BAB II
TINJAUAN PUSTAKA

2.1. Landasan Teori

Landasan teori merupakan kerangka pemikiran atau teori yang menjadi dasar dari sebuah penelitian. Hal ini mencakup teori, konsep, prinsip, atau kerangka pemikiran untuk memahami hal yang diteliti. Landasan teori membantu peneliti untuk merumuskan pernyataan penelitian, merancang metodologi, menginterpretasikan hasil, dan menyusun kesimpulan. Landasan teori memberikan landasan yang kokoh untuk mengembangkan penelitian dan memahami lebih lanjut tentang subjek yang di teliti.

2.1.1. Makanan

Makanan merupakan hal yang dimakan oleh makhluk hidup untuk mendapatkan nutrisi yang akan diolah menjadi energi. Nutrisi yang di butuhkan oleh manusia salahsatunya adalah karbohidrat, lemak, protein, vitamin, dan mineral.

Sebagian besar makanan berasal dari tumbuhan. Selain tumbuhan makanan juga dapat di peroleh dari zat hewani. Ada juga makanan yang tidak termasuk dari tumbuhan dan heawani, yaitu jamur-jamuran. Selain jamur, bakteri juga dapat di jadikan bahan untuk membuat makanan yang di fermentasikan.

Makanan merupakan kebutuhan pokok yang di butuhkan mayoritas makhluk hidup. Untuk manusia, makanan merupakan hal yang dibutuhkan untuk mengerjakan aktivitas sehari-hari. Dengan makanan, manusia bias membutuhkan energi juga membantu pertumbuhan badan dan otak.

Makanan mengandung gizi. Setiap gizi memiliki fungsi yang berbeda. Karbohidrat merupakan tenaga untuk kehidupan sehari-hari. Protein berguna untuk membantu pertumbuhan tubuh ataupun otak. Lemak merupakan zat yang disimpan oleh tubuh sebagai cadangan makanan dan cadangan energi. Lemak akan digunakan jika tubuh kekurangan karbohidrat. Lemak akan dipecah menjadi glukosa yang berguna untuk tubuh yang kekurangan karbohidrat. [5] [6]

2.1.2. Kuliner Indonesia

Kuliner Indonesia merupakan salah satu tradisi kuliner yang paling kaya di dunia. Bukan hanya itu, kuliner Indonesia juga merupakan makanan dengan cita rasa yang kuat dan beragam. Kekayaan makanan Indonesia merupakan cerminan keberagaman budaya dan tradisi nusantara, dan menempati peran penting dalam budaya nasional Indonesia.

Kuliner Indonesia seringkali merupakan masakan yang kaya dengan bumbu. Hal ini dikarenakan Indonesia merupakan negara yang kaya akan rempah seperti kemiri,cabai, remu kunci, lengkuas, jahe, kencur, kunyit, kelapa, dan masih banyak lagi.

Selain tradisi daerah, kuliner Indonesia juga dipengaruhi oleh tradisi dari negara lain. Hal ini di karenakan nusantara merupakan kepulauan-kepulauan yang melakukan perdagangan dengan dunia luar. Beberapa daerah yang melakuka dagang dengan kepulauan nusantara dan mempengaruhi tradisi kuliner Indonesia adalah India, Tiongkok, Timur Tengah, Afrika, dan Eropa.

Pada dasarnya tidak ada bentuk tunggal yang dapat menunjukan apa itu “makanan Indonesia”. Makan Indonesia merupakan keberagaman makanan dari setiap daerah di Indonesia. Dan makanan-makanan tersebut bisa berasal dari pengaruh local, pengaruh luar, atau bahkan gabungan dari pengaruh lokal dan pengaruh luar. [7]

2.1.3. Media Promosi

Media Promosi merupakan media yang digunakan sebagai sarana untuk memperkenalkan produk atau layanan kepada calon konsumen. Media promosi merupakan hal yang berperan penting untuk memperluas informasi tentang produk yang di promosikan. Dengan adanya media promosi, penjual bisa memberikan gambaran tentang produk atau jasa yang ditawarkan. Dengan demikian konsumen tertarik untuk melakukan transaksi. [8]

2.1.4. Game

Secara bahasa game dapat di artikan menjadi permainnan. Dengan berkembangnya teknologi sekarang game dapat diartikan menjadi sebuah program di suatu platform yang dapat dijalankan secara offline maupun online yang bertujuan untuk menghibur. Jika kita bongkar lebih dalam, game merupakan kumpulan objek

yang di beri perintah untuk menjalankan objek tersebut sesuai perintah pemain untuk menyelesaikan suatu misi.

Game disini merupakan game elektronik yang melibatkan interkasi dengan *user interface* atau sebuah input *device* seperti *joystick*, *keyboard* dan *mouse*, ataupun kontroler yang akan menampilkan feedback visual dari perangkat visual seperti TV (television), monitor, ataupun layar sentuh. Bukan hanya interaksi dan feedback visual, game jaman sekarang juga biasanya di dampingin dengan suara melalui perangkat suara seperti *headphone* atau *speaker*. Bukan hanya itu beberapa game menggunakan perangkat lain yang dapat dibilang unik seperti, *haptic feedback* untuk memberikan sensasi *tactile* seperti pada dualsense (kontroler PS5) ataupun webcam sebagai input pada game. [8]

2.1.5. Game Platform

Game biasanya di kategorikan sesuai platform *hardware* nya masing-masing. Berikut adalah kategori-kategori game berdasarkan platform *hardware* nya:

1. Arcade

Arcade merupakan sebuah mesin yang memiliki focus untuk bermain game. Mesin arcade biasanya merupakan mesin besar yang biasanya diharusnkan untuk memasukan koin atau scan kartu khusus untuk memainkannya. Mesin arcade merupakan mesin yang paling populer untuk memainkan game sampai tahun 2000 an dimana game konsol dan PC mulai beredar di pasaran. [9]

2. Konsol

Konsol merupakan mesin khusus untuk bermain game. Tidak seperti mesin arcade, konsol merupakan mesin yang lebih kecil dan kebanyakan dari konsol memerlukan TV dan kontroler khusus untuk memainkan gamenya. Ada juga konsol yang sudah menyediakan perangkat visual dan kontroler nya. Biasanya konsol tersebut merupakan konsol portable. [10]

3. PC (Komputer)

Komputer merupakan mesin yang bukan berfokus untuk bermain game. Meskipun game pertama dibuat di iterasi awal computer. Tetapi seiring berjalannya

waktu, komputer mulai menjadi salah satu platform paling populer di kalangan pemain game. Meskipun bukan mesin yang berfokus untuk bermain game, karena kepopuleran game di komputer sekarang banyak penjual hardware yang memasarkan hardware mereka untuk pemain game.Komputer merupakan platform game yang paling digemari oleh developer. Kurang lebih 60% dari developer game lebih tertarik untuk membuat game di komputer. [11]

4. Mobile

Mobile disini biasanya merujuk ke smartphone. Seperti halnya komputer, *smartphone* biasanya dibuat bukan untuk game. Tetapi dengan kepopuleran game dan dengan banyaknya game yang dibuat khusus untuk *smartphone*, sekarang platform mobile merupakan platform paling populer di kalangan pemain game. [12]

5. VR (Virtual Reality)

VR merupakan mesin yang mengharuskan penggunanya untuk menggunakan headset VR yang menampilkan visual *device* langsung didepan mata. control untuk vr biasanya merupakan kontroler khusus atau bahkan *pose tracking* dimana mesin VR akan mengikuti gerakan penggunanya. Untuk saat ini VR belum menjadi platform yang populer karena harganya yang mahal dan kurangnya game yang "bagus" untuk VR. [13] [14]

6. Cloud

Cloud Gaming merupakan platform game yang harus dimainkan secara online atau *stream* game tersebut secara *remote* dari server penyedia layanannya ke *device* pemainnya. Dengan demikian pemain tidak harus mempunyai mesin yang mahal untuk bermain game atau bahkan harus mempunyai konsol atau komputer untuk memainkan game yang berat. Pemain bisa memainkan game konsol atau PC lewat smartphone mereka. Yang mereka butuhkan adalah internet yang cepat dan stabil. [15]

2.1.6. Game Genre

Game genre merupakan kategori spesifik dari game-game yang memiliki *gameplay* karakteristik yang sama. Game genre biasanya bukan di lihat dari cerita ,setting, atau bahkan dimana game tersebut di mainkan, melainkan bagaimana pemain

berinteraraksi dengan gamenya. Berikut beberapa genre dan sub-genre game yang paling populer:

2.1.7. Action

Genre *action* menekankan tantangan fisik yang membutuhkan *hand-eye coordination* dan skill motorik untuk menyelesaikan game tersebut. Focus dari genre action adalah pada pemain yang mengontrol hamper semua tindakan yang ada pada game tersebut. Genre *action* memiliki banyak sub-genre diantaranya adalah :

1. Platformer
2. Shooter
3. Fighting
4. beat`em up
5. stealth
6. survival
7. rhythm
8. battle royale.

1. Adventure

Adventure sesuai namanya merupakan genre khusus untuk game petualangan. Tapi, bukan seperti film petualangan, genre *adventure* genre ini bukan merupakan game yang memiliki setting berpetualangang melainkan berfokus kepada *gameplay* nya. Genre *adventure* biasanya tidak memiliki *gameplay* yang mengharuskan pemain untuk melakukan aksi seperti genre action, melainkan berfokus untuk pemain menyelesaikan puzzle dengan berinteraksi dengan NPC(Non-Playabel Character) atau dengan environment gamenya. Biasanya tanpa konfrontasi apapun. Genre ini biasanya disebut sebagai genre "*purist*" dan menghindari aksi apapun kecuali dalam mini game. Berikut beberapa sub-genrenya: [17]

1. Text Adventure
2. Graphic Adventure
3. Visual Novel
4. Interactive Movie

2. Action-Adventure

Action-adventure merupakan genre yang menggabungkan dua genre sebelumnya yaitu *action* dan *adventure*. Genre ini biasanya mengharuskan player untuk mengumpulkan barang, menyelesaikan puzzle, dan melakukan pertarungan. Berikut beberapa sub-genrenya:

1. Survival Horor
2. Metroidvania

3. Puzzle

Puzzle merupakan genre yang memfokuskan pemain untuk menyelesikan masalah dalam bentuk puzzle. Genre puzzle biasanya akan menguji *problem-solving* skill pemain seperti logika, *pattern recognition*, *sequence sloving,* atau bahkan menyelesaikan kosa kata. Berikut sub-genrenya:

1. Breakout Clone
2. Logical
3. Trial and Error Exploration
4. Hidden Object
5. Reveal Picture
6. Tile-Matching
7. Traditional
8. Puzzel-Platform

4. Role-Play

Genre *Role-Play* mengambil *gameplay* dari game *tabletop* tradisional seperti D&D (Dungeon and Dragon). Game dari genre ini biasanya memberi pemain dari game ini sebuah *role* (peran) dimana karakter tersebut akan tumbuh dan menjadi lebih baik dan lebih kuat seiring berjalannya waktu. Pemain biasanya akan mendapatkan EXP (Experience Poin) dimana jika sudah terkumpul akan menambah *level* karakter pemain tersebut. EXP akan di dapatkan apabila pemain menyelesaikan rintangan atau mengalahkan musuh. Sub-genre dari *role-play* adalah :

1. Action RPG
2. MMOPRG
3. Rougelikes

4. Tactical RPG
5. Sandbox RPG
6. First-Person Party-Based RPG
7. Monster Tamer

5. Strategy

Genre ini memerlukan pemain untuk bisa merencanakan strategy tertentu untuk menyelesaikan gamenya. Pemain diberikan *poin of view* seperti komandan saat berperang dimana pemain harus mengontrol unit mereka. Genre ini biasanya mengambil satu dari empat form arsitektur tergantung dari apakah game tersebut *turn-base* atau *real-time* dan apakah game tersebut focus mengatur strategi atau taktik. Berikut beberapa sub-genre dari genre *strategy* :

1. 4X
2. Artillery
3. Auto Battler
4. MOBA (Multiplayer Online Battle Arena)
5. Real-Time Strategy
6. Real-Time Tactics
7. Tower Defense
8. Turn-Base Strategy
9. Turn-Base Tactics
10. Wargame
11. Grand Strategy Wargame

6. Simulation

Genre simulation merupakan genre game yang di design untuk menyerupai aspek dunia nyata. Sub-genre dari genre ini adalah:

1. Construction and Management Simulation
2. Life Simulation
3. Vehicle Simulation

2.1.8. Platformer

Platformer merupakan sub-genre dari genre game action. Objektif utama dari genre platformer adalah menggerakan karakter pemain dari poin a ke poin b dalam

environment yang di sediakan dalam game. game platformer biasanya memiliki karakter yaitu level yang tidak rata yang memiliki ketinggian yang bervariasi dimana pemain harus melewatinya dengan cara melompat dan memanjat. Beberapa game platformer memiliki aksi lain selain melompat dan memanjat seperti berayun , memantul, atau bahkan meluncur.

Sub-genre ini dimulai di tahun 1980 dengan game arcade yaitu *Space Panic*. *Space Panic* belum memiliki gerakan melompat melainkan hanya berjalan dari kiri ke kanan dan memanjat. Setelah itu Nintendo membuat *Donkey Kong* yang rilis pada tahun 1981. *Donke Kong* lah yang benar-benar membuat genre *Climbing Games* di peta. *Donkey Kong* memberi template untuk developer lain untuk membuat game dengan genre yang sama. Setelah beberapa game yang terinspirasi oleh *Donkey Kong,* Nintendo membuat game *core* nya yaitu *Super Mario Bros* yang rilis pada tahun 1985. Pada tahun 1985 sampai 1990-an genre platformer menjadi gendre game yang paling dominan.

Seiring berjalannya waktu dan majunya teknologi dalam pengembangan game, di tahun 1990-an grafis 3D sudah mulai masuk ke dunia game. Dengan demikian, genre platformer pun berevolusi. Game seperti *Super Mario* 64 (1996) dan *Banjo-Kazooie*(1998) membawa dunia platformer menjadi 3D. [18] [19]

2.1.9. Game Engine

Game *Engine* merupakan framework yang fungsi utamanya adalah untuk mengembangkan game. Game *engine* biasanya mempunyai library yang relevan dan program yang mensupport pengembangan game. Fungsi utama yang disediakan game engine biasanya terdiri dari *rendering engine* untuk grapik 2D dan 3D, *physics engine* atau *collision detection* dan *respond*, *sound*, *scripting*, *animation*, AI (*artificial intelegent*), *networking*, *memory management*, *threading*, *localization support*, *scene graph*, dan *cinematics* (*video support*). [20]

Berikut beberapa daftar game engine paling populer :

1. 4A Engine
2. Anvil
3. CryEngine
4. Dunia Engine

5. Frostbite
6. Godot
7. id Tech
8. REDengine
9. RAGE (Rockstar Advanced Game Engine)
10. RPG Maker
11. Source
12. Unity
13. Unreal

2.1.10. Unity

Unity merupakan game *engine* yang digunakan oleh peneliti untuk melakukan penelitian.Unity merupakan *game engine cross platform* yang di kembangkan oleh Unity Technologies pada tahun 2005. Unity pertama kali di umumkan di *Apple Worldwide Developers Conference* awalnya sebagai *game engine* eksklusif untuk Mac OS X. Seiring berjalannya waktu, *Unity* memperluas support ke platform lainnya seperti *desktop* PC, *mobile*, *console*, AR (*augmented reality*), dan VR (*virtual reality*). Unity merupakan game *engine* yang populer untuk mobile game development seperti di platform IOS dan Android. Unity juga merupakan salah satu game *engine* yang *beginner friendly* dan populer untuk pengembangan game indie.

Unity bisa digunakan untuk membuat game 3D dan 2D. Unity juga menggunakan *scripting* API dengan C#. Unity juga menyediakan plugins, dan fitur drag and drop. Dalam game 2D, unity memperbolehkan pengembang untuk mengimport sprites dan *renderer* 2D yang *advance*. Untuk game 3D *Unity* menyediakan *texture compression*, *minimaps*, setting resolusi untuk platform yang berbeda, menyediakan support untuk *bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion, dynamic shadows* dengan menggunakan *shadow maps, render-to-texture* dan *full screen post-processing effects*.

Nickolas Francis dan Joachim Ante pada tahun 2002 mengembangkan engine mereka. Tetapi pada saat itu mereka memilih untuk bekerja sama dengan David Helgason untuk membuat game studio yang nantinya akan membuat game *engine* yang akan di patenkan ke studio lain. Setelah beberapa tahun mengembangkan game engine

tersebut, game engine yang bernama unity pun akhirnya di luncurkan pada tahun 2005. [21]

Seiring berjalannya waktu, unity juga terus menerus di *update*. Berikut beberapa versi dari unity:

1. Unity 2.0
2. Unity 3.0
3. Unity 4.0
4. Unity 5.0
5. Unity

2.1.11. Android

Android merupakan sebuah sistem operasi mobile yang dikembangkan oleh Google. Dirancang untuk berbagai perangkat mobile seperti smartphone, tablet, dan smartwatch. Android didasarkan pada kernel Linux dan dirancang secara khusus untuk menawarkan pengalaman yang intuitif dan responsif kepada pengguna.

Salah satu ciri khas Android adalah sifatnya yang open source. Source code android dapat diakses, dimodifikasi, dan didistribusikan kembali oleh siapa saja, memungkinkan kolaborasi global dalam pengembangan Android. Android digunakan oleh berbagai produsen perangkat seperti Samsung, Xiaomi, dan OnePlus, Android telah menjadi salah satu sistem operasi mobile paling populer di dunia.

Android menyediakan platform pengembangan aplikasi yang kuat melalui Android SDK, memungkinkan pengembang untuk membuat aplikasi dan game dengan berbagai bahasa pemrograman seperti Java, Kotlin, dan C++.

pada tahun 2003, ketika Andy Rubin, Rich Miner, Nick Sears, dan Chris White mendirikan perusahaan Android, Inc. Mereka memiliki visi untuk mengembangkan sistem operasi mobile yang dapat bersaing dengan produk-produk yang sudah ada pada waktu itu, seperti BlackBerry dan Windows Mobile.

Pada awalnya, Android, Inc. berfokus pada pengembangan sistem operasi untuk kamera digital. Namun, dengan berjalannya waktu, fokus mereka beralih ke perangkat mobile. Pada tahun 2005, Google mengakuisisi Android, Inc., yang memicu spekulasi bahwa Google sedang merencanakan untuk memasuki pasar perangkat mobile.

Setahun kemudian, pada tahun 2007, Google bersama dengan produsen perangkat mobile dan operator telekomunikasi lainnya membentuk Open Handset Alliance (OHA), sebuah konsorsium yang bertujuan untuk mengembangkan standar terbuka untuk perangkat mobile. Android merupakan bagian integral dari visi OHA.

Pada tanggal 5 November 2007, Open Handset Alliance mengumumkan Android, sebuah platform open source untuk perangkat mobile yang didasarkan pada kernel Linux. Android SDK (Software Development Kit) pertama kali dirilis pada bulan November 2007 juga, memberikan kesempatan bagi para pengembang untuk mulai membuat aplikasi untuk sistem operasi baru ini.

Pada tahun 2008, Google merilis Android 1.0, versi pertama sistem operasi Android, yang dilengkapi dengan fitur-fitur dasar seperti browser web, aplikasi email, dan akses ke Google Maps. Tahun berikutnya, Android mulai mendapatkan momentum, dengan peluncuran HTC Dream (juga dikenal sebagai T-Mobile G1), yang merupakan perangkat Android pertama yang tersedia di pasaran.

Dari sana, Android terus berkembang pesat. Google secara teratur merilis versi baru Android dengan pembaruan fitur, peningkatan kinerja, dan perbaikan keamanan. Android juga mulai tersedia di berbagai perangkat, termasuk smartphone, tablet, smartwatch, TV pintar, mobil, dan perangkat IoT (Internet of Things). [22]

2.1.12. Usecase Diagram

Usecase diagram merupakan jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan fungsi, ruang lingkup, dan interaksi *actor* dengan sistem dan hal apa saja yang dilakukan *actor* kepada sistem secara rinci. Dengan adanya use case diagram, kita dapat mengidentifikasi kebutuhan *actor*, memperjelas persyaratan sistem, dan merancang fungsional sistem. Use case diagram juga bisa membantu komunikasi antara pengembang. Dengan demikian pengembang dapat memahami dengan baik antara aktor dan sistem dengan baik. [23]

Berikut adalah komponen dari use case diagram :

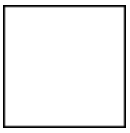
1. Actor

Actor merupakan entitas yang berinteraksi dengan system. *Actor* digambarakan sebagai manusia atau objek. Icon dari *actor* dapat di lihat pada gambar 2.1.

Gambar 2.1 Actor Usecase diagram

2. Sistem

Sistem merupakan aplikasi yang dimaksud. Sistem merupakan entitas yang menerima input dari interaksi *actor* dan sebagai responnya memberikan output. Icon dari Sistem dapat di lihat pada gambar 2.2.



Gambar 2.2 Sistem Usecase diagram

3. Usecase

Usecase disini merupakan tindakan yang bisa dilakukan oleh *actor* kepada system. Icon dari use case dapat di lihat pada gambar 2.3.

Gambar 2.3 Usecase Usecase

4. Relasi

Relasi merupakan hubungan antara *actor* dan use case. Terdapat beberapa relasi di dalam use case diagram. Berikut jenis relasinya:

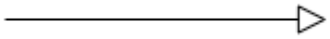
a. Association

Association merupakan jenis realsi yang menunjukan relasi *actor* dengan use case tanpa adanya interaksi yang spesifik. Icon dari *association* dapat di lihat pada gambar 2.4.

Gambar 2.4 Association Usecase

b. Generalization

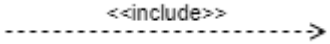
Generalization merupakan *actor* ataupun sistem yang lebih spesifik. Icon dari *Generalization* dapat di lihat pada gambar 2.5.



Gambar 2.5 Generalization Usecase

c. Include

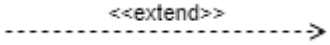
Include merupakan use case yang membutuhkan usecase lainnya dan tidak dapat berdiri sendiri. Icon dari *Include* dapat di lihat pada gambar 2.6.



Gambar 2.6 Include Usecase

d. Extend

Extend merupakan use case yang dapat di extend atau perluas ke satu atau beberapa use case. Icon dari *Extend* dapat di lihat pada gambar 2.7.



Gambar 2.7 Extend Usecase

2.1.13. Use Case Scenario

Use Case scenario memberikan narasi yang mendetail tentang bagaimana use case tertentu berkembang. *Use case scenario* menceritakan bagaimana urutan interaksi antara *actor* dan sistem yang menggambarkan berbagai langkah dan kondisi yang mungkin timbul selama eksekusi. *Use Case scenario* memberikan pandangan yang lebih terperinci, membantu pengembang memvisualisasikan interaksi pengguna tertentu. Contoh dari use case scenario dapat dilihat di tabel 2.1. [24]

Tabel 2.1 Tabel Contoh Use Case Scenario

Related Requirement	**SKPL-F-3**	
Actor	Pemain	
Goal in Context	Pemain berhasil memasak makanan	
Precondition	1. Pemain sudah didalam game 2. Pemain tidak berada di *memory land* 3. Pemain memiliki resep makanan yang akan di masak dari memory land	
Successful End Condition	Pemain memasak makanan sesuai resep	
Failed End Condition	Pemain memasak makanan	
Trigger	Pemain menekan button aksi di area trigger di dapur	
Main Flow	*Step*	*Action*
	1	Pemain menuju dapur dan menekan button aksi di area kompor
	2	Game menampilkan menu memasak
	3	Pemain memasukan bahan utama makanan, bumbu makanan, dan cara memasak

	4	Game memberi pemanen buff untuk kepada karakter pemain yang dapat di gunakan di memory land.
Extension	*Step*	*Branching Action*
	3	Pemain gagal mengikuti resep dan game menampilkan menu makanan gagal di buat

2.1.14. Activity Diagram

Activity diagram adalah salah satu jenis diagram UML (*Unified Modeling Language*) yang menggambarkan aliran kerja atau aliran aktivitas dalam suatu sistem, proses bisnis, atau fungsi perangkat lunak. Diagram ini sering digunakan untuk memodelkan proses bisnis, logika perangkat lunak, atau alur kerja dari suatu sistem.

Tujuan dari diagram aktivitas adalah untuk memberikan pemahaman yang jelas tentang bagaimana suatu sistem atau proses berjalan, sehingga memudahkan untuk menganalisis, perancangan, dan implementasi sistem atau proses yang efisien. Activity diagram sering digunakan dalam tahap analisis dan desain perangkat lunak untuk mengkomunikasikan alur kerja dengan jelas kepada pengembang sistem. [25]

Berikut adalah komponen dari use case diagram :

1. Initial State

Initial state merupakan simbol yang digunakan untuk menandakan status awal dari suatu proses di dalam sistem. Icon dari *initial state* dapat di lihat pada gambar 2.8.

Gambar 2.8 Initial State Activity Diagram

2. Activity

Merupakan sebuah aktivitas yang terjadi didalam sistem. Icon dari *Activity* dapat di lihat pada gambar 2.9.

Gambar 2.9 Activity Activity Diagram

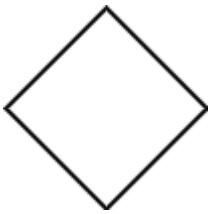
3. Control Flow

Control flow digunakan untuk menunjukan transisi dari satu simbol ke simbol lainnya. Icon dari *contra flow* dapat di lihat pada gambar 2.10.

Gambar 2.10 Control Flow Activity Diagram

4. Decision

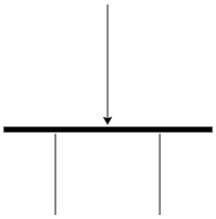
Decision merupakan simbol yang digunakan jika sistem diharuskan untuk memilih. Biasanya terdiri dari dua atau lebih jalur. Icon dari *decision* dapat di lihat pada gambar 2.11.



Gambar 2.11 Decision Activity Diagram

5. Fork

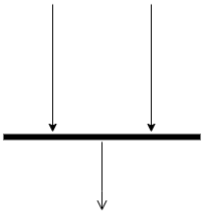
Fork biasanya digunakan jika sistem harus melakukan lebih dari satu aksi bersamaan. Icon dari *fork* dapat di lihat pada gambar 2.12.



Gambar 2.12 Form Activity Diagram

6. Join

Join biasanya digunakan jika terdapat 2 buat aktivitas yang berjalan bersamaan akan menyatu. Icon dari *join* dapat di lihat pada gambar 2.13.



Gambar 2.13 Join Activity Diagram

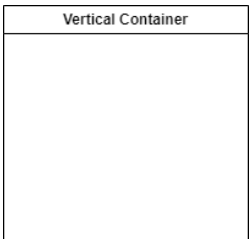
7. Merge

Merge akan digunakan apabila terdapat 2 buah aktivitas yang tidak berjalan bersamaan akan di satukan. Icon dari *merge* dapat di lihat pada gambar 2.14.

Gambar 2.14 Merge Activity Diagram

8. Swimlanes

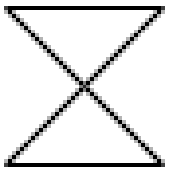
Swimlanes digunakan untuk memberi grup untuk aktivitas-aktivitas yang berhubungan satu sama lain. Penggunaan *swimlanes* tidak di haruskan. Tetapi akan membuat *activity diagram* menjadi lebih mudah dibaca. Icon dari *swimlanes* dapat di lihat pada gambar 2.15.



Gambar 2.15 Swimlanes Activity Diagram

9. Time Event

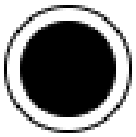
Time event digunakan bila sistem harus menghentikan proses sementara karena aktivitas tersebut memakan waktu untuk di proses. Icon dari *time event* dapat di lihat pada gambar 2.16.



Gambar 2.16 Time Event Activity Diagram

10. End State

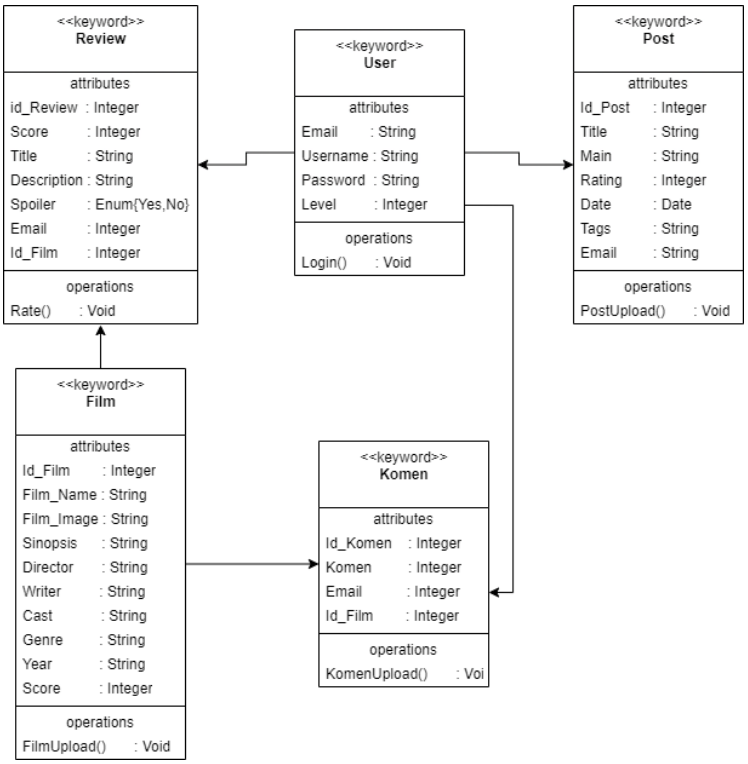
End state digunakan bila proses sistem berakhir. Icon dari *end state* dapat di lihat pada gambar 2.17.



Gambar 2.17 End State Activity Diagram

2.1.15. Class Diagram

Class diagram adalah jenis diagram UML (*Unified Modeling Language*) yang digunakan dalam rekayasa perangkat lunak untuk memodelkan struktur dan realsi dari *class-class* yang ada di dalam sistem. Digunakan untuk membangun dan memodelkan sistem *object-oriented*. *Class* diagram memberikan ringkasan yang berkualitas untuk membentu komunikasi antar pengembang dan mendokumentasikan struktur dari software yang dibuat. Contoh dari *class* diagram dapat dilihat di gambar 2.18. [26]



Gambar 2.18 Class Diagram

Seperti diagram lainnya class diagram juga mempunyai komponen. Berikut adalah komponen-komponen yang menyusun *class* diagram:

1. Class

Class merupakan sebuah template untuk membeuat sebuah objek. Objek merupakan kumpulan dari *class-class* yang setiap *class* nya terdiri dari tiga bagian yaitu :

a. Class Name

Sesuai namanya *class name* merupakan nama dari *class* tersebut

b. Attributes

Attributes atau *properties* merupakan representasi dari data class.

c. Methods

Methods merupakan representasi dari tindakan atau fungsi dari *class*.

2. Association

Association merupakan hubungan antara dua buah *class*. *Association* biasanya menghubungkan objek dari suatu *class* dengan objek dari *class* lain.

3. Directed Association

Directed Association menunjukan hubungan antara dua *class* dimana satu *class* menujukan arah, yang berarti satu *class* diasosiasikan dengan *class* lain secara spesifik. Contohnya jika suatu *class* bertanggung jawab memulai sebuah asosiasi atau *class* mana yang membutuhkan *class* lainnya.

4. Aggregation

Aggregation merupakan hubungan yang special dimana asosiasi tersebut mewakili hubungan yang menggunakan seluruh bagian *class nya*. Hubungan ini memiliki satu *class* yang berperan sebagai wadah dan lainnya sebagai isi dari wadah tersebut. Dalam hubungan *aggregation*, *class* isi masih dapat berdiri sendiri meskipun *class* wadah tidak ada.

5. Composition

Composition merupakan bentuk hubungan Aggregation yang lebih kuat dimana jika di hubungan aggregation class isi bisa berdiri sendiri, di hubungan composition *class* isi bergantung pada *class* wadah.

6. Generalization

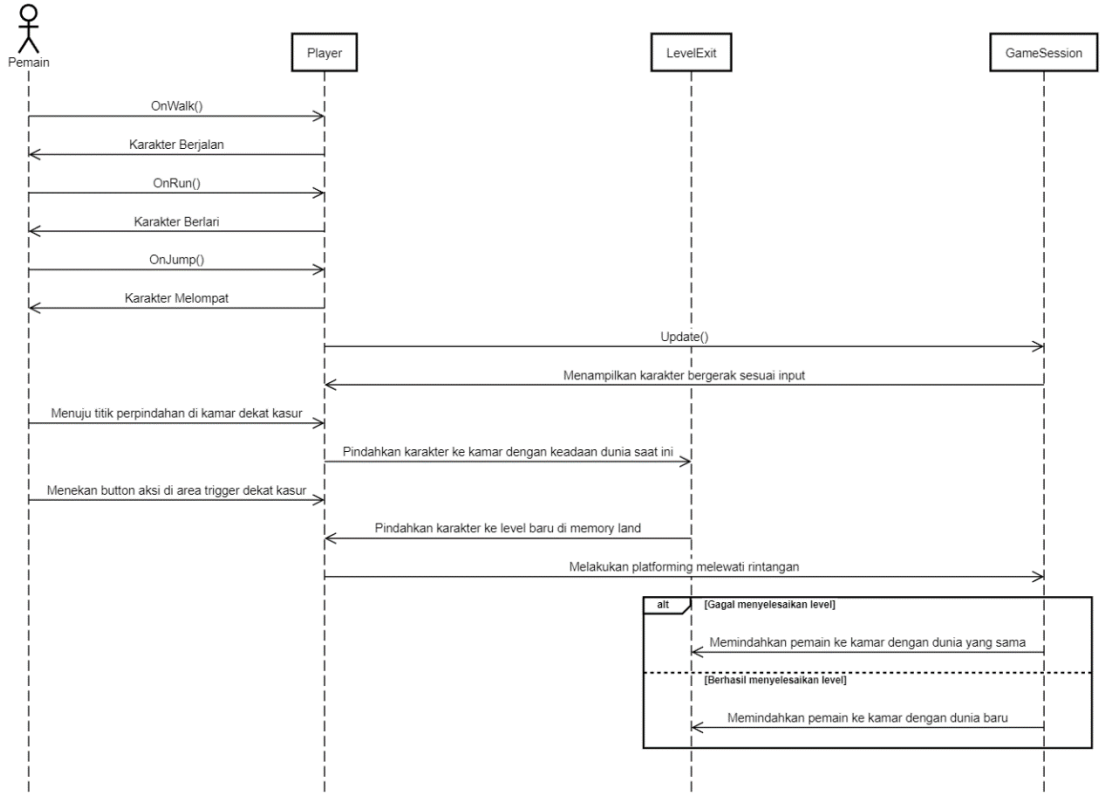
Generalization merupaka hubungan antara *class parent* atau *superclass* dan *class child* atau *subclass* dimana *class* parent akan mewariskan properti dan perilaku ke *class child.*

7. Realization

Realization merupakan hubungan antara *class* yang akan mengimplementasikan fitur dari sebuah *interface.*

2.1.16. Sequence Diagram

Sequence diagram adalah salah satu jenis diagram UML (*Unified Modeling Language*) yang digunakan untuk menampilkan interaksi antara objek dalam sebuah sistem dengan detail. Sequence diagram juga akan menampilkan pesan atau perintah yang dikirim. *Sequence* diagram akan menunjukan panggilan metode yang terjadi antara objek-objek selama proses use case atau scenario. Contoh dari *sequence* diagram dapat dilihat di gambar 2.19. [27]



Gambar 2.19 Sequence Diagram

Komponen-komponen dalam diagram sequence meliputi:

1. Actors

Merepresentasikan sebuah jenis peran yang akan berinteraksi dengan sistem dan objeknya. *Actors* selalu disimpan diluar dari sistem. *Actors* biasanya merepresentasikan manusia tapi bisa juga subjek *external*.

2. Object

Mewakili instance dari suatu *class*. Objek direpresentasikan dengan sebuah kotak yang berisi nama objek dan garis hidup (lifeline) yang menunjukkan durasi hidup objek selama proses.

3. Message

Menunjukkan panggilan metode atau pertukaran pesan antara objek-objek. Pesan direpresentasikan dengan garis panah yang menghubungkan objek-objek, dan dapat diberi label untuk menunjukkan nama metode yang dipanggil.

4. Lifeline

Menunjukkan durasi hidup objek selama proses. Garis hidup direpresentasikan oleh garis vertikal yang menghubungkan objek dengan garis waktu (time axis).

5. Activation

Menunjukkan periode waktu ketika suatu objek sedang melakukan operasi atau merespon pesan. Aktivasi direpresentasikan oleh kotak yang menggantikan garis hidup dan menunjukkan waktu di mana objek sedang aktif.

6. Interaction Fragment

Bagian dari diagram yang menunjukkan kondisi, percabangan, atau iterasi dalam urutan interaksi. Contoh fragment interaksi adalah fragment opsional dan loop.