UMKM, *enabler e-commerce* dengan integrasi *marketplace*, data *e-procurement* BUMN, dan data pembiayaan UMKM yang dapat dipantau pada *Control Tower* PADI UMKM. Layanan fitur yang ada pada PADI UMKM selain *marketplace* B2B dan B2C yaitu otomasi perpajakan yang digunakan pembeli BUMN untuk mempermudah penentuan nilai pajak sebelum bertransaksi, kemudian PADI *e-procurement* yaitu sebagai solusi transformasi digital pengadaan BUMN menggunakan *platform procurement* 4.0, dan *Control Tower Dashboard* untuk mengetahui informasi terkait UMKM dan pembelanjaan BUMN terhadap UMKM.

*Control Tower Dashboard* PADI UMKM adalah salah satu fitur PADI UMKM yang menyajikan informasi terkait UMKM dan pembelanjaan BUMN terhadap UMKM secara statistik dan komprehensif. *Control Tower* mempunyai tujuan agar dapat mendapatkan *insight* berbasis data yang akurat dan *real time* sehingga akan memudahkan untuk pembuatan kebijakan, pemberian insentif, dan aturan baru pada ekosistem UMKM [9]. Pengguna *Control Tower Dashboard* yang menggunakan sistem ini adalah pemerintah dan Kementerian BUMN sebagai dasar analisis data untuk penentuan kebijakan, bagi BUMN untuk melihat total transaksi ataupun nilai transaksi, dan juga bagi UMKM untuk melihat transaksi maupun aktivitas lain yang ada pada ekosistem PADI UMKM. Contoh visualisasi *Control Tower Dashboard* PADI UMKM dapat dilihat pada Gambar 1.



Gambar 1. Visualisasi Data Control Tower Dashboard PADI UMKM

## *machine learning*

*Machine learning* adalah pembelajaran mesin yang memprogram komputer untuk mengoptimalkan performa kriteria menggunakan contoh data atau pengalaman masa lalu. *Machine learning* menggunakan teori statistika dalam membangun model matematika, karena tugas intinya adalah membuat inferensi atau kesimpulan dari sampel. Peran ilmu komputer ada dua: pertama dalam hal pelatihan, perlu algoritma yang efisien untuk memecahkan masalah pengoptimalan, serta untuk menyimpan dan memproses sejumlah data besar yang biasanya dimiliki. Kedua, jika model sudah mempelajarinya, representasi dan solusi algoritmik untuk kebutuhan inferensi menjadi efisien juga. Dalam aplikasi tertentu, efisiensi algoritma pembelajaran atau inferensi dapat diukur dari kompleksitas ruang dan waktu, hal ini sama penting dengan akurasi prediksi nya [1].

Menurut Ibnu Daqiqil [10], *Machine learning* merupakan bidang studi yang berfokus terhadap desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. *Machine learning* juga dapat didefinisikan sebuah yang memiliki kemampuan belajar tanpa diprogram secara eksplisit. Program tersebut memanfaatkan data masa lalu untuk membangun model dan mengambil keputusan atas model yang telah dibuat sebelumnya. Aplikasi *machine learning* memiliki 3 komponen yaitu *task* T, *performance measure* P, dan *experience* E. Untuk membangun aplikasi *machine learning*, maka 3 komponen tersebut harus dapat diidentifikasi. Beberapa contoh *task* T dalam aplikasi *machine learning* adalah sebagai berikut:

* Klasifikasi, pengelompokkan data yang mempunyai kelas label atau target. Pada jenis *task* ini, komputer diperintahkan untuk “menebak” atau memilih kelompok atau kelas dari data. Contohnya adalah klasifikasi emosi pada foto yang mempunyai 3 kelas, yaitu *happy, angry,* dan *neutral*.
* Regresi, salah satu aplikasi *machine learning* untuk komputer diperintahkan “menebak” angka pada tipe data kontinu berdasarkan sejumlah data. Contohnya pada prediksi harga saham, maka *output* dari prediksi tersebut adalah nilai prediksi harga saham.
* Transkripsi, tipe *task machine learning* yang mencoba memahami sebuah data yang tidak terstruktur dan mempresentasikan data lain. Contohnya adalah aplikasi *speech to text* yang mana data berupa gelombang suara berisi teks lalu diubah menjadi teks.
* *Machine Translation*, proses mengubah sekumpulan masukan terurut yang berisi simbol-simbol tertentu menjadi simbol tertentu pada bahasa lainnya. Contohnya adalah aplikasi penerjemah bahasa Indonesia dan bahasa Inggris.
* *Anomaly Detection*, tipe *task* yang bertujuan untuk memeriksa sejumlah data lalu menandainya sebagai sesuatu yang tidak biasa.
* *Syntesis* dan *Sampling*, tipe *task* yang bertujuan untuk membuat sesuatu berdasarkan contoh yang ada. Contohnya adalah aplikasi yang dapat menghasilkan sebuah gambar pemandangan berdasarkan contoh sketsa tertentu.

*Machine learning* dapat diklasifikasikan berdasarkan bagaimana cara mesin belajar sehingga dapat melakukan tugasnya. Klasifikasi *machine learning* berdasarkan cara belajarnya adalah sebagai berikut:

* *Supervised Learning*, komputer akan mempelajari data latih yang diberi label. Contohnya adalah deteksi kucing dan anjing, komputer menerima masukan data kucing dan anjing lalu mempelajarinya dengan algoritma tertentu, kemudian membuat model berdasarkan data latih yang berisi contoh kucing dan anjing. Hasilnya komputer dapat mengelompokkan kucing dan anjing yang belum pernah dilihat. Contoh algoritma yang termasuk dalam *supervised learning* adalah Regresi Linear, Regresi Logistik, *k-Nearest Neighbor, Support Vector Machine* (SVM), *Random Forests*, *Neural Networks*, dan lain sebagainya.
* *Unsupervised Learning*, berbeda dengan *supervised learning*. *Unsupervised learning* adalah kebalikannya yang mana proses pembelajaran dilakukan tanpa petunjuk pada data yang tidak diberi label. Contohnya pada klasterisasi data dua item yang berbeda, kedua item tersebut akan dipisah menjadi beberapa kategori tergantung dari data. Komputer hanya mengetahui fitur-fitur yang akan digunakan untuk mengetahui perbedaan kedua item tersebut adalah warna bentuk. Dengan menerapkan algoritma klastering, komputer akan mampu membagi item-item menjadi dua kelompok tanpa harus diberi label sebelumnya dan otomatis terkelompok berdasarkan warna dan bentuk objek. Contoh algoritma yang termasuk dalam *unsupervised learning* yang sederhana adalah *K-Means*.
* *Reinforcement Learning,* dalam pembelajaran mesin sebuah komputer akan berinteraksi dengan sebuah lingkungan yang sangat dinamis yang mana komputer harus melakukan sebuah tugas tertentu. Mesin akan mempelajari bagaimana membuat keputusan yang spesifik berdasarkan suatu lingkungan yang berubah-ubah. Contoh kasus nya adalah permainan catur dan *self-driving car*.

## prediksi

Prediksi adalah seni dan ilmu untuk memprediksi kejadian di masa depan dengan melibatkan pengambilan data historis dan memproyeksikannya ke masa mendatang dengan model pendekatan sistematis [11]. Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

## *feature engineering*

*Feature engineering* adalah proses menggunakan domain untuk mengekstrak fitur, menguji coba *trial* and *error*, dan mengevaluasi model. *Feature* engineering menjadi salah satu langkah yang cukup kompleks dan cukup menghabiskan waktu karena merupakan tugas utama dalam fase *data preparation*, menemukan fitur yang sesuai adalah unsur penting dalam mempelajari model prediksi yang baik [12].

Menurut Sitorus, Rizal, dan Jajuli (2020) [14] *feature engineering* merupakan aktivitas mengekstraksi fitur dari data mentah dan mengubahnya ke dalam format yang sesuai dengan model *machine learning* yang dibuat. *Feature engineering* juga melibatkan penerapan fungsi transformasi seperti agregat dan operator aritmatika pada fitur yang diberikan untuk menghasilkan fitur yang baru. Transformasi ini membantu menskalakan fitur mengkonversi relasi non-linear antara fitur dan kelas target menjadi hubungan linear yang lebih mudah untuk dipelajari.

## *extreme gradient boosting (xgboost)*

*eXtreme Gradient Boosting (XGBoost)* adalah salah satu algoritma *boosting* implementasi dari pohon keputusan gradien yang dirancang untuk performa dan kecepatan. *XGBoost* digunakan untuk permasalahan *supervised learning* yang menggunakan data latih yang mana sebagai variabel independen untuk memprediksi variabel dependen [15].

*XGBoost* adalah alat yang sangat skalabel, fleksibel, dan serbaguna. Metode ini dirancang untuk eksploitasi *resources* secara benar dan untuk mengatasi keterbatasan *gradient boosting* sebelumnya. Perbedaan utama antara *XGBoost* dan *gradient boosting* lainnya adalah menggunakan teknik regularisasi yang baru untuk mengontrol *overfitting*. *Overfitting* adalah kondisi yang mana data digunakan untuk pelatihan adalah data yang “terbaik” tetapi apabila dilakukan tes dengan menggunakan data yang berbeda, akan mengurangi akurasi nya. Kunci kecepatan komputasi *XGBoost* terletak pada optimasi *cache* pada komputer dan penggunaan memori sehingga dapat bekerja dengan efisien walaupun dengan data berukuran besar. Oleh karena itu, *XGBoost* lebih cepat dan lebih kuat dalam penyetelan dan membuat model. [16].

Berdasarkan pemaparan diatas, *XGBoost* dapat diformulasikan secara matematika sebagai berikut:

Dimana adalah nilai prediksi, adalah himpunan yang mencakup semua pohon regresi, adalah salah satu dari pohon regresi, dan adalah jumlah dari pohon regresi. Nilai prediksi diharapkan sedekat mungkin dengan nilai sebenarnya . Dalam *XGBoost* terdapat fungsi objektif, fungsi *training loss*, dan *regularization*. Fungsi objektif pada *XGBoost* adalah sebagai berikut:

Dimana adalah *loss function* dan adalah *regularization*. *Loss function* berguna untuk merepresentasikan perbedaan antara nilai prediksi dan nilai aktual . *Loss function* dirumuskan sebagai berikut:

Kemudian adalah *regularization* yang mendefinisikan kompleksitas dari model. Semakin kecil nilai , maka semakin rendah kompleksitas dan semakin kuat kemampuan generalisasi. Rumus untuk menghitung adalah sebagai berikut:

Dimana dan adalah koefisien konstan, adalah jumlah simpul daun, dan adalah representasi skor dari simpul daun. Dalam menggunakan *XGBoost*, terdapat beberapa parameter yang dapat diatur dan disesuaikan nilainya agar mendapat hasil yang lebih baik.

## *mean absolute error (mae)*

*Mean Absolute Error* (MAE) merupakan salah satu indikator evaluasi dari model regresi yang mengukur nilai rata-rata selisih mutlak nilai sebenarnya (aktual) dengan nilai prediksi [17]. Nilai MAE dapat dikatakan baik apabila nilai nya semakin rendah atau mendekati nol, MAE dirumuskan sebagai:

Dengan keterangan adalah jumlah data, adalah nilai data aktual, dan adalah nilai data prediksi.

## koefisien determinasi

Koefisien determinasi atau lebih sering disebut *R Squared* () adalah pengukuran proporsi variasi satu variabel (variabel objektif atau respon) yang dijelaskan oleh variabel lain (variabel penjelas) dalam regresi. Pengukuran ini yang banyak dilakukan dalam regresi [18]. Pengujian dilakukan dengan atau koefisien determinasi yaitu melihat seberapa besar pengaruh yang diberikan variabel X secara simultan terhadap variabel Y. dirumuskan sebagai:

Dengan keterangan RSS adalah *Residuals Sum of Squares* atau kuadrat dari selisih nilai Y prediksi dengan nilai rata-rata Y dan TSS adalah *Total Sum of Squares* atau kuadrat dari selisih nilai Y aktual dengan nilai rata-rata Y. Nilai berkisar dari angka nol sampai satu, semakin mendekati satu maka semakin besar pengaruh yang dihasilkan. Hal ini dikarenakan RSS memilik nilai antara 0 dan jumlah kuadrat Y, maka koefisien determinasi dapat bernilai dari nol hingga satu, atau 0% hingga 100%.

# metodologi penelitian

## pengambilan dan pengumpulan data

Pada penelitian ini, data yang digunakan berasal dari *database control tower* yang didapatkan dari perusahaan tempat peneliti magang. Data ini merupakan data transaksi dan data UMKM yang terdaftar dari salah satu instansi tahun 2019-2021 sebanyak 48.259 data transaksi dan 3.076 data UMKM. Data UMKM hanya digunakan sebagai informasi tambahan untuk mengetahui jumlah UMKM yang terdaftar pada tahun 2019-2021 dan tidak mempengaruhi proses penelitian. Adapun data transaksi pada tahun 2019 sebanyak 93 data, tahun 2020 sebanyak 10.288 data, dan tahun 2021 sebanyak 37.878 data. Sedangkan untuk data UMKM pada tahun 2019 sebanyak 0 data, tahun 2020 sebanyak 2.165 data, dan tahun 2021 sebanyak 911 data. Rincian dan frekuensi pada data transaksi dan data UMKM dapat dilihat pada Tabel 1.

Tabel 1. Frekuensi Jumlah Data Transaksi dan UMKM 2019-2021

|  |  |  |
| --- | --- | --- |
| Tahun | Data (Frekuensi) | |
| Transaksi | UMKM |
| 2019 | 93 | 0 |
| 2020 | 10.288 | 2.165 |
| 2021 | 37.878 | 911 |

Selain itu, pada data transaksi terdapat atribut yang berjumlah 28 atribut seperti pada Tabel 2.

Tabel 2. Atribut Data Transaksi 2019-2021

|  |  |  |
| --- | --- | --- |
| No | Atribut | Keterangan |
| 1. | trx\_date | Tanggal transaksi |
| 2. | trx\_id | ID transaksi |
| 3. | bumnid | BUMN yang bertransaksi |
| 4. | project\_name | Nama proyek yang di beli oleh BUMN |
| 5. | project\_category | Kategori proyek |
| 6. | total\_project\_value | Total nilai proyek |
| 7. | type\_project\_value | Tipe nilai proyek |
| 8. | uid | ID UMKM |
| 9. | umkm\_name | Nama UMKM |
| 10. | umkm\_category | Kategori UMKM |
| 11. | umkm\_classification | Klasifikasi UMKM |
| 12. | pengampu | Pengampu UMKM |
| 13. | type\_pengampu | Tipe pengampu |
| 14. | provinsi | Provinsi *seller* |
| 15. | kota | Kota *seller* |
| 16. | buyer\_id | ID *buyer* |
| 17. | status | Status transaksi |
| 18. | users | User UMKM |
| 19. | post\_timestamp | Post tanggal transaksi di *database* |
| 20. | update\_timestamp | Update post tanggal transaksi di *database* |
| 21. | cluster | Klasterisasi BUMN |
| 22. | batch | *Batch* transaksi |
| 23. | bumn\_code | Kode BUMN |
| 24. | buyer\_group\_provinsi | Provinsi *buyer group* |
| 25. | purchase\_order\_date | Tanggal order transaksi |
| 26. | payment\_group | Grup pembayaran |
| 27. | payment\_term | Ketentuan pembayaran |
| 28. | payment\_method | Metode pembayaran |

## *Preprocessing*

Pada tahapan ini dilakukan pembersihan data agar data bebas dari *missing value*, *outlier*, dan juga penghapusan data yang tidak relevan. Pembersihan data ini dilakukan secara manual dengan mengecek ada atau tidak nya *missing value*, kemudian menghapus kolom yang tidak relevan terhadap penelitian. Dari *preprocessing* yang telah dilakukan, total kolom atau atribut yang sebelumnya berjumlah 28 menjadi 7 kolom. Adapun 7 kolom tersebut yaitu trx\_date, bumnid, project\_category, total\_project\_value, type\_project\_value, umkm\_category, dan umkm\_classification. Tahapan *preprocessing* juga dilakukan pemaparan data yang mengambil referensi dari Gambar 1 lalu beberapa diolah kembali dengan menampilkan data khusus perusahaan XYZ. Pemaparan ini hanya sebagai informasi tambahan analisis dari data yang diambil dan tidak masuk kedalam proses pembangunan model prediksi. Proses ini dijelaskan mengenai total transaksi, total UMKM yang terdaftar, total nilai transaksi, plot total transaksi, plot total transaksi berdasarkan tipe nilai proyek, plot total transaksi berdasarkan kategori UMKM, 10 transaksi terbanyak berdasarkan kategori, dan tabulasi. Adapun pemaparan dari data tersebut adalah sebagai berikut:

* Total transaksi, menampilkan jumlah transaksi perusahaan XYZ pada *platform* PADI UMKM yang berjumlah 48.259 transaksi seperti pada Tabel 3.

Tabel 3. Total Transaksi Perusahaan XYZ

|  |  |
| --- | --- |
| BUMNID | Total Transaksi |
| XYZ | 48.259 |

* Total UMKM terdaftar, menampilkan jumlah UMKM yang terdaftar perusahaan XYZ pada *platform* PADI UMKM yang berjumlah 3.076 UMKM seperti pada Tabel 4.

Tabel 4. Total Transaksi Perusahaan XYZ

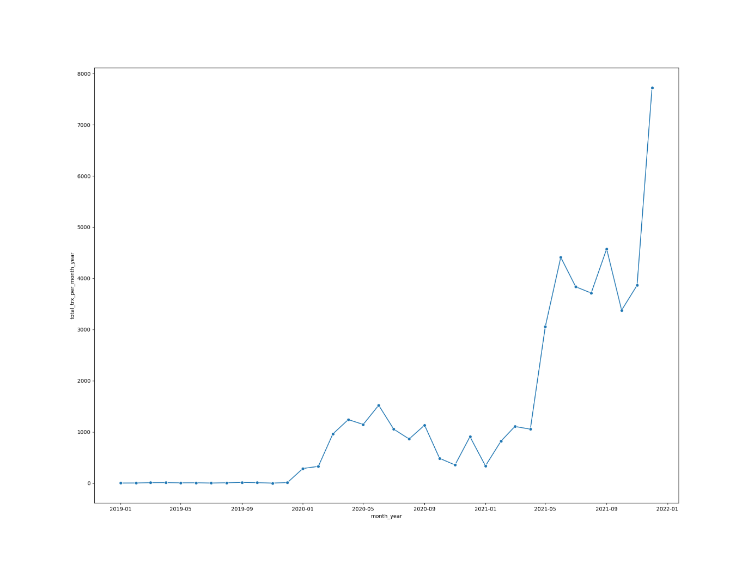
|  |  |
| --- | --- |
| BUMNID | Jumlah UMKM Terdaftar |
| XYZ | 3.076 |

* Total nilai transaksi, menampilkan total nilai transaksi perusahaan XYZ pada *platform* PADI UMKM yang berjumlah Rp.1.878.558.906.502 seperti pada Tabel 5.

Tabel 5. Total Transaksi Perusahaan XYZ

|  |  |
| --- | --- |
| BUMNID | Total Nilai Transaksi |
| XYZ | Rp.1.878.558.906.502 |

* Plot tren total transaksi, menampilkan grafik berupa *line chart* total transaksi perusahaan XYZ menunjukkan pola yang terus meningkat dari bulan ke bulan seperti pada Gambar 2. Detail total transaksi dapat dilihat pada Tabel 6.

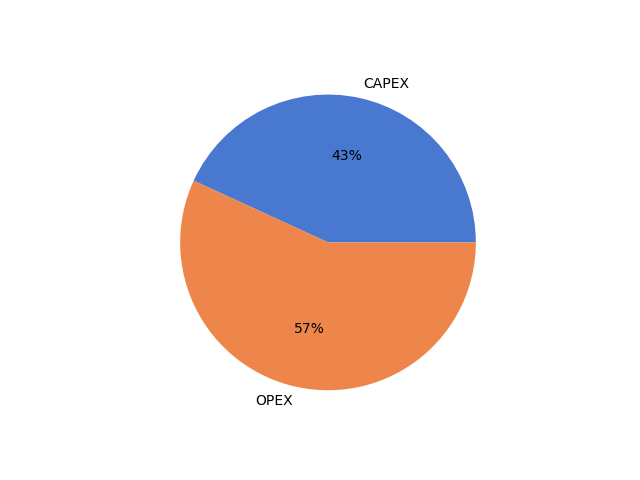


Gambar 2. Plot Total Transaksi Perusahaan XYZ

Tabel 6. Detail Total Transaksi per Bulan Perusahaan XYZ

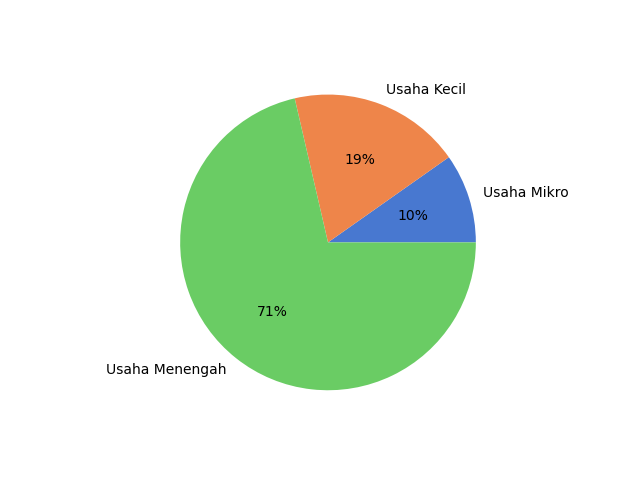
|  |  |  |
| --- | --- | --- |
| Tahun | Bulan | Total Transaksi per Bulan |
| 2019 | Januari | 3 |
| Februari | 4 |
| Maret | 10 |
| April | 11 |
| Mei | 7 |
| Juni | 8 |
| Juli | 3 |
| Agustus | 8 |
| September | 15 |
| Oktober | 10 |
| November | 2 |
| Desember | 12 |
| 2020 | Januari | 287 |
| Februari | 328 |
| Maret | 960 |
| April | 1242 |
| Mei | 1147 |
| Juni | 1522 |
| Juli | 1056 |
| Agustus | 863 |
| September | 1133 |
| Oktober | 486 |
| November | 356 |
| Desember | 908 |
| 2021 | Januari | 338 |
| Februari | 819 |
| Maret | 1108 |
| April | 1055 |
| Mei | 3057 |
| Juni | 4411 |
| Juli | 3835 |
| Agustus | 3713 |
| September | 4575 |
| Oktober | 3378 |
| November | 3864 |
| Desember | 7725 |

* Plot total transaksi berdasarkan tipe nilai proyek, menampilkan persentase berupa *pie chart* transaksi dua tipe proyek, *Capital Expenditure* (CAPEX) yaitu pengeluaran yang dilakukan perusahaan bertujuan untuk menambah nilai aset. *Operating Expenditure* (OPEX) yaitu pengeluaran yang dilakukan perusahaan untuk memenuhi kebutuhan operasional. Total transaksi masing-masing tipe proyek berjumlah 20.819 dan 27.440 dan persentase ditunjukkan seperti pada Gambar 3.



Gambar 3. Persentase Total Transaksi Berdasarkan Tipe Proyek

* Plot total transaksi berdasarkan kategori UMKM, menampilkan persentase berupa *pie chart* dari setiap kategori UMKM yaitu mikro, kecil, dan menengah. Total transaksi dari masing-masing kategori berjumlah 4.716, 9.093, dan 34.446 yang ditunjukkan seperti pada Gambar 4.



Gambar 4. Persentase Total Transaksi Berdasarkan Kategori UMKM

* 10 kategori transaksi terbanyak, menampilkan 10 kategori teratas transaksi berdasarkan nilai transaksi terbanyak. Kategori teratas yaitu jasa konstruksi dan renovasi dengan total transaksi sebanyak 15.488 transaksi dan jumlah nilai transaksi sebanyak Rp.907.235.781.343. Adapun detail 10 kategori dengan transaksi terbanyak ditunjukkan pada Tabel 7.

|  |  |  |
| --- | --- | --- |
| project\_category | total\_trx\_umkm\_category | total\_value\_category |
| Jasa Konstruksi & Renovasi | 15.488 | Rp.907.235.781.343 |
| Material Konstruksi | 7.057 | Rp.455.516.274.389 |
| Jasa *Event Organizer* | 1.495 | Rp.98.793.586.357 |
| Barang Elektronik, Komputer, & Periferal | 3.794 | Rp.87.552.461.861 |
| *Souvenir* & *Merchandise* | 3.386 | Rp.59.914.567.377 |
| Alat & Jasa Kesehatan-Keselamatan | 2.700 | Rp.38.125.971.924 |
| Alat Tulis Kantor | 6.503 | Rp.29.596.169.546 |
| Jasa Perawatan Elektronik & IT | 518 | Rp.29.321.858.455 |
| Jasa *Advertising* | 711 | Rp.27.854.583.637 |
| Barang dan Jasa Lainnya | 129 | Rp.23.722.115.843 |

* Tabulasi, penyajian data tiga kategori UMKM yang menampilkan nilai rata-rata, *min*, dan *max* dari total nilai transaksi. Tabulasi dapat dilihat pada Tabel 8.

|  |  |  |  |
| --- | --- | --- | --- |
| umkm\_category | total\_project\_value | | |
| Mean | Min | Max |
| Kecil | 1,344164e+07 | 3,300000e+03 | 8,714960e+08 |
| Menengah | 4,830605e+07 | -2,078367e+09 | 4,788000e+09 |
| Mikro | 1,958465e+07 | -2,095456e+09 | 1,790000e+09 |

## *Feature Engineering*

Pada dasarnya, semua algoritma pembelajaran mesin menggunakan fitur yang diekstraksi dari data masukan untuk membuat data keluaran. Untuk menyiapkan data masukan yang tepat, kompatibel, dan meningkatkan kinerja model pembelajaran mesin, perlu menerapkan teknik *feature engineering* untuk mendapat keluaran yang diharapkan. Teknik *feature engineering* yang dilakukan adalah sebagai berikut:

* Mengubah tipe data pada atribut trx\_date yang sebelumnya berupa tipe data *object* menjadi tipe data *datetime* menggunakan *library pandas*.
* Menambah atribut month\_year yaitu data per bulan dari tahun 2019-2021 yang didapatkan dari atribut trx\_date menggunakan *period* dalam *library pandas*.
* Menambah atribut *year* dan *month* dengan cara mengambil data dari atribut trx\_date lalu mengambil format *year* dan *month* secara terpisah sebagai atribut baru menggunakan *library datetime*.
* Mengganti *value* beberapa data kategori pada atribut umkm\_category karena menyebabkan *redundancy* menggunakan *library pandas*.
* Menambah atribut sum\_total\_value\_per\_month\_year yaitu atribut untuk mengetahui jumlah nilai transaksi yang diurutkan berdasarkan bulan dan tahun. Atribut tersebut diambil dari atribut month\_year dan total\_project\_value menggunakan fungsi *group by* dari *library pandas*.
* Menambah atribut total\_trx\_per\_month\_year yaitu atribut untuk mengetahui total transaksi yang diurutkan berdasarkan bulan dan tahun. Atribut tersebut diambil dari atribut month\_year dan total\_project\_value menggunakan fungsi *group by* dari *library pandas*.
* Menambah atribut total\_umkm\_category yaitu atribut untuk mengetahui total jumlah UMKM per kategori. Atribut tersebut diambil dari atribut project\_category menggunakan fungsi *group by* dari *library pandas*.
* Menambah atribut total\_value\_category yaitu atribut untuk mengetahui total nilai transaksi UMKM per kategori. Atribut tersebut diambil dari atribut project\_category dan total\_project\_value menggunakan fungsi *group by* dari *library pandas*.
* Mengubah atribut type\_project\_value, project\_category, umkm\_category, dan umkm\_classification dengan melakukan *encoding* menjadi kategorikal numerik. Dari *encoding* tersebut, didapatkan empat atribut tambahan yaitu type\_project\_value\_code, project\_category\_code, umkm\_category\_code, dan umkm\_classification\_code.

Dari *feature engineering* yang telah dilakukan, terdapat penambahan 10 atribut sehingga total atribut berjumlah 17.

## Pembagian Data

Langkah utama dalam mengimplementasikan pembelajaran mesin dan penambangan data adalah memasukkan sejumlah data ke dalam algoritma yang kemudian melatihnya untuk memahami pola data. Setelah algoritma mempelajari polanya, seseorang harus memasukkan data lain untuk memeriksa tingkat pemahaman yang dilakukan oleh algoritma tersebut. Oleh karena itu membagi data menjadi dua himpunan bagian untuk tujuan pelatihan dan pengujian.

Pembagian data latih dan data uji dapat disesuaikan rasio nya untuk mendapatkan kinerja yang lebih baik. Pada penelitian ini pembagian dilakukan dengan memilih proporsi yang sering digunakan dalam penelitian serupa yaitu 80:20. Proporsi dilakukan dengan menguji 80% untuk data latih yang berjumlah 38.607 dan 20% untuk data uji yang berjumlah 9.652.

## Analisis Data

Data penjualan pada *platform* PADI UMKM memiliki banyak atribut seperti trx\_date, trx\_id, bumnid, project\_name, project\_category, total\_project\_value, type\_project, uid, umkm\_name, umkm\_classification, umkm\_category, provinsi, kota, dan status. Data dari setiap atribut ini akan dilakukan *preprocessing* agar dapat diolah. Kemudian data akan dilakukan uji coba dengan membagi data menjadi data latih dan data uji untuk menguji model mana yang akan digunakan untuk melakukan prediksi total transaksi. Proses ini juga dilakukan penentuan parameter X dan Y sesuai yang dibutuhkan pada penelitian ini. Selanjutnya membagi data latih dan data uji yaitu variabel x\_train, x\_test, y\_train, dan y\_test dengan proporsi 80:20 untuk masing-masing data latih dan data uji. Setelah membagi data proses berikutnya yaitu *tuning model XGBoost* dengan beberapa hiperparameter agar mendapatkan model terbaik.

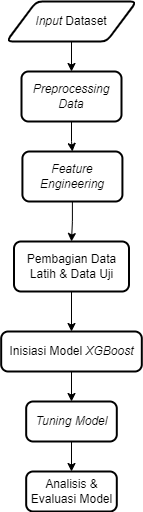
Dari hasil *preprocessing* yang dilakukan, didapatkan atribut sebanyak 17. Atribut tersebut akan dipilih sebagai variabel prediktor atau independen (X) dan variabel target atau dependen (Y). Untuk variabel yang dijadikan target atribut adalah total\_trx\_per\_month\_year atau total transaksi per bulan, variabel ini yang akan digunakan untuk memprediksi total transaksi. Adapun pemilihan atribut untuk dijadikan variabel prediktor menggunakan uji korelasi *pearson*. Uji korelasi *pearson* adalah pengujian korelasi yang digunakan untuk mengukur kekuatan hubungan dua variabel X dan Y sebagai acuan pemilihan fitur. Nilai koefisien korelasi *pearson* antara -1 sampai 1. -1 menunjukkan korelasi negatif, 0 berarti tidak ada korelasi linear antar dua variabel, dan +1 berarti terdapat korelasi positif (Rozy Fachrul dkk., 2018). Pengujian korelasi dilakukan perhitungan beberapa variabel X terhadap variabel Y untuk melihat nilai koefisien yang memiliki korelasi positif dan memilih atribut berdasarkan nilai *pearson* tertinggi. Adapun hasil uji korelasi *pearson* dapat dilihat pada Tabel 9.

Tabel 9. Hasil Uji Korelasi *Pearson*

|  |  |  |
| --- | --- | --- |
| Atribut | | |
|  | Total\_trx\_per\_month\_year | |
| *Month* | | 0,586097 |
| *Year* | | 0,694469 |
| total\_project\_value | | -0,058998 |
| total\_trx\_per\_month\_year | | 1.000000 |
| total\_trx\_umkm\_category | | 0,112172 |
| total\_value\_category | | 0,106778 |
| type\_project\_value\_code | | 0,026136 |
| project\_category\_code | | -0,107692 |
| umkm\_category\_code | | 0,141242 |
| umkm\_classification\_code | | -0,108786 |
| *Month* | | 0,586097 |

Pada Tabel 9, hasil menunjukkan bahwa korelasi tertinggi ada pada atribut *month* dan *year* yang masing-masing menghasilkan nilai koefisien korelasi sebesar 0,58 dan 0,69. Nilai tersebut menunjukkan terdapat korelasi positif terhadap variabel Y atau atribut total\_trx\_per\_month\_year. Oleh karena itu, atribut yang dipilih sebagai variabel X untuk menjadi prediktor adalah atribut *month* dan *year*. Untuk selanjutnya dilakukan skenario pengujian model agar didapatkan model terbaik dan melihat efektivitas metode *XGBoost* untuk melakukan prediksi regresi total transaksi.

## Pengujian Data



Gambar 5. Alur Eksperimen Pengujian Data

Pada Gambar 5 merupakan alur eksperimen pengujian data yang akan menentukan model untuk digunakan dalam prediksi regresi total transaksi. Tahap ini dilakukan beberapa skenario uji agar mendapat model terbaik. Adapun rincian tahapan pengolahan sampai pengujian diuraikan sebagai berikut:

* *Input Dataset*

*Dataset* penjualan yang akan diolah memiliki banyak atribut yang disatukan dalam satu file format *comma-separated values* (CSV).

* *Preprocessing Data*

Sebelum proses pembagian data latih dan data uji dilakukan, perlu adanya *preprocessing data* agar pengolahan data mendapatkan prediksi yang lebih akurat. Tahap *preprocessing* terdiri dari empat tahapan, yaitu *data cleaning* dengan mengisi data jika ada yang hilang dan menyelesaikan inkonsistensi yang ditemukan. Selanjutnya adalah *data integration* yang menyatukan data dengan representasi yang berbeda untuk membuat data lebih halus. Proses selanjutnya yaitu *data transformation* yang mana data akan dinormalisasikan dan digeneralisasikan. Proses terakhir adalah *data reduction*, yaitu data dalam jumlah yang besar akan direduksi agar meningkatkan efisiensi dan hasil analisis.

* *Feature Engineering*

Proses ini dilakukan pemilihan dan penambahan fitur dari dataset agar model yang dibuat dapat bekerja lebih akurat dalam memprediksi total transaksi. Strategi yang dilakukan terdiri dari *feature selection* yang merupakan proses pemilihan fitur yang relevan terhadap masalah yang akan diselesaikan. *Feature extraction* atau proses membuat fitur turunan dari fitur yang ada apabila fitur tersebut cukup sulit untuk diinterpretasikan. Kemudian *feature combination* yaitu proses menggabungkan beberapa fitur menjadi satu fitur agar model dapat dibuat sesederhana mungkin.

* Pembagian Data Latih dan Data Uji

Untuk kebutuhan evaluasi model yang akan dilatih perlu melakukan pembagian data latih dan data uji, proporsi yang dilakukan dengan membagi data latih sebanyak 80% dan data uji 20%. Selanjutnya inisiasi *random state* untuk melakukan pengacakan data latih dan data uji.

* Inisiasi Model *XGBoost* dan *Tuning* Model

Merancang model *XGBoost* dengan menggunakan *library XGBoost* dari *python*. Perancangan model dilakukan beberapa iterasi agar dapat mengetahui model mana yang cocok dengan dataset transaksi tersebut. *Tuning model* yang dilakukan dengan dua skenario, yaitu dengan menyetel parameter secara *default* dan penyetelan parameter menggunakan *GridSearchCV* dari *library sklearn*. Pengujian dengan *GridSearchCV* dilakukanuntuk mendapatkan beberapa parameter terbaik untuk data yang dimiliki. Kemudian dilakukan pengecekan kembali menggunakan *cross validation* yang merupakan fungsi bawaan dari *library* XGBoost untuk melihat apakah model baik, *overfitting*, atau *underfitting*.

* Analisis dan Evaluasi Model

Proses ini akan menampilkan plot *line chart* dari besarnya nilai error pada masing-masing model yang telah dibuat. Model yang memiliki nilai error terendah tersebut akan digunakan untuk melakukan prediksi regresi total transaksi.

## Interpretasi Data

Pada tahapan ini, peneliti terlebih dahulu menganalisis hasil yang diperoleh dari skenario pengujian data. Hasil tersebut dikaitkan dengan analisis dan teori dari studi literatur sebelumnya. Selanjutnya hasil tersebut juga dikaitkan dengan penelitian terdahulu untuk membandingkan hasil yang sudah diperoleh dengan hasil dari penelitian terdahulu. Kemudian yang terakhir peneliti akan menuliskan kesimpulan dan saran dari interpretasi hasil pengujian data yang dilakukan pada data total transaksi tersebut.

# hasil dan analisis

## Analisis

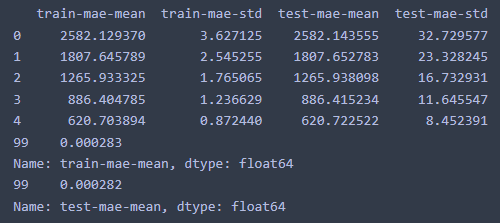
Pada tahap ini dilakukan analisis dari pengujian model yang telah dilakukan. Pengujian terdapat dua skenario yang dilakukan yaitu pengujian menggunakan semua parameter bawaan dari *library XGBoost* dan menggunakan *GridSearchCV* untuk menghasilkan parameter yang sesuai secara otomatis. Adapun tabel skenario beserta parameter yang digunakan setiap pengujian dapat dilihat pada Tabel 10.

Tabel 10. Parameter Pengujian Model Setiap Skenario

|  |  |  |
| --- | --- | --- |
| Parameter | Skenario Uji Model 1 | Skenario Uji Model 2 |
| *Default Parameter Values* | *GridSearchCV Parameters Values* |
| n\_estimators | 100 | 1000 |
| max\_depth | 6 | 3 |
| learning\_rate | 0,3 | 0,05 |
| reg\_lambda | 1 | 1 |
| reg\_alpha | 0 | 1 |
| gamma | 0 | 0 |
| min\_child\_weight | 1 | 0,5 |
| early\_stopping\_round | *None* | 10 |
| n\_estimators | 100 | 1000 |
| max\_depth | 6 | 3 |
| learning\_rate | 0,3 | 0,05 |

Pada skenario pengujian model satu, telah dilakukan pembuatan model yang menggunakan parameter bawaan *XGBoost*. Hasil dari skenario tersebut lalu dievaluasi untuk mengecek seberapa baik model tersebut. Evaluasi kedua skenario uji model tersebut dilakukan menggunakan *Mean Absolute Error* (MAE) dan R Square ().

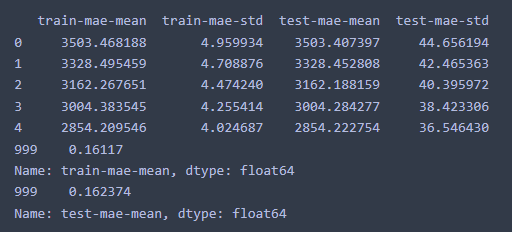
Hasil evaluasi nilai MAE pada skenario uji model satu untuk data latih dan data uji masing-masing bernilai 0,000295 dan 0,000293. Sedangkan evaluasi menggunakan pada data latih dan data uji masing-masing menunjukkan nilai yang sama yaitu 0,99. Dapat dilihat jika pemodelan dilakukan tanpa menyetel parameter menghasilkan nilai error yang sangat rendah. Untuk mengecek apakah model *overfitting*, *underfitting*, atau baik dilakukan *cross validation* dengan jumlah *fold* sebanyak 10 dan iterasi sebanyak 100. Dari validasi tersebut didapatkan hasil error untuk data latih dan data uji masing-masing bernilai 0,000283 dan 0,000282 pada iterasi ke-99 seperti ditunjukkan pada Gambar 6.



Gambar 6. Hasil *Cross Validation* Skenario Uji Model 1

Berdasarkan Gambar 6, nilai error dari data latih dan data uji yang dihasilkan *cross validation* terdapat selisih sebesar 0,000012 dan 0,000011. Makna dari nilai menggunakan *cross validation* menujukkan bahwa model dapat dikatakan cukup baik.

Pada skenario uji model dua, hasil evaluasi nilai MAE yaitu sebesar 0,1353 dan 0,1308 untuk masing-masing data latih dan data uji. Sedangkan evaluasi menggunakan pada data latih dan data uji masing-masing juga menunjukkan nilai yang sama yaitu 0,99. Nilai error yang dihasilkan sedikit lebih tinggi daripada skenario uji model pertama, pada data latih selisih nya sebesar 0,135005 dan data uji sebesar 0,130507. Validasi juga dilakukan untuk melihat baik tidak nya model dengan *fold* berjumlah 10 dan iterasi sebanyak 1000. Dari validasi tersebut didapatkan hasil error untuk data latih dan data uji masing-masing bernilai 0,16117 dan 0,162374 pada iterasi ke-999 seperti ditunjukkan pada Gambar 7.



Gambar 7. Hasil *Cross Validation* Skenario Uji Model 2

Berdasarkan Gambar 7, nilai error dari data latih dan data uji yang dihasilkan *cross validation* terdapat selisih sebesar 0,02587 dan 0,031574. Makna dari nilai menggunakan *cross validation* menujukkan bahwa model dapat dikatakan sangat baik. Hal ini dikarenakan nilai validasi error untuk data latih dan data uji sedikit lebih besar daripada nilai error untuk data latih dan data uji.

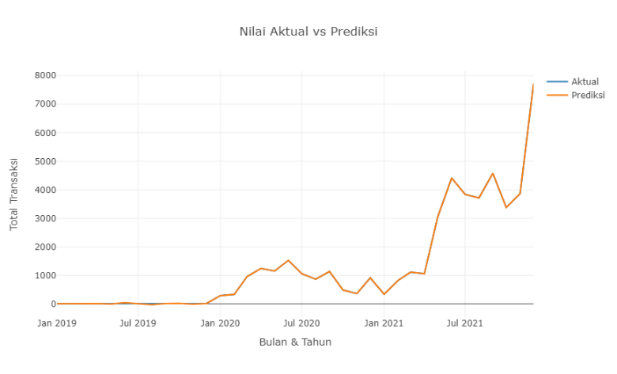
## Pembahasan

Pada proses sebelumnya telah melakukan dua skenario uji model dengan mencoba parameter bawaan dan juga parameter terbaik yang telah dihasilkan oleh *GridSearchCV*. Berdasarkan skenario uji model yang telah dibuat, didapatkan model terbaik yaitu pada skenario uji model dua. Pemodelan pada skenario tersebut walaupun menghasilkan nilai *error* sedikit dibandingkan skenario uji model satu tetapi model dapat dikatakan lebih baik dari skenario uji model dua. Nilai MAE pada data latih dan data uji masing-masing menghasilkan nilai 0.1353 dan 0.1308, maksud nilai ini menunjukkan antara nilai aktual dengan nilai prediksi memiliki selisih yang sangat rendah. Sedangkan evaluasi menggunakan pada data latih dan data uji masing-masing menunjukkan nilai yang sama yaitu 0,99, nilai ini bermakna bahwa atribut *month* (X1) dan *year* (X2) secara simultan atau bersama-sama berpengaruh terhadap atribut total\_trx\_per\_month\_year (Y) sehingga dapat dikatakan pengaruh yang dihasilkan sangat signifikan. Berdasarkan uji model tersebut, selisih nilai aktual dan nilai prediksi tergolong sangat kecil. Interpretasi perbandingan nilai aktual dan nilai prediksi disajikan dalam bentuk tabel dan visualisasi grafik seperti pada Tabel 11 dan Gambar 8.

Tabel 11. Hasil Perbandingan Nilai Aktual dan Prediksi Uji Model 2

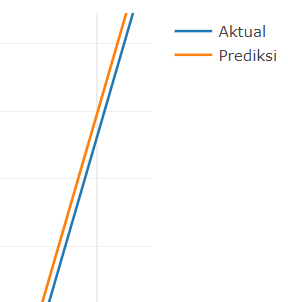
|  |  |  |  |
| --- | --- | --- | --- |
| Tahun | Bulan | Nilai Aktual | Nilai Prediksi |
| 2019 | Januari | 3 | 5.22 |
| Februari | 4 | 7.68 |
| Maret | 10 | 2.93 |
| April | 11 | 14.47 |
| Mei | 7 | -5.75 |
| Juni | 8 | 41.34 |
| Juli | 3 | 3.81 |
| Agustus | 8 | -32.17 |
| September | 15 | 13.13 |
| Oktober | 10 | 16.56 |
| November | 2 | -11.44 |
| Desember | 12 | 12.28 |
| 2020 | Januari | 287 | 286.86 |
| Februari | 328 | 327.84 |
| Maret | 960 | 960.43 |
| April | 1242 | 1241.75 |
| Mei | 1147 | 1147.41 |
| Juni | 1522 | 1521.88 |
| Juli | 1056 | 1055.65 |
| Agustus | 863 | 863.05 |
| September | 1133 | 1133.24 |
| Oktober | 486 | 484.98 |
| November | 356 | 356.85 |
| Desember | 908 | 907.96 |
| 2021 | Januari | 338 | 338.06 |
| Februari | 819 | 819.07 |
| Maret | 1108 | 1107.68 |
| April | 1055 | 1055.21 |
| Mei | 3057 | 3056.89 |
| Juni | 4411 | 4410.97 |
| Juli | 3835 | 3835.12 |
| Agustus | 3713 | 3713.01 |
| September | 4575 | 4574.92 |
| Oktober | 3378 | 3378.15 |
| November | 3864 | 3863.93 |
| Desember | 7725 | 7724.99 |

Pada Tabel 11 nilai prediksi yang dihasilkan dari metode *XGBoost* menunjukkan nilai yang beragam. Nilai prediksi tersebut ada yang lebih tinggi dari nilai aktual, ada yang lebih rendah dari aktual, dan bahkan terdapat nilai yang hampir sama dengan nilai aktual. Hal ini disebabkan karena model regresi pada metode *XGBoost* menghasilkan nilai residu yang berbeda-beda di setiap iterasi nya. Dengan nilai residu atau error yang berbeda tersebut, maka setelah dilakukan perhitungan didapatkan nilai prediksi yang kemungkinan hasilnya akan lebih tinggi, lebih rendah, atau bahkan hampir sama dengan nilai aktual. Berdasarkan nilai tersebut bukan berarti model yang dihasilkan tidak baik, melainkan hal ini dilakukan *XGBoost* untuk menghindari model *overfitting* atau *underfitting*. Dengan pemilihan parameter menggunakan *GridSearchCV*, metode *XGBoost* melakukan regularisasi dan batasan agar mencegah model dari *overfitting* atau *underfitting* sehingga didapatkan model yang baik.



Gambar 8. Plot Visualisasi Perbandingan Nilai Aktuan dan Prediksi Uji Model 2

Pada Gambar 8 dapat dilihat bahwa garis berwarna biru yang mendefinisikan nilai aktual dapat dikatakan hampir menempel dengan nilai prediksi. Tetapi jika dilihat lebih dekat lagi perbedaan nilai tersebut tetap terlihat walaupun selisih jarak nya sangat kecil dan dapat dilihat pada Gambar 9. Hal ini juga ditandai pada Tabel 11bahwa semua nilai prediksi yang dihasilkan terdapat perbedaan dengan nilai aktual.



Gambar 11. Perbedaan Garis Nilai Aktual dan Prediksi Uji Model 2

Berdasarkan hasil tersebut, hal ini dibuktikan dengan beberapa penelitian terdahulu yang menggunakan metode serupa. Pada penelitian yang dilakukan Dairu dan Shilong (2021), nilai RMSSE yang dihasilkan sebesar 0,655. Sedangkan pada penelitian yang dikerjakan oleh Ranjitha dan Spandana (2021), nilai MAE yang dihasilkan sebesar 0,029. Penelitian yang dilakukan Zhang dkk (2021), MAE yang dihasilkan dari dua *dataset* berbeda masing-masing sebesar 0,178 dan 0,046. Metode *XGBoost* terbukti menghasilkan nilai *error* terendah yang didukung dengan beberapa penelitian terdahulu yang telah dijelaskan sebelumnya dan dapat dikatakan juga model dari metode *XGBoost* bisa diimplementasikan untuk prediksi regresi total transaksi.

# penutup

## kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan pengujian yang dilakukan, pembuatan model menggunakan *XGBoost* memberikan kinerja yang baik pada penelitian ini. Hal ini ditandai dengan hasil MAE dan yang diberikan masing-masing sebesar 0,1308, dan 0,99.
2. Evaluasi performa model terbaik ada pada skenario uji model dua dengan nilai MAE pada data latih dan data uji masing-masing menghasilkan nilai sebesar 0,1353 dan 0,1308. Sedangkan evaluasi menggunakan pada data latih dan data uji masing-masing menunjukkan nilai yang sama yaitu 0,99.

## saran

Berdasarkan hasil yang diperoleh yang dilakukan selama penelitian ini, peneliti memberikan saran kepada peneliti selanjutnya dalam bidang dan objek penelitian yang sama sebagai berikut:

1. Penelitian selanjutnya dapat menambah analisis prediksi transaksi dengan deret waktu dapat dilihat per hari, per minggu, per bulan, dan per tahun sehingga prediksi selanjutnya dapat lebih informatif.
2. Pengujian model yang dilakukan dapat menambah parameter yang lebih bervariasi lagi jika menggunakan metode serupa.
3. Analisis dan visualisasi yang dilakukan masih sangat sederhana, maka perlu dikembangkan kembali. Untuk peneliti selanjutnya dapat mengembangkan keluaran yang dihasilkan dalam bentuk sistem atau *dashboard* agar informasi yang ditampilkan mempermudah pembaca untuk memahaminya.

# referensi

[1] Alpaydin Ethem, “Introduction to Machine Learning,” 2020.

[2] H. S. R. Rajula, G. Verlato, M. Manchia, N. Antonucci, and V. Fanos, “Comparison of conventional statistical methods with machine learning in medicine: Diagnosis, drug development, and treatment,” *Medicina (Lithuania)*, vol. 56, no. 9, pp. 1–10, Sep. 2020, doi: 10.3390/medicina56090455.

[3] D. Kurniawan, *Pengenalan Machine Learning dengan Python*. PT Elex Media Komputindo, 2020.

[4] N. Wayan Wardani, “Penerapan Data Mining Dalam Analytic CRM,” 2021. [Online]. Available: https://www.researchgate.net/publication/351776255

[5] Y. Niu, “Walmart Sales Forecasting using XGBoost algorithm and Feature engineering,” in *Proceedings - 2020 International Conference on Big Data and Artificial Intelligence and Software Engineering, ICBASE 2020*, Oct. 2020, pp. 458–461. doi: 10.1109/ICBASE51474.2020.00103.

[6] X. Dairu and Z. Shilong, “Machine Learning Model for Sales Forecasting by Using XGBoost,” in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering, ICCECE 2021*, Jan. 2021, pp. 480–483. doi: 10.1109/ICCECE51280.2021.9342304.

[7] T. Chen and T. He, “xgboost: eXtreme Gradient Boosting,” 2017.

[8] “Pengadaan Barang dan Jasa Pemerintah di Marketplace PaDi UMKM,” May 12, 2022. https://padiumkm.id/ (accessed May 12, 2022).

[9] “Frequently Asked Question,” May 12, 2022. https://control.padiumkm.id/faq (accessed May 12, 2022).

[10] Daqiqil Ibnu, “Machine Learning: Teori, Studi Kasus, dan Implementasi Menggunakan Python,” 2021.

[11] T. Indarwati, T. Irawati, and E. Rimawati, “PENGGUNAAN METODE LINEAR REGRESSION UNTUK PREDIKSI PENJUALAN SMARTPHONE,” *Jurnal Teknologi Informasi dan Komunikasi (TIKomSiN)*, vol. 6, no. 2, Jan. 2019, doi: 10.30646/tikomsin.v6i2.369.

[12] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, and D. Turaga, “Learning feature engineering for classification,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2017, vol. 0, pp. 2529–2535. doi: 10.24963/ijcai.2017/352.

[13] C. M. Sitorus, A. Rizal, and M. Jajuli, “Prediksi Risiko Perjalanan Transportasi Online Dari Data Telematik Menggunakan Algoritma Support Vector Machine,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2672.

[14] C. M. Sitorus, A. Rizal, and M. Jajuli, “Prediksi Risiko Perjalanan Transportasi Online Dari Data Telematik Menggunakan Algoritma Support Vector Machine,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, 2020, doi: 10.28932/jutisi.v6i2.2672.

[15] A. A. Firdaus, A. Komarudin, M. P. Statistika, F. Matematika, D. Ilmu, and P. Alam, “Klasifikasi Pemegang Polis Menggunakan Metode XGBoost,” 2021, doi: 10.29313/.v0i0.30320.

[16] Daoud Al Essam, “Comparison between XGBoost, LightGBM and  CatBoost Using a Home Credit Dataset,” 2019.

[17] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C. H. Lee, “On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression,” *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020, doi: 10.1109/LSP.2020.3016837.

[18] E. Kasuya, “On the use of r and r squared in correlation and regression,” *Ecological Research*, vol. 34, no. 1, pp. 235–236, Jan. 2019, doi: 10.1111/1440-1703.1011.