

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian ini menggunakan sepuluh jurnal sebagai referensi yang diambil dari sumber-sumber jurnal dan penelitian terdahulu yang relevan. Informasi dari referensi ini akan diuraikan secara singkat dalam bentuk tabel, mencakup judul, masalah penelitian, metode/algorithm yang digunakan, hasil penelitian, kekurangan yang teridentifikasi, dan perbedaan antar penelitian. Berikut adalah rangkuman singkat terkait sepuluh jurnal tersebut.

Tabel 2.1 Penelitian terdahulu

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
1	Pengembangan Aplikasi <i>Mobile</i> Pemesanan Katering Sekolah Berbasis <i>Platform</i> Android [13]	Kekurangan nutrisi pada siswa sekolah akibat ketidakmampuan orang tua untuk menyediakan bekal makan siang, ketidakpuasan siswa terhadap menu makanan yang disediakan oleh layanan katering sekolah	Metode <i>Waterfall</i>	Berdasarkan pengujian kegunaan, hasilnya menunjukkan nilai 87,125%, yang mengindikasikan bahwa aplikasi pemesanan katering ini layak untuk digunakan dan diterapkan. Selain itu, uji fungsionalitas juga	Aplikasi ini hanya mampu menghitung kebutuhan kalori untuk setiap komponen menu dan menampilkan daftar pesanan yang sedang berlangsung.	Aplikasi ini berbasis <i>mobile</i> dengan sistem operasi Android

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
				menunjukkan bahwa sistem ini valid dan cocok untuk digunakan dalam pemesanan katering sekolah.		
2	Sistem Informasi Aplikasi Pemesanan Makanan Restoran Berbasis Web Menggunakan Metode <i>Agile Development</i> [14]	Melakukan penelitian terkait penerapan metode <i>Agile</i> dalam pengembangan aplikasi pemesanan makanan restoran berbasis <i>web</i>	Metode <i>Agile Development</i> .	Hasil penelitian mencatat tingkat kepuasan pelanggan sebesar 87%, rata-rata waktu pemesanan 3 menit, dan tingkat kesalahan sekitar 2,5%, dengan fokus pada masalah koneksi internet. Sistem ini juga mencapai skor keamanan 92 dari 100.	Tidak diberikan penyelesaian pada masalah koneksi internet yang menyebabkan sebagian besar kesalahan dalam proses pemesanan	Metode pengembangan yang digunakan adalah <i>Agile Development</i>
3	Perancangan Sistem Informasi Pemesanan <i>Catering</i> Pada Buna	Pemesanan yang masih dilakukan secara manual dengan tulisan tangan meningkatkan	Metode <i>Waterfall</i>	Sistem informasi ini dapat meningkatkan kualitas kerja dengan mengurangi kesalahan dan membuat	Aplikasi Java berbasis <i>desktop</i> tidak dirancang untuk berjalan di perangkat	Sistem informasi ini berbasis <i>desktop</i>

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
	Homemade Berbasis Java [15]	kemungkinan terjadinya kesalahan, seperti pesanan yang terduplikasi, urutan penerimaan pesanan yang tidak sesuai karena penumpukan nota saat pelanggan ramai, serta risiko kesalahan manusia seperti lupa mencatat pesanan pelanggan.		proses pemesanan catering menjadi lebih cepat, tepat, dan akurat.	bergerak, seperti ponsel pintar dan tablet.	
4	Rancang Bangun Sistem Informasi Pemesanan Makanan Berbasis Web Pada Cafe Surabiku [16]	<i>Cafe Surabiku</i> menghadapi masalah dalam pelayanan dan masih menggunakan sistem manual yang tidak efisien. Pembeli harus menunggu lama untuk memesan, dan	Metode <i>Waterfall</i>	Sistem informasi yang dapat mempermudah antara pihak pembeli dan pihak petugas <i>cafe Surabiku</i> , selain itu sistem informasi ini dapat membantu efektifitas waktu pekerja	Pada tahap pengujian aplikasi terdapat beberapa fitur yang belum diuji contohnya pada fitur mengelola data pembayaran, mengelola data laporan dan cetak laporan	Sistem informasi ini dirancang tanpa menggunakan <i>framework</i> alias <i>native</i>

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
		pelayan kesulitan mencatat pesanan dengan kertas. Solusi komputerisasi diperlukan untuk meningkatkan efisiensi.		menjadi lebih cepat dan efisien.		
5	Rancang Bangun Aplikasi Pemesanan <i>Catering</i> Berbasis Web Menggunakan Metode <i>Waterfall</i> Pada <i>Catering</i> dapoer Bu Ana [17]	<i>Catering</i> Dapoer Bu Ana menghadapi beberapa permasalahan, antara lain proses pemesanan manual yang memakan biaya dan tenaga, metode promosi terbatas yang tidak efektif, keterbatasan informasi menu baru, kurangnya sistem konfirmasi otomatis setelah pemesanan, dan	Metode <i>Waterfall</i>	Aplikasi pemesanan katering yang dapat memberikan kemudahan pada pembeli, meningkatkan pelayanan serta pengelolaan data yang lebih baik pada <i>Catering</i> Dapoer Bu Ana.	Tidak adanya proses pengujian dari aplikasi yang telah selesai dibuat	Sistem informasi ini dirancang tanpa menggunakan <i>framework</i> alias <i>native</i>

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
		keterbatasan dalam jangkauan pemasaran.				
6	Rancang Bangun Sistem Informasi Pemesanan <i>Catering</i> Berbasis <i>Website</i> Pada <i>Rahayu Catering Madiun</i> [18]	<i>Rahayu Catering</i> menghadapi beberapa permasalahan yaitu Proses pemesanan yang masih manual dan pendataan yang tradisional, keterbatasan promosi dan pengenalan produk, tertinggal dari persaingan teknologi, kesulitan monitoring pembayaran dan status pengerjaan serta kurangnya kepercayaan calon pembeli	Metode <i>Waterfall</i>	Dihasilkan sebuah <i>website</i> pemesanan <i>katering</i> untuk mengatasi permasalahan yang dialami oleh pelanggan maupun pemilik <i>Rahayu Catering Madiun</i>	Dicantumkan pembahasan mengenai pengujian yang menggunakan metode <i>black box testing</i> namun tidak dicantumkan hasil pengujian yang telah dilakukan	Pada <i>website</i> ini tidak terintegrasi dengan sistem pembayaran

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
7	Perancangan Dan Implementasi <i>Catering Ordering System</i> “Hena <i>Catering</i> ” Menggunakan <i>Rapid Application Development</i> [19]	Permasalahan pada Hena <i>Catering</i> adalah penggunaan metode manual dalam proses pemesanan dan pencatatan pesanan yang rawan kesalahan, kurangnya kejelasan informasi yang diterima oleh konsumen, serta potensi ketidakseimbangan antara barang yang dipesan dan uang yang diterima akibat pembukuan manual.	<i>Rapid Application Development (RAD)</i>	Hasil akhir berupa sistem pemesanan katering berbasis <i>website</i> yang dapat mengatasi permasalahan yang dialami Hena <i>Catering</i> , berdasarkan hasil pengujian menggunakan metode <i>black box testing</i> menunjukkan hasil <i>test report</i> untuk 20 <i>use case</i> yang telah dikembangkan. Nilai rata-rata <i>test case</i> yang ‘ <i>passed</i> ’ adalah 76%.	Terdapat beberapa fitur-fitur pada aplikasi ini yang belum berjalan dengan sempurna	Metode pengembangan yang digunakan adalah <i>Rapid Application Development (RAD)</i>
8	Sistem Pemesanan <i>Catering</i> Berbasis <i>Web</i> Pada PT.	Permasalahan pada PT. <i>Catering</i> Servis Utama yaitu pada bagian	Metode <i>Waterfall</i>	Hasil akhir berupa sistem pemesanan katering berbasis <i>web</i> yang dapat	Halaman <i>website</i> yang masih terbilang statis terutama pada tata	Pada bagian desain sistem hanya terdapat <i>use case diagram</i>

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
	<i>Catering Servis Utama</i> [20]	promosi dan informasi layanan <i>catering</i> yang masih dilakukan hanya mengandalkan brosur serta nomor telepon yang tercatat pada brosur.		mengatasi permasalahan yang dialami PT. <i>Catering Servis Utama</i>	letak dan isi kontennya	
9	Perancangan dan Pembuatan Sistem Informasi Pemesanan Katering Kedai Mbak Tim Berbasis Android [21]	Mulai dari pencatatan konvensional menyebabkan pesanan yang terlewat karena kurang detailnya pencatatan, pembayaran pesanan di tempat acara karena pelanggan kurang yakin yang menyebabkan resiko pesanan palsu, dan permintaan informasi	<i>Prototyping</i>	Hasil akhir berupa sistem pemesanan katering berbasis android yang dapat mengatasi permasalahan yang dialami Katering Kedai Mbak Tim. Berdasarkan pengujian efektivitas menghasilkan peningkatan sebesar 15,18% dan hasil pengujian <i>usability</i> menunjukkan skor penilaian pada angka 75,75	Permasalahan sistem informasi yang berbasis <i>mobile</i> adalah kompatibilitas perangkat karena tidak semua orang menggunakan perangkat dengan spesifikasi yang sama	Sistem informasi ini berbasis <i>mobile</i> dengan sistem operasi Android

No	Judul	Masalah	Metode/Algoritma	Hasil	Kekurangan Penelitian	Perbedaan
		berulang pada setiap pesanan dapat menyebabkan proses memesan lebih lama.		yang berarti sistem dapat diterima		
10	Perancangan Sistem Informasi Pemesanan Katering Berbasis <i>Web</i> Pada Warteg Mita Jaya [22]	Penggunaan prosedur manual yang mengakibatkan kurangnya efektivitas dan efisiensi dalam proses pemesanan. Kendala lain mencakup ketidakselesaian transaksi ketika calon konsumen tidak puas menyebabkan kehilangan pelanggan potensial.	Metode <i>Waterfall</i>	Hasil akhir dari penelitian ini yaitu berupa sistem informasi pemesanan katering berbasis <i>web</i> yang dapat mengatasi permasalahan yang dialami Warteg Mita Jaya.	Dicantumkan pembahasan mengenai pengujian yang menggunakan metode <i>black box testing</i> namun tidak dicantumkan hasil pengujian yang telah dilakukan	Pada sistem informasi ini belum ada fitur pembayaran yang bekerja sama dengan payment gateway

Berdasarkan Tabel penelitian terdahulu dapat disimpulkan bahwa sepuluh penelitian sebelumnya yang telah dianalisis, terdapat persamaan pada tujuan penelitian yang sejalan, yaitu merancang dan membangun sistem untuk pemesanan makanan atau *catering*. Berdasarkan sepuluh penelitian tersebut, tujuh diantaranya menjadi acuan utama dalam penelitian karena mengadopsi metode yang sama, yaitu metode *waterfall*. Terdapat variasi pada *platform* yang dipilih, termasuk perbedaan antara pengembangan berbasis *mobile* dan *desktop*, serta beberapa yang menggunakan *platform* berbasis *website*. Perbedaan juga muncul dalam penggunaan atau tidaknya *framework* dalam tahap pengembangan, dan ada juga yang menggunakan *framework* namun tanpa melibatkan tahapan pengujian.

2.2 Landasan Teori

Ada beberapa teori dasar yang relevan yang dapat diterapkan sebagai pengetahuan dalam penelitian ini. Berikut adalah beberapa landasan teori yang digunakan dalam penelitian ini.

2.2.1 Rancang Bangun

Rancang bangun merupakan proses pembuatan aplikasi atau sistem yang belum pernah ada sebelumnya pada suatu instansi atau objek tertentu. Proses ini dapat dilakukan di berbagai bidang, seperti teknologi informasi, teknik mesin, pertanian, pariwisata, kesehatan, pendidikan, bisnis, rekayasa, dan game. Tujuan utama dari rancang bangun adalah menciptakan solusi yang efektif dan efisien untuk memenuhi kebutuhan yang ada [23].

2.2.2 Aplikasi

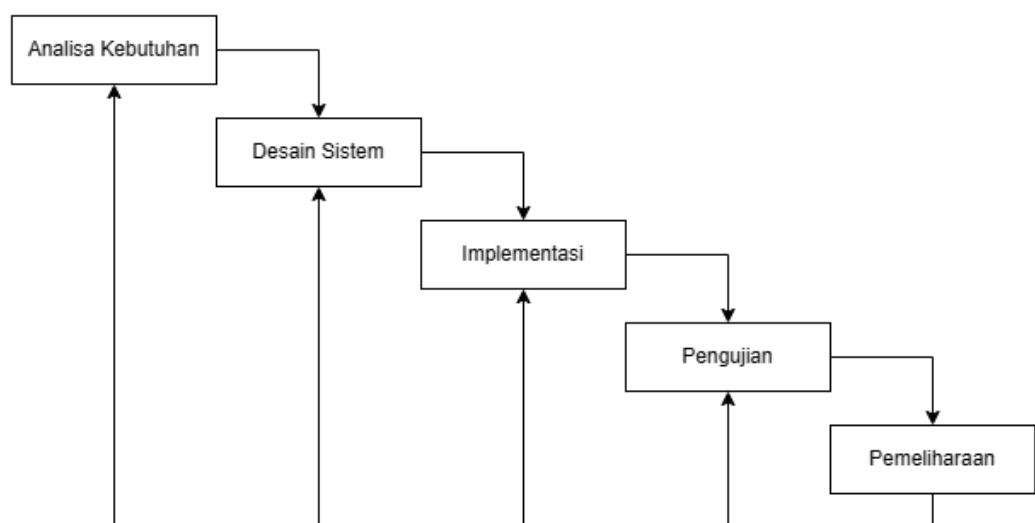
Aplikasi adalah program komputer yang dibuat untuk melakukan tugas-tugas khusus atau menyediakan layanan kepada pengguna. Aplikasi dapat dikembangkan di berbagai bidang, termasuk kesehatan, pendidikan, keuangan, dan lain-lain. Proses pengembangan aplikasi sering melibatkan penggunaan teknologi dan platform spesifik, seperti pengembangan aplikasi berbasis web, *mobile*, atau *desktop*. Aplikasi dapat dijelaskan sebagai sebuah alat terapan yang berfungsi sesuai dengan keperluannya [24].

2.2.3 Pemesanan

Menurut Kamus Besar Bahasa Indonesia, pemesanan adalah proses, pembuatan, atau cara memesan tempat, barang, jasa, dan sebagainya kepada orang lain [25]. Sistem pemesanan catering adalah suatu sistem yang digunakan untuk mengatur dan memfasilitasi proses pemesanan layanan catering. Sistem ini dapat berbasis website, aplikasi mobile, atau sistem informasi lainnya. Tujuan dari pengembangan sistem pemesanan catering adalah untuk meningkatkan efisiensi dan kenyamanan dalam proses pemesanan, baik bagi pelanggan maupun pihak penyedia layanan catering.

2.2.4 Metode *Waterfall*

Metode *waterfall*, juga dikenal sebagai pendekatan sekuensial linear, merujuk pada suatu pendekatan pengembangan perangkat lunak yang dilakukan secara berurutan, dimulai dari analisis, desain, pengkodean, pengujian, hingga tahap pemeliharaan. Keunikan dari metode ini terletak pada sifat sekuensialnya, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Oleh karena itu, kejelasan proses membuat metode ini relatif mudah dipahami, oleh karena itu metode ini sangat mudah untuk dipahami. Berikut adalah gambar metode *waterfall* [23].



Gambar 2.1 Metode *Waterfall* [23]

a) Analisis Kebutuhan

Dalam analisis kebutuhan perangkat lunak ini, dilakukan evaluasi kebutuhan sistem secara menyeluruh, termasuk aspek fungsional dan nonfungsional. Tujuannya adalah untuk memudahkan proses pengembangan aplikasi dengan memahami dengan baik kebutuhan dasar yang harus dipenuhi oleh sistem.

b) Desain

Dalam perancangan program ini, fokus diberikan pada perancangan basis data yang mencakup perancangan basis data. Selain itu, desain sistem juga melibatkan penggunaan diagram *Unified Modelling Language* (UML) untuk memberikan gambaran yang komprehensif tentang struktur dan fungsi sistem.

c) Implementasi

Tahapan implementasi merupakan tahapan selanjutnya ketika suatu program telah diketahui analisa kebutuhan dan desain yang akan dibangun, setiap komponen sistem dibangun sesuai dengan spesifikasi pada analisa kebutuhan dan desain.

d) Pengujian

Tahapan pengujian merupakan tahapan pemeriksaan pada program yang telah dirancang dan dibangun lalu dilakukan pengujian untuk mengetahui apakah program tersebut sudah sesuai dengan yang diharapkan atau belum.

e) Pemeliharaan

Tahap pemeliharaan dilakukan setelah implementasi program berhasil. Kemungkinan adanya kesalahan dalam program dapat terjadi akibat kesalahan dalam pengujian, sehingga perlu dilakukan pembaharuan dalam penulisan kode dan perawatan untuk menjaga atau memperbaiki aplikasi hingga mencapai kondisi yang benar.

2.2.5 Website

Website merupakan suatu *platform* yang memiliki banyak manfaat dan kegunaan. Salah satu manfaatnya adalah kemampuan untuk bekerja secara *real-time* dengan respon yang cepat. Hal ini berarti *website* dapat memberikan informasi yang terus diperbarui dengan cepat, sehingga pengguna dapat selalu mendapatkan informasi terbaru [26]. Selain itu, *website* juga memberikan kemudahan dalam mencari informasi yang dibutuhkan. Pengguna dapat dengan mudah mencari

informasi yang mereka perlukan melalui *website*. *Website* juga dapat menyediakan banyak informasi yang dibutuhkan oleh pengguna [27].

2.2.6 Bahasa Pemrograman PHP

PHP merupakan bahasa pemrograman yang sering digunakan dalam pembuatan *website* yang berbasis server-side dan dapat disematkan dalam *Hyper Text Markup Language* (HTML). Ini berarti skrip PHP dapat dimasukkan ke dalam dokumen HTML [28]. Terdapat dua cara utama untuk mengembangkan aplikasi *web* menggunakan Bahasa pemrograman PHP yaitu menggunakan PHP *native* atau menggunakan PHP *framework*:

- a) PHP *Native* adalah pendekatan dalam pembuatan *website* dinamis menggunakan PHP yang bersifat prosedural. Pada Metode ini, pengembangan dimulai dari awal tanpa ketergantungan pada *framework* atau *plugin* yang sudah ada [29]. PHP *Native* merupakan instruksi yang dibuat tanpa campur tangan pihak pengembang lain dalam prosesnya. Kode program atau instruksi yang dihasilkan hanya diketahui oleh pengguna dari PHP *Native* [30].
- b) PHP *Framework* adalah sekumpulan perpustakaan atau *library* yang memiliki berbagai kemampuan dan fungsi yang bervariasi, yang dikembangkan oleh para pengembang yang tergabung dalam komunitas pengembang *framework*. Pada sebuah *framework*, terdapat fungsi-fungsi yang dirancang untuk menangani permasalahan yang muncul pada PHP *Native*, khususnya dalam mengatasi tugas-tugas yang repetitif, sehingga mempermudah pelaksanaan pekerjaan. [30].

Situs *web* yang menggunakan *framework* Laravel menunjukkan tingkat efisiensi yang lebih tinggi dalam pembuatan fungsi kode program dibandingkan dengan PHP *Native*. Hal ini disebabkan oleh ketersediaan berbagai perpustakaan atau *library* yang telah disediakan oleh Laravel untuk mengeksekusi program tersebut. Meskipun PHP *Native* memiliki keunggulan dalam hal kecepatan dan ukuran halaman, hal ini disebabkan oleh fakta bahwa konten dan ukuran proyek dalam Laravel biasanya lebih besar. Meskipun struktur folder pada PHP *Native* terbukti lebih fleksibel ketika dikerjakan secara individu, namun dalam konteks pekerjaan tim atau kelompok, *Laravel* menonjol karena memiliki tempat dan struktur folder yang seragam [30].

2.2.7 Framework Laravel

Taylor Otwell adalah pengembang utama dari *framework* Laravel. Laravel merupakan suatu kerangka kerja pengembangan *web* berbasis PHP dengan arsitektur *Model View Controller* (MVC). Didesain untuk meningkatkan kualitas perangkat lunak sambil mengurangi biaya pengembangan dan perawatan, serta meningkatkan produktivitas dengan menyajikan sintaks yang bersih dan kaya fitur, yang dapat menghemat banyak waktu selama proses implementasi. [31].

2.2.8 Database

Sistem *database* dibuat dengan tujuan mengatur, menyimpan, dan mengambil data secara efisien. *Database* merupakan kumpulan data yang terstruktur yang digunakan untuk satu atau lebih tujuan, dan data tersebut dikelola dalam format digital. Pengelolaan *database* dilakukan melalui *Database Management System* (DBMS), yang memiliki tanggung jawab untuk menyimpan konten *database*, memfasilitasi pembuatan dan pemeliharaan data, serta menyediakan layanan pencarian dan akses lainnya[32].

2.2.9 MySQL



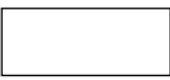


MySQL adalah sebuah perangkat lunak server *database* yang terus mengalami perkembangan, juga dikenal sebagai SQL yang merupakan singkatan dari *Structured Query Language*. SQL adalah sebuah bahasa terstruktur yang digunakan untuk mengelola *database*. Selain itu, SQL juga berperan sebagai sebuah bahasa pemrograman yang diciptakan khusus untuk mengirimkan perintah query, yang merujuk pada pengambilan data berdasarkan kriteria tertentu, kepada suatu *database*. MySQL, sebagai sistem manajemen *database*, memiliki sifat relasional, yang berarti data yang disimpan dalam basis data ditempatkan dalam beberapa tabel terpisah, memungkinkan manipulasi data secara lebih efisien. Kemampuan MySQL mencakup pengelolaan *database* dari skala kecil hingga besar [33].

2.2.10 Flowchart

Flowchart, juga dikenal sebagai diagram alur, ialah jenis diagram yang dipakai untuk mengilustrasikan algoritma atau langkah-langkah instruksi yang berjenjang dalam suatu sistem. Dengan cara ini, *flowchart* membantu dalam menemukan solusi untuk potensi permasalahan yang mungkin muncul selama

pengembangan sistem. *Flowchart* biasanya diwakili oleh simbol-simbol tertentu, di mana setiap simbol merepresentasikan suatu tahapan khusus. Garis penghubung dipergunakan untuk mengaitkan tahapan-tahapan tersebut secara berurutan [34]. Tabel 2.2 akan menjelaskan mengenai beberapa simbol-simbol *flowchart*, diantaranya:

Tabel 2.2 Simbol-simbol *flowchart*








Simbol	Nama	Deskripsi
	Connecting Line	Simbol ini menggambarkan arah proses pengolahan data
	Terminal	Simbol ini menggambarkan awal dan akhir
	Processing	Simbol ini menggambarkan suatu pengolahan atau pemrosesan data
	Decision	Simbol ini menggambarkan proses yang dilakukan berdasarkan kondisi tertentu
	Input / Output	Simbol ini menggambarkan input dan output

2.2.11 Unified Modelling Language (UML)

UML adalah standar bahasa industri yang populer untuk melakukan analisis dan desain, menguraikan persyaratan, dan menunjukkan arsitektur dalam pemrograman berorientasi objek[35]. Berikut ini merupakan diagram yang ada pada pemodelan UML:







- a) *Use Case Diagram* adalah gambaran visual yang digunakan untuk menunjukkan perilaku sistem informasi yang sedang dibangun. Diagram ini menjelaskan bagaimana pengguna berinteraksi dengan sistem melalui penjelasan naratif tentang bagaimana sistem digunakan. Tabel 2.3 menjelaskan simbol-simbol yang biasanya digunakan dalam diagram *use case*[36].

Tabel 2.3 Simbol - simbol *use case diagram*

Simbol	Nama	Keterangan
	Actor	Entitas yang berinteraksi dengan sistem
	Include	Suatu use case termasuk bagian dari use case lain
	Extend	Satu use case dapat diperluas dengan use case lain
	Association	Hubungan actor dengan use case
	System	Sistem yang sedang dikembangkan
	Use Case	Aktivitas yang dapat dilakukan actor pada system
	Collaboration	Dua atau lebih actor dan use case yang terhubung

b) *Activity Diagram* adalah grafik yang menunjukkan aktivitas atau alur kerja sistem perangkat lunak. Tabel 2.4 menjelaskan simbol-simbol yang sering digunakan dalam *activity diagram*.






Tabel 2.4 Simbol-simbol *activity diagram*

Simbol	Nama	Keterangan
	Initial	Simbol ini menggambarkan titik awal untuk memulai suatu aktivitas
	Final	Simbol ini menggambarkan titik akhir untuk mengakhiri aktivitas
	Activity	Simbol ini menggambarkan sebuah aktivitas
	Decision	Simbol ini menggambarkan pilihan untuk mengambil keputusan atau tindakan yang harus diambil pada kondisi tertentu
	Join	Simbol ini menggambarkan penggabungan dua aktivitas menjadi satu
	Swimlane	Simbol ini menggambarkan pemisahan actor yang bertanggung jawab terhadap aktivitas yang terjadi

c) *Sequence Diagram* menunjukkan perilaku objek dalam suatu *use case* dengan menjelaskan durasi hidup objek dan cara pesan dikirim dan diterima antara objek. *Sequence Diagram* dapat dibuat dengan jumlah minimal setara dengan jumlah *use case* yang memiliki prosesnya sendiri, atau dengan memastikan bahwa semua interaksi pesan dari *use case* yang telah ditentukan telah tercakup dalam *Sequence Diagram*. Semakin banyak *use case* yang didefinisikan, semakin








banyak *Sequence Diagram* yang diperlukan[36]. Tabel 2.5 merupakan simbol-simbol yang terdapat pada *sequence diagram*.

Tabel 2.5 Simbol-simbol *sequence diagram*

Simbol	Nama	Keterangan
	Actor	Simbol ini menggambarkan entitas yang berinteraksi dengan sistem
	Entity Class	Simbol ini menggambarkan hubungan kegiatan yang akan dilakukan
	Boundary Class	Simbol ini menggambarkan komunikasi antar lingkungan sistem
	Control Class	Simbol ini menggambarkan tanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Life Line	Simbol ini menggambarkan komponen yang terhubung dengan objek
	Message	Simbol ini menggambarkan komunikasi antar objek yang memuat informasi tentang aktivitas yang terjadi

d) *Class diagram* adalah gambaran struktur sistem yang menjelaskan kelas-kelas yang akan dibuat selama pembangunan sistem. *Class diagram* mencakup atribut dan operasi agar pengembang perangkat lunak dapat mengatur hubungan yang sesuai antara dokumentasi desain dan implementasi perangkat lunak [36]. Tabel 2.6 merupakan simbol-simbol yang terdapat pada *class diagram*.

Tabel 2.6 Simbol-simbol *class diagram*

Simbol	Nama	Keterangan
	Generalization	Simbol garis lurus ini menggambarkan hubungan dimana objek anak (descendant) memiliki perilaku dan struktur data dari objek yang ada di atas objek induk (ancestor).
	Nary Association	Simbol ini menggambarkan upaya untuk menghindari asosiasi dengan lebih dari 2 objek
	Class	Simbol ini menggambarkan himpunan dari objek-objek yang berbagai atribut serta operasi yang sama
	Collaboration	Simbol ini menggambarkan deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu actor
	Realization	Simbol ini menggambarkan operasi yang benar-benar hanya dilakukan oleh suatu objek
	Dependency	Simbol ini menggambarkan hubungan
	Association	Simbol ini menggambarkan apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.12 Black Box testing

Pengujian *black box* adalah cara untuk menguji perangkat lunak yang berfokus pada fungsionalitas. *Black box testing* dapat memeriksa masukan dan keluaran dari perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan yang diharapkan dalam berbagai kondisi serta untuk menemukan dan memperbaiki kesalahan [37].