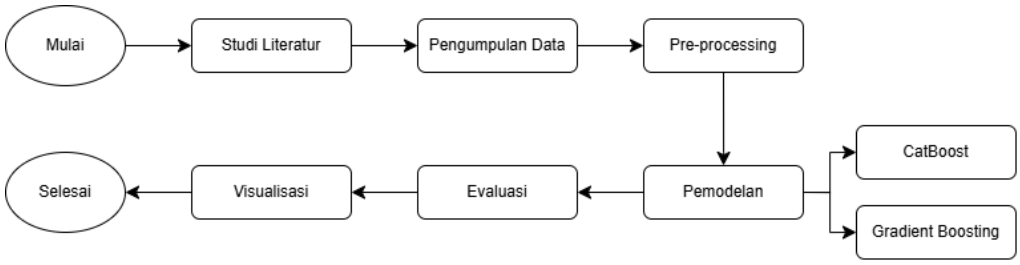
BAB III

DESAIN DAN IMPLEMENTASI SISTEM

3.1 Metode Penelitian

Dalam penelitian ini, peneliti mentukan model yang paling optimal dengan menggunakan *Gradient Boosting* dan *CatBoost* dalam memprediksi *churn* di Apartemen Trillium. Secara garis besar, tahapan penelitian terdiri dari 6 (enam) tahap, yaitu: (1) Studi literatur, (2) Pengumpulan data, (3) *Pre-processing* data, (4) Pemodelan, (5) Evaluasi, dan (6) Visualisasi. Proses alur tahapan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 menggambarkan alur lengkap penelitian yang selanjutnya akan dijelaskan setiap bagiannya secara rinci.

3.2 Studi Literatur

Penelitian ini mengandalkan studi literatur dari berbagai sumber ilmiah yang terpercaya, seperti tesis, disertasi, artikel penelitian, serta jurnal nasional dan internasional. Sumber-sumber tersebut digunakan untuk membangun dasar teori yang kokoh dengan memberikan wawasan mendalam mengenai konteks permasalahan, teori terkait, serta perkembangan terbaru di bidang studi. Selain itu, penelitian ini selektif dalam memilih literatur dengan memprioritaskan jurnal yang diterbitkan dalam lima tahun terakhir. Pendekatan ini memastikan bahwa informasi yang digunakan memiliki tingkat kredibilitas tinggi dan relevansi yang signifikan, baik di tingkat global maupun nasional, sehingga mendukung argumen, metodologi, dan validitas hasil penelitian.

3.3 Pengumpulan Data

Data riwayat penyewaan Apartemen Trillium tahun 2022-2024 digunakan untuk pengumpulan data, yang diperoleh dari sistem manajemen properti internal. Setiap bulan, data penyewaan dicatat, mencakup informasi seperti durasi sewa, jumlah unit yang disewa, harga sewa, serta waktu pembayaran. Selain data utama ini, juga dikumpulkan data

pendukung seperti tingkat hunian, tipe unit yang disewa, dan tingkat kepuasan pelanggan berdasarkan survei. Tujuan pengumpulan data ini adalah untuk memperoleh informasi yang representatif dan relevan mengenai perilaku pelanggan terkait potensi churn. Kehadiran data yang lengkap dan valid diharapkan dapat meningkatkan akurasi serta keandalan metode Gradient Boosting dan CatBoost dalam memprediksi pelanggan yang berpotensi churn di Apartemen Trillium. Selain itu, untuk memahami hubungan antara variabel input dan output, analisis statistik dan visualisasi data akan dilakukan menggunakan dataset ini.

3.4 *Pre-processing* Data

Tahap *pre-processing* adalah langkah penting untuk memastikan bahwa data yang digunakan dalam model prediksi *churn* pelanggan berada dalam format yang sesuai dan siap digunakan. Berikut adalah tahapan *pre-processing* yang dilakukan pada data riwayat penyewaan Apartemen Trillium:

1. Pengumpulan Data dan Validasi
    - Data riwayat penyewaan dikumpulkan dari sistem manajemen properti.
    - Dilakukan validasi untuk memastikan bahwa data yang diperoleh lengkap, tidak ada nilai yang hilang (*missing values*), dan konsisten dalam format yang digunakan (contoh: tanggal dalam format standar, angka dalam satuan yang seragam).
2. Pembersihan Data (*Data Cleaning*)
    - Mengatasi *Missing Values*: Nilai-nilai yang hilang (jika ada) ditangani menggunakan metode seperti imputasi median/mean atau penghapusan jika persentase data hilang terlalu kecil.
    - Menghapus *Outlier*: *Outlier* diidentifikasi menggunakan metode statistik (misalnya, *IQR* atau Z-*score*) dan ditangani dengan transformasi, imputasi, atau penghapusan jika diperlukan.
    - Standarisasi Format: Format data yang tidak seragam, seperti nama pelanggan atau ID unit, diselaraskan untuk menghindari inkonsistensi.
3. Transformasi Data
    - Pengkodean Variabel Kategorikal: Data seperti tipe unit atau tingkat kepuasan pelanggan diubah menjadi nilai numerik menggunakan metode seperti *One-Hot Encoding* atau *Label Encoding*.

- Skalasi Fitur: Data numerik seperti harga sewa atau lama penyewaan diskalakan menggunakan *Min-Max Scaling* atau *Standar Scaling* agar lebih sesuai dengan algoritma prediksi.
- Pembuatan Fitur Baru (*Feature Engineering*): Fitur tambahan, seperti rasio keterlambatan pembayaran atau tingkat penurunan kepuasan dari waktu ke waktu, dibuat untuk memperkaya informasi data.

4. Pengelompokan Waktu (*Time Aggregation*)
    - Data bulanan atau mingguan diubah menjadi format agregat jika diperlukan, untuk mempermudah analisis tren dan hubungan variabel.
    - Variabel waktu, seperti durasi sewa atau tren tahunan, ditambahkan sebagai input tambahan dalam model.
5. Pemeriksaan Korelasi dan Reduksi Dimensi
    - Analisis korelasi dilakukan untuk mengidentifikasi variabel yang memiliki hubungan kuat dengan churn pelanggan, menghindari multikolinearitas, dan memilih fitur yang paling relevan.
    - Reduksi Dimensi: Jika jumlah fitur terlalu banyak, digunakan metode seperti *PCA* (*Principal Component Analysis*) untuk mengurangi dimensi tanpa kehilangan informasi penting.
6. Pembagian Dataset
    - Data dibagi menjadi *training set* dan *test set* dengan perbandingan umum 70:30 atau 80:20, guna memastikan evaluasi model yang akurat.
    - Teknik *stratified* sampling digunakan untuk memastikan distribusi *churn/non-churn* serupa di semua subset data.

Tahapan pre-processing ini bertujuan untuk menghasilkan dataset yang bersih, terstruktur, dan optimal untuk digunakan oleh algoritma *Gradient Boosting* dan *CatBoost*, sehingga model dapat memprediksi pelanggan yang berpotensi *churn* dengan lebih akurat.

3.5 Pemodelan

Pengimplementasian metode Gradient Boosting dan CatBoost dalam memprediksi churn pada Apartemen Trillium.

3.5.1 Gradient Boosting

1. Inisialisasi Model

    Gradient Boosting dimulai dengan memprediksi rata-rata nilai target untuk semua data:

$$F_0(x) = argmin_c \sum_{i=1}^{N} L(y_i, c)$$

- $F_0(x)$: Model awal (prediksi rata-rata awal).
- $L(y_i, c)$: Fungsi loss yang mengukur kesalahan antara nilai sebenarnya (y_i) dan prediksi (c).

Untuk klasifikasi binari:

- Biasanya menggunakan **Log Loss**:

$$L(y, p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i))$$

di mana:

- $p_i = \frac{1}{1 + e^{-F(x_i)}}$: Probabilitas prediksi churn.

2. Membentuk Model Iteratif

Gradient Boosting bekerja secara iteratif untuk meminimalkan fungsi loss.

a. Menghitung Residual (Gradient Negative): Residual adalah gradien negatif dari fungsi loss terhadap prediksi sebelumnya $\left(F_m(x)\right)$:

$$r_i^{(m)} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F(x_i) = F_{m-1}(x_i)}$$

Untuk Log Loss:

$$r_i^{(m)} = y_i - \frac{1}{1 + e^{-F_{m-1}(x_i)}}$$

- $r_i^{(m)}$: Residual pada iterasi ke-mm.
- $F_{m-1}(x_i)$: Prediksi pada iterasi sebelumnya.

b. Melatih Pohon Keputusan Baru: Pohon baru dilatih untuk memprediksi residual $\left(r_i^{(m)}\right)$.

3. Memperbarui Prediksi

Prediksi diperbarui dengan menambahkan kontribusi dari pohon baru $\left(h_m(x)\right)$:

$$F_m(x) = F_{m-1}(x) + \eta\, .\, h_m(x)$$

- η: Learning rate (nilai kecil, biasanya 0.01–0.1, untuk mengontrol langkah update).
- $h_m(x)$: Output dari pohon ke-m.

4. Fungsi Loss untuk Klasifikasi

   Gradient Boosting menggunakan fungsi loss untuk mengevaluasi performa model pada setiap iterasi. Fungsi umum untuk klasifikasi binari:

   1. Log Loss (Binary Cross-Entropy):

$$L(y,p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i\, log(p_i) + (1-y_i)\, log(1-p_i))$$

   2. Accuracy**:**

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Data}$$

   3. Precision**:**

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

   4. Recall**:**

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

   5. F1-Score**:**

$$F1-Score = 2\left(\frac{Precision\ x\ Recall}{Precision + Recall}\right)$$

   6. ROC-AUC Score: Area di bawah kurva ROC, digunakan untuk mengevaluasi kemampuan model dalam membedakan antara churn (1) dan non-churn (0).

5. Probabilitas Prediksi

   Setelah semua iterasi selesai, Gradient Boosting menghasilkan skor prediksi ($F_M(x)$), yang kemudian diubah menjadi probabilitas:

$$p(x) = \frac{1}{1 + e^{-F_M(x)}}$$

   - Jika $p(x) \geq 0.5$, prediksi churn (1).
   - Jika $p(x) < 0.5$, prediksi non-churn (0).

3.5.2 CatBoost

CatBoost adalah salah satu implementasi Gradient Boosting yang memiliki sejumlah modifikasi khusus, seperti menangani data kategoris tanpa memerlukan one-hot encoding.

1. Inisialisasi Model

   CatBoost dimulai dengan model awal $F_0(x)$, yaitu prediksi rata-rata target untuk meminimalkan fungsi loss:

$$F_0(x) = argmin_c \sum_{i=1}^{N} L(y_i, c)$$

   - $F_0(x)$: Model awal (prediksi awal).
   - $L(y_i, c)$: Fungsi loss yang mengukur kesalahan antara nilai sebenarnya (y_i) dan prediksi (c).

Untuk klasifikasi biner, CatBoost biasanya menggunakan Log Loss, sama seperti Gradient Boosting:

$$L(y, p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i))$$

- $p_i = \frac{1}{1 + e^{-F(x_i)}}$: Probabilitas prediksi churn.

2. Modifikasi pada Residual

   Alih-alih menghitung residual menggunakan gradien loss murni, CatBoost memanfaatkan pendekatan Ordered Boosting untuk mengurangi overfitting. Hal ini dilakukan dengan:

   a. Membagi data menjadi subset berdasarkan urutan waktu.

   b. Menghindari kebocoran data (data leakage) dengan hanya menggunakan informasi yang tersedia sebelum setiap iterasi.

   Residual dihitung sebagai gradien loss terhadap prediksi sebelumnya:

$$r_i^{(m)} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x_i) = F_{m-1}(x_i)}$$

   Untuk Log Loss:

$$r_i^{(m)} = y_i - \frac{1}{1 + e^{-F_{m-1}(x_i)}}$$

3. Mengelola Data Kategoris

   CatBoost secara unik menangani data kategoris dengan Target Encoding pada setiap iterasi:

$$TE(category) = \frac{Sum\ of\ target\ for\ category}{Count\ of\ category}$$

   - Encoding ini dilakukan dengan pemisahan berdasarkan subset data untuk menghindari data leakage.

4. Memperbarui Prediksi

   Prediksi diperbarui dengan menambahkan kontribusi dari pohon baru $(h_m(x))$:

$$F_m(x) = F_{m-1}(x) + \eta \, . h_m(x)$$

   - η: Learning rate (nilai kecil, biasanya 0.01–0.1, untuk mengontrol langkah update).
   - $h_m(x)$: Output dari pohon ke-m.

5. Probabilitas Prediksi

   Setelah semua iterasi selesai, CatBoost menghasilkan skor prediksi $(F_M(x))$, yang kemudian diubah menjadi probabilitas:

$$p(x) = \frac{1}{1 + e^{-F_M(x)}}$$

   - Jika $p(x) \geq 0.5$, prediksi churn (1).
   - Jika $p(x) < 0.5$, prediksi non-churn (0).

CatBoost mengoptimalkan Gradient Boosting dengan fokus pada performa dan efisiensi untuk menangani data yang tidak seimbang dan memiliki fitur kategoris secara alami.

3.6 Evaluasi

CatBoost dan Gradien Boosting mendukung berbagai metrik evaluasi, termasuk:

a. Confusion Matrix:
   - True Positive (TP): Prediksi churn yang benar.
   - False Positive (FP): Prediksi churn yang salah.
   - True Negative (TN): Prediksi non-churn yang benar.
   - False Negative (FN): Prediksi non-churn yang salah.

b. ROC-AUC Score: Area di bawah kurva ROC untuk mengukur kemampuan model dalam membedakan antara churn dan non-churn.

c. Lift: Mengukur seberapa efektif model dibandingkan tebakan acak:

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)}$$

3.7 Visualisasi

Evaluasi model menggunakan Confusion Matrix dan AUC-ROC Curve adalah cara yang efektif untuk memahami kinerja model prediksi. Berikut adalah cara menyajikannya dengan lebih terstruktur dan memanfaatkan berbagai tools:

1. Confusion Matrix
    - Tujuan: Memberikan gambaran tentang jumlah prediksi benar (True Positive, True Negative) dan salah (False Positive, False Negative) yang dilakukan oleh model.
    - Tools:
        - Python (Seaborn & Matplotlib): Digunakan untuk menghasilkan heatmap yang interaktif.
        - Excel atau Tableau: Jika ingin visualisasi berbasis spreadsheet, hasil confusion matrix dapat diekspor sebagai tabel dan divisualisasikan dengan conditional formatting.
    - Interpretasi
        - True Positive (TP): Pelanggan churn yang berhasil diprediksi churn.
        - True Negative (TN): Pelanggan non-churn yang berhasil diprediksi non-churn.
        - False Positive (FP): Pelanggan non-churn yang salah diprediksi churn.
        - False Negative (FN): Pelanggan churn yang salah diprediksi non-churn.
2. AUC-ROC Curve
    - Tujuan: Mengevaluasi kemampuan model untuk membedakan antara churn (1) dan non-churn (0).
    - Tools:
        - Python (Scikit-learn): Digunakan untuk membuat kurva ROC dan menghitung AUC.

- Tableau: Tableau memungkinkan Anda untuk menampilkan ROC Curve secara interaktif dengan visualisasi yang mudah disesuaikan.
- Power BI: Dapat digunakan untuk membuat laporan visual dengan menyisipkan AUC-ROC yang dihasilkan melalui Python atau R.

- Interpretasi:
  - AUC (Area Under Curve):
    - 1.0: Model sempurna.
    - 0.5: Model tidak lebih baik dari tebakan acak.
    - > 0.8: Model memiliki kinerja yang baik.

3. Kombinasi Hasil Evaluasi
   - Menggunakan Dashboard: Hasil dari Confusion Matrix dan AUC-ROC Curve dapat digabungkan dalam dashboard untuk presentasi yang lebih komprehensif. Tools seperti Tableau, Power BI, atau Google Data Studio sangat membantu untuk tujuan ini.
   - Laporan Interaktif: Dengan Jupyter Notebook atau Google Colab, Anda dapat menyajikan visualisasi langsung bersama narasi hasil evaluasi.
4. Penekanan untuk Komparasi
   - Jika membandingkan Gradient Boosting dan CatBoost, buat grafik ROC Curve keduanya dalam satu plot.

Dengan menyajikan Confusion Matrix dan AUC-ROC Curve secara visual, pembaca akan lebih mudah memahami keunggulan dan kelemahan model. Pilihan tools seperti Python, Tableau, atau Power BI tergantung pada kebutuhan interaktivitas dan audiens.