

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Informasi

Sistem Informasi mengumpulkan, memproses, menyimpan, menganalisa, dan menyebarkan suatu informasi untuk tujuan tertentu[7].

Sistem informasi melibatkan teknologi informasi seperti komputer, software, database, sistem komunikasi, internet, gawai, dll, untuk melakukan suatu tugas tertentu, berinteraksi dan menginformasi berbagai aktor dalam organisasi atau konteks sosial[10].

Maka dapat disimpulkan bahwa sistem informasi menggunakan komponen teknologi informasi yang saling terintegrasi untuk mengumpulkan, menyimpan, mengakses, memproses, dan memberikan data untuk dan dengan tujuan tertentu yang dapat memberikan manfaat tertentu kepada suatu organisasi atau kalangan tertentu.

#### 2.2 Enterprise Resource Planning

Sistem enterprise resource planning (ERP) adalah suatu sistem yang mengintegrasikan berbagai proses bisnis dan fungsi pada organisasi menjadi satu sistem[11].

Sistem ERP berkembang dari yang tadinya berupa Material Requirement Planning (MRP) dan kemudian menjadi (MRP II). MRP lahir sekitar tahun 1970 MRP awalnya hanya memiliki fitur production planning dan inventory control. Satu decade kemudian MRP berkembang menjadi MRP II dan mencakup financial accounting, sales planning dan customer order processing. Pada tahun 1990an sistem ERP di bangun dengan target untuk mengintegrasikan business *function* utama dan menselaraskan business process dengan sistem software ERP[12].

Dapat disimpulkan bahwa sistem Enterprise Resource Planning (ERP) membantu perusahaan dalam mengintegrasikan, mengotomatisasi, serta mengkoordinasi function areas dan business process dari suatu organisasi atau perusahaan.

### **2.3 User Experience**

Menurut Jesse James Garrett, *User Experience* adalah suatu pengalaman yang dibuat oleh suatu produk saat digunakan oleh penggunanya[9]. *User experience* bukan tentang bagaimana program tersebut bekerja, tetapi bagaimana program tersebut berinteraksi dengan penggunanya. UX meliputi semua perasaan, pemikiran, sensasi, dan aksi yang dilakukan pada suatu aktifitas[13]. UX harus didesain supaya sistem menjadi *accessible*, *usable*, dan *engaging*. Selain itu yang menjadi perhatian dalam pengembangan sistem yang interaktif, produk, ataupun suatu layanan yang sesuai dengan kehidupan manusia merupakan tugas dari UX.

### **2.4 User Experience Questionnaire (UEQ)**

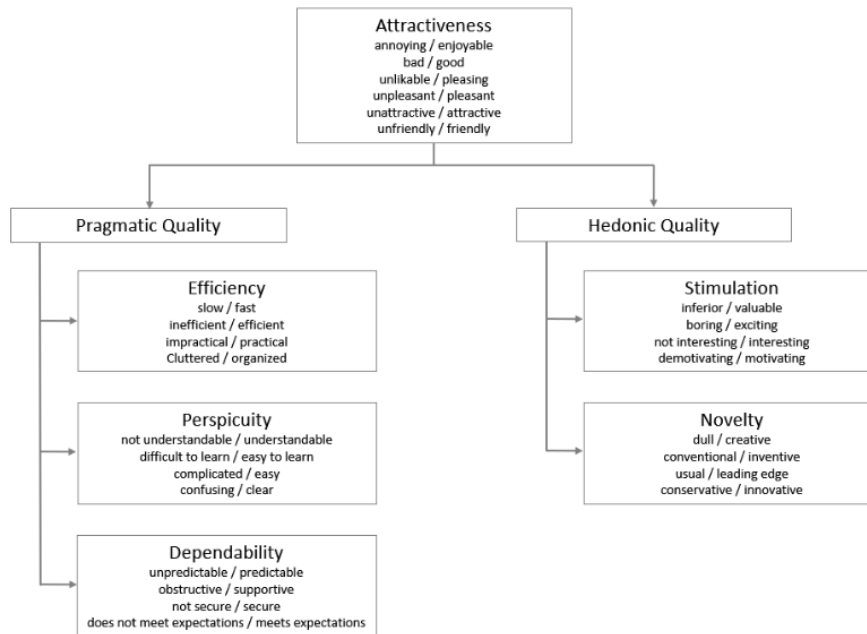
*UEQ* merupakan suatu kuisioner yang banyak digunakan untuk meneliti atau mengukur *User experience* secara subjektif terhadap suatu produk atau sistem oleh pengguna. *UEQ* memiliki 6 aspek pengukuran utama yang memiliki 26 penilaian dalam *UEQ*.

6 aspek dalam *UEQ* yaitu:

- *Attractiveness*: kesan dari aplikasi secara keseluruhan. Apakah pengguna menyukainya?
- *Perspicuity*: apakah aplikasi mudah dimengerti?
- *Efficiency*: apakah pengguna mampu menyelesaikan tugasnya dengan menggunakan aplikasi tanpa masalah?
- *Dependability*: apakah user merasa mengendalikan dalam interaksi dengan aplikasi?

- *Stimulation*: apakah menyenangkan menggunakan aplikasi tersebut?
- *Novelty*: apakah aplikasi yang digunakan bersifat inovatif dan kreatif.

Pada *Gambar 2.1 Attractiveness* memiliki dimensi sendiri. Sedangkan *Perspiciuity*, *Efficiency*, *Dependability* merupakan aspek dari *pragmatic quality* yang merupakan *goal-oriented aspect*. *Stimulation* dan *Novelty* merupakan bagian dari aspek *hedonic-quality* yang bukan merupakan *goal-oriented aspect*.

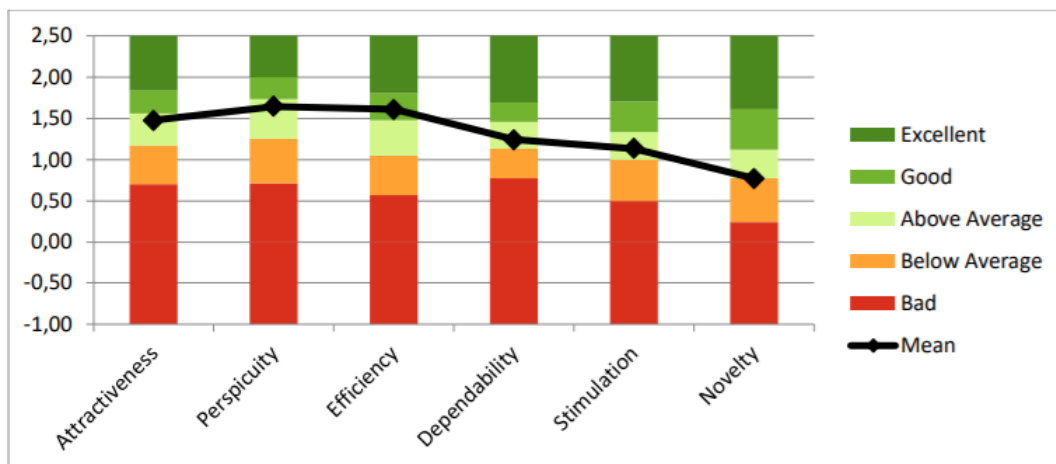


*Gambar 2.1 dimensi UEQ*

Sumber:[14]

UEQ memiliki standar penilaian yang dibagi kedalam 5 kategori per aspek yaitu *Excellent*, *Good*, *Above Average*, *Below Average*, dan *Bad*. Pada *Gambar 3* dapat dilihat bahwa setiap aspek memiliki nilai *mean* yang berbeda beda.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.2 Grafik Benchmark UEQ

Sumber:[14]

Penilaian dengan UEQ akan dianggap positif apabila nilai dari suatu aspek lebih memiliki nilai  $> 0.8$ . Jika nilai berada pada angka  $-0.8$  sampai  $0.8$  maka akan dianggap sebagai nilai netral. Apabila nilai  $< -0.8$  maka aspek tersebut akan mendapat nilai negatif. Gambar 2.2 menunjukkan penilaian suatu produk menggunakan UEQ dengan nilai positif.

## 2.5 Kuisisioner

*Questionnaire* atau kuisisioner dalam KBBI (Kamus Besar Bahasa Indonesia) memiliki arti alat riset atau survei yang terdiri atas serangkaian pertanyaan tertulis, bertujuan mendapatkan tanggapan dari kelompok orang terpilih melalui wawancara pribadi atau melalui pos; daftar pertanyaan. Kuisisioner adalah salah satu teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan secara tertulis kepada responden untuk dijawabnya[15].

Kuisisioner adalah suatu metode atau alat dalam riset yang dilakukan dengan cara memberikan serangkaian pertanyaan secara tertulis kepada responden yang telah dipilih sebelumnya sebagai narasumber atau responden pada penelitian.

## 2.6 Odoo Suite

*Odoo ERP suite* merupakan aplikasi *ERP open-source* yang menyediakan solusi untuk perusahaan seperti modul *CRM, eCommerce, accounting, inventory, PoS, project management*, dll. *Odoo ERP* merupakan sistem ERP yang berbentuk

*website*. Sistem ERP ini dapat diakses selama penggunanya memiliki perangkat yang dapat terkoneksi ke internet. Selain itu *Odoo Community Edition* merupakan sistem *ERP* *Odoo* yang mudah untuk dilakukan kustomisasi dan mudah dijangkau. Sistem *Odoo* sendiri merupakan sistem dengan 3 komponen utama yaitu: *database layer* yang berbasis *Postgresql*, *Odoo application server*, dan *client layer*[16].

## **2.7 Python**

*Python* merupakan suatu Bahasa pemrograman yang terintrepetasi, *object-oriented*, dan Bahasa pemrograman tingkat tinggi dengan semantik yang dinamis[17]. *Python* dapat digunakan untuk banyak hal, beberapa contohnya adalah pengembangan *web*, *analysis data*, *machine learning*, dan masih banyak lagi.

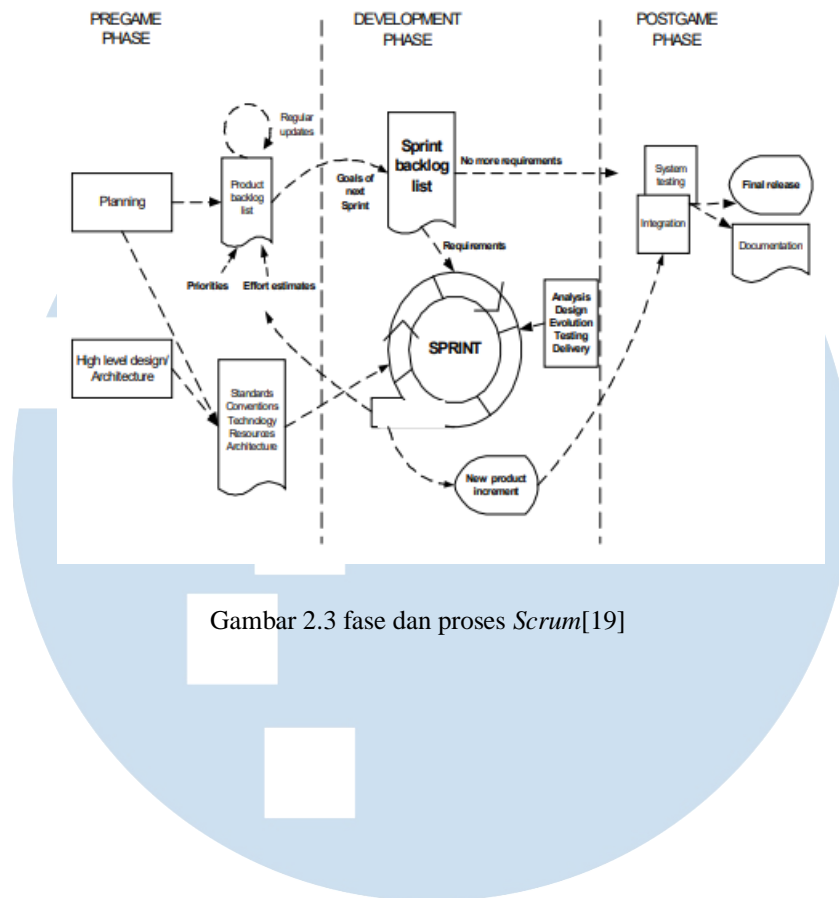
## **2.8 Agile Application Development**

*Agile* merupakan metodologi pengembangan perangkat lunak yang berulang. Pada model pengembangan *agile*, pengembang adalah tim yang *self-organized*, dan juga *cross-functional teams*. Dalam tahapan pengembangannya, pengembang akan terus menganalisa kebutuhan user dan beradaptasi dengan kebutuhan user[18].

### **2.8.1 SCRUM**

*SCRUM* merupakan salah satu turunan dari *Agile*. Ide *SCRUM* sendiri adalah aplikasi dari proses industri ke dalam pengembangan sistem. *SCRUM* tidak menjelaskan secara spesifik bagaimana carai mengembangkan suatu sistem, tetapi *scrum* berfokus pada bagaimana anggota tim harus berfungsi agar dapat menghasilkan sistem secara fleksibel di lingkungan yang terus berubah[19]. *Scrum* memiliki 3 fase yaitu: *pre-game*, *development*, dan *post-game*. Gambar 2.3 menunjukkan proses atau fase Scrum

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2.3 fase dan proses *Scrum*[19]

## 2.9 User Acceptance Testing

*User acceptance testing* atau biasa disebut UAT adalah kegiatan untuk melakukan pengecekan atau *testing* pada sistem yang dikembangkan. Pada saat UAT sistem akan di test dengan kondisi *real* oleh para penggunanya atau klien. Tujuan dari UAT sendiri bukan untuk melihat apakah sistem sudah sesuai dengan keinginan klien, tetapi untuk memastikan sistem yang dibuat sudah memenuhi kebutuhan klien[20].

### 2.9.1 Black Box Testing

*Black box* atau *behavioral testing* merupakan salah satu cara untuk melakukan UAT. Pengujian ini dilakukan dengan mengamati *input* dan *output* dari sistem tanpa mengetahui struktur kode dari sistem yang dikembangkan[21]. Tujuan dilakukannya testing ini adalah untuk mengetahui apakah sistem yang

dikembangkan sudah sesuai dengan apa yang dibutuhkan oleh klien. Selain itu *testing* ini juga dilakukan untuk mendeteksi apakah ada kesalahan pada sistem yang dapat diperbaiki atau tidak.

## **2.10 Unified Modeling Language**

*Unified modeling language* atau UML memiliki tujuan untuk menyediakan suatu Bahasa untuk pemrograman *object-oriented* dan pembuatan diagram yang mampu memodelkan semua sistem yang akan dikembangkan[22]. UML digunakan pada proses analisa dan perancangan awal dari suatu sistem hingga implementasi suatu sistem.


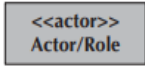



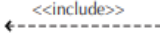
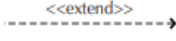

### **2.10.1 Use-case Diagram**

*Use-case diagram* menyediakan cara mudah untuk menginformasikan kepada user atau klien, bagaimana sistem akan berjalan dan fitur apa saja yang ada di dalamnya[22]. Selain itu *use-case diagram* juga dapat mendorong user atau klien untuk memberikan masukan atau *requirement* lagi agar sesuai dengan keinginannya.





Tabel 2.1 komponen use-case diagram

<p><b>An actor:</b></p> <ul style="list-style-type: none"> <li>■ Is a person or system that derives benefit from and is external to the subject.</li> <li>■ Is depicted as either a stick figure (default) or, if a nonhuman actor is involved, a rectangle with &lt;&lt;actor&gt;&gt; in it (alternative).</li> <li>■ Is labeled with its role.</li> <li>■ Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead.</li> <li>■ Is placed outside the subject boundary.</li> </ul>	 <p>Actor/Role</p> 
<p><b>A use case:</b></p> <ul style="list-style-type: none"> <li>■ Represents a major piece of system functionality.</li> <li>■ Can extend another use case.</li> <li>■ Can include another use case.</li> <li>■ Is placed inside the system boundary.</li> <li>■ Is labeled with a descriptive verb-noun phrase.</li> </ul>	 <p>Use Case</p>
<p><b>A subject boundary:</b></p> <ul style="list-style-type: none"> <li>■ Includes the name of the subject inside or on top.</li> <li>■ Represents the scope of the subject, e.g., a system or an individual business process.</li> </ul>	 <p>Subject</p>
<p><b>An association relationship:</b></p> <ul style="list-style-type: none"> <li>■ Links an actor with the use case(s) with which it interacts.</li> </ul>	
<p><b>An include relationship:</b></p> <ul style="list-style-type: none"> <li>■ Represents the inclusion of the functionality of one use case within another.</li> <li>■ Has an arrow drawn from the base use case to the used use case.</li> </ul>	
<p><b>An extend relationship:</b></p> <ul style="list-style-type: none"> <li>■ Represents the extension of the use case to include optional behavior.</li> <li>■ Has an arrow drawn from the extension use case to the base use case.</li> </ul>	
<p><b>A generalization relationship:</b></p> <ul style="list-style-type: none"> <li>■ Represents a specialized use case to a more generalized one.</li> <li>■ Has an arrow drawn from the specialized use case to the base use case.</li> </ul>	

Tabel 2.1 menunjukkan komponen atau elemen elemen yang ada dan digunakan pada pembuatan *use-case diagram* yang berupa *actor*, *use case*, *subject*, *association*, *include*, *extend*, dan *generalization*.

- *Actor*: merupakan orang yang mendapatkan keuntungan. Biasanya digambarkan berupa *stickman* dan dinamai *rolenya* di bagian bawah. *Actor* dapat dihubungkan dengan *actor* lainnya dengan spesialisasi atau *super user*.





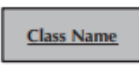





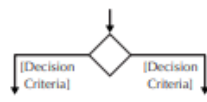
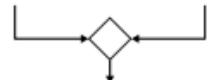


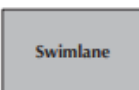
- *Use case*: merepresentasikan suatu bagian dari fungsionalitas sistem. Dapat di *extend* dan *include* dengan *use case* lainnya. Diletakkan di dalam *system boundry*.
- *Subject Boundry*: mereprentasikan *scope* dari suatu subjek seperti suatu *business process* atau sistem
- *Associationn relationship*: menghubungkan *actor* dengan *use-case*.
- *Include relationship*: merepresentasikan satu *use case* berisi *use case* lainnya.
- *Extend relationship*: merepresentasikan ekstensi dari satu *use case* dimana *use case* tersebut dapat memiliki *behaviour* lainnya.
- *Generalization relationship*: merepresentasikan *specialized use case* ke *general use case*.

### **2.10.2 Activity Diagram**

*Activity diagram* digunakan untuk membuat model segala jenis proses yang akan dilakukan oleh user pada sistem[22]. *Activity diagram* juga digunakan untuk memodelkan perilaku dari suatu *activity* dan juga *class*.



Tabel 2.2 komponen *activity diagram*

<p><b>An action:</b></p> <ul style="list-style-type: none"> <li>■ Is a simple, nondecomposable piece of behavior.</li> <li>■ Is labeled by its name.</li> </ul>	
<p><b>An activity:</b></p> <ul style="list-style-type: none"> <li>■ Is used to represent a set of actions.</li> <li>■ Is labeled by its name.</li> </ul>	
<p><b>An object node:</b></p> <ul style="list-style-type: none"> <li>■ Is used to represent an object that is connected to a set of object flows.</li> <li>■ Is labeled by its class name.</li> </ul>	
<p><b>A control flow:</b></p> <ul style="list-style-type: none"> <li>■ Shows the sequence of execution.</li> </ul>	
<p><b>An object flow:</b></p> <ul style="list-style-type: none"> <li>■ Shows the flow of an object from one activity (or action) to another activity (or action).</li> </ul>	
<p><b>An initial node:</b></p> <ul style="list-style-type: none"> <li>■ Portrays the beginning of a set of actions or activities.</li> </ul>	
<p><b>A final-activity node:</b></p> <ul style="list-style-type: none"> <li>■ Is used to stop all control flows and object flows in an activity (or action).</li> </ul>	
<p><b>A final-flow node:</b></p> <ul style="list-style-type: none"> <li>■ Is used to stop a specific control flow or object flow.</li> </ul>	
<p><b>A decision node:</b></p> <ul style="list-style-type: none"> <li>■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path.</li> <li>■ Is labeled with the decision criteria to continue down the specific path.</li> </ul>	
<p><b>A merge node:</b></p> <ul style="list-style-type: none"> <li>■ Is used to bring back together different decision paths that were created using a decision node.</li> </ul>	
<p><b>A fork node:</b></p> <ul style="list-style-type: none"> <li>Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)</li> </ul>	
<p><b>A join node:</b></p> <ul style="list-style-type: none"> <li>Is used to bring back together a set of parallel or concurrent flows of activities (or actions)</li> </ul>	
<p><b>A swimlane:</b></p> <ul style="list-style-type: none"> <li>Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action)</li> <li>Is labeled with the name of the individual or object responsible</li> </ul>	

Tabel 2.2 memperlihatkan komponen komponen yang ada dan dapat digunakan dalam pembuatan *activity diagram*. Komponen dari *activity diagram* terdiri dari:

- *Action*: merupakan suatu *behaviour*.
- *Activity*: merepresentasikan suatu aktivitas.
- *Object Node*: merepresentasikan objek yang terhubung ke suatu *object flow*.
- *Control Flow*: memperlihatkan urutan eksekusi

- *Object flow*: memperlihatkan *flow* suatu objek dari satu aktivitas ke aktivitas lainnya
- *Initial node*: mewakili awal dari suatu aktivitas.
- *Final-activity node*: digunakan untuk memberhentikan semua control flow dan objek flow.
- *Final-flow node*: digunakan untuk memberhentikan *control flow* atau *objek flow* spesifik.
- *Decision Node*: merepresentasikan *test condition* untuk memastikan *control* dan *object flow* mengalir ke satu arah.
- *Merge Node*: berfungsi untuk mengabungkan kembali *node* yang terpisah.
- *Fork Node*: berfungsi untuk memisahkan perilaku ke flow parallel terpisah.
- *Join node*: berfungsi untuk mengabungkan kembali *parallel node*.
- *Swimlane*: berfungsi untuk memisahkan *activity* diagram menjadi *rows* dan *columns* untuk menempatkan aktivitas ke dalam kolom individual dari objek.

## 2.11 Penelitian Terdahulu

Tabel 2. 3 penelitian terdahulu

Judul	Nama Jurnal	Penulis/Tahun	Kesimpulan
Analisis Sistem Informasi Eksekutif Menggunakan User Experience Questionnaire (UEQ)	International Journal of Natural Science and Engineering. Volume 5 Nomor 1 2021, pp 25-29	I Nyoman Tri Anindia Putra, Ketut Sepdyana Kartini, Putu Wirayudi Pratama, Sylvert Prian Tahalea (2021)	Sistem informasi eksekutif STMIK STIKOM masih perlu dikembangkan dan disegarkan dikarenakan sistemnya memiliki nilai above average pada stimulasi dan kebaruan.
Developing a UX KPI based on the user experience questionnaire	Computer Standards & Interfaces Volume 65, July 2019, Pages 38-44	Andreas Hinderks, Martin Schrepp, Francisco José Domínguez Mayo, María José Escalona Cuaresma, Jörg	Amazon memiliki nilai keseluruhan 1.336 Skype memiliki nilai keseluruhan 0.896.

		Thomaschewski (2019)	
User Experience of NAVID E-learning System in the School of Nutrition and Food Technology of Kermanshah University of Medical Sciences, Iran (2020)	Educ Res Med Sci. 2021 June; 10(1):e117418.	Mostafa Nachvak, Ehsan Sadeghi, Reza Mohammadi, Mansour Rezaei, Hadi Abdollahzad, Davood Soleimani (2020)	Dengan melakukan evaluasi sistem menggunakan metode UEQ, sistem NAVID memiliki nilai yang buruk pada segi <i>attractiveness</i> , <i>pragmatic</i> , dan <i>hedonic</i> .
Evaluating User Experience of a Mobile Health Application 'Halodoc' using User Experience Questionnaire and Usability Testing	Jurnal sistem informasi volume 17, Issue 1, April 2021	Mochammad Aldi Kushendriawan, Panca O. Hadi Putra, Harry B. Santoso, Martin Schrepp (2021)	hasil penelitian menunjukkan bahwa nilai UEQ yang dimiliki aplikasi halodoc adalah positif dikarenakan <i>average value</i> yang dimiliki lebih besar dari 0.8. selain itu penelitian menunjukkan bahwa terlibatnya <i>end-user</i> sangat penting. Metode UEQ direkomendasikan dalam penelitian pada observasi <i>solution based</i> .
Open-source ERP: ODOO implementation at micro small medium enterprises: (A case study approach at CV. XYZ in module purchasing and production).	2017 International Conference on Information Management and Technology (ICIMTech)	Limantara, N., Jingga, F. (2017).	Dengan dilakukannya implementasi sistem ERP Odoos perusahaan XYZ perusahaan menjadi lebih terorganisir dikarenakan terdapat pencatatan yang rapi dan jelas. Selain itu perusahaan juga mendapatkan tingkat integritas dan validitas yang tinggi antara PO dan juga Invoice sistem yang disediakan Odoos. Selain itu perusahaan juga mendapatkan kemudahan dengan terbentuknya laporan keuangan secara otomatis yang terstruktur pada sistem Odoos.
Analysis and Design of Web-Based Information System for Church	Vol 12 No 2 (2021): Ultima Infosys: Jurnal Ilmu Sistem Informasi	Jansen Wiratama, Ririn Ikana Desanti (2021)	Sistem berhasil di implementasi dengan metode WDLC dan dapat berhasil

Congregations Case Study: Church BNKP Pewarta			menyimpan data jemaat.
---	--	--	------------------------

Pada penelitian pertama dilakukan penelitian pada sistem informasi eksekutif STMIK STIKOM menggunakan metode UEQ hasilnya ditemukan bahwa sistem yang diteliti masih perlu dikembangkan dan juga dipugarkan Kembali. Sistem yang diteliti mendapatkan nilai *above average* pada nilai stimulasi(*stimulation*) dan kebaruan(*novelty*).

Pada penelitian kedua dilakukan penelitian dengan mengembangkan *KPI* untuk *UX* dengan dasar *UEQ* pada aplikasi skype dan juga amazon. Dimana penelitian dilakukan untuk melakukan pada *UX* dari kedua aplikasi tersebut. Hasil yang didapatkan pada penelitian menunjukkan bahwa nilai dari *Amazon* memiliki nilai 1.336 dan *Skype* memiliki nilai 0.896.

Pada penelitian ketiga dilakukan penelitian pada sistem *NAVID E-Learning* yang dipakai oleh universitas kedokteran di Iran. Penelitian tersebut dilakukan dikarenakan pada masa *COVID-19* sistem *E-Learning* merupakan sistem yang sering dipakai. Metode UEQ digunakan pada penelitian untuk mengukur kualitas dari sistem tersebut dan hasil yang didapatkan adalah sistem *NAVID* memiliki nilai yang buruk (*Bad*) pada segi *attractiveness*, *pragmatic*, dan *hedonic*.

Pada penelitian keempat dilakukan evaluasi *UX* pada aplikasi Halodoc. Penelitian bertujuan untuk mengevaluasi aplikasi Halodoc agar pengguna aplikasi tetap menggunakan aplikasi Halodoc dan mencegah terjadinya kehilangan pendapatan. Penelitian ini menggunakan campuran metode *UEQ* dan juga *Usability Testing* pada penelitiannya. Hasil yang didapatkan dari penelitian tersebut adalah aplikasi Halodoc menunjukkan hasil *attractiveness* 1.778, *perspicuity* 1.810, *efficiency* 1.818, *dependability* 1.622, *stimulation* 1.570, dan *novelty* sebesar 1.211. dari hasil tersebut dapat disimpulkan bahwa aplikasi Halodoc dengan menggunakan

*UEQ scale* memiliki nilai positif pada *UX* karena nilai rata-rata yang didapatkan memiliki nilai lebih dari 0.8.

Pada penelitian kelima dilakukan implementasi sistem *Odoo* pada perusahaan XYZ. Penelitian mengimplementasikan modul *purchasing* dan juga *production* dengan sistem *Odoo* untuk perusahaan XYZ. Keuntungan yang didapatkan oleh perusahaan XYZ dengan mengimplementasi sistem *Odoo* adalah: mudahnya untuk melakukan audit pada hasil produksi dikarenakan terdapat laporan yang rapih dan terstruktur yang dibuat secara otomatis oleh sistem. *PO* yang dibuat pada sistem dapat di periksa validitasnya dengan membandingkannya dengan *good receipt* dan juga *invoice*. Selain itu terdapat *JE* atau *journal entries* untuk proses produksi dan juga *purchasing*. Perusahaan juga mendapatkan laporan keuangan yang dapat digunakan oleh para manajer perusahaan untuk kebutuhan evaluasi perusahaan.

Penelitian keenam dilakukan dengan menganalisis dan perancangan sistem yang berfungsi untuk mendata jemaat pada gereja BNKP Pewarta. Penelitian ini menggunakan metodologi WDLC atau *web development life cycle*. Sistem yang dikembangkan pada penelitian ini adalah sistem berbasis web. Hasil yang didapatkan adalah sistem berhasil di implementasikan untuk melakukan pencatatan jemaat. Selain itu pengguna juga mendapatkan fitur CMS untuk melakukan CRUD. Fitur informasi dan berita juga berhasil dibuat dan ditampilkan pada website.

