

BAB II

TINJAUAN PUSTAKA & LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang berjudul **“Perancangan Aplikasi *Mobile E-Marketplace* Rumah Kost (Studi Kasus: Kabupaten Karawang)”** oleh Darmansyah, Popi Parpati, Rahmat Gunawan, Kabupaten karawang memiliki jumlah perantau mencapai 1,7 juta jiwa hingga tahun 2018 dengan itu para perantau memerlukan tempat tinggal sementara atau rumah kost. Penelitian ini membahas tentang perancangan aplikasi *mobile* untuk mengatasi permasalahan berupa pengolahan informasi kos yang didapatkan oleh pencari kos masih melalui cara dengan bertanya ke teman atau, kemasayarakat sekitar dan mendatangi setiap kosnya. Tujuan dilakukan penelitiannya untuk mengembangkan sebuah aplikasi *mobile* untuk rumah kos menggunakan metode SDLC. Aplikasi ini akan menggunakan *framework* UI *Bootstrap*, bahasa pemrograman PHP *framework Codeigniter*, serta sistem manajemen basis data (DBMS) *MySQL*. Pemodelan aplikasi ini diterapkan dengan metode berorientasi objek menggunakan *Unified Modelling Language* (UML) dan pengujian dilakukan menggunakan metode *blackbox testing*. Penelitian ini berhasil mengembangkan sebuah aplikasi *mobile* yang bertujuan membantu pengguna dalam pencarian informasi rumah kos di Kabupaten Karawang dengan mudah serta memfasilitasi pemilik kos dalam memasarkan rumah kos mereka [14].

Penelitian dengan judul **“Perancangan Sistem Informasi Pemesanan Kamar Kost Pada Rukost Ayah Bunda Menggunakan Metode *Design Thinking*”** yang dibuat oleh Nurul Fitrah Hayati, Arie Rafika Dewi, Fachrul Rozi Lubis, membahas tentang perancangan sebuah *website* sistem informasi untuk mengatasi permasalahan seperti penyewaan rumah kos dan penagihan biaya sewa yang masih dilakukan secara manual. Berdasar dari penelitian yang telah dibuat, terungkap tujuannya yaitu untuk merancang *website* sistem informasi pemesanan kamar kost yang memungkinkan calon penyewa untuk memesan kamar kost tanpa harus mengunjungi penyedia kamar kost *Website* ini dikembangkan dengan

penerapan metode *Design Thinking*, bahasa pemrograman PHP, *framework Codeigniter*, dan database *MySQL*. Penelitian ini berhasil membuat sistem informasi untuk membantu para *customer* melakukan pemesanan kamar kost dan mempermudah admin dalam pendataan penyewa [15].

Penelitian selanjutnya berjudul **“Sistem Informasi Manajemen Kost Menggunakan *Framework Laravel* (Studi Kasus Pada Kost Griya Stonen)”** yang dikerjakan oleh Khumaidah, Agung Riyantomo, mengangkat permasalahan meliputi Kurangnya sistem informasi yang dapat memantau keuangan, penyewa, dan pemesanan kamar kost. Dari penelitian yang dibuat memiliki tujuan dan hasil yaitu untuk mengembangkan sistem informasi berbasis internet yang mampu mengumpulkan data terkait ketersediaan kamar kos, fasilitas kost, serta fitur penyewaan kamar kos secara *online* untuk calon penyewa di kost Griya Stone. Sistem ini dikembangkan dengan menggunakan metode *Waterfall*, *framework Laravel*, database *MySQL*. Penelitian ini berhasil membuat sistem untuk memudahkan admin dalam memajemen kos dan calon penggunanya dalam melakukan sewa kamar kos [16].

Penelitian ini berjudul **“Implementasi Metode *Prototyping* Pada Rancang *Marketplace* Rumah Kost Berbasis *Mobile*”** yang dibuat oleh Dini Silvi Purnia, Ratningsih, Mumun Surahman, Widhiani Agustin memiliki permasalahan kesulitan dalam mencari informasi kost secara cepat dan akurat, serta sulitnya pemilik kost dalam mempromosikan tempat kostnya dengan luas. Penelitian ini memiliki tujuan dan hasil untuk merancang *marketplace* rumah kos berbasis *mobile* agar menyediakan solusi bagi pengguna dalam mencari dan memesan kost dengan mudah dan cepat melalui aplikasi *mobile*. Perancangan aplikasi ini menggunakan metode pengembangan *prototyping*. Dari penelitian ini memperoleh hasil yaitu menjadi sarana yang efektif dan memudahkan baik bagi penyewa maupun pemilik dalam menggunakan aplikasi rumah kos [17].

Penelitian selanjutnya berjudul **“Aplikasi *Booking* Kost Berbasis *Android* Di Kota Palangka Raya”** yang dibuat oleh Enny Dwi Oktaviani, Deddy Ronaldo,

Mustafa Arifin dengan mengangkat topik permasalahan berupa sulitnya mencari tempat tinggal sementara atau kos bagi mahasiswa yang akan menjalani pendidikan di tempat yang jauh dari tempat mereka berasal. Tujuan dan hasil dibuatnya penelitian ini adalah untuk membuat sebuah aplikasi *android* yang mempermudah mahasiswa saat mencari dan memesan tempat tinggal sementara atau kos di kota Palangka Raya. Aplikasi dibuat menggunakan metode *waterfall*, menggunakan bahasa pemrograman PHP, HTML, CSS dan Javascript, dengan menggunakan *framework Codeigniter* dan *bootstrap* [18].

Penelitian selanjutnya berjudul “**Sistem Informasi Manajemen Kost Putra Trisula Berbasis Web (Studi Kasus: Asrama Putra Trisula)**” yang dikerjakan oleh Wayan Darlin, Ade Dwi Putra, Nirwana Hendrastuty, terbuatnya penelitian tersebut melibatkan permasalahan yang didapat seperti proses manajemen kost yang masih dilakukan secara manual, termasuk dalam hal pencatatan data penyewa kost, pembayaran, dan pelaporan keluhan. Tujuan dan hasil dengan penelitian ini dilakukan untuk membangun sistem informasi manajemen kost yang bisa memperbaiki proses manajemen kost yang masih dilakukan secara manual. *Website* dibuat menggunakan metode *prototyping* dengan bahasa pemrograman PHP dan *framework codeigniter*. [19]

Tabel 2. 1 Tabel Penelitian Terdahulu

NO	JUDUL	METODE	HASIL	PERBEDAAN
1	Sistem Informasi Manajemen Kost Putra Trisula Berbasis Web (Studi Kasus: Asrama Putra Trisula) (2023)[19]	<i>Prototyping</i>	Sistem informasi manajemen kost berbasis <i>website</i> yang dapat memperbaiki proses manajemen kost yang masih dilakukan secara manual	Penulis menggunakan metode <i>extreme programming</i> , sedangkan penelitian ini menggunakan metode <i>prototyping</i> .
2	Perancangan Sistem Informasi Pemesanan Kamar Kost Pada Rukost Ayah Bunda Menggunakan Metode <i>Design Thinking</i> (2022)[15]	<i>Design Thinking</i>	<i>Website</i> sistem informasi pemesanan kamar kost untuk membantu calon penyewa memesan kamar kost tanpa harus pergi ke penyedia kamar kost	Pada penelitian ini mengembangkan sebuah <i>website</i> sistem informasi dengan menggunakan metode <i>design thinking</i> . Sedangkan penulis mengembangkan sistem informasi berbasis <i>website</i> dengan metode

				pengembangan <i>extreme programming</i> .
3	Implementasi Metode <i>Prototyping</i> Pada Rancang <i>Marketplace</i> Rumah Kost Berbasis <i>Mobile</i> (2021)[17]	<i>Prototyping</i>	<i>Marketplace</i> rumah kos berbasis <i>mobile</i> agar menyediakan solusi bagi pengguna dalam mencari dan memesan kost dengan mudah dan cepat melalui aplikasi <i>mobile</i>	Pada penelitian ini pengembangan aplikasinya berbasis <i>mobile</i> dengan menggunakan metode pengembangan <i>prototyping</i> . Sedangkan penulis mengembangkan sistem informasi mengenai rumah kos berbasis <i>website</i> dengan metode pengembangan <i>extreme programming</i>
4	Perancangan Aplikasi <i>Mobile E-Marketplace</i> Rumah Kost (Studi	SDLC	Aplikasi <i>mobile</i> yang dapat membantu pengguna untuk melakukan pencarian	Penelitian ini merancang sebuah aplikasi <i>E-Marketplace</i> Rumah

	Kasus: Kabupaten Karawang) (2021)[14]		informasi rumah kos di Kabupaten Karawang dengan mudah dan dapat membantu pemilik mengurangi biaya pemasaran rumah kosnya	Kost yang berbasis <i>mobile android</i> . Sedangkan penulis mengembangkan sistem informasi rumah kos berbasis <i>website</i> .
5	Sistem Informasi Manajemen Kost Menggunakan <i>Framework Laravel</i> (Studi Kasus Pada Kost Griya Stonen) (2019)[16]	<i>Waterfall</i>	Sistem informasi berbasis web yang dapat diakses oleh calon penyewa kost Griya Stonen untuk menemukan informasi tentang ketersediaan kamar kost, fasilitas kost, dan fitur sewa kamar kost secara <i>online</i> .	Penelitian ini dalam membangun sistem informasinya menggunakan metode pengembangan <i>waterfall</i> . Sedangkan penulis dalam membangun <i>websitenya</i> menggunakan metode pengembangan <i>extreme programming</i>

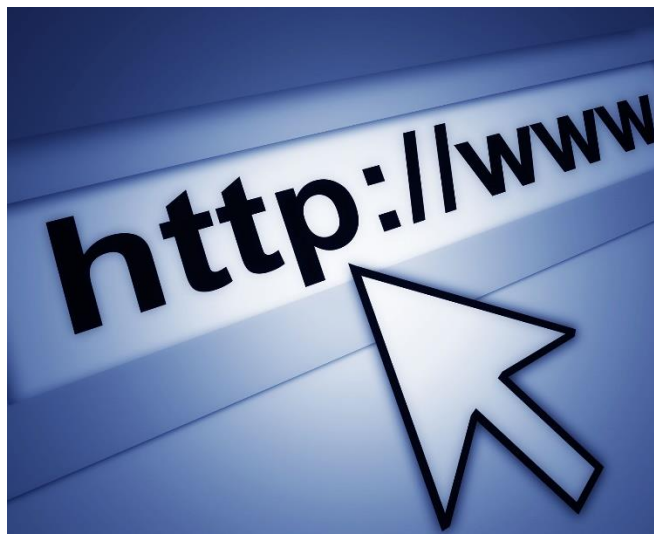
6	Aplikasi <i>Booking</i> Kost Berbasis <i>Android</i> Di Kota Palangka Raya (2019)[18]	<i>Waterfall</i>		Perbedaan penelitian ini dengan penulis terdapat pada metode pengembangannya. Penulis menggunakan metode <i>extreme programming</i> sedangkan pada penelitian ini menggunakan metode pengembangan <i>waterfall</i>
---	---	------------------	--	--

2.2 Landasan Teori

2.2.1 Rancang Bangun

Rancang bangun yaitu proses visualisasi, perencanaan, dan pembuatan gambar atau pengaturan elemen-elemen terpisah menjadi suatu kesatuan yang lengkap dan berfungsi. Jadi, mengubah hasil analisis menjadi perangkat lunak dan membangun atau memperbaiki sistem yang sudah ada dikenal sebagai rancang bangun. [20].

2.2.2 Website



Gambar 2. 1 Website

Istilah *website* merujuk pada sekumpulan halaman yang biasanya tergabung dalam suatu domain atau sub domain di WWW (*World Wide Web*) di internet. Masing – masing halaman yang ada pada web, biasanya ditulis dalam format HTML (*Hyper Text Markup Language*) yang bisa diakses melalui protokol HTTP. Protokol ini mengirimkan data ke *browser* web pengguna dari server. Maka dari situasi ini sebuah *website* dapat berisi berbagai informasi seperti teks, visual yang bergerak atau diam, animasi, suara, video atau kombinasi dari semuanya. Informasi yang disediakan bisa bersifat statis yang berarti tetap dan jarang berubah dengan konten yang hanya berasal dari pemilik *website*. Tetapi bisa juga bersifat dinamis jika informasinya selalu berubah dan interaktif dua arah, yang melibatkan kontribusi dari pemilik atau pengguna *website*. Pada *website* membentuk

struktur yang saling terkait dengan sebuah penghubung melalui jaringan halaman yang diberi nama *hyperlink* agar menciptakan navigasi yang mudah [21].

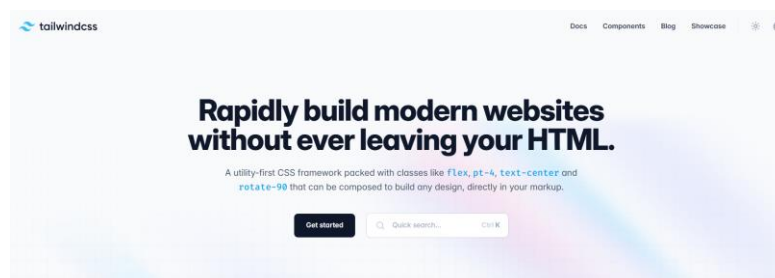
2.2.3 Framework Laravel



Gambar 2. 2 Laravel

Framework Laravel dibuat oleh Taylor Otwell yang mencakup berbagai fitur seperti *bundle*, migrasi, dan *Command Line Interface (CLI)* bernama *artisan*. *Laravel* adalah kerangka pengembangan web MVC (*Model-view-controller*) yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya untuk pengembangan dan perbaikan serta meningkatkan produktivitas melalui sintaks yang bersih dan fungsional yang dapat mempercepat proses implementasi. *Laravel* menggabungkan berbagai aspek fitur terbaik dari kerangka kerja seperti *Codeigniter*, *Yii*, *ASP.NET MVC*, *Ruby on Rails*, *Sinatra*, dan lainnya. Dengan banyaknya fitur tersebut, *Laravel* diakui bisa meningkatkan kecepatan pengembangan *website* secara signifikan [22].

2.2.4 Framework Tailwind CSS



Gambar 2. 3 Tailwind

Framework Tailwind CSS merupakan kerangka kerja yang bersifat utility-first untuk merancang tampilan antarmuka pengguna dengan kecepatan yang tinggi. Pendekatan ini membuat kemudahan dengan secara otomatis menyesuaikan format elemen induk pada halaman, sehingga menyesuaikan dengan cepat terhadap anak elemen yang ada didalamnya. Konsep ini mengakibatkan proses pengembangan tampilan menjadi lebih efisien dan responsif [23].

2.2.5 *MySQL*



Gambar 2. 4 *MySQL*

MySQL merupakan implementasi sistem manajemen basis data relasional (RDBMS) yang bisa digunakan gratis berdasarkan lisensi GPL (*General Public License*). *MySQL* tersedia untuk digunakan oleh siapa pun secara gratis. namun, perlu dicatat bahwa *MySQL* memiliki pembatasan yang mencegah penggunaannya untuk menghasilkan produk turunan yang bersifat komersial. *MySQL* berasal dari ide yang mendasari basis data sebelumnya yaitu SQL (*Structured Query Language*). SQL merupakan konsep dalam operasi basis data yang terlibat dalam pemilihan atau seleksi data dan penyisipan data yang mempermudah pengoperasian data dengan otomatis [24].

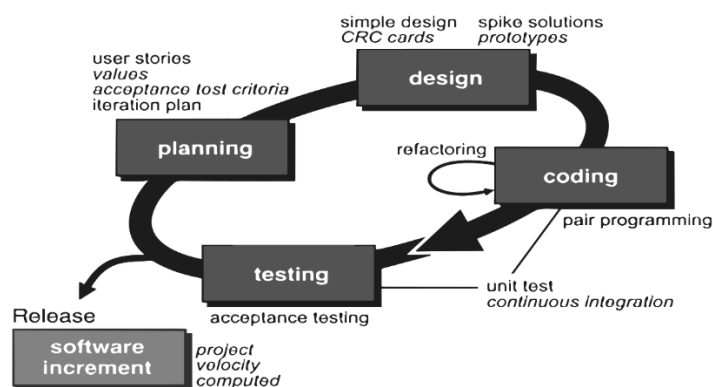
2.2.6 PHP



Gambar 2. 5 PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*, yang merupakan ide dari kepanjangan rekursif bahasa pemrograman server-side yang dirancang untuk pengembangan web. Pencipta dari PHP ini bernama Rasmus Lerdorf pada tahun 1994. PHP bisa dimanfaatkan untuk pengembangan *website* secara gratis dan memiliki sifat *Open Source*, PHP dirilis menggunakan lisensi dari *PHP License* yang sedikit berbeda dengan lisensi GNU yang umumnya digunakan pada proyek *Open Source* [25].

2.2.7 *Extreme programming*



Gambar 2. 6 Metode Extreme Programming

XP atau *extreme programming* merupakan metode pengembangan perangkat lunak yang mengedepankan produktivitas, mudah menyesuaikan dengan perubahan, informalitas, kolaborasi dengan tim dan keterbatasan

dalam penggunaan teknologi di luar pemrograman. Metode ini memiliki beberapa tahapan yang digambarkan pada gambar diatas:

1. Perencanaan (*Planning*): *Planning* berfokus pada memperoleh gambaran fitur dan fungsi dari sistem yang akan dikembangkan. Tahap ini penulis berdiskusi dengan pengembang dan pengguna dengan tujuan untuk mengidentifikasi kebutuhan yang diperlukan dalam mengembangkan sistemnya.
2. Desain (*Design*): Tujuan dari tahap ini adalah untuk mengatur pola logika sistem. Dengan desain sistem yang ideal, ketergantungan antar proses dapat dikurangi. Dengan kata lain, jika salah satu sistem yang rusak, secara keseluruhan sistem tidak terpengaruh. Penulis di tahap ini mulai membuat gambaran sistem dan arsitektur *Unified Modeling Language (UML)*.
3. Pengkodean (*Coding*): Tahap pengkodean merupakan tahap implementasi ke dalam baris koding dari desain sistem yang sudah dibuat di tahap sebelumnya. Di tahap ini bisa langsung dilakukan perulangan pengkodean jika ada perubahan yang dilakukan.
4. Testing (Pengujian): Pada tahapan pengujian dilakukan pengujian pada fitur dan konten yang dibuat. Metode pengujian *Blackbox testing* digunakan untuk melakukan pengujian ini, dengan metode ini dibuatkan tabel testing untuk memastikan bahwa fungsi berjalan dengan benar [26].

2.2.8 UML

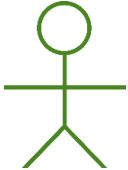
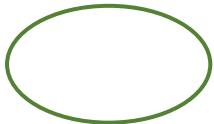



Menurut Windu Gata, Grace (2013), UML atau *Unified Modeling Language* merupakan bahasa spesifikasi standar yang digunakan untuk membuat pemodelan perangkat lunak. UML bukan hanya berfungsi sebagai sebuah metodologi pengembangan sistem, tetapi juga digunakan sebagai alat pendukung pada proses pengembangan sistem. Didalam UML tersedia notasi grafis dan aturan yang konsisten untuk memvisualisasikan struktur dan perilaku dari sistemnya [27].

UML mempunyai beberapa diagram yang biasanya digunakan pada pengembangan sistem perangkat lunak, seperti [28]:

a. *Use Case*

Use case Diagram adalah diagram yang menyimbolkan keterkaitan antar aktor dengan sistem yang akan dibuat. *Use case* diagram menguraikan secara visual bagaimana interaksi antara pengguna atau entitas lain yang berhubungan dengan sistem dan fitur yang diakses. Siapa saja yang berhak memakai fungsi yang dibuat dapat diidentifikasi dengan menggunakan *use case*. Berikut ini adalah simbol yang tersedia pada *use case*.




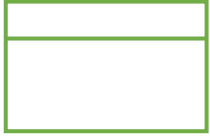
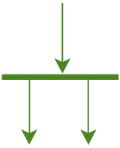
Tabel 2. 2 Tabel *Use Case Diagram*

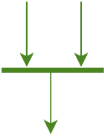

Simbol	Keterangan
	<p>Aktor adalah manusia, proses atau entitas lain yang terlibat dalam hubungan pada sistem yang sedang dibuat. Aktor tidak melulu individu berupa manusia, melainkan dapat berupa entitas atau proses lain tergantung situasinya.</p>
	<p><i>User Case</i> adalah fungsi yang disediakan oleh sistem yang melibatkan interaksi informasi antara aktor dan komponen sistem.</p>
	<p><i>Association</i> adalah interaksi antara aktor dan <i>use case</i> ketika <i>use case</i> memiliki hubungan dengan aktor.</p>
	<p><i>Extend</i> merupakan relasi penunjuk pewarisan dari suatu <i>use case</i> layaknya prinsip <i>inheritance</i> dalam pemrograman berorientasi objek.</p>
	<p><i>Include</i> merupakan relasi yang memungkinkan <i>use case</i> yang ditambahkan selalu dipanggil saat dijalankan dan mengecek apakah sudah dijalankan sebelum melanjutkannya.</p>

b. *Activity Diagram*

Activity Diagram merupakan proses pemberian ilustrasi dari aktivitas suatu sistem, proses bisnis atau menu yang terdapat didalam perangkat lunak. Diagram aktivitas berfokus pada pemberian gambaran dari aktivitas sistem, bukan dari aktivitas aktor. Berikut merupakan simbol yang ada pada *Activity diagram*.

Tabel 2. 3 Tabel *Activity Diagram*



Simbol	Keterangan
	Status awal menunjukkan kondisi awal atau permulaan aktivitas yang akan digambarkan dalam diagram.
	Status Akhir menunjukan kondisi akhir atau titik akhir dari rangkaian aktivitas yang dibuat dalam diagram. Setelah sampai pada status akhir dalam diagram, akan diperoleh pemahaman yang menyeluruh mengenai hasil akhir dari suatu proses dalam sistem.
	<i>Activity</i> menggambarkan bagaimana setiap kelas antarmuka saling berinteraksi satu sama lain dalam sistem. Biasanya aktivitas dimulai dengan kata kerja.
	<i>Swimlanes</i> merupakan simbol dari <i>Activity diagram</i> yang berfungsi untuk pemisah diagram menjadi beberapa jalur yang mewakili peran yang terlibat dalam aktivitas tersebut.
	<i>Fork</i> dalam diagram aktivitas merupakan simbol yang menunjukkan bahwa adanya pembagian satu aktivitas menjadi dua atau lebih aktivitas nantinya dilakukan secara parallel.





	<p><i>Join</i> merupakan simbol untuk menunjukkan kondisi dimana asosiasi penggabungan yang lebih dari satu aktivitas dapat digabungkan untuk mencapai hasil tertentu.</p>
	<p><i>Decision</i>, Ketika terdapat lebih dari satu kondisi aktivitas, <i>decision</i> digunakan untuk menunjukkan keputusan atau tindakan yang harus diambil. Dengan adanya <i>decision</i>, sistem dapat beralih antara jalur aktivitas dalam kondisi tertentu.</p>

c. *Sequence* Diagram

Untuk menyusun *Sequence* diagram, sangat penting memahami objek yang terlibat dalam sebuah *use case* dan metode dari kelas yang digunakan untuk membuat objek tersebut. Perilaku objek dalam suatu *use case* digambarkan dalam *sequence diagram*, menjelaskan rentang waktu objek serta pesan yang dikirim dan diterima di antara mereka. Di bawah ini adalah simbol-simbol yang ditemukan pada diagram *sequence*.

Tabel 2. 4 Tabel *Sequence* Diagram

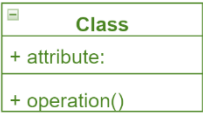
Simbol	Keterangan
	<p>Aktor adalah manusia, proses atau entitas lain yang terlibat dalam hubungan dengan sistem yang sedang dibuat. Aktor tidak melulu individu berupa manusia, melainkan dapat berupa entitas atau proses lain tergantung situasinya.</p>
	<p><i>Object</i> adalah simbol yang merepresentasikan entitas yang terlibat di sistem yang memiliki peran mengirim atau menerima pesan</p>



	<p><i>Lifeline</i> adalah simbol yang mempresentasikan kehidupan suatu objek.</p>
	<p><i>Activation</i> merupakan simbol yang berguna untuk menunjukkan ketika objek dalam keadaan aktif dan terlibat dalam interaksi.</p>
	<p><i>Message</i> merupakan simbol pada <i>Sequence</i> diagram, simbol ini menunjukkan komunikasi antar objek, yang menunjukkan bahwa informasi dikirimkan atau diterima dalam suatu proses.</p>
	<p><i>Return</i> merupakan simbol yang berguna sebagai penanda proses pengembalian nilai dari objek penerima kembali ke objek pengirim setelah suatu aktivitas telah selesai.</p>

d. *Class* Diagram

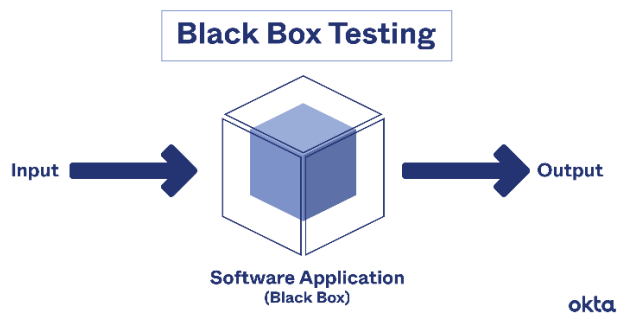
Class Diagram merupakan visualisasi dari struktur sistem yang memperlihatkan kelas yang digunakan untuk membangun sistem beserta hubungan dan propertinya. Berikut merupakan simbol dari *Class* diagram:

Tabel 2. 5 Tabel *Class* Diagram

Simbol	Keterangan
	<p><i>Class</i> merupakan simbol yang berguna untuk menggambarkan kelas yang ada pada sistem. Setiap kelas nantinya memiliki atribut dan metode sebagai karakteristik dan perilaku dari kelasnya.</p>

	<p><i>Association</i> adalah simbol yang berguna sebagai indikasi hubungan antara dua kelas atau lebih pada sistem.</p>
	<p><i>Agregation</i> adalah simbol dalam <i>Class</i> diagram yang berguna untuk menunjukkan hubungan antara kelas – kelas yang mana satu kelas (<i>whole</i>) memiliki objek dari kelas lain (<i>part</i>) tetapi tidak eksklusif.</p>

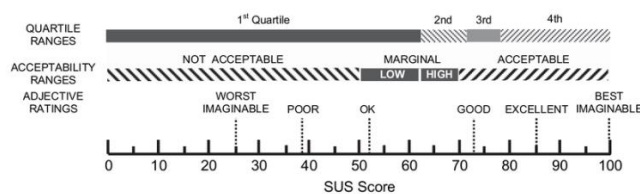
2.2.9 Blackbox testing



Gambar 2. 7 Blackbox Testing

Pengujian *Blackbox* merupakan metode pengujian yang berfokus pada spesifikasi fungsional perangkat lunak. Metode ini memungkinkan pengujian untuk mengidentifikasi kondisi inputan dan menguji kinerja perangkat lunak berdasarkan spesifikasi fungsionalnya [29].

2.2.10 SUS

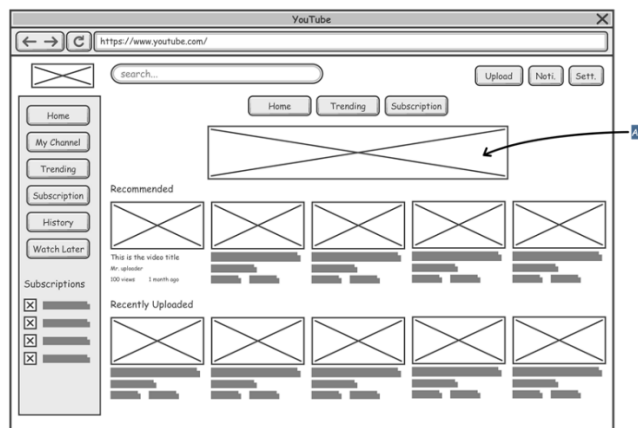


Gambar 2. 8 SUS

System Usability Scale merupakan metode pengukuran standar yang diperkenalkan oleh John Brooke pada tahun 1986 untuk mengevaluasi tingkat kegunaan suatu sistem. Metode ini efisien dan efektif dalam menilai berbagai

jenis produk seperti website dan aplikasi. Dengan menggunakan sampel yang relatif kecil, SUS mampu menghasilkan data evaluasi yang akurat sambil meminimalkan waktu dan biaya yang diperlukan. Nilai-nilai yang dihasilkan dari metode SUS dapat digunakan sebagai tolok ukur untuk menentukan apakah suatu aplikasi layak untuk diimplementasikan [30].

2.2.11 Wireframe



Gambar 2. 9 Wireframe

Wireframe adalah kerangka dasar yang digunakan untuk mengatur elemen-elemen pada halaman aplikasi sebelum memasuki tahap desain yang lebih detail. Proses pembuatan *wireframe* dapat dilakukan menggunakan alat desain seperti draw.io. Secara visual, *wireframe* terdiri dari garis dan kotak yang menentukan posisi berbagai elemen dalam aplikasi. *Wireframing* adalah teknik yang digunakan oleh desainer *UI/UX* untuk merancang struktur website atau aplikasi [31].