# BAB II TINJAUAN PUSTAKA

## Penelitian Terdahulu

Pada penelitian yang dilakukan oleh Sephira & Krisnanik (2019) meneliti proses pendaftaran program MBKM pada Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta yang masih menggunakan Google Form sebagai media untuk mengisi formulir. Oleh karenanya harus dibuatkan sebuah aplikasi yang dapat digunakan mahasiswa untuk mendaftar program MBKM, memilih dosen pembimbing, mendapatkan informasi seleksi, serta mempermudah pengelolaan data pendaftaran. Metodologi penelitian yang digunakan adalah metodologi prototyping, kemudian dirancang menggunakan UML (Unified Modeling Language) dan dibangun menggunakan framework Laravel dengan MySQL sebagai basis datanya.

Penelitian lain dilakukan oleh Agriza et al., (2021) membahas tentang pengembangan sistem informasi untuk memayungi program kewirausahaan yang merupakan salah satu program Merdeka Belajar Kampus Merdeka (MBKM). Karena program Kampus Merdeka masih terbilang baru di Universitas Catur Insan Cendekia (UCIC) sehingga masih belum ada sistem informasi yang bisa memfasilitasi kebutuhan dari program tersebut. Metode yang digunakan dalam pengembangan ini adalah metode Scrum. Hasil yang diperoleh pada pengembangan sistem menggunakan metode SCRUM ini terbukti mampu mengantisipasi perubahan requirements di tahap pengembangan sistem.

Kemudian pada penelitian yang dilakukan oleh Apriani et al., (2022) mengembangkan sistem informasi manajemen kerja praktek Universitas Raharja. Pengembangan sistem informasi ini menggunakan bahasa pemrograman HTML, PHP, dan basis data MySQL. Perancangan sistem dilakukan menggunakan DFD (Data Flow Diagram).

Berdasarkan penelitian-penelitian terdahulu yang sudah disebutkan di atas, akan dilakukan penelitian tentang pengembangan fitur *logbook* untuk program Studi Independen, Magang, dan Pertukaran Mahasiswa *outbond* serta fitur untuk SK peserta dan dosen pembimbing dikarenakan penelitian-penelitian sebelumnya masih belum memfasilitasi fitur-fitur tersebut.

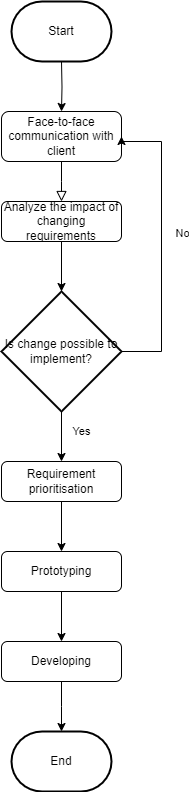
## Teori Terkait

### 2.2.1 Requirement Change Management

*Requirement change management* adalah sebuah proses untuk mengelola perubahan kebutuhan sistem informasi saat proses pengembangan sistem informasi. *Requirement change management* sangat penting agar perubahan kebutuhan terdokumentasi. Menurut Jayatillleke dan Lai (2017) kebutuhan perangkat lunak bisa berubah dan perubahan tersebut tidak dapat dihindari. Ada beberapa faktor yang menyebabkan perubahan kebutuhan perangkan lunak, diantaranya adalah kebutuhan pengguna yang berubah, perubahan pasar, kompetisi dengan produk pesaing, kebijakan pemerintah, dan lain-lain.

Proses-proses pada *Requirement change management* di antaranya adalah :

* + - 1. Komunikasi dengan klien untuk membahas perubahan yang diinginkan klien. Sebisa mungkin dilakukan secara tatap muka agar dapat mempermudah pembahasan dan diskusi mengenai permintaan klien.
      2. Analisis dampak dari perubahan kebutuhan. Analisis dilakukan untuk menentukan apakah perubahan yang diinginkan klien memungkinkan untuk diterapkan. Apabila perubahan yang diterapkan tidak memberikan dampak buruk pada sistem yang sudah ada, maka perubahan bisa diterapkan. Apabila ditemukan permintaan perubahan yang berpotensi akan mengganggu sistem yang sedang berjalan, hal tersebut akan dikomunikasikan dengan klien agar dapat ditemukan solusinya.
      3. Membuat prioritas pengerjaan. Setelah mendapatkan kebutuhan dari klien, selanjutnya adalah menyusun prioritas dari kebutuhan-kebutuhan tersebut. Penyusunan prioritas dilakukan dengan membandingkan kebutuhan baru dengan kebutuhan yang sedang berjalan.
      4. Merancang permintaan kebutuhan baru dari klien agar dapat diterapkan ke dalam sistem.
      5. Pengembangan sistem sesuai dengan rancangan.



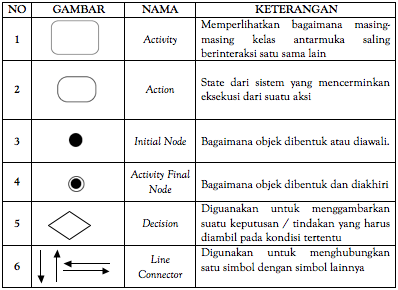
Gambar 2. 1 Flowchart Requirement Change Management

### 2.2.2 CodeIgniter

CodeIgniter merupakan sebuah *framework* PHP yang menggunakan konsep MVC (*Model, View, Controller).* *Framework* ini dapat menjadi *tools* untuk mengembangkan suatu situs dengan lebih mudah karena menyediakan *resource* yang lengkap (Sulistiani & Hendra Saputra, 2020).

### 2.2.3 Activity Diagram

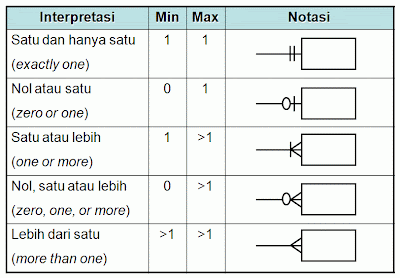
*Activity* *Diagram* adalah sebuah pemodelan yang menggambarkan sistem kerja dari sebuah objek atau sistem. *Activity Diagram* digambarkan dengan sebuah alur secara terstruktur dari *use case* yang sedang diproses dari titik awal sampai titik akhir (Aliman, 2021).



Gambar 2. 2 Notasi Activity Diagram

### 2.2.4 Entity Relationship Diagram (ERD)

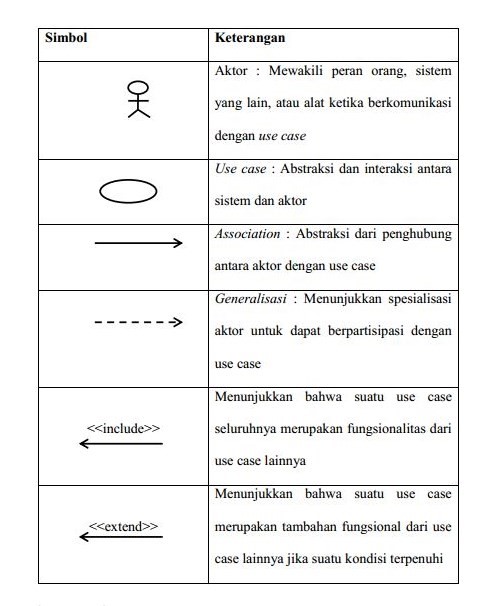
*Entity Relationship Diagram* (ERD) adalah diagram yang digunakan untuk merancang suatu basis data. ERD mendeskripsikan data yang akan disimpan dalam sebuah sistem (Larassati et al., 2019).



Gambar 2. 3 Notasi Kardinalitas

### 2.2.5 Use Case Diagram

*Use case diagram* adalah pemodelan yang digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. *Use case diagram* menjelaskan interaksi yang terjadi antara aktor dengan sistem (Fitriani et al., 2018).



Gambar 2. 4 Simbol Use Case Diagram