

BAB II

TINJAUAN PUSTAKA

2.1 E-Discussion

Diskusi merupakan pertemuan ilmiah untuk bertukar pikiran mengenai suatu masalah yang dilakukan oleh sekelompok orang. Diskusi merupakan suatu metode atau cara mengajar dengan cara memecahkan masalah yang dihadapi, baik atau lebih, dimana setiap peserta diskusi berhak mengajukan argumentasinya untuk memperkuat pendapatnya. (Primadasa et al., 2020)

E-Discussion adalah singkatan dari "Electronic Discussion" atau "Electronic Discussion Forum." Ini adalah bentuk komunikasi online di mana orang-orang dapat berpartisipasi dalam percakapan, berbagi informasi, dan bertukar gagasan melalui internet. E-Discussion biasanya terjadi dalam berbagai platform online seperti forum, milis (mailing lists), grup diskusi, jejaring sosial, atau aplikasi pesan instan

E-Discussion dapat digunakan untuk berbagai tujuan, termasuk berbagi pengetahuan, memecahkan masalah, berdiskusi tentang topik tertentu, atau bahkan untuk tujuan bisnis seperti mendiskusikan proyek, produk, atau layanan. Mereka dapat bersifat terbuka untuk publik atau dibatasi hanya untuk anggota tertentu. E-Discussion juga memungkinkan partisipasi dari berbagai lokasi geografis, sehingga memungkinkan kolaborasi dan pertukaran ide yang lebih luas.

E-Discussion telah menjadi salah satu cara yang populer untuk berkomunikasi dan berinteraksi secara online, dan banyak organisasi dan komunitas memanfaatkannya sebagai alat penting dalam berbagi informasi dan menjaga keterlibatan anggota.

2.2 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah editor kode sumber terkemuka yang dikembangkan oleh Microsoft. Ini adalah perangkat lunak sumber terbuka dan tersedia secara gratis. Visual Studio Code dirancang untuk memenuhi kebutuhan pengembang perangkat lunak dalam mengedit kode, mengelola proyek, dan berkolaborasi dengan tim pengembangan. Berikut beberapa fitur dan karakteristik utama dari Visual Studio Code:

1. Ringan: VS Code memiliki jejak memori yang kecil dan berjalan cepat, sehingga cocok untuk berbagai platform dan konfigurasi komputer.
2. Integrasi Luas: Ini menyediakan integrasi yang kuat dengan banyak bahasa pemrograman dan kerangka kerja, termasuk bahasa pemrograman populer seperti JavaScript, Python, C++, dan banyak lagi.
3. Ekstensi: Anda dapat memperluas fungsionalitas VS Code dengan mudah menggunakan ekstensi. Ada ribuan ekstensi yang tersedia di pasar VS Code untuk menyesuaikan pengalaman pengembangan Anda.
4. Editor Kode yang Kuat: Ini memiliki fitur-fitur seperti penyorotan sintaksis, otomatisasi penyelesaian kode, pemadanan tanda kurung, dan banyak lagi yang membuat pengeditan kode menjadi lebih efisien.
5. Integrasi Git: VS Code memiliki integrasi yang kuat dengan sistem kontrol versi Git, memungkinkan Anda untuk mengelola proyek dengan mudah.
6. Debugging: VS Code mendukung debugging kode dengan berbagai bahasa pemrograman dan dapat diintegrasikan dengan debugger eksternal.
7. Terminal Terintegrasi: Anda dapat menjalankan terminal langsung dari VS Code, yang berguna untuk menjalankan perintah atau skrip langsung dari editor.
8. Kolaborasi dan Ekstensi Live Share: Fitur Live Share memungkinkan Anda untuk berkolaborasi secara real-time dengan anggota tim Anda, yang sangat berguna dalam pengembangan bersama.
9. Cross-Platform: Tersedia untuk Windows, macOS, dan Linux, sehingga Anda dapat menggunakannya di berbagai sistem operasi.

Visual Studio Code telah menjadi pilihan populer di kalangan pengembang perangkat lunak karena fleksibilitas, performa, dan dukungan

komunitas yang kuat. Anda dapat mengunduh dan menginstal Visual Studio Code dari situs web resminya dan menyesuaikannya sesuai kebutuhan pengembangan Anda.

Adapun bahasa pemrograman yang di gunakan dalam pengembangan aplikasi E-Discussion pada VS Code sebagai berikut :

1. *React JS*

ReactJS, atau lebih umum dikenal sebagai React, adalah sebuah pustaka (library) JavaScript yang dikembangkan oleh Facebook. React digunakan untuk membangun antarmuka pengguna (UI) yang dinamis dan responsif untuk aplikasi web. Ini adalah salah satu dari sekian banyak pustaka yang populer untuk pengembangan aplikasi web yang mengikuti model pemrograman berbasis komponen.

Berikut adalah beberapa fitur dan konsep utama yang terkait dengan React:

1. **Komponen:** React berfokus pada pengembangan UI berbasis komponen. Komponen adalah unit terkecil yang dapat digunakan kembali dalam pembangunan aplikasi. Setiap komponen dapat memiliki logika dan tampilan sendiri.
2. **Virtual DOM:** React menggunakan konsep Virtual DOM untuk meningkatkan kinerja. Ketika data dalam aplikasi berubah, React tidak langsung memperbarui DOM aktual. Sebaliknya, ia membangun DOM virtual terlebih dahulu dan membandingkannya dengan DOM aktual untuk menentukan perubahan apa yang harus diterapkan, sehingga meminimalkan manipulasi langsung terhadap DOM yang mahal secara komputasi.
3. **JSX (JavaScript XML):** React memungkinkan Anda untuk menulis komponen dengan menggunakan sintaksis yang mirip dengan HTML, yang disebut JSX. JSX memungkinkan penciptaan elemen-elemen UI dengan nyaman dalam kode JavaScript.
4. **Reaktivitas:** React memungkinkan Anda untuk membuat UI yang merespons perubahan data secara otomatis. Ketika data berubah, tampilan akan diperbarui sesuai dengan perubahan tersebut.

5. Unidirectional Data Flow: React mengikuti pola aliran data satu arah (unidirectional data flow). Data mengalir dari komponen induk ke komponen anak, membuat pengelolaan state lebih mudah dipahami.
6. Komunitas yang Besar: React memiliki komunitas yang besar dan aktif, yang berarti Anda dapat menemukan banyak tutorial, paket, dan sumber daya online untuk membantu dalam pengembangan aplikasi Anda.
7. Kompatibilitas: React dapat digunakan dengan berbagai teknologi dan perpustakaan, seperti Redux untuk manajemen state, React Router untuk penanganan navigasi, dan banyak lagi.
8. Pengembangan Berbasis Komponen: React mendorong pendekatan pengembangan berbasis komponen, yang memungkinkan pembagian tugas dan pengembangan yang lebih terorganisir dalam tim.

React sangat populer dalam pengembangan aplikasi web modern, dan digunakan oleh banyak perusahaan besar untuk mengembangkan antarmuka pengguna yang canggih dan dinamis. React Native, yang dibangun berdasarkan React, juga memungkinkan pengembangan aplikasi mobile lintas platform menggunakan JavaScript dan React.

2. *Flutter*

Flutter adalah sebuah framework pengembangan perangkat lunak yang bersifat open source yang dikembangkan oleh Google. Framework ini digunakan untuk membangun aplikasi dengan antarmuka pengguna (UI) yang kaya dan ekspresif. Salah satu fitur utama Flutter adalah kemampuannya untuk membangun aplikasi natively kompilasi yang dapat dijalankan di berbagai platform seperti Android, iOS, web, dan desktop dengan menggunakan kode sumber yang sama.

Berikut adalah beberapa konsep dan fitur kunci yang perlu Anda ketahui tentang Flutter:

1. Bahasa Pemrograman Dart: Flutter menggunakan bahasa pemrograman Dart, yang juga dikembangkan oleh Google. Dart memiliki sintaksis yang mudah dipahami dan dieksekusi dengan cepat, sehingga cocok untuk pengembangan aplikasi mobile.

2. **Widget:** Flutter adalah framework berbasis widget, yang berarti Anda membangun UI Anda dengan menggabungkan berbagai widget. Flutter menyediakan berbagai macam widget yang dapat disesuaikan sesuai kebutuhan Anda.
3. **Kode Sumber Tunggal:** Dengan Flutter, Anda dapat menulis kode sumber satu kali dan menjalankannya di berbagai platform. Ini memungkinkan pengembangan lintas platform yang efisien.
4. **Hot Reload:** Fitur "Hot Reload" Flutter memungkinkan pengembang untuk mengubah kode aplikasi mereka dan melihat perubahan tersebut secara langsung tanpa harus mengulangi proses kompilasi, sehingga mempercepat siklus pengembangan.
5. **Kinerja Natif:** Aplikasi yang dibangun dengan Flutter dikompilasi menjadi kode native, yang memberikan kinerja yang mendekati aplikasi native murni pada setiap platform.
6. **Ekosistem Luas:** Flutter memiliki ekosistem yang berkembang dengan berbagai paket dan plugin yang dapat digunakan untuk memperluas fungsionalitas aplikasi Anda.
7. **Open Source:** Flutter adalah proyek open source, yang berarti memiliki komunitas pengembang yang aktif dan berkontribusi untuk pertumbuhannya.
8. **Widget Material Design dan Cupertino:** Flutter menyediakan widget yang dirancang sesuai pedoman desain Material Design dari Google dan Cupertino dari Apple, memungkinkan Anda untuk membuat aplikasi yang sesuai dengan prinsip desain yang diterapkan oleh kedua perusahaan tersebut.
9. **Pengembangan Lintas Platform:** Anda dapat mengembangkan aplikasi untuk berbagai platform menggunakan kode sumber yang sama, yang dapat menghemat waktu dan usaha dalam menciptakan dan menjaga aplikasi untuk berbagai platform.
10. **Dukungan Web dan Desktop:** Flutter dapat digunakan untuk membangun aplikasi web dan desktop juga, sehingga Anda dapat

mengembangkan aplikasi untuk berbagai platform dengan teknologi yang sama.

Flutter telah mendapatkan popularitas di komunitas pengembang aplikasi mobile karena fleksibilitas dan fitur yang ramah pengembang. Ini adalah pilihan yang kuat bagi mereka yang ingin membuat aplikasi lintas platform dengan fokus pada desain dan kinerja.

3. *GoLang*

Go (biasa disebut Golang) adalah bahasa pemrograman yang dikembangkan oleh Google. Go adalah bahasa pemrograman open source yang didesain untuk memudahkan pengembangan perangkat lunak yang efisien, cepat, dan andal. Berikut adalah beberapa karakteristik utama dari Go:

1. Sederhana dan Mudah Dipahami: Go memiliki sintaksis yang sederhana dan mudah dipahami, sehingga mudah bagi pengembang untuk mempelajarinya dan menulis kode dengan cepat.
2. Efisien dan Cepat: Go diketahui memiliki kinerja yang sangat baik. Bahasa ini dikompilasi menjadi kode mesin yang berjalan sangat cepat, sehingga cocok untuk pengembangan aplikasi berkinerja tinggi.
3. Pemrograman Paralel dan Konkuren: Go memiliki dukungan bawaan untuk pemrograman paralel dan konkuren. Ini membuatnya sangat cocok untuk pengembangan aplikasi yang membutuhkan kinerja tinggi dan tugas-tugas yang berjalan secara bersamaan.
4. Manajemen Memori Otomatis: Go memiliki pengelolaan memori otomatis (garbage collection), yang membebaskan pengembang dari sebagian besar tugas pemrograman rendah tingkat seperti alokasi dan dealokasi memori.
5. Paket Standar yang Kaya: Go dilengkapi dengan paket standar yang kaya, termasuk dukungan untuk HTTP, pemrosesan data, enkripsi, dan banyak lagi. Ini memungkinkan pengembang untuk membangun aplikasi yang kompleks dengan mudah.

6. Kode Sumber Tunggal: Go mendorong praktik pengembangan dengan kode sumber tunggal, yang membuatnya mudah untuk mengelola proyek-proyek yang lebih besar.
7. Komunitas yang Aktif: Go memiliki komunitas pengembang yang aktif dan berkembang, sehingga Anda dapat menemukan banyak sumber daya, paket, dan dukungan komunitas.
8. Kompilasi yang Mandiri: Go dikompilasi menjadi biner mandiri, yang berarti Anda tidak perlu menginstal dependensi eksternal di lingkungan tujuan untuk menjalankan aplikasi Go.
9. Dukungan Platform yang Luas: Go mendukung berbagai platform, termasuk Windows, macOS, Linux, dan banyak arsitektur perangkat keras lainnya.
10. Pembaruan yang Aman: Go didesain dengan keselamatan dalam pikiran, termasuk pengelolaan memori yang aman dan tidak rentan terhadap sebagian besar kerentanan umum.

Go sering digunakan dalam pengembangan berbagai jenis perangkat lunak, termasuk aplikasi server, mikro layanan, perangkat lunak berkinerja tinggi, dan banyak lagi. Go telah menjadi bahasa pemrograman yang sangat populer, terutama dalam pengembangan web dan cloud, serta dalam proyek-proyek yang membutuhkan efisiensi dan konkurensi.

2.3 Figma

Figma adalah sebuah platform desain kolaboratif berbasis web yang digunakan untuk membuat prototipe, desain antarmuka pengguna (UI), dan pengembangan desain grafis. Figma dirancang untuk memudahkan kolaborasi tim dalam proses desain, dan platform ini memiliki sejumlah fitur yang membedakannya dari perangkat lunak desain tradisional. Berikut beberapa aspek penting tentang Figma:

1. Kolaborasi Real-Time: Salah satu fitur utama Figma adalah kemampuan untuk berkolaborasi secara waktu nyata. Beberapa pengguna dapat bekerja pada proyek yang sama secara bersamaan, melihat perubahan yang terjadi secara langsung, dan berkomunikasi melalui fitur komentar.

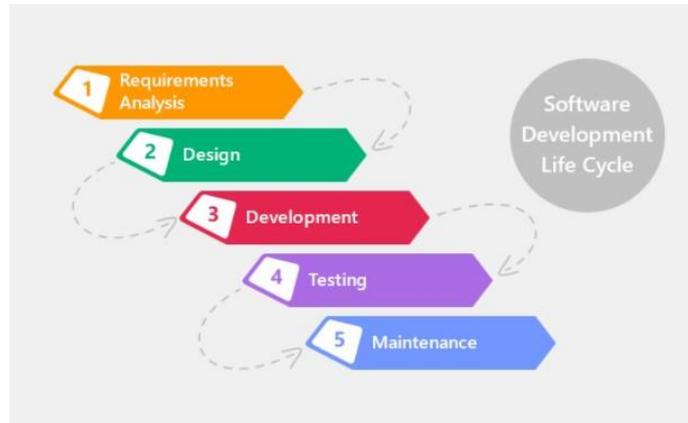
2. Berbasis Cloud: Figma sepenuhnya berbasis cloud, yang berarti desain Anda tersimpan secara online. Ini memungkinkan akses ke proyek dari berbagai perangkat dan memudahkan berbagi proyek dengan tim dan pemangku kepentingan.
3. Platform Agnostik: Anda dapat mengakses Figma dari berbagai platform, termasuk Windows, macOS, dan Linux, serta melalui browser web tanpa perlu instal perangkat lunak tambahan.
4. Prototipe Interaktif: Figma memungkinkan Anda membuat prototipe antarmuka pengguna interaktif, yang memungkinkan Anda menguji alur kerja dan fungsi aplikasi sebelum mengembangkan kode aktual.
5. Desain Berbasis Komponen: Anda dapat membuat dan menggunakan komponen yang dapat digunakan kembali dalam desain Anda. Ini membantu menjaga konsistensi desain dan mempercepat proses pembuatan UI.
6. Vector Editing: Figma memiliki fitur pengeditan vektor yang kuat, yang memungkinkan Anda membuat ikon, ilustrasi, dan elemen desain lainnya dengan presisi.
7. Plugins: Figma memiliki ekosistem plugin yang kaya, yang memungkinkan Anda memperluas fungsionalitasnya dengan berbagai ekstensi dan alat tambahan.
8. Integrasi: Figma dapat terintegrasi dengan berbagai alat pengembangan dan desain lainnya, seperti Slack, Trello, GitHub, dan lainnya, untuk memfasilitasi aliran kerja yang lebih mulus.

Figma digunakan oleh desainer UI/UX, pengembang web, tim produk, dan banyak individu dan tim lain yang terlibat dalam proses desain dan pengembangan produk. Keterbukaan platform, kolaborasi waktu nyata, dan fitur-fitur lainnya menjadikan Figma salah satu alat desain yang populer dalam industri desain dan pengembangan perangkat lunak.

2.4 Waterfall

Metode *Waterfall* adalah salah satu pendekatan yang digunakan dalam pengembangan perangkat lunak (*software development*) untuk mengelola proyek-proyek pengembangan perangkat lunak. Metode ini sering disebut juga sebagai

model pengembangan berjenjang karena menggambarkan proses pengembangan perangkat lunak sebagai serangkaian tahap yang harus diselesaikan secara berurutan, mirip dengan aliran air terjun (*waterfall*).



Gambar 2. 1 Metode Waterfall

Berikut adalah tahapan-tahapan dalam metode *Waterfall*:

1. Analisis dan Perencanaan (*Requirements*):
 - Identifikasi dan analisis kebutuhan.
 - Merencanakan proyek, termasuk penjadwalan, alokasi sumber daya, dan biaya.
2. Desain (*Design*):
 - Merancang arsitektur sistem.
 - Menentukan rancangan perangkat lunak secara keseluruhan.
3. Implementasi (*Implementation*):
 - Proses pengembangan perangkat lunak sesuai dengan rancangan yang telah dibuat.
4. Pengujian (*Testing*):
 - Menguji perangkat lunak untuk memastikan bahwa itu memenuhi kebutuhan dan berfungsi dengan benar.
 - Mengidentifikasi dan memperbaiki kesalahan atau bug.
5. Integrasi dan Verifikasi (*Integration and Verification*):
 - Mengintegrasikan berbagai komponen perangkat lunak.
 - Melakukan verifikasi apakah sistem bekerja seperti yang diharapkan.
6. Pemeliharaan (*Maintenance*):

- Merawat, mengelola, dan memperbarui perangkat lunak setelah implementasi.

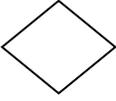
Metode *Waterfall* memiliki beberapa kelebihan, seperti jelasnya alur kerja, dokumentasi yang kuat, dan kemampuan untuk mengelola proyek dengan baik. Metode *Waterfall* masih digunakan dalam proyek-proyek tertentu di mana persyaratan tetap stabil dan perencanaan yang ketat diperlukan.

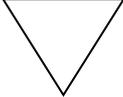
2.5 Flowchart

Flowchart adalah alat visual yang digunakan untuk merepresentasikan urutan langkah-langkah atau aliran proses dalam suatu sistem, tugas, atau prosedur. *Flowchart* adalah diagram yang berisi bentuk-bentuk dan panah-panah yang menghubungkannya untuk menggambarkan urutan logis dari tindakan atau keputusan dalam suatu proses. *Flowchart* digunakan dalam berbagai bidang, termasuk pengembangan perangkat lunak, ilmu komputer, manufaktur, bisnis, dan banyak lagi.

Berikut Simbol-simbol yang dipergunakan dalam pembuatan *Flowchart*:

TABEL 2. 1 Simbol pada *Flowchart*

No.	Simbol	Keterangan
1.		Simbol <i>Offline Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		Simbol <i>Manual</i> Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
3.		Simbol <i>Decision/Logika</i> Untuk menunjukkan suatu kondisi tertentu yang menghasilkan dua kemungkinan jawaban, ya/tidak.

No.	Simbol	Keterangan
4.		<p>Simbol <i>Predefined Proses</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
5.		<p>Simbol <i>Terminal</i> Untuk menyatakan permulaan atau akhir suatu program.</p>
7.		<p>Simbol <i>Off-Line Storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.</p>
8.		<p>Simbol <i>Manual Input</i> Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>.</p>
9.		<p>Simbol <i>Input-Output</i> Untuk menyatakan proses input dan <i>output</i> tanpa tergantung dengan jenis peralatanya.</p>
10.		<p>Simbol <i>Document</i> Untuk mencetak laporan ke printer.</p>
11.		<p>Simbol <i>Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.</p>

2.6 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak dan sistem berbasis informasi. UML digunakan secara luas dalam pengembangan perangkat lunak, rekayasa sistem, dan pemodelan proses bisnis. Ini adalah alat yang sangat berguna dalam mengkomunikasikan desain

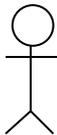
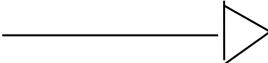
sistem dan pemahaman tentang sistem antara para pengembang, arsitek perangkat lunak, pemangku kepentingan, dan tim yang terlibat dalam pengembangan perangkat lunak.

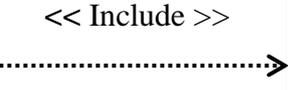
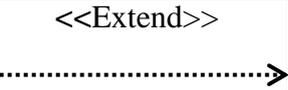
UML telah menjadi alat standar yang umum digunakan dalam pengembangan perangkat lunak dan pemodelan sistem, membantu dalam merancang, mengkomunikasikan, dan mendokumentasikan berbagai aspek dari proyek teknologi informasi.

2.6.1 Use Case Diagram

Diagram use case adalah representasi visual dari fungsionalitas sistem atau aplikasi perangkat lunak dari perspektif penggunanya. Diagram ini adalah bagian dari *Unified Modeling Language* (UML) dan digunakan untuk mengilustrasikan interaksi antara berbagai aktor (pengguna atau sistem eksternal) dan sistem itu sendiri melalui berbagai kasus penggunaan (*use case*).

TABEL 2. 2 Usecase Diagram

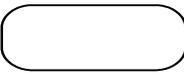
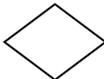
No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang

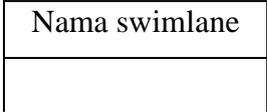
No	Simbol	Deskripsi
		lebih umum
5.		Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

2.6.2 Activity Diagram

Diagram aktivitas (*activity diagram*) adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan aliran kerja (*workflow*) dari suatu proses atau aktivitas dalam sistem. Diagram ini membantu dalam memodelkan dan memvisualisasikan bagaimana suatu aktivitas atau proses berlangsung dari awal hingga akhir, dengan menunjukkan aktivitas, keputusan, kondisi, dan aliran data yang terlibat.

TABEL 2. 3 Simbol Activity Diagram

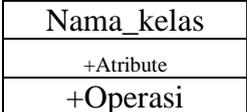
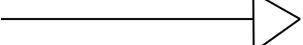
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

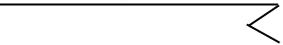
No.	Simbol	Keterangan
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.6.3 Class Diagram

Diagram kelas (*class diagram*) adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan struktur kelas, objek, dan hubungan antara kelas dalam suatu sistem perangkat lunak. Diagram kelas menggambarkan entitas (kelas) dalam sistem dan bagaimana mereka berinteraksi satu sama lain. Ini adalah alat yang sangat penting dalam pemrograman berorientasi objek dan analisis perangkat lunak.

TABEL 2. 4 Simbol Class Diagram

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>symbol</i>
4.	Asosiasi Berarah / <i>Digunakan Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>symbol</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

No.	Simbol	Deskripsi
6.	Ketergantungan / dependency 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.7 Peneliti Terdahulu

Pada penelitian ini terdapat beberapa acuan atau referensi dari penelitian terdahulu.

TABEL 2. 5 Peneliti Terdahulu

No	Judul	Penulis & Tahun	Masalah	Hasil
1	Penerapan E-Discussion pada SMA Negeri 1 Tanah Merah Berbasis WEB	Willyansah, Jurnal Informatika, Manajemen dan Komputer, Vol.12 No.1, Mei 2020	Permasalahan yang di dapat dari penelitian ini adalah untuk mempermudah guru dan siswa untuk saling berkomunikasi dimana guru dan siswa dapat berkomunikasi tanpa ada batas ruang dan waktu .	Penerapan E-Diskusi pada SMA NEGERI 1 Tanah Merah
2	Rancang Bangun Sistem E-Discussion Untuk Mahasiswa Kota Lubuklinggau	Yogi Primadasa, Hengki Juliansa, December 2020	Permasalahan yang di hasilkan dari penelitian ini adalah untuk membantu mahasiswa melakukan komonukasi dan belajar dengan mahasiswa lain di	Sistem E-Discussion ini membantu mahasiswa yang ada di Kota Lubuklinggau dalam berbagi / sharing ilmu pengetahuan seputaran materi perkuliahan

No	Judul	Penulis & Tahun	Masalah	Hasil
			perguruan tinggi.	yang mereka jalani.
3	Online Discussion Forum as a Tool for Interactive Learning and Communication	Edeh Michael Onyema, Edeh Chinecherem Deborah, Alhuseen Omar Alsayed, Quadri Noorulhasan Naveed, Sumaya Sanober, November 2019	Kurang efektif dan efisien dalam pembelajaran online, Forum diskusi online menghadirkan peluang untuk pembelajaran interaktif, pembelajaran berbasis inkuiri, dan siswa perolehan dan penciptaan pengetahuan. Diskusi Daring forum yang disajikan dalam penelitian ini terbukti sangat efektif pembelajaran interaktif dan komunikasi antar siswa dan guru	Forum Diskusi Online Sebagai Sarana Pembelajaran dan Komunikasi
4	RANCANG BANGUN APLIKASI ONLINE FOCUS GROUP DISCUSSION	Natasha Bunga Pertiwi, 2016	Tool diskusi belum dapat sepenuhnya memberi kepuasan pada user dalam hal diskusi, seperti pertanyaan mereka terkadang tidak benar-benar terjawab atau bahkan terlalu banyak jawaban	Terbentuknya Aplikasi Online Focus Group Discussion

No	Judul	Penulis & Tahun	Masalah	Hasil
			sehingga tidak tahu jawaban mana yang paling tepat.	
5	Pengembangan Aplikasi Group Discussion Berdasarkan Multiple Intelligences	Bait Syaiful Rijal, Dian Novian, Huzaima Mas'ud, Desember 2022	Praktisi Pendidikan di Indonesia belum banyak yang melibatkan penilaian multiple intelligence dalam metode diskusi	Hasil pengujian pada aplikasi yang dikembangkan menggunakan analisis standar Web-QEM terdiri dari aspek functionality, reliability, dan efficiency. Aspek functionality diuji dan mendapatkan hasil penilaian baik.