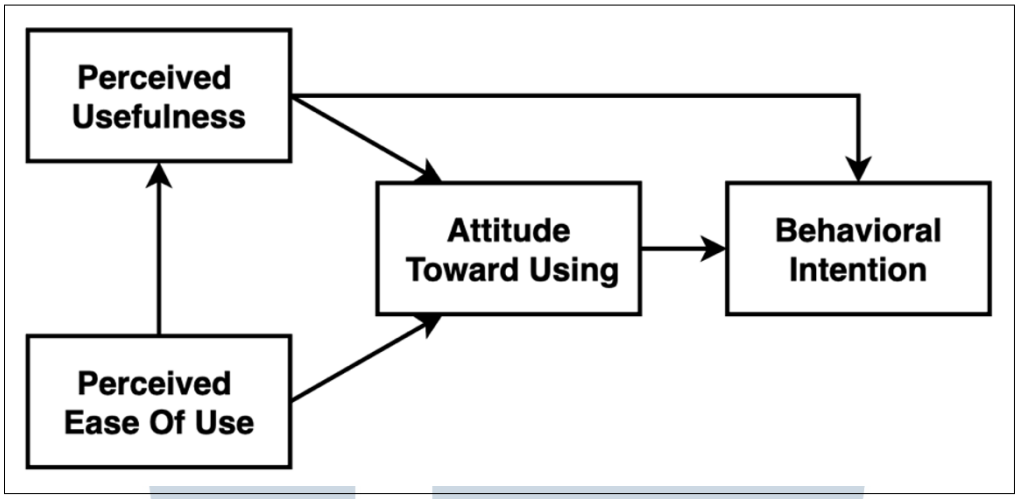
BAB 2
LANDASAN TEORI

Bab ini menjelaskan teori-teori yang menjadi dasar penelitian, meliputi teori mengenai *Technology Acceptance Model* (TAM), *artificial intelligence* (AI), *machine learning*, *logistic regression*, *random forest*, evaluasi model, dan celah riset dengan penelitian sebelumnya.

2.1 Technology Acceptance Model (TAM)

Technology Acceptance Model (TAM) merupakan salah satu model teori terkemuka yang dipakai untuk mendefinisikan proses adopsi dan penerimaan teknologi oleh individu. Pada awalnya, model ini diperkenalkan oleh Fred D. Davis pada 1989, namun karena perkembangan teknologi modern dan konteks yang lebih kompleks, banyak studi terkini yang menerapkan TAM dalam konteks teknologi digital terbaru [6, 12]. TAM mengemukakan bahwa dua konstruk utama, yakni *perceived usefulness* dan *perceived ease of use*. *Perceived usefulness* dapat didefinisikan sebagai seberapa jauh individu percaya penggunaan teknologi akan manaikkan kinerja, sementara *perceived ease of use* dipahami sebagai seberapa besar individu percaya bahwa penggunaan teknologi tidak memerlukan usaha tinggi, yang menjadi determinan penting dalam membentuk sikap terhadap teknologi (*attitude toward using*) dan niat perilaku untuk menggunakan teknologi tersebut (*behavioral intention*). Hubungan antar konstruk TAM secara konseptual disajikan pada Gambar 2.1.



Gambar 2.1. Konstruk *technology acceptance model*

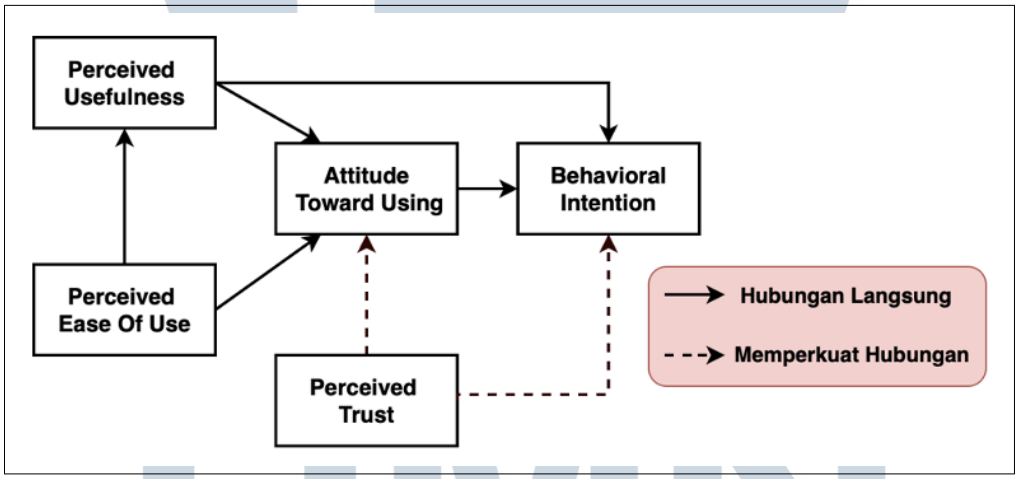
Gambar 2.1 menunjukkan model TAM yang dikembangkan oleh Fred D. Davis (1989). Pada model ini terdapat dua konstruk utama, yaitu *perceived usefulness* dan *perceived ease of use*. *Perceived ease of use* memengaruhi *perceived usefulness*, kemudian kedua konstruk tersebut memengaruhi sikap pengguna terhadap teknologi (*attitude toward using*). Selanjutnya, sikap tersebut memengaruhi *behavioral intention* atau niat menggunakan teknologi. Dengan kata lain, semakin mudah dan bermanfaat suatu teknologi dirasakan oleh pengguna, maka semakin positif sikap dan semakin tinggi niat untuk menggunakannya.

Berbagai penelitian telah mengimplementasikan TAM untuk mengevaluasi penerimaan teknologi dalam konteks pendidikan, termasuk dalam konteks penggunaan *artificial intelligence*. Hasil penelitian [6, 13] menunjukkan, *perceived usefulness* dan *perceived ease of use* memberikan pengaruh yang berarti bagi *behavioral intention* dalam penggunaan teknologi. Penelitian [14] juga menjelaskan, persepsi kegunaan dan kemudahan tetap menjadi faktor penting dalam menentukan tingkat penerimaan teknologi bagi mahasiswa.

Seiring perkembangan teknologi, terutama dalam konteks perkembangan *artificial intelligence*, model TAM juga mengalami perkembangan dengan menambahkan variabel lain untuk memperkuat kemampuan penjelasannya. Salah satu variabel yang sering ditambahkan adalah variabel *perceived trust*. Variabel ini mencerminkan tingkat kepercayaan penggunaan terhadap teknologi yang digunakannya. Beberapa penelitian menunjukkan bahwa *perceived trust* memegang peran penting dalam mempengaruhi niat dan sikap penggunaan teknologi, serta memperkuat keterikatan hubungan antara *perceived usefulness, perceived ease of*

use, dan juga behavioral intention [12, 15].

Berdasarkan temuan-temuan tersebut, TAM merupakan kerangka konseptual yang valid dan relevan untuk diterapkan dalam penelitian ini. Dalam kerangka penelitian ini, konstruk yang digunakan meliputi *perceived usefulness*, *perceived ease of use*, *attitude toward using*, dan *behavioral intention*, dengan menambahkan variabel *perceived trust* sebagai pengembangan model dalam konteks penerimaan *artificial intelligence* oleh mahasiswa. Penambahan variabel *perceived trust* memiliki tujuan memberikan pemahaman yang lebih mendalam mengenai faktor-faktor yang mempengaruhi minat mahasiswa, mengingat aspek kepercayaan menjadi satu dari beberapa faktor utama dalam proses penerimaan teknologi berbasis kecerdasan buatan. Model penelitian dalam penelitian ini dapat dilihat dalam Gambar 2.2.



Gambar 2.2. Konstruk *technology acceptance model extended*
Sumber: [16]

Gambar 2.2 merupakan pengembangan dari TAM dengan menambahkan variabel *perceived trust*. Penambahan variabel ini dilakukan karena pada konteks *artificial intelligence*, aspek kepercayaan menjadi penting mengingat adanya isu terkait akurasi, keamanan, dan privasi data. Dalam penelitian ini digunakan empat variabel prediktor, yaitu *perceived usefulness*, *perceived ease of use*, *attitude toward using*, dan *perceived trust*, untuk memprediksi *behavioral intention.* Dibandingkan TAM orisinal, model ini diharapkan mampu memberikan penjelasan yang lebih komprehensif mengenai faktor-faktor yang memengaruhi minat mahasiswa menggunakan AI dalam pembelajaran.

2.2 Artificial Intelligence dalam Pembelajaran

Artificial intelligence adalah teknologi yang dirancang untuk memungkinkan komputer melakukan kemampuan manusia. Salah satu kemampuan yang umum dilakukan oleh *artificial intelligence* adalah menganalisis data dan membuat keputusan secara otomatis. Dalam lingkungan pendidikan *artificial intelligence* dimanfaatkan guna meningkatkan efektivitas pembelajaran melalui analisa data serta pemberian rekomendasi pembelajaran yang bersifat adaptif sesuai kebutuhan individu [17].

Penggunaan *artificial intelligence* dalam pembelajaran mencakup berbagai aspek, seperti *personalized learning*, evaluasi otomatis, serta pengembangan sistem pembelajaran cerdas yang mampu memberikan *feedback* secara *real-time* [18], [19]. Implementasi ini memungkinkan proses pembelajaran yang lebih efektif, interaktif dan responsif terhadap karakteristik masing-masing siswa [20]. Meskipun demikian, penerapan *artificial intelligence* dalam pembelajaran juga masih mengalami rintangan seperti, infrastruktur yang terbatas, sumber daya manusia yang belum siap menerima, dan isu-isu terkait privasi/keamanan data dan potensi ketergantungan teknologi.

Penelitian ini berfokus pada faktor-faktor penerimaan (*prediction*), bukan pada *learning outcomes* atau manfaat akademik. Dengan menganalisis pandangan mahasiswa menggunakan *Technology Acceptance Model Framework* dan *predictive modeling*, penelitian ini dapat mengidentifikasi profil mahasiswa berdasarkan tingkat minat menggunakan *artificial intelligence*.

2.3 Machine Learning

Sebagai salah satu cabang kecerdasan buatan yang difokuskan pada pengembangan algoritma dan model statistik, *machine learning* memungkinkan sistem komputer belajar dari data tanpa harus diprogram secara spesifik. Sistem berbasis *machine learning* memiliki kemampuan untuk menganalisis pola data, menyusun prediksi, dan memberikan jalan keluar dari suatu keputusan berdasarkan informasi yang diperoleh. *Machine learning* memungkinkan komputer memperoleh pengetahuan dari data, mengenali pola, melakukan prediksi, serta mengambil keputusan dengan intervensi manual dari manusia. Kondisi tersebut menjadikan *machine learning* sebagai salah satu metode untuk pengembangan analisis data *modern* [21].

Berdasarkan cara sistem mempelajari data yang diberikan, *machine learning* umumnya dibagi menjadi dua jenis, yaitu *supervised learning* dan *reinforcement learning*. *Supervised learning* didefinisikan sebagai metode *machine learning* yang menggunakan dataset berlabel, setiap data *input* telah memiliki pasangan *output* yang benar. Model dilatih agar dapat mempelajari hubungan antara variabel *input* dan *output*, agar model dapat digunakan dalam memprediksi data baru. *Supervised learning* pada umumnya digunakan untuk jenis permasalahan *classification* dan *regression*. *Classification* untuk memprediksi kategori atau kelas tertentu dari suatu data, sementara *regression* untuk memprediksi nilai numerik atau kontinu.

Sementara, *reinforcement learning* merupakan metode *machine learning* yang melibatkan interaksi antara agen dan lingkungan. Sistem belajar melalui proses *trial and error* dengan menerima *feedback* berupa *reward* atau *punishment* untuk setiap tindakan yang dilakukan. Metode ini memiliki tujuan memaksimalkan *reward* yang diperoleh selama proses pembelajaran [21]. Dalam penelitian ini *supervised learning* dipilih karena dataset memiliki *labeled outcome* (minat tinggi/rendah) berdasarkan *survey responses*.

Klasifikasi data merupakan salah satu tugas utama yang banyak diterapkan dalam *machine learning*. Dalam *supervised learning*, model klasifikasi dibangun menggunakan dataset berlabel agar dapat mempelajari keterikatan antara variabel prediktor dan variabel target [21]. Beberapa algoritma *machine learning* dapat digunakan untuk melakukan klasifikasi, seperti *logistic regression* dan *random forest*. Melalui proses pelatihan, algoritma tersebut dapat menghasilkan model prediksi yang mampu mengelompokkan data ke dalam kelas tertentu berdasarkan pola yang telah dipelajari.

2.4 Logistic Regression

Pada permasalahan klasifikasi biner, *logistic regression* digunakan untuk memperkirakan peluang suatu observasi termasuk ke dalam kelas tertentu berdasarkan pengaruh satu atau lebih variabel prediktor [22]. *Logistic regression* berfokus pada estimasi probabilitas suatu kejadian, yang dinyatakan dalam nilai 0 hingga 1. Jurnal [22] menunjukkan bahwa *logistic regression* digunakan untuk melakukan klasifikasi biner berbasis probabilitas dengan memanfaatkan fungsi *logistic* untuk mentransformasikan *output* linear menjadi nilai probabilitas. Model *logistic regression* banyak diterapkan dalam *machine learning* karena kesederhanaan dan kemampuannya menghasilkan estimasi probabilistik kelas

tertentu. *Logistic regression* memiliki beberapa bentuk, yaitu biner, *multinominal*, dan *ordinal* [23]. Regresi logistik biner cocok digunakan ketika variabel dependen dikategorikan menjadi dua kelas, yaitu mahasiswa dengan minat tinggi dan minat rendah.

Konsep dasar *logistic regression* adalah mentransformasikan hubungan linear antara variabel independen menjadi probabilitas menggunakan fungsi *sigmoid*, yang memungkinkan model memperkirakan kemungkinan suatu observasi termasuk ke dalam kelas tertentu.

2.4.1 Cara Kerja Logistic Regression

Logistic regression bekerja dengan memanfaatkan hubungan antara variabel independen dan variabel dependen untuk memperkirakan probabilitas suatu data termasuk ke dalam salah satu dari dua kelas. Model ini mengolah seluruh variabel independen secara bersamaan untuk menghasilkan nilai probabilitas yang menggambarkan kemungkinan suatu observasi berada pada kelas positif. Berbeda dengan *linear regression* yang menghasilkan nilai kontinu, *logistic regression* menghasilkan nilai probabilitas sehingga sesuai digunakan pada permasalahan klasifikasi biner [24].

Proses pertama dalam *logistic regression* adalah menerima variabel independen sebagai masukan (*input features*). Pada penelitian ini, variabel tersebut terdiri atas *perceived usefulness*, *perceived ease of use*, *attitude toward using*, dan *perceived trust*. Selanjutnya, model mempelajari hubungan antara seluruh variabel independen dengan variabel dependen melalui proses pelatihan untuk memperoleh koefisien yang paling optimal. Koefisien tersebut menggambarkan besarnya kontribusi masing-masing variabel terhadap peluang terjadinya suatu kejadian.

Setelah hubungan antarvariabel dipelajari, model mengubah hasil perhitungan menjadi nilai probabilitas menggunakan fungsi *sigmoid*. Fungsi ini berperan mengubah nilai hasil perhitungan menjadi rentang probabilitas antara 0 dan 1 sehingga setiap observasi memiliki peluang untuk termasuk ke dalam kelas positif maupun negatif. Semakin besar nilai probabilitas yang dihasilkan, semakin besar pula kemungkinan observasi tersebut termasuk ke dalam kelas positif.

Tahap berikutnya adalah proses klasifikasi menggunakan nilai ambang (*threshold*). Nilai probabilitas yang dihasilkan dibandingkan dengan nilai *threshold* yang telah ditentukan, umumnya sebesar 0.5. Apabila probabilitas lebih besar atau

sama dengan 0.5, maka data diklasifikasikan ke dalam kelas positif, sedangkan apabila lebih kecil dari 0.5 maka data diklasifikasikan ke dalam kelas negatif [25].

Selama proses pelatihan, *logistic regression* menggunakan metode *Maximum Likelihood Estimation* (MLE) untuk menentukan nilai koefisien yang paling sesuai dengan data. Melalui proses optimasi ini, model berusaha menghasilkan probabilitas prediksi yang memiliki tingkat kesesuaian tertinggi terhadap data observasi. Setelah proses pelatihan selesai, model dapat digunakan untuk memprediksi kelas data baru sekaligus memberikan interpretasi mengenai pengaruh masing-masing variabel independen terhadap hasil prediksi.

2.4.2 Model Matematis Logistic Regression

Logistic regression dibangun berdasarkan fungsi *logit* terhadap kombinasi linear variabel independen. Secara matematis, probabilitas suatu kejadian $P(Y = 1)$ dinyatakan dalam Rumus 2.1.

$$P(Y = 1) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

Dengan:

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{2.2}$$

Di mana:

- β_0 = Konstanta
- β_1 = Koefisien Regresi
- X_1 = Variabel independen
- e = Bilangan Eksponensial

Persamaan di atas dikenal sebagai fungsi *sigmoid*, yang mengklasifikasikan nilai linear menjadi probabilitas dalam rentang 0 hingga 1. Selain itu, *logistic regression* juga dapat dinyatakan dalam bentuk *logit* (*log-odds*) yang ditunjukkan dalam Rumus 2.3.

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n \tag{2.3}$$

Melalui transformasi tersebut, parameter β diestimasi menggunakan pendekatan *Maximum Likelihood Estimation* (MLE), yaitu dengan mencari nilai parameter yang memaksimalkan fungsi *log-likelihood* sebagaimana ditunjukkan pada Rumus 2.4.

$$\ell(\beta) = \sum_{i=1}^{n} [y_i \ln(\widehat{p_i}) + (1 - y_i) \ln(1 - \widehat{p_i})] \tag{2.4}$$

Nilai β yang memaksimalkan fungsi *log-likelihood* merupakan parameter optimal yang membuat model menghasilkan prediksi paling sesuai dengan data observasi.

2.4.3 Hyperparameter Logistic Regression

Hyperparameter berfungsi mengendalikan proses pembentukan model sehingga dapat memengaruhi performa, stabilitas, dan kemampuan generalisasi model. Pada algoritma *logistic regression*, beberapa *hyperparameter* yang umum digunakan meliputi *regularization strength* (C), *solver*, *maximum iteration* (*max_iter*), dan *class weight*. Parameter C digunakan untuk mengatur tingkat regularisasi model, nilai yang lebih kecil menghasilkan regularisasi yang lebih kuat untuk mengurangi risiko *overfitting*. Parameter *solver* menentukan metode optimasi yang digunakan untuk mencari nilai koefisien terbaik pada model. Sementara itu, *max_iter* mengatur jumlah maksimum iterasi yang diberikan kepada algoritma untuk mencapai konvergensi, dan *class weight* dipakai untuk mengatasi permasalahan kelas yang tidak seimbang pada data klasifikasi.

Pemilihan *hyperparameter* yang tepat dapat membantu meningkatkan performa model serta menghasilkan prediksi yang lebih seimbang dan dapat diterapkan pada data baru.

2.4.4 Interpretasi Koefisien dan Odds-Rasio

Kemampuannya dalam memberikan interpretasi koefisien melalui nilai *odds-ratio* (OR) merupakan salah satu keunggulan *logistic regression*. Nilai OR diperoleh dari transformasi eksponensial terhadap koefisien regresi. Hubungan antara *log-odds* dan *odds ratio* menjadi karakteristik unik *logistic regression*

dibandingkan model *machine learning* lainnya, karena memungkinkan interpretasi langsung terhadap kontribusi masing-masing variabel dalam model klasifikasi [26].

2.4.5 Kelebihan Logistic Regression

Logistic regression memiliki beberapa keunggulan yang menjadikannya suatu metode klasifikasi yang umum digunakan dalam analisis data dan *machine learning*. Salah satu keunggulannya adalah modelnya yang mudah diinterpretasikan, koefisien model dalam *logistic regression* dapat diinterpretasikan menggunakan konsep *odds ratio*, sehingga memudahkan analisis pengaruh setiap variabel terhadap hasil prediksi. *Logistic regression* juga memiliki kompleksitas perhitungan yang relatif rendah sehingga dapat digunakan pada dataset ukuran besar. Selain itu, *logistic regression* sangat efektif digunakan pada permasalahan klasifikasi yang memiliki dua kategori *output*, seperti klasifikasi positif dan negatif.

Tidak hanya itu, model *logistic regression* juga dapat dengan mudah diimplementasikan menggunakan berbagai perangkat lunak analisis data maupun bahasa pemrograman seperti R dan Python. Karena kelebihan tersebut, *logistic regression* banyak digunakan dalam berbagai penelitian yang melibatkan analisis faktor yang mempengaruhi suatu kejadian.

2.4.6 Penanganan Multicollinearity dalam Logistic Regression

Dalam penelitian ini terdapat potensi *multicollinearity* antara variabel *perceived usefulness* dengan variabel *perceived ease of use*, karena kedua konstruk mengukur aspek-aspek yang terkait evaluasi teknologi. Keberadaan korelasi yang terlalu kuat antar variabel independen dapat menyebabkan estimasi *Maximum Likelihood Estimator* (MLE) pada regresi logistik menjadi kurang efisien, ditandai dengan *varians* yang besar dan koefisien tidak stabil [27]. Untuk mendeteksi adanya *multicollinearity*, dapat dilakukan dengan dua pendekatan utama, yaitu *variance Inflation Factor* (VIF) dan *correlation matrix*. Nilai VIF yang melebihi batas tertentu, umumnya > 5, mengindikasikan adanya *multicollinearity* [28].

Dalam penanganannya, langkah yang dapat dilakukan adalah standarisasi variabel independen untuk meningkatkan stabilitas numerik model. Selanjutnya, dilakukan analisis diagnostik menggunakan VIF untuk mengidentifikasi variabel yang berpotensi menyebabkan *multicollinearity*. Apabila ditemukan variabel dengan nilai VIF yang tinggi maka akan dipertimbangkan untuk dilakukan

penghapusan atau penyesuaian, guna menjaga stabilitas model.

2.4.7 Logistic Regression dalam Konteks Pendidikan dan Data Survei

Logistic regression banyak digunakan dalam konteks pendidikan tinggi. Jurnal [29] menggunakan regresi logistik biner untuk menganalisis faktor pengaruh masa studi dengan pendekatan data survei, serta menginterpretasikan nilai *Exp(B)* sebagai *odds ratio*. Jurnal juga menerapkan *logistic regression* dalam analisis *credit scoring*, yang menegaskan metode ini unggul dalam interpretabilitas koefisien dibandingkan dengan algoritma berbasis *ensemble* seperti *random forest*. Temuan-temuan tersebut menguatkan *logistic regression* merupakan metode yang tepat dilakukan dalam penelitian ini, karena variabel dependen bersifat kategorial, data bersumber dari kuesioner mahasiswa, dan model menghasilkan probabilitas yang dapat dievaluasi menggunakan metrik klasifikasi.

2.4.8 Relevansi Logistic Regression dalam Penelitian

Logistic regression dipilih sebagai model utama dalam penelitian ini karena beberapa pertimbangan. Pertama, metode ini memiliki interpretabilitas yang tinggi, di mana koefisien dan nilai *odds-ratio* yang dihasilkan dapat digunakan untuk menjelaskan pengaruh masing-masing variabel dalam kerangka *Technology Acceptance Model*, yang memungkinkan identifikasi faktor yang paling berpengaruh terhadap minat penggunaan *artificial intelligence*. Kedua, dengan jumlah sampel 535 responden dan jumlah variabel yang relatif terbatas *logistic regression* dinilai memadai dan efektif secara komputasi dalam pemodelan. Ketiga, variabel dependen dalam penelitian ini berbentuk biner, sehingga sesuai dengan karakteristik model klasifikasi biner seperti *logistic regression*.

2.5 Random Forest

Random forest sering kali digunakan dalam menyelesaikan permasalahan klasifikasi dan regresi. Metode ini membangun pohon keputusan (*decision tree*) secara acak sebagai landasan pengambilan keputusan, selanjutnya hasil prediksi dari setiap pohon dikombinasikan untuk menghasilkan keputusan akhir yang stabil dan akurat. *Random forest* menggabungkan dua konsep utama dalam *machine learning*, yaitu *random feature selection* dan *bootstrap sampling*, untuk mendapatkan model yang memiliki generalisasi lebih baik dibandingkan pohon

keputusan tunggal. Dengan menerapkan banyak pohon keputusan, *random forest* dapat mengurangi risiko *overfitting*, sehingga meningkatkan akurasi model [30]. *Random forest* memiliki kemampuan dalam menanggulangi dataset kompleks, memiliki fitur yang besar, dan dapat memberikan estimasi kepentingan variabel (*feature importance*) yang membantu memahami masing-masing variabel terhadap hasil prediksi.

Random forest merupakan algoritma *ensemble* yang membangun banyak model pohon keputusan menggunakan sampel data pelatihan yang berbeda, sehingga masing-masing pohon mempelajari pola beragam dalam data. Tiap-tiap pohon dilatih menggunakan *bootstrap aggregating*, yakni pengambilan sampel data secara acak dari dataset asli dengan pengembalian (*sampling with replacement*). Proses pembentukan *random forest* secara umum dimulai dari mengambil data secara acak dari dataset menggunakan *bootstrap*.

Dari setiap sampel tersebut kemudian dibangun *decision tree*, di mana pada setiap node pohon, dipilih subset fitur secara acak untuk menentukan pemisahan data. Proses tersebut diulang hingga membentuk banyak pohon keputusan. Terakhir, hasil prediksi diperoleh dengan menggabungkan hasil prediksi dari seluruh pohon. Hal tersebut menjadikan *random forest* menghasilkan model yang lebih *robust* dibandingkan dengan algoritma berbasis pohon tunggal.

2.5.1 Decision Tree Random Forest

Decision Tree merupakan algoritma *supervised learning* yang digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi. Algoritma ini bekerja dengan membentuk struktur pohon yang terdiri atas *root node*, *internal node*, cabang (*branch*), dan *leaf node*. Setiap internal node merepresentasikan proses pengambilan keputusan berdasarkan suatu variabel, sedangkan *leaf node* menunjukkan hasil prediksi berupa kelas atau nilai akhir. Struktur tersebut membuat *decision tree* mudah dipahami karena proses pengambilan keputusannya dapat divisualisasikan secara hierarkis.

Proses pembentukan *decision tree* diawali dengan seluruh data ditempatkan pada *root node*. Selanjutnya, algoritma mengevaluasi seluruh variabel independen untuk menentukan pemisahan (*split*) terbaik berdasarkan ukuran *impurity*, seperti *gini impurity* atau *entropy*. Variabel yang menghasilkan penurunan *impurity* terbesar akan dipilih sebagai dasar pemisahan *node*.

Setelah proses pemisahan dilakukan, setiap subset data akan diperlakukan

sebagai *node* baru dan proses yang sama dilakukan secara berulang hingga memenuhi kriteria penghentian (*stopping criteria*), seperti kedalaman maksimum pohon telah tercapai, jumlah sampel pada suatu *node* terlalu sedikit, atau seluruh data pada *node* telah berasal dari kelas yang sama. Hasil akhir berupa *leaf node* menunjukkan kelas prediksi yang digunakan untuk mengklasifikasikan data baru [31].

2.5.2 Gini Impurity dan Entropy Random Forest

Dalam proses pembentukan *decision tree*, diperlukan suatu ukuran untuk menentukan kualitas pemisahan (*split*) pada setiap *node*. Dua ukuran yang paling umum digunakan adalah *gini impurity* dan *entropy*. Kedua ukuran tersebut bertujuan untuk mengukur tingkat ketidakmurnian (*impurity*) suatu node, sehingga algoritma dapat memilih variabel yang menghasilkan pemisahan kelas paling baik. Semakin kecil nilai *impurity*, maka distribusi kelas pada node semakin homogen sehingga kualitas pemisahan dianggap semakin baik.

Gini impurity mengukur peluang suatu sampel dipilih secara acak kemudian diklasifikasikan secara salah apabila proses klasifikasi mengikuti distribusi kelas pada node tersebut. Nilai *gini* dihitung menggunakan persamaan 2.5.

$$\text{Gini} = 1 - \sum_{i=1}^{k} p_i^2 \tag{2.5}$$

dengan:

- p_i = proporsi data pada kelas ke-i.
- k = jumlah kelas.

Nilai *gini* berada pada rentang 0 hingga 0.5 untuk klasifikasi biner. Nilai 0 menunjukkan bahwa seluruh data pada node berasal dari kelas yang sama (*pure node*), sedangkan nilai yang semakin besar menunjukkan tingkat *impurity* yang semakin tinggi [31]. Selain *gini impurity*, ukuran lain yang sering digunakan adalah *entropy*. Konsep *entropy* digunakan untuk mengukur tingkat ketidakpastian (*uncertainty*) pada suatu node.

Persamaan *entropy* dituliskan dalam Rumus 2.6.

$$\text{Entropy} = -\sum_{i=1}^{k} p_i \log_2(p_i) \tag{2.6}$$

dengan:

- p_i = proporsi data pada kelas ke-i.
- k = jumlah kelas.

Semakin kecil nilai *entropy* maka node semakin homogen. Nilai *entropy* sama dengan nol apabila seluruh data pada node berasal dari satu kelas. Baik *gini impurity* maupun *entropy* memiliki tujuan yang sama, yaitu menentukan kualitas pemisahan *node* pada *decision tree*. Perbedaannya terletak pada metode perhitungan. *Entropy* menggunakan fungsi logaritma sehingga membutuhkan komputasi yang lebih kompleks, sedangkan *gini impurity* menggunakan operasi kuadrat yang lebih sederhana sehingga proses perhitungan menjadi lebih cepat [32].

2.5.3 Mekanisme Kerja Random Forest

Random Forest merupakan algoritma *ensemble learning* yang membangun sejumlah *decision tree* secara independen, kemudian menggabungkan hasil prediksi dari seluruh pohon untuk menghasilkan keputusan akhir. Pendekatan ini bertujuan mengurangi *varians* dan meningkatkan kemampuan generalisasi model dibandingkan penggunaan satu *decision tree* saja. Selain itu, *random forest* mampu menangani hubungan nonlinier antarvariabel serta lebih tahan terhadap *overfitting* karena setiap pohon dibangun dari sampel dan subset fitur yang berbeda [33].

Mekanisme kerja *random forest* diawali dengan proses *bootstrap sampling* (bagging), yaitu pengambilan sampel secara acak dengan pengembalian (*sampling with replacement*) dari data pelatihan. Setiap sampel hasil *bootstrap* digunakan untuk membangun satu *decision tree*, sehingga setiap pohon dilatih menggunakan kombinasi data yang berbeda. Proses ini menghasilkan keragaman (*diversity*) antar pohon yang menjadi dasar peningkatan performa model.

Pada setiap node dalam *decision tree*, *random forest* tidak mengevaluasi seluruh variabel independen, melainkan hanya memilih sejumlah fitur secara acak (*random feature selection*). Dari subset fitur tersebut dipilih pemisahan (*split*) terbaik menggunakan kriteria seperti *gini impurity* untuk permasalahan klasifikasi. Strategi pemilihan fitur secara acak bertujuan mengurangi korelasi antar pohon sehingga model menjadi lebih stabil dan memiliki kemampuan generalisasi yang lebih baik.

Setelah seluruh *decision tree* selesai dibangun, proses prediksi dilakukan dengan meneruskan data uji ke setiap pohon. Pada permasalahan klasifikasi, setiap pohon menghasilkan satu prediksi kelas, kemudian *random forest* menentukan hasil akhir menggunakan mekanisme *majority voting*, yaitu kelas yang memperoleh suara terbanyak dari seluruh pohon dipilih sebagai hasil prediksi akhir. Pendekatan ini mampu meningkatkan akurasi prediksi sekaligus mengurangi pengaruh kesalahan yang mungkin terjadi pada masing-masing *decision tree* [34].

2.5.4 Model Prediksi Random Forest dalam Klasifikasi

Dalam permasalahan klasifikasi, *random forest* menentukan kelas suatu data berdasarkan mekanisme suara terbanyak, yakni kelas yang paling sering diprediksi oleh semua pohon keputusan dalam model. Dalam bentuk matematis, prediksi akhir *random forest* dinyatakan dalam Rumus 2.7.

$$\widehat{Y} = \text{mode}(h_1(x), h_2(x), \ldots, h_n(x)) \tag{2.7}$$

Di mana:

- $\widehat{Y}$ = Prediksi akhir kelas
- mode = Kelas paling sering muncul dari tiap-tiap hasil prediksi
- h_n = Prediksi pohon ke-n
- n = Jumlah Pohon

Dengan metode ini, *random forest* mampu meningkatkan stabilitas model serta mengurangi kesalahan prediksi yang dapat terjadi dalam pohon keputusan tunggal.

2.5.5 Hyperparameter Random Forest

Performa model *random forest* dipengaruhi oleh beberapa *hyperparameter* yang digunakan dalam proses pelatihan model. *Hyperparameter* didefinisikan sebagai parameter yang ditentukan sebelum proses *training* dan memiliki peranan dalam mengontrol kompleksitas model serta kualitas prediksi yang dihasilkan. Sejumlah *hyperparameter* yang paling umum digunakan antara lain *n_estimators*,

yaitu jumlah pohon keputusan (*decision tree*) yang dibangun dalam model. Semakin besar jumlah pohon, umumnya performa model akan meningkat, namun dengan konsekuensi waktu komputasi yang lebih besar. Selanjutnya, *max_depth*, yang mengatur kedalaman maksimum setiap pohon keputusan. Pembatasan kedalaman pohon dapat mengurangi risiko *overfitting*.

Parameter *min_samples_split* digunakan untuk mengatur jumlah sampel minimum yang diperlukan untuk melakukan pemisahan pada suatu node. Sedangkan, parameter *min_samples_leaf* menentukan jumlah sampel minimal yang harus terdapat pada node daun (*leaf node*), sehingga membantu meningkatkan generalisasi model. Terakhir, *max_features*, yang menentukan jumlah fitur yang dipertimbangkan dalam setiap proses pemilihan split. Pemilihan fitur ini merupakan salah satu karakteristik utama *random forest* dalam mengurangi korelasi antar pohon.

Untuk mendapatkan performa model yang optimal, dilakukan *tuning hyperparameter* menggunakan metode seperti *grid search* atau *random search*. Proses ini bertujuan untuk menemukan kombinasi parameter terbaik yang mampu menghasilkan akurasi dan generalisasi model yang maksimal.

2.5.6 Feature Importance Calculation Random Forest

Feature Importance merupakan metode untuk mengukur tingkat kontribusi fitur dalam proses pembentukan model klasifikasi, dengan tujuan untuk mengetahui fitur yang paling berpengaruh dalam meningkatkan akurasi model. Dalam *random forest*, *feature importance* umumnya dihitung berdasarkan penurunan nilai *impurity*, khususnya menggunakan *Gini Index*. Setiap kali suatu fitur digunakan untuk melakukan pemisahan (*split*) pada node dalam pohon keputusan, akan terjadi penurunan nilai *impurity*. Semakin besar penurunan *impurity* yang dihasilkan oleh suatu fitur, maka semakin tinggi nilai kepentingannya. Nilai *feature importance* diperoleh dengan menjumlahkan seluruh kontribusi penurunan *Gini* dalam setiap fitur di semua pohon, kemudian dinormalisasi sehingga menghasilkan skor relatif antar fitur. Fitur dengan nilai *importance* tinggi menunjukkan bahwa fitur tersebut memiliki pengaruh signifikan terhadap hasil prediksi model. Selain itu, *cross-validation* dapat diterapkan untuk memastikan bahwa fitur yang dipilih memberikan kontribusi yang konsisten terhadap model [35].

Feature importance dalam *random forest* memberikan perspektif yang berbeda dari koefisien *logistic regression*. *Random forest feature importance*

menunjukkan kontribusi variabel yang relatif dalam *decision trees*, sementara koefisien *logistic regression* menunjukkan *magnitude* dan *direction* pengaruh linier. Dalam konteks penelitian ini *feature importance* akan diterapkan untuk *ranking* variabel TAM berdasarkan *importance*, komparasi dengan *logistic regression* koefisien *magnitude*, dan *understanding* pola *non-linear* yang mungkin *missed* oleh *logistic regression*. *Feature importance* yang tinggi menunjukkan variabel tersebut sering digunakan untuk *splitting* dalam *decision trees* dan mengindikasikan kekuatan prediktif tinggi. Variabel dengan *low importance* menunjukkan kontribusi minim dalam klasifikasi *non-linier*.

2.5.7 Kelebihan Random Forest dan Relevansi dalam Penelitian

Random forest memiliki beberapa keunggulan dibandingkan dengan metode klasifikasi lainnya. [24]. Beberapa kelebihan utama dari algoritma ini seperti, mampu mengurangi *overfitting* karena menggunakan banyak pohon keputusan, mampu menangani dataset dengan fitur yang besar, tidak sensitif terhadap *noise* atau *outlier*, memiliki kemampuan *feature importance*, serta memberikan performa prediksi yang stabil dan akurat. Karena kelebihan tersebut *random forest* banyak digunakan dalam berbagai bidang penelitian.

Dalam penelitian ini, *random forest* dipilih sebagai salah satu algoritma untuk memprediksi minat mahasiswa terhadap penggunaan *artificial intelligence* dalam pembelajaran berdasarkan variabel yang berasal dari kerangka *Technology Acceptance Model* (TAM). *Random forest* dipilih karena kemampuannya dalam menangani dataset yang memiliki banyak variabel, serta kemampuannya menghasilkan model yang presisi dan akurat. Selain itu, algoritma ini juga digunakan sebagai model pembanding terhadap *logistic regression* untuk mengetahui metode yang memiliki performa terbaik dalam memprediksi minat mahasiswa terhadap penggunaan *artificial intelligence*.

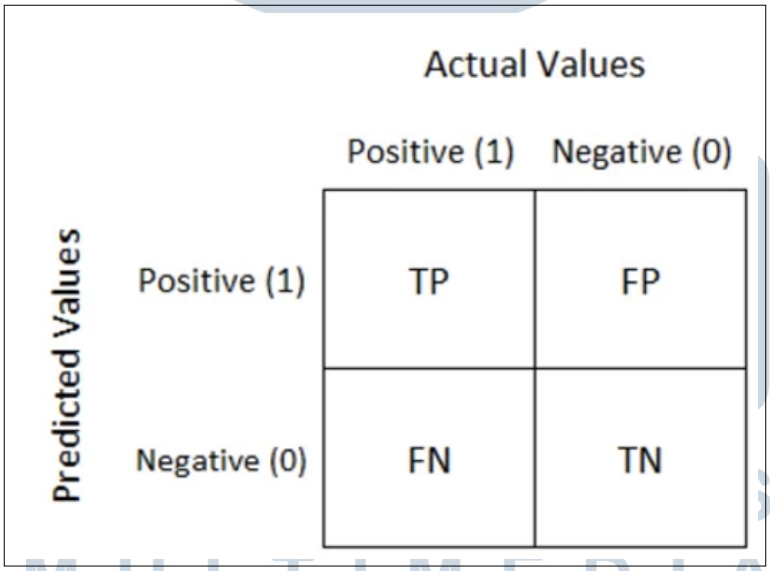
2.6 Evaluasi Model Klasifikasi

Dalam pengembangan model *machine learning*, evaluasi model klasifikasi merupakan salah satu tahapan penting untuk menilai seberapa jauh model mampu melakukan prediksi secara akurat terhadap data yang diolah. Evaluasi model dilakukan dengan memanfaatkan berbagai metrik yang dikalkulasikan berdasarkan hasil prediksi model terhadap data uji. *Confusion matrix*, *accuracy*, *precision*,

recall, dan *F1-score*, merupakan beberapa metrik yang umum digunakan dalam evaluasi model klasifikasi. Metrik tersebut memberikan gambaran besar terkait performa model dalam mengklasifikasikan data secara benar ataupun kesalahan yang dilakukan oleh model [36]. Penggunaan metrik evaluasi dapat membantu peneliti dalam memahami kekuatan dan kelemahan model klasifikasi yang dipakai. Penggunaan beberapa metrik evaluasi juga diperlukan untuk memperoleh gambaran komprehensif terkait performa model [37].

2.6.1 Confusion Matrix

Dalam evaluasi model klasifikasi *confusion matrix* digunakan untuk menggambarkan kemampuan model mengelompokkan data dengan tepat. *Matrix* ini menampilkan distribusi hasil prediksi terhadap kelas sehingga memungkinkan identifikasi jumlah prediksi yang sesuai maupun keliru. Dalam klasifikasi biner, *confusion matrix* terdiri atas empat kategori hasil prediksi, yaitu *TP* (*true positive*), *FP* (*false positive*), *TN* (*true negative*), dan *FN* (*false negative*), yang ditunjukkan dalam Gambar 2.3.



Gambar 2.3. *Confusion matrix*
Sumber: [38]

True Positive menunjukkan total data yang diperkirakan positif dan benar-benar positif, sedangkan *True Negative* menunjukkan total data yang diperkirakan negatif dan hasilnya memang negatif. *False Positive* menunjukkan kondisi ketika

model memprediksi positif tetapi sebenarnya negatif, sementara *False Negative* terjadi saat model memprediksi negatif tetapi sebenarnya positif [39].

2.6.2 Accuracy

Dalam evaluasi model *accuracy* digunakan untuk mengukur tingkat akurasi model pada saat melakukan klasifikasi seluruh data. *Accuracy* dihitung sebagai rasio antara jumlah prediksi yang tepat terhadap total seluruh data yang diuji. Jika nilai *accuracy* semakin tinggi, maka disimpulkan model mampu melakukan prediksi dengan nilai ketepatan yang tinggi. Secara matematis, *accuracy* dapat dihitung menggunakan Rumus 2.8.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

Meski *accuracy* sering diterapkan dalam indikator performa model, metrik ini memiliki keterbatasan pada dataset yang tak seimbang. Dalam kondisi tersebut, nilai *accuracy* dapat terlihat tinggi meskipun model sebenarnya kurang mampu mengidentifikasi kelas minoritas dengan baik [37].

2.6.3 Precision

Precision merupakan metrik yang digunakan dalam mengukur tingkat ketepatan prediksi positif yang dihasilkan model. *Precision* menunjukkan seberapa banyak data yang diprediksi sebagai kelas positif yang benar-benar merupakan data positif (TP). Perhitungan *precision* dapat dinyatakan dalam Rumus 2.9.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.9}$$

Semakin tinggi nilai *precision* menunjukkan bahwa model memiliki tingkat kesalahan *false positive* yang rendah, sehingga prediksi positif yang dihasilkan lebih akurat. Metrik ini memiliki peran penting pada skenario di mana kesalahan prediksi positif dapat menimbulkan dampak signifikan [37].

2.6.4 Recall

Dalam evaluasi model *recall* digunakan untuk mengukur keakuratan model dalam melakukan identifikasi terhadap seluruh data yang masuk dalam kelas positif. *Recall* menunjukkan seberapa banyak data positif yang dengan tepat dikenali oleh model dari total data positif yang ada. Pada formulasi matematis *recall* dapat dituliskan dalam Rumus 2.10.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.10}$$

Nilai *recall* yang semakin tinggi menunjukkan bahwa model dapat mendeteksi sebagian besar data positif dengan baik. Metrik ini penting digunakan dalam skenario di mana kesalahan *false negative* harus diminimalkan [37].

2.6.5 F1-Score

F1-Score adalah metrik evaluasi yang bekerja dengan menggabungkan nilai *precision* dan *recall* dalam satu ukuran. *F1-Score* dikalkulasikan sebagai rata-rata harmonis dari *precision* dan *recall*. Sehingga, *F1-Score* menciptakan keseimbangan antara kedua metrik *precision* dan *recall* tersebut. *F1-Score* secara konteks matematis dapat dituliskan seperti Rumus 2.11.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.11}$$

Penerapan *F1-Score* sangat berguna ketika dataset memiliki distribusi kelas yang tidak seimbang, karena *F1-Score* mampu memberikan gambaran performa model yang lebih seimbang dibandingkan dengan *accuracy* saja. Sehingga, *F1-Score* sering diterapkan sebagai indikator utama dalam evaluasi model klasifikasi pada berbagai penelitian *machine learning* [37].

2.6.6 ROC-AUC

ROC-AUC merupakan metrik evaluasi yang diterapkan dalam mengukur kemampuan model untuk mengidentifikasi kelas positif dan negatif, dengan mempertimbangkan berbagai nilai *threshold* klasifikasi. Kurva ROC merepresentasikan hubungan antara *True Positive Rate* (TPR) dan *False Positive*

Rate (FPR) pada berbagai *threshold*. Nilai AUC berada pada rentang 0 (*equivalent to chance*) hingga 1 (*indicating perfect discrimination*) [40]. Nilai area dan rentang interpretasi AUC ditunjukkan dalam Tabel 2.1.

Tabel 2.1. Nilai area AUC dan interpretasinya

Nilai AUC	**Interpretasi**
$0.50 \leq \text{AUC} < 0.60$	Sangat buruk
$0.60 \leq \text{AUC} < 0.70$	Kurang
$0.70 \leq \text{AUC} < 0.80$	Cukup
$0.80 \leq \text{AUC} < 0.90$	Baik
$0.90 \leq \text{AUC}$	Sangat baik

ROC-AUC memiliki keunggulan karena bersifat *threshold independent*, sehingga dapat mengevaluasi performa model tanpa bergantung pada suatu nilai *cutoff* tertentu. Selain itu ROC-AUC juga sangat baik dalam menangani ketidakseimbangan kelas (*class imbalance*). Berbeda dengan *accuracy* yang cenderung menyesatkan pada data timpang, ROC-AUC mampu mengevaluasi kedua kelas secara setara sehingga menghasilkan hasil yang cenderung akurat [41].

2.7 Research Gap

Bagian ini menjelaskan kajian terhadap penelitian-penelitian terdahulu yang digunakan untuk mengidentifikasi posisi dan kontribusi penelitian ini dibandingkan penelitian sebelumnya. Kajian ini bertujuan untuk mengetahui perkembangan penelitian terkait *Technology Acceptance Model* (TAM), penggunaan *artificial intelligence* dalam pendidikan, serta penerapan metode *machine learning* seperti *logistic regression* dan *random forest* dalam permasalahan klasifikasi. Selain itu, analisis ini juga digunakan untuk menemukan *research gap* yang menjadi dasar perlunya penelitian ini dilakukan. Ringkasan penelitian terdahulu beserta *gap* penelitian dapat dilihat pada Tabel 2.2.

Tabel 2.2. *Research Gap*

Publikasi	**Kontribusi**	***Research Gap***
Tim Fütterer et al. (2023) [4]	Menganalisis penerimaan dan penggunaan *artificial intelligence* dalam konteks pendidikan serta faktor perilaku pengguna terhadap *artificial intelligence*.	Penelitian lebih berfokus pada analisis perilaku dan hubungan antar variabel, belum mengintegrasikan *predictive modeling* berbasis *machine learning*.
Elfirdaus, I. et al. (2024) [6]	Menggunakan *Technology Acceptance Model* (TAM) dalam menganalisis faktor penyebab penerimaan teknologi.	Pendekatan belum membandingkan performa model prediktif.
Salsabilla, W. et al. (2025) [12]	Menambahkan variabel *trust* dalam model *Technology Acceptance Model* untuk menjelaskan penerimaan teknologi.	Fokus pada hubungan struktural antar variabel dan belum mengukur kemampuan klasifikasi model.
Thomas, N. et al. (2024) [42]	Menunjukkan *random forest* memiliki performa prediktif tinggi pada data klasifikasi kompleks.	Tidak mengevaluasi aspek hubungan dengan konstruk perilaku *Technology Acceptance Model*.
Billy & Andarsyah (2024) [43]	Membandingkan *logistic regression* dan *random forest* pada kasus klasifikasi.	Fokus pada performa klasifikasi umum dan tidak membahas interpretabilitas model dalam konteks perilaku pengguna teknologi.
Sitanggang et al. (2026) [44]	Membandingkan *logistic regression* dan *random forest* pada data pendidikan.	Tidak menggunakan konstruk *Technology Acceptance Model*.
Saiz-Alvarez, J. & Huezo-Ponce, L. (2026) [45]	Menggunakan *logistic regression* untuk memprediksi penggunaan *artificial intelligence*.	Tidak melakukan komparasi dengan model *ensemble* seperti *random forest*.
		Lanjut pada halaman berikutnya

Tabel 2.2. *Research Gap* (lanjutan)

Publikasi	**Kontribusi**	***Research Gap***
Xue et al. (2026) [46]	*Meta-analysis Technology Acceptance Model* pada *artificial intelligence* dalam pendidikan, membuktikan hubungan *perceived usefulness*, *perceived ease of use*, dan *behavioral intention* signifikan dalam konteks *artificial intelligence* pendidikan.	Masih berfokus pada hubungan antar variabel *Technology Acceptance Model*, belum mengintegrasikan *predictive modeling* berbasis *machine learning*.

Berdasarkan Tabel 2.2, masih terdapat beberapa *research gap* yang belum dibahas pada penelitian sebelumnya. Pertama, sebagian besar penelitian TAM dalam konteks *artificial intelligence* pendidikan masih berfokus pada analisis hubungan antar variabel dan belum mengintegrasikan pendekatan *predictive modeling* berbasis *machine learning* di dalamnya. Kedua, penelitian yang menggunakan *machine learning* pada umumnya hanya berfokus pada performa prediksi tanpa mengaitkannya dengan kerangka teoritis seperti TAM. Ketiga, masih terbatas penelitian yang membandingkan model *interpretable* seperti *logistic regression* dengan model *ensemble non-linear* seperti *random forest* dalam konteks penerimaan teknologi *artificial intelligence* di pendidikan tinggi.

Maka daripada itu, penelitian ini berupaya mengisi celah tersebut dengan mengkombinasikan konstruk TAM ke dalam pendekatan klasifikasi *machine learning* menggunakan *logistic regression* dan *random forest*.