

BAB 2

LANDASAN TEORI

2.1 Manajemen Proyek *Agile* dan Metodologi Scrum

Manajemen proyek perangkat lunak telah mengalami pergeseran paradigma dari pendekatan tradisional yang linier menuju metodologi *Agile* yang lebih adaptif. Secara konseptual, *Agile Project Management* (APM) difokuskan pada fleksibilitas, kecepatan, adaptabilitas terhadap perubahan, dan inovasi yang berkelanjutan [4]. Berbeda dengan manajemen proyek tradisional di mana manajer mengarahkan tim secara kaku, pendekatan *Agile* menekankan pada tim lintas fungsi yang mampu mengatur dirinya sendiri (*self-organizing*), siklus umpan balik yang berkelanjutan, serta kolaborasi erat untuk menyelesaikan pekerjaan yang kompleks [4]. Munculnya otonomi tim (*team autonomy*) ini sangat dipengaruhi oleh gaya kepemimpinan dan struktur organisasi, dan terbukti berkorelasi dengan keberhasilan proyek *Agile* [6].

Salah satu kerangka kerja (*framework*) *Agile* yang paling banyak diadopsi dalam industri perangkat lunak adalah Scrum. Scrum merupakan metode manajemen proyek yang ringan (*lightweight*) yang membagi proses pengembangan ke dalam iterasi-iterasi pendek yang disebut *Sprint*, biasanya berdurasi satu hingga empat minggu [1]. Dalam metodologi ini, terdapat tiga peran utama: Product Owner yang mengelola prioritas kebutuhan produk (*product backlog*), *Scrum Master* yang memastikan prinsip-prinsip Scrum berjalan dengan baik tanpa hambatan, dan *Development Team* yang mengeksekusi tugas-tugas teknis [1][7].

Gambar 2.1 memperlihatkan hubungan antara peran, artefak, dan kegiatan utama Scrum dalam satu siklus *sprint*. Alur tersebut menempatkan *product backlog* sebagai sumber pekerjaan, *sprint planning* sebagai tahap pemilihan pekerjaan, serta *daily scrum*, *sprint review*, dan *retrospective* sebagai mekanisme koordinasi dan umpan balik.



Gambar 2.1. Kerangka Kerja Metodologi Scrum

Sumber: [8]

Meskipun *Agile* dan Scrum menawarkan banyak keunggulan, penerapannya di lapangan sering kali menemui kendala terkait faktor sumber daya manusia. Ketergantungan yang tinggi pada tim yang mandiri dan kecepatan (*velocity*) *Sprint* yang konstan dapat memicu kelelahan mental (*burnout*) serta menurunkan kesejahteraan (*well-being*) pengembang [1][9]. Keberhasilan proyek perangkat lunak menggunakan *Agile* tidak hanya bergantung pada kecanggihan teknologi, melainkan sangat dipengaruhi oleh kapabilitas tim (*team capability*), komunikasi interpersonal, serta keterlibatan aktif pengembang (*developer engagement*) [4][10]. Penerapan praktik manajemen proyek *Agile* yang baik bahkan terbukti meningkatkan produktivitas, adaptabilitas, serta *time-to-market* tim pengembang perangkat lunak [11].

2.2 Prokrastinasi dalam Lingkungan Kerja

Prokrastinasi didefinisikan sebagai tindakan menunda tugas penting secara sukarela meskipun individu tersebut menyadari bahwa penundaan tersebut akan membawa konsekuensi negatif di masa depan [3]. Tingkat prevalensi prokrastinasi menunjukkan angka yang signifikan, di mana sekitar 15% hingga 20% populasi orang dewasa secara umum melakukan prokrastinasi, dan persentase melonjak hingga 40%-50% di lingkungan pra-profesional maupun akademis [3]. Di tempat kerja, perilaku ini telah menjadi salah satu sumber utama stres terkait pekerjaan yang dapat melumpuhkan produktivitas [3].

Fenomena penundaan tugas ini didorong oleh berbagai pemicu, baik dari karakteristik tugas maupun dari kerentanan individu. Secara umum, tugas yang dipersepsikan terlalu sulit, membosankan, atau memicu kecemasan akan mendorong individu untuk menghindarinya dan mencari aktivitas lain yang

memberikan gratifikasi instan [3]. Selain itu, gangguan eksternal atau "pencuri waktu" (*time thieves*) seperti notifikasi media sosial sering kali menjadi pemicu awal yang dengan mudah mengalihkan fokus pengembang perangkat lunak dari tugas utama mereka [3]. Lebih lanjut, rendahnya kemampuan regulasi diri (*self-regulation*) terbukti menjadi salah satu prediktor signifikan munculnya perilaku prokrastinasi [12].

Dampak dari prokrastinasi yang tidak tertangani memiliki efek domino yang merugikan baik bagi individu maupun siklus manajemen proyek. Dari segi psikologis, prokrastinasi dikaitkan dengan peningkatan kecemasan, penurunan harga diri (*self-esteem*), rasa bersalah, hingga potensi depresi [3]. Dari segi operasional, kebiasaan menunda menyebabkan tenggat waktu yang terlewat (*missed deadlines*), pekerjaan yang diselesaikan dengan terburu-buru, dan pada akhirnya menghasilkan penurunan kualitas dari perangkat lunak yang dikembangkan [3]. Untuk menanggulangnya, berbagai intervensi digital telah dikembangkan; sebagai contoh, intervensi berbasis internet dan seluler yang dioptimalkan dengan *persuasive design* terbukti efektif menurunkan tingkat prokrastinasi, bahkan setara dengan pendampingan manusia secara langsung [13].

2.3 Konsep Gamifikasi dalam Pengembangan Perangkat Lunak

Gamifikasi merupakan proses pengintegrasian elemen-elemen dan mekanisme desain permainan ke dalam konteks non-permainan dengan tujuan untuk memengaruhi perilaku, meningkatkan motivasi, dan mendorong keterlibatan (*engagement*) [1][10][14]. Ditinjau dari perspektif *Self-Determination Theory*, efektivitas gamifikasi dalam menumbuhkan motivasi intrinsik berakar pada kemampuannya memenuhi kebutuhan psikologis dasar pengguna, yaitu kompetensi, otonomi, dan keterhubungan [5][15]. Dalam domain pendidikan maupun profesional, gamifikasi umumnya terbagi menjadi dua pendekatan: *Structural Gamification* (gamifikasi struktural) dan *Content Gamification* (gamifikasi konten) [7]. Gamifikasi struktural menerapkan elemen seperti lencana dan poin di sekeliling materi tanpa mengubah inti tugasnya, sedangkan gamifikasi konten mengubah seluruh substansi tugas menjadi layaknya sebuah simulasi atau permainan penuh [7][16].

Implementasi gamifikasi pada lingkungan rekayasa perangkat lunak mengandalkan sejumlah elemen mekanik tertentu. Berdasarkan tinjauan literatur sistematis pada pengembangan perangkat lunak, *Point* (Poin) merupakan elemen

gamifikasi yang paling banyak digunakan, diikuti secara berurutan oleh Level (Tingkat), *Badge* (Lencana), Tantangan (*Challenges*), dan Papan Peringkat (*Leaderboards*) [14]. Elemen seperti poin dan lencana bertindak sebagai motivator ekstrinsik yang dapat meningkatkan kepuasan karyawan dengan memberikan umpan balik instan atas pencapaian tugas [17]. Sejumlah studi empiris menunjukkan bahwa elemen seperti lencana dan papan peringkat mampu meningkatkan keterlibatan dan motivasi pengguna, meskipun besar efeknya bervariasi bergantung pada konteks dan intensitas penerapan gamifikasi [18]. Di sisi lain, representasi visual progres seperti *Progress Bar* terbukti efektif dalam memitigasi prokrastinasi karena memberikan rasa kontrol terhadap waktu yang selaras dengan prinsip motivasi intrinsik [3].

Kendati menjanjikan banyak manfaat, penerapan gamifikasi yang serampangan atau "gamifikasi dangkal" (*shallow gamification*) dapat memicu efek yang kontraproduktif. Penggunaan papan peringkat yang murni kompetitif dapat merusak dinamika kerja sama tim jika tidak dirancang dengan hati-hati [3]. Studi komparatif menunjukkan bahwa papan peringkat cenderung menciptakan lingkungan yang lebih kompetitif, sedangkan lencana lebih kecil kemungkinannya menimbulkan tekanan internal pada pengguna [19]. Risiko lain yang kerap muncul meliputi stres akibat tekanan kinerja, kebosanan yang cepat saat efek kebaruan (*novelty effect*) memudar, serta hilangnya motivasi intrinsik jangka panjang [1]. Oleh karena itu, perancangan gamifikasi harus berpijak pada metodologi dan kerangka kerja yang solid.

Lebih jauh, penerapan gamifikasi dalam konteks profesional tidak hanya bertujuan untuk mengubah perilaku penyelesaian tugas secara mekanis, tetapi juga berfungsi sebagai katalisator utama untuk inovasi proses bisnis [20]. Tinjauan literatur sistematis menegaskan bahwa elemen-elemen permainan secara terbukti mampu meningkatkan motivasi intrinsik dan keterlibatan (*engagement*) karyawan secara signifikan [20]. Peningkatan motivasi intrinsik ini merupakan fondasi yang krusial bagi terciptanya budaya inovasi, eksperimentasi, dan kolaborasi di tempat kerja [20]. Melalui elemen gamifikasi seperti sistem penghargaan dan umpan balik instan, lingkungan kerja yang dinamis dapat diciptakan untuk mendorong pemecahan masalah yang lebih kreatif dan komitmen yang lebih tinggi terhadap tujuan organisasi [20]. Meskipun demikian, keberhasilan strategi gamifikasi ini tidak bersifat universal; implementasinya sangat bergantung pada penyesuaian (kustomisasi) desain gamifikasi dengan budaya organisasi guna meminimalkan risiko hambatan struktural, seperti penolakan terhadap perubahan (*resistance to*

change) maupun perbedaan pandangan antargenerasi [20].

2.4 Framework TechMark

Framework TechMark merupakan kerangka kerja komprehensif yang dirancang secara spesifik untuk merancang, mengimplementasikan, dan mengevaluasi solusi gamifikasi bagi tim perangkat lunak di dalam organisasi TI [2]. Kerangka kerja ini dibangun dengan mengadopsi prinsip *Flow Theory*, yang bertujuan menciptakan keseimbangan antara tantangan pekerjaan dan kemampuan individu, sehingga pengguna mencapai kondisi fokus mendalam, memiliki kendali penuh atas pekerjaan, serta termotivasi secara intrinsik tanpa menyadari berlalunya waktu [2].

Arsitektur kerangka kerja TechMark terdiri dari empat tingkatan (*tier*) utama yang sistematis. Tingkat pertama, *Goal Identification* (Identifikasi Tujuan), difokuskan pada penentuan masalah spesifik seperti rendahnya moral tim atau kendala manajemen tugas, serta menetapkan target audiens pengembang [2]. Tingkat kedua, *Game Design* (Desain Permainan), adalah fase merancang mekanisme permainan, aturan, elemen insentif (poin/lencana), dan memetakan perjalanan interaksi pengguna [2]. Tingkat ketiga berpusat pada *UI/UX Design*, yang bertujuan merancang antarmuka ramah pengguna serta sistem umpan balik yang visual dan interaktif. Tingkat keempat adalah *Integration & Iteration*, di mana sistem secara iteratif disempurnakan berdasarkan umpan balik pengguna di lingkungan kerja nyata [2].

Pemilihan TechMark pada pengembangan Agilify didasarkan pada kesesuaian konteks dan cakupan prosesnya. Kerangka ini secara spesifik ditujukan bagi tim perangkat lunak dalam organisasi TI dan menghubungkan identifikasi tujuan, perancangan mekanisme permainan, desain UI/UX, serta integrasi dan iterasi dalam satu alur [2]. Alur tersebut memungkinkan setiap fitur Agilify ditelusuri kembali pada masalah dan kebutuhan pengguna, kemudian diperiksa melalui evaluasi purwarupa dan sistem. Keterlacakan ini membantu mengurangi risiko "gamifikasi dangkal", tetapi tidak menjamin bahwa motivasi, produktivitas, atau perilaku pengguna akan berubah. Dampak tersebut tetap memerlukan pengukuran empiris terhadap pengguna nyata.

2.5 Formula Perhitungan Experience Points (XP)

Dalam konteks portal Agilify, mekanisme pemberian *Experience Points* (XP) dirancang untuk mencerminkan proporsi kontribusi nyata setiap pengembang terhadap penyelesaian tugas dalam sebuah *sprint*. Besaran XP dasar yang diperoleh ditentukan berdasarkan tingkat kesulitan (*difficulty*) dari masing-masing tugas, bukan berdasarkan kuantitas atau bobot estimasi (*story points*), sehingga mendorong penyelesaian tugas yang secara substansi lebih menantang. Penetapan nilai XP dasar per tingkat kesulitan adalah sebagai berikut: tugas dengan tingkat kesulitan *Easy* bernilai 10 XP, tingkat kesulitan *Medium* bernilai 20 XP, dan tingkat kesulitan *High* bernilai 40 XP.

Sebagai insentif bagi penyelesaian tugas yang memiliki target waktu, sistem menerapkan mekanisme *deadline bonus*. Dalam implementasi saat ini, bonus diberikan kepada setiap tugas berstatus selesai yang memiliki tanggal tenggat (*due date*) ketika *sprint* diselesaikan. Sistem belum membandingkan waktu penyelesaian aktual dengan tanggal tenggat. Tugas yang memenuhi kondisi tersebut memperoleh pengganda XP sebesar 1,25 (25%) dari nilai XP dasarnya, sehingga formula XP akhir per tugas menjadi:

$$XP_{akhir} = XP_{dasar} \times 1,25 \quad (\text{jika tugas memiliki dueDate}) \quad (2.1)$$

Adapun untuk menerjemahkan akumulasi *lifetime* XP seorang pengguna menjadi sebuah *level* yang bermakna, sistem mengadopsi formula eksponensial yang memastikan tantangan terus meningkat seiring pertumbuhan kemampuan pengguna, selaras dengan prinsip *Flow Theory* yang mendasari Framework TechMark. Formula perhitungan level ditetapkan sebagai berikut.

$$\text{Level} = \left\lceil \left(\frac{XP_{lifetime}}{100} \right)^{\frac{2}{3}} \right\rceil + 1 \quad (2.2)$$

Formula ini memastikan bahwa semakin tinggi level pengguna, semakin besar akumulasi XP yang dibutuhkan untuk naik ke level berikutnya, sehingga perkembangan terasa progresif dan tidak trivial. Untuk mengetahui total XP minimum yang dibutuhkan agar seorang pengguna dapat berada pada suatu level tertentu, digunakan formula invers berikut.

$$XP_{untuk\ Level} = \left\lfloor 100 \times (Level - 1)^{1,5} \right\rfloor \quad (2.3)$$

Ketiga formula tersebut diterapkan pada bagian sistem yang berbeda. Nilai XP dasar 10, 20, atau 40 ditetapkan dan disimpan sebagai `xpReward` ketika tugas dibuat atau ketika tingkat kesulitannya diubah. Persamaan 2.1 diterapkan oleh `endpoint` POST `/api/teams/complete-sprint` ketika Project Manager mengonfirmasi penyelesaian *sprint*; perhitungan hanya dilakukan terhadap tugas berstatus `DONE`. Setelah XP tugas, bonus *epic*, dan pengali avatar diakumulasikan ke `lifetimeXp`, Persamaan 2.2 digunakan untuk memperbarui nilai `level` pada statistik pengguna. Persamaan 2.3 digunakan untuk menentukan ambang XP level berjalan dan level berikutnya pada indikator progres *dashboard*, serta ambang XP untuk membuka tingkatan rencana pada halaman *Hall of Ranks*. Alur penerapan saat penyelesaian *sprint* dijelaskan lebih lanjut pada Bab 3 dan ditunjukkan pada hasil implementasi Gambar 4.13.

Di samping XP dari penyelesaian tugas, sistem juga memberikan bonus XP tambahan melalui mekanisme *Epic Boss Defeat*. Ketika seluruh sub-tugas dari sebuah *epic* berhasil diselesaikan, setiap anggota tim yang berkontribusi memperoleh bonus sebesar 100 XP untuk kekalahan pertama (*first defeat*) atau 20 XP untuk kekalahan berikutnya. Seluruh XP yang diperoleh selanjutnya dikalikan dengan pengali avatar aktif pengguna (*avatar multiplier*) sebelum ditambahkan ke total XP *lifetime* dan XP yang dapat dibelanjakan (*spendable XP*).

2.6 Heuristic Evaluation for Gameful Design

Heuristic evaluation merupakan metode inspeksi kegunaan (*usability inspection*) di mana sejumlah evaluator ahli menilai sebuah sistem terhadap seperangkat prinsip atau pedoman desain untuk mengidentifikasi masalah desain. Karena bersifat cepat dan tidak mahal, metode ini umum digunakan pada tahap awal maupun akhir pengembangan sebagai pelengkap pengujian pengguna.

Untuk konteks sistem bergamifikasi, Tondello et al. (2016) mengusulkan *Heuristic Evaluation for Gameful Design* [21], yaitu seperangkat 28 heuristik yang diturunkan dari dimensi-dimensi afordansi motivasi (*motivational affordances*) yang paling umum pada berbagai kerangka kerja *gameful design*. Efektivitas masing-masing elemen desain permainan tersebut bahkan dapat diuji secara empiris untuk mengetahui elemen mana yang paling berpengaruh terhadap motivasi dan

keterlibatan pengguna [22]. Heuristik tersebut dikelompokkan menjadi tiga kategori berdasarkan teori motivasi intrinsik dan ekstrinsik (*Self Determination Theory*) [5].

1. Heuristik motivasi intrinsik (16 heuristik, mencakup dimensi *purpose & meaning, challenge & competence, completeness & mastery, autonomy & creativity, relatedness*, dan *immersion*);
2. Heuristik motivasi ekstrinsik (5 heuristik, mencakup *ownership & rewards, scarcity*, dan *loss avoidance*);
3. Heuristik *context dependent* (7 heuristik, mencakup *feedback, unpredictability*, dan *change & disruption*)

Pendekatan ini memungkinkan evaluator menilai secara cepat apakah suatu sistem telah mengimplementasikan keseimbangan afordansi motivasi yang sesuai, sekaligus menyadari bahwa tidak semua afordansi wajib hadir pada setiap sistem.

2.7 Inter-Rater Reliability (IRR)

Inter-Rater Reliability (IRR) secara umum digunakan untuk menilai reliabilitas penilaian antara dua atau lebih penilai yang mengevaluasi objek yang sama secara independen [23]. Namun, perhitungan dalam penelitian ini secara khusus mengikuti *percentage acceptance agreement* yang digunakan pada Framework TechMark [2]. Untuk tiga evaluator dengan penilaian biner A_i , B_i , dan C_i pada butir ke- i , nilai setiap pasangan dihitung melalui perkalian biner:

$$AB_i = A_i B_i, \quad AC_i = A_i C_i, \quad BC_i = B_i C_i. \quad (2.4)$$

Dengan operasi tersebut, hanya pasangan yang sama-sama memberikan nilai 1 (1×1) yang menghasilkan nilai 1. Pasangan 1×0 , 0×1 , dan 0×0 menghasilkan nilai 0. Persentase penerimaan pasangan dihitung menggunakan rumus:

$$IRR_{\text{penerimaan}} = \frac{\sum_{i=1}^H (A_i B_i + A_i C_i + B_i C_i)}{3H} \times 100\%, \quad (2.5)$$

dengan H adalah jumlah butir yang dinilai. Oleh karena pasangan 0–0 tidak dihitung, nilai ini tidak menunjukkan persentase kesamaan seluruh keputusan evaluator. Nilai tersebut menunjukkan proporsi kesempatan ketika dua evaluator

dalam suatu pasangan secara bersama-sama menyatakan bahwa komponen telah terpenuhi. Ambang 75% mengikuti kriteria operasional pada TechMark untuk menentukan kecukupan penerimaan desain dalam penelitian ini. Nilai ini bukan nilai signifikansi statistik, bukan ukuran besar pengaruh gamifikasi, dan tidak membuktikan perubahan perilaku pengguna. Selain tidak mengoreksi peluang kesepakatan kebetulan, ukuran ini lebih sempit daripada *percentage agreement* umum karena tidak memasukkan penolakan bersama (0–0). Oleh sebab itu, hasil diinterpretasikan secara deskriptif sebagai persentase penerimaan pasangan dan harus dibaca bersama distribusi nilai tiap butir.

Evaluasi heuristik idealnya dilakukan secara independen oleh lebih dari satu evaluator. Rekomendasi praktis menempatkan tiga sampai lima evaluator sebagai rentang yang memadai untuk inspeksi heuristik [24]. Penggunaan tiga evaluator dalam penelitian ini berada pada batas bawah rentang tersebut dan menghasilkan tiga produk penerimaan pasangan pada setiap butir, yaitu AB, AC, dan BC. Jumlah ini memadai untuk evaluasi desain berskala skripsi, tetapi tidak dimaksudkan sebagai sampel inferensial untuk generalisasi populasi.

2.8 Tinjauan Penelitian Terdahulu

Sebagai landasan untuk menemukan celah penelitian (*research gap*) dan menegaskan kebaruan (*novelty*) dari sistem yang dibangun, berikut diuraikan tinjauan terhadap beberapa studi terdahulu yang paling relevan. Kajian literatur ini meringkas temuan dari sejumlah penelitian utama terkait adopsi gamifikasi pada manajemen tugas, prokrastinasi, dan lingkungan rekayasa perangkat lunak sebagaimana dibandingkan pada Tabel 2.1.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu

No.	Penulis & Tahun	Judul	Temuan Utama	Gap
1	Obaid & Farooq (2024)	TechMark A framework for the development, engagement, and motivation of software teams in IT organizations based on gamification	Mengusulkan framework TechMark (Goal, Game Design, UI/UX, Integration) dan portal kerja (GWP). Terbukti meningkatkan penyelesaian tugas dan kualitas kode.	Studi ini berfokus umum pada Internal Marketing di organisasi IT dan belum secara spesifik mengintegrasikan framework ini secara mendalam ke dalam siklus harian metodologi Agile/Scrum.
2	Elsalmy et al. (2025)	Agile Gamification Risk Management Process A Comprehensive Process for Identifying and Assessing Gamification Risks	Mengidentifikasi risiko "gamifikasi dangkal" (stres, kompetisi toksik) dan menawarkan proses manajemen risiko (AGRM).	Fokus penelitian adalah pada proses manajemen risiko secara manajerial, bukan pada pembangunan tools manajemen proyek yang memiliki fitur mitigasi risiko secara teknis/otomatis.

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu (Lanjutan)

3	Stol et al. (2021)	Gamification in software engineering the mediating role of developer engagement and job satisfaction	Menemukan korelasi positif antara partisipasi gamifikasi dengan keterlibatan pengembang (developer engagement) dan kepuasan kerja.	Penelitian ini merupakan studi korelasional (survei) yang membuktikan hubungan antar variabel, tetapi tidak menghasilkan artefak perangkat lunak (software artifact).
4	Priyadi et al. (2023)	Gamification in Software Development Systematic Literature Review	Elemen Poin, Level, dan Badge paling banyak digunakan. Banyak studi tidak menggunakan framework baku (membangun dari nol).	Ditemukan bahwa mayoritas pengembangan gamifikasi dilakukan secara ad-hoc tanpa dasar framework desain yang jelas, sehingga rentan gagal.
5	Fatima et al. (2024)	Exploring the use of gamification in human-centered agile-based requirements engineering	Mengembangkan aplikasi "ReGile" (simulasi 3D). Efektif meningkatkan performa dalam fase requirements engineering.	Fokus hanya pada satu fase (Requirements) dan menggunakan pendekatan Content Gamification (simulasi 3D) yang berat untuk penggunaan manajemen proyek harian.

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu (Lanjutan)

6	Koudriachov et al. (2025)	Success with Agile Project Management Looking back and into the future	Gamifikasi berpengaruh positif terhadap aspek teknologi tim, namun dampaknya pada kapabilitas tim bergantung pada konteks dan faktor pendukung lain.	Menunjukkan bahwa gamifikasi sering kali tidak efektif jika berdiri sendiri tanpa integrasi yang baik dengan peningkatan kemampuan tim (team capability).
7	Pereira et al. (2025)	Investigating Gamification to Reduce Procrastination - Systematic Literature Review	Elemen progress bar dan rewards efektif mengurangi penundaan tugas (prokrastinasi), namun personalisasi sering diabaikan.	Kurangnya fitur personalisasi dalam sistem gamifikasi yang ada (pendekatan one-size-fits-all), yang dapat menurunkan motivasi jangka panjang.
8	Capatina et al. (2024)	Leveling up in corporate training Unveiling the power of gamification to enhance knowledge retention, knowledge sharing, and job performance	Elemen visual (leaderboard) paling menarik perhatian mata dan meningkatkan retensi pengetahuan serta kinerja.	Studi dilakukan dalam konteks pelatihan/training sesaat, bukan dalam konteks manajemen proyek jangka panjang yang membutuhkan keterlibatan berkelanjutan.

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu (Lanjutan)

9	Huang et al. (2023)	Gamified Project-Based Learning A Systematic Review of the Research Landscape	Gamifikasi efektif dalam pembelajaran berbasis proyek, terutama tipe Game-Making dan Structural Gamification.	Konteks utamanya adalah pendidikan (siswa), bukan profesional industri. Tantangan teknis sering menghambat efektivitas implementasi.
10	Campos Rosendo & Navas Gotopo (2026)	Gamification and its impact on improving training and business innovation processes	Gamifikasi meningkatkan motivasi intrinsik dan budaya inovasi, namun ada hambatan budaya organisasi.	Studi ini bersifat tinjauan literatur umum pada bisnis dan inovasi, kurang spesifik pada fitur teknis perangkat lunak manajemen proyek.
11	Mutter et al. (2023)	StudiCare Procrastination: an internet- and mobile-based intervention with digital coach targeting procrastination	Intervensi digital berbasis internet/seluler dengan <i>persuasive design</i> dan pendamping digital efektif menurunkan prokrastinasi, setara dengan pendampingan manusia.	Berbentuk program intervensi terapeutik (iCBT) yang berdiri sendiri, bukan terintegrasi ke dalam alat manajemen tugas <i>Agile</i> harian.

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu (Lanjutan)

12	Prieto Andreu & Moreno-Ger (2026)	Leaderboards or badges in two gamified forums: motivational and engagement differences	Papan peringkat menciptakan lingkungan yang lebih kompetitif, sedangkan lencana cenderung tidak menimbulkan tekanan internal pada pengguna.	Menyoroti risiko kompetisi papan peringkat pada konteks forum pembelajaran, tanpa menawarkan rancangan <i>tools</i> manajemen tugas yang menghindarinya.
13	Rahman et al. (2024)	Agile Project Management Impacts Software Development Team Productivity	Penerapan manajemen proyek <i>Agile</i> meningkatkan produktivitas, adaptabilitas, dan <i>time-to-market</i> tim pengembang.	Berfokus pada dampak praktik <i>Agile</i> secara manajerial tanpa elemen gamifikasi untuk mengatasi motivasi dan prokrastinasi.
14	Mazarakis & Bräuer (2023)	Gamification is Working, but Which One Exactly? Results from an Experiment with Four Game Design Elements	Eksperimen membuktikan bahwa elemen desain permainan tertentu lebih efektif dari yang lain dalam mendorong motivasi dan keterlibatan; tidak semua elemen berdampak sama.	Studi eksperimental menguji elemen game secara terisolasi, belum diintegrasikan ke dalam portal manajemen tugas <i>Agile</i> yang utuh.

Tabel 2.1 menunjukkan tiga celah yang menjadi dasar penelitian ini. Pertama, sebagian penelitian membahas manfaat atau risiko gamifikasi tanpa menghasilkan artefak portal tugas yang terintegrasi dengan alur Scrum. Kedua, penerapan elemen permainan masih berisiko dilakukan secara ad hoc dan memicu kompetisi yang tidak sehat. Ketiga, bukti dari konteks pendidikan, pelatihan, atau intervensi prokrastinasi tidak dapat langsung dipindahkan ke tim perangkat lunak. Oleh karena itu, penelitian ini menempatkan kebaruannya pada rancang bangun Agilify menggunakan TechMark dan evaluasi kesesuaian elemen gamifikasinya, bukan pada klaim bahwa dampak perilaku telah terbukti.

