

BAB 2

DASAR TEORI

2.1. Pondok Pesantren Mahasiswa

Adapun arti dari pondok pesantren mahasiswa atau PPM merupakan pesantren yang basis nya lebih modern, dimana bukan hanya nyantri saja yang notabene nya mengejar akhirat saja, namun juga diimbangi dengan kuliah mengejar akademisi. Dari namanya, PPM hanya dikhususkan untuk calon santri yang sedang atau akan menempuh pendidikan tinggi disuatu daerah atau kota. Yang menginginkan atau bercita-cita berhasil mendapat ilmu dunia dan akhiratnya. Dimana fasilitas-fasilitas yang diberikan juga berbeda dengan pondok pesantren pada umumnya. Termasuk diantaranya yaitu kamar asrama yang tiap kamar dihuni oleh dua sampai tiga santri, dengan perlengkapan kamar yang lebih baik karena untuk tingkat mahasiswa. Kemudian dewan guru yang lebih berkompeten untuk mengajar mahasiswa. Dan juga, waktu mengaji dan kuliah dibagi sesuai porsinya masing-masing, artinya ketika santri tidak sedang melaksanakan kuliah, maka di asrama santri mengikuti kegiatan pondok pesantren, sedangkan ketika santri sedang kuliah maka santri tidak wajib mengikuti kegiatan pondok pesantren sampai akhir sesi perkuliahan. Karena di PPM sendiri, memiliki semboyan mondok nomer satu kuliah nomer satu. PPM ini biasanya didirikan di sekitar kampus-kampus yang ada di Indonesia, salah satunya di Kota Malang.

2.2. Sistem Informasi Akademik

Pengertian akademik menurut Catur yang dikutip (Liatmaja & Wardati, 2013) akademik adalah suatu bidang yang mempelajari tentang kurikulum atau pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan/pembelajaran yang dapat dikelola oleh suatu sekolah atau lembaga pendidikan.

Sistem informasi akademik adalah suatu sistem yang dibangun untuk mengelola data-data akademik sehingga memberikan kemudahan kepada pengguna dalam kegiatan administrasi akademik secara online menurut balelol dalam (Djaelangkara et al., 2015).

Sedangkan sistem informasi akademik menurut sutoto dalam (Liatmaja & Wardati, 2013) Sistem informasi akademik adalah perangkat lunak yang digunakan untuk menyajikan informasi dan menata administrasi yang berhubungan dengan kegiatan akademik. Dengan penggunaan perangkat lunak seperti ini diharapkan kegiatan administrasi akademik dapat dikelola dengan baik dan informasi yang diperlukan dapat diperoleh dengan mudah dan cepat.

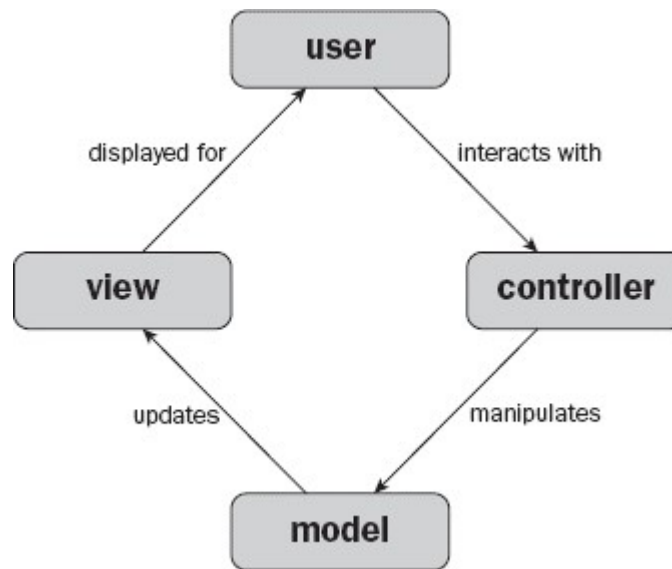
2.3. *Framework Codeigniter*

Framework secara sederhana dapat diartikan kumpulan dari fungsi fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal. (Villela, 2013).

Ada beberapa alasan mengapa menggunakan *Framework*:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi *web*.
- Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada).
- Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, *ORM*, *pagination*, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dll).
- Lebih bebas dalam pengembangan jika dibandingkan *CMS*.

Codeigniter merupakan aplikasi sumber terbuka yang berupa *framework PHP* dengan model *MVC* (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan *PHP*. *Codeigniter* memudahkan *developer* untuk membuat aplikasi *web* dengan cepat mudah dibandingkan dengan membuatnya dari awal. *Codeigniter* dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.4



Gambar 2. 1 MVC Pada Halaman *Website*

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi *web*, berawal pada bahasa pemrograman *Small Talk*, *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu *MVC pattern* dalam suatu aplikasi yaitu:

1. *View* merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi *website* bagian ini biasanya berupa *file template HTML*, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller* merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip *MVC* suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*-nya, yaitu *programmer* yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur *MVC* dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

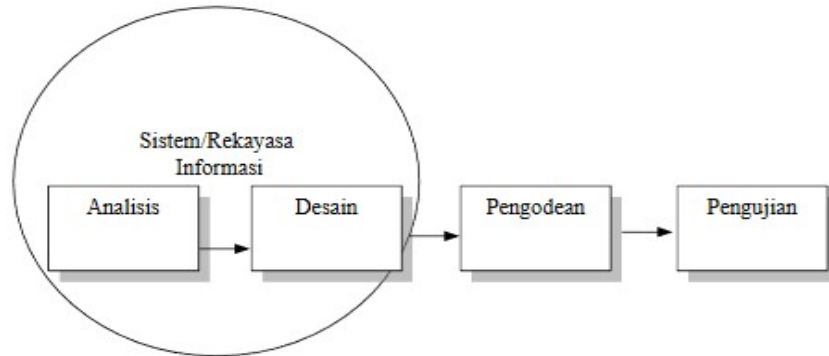
2.4. MySQL

MySQL adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk *server* atau membuat *web*. *MySQL* berfungsi sebagai *SQL* (*Structured Query Language*) yang dimiliki sendiri dan sudah diperluas oleh *MySQL* umumnya digunakan bersamaan dengan *PHP* untuk membuat aplikasi *server* yang dinamis dan *powerfull*. (Istiono et al., 2016).

2.5. Metode Waterfall

Model *Waterfall* merupakan salah satu model pengembangan perangkat lunak yang ada didalam model *SDLC* (*Sequential Development Life Cycle*). Menurut Sukamto dan Shalahuddin (2013:26) mengemukakan bahwa “*SDLC* atau *Software Development life Cycle* atau sering disebut juga *System Development life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan *best practice* atau cara-cara yang sudah teruji baik.”

Sedangkan Sukamto dan Shalahuddin (2013:28) dijelaskan bahwa model *waterfall* sering juga disebut model sekuensi linier atau alur hidup klasik. Pengembangan sistem dikerjakan secara terurut mulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung. (Firmansyah & Udi, 2017). Adapun penjelasan dari metode ini adalah sebagai berikut.



Gambar 2. 2 Metode *Waterfall*

(Sumber: Sukamto dan Shalahuddin (2013:29))

2.6. Pengujian *Blackbox Testing*

Menurut Rosa dan Shalahuddin (2013:275) “*Blackbox testing* (atau yang disebut pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pengujian dimaksudkan untuk mengetahui apakah fungsi, masukan, serta keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat sebuah kasus uji bersifat mencoba semua fungsi yang ada dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. (Kesuma & Kholifah, 2019).