BAB II
KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Bagian ini memuat dasar-dasar teoritis yang digunakan serta tinjauan terhadap penelitian terdahulu yang relevan untuk mendukung pelaksanaan penelitian ini.

2.1.1 Preferensi Pengguna

Dengan semakin melekatnya *smartphone* dalam kehidupan sehari-hari, jumlah pengguna perangkat ini telah mengalami lonjakan selama sepuluh tahun terakhir. Melalui *smartphone,* para pengembang aplikasi dan penyedia layanan dapat merekam pola penggunaan aplikasi secara rinci, sehingga memungkinkan identifikasi hubungan antara pengguna, aplikasi dan perangkat itu sendiri. Dengan demikian, pola perilaku pengguna dapat teridentifikasi, dan hal ini memiliki dampak penting bagi berbagai pihak, baik di kalangan akademisi maupun industri [16]. Analisis data penggunaan aplikasi pada *smartphone* dapat mengungkap pola penggunaan aplikasi, dimana hasilnya dapat dimanfaatkan untuk memahami preferensi pengguna sekaligus memprediksi preferensi pengguna di masa mendatang [17].

Preferensi konsumen/pengguna merupakan penilaian produk berdasarkan atribut tertentu yang diukur melalui metode seperti *ranking*, perbandingan berpasangan, atau *rating*. Pemahaman ini membantu perusahaan/pengembang menciptakan produk yang tepat dan menjangkau segmen pasar yang relevan, mendalami kebutuhan pasar, mengoptimalkan produk berdasarkan atribut yang paling diminati, serta mengenali keragaman preferensi pengguna [18].

Beberapa jenis aplikasi *smartphone* didasarkan pada tujuan utama atau fungsi dari sebuah aplikasi, meliputi: sosial media, produktivitas, game/permainan, hiburan, pendidikan, kesehatan, berita, keuangan, dan lain sebagainya [19]. Pada penelitian ini, digunakan 3 jenis aplikasi yang akan dibahas, yaitu: social media, produktivitas dan game/permainan, sesuai dengan atribut dari dataset yang digunakan.

Sejumlah penelitian terdahulu masih memiliki tingkat akurasi yang rendah dalam menganalisis pola penggunaan aplikasi pada *smartphone*, dan cenderung terbatas pada analisis deskriptif, tanpa mengintegrasikan pendekatan prediktif yang mampu mengungkap

insight lebih mendalam mengenai preferensi pengguna [20], [21], [22]. Untuk mengatasi keterbatasan tersebut, proses prediksi preferensi pengguna dapat dimulai dengan menganalisis data riwayat penggunaan aplikasi pada *smartphone* [16], kemudian diolah untuk dilakukan segmentasi terhadap pengguna, sehingga dapat dikelompokkan berdasarkan pola perilaku yang serupa. Dengan pendekatan ini, prediksi preferensi pengguna menjadi lebih akurat [13], [23].

2.1.2 Segmentasi Pengguna

Segmentasi Pengguna (*User Segmentation*) merupakan proses pengelompokan pelanggan ke dalam sejumlah segmen yang memiliki kesamaan dalam pola perilaku, karakteristik, atau atribut tertentu [24]. Dalam konteks penggunaan aplikasi *smartphone*, segmentasi ini dilakukan dengan menganalisis pola interaksi pengguna terhadap aplikasi, seperti frekuensi penggunaan, durasi, dan jenis aplikasi yang digunakan. Pendekatan ini lebih efektif dibandingkan segmentasi tradisional yang hanya berdasarkan demografi, karena mampu mengungkap preferensi aktual pengguna melalui data penggunaan yang nyata [25].

Proses segmentasi pengguna dalam memahami perilaku pelanggan terdiri dari beberapa langkah penting. Pertama-tama, data pelanggan dikumpulkan untuk keperluan analisis lebih lanjut. Kemudian, pengguna diwakili oleh fitur-fitur yang dipilih, baik secara manual maupun melalui metode pemilihan fitur. Setelah itu, pengguna dikelompokkan menjadi segmen-segmen yang lebih homogen berdasarkan kesamaan perilaku. Akhirnya, informasi yang diperoleh dari analisis perilaku digunakan untuk menargetkan pengguna dengan keputusan yang tepat [26]. Preferensi pengguna *smartphone* dapat diprediksi dengan membangun model menggunakan metode *machine learning* [15].

2.1.3 Machine Learning

Machine learning dalam pengertian yang lebih luas, adalah metode berbasis komputer yang memanfaatkan pengalaman sebelumnya untuk meningkatkan performa atau akurasi prediksi. Dalam konteks ini, pengalaman merujuk pada informasi historis yang tersedia sebagai data elektronik, dimana keberhasilan prediksi sangat dipengaruhi oleh kualitas dan ukuran data tersebut. Oleh karena itu, *machine learning* juga mencakup perancangan algoritma prediksi yang efisien dan akurat, dengan mempertimbangkan kompleksitas waktu, ruang, dan ukuran sampel. Keberhasilan algoritma pembelajaran sangat bergantung pada data yang digunakan, hal ini menunjukkan bahwa pembelajaran mesin

memiliki hubungan erat dengan analisis data dan statistik, yang pada gilirannya mempengaruhi efektivitas dan keandalan prediksi yang dihasilkan [27], [28].

Machine Learning dapat dikelompokkan berdasarkan tiga aspek utama: *supervised* (dengan data berlabel untuk prediksi)*, unsupervised* (tanpa label untuk eksplorasi pola), dan *reinforcement* (berbasis interaksi lingkungan). Beberapa algoritma untuk *supervised learning*: *KNN, SVM, Random Forest,* dan *Neural Network* untuk klasifikasi/regresi, *unsupervised* seperti: *K-Means (clustering), PCA, dan Apriori* [28], serta *reinforcement,* seperti: DQN (*Deep Q-Network*), PPO (*Proximal Policy Optimization*), SAC (*Soft Actor-Critic*), dan lain-lain [29].

2.1.4 K-Means Clustering

K-Means Clustering merupakan teknik pengelompokan data (*clustering*) yang membagi data menjadi beberapa kelompok (*cluster*) berdasarkan karakteristik yang mirip. Algoritma ini termasuk dalam *unsupervised learning,* karena bekerja tanpa memerlukan label data [28], [30].

Langkah-langkah algoritma *K-Means Clustering* [31]:

1. Tentukan jumlah *cluster* (k): menentukan jumlah *cluster* yang diinginkan berdasarkan ekplorasi data atau metode, seperti: *Elbow Method, Silhouette Method,* dan lain sebagainya.
   - *Elbow Method* merupakan salah satu metode yang digunakan untuk menentukan jumlah cluster (K) optimal, dengan cara menghitung jarak total titik data ke *centroid* (titik pusat dari sebuah *cluster*) *Sum of Square Error/ Within-Cluster Sum of Squares* untuk berbagai nilai K, lalu memilih nilai K dimana penurunan SSE/WCSS mulai melambat secara signifikan (titik "siku" pada grafik), yang menandakan penambahan *cluster* tidak lagi memberikan peningkatan berarti; metode ini intuitif tetapi terkadang subjektif karena tidak semua dataset menunjukkan titik siku yang jelas. Berikut rumus untuk menghitung *Elbow Method* [32]:

$$SSE = \sum_{K=1}^{K} \sum_{x_i \in C_k} |x - c_k|^2 \quad \text{.....} \quad (1)$$

dimana,

K = Jumlah total cluster yang ditentukan.

C_k = Cluster ke-k (kelompok data ke-k) (berisi semua titik data yang termasuk dalam cluster pertama).

x_i = Sebuah titik data (vektor fitur) dalam cluster Ck.

c_k = Centroid (pusat) dari cluster Ck.

- *Silhouette Score* merupakan metrik untuk mengukur seberapa baik sebuah titik data ditempatkan dalam *cluster*-nya dibandingkan dengan *cluster* lain. Skornya berkisar dari -1 hingga 1, dimana +1 → titik sangat cocok dengan clusternya dan jauh dari cluster lain (ideal), 0 → titik berada di perbatasan antara dua cluster, dan -1 → titik salah dikelompokkan dan lebih cocok dengan *cluster* tetangga. Berikut rumus yang digunakan untuk menghitung *Silhouette Score* [33]:

$$s(i) = \frac{b(i) - a(i)}{max\big(a(i), b(i)\big)} \quad \text{..... (2)}$$

dimana,

$a(i)$ = Jarak Intra-Cluster (Rata-rata jarak antara titik i dengan semua titik lain dalam cluster yang sama).

$b(i)$ = Jarak rata-rata terkecil dari titik i ke semua titik dalam klaster lain yang paling dekat. Ini mengukur seberapa jauh titik tersebut dari klaster tetangga terdekat (separation).

$max\big(a(i), b(i)\big)$ = Digunakan untuk menormalkan nilai, agar hasilnya berada di antara -1 dan +1.

Interpretasi nilai $s(i)$:

$s(i) \approx 1$ Titik sangat cocok di *cluster*-nya sendiri dan jauh dari *cluster* lain.

$s(i) \approx 0$ Titik sangat cocok di *cluster*-nya sendiri dan jauh dari *cluster* lain.

$s(i) < 1$ Titik sangat cocok di *cluster*-nya sendiri dan jauh dari *cluster* lain.

Untuk menghitung $a(i)$, dapat dilihat pada rumus di bawah ini:

$$a(i) = \frac{1}{|C_i|} \sum_{j \in C_i} \text{distance}(i, j) \quad \text{..... (3)}$$

keterangan:

C_i = Jumlah anggota dalam *cluster* Ci.

$\sum_{j \in C_i} \text{distance}(i, j)$ = Penjumlahan terhadap semua anggota j dalam *cluster* Ci.

$\text{distance}(i, j)$ = Jarak antara data i dan j.

2. Inisialisasi *centroid cluster*: Pilih titik awal (*centroid*) secara acak dari data, dimana *centroid* merupakan titik pusat dari sekumpulan data.
3. Tentukan *cluster* untuk setiap titik data, dengan cara: hitung jarak *Euclidean* setiap titik ke *centroid*, dan tetapkan titik ke *cluster* terdekat. Cara kerja *Euclidean Distance* adalah

dengan menghitung selisih tiap elemen dari dua vektor, mengkuadratkannya, menjumlahkan seluruh kuadrat tersebut, lalu mengambil akar kuadrat dari hasilnya. Semakin kecil nilai *Euclidean Distance*, semakin mirip atau dekat dua data tersebut secara spasial [34].

Berikut rumus *Euclidean Method:*

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

dimana,

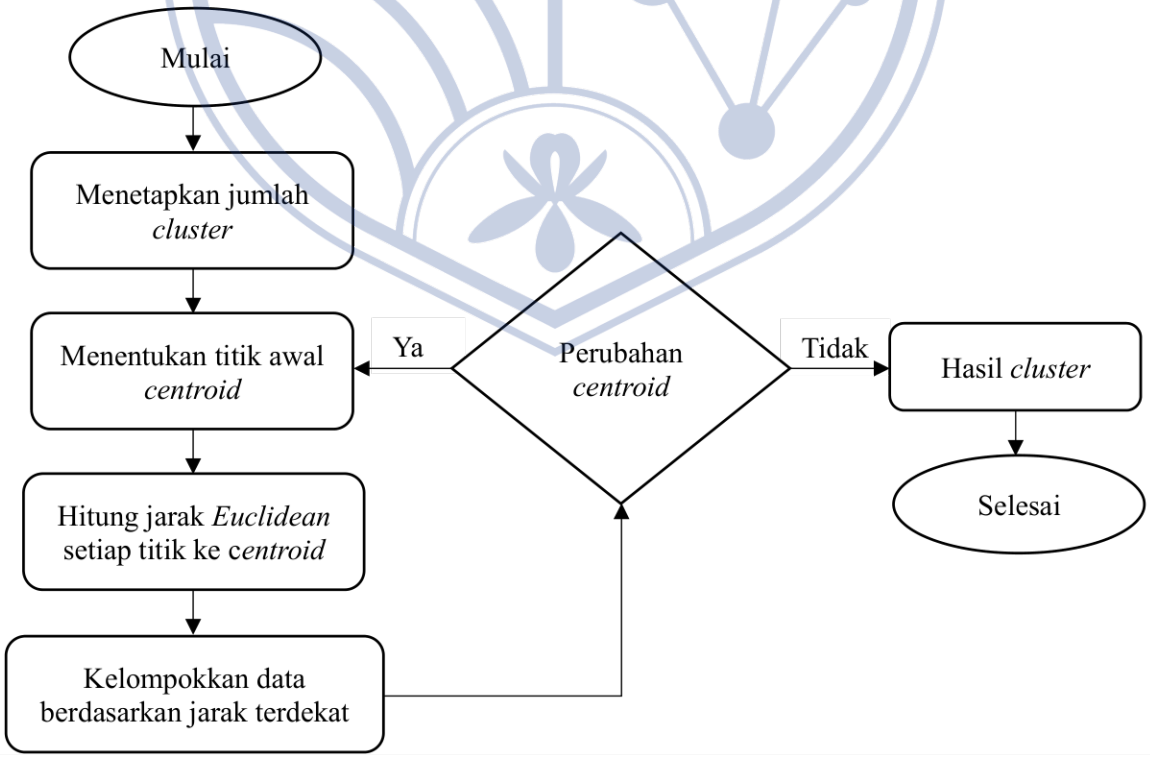
$d(x,y)$ = Jarak *Euclidean* antara dua titik atau vektor x dan y

$\sum_{k=1}^{n}$ = Penjumlahan dari elemen ke-1 sampai ke-n

$(x_k - y_k)^2$ = Selisih kuadrat dari elemen ke-k di x dan y

4. Perbarui *centroid cluster*: hitung *centroid* baru dengan menghitung rata-rata dari semua titik dalam *cluster* tersebut.
5. Ulangi langkah 3 dan 4 hingga konvergen (tidak berubah secara signifikan): proses berlanjut sampai tidak ada perubahan dalam keanggotaan *cluster* atau *centroid* sudah stabil.

Berikut merupakan alur kerja metode *K-Means* [35], [36].



Gambar 2.1 *Flowchart K-Means*

2.1.5 Random Forest

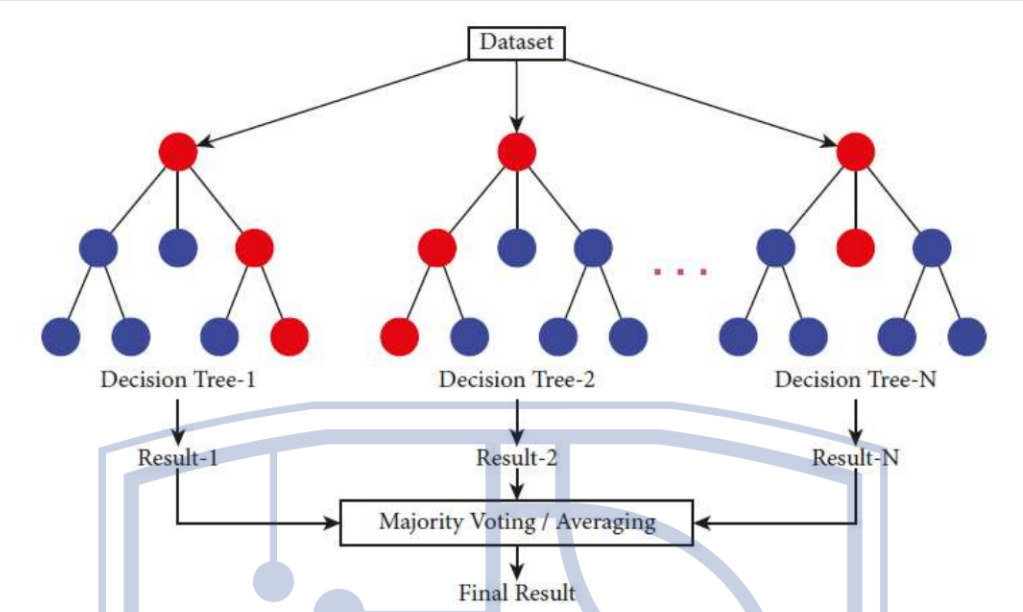
Random Forest merupakan suatu *model machine learning* berbasis *ensemble learning* yang banyak digunakan untuk tugas klasifikasi dan regresi. Model ini beroperasi dengan membangun sejumlah *decision tree* (pohon keputusan) selama fase pelatihan, kemudian mengagregasikan hasil prediksi dari seluruh *tree* tersebut guna memperoleh keluaran akhir yang lebih stabil dan akurat. *Random Forest* memiliki kelebihan khusus dalam mencegah *overfitting* melalui dua mekanisme utama: (1) penggunaan subset data acak (*bagging*) dan (2) seleksi fitur acak pada setiap *node*. Algoritma ini juga memiliki fleksibilitas tinggi dalam mengolah data dengan nilai kosong tanpa tahap pra-pemrosesan yang rumit, serta mampu menilai tingkat pengaruh setiap fitur terhadap model [37].

Dalam konteks penelitian ini, *Random Forest* digunakan bukan untuk memprediksi label preferensi pengguna yang telah tersedia, melainkan untuk memprediksi hasil *cluster* dari proses segmentasi menggunakan algoritma *K-Means* yang telah dikerjakan sebelumnya. Setelah pengguna dikelompokkan ke dalam *cluster* berdasarkan pola penggunaan aplikasi dan durasi layar, hasil *cluster* ini dijadikan sebagai variable target dalam pelatihan model *Random Forest*. Dengan demikian, *Random Forest* dilatih untuk mengenali pola dari fitur demografi (usia, gender, lokasi) dan pola penggunaan aplikasi (*screen time,* jumlah aplikasi, dan lain-lain) yang mengarah pada segmentasi preferensi tertentu seperti *Power User, Gamer*, *Social Media Enthuasiast*, dan lain sebagainya. Pendekatan semi-*supervised* ini penting karena pada banyak kasus-termasuk data perilaku pengguna *smartphone*-tidak tesedia label secara eksplisit mengenai preferensi atau kategori pengguna. Oleh karena itu, segmentasi telebih dahulu diperlukan untuk menyusun label dasar yang kemudian dapat digunakan dalam klasifikasi.

Dalam implementasinya, *Random Forest* efektif digunakan untuk:

- Masalah klasifikasi (biner maupun multikelas).
- Prediksi nilai numerik (regresi).
- Identifikasi data tidak normal (anomali).
- Penentuan fitur paling berpengaruh.

Dengan akurasi yang tinggi, stabilitas model, dan kemudahan penerapan, *Random Forest* banyak dimanfaatkan baik dalam penelitian akademik maupun solusi industri, khususnya untuk data terstruktur. Algoritma ini secara konsisten menunjukkan performa yang sebanding dengan metode *machine learning* lainnya [31], [37].



Gambar 2.2 Ilustrasi *Random Forest* [38]

Random Forest merupakan pengembangan dari *Bagging* yang menambahkan *random* seleksi fitur untuk setiap *tree*. Berikut adalah langkah-langkah algoritma *Random Forest* [31]:

1. *Bagging (Bootstrap Aggregating)*:
   - Ambil B sampel *bootstrap* dari data training.
   - Setiap sampel berukuran sama dengan data asli, tetapi mungkin berisi baris acak ataupun duplikat, tujuan untuk menciptakan variasi data untuk setiap pohon (mencegah *overfitting*).
2. Membangun *Multiple Decision Trees* dengan fitur acak:
   - Untuk setiap *bootstrap sample,* bangun pohon keputusan (*decision tree)* dengan:
     - Pemilihan fitur acak: Di setiap split, hanya subset fitur (√n fitur) yang dipertimbangkan.
     - *Split node* berdasarkan kriteria (*Gini impurity* untuk klasifikasi, dan MSE untuk regresi).
3. *Aggregating Prediction*:
   - Klasifikasi: Vote terbanyak dari semua *tree.*
   - Regresi: Rata-rata prediksi semua *tree.*

dimana:

- Prediksi *Random Forest*:
  - Klasifikasi:

$$\widehat{y_{\text{RF}}} = \text{mode}(\{\widehat{y_b}(x)\}_{b=1}^{B}) \quad \ldots\ldots \quad (5)$$

- Regresi:

$$\widehat{y_{\text{RF}}} = \frac{1}{B}\sum_{b=1}^{B}\widehat{y_b}(x) \quad \ldots\ldots \quad (6)$$

Keterangan:

- $\widehat{y_{\text{RF}}}$: Prediksi akhir oleh *Random Forest.*
- B : Jumlah pohon (*trees*) dalam *Random Forest.*
- $\widehat{y_b}(x)$: Prediksi oleh pohon ke-b untuk input *x*.
- mode{}: Mengambil nilai modus dari prediksi semua pohon (untuk klasifikasi)
- Σ: Penjumlahan untuk regresi.

2.1.6 Metrik Evaluasi Model

Berikut adalah beberapa metrik evaluasi yang digunakan untuk menguji hasil model *Random Forest:*

- *Accuracy*

  Metrik *Accuracy* menghitung persentase prediksi benar dari total prediksi. Metrik ini cocok untuk data seimbang [39]. Berikut rumus untuk *Accuracy*:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots\ldots \quad (7)$$

  Keterangan:

  TP (*True Positive*) = Prediksi benar kelas positif.

  TN (*True Negative*) = Prediksi benar kelas negatif.

  FP (*False Positive*) = Prediksi salah kelas positif (aktual negatif).

  FN (*False Negative*) = Prediksi salah kelas negative (aktual positif).

- *Precision*

  Metrik *Precision* menghitung ketepatan model dalam memprediksi kelas positif (minimalkan *False Positive*). Berikut rumus untuk *Precision:*

$$\text{Precision} = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots\ldots \quad (8)$$

- *Recall*

  Metrik *Recall* menghitung kemampuan model dalam menemukan kasus positif. (minimalkan *False Negative*). Berikut rumus untuk *Recall:*

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{.....} \quad (9)$$

- *F1-Score*

  F1-Score merupakan rata-rata harmonik yang menyeimbangkan *precision* dan *recall.* Rata-rata harmonik digunakan karena lebih sensitif terhadap nilai rendah dibandingkan rata-rata artimetik. Berikut rumus *F1-Score:*

$$\text{F1} = 2\,X\,\frac{Precision\ x\ Recall}{Precision\ +\ Recall} \quad \text{.....} \quad (10)$$

- *Confusion Matrix*

  Confusion Matrix merupakan tabel yang menggambarkan kinerja model klasifikasi dengan membandingkan prediksi dan nilai aktual [40]. Terdiri dari 4 komponen:

  1. *True Positive* (TP): Kasus di mana model benar memprediksi kelas positif.
  2. *False Positive* (FP): Kesalahan tipe I di mana model salah mengidentifikasi negatif sebagai positif.
  3. *False Negative* (FN): Kesalahan tipe II di mana model gagal mendeteksi kasus positif.
  4. *True Negative* (TN): Kasus negatif yang benar diidentifikasi.

- *ROG-AUG*

  Kurva *ROC (Reciever Operating Characteristic)* adalah: grafik yang menunjukkan seberapa bagus model *machine learning* membedakan dua hal (missal: benar/salah). Garis yang terdapat pada kurva menggambarkan:

  - TPR (*Recall*): Seberapa sering model benar memprediksi "positif".
  - FPR: Seberapa sering model salah mengira "negatif" sebagai "positif".

  Semakin melengkung kurva ke atas, semakin baik modelnya.

  AUC adalah angka (0-1) yang mengukur "kelengkungan" kurva ROC:

  - AUC = 1 : Model sempurna.
  - AUC = 0.5 : Model hanya menebak-nebak.

2.1.7 Penelitian Terdahulu

Perkembangan penggunaan *smartphone* yang pesat telah menghasilkan *volume* data yang masif terkait pola penggunaan aplikasi. Data ini memiliki potensi besar untuk dimanfaatkan dalam berbagai bidang, seperti pemasaran digital, rekomendasi produk, dan pengembangan layanan yang lebih personalisasi. Dengan menganalisis data penggunaan aplikasi, preferensi pengguna dapat diprediksi, sehingga perusahaan dapat menyusun strategi yang lebih tepat sasaran, meningkatkan *engagement*, dan mengoptimalkan pengalaman pengguna. Beberapa penelitian terdahulu telah mencoba memanfaatkan data ini dengan berbagai pendekatan.

Tanuwijaya, et al. melakukan penelitian pada tahun 2021 mengenai perilaku pengguna aplikasi *Netflix*. Mereka menggabungkan metode *K-Means* (mengklasterisasi pengguna ke dalam 3 *cluster* berdasarkan konsumsi data) dan *CatBoost* (memprediksi pola pengguna dengan membandingkan ke *e-commerce*), hasil akurasinya mencapai 86.5% [8].

Alruban (2022) meneliti prediksi penggunaan aplikasi *smartphone* menggunakan algoritma LSTM (*Long Short-Term Memory*). Penelitian ini berfokus pada urutan dan waktu penggunaan aplikasi sebagai variabel utama, untuk memprediksi aplikasi berikutnya yang akan digunakan pengguna, dan berhasil mencapai akurasi prediksi sebesar 80% [9]. Namun, penelitian ini tidak memasukkan faktor demografis seperti usia atau gender dalam analisisnya, sehingga kurang mampu mengidentifikasi preferensi pengguna secara lebih spesifik berdasarkan karakteristik individu.

Selvakumar, et al. (2023) memprediksi preferensi pembelajaran *online* menggunakan algoritma *K-Nearest Neighbors* (KNN). Dengan variabel seperti durasi penggunaan, jenis perangkat, dan kondisi kesehatan mental, penelitian ini mencapai akurasi 92,13%. Hasilnya menunjukkan bahwa mayoritas siswa lebih memilih pembelajaran *offline* karena dampak kesehatan [10]. Namun, penelitian ini memiliki keterbatasan dalam konteksnya yang spesifik pada sektor pendidikan, sehingga temuan tidak dapat digeneralisasi untuk preferensi penggunaan aplikasi secara umum. Pentingnya generalisasi dalam analisis perilaku pengguna smartphone kemudian ditekankan dalam penelitian Rao dan Vaishnavi (2024).

Rao dan Vaishnavi (2024) mengembangkan model prediksi kecanduan *smartphone* menggunakan *Random Forest*. Variabel seperti durasi penggunaan, jenis aplikasi, dan tingkat stres digunakan sebagai *predictor* [11]. Penelitian ini berhasil mengidentifikasi faktor-faktor dominan yang memengaruhi kecanduan, tetapi tidak melakukan segmentasi pengguna terlebih dahulu, sehingga hasil prediksi kurang mencerminkan perbedaan preferensi antarkelompok pengguna. Kebutuhan akan pengelompokan pengguna yang lebih

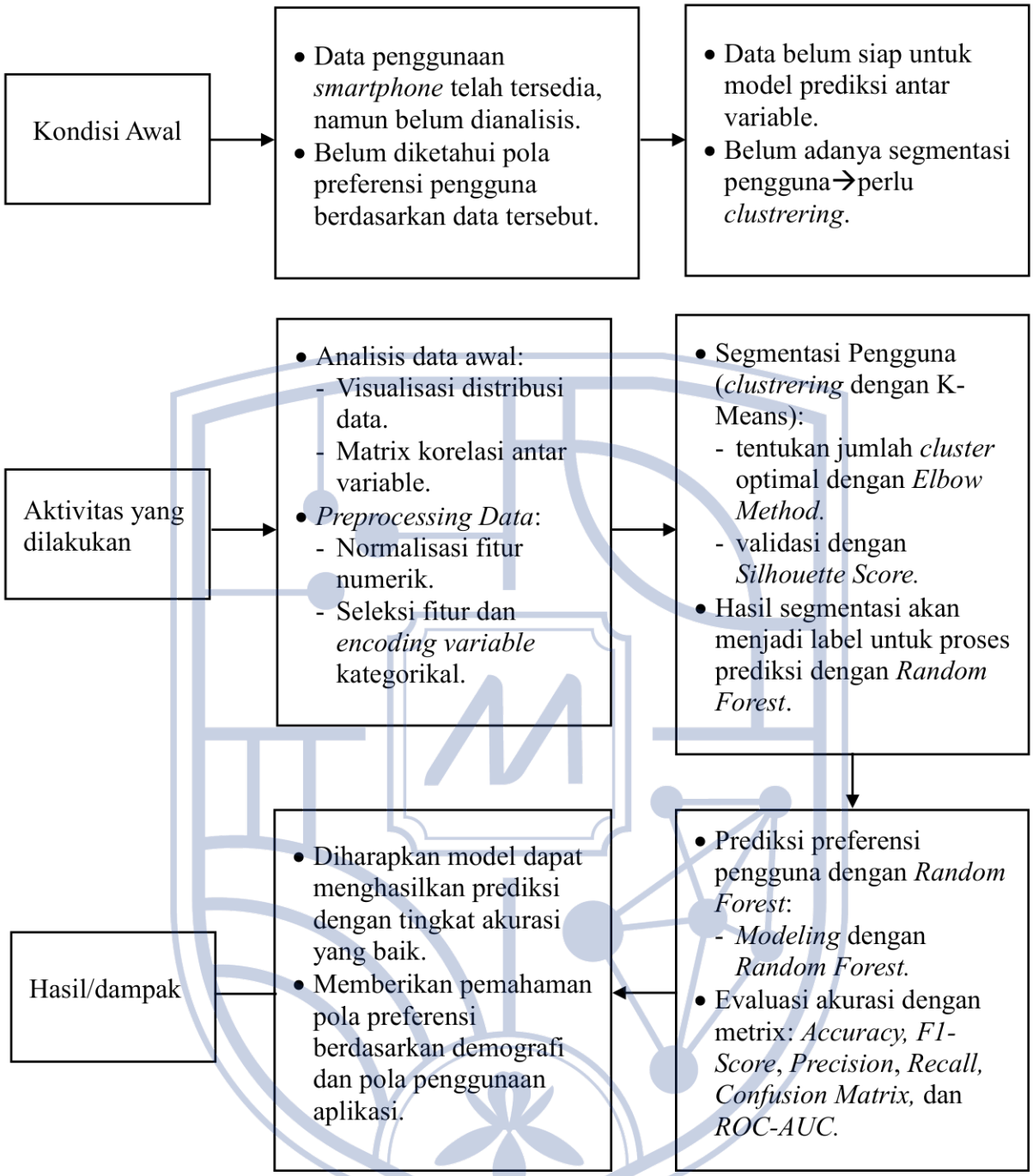
terstruktur ini dibahas oleh Ernawati dan Wahyuni (2024) melalui pendekatan klasifikasi berbasis pohon keputusan.

Ernawati dan Wahyuni (2024) mengklasifikasikan perilaku pengguna *smartphone* menggunakan algoritma C4.5. Dengan variabel seperti jenis perangkat, durasi penggunaan, dan konsumsi data, penelitian ini mencapai akurasi 41,71%. Hasilnya menunjukkan bahwa aplikasi media sosial dan *streaming* paling dominan digunakan, serta pengguna Android lebih banyak daripada iOS [20]. Namun, akurasi yang relatif rendah dan tidak digunakannya pendekatan pengelompokan (*clustering*) menjadi keterbatasan utama penelitian ini. Temuan ini mengindikasikan perlunya pendekatan yang lebih baik, yang mampu menangani keragaman karakteristik pengguna secara lebih menyeluruh.

2.2 Kerangka Pikir Pemecahan Masalah

Kerangka ini menggambarkan bagaimana masalah penelitian dapat diselesaikan melalui solusi yang diusulkan dan dampak yang diberikan dari penyelesaian masalah penelitian. Pada gambar 2.1 Kerangka Konseptual di bawah, penelitian ini diawali dengan tahap kondisi awal, dimana data penggunaan *smartphone* telah tersedia, namun belum dianalisis secara mendalam. Data mentah ini belum menunjukkan pola preferensi pengguna dan memerlukan pra-pemrosesan sebelum nantinya dapat digunakan untuk memuat model prediksi.

Tahap pertama adalah analisis eksplorasi data yang meliputi visualisasi distribusi data dan pembuatan matriks korelasi untuk memahami hubungan antar variabel. Hasil analisis ini menjadi dasar untuk tahap pra-pemrosesan data berikutnya, yaitu: normalisasi fitur numerik dan *encoding variable* kategorikal. Tahap berikutnya adalah proses segmentasi pengguna menggunakan algoritma *K-Means Clustering*. Pada tahap ini, jumlah *cluster optimal* ditentukan dengan menggunakan *Elbow Method* dan kemudian divalidasi dengan *Silhouette Score* untuk memastikan kualitas segmentasi. Hasil dari segmentasi ini adalah pengelompokan pengguna berdasarkan karakteristik penggunaan *smartphone*, yang kemudian dijadikan sebagai label baru untuk proses prediksi. Dengan *dataset* yang telah tersegmentasi ini kemudian dilakukan proses prediksi preferensi pengguna menggunakan algoritma *Random Forest.* Model ini dilatih untuk memprediksi preferensi pengguna berdasarkan pola penggunaan aplikasi dan demografi, kemudian dievaluasi menggunakan metrik: *Accuracy, F1-Score*, *Precision*, *Recall, Confusion Matrix,* dan *ROC-AUC.*, untuk mengetahui performa prediksi. Berikut gambar kerangka konseptual:



Gambar 2.3 Kerangka Konseptual

Pada tahap akhir, yaitu hasil atau dampak, diharapkan bahwa model yang dibangun mampu menghasilkan prediksi preferensi pengguna dengan akurasi yang tinggi yang dapat mengidentifikasi preferensi pengguna secara tepat. Dengan demikian, penelitian ini dapat memberikan kontribusi dalam memahami perilaku pengguna *smartphone* dan membantu pengembangan strategi personalisasi layanan yang lebih efektif berdasarkan segmentasi dan klasifikasi yang telah dilakukan.