

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam penulisan skripsi ini peneliti melakukan Studi Literature dengan menggali dari berbagai penelitian sebelumnya yang berkaitan dengan penelitian dari penulis sebagai bahan rujukan serta perbandingan baik mengenai kelebihan maupun kekurangan yang sudah ada.

Penelitian yang dilakukan oleh (Anna, 2020) “Perancangan *Game* Petualang Sebagai Sarana Pembelajaran Edukasi Berbasis Android” Penelitian ini dibangun dengan tujuan memberikan media pembelajaran yang berbeda melalui media *game* edukasi bertema pembelajaran akutansi dasar berisi hitungan yang rumit didalamnya. Sistem yang dibangun menggunakan metode waterfall dengan melalui beberapa tahapan analisis, rancangan, penerapan, integrating, operation selain itu peneliti juga menerapkan metodologi pengembangan multimedia dimana kerangka kerja menggunakan pendekatan sistem informasi yaitu pengonsepan, perancangan tampilan, pengumpulan bahan sesuai kebutuhan, pembuatan objek, pengujian dan penyimpanan. Pada penelitian tersebut dapat disimpulkan dalam pengujian yang telah dilakukan mendapat respon sangat baik dalam kuesioner yang dilakukannya.

Peneliti yang dilakukan oleh (Rahmawati, 2020) “Pengembangan *Game* Petualang Untuk Pembelajaran Berhitung”. Peneliti melihat bahwa anak yang masih duduk di jenjang Pendidikan Sekolah Dasar cenderung bermain Handphone secara terus menerus yang akan mengurangi waktu untuk belajar. Sehingga peneliti mengangkat tema berhitung yang ditujukan untuk anak yang masih berada di jenjang Pendidikan Sekolah Dasar. Metode yang digunakan adalah Penelitian dan Pengembangan di bidang pendidikan yang berfokus pada pengembangan produk dan menguji keefektifan produk dalam mencapai tujuan yang

diharapkan. Produk yang dikembangkan dalam penelitiannya berupa media pembelajaran berhitung untuk sekolah dasar berupa *game* petualang. Pada penelitian tersebut dapat disimpulkan bahwa *game* petualangan untuk pembelajaran berhitung mendapatkan 81% respon sangat baik dari siswa.

Penelitian yang dilakukan (Apriyandi, 2019) “Penerapan *Endless Runner Game* Untuk Memperkenalkan Pariwisata Kota Pontianak” peneliti melihat wisatawan lokal dan internasional di Kota Pontianak sangat rendah dari kota lain. Salah satu faktornya adalah publikasi. Sehingga peneliti membuat media untuk memperkenalkan pariwisata yang ada di Kota Pontianak melalui *game endless runner*. Metode yang digunakan adalah *Game Development Live Cycle* yang memiliki beberapa tahapan yaitu *Inititation, Pre-productions, Productions, Testing, Beta, Release*. Berdasarkan hasil yang ditunjukkan menggunakan *likert’s Summated Rating* bahwa *game* tersebut dapat diterima baik. Dengan adanya kuesioner yang dibuat menunjukkan nilai 1139 dari nilai maksimal 1300, dapat disimpulkan bahwa *game* tersebut telah dirancang dengan cukup berhasil sebagai media untuk memperkenalkan Kota Pontianak

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Kategori <i>Game</i>	Materi
1	Anna, Annisa, R., & Parwandar, R. (2020).	Aksi	Soal-Soal tentang Akuntansi
2	Rahmawati, I., & Leksono, I. P. (2020).	Aksi	Soal Matematika
3	Apriyandi, D., Anra, H., & Pratiwi, H. S. (2019).	Aksi	Pariwisata kota Pontianak
4	Penelitian yang diusulkan	RPG	Pendakian Gunung Argopura

1.2 Landasan Teori

1.2.1 *Game* Edukasi

Game Edukasi adalah permainan yang di dalamnya terdapat informasi dan pembelajaran yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi aturan tertentu. (Handriyantini, 2009).

Permainan edukasi adalah permainan edukatif, yang mengandung unsur mendidik, dan informatif yang dirancang dan dibuat untuk merangsang daya pikir dan melatih untuk memecahkan suatu masalah

1.2.2 *Game* RPG

Prinsip dasar dari RPG (*Role Playing Game*) adalah untuk mengendalikan karakter yang kita ikuti pengembaraan di alam semesta fiksi dimana bisa berbaur dengan unsur fantasi. Fitur utama dari genre terletak pada pengembangan karakter. Keuntungan dan peningkatan tingkat statistik karakter, juga harus memperhatikan peralatan yang akan digunakan karakter. karena mempengaruhi statistik karakter, yang dapat berguna untuk menghadapi monster, kesulitan tumbuh, yang berjalan. *Quest* yang diberikan oleh NPC (*Non Karakter* pemain) tersebar luas dalam permainan seperti ini. Dalam sebuah permainan RPG, jarang ada yang "kalah" atau "menang". Ini membuat permainan RPG berbeda dari jenis permainan lainnya seperti Monopoli atau Ular Tangga.

Permainan RPG mempunyai daya tarik karena permainan-permainan ini mengajak para pemain untuk menggunakan imajinasi mereka. RPG biasa lebih mengarah ke kolaborasi sosial daripada kompetisi. Pada umumnya dalam RPG, para pemain tergabung dalam satu kelompok. Permainan RPG rata-rata dimainkan seperti sebuah drama radio : ketika seorang pemain "berbicara", dia berbicara sebagai

tokohnya dan ketika si pemain ingin tokohnya melakukan sesuatu yang fisik (seperti menyerang sebuah monster atau membuka sebuah gembok) dia harus menggambarkannya secara lisan.

1.2.3 Gunung Argopura

Menurut (BKSDA 2018), Jawa Timur, Argopura merupakan salah satu puncak dari rangkaian pegunungan Hyang. Jalur pendakian Gunung Argopura merupakan salah satu jalur pendakian terpanjang di Pulau Jawa. Jalur pendakian yang menawarkan berbagai hal menarik di dalamnya. Mulai dari Pesona alam seperti hutan hujan, savana, dan hutan cemara, Danau, peninggalan zaman kerajaan, peninggalan kolonial, hingga perjumpaan dengan satwa ikonik seperti Merak yang tidak dapat dijumpai di gunung lain dapat dinikmati dalam perjalanan ke Argopuro.

1.2.4 Jenis - Jenis *Game*

Menurut (Sulistyo, 2010), ada beberapa jenis platform di dunia *game* yang selalu dipilih oleh pengguna *games* , yaitu:

1. Arcade *games*

adalah yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis video *games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati, seperti Capit Boneka, pistol, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).

2. PC *Games*

yaitu video *game* yang dimainkan menggunakan Personal Computers atau Laptop.

3. Console *games*

yaitu video *games* yang dimainkan menggunakan console tertentu, seperti Playstation 1-5 , XBOX 360, dan Nintendo Wii.

4. *Handheld games*

yaitu yang dimainkan di console khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.

5. *Mobile games*

yaitu yang dapat dimainkan atau khusus untuk mobile phone atau Android.

1.2.5 *Genre game*

Game dibagi atas beberapa genre, diantaranya yaitu :

1. *Action*

permainan pada genre ini menunjukkan aksi yang cukup memiliki konten kekerasan tinggi, dimana terdapat aksi tembak menembak, memukul, menendang tergantung cerita dan tokoh di dalamnya. Pada permainan jenis ini, pemain memerlukan kecepatan dalam reflex serta kordinasi yang baik dalam memainkannya. Contoh : PB (Point Blank), Into The Dead dan PUBG.

2. *Fighting* (pertarungan).

Ada yang mengelompokan permainan genre *fighting* di bagian Aksi, namun penulis berpendapat berbeda, permainan ini memang memerlukan kecepatan refleks dan koordinasi mata dan tangan, tetapi inti dari permainan ini adalah penguasaan pada jurus atau special action (hafal caranya dan lancar mengeksekusinya), pengenalan karakter 7 dan timing sangatlah penting, combo-pun menjadi cara untuk mengalahkan lawan secepat mungkin. Contoh : Naruto, Dragon Ball, Mortal Kombat dan Tekken.

3. *Adventure* (Petualangan).

Permainan genre ini merupakan permainan yang melakukan penjelajahan seperti memanjat, menelusuri hutan, meloncati tebing yang terpisah jurang, berayun dari pohon ke pohon lainnya, bergulat melawan tanaman atau pun hewan liar demi mencari clue atau petunjuk menuju rintangan berikutnya. Adapun yang bertualang diantara jalan jalan perkotaan sekedar mencari tongkat kayu ataupun sabuk untuk membuat alat untuk misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam permainan jenis ini. Contoh : Kings Quest, dan Space Quest.

4. *Strategy* (strategi).

Video *game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Kebanyakan *game* strategi adalah *game* perang. Contoh : Warcraft, Red Alert. Clash of Clans

5. *Simulation* (Simulasi).

Permainan jenis ini seringkali menggambarkan kehidupan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari makan hingga pekerjaan, membangun tempat tinggal hingga kota, mengatur pajak penghasilan dan dana kota. Permainan genre ini selayaknya hidup dari awal lahir yang tidak memiliki apa-apa hingga menjadi konglomerat penguasa bisnis dan lain sebagainya. Ada juga seperti melakukan eksperimen percobaan antara gen A terhadap gen lainnya hingga mendapatkan hasil kloning yang unik. Pada permainan jenis ini membuat pemain harus berpikir dalam mendirikan, membangun dan

mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: The Sims, Metropolis Mania, Zoo Tycoon.

6. *Puzzle* (teka-teki).

Permainan jenis ini sesuai dengan namanya mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna, 8 menyamakan bentuk, memecahkan perhitungan matematika, menggeser, menarik dan mendorong kotak ke tempat yang seharusnya. Sering pula permainan jenis ini merupakan unsur dalam permainan genre petualangan maupun edukasi. Contoh: Tetris, Bubble Party.

7. *Sport game* (Olahraga).

Game ini merupakan adaptasi dari kehidupan nyata, pemain *game* jenis sport membutuhkan kelincahan dan juga strategi dalam memainkannya. *Game* ini berupa kompetisi antara dua pemain atau lebih, dimana pemain dapat melakukan secara individual atau tim. Contoh *game* tipe ini antara lain, PES (pro evolution soccer), Mario Kart, tenis.

8. *RPG* (Role Playing *Game*).

RPG (*Role Playing Game*) yang artinya bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam *game*, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan. Contoh: FinalFantasy, DungeonHunter, Ragnarok.

9. *Education* (edukasi).

Game edukasi merupakan paket software yang menciptakan kemampuan pada lingkungan *game* yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur *game* secara teliti untuk mengembangkan kemampuannya. Developer yang membuatnya, harus memperhitungkan berbagai hal agar *game* ini benarbenar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan design visual ataupun animasinya.

1.2.6 Unsur Unsur *Game*

Game terdiri dari berbagai unsur dasar yang dipadukan sehingga menciptakan suatu *game* yang utuh, adapun komponen-komponen *game* antara lain:

1. Fitur,

Fitur merupakan suatu hal yang menggambarkan *game* kedalam bentuk bentuk yang dapat dilihat maupun dirasakan.

2. *Gameplay*

Gameplay merupakan cara kerja suatu *game*, dimana fitur-fitur yang ada akan membentuk suatu *gameplay*.

3. Interface

Interface merupakan semua bentuk tampilan yang ada pada sebuah *game*. Interface yang baik akan membuat pemainnya betah dan tidak bosan dalam memainkan *game*.

4. Rules

Rules merupakan sekumpulan aturan yang ada dalam memainkan sebuah *game*.

5. Desain level

Desain level merupakan tingkat kesulitan yang menggambarkan jalan cerita pada sebuah *game*.

1.2.7 RPG Maker MV

Menurut (Wicaksono, 2013) game RPG adalah sebuah permainan dimana para pemain memainkan peran untuk merajut sebuah cerita. Para pemain memiliki sebuah karakteristik yang terdapat pada tokoh di game tersebut. *Game* RPG berbeda dengan jenis *game* lainnya seperti catur yang merupakan *strategy game* atau Mario Bros yang berupa *action game*. Sebuah *game* RPG memiliki beberapa karakteristik tertentu seperti, pemain harus membunuh beberapa monster agar tokohnya menjadi kuat, pemain dapat menginput nama dari tokoh yang sedang dimainkan, menentukan peraturan pertempuran tokohnya dan pertinjuannya di dunia khayalan yang akan digunakan (sejarah, geografi, nama raja, dan lain-lain) . RPG Maker merupakan sebuah program untuk membuat *game* RPG. Proses pembuatan dibantu tools yang sudah tersedia didalam sebuah program. Program ini dibuat oleh grup Jepang bernama ASCII, kemudian diteruskan oleh Enterbrain. RPG Maker telah diluncurkan di Asia, Amerika Utara, Eropa, dan Australia.

Menurut (Wahana, 2014) RPG Maker MV merupakan program untuk membuat *game* RPG 2 dimensi. Program ini yang dibuat untuk dapat digunakan siapa saja. Dibawah ini beberapa macam komponen utama pada RMVA:

1. Javascript

Menurut (Steven, 2007) JavaScript digunakan oleh milyaran aplikasi web yang ada untuk desain, validasi data, deteksi browser, membuat cookie, dan lain-lain. JavaScript merupakan scripting language yang paling populer pada pemrograman web atau internet. JavaScript dapat bekerja atau dieksekusi pada berbagai jenis browser

seperti Internet Explorer, Firefox, Netscape, dan Opera. JavaScript didesain untuk menambah interaktifitas dari sebuah aplikasi web. Kode JavaScript biasanya disisipkan pada sebuah halaman HTML atau dapat juga disimpan pada file terpisah dan dipanggil dari sebuah halaman HTML yang membutuhkannya.

2. Database

Database merupakan tempat menyimpan berbagai macam komponen pada *game*. Database dibedakan dalam bentuk tab. Setiap tab memiliki sebuah katagori. Tab yang terdapat pada database antara lain:

- a. Actor. Tab ini berisi data untuk membuat karakter.
- b. Classes. Tab ini mengatur kelas yang digunakan pada karakter.
- c. Skill. Animation, State. Tab ini mengatur keahlian yang dimiliki oleh karakter, serta animasi dan efek dari keahlian tersebut.
- d. Item, Weapon, Armor. Tab ini mengatur barang, senjata, dan perlengkapan karakter.
- e. System dan Term. Tab ini berisi standar pengaturan, seperti Sound Effect dan menu pada *game*.
- f. Commont Event dan Tileset. Tab ini mengatur event yang terjadi di dalam *game* dan aksesoris pendukung pembuatan *game*.

3. Map Editor

Map Editor merupakan tempat untuk membuat peta yang akan dijelajah oleh karakter. Peta dapat dibuat dengan memilih tileset yang ada di kiri map editor. Pada map editor developer dapat menentukan daerah mana saja yang dapat di jelajahi

4. Event

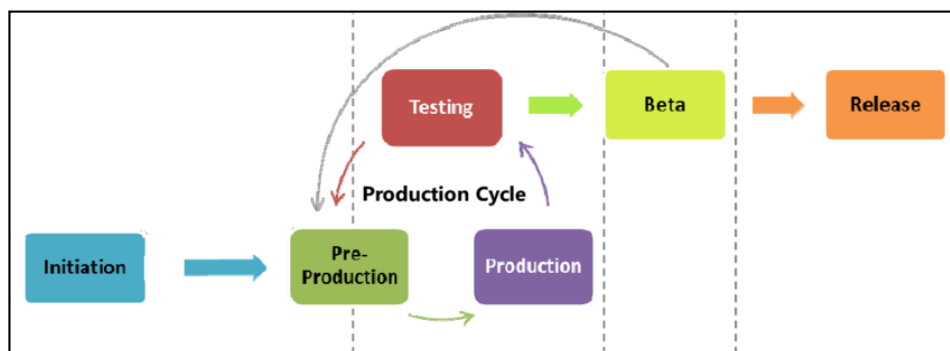
Event memiliki peran yang sangat penting seperti memberikan sebuah interaksi karakter dengan NPC, kendaraan, Perpindahan map, dan lain sebagainya. Selain itu event mengatur aliran adegan yang ada dalam *game* seperti perubahan percakapan aktor dengan beberapa NPC.

1.2.7. Android

Android adalah system perangkat operasi berbasis Linux, dirancang untuk berbagai perangkat layar sentuh, seperti telepon pintar atau yang lebih dikenal dengan sebutan *smartphone*. Android ini dikembangkan pertama kali oleh AndroidInc (Angga, 2019). Android sebagai perangkat yang bergerak pada sistem operasi pada telepon seluler berbasis Linux sangat berkembang pesat saat ini. Android dapat diartikan juga suatu perangkat lunak yang di desain sebagai *System* operasi untuk sebuah *smartphone* yang berjalan dengan *System* operasi untuk sebuah *smartphone* yang berjalan dengan *System* operasi (OS) berbasis Linux.

1.2.8. Game Development Life Cycle (GDLC)

GDLC adalah metode pembangunan dan pengembangan *game*, dimana di dalamnya ada tahapan *inititation*, *pre-production*, *production*, *testing*, *beta*, *release*.



Gambar 2.1 Game Development Life Cycle
Sumber : Krisdiawan, R.A. 2018.

a. *Inititation*

Langkah awal yang harus dilakukan dalam membuat sebuah *Game* adalah dengan membuat konsep kasar *game* yang akan dibuat. Hasil dari langkah ini adalah konsep permainan dan deskripsi permainan yang dijabarkan dengan sederhana.

b. *Pre-production*

Pre-Production atau Pra-produksi adalah salah satu fase utama dan terpenting dalam siklus produksi. Pra-produksi melibatkan desain dan pembuatan *game*. Pada desain *game* menentukan *genre game*, *gameplay*, mekanik, alur cerita, karakter, dan tantangan.

c. *Production*

Produksi adalah proses inti yang membahas seputar pembuatan aset, pembuatan program, dan integrasi kedua elemen. Terkait prototipe dalam fase ini adalah detail formal dan perbaikan. Detail Formal adalah struktur yang disempurnakan dengan lebih banyak mekanik dan aset lengkap. Produksi merupakan kegiatan yang terkait dengan pembuatan dan penyempurnaan detail formal menyeimbangkan, menambahkan fitur yang baru, meningkatkan kinerja, dan memperbaiki bug (terkait dengan fungsional dan penyelesaian internal kriteria kualitas).

Penyeimbangan *game* berarti penyesuaian terkait dengan kesulitan *game* untuk membuat *game* tersebut memiliki kesulitan yang sesuai. Perbaikan adalah prototipe lengkap yang merupakan subjek pemolesan *game*. Kriteria kualitas pada tahap ini terkait permainan harus menyenangkan dan mudah diakses. Kegiatan selama penyempurnaan diarahkan untuk membuat *game* lebih menyenangkan, menantang, dan lebih mudah dipahami. Hanya perubahan kecil diperbolehkan dalam fase ini.

d. *Testing*

Pengujian dalam konteks ini berarti pengujian internal yang dilakukan untuk menguji fungsi operasional dan kemampuan bermain *game*. Metode pengujian khusus untuk setiap tahap prototipe. Pengujian Detail Formal dilakukan dengan menggunakan *playtest* untuk menilai fungsionalitas fitur dan kesulitan permainan (terkait dengan keseimbangan).

Metode untuk menguji kriteria kualitas fungsional melalui fitur *playtesting* untuk menguji kualitas lengkap secara internal bisa dilakukan melalui *playtesting* secara bersamaan dengan uji fungsionalitas. Saat penguji menemukan bug, celah, atau jalan buntu selama *playtesting*, penyebabnya dan skenario untuk mereproduksi kesalahan yang diperlukan didokumentasikan dan dianalisis. Untuk menguji keseimbangan kriteria kualitas, *playtesting* dengan beberapa perbedaan perbaikan digunakan untuk mengkategorikan apakah suatu perbaikan terlalu sulit, terlalu mudah, atau tidak diperlukan.

Pengujian Perbaikan berhubungan dengan kualitas kesenangan dan kriteria kualitas aksesibilitas. Dalam pengujian perbaikan, kesenangan diuji melalui tes bermain dan umpan balik langsung dari sesama pengembang, baik itu membosankan, membuat frustrasi, menantang, dll. Aksesibilitas dapat diuji melalui mengamati perilaku penguji. Jika penguji merasa kesulitan untuk memainkan dan memahami permainan, itu berarti bahwa *game* belum cukup mudah diakses. Output dari pengujian adalah laporan *bug*, permintaan perubahan, dan keputusan pembangunan. Hasilnya akan menentukan apakah sudah waktunya untuk maju ke fase berikutnya (*Beta*) atau mengulangi siklus produksi.

e. *Beta*

Beta adalah fase untuk melakukan pengujian oleh pihak ketiga atau eksternal yang disebut pengujian beta atau *beta testing*. Pengujian beta masih menggunakan metode pengujian yang sama dengan metode pengujian sebelumnya, karena prototipe terkait dalam pengujian beta merupakan detail formal dan penyempurnaan. Metode pemilihan penguji hadir dalam dua jenis: beta tertutup dan beta terbuka. Beta tertutup hanya mengizinkan individu yang diundang untuk menjadi peserta, sedangkan beta terbuka memungkinkan siapa saja yang mendaftar menjadi peserta. Kriteria kualitas dalam versi beta terkait erat dengan tahap prototipe.

Dalam pengujian detail formal, penguji diminta untuk menemukan bug (terkait dengan kriteria kualitas fungsional dan lengkap secara internal). Dalam uji perbaikan, penguji diberikan kebebasan lebih untuk menikmati permainan, karena tujuannya lebih diarahkan untuk mendapatkan umpan balik (terkait kriteria kualitas kesenangan dan aksesibilitas). Keluaran dari pengujian beta adalah laporan bug dan masukan pengguna. Sesi beta ditutup terutama karena 2 alasan, baik istilah beta berakhir atau jumlah penguji beta yang ditentukan telah memberikan laporan pengujian mereka. Dari sini, dapat mengarah ke siklus produksi lagi untuk menyempurnakan produk atau terus merilis *game* jika hasilnya memuaskan.

f. *Release*

Ini adalah fase dimana pengembangan *game* telah mencapai tahap akhir dan siap dirilis ke publik. Rilis melibatkan peluncuran produk, dokumentasi proyek, berbagi pengetahuan, post-mortem, dan perencanaan untuk pemeliharaan dan perluasan *game*.





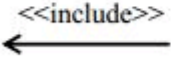
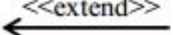
1.2.8 *Unified Modeling Language (UML)*

Menurut M Teguh Prihandoyo (2018), *Unified modeling language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan pembuatan *software* yang berorientasikan pada objek. UML merupakan standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pembangunan sebuah sistem, yaitu:

1. *Use Case*

Use Case adalah gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara actor dan sistem (Teguh, 2018). *Use case* memiliki *actor* yang merupakan sebuah gambaran entitas dari sebuah sistem yang melakukan pekerjaan sistem. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor use case.

Tabel 2.2 *Use Case*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi


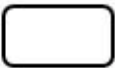




2. *Activity Diagram*

Activity diagram atau diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Dalam aktivitas juga banyak digunakan untuk mendefinisikan hal – hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem / user interface dimana aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

Tabel 2.3 *Activity Diagram*




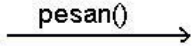
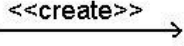
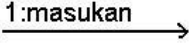
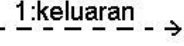
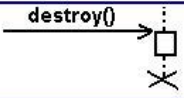
Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

3. *Sequence Diagram*

sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek juga interaksi antara objek. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Berikut komponen-komponen yang ada pada sequence diagram :

Tabel 2.4 *Sequence Diagram*

<p>aktor</p>  atau nama_aktor	<ul style="list-style-type: none"> ● orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari system. ● Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan. ● Ditempatkan di bagian atas diagram.
<p>objek</p> objek:kelas	<p>Sebuah objek:</p> <ul style="list-style-type: none"> ● Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan. ● Ditempatkan di bagian atas diagram.
<p>Garis hidup objek</p> 	<ul style="list-style-type: none"> ● Menandakan kehidupan obyek selama urutan. ● diakhiri tanda X pada titik di mana kelas tidak lagi berinteraksi.
<p>Objek sedang aktif berinteraksi</p> 	<p>Fokus kontrol:</p> <ul style="list-style-type: none"> ● Adalah persegi panjang yang sempit panjang ditempatkan di atas sebuah garis hidup. ● Menandakan ketika suatu objek mengirim atau menerima pesan.
<p>pesan</p> 	<p>objek mengirim satu pesan ke objek lainya</p>
<p><<create>></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>1:masukan</p> 	<p>menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim</p>
<p>1:keluaran</p> 	<p>objek/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>destroy()</p> 	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>