

## **BAB II**

### **KAJIAN TEORITIS**

#### **A. Kajian Teoritis**

##### **1. Rancang bangun**

Rancang bangun adalah langkah-langkah atau prosedur yang digunakan untuk mengubah hasil analisis dari suatu sistem ke dalam kode pemrograman, dengan tujuan menjelaskan secara terperinci bagaimana komponen-komponen sistem akan diimplementasikan (Nizar 2021:2). Sedangkan penelitian yang dilakukan oleh Rahmat Gunawan et al., (2021:48), rancang bangun adalah proses menggambarkan, merencanakan, dan membuat susunan atau penyusunan elemen-elemen terpisah ke dalam satu kesatuan yang lengkap dan berfungsi. Setiap tahapan dalam proses rancang bangun saling terkait dan mempengaruhi hasil akhir, sehingga perencanaan dan eksekusi yang baik sangat krusial untuk kesuksesan proyek atau produk tersebut.

Dari kesimpulan di atas dapat dijelaskan bahwa rancang bangun adalah proses yang melibatkan penerjemahan hasil analisa sistem ke dalam bahasa pengkodean untuk menjelaskan secara detail implementasi komponen-komponen sistem. Selain itu, rancang bangun juga mencakup proses menggambarkan, merencanakan, dan menyusun elemen-elemen terpisah menjadi satu kesatuan yang lengkap dan berfungsi. Setiap tahapan dalam proses ini saling terkait dan mempengaruhi hasil akhir, sehingga perencanaan

dan eksekusi yang baik sangat penting untuk kesuksesan proyek atau produk tersebut.

## **2. Media Pembelajaran**

Media pembelajaran adalah alat untuk menyampaikan informasi dari guru kepada siswa sebagai penerima. Jika lingkungan belajar dirancang dengan sistematis, tujuan pembelajaran dapat dicapai secara optimal (Saleh & Syahrudin 2023:6). Sedangkan penelitian yang dilakukan oleh Pagarra H & Syawaludin (2022:11), Media pembelajaran adalah berbagai perangkat yang dimanfaatkan oleh pendidik sebagai sarana untuk menyampaikan materi pelajaran, sehingga dapat diterima oleh siswa dengan tepat dan efektif.

Dari kesimpulan di atas dapat dijelaskan bahwa media pembelajaran memainkan peran penting dalam proses pendidikan sebagai alat atau perangkat untuk menyampaikan informasi dari guru kepada siswa. Penggunaan media pembelajaran yang tepat dan dirancang secara sistematis dapat membantu mencapai tujuan pembelajaran secara optimal dengan memastikan bahwa materi pelajaran diterima oleh siswa dengan tepat dan efektif. Kedua penelitian tersebut menekankan bahwa penggunaan media pembelajaran yang tepat dan terstruktur sangat penting untuk mencapai hasil pembelajaran yang maksimal.

### **3. IPA**

Ilmu Pengetahuan Alam (IPA) adalah pengetahuan yang terbentuk melalui proses mengumpulkan data dari eksperimen, observasi, dan deduksi, yang bertujuan untuk memberikan penjelasan yang dapat diandalkan mengenai fenomena alam (Nana Sutrisna 2022:2860). Sedangkan penelitian yang dilakukan oleh Jazuli Yusup et al., (2021), Ilmu Pengetahuan Alam (IPA) meliputi kumpulan pengetahuan, konsep, dan struktur konseptual. Sebagai proses, IPA digunakan untuk mempelajari objek studi, menemukan, dan mengembangkan produk ilmiah. Di sisi lain, dalam aplikasinya, teori-teori IPA dapat menghasilkan teknologi yang meningkatkan kualitas hidup sehari-hari.

Dari kesimpulan di atas dapat dijelaskan bahwa Ilmu Pengetahuan Alam (IPA) adalah pengetahuan yang terbentuk melalui proses pengumpulan data eksperimen, observasi, dan deduksi guna memberikan penjelasan yang dapat diandalkan mengenai fenomena alam. Sebagai kumpulan pengetahuan, konsep, dan struktur konseptual, IPA digunakan dalam proses mempelajari objek studi, menemukan, dan mengembangkan produk ilmiah. Selain itu, penerapan teori-teori IPA juga dapat menghasilkan teknologi yang memberikan kemudahan dalam kehidupan sehari-hari.

### **4. Tata Surya**

Tata surya adalah sistem yang terdiri dari sebuah bintang yang disebut matahari dan semua objek langit yang mengorbit di sekitarnya. Sistem tata surya ini berada di dalam galaksi yang disebut Bima Sakti. Galaksi sendiri

merupakan kumpulan bintang, di mana bintang adalah objek langit yang memancarkan cahaya sendiri (Ramlawati et al., 2017:5). Sedangkan penelitian yang dilakukan oleh Nuzulia (2023), tata surya adalah sistem yang terdiri dari satu bintang yang dikenal sebagai Matahari dan semua benda langit yang terikat oleh gaya gravitasi dari Matahari tersebut. Di dalamnya termasuk delapan planet yang berputar mengelilingi Matahari dengan orbit elips, lima planet kerdil, 173 satelit alami yang telah teridentifikasi, serta jutaan objek langit lainnya seperti meteor, asteroid, dan komet.

Dari kesimpulan di atas dapat dijelaskan bahwa tata surya adalah sistem yang terdiri dari Matahari sebagai pusatnya, dikelilingi oleh planet, satelit alami, dan objek langit lainnya yang terikat oleh gaya gravitasi Matahari. Salah satu penelitian menekankan bahwa tata surya berada dalam galaksi Bima Sakti. Sementara penelitian lainnya memberikan detail lebih spesifik mengenai struktur dan komponen-komponen tata surya.

## **5. Android**

Menurut Omer et al., (2021:59) *android* merupakan salah satu sistem operasi yang paling banyak digunakan. Saat ini, *android* mendominasi industri seluler berkat arsitekturnya yang open-source. Sistem operasi ini menawarkan berbagai macam aplikasi dan fitur dasar. Meskipun pengguna aplikasi biasanya mempercayakan keamanan data mereka kepada Android, kenyataannya Android terbukti lebih rentan dan kurang stabil.

*Android* adalah sebuah platform operasi yang dibangun di atas *kernel Linux*, yang dirancang khusus untuk digunakan pada perangkat mobile seperti

smartphone dan komputer tablet (Prabowo et al., 2020). Sedangkan penelitian yang dilakukan oleh Liu et al., (2023), *Android* adalah sistem berbasis *Java* yang rumit dengan banyak file, direktori, dan hubungan antar komponen. Membangun aplikasi-aplikasi ini membutuhkan dukungan alat pembangun (*build tool*) yang canggih bagi para pengembang.

Dari kesimpulan di atas dapat dijelaskan bahwa *Android* adalah sebuah platform operasi yang dibangun di atas *kernel Linux*, yang dirancang khusus untuk perangkat mobile seperti smartphone dan komputer tablet. Pembangunan aplikasi membutuhkan dukungan alat pembangun (*build tool*) yang canggih bagi para pengembang. *Android* sebagai sistem operasi menyediakan lingkungan yang kompleks dan *fleksibel* untuk pengembangan aplikasi.

## 6. C#

C# adalah bahasa pemrograman yang dibuat oleh Microsoft untuk memperkuat *Framework .NET*. Bahasa ini dirancang bersamaan dengan pengembangan *Framework .NET*. Dalam praktiknya, C# menjanjikan efisiensi, *fleksibilitas*, dan kemudahan yang sudah dikenal dari aplikasi-aplikasi sebelumnya seperti Visual Basic, C++, dan C# (Yahya & Sadali 2019:13). Sedangkan penelitian yang dilakukan oleh Raharjo (2022:2) C# atau C Sharp adalah sebuah bahasa pemrograman berbasis objek yang diciptakan oleh Microsoft pada awal 2000-an dengan Anders Hejlsberg sebagai pemimpin pengembangannya. Bahasa ini dirancang untuk tujuan umum yang simpel, bisa digunakan untuk mengembangkan berbagai jenis

aplikasi seperti aplikasi konsol, desktop, *web*, dan *mobile*, sebagaimana bahasa-bahasa pemrograman modern pada umumnya.

Dari kesimpulan di atas dapat dijelaskan bahwa C# adalah bahasa pemrograman yang dikembangkan oleh Microsoft dengan tujuan memperkuat *Framework .NET*. C# dirancang untuk mengikuti perkembangan *Framework .NET* dan menawarkan produktivitas, fleksibilitas, serta kemudahan yang ada pada bahasa pemrograman sebelumnya seperti Visual Basic, C#, dan C++. Selain itu, C# adalah bahasa pemrograman berorientasi objek yang diciptakan pada awal 2000-an di bawah kepemimpinan Anders Hejlsberg, dan dirancang sebagai bahasa tujuan umum yang sederhana untuk mengembangkan berbagai jenis aplikasi, termasuk yang berjalan di konsol, desktop, web, dan perangkat seluler.

## 7. Unity

*Unity* adalah mesin permainan profesional yang digunakan untuk mengembangkan video game untuk berbagai platform. Alat ini tidak hanya digunakan oleh ribuan pengembang game berpengalaman setiap hari, tetapi juga sangat mudah diakses oleh pengembang game pemula (Budiyati & Yanti 2021:182). Sedangkan penelitian yang dilakukan oleh Oktrilani et al., (2023:81), *Unity engine* adalah mesin pemrograman yang digunakan untuk membuat game 2D/3D. Mesin game ini adalah elemen yang bekerja di balik layar setiap permainan. Mesh merujuk pada bentuk standar objek tiga dimensi dan tidak dibuat langsung di platform *Unity*. Sebaliknya, objek game adalah komponen yang digunakan dalam berbagai aspek permainan.

Dari kesimpulan di atas dapat dijelaskan bahwa *Unity* merupakan sebuah mesin permainan profesional yang sangat luas penggunaannya dalam pengembangan video game untuk berbagai platform. Penelitian menyoroti popularitas *Unity* di kalangan pengembang game dari berbagai tingkat pengalaman, baik yang sudah berpengalaman maupun yang masih pemula. Sementara itu, penelitian lain menjelaskan bahwa *Unity engine* merupakan inti dari proses pemrograman untuk membuat game 2D dan 3D, dengan mesin ini mengoperasikan berbagai elemen di balik layar setiap permainan. Dengan demikian, *Unity* tidak hanya menyediakan alat yang mudah diakses tetapi juga mendukung pengembangan game dengan berbagai kompleksitas, mulai dari objek standar hingga integrasi yang lebih mendalam dalam aspek-aspek permainan.

## 8. UML

UML (*Unified Modeling Language*) digambarkan sebagai sebuah bahasa baku yang sering diterapkan dalam industri untuk menetapkan kebutuhan, menganalisis, merancang, dan mengilustrasikan arsitektur dalam pemrograman yang berorientasi objek (Wira et al., 2019:33). Sedangkan penelitian yang dilakukan oleh Rahmat Gunawan et al., (2021:49) UML (*Unified Modeling Language*) dikenali sebagai bahasa pemodelan yang dipergunakan untuk menetapkan, menggambarkan secara visual, membangun, dan mendokumentasikan elemen-elemen dari suatu sistem. UML berfungsi sebagai kerangka kerja yang memfasilitasi pengembang

untuk merancang dan menggambarkan berbagai aspek sistem perangkat lunak dengan efektif dan komprehensif.

Dari kesimpulan di atas dapat dijelaskan bahwa UML (*Unified Modeling Language*) merupakan bahasa pemodelan yang secara luas dipakai dalam industri perangkat lunak. Fungsi utama UML adalah untuk menetapkan kebutuhan, melakukan analisis, merancang, memvisualisasikan, membangun, dan mendokumentasikan sistem perangkat lunak. Melalui UML, pengembang dan pemangku kepentingan mampu dengan efisien menggambarkan, merencanakan, dan berkomunikasi mengenai sistem perangkat lunak, memudahkan dalam pengembangan dan pemeliharaan proyek perangkat lunak berbasis objek.

## **9. R&D**

R&D adalah proses yang digunakan untuk mengembangkan dan memvalidasi produk pendidikan. Hasil dari penelitian pengembangan tidak hanya mencakup penciptaan produk baru, tetapi juga penemuan pengetahuan atau solusi untuk masalah praktis (Alimiah et al., 2022:231). Sedangkan penelitian yang dilakukan oleh Waruwu (2024:1221), R&D adalah metode penelitian yang saat ini banyak diadopsi oleh dunia akademik untuk menciptakan produk dengan melalui proses identifikasi masalah potensial, desain, serta pengembangan produk sebagai solusi yang paling optimal.

Dari kesimpulan di atas dapat dijelaskan bahwa metode penelitian dan pengembangan (R&D) adalah proses yang digunakan untuk mengembangkan dan memvalidasi produk, khususnya dalam bidang pendidikan. Metode ini

tidak hanya berfokus pada penciptaan produk baru, tetapi juga pada penemuan pengetahuan atau solusi untuk masalah praktis. R&D banyak diadopsi oleh dunia akademik karena melalui proses identifikasi masalah potensial, desain, dan pengembangan produk, mampu menghasilkan solusi yang paling optimal untuk berbagai kebutuhan.

## 10. Use Case Diagram

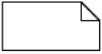
*Use Case Diagram* adalah gambaran visual yang digunakan untuk menunjukkan bagaimana pengguna berinteraksi dengan sistem informasi dalam berbagai situasi atau skenario yang berbeda. Diagram ini membantu untuk memodelkan cara sistem digunakan dari sudut pandang pengguna atau aktor yang terlibat. Diagram ini memusatkan perhatian pada hubungan antara pengguna sistem dan sistem itu sendiri (Wira et al., 2019:33). Sedangkan penelitian yang dilakukan oleh Simatupang & Sianturi (2019:16), *Use Case Diagram* adalah jenis diagram yang digunakan dalam pemodelan perangkat lunak untuk menggambarkan cara aktor, seperti pengguna atau entitas eksternal lainnya, berinteraksi dengan sistem perangkat lunak yang sedang dibuat.

Dari kesimpulan di atas dapat dijelaskan bahwa *Use Case Diagram* adalah representasi visual yang digunakan untuk mengilustrasikan bagaimana pengguna atau aktor berinteraksi dengan sistem informasi atau perangkat lunak dalam berbagai situasi atau skenario. Diagram ini membantu dalam memodelkan cara pengguna atau aktor menggunakan sistem dari perspektif interaksi mereka dengan sistem tersebut, serta menjelaskan hubungan antara

pengguna (aktor) dan sistem yang dibuat. Melalui *Use Case Diagram*, kita dapat mengidentifikasi fungsi-fungsi dalam sistem dan juga mengilustrasikan interaksi aktor dengan sistem tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat dalam Tabel 2.1

Tabel 2. 1. Simbol-simbol Use Case

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	menggambarkan peran atau entitas
2		<i>Dependency</i>	kondisi di mana perubahan pada satu elemen dalam sistem dapat mempengaruhi elemen lain yang tergantung padanya.
3		<i>Generalization</i>	perilaku dan struktur data dari objek induk di atasnya dalam hierarki objek
4		<i>Include</i>	penggunaan yang menyatakan bahwa use case sumber akan mencakup use case yang dituju dalam alur kerja yang dijelaskan.
5		<i>Extend</i>	menambah perilaku tambahan ke use case sumber pada titik tertentu jika kondisi tertentu terpenuhi.
6		<i>Association</i>	koneksi yang mengaitkan satu objek pada objek lainnya dalam pemodelan perangkat lunak.
7		<i>System</i>	menggambarkan kumpulan atau paket yang mewakili suatu sistem dalam pemodelan perangkat lunak
8		<i>Use Case</i>	urutan langkah-langkah dalam sistem yang menghasilkan hasil yang dapat diukur untuk aktor tertentu
9		<i>Collaboration</i>	hubungan aturan antara elemen-elemen dalam sistem yang menghasilkan sinergi untuk menciptakan perilaku yang melebihi sumbangan individu mereka

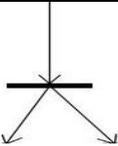
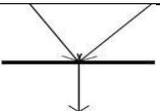
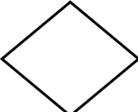
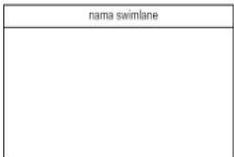
10		<i>Note</i>	informasi tambahan dalam pemodelan perangkat lunak yang memberikan wawasan lebih lanjut tentang elemen-elemen dalam diagram pemodelan
----	---	-------------	---

## 11. Activity Diagram

*Activity diagram* adalah bentuk diagram yang memvisualisasikan urutan kerja atau aktivitas dalam sebuah sistem perangkat lunak. Diagram ini menekankan serangkaian kegiatan dan bagaimana aliran informasi atau kendali terjadi di antara kegiatan-kegiatan tersebut (Wira et al., 2019:33). Sedangkan penelitian yang dilakukan oleh Simatupang & Sianturi (2019:18), *activity diagram* atau diagram aktivitas digunakan untuk menggambarkan alur kerja atau aktivitas dalam suatu sistem atau proses bisnis, termasuk menu-menu yang terdapat dalam perangkat lunak.

Dari kesimpulan di atas dapat dijelaskan bahwa *activity diagram* adalah jenis diagram yang menggambarkan urutan kerja atau aktivitas dalam sebuah sistem perangkat lunak. Fokus diagram ini adalah menyoroti rangkaian aktivitas serta bagaimana aliran informasi atau kendali terjadi di antara aktivitas-aktivitas tersebut. Tujuan utamanya adalah untuk mengilustrasikan aliran kerja atau proses, termasuk menu-menu yang ada dalam perangkat lunak. *Activity diagram* membantu dalam memvisualisasikan interaksi dan alur kerja sistem atau proses bisnis secara jelas dan terstruktur. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat dalam Tabel 2.2.

Tabel 2. 2. Simbol-simbol Activity Diagram

No	Simbol	Nama	Deskripsi
1		<i>Start Point</i>	ditempatkan di sudut kiri atas sebagai titik awal untuk memulai aktivitas atau proses dalam diagram atau representasi
2		<i>End Point</i>	ditempatkan sebagai penutup aktivitas atau proses dalam diagram atau representasi
3		<i>Activities</i>	Digunakan untuk menjelaskan dan menggambarkan aktivitas dalam suatu sistem atau proses bisnis.
4		<i>Fork</i>	menunjukkan kegiatan yang berlangsung secara simultan dalam diagram, sedangkan Join menggabungkan alur proses yang sebelumnya bercabang menjadi satu alur tunggal.
5		<i>Join</i>	digunakan untuk menunjukkan adanya penggabungan atau penyatuan alur proses.
6		<i>Decision Points</i>	gambaran dalam pemodelan yang menunjukkan pilihan atau kondisi dengan pengambilan keputusan, biasanya disimbolkan dengan kondisi "true" atau "false".
7		<i>Swimlane</i>	pembagian dalam diagram aktivitas yang menunjukkan peran dan tanggung jawab masing-masing entitas dalam suatu proses atau aktivitas.

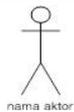
## 12. Sequence Diagram

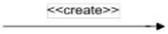
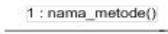
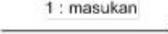
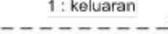
*Sequence diagram* adalah alat visual yang digunakan untuk mengilustrasikan aktivitas obyek pada suatu *use case*. Diagram ini mencatat waktu di mana obyek-obyek hidup dan berinteraksi dengan mengirim dan

menerima pesan satu sama lain (Wira et al., 2019:33). Sedangkan penelitian yang dilakukan oleh Prasetya et al., (2022) *sequence diagram* adalah representasi visual yang digunakan untuk menggambarkan dengan detail bagaimana objek-objek dalam sistem berinteraksi, termasuk proses pengiriman pesan atau perintah beserta urutan waktu pelaksanaannya. Diagram ini biasanya mengatur objek-objek terlibat dalam operasi secara berurutan dari kiri ke kanan.

Dari kesimpulan di atas dapat dijelaskan bahwa *sequence diagram* adalah alat visual yang digunakan untuk merepresentasikan interaksi antara objek-objek dalam suatu sistem secara grafis. Fokus diagram ini adalah pada urutan waktu di mana objek-objek tersebut hidup dan berinteraksi dengan mengirim dan menerima pesan atau perintah. Secara umum, *sequence diagram* membantu memvisualisasikan dan menjelaskan bagaimana objek-objek berkomunikasi dan bekerja bersama dalam konteks suatu use case atau operasi sistem. Diagram ini menggambarkan hubungan sekuensial objek-objek dari kiri ke kanan, memberikan pemahaman yang rinci tentang urutan dan ketergantungan antar objek dalam sistem tersebut. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat dalam Tabel. 2.3

Tabel 2. 3. Simbol-simbol Sequence Diagram

Simbol	Nama	Deskripsi
	<i>Aktor</i>	Komponen dalam diagram urutan yang menggambarkan entitas di luar sistem yang terlibat pada sistem.
	<i>Garis Hidup/Lifeline</i>	Mengilustrasikan keberadaan suatu objek dalam diagram urutan.

	<i>Objek</i>	Menunjukkan entitas yang berkomunikasi melalui pesan dalam diagram urutan.
	<i>Waktu aktif</i>	Durasi yang diperlukan oleh sebuah objek untuk menjalankan aktivitas atau operasi dalam diagram urutan.
	<i>Pesan Tipe Create</i>	Menandakan bahwa satu objek menciptakan objek lain, dengan panah yang menunjuk ke objek baru yang dibuat.
	<i>Pesan Tipe Call</i>	Menunjukkan bahwa suatu objek menjalankan operasi atau metode pada objek lain atau pada dirinya sendiri.
	<i>Pesan Tipe Send</i>	Mengindikasikan bahwa suatu objek mentransfer data, masukan, atau informasi ke objek lain.
	<i>Pesan Tipe Return</i>	Digunakan untuk menggambarkan hasil pengiriman pesan dari satu objek ke objek lainnya dalam diagram urutan.
	<i>Pesan Tipe Destroy</i>	Menunjukkan bahwa suatu objek menghapus objek lain, dengan panah yang mengarah ke objek yang dihapus.

### 13. Flowchart

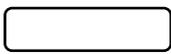
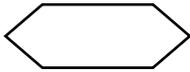
Menurut Tarsini & Anggraeni (2024:3) Flowchart merupakan rangkaian langkah-langkah untuk menyelesaikan suatu masalah yang diwakili oleh simbol-simbol khusus. Diagram ini menunjukkan logika alur program. Flowchart berfungsi tidak hanya sebagai alat komunikasi, tetapi juga sebagai panduan. Sebelum komponen-komponen tersebut dapat dimengerti dengan lebih baik, aturan desain diagram organisasi harus disampaikan.

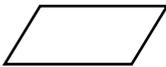
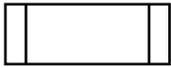
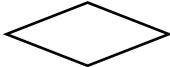
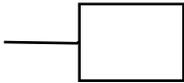
*Flowchart adalah* diagram yang digunakan untuk mendeskripsikan algoritma atau urutan langkah dalam suatu sistem. Analisis sistem

memanfaatkan *flowchart* sebagai alat dokumentasi untuk menyajikan gambaran logis dari sistem yang akan dikembangkan kepada para programmer secara visual (Rosaly & Prasetyo 2019:2). Sedangkan berdasarkan penelitian yang dilakukan oleh Syamsiah (2019:88), *Flowchart* adalah cara untuk menggambarkan proses penyelesaian masalah dengan menggunakan simbol-simbol khusus yang mudah dipahami, praktis dalam penggunaannya, dan diakui sebagai standar.

Dari kesimpulan di atas dapat dijelaskan bahwa *flowchart* adalah alat visual yang digunakan dalam analisis sistem dan pengembangan perangkat lunak. Fungsinya adalah menggambarkan algoritma atau langkah-langkah instruksi dalam sistem secara visual. Flowchart membantu analisis sistem dalam dokumentasi untuk menjelaskan logika sistem kepada programmer dengan lebih mudah. Simbol-simbol yang digunakan dalam flowchart dapat dilihat dalam Tabel. 2.4

Tabel 2. 4. Simbol-simbol Flowchart

Bagan	Nama	Keterangan
	<i>Terminator</i>	simbol yang menandai titik awal atau akhir dari sebuah program atau proses dalam flowchart.
	<i>Flow</i>	simbol yang menunjukkan arah atau hubungan antara berbagai simbol dalam flowchart.
	<i>Preparation</i>	simbol yang digunakan untuk menunjukkan tahap persiapan sebelum proses utama dalam sistem atau proses yang sedang dijelaskan.
	<i>Proses</i>	simbol yang menggambarkan kegiatan atau aktivitas yang dilakukan dalam suatu sistem

		atau program dalam flowchart.
	<i>Input/output data</i>	simbol yang menunjukkan interaksi antara sistem dan pengguna melalui input dan output data dalam flowchart.
	<i>Sub Program</i>	simbol yang mewakili subproses atau langkah-langkah terpisah dari program utama dalam flowchart.
	<i>Decision</i>	simbol yang digunakan untuk menunjukkan keputusan atau kondisi yang harus dipertimbangkan dalam alur proses flowchart.
	<i>On Page Connector</i>	simbol yang menghubungkan dua simbol yang terletak pada halaman yang sama dalam flowchart.
	<i>Comment</i>	simbol yang digunakan untuk memberikan keterangan atau penjelasan tambahan dalam flowchart.
	<i>Off Page Connector</i>	simbol yang menghubungkan dua simbol yang berada di halaman flowchart yang berbeda, memfasilitasi aliran informasi antar halaman.

#### 14. Black Box Testing

Menurut Maulana et al., (2019:28) *black box testing* adalah pendekatan di mana sistem dianggap sebagai entitas keseluruhan tanpa perlu mengakses detail internalnya. Fokus utama pengujian ini adalah pada input dan output yang terkait dengan sistem atau perangkat lunak tersebut, tanpa memperhatikan atau menyelidiki rincian internalnya. Sedangkan penelitian yang dilakukan oleh Lesmono (2019:23), menyatakan pengujian *black box* tidak dapat menggantikan pengujian *white box*. Sebaliknya, bahwa pengujian

black box dapat melengkapi pengujian *white box* dengan kemampuan untuk menemukan jenis kesalahan yang berbeda yang mungkin tidak terdeteksi oleh pengujian *white box*.

Pengujian *black box testing* bertujuan untuk mengidentifikasi kesalahan yang termasuk dalam kategori-kategori berikut, sesuai dengan penjelasan (Lesmono, 2019:23) :

- a). Fungsi yang tidak tepat atau hilang.
- b). Kesalahan dalam antar muka.
- c). Kesalahan terkait struktur data atau akses basis data eksternal.
- d). Kesalahan dalam perilaku atau performa.

## **B. Kajian Empiris**

Berikut adalah kajian empiris tentang pembuatan aplikasi *android* menggunakan *Unity*: Tinjauan Terhadap Penelitian Terdahulu :

Penelitian yang dilakukan oleh Putra (2022) bertujuan untuk menyelidiki cara memanfaatkan aplikasi *Unity* untuk mengajarkan kepada anak-anak tentang operasi matematika seperti penambahan, pengurangan, perkalian, dan pembagian. Permainan ini juga dirancang agar dapat menjadi alternatif pembelajaran yang lebih fleksibel dan menyenangkan, dengan antarmuka pengguna yang menarik dan mudah digunakan sehingga pemain dapat menikmati pengalaman bermainnya dengan nyaman. Pengujian black box menunjukkan bahwa semua fungsi yang telah dikembangkan berjalan dengan baik.

Penelitian yang dilakukan oleh Ezza et al., (2022) bertujuan untuk mengembangkan media pembelajaran *Augmented Reality* berbasis *Android* dengan menggunakan perangkat lunak *Unity*. Tujuannya adalah untuk meningkatkan hasil belajar siswa, terutama bagi guru mata pelajaran teknik animasi 3 dimensi dan siswa kelas XI Multimedia di SMK Negeri 1 Gunung Meriah. Penelitian ini mengadopsi pendekatan model *Waterfall* yang terdiri dari lima tahapan, yaitu Definisi Persyaratan, Desain Sistem dan Perangkat Lunak, Implementasi dan Pengujian Unit, Integrasi dan Pengujian Sistem, serta Operasi dan Pemeliharaan. Selain itu, penelitian ini juga menguji validitas media pembelajaran dari sisi teknis dan materi, serta menganalisis tingkat penerimaan atau akseptansi dari pengguna.

Penelitian yang dilakukan oleh Alkurdi & Bahzad Ahmed (2022) bertujuan untuk mengeksplorasi pengembangan aplikasi realitas campuran (*mixed reality*) menggunakan *Unity* sebagai lingkungan pengembangan utama, sekaligus memanfaatkan *Arduino* untuk menciptakan berbagai metode input/output dalam pengalaman pengguna. Studi ini difokuskan pada memanfaatkan kemampuan *Unity* sebagai mesin pengembangan game serbaguna untuk membangun aplikasi AR/VR/MR yang mengintegrasikan model 3D untuk interaksi melalui perangkat yang mendukung AR. Peran *Arduino* sangat penting dalam meningkatkan interaksi fisik dalam lingkungan AR melalui mikrokontroler dan sensor, dengan tujuan untuk memperkaya interaktivitas dan realisme dari pengalaman AR.

Penelitian yang dilakukan oleh Dwipayana et al., (2019) bertujuan untuk merancang dan mengimplementasikan sebuah aplikasi Gamelan Gong Kebyar berbasis teknologi *Virtual Reality*. Tujuan utamanya adalah untuk memfasilitasi pembelajaran dan pengalaman bermain Gong Kebyar secara interaktif melalui simulasi virtual. Penelitian menggunakan model ADDIE dalam proses pengembangannya, yang meliputi perancangan menggunakan *Unified Modeling Language* (UML) untuk model fungsional aplikasi dan implementasi dengan bahasa pemrograman C# melalui platform *MonoDevelopUnity*.

Penelitian yang dilakukan oleh Pradibta et al., (2021) bertujuan untuk mengembangkan sebuah aplikasi pembelajaran berbasis teknologi realitas untuk memperkenalkan dan mengajarkan bahasa Inggris kepada anak-anak usia dini, khususnya pada pendidikan prasekolah. Tujuan utamanya adalah untuk mengatasi kendala-kendala yang sering dihadapi, seperti kurangnya motivasi anak-anak dalam belajar bahasa Inggris, dengan menyajikan konten yang menarik dan interaktif menggunakan teknologi realitas. Aplikasi ini dirancang menggunakan pendekatan pembelajaran berpusat pada siswa (*student-centered*), yang memungkinkan siswa untuk belajar secara aktif dan melibatkan aspek afektif (emosional) serta psikomotorik (gerakan fisik). Media pembelajaran berbentuk kubus dengan penanda huruf memberikan pengalaman pemindaian untuk menghasilkan tampilan 3D dari objek-objek penanda huruf.

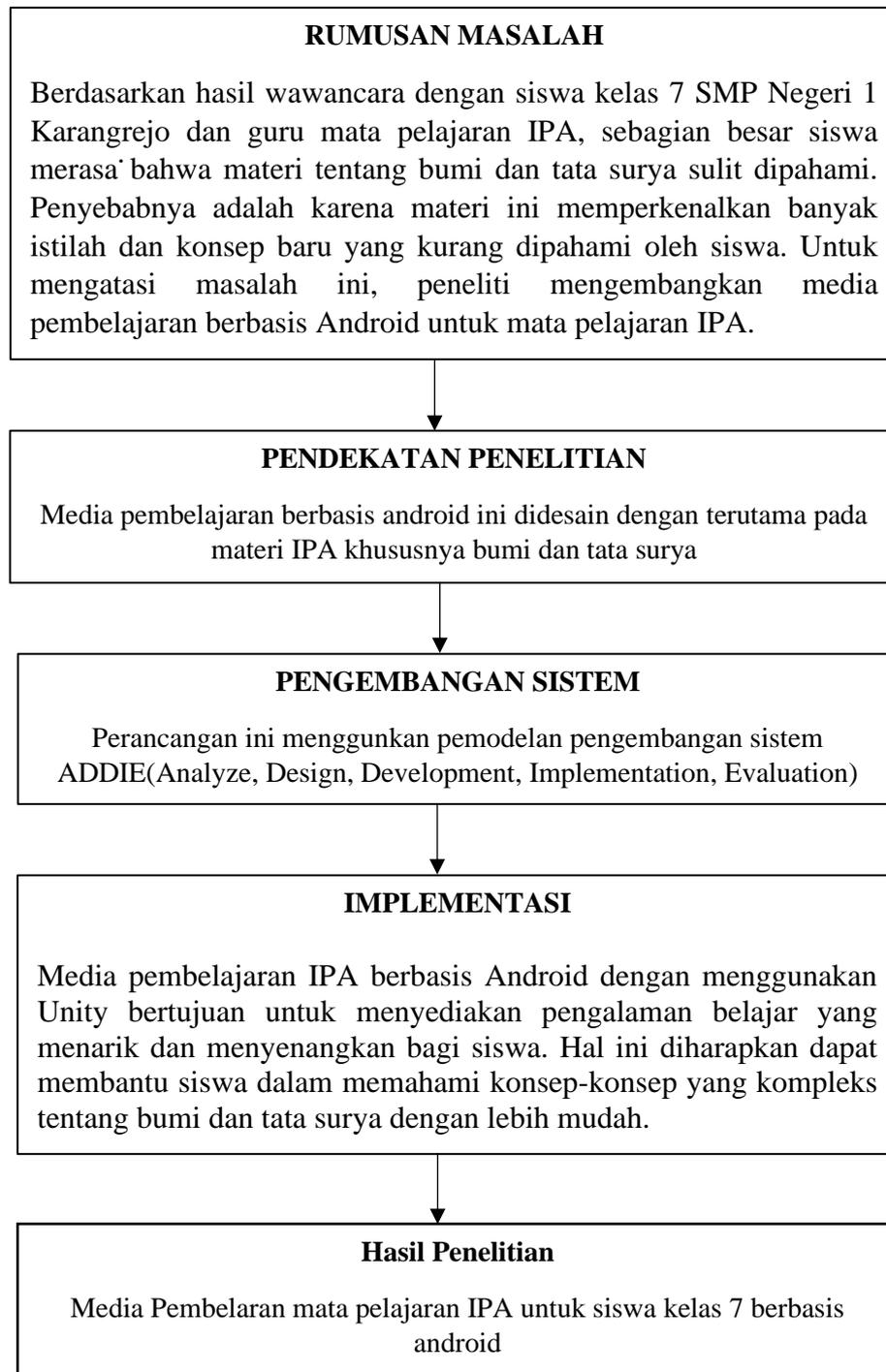
Dari kesimpulan diatas tentang analisis empiris terhadap lima penelitian sebelumnya adalah dapat disimpulkan bahwa penggunaan teknologi dalam pendidikan terus mengalami kemajuan yang pesat. Penelitian menggunakan *Unity* sebagai platform utama, dalam mengajarkan matematika kepada anak-anak, dan dalam mengembangkan *Augmented Reality* untuk pembelajaran, menunjukkan bahwa teknologi ini dapat signifikan meningkatkan keterlibatan dan pemahaman siswa. Penelitian lainnya yang mengembangkan aplikasi *Mixed Reality* dengan *Arduino* juga menyoroti potensi teknologi untuk meningkatkan interaktivitas dan realisme dalam pembelajaran. Sementara itu, penelitian tentang aplikasi Gamelan Gong Kebyar berbasis *Virtual Reality*, serta penelitian dalam mengajarkan bahasa Inggris kepada anak-anak usia dini melalui teknologi realitas, menunjukkan bahwa integrasi teknologi dalam pendidikan dapat menciptakan pengalaman belajar yang lebih menyenangkan dan efektif. Secara keseluruhan, penggunaan teknologi modern seperti *Unity*, *Augmented Reality*, *Virtual Reality*, dan *Mixed Reality* memberikan peluang besar untuk meningkatkan kualitas dan aksesibilitas pembelajaran di era digital saat ini.

### **C. Kerangka Berfikir**

Dalam proses pengembangan media pembelajaran IPA berbasis *android* menggunakan metode penelitian R&D, dengan model pengembangan ADDIE yang melibatkan beberapa tahap berurutan: yaitu Analisis, Desain, Pengembangan, Implementasi, dan Evaluasi. Tahap pertama, Analisis, melibatkan pengumpulan data dan informasi untuk

memahami kebutuhan pembelajaran atau pelatihan, sementara tahap Desain melibatkan perencanaan rinci mengenai bagaimana materi atau program akan disusun. Selanjutnya, tahap Pengembangan melibatkan pembuatan materi atau program berdasarkan desain yang telah dirancang sebelumnya. Materi atau program yang telah dikembangkan kemudian diterapkan dalam situasi nyata pada tahap Implementasi. Terakhir, tahap Evaluasi digunakan untuk mengevaluasi efektivitas materi atau program yang telah diimplementasikan dan mengidentifikasi area yang perlu ditingkatkan. Proses ini memberikan kerangka kerja yang komprehensif dalam menciptakan materi pembelajaran atau program pelatihan yang efektif dan efisien. Kerangka berfikir penelitian ini dapat dilihat dalam Gambar 2.1

Adapun kerangka berpikir pada penelitian ini sebagai berikut :



Gambar 2. 1 Kerangka berpikir