MENINGKATKAN RETENSI PENGGUNA APLIKASI FITRACKER: SEGMENTASI PENGGUNA DENGAN IMPLEMENTASI *K-MEANS*

PROYEK UAS PEMBELAJARAN MESIN
KELAS

oleh
ANDRE ZULIANI
202131031

FAKULTAS TELEMATIKA ENERGI

INSTITUT TEKNOLOGI PERUSAHAAN LISTRIK NEGARA

JAKARTA

2023

Abstrak

Perkembangan teknologi telah menciptakan era dimana aplikasi digital memegang peranan sentral dalam kehidupan sehari-hari.Dalam konteks ini, pemahaman mendalam tentang perilaku pengguna sangat penting bagi perusahaan untuk meningkatkan retensi dan kepuasan pengguna. Penelitian ini dilakukan dengan tujuan untuk meningkatkan retensi pengguna aplikasi FitRacker melalui pendekatan segmentasi pengguna menggunakan teknik clustering mean.Peluncuran mencerminkan pentingnya memahami perilaku pengguna di tengah ketatnya persaingan di dunia aplikasi kebugaran.Ketika populasi pengguna semakin beragam, diperlukan strategi yang lebih canggih untuk memenuhi kebutuhan dan preferensi setiap pengguna. Penelitian ini berfokus pada identifikasi pola perilaku pengguna melalui segmentasi, dengan tujuan menciptakan pengalaman pengguna yang lebih personal dan memastikan keterlibatan yang berkelanjutan. Metode penelitian meliputi pengumpulan data melalui analisis penggunaan aplikasi, penerapan algoritma ukuran clustering, dan evaluasi hasil segmentasi. Pengguna dikategorikan ke dalam kelompok berdasarkan preferensi dan karakteristik yang serupa, yang mengarah pada pemahaman yang lebih mendalam tentang kelompok pengguna tertentu. Temuan kami menunjukkan potensi untuk mengembangkan strategi keterlibatan yang lebih efektif melalui pemahaman yang lebih mendalam tentang berbagai jenis pengguna. Hasil Segmentasi memungkinkan FitRacker mengembangkan solusi dan produk yang disesuaikan dengan kebutuhan setiap kelompok pengguna. Strategi retensi khusus dapat dikembangkan untuk meningkatkan keterlibatan pengguna dan mengurangi churn. Studi ini secara signifikan berkontribusi pada pemahaman yang lebih mendalam tentang perilaku pengguna aplikasi kebugaran dan membuka jalan bagi pengembangan strategi keterlibatan yang lebih efektif dan tepat sasaran.

Kata Kunci — Retensi Pengguna, Segmentasi Pengguna, Klustering Means, Aplikasi FitRacker, Personalisasi Pengalaman Pengguna.

Abstract

Technological developments have created an era where digital applications play an important role in everyday life. In this context, a deep understanding of user behavior is critical for companies to increase user retention and satisfaction. This research was conducted with the aim of increasing user retention of the FitRacker application through a user segmentation approach using the clustering mean technique. The launch reflects the importance of understanding user behavior amidst intense competition in the world of fitness applications. As the population of diverse users increases, more sophisticated strategies are needed to meet the needs and preferences of each user. This research focuses on understanding user behavior patterns through segmentation, with the goal of creating a more personalized user experience and ensuring continued engagement. Research methods include data collection through the use of analysis, application of clustering algorithms, and evaluation of segmentation results. Users are suggested into groups based on similar preferences and characteristics, leading to a deeper understanding of specific user groups. Our findings demonstrate the potential for developing more effective engagement strategies through a deeper understanding of different types of users. Segmentation results enable FitRacker to develop solutions and products tailored to the needs of each user group. Custom retention strategies can be developed to increase user engagement and reduce churn. This study significantly contributes to a deeper understanding of fitness app user behavior and paves the way for more effective and targeted engagement development strategies.

Keywords- *User Retention, User Segmentation, Clustering Means, FitRacker App, Personalization of User Experience.*

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Proyek UAS ini bertujuan untuk mengembangkan dan menguji model segmentasi pengguna pada aplikasi FitRacker untuk meningkatkan retensi pengguna. Motivasi utama di balik proyek ini terkait dengan kebutuhan mendalam untuk memahami perilaku pengguna dan memberikan pengalaman yang lebih personal. Dalam industri aplikasi kebugaran yang kompetitif, pemahaman mendalam tentang preferensi pengguna adalah kunci untuk meningkatkan retensi dan kepuasan.

Pemilihan algoritma pengelompokan merupakan langkah penting dalam proyek ini, dan tiga opsi yang diuji adalah K-Means, Agglomerative Hierarchical Clustering (AHC), dan Neural Networks (NN). Pengujian ini didasarkan pada banyak pertimbangan, termasuk kompleksitas algoritma, kemampuannya menangani data yang kompleks, dan kemampuannya memberikan hasil yang dapat diinterpretasikan.

Algoritma K-means dipilih karena dapat menangani data dalam jumlah besar dengan baik dan relatif sederhana untuk diimplementasikan. Kumpulan data yang digunakan dalam pengujian ini berisi data penggunaan aplikasi FitRacker dari waktu ke waktu. Data mencakup variabel seperti frekuensi penggunaan, jenis aktivitas fisik yang dilakukan, dan preferensi konten. Algoritma diuji pada kumpulan data ini untuk menentukan seberapa efektif setiap algoritma dalam mengidentifikasi pola perilaku pengguna.

Melalui pengujian ini, kami berharap dapat memahami keragaman perilaku pengguna dan mengidentifikasi algoritma pengelompokan yang paling tepat yang akan membantu FitRacker mengembangkan strategi retensi yang lebih bertarget. Dengan demikian, proyek ini berdampak positif terhadap peningkatan kualitas layanan aplikasi kebugaran dan pengalaman pengguna secara keseluruhan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan masalah beberapa masalah yang penting sebagai berikut:

1. Bagaimana pola perilaku pengguna dalam aplikasi FitRacker dapat diidentifikasi dan dipahami untuk meningkatkan retensi pengguna?
2. Apakah terdapat kelompok pengguna dengan preferensi dan kebiasaan yang serupa

3. dalam penggunaan aplikasi FitRacker?
4. Bagaimana kontribusi algoritma K-Means dalam mengelompokkan pengguna berdasarkan aktivitas mereka dalam aplikasi FitRacker?
5. Apa implikasi hasil segmentasi terhadap pengembangan strategi retensi pengguna yang lebih terfokus dalam FitRacker?
6. Apakah hasil pengujian algoritma klustering memberikan wawasan baru terhadap diversitas perilaku pengguna dalam konteks aplikasi kebugaran?

1.3 Tujuan

Berikut Tujuan Dari dibuatnya Proyek UAS ini untuk menyelesaikan rumusan masalah diatas:

1. Mengidentifikasi Pola Perilaku Pengguna.
2. Mengelompokkan Pengguna dengan Preferensi Serupa.
3. Mengembangkan Strategi Retensi yang Terfokus.
4. Meningkatkan Kualitas dan Personalisasi Layanan.
5. Memperkaya Wawasan terhadap Diversitas Perilaku Pengguna.

1.4 Manfaat

Berikut Manfaat dari dibuatnya Proyek UAS ini untuk memberi manfaat dari pandangan akademik maupun praktis:

1.4.1 Manfaat dari pandangan Akademik

1. Kontribusi terhadap Penelitian Kebugaran Digital.
2. Pengembangan Metode Klustering.
3. Penerapan Model Pembelajaran Mesin dalam Konteks Kesehatan.

1.4.2 Manfaat Dari pandangan Praktis

1. Peningkatan Retensi Pengguna.
2. Optimalisasi Layanan Berbasis Preferensi Pengguna .
3. Pengembangan Strategi Pemasaran yang Tepat Sasaran .
4. Keputusan Berbasis Data.

BAB II

KAJIAN PUSTAKA

2.1 Penelitian yang Relevan

Untuk memperkuat hasil penelitian, pada Bab ini berisikan tentang beberapa penelitian terdahulu yang akan dibahas sebagai pembanding serta pedoman dalam memahami dan merancang sebuah metode yang digunakan. Sebagai pembanding penelitian maka akan dirangkum penelitian terdahulu pada Tabel 2.1 sebagai berikut :

Tabel 2.1 Perbandingan Penelitian Dengan Penelitian yang Relevan

No.	09
Judul	The k-means Algorithm: A Comprehensive Survey and Performance Evaluation
Penulis	Ahmed, M., Seraj, R., & Islam, S. M. S.
Tahun	2020
Hasil	Dalam berbagai domain aplikasi, tugas analisis data sangat bergantung pada pengelompokan. makalah ini berfokus pada algoritma k-means yang populer dan masalah inisialisasi dan ketidakmampuan menangani data dengan jenis fitur campuran. Berbeda dengan makalah review atau survei lainnya, makalah ini memuat kedua hal tersebut secara kritis analisis literatur yang ada dan analisis eksperimental pada setengah lusin kumpulan data benchmark mendemonstrasikan kinerja berbagai varian k-means. Analisis eksperimental diungkapkan bahwa tidak ada solusi universal untuk permasalahan algoritma k-means; melainkan masing-masing yang ada varian algoritme bersifat spesifik aplikasi atau spesifik data. Penelitian masa depan kami akan fokus tentang pengembangan algoritma k-means yang kuat yang dapat mengatasi kedua masalah secara bersamaan. makalah ini juga akan membantu komunitas riset penambangan data untuk merancang dan

	mengembangkan jenis pengelompokan yang lebih baru algoritma yang dapat mengatasi masalah penelitian seputar Big Data.
Keterkaitan Penelitian	pada algoritma k-means dan tantangan inisialisasi. Hal ini relevan dengan implementasi k-means dalam FitRacker, di mana pemilihan algoritma dan penanganan inisialisasi sentroid dapat memengaruhi hasil segmentasi pengguna dan, pada gilirannya, retensi pengguna.

Tabel 2.2 Perbandingan Penelitian Dengan Penelitian yang Relevan

No.	08
Judul	*Unsupervised K-Means Clustering Algorithm*
Penulis	Sinaga & Yang
Tahun	(2020)

Hasil	Kami mengambil manfaat ketentuan penalti tipe entropi untuk membangun skema kompetisi. Algoritma U-k-means yang diusulkan menggunakan jumlah poin sebagai jumlah awal cluster untuk penyelesaian masalah inisialisasi. Selama iterasi, U-k-berarti algoritma akan membuang cluster tambahan, dan kemudian menjadi optimal jumlah cluster dapat ditemukan secara otomatis sesuai dengan struktur data. Keunggulan U-k-means ini gratis inisialisasi dan parameter yang juga kuat untuk berbeda volume dan bentuk cluster dengan menemukan secara otomatis jumlah cluster. Algoritma Uk-means yang diusulkan adalah dilakukan pada beberapa kumpulan data sintetis dan nyata dan juga dibandingkan dengan sebagian besar algoritma yang ada, seperti R-EM, C-FS, k-means dengan bilangan sebenarnya c, k-means+gap, dan Algoritma X-means. Hasilnya benar-benar menunjukkan hal tersebut keunggulan algoritma pengelompokan U-k-means.

Keterkaitan Penelitian	penggunaan jumlah poin sebagai inisialisasi cluster dalam U-k-means, yang menghasilkan penanganan inisialisasi yang lebih baik. Dalam implementasi k-means untuk FitRacker, penanganan inisialisasi dan parameter yang efisien dapat membantu meminimalkan dampak sensitivitas terhadap inisialisasi yang mungkin terjadi pada k-means biasa.

Tabel 2.3 Perbandingan Penelitian Dengan Penelitian yang Relevan

No.	02
Judul	*An extensive empirical comparison of k-means initialisation algorithms*
Penulis	Harris & De Amorim
Tahun	2022
Hasil	k-means yang sangat populer algoritma clustering, dan kekurangannya. Di sini, perhatian utama kami adalah bahwa pengelompokan yang dihasilkan oleh k-means sensitif terhadap kumpulan awalnya sentroid. Dengan kata lain, centroid awal yang tidak selaras dengan struktur data cenderung mendorong k-means menjadi lebih buruk. pengelompokan yang kurang optimal. Banyak algoritma telah diusulkan dengan tujuan tersebut menyediakan centroid awal yang baik untuk k-means. Kami melakukan studi berdampingan yang ekstensif yang membandingkan kinerja 17 algoritma tersebut dengan menggunakannya untuk menginisialisasi k-means pada 6.000 kumpulan data sintetis (di bawah konfigurasi berbeda), dan 28 kumpulan data dunia nyata.
Keterkaitan Penelitian	Keduanya mengindikasikan bahwa kelemahan k-means, khususnya terkait dengan inisialisasi sentroid yang buruk, dapat diatasi dengan menggunakan algoritma yang lebih baik. Dalam konteks FitRacker, implementasi k-means yang lebih baik dapat membantu meningkatkan segmentasi pengguna, yang pada gilirannya dapat berkontribusi pada meningkatkan retensi pengguna.

Tabel 2.4 Perbandingan Penelitian Dengan Penelitian yang Relevan

No.	01
Judul	*Decentralized and Adaptive K-Means Clustering for Non-IID Data Using HyperLogLog Counters*
Penulis	Soliman, A., Girdzijauskas, S., Bouguelia, M. R., Pashami, S., & Nowaczyk, S.
Tahun	2020
Hasil	pengelompokan k-means yang terdesentralisasi dan adaptif algoritma yang sangat bermanfaat untuk lingkungan yang dinamis dan terdistribusi penuh. Kontribusi utama kami adalah menyediakan metode k-means yang terdesentralisasi untuk distribusi data yang miring dan komputasi asinkron dalam jaringan P2P. Kami mengintegrasikan penghitung HyperLogLog dengan algoritma k-means kami secara efisien menangani kemiringan data dalam lingkungan eksekusi dinamis seperti itu. Selain itu, kami algoritma clustering memungkinkan node untuk secara individual menentukan jumlah cluster yang sesuai dengan data lokalnya. Evaluasi eksperimental kami menegaskan kemampuannya algoritma kami untuk beradaptasi dengan skenario sulit di mana k-means P2P yang ada metode gagal memberikan hasil yang dapat diterima
Keterkaitan Penelitian	gagasan umumnya tentang meningkatkan kinerja algoritma k-means dalam skenario yang sulit dapat menjadi relevan untuk peningkatan retensi pengguna dalam konteks segmentasi pengguna di FitRacker. Jika segmentasi dapat dilakukan secara lebih efisien dan akurat, hal ini dapat berkontribusi pada meningkatkan kepuasan dan retensi pengguna.

Tabel 2.5 Perbandingan Penelitian Dengan Penelitian yang Relevan

No.	06
Judul	*Parallel K-means Clustering Algorithm on NOWs*

Penulis	Kantabutra, S., & Couch, A. L.
Tahun	2000
Hasil	Kita dapat menyimpulkan bahwa K-means paralel kita algoritma mencapai efisiensi waktu 50%. kompleksitas. 50% relatif efisien dan biaya efektif jika kita mempertimbangkan sistem yang digunakan dalam hal ini paper adalah sistem penyampaian pesan berbasis Ethernet dan algoritma pengelompokan K-means beroperasi secara global secara alami. Dalam hal kompleksitas ruang, algoritma K-means paralel memiliki total yang sama kompleksitas O(N) sebagai kompleksitas serial Versi: kapan. Namun, versi paralel mengizinkannya untuk menggunakan ukuran masalah yang lebih besar karena itu sifat distributif. Algoritma paralel dapat diskalakan ukuran masalah hingga O(K) kali ukuran masalah pada satu mesin.
Keterkaitan Penelitian	algoritma K-means beroperasi secara global secara alami. Dalam konteks FitRacker, di mana segmentasi pengguna dan retensi pengguna dapat dipengaruhi oleh pola perilaku pengguna secara keseluruhan, pemahaman global dari algoritma K-means dapat menjadi faktor kunci dalam memberikan rekomendasi yang lebih baik.

2.2 Pembelajaran Mesin

Menurut IBM, Pembelajaran mesin adalah cabang ilmu komputer yang mempelajari bagaimana komputer dapat belajar dari data dan pengalaman tanpa diprogram secara eksplisit. Algoritma pembelajaran mesin dilatih untuk membuat klasifikasi dan prediksi seiring pengembangan data. Para ahli mengatakan pembelajaran mesin menggunakan statistik tanpa diprogram secara eksplisit untuk membantu menciptakan algoritma yang berguna.

Tom M. Mitchell, dalam bukunya Machine Learning, mendeskripsikan pembelajaran mesin sebagai “sebuah program yang dikatakan belajar dari pengalaman E dengan mengukur kinerjanya pada T, dan jika kinerjanya pada T meningkat seiring dengan E. menjelaskan bahwa program tersebut adalah sebuah program yang disebut "Belajar". Seiring kemajuan teknologi, pembelajaran mesin menjadi semakin penting dalam aplikasi seperti pengenalan wajah, pengenalan suara, dan analisis data bisnis.

Istilah machine learning pada dasarnya menjelaskan proses komputer dalam mempelajari data. Oleh karena itu, kita pasti akan terus bersinggungan dengan data ketika mempelajari machine learning. Data bisa saja sama, akan tetapi algoritma dan pendekatannya berbeda-beda untuk mendapatkan hasil yang optimal. Machine learning sendiri merupakan salah satu cabang dari disiplin dalam kecerdasan buatan (artificial intelligence) yang membahas pembangunan sistem berdasarkan data.

Menurut arthur samuel, mendefinisikan pengertian machine learning sebagai sebuah pertanyaan "how can computers learn to solve problems without being explicitly programmed?" yaitu bagaimana agar komputer dapat berjalan untuk memecahkan masalah sendiri tanpa harus diprogram secara eksplisit. Untuk pemrograman sendiri, ML biasanya menggunakan Library Python atau menggunakan R.

"Machine learning dapat diartikan secara luas sebagai teknik komputasi yang menggunakan pengalaman untuk meningkatkan kinerja atau membuat prediksi yang akurat". Mohri bersama temannya menjelaskan apa yang dimaksud dengan machine learning dengan menggunakan pengalaman untuk meningkatkan akurasi prediksi. yang meningkatkan Pengalaman di sini merujuk pada informasi historis (data pelatihan), biasanya dalam bentuk data elektronik yang telah dikumpulkan dan tersedia untuk dianalisis. Data ini dapat berupa set pelatihan digital berlabel manusia atau jenis informasi lain yang diperoleh melalui interaksi dengan lingkungan. Dalam kedua kasus tersebut, kualitas dan ukuran kumpulan data sangat penting untuk keberhasilan prediksi.

Menurut Pioneerlabs, machine learning merupakan domain ilmu komputer dengan basis matematika komputasi dan statistik yang dapat mempelajari pola dalam data untuk membuat prediksi masa depan. Dalam perkembangannya, machine learning dijalankan dengan tiga metode utama yaitu,

1. Supervised Learning

Metode supervised learning dilakukan dengan pemberian label pada dataset yang digunakan oleh machine learning dan diklasifikasikan oleh pengembang dengan memungkinkan algoritma melihat tingkat akurasi kinerjanya. Pengawasan machine learning dalam metode ini dilakukan oleh data berlabel yang nantinya membuat machine learning mempelajari apa hubungan dan ketergantungan antar data.

Cara kerja metode ini adalah memasukkan informasi sebagai input dan data berlabel sebagai hasil atau output. Input dalam machine learning pinjaman bank misalnya dapat berupa data rinci seperti usia, gaji, jumlah pinjaman, jumlah terutan, riwayat pinjaman, dan lain sebagainya. Sedangkan output-nya dapat

berupa hasil dari keseluruhan jumlah orang yang membayar pinjaman dan berapa jumlah orang gagal membayar.

2. Semi-supervised Learning (Unsupervised)

Metode semi-supervised learning bisa disebut juga sebagai metode machine learning tanpa pengawasan. Sehingga, prosesnya dilakukan pada dataset mentah yang tidak berlabel dan algoritma machine learning akan mencoba mengidentifikasi pola dan relasi antar data tanpa bantuan dari pengembang.

Metode unsupervised learning pada umumnya memang tidak ada bantuan dari manusia agar komputer benar-benar mempelajari sebuah data dan relasinya secara mandiri. Dalam kasusnya, dataset tidak berlabel dan mesin secara komputasi akan mengidentifikasi pola dalam data. Unsupervised learning digunakan untuk memudahkan pengembang mengambil keputusan.

Dalam kasus machine learning pinjaman bank tadi, sebuah unsupervised learning dapat mendeteksi anomali atau mengungkap transaksi atau pembayaran yang curang. Unsupervised learning dapat secara otomatis mencari informasi setelah mengelompokkan pola dari semua data peminjam dari sebuah bank dan memunculkannya sebagai sebuah output tanpa harus memasukkan data berlabel secara rinci.

3. Reinforcement Learning

Metode machine learning yang satu ini dijalankan dengan menggunakan dataset bersistem “rewards/punishment” dan menawarkan umpan balik ke algoritma untuk belajar dari pengalamannya secara coba-coba (random). Metode “coba-coba” ini hampir sama dengan sistem pemahaman pola yang dilakukan manusia yaitu belajar dari percobaan.

Hal ini yang lantas membuat metode ini disebut sebagai machine learning dengan tipe penguatan pembelajaran. Algoritma dalam metode ini akan belajar secara terus-menerus dari lingkungan atau kebiasaan interaksi yang berhubungannya dengannya. Dari sana nantinya algoritma akan mendapat “rewards” atau “punishment” sebagai impresi positif dan negatif berdasarkan tindakan percobaannya.

Dalam kasus machine learning pinjaman bank, algoritma reinforcement learning akan mengklasifikasikan pelanggan berisiko tinggi secara default dan akan mengelompokkan pelanggan yang gagal bayar sebagai aspek negatif secara otomatis.

2.3 Unsupervised (Clustering)

Unsupervised learning adalah salah satu tipe algoritma machine learning

yang digunakan untuk menarik kesimpulan dari datasets yang terdiri dari input data labeled response. Metode unsupervised learning yang paling umum adalah Analisa cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data ("Machine learning technique for building predictive models from known input and response data," n.d.). Salah satu algoritma yang digunakan metode unsupervised learning adalah K-Means
Algoritma(Pradono Iswara et al., 2019)

Algoritma K-Means adalah metode partisi yang terkenal untuk clustering. K-Means merupakan salah satu metode data clustering non hierarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih cluster atau kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lainnya.

Unsupervised learning adalah teknik pembelajaran mesin di mana model diajarkan untuk mengidentifikasi pola dalam dataset tanpa adanya label atau panduan sebelumnya. Di sisi lain, clustering adalah salah satu teknik dalam unsupervised learning. Tujuan utama dari clustering yaitu mengelompokkan atau "mengklaster" data yang serupa berdasarkan fitur atau karakteristik tertentu. Sebagai contoh, analyst mungkin perlu mengelompokkan berbagai jenis kesalahan berdasarkan karakteristik mereka untuk memahami jenis masalah yang paling sering muncul dalam aplikasi yang sedang dikerjakan.

Karenanya, ada hubungan langsung antara unsupervised learning dan clustering. Unsupervised learning menyediakan kerangka kerja di mana clustering dan teknik lainnya bisa diterapkan untuk mengekstrak informasi berharga dari data. Clustering merupakan salah satu pendekatan yang dalam unsupervised learning. Artinya, saat melakukan clustering, sebenarnya analyst juga sedang menerapkan unsupervised learning

2.4 K-MEANS

K-Means adalah suatu metode penganalisaan data atau metode data mining yang melakukan proses pemodelan tanpa supervisi (unsupervised) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi. Metode ini berusaha untuk meminimalkan variasi antar data yang ada di dalam suatu cluster dan memaksimalkan variasi dengan data yang ada di cluster lainnya(Rahmawati et al., n.d.).

2.5 Evaluation Model

Evaluasi model sangat penting dalam menilai kinerja algoritma klastering

Berbagai metrik evaluasi dapat digunakan untuk mengukur sejauh mana model dapat mencapai tujuannya. Berikut adalah metrik evaluasi yang relevan untuk membahas kinerja model klastering:

Silhouette Score

Silhouette Score merupakan metrik yang mengukur seberapa baik sebuah objek cocok dengan klaster sendiri dibandingkan dengan klaster lainnya. Rentang nilai Silhouette Score antara -1 hingga 1, di mana nilai positif menunjukkan bahwa objek berada dalam klaster yang sesuai, dan nilai negatif menunjukkan bahwa objek mungkin ditempatkan di klaster yang salah.

Davies-Bouldin Index:

Davies-Bouldin Index mengukur sejauh mana sebuah klaster berada pada jarak yang optimal dari klaster lainnya. Semakin rendah nilai Davies-Bouldin Index, semakin baik pengelompokan klaster.

Inertia (Within-Cluster Sum of Squares):

Inertia mengukur seberapa kompak klaster-klasternya. Ini dihitung sebagai jumlah kuadrat jarak antara setiap titik data dalam klaster dengan pusat klasternya. Tujuan evaluasi adalah untuk meminimalkan nilai inertia.

Adjusted Rand Index (ARI):

ARI mengukur sejauh mana pengelompokan klaster cocok dengan pengelompokan yang sebenarnya. Rentang nilai ARI antara -1 hingga 1, di mana nilai 1 menunjukkan kesesuaian sempurna, dan nilai 0 atau negatif menunjukkan ketidaksesuaian.

Calinski-Harabasz Index:

Calinski-Harabasz Index mengukur sejauh mana klaster-klasternya terpisah dengan baik. Semakin tinggi nilai Calinski-Harabasz Index, semakin baik pemisahan klaster.

Dunn Index:

Dunn Index mengukur rasio minimum antara jarak antara klaster dengan jarak dalam klaster. Tujuan evaluasi adalah untuk memaksimalkan nilai Dunn Index.

2. 16 Penghitungan Jarak Menggunakan K-MEANS

Algoritma K-Means menggunakan jarak sebagai ukuran kesamaan antara data dan pusat cluster (centroid) dengan melalui iterasi-iterasi yang dicari agar sama. Ada beberapa jenis jarak yang dapat digunakan dalam K-Means, antara lain:

1. Jarak Euclidean

Jarak Euclidean adalah jarak terpendek antara dua titik dalam ruang Euclidean. Jarak ini dihitung dengan menggunakan akar kuadrat dari jumlah kuadrat selisih nilai setiap fitur. Rumus jarak Euclidean adalah sebagai berikut:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

d:jarak

x_1=nilai x pertama

x_2=nilai x kedua

y_1=nilai y pertama

y_2=nilai y kedua

2. Jarak Manhattan

Jarak Manhattan adalah jarak antara dua titik yang dihitung dengan menjumlahkan selisih nilai setiap fitur. Rumus jarak Manhattan adalah sebagai berikut:

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i| = |p_1 - q_1| + |p_2 - q_2| + \cdots + |p_n - q_n|$$

d:jarak

p_1=nilai p pertama

p_2=nilai p kedua

q_1=nilai q pertama

q_2=nilai q kedua

3. Jarak Minkowski

Jarak Minkowski adalah generalisasi dari jarak Euclidean dan Manhattan. Rumus jarak Minkowski adalah sebagai berikut:

Jarak Minkowski derajat p (p adalah bilangan bulat) antara dua titik riil, X = (x1, x2,..., xn) dan Y = (y1, y2,..., yn), dapat didefinisikan sebagai berikut.

$$D(\mathbf{X}, \mathbf{Y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

BAB III

HASIL DAN PEMBAHASAN

3.1 Algoritma K-MEANS

3.1.1 Pengumpulan Data

Sumber data didapatkan dari Kaggle dengan judul:‘‘App Users Segmentation: Case Study’’ dengan link: ‘‘https://www.kaggle.com/datasets/bhanupratapbiswas/app-users segmentation-case-study/data’’ dengan jumlah kolom 500 berisi informasi penggunaan aplikasi Fitracker pada kegiatan olahraga yang Dimana ada banyak berbagai kolom yaitu kolom userid, Average Screen Time, Average Spent on App (INR), Left Review, Ratings, New Password Request, Last Visited Minutes, Status.

3.1.2 Preprocessing Data

Dataset ini dikumpulkan karena ingin meningkatkan retensi pengguna aplikasi FiTracker dengan pemilihan variable indenpenden(w1 dan w2) pda dua kolom data pada *Average Screen Time* dan *Average Spent on App (INR).* Dua kolom data tersebut merupakan Waktu layer dibuka rata-rata dan waktu pembelian pada aplikasi rata-rata yang dimana dari dua kolom ini kita bisa melihat retensi pengguna Aplikasi Fitracker.

Memilih data dari dataset untuk digunakan sebagai atribut X

```
In [9]: data = data[["Average Screen Time", "Average Spent on App (INR)"]] #Membuat kolom W1 sebagai X1 dan kolom W2 sebagai x2 dengan m
        data.head(10) #Menampilkan data dari urutan 10 teratas pada kolom W1 dan W2
        #Andre Zuliani_202131031
```

Out[9]:

	Average Screen Time	Average Spent on App (INR)
0	1700	63400
1	0	5400
2	3700	20700
3	3200	44500
4	4500	42700
5	2800	59900
6	4900	88700
7	800	3100
8	2800	74100
9	2800	52400

3.1 gambar tahap Preprocessing

Pada tahap Preprocessing Data seperti gambar diatas, data dicek apakah ada yang hilang tetapi tidak ada yang hilang dan tidak perlu dibersihkan, serta pada tahap ini juga menentukan variabel indenpenden yaitu atribut x yang dipilih dua kolom data pada *Average Screen Time* dan *Average Spent on App (INR)*,Serta memunculkan nilai 10 teratas dengan dua kolom yang dipilih.

3.1.3 Pembentukan Model

```
Out[21]: array([[ 0.30522063,  1.06057331],
                [-1.0780563 , -0.92031375],
                [ 0.88819953, -0.40918923]])
```

3.2 gambar tahap Pembentukan model

Model Yang digunakan K-means dengan nilai kluster 3 dan inisialisasi dari nilai acak 10 dan memiliki nilai kluster pusatnya ([[0.30522063, 1.06057331], [-1.0780563 , -0.92031375], [0.88819953, -0.40918923]]) yang berarti dari hasil K-means ini pengguna aplikasi tidak berbeda jauh dari yang sudah di install atau mengunistall karena dilihat nilai kluster.

3.1.4 Analisis akurasi Model

Pada tahap evaluasi model K-Means dapat digunakan untuk segmentasi pengguna aplikasi FiTracker berdasarkan Average Screen Time dan Average Spent on App (INR). Dengan hasil nilai yang diperoleh 0.8452462307751949(84%). Hasil evaluasi menunjukkan bahwa model KMeans Clustering dengan 3 cluster menghasilkan pemisahan data yang baik. Data dalam cluster cukup mirip satu sama lain dan antar cluster terdapat pemisahan yang baik. Dari hasil segmentasi ini dapat digunakan untuk meningkatkan retensi pengguna dengan:

- Mengirimkan notifikasi dan promosi yang ditargetkan kepada pengguna di each cluster.
- Menawarkan fitur dan layanan yang berbeda kepada pengguna di each cluster.
- Meningkatkan pengalaman pengguna di each cluster.

3.1.5 Pengujian Model

Pengujian model melalui Code:

Pengujian model ini dilakukan dengan metode elbow yang dimana menghitung iterasi dari index ke 1-10 yang berarti 9 kali sehingga mendapatkan nilai: 1 998.0, 2 480.0749213520157, 3 293.4202053418868,

4 195.07964175011125, 5 143.7752947040899, 6 120.71603733823632, 7 102.41997774761009, 8 85.31669111825902, 9 72.57566302130985.

Dari hasil iterasi ditemukan kesimpulan bahwa:

- Nilai k yang optimal kemungkinan adalah 3 atau 4.
- Pada k=3, terdapat 3 cluster dengan SSE yang masih cukup rendah.
- Pada k=4, terdapat 4 cluster dengan SSE yang lebih rendah lagi.
- Pilihan nilai k tergantung pada kebutuhan dan interpretasi terhadap cluster.

Pengujian model Melalui hitungan manual

Pengujian model dilakukan dengan mengambil nilai 10 teratas dari kolom *Average Screen Time* dan *Average Spent on App (INR)* Yang diambil dari dataset dengan skala ratusan $\times 10^2$. Dengan memakai rumus jarak euclidean.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Data Awal	
Average Screen Time	Average Spent on App (INR)
17	634
0	54
37	207
32	445
45	42
28	599
49	887
8	31
28	741
28	524

Membuat tujuan klasifikasi dengan memberikan kelas unistalled dengan nilai 1 dan kelas installed dengan nilai 2.

Tujuan Klasifikasi	
Klas	
unistalled	1
Installed	2

Membuat 2 cluster pusat dan memilih salah satu nilai acak k1 dari kolom *Average screen time* dan salah satu nilai acak k2 dari kolom *Average Spent on App (INR).*

2 cluster pusat		
k=2	Nilai Acak	
k1	17	8
k2	42	31

Menghitung jarak k1 dan jarak k2 dengan rumus euclidean:

Contoh Menghitung k1 pada nomor 1:
$=SQRT((17-17)^2+(634-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 1:
$=SQRT((17-42)^2+(634-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 2:
$=SQRT((0-17)^2+(54-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 2:
$=SQRT((0-42)^2+(54-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 3:
$=SQRT((37-17)^2+(207-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 3:
$=SQRT((37-42)^2+(207-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 4:
$=SQRT((32-17)^2+(445-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 4:
$=SQRT((32-42)^2+(445-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 5:
$=SQRT((45-17)^2+(42-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 5:
$=SQRT((45-42)^2+(42-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 6:
$=SQRT((28-17)^2+(599-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 6:
$=SQRT((28-42)^2+(599-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 7:
$=SQRT((49-17)^2+(887-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 9:
$=SQRT((28-17)^2+(741-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 7:
$=SQRT((49-42)^2+(887-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 9:
$=SQRT((28-42)^2+(741-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 8:
$=SQRT((8-17)^2+(31-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 10:
$=SQRT((28-17)^2+(524-8)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 8:
$=SQRT((8-42)^2+(31-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 10:
$=SQRT((28-42)^2+(524-31)^2)$
(SQRT=Akar(√))

Berikut Hasilnya:

	Jarak masing-masing cluster	
No	Jarak dg k1	jarak dg k2
1	626	603,5180196
2	49,04079934	47,88527958
3	200,0025	176,0710084
4	437,2573613	414,1207553
5	44,04543109	11,40175425
6	591,10236	568,172509
7	879,5822872	856,028621
8	24,69817807	34
9	733,0825329	710,1380148
10	516,1172347	493,1987429

Jarak Terkecil dipilih tergantung kecilnya agar mendekati hasil centroidnya:

Jarak Terkecil
2
2
2
2
2
2
2
1
2
2

Mendapatkan Nilai Iterasi -1 dari perhitungan rata-rata:

2 cluster pusat		
k=2	Iterasi 1	
k1	8	31
k2	29,33	459

k1=8 didapatkan dari hasil jarak terkecil 8 pada kolom *Average Screen Time*.
k1= 31 didapatkan dari hasil jarak terkecil 31 pada kolom *Average Spent on App (INR)*.
k2= 29,33 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [=(17+0+37+32+45+28+49+28+28)/9 pada kolom *Average Screen Time*.
k2= 459 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [=(634+54+207+445+42+599+887+741+524)/9 pada kolom *Average Spent on App (INR)*.

Mencari hasil jarak k1 dan k2 melalui k=2 dengan iterasi-1:

Contoh Menghitung k1 pada nomor 1:
$=SQRT((17-8)^2+(634-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 2:
$=SQRT((0-8)^2+(54-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 1:
$=SQRT((17-29{,}33)^2+(634-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 2:
$=SQRT((0-29{,}33)^2+(54-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 3:
$=SQRT((37-8)^2+(207-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 3:
$=SQRT((37-29,33)^2+(207-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 4:
$=SQRT((32-8)^2+(445-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 4:
$=SQRT((32-29,33)^2+(445-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 5:
$=SQRT((45-8)^2+(42-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 5:
$=SQRT((45-29,33)^2+(42-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 6:
$=SQRT((28-8)^2+(599-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 6:
$=SQRT((28-29,33)^2+(599-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 7:
$=SQRT((49-8)^2+(887-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 7:
$=SQRT((49-29,33)^2+(887-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 8:
$=SQRT((8-8)^2+(31-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 8:
$=SQRT((8-29,33)^2+(31-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 9:
$=SQRT((28-8)^2+(741-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 9:
$=SQRT((28-29,33)^2+(741-459)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 10:
$=SQRT((28-8)^2+(524-31)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 10:
$=SQRT((28-29,33)^2+(524-459)^2)$
(SQRT=Akar(√))

Hasil dari perhitungan dengan rumus euclidean yang diatas:

	Jarak masing-masing cluster	
No	Jarak dg k1	jarak dg k2
1	603,0671604	175,2123932
2	24,35159132	406,2825296
3	178,3732043	252,3387151
4	414,6950687	14,47006275
5	38,60051813	417,5162598
6	568,3520036	139,7841369
7	856,98133	428,2296171
8	0	428,7532889
9	710,2816343	281,7809323
10	493,4055127	64,79149845

Sehingga mendapatkan jarak terkecil dari kedua kolom:

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Karena, nilai jarak terkecil tidak sama antar jarak terkecil awal dan setelah perhitungan tadi, maka diitung iterasi kedua seperti cara sebelumnya:

Jarak Terkecil
2
2
2
2
2
2
2
1
2
2

Jarak sebelum dihitung

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Jarak setelah dihitung

Mendapatkan Nilai Iterasi -2 dari perhitungan rata-rata:

2 cluster pusat		
k=2	Iterasi 2	
k1	23,6	172
k2	30,8	661

k1=23,6 didapatkan dari hasil jarak terkecil(1), contoh : [(0+37+45+8+28)/5] pada kolom *Average Screen Time.*
k1= 172 didapatkan dari hasil jarak terkecil(1) contoh: [(54+207+42+31+524)/5] pada kolom *Average Spent on App (INR).*
k2= 30,8 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [(17+32+28+49+28)/5] pada kolom *Average Screen Time.*
k2= 661 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [(634+445+599+887+741)/5] pada kolom *Average Spent on App (INR).*

Mencari hasil jarak k1 dan k2 melalui k=2 dengan iterasi-2:

Contoh Menghitung k1 pada nomor 1:
$=SQRT((17-23,6)^2+(634-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 2:
$=SQRT((0-23,6)^2+(54-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 1:
$=SQRT((17-30,8)^2+(634-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 2:
$=SQRT((0-30,8)^2+(54-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 3:
$=SQRT((37-23,6)^2+(207-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 4:
$=SQRT((32-23,6)^2+(445-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 3:
$=SQRT((37-30,8)^2+(207-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 4:
$=SQRT((32-30,8)^2+(445-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 5:
$=SQRT((45-23,6)^2+(42-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 5:
$=SQRT((45-30,8)^2+(42-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 6:
$=SQRT((28-23,6)^2+(599-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 6:
$=SQRT((28-30,8)^2+(599-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 7:
$=SQRT((49-23,6)^2+(887-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 7:
$=SQRT((49-30,8)^2+(887-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 8:
$=SQRT((8-23,6)^2+(31-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 8:
$=SQRT((8-30,8)^2+(31-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 9:
$=SQRT((28-23,6)^2+(741-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 9:
$=SQRT((28-30,8)^2+(741-661)^2)$
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 10:
$=SQRT((28-23,6)^2+(524-172)^2)$
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 10:
$=SQRT((28-30,8)^2+(524-661)^2)$
(SQRT=Akar(√))

Hasil dari perhitungan dengan rumus euclidean yang diatas:

	Jarak masing-masing cluster	
No	Jarak dg k1	jarak dg k2
1	462,4470997	30,5004918
2	119,9446539	607,9806576
3	37,85128796	454,2423142
4	273,5290113	216,2033302
5	131,354939	619,3628016
6	427,422648	62,26299061
7	715,8507666	226,5322935
8	141,4627866	630,6123056
9	569,4170001	79,8491077
10	352,4274677	137,2285685

Sehingga mendapatkan jarak terkecil dari kedua kolom:

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Karena, nilai jarak terkecil tidak sama antar jarak terkecil awal dan setelah perhitungan tadi, maka diitung iterasi ketiga seperti cara sebelumnya:

Jarak Terkecil
2
1
1
2
1
2
2
1
2
1

Jarak sebelum dihitung

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Jarak setelah dihitung

Mendapatkan Nilai Iterasi -3 dari perhitungan rata-rata:

2 cluster pusat		
k=2	Iterasi 3	
k1	22,5	83,5
k2	30,33	638

k1=22,5 didapatkan dari hasil jarak terkecil(1), contoh : [(0+37+45+8)/4] pada kolom *Average Screen Time*.

k1= 83,5 didapatkan dari hasil jarak terkecil(1) contoh: [(54+207+42+31)/4] pada kolom *Average Spent on App (INR).*
k2= 30,33 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [(17+32+28+49+28+28)/6] pada kolom *Average Screen Time*.
k2= 638 didapatkan dari jumlah jarak terkecil(2) dibagi banyaknya baris yang jarak terkecilnya(2), contoh: [(634+445+599+887+741+524)/6] pada kolom *Average Spent on App (INR).*
Mencari hasil jarak k1 dan k2 melalui k=2 dengan iterasi-3:

Contoh Menghitung k1 pada nomor 1:
=SQRT((17-22,5)^2+(634-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 2:
=SQRT((0-22,5)^2+(54-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 1:
=SQRT((17-30,33)^2+(634-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 2:
=SQRT((0-30,33)^2+(54-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 3:
=SQRT((37-22,5)^2+(207-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 4:
=SQRT((32-30,33)^2+(445-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 3:
=SQRT((37-30,33)^2+(207-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 5:
=SQRT((45-22,5)^2+(42-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 4:
=SQRT((32-22,5)^2+(445-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 5:
=SQRT((45-30,33)^2+(42-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 6:
=SQRT((28-22,5)^2+(599-83,5)^2) (SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 6:

=SQRT((28-30,33)^2+(599-638)^2) (SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 7:
=SQRT((49-22,5)^2+(887-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 7:
=SQRT((49-30,33)^2+(887-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 8:
=SQRT((8-22,5)^2+(31-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 9:
=SQRT((28-30,33)^2+(741-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 8:
=SQRT((8-30,33)^2+(31-638)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 10:
=SQRT((28-22,5)^2+(524-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k1 pada nomor 9:
=SQRT((28-22,5)^2+(741-83,5)^2)
(SQRT=Akar(√))

Contoh Menghitung k2 pada nomor 10:
=SQRT((28-30,33)^2+(524-638)^2)
(SQRT=Akar(√))

Hasil dari perhitungan dengan rumus euclidean yang diatas:

	Jarak masing-masing cluster	
No	Jarak dg k1	jarak dg k2
1	550,5274743	14,01982723
2	37,10121292	585,1201206
3	124,3483012	431,3848501
4	361,6248056	193,3405171
5	47,20699101	596,5136675
6	515,5293396	39,40248159
7	803,9368756	249,366308
8	54,46558546	607,743824
9	657,5230034	102,6931784
10	440,5343346	114,3571404

Sehingga mendapatkan jarak terkecil dari kedua kolom:

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Karena, nilai jarak terkecil sama antar jarak terkecil awal dan setelah perhitungan tadi, maka selanjutnya mengklasifikasi tujuan kolom melalui jumlah classnya:

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Jarak sebelum dihitung

Jarak Terkecil
2
1
1
2
1
2
2
1
2
2

Jarak setelah dihitung

Tujuan Klasifikasi berdasarkan klasnya:

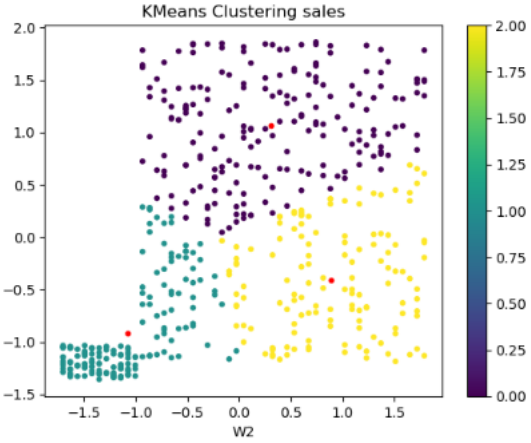
Tujuan Klasifikasi	
Klas	
1	Unistalled
2	Installed

Hasil keterangan dua kolom yang sesuai dengan klasnya masing-masing:

Average Screen Time	Average Spent on App (INR)	klas	Keterangan
17	634	2	Installed
0	54	1	Unistalled
37	207	1	Unistalled
32	445	2	Installed
45	42	1	Unistalled
28	599	2	Installed
49	887	2	Installed
8	31	1	Unistalled
28	741	2	Installed
28	524	2	Installed

Maka, dapat disimpulkan dari 10 data teratas yang sudah diklasifikasi melalui perhitungan jarak euclidean maka aplikasi yang diinstall lebih banyak dibandingkan tidak menginstall maka perlu ditingkatkan lagi retensi aplikasinya agar lebih banyak yang menginstall aplikasi Fitracker selanjutnya.

3.1.6 Visualisasi Model



3.3 gambar tahap Visualisasi Model

pada tahap ini, menunjukkan visualisasi model klasifikasi K-Means dengan 3 cluster. yang dimana terdapat:

- Model K-Means telah membagi data menjadi 3 cluster yang berbeda.
- Data di each cluster ditandai dengan warna yang berbeda.
- Titik-titik hitam menunjukkan pusat cluster.

maka, dari itu terdapat beberapa kesimpulan yaitu:

- Pengguna di cluster dengan Average Screen Time tinggi dan Average Spent on App (INR) tinggi dapat ditargetkan dengan notifikasi tentang fitur baru dan penawaran eksklusif.

- Pengguna di cluster dengan Average Screen Time rendah dan Average Spent on App (INR) rendah dapat ditargetkan dengan notifikasi tentang cara meningkatkan penggunaan aplikasi.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan analisis yang dilakukan dapat disimpulkan bahwa penggunaan algoritma K-Means pada saat melakukan segmentasi pengguna aplikasi FitRacker sangat berguna untuk meningkatkan loyalitas pengguna. Anda dapat menggunakan data rata-rata waktu pemakaian perangkat dan rata-rata pembelanjaan dalam aplikasi (INR) untuk mengidentifikasi pola perilaku pengguna dan mengelompokkannya ke dalam kelompok serupa. Hasil evaluasi model K-Means menunjukkan bahwa membagi data menjadi tiga cluster memberikan pemisahan yang baik dan memungkinkan terciptanya strategi keterlibatan pengguna yang lebih tepat sasaran. Menggunakan pemberitahuan dan promosi yang ditargetkan, menawarkan fitur yang berbeda, dan meningkatkan pengalaman pengguna dapat menjadi strategi yang efektif untuk meningkatkan retensi pengguna pada aplikasi FitRacker Anda. Visualisasi model K-Means juga membantu Anda memahami karakteristik setiap cluster dan mengidentifikasi target pengguna potensial untuk strategi keterlibatan yang lebih efektif. Oleh karena itu, algoritma K-Means memberikan wawasan baru mengenai keragaman perilaku pengguna dalam konteks aplikasi kebugaran dan dapat digunakan sebagai dasar untuk mengembangkan strategi retensi pengguna yang lebih efektif dan berorientasi pada pengguna.

4.2 Saran

Berdasarkan hasil proyek ini, pengambil kebijakan dan pengembang aplikasi direkomendasikan untuk menerapkan strategi pemasaran yang ditargetkan berdasarkan segmentasi pengguna dengan algoritma K-Means. Kami dapat melakukan hal ini dengan mengirimkan kepada Anda pemberitahuan, promosi, dan penawaran yang disesuaikan dengan preferensi dan kebiasaan pengguna di setiap cluster, dan dengan meningkatkan fitur dan layanan kami untuk memenuhi kebutuhan pengguna di setiap cluster. Dengan cara ini, pengguna aplikasi FitRacker dapat merasa lebih terlibat dan terhubung secara pribadi dengan aplikasi, yang dapat meningkatkan loyalitas pengguna secara signifikan.

DAFTAR PUSTAKA

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. In *Electronics (Switzerland)* (Vol. 9, Issue 8, pp. 1–12). MDPI AG. https://doi.org/10.3390/electronics9081295

Harris, S., & De Amorim, R. C. (2022). An Extensive Empirical Comparison of k-means Initialization Algorithms. *IEEE Access*, *10*, 58752–58768. https://doi.org/10.1109/ACCESS.2022.3179803

Kantabutra, S., & Couch, A. L. (2000). *Parallel K-means Clustering Algorithm on NOWs*.*1*,1–6. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6197ce32309824420aac79f975e028e440c2be96

Pradono Iswara, R., Informatika, T., Sains dan Teknologi, F., Syarif Hidayatullah Jakarta, U., & Gotong Royong Jakarta, S. (2019). PENGEMBANGAN ALGORITMA UNSUPERVISED LEARNING TECHNIQUE PADA BIG DATA ANALYSIS DI MEDIA SOSIAL SEBAGAI MEDIA PROMOSI ONLINE BAGI MASYARAKAT. *JURNAL TEKNIK INFORMATIKA*, *12*(1).

Rahmawati, L., Informatika, J., Sihwi, S. W., & Suryani, E. (n.d.). *ANALISA CLUSTERING MENGGUNAKAN METODE K-MEANS DAN HIERARCHICAL CLUSTERING (STUDI KASUS : DOKUMEN SKRIPSI JURUSAN KIMIA, FMIPA, UNIVERSITAS SEBELAS MARET)*.

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Soliman, A., Girdzijauskas, S., Bouguelia, M. R., Pashami, S., & Nowaczyk, S. (2020). Decentralized and Adaptive K-Means Clustering for Non-IID Data Using HyperLogLog Counters. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12084 LNAI*, 343–355. https://doi.org/10.1007/978-3-030-47426-3_27

https://youtu.be/iqjOh3w44zw?si=SMI6wtUxC4gamVFY

LAMPIRAN

Hitungan manual dataset.xlsx
202131031_Andre Zuliani_UAS_Pemsin_B.ipynb
https://github.com/Andre018-zuliani/202131031_Andre-Zuliani_UAS_Pemsin_B.git