

## BAB 2

### LANDASAN TEORI

#### 2.1 Intelligent Tutoring System (ITS)

Intelligent Tutoring System (ITS) adalah sistem pembelajaran berbasis komputer yang dapat menyesuaikan materi dan tingkat kesulitan soal berdasarkan pemahaman pelajar dan pada umumnya tidak melibatkan pengajar secara langsung [14]. Sistem pembelajaran ini dibuat dengan tujuan untuk mereplikasikan proses belajar mengajar *one on one* dan sebagai tutor virtual yang memberikan pembelajaran secara personal. Kedua tujuan tersebut dicapai dengan memanfaatkan model *machine learning* untuk menganalisis kinerja pelajar dalam menyelesaikan tugas atau soal.

Komponen-komponen utama dalam sebuah ITS meliputi [14]:

1. *Domain Model*

Berisi materi pembelajaran dan soal yang akan disampaikan kepada pelajar. Pada penelitian ini, materi yang disampaikan kepada pelajar adalah materi mata pelajaran matematika dengan topik aljabar. Materi ini disampaikan dalam bentuk soal-soal yang tingkat kesulitannya meningkat seiring dengan seberapa jauh pelajar dalam menyelesaikan materi.

2. *Student Model*

Memetakan tingkat pemahaman pelajar berdasarkan jawaban mereka terhadap soal-soal sebelumnya. Model ini menyimpan informasi seperti kemampuan siswa, riwayat percobaan menjawab, soal-soal yang pernah dikerjakan, kebenaran jawaban, dan probabilitas keberhasilan menjawab soal berikutnya.

3. *Tutoring Model*

Menentukan strategi pemberian soal yang sesuai dengan kebutuhan pelajar. Pada penelitian ini, ITS akan menentukan seberapa cepat pelajar dapat menyelesaikan materi dengan memberikan urutan soal yang lebih sulit atau lebih mudah berdasarkan hasil analisis performa pelajar tersebut pada komponen *Student Model*.

4. *User Interface*

Tampilan antarmuka yang memungkinkan interaksi antara siswa dan sistem.

*User Interface* harus dirancang agar ramah pengguna, intuitif, dan mampu menampilkan pertanyaan serta feedback secara jelas.

## **2.2 Edukasi Mata Pelajaran Matematika pada Kelas 8**

Kelas delapan merupakan tahap penting dalam perjalanan pendidikan siswa, terutama dalam bidang matematika, di mana keterampilan dasar diperkuat untuk mempersiapkan mereka menghadapi tantangan di jenjang sekolah menengah atas dan seterusnya. Penelitian menunjukkan beberapa alasan utama mengapa tingkat kelas ini sangat krusial untuk pendidikan matematika. Kelas delapan berfungsi sebagai masa transisi penting antara sekolah menengah pertama dan menengah atas, memperkenalkan siswa pada konsep-konsep matematika yang lebih maju seperti persamaan linear, penalaran aljabar, dan fungsi. Penguasaan topik-topik ini sangat penting untuk keberhasilan dalam mata pelajaran STEM berikutnya dan untuk kesiapan masuk perguruan tinggi secara keseluruhan [4].

Studi lain juga menunjukkan bahwa motivasi belajar yang kuat di kelas 8 juga terbukti berpengaruh signifikan terhadap hasil belajar siswa, sehingga guru perlu menggunakan media dan metode pembelajaran yang kreatif untuk meningkatkan motivasi tersebut, menandakan bahwa kelas 8 adalah masa penting dalam membangun sikap positif terhadap belajar secara umum [15]. Kurikulum matematika di kelas delapan menghadirkan tantangan baru yang mengharuskan siswa mengembangkan keterampilan analitis dan penalaran yang melampaui aritmetika dasar. Salah satu dampak negatif dari kurangnya pemahaman terhadap konsep-konsep matematika yang lebih kompleks ini tercerminkan pada angka putus kuliah pada jurusan perguruan tinggi yang memiliki dasar pada konsep-konsep matematika. Berdasarkan data dari Pangkalan Data Pendidikan Tinggi (PDDikti) pada tahun 2020, ditemukan bahwa tiga jurusan dengan angka putus kuliah tertinggi adalah jurusan ekonomi pada angka 23,5%, jurusan teknik pada angka 22,6%, dan jurusan pendidikan pada angka 20% [5]. Selain dalam studi tingkat lanjut, kesulitan dalam memahami konsep-konsep matematika pada tahap ini dapat menjadi hambatan dalam menerapkan konsep-konsep matematika tersebut di kehidupan nyata [4]. Selain itu, beberapa faktor eksternal seperti disiplin dan bakat serta internal seperti sarana dan prasarana sekolah, mempengaruhi minat belajar siswa, keterlibatan di kelas, dan pada akhirnya, pencapaian mereka dalam matematika [16].

### 2.3 Self-Attentive Model for Knowledge Tracing (SAKT)

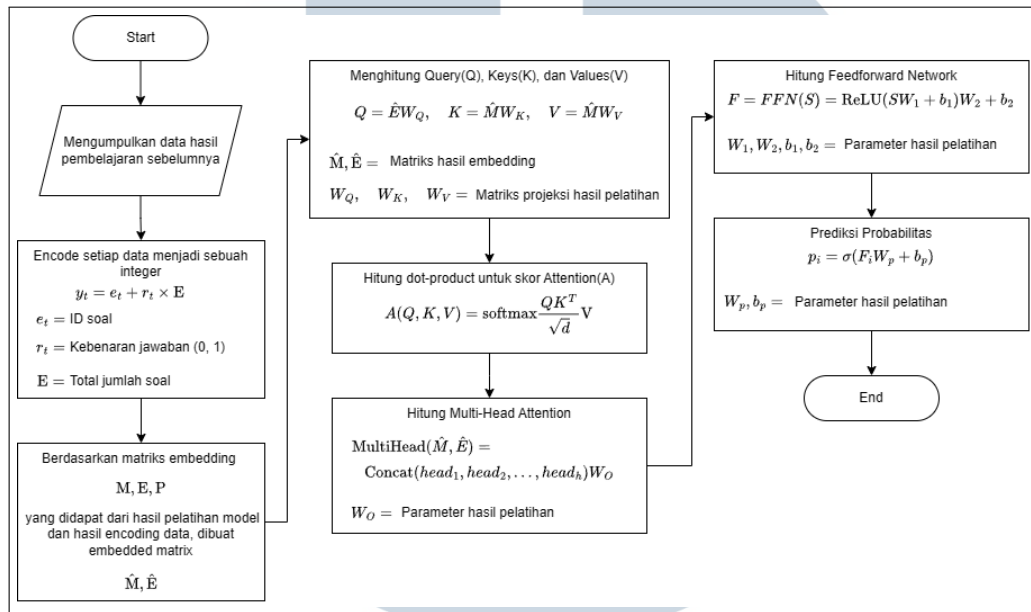
*Self-Attentive Model for Knowledge Tracing* (SAKT) adalah sebuah model *machine learning* yang menggunakan mekanisme *self attention* untuk melakukan *knowledge tracing* dengan tujuan untuk membuat representasi dari performa pelajar. *Self-attention* adalah mekanisme yang digunakan dalam sebuah model *machine learning* untuk menangkap dependensi dan relasi dalam sebuah urutan data [17]. Mekanisme *self-attention* pertama kali diperkenalkan sebagai komponen inti dari arsitektur transformer dalam penelitian “Attention Is All You Need” oleh Vaswani et al (2017) dengan tujuan untuk menggantikan *Recurrent Neural Network* (RNN) serta *Convolutional Neural Network* (CNN) dalam memproses data sekuensial [18]. Penelitian tersebut juga membuktikan bahwa penggunaan mekanisme *self-attention* lebih efektif dalam memproses data sekuensial daripada RNN dan CNN [18].

Model SAKT menggunakan mekanisme *self-attention* untuk melakukan tugas *knowledge tracing*. *Knowledge Tracing* adalah proses pemodelan dari pemahaman setiap pelajar pada suatu *knowledge concept*/konsep materi seiring mereka berinteraksi dengan berbagai aktivitas pembelajaran. Hasil dari *knowledge tracing* seringkali berbentuk prediksi probabilitas performa pelajar dalam proses pembelajaran selanjutnya. Prediksi probabilitas ini yang digunakan oleh aplikasi untuk menyesuaikan konten pembelajaran yang diberikan kepada pelajar berdasarkan performa mereka pada proses pembelajaran sebelumnya.

Berdasarkan penelitian yang dilakukan oleh Pandey dan Karypis (2019) yang pertama kali mengemukakan model SAKT sebagai metode untuk mengevaluasi performa pelajar, model SAKT lebih fleksibel karena mampu mengenali keterkaitan antar soal secara lebih efisien dan tidak dipengaruhi oleh urutan kronologis data [19]. Hal ini memungkinkan sistem untuk tidak hanya melihat soal yang dijawab sebelumnya, tetapi juga mempertimbangkan hubungan antara soal-soal yang ada pada materi yang sama.

Berdasarkan penelitian yang sama, model SAKT memiliki performa yang lebih baik dalam menjalankan tugas *Knowledge Tracing* daripada model lainnya seperti *Deep Knowledge Tracing* (DKT) dan *Dynamic Key-Value Memory Network* (DKVMN) berdasarkan evaluasi model menggunakan skor *Area Under the ROC Curve* (AUC). Skor AUC digunakan dalam mengevaluasi model SAKT karena metrik evaluasi tersebut dapat menunjukkan seberapa akurat prediksi probabilitas dari model. Semakin tinggi skor AUC, hasil prediksi dari model semakin dekat dengan kejadian di dunia nyata. Apabila skor AUC yang didapat 0.85, artinya

ketika model memprediksi probabilitas dan mendapatkan 0.75, probabilitas tersebut merepresentasikan persentase pelajar yang dapat menjawab dengan benar (75%) di dunia nyata dengan kemungkinan prediksi tersebut tergolong benar/sesuai dengan data nyata sekitar 85%.



Gambar 2.1. Flowchart proses prediksi model SAKT

SAKT menggunakan mekanisme *self-attention* untuk menentukan hubungan antara soal yang telah dijawab oleh pelajar dengan soal berikutnya. Langkah-langkah utama dalam arsitektur SAKT adalah sebagai berikut:

### 1. Pengumpulan data hasil pembelajaran sebelumnya

Model menerima riwayat percobaan menjawab pelajar sebelumnya dalam bentuk sekuens yang ditampilkan pada Rumus 2.1.

$$X = (x_1, x_2, x_3, \dots, x_t) \text{ dengan } x_t = (e_t, r_t) \quad (2.1)$$

Di mana  $e_t$  adalah ID soal, dan  $r_t$  adalah hasil jawaban (1 jika benar, 0 jika salah).

### 2. Encoding setiap data menjadi sebuah integer

Setiap data pada sekuens  $X$  di-encode menjadi sebuah sekuens  $Y = (y_1, y_2, y_3, \dots, y_t)$  yang berisi integer/angka dengan menggunakan Rumus 2.2.

$$y_t = e_t + r_t \times E \quad (2.2)$$

Di mana  $y_t$  adalah hasil *encoding* dan  $E$  adalah total jumlah soal yang ada pada materi.

### 3. *Embedding* hasil *encoding* ke dalam matriks

Setelah dilakukan *encoding* dan didapat sekuens  $Y$ , panjang total sekuens  $Y(t)$  dibandingkan dengan maksimum panjang sekuens yang dapat diterima oleh model ( $n$ ), dibuat sekuens baru ( $S$ ). Apabila  $t$  lebih kecil dari  $n$ , maka pada sekuens  $S$  ditambahkan *padding* menggunakan angka 0. Sebagai contoh asumsikan  $t = 2$  dan  $n = 4$ :

$$\text{Jika } Y = (7, 4), \text{ maka } S = (0, 0, 7, 4)$$

Di sisi lain, apabila  $t$  lebih besar dari  $n$ , maka dibuat subsekuens dengan jumlah  $t/n$  yang panjang sekuensnya tidak lebih besar dari  $n$ . Sebagai contoh asumsikan  $t = 4$  dan  $n = 2$ :

$$\text{Jika } Y = (7, 4, 11, 9), \text{ maka } S_1 = (7, 4) \text{ dan } S_2 = (11, 9)$$

Untuk melakukan proses *embedding* sekuens  $S$  diperlukan tiga matriks, yaitu *interaction embedding matrix* ( $M \in \mathbb{R}^{2E \times d}$ ), *excerise lookup matrix* ( $E \in \mathbb{R}^{E \times d}$ ), dan *position embedding matrix* ( $P \in \mathbb{R}^{n \times d}$ ):

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & \dots & M_{1,d} \\ M_{2,1} & M_{2,2} & M_{2,3} & \dots & M_{2,d} \\ M_{3,1} & M_{3,2} & M_{3,3} & \dots & M_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ M_{2E,1} & M_{2E,2} & M_{2E,3} & \dots & M_{2E,d} \end{bmatrix}$$

$$E = \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} & \dots & E_{1,d} \\ E_{2,1} & E_{2,2} & E_{2,3} & \dots & E_{2,d} \\ E_{3,1} & E_{3,2} & E_{3,3} & \dots & E_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ E_{E,1} & E_{E,2} & E_{E,3} & \dots & E_{E,d} \end{bmatrix}$$

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \dots & P_{1,d} \\ P_{2,1} & P_{2,2} & P_{2,3} & \dots & P_{2,d} \\ P_{3,1} & P_{3,2} & P_{3,3} & \dots & P_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ P_{n,1} & P_{n,2} & P_{n,3} & \dots & P_{n,d} \end{bmatrix}$$

Dimana  $E$  adalah jumlah total soal,  $d$  adalah jumlah dimensi *embedding*, dan  $n$  adalah maksimum panjang sekuens. Jumlah dimensi *embedding* dan maksimum panjang sekuens adalah dua angka yang diatur pada *hyperparameter* model saat memulai pelatihan.

Ketiga matriks tersebut diperoleh dari hasil pelatihan model dan digunakan dalam melakukan *embedding* sekuens hasil *encoding* ke dalam dua matriks, yaitu *embedded interaction input matrix* ( $\hat{M}$ ) dan *embedded exercise matrix* ( $\hat{E}$ ):

$$\hat{M} = \begin{bmatrix} M_{s_1} + P_1 \\ M_{s_2} + P_2 \\ M_{s_3} + P_3 \\ \dots \\ M_{s_n} + P_n \end{bmatrix}, \hat{E} = \begin{bmatrix} E_{s_{e1}} \\ E_{s_{e2}} \\ E_{s_{e3}} \\ \dots \\ E_{s_{en}} \end{bmatrix}$$

Proses *embedding* dilakukan dengan mengambil angka dari setiap elemen pada sekuens  $S$  dan mencari nomor baris yang sesuai pada matriks  $M$  dan  $E$ . Setelah itu, elemen-elemen pada baris yang ditemukan pada matriks  $M$  ditambahkan dengan elemen-elemen dari baris data pada matriks  $P$  berdasarkan posisi elemen pada sekuens dan disimpan pada matriks  $\hat{M}$ . Baris yang ditemukan pada matriks  $E$  disimpan pada matriks  $\hat{E}$ . Contohnya elemen sekuens  $S$  ke-3 ( $s_3$ ) adalah 7 dan soal ( $s_{e3}$ ) pada elemen itu adalah soal ke-2, maka:

$$\text{Jika } s_3 = 7 \text{ dan } s_{e3} = 2, \text{ maka } M_{s_3} = [M_7 + P_3], E_{s_{e3}} = E_2$$

$$M_{s_3} = \begin{bmatrix} M_{7,1} + P_{3,1} & M_{7,2} + P_{3,2} & M_{7,3} + P_{3,2} & \dots & M_{7,d} + P_{3,d} \end{bmatrix}$$

$$E_{s_{e3}} = \begin{bmatrix} E_{2,1} & E_{2,2} & E_{2,3} & \dots & E_{2,d} \end{bmatrix}$$

#### 4. Pembentukan matriks *Query*, *Key*, dan *Value*

Matriks  $\hat{M}$  dan  $\hat{E}$  dari tahap sebelumnya dilakukan perkalian dengan matriks  $W^Q, W^K, W^V$  untuk memperoleh matriks *Query*, *Key*, dan *Value* seperti pada

Rumus 2.3.

$$Q = \hat{E}W^Q, \quad K = \hat{M}W^K, \quad V = \hat{M}W^V \quad (2.3)$$

dengan  $W^Q \in \mathbb{R}^{d \times d}$ ,  $W^K \in \mathbb{R}^{d \times d}$ ,  $W^V \in \mathbb{R}^{d \times d}$  sebagai matriks proyeksi hasil pelatihan.

$$W^Q = \begin{bmatrix} W_{1,1}^Q & W_{1,2}^Q & \dots & W_{1,d}^Q \\ W_{2,1}^Q & W_{2,2}^Q & \dots & W_{2,d}^Q \\ \dots & \dots & \dots & \dots \\ W_{d,1}^Q & W_{d,2}^Q & \dots & W_{d,d}^Q \end{bmatrix}, \quad W^K = \begin{bmatrix} W_{1,1}^K & W_{1,2}^K & \dots & W_{1,d}^K \\ W_{2,1}^K & W_{2,2}^K & \dots & W_{2,d}^K \\ \dots & \dots & \dots & \dots \\ W_{d,1}^K & W_{d,2}^K & \dots & W_{d,d}^K \end{bmatrix},$$

$$W^V = \begin{bmatrix} W_{1,1}^V & W_{1,2}^V & \dots & W_{1,d}^V \\ W_{2,1}^V & W_{2,2}^V & \dots & W_{2,d}^V \\ \dots & \dots & \dots & \dots \\ W_{d,1}^V & W_{d,2}^V & \dots & W_{d,d}^V \end{bmatrix}$$

## 5. Perhitungan Attention Score

Menggunakan mekanisme utama dalam *self-attention* adalah menghitung *scaled dot-product attention* dari *Query*, *Key*, dan *Value* yang didapat pada tahap sebelumnya dengan Rumus 2.4.

$$Attention(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V \quad (2.4)$$

Di mana  $d$  adalah dimensi embedding.

## 6. Multi-Head Attention

Salah satu *hyperparameter* pada model SAKT yang dapat diatur sebelum dilakukan pelatihan adalah *num\_heads* ( $h$ ) untuk mengatur jumlah *heads* pada mekanisme *attention*. Jumlah *heads* ( $h$ ) menentukan jumlah proses penghitungan *attention* yang berjalan secara paralel. Jumlah dimensi *embedding* dari setiap *head* ditentukan dengan membagi jumlah dimensi *embedding* ( $d$ ) yang ditetapkan pada *hyperparameter* dengan jumlah *head* ( $h$ ):

$$\text{Jumlah dimensi embedding pada setiap head} = d_h$$

$$d_h = d/h$$

$$\text{Asumsi : } h = 4, d = 64$$

$$d_h = 64/4$$

$$d_h = 16$$

Hasil dari setiap *head* akan di-*concat* atau digabungkan menjadi satu matriks dengan menggunakan Rumus 2.5.

$$\text{MultiHead}(\hat{M}, \hat{E}) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W_O \quad (2.5)$$

dengan  $W_O$  sebagai matriks yang dihasilkan dari proses pelatihan model dan matriks hasil ( $\text{MultiHead}(\hat{M}, \hat{E})$ ) yang didefinisikan ulang sebagai matriks  $S$ .

#### 7. Feedforward Neural Network (FFN)

Hasil dari *concat Multi-Head Attention* (matriks  $S$ ) dihitung ke dalam sebuah *Feed-Forward Network* (FFN). Untuk model SAKT, fungsi aktivasi yang digunakan adalah ReLU. Fungsi aktivasi ini tidak mengubah nilai positif, tetapi mengubah semua nilai negatif menjadi 0. Penghitungan FFN dilakukan dengan Rumus 2.6.

$$F = \text{FFN}(S) = \text{ReLU}(SW_1 + b_1)W_2 + b_2 \quad (2.6)$$

di mana  $W_1 \in \mathbb{R}^{d \times d}, W_2 \in \mathbb{R}^{d \times d}, b_1 \in \mathbb{R}^d, b_2 \in \mathbb{R}^d$  adalah parameter-parameter hasil pelatihan dan hasil dari FFN adalah matriks  $F$ .

#### 8. Prediksi Probabilitas Jawaban Benar

Hasil FFN digunakan untuk menghitung probabilitas siswa menjawab soal ke- $(i)$  dengan benar menggunakan Rumus 2.7.

$$p_i = \sigma(F_i W_p + b_p) \quad (2.7)$$

dengan  $\sigma$  adalah fungsi sigmoid,  $F_i$  adalah elemen-elemen dari matriks  $F$  pada baris ke- $i$ ,  $W_p, b_p$  adalah parameter yang didapat dari pelatihan, dan  $p_i$  adalah nilai skalar yang menunjukkan probabilitas pelajar menjawab soal ke- $i$  dengan benar berdasarkan riwayat mereka.

Model membuat prediksi probabilitas ketika pelajar menjawab soal dengan benar dan aplikasi akan membuka soal baru pada materi yang sedang dipelajari. Model SAKT akan mengambil semua data hasil pembelajaran sebelumnya yang

meliputi kode identifikasi soal serta kebenaran dari jawaban pelajar. Model akan melakukan beberapa proses dan perhitungan untuk mendapatkan probabilitas pelajar menjawab soal selanjutnya dengan benar. Probabilitas ini akan menjadi nilai banding untuk menentukan kesulitan soal selanjutnya yang diberikan kepada pelajar. Apabila probabilitasnya rendah, maka soal selanjutnya akan lebih mudah dan apabila probabilitasnya tinggi, maka soal selanjutnya akan lebih rumit.

## **2.4 Adaptive Learning**

Adaptive learning adalah sebuah teknologi edukasi yang mengubah materi yang disampaikan sesuai dengan kebutuhan dan performa setiap individu. ITS menerapkan *adaptive learning* dengan memberikan alur belajar yang dipersonalisasi melalui adaptasi materi terhadap performa pelajar, meningkatkan retensi/fokus dan pengertian pelajar pada materi yang diberikan oleh ITS [20]. Integrasi *adaptive learning* dalam sebuah ITS dapat meningkatkan keterlibatan dan performa akademik pelajar.

Implementasi *adaptive learning* dalam ITS yang dibangun dalam penelitian ini dilakukan melalui metode *early exit*. Aplikasi ITS akan menyesuaikan total jumlah soal yang ditampilkan berdasarkan atas performa pelajar pada pertanyaan-pertanyaan sebelumnya. Pelajar yang performa baik akan menyelesaikan materi dengan lebih cepat, sedangkan pelajar yang mengalami kesulitan akan mengerjakan soal yang lebih banyak dengan tujuan untuk melatih kemampuan mereka semaksimal mungkin. Metode ini didasarkan atas *intelligent curriculum sequencing* yang melewati topik-topik yang sudah dikuasai oleh pelajar dan berhasil diterapkan serta dievaluasi dalam sistem adaptif perintis seperti ELM-ART dan ISIS-Tutor, yang menunjukkan bahwa pengurutan dan navigasi yang dipersonalisasi dapat menghasilkan proses pembelajaran yang lebih efisien dan efektif [22].

## **2.5 Pengujian Aplikasi**

Dalam evaluasi sistem perangkat lunak, penggunaan metode pengujian yang tepat sangat penting untuk memastikan bahwa sistem tidak hanya berfungsi sesuai spesifikasi, tetapi juga mampu bekerja secara efisien dalam berbagai kondisi. Salah satu metode yang digunakan dalam pengujian fungsional adalah *Black Box Testing*. Metode ini berfokus pada pengujian berdasarkan spesifikasi dan kebutuhan pengguna tanpa memperhatikan struktur internal program atau kode

sumber [21]. Penguji hanya melihat masukan (*input*) dan keluaran (*output*) dari sistem, lalu membandingkannya dengan hasil yang diharapkan. Dengan pendekatan ini, pengujian dapat dilakukan untuk memverifikasi apakah setiap fitur atau fungsi yang diimplementasikan dalam aplikasi berjalan dengan benar sesuai skenario pengguna. *Black Box Testing* sangat bermanfaat terutama dalam konteks pengembangan aplikasi *Intelligent Tutoring System* (ITS), karena dapat menguji apakah sistem memberikan respons yang sesuai terhadap interaksi siswa dalam proses pembelajaran.

Selain memastikan kebenaran fungsionalitas sistem, penting juga untuk mengevaluasi performa model SAKT yang dibangun. Untuk mengevaluasi model SAKT, digunakan metrik *Area Under the ROC Curve* (AUC). Skor dari metrik ini menunjukkan kebenaran dari hasil prediksi probabilitas yang dihasilkan oleh model SAKT. Semakin tinggi skor AUC yang didapat, semakin benar prediksi yang dihasilkan oleh model. Dalam konteks aplikasi ITS, model SAKT digunakan memerlukan skor AUC yang baik agar dapat merepresentasikan pemahaman pelajar pada materi dengan baik sehingga aplikasi dapat melakukan fungsi adaptasi konten berdasarkan riwayat proses pembelajaran.

Pengujian ketiga yang dilakukan adalah *Performance Benchmarking*, yaitu proses pengukuran dan analisis terhadap kinerja sistem dalam hal kecepatan, efisiensi, dan kapasitas penanganan beban [23]. Benchmarking dilakukan dengan menjalankan tes untuk mengamati waktu respon sistem terhadap permintaan pengguna. Dalam konteks aplikasi ITS berbasis model SAKT, *benchmarking* penting untuk memastikan bahwa proses prediksi dapat dilakukan dengan cepat dan efisien tanpa mengganggu kelancaran proses belajar.

Dengan menggabungkan *Black Box Testing* untuk validasi fungsional, skor AUC untuk evaluasi model SAKT, dan *Performance Benchmarking* untuk evaluasi waktu respon *backend* diperoleh gambaran menyeluruh mengenai kualitas sistem yang dibangun. Evaluasi menyeluruh ini menjadi krusial terutama ketika sistem ditujukan untuk digunakan dalam konteks pendidikan digital, di mana akurasi dan performa berpengaruh langsung terhadap pengalaman belajar siswa.