

BAB II

Tinjauan Pustaka

2.1 Kajian Pustaka

Pada penelitian ini, peneliti sudah melakukan studi literatur terhadap beberapa jurnal ilmiah sebelumnya. Dilakukannya studi literatur ini adalah sebagai suatu proses dalam penelitian, yang dimana mencari jurnal-jurnal ilmiah sebelumnya itu dapat digunakan sebagai referensi sekaligus pembanding dengan menggunakan metode lain. Jurnal-jurnal ilmiah yang dipilih, tentunya sesuai dengan topik penelitian ini, dimana merancang suatu sistem informasi.

Penelitian pertama adalah penelitian yang berjudul Sistem Informasi Pariwisata Berbasis *Website* Pada Taman Nasional Gunung Ciremai Jawa Barat. Penggunaan metode *Waterfall* dan *Black Box Testing* sebagai pengujiannya, dapat diambil kesimpulan bahwa perancangan sistem informasi ini diharapkan mampu membantu pengunjung dalam mendapat informasi dan pemerintah dapat meningkatkan pendapatan daerah [12].

Penelitian kedua adalah penelitian yang berjudul Perancangan Pembuatan *Website* Pariwisata Unggulan Daerah Di Kota Jepara. Penggunaan metode *Waterfall* dan *Kuesioner* pada pengujiannya. Tahap pengujian yang dilakukan pada penelitian ini menghasilkan skor sebesar 81% dengan masuk pada kategori “Sangat Baik”. Dengan begitu, dapat disimpulkan bahwa penelitian yang dilakukan dapat dijadikan sebagai suatu alat dalam mengenalkan dan mempromosikan wisata-wisata yang ada di Kota Jepara [13].

Penelitian ketiga adalah penelitian yang berjudul Pengembangan Sistem Informasi Terpadu Industri Pariwisata Kota Batam Menerapkan User Centered Design Berbasis *Website*. Penggunaan metode *User Centered Design* dan *Evaluate Design Against User Requirements* pada pengujiannya, dapat disimpulkan bahwa penelitian ini diharapkan mampu dalam menyediakan media promosi pada pariwisata di kota Batam [14].

Penelitian keempat adalah penelitian yang berjudul Perancangan Sistem Informasi Geografis Pariwisata DKI Jakarta Berbasis *Website*. Penggunaan metode *Waterfall* dan *Black Box Testing* sebagai tahap pengujian *alpha* dan menggunakan kuesioner sebagai tahap pengujian *beta*. Penelitian ini dilakukan dua tahap pengujian, dimana pengujian *alpha* menghasilkan output yang sesuai pada fungsionalnya, tetapi masih ada beberapa kesalahan yang terjadi, dan untuk pengujian *beta* menghasilkan respon setuju dari *guest* dengan adanya sistem ini. Dilihat dari dilakukannya pengujian pada penelitian ini, akan menghasilkan perancangan aplikasi SIG yang diharapkan mampu dalam hal mengolah informasi secara efisien untuk memudahkan pengunjung dalam mendapatkan informasi dan rute sesuai tujuan [15].

Penelitian kelima adalah penelitian yang berjudul Perancangan *Website* Desa Wisata Wukirsari Bantul Sebagai Media Promosi dan Pemesanan. Penggunaan metode *Waterfall* dan *Black Box Testing* pada pengujiannya, diambil kesimpulan, *website* desa wisata Wukirsari terbukti efektif dalam dua aspek utama yaitu sebagai media promosi dan penyebaran informasi bagi pengunjung, serta sebagai alat rekapitulasi pemesanan [16].

Penelitian keenam adalah penelitian yang berjudul Perancangan Prototype Aplikasi Tiket Wisata Di Kota Pemalang Menggunakan UX Lifecycle. Penggunaan metode *User Experience Lifecycle* (UXL) dan *System Usability Score* (SUS) untuk pengujiannya menghasilkan, tingkat efisiensi 77,1667 dan tingkat efektifitasnya 96,1%, untuk semua tugas, menunjukkan performa yang baik. Pengujian terhadap Aplikasi Tiket Wisata Kota Pemalang menunjukkan performa yang memuaskan dengan skor rata-rata 77,1667. Skor ini lebih tinggi dibandingkan rata-rata skor 68 [17].

Dari semua jurnal ilmiah yang sudah dipaparkan di atas, peneliti dapat merangkum semuanya pada Tabel 2.1 berikut ini:

Tabel 2.1 Penelitian Terdahulu

| Judul | Metode | Kesamaan | Perbedaan | Hasil |
|--|------------------|---|---|---|
| S. Utarki, E. A. Pratama, and C. M. Hellyana, "Sistem Informasi Pariwisata Berbasis <i>Website</i> Pada Taman Nasional Gunung Ciremai Jawa Barat," <i>Indones. J. Softw. Eng.</i> , vol. 6, no. 1, pp. 19–32, 2020 [12]. | <i>Waterfall</i> | Perancangan sistem informasi untuk mengelola wisata dan mempermudah pengunjung dalam mendapatkan informasi. | Penelitian menggunakan metode <i>waterfall</i> . | Perancangan sistem informasi ini diharapkan mampu membantu pengunjung dalam mendapat informasi dan pemerintah dapat meningkatkan pendapatan daerah. |
| S. Hidayatullah, H. S. Mangiri, and R. I. Sudomo, "Pembuatan <i>Website</i> Pariwisata Unggulan di Kota Jepara," <i>Joined J. (Journal Informatics Educ.</i> , vol. 2, no. 2, p. 19, 2020 [13]. | <i>Waterfall</i> | Membuat sistem untuk mengenalkan objek wisata, membantu wisatawan dalam menemukan lokasi dengan fitur maps, dan membantu wisatawan dengan mudah | Penelitian menggunakan <i>waterfall</i> dan bukan membuat sistem informasi untuk pemesanan tiket. | Penelitian yang dilakukan dapat dijadikan sebagai suatu alat dalam mengenalkan dan mempromosikan wisata-wisata yang ada di Kota Jepara. |

| Judul | Metode | Kesamaan | Perbedaan | Hasil |
|---|-----------------------------|--|---|---|
| | | menghubungi pengelola. | | |
| D. Aldo, W. L. Army, W. J. Lestari, A. H. Saputra, and Z. Munir, "Pengembangan Sistem Informasi Terpadu Industri Pariwisata Kota Batam Menerapkan User Centered Design Berbasis <i>Website</i> ," <i>J. Media Inform. Budidarma</i> , vol. 6, no. 2, p. 898, 2022 [14]. | <i>User Centered Design</i> | Membuat sistem informasi untuk menyebarkan informasi ke wisatawan. | Pada sistem informasi tidak ada sistem untuk pemesana tiket. | Penelitian ini diharapkan mampu dalam menyediakan media promosi pada pariwisata di kota Batam. |
| D. R. A. Permana and G. M. Rahmah, "Perancangan Sistem Informasi Geografis Pariwisata Dki Jakarta Berbasis <i>Website</i> ," <i>J. Teknol.</i> | <i>Waterfall</i> | Merancang sistem informasi wisata. | Menggunakan metode <i>waterfall</i> dan tidak ada sistem pemesanan tiket. | Perancangan sistem diharapkan mampu dalam hal mengolah informasi secara efisien untuk memudahkan pengunjung dalam mendapatkan informasi dan rute sesuai tujuan. |

| Judul | Metode | Kesamaan | Perbedaan | Hasil |
|--|--|--|--|--|
| <i>dan Manaj.</i> , vol. 19, no. 1, pp. 7–16, 2021 [15]. | | | | |
| W. F. Hidayat, P. T. Rapiyanta, and F. Shidiq, “Perancangan <i>Website</i> Desa Wisata Wukirsari Bantul Sebagai Media Promosi dan Pemesanan,” <i>J. Infortech</i> , vol. 2, no. 1, pp. 1–7, 2020 [16]. | <i>Waterfall</i> | Perancangan sistem untuk promosi dan pemesanan wisata. | Perancangan <i>website landing page</i> untuk promosi. Metode yang digunakan adalah <i>Waterfall</i> . | <i>Website</i> Desa Wisata Wukirsari merupakan sarana yang sangat efektif untuk mempromosikan destinasi wisata yang eksotis ini kepada para pengunjung. Manfaat lain dari situs ini adalah sebagai <i>platform</i> untuk mengumpulkan data pesanan dengan cara yang lebih efisien. |
| K. Rida, G. Fitriana, and Darmansyah, “Perancangan Prototype Aplikasi Tiket Wisata Di Kota Pemandang Menggunakan UX Lifecycle,” vol. 7, no. 1, pp. 90–100, 2022 [17]. | <i>User Experience Lifecycle (UXL)</i> | Perancangan sistem tiket wisata. | Perancangan <i>prototype</i> aplikasi. | Aplikasi Tiket Wisata Di Kota Pemandang ini menunjukkan skor rata-rata 77,1667, yang jauh melebihi angka rata-rata sebesar 68. |

2.2 Dasar Teori

Peneliti akan menyajikan dan menjelaskan beberapa dasar teori yang mendukung pada penelitian ini dan berkaitan dengan judul dan pembahasan.

2.2.1 Sistem Informasi

Sistem informasi menyediakan fungsionalitas untuk menambah, mengurangi, dan menghapus data, sehingga memudahkan pengelolaan informasi. Biasanya, sistem informasi ini berbentuk perangkat lunak yang dapat berbasis *website* atau *mobile*. Oleh karena itu, sistem informasi adalah hasil kolaborasi komputer dan manusia yang melakukan pengolahan data yang dimasukkan ke dalam basis data dan pada akhirnya akan menghasilkan informasi bagi manusia sebagai penerima hasil pengolahan data tersebut [18].

2.2.2 Desa Wisata

Undang-Undang Nomor 6 Tahun 2014 tentang Desa mengartikan desa dengan suatu wilayah yang memiliki batas secara hukum yang berwenang untuk mengatur dan mengurus urusan pemerintahan, serta kepentingan masyarakat, hak asal usul, dan/atau hak tradisional harus diakui dan dihormati dalam sistem pemerintahan Negara Kesatuan Republik Indonesia [19]. Sedangkan, definisi dari wisata berdasarkan Undang-Undang Nomor 10 Tahun 2009, adalah suatu kegiatan perjalanan untuk mengunjungi tempat tertentu dengan tujuan rekreasi, pengembangan diri, atau mempelajari keunikan dari wisata yang dikunjungi oleh seseorang atau sekelompok orang dalam jangka waktu sementara [20]. Dengan demikian, dapat diambil kesimpulan bahwa desa wisata adalah sebuah bentuk penerapan dari pengembangan desa untuk meningkatkan pendapatan desa yang berfokus pada wisata.

2.2.3 Dashboard

Dashboard adalah sebuah tampilan *website* yang menampilkan grafik, tabel, laporan, bagan, dan sebagainya, yang digabungkan menjadi sebuah sistem informasi yang dinamis [21]. *Dashboard* ini umumnya digunakan

pada sistem yang dirancang untuk mengelola dan menganalisis data dan informasi. *Dashboard* yang terintegrasi dalam sistem informasi tidak memerlukan optimasi *SEO (Search Engine Optimization)* karena tidak ditujukan untuk muncul di hasil pencarian *Google*.

2.2.4 Frontend

Frontend adalah bagian dari aplikasi *web* yang bertanggung jawab untuk menampilkan informasi dan interaksi kepada pengguna. *Frontend* umumnya terdiri dari kode *HTML*, *CSS*, dan *JavaScript* yang bekerja sama untuk menghasilkan pengalaman pengguna yang menarik dan mudah digunakan [22]. *Frontend* bertanggung jawab dalam menampilkan informasi yang dinamis dari *Backend*.

2.2.5 Backend

Backend adalah sisi *server* dari suatu aplikasi *web* yang bertanggung jawab untuk menangani logika bisnis, mengelola data, dan berkomunikasi dengan *database* [23]. *Backend* umumnya ditulis dalam bahasa pemrograman seperti *Python*, *Java*, *Javascript*, atau *PHP*.

2.2.6 Javascript

JavaScript adalah bahasa pemrograman serbaguna dimana banyak dipakai dan mendukung pengembangan *web* dinamis, memungkinkan antarmuka pengguna yang interaktif dan responsif [24]. Sampai saat ini, *javascript* menjadi bahasa yang populer dengan banyak *framework* yang dibuat seperti, *ReactJS*, *Vue*, *Angular*, dan masih banyak lagi.

2.2.7 React JS

ReactJS merupakan pustaka *JavaScript* yang mempermudah pembuatan antarmuka pengguna (UI) interaktif untuk situs *web*. *React JS* ini digunakan pada sisi *Frontend* untuk membuat tampilan. *React JS* ini sering digunakan karena pada *React JS* ini, tampilan-tampilannya akan dipecah-pecah kedalam sebuah komponen dan nantinya disatukan. Hal ini membuat *React JS* ini cocok jika digunakan tim. Berdasarkan hal tersebut,

React JS ini dapat digunakan untuk membangun sistem informasi efisien, deklaratif, dan fleksibel pada sisi *Frontend* [25].

2.2.8 Express JS

Express js adalah kerangka kerja *Node JS* dan menangani JWT dan autentikasi saat pengguna masuk. *Programmer* atau *developer* dapat memilih jenis permintaan yang akan dibuat dan templat yang akan dimuat untuk respons menggunakan opsi yang disediakan [26]. Tentunya, penggunaan *Express JS* ini dapat memudahkan *programmer* dalam merancang sebuah *backend* yang berarsitektur *REST API*.

2.2.9 PostgreSQL

PostgreSQL merupakan RDBMS *open-source* yang tangguh dan fleksibel dalam mengelola basis data relasional. [27]. *PostgreSQL* memungkinkan Anda untuk menyimpan data dalam tabel yang saling terhubung, memungkinkan Anda untuk membangun hubungan yang kompleks antara data Anda. *PostgreSQL* mendukung standar *SQL* yang populer, sehingga Anda dapat menggunakan perintah *SQL* yang familier untuk mengakses dan memanipulasi data Anda.

2.2.10 REST API

REST adalah solusi yang efisien dan andal untuk mengintegrasikan aplikasi secara mudah. *REST API* memungkinkan aplikasi *web* untuk saling berkomunikasi dengan mudah dan efisien. Dengan menggunakan *REST API*, pertukaran data antar aplikasi dapat dilakukan dengan format yang terstandardisasi, sehingga memudahkan penggunaan dan pemahaman data [28]. Jadi, *REST API* ini digunakan untuk komunikasi antara *Frontend* dan *Backend*.

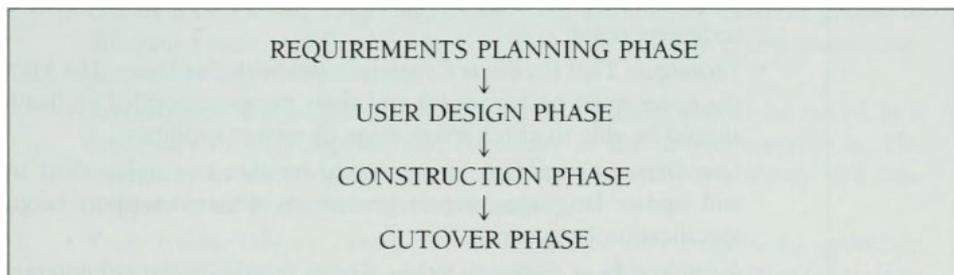
2.2.11 Postman

Postman adalah *platform* pengembangan *API* yang memungkinkan Anda untuk merancang, menguji, dan men-debug *API RESTful* [29]. *Postman* mendukung berbagai metode *HTTP* yaitu *GET*, *POST*, *PUT*, dan

DELETE, dan memungkinkan pengguna dalam menambahkan *header*, *parameter*, dan *body* ke permintaan.

2.2.12 *Rapid Application Development (RAD)*

Rapid Application Development atau yang bisa disingkat dengan RAD ini adalah metode yang lebih berfokus pada pengembangan sistem. RAD menawarkan pendekatan pengembangan perangkat lunak yang gesit dan efektif, menghasilkan produk yang lebih cepat dan berkualitas tinggi dibandingkan dengan metode tradisional. [30].



Gambar 2.1 Tahapan *Rapid Application Development (RAD)* [30]

Pada metode RAD ini, terdapat 4 tahap pengembangan sistem, yaitu: [30]

1. *Requirements Planning* (Rencana Kebutuhan)

Pada tahap ini, dilakukannya perencanaan untuk memahami lebih dalam apa yang dibutuhkan oleh pengguna. Setelah memahami kebutuhan pengguna, akan digambarkan dalam bentuk diagram alur sistem dan alur data agar pengguna dapat tau proses bisnis pada sistem. Tahap ini adalah tahapan yang penting sebagai tahap awal dalam pengembangan sistem dan dibutuhkan komunikasi yang teratur dan baik antara pengembang dan pengguna.

2. *User Design* (Desain Pengguna)

Pada tahap ini, dilakukannya pembuatan diagram-diagram untuk dapat menggambarkan apa yang ada pada sistem yang dibuat. Tahap ini tentunya adalah tahap lanjutan dari tahap sebelumnya yang diubah menjadi desain sistem sesuai alur yang sudah dibuat sebelumnya.

3. *Construction* (Pengembangan Sistem)

Setelah dibuatnya diagram-diagram dan desain sistem pada tahap sebelumnya, pada tahapan ini dilakukannya pengembangan sistem menjadi sebuah *website* yang dapat digunakan pengguna. Pada tahap ini tentunya tahapan paling utama dalam pengembangan sistem dan tahap ini akan berjalan lancar jika pada tahapan-tahapan sebelumnya dilakukan dengan benar.

4. *Cutover* (Pengujian Sistem)

Tahap *cutover* dilakukan untuk melakukan pengujian dari sistem yang sudah selesai dibuat di tahap sebelumnya. Penguji memiliki beberapa pilihan metode, seperti *Black Box Testing*, *White Box Testing*, *System Usability Scale*, , untuk mengevaluasi sistem.

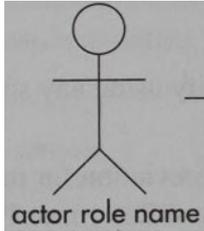
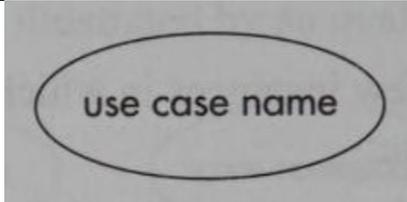
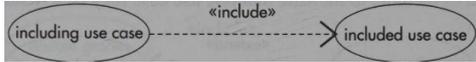
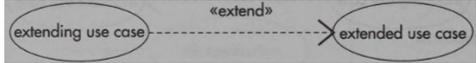
2.2.13 *Unified Modelling Language* (UML)

Unified Modelling Language merupakan sebuah alat untuk menganalogikan proses pada sebuah sistem yang berbentuk dalam berbagai macam diagram dan berorientasi objek. Diagram UML menjadi alat bantu yang efektif untuk meningkatkan kolaborasi dan pemahaman antar anggota tim pengembangan proyek, memungkinkan mereka untuk mengeksplorasi dan memvalidasi desain perangkat lunak secara bersama-sama [31].

UML terdapat beberapa macam diagram diantaranya sebagai berikut: [32]

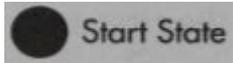
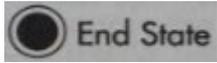
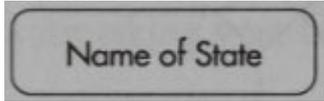
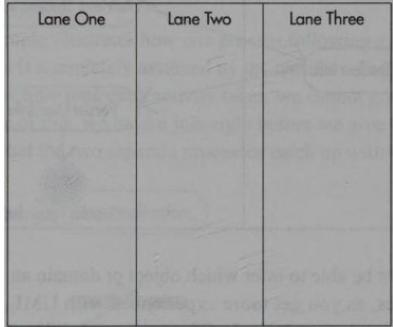
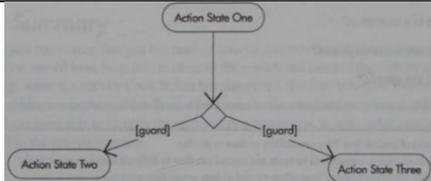
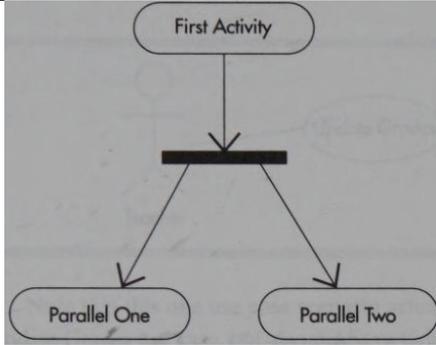
1. *Use Case Diagram* merupakan alat yang bermanfaat dalam memahami dan merancang sistem yang berpusat pada aktor dan sesuai kebutuhan. Berikut adalah Tabel 2.1 yang berisi hal-hal yang ada pada *Use Case Diagram*:

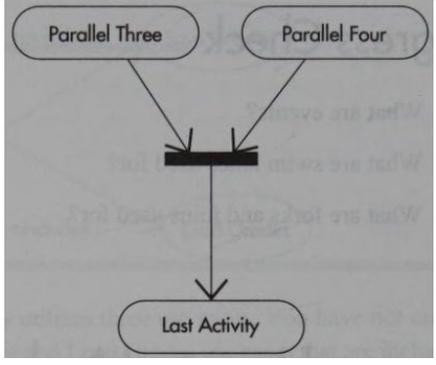
Tabel 2.2 Komponen *Use Case Diagram* [33]

| Gambar | Penjelasan |
|---|---|
|  | <p><i>Actor</i> digunakan untuk menggantikan sesuatu yang menggunakan sistem. <i>Actor</i> dapat berupa orang atau sistem. Pada <i>actor</i>, penting untuk memberikan nama pada <i>actor</i> tersebut.</p> |
|  | <p><i>Use Case</i> serangkaian langkah-langkah yang dilakukan oleh aktor untuk mencapai tujuan tertentu dalam sistem.</p> |
|  | <p><i>Include</i> digunakan saat suatu <i>case</i> memerlukan fungsionalitas dari <i>case</i> lainnya. Jadi, <i>case</i> pertama harus dilakukan terlebih dahulu agar <i>case</i> kedua dapat dilakukan.</p> |
|  | <p><i>Extend</i> mirip dengan <i>include</i> dimana digunakan untuk menghubungkan <i>case</i> satu dengan lainnya. Dimana pada <i>extend</i>, <i>case</i> yang ditunjuk oleh panah terbuka disebut perluasan. Jadi <i>case</i> kedua tidak harus dilakukan saat setelah <i>case</i> kesatu dilakukan.</p> |

2. *Activity Diagram* merupakan model visual yang menggambarkan urutan langkah-langkah dan aliran proses dalam sebuah sistem.

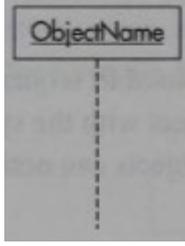
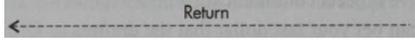
Tabel 2.3 Komponen *Activity Diagram* [33]

| Gambar | Penjelasan |
|---|--|
|  | <p><i>Start State</i> digunakan untuk mengawali sebuah aktivitas dan hanya terdapat satu awalan.</p> |
|  | <p><i>End State</i> digunakan untuk mengakhiri sebuah aktivitas sistem dan boleh terdapat banyak akhir pada sebuah aktivitas sistem.</p> |
|  | <p><i>State</i> digunakan untuk menunjukkan keadaan pada sistem saat ini.</p> |
|  | <p><i>Swimming</i> digunakan untuk mengelompokkan <i>state</i> pada <i>actor</i> tertentu yang menjalankannya.</p> |
|  | <p><i>Decision</i> digunakan ketika terdapat 2 kondisi yang berbeda.</p> |
|  | <p><i>Fork</i> digunakan ketika terdapat transisi ke beberapa aliran kontrol.</p> |

| Gambar | Penjelasan |
|---|---|
|  | <p><i>Join</i> digunakan ketika menggabungkan banyak transisi dari <i>fork</i>.</p> |

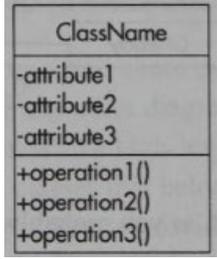
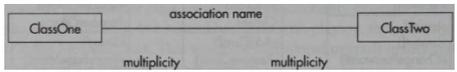
3. *Sequence Diagram* adalah model visual yang memvisualisasikan interaksi antar objek dalam sebuah *use case* dan aliran data yang terjadi di antara mereka.

Tabel 2.4 Komponen *Sequence Diagram* [33]

| Gambar | Penjelasan |
|---|--|
|  | <p><i>Object</i> digunakan untuk suatu kelas.</p> |
|  | <p><i>Synchronous</i> digunakan ketika aliran berhenti hingga pesan selesai.</p> |
|  | <p><i>Return</i> digunakan ketika aliran aktif kembali ke objek aktif pemanggil.</p> |

4. *Class Diagram* adalah permodelan untuk menggambarkan struktur sistem basis data yang berbentuk kelas-kelas dan didalamnya ada atribut-atributnya.

Tabel 2.5 Komponen *Class Diagram* [33]

| Gambar | Penjelasan |
|---|--|
|  | <p><i>Class</i> berfungsi untuk menggambarkan kelas yang ada. Dimana berisi <i>attribute</i> dan <i>operation</i> dari kelas tersebut.</p> |
|  | <p><i>Association</i> digunakan untuk menghubungkan antar <i>class</i> dan terdapat <i>multiplicity</i> untuk menggambarkan hubungannya.</p> |

2.2.14 *Black Box Testing*

Black Box Testing merupakan metode yang digunakan oleh tim pengembang *internal* untuk memverifikasi bahwa perangkat lunak mereka memenuhi semua persyaratan fungsional yang telah ditentukan, menghasilkan produk akhir yang berkualitas tinggi. Tujuan utama *Black Box Testing* adalah untuk mengidentifikasi berbagai jenis cacat dan kesalahan dalam perangkat lunak, termasuk: [34]

1. Fungsi tidak bekerja sebagaimana mestinya.
2. Ketidaksesuaian pada antarmuka.
3. Struktur data tidak terdefinisi dengan baik.
4. Performansi yang masih salah.

2.2.15 *Equivalence Partitioning*

Teknik *equivalence partition* adalah cara mengelompokkan data input menjadi beberapa kelompok atau partisi yang memiliki karakteristik yang sama [35]. Jadi, *equivalence partitioning* itu seperti membagi kelas menjadi beberapa kelompok berdasarkan nilai tertentu. Dengan menguji satu perwakilan dari setiap kelompok, bisa menghemat waktu dalam mendapat hasil yang valid untuk semua anggota kelompok tersebut.