System Requirement Specification

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 6**](#_5oqf7pfs8w77)

[**1.1 Tujuan Penulisan Dokumen 6**](#)

[**1.2 Konvensi Dokumen 6**](#)

[**1.3 Dokumen Referensi 7**](#)

[**1.4 Ikhtisar Dokumen 7**](#)

[**2 Gambaran Umum Sistem 8**](#)

[**2.1 Gambaran Umum Sistem Saat Ini 8**](#)

[**2.1.1 Proses Bisnis Membuat Jadwal dan Merilis Jadwal [PB01] 8**](#)

[2.1.1.1 Waktu Layanan 9](#_lzb5exfblqvk)

[**2.2.2 Proses Bisnis Melakukan Ticketing dengan Preferensi User [PB02] 12**](#)

[**2.1.1.2. Waktu Layanan 12**](#)

[**2.1.3.1 Waktu Layanan 13**](#)

[**3. Kebutuhan Fungsional 14**](#)

[**3.1 Use Case Diagram 14**](#)

[3.2 Use Case Scenario 14](#_8rp2yagiy446)

[**3.3 Context Diagram 22**](#)

[3.4 DFD Level 1 22](#_tqgr603x96go)

[3.5 Main Features 22](#_1vyl73486i5n)

[**3.5.1 [SyRS-FR-01] Fitur Registrasi 22**](#)

[3.5.1.1 Description of Function 22](#_y8p9nqpee9ny)

[3.5.1.2 Inputs 22](#_z5xau4brak2x)

[**3.5.1.3 Processing 22**](#)

[**3.5.1.4 Outputs 23**](#)

[**3.5.2 [SyRS-FR-02] Fitur Login 23**](#)

[3.5.2.1 Description of Function 23](#_hailldwct2lf)

[3.5.2.2 Inputs 23](#_kg514erqgx09)

[**3.5.2.3 Processing 23**](#)

[**3.5.2.4 Outputs 23**](#)

[3.5.3 [SyRS-FR-03] Fitur Mengelola Akun 24](#_an6cehe4w9it)

[3.5.3.1 Description of Function 24](#_hi3y3sbz97fz)

[3.5.3.2 Inputs 24](#_3fabkfkavhl)

[3.5.3.3 Processing 24](#_840gw4ynlpbq)

[3.5.3.4 Outputs 24](#_c6ublz8sefq3)

[3.5.4 [SyRS-FR-04] Fitur Publikasi Jadwal 25](#_q0sbxra9q4bi)

[3.5.4.1 Description of Function 25](#_3ejhzqous51c)

[3.5.4.2 Inputs 25](#_9fxl1sod4hc7)

[3.5.4.3 Processing 25](#_pvfzkwih9ogl)

[3.5.4.4 Outputs 25](#_t3omhxxlo6yz)

[3.5.5 [SyRS-FR-05] Fitur Melihat Jadwal Perkuliahan 25](#_9ktq3owfv3ry)

[3.5.5.1 Description of Function 25](#_40ctxrpribbs)

[3.5.5.2 Inputs 25](#_huit5m97h7fl)

[3.5.5.3 Processing 26](#_ywbiyum00ktk)

[3.5.5.4 Outputs 26](#_oz2l28e7xc38)

[**3.6 Users Characteristics 26**](#)

[**3.6.1 User-Group-[XX\_] 26**](#)

[**3.6.2 User-Group-[XX\_] 26**](#)

[**4 Kebutuhan Data dan Antarmuka 28**](#_qipoc4v61fdm)

[**4.1 Data Requirement 28**](#)

[4.1.1 ER-Diagram 28](#_5iyl0kc2upsh)

[**4.2 Kebutuhan Antarmuka 28**](#)

[**4.2.1 External Interface 28**](#)

[**4.2.2 User Interface 29**](#)

[**4.2.3 Hardware Interface 29**](#)

[**4.2.4 Software Interface 30**](#)

[**5 Kebutuhan Lainnya 31**](#_hrkpr47dcpfa)

[**5.1 System Performance Requirement 31**](#)

[**5.2 Enabling Requirement. 32**](#)

[**5.3 Constraint Requirement 33**](#)

[**5.4. SW Environment 33**](#)

[**5.4.1. Development Environment 33**](#)

[**5.4.2. Operational Environment 33**](#)

[**6. Traceability 35**](#)

[**Traceability Functional Requirement Summary 35**](#)

[**LAMPIRAN 38**](#)

[**Sejarah Versi 39**](#)

[**Sejarah Perubahan 40**](#)

**DAFTAR TABEL**

**DAFTAR GAMBAR**

# 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

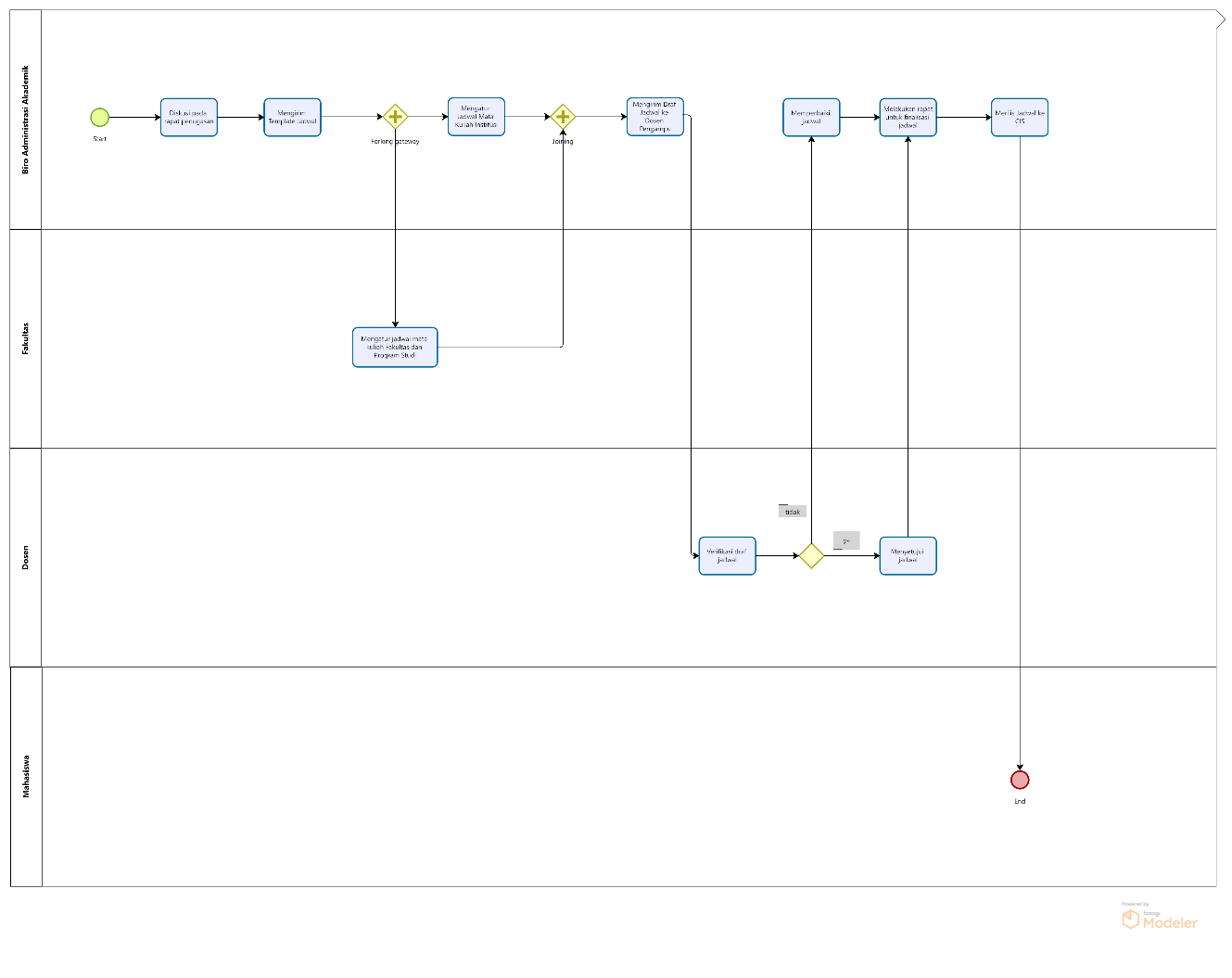
Pada bab ini dijelaskan deskripsi secara umum mengenai sistem yang sedang berjalan saat ini (*current system*) dan sistem yang ditargetkan (target system) dengan adanya Sistem Informasi Penjadwalan Perkuliahan IT Del. Bagian - bagian yang dijelaskan mencakup proses bisnis, prosedur dari setiap proses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



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

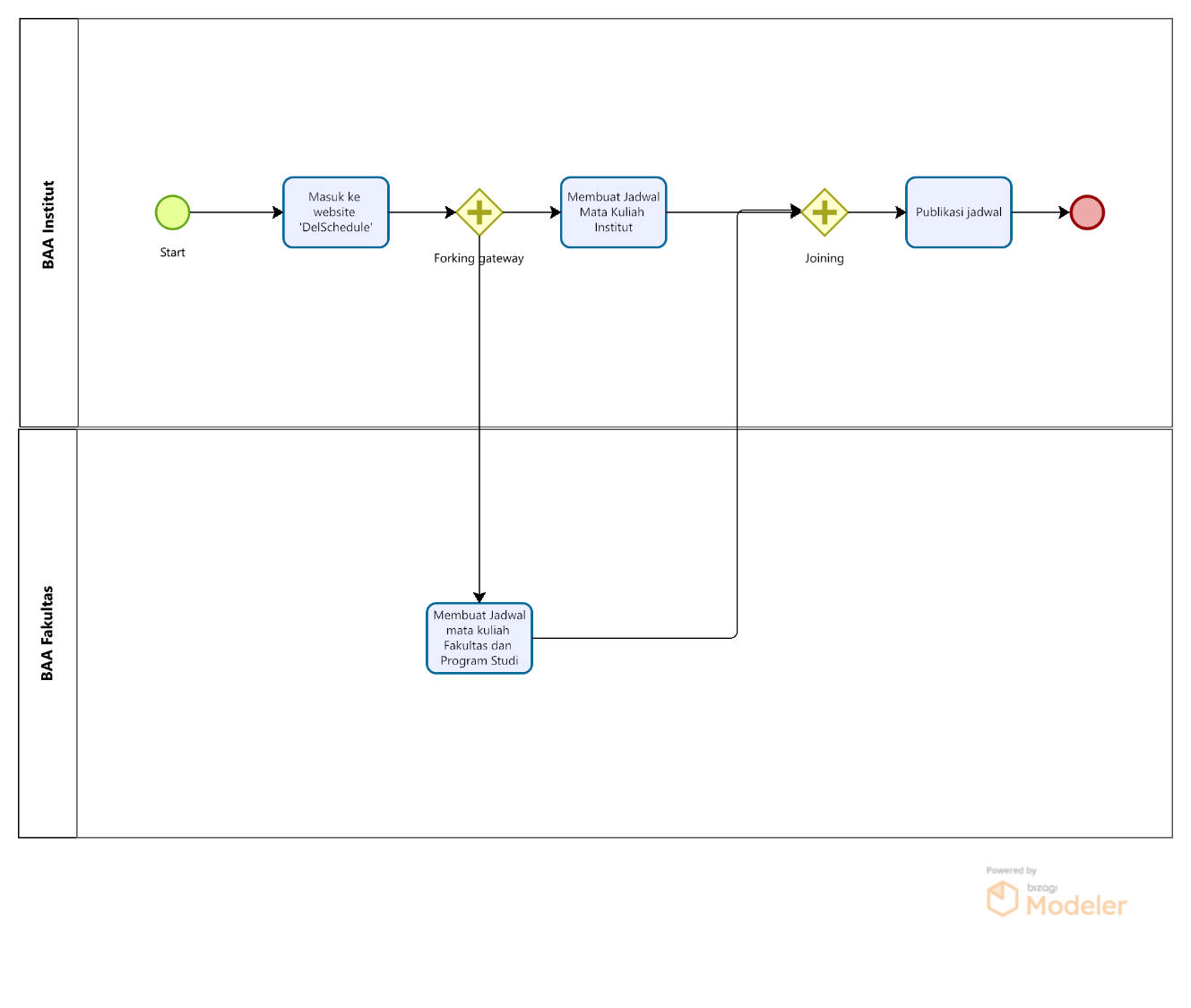
Waktu yang dibutuhkan untuk menyusun dan mengelola jadwal akademik secara manual di IT Del memakan waktu lebih dari 2 minggu setiap semester. Dalam rentang waktu tersebut sudah termasuk proses pengumpulan data jadwal, koordinasi dengan dosen dan staf akademik, penyusunan jadwal, pengecekan konflik jadwal, 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]**

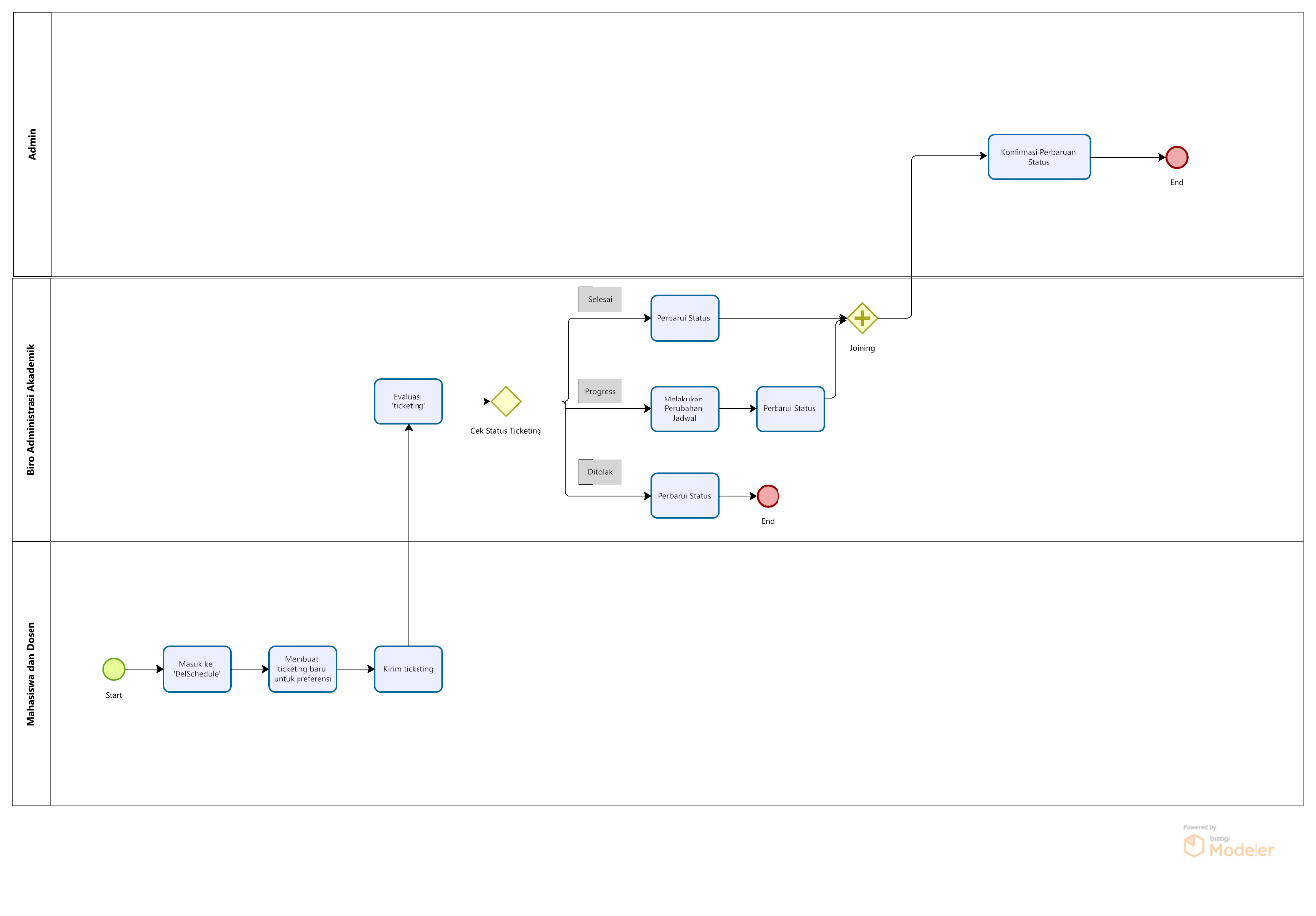
**

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]

**

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]**

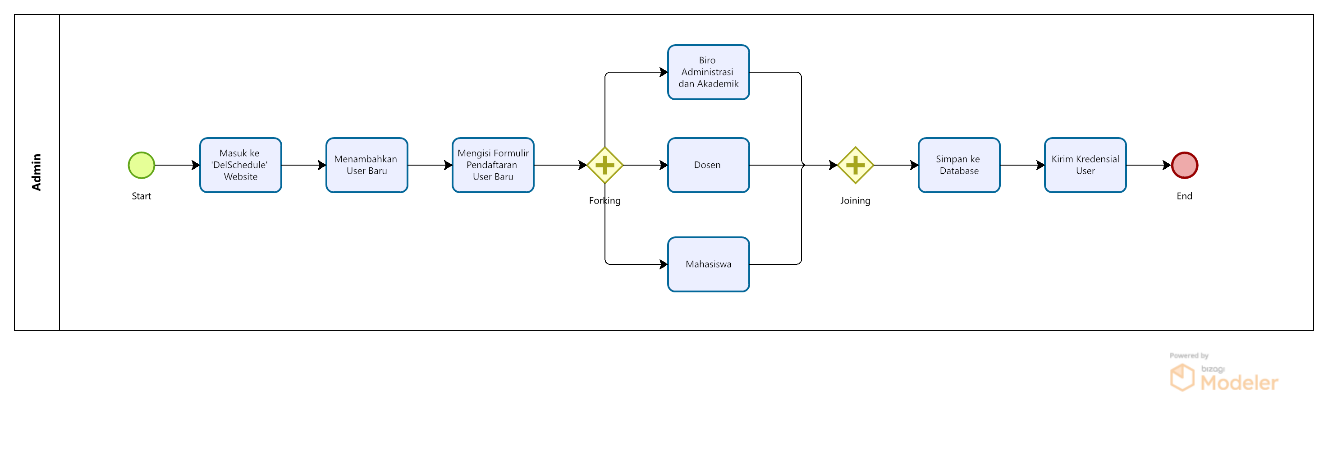


Diagram BPMN ini menggambarkan proses bisnis pengelolaan user pada sistem DelSchedule yang dijalankan sepenuhnya oleh Admin. Proses dimulai dengan Admin masuk ke website *DelSchedule*, dilanjutkan dengan menambahkan user baru ke dalam sistem. Setelah itu, Admin mengisi formulir pendaftaran dengan informasi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

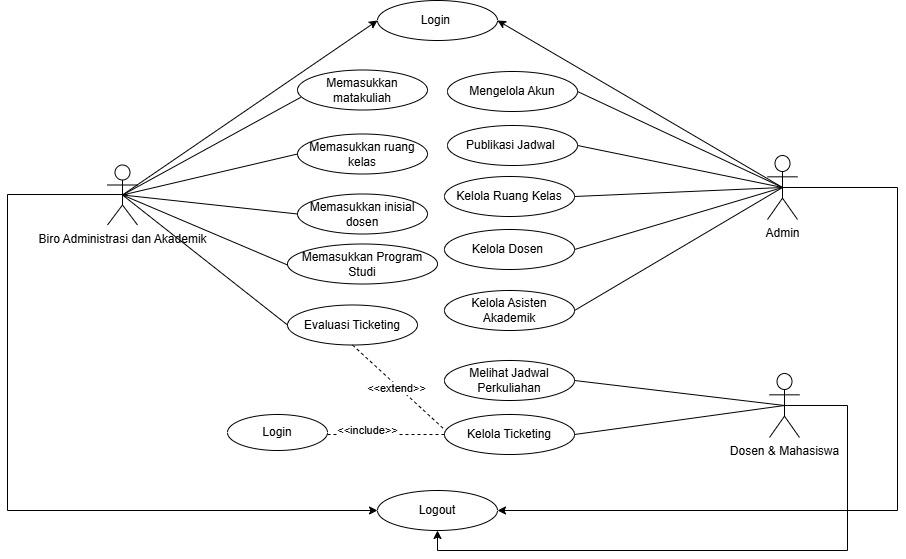
Pengisian formulir pendaftaran user merupakan tahap yang membutuhkan waktu lebih lama, sekitar 3-5 menit, karena melibatkan input data yang cukup detail. Pemrosesan berdasarkan kategori user memerlukan waktu tambahan 2-3 menit per user untuk memastikan atribut spesifik kategori terisi dengan benar.

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



## 3.2 Use Case Scenario

**Use Case Scenario 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. |  |
|  |  | 1. Sistem menampilkan pesan error, meminta pengguna untuk mencoba lagi. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem mengalami kegagalan dalam memverifikasi kredensial. |
|  |  | 1. 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). | |

**Use Case Scenario Manage Account**

| *Use Case ID* | UC-2 | |
| --- | --- | --- |
| *Use Case Name* | *Manage Account* | |
| *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. |  |
|  |  | 1. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 1. Admin memasukkan data yang diperlukan (misalnya, nama, email, peran, dll.). |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan |
|  |  | 1. Sistem mengupdate database dengan data baru atau menghapus akun sesuai permintaan. |
|  |  | 1. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of*  *Events* | User Action | System Response |
|  | 1. Admin memasukkan data yang tidak valid. | - |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui database. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal memperbarui akun, coba lagi nanti." |
| *Post Condition* | Akun pengguna berhasil diperbarui atau dihapus sesuai dengan permintaan admin. | |

**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. |
|  | 1. Admin memilih jadwal yang akan dipublikasikan. |  |
|  | 1. Admin mengklik tombol "Publikasikan." |  |
|  |  | 1. Sistem memproses dan mempublikasikan jadwal yang dipilih. |
|  |  | 1. 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. | - |
|  |  | 1. Sistem menampilkan pesan error dan meminta admin untuk memperbaiki jadwal. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal mempublikasikan jadwal. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal mempublikasikan jadwal, coba lagi nanti." |
| *Post Condition* | Jadwal berhasil dipublikasikan dan tersedia untuk diakses oleh dosen dan mahasiswa. | |

**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. |  |
|  |  | 1. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 1. Admin memasukkan data ruang kelas (misalnya, nama ruang, kapasitas, lokasi, dll.). |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem mengupdate database dengan data ruang kelas yang baru atau menghapus ruang kelas sesuai permintaan. |
|  |  | 1. 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. | - |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui data ruang kelas. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal memperbarui ruang kelas, coba lagi nanti." |
| *Post Condition* | Ruang kelas berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

**Usecase 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. |  |
|  | 1. Admin memilih untuk menambah, mengubah, atau menghapus data dosen. |  |
|  |  | 1. Sistem menampilkan formulir yang sesuai berdasarkan pilihan admin. |
|  | 1. Admin memasukkan data dosen (misalnya, nama, email, mata kuliah yang diampu, dll.). |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem mengupdate database dengan data dosen yang baru atau menghapus data dosen sesuai permintaan. |
|  |  | 1. Sistem menampilkan pesan konfirmasi bahwa perubahan telah berhasil. |
| *Alternate Flow of*  *Events* | User Action | System Response |
|  | 1. Admin memasukkan data dosen yang tidak valid. | - |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui data dosen. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal memperbarui data dosen, coba lagi nanti." |
| *Post Condition* | Data dosen berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

**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 telah login dan berada di halaman pengelolaan asisten akademik. | |
| *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. | - |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memperbarui data asisten akademik. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal memperbarui data asisten akademik. |
| *Post Condition* | Data asisten akademik berhasil ditambahkan, diubah, atau dihapus sesuai permintaan admin. | |

**Use Case Scenario Melihat Jadwal Perkuliahan**

| *Use Case ID* | | | UC-07 | |
| --- | --- | --- | --- | --- |
| *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 Events* | *Flow* | *of* | User Action | System Response |
|  |  |  | 1. Dosen atau mahasiswa mengakses halaman jadwal perkuliahan. |  |
|  |  |  |  | 1. Sistem menampilkan daftar jadwal perkuliahan yang relevan berdasarkan kategori (misalnya, semua jadwal atau jadwal tertentu). |
|  |  |  | 1. Dosen atau mahasiswa memilih jadwal yang ingin dilihat lebih detail. |  |
|  |  |  |  | 1. Sistem menampilkan informasi lengkap jadwal, termasuk waktu, ruang, dan mata kuliah yang relevan. |
| *Alternate Events* | *Flow* | *of* | User Action | System Response |
|  |  |  | 1. Dosen atau mahasiswa tidak melihat jadwal yang sesuai. |  |
|  |  |  |  | 1. Sistem menampilkan pesan error seperti "Jadwal tidak tersedia untuk anda." |
| *Error Flow of Events* | | | User Action | System Response |
|  | | |  | 1. Sistem gagal menampilkan jadwal. |
|  | | |  | 1. Sistem menampilkan pesan error seperti "Gagal mengambil jadwal, coba lagi nanti." |
| *Post Condition* | | | Dosen atau mahasiswa berhasil melihat jadwal perkuliahan yang relevan. | |

**Use Scenario Kelola Ticketing**

| *Use Case ID* | UC-8 | |
| --- | --- | --- |
| *Use Case Name* | *Kelola Ticketing* | |
| *Use Case Description* | 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. |  |
|  |  | 1. Sistem menampilkan formulir untuk membuat tiket (misalnya, kategori masalah, deskripsi, dll.). |
|  | 1. Dosen atau mahasiswa mengisi formulir dan mengirimkan tiket. |  |
|  |  | 1. Sistem memproses tiket dan memberikan nomor tiket kepada pengguna. |
|  |  | 1. 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. |  |
|  |  | 1. Sistem menampilkan pesan error dan meminta pengguna untuk memperbaiki data. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal memproses tiket. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal membuat tiket". |
| *Post Condition* | Tiket berhasil dibuat dan pengguna dapat melacak statusnya melalui sistem. | |

## 

**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. |  |
|  |  | 1. Sistem menampilkan formulir untuk memasukkan data mata kuliah. |
|  | 1. Biro Administrasi dan Akademik memasukkan data mata kuliah (nama mata kuliah, kode mata kuliah, dsb.). |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem menyimpan data mata kuliah ke dalam database. |
|  |  | 1. 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. |  |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data mata kuliah. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal menyimpan mata kuliah |
| *Post Condition* | Mata kuliah berhasil dimasukkan dan tersedia di sistem untuk perkuliahan. | |

**Use Case Scenarion 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. |  |
|  |  | 1. Sistem menampilkan formulir untuk memasukkan data ruang kelas (nama ruang, kapasitas, dll.).. |
|  | 1. Biro Administrasi dan Akademik memasukkan data ruang kelas. |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem menyimpan data ruang kelas ke dalam database. |
|  |  | 1. Sistem menampilkan pesan konfirmasi bahwa ruang kelas telah berhasil dimasukkan. |
| *Alternate Flow of*  *Events* | User Action | System Response |
|  | 1. Biro Administrasi dan Akademik memasukkan data yang tidak lengkap atau tidak valid. |  |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data ruang kelas. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal menyimpan ruang kelas. |
| *Post Condition* | Ruang kelas berhasil dimasukkan dan tersedia di sistem untuk penjadwalan. | |

## 

## 

## 

**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. |  |
|  |  | 1. Sistem menampilkan formulir untuk memasukkan inisial dosen.. |
|  | 1. Biro Administrasi dan Akademik memasukkan inisial dosen yang diperlukan. |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem menyimpan inisial dosen ke dalam database. |
|  |  | 1. 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. |  |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data inisial dosen. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal menyimpan inisial dosen |
| *Post Condition* | Inisial dosen berhasil dimasukkan dan dapat digunakan untuk administrasi dosen. | |

**Use Case Scenario 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. |  |
|  |  | 1. Sistem menampilkan formulir untuk memasukkan data program studi (nama program studi, kode, dll.). |
|  | 1. Biro Administrasi dan Akademik memasukkan data program studi. |  |
|  |  | 1. Sistem memvalidasi data yang dimasukkan. |
|  |  | 1. Sistem menyimpan data program studi ke dalam database.. |
|  |  | 1. 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. |  |
|  |  | 1. Sistem menampilkan pesan error untuk memperbaiki data yang salah. |
| *Error Flow of Events* | User Action | System Response |
|  |  | 1. Sistem gagal menyimpan data program studi. |
|  |  | 1. Sistem menampilkan pesan error seperti "Gagal menyimpan program studi. |
| *Post Condition* | Program studi berhasil dimasukkan dan tersedia dalam sistem untuk administrasi akademik. | |

**Use Case Scenario Evaluasi Ticketing**

| *Use Case ID* | UC-12 | |
| --- | --- | --- |
| *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.. |  |
|  |  | 1. Sistem menampilkan daftar tiket yang telah dikelola dan perlu evaluasi. |
|  | 1. Biro Administrasi dan Akademik memilih tiket yang akan dievaluasi. |  |
|  | 1. Biro Administrasi dan Akademik memeriksa detail tiket dan memberikan status (diselesaikan atau ditindaklanjuti). |  |
|  |  | 1. Sistem memperbarui status tiket sesuai dengan keputusan Biro Administrasi dan Akademik. |
|  |  | 1. Sistem menampilkan pesan konfirmasi bahwa tiket telah dievaluasi. |
| *Alternate Flow of*  *Events* | User Action | System Response |
|  | 1. Tidak ada tiket yang tersedia untuk evaluasi. |  |
|  |  | 1. 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. |
|  |  | 1. 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

## 3.4 DFD Level 1

## 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

*Bagian ini memuat karakteristik Pengguna Dapat diresumekan dalam sebuah tabel dengan Kolom: Pengguna, Pekerjaan, Hak Akses. Kolom Hak Akses dihubungkan dengan Fungsi utama yang muncul pada Fungsi Produk. Silakan merujuk pada actor yang tercantum pada use case diagram anda.*

***ADMIN atau SUPER USER, TIDAK*** *perlu dilampirkan pada dokumen ini. Silahkan fokus menjelaskan user – user yang benar-benar berkaitan langsung dengan proses bisnis sistem informasi yang kalian bangun.*

*Penamaan group harus disesuasikan dengan level tertinggi dari user yang terdapat di dalam group tersebut. Misalnya, terdapat 2 user dalam satu group. User tersebut terdiri dari:*

1. *Cashier*
2. *Sales Manager*

*Sales Manager lebih tinggi level-nya dibandingkan Cashier, oleh karena itu nama group yang digunakan mengacu kepada Sales Manager. Sales Manager berada pada departemen Sales, maka nama group yang diberikan adalah,* ***User-Group-Sales***

### 3.6.1 User-Group-[XX\_]

Description of User :

Role :

Prerequisit :

Task description :

### 3.6.2 User-Group-[XX\_]

Description of User :

Role :

Prerequisit :

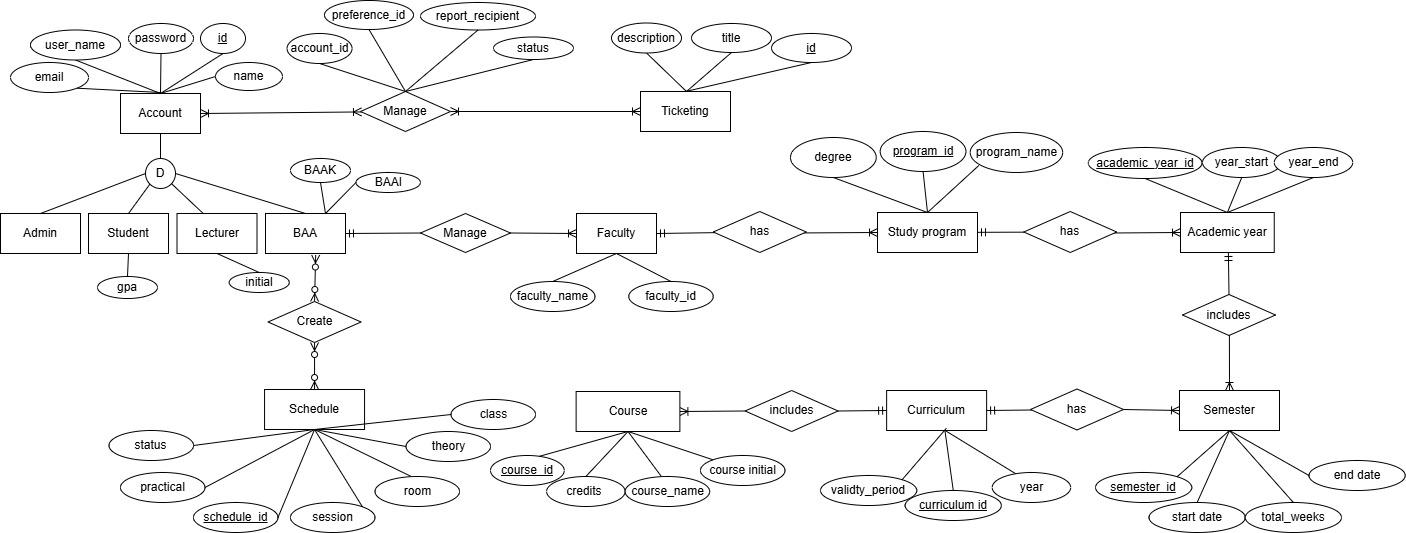
Task description :

# 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



## 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…

| 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 Warta Jemaat Gereja. Adapun yang menjadi kebutuhan perangkat keras yang dibutuhkan dapat dilihat pada tabel ..

| 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, Mozila 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

**4.2.5 Communication Interface**

# 5 Kebutuhan Lainnya

## 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.

| **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.

| **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 Warta Jemaat Gereja 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

| Komponen | Spesifikasi |
| --- | --- |
| Server (Development) | * Processor: Intel Core i5-1135G7 @ 2.40GHz atau i5-11300H @ 3.10GHz. * CPU: Minimal Intel Core i5 11th Gen atau lebih tinggi * RAM: Minimal 8GB. * Storage: SSD 256GB atau lebih * Network: Standar jaringan lokal kampus * Web Server: Apache |
| Client | * PC/Laptop: Minimal Intel Core i5 11th Gen, RAM 8GB. * Browser: Google Chrome, Mozilla Firefox, Microsoft Edge. * Resolusi Layar: 1366x768 atau lebih tinggi |
| DBMS | * MySQL |
| Operating System | * 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 Metodolodi Structure Analysis. Jika menggunakan metodologi lain harus disesuaikan .

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

## Traceability Functional Requirement Summary

| **Features ID** | **Description** |
| --- | --- |
| SyRS-FR-01 | Integrasi dengan API CIS: Sinkronisasi otomatis data mata kuliah, dosen, asisten akademik, dan mahasiswa dari CIS |
| SyRS-FR-02 | Pengelolaan Ruang Kuliah: Alokasi ruang berdasarkan jumlah mahasiswa, menghindari bentrok jadwal, dan penjadwalan ulang otomatis. |
| SyRS-FR-03 | Autentikasi dan Otorisasi: Login menggunakan data CIS, mendukung SSO, serta otorisasi berbasis peran (RBAC). |
| SyRS-FR-04 | Pencarian Jadwal: Pencarian berdasarkan mata kuliah, dosen, hari, ruangan, atau jam kuliah dengan tampilan per minggu/bulan. |
| SyRS-FR-04 | Status dan Pengingat: Status real-time untuk perubahan jadwal, pembatalan kelas, dan pengingat sebelum kelas dimulai. |
| SyRS-FR-04 | Laporan Penjadwalan: Penyediaan laporan penggunaan ruangan, distribusi jadwal, jumlah mahasiswa per kelas, serta analisis kepadatan jadwal. |

## 

| **SyRS** | **No. Proses** | **Keterangan** |
| --- | --- | --- |
| R1 | 01 | Sistem harus mengambil data akademik dari CIS melalui API. |
| R2 | 02 | Sistem harus memastikan tidak ada bentrok penggunaan ruangan. |
| R3 | 03 | Sistem harus menggunakan autentikasi berbasis CIS dan SSO. |
| R4 | 04 | Pengguna harus dapat mencari jadwal berdasarkan berbagai parameter. |
| R5 | 05 | Sistem harus memberikan *update status* real-time terhadap perubahan jadwal. |
| R6 | 06 | Sistem harus menyediakan laporan terkait penggunaan ruangan dan distribusi jadwal. |

**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** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |