

PENERAPAN KONSEP MODEL VIEW CONTROLLER PADA PERANCANGAN SISTEM MANAJEMEN SOFTWARE BERBASIS WEB

Suendri

Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

Email: suendri@gmail.com

Abstrak : Sistem manajemen *software* merupakan sistem yang akan dibangun dengan tujuan untuk melakukan pencatatan seluruh produk *software* yang dikembangkan oleh perusahaan yang bergerak dibidang perangkat lunak. Semakin banyaknya pelanggan yang menggunakan produk tersebut akan lebih memudahkan jika perusahaan mempunyai sebuah sistem yang bisa diakses secara *realtime* dan mencatat seluruh data-data penting *software* dan pelanggan serta menampilkan hal penting lainnya untuk menjaga hak cipta atas *software* tersebut disertai garansi yang diberikan. Konsep *Model View Contoller* atau biasa disingkat dengan MVC merupakan sebuah arsitektur dimana proses pada sistem dipisah menjadi tiga bagian dengan tugas-tugas dan tanggungjawab masing-masing bagian. Konsep ini lebih memudahkan *programmer* dalam mengatur *query database*, manajemen kode, validasi data dan keamanan. Bahasa pemrograman yang digunakan adalah PHP sedangkan untuk DBMS sistem dibangun menggunakan *database MySQL*. Penelitian ini diharapkan bisa membantu perusahaan perangkat lunak dalam melakukan pencatatan produk yang telah dijual kepada pelanggan.

Kata kunci : *software*, manajemen, model, *view*, *controller*

Abstract : *Software management system is a system that will be built with the aim of recording all software products developed by companies engaged in software. The increasing number of customers who use these products will be easier if the company has a system that can be accessed in realtime and records all important data software and customers and displays other important things to maintain the copyright of the software along with the warranty provided. The Model View Contoller concept or commonly abbreviated as MVC is an architecture where the process on the system is divided into three parts with the tasks and responsibilities of each part. This concept makes it easier for programmers to manage database queries, code management, data validation and security. The programming language used is PHP while for DBMS, the system is built using a MySQL database. This research is expected to help software companies in recording products that have been sold to customers.*

Keywords : *software*, management, model, *view*, *controller*

1. PENDAHULUAN

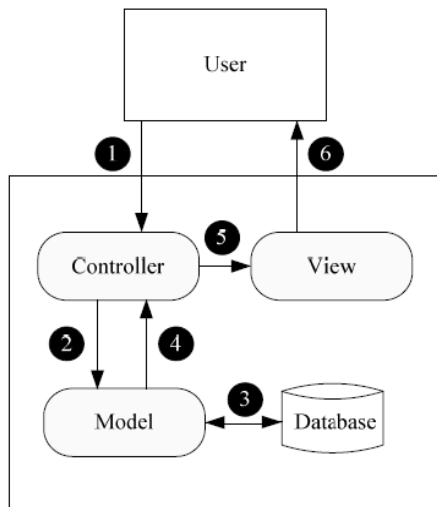
Perkembangan Teknologi Informasi telah banyak mempengaruhi berbagai aspek kehidupan umat manusia dalam menjalankan aktivitas sehari-hari. Penggunaan komputer telah jauh mengalami kemajuan dari sekedar teknologi alat hitung hingga pengambilan keputusan (Suendri, 2017). “Pentingnya teknologi *website* dalam bidang komersial, sosial dan pendidikan meningkatkan kegiatan penelitian dalam pemrograman *web/scripting*. Beberapa metode untuk menulis kode PHP seperti *Object Oriented Programming* (OOP), *coding* PHP prosedural dan pola *Model View Controller* (MVC) telah diusulkan” (Rashidah F.Olanrewaju, 2014). “Pemrograman Berorientasi Objek, atau OOP, merujuk pada metode pemrograman yang memanggil penggunaan kelas untuk mengatur data dan struktur aplikasi. *Object-Oriented Programming* (OOP) adalah model pemrograman yang didasarkan pada konsep kelas dan objek” (Ms. Namrata Jadhav, 2018).

Perkembangan teknologi yang cepat harus dibarengi dengan sistem yang mendukung terciptanya proses ideal untuk sebuah perusahaan. Salah satunya adalah sistem untuk mengatur pencatatan produk bagi perusahaan yang bergerak dibidang perangkat lunak.

2. LANDASAN TEORI

Design-pattern MVC telah diterapkan dibeberapa pengembangan aplikasi diantaranya pada: pendaftaran mahasiswa baru *online* (Fathiah, 2013), sistem informasi keuangan (Hartomo, 2009), kuis online adaptif (Hidayat, 2012). Hasil menunjukkan bahwa penerapan *Design-pattern MVC* lebih baik dari pada pengembangan aplikasi secara konvensional (Arochman, 2016). “Salah satu konsep yang dibuat oleh pakar pemrograman adalah memecah sebuah aplikasi *web* menjadi tiga bagian yaitu model yang berkaitan dengan operasi yang berhubungan dengan basis data, *view* yang berhubungan dengan antarmuka aplikasi dan terakhir *controller* yang berhubungan dengan logika aplikasi dan mengendalikan alur data antara *view* dan *controller* konsep

ini merupakan pengembangan dari konsep pemrograman berorientasi objek dan dalam pemrograman dikenal dengan MVC” (Hasyrif SY, 2016)



Gambar 1 Arsitektur MVC

Pola MVC memecah aplikasi menjadi tiga modul: Model View Controller. Gambar 1 menggambarkan arsitektur kerangka kerja PHP untuk manajemen basis data berdasarkan MVC (Chanchai Supaartagorn, 2011). PHP adalah alat *scripting* untuk *web* yang memungkinkan pengembangan *web* interaktif yang dinamis, intuitif, kompilasi yang cepat, lintas *platform*, sumber terbuka, fleksibel serta pengaturan yang minimal (Rashidah F.Olanrewaju, 2014).

Sumber gambar : Chanchai Supaartagorn, 2011.

3. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah Research and Development (R&D) melalui tahapan sebagai berikut:

1. Wawancara

Pengumpulan data dilakukan melalui wawancara langsung pada pihak perusahaan yang bergerak dibidang perangkat lunak.

2. Studi Pustaka

Studi Pustaka dilakukan untuk mendapatkan semua teori-teori yang mendukung pada arsitektur yang digunakan untuk merancang sistem yaitu *Model View Controller* (MVC).

3. Perancangan Sistem

Setelah mendapatkan data-data yang diperlukan dan teori-teori yang mendukung, maka selanjutnya merancang sistem sesuai dengan tujuan dari penelitian ini.

4. PERANCANGAN

a. Database

Sebelum perancangan sistem dilakukan, perlu disiapkan *database* yang digunakan untuk menyimpan seluruh data-data yang berhubungan dengan proses manajemen *software*. Adapun tabel-tabel yang digunakan adalah sebagai berikut:

1) Tabel Program

Tabel program digunakan untuk menyimpan seluruh data produk perangkat lunak yang telah digunakan oleh pelanggan.

NO	KOLOM	TIPE DATA	LEBAR
1	Program_id	INT	11
2	Program_id_kat	TINYINT	3
3	Program_key	VARCHAR	50
4	Program_nama	VARCHAR	100
5	Program_versi	VARCHAR	20
6	Program_tanggal	DATE	-
7	Program_expire	DATE	-
8	Program_perusahaan	VARCHAR	100
9	Program_alamat	VARCHAR	100
10	Program_pj	VARCHAR	100
11	Program_hp	VARCHAR	20
12	Program_harga	DECIMAL	-
13	url	VARCHAR	100
14	Created_by	VARCHAR	20

2) Tabel Kategori

Tabel kategori berfungsi untuk menyimpan data kategori program.

NO	KOLOM	TIPE DATA	LEBAR
1	Kat_id	INT	3
2	Kat_kode	VARCHAR	50
3	Kat_nama	VARCHAR	100
4	Kat_Desk	VARCHAR	255
5	active	ENUM	-

3) Tabel *User*

Tabel *user* berfungsi untuk menyimpan data pengguna yang berhak menggunakan sistem.

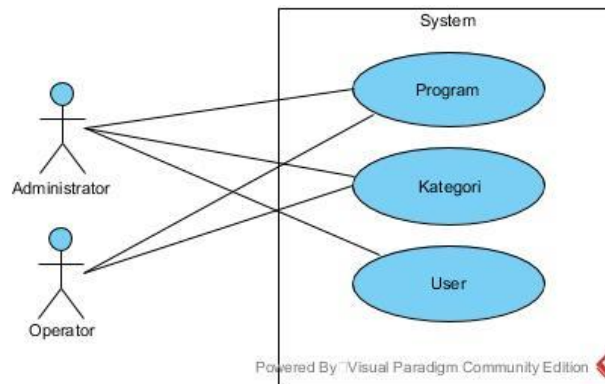
NO	KOLOM	TIPE DATA	LEBAR
1	id	INT	11
2	name	VARCHAR	25
3	email	VARCHAR	50
4	password	VARCHAR	255
5	active	ENUM	-

b. Pemodelan

Pemodelan digunakan agar *programmer* dapat merancang sistem lebih terarah. *Unified Modelling Language* (UML) merupakan bahasa pemodelan yang dapat menjabarkan secara rinci untuk analisa dan mencari apa yang diperlukan oleh sistem (Suendri, 2018).

1) *Use Case Diagram*

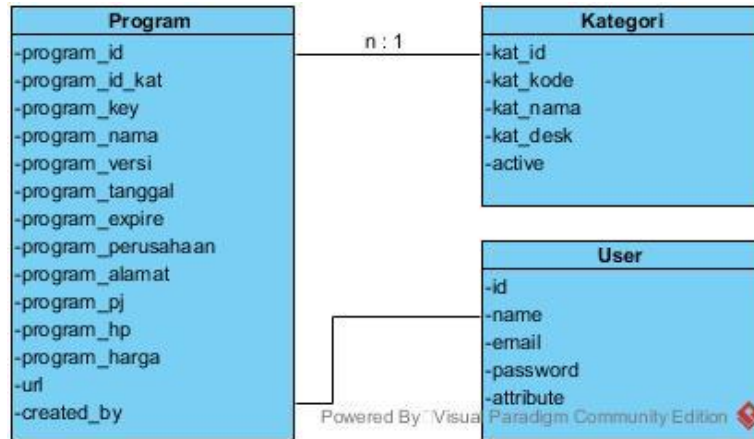
Use Case Diagram menggambarkan tugas dan hak pengguna sistem. Pada sistem ini terdapat 2 tipe pengguna atau *actor* yaitu administrator yang memiliki hak akses penuh terhadap sistem dan operator yang berhak atas menu program dan kategori. *Use Case Diagram* bisa dilihat pada gambar 1 berikut ini.



Gambar 2 Use Case Diagram

2) Class Diagram

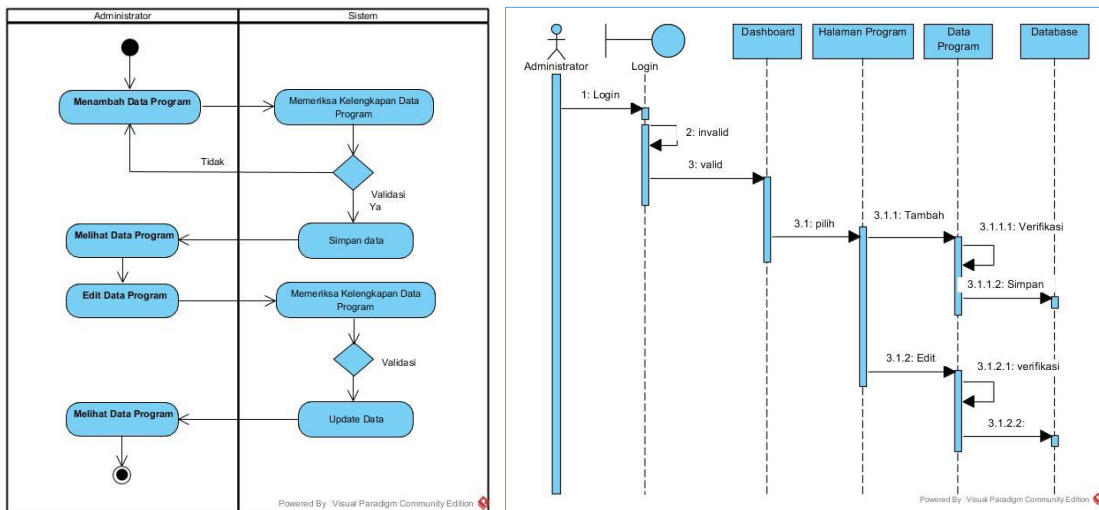
Class Diagram menggambarkan hubungan antar class atau tabel yang terdapat pada sistem. Class Diagram bisa dilihat pada gambar 2 berikut ini.



Gambar 3 Class Diagram

3) Activity Diagram dan Sequence Diagram

Activity Diagram menggambarkan aktivitas pengguna terhadap sistem serta proses yang bisa dilalui pengguna. Sedangkan Sequence Diagram merupakan gambaran kegiatan user berdasarkan rangkaian urutan waktu penggunaan sistem. Activity Diagram dan Sequence Diagram bisa dilihat pada gambar 3 berikut ini.



Gambar 4 Activity Diagram dan Sequence Diagram

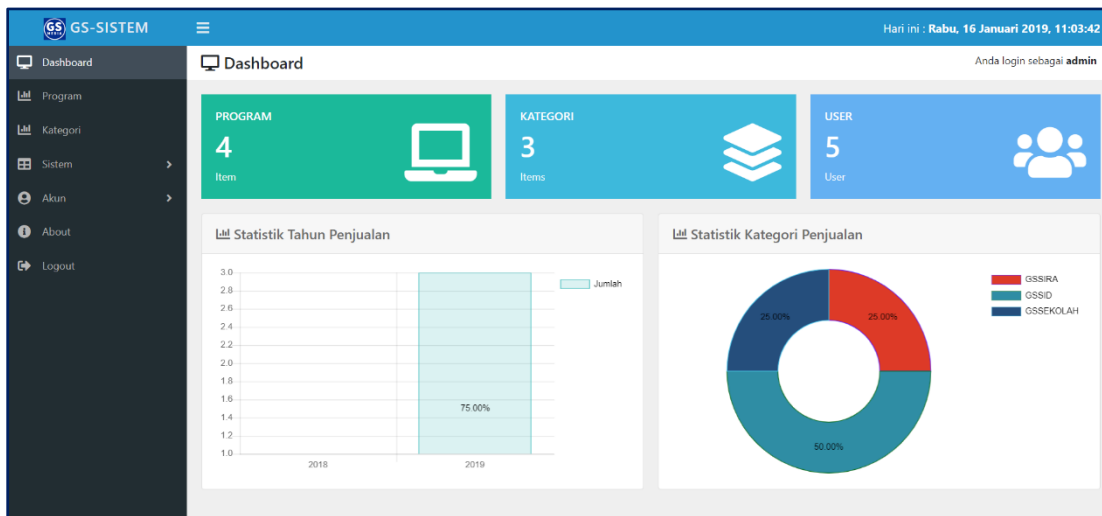
c. Perancangan Sistem

Setelah *database* dan pemodelan telah dibuat selanjutnya proses perancangan sistem. Sistem dikembangkan menggunakan bahasa pemrograman PHP dengan arsitektur MVC. Banyak *framework* dengan arsitektur MVC yang sudah dibuat oleh para *programmer*, namun untuk penelitian ini menggunakan *code* yang dibuat sendiri, dengan memisahkan bagian *Model* untuk *database*, *View* untuk manipulasi tampilan dan *Controller* untuk proses dari bagian *model* ke bagian *view*.

5. IMPLEMENTASI

a. Tampilan *Dashboard*

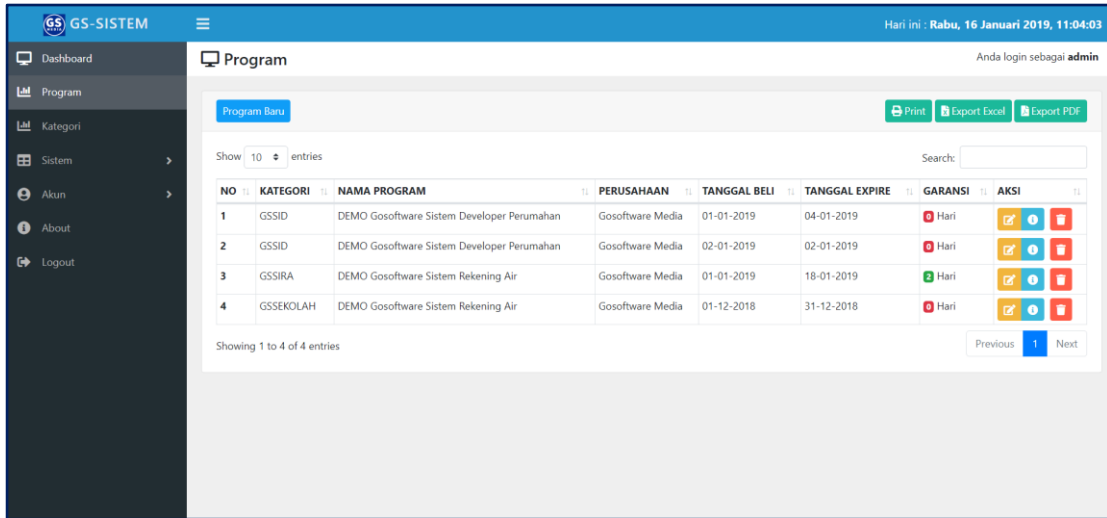
Dashboard merupakan halaman depan sistem, pada halaman ini ditampilkan jumlah program, jumlah kategori dan jumlah pengguna. Pada halaman ini juga ditampilkan statistik program yang telah digunakan oleh pelanggan



Gambar 5 Dashboard

b. Tampilan Menu Program

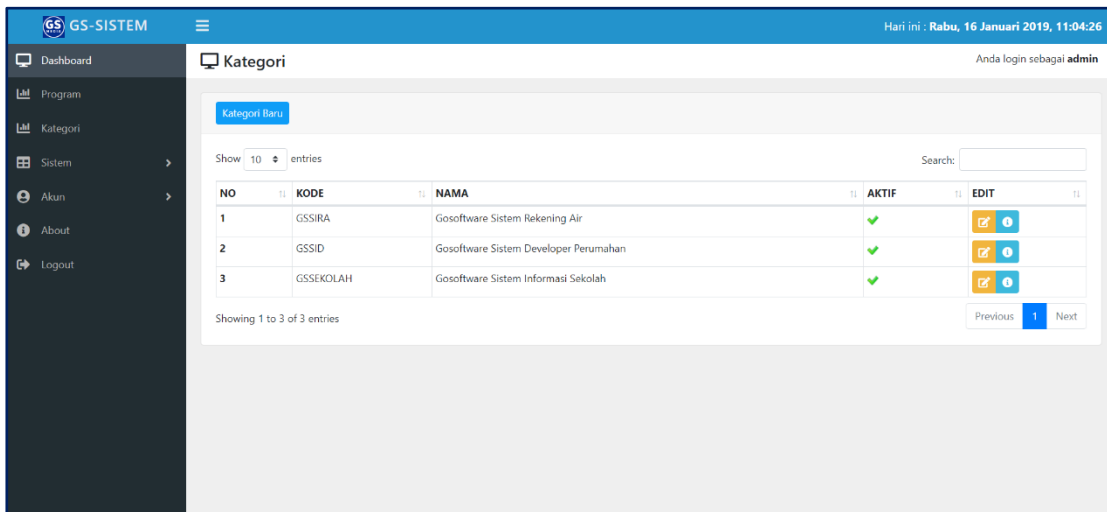
Halaman Program digunakan untuk menampilkan seluruh program yang digunakan oleh pelanggan, seperti data perusahaan, tanggal pembelian dan lama garansi yang diberikan.



Gambar 6 Program

3) Tampilan Menu Kategori

Halaman kategori digunakan untuk menampilkan seluruh kategori program yang ada. Menu ini memudahkan dalam pengelompokan jenis-jenis produk program yang dikembangkan.



Gambar 7 Kategori

6. KESIMPULAN

Konsep atau Arsitektur *Model View Controller* (MVC) lebih memudahkan *programmer* dalam mengatur *query database*, manajemen kode, validasi data dan keamanan. Ada beberapa *framework* yang bisa digunakan oleh *programmer* yang menggunakan bahasa pemrograman PHP seperti Laravel, Zend Framework, Yii, Codeigniter dan lainnya. *Framework* telah dirancang dengan arsitektur MVC dilengkapi dengan *tool-tool* yang memudahkan perancangan sistem dan tentunya dengan tingkat keamanan yang tinggi. Untuk pengembangan sistem ini kedepannya, bisa ditingkatkan dengan menggunakan salah satu *framework* tersebut. Penulis berharap sistem yang telah dirancang ini bisa bermanfaat untuk perusahaan yang bergerak dalam bidang produk perangkat lunak dan memberikan kemudahan dalam mencatat seluruh produk-produk yang telah digunakan oleh pelanggan.

DAFTAR PUSTAKA

- Suendri, S. (2017). Implementasi Algoritma Linear Congruentials Generator Untuk Menentukan Posisi Jabatan Kepanitiaan. *QUERY: JURNAL SISTEM INFORMASI*, 01(02), 15–22. Retrieved from <http://jurnal.uinsu.ac.id/index.php/query/article/view/1043>
- Rashidah F.Olanrewaju, Thouhedul Islam dan Nor'ashikin Bte. Ali. 2014. AN EMPIRICAL STUDY OF THE EVOLUTION OF PHP MVC FRAMEWORK. Retrieved from https://link.springer.com/chapter/10.1007%2F978-3-319-07674-4_40
- Ms. Namrata Jadhav, Prof. Vaishali Londhe. 2014. Lightweight MVC Framework in PHP. *International Research Journal of Engineering and Technology (IRJET)* Volume: 05 Issue: 03 | Mar-2018
- Fathiah, 2013. Studi Dan Penerapan Framework Mvc Dan Ajax Pada Pengembangan Sistem Pendaftaran Mahasiswa Baru Online. *Jurnal Ilmiah STMIK U'Budiyah*, pp.84-70.

- Hartomo, K.D., 2009. Implementasi Model View Controller dan Object Relational Mapping pada Content Management System Sistem Informasi Keuangan. *Jurnal Teknologi Informasi-Aiti*, pp.31-44.
- Hidayat, A., 2012. Penerapan Arsitektur Model View Controller (MVC) Dalam Rancang Bangun Sistem Kuis Online adaptif. In *Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012)*. Yogyakarta, 2012.
- Arochman, Arief Soma Darmwan, Fx Heru Santoso. 2016. PENERAPAN MODEL DESIGN-PATTERN MVC UNTUK EFEKTIFITAS DAN EFISIENSI PENGEMBANGAN WEB KANTOR RISTEKIN KOTA PEKALONGAN. Retrieved from <https://jurnal.stmik-wp.ac.id/gdl.php?mod=browse&op=read&id=ictech--arochmanar-84>
- Hasyrif SY, Rismayani. 2016. Penerapan Konsep MVC Pada Aplikasi Web Menggunakan Framework Laravel. *PROSIDING SEMINAR ILMIAH SISTEM INFORMASI DAN TEKNOLOGI INFORMASI Vol. V, No. 2 Agustus 2016* : 174 – 183
- Chanchai Supaartagorn. 2011. PHP FRAMEWORK FOR DATABASE MANAGEMENT BASED ON MVC PATTERN. *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 3, No 2, April 2011. DOI : 10.5121/ijcsit.2011.3219
- Suendri. 2018. Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: Uin Sumatera Utara Medan). Retrieved from <http://jurnal.uinsu.ac.id/index.php/algorithm/article/view/3148>