Penerapan Konsep *Model View Controller* Pada Sistem Informasi Manajemen Data Terintegrasi

Rino Ramadan[1,*]

[1] Sistem Informasi; Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Kwitang, Kec. Senen, Kota Jakarta Pusat, DKI Jakarta, 10420, (021)54376399; e-mail: rino.rim@bsi.ac.id.

* Korespondensi: e-mail: rino.rim@bsi.ac.id

Diterima: 08 Desember 2020; Review:10 Desember 2020; Disetujui:23 Desember 2020

Cara sitasi: Ramadan R. 2020. Judul Artikel Ilmiah. Penerapan Konsep *Model View Controller* Pada Sistem Informasi Manajemen Data Terintegrasi. Information Management for Educators and Professionals. 5 (1): 45-54.

Abstrak: Manajemen data terintegrasi merupakan sebuah pendekatan untuk memfasilitasi manajemen data dan meningkatkan kinerja. Biasanya manajemen data terintegrasi terdiri dari lingkungan modular yang terintegrasi untuk mengelola data aplikasi perusahaan, dan mengoptimalkan aplikasi berbasis data selama masa pakainya. Saat ini manajemen data yang ada masih kurang konsisten dalam melakukan capaian target tingkat pengelolaan, beberapa operasi masih perlu di otomatiskan dan di sederhanakan, diperlukannya dukungan manajemen data untuk pertumbuhan, dibutuhkannya fasilitas penyelarasan serta konsistensi tata kelola data. Tujuan penelitian ini adalah membangun sebuah sistem informasi manajemen data terintegrasi berbasis website untuk membantu staff dalam mengelola data penerimaan mahasiswa baru ubsi secara terintegrasi. Pembangunan Sistem berbasis web ini menggunakan bahasa pemrograman PHP dan database MySQL, dengan menerapkan konsep Model View Controller (MVC) dalam teknik pemrogramannya. Penelitian ini menghasilkan sebuah aplikasi sistem informasi manajemen data terintegrasi berbasis web, yang mengintegrasikan data terhadap aplikasi lainnya agar siklus hidup aplikasi menjadi lebih praktis dan efisien. Penelitian ini memberikan arahan yang jelas tentang bagaimana sebaiknya membangun sebuah sistem informasi berbasis web dengan baik, sehingga tidak menyulitkan pemrogram web saat harus memperbaiki atau mengembangkan sistem di kemudian hari.

Kata kunci: Data terintegrasi, Konsep model view controller, Sistem manajemen.

Abstract: *Integrated data management is an approach to facilitate data management and improve performance. Typically integrated data management consists of an integrated modular environment for managing enterprise application data, and optimizing data-driven applications over their lifetime. Currently, existing data management is still inconsistent in achieving management level targets, several operations still need to be automated and simplified, data management support is needed for growth, alignment facilities and consistency of data governance are needed. The purpose of this research is to build an integrated data management information system based on a website to assist staff in managing the data of new student admissions in an integrated way. This web-based system development uses the PHP programming language and MySQL database, by applying the Model View Controller (MVC) concept in its programming techniques. This research produces a web-based integrated data management information system application, which integrates data with other applications so that the application life cycle becomes more practical and efficient. This study provides clear directions on how best to build a web-based information system properly, so that it doesn't make it difficult for web programmers to repair or develop the system at a later date.*

Keywords*: Integrated data, Management system, Model View Controller Concept.*

1. Pendahuluan

UBSI atau Universitas Bina Sarana Informatika adalah salah satu perguruan tinggi yang selalu mengadakan event penerimaan mahasiswa baru, biasanya data tersebut berjumlah 10 ribu lebih dalam sekali penerimaan. Data penerimaan mahasiswa baru tersebut perlu dilakukan manajemen data terintegrasi. Manajemen data terintegrasi merupakan sebuah pendekatan untuk memfasilitasi manajemen data dan meningkatkan kinerja. Biasanya manajemen data terintegrasi terdiri dari lingkungan modular yang terintegrasi untuk mengelola data aplikasi perusahaan, dan mengoptimalkan aplikasi berbasis data selama masa pakainya.

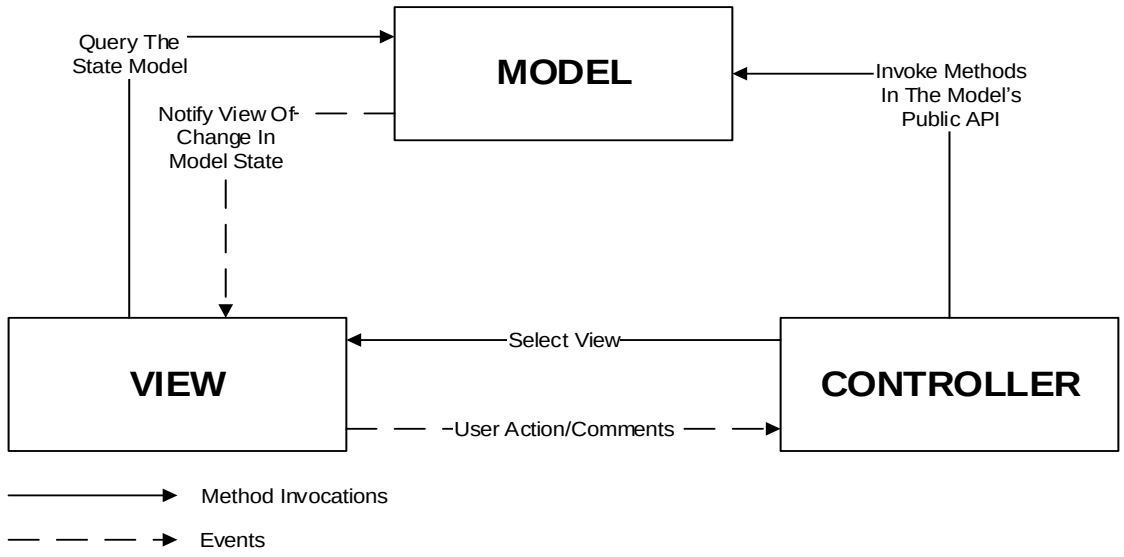
Penelitian ini menjelaskan tentang pembangunan sistem informasi manajemen data terintegrasi berbasis website untuk mengatasi permasalahan dalam mengelola data operasional penerimaan mahasiswa baru. Aplikasi web dibangun menggunakan bahasa pemrogramam PHP dan database MySQL, dengan konsep pemrograman Model View Controller (MVC).

Gaya seorang programmer biasanya berbeda-beda dalam membangun sebuah aplikasi, hal tersebut menyebabkan programmer lain mengalami kesulitan untuk melanjutkan pekerjaannya, sehingga ketika programmer lama berhenti, maka programmer baru harus mempelajari alur logika script program dari awal. Bahkan mungkin programmer baru lebih suka membangun ulang sebuah aplikasi dibanding harus melanjutkan pembangunan aplikasi dari programmer lainnya.

Penerapan konsep MVC dalam pembangunan sistem informasi manajemen data terintegrasi bertujuan untuk memberikan kesederhanaan dan kemudahan bagi programmer web dalam pemeliharaan sistem, karena memisahkan data (model) dari tampilannya (view) dan cara bagaimana mengolahnya (controller), sehingga tidak menyulitkan saat harus memperbaiki atau mengembangkan sistem di masa yang akan datang.

2. Metode Penelitian

Sistem informasi manajemen data terintegrasi berbasis website ini dirancang dengan konsep arsitektur MVC atau Model View Controller. Model, view dan controller sangat erat terkait, oleh karena itu, mereka harus merujuk satu sama lain. Gulzar menjelaskan dalam penelitian dayat dan anggriani sebagaimana gambar 1 [1].

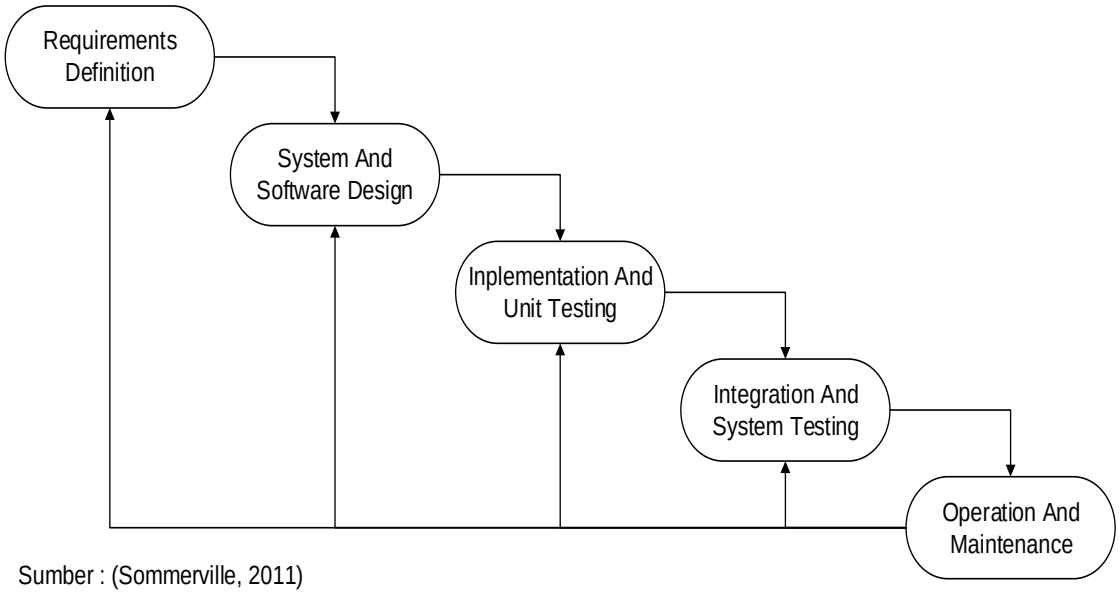


Sumber: (Gulzar, 2002)

Gambar 1. Hubungan antara model, view, dan controller.

Selanjutnya untuk metodologi menggunakan SDLC dengan model pengembangan sistem waterfall. Waterfall adalah sebuah proses dasar seperti spesifikasi, pengembangan, validasi, evolusi dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya [2], yang disajikan pada gambar 2.

Vol. 5, No. 1, Desember 2020, 45-54



Sumber : (Sommerville, 2011)

Gambar 2. Fase-fase dalam Waterfall Model

Requirements analysis and definition merupakan proses mengumpulkan kebutuhan secara komprehensif, lalu dilakukan analisa dan didefinisikan kebutuhan yang harus dipenuhi oleh program nanti. *System and software design* merupakan proses desain sistem, dikerjakan setelah kebutuhan selesai dan dikumpulkan secara lengkap. *Implementation and unit testing* merupakan proses desain program, kemudian diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan program yang dibuat lalu dilakukan pengujian secara per modul. *Integration and system testing* merupakan proses penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing). *Operation and maintenance* merupakan proses mengoperasionalkan program aplikasi dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

3. Hasil dan Pembahasan

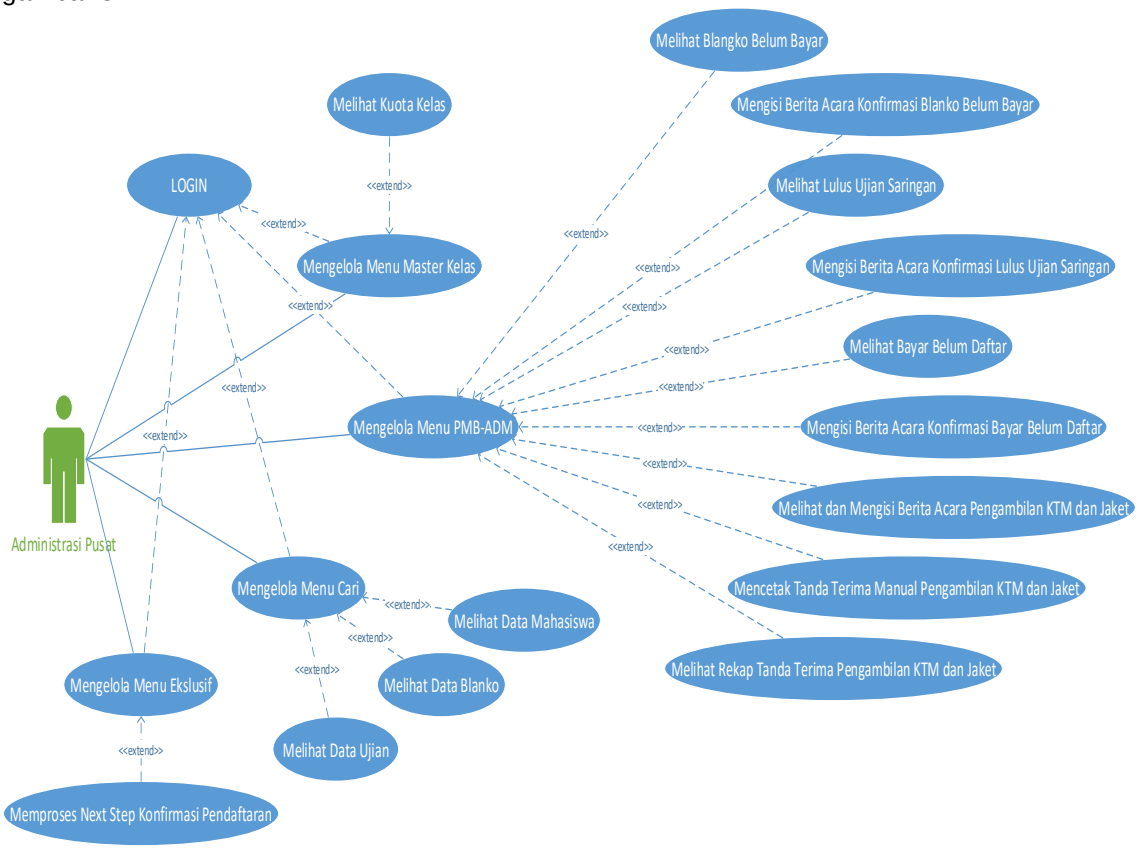
Berdasarkan hasil pengamatan pada proses manajemen data penerimaan mahasiswa baru di Universitas Bina Sarana Informatika, terdapat beberapa kebutuhan pengguna dari sisi administrasi, baik staff administrasi kampus sampai staff administrasi yang berada di kantor pusat terhadap sistem informasi manajemen data yang didefinisikan sebagai berikut: 1). Staff Administrasi Cabang dapat melihat data kuota kelas yang di buka di cabang masing-masing, data pendaftaran calon mahasiswa yang sudah mengisi blanko pendaftaran tetapi belum melakukan pembayaran formulir di cabang masing-masing, data calon mahasiswa yang telah berhasil lulus ujian saringan masuk di cabang masing-masing, data calon mahasiswa yang sudah melakukan pembayaran daftar ulang kuliah tetapi belum melakukan daftar ulang di cabang masing-masing dan mengelola proses pengambilan ktm dan jaket oleh mahasiswa baru di cabang masing-masing. 2). Staff Administrasi Pusat dapat melihat data kuota kelas yang di buka di seluruh cabang, data pendaftaran calon mahasiswa yang sudah mengisi blanko pendaftaran tetapi belum melakukan pembayaran formulir di seluruh cabang, data calon mahasiswa yang telah berhasil lulus ujian saringan masuk di seluruh cabang, data calon mahasiswa yang sudah melakukan pembayaran daftar ulang kuliah tetapi belum melakukan daftar ulang di seluruh cabang, mengelola proses pengambilan ktm dan jaket oleh mahasiswa baru di seluruh cabang, mencari data mahasiswa baru, mencari data blanko pendaftaran, mencari data ujian saringan, dan melakukan proses pengembalian pendaftaran ke tahap next konfirmasi pendaftaran.

Desain Sistem

Unified Modeling Language selanjutnya disebut (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk

pendokumentasian dan melakukan spesifikasi sistem [3]. Dalam penelitian kali ini, penulis menggunakan 3 bahasa grafis pengembangan sistem dalam proses pendokumentasian.

Use Case Diagram, menurut ade hendini dalam penelitiannya, *use case diagram* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [4], Dalam hal ini disajikan pada gambar 3.



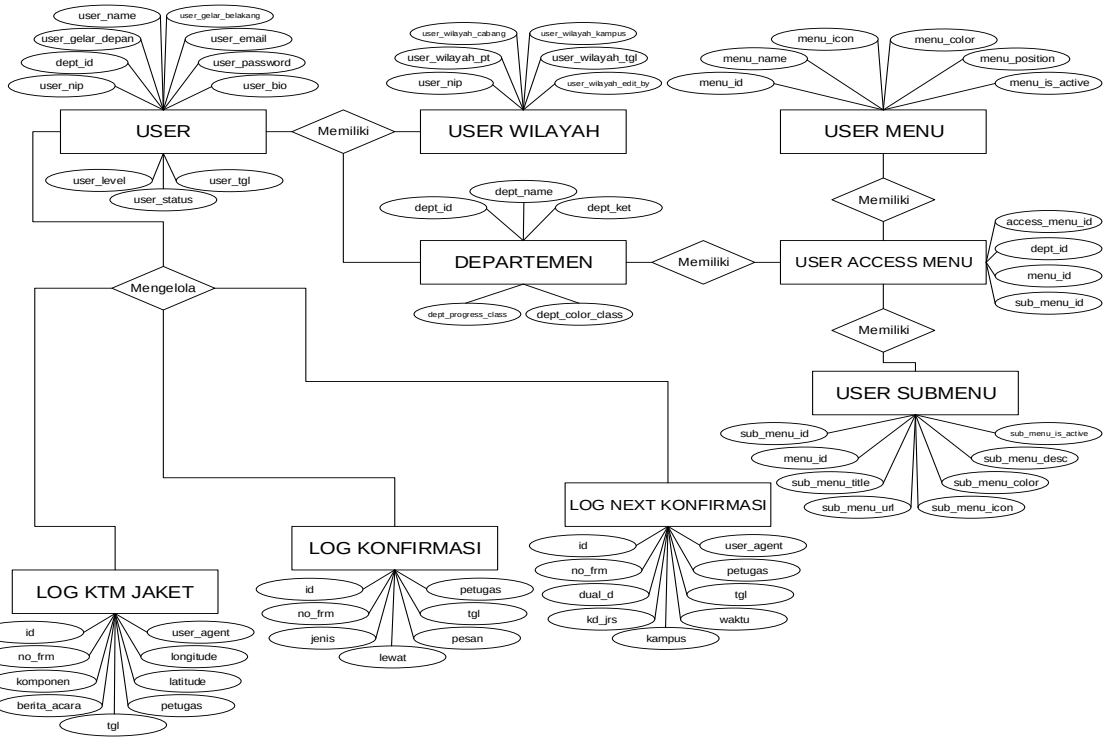
Sumber: Hasil penelitian (2020)

Gambar 3. *Use Case Diagram* Administrator Pusat.

Activity Diagram, menurut rachman dalam penelitiannya, *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak [5].

Class Diagram menurut wira, putra dan andriani dalam penelitiannya, *class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai [6].

Entity Relationship Diagram (*Diagram E-R*), menurut Fathansyah dalam penelitian sukmaindrayana dan sidik menyimpulkan bahwa, *Entity Relationship Diagram* (*Diagram E-R*) adalah yang digunakan untuk menggambarkan *model Entity Relationship* yang berisi komponen-komponen. Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempersentasikan seluruh fakta dari dunia nyata yang kita tinjau [7], Dalam hal ini disajikan pada gambar 4.

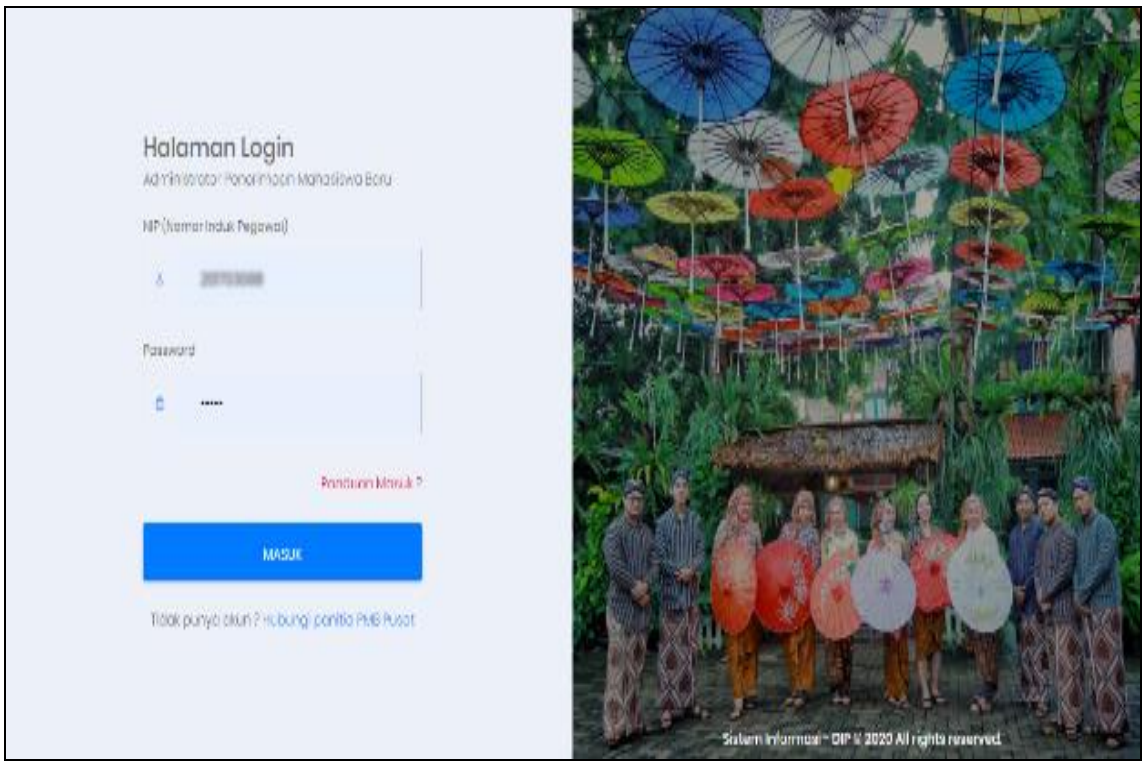


Sumber: Hasil penelitian (2020)

Gambar 4. *Entity Relationship Diagram*.

Implementasi Tampilan

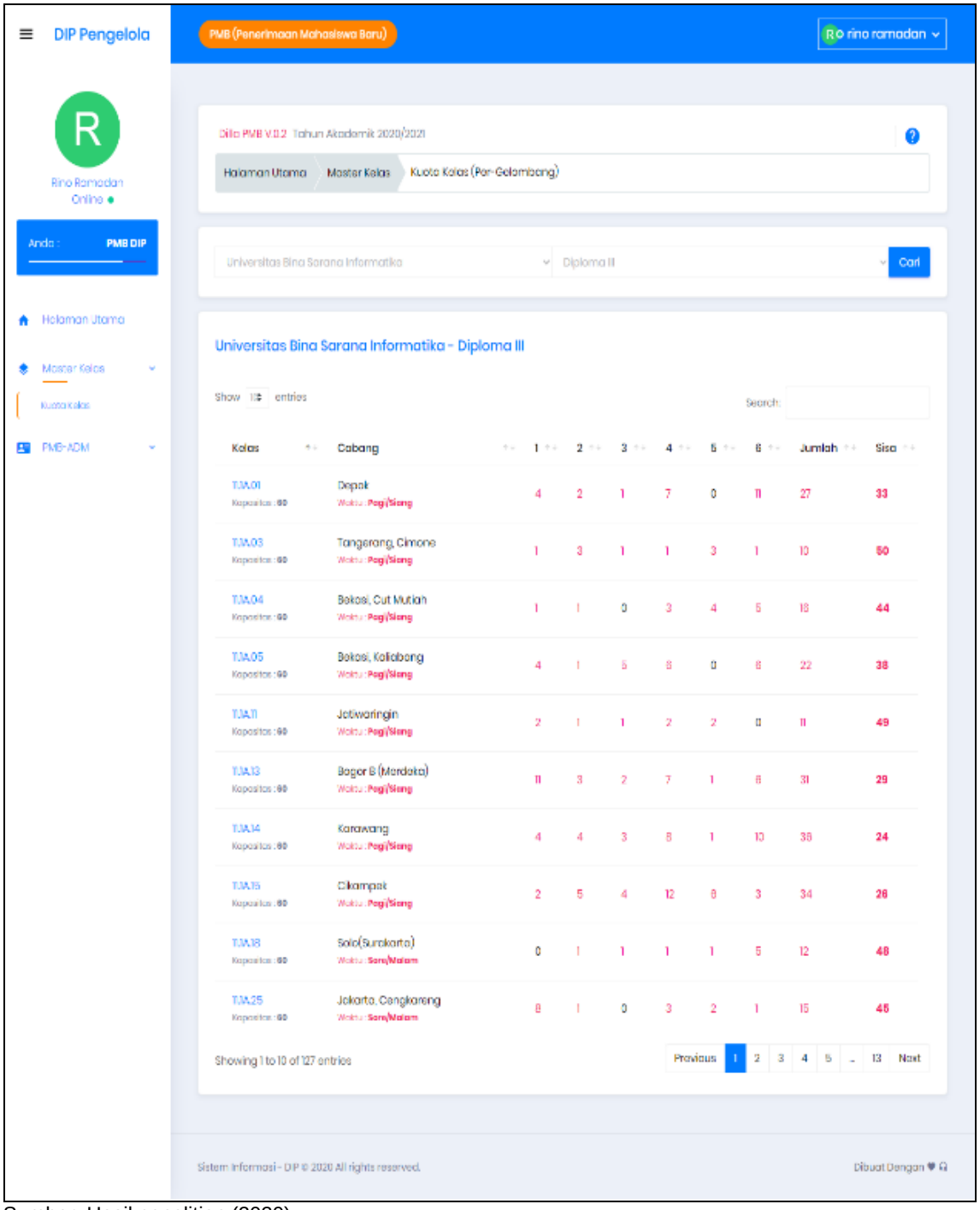
Gambar 5 sampai dengan gambar 6, menunjukan tampilan antar muka aplikasi manajemen data terintegrasi.



Sumber: Hasil penelitian (2020)

Gambar 5. Tampilan Login.

Berikut adalah tampilan halaman kuota kelas serta terdapat tombol cari untuk memudahkan proses pencarian data.



Sumber: Hasil penelitian (2020)

Gambar 6. Tampilan Kuota Kelas.

Implementasi Arsitektur *MVC*

Arsitektur *MVC* diimplementasikan di dalam sistem ini, sistem ini terdiri dari beberapa modul, tiap-tiap modul terdiri dari komponen-komponen utama di dalam aplikasi seperti bagian yang memanipulasi data (*model*), bagian *user interface* (*view*) serta bagian control (*controller*).

Vol. 5, No. 1, Desember 2020, 45-54

Model

Di dalam sistem ini arsitektur *MVC Model* digunakan untuk mengelola aksi dan interkoneksi terhadap basis data, semua kebutuhan *user interface* yang disampaikan ke bagian *controller* akan diproses di sini. *Function* get_kuota_kelas merupakan fungsi yang berisikan perintah untuk untuk mengambil data dari tabel v_kuota_kelas_transform_ubsi dan hasilnya di konversi menjadi *json*. Sedangkan *function* get_kuota_kelas_detail merupakan fungsi yang berisikan perintah untuk mengambil data dari tabel mhs dengan memasukan parameter berupa kode_kelas dan periode pendaftaran. Berikut adalah penggalan script yang menampilkan data pasien pada bagian antarmuka seperti yang ditunjukkan dalam tulisan dibawah.

```
<?php
        class Kuota_kelas_model extends CI_Model{

                public function get_kuota_kelas($pt,$dual_d){
                        $user = $this->session->userdata('user_nip');
                        $wh =array();
                        $SQL ='SELECT * FROM v_kuota_kelas_transform_'.$pt;
                        $pembatas_level = array('6','7');
                        $pt = ($pt=="ubsi")?"1":"2";

                        if(in_array($this->session->userdata('user_level'), $pembatas_level)){
                                $wh[] = "kd_kampus IN (SELECT user_wilayah_kampus FROM
pmbbsi_dilla.dip_user_wilayah WHERE user_nip='".$user."' AND user_wilayah_pt=".$pt.")";
                        }
                        $wh[] = "dual_d='".$dual_d."'";
                        if(count($wh)>0)
                        {
                                $WHERE = implode(' and ',$wh);
                                return $this->datatable->LoadJson($SQL,$WHERE);
                        }
                        else
                        {
                                return $this->datatable->LoadJson($SQL);
                        }
                }

                public function get_kuota_kelas_detail($db,$kd_kelas,$prd){
                        $wh =array();
                        $SQL ='SELECT * FROM '.$db.'.mhs';
                        $pembatas_level = array('6','7');
                        $wh[] = "kd_lokal='".$kd_kelas."'";
                        $wh[] = "prd<>'".$prd."-1'";
                        if(count($wh)>0)
                        {
                                $WHERE = implode(' and ',$wh);
                                return $this->datatable->LoadJson($SQL,$WHERE);
                        }
                        else
                        {
                                return $this->datatable->LoadJson($SQL);
                        }
                }
        }

?>
```

Sumber: Hasil penelitian (2020)

Gambar 7. Script Kode Kuota Kelas Model.

Controller

Arsitektur *MVC Controller* digunakan sebagai penghubung antar *view* dengan *model*. Setiap perintah yang terdapat di dalam *controller* merupakan perintah pemrograman *PHP*

berorientasi obyek, yang di dalamnya terdapat fungsi-fungsi dan prosedur-prosedur yang mengatur alur perintah permintaan data dari *view* kepada *model*.

View

View merupakan arsitektur *MVC* yang berfungsi sebagai antarmuka pengguna, semua informasi yang diperoleh dari hasil proses pada bagian *controller* yang didapat dari model ditampilkan pada bagian ini. Pada *view script php* di dukung oleh bahasa pemrograman lain seperti *css (cascading style sheet)* serta *javascript* agar tampilan lebih menarik.

Pengujian Sistem

Menurut zweben dalam penelitian babaeian, tujuan dari pengujian perangkat lunak adalah untuk mengetahui apakah perangkat lunak tersebut benar atau tidak [8]. Tujuan dilakukannya pengujian sistem informasi manajemen data terintegrasi berbasis *website* adalah untuk mengetahui kesesuaian antara modul sistem yang sudah di implementasikan dengan kebutuhan yang sudah di definisikan sebelumnya pada tahap analisa kebutuhan sistem. Pengujian dilakukan dengan beberapa *web browser* yang umum digunakan, untuk memeriksa setiap kesalahan baik dari proses *coding* (logika maupun sintaks) sampai keluaran program, sehingga dapat segera diperbaiki segala kekurangannya.

Pengujian sistem informasi manajemen data terintegrasi berbasis *website* menggunakan metode *black box testing*. Menurut mustaqbal dalam penelitian cholifah, *black box testing* pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode *black box testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang *valid* [9].

Menurut ammann dalam penelitian jaya, keuntungan penggunaan metode *black box testing* adalah : (1) penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; (3) *programmer* dan tester keduanya saling bergantung satu sama lain. Kekurangan dari metode *Blackbox Testing* adalah : (1) uji kasus sulit disain tanpa spesifikasi yang jelas; (2) kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*; (3) beberapa bagian *back end* tidak diuji sama sekali [10].

Pengujian *black box* dilakukan dengan cara memberi input dari pengguna kepada sistem yang sudah berjalan kemudian mengamati hasil *output* dari sistem. Pengujian dilakukan pada setiap *use case* untuk mengetahui kesesuaian fungsi sistem, dengan prosedur sebagai berikut : (1) Menentukan data yang akan digunakan untuk keperluan pengujian perangkat lunak. (2) Menentukan metode pengujian yang akan digunakan. (3) Melakukan pengujian pada setiap *use case* menggunakan data yang sudah disiapkan dan membandingkan hasilnya dengan kriteria pengujian, seperti dalam tabel 1 berikut ini :

Tabel 1. Daftar *Use Case* dan Kriteria Evaluasi Hasil Pengujian

ID Kasus Uji	Use Case	Kriteria Evaluasi Hasil	Pengujian Kesimpulan
UC-1	Mengelola data kuota kelas	Sistem dapat menampilkan data kuota kelas untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-2	Mengelola data blanko belum bayar	Sistem dapat menampilkan dan menambah konfirmasi data blanko belum bayar untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid

ID Kasus Uji	Use Case	Kriteria Evaluasi Hasil	Pengujian Kesimpulan
UC-3	Mengelola data lulus ujian saringan	Sistem dapat menampilkan dan menambah konfirmasi data lulus ujian saringan untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-4	Mengelola data bayar belum daftar	Sistem dapat menampilkan dan menambah konfirmasi data bayar belum daftar untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-5	Mengelola data ktm dan jaket	Sistem dapat menambah berita acara dan menampilkan data ktm dan jaket beserta rekap ataupun cetakannya untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-6	Mengelola data mahasiswa	Sistem dapat menampilkan data mahasiswa untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-7	Mengelola data blanko	Sistem dapat menampilkan data blanko untuk semua user yang memiliki hak akses departemen sebagai administrasi cabang dan administrasi pusat sesuai dengan ketentuan	Valid
UC-8	Mengelola data ujian	Sistem dapat menampilkan data ujian untuk semua user yang memiliki hak akses departemen sebagai administrasi pusat	Valid
UC-9	Mengelola data next step konfirmasi pendaftaran	Sistem dapat menampilkan, memproses, dan menambahkan data next step konfirmasi untuk semua user yang memiliki hak akses departemen sebagai administrasi pusat sesuai dengan ketentuan	Valid

Sumber: Hasil penelitian (2020)

4. Kesimpulan

Penggunaan sistem informasi manajemen data terintegrasi berbasis website mampu menjawab kebutuhan para pengguna dalam melakukan proses manajemen data. Sistem yang dibuat telah mampu memberikan kesederhanaan dan kemudahan bagi programmer website dalam pemeliharaan sistem, karena memisahkan data (*model*) dari tampilannya (*view*) dan cara bagaimana mengolahnya (*controller*), sehingga programmer baru tidak akan merasa sulit saat harus memperbaiki atau mengembangkan sistem di masa yang akan datang.

Referensi

[1] A. R. Dayat and L. Angriani, “Pemanfaatan Model-View-Controller (MVC) Dalam Rancang Bangun Sistem Informasi Rakornas Aptikom 2017,” *Semin. Nas. APTIKOM*, no. November, pp. 416–420, 2017.
[2] Sommerville, *Software Engineering (9th ed.)*. Massachusetts: Addison-Wesley, 2011.
[3] S. Mulyani, *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sistematika, 2016.
[4] A. Hendini, “Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang,” *J. Khatulistiwa Inform.*, vol. 2, no. 9, pp. 107–116, 2016.
[5] Rachman, “Sistem Informasi Wisata Di Ampera Waterpark,” *J. Siliwangi*, vol. 4, no. 2, pp. 87–92, 2018.
[6] D. Wira, T. Putra, and R. Andriani, “Unified Modelling Language (UML) dalam

Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," vol. 7, no. 1, 2019.

[7] A. Sukmaindrayana and R. Sidik, "Aplikasi Grosir Pada Toko RSIDIK Bungursari Tasimalaya," *J. Manaj. Inform.*, vol. 4, no. 2, pp. 31–40, 2017.

[8] M. Babaeian, V. Ghasemiyan, and R. Nourmandi-pour, "Comparison of software testing review Black Box and White Box and Gray Box," *Int. J. Math. Comput. Sci.*, vol. 43, pp. 1442–1447, 2015.

[9] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018.

[10] T. S. Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *J. Inform. Pengemb. IT*, vol. 3, no. 2, pp. 45–46, 2018.