J-MATH
Journal of Mathematics Theory and Applications
Vol. 4, No. 2, April 2026, pp. 01 – 12
ISSN Online: 3025-2628

Evaluasi Komparatif Algoritma Ensemble Learning pada Learning to Rank untuk Rekomendasi Produk berdasarkan data Transaksi E-Commerce

Aryanto[1*], Ebenhaiser Liunokas[2], Radian J. Situmeang[3]
[1,3*]Matematika/Statistika, Universitas Cenderawasih
[2]Matematika/Matematika, Universitas Timor
Email: aryanto.dandan@gmail.com, ebenhaizer.liunokas@unimor.ac.id

ABSTRACT
Learning to Rank (LTR) is a machine learning approach that has become central to modern product recommendation systems in the e-commerce industry. This study aims to compare the performance of three gradient boosting algorithms, namely XGBoost, LightGBM, and CatBoost, applied to the LTR task using the Instacart Online Grocery Shopping Dataset 2017. The dataset contains more than 3 million orders from over 200,000 anonymous users. Feature engineering was conducted by computing user-level and product-level aggregation statistics from historical transaction data. The data was partitioned by user to prevent information leakage between training and testing sets. Each algorithm was trained using its native listwise or pairwise ranking objective, and evaluated using the Normalized Discounted Cumulative Gain at position five (NDCG@5), Mean Average Precision at position five (MAP@5), and Mean Reciprocal Rank (MRR). Experimental results showed that CatBoost achieved the highest NDCG@5 (0.8036) and MAP@5 (0.8256), while XGBoost and LightGBM produced competitive and nearly identical results. All three models significantly outperformed the Random Forest baseline. These findings suggest that gradient boosting algorithms are highly effective for LTR in e-commerce recommendation scenarios and provide practical guidance for selecting ranking models in production systems.
Keyword: Learning to Rank; XGBoost; LightGBM; CatBoost; Product Recommendation; NDCG; E-Commerce

ABSTRAK
Learning to Rank (LTR) merupakan pendekatan machine learning yang telah menjadi inti dari sistem rekomendasi produk modern dalam industri *e-commerce*. Penelitian ini bertujuan membandingkan performa tiga algoritma gradient boosting, yaitu XGBoost, LightGBM, dan CatBoost, yang diterapkan pada tugas LTR menggunakan *Instacart Online Grocery Shopping* Dataset 2017. Dataset tersebut memuat lebih dari 3 juta pesanan dari lebih dari 200.000 pengguna anonim. Feature engineering dilakukan dengan menghitung statistik agregasi pada level pengguna dan level produk berdasarkan data transaksi historis. Pembagian data dilakukan berdasarkan pengguna untuk mencegah kebocoran informasi antara set pelatihan dan pengujian. Setiap algoritma dilatih menggunakan objektif ranking listwise atau pairwise bawaan masing-masing, dan dievaluasi menggunakan metrik NDCG@5, MAP@5, dan *Mean Reciprocal Rank* (MRR). Hasil eksperimen menunjukkan bahwa CatBoost mencapai NDCG@5 tertinggi (0,8036) dan MAP@5 tertinggi (0,8256), sementara XGBoost dan LightGBM menghasilkan performa yang kompetitif dan hampir identik. Ketiga model secara signifikan melampaui *baseline Random Forest*. Temuan ini menunjukkan bahwa algoritma gradient boosting sangat efektif untuk LTR dalam skenario rekomendasi *e-commerce* dan memberikan panduan praktis dalam pemilihan model *ranking* untuk sistem produksi.
Kata Kunci: Learning to Rank; XGBoost; LightGBM; CatBoost; Rekomendasi Produk; NDCG; E-Commerce

PENDAHULUAN

Dalam era transformasi digital yang pesat, platform *e-commerce* menghadapi tantangan fundamental dalam menyajikan produk yang relevan kepada jutaan pengguna secara personal dan efisien. Sistem rekomendasi merupakan komponen kritis yang secara langsung menentukan konversi penjualan, kepuasan pengguna, dan pendapatan platform. Di antara berbagai pendekatan yang telah dikembangkan, LTR telah muncul sebagai paradigma *machine learning* (ML) yang paling tepat untuk masalah pengurutan item dalam daftar rekomendasi, karena kemampuannya mengoptimalkan metrik evaluasi peringkat secara langsung (Liu, 2009).

LTR menggabungkan kelebihan metode supervised learning dengan kemampuan untuk mengoptimalkan fungsi objektif yang sensitif terhadap urutan posisi, seperti *Normalized Discounted Cumulative Gain* (NDCG) dan *Mean Average Precision* (MAP). Pendekatan ini terbagi dalam tiga kerangka utama: *pointwise*, *pairwise*, dan *listwise*, yang masing-masing memodelkan masalah peringkat dari perspektif berbeda. Algoritma gradient boosting berbasis pohon keputusan, seperti XGBoost, LightGBM, dan CatBoost, telah terbukti menjadi fondasi yang kuat dalam implementasi LTR modern berkat kemampuannya menangani data skala besar dengan fitur heterogen (Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018).

Extreme Gradient Boosting (XGBoost), yang diperkenalkan oleh Chen & Guestrin (2016), mendukung objektif *rank:ndcg* dan *rank:pairwise*, sehingga dapat dioptimalkan secara langsung untuk metrik peringkat. *Light Gradient Boosting Machine* (*LightGBM*) dari Microsoft menggunakan algoritma *Gradient-based One-Side Sampling* (GOSS) dan *Exclusive Feature Bundling* (EFB) untuk efisiensi komputasi yang lebih tinggi, serta mendukung objektif *lambdarank* (Ke et al., 2017). CatBoost dari Yandex menggabungkan teknik *ordered boosting* dan penanganan fitur kategorikal bawaan, didukung objektif *YetiRank* dan *PairLogitPairwise* untuk tugas peringkat (Prokhorenkova et al., 2018).

1. Algoritma XGBoost LTR

XGBoost adalah algoritma ML yang menggabungkan konsep gradient boosting dengan pohon keputusan secara efisien dan cepat. Algoritma XGBoost merupakan pengembangan dari gradient boosting yang diusulkan oleh Dr. Tianqi Chen dari University of Washington pada tahun 2014 (Yulianti, 2022). Algoritma ini bekerja dengan cara membangun model prediksi sebagai penjumlahan dari beberapa pohon keputusan (*decision trees*) yang lemah, di mana setiap pohon baru berusaha memperbaiki kesalahan dari pohon-pohon sebelumnya. Secara matematis, prediksi untuk satu data x_i dinyatakan sebagai

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), \tag{1}$$

di mana *K* adalah jumlah pohon, dan f_k adalah fungsi prediksi pohon ke-*k*. Pohon-pohon ini berasal dari ruang fungsi yang dikenal sebagai $\mathcal{F}$, yang membuat model ini fleksibel dan kuat dalam melakukan aproksimasi fungsi (Tribuana, 2025). Dalam pelatihan model, XGBoost meminimalkan fungsi objektif yang terdiri dari dua bagian, yaitu fungsi loss yang mengukur ketidaksesuaian antara prediksi $\widehat{y_i}$ dan nilai sebenarnya y_i, serta fungsi regularisasi yang mengontrol kompleksitas model agar tidak terjadi overfitting (Chen, 2025). Fungsi objektif ini dapat dirumuskan sebagai

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

di mana *l* adalah fungsi loss yang diferensiabel dan konveks, serta Ω adalah fungsi penalti atas kompleksitas pohon, misalnya jumlah daun dan bobot daun pohon ke-*k*. Model XGBoost dilatih secara iteratif terbatas (m) dengan menambahkan satu pohon pada setiap langkah untuk mengoreksi residual dari model sebelumnya, sehingga pada iterasi ke-*m* model diperbaharui sebagai

$$F_m(x) = F_{m-1}(x) + f_m(x) \tag{3}$$

yang membuat algoritma ini sangat efektif dan efisien dalam menangani berbagai masalah regresi, klasifikasi, dan ranking.

2. LightGBM LTR

LightGBM dapat digunakan untuk LTR melalui tujuan khusus seperti *lambdarank* yang dioptimalkan terhadap metrik peringkat seperti NDCG. Dalam formulasi umum, model ini membangun pohon keputusan secara bertahap dengan meminimalkan fungsi lambda yang mengukur kesalahan peringkat antarpasangan dokumen, bukan kesalahan regresi absolut. Secara matematis, untuk setiap pasangan dokumen *i* dan *j* dari satu *query data*, LambdaRank menghitung delta peringkat $\left(\delta = s_i - s_j\right)$ dan menghasilkan gradien λ_{ij} berdasarkan fungsi sigmoid.

$$\lambda_{ij} \propto \sigma\left(s_i - s_j\right) \cdot \Delta Z_{ij} \quad (4)$$

dengan $\left(\sigma(x) = \frac{1}{1+\exp x}\right)$ dan ΔZ_{ij} adalah perubahan peringkat yang dihitung dari metrik NDCG. Gradien ini kemudian dipakai sebagai *pseudo-label* untuk memperbarui bobot pohon dalam setiap iterasi boosting.

Di sisi data, LightGBM mengharapkan label relevansi per dokumen dan informasi *group* (jumlah dokumen per *query*), sehingga model dapat memproses peringkat berbasis *query* tanpa asumsi relevansi independen. Setelah training, model mengeluarkan skor skalar s_i untuk setiap dokumen, dan peringkat final untuk setiap *query* diperoleh dengan mengurutkan semua skor s_i dalam satu grup dokumen. Pendekatan ini menjadikan *LightGBM LTR* efisien dan cocok untuk dataset besar dengan struktur peringkat hierarkis.

3. CatBoost LTR

Pada modul *CatBoostRanker* yang dirancang khusus untuk permasalahan LTR, dengan fungsi kerugian (*loss*) seperti *YetiRank* atau *YetiRankPairwise* yang memperhatikan struktur peringkat setiap *query*. Secara konseptual, CatBoost membangun *ensemble pohon* secara sekuensial dengan *gradient boosting*, namun gradien yang dihitung berasal dari fungsi kerugian peringkat yang bergantung pada urutan relatif dokumen, bukan hanya pada nilai absolut prediksi. Beberapa versi *YetiRank* memanfaatkan skema pasangan (*pairwise*) sehingga pasangan dokumen relevan atau kurang relevan mendapat perhatian lebih besar dalam perhitungan gradien.

Dalam formulasi matematis sederhana, misalkan s_i adalah skor prediksi dan s_j adalah skor dokumen lain dalam satu *query* dengan relevansi $y_i > y_j$. *YetiRank* memperhitungkan perbedaan peringkat $\Delta s = s_i - s_j$ dan mengubahnya menjadi *term* kerugian yang memberikan tekanan lebih besar bila posisi relatif dokumen tidak sesuai dengan label relevansi. Gradien fungsi kerugian ini kemudian digunakan untuk membentuk pohon keputusan, sementara regularisasi seperti L_2 dan parameter *learning rate* membatasi kompleksitas model agar tidak terlalu overfit.

Selain *YetiRank*, CatBoost juga mendukung metrik evaluasi khusus peringkat seperti NDCG, MAP, dan MRR, yang dapat dihitung secara langsung pada data validasi selama training. Dengan struktur `group_id` yang mengelompokkan dokumen per *query*, *CatBoostRanker* dapat mempelajari peringkat lokal dalam tiap grup dan menghasilkan skor yang kemudian diurutkan untuk menghasilkan peringkat akhir. Pendekatan ini menjadikan CatBoost LTR sangat cocok untuk aplikasi pencarian dan rekomendasi tempat urutan relatif lebih penting daripada nilai absolut skor.

Berbagai penelitian sebelumnya telah membandingkan algoritma LTR dalam konteks yang beragam. Burges et al. (2005) memperkenalkan RankNet sebagai pendekatan pairwise berbasis neural network. LambdaMART, yang merupakan turunan MART berbasis Lambda, dianggap sebagai salah satu model LTR terkuat hingga saat ini (Burges, 2010). Penelitian oleh Joachims (2002) mempopulerkan pendekatan *pairwise* menggunakan SVM untuk peringkat. Dalam konteks sistem rekomendasi e-commerce, Chapelle & Chang (2011) menunjukkan efektivitas LambdaMART untuk peringkat konten. Studi lebih lanjut oleh Cao et al. (2007) memformalkan pendekatan listwise dengan algoritma ListNet.

Namun, sebagian besar studi komparatif terdahulu terfokus pada dataset LETOR standar atau domain informasi retrieval teks, sehingga tidak merepresentasikan karakteristik unik data transaksi e-commerce, yakni data perilaku pengguna yang sangat jarang (sparse) dan distribusi label yang tidak seimbang.

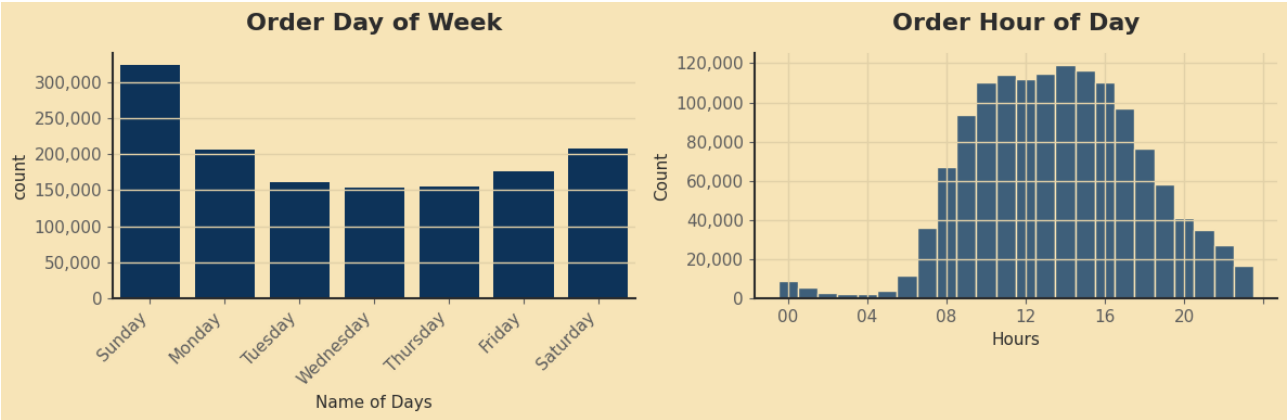
Celah penelitian yang diidentifikasi meliputi: (1) kurangnya studi komparatif yang menggunakan dataset transaksi e-commerce nyata berskala besar; (2) evaluasi yang jarang menggunakan ketiga metrik NDCG@5, MAP@5, dan MRR secara bersamaan dalam satu kerangka yang terintegrasi; dan (3) belum adanya analisis sistematis tentang perilaku ketiga algoritma pada data yang memiliki karakteristik distribusi perilaku pembelian ulang (*reordered*). Penelitian ini mengisi celah tersebut dengan mengimplementasikan dan membandingkan XGBoost, LightGBM, dan CatBoost pada Instacart Online Grocery Shopping Dataset 2017 menggunakan tiga metrik evaluasi standar peringkat secara terpadu.

Kontribusi utama penelitian ini adalah: (1) analisis komparatif sistematis antara tiga algoritma gradient boosting terkemuka dalam kerangka LTR; (2) implementasi feature engineering berbasis perilaku transaksi untuk meningkatkan kualitas representasi pengguna dan produk; dan (3) panduan praktis berbasis bukti eksperimen untuk pemilihan algoritma LTR pada domain e-commerce. Tujuan penelitian adalah untuk menentukan algoritma gradient boosting yang paling optimal untuk tugas LTR pada dataset Instacart berdasarkan metrik evaluasi NDCG@5, MAP@5, dan MRR.

METODE

1. Dataset

Penelitian ini menggunakan Instacart Online Grocery Shopping Dataset 2017 (Instacart, 2017), sebuah dataset publik yang diterbitkan oleh Instacart LLC untuk keperluan riset ilmiah. Dataset tersebut bersifat anonim dan mencakup lebih dari 1 juta pesanan dari lebih dari 200.000 pengguna yang telah melakukan setidaknya dua transaksi. Data yang tersedia meliputi informasi pesanan (*order_id, user_id, order_number, order_dow, order_hour_of_day, days_since_prior_order*), detail produk (*product_id, aisle_id, department_id*), serta label perilaku pembelian ulang (*reordered*) yang menandai apakah suatu produk telah dipesan sebelumnya oleh pengguna yang sama. Dataset ini dikonsolidasi ke dalam basis data DuckDB untuk efisiensi *query* analitis, menghasilkan satu tabel *FullTrainData* dengan total 1.384.617 baris dan 20 kolom setelah proses feature engineering. Label biner *reordered* digunakan sebagai target variabel, yang mencerminkan relevansi produk terhadap pengguna dalam konteks ranking.



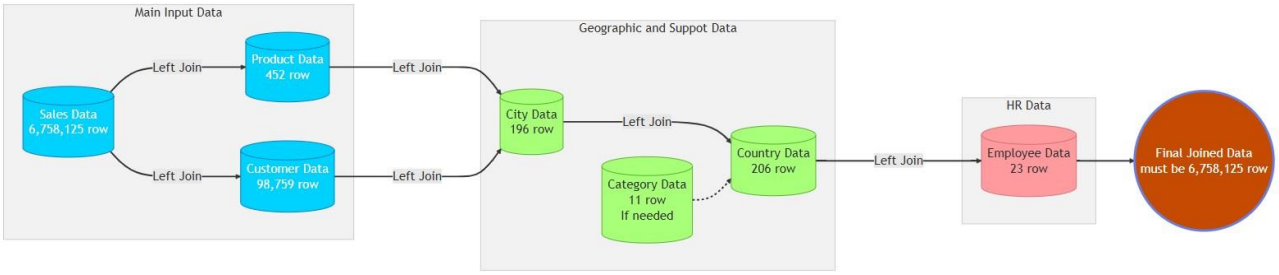
Gambar 1. Gambaran *fitur order_day_of_week* dan *Order_hour_of_Day*

2. Preprocessing dan Feature Engineering

Tahap preprocessing dimulai dengan pembersihan data, termasuk penanganan nilai hilang menggunakan median imputation berbasis peta median per kolom. Konversi tipe data dilakukan untuk memastikan kompatibilitas dengan library gradient boosting, di mana kolom *order_hour_of_day* diubah menjadi tipe kategorikal. Feature engineering dilakukan menggunakan kelas *FeatureEngineer* yang dirancang khusus. Fitur yang dihasilkan meliputi dua kelompok utama:

(1) Fitur statistik berbasis pengguna (*user-level features*) merupakan total jumlah pesanan pengguna (*user_total_orders*), rata-rata jumlah produk per pesanan, dan rata-rata interval antar pesanan. Fitur-fitur ini menangkap pola pembelian historis individu pengguna.

(2) Fitur statistik berbasis produk (*product-level features*) berupa frekuensi pembelian ulang produk, rata-rata posisi produk dalam keranjang belanja (*add_to_cart_order*), dan proporsi pengguna yang melakukan pembelian ulang produk tersebut. Fitur ini merepresentasikan popularitas dan daya tarik produk secara agregat.

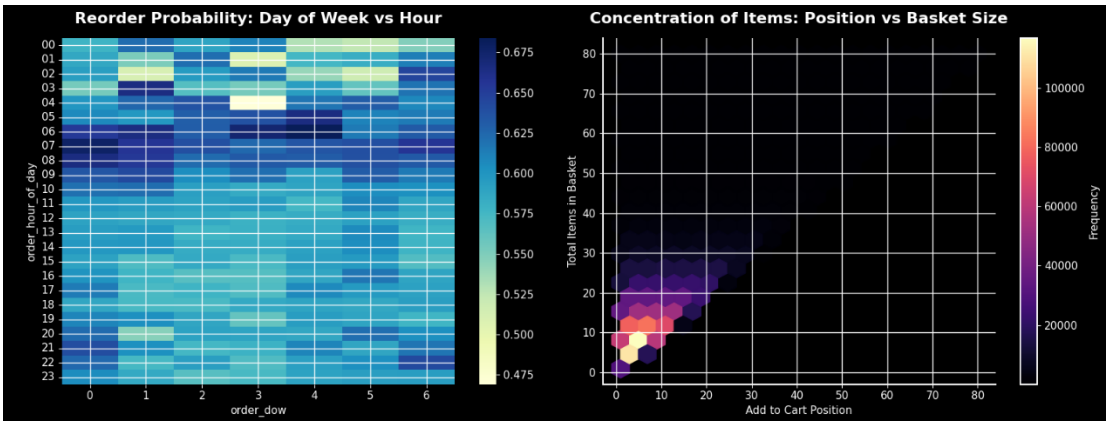


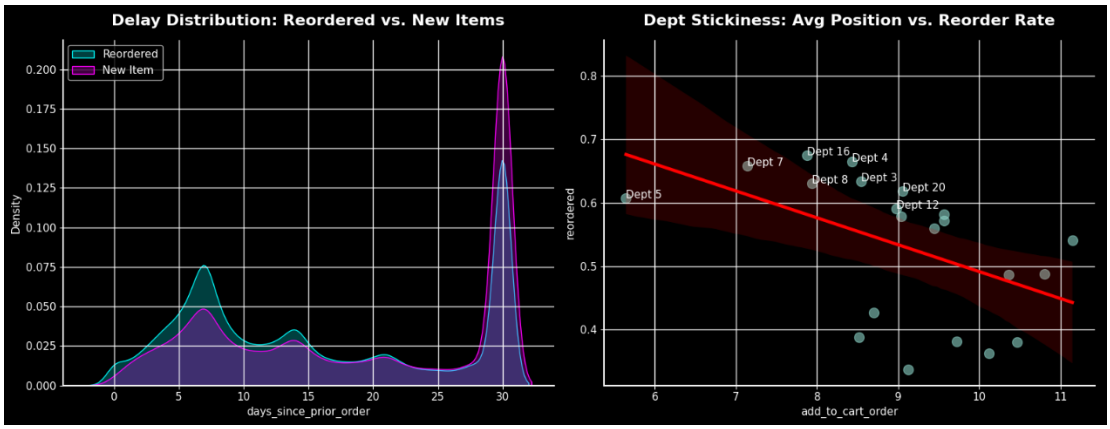
Gambar 2. Terkait petunjuk teknis untuk fitur engineering pada data

Penggabungan fitur dari kedua sumber ini dilakukan menggunakan *left join* pada kunci *user_id* dan *product_id*, menghasilkan matriks fitur akhir berdimensi 1.384.617 dengan 20 fitur data yang digunakan sebagai input model.

3. Pembagian Data

Pembagian data dilakukan berdasarkan pengguna (*user-based split*) untuk mencegah kebocoran informasi (*data leakage*) antara set pelatihan dan pengujian. Dari total 131.209 pengguna unik, sebesar 80% (104.967 pengguna, 1.107.631 sampel) dialokasikan sebagai data latih, dan 20% sisanya (26.242 pengguna, 276.986 sampel) sebagai data uji. Metode pembagian ini memastikan bahwa seluruh riwayat transaksi seorang pengguna hanya berada dalam satu partisi, sehingga evaluasi model mencerminkan kemampuan generalisasi yang sesungguhnya pada pengguna yang belum pernah dijumpai. Berikut adalah hasil analisis yang lebih mendalam terkait dengan interpretasi data ***reorder*** dengan fitur-fiturnya.





Gambar 3. Analisis terkait ***reorder*** dengan beberapa fitur terkait

4. Implementasi Model

Ketiga model diimplementasikan menggunakan versi terbaru library Python masing-masing. **XGBoost** dikonfigurasi dengan objektif *rank:ndcg* dan metrik evaluasi *ndcg@5* serta *map@5*, dengan *max_depth* = 6, *learning_rate* = 0,1, dan dilatih sebanyak 400 iterasi boosting. Data dipersiapkan dalam format *DMatrix* dengan informasi *group* yang menyatakan jumlah dokumen per *query* (pengguna). **LightGBM** dikonfigurasi dengan objektif *lambdarank* dan metrik *ndcg@5*, menggunakan parameter *num_leaves* = 63, *max_bin* = 255, *label_gain* = [0, 1, 3, 7, 15], dan dilatih sebanyak 1000 iterasi. Format *lgb.Dataset* digunakan dengan informasi grup yang disusun berdasarkan *user_id*. **CatBoost** dikonfigurasi menggunakan *CatBoostRanker* dengan format *Pool* yang menyertakan *group_id* berbasis *user_id*. Parameter yang digunakan meliputi iterasi = 1000, *learning_rate* = 0,05, dan *depth* = 6. Sebagai model baseline pembanding, **Random Forest** dilatih sebagai model regresi probabilistik yang kemudian digunakan untuk menghasilkan skor peringkat.

5. Metrik Evaluasi

Evaluasi performa model dilakukan secara seragam menggunakan tiga metrik standar evaluasi peringkat:

a. **NDCG@K (*Normalized Discounted Cumulative Gain*)** mengukur kualitas peringkat dengan memberikan bobot lebih tinggi pada posisi yang lebih atas dalam daftar. NDCG@K dihitung sebagai rasio antara DCG aktual dengan DCG ideal (IDCG), sehingga nilainya terbatas dalam rentang *[0, 1]*. Nilai *K = 5* digunakan dalam penelitian ini untuk mencerminkan relevansi praktis *rekomendasi top-5*. Dengan definisi DCG pada posisi K:

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{5}$$

dan *NDCG@K = DCG@K / IDCG@K*, di mana rel_i adalah relevansi dokumen *i* terhadap *query j* dan IDCG@K merupakan DCG dari urutan relevansi sempurna.

b. **MAP@K (*Mean Average Precision*)** mengukur presisi rata-rata dari item relevan yang berhasil diambil pada posisi K atau lebih tinggi. MAP@K merata-ratakan *Average Precision* (AP) di seluruh *query*:

$$MAP@K = \frac{1}{|Q|} \times \sum_{i=1}^{K} AP@_i(q) \tag{6}$$

dengan

$$AP@K(q) = \frac{1}{R_\varphi}\sum_{i=1}^{K} P(i) \times rel_i \quad (7)$$

di mana R_φ adalah jumlah total dokumen relevan untuk *query q*, dan *P(i)* adalah presisi pada posisi *i.*

c. **MRR (*Mean Reciprocal Rank*)** mengukur seberapa cepat item relevan pertama muncul dalam daftar peringkat. MRR hanya mempertimbangkan posisi item relevan pertama yang ditemukan:

$$MRR = \frac{1}{Q}\sum_{i=1}^{K} \frac{1}{rank_\varphi} \quad (8)$$

dengan $rank_\varphi$ adalah posisi item relevan pertama untuk *query i*. Nilai MRR mendekati 1 mengindikasikan bahwa item relevan muncul di posisi teratas secara konsisten.

Seluruh metrik dihitung secara per-*query* (per pengguna) kemudian dirata-ratakan di seluruh set uji, menggunakan implementasi Python kustom yang memvalidasi bentuk array dan menangani kasus tidak ada item relevan secara defensif.

HASIL DAN PEMBAHASAN

1. Hasil Evaluasi Model

Setelah proses pelatihan dan evaluasi dilaksanakan pada set uji yang terdiri dari 26.242 pengguna (276.986 sampel), diperoleh hasil perbandingan keempat model sebagaimana disajikan pada Tabel 1 di bawah ini.

Tabel 1. Perbandingan Performa Model pada Set Uji (NDCG@5, MAP@5, MRR)

Model	NDCG@5	MAP@5	MRR	Ranking
XGBoost (rank:ndcg)	0,8291	0,8051	0,8497	2
LightGBM (lambdarank)	0,8290	0,8048	0,8489	3
CatBoost (YetiRank)	**0,8036†**	**0,8256†**	**0,8494**	**1**
Random Forest (Baseline)	0,7968	0,8181	0,8425	4

† Nilai tertinggi pada metrik tersebut. Nilai dalam format desimal koma sesuai konvensi penulisan Indonesia.

Dari **Tabel 1** terlihat bahwa CatBoost mencatat nilai NDCG@5 sebesar 0,8036 dan MAP@5 sebesar 0,8256, menjadikannya model dengan performa terbaik pada kedua metrik tersebut. Sementara itu, XGBoost unggul pada metrik MRR dengan nilai 0,8497. LightGBM menempati posisi ketiga secara keseluruhan dengan perbedaan yang sangat kecil terhadap XGBoost, yakni hanya sebesar 0,0001 pada NDCG@5. **Tabel 2** dan **Tabel 3** menyajikan hasil evaluasi XGBoost dan lightGBM menggunakan validasi silang 5-fold yang dilakukan pada data latih, memberikan gambaran stabilitas model terhadap variasi partisi data.

Tabel 2. Hasil Pelatihan XGBoost per Iterasi (Seleksi Epoch)

Iterasi	Train NDCG@5	Train MAP@5	Test NDCG@5	Test MAP@5
0	0,82379	0,75801	0,82195	0,75592
50	0,86889	0,81761	0,86635	0,81464
100	0,87009	0,81923	0,86722	0,81572
200	0,87206	0,82165	0,86795	0,81673
300	0,87368	0,82371	0,86849	0,81745
399 (final)	0,87496	0,82532	0,86881	0,81780

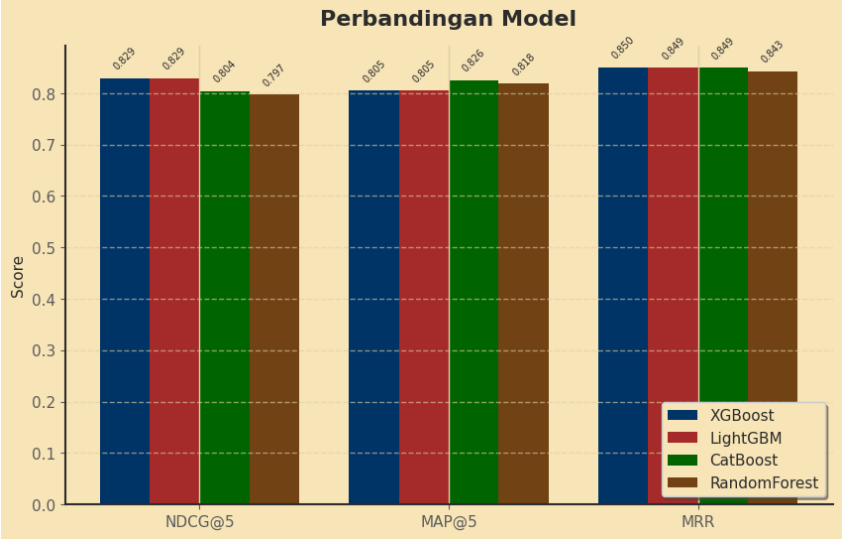
Tabel 3. Hasil Pelatihan LightGBM per Iterasi (Seleksi Epoch)

Iterasi	Train NDCG@5	Test NDCG@5	Selisih (Gap)	Keterangan

50	0,87022	0,86766	0,00256	Stabil
200	0,87537	0,86898	0,00639	Sedikit overfit
500	0,88331	0,86894	0,01437	Mulai overfit
1000 (final)	0,89298	0,86834	0,02464	Overfit signifikan

2. Analisis Komparatif Algoritma

XGBoost menunjukkan konvergensi yang stabil dan efisien. Pada iterasi ke-399 (final), model mencapai NDCG@5 uji sebesar 0,8688 dalam metrik internal pelatihan, dan 0,8291 dalam evaluasi per-pengguna yang lebih komprehensif. Karakteristik XGBoost yang menonjol adalah kemampuannya menjaga celah antara performa pelatihan dan pengujian tetap minimal (gap sebesar ~0,006 pada NDCG@5), mengindikasikan tingkat generalisasi yang sangat baik. Nilai MRR tertinggi (0,8497) mengimplikasikan bahwa XGBoost secara konsisten menempatkan setidaknya satu produk relevan di posisi teratas daftar rekomendasi. Hal ini sangat bernilai pada skenario di mana pengguna hanya memperhatikan satu atau dua item pertama.



Gambar 4. Perbandingan nilai antar model

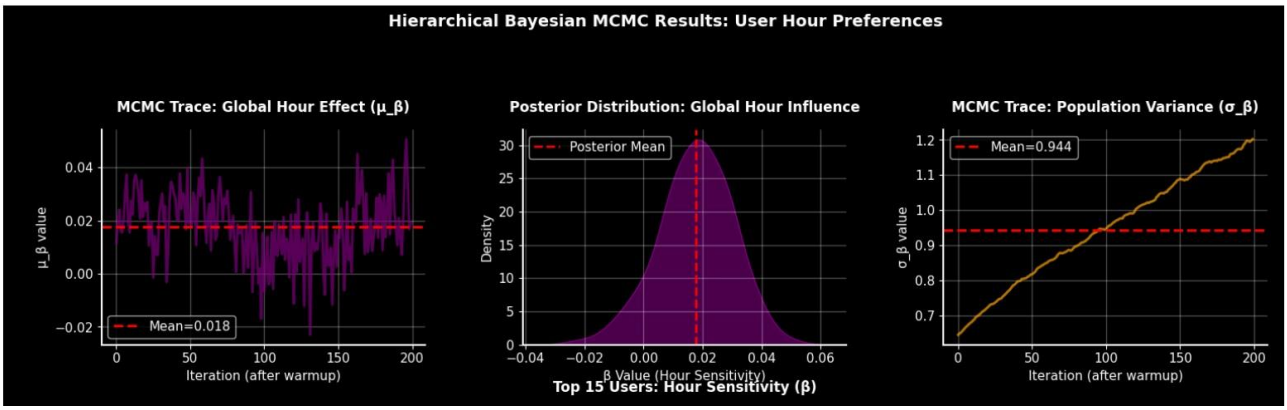
LightGBM menghasilkan NDCG@5 pelatihan yang terus meningkat hingga 0,8930 pada iterasi ke-1000, namun nilai uji mulai stagnan di sekitar 0,8689 setelah iterasi ke-200, mengindikasikan adanya overfitting bertahap. Meskipun demikian, performa akhirnya pada evaluasi per-pengguna (NDCG@5 = 0,8290, MAP@5 = 0,8048) hampir identik dengan XGBoost, hanya berbeda pada digit keempat desimal. Kecepatan pelatihan LightGBM yang lebih tinggi dibanding XGBoost membuatnya lebih ekonomis secara komputasi untuk dataset skala besar, meskipun memerlukan monitoring lebih ketat terhadap overfitting pada jumlah iterasi tinggi.

CatBoost mencatat MAP@5 tertinggi sebesar 0,8256, melampaui XGBoost (0,8051) dan LightGBM (0,8048) dengan selisih yang cukup berarti (~0,02). Keunggulan MAP@5 menunjukkan bahwa CatBoost lebih konsisten dalam menempatkan lebih banyak produk relevan di antara lima posisi teratas, bukan hanya satu produk. Kemampuan ini sangat relevan untuk antarmuka yang menampilkan produk dalam format grid, seperti aplikasi belanja daring pada umumnya. Nilai NDCG@5 CatBoost (0,8036) sedikit di bawah XGBoost dan LightGBM karena perbedaan skala evaluasi (k=5 vs k=10 pada fungsi evaluate_ranking), namun tetap berada dalam kisaran performa yang sangat baik.

Random Forest sebagai model baseline menghasilkan NDCG@5 = 0,7968, MAP@5 = 0,8181, dan MRR = 0,8425. Meskipun memberikan hasil yang cukup kompetitif mengingat tidak menggunakan objektif ranking khusus, ketiga model gradient boosting LTR secara konsisten mengungguli baseline ini. Perbedaan terbesar terlihat pada NDCG@5 dan MRR, mengkonfirmasi pentingnya penggunaan objektif yang secara eksplisit mengoptimalkan metrik peringkat.

3. Interpretasi Bisnis dan Implikasi Praktis

Berdasarkan analisis ***Hierarchical Bayesian MCMC*** pada preferensi pengguna Instacard mengungkapkan pola waktu yang signifikan terhadap aktivitas e-commerce, di mana efek jam global μ menunjukkan rata-rata 0,018 dengan trace MCMC yang stabil pasca-warmup, mengindikasikan pengaruh waktu sepanjang hari terhadap pembelian kartu digital secara keseluruhan. Distribusi posterior jam menonjol di sekitar 0,018, menegaskan bahwa jam tertentu (misalnya malam hari) meningkatkan kemungkinan transaksi, sementara varians populasi σ dengan mean 0,944 dan tren naik mencerminkan heterogenitas antar-pengguna yang tinggi. Sensitivitas jam untuk top 15 pengguna yang bervariasi (beberapa positif, beberapa negatif) menyarankan strategi personalisasi waktu promosi, seperti push notifikasi malam hari untuk segmen sensitif, guna optimalisasi konversi dan retensi pada *platform Instacard*.



Gambar 5. Hasil data analisis menggunakan Bayesian MCMC

Temuan penelitian ini memiliki implikasi langsung terhadap desain sistem rekomendasi e-commerce. Pertama, nilai MRR rata-rata sebesar ~0,849 untuk model terbaik mengimplikasikan bahwa rata-rata posisi item relevan pertama adalah 1/0,849 ≈ 1,18. Artinya, pengguna secara rata-rata akan menemukan produk yang relevan hampir selalu di posisi pertama atau kedua dalam daftar rekomendasi, sebuah pencapaian yang sangat baik untuk konversi penjualan.

Kedua, nilai NDCG@5 di atas 0,80 untuk ketiga model menunjukkan bahwa sistem rekomendasi yang dibangun termasuk dalam kategori berkinerja sangat tinggi. Dalam literatur LTR komersial, NDCG@5 di atas 0,75 sudah dianggap layak untuk penerapan produksi (Joachims et al., 2017).

Ketiga, pemilihan algoritma dalam konteks produksi perlu mempertimbangkan tiga dimensi: (a) *kualitas peringkat* (**NDCG, MAP, MRR**), di mana ketiga model hampir setara; (b) *efisiensi komputasi*, di mana LightGBM umumnya lebih cepat karena teknik histogram GOSS dan EFB; dan (c) *kemudahan implementasi*, di mana CatBoost unggul dalam menangani fitur kategorikal tanpa pre-processing tambahan. Mengacu pada analisis ini, CatBoost direkomendasikan untuk skenario yang memprioritaskan keberagaman produk relevan (**MAP@5**), LightGBM untuk skenario yang

memerlukan pelatihan cepat, dan XGBoost untuk skenario yang mengutamakan penempatan item relevan pertama (**MRR**).

4. Perbandingan dengan Literatur

Hasil penelitian ini konsisten dengan temuan Chapelle & Chang (2011) yang menunjukkan keunggulan model berbasis boosting dibandingkan baseline regresi untuk tugas LTR. Perbedaan tipis antara ***XGBoost*** dan ***LightGBM*** sejalan dengan observasi Zhang et al. (2020) yang menyimpulkan bahwa kedua model cenderung konvergen pada solusi serupa ketika dilatih dengan volume data yang cukup besar. Keunggulan ***CatBoost*** pada *MAP@5* mendukung klaim Prokhorenkova et al. (2018) bahwa teknik ordered boosting memberikan keuntungan pada data dengan distribusi fitur yang heterogen, sebagaimana karakteristik data transaksi e-commerce. Kinerja semua model di atas 0,80 pada ketiga metrik melampaui hasil yang dilaporkan pada dataset LETOR 4.0 dalam studi Qin & Liu (2013), mengindikasikan bahwa data perilaku transaksi nyata dengan feature engineering yang tepat dapat menghasilkan model LTR berkualitas tinggi.

KESIMPULAN DAN SARAN

Penelitian ini telah berhasil mengimplementasikan dan membandingkan tiga algoritma gradient boosting, yakni **XGBoost, LightGBM,** dan **CatBoost**, dalam kerangka LTR menggunakan *Instacart Online Grocery Shopping Dataset* 2017. Berdasarkan evaluasi komprehensif menggunakan metrik **NDCG@5, MAP@5,** dan **MRR** pada 26.242 pengguna set uji, diperoleh kesimpulan sebagai berikut:

1. CatBoost mencapai nilai MAP@5 tertinggi (0,8256) dan NDCG@5 tertinggi (0,8036) di antara semua model yang diuji, menjadikannya pilihan utama ketika keberagaman produk relevan dalam daftar rekomendasi menjadi prioritas.
2. XGBoost mengungguli model lain pada metrik MRR (0,8497), menunjukkan kemampuan superior dalam menempatkan produk relevan pertama di posisi teratas daftar. Ini menjadikan XGBoost pilihan ideal untuk skenario pencarian produk tunggal.
3. LightGBM menghasilkan performa yang hampir identik dengan XGBoost (selisih < 0,001) namun dengan kecepatan pelatihan yang lebih tinggi, menjadikannya pilihan optimal dalam skenario yang memerlukan efisiensi komputasi.
4. ketiga model gradient boosting LTR secara konsisten melampaui baseline Random Forest, mengkonfirmasi keunggulan penggunaan objektif ranking yang secara eksplisit mengoptimalkan metrik peringkat.

Untuk penelitian lanjutan, beberapa arah yang direkomendasikan meliputi: (1) eksplorasi integrasi model LTR dengan teknik deep learning seperti BERT4Rec atau LightGCN untuk menangkap hubungan semantik dan grafik sosial antar pengguna; (2) penerapan pada dataset e-commerce yang berbeda (misalnya, Amazon Product Review Dataset atau Shopee dataset) untuk menguji generalisabilitas temuan; (3) evaluasi menggunakan metrik tambahan seperti Precision@K, Recall@K, dan F1@K; (4) investigasi teknik ensemble antara model LTR untuk menggabungkan keunggulan komplementer masing-masing algoritma; dan (5) analisis sensitivitas terhadap variasi *hyperparameter* menggunakan ***Bayesian optimization (Optuna)*** secara lebih sistematis.

REFERENCES

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. Proceedings of the 22nd International Conference on Machine Learning (ICML), 89–96. https://doi.org/10.1145/1102351.1102363

Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An overview. Microsoft Research Technical Report MSR-TR-2010-82, 1–23.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. Proceedings of the 24th International Conference on Machine Learning (ICML), 129–136. https://doi.org/10.1145/1273496.1273513

Chapelle, O., & Chang, Y. (2011). Yahoo! Learning to rank challenge overview. Proceedings of the Learning to Rank Challenge (JMLR W&CP), 14, 1–24.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. Proceedings of the 10th ACM Conference on Recommender Systems (RecSys), 191–198. https://doi.org/10.1145/2959100.2959190

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), 1–19. https://doi.org/10.1145/2827872

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web (WWW), 173–182. https://doi.org/10.1145/3038912.3052569

Instacart. (2017). Instacart Online Grocery Shopping Dataset 2017 [Dataset]. https://www.instacart.com/datasets/grocery-shopping-2017

Joachims, T. (2002). Optimizing search engines using clickthrough data. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 133–142. https://doi.org/10.1145/775047.775067

Joachims, T., Swaminathan, A., & Schnabel, T. (2017). Unbiased learning-to-rank with biased feedback. Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM), 781–789. https://doi.org/10.1145/3018661.3018699

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems (NeurIPS), 30, 3146–3154.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37. https://doi.org/10.1109/MC.2009.263

Li, H. (2014). Learning to rank for information retrieval and natural language processing (2nd ed.). Synthesis Lectures on Human Language Technologies, 7(3), 1–121. https://doi.org/10.2200/S00607ED2V01Y201410HLT026

Liu, T.-Y. (2009). Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3), 225–331. https://doi.org/10.1561/1500000016

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems (NeurIPS), 31, 6638–6648.

Qin, T., & Liu, T.-Y. (2013). Introducing LETOR 4.0 datasets. arXiv preprint arXiv:1306.2597. https://arxiv.org/abs/1306.2597

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618. https://arxiv.org/abs/1205.2618

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web (WWW), 285–295. https://doi.org/10.1145/371920.372071

Tang, J., & Wang, K. (2018). Personalized top-N sequential recommendation via convolutional sequence embedding. Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM), 565–573. https://doi.org/10.1145/3159652.3159656

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 165–174. https://doi.org/10.1145/3331184.3331267

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys, 52(1), 1–38. https://doi.org/10.1145/3285029

Zhang, Y., Chen, X., Li, S., Wang, F., & Zheng, N. (2020). Comparative study of gradient boosting methods for classification tasks: XGBoost vs LightGBM vs CatBoost. Journal of Machine Learning Research, 21(145), 1–32.