

BAB II KAJIAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian tentang *website* yang menyediakan informasi potensi wisata daerah merupakan penelitian yang sudah dilakukan. Pada penelitian ini penulis menggunakan beberapa referensi dari penelitian terdahulu sebagai gambaran dan acuan yang sesuai. Penulis mengangkat penelitian ini pada tempat dan kasus yang berbeda dengan penelitian terdahulu. Pada kasus ini berupa peluang tumbuhnya potensi wisata daerah yang harus di kembangkan. Peneliti sudah merangkum beberapa penelitian sebelumnya yang dapat digunakan sebagai acuan dalam penelitian sebagai berikut:

Terdapat jurnal terkait yang berjudul “Pembuatan *Website* Portal Ekonomi Dan Wisata (Ekowisata) Di Pulau Maitara Kota Tidore Kepulauan” dengan penulis Amal khairan dan Salkin Lutfi, penelitian tersebut menggunakan metode *waterfall*, dalam pembuatan *website* tersebut menghasilkan kesimpulan sebuah *website* portal yang memiliki fitur konten viewer yang berfungsi sebagai pemantauan konten *website* yang populer serta banyak diminati oleh pengunjung *website*. *Website* ini dapat diakses dengan mengunjungi alamat <http://visit-maitaraisland.com>[7].

Penelitian lainnya yang berjudul “Implementasi Metode Rapid Application Development Pada *Website* Service Guide” dengan penulis Meidyan Permata Putri dan Hendra Effendi, yang menggunakan metode *Rapid Application Development (RAD)*, Penelitian ini menghasilkan sebuah kesimpulan bahwa pengunjung merasa mudah dan terbantu dengan adanya *website* informasi wisata air terjun di sumatera selatan[8].

Penelitian lainnya pada yang berjudul “Penerapan *Website* Sebagai Sarana Promosi Wisata Budaya pada Kabupaten Lombok Timur” dengan penulis Aris Sudioanto, yang menggunakan metode teknik *survey*, Penelitian ini menghasilkan *website* yang dapat mendorong minat wisatawan dengan

memberikan sarana media promosi dan informasi yang dapat diakses oleh para wisatawan yang ingin berkunjung ke pulau lombok[9].

Penelitian lainnya yang berjudul “Pembuatan *Website* Sebagai Media Promosi Wisata Pulau Mubut” dengan penulis Swono Sibagariang, Yeni Rokhayati, Afdhol Dzikri, Siskha Handayani, Metta Santiputri, Agung Riyadi, Nur Zahra Janah dan Muhamad Sahrul Nizan yang menggunakan metode metode observasi, sosialisasi dan pelatihan, penelitian ini menghasilkan kesimpulan bahwa *website* pulau mubut dapat dikategorikan baik dan mudah dimengerti, sebanyak 81.9% pertanyaan dapat dijawab oleh responden sedangkan 18.1% tidak dapat menjawab pertanyaan yang diajukan[10].

Penelitian lainnya pada yang berjudul “Rancang Bangun *Website* Desa Citengah Untuk Pengembangan Promosi Potensi Desa” dengan penulis Reny Rian Marlina, Wagus Sejati, Wilda Ainun Nisa, Uun Pujayanti, Rizki Sopian dan Wahyu Noergana, penelitian ini menghasilkan bahwa *website* yang dibangun oleh para pemuda desa citengah kabupaten sumedang yang dapat diakses dengan mengunjungi akses link <https://citengah.sumedangdesa.id> kegiatan pembuatan *website* ini berdampak positif terhadap perkembangan informasi desa yang dapat diakses oleh banyak pihak termasuk oleh para calon wisatawan desa[11].

Penelitian lainnya pada yang berjudul “Perancangan Sistem Aplikasi Penjualan Paket Wisata Pulau Seribu Berbasis Java Dengan Metode *Scrum* (Studi Kasus PT. Abarter Global Indonesia)” dengan penulis Irfan Fakhruddin, Novita Rismawati dan Rini Sriyanti yang menghasilkan kesimpulan bahwa dengan membuat sistem aplikasi penjualan paket wisata pulau seribu yang dibangun dengan metode scrum untuk PT. Abater Global Indonesia memberikan dampak yang positif seperti kemudahan dalam memesan paket wisata, gerbang pembayaran yang sederhana, dan pelaporan yang lebih efisien dan mudah dilakukan oleh pihak-pihak yang terkait[12].

Tabel 2. 1 Perbandingan Referensi Penelitian Sebelumnya

| no | Judul Penelitian | Penulis | Metode Penelitian | Hasil |
|----|---|---|-------------------------------------|---|
| 1 | Pembuatan <i>Website</i> Portal Ekonomi Dan Wisata (Ekowisata) Di Pulau Maitara Kota Tidore Kepulauan | Amal khairan, Salkin Lutfi | Metode Waterfall | pembuatan <i>website</i> tersebut menghasilkan kesimpulan sebuah <i>website</i> portal yang memiliki fitur konten viewer yang berfungsi sebagai pemantauan konten <i>website</i> yang populer serta banyak diminati oleh pengunjung <i>website</i> . <i>Website</i> ini dapat diakses dengan mengunjungi alamat http://visit-maitaraisland.com [3] |
| 2 | Implementasi Metode Rapid Application Development Pada <i>Website</i> Service Guide “Waterfall Tour South Sumatera” | Meidyan Permata Putri, Hendra Effendi | Rapid Application Development (RAD) | Penelitian ini menghasilkan sebuah kesimpulan bahwa pengunjung merasa mudah dan terbantu dengan adanya <i>website</i> informasi wisata air terjun di sumatera selatan. |
| 3 | Penerapan <i>Website</i> Sebagai Sarana | Aris Sudianto | teknik survey | Penelitian ini menghasilkan <i>website</i> yang dapat mendorong minat wisatawan dengan memberikan sarana media |

| | | | | |
|---|---|---|---|--|
| | Promosi Wisata Budaya pada Kabupaten Lombok Timur | | | promosi dan informasi yang dapat diakses oleh para wisatawan yang ingin berkunjung ke pulau Lombok |
| 4 | Pembuatan <i>Website</i> Sebagai Media Promosi Wisata Pulau Mubut | Swono Sibagariang, Yeni Rokhayati, Afdhol Dzikri, Siskha Handayani, Metta Santiputri, Agung Riyadi, Nur Zahrati Janah, Muhamad Sahrul Nizan | metode observasi, sosialisasi dan pelatihan | penelitian ini menghasilkan kesimpulan bahwa <i>website</i> pulau Mubut dapat dikategorikan baik dan mudah dimengerti, sebanyak 81.9% pertanyaan dapat dijawab oleh responden sedangkan 18.1% tidak dapat menjawab pertanyaan yang diajukan |
| 5 | Rancang Bangun <i>Website</i> Desa Citengah Untuk Pengemban gan Promosi | Reny Rian Marlina, Wagus Sejati, Wilda Ainun Nisa, Uun Pujayanti, Rizki Sopian, Wahyu Noergana | metode prototype | penelitian ini menghasilkan bahwa <i>website</i> yang dibangun oleh para pemuda desa Citengah kabupaten Sumedang yang dapat diakses dengan mengunjungi akses link https://citengah.sumedangdesa.id kegiatan pembuatan <i>website</i> ini berdampak positif terhadap |

| | | | | |
|---|---|--|---------------------|---|
| | Potensi Desa | | | perkembangan informasi desa yang dapat diakses oleh banyak pihak termasuk oleh para calon wisatawan desa |
| 6 | Perancangan Sistem Aplikasi Penjualan Paket Wisata Pulau Seribu Berbasis Java Dengan Metode <i>Scrum</i> (Studi Kasus PT. Abarter Global Indonesia) | Irfan Fakhruddin, Nofita Rismawati dan Rini Sriyanti | Metode <i>Scrum</i> | bahwa dengan membuat sistem aplikasi penjualan paket wisata pulau seribu yang dibangun dengan metode scrum untuk PT. Abater Global Indonesia memberikan dampak yang positif seperti kemudahan dalam memesan paket wisata, gerbang pembayaran yang sederhana, dan pelaporan yang lebih efisien dan mudah dilakukan oleh pihak-pihak yang terkait |

Dari beberapa jurnal penelitian terdahulu terdapat kelebihan dan kekurangannya masing-masing, tetapi peneliti memilih menggunakan metode *scrum* karena metode *scrum* merupakan metode *iterative* dimana metode ini membutuhkan feedback dari pengguna. Terdapat pula *short Sprint* dan *constant feedback*, *Scrum* bisa mengatasi perubahan yang terjadi. Dengan metode *scrum*,

ketika terjadi masalah mudah identifikasi pada pertemuan harian, sehingga masalah dapat terselesaikan dengan cepat.

2.2 Dasar Teori

Pada Penelitian ini menggunakan beberapa dasar teori sebagai pendukung penelitian yang diambil dari beberapa sumber. Berikut dasar teori yang akan digunakan pada penelitian ini:

2.2.1 Sistem Informasi

Sistem informasi yaitu suatu alur atau rangkaian yang membentuk suatu kesatuan informasi dan terbentuk dari elemen dan komponen yang saling berhubungan guna memperlancar arus informasi. Sistem informasi juga dapat diaplikasikan pada berbagai macam bidang sehingga pengertian sistem informasi juga dapat berbeda-beda. Sistem merupakan bagian-bagian yang saling melengkapi, dapat berinteraksi dan bekerja sama untuk mencapai suatu tujuan tertentu (Pinontoan, 2008).

2.2.2 Potensi Wisata

Potensi wisata merupakan kumpulan benda (kekayaan alam, nilai kebudayaan, benda buatan manusia) yang perlu untuk diolah untuk mendatangkan nilai daya tarik bagi wisatawan (Janianto Damanik dan Helmut F. Weber, 2006).

Desa wisata adalah desa nilai-nilai kekayaan berupa keunikan budaya, lingkungan alam yang indah dan atau kehedipuan sosial masyarakat yang berbeda, memberikan nya nilai potensi keunikan dan daya tarik sendiri (Atmoko, 2014).

Berpendapat bahwa desa wisata adalah tempat atau kawasan yang memiliki nilai kearifan lokal seperti budaya, nilai sosial masyarakat dan adat istiadat yang dikelola untuk keuntunagn ekonomi sesuai dengan kapasitasnya. Dari pendapat beberapa ahli menyimpulkan bahwasannya desa wisata merupakan objek desa yang memiliki nilai kebudayaan, adat istiadat, nilai sosial masyarakat dan kekayaan alam yang unik untuk

dikembangkan menjadi daya tarik bagi wisatawan yang ingin berkunjung (Hermawan, 2016).

2.2.3 Website

Pengertian halaman web biasa disebut dengan Web, yang dapat dipahami sebagai situs yang berisi halaman untuk memberikan informasi berupa teks, audio, gambar statis dan dinamis maupun gabungan dari informasi tersebut. Halaman pada web saling berhubungan dan membentuk satu kesatuan utuh informasi yang dihubungkan oleh halaman web atau *hyperlink*[13].

Web dapat melakukan penyorotan terhadap bentuk informasi seperti teks, audio, gambar statis, gambar dinamis maupun gabungan dari informasi tersebut. Dalam hal ini penyorotan dapat berupa garis bawah, penekanan maupun pemberian label khusus untuk informasi tersebut agar mudah diakses dan mendapatkan perhatian yang lebih tinggi. Dengan menggunakan *browser* antarmuka pengguna (GUI) tautan dapat dihubungkan ke tujuannya dengan mengarahkan cursor atau mengetuk tautan informasi yang sudah disediakan[14].

2.2.4 Hypertext Markup Language (HTML)

HTML merupakan bahasa pemrograman yang dapat digunakan dalam membangun sebuah halaman atau *landing page*. Dalam pembangunan web setiap halaman web ditulis dengan format HTML. Seluruh informasi konten, dokumen, hyperlink yang dapat diklik, gambar, formulir yang dapat diisi, dokumen multimedia, dan lainnya ditulis dalam bahasa pemrograman HTML. HTML lebih berkaitan dengan penyajian elemen struktural dan pemformatan halaman web daripada tampilannya. HTML tidak ditujukan sebagai sebuah bahasa pemrograman yang untuk membuat aplikasi desktop melainkan sebuah bahasa yang dirancang untuk pengkodean *World Wide Web*. Contoh HTML[15]. HTML adalah bahasa pemrograman untuk pembuatan web yang fleksibel dan tidak bergantung pada sistem operasi tertentu. Pada

HTML menjelaskan elemen dari komponen HTML tertentu dalam sebuah halaman web[16].

2.2.5 Hypertext Preprocessor (PHP)

PHP merupakan sebuah bahasa pemrograman yang berfungsi untuk pembuatan web aplikasi dengan kemampuan pemrosesan [17]. *Hypertext Preprocessor* adalah salah satu bahasa pemrograman yang hanya dapat dijalankan pada sisi server. PHP merupakan skrip pemrograman yang ditempatkan dan dieksekusi di server seperti dapat menulis dan menerima, serta dapat mengolah dan dapat menampilkan data dari *website* maupun ke *website*. Jika tidak ada server web yang berjalan terus menerus, itu tidak akan berfungsi[18]

2.2.6 XAMPP

XAMPP adalah perangkat lunak yang namanya diambil dari singkatan Tanda “X” yang memiliki arti cross-platform sebagai simbol bahwa perangkat lunak ini dapat bekerja di empat sistem operasi yang berbeda, seperti Linux, Windows, Macintosh, dan Solaris. Xampp juga mendukung beberapa model data base seperti Apache, MySQL/MariaDB, PHP dan Perl. [19].

XAMPP berfungsi seperti web di mesin lokal. XAMPP juga dikenal sebagai server Cpanel virtual, sehingga memudahkan dalam pratinjau dan mengedit web tanpa mengakses internet.

2.2.7 PhpMyAdmin

PhpMyAdmin merupakan alat yang mudah digunakan dalam mengelola database MySQL dan MySQL Server dengan cara yang intuitif, jadi tidak perlu lagi untuk menulis query pada SQL setiap kali kita mengeksekusi data perintah dasar. Anda bisa mendapatkan fitur ini ketika Anda menginstal triple pack phpMyAdmin, karena sudah termasuk dalam xampp pra-instal [20].

2.2.8 MySQL

MySQL merupakan server basis data yang banyak digunakan untuk membuat aplikasi web dengan basis data sebagai sumber dan mengelola data. Mudah-mudahan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya, sehingga *tools* ini mudah digunakan.[21]. MySQL adalah perangkat lunak yang dapat digunakan dalam pembuatan database yang bersifat open source.

Pada MySQL terdapat bahasa perintah atau query yaitu Structured Query Language atau yang biasa disebut SQL. Terdapat 3 struktur dasar dalam ekspresi SQL:

- a. Terdapat Select yang digunakan dalam mendaftar seluruh atribut yang mau kita gunakan sebagai hasil suatu query.
- b. Pada from, digunakan untuk mencatat seluruh relasi yang diamati ketika melakukan evaluasi sebuah query.
- c. Pada where, merupakan suatu predikat yang dapat menyangkut atribut dari relasi yang ada dalam klausa from.

Ekspresi dasar modifikasi data ada 3, yaitu:

- a. Delete, digunakan dalam menghapus data.
- b. Insert, digunakan untuk memasukkan data, dan
- c. Update, digunakan dalam mengganti data.

2.2.9 Visual Studio Code

Visual Studio Code adalah alat pemrograman aplikasi visual. Program *Visual Studio Code* adalah salah satu perangkat lunak pembuatan dan desain situs web sederhana. *Visual Studio Code* merupakan sebuah editor kode yang telah dikembangkan oleh Microsoft untuk berbagai macam sistem operasi seperti Windows, Linux dan Macintosh. Didalamnya termasuk dukungan debug, kontrol versi git dan GitHub terintegrasi, *highlight* sintaks, rekomendasi penyelesaian kode, cuplikan kode, dan perubahan format ulang kode. Jadi mudah dimanipulasi, sehingga

pengguna dapat mengubah tema, hotkey, opsi, dan melakukan pengaturan add-on yang menambah fungsionalitas (Budi, 2012).

2.2.10 Scrum

Scrum adalah kerangka kerja inkremental responsif untuk pengembangan dan pembuatan perangkat lunak yang digunakan dalam mengatur produk atau pengembangan aplikasi yang menjadi fokus pada "bekerja secara strategis dan fleksibel sebagai satu unit untuk mencapai tujuan." bersama"[22].

Scrum dapat digunakan dalam pengembangan sistem menyeluruh, dapat mengembangkan sistem sebagian ataupun dalam proyek pribadi/internal [23].

Fokus utama pada *Scrum* yaitu *test* dan *adapt*, artinya Anda melihat masalah tersebut dan anda beradaptasi dengan masalah tersebut. Pengembangan *software* dengan *Scrum* memfokuskan untuk melihat setiap fase pengembangan perangkat lunak dengan lebih singkat.

Fungsionalitas *Scrum* memfokuskan pada penyelesaian suatu fungsi atau fitur dimana sudah terdapat kualitas, pengujian, dokumentasi, evaluasi, dll. [23].

1. Product Backlog

Product Backlog daftar elemen yang dibutuhkan dalam produk dan asal persyaratan yang dibutuhkan untuk pembuatan produk. Backlog harus diambil dan diurutkan berdasarkan prioritas tertentu. Faktor-faktor ini dapat menjadi hasil penemuan kebutuhan pelanggan, menjelaskan fungsional dan non-fungsional dan apa yang dibutuhkan untuk menciptakan produk jadi.

2. Sprint Backlog

Sprint Backlog yaitu sekumpulan barang *product backlog* yang sudah ditunjuk pada proses selama *Sprint*, termasuk juga rencana dalam mengembangkan bagian produk tambahan dan mencapai tujuan *Sprint*. Fungsionalitas pada *Sprint backlog* merupakan bagian yang

tersedia pada iterasi berikutnya dan juga pekerjaan yang diperlukan dalam membawa fungsionalitas ke dalam hasil akhir produk.

3. *Sprint*

Sprint adalah *Scrum-centric*, yang berarti jangka waktu satu bulan atau kurang, setelah itu pekerjaan dapat dianggap "selesai", dapat digunakan, dan kemungkinan besar akan dirilis. *Sprint* pada umumnya berdurasi konstan selama pengembangan produk dalam metode *scrum*, terdapat beberapa bagian, yaitu:

a. *Sprint Planning*

Sprint Planning merupakan kegiatan untuk membuat susunan rencana Product Backlog Item yang telah siap dikembangkan oleh tim pengembang tiap satu kali *sprint*. Terdapat hal-hal yang harus diperhatikan ketika ingin melakukan *Sprint Planning*, berikut penjelasannya:

1) Menentukan tujuan *sprint*

Tujuan *sprint* sangat penting dalam proses perencanaan *sprint*. Karena berfungsi untuk menghindari kebingungan tim dalam melaksanakan pengerjaan tiap *sprint*[24]

2) Menentukan panjang *sprint*

Salah satu hasil dari *Sprint Planning* adalah panjang *sprint* atau penetapan tanggal demo. Sebuah *sprint* yang pendek memungkinkan perusahaan menjadi lebih lincah, lebih sering melakukan delivery, dan lebih banyak memberikan umpan balik. Namun, sebuah *sprint* yang panjang juga memiliki kelebihan yaitu tim memiliki lebih banyak waktu untuk membangun momentum serta memiliki ruang yang cukup untuk menyelesaikan masalah dan mencapai tujuan *sprint*. Dari dua batasan waktu di atas, menentukan waktu yang tepat untuk *sprint* merupakan hal yang penting untuk dilakukan[24].

3) Perkiraan Waktu Pengerjaan Dan Memecah Item Backlog Ke Ukuran Yang Lebih Kecil Jika Diperlukan

Perkiraan waktu pengerjaan sangat dibutuhkan untuk menyamakan nilai estimasi yang telah dipikirkan oleh pemilik produk pada product backlog yang tidak sesuai dengan tim pengembang. Perkiraan besarnya nilai estimasi akan lebih mudah dan akurat bila dipecah ke dalam bentuk yang lebih kecil[24].

- 4) Memutuskan Item Backlog Yang Akan Diikuti dalam Sprint
Salah satu aktivitas utama dalam perencanaan sprint adalah menentukan item Backlog yang akan diikuti dalam Sprint. Aktivitas ini akan mengubah product backlog menjadi sprint backlog. Untuk memutuskan item backlog yang akan diikuti tim dapat menggunakan teknik perhitungan kecepatan yang terdiri dari 2 tahap sebagai berikut[24].

- a) Memutuskan perkiraan kecepatan

Untuk memutuskan perkiraan kecepatan, pertama yang harus dilakukan adalah menghitung *man-days* yang dimiliki oleh tim pengembang. Cara perhitungannya adalah dengan mengalikan hari kerja dalam sprint dengan jumlah tim pengembang. Contoh: panjang Sprint yang ditetapkan 3 (tiga) minggu atau 18 (delapan belas) hari kerja dengan anggota tim pengembang 3 (tiga) orang, maka didapatkan perhitungan $man-days = 18 \times 3 = 54$. Jumlah *man-days* yang didapatkan merupakan jumlah kecepatan tim ideal ketika kondisi pengerjaan tidak mendapat gangguan. Namun hal itu sulit terjadi dikarenakan adanya kejadian-kejadian tidak terduga yang dialami tim pengembang, salah satu contohnya adalah sakit atau waktu terbagi dengan aktivitas pengembangan lain. Oleh karena itu dibutuhkan sebuah *focus factor* dibutuhkan untuk menangani masalah ini. *Focus factor* adalah estimasi tingkat fokus tim. Tingkat *focus factor* yang kecil menunjukkan tim mendapatkan

banyak gangguan atau memperkirakan waktu pengerjaan dengan terlalu optimis. Cara terbaik menentukan *focus factor* adalah melihat performa dari sprint sebelumnya.

Last Sprint's Focus Factor:

$$\text{Focus Factor} = \frac{\text{Actual Velocity}}{\text{Vailable Man-Day}}$$

Perkiraan kecepatan dapat ditentukan dengan rumus berikut:

Sprint's Estimated Velocity:

$$\text{Available Man-Days} \times \text{Focus Factor} = \text{Estimated Velocity}$$

- b) Menghitung banyak item backlog yang akan ditambahkan tanpa melebihi perkiraan kecepatan yang telah ditentukan. Setelah mendapatkan perkiraan kecepatan tim pada sprint, tim akan memilih item backlog yang akan diikuti ke dalam sprint selanjutnya dengan pertimbangan pemilik produk. Pertimbangan yang akan dilakukan oleh pemilik produk bersifat tidak wajib diikuti, karena yang bertanggung jawab dalam penentuan item backlog dalam sprint adalah tim pengembang[24].

5) Memecah Backlog menjadi Task

Backlog yang telah dimasukkan pada Sprint Backlog akan dipecah atau dijabarkan menjadi task. Hal ini akan mempermudah tim Scrum dalam melakukan pengerjaan item Backlog[24].

b. *Daily Scrum*

Daily Scrum merupakan pertemuan sementara pada batasan waktu maksimal 15 menit, sehingga tim pengembang dapat mengatur pekerjaan dalam perencanaan untuk 24 jam ke depan. *Daily Scrum* menjadikan tim mengatur pekerjaan dengan mudah, sehingga mengungkapkan seluruh hambatan yang di dapat setiap harinya. Terdapat juga *product owner* yang selalu memantau *daily*

scrum untuk menyaksikan secara langsung progress yang telah dibuat oleh tim pengembang.

c. *Sprint Review*

Sprint Review merupakan kegiatan atau acara yang dilakukan pada akhir sprint dengan tujuan untuk memantau proses pekerjaan tim pengembang saat sprint terakhir dikerjakan.

d. *Sprint Retrospective*

Sprint Retrospective merupakan kegiatan untuk tim *Scrum* melihat kinerja sendiri sehingga dapat melakukan peningkatan untuk diimplementasikan saat sprint selanjutnya[23].


2.2.11 *Unified Modeling Language (UML)*




Unified Modeling Language adalah proses pemrograman dengan konsep berorientasi objek yang terjadi karena adanya pemodelan dengan gambar visual yang berfungsi dalam menampilkan serta mencatat prosesnya pada sistem *software*. Ada beberapa diagram UML yang populer digunakan saat mengembangkan ataupun pembuatan sistem, keterangannya sebagai berikut:

A. *Use Case Diagram*

Use case diagram merupakan bagian dari diagram dalam kategori *Unified Modeling Language* yang digunakan dalam menampilkan hubungan dan interaksi pengguna dengan sistem. Pada diagram lebih terfokus pada "*use case*" atau skenario penggunaan yang menampilkan interaksi antara aktor-aktor pada sistem agar mencapai mencapai suatu tujuan tertentu.

Tabel 2.2 Simbol Use Case Diagram

| Simbol | Keterangan |
|---|--|
|  | <i>Use Case</i> ini berfungsi sebagai bagian yang dapat bertukar |



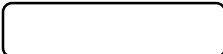
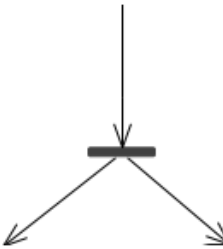
| Simbol | Keterangan |
|---|---|
| | komunikasi dengan para actor sehingga dapat dinyatakan dengan dengan kata kerja |
|  | <p><i>Actor</i> atau pemeran adalah simbol dari orang atau pemeran lain yang dapat menggunakan fungsi dari sistem tertarget. Actor harus diidentifikasi terlebih dahulu dengan menentukan pembagian tugas kerja dan tugas yang juga berkaitan pada peran dalam konteks system tertarget. Actor dapat ditampilkan dalam beberapa peran sekaligus maupun tidak. Perlu diingat bahwa aktor dapat berkomunikasi dengan <i>Use Case</i>, tetapi tidak memiliki akses untuk mengatur terhadap <i>use case</i></p> |
|  | Hubungan antara actor dan <i>use case</i> disimbolkan dengan garis tanpa panah dan tidak putus-putus yang menunjukkan siapa atau apa berinteraksi secara langsung dan tidak menunjukkan data. |
|  | Hubungan antara actor dan <i>use case</i> yang disimbolkan dengan panah terbuka untuk menunjukkan jika aktor berkomunikasi pasif dengan sistem |

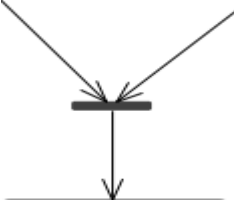
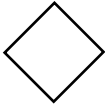
B. Activity Diagram

Activity diagram menampilkan kegiatan untuk sistem yang sedang dalam proses merancang, dan bagaimana setiap aliran berawal, setiap pilihan yang mungkin dapat terjadi dan bagaimana proses mereka dapat berakhir.

Activity diagram menampilkan proses sebuah system yang berjalan secara paralel yang dapat terjadi dari beberapa hasil eksekusi.

Tabel 2. 3 Activity Diagram


| Simbol | Keterangan |
|---|--|
|  | Titik mulai, berada pada posisi bagian pojok sampig kiri atas dan berfungsi sebagai penanda awal dari aktivitas |
|  | Titik akhir, penandai dai beakhirnya aktivitas |
|  | Aktivitas, menunjukan suatu pekerjaan atau proses dari kegiatan bisnis |
|  | <i>Fork</i> atau percabangan, adalah indikasi untuk menunjukan kegiatan atau aktivitas yang dapat dilakukan secara paralel |




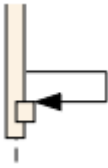

| Simbol | Keterangan |
|--|--|
| | atau untuk dapat menggabungkan dua aktivitas menjadi satu aktiviats |
|  | <i>Join</i> (penggabungan) atau <i>rake</i> , adaalah indikasi untuk mennjukan adanya proses dekomposisi pada aktivitas yang bercabang |
|  | <i>Decision Points</i> (pengambilan keputusan), adalah indikasi untuk menunjukan pemilihan keputusan <i>true or false</i> |


C. Sequence Diagram

Daigram ini dapat digunakan sebagai gambaran visual yang menunjukan komunikasi antara objek didalam dan aktivitas disekitar system yang berbentuk pesan yang digambarkan dari langkah ke langkah. Sequence diagram meliputi vertikal (waktu) dan horizontal (objek terkait).

Tabel 2. 4 Sequence Diagram

| Simbol | Keterangan |
|---|---|
|  | <i>Entity Class</i> (kelas entitas), adalah bagian dari sistem yang terdiri dari kumpulan beberapa kelas yang berupa entitas-entitas sehingga membentuk |


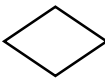
| Simbol | Keterangan |
|---|--|
| | rancangan awal sistem dan menjadi acuan untuk menyusun basis data |
|  | <i>Boundary Class</i> (kelas penghubung), adalah kelas yang terdiri dari kelas-kelas yang menjadi interfaces atau interaksi antara actor, sistem, dan kelas lainnya, seperti contoh pada <i>form entry</i> dan <i>form cetak</i> |
|  | <i>Control class</i> (kelas kontrol), adalah sebuah kelas yang di dalamnya terdapat logika aplikasi dan tidak memiliki tanggung jawab atas entitas. |
|  | <i>Message</i> (pesan), adalah sebuah simbol yang menunjukkan pengiriman pesan anatar <i>class</i> |
|  | <i>Recursive</i> (rekursif), adalah sebuah simbol yang menunjukan pengiriman pesan untuk dirinya sendiri |
|  | <i>Activation</i> (aktivasi), adalah sebuah simbol yang menunjukan tampilan hasil operasi objek panjang persegi |

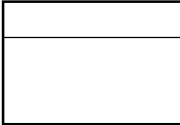



| Simbol | Keterangan |
|---|--|
| | selaras dengan penagktifan proses |
|  | <i>Lifeline</i> (garis alur), adalah sebuah simbol yang menunjukkan hubungan antara objek dan terdapat aktivitas didalam nya |

D. Class Diagram

Class diagram merupakan tampilan struktur kelas pada suatu sistem serta merupakan jenis diagram yang paling sering digunakan. *Class diagram* dapat melihat hubungan interaksi antar class dan memberikan penjelasan detail dari setiap layer pada sebuah model desain dari suatu sistem.

Tabel 2.5 Class Diagram

| Gambar | Nama | Keterangan |
|---|-------------------------|---|
|  | <i>Generalization</i> | Adalah simbol hubungan antara ahliwaris (<i>descendant</i>) dapat berbagi perilaku dan stuktur daata yang sama dengan pewaris (<i>ancestor</i>) |
|  | <i>Nary Association</i> | Simbol untuk mengindari terjadinya asosiasi yang lebih dari 2 objek. |

| Gambar | Nama | Keterangan |
|---|----------------------|--|
|  | <i>Class</i> | Adalah simbol untuk kumpulan objek yang berbagi atribut dan operasi yang sama. |
|  | <i>Collaboration</i> | Adalah simbol untuk menunjukkan urutan dari aksi-aksi yang ditampilkan oleh sistem sehingga menghasilkan hasil yang terukur bagi aktor |
|  | <i>Realization</i> | Adalah simbol yang menunjukkan operasi dari objek yang benar-benar dikerjakan |
|  | <i>Dependency</i> | Adalah simbol yang menunjukkan letak perubahan yang terjadi didalam suatu elemen mandiri (<i>indepent</i>) |

2.2.12 Black-box Testing

Pengujian *Black Box* merupakan pengujian yang dilakukan pada software untuk menguji fungsionalitasnya tanpa pengujian desain dan kode program untuk memastikan bahwa apakah fungsi *website*, *input* dan *output software* sesuai kebutuhan yang di inginkan[25].berikut merupakan teknik pengujian *Black Box*



Gambar 2. 1 Pengujian *Black Box*

Banyak jenis pada *Black Box* yang di ujikan oleh penguji. Namun, terdapat beberapa jenis tes yang sering digunakan dalam pengujian, seperti *functional testing* yang bertujuan untuk percobaan fitur secara rinci apakah *website* dapat berjalan sesuai yang dibutuhkan atau tidak, *non functional testing* yang bertujuan untuk menguji aspek tambahan dari *software* dan *regression testing* dengan melakukan uji regresi melalui dua aspek, yaitu *aspek fungsional* dan *non fungsional*. [26].

2.2.13 System Usability Scale (SUS)

Sistem Usability Scale adalah metode penilaian yang biasa digunakan dalam menghitung usability dari produk, layanan, atau sistem. Dalam pengembangannya metode ini berhasil dikembangkan pada tahun 1986 oleh John Brooke yang telah membuatnya menjadi populer digunakan untuk mengukur persepsi pengguna terhadap tingkat kegunaan suatu produk atau sistem. *System usability scale* ini digunakan agar mendapatkan penilaian dan kemanfaatan pada aplikasi ini setelah menggunakan metode tersebut. Sistem ini umumnya terdiri dari 10 soal menggunakan skala Likert dari 1 sampai dengan 5. Terdapat soal ganjil dengan nilai (1, 3, 5, 7, 9) adalah soal positif, soal genap (2, 4, 6, 8, 10) adalah soal negatif. Dimana negatif ini seperti yang telah ditunjukkan pada Tabel 1. Dimana responden diharuskan untuk memberikan jawaban atas semua pertanyaan yang berkaitan dengan SUS setelah pengguna telah selesai mengisi data menggunakan sistem secara penuh [27].

Tabel 2. 6 Daftar pertanyaan SUS

| No | Pertanyaan |
|----|---|
| 1 | Saya sepertinya akan sering menggunakan <i>website ini</i> |
| 2 | Saya melihat ada bagian menu <i>website ini</i> yang cukup merepotkan |
| 3 | Saya rasa <i>website ini</i> mudah digunakan |
| 4 | Saya sepertinya membutuhkan bantuan teknisi agar dapat menggunakan <i>website ini</i> dengan lancar |
| 5 | Saya rasa menu pada <i>website ini</i> sudah terintegrasi dengan baik |
| 6 | Saya menemukan terlalu banyak ketidak konsistenan dalam <i>website ini</i> |
| 7 | Saya pikir orang-orang akan dapat menggunakan <i>website ini</i> dalam waktu yang sangat cepat |
| 8 | Saya rasa <i>website ini</i> sangat sulit digunakan |
| 9 | Saya merasa tidak ada hambatan dalam menggunakan system ini |
| 10 | Saya harus belajar banyak hal terlebih dahulu sebelum menggunakan <i>website</i> |

Terdapat skor pada pengisian ini dimana item-item pertanyaan mempunyai nilai kontribusi. Setiap nilai tersebut dapat berkontribusi item dengan nilai berkisar antara 0-4[28].

- a. Pada soal ganjil (1, 3, 5, 7, 9) penilaian skor kontribusinya yaitu nilai posisi dari skala dikurangi 1.
- b. Pada soal genap yang terdiri dari 2, 4, 6, 8, 10 memiliki nilai kontribusi skor yaitu 5 dikurangi dari posisi skala.
- c. Hitung dengan mengkalikan jumlah skor kontribusi dengan 2,5 untuk menentukan nilai keseluruhan pada sistem.
- d. Apabila nilai dari skor tiap responden telah diketahui, lakukan perhitungan untuk mencari nilai skor rata-rata dengan menjumlahkan semua hasil skor dan dibagi dengan jumlah

responden yang ada. Berikut rumus untuk menghitung skor *System usability scale* (SUS):

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

\bar{x} = merupakan rata-rata skor

$\sum x$ = Jumlah seluruh skor SUS

n = merupakan jumlah responden

Nilai rata-rata pada skor SUS minimal bernilai 68, apabila nilai kurang dari 68 maka dinyatakan bahwa pengguna tidak puas pada sistem ini, untuk mendapatkan nilai tingkat hasil evaluasi dapat dilakukan dengan menggunakan 2 cara, seperti dapat perspektif penerimaan pengguna, skala peringkat dan peringkat objektif mencakup tingkat penerimaan pengguna, terdapat 3 kategori sebagai tidak dapat diterima, marginal dan dapat diterima.

Berdasarkan hasil tingkatan skalanya, terdapat enam skala penilaiannya antara lain A, B, C, D, E, F. *Sistem Usability Scale* (SUS) adalah metode penilaian yang digunakan untuk mengukur tingkat kegunaan atau usability dari suatu produk, layanan, atau sistem. Metode ini dikembangkan oleh John Brooke pada tahun 1986 dan telah menjadi salah satu alat yang populer digunakan untuk penentuan ini dapat ditunjukkan pada Tabel 2.7 di bawah ini:

Tabel 2. 7 Kategori SUS

| <i>Grade</i> | Keterangan |
|--------------|------------------------------|
| A | skor $\geq 80,3$ |
| B | $74 \leq \text{skor} < 80,3$ |
| C | $68 \leq \text{skor} < 74$ |
| D | $51 \leq \text{skor} < 68$ |
| E | skor < 51 |

Dalam mengukur persepsi pengguna terhadap tingkat kegunaan suatu produk atau sistem.terdiri dari worst imaginable, poor, ok, good, excellent dan best imaginable. Penentuan kedua ini dapat dilihat dari rentang persentase SUS yang peringkatnya meliputi A, B, C, D, E. Penentuan peringkat didasarkan pada peringkat persentil dari skor SUS yang biasa digunakan. dibuat berdasarkan hasil perhitungan review pengguna[29].