# BAB II

**LANDASAN TEORI**

# Landasan Kajian Penelitian Sejenis

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

Penelitian pertama yaitu “Analisis Dan Desain Sistem Informasi Manajemen Event Di Perguruan Tinggi Universitas Brawijaya” penelitian ini bertujuan untuk melakukan manajemen event dimana pihak penyelenggara dapat mengajukan proposal kegiatan event kemudian melakukan reservasi ruangan berbasis *website* pada Fakultas Ilmu Administras Universitas Brawijawa (Negoro & Dewantara, 2018). Atas dasar itu sistem tersebut berbeda dengan sistem yang penulis rancang, dimana sistem yang penulis buat berfokus pada pendaftaran, verifikasi syarat pendaftaran dan penyampaian informasi baik pada portal halaman penyelenggara dan melalui email peserta.

Penelitian kedua berjudul “Sistem Informasi Lomba Kota Bekasi” dimana penelitian ini bertujuan menjembatani antara penyelenggara lomba dengan berbagai pihak sponsor dengan menyediakan layananan promosi *event* ataupun perlombaan bagi pihak di daerah Kota Bekasi (Syahputra, Al Akbar, Hutabarat, & Setiawan, 2019). Perbedaan dengan sistem yang dibuat ada pada penyampaian informasi dimana sistem yang kembangan berfokus pada penyampaian informasi proses perlombaan atau *event* dan ditujukan kepada para peserta dan pihak yang terlibat pada masing-masing kegiatan.

Penelitian ketiga yaitu “Rancang Bangun Sistem Informasi Lomba Karya Tulis Ilmiah Di Politeknik Negeri Banjarmasin Berbasis Web” penelitian ini dibuat sebagai tempat arsip dari data-data terkait karya tulis ilmiah yang telah dinilai oleh PHBD internal Poliban pada Politeknik Neger Banjarmasin (Ramadhani, Fitri, & Wardiah, 2019). Berbeda dengan sistem tersebut sistem yang dirancang oleh penulis tidak hanya berfokus pada satu bidang yaitu KTI saja tetapi berbagai bidang atau kategori sesuai yang dibutuhkan oleh masing-masing pihak penyelenggara.

Maka dari itu, berdasarkan dari penelitian sejenis yang telah di jabarkan sebelumnya, perbedaan pada sistem yang akan penulis buat berfokus pada fleksibelitas yaitu penyelenggara dapat mengatur seperti apa kebutuhan berkas pada saat pendaftaran serta bagaimana proses penilaian berlangsung.

# *Website*

*Website* merupakan salah satu *platform* yang memuat berupa gambar, teks, *audio*, dan *video* serta animasi yang dapat diakses melalui jaringan *internet*. Menurut Hidayat (Wibisono & Susanto, 2015) "*Website* dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, dan suara, baik yang bersifat statis maupun dinamis yang membentuk satu rangkain bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman".

### *Website* Sistem Manajemen Informasi

Sistem informasi manajemen merupakan serangkaian sistem informasi yang menyeluruh dan terkoordinasi secara rasional dan terpadu yang mampu mentransformasi data sehingga menjadi suatu informasi lewat serangkaian cara guna meningkatkan produktivitas atas dasar parameter kriteria tertentu yang telah ditetapkan (Purnama, 2016).

# Kompetisi

Menurut Bernstein dalam (Saputra, 2019) kompetisi merupakan suatu kegiatanyang diselenggarakan oleh organisasi tertentu dimana setiap peserta saling bersaing untuk mendapatkan posisi tertentu di kompetisi tersebut. Sebagai persaingan yang terjadi dalam kalangan mahluk hidup di samping kooperasi atau kerja sama.

# *Event*

*Event* merupakan kegiatan yang diselenggarakan oleh penyelenggara tertentu dimana *event* tersebut bertujuan untuk memperingati suatu kejadian penting yang terjadi baik di penyelenggara itu sendiri maupun dimasyarakat pada waktu tertentu dimana *event*  tersebut dapat terkait dengan budaya, agama dan adat. (Noor, 2013).

### Seminar

Seminar merupakan suatu kegiatan atau pertemuan yang diselanggarakan seiringan dengan *event* tertentu dimana kegiatan tersebut membahas topik tertentu dimana pihak penyelenggara mengundang beberapa narasumber ahli sesuai dengan topik yang dibahas (Damanik, 2018).

# Bahasa Pemrograman PHP

PHP (*Hypertext Prosesor*) adalah suatu bahasa pemograman berbasiskan kode-kode (*script*) yang di gunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web* *browser* menjadi kode *HTML*. Berbeda dengan HTML, PHP bersifat *server*-*side* dimana kompilasi kode program dilakukan pada sisi *server*. *Skrip* akan dijalankan pada komputer *server*, kemudian *server* akan mengirimkan hasilnya pada *browser* komputer klien. Bahasa pemrograman yang bersifat *server*-*side* diantaranya Perl, PHP, ASP, JSP, *Coldfusion* dan lain-lain (Supono & Putratama, 2018).

# Framework Codeigniter

Codeigniter adalah sebuah *framework* PHP yang bersifat *open* *source* dan menggunakan metode *Model*, *View* dan *Controller* (MVC). codeigninter dibuat dengan tujuan untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *website* tanpa harus membuat dari awal. MVC sendiri merupakan konsep yang memisahkan kompenen utama menjadi tiga bagian yaitu *model*, *View* dan *controller* (Habibi & Sandi, 2020)

*Model*

Merupakan bagian yang berisi perintah pengolahan mengenai data yang terdapat pada *database*. Seperti mengambil data dari *database*, memasukkan dan mengolah *database*.

*View*

Merupakan bagian yang menangani *user* *interface* dimana *view* merupakan bagian tersendiri dari *controller* dan *model* sehingga memudahkan *web* *designer* dalam membuat halaman *website* yang dibutuhkan.

*Controller*

Merupakan bagian yang berisi kumpulan intruksi menghubungkan *model* dan *view*, dimana data yang tersimpan di *database* (*model*) akan di ambil oleh *controller* kemudian mengirmkan atau menampilkannya ke *view*.

# *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*.

# MySql

MySQL merupakan DBMS yang sangat populer diantara manajemen sistem database yang lain. Menurut (Jubilee, 2014) “MySQL adalah RDBMS (*Relasional Database Manajement System*) yang dapat menangani data yang kapasitas besar. Meskipun begitu, MySQL tidak menuntut resource yang cukup besar”. MySQL juga mampu mengirim dan menerima data dengan sangat cepat dan multi user..

1. **Unified Modeling Language (UML)**

UML atau *Unified* *Modeling* *Language* merupakan bahasa standar yang digunakan untuk mendokumentasikan atau memspesifikasi pembangunan suatu perangkat lunak. UML sendiri menurut (Suendri, 2018) "Merupakan metodologi dalam mengembangkan sistem berorientasi objek serta dapat digunakan sebagai pendukung dalam pembangunan suatu sistem".

1. **Use Case**

Diagram use case adalah pemodelan perilaku dari sistem informasi yang akan dikembangkan atau dibuat. Menurut (Heriyanto, 2018) “*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar*, use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut”.

Tabel 2.1 Tabel simbol *Use* *Case*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Use Case* | Berfungsi menggambarkan fungsionalitas dari sistem sebagai unit yang bisa bertukar pesan atar aktor dan unit |
|  | 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 anak panah menunjukan bahwa siapa atau apa dapat berinteraksi secara langsung |
|  | Asosiasi antara aktor dan *use case* | Asosiasi yang digambarkan dengan garis menggunakan panah menunjukkan bahwa aktor berinteraksi secara pasif. |
|  | *Include* | disebut juga pemanggilan usecase oleh *use* case lain, misalnya pemanggilan sebuah fungsi. |
|  | *Extend* | Extend adalah sebuah perluasan dari *use* *case* apabila memenuhi kondisi dan syarat tertentu. |

1. ***Activity Diagram***

*Activity* Diagram merupakan aktivitas dalam 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 lebih dari satu aksi salam waktu bersamaan. A*ctivity* d*iagram* adalah aktifitas-aktifitas, objek, state, transisi *state* 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. **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. |