BAB 2

LANDASAN TEORI

2.1 Intelligent Tutoring System (ITS)

Intelligent Tutoring System (ITS) adalah perangkat lunak yang dirancang untuk meniru peran tutor manusia. Hal ini bertujuan untuk memberikan instruksi dan umpan balik yang dipersonalisasi secara langsung kepada siswa, seringkali tanpa harus memerlukan peran guru secara tatap muka. Intelligent Tutoring System (ITS) memanfaatkan kemampuan Kecerdasan Buatan (AI) untuk melahirkan lingkungan belajar yang adaptif sesuai dengan kebutuhan per individu siswa[8].

2.1.1 Tujuan dan Manfaat ITS

Tujuan utama dari penelitian ITS yaitu memberikan panduan personalisasi tingkat lanjut yang melebihi tutor berbantuan komputer tradisional dan melebihi efektivitas tutor manusia. Dan ITS juga bertujuan agar memajukan pemahaman tentang proses kognitif yang terlibat dalam tutor dan pembelajaran siswa[9].

Manfaat Intelligent Tutoring System (ITS) dalam pendidikan sangat signifikan, yaitu:

1. Personalisasi Pembelajaran: ITS dapat menyesuaikan materi pembelajaran dengan kebutuhan siswa, gaya belajar siswa, kecepatan siswa, dan latar belakang individual siswa . Sistem ITS menganalisis data interaksi pengguna secara real-time untuk mengevaluasi kemampuan dan merekomendasikan materi yang sesuai dengan kebutuhan siswa[10].
2. Umpan Balik Real-time: ITS mampu memberikan umpan balik secara langsung, hal ini akan membantu siswa mengidentifikasi dan memperbaiki kesalahan mereka sendiri, serta membangun kepercayaan diri dalam proses pembelajaran[11].
3. Peningkatan Motivasi dan Pemahaman: ITS terbukti mampu secara positif memengaruhi motivasi dan capaian hasil pembelajaran siswa, dalam hal ini seperti pada pendidikan STEM (yang mencakup matematika). ITS dapat meningkatkan minat belajar melalui pengalaman interaktif, multimedia, simulasi, dan fiturisasi[12].

4. Pemerataan Kualitas Pendidikan: ITS menjadi peranan penting dalam membantu pemerataan kualitas pendidikan melalui program e-learning melalui penyediaan akses pembelajaran yang adaptif dan berkualitas tinggi kepada siswa di berbagai kondisi dan tempat[13].

2.1.2 Arsitektur ITS

Arsitektur Intelligent Tutoring System (ITS) pada umumnya memiliki 4 komponen utama yang saling berinteraksi secara dinamis, yaitu:

1. Model Pakar (Domain Model): Komponen ini adalah fondasi pengetahuan dari ITS itu sendiri, yang berisikan fakta-fakta dan aturan-aturan domain subjek yang akan diajarkan dan yang akan merepresentasikan pengetahuan para ahli di bidang tersebut[14].
2. Model Pembelajar (Student Model): Komponen ini akan mencari pengetahuan, keterampilan, dan perilaku belajar dari siswa secara individualistik. Dengan menganalisa interaksi siswa (materi yang dibaca, hasil asesmen, pola respon), ITS dapat membangun representasi akurat tentang tingkat pemahaman dan kemajuan siswa[15].
3. Model Pedagogis (Tutoring Model): Komponen ini bertanggung jawab untuk memilih strategi, taktik, aturan pengajaran yang akan diaplikasikan di ITS. Atas informasi dari Model Pembelajaran, Model Pedagogis akan memutuskan topik apa yang akan dipelajari oleh pengguna selanjutnya dan bagaimana cara terbaik untuk menyajikan materi tersebut[16].
4. Antarmuka Pengguna (User Interface): Ini adalah komponen dari sistem yang berinteraksi langsung dengan siswa, yang akan menyajikan informasi, menerima input, dan memberikan pengalaman belajar yang menarik[17].

2.2 Pembelajaran Matematika Sekolah Dasar

Matematika memegang peranan penting dalam pengembangan daya pikir manusia, membekali siswa dengan kemampuan berpikir logis, analitis, sistematis, kritis, dan kreatif, serta kemampuan bekerja sama. Kemampuan tersebut sangat penting untuk menguasai sains dan teknologi, sehingga menjadi mata pelajaran

yang perlu diajarkan di semua jenjang pendidikan, mulai dari sekolah dasar hingga perguruan tinggi[2].

Tetapi pembelajaran matematika di sekolah dasar pada negara Indonesia diperhadapkan dengan berbagai tantangan yang kompleks, seperti:

1. Kesulitan Kognitif: Banyak siswa mengalami kesulitan yang signifikan dengan keterampilan berhitung dan kemampuan memecahkan masalah matematika. Kesulitan-kesulitan ini sering kali berakar pada kurangnya pemahaman yang kuat tentang konsep-konsep matematika dasar. Siswa juga menghadapi tantangan dalam menerjemahkan soal cerita ke dalam bentuk matematika[2].
2. Kesulitan Afektif: Matematika sering dianggap sebagai mata pelajaran yang sulit dan membosankan oleh sebagian besar siswa, sehingga menyebabkan kurangnya minat dan motivasi. Rasa takut membuat kesalahan dan kecemasan saat belajar matematika juga menjadi kendala yang signifikan[2].
3. Keterbatasan Model Pengajaran Konvensional Model: Pengajaran tradisional sering kali bersifat "one-size-fits-all", yang membuatnya sulit mengakomodasi gaya belajar, kecepatan, dan latar belakang siswa yang beragam. Guru menghadapi tantangan yang signifikan dalam menerapkan pembelajaran yang dibedakan karena keberagaman karakter, gaya belajar, dan minat siswa[18].

2.3 Metode K-Means Clustering

Clustering pada K-Means merupakan algoritma pembelajaran tanpa pengawasan (*unsupervised learning*) yang populer, yang digunakan dalam mempartisi kumpulan data ke dalam sejumlah kluster (K) yang telah ditentukan sebelumnya berdasarkan kesamaan data. Tujuannya adalah mengelompokkan titik data yang serupa dan menemukan pola atau struktur yang mendasarinya dalam data.

Dalam penelitian ini, K-Means digunakan untuk menganalisis dan mengelompokkan topik-topik pelajaran berdasarkan karakteristik interaksi siswa. Proses ini diawali dengan tahap rekayasa fitur untuk mempersiapkan data.

2.3.1 Rekayasa Fitur (Feature Engineering)

Sebelum proses clustering, data mentah dari tabel `answer_logs` diagregasi untuk menghasilkan fitur yang representatif untuk setiap topik pelajaran. Misalkan T_j adalah himpunan semua jawaban dari seluruh siswa untuk topik ke- j .

A Rata-rata Kesulitan Topik (`avg_difficulty`)

Rumus 2.1 ini mengukur seberapa sering siswa melakukan kesalahan pada suatu topik. Misalkan x_i adalah nilai `is_correct` untuk jawaban ke- i dalam himpunan T_j (di mana $x_i = 1$ jika benar dan $x_i = 0$ jika salah), dan N_j adalah jumlah total jawaban untuk topik j . Maka, `avg_difficulty` (D_j) untuk topik j dihitung sebagai:

$$D_j = 1 - \frac{1}{N_j} \sum_{i=1}^{N_j} x_i \quad (2.1)$$

B Rata-rata Kompleksitas Topik (`avg_complexity`)

Rumus 2.2 ini mengukur rata-rata waktu yang dibutuhkan siswa untuk menjawab soal pada suatu topik. Misalkan y_i adalah nilai `time_spent` untuk jawaban ke- i dalam himpunan T_j . Maka, `avg_complexity` (C_j) untuk topik j dihitung sebagai:

$$C_j = \frac{1}{N_j} \sum_{i=1}^{N_j} y_i \quad (2.2)$$

Setiap topik pelajaran kini direpresentasikan sebagai sebuah titik data dalam ruang dua dimensi (D_j, C_j).

2.3.2 Prinsip Kerja K-Means Clustering

Algoritma *K-Means* bekerja secara iteratif melalui langkah-langkah berikut:

1. Inisialisasi: Pilih K titik data secara acak dari himpunan data topik sebagai *centroid* awal.

2. Langkah Penugasan (Assignment Step): Setiap titik data topik ditetapkan ke klaster yang *centroid*-nya paling dekat. Kedekatan diukur menggunakan Jarak Euclidean. Misalkan sebuah titik data topik adalah $p = (p_1, p_2)$ dan sebuah *centroid* adalah $c = (c_1, c_2)$, maka jarak Euclidean $d(p, c)$ dihitung dengan rumus:

$$d(p, c) = \sqrt{(p_1 - c_1)^2 + (p_2 - c_2)^2} \quad (2.3)$$

3. Langkah Pembaruan (Update Step): *Centroid* baru untuk setiap klaster dihitung ulang dengan mengambil rata-rata dari semua titik data yang telah ditetapkan ke klaster tersebut. Misalkan S_k adalah himpunan semua titik data (topik) yang termasuk dalam Klaster k , dan $|S_k|$ adalah jumlah anggota dalam himpunan tersebut. Maka, *centroid* baru μ_k untuk Klaster k dihitung dengan rumus:

$$\mu_k = \frac{1}{|S_k|} \sum_{p \in S_k} p \quad (2.4)$$

4. Iterasi: Langkah 2 dan 3 diulang hingga posisi *centroid* tidak lagi berubah secara signifikan (konvergen).

Tujuan akhir dari algoritma K-Means adalah untuk meminimalkan fungsi objektif *Within-Cluster Sum of Squares* (WCSS), yang merupakan jumlah dari kuadrat jarak antara setiap titik data dengan *centroid* dari klaster-nya, seperti yang ditunjukkan pada Rumus 2.5.

$$WCSS = \sum_{k=1}^K \sum_{p \in S_k} \|p - \mu_k\|^2 \quad (2.5)$$

Dengan penjelasan komponen sebagai berikut:

- K : Jumlah klaster yang telah ditentukan sebelumnya.
- S_k : Himpunan semua titik data (topik) di Klaster k .
- μ_k : *Centroid* dari semua titik data yang termasuk dalam klaster S_k .

- $\|p - \mu_k\|^2$: Kuadrat jarak Euclidean antara titik data p dan *centroid* kluster μ_k .

2.3.3 Aplikasi K-Means Clustering dalam ITS (Pemodelan Siswa)

Dalam konteks ITS, K-Means Clustering memiliki potensi besar untuk meningkatkan kemampuan adaptasi dan tingkat efektivitas platform pendidikan dengan mengorganisasikan peserta didik ke dalam kelompok-kelompok yang bermakna. Aplikasi utamanya meliputi:

1. Pengelompokan Siswa: Pengelompokan siswa berdasarkan berbagai kriteria seperti preferensi belajar, tingkat kemahiran, dan pola keterlibatan mereka dalam proses pembelajaran. Hal ini membantu mengatasi tantangan karakter siswa dan gaya belajar yang beragam yang dihadapi guru[5].
2. Analisis Pola Respons: Digunakan untuk menganalisis respons siswa terhadap pertanyaan atau kuesioner, untuk mengidentifikasi pola umum dalam pemahaman atau kesulitan mereka, dan bahkan untuk menyimpulkan alur pemikiran yang digunakan oleh siswa[19].
3. Sistem Rekomendasi: Membantu mengatasi masalah “cold-start” dalam sistem rekomendasi pendidikan dengan mengelompokkan pengguna berdasarkan pola interaksi mereka, sehingga sistem dapat merekomendasikan konten yang relevan sejak awal. Penerapan pengelompokan dapat meningkatkan akurasi model rekomendasi[20].
4. Personalisasi Skalabel: Memungkinkan ITS menerapkan strategi pedagogis khusus kelompok, seperti jalur pembelajaran yang berbeda atau tingkat kesulitan konten, tanpa memerlukan model yang unik untuk setiap siswa[21].

2.4 Metode Rule-Based System

Rule-Based System (RBS) dalam Intelligent Tutoring System (ITS) merepresentasikan pengetahuan domain dan keahlian pedagogis melalui serangkaian aturan produksi. Sistem ini berfungsi sebagai “otak” pedagogis ITS, membantu memahami pemikiran dan kesalahan siswa, memandu desain tutor, dan menerapkan strategi pengajaran yang efektif[4].

2.4.1 Prinsip Kerja Rule-Based System

Algoritma Rule-Based beroperasi berdasarkan serangkaian aturan "IF-THEN" yang telah ditentukan sebelumnya, yang biasanya dikodekan oleh ahli domain. Aturan-aturan ini menggunakan konektif logis seperti AND, OR, NOT untuk membentuk fungsi logis. Sifat eksplisit dari aturan dalam sistem menandakan bahwa proses pengambilan keputusan dalam ITS ini sangat transparan dan dapat diinterpretasikan[4].

Dalam aplikasi ini, RBS digunakan untuk dua tugas utama yang dapat direpresentasikan secara formal sebagai berikut:

A Penentuan Pathway Siswa

Tugas pertama dari sistem ini adalah untuk mengklasifikasikan tingkat penguasaan siswa ke dalam salah satu dari tiga kategori ('Pathway') berdasarkan skor akhir yang mereka peroleh.

Misalkan:

- S adalah **skor akhir** siswa (jumlah jawaban yang benar).
- N adalah **jumlah total soal** dalam satu sesi kuis (dalam kasus ini, $N = 20$).

Logika penentuan 'Pathway' dapat direpresentasikan secara formal sebagai sebuah **fungsi piecewise** $f(S)$ sebagai berikut:

$$f(S) = \begin{cases} \text{Advanced} & \text{jika } S \geq 15 \\ \text{Beginner} & \text{jika } 8 \leq S \leq 14 \\ \text{Amateur} & \text{jika } S < 8 \end{cases} \quad (2.6)$$

Fungsi ini secara langsung memetakan setiap kemungkinan nilai skor S ke salah satu dari tiga 'Pathway' yang telah ditentukan.

Logika penentuan 'Pathway' dihasilkan menurut konsep *Mastery Learning*[22] dan *Learning Hierarchies*[23].

B Penentuan Aksi Remediasi

Tugas kedua dari sistem ini adalah menentukan tindakan selanjutnya yang harus diambil oleh siswa berdasarkan 'Pathway' yang baru saja mereka peroleh.

Logika ini dapat direpresentasikan sebagai **aturan produksi (production rules)** sederhana:

1. **Aturan untuk ‘Advanced’:**

IF ‘Pathway’ = ‘Advanced’

THEN

- ‘Aksi’ = Lanjut ke kelas berikutnya.
- ‘Status Kelas’ = Tandai sebagai ‘Selesai’.
- ‘Rekomendasi’ = Aktifkan sistem rekomendasi K-Means.

2. **Aturan untuk ‘Beginner’:**

IF ‘Pathway’ = ‘Beginner’

THEN

- ‘Aksi’ = Mengulang soal yang dijawab salah.
- ‘Simpan Data’ = Daftar soal yang salah disimpan ke dalam sesi.

3. **Aturan untuk ‘Amateur’:**

IF ‘Pathway’ = ‘Amateur’

THEN

- ‘Aksi’ = Mengulang seluruh topik dari awal.
- ‘Hapus Data’ = Riwayat soal salah (jika ada) dihapus.

Representasi formal ini menunjukkan bagaimana sistem secara konsisten dan transparan menerapkan aturan yang telah didefinisikan untuk menilai dan membimbing siswa melalui kurikulum.

2.4.2 Aplikasi Rule-Based System dalam ITS

Rule-Based System memungkinkan ITS untuk memberikan umpan balik adaptif dan menentukan strategi pengajaran yang tepat, seperti:

1. Umpan Balik Adaptif: Memberikan instruksi langkah demi langkah atau koreksi saat siswa memecahkan masalah (tingkat mikro).

2. Penentuan Jalur Pembelajaran: Menentukan masalah atau materi berikutnya yang akan disajikan kepada siswa berdasarkan kemajuan dan pemahaman mereka (Level Makro).
3. Representasi Kompetensi Siswa: Mengkodekan pengetahuan domain dan keahlian pedagogi ke dalam aturan yang dapat dijalankan oleh sistem, memastikan bahwa intervensi pedagogi yang diberikan tepat sasaran dan relevan dengan kebutuhan spesifik siswa[4].
4. Peningkatan Pembelajaran: Cognitive Tutors yang menggunakan model rule-based telah terbukti meningkatkan pembelajaran siswa secara signifikan di berbagai domain, termasuk matematika[4].

2.5 Evaluasi Metrik Internal untuk K-Means

Dalam algoritma K-Means, penentuan jumlah *cluster* (K) yang optimal merupakan langkah yang penting. Hal ini dilakukan, karena K-Means adalah metode *unsupervised learning*, tidak ada "jawaban benar" yang dapat digunakan sebagai acuan. Oleh karena itu, digunakan **metrik evaluasi internal** untuk menilai kualitas struktur *cluster* yang terbentuk berdasarkan karakteristik data itu sendiri.

Metode ini disebut "internal" karena tidak memerlukan label eksternal untuk validasi. Dalam penelitian ini, dua metrik internal yang umum digunakan untuk menentukan nilai K yang optimal adalah Metode Siku dan Skor Silhouette.

2.5.1 Metode Siku (Elbow Method)

Metode Siku adalah sebuah heuristik visual yang digunakan untuk menemukan jumlah *cluster* yang paling baik dengan cara memplot nilai **Inersia** atau *Within-Cluster Sum of Squares* (WCSS) terhadap jumlah *cluster* (K) yang berbeda.

- Konsep: Inersia mengukur seberapa padat dan kompak sebuah *cluster*. Secara spesifik, ia menghitung jumlah dari kuadrat jarak antara setiap titik data dengan *centroid* dari *cluster*-nya. Nilai inersia yang lebih rendah menandakan bahwa anggota-anggota dalam sebuah *cluster* sangat berdekatan satu sama lain, yang berarti *cluster* tersebut bagus.
- Proses:

1. Algoritma K-Means dijalankan beberapa kali untuk rentang nilai K yang berbeda (misalnya, K dari 2 hingga 10).
 2. Untuk setiap nilai K, total WCSS dihitung.
 3. Nilai WCSS kemudian diplot pada sebuah grafik, dengan jumlah K sebagai sumbu-X dan WCSS sebagai sumbu-Y.
- Interpretasi: Seiring bertambahnya jumlah K, nilai WCSS akan selalu menurun. Namun, kita mencari titik di mana penurunan ini mulai melambat secara drastis, membentuk sebuah "siku" (*elbow*) pada grafik. Titik siku ini dianggap sebagai titik keseimbangan yang baik antara meminimalkan WCSS dan menghindari *overfitting* (memiliki terlalu banyak *cluster*). Titik siku ini merepresentasikan jumlah *cluster* di mana penambahan *cluster* baru tidak lagi memberikan peningkatan kualitas yang signifikan. [24]
 - Rumus Matematis (WCSS):

$$WCSS = \sum_{k=1}^K \sum_{p \in S_k} \|p - \mu_k\|^2 \quad (2.7)$$

Di mana K adalah jumlah *cluster*, S_k adalah himpunan titik data di Cluster k , dan μ_k adalah *centroid* dari Cluster k .

2.5.2 Skor Silhouette (Silhouette Score)

Skor Silhouette adalah metrik yang lebih kuat secara matematis karena tidak hanya mengukur kepadatan internal sebuah *cluster* (kohesi), tetapi juga tingkat pemisahannya dengan *cluster* lain (separasi).

- **Konsep:** Untuk setiap titik data, Skor Silhouette menghitung seberapa mirip titik tersebut dengan *cluster*-nya sendiri dibandingkan dengan *cluster* tetangga terdekat.
- **Komponen Perhitungan:** Untuk setiap titik data i , dua nilai utama dihitung:
 1. Kohesi ($a(i)$): Mengukur seberapa dekat sebuah titik data dengan anggota lain di dalam *cluster*-nya sendiri. Nilai ini adalah **jarak rata-rata** dari titik i ke semua titik lain di dalam *cluster* yang sama (C_k). Nilai

$a(i)$ yang kecil menandakan kohesi yang baik.

$$a(i) = \frac{1}{|C_k| - 1} \sum_{j \in C_k, j \neq i} d(i, j) \quad (2.8)$$

Di mana $d(i, j)$ adalah jarak (misalnya, Euclidean) antara titik i dan titik j .

2. Separasi ($b(i)$): Mengukur seberapa jauh sebuah titik data dengan *cluster* tetangga terdekat. Nilai ini adalah **jarak rata-rata minimum** dari titik i ke semua titik di dalam *cluster* lain (C_l , di mana $l \neq k$). Nilai $b(i)$ yang besar menandakan separasi yang baik.

$$b(i) = \min_{l \neq k} \left(\frac{1}{|C_l|} \sum_{j \in C_l} d(i, j) \right) \quad (2.9)$$

- **Interpretasi:** Skor Silhouette untuk satu titik data dihitung dengan menggabungkan nilai kohesi dan separasi, dan berkisar dari -1 hingga +1. Skor Silhouette untuk keseluruhan model adalah rata-rata dari skor semua titik data. Nilai K yang optimal adalah yang menghasilkan rata-rata Skor Silhouette tertinggi.[25]

- Skor mendekati +1: Menandakan *cluster* padat dan terpisah dengan baik (kohesi baik, separasi baik).
- Skor mendekati 0: Menandakan *cluster* tumpang tindih.
- Skor negatif: Menandakan titik data kemungkinan besar salah diklasifikasikan.

- **Rumus Matematis (Skor Silhouette untuk satu titik data i):**

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2.10)$$