

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam sub-bab ini, penulis melakukan *review* literatur terdahulu yang relevan dengan topik penelitian, dengan tujuan untuk menyajikan referensi yang dapat digunakan sebagai dasar teoritis dan pertimbangan dalam penyusunan penelitian ini. Mengkaji penelitian terdahulu termasuk upaya peneliti untuk menambah pengetahuan serta penyempurnaan dalam penelitian ini.

Penelitian yang dilakukan oleh Karno Diantoro, Dian Gustana, dan Toad Maulana dengan judul "Rancang Bangun Sistem *Booking Barbershop* Dengan Metode RAD Berbasis *Mobile*" mengidentifikasi beberapa permasalahan. Diantaranya adalah pemantauan pelanggan prioritas yang belum optimal dalam pemesanan, serta ketidakteraturan antrian pemesanan yang mengakibatkan hasil pemangkasan yang kurang memuaskan dan membutuhkan waktu serta tenaga lebih karena pelanggan harus datang langsung ke *barbershop* untuk melakukan pemesanan. Untuk mengatasi masalah ini, digunakan metode *Rapid Application Development* (RAD). Hasil penelitian menunjukkan bahwa aplikasi *Booking Barbershop* yang telah dirancang mampu melacak pelanggan, sehingga mengatasi masalah antrean panjang dan memungkinkan pelanggan untuk memesan tanpa harus datang langsung ke *barbershop*. [9].

Penelitian yang dilakukan oleh Muhammad Syarif Mustapa, Sitti Suhadah, Moh. Ramdhan Arif Kaluku, dan Alfian Zakaria dengan judul "APLIKASI PEMESANAN JASA *BARBERSHOP* BERBASIS ANDROID (Studi Kasus Pada *Barbershop* Kota Gorontalo)" menghadapi masalah krusial. Dengan bertambahnya jumlah *barbershop* di Kota Gorontalo, ketersediaan informasi mengenai *barbershop* yang ada menjadi sangat penting. Hal ini membuat para pelanggan memerlukan sarana aplikasi yang dapat mempermudah akses terhadap informasi, seperti daftar layanan yang

disediakan oleh *barbershop*, *list* harga, dan profil lengkap dari masing-masing *barbershop*. Penelitian ini menerapkan metode *prototype* dalam pengembangan sistemnya. Hasil dari penelitian ini adalah pengembangan aplikasi pemesanan jasa *barbershop* yang diharapkan dapat memberikan kemudahan bagi pengguna dalam memilih *barbershop* di Kota Gorontalo sesuai dengan preferensi dan selera pribadi mereka. Aplikasi ini juga bertujuan memastikan bahwa layanan yang disediakan dapat memenuhi kebutuhan penampilan pelanggan[10].

Penelitian yang dilakukan oleh Cheong Wai Feng dan Mohd Zainuri Saringat dengan judul “*Barbershop Service Booking Application for Hair Emotion Matrix Saloon*” menghadapi situasi di mana pelanggan yang ingin mendapatkan layanan potong rambut harus menunggu giliran ketika ada banyak orang lain yang juga ingin melakukan hal yang sama. Oleh karena itu penelitian ini membuat sistem mengatur janji pemesanan pelanggan sesuai waktu dan tanggal yang diinginkan agar pelanggan tidak perlu menunggu lama di *barbershop* saat ingin mendapatkan pelayanan. Penelitian ini dibuat dengan menerapkan metode *prototype* dalam pengembangan sistemnya. Hasil dari penelitian ini adalah pengembangan sebuah aplikasi Android yang memungkinkan pelanggan untuk melakukan pemesanan layanan di *barbershop* serta memberikan kemampuan bagi pemilik usaha untuk melakukan manajemen yang lebih efisien terhadap pelanggan dan karyawan[11].

Penelitian yang dilakukan oleh Rizki Setyawatu dan Adam Bachtiar Maulachela dengan judul “Penerapan Algoritma *Dynamic Priority Scheduling* Pada Aplikasi Antrian Pencucian Mobil Berbasis *Mobile*” mengulas keluhan pelanggan terkait antrian panjang dan keterlambatan informasi tentang waktu layanan. Penelitian ini menerapkan algoritma *Dynamic Priority Scheduling* pada aplikasi antrian pencucian mobil berbasis *mobile*. Hasilnya menunjukkan bahwa algoritma tersebut berhasil diimplementasikan dalam aplikasi pencucian mobil. Algoritma ini mengatur prioritas pelanggan dengan mempertimbangkan faktor seperti jarak dan waktu

pemesanan. Dengan menggunakan algoritma *Dynamic Priority Scheduling*, urutan antrian menjadi lebih fleksibel dan sesuai dengan aturan prioritas yang telah ditetapkan.[12].

Penelitian yang dilakukan Amilia Trianasari, Baren Frianto Debatara yang berjudul “Sistem Reservasi pada Mores *Barbershop* berbasis *Web* di Jatiwarna – Bekasi” yang mulanya menyediakan layanan melalui telepon ataupun datang langsung ke tempat. Karena langsung datang ke tempat, itu mengakibatkan pelanggan ada yang tidak jadi cukur rambut di situ karena lama menunggu. Metode yang diterapkan dalam pembuatan *website* untuk menyelesaikan masalah tersebut adalah metode *action research*. Hasil yang didapatkan adalah *website* reservasi Mores *Barbershop* dengan tujuan dapat mempermudah pengolahan data pemesanan[13].

Pada penelitian yang dilakukan Muhamad Syarif, Eri Bayu Pratama dengan judul “ANALISIS METODE PENGUJIAN PERANGKAT LUNAK *BLACK BOX TESTING* DAN PEMODELAN DIAGRAM UML PADA APLIKASI *VETERINARY SERVICES* YANG DIKEMBANGKAN DENGAN MODEL *WATERFALL*” memiliki sistem manajemen dokumen di klinik layanan kesehatan hewan yang mengandalkan penyimpanan fisik, dokumen disimpan di lemari atau ruang khusus. Seiring berjalannya waktu, jumlah dokumen bertambah, menyebabkan kebutuhan ruang semakin besar dan kesulitan dalam menemukan dokumen tertentu. Penelitian ini menerapkan metode *waterfall* untuk pengembangan *websitenya*. Kesuksesan pengujian *black box testing* terbukti karena semua fitur sistem berfungsi dengan baik, memastikan pemenuhan kebutuhan *stakeholder* melalui sistem tersebut[14].

Penelitian terakhir yang ditulis Ilham Rizky Widiyanto, Wowon Priatna, Hendarman Lubis yang berjudul “Algoritma *First in First Out (FIFO)* Untuk Perancangan Aplikasi Pemesanan Kaos Sablon” Penggunaan proses bisnis manual ini sering kali mengalami kendala, seperti ketidakteraturan dalam urutan pemesanan dan lamanya waktu tunggu pelanggan. Akibatnya, proses pemesanan kaos menjadi tidak teratur. Penelitian ini menggunakan Metode

Extreme Programing untuk merancang aplikasi. Hasil dari penelitian ini adalah pengembangan sebuah aplikasi yang menggunakan algoritma FIFO sebagai fondasi untuk menetapkan urutan antrian pemesanan yang akan diproses[15].

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul / Penulis / Tahun	Studi Kasus	Metode Pengembangan	Hasil	Perbedaan dengan penelitian yang dilakukan
1	Rancang Bangun Sistem <i>Booking Barbershop</i> Dengan Metode RAD Berbasis <i>Mobile</i> . / Karno Diantoro, Dian Gustina, Toad Maulana / 2019[9]	<i>SeeCut Hairstylizh</i>	Metode RAD	Aplikasi <i>Booking Barbershop</i> dapat mendukung <i>customer tracking</i> , Dengan demikian dapat mengatasi masalah antrean yang panjang dan memungkinkan pelanggan untuk melakukan pemesanan tanpa harus datang langsung ke <i>SeeCut Barbershop</i> secara manual.	Metode yang digunakan sama, pada penelitian penulis tidak ada <i>customer tracking</i> .
2	Aplikasi Pemesanan Jasa <i>Barbershop</i> Berbasis Android / Muhammad Syarif Mustapa, Sitti Suhadab, Moh. Ramdhan Arif Kalukuc, Alfian Zakaria / 202[10]	<i>Barbershop Kota Gorontalo</i>	Metode <i>Prototype</i>	Harapannya, aplikasi pemesanan jasa <i>barbershop</i> akan memberikan kenyamanan kepada pelanggan dalam memilih <i>barbershop</i> di Kota Gorontalo sesuai dengan preferensi mereka, sambil memastikan bahwa layanan yang ditawarkan sesuai dengan kebutuhan penampilan pelanggan.	Pada penelitian penulis, hanya berfokus pada satu <i>barbershop</i> dan menyediakan layanan reservasi <i>online</i> . Sedangkan pada penelitian ini, aplikasi dirancang untuk mencari <i>barbershop</i> yang ada di kota Gorontalo.

No	Judul / Penulis / Tahun	Studi Kasus	Metode Pengembangan	Hasil	Perbedaan dengan penelitian yang dilakukan
3	<i>Barbershop Service Booking Application for Hair Emotion Matrix Saloon</i> / Cheong Wai Feng, Mohd Zainuri / 2023[11]	<i>Hair Emotion Matrix Saloon</i>	Metode <i>Prototype</i>	Hasil dari penelitian ini adalah sebuah aplikasi Android yang memungkinkan pelanggan untuk memesan layanan <i>barbershop</i> , sambil memberikan kemampuan kepada pemilik usaha untuk mengelola karyawan, serta menyediakan fitur faktur dan data penjualan.	Pada penelitian ini tidak ada fitur untuk ulasan terhadap <i>barbershop</i> atau <i>barberman</i> . Penulis membuat aplikasi yang memiliki fitur ulasan pelanggan terhadap <i>barberman</i> yang melayaninya.
4	Penerapan Algoritma <i>Dynamic Priority Scheduling</i> Pada Aplikasi Antrian Pencucian Mobil Berbasis <i>Mobile</i> / Rizki Setyawatu, Adam Bachtiar Maulachela / 2020[12]	Cuci Mobil	Algoritma <i>Dynamic Priority Scheduling</i>	Aplikasi pencucian mobil telah berhasil menerapkan Algoritma <i>Dynamic Priority Scheduling</i> .	Kesamaan antara penelitian ini dan penelitian penulis adalah keduanya merupakan aplikasi antrian, namun perbedaannya terletak pada algoritma yang digunakan. Penelitian ini menggunakan algoritma <i>dynamic priority scheduling</i> , sedangkan

No	Judul / Penulis / Tahun	Studi Kasus	Metode Pengembangan	Hasil	Perbedaan dengan penelitian yang dilakukan
					penulis menggunakan algoritma <i>first in first out</i> .
5	Sistem Reservasi pada Mores <i>Barbershop</i> berbasis <i>Web</i> di Jatiwarna – Bekasi / Amilia Trianasari, Baren Frianto Debataraja / 2023[13]	<i>Mores Barbershop</i>	Metode <i>Action Research</i>	<i>Website</i> reservasi <i>Mores Barbershop</i> dengan tujuan dapat mempermudah pengolahan data pemesanan.	Tujuan aplikasi yang dibuat sama dengan penulis, namun perbedaan utama dengan penelitian penulis adalah bahwa aplikasi dalam penelitian ini dirancang berbasis <i>website</i> , sementara penulis merancang aplikasi berbasis <i>Android</i> .
6	Analisis Metode Pengujian Perangkat Lunak <i>Black Box Testing</i> dan Pemodelan Diagram UML pada Aplikasi <i>Veterinary Services</i> yang	Aplikasi <i>Veterinary Services</i>	Metode <i>Waterfall</i>	Pengujian <i>black box</i> testing berhasil menunjukkan keberhasilan, karena semua fitur sistem berfungsi dengan baik, memastikan kebutuhan <i>stakeholder</i> terpenuhi melalui sistem tersebut.	Dalam penelitian ini, analisis dilakukan terhadap <i>black box</i> testing yang umumnya digunakan untuk menguji perangkat lunak. Namun, penulis menerapkan <i>black box</i>

No	Judul / Penulis / Tahun	Studi Kasus	Metode Pengembangan	Hasil	Perbedaan dengan penelitian yang dilakukan
	Dikembangkan dengan Model <i>Waterfall</i> / Muhamad Syarif, Eri Bayu Pratama / 2021[14]				<i>testing</i> untuk menguji fungsionalitas aplikasi Android yang sedang dikembangkan.
7	Algoritma <i>First in First Out</i> (FIFO) Untuk Perancangan Aplikasi Pemesanan Kaos Sablon / Ilham Rizky Widiyanto, Wowon Priatna, Hendarman Lubis / 2023[15]	<i>Zelion Screen Printing</i>	Metode Pengembangan <i>Extreme Programing</i> (XP)	Tujuan dari penelitian ini adalah untuk mengatasi hambatan yang terjadi dalam proses manual pemesanan kaos. Hasilnya adalah sebuah aplikasi yang menggunakan algoritma FIFO untuk menetapkan urutan antrian pemesanan yang akan diproses. Selain itu, aplikasi ini dikembangkan dengan menerapkan pendekatan <i>Extreme Programming</i> , yang mencakup tahapan perencanaan, desain, pengkodean, dan pengujian.	Pada penelitian ini menggunakan algoritma FIFO untuk merancang aplikasi, sama seperti penulis yang menggunakan algoritma FIFO, bedanya dengan penelitian penulis ialah penulis menggunakan metode RAD sedangkan penelitian ini menggunakan metode XP.

2.2 Landasan Teori

Landasan teori melibatkan pengetahuan atau informasi yang relevan dengan penelitian yang sedang dilakukan. Tujuannya adalah membantu peneliti dalam melaksanakan studi dengan menyediakan dasar yang relevan. Dalam konteks penelitian ini, berikut adalah informasi dasar yang terkait atau berkaitan dengan penelitian.

2.2.1 *Barbershop*

Barbershop, sebagai bagian dari sektor Usaha Kecil dan Menengah (UKM), memiliki dampak yang signifikan terhadap perekonomian Indonesia dengan menciptakan lapangan kerja dan memberikan kontribusi positif bagi masyarakat melalui hasil yang dihasilkannya. Kualitas layanan yang diberikan oleh *barbershop* kepada pelanggan sangat memengaruhi keberhasilan dalam mencapai hal ini. Dalam menghadapi persaingan, upaya terus-menerus untuk meningkatkan kualitas menjadi krusial guna memenuhi preferensi konsumen. Kualitas layanan yang tinggi menjadi kunci utama dalam mencapai keberhasilan bisnis, karena mencerminkan produktivitas dan kemampuan dalam memuaskan konsumen. Suksesnya sebuah perusahaan dapat diukur dari sejauh mana perusahaan memahami dan merespons keinginan konsumen. Kenyamanan konsumen dan keinginan untuk kembali menggunakan jasa perusahaan merupakan indikator keberhasilan yang penting [16].

2.2.2 Algoritma Antrian

Algoritma antrian adalah serangkaian prosedur atau langkah yang digunakan untuk mengelola urutan pemrosesan pelanggan atau tugas dalam berbagai konteks, seperti layanan pelanggan, pemrosesan data, atau manajemen sumber daya. Adapun jenis-jenis dari antrian yaitu *first in first out* (FIFO), *last in first out* (LIFO), *priority queue*, *round robin* dan *multilevel queue*. Penelitian ini menggunakan algoritma *first in first out*, karena tidak ada prioritas dalam antrian dan pelanggan yang masuk lebih awal akan dilayani terlebih dahulu.

2.2.3 Algoritma *First In First Out*

Dalam pendekatan ini, algoritma menitikberatkan pada proses penentuan urutan antrian berdasarkan prinsip "*first come, first served*", di mana yang pertama mengantri akan dilayani terlebih dahulu, tanpa memperhatikan prioritas. FIFO (*First In First Out*) adalah konsep yang menjamin urutan dan giliran yang berkelanjutan, di mana item yang pertama masuk ke dalam antrian akan dilayani pertama kali, dan seterusnya, mengikuti alur tersebut[17].

2.2.4 Metode *Rapid Application Development*

RAD, atau *Rapid Application Development*, merupakan sebuah pendekatan dalam pengembangan sistem yang menekankan pada percepatan proses pengembangan dengan melibatkan partisipasi aktif pengguna dalam proses pembuatan[18]. Proses ini dilakukan dengan cepat, berulang, dan melibatkan serangkaian prototipe yang secara bertahap berkembang menjadi sistem akhir. Metode pengembangan RAD umumnya terdiri dari tiga tahap utama. Adapun tahapan-tahapan dari metode RAD, yaitu:



Gambar 2. 1 Metode RAD[18]

1. Perencanaan Syarat-syarat

Pada tahap awal ini, tujuan dari sistem yang akan dikembangkan dan syarat-syarat informasi yang dibutuhkan diidentifikasi. Hal ini melibatkan pengumpulan kebutuhan dari pengguna untuk memastikan bahwa sistem yang dibangun akan memenuhi kebutuhan mereka.

2. *Workshop* Desain RAD

Dalam tahap ini, dilakukan perancangan desain aplikasi, termasuk desain sistem, antarmuka, dan basis data. Setelah penulis merancang desain aplikasi, penulis akan masuk ke tahap membuat aplikasi, penulis juga melakukan evaluasi pada tahap ini jika terjadi perubahan ataupun perbaikan. Setelah aplikasi selesai dibangun, maka penulis juga melakukan testing pada aplikasi.

3. Implementasi

Tahap akhir adalah implementasi, di mana sistem yang telah dibangun diperkenalkan dan diintegrasikan ke dalam lingkungan operasional. Penulis memberi aplikasi kepada pengguna untuk siap dipakai[19].

2.2.5 *Android*

Android merupakan sistem operasi yang dirancang khusus untuk perangkat *mobile* dan berbasis Linux. Selain mencakup inti dari sistem operasi, *platform* ini juga mencakup *middleware* dan aplikasi[20]. Didesain untuk digunakan pada *smartphone* dan tablet, Android menyajikan beragam fitur dengan tujuan meningkatkan kenyamanan dalam kehidupan sehari-hari. Sejak diperkenalkan, Android terus mengalami perkembangan yang signifikan, menunjukkan tingkat kecanggihan yang terus berkembang hingga saat ini[21]. Android memberikan kesempatan bagi pengembang untuk menciptakan aplikasi mereka sendiri, mewakili generasi terbaru dari *platform mobile* dengan kemampuan pengembangan yang sesuai dengan harapan pengguna. Sistem operasi dasar Android dilisensikan di bawah GNU *General Public License* versi 2 (GPLv2), yang dikenal sebagai lisensi "*copyleft*", yang menjamin bahwa setiap perbaikan oleh pihak ketiga tetap tunduk pada syarat yang sama[20].

2.2.6 *Dart*

Dart merupakan sebuah bahasa pemrograman yang dikembangkan oleh Google dan sering digunakan dalam pengembangan aplikasi *Android* dengan menggunakan *framework* Flutter. Dengan Dart dan Flutter, pengembang dapat menciptakan aplikasi *mobile* yang berkinerja tinggi dan

menarik secara visual menggunakan satu basis kode untuk Android dan iOS. Dart mendukung kompilasi *Ahead-of-Time* (AOT) untuk performa mendekati *native* dan *Just-in-Time* (JIT) untuk hot *reload* selama pengembangan, yang mempercepat proses iterasi. Dart, sebagai bahasa pemrograman serbaguna, dapat digunakan untuk membangun *platform-platform* beragam, termasuk web, *mobile*, server, dan IoT[22]. Dart juga menjadi bahasa standar yang digunakan dalam pengembangan aplikasi menggunakan Flutter[23]. Dengan fleksibilitas, performa tinggi, dan dukungan berkelanjutan dari Google, Dart telah menjadi pilihan utama untuk pengembangan aplikasi web dan *mobile modern*, didukung oleh komunitas pengembang yang aktif dan terus berkembang.

2.2.7 Flutter

Flutter adalah sebuah *framework* antarmuka pengguna (UI) yang *open-source* yang dikembangkan oleh Google. Ia digunakan untuk membangun aplikasi yang dikompilasi secara *native* dan memberikan kinerja yang optimal untuk perangkat *mobile*, web, dan desktop, semuanya dari satu kode sumber yang sama. Dalam konteks pengembangan aplikasi Android, Flutter memungkinkan pengembang menciptakan aplikasi dengan kinerja tinggi dan tampilan yang menarik menggunakan bahasa pemrograman Dart. Sebagai *framework open-source*, Flutter mengizinkan pembuatan antarmuka pengguna (UI) yang memiliki kinerja tinggi, yang dapat disebarkan ke platform Android dan iOS dari satu basis kode tunggal[24]. Dukungan dari Google dan komunitas pengembang yang aktif menjadikan Flutter pilihan populer untuk pengembangan aplikasi modern dengan efisiensi tinggi dan pengalaman pengguna yang mulus.

2.2.8 *Structured Query Language* (SQL)

SQL merupakan fondasi dalam operasi basis data, terutama dalam melakukan pemilihan, penambahan, dan manipulasi data, yang mempermudah pengelolaan data secara otomatis. SQL, yang merupakan singkatan dari *Structured Query Language*, digunakan untuk mengelola dan melakukan manipulasi terhadap basis data relasional[25]. Dalam

pengembangan aplikasi *Android*, penggunaan SQL memungkinkan pengembang untuk melakukan operasi seperti pembuatan, pembacaan, pembaruan, dan penghapusan data dari basis data. Ini memfasilitasi penyimpanan informasi pengguna, pengelolaan konten, dan interaksi dengan sumber daya eksternal dalam aplikasi dengan mudah. Dengan menggunakan SQL, pengembang dapat membuat *query* untuk mengambil data yang dibutuhkan, memperbarui entri yang sudah ada, atau melakukan analisis data di aplikasi *Android* mereka. Keunggulan SQL adalah kemampuannya yang luas digunakan dan dipahami oleh banyak pengembang, menjadikannya alat yang berguna dalam pengelolaan data aplikasi *Android*.

2.2.9 *Unified Modelling Language (UML)*




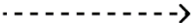
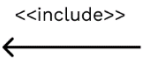

Grady Booch, James Rumbaugh, dan Ivar Jacobson bekerja bersama dalam pengembangan UML di bawah naungan *Rational Software Corps*. UML, yang merupakan kependekan dari *Unified Modeling Language*, adalah bahasa visual yang digunakan untuk menggambarkan dan mengkomunikasikan informasi tentang sistem tertentu melalui diagram dan teks pendukung. Berbagai jenis pemodelan dalam UML mencakup diagram *use case*, *activity*, *sequence*, dan *class*[26]. UML berperan utama sebagai bahasa yang digunakan untuk menentukan, menggambarkan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak, termasuk model, deskripsi, dan perangkat lunak itu sendiri. UML tidak hanya berlaku dalam pemodelan bisnis, tetapi juga dalam pemodelan sistem yang berbeda dari perangkat lunak. UML, sebagai bahasa pemodelan, mengambil konsep orientasi objek dan menyediakan notasi yang efisien untuk menggambarkan sistem dari berbagai sudut pandang. Penggunaan UML tidak terbatas hanya pada pemodelan perangkat lunak, tetapi juga mencakup berbagai bidang lain yang membutuhkan teknik pemodelan [27].

1. *Use Case Diagram*

Use case diagram menggambarkan hubungan antara aktor-aktor dan sistem. Komponen-komponen dalam *use case* diagram meliputi aktor, *use case*, asosiasi, *include*, *extend*, dan hubungan generalisasi[28]. Dengan

memanfaatkan *use case* diagram, pengembang dapat memverifikasi bahwa semua fungsi utama yang diperlukan oleh pengguna telah diidentifikasi dan dipahami secara menyeluruh sebelum memulai proses pengembangan sistem.






Tabel 2.2 Simbol *Use Case* Diagram

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

2. Activity Diagram

Activity diagram merupakan tipe diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan aliran kerja atau serangkaian aktivitas dalam suatu sistem. Diagram ini membantu menggambarkan urutan langkah atau tindakan yang terjadi dalam proses sistem, serta menunjukkan bagaimana proses berpindah dari satu aktivitas ke aktivitas lainnya.

Tabel 2.3 Simbol *Activity* Diagram

Nama	Simbol	Fungsi
Initial State		Menggambarkan awal dimulainya suatu aliran aktivitas
Final State		Menggambarkan berakhirnya suatu aliran aktivitas
Activity		Menggambarkan aktivitas yang dilakukan dalam suatu aliran aktivitas
Decision		Menggambarkan pilihan kondisi atau cabang-cabang aktivitas tertentu
Transition		Berguna untuk menghubungkan satu komponen dengan komponen lainnya.

3. *Sequence* Diagram

Sequence diagram merupakan diagram dalam *Unified Modeling Language* (UML) yang mengilustrasikan interaksi antara objek-objek dalam sistem berdasarkan urutan waktu. *Sequence* diagram menampilkan bagaimana objek berkomunikasi melalui pesan selama proses atau skenario tertentu dieksekusi, yang bermanfaat untuk memvisualisasikan alur kontrol dan kolaborasi antara komponen sistem yang berbeda. Selain itu, *sequence* diagram digunakan untuk menggambarkan perilaku aktor dalam sistem secara rinci berdasarkan waktu, dengan menampilkan objek-objek yang terlibat dan pesan yang dipertukarkan di antara objek-objek tersebut dalam konteks *use case*[29].








Tabel 2.4 Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

4. Class Diagram

Class diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang bertujuan untuk memodelkan struktur statis suatu sistem dengan menampilkan kelas-kelas, atribut, metode, dan hubungan antar kelas. Diagram ini menampilkan kelas-kelas, atribut, metode, dan hubungan antara kelas-kelas tersebut. Diagram ini penting dalam desain dan analisis sistem berbasis objek karena memberikan gambaran lengkap tentang elemen-elemen utama dalam sistem serta interaksi di antara mereka.

Tabel 2.5 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<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>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.10 Black Box Testing

Black box testing adalah metode pengujian kualitas perangkat lunak yang difokuskan pada fungsionalitasnya. Tujuannya adalah untuk mengidentifikasi fungsi yang tidak sesuai, kesalahan antarmuka, masalah pada struktur data, performa yang kurang optimal, dan masalah dalam inialisasi serta terminasi[30]. *Black box testing* fungsional merupakan metode pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem tanpa memperhatikan struktur internal atau logika implementasinya. Dalam *black box testing*, pengujian dilakukan dari sudut pandang pengguna dengan tujuan memverifikasi bahwa perangkat lunak berperilaku sesuai dengan spesifikasi yang telah ditetapkan.