

BAB II TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian menggunakan metode *agile* sudah sangat banyak digunakan dalam pengembangan suatu aplikasi termasuk pengembangan *game*. Ada berbagai macam tipe metode *agile* seperti *SCRUM*, *Kanban*, *XP*, dan banyak lainnya. Berikut merupakan penelitian terdahulu yang berkaitan dengan pokok bahasan.

Penelitian pertama yang berjudul “Revamp Aplikasi Teman Bumil Lebih Interaktif Dengan Pendekatan Agile” oleh Tofid, dkk, 2020, dimana aplikasi ini dirancang untuk membantu ibu hamil dengan memuat berbagai informasi mulai dari memantau tumbuh kembang sang bayi dari ibu hamil hingga menjalankan program hamil. Penelitian ini menggunakan metode pengembangan *Agile* dengan model kombinasi *Scrum* dan *Kanban* atau bisa disebut *Scrumban*. Hasil dari penelitian ini berupa aplikasi yang sudah diperbaharui dengan adanya sumber daya yang optimal, visualisasi alur kerja dengan memberi batasan pada Work in Progress (WIP), dan kepuasan *client* dengan hasil berupa “Baik”. Akan tetapi dalam penelitian ini terdapat beberapa kekurangan diantaranya perbandingan jumlah pengguna setelah menggunakan sistem yang baru, beserta desain *icon* yang masih membuat pengguna kurang *familiar* [19].

Penelitian kedua yang berjudul “Pengembangan *Game Visual Novel* Folklor Menggunakan Metode *Agile Kanban Board*”, oleh Jihan Aferiansyah, 2022, dimana aplikasi ini dirancang peneliti untuk membuat *game visual novel* menggunakan metode *kanban board*. Hasil dari penelitian ini berupa aplikasi *game visual novel* berbasis website berhasil dibangun sesuai dengan *user’s story* dengan *Kanban* yang dapat mengorganisir pembuatan dan menyesuaikan dengan perkembangan task yang ada dengan nilai “Baik”. Akan tetapi dalam penelitian ini terdapat beberapa kekurangan diantaranya perbaikan pada *User Interface* (UI) pada setiap fitur, pengembangan aplikasi

dalam *handphone* sehingga dapat dimainkan dimana saja dan kapan saja, membuat interaksi yang bukan hanya *text* namun *voice* [20].

Penelitian ketiga yang berjudul “Pengembangan Aplikasi Jual Beli Sampah Plastik Berbasis Android dan Web Menggunakan Kanban”, oleh Ardi Ragil Saputra, 2023, dimana aplikasi ini bertujuan untuk jual beli sampah daur ulang dengan mengintegrasikan web admin dan aplikasi android. Penelitian ini menggunakan metode *agile* dan *kanban*. Hasil dari penelitian ini mencapai nilai 76% atau bernilai “Baik”, namun aplikasi ini perlu dikembangkan untuk mendukung banyak *platform* [21].

Penelitian keempat yang berjudul “Implementasi Multimedia *Software Engineering Methodology* Pada Pembuatan Perangkat Promosi Wisata”, oleh Tyrza Adelia, 2023, dimana aplikasi ini menyatukan dua metodologi pengembangan produk multimedia interaktif dan non interaktif dalam pembuatan produk materi wisata untuk mempromosikan wisata kampung Warna Warni Greges dengan mengikuti *guidelinedan* tata kelola serta alur proses MSEM. Hasil yang didapatkan adalah 62.83% menjawab dengan benar, namun aplikasi ini memerlukan pembahasan yang lebih komprehensif untuk implementasi MSEM pada bidang multimedia yang lain seperti game, virtual tour, serta multimedia edukasi [22].

Lalu penelitian terakhir yang berjudul “Pengujian *BlackBox* dan Kuesioner Pada Game Gems Adventure”, oleh Rizal Dwi *et al.*, 2022, dimana aplikasi ini merupakan permainan edukasi dimana *user* akan diminta untuk menjawab soal disetiap *levelnya*, Hasil dari penelitian ini adalah *game* edukasi yang sudah diuji menggunakan *blackbox* dan mendapatkan hasil 88% atau bernilai “Sangat Baik”, namun aplikasi ini berbasis *online* dan tidak terdapat banyak varian tingkat *level*, serta *asset* yang digunakan masih belum sempurna [23].

Berdasarkan hasil penelitian di atas, peneliti dapat menyimpulkan bahwa kelebihan dalam *game* “Yuk Berhitung” dalam menggunakan kanban adalah sifatnya yang fleksibel dan progresnya dapat dilakukan tanpa runtutan yang begitu spesifik. Dengan 30 responden pengujian UAT menunjukkan hasil 91% atau masuk dalam kategori “Sangat Baik” hal ini menunjukkan

kelebihan penulis dibanding penelitian-penelitian terdahulu. Beserta game ini telah sepenuhnya mencapai keseluruhan dalam tahap *agile*, dikembangkan dalam perangkat yang *mobile*, memiliki *icon* yang *simple*, menggunakan suara sehingga lebih interaktif. Hasil penelitian terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No.	Judul Penelitian	Nama	Tahun	Metode	Hasil
1.	Revamp Aplikasi Teman Bumil Lebih Interaktif Dengan Pendekatan Agile	Tofid , Eddy Julianto , Yulius Harjoseputro	2021	Scrumban (Scrum + Kanban)	Hasil dari penelitian ini berupa aplikasi yang sudah di perbaharui dengan adanya sumber daya yang optimal, visualisasi alur kerja dengan memberi batasan pada Work in Progress (WIP), dan kepuasan client dengan hasil berupa “Baik”. Akan tetapi dalam penelitian ini terdapat beberapa kekurangan diantaranya perbandingan jumlah pengguna setelah menggunakan sistem yang baru, beserta desain <i>icon</i> yang masih membuat pengguna kurang <i>familiar</i>
2	Pengembangan <i>Game Visual Novel</i> Folklor Menggunakan Metode <i>Agile Kanban Board</i>	Jihan Aferiansyah	2022	Kanban	Hasil dari penelitian ini berupa aplikasi <i>game visual novel</i> berbasis website berhasil dibangun sesuai dengan <i>user’s story</i> dengan <i>Kanban</i> yang dapat mengorganisir pembuatan dan menyesuaikan dengan perkembangan task yang ada. Dengan hasil UAT “Baik”. Akan tetapi dalam penelitian ini terdapat beberapa kekurangan diantaranya perbaikan pada <i>User Interface</i> (UI) pada setiap fitur, Pengembangan aplikasi dalam <i>handphone</i> sehingga dapat dimainkan dimana saja dan kapan saja, Membuat interaksi yang bukan hanya <i>text</i> namun <i>voice</i>
3	Pengembangan Aplikasi Jual Beli Sampah Plastik Berbasis Android dan Web Menggunakan Kanban	Ardi Ragil Saputra	2023	Kanban	Hasil dari penelitian ini mencapai nilai 76% atau bernilai “Baik”, namun aplikasi ini perlu dikembangkan untuk mendukung banyak <i>platform</i>

No.	Judul Penelitian	Nama	Tahun	Metode	Hasil
4	Implementasi Multimedia Software Engineering Methodology Pada Pembuatan Perangkat Promosi Wisata	TyrzaAdelia, Andre, JoshiaEkiManuel	2023	Kanban	Hasil yang didapatkan adalah 62.83% menjawab dengan benar, namun aplikasi ini pembahasan yang lebih komprehensif untuk implementasi MSEM pada bidang multimediyang lain seperti game, virtual tour, serta multimedia edukasi
5	Pengujian <i>BlackBox</i> dan Kuesioner Pada Game Gems Adventure	Rizal Dwi <i>et al</i>	2022	BlackBox	Hasil dari penelitian ini adalah <i>game</i> edukasi yang sudah diuji menggunakan <i>blackbox</i> dan mendapatkan hasil 88% atau bernilai “baik” namun aplikasi ini berbasis <i>online</i> dan tidak terdapat banyak varian tingkat <i>level</i> , serta <i>asset</i> yang digunakan masih belum sempurna

2.2 Dasar Teori

2.2.1 Dasar Berhitung

Kemampuan berhitung merupakan kemampuan yang dimiliki anak dimulai dari lingkungan yang terdekat dengan dirinya, sejalan dengan perkembangannya anak dapat meningkat ke tahap pengertian mengenai penjumlahan dan pengurangan [24]. Menurut Permendiknas Nomor 58 Tahun 2003, bahwa hingga usia 6 tahun dapat menyebutkan lambang bilangan 1 hingga 10, Mencocokkan bilangan dengan lambang bilangan dan mengenal berbagai macam lambang huruf vokal dan konsonan [25].

2.2.2 *Game*

Game merupakan suatu aktivitas bermain [9] yang dapat dimainkan dengan aturan tertentu untuk mencapai suatu hasil, baik menang maupun kalah atau seri yang dapat dimainkan secara tidak serius maupun dengan tujuan kompetitif [6]. *Game* juga dapat membantu anak-anak dalam belajar melatih pola pikir, konsentrasi, penyelesaian masalah, dan beradaptasi terhadap perubahan yang ada [9]. *Game* edukasi merupakan permainan yang dirancang untuk pengayaan pendidikan yang mana mendukung pengajaran dan pembelajaran, menggunakan teknologi multimedia interaktif [26].

2.2.3 *Android*

Android merupakan sebuah sistem operasi yang banyak digunakan untuk perangkat *mobile* yang pengembangannya dipimpin oleh Google [27]. Sistem operasi ini sangat populer karena sistem yang *compactible* dengan berbagai merek perangkat *mobile* dan harganya yang terjangkau dibanding sistem operasi lainnya. *Android* juga begitu cepat dalam melakukan inovasi [18], hal ini menjadi alasan mengapa sebagian besar banyak didominasi oleh sistem operasi *Android*.

2.2.4 Metode Agile

Metode *Agile* merupakan salah satu metode dalam pengembangan software yang sangat efektif karena dari namanya sendiri yang berarti ringkas, cepat dan bebas bergerak serta menawarkan fleksibilitas [10]. *Agile* juga merupakan pendekatan pengembangan yang berfokus pada kecepatan selesainya suatu proyek dan memungkinkan terjadinya perubahan sesuai kondisi yang ada [14]. Hal ini mengacu kepada sifat metode *agile* yang fleksibel. Berikut tahapan dalam penerapan metode *agile* yaitu sebagai berikut:

- *Planning*

Pada tahap ini terjadi proses pengumpulan kebutuhan perancangan sistem [28], melakukan pengumpulan data [10], dan menentukan perencanaan terhadap *game* yang akan dibuat.

- *Design*

Pada tahap ini dilakukannya pembuatan *asset-asset* beserta *prototype* sehingga ada gambaran bagaimana *game* yang nantinya dibuat [10].

- *Coding / Implementation*

Pada tahap ini hasil desain yang ada akan digabungkan menjadi nyata oleh bantuan compiler, *game engine* dan aplikasi *editing* lainnya sehingga *game* dapat dimainkan. Beserta dilakukan penyesuaian sistem sehingga meminimalisir adanya *bug* dalam program [28].

- *Testing*

Tahapan dimana produk yang sudah jadi akan dilakukan pengecekan oleh *game tester* yang bertugas mencari *bug in game*, kecacatan dan kemungkinan error. Sehingga dapat diketahui apakah *game* ini layak atau tidak [10].

2.2.5 Kanban

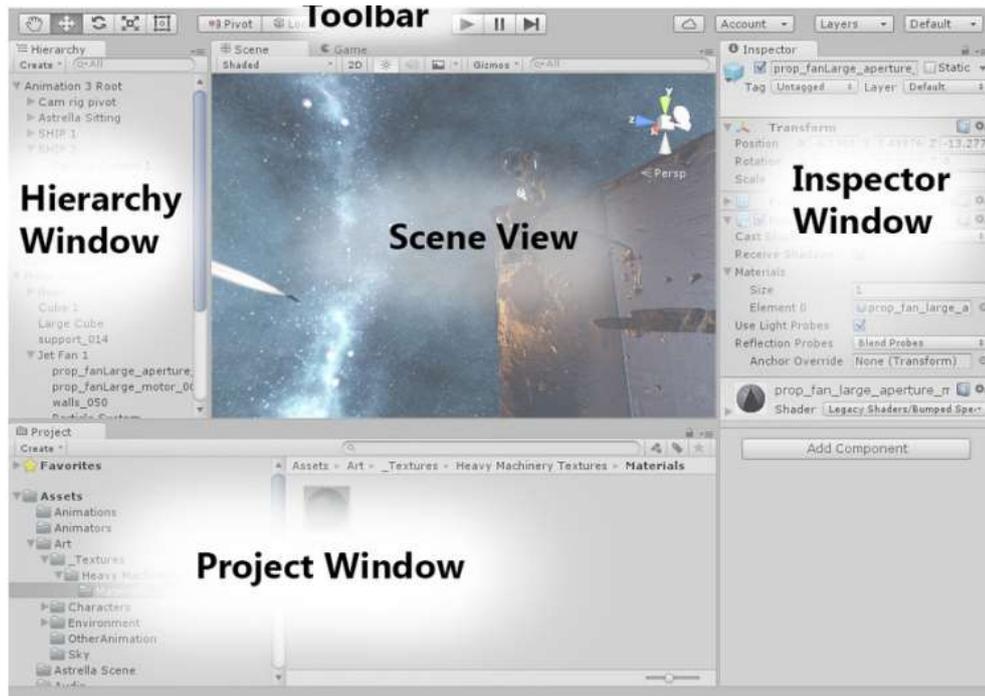
Kanban merupakan metode untuk secara bertahap meningkatkan *task*. Kanban muncul menggunakan rasa "kartu visual," "papan nama", atau "BillBoard" untuk menunjukkan alur kerja *Work In Progress* (WIP). Tugas atau *task* direpresentasikan pada kartu tersebut. Status tiap tugas dikenal dengan menampilkan kartu pada kolom-kolom terpisah yang terdapat pada papan kanban [20]. Adapun konsep yang dimiliki oleh metode Kanban ini meliputi:

- Visualisasikan Alur Kerja
- Batasi WIP (*Work In Progress*)
- Mengukur *Lead Time*.

2.2.6 Unity Engine

Unity Engine adalah sebuah alat pengembangan untuk menghasilkan konten interaktif seperti *game*, animasi, hingga visualisasi arsitektur yang dapat bergerak dalam berbagai sistem operasi dan menghasilkan konten tersebut ke berbagai bentuk seperti *mobile based* (*Android/ios*), *desktop/PC*, bahkan *Web based game* [16]. *Unity* merupakan aplikasi *game engine* yang berguna sebagai alat untuk mengembangkan *game* dengan berbagai macam *platform* yang mudah digunakan [9]. *Unity* sangat sederhana, padat dan lengkap untuk mengembangkan *game*. Bahkan bukan hanya *game*, namun dapat juga membuat desain 3d/2d dan untuk keperluan *cinematic* memanfaatkan keunggulan kamera dari aplikasi ini. *Unity* sangat lengkap dan memiliki komunitas yang luas sebagai penghasil *asset-asset* untuk membuat suatu aplikasi *game*. *Unity* sudah terintegrasi dengan *Visual Studio Code* sebagai compiler dasarnya yang menggunakan bahasa C# (*C Sharp*), Bahkan dapat menggunakan asset tertentu sehingga tidak perlu melakukan coding sama sekali, namun perlu untuk merangkai kejadian kejadian tersebut dalam sebuah rangkaian pohon. Terdapat banyak fungsi

dalam *unity engine* yang mendukung pengembang seperti *shaders*, *physics engine*, *network*, *terrain manipulation*, *audio*, *video*, *animation* dan masih banyak fitur lain yang membuat proses pengembangan mudah [29]. Contoh tampilan *Unity Engine* dapat dilihat pada Gambar dibawah ini:



Gambar 2.1 Tampilan keseluruhan Unity Engine [30]

(Sumber: Unity Technology, 2018)

Dari tampilan tersebut terdapat 5 window utama, yaitu window *Toolbar*, *Scene*, *Project*, *Hierarchy*, *Inspector* dan *Game*. Secara singkat, berikut merupakan penjelasan dari masing-masing window :

- *Toolbar*: *Toolbar* merupakan alat yang membantu mengatur komponen-komponen atau objek di dalam *scene* [30]. *Toolbar* dapat dilihat pada Gambar 2.2.

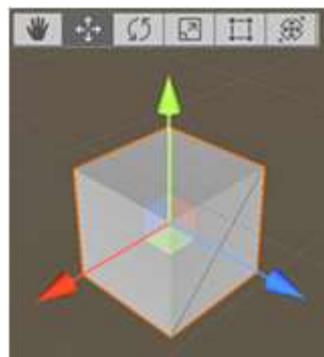


Gambar 2.2 Tampilan Tools [30]

(Sumber: Unity Technology, 2018)

Secara berurutan berikut merupakan nama dan kegunaan dari *tools* tersebut.

- *Hand Tool*: Untuk menggerakkan posisi sudut pandang dalam *scene* [30].
- *Move Tool*: Untuk mengubah posisi sebuah objek terhadap sumbu x, y, dan z. (tergantung tipe *game object*, jika 2D maka hanya x dan y) [30]. *Move tool* dapat dilihat pada Gambar 2.3.



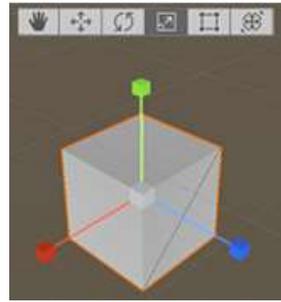
Gambar 2.3 *Move Tool* [30]
(Sumber: Unity Technology, 2018)

- *Rotate Tool*: Untuk mengubah rotasi pada objek terhadap sumbu x, y, dan z [30]. *Rotate tool* dapat dilihat pada Gambar 2.4.



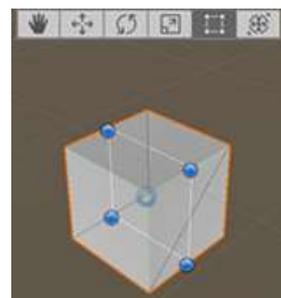
Gambar 2.4 *Rotate Tool* [30]
(Sumber: Unity Technology, 2018)

- *Scale Tools*: Untuk mengubah ukuran dari objek terhadap sumbu x, y, dan z [30]. *Scale tools* dapat dilihat pada Gambar 2.5.



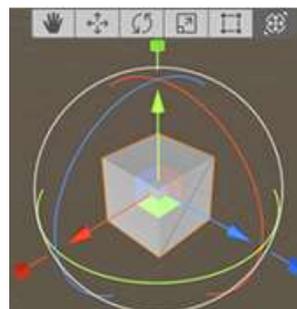
Gambar 2.5 *Scale Tool* [30]
(Sumber: *Unity Technology, 2018*)

- *Rect Tool*: Untuk memodifikasi objek sehingga berubah bentuk sesuai keinginan *user* namun dengan batasan berbentuk persegi [30]. *Rect tool* dapat dilihat pada Gambar 2.6.



Gambar 2.6 *Rect Tool* [30]
(Sumber: *Unity Technology, 2018*)

- *Move/Rotate/Scale Tools*: Merupakan gabungan dari beberapa *tools* untuk mengubah sebuah objek. *move/rotate/scale tools* dapat dilihat pada Gambar 2.7.



Gambar 2.7 *Move/Rotate/Scale Tools* [30]
(Sumber: *Unity Technology, 2018*)

Selain alat untuk mengubah bentuk *Game Object* ada pula beberapa *tools* yang membantu dalam *scene* secara umum yaitu dapat dilihat pada Gambar 2.8:



Gambar 2.8 *Toolbar* dalam *Scene* [30]
(Sumber: *Unity Technology, 2018*)

- *Shaded*: Untuk mengubah mode tampilan objek-objek di dalam *scene*.
 - *2D*: Untuk mengubah tampilan *scene* dan objek dari 3Dimensi menjadi 2Dimensi dan sebaliknya.
 - *Scene Lighting*: Untuk mengatur pencahayaan pada *scene*.
 - *Audio*: Untuk mengaktifkan atau meng-nonaktifkan *audio* dalam *scene*.
 - *Effect*: Untuk menampilkan atau menghilangkan efek dari *Skybox, Flares, Fog*, dan lain-lain.
 - *Gizmos*: Untuk memberikan *debugging* visual.
 - *Search Bar*: Digunakan untuk melakukan pencarian dalam *scene* [30].
- *Scene*: Merupakan *window* utama untuk merancang *game*. Di dalamnya sebagai pengguna dapat melihat, memanipulasi, dan mengatur *object* di dalam sebuah *scene*.
 - *Project*: Merupakan *window* yang berisikan seluruh kumpulan *asset* yang digunakan untuk membuat proyek. Dapat terdiri dari *file, Script, animation, timeline, audio clip*, dan lain-lain.
 - *Hierarchy*: Merupakan *window* yang berisi kumpulan dari *game object* yang digunakan dalam *scene*. *Urutan game object* dapat diatur posisinya dan dapat digabungkan menjadi *parent and child*.
 - *Inspector*: Merupakan *window* yang menampilkan informasi dan keterangan dari *object* atau *asset* yang sedang dipilih.

- *Game: Window* ini merupakan salah satu tampilan utama yang digunakan untuk melihat tampilan ketika permainan di jalankan [30].

2.2.7 *Blackbox Testing*

Blackbox Testing adalah sebuah metode pengujian yang bertujuan untuk mengetahui spesifikasi fungsional dari sebuah perangkat lunak [31]. *Tester* dapat mendefinisikan kondisi dan melakukan penyetelan pada program tersebut. Proses ini berjalan dengan cara mencoba program secara langsung dan memasukkan data kepada setiap formnya. Responden yang digunakan adalah 30 orang yang dipilih [32]. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan fungsi yang dibutuhkan atau tidak.

2.2.8 *User Acceptance Test (UAT)*

User Acceptance Test adalah pengujian yang dilakukan setelah pengujian fungsionalitas [33]. Proses ini merupakan verifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai untuk pengguna [34]. Kemudian akan melakukan perhitungan kelayakan yang diterjemahkan menjadi nilai kualitatif melalui interpretasi menggunakan Skala *Likert* seperti ditunjukkan dalam Tabel 2.2 [23].

Tabel 2.2 Skala *Likert*

Interval Skor	Persentase	Kategori Tingkat
1	Nilai 0 hingga 20%	Sangat Buruk
2	Nilai 21 hingga 40%	Buruk
3	Nilai 41 hingga 60%	Cukup
4	Nilai 61 hingga 80%	Baik
5	Nilai 81 hingga 100%	Sangat Baik

Kemudian akan dihitung menggunakan rumus 2.1

$$\text{Presentase Kelayakan} = \frac{\text{Total Skor Pengujian}}{\text{Total Skor Maksimal}} \times 100\% \dots \dots \dots (2.1)$$