

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan (Nas, 2018). Sistem informasi merupakan kumpulan dari komponen-komponen yang saling berhubungan dan melakukan pengolahan data menjadi informasi sehingga informasi tersebut dapat digunakan untuk tujuan tertentu (Hendra K., dan Wicaksono B.B., 2019).

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.2 Kuliner

Kuliner adalah hasil olahan yang berupa masakan berupa lauk-pauk dalam makanan sehari-hari, panganan maupun minuman. Kuliner tidak terlepas dari masak-memasak yang erat kaitannya dengan konsumsi makanan sehari-hari atau bisa juga makanan khas dari berbagai kota maupun provinsi. Kata kuliner merupakan unsur serapan bahasa Inggris yaitu *Culinary* yang berarti berhubungan dengan memasak (Kartini dan Silitonga, 2018).

2.3 Website

Website adalah keseluruhan halaman-halaman yang terdapat dari sebuah domain yang mengandung informasi atau yang berisikan dokumen-dokumen multimedia teks, gambar, suara, animasi, dan video (Urbieta *et al*, 2019). *Website* adalah keseluruhan halaman-halaman yang terdapat dari sebuah domain yang mengandung informasi atau yang berisikan dokumen-dokumen multimedia teks, gambar, suara, animasi, dan video (Prasetya *et al*, 2020).

2.4 Basis Data (*Database*)

Basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari *database* (MySQL, 2018).

Berikut ini beberapa tujuan basis data :

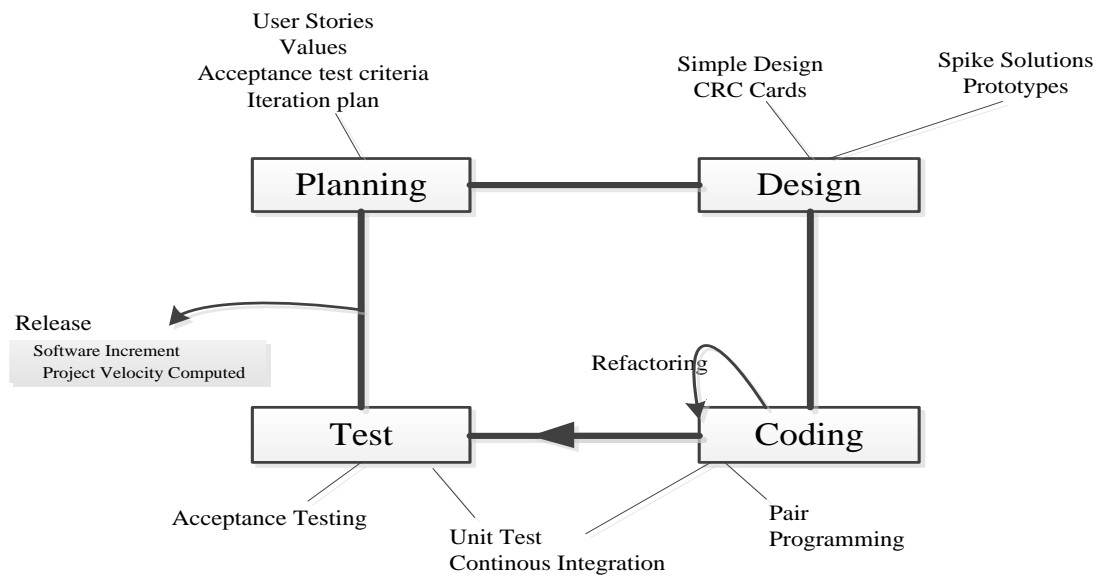
1. Kecepatan dan kemudahan (*Speed*).
2. Efisiensi ruang penyimpanan (*Space*).
3. Ketersediaan (*Availability*).
4. Kelengkapan (*Completeness*).
5. Keamanan (*Security*).
6. Kebersamaan (*Sharability*).

2.5 *Structured Query Language (SQL)*

Structured Query Language (SQL) adalah bahasa pemrograman yang digunakan dalam mengakses, mengubah, dan memanipulasi data yang berbasis *relational Structured Query Language (SQL)* merupakan sistem manajemen *database relation (RDBMS)* yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk pada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak (MySQL, 2018).

2.6 *Metode Extreme Programming*

Extreme programming berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming* (Suryantara, 2017).



Gambar 2.1 *Extreme Programming*

Sumber: (Suryantara, 2017)

2.6.1 Kerangka Kerja *Extreme Programming*

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing*.

1. *Planning*/Perencanaan

Tahap ini dimulainya dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

2. *Design*/Perancangan

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan digunakan pada diagram UML.

3. *Coding*/Pengkodean

Hal utama dalam pengembangan menggunakan XP yaitu *pari programming* (Proses pembuatan program melibatkan 2 atau lebih *programmer*).

4. *Testing*/Pengujian

Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

2.6.2 Keuntungan *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu:

- a. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi.
- b. Saling menghargai antar *developer* dan meningkatkan komunikasi.
- c. Dapat menjadi pembelajaran bagi orang lain.
- d. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
- e. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat.
- f. *Developer* tidak berkerja secara berlebihan.
- g. Dengan XP dapat membuat keputusan yang bersifat teknikal.

2.6.3 *Class Responsibility Collaboration (CRC) Card*

Class Responsibility Collaboration (CRC) Card merupakan *tools* yang digunakan untuk mendefinisikan *behavior* dan *responsibility* dari masing-masing *class* dan hubungan kolaborasi antara *class-class* tersebut. *CRC Card* dikembangkan dengan interaksi antara analis dan pengguna. Masing-masing orang diminta untuk mendeskripsikan logika yang diperlukan untuk memenuhi suatu *responsibility*, dan informasi apa saja yang dibutuhkan tetapi tidak dimilikinya. *Class* lain yang memiliki informasi yang dibutuhkan akan menjadi *collaborator* untuk *behavior* tersebut (Rosa dan Salahuddin, 2019).

2.7 Alat Pengembang Sistem

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram (Agarina dan Karim, 2019). Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut (Rosa dan Salahuddin, 2019) *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Tujuan penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep

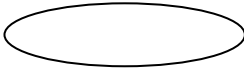
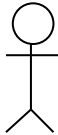

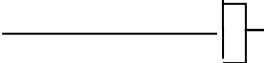
berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia (Purwati N., Halimah, dan Rahardi A., 2018).

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

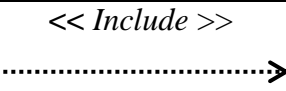
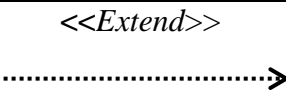
2.7.1 Use Case Diagram

Menurut (Rosa dan Salahuddin, 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 2.1.

Tabel 2.1 Simbol *Use Case Diagram*

| No | Simbol | Deskripsi |
|----|---|---|
| 1. |  | <i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> . |
| 2. |  | Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda. |
| 3. |  | Asosiasi(<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| 4. |  | Generalisasi (<i>generalization</i>) : merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum. |

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)


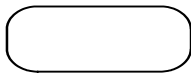
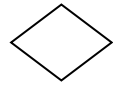

| No | Simbol | Deskripsi |
|----|---|--|
| 5. |  | <i>Include</i> berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan. |
| 6. |  | Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. |

Sumber: (Rosa dan Salahuddin, 2019)

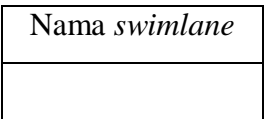

2.7.2 Activity Diagram

Menurut (Rosa dan Salahuddin, 2019) *Activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi aktivitas langkah lain (Yuliawati D., Sushanty S., dan Indra, 2019). Simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*

| No. | Simbol | Keterangan |
|-----|---|---|
| 1. |  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2. |  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3. |  | Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4. |  | Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |

Tabel 2.2 Simbol *Activity Diagram* (Lanjutan)




| No. | Simbol | Keterangan |
|-----|---|--|
| 5. |  | <i>Swimlane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas. |
| 6. |  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

Sumber: (Rosa dan Salahuddin, 2019)

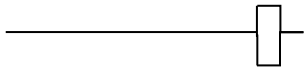
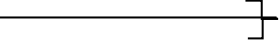
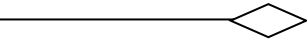
2.7.3 Class Diagram

Menurut (Rosa dan Salahuddin, 2019) *Class Diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* menggambarkan struktur dari sistem (Febriani, O.M., Putra A., Prayogie, 2020). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

| No. | Simbol | Deskripsi |
|-----|---|--|
| 1. |  | Kelas pada struktur sistem. |
| 2. | Antar Muka/ <i>Interface</i>  Nama_ <i>Interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| 3. | Asosiasi / <i>Asociation</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>symbol</i> . |

Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

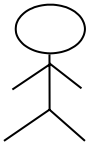
| No. | Simbol | Deskripsi |
|-----|--|--|
| 4. | Generalisasi  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus). |
| 5. | Ketergantungan / <i>dependency</i>  | Relasi antar kelas dengan makna ketergantungan antar kelas. |
| 6. | Agregasi / <i>aggregation</i>  | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>). |

Sumber: (Rosa dan Salahuddin, 2019)


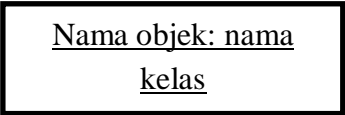



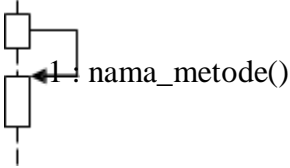
2.7.4 Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada *Sequence Diagram* pada Tabel 2.4.


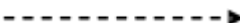
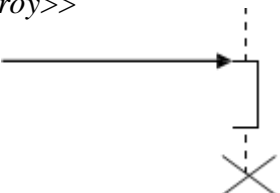
Tabel 2.4 Simbol-simbol *Sequence Diagram*

| No. | Simbol | Deskripsi |
|-----|---|---|
| 1. | Aktor  Atau <u>Nama</u> aktor Tanpa waktu aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal <i>frase</i> nama aktor. |

Tabel 2.4 Simbol-simbol *Sequence Diagram* (Lanjutan)

| No. | Simbol | Deskripsi |
|-----|--|--|
| 2. | Garis hidup /lifeline  | Menyatakan kehidupan suatu objek. |
| 3. | Objek  | Menyatakan objek yang berinteraksi peran. |
| 4. | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
| 5. | Pesan tipe <i>create</i> <<create>>  | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| 6. | Pesan tipe <i>call</i> 1 : nama_metode()  | Menyatakan suatu objek memanggil oprasi/metode yang ada pada objek lain atau dirinya sendiri.  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode, maka operasi / metode yang dipanggil harus ada. |

Tabel 2.4 Simbol-simbol *Sequence Diagram* (Lanjutan)

| No. | Simbol | Deskripsi |
|-----|--|---|
| 7. | Pesan tipe <i>send</i> 1 : masukan  | Meyatakan bahwa suatu objek mengirimkan data/masukan/informasi/ ke objek lainnya, arah panah mengarah pada objek yang menerima kembalian. |
| 8. | Pesan tipe <i>return</i> 1: keluaran  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 9 | Pesan tipe <i>destroy</i> <<destroy>>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i> . |

Sumber : (Rosa dan Salahuddin, 2019)

2.8 Visual Studio Code

Visual Studio Code adalah sebuah *teks editor* ringan dan handal yang dibuat oleh Microsoft untuk sistem informasi *multi platform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. *Teks editor* ini secara langsung mendukung beberapa bahasa pemrograman (Sulthan Raka, 2020).

2.9 Microsoft Visio

Microsoft visio adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*Flowchart*) dan skema jaringan. Dengan *software* ini dapat membantu pengguna dalam meningkatkan kinerja, mulai dari mempersiapkan penggambaran diagram seperti DFD, ERD, UML, jaringan, rancangan *User Interface* dan sejenisnya.

2.10 *Black-Box Testing*

Pengujian *Black-Box testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Taufik H., Mahmudin M., 2018). *Black-Box Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black-Box Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Black-Box Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Tri S. J., 2018).

2.11 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu juga, untuk menghindari anggapan kesamaan dengan penelitian ini. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Tabel 2.5 Penelitian Terdahulu

| No | Nama Peneliti | Judul Penelitian | Hasil Penelitian | Perbedaan |
|----|--------------------------------------|---|---|--|
| 1. | Farida Eni, Setyowibowo Sigit, 2018. | Sistem informasi wisata kuliner berbasis web sebagai media promosi kuliner. | Hasil penelitian ini yaitu memberikan sebuah sistem informasi mengenai kuliner. | Tidak mencantumkan lokasi melalui Google Maps. |

Table 2.5 Penelitian Terdahulu (Lanjutan)

| No | Nama Peneliti | Judul Penelitian | Hasil Penelitian | Perbedaan |
|-----------|---|---|--|---|
| 2. | Muhardi, Febriani Anita, Oktavia Yani, 2018. | Sistem informasi geografis kuliner di Kota Pekanbaru berbasis web. | Menampilkan daftar katalog menu dan lokasi kuliner yang ada di kota Pekanbaru berdasarkan kecamatan yang ada di Kota Pekanbaru. | Tidak menampilkan harga dan kontak untuk menghubungi tempat kuliner, hanya memberikan informasi mengenai lokasi. |
| 3. | Andika, 2020. | pemasaran ukm kuliner di media sosial menggunakan SWOT. | hasil penelitian dapat dijadikan bahan pertimbangan oleh UMKM dalam merumuskan strategi promosi yang dapat dilakukan di media sosial agar dapat dikenal secara luas dan meningkatkan pendapatan. | metode SWOT serta perbedaan observasi didalam metode penelitiannya. |
| 4. | Christian Yefta, Tan Kurf Juriant, 2021. | Perancangan dan implementasi sistem informasi berbasis <i>website</i> untuk lapak kuliner tradisional kepulauan riau menggunakan php 5. | Hasil penelitian pada jurnal ini yaitu memberikan informasi mengenai kuliner khas kepulauan riau . | Penelitian ini khusus untuk kuliner tradisional pada wilayah kepulauan riau, sistem ini hanya sebagai tempat list . |
| 5. | Maulana Zevi Adam, Munthe Ibnu Rasyid, Mutia, 2021. | Pembuatan media informasi kuliner kota medan berbasis web sebagai sarana promosi menggunakan metode Ucd (User Center Design). | Hasil penelitian pada jurnal ini yaitu menampilkan daftar kuliner di kota Medan, pengunjung bisa memasukan saran melalui email. | Penelitian ini menggunakan metode Ucd), tidak adanya informasi mengenai harga yang pada informasi web. |