

## BAB 2

### LANDASAN TEORI

#### 2.1 Next.js Framework

Next.js merupakan framework React yang dikembangkan oleh Vercel, dirancang untuk membangun aplikasi web modern dengan performa tinggi dan pengalaman pengembang yang optimal. Framework ini telah menjadi pilihan populer dalam pengembangan aplikasi web pendidikan karena kemampuannya dalam mengoptimalkan performa, SEO, dan kemudahan deployment [24, 25]. Dalam konteks pengembangan sistem pembelajaran cerdas, Next.js menawarkan berbagai keunggulan teknis yang mendukung implementasi fitur-fitur kompleks seperti gamifikasi dan integrasi AI agent.

Salah satu keunggulan utama Next.js adalah kemampuan *Server-Side Rendering* (SSR) dan *Static Site Generation* (SSG) yang memungkinkan aplikasi memiliki waktu muat yang lebih cepat dan performa SEO yang lebih baik [26, 27]. Fitur ini sangat penting untuk aplikasi pembelajaran online yang membutuhkan aksesibilitas tinggi dan pengalaman pengguna yang responsif. Penelitian menunjukkan bahwa kecepatan loading aplikasi web berpengaruh signifikan terhadap engagement dan retensi pengguna dalam platform pembelajaran digital [28].

Next.js juga mendukung *API Routes* yang memungkinkan pengembangan backend dan frontend dalam satu framework, menciptakan arsitektur *full-stack* yang efisien. Kemampuan ini memfasilitasi integrasi yang lebih seamless antara komponen frontend dengan layanan backend, termasuk integrasi dengan AI microservices dan database untuk menyimpan data gamifikasi dan progress pembelajaran [24]. Dalam pengembangan sistem pembelajaran cerdas, arsitektur terintegrasi ini memungkinkan real-time updates untuk fitur-fitur seperti leaderboard, progress tracking, dan adaptive feedback.

##### a. *Performance Optimization*

Next.js menyediakan berbagai fitur optimasi performa out-of-the-box, termasuk *automatic code splitting*, *image optimization*, dan *prefetching*. Fitur-fitur ini sangat relevan untuk aplikasi pembelajaran yang sering menampilkan konten multimedia dan interaktif [25, 28]. Optimasi performa yang baik berkontribusi langsung terhadap pengalaman belajar yang lebih

smooth dan engagement yang lebih tinggi.

b. *TypeScript Integration*

Dukungan TypeScript yang built-in dalam Next.js memberikan keuntungan dalam pengembangan aplikasi skala besar seperti sistem pembelajaran cerdas. Type safety yang ditawarkan TypeScript membantu mengurangi bug dan meningkatkan maintainability kode, yang sangat penting dalam pengembangan sistem pembelajaran yang kompleks [24]. Penelitian menunjukkan bahwa penggunaan TypeScript dalam proyek web skala besar dapat mengurangi bug production hingga 38%.

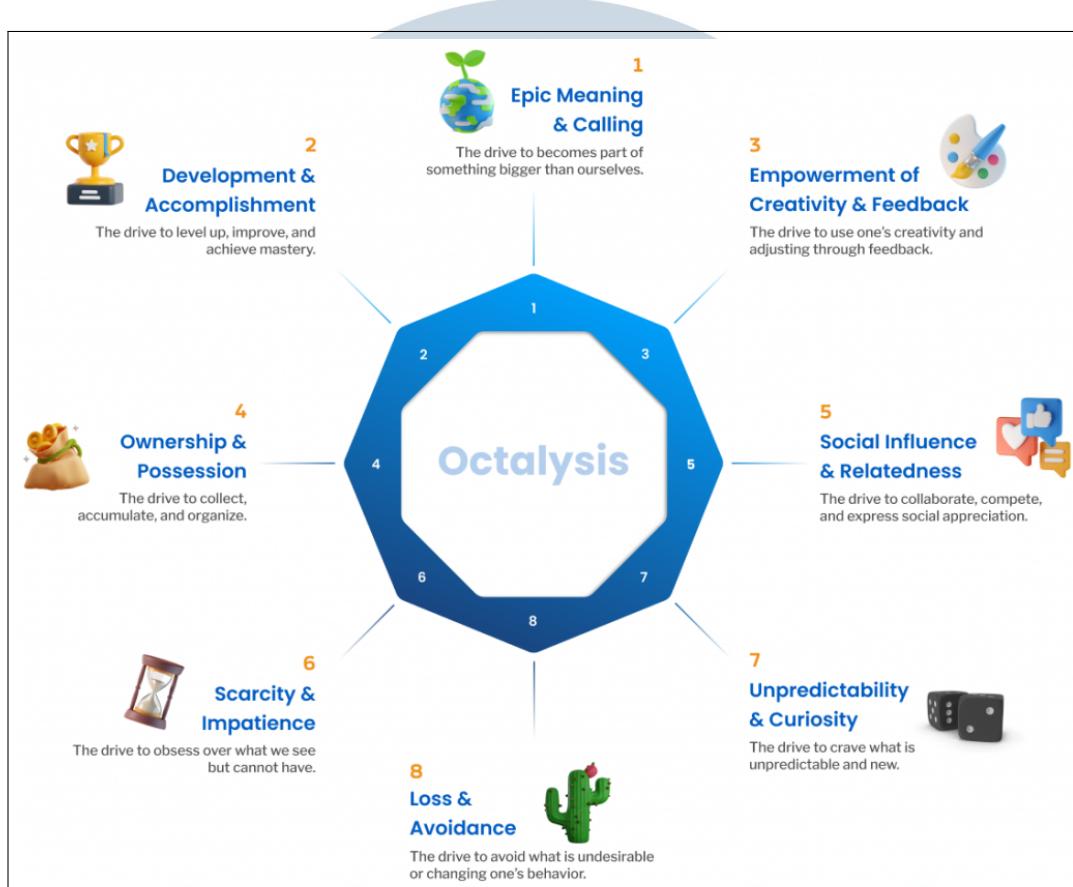
## 2.2 Gamifikasi dan Framework Octalysis

Gamifikasi telah menjadi topik populer dalam penelitian pendidikan, karena kemampuannya meningkatkan motivasi serta keterlibatan siswa. Elemen permainan seperti poin, level, dan penghargaan terbukti menjadi insentif yang efektif untuk mendorong siswa lebih aktif dalam proses belajar. Beberapa studi melaporkan bahwa penerapan gamifikasi mampu menghadirkan suasana belajar yang lebih hidup, menyenangkan, sekaligus membuat siswa lebih terhubung dengan materi ajar [1]. Dengan kata lain, gamifikasi bukan hanya strategi motivasional, melainkan juga cara untuk memperkuat pengalaman belajar.

Gamifikasi dapat menciptakan ekosistem belajar yang kompetitif namun tetap kolaboratif. Dalam lingkungan seperti ini, siswa tidak hanya termotivasi untuk berprestasi, tetapi juga saling mendukung teman sekelas mereka. Interaksi sosial yang muncul dari elemen permainan ini terbukti dapat meningkatkan keterampilan sosial dan emosional, yang pada akhirnya turut berkontribusi terhadap perkembangan pribadi siswa [5]. Framework Octalysis hadir sebagai salah satu pendekatan sistematis untuk merancang gamifikasi dalam pendidikan. Framework ini menekankan delapan pendorong motivasi intrinsik yang memengaruhi perilaku manusia, seperti pencapaian, pengusaan, kepemilikan, hingga pengaruh sosial [29]. Dengan memahami motivasi di balik perilaku siswa, pendidik dapat menyusun strategi gamifikasi yang lebih relevan dan personal, serta menyesuaikan berdasarkan umpan balik siswa.

Framework Octalysis dapat divisualisasikan melalui diagram yang menunjukkan kedelapan pendorong motivasi yang saling berinteraksi dalam bentuk oktagon. Gambar 2.1 menampilkan struktur lengkap framework ini dengan

penempatan setiap core drive pada posisi strategis yang mencerminkan karakteristik motivasionalnya.



Gambar 2.1. Diagram Framework Octalysis

Sumber: [30]

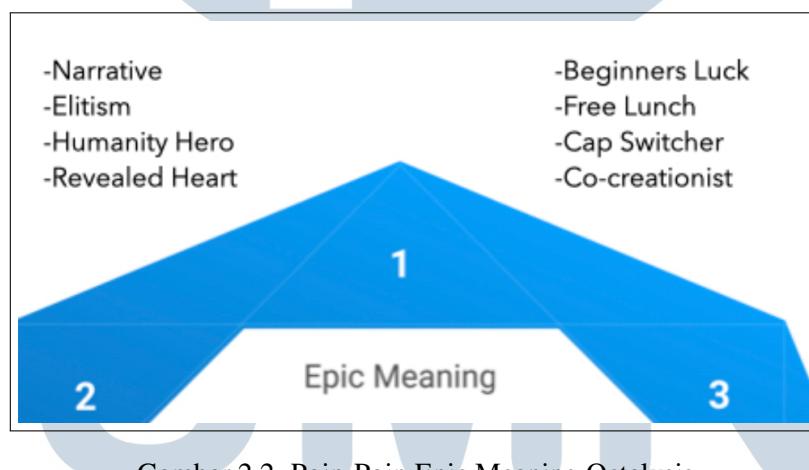
Octalysis, yang dikembangkan oleh Yu-kai Chou, memberikan kerangka untuk memahami bagaimana mekanisme permainan dapat memengaruhi motivasi dan keterlibatan pengguna. Framework ini membagi motivasi ke dalam delapan pendorong inti: *Epic Meaning*, *Development & Accomplishment*, *Empowerment of Creativity*, *Social Influence*, *Scarcity*, *Unpredictability*, *Loss & Avoidance*, serta *Ownership*. Setiap pendorong ini berperan penting dalam perancangan strategi gamifikasi yang efektif untuk sistem pembelajaran cerdas.

Penerapan Octalysis dalam sistem pembelajaran cerdas memungkinkan pengalaman belajar yang lebih personal. Sebagai contoh, pendorong *Development & Accomplishment* bisa diwujudkan dengan memberikan penghargaan ketika siswa berhasil menyelesaikan tugas, sementara *Empowerment of Creativity* mendorong mereka untuk mengeksplorasi berbagai solusi kreatif. Jika dikombinasikan dengan

agen AI, strategi ini dapat menciptakan interaksi yang dinamis serta mendukung hasil pembelajaran yang berkelanjutan [31]. Integrasi agen AI dengan Octalysis memberi peluang untuk menyesuaikan pengalaman belajar berdasarkan profil masing-masing siswa. Melalui analitik pembelajaran, sistem dapat mengenali kelebihan maupun kelemahan siswa, lalu memberikan intervensi yang relevan dan tepat waktu [32]. Hal ini membuat pengalaman belajar tidak hanya adaptif, tetapi juga terasa lebih personal.

a. *Epic Meaning & Calling*

Epic Meaning adalah ketika pengguna merasa sedang melakukan sesuatu yang lebih besar dari dirinya sendiri atau merasa “dipilih” untuk suatu tujuan. Contoh penerapannya adalah memberikan narasi atau misi yang bermakna dalam proses belajar, sehingga siswa merasa memiliki peran penting dalam pencapaian tujuan pembelajaran. Elemen-elemen spesifiknya dapat dilihat pada Gambar 2.2 yang menampilkan berbagai cara untuk mengimplementasikan pendorong motivasi ini.

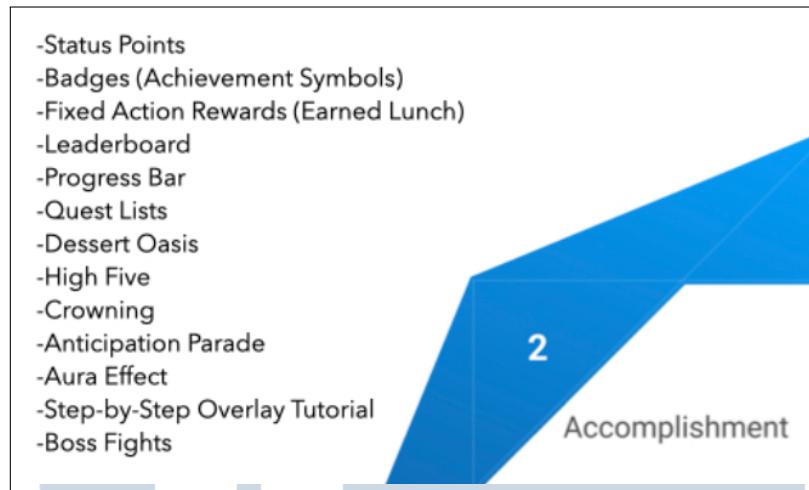


Gambar 2.2. Poin-Poin Epic Meaning Octalysis

Sumber: [30]

b. *Development & Accomplishment*

Development & Accomplishment berkaitan dengan dorongan untuk meraih pencapaian, mengatasi tantangan, dan merasakan perkembangan kemampuan. Elemen-elemen spesifik dari core drive ini dapat dilihat pada Gambar 2.3 yang menunjukkan berbagai strategi implementasi dalam konteks pembelajaran.



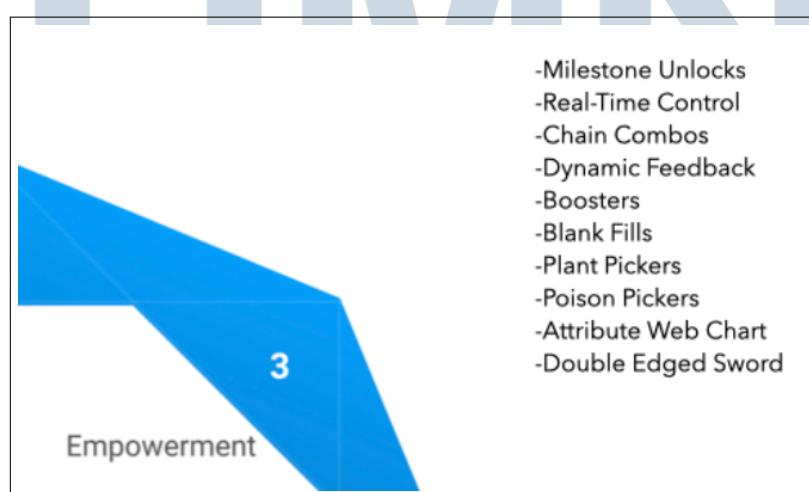
Gambar 2.3. Poin-Poin Development & Accomplishment Octalysis

Sumber: [30]

Dalam konteks sistem pembelajaran cerdas, hal ini dapat diwujudkan melalui pemberian penghargaan, lencana, atau tingkat pencapaian yang hanya bisa diraih setelah melewati tantangan tertentu.

c. *Empowerment of Creativity & Feedback*

Empowerment muncul ketika siswa diberi ruang untuk berpikir kreatif dan menemukan solusi mereka sendiri. Aspek-aspek utama dari pendorong motivasi ini diilustrasikan dalam Gambar 2.4 yang menampilkan berbagai cara untuk memberikan pemberdayaan dalam pembelajaran.



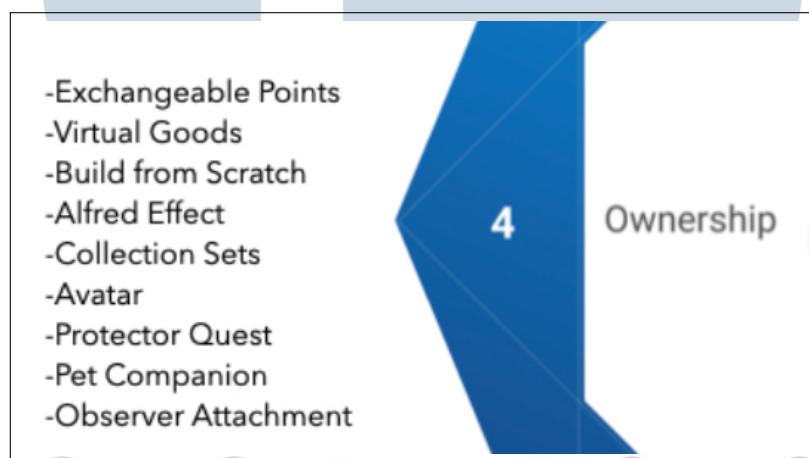
Gambar 2.4. Poin-Poin Empowerment Octalysis

Sumber: [30]

Sistem pembelajaran cerdas dapat memfasilitasi hal ini dengan menyediakan berbagai cara menyelesaikan tugas dan memberikan umpan balik adaptif secara real-time [32].

d. *Ownership & Possession*

Ownership muncul ketika siswa merasa memiliki sesuatu dalam sistem, misalnya avatar, progres belajar, atau item digital tertentu. Rasa kepemilikan ini meningkatkan keterlibatan karena siswa ingin terus merawat dan mengembangkan “kepunyaan” mereka dalam sistem. Elemen-elemen spesifik dari core drive ini dapat dilihat pada Gambar 2.5 yang menunjukkan berbagai strategi implementasi dalam konteks pembelajaran.



Gambar 2.5. Poin-Poin Ownership Octalysis

Sumber: [30]

e. *Social Influence & Relatedness*

Social Influence melibatkan interaksi sosial seperti kompetisi, kolaborasi, atau pengakuan dari orang lain. Berbagai strategi untuk mengimplementasikan pengaruh sosial dalam pembelajaran dapat dilihat pada Gambar 2.6 yang menunjukkan elemen-elemen kunci dari core drive ini.



Gambar 2.6. Poin-Poin Social Influence Octalysis

Sumber: [30]

Contohnya adalah papan peringkat atau fitur berbagi pencapaian dengan teman sebaya, yang dapat mendorong motivasi belajar [1].

#### f. *Scarcity & Impatience*

Scarcity adalah motivasi yang muncul ketika sesuatu sulit untuk diperoleh atau dibatasi. Gambar 2.7 menampilkan berbagai teknik untuk menciptakan kelangkaan yang dapat meningkatkan motivasi dalam sistem pembelajaran.



Gambar 2.7. Poin-Poin Scarcity Octalysis

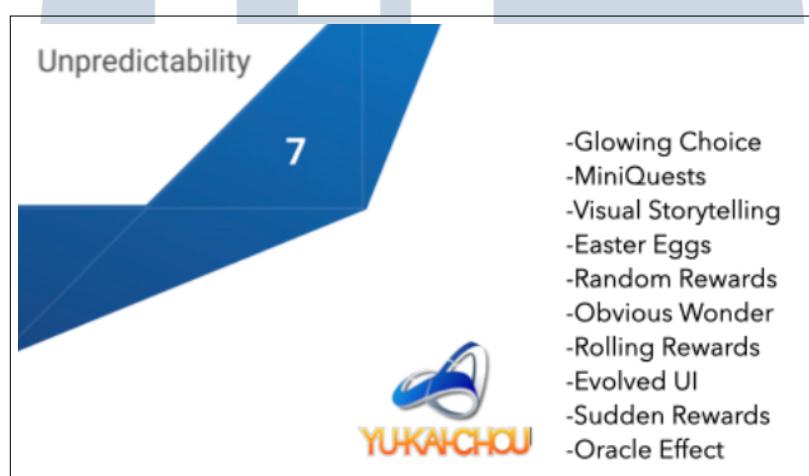
Sumber: [30]

Dalam pembelajaran, hal ini dapat diimplementasikan dengan akses terbatas

ke konten tertentu yang hanya bisa dibuka setelah syarat tertentu dipenuhi.

g. *Unpredictability & Curiosity*

Unpredictability berkaitan dengan kejutan dan ketidakpastian. Dalam konteks sistem pembelajaran cerdas, elemen ini dapat diwujudkan dengan memberikan “misi misterius” atau soal acak yang tidak selalu sama, sehingga menjaga rasa penasaran siswa. Elemen-elemen spesifik dari core drive ini dapat dilihat pada Gambar 2.8 yang menunjukkan berbagai strategi implementasi dalam konteks pembelajaran.



Gambar 2.8. Poin-Poin Unpredictability Octalysis

Sumber: [30]

h. *Loss & Avoidance*

Loss & Avoidance memotivasi siswa untuk bertindak agar tidak kehilangan sesuatu yang berharga, misalnya kehilangan poin atau akses ke fitur jika tidak menyelesaikan tugas. Gambar 2.9 menjelaskan berbagai aspek dari motivasi penghindaran kerugian yang dapat diterapkan dalam sistem pembelajaran.



Gambar 2.9. Poin-Poin Avoidance Octalysis

Sumber: [30]

Meskipun Octalysis menawarkan banyak potensi, tantangan tetap ada. Salah satunya adalah memahami preferensi individu siswa agar strategi yang dipilih benar-benar tepat sasaran. Selain itu, keterbatasan teknologi dalam menyesuaikan konten secara real-time masih menjadi hambatan [33]. Oleh karena itu, penelitian lebih lanjut dibutuhkan untuk mengoptimalkan penerapan framework ini dalam konteks pendidikan modern.

Penting untuk membedakan antara konsep permainan (*game*) utuh dan gamifikasi. Perbedaan mendasar terletak pada tujuan dan konteks penggunaannya. Sebuah *game* dirancang sebagai sistem tertutup yang bertujuan utama untuk hiburan, di mana pemain masuk ke dalam dunia permainan yang terpisah dari realitas. Sebaliknya, gamifikasi didefinisikan sebagai penerapan elemen-elemen desain permainan, seperti poin, lencana, dan papan peringkat, ke dalam konteks non-permainan (seperti pendidikan atau produktivitas) untuk memengaruhi perilaku dan meningkatkan motivasi pengguna dalam menyelesaikan aktivitas dunia nyata [10]. Dalam penelitian ini, sistem yang dibangun menerapkan gamifikasi karena elemen permainan digunakan untuk mendukung aktivitas pembelajaran pemrograman, bukan menggantikannya dengan permainan simulasi murni.

### 2.3 Perbedaan Serious Game dan Aplikasi Gamifikasi

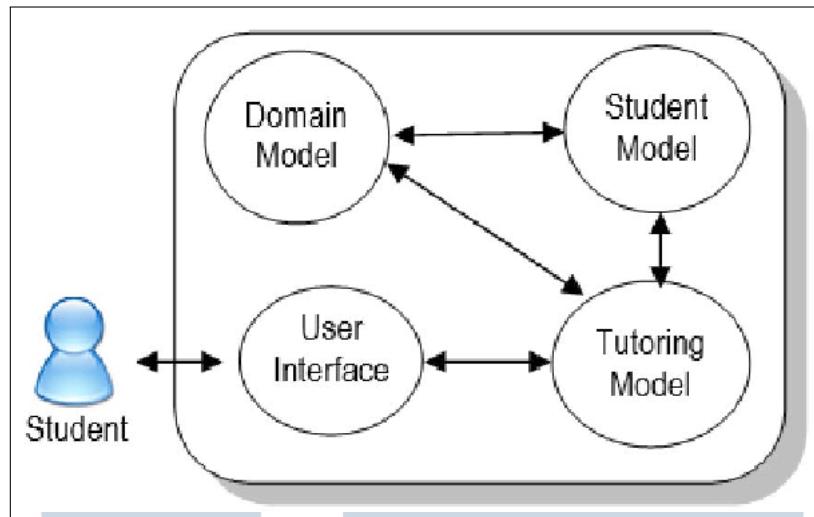
Dalam pengembangan perangkat lunak edukasi, terdapat perbedaan mendasar antara *Serious Game* dan Aplikasi Tergamifikasi (*Gamified Application*). *Serious Game* didefinisikan sebagai permainan utuh yang dirancang khusus untuk tujuan pembelajaran, di mana konten edukasi terintegrasi secara intrinsik ke dalam

mekanik permainan itu sendiri. Sebaliknya, gamifikasi merupakan penerapan elemen desain dan mekanik permainan ke dalam konteks non-permainan untuk meningkatkan keterlibatan dan motivasi siswa tanpa mengubah aktivitas utama menjadi sebuah permainan simulasi penuh [10].

Perbedaan struktural utama antara keduanya terletak pada fokus aktivitas pengguna. Pada *Serious Game*, pembelajaran terjadi melalui pengalaman bermain (*gameplay*), sedangkan pada aplikasi tergamifikasi, sistem dibangun di atas kerangka instruksional tradisional yang diperkaya dengan lapisan motivasional. Pendekatan ini memungkinkan integrasi elemen permainan seperti poin, lencana, dan papan peringkat ke dalam sistem manajemen pembelajaran (LMS) atau aplikasi edukasi untuk mempertahankan keterlibatan siswa dalam proses belajar yang terstruktur [18]. Berdasarkan definisi tersebut, platform "Starcoder" dalam penelitian ini diklasifikasikan sebagai Aplikasi Tergamifikasi (*Gamified Application*), bukan *Serious Game*. Aktivitas inti yang dilakukan pengguna tetaplah aktivitas pembelajaran fungsional, yaitu membaca materi, mengerjakan kuis, dan menulis kode sintaksis. Elemen *framework Octalysis* dan agen AI diterapkan sebagai lapisan pendukung untuk mempersonalisasi pengalaman dan menjaga motivasi pengguna dalam menyelesaikan aktivitas pembelajaran tersebut [21].

## 2.4 Sistem Pembelajaran Cerdas

Sistem pembelajaran cerdas (*Intelligent Tutoring Systems*, ITS) dirancang untuk memberikan instruksi dan umpan balik yang personal dengan menyesuaikan proses belajar sesuai kebutuhan individual siswa [14]. Arsitektur ITS tradisional terdiri dari empat komponen utama seperti yang ditunjukkan pada Gambar 2.10.



Gambar 2.10. Komponen Utama *Intelligent Tutoring System*

Sumber: [34]

Komponen utama tersebut meliputi Domain Model (pengetahuan pembelajaran), Learner Model (karakteristik siswa), Tutoring Model (strategi instruksional), dan Interface Model (interaksi pengguna) [35]. Dalam implementasi modern, struktur ini diperkuat dengan algoritma AI untuk personalisasi yang lebih akurat.

Integrasi gamifikasi dalam ITS menggunakan framework Octalysis terbukti efektif meningkatkan keterlibatan dan hasil akademik siswa [31]. Elemen permainan seperti poin dan sistem *level-up* dapat mendorong partisipasi aktif dalam pembelajaran [1]. Namun, perancangan harus mempertimbangkan variasi respons individual karena tidak semua siswa merespons gamifikasi dengan cara yang sama [36]. Selain itu, aspek privasi data dan keamanan informasi harus menjadi perhatian utama dalam implementasi ITS berbasis AI [33].

Sifat "adaptif" dalam konteks ITS mengacu pada kemampuan sistem untuk secara dinamis menyesuaikan materi dan strategi pengajaran berdasarkan respons dan kinerja siswa secara waktu nyata, berbeda dengan sistem linier yang menyajikan konten yang sama untuk semua pengguna. Dalam sistem Starcoder, adaptabilitas ini diwujudkan melalui mekanisme deteksi kegagalan otomatis. Ketika nilai siswa berada di bawah ambang batas tertentu, sistem secara cerdas mengubah alur pembelajaran dengan membuka misi remedial (*foundational mission*) yang relevan dan memberikan umpan balik yang dipersonalisasi, memastikan bahwa setiap siswa mendapatkan intervensi yang sesuai dengan tingkat pemahamannya [19].

Secara keseluruhan, ITS memberikan landasan yang kuat bagi inovasi pendidikan berbasis personalisasi. Integrasi AI dengan gamifikasi membuka peluang besar untuk meningkatkan keterlibatan siswa, meskipun tantangan terkait privasi dan perbedaan respon individu tetap harus diperhatikan [33].

## 2.5 Gemini AI

Gemini AI merupakan model kecerdasan buatan generatif multimodal (teks dan kode) dari Google yang dirancang untuk konteks pendidikan dan *tutoring* [37]. Dalam konteks *Sistem Pembelajaran Cerdas*, Gemini AI mampu memberikan penjelasan yang kontekstual dan adaptif berdasarkan tingkat pemahaman siswa [38], serta menganalisis kode pemrograman untuk memberikan umpan balik yang spesifik dan konstruktif.

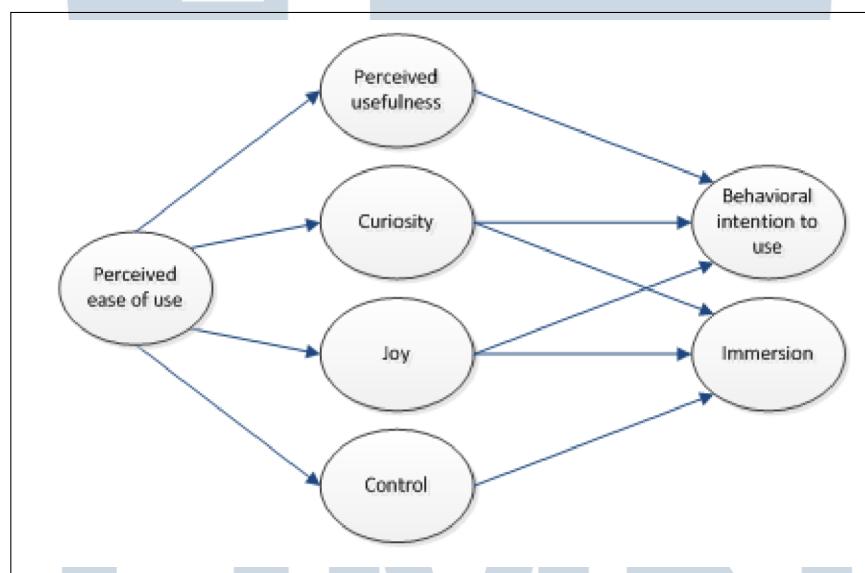
Implementasi Gemini AI mendukung pembelajaran *real-time* yang responsif dengan kemampuan mengidentifikasi kesalahan konseptual siswa dan memberikan panduan korektif yang tepat sasaran [32]. Dari perspektif teknis, API Gemini mudah diintegrasikan dengan *framework* web modern seperti Next.js dan mendukung *prompt engineering* yang kompleks. Hal ini memungkinkan pengembang merancang interaksi AI yang sesuai konteks (seperti mode tutor atau *co-pilot*), menjadikannya pilihan yang tepat untuk mengimplementasikan AI Companion dalam *Sistem Pembelajaran Cerdas yang gamified* [39]. Implementasi ini membedakan peran M.E.C.H.A sebagai Agen AI (*AI Agent*) dan bukan sekadar *chatbot* konvensional. Sebuah *chatbot* umumnya bekerja secara reaktif, hanya memproses input teks menjadi output teks berdasarkan pola statistik tanpa memiliki tujuan instruksional yang spesifik. Sebaliknya, Agen AI dalam sistem ini dirancang dengan kesadaran konteks (*context-awareness*) terhadap progres misi, riwayat kesalahan kode, dan tujuan pembelajaran siswa. Agen ini bertindak secara proaktif untuk mencapai tujuan pedagogis, yaitu memberikan perancah (*scaffolding*) berupa petunjuk bertahap dan bukan jawaban langsung, sehingga mendorong kemandirian berpikir siswa dalam memecahkan masalah pemrograman [21].

## 2.6 Hedonic Motivation System Adoption Model

Hedonic Motivation System Adoption Model (HMSAM) merupakan model yang digunakan untuk mengukur/mengevaluasi motivasi intrinsik pengguna dari penggunaan suatu sistem berbasis motivasi hedonik [40]. Dalam model HMSAM,

nilai *Behavioral Intention to Use* (BIU) dipengaruhi oleh kombinasi faktor kognitif dan hedonik. *Perceived Usefulness* (PU) dan *Perceived Ease of Use* (PEOU) mendorong niat menggunakan sistem karena pengguna merasa sistem bermanfaat dan mudah digunakan. Sementara itu, faktor hedonik seperti *Joy* dan *Curiosity* juga berpengaruh positif, di mana perasaan senang dan rasa ingin tahu terhadap sistem meningkatkan niat untuk terus menggunakan aplikasi [40].

Sedangkan *Focused Immersion* (FI) dipengaruhi oleh *Joy*, *Curiosity*, dan *Control* yang mendorong keterlibatan mendalam selama penggunaan sistem. FI juga dapat memperkuat BIU, karena pengalaman yang imersif meningkatkan kepuasan dan keinginan untuk menggunakan sistem secara berkelanjutan [40].



Gambar 2.11. Hedonic Motivation System Adoption Model

Sumber: [40]

Gambar 2.11 menunjukkan komponen-komponen yang digunakan pada Hedonic Motivation System Adoption Model (HMSAM). Berikut penjelasan dari masing-masing komponen tersebut [40]:

1. *Perceived ease of use* (PEOU), mengukur kemudahan penggunaan sistem.
2. *Perceived usefulness* (PU), mengukur tingkat kinerja pengguna ketika menggunakan sistem.
3. *Curiosity* (CUR), mengukur rasa penasaran atau rasa ingin tahu pengguna ketika menggunakan sistem.

4. *Joy* (JOY), mengukur tingkat kesenangan pengguna ketika menggunakan sistem.
5. *Control* (CTL), mengukur persepsi pengguna berdasarkan kendali/kontrol yang dimiliki terhadap sebuah sistem.
6. *Behavioral intention to use* (BIU), mengukur keinginan pengguna untuk menggunakan sebuah sistem.
7. *Focused immersion* (FI), mengukur tingkat fokus pengguna ketika menggunakan sistem.

## 2.7 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur sikap, pendapat, atau persepsi seseorang terhadap suatu subjek. Skala likert diambil dari nama penciptanya, yaitu Rensis Likert [41]. Skala Likert biasa digunakan dalam survei atau kuesioner dengan serangkaian pertanyaan yang dijawab melalui respons dalam bentuk skala, seperti 'sangat setuju' hingga 'sangat tidak setuju'. Respons dari pertanyaan yang diberikan dapat dijawab melalui lima pilihan jawaban pada skala 1 sampai 5 [42].

1. Sangat Setuju (SS) memiliki nilai 5.
2. Setuju (S) memiliki nilai 4.
3. Ragu-ragu (RR) memiliki nilai 3.
4. Tidak Setuju (TS) memiliki nilai 2.
5. Sangat Tidak Setuju (STS) memiliki nilai 1.

Persentase skor dapat dihitung menggunakan Rumus 2.1 dan Rumus 2.2. Rumus 2.2 digunakan untuk pertanyaan dengan skala nilai terbalik [42].

$$PS = \frac{(SS \times 5) + (S \times 4) + (RR \times 3) + (TS \times 2) + (STS \times 1)}{5 \times \text{Jumlah Responden}} \times 100 \quad (2.1)$$

$$PS = \frac{(STS \times 5) + (TS \times 4) + (RR \times 3) + (S \times 2) + (SS \times 1)}{5 \times \text{Jumlah Responden}} \times 100 \quad (2.2)$$

Keterangan:

*PS* = Persentase skor

*SS* = Sangat Setuju

*S* = Setuju

*RR* = Ragu-ragu

*TS* = Tidak Setuju

*STS* = Sangat Tidak Setuju

