|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Create Pendaftaran Sidang PKL** | | | | | **ID:** US1101 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** use case ini membahas alur Pendaftaran sidang PKL yang dilakukan oleh mahasiswa dan diproses oleh admin | | | | | | | | |
| **Trigger:** Mahasiswa mengajukan permohonan pendaftaran sidang PKL melalui sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Mahasiswa sudah login dari web | | | | | | | | |
| **Normal Course:**   1. Mahasiswa mengakses sistem dan memilih menu Pendaftaran Sidang PKL 2. Mahasiswa mengisi formulir pendaftaran yang mencakup data diri 3. Mahasiswa mengunggah dokumen yang dipersyaratkan 4. Sistem memvalidasi kelengkapan data dan dokumen 5. Jika valid, sistem mengirimkan notifikasi pendaftaran berhasil kepada mahasiswa | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Mahasiswa menerima jadwal sidang PKL 2. Admin memiliki data status sidang mahasiswa 3. Sistem menyimpan hasil sidang mahasiswa | | | | | | | | |
| **Exceptions:**  **E1:** Jika mahasiswa tidak melengkapi data atau dokumen yang dipersyaratkan, sistem menampilkan pesan error dan meminta perbaikan (terjadi pada langkah 4). | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Mahasiswa  Dokumen PKL  Jadwal Sidang | | | Form Pendaftaran  Unggahan Mahasiswa | Notifikasi Pendaftaran | | | Mahasiswa  Dosen Penguji  Database Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Read Pendaftaran Sidang PKL** | | | | | **ID:** US1102 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini membahas proses melihat status pendaftaran sidang PKL oleh mahasiswa, dosen, dan admin jurusan | | | | | | | | |
| **Trigger:** Pengguna ingin melihat status pendaftaran sidang PKL. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Sistem telah memiliki data pendaftaran sidang PKL 2. Pengguna memiliki akses ke sistem | | | | | | | | |
| **Normal Course:**   1. Aktor (Mahasiswa, Dosen, atau Admin Jurusan) mengakses sistem 2. Aktor memilih menu "Melihat Page Status Pendaftaran Sidang PKL" 3. Sistem menampilkan daftar pendaftaran sidang PKL 4. Mahasiswa melihat status pendaftarannya 5. Admin jurusan melihat daftar pendaftaran yang masuk 6. Admin jurusan melakukan validasi status pendaftaran 7. Sistem memperbarui status pendaftaran jika validasi berhasil 8. Sistem mengirimkan notifikasi kepada mahasiswa terkait status pendaftaran mereka | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Pengguna mendapatkan informasi terbaru terkait status pendaftaran sidang PKL 2. Sistem memperbarui status pendaftaran mahasiswa jika divalidasi oleh admin | | | | | | | | |
| **Exceptions:**  **E1: Jika sistem tidak dapat menampilkan data karena kesalahan teknis, maka sistem akan menampilkan pesan error**  **E2:** Jika admin belum melakukan validasi, mahasiswa tetap dapat melihat status pendaftarannya sebagai "Menunggu Validasi"  **E3:** Jika mahasiswa tidak menemukan pendaftarannya, mereka dapat menghubungi admin untuk verifikasi | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Pendaftaran Sidang | | | Database Sistem  Mahasiswa | Status Pendaftaran  Notifikasi Status | | | Mahasiswa  Admin | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Update Pendaftaran Sidang PKL** | | | | | **ID:** US1103 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini membahas proses pembaruan data pendaftaran sidang PKL yang dilakukan oleh mahasiswa dan admin jurusan | | | | | | | | |
| **Trigger:** Mahasiswa atau admin jurusan melakukan perubahan pada status atau informasi pendaftaran sidang PKL | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Data pendaftaran sidang PKL telah tersedia di sistem 2. Mahasiswa atau admin jurusan memiliki akses ke sistem. | | | | | | | | |
| **Normal Course:**   1. Aktor (Mahasiswa atau Admin Jurusan) mengakses sistem 2. Mahasiswa memilih menu "Melihat Page Status Daftar Sidang PKL". 3. Mahasiswa dapat mengedit informasi pendaftaran jika diperlukan. 4. Admin jurusan dapat memperbarui status pendaftaran berdasarkan validasi. 5. Sistem menyimpan perubahan data yang dilakukan oleh mahasiswa atau admin 6. Sistem mengirimkan notifikasi update kepada mahasiswa mengenai perubahan status pendaftaran. | | | | | | **Information for Steps**  Flight request  Flight location details  Pilot locations Flight details Pricing guidelines  Flight Request Notification message New Flight Request Notification Record New pilot dashboard notice | | |
| **Postconditions:**   1. Data pendaftaran sidang PKL diperbarui sesuai perubahan yang dilakukan oleh mahasiswa atau admin jurusan. 2. Mahasiswa menerima notifikasi update status pendaftaran mereka | | | | | | | | |
| **Exceptions:**  **E1:** Jika mahasiswa tidak melengkapi data atau dokumen yang dipersyaratkan, sistem menampilkan pesan error dan meminta perbaikan (terjadi pada langkah 4). | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Mahasiswa  Dokumen PKL | | | Form Pendaftaran  Unggahan Mahasiswa | Notifikasi Pendaftaran  Jadwal Sidang | | | Mahasiswa  Dosen Penguji  Database Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Create Sidang TA** | | | | | **ID:** US1401 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini mencakup proses pendaftaran sidang tugas akhir oleh mahasiswa dan pengelolaan data oleh admin jurusan | | | | | | | | |
| **Trigger:** Mahasiswa ingin mendaftar sidang tugas akhir dengan mengunggah persyaratan | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Mahasiswa harus memiliki akun dan telah login ke sistem 2. Mahasiswa telah memenuhi syarat akademik dan administratif untuk sidang TA | | | | | | | | |
| **Normal Course:**   1. Aktor (Mahasiswa atau Admin Jurusan) mengakses sistem 2. Mahasiswa atau admin memilih menu “melihat Page Pendaftaran Sidang TA” 3. Mahasiswa mengunggah persyaratan sidang TA. 4. Sistem menyimpan dokumen yang diunggah. 5. Sistem menyimpan dokumen yang diunggah. 6. Admin jurusan dapat melihat data pendaftaran mahasiswa. 7. Sistem memberikan konfirmasi bahwa pendaftaran berhasil. | | | | | |  | | |
| **Postconditions:**   1. Data pendaftaran sidang TA tersimpan di sistem. 2. Mahasiswa mendapatkan notifikasi bahwa pendaftaran telah berhasil. | | | | | | | | |
| **Exceptions:**  **E1: Jika mahasiswa belum login, sistem meminta mahasiswa untuk login terlebih dahulu.**  **E2: Jika dokumen yang diunggah tidak memenuhi persyaratan, sistem akan menampilkan pesan error.**  **E3: Jika terjadi kesalahan sistem dalam menyimpan data, sistem akan memberikan notifikasi kegagalan.** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Pendaftaran Sidang TA  Dokumen Persyaratan | | | Database Sistem  Mahasiswa | Data Pendaftaran yang Diperbarui  Notifikasi Pendaftaran Berhasil | | | Database Sistem  Mahasiswa | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Read Sidang TA** | | | | | **ID:** US1402 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin, Dosen** | | | | | | | | |
| **Description:** Use case ini memungkinkan mahasiswa, admin jurusan, dan dosen untuk melihat status pendaftaran sidang TA yang telah masuk ke sistem | | | | | | | | |
| **Trigger:** Mahasiswa mengajukan permohonan pendaftaran sidang PKL melalui sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Sistem sudah memiliki data pendaftaran sidang TA. 2. Pengguna telah login ke dalam sistem. | | | | | | | | |
| **Normal Course:**   1. Mahasiswa, admin jurusan, atau dosen mengakses halaman status pendaftaran sidang TA. 2. Sistem menampilkan daftar pendaftaran sidang yang telah masuk 3. Admin jurusan dapat memberikan status pada pendaftaran 4. Sistem mengirimkan notifikasi status kepada mahasiswa. 5. Mahasiswa dapat melihat semua pendaftaran yang masuk. 6. Dosen dapat melihat daftar mahasiswa yang akan sidang. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Pengguna mendapatkan informasi terkini mengenai status pendaftaran sidang TA. 2. Mahasiswa menerima notifikasi terkait status pendaftarannya. | | | | | | | | |
| **Exceptions:**  **E1: Jika mahasiswa mencoba mengedit pendaftaran setelah batas waktu yang ditentukan, sistem akan menampilkan pesan error.**  **E2: Jika admin tidak dapat memperbarui status pendaftaran karena kesalahan teknis, sistem akan memberikan notifikasi kegagalan.**  **E3: Jika mahasiswa atau admin tidak memiliki hak akses yang cukup, sistem akan menolak permintaan update.** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Pendaftaran Sidang  Permintaan Update Data | | | Database Sistem  Mahasiswa, Admin | Data Pendaftaran yang diperbarui  Notifikasi Update | | | Database Sistem  Mahasiswa | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:Update Sidang TA** | | | | | **ID:** US140 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin, Dosen** | | | | | | | | |
| **Description:** Use case ini memungkinkan mahasiswa, admin jurusan, dan dosen untuk memantau dan memperbarui status pendaftaran sidang TA yang telah masuk ke sistem | | | | | | | | |
| **Trigger:** Mahasiswa mengajukan permohonan pendaftaran sidang TA dan ingin mengetahui atau memperbarui statusnya. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Data pendaftaran siding TA sudah tersedia di sistem | | | | | | | | |
| **Normal Course:**   1. Pengguna (Mahasiswa admin jurusan dan dosen) mengakses halaman status pendaftaran sidang TA 2. Sistem menampilkan daftar pendaftaran sidang TA yang telah masuk 3. Admin jurusan dapat memberikan atau memperbarui status pendaftaran 4. Sistem menyimpan status baru dan mengirim notifikasi ke mahasiswa | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Pengguna mendapatkan informasi terkini mengenai status pendaftaran siding TA 2. Mahasiswa menerima notifikasi terkait perubahan status pendaftarannya | | | | | | | | |
| **Exceptions:**  **E1: Jika mahasiswa mencoba mengedit data pendaftaran setelah melewati batas waktu pengajuan, sistem akan menampilkan pesan error dan menolak perubahan.**  **E2: Jika terjadi kesalahan teknis saat admin memperbarui status, sistem akan memberikan notifikasi kegagalan.**  **E3: Jika pengguna (mahasiswa atau admin) tidak memiliki hak akses yang cukup untuk melakukan perubahan, sistem akan menolak permintaan update dan memberikan pesan peringatan.** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Pendaftaran Sidang TA  Permintaan Update Status | | | Mahasiswa, Admin  Admin | Data Pendaftaran yang Diperbarui  Notifikasi Update | | | Database Sistem  Mahasiswa | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:Create Mahasiswa** | | | | | **ID:** US140 | | | **Priority:** High |
| **Actor:Admin** | | | | | | | | |
| **Description:** Use case ini mencakup proses penambahan data mahasiswa baru ke dalam sistem oleh admin. | | | | | | | | |
| **Trigger:** Admin ingin menambahkan data mahasiswa baru ke sistem akademik. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Admin telah login ke dalam sistem dengan hak akses yang sesuai. 2. Sistem dalam keadaan online dan dapat diakses. | | | | | | | | |
| **Normal Course:**   1. Admin Mengakses halaman Mahasiswa 2. Admin memilih opsi Tambah Mahasiswa 3. Sistem menampilkan form input data mahasiswa (nama, NIM, email, Jurusan, Prodi, dll) 4. Admin mengisi form dengan data lengkap dan mengklik tombol simpan 5. Sistem memvalidasi data yang dimasukkan 6. Sistem menyimpan data mahasiswa ke dalam database 7. Sistem menampilkan notifikasi bahwa data mahasiswa berhasil disimpan | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Data mahasiswa tersimpan didalam database 2. Admin menerima notifikasi bahwa data berhasil disimpan | | | | | | | | |
| **Exceptions:**  **E1: Jika terdapat kesalahan Input (misal NIM sudah terdaftar, format email tidak valid), sistem akan menampilkan pesan error dan meminta perbaikan data**  **E2: Jika terjadi kegagalan teknis saat menyimpan data, sistem akan menampilkan notifikasi bahwa proses gagal** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Mahasiswa | | | Admin | Notifikasi Berhasil tambah data  Data mahasiswa tersimpan | | | Database Sistem  Admin | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:Read Mahasiswa** | | | | | **ID:** US140 | | | **Priority:** High |
| **Actor:Admin, Dosen** | | | | | | | | |
| **Description:** Use case ini mencakup proses pencarian dan peninjauan data mahasiswa oleh admin atau dosen untuk keperluan administrative, seperti verifikasi pendaftaran siding PKL dan TA | | | | | | | | |
| **Trigger:** Admin atau dosen ingin melihat data mahasiswa aktif untuk proses pendaftaran sidang | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Pengguna (Admin atau dosen) telah login ke sistem 2. Data mahasiswa telah tersedia di dalam | | | | | | | | |
| **Normal Course:**   1. Pengguna mengakses menu Data Mahasiswa atau Daftar Mahasiswa Aktif 2. Sistem menampilkan daftar mahasiswa yang aktif 3. Pengguna dapat mem-filter atau mencari mahasiswa berdasarkan kriteria tertentu (misal NIM, Nama jurusan, dll) 4. Pengguna memilih salah satu mahasiswa untuk melihat detailnya 5. Sistem menampilkan informasi lengkap mahasiswa (NIM, nama, email, jurusan dll) | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Pengguna mendapatkan informasi yang diperlukan tentang mahasiswa 2. Informasi digunakan untuk proses verifikasi atau pendaftaran siding | | | | | | | | |
| **Exceptions:**  **E1: Jika data mahasiswa tidak ditemukan berdasarkan filter/cari, sistem menampilkan pesan “data tidak ditemukan”**  **E2: Jika Pengguna tidak memiliki untuk melihat data tertentu, sistem menolak permintaan dan menampilkan pesan “Akses Ditolak”** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Permintaan Lihat Mahasiswa  Filter/ Cari Mahasiswa | | | Admin, Dosen  Admin, Dosen | Daftar & Detail Mahasiswa  Informasi Mahasiswa Ditemukan | | | Sistem (Frontend)  Admin,Dosen | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:Update Mahasiswa** | | | | | **ID:** US140 | | | **Priority:** High |
| **Actor: Admin** | | | | | | | | |
| **Description:** Use case ini mencakup proses pengubahan data mahasiswa yang sudah ada di sistem oleh admin, seperti pembaruan data identitas, jurusan, atau status akademik | | | | | | | | |
| **Trigger:** Admin ingin memperbarui informasi mahasiswa yang telah tersimpan di dalam sistem | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Admin telah login ke sistem dengan hak akses yang sesuai 2. Data mahasiswa yang akan diperbarui sudah tersedia dalam sistem | | | | | | | | |
| **Normal Course:**   1. Admin mengakses menu Manajemen Mahasiswa 2. Admin mencari dan memilih data mahasiswa yang ingin diperbarui 3. Sistem menampilkan detail data mahasiswa yang tersimpan 4. Admin mengklik tombol edit dan mengubah data yang diperlukan (contoh: email, jurusan, dll) 5. Admin menekan tombol Simpan Perubahan 6. Sistem melakukan validasi terhadap data baru yang dimasukkan 7. Sistem menyimpan perubahan ke dalam database 8. Sistem menampilkan notifikasi bahwa data mahasiswa berhasil diperbarui | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Data mahasiswa di dalam sistem berhasil diperbarui sesuai input dari admin 2. Sistem menyimpan jejak perubahan (log) jika diperlukan | | | | | | | | |
| **Exceptions:**  **E1: Jika data yang dimasukkan tidak valid (misalnya format email salah), sistem menampilkan pesan error**  **E2: Jika terjadi kesalahan teknis saat menyimpan data, sistem menampilkan notifikasi kegagalan**  **E3: Jika admin mencoba memperbarui data mahasiswa yang tidak ada, sistem menampilkan pesan “Data Tidak Ditemukan”** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data perubahan Mahasiswa | | | Admin | Notifikasi Update Berhasil  Data Mahasiswa yang diperbarui | | | Admin  Database Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:Delete Mahasiswa** | | | | | **ID:** US140 | | | **Priority:** High |
| **Actor: Admin** | | | | | | | | |
| **Description:** Use case ini mencakup proses penghapusan data mahasiswa dari sistem oleh admin, misalnya karena mahasiswa tidak aktif, dropout, atau data duplikat | | | | | | | | |
| **Trigger:** Admin ingin menghapus data mahasiswa yang tidak lagi diperlukan di sistem | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Admin telah login ke sistem dengan hak akses yang sesuai 2. Data mahasiswa yang akan dihapus sudah tersedia dalam sistem | | | | | | | | |
| **Normal Course:**   1. Admin membuka Manajemen Mahasiswa 2. Admin mencari dan memilih mahasiswa yang akan dipilih 3. Sistem menampilkan detail data mahasiswa yang dipilih 4. Admin mengklik tombol hapus dan sistem menampilkan pesan konfirmasi 5. Admin mengonfirmasi penghapusan data 6. Sistem menghapus data mahasiswa dari database 7. Sistem menampilkan notifikasi bahwa data berhasil dihapus | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Data mahasiswa yang dipilih telah dihapus dari sistem 2. Sistem mencatat penghapusan untuk keperluan audit/log jika diperlukan | | | | | | | | |
| **Exceptions:**  **E1: Jika admin mencoba menghapus mahasiswa yang tidak ditemukan, sistem akan menampilkan pesan “Data tidak ditemukan”**  **E2: Jika penghapusan gagal karena error teknis (misal: integritas data terganggu karena relasi ke data lain), sistem akan menampilkan notifikasi kegagalan**  **E3: Jika admin tidak memiliki hak akses untuk melakukan penghapusan, sistem akan menolak aksi dan menampilkan pesan “Akses Ditolak”** | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Permintaan Hapus Data  Data Mahasiswa | | | Admin  Database | Notifikasi Penghapusan Berhasil  Data Mahasiswa Terhapus | | | Admin  Database Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Create Penguji Sidang PKL** | | | | | **ID:** US1101 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini membahas proses pendaftaran sidang yang dilakukan oleh mahasiswa melalui sistem dan diproses oleh admin hingga menghasilkan jadwal sidang. | | | | | | | | |
| **Trigger:** Mahasiswa mengajukan permohonan pendaftaran sidang melalui sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Mahasiswa mengakses sistem dan melakukan login. | | | | | | | | |
| **Normal Course:**   1. Mahasiswa mengakses sistem dan melakukan login. 2. Sistem melakukan autentikasi dan menampilkan halaman utama. 3. Mahasiswa memilih aksi atau opsi (Upload Berkas atau Daftar Sidang). 4. Jika mahasiswa memilih "Upload Berkas Sidang":  * Mahasiswa mengunggah dokumen yang dipersyaratkan. * Sistem memverifikasi dokumen. * Admin menyetujui berkas yang telah diunggah. * Sistem memperbarui status berkas menjadi "Berkas telah diupdate".  1. Jika mahasiswa memilih "Daftar Sidang":  * Mahasiswa mengisi dan mengirimkan formulir pendaftaran. * Sistem melakukan validasi pendaftaran. * Admin menyetujui pendaftaran sidang. * Sistem membuat jadwal sidang.  1. Sistem menampilkan notifikasi bahwa pendaftaran sidang berhasil. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Mahasiswa menerima jadwal sidang. 2. Admin memiliki data status pendaftaran dan berkas sidang mahasiswa. 3. Sistem menyimpan data hasil verifikasi dan penjadwalan sidang. | | | | | | | | |
| **Exceptions:**  **E1:** Jika mahasiswa tidak mengisi data atau mengunggah dokumen yang dipersyaratkan, sistem akan menampilkan pesan kesalahan dan meminta mahasiswa untuk memperbaiki isian atau mengunggah ulang dokumen (terjadi pada langkah 4 atau 5). | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Mahasiswa  Dokumen Sidang  Jadwal Sidang  Status Berkas | | | Form Pendaftaran  Unggahan Mahasiswa  Sistem  Sistem | Notifikasi Pendaftaran  Status Verifikasi Berkas  Jadwal Sidang Dibuat  Data Tersimpan | | | Mahasiswa  Admin  Dosen Penguji, Mahasiswa  Database Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Read Penguji Sidang PKL** | | | | | **ID:** US1102 | | | **Priority:** Medium |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini menjelaskan proses mahasiswa dalam mengakses hasil sidang melalui sistem setelah mengunggah berkas sidang. Sistem secara otomatis menampilkan jadwal dan hasil sidang yang relevan. | | | | | | | | |
| **Trigger:** Mahasiswa melakukan login ke sistem dan memilih aksi untuk melihat hasil sidang atau mengunggah berkas. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Mahasiswa sudah berhasil login ke dalam sistem. 2. Data hasil sidang sudah tersedia di sistem (untuk melihat hasil). | | | | | | | | |
| **Normal Course:**   1. Mahasiswa login ke dalam sistem. 2. Sistem melakukan autentikasi dan mengizinkan akses ke halaman utama. 3. Mahasiswa memilih salah satu aksi:  * Upload Berkas Sidang:  1. Mahasiswa mengunggah berkas sidang. 2. Sistem memproses dan menampilkan jadwal sidang.  * Melihat Hasil Sidang:  1. Mahasiswa memilih opsi melihat hasil. 2. Sistem menampilkan hasil sidang kepada mahasiswa. 3. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Mahasiswa mendapatkan jadwal sidang yang sesuai (jika mengunggah berkas). 2. Mahasiswa melihat hasil sidang (jika memilih aksi melihat hasil). 3. Sistem mencatat aktivitas mahasiswa. | | | | | | | | |
| **Exceptions:**  **E1:** Jika mahasiswa belum mengunggah berkas, sistem tidak dapat menampilkan jadwal dan memberi pesan bahwa unggahan belum tersedia.  **E2:** Jika hasil sidang belum diinput oleh admin, sistem menampilkan informasi bahwa hasil belum tersedia. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Berkas Sidang  Permintaan Hasil  Data Hasil Sidang | | | Mahasiswa  Mahasiswa  Admin | Jadwal Sidang  Hasil Sidang  Informasi ditampilkan | | | Mahasiswa  Mahasiswa  Sistem ke Mahasiwa | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Update Penguji Sidang PKL** | | | | | **ID:** US1103 | | | **Priority:** High |
| **Actor: Mahasiswa, Admin** | | | | | | | | |
| **Description:** Use case ini menjelaskan proses pengunggahan berkas sidang oleh mahasiswa yang kemudian diverifikasi oleh sistem dan disetujui oleh admin. Setelah disetujui, sistem memperbarui status berkas menjadi "disetujui". | | | | | | | | |
| **Trigger:** Mahasiswa mengunggah berkas sidang melalui sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Mahasiswa telah login ke dalam sistem. 2. Berkas sidang telah disiapkan untuk diunggah. | | | | | | | | |
| **Normal Course:**   1. Mahasiswa mengakses sistem dan mengunggah berkas sidang. 2. Sistem melakukan verifikasi terhadap berkas yang diunggah. 3. Setelah verifikasi sistem, data berkas diteruskan ke admin. 4. Admin meninjau dan menyetujui berkas. 5. Sistem memperbarui status berkas menjadi "Berkas telah diupdate". 6. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Berkas mahasiswa berhasil diverifikasi dan disetujui. 2. Status berkas di sistem telah diperbarui. | | | | | | | | |
| **Exceptions:**  **E1:** Jika berkas tidak sesuai format atau tidak lengkap, sistem menolak unggahan dan memberikan pesan kesalahan kepada mahasiswa.  **E2:** Jika admin menolak berkas, sistem mengembalikan berkas ke mahasiswa untuk diperbaiki. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Berkas Sidang  Verifikasi Sistem  Persetujuan | | | Mahasiswa  Sistem  Admin | Status Berkas Disetujui  Persetujuan Admin  Notifikasi Status | | | Sistem  Admin  Mahasiswa | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Create Nilai PKL** | | | | | **ID:** US1104 | | | **Priority:** High |
| **Actor: Dosen** | | | | | | | | |
| **Description:** Use case ini menjelaskan proses input nilai sidang PKL oleh dosen ke dalam sistem yang kemudian divalidasi dan disimpan ke dalam database. | | | | | | | | |
| **Trigger:** Dosen mengakses menu nilai sidang dan memilih opsi untuk menginput nilai. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Dosen telah login ke sistem. 2. Mahasiswa sudah terdaftar untuk sidang dan memiliki data sidang aktif. | | | | | | | | |
| **Normal Course:**   1. Dosen login ke sistem. 2. Sistem melakukan autentikasi dan menampilkan halaman utama. 3. Dosen memilih menu "Nilai Sidang". 4. Sistem menampilkan halaman nilai. 5. Dosen mengklik tombol “Input Nilai”. 6. Dosen mengisi data nilai untuk mahasiswa. 7. Dosen menekan tombol simpan. 8. Sistem memvalidasi data nilai. 9. Jika valid, sistem menyimpan data nilai ke dalam database. 10. Sistem memberikan notifikasi bahwa nilai berhasil disimpan. 11. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Nilai mahasiswa berhasil tersimpan dalam database. 2. Dosen mendapatkan notifikasi bahwa proses input nilai berhasil. | | | | | | | | |
| **Exceptions:**  **E1:** Jika terdapat data nilai yang tidak valid (misalnya di luar rentang nilai), sistem akan menampilkan pesan error dan meminta perbaikan sebelum menyimpan.  **E2:** Jika terjadi gangguan koneksi saat penyimpanan, sistem akan memberi tahu dosen untuk mencoba kembali. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Data Nilai  Aksi Simpan  Notifikasi | | | Dosen  Dosen  Sistem | Validasi Nilai  Nilai Tersimpan  Notifikasi Berhasil | | | Sistem  Database  Dosen | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Read Nilai PKL** | | | | | **ID:** US1105 | | | **Priority:** Medium |
| **Actor: Dosen, Admin Jurusan** | | | | | | | | |
| **Description:** Use case ini menggambarkan proses dosen atau admin jurusan dalam melihat daftar nilai PKL mahasiswa yang telah tersimpan di dalam sistem. | | | | | | | | |
| **Trigger:** Dosen/Admin jurusan ingin melihat daftar nilai mahasiswa. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Pengguna (dosen/admin jurusan) telah login ke sistem. 2. Data nilai PKL mahasiswa sudah tersedia dalam database. | | | | | | | | |
| **Normal Course:**   1. Pengguna melakukan login. 2. Sistem melakukan proses autentikasi. 3. Sistem menampilkan halaman utama. 4. Pengguna memilih menu "Nilai Sidang". 5. Sistem menampilkan daftar nilai sementara. 6. Pengguna mengklik opsi “Lihat Nilai”. 7. Sistem mengambil data nilai dari database. 8. Sistem menampilkan daftar nilai lengkap dari database. 9. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Data nilai berhasil ditampilkan kepada pengguna. | | | | | | | | |
| **Exceptions:**  **E1:** Sistem gagal mengambil data dari database karena gangguan jaringan atau kesalahan query, maka akan ditampilkan pesan kesalahan.  **E2:** Jika pengguna belum login, sistem akan meminta login terlebih dahulu. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Permintaan daftar nilai  Aksi “Lihat Nilai” | | | Dosen / Admin Jurusan  Dosen / Admin Jurusan | Autentikasi & Tampilan  Data Nilai Mahasiswa | | | Sistem  Sistem & Database | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Update Nilai PKL** | | | | | **ID:** US1106 | | | **Priority:** Medium |
| **Actor: Dosen** | | | | | | | | |
| **Description:** Use case ini menggambarkan proses dosen dalam memperbarui atau mengedit nilai PKL mahasiswa yang telah diinput sebelumnya. | | | | | | | | |
| **Trigger:** Dosen ingin mengubah nilai PKL mahasiswa karena ada kesalahan atau revisi nilai. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Dosen telah login ke dalam sistem. 2. Data nilai PKL mahasiswa sebelumnya telah tersedia di sistem. | | | | | | | | |
| **Normal Course:**   1. Dosen melakukan login ke sistem. 2. Sistem melakukan proses autentikasi. 3. Sistem menampilkan halaman utama. 4. Dosen memilih menu “Nilai Sidang”. 5. Sistem menampilkan daftar nilai yang tersedia. 6. Dosen mengklik “Edit” pada salah satu nilai. 7. Dosen melakukan perubahan nilai. 8. Dosen mengklik tombol "Simpan". 9. Sistem melakukan validasi data nilai. 10. Sistem mengupdate data nilai ke dalam database. 11. Sistem menampilkan notifikasi bahwa nilai berhasil diperbarui. 12. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Nilai yang telah diubah tersimpan dan diperbarui di dalam database. | | | | | | | | |
| **Exceptions:**  **E1:** Data nilai yang diubah tidak valid → sistem menampilkan pesan kesalahan dan meminta revisi input.  **E2:** Terjadi kesalahan saat proses penyimpanan ke database → sistem menampilkan pesan kegagalan simpan. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Permintaan Edit Nilai  Data nilai yang diubah | | | Dosen  Dosen | Validasi data & Update nilai  Konfirmasi Penyimpanan Berhasil | | | Sistem  Sistem | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Delete Nilai PKL** | | | | | **ID:** US1107 | | | **Priority:** High |
| **Actor: Admin Jurusan** | | | | | | | | |
| **Description:** Use case ini membahas proses penghapusan nilai sidang PKL oleh Admin Jurusan melalui sistem. Proses dimulai dari login hingga nilai sidang berhasil dihapus dari sistem dan database. | | | | | | | | |
| **Trigger:** Admin jurusan ingin menghapus nilai sidang PKL tertentu dari sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Admin Jurusan telah memiliki akun yang valid dan dapat login ke sistem. 2. Nilai sidang PKL mahasiswa sudah tersimpan sebelumnya di database. | | | | | | | | |
| **Normal Course:**   1. Admin jurusan melakukan login ke dalam sistem. 2. Sistem melakukan proses autentikasi dan menampilkan halaman utama. 3. Admin memilih menu “Nilai Sidang”. 4. Sistem menampilkan daftar nilai sidang PKL. 5. Admin memilih salah satu nilai untuk dihapus. 6. Admin mengonfirmasi penghapusan nilai. 7. Sistem menghapus data nilai dari database. 8. Sistem menampilkan pesan bahwa nilai berhasil dihapus. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Nilai sidang PKL yang dipilih telah terhapus dari sistem dan database. 2. Admin tidak lagi melihat nilai yang telah dihapus pada daftar nilai. | | | | | | | | |
| **Exceptions:**  **E1:** Jika autentikasi gagal saat login, sistem menampilkan pesan kesalahan dan meminta login ulang.  **E2:** Jika proses penghapusan gagal (misal database error), sistem menampilkan pesan gagal dan meminta coba ulang. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Username & Password  Perintah Hapus Nilai  Konfirmasi Penghapusan | | | Admin Jurusan  Admin Jurusan  Admin Jurusan | Halaman Utama  Data Nilai Dihapus dari DB  Notifikasi Nilai Berhasil Dihapus | | | Admin  Database  Admin | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Create Dosen** | | | | | **ID:** US1108 | | | **Priority:** High |
| **Actor: Dosen** | | | | | | | | |
| **Description:** Use case ini menggambarkan proses input nilai sidang oleh dosen setelah melakukan login ke sistem. Proses ini melibatkan interaksi antara dosen dan sistem hingga data nilai sidang tersimpan di database. | | | | | | | | |
| **Trigger:** Dosen memilih opsi untuk menginput nilai sidang dari menu utama sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Dosen sudah memiliki akun dan login ke sistem dengan kredensial yang valid. 2. Sidang PKL mahasiswa sudah dilaksanakan sehingga nilai bisa diinput. | | | | | | | | |
| **Normal Course:**   1. Dosen melakukan login ke sistem. 2. Sistem melakukan proses autentikasi dan menampilkan halaman utama. 3. Dosen memilih aksi atau opsi pada halaman utama. 4. Sistem menampilkan menu yang relevan. 5. Dosen memilih opsi untuk menginput nilai sidang. 6. Dosen mengisi formulir nilai sidang. 7. Sistem menyimpan nilai ke database. 8. Sistem mengonfirmasi bahwa nilai telah tersimpan. 9. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Nilai sidang mahasiswa tersimpan dalam sistem dan database. 2. Dosen mendapat notifikasi bahwa data telah berhasil disimpan. | | | | | | | | |
| **Exceptions:**  **E1:** Jika login gagal (misal: username/password salah), sistem menampilkan pesan kesalahan.  **E2:** Jika input nilai gagal disimpan (misal: masalah koneksi atau validasi data), sistem menampilkan pesan error dan meminta untuk mencoba lagi. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Username & Password  Opsi Input Nilai Sidang  Nilai Sidang Mahasiswa | | | Dosen  Dosen  Dosen | Halaman Utama  Data Nilai Sidang  Konfirmasi Nilai Tersimpan | | | Dosen  Sistem & Database  Dosen | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Read Dosen** | | | | | **ID:** US1109 | | | **Priority:** Medium |
| **Actor: Dosen** | | | | | | | | |
| **Description:** Use case ini menjelaskan proses dosen dalam melihat hasil nilai sidang mahasiswa melalui sistem setelah berhasil login dan memilih aksi yang tersedia. | | | | | | | | |
| **Trigger:** Dosen memilih aksi untuk melihat hasil nilai sidang dari halaman utama sistem. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Dosen telah memiliki akun dan login ke sistem dengan kredensial yang benar. 2. Data nilai sidang sudah tersedia dalam database. | | | | | | | | |
| **Normal Course:**   1. Dosen melakukan login ke sistem. 2. Sistem melakukan autentikasi dan menampilkan halaman utama. 3. Dosen memilih aksi atau opsi yang tersedia di halaman utama. 4. Dosen memilih untuk melihat hasil nilai sidang. 5. Sistem mengambil data hasil nilai sidang dari database. 6. Sistem menampilkan hasil nilai sidang kepada dosen. 7. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Dosen berhasil melihat data hasil nilai sidang mahasiswa yang tersimpan di sistem. 2. Tidak ada perubahan data pada sistem. | | | | | | | | |
| **Exceptions:**  **E1:** Jika login gagal, sistem menampilkan pesan kesalahan dan meminta dosen mengulangi proses login.  **E2:** Jika data nilai sidang belum tersedia di database, sistem menampilkan pesan “Data tidak ditemukan”. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Username & Password  Permintaan Hasil Sidang  Nilai Sidang dari DB | | | Dosen  Dosen  Database | Halaman Utama  Data Hasil Nilai Sidang  Tampilan Hasil Nilai Sidang | | | Dosen  Sistem  Dosen | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name: Update Dosen** | | | | | **ID:** US1110 | | | **Priority:** Medium |
| **Actor: Dosen** | | | | | | | | |
| **Description:** Use case ini menjelaskan proses dosen dalam mengedit atau memperbarui nilai sidang mahasiswa yang telah tersimpan di sistem. | | | | | | | | |
| **Trigger:** Dosen memilih aksi untuk melakukan pengeditan nilai sidang dari halaman utama. | | | | | | | | |
| **Type:** | External | Temporal |  |  | |  |  | |
| **Preconditions:**   1. Dosen telah memiliki akun dan berhasil login ke sistem. 2. Nilai sidang yang ingin diedit sudah tersedia dalam database. | | | | | | | | |
| **Normal Course:**   1. Dosen melakukan login ke sistem. 2. Sistem melakukan autentikasi dan menampilkan halaman utama. 3. Dosen memilih aksi atau opsi yang tersedia. 4. Dosen memilih untuk mengedit nilai sidang. 5. Sistem memproses perubahan dan memperbarui nilai pada database. 6. Sistem mengonfirmasi bahwa nilai telah berhasil diubah. 7. Proses selesai. | | | | | | **Information for Steps** | | |
| **Postconditions:**   1. Nilai sidang mahasiswa telah diperbarui di dalam database. | | | | | | | | |
| **Exceptions:**  **E1:** Jika login gagal, sistem akan menampilkan pesan kesalahan dan meminta pengguna untuk mengulangi proses login.  **E2:** Jika data tidak ditemukan saat ingin diubah, sistem menampilkan pesan kesalahan “Data tidak ditemukan”.  **E3:** Jika terjadi kegagalan saat menyimpan ke database, sistem akan menampilkan notifikasi error. | | | | | | | | |
| **Summary Inputs** | | | **Source** | **Summary Outputs** | | | **Destination** | |
| Username & Password  Perubahan Nilai  Konfirmasi Perubahan | | | Dosen  Dosen  Sistem | Halaman Utama  Nilai disimpan di database  Notifikasi “Nilai diubah” | | | Dosen  Database  Dosen | |