Sistem Informasi Penjadwalan Perkuliahan IT Del



Disusun Oleh :

|  |  |
| --- | --- |
| 12S22005 | Nikita Simanjuntak |
| 12S22016 | Desri Stevie Natalie Dabukke |
| 12S22019  12S22049  12S22050 | Liony Tamara Lewinsky  Agnes Monica Sanjani Harefa  Yohana Christine Sitanggang |

Untuk :

Biro Administrasi Akademik Institut Teknologi Del

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Proyek Sistem Informasi 2025**  **Teknologi Informatika Del** | | | |  |
| *No. Dokumen: SyRS-PSI-25-07* | | *Versi: 01.00* | *Tanggal : 23-01-2025* | *Jumlah Halaman :* | |

**DAFTAR ISI**

[1 Pendahuluan 7](#_Toc194324313)

[1.1 Tujuan Penulisan Dokumen 7](#_Toc194324314)

[1.2 Konvensi Dokumen 7](#_Toc194324315)

[1.3 Dokumen Referensi 8](#_Toc194324316)

[Ikhtisar Dokumen 8](#_Toc194324317)

[2 Gambaran Umum Sistem 9](#_Toc194324318)

[2.1 Gambaran Umum Sistem Saat Ini 9](#_Toc194324319)

[2.1.1 Proses Bisnis Membuat Jadwal dan Merilis Jadwal [PB01] 9](#_Toc194324320)

[2.2.2 Proses Bisnis Melakukan *Ticketing* dengan Preferensi *User* [PB02] 13](#_Toc194324321)

[3. Kebutuhan Fungsional 16](#_Toc194324322)

[3.1 Use Case Diagram 16](#_Toc194324323)

[3.2 Use Case Scenario 17](#_Toc194324324)

[3.2.1 Use Case Scenario Login 17](#_Toc194324325)

[3.2.2 Use Case Scenario Mengelola Akun 18](#_Toc194324326)

[3.2.3 Use Case Scenario Publikasi Jadwal 19](#_Toc194324327)

[3.2.4 Use Case Scenario Kelola Ruang Kelas 20](#_Toc194324328)

[3.2.5 Use Case Scenario Kelola Dosen 21](#_Toc194324329)

[3.2.6 Use Case Scenario Kelola Asisten Akademik 23](#_Toc194324330)

[3.2.7 Use Case Scenario Melihat Jadwal Perkuliahan 24](#_Toc194324331)

[3.2.8 Use Scenario Kelola Ticketing 25](#_Toc194324332)

[3.2.9 Use Case Scenario Memasukkan Mata Kuliah 26](#_Toc194324333)

[3.2.10 Use Case Scenarion Memasukkan Ruang Kelas 28](#_Toc194324334)

[3.2.11 Use Case Scenario Memasukkan Inisial Dosen 30](#_Toc194324335)

[3.2.12. Use Case Scenario Memasukkan Program Studi 31](#_Toc194324336)

[3.2.13 Use Case Scenario Evaluasi Ticketing 33](#_Toc194324337)

[3.3 Context Diagram 35](#_Toc194324338)

[3.4 DFD Level 1 36](#_Toc194324339)

[3.5 Main Features 37](#_Toc194324340)

[3.5.1 [SyRS-FR-01] Fitur Registrasi 37](#_Toc194324341)

[3.5.2 [SyRS-FR-02] Fitur Login 38](#_Toc194324342)

[3.5.3 [SyRS-FR-03] Fitur Mengelola Akun 39](#_Toc194324343)

[3.5.4 [SyRS-FR-04] Fitur Publikasi Jadwal 40](#_Toc194324344)

[3.5.5 [SyRS-FR-05] Fitur Melihat Jadwal Perkuliahan 41](#_Toc194324345)

[3.5.6 [SyRS-FR-06] Fitur Ticketing untuk Perubahan Jadwal 41](#_Toc194324346)

[3.6 Users Characteristics 42](#_Toc194324347)

[3.6.1 User-Group-[Akademik] 43](#_Toc194324348)

[3.6.2 User-Group-[Dosen] 43](#_Toc194324349)

[3.6.3 User-Group-[Mahasiswa] 43](#_Toc194324350)

[4 Kebutuhan Data dan Antarmuka 44](#_Toc194324351)

[4.1 Data Requirement 44](#_Toc194324352)

[4.1.1 ER-Diagram 44](#_Toc194324353)

[4.2 Kebutuhan Antarmuka 44](#_Toc194324354)

[4.2.1 External Interface 44](#_Toc194324355)

[4.2.2 User Interface 45](#_Toc194324356)

[4.2.3 Hardware Interface 45](#_Toc194324357)

[4.2.4 Software Interface 46](#_Toc194324358)

[5 Kebutuhan Lainnya 47](#_Toc194324359)

[5.1 System Performance Requirement 47](#_Toc194324360)

[5.2 Enabling Requirement 48](#_Toc194324361)

[5.3 Constraint Requirement 48](#_Toc194324362)

[5.4. SW Environment 49](#_Toc194324363)

[5.4.1. Development Environment 49](#_Toc194324364)

[5.4.2. Operational Environment 49](#_Toc194324365)

[6. Traceability 51](#_Toc194324366)

[Traceability Functional Requirement Summary 52](#_Toc194324367)

[LAMPIRAN 55](#_Toc194324368)

[Sejarah Versi 56](#_Toc194324369)

[Sejarah Perubahan 57](#_Toc194324370)

**DAFTAR TABEL**

[Tabel 1 1 Use Case Scenarion Login 15](#_Toc194317312)

[Tabel 1 2 Use Case Scenario Mengelola Akun 17](#_Toc194317313)

[Tabel 1 3 Use Case Scenario Publikasi Jadwal 18](#_Toc194317314)

[Tabel 1 4 Use Case Scenario Kelola Ruang Kelas 19](#_Toc194317315)

[Tabel 1 5 Use Case Scenario Kelola Dosen 21](#_Toc194317316)

[Tabel 1 6 Use Case Scenario Kelola Asisten Akademik 23](#_Toc194317317)

[Tabel 1 7 Use Case Scenario Melihat Jadwal Perkuliahan 24](#_Toc194317318)

[Tabel 1 8 Use Scenario Kelola *Ticketing* 25](#_Toc194317319)

[Tabel 1 9 Use Case Scenario Memasukkan Mata Kuliah 27](#_Toc194317320)

[Tabel 1 10 Use Case Scenario Memasukkan Ruang Kelas 29](#_Toc194317321)

[Tabel 1 11 Use Case Scenario Memasukkan Inisial Dosen 31](#_Toc194317322)

[Tabel 1 12 Memasukkan Program Studi 33](#_Toc194317323)

[Tabel 1 13 Use Case Scenario Evaluasi *Ticketing* 35](#_Toc194317324)

[Tabel 2 1 User Interface 47](#_Toc194317325)

[Tabel 2 2 Hardware Interface 48](#_Toc194317326)

[Tabel 3 1 System Performance Requierement 49](#_Toc194317327)

[Tabel 3 2 Enabling Requirement 50](#_Toc194317328)

[Tabel 4 1 Traceability 53](#_Toc194317333)

**DAFTAR GAMBAR**

[Gambar 1 BPMN Current System 9](#_Toc194317356)

[Gambar 2 Bussines Process Target System 11](#_Toc194317357)

[Gambar 3 Bussiness Process *Ticketing* oleh User 12](#_Toc194317358)

[Gambar 4 Bussiness Process Mengelola User 13](#_Toc194317359)

[Gambar 5 Use Case Diagram 15](#_Toc194317360)

[Gambar 6 Context Diagram 37](#_Toc194317361)

[Gambar 7 DFD Level 1 38](#_Toc194317362)

[Gambar 8 ER Diagram 46](#_Toc194317363)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen *System Requirement Specification* (SyRS) ini dibuat untuk mendefinisikan spesifikasi sistem yang akan dikembangkan, yaitu Sistem Informasi Penjadwalan IT Del berbasis website. Dokumen ini digunakan oleh pengembang sistem, termasuk *programmer*, *system analyst*, dan *desainer*, sebagai panduan dalam perancangan dan implementasi sistem.

## Konvensi Dokumen

Konvensi atau aturan yang digunakan dalam dokumen ini mencakup aturan penulisan penamaan dokumen ini menggunakan format penulisan dengan aturan sebagai berikut:

1. Aturan penamaan dimana YY adalah tahun pembuatan dokumen dan GG adalah nomor kelompok. Maka dokumen ini dinamai dengan SyRS-PSI-20-10 dokumen dengan ketentuan: SyRS-PSI-YY GG.

1. Aturan penulisan:
2. *Font: Times New Roman,* ukuran 12pt.

* Heading 1: bold, 14pt.
* Heading 2: bold, 12pt.
* Heading 3: bold, 12pt.

1. Paragraf:

* *Line spacing*: 1.5 *lines*.
* *Before and after spacing* pada paragraf.
* Paragraf rata kiri-kanan (justify).
* Istilah asing ditulis dengan format *italic*.

1. Judul Tabel/Gambar:

* *Line-spacing: single, before and after*.
* Ukuran huruf: 10pt.
* *Caption tabel* rata tengah dan diletakkan di atas tabel.
* *Caption* gambar rata tengah dan diletakkan di bawah gambar.

1. Penulisan header:

* *Times New Roman*, ukuran 10pt, rata tengah (*justify*).
* Menggunakan *repeated table* (mengulang header tabel jika isi tabel berada di beberapa halaman dokumen).

3. Aturan penomoran dan penamaan bab dan subbab:

1. Bab: 1,2,3 dan seterusnya.

Contoh: 1 Introduction.

1. Subbab: 1.1, 1.2, 1.3, dan seterusnya.

Contoh: 1.1 Purpose of Document 3.

1. Sub subbab: 1.1.1, 1.1.2, 1.1.3, dan seterusnya.

Contoh: 2.2.1 Business Process Membuat Pengaduan

1. Penomoran dan penulisan pada tabel dan gambar.

Contoh: Tabel 1. Daftar Definisi

## Dokumen Referensi

Dokumen yang menjadi rujukan dokumen ini

1. Std-KA-09, Standar Penomoran Dan Tatanama artefak Proyek Sistem Informasi dan Tugas Akhir, terbitan tahun 2009.
2. ToR-PSI-25-07, Term of Reference Sistem Informasi Penjadwalan Perkuliahan IT Del
3. PiP-PSI-25-07, Project Implementation Plan Sistem Informasi Penjadwalan Perkuliahan IT Del

## Ikhtisar Dokumen

Dokumen System Requirement Specification (SyRS) *Sistem Informasi Penjadwalan IT Del Berbasis Website* terdiri dari 6 bab yaitu:

1. Bab 1 *Introduction* menjelaskan tentang latar belakang dan tujuan penulis dokumen, aturan penulis dan referensi yang dipakai untuk membangun sistem.
2. Bab 2 *System Overview* menjelaskan tentang ruang lingkup dari *Current system* dan *Target system.*
3. *Bab 3 Functional Requirement* menjelaskan tentang deskripsi dari fungsi utama dari sistem serta karakteristik *user* yang mencakup hak akses setiap *user* dalam mengoperasi sistem.
4. Bab 4 *Data and Interface* Requirement menjelaskan tentang kebutuhan data yang digunakan dalam melakukan pengoperasian terhadap sistem.
5. Bab 5 *Other Requirement* menjelaskan tentang kebutuhan non-fungsional sistem, enabling requirement sistem, dan batasan yang dibutuhkan oleh sistem.

# Gambaran Umum Sistem

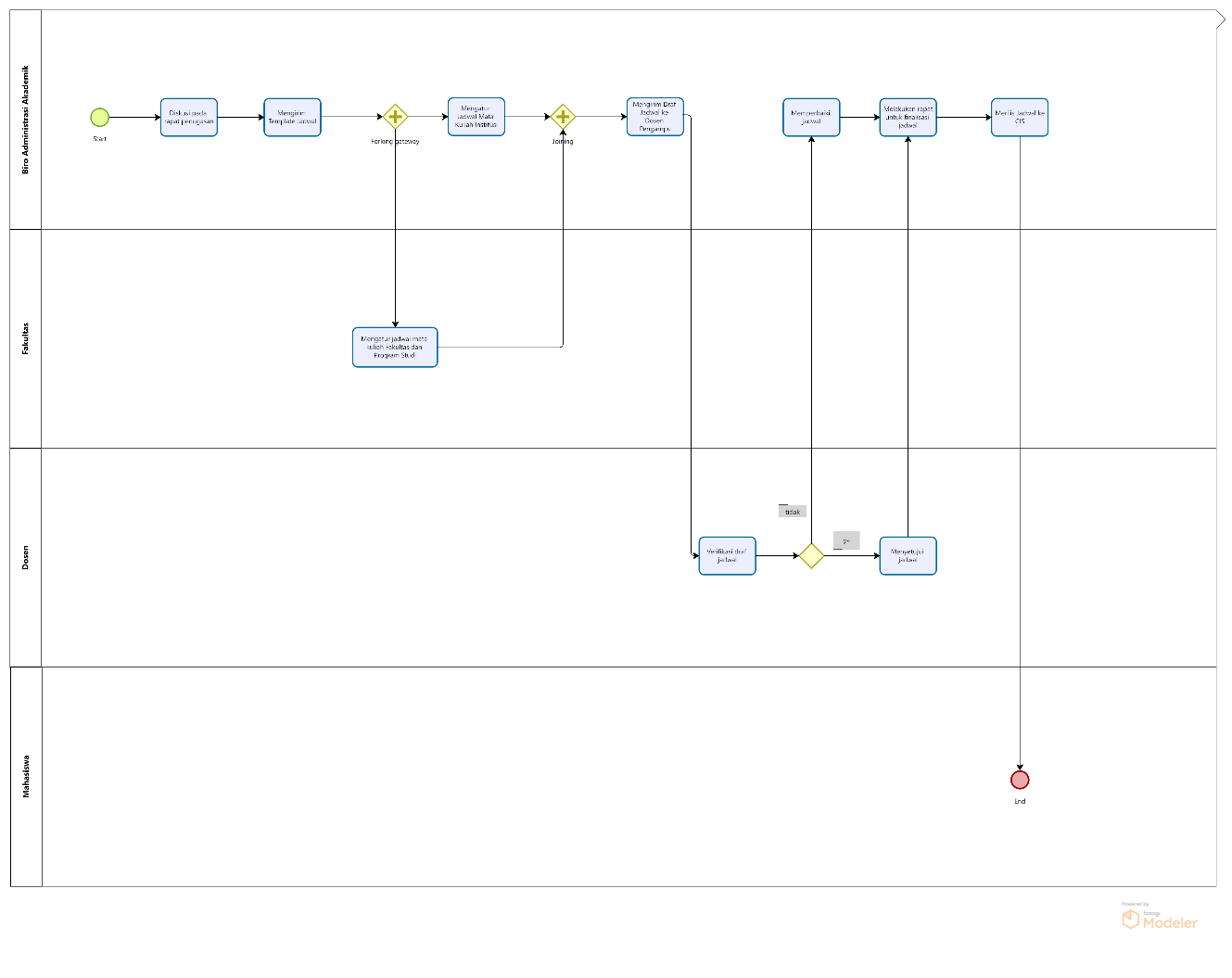
Pada bab ini dijelaskan deskripsi secara umum mengenai sistem yang sedang berjalan saat ini (*current system*) dan sistem yang ditargetkan (*target system*) dengan adanya Sistem Informasi Penjadwalan Perkuliahan IT Del. Bagian - bagian yang dijelaskan mencakup proses bisnis, prosedur dari setiap proses bisnis, serta *service time* yang digunakan dalam pengembangan sistem informasi.

## Gambaran Umum Sistem Saat Ini

Sistem penjadwalan perkuliahan di IT Del saat ini masih menggunakan metode semi-manual yang membutuhkan banyak intervensi dari Bagian Administrasi Akademik (BAA). Pembuatan dan pengelolaan jadwal dilakukan dengan spreadsheet atau sistem sederhana berbasis algoritma dan formula di *Excel*, yang masih memiliki keterbatasan dalam menangani perubahan jadwal secara dinamis serta menghindari konflik antara mata kuliah, dosen, dan ruang perkuliahan. Dengan adanya sistem informasi penjadwalan, diharapkan proses penjadwalan dapat terotomatisasi dengan lebih baik, mengurangi intervensi manual, dan meningkatkan efisiensi dalam pengelolaan jadwal perkuliahan. Sistem ini juga akan memberikan akses yang lebih mudah dan transparan bagi semua pihak terkait, serta mengintegrasikan fitur *update* status untuk memastikan setiap perubahan jadwal dapat segera diketahui oleh dosen dan mahasiswa.

### Proses Bisnis Membuat Jadwal dan Merilis Jadwal [PB01]

Saat ini, sistem yang ada dalam penyusunan dan pemeliharaan jadwal perkuliahan masih melibatkan proses manual dengan menggunakan *Microsoft Excel***.** Proses ini dimulai dengan Biro Administrasi Akademik yang mendistribusikan templatejadwal kepada pihak terkait, yaitu Dosen dan ProgramStudi. Setelah menerima template, proses selanjutnya melibatkan PenjadwalanDosen dan Mata Kuliah, serta Manajemen jadwal oleh pihak terkait.



**Gambar 1 BPMN Current System**

Proses dimulai dengan pembuatan draft jadwal yang kemudian diverifikasi untuk memastikan kesesuaian dengan kebutuhan ruang kelas dan waktu yang tersedia. Setelah proses verifikasi selesai, JadwalFinal akan diposting dan diperbarui di *CIS (Campus Information System)* untuk dapat diakses oleh Mahasiswa.

Namun, pada sistem yang ada saat ini, pembaruan jadwal dilakukan secara manual, yang menyebabkan kesulitan dalam menyebarkan perubahan jadwal secara cepat dan efisien kepada seluruh pengguna. Proses manual ini juga meningkatkan risiko kesalahan dalam alokasi ruang, waktu kuliah, dan pengelolaan jadwal, yang mengarah pada ketidakefisienan dalam penyusunan jadwal perkuliahan.

Untuk mengatasi masalah ini, website berbasis Sistem Informasi Penjadwalan akan dibangun dengan tujuan untuk mengotomatisasi dan modernisasi pengelolaan jadwal, memungkinkan proses yang lebih cepat, akurat, dan terintegrasi. Dengan sistem ini, Biro Administrasi dan Akademik dapat menyusun, memperbarui, dan mendistribusikan jadwal secara lebih efisien, mengurangi ketergantungan pada metode manual, serta meningkatkan aksesibilitas informasi bagi Dosen**,** Mahasiswa**,** danBiro Administrasi dan Akademik.

#### Waktu Layanan

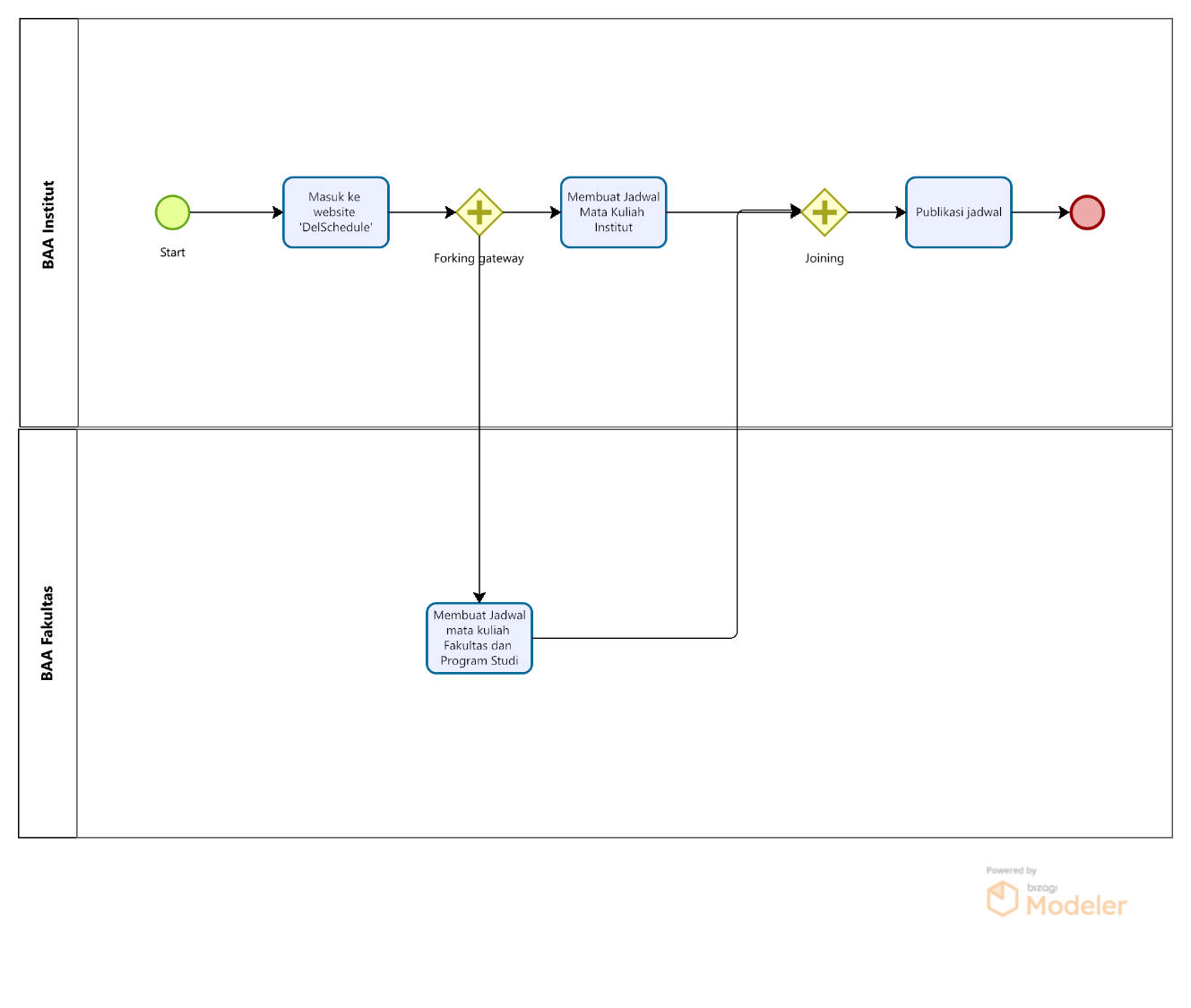
Waktu yang dibutuhkan untuk menyusun dan mengelola jadwal akademik secara manual di IT Del memakan waktu lebih dari 2 minggu setiap semester. Dalam rentang waktu tersebut sudah termasuk proses pengumpulan data jadwal, koordinasi dengan dosen dan staf akademik, penyusunan jadwal, pengecekan konflik jadwal, serta distribusi jadwal kepada mahasiswa dan dosen. Proses ini menunjukkan bahwa sistem manual yang digunakan saat ini memiliki efisiensi yang rendah dalam pengelolaan penjadwalan akademik.Waktu yang dibutuhkan untuk menyusun dan mengelola jadwal akademik secara manual di IT Del memakan waktu lebih dari 2 minggu setiap semester. Dalam rentang waktu tersebut sudah termasuk proses pengumpulan data jadwal, koordinasi dengan dosen dan staf akademik, penyusunan jadwal, pengecekan konflik jadwal, serta distribusi jadwal kepada mahasiswa dan dosen. Selain itu, perubahan atau penyesuaian jadwal seringkali memakan waktu tambahan karena perlu dilakukan secara manual. Proses ini menunjukkan bahwa sistem manual yang digunakan saat ini memiliki efisiensi yang rendah dalam pengelolaan penjadwalan akademik, serta berpotensi menimbulkan kesalahan dalam alokasi ruang dan waktu kuliah.

**2.2 Target Sistem**

Sistem yang diusulkan bertujuan untuk mengembangkan *DelSchedule*, sebuah platform berbasis web yang dirancang untuk mengotomatisasi dan menyederhanakan proses penjadwalan perkuliahan di IT Del. Sistem ini akan menggantikan metode semi-manual yang saat ini masih digunakan, seperti pengelolaan jadwal melalui spreadsheet dan sistem berbasis formula di *Excel*. Dengan adanya sistem ini, proses pembuatan, pengelolaan, dan publikasi jadwal perkuliahan dapat dilakukan dengan lebih efisien, akurat, dan terintegrasi.

Dalam sistem yang dicita-citakan, *DelSchedule* akan memiliki beberapa fitur utama, termasuk otomatisasi penjadwalan berdasarkan ketersediaan dosen, ruangan, dan slot waktu perkuliahan. Selain itu, sistem ini akan menyediakan validasi otomatis untuk mencegah konflik jadwal serta memungkinkan pembaruan jadwal secara real-time. Setiap pemangku kepentingan, seperti BAA Institut, BAA Fakultas, dosen, dan mahasiswa, akan memiliki akses ke jadwal yang telah diperbarui sesuai dengan peran masing-masing.

**2.2.1 Proses Bisnis Membuat Jadwal Baru [PB02]**

**

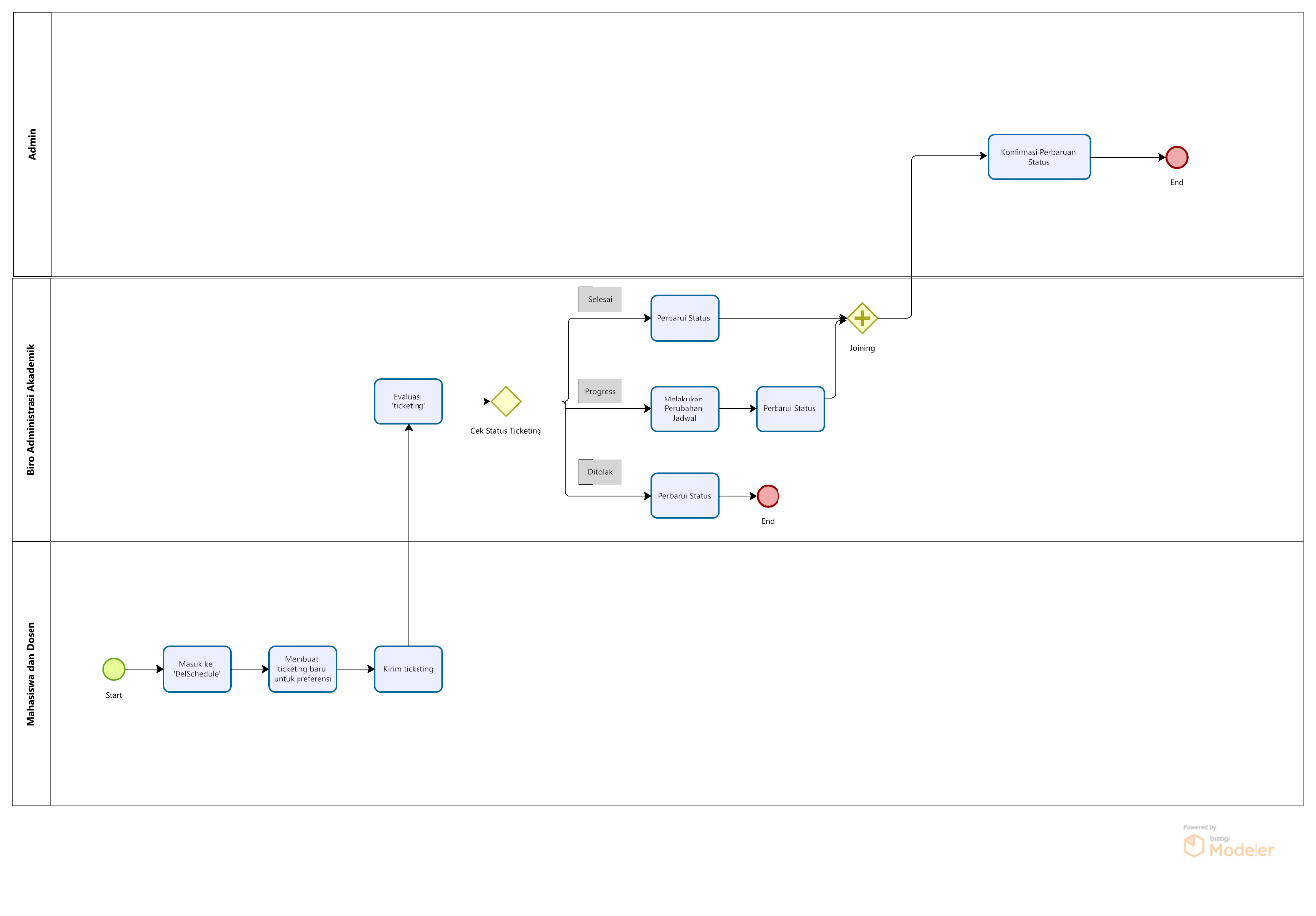
**Gambar 2 Bussines Process Target System**

Proses bisnis untuk pembuatan jadwal perkuliahan dimulai dengan langkah pertama yang dilakukan oleh Biro Administrasi Akademik (BAA Institut), yaitu masuk ke website *DelSchedule*. Platform ini digunakan untuk mengelola jadwal mata kuliah secara terpusat, menggantikan metode manual yang sebelumnya digunakan. Setelah masuk ke sistem, BAA Institut akan membuat jadwal mata kuliah tingkat institut. Pada tahap ini, sistem memungkinkan pembuatan jadwal untuk mata kuliah yang bersifat umum atau wajib bagi seluruh program studi. Proses ini kemudian bercabang (*forking gateway*), di mana BAA Fakultas turut serta dalam penyusunan jadwal mata kuliah yang lebih spesifik berdasarkan fakultas dan program studi. Setelah kedua pihak BAA Institut dan BAA Fakultas menyelesaikan pembuatan jadwal masing-masing, sistem akan melakukan penggabungan (*joining gateway*) sebelum jadwal akhir dipublikasikan. Publikasi ini memungkinkan dosen dan mahasiswa untuk mengakses serta melihat jadwal yang telah tersusun melalui platform *DelSchedule*.

**2.2.1.1 Waktu Layanan**

Waktu layanan dalam sistem *DelSchedule* akan lebih efisien dibandingkan dengan metode manual yang digunakan saat ini. Sistem berbasis website ini memungkinkan proses pembuatan dan pengelolaan jadwal dilakukan dalam waktu yang lebih singkat dan terstruktur. Layanan pembuatan jadwal akan tersedia selama 5 hingga 7 hari kerja dalam setiap periode penyusunan jadwal akademik.

### 2.2.2 Proses Bisnis Melakukan *Ticketing* dengan Preferensi *User* [PB02]

**

**Gambar 3 Bussiness Process Ticketing oleh User**

Proses dimulai dari *swimlane* Mahasiswa dan Dosen di bagian bawah diagram. Awalnya, Mahasiswa atau Dosen masuk ke website "*DelSchedule*" yang ditandai dengan *start event*, kemudian membuat *ticketing* baru untuk menyampaikan preferensi jadwal mereka, dan akhirnya mengirimkan *ticketing* tersebut ke dalam sistem untuk diproses lebih lanjut.

Selanjutnya, alur proses berpindah ke *swimlane* Biro Administrasi Akademik di bagian tengah diagram. Di sini, staf Biro Administrasi Akademik melakukan evaluasi terhadap *ticketing* yang masuk, lalu melakukan pengecekan status *ticketing* yang ditunjukkan dengan gateway decision. Berdasarkan hasil evaluasi, *ticketing* dapat memiliki tiga kemungkinan status: Selesai (jika permintaan dapat langsung diselesaikan), *Progress* (jika permintaan memerlukan penanganan lebih lanjut), atau Ditolak (jika permintaan tidak dapat dipenuhi).

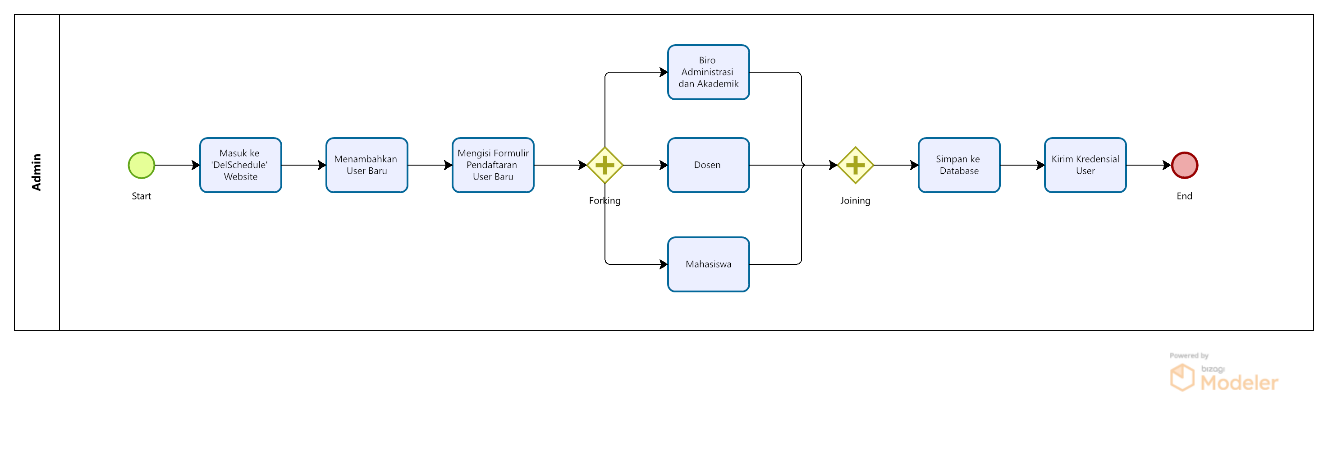
Jika status *ticketing* adalah "*Progress*", Biro Administrasi Akademik akan melakukan penjelasan jadwal terlebih dahulu, kemudian memperbarui status *ticketing* setelah penjelasan selesai dilakukan. Untuk setiap status akhir (Selesai, *Progress* setelah penjelasan, atau Ditolak), Biro Administrasi Akademik akan melakukan pembaruan status *ticketing* dalam sistem.

Pada bagian akhir proses, untuk *ticketing* dengan status Selesai atau Progress yang telah diperbarui, alur akan bergabung (*joining gateway*) dan berpindah ke *swimlane* Admin di bagian atas diagram. Admin kemudian melakukan konfirmasi perubahan status, dan proses berakhir dengan *end event.* Sementara itu, untuk *ticketing* dengan status Ditolak, proses langsung berakhir setelah pembaruan status tanpa melalui konfirmasi Admin, yang ditunjukkan dengan *end event* terpisah pada *swimlane* Biro Administrasi Akademik.

#### 2.1.1.2. Waktu Layanan

Proses *ticketing* dalam sistem *DelSchedule* dapat diajukan kapan saja melalui website. Permintaan akan dievaluasi oleh Biro Administrasi Akademik (BAA) dalam waktu maksimal 2x24 jam kerja. Jika tiket disetujui, perubahan jadwal akan segera diproses dengan mempertimbangkan ketersediaan ruang dan jadwal dosen. Sistem ini mempercepat layanan dibandingkan metode manual, memastikan pengguna mendapatkan pembaruan status secara otomatis.

**2.2.3 Proses Bisnis Mengelola User [PB03]**



**Gambar 4 Bussiness Process Mengelola User**

Diagram BPMN ini menggambarkan proses bisnis pengelolaan user pada sistem *DelSchedule* yang dijalankan sepenuhnya oleh Admin. Proses dimulai dengan Admin masuk ke website *DelSchedule*, dilanjutkan dengan menambahkan user baru ke dalam sistem. Setelah itu, Admin mengisi formulir pendaftaran dengan informasi yang diperlukan untuk akun pengguna baru tersebut.

Setelah pengisian formulir selesai, proses bercabang (*forking*) berdasarkan tipe user yang akan dikelola. Terdapat tiga jalur berbeda untuk kategori user: Biro Administrasi dan Akademik, Dosen, atau Mahasiswa. Masing-masing kategori memiliki kebutuhan data dan hak akses yang berbeda dalam sistem. Setelah kategori ditentukan dan informasi spesifik diinput, alur proses bergabung kembali (*joining*) dan data user disimpan ke database sistem.

Proses pengelolaan user diakhiri dengan pengiriman kredensial login kepada pengguna yang baru terdaftar, memastikan mereka dapat segera mengakses sistem *DelSchedule* sesuai dengan peran dan hak akses yang telah ditetapkan.

#### 2.1.3.1 Waktu Layanan

Proses pengelolaan user pada sistem *DelSchedule* memiliki *service time* yang cukup efisien. Tahap awal masuk ke website *DelSchedule* membutuhkan waktu sekitar 10-20 detik, diikuti dengan proses menambahkan user baru yang memerlukan waktu 1-2 menit tergantung pada kecepatan akses sistem.

Pengisian formulir pendaftaran user merupakan tahap yang membutuhkan waktu lebih lama, sekitar 3-5 menit, karena melibatkan input data yang cukup detail. Pemrosesan berdasarkan kategori user memerlukan waktu tambahan 2-3 menit per user untuk memastikan atribut spesifik kategori terisi dengan benar.

Tahap akhir yang terdiri dari penyimpanan data ke database dan pengiriman kredensial user membutuhkan waktu total sekitar 1,5-3 menit. Dengan demikian, total *service time* untuk keseluruhan proses pengelolaan user berkisar antara 8-15 menit per user, bergantung pada kompleksitas data yang dikelola..

# 3. Kebutuhan Fungsional

Pada bab ini akan mendeskripsikan kebutuhan fungsional dalam pengembangan sistem informasi penjadwalan perkuliahan. Kebutuhan fungsional tersebut meliputi gambaran secara umum mengenai fungsi-fungsi yang ada pada sistem.

## 3.1 Use Case Diagram



**Gambar 5 Use Case Diagram**

Pada gambar di atas, terdapat tiga aktor utama: Biro Administrasi dan Akademik, Admin, serta Dosen & Mahasiswa. Biro Administrasi dan Akademik bertugas memasukkan data terkait perkuliahan, Admin mengelola sistem, dan Dosen & Mahasiswa dapat melihat jadwal serta mengelola *ticketing*.

## 3.2 Use Case Scenario

### 3.2.1 Use Case Scenario Login

Tabel 1 1 Use Case Scenarion Login

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-1 | |
| *Use Case Name* | *Login* | |
| *Use Case Description* | Proses login dimana pengguna memasukkan kredensial untuk mengakses sistem. | |
| *Actor* | Admin, Dosen, Mahasiswa | |
| *Precondition* | Pengguna memiliki akun yang sudah terdaftar. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Pengguna membuka halaman login. |  |
|  | 2. Pengguna memasukkan *username* dan *password*. |  |
|  |  | 3. Sistem memvalidasi kredensial. |
|  | 4. Sistem memberikan akses ke dashboard dan mengarahkan pengguna ke antarmuka yang sesuai (Admin, Dosen, atau Mahasiswa). |  |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Pengguna memasukkan kredensial yang salah. |  |
|  |  | 2. Sistem menampilkan pesan error, meminta pengguna untuk mencoba lagi. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem mengalami kegagalan dalam memverifikasi kredensial. |
|  |  | 2. Sistem menampilkan pesan error umum seperti "Tidak dapat login saat ini." |
| *Post Condition* | Pengguna berhasil login dan diarahkan ke bagian yang sesuai (Admin, Dosen, atau Mahasiswa). | |

### 3.2.2 Use Case Scenario Mengelola Akun

Tabel 1 2 Use Case Scenario Mengelola Akun

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-2 | |
| *Use Case Name* | Mengelola Akun | |
| *Use Case Description* | Admin dapat mengelola akun pengguna, termasuk menambah, mengubah, dan menghapus akun. | |
| *Actor* | Admin | |
| *Precondition* | Admin sudah login dan berada di halaman pengelolaan akun. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Admin memilih opsi untuk mengelola akun. |  |
|  | 2. Admin memilih untuk menambah, mengubah, atau menghapus akun. |  |
|  |  | 3.Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 4. Admin memasukkan data yang diperlukan (misalnya, nama, email, peran, dll.). |  |
|  |  | 5. Sistem memvalidasi data yang dimasukkan |
|  |  | 6. Sistem mengupdate database dengan data baru atau menghapus akun sesuai permintaan. |
|  |  | 7. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Admin memasukkan data yang tidak valid. | - |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui database. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal memperbarui akun, coba lagi nanti." |
| *Post Condition* | Akun pengguna berhasil diperbarui atau dihapus sesuai dengan permintaan admin. | |

### 3.2.3 Use Case Scenario Publikasi Jadwal

Tabel 1 3 Use Case Scenario Publikasi Jadwal

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC- 3 | |
| *Use Case Name* | *Publikasi Jadwal* | |
| *Use Case Description* | Admin mempublikasikan jadwal perkuliahan setelah disetujui atau disusun. | |
| *Actor* | Admin | |
| *Precondition* | Admin telah login dan memiliki jadwal yang siap untuk dipublikasikan. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1.Admin memilih opsi untuk mempublikasikan jadwal. |  |
|  |  | 2. Sistem menampilkan jadwal yang telah disusun atau disetujui. |
|  | 3. Admin memilih jadwal yang akan dipublikasikan. |  |
|  | 4. Admin mengklik tombol "Publikasikan." |  |
|  |  | 5. Sistem memproses dan mempublikasikan jadwal yang dipilih. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa jadwal berhasil dipublikasikan. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Admin memilih jadwal yang belum lengkap atau ada kesalahan data. | - |
|  |  | 2. Sistem menampilkan pesan error dan meminta admin untuk memperbaiki jadwal. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal mempublikasikan jadwal. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal mempublikasikan jadwal, coba lagi nanti." |
| *Post Condition* | Jadwal berhasil dipublikasikan dan tersedia untuk diakses oleh dosen dan mahasiswa. | |

### 3.2.4 Use Case Scenario Kelola Ruang Kelas

Tabel 1 4 Use Case Scenario Kelola Ruang Kelas

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-4 | |
| *Use Case Name* | *Kelola Ruang Kelas* | |
| *Use Case Description* | Admin mengelola ruang kelas yang digunakan untuk perkuliahan, termasuk menambah, mengubah, dan menghapus ruang kelas. | |
| *Actor* | Admin | |
| *Precondition* | Admin telah login dan berada di halaman pengelolaan ruang kelas. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Admin memilih opsi untuk mengelola ruang kelas. |  |
|  | 2. . Admin memilih untuk menambah, mengubah, atau menghapus ruang kelas. |  |
|  |  | 3. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 4. Admin memasukkan data ruang kelas (misalnya, nama ruang, kapasitas, lokasi, dll.). |  |
|  |  | 5. Sistem memvalidasi data yang dimasukkan. |
|  |  | 6. Sistem mengupdate database dengan data ruang kelas yang baru atau menghapus ruang kelas sesuai permintaan. |
|  |  | 7. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Admin memasukkan data ruang kelas yang tidak valid. | - |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui data ruang kelas. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal memperbarui ruang kelas, coba lagi nanti." |
| *Post Condition* | Ruang kelas berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

### 3.2.5 Use Case Scenario Kelola Dosen

Tabel 1 5 Use Case Scenario Kelola Dosen

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-5 | |
| *Use Case Name* | *Kelola Dosen* | |
| *Use Case Description* | Admin mengelola data dosen, termasuk menambah, mengubah, dan menghapus informasi dosen. | |
| *Actor* | Admin | |
| *Precondition* | Admin telah login dan berada di halaman pengelolaan dosen. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Admin memilih opsi untuk mengelola data dosen. |  |
|  | 2. Admin memilih untuk menambah, mengubah, atau menghapus data dosen. |  |
|  |  | 3. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 4. Admin memasukkan data dosen (misalnya, nama, email, mata kuliah yang diampu, dll.). |  |
|  |  | 5. Sistem memvalidasi data yang dimasukkan. |
|  |  | 6. Sistem mengupdate database dengan data dosen yang baru atau menghapus data dosen sesuai permintaan. |
|  |  | 7. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Admin memasukkan data dosen yang tidak valid. | - |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1.Sistem gagal memperbarui data dosen. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal memperbarui data dosen, coba lagi nanti." |
| *Post Condition* | Data dosen berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

### 3.2.6 Use Case Scenario Kelola Asisten Akademik

Tabel 1 6 Use Case Scenario Kelola Asisten Akademik

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-6 | |
| *Use Case Name* | *Kelola Asisten Akademik* | |
| *Use Case Description* | Admin mengelola data asisten akademik, termasuk menambah, mengubah, dan menghapus data asisten akademik. | |
| *Actor* | Admin | |
| *Precondition* | Actor sudah login ke sistem | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Admin memilih opsi untuk mengelola asisten akademik. |  |
|  | 2. Admin memilih untuk menambah, mengubah, atau menghapus data asisten akademik. |  |
|  |  | 3. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 4. Admin memasukkan data asisten akademik (misalnya, nama, mata kuliah yang dibantu, jadwal kerja, dll.). |  |
|  |  | 5. Sistem memvalidasi data yang dimasukkan. |
|  |  | 6. Sistem mengupdate database dengan data asisten akademik yang baru atau menghapus data sesuai permintaan. |
|  |  | 7. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Admin memasukkan data asisten akademik yang tidak valid. | - |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui data asisten akademik. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal memperbarui data asisten akademik. |
| *Post Condition* | Data asisten akademik berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

### 3.2.7 Use Case Scenario Melihat Jadwal Perkuliahan

Tabel 1 7 Use Case Scenario Melihat Jadwal Perkuliahan

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-7 | |
| *Use Case Name* | *Melihat Jadwal Perkuliahan* | |
| *Use Case Description* | Dosen dan mahasiswa dapat melihat jadwal perkuliahan yang telah dipublikasikan oleh admin. | |
| *Actor* | Dosen, Mahasiswa | |
| *Precondition* | Tidak ada kondisi login yang diperlukan. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Dosen atau mahasiswa mengakses halaman jadwal perkuliahan. |  |
|  |  | 2. Sistem menampilkan daftar jadwal perkuliahan yang relevan berdasarkan kategori (misalnya, semua jadwal atau jadwal tertentu). |
|  | 3. Dosen atau mahasiswa memilih jadwal yang ingin dilihat lebih detail. |  |
|  |  | 4. Sistem menampilkan informasi lengkap jadwal, termasuk waktu, ruang, dan mata kuliah yang relevan. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Dosen atau mahasiswa tidak melihat jadwal yang sesuai. | - |
|  |  | 2. Sistem menampilkan pesan error seperti "Jadwal tidak tersedia untuk anda." |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menampilkan jadwal. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal mengambil jadwal, coba lagi nanti." |
| *Post Condition* | Dosen atau mahasiswa berhasil melihat jadwal perkuliahan yang relevan. | |

### 3.2.8 Use Scenario Kelola Ticketing

Tabel 1 8 Use Scenario Kelola Ticketing

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-8 | |
| *Use Case Name* | *Kelola Ticketing* | |
| *Use CaseDescription* | Dosen dan mahasiswa dapat mengelola tiket untuk permasalahan atau pertanyaan terkait administrasi akademik. | |
| *Actor* | Dosen, Mahasiswa | |
| *Precondition* | Dosen atau mahasiswa telah login dan berada di halaman ticketing. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Dosen atau mahasiswa memilih opsi untuk membuat tiket baru. |  |
|  |  | 2. Sistem menampilkan formulir untuk membuat tiket (misalnya, kategori masalah, deskripsi, dll.). |
|  | 3. Dosen atau mahasiswa mengisi formulir dan mengirimkan tiket. |  |
|  |  | 4. Sistem memproses tiket dan memberikan nomor tiket kepada pengguna. |
|  |  | 5. Sistem mengirimkan konfirmasi pembuatan tiket kepada pengguna. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Dosen atau mahasiswa memasukkan data yang tidak lengkap atau salah. |  |
|  |  | 2. Sistem menampilkan pesan error dan meminta pengguna untuk memperbaiki data. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memproses tiket. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal membuat tiket". |
| *Post Condition* | Tiket berhasil dibuat dan pengguna dapat melacak statusnya melalui sistem. | |

### 3.2.9 Use Case Scenario Memasukkan Mata Kuliah

Tabel 1 9 Use Case Scenario Memasukkan Mata Kuliah

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-9 | |
| *Use Case Name* | *Memasukkan Mata Kuliah* | |
| *Use Case Description* | Biro Administrasi dan Akademik memasukkan data mata kuliah ke dalam sistem untuk keperluan perkuliahan. | |
| *Actor* | Biro Administrasi dan Akademik | |
| *Precondition* | Biro Administrasi dan Akademik telah login dan berada di halaman pengelolaan mata kuliah. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memilih opsi untuk memasukkan mata kuliah. |  |
|  |  | 2. Sistem menampilkan formulir untuk memasukkan data mata kuliah. |
|  | 3. Biro Administrasi dan Akademik memasukkan data mata kuliah (nama mata kuliah, kode mata kuliah, dsb.). |  |
|  |  | 4. Sistem memvalidasi data yang dimasukkan. |
|  |  | 5. Sistem menyimpan data mata kuliah ke dalam database. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa mata kuliah telah berhasil dimasukkan. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memasukkan data yang tidak lengkap atau tidak valid. |  |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data mata kuliah. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal menyimpan mata kuliah |
| *Post Condition* | Mata kuliah berhasil dimasukkan dan tersedia di sistem untuk perkuliahan. | |

### 3.2.10 Use Case Scenarion Memasukkan Ruang Kelas

Tabel 1 10 Use Case Scenario Memasukkan Ruang Kelas

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-10 | |
| *Use Case Name* | *Memasukkan Ruang Kelas* | |
| *Use Case Description* | Biro Administrasi dan Akademik memasukkan data ruang kelas yang digunakan dalam perkuliahan. | |
| *Actor* | Biro Administrasi dan Akademik | |
| *Precondition* | Biro Administrasi dan Akademik telah login dan berada di halaman pengelolaan ruang kelas. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memilih opsi untuk memasukkan ruang kelas. |  |
|  |  | 2. Sistem menampilkan formulir untuk memasukkan data ruang kelas (nama ruang, kapasitas, dll.). |
|  | 3. Biro Administrasi dan Akademik memasukkan data ruang kelas. |  |
|  |  | 4. Sistem memvalidasi data yang dimasukkan. |
|  |  | 5. Sistem menyimpan data ruang kelas ke dalam database. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa ruang kelas telah berhasil dimasukkan. |
| *Alternate Flo of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memasukkan data yang tidak lengkap atau tidak valid. |  |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data ruang kelas. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal menyimpan ruang kelas. |
| *Post Condition* | Ruang kelas berhasil dimasukkan dan tersedia di sistem untuk penjadwalan. | |

### 3.2.11 Use Case Scenario Memasukkan Inisial Dosen

Tabel 1 11 Use Case Scenario Memasukkan Inisial Dosen

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-11 | |
| *Use Case Name* | *Memasukkan Inisial Dosen* | |
| *Use Case Description* | Biro Administrasi dan Akademik memasukkan inisial dosen untuk keperluan administrasi pengajaran. | |
| *Actor* | Biro Administrasi dan Akademik | |
| *Precondition* | Biro Administrasi dan Akademik telah login dan berada di halaman pengelolaan dosen. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memilih opsi untuk memasukkan inisial dosen. |  |
|  |  | 2. Sistem menampilkan formulir untuk memasukkan inisial dosen.. |
|  | 3. Biro Administrasi dan Akademik memasukkan inisial dosen yang diperlukan. |  |
|  |  | 4. Sistem memvalidasi data yang dimasukkan. |
|  |  | 5. Sistem menyimpan inisial dosen ke dalam database. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa inisial dosen telah berhasil dimasukkan. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memasukkan inisial dosen yang tidak valid. |  |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data inisial dosen. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal menyimpan inisial dosen |
| *Post Condition* | Inisial dosen berhasil dimasukkan dan dapat digunakan untuk administrasi dosen. | |

### 3.2.12. Use Case Scenario Memasukkan Program Studi

Tabel 1 12 Memasukkan Program Studi

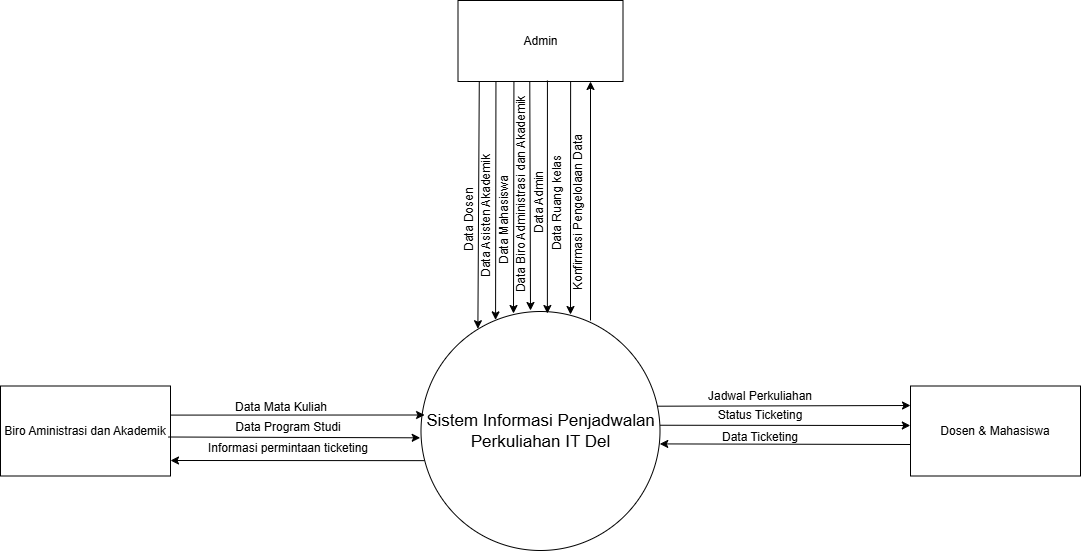
|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-12 | |
| *Use Case Name* | *Memasukkan Program Studi* | |
| *Use Case Description* | Biro Administrasi dan Akademik memasukkan data program studi yang ada di institusi. | |
| *Actor* | Biro Administrasi dan Akademik | |
| *Precondition* | Biro Administrasi dan Akademik telah login dan berada di halaman pengelolaan program studi. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memilih opsi untuk memasukkan program studi. |  |
|  |  | 2. Sistem menampilkan formulir untuk memasukkan data program studi (nama program studi, kode, dll.). |
|  | 3. Biro Administrasi dan Akademik memasukkan data program studi. |  |
|  |  | 4. Sistem memvalidasi data yang dimasukkan. |
|  |  | 5. Sistem menyimpan data program studi ke dalam database.. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa program studi telah berhasil dimasukkan. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memasukkan data program studi yang tidak valid. |  |
|  |  | 2. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data program studi. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal menyimpan program studi. |
| *Post Condition* | Program studi berhasil dimasukkan dan tersedia dalam sistem untuk administrasi akademik. | |

### 3.2.13 Use Case Scenario Evaluasi Ticketing

Tabel 1 13 Use Case Scenario Evaluasi Ticketing

|  |  |  |
| --- | --- | --- |
| *Use Case ID* | UC-13 | |
| *Use Case Name* | *Evaluasi Ticketing* | |
| *Use Case Description* | Biro Administrasi dan Akademik mengevaluasi tiket yang telah diajukan untuk memastikan masalah diselesaikan atau tidak, setelah tiket dikelola terlebih dahulu. | |
| *Actor* | Biro Administrasi dan Akademik | |
| *Precondition* | Biro Administrasi dan Akademik telah login dan telah menerima kelola tiket  dari dosen dan mahasiswa yang membutuhkan evaluasi. | |
| *Primary Flow of Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memilih opsi untuk mengevaluasi tiket.. |  |
|  |  | 2. Sistem menampilkan daftar tiket yang telah dikelola dan perlu evaluasi. |
|  | 3. Biro Administrasi dan Akademik memilih tiket yang akan dievaluasi. |  |
|  | 4. Biro Administrasi dan Akademik memeriksa detail tiket dan memberikan status (diselesaikan atau ditindaklanjuti). |  |
|  |  | 5. Sistem memperbarui status tiket sesuai dengan keputusan Biro Administrasi dan Akademik. |
|  |  | 6. Sistem menampilkan pesan konfirmasi bahwa tiket telah dievaluasi. |
| *Alternate Flow of Events* | User Action | System Response |
|  | 1. Tidak ada tiket yang tersedia untuk evaluasi. |  |
|  |  | 2. Sistem menampilkan pesan error seperti "Tidak ada tiket yang perlu dievaluasi." |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui status tiket. |
|  |  | 2. Sistem menampilkan pesan error seperti "Gagal memperbarui status tiket, |
| *Post Condition* | Status tiket berhasil diperbarui dan pengguna yang mengajukan tiket menerima pembaruan status. | |

## 3.3 Context Diagram



Gambar 6 Context Diagram

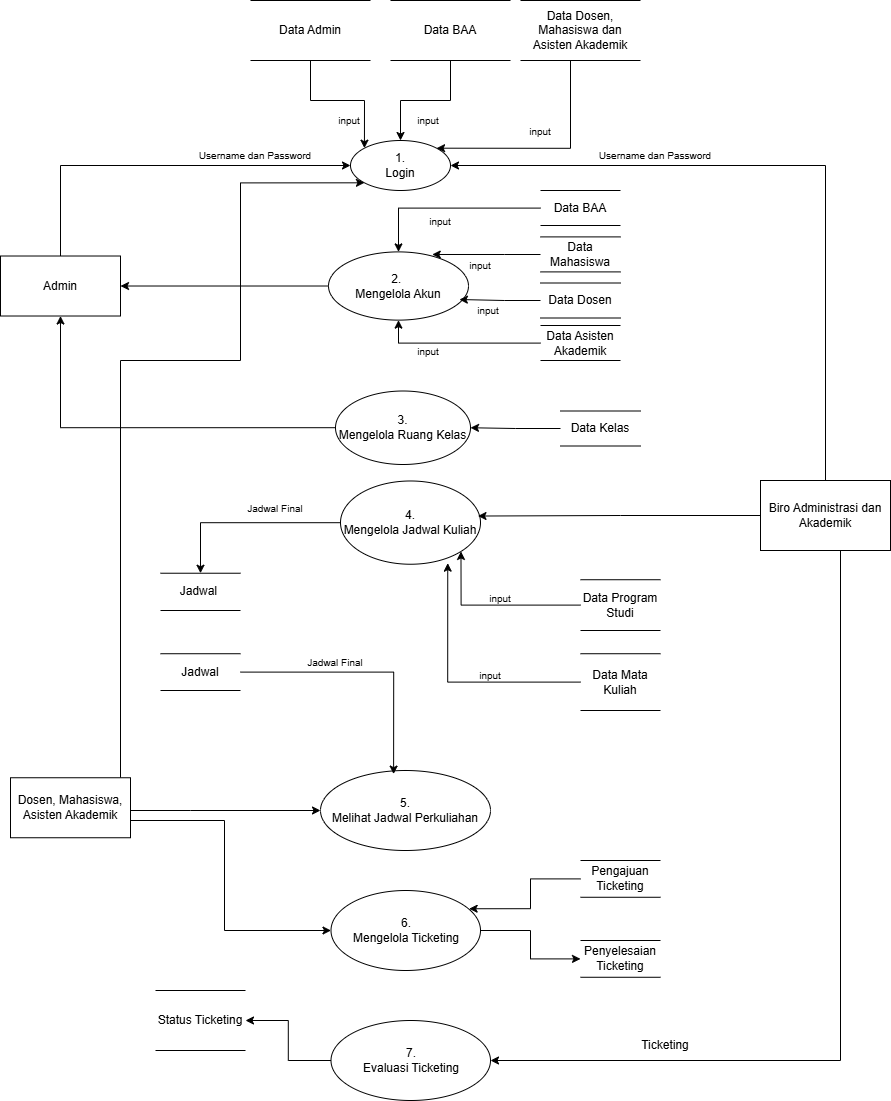
Pada *context* diagram dari Sistem Informasi Penjadwalan Perkuliahan IT Del, yang menggambarkan hubungan antara sistem dan entitas eksternal yang berinteraksi dengannya. Dalam diagram ini, terdapat tiga entitas utama, yaitu Admin, Biro Administrasi dan Akademik (BAA), serta Dosen & Mahasiswa, yang masing-masing memiliki peran penting dalam pengelolaan data dan informasi akademik.

Admin bertanggung jawab atas pengelolaan data dalam sistem, termasuk data dosen, asisten akademik, mahasiswa, administrasi akademik, serta data ruang kelas. Setiap perubahan yang dilakukan oleh admin akan dikonfirmasi kembali oleh sistem sebagai bentuk validasi pengelolaan data. Selain itu, Biro Administrasi dan Akademik (BAA) berperan dalam memberikan informasi akademik, seperti data mata kuliah dan data program studi, yang akan digunakan dalam proses penjadwalan perkuliahan. BAA juga menerima informasi terkait permintaan *ticketing*, yang memungkinkan mereka menangani berbagai permasalahan akademik yang diajukan melalui sistem.

Sementara itu, Dosen & Mahasiswa menggunakan sistem ini untuk melihat jadwal perkuliahan yang telah ditentukan. Mereka juga memiliki akses ke sistem ticketing, di mana mereka dapat melihat status *ticketing* yang diajukan serta mengajukan permintaan baru terkait masalah akademik atau jadwal perkuliahan yang perlu diperbaiki.

Sistem ini berfungsi sebagai pusat informasi yang mengelola data akademik, memproses penjadwalan perkuliahan, serta menangani keluhan atau perubahan yang diajukan melalui mekanisme *ticketing*. Secara keseluruhan, diagram ini menunjukkan bagaimana Sistem Informasi Penjadwalan Perkuliahan IT Del berinteraksi dengan berbagai pihak untuk memastikan kelancaran administrasi akademik dan perkuliahan di institusi tersebut.

## 3.4 DFD Level 1



Gambar 7 DFD Level 1

Pada DFD Level 1 dari Sistem Informasi Penjadwalan Perkuliahan IT Del, yang merinci aliran data dan proses utama yang terjadi di dalam sistem. Proses dimulai dengan Login, di mana Admin atau pengguna lain memasukkan *username* dan *password* untuk mengakses sistem. Setelah berhasil masuk, Admin memiliki akses untuk mengelola akun, yang mencakup pengelolaan data BAA, mahasiswa, dosen, dan asisten akademik.

Selanjutnya, Admin dapat mengelola ruang kelas dengan memasukkan atau memperbarui data kelas yang digunakan dalam penjadwalan. Proses berikutnya adalah mengelola jadwal kuliah, di mana sistem menerima input berupa data program studi dan mata kuliah dari Biro Administrasi dan Akademik (BAA), lalu menghasilkan jadwal perkuliahan yang nantinya dapat diakses oleh dosen, mahasiswa, dan asisten akademik.

Dosen, mahasiswa, dan asisten akademik dapat melihat jadwal perkuliahan yang telah diproses oleh sistem. Jika terdapat kendala atau perubahan yang perlu dilakukan, mereka dapat menggunakan fitur *ticketing*, di mana mereka dapat mengajukan permintaan tiket terkait permasalahan akademik. Proses ini dikelola dalam mengelola *ticketing*, yang mencakup pengajuan dan penyelesaian tiket. Setelah itu, sistem akan melakukan evaluasi ticketing, di mana setiap tiket yang telah diproses akan dinilai dan diperiksa status penyelesaiannya. Secara keseluruhan, sistem ini dirancang untuk memastikan kelancaran pengelolaan akun, ruang kelas, jadwal kuliah, serta menangani keluhan dan kendala akademik melalui sistem ticketing, dengan koordinasi antara Admin, BAA, serta dosen dan mahasiswa.

## 3.5 Main Features

Bagian ini menjelaskan fungsi utama yang tersedia dalam Sistem Informasi Penjadwalan Perkuliahan IT Del. Subbab ini akan menguraikan deskripsi setiap fungsi, termasuk data yang digunakan sebagai input, proses yang terjadi di dalamnya, serta output yang dihasilkan.

### 3.5.1 [SyRS-FR-01] Fitur Registrasi

Fitur registrasi digunakan untuk membuat akun pengguna baru agar dapat mengakses sistem.

#### 3.5.1.1 Description of Function

Dalam fungsi ini, pengguna mengisi formulir registrasi yang disediakan oleh sistem untuk mendapatkan akun.

#### 3.5.1.2 Inputs

* Nama lengkap
* NIM/NIDN (untuk mahasiswa dan dosen)
* Program studi (jika pengguna adalah mahasiswa)
* Email institusi
* Kata sandi

#### 3.5.1.3 Processing

Proses registrasi dilakukan melalui tahapan sebagai berikut:

1. Pengguna mengakses halaman registrasi pada sistem.
2. Pengguna mengisi data yang diminta dalam formulir registrasi.
3. Sistem melakukan validasi terhadap data yang dimasukkan untuk memastikan kelengkapan dan keabsahannya.
4. Jika data valid, sistem akan menyimpan informasi pengguna ke dalam basis data dan membuat akun baru.
5. Setelah registrasi berhasil, pengguna dapat beralih ke halaman login untuk melakukan autentikasi dan mengakses sistem.

#### 3.5.1.4 Outputs

Hasil dari proses registrasi meliputi:

1. Akun pengguna yang telah dibuat dan tersimpan dalam basis data sistem.
2. Notifikasi bahwa registrasi telah berhasil dan pengguna dapat melanjutkan ke proses login.
3. Pengguna diarahkan ke halaman login untuk masuk ke dalam sistem.

### 3.5.2 [SyRS-FR-02] Fitur Login

Fitur login merupakan bagian dari sistem yang bertujuan untuk mengautentikasi pengguna sebelum diberikan akses ke layanan yang tersedia.

#### 3.5.2.1 Description of Function

Fitur ini memungkinkan pengguna yang telah terdaftar untuk masuk ke dalam sistem dengan memasukkan kredensial yang telah dibuat sebelumnya.

#### 3.5.2.2 Inputs

Data yang diperlukan dalam proses login mencakup:

* Alamat email institusi
* Kata sandi yang telah dibuat saat registrasi

#### 3.5.2.3 Processing

Proses login dilakukan melalui tahapan berikut:

1. Pengguna mengakses halaman login pada sistem.
2. Pengguna memasukkan alamat email dan kata sandi yang telah terdaftar.
3. Sistem melakukan verifikasi terhadap kredensial yang dimasukkan dengan mencocokkannya dengan data yang tersimpan dalam basis data.

* Jika kredensial sesuai, pengguna diberikan akses ke dalam sistem dan diarahkan ke halaman utama sesuai dengan peran yang dimiliki.
* Jika kredensial tidak sesuai, sistem menampilkan pesan kesalahan dan meminta pengguna untuk memasukkan ulang data yang benar.

#### 3.5.2.4 Outputs

Hasil dari proses login meliputi:

1. Akses pengguna ke dalam sistem apabila autentikasi berhasil.
2. Tampilan dashboard yang sesuai dengan peran pengguna (misalnya mahasiswa, dosen, atau administrator).
3. Jika autentikasi gagal, sistem menampilkan pesan kesalahan yang menginformasikan bahwa kredensial yang dimasukkan tidak valid.

### 3.5.3 [SyRS-FR-03] Fitur Mengelola Akun

Fitur ini digunakan oleh admin untuk mengelola akun pengguna, termasuk pembuatan, pengubahan, dan penghapusan akun dalam sistem.

#### 3.5.3.1 Description of Function

Fitur ini memungkinkan admin untuk menambah, memperbarui, atau menghapus akun pengguna sesuai dengan kebutuhan sistem.

#### 3.5.3.2 Inputs

* Data akun pengguna, termasuk nama, alamat email, dan peran (mahasiswa, dosen, atau admin).
* Perubahan informasi akun, jika diperlukan.

#### 3.5.3.3 Processing

* Admin mengakses fitur pengelolaan akun melalui dashboard.
* Admin dapat menambahkan akun baru dengan mengisi formulir yang tersedia.
* Admin dapat memperbarui informasi akun pengguna, seperti mengubah kata sandi atau peran pengguna.
* Admin dapat menghapus akun pengguna yang tidak lagi aktif.
* Sistem akan memperbarui basis data sesuai dengan perubahan yang dilakukan.

#### 3.5.3.4 Outputs

* Akun pengguna baru berhasil dibuat dan dapat digunakan untuk login.
* Perubahan informasi akun tersimpan dalam sistem.
* Akun yang dihapus tidak dapat lagi digunakan untuk mengakses sistem.

### 3.5.4 [SyRS-FR-04] Fitur Publikasi Jadwal

Fitur ini digunakan untuk mempublikasikan jadwal perkuliahan yang telah ditentukan oleh admin atau biro akademik.

#### 3.5.4.1 Description of Function

Fitur ini memungkinkan admin untuk mengunggah, memperbarui, dan mempublikasikan jadwal kuliah yang dapat diakses oleh mahasiswa dan dosen.

#### 3.5.4.2 Inputs

Masukan yang diperlukan dalam fitur publikasi jadwal meliputi:

* Data mata kuliah (kode, nama, sks)
* Jadwal kelas (hari, jam, ruang)
* Dosen pengampu
* Program studi terkait

#### 3.5.4.3 Processing

Proses publikasi jadwal dilakukan melalui tahapan berikut:

1. Admin mengakses fitur publikasi jadwal dalam sistem.
2. Admin mengunggah atau memperbarui jadwal perkuliahan.
3. Sistem menyimpan dan memverifikasi kelengkapan data yang diunggah.
4. Jika data valid, sistem memperbarui jadwal yang dapat diakses oleh mahasiswa dan dosen.
5. Jika ada kesalahan atau jadwal bentrok, sistem akan memberikan peringatan kepada admin untuk melakukan revisi.

#### 3.5.4.4 Outputs

* Jadwal perkuliahan yang dapat dilihat oleh mahasiswa dan dosen.

### 3.5.5 [SyRS-FR-05] Fitur Melihat Jadwal Perkuliahan

Fitur ini memungkinkan mahasiswa dan dosen untuk melihat jadwal perkuliahan yang telah dipublikasikan.

#### 3.5.5.1 Description of Function

Fitur ini memberikan akses kepada pengguna untuk melihat jadwal perkuliahan mereka berdasarkan program studi atau kelas masing-masing.

#### 3.5.5.2 Inputs

* Permintaan pengguna untuk melihat jadwal berdasarkan kelas atau dosen tertentu.

#### 3.5.5.3 Processing

* Pengguna mengakses fitur melalui dashboard.
* Sistem mengambil data jadwal dari basis data.
* Sistem menampilkan jadwal perkuliahan sesuai dengan peran pengguna (dosen atau mahasiswa).

#### 3.5.5.4 Outputs

* Tampilan jadwal perkuliahan yang sesuai dengan permintaan pengguna.

### 3.5.6 [SyRS-FR-06] Fitur Ticketing untuk Perubahan Jadwal

Fitur ini memungkinkan dosen dan mahasiswa untuk mengajukan permintaan perubahan jadwal perkuliahan jika ada kendala, seperti bentrok jadwal, perubahan kelas, atau kendala teknis lainnya.

#### 3.5.6.1 Description of Function

Fitur ticketing ini digunakan untuk mempermudah komunikasi antara dosen, mahasiswa, dan admin dalam menangani permasalahan terkait jadwal perkuliahan.

#### 3.5.6.2 Inputs

Masukan yang diperlukan dalam fitur ini meliputi:

* Identitas pengguna (dosen atau mahasiswa)
* Mata kuliah terkait
* Jenis permintaan (perubahan jadwal, laporan bentrok, atau kendala lain)
* Deskripsi masalah atau permintaan perubahan
* Bukti pendukung (opsional)

#### 3.5.6.3 Processing

1. Pengguna (dosen atau mahasiswa) mengakses fitur ticketing dan mengisi form permintaan perubahan jadwal.
2. Sistem mencatat dan mengirimkan permintaan ke admin untuk ditinjau.
3. Admin menerima dan memverifikasi permintaan berdasarkan kebijakan akademik dan ketersediaan jadwal.
4. Jika permintaan valid, admin memperbarui jadwal dan memberikan notifikasi kepada pengguna.
5. Jika permintaan tidak dapat diproses, admin memberikan alasan penolakan kepada pengguna.

#### 3.5.6.4 Outputs

Hasil dari fitur ticketing ini meliputi:

* Notifikasi status tiket (diterima, diproses, ditolak, atau disetujui).
* Pembaruan jadwal jika permintaan perubahan disetujui.
* Pesan penolakan jika permintaan tidak dapat dipenuhi

## 3.6 Users Characteristics

Dalam penggunaan Sistem Informasi Penjadwalan Perkuliahan IT Del, terdapat beberapa *user-group* yang dapat menggunakannya. Beberapa kelompok karakteristik user dapat di uraikan sebagai berikut.

### 3.6.1 User-Group-[Akademik]

Description of User : Pengguna yang bertanggung jawab atas pembuatan, pengelolaan, dan publikasi jadwal perkuliahan.

Role : BAA

Prerequisit : BAA harus memiliki akun yang terdaftar agar dapat login ke dalam sistem.

Task description : 1. Membuat dan mempublikasikan jadwal perkuliahan.

2. Mengelola data ruang kelas, dosen, dan mata kuliah.

3. Mengevaluasi permintaan perubahan jadwal via *ticketing*.

4. Memastikan tidak ada konflik jadwal (waktu, ruang, dosen).

### 3.6.2 User-Group-[Dosen]

Description of User : Pengguna yang mengampu mata kuliah dan membutuhkan akses ke jadwal mengajar

Role : Dosen

Prerequisit : Dosen harus memiliki akun yang terdaftar agar dapat login ke dalam sistem.

Task description : 1. Melihat Jadwal perkuliahan

2. Mengajukan permintaan perubahan jadwal via *ticketing*

### 3.6.3 User-Group-[Mahasiswa]

Description of User : Pengguna yang membutuhkan akses untuk melihat jadwal akademik.

Role : Mahasiswa

Prerequisit : Mahasiswa harus memiliki akun yang terdaftar agar dapat login ke dalam sistem.

Task description : 1. Melihat Jadwal perkuliahan

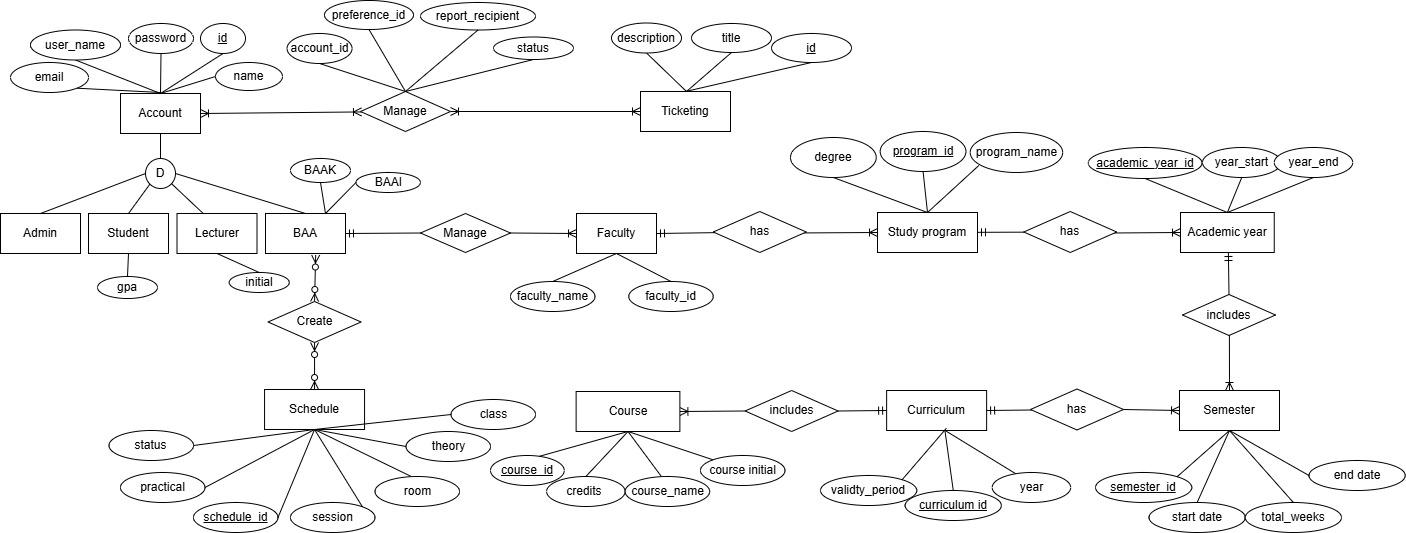
2. Mengajukan permintaan via *ticketing*

# 4 Kebutuhan Data dan Antarmuka

Bagian ini menjelaskan kebutuhan data dan antarmuka sistem yang akan direpresentasikan melalui diagram. Diagram yang digunakan adalah *Entity-Relationship Diagram* (ERD). Setiap atribut dalam diagram tersebut akan dijelaskan lebih lanjut dalam tabel data dictionary.

## 4.1 Data Requirement

### 4.1.1 ER-Diagram



Gambar 8 ER Diagram

## 4.2 Kebutuhan Antarmuka

Kebutuhan sistem ini dibuat dengan *Entity-Relationship Diagram* (ERD) yang akan menunjukkan entitas, atribut, dan hubungan antar entitas dalam sistem penjadwalan perkuliahan. Setiap entitas dalam diagram merepresentasikan komponen utama sistem, yaitu jadwal (*schedule*), mata kuliah (*course*), ruang kelas (*room*), dosen (*lecturer*), dan mahasiswa (*student*).

### 4.2.1 External Interface

Untuk memastikan Sistem Informasi Penjadwalan Perkuliahan dapat berfungsi secara optimal, terdapat kebutuhan data eksternal yang harus dipenuhi. Sistem ini akan berintegrasi dengan *Campus Information System* *(CIS)* melalui API untuk mendapatkan data yang diperlukan, seperti:

* Data mata kuliah (*course*): Informasi tentang kode mata kuliah, nama mata kuliah, dan jumlah SKS.
* Data dosen (*lecturer*): Informasi identitas dosen yang mengajar, termasuk nama dan jadwal mengajar.
* Data mahasiswa (*student*): Informasi mahasiswa yang terdaftar dalam suatu mata kuliah.
* Data ruang kelas (*room*): Informasi tentang ruang kuliah yang tersedia, kapasitas ruangan, dan fasilitasnya.
* Data jadwal perkuliahan (*schedule*): Informasi jadwal yang sudah dibuat, termasuk waktu, mata kuliah, dosen, dan ruang kelas.

### 4.2.2 User Interface

Antarmuka pengguna dalam Sistem Informasi Penjadwalan Perkuliahan berperan sebagai media interaksi antara sistem dengan pengguna, seperti staf administrasi (BAAK), dosen, dan mahasiswa. Sistem ini akan menyediakan tampilan yang intuitif dan mudah digunakan agar pengguna dapat mengakses serta mengelola informasi penjadwalan perkuliahan secara efisien.

Beberapa kebutuhan antarmuka pengguna yang diperlukan dalam sistem ini dapat dilihat pada Tabel 2.1.

Tabel 2 1 User Interface

|  |  |  |
| --- | --- | --- |
| No | User Interface | Fungsi |
|  | *Keyboard* | Memasukkan data jadwal, mata kuliah, dosen, mahasiswa, dan ruang kelas ke dalam sistem. |
|  | *Mouse* | Navigasi dan pengelolaan jadwal dalam sistem. |
|  | Monitor | Menampilkan informasi jadwal perkuliahan, daftar mata kuliah, serta laporan sistem. |
|  | Internet/LAN | Memungkinkan integrasi dengan *Campus Information System (CIS)* melalui API untuk pembaruan data secara real-time. |
|  | Hard Drive/SSD | Menyimpan data sistem secara lokal untuk backup atau penyimpanan offline. |
|  | Smartphone/Tablet | Memungkinkan akses jadwal bagi dosen dan mahasiswa melalui aplikasi atau browser. |

### 4.2.3 Hardware Interface

Kebutuhan antarmuka perangkat keras (hardware interface) merupakan kebutuhan yang digunakan untuk berinteraksi dengan website Sistem Informasi Penjadwalan Perkuliahan IT Del. Adapun yang menjadi kebutuhan perangkat keras yang dibutuhkan dapat dilihat pada tabel 2.2.

Tabel 2 2 Hardware Interface

|  |  |  |
| --- | --- | --- |
| No | User Interface | Fungsi |
|  | *Hard Drive/SSD* | Menyimpan data sistem secara lokal untuk backup atau penyimpanan offline. |
|  | Smartphone/Tablet | Memungkinkan akses jadwal bagi dosen dan mahasiswa melalui aplikasi atau browser. |

### 4.2.4 Software Interface

Kebutuhan antarmuka perangkat lunak (*software interface*) merupakan kebutuhan antarmuka perangkat lunak untuk membantu berjalannya sistem yang akan dibangun. Beberapa kebutuhan antarmuka perangkat lunak yang diperlukan adalah sebagai berikut:

* Web *Server*  : *Apache, Nginx*
* *Operating System*  : *Windows* 11
* *DBMS*  : *MySQL*
* *Browser*  : *Google Chrome, Mozilla Firefox*
* *Framework*  : Laravel 11
* *Text Editor* : *Visual Code Studio*
* *Process Modeler* : *Bizagi Modeler, Draw.io*
* *Design* : *Figma*
* *Testing & Development Tools* : *Katalon, Postman*
* *Documentation & Collaboration* : *Microsoft Word 2021, GitHub, Trello*
* *Database* : *MySQL*

# 5 Kebutuhan Lainnya

Pada bab ini akan dijelaskan mengenai kebutuhan-kebutuhan lain yang diperlukan untuk pengembangan Sistem Informasi Penjadwalan Perkuliahan IT Del.

## 5.1 System Performance Requirement

Kebutuhan non-fungsional dalam sistem penjadwalan kampus mencakup berbagai aspek yang memastikan sistem dapat beroperasi secara handal, aman, dan efisien. Karakteristik utama seperti *reliability*, *maintainability*, *safety*, *environmental*, dan aspek lainnya harus dipenuhi agar sistem dapat mendukung proses penjadwalan dengan optimal.

Kebutuhan ini mencakup ketersediaan sistem yang tinggi, keandalan dalam mengelola dan memproses data jadwal, keamanan akses pengguna, serta kemudahan dalam penggunaan. Dengan memenuhi standar performa yang baik, sistem akan memungkinkan admin, dosen, dan mahasiswa untuk mengakses serta mengelola jadwal secara mudah, cepat, dan akurat. Untuk keterangan kebutuhan non-fungsional dapat dilihat pada Tabel 3.1.

Tabel 3 1 System Performance Requierement

| **ID Features** | **Parameter** | **Requirement** |
| --- | --- | --- |
| SyRS-NF01 | *Availability* | Sistem harus dapat beroperasi 7 hari dalam seminggu dan 24 jam sehari untuk memastikan akses jadwal selalu tersedia bagi pengguna. |
| SyRS-NF02 | *Reliability* | Sistem harus memiliki mekanisme redundansi data dan *backup* otomatis untuk memastikan data jadwal tidak hilang atau rusak. |
| SyRS-NF03 | *Ergonomy* | Sistem harus memiliki antarmuka yang *user-friendly*, dengan fitur *drag-and-drop scheduling*, serta tampilan jadwal yang mudah dibaca dan dipahami. |
| SyRS-NF04 | *Portability* | Sistem harus dapat digunakan pada *web browser* *(Chrome, Firefox, Edge)* di *Windows, macOS*, serta perangkat *mobile* berbasis *Android* dan *iOS*. |
| SyRS-NF05 | *Memory* | Sistem penjadwalan harus dapat berjalan dengan efisien tanpa mengonsumsi memori berlebihan, dengan penggunaan sumber daya maksimal pada server. |
| SyRS-NF06 | *Response time* | Waktu respon sistem maksimal 3 detik untuk pemrosesan jadwal dan 5 detik untuk pencarian jadwal. |
| SyRS-NF07 | *Safety* | Sistem harus memiliki mekanisme audit log untuk mencatat setiap perubahan jadwal yang dilakukan oleh admin atau dosen. |
| SyRS-NF08 | *Security* | Sistem harus memiliki mekanisme login yang aman, mendukung autentikasi dua faktor (2FA) untuk admin, dan mengenkripsi data jadwal serta informasi pengguna. |

## 5.2 Enabling Requirement

Enabling requirement merupakan deskripsi dari kebutuhan yang diperlukan dalam proses pembangunan sebuah sistem. Adapun *Enabling Requirement* pada Sistem Informasi Penjadwalan Perkuliahan IT Del dapat dilihat pada table 3.2.

Tabel 3 2 Enabling Requirement

| **ID Features** | **Parameter** | **Requirement** |
| --- | --- | --- |
| SyRS-ER-01 | *Infrastruktur* | Sistem penjadwalan harus memiliki infrastruktur yang memadai, termasuk jaringan yang stabil, perangkat keras yang kompatibel, dan perangkat lunak pendukung agar sistem dapat berjalan dengan baik |
| SyRS-ER-02 | *Security* | Sistem harus memiliki mekanisme keamanan yang memadai, seperti autentikasi pengguna, enkripsi data, dan kontrol akses untuk mencegah perubahan jadwal oleh pihak yang tidak berwenang. |
| SyRS-ER-03 | *Database* | Sistem harus memiliki basis data yang dapat menyimpan informasi jadwal, dosen, mata kuliah, dan ruang kelas secara terstruktur agar dapat diakses dengan cepat dan akurat. |
| SyRS-ER-04 | *Training* | Selama proses pelatihan sistem, pengguna harus diberikan akses ke fitur simulasi penjadwalan untuk memahami cara kerja sistem sebelum digunakan secara penuh. |
| SyRS-ER-05 | *Deployment* | Saat proses *deployment*, sistem harus berjalan dengan baik dan menampilkan data penjadwalan yang telah diinputkan sesuai dengan aturan akademik yang berlaku. |
| SyRS-ER-06 | *Production* | Setelah diterapkan, sistem harus mampu menghasilkan jadwal kuliah yang optimal dan menghindari konflik jadwal antar mata kuliah, ruang kelas, dan dosen |
| SyRS-ER-07 | *Support* | Sistem harus memiliki fitur pemantauan dan dukungan teknis agar kendala dalam penjadwalan dapat segera diatasi tanpa mengganggu operasional akademik. |

## 5.3 Constraint Requirement

Website Sistem Informasi Penjadwalan Perkuliahan IT Del ini memiliki batasan khusus seperti:

1. Sistem harus kompatibel dengan infrastruktur IT Del yang ada, baik dari sisi *hardware* (server internal IT Del) maupun *software* (terintegrasi dengan sistem akademik IT Del).
2. Sistem harus mendukung fitur kalender untuk menampilkan jadwal secara terstruktur dengan informasi lengkap mengenai waktu, lokasi, dan aktivitas.
3. Sistem harus mampu menangani perubahan mendadak, termasuk pembatalan atau pemindahan jadwal kuliah.
4. Harus menyediakan fitur *Drag and Drop* untuk penyusunan jadwal, dengan mekanisme validasi otomatis untuk menghindari tabrakan jadwal.
5. Harus memastikan bahwa setiap jadwal yang dibuat tidak bentrok, dengan mekanisme validasi dan pemberian notifikasi jika ada konflik.
6. Sistem harus dapat mengambil data dari Sistem Akademik IT Del, termasuk informasi mengenai jumlah ruangan, kapasitas ruangan, prioritas pengguna ruangan, mata kuliah, jumlah pendaftar mata kuliah, jadwal teori dan praktikum.
7. Harus mempertimbangkan *Hard Constraint* (aturan yang tidak boleh dilanggar, seperti kapasitas ruangan dan jam operasional) dan *Soft Constraint* (aturan fleksibel yang diutamakan, seperti preferensi dosen).

## 5.4. SW Environment

Website Sistem Informasi Penjadwalan IT del ini dapat dikembangkan dengan spesifikasi kebutuhan yang dapat dilihat pada Tabel

### 5.4.1. Development Environment

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| *Server (Development)* | * *Processor: Intel Core* i5-1135G7 @ 2.40GHz atau i5-11300H @ 3.10GHz. * *RAM:* Minimal 8GB. * *Storage:* SSD 256GB atau lebih * *Network*: Standar jaringan lokal kampus * *Web Server: Apache* |
| *Operating System* | * *Windows 11* |
| *DBMS* | * *MySQL* |
| *Development Tools* | * *Framework: Laravel 11 (Backend).* * *IDE/Text Editor: Visual Studio Code.* * *Process Modeler: Bizagi Modeler, Draw.io.* * *Version Control: Git (GitHub/GitLab)* |

### 5.4.2. Operational Environment

Aplikasi *client-server* ini akan berfungsi dengan spesifikasi sebagai berikut:

* *Server: Processor Intel Core* i5-1135G7 @ 2.40GHz atau i5-11300H @ 3.10GHz, CPU minimal Intel Core i5 11th Gen, RAM minimal 8GB, storage SSD 256GB atau lebih, standar jaringan lokal kampus, dan menggunakan Apache sebagai web server.
* *Client:* PC/Laptop dengan minimal Intel Core i5 11th Gen, RAM 8GB, serta mendukung resolusi layar 1366x768 atau lebih tinggi. Akses melalui *browser web* seperti *Google Chrome, Microsoft Edge, atau Mozilla Firefox.*
* *OS: Windows* 10/11.

# 6. Traceability

Diisi dengan tabel yang berisi traceability dari hasil analisis. Gunanya untuk menilai apakah hasil analisis “runut” dan lojik. Untuk sementara, baru didefinisikan Data-store versus E-R. Anda boleh mendefinisikan subbab yang diperlukan untuk keperluan melakukan traceability yang ingin dicapai. Pembagian traceability yang diuraikan disini dipakai jika menggunakan *Metodologi Structure Analysis*. Jika menggunakan metodologi lain harus disesuaikan .

Tabel 4 1 Traceability

|  |  |  |
| --- | --- | --- |
| **Data Store** | **Entity** | **Relasi** |
| Data Mahasiswa | Mahasiswa | 1,M |
| Data Mata Kuliah | Mata Kuliah | 1,M |
| Data Dosen | Dosen | 1,M |
| Data Asisten Akademik | Asisten Akademik | 1,M |
| Data Ruangan | Ruangan | 1,M |
| Data Jadwal | Jadwal | 1,1 |
| Data Kelas Mahasiswa | Mahasiswa, Mata Kuliah | M,M |

Penjelasan :

Data Mahasiswa memiliki hubungan one-to-many (1,M) dengan mata kuliah, karena seorang mahasiswa dapat mengambil banyak mata kuliah, dan setiap mata kuliah dapat diikuti oleh banyak mahasiswa. Data Mata Kuliah juga memiliki relasi one-to-many (1,M) dengan dosen, sebab satu mata kuliah bisa diajarkan oleh lebih dari satu dosen dalam skema team teaching.

Selanjutnya, Data Dosen memiliki hubungan one-to-many (1,M) dengan mata kuliah, karena seorang dosen dapat mengajar beberapa mata kuliah dalam satu semester. Begitu juga dengan Data Asisten Akademik, yang memiliki relasi one-to-many (1,M) dengan kelas, sebab satu asisten akademik bisa membantu di berbagai kelas, sementara satu kelas dapat memiliki lebih dari satu asisten akademik.

Data Ruangan juga memiliki relasi one-to-many (1,M) dengan jadwal, karena satu ruangan bisa digunakan untuk berbagai jadwal kuliah pada waktu yang berbeda. Namun, satu jadwal hanya bisa memiliki satu ruangan tertentu yang telah ditentukan sebelumnya. Sementara itu, Data Jadwal memiliki relasi one-to-one (1,1) dengan jadwal kuliah yang sudah final, karena setiap jadwal hanya bisa digunakan untuk satu kombinasi spesifik dari mata kuliah, kelas, dosen, dan ruangan.

Terakhir, Data Kelas Mahasiswa memiliki hubungan many-to-many (M,M) dengan mahasiswa dan mata kuliah, karena seorang mahasiswa bisa tergabung dalam beberapa kelas berbeda, dan satu kelas dapat berisi banyak mahasiswa.

## Traceability Functional Requirement Summary

| **Features ID** | **Description** |
| --- | --- |
| SyRS-FR-01 | Pengelolaan Data Akademik: Sistem memungkinkan admin untuk menginput dan mengelola data mata kuliah, dosen, asisten akademik, mahasiswa, serta informasi akademik lainnya secara manual. Data ini digunakan sebagai dasar dalam proses penjadwalan perkuliahan. |
| SyRS-FR-02 | Pengelolaan Ruang Kuliah: Sistem dapat mengalokasikan ruang kuliah berdasarkan jumlah mahasiswa yang terdaftar dalam kelas. |
| SyRS-FR-03 | Autentikasi dan Otorisasi: Sistem mendukung *login* berbasis *username* dan *password* yang diberikan oleh admin, serta menerapkan *Role-Based Access Control* (RBAC) untuk memastikan hanya pengguna dengan hak akses tertentu yang dapat mengelola data akademik, penjadwalan, dan *tiketing.* |
| SyRS-FR-04 | Pencarian Jadwal Kuliah: Sistem memungkinkan pengguna untuk mencari jadwal berdasarkan mata kuliah, dosen, hari, ruangan, atau sesi dengan tampilan yang dapat disesuaikan dalam format mingguan. |
| SyRS-FR-05 | Manajemen *Ticketing* Akademik: Sistem memungkinkan dosen, mahasiswa, dan asisten akademik untuk mengajukan tiket permasalahan terkait jadwal perkuliahan, dengan evaluasi tiket oleh BAA dan statusnya kembali ke mahasiswa, dosen dan asisten akademik. |

**6.1 Data Store vc E-R**

**Mapping data store pada DFD dengan Entity – Relasi.**

|  |  |  |
| --- | --- | --- |
| **Data Store** | **Entity** | **Relasi** |
|  |  |  |

**6.2 Tracebility Functional Requirement Summary**

Requirement item ini mencerminkan semua hal yang harus dipenuhi, dan nantinya akan menjadi arahan untuk tahapan testing, karena pada dasarnya, semua requirement harus dapat ditest supaya dapat dibuktikan dipenuhi. Dibagi menjadi dua bagian: functional dan non functional.

|  |  |
| --- | --- |
| **Features ID** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**6.3 Tracebility lain (...)**

|  |  |  |
| --- | --- | --- |
| **SyRS** | **No. Proses** | **Keterangan** |
|  |  |  |

Catatan: Semua requirement dalam metodologi structure analysis harus diterjemahkan menjadi proses.

# LAMPIRAN

# Sejarah Versi

*Pada bagian ini, dijelaskan semua versi yang pernah di-deliver, dan ciri serta perubahannya.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versi** | **Ditulis Oleh** | **Tanggal** | **Disetujui Oleh** | **Tanggal** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Sejarah Perubahan

*Bagian ini memuat sejarah perubahan dokumen (no. versi terbaru dibandingkan versi sebelumnya).*

**No. dokumen :**

**No. versi :**

| **Halaman** | **Semula** | **Menjadi** | **Alasan perubahan** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**No. dokumen :**

**No. versi :**

| **Halaman** | **Semula** | **Menjadi** | **Alasan perubahan** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |