

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penelitian yang membangun atau merancang sistem informasi untuk gereja telah banyak diaplikasikan dalam pengembangan aplikasi. Berikut penelitian terdahulu yang bagi penulis memiliki keterkaitan serta pula bisa memudahkan penelitian yang bakal dilakukan.

Pertama, penelitian yang berjudul “Perancangan Sistem Informasi Pengelolaan Keuangan Studi Kasus di Gereja Kanaan Suwawa” dilakukan oleh [12] pada tahun 2022. Tujuan penelitian ini adalah untuk membangun Sistem Informasi Pengelolaan Keuangan Gereja Kanaan di Suwawa agar dapat membantu bendahara gereja dalam mengorganisir dan mendetailkan data keuangan guna pelaporan yang lebih akurat, serta mempermudah manajemen data keuangan yang lebih efisien. Metode yang digunakan dalam penelitian ini adalah pengembangan *prototype*, yang melibatkan beberapa tahap seperti pengumpulan kebutuhan, pembuatan prototipe, evaluasi, pengkodean sistem, pengujian, evaluasi ulang, dan implementasi. Dengan metode ini, peneliti dapat secara iteratif mengembangkan sistem berdasarkan umpan balik dari pengguna untuk memastikan kebutuhan mereka terpenuhi. Hasil dari penelitian ini adalah Sistem Informasi Pengelolaan Keuangan untuk Gereja Kanaan di Suwawa yang dibangun menggunakan metode pengembangan prototipe, model DFD, bahasa pemrograman PHP, dan basis data MySQL. Sistem ini berbasis *website* dan bertujuan untuk membantu bendahara gereja dalam mengelola data keuangan, membuat laporan keuangan, serta mempermudah proses input dan pengelolaan data keuangan secara lebih efisien[12]. Dalam konteks perkembangan teknologi di lingkungan gereja, Gereja GKI Sul-Sel jemaat Tigaraksa menitikberatkan pada pembangunan sistem informasi gereja berbasis *website* dengan menerapkan metode *Rapid Application Development (RAD)*. Pendekatan ini

dipilih karena cocok untuk proyek dengan kebutuhan fleksibilitas, responsibilitas terhadap perubahan, dan keterlibatan pengguna yang aktif. Sementara itu, Gereja Kanaan di Suwawa lebih fokus pada pengembangan Sistem Informasi Pengelolaan Keuangan dengan menerapkan metode pengembangan *Prototype*. Pendekatan ini memungkinkan pengembangan sistem secara iteratif berdasarkan umpan balik dari pengguna, memastikan kebutuhan mereka terpenuhi seiring waktu.

Kedua, penelitian yang berjudul “Sistem Informasi Administrasi Gereja Paroki St. Stephanus Jumapolo Berbasis *Android*” dilakukan oleh [13] pada tahun 2021. Penelitian ini bertujuan untuk memberikan bantuan kepada anggota gereja dalam memperoleh informasi dengan lebih praktis. Dengan memanfaatkan sistem informasi yang berbasis *Android*, penelitian ini bertujuan untuk mempermudah akses anggota gereja terhadap informasi yang dibutuhkan. Metode pengembangan sistem yang diterapkan adalah metode *waterfall* dengan analisis *PIECES*. Hasil dari penelitian ini adalah sebuah Sistem Informasi yang mampu mengelola data gereja, pastor, kalender liturgi, tim liturgi, dewan paroki, dan pengumuman. Selain itu, sistem ini juga menyediakan informasi mengenai profil gereja, lokasi ibadah, daftar pastor, dewan paroki, tim liturgis, arsip kegiatan dalam format foto, serta pengumuman tentang kegiatan gereja[13]. Pada penelitian kedua memusatkan perhatian pada pembuatan Sistem Informasi berbasis *Android* dengan menerapkan metode pengembangan *waterfall* sedangkan peneliti akan menggunakan metode *Rapid Application Development (RAD)* yang dikenal karena fleksibilitas, responsibilitas terhadap perubahan, dan keterlibatan aktif pengguna. Sedangkan metode *waterfall* lebih bersifat linier, dengan tahapan pengembangan yang berurutan, seperti analisis, desain, implementasi, pengujian, dan pemeliharaan.

Ketiga, penelitian yang berjudul “Sistem Informasi Gereja Hki Resort Gunung Maria Berbasis Website Dengan Metode *Waterfall*” dilakukan oleh [14] pada tahun 2023. Penelitian ini bertujuan untuk mengembangkan sebuah sistem informasi gereja yang dapat membantu dalam pengelolaan data dan

informasi gereja HKI Resort Gunung Maria. Sistem ini bertujuan untuk meningkatkan efektivitas dan efisiensi dalam pengolahan data jemaat, data katekumen, data baptisan, dan data pernikahan. Metode yang digunakan dalam penelitian ini adalah metode *Waterfall*. Metode ini terdiri dari beberapa tahap, yaitu studi literatur, pengumpulan data, analisis sistem, implementasi, dan pengujian sistem. Hasil dari penelitian ini adalah pengembangan sebuah sistem informasi gereja HKI Resort Gunung Maria berbasis *website*. Sistem ini memiliki beberapa fitur, seperti halaman renungan yang memungkinkan pengguna untuk membaca dan berkomentar pada tulisan renungan, halaman agenda yang menampilkan jadwal acara gereja, dan halaman dashboard yang memberikan akses kepada admin untuk mengelola data dan informasi gereja. Sistem ini diharapkan dapat meningkatkan efektivitas dan efisiensi dalam pengolahan data gereja, serta memudahkan akses dan interaksi antara jemaat dan gereja[14]. Pada penelitian ketiga menggunakan metode *Waterfall* sedangkan peneliti menggunakan metode *Rapid Application Development* (RAD) dengan fokus pada efisiensi administratif dan kemampuan beradaptasi dengan perubahan. Sedangkan *waterfall* lebih menitikberatkan pada langkah-langkah terinci untuk meningkatkan efektivitas pengelolaan data jemaat dan informasi gereja.

Keempat, penelitian yang berjudul “Sistem Informasi Pelayanan Gereja Anugerah Injil Sepenuh Gideon Kaplingan Surakarta Menggunakan Metode *Rapid Application Development*” dilakukan oleh [10] pada tahun 2023. Penelitian ini bertujuan menciptakan sistem administrasi gereja berbasis *website* yang mendukung pelayanan bagi jemaat Gereja Anugerah Injil Sepenuh Gideon Kaplingan Surakarta. Pendekatan RAD (*Rapid Application Development*) dipilih karena memungkinkan penyelesaian proyek dalam waktu yang singkat. Hasilnya adalah pengembangan sistem informasi pelayanan gereja berbasis *website* menggunakan metode RAD. Sistem ini mampu mengelola data jemaat untuk pembuatan surat baptis, surat pernikahan, kartu anggota, serta mengatur keuangan gereja dan menyajikan laporan yang diperlukan[10]. Dalam penelitian keempat, keduanya

mengadopsi metode pengembangan yang sama, yakni *Rapid Application Development* (RAD). Namun, perlu dicatat bahwa pada penelitian ini, tidak ada pengujian yang dilakukan terhadap sistem yang dibangun. Sebaliknya, peneliti melaksanakan pengujian menggunakan metode *Blackbox Testing* dan *System Usability Scale* (SUS) pada sistem yang akan dibuat.

Kelima, penelitian yang berjudul “Perancangan Sistem Informasi Jemaat Berbasis Web Di GKI Pulomas” dilakukan oleh [15] pada tahun 2022. Penelitian ini bertujuan mengembangkan sebuah sistem informasi untuk pengelolaan data jemaat dan pengurus di Gereja Kristen Indonesia (GKI) Pulomas Jakarta, dengan tujuan mempermudah proses pengolahan data dan penyampaian informasi yang cepat, akurat, dan tepat guna meningkatkan efisiensi dan efektivitas kerja gereja. Dalam pengembangan sistem informasi ini, penelitian menggunakan pendekatan metode *waterfall*. Temuan penelitian menunjukkan bahwa pembuatan sistem berbasis *website* untuk pendataan anggota Gereja Kristen Indonesia (GKI) Pulomas Jakarta dapat meningkatkan layanan dan penyebaran informasi kepada anggota gereja. Sistem ini mencakup sejumlah fitur, seperti pendaftaran anggota, baptisan, pemberkatan nikah, sidi, peminjaman sarana ibadah, serta penyediaan dokumen piagam anggota dan informasi kegiatan gereja[15]. Dalam penelitian kelima menggunakan metode *Waterfall* sebagai metode pengembang perangkat lunak sedangkan peneliti menggunakan metode *Rapid Application Development* (RAD) dengan fokus pada efisiensi administratif dan kemampuan beradaptasi dengan perubahan. Dalam menggunakan metode *Rapid Application Development* (RAD) peneliti bertujuan untuk membangun sistem informasi gereja berbasis *website* yang dapat membantu jemaat dalam mencari informasi dan mengurus administrasi kegerejaan dan dengan melakukan pengujian menggunakan metode *black box* dan *system usability scale*. Sedangkan *waterfall* lebih menitikberatkan pada langkah-langkah terinci untuk meningkatkan efektivitas pengelolaan data jemaat dan informasi gereja dan pengujian yang dilakukan hanya berfokus secara logis dan fungsional pada perangkat lunak.

Keenam, penelitian yang berjudul “Model *Scrum* Untuk Perancangan Sistem Informasi Gereja Berbasis Mobile Pada Gereja Toraja Jemaat Taronдон” dilakukan oleh [16] Bangkalang pada tahun 2022. Penelitian tersebut bertujuan untuk mengembangkan sistem informasi gereja berbasis mobile yang dapat memenuhi kebutuhan pengguna, khususnya jemaat Gereja Toraja Jemaat Taronдон. Penelitian ini bertujuan untuk mengumpulkan data yang diperlukan sesuai dengan tujuan pengamatan, melakukan wawancara dengan pihak terkait untuk menentukan kebutuhan *user*, serta mempelajari bahan referensi karya ilmiah yang relevan dengan permasalahan yang dihadapi. Dalam Penelitian ini menggunakan metode pengembangan metode *Agile Scrum* dalam pengembangan sistem informasi gereja berbasis mobile. Tahapan metode *Agile Scrum* yang dilakukan mencakup pembuatan *Product Backlog* untuk memprioritaskan fitur-fitur yang akan dirancang, serta melakukan Sprints untuk melakukan pengembangan sistem secara bertahap. Di samping itu, teknik pengumpulan data yang diterapkan mencakup observasi, interaksi langsung, dan analisis literatur. Temuan dari penelitian ini menunjukkan bahwa pembuatan aplikasi mobile untuk keperluan gereja dapat memfasilitasi jemaat dalam menemukan informasi terkait gereja, jadwal ibadah, serta kegiatan lainnya. Aplikasi ini juga dinilai lebih efisien dan dapat mengurangi pemakaian kertas dari sistem manual sebelumnya. Selain itu, penelitian ini juga menghasilkan perancangan sistem informasi gereja dengan menggunakan teknologi *Google push notification* untuk memfasilitasi penyediaan informasi yang semakin mudah. Dengan demikian, penelitian ini memberikan kontribusi pada penyelesaian masalah yang mempengaruhi media informasi gereja, serta memberikan alternatif solusi yang lebih efisien dan efektif dalam memenuhi kebutuhan jemaat gereja[16].

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul	Perbandingan	Tujuan	Persamaan	Hasil
1	Perancangan Sistem Informasi Pengelolaan Keuangan Studi Kasus di Gereja Kanaan Suwawa (2022).	Metode pengembangan perangkat lunak yang diterapkan adalah metode prototipe dengan fokus pada pembuatan Sistem Informasi Pengelolaan Keuangan Gereja.	Penelitian ini bertujuan untuk merancang Sistem Informasi Pengelolaan Keuangan Gereja Kanaan di Suwawa, dengan tujuan membantu bendahara gereja dalam mengatur dan merinci data keuangan untuk pelaporan yang lebih akurat, serta memudahkan pengelolaan data keuangan secara lebih efisien.	Merancang sistem informasi gereja berbasis <i>website</i> .	Penelitian ini menciptakan sebuah Sistem Informasi Manajemen Keuangan untuk Gereja Kanaan di Suwawa dengan menerapkan metode <i>prototype</i> pengembangan, model DFD, bahasa pemrograman PHP, dan <i>database</i> MySQL. Sistem ini diterapkan dalam bentuk <i>website</i> dengan tujuan membantu bendahara gereja dalam mengatur data keuangan, menyusun laporan keuangan, dan menyederhanakan proses input dan manajemen data keuangan dengan lebih efektif.
2	Sistem Informasi Administrasi Gereja Paroki St. Stephanus Jumapolo Berbasis <i>Android</i> (2021).	Dengan menggunakan metode <i>Waterfall</i> dalam pengembangan perangkat lunak, sistem informasi yang dibangun berbasis mobile.	Penelitian ini bertujuan untuk memberikan kemudahan bagi warga gereja dalam mengakses informasi	Sistem dibuat untuk membantu jemaat mengurus administrasi dan mencari informasi seputar gereja.	Memudahkan warga jemaat mencari informasi terkait dengan informasi – informasi yang ada di gereja.

No	Judul	Perbandingan	Tujuan	Persamaan	Hasil
3	Sistem Informasi Gereja Hki Resort Gunung Maria Berbasis Website Dengan Metode <i>Waterfall</i> (2023).	Menerapkan metode <i>waterfall</i> sebagai metode untuk pengembangan sistem informasi.	Penelitian ini bertujuan untuk menciptakan sistem informasi gereja yang bertujuan meningkatkan efektivitas dan efisiensi pengelolaan data di gereja HKI Resort Gunung Maria, terutama terkait dengan data jemaat, katekumen, baptisan, dan pernikahan.	Merancang sistem informasi gereja berbasis <i>website</i> .	Penelitian ini menghasilkan pengembangan sistem informasi gereja berbasis <i>website</i> untuk HKI Resort Gunung Maria. Dengan fitur-fitur seperti halaman renungan, agenda, dan <i>dashboard</i> admin, sistem ini bertujuan meningkatkan efektivitas dan efisiensi pengolahan data gereja, serta memudahkan interaksi antara jemaat dan gereja.
4	Sistem Informasi Pelayanan Gereja Anugerah Injil Sepenuh Gideon Kaplingan Surakarta Menggunakan Metode <i>Rapid Application Development</i> (2023).	Menerapkan <i>usability testing</i> sebagai metode untuk pengujian aplikasi sistem informasi.	Tujuan dari penelitian ini adalah menciptakan sebuah sistem administrasi gereja yang menggunakan platform <i>website</i> , dengan tujuan mempermudah pelayanan kepada jemaat di Gereja	Menerapkan metode RAD dalam mengembangkan sistem informasi dan sistem informasi gereja yang di bangun berbasis <i>website</i> .	Penelitian ini menghasilkan pengembangan sistem informasi pelayanan gereja berbasis <i>website</i> dengan menerapkan metode RAD. Sistem yang dikembangkan dapat mengelola data jemaat untuk pembuatan surat baptisan, surat pernikahan, kartu anggota, dan administrasi keuangan gereja, serta menyediakan laporan dan ringkasan yang diperlukan.

No	Judul	Perbandingan	Tujuan	Persamaan	Hasil
			Anugerah Injil Sepenuh Gideon Kaplingan Surakarta.		
5	Perancangan Sistem Informasi Jemaat Berbasis Web Di GKI Pulomas (2022).	Metode pengembangan aplikasi yang digunakan adalah <i>Waterfall</i> .	Penelitian ini bertujuan untuk merancang sistem informasi pengelolaan data bagi Gereja Kristen Indonesia (GKI) Pulomas Jakarta. Sistem ini dirancang untuk memudahkan jemaat dan pengurus gereja dalam mengelola data serta menyediakan informasi secara cepat, tepat, dan akurat.	Merancang sistem informasi gereja berbasis <i>website</i> .	Penelitian ini menunjukkan bahwa pengembangan sistem berbasis <i>website</i> untuk pendataan anggota gereja, seperti yang diterapkan di Gereja Kristen Indonesia (GKI) Pulomas Jakarta, memiliki potensi besar dalam meningkatkan efisiensi pelayanan dan komunikasi internal. Sistem ini mencakup berbagai fitur, mulai dari pendaftaran anggota hingga layanan seperti pemberkatan nikah dan peminjaman sarana ibadah. Selain itu, memberikan kemudahan akses untuk mengunduh dokumen-dokumen penting dan mendapatkan informasi terkini seputar kegiatan gereja.
6	Model <i>Scrum</i> Untuk Perancangan Sistem Informasi Gereja Berbasis <i>Mobile</i> Pada Gereja Toraja Jemaat Tarondon (2022).	Menggunakan Metode <i>Agile Scrum</i> sebagai metode pengembangan aplikasi dan sistem informasi yang dirancang berbasis <i>mobile</i> .	Tujuan dari penelitian ini adalah untuk menciptakan sebuah sistem informasi gereja yang berbasis <i>mobile</i> yang mampu memenuhi kebutuhan	Sistem dibuat untuk membantu jemaat dalam mencari informasi seputar gereja.	dari penelitian ini menunjukkan bahwa pengembangan aplikasi berbasis <i>mobile</i> untuk gereja dapat mempermudah jemaat dalam mencari informasi seputar gereja, jadwal ibadah, dan kegiatan lainnya. Aplikasi ini juga dinilai



No	Judul	Perbandingan	Tujuan	Persamaan	Hasil
			para pengguna, khususnya anggota jemaat Gereja Toraja di Tarondon.		lebih efisien dan dapat mengurangi pemakaian kertas dari sistem manual sebelumnya. Selain itu, penelitian ini juga menghasilkan perancangan sistem informasi gereja dengan menggunakan teknologi <i>Google push notification</i> untuk memfasilitasi penyediaan informasi yang semakin mudah.

Melalui hasil pengkajian penelitian terdahulu dapat dibuat simpulan bahwasanya enam penelitian tersebut sudah dijalankan dalam pengembangan sistem informasi gereja berbasis *website* dan *mobile* diberbagai daerah Indonesia. Metode yang digunakan pengembangan perangkat lunak juga bervariasi antara satu penelitian dengan penelitian lainnya, seperti metode *prototype*, *waterfall*, *agile scrum* dan *rapid application development (RAD)*. Secara keseluruhan menunjukkan bahwa pengembangan sistem informasi gereja telah memberikan kontribusi yang signifikan dalam mempermudah akses informasi mengenai gereja. dengan demikian pembuatan sistem informasi gereja berbasis *website* dan *mobile* dianggap penting karena dapat meningkatkan akses informasi serta pengelolaan data bagi jemaat dan pihak gereja.

Dengan kemajuan teknologi dan berdasarkan penelitian sebelumnya, penelitian yang dilakukan di gereja GKI SulSel Tigaraksa juga berfokus pada sistem informasi yang menampilkan jadwal kegiatan/ibadah, jadwal petugas ibadah, dan berita terkait lingkungan gereja. Namun, berbeda dengan penelitian sebelumnya, penelitian ini tidak hanya menampilkan informasi, tetapi juga memungkinkan jemaat untuk mengisi formulir pengajuan kegiatan atau surat. Selanjutnya, sistem akan secara otomatis menghasilkan surat dari inputan formulir tersebut, dan jemaat serta pihak gereja dapat mengunduh file dalam format PDF. Dengan demikian, penelitian ini tidak hanya mempermudah akses informasi gereja, tetapi juga menyederhanakan aspek administrasi, sehingga meningkatkan aksesibilitas dalam penyampaian informasi dan administrasi.

## **2.2 Landasan Teori**

### **2.2.1 Perancangan**

Perancangan adalah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses

pengerjaannya. Proses ini mencakup berbagai langkah dan metode yang digunakan untuk merencanakan dan menentukan spesifikasi dari suatu proyek atau produk. Selain itu, perancangan juga mempertimbangkan kendala-kendala yang mungkin dihadapi, baik dari segi teknis maupun non-teknis, untuk memastikan hasil akhir yang optimal[17].

### **2.2.2 Sistem Informasi**

Sistem informasi adalah kombinasi dari berbagai sistem yang ada di dalam suatu organisasi, yang berperan dalam mengumpulkan, mengolah, menyimpan, dan menyebarkan informasi untuk membantu dalam pengambilan keputusan dan pengendalian operasional di organisasi tersebut[18]. Sistem informasi juga merupakan kumpulan data yang diakuisisi, dikategorikan, dan diproses dengan cermat hingga membentuk sebuah entitas informasi yang saling mendukung, menghasilkan informasi yang sangat berharga bagi penerimanya.[19]. Jadi bisa disimpulkan jika Sistem informasi merupakan rangkaian sistem dalam organisasi yang mengumpulkan, olah, serta memberikan informasi guna menolong pengambilan keputusan serta pengendalian dengan mengganti data sebagai informasi berharga.

### **2.2.3 Gereja**

Menurut kamus besar Bahasa Indonesia," gereja" bisa diartikan sebagai gedung tempat ibadah serta upacara agama Kristen, dan sebagai badan organisasi umat Kristen dengan kepercayaan, ajaran, serta tata cara yang sama (semacam Katolik, Protestan, serta lain- lain). Gereja merupakan tempat di mana orang berkumpul untuk beribadah, berdoa, serta melakukan upacara agama dengan kepercayaan, ajaran, serta tata cara yang sama[20].

### **2.2.4 Website**

*Website* adalah suatu sistem informasi global yang memungkinkan individu untuk mengakses dan berbagi informasi

melalui internet. Situs web merupakan komponen krusial dari internet yang memungkinkan pengguna untuk mengakses halaman web, yang merupakan dokumen elektronik yang berisi teks, gambar, video, dan *audio*[21].

Sistem *World Wide Web* (WWW) adalah sebuah kerangka kerja untuk mengakses dokumen yang tersebar di ribuan mesin di internet. Secara mendasar, web adalah suatu sistem yang menghubungkan client dan server. Dari perspektif pengguna (*client*), web adalah koleksi dokumen yang tersebar di mesin-mesin di internet yang berisi beragam jenis data, termasuk teks, gambar, animasi, dan *audio*. Di sisi lain, dari perspektif server, web menerima permintaan dari mesin *client* melalui sebuah koneksi, merespons permintaan tersebut, dan mengirimkan balasan yang ditampilkan pada sebuah *browser*[22].

### 2.2.5 HTML

HTML, singkatan dari *Hyper Text Markup Language*, adalah bahasa pemrograman dasar yang digunakan dalam pembuatan *website*. HTML terdiri dari tiga bagian utama: *Head*, *Body*, dan berisi *Tag* serta *Attribute*. Meskipun sering disebut sebagai bahasa pemrograman, sebenarnya HTML tidak dapat disebut sebagai bahasa pemrograman karena kurangnya elemen-elemen logika yang umumnya ditemukan dalam bahasa pemrograman. HTML lebih tepatnya digunakan untuk memberikan tampilan atau output, sehingga sering dianggap sebagai fondasi atau struktur dasar dari suatu situs web. Bahasa pemrograman yang sebenarnya digunakan untuk mengembangkan fungsi dan logika di situs web adalah PHP dan JavaScript[23].

### 2.2.6 CSS Bootstrap

CSS atau *Cascading Style Sheet*, yang lebih dikenal sebagai CSS, adalah sebuah fitur dari HTML yang memungkinkan Anda mengatur tampilan halaman web yang dibangun. Meskipun tidak wajib

menggunakan style sheet untuk memformat halaman web, penambahan CSS memberikan keunggulan tersendiri pada aspek tampilan halaman web. Selain itu, CSS memungkinkan pemberian efek khusus pada isi halaman web secara lebih spesifik dengan menggunakan kode yang lebih ringkas. Sebagai contoh, untuk mengubah semua teks menjadi tebal, miring, dan berwarna merah, hanya diperlukan satu baris tambahan kode. CSS dapat diterapkan pada satu atau lebih halaman HTML untuk mendefinisikan gaya untuk beberapa halaman HTML, seringkali disimpan dalam file terpisah dari HTML. Pendekatan ini mempermudah perancang web dalam melakukan perubahan gaya, karena hanya perlu mengedit satu file CSS untuk mengubah tampilan beberapa halaman HTML sekaligus[24].

Bootstrap adalah sebuah *framework front-end* yang luar biasa, dirancang khusus untuk mengoptimalkan tampilan pada perangkat *mobile* seperti *handphone* dan *smartphone*, dengan tujuan mempercepat dan menyederhanakan pengembangan website. Dengan menyediakan HTML, CSS, dan JavaScript siap pakai yang mudah dikembangkan, Bootstrap menjadi solusi ideal untuk menciptakan desain web responsif. Artinya, tampilan web yang dibangun dengan Bootstrap akan secara otomatis menyesuaikan ukuran layar, baik itu pada desktop, tablet, atau perangkat *mobile*. Keunggulan ini juga dapat dikustomisasi sesuai preferensi kita, memungkinkan kita untuk membuat tampilan web yang dioptimalkan untuk *desktop*, dan jika diakses melalui *browser mobile*, tampilan web tersebut tetap konsisten tanpa adaptasi layar yang buruk[25].

### 2.2.7 Javascript

*Javascript*, awalnya dikenal sebagai "*Livescript*," diperkenalkan oleh Netscape pada tahun 1995. Sejak saat itu, bahasa ini menjadi bahasa skrip pertama untuk web dalam sejarah internet. Meskipun awalnya dinamakan "*LiveScript*," JavaScript berkembang menjadi

bahasa yang memainkan peran kunci dalam memberikan kemampuan tambahan terhadap bahasa HTML. Dengan memungkinkan eksekusi perintah di sisi pengguna (*user*) pada *browser*, bukan di sisi server web, *JavaScript* membuka pintu untuk pengembangan web yang lebih dinamis[26].

Bahasa *JavaScript* adalah serangkaian skrip yang dieksekusi di dalam dokumen HTML. *JavaScript* bergantung pada *browser* yang membuka halaman web dengan skrip-skrip tersebut, yang kemudian diintegrasikan ke dalam dokumen HTML. Seiring dengan perkembangan internet, *JavaScript* terus menjadi bahasa *pemrograman* yang sangat penting untuk meningkatkan interaktivitas dan responsivitas dalam pengembangan aplikasi web.[26].

### 2.2.8 Laravel

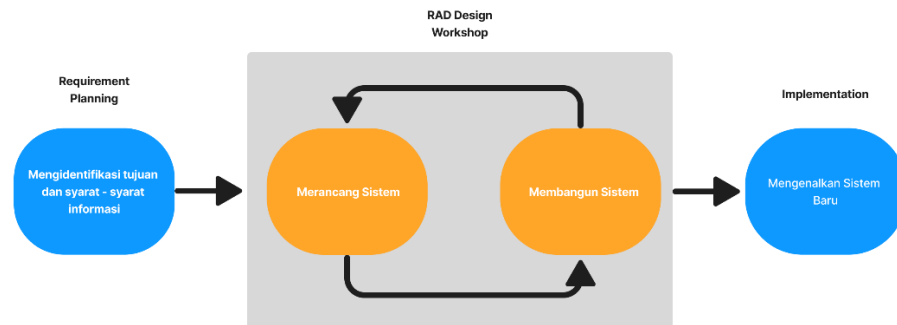
Laravel adalah sebuah *framework open-source* untuk PHP yang mengandung berbagai modul dasar untuk meningkatkan performa pengembangan aplikasi web berbasis PHP. Laravel juga menyertakan berbagai fitur bawaan yang lengkap, termasuk salah satunya adalah fitur otentikasi. *Framework* ini utamanya berfokus pada *level end-user*, dengan keunggulan dalam kesederhanaan baik dalam penulisan kode maupun tampilan aplikasi[27].

### 2.2.9 Metode *Rapid Application Development*

Dalam pengembangan sistem atau aplikasi, *Rapid Application Development (RAD)* menempatkan penekanan pada kecepatan dan kualitas tinggi dengan menggabungkan berbagai teknik terstruktur, prototyping, dan pengembangan aplikasi bersama. Hal ini memungkinkan pembuatan sistem atau aplikasi dalam waktu yang relatif singkat. Selain itu, *Rapid Application Development (RAD)* juga fokus pada pemenuhan kebutuhan bisnis yang berubah dengan cepat

dengan mempersingkat waktu antara perancangan dan implementasi sistem informasi[28].

Metode *Rapid Application Development* (RAD) terdiri dari 3 tahapan yang terstruktur dan saling bergantung disetiap tahapannya:



Gambar 2.1 Tahapan RAD

#### a. *Requiment Planning*

Pada tahap *Requiment Planning* (perencanaan kebutuhan), langkah pertama adalah mengidentifikasi kebutuhan sistem dengan memahami informasi yang diperlukan dan masalah yang dihadapi. Tujuannya adalah menetapkan tujuan, mengidentifikasi batasan sistem, mengenali kendala, dan mencari alternatif pemecahan masalah. Proses ini melibatkan analisis untuk memahami perilaku sistem dan aktivitas yang terdapat dalam sistem tersebut[28].

Dalam menentukan kebutuhan sistem dapat menggunakan *Functional Requirement* (FR). FR adalah jenis kebutuhan yang mencakup proses-proses yang akan dilakukan oleh sistem[29]. FR merinci perilaku sistem, interaksi antara pengguna dan sistem, serta alur kerja yang harus didukung. Pada tabel merupakan contoh output dari tahapan requiment planning.

Tabel 2.2 Contoh Funcional Requirement[29]

No.	Kebutuhan Fungsional (Functional Requirement)
1.	Sistem harus dapat melakukan Login untuk Admin
2.	Sistem dapat merubah dan menghapus data siswa serta menyimpan data ke database

No.	Kebutuhan Fungsional (Functional Requirement)
3.	Sistem dapat melakukan penyelesaian dengan metode AHP. Melakukan proses perhitungan dengan bobot.
4.	Sistem dapat menampilkan hasil seleksi pemilihan program studi sesuai kemampuan minat bakat siswa

b. *Design Workshop*

*Design Workshop* RAD adalah proses untuk menemukan berbagai ide solusi dan memilih yang paling baik. Langkah selanjutnya adalah membuat desain langkah-langkah kerja dan desain cara menyimpan data berdasarkan informasi yang telah dikumpulkan dan dimodelkan dalam rancangan sistem informasi. Alat yang umum digunakan dalam proses pemodelan ini adalah *unified modeling language* (UML)[28] seperti usecase diagram, activity diagram, sequence diagram dan class diagram. Selain itu untuk memberikan gambaran visual tentang tampilan dan interaksi antarmuka pengguna menggunakan *design Wireframe* dan *design High Fidelity*.

c. *Implementation*

Setelah *Design Workshop* selesai, langkah berikutnya adalah melakukan *Implementation*. Pada tahap ini, sistem diterjemahkan ke dalam bahasa mesin melalui proses pemrograman, menciptakan program atau unit program yang dapat dimengerti oleh mesin. Implementasi sistem merupakan langkah untuk mengonversi desain menjadi bentuk yang siap dioperasikan.

Metode pengembangan perangkat lunak RAD memiliki berbagai kelebihan dan kekurangan. Berikut adalah kelebihan dan kekurangan dalam pengembangan aplikasi menggunakan metode RAD:

1. Kelebihan[30]



- a) Dapat menggunakan kembali komponen yang ada (reusable object) sebelumnya sehingga tidak perlu membuat dari awal lagi.
  - b) Integrasi proses yang lebih cepat dan efektif.
  - c) Penyesuaian kebutuhan dan keinginan user menjadi lebih mudah.
  - d) Memperkecil kemungkinan kesalahan atau error.
2. Kekurangan[30]
- a) Memerlukan kolaborasi tim yang kuat dan memadai.
  - b) Memerlukan komitmen yang kuat antara pengembang dan stakeholder.
  - c) Hanya cocok diterapkan untuk proyek kecil dan memiliki waktu pengerjaan yang singkat.
  - d) Hanya cocok digunakan untuk mengembangkan aplikasi yang memiliki fokus pada suatu fitur untuk dijadikan modular terpisah.

#### **2.2.10 System Usability Scale (SUS)**

*System Usability Scale (SUS)* adalah sebuah instrumen evaluasi yang dapat diandalkan untuk mengevaluasi tingkat kemudahan penggunaan suatu produk. SUS terdiri dari sepuluh pertanyaan yang disajikan dalam format kuesioner dengan lima opsi jawaban. Pengembangan alat ini dilakukan oleh John Brooke pada tahun 1986, dan dapat diterapkan untuk mengevaluasi berbagai produk dan layanan, seperti perangkat keras, perangkat lunak, perangkat seluler, serta situs web dan aplikasi.[31]. Pertanyaan dalam System Usability Scale (SUS) dapat ditemukan dalam tabel 2.2, dan untuk penilaian masing-masing pertanyaan menggunakan skala lima poin, mulai dari sangat tidak setuju hingga sangat setuju. Pengguna dapat memilih nilai yang paling cocok dengan pendapatnya, dimulai dari 1 untuk sangat tidak setuju, 2 untuk tidak setuju, 3 untuk netral, 4 untuk setuju,

dan 5 untuk sangat setuju. Skala penilaian SUS tersedia dalam tabel 2.3.

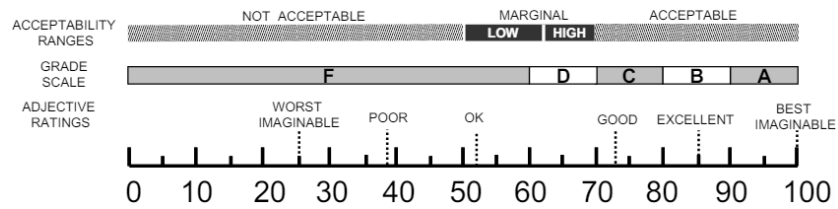
Tabel 2.3 Daftar Pertanyaan SUS[32]

No	Pertanyaan
1.	Saya berpikir akan menggunakan sistem ini lagi.
2.	Saya merasa sistem ini rumit untuk digunakan.
3.	Saya merasa sistem ini mudah untuk digunakan.
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8.	Saya merasa sistem ini membingungkan.
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Tabel 2.4 Jawaban SUS[33]

Jawaban	Skor
Sangat tidak setuju (STS)	1
Tidak setuju (TS)	2
Ragu-ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Dalam penilaian SUS, terdapat beberapa bagian penting. Pertama, ada "adjective ratings" yang menunjukkan peringkat dengan menggunakan kata sifat. Kemudian, ada "the school grading" yang memberikan nilai dengan huruf seperti A, B, dan C. Terakhir, ada "acceptability ranges" yang menunjukkan seberapa dapat diterimanya sebuah sistem aplikasi [33]. Gambar penilaian SUS dapat dilihat pada Gambar 2.2.



Gambar 2.2 Skala Penilaian SUS

### 2.2.11 *Blackbox Testing*

Metode *Blackbox Testing* adalah salah satu teknik yang simpel karena hanya memerlukan definisi nilai minimum dan maksimum dari data yang diharapkan. Estimasi jumlah data uji bisa dihitung dengan mempertimbangkan jumlah kolom data yang akan diuji, persyaratan input yang harus dipenuhi, dan situasi batas atas dan batas bawah yang relevan. Dengan menggunakan metode ini, dapat mengetahui apakah fungsionalitas masih dapat menerima input data yang tidak diinginkan dan menghasilkan data yang kurang valid saat disimpan [34]. Tabel di bawah ini merupakan contoh test case pada pengujian blackbox testing:

Tabel 2.5 Contoh Pengujian Blackbox Testing

ID	Field Uji	Skenario Pengujian	Hasil Diharapkan	Hasil Pengujian	Status
1.	Field Nama	Memasukkan karakter “Knuckle 1 Kg”	Tidak ada pesan kesalahan	Tidak ada pesan kesalahan	Diterima
		Memasukkan karakter “Beef & 4 Kg”	Karakter “&” tidak dapat dimasukkan	Karakter “&” tidak dapat dimasukkan	Diterima
		Memasukkan “Mutton Ribe Eye 1 Kg John De	Terdengar bunyi beep setelah karakter “Kg”	Terdengar bunyi beep setelah karakter “Kg”	Diterima
2.	Field Harga Beli	Memasukkan karakter “10”	Tidak ada pesan kesalahan	Tidak ada pesan kesalahan	Diterima

ID	Field Uji	Skenario Pengujian	Hasil Diharapkan	Hasil Pengujian	Status
		Memasukkan karakter "1000000"	Terdengar beep saat memasukkan angka "0" yang ke 6	Terdengar beep saat memasukkan angka "0" yang ke 6	Diterima
		Memasukkan karakter "-200"	Karakter "-" tidak dapat dimasukkan	Karakter "-" tidak dapat dimasukkan	Diterima

### 2.2.12 Wireframe

*Wireframe* merupakan pendahuluan berupa rancangan tata letak antarmuka dengan tingkat ketelitian rendah yang membantu *desainer* dalam menyampaikan konsep antarmuka dengan memberikan gambaran umum mengenai struktur dan tata letak, sehingga mempercepat proses *brainstorming*. Umumnya, *wireframe* dibuat dengan warna abu-abu dan tidak termasuk elemen gambar.[35]. Pada 2.3 gambar merupakan contoh dari penerapan wireframe.

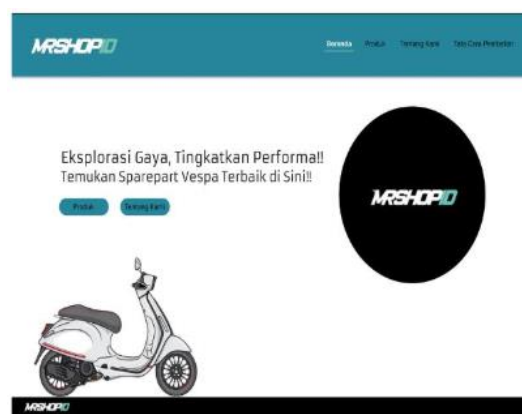


Gambar 2.3 Contoh Wireframe[36]

### 2.2.13 High Fidelity

*High-Fidelity Prototype*, sering disebut hi-fi, adalah desain akhir dari aplikasi yang akan dikembangkan[37]. Hi-fi *prototyping* merupakan versi yang lebih rinci dibandingkan dengan lo-fi *prototyping*, mencakup tambahan detail seperti warna, jenis huruf,

gambar, ikon, dan elemen lainnya[37][38]. *High fidelity* adalah jenis *wireframe* dengan tingkat detail tertinggi dibandingkan dua tipe lainnya. *High fidelity* dibuat ketika konsep telah matang dan siap untuk tahap pengembangan yang lebih kompleks, seperti sistem menu atau peta interaktif[39]. Visualisasi hi-fi ini sangat mendekati tampilan produk akhir[38]. Pada 2.4 gambar merupakan contoh dari penerapan *high fidelity*.



Gambar 2.4 Contoh high fidelity[36]

#### 2.2.14 Unified Modeling Language (UML)


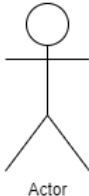

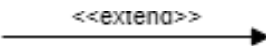

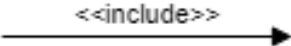
*Unified Modeling Language* (UML) merupakan bahasa pemodelan objek yang terkenal digunakan dalam pengembangan aplikasi serta web. UML menunjang menggambarkan proses pengembangan sistem dengan memakai diagram serta teks pendukung, semacam *usecase* diagram, *class* diagram, *activity* diagram, serta *sequence* diagram. Kemudahan menguasai UML menciptakan hasil pemodelannya sanggup menjadi pondasi yang kokoh dalam pengembangan sistem aplikasi maupun web[40].

##### a. Use case diagram

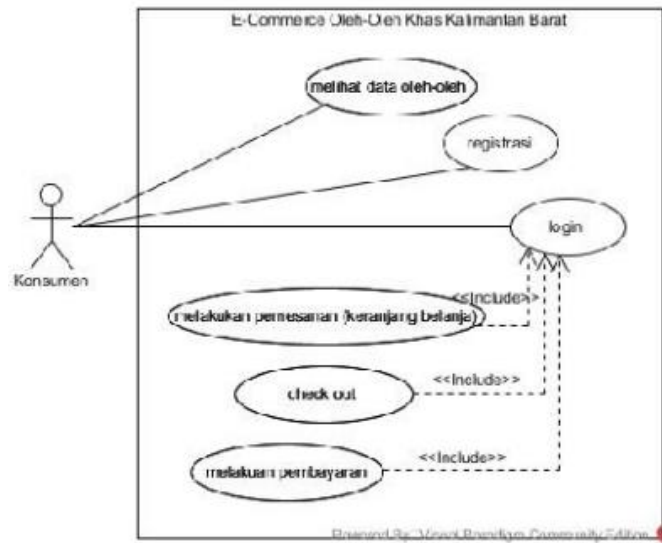
*Usecase* diagram digunakan untuk menggambarkan perilaku sistem informasi yang sedang dibangun. Ini membantu memahami tindakan yang dapat dilakukan pengguna dalam

sistem, seperti dalam aplikasi atau situs web, mempermudah pengembangan sistem informasi[40].

Tabel 2.6 Simbol Usecase Diagram[41]

Simbol	Deskripsi
Usecase 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case.
Aktor/Actor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
Asosiasi/Association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.
Ekstensi/Extend 	Relasi use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.
Generalisasi/Generalization 	Hubungan generalisasi dan spesialisasi (umumkhusus) antara dua buah use case dimana fungsi yang lebih umum dari lainnya
Include 	Ada dua sudut pandang yang cukup besar mengenai include di use case: 1. Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan

Simbol	Deskripsi
	2. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan

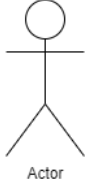


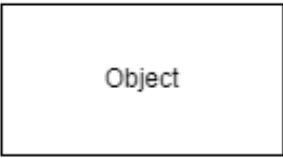
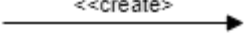
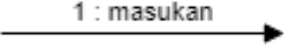
Gambar 2.5 Contoh *Usecase Diagram*[40]

#### b. *Sequence diagram*

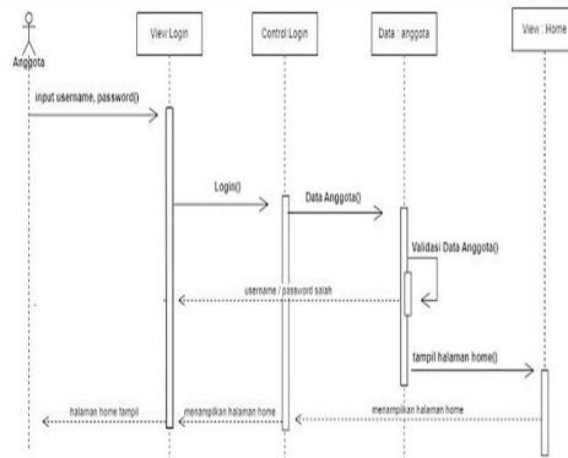
Diagram urutan atau biasa disebut *sequence diagram* adalah bagian dari UML yang menggambarkan cara objek-objek berinteraksi dalam dan sekitar sistem. Diagram ini menunjukkan interaksi antara pengguna, tampilan, dan elemen-elemen lain melalui pesan-pesan yang terjadi dalam urutan waktu tertentu.[42].

Tabel 2.7 Simbol *Sequence Diagram*[41]

Simbol	Deskripsi
Aktor	Digunakan untuk menggambarkan user/pengguna.

Simbol	Deskripsi
	
<p>Lifeline</p> 	Menyatakan kehidupan suatu objek.
<p>Activations</p> 	Penjelasan mengenai pelaksanaan fungsi yang dimiliki suatu objek.
<p>Objek</p> 	Menyampaikan pesan dengan objek yang aktif dan berinteraksi.
<p>Pesan tipe create</p> 	Mengungkapkan sebuah objek menciptakan objek lain, dengan arah panah menunjuk pada objek yang diciptakan.
<p>Pesan tipe send</p> 	Mengindikasikan bahwa suatu objek mengirimkan data ke objek lain, arah panah menunjuk ke objek yang menerima.



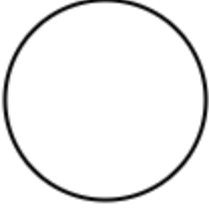

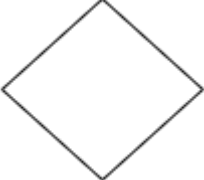
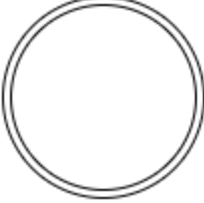
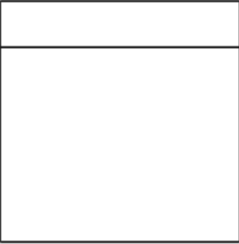

Gambar 2.6 Contoh *Sequence* Diagram[42]

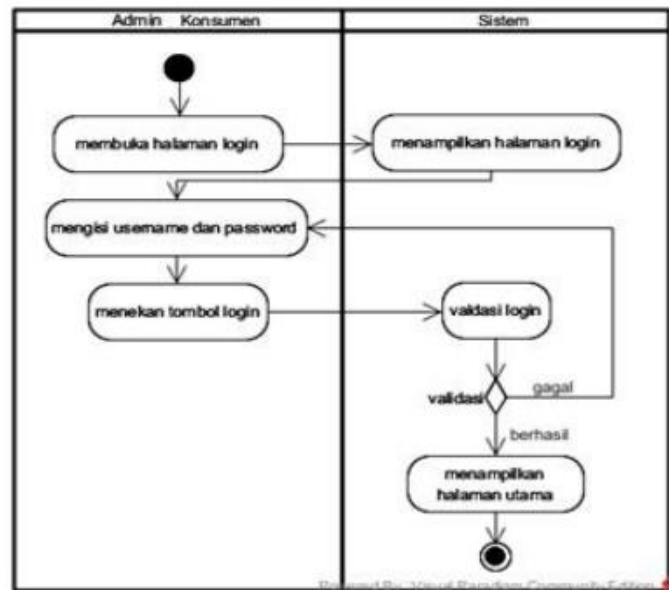
### c. Activity Diagram

*Activity* diagram merupakan alat visual yang dipakai guna mengilustrasikan urutan aksi ataupun kegiatan yang berlangsung dalam suatu sistem, proses bisnis, ataupun menu pada aplikasi fitur lunak. Diagram ini membantu menjelaskan bagaimana berbagai langkah dan tindakan terhubung satu sama lain dalam suatu proses, sehingga memudahkan pemahaman tentang cara kerja sistem tersebut. Dengan demikian, setiap langkah dapat dijelaskan bagaimana berkontribusi terhadap keseluruhan proses, serta mengidentifikasi titik-titik potensial untuk perbaikan atau efisiensi. Penggunaan diagram aktivitas mempermudah pemahaman kompleksitas suatu sistem atau proses, yang pada gilirannya memfasilitasi analisis dan pengembangan lebih lanjut [40].

Tabel 2.8 Simbol Activity Diagram[41]

Simbol	Deskripsi
Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

Simbol	Deskripsi
	
<p data-bbox="544 598 667 629">Aktivitas</p> 	<p data-bbox="919 598 1358 741">Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p data-bbox="544 799 826 831">Percabangan/decision</p> 	<p data-bbox="919 799 1369 887">Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p data-bbox="544 1068 708 1099">Status Akhir</p> 	<p data-bbox="919 1068 1342 1200">Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p data-bbox="544 1352 676 1384">Swimlane</p> 	<p data-bbox="919 1352 1358 1485">Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
<p data-bbox="544 1677 751 1709">State Transition</p> 	<p data-bbox="919 1677 1358 1765">Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan.</p>

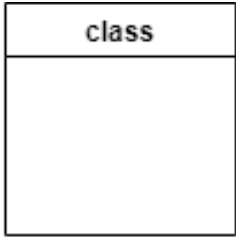



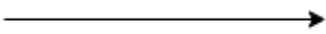
Gambar 2.7 Contoh Activity Diagram[40]

#### d. Class Diagram

Class diagram ialah pemahaman penting yang kerap digunakan dalam sistem yang berbasis objek[39]. Diagram ini menunjukkan struktur yang tetap dari kelas- kelas utama yang membentuk sistem. Diagram kelas mencerminkan atribut serta metode yang dimiliki oleh tiap kelas, dan keterkaitan yang terdapat di antara kelas- kelas tersebut [42].

Tabel 2.9 Simbol Class Diagram

Simbol	Deskripsi
Class  	Class adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir

Simbol	Deskripsi
	mendefinisikan methodmethod dari sebuah class.
Association 	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship. (Contoh: One-to-one, one-to-many, many-to-many).
Asosiasi Berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity



Gambar 2.8 Contoh Class Diagram