**BAB II**

**LANDASAN TEORI**

**2. 1. Landasan Kajian Penelitian Sejenis**

Beberapa penelitian yang telah didapat penulis sebagai kajian penelitian sebagai berikut:

Penelitian pertama adalah Jurnal Sistem Informasi yang dibuat oleh Sri

Widaningsih dan Agus Suheri (tahun 2019) mengenai Pengelolaan Data Bank Sampah Berbasis Web di Kabupaten Cianjur. Pada jurnal ini tertulis bahwa untuk melakukan pengelolaan data di bank sampah, Sri Widaningsih dan Agus suheri menggunakan sistem informasi menggunakan PHP dengan database MySQL dan framework CodeIgniter. Sistem yang dibuat disini dapat mengelolah data hasil dari proses transaksi (misalnya penjualan ke pengepul, setoran (pemberian ke petugas) sampah, dan pengambilan saldo/simpanan). Sistem Informasi tersebut tidak terdapat data kategori sampah serta cek saldo nasabah. Akan lebih baik jika ada fitur data kategori sampah sehingga bisa membedakan sampah lebih jelas lagi dan juga fitur cek saldo nasabah agar nasabah bisa mengecek saldonya dengan mudah dari website, tidak harus menuju ke tempat bank sampah.

Penelitian kedua ditulis oleh I Putu Bagus C.D.P, I Gede Suardika, dan Gusti Ngurah Mega Nata yaitu Sistem Informasi Pengelolaan Bank Sampah yang ada di Desa Adat Pemogan, menggunakan Framework Laravel. Pada penelitian ini disampaikan bahwa untuk melakukan pengelolaan data bank sampah, penelitian ini membuat sistem informasi yang menggunakan PHP dengan framework Laravel dan database MySQL. Sistem Informasi tersebut berfokus pada proses transaksi sampah, tidak menjelaskan sampah apa yang di setorkan. Tidak terdapat fitur pendaftaran nasabah. Akan lebih baik jika terdapat fitur data jenis maupun data kategori agar lebih mudah dalam memilah sampah, dan juga fitur registrasi nasabah agar nasabah tidak perlu datang ke tempat untuk melakukan registrasi, dan langsung bisa melakukannya dengan mudah dari web.

Penelitian ketiga adalah Sistem Pengolahan Data Bank Sampah yang ditulis oleh Dhita Prima Kusuma dan Yuli Astuti, sebuah Studi Kasus di Pondok I Ngemplak Sleman. Pada penelitian ini disampaikan bahwa untuk melakukan pengelolaan data bank sampah, dibuat sistem informasi yang menggunakan PHP dan database MySQL. Sistem Informasi tersebut berfokus pada user admin saja, tidak terdapat fitur untuk nasabah, serta tidak ada login untuk nasabah. Akan lebih baik jika terdapat fitur registrasi untuk nasabah serta nasabah bisa mengecek saldonya melalu web agar tidak perlu datang ke tempat bank sampah.

**2. 2. Sistem informasi**

Sistem merupakan gabungan dari beberapa subsistem yang terhubung dan berkerja sama untuk suatu tujuan. Informasi merupakan data yang sudah diolah hingga memiliki nilai dan bermanfaat. Menurut (Sutanta, 2011:10) informasi yaitu hasil pengolahan data sehingga mempunyai kegunaan dalam pengambilan sebuah keputusan dan menjadi bentuk penting bagi penerima.

Sistem informasi merupakan rangkaian prasedur formal untuk menggabungkan data, memprosesnya menjadi informasi, dan

mendistribusikannya kepada pengguna (Kadir, 2014:9).

**2. 3. Sampah**

Sampah adalah benda padat yang yang umumnya tidak dimanfaatkan oleh masyarakat. Pakar kesehatan Amerika mencirikan pengelolaan limbah sampah yang tidak digunakan atau tidak dipakai karena aktivitas manusia, dan tidak akan terjadi tanpa manusia

(Notoatmodjo, 2011: 190).

**2. 4. *Bank* Sampah**

*Bank* sampah merupakan tempat untuk mengumpulkan sampah yang telah melalui tahapan pemilahan. Hasil pemilahan ini diatur dan disimpan pada fasilitas kerajinan sampah atau ke pengepul sampah. Sistem pengelolaan *Bank* sampah mirip seperti bank. Penabung adalah warga yang tinggal di dekat daerah *bank* sampah dan nanti akan mendapat buku tabungan (Rozak, 2014).

**2. 5. Bahasa Pemrograman PHP**

 PHP (Hypertext Prosesor) merupakan bahasa pemrograman

berdasarkan kode yang berjalan di web server, digunakan untuk menangani informasi dan kemudian mengirimkan kembali ke browser internet menjadi HTML. Berbeda dengan HTML, PHP bersifat server-side dimana kompilasi kode program dilakukan pada sisi server. Skrip akan dijalankan di PC server, kemudian server akan mengirimkan hasil program ke PC dari klien. Bahasa pemrograman yang bersifat server-side diantaranya PHP, Perl, JSP, ASP, Coldfusion dan lain-lain (Supono & Putratama, 2018).

**2. 6. Framework Codeigniter**

CodeIgniter merupakan struktur web untuk PHP yang dibuat tahun 2006 oleh Rick Ellis. EllisLab didirikan oleh Rick Ellis yang merupakan kelompok kerja yang dibentuk tahun 2002 yang berspesialisasi dalam pembuatan aplikasi web dan alat bantu untuk pengembang web" (Budi Raharjo, 2015:3). CodeIgniter memiliki fitur yang cukup banyak untuk dapat membantu desainer PHP dalam membangun aplikasi web secara cepat dan mudah.

**2. 7. *Database***

*Database* merupakan sekumpulan data yang terkelola dengan baik dan saling terhubung satu sama lain. Dengan pengelolaan tersebut pengguna dapat mecari informasi atau suatu data, menyimpan data dan menghapus data dengan mudah.. Dimana untuk dapat menampung dan mengatur data yang begitu banyak *database* menggunakan *Relational* *Database* *Management* *Systems* (RDBMS). Menurut (Jubilee, 2014) hal tersebut dikarenakan semua data disimpan dalam tabel-tabel yang berbeda dan dihubungkan berdasarkan relasinya dengan menggunakan *primary* *key* dan *foreign* *key*.

**2. 8. MySql**

MySQL merupakan DBMS yang sangat populer diantara

manajemen sistem database yang lain. Menurut (Jubilee, 2014) “MySQL adalah *Relasional Database Manajement System* (RDMS) yang menangani banyak data. Semua hal dianggap sama, MySQL tidak meminta aset yang cukup besar”. MySQL juga mampu menerima dan mengirim data dengan cepat dan bersifat multi pengguna.

**2. 9. Unified Modeling Language (UML)**

*Unified Modeling Language* merpakan language yang berfungsi mendokumentasikan atau memspesifikasi pembangunan suatu *software*. UML sendiri menurut (Suendri, 2018) "Ini adalah strategi dalam membuat kerangka kerja berbasis objek dan dapat dimanfaatkan sebagai bantuan dalam pengembangan kerangka kerja".

**2.9.1. Use Case**

Use Case adalah diagram pemodelan perilaku dari sistem informasi atau aplikasi yang nantinya akan dikembangkan. Menurut Heriyanto (2018) “*Use case* mendeskripsikan hubungan antara setidaknya satu aktor dan kerangka data. Secara umum, *use case* dipakai untuk mengetahui kemampuan apa yang terkandung dalam kerangka kerja dan siapa yang dapat memanfaatkannya”.

**Tabel 2.1** Tabel simbol *Use* *Case*

|  |  |  |
| --- | --- | --- |
| **Simbol**  | **Nama**  | **Keterangan**  |
|   | *Use Case*  | menggambarkan interaksi dari satu atau lebih aktor dengan sistem yang akan dibuat.  |
|   | Aktor  | Merupakan pengguna dari sistem yang harus ditentukan berdasarkan pembagian tugas dalam peranan di sistem yang akan dibuat. Aktor tersebut dapat berinteraksi dengan *use case* yang ada  |
|   | Asosiasi antara actor dan use case  | Asosiasi berupa garis tanpa panah yang menunjukkan bahwa dapat berinteraksi langsung  |
|   | Asosiasi antara aktor dan *use case*  | Asosiasi dengan garis menggunakan panah yang menunjukkan aktor berinteraksi secara pasif.  |
|   | *Include*  | disebut juga pemanggilan use case olah *use* case lain.  |
|   | *Extend*  | Extend adalah sebuah pengembangan dari *use* *case* apabila memenuhi kondisi dan syarat tertentu.  |

***2.9.2. Activity Diagram***

*Activity* Diagram merupakan aktivitas sistem yang berbentuk kumpulan aksi dimana mencakup mulai dari dimulainya akhis tersebut, proses atau keputusan yang dapat terjadi hingga berakhirnya aksi tersebut. Menurut (Suendri, 2018) "*Activity diagram* juga dapat menggambarkan proses yang dapat membuat lebih dari satu gerakan secara bersamaan. *Activity diagram* adalah sebuah aktifitas, objek, keadaan, transisi keadaan dan *event*".

**Tabel 2.2** Tabel simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol**  | **Nama**  | **Keterangan**  |
|   | *Start*  | Start adalah awal dari suatu aktifitas dan berada di bagian atas *Activity diagram*.  |
|   | *End*  | End adalah akhir dari suatu aktifitas biasanya berada pada bagian bawah.  |
|   | *Activities*  | Activities adalah suatu proses atau kegiatan yang telah terjadi  |
|   | *Swimlane*  | Bertujuan untuk memberikan informasi siapa saja yang sedang mengakses aktifitas.  |
|   | *Decision*  | Decision points digunakan untuk penggambaran pilihan  |
|  |  | pengambilan keputusan true atau false  |

1. **10. Pemodelan Data Conceptual dan Physical**

Pemodelan data secara *conceptual* (CDM) dibuat berdasarkan berbagai obyek yang disebut sebagai entitas dimana masing masing entitas tersebut memiliki relasi (*relationship*) dengan entitas lainnya. Sedangkan pemodelan data secara *physical* (PDM) menggunakan beberapa tabel tertentu untuk menggambarkan informasi terkait data yang ada dan menggambarkan hubungan antara data tersebut dengan lainnya dimana setiap tabel memiliki kolom dengan nama atau identitas yang unik. (Oktafiani, 2018:36). Simbol-simbol yang digunakan diuraikan kedalam tabel dibawah ini.

Tabel 2.3 Tabel simbol CDM dan PDM

|  |  |
| --- | --- |
| **Simbol**  | **Keterangan**  |
|   | *One to One* setiap entitas pada himpunan A memiliki hubungan dengan hanya 1 entitas pada himpunan B, dan sebaliknya.  |
|   | *One to many* setiap entitas pada himpunan A memiliki hubungan 1 atau lebih dengan entitas pada himpunan B, tetapi entitas pada himpunan B hanya memiliki 1 hubungan dengan himpunan A.  |
|   | *Many to one* setiap entitas pada himpunan A memiliki hubungan hanya 1 dengan entitas pada himpunan B, tetapi entitas pada himpunan B dapat memiliki lebih dari 1 hubungan dengan himpunan A.  |
|   | Many to Many setiap entitas pada himpunan A dapat memiliki hubungan dengan banyak entitas pada himpunan B, dan sebaliknya.  |