

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan kajian pustaka terhadap penelitian-penelitian sebelumnya untuk mendukung penelitian yang dilakukan oleh penulis. Berikut ini adalah tinjauan literatur yang diulas sejauh ini:

Tabel 2.1. Daftar Literatur

No	Penulis	Tahun	Judul
1	Rizky Doli Pulungana, Rita Novita Sarib, Idzhari Rahman	2023	Rancang Bangun Aplikasi Penjadwalan Seminar Dan Sidang Tugas Akhir / Skripsi Berbasis <i>Android</i> Di Universitas Potensi Utama
2	Robbie Christhover, Sugiyatno, Herlawati	2022	Penerapan <i>Restapi</i> Menggunakan <i>Retrofit</i> Untuk Sistem Informasi Film Berbasis <i>Android</i>
3	Fitrah Amalia, I Wayan Agus Arimbawa, Royana Afwani	2019	IMPLEMENTASI <i>RESTFUL API</i> PADA PENGEMBANGAN APLIKASI IF-KU BERBASIS <i>ANDROID</i>
4	Hasyrif SY, Indo Intan	2019	Implementasi <i>Restful API</i> Portal Akademik Stmik Dipanegara Berbasis <i>Android</i>
5	Kiki Rizki,	2018	IMPLEMENTASI <i>GOOGLE MAPS API</i>

	Ahmat Adil		<p style="text-align: center;">BERBASIS <i>ANDROID</i> UNTUK LOKASI FASILITAS UMUM DI KABUPATEN SUMBAWA</p>
--	------------	--	---

2.1.1. Tinjauan Literatur 1

Pada tahun 2023, Rizky Doli Pulungana, Rita Novita Sarib dan Idzhari Rahman melakukan penelitian dengan judul “Rancang Bangun aplikasi penjadwalan seminar dan sidang tugas akhir/skripsi berbasis *Android* di Universitas Potensi Utama” untuk mengatasi masalah penjadwalan dimana penjadwalan masih dilakukan dengan *Ms. Excel* yang memiliki banyak kekurangan dalam update jadwal serta jadwal bentrok dengan jadwal mahasiswa lain oleh karena itu dibangunlah sistem yang dapat membantu BAAK dalam mengatur penjadwalan dan pendaftaran semindar dan sidang tugas akhir berbasis *Android* dengan menggunakan metode pengembangan sistem *Waterfall* yang merupakan salah satu metode *SDLC*(*Software Development Life Cycle*).

2.1.2. Tinjauan Literatur 2

Pada tahun 2022, Robbie Christover, Sugiyatno, Herlawati melakukan penelitian dengan judul ”Penerapan *restapi* Menggunakan *Retrofit* Untuk Sistem Informasi Film Berbasis *Android*” serta mengembangkan aplikasi dengan menerapkan *restapi* menggunakan *Retrofit* yang dapat melakukan pencarian informasi sinopsis film, dengan memanfaatkan API pada *website* <https://www.themoviedb.org/> (*The Movie Database*(*TMDB*), n.d.). Sehingga dengan ini, aplikasi tersebut dapat menunjukan Informasi Sinopsis Film serta untuk

metode pengembangan sistem menggunakan *Waterfall* yang merupakan salah satu metode *SDLC (Software Development Life Cycle)*.

2.1.3. Tinjauan Literatur 3

Pada tahun 2019, Fitrah Amalia, I Wayan Agus Arimbawa, Royana Afwani dari Universitas Mataram melakukan penelitian dengan judul "Implementasi *Restful API* pada pengembangan aplikasi IF-KU berbasis *Android*". Penelitian ini dilakukan untuk mengatasi masalah yang terjadi adalah informasi itu diakses melalui browser web dengan alamat yang berbeda. Dalam mengatasi masalah yang telah dideskripsikan, diperlukan suatu sistem yang dapat memberikan informasi dalam satu sistem. Sehubungan dengan ketersediaan Sistem Informasi Akademik (SIA) dan sistem Sidik Jari pada PSTI yang dilengkapi dengan *API* dengan metode *REST*, sehingga data tentang SIA dan Sidik Jari dapat ditampilkan pada sistem yang akan dikembangkan. Agar sistem yang telah dimiliki oleh PSTI dapat ditampilkan oleh sistem lain menggunakan layanan web. Oleh karena itu, aplikasi IF berbasis *Android* akan dikembangkan yang mengimplementasikan *API* yang tenang. Sistem ini menggunakan *Codeigniter* untuk membangun layanan web sementara kerangka *ionik* untuk membangun aplikasi *mobile*.

2.1.4. Tinjauan Literatur 4

Pada tahun 2019, Hasyrif SY dan Indo Intan melaksanakan sebuah penelitian yang berjudul "Implementasi *Restful API* Portal Akademik Stmik Dipanegara Berbasis *Android*". Penelitian ini dilakukan untuk mengatasi Masalah yang dihadapi saat ini adalah bahwa kita tahu saat ini hampir setiap mahasiswa telah memiliki ponsel berbasis *Android*, ini merupakan sebuah media yang dapat

dimanfaatkan dan semakin memperbesar peluang sampainya informasi kepada mahasiswa namun disisi lain website kampus saat ini hanya bisa diakses melalui browser sedangkan aplikasi *Android* memiliki mekanisme berbeda dalam mengakses informasi. Untuk menjembatani masalah ini maka akan dibuat sebuah aplikasi yang mengikuti konsep *Restful API*, pada metode ini akan dibuat sebuah aplikasi yang akan menampilkan data akademik dalam format *JSON*. *JSON* merupakan salah satu format data yang bisa di olah oleh semua teknologi akses data yang ada pada perangkat komunikasi. Struktur data ini sudah distandarisasi sehingga semua perangkat bisa mengakses data berupa json ini untuk diolah sesuai dengan kepentingannya masing masing. Dengan penerapan konsep tersebut diharapkan informasi akademik kampus bisa sampai dengan lebih cepat dan lebih mudah karena informasi dapat diakses melalui perangkat ponsel yang notabene telah dimiliki oleh hampir semua mahasiswa.

2.1.5. Tinjauan Literatur 5

Pada tahun 2018, Kiki Rizki dan Ahmat Adil melakukan penelitian yang berjudul "Implementasi *Google Maps API* Berbasis *Android* untuk lokasi fasilitas umum di kabupaten sumbawa". Penelitian ini dibuat untuk menyampaikan informasi lokasi fasilitas umum agar setiap orang yang berkunjung ke Kabupaten Sumbawa akan mempertimbangkan efisiensi waktu dan biaya untuk menuju lokasi bengkel tambal ban, toko *sparepart*, dan ATM. Sehingga informasi mengenai lokasi bengkel tambal ban, toko *sparepart*, dan ATM sangat diperlukan untuk pengguna *smartphone Android*. Informasi lokasi yang telah di definisikan tersebut dapat dilihat pada aplikasi *Google Maps Android*, namun data lokasi yang terkait masih terbatas. Adanya teknologi *API* dan pemetaan *Google* yaitu *Google*

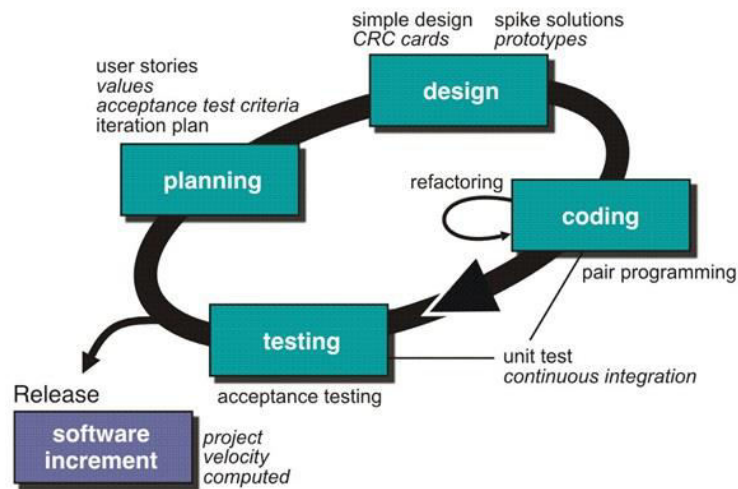
Maps dapat digunakan untuk mengembangkan suatu aplikasi pemetaan lokasi bengkel tambal ban, toko *sparepart*, dan ATM. Hasil dari aplikasi yang dikembangkan adalah dapat memberikan informasi jarak, waktu, dan rute menuju lokasi bengkel tambal ban, toko *sparepart*, dan ATM. Hasil pengujian sistem menunjukkan bahwa sistem pemetaan lokasi bengkel tambal ban, toko *sparepart*, dan ATM ini layak dan dapat dipergunakan di *smartphone Android* karena data yang terkandung pada aplikasi tersebut lebih fokus memiliki data lokasi bengkel tambal ban, toko *sparepart*, dan ATM.

2.2 Metode Pengumpulan Data

Metode pengumpulan data merupakan masalah penting dalam penelitian karena metode ini merupakan strategi untuk mendapatkan data yang dibutuhkan. Keberhasilan penelitian sangat tergantung pada teknik pengumpulan data yang digunakan. Dapatkan materi, informasi, fakta dan informasi terpercaya. Penelitian ini menggunakan teknik, prosedur, alat, dan aktivitas dunia nyata untuk mendapatkan data sebagaimana dimaksud. Proses pengumpulan data dapat dilakukan melalui wawancara, observasi dan dokumentasi (Nurdiansyah & Siti Rugoyah, 2021).

2.3 Extreme Programming (XP)

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang bertujuan untuk meningkatkan kualitas perangkat lunak dalam menanggapi perubahan dan kebutuhan pelanggan (Ulfi et al., 2020).



Gambar 2.1 Tahapan Metode Extreme Programming

Ada empat tahapan dalam metode *Extreme Programming* yaitu sebagai berikut:

1. *Planning* (perencanaan)

Pada tahapan ini pertama yang dibutuhkan oleh pengembang sebelum membuat sistem. Pada tahapan ini yang dibutuhkan yaitu perencanaan dan analisa kebutuhan dari pengguna yang akan masuk kedalam proses pembuatan sistem. Pengembang akan memberikan keputusan bagaimana untuk membangun hasil dari cerita yang disampaikan oleh pengguna sesuai dengan kesepakatan bersama, adapun *story-story* yang dapat dilakukan dengan beberapa cara berikut:

 - a. Pengguna akan mendeskripsikan cerita apa permasalahan yang terjadi pada sistem yang digunakan dan akan seperti apa sistem tersebut dibangun.

- b. Berdasarkan dari deskripsi cerita pengguna maka peneliti akan memberikan penentuan untuk poin pada bagian *value* yang fungsinya untuk memutuskan poin-poin apa yang akan dibangun.
- c. Dari hasil persetujuan ataupun kesepakatan maka peneliti selanjutnya menentukan pada bagian *acceptance test criteria* yaitu penentuan kriteria yang nantinya akan dipergunakan sebagai acuan pada sistem yang akan dilakukan pengujian.
- d. Dan selanjutnya peneliti akan memberikan kesimpulan berapa kali untuk dilakukan *release* dan perbaikan pada tahap berikutnya, yaitu tahap *iteration plan* yang memuat perencanaan berapa kali sistem yang dibangun akan dilakukan pengujian.

2. *Design* (perancangan)

Tahap selanjutnya yaitu perancangan atau desain setelah tahap perencanaan sudah terselesaikan. Ditahap ini pengembang akan melakukan perancangan dengan membuat model yang dimulai dengan pemodelan sistem terlebih dahulu, setelahnya dilakukan pemodelan arsitektur dan pemodelan basis data yang tujuannya untuk memberikan gambaran sistem yang akan dibangun

- a. *Simple Design* merupakan tahapan untuk pengembang dalam membangun perangkat lunak dengan desain yang sederhana. Diawali dengan menggunakan UML seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

- b. *CRC Card* fungsinya untuk mengatur dan mengenali *object oriented class* yang disesuaikan dengan pengembangannya. Ketika saat perancangan terjadi ketidaksesuaian maka akan di lakukan perbaikan
 - c. *Prototype* merupakan bagian dari perancangan yang di dalamnya berupa *user interface* dalam bentuk *wireframing* yang digunakan untuk mempermudah pengguna dalam melihat desain.
3. *Coding* (Pengkodean)

Pada tahap pengkodean yaitu proses peneliti akan menyesuaikan terhadap story pengguna yang berada di tahap perencanaan sehingga sistem yang dibangun akan sesuai, proses-proses pengkodean yang dilakukan:

 - a. *Pair programming* yaitu tahapan sistem yang akan di bangun menggunakan bahasa pemrograman dan media penyimpanan yang sudah disepakati.
 - b. *Refactory* yaitu tahapan pada saat terjadi ketidak sesuain kode program dan akan dilakukan perbaikan untuk mendapatkan hasil yang sesuai.
4. *Testing* (Pengujian)

Tahap pengujian atau biasa disebut dengan testing akan dilakukan oleh pengguna karena sebagai user dengan melakukan pengujian sesuai dengan *acceptance test* yang sudah ditentukan dan sepakati. *Unit test* yang sudah berhasil dibuat hanya berfokus pada keseluruhan fitur dan fungsional sistem. Supaya sistem dapat di simpulkan telah sesuai dengan yang dibutuhkan dan dapat dilakukan perilisan.

2.4 Aplikasi

Aplikasi merupakan salah satu jenis perangkat lunak (*software*) yang memiliki tugas penting sebagai antarmuka pengguna (*front-end*) dalam sistem komputer. Aplikasi memungkinkan pengguna untuk mengolah berbagai jenis data dan menghasilkan informasi yang bermanfaat sesuai dengan kebutuhan mereka. Selain itu, aplikasi juga dapat terintegrasi dengan sistem lainnya yang berkaitan (Sarwono, n.d.).

Seperti yang dikemukakan oleh (Soraya & Wahyudi, 2021), aplikasi berperan penting dalam mengoptimalkan penggunaan sistem komputer. Dalam konteks bisnis atau organisasi, aplikasi dapat digunakan untuk meningkatkan efisiensi operasional, mengoptimalkan pemanfaatan sumber daya, dan menghasilkan informasi yang akurat dan relevan bagi pengambilan keputusan.

Aplikasi juga memiliki peran yang krusial dalam memenuhi kebutuhan pengguna, terutama dalam era digital yang semakin berkembang pesat. Pengguna saat ini membutuhkan akses cepat dan mudah terhadap informasi yang dibutuhkan, dan aplikasi merupakan solusi yang efektif untuk memenuhi kebutuhan tersebut (Juansyah, 2015).

Dalam rangka memaksimalkan manfaat dari aplikasi, pengembang perangkat lunak perlu memperhatikan aspek-aspek seperti keamanan, performa, dan *user experience* (pengalaman pengguna). Dengan memperhatikan hal-hal tersebut, pengguna akan merasa nyaman dan terbantu dalam mengolah data dan informasi yang dibutuhkan melalui aplikasi yang mereka gunakan.

2.5 Pengelolaan Jadwal

Pengelolaan dan jadwal merupakan dua konsep yang berbeda namun saling terkait. Menurut KBBI, pengelolaan adalah tindakan atau cara mengatur, mengurus, atau mengawasi suatu kegiatan, proyek, atau sumber daya agar dapat berjalan dengan efektif dan efisien. Hal ini meliputi pengaturan sumber daya manusia, keuangan, sarana dan prasarana, serta kebijakan yang terkait dengan pengelolaan tersebut. Di sisi lain, jadwal adalah rencana waktu atau jangka waktu yang ditetapkan untuk melakukan suatu kegiatan atau acara. Jadwal bisa berupa daftar kegiatan yang harus dilakukan pada waktu-waktu tertentu, misalnya jadwal pelajaran di sekolah, jadwal kerja di kantor, atau jadwal latihan olahraga (Sallaby & Kanedi, n.d.).

Dalam konteks bisnis atau proyek, jadwal seringkali digunakan untuk mengatur waktu pelaksanaan setiap tahap atau aktivitas dalam proyek tersebut. Oleh karena itu, pengelolaan jadwal dapat diartikan sebagai tindakan atau cara mengatur dan mengawasi jadwal atau rencana waktu suatu kegiatan atau acara dengan efektif dan efisien. Hal ini mencakup pengaturan dan pengawasan waktu pelaksanaan setiap tahap atau aktivitas dalam suatu proyek atau bisnis, serta pengaturan dan pengawasan waktu pelaksanaan kegiatan atau acara secara umum.

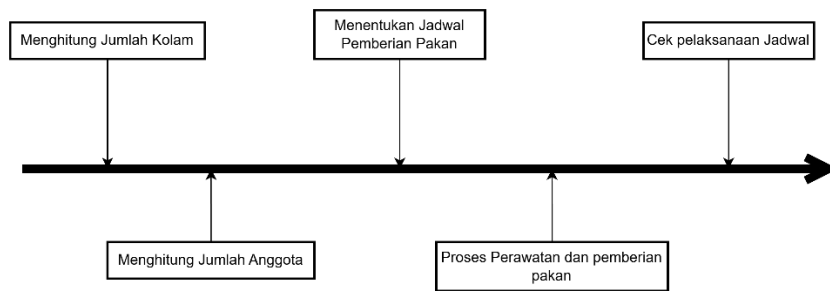
Dengan melakukan pengelolaan jadwal yang baik, diharapkan suatu kegiatan atau acara dapat dilaksanakan tepat waktu dan sesuai dengan tujuan yang diinginkan. Sebaliknya, jika pengelolaan jadwal tidak efektif dan efisien, dapat mengakibatkan keterlambatan, ketidakseimbangan, dan bahkan kegagalan dalam pelaksanaan kegiatan atau proyek tersebut. Oleh karena itu, penting bagi organisasi atau individu untuk memahami dan mengimplementasikan pengelolaan

jadwal yang baik guna mencapai tujuan yang diinginkan secara efektif dan efisien(Puspaningrum et al., 2013).

Untuk sistem penjadwalan di Pokdakan Karya Bersama sendiri menggunakan rekomendasi dari metode bioflok dengan pemberian pakan ikan dilakukan dua kali sehari, yaitu pada pagi dan sore hari. Proses pencernaan pada ikan lele memerlukan waktu sekitar 8 hingga 9 jam untuk mencerna makanan dengan sempurna. Oleh karena itu, penting untuk memastikan bahwa pemberian pakan dilakukan dengan interval waktu yang cukup. Jika pakan diberikan dengan interval kurang dari 8 jam, hal ini dapat berdampak negatif pada kesehatan ikan lele. Salah satu dampak negatif yang mungkin terjadi adalah terjadinya kembung pada ikan lele. Jika kondisi kembung parah dan tidak segera diatasi, hal ini dapat berujung pada kematian ikan lele. Oleh karena itu, menjaga jadwal pemberian pakan yang tepat sangat penting dalam sistem bioflok untuk menjaga kesehatan dan pertumbuhan ikan lele oleh karena itu pemberian pakan dilakukan dua kali sehari lebih tepatnya pada jam 9 pagi dan jam 9 malam. Di pokdakan karya bersama jadwal ditentukan dengan beberapa syarat yaitu:

- 1 . Setiap Anggota memiliki 2 jadwal (siang dan malam) untuk satu kolam dalam satu hari setiap minggunya
- 2 . Jadwal akan di ganti satu minggu sekali

Berikut jadwal pemberian pakan ikan dengan menggunakan metode bioflok selama satu minggu dan *timeline* pembuatan jadwal di pokdakan karya bersama.



Gambar 2.2 *Timeline* Penentuan Jadwal

Tabel 2.2. Jadwal Pemberian Pakan

Hari	Pagi 09.00		Malam 21.00	
Senin	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8
Selasa	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8
Rabu	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8

Hari	Pagi 09.00		Malam 21.00	
Kamis	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8
Jumat	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8
Sabtu	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8
Minggu	Andika	Kolam 1	Andika	Kolam 1
	Andi	Kolam 2	Andi	Kolam 2
	Joni Saputra	Kolam 3	Joni Saputra	Kolam 3
	Hendrik	Kolam 4	Hendrik	Kolam 4
	Romadon	Kolam 5	Romadon	Kolam 5
	Mujiono	Kolam 6	Mujiono	Kolam 6
	Suyanto	Kolam 7	Suyanto	Kolam 7
	Sunarman	Kolam 8	Sunarman	Kolam 8

2.6 Analisis SWOT

Analisis SWOT adalah kerangka kerja yang digunakan untuk menganalisis kekuatan (*strengths*), kelemahan (*weaknesses*), peluang (*opportunities*), dan ancaman (*threats*) yang terkait dengan suatu bisnis atau proyek. Analisis ini dapat

diterapkan dalam pengembangan perangkat lunak untuk membantu mengidentifikasi faktor internal dan eksternal yang dapat mempengaruhi keberhasilan pengembangan perangkat lunak (Mayang et al., n.d.).

Analisis SWOT dalam pengembangan perangkat lunak dapat membantu pengembang untuk memperkuat kekuatan mereka, mengatasi kelemahan mereka, mengeksplorasi peluang yang ada, dan mengatasi ancaman yang mungkin timbul. Dalam pengembangan perangkat lunak yang sukses, penting untuk mempertimbangkan faktor internal dan eksternal yang dapat mempengaruhi keberhasilan proyek dan untuk mengembangkan strategi untuk mengatasi faktor-faktor tersebut.

2.7 *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang digunakan untuk membangun, mendokumentasikan, dan menspesifikasi perangkat lunak. UML adalah metodologi yang digunakan dalam mengembangkan sistem berorientasi objek dan alat yang dapat mendukung pengembangan sistem. UML berfungsi sebagai bahasa yang berbasis grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan *software* berbasis OO (*Object-Oriented*). Dalam UML, terdapat standar penulisan sistem blue print yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang dibutuhkan dalam sistem software (Suendri, 2018). Terdapat beberapa jenis diagram dalam UML seperti diagram *use case*, *diagram class* dan *diagram activity* yang membantu dalam memodelkan dan menggambarkan interaksi dan komponen-komponen

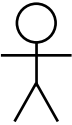

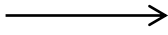
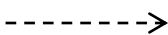
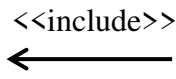
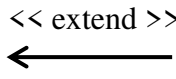
dalam sebuah sistem. Adapun penjelasan dari jenis-jenis diagram tersebut sebagai berikut:

2.7.1. Use case diagram

Use case diagram merupakan pemodelan untuk menggambarkan perilaku (*behavior*) sistem yang akan dibuat. *Diagram usecase* mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem. *Diagram usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Sari & Alita, 2022).

Adapun simbol-simbol yang terdapat pada *Use case diagram* terdapat pada tabel dibawah ini.

Tabel 2.3. Tabel *Use case diagram*



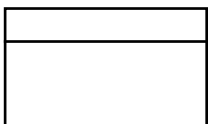

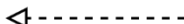
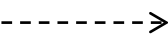
Nama	Simbol	Keterangan
<i>Aktor</i>		Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
<i>Use Case</i>		Abstraksi dan interaksi antara sistem dan aktor
<i>Association</i>		Abstraksi dari penghubung antara aktor dengan use case
<i>Generalisasi</i>		Menunjukkan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


2.7.2. Class diagram

Class diagram merupakan model yang menggambarkan struktur dan deskripsi class serta dapat menghubungkan antara class yang lain. *Class diagram* menjelaskan model yang digunakan dalam perancangan atribut dan fungsi-fungsi yang akan digunakan untuk membangun sistem baru (Ramadhanu & Priandika, 2021).

Adapun simbol-simbol yang terdapat pada *Class diagram* terdapat pada tabel dibawah ini.

Tabel 2.4 Tabel *Class diagram*

Nama	Simbol	Keterangan
<i>Generalication</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>Navy Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
<i>Realication</i>		Operasi yang benar-benar dilakukan oleh suatu objek
<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



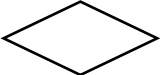


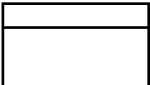
<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya
--------------------	---	---

2.7.3. Activity diagram

Activity Diagram menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity diagram* juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan (Suendri, 2018).

Adapun simbol-simbol yang terdapat pada *Activity diagram* terdapat pada tabel dibawah ini.

Tabel 2.5. Tabel *Activity diagram*

Nama	Simbol	Keterangan
Status awal		Sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan		Percabangan dimana ada pilihan aktivitas yang lebih dari satu
Penggabungan		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
<i>Swinlane</i>		<i>Swinlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.7.4. CRC diagram

Dalam perancangan sebuah perangkat lunak, salah satu tahap yang paling penting adalah membuat CRC Card dan Class Diagram. CRC Card berguna untuk mendeskripsikan kelas apa saja yang akan dipakai beserta fungsionalitas yang dibutuhkan dan hubungannya dengan kelas lain. Class Diagram tidak jauh berbeda dengan CRC Card, hanya saja terdapat penjelasan-penjelasan lebih detail seperti tipe data dan enkapsulasi fungsi serta atribut dari suatu kelas (Armanda Tyas, 2020).

2.8 Java

Java adalah bahasa pemrograman dan platform komputasi yang pertama kali dirilis oleh Sun Microsystems pada tahun 1995. Pengguna aplikasi biasanya menggunakan *Java Runtime Environment (JRE)* yang diinstal di komputer mereka sendiri untuk menjalankan aplikasi *Java* atau aplikasi *Java* di browser web. Untuk membuat dan mengembangkan aplikasi berbasis Java diperlukan *Java Development Kit (JDK)*, saat ini JDK dilisensikan oleh Oracle Corporation yang secara resmi dibeli oleh Sun Microsystems pada awal tahun 2010. Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. (Muharam & Fauziah, 2020).

2.9 Kotlin

Kotlin merupakan bahasa pemrograman berbasis *Java Virtual Machine (JVM)* yang dikembangkan oleh JetBrains. Kotlin merupakan bahasa pemrograman yang pragmatis untuk *Android* yang mengkombinasikan *object*

oriented (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java.(Iqbal & Wali, 2022).

2.10 *Representational State Transfer (REST)*

REST (Representational State Transfer) adalah suatu arsitektur komunikasi berbasis web yang digunakan dalam pengembangan layanan berbasis web. *REST* umumnya menggunakan protokol HTTP (*Hypertext Transfer Protocol*) sebagai protokol untuk komunikasi data. Roy Fielding memperkenalkan *REST* pertama kali pada tahun 2000. Format umum yang digunakan dalam *REST* adalah JSON dan XML (Muharam & Fauziah, 2020).

Restful web service merupakan salah satu kriteria desain *web service* yang paling sering digunakan. Restful bekerja dengan cara *resource-oriented*, di mana client (*requester*) mengakses layanan yang ditawarkan oleh server dengan mengakses URL dari *resource* menggunakan method pada HTTP. Dalam dunia web *API*, protokol yang digunakan adalah HTTP. *APIclient* dapat berinteraksi dengan *API* dengan mengirimkan berbagai jenis pesan HTTP yang memiliki standar pesan yang ditentukan. Standar-standar pesan tersebut yaitu:

1. *GET* (Method *Get* mengambil data dari web server dengan menentukan parameter di bagian URL dari permintaan.
2. *DELETE* (Method *Delete* menghapus sumber daya)
3. *POST* (Method *Post* memanfaatkan badan pesan untuk mengirim data ke server web)
4. *PUT* (Method *Put* mirip dengan post memanfaatkan badan pesan untuk mentransfer data)

2.11 Java Script Object Notation (JSON)

Java Script Object Notation (JSON) adalah sebuah format data yang memungkinkan aplikasi untuk saling berkomunikasi di dalam sebuah jaringan, yang melalui *restfulApi*. JSON menyediakan semua bahasa pemrograman modern dengan bantuan untuk menghasilkan dan menerima data (Muharam & Fauziah, 2020).

2.12 Database

Database adalah sebuah system yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. *Database* terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. *Database* digital di manage menggunakan *Database Management System* (DBMS), yang menyimpan isi *database*, mengizinkan pembuatan dan *maintenance* data dan pencarian dan akses yang lain (Ramadhan & Mukhaiyar, 2020).

2.13 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. Tag Mysql adalah “*The World's most popular open-source database*”. MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql, anda dapat menggunakan software seperti phpmyadmin dan mysql yog (Ramadhan & Mukhaiyar, 2020).

2.14 JSON Web Token (JWT)

JWT adalah sebuah token yang digunakan untuk *autentikasi* dan pertukaran informasi. Token ini berbentuk string JSON yang sangat padat sehingga dapat dikirim melalui URL, parameter HTTP POST atau dalam Header HTTP. Keuntungan dari penggunaan token JWT adalah ukurannya yang kecil sehingga dapat ditransmisikan dengan lebih cepat dan memiliki informasi mandiri yang tidak perlu *query* ke basis data lebih dari satu kali. Proses *Login* menggunakan *session*, tetapi *API* hanya menggunakan konsep JWT yang disebut "jot". Token JWT dapat di-sign secara digital menggunakan *secret* atau pasangan *public/private key*. Struktur JWT terdiri dari tiga bagian: *header*, *payload*, dan *signature*. Cara JWT beroperasi serupa dengan kata sandi, ketika pengguna berhasil masuk (*login*), server akan menghasilkan token yang akan disimpan di penyimpanan lokal (*local storage*) atau kuki (*cookies*) pada peramban (*browser*). (Yoda Gustiegan, 2022).

2.15 Insomnia

Insomnia adalah aplikasi desktop lintas platform gratis yang digunakan untuk berinteraksi dengan dan merancang *API* berbasis HTTP (Satyra Fajar Kusumah et al., 2019).

2.16 Black Box Testing

Metode Black Box Testing adalah salah satu pendekatan pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas atau perilaku aplikasi tanpa memperhatikan detail internal atau logika kode sumber. Dalam metode ini, pengujian dilakukan dari sudut pandang eksternal seperti yang akan

dilihat oleh pengguna akhir, tanpa pengetahuan tentang bagaimana sistem diimplementasikan.

Berikut adalah beberapa poin penting tentang metode Black Box Testing:

- a. Fokus pada Input dan Output: Dalam Black Box Testing, fokus utama adalah menguji bagaimana sistem merespons berbagai input yang diberikan dan menghasilkan output yang diharapkan. Tujuannya adalah memastikan bahwa aplikasi berperilaku sesuai dengan spesifikasi dan persyaratan fungsional.
- b. Tidak Memperhatikan Struktur Kode: Pengujian ini tidak memperhatikan kode sumber atau detail implementasi aplikasi. Seorang pengujian tidak perlu tahu bagaimana kode di dalamnya tertata.
- c. Menggunakan Spesifikasi dan Dokumentasi: Untuk melakukan Black Box Testing, pengujian biasanya mengacu pada dokumen spesifikasi, persyaratan fungsional, kasus penggunaan, atau dokumentasi lainnya yang menjelaskan perilaku yang diharapkan dari sistem.
- d. Pengujian Fungsional: Black Box Testing adalah bentuk pengujian fungsional di mana aplikasi diuji untuk memastikan bahwa semua fungsi yang diharapkan bekerja dengan benar. Ini termasuk pengujian input yang valid dan tidak valid, aliran proses, interaksi dengan sistem lain, dan sebagainya.
- e. Pengujian Berbagai Skenario: Pengujian Black Box mencakup berbagai skenario penggunaan yang mungkin terjadi dalam aplikasi, termasuk pengujian kasus penggunaan positif (sesuai dengan spesifikasi) dan

pengujian kasus penggunaan negatif (input yang tidak valid atau situasi tidak terduga).

- f. **Black Box Testing Tools:** Ada berbagai alat otomatis yang dapat digunakan untuk mengotomatisasi pengujian Black Box, seperti Selenium untuk pengujian web atau Postman untuk pengujian REST API.
- g. **Keuntungan:** Keuntungan utama dari Black Box Testing adalah bahwa pengujian dapat dilakukan oleh tim yang tidak memiliki pengetahuan mendalam tentang struktur kode. Ini memungkinkan tim pengujian untuk berfokus pada persyaratan fungsional dan pengalaman pengguna.
- h. **Keterbatasan:** Keterbatasan utama adalah bahwa metode ini mungkin tidak mengidentifikasi masalah internal seperti bug logika atau masalah kinerja yang terkait dengan struktur internal aplikasi.

Metode Black Box Testing adalah komponen penting dari strategi pengujian perangkat lunak yang menyeluruh. Ini membantu memastikan bahwa aplikasi berperilaku sesuai dengan yang diharapkan oleh pengguna akhir dan memenuhi persyaratan fungsional yang telah ditentukan. (Pradana Putra et al., 2020).

2.17 Retrofit

Retrofit merupakan REST client library yang aman untuk *Android* dan *java*. Dibuat oleh Square. *Retrofit* menyediakan cara yang aman untuk autentikasi dan interaksi dengan berbagai API lainnya. Sehingga memungkinkan pengiriman permintaan jaringan dengan OkHttp. *Retrofit* mengambil data JSON atau XML dari web API dan saat data diterima akan langsung diubah ke Plain Old Java Object (POJO). Sehingga harus ditentukan setiap class yang akan dipakai saat

response diterima (Codepath,2015). *Retrofit* juga bekerja dengan *REST API* menggunakan implementasi java interface, yang dapat dihasilkan dengan bantuan RestAdapter. Implementasi dalam hal ini bertindak sebagai local instance dari layanan dan setiap panggilan sesuai dengan permintaan HTTP (Iqbal & Wali, 2022)

2.18 Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode *Android* di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.(Lumba, 2021)