

Prediksi *Churn* Pelanggan B2B dengan Segmentasi Menggunakan Bisecting K-Means dan Long Short-Term Memory

Khairina Wardina¹, Ade Rahmat², Mohammad Syafrullah³

^{1,2}Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur, Indonesia

³Fakultas Teknologi Informasi, Universitas Budi Luhur, Indonesia

Article Info

Article history:

Received Dec 19, 2024

Revised Nov 10, 2025

Accepted Nov 12, 2025

Keywords:

Bisecting K-Means

Churn Customer

Davies-Bouldin Index

Long Short-Term Memory

Random Search

ABSTRACT (10 PT)

In the competitive Business-to-Business (B2B) sector, customer churn presents a critical challenge, particularly for Fast-Moving Consumer Goods (FMCG) distributors. Data from a North Sulawesi distributor revealed increasing churn rates from 2019 to 2022, attributing ineffective retention to limited data-driven analysis. This study addresses churn identification in a non-contractual B2B context by integrating Bisecting K-Means for segmentation and Long Short-Term Memory (LSTM) for prediction based on revenue. Using Length, Recency, Frequency, and Monetary (LRFM) attributes, the segmentation yielded three clusters with a Davies-Bouldin Index (DBI) of 0.46. The LSTM model demonstrated robust performance, achieving 96% validation accuracy, 95% test accuracy, and an AUC-ROC of 0.97. Findings reveal distinct behaviors: Cluster 1 exhibited the highest churn (100%), Cluster 2 moderate churn (16.31%), and Cluster 0 the lowest (3.21%), representing loyal customers. This integrated model effectively identifies churn-prone clients, facilitating targeted retention strategies to enhance B2B loyalty.

Corresponding Author:

Khairina Wardina,

Magister Ilmu Komputer,

Universitas Budi Luhur,

Jl. Ciledug Raya, Petungkang Utara, Jakarta Selatan, DKI Jakarta 12260

Email: 2211600057@student.budiluhur.ac.id

1. PENDAHULUAN

Dalam era bisnis yang semakin kompetitif, keterlibatan dan loyalitas pelanggan menjadi fokus penting bagi sektor *Business to Consumer* (B2C) maupun *Business to Business* (B2B) [1]. Digitalisasi dan tren *e-commerce* memengaruhi perilaku serta loyalitas pelanggan, sehingga menuntut perusahaan untuk beradaptasi secara cepat terhadap perubahan pasar [2]. Salah satu tantangan utama dalam retensi pelanggan B2B adalah *churn*, yaitu berhentinya pelanggan dari perusahaan [3]. Fenomena tersebut juga menjadi perhatian di industri distribusi *Fast Moving Consumer Goods* (FMCG), di mana hubungan jangka panjang dengan pelanggan menjadi faktor kunci dalam menjaga stabilitas pendapatan. Perusahaan distribusi atau yang sering disebut dengan distributor, adalah entitas yang memperoleh produk langsung dari produsen atau rantai distribusi awal dan menjualnya kembali kepada pengecer atau konsumen akhir [4].

Analisis data historis dari sebuah perusahaan distribusi FMCG yang beroperasi di Sulawesi Utara menunjukkan peningkatan *churn rate* sepanjang periode 2019 hingga 2022 seperti pada Tabel 1.

Tabel 1. *Churn Rate* Tahun 2019-2022

Tahun	Jumlah Pelanggan Aktif	Jumlah Pelanggan Berhenti	<i>Churn Rate</i>
2019	3986	305	7.65%
2020	5482	1046	19.08%
2021	5920	945	15.96%
2022	5877	1185	20.16%

Tabel 1 menunjukkan adanya peningkatan pelanggan yang berhenti bertransaksi setiap tahunnya. Pada tahun 2019, terdapat 305 pelanggan yang menghentikan transaksi dari total 3986 pelanggan aktif, menghasilkan *churn rate* sebesar 7.65%. Pada tahun 2020, angka *churn rate* meningkat tajam menjadi 19.08% dengan 1046 pelanggan berhenti dari total 5482 pelanggan aktif. Meskipun terjadi penurunan pada tahun 2021 dengan *churn rate* sebesar 15.96%, namun pada tahun 2022 angka *churn rate* kembali naik menjadi 20.16%.

Perusahaan telah berupaya menurunkan *churn rate* melalui strategi seperti harga jual alternatif dan promosi. Namun, kurangnya analisis pelanggan menyebabkan strategi ini kurang efektif, sehingga *churn rate* kembali meningkat pada 2022. Karena itu, perusahaan perlu mengevaluasi strategi dan menganalisis data pelanggan secara mendalam. Dalam interaksi dengan berbagai entitas bisnis seperti toko eceran, supermarket, dan grosir, kehilangan satu pelanggan saja berdampak besar pada pendapatan. Selain itu, Grönroos menyatakan bahwa biaya menarik pelanggan baru cenderung lebih tinggi dibandingkan mempertahankan pelanggan lama [5].

Salah satu strategi yang dapat diadopsi untuk mengatasi tantangan ini adalah dengan melakukan prediksi *churn* pelanggan. Prediksi *churn* merupakan proses identifikasi pelanggan yang cenderung beralih ke layanan atau produk lain. Definisi pelanggan yang beralih atau *churn* dapat bervariasi tergantung pada konteks bisnis [3]. Melalui klasifikasi pelanggan menjadi dua kategori, yaitu pelanggan yang terindikasi akan mengalami *churn* dan pelanggan yang tidak mengalami *churn*, perusahaan dapat mengidentifikasi secara proaktif pelanggan yang berpotensi untuk berhenti bertransaksi. Dengan demikian, perusahaan dapat merancang kebijakan retensi yang lebih tepat dan sesuai dengan kebutuhan pelanggan yang mungkin akan mengalami *churn*.

Penelitian sebelumnya telah mengkaji prediksi *churn* menggunakan berbagai pendekatan. Bach, Pivar, dan Jaković menerapkan pendekatan *hybrid* dengan K-Means dan Decision Tree pada dataset Telco Customer Churn dari Kaggle, menghasilkan prediksi *churn* yang efektif [6]. Xiahou dan Harada menggunakan kombinasi K-Means dan Support Vector Machine (SVM) pada data Alibaba Cloud Tianchi, dengan SVM menunjukkan akurasi, recall, dan precision yang baik setelah segmentasi [7]. Tran, Le, dan Nguyen mengevaluasi dampak segmentasi pelanggan terhadap akurasi prediksi *churn* dengan beberapa model *machine learning*, seperti K-NN, Logistic Regression, Decision Tree, SVM, dan Random Forest, dan menemukan Random Forest serta SVM memberikan precision tertinggi [8]. Wachwanakijkul dkk. juga menerapkan berbagai model *machine learning* dan deep learning seperti XGBoost, ANN, dan Transformer, dan menunjukkan bahwa model berbasis Transformer dan ANN paling efektif dalam memprediksi *churn* [9].

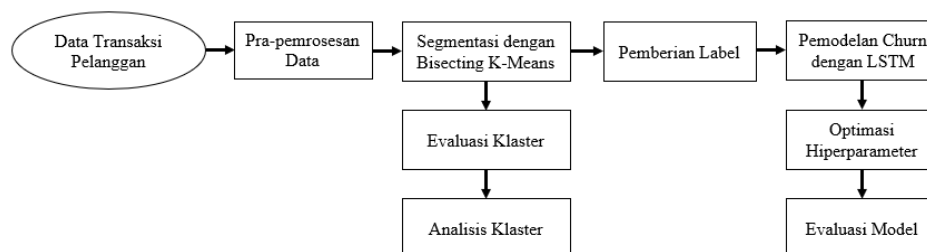
Namun, sebagian besar penelitian tersebut berfokus pada data pelanggan kontraktual, seperti sektor telekomunikasi dan layanan berlangganan, di mana status pelanggan lebih mudah diidentifikasi karena adanya kontrak atau langganan tetap. Kondisi ini berbeda dengan konteks pelanggan non-kontraktual, seperti pada industri distribusi B2B FMCG, di mana pelanggan tidak terikat kontrak formal dan perilaku *churn* lebih sulit diprediksi karena bergantung pada pola pembelian yang fluktuatif.

Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut dengan mengusulkan pendekatan yang mengintegrasikan Bisecting K-Means untuk segmentasi pelanggan berdasarkan atribut Length, Recency, Frequency, dan Monetary (LRFM), serta Long Short-Term Memory (LSTM) untuk memprediksi potensi *churn* berdasarkan perubahan pendapatan bulanan (*monthly revenue*) pelanggan. Pendekatan ini dirancang untuk konteks pelanggan B2B non-kontraktual di industri distribusi FMCG, yang memiliki karakteristik perilaku pembelian lebih kompleks dan fluktuatif dibandingkan pelanggan kontraktual.

Melalui integrasi segmentasi dan *deep learning*, penelitian ini diharapkan dapat memberikan wawasan yang lebih mendalam mengenai pola *churn* di tiap segmen pelanggan, membantu perusahaan dalam mengidentifikasi pelanggan berisiko secara lebih akurat, serta menjadi dasar dalam merancang strategi retensi yang adaptif, prediktif, dan berbasis data.

2. METODE

Model prediksi yang diusulkan dalam penelitian ini akan mengintegrasikan metode segmentasi dengan Bisecting K-Means dan prediksi *churn* menggunakan LSTM. LSTM akan dilatih untuk memproyeksikan *monthly revenue* dari setiap pelanggan per tahun. Tahapan yang dilakukan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan bahwa penelitian ini dimulai dengan data transaksi pelanggan yang melalui tahap pra-pemrosesan untuk memastikan data siap digunakan. Selanjutnya, data tersebut diolah menggunakan metode segmentasi Bisecting K-Means untuk mengelompokkan pelanggan ke dalam kluster-kluster tertentu. Hasil segmentasi dievaluasi menggunakan Davies-Bouldin Index (DBI) untuk menilai kualitas pengelompokan yang dihasilkan, diikuti dengan analisis kluster guna memahami karakteristik masing-masing kluster. Setelah itu, dilakukan pemberian label pada data berdasarkan hasil segmentasi untuk mendukung tahap pemodelan *churn*. Pada tahap akhir, model *churn* dibangun menggunakan pendekatan LSTM, yang diawali dengan optimasi hiperparameter menggunakan metode Random Search untuk meningkatkan performa model. Terakhir, model *churn* dievaluasi menggunakan confusion matrix dan AUC-ROC untuk memastikan akurasi serta keandalannya.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dan pembahasan dari penelitian yang dilakukan menggunakan Jupyter Notebook sebagai alat analisis. Python digunakan sebagai bahasa pemrograman utama untuk membantu proses analisis data, segmentasi pelanggan, dan pemodelan *churn*.

3.1. Pra-pemrosesan Data

Data yang dikumpulkan adalah data transaksi pelanggan yang berasal dari database Distribution Management System (DMS) yang digunakan oleh perusahaan. Data yang diambil berasal dari tabel salesInvoice, yang mencakup riwayat transaksi pelanggan mulai dari tanggal 2 Mei 2019 hingga 7 September 2023. Dalam rentang waktu tersebut, tercatat sebanyak 739.863 transaksi dari total 9.233 pelanggan. Tabel salesInvoice memiliki 129 kolom yang menyoroti berbagai informasi terkait dengan detail transaksi pelanggan, termasuk tanggal transaksi, area pelanggan, lokasi gudang, rincian diskon, serta nilai transaksi.

Penelitian ini akan menggunakan data transaksi pelanggan dalam rentang waktu dari Januari 2020 hingga Desember 2022. Rentang waktu tersebut dipilih untuk memastikan adanya data yang cukup untuk analisis tren dan pola perilaku pelanggan dalam beberapa tahun terakhir. Variabel penelitian diperoleh dari studi pustaka yang dilakukan dan disesuaikan dengan kebutuhan objek penelitian. Dari proses tersebut, variabel yang akan digunakan dalam penelitian adalah customerID, invoiceDate, dan totalNetAmount.

Pra-pemrosesan Data merupakan langkah kritis dalam memastikan kualitas data, melibatkan proses pembersihan, transformasi, dan integrasi data untuk memudahkan analisis [10]. Keputusan pra-pemrosesan data juga memiliki dampak besar pada kinerja model kecerdasan buatan karena memengaruhi karakteristik data masukan [11].

Pada penelitian ini, langkah pertama adalah *Data Reduction*, yang bertujuan untuk mengurangi dimensi atau kompleksitas data. Selanjutnya dilakukan *Data Cleaning*, yang mencakup pengecekan dan penanganan *missing value*, penghapusan nilai totalNetAmount yang bernilai 0 atau negatif, serta normalisasi pada atribut customerID. Setelah itu, dilakukan *Data Transformation*, yaitu mengubah data menjadi atribut LRFM dan *Monthly Revenue*, serta menerapkan normalisasi menggunakan metode Min-Max untuk memastikan skala data seragam. Tahapan terakhir adalah *Data Partition*, yaitu membagi data ke dalam subset yang sesuai untuk proses analisis dan pemodelan lebih lanjut.

3.1.1. Data Reduction

Dalam penelitian ini, *data reduction* dilakukan dengan menyeleksi variabel yang relevan, yaitu customerID, invoiceDate, dan totalNetAmount, yang diekstraksi dari database menggunakan query SQL. Dari proses tersebut, diperoleh 531.869 baris data dari 739.863 baris data yang kemudian diekspor ke dalam format CSV untuk meminimalkan gangguan pada data operasional perusahaan dan mempermudah pengolahan data selanjutnya. Pendekatan ini memastikan hanya data yang relevan digunakan, sehingga mengurangi volume data yang diproses dan meningkatkan efisiensi analisis.

3.1.2. Data Cleaning

Pada tahap *data cleaning*, data yang telah diekspor ke format CSV diolah menggunakan Python dengan Jupyter Notebook. Langkah pertama adalah memeriksa adanya *missing value* dalam dataset. *Missing value* dapat mempengaruhi analisis, sehingga setiap kolom diperiksa menggunakan metode `info()` dari `pandas` untuk melihat jumlah nilai yang hilang.

Setelah memastikan tidak ada *missing value*, langkah selanjutnya adalah menghapus baris dengan nilai `totalNetAmount` yang bernilai 0 atau negatif. Data ini dianggap sebagai transaksi yang dibatalkan atau barang bonus, yang tidak relevan dengan analisis *churn*. Hasilnya, jumlah baris dataset berkurang dari 531,869 menjadi 519,883, dengan 11,986 baris yang dihapus.

Langkah ketiga adalah normalisasi pada `customerID`, karena beberapa ID pelanggan tercatat dengan huruf kecil. Semua `customerID` diubah menjadi huruf kapital untuk memastikan konsistensi identifikasi pelanggan. Sebelum normalisasi, terdapat 8,672 pelanggan unik, namun setelah proses ini jumlah pelanggan unik turun menjadi 7,957, menunjukkan adanya penggabungan ID yang sebelumnya dianggap berbeda karena perbedaan huruf besar dan kecil.

3.1.3. Data Transformation

Transformasi data menjadi atribut LRFM dan Monthly Revenue

Proses ekstraksi dan transformasi data dimulai dengan mengonversi kolom `invoiceDate` menjadi format `datetime` untuk memudahkan manipulasi tanggal. Dari tanggal transaksi, tahun dan bulan diekstrak untuk mengelompokkan transaksi berdasarkan periode tersebut. Selain itu, ditambahkan kolom yang menunjukkan tanggal akhir tahun (31 Desember) untuk menghitung metrik `Recency` dan `Length`.

Selanjutnya, transaksi dikelompokkan berdasarkan `customerID` dan `transactionYear`. Metrik yang dihitung meliputi tanggal transaksi pertama dan terakhir, jumlah transaksi (`Frequency`), serta total nilai transaksi (`Monetary`). `Recency` dihitung berdasarkan jumlah hari sejak transaksi terakhir hingga akhir tahun, sedangkan `Length` dihitung dari durasi antara transaksi pertama dan akhir tahun. Kolom `FirstPurchaseDate` dan `LastPurchaseDate` kemudian dihapus karena sudah tidak diperlukan.

Untuk mengatur transaksi per bulan, dibuat indeks untuk setiap kombinasi `customerID`, `transactionYear`, dan `transactionMonth`, kemudian dihitung total nilai transaksi per bulan. Data ini dipivot sehingga setiap kolom mewakili bulan (`month1` hingga `month12`). Tabel yang berisi metrik LRFM digabungkan dengan tabel transaksi bulanan berdasarkan `customerID` dan `transactionYear`. Hasil akhirnya adalah dataset dengan 17,209 baris dan 18 kolom yang siap digunakan untuk analisis segmentasi pelanggan menggunakan `bisecting k-means` dan prediksi *churn* dengan model LSTM. Setiap baris mewakili satu pelanggan per tahun, dengan kolom-kolom yang menunjukkan metrik LRFM dan pendapatan bulanan.

Normalisasi dengan Min-Max

Normalisasi data adalah langkah pra-pemrosesan untuk menskalakan fitur ke skala yang sama, sehingga mencegah bobot tidak proporsional pada fitur dengan skala lebih besar, yang dapat meningkatkan kinerja algoritma pembelajaran mesin [12]. Proses normalisasi data menggunakan metode `Min-Max Normalization` dilakukan untuk memastikan bahwa data memiliki skala yang konsisten dan siap untuk analisis lebih lanjut. Normalisasi `Min-Max` ini mengubah nilai data sehingga berada dalam rentang 0 hingga 1. Proses normalisasi diterapkan dengan menggunakan fitur `MinMaxScaler` dari `sklearn.preprocessing`. Langkah pertama adalah menentukan kolom-kolom yang akan dinormalisasi, yaitu semua kolom kecuali `customerID` dan `transactionYear`. Setelah itu, `scaler` di-fit dan di-transform pada kolom-kolom tersebut.

3.1.4. Data Partition

Salah satu langkah penting dalam persiapan data adalah mempartisi data menjadi data training dan data testing. Tujuannya adalah melatih model dengan sebagian data dan menguji kinerjanya menggunakan data yang belum pernah dilihat oleh model. Setelah mendapatkan hasil cluster dari tahap segmentasi menggunakan `bisecting k-means`, data dibagi dengan perbandingan 70% untuk training dan 30% untuk testing menggunakan fungsi `train_test_split` dari pustaka `scikit-learn`. Fungsi ini membagi data secara acak namun konsisten dengan parameter `random_state`. Hasilnya, 12,046 baris data digunakan untuk training, sementara 5,163 baris digunakan untuk testing.

3.2. Segmentasi dengan Bisecting K-Means

Algoritma `bisecting k-means` adalah pengembangan dari algoritma `k-means` yang diperkenalkan oleh Savaresi & Boley [13]. Algoritma ini mengatasi beberapa kelemahan dari algoritma `k-means` tradisional dan lebih tangguh dalam menghadapi data. Perbedaan utama antara algoritma `bisecting k-means` dan `k-means` tradisional adalah pada setiap iterasi, algoritma `bisecting k-means` hanya perlu menangani satu pusat cluster

sesuai dengan distribusi sampel. Hal ini membuat algoritma bisecting k-means lebih efektif dalam mengurangi kemungkinan terjebak dalam optima lokal dan kurang sensitif terhadap gangguan atau noise [14].

Tahap ini mencakup penentuan nilai k optimal menggunakan Davies-Bouldin Index (DBI), segmentasi pelanggan dengan k optimal menggunakan algoritma bisecting k-means, dan analisis hasil klusterisasi. Proses ini bertujuan mengelompokkan pelanggan ke dalam kluster dengan karakteristik serupa berdasarkan atribut LRFM, guna memahami perilaku pelanggan di setiap kluster.

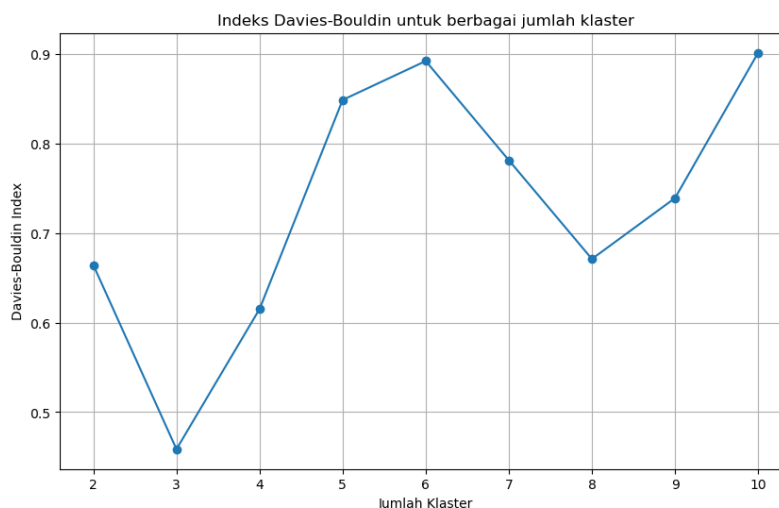
3.2.1. Menentukan Nilai k

Tahap awal segmentasi dengan bisecting k-means adalah menentukan nilai k optimal berdasarkan DBI, karena saat ini perusahaan belum memiliki segmentasi pelanggan yang jelas. Kluster optimal ditandai dengan pemisahan tinggi antar kluster serta homogenitas dan kekompakan dalam kluster. DBI digunakan untuk mengevaluasi kluster dengan prinsip tersebut [15]. Untuk menghitung DBI, perlu ditetapkan ukuran dispersi dalam kluster dan ukuran kesamaan antar kluster [16]. Pada penelitian ini iterasi dilakukan dengan nilai k antara 2 hingga 10, dan nilai k dengan DBI terendah dipilih sebagai yang terbaik, karena menunjukkan kluster yang lebih kompak dan terpisah dengan baik.

Tabel 2 dan Gambar 2 menampilkan hasil perhitungan DBI untuk masing-masing jumlah kluster, mulai dari 2 hingga 10, beserta grafiknya.

Tabel 2. Hasil DBI per Kluster

Jumlah Kluster	DBI
2	0.6642843816751438
3	0.45858169364487705
4	0.6152429023704591
5	0.848965315214197
6	0.8924524515865374
7	0.7815798139514667
8	0.6712051708495781
9	0.7392665376408534
10	0.9016332480975999



Gambar 2. Grafik DBI per Kluster

Dari Tabel 2 dan Gambar 2, dapat dilihat bahwa nilai DBI terendah diperoleh saat jumlah kluster (k) adalah 3, dengan nilai DBI sebesar 0.45858169364487705. Hal ini menunjukkan bahwa k=3 adalah jumlah kluster yang optimal untuk segmentasi pelanggan dalam penelitian ini.

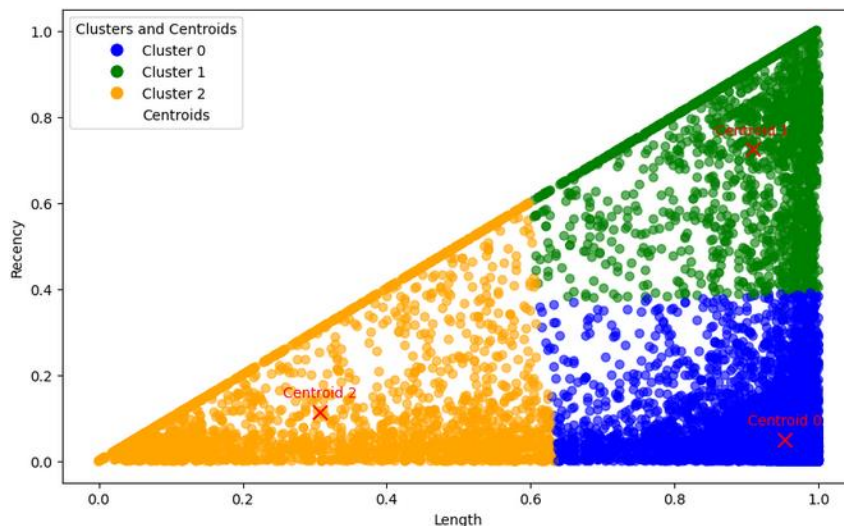
3.2.2. Segmentasi dengan Nilai k Optimal

Setelah menentukan nilai k optimal menggunakan DBI dan menemukan k=3 sebagai jumlah kluster terbaik, segmentasi dilakukan dengan algoritma bisecting k-means. Algoritma ini dimulai dengan satu kluster besar, lalu membaginya secara berulang menjadi dua kluster lebih kecil hingga mencapai k=3. Data transaksi pelanggan dikelompokkan berdasarkan atribut LRFM, menghasilkan distribusi jumlah transaksi per kluster dan centroid untuk masing-masing atribut seperti ditunjukkan pada Tabel 3.

Tabel 3. Distribusi Jumlah Transaksi Pelanggan per Klaster

Klaster	Jumlah	Centroids			
		Length	Recency	Frequency	Monetary
0	12.028	0.952748	0.050477	0.003629	0.005508
1	2.170	0.908501	0.725363	0.000484	0.000596
2	3.011	0.306798	0.114541	0.000548	0.000596

Gambar 3 menampilkan *scatter plot* antara Recency dan Length yang menunjukkan bagaimana pelanggan dengan nilai recency dan length yang berbeda dikelompokkan ke dalam klaster yang berbeda.



Gambar 3. Scatter Plot Recency dan Length

3.2.3. Analisis Klaster

Tahap ini menganalisis hasil klasterisasi dari segmentasi bisecting k-means dengan k=3. Tabel 4 menunjukkan rata-rata atribut LRFM dan total pelanggan per klaster. Analisis ini bertujuan memahami karakteristik tiap klaster dan memberikan rekomendasi strategi yang sesuai untuk setiap segmen.

Tabel 4. Nilai Rata-rata Atribut LRFM per Klaster

Klaster	Jumlah	Rata-rata			
		Length	Recency	Frequency	Monetary
0	12.028	346	19	40	100.442.233,41
1	2.170	330	263	6	10.854.487,52
2	3.011	112	42	7	9.680.087,58
Keseluruhan	17.209	303	54	30	73.265.161,55

Berikut ini adalah hasil analisis mendetail untuk setiap klaster yang terbentuk berdasarkan Tabel 4.

Klaster 0

Klaster 0 terdiri dari pelanggan loyal dengan durasi keanggotaan rata-rata 346 hari, frekuensi transaksi 40 kali per tahun, dan nilai transaksi tinggi, rata-rata di atas 100 juta rupiah. Dengan recency rendah dengan rata-rata 19 hari, pelanggan ini baru saja bertransaksi. Klaster ini merupakan pelanggan sangat berharga, sehingga perusahaan sebaiknya mempertahankan mereka melalui program loyalitas dan penawaran khusus.

Klaster 1

Klaster 1 terdiri dari pelanggan dengan durasi keanggotaan panjang yaitu rata-rata 330 hari, tetapi frekuensi transaksi rendah dengan rata-rata 6 kali per tahun dan nilai transaksi rata-rata hanya 10,8 juta rupiah. Dengan recency tinggi mencapai rata-rata 263 hari, pelanggan ini sudah lama tidak bertransaksi. Klaster ini berpotensi besar, sehingga perusahaan dapat mengaktifkan mereka kembali melalui strategi retensi seperti penawaran eksklusif atau program pengingat.

Klaster 2

Klaster 2 terdiri dari pelanggan dengan durasi keanggotaan pendek yaitu rata-rata 112 hari, frekuensi transaksi rendah hanya sekitar 7 kali per tahun, dan nilai transaksi rata-rata 9,6 juta rupiah. Dengan recency 42

hari, pelanggan ini masih bertransaksi tetapi kurang aktif dibanding Klaster 0. Klaster ini mungkin terdiri dari pelanggan baru, sehingga perusahaan dapat meningkatkan keterlibatan mereka melalui strategi akuisisi dan pengenalan produk.

3.3. Prediksi *Churn* dengan LSTM

Setelah segmentasi pelanggan menggunakan bisecting k-means, langkah selanjutnya adalah memprediksi *churn* menggunakan LSTM. Long Short-Term Memory (LSTM), yang diperkenalkan oleh Hochreiter dan Schmidhuber pada 1997, adalah jaringan saraf yang dirancang untuk mengatasi masalah vanishing dan exploding gradient pada Recurrent Neural Network (RNN) [17][18]. LSTM terdiri dari tiga lapisan utama, yaitu input, output, dan lapisan tersembunyi, yang dilengkapi dengan sel memori dan tiga gerbang untuk mengontrol aliran informasi [19].

Pemodelan ini memanfaatkan atribut Length, Recency, Frequency dan Monetary yang dipecah menjadi *monthly revenue*, serta hasil klaster dari segmentasi sebelumnya. Model *churn* dirancang untuk mempertimbangkan pola transaksi bulanan dan segmentasi pelanggan berdasarkan karakteristik LRFM. Tahapan prediksi *churn* mencakup pemberian label, optimasi hiperparameter menggunakan random search, dan pemodelan *churn* dengan LSTM.

3.3.1. Pemberian Label

Sebagai langkah awal dalam prediksi *churn*, pemberian label pada data pelanggan adalah hal yang esensial. Label ini akan menunjukkan apakah seorang pelanggan mengalami *churn* atau tidak, berdasarkan definisi yang telah ditetapkan. Berdasarkan penelitian yang dilakukan oleh Gattermann, pelanggan didefinisikan sebagai *churn* jika memenuhi dua kondisi utama. Pertama, revenue mengalami penurunan sebesar 30% atau lebih secara berturut-turut. Kedua, pelanggan tersebut tidak memiliki transaksi selama tiga bulan terakhir secara berturut-turut [20].

Proses pemberian label ini melibatkan beberapa langkah penting. Pertama, data transaksi setiap pelanggan dianalisis untuk mendeteksi adanya periode tanpa transaksi dan perubahan revenue. Jika ditemukan penurunan sebesar 30% atau lebih secara berturut-turut dan pelanggan tidak memiliki transaksi selama tiga bulan terakhir secara berturut-turut, maka pelanggan tersebut akan diberi label "*churn*" dengan nilai biner 1. Sebaliknya, jika salah satu atau kedua kondisi tidak terpenuhi, pelanggan diberi label "*non-churn*" dengan nilai biner 0.

3.3.2. Hiperparameter Optimization dengan Random Search

Dalam pembelajaran mesin, hiperparameter adalah parameter yang ditentukan sebelum pembelajaran dimulai. Hyperparameter optimization bertujuan menemukan kombinasi terbaik untuk meminimalkan fungsi kerugian pada data independen, sehingga menghasilkan model yang optimal [21]. Random Search adalah teknik optimasi hiperparameter yang memilih kombinasi secara acak dari ruang pencarian, memungkinkan efisiensi lebih tinggi dibanding Grid Search [22].

Pada tahap ini, dilakukan optimasi hiperparameter model LSTM untuk meningkatkan akurasi prediksi *churn* menggunakan metode Random Search. Proses ini bertujuan mencari kombinasi terbaik dari parameter model, seperti jumlah unit dalam layer LSTM, learning rate, batch size, dan dropout rate, yang dapat mempengaruhi kinerja model secara signifikan, seperti yang ditunjukkan pada Tabel 5.

Tabel 5. Hiperparameter dan Ruang Pencarian yang digunakan

Hiperparameter	Ruang Pencarian	Keterangan
<i>Units</i>	64, 128, 192, 256, 320, 384, 448	Jumlah unit dalam lapisan LSTM. Variasi dari 64 hingga 448 dengan langkah 64.
<i>Dropout Rate</i>	0.1, 0.2, 0.3, 0.4, 0.5	Tingkat dropout untuk lapisan Dropout. Variasi dari 0.1 hingga 0.5.
<i>Learning Rate</i>	0.0001, 0.001, 0.01, 0.1	Laju pembelajaran untuk optimizer Adam. Dipilih dari rentang 0.0001 hingga 0.1 dengan skala logaritmik.
<i>Batch Size</i>	32	Jumlah sampel per batch pelatihan.
<i>Epochs</i>	50	Jumlah epoch untuk pelatihan model.
<i>Validation Split</i>	20%	Persentase data pelatihan yang digunakan sebagai data validasi.
<i>Max Trials</i>	20	Jumlah maksimal kombinasi hiperparameter yang diuji selama proses optimasi.
<i>Executions per Trial</i>	2	Jumlah pengulangan untuk setiap kombinasi hiperparameter.

Berdasarkan Tabel 5, model LSTM ini terdiri dari lapisan LSTM dengan jumlah unit antara 64 hingga 448, yang dioptimalkan untuk mengurangi *overfitting* dengan lapisan Dropout (tingkat 0.1 hingga 0.5). Lapisan terakhir adalah Dense dengan satu neuron dan fungsi aktivasi sigmoid, menghasilkan output biner (*churn* atau *non-churn*). Model dikompilasi menggunakan `binary_crossentropy` dan optimizer Adam dengan learning rate yang dioptimalkan antara 0.0001 hingga 0.1. Setiap kombinasi hiperparameter diuji dua kali untuk

mendapatkan kinerja rata-rata yang lebih andal. Pelatihan dilakukan selama 50 epoch dengan 20% data sebagai validasi, menggunakan batch size 32. Proses optimasi hiperparameter dengan Random Search dilakukan dengan 20 percobaan untuk menemukan kombinasi terbaik, yang hasilnya ditampilkan pada Tabel 6.

Tabel 6. Hasil Val. Accuracy dan Val. Loss untuk Setiap Percobaan

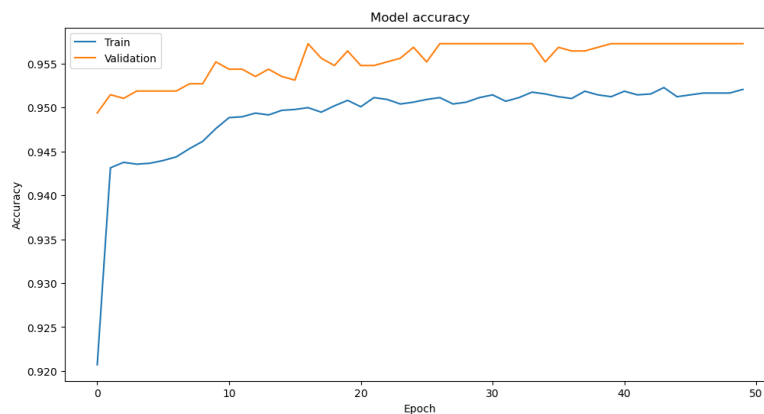
Trial	Hyperparameter			Val. Accuracy	Val. Loss
	Units	Dropout Rate	Learning Rate		
1	448	0.4	0.000693291657	0.9558091164	0.1735642329
2	320	0.2	0.007243737160	0.9572613835	0.1394070312
3	256	0.3	0.037541299883	0.9572613835	0.1468811557
4	192	0.4	0.000427669288	0.9551867247	0.1750657111
5	256	0.1	0.000141515766	0.9556016326	0.1759284213
6	384	0.4	0.000317478916	0.9556016326	0.1739409417
7	192	0.3	0.000188658087	0.9556016624	0.1759450957
8	384	0.1	0.044473133208	0.9572613835	0.1459484249
9	192	0.4	0.001494790019	0.9572613835	0.1584269553
10	64	0.2	0.000999982408	0.9560165703	0.1751130819
11	192	0.5	0.067817654963	0.9570539296	0.1485604420
12	256	0.4	0.030077113359	0.9572613835	0.1461057514
13	384	0.5	0.001466765052	0.9572613835	0.1561013535
14	64	0.5	0.005656262608	0.9572613835	0.1508153230
15	320	0.4	0.021484372501	0.9572613835	0.1500603929
16	192	0.3	0.000654487989	0.9556016326	0.1741091236
17	384	0.4	0.000531326805	0.9560166001	0.1743794680
18	384	0.1	0.004332066367	0.9576763511	0.1318387687
19	448	0.4	0.000837367584	0.9553941786	0.1779503077
20	256	0.4	0.029940308361	0.9572613835	0.1504767984

Hasil optimasi hiperparameter untuk model LSTM ditampilkan dalam Tabel 6, yang menunjukkan Val. Accuracy dan Val. Loss untuk setiap percobaan. Proses ini berlangsung selama 1 jam, 51 menit, dan 7 detik dengan 20 kombinasi hiperparameter yang diuji. Percobaan terbaik menggunakan 384 unit LSTM, dropout rate 0.1, dan learning rate 0.004332066367, menghasilkan Val. Accuracy sebesar 0.9577 dan Val. Loss 0.1318, menunjukkan model dengan akurasi tertinggi dan loss terendah dalam prediksi *churn*.

3.3.3. Pemodelan *Churn* dengan LSTM

Setelah menemukan kombinasi hiperparameter optimal, yaitu 384 unit LSTM, dropout rate 0.1, dan learning rate 0.004332066367, model LSTM dibangun dan dilatih untuk memprediksi *churn*. Arsitektur model ini didefinisikan menggunakan Keras Sequential API, yang terdiri dari satu lapisan LSTM dengan 384 unit dan fungsi aktivasi ReLU. Lapisan Dropout dengan tingkat dropout 0.1 ditambahkan untuk mengurangi risiko *overfitting*, sementara lapisan terakhir adalah Dense layer dengan satu neuron dan fungsi aktivasi sigmoid, menghasilkan output biner (*churn* atau *non-churn*). Model ini dikompilasi menggunakan binary cross-entropy sebagai fungsi loss dan optimizer Adam dengan learning rate yang telah dioptimalkan.

Proses pelatihan dilakukan selama 50 epoch, dengan 20% dari data pelatihan digunakan sebagai data validasi untuk memantau kinerja model dan menghindari *overfitting*. Parameter lainnya termasuk batch size 32 dan verbosity 1. Validation split ini memastikan bahwa 20% data pelatihan digunakan untuk validasi selama pelatihan. Grafik akurasi model selama pelatihan dan validasi terhadap jumlah epoch ditampilkan pada Gambar 4, yang menunjukkan kinerja model sepanjang proses pelatihan.



Gambar 4. Akurasi Model Selama Proses Pelatihan dan Validasi Terhadap Jumlah Epoch

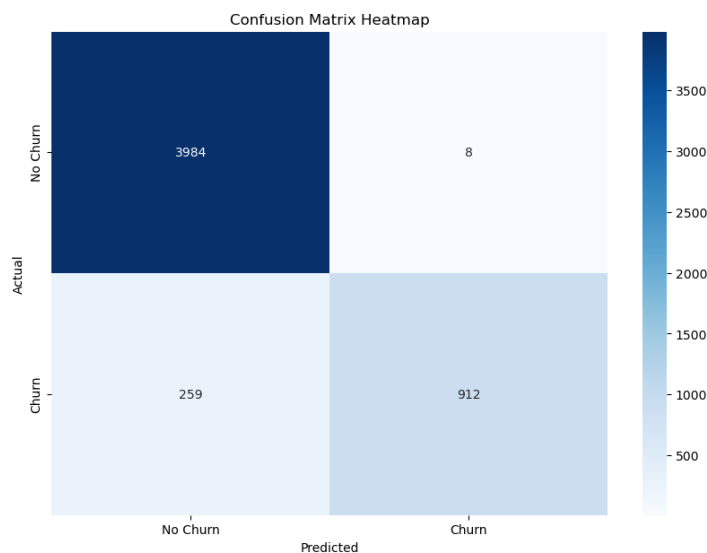
Berdasarkan Gambar 4, sumbu Y menunjukkan akurasi model yang berkisar antara 0.92 hingga 0.96, sedangkan sumbu X menunjukkan jumlah epoch. Garis biru mewakili akurasi pada data pelatihan, dan garis oranye pada data validasi. Pada awal pelatihan (epoch 0-10), akurasi pelatihan meningkat tajam, lalu melambat dan stabil setelah epoch 10, menandakan model telah mempelajari pola utama dalam data. Akurasi validasi yang lebih tinggi menunjukkan model mampu melakukan generalisasi dengan baik, dan stabilitas kedua garis setelah beberapa epoch menunjukkan model telah mencapai konvergensi.

3.3.4. Evaluasi Model

Tahap evaluasi adalah langkah penting untuk menilai kinerja model *churn* pelanggan. Evaluasi dilakukan menggunakan confusion matrix dan AUC-ROC untuk menggambarkan kemampuan model dalam memprediksi *churn* dengan akurat. Dengan data uji, evaluasi ini memastikan bahwa model tidak hanya berkinerja baik pada data pelatihan, tetapi juga dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat.

Evaluasi dengan Confusion Matrix

Pengujian dilakukan terhadap data testing dengan jumlah 5.163 baris yang menghasilkan akurasi sebesar 0.9482858803021499. Gambar 5 menampilkan *heatmap* dari confusion matrix yang merupakan hasil dari pengujian tersebut.



Gambar 5. Confusion Matrix *Heatmap*

Confusion matrix pada gambar 5 memperlihatkan bahwa model berhasil mengidentifikasi 3.984 pelanggan yang tidak *churn* dengan benar (*True Negatives*) dan hanya salah mengidentifikasi 8 pelanggan yang tidak *churn* sebagai *churn* (*False Positives*). Di sisi lain, model juga berhasil mengidentifikasi 912 pelanggan yang *churn* dengan benar (*True Positives*) dan salah mengidentifikasi 259 pelanggan *churn* sebagai tidak *churn* (*False Negatives*). Tabel 7 menampilkan *classification report* dari hasil pengujian yang dilakukan.

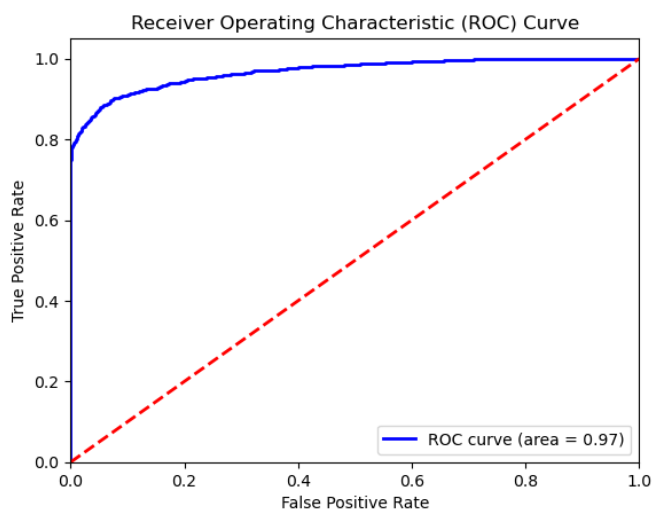
Tabel 7. *Classification Report*

	Precision	Recall	F1-Score	Support
<i>No Churn</i>	0.94	1.00	0.97	3992
<i>Churn</i>	0.99	0.78	0.87	1171
Accuracy			0.95	5163
Macro avg.	0.97	0.89	0.92	5163
Weighted avg.	0.95	0.95	0.95	5163

Classification report yang ditampilkan pada Tabel 7 menunjukkan nilai precision untuk kelas *No Churn* dan *Churn* masing-masing adalah 0.94 dan 0.99, menunjukkan tingkat akurasi prediksi *churn* yang tinggi. Recall untuk kelas *No Churn* adalah 1.00, yang berarti model berhasil menangkap semua pelanggan yang tidak *churn*. Namun, recall untuk kelas *Churn* adalah 0.78, menunjukkan model gagal menangkap 22% pelanggan yang *churn*. F1-Score, yang merupakan harmonic mean dari precision dan recall, menunjukkan nilai tinggi untuk kedua kelas, menunjukkan keseimbangan yang baik antara precision dan recall.

Evaluasi dengan AUC-ROC

Selain confusion matrix, AUC-ROC juga digunakan untuk evaluasi kinerja model. Gambar 6 menampilkan kurva Receiver Operating Characteristic (ROC) dari hasil pengujian model.



Gambar 6. Kurva Receiver Operating Characteristic (ROC)

Gambar 6 menunjukkan kinerja model dalam memprediksi *churn* dengan menampilkan *trade-off* antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). AUC-ROC yang dihasilkan adalah 0.9682422060175004, menunjukkan bahwa model memiliki kinerja yang sangat baik dalam membedakan antara pelanggan yang *churn* dan yang tidak *churn*.

Dari proses pemodelan yang telah dilakukan pada data training, diperoleh hasil prediksi *churn* pada data testing yang dapat dilihat pada Tabel 8.

Tabel 8. Prediksi *Churn* per Klaster

Klaster	Total	L	R	F	M	<i>Churn</i>	%
0	3.615	346	19	43	101,748,693.20	116	3.21
1	659	329	262	5	6,026,256.81	659	100
2	889	115	42	8	12,058,809.80	145	16.31
Keseluruhan	5.163	304	54	32	74,087,373.83	920	

Dari proses pemodelan *churn*, diperoleh hasil prediksi *churn* pada data testing yang dirangkum dalam Tabel 8. Tabel tersebut menunjukkan distribusi *churn* di setiap klaster. Klaster 0 memiliki 3.615 pelanggan dengan 116 (3.21%) berpotensi *churn*. Klaster 1 memiliki 659 pelanggan dengan semua pelanggan (100%) berpotensi *churn*. Klaster 2 memiliki 889 pelanggan dengan 145 (16.31%) berpotensi *churn*. Dengan demikian, hasil prediksi *churn* menunjukkan bahwa klaster 1 memiliki tingkat *churn* tertinggi (100%), klaster 0 memiliki persentase *churn* terendah (3.21%), dan klaster 2 memiliki persentase *churn* sebesar 16.31%. Total pelanggan yang *churn* adalah 920 dari 5.163 pelanggan.

Distribusi *churn* per klaster memperlihatkan variasi perilaku pelanggan yang signifikan. Klaster 1 yang seluruhnya berpotensi *churn* menggambarkan kelompok pelanggan dengan tingkat aktivitas rendah atau penurunan nilai transaksi yang drastis, sementara klaster 0 menunjukkan kelompok pelanggan yang paling stabil dan loyal. Temuan ini menunjukkan bahwa pendekatan segmentasi menggunakan Bisecting K-Means berbasis atribut LRFM berhasil mengelompokkan pelanggan berdasarkan perilaku dan nilai ekonominya secara bermakna. Dengan demikian, strategi retensi dapat difokuskan pada klaster dengan risiko *churn* tertinggi melalui penawaran promosi, peningkatan pengalaman pelanggan, atau program loyalitas yang disesuaikan dengan karakteristik tiap segmen.

4. PENUTUP

Berdasarkan hasil penelitian, dapat disimpulkan bahwa integrasi metode Bisecting K-Means dan LSTM menunjukkan kinerja yang sangat baik dalam memprediksi customer *churn* pada pelanggan B2B di sektor distribusi FMCG. Segmentasi menggunakan Bisecting K-Means berhasil mengelompokkan pelanggan ke dalam tiga klaster optimal dengan nilai Davies-Bouldin Index (DBI) sebesar 0.46, yang menunjukkan bahwa setiap klaster memiliki karakteristik perilaku yang berbeda secara bermakna. Pendekatan ini mampu

memisahkan pelanggan berdasarkan dimensi Length, Recency, Frequency, dan Monetary (LRFM), sehingga menghasilkan kelompok pelanggan yang representatif terhadap tingkat loyalitas dan risiko *churn*.

Model prediksi *churn* yang dikembangkan dengan arsitektur LSTM berisi 384 unit, dropout rate 0.1, dan learning rate 0.0043 menunjukkan kinerja yang sangat baik dengan akurasi validasi 96%, akurasi pengujian 95%, dan AUC-ROC 0.97. Nilai precision dan recall yang tinggi pada masing-masing kategori menunjukkan bahwa model tidak hanya mampu mengenali pelanggan yang berpotensi *churn* dengan tepat, tetapi juga menjaga keseimbangan dalam mengidentifikasi pelanggan yang tetap loyal. Temuan ini menegaskan bahwa pemanfaatan LSTM efektif dalam menangkap pola temporal pada data pendapatan bulanan, yang menjadi indikator penting perilaku pelanggan B2B.

Secara substantif, hasil penelitian ini memberikan implikasi praktis bagi perusahaan dalam mengelola hubungan pelanggan dan menyusun strategi retensi yang lebih tepat sasaran. Klaster 1, dengan tingkat *churn* tertinggi (100%), menunjukkan kelompok pelanggan yang memerlukan perhatian khusus melalui strategi retensi seperti penawaran ulang (*re-engagement*) atau insentif pembelian berulang. Klaster 2 dengan tingkat *churn* menengah (16.31%) memerlukan pendekatan pencegahan *churn* berbasis penguatan loyalitas, sedangkan klaster 0 dengan tingkat *churn* terendah (3.21%) dapat dijaga melalui program penghargaan loyalitas dan peningkatan hubungan bisnis jangka panjang.

Sebagai saran untuk penelitian selanjutnya, eksperimen dengan menggunakan metode normalisasi yang berbeda dan teknik optimasi hiperparameter lainnya dapat dilakukan untuk meningkatkan kinerja model. Selain itu, pengujian model dengan dataset yang lebih besar dan lebih beragam diharapkan dapat meningkatkan kemampuan generalisasi dan akurasi prediksi model.

DAFTAR PUSTAKA

- [1] T. Hargyatni, K. Djati Purnama, D. Wiratnoko, R. A. Kusumajaya, S. Handoko, and S. Stekom Surakarta, "The Framework of Customer Engagement on Customer Satisfaction : The Antecedents and Consequences," 2022. [Online]. Available: <http://iconesia.co.id/>
- [2] J. Saura, A. Reyes-Menendez, N. Matos, M. Correia, and P. Palos-Sanchez, "Consumer Behavior in the Digital Age," *Journal of Spatial and Organizational Dynamics*, vol. 8, pp. 190–194, Sep. 2020.
- [3] M. Mirkovic, T. Vučković, D. Stefanović, A. Anderla, and D. Gracanin, "Customer Churn Prediction in B2B Non-Contractual Business Settings Using Invoice Data," *Applied Sciences*, vol. 12, p. 5001, May 2022, doi: 10.3390/app12105001.
- [4] T. Nanang, *Panduan Lengkap Manajemen Distribusi*. Yogyakarta: Quadrant, 2019.
- [5] C. Grönroos, "Relationship Marketing: The Strategy Continuum," 1995.
- [6] M. Pejić Bach, J. Pivar, and B. Jaković, "Churn Management in Telecommunications: Hybrid Approach Using Cluster Analysis and Decision Trees," *Journal of Risk and Financial Management*, vol. 14, no. 11, Nov. 2021, doi: 10.3390/jrfm14110544.
- [7] X. Xiahou and Y. Harada, "B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 17, no. 2, pp. 458–475, Jun. 2022, doi: 10.3390/jtaer17020024.
- [8] H. Tran, N. Le, and V. H. Nguyen, "Customer Churn Prediction in The Banking Sector using Machine Learning-Based Classification Models," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 18, pp. 87–105, 2023, doi: 10.28945/5086.
- [9] P. Wachwanakijkul, S. Junsiritrakhoon, N. Kantanantha, G. Narayanamurthy, and P. Jarumaneeroj, "Data-driven approaches to predicting customer churn in a non-contractual car-sharing company," *Transp Res Interdiscip Perspect*, vol. 33, p. 101600, Sep. 2025, doi: 10.1016/J.TRIP.2025.101600.
- [10] W. Bhaya, "Review of Data Preprocessing Techniques in Data Mining," *Journal of Engineering and Applied Sciences*, vol. 12, pp. 4102–4107, Jan. 2017, doi: 10.3923/jeasci.2017.4102.4107.
- [11] A. Aldoseri, K. N. Al-Khalifa, and A. M. Hamouda, "Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges," Jun. 01, 2023, *MDPI*. doi: 10.3390/app13127082.
- [12] D. Singh and B. Singh, "Investigating the Impact of Data Normalization on Classification Performance," *Appl Soft Comput*, vol. 97, no. xxxx, p. 105524, 2020, doi: 10.1016/j.asoc.2019.105524.
- [13] S. M. Savaresi and D. L. Boley, "A comparative analysis on the bisecting K-means and the PDDP clustering algorithms," *Intelligent Data Analysis*, vol. 8, no. 4, pp. 345–362, 2004, doi: 10.3233/ida-2004-8403.
- [14] Z. Zhou *et al.*, "A Fast Screening Framework for Second-Life Batteries Based on an Improved Bisecting K-Means Algorithm Combined with Fast Pulse Test," *J Energy Storage*, vol. 31, no. May, p. 101739, 2020, doi: 10.1016/j.est.2020.101739.

- [15] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognit*, vol. 30, no. 6, pp. 847–857, 1997, doi: [https://doi.org/10.1016/S0031-3203\(96\)00127-6](https://doi.org/10.1016/S0031-3203(96)00127-6).
- [16] G. Gan, C. Ma, and J. Wu, *Density-Based Clustering Algorithms*. in ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics, 2007. [Online]. Available: <https://books.google.co.id/books?id=rIQZAQAAIAAJ>
- [17] L. Skovajsová, "Long short-term memory description and its application in text processing," in *2017 Communication and Information Technologies (KIT)*, 2017, pp. 1–4. doi: 10.23919/KIT.2017.8109465.
- [18] A. Arfan and L. Etp, "Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia," *PETIR*, vol. 13, pp. 33–43, Mar. 2020, doi: 10.33322/petir.v13i1.858.
- [19] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Long short-term memory based operation log anomaly detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 236–242. doi: 10.1109/ICACCI.2017.8125846.
- [20] T. Gattermann-Itschert, U. W. Thonemann, and T. Gattermann, "Proactive customer retention management in a non-contractual B2B setting based on churn prediction with random forests," 2022. [Online]. Available: <https://www.researchgate.net/publication/353794359>
- [21] R. Ghawi and J. Pfeffer, "Efficient Hyperparameter Tuning with Grid Search for Text Categorization Using kNN Approach with BM25 Similarity," vol. 9, no. 1, pp. 160–180, 2019, doi: doi:10.1515/comp-2019-0011.
- [22] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012, [Online]. Available: <http://jmlr.org/papers/v13/bergstra12a.html>