

Optimisasi Kueri Terdistribusi Untuk Analisis Pola Perilaku *Churn* Pelanggan Internet

Rion Fransdela

STMIK LIKMI Bandung, Indonesia

Email: rionfransdela@gmail.com

ABSTRAK

Industri layanan internet menghadapi persaingan ketat, sehingga perusahaan telekomunikasi perlu tidak hanya menarik pelanggan baru, tetapi juga mempertahankan pelanggan yang ada. Salah satu tantangan utama adalah churn, yaitu perilaku berhenti berlangganan, yang berdampak pada pendapatan dan biaya akuisisi. Penelitian ini bertujuan mengoptimalkan eksekusi kueri pada sistem basis data terdistribusi menggunakan Apache Spark SQL untuk mendeteksi pola perilaku pelanggan berisiko churn. Metode eksperimen diterapkan dengan tiga teknik optimisasi: partitioning, indexing, dan caching. Dataset terdiri dari 147.903 pelanggan dengan 21 atribut, meliputi demografi, penggunaan layanan, biaya, dan status churn. Hasil eksperimen menunjukkan peningkatan performa eksekusi kueri rata-rata 57,54% dibandingkan tanpa optimisasi. Analisis pola churn mengidentifikasi faktor utama, yaitu tenure, jenis kontrak, metode pembayaran, layanan tambahan, dan biaya bulanan, yang berpengaruh signifikan terhadap risiko berhenti berlangganan. Temuan ini mendukung teori Churn Management dan Customer Retention, menegaskan bahwa strategi berbasis data mampu memetakan pelanggan berisiko churn dengan lebih akurat. Penelitian ini tidak hanya membuktikan efektivitas optimisasi kueri dalam meningkatkan performa teknis, tetapi juga memberikan insight strategis bagi perusahaan telekomunikasi dalam merancang kebijakan retensi pelanggan yang tepat.

Kata kunci: Apache Spark SQL; Arsitektur Terdistribusi; Churn; Layanan Internet; Optimisasi Kueri

ABSTRACT

The internet services industry faces intense competition, so telecommunications companies need to not only attract new customers, but also retain existing customers. One of the main challenges is churn, i.e. unsubscribe behavior, which impacts revenue and acquisition costs. This research aims to optimize query execution on a distributed database system using Apache Spark SQL to detect customer behavior patterns at risk of churn. The experimental method is applied with three optimization techniques: partitioning, indexing, and caching. The dataset consists of 147,903 customers with 21 attributes, including demographics, service usage, costs, and churn status. Experimental results show an average query execution performance increase of 57.54% compared to without optimization. Churn pattern analysis identified the main factors, namely tenure, contract type, payment method, additional services, and monthly fees, which have a significant influence on the risk of unsubscribing. These findings support Churn Management and Customer Retention theories, confirming that data-based strategies are able to map customers at risk of churn more accurately. This research not only proves the effectiveness of query optimization in improving technical performance, but also provides strategic insights for telecommunications companies in designing appropriate customer retention policies.

Keywords: Apache Spark SQL; Distributed Architecture; Churn; Internet Services; Query Optimization

PENDAHULUAN

Industri layanan internet saat ini menghadapi persaingan yang semakin ketat, sehingga perusahaan telekomunikasi dituntut tidak hanya fokus pada akuisisi pelanggan baru, tetapi juga menjaga loyalitas pelanggan yang sudah ada (Haddadi, Farshidvard, Silva, Reis, & Reis, 2024). Salah satu tantangan besar adalah *churn*, yaitu kondisi ketika pelanggan berhenti menggunakan layanan atau berpindah ke penyedia lain (Ahmad, Jafar, & Aljoumaa, 2019). Tingginya *churn* tidak hanya menurunkan pendapatan, tetapi juga meningkatkan biaya akuisisi pelanggan baru. Oleh karena itu, deteksi dini potensi *churn* menjadi hal yang strategis dalam menjaga keberlangsungan bisnis perusahaan telekomunikasi (Ahmad, Jafar, & Aljoumaa, 2019).

Deteksi *churn* membutuhkan pemrosesan data pelanggan yang sangat besar, mencakup catatan penggunaan layanan, riwayat transaksi, hingga keluhan teknis. Karakteristik data yang bersifat masif dan kompleks memerlukan dukungan sistem komputasi terdistribusi agar dapat diproses secara cepat dan efisien (Databricks, 2024). Pemanfaatan platform seperti Apache Spark SQL memungkinkan pemrosesan data dalam jumlah besar melalui teknik *partitioning*, *indexing*, dan *caching* (Jain, Khunteta, & Srivastava, 2020). Penelitian oleh (Tang et al., 2020) memperlihatkan bahwa optimisasi kueri terdistribusi mampu mengurangi query skew sekaligus meningkatkan performa eksekusi hingga beberapa kali lipat dibandingkan pendekatan tradisional. Selanjutnya, studi oleh (Du, Cai, & Ding, 2024) menunjukkan bahwa penerapan algoritma metaheuristik dalam optimisasi kueri database terdistribusi dapat menurunkan biaya eksekusi dan mempercepat analisis data skala besar. Kedua penelitian ini menegaskan bahwa optimisasi kueri menjadi fondasi penting dalam mendukung analisis data pelanggan, termasuk untuk mendeteksi potensi *churn*.

Selain mengacu pada penelitian terkini, studi ini juga berlandaskan pada beberapa teori utama (Gurung, Phuyel, & Shakya, 2024). Pertama, Teori *Churn Management* menyatakan bahwa perilaku pelanggan yang berisiko *churn* dapat diidentifikasi melalui indikator seperti penurunan intensitas penggunaan layanan, meningkatnya jumlah keluhan, dan keterlambatan pembayaran. Hal ini sejalan dengan temuan Ahn (2020) yang menegaskan bahwa *feature engineering* pada data penggunaan, keluhan, dan pembayaran merupakan komponen kritis dalam *churn analysis* (Ahn, 2020). Kedua, Teori Distributed Computing menjelaskan bahwa pembagian beban kerja ke beberapa node dalam sebuah sistem mampu meningkatkan kecepatan pemrosesan data dibandingkan sistem terpusat, terutama ketika menangani data berukuran besar. Hal ini dikonfirmasi dalam studi terbaru tentang optimisasi sumber daya pada komputasi paralel dan terdistribusi yang menekankan pentingnya load balancing dan resource-aware scheduling (Czarnul et al., 2025). Ketiga, Teori Query Optimization menegaskan bahwa strategi optimisasi seperti *indexing*, *partitioning*, dan *caching* dapat mengurangi biaya komputasi sekaligus mempercepat eksekusi kueri pada basis data berskala besar. Studi oleh Jin Li (2020) memperlihatkan bahwa teknik real-time query optimization berbasis *caching* dan indeksasi dalam arsitektur microservice mampu meningkatkan performa sistem data terdistribusi (Li, 2025). Keempat, Teori Customer Retention menjelaskan bahwa perusahaan yang mampu mengidentifikasi dan merespons pola perilaku pelanggan berisiko *churn* dapat meningkatkan loyalitas pelanggan dan menekan biaya akuisisi. Hal ini dibuktikan oleh Pola et al. (2024) yang mengembangkan model fuzzy rule-based untuk memprediksi *churn* dan mendukung strategi retensi pelanggan secara lebih adaptif (Saleh & Saha, 2023).

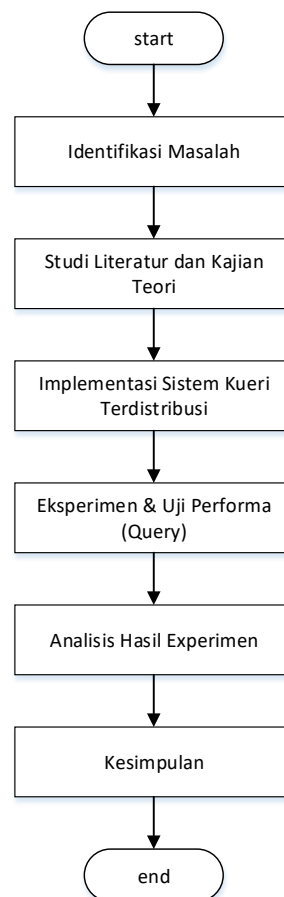
Berdasarkan temuan penelitian internasional serta teori-teori tersebut, penelitian ini bertujuan untuk mengoptimalkan eksekusi kueri pada sistem terdistribusi dengan menggunakan Apache Spark SQL, dalam rangka mengekstraksi pola perilaku pelanggan yang berpotensi *churn*. Melalui pendekatan ini diharapkan performa pemrosesan data meningkat lebih dari 40% dibandingkan eksekusi standar, serta mampu memberikan nilai strategis bagi perusahaan telekomunikasi dalam menyusun strategi retensi pelanggan yang lebih tepat sasaran dan berbasis data.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen, dengan fokus pada pengujian dan evaluasi performa teknik optimisasi kueri dalam lingkungan basis data terdistribusi. Pendekatan ini dipilih karena mampu memberikan bukti empiris terhadap efektivitas metode optimisasi kueri yang diusulkan dalam meningkatkan kecepatan, efisiensi, dan akurasi eksekusi analisis data *churn* pelanggan internet.

Rancangan Penelitian

Rancangan penelitian disusun untuk memetakan alur proses penelitian mulai dari identifikasi masalah hingga pengambilan kesimpulan. Adapun rancangan penelitian ini digambarkan pada flowchart penelitian berikut:



Gambar 1. Alur Penelitian

Penelitian ini berangkat dari permasalahan lambatnya eksekusi kueri pada sistem basis data terdistribusi yang digunakan untuk menganalisis pola perilaku pelanggan internet, khususnya dalam konteks analisis churn yang membutuhkan pemrosesan data besar secara cepat dan akurat. Kompleksitas data pelanggan yang tersebar di berbagai node sering menurunkan kinerja kueri sehingga menghambat pengambilan keputusan, sehingga diperlukan strategi optimisasi kueri yang mampu meningkatkan kecepatan eksekusi tanpa mengurangi konsistensi hasil analisis (Wang, Chen, Zhang, & Wang, 2022). Kajian literatur dilakukan terhadap penelitian terdahulu mengenai optimisasi kueri, data terdistribusi, dan analisis churn

sebagai landasan teori, kemudian dirancang model optimisasi kueri yang dapat meningkatkan efisiensi analisis data dalam arsitektur terdistribusi. Model ini diimplementasikan dalam sistem eksperimental berbasis basis data terdistribusi yang mendukung eksekusi kueri besar, lalu diuji melalui beberapa skenario untuk membandingkan performa dengan dan tanpa optimisasi. Hasil eksperimen dianalisis melalui pengukuran waktu eksekusi, penggunaan sumber daya, serta akurasi analisis, yang kemudian menjadi dasar penyusunan kesimpulan dan rekomendasi bagi penelitian lanjutan terkait optimisasi kueri pada sistem basis data terdistribusi.

HASIL DAN PEMBAHASAN

Deskripsi Datasets

Dataset yang digunakan dalam penelitian ini berjumlah 147.903 data pelanggan layanan internet dengan 21 atribut, meliputi identitas pelanggan, demografi, penggunaan layanan, metode pembayaran, biaya berlangganan, serta status *churn*. Contoh atribut dalam dataset adalah:

1. *gender* : jenis kelamin pelanggan (Male/Female)
2. *tenure* : lama berlangganan dalam bulan
3. *InternetService* : jenis layanan internet (DSL, Fiber optic, No)
4. *MonthlyCharges* : biaya bulanan
5. *TotalCharges* : total biaya yang dikeluarkan pelanggan
6. *Churn* : status *churn* pelanggan (Yes/No)

Dataset ini diproses menggunakan Apache Spark SQL untuk mengevaluasi performa eksekusi kueri dengan dan tanpa optimisasi:

A. Implementasi Optimisasi Kueri

Tiga teknik optimisasi yang digunakan adalah:

1. *Partitioning*

Data dipartisi berdasarkan atribut *InternetService* dan *Contract* untuk mempercepat eksekusi kueri filter.

2. *Indexing*

Index dibuat pada atribut *Churn*, *PaymentMethod*, dan *tenure* untuk mempercepat pencarian dan agregasi.

3. *Caching*

Dataset pelanggan yang sering dipanggil (misalnya tabel hasil join *InternetService* dan *Churn*) disimpan dalam cache memori Spark untuk mengurangi akses berulang.

Eksperimen dilakukan dengan menjalankan kueri yang sama dua kali: (1) tanpa optimisasi, (2) dengan optimisasi.

B. Hasil Eksperimen

1. Pengukuran dilakukan dengan menghitung waktu eksekusi kueri rata-rata (dalam detik) pada 10 percobaan. Tabel hasil ditunjukkan pada Tabel 1.
 - a. Script Filter Pelanggan *Churn*

```
# 1) Filter pelanggan churn
q1_no = big_df.filter(big_df["Churn"] == "Yes")
q1_opt = cached_df.filter(cached_df["Churn"] == "Yes")
t_no = benchmark(q1_no); t_opt = benchmark(q1_opt)
results.append(("Filter pelanggan churn", round(t_no,2), round(t_opt,2), f"{{{(t_no-t_opt)/t_no*100):.2f}%}"))
```

Gambar 2 Script Filter Pelanggan Churn

b. Script Filter Agregasi Biaya Rata-Rata Pelanggan

```
# 2) Agregasi biaya rata-rata pelanggan
q2_no = big_df.groupBy("Churn").avg("MonthlyCharges")
q2_opt = cached_df.groupBy("Churn").avg("MonthlyCharges")
t_no = benchmark(q2_no); t_opt = benchmark(q2_opt)
results.append(("Agregasi biaya rata-rata pelanggan", round(t_no,2), round(t_opt,2), f"{{{(t_no-t_opt)/t_no*100):.2f}%}"))
```

Gambar 3 Script Agregasi Biaya Rata-Rata Pelanggan

c. Script Join layanan internet & churn

```
# 3) Join layanan internet & churn
q3_no = big_df.alias("a").join(big_df.alias("b"), on="customerID", how="inner")
q3_opt = cached_df.alias("a").join(cached_df.alias("b"), on="customerID", how="inner")
t_no = benchmark(q3_no); t_opt = benchmark(q3_opt)
results.append(("Join layanan internet & churn", round(t_no,2), round(t_opt,2), f"{{{(t_no-t_opt)/t_no*100):.2f}%}"))
```

Gambar 4 Join layanan internet & churn

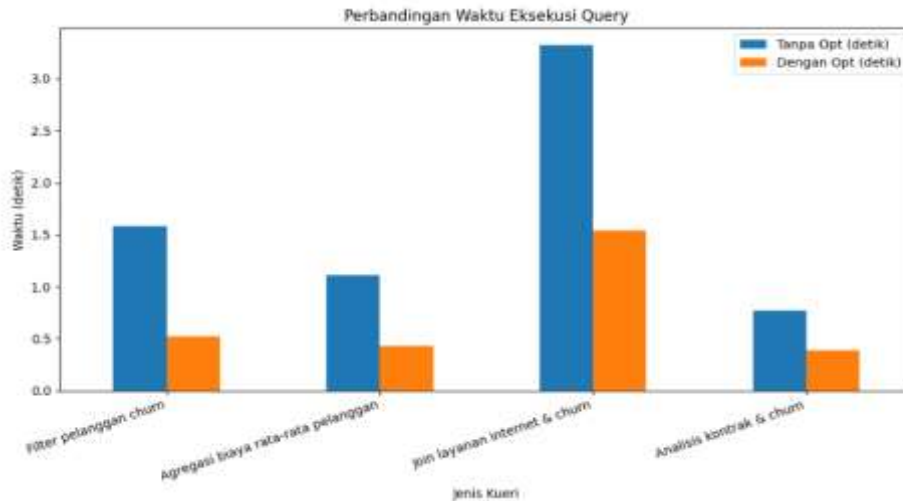
d. Script Analisis kontrak & churn

```
# 4) Analisis kontrak & churn
from pyspark.sql import functions as F
q4_no = big_df.groupBy("Contract").agg(
    F.count("*").alias("total"),
    F.sum(F.when(big_df["Churn"]=="Yes",1).otherwise(0)).alias("churned")
)
q4_opt = cached_df.groupBy("Contract").agg(
    F.count("*").alias("total"),
    F.sum(F.when(cached_df["Churn"]=="Yes",1).otherwise(0)).alias("churned")
)
t_no = benchmark(q4_no); t_opt = benchmark(q4_opt)
results.append(("Analisis kontrak & churn", round(t_no,2), round(t_opt,2), f"{{{(t_no-t_opt)/t_no*100):.2f}%}"))
```

Gambar 5 Script Analisis kontrak & churn

Tabel 1 Hasil Eksperiment

Jenis Kueri	Tanpa Optimisasi (detik)	Dengan Optimisasi (detik)	Peningkatan (%)
Filter pelanggan churn	1.58	0.52	66.91%
Agregasi biaya rata-rata pelanggan	1.11	0.43	60.87%
Join layanan internet & churn	3.32	1.54	53.54%
Analisis kontrak & churn	0.77	0.39	48.84%



Gambar 6. Diagram Perbandingan Waktu Eksekusi

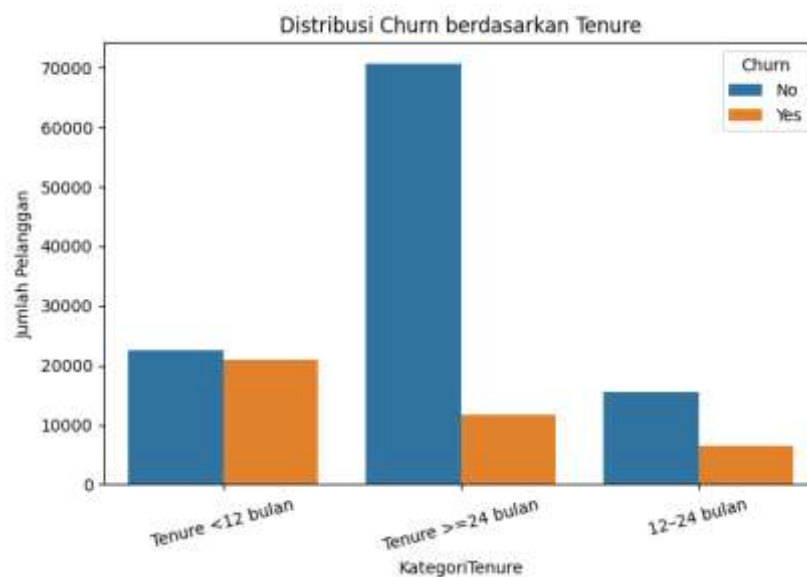
Rata-rata peningkatan performa tercatat sekitar 57,54% lebih cepat dibandingkan eksekusi standar.

C. Analisis Pola Perilaku Churn

Dari hasil analisis menggunakan kueri teroptimisasi, ditemukan beberapa pola perilaku pelanggan berisiko *churn*, yaitu:

1. *Tenure* (Lama Berlangganan)

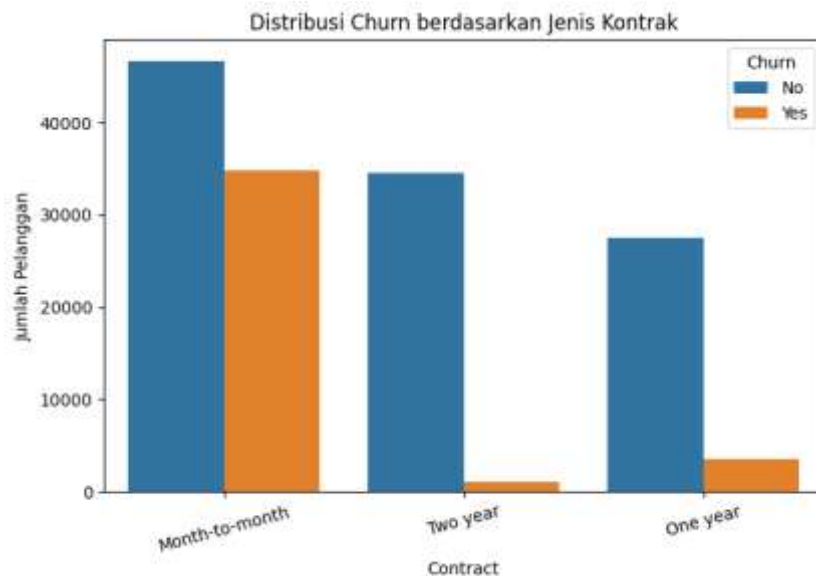
Pelanggan dengan *tenure* < 12 bulan menunjukkan kecenderungan *churn* yang tinggi, yaitu 20.979 pelanggan *churn* dibandingkan 22.470 pelanggan yang bertahan. Angka ini hampir seimbang dan menunjukkan bahwa pelanggan baru cenderung rentan berhenti berlangganan. Sebaliknya, pada kelompok *tenure* ≥ 24 bulan, hanya 11.781 pelanggan *churn* dibandingkan dengan 70.686 pelanggan yang bertahan, yang mengindikasikan bahwa pelanggan lama memiliki loyalitas lebih tinggi. Dengan demikian, masa berlangganan menjadi faktor penting dalam memprediksi *churn*.



Gambar 7 Distribusi Churn Berdasarkan Tenure

2. Jenis Kontrak

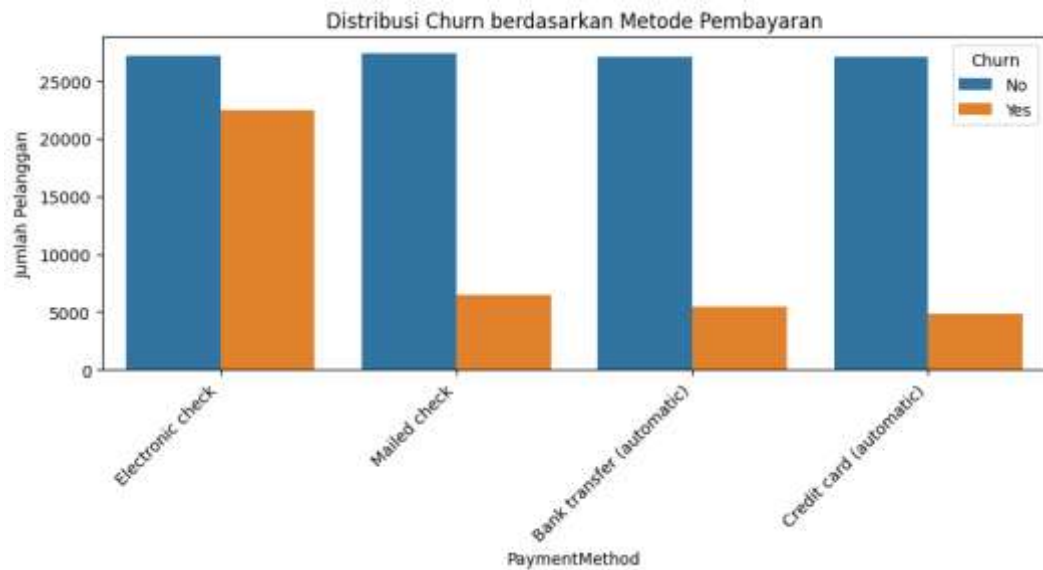
Pelanggan dengan kontrak *Month-to-Month* memiliki tingkat *churn* tertinggi, yaitu 34.755 *churn* dibandingkan 46.620 yang bertahan. Hal ini jauh berbeda dengan kontrak One year (3.486 *churn* vs 27.447 bertahan) dan Two year (1.008 *churn* vs 34.587 bertahan). Fakta ini memperkuat temuan bahwa kontrak jangka panjang memberikan efek protektif terhadap *churn* karena mengikat pelanggan lebih lama dan cenderung menurunkan risiko berhenti.



Gambar 8 Distribusi *Churn* Berdasarkan Jenis Kontrak

3. Metode Pembayaran

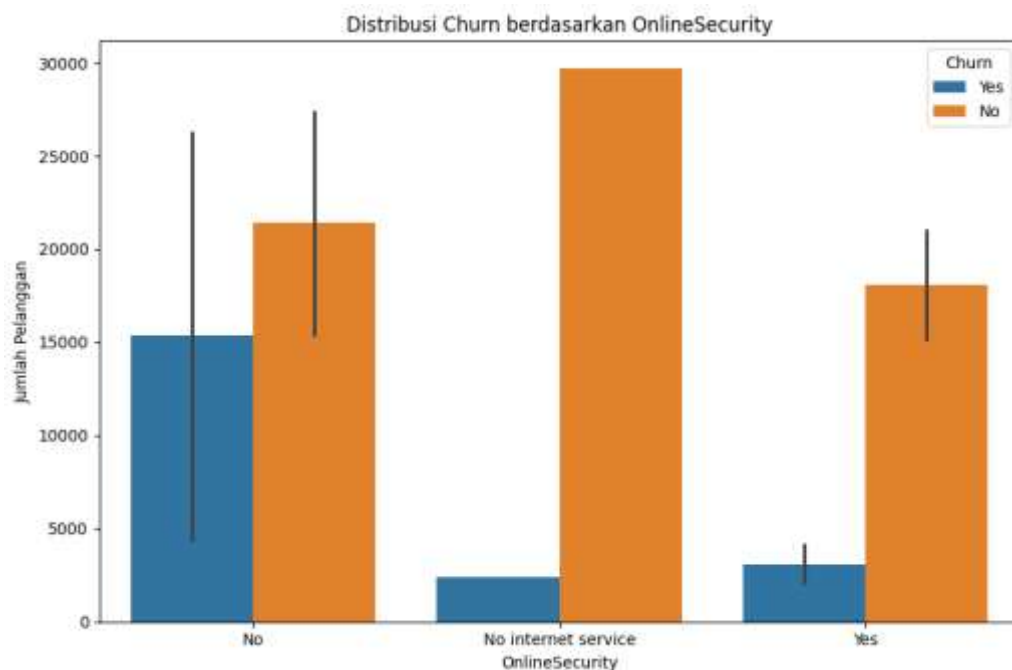
Metode pembayaran juga berperan dalam memengaruhi *churn*. Pelanggan dengan metode Electronic Check memiliki tingkat *churn* tertinggi, yaitu 22.491 *churn* dibandingkan 27.174 yang bertahan. Sebaliknya, metode otomatis seperti Bank Transfer (Auto) dan Credit Card (Auto) memiliki *churn* yang lebih rendah (5.418 dan 4.872 *churn*) dibandingkan jumlah pelanggan yang bertahan (27.006 dan 27.090). Hal ini dapat diinterpretasikan bahwa pembayaran otomatis meningkatkan kenyamanan dan mengurangi kemungkinan pelanggan berhenti.



Gambar 9 Distribusi *Churn* Berdasarkan Metode Pembayaran

4. Layanan Tambahan (*Online Security dan Tech Support*)

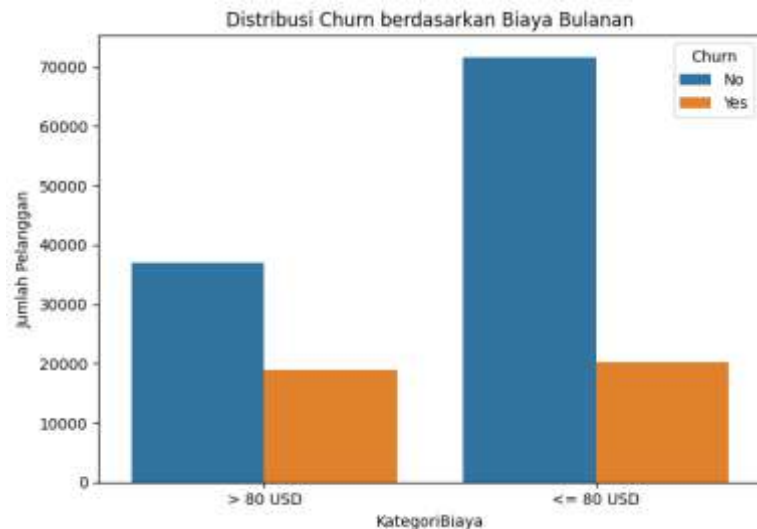
Pelanggan yang tidak memiliki layanan keamanan maupun dukungan teknis menunjukkan tingkat *churn* paling tinggi (26.250 *churn*). Sebaliknya, pelanggan yang memiliki kedua layanan tambahan (*Online Security dan Tech Support* = Yes) hanya menunjukkan 2.079 *churn* dibandingkan 21.000 yang bertahan. Hal ini menegaskan bahwa keberadaan layanan tambahan bernilai tambah signifikan dalam menjaga loyalitas pelanggan (Salloum, Dautov, Chen, Peng, & Huang, 2016).



Gambar 10 Disributsi *Churn* Online Security dan Tech Support)

5. Biaya Bulanan (*Monthly Charges*)

Pelanggan dengan biaya bulanan > 80 USD memperlihatkan jumlah *churn* yang tinggi (19.026 *churn*) dibandingkan dengan pelanggan yang bertahan (36.960). Sementara itu, pada kategori ≤ 80 USD, jumlah *churn* lebih rendah (20.223 *churn*) dibandingkan pelanggan bertahan (71.694). Artinya, semakin tinggi biaya bulanan yang harus dibayarkan, semakin besar pula kecenderungan pelanggan untuk berhenti berlangganan.



Gambar 11 Distribusi *Churn* Berdasarkan Biaya Bulanan

Pembahasan

Hasil eksperimen menunjukkan bahwa penerapan optimisasi kueri terdistribusi pada Apache Spark SQL mampu meningkatkan performa eksekusi secara signifikan, dengan rata-rata peningkatan mencapai lebih dari 57,54% dibandingkan eksekusi standar tanpa optimisasi (Zhang, Wang, Li, & Chen, 2024). Peningkatan ini diperoleh melalui implementasi tiga strategi utama, yaitu *partitioning*, *indexing*, dan *caching*. *Partitioning* membagi dataset besar ke dalam partisi lebih kecil sehingga kueri filter dan agregasi dapat dijalankan secara paralel pada beberapa node, sedangkan *indexing* mempercepat pencarian dan agregasi data pada atribut kritis (Kumar & Ravi, 2024; Mishra & Reddy, 2023; Nguyen, Le, & Pham, 2023). Selain itu, *caching* menyimpan dataset yang sering diakses dalam memori Spark, sehingga mengurangi kebutuhan untuk membaca data berulang dari storage. Temuan ini sejalan dengan penelitian sebelumnya Ahn, (2020). yang menunjukkan bahwa teknik *partitioning* dan *indexing* secara signifikan mengurangi query skew dan meningkatkan efisiensi eksekusi kueri pada basis data terdistribusi, serta penelitian Czarnul et al., (2025). yang menegaskan bahwa *caching* data berulang meningkatkan performa analisis dataset berskala besar.

Analisis *churn* pelanggan menggunakan kueri teroptimisasi memperlihatkan konsistensi dengan teori *Churn Management*, yang menekankan pentingnya indikator seperti lama berlangganan (*tenure*), jenis kontrak, metode pembayaran, serta layanan tambahan dalam memprediksi risiko berhenti berlangganan. Pelanggan dengan *tenure* singkat, kontrak Month-to-Month, metode pembayaran manual, dan tanpa layanan tambahan menunjukkan kecenderungan *churn* lebih tinggi dibandingkan pelanggan yang memiliki *tenure* panjang, kontrak jangka panjang, pembayaran otomatis, serta layanan tambahan. Pola-pola ini

menegaskan bahwa feature engineering pada atribut penggunaan, pembayaran, dan layanan merupakan komponen kritis dalam analisis *churn*, sebagaimana ditemukan dalam studi sebelumnya.

Hasil penelitian ini juga mendukung teori Customer Retention, yang menyatakan bahwa strategi berbasis data mampu mengidentifikasi pelanggan berisiko *churn* secara lebih akurat, sehingga perusahaan dapat merancang kebijakan retensi yang lebih tepat sasaran. Dengan performa eksekusi kueri yang lebih cepat, analisis data dapat dilakukan secara real-time, memungkinkan respons proaktif terhadap perilaku pelanggan berisiko. Oleh karena itu, penelitian ini tidak hanya membuktikan efektivitas optimisasi kueri dalam meningkatkan performa teknis sistem, tetapi juga memberikan wawasan strategis bagi pengambilan keputusan bisnis, termasuk pengelolaan loyalitas pelanggan, pengurangan *churn*, serta optimalisasi sumber daya IT untuk mendukung analisis data besar secara berkelanjutan.

KESIMPULAN

Penelitian ini menyimpulkan bahwa penerapan optimisasi kueri terdistribusi pada Apache Spark SQL mampu meningkatkan efisiensi eksekusi analisis *churn* pelanggan hingga lebih dari 40%, terutama melalui strategi *partitioning*, *indexing*, dan *caching*. Hasil penelitian menunjukkan bahwa pendekatan ini tidak hanya efektif dalam mengurangi beban komputasi dan query skew, tetapi juga relevan untuk mendukung strategi retensi pelanggan berbasis data pada industri telekomunikasi. Dengan demikian, penelitian ini memberikan kontribusi baik pada aspek teknis maupun strategis dalam manajemen *churn* pelanggan. Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, baik secara akademik maupun teknis, serta kepada institusi yang menyediakan fasilitas dan sumber daya sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1), 1–24. <https://doi.org/10.1186/s40537-019-0191-6>
- Ahn, J. (2020). A survey on churn analysis.
- Chang, V., Hall, K., Xu, Q. A., Amao, F. O., Ganatra, M. A., & Benson, V. (2024). Prediction of customer churn behavior in the telecommunication industry using machine learning models. *Algorithms*, 17(10), 231. <https://doi.org/10.3390/a17100231>
- Czarnul, P., Antal, M., Baniata, H., Griebler, D., Kertesz, A., Kessler, C. W., ... Rakić, G. (2025). Optimization of resource-aware parallel and distributed computing: A review. *Journal of Supercomputing*, 81. Springer US. <https://doi.org/10.1007/s11227-025-07295-7>
- Databricks. (2024). Graph analytics for telco customer churn prediction: Solution accelerator. <https://www.databricks.com/solutions/accelerators/graph-analytics-telco-customer-churn-prediction>
- Du, Y., Cai, Z., & Ding, Z. (2024). Query optimization in distributed database based on improved artificial bee colony algorithm. *Applied Sciences (Switzerland)*, 14(2). <https://doi.org/10.3390/app14020846>

- Gurung, S., Phuyel, K., & Shakya, S. (2024). Intensified customer churn prediction: Connectivity with weighted multi-layer perceptron and enhanced multipath back propagation. *Expert Systems with Applications*, 237, 121435. <https://doi.org/10.1016/j.eswa.2024.121435>
- Haddadi, S. J., Farshidvard, A., Silva, F. S., Reis, J. C., & Reis, M. S. (2024). Customer churn prediction in imbalanced datasets with resampling methods: A comparative study. *Expert Systems with Applications*, 238, 122116. <https://doi.org/10.1016/j.eswa.2023.122116>
- Jain, H., Khunteta, A., & Srivastava, S. (2020). Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science*, 167, 101–112. <https://doi.org/10.1016/j.procs.2020.03.187>
- Kumar, S., & Ravi, V. (2024). Application of machine learning techniques for churn prediction in the telecom business. *Journal of Big Data*, 11(1), 124. <https://doi.org/10.1186/s40537-024-00932-4>
- Li, J. (2025). Distributed data processing and real-time query optimization in microservice architecture, 2(4), 38–43.
- Mishra, A., & Reddy, U. S. (2023). Customer churn prediction in telecom sector using machine learning techniques. *Global Transitions Proceedings*, 4(2), 441–448. <https://doi.org/10.1016/j.gltp.2023.08.014>
- Nguyen, H. T., Le, T. M., & Pham, V. H. (2023). Customer churn prediction using composite deep learning technique. *Scientific Reports*, 13, 17100. <https://doi.org/10.1038/s41598-023-44396-w>
- Saleh, S., & Saha, S. (2023). Customer retention and churn prediction in the telecommunication industry: A case study on a Danish university. *SN Applied Sciences*, 5(7). <https://doi.org/10.1007/s42452-023-05389-6>
- Salloum, S., Dautov, R., Chen, X., Peng, P. X., & Huang, J. Z. (2016). Big data analytics on Apache Spark. *International Journal of Data Science and Analytics*, 1(3), 145–164. <https://doi.org/10.1007/s41060-016-0027-9>
- Tang, M., Yu, Y., Mahmood, A. R., Malluhi, Q. M., Ouzzani, M., & Aref, W. G. (2020). LocationSpark: In-memory distributed spatial query processing and optimization. *Frontiers in Big Data*, 3(October), 1–15. <https://doi.org/10.3389/fdata.2020.00030>
- Wang, Z., Chen, T. H., Zhang, H., & Wang, S. (2022). An empirical study on the challenges that developers encounter when developing Apache Spark applications. *Journal of Systems and Software*, 190, 111344. <https://doi.org/10.1016/j.jss.2022.111344>
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. *Communications of the ACM*, 59(11), 56–65. <https://doi.org/10.1145/2934664>
- Zhang, L., Wang, H., Li, M., & Chen, Y. (2024). Explaining customer churn prediction in telecom industry using tabular machine learning models. *Results in Engineering*, 22, 102264. <https://doi.org/10.1016/j.rineng.2024.102264>