

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Enterprise Resource Planning (ERP)***

*Enterprise Resource Planning (ERP)* merupakan sistem yang melakukan integrasi dan otomatisasi sektor-sektor bisnis perusahaan [1]. Integrasi tersebut memberikan kenaikan performa dan mengurangi waktu yang dibutuhkan untuk melakukan proses yang sama.

Sistem *ERP* dalam proses implementasinya memiliki tiga jenis *deployment* yaitu *On Premise ERP*, *Cloud ERP* dan *Hybrid ERP* [2]. *Deployment ERP* merupakan metode perancangan infrastruktur dimana sistem *ERP* di *install* dan dijalankan sehingga perusahaan memiliki opsi memilih tipe *deployment* apa yang mereka butuhkan.

##### **2.1.1 *On Premise ERP***

*On Premise ERP* adalah sistem *ERP* yang menggunakan infrastruktur dari perusahaan itu sendiri [5]. *ERP on premise* memerlukan biaya dan waktu yang banyak karena sistem *ERP* akan dipasang dan dijalankan di server perusahaan milik perusahaan itu sendiri.

##### **2.1.2 *Cloud ERP***

*Cloud ERP* adalah sistem *ERP* dengan model *Software-as-a-Service (SAAS)* [5], yang berarti sistem *ERP* disediakan oleh *provider*, pengguna tidak perlu membuat infrastruktur sendiri karena semua sudah disediakan oleh *provider*, pengguna hanya perlu membayar biaya sistem

ERP secara berkala. Tabel 2.1 merupakan perbandingan antara *on premise* dan *cloud ERP*.

Tabel 2.1 Perbandingan antara *On Premise ERP* dan *Cloud ERP*

<b>Kategori</b>	<b><i>On Premise ERP</i></b>	<b><i>Cloud ERP</i></b>
Biaya	Biaya lebih mahal karena membutuhkan infrastruktur sendiri	Biaya lebih murah karena infrastruktur telah disediakan <i>provider</i>
Kecepatan <i>deployment</i>	Lebih lama	Lebih cepat
Keahlian teknologi	Memperlukan lebih banyak ahli untuk mengatur sistem	Tidak diperlukan ahli karena <i>update</i> dan <i>maintenance</i> dilakukan oleh <i>provider</i> sistem
Kemudahan pemakaian	Lebih susah	Lebih mudah
Skalabilitas	Lebih susah karena perlu menambah infrastruktur baru	Lebih mudah
Fitur dan kustomisasi	Lebih leluasa dalam kustomisasi dan lebih banyak fiturnya	Lebih sedikit dan lebih ketat dalam kustomisasi
Fitur baru dan <i>upgrades</i>	Lebih sulit dan membutuhkan waktu yang lama, terutama sistem yang memiliki banyak kustomisasi	Lebih mudah dan cepat
Kontrol	Lebih terkontrol data dan proses karena memakai infrastruktur sendiri	Kurang karena data dan infrastruktur terdapat di <i>provider</i>

Sumber: [5]

### 2.1.3 *Hybrid ERP*

*Hybrid ERP* merupakan campuran dari *on premise ERP* dan *cloud ERP*, *hybrid ERP* mempunyai *benefit* dari kedua tipe *ERP* tersebut. Secara umum, *hybrid ERP* memakai 2 *tier* fondasi, dimana *tier 1* merupakan *on premise ERP* untuk menampung bagian krusial bisnis dan *tier 2* merupakan *Cloud ERP* sebagai pendukung salah satu atau semua bagian bisnis atau daerah [3].

## 2.2 **Infor CloudSuite**

Infor CloudSuite adalah sistem *ERP* berbasis *Cloud* yang dibuat oleh Infor. Infor CloudSuite menggunakan *Amazon Web Service (AWS)* sebagai infrastruktur untuk sistem *ERP*, Infor CloudSuite dapat dipakai oleh semua tingkat perusahaan mulai dari perusahaan skala kecil sampai perusahaan skala internasional.

## 2.3 **Point of Sale (POS)**

*Point of Sale (POS)* merupakan bagian penting dari titik pembelian, *POS* mengacu pada tempat dimana pelanggan melakukan pembayaran untuk barang atau jasa. Secara sederhana *POS* merupakan perangkat lunak yang melakukan pencatatan terhadap transaksi jual beli yang terjadi di perusahaan [6]

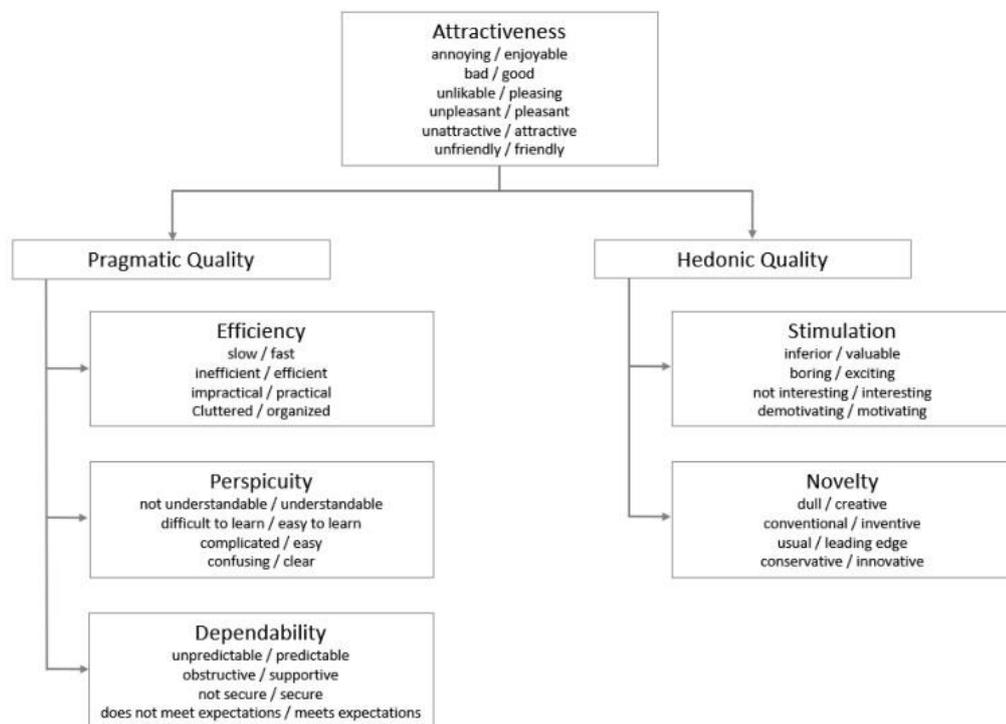
## 2.4 **User Experience (UX)**

*User Experience* adalah persepsi dan tanggapan seseorang yang dihasilkan dari penggunaan atau antisipasi penggunaan suatu produk, sistem, atau layanan. *User Experience* berfokus dengan apa yang diperlukan oleh pengguna sehingga pengguna dapat menggunakan produk secara maksimal [4].

## 2.5 User Experience Questionnaire (UEQ)

User Experience Questionnaire (UEQ) merupakan alat untuk mengukur User Experience suatu sistem/aplikasi/benda, penilaian UEQ di lakukan dengan menggunakan 7 point Likert scale [7].

### 1) Struktur UEQ



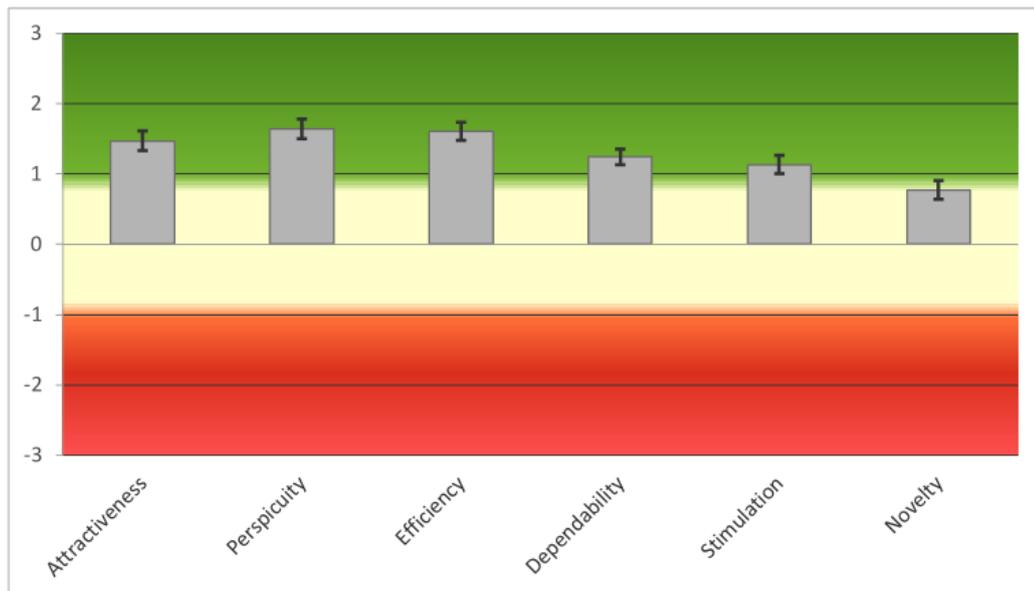
Gambar 2.1 Struktur Skala yang Diasumsikan dari UEQ

Sumber: [7]

Gambar 2.1 merupakan struktur UEQ, UEQ memiliki 6 faktor utama dengan 26 item. Berikut penjelasan dari 6 faktor utama UEQ:

1. *Attractiveness*, impresi secara keseluruhan mengenai objek.
2. *Perspicuity*, apakah mudah untuk terbiasa dengan objek?
3. *Efficiency*, apakah pengguna dapat menyelesaikan tugasnya tanpa memerlukan energi yang lebih?
4. *Dependability*, apakah pengguna merasa seperti di dalam kontrol? Dapatkah pengguna memprediksi perilaku sistem?

5. *Stimulation*, apakah pengguna merasa bersemangat ketika menggunakan produk/objek?
6. *Novelty*, apakah produk ini inovatif atau kreatif?



Gambar 2.2 Contoh Produk dengan Hasil yang Baik

Sumber: [7]

Gambar 2.2 merupakan grafik contoh produk dengan hasil yang baik dari *UEQ*. Grafik ini merupakan contoh hasil penilaian suatu produk menggunakan *User Experience Questionnaire (UEQ)*. Standar rentang skala yang digunakan untuk menilai setiap aspek *UEQ* adalah rentang nilai antara -0,8 dan 0,8, nilai  $< -0.8$  menandakan hasil yang negatif dan nilai  $> 0.8$  menandakan hasil yang positif [7].

Rentang skala pada gambar diatas adalah antara -3 (sangat buruk) dan +3 (sangat baik). Dalam penilaian pada umumnya hanya nilai dalam rentang terbatas saja yang akan diamati, karena dari perhitungan rata-rata dari orang yang berbeda dengan penilaian yang berbeda dan kecenderungan menjawab (misalnya menghindari kategori jawaban yang ekstrim) sangat tidak mungkin untuk mengamati nilai di atas +2 atau di bawah -2 [7].

## 2) Rumus Penilaian *UEQ*

$$X = \frac{\Sigma x \text{ person}}{\Sigma \text{ item}} \quad (2.1)$$

Rumus 2.1 Rumus Skor *Item*

Rumus 2.1 merupakan rumus untuk menghitung skor setiap item pada *UEQ*, berikut penjelasan rumus:

- X : rata-rata skala item per orang  
 $\Sigma x \text{ person}$  : total nilai per item  
 $\Sigma \text{ item}$  : jumlah per skala item

$$X = \frac{\Sigma x \text{ skala}}{\Sigma \text{ item}} \quad (2.2)$$

Rumus 2.2 Rumus Hasil Akhir

Rumus 2.2 merupakan rumus untuk menghitung hasil akhir pada *UEQ*, berikut penjelasan rumus:

- X : rata-rata skala  
 $\Sigma x \text{ skala}$  : total nilai per skala  
 $\Sigma \text{ item}$  : jumlah per skala item

## 2.6 *User Interface (UI)*

*User Interface* merupakan bagian yang menjadi tempat berinteraksi antara sistem dengan *user*. Menurut Wilbert O. Galitz, *user interface* diartikan sebagai bagian dari komputer dan *software* yang dapat berinteraksi (dilihat, didengar, disentuh) dan dapat dimengerti oleh manusia [8]. Dalam pembuatan prototipe terdapat alat yang digunakan untuk membuat desain UI, salah satunya adalah Figma.

### 2.6.1 Figma

Figma merupakan *platform* gratis untuk membuat desain dan prototipe suatu sistem. Figma beroperasi secara *browser-based*

yang artinya dapat digunakan langsung secara *online* menggunakan *browser* dan melalui aplikasi Figma yang perlu instalasi melalui sistem operasi Windows dan macOS.

### **2.7 Responsive Web Design (RWD)**

*Responsive Web Design (RWD)* merupakan bagian dari *website development* yang membuat perubahan dinamis dari penampilan *website* berdasarkan ukuran layar dan orientasi layar [9]. Dalam penerapan *RWD*, *website* dirancang dengan menambahkan beberapa desain khusus berdasarkan ukuran *device* sehingga terdapat lebih dari satu desain dalam satu halaman *website* yang sama.

### **2.8 Chatbot**

*Chatbot* merupakan perangkat lunak yang dapat melakukan konversasi via *text* yang memberikan respon berdasarkan interaksi dari penggunaannya [10]. Dalam penggunaannya *Chatbot* memiliki berbagai fungsi seperti menyediakan informasi dan mengumpulkan informasi berdasarkan *input* dari penggunaannya.

### **2.9 Website**

*Website* merupakan koleksi dari halaman *website* yang tersambung yang berbagi *unique domain name* [11]. Dalam development *website* terdapat banyak bahasa pemrograman untuk membuat suatu *website*, bahasa pemrograman dasar yang sering dipakai untuk membuat *website* adalah HTML, CSS, PHP, MYSQL, dan Javascript.

### **2.9.1 *HyperText Markup Language (HTML)***

*HyperText Markup Language (HTML)* adalah bahasa pemrograman *website* paling dasar, HTML mendefinisikan arti dan struktur dari konten suatu *website* [11].

### **2.9.2 *Cascading Style Sheets (CSS)***

*Cascading Style Sheets (CSS)* adalah bahasa yang digunakan untuk mendefinisikan tampilan dari dokumen yang ditulis menggunakan HTML dan *Extensible Markup Language (XML)* [11].

### **2.9.3 *PHP: Hypertext Preprocessor (PHP)***

*PHP: Hypertext Preprocessor (PHP)* adalah bahasa pemrograman *open source* yang digunakan dalam *development* web dan dapat di embedded ke *HTML* [12].

### **2.9.4 *MySQL***

MySQL adalah sistem manajemen SQL *database open source* yang dibuat oleh Oracle Corporation [13]. MySQL digunakan untuk mengatur data dalam *database* seperti menambahkan, membaca, melakukan *update*, dan menghapus data.

### **2.9.5 *phpMyAdmin***

phpMyAdmin adalah *software* gratis yang dibuat dengan menggunakan PHP, phpMyAdmin merupakan alat yang berfungsi mengatur administrasi MySQL dalam suatu *website*.

### 2.9.6 XAMPP

XAMPP merupakan kepanjangan dari *cross-platform*, Apache, MySQL, *PHP*, and Perl, XAMPP merupakan *software* distribusi yang menyediakan Apache *web server*, MySQL *database*, *PHP*, dan Perl. XAMPP memungkinkan penggunaanya untuk membangun suatu *local web server* di komputer.

### 2.9.7 JavaScript

JavaScript adalah bahasa pemrograman yang memungkinkan penggunaanya untuk melakukan implementasi fitur rumit pada *web pages* seperti animasi dan *web page* interaktif [11].

### 2.9.8 jQuery

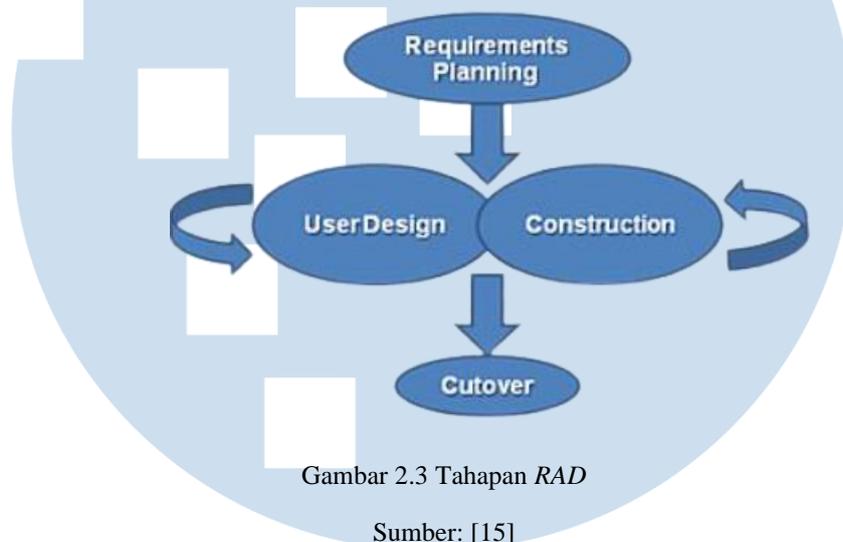
jQuery merupakan Javascript *library* yang membuat manipulasi *event handling*, *animation*, dan *Asynchronous JavaScript and XML (AJAX)* menjadi lebih sederhana dengan *easy-to-use Application Programming Interface (API)* yang dapat bekerja di banyak *browser* [14].

### 2.9.9 Asynchronous JavaScript and XML (AJAX)

*Asynchronous JavaScript and XML (AJAX)* adalah istilah yang diciptakan pada tahun 2005 oleh Jesse James Garrett, *AJAX* menjelaskan pendekatan baru dalam menggunakan bahasa pemrograman secara bersamaan, ketika bahasa pemrograman digunakan secara bersamaan dengan menggunakan *AJAX model*, aplikasi web dapat membuat update yang cepat ke *user interface* tanpa perlu melakukan *reload* pada browser, hal ini membuat aplikasi *web* lebih cepat dan responsif terhadap interaksi *user* [14].

## 2.10 *Rapid Application Development (RAD)*

*Rapid Application Development (RAD)* merupakan pembangunan siklus yang dirancang untuk memberikan pengembangan yang jauh lebih cepat dan hasil yang lebih berkualitas tinggi daripada yang dicapai dengan siklus tradisional [15]. Model ini cocok digunakan untuk pembangunan sistem dengan skala kecil sampai menengah.



Gambar 2.3 Tahapan RAD

Sumber: [15]

Gambar 2.3 merupakan tahapan *Rapid Application Development (RAD)*, terdapat 4 tahap yaitu:

### 1) *Requirements Planning*

*Requirements Planning* adalah proses identifikasi tujuan dari sistem dan kebutuhan informasi untuk perancangan sistem. Proses identifikasi dilakukan dengan pendekatan dengan *user* untuk mengetahui poin-poin penting yang dapat digunakan untuk proses perancangan sistem.

### 2) *User Design*

*User Design* adalah proses perancangan organisasi sistem secara umum yaitu dengan membuat model dan prototipe yang merepresentasikan semua proses pada sistem, *input*, dan *output*. Pada proses ini, perancangan sistem dilakukan dengan membuat *Unified Modeling Language (UML)* dari struktur dan fungsi sistem.

### 3) *Construction*

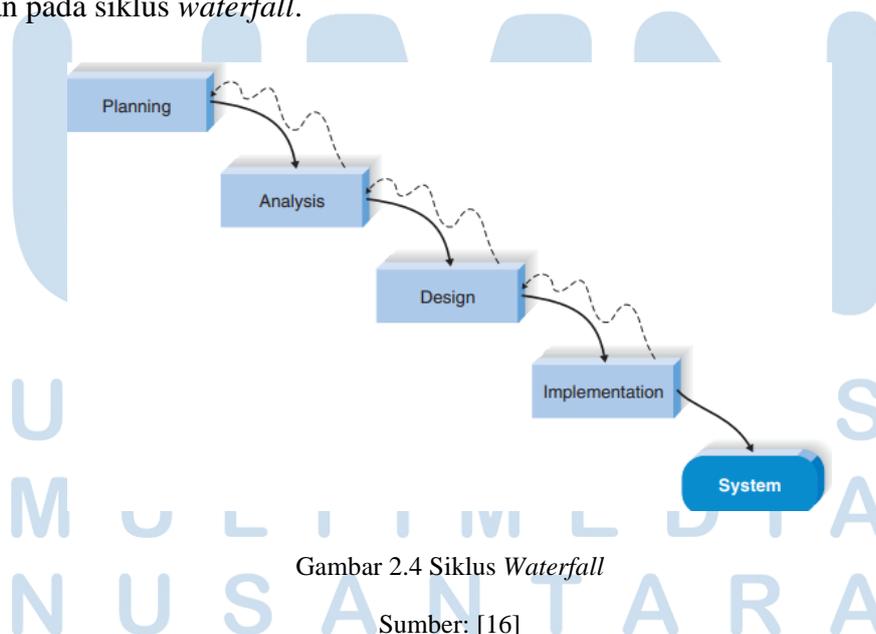
*Construction* adalah proses pembangunan sistem yang telah di rancang, *user* dapat berpartisipasi dalam tahap ini dengan memberikan saran dalam pengembangan sistem. Pada tahap ini *programmer* akan membangun sistem yang telah di *design* sampai selesai.

### 4) *Cutover*

*Cutover* adalah tahap terakhir dari *RAD*, tahap ini berisi kegiatan seperti *testing*, *data conversion*, dan *user training*. Dibandingkan dengan metode tradisional, seluruh proses dikompres sehingga sistem baru dibuat, disalurkan, dan dioperasikan lebih cepat [15].

## 2.11 *Waterfall Development*

*Waterfall* merupakan salah satu siklus pengembangan sistem tradisional. Dalam penerapannya, setiap fase pada *waterfall* biasanya memiliki dokumentasi yang besar dan dipresentasikan ke komite *approval* dan *project sponsor* untuk *approval* sembari proyek maju dari fase ke fase [16]. Gambar 2.4 merupakan tahapan pada siklus *waterfall*.



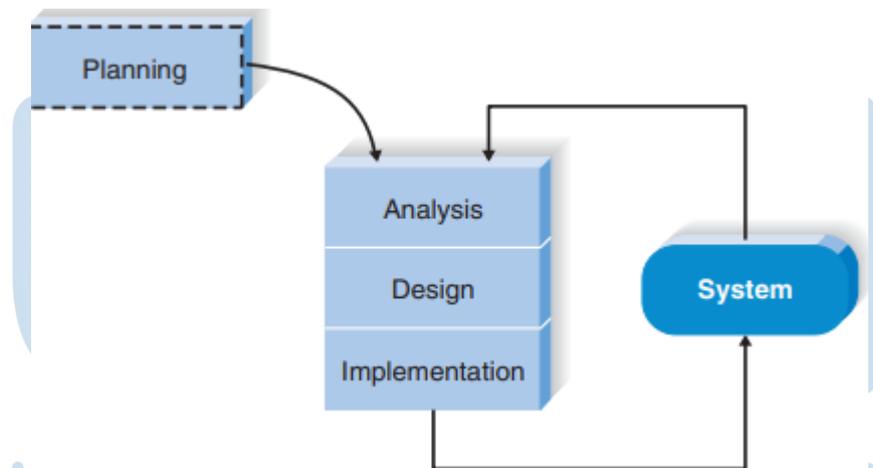
Gambar 2.4 Siklus *Waterfall*

Sumber: [16]

1. *Planning*: merencanakan dan identifikasi kebutuhan proyek.
2. *Analysis*: membuat model, skema, dan aturan sistem.
3. *Design*: merancang arsitektur sistem.
4. *Implementation*: rancang bangun sistem dan *testing*.
5. *System*: sistem telah selesai dibangun dan siap digunakan.

## 2.12 Agile Development

*Agile Development* merupakan *programming-centric* metodologi, dimana *modeling* dan *documentation overhead* dihilangkan [16]. Metode ini dirancang untuk memotong waktu pengembangan sistem. Perbedaan dari metode *agile* dan *waterfall*, metode *agile* memiliki metode *development* yang *iterative* yaitu setiap iterasi adalah proyek *software* yang komplit, termasuk *planning*, *analysis*, *requirement analysis*, *design*, *coding*, *testing*, dan *documentation* [16]. Gambar 2.5 merupakan tahapan model *Agile*.



Gambar 2.5 Siklus *Agile*

Sumber: [16]

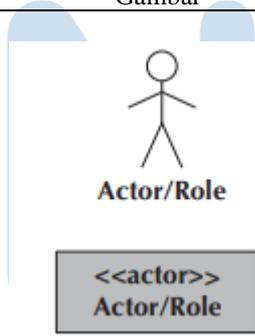
## 2.13 Unified Modeling Language (UML)

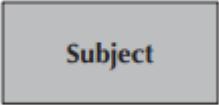
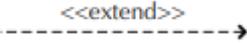
Unified Modeling Language (UML) merupakan bahasa yang digunakan untuk menyediakan bahasa desain yang stabil dan umum yang digunakan untuk membangun model sistem apapun mulai dari proses analisis sampai implemetasi [17]. UML digunakan untuk melakukan desain sistem dari informasi yang didapatkan dari *requirement planning*, berikut adalah diagram-diagram yang terdapat pada UML:

### 2.13.1 Use Case Diagram

Use Case diagram merupakan diagram yang digunakan untuk mengetahui fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang dapat menggunakan fitur tersebut. Tabel 2.2 merupakan penjelasan elemen-elemen yang terdapat pada *use case* diagram.

Tabel 2.2 Elemen – Elemen pada Use Case Diagram

Nama	Penjelasan	Gambar
<i>Actor</i>	<ol style="list-style-type: none"> <li>1. Orang/Sistem yang mendapat <i>benefit</i> dan berasal dari luar subjek.</li> <li>2. Digambarkan dengan gambar orang berbentuk stik atau jika aktor merupakan sistem maka digambarkan kotak dengan label &lt;&lt;actor&gt;&gt; di dalamnya.</li> <li>3. <i>Actor</i> ditandai dengan label berdasarkan <i>role</i>.</li> </ol>	
<i>Use Case</i>	<i>Use Case</i> adalah bagian utama dari fungsi sistem, sebuah <i>use case</i> dapat melakukan <i>extend use case</i> lain dan <i>include use case</i> lain.	
<i>Subject</i>	Mewakili <i>scope</i> dari subjek yaitu proses bisnis suatu subjek.	

		
<i>Association</i>	Menghubungkan <i>actor</i> dengan <i>use case</i> .	
<i>Include</i>	<ol style="list-style-type: none"> <li>Menggambarkan inklusi suatu fungsionalitas dari satu <i>use case</i> dengan <i>use case</i> lainnya.</li> <li>Mempunyai tanda anak panah yang digambarkan dari dasar <i>use case</i> ke <i>use case</i> yang digunakan.</li> </ol>	
<i>Extend</i>	<ol style="list-style-type: none"> <li>Menggambarkan perpanjangan dari <i>use case</i>.</li> <li>Mempunyai anak panah yang digambarkan dari perpanjangan <i>use case</i> ke <i>use case</i> yang digunakan.</li> </ol>	
<i>Generalization</i>	<ol style="list-style-type: none"> <li>Menggambarkan sebuah <i>specialized use case</i> ke <i>use case</i> yang lebih general.</li> <li>Mempunyai anak panah yang digambarkan dari <i>specialized use case</i> ke <i>base use case</i>.</li> </ol>	

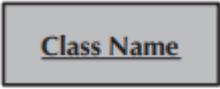
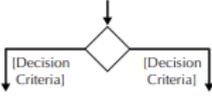
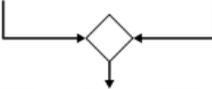
Sumber: [17]

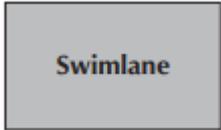
### 2.13.2 Activity Diagram

*Activity* diagram menggambarkan aliran proses yang sistem lakukan, diagram ini hanya menggambarkan apa yang sistem lakukan bukan apa yang dilakukan oleh aktor. Tabel 2.3 merupakan penjelasan elemen-elemen yang terdapat pada *activity* diagram.

Tabel 2.3 Elemen-Elemen pada *Activity* Diagram

Nama	Penjelasan	Gambar
<i>Action</i>	Menggambarkan suatu aksi.	
<i>Activity</i>	Mewakili suatu kumpulan dari aksi.	

		
<i>Object Node</i>	Menggambarkan objek yang terkoneksi.	
<i>Control Flow</i>	Menggambarkan <i>sequence</i> dari eksekusi.	
<i>Object Flow</i>	Menggambarkan aliran dari satu objek dari suatu aksi ke aksi yang lain.	
<i>Initial Node</i>	Menggambarkan awal dari suatu aktivitas atau aksi.	
<i>Final-Activity Node</i>	Menggambarkan akhir dari suatu aktivitas atau aksi.	
<i>Final-Flow Node</i>	Digunakan untuk menghentikan aliran objek.	
<i>Decision Node</i>	Menggambarkan sebuah tes kondisi yang memastikan alur hanya mengalir ke satu arah.	
<i>Merge Node</i>	Digunakan untuk menggabungkan kembali <i>decision path</i> yang berbeda.	
<i>Fork Node</i>	Digunakan untuk memisahkan <i>behaviour</i> ke kumpulan parallel aksi atau aktivitas.	
<i>Join Node</i>	Digunakan untuk menggabungkan kembali <i>fork node</i> .	

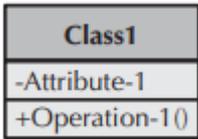
		
<i>Swimlane</i>	Digunakan untuk memisahkan <i>activity</i> diagram menjadi <i>rows</i> dan <i>columns</i> untuk menetapkan aktivitas individu atau objek yang bertanggung jawab melaksanakan aktivitas.	

Sumber: [17]

### 2.13.3 Class Diagram

*Class* diagram menggambarkan struktur sistem beserta relasi kelas-kelas yang akan dibuat dalam membangun sistem, setiap kelas memiliki atribut dan metode. Tabel 2.4 merupakan penjelasan elemen-elemen yang terdapat pada *class* diagram.

Tabel 2.4 Elemen-Elemen pada *Class* Diagram

Nama	Penjelasan	Gambar
<i>Class</i>	<ol style="list-style-type: none"> <li>1. Mewakili orang, tempat, atau sesuatu yang sistem perlukan untuk menampung informasi.</li> <li>2. Terdapat daftar atribut di bagian tengah.</li> <li>3. Terdapat daftar operasi di bagian bawah.</li> </ol>	
<i>Attribute</i>	Menggambarkan properti yang menjelaskan situasi dari suatu objek.	attribute name /derived attribute name
<i>Operation</i>	Menggambarkan fungsi atau aksi dari suatu kelas yang dapat dilaksanakan.	operation name ()
<i>Association</i>	Menggambarkan relasi antara kelas-kelas atau suatu kelas dan dirinya sendiri.	AssociatedWith 0..*      1

<i>Generalization</i>	Menggambarkan <i>a-kind-of relationship</i> antara kelas-kelas.	
<i>Aggregation</i>	Menggambarkan suatu <i>logical a-part-of relationship</i> antara kelas-kelas atau suatu kelas dan dirinya sendiri.	
<i>Composition</i>	Menggambarkan suatu <i>physical a-part-of relationship</i> antara kelas-kelas atau kelas dan dirinya sendiri.	

Sumber: [17]

## 2.14 User Acceptance Testing (UAT)

*User Acceptance Testing (UAT)* terdiri dari proses untuk verifikasi bahwa solusi yang telah dibuat berfungsi untuk *user* [18]. *UAT* merupakan salah satu metode *testing* yang dipakai untuk menentukan apakah sistem yang dibuat sudah sesuai atau tidak berdasarkan *feedback* dari *user*, dokumen *UAT* berisi *use case* beserta proses dan *output* dalam suatu fungsi dari sistem.

## 2.15 Penelitian Terdahulu

Tabel 2.5 Tabel Penelitian Terdahulu

Judul, Peneliti, dan Tanggal	Kesimpulan
Judul: Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching  Peneliti: 1. David Domarco	Hasil penelitian menunjukkan bahwa aplikasi <i>chatbot</i> berhasil memenuhi standar pengambilan informasi dengan hasil yang sangat baik, nilai presisi 72% dan 100% recall menunjukkan rata-rata nilai harmonik sebesar 83,7%. Sebagai penerapan hedonic, <i>chatbot</i> sudah mempengaruhi niat

<p>2. Ni Made Satvika Iswari</p> <p>Tanggal: berdasarkan <a href="http://ejournals.umn.ac.id">ejournals.umn.ac.id</a> penelitian ini dilakukan pada tahun 2017</p>	<p>pemakaian aplikasi sebesar 83% dan <i>immersion</i> sebesar 82%.</p>
<p>Judul: Usability Measurement of Mobile Applications with System Usability</p> <p>Peneliti: 1. Aycan Kaya 2. Reha Ozturk 3. Cigdem Altin Gumussoy</p> <p>Tanggal: berdasarkan <a href="http://semanticsscholar.org">semanticsscholar.org</a>, penelitian ini dilakukan pada bulan Januari 2019</p>	<p>Hasil penelitian menunjukkan bahwa <i>usability</i> semua aplikasi memuaskan dan di atas standar. Perbandingan aplikasi seluler satu sama lain menunjukkan bahwa WhatsApp memiliki skor kegunaan tertinggi, sedangkan Facebook memiliki skor terendah. Selain itu, menurut hasil, tidak ada perbedaan yang signifikan antara sistem operasi dalam hal kegunaan aplikasi seluler.</p>
<p>Judul: Assessing User Experience of the Online Petition System in Indonesia based on UEQ</p> <p>Peneliti: 1. Ika Nurlaili Isnainiyah 2. Bambang Triwahyono</p> <p>Tanggal: berdasarkan <a href="http://ejurnal.stmik-budidarma.ac.id">ejurnal.stmik-budidarma.ac.id</a>, penelitian ini dilakukan pada bulan Agustus 2021</p>	<p>Berdasarkan hasil skoring 26 <i>item</i> dari <i>UEQ</i>, skala <i>dependability</i> dan kebaruan mendapat skor paling rendah di antara skala lainnya. <a href="http://Petitionline.com">Petitionline.com</a> perlu meningkatkan aspek terkait kedua skala untuk meningkatkan perannya dalam memfasilitasi budaya partisipatif.</p>
<p>Judul: Evaluating User Experience of English Learning Interface using User Experience Questionnaire and System Usability Scale</p> <p>Peneliti: 1. Ni Putu Indah Rosita Devy 2. Sunu Wibirama 3. Paulus Insap Santosa</p> <p>Tanggal: berdasarkan <a href="http://ieeexplore.ieee.org">ieeexplore.ieee.org</a>,</p>	<p>Hasil dari penelitian ini adalah Kuesioner Pengalaman Pengguna (<i>UEQ</i>) dan Sistem <i>Usability Scale</i> (<i>SUS</i>) menunjukkan bahwa multimedia tanpa isyarat lebih baik daripada multimedia dengan isyarat dan multimedia dengan hanya teks dalam hal pengalaman pengguna dan kegunaan.</p>

penelitian ini dilakukan pada bulan November 2017	
Judul: Analysis and Design of Web-Based Information System for Church Congregations Case Study: Church BNKP Pewarta	Dalam hasil <i>testing</i> fungsional menggunakan <i>black box testing</i> . Hasil menunjukkan bahwa website sistem informasi yang telah dibuat dapat menyimpan data jemaat Gereja dan menyajikan informasi umum Gereja sesuai kebutuhan.
Peneliti: 1. Jansen Wiratama 2. Ririn Ikana Desanti	
Tanggal: berdasarkan <a href="http://ejournals.umn.ac.id">ejournals.umn.ac.id</a> penelitian ini dilakukan pada tahun 2022	

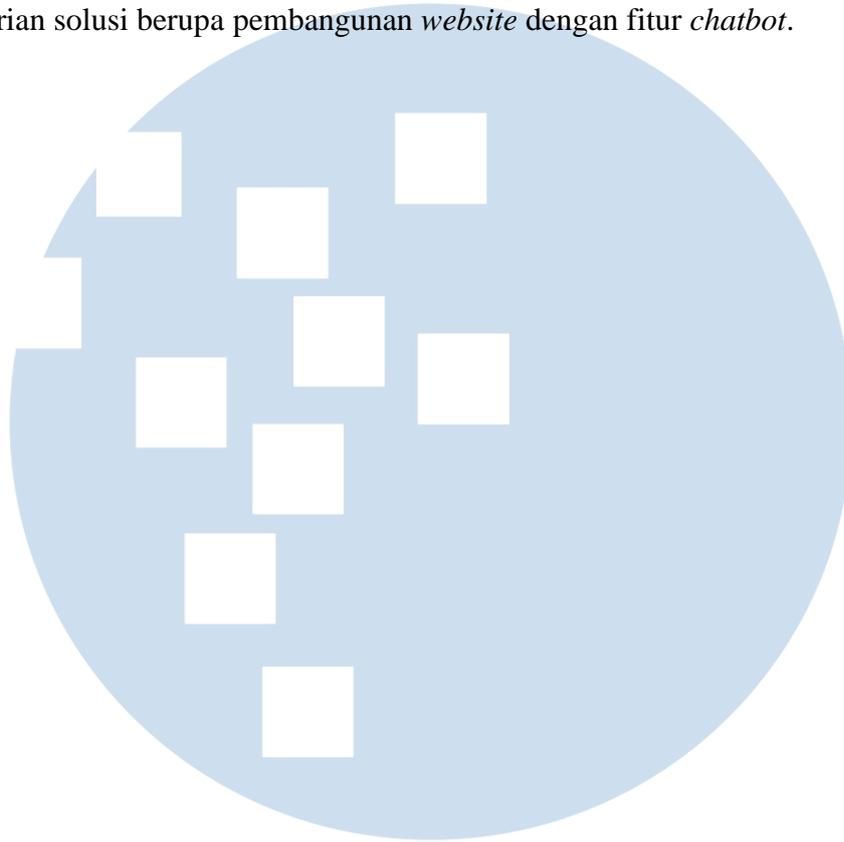
Berdasarkan tabel 2.5 yaitu tabel penelitian terdahulu, penelitian 1 meneliti pembuatan *Chatbot* untuk pencarian informasi yang berkaitan dengan *anime*, *Chatbot* dibuat dengan menggunakan *pseudocode*, *Asynchronous JavaScript And XML (AJAX)*, dan database. Penelitian 2 meneliti *User Experience* dari 4 *mobile application* bernama WhatsApp, Facebook, YouTube, dan Mail, Penelitian dilakukan dengan 1 metode yaitu *System Usability Scale (USU)*.

Penelitian 3 meneliti *User Experience* dari *website* petisi online bernama *petitiononline.com*, penelitian dilakukan dengan 1 metode yaitu *User Experience Questionary (UEQ)*. Penelitian 4 meneliti *User Experience* dari *software* belajar bahasa Inggris bernama *multimedia*, penelitian dilakukan dengan 2 metode yaitu *User Experience Questionary (UEQ)* dan *System Usability Scale (USU)*.

Penelitian 5 meneliti analisis dan perancangan *website* sistem informasi jemaat Gereja, penelitian dilakukan dengan menggunakan metode *Web Improvement Life Cycle (WDLC)*, *Unified Modelling Language (UML)*, dan *Black Box testing*.

Penelitian ini mempunyai perbedaan dengan kelima penelitian sebelumnya karena pada penelitian ini terdapat evaluasi *UX* dengan menggunakan metode

*UEQ*. Dalam penelitian ini, hasil dari evaluasi akan dijadikan acuan sebagai pemberian solusi berupa pembangunan *website* dengan fitur *chatbot*.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA