

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian pertama yang dilakukan oleh Aprida Puji Lestari, dkk., berjudul Pembangunan Sistem Aplikasi Kolaborasi Peneliti berbasis *Website* menggunakan Metode *Rapid Application Development*, memiliki permasalahan mengenai ketersediaan tempat untuk bertukar informasi, berdebat, atau berkolaborasi dengan peneliti lain yang belum tersedia secara daring (*online*). Solusinya adalah dengan menciptakan cara komunikasi dan kolaborasi antara dosen dan mahasiswa serta peneliti berbasis *website* yang menggunakan metode *Rapid Application Development*. Sistem yang dikembangkan diuji dengan menggunakan teknik *whitebox testing* dan *blackbox testing*. Penelitian ini menghasilkan sistem aplikasi kolaborasi peneliti berbasis *website* dengan hasil pengujian 100% akurat di semua kasus pengujian [12].

Penelitian kedua oleh Muhammad Surya Jati Pradana, dkk., berjudul Implementasi Sistem Administrasi di Unit Pelaksana Teknis Pusat Pengembangan Publikasi Ilmiah Dosen Universitas Semarang Berbasis *Website*, memiliki permasalahan terkait pengelolaan surat pada Pusat Pengembangan Publikasi Ilmiah Dosen (P3ID) masih menggunakan sistem manual, sehingga berisiko terjadi kesalahan, apalagi jika jumlah surat yang dijalankan sekaligus secara langsung. Oleh karena itu, solusi dari permasalahan tersebut adalah dengan membuat sistem manajemen berbasis web yang diimplementasikan algoritma *Sequential Search* dan metode *Rapid Application Development*. Hasil dari penelitian ini, sistem pengelolaan P3ID berbasis web dapat berjalan efektif. Algoritma *Sequential Search* digunakan yang dapat menampilkan hasil pencarian berdasarkan frase kata kunci [13].

Penelitian ketiga oleh Wawan Kurniawan, dkk., berjudul RANCANG BANGUN APLIKASI RISET DAN PENGABDIAN PADA MASYARAKAT BERBASIS WEB MENGGUNAKAN METODE *RAPID APPLICATION*

DEVELOPMENT, memiliki permasalahan terkait hambatan pelaksanaan penelitian dan pengabdian kepada masyarakat (PPM). Kurang akuratnya pemetaan dosen – dosen yang akan memperoleh kesempatan melaksanakan PPM di semester berikutnya. Sistem manajemen berbasis *web* merupakan solusi yang dapat mengatasi permasalahan tersebut. Menggunakan metode RAD dalam proses pembuatan situs *web*, membuat alur kerja lebih terstruktur. Menggunakan basis data untuk sistem pencatatan kegiatan PPM, dapat menghasilkan data yang akurat dan detail [14].

Penelitian keempat oleh Erni Ermawati, dkk., dengan judul Sistem Informasi Penjualan *Furniture* Berbasis *Web*, memiliki permasalahan pada proses administrasi di perusahaan Jugala *Furniture* yang masih menggunakan proses manual, mulai dari penyediaan produk, penjualan, stok, pelaporan. Hal ini tentunya memiliki kekurangan, terutama dalam hal akurasi data. Oleh karena itu, penting untuk membuat suatu sistem informasi berbasis *website* yang dapat mengatasi permasalahan tersebut. Dengan menggunakan metode RAD, penelitian ini menciptakan sistem informasi yang dapat mendukung bisnis furnitur yang berperan sebagai media promosi, pemesanan, sekaligus pembelian *online* berbasis *website* [15].

Penelitian kelima oleh Puspa Arundini, dkk., dengan judul Rancang Bangun Sistem Informasi Desa Berbasis Web Menggunakan Metode *Rapid Application Development* (RAD) Pada Desa Sukatani, Kecamatan Cikande, Kabupaten Serang – Banten. Memiliki permasalahan berdasarkan Undang - Undang No. 6 Tahun 2014 tentang Desa yang berbunyi setiap desa wajib membangun hubungan sosial, menyediakan pelayanan dasar, mengatasi kemiskinan, meningkatkan kualitas sumber daya manusia dan meningkatkan kesejahteraan rakyat. Dalam memahami peran desa tersebut, teknologi informasi menjadi penting sebagai sarana memberikan dukungan kepada pemerintah desa serta pihak lain yang bertanggung jawab. Pengembangan sistem informasi desa berbasis *web* cocok untuk mengatasi permasalahan tersebut. Dengan menggunakan metode RAD sebagai metode pengembangan sistem, peneliti dapat menyelesaikan proses langkah demi langkah pembuatan sistem informasi desa. Penelitian ini menghasilkan suatu sistem

informasi desa yang berfungsi sebagai sarana komunikasi dan diseminasi, seperti menampilkan informasi tentang profil desa, pemerintahan, lembaga masyarakat, kependudukan, sarana dan prasarana, serta kegiatan desa [16].

Tabel 2. 1. Penelitian sebelumnya

No.	Nama Penulis	Judul	Tahun Terbit	Penerbit	Hasil	Perbandingan
1	Aprida Puji Lestari, Nurudin Santoso, Lutfi Fanani [12]	Pembangunan Sistem Aplikasi Kolaborasi Peneliti berbasis <i>Website</i> menggunakan Metode <i>Rapid Application Development</i>	2020	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer	Dengan menggunakan metode RAD menghasilkan proses pembangunan <i>website</i> yang terstruktur mulai dari perancangan hingga implementasi dan dilakukan secara berulang. Pengujian dengan menggunakan metode <i>blackbox</i> dan <i>whitebox</i> menghasilkan 100% validasi pada semua kasus uji.	Hanya menggunakan metode pengujian <i>Blackbox testing</i> , untuk menguji seluruh fitur <i>website</i> . Dikarenakan fitur yang cukup sederhana, sehingga tidak diperlukan pengujian lainnya.
2	Muhammad Surya Jati Pradana, Aria Hendrawan, B.Very Christioko, Agusta Praba Ristadi Pinem [13]	Implementasi Sistem Administrasi di Unit Pelaksana Teknis Pusat Pengembangan Publikasi Ilmiah Dosen Universitas Semarang Berbasis <i>Website</i>	2022	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)	<i>Website</i> dapat bekerja menampilkan hasil yang dicari menggunakan algoritma <i>Sequential Search</i> .	<i>Website</i> sebagai media publikasi informasi maupun kegiatan internal Balai Diklat KKB Banyumas.
3	Wawan Kurniawan, Yusriani Sapta Dewi, Riama Sibarani [14]	Rancang Bangun Aplikasi Riset dan Pengabdian Pada Masyarakat Berbasis Web Menggunakan Metode <i>Rapid Application Development</i>	2022	Metrik Serial Humaniora Dan Sains	<i>Website</i> dapat membantu Universitas Satya Negara Indonesia, dalam mengelola dan mengorganisir data penelitian, pengabdian, dan publikasi dosen.	<i>Website</i> sebagai media publikasi informasi maupun kegiatan internal Balai Diklat KKB Banyumas.

No.	Nama Penulis	Judul	Tahun Terbit	Penerbit	Hasil	Perbandingan
4	Erni Ermawati, Nurul Ichsan, Tri Wahyuni [15]	Sistem Informasi Penjualan <i>Furniture</i> Berbasis Web	2018	Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi (INTERKOM)	Penelitian ini menghasilkan sebuah <i>website</i> sebagai media promosi, pemesanan dan pembelian yang dapat digunakan sebagai salah satu bentuk pembelian <i>furniture</i> secara <i>online</i> .	<i>Website</i> sebagai media publikasi informasi maupun kegiatan internal Balai Diklat KKB Banyumas.
5	Puspa Arundini, Rudhy Ho Purabaya, Ati Zaidiah [16]	Rancang Bangun Sistem Informasi Desa Berbasis Web Menggunakan Metode Rapid Application Development (RAD) Pada Desa Sukatani, Kecamatan Cikande, Kabupaten Serang – Banten	2021	Senamika	Hasil dari penelitian ini berupa <i>website</i> sistem informasi desa yang berfungsi sebagai profil desa, pemerintahan, lembaga masyarakat, kependudukan, dan sebagai media untuk menyebarluaskan informasi tentang kegiatan Desa Sukatani Kecamatan Cikande Kabupaten Serang – Banten.	<i>Website</i> sebagai media publikasi informasi maupun kegiatan internal Balai Diklat KKB Banyumas.

2.2. Landasan Teori

2.2.1. Sistem informasi

Secara bahasa terdiri dari 2 kata yaitu sistem dan informasi. Sistem ini merupakan sekelompok orang yang bergabung dan bekerja sama dalam merealisasikan suatu tujuan dengan mengikuti ketentuan dan struktur organisasi [17].

Karakteristik sistem informasi meliputi sumber daya sistem, batasan sistem, lingkungan eksternal, antarmuka sistem, *output* sistem, struktur sistem, dan tujuan sistem. Sedangkan informasi adalah kumpulan data yang dapat diolah dan digunakan oleh penerimanya. Kemudian, konsep sistem informasi itu sendiri adalah perpaduan/kombinasi dari pengguna, perangkat keras, perangkat lunak, jaringan komunikasi dan sumber data [18].

2.2.2. Website

Website merupakan sebuah media digital berisi informasi berupa gambar, ilustrasi, teks dan video. *Website* dapat diakses oleh seluruh pengguna internet melalui domain di *World Wide Web* (WWW) dengan menggunakan browser [19].

Halaman *web* ditulis dalam format HTML (*Hyper Text Markup Language*) dan dikirim melalui HTTP (*Hypertext Transfer Protocol*), yang ditampilkan kepada pengguna melalui *browser*. Beragam jenis *browser* dapat digunakan untuk mengakses situs *web* termasuk *Internet Explorer*, *Mozilla Firefox*, *Netscape*, *Opera*, *Brave*, *Google Chrome*, dan sejenisnya [20].

2.2.3. Hypertext preprocessor

Hypertext Preprocessor atau PHP, adalah sebuah bahasa pemrograman bersifat *open source* yang biasa digunakan untuk mengembangkan *web* dinamis dan terintegrasi dengan basis data. Penggunaan bahasa pemrograman PHP sering kali bersamaan dengan penggunaan *syntax* HTML dalam proses pembuatan *website* [4].

Bahasa pemrograman PHP memiliki beberapa keunggulan [21], di antaranya:

1. Bahasa pemrograman PHP berjalan pada sisi *server* dan terhubung dengan *database*, sehingga memungkinkan untuk membangun sistem *website* secara keseluruhan.

2. Mendukung beragam jenis basis data seperti *MySQL*, *Oracle*, dan lain sebagainya.
3. Mendukung jumlah protokol yang besar seperti POP3, IMAP, LDAP.
4. *Syntax* PHP sama dengan bahasa pemrograman C.

2.2.4. *Framework*

Framework merupakan kerangka kerja dalam proses pengembangan sistem yang berfungsi untuk membantu *developer* dalam penulisan baris *syntax code*. *Framework* terdiri dari *library*, model perangkat lunak, *application programming interface (APIs)*, dan elemen – elemen lain yang saling terintegrasi. Dalam proses pengembangan *website*, *developer* menggunakan *framework* agar dapat menciptakan sistem yang lebih mudah, cepat, dan terstruktur [22].

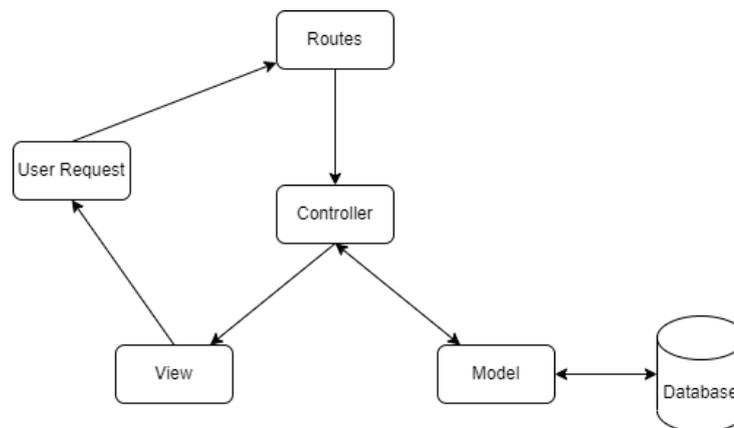
Ada beragam jenis *framework* yang sering digunakan untuk pembuatan *website*, di antaranya sebagai berikut:

1. *Framework CSS: Bootstrap, Semantic UI, Materialize, Tailwind*
2. *Framework JavaScript: AngularJS, ReactJS*
3. *Framework PHP: Laravel, CodeIgniter, Symfony*

2.2.5. *Laravel*

Laravel merupakan *framework* berbasis PHP yang dapat membantu dalam proses pembuatan *website* agar dapat di optimasi dengan menggunakan arsitektur MVC (*Model View Controller*). Selain itu, *Laravel* juga memiliki *library* yang lengkap dan siap pakai (*re-use*), sehingga segala kebutuhan *developer* terkait pengembangan sistem dapat terpenuhi [5].

Framework laravel memiliki keunggulan [23], di antaranya, *template* yang ringan, *library* yang cukup lengkap, dan menggunakan model MVC. Model MVC pada *laravel* memiliki sedikit perbedaan, terdapat *routing* yang menjadi sarana komunikasi dan *transfer data request* antara *user* dan *controller* [24].



Gambar 2. 1. Model View Controller Laravel

Terdapat 5 (lima) konsep dasar MVC pada *Laravel* [24], yaitu:

1. *Routes*, memiliki fungsi sebagai *gateway* yang memungkinkan akses ke permintaan apa pun dengan cara yang telah ditentukan.
2. *Controller*, adalah penghubung antara model dan *view*. *Controller* memiliki fungsi yang dapat menampilkan data dari model ke *view* maupun sebaliknya.
3. *Model*, merupakan sekumpulan *function* untuk mengelola suatu *table* dalam *database*.
4. *View*, merupakan *file* berisi *script code* HTML, PHP, dan sebagainya, yang memiliki fungsi untuk menampilkan data pada *browser* atau permintaan *user*.
5. *User Request*, merupakan proses *input* dari pengguna yang dikirim menuju proses *routes*.

Laravel juga memiliki karakteristik serta keunggulan, di antaranya sebagai berikut [24]:

1. Dokumentasi yang lengkap, mencakup seluruh kebutuhan *developer* dalam proses pengembangan sebuah sistem aplikasi.
2. ORM (*Object-Relational Mapping*) atau biasa disebut *Eloquent*, yaitu sebuah fitur yang dapat membantu *developer* dalam pengelolaan *database*, dengan cara menggunakan *query* yang lebih mudah dipahami.

3. *Blade Templating*, yaitu sebuah fitur dalam *laravel* yang dapat membantu developer dalam pembuatan sebuah tampilan antarmuka dengan cepat. *Blade templating* juga memudahkan dalam penggabungan *syntax* HTML dan PHP.
4. *Artisan*, yaitu sebuah *command-line* milik *laravel* yang berfungsi untuk membantu *developer* dalam pengelolaan sebuah aplikasi menggunakan perintah – perintah yang tersedia pada dokumentasi *laravel*.
5. *Middleware*, yaitu sebuah fitur yang berfungsi untuk *autentikasi*, serta validasi pada sebuah sistem dengan *framework Laravel*.
6. *Event and Listener*, yaitu sebuah fitur dalam *framework Laravel* yang berfungsi untuk mengatur serta mengelola suatu proses yang berjalan dilatar belakang sistem.

Selain dari beberapa keunggulan yang dimiliki, terdapat juga beberapa kekurangan dalam penggunaan *framework Laravel*, di antaranya sebagai berikut [24]:

1. Fitur yang banyak terkadang menyebabkan aplikasi berjalan lambat.
2. *Framework* sangat bergantung kepada bahasa pemrograman PHP, sehingga *developer* harus sudah menguasai dasar – dasar PHP sebelum menggunakan *framework Laravel*.
3. *Developer* harus mempelajari serta memahami fitur dan konsep yang tersedia di *Laravel*, sebelum dapat menggunakannya.

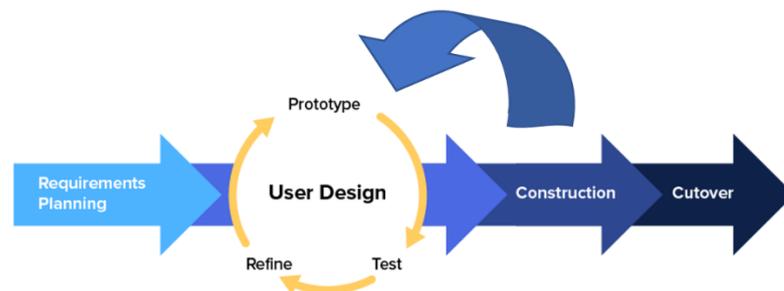
2.2.6. Database

Basis data atau *database* merupakan data atau informasi yang digabungkan dan disimpan dalam komputer secara terstruktur dan terorganisir [20]. Relasi *database* dapat dilakukan menggunakan *key* dari setiap *file* di dalamnya.

Terdapat berbagai jenis *database* yang biasa digunakan oleh *developer* dalam mengembangkan sebuah *website*. Beberapa jenis di antaranya: *Microsoft SQL Server*, *Oracle Database*, *MySQL*, *PostgreSQL*, *Cassandra*, *Titan*, *Couch DB*, *SQLite*, *MongoDB* dan lain sebagainya [20].

2.2.7. Rapid application development

Rapid Application Development adalah bagian dari model pengembangan sistem *System Development Life Cycle (SDLC)*, yang merupakan salah satu metode pengembangan perangkat lunak, dengan mempertimbangkan kinerja waktu yang singkat [6]. Proses ini terdiri dari berbagai fase, di antaranya: *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*.



Gambar 2. 2. Metode *Rapid Application Development*

Gambar di atas merupakan tahapan pengembangan sistem *Rapid Application Development*. RAD memiliki beberapa tahapan, di antaranya sebagai berikut:

1. *Requirements Planning* (Perencanaan Kebutuhan)

Perencanaan kebutuhan atau *Requirements Planning* merupakan langkah awal dalam proses pengembangan sistem dengan menggunakan metode RAD. Mengidentifikasi masalah dan mengumpulkan data dari pengguna atau pemangku kepentingan dengan tujuan untuk memperoleh gambaran proses yang akan dilakukan.

2. *User Design* (Desain Pengguna)

User Design merupakan tahapan kedua proses pengembangan sistem menggunakan metode RAD. Tahapan ini dilakukan pembuatan model dan prototipe awal agar langsung dapat didemonstrasikan kepada klien. Selanjutnya pada tahapan ini juga dilakukan perulangan yang meningkat, dengan tujuan menghasilkan luaran yang sesuai dengan keinginan pengguna atau *stakeholder* terkait. Terdiri dari 3 (tiga) tahapan dimulai dari *prototype*, *test* dan *refine*. Pada tahap *prototype*, klien dapat melihat rancangan awal yang dibuat. Kemudian lanjut tahap *test*, di mana *user*

akan memberikan *feedback* terkait kesesuaian dengan kebutuhan. Terakhir adalah *refine*, di mana pada tahap ini akan dilakukan perbaikan dan penyesuaian ulang sebelum lanjut pada iterasi proses berikutnya.

3. *Construction* (Pengembangan)

Construction merupakan tahap ketiga dalam proses pengembangan sistem dengan menggunakan metode RAD. Tahapan ini dilakukan dengan memahami proses berdasarkan hasil *user design* (tahapannya sebelumnya), dari versi beta hingga versi final. Pada saat ini, proses pembuatan sistem akan terus berlanjut hingga mendapat respons dari pengguna atau pelanggan. Jika umpan balik yang diterima tidak sesuai dengan kebutuhan pelanggan, maka akan kembali ke level sebelumnya, yaitu *user design*.

4. *Cutover / Implementation* (Penerapan)

Cutover merupakan tahap penyelesaian sebuah sistem dari segala aspek, termasuk dokumentasi, tata kelola, dan masalah lainnya untuk memastikan stabilitas sistem sebelum menyerahkannya kepada pelanggan.

Sebagai salah satu metode pengembangan sistem, RAD memiliki kelebihan dan kekurangan [7], di antaranya: dapat memudahkan proses integrasi sistem & pengurangan risiko, jangka waktu pengembangan yang lebih efisien, melibatkan pengguna atau klien dalam proses pengembangannya. Sedangkan kekurangannya yaitu, membutuhkan tim yang berkompeten dibidangnya, membutuhkan komitmen antara pengembang dengan klien, dan hanya diperuntukkan untuk proyek kecil jangka pendek.

2.2.8. *Blackbox testing*

Pengujian *blackbox* atau pengujian fungsional adalah proses pengujian perangkat lunak yang berfokus pada masukan (*input*) dan keluaran (*output*) sistem tanpa memperhatikan implementasi internalnya. Tujuannya adalah untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan, serta melakukan pengujian *end-to-end* yang mencakup seluruh proses dari awal hingga akhir [25].

Dalam pengujian *blackbox*, jumlah data uji ditentukan berdasarkan *input* yang diuji, pemenuhan aturan masukan, dan batasan yang relevan. Hal ini memastikan

bahwa berbagai kemungkinan kondisi dan skenario telah diuji secara representatif [26].

Ada beberapa teknik yang digunakan dalam pengujian *blackbox* [27], antara lain:

1. *Equivalence Partitioning*: Teknik ini membagi data masukan perangkat lunak ke dalam partisi yang setara, di mana setiap partisi memiliki karakteristik yang sama. Sebuah tes kasus diambil dari setiap partisi untuk menguji perangkat lunak. Tujuannya adalah untuk mengurangi jumlah kasus uji yang diperlukan sambil memastikan bahwa setiap partisi diuji secara representatif.
2. *Boundary Value Analysis*: Teknik ini berfokus pada pengujian pada batas-batas nilai, baik itu nilai minimum, maksimum, tepat di dalam/luar batas, nilai yang mungkin menyebabkan kesalahan, dan nilai yang umum. Tujuannya adalah untuk mengidentifikasi *bug* atau kesalahan yang sering terjadi pada batas nilai.
3. *Fuzzing*: *Fuzz testing* digunakan untuk menemukan *bug* implementasi dengan menyuntikkan data yang cacat atau setengah cacat secara otomatis atau semi-otomatis. Teknik ini menguji bagaimana perangkat lunak merespons masukan yang tidak valid atau tidak terduga.
4. *Cause-Effect Graph*: Teknik pengujian ini dimulai dengan membuat grafik yang menunjukkan hubungan antara efek dan penyebabnya. Grafik ini menggunakan simbol identitas, negasi, logika OR, dan logika AND untuk mengekspresikan ketergantungan antara penyebab dan efek. Tujuannya adalah untuk mengidentifikasi kombinasi *input* yang dapat menghasilkan efek yang diinginkan atau tidak diinginkan.
5. *Orthogonal Array Testing*: Teknik ini digunakan untuk masalah di mana domain masukan relatif kecil, tetapi terlalu besar untuk diuji secara menyeluruh. Dengan menggunakan *array orthogonal*, teknik ini memilih kombinasi uji yang paling efisien untuk mencakup berbagai kemungkinan.
6. *All Pair Testing*: Teknik pengujian ini merancang kasus uji untuk mengeksekusi semua kombinasi diskrit yang mungkin dari setiap

pasangan parameter masukan. Tujuannya adalah untuk memiliki sekumpulan kasus uji yang mencakup semua pasangan tersebut.

7. *State Transition Testing*: Jenis pengujian ini berguna untuk menguji mesin keadaan (*state machine*) dan navigasi antarmuka pengguna grafis. Pengujian ini berfokus pada perubahan keadaan dan transisi antara keadaan yang berbeda.

Penggunaan teknik-teknik di atas dalam pengujian *blackbox* membantu memastikan cakupan pengujian yang memadai dan mengidentifikasi kesalahan yang mungkin terjadi dalam perangkat lunak, tanpa memperhatikan implementasi internalnya.

2.2.9. *Unified Modelling Language*

UML (*Unified Modeling Language*) adalah salah satu prinsip pemodelan yang banyak digunakan dalam pengembangan sistem aplikasi berorientasi objek. UML membantu menggambarkan persyaratan, analisis, dan desain dari sistem aplikasi tersebut [28]. Pada tahap ini, peneliti telah menggunakan tiga jenis diagram UML yang berbeda untuk mencapai tujuan tersebut, di antaranya sebagai berikut:

1. *Use case diagram*

Use case diagram adalah salah satu bagian dari UML yang berfungsi menggambarkan fungsionalitas sebuah sistem. *Use case diagram* terdiri dari beberapa simbol khusus dengan makna berbeda – beda [28], di antaranya sebagai berikut:

Tabel 2. 2. Simbol *Usecase Diagram*

Simbol	Keterangan
 <i>Actor</i>	Mewakili manusia, sistem lain, maupun perangkat pada sebuah <i>use case diagram</i>
 <i>Use Case</i>	Abstraksi interaksi yang digunakan antara sistem dan aktor

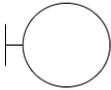
 <i>Association</i>	Abstraksi penghubung aktor dengan <i>use case</i>
 <i>Generalization</i>	Abstraksi spesialisasi aktor dengan <i>use case</i>
«include»  <i>Include</i>	Abstraksi suatu <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lainnya
«extend»  <i>Extend</i>	Abstraksi suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya, apabila memenuhi suatu kondisi

2. *Sequence diagram*

Diagram proses atau *sequence diagram* adalah metode untuk memvisualisasikan dan mendukung perilaku eksekusi tertentu [28]. Diagram proses memiliki banyak simbol unik dengan makna yang berbeda, antara lain:

Tabel 2. 3. Simbol *Sequence Diagram*

Simbol	Keterangan
 <i>Activation Box</i>	Abstraksi waktu suatu objek dalam menyelesaikan tugas tertentu
 <i>Object</i>	Abstraksi sebuah objek pada suatu sistem

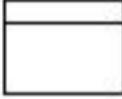
Simbol	Keterangan
 <p data-bbox="592 757 703 790"><i>Lifelines</i></p>	<p data-bbox="868 533 1260 622">Abstraksi sebuah proses yang berurutan</p>
 <p data-bbox="544 1048 751 1081"><i>Loop Condition</i></p>	<p data-bbox="868 880 1260 1025">Abstraksi untuk melaksanakan sebuah proses berdasarkan suatu keadaan tertentu</p>
 <p data-bbox="608 1279 687 1312"><i>Actor</i></p>	<p data-bbox="868 1137 1260 1283">Mewakili manusia, sistem lain, maupun perangkat pada sebuah <i>sequence diagram</i></p>
 <p data-bbox="568 1503 727 1536"><i>Entity Class</i></p>	<p data-bbox="868 1391 1260 1469">Abstraksi proses yang akan dilakukan</p>
 <p data-bbox="544 1715 751 1749"><i>Boundary Class</i></p>	<p data-bbox="868 1608 1260 1686">Abstraksi dari media komunikasi antar sistem</p>
 <p data-bbox="560 1939 735 1973"><i>Control Class</i></p>	<p data-bbox="868 1798 1260 1944">Abstraksi pengatur arus informasi pada sebuah skenario sistem</p>

Simbol	Keterangan
 <i>A focus of control & alife line</i>	Abstraksi mulai dan berakhirnya sebuah pesan
 <i>Message</i>	Abstraksi untuk mengirim pesan sebelum melanjutkan proses selanjutnya
 <i>Reply Message</i>	Abstraksi balasan untuk sebuah panggilan tertentu
 <i>Message Return</i>	Abstraksi hasil dari pengiriman sebuah pesan
 <i>Delete Message</i>	Abstraksi untuk menghapus sebuah objek

3. Activity diagram

Activity diagram atau diagram aktivitas adalah salah satu jenis diagram UML yang berfungsi untuk menunjukkan proses-proses yang bekerja dalam sistem, program, perangkat lunak dan orang atau aktor dalam sebuah sistem [28]. *Activity diagram* memiliki berbagai jenis simbol dengan kegunaannya masing - masing, di antaranya sebagai berikut:

Tabel 2. 4. Simbol *Activity Diagram*

Simbol	Keterangan
 <i>Start Status</i>	Abstraksi status awal pada diagram <i>activity</i>
 <i>Activity</i>	Abstraksi aktivitas sistem pada diagram <i>activity</i>
 <i>Decision</i>	Abstraksi percabangan pada aktivitas yang lebih dari satu
 <i>Join</i>	Abstraksi penggabungan dua atau lebih aktivitas menjadi satu
 <i>End Status</i>	Abstraksi status akhir pada sebuah sistem
 <i>Swimlane</i>	Abstraksi suatu alur bisnis dari awal hingga akhir beserta aktor terkait