



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Menurut Sanders dikutip (Yulindawati dkk., 2023) Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan *input* data, memperoses nya dan menghasilkan *outpot* berdasarkan intruksi-intruksi program yang tersimpan di memori.

Menurut (Munawar dkk., 2023) Komputer adalah perangkat elektronik, yang beroperasi dibawah kendali instruksi menerima data, mengelola data secara aritmatika dan logis, menghasilkan keluaran dari pengolahan, dan menyimpan dalam memori.

Menurut Prawiro (2019) Komputer adalah suatu perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunaanya.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Komputer adalah sebuah perangkat elektronik yang dirancang untuk mengolah data secara cepat dan tepat berdasarkan instruksi program yang tersimpan di memori.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut (Suparman dkk., 2023) *Software* merupakan program yang berjalan pada komputer dan memungkinkan pengguna untuk melakukan berbagai macam tugas, memiliki 2 jenis utama yaitu sebagai sistem operasi dan aplikasi.



Menurut (Munawar dkk., 2023) Perangkat lunak merupakan sebuah program yang berada didalam komputer, tidak dapat disentuh secara fisik tetapi dapat dioperasikan oleh pengguna melalui antarmuka yang disediakan.

Menurut Harmayani (2021) *Software* dapat diartikan sebagai suatu kumpulan data elektronik yang tersimpan dan diatur oleh komputer, bisa berupa program ataupun koneksi untuk menjalankan berbagai macam instruksi perintah.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Software* adalah program yang berjalan pada komputer dan memungkinkan pengguna untuk melakukan berbagai macam tugas.

2.1.3 Pengertian Perangkat Keras (*Hardware*)

Menurut (Suparman dkk., 2023) *Hardware* merupakan komponen fisik yang terdiri dari berbagai macam komponen yang bekerja sama untuk menjalankan fungsi tertentu.

Menurut (Yulindawati dkk., 2023) *Hardware* adalah perangkat atau komponen komputer yang memiliki wujud fisik yang dapat dipegang dan juga dilihat, perangkat keras menyusun fisik dari sistem komputer.

Menurut (Harmayani, 2021) *Hardware* merupakan perangkat keras yang kasat mata dan memiliki wujud fisik yang dapat diraba, memiliki empat bagian yaitu *input device*, *output device*, *processing device* dan *storage device*.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Hardware* adalah komponen fisik dari sebuah komputer yang terdiri dari berbagai macam komponen yang bekerja sama untuk menjalankan fungsi tertentu.



2.1.4 Pengertian Basis Data

Menurut (Hadiprakoso, 2021) Basis data atau *database* adalah kumpulan data yang memiliki format stuktur tertentu sehingga memungkinkan sistem berbasis komputer untuk menyimpan, mengelola, dan mengambil data dengan cepat”.

Menurut (Rachmadi, 2020) Basis data adalah kumpulan data dalam bentuk file/tabel/arsip yang saling berhubungan dan tersimpan dalam media penyimpanan elektronis, untuk kemudahan dalam pengaturan, pemilihan, pengelompokan dan pengorganisasian data sesuai tujuan.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Basis data adalah kumpulan data yang terstruktur dan tersimpan dalam media penyimpanan elektronik.

2.2. Teori Judul

2.2.1. Pengertian Rancang Bangun

Menurut Bambang dikutip (Mulyanto dkk, 2020) Rancang bangun adalah menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian.

Menurut Jogianto dikutip (Mulyanto dkk, 2020) Rancang bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dari suatu sistem.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Rancang bangun sistem adalah proses menciptakan sistem baru atau mengganti atau memperbaiki sistem yang telah ada..



2.2.2. Pengertian Aplikasi

Menurut Hakim dikutip (Gunawan dkk, 2022), Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur windows, melakukan permainan(*game*), dan sebagainya.

Menurut Anendya (2023) Aplikasi adalah program perangkat lunak yang dirancang untuk menjalankan tugas tertentu pada perangkat komputer, *smartphone*, *tablet*, atau perangkat lainnya.

Berdasarkan pengertian di atas, dapat disimpulkan aplikasi adalah program perangkat lunak yang dirancang untuk menjalankan tugas tertentu pada perangkat komputer, *smartphone*, *tablet*, atau perangkat lainnya.

2.2.3. Pengertian *Tracking*

Menurut Tysara (2022), *Tracking* adalah mengarah pada tindakan mengikuti atau menelusuri atau mencari. Kamus Besar Bahasa Indonesia (KBBI) menjelaskan arti *tracking* adalah mengikuti sama dengan memperhatikan dengan mendengarkan, melihat, membaca, dan sebagainya secara teliti atau baik-baik..

Menurut Nugroho & Suryadi (2021), *Tracking* bagian penting dari *search advertising* yang memungkinkan untuk melacak hasil dari iklan, proses *tracking* biasanya melibatkan beberapa alat seperti *tag tracking*, *cookie*, atau perangkat lunak analitik.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Tracking* adalah tindakan mengikuti, menelusuri, atau mencari sesuatu. *Tracking* dapat dilakukan dengan berbagai cara, seperti mendengarkan, melihat, membaca, atau menggunakan alat tertentu.

2.2.4. Pengertian Disposisi Surat

Menurut Kamus Besar Bahasa Indonesia dikutip (Rahmadhani, 2024), Disposisi adalah pendapat seorang pejabat mengenai urusan yang termuat dalam suatu surat dinas, yang langsung dituliskan pada surat yang bersangkutan atau pada lembar khusus. Lembar khusus ini disebut dengan lembar disposisi.



Menurut Efendi (2019), Disposisi adalah petunjuk tertulis mengenai tindak lanjut pengelolaan surat, ditulis secara jelas pada lembar disposisi, tidak pada naskah asli.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Disposisi surat adalah pendapat atau petunjuk tertulis dari seorang pejabat mengenai tindak lanjut pengelolaan surat dinas. Disposisi dituliskan pada surat yang bersangkutan atau pada lembar khusus yang disebut lembar disposisi.

2.2.5. Pengertian *QR Code*

Menurut (Habibi dkk, 2020), *QR Code* merupakan kepanjangan dari Quick Respons yaitu kode batang dua dimensi yang diciptakan tahun 1994 yang dapat menyimpan data lebih banyak dibanding *barcode*.

Menurut Rahmalia (2023), *QR Code* adalah singkatan dari *quick response code*. Arti dari kode dalam *QR code* ini adalah *barcode* dua dimensi yang bisa memberikan beragam jenis informasi secara langsung.

Berdasarkan dua definisi di atas, dapat disimpulkan bahwa *QR Code* adalah kode batang dua dimensi yang dapat menyimpan data lebih banyak dibandingkan *barcode* biasa.

2.2.6. Pengertian *Mobile*

Menurut (Putra dkk, 2023), *Mobile* atau bisa disebut juga Aplikasi Seluler (Perangkat Bergerak) adalah sejenis perangkat lunak aplikasi yang dimaksudkan untuk berjalan di ponsel, misalnya *smartphone* atau PC tablet.

Menurut Pane (2022), *Mobile* adalah kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah atau dapat diartikan sebuah benda yang berteknologi tinggi dan dapat bergerak tanpa menggunakan tabel.

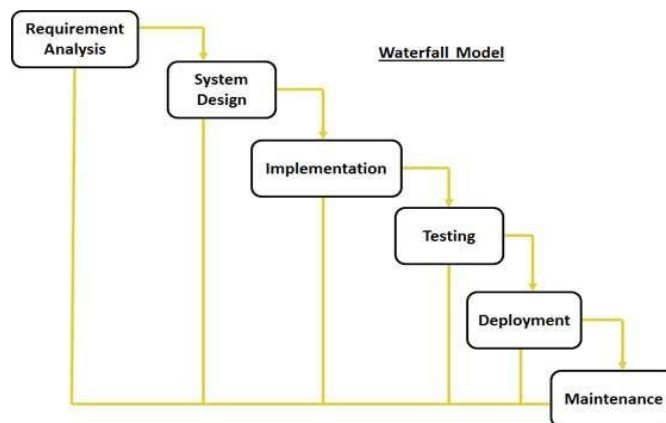
Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Mobile* memiliki dua makna utama, yaitu sebagai kata sifat dan kata benda. Sebagai kata sifat, *mobile* berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah, atau berteknologi tinggi dan dapat bergerak tanpa menggunakan



tabel. Sebagai kata benda, *mobile* mengacu pada aplikasi seluler (Perangkat Bergerak).

2.3. Teori Khusus

2.3.1 Pengertian Metode Pengembangan *Waterfall*



Gambar 2. 1 Tahapan Pengembangan Waterfall

Menurut (Rasjid dkk., 2023) *Waterfall* adalah salah satu model pengembangan perangkat lunak yang paling tua dan paling terstruktur, melalui pedelakan SDLC dan bersifat serial mulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem.

Menurut Sukamto & Shalahuddin dikutip (Gunawan dkk, 2022), *Waterfall* merupakan model yang melakukan pendekatan alur hidup pada perangkat lunak secara terurut yang dimulai dari analisis sistem, desain sistem baru, pengkodean dan pengujian. Metode *waterfall* membagi proses pengembangan perangkat lunak menjadi beberapa tahapan yang saling berurutan, yaitu:

1. *Requirement Analysis*: Tahap ini mengumpulkan dan menganalisa kebutuhan pengguna.
2. *System Design*: Tahap ini mendesain sistem yang akan dibangun.
3. *Implementation*: Tahap ini mengimplementasikan sistem yang telah dirancang.



4. *Testing*: Tahap ini menguji sistem yang telah diimplementasikan.
5. *Deployment*: Tahap ini menempatkan sistem ke lingkungan produksi.
6. *Maintenace*: Tahap ini memelihara sistem yang telah diuji.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Metode *Waterfall* adalah model pengembangan perangkat lunak yang bersifat sistematis dan berurutan, mulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem.

2.3.2 Pengertian Analisis PIECES

Menurut Sudiati dkk dikutip (Nurhayati dkk, 2021) Metode PIECES digunakan untuk melakukan analisa terhadap kinerja informasi, ekonomi, keamanan, efisiensi dan pelayanan guna mengidentifikasi kelemahan sistem berjalan agar dapat direkomendasikan perbaikan yang harus dibuat pada sistem yang baru.

Menurut Umar dkk dikutip (Alifariki dkk, 2023) Analisis PIECES diartikan sebagai suatu sistem yang digunakan untuk menganalisis sistem kerja suatu perusahaan, lembaga atau organisasi.

Adapun bagian-bagian analisis PIECES yaitu:

1. Analisis Kinerja (*Performance*)

Analisis kinerja adalah kemampuan dalam menyelesaikan tugas bisnis dengan cepat sehingga sasaran dapat segera tercapai.

2. Analisis Informasi (*Information*)

Informasi merupakan hal yang penting bagi pengguna akhir untuk pengambilan keputusan. Evaluasi terhadap kemampuan informasi agar menghasilkan informasi yang bermanfaat.

3. Analisis Ekonomi (*Economic*)

Analisis ekonomi merupakan kemampuan atas penghematan biaya dan meningkatkan keuntungan perusahaan.



4. Analisis Keamanan (*Security*)

Analisis keamanan merupakan penilaian yang digunakan untuk mengamankan data dari kerusakan.

5. Analisis Efisiensi (*Efficiency*)

Analisis efisiensi merupakan pemakaian secara maksimal atas sumber daya yang tersedia meliputi manusia, informasi, waktu, dan keterlambatan pengolahan data.

6. Pelayanan (*Service*)

Merupakan pemecahan masalah yang terjadi agar mendapatkan pelayanan yang berkualitas.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Analisis PIECES adalah metode yang digunakan untuk mengevaluasi kinerja dan kualitas sistem informasi.

2.3.3 Pengertian UML

Menurut (Destriana dkk, 2021) UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifact (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan non perangkat lunak lainnya.

Menurut Nugroho dikutip (Destriana dkk, 2021), UML (*Unified Modeling Language*) adalah metodologi kolaborasi antara metode Booch, OMT (*Object Modeling Technique*) serta OOSE (*Object Oriented Software Engineering*) dan beberapa metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk Analisa dan perancangan sistem dengan metodologi berorientasi objek.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan untuk



mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak melalui metodologi berorientasi objek.

2.3.4 Pengertian *Use Case*

Menurut (Rasjid dkk., 2023) *Use case* atau diagram *use case* menggambarkan kelakuan (behavior) sistem informasi yang akan dibuat, mendeskripsikan sebuah interaksi antara satu aktor atau lebih aktor dengan sistem yang akan dibuat.

Menurut (Gunawan dkk, 2022), *Use Case* merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem, yang dapat mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.

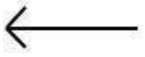
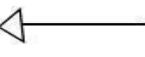
Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Use Case* adalah deskripsi yang menjelaskan interaksi antara sistem informasi dengan penggunanya. *Use Case* digunakan untuk menggambarkan fungsionalitas sistem informasi dari sudut pandang pengguna.

Tabel 2. 1 Deskripsi *Use Case*

Simbol	Nama	Deskripsi
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).



Tabel 2. 1 Lanjutan Deskripsi *Use Case*

Simbol	Nama	Deskripsi
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).



Tabel 2. 1 Lanjutan Deskripsi *Use Case*

Simbol	Nama	Deskripsi
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.3.5 Pengertian *Activity Diagram*

Menurut (Destriana dkk, 2021), *Activity* diagram adalah diagram yang menggambarkan alur kerja dari berbagai aktivitas user atau sistem, orang yang melakukan aktivitas, dan aliran berurutan dari aktivitas ini.

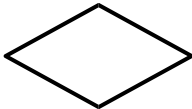
Menurut Sari dan Indra (2021) *Activity* diagram merupakan penggambaran aktivitas dalam diagram yang dipersentasikan dengan bentuk bujur sangkar bersudut tidak lancip, yang didalamnya berisi langkah-langkah apa saja yang terjadi dalam aliran kerja.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Activity* diagram adalah diagram yang menggambarkan aliran kerja atau proses bisnis dalam suatu sistem.

Tabel 2. 2 Deskripsi *Activity* Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal



Aktivitas/activity 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Interaksi/interaction 	Alur dari sebuah activity

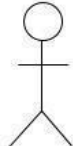
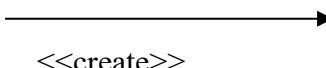
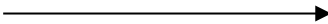
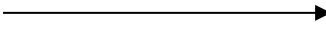
2.3.6 Pengertian *Sequence Diagram*

Menurut (Destriana dkk, 2021), *Sequence* diagram adalah diagram yang menggambarkan interaksi user dengan sistem secara sequensial (berurutan).

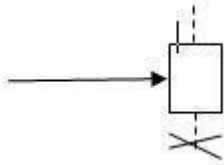
Menurut Sari dan Indra (2021) *Sequence* diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Berdasarkan Pengertian di atas, dapat disimpulkan bahwa *Sequence* diagram adalah diagram yang menggambarkan interaksi dan komunikasi antara objek-objek dalam sebuah sistem secara berurutan.

Tabel 2. 3 Deskripsi *Sequence Diagram*

Simbol	Keterangan
Aktor  Nama_aktor Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal frase nama actor.
Garis hidup/lifeline 	Menyatakan kehidupan suatu objek.
Objek Nama objek : Nama kelas	Menyatakan objek yang berinteraksi pesan.
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan beriteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe create  <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1: nama_metode () 	Menyatakan sebuah objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send 1: Masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

Tabel 2.3 Lanjutan Deskripsi *Sequence Diagram*

Simbol	Keterangan
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

2.3.7 Pengertian *Class Diagram*

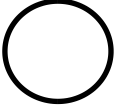
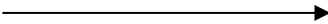
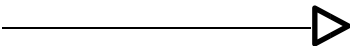
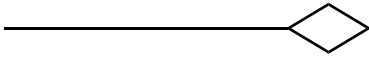
Menurut (Rasjid dkk., 2023) *Class diagram* digunakan untuk menggambarkan struktur dalam objek sistem. Diagram ini menunjukan *class object* yang menyusun sistem dan juga hubungan antara *class object*.

Menurut (Sari & Indra, 2021) *Class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Class diagram* adalah diagram yang menggambarkan struktur sistem dari segi kelas-kelas yang akan dibuat untuk membangun sistem.



Tabel 2.4 Deskripsi *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka/Interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah/ directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna kebergantungan antarmuka.
Kebergantungan/ dependency 	Relasi antar kelas dengan makna kebergantungan antarmuka.
Agregasi / Aggregation 	Relasi antar kelas dengan makna semua bagian (whole-part).



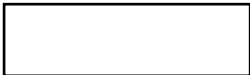
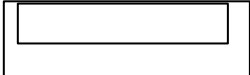
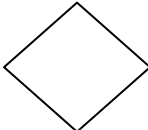
2.3.8 Pengertian *Entity Relationship Diagram* (ERD)

Menurut (Rasjid dkk., 2023), *Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

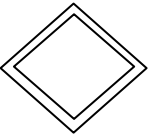
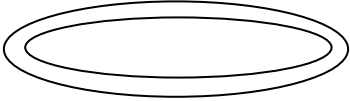
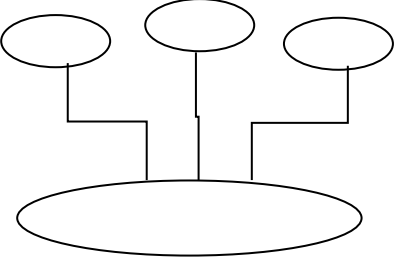
Menurut Suprpto (2021) ERD atau *Entity Relationship Diagram* adalah sebuah konsep yang mendeskripsikan hubungan antara penyimpanan basis data dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek yaitu disebut sebagai entity dan hubungan atau relasi antar objek-objek tersebut.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Entity Relationship Diagram* (ERD) adalah sebuah diagram yang menggambarkan desain konseptual dari model data relasional. ERD menunjukkan hubungan antar entitas dalam suatu sistem dan bagaimana entitas-entitas tersebut saling terkait.

Tabel 2. 5 Deskripsi ERD

Simbol	Deskripsi
Entity 	Objek data yang dapat dibedakan dalam dunia nyata berupa orang, tempat, benda, atau konsep.
Weak Entity 	Entitas yang keberadaannya bergantung pada entitas lain, memiliki atribut kunci yang tidak unik, sehingga harus memiliki kunci asing yang menunjuk ke entitas lain.
Relationship 	Hubungan antara dua entitas atau lebih.

Tabel 2. 5 Lanjutan Deskripsi ERD

Simbol	Deskripsi
Identifying Relationship 	Relationship yang membuat entitas lemah menjadi entitas kuat.
Atribut 	Karakteristik dari suatu entitas, berupa data tunggal, data gabungan, atau data turunan.
Atribut Kunci 	Atribut yang unik dan tidak boleh memiliki nilai NULL yang digunakan untuk mengidentifikasi secara unik setiap entitas.
Atribut Multivalue 	Atribut yang dapat memiliki beberapa nilai, biasanya digunakan untuk mewakili nilai-nilai yang dapat memiliki banyak kemungkinan.
Atribut Komposit 	Atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.

2.3.9 Pengertian Kamus Data

Menurut Hadiprakoso (2021), Kamus data berisi penjelasan rinci yang dibuat oleh perancang basis data tentang semua tabel yang terdapat dalam basis data.



Menurut Weli (2019), Kamus data mendefinisikan isi aliran data dan simpanan data agar desainer sistem memahami secara tepat potongan data macam apa yang dimiliki atau dengan kata kamus data merupakan katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak yang digunakan untuk menjelaskan arti, format, dan hubungan antar data sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut.

Tabel 2. 6 Deskripsi Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau...
{ }n	n kali diulang/bernilai banyak
()	Data opsional
* ... *	Batas komentar

2.4. Teori Program

2.4.1 Pengertian Figma



Gambar 2. 2 Logo Figma

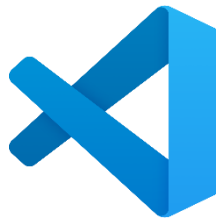


Menurut Cholil (2024) Figma adalah aplikasi untuk mendesain UI dan UX yang dapat digunakan untuk membuat aplikasi, web, serta berbagai komponen *user interface* yang dapat digabungkan ke dalam *project* lainnya, dan dapat berkolaborasi dengan desainer untuk membuat desain secara bersama.

Menurut Wibowo (2023) Figma adalah alat desain grafis berbasis web yang memungkinkan pengguna untuk membuat desain yang berkualitas tinggi untuk *website*, aplikasi, dan produk digital lainnya.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Figma adalah sebuah aplikasi desain UI dan UX berbasis web yang memungkinkan pengguna untuk membuat desain yang berkualitas tinggi untuk *website*, aplikasi, dan produk digital lainnya.

2.4.2 Pengertian Visual Studio Code



Gambar 2. 3 Logo Visual Studio Code

Menurut Huda (2022), *Visual Studio Code* adalah aplikasi *code editor* buatan Microsoft yang dapat dijalankan di semua perangkat desktop secara gratis. Kelengkapan fitur dan ekstensi membuat *code editor* ini menjadi pilihan utama para pengembang. *Visual Studio Code* bahkan mendukung hampir semua sistem operasi seperti Windows, Mac OS, Linux, dan lain sebagainya.

Menurut Martin (2024) *Visual Studio Code* adalah bagian dari Microsoft Visual Studio yang merupakan sebuah perangkat lunak lengkap yang dapat dijadikan sebagai pengembangan aplikasi baik itu aplikasi bisnis,



personal, maupun komponen aplikasi yang berbentuk aplikasi console, aplikasi Windows, ataupun aplikasi web.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *Visual Studio Code* adalah editor kode sumber yang populer karena ringan, mudah digunakan, dan memiliki banyak fitur yang bermanfaat bagi programmer.

2.4.3 Pengertian Dart



Gambar 2. 4 Logo Dart

Menurut (Putra & Fu'adi, 2024), Dart adalah sebuah bahasa pemrograman yang dibuat oleh Google, dieksekusi oleh *Dart Virtual Machine* tanpa ada proses penerjemahan ke *byte code*.

Menurut (Kurosaki, 2020), Dart adalah bahasa pemrograman utama yang dipakai untuk mengembangkan *framework* flutter, dan dikembangkan langsung oleh Google.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Bahasa Pemrograman Dart adalah bahasa pemrograman *framework* flutter, dan dikembangkan langsung oleh Google.

2.4.4 Pengertian Flutter



Gambar 2. 5 Logo Flutter

Menurut (Putra & Fu'adi, 2024), Flutter adalah sebuah *framework* yang dibuat oleh Google untuk membuat aplikasi yang *cross platform*, tidak



hanya aplikasi mobile, flutter juga dikembangkan untuk aplikasi berbasis web dan desktop.

Menurut (Kurosaki, 2020), Flutter adalah framework yang dipakai untuk membuat aplikasi android, ios, web app, desktop app, windows, linux, dan mac os.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Flutter adalah sebuah framework menggunakan bahasa pemrograman Dart dan dapat digunakan untuk mengembangkan aplikasi untuk platform Android, iOS, dan web.

2.4.5 Pengertian Firebase



Gambar 2. 6 Logo Firebase

Menurut (Sari & Indra, 2021) Firebase merupakan penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data, bersifat *Backend as a Service* (BaaS) yang dimiliki oleh Google untuk mempermudah pengembangan aplikasi mobile.

Menurut (Purnomo dkk, 2020), *Firebase Realtime Database* merupakan *database* yang sudah mendukung fasilitas multiplatform seperti Android, iOS, dan Web yang menyimpan dan menyinkronkan data antara user dan perangkat secara *realtime* menggunakan *database* NoSQL.

Berdasarkan pengertian diatas dapat disimpulkan bahwa Firebase Realtime Database adalah sebuah layanan penyimpanan data non-SQL disediakan oleh Google yang memungkinkan pengembang untuk menyimpan data secara *realtime* antara pengguna dan perangkat.

