BAB 2
DASAR TEORI

1.1. Sistem

Sistem merupakan sebuah kesatuan yang terdiri dari beberapa komponen atau elemen yang dihubungkan bersama yang ditujukan untuk memudahkan aliran informasi, materi atau energi untuk mencapai tujuan. Sistem itu sendiri juga bisa bisa terdiri dari beberapa kumpulan *hardware, software* yang diolah agar bisa digunakan dan dimanfaatkan (Wibowo, Arhandi, Mentari, & Amalia, 2019).

1.2. Pembukuan

Pembukuan di setiap instansi memiliki caranya masing-masing, sehingga tidak dapat disamakan, namun tujuannya adalah sama supaya semua keluar-masuk data dapat diketahui kejelasannya. Integrasi dari orang yang bertanggung jawab atas pembukuan diperlukan, baik instansi yang dijalankan kecil maupun besar.

Bahkan pada undang-undang sudah diatur sebagaimana “Pembukuan adalah suatu proses pencatatan yang dilakukan secara teratur untuk mengumpulkan data dan informasi keuangan yang meliputi harta, kewajiban, modal, penghasilan, dan biaya, serta jumlah harga perolehan dan penyerahan barang atau jasa, yang ditutup dengan menyusun laporan keuangan berupa neraca, dan laporan laba rugi untuk periode tahun pajak tersebut . (DPR, 2007).

1.3. Gudang

Gudang merupakan sebuah ruangan atau fasilitas yang digunakan untuk menyimpan berbagai macam barang. Tidak hanya took yang mempunyai gudang, bahkan rumah dan gedung pun juga mempunyai gudang untuk menyimpan barang (Ramadhani, Suryadi, & Irmayani, 2018).Gudang merupakan sebuah fasilitas yang sangat diperlukan dalam instansi terlebih lagi dalam sebuah perusahaan dimana gudang bisa tempat untuk menyimpan produksi yang dihasilkan dari suatu instansi tersebut.

1.4. Database

Database (basis data) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, Database merupakan salah satu komponen yang penting di sistem, karena berfungsi sebagai basis penyedia informasi maupun tempat memanipulasi informasi bagi para pemakainya (Jogiyanto, 2015). Database juga merupakan komponen penting dalam sebuah sistem karena dapat digunakan sebagai pengoganisasian data (Habibi & Suryansah, 2018)

1.5. Solid Design Principle

Terdapat konsep SOLID yang bisa diterapkan pada saat pengembangan aplikasi menggunakan framework Laravel. Desain software SOLID digunakan untuk menjaga skalabilitas aplikasi yang dibuat sehingga dapat dikembangkan tanpa melakukan desain ulang (Habibi & Suryansah, 2018). Berikut adalah gambaran umum dari konsep SOLID :

1. *Single Responsibility Principle*

   Dimana 1 *class* bertanggung jawab atas 1 fitur saja. Sehingga saat ingin mengubah alur data atau distribusi data tidak perlu mengubah keseluruhan kode

2. *Open close Principle*

   Prinsip ini memiliki akronim "*Open for Extension, Close for Modification*" dimana class tersebut bisa digunakan di class lain yang baru, namun tidak perlu dimodifikasi untuk digunakan. Supaya prinsip ini dapat diaplikasikan kita perlu memiliki abstraksi software yang kuat sehingga prinsip open/close dapat diterapkan

3. *Liskov Subtitution Principle*

   Dimana terdapat kelas turunan yang menjadi pengganti dari kelas parentnya dengan cara melakukan override dimana subclass bekerja seperti dengan superclass

4. *Interface Segregation Principle*

   Client tidak perlu mengakses semua method untuk mendapat data yang diperlukan

5. *Dependency Inversion Principle*

   Terdapat 2 module yang ada yaitu *high level* dan *low level*. *High level* module merupakan bagian yang paling dekat dengan presentation layer dimana terjadi

permintaan data dan pengiriman data. *Low level module* merupakan bagian yang mengolah data tersebut dengan cara menerima data dari *high level* dan mengembalikan data tersebut

1.6. Laravel

Laravel adalah *framework* aplikasi web yang dapat menggunakan syntax expressive. Framework Laravel menggunakan secara komponen yang telah dibuat yang dapat menyediakan layer yang saling terhubung dan dapat digunakan untuk membuat aplikasi web lebih terstruktur dan pragmatis (Pakpahan, 2020). *Framework* Laravel ini didirikan di bawah lisensi MIT dan ini mengikuti struktur MVC (*model, view, controller*), dimana struktur tersebut memisahkan data dari tampilan berdasarkan komponen dari aplikasi (Yudhanto & Prasetyo, 2019)

1.7. VPS

VPS (*Virtual Private Server*) sebuah disebut juga *Virtual Dedicated Server* (VDS) atau *Virtual Machine* (VM) adalah sistem yang berjalan di bawah virtualisasi Mesin virtual ini memungkinkan anda untuk membagi resource sebuah server ke dalam beberapa server yang berjalan secara mandiri, tanpa saling mempengaruhi (Laswi, 2017). VPS sendiri bisa membantu meningkatkan fleksibilitas yang tersedia pada administrator sistem untuk memilih konfigurasi *software* yang akan dijalankan

1.8. MVC

MVC (*Model, View, Controller*) adalah sebuah teknik pemrograman yang memisahkan antara alur, data dan antarmuka suatu sistem atau lebih mudahnya MVC adalah framework yang memisahkan antara desain, data dan proses. Model merupakan penghubung langsung dengan *database*, view adalah bagian yang menangani *presentasi logic* yang berfungsi mempresentasikan data kepada *user*, sedangkan *controller* yang mengatur hubungan antara bagian model dan bagian *view* (Wibowo, Arhandi, Mentari, & Amalia, 2019).

1.9. MySQL

MySQL merupakan sebuah Manajemen Basis Data yang bersifat Relasional (RDBMS) yang bersifat *Open Source* dan bisa digunakan pada semua *platform* (Agusvianto, 2017)dan memiliki lisensi di bawah GNU *General Public License* (GPL) (Ramadhani et al., 2019). MySQL berfungsi sebagai perangkat lunak untuk mengolah *Database* menggunakan Bahasa SQL. MySQL juga mendukung Bahasa Pemrograman PHP. (Harianto, Pratiwi, & Suhariyadi,

2019). SQL sendiri merupakan sebuah Bahasa yang dipakai pada *relational database* yang menjadi penghubung antara perangkat lunak dengan aplikasi *database* (Setyawan, 2019)

1.10. Web Service

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (software) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. *Web service* dapat diidentifikasikan dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service* . Berbeda dengan URL web pada umumnya, URL *web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi (Richardson, 2007)