**(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]

**KATA PENGANTAR**



Segala puji dan syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“*Rancang Bangun Sistem Informasi Akademik Generik Pada Modul Pembelajaran Dan IPD Menggunakan Pola Perancangan Repository-Service*”**.

Dalam pelaksanaan tugas akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah S.W.T dan Nabi Muhammad S.A.W.
2. Orang tua dan adik penulis yang senantiasa memberikan semangat dan doa agar penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
3. Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku pembimbing I sekaligus dosen wali yang telah membantu, membimbing, dan memberikan ilmunya kepada penulis dalam menyelesaikan Tugas Akhir ini.
4. Ibu Dr. Ir. Siti Rochimah, M.T. selaku pembimbing II yang telah membentu, membimbing, dan memberikan ilmunya kepada penulis dalam menyelesaikan Tugas Akhir ini.
5. Teman-teman Tugas Akhir RMK RPL: Farhan, Aldo, Faishal, Rara, Nia, Elva, Winda, Raras, Sabila, Nurul, Angga, Fito, Arya, Ikhsan, Steven, William yang telah menemani dan memberikan semangat kepada penulis selama pengerjaan Tugas Akhir.
6. Arin Indreswari yang telah memberikan waktunya, selalu memberikan dukungan dan semangat kepada penulis.
7. Resha, Aldo, Rara, Deni, Tosca, Pina, Faishal, Shafly, Rifat, Aufar, Dwika, Bimo, Ikhsan, Elva yang telah menemani dan memberikan semangat kepada penulis selama pengerjaan Tugas Akhir.
8. Nafia Rizky Yogayana yang selalu menemani, menyemangati, dan membuat ramai hidup penulis dan teman-teman Tugas Akhir RPL.
9. Teman-teman admin lab RPL angkatan 2014, 2015, 2016 yang menemani dan memberikan semangat kepada penulis selama pengerjaan Tugas Akhir.
10. Dira Ayu Meigasari yang selalu memberikan semangat dan meminta agar namanya dimasukkan di dalam kata pengantar.
11. Pihak lain yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak, khususnya bagi diri penulis sendiri dan seluruh *civitas academica* Teknik Informatika ITS, serta bagi agama, bangsa, dan negara. Tak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan tugas akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Juli 2018

Ivaldy Putra Lifiari

# DAFTAR ISI

[Abstrak vii](#_Toc520464548)

[Abstract ix](#_Toc520464549)

[KATA PENGANTAR xi](#_Toc520464550)

[DAFTAR ISI xiii](#_Toc520464551)

[DAFTAR GAMBAR xvii](#_Toc520464552)

[DAFTAR TABEL xxi](#_Toc520464553)

[1 BAB I PENDAHULUAN 1](#_Toc520464554)

[1.1. Latar Belakang 1](#_Toc520464555)

[1.2. Rumusan Masalah 3](#_Toc520464556)

[1.3. Batasan Masalah 3](#_Toc520464557)

[1.4. Tujuan 3](#_Toc520464558)

[1.5. Metodologi 4](#_Toc520464559)

[1.6. Sistematika Penulisan 5](#_Toc520464560)

[2 BAB II TINJAUAN PUSTAKA 7](#_Toc520464561)

[2.1. Pembelajaran 7](#_Toc520464564)

[2.2. IPD 7](#_Toc520464565)

[2.3. Sistem Informasi Akademik Generik 8](#_Toc520464566)

[2.4. Phalcon PHP 8](#_Toc520464567)

[2.5. Microsoft SQL Server 9](#_Toc520464568)

[2.6. Repository-Service Pattern 10](#_Toc520464569)

[2.7. MVC (Model-View-Controller) 11](#_Toc520464570)

[3 BAB III ANALISIS DAN PERANCANGAN 13](#_Toc520464571)

[3.1. Analisis 13](#_Toc520464573)

[3.1.1. Spesifikasi Kebutuhan Perangkat Lunak 14](#_Toc520464574)

[3.1.2. Kebutuhan Fungsional 14](#_Toc520464575)

[3.1.3. Aktor 16](#_Toc520464576)

[3.1.4. Kasus Penggunaan Modul Pembelajaran 18](#_Toc520464577)

[3.1.5. Kasus Penggunaan Modul IPD 53](#_Toc520464578)

[3.2. Perancangan 66](#_Toc520464579)

[3.2.1. Desain Arsitektur dan Pola Perancangan 67](#_Toc520464580)

[3.2.2. Perancangan Diagram Kelas 68](#_Toc520464581)

[3.2.3. Perancangan Basis Data 68](#_Toc520464582)

[3.2.4. Perancangan Antarmuka Modul pembelajaran 70](#_Toc520464583)

[3.2.5. Perancangan Antarmuka Modul IPD 95](#_Toc520464584)

[3.2.6. Proses Bisnis Sistem 102](#_Toc520464585)

[3.2.7. Relasi Antar Modul 105](#_Toc520464586)

[4 BAB IV IMPLEMENTASI 107](#_Toc520464587)

[4.1. Lingkungan Implementasi 107](#_Toc520464589)

[4.2. Implementasi Antarmuka Sistem 107](#_Toc520464590)

[4.2.1. Antarmuka Modul Pembelajaran 108](#_Toc520464591)

[4.2.2. Antarmuka Modul IPD 125](#_Toc520464592)

[*4.3.* Implementasi *Repository-Service Pattern* pada Arsitektur *Model-View-Controller* 131](#_Toc520464593)

[5 BAB V PENGUJIAN DAN EVALUASI 139](#_Toc520464594)

[5.1. Lingkungan Uji Coba 139](#_Toc520464600)

[5.2. Skenario Uji Coba 139](#_Toc520464601)

[5.2.1. Kasus Pengujian Mengelola Kelas Pembelajaran 140](#_Toc520464602)

[5.2.2. Kasus Pengujian Mengelola Pendamping Akademik 142](#_Toc520464603)

[5.2.3. Kasus Pengujian Mengelola Kehadiran Dosen 144](#_Toc520464604)

[5.2.4. Kasus Pengujian Mengelola Kehadiran Mahasiswa 146](#_Toc520464605)

[5.2.5. Kasus Pengujian Menyusun FRS 148](#_Toc520464606)

[5.2.6. Kasus Pengujian Mencetak FRS 151](#_Toc520464607)

[5.2.7. Kasus Pengujian Menyetujui FRS 152](#_Toc520464608)

[5.2.8. Kasus Pengujian Menampilkan Anak Wali 153](#_Toc520464609)

[5.2.9. Kasus Pengujian Menampilkan Kelas Departemen 154](#_Toc520464610)

[5.2.10. Kasus Pengujian Menampilkan Kelas UPMB 155](#_Toc520464611)

[5.2.11. Kasus Pengujian Mengelola Tahun Ajaran 155](#_Toc520464612)

[5.2.12. Kasus Pengujian Mengelola Beban Studi 157](#_Toc520464613)

[5.2.13. Kasus Pengujian Menampilkan Riwayat Pembayaran 159](#_Toc520464614)

[5.2.14. Kasus Pengujian Menampilkan Tagihan 160](#_Toc520464615)

[5.2.15. Kasus Pengujian Mengelola Tanggal Penting FRS 160](#_Toc520464616)

[5.2.16. Kasus Pengujian Mengelola Kuesioner IPD 162](#_Toc520464617)

[5.2.17. Kasus Pengujian Mengisi Kuesioner IPD 164](#_Toc520464618)

[5.2.18. Kasus Pengujian Menampilkan Daftar Kuesioner 165](#_Toc520464619)

[5.2.19. Kasus Pengujian Menampilkan Hasil IPD Dosen 166](#_Toc520464620)

[5.3. Rekapitulasi Kasus Pengujian 166](#_Toc520464621)

[6 BAB VI KESIMPULAN DAN SARAN 171](#_Toc520464622)

[6.1. Kesimpulan 171](#_Toc520464624)

[6.2. Saran 172](#_Toc520464625)

[7 DAFTAR PUSTAKA 173](#_Toc520464626)

[8 LAMPIRAN 175](#_Toc520464627)

[9 BIODATA PENULIS 179](#_Toc520464628)

# DAFTAR GAMBAR

[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)

# DAFTAR TABEL

[Tabel 3.1 Tabel Fungsional Modul Pembelajaran 14](#_Toc520464719)

[Tabel 3.2 Tabel Fungsional Modul IPD 16](#_Toc520464720)

[Tabel 3.3 Tabel Aktor Modul Pembelajaran 17](#_Toc520464721)

[Tabel 3.4 Tabel Aktor Modul IPD 18](#_Toc520464722)

[Tabel 3.5 Rincian Kasus Penggunaan Membuat Kelas 20](#_Toc520464723)

[Tabel 3.6 Rincian Kasus Penggunaan Mengubah Kelas 22](#_Toc520464724)

[Tabel 3.7 Rincian Kasus Penggunaan Menghapus Kelas 24](#_Toc520464725)

[Tabel 3.8 Rincian Kasus Penggunaan Mengelola Pendamping Akademik 26](#_Toc520464726)

[Tabel 3.9 Rincian Kasus Penggunaan Menampilkan Riwayat Perwalian 27](#_Toc520464727)

[Tabel 3.10 Rincian Kasus Penggunaan Mengelola Kehadiran Dosen 29](#_Toc520464728)

[Tabel 3.11 Rincian Kasus Penggunaan Mengelola Kehadiran Mahasiswa 31](#_Toc520464729)

[Tabel 3.12 Rincian Kasus Penggunaan Menyusun FRS 32](#_Toc520464730)

[Tabel 3.13 Rincian Kasus Penggunaan Mencetak FRS 35](#_Toc520464731)

[Tabel 3.14 Rincian Kasus Penggunaan Menyetujui FRS 37](#_Toc520464732)

[Tabel 3.15 Rincian Kasus Penggunaan Mengubah FRS 39](#_Toc520464733)

[Tabel 3.16 Rincian Kasus Penggunaan Menampilkan Anak Wali 40](#_Toc520464734)

[Tabel 3.17 Rincian Kasus Penggunaan Menampilkan Kelas Departemen 42](#_Toc520464735)

[Tabel 3.18 Rincian Kasus Penggunaan Menampilkan Kelas UPMB 43](#_Toc520464736)

[Tabel 3.19 Rincian Kasus Penggunaan Mengelola Tahun Ajaran 45](#_Toc520464737)

[Tabel 3.20 Rincian Kasus Penggunaan Mengelola Beban Studi 47](#_Toc520464738)

[Tabel 3.21 Rincian Kasus Penggunaan Menampilkan Riwayat Pembayaran 49](#_Toc520464739)

[Tabel 3.22 Rincian Kasus Penggunaan Menampilkan Tagihan 50](#_Toc520464740)

[Tabel 3.23 Rincian Kasus Penggunaan Mengelola Tanggal FRS 52](#_Toc520464741)

[Tabel 3.24 Rincian Kasus Penggunaan Mengelola Kuesioner Mata Kuliah 54](#_Toc520464742)

[Tabel 3.25 Rincian Kasus Penggunaan Mengelola Kuesioner Dosen 56](#_Toc520464743)

[Tabel 3.26 Rincian Kasus Penggunaan Mengisi Kuesioner Mata Kuliah 58](#_Toc520464744)

[Tabel 3.27 Rincian Kasus Penggunaan Mengelola Kuesioner Dosen 61](#_Toc520464745)

[Tabel 3.28 Rincian Kasus Penggunaan Menampilkan Kuesioner IPD 63](#_Toc520464746)

[Tabel 3.29 Rincian Kasus Penggunaan Menampilkan Hasil IPD Dosen 65](#_Toc520464747)

[Tabel 3.30 Penjelasan Antarmuka Menampilkan kelas 71](#_Toc520464748)

[Tabel 3.31 Penjelasan Antarmuka *Form* Kelas 73](#_Toc520464749)

[Tabel 3.32 Penjelasan Antarmuka Mengelola Pendamping Akademik 75](#_Toc520464750)

[Tabel 3.33 Penjelasan Antarmuka Mengelola Kehadiran Dosen 77](#_Toc520464751)

[Tabel 3.34 Penjelasan Antarmuka *Form* Kehadiran Dosen 78](#_Toc520464752)

[Tabel 3.35 Penjelasan Antarmuka Mengelola Kehadiran Mahasiswa 80](#_Toc520464753)

[Tabel 3.36 Penjelasan Antarmuka Menyusun FRS 81](#_Toc520464754)

[Tabel 3.37 Penjelasan Antarmuka Menyutujui FRS 84](#_Toc520464755)

[Tabel 3.38 Penjelasan Antarmuka Menampilkan Anak Wali 85](#_Toc520464756)

[Tabel 3.39 Penjelasan Antarmuka Menampilkan Kelas Departemen 86](#_Toc520464757)

[Tabel 3.40 Penjelasan Antarmuka Menampilkan Kelas UPMB 87](#_Toc520464758)

[Tabel 3.41 Penjelasan Antarmuka Mengelola Tahun Ajaran 88](#_Toc520464759)

[Tabel 3.42 Penjelasan Antarmuka *Form* Tahun Ajaran 89](#_Toc520464760)

[Tabel 3.43 Penjelasan Antarmuka Mengelola Beban Studi 91](#_Toc520464761)

[Tabel 3.44 Penjelasan Antarmuka *Form* Beban Studi 92](#_Toc520464762)

[Tabel 3.45 Penjelasan Antarmuka Mengelola Tanggal Penting FRS 93](#_Toc520464763)

[Tabel 3.46 Penjelasan Antarmuka *Form* Tanggal Penting FRS 94](#_Toc520464764)

[Tabel 3.47 Penjelasan Antarmuka Mengelola Kuesioner 96](#_Toc520464765)

[Tabel 3.48 Penjelasan Antarmuka *Form* Kelola Kuesioner 97](#_Toc520464766)

[Tabel 3.49 Penjelasan Antarmuka Mengisi Kuesioner 99](#_Toc520464767)

[Tabel 3.50 Penjelasan Antarmuka Melihat Daftar Kuesioner 100](#_Toc520464768)

[Tabel 3.51 Penjelasan Antarmuka Melihat Hasil IPD Dosen 102](#_Toc520464769)

[Tabel 4.1 Lingkungan Implementasi Sistem 107](#_Toc520464770)

[Tabel 5.1 Lingkungan Pengujian 139](#_Toc520464771)

[Tabel 5.2 Rincian Pengujian Fungsi Mengelola Kelas 140](#_Toc520464772)

[Tabel 5.3 Rincian Pengujian Fungsi Mengelola Pendamping Akademik 142](#_Toc520464773)

[Tabel 5.4 Rincian Pengujian Fungsi Mengelola Kehadiran Dosen 144](#_Toc520464774)

[Tabel 5.5 Rincian Pengujian Fungsi Mengelola Kehadiran Mahasiswa 146](#_Toc520464775)

[Tabel 5.6 Rincian Pengujian Fungsi Menyusun FRS 148](#_Toc520464776)

[Tabel 5.7 Rincian Pengujian Fungsi Mencetak FRS 151](#_Toc520464777)

[Tabel 5.8 Rincian Pengujian Fungsi Menyetujui FRS 152](#_Toc520464778)

[Tabel 5.9 Rincian Pengujian Fungsi Menampilkan Anak Wali 153](#_Toc520464779)

[Tabel 5.10 Rincian Pengujian Fungsi Menampilkan Kelas Deparetmen 154](#_Toc520464780)

[Tabel 5.11 Rincian Pengujian Fungsi Menampilkan Fungsi Kelas UPMB 155](#_Toc520464781)

[Tabel 5.12 Rincian Pengujian Fungsi Mengelola Tahun Ajaran 155](#_Toc520464782)

[Tabel 5.13 Rincian Pengujian Fungsi Mengelola Beban Studi 157](#_Toc520464783)

[Tabel 5.14 Rincian Pengujian Fungsi Menampilkan Riwayat Pembayaran 159](#_Toc520464784)

[Tabel 5.15 Rincian Pengujian Fungsi Menampilkan Tagihan 160](#_Toc520464785)

[Tabel 5.16 Rincian Pengujian Fungsi Mengelola Tanggal FRS 161](#_Toc520464786)

[Tabel 5.17 Rincian Pengujian Fungsi Mengelola Kuesioner IPD 162](#_Toc520464787)

[Tabel 5.18 Rincian Pengujian Fungsi Mengisi Kuesioner IPD 164](#_Toc520464788)

[Tabel 5.19 Rincian Pengujian Fungsi Menampilkan Daftar Kuesioner 165](#_Toc520464789)

[Tabel 5.20 Rincian Pengujian Fungsi Menampilkan Hasil IPD Dosen 166](#_Toc520464790)

[Tabel 5.21 Rekapitulasi Hasil Pengujian Modul Pembelajaran 167](#_Toc520464791)

*[Halaman ini sengaja dikosongkan]*

# CHAPTER I PENDAHULUAN

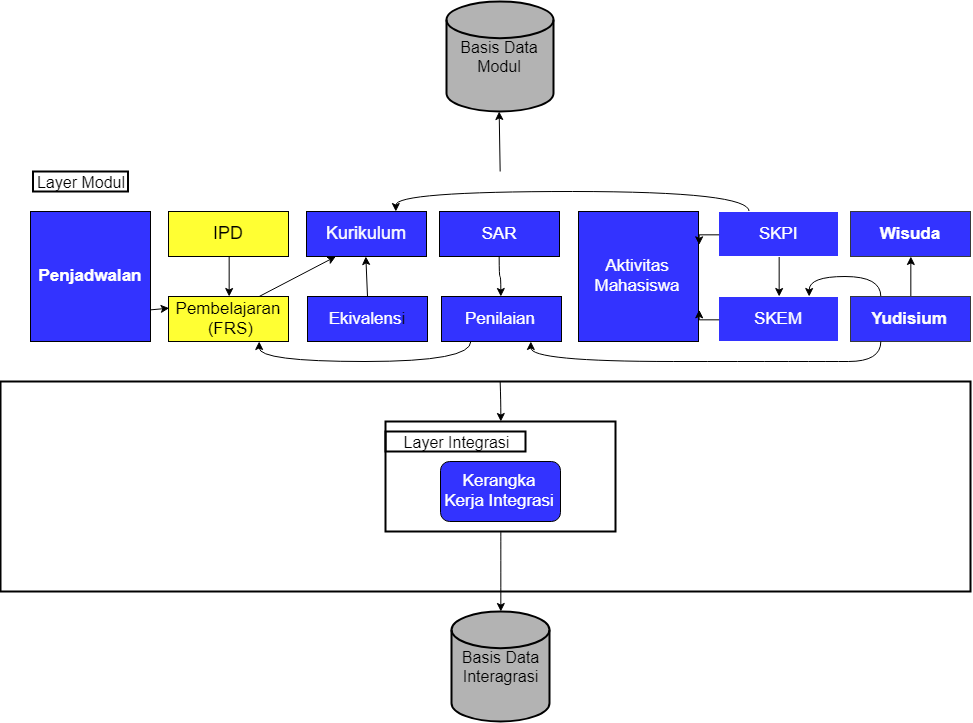
Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

## Latar Belakang

Jumlah perguruan tinggi baik perguruan tinggi negeri (PTN) maupun perguruan tinggi swasta (PTS) di Indonesia semakin bertambah setiap tahun. Hal ini menunjukan bahwa sistem informasi akademik (SIA) banyak dibutuhkan. Setiap perguruan tinggi memiliki kebijakan tersendiri terkait kebutuhan informasi akademiknya. Kebijakan ini pun tentunya selalu berubah setiap diadakan perubahan tahun ajaran.

Saat ini, banyak mahasiswa Institut Teknologi Sepuluh Nopember (ITS) yang berstatus sebagai mahasiswa asing dan berasal dari luar negeri. Hal ini menyebabkan bertambahnya kebutuhan pada SIA yaitu penambahan tampilan berbahasa Inggris atau dwibahasa. Dengan adanya 2 bahasa ini diharapkan dapat membantu para mahasiswa asing dalam mengakses SIA yang ada kedepannya.

Pada modul pembelajaran, proses bisnis Formulir Rencana Studi (FRS) atau Kartu Rencana Studi (KRS) memiliki alur yang berbeda sesuai dengan kebijakan pada perguruan tinggi. Ada perguruan tinggi yang membebaskan mahasiswa melakukan pemilihan kelas pembelajaran dan ada juga perguruan tinggi yang menggunakan sistem paket sehingga mahasiswa tidak perlu memilih kelas. Pada modul Indeks Pengajaran Dosen (IPD), terdapat 2 macam penilaian yaitu Indeks Pengajaran Dosen, yaitu penilaian berdasarkan pengajaran dosen dan Indeks Pengajaran Mata Kuliah, yaitu penilaian tentang mata kuliah yang diambil. Tujuan utama dari IPD ini adalah mendapatkan informasi tentang bagaimana hasil performa dosen pengajar dan tentang matakuliah yang diambil mahasiswa. Gambar 1.1 menunjukkan diagram relasi antar modul dan lokasi modul IPD dan pembelajaran pada SIA.



Gambar 1.1 Ilustrasi Rancang Bangun Perangkat Lunak

Sekarang ini Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) ITS sedang melakukan perombakan besar pada SIAKAD. Oleh karena itu, akan dibuat SIA pada modul pembelajaran dan IPD untuk menangani masalah tersebut. Modul pembelajaran memiliki berbagai kebutuhan seperti pengambilan matakuliah, pembatalan matakuliah, pembuatan kelas pembelajaran, dan penetapan tanggal penting pada FRS. Sementara modul IPD memiliki berbagai kebutuhan seperti mengelola kuisioner mata kuliah dan dosen, menampilkan rekap IPD, dan menampilkan ranking IPD. Kelompok pengguna pada dua modul pembelajaran adalah Petugas Biro Akademik, Kepala Satuan Manajemen, Mahasiswa, Dosen, dan Petugas Tata Usaha Akademik Departemen. Sementara kelompok pengguna pada modul IPD yaitu Petugas Biro Akademik, Mahasiswa, dan Dosen.

## Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut.

1. Bagaimana penggalian kebutuhan terhadap aktivitas dan fungsi umum SIA, secara khusus pada modul pembelajaran dan modul IPD yang dapat menangani variasi proses bisnis pada perguruan tinggi?
2. Bagaimana perancangan SIA untuk modul pembelajaran dan modul IPD yang bersifat generik?
3. Bagaimana modul pembelajaran dan modul IPD dapat berkomunikasi dengan modul lain pada SIA?
4. Bagaimana melakukan validasi pekerjaan agar produk yang dibuat dapat memenuhi kebutuhan yang diinginkan?

## Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yaitu sebagai berikut.

1. Rancang bangun modul pembelajaran dan IPD ini menggunakan bahasa pemrograman PHP dan sistem manajemen basis data SQL Server.
2. Modul ini dibuat menggunakan kerangkakerja integrasi.
3. Modul ini berbasis web.

## Tujuan

Tugas Akhir ini mempunyai beberapa tujuan, yaitu sebagai berikut.

1. Melakukan penggalian kebutuhan untuk membuat SKPL yang dibutuhkan dalam pembuatan modul pembelajaran dan IPD.
2. Membuat modul pembelajaran dan IPD yang dapat menangani variasi proses bisnis yang ada.
3. Membuat modul yang dapat terintegrasi dengan modul lain yang ada pada SIA.
4. Melakukan validasi agar produk yang dibuat dapat memenuhi kebutuhan yang diinginkan.

## Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

* + - 1. **Penyusunan proposal tugas akhir**

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

* + - 1. **Studi literatur**

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai kerangka kerja dan Repository-Service Pattern.

* + - 1. **Analisis dan desain perangkat lunak**

Pada tahap ini dilakukan analisis dan perancangan desain sistem informasi akademik yang akan dibuat. Analisis dilakukan dengan menentukan kebutuhan fungsional sistem dan perancangan desain sistem dilakukan dengan membuat desain basis data, mock up tampilan halaman web serta diagram-diagram yang dibutuhkan.

* + - 1. **Implementasi**

Adapun batasan-batasan yang dimiliki oleh perangkat lunak ini, yaitu sebagai berikut.

* + 1. Modul ini berbasis web.
    2. Modul ini dibuat menggunakan *framework* PhalconPHP.
    3. Modul ini menggunakan bahasa pemrograman PHP.
    4. Modul ini menggunakan basis data SQL Server.
       1. **Uji Coba dan Evaluasi**

Pada tahapan ini dilakukan uji coba pada perangkat lunak dengan pengujian *Control Test* dan peserta pengujian merupakan pendidik, peserta didik, tenaga kependidikan dan kepala. Setelah pengujian akan diberikan lembar *feedback* dan *checklist* berupa kebutuhan fungsionalitas.

* + - 1. **Penyusunan Buku Tugas Akhir**

Pada tahapan ini disusun buku yang membuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## Sistematika Penulisan

Buku tugas akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

1. Bab II. Tinjauan Pustaka

Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir.

1. Bab III. Analisis dan Perancangan

Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.

1. Bab IV. Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

1. Bab V. Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

1. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

# BAB II TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai

pendukung dalam pengembangan perangkat lunak.



## 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

Tabel 3.1 Functional Table of the Software

| No | Functional Requirements | Description |
| --- | --- | --- |
|
| 1 | Membuat kelas pembelajaran baru | Membuat kelas pembelajaran untuk kegiatan perkuliahan |
| 2 | Mengubah kelas pembelajaran | Mengubah data kelas pembelajaran yang sudah ada |
| 3 | Menghapus kelas pembelajaran | Menghapus data kelas pembelajaran yang sudah ada |
| 4 | Mengelola pendamping akademik | Mengelola penempatan anak wali dengan dosen wali |
| 5 | Menampilkan Riwayat Perubahan Perwalian | Menampilkan daftar perubahan dosen wali dari suatu mahasiswa |
| 6 | Mengelola kehadiran dosen | Mengelola data kehadiran dosen terhadap suatu kelas pembelajaran |
| 7 | Mengelola kehadiran mahasiswa | Mengelola data kehadiran mahasiswa terhadap suatu kelas pembelajaran |
| 8 | Menyusun FRS | Menyusun FRS dan memilih mata kuliah yang diinginkan |
| 9 | Mencetak FRS | Mengubah halaman FRS menjadi versi cetak |
| 10 | Menyetujui FRS | Melakukan persetujuan terhadap FRS anak wali |
| 11 | Mengubah FRS | Mengubah status dan isi FRS |
| 12 | Menampilkan anak wali | Menampilkan daftar anak wali suatu dosen |
| 13 | Menampilkan kelas pembelajaran departemen | Menampilkan daftar kelas pembelajaran suatu departemen |
| 14 | Menampilkan kelas UPMB | Menampilkan daftar kelas pembelajaran dari UPMB |
| 15 | Mengelola tahun ajaran | Mengelola data tahun ajar dan membuat suatu tahun ajar menjadi tahun ajaran aktif |
| 16 | Mengelola beban studi | Mengelola batas pengambilan SKS berdasarkan nilai IPS semester lalu |
| 17 | Menampilkan Riwayat Pembayaran SPP | Menampilkan data riwayat pembayaran SPP dari suatu mahasiswa |
| 18 | Menampilkan Riwayat Tagihan | Menampilkan data tagihan suatu mahasiswa dalam satu periode |
| 19 | Mengelola tanggal penting FRS | Mengelola tanggal penting FRS seperti tanggal mulai, batasa ubah, dan batas *drop* FRS |

### Actors

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Dalam sistem untuk tugas akhir ini memiliki empat aktor pada modul kurikulum dan lima aktor pada modul ekivalensi. Penjelasan lebih rinci akan dibahas pada Tabel 3.3 dan Tabel 3.4

Tabel 3.3 Tabel Aktor Modul Pembelajaran

| No | Aktor | Deskripsi |
| --- | --- | --- |
| 1 | Admin | * Membuat kelas pembelajaran baru. * Mengubah kelas pembelajaran. * Menghapus kelas pembelajaran. * Mengelola pendamping akademik (dosen wali). * Menampilkan riwayat perubahan dosen wali |
| 2 | Customer | * Mengelola kehadiran dosen * Mengelola kehadiran mahasiswa |

Tabel 3.4 Tabel Aktor Modul IPD

| No | Actor | Description |
| --- | --- | --- |
| 1 | Mahasiswa | * Mengisi kuesioner mata kuliah * Mengisi kuesioner dosen |
| 2 | Dosen | * Menampilkan kuesioner mata kuliah dan dosen * Menampilkan hasil IPD dosen |
| 3 | Direktorat Akademik | * Mengelola kuesioner mata kuliah. * Mengelola kuesioner dosen. |

### Kasus Penggunaan Modul Pembelajaran

Bagian ini menjelaskan secara rinci kasus penggunaan modul pembelajaran yang terdapat pada perangkat lunak seperti yang ditunjukkan pada Gambar 3.2. Selain itu, terdapat juga spesifikasi kasus penggunaan dan diagram aktivitas pada tiap kasus penggunaan.

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 create the travel destinations that shown on the apps and 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

Tabel 3.5 Rincian Kasus Penggunaan Membuat Kelas

| Components | Description |
| --- | --- |
| Name | Create travel destination |
| Code | UC-001 |
| Description | Kasus penggunaan ini digunakan untuk membuat kelas pembelajaran baru. |
| Tipe | Fungsional |
| Aktor | Kepala Prodi |
| Kondisi Awal | Kelas pembelajaran baru belum ditambahkan |
| Kondisi Akhir | Kelas pembelajaran baru berhasil dibuat. |
| Alur Normal | 1. Aktor memilih menu kelas 2. Sistem menampilkan halaman kelas pembelajaran. 3. Aktor memilih untuk menambah kelas. 4. Sistem menampilkan form untuk menambah kelas pembelajaran. 5. Aktor mengisi form untuk menambahkan kelas baru 6. Sistem menyimpan data kelas pembelajaran baru |
| Alur Alternatif | - |



Gambar 3.3 Diagram Aktivitas Membuat Kelas

#### Kasus Penggunaan Mengubah Kelas Pembelajaran

Pada kasus penggunaan ini, aktor dapat mengubah data kelas pembelajaran yang sudah ada. Rincian kasus penggunaan dapat dilihat pada Tabel 3.6 dan diagram aktivitas pada Gambar 3.4.

Tabel 3.6 Rincian Kasus Penggunaan Mengubah Kelas

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengubah kelas pembelajaran |
| Nomor | UC-002 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengubah data kelas pembelajaran yang sudah ada. |
| Tipe | Fungsional |
| Aktor | Kepala Prodi |
| Kondisi Awal | Kelas pembelajaran belum diubah |
| Kondisi Akhir | Kelas pembelajaran berhasil diubah |
| Alur Normal | 1. Aktor memilih menu kelas 2. Sistem menampilkan halaman kelas pembelajaran 3. Aktor memilih kelas pembelajaran yang ingin diubah 4. Sistem menampilkan form untuk mengubah data kelas 5. Aktor mengisi form untuk mengubah kelas 6. Sistem menyimpan data kelas pembelajaran yang sudah diubah |
| Alur Alternatif | - |



Gambar 3.4 Diagram Aktivitas Mengubah Kelas

#### Kasus Penggunaan Menghapus Kelas Pembelajaran

Pada kasus penggunaan ini, aktor dapat menghapus data kelas pembelajaran yang sudah ada. Rincian kasus penggunaan dapat dilihat pada Tabel 3.7 dan diagram aktivitas pada Gambar 3.5.

Tabel 3.7 Rincian Kasus Penggunaan Menghapus Kelas

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menghapus kelas pembelajaran |
| Nomor | UC-003 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menghapus data kelas pembelajaran yang sudah ada. |
| Tipe | Fungsional |
| Aktor | Kepala Prodi |
| Kondisi Awal | Kelas pembelajaran belum dihapus |
| Kondisi Akhir | Kelas pembelajaran berhasil dihapus |
| Alur Normal | 1. Aktor memilih menu kelas 2. Sistem menampilkan halaman kelas pembelajaran 3. Aktor memilih kelas pembelajaran yang ingin dihapus 4. Sistem menghapus data kelas pembelajaran yang dipilih |
| Alur Alternatif | - |



Gambar 3.5 Diagram Aktivitas Menghapus Kelas

#### Kasus Penggunaan Mengelola Pendamping Akademik

Pada kasus penggunaan ini, aktor dapat mengatur penempatan pendamping akademik (dosen wali) dengan anak wali nya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.8 dan diagram aktivitas pada Gambar 3.6.

Tabel 3.8 Rincian Kasus Penggunaan Mengelola Pendamping Akademik

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Pendamping Akademik |
| Nomor | UC-004 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengatur penempatan dosen wali dengan anak wali nya. |
| Tipe | Fungsional |
| Aktor | Kepala Prodi |
| Kondisi Awal | Ada mahasiswa yang belum memiliki dosen wali |
| Kondisi Akhir | Mahasiswa telah memiliki seorang dosen wali |
| Alur Normal | 1. Aktor memilih halaman kelola pendamping akademik 2. Sistem menampilkan halaman pendamping akademik 3. Aktor memilih dosen wali yang ingin dikelola 4. Sistem menampilkan daftar anak wali dari dosen yang dipilih 5. Aktor memilih anak wali yang ingin ditempatkan kepada dosen tersebut 6. Sistem menyimpan penempatan dosen wali dengan anak wali nya |
| Alur Alternatif | * 1. Sistem menampilkan bahwa anak wali tersebut telah memiliki dosen wali  1. Kembali ke langkah no. 4 |



Gambar 3.6 Diagram Aktivitas Pendamping Akademik

#### Kasus Penggunaan Menampilkan Riwayat Perwalian

Pada kasus penggunaan ini, aktor dapat mengelola kehadiran dosen terhadap suatu kelas pembelajaran yang diajarkannya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.9 dan diagram aktivitas pada Gambar 3.7

Tabel 3.9 Rincian Kasus Penggunaan Menampilkan Riwayat Perwalian

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Riwayat Perwalian |
| Nomor | UC-005 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan daftar riwayat perubahan dosen wali suatu mahasiswa |
| Tipe | Fungsional |
| Aktor | Kepala Prodi |
| Kondisi Awal | Data dosen wali sudah ada |
| Kondisi Akhir | Data riwayat perwalian berhasil ditampilkan |
| Alur Normal | 1. Aktor memilih halaman kelola pendamping akademik 2. Sistem menampilkan data pendamping akademik 3. Aktor memilih tombol untuk melihat riwayat perubahan perwalian 4. Sistem menampilkan daftar riwayat perubahan perwalian mahasiswa |
| Alur Alternatif | - |



Gambar 3.7 Diagram Aktivitas Menampilkan Riwayat Perwalian

#### Kasus Penggunaan Mengelola Kehadiran Dosen

Pada kasus penggunaan ini, aktor dapat mengelola kehadiran dosen terhadap suatu kelas pembelajaran yang diajarkannya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.10 dan diagram aktivitas pada Gambar 3.8.

Tabel 3.10 Rincian Kasus Penggunaan Mengelola Kehadiran Dosen

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Kehadiran Dosen |
| Nomor | UC-006 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola kehadiran dosen terhadap suatu kelas pembelajaran yang diajarkannya |
| Tipe | Fungsional |
| Aktor | Petugas Tata Usaha Akademik Departemen, Dosen |
| Kondisi Awal | Data kehadiran dosen sudah ada |
| Kondisi Akhir | Data kehadiran dosen telah diperbarui |
| Alur Normal | 1. Aktor memilih halaman untuk melihat kehadiran dosen 2. Sistem menampilkan data kehadiran dosen pada suatu mata kuliah 3. Aktor memilih untuk mengubah data kehadiran dosen 4. Sistem menyimpan perubahan kehadiran dosen |
| Alur Alternatif | * 1. Aktor memilih untuk menambah kehadiran dosen  1. Sistem menyimpan data kehadiran dosen baru    1. Aktor memilih untuk menghapus kehadiran dosen 2. Sistem menghapus kehadiran dosen |



Gambar 3.8 Diagram Aktivitas Kehadiran Dosen

#### Kasus Penggunaan Mengelola Kehadiran Mahasiswa

Pada kasus penggunaan ini, aktor dapat mengelola kehadiran mahasiswa terhadap suatu kelas pembelajaran. Rincian kasus penggunaan dapat dilihat pada Tabel 3.11 dan diagram aktivitas pada Gambar 3.9.

Tabel 3.11 Rincian Kasus Penggunaan Mengelola Kehadiran Mahasiswa

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Kehadiran Mahasiswa |
| Nomor | UC-007 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola kehadiran mahasiswa terhadap suatu kelas pembelajaran |
| Tipe | Fungsional |
| Aktor | Petugas Tata Usaha Akademik Departemen, Dosen |
| Kondisi Awal | Data kehadiran mahasiswa belum ada |
| Kondisi Akhir | Data kehadiran mahasiswa telah terisi |
| Alur Normal | 1. Aktor memilih halaman untuk melihat kehadiran mahasiswa 2. Sistem menampilkan daftar mahasiswa 3. Aktor mengisi tipe kehadiran setiap mahasiswa 4. Sistem menyimpan kehadiran mahasiswa |
| Alur Alternatif | - |



Gambar 3.9 Diagram Aktivitas Kehadiran Mahasiswa

#### Kasus Penggunaan Menyusun FRS

Pada kasus penggunaan ini, aktor dapat menyusun FRS dan mengambil mata kuliah yang diinginkan. Rincian kasus penggunaan dapat dilihat pada Tabel 3.12 dan diagram aktivitas pada Gambar 3.10.

Tabel 3.12 Rincian Kasus Penggunaan Menyusun FRS

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menyusun FRS |
| Nomor | UC-008 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menyusun FRS dan memilih mata kuliah yang diinginkan |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | FRS belum disusun |
| Kondisi Akhir | FRS telah disusun |
| Alur Normal | 1. Aktor membuka halaman untuk menyusun FRS 2. Sistem menampilkan halaman FRS 3. Aktor memilih mata kuliah yang diinginkan 4. Sistem memeriksa apakah SKS aktor cukup untuk mengambil mata kuliah yang dipilih 5. Sistem memeriksa apakah masih ada tempat kosong untuk aktor didalam kelas yang dipilih 6. Sistem mengambil kelas yang dipilih aktor |
| Alur Alternatif | * 1. Sistem menampilkan notifikasi bahwa SKS aktor tidak cukup untuk mengambil kelas yang dipilih  1. Kembali ke langkah no.2    1. Sistem menampilkan notifikasi bahwa kelas yang dipilih aktor telah penuh 2. Kembali ke langkah no.2 |



Gambar 3.10 Diagram Aktivitas Menyusun FRS

#### Kasus Penggunaan Mencetak FRS

Pada kasus penggunaan ini, aktor dapat mengubah halaman FRS menjadi versi cetak dan mencetaknya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.13 dan diagram aktivitas pada Gambar 3.11.

Tabel 3.13 Rincian Kasus Penggunaan Mencetak FRS

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mencetak FRS |
| Nomor | UC-009 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mencetak halaman FRS |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | Halaman FRS masih belum ada perubahan |
| Kondisi Akhir | Halaman FRS menjadi halaman versi cetak |
| Alur Normal | 1. Aktor membuka halaman untuk FRS 2. Sistem menampilkan halaman FRS 3. Aktor memilih untuk mencetak halaman FRS 4. Sistem menampilkan halaman FRS versi cetak |
| Alur Alternatif | - |



Gambar 3.11 Diagram Aktivitas Mencetak FRS

#### Kasus Penggunaan Menyetujui FRS

Pada kasus penggunaan ini, aktor dapat menyetujui status FRS yang telah selesai disusun. Rincian kasus penggunaan dapat dilihat pada Tabel 3.14 dan diagram aktivitas pada Gambar 3.12.

Tabel 3.14 Rincian Kasus Penggunaan Menyetujui FRS

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menyetujui FRS |
| Nomor | UC-010 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menyetujui FRS yang telah disusun |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | FRS belum disetujui |
| Kondisi Akhir | FRS telah disetujui |
| Alur Normal | 1. Aktor membuka halaman anak wali 2. Sistem menampilkan halaman anak wali 3. Aktor memilih FRS salah satu anak wali 4. Sistem menampilkan halaman FRS anak wali yang dipilih 5. Aktor memilih untuk menyetujui FRS tersebut 6. Sistem menampilkan notifikasi bahwa FRS telah disetujui |
| Alur Alternatif | - |



Gambar 3.12 Diagram Aktivitas Menyetujui FRS

#### Kasus Penggunaan Mengubah FRS

Pada kasus penggunaan ini, aktor dapat mengubah status FRS yang telah disetujui dan mengubah isi dari FRS tersebut. Rincian kasus penggunaan dapat dilihat pada Tabel 3.15 dan diagram aktivitas pada Gambar 3.13.

Tabel 3.15 Rincian Kasus Penggunaan Mengubah FRS

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengubah FRS |
| Nomor | UC-011 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengubah status dan isi FRS |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | FRS belum diubah |
| Kondisi Akhir | FRS telah diubah |
| Alur Normal | 1. Aktor membuka halaman anak wali 2. Sistem menampilkan halaman anak wali 3. Aktor memilih FRS salah satu anak wali 4. Sistem menampilkan halaman FRS anak wali yang dipilih 5. Aktor memilih untuk membatalkan persetujuan FRS tersebut 6. Sistem menampilkan notifikasi bahwa FRS telah berhasil dibatalkan |
| Alur Alternatif | * 1. Aktor memilih kelas untuk anak wali nya  1. Sistem menyimpan kelas pilihan aktor |



Gambar 3.13 Diagram Aktivitas Mengubah FRS

#### Kasus Penggunaan Menampilkan Anak Wali

Pada kasus penggunaan ini, aktor dapat menampilkan seluruh daftar anak wali nya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.16 dan diagram aktivitas pada Gambar 3.14.

Tabel 3.16 Rincian Kasus Penggunaan Menampilkan Anak Wali

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Anak Wali |
| Nomor | UC-012 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan data seluruh anak wali dari suatu dosen |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | Data anak wali sudah ada di basis data |
| Kondisi Akhir | Halaman berhasil menampilkan daftar anak wali |
| Alur Normal | 1. Aktor membuka halaman anak wali 2. Sistem menampilkan data seluruh anak wali nya |
| Alur Alternatif | * 1. Sistem tidak menemukan data anak wali dari basis data  1. Sistem menampilkan tabel dengan data kosong |



Gambar 3.14 Diagram Aktivitas Menampilkan Anak Wali

#### Kasus Penggunaan Menampilkan kelas Pembelajaran Departemen

Pada kasus penggunaan ini, aktor dapat menampilkan seluruh daftar kelas pembelajaran yang ada pada departemen tempatnya mengajar. Rincian kasus penggunaan dapat dilihat pada Tabel 3.17 dan diagram aktivitas pada Gambar 3.15.

Tabel 3.17 Rincian Kasus Penggunaan Menampilkan Kelas Departemen

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Kelas Pembelajaran Departemen |
| Nomor | UC-013 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan seluruh kelas yang ada pada suatu departemen |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | Data kelas pembelajaran sudah ada di basis data |
| Kondisi Akhir | Halaman berhasil menampilkan daftar kelas pembelajaran suatu departemen |
| Alur Normal | 1. Aktor memilih halaman kelas departemen 2. Sistem menampilkan daftar kelas departemen |
| Alur Alternatif | * 1. Sistem tidak menemukan data kelas didalam basis data  1. Sistem menampilkan tabel dengan data kosong |



Gambar 3.15 Diagram Aktivitas Menampilkan Kelas Departemen

#### Kasus Penggunaan Menampilkan Kelas UPMB

Pada kasus penggunaan ini, aktor dapat menampilkan seluruh daftar kelas pembelajaran UPMB yang ada. Rincian kasus penggunaan dapat dilihat pada Tabel 3.18 dan diagram aktivitas pada Gambar 3.16.

Tabel 3.18 Rincian Kasus Penggunaan Menampilkan Kelas UPMB

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Kelas Pembelajaran UPMB |
| Nomor | UC-014 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan seluruh kelas UPMB yang ada |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | Data kelas UPMB telah ada didalam basis data |
| Kondisi Akhir | Halaman berhasil menampilkan daftar kelas pembelajaran UPMB |
| Alur Normal | 1. Aktor memilih halaman kelas UPMB 2. Sistem menampilkan daftar kelas UPMB |
| Alur Alternatif | * 1. Sistem tidak menemukan data kelas didalam basis data  1. Sistem menampilkan tabel dengan data kosong |



Gambar 3.16 Diagram Aktivitas Menampilkan Kelas UPMB

#### Kasus Penggunaan Mengelola Tahun Ajaran

Pada kasus penggunaan ini, aktor dapat mengelola data tahun ajaran aktif. Rincian kasus penggunaan dapat dilihat pada Tabel 3.19 dan diagram aktivitas pada Gambar 3.17.

Tabel 3.19 Rincian Kasus Penggunaan Mengelola Tahun Ajaran

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Tahun Ajaran Aktif |
| Nomor | UC-015 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola data tahun ajaran |
| Tipe | Fungsional |
| Aktor | Petugas Biro Akademik |
| Kondisi Awal | Data tahun ajaran sudah ada pada basis data |
| Kondisi Akhir | Data tahun ajaran berhasil diperbarui |
| Alur Normal | 1. Aktor memilih halaman tahun ajaran 2. Sistem menampilkan halaman tahun ajaran 3. Aktor memilih untuk mengubah tahun ajaran baru 4. Sistem perubahan data tahun ajaran |
| Alur Alternatif | 3.1.Aktor memilih untuk membuat data tahun ajaran baru   1. Sistem menyimpan data tahun ajaran baru   3.2.Aktor memilih untuk menghapus data tahun ajaran   1. Sistem menghapus data tahun ajaran |



Gambar 3.17 Diagram Aktivitas Mengelola Tahun Ajaran

#### Kasus Penggunaan Mengelola Beban Studi

Pada kasus penggunaan ini, aktor dapat mengelola data beban studi yaitu batas pengambilan SKS yang didasarkan oleh IPS semester lalu. Rincian kasus penggunaan dapat dilihat pada Tabel 3.20 dan diagram aktivitas pada Gambar 3.18.

Tabel 3.20 Rincian Kasus Penggunaan Mengelola Beban Studi

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Beban Studi |
| Nomor | UC-016 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola data beban studi |
| Tipe | Fungsional |
| Aktor | Petugas Direktorat Akademik |
| Kondisi Awal | Data beban studi ada di dalam basis data |
| Kondisi Akhir | Data beban studi berhasil diperbarui |
| Alur Normal | 1. Aktor memilih halaman beban studi 2. Sistem menampilkan halaman beban studi 3. Aktor memilih untuk mengubah data beban studi 4. Sistem menyimpan data beban studi yang baru |
| Alur Alternatif | 3.1.Aktor memilih untuk membuat data beban studi baru   1. Sistem menyimpan beban studi baru   3.2.Aktor memilih untuk menghapus data beban studi   1. Sistem menghapus data beban studi |



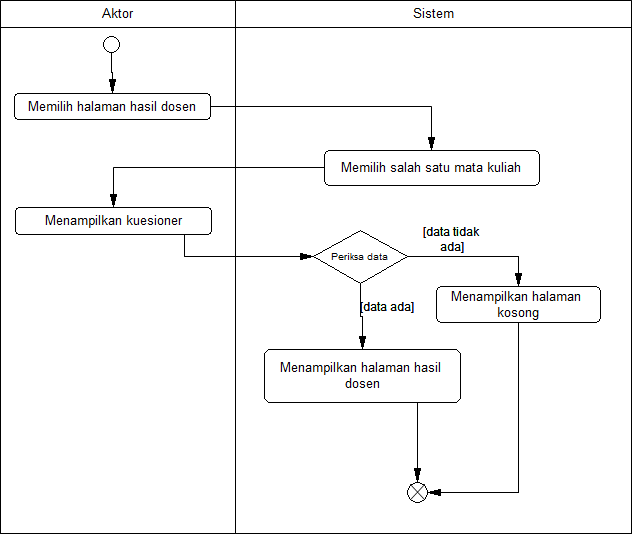
Gambar 3.18 Diagram Aktivitas Mengelola Beban Studi

#### Kasus Penggunaan Menampilkan Riwayat Pembayaran

Pada kasus penggunaan ini, aktor dapat data riwayat pembayaran UKT selama masa perkuliahannya. Rincian kasus penggunaan dapat dilihat pada Tabel 3.21 dan diagram aktivitas pada Gambar 3.19.

Tabel 3.21 Rincian Kasus Penggunaan Menampilkan Riwayat Pembayaran

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Riwayat Pembayaran |
| Nomor | UC-017 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan data riwayat pembayaran UKT |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | Data riwayat pembayaran ada di dalam basis data |
| Kondisi Akhir | Data riwayat pembayaran berhasil ditampilkan |
| Alur Normal | 1. Aktor memilih halaman riwayat pembayaran 2. Sistem menampilkan halaman riwayat pembayaran |
| Alur Alternatif | - |



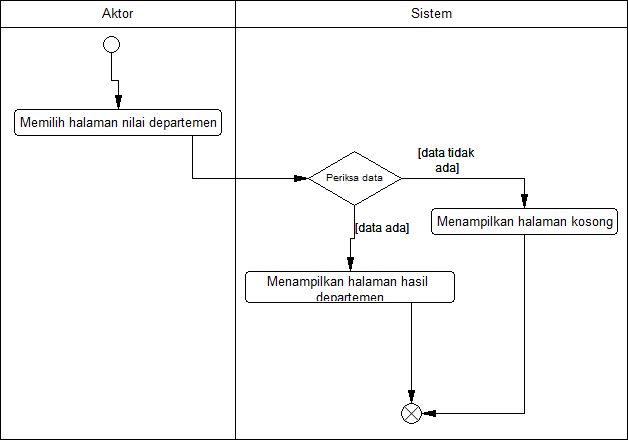
Gambar 3.19 Diagram Aktivitas Menampilkan Riwayat Pembayaran

#### Kasus Penggunaan Menampilkan Tagihan

Pada kasus penggunaan ini, aktor dapat menampilkan data tagihan keuangannya selama satu periode. Rincian kasus penggunaan dapat dilihat pada Tabel 3.22 dan diagram aktivitas pada Gambar 3.20.

Tabel 3.22 Rincian Kasus Penggunaan Menampilkan Tagihan

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Riwayat Pembayaran |
| Nomor | UC-018 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan data tagihan |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | Data tagihan ada di dalam basis data |
| Kondisi Akhir | Data tagihan berhasil ditampilkan |
| Alur Normal | 1. Aktor memilih halaman tagihan 2. Sistem menampilkan halaman tagihan |
| Alur Alternatif | - |



Gambar 3.20 Diagram Aktivitas Menampilkan Tagihan

#### Kasus Penggunaan Mengelola Tanggal Penting FRS

Pada kasus penggunaan ini, aktor dapat mengelola tanggal penting FRS seperti untuk menyusun, mengubah, dan membatalkan FRS. Rincian kasus penggunaan dapat dilihat pada Tabel 3.23 dan diagram aktivitas pada Gambar 3.21.

Tabel 3.23 Rincian Kasus Penggunaan Mengelola Tanggal FRS

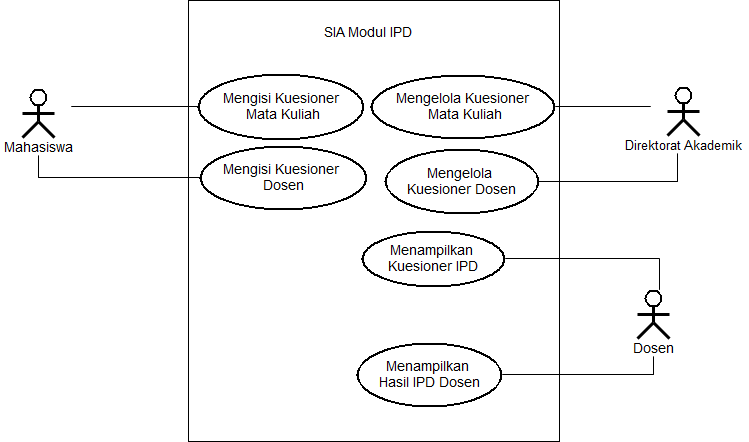
| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Tanggal Penting FRS |
| Nomor | UC-019 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola tanggal penting untuk melakukan FRS |
| Tipe | Fungsional |
| Aktor | Petugas Biro Akademik |
| Kondisi Awal | Data tanggal FRS ada di dalam basis data |
| Kondisi Akhir | Data tanggal FRS berhasil diperbarui |
| Alur Normal | 1. Aktor memilih halaman pengaturan 2. Sistem menampilkan halaman pengaturan 3. Aktor memilih pengaturan tanggal FRS 4. Sistem menampilkan halaman pengaturan tanggal FRS 5. Aktor mengisi tanggal untuk memulai, batas ubah, dan batas *drop* FRS 6. Sistem menyimpan data tanggal FRS |
| Alur Alternatif | - |



Gambar 3.21 Diagram Aktivitas Mengelola Tanggal Penting FRS

### Kasus Penggunaan Modul IPD

Bagian ini menjelaskan secara rinci kasus penggunaan modul IPD yang terdapat pada perangkat lunak. Selain itu, terdapat juga spesifikasi kasus penggunaan dan diagram aktivitas pada tiap kasus penggunaan.



Gambar 3.22 Diagram Kasus Penggunaan Modul IPD

#### Kasus Penggunaan Mengelola Kuesioner Mata Kuliah

Pada kasus penggunaan ini, aktor dapat mengelola pertanyaan dan jawaban kuesioner mata kuliah untuk IPD. Rincian kasus penggunaan dapat dilihat pada Tabel 3.24 dan Gambar 3.23.

Tabel 3.24 Rincian Kasus Penggunaan Mengelola Kuesioner Mata Kuliah

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Kuesioner Mata Kuliah |
| Nomor | UC-001 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola pertanyaan dan jawaban dari kuesioner mata kuliah IPD |
| Tipe | Fungsional |
| Aktor | Petugas Direktorat Akademik |
| Kondisi Awal | Data kuesioner sudah ada di dalam basis data |
| Kondisi Akhir | Data kuesioner berhasil diperbarui |
| Alur Normal | 1. Aktor memilih halaman kelola kuesioner mata kuliah 2. Sistem menampilkan halaman kuesioner mata kuliah 3. Aktor memilih untuk mengubah pertanyaan dan jawaban kuesioner baru 4. Sistem menyimpan perubahan data |
| Alur Alternatif | * 1. Aktor memilih untuk membuat data kuesioner baru  1. Sistem menyimpan data kuesioner baru    1. Aktor memilih untuk menghapus salah satu kuesioner yang sudah ada 2. Sistem menghapus data kuesioner yang dipilih |



Gambar 3.23 Diagram Aktivitas Mengelola Kuesioner Mata Kuliah

#### Kasus Penggunaan Mengelola Kuesioner Dosen

Pada kasus penggunaan ini, aktor dapat mengelola pertanyaan dan jawaban kuesioner dosen untuk IPD. Rincian kasus penggunaan dapat dilihat pada Tabel 3.25 dan Gambar 3.24.

Tabel 3.25 Rincian Kasus Penggunaan Mengelola Kuesioner Dosen

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengelola Kuesioner Dosen |
| Nomor | UC-002 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengelola pertanyaan dan jawaban dari kuesioner dosen IPD |
| Tipe | Fungsional |
| Aktor | Petugas Direktorat Akademik |
| Kondisi Awal | Data kuesioner sudah ada di dalam basis data |
| Kondisi Akhir | Data kuesioner berhasil diperbarui |
| Alur Normal | 1. Aktor memilih halaman kelola kuesioner mata kuliah 2. Sistem menampilkan halaman kuesioner mata kuliah 3. Aktor memilih untuk mengubah pertanyaan dan jawaban kuesioner 4. Sistem menyimpan perubahan data kuesioner |
| Alur Alternatif | 3.1.Aktor memilih untuk menambah kuesioner  1. Sistem menyimpan data kuesioner baru  3.2. Aktor memilih untuk menghapus salah satu kuesioner yang sudah ada  1. Sistem menyimpan perubahan data kuesioner |



Gambar 3.24 Diagram Aktivitas Mengelola Kuesioner Dosen

#### Kasus Penggunaan Mengisi Kuesioner Mata Kuliah

Pada kasus penggunaan ini, aktor dapat mengisi kuesioner mata kuliah IPD. Rincian kasus penggunaan dapat dilihat pada tabel 3.26 dan Gambar 3.25.

Tabel 3.26 Rincian Kasus Penggunaan Mengisi Kuesioner Mata Kuliah

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengisi Kuesioner Mata Kuliah |
| Nomor | UC-003 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengisi kuesioner mata kuliah IPD |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | Kuesioner mata kuliah belum diisi |
| Kondisi Akhir | Kuesioner mata kuliah telah diisi |
| Alur Normal | 1. Aktor memilih halaman kuesioner mata kuliah 2. Sistem menampilkan halaman kuesioner mata kuliah 3. Aktor memilih mata kuliah yang belum diisi 4. Sistem menampilkan kuesioner mata kuliah 5. Aktor mengisi kuesioner 6. Sistem menyimpan kuesioner yang telah diisi aktor |
| Alur Alternatif | - |



Gambar 3.25 Diagram Aktivitas Mengisi Kuesioner Mata Kuliah

#### Kasus Penggunaan Mengisi Kuesioner Dosen

Pada kasus penggunaan ini, aktor dapat mengisi kuesioner dosen IPD. Rincian kasus penggunaan dapat dilihat pada tabel 3.27 dan Gambar 3.26.

Tabel 3.27 Rincian Kasus Penggunaan Mengelola Kuesioner Dosen

| Komponen | Deskripsi |
| --- | --- |
| Nama | Mengisi Kuesioner Dosen |
| Nomor | UC-004 |
| Deskripsi | Kasus penggunaan ini digunakan untuk mengisi kuesioner dosen IPD |
| Tipe | Fungsional |
| Aktor | Mahasiswa |
| Kondisi Awal | Kuesioner dosen belum diisi |
| Kondisi Akhir | Kuesioner dosentelah diisi |
| Alur Normal | 1. Aktor memilih halaman kuesioner dosen 2. Sistem menampilkan halaman kuesioner dosen 3. Aktor memilih dosen yang belum diisi 4. Sistem menampilkan kuesioner dosen 5. Aktor mengisi kuesioner 6. Sistem menyimpan kuesioner yang telah diisi aktor |
| Alur Alternatif | - |



Gambar 3.26 Diagram Aktivitas Mengisi Kuesioner Dosen

#### Kasus Pengelolaan Menampilkan Kuesioner IPD

Pada kasus penggunaan ini, aktor dapat menampilkan semua pertanyaan dan jawaban kuesioner mata kuliah dan dosen. Rincian kasus penggunaan dapat dilihat pada tabel 3.28 dan Gambar 3.27.

Tabel 3.28 Rincian Kasus Penggunaan Menampilkan Kuesioner IPD

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Kuesioner IPD |
| Nomor | UC-005 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan daftar pertanyaan dan jawaban kuesioner mata kuliah dan dosen IPD |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | Data kuesioner IPD sudah ada di basis data |
| Kondisi Akhir | Halaman berhasil menampilkan data kuesioner IPD |
| Alur Normal | 1. Aktor memilih halaman daftar kuesioner 2. Sistem menampilkan daftar kuesioner mata kuliah dan dosen IPD |
| Alur Alternatif | * 1. Sistem tidak menemukan data kuesioner IPD didalam basis data  1. Sistem menampilkan tabel dengan data kosong |



Gambar 3.27 Diagram Aktivitas Menampilkan Kuesioner IPD

#### Kasus Penggunaan Menampilkan Hasil IPD Dosen

Pada kasus penggunaan ini, aktor dapat menampilkan hasil penilaian kuesioner IPD dirinya berdasarkan mata kuliah yang diajarkannya. Rincian kasus penggunaan dapat dilihat pada tabel 3.29 dan Gambar 3.28.

Nilai hasil IPD setiap dosen didapatkan dari menghitung total bobot kuesioner dibagi dengan total pertanyaan kuesioner yang telah diisi dari setiap mata kuliah yang diajarkan oleh dosen tersebut. Total bobot kuesioner didapatkan dari menjumlahkan nilai setiap pilihan pertanyaan kuesioner yang disii oleh mahasiswa.

Tabel 3.29 Rincian Kasus Penggunaan Menampilkan Hasil IPD Dosen

| Komponen | Deskripsi |
| --- | --- |
| Nama | Menampilkan Hasil IPD Dosen |
| Nomor | UC-006 |
| Deskripsi | Kasus penggunaan ini digunakan untuk menampilkan hasil penilaian kuesioner IPD |
| Tipe | Fungsional |
| Aktor | Dosen |
| Kondisi Awal | Data hasil IPD dosen sudah ada di basis data |
| Kondisi Akhir | Halaman berhasil menampilkan data hasil IPD dosen |
| Alur Normal | 1. Aktor memilih halaman hasil IPD dosen 2. Sistem menampilkan halaman hasil IPD dosen 3. Aktor memilih salah satu mata kuliah yang diajar 4. Sistem menampilkan penilaian dan komentar IPD pada mata kuliah tersebut |
| Alur Alternatif | - |



Gambar 3.28 Diagram Aktivitas Menampilkan Hasil IPD Dosen

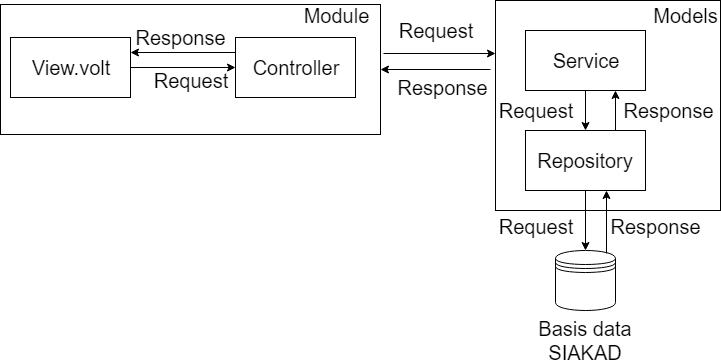
## Perancangan

Pada bagian perancangan akan dijelaskan mengenai arsitektur sistem yang digunakan, perancangan diagram kelas, perancangan basis data, dan perancangan antarmuka.

### Desain Arsitektur dan Pola Perancangan

Arsitektur sistem yang digunakan pada Tugas Akhir ini menggunakan arsitektur sistem framework Phalcon pada setiap modul. Antarmuka pengguna merupakan lapisan yang berhubungan langsung dengan pengguna. Controller adalah penghubung antara antarmuka dengan lapisan service dari aplikasi. Lapisan service menyediakan proses pengolahan data dari lapisan repository. Kemudian, controller mendapatkan data yang dikembalikan. Setelah itu, controller akan menampilkan di antarmuka pengguna.

Kelas repository dan entitas dalam basis data merupakan representasi dari model arsitektur MVC. Kelas service dan controller, sedangkan antarmuka pengguna merupakan representasi dari view. Ilustrasi pada Gambar 3.29 menunjukkan diagram arsitektur MVC dengan pola perancangan *Repository-Service*.



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

### Perancangan Diagram Kelas

Pada bab Lampiran, model arsitektur sistem modul pembelajaran dan IPD yang memiliki kelas berupa *controller*, *service*, dan *repository*. Penggunaan arsitektur sistem seperti gambar di atas digunakan agar lebih mudah dalam melakukan *maintenance* dan lebih mudah dalam implementasi menjadi SIA yang bersifat modular.

Kelas *controller* bergantung pada kelas *service* yang merupakan tempat pemrosesan data dan kelas *service* mengirimkan permintaan data ke basis data melewati kelas *repository*. Kelas *repository* akan mengirimkan permintaan ke basis data dan mengirimkan kembali data dari basis data ke kelas *service.*

### Perancangan Basis Data

Dalam membuat suatu aplikasi berupa sistem informasi, diperlukan sebuah analisis berupa perancangan basis data. Microsoft SQL Server dipilih menjadi aplikasi basis data karena dapat menampung data dalam skala besar. Microsoft SQL Server juga menduking UUID (*Universal Unique Identifiers*).

Rancangan basis data ditampilkan dalam bentuk Conceptual Data Model (CDM) dan Physical Data Model (PDM). Untuk penjelasan lebih rinci, PDM akan dijelaskan pada bab Lampiran.

#### Penjelasan Tabel

1. Tabel spesifik untuk modul pembelajaran
2. Tabel akt\_ajar

Tabel ini digunakan untuk menyimpan data urutan mengajar dosen pada suatu kelas pembelajaran.

1. Tabel beban\_studi

Tabel ini digunakan untuk menyimpan data beban studi yang menunjukan batas pengambilan SKS berdasarkan IPS semester lalu.

1. Tabel dosen\_wali

Tabel ini digunakan untuk menyimpan data relasi perwalian antara dosen dengan mahasiswa anak wali nya.

1. Tabel frs

Tabel ini digunakan untuk menyimpan data frs suatu mahasiswa.

1. Tabel hadir\_dosen

Tabel ini digunakan untuk menyimpan data kehadiran dosen terhadap suatu kelas.

1. Tabel hadir\_mhs

Tabel ini digunakan untuk menyimpan data kehadiran mahasiswa terhadap suatu kelas.

1. Tabel keaktifan\_mhs

Tabel ini digunakan untuk menyimpan data umum suatu mahasiswa dalam satu semester seperti sks\_tempuh, ips, dan ipk.

1. Tabel kuliah

Tabel ini digunakan untuk menyimpan data relasi mahasiswa dengan kelas yang dipilihnya ketika melakukan FRS.

1. Tabel rwy\_ubah\_frs

Tabel ini menyimpan riwayat perubahan FRS suatu mahasiswa, seperti menambah atau mengubah kelas pilihan nya.

1. Tabel tm\_kuliah

Tabel ini digunakan untuk menyimpan data tatap muka tiap minggu perkuliahan dan juga data berita acara apabila ada.

1. Tabel spesifik untuk modul IPD
2. Tabel jawaban\_kuesioner

Tabel ini menyimpan daftar jawaban kuesioner IPD baik kuesioner mata kuliah maupun kuesioner dosen.

1. Tabel pertanyaan\_kuesioner

Tabel ini menyimpan daftar pertanyaan kuesioner IPD baik kuesioner mata kuliah maupun kuesioner dosen.

1. Tabel kuesioner

Tabel ini menyimpan relasi antara mahasiswa dengan kuesioner yang diisi nya, seperti waktu pengisian, kelas yang diisi, dan komentar yang diisi.

1. Tabel respon\_kuesioner

Tabel ini menyimpan jawaban kuesioner yang dipilih oleh mahasiswa ketika mengisi kuesioner IPD.

1. Tabel umum yang dipakai modul pembelajaran dan IPD
2. Tabel mahasiswa

Tabel ini menyimpan data umum mahasiswa seperti nama, tempat & tanggal lahir, data wali, nomor telepon, dan email.

1. Tabel reg\_mhs

Tabel ini menyimpan data perkuliahan mahasiswa seperti NRP, semester & tanggal masuk kuliah, satker, dan ipk nya.

1. Tabel satker

Tabel ini menyimpan seluruh data satker seperti nama satker, satuan pendidikan, dan jenjang didik nya

1. Tabel sdm

Tabel ini menyimpan data Sumber Daya Manusia seperti nama, nik, satker, status keaktifan, dan jenis profesi nya.

1. Tabel semester

Tabel ini menyimpan data semester seperti nama semester, tanggal mulai, tanggal selesai, tahun ajar, dan semester aktif nya.

1. Tabel tahun\_ajaran

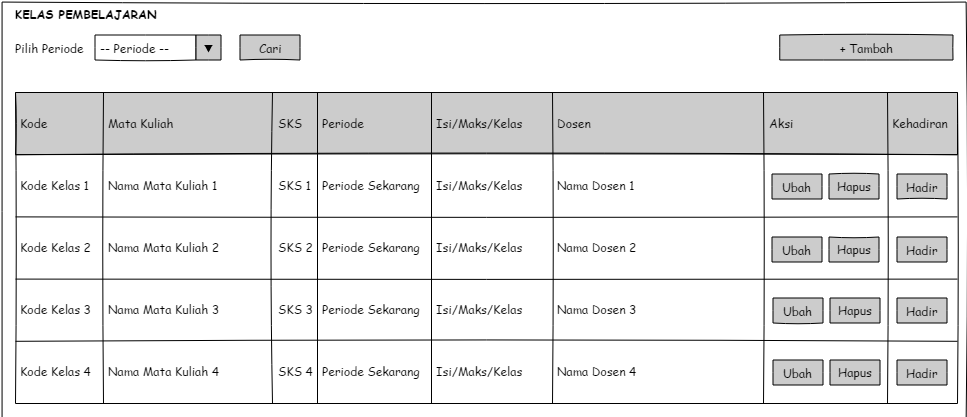
Tabel ini menyimpan data tahun ajaran seperti nama tahun ajar, tanggal mulai, tanggal selesai, dan tahun ajar aktif nya.

### Perancangan Antarmuka Modul pembelajaran

Pada bagian ini akan dibahas mengenai rancangan antarmuka modul pembelajaran bagi pengguna untuk memenuhi kasus penggunaan yang sudah dirancang.

#### Halaman Mengelola Kelas pembelajaran

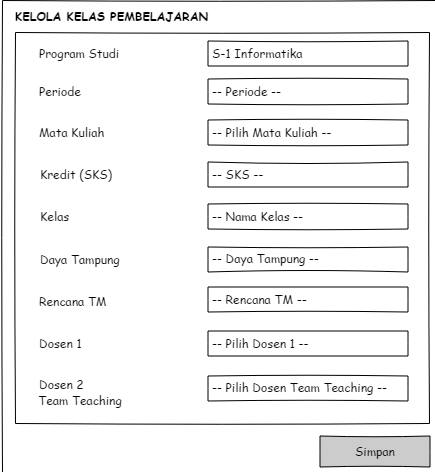
Halaman ini digunakan untuk mengelola data kelas pembelajaran. Pada halaman ini terdapat daftar kelas yang ada dan pengguna dapat menambah, mengubah, atau menghapus data kelas tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.30 dan form untuk halaman ini dapat dilihat pada Gambar 3.31.



Gambar 3.30 Rancangan Antarmuka Menampilkan Kelas

Tabel 3.30 Penjelasan Antarmuka Menampilkan kelas

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan /Keluaran |
| --- | --- | --- | --- | --- |
| 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* |



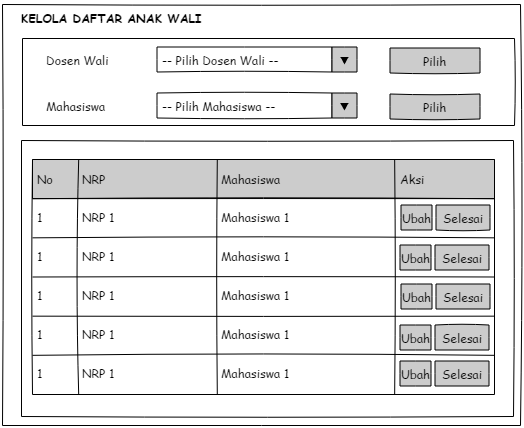
Gambar 3.31 Rancangan Antarmuka *Form* Kelas

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* |

#### Halaman Mengelola Pendamping Akademik

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.



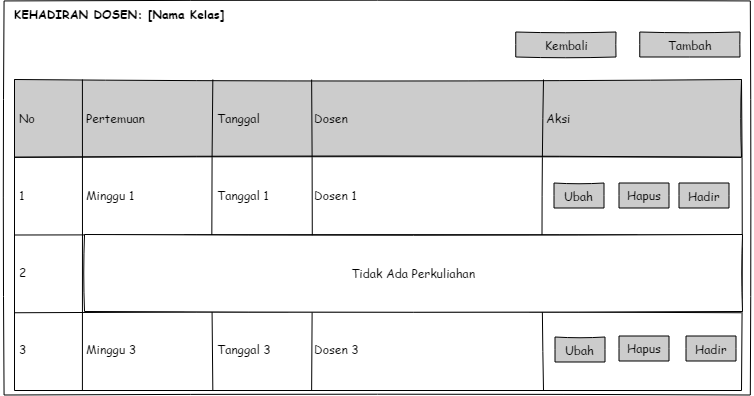
Gambar 3.32 Rancangan Antarmuka Mengelola Pendamping Akademik

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* |

#### Halaman Mengelola Kehadiran Dosen

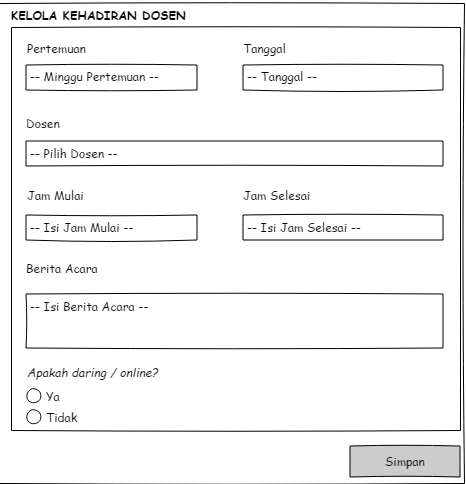
Halaman ini digunakan untuk mengelola data kehadiran dosen terhadap suatu kelas pembelajaran. Pada halaman ini terdapat daftar kehadiran dosen yang sudah dibuat. Pengguna juga dapat membuat, mengubah, atau menghapus data kehadiran dosen. Halaman antarmuka dapat dilihat pada Gambar 3.33 dan form untuk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



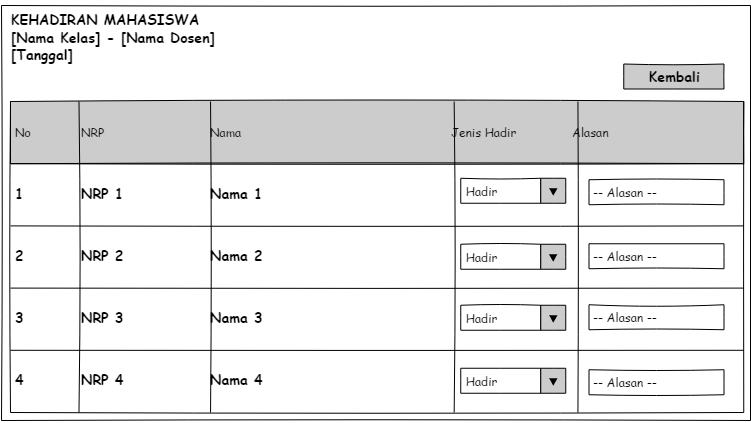
Gambar 3.34 Rancangan Antarmuka *Form* Kehadiran Dosen

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* |

#### Halaman Mengelola Kehadiran mahasiswa

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.



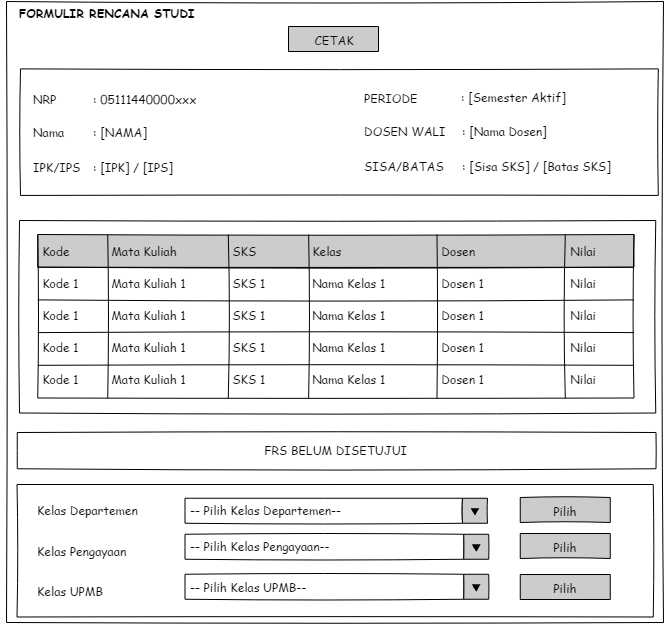
Gambar 3.35 Rancangan Antarmuka Mengelola Kehadiran Mahasiswa

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* |

#### Halaman Menyusun FRS

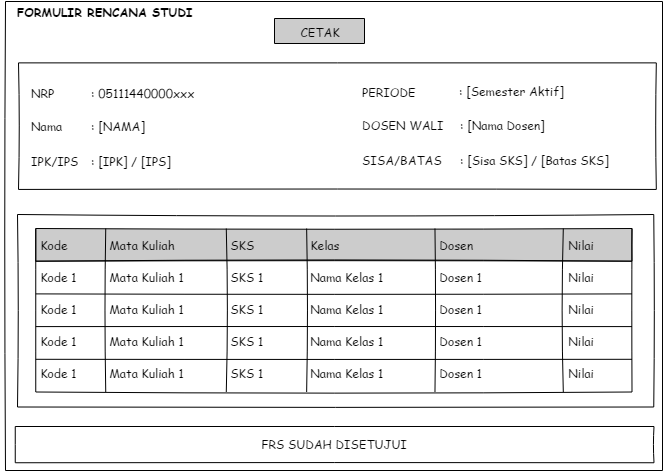
Halaman ini digunakan untuk menyusun FRS. Pada halaman ini terdapat daftar kelas yang dipilih ketika melakukan FRS. Pengguna juga dapat memilih kelas yang diinginkan apabila FRS tersebut belum disetujui oleh dosen wali yang bersangkutan. Halaman antarmuka dapat dilihat pada Gambar 3.36 dan Gambar 3.37.



Gambar 3.36 Rancangan Antarmuka FRS Belum Setuju

Tabel 3.36 Penjelasan Antarmuka Menyusun FRS

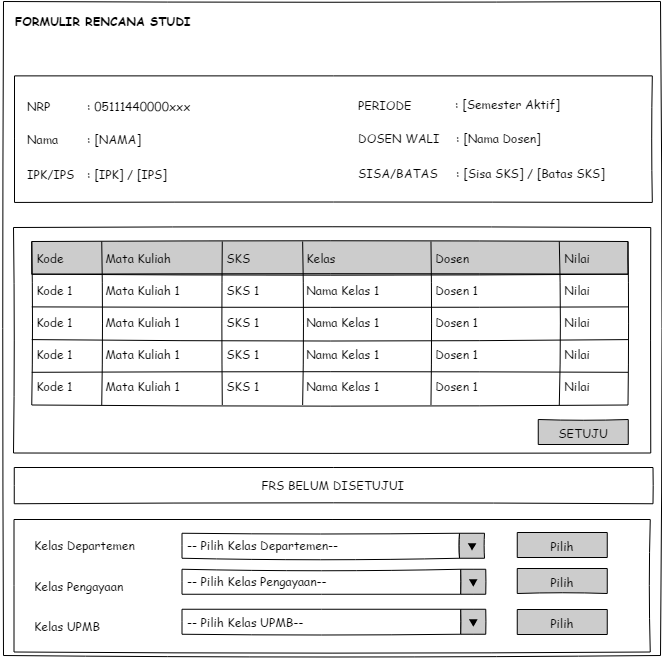
| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *cetakButton* | *Button* | Tombol untuk mengubah halaman menjadi versi cetak. | *Button* |
| 2 | *datatableMahasiswa* | *Table* | Menampilkan data mahasiswa pada halaman FRS | *Table* |
| 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

#### Halaman Menyetujui FRS

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.



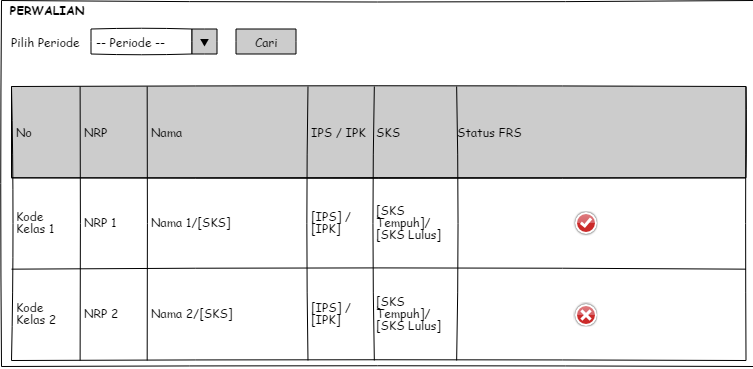
Gambar 3.38 Rancangan Antarmuka Menyetujui FRS

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* |

#### Halaman Menampilkan Anak Wali

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.



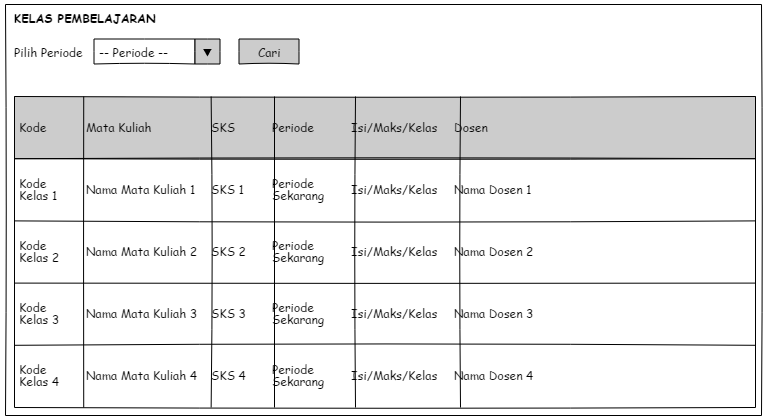
Gambar 3.39 Rancangan Antarmuka Menampilkan Anak Wali

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* |

#### Halaman Menampilkan Kelas Departemen dan UPMB

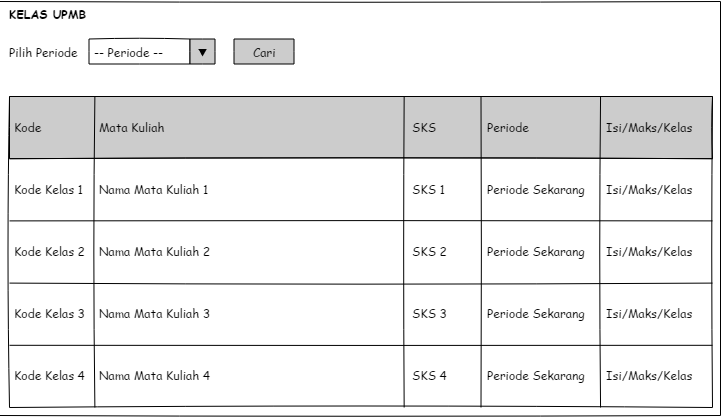
Halaman ini digunakan untuk menampilkan daftar kelas pembelajaran. Pengguna dapat memilih untuk menampilkan kelas pilihannya yaitu kelas departemen atau kelas UPMB. Halaman antarmuka dapat dilihat pada Gambar 3.40 dan Gambar 3.41.



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* |



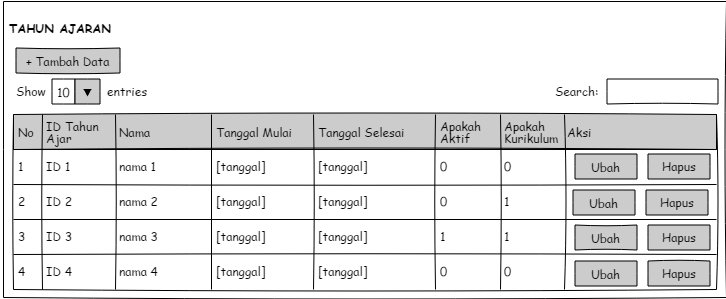
Gambar 3.41 Rancangan Antarmuka Menampilkan Kelas UPMB

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* |

#### Halaman Mengelola Tahun Ajaran

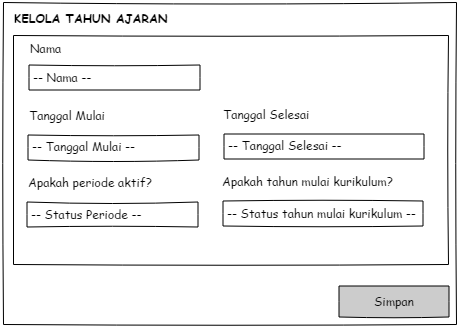
Halaman ini digunakan untuk mengelola data tahun ajaran. Pada halaman ini terdapat daftar tahun ajaran yang sudah tersimpan di basis data. Pengguna dapat memilih untuk menambah, mengubah, atau menghapus data tahun ajaran. Pengguna juga dapat mengganti status keaktifan tahun ajaran. Halaman antarmuka 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* |



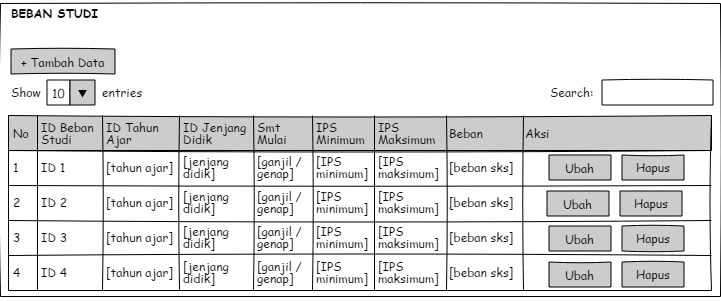
Gambar 3.43 Rancangan Antarmuka *Form* Tahun Ajaran

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* |

#### Halaman Mengelola Beban Studi

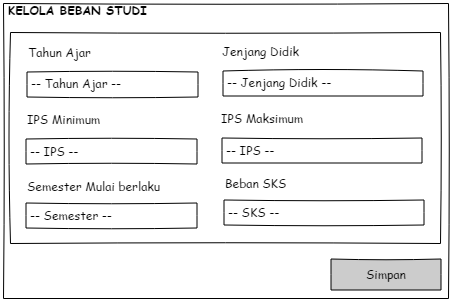
Halaman ini digunakan untuk mengelola data beban studi. Pada halaman ini terdapat daftar beban studi yang sudah tersimpan di basis data. Pengguna dapat memilih untuk menambah, mengubah, atau menghapus data beban studi. Halaman antarmuka dapat dilihat pada Gambar 3.44 dan 3.45.



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* |



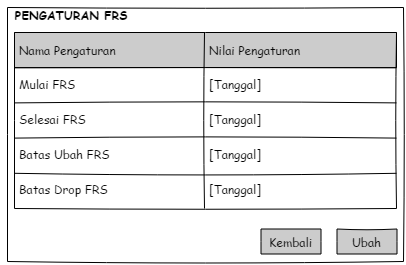
Gambar 3.45 Rancangan Antarmuka *Form* Beban Studi

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* |

#### Halaman Mengelola Tanggal Penting FRS

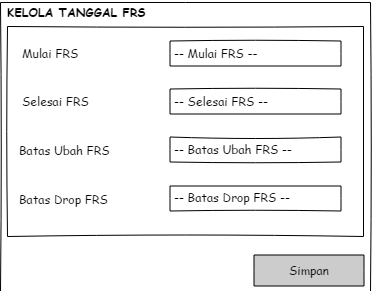
Halaman ini digunakan untuk mengelola data tanggal penting untuk FRS. Pada halaman ini terdapat data tanggal FRS yang sudah tersimpan di basis data. Pengguna dapat mengubah data tanggal FRS tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.46 dan Gambar 3.47.



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

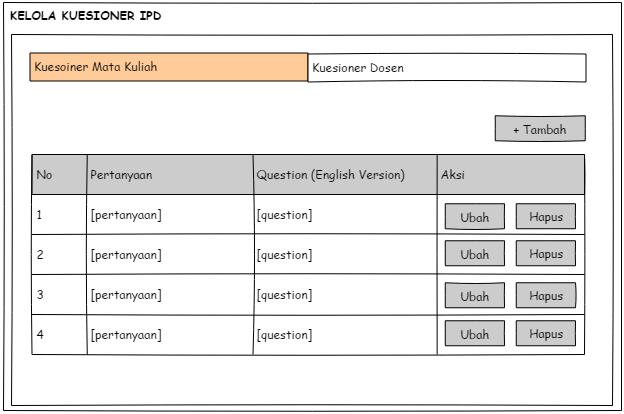
| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *mulaiFRS* | *Date* | Input untuk mengisi tanggal mulai FRS | *Date* |
| 2 | *selesaiFRS* | *Date* | Input untuk mengisi tanggal selesai FRS | *Date* |
| 3 | *batasUbahFRS* | *Date* | 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* |

### Perancangan Antarmuka Modul IPD

Pada bagian ini akan dibahas mengenai rancangan antarmuka modul IPD bagi pengguna untuk memenuhi kasus penggunaan yang sudah dirancang.

#### Halaman Mengelola Kuesioner Mata Kuliah dan Dosen

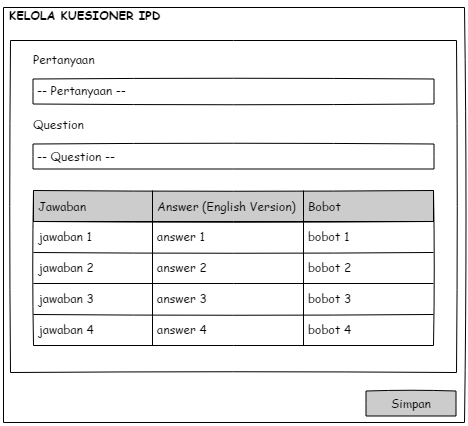
Halaman ini digunakan untuk mengelola pertanyaan dan jawaban kuesioner IPD. Pada halaman ini terdapat daftar kuesioner yang sudah tersimpan di basis data. Pengguna dapat memilih untuk menambah, mengubah, atau menghapus data kuesioner. Halaman antarmuka dapat dilihat pada Gambar 3.48 dan 3.49.



Gambar 3.48 Rancangan Antarmuka Mengelola Kuesioner

Tabel 3.47 Penjelasan Antarmuka Mengelola Kuesioner

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tabsJenisKuesioner* | *Tabs* | Memilih jenis kuesioner | *TabsClick* |
| 2 | *tambahButton* | *Button* | Tombol untuk membuat data kuesioner baru | *ButtonClick* |
| 3 | *ubahButton* | *Button* | Tombol untuk mengubah data kuesioner | *ButtonClick* |
| 4 | *hapusButton* | *Button* | Tombol untuk menghapus data kuesioer | *ButtonClick* |
| 5 | *datatableKuesioner* | *Table* | Menampilkan data kuesioner dan jawaban IPD | *Table* |



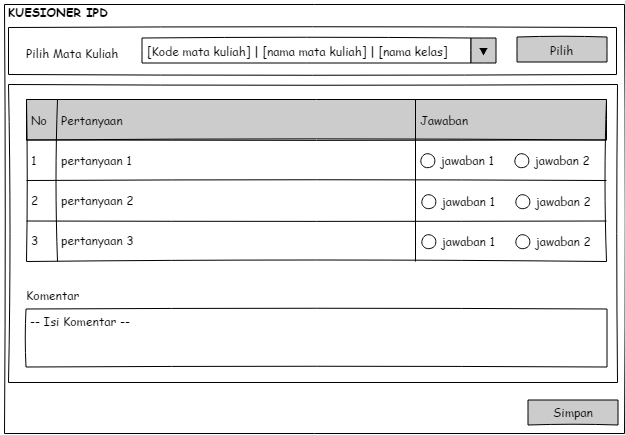
Gambar 3.49 Rancangan Antarmuka *Form* Kelola Kuesioner

Tabel 3.48 Penjelasan Antarmuka *Form* Kelola Kuesioner

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pertanyaan* | *Text* | Input untuk mengisi pertanyaan IPD | *String* |
| 2 | *question* | *Text* | Input untuk mengisi pertanyaan IPD versi inggris | *String* |
| 3 | *jawaban* | *Text* | Input untuk mengisi jawaban IPD | *String* |
| 4 | *answer* | *Text* | Input untuk mengisi jawaban IPD versi inggris | *String* |
| 5 | *simpanButton* | *Button* | Tombol untuk menyimpan kelola data kuesioner | *ButtonClick* |

#### Halaman Mengisi Kuesioner Mata Kuliah dan Dosen

Halaman ini digunakan untuk mengisi kuesioner mata kuliah dan dosen. Pada halaman ini terdapat pertanyaan kuesioner yang harus diisi. Pengguna juga dapat mengisi komentar sebagai tambahan. Halaman antarmuka dapat dilihat pada Gambar 3.50.



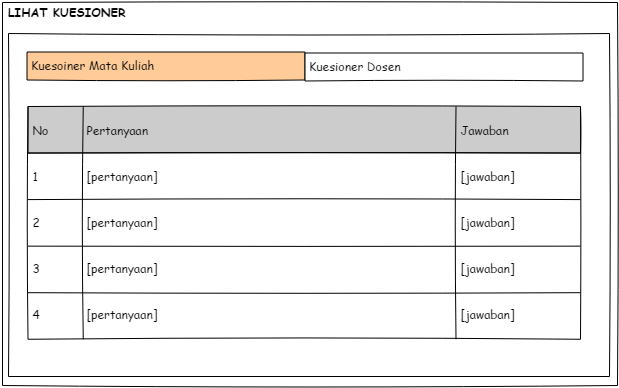
Gambar 3.50 Rancangan Antarmuka Mengisi Kuesioner

Tabel 3.49 Penjelasan Antarmuka Mengisi Kuesioner

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pilihKelas* | *Dropdown* | Input untuk memilih kelas | *String* |
| 2 | *datatableIPD* | *Table* | Menampilkan daftar pertanyaan IPD | *Table* |
| 3 | *jawabanRadio* | *Radio* | Menampilkan jawaban IPD | *Integer* |
| 4 | *komentarIPD* | *Text* | Input untuk mengisi komentar | *String* |
| 5 | *simpanButton* | *Button* | Tombol untuk menyimpan hasil mengisi kuesioner | *ButtonClick* |

#### Halaman Melihat Daftar Kuesioner

Halaman ini digunakan untuk menampilkan daftar kuesioner. Pada halaman ini, daftar pertanyaan dan jawaban yang sudah tersimpan di basis data akan ditampilkan. Halaman antarmuka dapat dilihat pada Gambar 3.51.



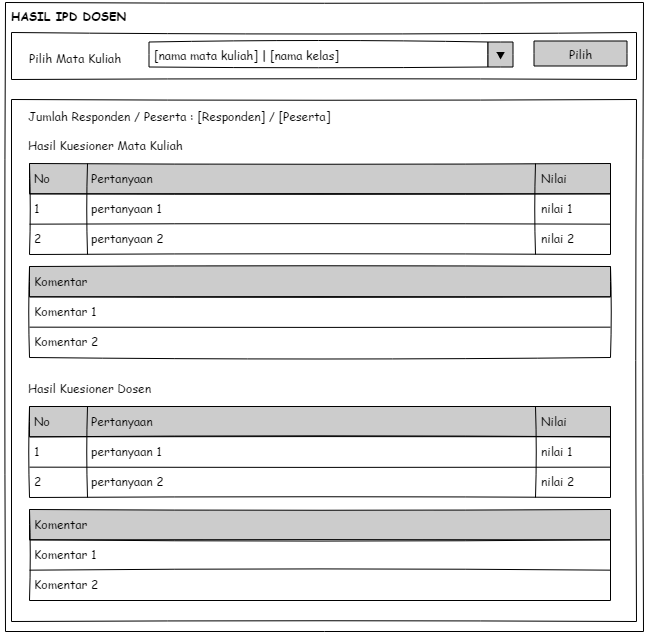
Gambar 3.51 Rancangan Antarmuka Melihat Daftar Kuesioner

Tabel 3.50 Penjelasan Antarmuka Melihat Daftar Kuesioner

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *tabsJenisKuesioner* | *Tabs* | Memilih jenis kuesioner IPD | *TabsClick* |
| 2 | *datatableIPD* | *Table* | Menampilkan daftar pertanyaan dan jawaban IPD | *Table* |

#### Halaman Melihat Hasil IPD Dosen

Halaman ini digunakan untuk menampilkan hasil IPD nya. Pengguna dapat memilih kelas yang diajarkannya dan halaman akan menampilkan penilaian IPD pada kelas tersebut. Halaman antarmuka dapat dilihat pada Gambar 3.52.



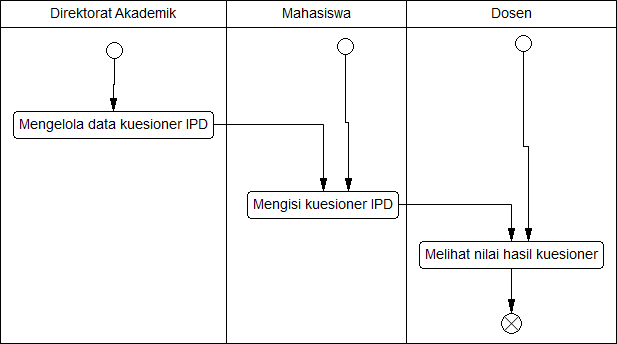
Gambar 3.52 Rancangan Antarmuka Melihat Hasil IPD Dosen

Tabel 3.51 Penjelasan Antarmuka Melihat Hasil IPD Dosen

| No | Nama Atribut Antarmuka | Jenis Atribut | Kegunaan | Jenis Masukan/  Keluaran |
| --- | --- | --- | --- | --- |
| 1 | *pilihKelas* | *Dropdown* | Input untuk memilih kelas | *String* |
| 2 | *datatableKuesionerMK* | *Table* | Menampilkan daftar nilai IPD MK | *Table* |
| 3 | *datatableKuesionerDosen* | *Table* | Menampilkan daftar nilai IPD Dosen | *Table* |
| 4 | *datatableKomentarMK* | *Table* | Menampilkan komentar IPD MK | *Table* |
| 5 | *datatableKomentarDosen* | *Table* | Menampilkan komentar IPD Dosen | *Table* |

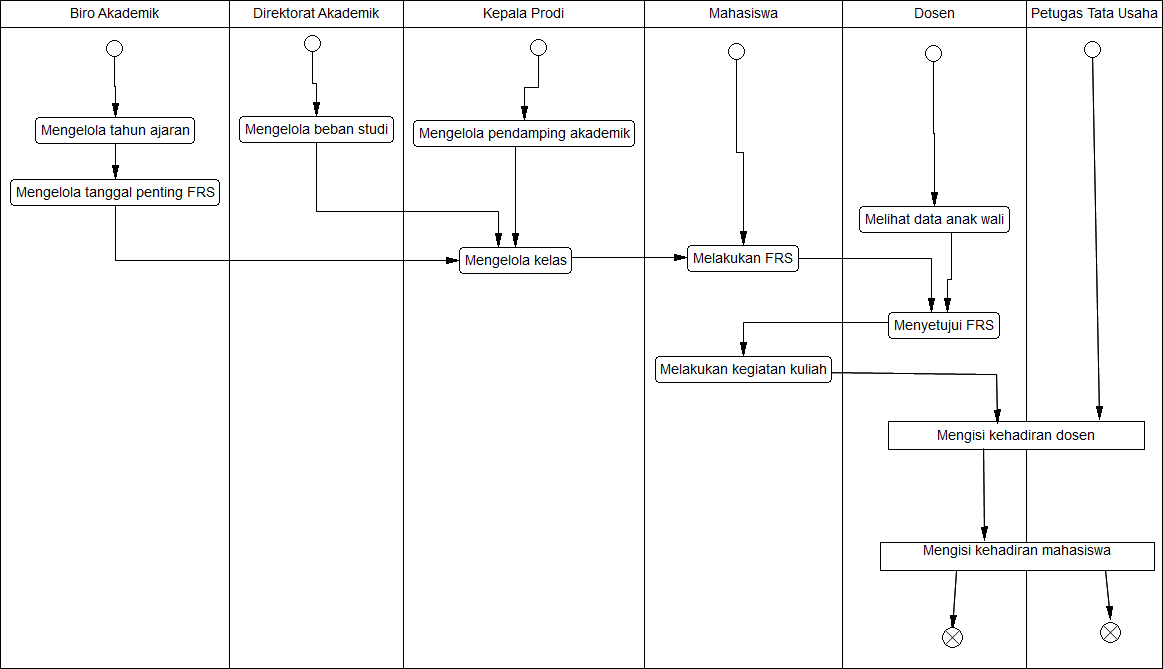
### Proses Bisnis Sistem

Pada subbab ini akan ditunjukkan proses bisnis untuk modul pembelajaran dan modul IPD. Proses bisnis modul IPD dimulai oleh aktor petugas direktorat akademik. Setiap awal tahun ajar baru, Petugas akan mulai membuat daftar kuesioner IPD baru atau mengubah data kuesioner yang sudah ada. Lalu pada akhir semester, mahasiswa dapat mengisi kuesioner IPD yang sudah ada. Lalu dosen dapat melihat nilai hasil kuesioner IPD yang sudah diisi mahasiswa tadi. Alur proses bisnis modul IPD ditunjukkan pada Gambar 3.53.



Gambar 3.53 Proses bisnis modul IPD

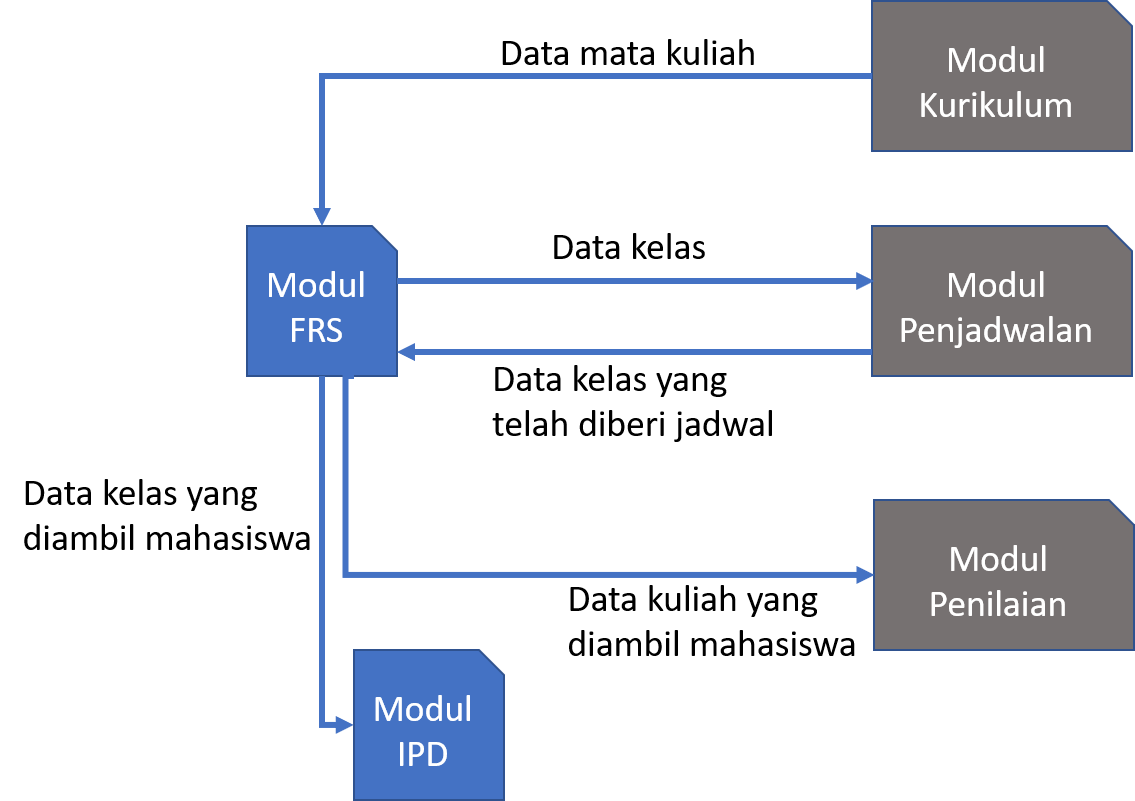
Proses bisnis modul pembelajaran dimulai dengan petugas biro akademik dan petugas direktorat akademik membuat atau mengubah data-data umum yang penting seperti mengganti tahun ajaran, mengubah data beban studi (SKS yang didapatkan berdasarkan IPS mahasiswa semester lalu), dan data tanggal-tanggal penting untuk melakukan FRS. Pada saat yang sama, kepala prodi dapat mengelola data pendamping akademik apabila terdapat perubahan untuk dosen wali atau anak wali. Selanjutnya, kepala prodi membuat data kelas baru data mata kuliah dan kurikulum yang telah dibuat oleh modul kurikulum. Lalu mahasiswa dapat menyusun FRS nya setelah kelas berhasil dibuat dan diberi jadwal oleh modul penjadwalan. Lalu dosen dapat menyetujui FRS anak wali nya apabila FRS telah selesai disusun atau mengubah FRS anak wali nya apabila terdapat jadwal yang bertabrakan ketika FRS telah disetujui. Selanjutnya ketika kegiatan perkuliahan berlangsung, dosen dan petugas tata usaha dapat mengisi kehadiran untuk dosen dan mahasiswa setiap minggu nya. Alur untuk proses bisnis modul Pembelajaran ditunjukkan pada Gambar 3.54.



Gambar 3.54 Proses bisnis modul FRS

### Relasi Antar Modul

Modul pembelajaran (FRS) memiliki relasi dengan modul yang ada pada SIA. Modul FRS mengambil data mata kuliah dari modul kurikulum. Selanjutnya dibuat data kelas dan diberikan kepada modul penjadwalan dan modul penjadwalan memberikan data kelas yang telah diberi jadwal kembali kepada modul FRS. Modul penilaian juga mengambil data kuliah yang diambil mahasiswa dari modul FRS. Modul IPD mengambil data kelas yang diambil mahasiswa ketika melakukan FRS agar kuesioner IPD dapat diisi.



Gambar 3.55 Relasi antar modul FRS dan IPD

*[Halaman ini sengaja dikosongkan]*

# BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem sesuai dengan analisis dan perancangan proses bisnis pada SIA modul pembelajaran dan modul IPD yang dijelaskan pada bab sebelumnya.



## Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Sistem

|  |  |
| --- | --- |
| **Perangkat** | **Spesifikasi** |
| Perangkat keras | Prosesor: Intel® Core™ i5-7400 CPU @ 3.00GHz (4 CPUs) , ~3.0GHz  Memori: 8192 MB |
| Perangkat lunak | Sistem Operasi: Microsoft Windows 10 Pro 64-bit  Perangkat Pengembang: Phalcon  Perangkat Perancang Diagram:  Sybase Power Designer 16  Perangkat Database:  Microsoft SQL Server 2017 |

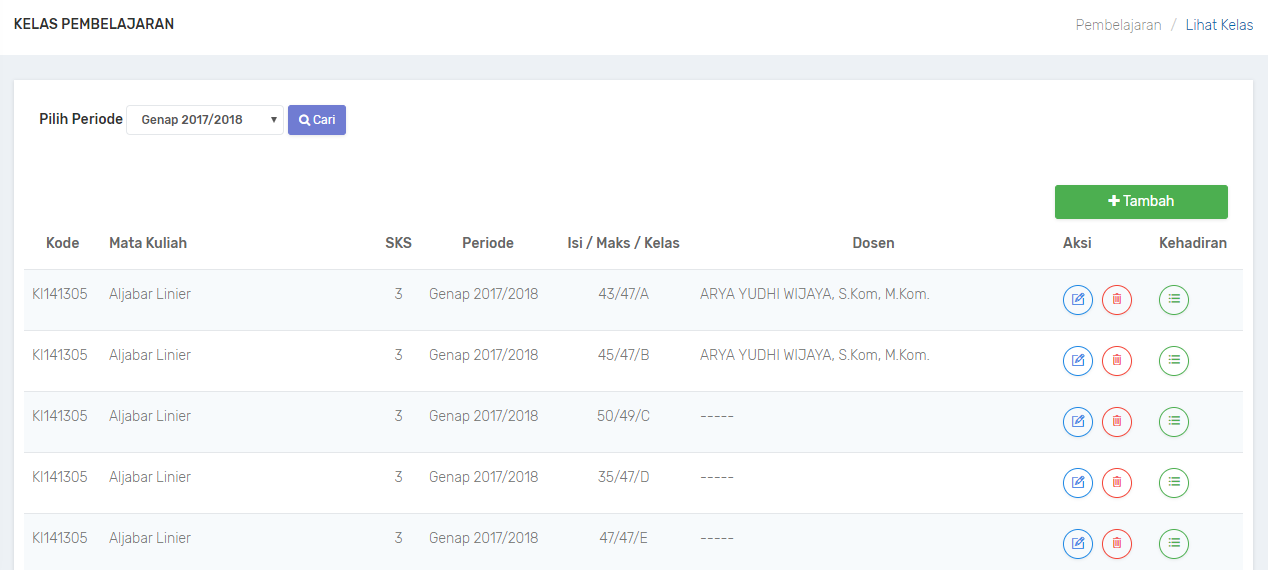
## Implementasi Antarmuka Sistem

Implementasi antarmuka sistem dilakukan dengan menggunakan file volt untuk masing-masing halaman. Berikut ini akan dijelaskan mengenai implementasi antarmuka sistem yang sudah direalisasikan.

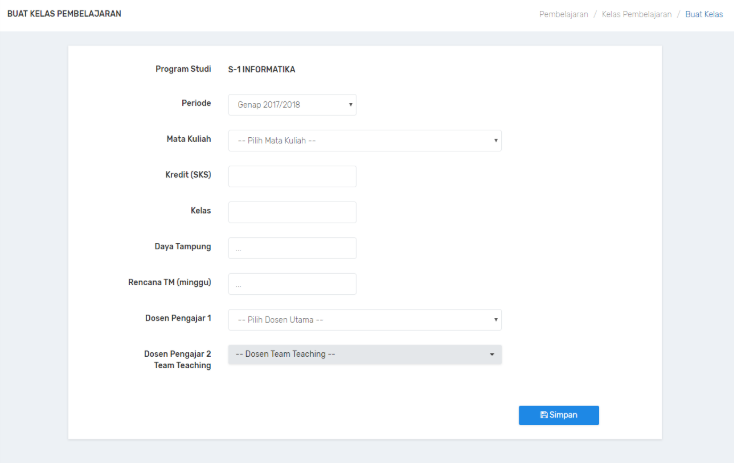
### Antarmuka Modul Pembelajaran

#### Halaman Mengelola Kelas Pembelajaran

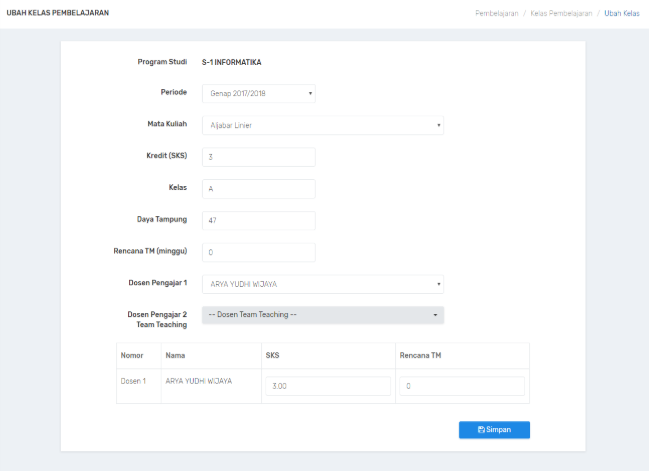
Halaman ini merupakan halaman yang digunakan untuk mengelola kelas pembelajaran. Halaman antarmuka yang menampilkan kelas pembelajaran ditunjukkan pada Gambar 4.1, menambah kelas pembelajaran ditunjukkan pada Gambar 4.2, dan pengubahan kelas ditunjukkan pada Gambar 4.3.



Gambar 4.1 Halaman Antarmuka Kelas Pembelajaran



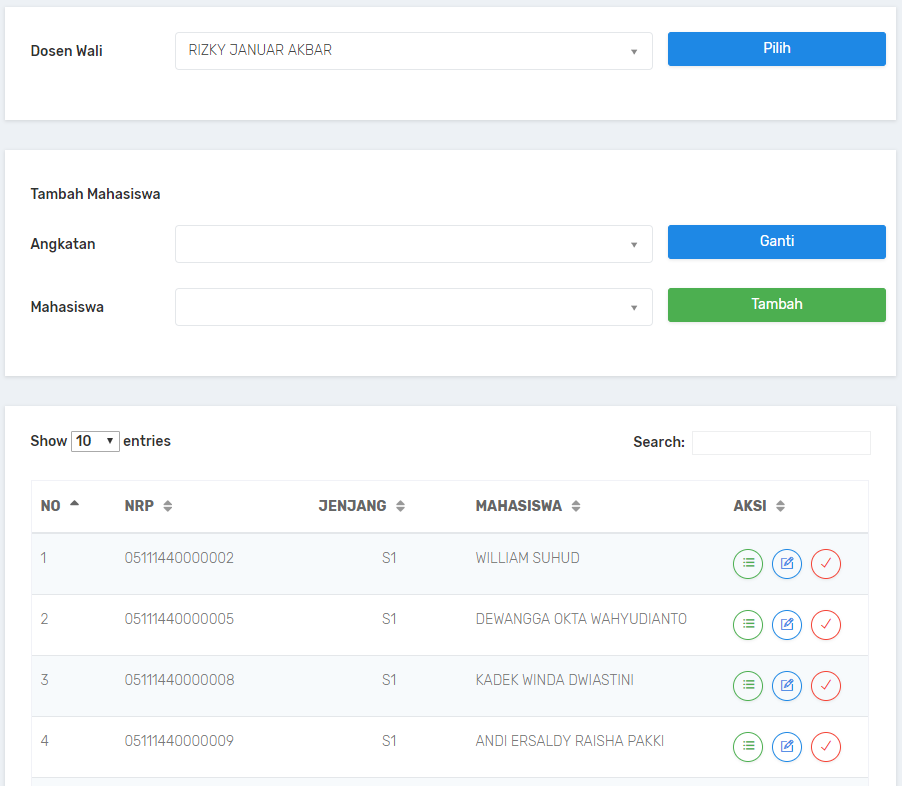
Gambar 4.2 Halaman Antarmuka *Form* Buat Kelas



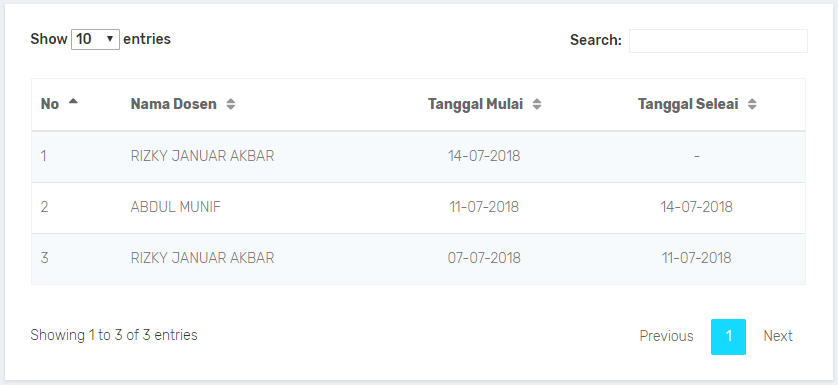
Gambar 4.3 Halaman Antarmuka *Form* Ubah Kelas

#### 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.



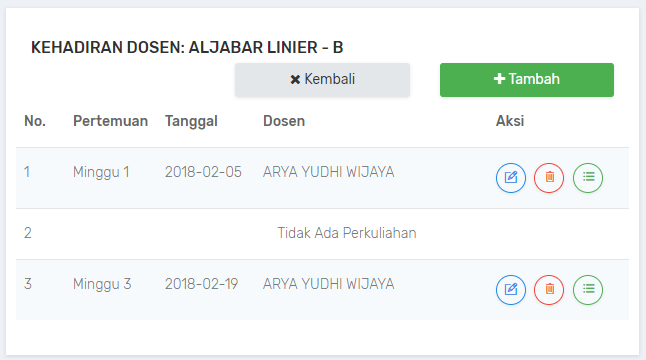
Gambar 4.4 Halaman Antarmuka Kelola Pendamping Akademik



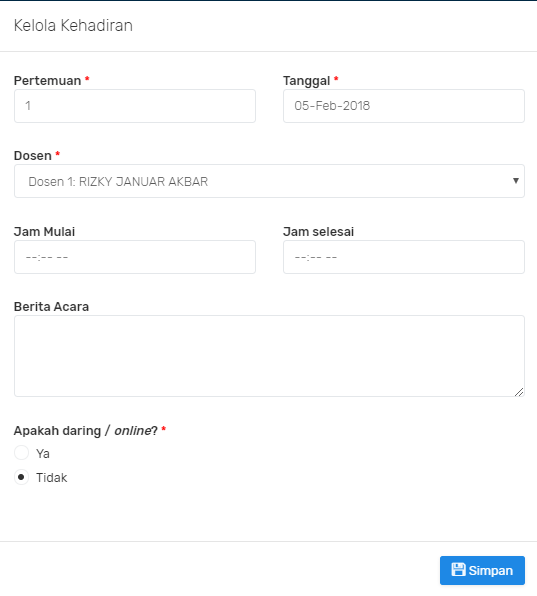
Gambar 4.5 Halaman Antarmuka Riwayat Perwalian Mahasiswa

#### 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

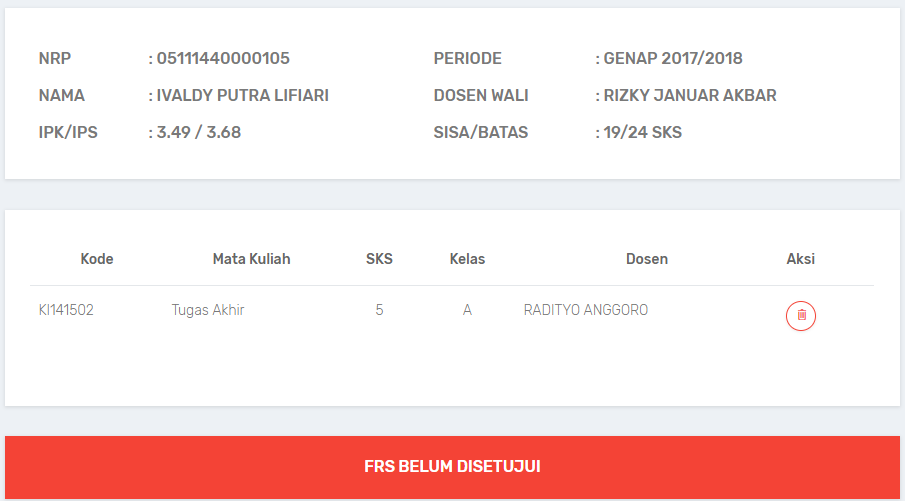
Halaman ini merupakan halaman yang digunakan untuk mengelola kehadiran mahasiswa pada suatu kelas pembelajaran. Halaman antarmuka yang menampilkan daftar kehadiran mahasiswa ditunjukkan pada Gambar 4.8.



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

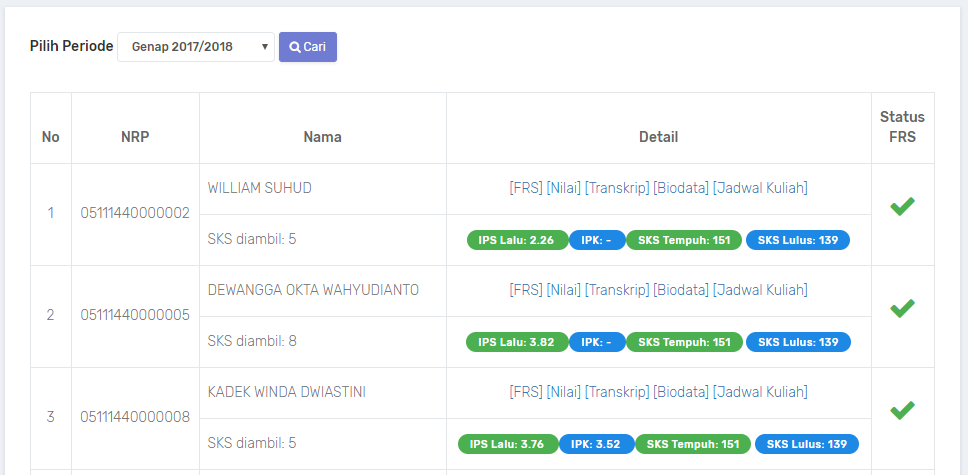
Halaman ini merupakan halaman yang digunakan untuk menyetujui halaman FRS. Pada Gambar 4.11 dosen yang bersangkutan dapat melakukan persetujuan terhadap FRS salah satu anak wali yang dipilihnya



Gambar 4.11 Halaman Antarmuka Menyetujui FRS

#### Halaman Menampilkan Anak Wali

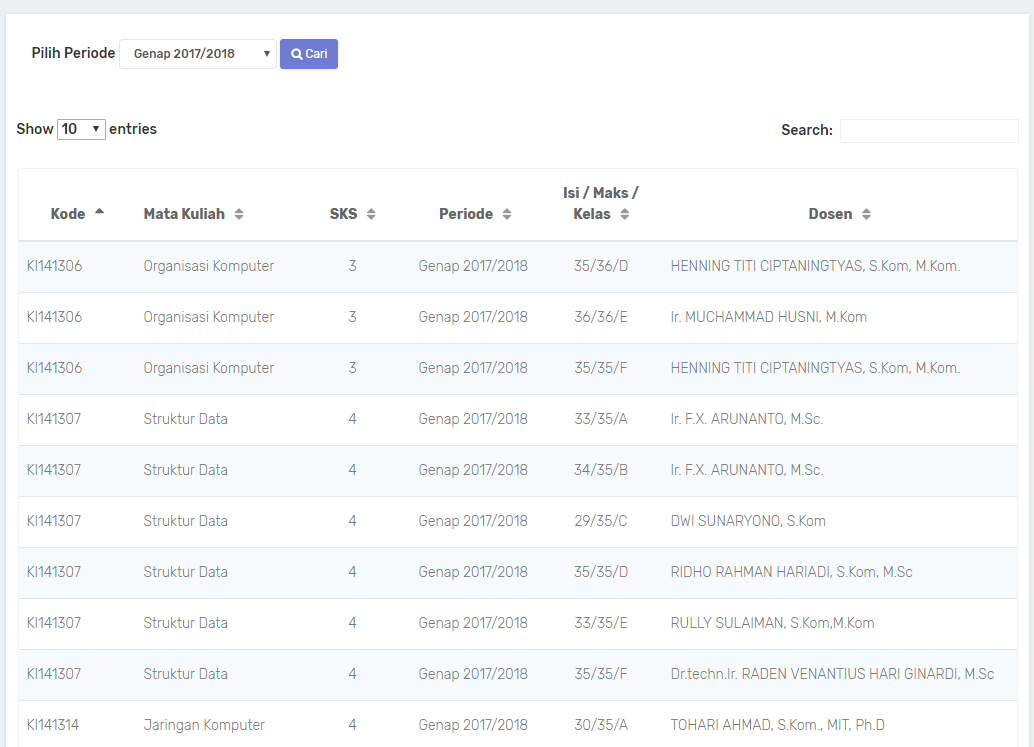
Halaman ini merupakan halaman yang digunakan untuk menampilkan daftar seluruh anak wali dari suatu dosen seperti yang ditunjukkan pada Gambar 4.12.



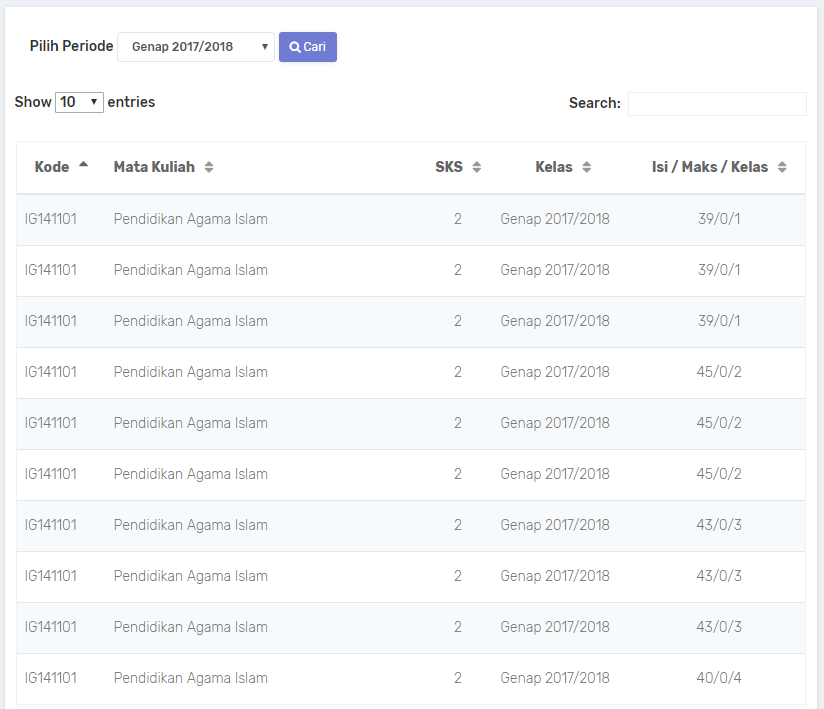
Gambar 4.12 Halaman Antarmuka Daftar Anak Wali

#### 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.



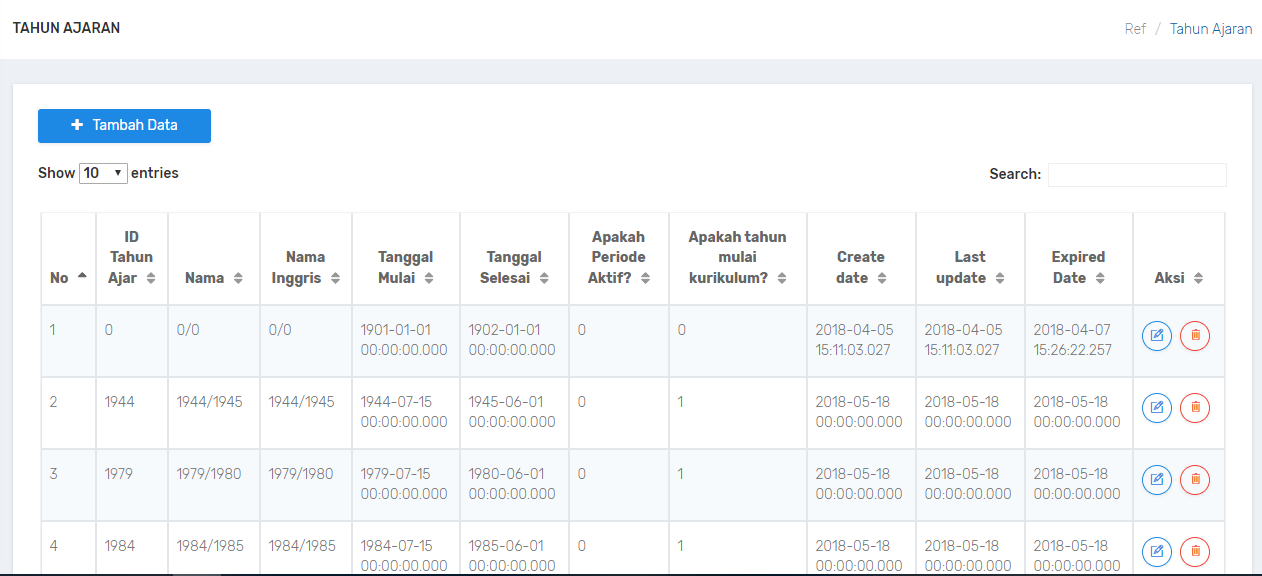
Gambar 4.13 Halaman Antarmuka Daftar Kelas Departemen



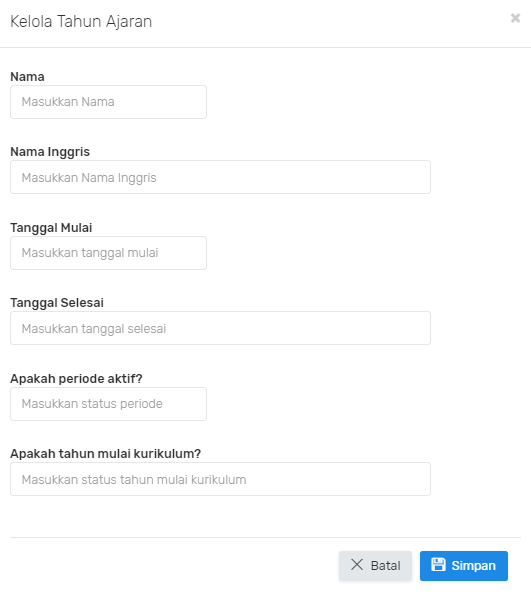
Gambar 4.14 Halaman Antarmuka Daftar Kelas UPMB

#### 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.



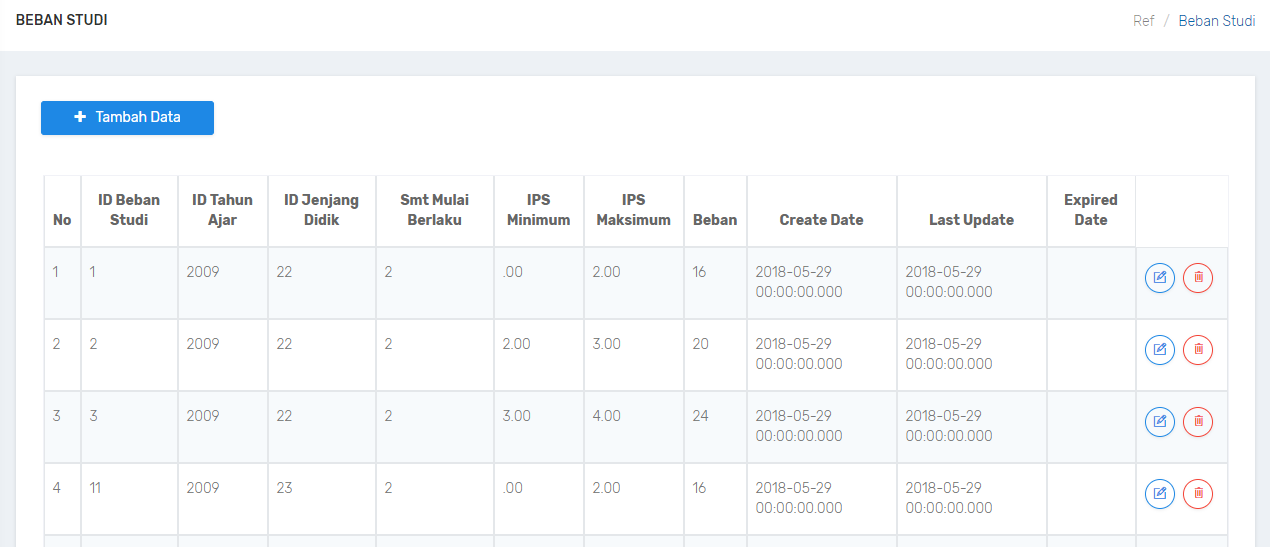
Gambar 4.15 Halaman Antarmuka Tahun Ajaran



Gambar 4.16 Halaman Antarmuka *Form* Tahun Ajaran

#### Halaman Mengelola Beban Studi

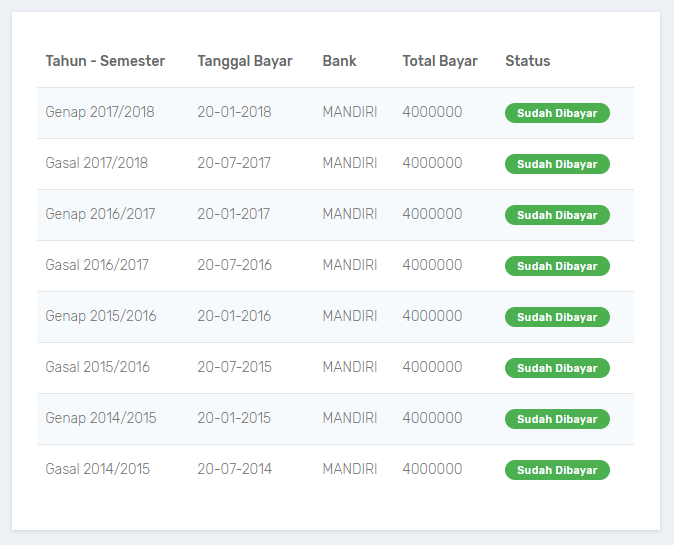
Halaman ini merupakan halaman yang digunakan untuk mengelola data beban studi. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.17.



Gambar 4.17 Halaman Antarmuka Beban Studi

#### Halaman Menampilkan Riwayat Pembayaran

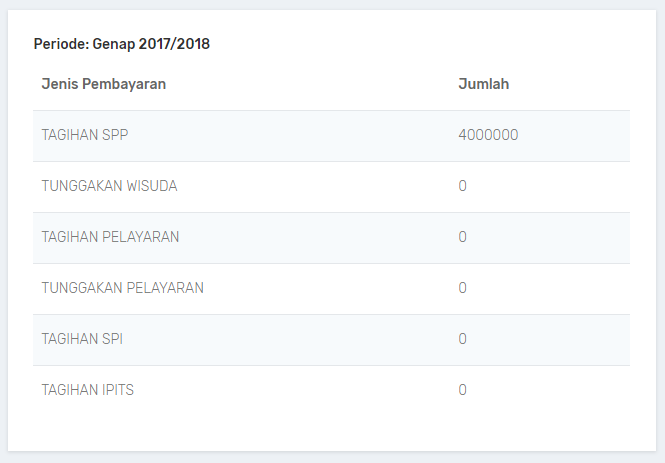
Halaman ini merupakan halaman yang digunakan untuk menampilkan riwayat pembayaran UKT milik aktor. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.18.



Gambar 4.18 Halaman Antarmuka Riwayat Pembayaran

#### Halaman Menampilkan Tagihan

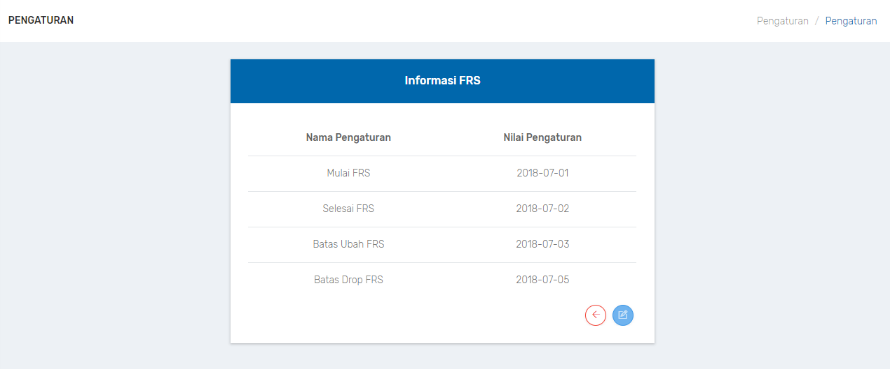
Halaman ini merupakan halaman yang digunakan untuk menampilkan data tagihan milik aktor pada suatu periode. Halaman antarmuka yang menampilkan data seluruh beban studi ditunjukkan pada Gambar 4.19.



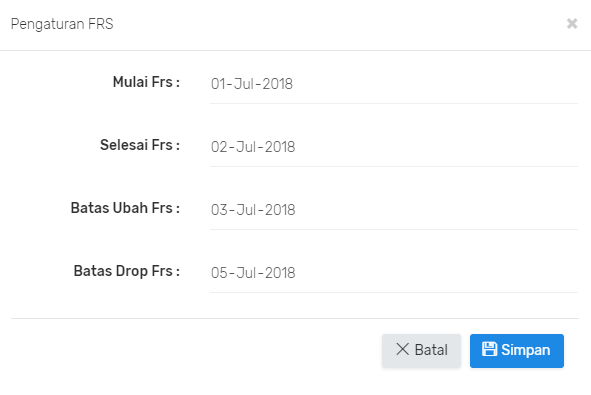
Gambar 4.19 Halaman Antarmuka Tagihan

#### 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

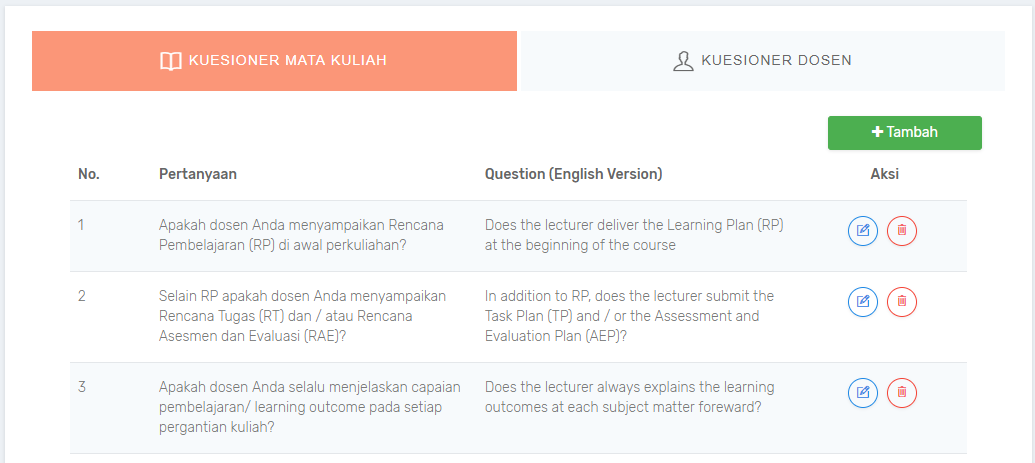


Gambar 4.21 Halaman Antarmuka *Font* Tanggal Penting FRS

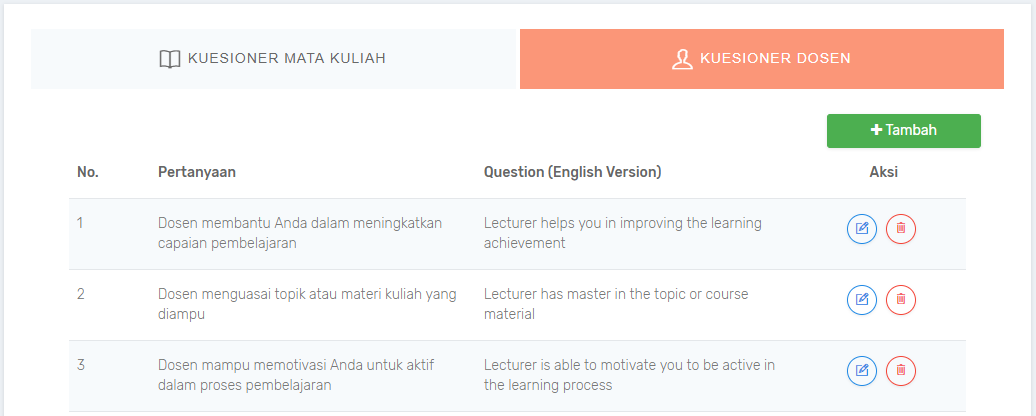
### Antarmuka Modul IPD

#### Halaman Mengelola Kuesioner Mata Kuliah dan Dosen

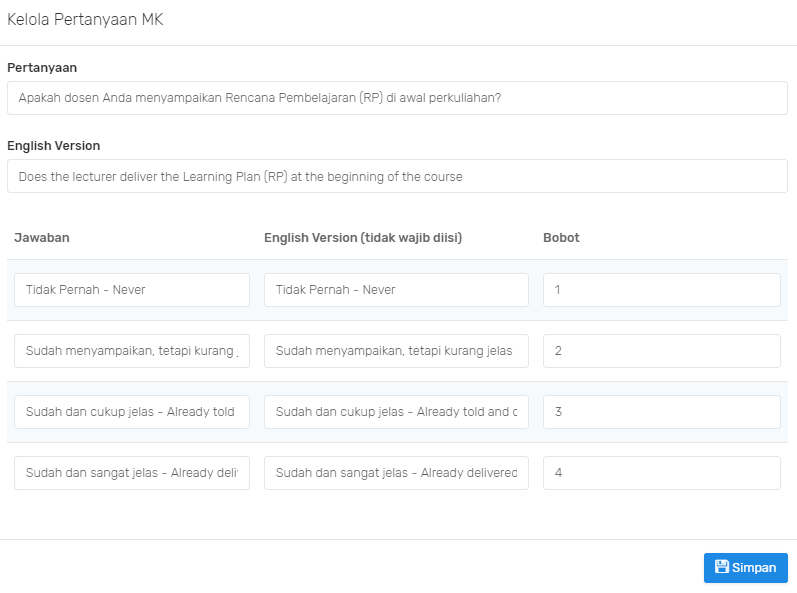
Halaman ini merupakan halaman yang digunakan untuk mengelola pertanyaan dan jawaban kuesioner IPD. Halaman yang menampilkan daftar kuesioner ditunjukkan pada Gambar 4.22 dan Gambar 4.23, dan pengelolaan kuesioner ditunjukkan pada Gambar 4.24.



Gambar 4.22 Halaman Antarmuka Kelola Kuesioner Mata Kuliah



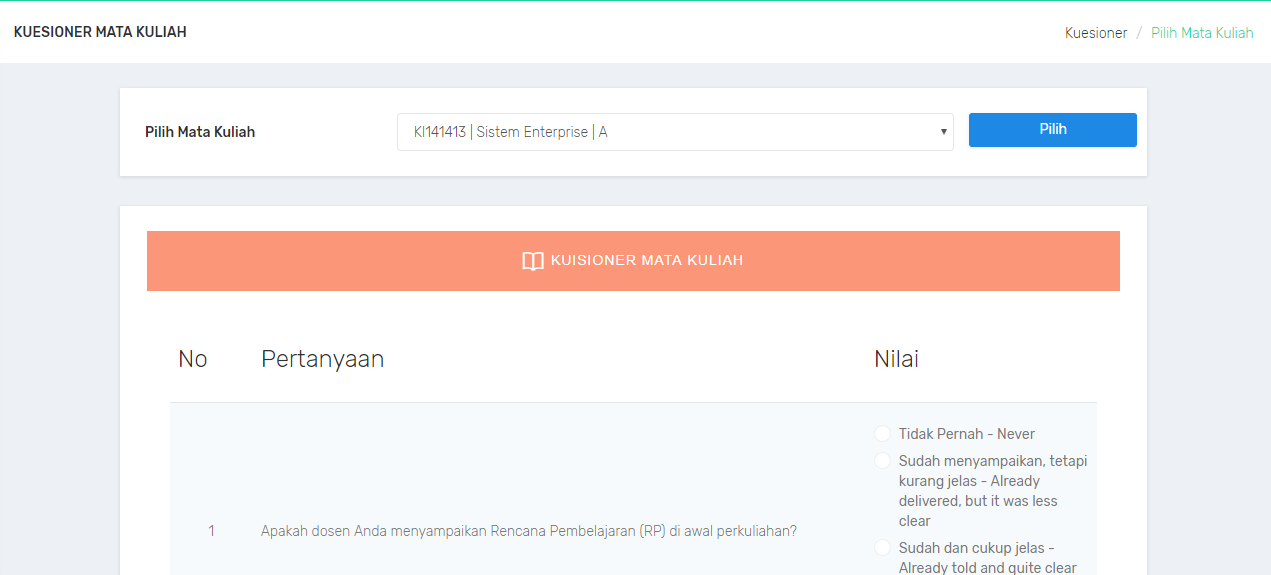
Gambar 4.23 Halaman Antarmuka Kelola Kuesioner Dosen



Gambar 4.24 Halaman Antarmuka *Form* Kelola Kuesioner

#### Halaman Mengisi Kuesioner Mata Kuliah dan Dosen

Halaman ini merupakan halaman yang digunakan untuk mengisi kuesioner IPD mata kuliah atau dosen. Aktor diminta untuk memilih salah satu mata kuliah nya sebelum dapat mengisi kuesioner seperti yang ditunjukkan pada Gambar 4.25.



Gambar 4.25 Halaman Antarmuka Mengisi Kuesioner IPD

#### Halaman Melihat Daftar Kuesioner

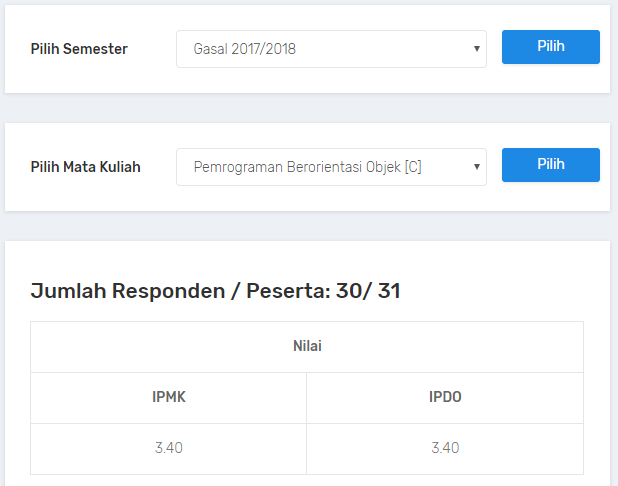
Halaman ini merupakan halaman yang digunakan untuk melihat daftar pertanyaan dan jawaban IPD. Halaman antarmuka dapat menampilkan daftar pertanyaan dan jawaban yang ada seperti yang ditunjukkan pada Gambar 4.26



Gambar 4.26 Halaman Antarmuka Melihat Daftar Kuesioner

#### Halaman Melihat Hasil IPD Dosen

Halaman ini merupakan halaman yang digunakan untuk melihat hasil IPD suatu dosen terhadap mata kuliah yang diajarkannya. Halaman antarmuka dapat menampilkan hasil nilai IPD seperti yang ditunjukkan pada Gambar 4.27.



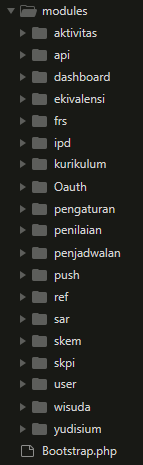
Gambar 4.27 Halaman Antarmuka Melihat Hasil IPD Dosen

## 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.

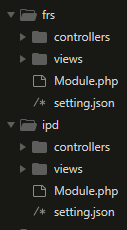
Implementasi MVC pada aplikasi dilakukan dengan pengadaan *package controller* yang berisikan kelas-kelas *controller*, *package service* yang berisikan kelas-kelas *service*, *package repository* yang berisikan kelas-kelas *repository* dan *package domain* yang berisikan implementasi basis data.

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



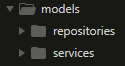
Gambar 4.28 Struktur direktori pada *folder* *modules*

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

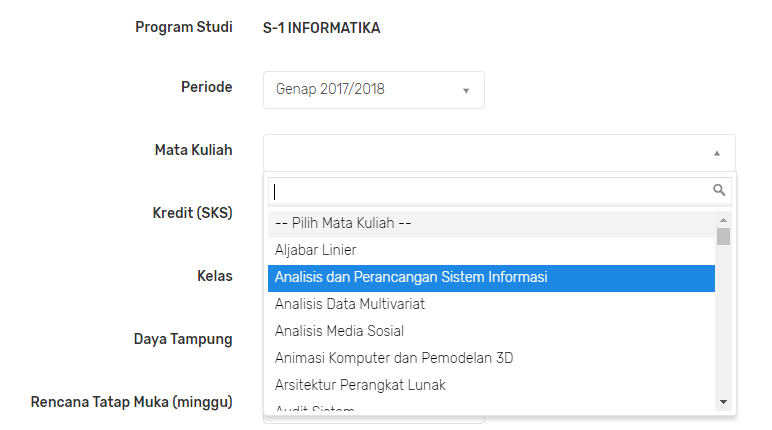
Seperti yang telah ditunjukkan oleh Gambar 3.55, modul FRS memiliki relasi dengan modul kurikulum dan modul penjadwalan. Kode Sumber 4.1 menunjukkan implementasi kode pada *controller* untuk memanggil *service* yang memanggil data mata kuliah. Kode Sumber 4.2 menunjukkan implementasi kode pada *service* 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

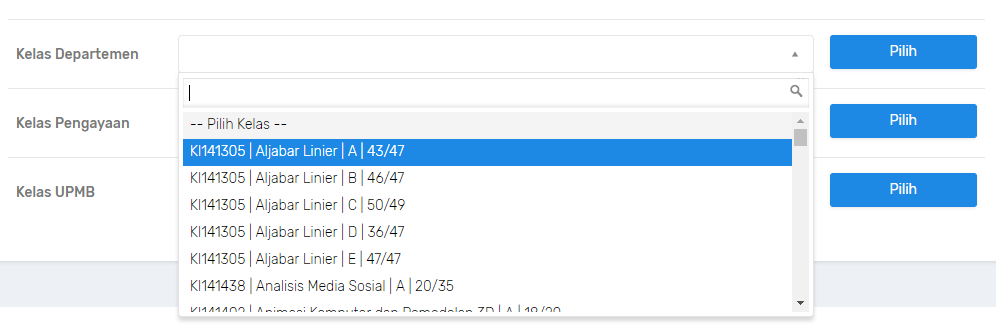
Sementara Kode Sumber 4.3 menujukkan fungsi untuk memanggil daftar kelas dari *controller* kepada *service* dan Kode Sumber 4.4 menunjukkan fungsi untuk memanggil daftar kelas dari *service* kepada *repository.* Gambar 4.32 menunjukkan implementasi relasi antar modul pembelajaran dan modul penjadwalan 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

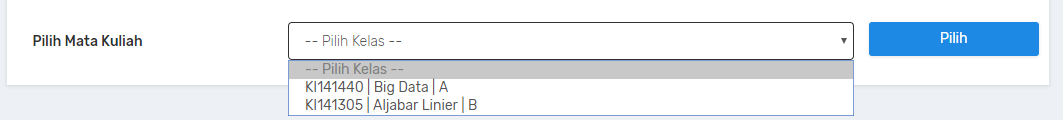
IPD juga melakukan relasi dengan modul pembelajaran dengan cara mengambil data kelas yang dipilih seorang mahasiswa ketika melakukan FRS. Kode Sumber 4.5 menunjukkan kode pada *controller* yang tersambung kepada *service* untuk memanggil data kelas yang diambil mahasiswa dan Kode Sumber 4.6 menunjukkan 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]*

# BAB V PENGUJIAN DAN EVALUASI

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



## Lingkungan Uji Coba

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]*

# BAB VI KESIMPULAN DAN SARAN

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



## Kesimpulan

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.

## Saran

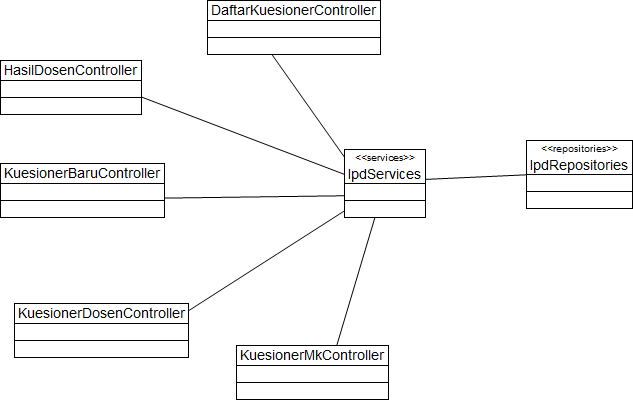
Terdapat beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

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.

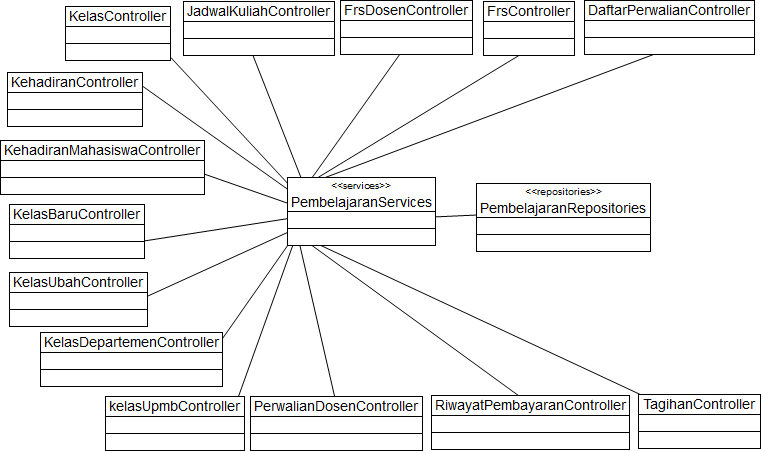
# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [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. |

# LAMPIRAN



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).