

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 *Software Development*

Pengembangan perangkat lunak (*software development*) adalah proses pembuatan, perancangan, implementasi, dan pemeliharaan perangkat lunak atau aplikasi komputer. Tujuan dari pengembangan perangkat lunak adalah menciptakan program komputer yang memenuhi kebutuhan pengguna, baik dalam hal fungsionalitas, kinerja, keamanan, maupun kegunaan. Proses pengembangan perangkat lunak melibatkan analisis kebutuhan, perancangan, pengkodean, pengujian, dan pemeliharaan, serta dapat mengikuti berbagai Smetodologi pengembangan seperti model *waterfall*, model iteratif, atau model agile. Pengembangan perangkat lunak merupakan disiplin yang luas dan penting dalam dunia teknologi informasi (Muhammad et al., 2023).

Cara kerja *software development* melibatkan serangkaian tahapan yang dimulai dari analisis kebutuhan pengguna, perancangan konsep, pengembangan kode, uji coba, hingga peluncuran produk final. Pengembangan perangkat lunak melibatkan serangkaian tahap dan aktivitas, yang umumnya mencakup (Ahmad et al., 2020):

1. Perencanaan.

Identifikasi tujuan dan kebutuhan sistem, serta perencanaan proyek termasuk alokasi sumber daya dan jadwal.

2. Analisis.

Pemahaman mendalam tentang kebutuhan pengguna dan masalah yang akan dipecahkan oleh perangkat lunak. Ini melibatkan identifikasi fitur, fungsionalitas, dan batasan sistem.

3. Perancangan.

Membuat desain sistem secara rinci, termasuk arsitektur, pola desain, user interface, dan rencana database.

4. Implementasi.

Proses penulisan dan pengkodean perangkat lunak berdasarkan desain yang telah dibuat. Ini adalah langkah di mana kode sumber sebenarnya dibangun.

5. Uji Coba.

Pengujian perangkat lunak untuk memastikan bahwa sistem berfungsi sesuai yang diharapkan, dan untuk mengidentifikasi dan memperbaiki bug atau masalah.

6. Pengiriman.

Penyampaian perangkat lunak kepada pengguna atau pelanggan.

7. Pemeliharaan.

Menjaga dan memperbarui perangkat lunak seiring berjalannya waktu, termasuk penambahan fitur, perbaikan keamanan, dan perbaikan kesalahan.

Kelebihan dari *software development* melibatkan kemampuan untuk merespons dengan cepat terhadap perubahan kebutuhan pengguna, meningkatkan efisiensi operasional, dan memberikan solusi inovatif untuk menyelesaikan berbagai tantangan bisnis. Dan berikut adalah beberapa kelebihanannya (Aji et al., 2021):

1. Automatisasi

Perangkat lunak dapat mengotomatisasi tugas-tugas yang repetitif dan membebaskan waktu manusia untuk tugas-tugas lebih kreatif.

2. Skalabilitas

Perangkat lunak dapat ditingkatkan dan disesuaikan dengan pertumbuhan bisnis atau kebutuhan pengguna.

3. Akurasi

Perangkat lunak dapat melakukan perhitungan dan pemrosesan dengan akurasi tinggi.

4. Efisiensi

Memungkinkan efisiensi operasi, penghematan biaya, dan pengurangan kesalahan manusia.

Meskipun memiliki banyak kelebihan, *software development* juga memiliki beberapa kekurangan. Beberapa kekurangan tersebut antara lain (Kurniawan & Sani, 2019):

1. Biaya

Pengembangan perangkat lunak dapat mahal, terutama untuk proyek besar.

2. Kesalahan

Kesalahan dalam perangkat lunak dapat memiliki dampak yang serius, termasuk kegagalan sistem.

3. Kebutuhan Sumber Daya

Memerlukan sumber daya seperti komputer, server, dan personil yang terlatih.

4. Perubahan Kebutuhan

Kebutuhan pengguna atau perubahan bisnis dapat menyebabkan perangkat lunak harus diubah atau diperbarui.

Berbagai metode *software development* telah dikembangkan untuk membimbing dan mengelola proses pengembangan perangkat lunak. Beberapa kalimat mengenai metode *software development* melibatkan Menurut (Muhammad et al., 2023):

1. *Waterfall* (Model Air Terjun) Metode ini memiliki tahapan yang terstruktur dan linear, dimulai dari perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Setiap tahap harus selesai sebelum memasuki tahap berikutnya.
2. *Agile* Metode ini berfokus pada kolaborasi tim, komunikasi yang kuat, dan fleksibilitas. Agile melibatkan iterasi singkat dan sering, dengan

pengembangan yang adaptif dan respon terhadap perubahan kebutuhan pengguna.

3. *Scrum* adalah kerangka kerja Agile yang menekankan kolaborasi tim, perencanaan sprint (periode waktu tertentu), dan evaluasi reguler. Proyek dibagi menjadi tugas yang dapat diselesaikan dalam sprint.
4. Kanban Metode ini memvisualisasikan aliran kerja dengan papan kanban dan kartu tugas. Tim menarik tugas dari tahap satu ke tahap berikutnya seiring dengan kemajuan.
5. *Extreme Programming (XP)* Metode ini berfokus pada pengujian yang ekstensif, integrasi yang sering, dan kolaborasi yang erat antara pengembang. Prinsip-prinsip XP mencakup pemrograman berpasangan, pengujian otomatis, dan pengiriman perangkat lunak yang sederhana dan mudah dimengerti.
6. DevOps adalah pendekatan yang mengintegrasikan pengembangan (*Development*) dan operasi (*Operations*) untuk mempercepat pengiriman perangkat lunak dan memperbaiki kerja sama antara tim pengembang dan tim operasi.
7. *Rapid Application Development (RAD)* Metode ini berfokus pada pengembangan yang cepat dan responsif terhadap kebutuhan pengguna. Prototipe dan iterasi digunakan untuk menghasilkan perangkat lunak dengan cepat.

8. *Lean Software Development* Terinspirasi oleh prinsip-prinsip Lean Manufacturing, metode ini menekankan pengurangan limbah, pengiriman yang cepat, dan nilai yang diberikan kepada pengguna.
9. *Spiral Model* Metode ini adalah kombinasi dari pendekatan siklus hidup tradisional dan iteratif. Proses bergerak dalam siklus spiral dan melibatkan tahap perencanaan, analisis risiko, desain, implementasi, dan evaluasi.
10. *Big Bang Model* ini adalah metode yang kurang terstruktur di mana pengembangan dimulai tanpa perencanaan atau desain yang jelas. Perangkat lunak dikembangkan tanpa tahapan yang terstruktur.
11. *Prototyping* metode ini melibatkan pembuatan prototipe perangkat lunak untuk memahami kebutuhan pengguna. Prototipe ini digunakan untuk mengarahkan pengembangan perangkat lunak sebenarnya.

Setiap metode memiliki karakteristik dan kecocokan yang berbeda tergantung pada jenis proyek dan tujuan pengembangan perangkat lunaknya. Pemilihan metode harus mempertimbangkan kebutuhan proyek, tim pengembang, dan lingkungan kerja.

2.1.2 Implementasi *Software Development*

Menurut (Muhammad et al., 2023) Implementasi perangkat lunak dapat dilakukan dalam berbagai bentuk, termasuk aplikasi desktop, aplikasi web, aplikasi *mobile*, perangkat lunak berbasis *cloud*, perangkat lunak *open source*, dan banyak lainnya. Penggunaan perangkat lunak

sangat luas dan mencakup berbagai sektor seperti bisnis, pendidikan, hiburan, kesehatan, industri, dan pemerintahan. Contohnya termasuk sistem manajemen perusahaan, aplikasi perbankan online, permainan video, dan banyak lagi.

Berikut beberapa contoh implementasi lebih lanjut dari pengembangan perangkat lunak:

1. Sistem Manufaktur

Pengembangan perangkat lunak yang mengotomatisasi dan mengoptimalkan proses manufaktur dalam industri.

2. Sistem Informasi Keuangan

Pembuatan sistem informasi keuangan untuk manajemen keuangan perusahaan dan pelaporan keuangan.

3. Sistem *E-commerce*

Pengembangan platform e-commerce untuk perdagangan online, termasuk situs web dan aplikasi mobile.

4. Sistem Pendidikan

Pembuatan perangkat lunak pendidikan untuk pembelajaran online, manajemen sekolah, dan evaluasi siswa.

5. Sistem Kesehatan

Pengembangan sistem informasi kesehatan untuk catatan medis elektronik, manajemen rumah sakit, dan aplikasi kesehatan.

6. Sistem Transportasi

Pembuatan perangkat lunak untuk manajemen logistik, perutean, dan pelacakan kendaraan.

7. Sistem Perbankan

Pengembangan perangkat lunak perbankan untuk layanan perbankan online, ATM, dan manajemen transaksi keuangan.

2.1.3 Metode SDLC (*Systems Development Life Cycle*).

Menurut (Samudra et al., 2023) *Software Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model - model dan metodologi yang digunakan orang untuk mengembangkan sistem - sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara -cara yang sudah teruji baik).

SDLC memberikan kerangka kerja yang terdiri dari langkah-langkah yang harus diikuti untuk mengembangkan perangkat lunak. Langkah kerja ini mencakup perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Metodologi SDLC adalah dasar bagi berbagai jenis pendekatan pengembangan perangkat lunak. metodologi ini membentuk kerangka kerja untuk merencanakan dan mengendalikan pembuatan sistem informasi. Ada tiga jenis metode siklus hidup sistem yang umum digunakan, yaitu siklus hidup sistem tradisional, siklus hidup menggunakan *prototyping*, dan siklus hidup sistem berorientasi objek (Gunadarma University Staff Site, 2019).

SDLC memiliki berbagai fungsi yang penting yaitu sebagai berikut (Ultariani et al., 2020):

1. Sarana Komunikasi: SDLC memberikan kerangka kerja yang membantu tim pengembang berkomunikasi dengan pemangku kepentingan. Ini membantu memastikan bahwa semua pihak memiliki pemahaman yang sama tentang proyek perangkat lunak.
2. Pembagian Peran dan Tanggung Jawab: SDLC membantu membagi peran dan tanggung jawab dengan jelas. Ini memastikan bahwa setiap anggota tim tahu apa yang diharapkan dari mereka dalam setiap tahap pengembangan.
3. Gambaran Input dan Output: SDLC memberikan gambaran yang jelas tentang apa yang diharapkan sebagai input dan output dari setiap tahap. Ini membantu memandu proses pengembangan.

Metode Software Development Life Cycle (SDLC) bekerja dengan mendefinisikan serangkaian tahapan yang sistematis dalam pengembangan perangkat lunak, dimulai dari perencanaan hingga implementasi, evaluasi, dan pemeliharaan dan berikut cara kerja metode tersebut meliputi (Romli et al., 2023) :

1. Perencanaan Sistem (*Systems Planning*) : Tahap ini menekankan pada studi kelayakan pengembangan sistem. Ini termasuk membentuk tim pengembang, mendefinisikan tujuan dan ruang lingkup pengembangan, mengidentifikasi masalah yang dapat dipecahkan melalui pengembangan sistem, menentukan strategi, dan menentukan prioritas teknologi.

2. Analisis Sistem (*Systems Analysis*): Tahap ini melibatkan analisis mendalam tentang bagaimana sistem akan berjalan. Ini termasuk menganalisis kelebihan dan kekurangan sistem, fungsi sistem, dan perbaikan yang mungkin diperlukan.
3. Perancangan Sistem (*Systems Design*): Tahap ini menghasilkan desain sistem yang mencakup komponen dan dokumen yang diperlukan untuk mewujudkan proyek tersebut. Ini mencakup pembuatan prototipe, merinci operasi sistem, mengklasifikasikan masalah, dan mendefinisikan kebutuhan sistem.
4. Implementasi Sistem (*Systems Implementation*): Tahap implementasi adalah tahap di mana rancangan dari tahap-tahap sebelumnya diimplementasikan dalam bentuk nyata. Ini mencakup pembuatan database, pembuatan aplikasi, serta pengujian dan debugging.
5. Pemeliharaan Sistem (*Systems Maintenance*): Setelah sistem diimplementasikan, tahap pemeliharaan diperlukan untuk memastikan bahwa sistem tetap beroperasi dengan baik. Ini melibatkan pemeliharaan database, aplikasi, serta penyesuaian sistem dengan perubahan kebutuhan. SDLC membantu memastikan bahwa pengembangan perangkat lunak dilakukan dengan metode yang terstruktur dan memungkinkan identifikasi dan perbaikan masalah sepanjang proses pengembangan (Ahmad Lutfi, 2023).

2.1.4 *Location Based Service.*

(LBS) *Location Based Service* adalah fitur yang memungkinkan pengguna untuk mengakses informasi berdasarkan lokasi geografis mereka menggunakan teknologi lokasi, seperti satelit. Dalam penelitian ini, metode SDLC pengembangan perangkat lunak berbasis prototipe digunakan untuk mengembangkan aplikasi LBS (Pidu, 2023).

2.1.5 *GPS (Global Positioning System)*

(GPS) *Global Positioning System* merupakan sebuah sistem navigasi dan penentuan posisi melalui satelit yang dimiliki dan dioperasikan oleh Amerika Serikat. Tujuan utama dari sistem ini adalah untuk menyediakan informasi tentang posisi, kecepatan dalam tiga dimensi, dan informasi waktu. Sistem GPS terdiri dari tiga segmen utama, yaitu segmen angkasa, segmen kontrol atau pengendali, dan segmen pengguna (Kalifia, 2023).

1. Segmen Angkasa: Bagian ini melibatkan 24 satelit yang mengorbit Bumi dalam enam orbit berbeda pada ketinggian sekitar 20.200 km. Setiap satelit memiliki periode orbit sekitar 12 jam, yang berarti mereka akan kembali ke posisi yang sama setiap 12 jam. Satelit-satelit ini membantu dalam mentransmisikan sinyal ke perangkat GPS di Bumi.
2. Segmen Kontrol/Pengendali : Pusat pengendali utama GPS terletak di Colorado Springs, Amerika Serikat, dan bertanggung jawab atas pengaturan dan pemeliharaan sistem GPS secara keseluruhan. Terdapat juga lima stasiun pemantau lainnya dan tiga antena yang tersebar di seluruh dunia untuk memantau dan mengendalikan satelit-satelit GPS.

3. Segmen Pengguna: Segmen ini mencakup semua perangkat GPS yang digunakan oleh pengguna, seperti ponsel pintar, kendaraan, dan perangkat navigasi lainnya. Perangkat GPS menerima sinyal dari satelit-satelit GPS dan menghitung posisi dan waktu berdasarkan informasi tersebut.

GPS telah menjadi teknologi yang sangat penting dalam berbagai aplikasi, termasuk navigasi kendaraan, penerbangan, penelitian ilmiah, pemetaan, dan banyak lagi. Informasi posisi yang akurat dan waktu yang disediakan oleh GPS telah mengubah cara kita bergerak dan berkomunikasi di seluruh dunia.(Budiawan et al., 2019).

GPS (*Global Positioning System*) adalah sistem navigasi berbasis satelit yang terdiri dari paling tidak 24 satelit. GPS beroperasi dalam berbagai kondisi cuaca, di seluruh dunia, 24 jam sehari, dan tanpa biaya berlangganan. Dalam konteks penelitian ini, metode yang digunakan untuk menganalisis dan merancang sistem adalah metode Object Oriented Analysis and Design (Analisis dan Desain Berbasis Objek) dengan menggunakan Unified Modeling Language (Suroso & Seta, 2021).

2.1.6 Aplikasi *Mobile*

Aplikasi *mobile* merupakan gabungan dari dua kata, yaitu "aplikasi" dan "*mobile*." Secara terminologi, sebuah aplikasi adalah program yang dirancang untuk menjalankan fungsi tertentu untuk pengguna atau aplikasi lainnya. Sedangkan "*mobile*" merujuk pada kemampuan perangkat untuk berpindah

dari satu tempat ke tempat lain. Dalam pengertian yang lebih lengkap, aplikasi *mobile* adalah program siap pakai yang dirancang untuk menjalankan fungsi tertentu dan dipasang pada perangkat mobile, yang dapat digunakan oleh pengguna ketika mereka berpindah-pindah atau dalam perjalanan. Aplikasi mobile seringkali dirancang khusus untuk perangkat seluler seperti smartphone dan tablet, sehingga pengguna dapat mengakses dan menggunakan aplikasi tersebut secara praktis di mana pun mereka berada (Imron *et al.*, 2021).

2.1.7 Android

Android dikenal dengan sebutan "robot hijau." Android adalah sistem operasi untuk perangkat seluler yang berbasis platform Linux. Sistem operasi Android memberikan kebebasan kepada para pengembang untuk membuat aplikasi mereka sendiri. Awalnya, Google Inc. mengakuisisi Android Inc., sebuah perusahaan pendatang baru yang mengembangkan perangkat lunak untuk ponsel. Dalam upaya untuk mengembangkan Android, Google membentuk Open Handset *Alliance*, yang merupakan konsorsium yang terdiri dari 34 perusahaan yang bergerak di bidang perangkat keras, perangkat lunak, dan telekomunikasi. Anggota konsorsium termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Android telah menjadi salah satu sistem operasi ponsel yang paling populer dan digunakan di seluruh dunia. (Suhaimi, 2019).

2.1.8 Database

Dari Database itu sendiri sering digunakan untuk menyimpan dan mengelola sekumpulan data. Setiap database memiliki API (Antarmuka

Pemrograman Aplikasi) yang khusus dirancang untuk membuat, mengakses, mengatur, mencari, dan mengambil data yang tersimpan di dalamnya. Database merupakan sebuah kumpulan file yang saling terkait dan diorganisir sedemikian rupa sehingga data tersebut dapat diakses dan dimanfaatkan oleh berbagai program atau aplikasi yang memerlukannya. Database memainkan peran penting dalam berbagai aspek kehidupan, termasuk dalam sistem informasi perusahaan, aplikasi web, dan banyak lagi, untuk menyimpan dan mengelola informasi dengan efisien. (Ultariani et al., 2020).

2.1.9 Olahraga

Kebugaran jasmani adalah kesanggupan dan kemampuan tubuh melakukan penyesuaian terhadap pembebanan fisik yang diberikan kepadanya dari kerja yang dilakukan sehari-hari tanpa menimbulkan kelelahan yang berlebihan. Faktor pandemik yang tengah mewabah dikalangan masyarakat beberapa waktu ini dan banyaknya aktifitas yang dilakukan oleh seseorang, membuat waktu yang dimiliki terbatas (Anggraini et al., 2022).

Setiap cabang olahraga memiliki fokus dan manfaat tersendiri yang dapat disesuaikan dengan preferensi dan kebutuhan individu (Noviantoro et al., 2022). Dua di antaranya yang cukup populer adalah gym dan futsal. Keduanya tidak hanya memberikan manfaat fisik, tetapi juga mendukung peningkatan kesehatan mental dan kemampuan sosial. Berikut adalah uraian mengenai gym dan futsal serta keunggulan yang mereka tawarkan:

1. Gym & Fitnees

Gym, atau pusat kebugaran, adalah tempat yang menyediakan fasilitas dan alat-alat untuk berolahraga, baik itu untuk tujuan kebugaran, pembentukan tubuh, atau rehabilitasi fisik. Di gym, orang dapat melakukan berbagai jenis latihan, seperti angkat beban, kardio, atau latihan fleksibilitas, dengan menggunakan peralatan khusus seperti dumbbell, treadmill, mesin latihan, atau bahkan mengikuti kelas kebugaran seperti yoga, pilates, dan aerobic (Jessica Luckminto, 2020).

Tujuan utama gym adalah untuk membantu individu mencapai kondisi fisik yang lebih baik, meningkatkan kesehatan jantung, meningkatkan kekuatan otot, serta menjaga kebugaran tubuh secara keseluruhan. Gym juga sering menjadi tempat untuk melakukan latihan pribadi atau bersama pelatih (personal trainer) yang memberikan panduan dan program latihan sesuai dengan kebutuhan masing-masing (Koloay et al., 2020).

2. Futsal

Futsal adalah varian dari sepak bola yang dimainkan dengan peraturan dan elemen yang lebih sederhana. Berbeda dengan sepak bola lapangan besar, futsal berlangsung di lapangan yang lebih kecil, biasanya di dalam ruangan, dan menggunakan ukuran bola yang lebih kecil. Selain itu, jumlah pemain di setiap tim juga lebih sedikit, biasanya lima orang per tim, yang membuat permainan ini lebih cepat dan dinamis. Format permainan yang lebih ringkas ini menjadikan futsal lebih fleksibel dan mudah diakses, sehingga populer di berbagai kalangan dan

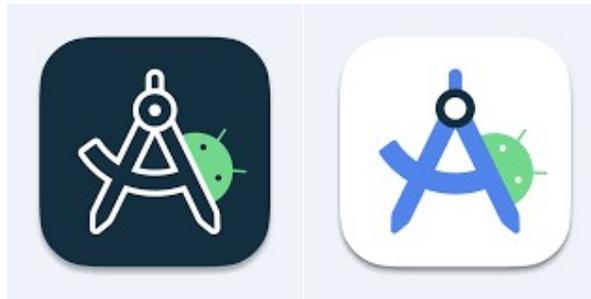
memungkinkan pemain berfokus pada teknik individu serta kerjasama tim dalam ruang yang terbatas (Syahni et al., 2021).

Untuk mengurangi dampak negatif dari penggunaan game online pada anak-anak di Kelurahan Mangunharjo, dibutuhkan langkah-langkah yang efektif dalam mengatasinya. Salah satu upaya yang dilakukan oleh tim pengabdian masyarakat adalah dengan mengadakan pelatihan futsal di Kelurahan Mangunharjo. Futsal merupakan permainan bola yang dimainkan oleh dua tim, dengan masing-masing tim beranggotakan lima orang. Pelatihan futsal ini bertujuan untuk menyediakan alternatif kegiatan positif bagi anak-anak di Kelurahan Mangunharjo, sekaligus mengurangi intensitas mereka dalam bermain game online (Septian Raibowo, 2020).

2.1.11 Sistem Informasi Geografis

Salah satu contoh pemanfaatan *smartphone* berbasis Android adalah dalam bidang geografis, di mana dikembangkan sistem terpadu yang menyediakan berbagai fasilitas untuk membantu pengolahan, penyimpanan, dan organisasi informasi geografis. Sistem ini memungkinkan pengguna untuk mengakses, menganalisis, dan memvisualisasikan data geografis secara lebih efisien, yang dapat digunakan dalam berbagai konteks, seperti pemetaan, manajemen sumber daya alam, perencanaan perkotaan, dan banyak lagi. Pemanfaatan *smartphone* Android dalam GIS memungkinkan akses data geografis secara lebih mudah dan cepat, serta memungkinkan para pengguna untuk mengambil keputusan berdasarkan informasi spasial.(Licantik, 2020).

2.1.12 Android Studio



Gambar 2.1 Android Studio

Sumber: (Suhaimi, 2019)

Android Studio menyediakan berbagai alat dan fitur yang memungkinkan pengembang untuk merancang, mengkode, menguji, dan menerapkan aplikasi Android dengan mudah dan efektif. Dengan demikian, pengembang dapat menggunakan satu perangkat lunak (Android Studio) untuk mengelola seluruh siklus pengembangan aplikasi Android, mulai dari pemodelan hingga pengujian dan distribusi aplikasi ke pengguna akhir. (Daffa & Chrisnawati, 2022).

2.1.13 Bahasa Pemrograman Dart



Gambar 2.2 Dart Logo

Sumber: (bidin A, 2020)

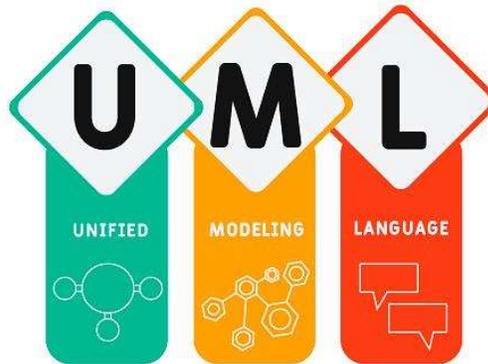
Dart adalah bahasa pemrograman yang dikembangkan oleh Google, dirancang untuk membangun aplikasi web, server, dan aplikasi mobile. Dart pertama kali diperkenalkan pada tahun 2011, dan sejak itu telah berkembang pesat, terutama dalam konteks pengembangan aplikasi mobile dengan menggunakan framework Flutter (Isa Faqihuddin Hanif, 2020).

2.1.14 UML (*Unified Modeling Language*)

(UML) *Unified Modeling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rohmat & Pertiwi, 2020).

UML merupakan singkatan dari *Unified Modeling Language*. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. UML juga dapat menjadi alat bantu untuk transfer ilmu

tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya (Noviantoro et al., 2022).



Gambar 2.3 UML

Sumber: (Mubarak, 2019)

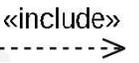
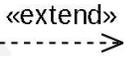
Menurut (Hanafri et al., 2019) UML menyediakan notasi grafis yang terstruktur untuk menggambarkan berbagai aspek dari sistem atau perangkat lunak, seperti entitas, hubungan, proses, dan interaksi antara komponen. Ini memungkinkan pemrogram dan pemangku kepentingan untuk lebih mudah memahami, merencanakan, dan berkomunikasi tentang sistem yang akan dibangun. Berikut beberapa jenis-jenis diagram yang terdapat pada UML (*Unified Modeling Language*):

1) *Use Case Diagram*

Use Case diagram adalah teknik yang digunakan untuk menjelaskan interaksi antara "aktor" yang berinteraksi atau beraktivitas dalam suatu sistem. Diagram ini juga menggambarkan tipe interaksi yang terjadi antara

aktor, yang merupakan pengguna sistem, dan sistem itu sendiri. *Use Case* diagram memberikan gambaran garis besar tentang urutan proses yang akan terjadi dalam sistem yang akan dikembangkan. Dalam *use case* diagram, terdapat beberapa jenis-jenis simbol atau komponen, dan berikut merupakan rinciannya (Rama, 2022):

Tabel 2.1 *Use Case* Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Peran seseorang yang nantinya akan berinteraksi dengan sistem yang berjalan
	<i>Use Case</i>	Suatu bentuk dari gambaran fungsional sistem
	<i>Association</i>	Penghubung antara sistem dengan aktor
	Generalisasi	Elemen spesialisasi dari elemen lainnya
	<i>Include</i>	Petunjuk bahwa suatu use case seluruhnya merupakan fungsionalitas dari <i>use case</i> lain
	<i>Extend</i>	Petunjuk bahwa suatu use case merupakan tambahan fungsionalitas dari <i>use case</i> lainnya apabila kondisi tertentu terpenuhi

Sumber: (Studi et al., 2024)

2) *Activity Diagram*

Activity diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan aktivitas yang terjadi dalam sistem. Diagram ini menggambarkan aliran kerja atau urutan aktivitas yang terjadi dalam proses bisnis atau sistem. *Activity diagram* sering dibuat berdasarkan *Use Case* diagram, yang menggambarkan interaksi antara pengguna (aktor) dan sistem. Aktivitas yang digambarkan dalam diagram ini dapat mencakup tugas, pengambilan keputusan, dan aliran kontrol dalam sistem. Komponen-komponen utama dalam *Activity diagram* meliputi (Siregar et al., 2023):

Tabel 2.2 *Activity Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Initial State/Status</i> Awal	dimulainya aliran kerja
	<i>Activity/Aktivitas</i>	Suatu bentuk pekerjaan yang dilakukan pada aliran kerja
	<i>Decision</i>	Suatu bentuk percabangan dimana terdapat pilihan

		kondisi yang lebih dari satu
	<i>Final State/Status</i> Akhir	bagian akhir dari suatu aliran kerja
	<i>Transition</i>	Sebagai penghubung antara aktivitas sebelumnya dan aktivitas selanjutnya

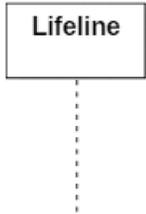
Sumber: (Studi et al., 2024)

3) *Sequence Diagram*

Sequence diagram adalah jenis *diagram* dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara objek atau entitas dalam sistem. Diagram ini memberikan gambaran tentang bagaimana objek berkomunikasi dan berinteraksi satu sama lain dalam urutan sekuensial (Hanafri et al., 2019).

Sequence diagram digunakan untuk menunjukkan serangkaian pesan atau panggilan metode yang dikirim antara objek dalam sistem. *Diagram* ini menggambarkan pesan-pesan tersebut secara berurutan, sehingga memungkinkan pemahaman yang lebih baik tentang bagaimana objek berperilaku dan berkomunikasi selama proses tertentu. Komponen-komponen utama dalam *Sequence diagram* meliputi (Putra & Andriani, 2019):

Tabel 2.3 *Sequence Diagram*

SIMBOL	NAMA	KETERANGAN
	<p style="text-align: center;"><i>Actor</i></p>	<p>Peran seseorang yang nantinya akan berinteraksi dengan sistem yang berjalan</p>
	<p style="text-align: center;"><i>Lifeline</i></p>	<p>Menghubungkan <i>sequence</i> selama objek (<i>Message</i> dikirim dan diterima serta aktivasinya.)</p>
	<p style="text-align: center;"><i>Message</i></p>	<p>Sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi</p>
	<p style="text-align: center;"><i>Reply Message</i></p>	<p>Jenis pesan yang mewakili informasi yang dikirimkan</p>

		kembali ke pengirim pesan berdasarkan pesan sebelumnya
	<i>Include</i>	Jenis pesan yang mewakili permohonan pesan dari <i>Lifeline</i> yang sama

Sumber: (Studi et al., 2024)

4) *Class Diagram*

Class diagram, juga dikenal sebagai diagram kelas, adalah jenis *diagram* dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan digunakan untuk membangun sistem. *Diagram* ini menggambarkan kelas-kelas, atribut, dan hubungan antara kelas-kelas dalam sistem. Komponen-komponen utama dalam *Class diagram* meliputi (Rohmat & Pertiwi, 2020) .

Tabel 2.4 *Class Diagram*

SIMBOL	NAMA	KETERANGAN
	<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>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua 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 <i>actor</i>
	<i>Realization</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 pada elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objeklainnya

Sumber: (Studi et al., 2024)

2.2 Penelitian Terdahulu

Riset atau penelitian terdahulu sangat penting dalam suatu bagian ilmiah. Fungsi dari riset terdahulu untuk menguatkan konsep dan fakta yang berhubungan atau saling mempunyai peran antar variable (Fauzi et al., 2022). Penelitian terdahulu ini menjadi dasar atau menjadi tolak ukur untuk penelitian yang sedang dilakukan.

- a. (Benny Korain et al., 2024) (Vol. 7 No. 5, Oktober 2023), “Aplikasi Pencarian Tempat Penyewaan Lapangan Bola Basket Di Kota Malang Berbasis Android Dengan Metode *Location Based Service*” di dalam

penelitian ini memiliki kesamaan tujuan penelitian, beberapa daerah di kota malang masih memiliki banyak bangunan, sehingga mencari lapangan menjadi sulit. Oleh karena itu di butuhkan suatu aplikasi yang dapat memberikan informasi mengenai lokasi lapangan basket di kota malang. Merancang system untuk menemukan lokasi tempat penyewaan lapangan basket di kota malang dan menerapkan layanan *Location Based Service* (LBS) dalam mencari lapangan bola basket di kota malang.

- b. (Licantik & Nova Noor Kamala Sari, 2020) (Vol. 14 No. 1, Januari 2020) “Sistem Informasi Geografis Fasilitas Kesehatan Bpjs Di Kota Palangka Raya Berbasis Android ” dari penelitian ini menghasilkan sebuah aplikasi tempat pelayanan BPJS Di Kota Palangka Raya, dengan dirancangnya sistem informasi geografis ini kita dapat memetakan fasilitas kesehatan yang melayani BPJS sehingga diharapkan dapat membantu dan memfasilitasi masyarakat dalam mencari informasi letak lokasi fasilitas kesehatan tersebut.
- c. (Andryani, 2020) (VOL.14 No. 03, Maret 2020) “Aplikasi *Mobile Location Based Service* Untuk Informasi Kolam Renang Di Kota Bekasi” penelitian ini mengangkat masalah tentang pencarian lokasi kolam renang, dimana sangat minimnya informasi terkait tempat dan sarana prasana tempat, maka di bangunla Aplikasi ini dibuat guna memudahkan pengguna *smartphone* berbasis *android* untuk dapat memanfaatkan waktu dengan efisien dalam mencari kolam renang dengan mudah dan praktis, terutama untuk masyarakat di kota Bekasi.

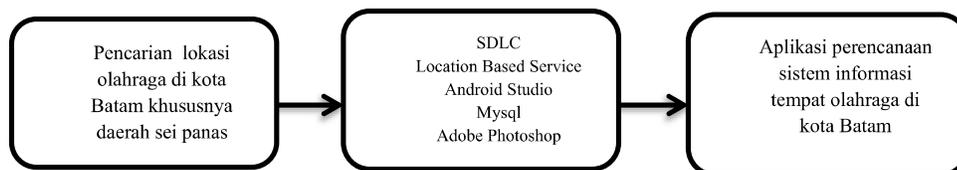
- d. (Suroso & Seta, 2021) (Vol. 9 No.1, Juni 2021) “Perancangan Sistem Pemantauan Kendaraan Pengiriman Barang Menggunakan Gps Pada Pt. Albi Berbasis Web” di dalam penelitian ini memiliki masalah terkait informasi posisi lokasi sebuah kendaraan, dengan adanya teknologi GPS ini sangat membantu perusahaan dalam memantau rute dan juga akses lokasi yang di tempu oleh kendaraan perusahaan tersebut. Sehingga memudahkan perusahaan mengecek secara *real time* menggunakan aplikasi tersebut di perusahaannya.
- e. (Muhkamaat Abdillah & Ika Kurniati, 2019) (Vol. 7 No.2, November 2019) “Penerapan Metode *Haversine* pada Aplikasi Layanan Perbaikan Kendaraan Berbasis *Location Based Service*” kasus kendaraan rusak secara mendadak tersebut adalah dengan membangun sebuah aplikasi android untuk memberikan informasi mengenai bengkel kendaraan yang berada di sekitar lokasi kendaraan pengguna yang mengalami kerusakan menggunakan *Location Based Service*. *Location Based Service* merupakan layanan yang mampu mendeteksi letak pengguna berada dengan memanfaatkan teknologi *Global Positioning System* yang terintegrasi dengan *smartphone*.

2.3 Kerangka Pemikiran

Di suatu kota yang aktif dan sehat, olahraga adalah bagian penting dari kehidupan sehari-hari. Namun, warga kota ini sering menghadapi masalah dalam mencari tempat olahraga yang sesuai dengan kebutuhan mereka.

Beberapa ingin berlari di taman terbuka, sementara yang lain mencari fasilitas kebugaran dalam ruangan. Di tengah tantangan ini, seorang pengembang aplikasi memiliki ide untuk membantu warga kota dalam mencari tempat olahraga dengan mudah.

Oleh karena itu, dibutuhkanlah sebuah aplikasi berbasis android yang dimana kedepannya akan digunakan secara umum, agar masyarakat dapat mengakses aplikasi dan mendapatkan informasi lokasi dimana tempat olahraga tersebut. Dengan adanya aplikasi informasi teknologi GPS berbasis Android, informasi dan lokasi pun dapat diperoleh secara mudah dan *realtime*. Tanpa perlu lagi mencari cari lokasi secara langsung. Berdasarkan uraian diatas, maka kerangka pemikiran dalam penelitian dan perancangan aplikasi ini digambarkan sebagai berikut :



Gambar 2.4 Kerangka Pemikiran

Kerangka pemikiran untuk merancang dan membangun aplikasi informasi tempat olahraga di Kota Batam dengan menggunakan metode Aplikasi Location-Based Service (LBS) dan metode penelitian *Software Development Life Cycle* (SDLC) dapat dijelaskan sebagai berikut:

1. **Identifikasi Kebutuhan**

Aplikasi informasi tempat olahraga di Kota Batam berbasis Android menghadapi masalah seperti kurangnya data yang terintegrasi dan terupdate, kesulitan pencarian lokasi, serta minimnya fitur ulasan dan reservasi. Selain itu, aplikasi seringkali tidak ramah pengguna, tidak kompatibel dengan berbagai perangkat, dan kurang bekerja sama dengan tempat olahraga lokal, mengurangi efektivitasnya dalam memberikan informasi yang lengkap dan relevan.

2. **Analisis Lokasi Berbasis Pelayanan (LBS)**

- a. Menganalisis potensi penerapan LBS untuk memanfaatkan informasi lokasi pengguna.
- b. Menentukan algoritma pencarian berbasis lokasi yang efektif dan efisien.

3. **Perancangan Sistem**

- a. Merancang antarmuka pengguna yang intuitif dengan fokus pada penampilan dan pengalaman pengguna yang baik.
- b. Menentukan struktur database untuk menyimpan informasi tempat olahraga dan data terkait.

4. **Implementasi**

- a. Menerapkan metode SDLC untuk membangun aplikasi berbasis Android secara bertahap.
- b. Melibatkan pengembang untuk menulis kode, mengintegrasikan fungsi LBS, dan memastikan kestabilan aplikasi.

5. Uji Coba

- a. Melakukan pengujian fungsional dan keamanan aplikasi untuk memastikan kinerja yang baik.
- b. Mengumpulkan umpan balik dari pengguna uji coba untuk perbaikan lebih lanjut.

6. Evaluasi dan Perbaikan

- a. Mengevaluasi kinerja aplikasi setelah diluncurkan dan menerima tanggapan dari pengguna.
- b. Melakukan perbaikan dan peningkatan berdasarkan umpan balik dan analisis performa.

7. Peluncuran dan Pemeliharaan

- a. Meluncurkan aplikasi ke pasar setelah memastikan kualitas dan kelayakan.
- b. Melakukan pemeliharaan rutin dan peningkatan berkelanjutan sesuai dengan kebutuhan dan perkembangan teknologi.

8. Penggunaan Metode Penelitian SDLC

- a. Menggunakan fase-fase SDLC, termasuk perencanaan, analisis, perancangan, implementasi, pengujian, dan pemeliharaan untuk memastikan proses pengembangan yang terstruktur dan terdokumentasi.

Dengan menggabungkan metode Aplikasi *Location-Based Service* dan SDLC, kerangka pemikiran ini bertujuan untuk menciptakan aplikasi yang responsif terhadap lokasi, memberikan informasi yang akurat, dan menjalani

proses pengembangan yang terkelola dengan baik. Implementasi metode ini diharapkan dapat menghasilkan aplikasi informasi tempat olahraga yang efektif dan memuaskan bagi pengguna di Kota Batam.