**(2.Ado.Ok**

**UNDERGRADUATE THESIS – KI141502**

**DESIGN AND IMPLEMENTATION OF TRAVEL INFORMATION SYSTEM IN FIJI USING REST-API CONCEPT**

NEMESIO R RAITUBU

NRP 05111640007006

Thesis Supervisor I

Radityo Anggoro, S.Kom., M.Sc.

INFORMATICS DEPARTMENT

Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember

Surabaya 2020



**UNDERGRADUATE THESIS – KI141502**

**DESIGN AND IMPLEMENTATION OF TRAVEL INFORMATION SYSTEM IN FIJI USING REST-API CONCEPT**

**NEMESIO R RAITUBU**

**NRP 05111640007006**

**Thesis Supervisor I**

**Radityo Anggoro, S.Kom., M.Sc.**

**INFORMATICS DEPARTMENT**

**Faculty of Intelligent Electrical and Informatics Technology**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2020**

[This page has been intentionally left blank]

**APPROVAL SHEET**

**DESIGN AND IMPLEMENTATION OF TRAVEL INFORMATION SYSTEM IN FIJI USING REST-API CONCEPT**

**UNDERGRADUATE THESIS**Submitted to Meet One Provision  
Obtaining a Bachelor of Computer Degree  
on  
Undergraduate Program of Informatics Department  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember

By :  
**NEMESIO R RAITUBU**NRP : 05111640007006

Approved by Thesis’s Supervisor :

|  |  |
| --- | --- |
| Radityo Anggoro, S.Kom., M.Sc.  NIP: 19841016 2008121002 | ........................... (Supervisor 1) |

**Surabaya  
APRIL 2020**

[This page has been intentionally left blank]

**DESIGN AND IMPLEMENTATION OF TRAVEL INFORMATION SYSTEM IN FIJI USING REST-API CONCEPT**

Name : NEMESIO R RAITUBU

NRP : 05111640007006

Major : Informatics Department – ITS

Supervisor I : Radityo Anggoro, S.Kom., M.Sc.

**Abstract**

Nearly everyone goes on a vacation and a Travel information system would play a vital role in planning the perfect trip. The travel information system allows the user of the system access all the details such as hotels and flights. The main purpose is to help individual to manage customer, flights and hotels . The system can also be used for both professional and business trips. The proposed system maintains centralized repository to make necessary travel arrangements and to retrieve information easily.

With the rapid economic development in Fiji, the government has invest a lot of money into the tourism industry for the construction of many hotels to attract tourists to choose Fiji as their holiday destination. Therefore, Travel Information System in Fiji will be very useful for the tourists to use in booking flights and hotels.

**Keywords: Information System, Web, Travel, VueJs, REST-API, Laravel.**

[This page has been intentionally left blank]

**ACKNOWLEDGMENT**

Writer gives Praise and gratitude to the Almighty God for the gracious mercy and tremendous blessing that enables the writer to accomplish the final project entitled ““Design and Implementation of Travel Information System in Fiji”.

On the process of doing this project, I am fully aware that there are many parties involved in supporting this project to be a success. Therefore. I would like to express special appreciation and gratitude to the followings:

1. Writer’s parents, brothers, sisters and relatives that always gives moral and spiritual support during my bachelor study.
2. Mr. Radityo Anggoro, S.Kom., M.Sc., as my supervisor who always give great help, advice. And taught me in order to accomplish this final project.
3. Bapak Rizky Januar Akbar, S.Kom., M.Eng. as a supportive lecturer who has helped, guided, and provided his knowledge to the completing of this Final Project.
4. Darlis Herumurti, S.Kom., M.Sc., as the Head of Informatics Department and also mu guardian lecturer which he helps me and guarded me through the whole of my studies and enabling the completion of this final project.
5. Adetiya Bagus class of 2014 was a great and supportive friend of mine in doing this final project.
6. Write would like to acknowledge my friends’ class of 2014 for making each and everyday class a memorable one and their tremendous support in this piece of project.
7. Writer would like take this opportunity to give thanks and appreciation to all the hardworking lecturers of Informatics Engineering Department who have taught me, and support me in one way or the other during my bachelor program.
8. The writer also would like to say a special thanks to the Head of International Office ITS and all her hardworking staffs and volunteers who support me during the time of my bachelor program
9. Last but not the least, the writer would like to take this time to say thank you to ITS Rector, all the ITS staffs, and the Indonesian Government for giving me the opportunity to study in ITS.
10. Everyone else that cannot be mention one by one in making this research a success, may God repay all the good deeds.

The writer also fully aware that this report still misses a lot of part. Therefore, any critics and suggestion regarding this report is welcome. Hopefully this final project report can give benefit to the readers and please forgive me if there any unpleasant words and mistake during the process of making this final report.

Surabaya, April 2020

Writer

# TABLE OF CONTENTS

[Abstract v](#_Toc36611912)

[ACKNOWLEDGMENT vii](#_Toc36611913)

[TABLE OF CONTENTS ix](#_Toc36611914)

[DAFTAR GAMBAR xii](#_Toc36611915)

[DAFTAR TABEL xvi](#_Toc36611916)

[1 CHAPTER I INTRODUCTION 19](#_Toc36611917)

[1.1. Background 19](#_Toc36611918)

[1.2. Problem Formulation 20](#_Toc36611919)

[1.3. Problem Limitation 20](#_Toc36611920)

[1.4. Goals 20](#_Toc36611921)

[1.5. Methodology 21](#_Toc36611922)

[1.6. Sistematika Penulisan 22](#_Toc36611923)

[2 CHAPTER II LITERATURE REVIEW 24](#_Toc36611924)

[2.1. Pembelajaran 24](#_Toc36611927)

[2.2. IPD 24](#_Toc36611928)

[2.3. Sistem Informasi Akademik Generik 25](#_Toc36611929)

[2.4. Phalcon PHP 25](#_Toc36611930)

[2.5. Microsoft SQL Server 26](#_Toc36611931)

[2.6. Repository-Service Pattern 27](#_Toc36611932)

[2.7. MVC (Model-View-Controller) 28](#_Toc36611933)

[3 CHAPTER III ANALYSIS AND DESIGN 30](#_Toc36611934)

[3.1. Analysis 30](#_Toc36611936)

[3.1.1. Software Requirements Specifications 31](#_Toc36611937)

[3.1.2. Functional Requirements 31](#_Toc36611938)

[3.1.3. Actors 32](#_Toc36611939)

[3.1.4. Use Case Diagram 32](#_Toc36611940)

[3.2. Design 67](#_Toc36611941)

[3.2.1. Desain Arsitektur dan Pola Perancangan 68](#_Toc36611942)

[3.2.2. Perancangan Diagram Kelas 69](#_Toc36611943)

[3.2.3. Perancangan Basis Data 69](#_Toc36611944)

[3.2.4. Perancangan Antarmuka Modul pembelajaran 71](#_Toc36611945)

[3.2.5. Perancangan Antarmuka Modul IPD 96](#_Toc36611946)

[3.2.6. Proses Bisnis Sistem 103](#_Toc36611947)

[3.2.7. Relasi Antar Modul 106](#_Toc36611948)

[4 CHAPTER IV IMPLEMENTATION 108](#_Toc36611949)

[4.1. Implementation Environment 108](#_Toc36611951)

[4.2. Implementasi Antarmuka Sistem 108](#_Toc36611952)

[4.2.1. Antarmuka Modul Pembelajaran 109](#_Toc36611953)

[4.2.2. Antarmuka Modul IPD 126](#_Toc36611954)

[*4.3.* Implementasi *Repository-Service Pattern* pada Arsitektur *Model-View-Controller* 132](#_Toc36611955)

[5 CHAPTER V TESTING AND EVALUATION 141](#_Toc36611956)

[5.1. Lingkungan Uji Coba 141](#_Toc36611962)

[5.2. Skenario Uji Coba 141](#_Toc36611963)

[5.2.1. Kasus Pengujian Mengelola Kelas Pembelajaran 142](#_Toc36611964)

[5.2.2. Kasus Pengujian Mengelola Pendamping Akademik 144](#_Toc36611965)

[5.2.3. Kasus Pengujian Mengelola Kehadiran Dosen 146](#_Toc36611966)

[5.2.4. Kasus Pengujian Mengelola Kehadiran Mahasiswa 148](#_Toc36611967)

[5.2.5. Kasus Pengujian Menyusun FRS 150](#_Toc36611968)

[5.2.6. Kasus Pengujian Mencetak FRS 153](#_Toc36611969)

[5.2.7. Kasus Pengujian Menyetujui FRS 154](#_Toc36611970)

[5.2.8. Kasus Pengujian Menampilkan Anak Wali 155](#_Toc36611971)

[5.2.9. Kasus Pengujian Menampilkan Kelas Departemen 156](#_Toc36611972)

[5.2.10. Kasus Pengujian Menampilkan Kelas UPMB 157](#_Toc36611973)

[5.2.11. Kasus Pengujian Mengelola Tahun Ajaran 157](#_Toc36611974)

[5.2.12. Kasus Pengujian Mengelola Beban Studi 159](#_Toc36611975)

[5.2.13. Kasus Pengujian Menampilkan Riwayat Pembayaran 161](#_Toc36611976)

[5.2.14. Kasus Pengujian Menampilkan Tagihan 162](#_Toc36611977)

[5.2.15. Kasus Pengujian Mengelola Tanggal Penting FRS 162](#_Toc36611978)

[5.2.16. Kasus Pengujian Mengelola Kuesioner IPD 164](#_Toc36611979)

[5.2.17. Kasus Pengujian Mengisi Kuesioner IPD 166](#_Toc36611980)

[5.2.18. Kasus Pengujian Menampilkan Daftar Kuesioner 167](#_Toc36611981)

[5.2.19. Kasus Pengujian Menampilkan Hasil IPD Dosen 168](#_Toc36611982)

[5.3. Rekapitulasi Kasus Pengujian 168](#_Toc36611983)

[6 CHAPTER VI KESIMPULAN DAN SARAN 173](#_Toc36611984)

[6.1. Kesimpulan 173](#_Toc36611986)

[6.2. Saran 174](#_Toc36611987)

[7 DAFTAR PUSTAKA 175](#_Toc36611988)

[8 LAMPIRAN 177](#_Toc36611989)

[9 BIODATA PENULIS 181](#_Toc36611990)

# LIST OF PICTURES

[Gambar 1.1 Ilustrasi Rancang Bangun Perangkat Lunak 2](#_Toc520464629)

[Gambar 2.1 Diagram Pola Perancangan *Repository-Service* 10](#_Toc520464630)

[Gambar 3.1 Tahap Analisis Kebutuhan Perangkat Lunak 13](#_Toc520464631)

[Gambar 3.2 Kasus Penggunaan Modul Pembelajaran 19](#_Toc520464632)

[Gambar 3.3 Diagram Aktivitas Membuat Kelas 21](#_Toc520464633)

[Gambar 3.4 Diagram Aktivitas Mengubah Kelas 23](#_Toc520464634)

[Gambar 3.5 Diagram Aktivitas Menghapus Kelas 25](#_Toc520464635)

[Gambar 3.6 Diagram Aktivitas Pendamping Akademik 27](#_Toc520464636)

[Gambar 3.7 Diagram Aktivitas Menampilkan Riwayat Perwalian 28](#_Toc520464637)

[Gambar 3.8 Diagram Aktivitas Kehadiran Dosen 30](#_Toc520464638)

[Gambar 3.9 Diagram Aktivitas Kehadiran Mahasiswa 32](#_Toc520464639)

[Gambar 3.10 Diagram Aktivitas Menyusun FRS 34](#_Toc520464640)

[Gambar 3.11 Diagram Aktivitas Mencetak FRS 36](#_Toc520464641)

[Gambar 3.12 Diagram Aktivitas Menyetujui FRS 38](#_Toc520464642)

[Gambar 3.13 Diagram Aktivitas Mengubah FRS 40](#_Toc520464643)

[Gambar 3.14 Diagram Aktivitas Menampilkan Anak Wali 41](#_Toc520464644)

[Gambar 3.15 Diagram Aktivitas Menampilkan Kelas Departemen 43](#_Toc520464645)

[Gambar 3.16 Diagram Aktivitas Menampilkan Kelas UPMB 44](#_Toc520464646)

[Gambar 3.17 Diagram Aktivitas Mengelola Tahun Ajaran 46](#_Toc520464647)

[Gambar 3.18 Diagram Aktivitas Mengelola Beban Studi 48](#_Toc520464648)

[Gambar 3.19 Diagram Aktivitas Menampilkan Riwayat Pembayaran 50](#_Toc520464649)

[Gambar 3.20 Diagram Aktivitas Menampilkan Tagihan 51](#_Toc520464650)

[Gambar 3.21 Diagram Aktivitas Mengelola Tanggal Penting FRS 53](#_Toc520464651)

[Gambar 3.22 Diagram Kasus Penggunaan Modul IPD 54](#_Toc520464652)

[Gambar 3.23 Diagram Aktivitas Mengelola Kuesioner Mata Kuliah 56](#_Toc520464653)

[Gambar 3.24 Diagram Aktivitas Mengelola Kuesioner Dosen 58](#_Toc520464654)

[Gambar 3.25 Diagram Aktivitas Mengisi Kuesioner Mata Kuliah 60](#_Toc520464655)

[Gambar 3.26 Diagram Aktivitas Mengisi Kuesioner Dosen 62](#_Toc520464656)

[Gambar 3.27 Diagram Aktivitas Menampilkan Kuesioner IPD 64](#_Toc520464657)

[Gambar 3.28 Diagram Aktivitas Menampilkan Hasil IPD Dosen 66](#_Toc520464658)

[Gambar 3.29 Diagram Arsitektur MVC dengan pola perancangan *Repository-Service* 67](#_Toc520464659)

[Gambar 3.30 Rancangan Antarmuka Menampilkan Kelas 71](#_Toc520464660)

[Gambar 3.31 Rancangan Antarmuka *Form* Kelas 73](#_Toc520464661)

[Gambar 3.32 Rancangan Antarmuka Mengelola Pendamping Akademik 75](#_Toc520464662)

[Gambar 3.33 Rancangan Antarmuka Mengelola Kehadiran Dosen 76](#_Toc520464663)

[Gambar 3.34 Rancangan Antarmuka *Form* Kehadiran Dosen 78](#_Toc520464664)

[Gambar 3.35 Rancangan Antarmuka Mengelola Kehadiran Mahasiswa 79](#_Toc520464665)

[Gambar 3.36 Rancangan Antarmuka FRS Belum Setuju 81](#_Toc520464666)

[Gambar 3.37 Rancangan Antarmuka FRS Sudah Setuju 82](#_Toc520464667)

[Gambar 3.38 Rancangan Antarmuka Menyetujui FRS 83](#_Toc520464668)

[Gambar 3.39 Rancangan Antarmuka Menampilkan Anak Wali 85](#_Toc520464669)

[Gambar 3.40 Rancangan Antarmuka Menampilkan Kelas Departemen 86](#_Toc520464670)

[Gambar 3.41 Rancangan Antarmuka Menampilkan Kelas UPMB 87](#_Toc520464671)

[Gambar 3.42 Rancangan Antarmuka Mengelola Tahun Ajaran 88](#_Toc520464672)

[Gambar 3.43 Rancangan Antarmuka *Form* Tahun Ajaran 89](#_Toc520464673)

[Gambar 3.44 Rancangan Antarmuka Mengelola Beban Studi 90](#_Toc520464674)

[Gambar 3.45 Rancangan Antarmuka *Form* Beban Studi 91](#_Toc520464675)

[Gambar 3.46 Rancangan Antarmuka Mengelola Tanggal FRS 93](#_Toc520464676)

[Gambar 3.47 Rancangan Antarmuka *Form* Tanggal FRS 94](#_Toc520464677)

[Gambar 3.48 Rancangan Antarmuka Mengelola Kuesioner 96](#_Toc520464678)

[Gambar 3.49 Rancangan Antarmuka *Form* Kelola Kuesioner 97](#_Toc520464679)

[Gambar 3.50 Rancangan Antarmuka Mengisi Kuesioner 99](#_Toc520464680)

[Gambar 3.51 Rancangan Antarmuka Melihat Daftar Kuesioner 100](#_Toc520464681)

[Gambar 3.52 Rancangan Antarmuka Melihat Hasil IPD Dosen 101](#_Toc520464682)

[Gambar 3.53 Proses bisnis modul IPD 103](#_Toc520464683)

[Gambar 3.54 Proses bisnis modul FRS 104](#_Toc520464684)

[Gambar 3.55 Relasi antar modul FRS dan IPD 105](#_Toc520464685)

[Gambar 4.1 Halaman Antarmuka Kelas Pembelajaran 108](#_Toc520464686)

[Gambar 4.2 Halaman Antarmuka *Form* Buat Kelas 109](#_Toc520464687)

[Gambar 4.3 Halaman Antarmuka *Form* Ubah Kelas 110](#_Toc520464688)

[Gambar 4.4 Halaman Antarmuka Kelola Pendamping Akademik 111](#_Toc520464689)

[Gambar 4.5 Halaman Antarmuka Riwayat Perwalian Mahasiswa 111](#_Toc520464690)

[Gambar 4.6 Halaman Antarmuka Kehadiran Dosen 112](#_Toc520464691)

[Gambar 4.7 Halaman Antarmuka *Form* Kehadiran Dosen 113](#_Toc520464692)

[Gambar 4.8 Halaman Antarmuka Kehadiran Mahasiswa 114](#_Toc520464693)

[Gambar 4.9 Halaman Antarmuka FRS Belum Disetujui 115](#_Toc520464694)

[Gambar 4.10 Halaman Antarmuka FRS Telah Disetujui 115](#_Toc520464695)

[Gambar 4.11 Halaman Antarmuka Menyetujui FRS 116](#_Toc520464696)

[Gambar 4.12 Halaman Antarmuka Daftar Anak Wali 117](#_Toc520464697)

[Gambar 4.13 Halaman Antarmuka Daftar Kelas Departemen 118](#_Toc520464698)

[Gambar 4.14 Halaman Antarmuka Daftar Kelas UPMB 119](#_Toc520464699)

[Gambar 4.15 Halaman Antarmuka Tahun Ajaran 120](#_Toc520464700)

[Gambar 4.16 Halaman Antarmuka *Form* Tahun Ajaran 121](#_Toc520464701)

[Gambar 4.17 Halaman Antarmuka Beban Studi 122](#_Toc520464702)

[Gambar 4.18 Halaman Antarmuka Riwayat Pembayaran 123](#_Toc520464703)

[Gambar 4.19 Halaman Antarmuka Tagihan 124](#_Toc520464704)

[Gambar 4.20 Halaman Antarmuka Tanggal Penting FRS 125](#_Toc520464705)

[Gambar 4.21 Halaman Antarmuka *Font* Tanggal Penting FRS 125](#_Toc520464706)

[Gambar 4.22 Halaman Antarmuka Kelola Kuesioner Mata Kuliah 126](#_Toc520464707)

[Gambar 4.23 Halaman Antarmuka Kelola Kuesioner Dosen 127](#_Toc520464708)

[Gambar 4.24 Halaman Antarmuka *Form* Kelola Kuesioner 128](#_Toc520464709)

[Gambar 4.25 Halaman Antarmuka Mengisi Kuesioner IPD 129](#_Toc520464710)

[Gambar 4.26 Halaman Antarmuka Melihat Daftar Kuesioner 130](#_Toc520464711)

[Gambar 4.27 Halaman Antarmuka Melihat Hasil IPD Dosen 131](#_Toc520464712)

[Gambar 4.28 Struktur direktori pada *folder* *modules* 132](#_Toc520464713)

[Gambar 4.29 Struktur *controller* dan *view* untuk tiap modul 133](#_Toc520464714)

[Gambar 4.30 Struktur *service* dan *repository* untuk tiap modul 133](#_Toc520464715)

[Gambar 4.31 Relasi dengan modul kurikulum dengan cara mengambil data mata kuliah 134](#_Toc520464716)

[Gambar 4.32 Relasi dengan modul penjadwalan dengan cara mengambil kelas yang telah diberi jadwal 135](#_Toc520464717)

[Gambar 4.33 Relasi antar modul pembelajaran dan modul IPD 137](#_Toc520464718)

# LIST OF TABLES

[Tabel 3.1 Functional Table of the Software 32](#_Toc36611657)

[Tabel 3.3 Table Actors of System 33](#_Toc36611658)

[Table 3.5 Detail of Create Travel Destination’s Use Case 34](#_Toc36611659)

[Table 3.6 Detail of Update Travel Destination’s Use Case 37](#_Toc36611660)

[Tabel 3.7 Rincian Kasus Penggunaan Menghapus Kelas 39](#_Toc36611661)

[Tabel 3.8 Rincian Kasus Penggunaan Mengelola Pendamping Akademik 41](#_Toc36611662)

[Tabel 3.9 Rincian Kasus Penggunaan Menampilkan Riwayat Perwalian 42](#_Toc36611663)

[Tabel 3.10 Rincian Kasus Penggunaan Mengelola Kehadiran Dosen 44](#_Toc36611664)

[Tabel 3.11 Rincian Kasus Penggunaan Mengelola Kehadiran Mahasiswa 46](#_Toc36611665)

[Tabel 3.12 Rincian Kasus Penggunaan Menyusun FRS 47](#_Toc36611666)

[Tabel 3.13 Rincian Kasus Penggunaan Mencetak FRS 50](#_Toc36611667)

[Tabel 3.14 Rincian Kasus Penggunaan Menyetujui FRS 52](#_Toc36611668)

[Tabel 3.15 Rincian Kasus Penggunaan Mengubah FRS 54](#_Toc36611669)

[Tabel 3.16 Rincian Kasus Penggunaan Menampilkan Anak Wali 55](#_Toc36611670)

[Tabel 3.17 Rincian Kasus Penggunaan Menampilkan Kelas Departemen 57](#_Toc36611671)

[Tabel 3.18 Rincian Kasus Penggunaan Menampilkan Kelas UPMB 58](#_Toc36611672)

[Tabel 3.19 Rincian Kasus Penggunaan Mengelola Tahun Ajaran 60](#_Toc36611673)

[Tabel 3.20 Rincian Kasus Penggunaan Mengelola Beban Studi 62](#_Toc36611674)

[Tabel 3.21 Rincian Kasus Penggunaan Menampilkan Riwayat Pembayaran 64](#_Toc36611675)

[Tabel 3.22 Rincian Kasus Penggunaan Menampilkan Tagihan 65](#_Toc36611676)

[Tabel 3.23 Rincian Kasus Penggunaan Mengelola Tanggal FRS 67](#_Toc36611677)

[Tabel 3.30 Penjelasan Antarmuka Menampilkan kelas 73](#_Toc36611678)

[Tabel 3.31 Penjelasan Antarmuka *Form* Kelas 75](#_Toc36611679)

[Tabel 3.32 Penjelasan Antarmuka Mengelola Pendamping Akademik 77](#_Toc36611680)

[Tabel 3.33 Penjelasan Antarmuka Mengelola Kehadiran Dosen 79](#_Toc36611681)

[Tabel 3.34 Penjelasan Antarmuka *Form* Kehadiran Dosen 80](#_Toc36611682)

[Tabel 3.35 Penjelasan Antarmuka Mengelola Kehadiran Mahasiswa 82](#_Toc36611683)

[Tabel 3.36 Penjelasan Antarmuka Menyusun FRS 83](#_Toc36611684)

[Tabel 3.37 Penjelasan Antarmuka Menyutujui FRS 86](#_Toc36611685)

[Tabel 3.38 Penjelasan Antarmuka Menampilkan Anak Wali 87](#_Toc36611686)

[Tabel 3.39 Penjelasan Antarmuka Menampilkan Kelas Departemen 88](#_Toc36611687)

[Tabel 3.40 Penjelasan Antarmuka Menampilkan Kelas UPMB 89](#_Toc36611688)

[Tabel 3.41 Penjelasan Antarmuka Mengelola Tahun Ajaran 90](#_Toc36611689)

[Tabel 3.42 Penjelasan Antarmuka *Form* Tahun Ajaran 91](#_Toc36611690)

[Tabel 3.43 Penjelasan Antarmuka Mengelola Beban Studi 93](#_Toc36611691)

[Tabel 3.44 Penjelasan Antarmuka *Form* Beban Studi 94](#_Toc36611692)

[Tabel 3.45 Penjelasan Antarmuka Mengelola Tanggal Penting FRS 95](#_Toc36611693)

[Tabel 3.46 Penjelasan Antarmuka *Form* Tanggal Penting FRS 96](#_Toc36611694)

[Tabel 3.47 Penjelasan Antarmuka Mengelola Kuesioner 98](#_Toc36611695)

[Tabel 3.48 Penjelasan Antarmuka *Form* Kelola Kuesioner 99](#_Toc36611696)

[Tabel 3.49 Penjelasan Antarmuka Mengisi Kuesioner 101](#_Toc36611697)

[Tabel 3.50 Penjelasan Antarmuka Melihat Daftar Kuesioner 102](#_Toc36611698)

[Tabel 3.51 Penjelasan Antarmuka Melihat Hasil IPD Dosen 104](#_Toc36611699)

[Tabel 4.1 Lingkungan Implementasi Sistem 109](#_Toc36611700)

[Tabel 5.1 Lingkungan Pengujian 141](#_Toc36611701)

[Tabel 5.2 Rincian Pengujian Fungsi Mengelola Kelas 142](#_Toc36611702)

[Tabel 5.3 Rincian Pengujian Fungsi Mengelola Pendamping Akademik 144](#_Toc36611703)

[Tabel 5.4 Rincian Pengujian Fungsi Mengelola Kehadiran Dosen 146](#_Toc36611704)

[Tabel 5.5 Rincian Pengujian Fungsi Mengelola Kehadiran Mahasiswa 148](#_Toc36611705)

[Tabel 5.6 Rincian Pengujian Fungsi Menyusun FRS 150](#_Toc36611706)

[Tabel 5.7 Rincian Pengujian Fungsi Mencetak FRS 153](#_Toc36611707)

[Tabel 5.8 Rincian Pengujian Fungsi Menyetujui FRS 154](#_Toc36611708)

[Tabel 5.9 Rincian Pengujian Fungsi Menampilkan Anak Wali 155](#_Toc36611709)

[Tabel 5.10 Rincian Pengujian Fungsi Menampilkan Kelas Deparetmen 156](#_Toc36611710)

[Tabel 5.11 Rincian Pengujian Fungsi Menampilkan Fungsi Kelas UPMB 157](#_Toc36611711)

[Tabel 5.12 Rincian Pengujian Fungsi Mengelola Tahun Ajaran 157](#_Toc36611712)

[Tabel 5.13 Rincian Pengujian Fungsi Mengelola Beban Studi 159](#_Toc36611713)

[Tabel 5.14 Rincian Pengujian Fungsi Menampilkan Riwayat Pembayaran 161](#_Toc36611714)

[Tabel 5.15 Rincian Pengujian Fungsi Menampilkan Tagihan 162](#_Toc36611715)

[Tabel 5.16 Rincian Pengujian Fungsi Mengelola Tanggal FRS 163](#_Toc36611716)

[Tabel 5.17 Rincian Pengujian Fungsi Mengelola Kuesioner IPD 164](#_Toc36611717)

[Tabel 5.18 Rincian Pengujian Fungsi Mengisi Kuesioner IPD 166](#_Toc36611718)

[Tabel 5.19 Rincian Pengujian Fungsi Menampilkan Daftar Kuesioner 167](#_Toc36611719)

[Tabel 5.20 Rincian Pengujian Fungsi Menampilkan Hasil IPD Dosen 168](#_Toc36611720)

[Tabel 5.21 Rekapitulasi Hasil Pengujian Modul Pembelajaran 169](#_Toc36611721)

[Tabel 5.22 Rekapitulasi Hasil Pengujian Modul IPD 171](#_Toc36611722)

*[Halaman ini sengaja dikosongkan]*

# CHAPTER I INTRODUCTION

This chapter discusses the outline of the preparation of the final project which includes the background, the purpose of making it, the formulation and boundaries of the problem, the methodology for the preparation of the final project, and the systematic writing.

## Background

Nearly everyone goes on a vacation and a Travel information system would play a vital role in planning the perfect trip. The travel information system allows the user of the system access all the details such as hotels and flights. The main purpose is to help individual to manage customer, flights and hotels. The system can also be used for both professional and business trips. The proposed system maintains centralized repository to make necessary travel arrangements and to retrieve information easily.

With the rapid economic development in Fiji, the government has invested a lot of money into the tourism industry for the construction of many hotels to attract tourists to choose Fiji as their holiday destination. Therefore, Travel Information System in Fiji will be very useful for the tourists to use in booking flights and hotels.

This research will implement REST (Representational State Transfer) in developing API as the backend of the travel information system. API was developed using Javascript Object Notation (JSON) as the standard format for data communication and JSON Web Token (JWT) as user authentication code.

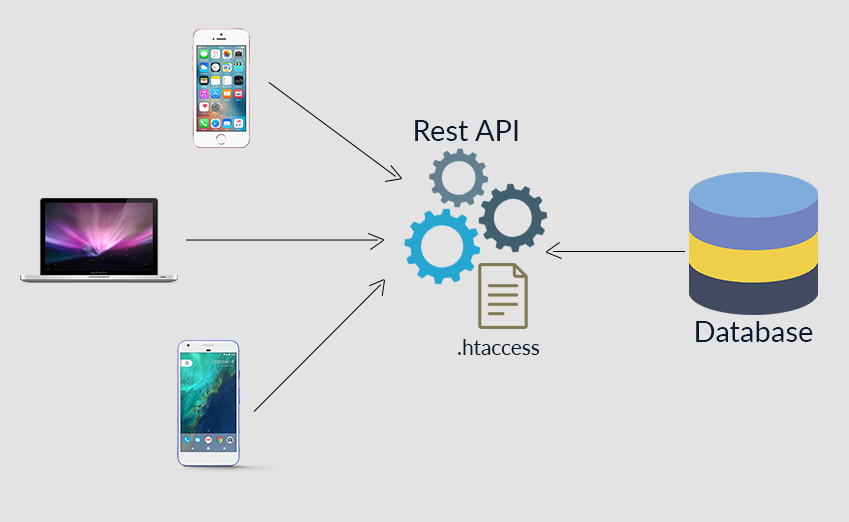


Figure 1.1 REST API Concept

## Problem Formulation

The formulation of the issues raised in this Final Project can be explained as follows.

1. How to design a traveling information system to be used in Fiji island for domestic flights booking.
2. How to implement this domestic traveling booking system using REST API

## Problem Limitation

The problems discussed in the final project have several limitations, which are as follows.

1. Fiji travel Information System use for domestic flights booking
2. Fiji travel Information System will’ be in web app.

## Goals

The goals of this final project are as follow.

1. To implement a travel information system in Fiji.
2. The travel information system to make it easy for booking of domestic flights around Fiji island

## Methodology

The steps taken to complete this Final Project are as follows:

* + - 1. **Completion of the final project**

This final project proposal contains a preliminary description of the final project to be made. This introduction consists of the background of the proposed final project, the formulation of the issues raised, the problem boundaries for the final project, the purpose of making the final project, and the benefits of the results of the final project. In addition, it also describes the literature review which is used as a reference to support the final project. The methodology sub-section contains an explanation of the stages of preparing the final project starting from the preparation of proposals to the preparation of the final project. There is also a sub-chapter of the activity schedule that explains the final work schedule.

* + - 1. **Literature Review**

At this stage the researcher collects information by conducting a study towards research that raises topics around development information systems with API concepts and REST architectural styles. Researchers to do direct communication and discuss with the employee section administration to one of the transport providers. This matter done to avoid possible mistakes in previous studies and strengthening the concepts used in research with similar cases.

* + - 1. **Software Analysis and Design**

At this stage an analysis and design of the academic information system design will be made. The analysis is carried out by determining the functional requirements of the system and designing the system design is done by making a database design, mock up the appearance of web pages and diagrams needed.

* + - 1. **Implementation**

The implementation phase in this research is carried out in several sub-processes, among others :

* + 1. Building a system development environment.
    2. Designing the structure of the table, functions and stored procedures on database.
    3. Develop a back-end (coding) system and adjust it to database.
    4. Design the rewrite structure on the web server.
       1. **Testing and Evaluation**

To ensure that the system runs according to the development plan facilitated business systems and processes, a testing scenario is needed system. In this study, testing will be divided into two test scenarios, including:

1. Web Apps Testing

Software testing is a test that focuses on the functional specifications of the web apps. This test is done to test whether the application is running well or not.

1. Mobile Apps Testing

Software testing is a test that focuses on the functional specifications of the web apps. This test is done to test whether the application is running well or not.

* + - 1. **Completion of Thesis**

At this stage, a book was prepared that made documentation regarding the design and implementation of travel information system in Fiji.

## Preparation of Thesis

This thesis book consists of several chapters that are arranged systematically, as follows.

1. Chapter 1. Introduction

The introductory chapter contains an explanation of the background of the problem, problem formulation, problem boundaries, objectives, benefits and systematic writing of the final project.

1. Chapter II. Literature review

The literature review chapter contains an explanation of the theoretical basis that supports the completion of the final project.

1. Chapter III. Analysis and Design

This chapter contains system design, database design, use case diagrams, activity diagrams and user interface designs.

1. Chapter IV. Implementation

This chapter discusses the implementation of the designs made in the previous chapter. Explanation in the form of code used for the implementation process.

1. Chapter V. Trials and Evaluations

This chapter explains the ability of the software by testing the truth and testing the performance of the system that has been made.

1. Chapter VI. Conclusions and recommendations

This chapter is the last chapter that provides conclusions from the results of trials conducted and suggestions for future software development.

# CHAPTER II LITERATURE REVIEW

The literature review chapter contains explanations of theories related to software implementation. The explanation aims to provide an overview of the system to be built and is useful as a support in design and implementation of travel information system.



## Pembelajaran

Menurut Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 Tahun 2014 [1] tentang Standar Nasional Perguruan Tinggi, pembelajaran adalah proses interaksi mahasiswa dengan dosen dan sumber belajar pada suatu lingkungan belajar. Sementara Satuan Kredit Semester (SKS) adalah suatu sistem yang umumnya digunakan di perguruan tinggi. Sistem SKS ini biasanya digunakan untuk mengukur besarnya beban studi mahasiswa, usaha belajar yang diperlukan mahasiswa untuk menyelesaikan suatu program, dan usaha penyelenggaraan pendidikan bagi tenaga pengajar [2].

Dari definisi tersebut, lingkup pembelajaran meliputi berbagai proses di dalam aktivitas akademik yaitu pengelolaan kelas pembelajaran, perencanaan kontrak belajar, dan proses pembelajaran itu sendiri.

## IPD

Indeks Pengajaran Dosen (IPD) adalah layanan yang dapat membantu dosen untuk melihat sejauh apa efektivitas dari pengajaran dosen tersebut selama satu semester [3]. IPD biasanya berbentuk kuesioner yang berisi 2 macam penilaian, yaitu penilaian kuesioner dosen dan penilaian kuesioner mata kuliah yang bersangkutan. Penilaian kuesioner dosen adalah kuesioner yang berisi tingakt kepuasan mahasiswa terhadap seorang dosen yang mengajarkan suatu kelas pembelajaran. Sementara penilaian kuesioner mata kuliah adalah kuesioner yang berisi tingkat kepuasan mahasiswa tentang mata kuliah yang diambilnya.

## Sistem Informasi Akademik Generik

Perangkat lunak generik adalah jenis perangkat lunak yang siap mendukung proses bisnis yang umum dan berlawanan dengan pengertian perangkat lunak yang dikembangkan untuk kebutuhan tertentu [4]. Sinonim dari perangkat lunak generik adalah Commercial Off-The Shelf (COTS). COTS merupakan produk perangkat lunak yang dirancang sedemikian rupa agar mudah dipasang dan beroperasi dengan komponen sistem yang sudah ada.

Penerapan perangkat lunak generik telah dilakukan pada aplikasi, contohnya yang aplikasi yang memiliki fokus pada sistem informasi akademik (SIA).

SIA yang bersifat generik merupakan SIA yang dirancang sehingga memiliki proses bisnis yang bersifat umum. Pada modul pembelajaran atau FRS, terdapat beberapa fungsi yang bersifat umum yaitu fungsi menyusun FRS, fungsi pembuatan kelas. Sementara pada modul IPD, terdapat juga beberapa fungsi yang bersifat umum yaitu fungsi mengelola data kuesioner IPD dan fungsi mengisi kuesioner IPD.

## Phalcon PHP

Phalcon merupakan framework PHP open source. Fakta bahwa Phalcon ditulis sebagai ekstensi PHP menggunakan bahasa pemrograman C yang membedakan dari framework yang lain. Phalcon menggunakan prinsip-prinsip MVC dan dikembangkan oleh Phalcon Team [5].

Framework Phalcon memiliki karakteristik sebagai berikut:

• Semua komponen ditulis dalam bahasa pemrograman C.

• Ada berbagai versi untuk sistem operasi popular: Linux, Windows, dan Mac.

• Kinerja tinggi dan biaya sumber daya server rendah.

• Phalcon merupakan salah satu framework PHP tercepat.

Ketika melihat kemampuan routing bahwa Phalcon memiliki dua mode routing yaitu, mode MVC dan match only mode secara otomatis mencoba menemukan controller dan metodenya berdasarkan input URL. Mode MVC memungkinkan programmer untuk secara manual mengkonfigurasi route dan mengarahkan permintaan ke controller dan metode yang sesuai. Router dapat dikonfigurasi dengan kode PHP atau menggunakan anotasi yang ditulis langsung pada controller.

Kerangka Phalcon memiliki template sendiri yang disebut “Volt” [6]. Volt ditulis menggunakan Bahasa C dan dikompilasi bersama dengan Phalcon sebagai ekstensi PHP. Volt juga mengubah semua template ke kode PHP sehingga memungkinkan untuk menggunakan kode PHP (raw PHP) mentah di dalam template. Untuk penggunaan kembali kode yang sudah ditulis, Volt menawarkan berbagai mekanisme. Kode dapat dibagi menggunakan operator {% blok %}. Fungsionalitas ini dapat digunakan, misalnya untuk membuat file master tunggal dan kemudian cukup memasukkan template yang lain. Saat ingin melakukan extend pada template, perlu untuk menentukan blok yang akan diganti.

## Microsoft SQL Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft [7]. Bahasa query utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Microsoft SQL Server banyak digunakan pada dunia bisnis, pendidikan, dan juga pemerintahan sebagai solusi penyimpanan data. Kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan menggunakan protokol TDS (Tabular Data Stream). Selain itu, Microsoft SQL Server juga mendukung ODBC (Open Database Connectivity), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data mirroring dan clustering.

## Repository-Service Pattern

A screenshot of a cell phone

Description generated with high confidence

Gambar 2.1 Diagram Pola Perancangan *Repository-Service*

*Pattern* atau pola ini digunakan untuk menerapkan konsep pemisahan kode program berdasarkan fungsinya seperti yang digambarkan pada Gambar 2.1. Semua kode untuk akses database harus dipisahkan dengan kode untuk pengaturan user interface. Hal ini memungkinkan kode akses database yang dibuat untuk aplikasi desktop, dengan mudah digunakan untuk aplikasi web [8]. Selain itu, penerapan konsep ini secara disiplin, dapat menghasilkan kode program yang dapat dites secara otomatis menggunakan *Unit Testing*.

Pada pola perancangan *Repository Pattern,* ada 4 lapisan di antara pengguna dengan basis data, yaitu *view, controller, service,* dan *repository*. Lapisan *View* merupakan lapisan yang berisi tampilan yang dapat dilihat oleh pengguna. Lapisan *Controller* adalah penyambung antara lapisan *View* dengan lapisan *Service*. Lapisan *Service* berisi seluruh logika algoritma dari suatu modul. Lapisan *Repository* berisi *query* yang digunakan untuk memanggil basis data [9].

## MVC (Model-View-Controller)

MVC (Model-View-Controller) adalah pola desain perangkat lunak untuk mengembangkan aplikasi web. Pola Model-View-Controller terdiri dari tiga komponen, yaitu model, view, dan controller [10]. Setiap komponen tersebut dibuat untuk menangani pengembangan aspek spesifik dari sebuah aplikasi. Model adalah tingkat terendah dari pola yang bertanggung jawab untuk memelihara data. View bertanggung jawab untuk menampilkan semua atau sebagian data kepada pengguna. Controller merupakan kode perangkat lunak yang mengontrol interaksi antara Model dan View [11].

*[Halaman ini sengaja dikosongkan]*

# CHAPTER III ANALYSIS AND DESIGN

In this chapter, We will discused about the design of the system that will be developed. The system design will include the analysis of the requirements that needed in the system after getting a generic business process. The design of this system will be represented by various Unified Modeling Languages ​​(called UML).



## Analysis

The analysis phase is divided into several parts, including analysis of business process references, use cases of the system, and software requirements. In general, the steps above can be explained with Figure 3.1

Collecting the business process requirement from several references

Creating process diagrams

Performing a detailed analysis

Figure 3.1 Software Requirements Analysis Phase

### Software Requirements Specifications

Based on the description of scope of the software that will be developed, it is necessary to have a software specification in order to provide solutions to the problems given and be able to accommodate the needs. It is expected that this specification can adjust to user needs. Software requirements specification in this thesis consists of functional requirements which can be seen in Table 3.1 and Table 3.2

### Functional Requirements

Table 3.1 Functional Table of the Software

| No | Functional Requirements | Description |
| --- | --- | --- |
|
| 1 | Create travel destination | Create new travel destination |
| 2 | Update travel destination | Update existing travel destination |
| 3 | View travel destination | View the list and detail of travel destination |
| 4 | Manage Users | Manage all of registered users |
| 5 | View dashboard | View the informations on dashboard |
| 6 | Register | Register for a new user |
| 7 | Login | Login for registered user based on roles |
| 8 | Search travel destination | Search the travel destinations |
| 9 | Manage booking cart | Manage booking cart |
| 10 | Checkout | Checkout items on cart |
| 11 | View tickets | View detail of tickets that have been ordered |
| 12 | Manage user profile | Manage detail of user profile |

### Actors

Actors are people who interact with system. They have different roles and privileges. In this information system, there are four actors that explained on Table 3.3

Table 3.3 Table Actors of System

| No | Actor | Description |
| --- | --- | --- |
| 1 | Admin | * Create travel destination * Update travel destination * View travel destination * Manage Users * View dashboard |
| 2 | Customer | * Register * Login * Search travel destination * Manage booking cart * Payment * View tickets * Manage user profile |

### Use Case Diagram

This section explains the detail of the use cases contained in the software as shown in Figure 3.2. There are also table description and activity diagrams for each use case.

A picture containing text

Description automatically generated

Gambar 3.2 Fiji Travel Information System Use Case Diagram

#### Create Travel Destinations Use Case

In this use case, Admin can create new travel destinations that shown on the apps.Then, this travel destinations can be ordered by registered customer. Detail of the usecase is shown on Table 3.5 and the activity diagram is shown on Figure 3.3

Table 3.5 Detail of Create Travel Destination’s Use Case

| Components | Description |
| --- | --- |
| Name | Create travel destination |
| Code | UC-001 |
| Description | This usecase is used to create new travel destination. |
| Type | Functional |
| Actor | Admin |
| Initial Conditition | Travel destination has not been added in the system. |
| End Condition | Travel destination has successfully added in the system. |
| Normal Flow | 1. Actor choose destinations menu 2. System show the list of destinations 3. Actor click Create Destination Button 4. System show the form to create new destination. 5. Actor fill the form to create new travel destination 6. System save the new data |
| Alternate Flow | - |

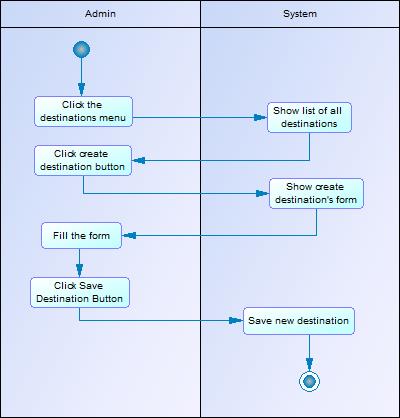


Figure 3.3 Create Travel Destination Activity Diagram

#### Update Travel Destination Use Case

In this use case, Admin can update the existing travel destinations that shown on the apps.Then, this travel destinations can be ordered by registered customer. Detail of the usecase is shown on Table 3.6 and the activity diagram is shown on Figure 3.4.

Table 3.6 Detail of Update Travel Destination’s Use Case

| Components | Description |
| --- | --- |
| Name | Update travel destination |
| Code | UC-002 |
| Description | This usecase is used to update travel destination |
| Type | Functional |
| Actor | Admin |
| Initial Conditition | Travel destination has not been changed. |
| End Condition | Travel destination has successfully updated |
| Normal Flow | 1. Actor choose destinations menu 2. System show the list of destinations 3. Actor choose the travel destination that want to be changed 4. Actor click update button 5. System show the form to update the destination. 6. Actor fill the form to update travel destination 7. System save the updated data |
| Alternate Flow | - |

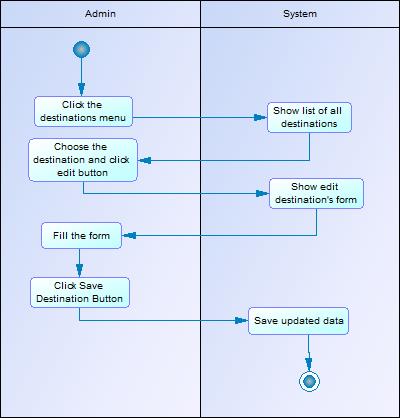


Figure 3.4 Update Travel Destination Activity Diagram

#### View Travel Destination Use Case

In this use case, Admin can view the detail of existing travel destinations that shown on the apps. Detail of the usecase is shown on Table 3.7 and the activity diagram is shown on Figure 3.5.

Table 3.7 Detail of View Travel Destination’s Use Case

| Components | Description |
| --- | --- |
| Name | View travel destination |
| Code | UC-003 |
| Description | This usecase is used to view the detail of existing travel destination |
| Type | Functional |
| Actor | Admin |
| Initial Conditition | User choose the destination menu |
| End Condition | System successfully show the detail of destination |
| Normal Flow | 1. Actor choose destinations menu 2. System show the list of destinations 3. Actor choose which travel destination that want to be viewed 4. Actor click show button 5. System show the detail of destination. |
| Alternate Flow | - |

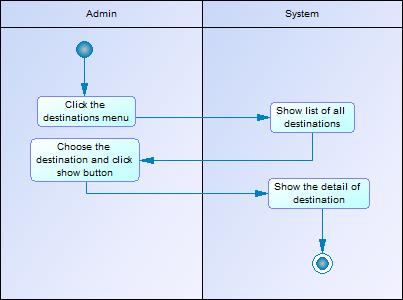


Figure 3.5 View Travel Destination Activity Diagram

#### Manage Users Use Case

In this use case, Admin can manage the users data. Admin can update the data of all existing user. Detail of the usecase is shown on Table 3.8 and the activity diagram is shown on Figure 3.6.

Table 3.8 Detail of Manage Users’s Use Case

| Components | Description |
| --- | --- |
| Name | Manage Users |
| Code | UC-004 |
| Description | This use case is used to manage the users on system |
| Type | Functional |
| Actor | Admin |
| Initial Conditition | User has registered on system |
| End Condition | Admin has successfully update the user’s data |
| Normal Flow | 1. Actor choose manage users menu 2. System show the list of users 3. Actor choose the user that want to be changed 4. System show the form to update the user. 5. Actor fill the form to update user’s data 6. System save the updated data |
| Alternate Flow | - |

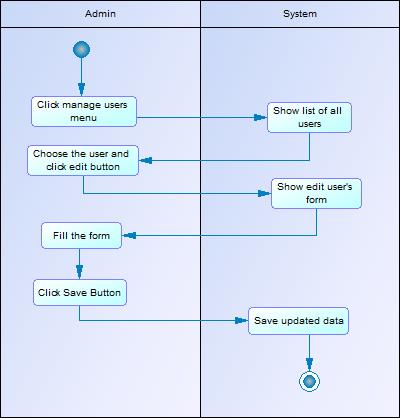


Figure 3.6 Manage Users Activity Diagram

#### View Dashboard Use Case

In this use case, Admin can view the dashboar after login. Detail of the usecase is shown on Table 3.9 and the activity diagram is shown on Figure 3.7.

Table 3.9 Detail of View Dashboard use Case

| Components | Description |
| --- | --- |
| Name | View Dashboard |
| Code | UC-005 |
| Description | This use case is used to view the dashboard of the application |
| Type | Functional |
| Actor | Admin |
| Initial Conditition | User is exist on the database and has role as admin |
| End Condition | System redirect the user to dashboard |
| Normal Flow | 1. System shows the login page 2. User enter his credential as an admin 3. System redirect the user to the dashboard |
| Alternate Flow | 1. System redirect user to login page |

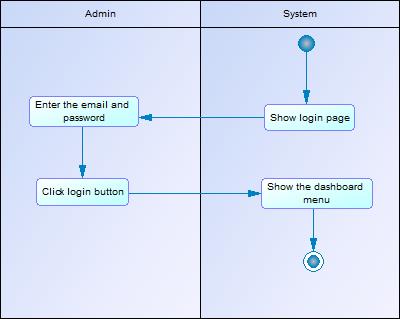


Figure 3.7 View Dashboard Activity Diagram

#### Register Use Case

In this use case, guest can register to system. User must registered to system to book the travel destination. Detail of the usecase is shown on Table 3.10 and the activity diagram is shown on Figure 3.8.

Table 3.10 Detail of Register Use Case

| Components | Description |
| --- | --- |
| Name | Register |
| Code | UC-006 |
| Description | This usecase is used to register for a new user |
| Type | Functional |
| Actor | Guest |
| Initial Conditition | User has not been registered on system |
| End Condition | User successful registerd on system |
| Normal Flow | 1. Actor click register button 2. System show registration form 3. Actor fill the registration form 4. System save the registration data |
| Alternate Flow | * 1. Actor failed to fill the form  1. System redirect to registration form and give the warning message |

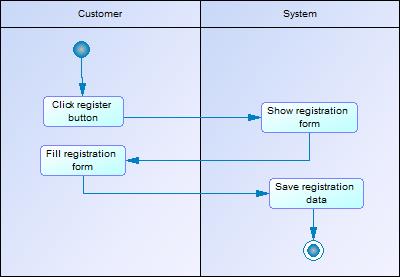


Figure 3.8 Register Activity Diagram

#### 3.1.4.7 Login Use Case

In this use case, registered user can login to the system. There are two roles of users : admin and customer.Each of role has different capabilities and permissions. The detail of use cas is shown on Table 3.11 and the activity diagram is shown on Figure Gambar 3.9.

Table 3.11 Detail of Login Use Case

| Components | Description |
| --- | --- |
| Name | Login |
| Code | UC-007 |
| Description | This usecase is used to login into system for registered users |
| Type | Functional |
| Actor | Registered User |
| Initial Conditition | User has not logged in into system |
| End Condition | User has successfully login into system |
| Normal Flow | 1. Actor go to login page 2. Actor enter his credential 3. System redirect the user into system based on role |
| Alternate Flow | * 1. Actor login as admin   1. System redirect into admin page  3.3 Actor login as customer  1. System redirect into customer page |

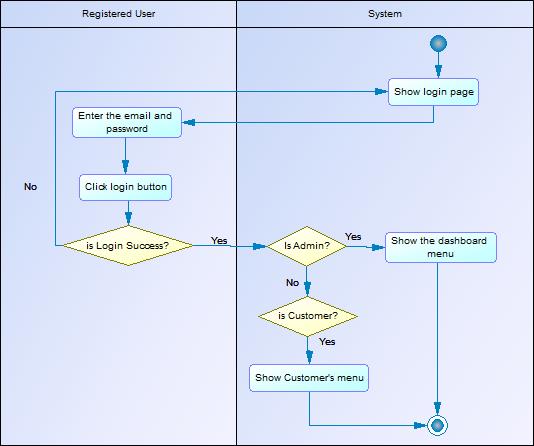


Figure 3.9 Login Activity Diagram

#### Search Travel Destinations Use Case

In this use case, Customer can search the travel destinations that they want.Then, this travel destinations can be ordered by registered customer. Detail of the usecase is shown on Table 3.12 and the activity diagram is shown on Figure 3.10.

Table 3.12 Detail of Search Travel Destinations Use Case

| Components | Description |
| --- | --- |
| Name | Search travel destination |
| Code | UC-008 |
| Description | This use case is used to sear travel destinations on system |
| Type | Functional |
| Actor | Cutomer |
| Initial Conditition | System show the homepage |
| End Condition | System show the destinations based on keyword entered by actor |
| Normal Flow | 1. Actor go to search destination tab content 2. Actor enter the keyword of the destination 3. Actor click search button 4. System show the result |
| Alternate Flow | * 1. The entered keyword is match with the existing destination on system  1. System show the destinations    1. The entered keyword is not match with any existing destination on system 2. System show the warning text |

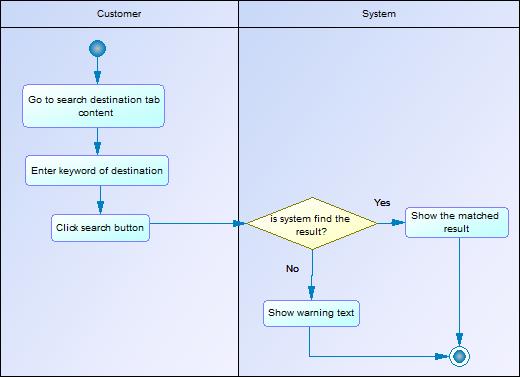


Figure 3.10 Search Destination Activity Diagram

#### Manage Booking Cart Use Case

In this use case, Customer can manage their cart.Customer can add destination on the cart and can clear the cart before they can purchase it. Detail of the usecase is shown on Table 3.13 and the activity diagram is shown on Figure 3.11.

Table 3.13 Detail of Manage Booking Cart Use Case

| Components | Description |
| --- | --- |
| Name | Manage booking cart |
| Code | UC-009 |
| Description | This use case is used to manage the customer’s booking cart |
| Type | Functional |
| Actor | Customer |
| Initial Conditition | The cart is empty |
| End Condition | Actor successfully add destination on cart |
| Normal Flow | 1. Actor choose the destination 2. Actor click Add to Cart Button 3. The destination has added into cart 4. Actor click the cart icon 5. System show the actor’s cart |
| Alternate Flow | 4.1 User want to clear the cart  1. User click clear cart button  4.2 User want to purchase the cart  1. User click continue to payment button |

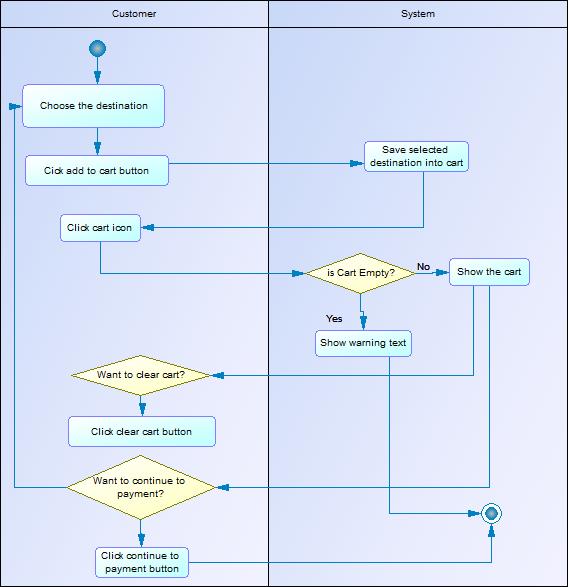


Figure 3.11 Manage Booking Cart Activity Diagram

#### Payment Use Case

In this use case, Customer can pay the existing destinations on his cart. Detail of the usecase is shown on Table 3.14 and the activity diagram is shown on Figure 3.12.

Table 3.14 Detail of Payment Use Case

| Components | Description |
| --- | --- |
| Name | Payment |
| Code | UC-010 |
| Description | This use case is used to make customer’s payment |
| Type | Functional |
| Actor | Customer |
| Initial Conditition | The cart is not empty |
| End Condition | System show the t\detail of payment |
| Normal Flow | 1. Actor click the cart icon 2. Actor click continue to pay 3. System show the detail of payment 4. Actor gets the ticket |
| Alternate Flow | - |

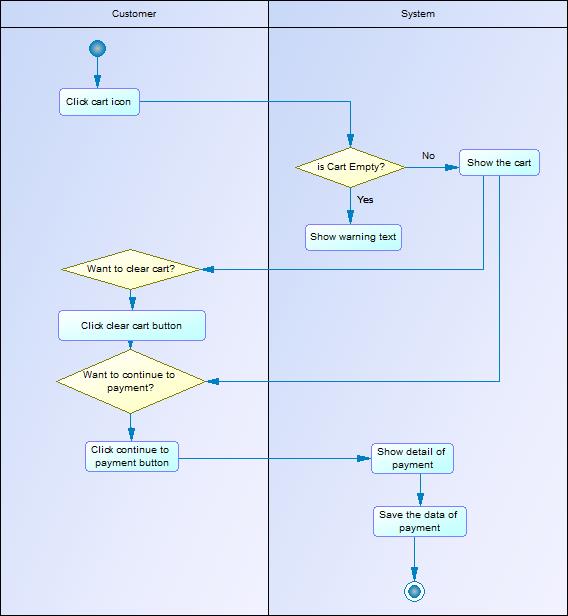


Figure 3.12 Payment Activity Diagram

#### View Tickets Use Case

In this use case, Customer can view their booking ticket .Detail of the usecase is shown on Table 3.15 and the activity diagram is shown on Figure 3.13.

Table 3.15 Detail of View Tickets Use Case

| Components | Description |
| --- | --- |
| Name | View tickets |
| Code | UC-011 |
| Description | This usecase is used by customers to view their tickets |
| Type | Functional |
| Actor | Customer |
| Initial Conditition | Payment is complete |
| End Condition | Actor can view the ticket |
| Normal Flow | 1. Actor click My Tickets menu 2. System show the list of actor’s ticket 3. Actor click show ticket button 4. System show the detail of ticket |
| Alternate Flow | - |

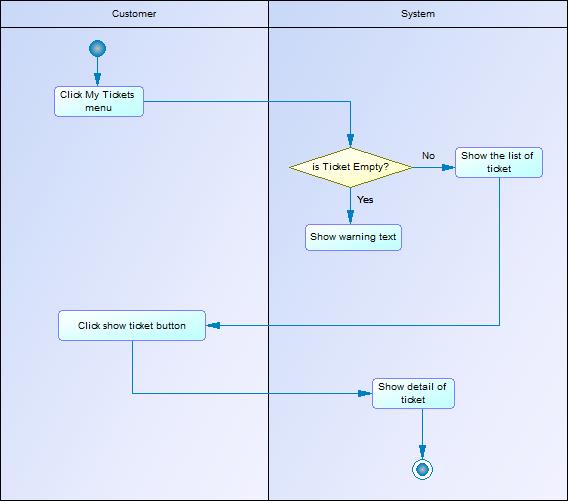


Figure 3.13 View Tickets Activity Diagram

#### Manage User Profile Use Case

In this use case, User can update their user profile. Detail of the usecase is shown on Table 3.16 and the activity diagram is shown on Figure 3.14.

Table 3.16 Detail of Manage User Profile Use Case

| Components | Description |
| --- | --- |
| Name | Manage user profile |
| Code | UC-012 |
| Description | This use case is used by user to manage their profile |
| Type | Functional |
| Actor | Customer |
| Initial Conditition | User’s Profile has not been changed |
| End Condition | User’s Profile has successfully updated |
| Normal Flow | 1. Actor click My Profile menu 2. System show existing the actor’s profile 3. Actor click update button 4. System show the update profile form 5. Actor click save button 6. System save the updated data |
| Alternate Flow | - |

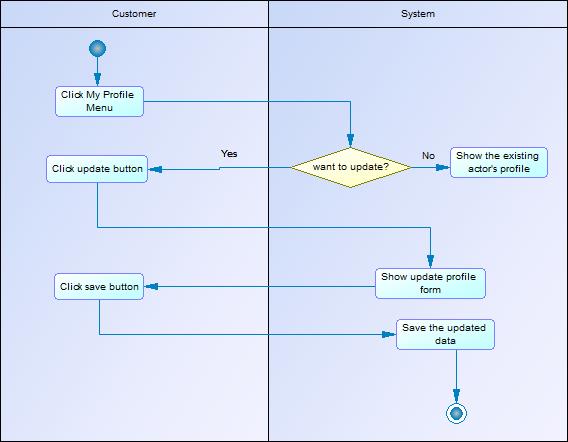


Figure 3.14 Manage User Profile Activity Diagram

## Design

In the design section will be explain about the system architecture used, class diagram design, database design, and interface design.

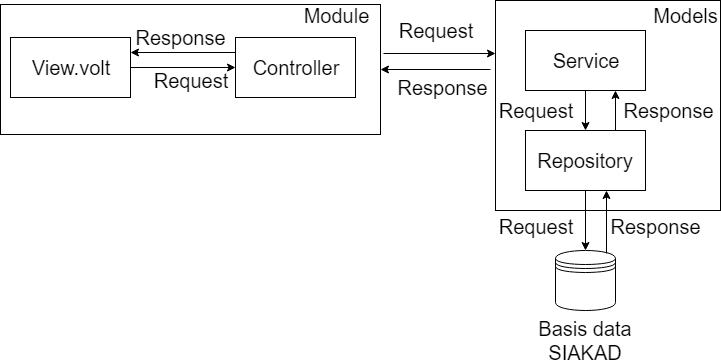
### Architectural Design and Design Pattern

The system architecture used in this Final Project is REST API. Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations. An API is an application programming interface. It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

REST determines how the API looks like.It is a set of rules that developers follow when they create their API. In this project , Laravel framework is used to manage the REST API (Backend) and Vue JS is used for User Interface (Frontend). The user interface is a layer that is directly related to the user.

The controller is the link between the interface and the service layer of the application. The service layer provides data processing from the repository layer. Then, the controller gets the data returned. After that, the controller will display in the user interface.

Repository classes and entities in the database are representations of the MVC architecture model. The service and controller classes, while the user interface is a representation of the view. The illustration in Figure 3.29 shows an MVC architecture diagram with a Repository-Service design pattern.



Gambar 3.29 Diagram Arsitektur MVC dengan pola perancangan *Repository-Service*

### Class Diagram Design

In the Appendix chapter, the architecture model has classes in the form of controller, service, and repository. The use of system architecture like the picture above is used to make maintenance easier and easier to implement.

The controller class depends on the service class that is the place of data processing and the service class sends data requests to the database through the repository class. The repository class will send requests to the database and send data back from the database to the service class.

*.*

### Database Design

Analysis of database design is needed to make an information system. MySQL was chosen as a database application because it can hold data on a large scale and free.

The database design is displayed in the form of Conceptual Data Model (CDM) and Physical Data Model (PDM). For a more detailed explanation, PDM will be explained in the Appendix chapter.

### User Interface Design

This section will discuss about the design of interface for users based on use cases that have been designed.

#### Landing Page

Landing page is page where website visitors land when they first reach the website. A landing page is also an important term and concept for inbound marketing and lead generation. This page contain some important information of the information system. The design of user interface is shown on Figure 3.30. and Figure 3.31.

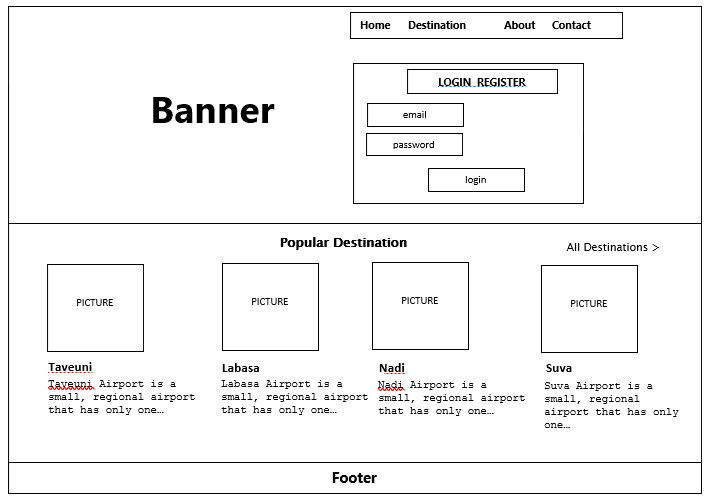
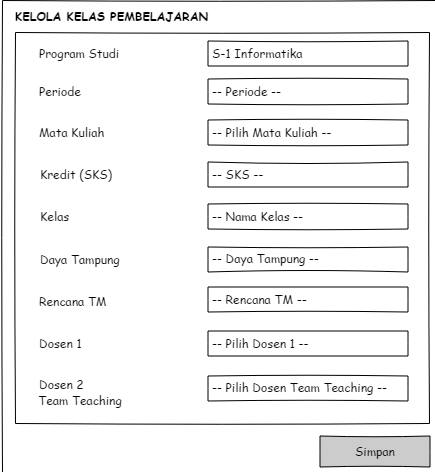


Figure 3.30 Desktop Landing Page User Interface Design

Tabel 3.30 Penjelasan Antarmuka Menampilkan kelas

| No | Attribute Name | Type | Function | Input/ Output |
| --- | --- | --- | --- | --- |
| 1 | *datatableKelas* | *Table* | Menampilkan data kelas. | *Table* |
| 2 | *editButton* | *Button* | Tombol aksi untuk memperbarui data kelas. | *ButtonClick* |
| 3 | *deleteButton* | *Button* | Tombol aksi untuk menghapus data kelas. | *ButtonClick* |
| 4 | *addButton* | *Button* | Tombol untuk menambahkan data kelas. | *ButtonClick* |
| 5 | *pilihPeriode* | *Dropdown* | Input untuk mengisi data periode | *String* |
| 6 | *hadirKelas* | *Button* | Tombol untuk pindah ke halaman kehadiran dosen | *ButtonClick* |



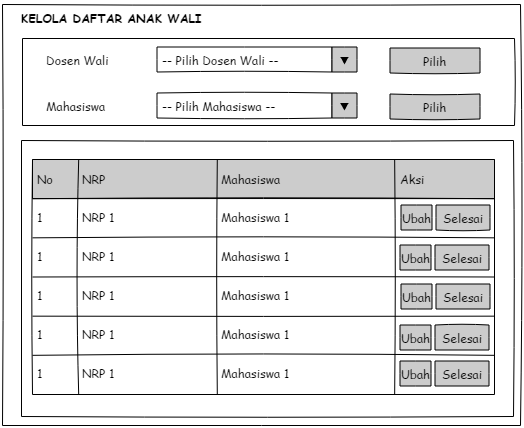
Gambar 3.31 Rancangan Antarmuka *Form* Kelas

Tabel 3.31 Penjelasan Antarmuka *Form* Kelas

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *namaProdi* | *Text* | Nama program studi. | *String* |
| 2 | *periode* | *Dropdown* | Input nama periode. | *String* |
| 3 | *namaMataKuliah* | *Dropdown* | Input nama mata kuliah. | *String* |
| 4 | *SKS* | *Text* | Input SKS. | *Integer* |
| 5 | *namaKelas* | *Text* | Input nama kelas. | *String* |
| 6 | *dayaTampungKelas* | *Text* | Input daya tampung kelas. | *Integer* |
| 7 | *rencanaTatapMukaKelas* | *Text* | Input rencana tatap muka. | *Integer* |
| 8 | *namaDosenUtama* | *Dropdown* | Input nama dosen utama. | *String* |
| 9 | *namaDosenTeam* | *Dropdown* | Input nama dosen untuk kelas *team teaching*. | *String* |
| 10 | *simpanButton* | *Button* | Tombol simpan data kelas. | *ButtonClick* |

#### Dashboard Page

Halaman ini digunakan untuk mengelola penempatan dosen wali dengan anak wali nya. Pada halaman ini terdapat daftar anak wali dari seorang dosen yang dipilih. Pengguna juga dapat memilih anak wali untuk ditempatkan pada perwalian dengan dosen tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.32.



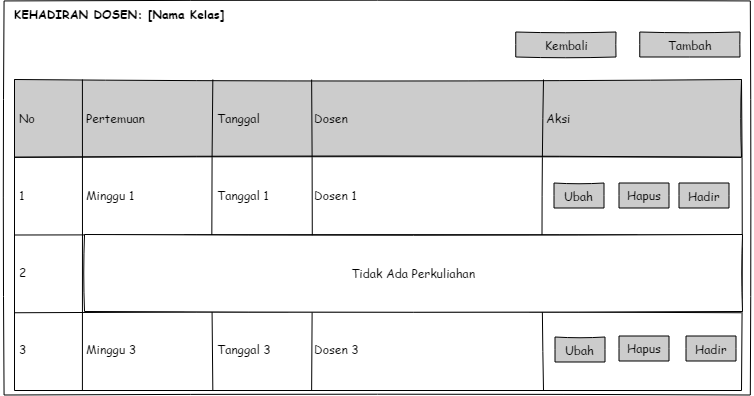
Gambar 3.32 Rancangan Antarmuka Mengelola Pendamping Akademik

Tabel 3.32 Penjelasan Antarmuka Mengelola Pendamping Akademik

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *NamaDosenWali* | *Dropdown* | Input nama dosen wali. | *String* |
| 2 | *NamaAnakWali* | *Dropdown* | Input nama anak wali. | *String* |
| 3 | *datatableAnakWali* | *Table* | Menampilkan data anak wali | *Table* |
| 4 | *editButton* | *Button* | Tombol aksi untuk mengubah data anak wali | *ButtonClick* |
| 5 | *deleteButton* | *Button* | Tombol aksi untuk menghapus data anak wali. | *ButtonClick* |
| 6 | *pilihDosenWali* | *Button* | Tombol aksi untuk memilih dosen wali. | *ButtonClick* |
| 7 | *PilihAnakWali* | *Button* | Tombol aksi untuk memilih anak wali. | *ButtonClick* |

#### Destinations Management Page

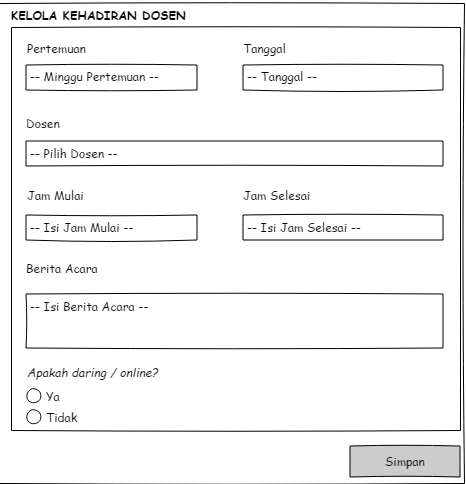
Halaman ini digunakan untuk mengelola data kehadiran dosen terhadap suatu kelas pembelajaran. Pada halaman ini terdapat daftar kehadiran dosen yang sudah dibuat. Pengguna juga dapat membuat, mengubah, atau menghapus data kehadiran dosen. Halaman antarmuka dapat dilihat pada Gambar 3.33 dan form untuk halaman ini dapat dilihat pada Gambar 3.34.



Gambar 3.33 Rancangan Antarmuka Mengelola Kehadiran Dosen

Tabel 3.33 Penjelasan Antarmuka Mengelola Kehadiran Dosen

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tambahButton* | *Button* | Tombol aksi untuk membuat data kehadiran dosen baru | *ButtonClick* |
| 2 | *ubahButton* | *Button* | Tombol aksi untuk mengubah data kehadiran dosen. | *ButtonClick* |
| 3 | *hapusButton* | *Button* | Tombol aksi untuk menghapus data kehadiran dosen. | *ButtonClick* |
| 4 | *hadirButton* | *Button* | Tombol aksi pindah ke halaman kehadiran mahasiswa | *ButtonClick* |
| 5 | *datatableHadirDosen* | *Table* | Menampilkan data kehadiran dosen. | *Table* |



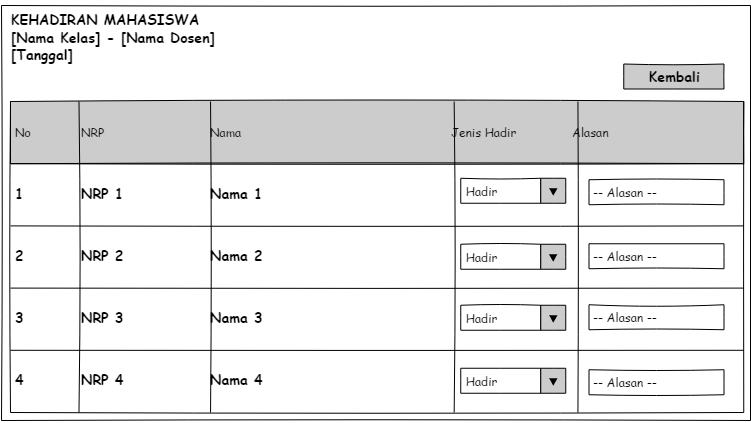
Gambar 3.34 Rancangan Antarmuka *Form* Kehadiran Dosen

Tabel 3.34 Penjelasan Antarmuka *Form* Kehadiran Dosen

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *mingguPertemuan* | *Text* | Input minggu pertemuan tatap muka. | *Integer* |
| 2 | *tanggalPertemuan* | *Date* | Input tanggal pertemuan. | *Date* |
| 3 | *dosen* | *Text* | Input nama dosen | *String* |
| 4 | *jamMulai* | *Number* | Input jam mulai pertemuan | *Integer* |
| 5 | *jamSelesai* | *Number* | Input jam selesai pertemuan. | *Integer* |
| 6 | *beritaAcara* | *Text* | Input berita acara. | *String* |
| 7 | *statusDaring* | *RadioButton* | Input jenis status kelas. | *Integer* |
| 8 | *simpanButton* | *Button* | Tombol untuk menyimpan data kelola kehadiran | *Button* |

#### Users Management Page

Halaman ini digunakan untuk mengelola data kehadiran mahasiswa terhadap suatu kelas pembelajaran. Pada halaman ini terdapat daftar mahasiswa dan jenis kehadiran nya. Pengguna dapat mengubah data kehadiran tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.35.



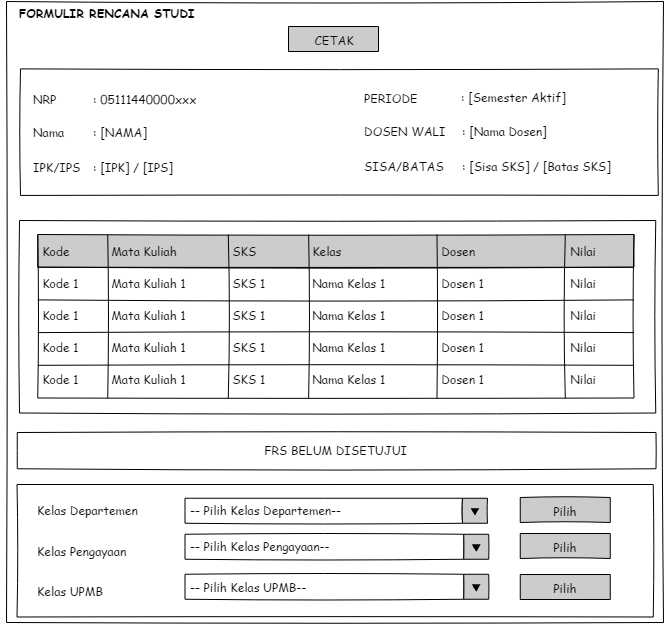
Gambar 3.35 Rancangan Antarmuka Mengelola Kehadiran Mahasiswa

Tabel 3.35 Penjelasan Antarmuka Mengelola Kehadiran Mahasiswa

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *kembaliButton* | *Button* | Tombol untuk kembali ke halaman kehadiran dosen. | *Button* |
| 2 | *datatableHadirMahasiswa* | *Table* | Menampilkan daftar nama dan kehadiran mahasiswa. | *Table* |
| 3 | *jenisHadir* | *Dropdown* | Input untuk memilih jenis kehadiran mahasiswa | *String* |
| 4 | *alasanHadir* | *Text* | Input untuk mengisi alasan kehadiran mahasiswa | *String* |

#### Payment History Page

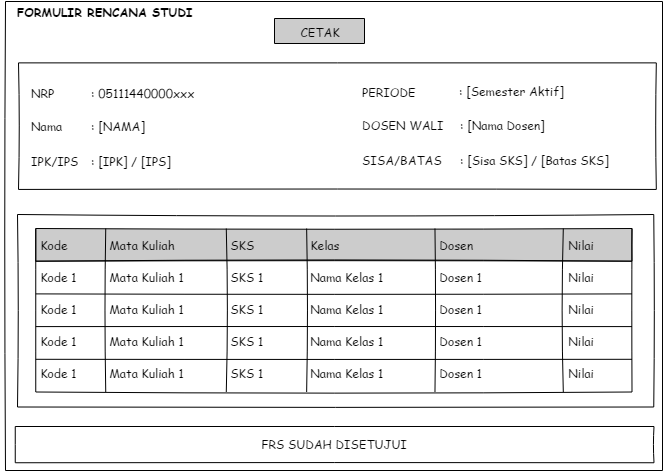
Halaman ini digunakan untuk menyusun FRS. Pada halaman ini terdapat daftar kelas yang dipilih ketika melakukan FRS. Pengguna juga dapat memilih kelas yang diinginkan apabila FRS tersebut belum disetujui oleh dosen wali yang bersangkutan. Halaman antarmuka dapat dilihat pada Gambar 3.36 dan Gambar 3.37.



Gambar 3.36 Rancangan Antarmuka FRS Belum Setuju

Tabel 3.36 Penjelasan Antarmuka Menyusun FRS

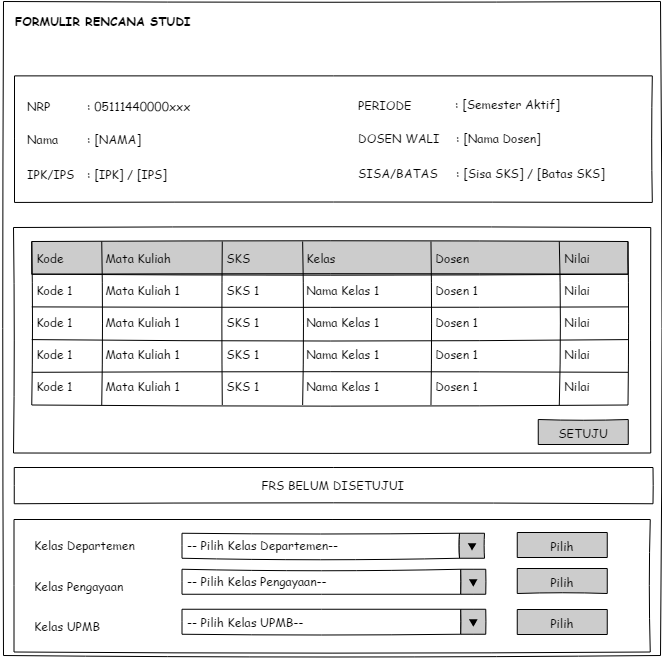
| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *cetakButton* | *Button* | Tombol untuk mengubah halaman menjadi versi cetak. | *Button* |
| 2 | *datatableMahasiswa* | *Table* | Menampilkan data mahasiswa pada halaman FRS | *Table* |
| 3 | *datatableFRS* | *Table* | Menampilkan data kelas yang dipilih mahasiswa ketika FRS | *Table* |
| 4 | *statusLabel* | *Text* | Label yang menampilkan status persetujuan FRS | *String* |
| 5 | *namaKelas* | *Dropdown* | Input untuk memilih nama kelas | *String* |
| 6 | *pilihKelasButton* | *Button* | Tombol aksi untuk memilih kelas ketika FRS. | *ButtonClick* |



Gambar 3.37 Rancangan Antarmuka FRS Sudah Setuju

#### Detail of Destination Page

Halaman ini digunakan untuk menyetujui atau membatalkan FRS seorang anak wali. Pada halaman ini pengguna dapat menyetujui atau membatalkan status FRS seorang anak wali. Halaman antarmuka dapat dilihat pada Gambar 3.38.



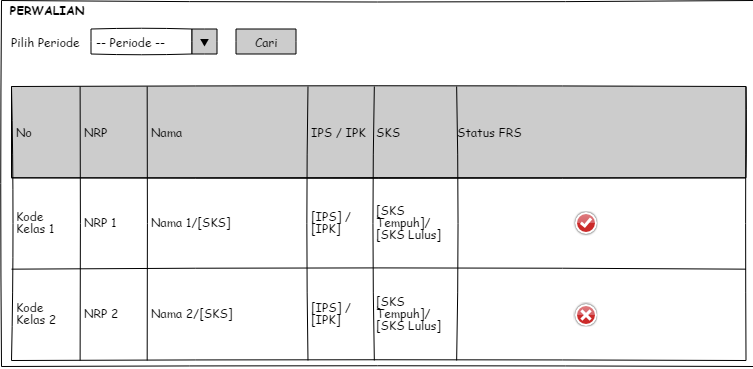
Gambar 3.38 Rancangan Antarmuka Menyetujui FRS

Tabel 3.37 Penjelasan Antarmuka Menyutujui FRS

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *datatableMahasiswa* | *Table* | Menampilkan data mahasiswa pada halaman FRS | *Table* |
| 2 | *datatableFRS* | *Table* | Menampilkan data kelas yang dipilih mahasiswa ketika FRS | *Table* |
| 3 | *setujuButton* | *Button* | Tombol untuk menyetujui atau membatalkan FRS | *ButtonClick* |
| 4 | *statusLabel* | *Text* | Label yang menampilkan status persetujuan FRS | *String* |
| 5 | *namaKelas* | *Dropdown* | Input untuk memilih nama kelas | *String* |
| 6 | *pilihKelasButton* | *Button* | Tombol aksi untuk memilih kelas ketika FRS. | *ButtonClick* |

#### Customer’s Cart Page

Halaman ini digunakan untuk menampilkan daftar seluruh anak wali suatu dosen wali. Pada halaman ini dosen wali mendapatkan akses untuk masuk ke halaman frs, transkrip, dan jadwal kuliah anak wali nya. Halaman antarmuka dapat dilihat pada Gambar 3.39.



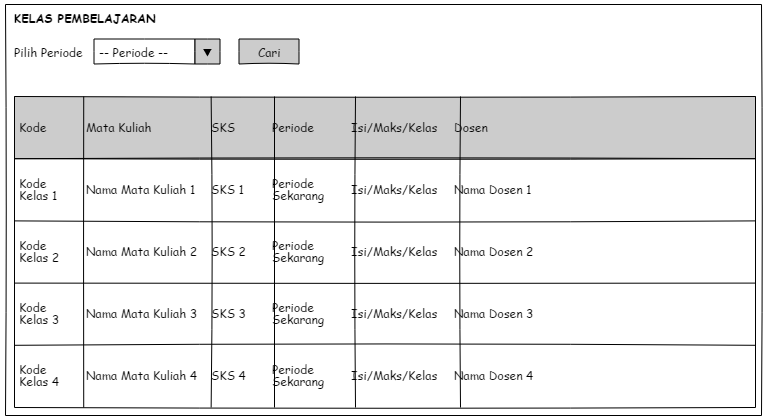
Gambar 3.39 Rancangan Antarmuka Menampilkan Anak Wali

Tabel 3.38 Penjelasan Antarmuka Menampilkan Anak Wali

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pilihPeriode* | *Dropdown* | Input untuk mengisi data periode | *String* |
| 2 | *datatablePerwalian* | *Table* | Menampilkan daftar anak wali suatu dosen | *Table* |

#### Payment Page

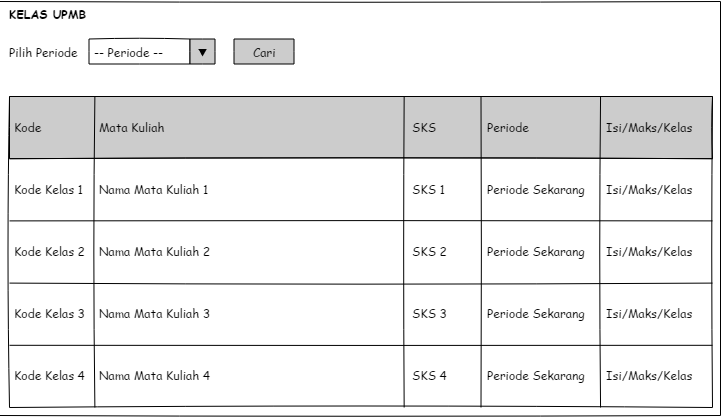
Halaman ini digunakan untuk menampilkan daftar kelas pembelajaran. Pengguna dapat memilih untuk menampilkan kelas pilihannya yaitu kelas departemen atau kelas UPMB. Halaman antarmuka dapat dilihat pada Gambar 3.40 dan Gambar 3.41.



Gambar 3.40 Rancangan Antarmuka Menampilkan Kelas Departemen

Tabel 3.39 Penjelasan Antarmuka Menampilkan Kelas Departemen

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pilihPeriode* | *Dropdown* | Input untuk mengisi data periode | *String* |
| 2 | *datatableKelas* | *Table* | Menampilkan daftar kelas pembelajaran suatu departemen | *Table* |



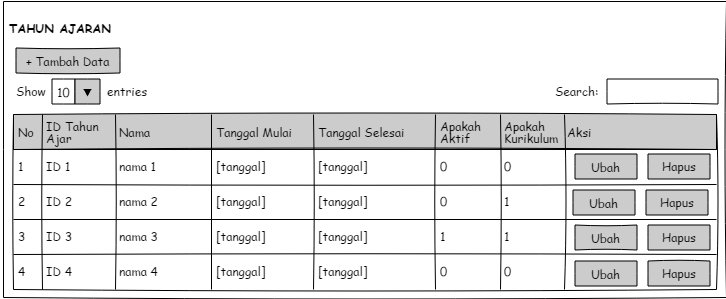
Gambar 3.41 Rancangan Antarmuka Menampilkan Kelas UPMB

Tabel 3.40 Penjelasan Antarmuka Menampilkan Kelas UPMB

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pilihPeriode* | *Dropdown* | Input untuk mengisi data periode | *String* |
| 2 | *datatableKelas* | *Table* | Menampilkan daftar kelas pembelajaran UPMB | *Table* |

#### Customer’s Ticket Page

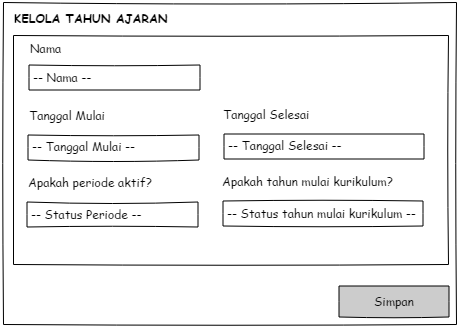
Halaman ini digunakan untuk mengelola data tahun ajaran. Pada halaman ini terdapat daftar tahun ajaran yang sudah tersimpan di basis data. Pengguna dapat memilih untuk menambah, mengubah, atau menghapus data tahun ajaran. Pengguna juga dapat mengganti status keaktifan tahun ajaran. Halaman antarmuka dapat dilihat pada Gambar 3.42 dan Gambar 3.43.



Gambar 3.42 Rancangan Antarmuka Mengelola Tahun Ajaran

Tabel 3.41 Penjelasan Antarmuka Mengelola Tahun Ajaran

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tambahButton* | *Button* | Tombol untuk membuat data tahun ajaran baru | *ButtonClick* |
| 2 | *datatableTahunAjar* | *Table* | Menampilkan data tahun ajaran yang ada | *Table* |
| 3 | *ubahButton* | *Button* | Tombol untuk mengubah data tahun ajaran | *ButtonClick* |
| 4 | *HapusButton* | *Button* | Tombol untuk menghapus data tahun ajaran | *ButtonClick* |



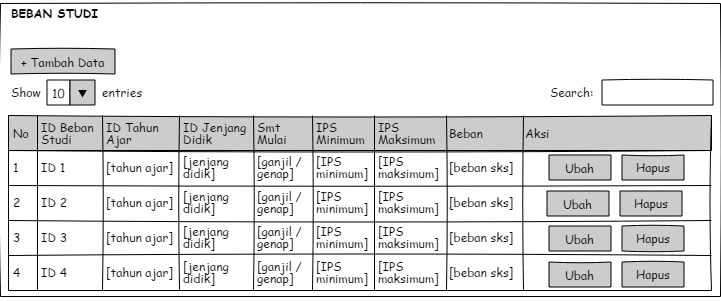
Gambar 3.43 Rancangan Antarmuka *Form* Tahun Ajaran

Tabel 3.42 Penjelasan Antarmuka *Form* Tahun Ajaran

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *namaTahunAjar* | *Text* | Input untuk mengisi data nama tahun ajar | *String* |
| 2 | *tanggalMulai* | *Date* | Input untuk mengisi tanggal mulai tahun ajar | *Date* |
| 3 | *tanggalSelesai* | *Date* | Input untuk mengisi tanggal selesai tahun ajar | *Date* |
| 4 | *statusAktif* | *Number* | Input untuk mengisi status keaktifan tahun ajar | *Integer* |
| 5 | *statusTahunKurikulum* | *Number* | Input untuk mengisi status keaktifan tahun kurikulum | *Integer* |
| 6 | *simpanButton* | *Button* | Tombol untuk menyimpan data kelola tahun ajar | *ButtonClick* |

#### Customer’s Profile Page

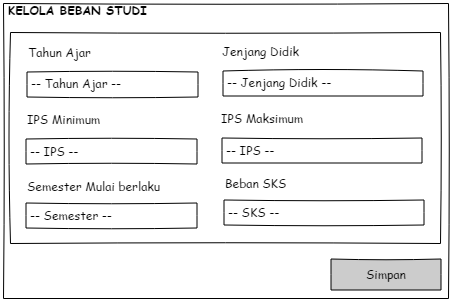
Halaman ini digunakan untuk mengelola data beban studi. Pada halaman ini terdapat daftar beban studi yang sudah tersimpan di basis data. Pengguna dapat memilih untuk menambah, mengubah, atau menghapus data beban studi. Halaman antarmuka dapat dilihat pada Gambar 3.44 dan 3.45.



Gambar 3.44 Rancangan Antarmuka Mengelola Beban Studi

Tabel 3.43 Penjelasan Antarmuka Mengelola Beban Studi

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tambahButton* | *Button* | Tombol untuk membuat data beban studi baru | *ButtonClick* |
| 2 | *datatableBebanStudi* | *Table* | Menampilkan daftar beban studi yang ada di basis data | *Table* |
| 3 | *ubahButton* | *Button* | Tombol untuk mengubah data beban studi | *ButtonClick* |
| 4 | *hapusButton* | *Button* | Tombol untuk menghapus data beban studi | *Button* |



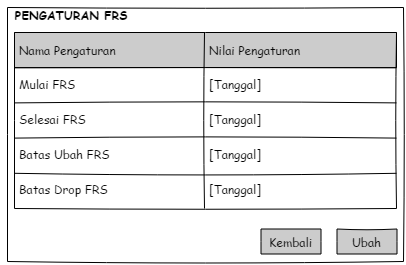
Gambar 3.45 Rancangan Antarmuka *Form* Beban Studi

Tabel 3.44 Penjelasan Antarmuka *Form* Beban Studi

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tahunAjar* | *Dropdown* | Input untuk mengisi data tahun ajaran | *String* |
| 2 | *jenjangDidik* | *Dropdown* | Input untuk mengisi data jenjang didik | *String* |
| 3 | *IPSMinimum* | *Number* | Input untuk mengisi data IPS minimum | *Number* |
| 4 | *IPSMaksimum* | *Number* | Input untuk mengisi data IPS Maksimum | *Number* |
| 5 | *semesterMulai* | *Dropdown* | Input untuk mengisi semester mulai | *String* |
| 6 | *bebanSKS* | *Number* | Input untuk mengisi beban SKS | *Integer* |
| 7 | *simpanButton* | *Button* | Tombol untuk menyimpan data kelola beban studi | *ButtonClick* |

#### Search Destination Page

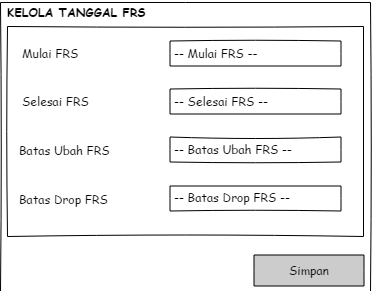
Halaman ini digunakan untuk mengelola data tanggal penting untuk FRS. Pada halaman ini terdapat data tanggal FRS yang sudah tersimpan di basis data. Pengguna dapat mengubah data tanggal FRS tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.46 dan Gambar 3.47.



Gambar 3.46 Rancangan Antarmuka Mengelola Tanggal FRS

Tabel 3.45 Penjelasan Antarmuka Mengelola Tanggal Penting FRS

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *datatableFRS* | *Table* | Menampilkan tanggal penting untuk FRS | *Table* |
| 2 | *ubahButton* | *Button* | Tombol untuk mengubah data tanggal FRS | *ButtonClick* |



Gambar 3.47 Rancangan Antarmuka *Form* Tanggal FRS

Tabel 3.46 Penjelasan Antarmuka *Form* Tanggal Penting FRS

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *mulaiFRS* | *Date* | Input untuk mengisi tanggal mulai FRS | *Date* |
| 2 | *selesaiFRS* | *Date* | Input untuk mengisi tanggal selesai FRS | *Date* |
| 3 | *batasUbahFRS* | *Date* | Input untuk mengisi tanggal batas ubah FRS | *Date* |
| 4 | *batasDropFRS* | *Date* | Input untuk mengisi tanggal batas drop FRS | *Date* |
| 5 | *simpanButton* | *Button* | Tombol untuk menyimpan tanggal FRS | *ButtonClick* |

### Business Process

This section will show business processes for the Information System. The business process module is started by an admin posts the destination on the apps. Then, the customer create an account and login to order the destination on the apps. After order destination and complete the payment, customer gets the ticket.An admn can views the order of all customers and can manage users accounts too. The business process flow is shown in Figure 3.53.

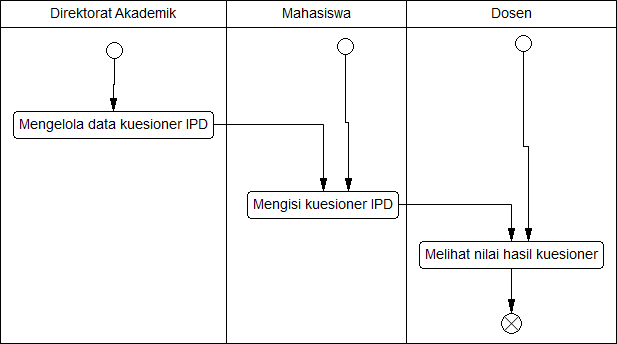


Figure 3.53 Business Process

*[Halaman ini sengaja dikosongkan]*

# CHAPTER IV IMPLEMENTATION

This chapter will discuss the implementation of the system based on business process that has been explained before.



## Implementation Environment

The system implementation environment that used to develop the final project has specifications as shown on Table 4.1.

Tabel 4.1 Lingkungan Implementasi Sistem

|  |  |
| --- | --- |
| **Device** | **Specifications** |
| Hardware | Prosesor: Intel® Core™ i5-7400 CPU @ 3.00GHz (4 CPUs) , ~3.0GHz  Memory: 8192 MB |
| Software | Operating System: Microsoft Windows 10 Pro 64-bit  Perangkat Pengembang: Phalcon  Perangkat Perancang Diagram:  Sybase Power Designer 16  Perangkat Database:  Microsoft SQL Server 2017 |

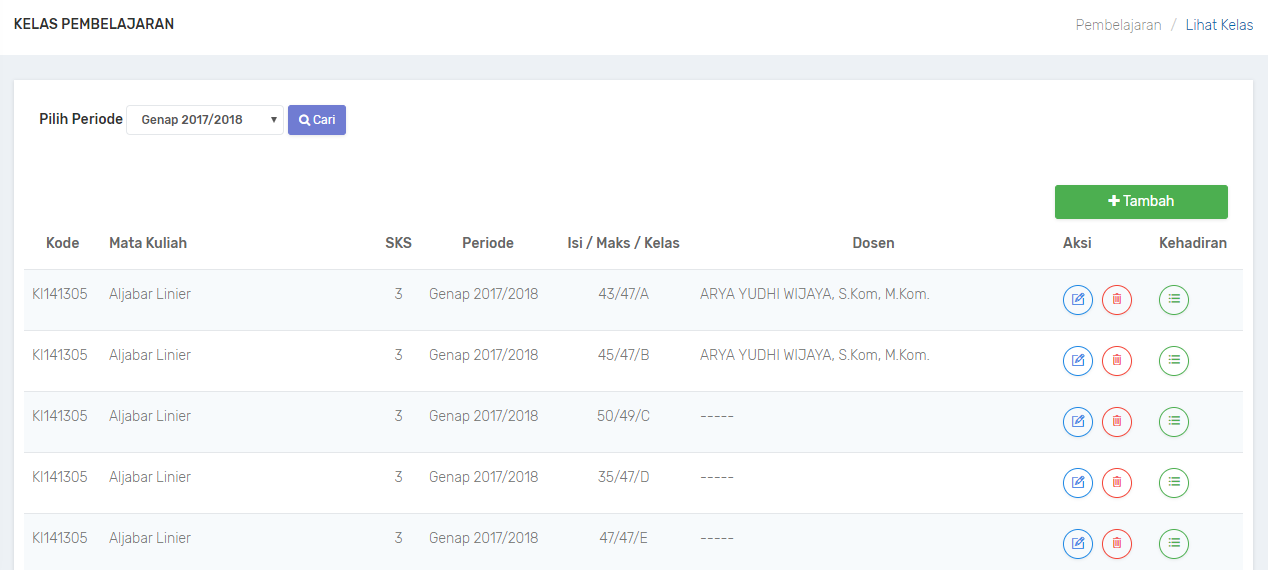
## Implementation of User Inteface

Implementation of the user interface is using a vue file and blade file for each page. The following will be discussed regarding the implementation of the system that has been realized.

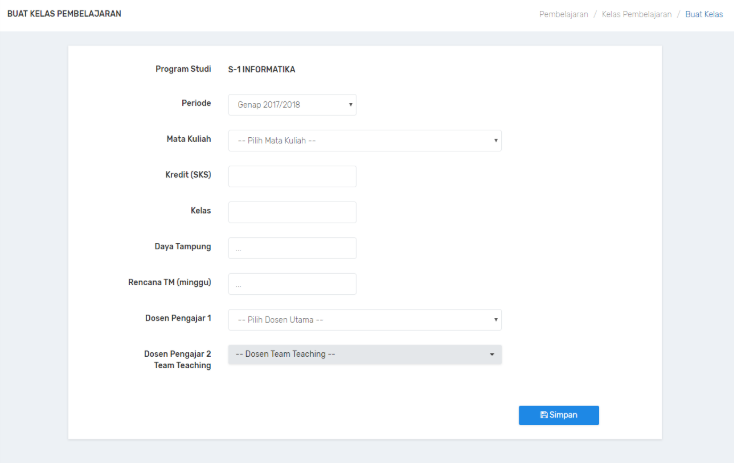
### Mobile User Interface with Vue JS

#### Halaman Mengelola Kelas Pembelajaran

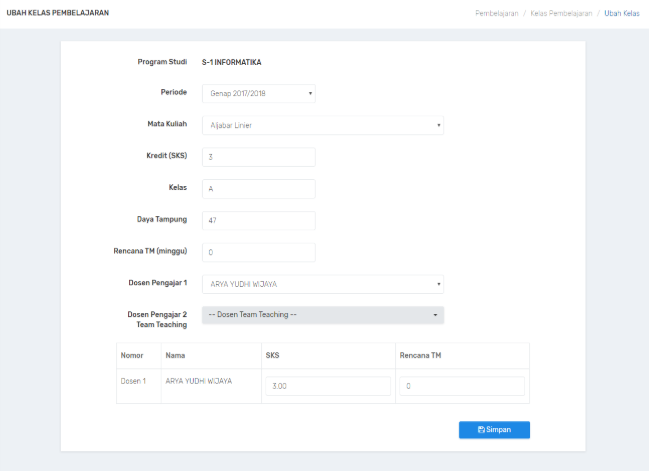
Halaman ini merupakan halaman yang digunakan untuk mengelola kelas pembelajaran. Halaman antarmuka yang menampilkan kelas pembelajaran ditunjukkan pada Gambar 4.1, menambah kelas pembelajaran ditunjukkan pada Gambar 4.2, dan pengubahan kelas ditunjukkan pada Gambar 4.3.



Gambar 4.1 Halaman Antarmuka Kelas Pembelajaran



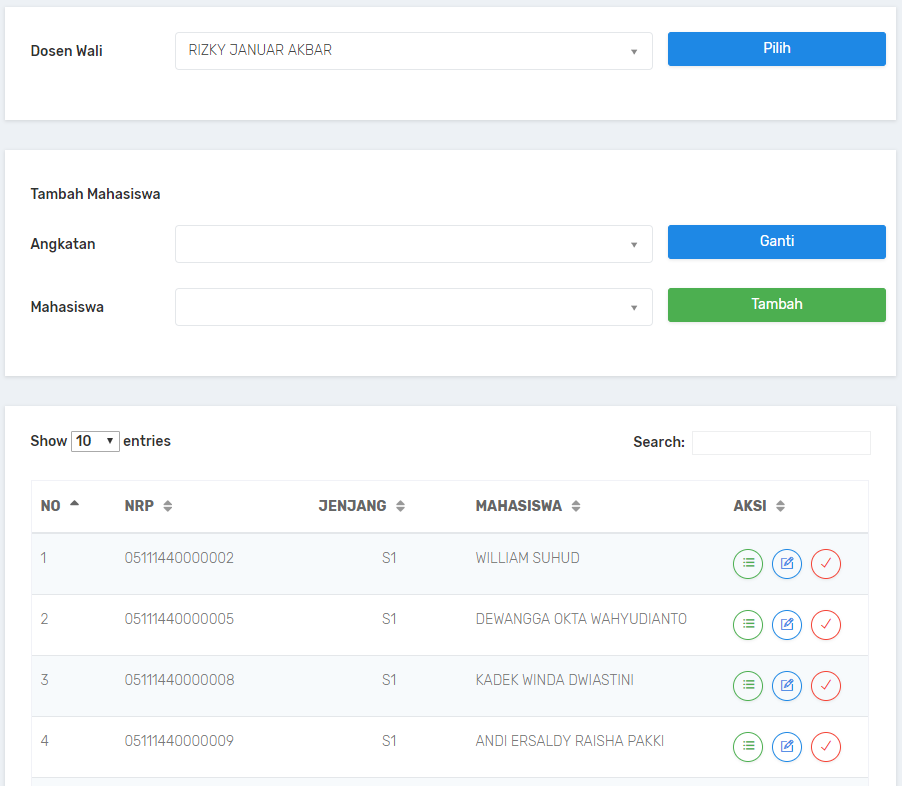
Gambar 4.2 Halaman Antarmuka *Form* Buat Kelas



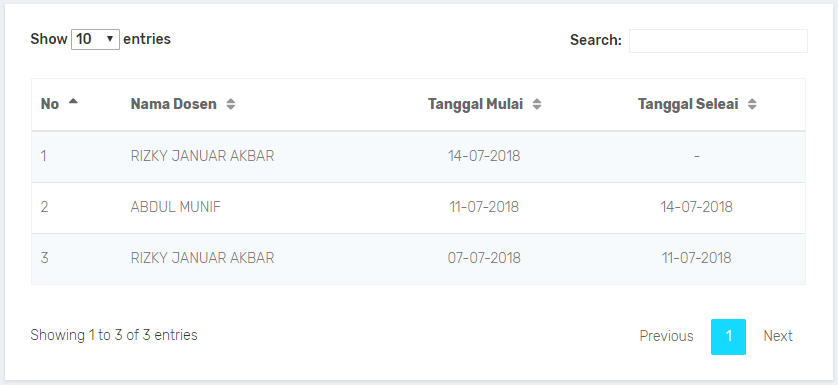
Gambar 4.3 Halaman Antarmuka *Form* Ubah Kelas

#### Halaman Mengelola Pendamping Akademik

Halaman ini merupakan halaman yang digunakan untuk mengelola kelas pendamping akademik. Pada halaman ini terdapat form untuk memilih dosen wali & anak wali dan juga terdapat tabel yang menunjukan daftar anak wali dari dosen yang dipilih seperti yang ditunjukkan pada Gambar 4.4 dan halaman riwayat dosen wali mahasiswa ditunjukkan pada Gambar 4.5.



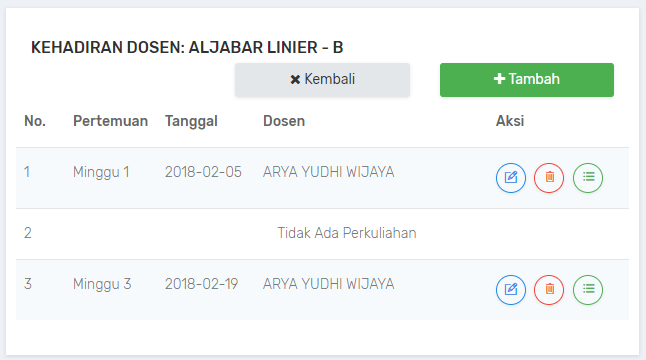
Gambar 4.4 Halaman Antarmuka Kelola Pendamping Akademik



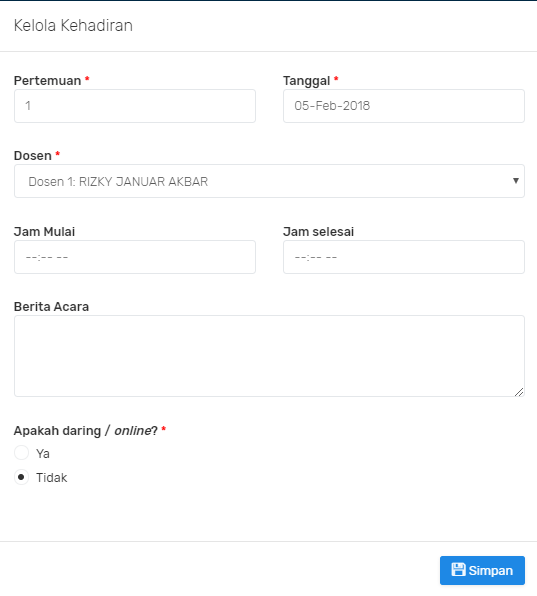
Gambar 4.5 Halaman Antarmuka Riwayat Perwalian Mahasiswa

#### Halaman Mengelola Kehadiran Dosen

Halaman ini merupakan halaman yang digunakan untuk mengelola kehadiran dosen pada suatu kelas pembelajaran. Halaman antarmuka yang menampilkan daftar kehadiran dosen tersebut ditunjukkan pada Gambar 4.6 dan pengelolaan data kehadiran dosen ditunjukkan pada Gambar 4.7.



Gambar 4.6 Halaman Antarmuka Kehadiran Dosen



Gambar 4.7 Halaman Antarmuka *Form* Kehadiran Dosen

#### Halaman Mengelola Kehadiran Mahasiswa

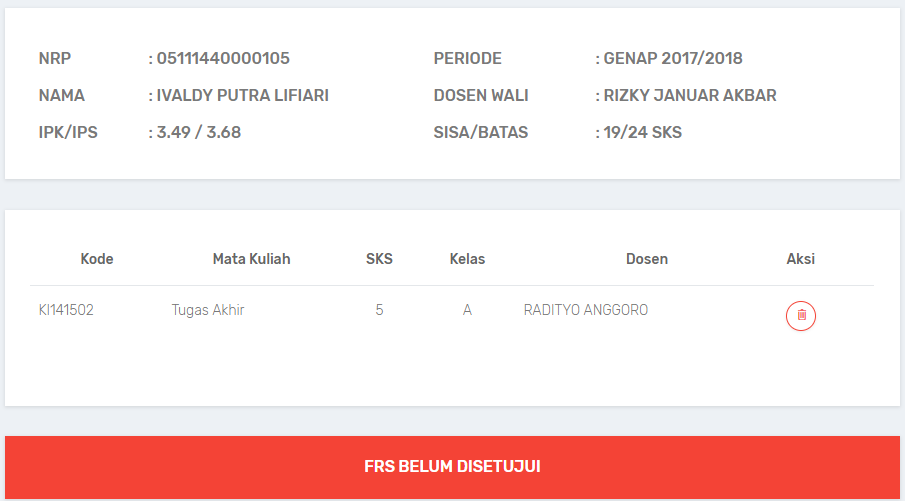
Halaman ini merupakan halaman yang digunakan untuk mengelola kehadiran mahasiswa pada suatu kelas pembelajaran. Halaman antarmuka yang menampilkan daftar kehadiran mahasiswa ditunjukkan pada Gambar 4.8.



Gambar 4.8 Halaman Antarmuka Kehadiran Mahasiswa

#### Halaman Menyusun FRS

Halaman ini merupakan halaman yang digunakan untuk menyusun FRS. Kelas yang dipilih ketika FRS akan muncul pada tabel seperti pada Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Halaman Antarmuka FRS Belum Disetujui



Gambar 4.10 Halaman Antarmuka FRS Telah Disetujui

#### Halaman Menyetujui FRS

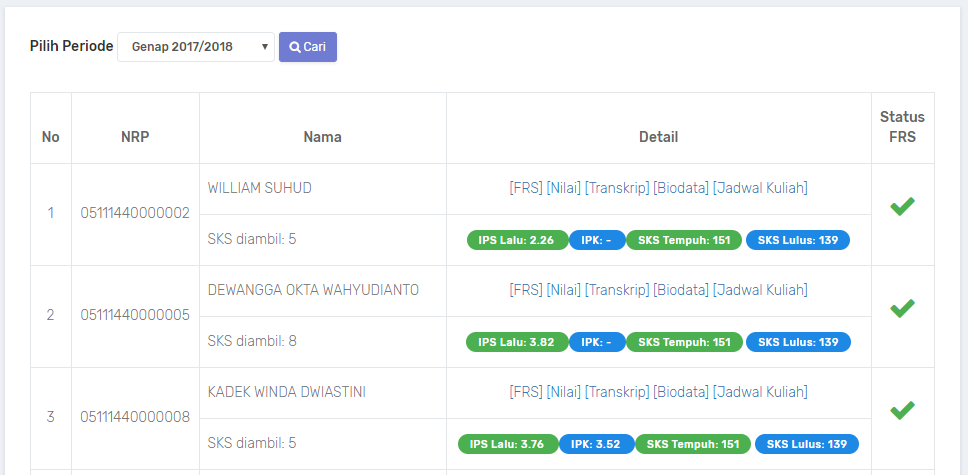
Halaman ini merupakan halaman yang digunakan untuk menyetujui halaman FRS. Pada Gambar 4.11 dosen yang bersangkutan dapat melakukan persetujuan terhadap FRS salah satu anak wali yang dipilihnya



Gambar 4.11 Halaman Antarmuka Menyetujui FRS

#### Halaman Menampilkan Anak Wali

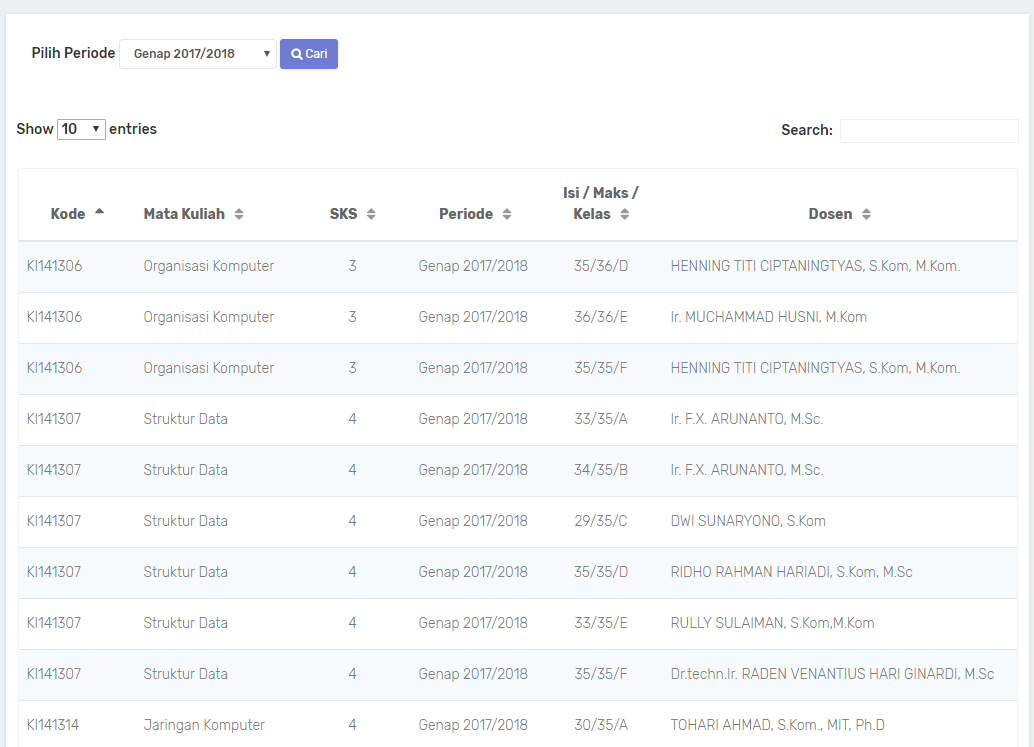
Halaman ini merupakan halaman yang digunakan untuk menampilkan daftar seluruh anak wali dari suatu dosen seperti yang ditunjukkan pada Gambar 4.12.



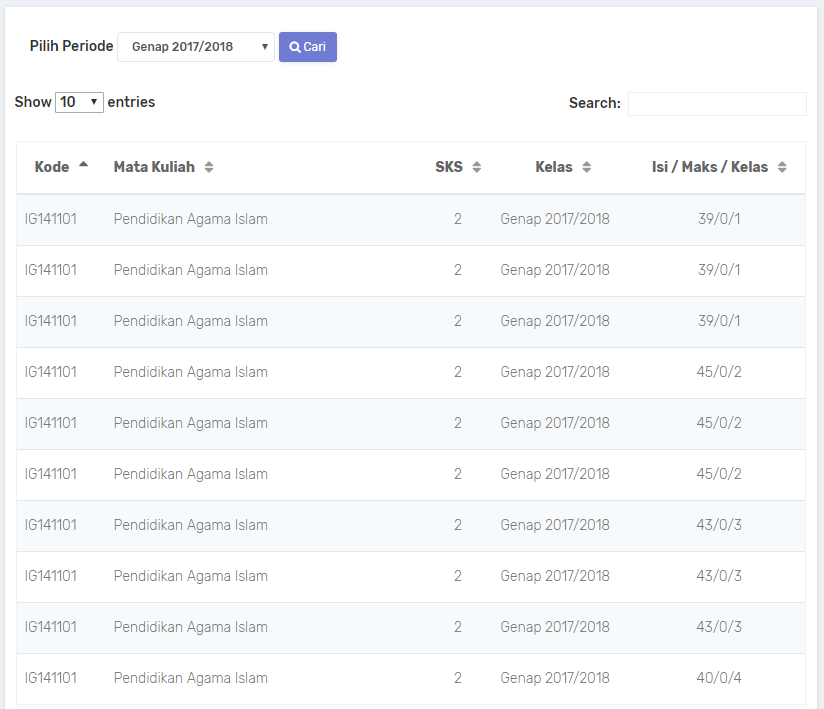
Gambar 4.12 Halaman Antarmuka Daftar Anak Wali

#### Halaman Menampilkan Kelas Departemen dan UPMB

Halaman ini merupakan halaman yang digunakan untuk menampilkan daftar seluruh kelas pembelajaran yang ada. Halaman antarmuka yang menampilkan kelas pembelajaran suatu departemen ditunjukkan pada Gambar 4.13 dan halaman yang menampilkan kelas pembelajaran UPMB ditunjukkan pada Gambar 4.14.



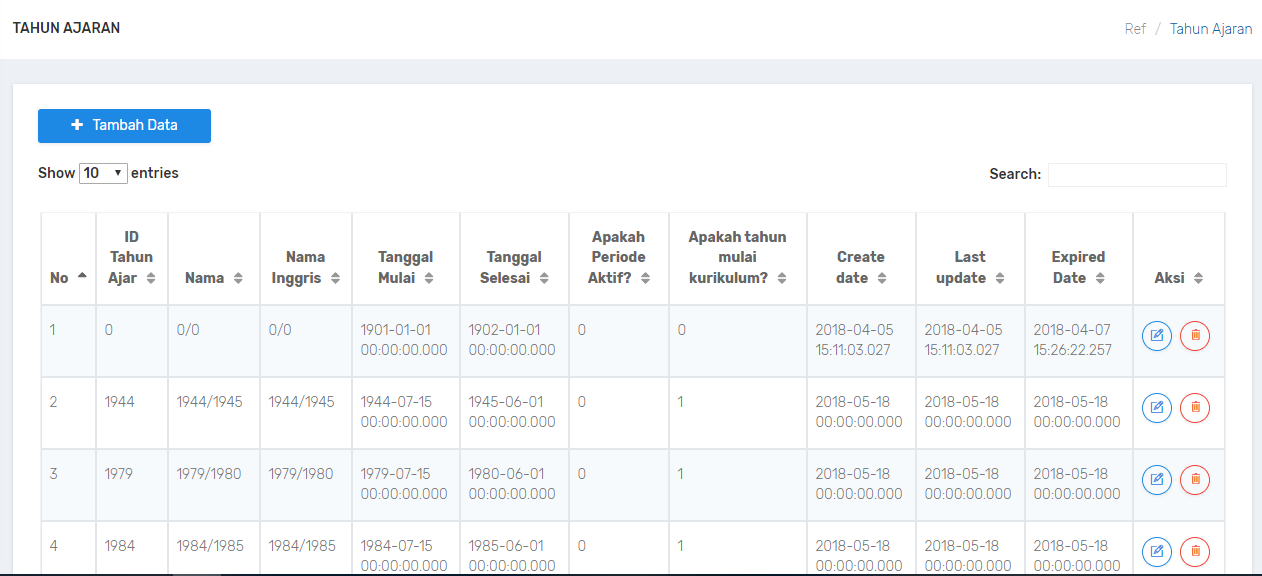
Gambar 4.13 Halaman Antarmuka Daftar Kelas Departemen



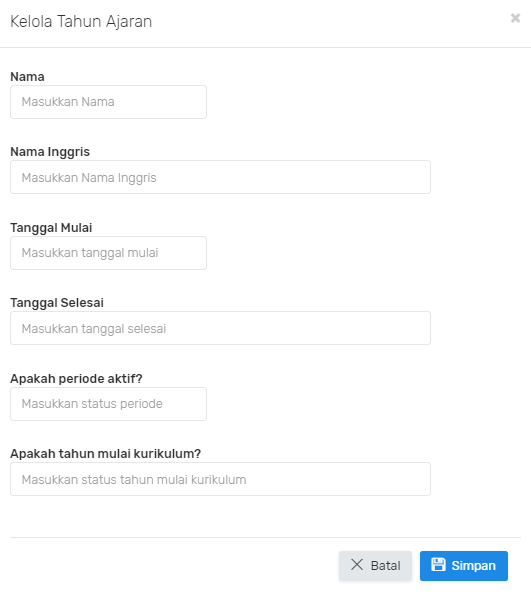
Gambar 4.14 Halaman Antarmuka Daftar Kelas UPMB

#### Halaman Mengelola Tahun Ajaran

Halaman ini merupakan halaman yang digunakan untuk mengelola tahun ajaran. Halaman antarmuka yang menampilkan daftar tahun ajaran ditunjukkan pada Gambar 4.15 dan pengelolaan data tahun ajaran ditunjukkan pada Gambar 4.16.



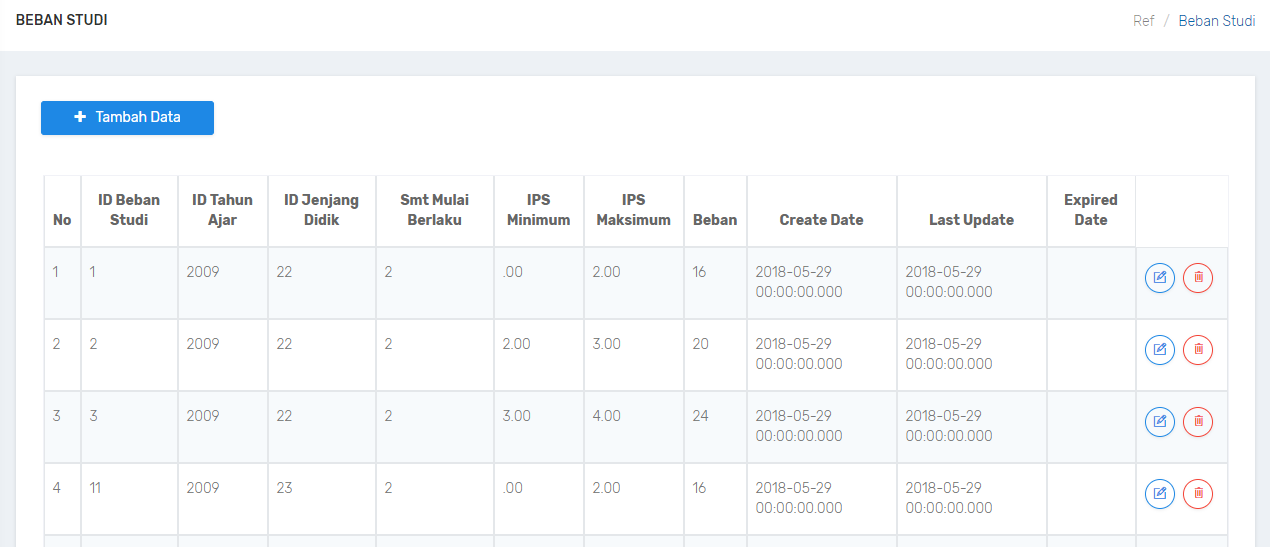
Gambar 4.15 Halaman Antarmuka Tahun Ajaran



Gambar 4.16 Halaman Antarmuka *Form* Tahun Ajaran

#### Halaman Mengelola Beban Studi

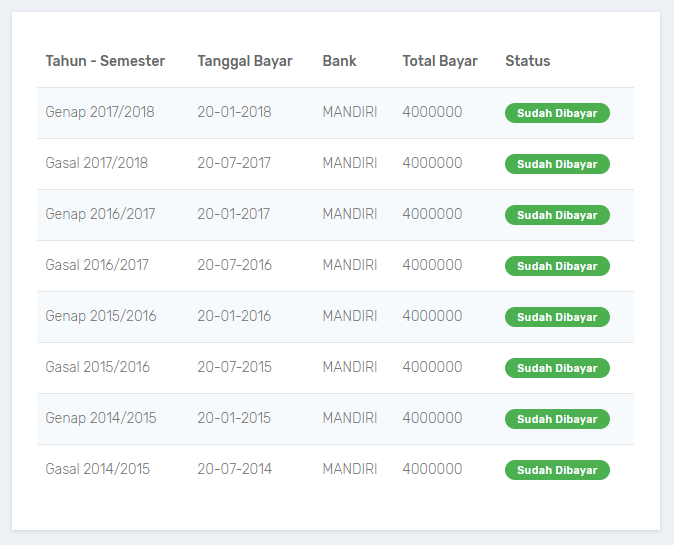
Halaman ini merupakan halaman yang digunakan untuk mengelola data beban studi. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.17.



Gambar 4.17 Halaman Antarmuka Beban Studi

#### Halaman Menampilkan Riwayat Pembayaran

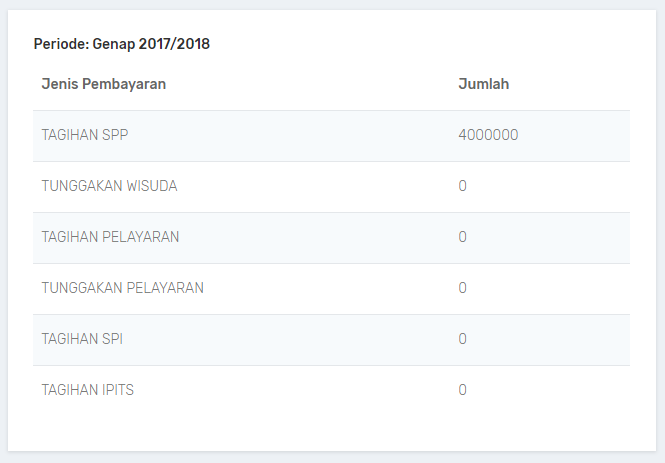
Halaman ini merupakan halaman yang digunakan untuk menampilkan riwayat pembayaran UKT milik aktor. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.18.



Gambar 4.18 Halaman Antarmuka Riwayat Pembayaran

#### Halaman Menampilkan Tagihan

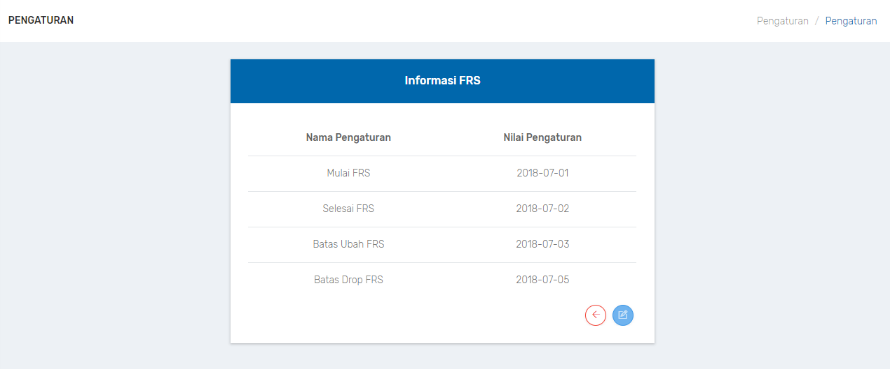
Halaman ini merupakan halaman yang digunakan untuk menampilkan data tagihan milik aktor pada suatu periode. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.19.



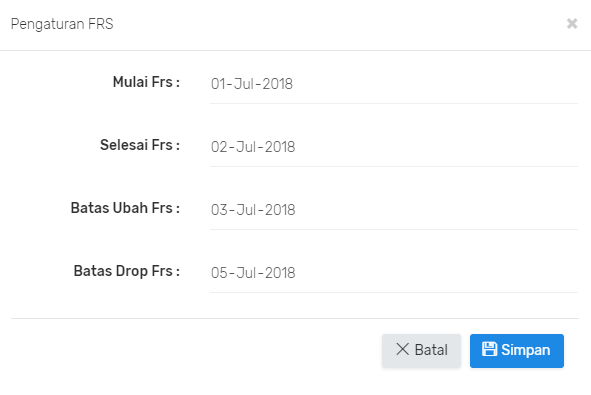
Gambar 4.19 Halaman Antarmuka Tagihan

#### Halaman Mengelola Tanggal Penting FRS

Halaman ini merupakan halaman yang digunakan untuk tanggal penting untuk melakukan FRS. Halaman antarmuka yang tanggal tersebut ditunjukkan pada Gambar 4.20 dan pengelolaan data tanggal tersebut ditunjukkan pada Gambar 4.21.



Gambar 4.20 Halaman Antarmuka Tanggal Penting FRS



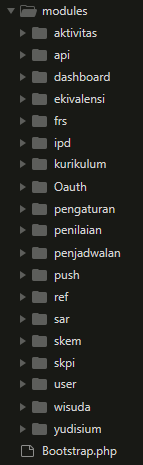
Gambar 4.21 Halaman Antarmuka *Font* Tanggal Penting FRS

## Implementasi *Repository-Service Pattern* pada Arsitektur *Model-View-Controller*

Sistem yang dibuat memiliki lapisan-lapisan yang direpresentasikan dalam kelas, yaitu *view* sebagai lapisan antarmuka pengguna, *controller* sebagai tempat untuk menerima *request* yang dikirim oleh aplikasi *client* atau *browser*, *service* sebagai tempat pemrosesan data komputasi, *repository* sebagai tempat untuk melakukan pengelolaan terhadap basis data dan *entity* sebagai representasi dari setiap tabel di basis data beserta relasinya.

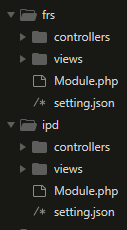
Implementasi MVC pada aplikasi dilakukan dengan pengadaan *package controller* yang berisikan kelas-kelas *controller*, *package service* yang berisikan kelas-kelas *service*, *package repository* yang berisikan kelas-kelas *repository* dan *package domain* yang berisikan implementasi basis data.

Gambar 4.28 menunjukkan struktur direktori yang berisi modul-modul yang ada



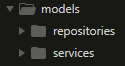
Gambar 4.28 Struktur direktori pada *folder* *modules*

Didalam tiap direktori terdapat *controller* dan *view* untuk masing-masing modul seperti yang ditunjukkan pada Gambar 4.29.



Gambar 4.29 Struktur *controller* dan *view* untuk tiap modul

Sementara Gambar 4.30 menunjukkan direktori *models* yang berisi *service* yang merupakan tempat penyimpanan *logic* dan berisi *repository* yang merupakan tempat penyimpanan *query* untuk menyambungkan ke basis data.



Gambar 4.30 Struktur *service* dan *repository* untuk tiap modul

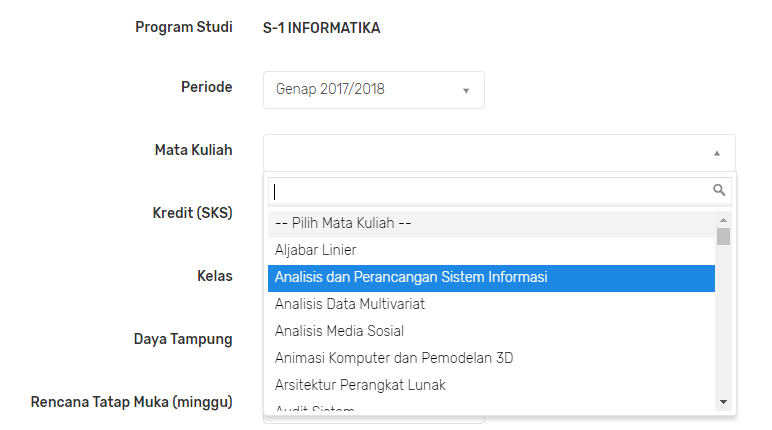
Seperti yang telah ditunjukkan oleh Gambar 3.55, modul FRS memiliki relasi dengan modul kurikulum dan modul penjadwalan. Kode Sumber 4.1 menunjukkan implementasi kode pada *controller* untuk memanggil *service* yang memanggil data mata kuliah. Kode Sumber 4.2 menunjukkan implementasi kode pada *service* untuk memanggil *repository* yang memanggil data mata kuliah. Gambar 4.31 menunjukkan implementasi relasi antar modul pembelajaran dan modul kurikulum dengan cara mengambil data mata kuliah ketika melakukan pembuatan kelas baru.

|  |
| --- |
| 1. Services::getService('Pembelajaran')->getMk($id\_satker); |

Kode Sumber 4.1 Fungsi pada *controller* untuk memanggil data mata kuliah

|  |
| --- |
| 1. Repositories::getRepository('Pembelajaran')->getKelas($id\_satker, $id\_smt); |

Kode Sumber 4.2 Fungsi pada *service* untuk memanggil data mata kuliah



Gambar 4.31 Relasi dengan modul kurikulum dengan cara mengambil data mata kuliah

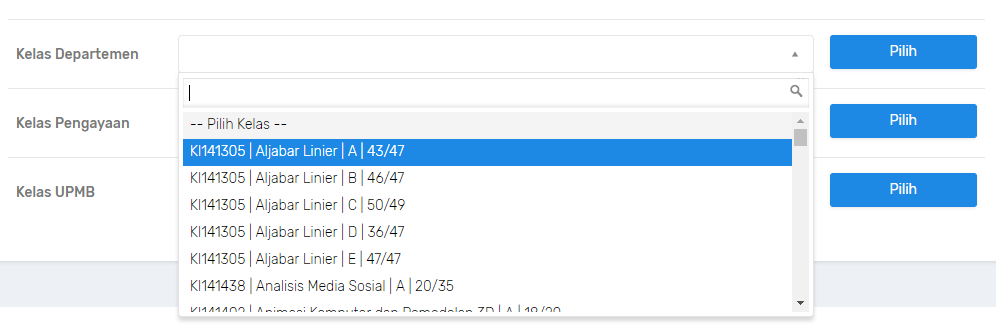
Sementara Kode Sumber 4.3 menujukkan fungsi untuk memanggil daftar kelas dari *controller* kepada *service* dan Kode Sumber 4.4 menunjukkan fungsi untuk memanggil daftar kelas dari *service* kepada *repository.* Gambar 4.32 menunjukkan implementasi relasi antar modul pembelajaran dan modul penjadwalan dengan cara mengambil data kelas yang telah diberikan jadwal.

|  |
| --- |
| 1. Services::getService('Pembelajaran')->getKelasJurusan($id\_satker, $id\_smt; 2. Services::getService('Pembelajaran')->getKelasUpmbFrs($id\_satker, $id\_smt; |

Kode Sumber 4.3 Fungsi pada *controller* untuk memanggil data kelas yang telah diberi jadwal

|  |
| --- |
| 1. Repositories::getRepository('Pembelajaran')->getKelasJurusan($id\_satker, $id\_smt); 2. Repositories::getRepository('Pembelajaran')->getKelasUpmbFrs($id\_satker, $id\_smt); |

Kode Sumber 4.4 Fungsi pada *service* untuk memanggil data kelas yang telah diberi jadwal



Gambar 4.32 Relasi dengan modul penjadwalan dengan cara mengambil kelas yang telah diberi jadwal

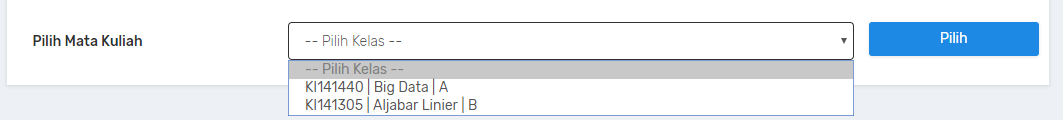
IPD juga melakukan relasi dengan modul pembelajaran dengan cara mengambil data kelas yang dipilih seorang mahasiswa ketika melakukan FRS. Kode Sumber 4.5 menunjukkan kode pada *controller* yang tersambung kepada *service* untuk memanggil data kelas yang diambil mahasiswa dan Kode Sumber 4.6 menunjukkan kode pada *service* yang tersambung kepada *repository* untuk memanggil data kelas yang diambil mahasiswa. Untuk implementasi relasi antar modul IPD dan modul pembelajaran ditunjukkan pada Gambar 4.33 dibawah.

|  |
| --- |
| 1. Services::getService('Ipd')->getKelasIpmk($id\_frs, $id\_smt); 2. Services::getService('Ipd')->getKelasIpdo($id\_frs); |

Kode Sumber 4.5 Fungsi pada *controller* untuk memanggil data kelas yang diambil mahasiswa

|  |
| --- |
| 1. Repositories::getRepository('Ipd')->getKelasIpmk($id\_frs, $id\_smt); 2. Repositories::getRepository('Ipd')->getKelasIpdo($id\_frs); |

Kode Sumber 4.6 Fungsi pada *service* untuk memanggil data kelas yang diambil mahasiswa



Gambar 4.33 Relasi antar modul pembelajaran dan modul IPD

*[Halaman ini sengaja dikosongkan]*

# CHAPTER V TESTING AND EVALUATION

Bab ini membahas pengujian dan evaluasi terhadap aplikasi SIA untuk modul pembelajaran dan IPD. Pengujian yang dilakukan adalah pengujian dalam bentuk pengujian fungsionalitas.



## Testing Environment

Lingkungan pengujian adalah lingkungan, baik perangkat keras maupun perangkat lunak tempat pengujian sistem dilakukan. Pengujian untuk modul pembelajaran dan IPD pada :

Tabel 5.1 Lingkungan Pengujian

|  |  |
| --- | --- |
| Spesifikasi | Deskripsi |
| CPU | Intel ® Core ™ i3-2120 CPU @ 3.30 GHz |
| RAM | 12.0 GB |
| Sistem Operasi | Windows 10 Enterprise 64-bit |
| Koneksi Internet | Jaringan intranet 10.199.14.36 |

## Skenario Uji Coba

Pada bagian ini akan dibahas mengenai proses uji coba yang digunakan. Pengujian dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang sudah dirancang pada sistem. Metode *black box* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya.

Pada proses uji coba, setiap peserta uji coba diminta untuk melakukan serangkaian perintah terhadap sistem yang selanjutnya akan disebut kasus pengujian. Kasus pengujian ini berkorelasi dengan kasus-kasus penggunaan dan kebutuhan fungsional yang sebelumnya sudah dirancang dan dijelaskan pada Bab III.

### Kasus Pengujian Mengelola Kelas Pembelajaran

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran Kepala Prodi dapat melakukan pengelolaan kelas pembelajaran. Terdapat empat skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.2.

Tabel 5.2 Rincian Pengujian Fungsi Mengelola Kelas

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola kelas pembelajaran |
| Kode | UJ-001 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data kelas |
| *Skenario 1* | *Kepala Prodi membuat kelas baru* |
| Kondisi Awal | Kelas pembelajaran belum dibuat |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kelas 2. Pengguna menekan tombol buat kelas baru 3. Pengguna mengisi data kelas baru 4. Pengguna menekan tombol simpan |
| Masukan | 1. Periode 2. Satker 3. Mata Kuliah 4. SKS 5. Nama Kelas 6. Daya Tampung 7. Rencana Tatap Muka 8. Nama Dosen Pengajar Utama 9. Nama Dosen Pengajar Team Teaching |
| Hasil yang diharapkan | Data kelas baru berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data kelas baru berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Kepala Prodi mengubah data kelas* |
| Kondisi Awal | Kelas pembelajaran belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kelas 2. Pengguna memilih salah satu kelas yang sudah ada 3. Pengguna mengubah data kelas tersebut 4. Pengguna menekan tombol simpan |
| Masukan | 1. Periode 2. Satker 3. Mata Kuliah 4. SKS 5. Nama Kelas 6. Daya Tampung 7. Rencana Tatap Muka 8. Nama Dosen Pengajar Utama 9. Nama Dosen Pengajar Team Teaching |
| Hasil yang diharapkan | Data kelas berhasil diubah |
| Hasil yang diperoleh | Data kelas berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Kepala Prodi menghapus data kelas* |
| Kondisi Awal | Kelas pembelajaran ada pada basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kelas 2. Pengguna memilih salah satu kelas yang sudah ada 3. Pengguna memilih untuk menghapus data kelas tersebut |
| Masukan | ID Kelas |
| Hasil yang diharapkan | Data kelas berhasil dihapus |
| Hasil yang diperoleh | Data kelas berhasil dihapus |
| Hasil pengujian | Berhasil |
| *Skenario 4* | *Kepala Prodi ingin melihat daftar kelas pada periode tertentu* |
| Kondisi Awal | Data kelas pembelajaran ada di dalam basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kelas 2. Pengguna memilih untuk mengganti periode 3. Pengguna menekan tombol cari |
| Masukan | Periode |
| Hasil yang diharapkan | Data kelas sesuai periode tertentu berhasil ditampilkan |
| Hasil yang diperoleh | Data kelas sesuai periode tertentu berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Pendamping Akademik

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran Kepala Prodi dapat mengelola data pendamping akademik. Terdapat empat skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.3

Tabel 5.3 Rincian Pengujian Fungsi Mengelola Pendamping Akademik

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola pendamping akademik |
| Kode | UJ-002 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data pendamping akademik |
| *Skenario 1* | *Kepala Prodi menempatkan mahasiswa sebagai anak wali suatu dosen* |
| Kondisi Awal | Mahasiswa belum memiliki dosen wali |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola pendamping akademik 2. Pengguna memilih dosen sebagai dosen wali 3. Pengguna memilih mahasiswa dari *dropdown* |
| Masukan | 1. ID Dosen 2. ID Mahasiswa |
| Hasil yang diharapkan | Mahasiswa yang dipilih berhasil memiliki dosen wali |
| Hasil yang diperoleh | Mahasiswa yang dipilih berhasil memiliki dosen wali |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Mahasiswa yang dipilih telah memiliki dosen wali* |
| Kondisi Awal | Mahasiswa telah memiliki dosen wali |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola pendamping akademik 2. Pengguna memilih dosen wali 3. Pengguna memilih mahasiswa dari *dropdown* |
| Masukan | 1. ID Dosen 2. ID Mahasiswa |
| Hasil yang diharapkan | Muncul notifikasi bahwa mahasiswa yang dipilih sudah memiliki dosen wali |
| Hasil yang diperoleh | Muncul notifikasi bahwa mahasiswa yang dipilih sudah memiliki dosen wali |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Kepala Prodi mengganti dosen wali suatu mahasiswa* |
| Kondisi Awal | Mahasiswa telah memiliki dosen wali |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola pendamping akademik 2. Pengguna memilih dosen wali 3. Pengguna memilih salah satu mahasiswa dari daftar anak wali 4. Pengguna memilih untuk mengubah dosen wali anak tersebut 5. Pengguna memilih dosen wali baru 6. Pengguna menekan tombol simpan |
| Masukan | 1. ID Dosen 2. ID Mahasiswa |
| Hasil yang diharapkan | Dosen wali telah berhasil diubah |
| Hasil yang diperoleh | Dosen wali telah berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 4* | *Kepala Prodi ingin menghentikan perwalian suatu anak wali* |
| Kondisi Awal | Mahasiswa telah memiliki dosen wali |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola pendamping akademik 2. Pengguna memilih dosen wali 3. Pengguna memilih salah satu mahasiswa dari daftar anak wali 4. Pengguna memilih untuk menghentikan perwalian anak wali tersebut |
| Masukan | ID Dosen Wali |
| Hasil yang diharapkan | Data perwalian anak tersebut berhasil dihapus |
| Hasil yang diperoleh | Data perwalian anak tersebut berhasil dihapus |
| Hasil pengujian | Berhasil. |

### Kasus Pengujian Mengelola Kehadiran Dosen

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran Petugas Tata Usaha dan Dosen dapat mengelola data kehadiran dosen. Terdapat tiga skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.4.

Tabel 5.4 Rincian Pengujian Fungsi Mengelola Kehadiran Dosen

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola kehadiran dosen |
| Kode | UJ-003 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data kehadiran dosen |
| *Skenario 1* | *Petugas Tata Usaha atau Dosen membuat kehadiran dosen baru* |
| Kondisi Awal | Kehadiran dosen belum dibuat |
| Prosedur Pengujian | 1. Pengguna membuka menu kehadiran dosen 2. Pengguna menekan tombol buat kehadiran baru 3. Pengguna mengisi data kehadiran 4. Pengguna menekan tombol simpan |
| Masukan | 1. Minggu Pertemuan 2. Tanggal Pertemuan 3. Dosen Pengajar 4. Jam Mulai 5. Jam Selesai 6. Berita Acara 7. Status Kelas Daring |
| Hasil yang diharapkan | Data kehadiran dosen berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data kehadiran dosen berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Tata Usaha atau Dosen mengubah data kehadiran dosen* |
| Kondisi Awal | Kehadiran dosen belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kehadiran dosen 2. Pengguna memilih salah satu kehadiran yang sudah ada 3. Pengguna mengubah data kehadiran tersebut 4. Pengguna menekan tombol simpan |
| Masukan | 1. Minggu Pertemuan 2. Tanggal Pertemuan 3. Dosen Pengajar 4. Jam Mulai 5. Jam Selesai 6. Berita Acara 7. Status Kelas Daring |
| Hasil yang diharapkan | Data kehadiran dosen berhasil diubah |
| Hasil yang diperoleh | Data kehadiran dosen berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Petugas Tata Usaha atau Dosen menghapus data kehadiran dosen* |
| Kondisi Awal | Data kehadiran dosen ada pada basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kehadiran dosen 2. Pengguna memilih salah satu kehadiran dosen yang sudah ada 3. Pengguna memilih untuk menghapus data kehadiran tersebut |
| Masukan | 1. ID Tatap Muka Kuliah 2. ID Hadir Dosen |
| Hasil yang diharapkan | Data kehadiran berhasil dihapus |
| Hasil yang diperoleh | Data kehadiran berhasil dihapus |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Kehadiran Mahasiswa

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran Petugas Tata Usaha dan Dosen dapat mengelola data kehadiran mahasiswa. Terdapat dua skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.5.

Tabel 5.5 Rincian Pengujian Fungsi Mengelola Kehadiran Mahasiswa

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola kehadiran mahasiswa |
| Kode | UJ-004 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data kehadiran mahasiswa |
| *Skenario 1* | *Petugas Tata Usaha atau Dosen membuat kehadiran mahasiswa baru* |
| Kondisi Awal | Kehadiran mahasiswa belum tersimpan di dalam basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kehadiran mahasiswa 2. Pengguna memilih jenis kehadiran mahasiswa 3. Pengguna mengisi alasan untuk data kehadiran mahasiswa 4. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. Jenis kehadiran 3. Alasan |
| Hasil yang diharapkan | Data kehadiran mahasiswa berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data kehadiran mahasiswa berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Tata Usaha atau Dosen mengubah data kehadiran mahasiswa* |
| Kondisi Awal | Kehadiran mahasiswa belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kehadiran mahasiswa 2. Pengguna mengubah data kehadiran mahasiswa 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. Jenis kehadiran 3. Alasan |
| Hasil yang diharapkan | Data kehadiran mahasiswa berhasil diubah |
| Hasil yang diperoleh | Data kehadiran mahasiswa berhasil diubah |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menyusun FRS

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran mahasiswa dapat melakukan penyusunan FRS. Terdapat tujuh skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.6.

Tabel 5.6 Rincian Pengujian Fungsi Menyusun FRS

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menyusun FRS |
| Kode | UJ-005 |
| Tujuan Pengujian | Menguji fungsi penyusunan FRS |
| *Skenario 1* | *Mahasiswa menyusun FRS* |
| Kondisi Awal | Frs belum disusun |
| Prosedur Pengujian | 1. Pengguna membuka halaman FRS 2. Pengguna memilih kelas yang diinginkan 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Data FRS berhasil tersimpan di basis data |
| Hasil yang diperoleh | Data FRS berhasil tersimpan di basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Mahasiswa belum membayar UKT ketika mengisi FRS* |
| Kondisi Awal | FRS belum disusun |
| Prosedur Pengujian | 1. Pengguna membuka halaman FRS 2. Pengguna melihat notifikasi pada halaman FRS |
| Masukan | - |
| Hasil yang diharapkan | FRS tidak dapat disusun |
| Hasil yang diperoleh | FRS tidak dapat disusun |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Kelas telah penuh ketika mahasiswa memilih kelas* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas yang kuota nya telah penuh 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas tidak dapat dipilih |
| Hasil yang diperoleh | Kelas tidak dapat dipilih |
| Hasil pengujian | Berhasil |
| *Skenario 4* | *SKS tidak cukup untuk memilih kelas* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas tidak dapat dipilih |
| Hasil yang diperoleh | Kelas tidak dapat dipilih |
| Hasil pengujian | Berhasil |
| *Skenario 5* | *Mahasiswa mengambil mata kuliah semester dibawahnya* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas berhasil dipilih dan muncul *background* warna kuning di sekitar kelas tersebut |
| Hasil yang diperoleh | Kelas berhasil dipilih dan muncul *background* warna kuning di sekitar kelas tersebut |
| Hasil pengujian | Berhasil |
| *Skenario 6* | *Mahasiswa mengambil mata kuliah semester diatasnya* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas berhasil dipilih dan muncul *background* warna merah di sekitar kelas tersebut |
| Hasil yang diperoleh | Kelas berhasil dipilih dan muncul *background* warna merah di sekitar kelas tersebut |
| Hasil pengujian | Berhasil |
| *Skenario 7* | *Mahasiswa mengambil mata kuliah bersama / UPMB* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas berhasil dipilih |
| Hasil yang diperoleh | Kelas berhasil dipilih |
| Hasil pengujian | Berhasil |
| *Skenario 8* | *Mahasiswa mengambil mata kuliah dengan jadwal yang bertabrakan* |
| Kondisi Awal | FRS sedang disusun |
| Prosedur Pengujian | 1. Pengguna sedang membuka halaman FRS 2. Pengguna memilih kelas 3. Pengguna menekan tombol simpan |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Terdapat notifikasi bahwa jadwal kelas tersebut bertabrakan |
| Hasil yang diperoleh | Terdapat notifikasi bahwa jadwal kelas tersebut bertabrakan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mencetak FRS

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran mahasiswa dapat mencetak FRS. Terdapat dua skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.7.

Tabel 5.7 Rincian Pengujian Fungsi Mencetak FRS

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mencetak FRS |
| Kode | UJ-006 |
| Tujuan Pengujian | Menguji fungsi mencetak FRS |
| *Skenario 1* | *FRS Mahasiswa belum disetujui dosen wali* |
| Kondisi Awal | FRS telah disusun |
| Prosedur Pengujian | 1. Pengguna membuka halaman FRS |
| Masukan | - |
| Hasil yang diharapkan | Mahasiswa tidak dapat mencetak FRS |
| Hasil yang diperoleh | Mahasiswa tidak dapat mencetak FRS |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *FRS Mahasiswa telah disetujui dosen wali* |
| Kondisi Awal | FRS telah disusun |
| Prosedur Pengujian | 1. Pengguna membuka halaman FRS |
| Masukan | - |
| Hasil yang diharapkan | Mahasiswa berhasil mencetak FRS |
| Hasil yang diperoleh | Mahasiswa berhasil mencetak FRS |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menyetujui FRS

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menyetujui FRS. Terdapat 2 skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.8.

Tabel 5.8 Rincian Pengujian Fungsi Menyetujui FRS

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menyetujui FRS |
| Kode | UJ-007 |
| Tujuan Pengujian | Menguji fungsi persetujuan FRS |
| *Skenario 1* | *FRS anak wali belum disetujui dosen wali* |
| Kondisi Awal | FRS belum disetujui |
| Prosedur Pengujian | 1. Pengguna membuka halaman perwalian 2. Pengguna memilih FRS anak wali nya |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. Status FRS |
| Hasil yang diharapkan | FRS berhasil disetujui |
| Hasil yang diperoleh | FRS berhasil disetujui |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *FRS anak wali telah disetujui dosen wali* |
| Kondisi Awal | FRS telah disetujui |
| Prosedur Pengujian | 1. Pengguna membuka halaman perwalian 2. Pengguna memilih FRS anak wali nya |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. Status FRS |
| Hasil yang diharapkan | FRS berhasil dibatalkan |
| Hasil yang diperoleh | FRS berhasil dibatalkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Anak Wali

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menampilkan data anak wali nya. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.9.

Tabel 5.9 Rincian Pengujian Fungsi Menampilkan Anak Wali

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan anak wali |
| Kode | UJ-008 |
| Tujuan Pengujian | Menguji fungsi menampilkan data anak wali |
| *Skenario 1* | *Dosen menampilkan data anak wali* |
| Kondisi Awal | Data anak wali belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman perwalian |
| Masukan | ID Dosen |
| Hasil yang diharapkan | Data anak wali berhasil ditampilkan |
| Hasil yang diperoleh | Data anak wali berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Kelas Departemen

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menampilkan data kelas yang ada pada departemen nya. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.10.

Tabel 5.10 Rincian Pengujian Fungsi Menampilkan Kelas Deparetmen

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan data kelas departemen |
| Kode | UJ-009 |
| Tujuan Pengujian | Menguji fungsi menampilkan data kelas departemen |
| *Skenario 1* | *Dosen menampilkan data kelas departemen* |
| Kondisi Awal | Data kelas departemen belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman kelas departemen |
| Masukan | ID Satker |
| Hasil yang diharapkan | Data kelas departemen berhasil ditampilkan |
| Hasil yang diperoleh | Data kelas departemen berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Kelas UPMB

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menampilkan data kelas yang ada pada UPMB. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.11.

Tabel 5.11 Rincian Pengujian Fungsi Menampilkan Fungsi Kelas UPMB

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan data kelas UPMB |
| Kode | UJ-010 |
| Tujuan Pengujian | Menguji fungsi menampilkan data kelas UPMB |
| *Skenario 1* | *Dosen menampilkan data kelas UPMB* |
| Kondisi Awal | Data kelas UPMB belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman kelas UPMB |
| Masukan | ID Satker |
| Hasil yang diharapkan | Data kelas UPMB berhasil ditampilkan |
| Hasil yang diperoleh | Data kelas UPMB berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Tahun Ajaran

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran petugas biro akademik dapat melakukan pengelolaan tahun ajaran. Terdapat tiga skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.12.

Tabel 5.12 Rincian Pengujian Fungsi Mengelola Tahun Ajaran

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola tahun ajaran |
| Kode | UJ-011 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data tahun ajaran |
| *Skenario 1* | *Petugas Biro Akademik membuat tahun ajaran baru* |
| Kondisi Awal | Tahun ajaran belum dibuat |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola tahun ajaran 2. Pengguna menekan tombol buat tahun ajaran baru 3. Pengguna mengisi data tahun ajaran baru 4. Pengguna menekan tombol simpan |
| Masukan | 1. ID Tahun Ajar 2. Nama Tahun Ajar 3. Tanggal Mulai 4. Tanggal Selesai 5. Status Periode Aktif 6. Status Tahun Kurikulum |
| Hasil yang diharapkan | Data tahun ajar berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data tahun ajar berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Biro Akademik mengubah data tahun ajar* |
| Kondisi Awal | Tahun ajar belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola tahun ajaran 2. Pengguna memilih salah satu data tahun ajar yang sudah ada 3. Pengguna mengubah data tahun ajaran tersebut 4. Pengguna menekan tombol simpan |
| Masukan | 1. ID Tahun Ajar 2. Nama Tahun Ajar 3. Tanggal Mulai 4. Tanggal Selesai 5. Status Periode Aktif 6. Status Tahun Kurikulum |
| Hasil yang diharapkan | Data tahun ajaran berhasil diubah |
| Hasil yang diperoleh | Data tahun ajaran berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Petugas Biro Akademik menghapus data tahun ajar* |
| Kondisi Awal | Data tahun ajaran ada pada basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola tahun ajaran 2. Pengguna memilih salah satu tahun ajaran yang sudah ada 3. Pengguna memilih untuk menghapus data tahun ajar tersebut |
| Masukan | ID Tahun ajar |
| Hasil yang diharapkan | Data tahun ajar berhasil dihapus |
| Hasil yang diperoleh | Data tahun ajar berhasil dihapus |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Beban Studi

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran petugas direktorat akademik dapat melakukan pengelolaan data beban studi. Terdapat tiga skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.13.

Tabel 5.13 Rincian Pengujian Fungsi Mengelola Beban Studi

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola data beban studi |
| Kode | UJ-012 |
| Tujuan Pengujian | Menguji fungsi pengelolaan data beban studi |
| *Skenario 1* | *Petugas Direktorat Akademik membuat data beban studi baru* |
| Kondisi Awal | Data beban studi belum dibuat |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola beban studi 2. Pengguna menekan tombol buat beban studi baru 3. Pengguna mengisi data beban studi baru 4. Pengguna menekan tombol simpan |
| Masukan | 1. ID Tahun Ajar 2. ID Jenjang Didik 3. Semester Mulai Berlaku 4. IPS Minimum 5. IPS Maksimum 6. Beban |
| Hasil yang diharapkan | Data beban studi berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data beban studi berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Direktorat Akademik mengubah data beban studi* |
| Kondisi Awal | Data beban studi belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola beban studi 2. Pengguna memilih salah satu data beban studi yang sudah ada 3. Pengguna mengubah data beban studi tersebut 4. Pengguna menekan tombol simpan |
| Masukan | 1. Nama Tahun Ajar 2. Tanggal Mulai 3. Tanggal Selesai 4. Status Periode Aktif 5. Status Tahun Kurikulum |
| Hasil yang diharapkan | Data beban studi berhasil diubah |
| Hasil yang diperoleh | Data beban studi berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Petugas Direktorat Akademik menghapus data beban studi* |
| Kondisi Awal | Data beban studi ada pada basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola beban studi 2. Pengguna memilih salah satu beban studi yang sudah ada 3. Pengguna memilih untuk menghapus data beban studi tersebut |
| Masukan | ID Beban Studi |
| Hasil yang diharapkan | Data beban studi berhasil dihapus |
| Hasil yang diperoleh | Data beban studi berhasil dihapus |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Riwayat Pembayaran

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran mahasiswa dapat menampilkan riwayat pembayaran nya. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.14.

Tabel 5.14 Rincian Pengujian Fungsi Menampilkan Riwayat Pembayaran

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan riwayat pembayaran UKT |
| Kode | UJ-013 |
| Tujuan Pengujian | Menguji fungsi menampilkan data riwayat pembayaran UKT |
| *Skenario 1* | *Mahasiswa menampilkan data riwayat pembayaran* |
| Kondisi Awal | Data riwayat pembayaran belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman riwayat pembayaran |
| Masukan | ID Mahasiswa |
| Hasil yang diharapkan | Data riwayat pembayaran berhasil ditampilkan |
| Hasil yang diperoleh | Data riwayat pembayaran berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Tagihan

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran mahasiswa dapat menampilkan data tagihan nya. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.15.

Tabel 5.15 Rincian Pengujian Fungsi Menampilkan Tagihan

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan tagihan |
| Kode | UJ-014 |
| Tujuan Pengujian | Menguji fungsi menampilkan data tagihan |
| *Skenario 1* | *Mahasiswa menampilkan data tagihan* |
| Kondisi Awal | Data tagihan belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman tagihan |
| Masukan | ID Mahasiswa |
| Hasil yang diharapkan | Data tagihan berhasil ditampilkan |
| Hasil yang diperoleh | Data tagihan berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Tanggal Penting FRS

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran petugas biro akademik dapat melakukan pengelolaan data tanggal penting FRS. Pada kasus uji ini, rincian kasus uji ditunjukkan pada Tabel 5.16.

Tabel 5.16 Rincian Pengujian Fungsi Mengelola Tanggal FRS

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola tanggal penting FRS |
| Kode | UJ-015 |
| Tujuan Pengujian | Menguji fungsi pengelolaan tanggal penting FRS |
| *Skenario 1* | *Petugas Biro Akademik melihat data tanggal penting FRS* |
| Kondisi Awal | Data tanggal penting FRS ada di basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu pengaturan tanggal akademik 2. Pengguna memilih pengaturan tanggal FRS |
| Masukan | 1. Tanggal FRS |
| Hasil yang diharapkan | Data tanggal penting FRS berhasil ditampilkan |
| Hasil yang diperoleh | Data tanggal penting FRS berhasil ditampilkan |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Biro Akademik mengubah data tanggal penting FRS* |
| Kondisi Awal | Data tanggal penting FRS belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu pengaturan tanggal akademik 2. Pengguna memilih pengaturan tanggal penting FRS 3. Pengguna mengubah data salah satu tanggal penting FRS 4. Pengguna menekan tombol simpan |
| Masukan | 1. Tanggal FRS |
| Hasil yang diharapkan | Data tanggal FRS berhasil diubah |
| Hasil yang diperoleh | Data tanggal FRS berhasil diubah |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengelola Kuesioner IPD

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran petugas direktorat akademik dapat melakukan pengelolaan data kuesioner IPD. Terdapat tiga skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.17.

Tabel 5.17 Rincian Pengujian Fungsi Mengelola Kuesioner IPD

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengelola kuesioner IPD |
| Kode | UJ-016 |
| Tujuan Pengujian | Menguji fungsi pengelolaan kuesioner IPD |
| *Skenario 1* | *Petugas Direktorat Akademik membuat data kuesioner baru* |
| Kondisi Awal | Data kuesioner belum dibuat |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kuesioner 2. Pengguna menekan tombol buat kuesioner baru 3. Pengguna mengisi data pertanyaan dan jawaban kuesioner baru 4. Pengguna menekan tombol simpan |
| Masukan | 1. Pertanyaan Kuesioner 2. Jawaban Kuesioner |
| Hasil yang diharapkan | Data kuesioner berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data kuesioner berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Petugas Direktorat Akademik mengubah data kuesioner* |
| Kondisi Awal | Data kuesioner belum diubah |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola kuesioner 2. Pengguna memilih salah satu data kuesioner yang sudah ada 3. Pengguna mengubah data kuesioner tersebut 4. Pengguna menekan tombol simpan |
| Masukan | 1. Pertanyaan Kuesioner 2. Jawaban Kuesioner |
| Hasil yang diharapkan | Data kuesioner berhasil diubah |
| Hasil yang diperoleh | Data kuesioner berhasil diubah |
| Hasil pengujian | Berhasil |
| *Skenario 3* | *Petugas Direktorat Akademik menghapus data kuesioner* |
| Kondisi Awal | Data kuesioner ada pada basis data |
| Prosedur Pengujian | 1. Pengguna membuka menu kelola data kuesioner 2. Pengguna memilih salah satu kuesioner yang sudah ada 3. Pengguna memilih untuk menghapus data kuesioner tersebut |
| Masukan | 1. ID Kuesioner 2. ID Jawaban |
| Hasil yang diharapkan | Data kuesioner berhasil dihapus |
| Hasil yang diperoleh | Data kuesioner berhasil dihapus |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Mengisi Kuesioner IPD

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran petugas mahasiswa dapat melakukan pengisian data kuesioner IPD. Terdapat tiga skenario pada kasus uji ini dimana rincian kasus uji ditunjukkan pada Tabel 5.18.

Tabel 5.18 Rincian Pengujian Fungsi Mengisi Kuesioner IPD

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas mengisi kuesioner IPD |
| Kode | UJ-017 |
| Tujuan Pengujian | Menguji fungsi pengisian kuesioner IPD |
| *Skenario 1* | *Mahasiswa mengisi kuesioner IPD* |
| Kondisi Awal | Kuesioner belum diisi |
| Prosedur Pengujian | 1. Pengguna membuka menu kuesioner IPD 2. Pengguna memilih mata kuliah 3. Pengguna mengisi kuesioner IPD 4. Pengguna menekan tombol simpan |
| Masukan | 1. ID Kuesioner 2. Respon Kuesioner 3. ID Kelas 4. ID Mahasiswa 5. ID FRS |
| Hasil yang diharapkan | Data kuesioner berhasil disimpan didalam basis data |
| Hasil yang diperoleh | Data kuesioner berhasil disimpan didalam basis data |
| Hasil pengujian | Berhasil |
| *Skenario 2* | *Mahasiswa hanya memiliki kelas yang tidak dapat diisi IPD nya* |
| Kondisi Awal | Kuesioner belum diisi |
| Prosedur Pengujian | 1. Pengguna membuka menu kuesioner IPD 2. Pengguna memilih mata kuliah |
| Masukan | 1. ID Mahasiswa 2. ID FRS 3. ID Kelas |
| Hasil yang diharapkan | Kelas tidak dapat dipilih |
| Hasil yang diperoleh | Kelas tidak dapat dipilih |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Daftar Kuesioner

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menampilkan data daftar kuesioner. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.19.

Tabel 5.19 Rincian Pengujian Fungsi Menampilkan Daftar Kuesioner

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan daftar kuesioner |
| Kode | UJ-018 |
| Tujuan Pengujian | Menguji fungsi menampilkan daftar kuesioner |
| *Skenario 1* | *Dosen menampilkan data daftar kuesioner* |
| Kondisi Awal | Data kuesioner belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman daftar kuesioner |
| Masukan | 1. ID SDM 2. ID Satker |
| Hasil yang diharapkan | Data kuesioner berhasil ditampilkan |
| Hasil yang diperoleh | Data kuesioner berhasil ditampilkan |
| Hasil pengujian | Berhasil |

### Kasus Pengujian Menampilkan Hasil IPD Dosen

Kasus uji ini digunakan untuk menguji apakah aktor dengan peran dosen dapat menampilkan data hasil IPD nya. Pada kasus uji ini rincian kasus uji ditunjukkan pada Tabel 5.20.

Tabel 5.20 Rincian Pengujian Fungsi Menampilkan Hasil IPD Dosen

|  |  |
| --- | --- |
| Nama Skenario Pengujian | Fungsionalitas menampilkan nilai hasil IPD Dosen |
| Kode | UJ-019 |
| Tujuan Pengujian | Menguji fungsi menampilkan nilai hasil IPD dosen |
| *Skenario 1* | *Dosen menampilkan nilai hasil IPD dosen* |
| Kondisi Awal | Data nilai hasil IPD belum ditampilkan |
| Prosedur Pengujian | 1. Pengguna membuka halaman hasil nilai IPD 2. Pengguna memilih nama kelas yang diajarkannya |
| Masukan | 1. ID SDM 2. ID Kelas |
| Hasil yang diharapkan | Data nilai hasil IPD berhasil ditampilkan |
| Hasil yang diperoleh | Data nilai hasil IPD berhasil ditampilkan |
| Hasil pengujian | Berhasil |

## Rekapitulasi Kasus Pengujian

Dibawah ini merupakan rekapitulasi dari kasus pengujian yang telah dilakukan diatas. Hasil rekapitulasi kasus pengujian dari modul pembelajaran ditunjukkan pada Tabel 5.21.

Tabel 5.21 Rekapitulasi Hasil Pengujian Modul Pembelajaran

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Fungsionalitas** | **Pengujian** | **Terpenuhi** | |
| **Ya** | **Tidak** |
| 1 | Mengelola kelas pembelajaran | Membuat kelas | ✓ |  |
| Mengubah kelas | ✓ |  |
| Menghapus kelas | ✓ |  |
| Melihat kelas | ✓ |  |
| 2 | Mengelola pendamping akademik | Menempatkan anak wali | ✓ |  |
| Memilih anak wali yang telah memiliki dosen wali | ✓ |  |
| Mengganti dosen wali | ✓ |  |
| Menghentikan perwalian | ✓ |  |
| 3 | Mengelola kehadiran dosen | Membuat kehadiran | ✓ |  |
| Mengubah kehadiran | ✓ |  |
| Menghapus kehadiran | ✓ |  |
| 4 | Mengelola kehadiran mahasiswa | Membuat kehadiran | ✓ |  |
| Mengubah kehadiran | ✓ |  |
| 5 | Menyusun FRS | Mahasiswa menyusun FRS | ✓ |  |
| Mahasiswa belum membayar UKT | ✓ |  |
| Kelas terpilih belum terpenuhi | ✓ |  |
| SKS tidak memenuhi | ✓ |  |
| Mahasiswa mengambil mata kuliah semester diatasnya | ✓ |  |
| Mahasiswa mengambil mata kuliah semester dibawahnya | ✓ |  |
| Mahasiswa mengambil mata kuliah bersama | ✓ |  |
| Mahasiswa mengambil kelas yang memiliki jadwal yang bertabrakan | ✓ |  |
| 6 | Mencetak FRS | FRS belum disetujui dosen wali | ✓ |  |
| FRS sudah disetujui dosen wali | ✓ |  |
| 7 | Menyetujui FRS | FRS belum disetujui dosen wali | ✓ |  |
| FRS sudah disetujui dosen wali | ✓ |  |
| 8 | Menampilkan anak wali | Menampilkan data anak wali | ✓ |  |
| 9 | Menampilkan kelas departemen | Menampilkan data kelas departemen | ✓ |  |
| 10 | Menampilkan kelas UPMB | Menampilkan data kelas UPMB | ✓ |  |
| 11 | Mengelola tahun ajaran | Membuat tahun ajaran | ✓ |  |
| Mengubah tahun ajaran | ✓ |  |
| Menghapus tahun ajaran | ✓ |  |
| 12 | Mengelola beban studi | Membuat beban studi | ✓ |  |
| Mengubah beban studi | ✓ |  |
| Menghapus beban studi | ✓ |  |
| 13 | Menampilkan riwayat pembayaran | Menampilkan data riwayat pembayaran | ✓ |  |
| 14 | Menampilkan tagihan | Menampilkan data tagihan | ✓ |  |
| 15 | Mengelola tanggal penting FRS | Menampilkan data tanggal penting FRS | ✓ |  |
| Mengubah tanggal penting FRS | ✓ |  |

Hasil rekapitulasi kasus pengujian pada modul IPD ditunjukkan pada Tabel 5.22.

Tabel 5.22 Rekapitulasi Hasil Pengujian Modul IPD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Fungsionalitas** | **Pengujian** | **Terpenuhi** | |
| **Ya** | **Tidak** |
| 1 | Mengelola kuesioner IPD | Membuat kuesioner | ✓ |  |
| Mengubah kuesioner | ✓ |  |
| Menghapus kuesioner | ✓ |  |
| 2 | Mengisi kuesioner IPD | Mahasiswa mengisi kuesioner | ✓ |  |
| Kelas yang dipilih mahasiswa tidak memiliki kuesioner | ✓ |  |
| 3 | Menampilkan daftar kuesioner | Menampilkan daftar kuesioner | ✓ |  |
| 4 | Menampilkan hasil IPD dosen | Menampilkan hasil IPD dosen | ✓ |  |

*[Halaman ini sengaja dikosongkan]*

# CHAPTER VI CONCLUSIONS AND RECOMMENDATIONS

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan tugas akhir ini. Selain itu, juga terdapat beberapa saran terhadap tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik lagi



## Conclusions

Dari hasil uji coba yang telah dilakukan terhadap pembuatan sistem informasi akademik pada modul pembelajaran dan IPD, dapat diambil kesimpulan sebagai berikut.

* + 1. Penggalian kebutuhan terhadap aktivitas dan fungsi umum SIA yang berfokus pada modul pembelajaran dan modul IPD telah berhasil dilaksanakan dengan melakukan studi literatur dan wawancara yang dilakukan kepada dosen, mahasiswa, kepala prodi, petugas biro akademik, dan petugas direktorat akademik. Telah didapatkan pula hasil data berupa poin-poin fungsional yang nantinya dapat menangani variasi proses bisnis pada perguruan tinggi.
    2. Perancangan pada modul pembelajaran dan modul IPD dilakukan dengan melihat hasil studi literatur, wawancara dan analisis kebutuhan yang telah dilakukan pada para aktor yang bersangkutan.
    3. Modul pembelajaran dan modul IPD dapat berkomunikasi dengan modul lainnya dengan cara memanfaatkan lapisan *service* dan *repository* pada SIA.
    4. Validasi terhadap modul pembelajaran dan modul IPD dilakukan dengan cara menguji dan menunjukkan SIA yang telah dibuat kepada para aktor yang bersangkutan.

## Recommendations

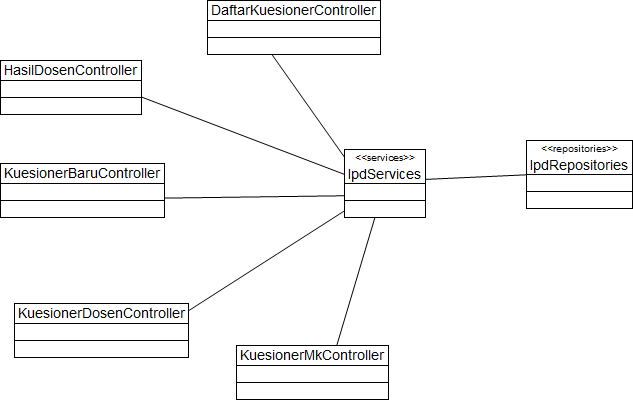
Terdapat beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Dalam merancang perangkat lunak yang sifatnya generik, selain melalakukan riset dalam aspek fungsionalitas, seharusnya juga dilakukan riset mengenai aspek antarmuka.
2. Komponen-komponen penyusun nilai pada fungsionalitas pengelolaan nilai kedepannya dapat dikolerasikan dengan capaian-capaian pembelajaran pada masing-masing mata kuliah.
3. Perlunya penggalian kebutuhan lebih lanjut yang berkoordinasi dengan para pengguna terkait untuk pengembangan sistem kedepannya.

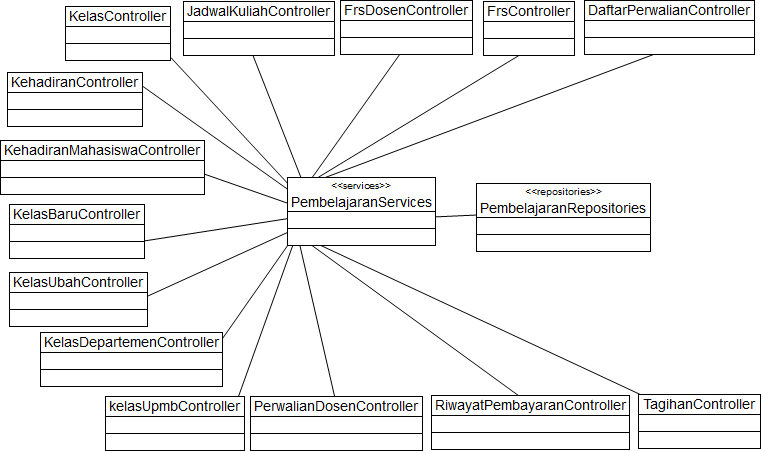
# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | "Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 Tahun 2014 tentang Standar Nasional pendidikan Tinggi," Republik Indonesia, 2014. |
| [2] | "Satuan Kredit Semester," 2017 Januari 26. [Online]. Available: https://id.wikipedia.org/wiki/Satuan\_kredit\_semester. [Accessed 2017 Desember 26]. |
| [3] | D. Prayogo, "Evaluasi Kinerja Aplikasi Indeks Pengajaran Dosen," Digital Library Institut Teknologi Sepuluh Nopember, 23 Juli 2013. [Online]. Available: http://digilib.its.ac.id/ITS-paper-52021130003016/28671. [Accessed 2017 December 27]. |
| [4] | A. A. R. V. J. P. Faria, "Wiki Based Requirements Documentation of Generic Software Products," *IEEE.* |
| [5] | "Phalcon (Framework)," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Phalcon\_(framework). [Accessed 2018 July 4 ]. |
| [6] | "Phalcon PHP," Phalcon, [Online]. Available: https://olddocs.phalconphp.com/en/3.0.3/reference/volt.html. [Accessed 2018 July 4]. |
| [7] | M. Rouse, "Search SQL Server," Tech Target, [Online]. Available: https://searchsqlserver.techtarget.com/definition/SQL-Server. [Accessed 2018 July 4]. |
| [8] | D. Riehle and H. Züllighoven, "Understanding and Using Pattern in Software Development," pp. 5-6, 1996. |
| [9] | J. Pichardo, "What are Repositories, Services, and Actions/Controllers?," Stack Exchange, 10 December 2016. [Online]. Available: https://softwareengineering.stackexchange.com/questions/337274/what-are-repositories-services-and-actions-controllers. [Accessed 15 July 2018]. |
| [10] | P. Steve Burbeck, "Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC)," p. 2, 1992. |
| [11] | A. Leff and J. T. Rayfield, "Web-Application Development Using the Model/View/Controller Design Pattern," pp. 1-2, 2001. |

# APPENDIX



Gambar A. 1 Diagram Kelas Modul IPD



Gambar A. 2 Diagram Kelas Modul FRS



Gambar A. 3 Conceptual Data Model FRS



Gambar A. 4 Conceptual Data Model IPD

# BIODATA PENULIS

Penulis lahir di Jakarta, 2 Juli 1996. Penulis telah menempuh Pendidikan SD, SMP dan SMA di Jakarta. Sejak kecil, penulis memiliki ketertarikan yang besar pada bidang komputer dan matematika sehingga penulis memutuskan untuk mengambil pendidikan sarjana S1 di Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis pernah mengemban amanah sebagai asisten dosen pada mata kuliah Penjaminan Mutu Perangkat Lunak. Selain itu, penulis juga aktif berorganisasi menjadi staf Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS 2015/2016, staf BEM FTIf 2015/2016, dan staf schematics 2015 & schematics 2016

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Rekayasa Perangkat Lunak serta memiliki ketertarikan di bidang Manajemen Basis Data, Pemrograman *Web* dan *Mobile*. Untuk komunikasi, penulis dapat dihubungi melalui surel: [vivaaldy@gmail.com](mailto:vivaaldy@gmail.com).