BAB II
LANDASAN TEORI

1.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu pada Tabel 2.1:

Tabel 0.1 Tinjauan Pustaka

No	Nama	Judul	Metode	Hasil
1	Irna Yunita dan Ilin Sukma (2016)	Sistem Informasi Wedding Organizer Pada Yusma Salon Berbasis Web	Metode yang digunakan kualitatif dan kuantitatifdan	wedding organizer berbasis web sehingga para calon pengantin dapat dengan mudah melihat harga dan apa saja yang ditawarkan wedding organizer pada Yusma salon dan promosi yang dilakukan Yusma Salon dapat terkomputerisasi
2	Ina Najiyah dan Suharyanto (2017)	Sistem Informasi *Wedding Planner* Berbasis *Web*	Metode yang digunakan dalam pembuatan website wedding planner adalah perencanaan, analisis, desain, dan implementasi	hasil penelitian dengan adanya sistem informasi wedding planner berbasis website ini memberikan gambaran kepada calon pengantin (member) untuk merencanakan pernikahannya dengan informasi-informasi yang tersedia di website wedding planner

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

No	Nama	Judul	Metode	Hasil
3	Sri Mulyati dan Miftahur Hisyam (2018)	Rancang Bangun Sistem Informasi Penyewaan Wedding Organizer Berbasis Web Dengan PHP Dan Mysql Pada Kiki Rias	Metode yang digunakan waterfall	hasil penelitian sistem informasi pemesanan pada Kiki Rias wedding organizer yang saat ini masih berjalan secara manual, jadi diperlukannya sistem informasi mobile yaitu berbasis web
4	Vicky Ariandi (2019)	Sistem informasi pemesanan jasa wedding Organizer pada youzhie Penganten Dengan Menggunakan bahasa pemograman PHP dan mysql	Metode yang digunakan waterfall	hasil penelitian Aplikasi Pemesanan Jasa Wedding Organizer berbasis web pada Youzhie Penganten ini dapat mempermudah user mencari dan mendapatkan informasi yang dibutuhkan tentang Jasa Wedding realtime dan online
5	Haris Ginanjar, Ridwan Setiawan (2020)	Perancangan Aplikasi Wedding Organizer Berbasis Android	Metode penelitian yang digunakan adalah Rational Unified Process	Hasil yang didapatkan berupa aplikasi wedding organizer yang dapat di jalankan pada media smarphone dengan sistem operasi android yang telah melalui tahap pengujian

Berdasarkan kelima jurnal tersebut terdapat perbedaan pada penelitian yang akan dilakukan seperti:

1. Tempat studi kasus dalam penelitian yang diambil pada Nineteen Project
2. Penerapan metode yang digunakan yaitu *extreme programming*
3. Sistem yang akan dibangun seperti proses *booking*, info perlengkapan *wedding*, pembayaran, konfirmasi *booking* dan penilaian konsumen.

1.2 *Wedding Organizer*

Menurut Karlina (2020) *Wedding* merupakan pernikan dalam akad perkawinan sesuai dengan ketentuan hukum dan ajaran agama yang tercantum pada UU. No. 1 Tahun 1974.

Menurut Mulyati and Hisyam (2018) *Wedding Organizer* merupakan suatu kelompok atau tim yang membantu dalam penyelenggaraan acara pernikahan maupun kegiatan resmi lainnya. Munculnya pesaing-pesaing baru, maka sangat penting memberikan pelayanan yang baik bagi konsumen.

1.3 ***Mobile***

Aplikasi *mobile* menurut Maressa (2014) yaitu program siap pakai yang direkap untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ketempat yang lain. Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ketempat yang lain serta mempunyai ukuran yang kecil.

1.3.1 ***MySql***

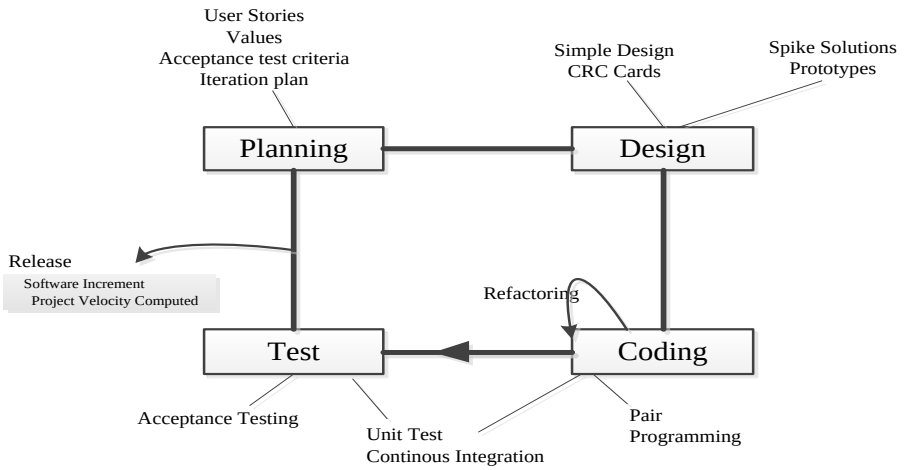
Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structue Query Language* yang digunakan untuk mendefinisikan *structure* data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

Menurut Amin (2018) mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

1.4 *Extreme Programming*

Menurut Suryantara (2017) *extreme programming* adalah pendekatan *extreme programming* merupakan suatu pendekatan berorientasi objek dan sebagai pengembangan perangkat lunak cepat sedikit lebih rinci dengan tujuan memberikan ulasan secara ringkas. Paradigma yang diinginkan mencakup didalam seperangkat aturan dan praktik-praktik dalam empat konteks kegiatan kerangka yang dapat dilihat pada Gambar 2.1:



Gambar 0.1 *Extreme Programming*
Sumber: (Suryantara, 2017)

1.4.1 Tahapan Pengembangan *Extreme Programming*

Tahapan dalam penelitian sebagai langkah-langkah penelitian yang harus dikerjakan, berkut adalah tahapan penelitian *extreme progrmamming* menurut (Suryantara, 2017).

1. Tahap Planning (Perencanaan)

Peneliti atau pengembang memutuskan bagaimana hasil *story* dari pengguna dibangun dengan komitmen telah disepakati, adapun *story-story* yang dilakukan dengan cara :

a. Pengguna menceritakan apa permasalahan pada sistem yang digunakan dan sistem seperti apa yang akan dibangun.Menurut (Schwaber and Sutherland, 2017), *User Story* adalah menceritakan dari perspektif pengguna mengenai apa yang dia inginkan agar lebih seperti yang dilakukan oleh sistem. Berdasarkan uraian tersebut, maka tampilan berubah dari produk ke pengguna sepenuhnya dan *User Stories* menjadi standar persyaratan disemua kerangka kerja sistem.

b. Berdasarkan hasil cerita pengguna maka peneliti menentukan poin pada bagian *value* untuk memutuskan apa saja yang akan dibangun.

c. Dari hasil kesepakatan tersebut maka peneliti menentukan *acceptence criteria test* yaitu menentukan kriteria-kriteria apa saja yang nantinya sebagai acuan terhadap sistem yang akan di uji.

d. Sehingga hasil peneliti menyimpulkan berapa kali akan dilakukan *realies* dan perbaikan pada tahap *interation plan* merencanakan berapa kali akan dilakukan uji terhadap sistem yang dibangun.

2. Tahap Design (Pemodelan)

Extreme programming pada proses pembuatan desain di lakukan untuk memberikan informasi gambaran sistem yang akan dibangun, berikut adalah beberapa desain yang akan dilakukan oleh peneliti:

a. CRC card untuk mengenali dan mengatur *object oriented class* yang sesuai dengan pengembangan. Jika pada saat perancangan terdapat ketidak sesuaian maupun perbaikan maka akan dilakukan
b. *Spike solution* yang dilakukan kepada pengguna untuk mendapatkan kesesuaian antara ke inginan pengguna dengan pengembangan yang dilakukan.
c. *Prototype* adalah bagian perancangan berupa *user interface* dalam bentuk *wireframing* untuk mempermudah pengguna melihat desain sistem.

3. Tahap Coding (Pengkodean)

Pada proses pengkodean peneliti menyesuiakan terhadap *story* pengguna sehingga sistem yang dibangun sesuai, proses pengkodean yang diakukan yaitu:

a. *Pair Programming* merupakan tahap sistem dibangun dengan bahasa pemrograman dan media penyimpanan yang telah disepakati.
b. *Refactory* merupakan tahapan yang dilakukan ketika terjadi ketidak sesuaian kode program sehingga dilakukan perbaikan guna mendapatkan hasil yang sesuai.

4. Tahap Testing (Pengujian)

Tahap pengujian dilakukan oleh pengguna sebagai user dengan melakukan uji sesuai dengan *acceptance test* yang telah ditentukan dan disetujui. Unit test yang

telah dibuat fokus pada keseluruhan fitur dan fungsional sistem. Sehingga sistem dapat disimpulkan telah sesuai dan dapat di *realies*.

Menurut Suryantara (2017) kelebihan pada XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: "*What is the simplest thing that could possibly work?*" Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan. Komunikasi yang lebih banyak mempermudah, dan rancangan yang sederhana mengurangi penjelasan.
2. Setiap *feed back* ditanggapi dengan melakukan tes, *unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

Menurut Suryantara (2017) kelemahan model *Extreme Programming* yaitu:

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Tidak bisa membuat kode yang detail di awal (*prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).

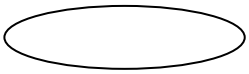
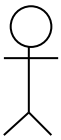
1.5 UML *(Unified Modelling Language)*

Menurut A.S. and Shalahuddin (2019), UML *(Unified Modelling Language)* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML *(Unified Modelling Language).*

1.5.1 *Use Case Diagram*

Menurut A.S. and Shalahuddin (2019) *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.2.

Tabel 0.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		*Usecase* Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal *frase* nama *use case*.
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda

Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)

No	Simbol	Deskripsi
3.		Asosiasi/association merupakan komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
4.		Generalisasi *(generalization)* merupakan hubungan (umum – khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<<*Include*>>	Include berarti *use case* yang ditambahkan akan dipanggil saat *use case* tambahan dijalankan.
6.	<<*Extend*>>	Ekstensi *(extend)* merupakan use case tambahan ke sebuah *use case* yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

Sumber :(A.S. and Shalahuddin, 2019)

1.5.2 *Activity Diagram*

MenurutA.S. and Shalahuddin (2019) *activity* diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 0.3 Simbol *Activity* Diagram

No.	Simbol	Keterangan
1.	●	Status awal aktivitas sitem, sebuah diagram aktivitas memiliki sebuah status awal.

Tabel 2.3 Simbol *Activity* Diagram Lanjutan

No.	Simbol	Keterangan
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan *(Decision)* merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan *(Join)* merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Nama *swimlane*	*Swimlane* Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber :(A.S. and Shalahuddin, 2019)

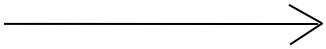
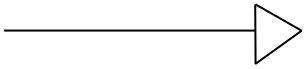
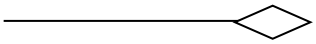
1.5.3 *Class Diagram*

Menurut A.S. and Shalahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4.

Tabel 0.4 Simbol *Class* Diagram

No.	Simbol	Deskripsi
1.	Nama_kelas *+Atribute* +Operasi	Kelas pada struktur sistem**.**

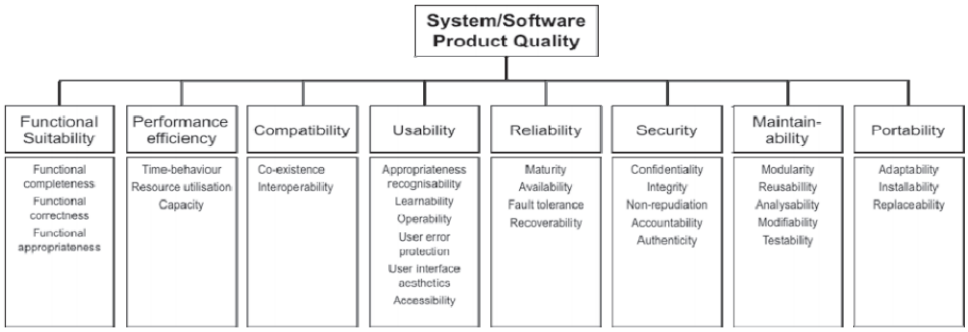
Tabel 2.4 Simbol *Class* Diagram

No.	Simbol	Deskripsi
2.	Antar Muka/*Interface* Nama_*Interface*	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
3.	Asosiasi / *Asociation*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / *Directed Association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / *aggregation* 	Relasi antar kelas dengan maksna semua bagian *(whole-part)*

Sumber: (A.S. and Shalahuddin, 2019)

1.6 Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* (International Organisation for Standardisation, 2011) dapat dilihat pada gambar berikut:



Gambar 0.2 ISO 25010

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer.

Berdasarkan tahapan-tahapan ISO 25010 tersebut maka peneliti menggunakan pengujian terhadap kualitas perangkat lunak berupa aplikasi *web* dapat dinilai dari empat aspek yaitu:

1. *Functional Suitability*

   Sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan di bawah kondisi tertentu. Sub pengujian tersebut sebagai berikut:

   a. *Functional Completeness* (Kelengkapan Fungsional)

      Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan

   b. *Functional Correctness* (Kebenaran Fungsional)

      Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.

   Alat pengujian yang digunakan yaitu kuisioner yang memiliki 2 jawaban pertanyaan seperti sukses atau gagal.

2. *Usability*

   Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Sub pengujian tersebut sebagai berikut :

a. *Appropriateness Recognizability*

Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.

b. *Learnability*

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran yang ditentukan untuk menggunakan produk atau sistem dengan efektifitas, efisiensi, bebas dari resiko dan kepuasan dengan tujuan tertentu konteks penggunaan.

Alat pengujian yang digunakan yaitu kuisioner yang memiliki 5 jawaban pertanyaan seperti Sangat Setuju, Setuju, Ragu-ragu, Tidak Setuju dan Sangat Tidak Setuju.

3. *Portability*

Tingkat efektivitas dan efisiensi dimana sistem, produk atau komponen dapat ditransfer dari satu perangkat keras, perangkat lunak atau lingkungan operasional atau penggunaan lainnya ke yang lain. Berikut sub pengujian tersebut:

a. *Adaptability*

Sejauh mana suatu produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan yang berbeda atau berkembang.

b. *Coexistence*

Sejauh mana sistem dapat bekerja pada perangkat lunak yang ada.

Alat yang digunakan untuk pengujian tersebut berupa *cross browsing compability testing* seperti *chrome, firefox* dan *opera*.

4. *Performance efeciency*.

kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang dinyatakan, berikut sub pengujian tersebut:

a. *Time Behaviour*

Sejauh mana respon dan waktu pemrosesan dan tingkat throughput dari suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan

b. *Resource Utilization*

Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan

Alat pengujian yang digunakan yaitu dengan melakukan pengecekan performa sistem pada media *web test* secara online sehingga mampu menampilkan detail performa sistem, adapun link *web test* yaitu https://www.webpagetest.org/.