

BAB 2

DASAR TEORI

2.1. Sistem Informasi Keuangan

Sistem Informasi keuangan adalah sistem Informasi yang memberikan Informasi kepada orang atau kelompok baik yang berada didalam perusahaan maupun diluar perusahaan yang memuat mengenai permasalahan keuangan dan juga menyediakan atau berisi Informasi tentang arus uang bagi para pemakai yang ada pada seluruh perusahaan. Adapun arahan untuk menunjukkan fungsi-fungsi Sistem Informasi keuangan dengan pengertian yang telah dijelaskan sebelumnya, maka wujud Sistem Informasi keuangan secara administrasi tertera pada bentuk-bentuk formulir, buku – buku dan catatan – catatan akuntansi serta laporan – laporan yang disajikan. (Setiawan, 2020).

2.2. Codeigniter

Codeigniter menjadi sebuah *framework PHP* dengan model *MVC (Model, View, Controller)* untuk membangun *Website* dinamis dengan menggunakan *PHP* yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *Codeigniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *Codeigniter* sebagai *framework* pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh *Codeigniter*, pembuat *PHP* Rasmus Lerdorf memuji *Codeigniter* di *frOSCon* (Agustus 2008) dengan mengatakan bahwa dia menyukai *Codeigniter* karena “*it is faster, lighter and the least like a framework.*” (Knowledge Base, 2017).

2.3. MySQL

MySQL adalah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, multi user serta menggunakan perintah standar *SQL (Structured Query Language)* dan baik digunakan sebagai client maupun server. (Elisa Usada, 2012)

2.4. PHP

PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. *PHP* dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman *HTML* biasa. Aplikasi-aplikasi yang dibangun oleh *PHP* pada umumnya akan memberikan hasil pada web browser tetapi prosesnya secara keseluruhan dijalankan di *server*, pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode *PHP* untuk mengirimkan permintaan ke *server*. (Elisa Usada, 2012).

2.5. Payment Gateway

Payment Gateway merupakan suatu aplikasi yang melayani *e-commerce* dengan cara kerjanya memproses suatu otorisasi pembayaran seperti kartu debit, transaksi online yang memerlukan pembayaran, dan kartu kredit. Aplikasi *Payment Gateway* ini memberikan fasilitas transaksi dengan mengirim Informasi yang penting antara bank dan pihak perusahaan. (Cermati.com, 2017).

2.6. Midtrans

Midtrans merupakan salah satu layanan *Payment Gateway* yang berada di Indonesia. Teknologi dari *Midtrans* tersebut bisa menjadi salah satu cara untuk mengembangkan bisnis Anda. Caranya yaitu dengan pengembangan Teknologi yang memfasilitasi pembayaran atau transaksi secara online. (IDwebhost, 2019).

2.7. Use Case

Use Case merupakan gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya. *Use Case* adalah teknik penemuan kebutuhan perangkat lunak yang dikenalkan pertama kali dalam metode pendekatan berbasis objek yang dikembangkan oleh Jacobson dan kawan-kawan pada tahun 1990an. (Sommerville, 2011). Untuk

menggambarkan interaksi antara sistem dan lingkungannya, terdapat beberapa simbol yang digunakan pada *Use Case* diagram. *Use Case* yang merepresentasikan tugas tertentu yang melibatkan interaksi dengan lingkungannya dilambangkan dengan elips. Sementara aktor yang terlibat dalam *Use Case* tersebut dilambangkan dengan sosok tongkat (Sommerville, 2011). Terdapat beberapa jenis relasi yang terjadi, baik antar *Use Case*, antar aktor, maupun antara *Use Case* dan aktor. Relasi yang terjadi antar *Use Case*, antara lain *extend*, *include* dan generalisasi. Relasi <<*extend*>> digunakan saat sebuah *Use Case* secara kondisional menambahkan tahapan lain dari *Use Case* pusat (*base Use Case*).

2.8. *Activity Diagram*

Activity Diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. *Activity Diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir. (Rizky, 2019).

2.9. *Conceptual Data Model (CDM)*

Conceptual Data Model (CDM) merupakan model yang merepresentasikan tabel yang merupakan entitas yang berisi atribut. Setiap entitas memiliki satu *primary key* yang bersifat unik (nilainya tidak sama dengan nilai yang lainnya) dan setiap entitas berhubungan dengan entitas lain yang disebut *relationships* (Dwi Oktafiyah Sumadya, R.V. Hari Ginardi, Rizky Januar Akbar, 2016).

2.10. *Physical Data Model (PDM)*

Physical Data Model (PDM) merupakan model yang merepresentasikan tabel yang terstruktur, termasuk nama kolom, tipe data kolom, *primary key*, *foreign key* dan *relationships* yang menghubungkan satu tabel dengan tabel lainnya. *PDM* dan *CDM* memiliki kemiripan dari segi entitas, atribut dan *relationships*. Perbedaannya yaitu pada *CDM* tidak terdapat *foreign key*, sedangkan pada *PDM* terdapat *foreign key* dimana satu

tabel bergantung pada tabel lainnya [4]. *Foreign key* yaitu *primary key* yang berada pada tabel yang memiliki hubungan dengan tabel tersebut (Dwi Oktafiyah Sumadya, R.V. Hari Ginardi, Rizky Januar Akbar, 2016).

2.11. Class Diagram

Class Diagram adalah Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek (Suendri, 2018). *Class* memiliki tiga area pokok yaitu :

- 1) Nama, kelas harus mempunyai sebuah nama.
- 2) Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
- 3) Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.