**SKPL**-xxxx

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

ScholarSeeker - Sistem Informasi Beasiswa Berbasis Web

untuk:

ScholarSeeker

Dipersiapkan oleh:

|  |  |  |
| --- | --- | --- |
| **Role** | **Nama** | **NIM** |
|  | Aanisah Rifda R. | 1301174057 |
|  | Irbah Salsabila | 1301174021 |
|  | Iriyanto | 1301174295 |
|  | Rachmansyah Adhi W. | 1301170066 |

Program Studi S1 Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  -  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *SKPL-xxx* | |  |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

# Daftar Perubahan

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | Penambahan Use Case |
| B | Penambahan Use Case Scenario |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| TGL |  |  |  |  |  |  |  |  |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

# Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

# Daftar Isi

[Daftar Perubahan 3](#_Toc9019805)

[Daftar Halaman Perubahan 4](#_Toc9019806)

[Daftar Isi 5](#_Toc9019807)

[1. Pendahuluan 7](#_Toc9019808)

[1.1 Tujuan Penulisan Dokumen 7](#_Toc9019809)

[1.2 Ruang Lingkup / Cakupan Dokumen 7](#_Toc9019810)

[1.3 Definisi, Singkatan, dan Akronim 7](#_Toc9019811)

[1.4 Referensi 8](#_Toc9019812)

[2. Deskripsi Rinci Perangkat Lunak 9](#_Toc9019813)

[2.1 Statement of Objective Perangkat Lunak 9](#_Toc9019814)

[2.2 Profil dan Karakteristik Pengguna 10](#_Toc9019815)

[2.3 Lingkungan Operasi 12](#_Toc9019816)

[2.4 Asumsi dan Dependensi 12](#_Toc9019817)

[3. Deskripsi Rinci Perangkat Lunak 14](#_Toc9019818)

[3.1 Deskripsi Kebutuhan 14](#_Toc9019819)

[3.1.1 Kebutuhan Fungsional 14](#_Toc9019820)

[3.1.2 Kebutuhan Non-Fungsional 15](#_Toc9019821)

[3.2 Pemodelan Analisis 16](#_Toc9019822)

[3.2.1 Usecase Diagram 16](#_Toc9019823)

[3.2.1.1 Usecase Scenario #1 17](#_Toc9019824)

[3.2.1.2 Usecase Scenario #2 17](#_Toc9019825)

[3.2.1.3 Usecase Scenario #3 18](#_Toc9019826)

[3.2.1.4 Usecase Scenario #4 18](#_Toc9019827)

[3.2.1.5 Usecase Scenario #5 19](#_Toc9019828)

[3.2.1.6 Usecase Scenario #6 19](#_Toc9019829)

[3.2.1.7 Usecase Scenario #7 20](#_Toc9019830)

[3.2.1.8 Usecase Scenario #8 20](#_Toc9019831)

[3.2.1.9 Usecase Scenario #9 21](#_Toc9019832)

[3.2.1.10 Usecase Scenario #10 21](#_Toc9019833)

[3.2.1.11 Usecase Scenario #11 22](#_Toc9019834)

[3.2.1.12 Usecase Scenario #12 23](#_Toc9019835)

[3.2.2 Class Diagram: 24](#_Toc9019836)

[4. Requirements Antarmuka Eksternal 25](#_Toc9019837)

[4.1 Antarmuka Pengguna 25](#_Toc9019838)

[4.2 Antarmuka Perangkat Keras 26](#_Toc9019839)

[4.3 Antarmuka Perangkat Lunak 26](#_Toc9019840)

[4.4 Antarmuka Komunikasi 26](#_Toc9019841)

[5. Requirements Lain 27](#_Toc9019842)

[Lampiran A: Daftar Kata-Kata Sukar 28](#_Toc9019843)

[Lampiran B: Analysis Models 29](#_Toc9019844)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi Spesifikasi Kebutuhan Perangkat Lunak (SKPL) untuk sistem *ScholarSeeker*. Tujuan penulisan dokumen ini adalah untuk memberi penjelasan mengenai sistem *ScholarSeeker* yang akan dibangun. Dokumen ini digunakan sebagai acuan dari proses pengembangan aplikasi ini. Diharapkan pengembangan dari website menjadi lebih terfokus dan tidak menimbulkan pengambiguan bagi pengembang atau *programmer*.

## Ruang Lingkup / Cakupan Dokumen

Semua hal yang tercantum di dalam dokumen ini merupakan bagian dari ruang lingkup kebutuhan pembangunan perangkat lunak yang berupa *website* yang digunakan untuk pengelolaan sistem informasi pencarian dan pendaftaran beasiswa untuk mahasiswa, mengelola data mahasiswa, mengelola data pemberi mahasiswa, mengelola beasiswa, mengelola data admin, dan mengelola data berita.

## Definisi, Singkatan, dan Akronim

SKPL : Spesifikasi Kebutuhan Perangkat Lunak / Software Requirement Specification. Spesifikasi dari suatu produk/program yang melakukan suatu fungsi tertentu pada lingkungan tertentu.

DPPL : Deskripsi Perancangan Perangkat Lunak merupakan deskripsi dari perangkat lunak yang akan di kembangkan.

*ScholarSeeker* : adalah sistem informasi berbasis TI yang digunakan oleh Pelamar/Pendaftar Beasiswa, Pendonor, dan Pengelola dalam mengelola segala proses yang terkait (Pendaftaran, Operasional, Laporan, dll) untuk setiap Paket Beasiswa.

Beasiswa adalah bantuan yang diberikan kepada mahasiswa sebagai bentuk apresiasi prestasi mahasiswa ataupun berupa bantuan Biaya Pendidikan bagi mahasiswa yang mempunyai keterbatasan finansial.

Paket Beasiswa adalah sebuah penawaran Beasiswa dengan syarat dan ketentuan yang berlaku.Pelamar/Pendaftar Beasiswa adalah seseorang yang membutuhkan Dukungan biaya pendidikan dan biaya penunjang pendidikan baik yang telah terdaftar sebagai Mahasiswa S1 atau S2.Penerima Beasiswa adalah Pelamar/Pendaftar Beasiswa yang sudah disetujui oleh Pengelola dan Pendonor dari suatu Paket Beasiswa.Pengelola adalah pihak Telkom (Direktorat Kemahasiswaan Telkom/Fakultas) yang mengelola Paket Beasiswa.Pendonor adalah pihak yang memberikan Beasiswa.

## Referensi

Dokumen yang digunakan sebagai referensi dalam pembuatan SKPL ini adalah sebagai berikut:

1. *IEEE Std 830 1993, IEEE Recommended Parctice for Software Requirement Specifications.*
2. *Software Engineering, Aparctitioner’s Approach 5th edition, Roger S Pressman, Mc Graw Hill, 2001.*
3. Spesifikasi Kebutuhan Perangkat Lunak Aplikasi Komunitas Auditor Sistem Informasi (ISAC) Berbasis Web.
4. Template SKPL Analisis Berorientasi Objek.
5. Spesifikasi Kebutuhan Perangkat Lunak Pemanfaatan Teknologi Aplikasi Bergerak Untuk Pencarian Pemondokan Bagi Mahasiswa.

# Deskripsi Rinci Perangkat Lunak

## Statement of Objective Perangkat Lunak

ScholarSeeker dibuat untuk mempermudah user untuk mendaftarkan beasiswa. Beasiswa yang ditawarkan bukan hanya beasiswa dari universitas untuk S1/S2/S3, tetapi juga beasiswa dari perusahaan, lembaga, atau perorangan. Sudah banyak website tentang beasiswa tetapi website beasiswa kebanyakan yang ada hanya memberikan informasi atau hanya untuk mendaftar. Berbeda dengan ScholarSeeker, ScholarSeeker merupakan website beasiswa dimana mahasiswa dapat mencari informasi serta sekaligus mendaftar beasiswa langsung di website tersebut. Adanya, ScholarSeeker diharapkan berguna untuk mahasiswa mencari beasiswa dan pemberi beasiswa untuk mempermudah pengumpulan pendaftaran. Fitur-fitur akan mempermudah penggunanya dengan tampilan yang interaktif dan user-friendly.

Perspektif dan Fungsi Perangkat Lunak

Perangkat lunak yang akan dikembangkan adalah ScholarSeeker, yaitu merupakan perangkat lunak yang digunakan oleh mahasiswa untuk mencari dan mendaftar beasiswa dan pemberi beasiswa sebagai penyalur beasiswa. Perangkat lunak ini dapat memberikan informasi mengenai beasiswa untuk mahasiswa berdasarkan kategori, mencari dan mendaftar beasiswa, menginput beasiswa untuk mahasiswa, berita tentang beasiswa. ScholarSeeker dapat melakukan hal-hal berikut ini:

1. Dapat melakukan registrasi dan login untuk mahasiswa dan pemberi beasiswa. Hal ini dikarenakan mahasiswa dan pemberi mahasiswa yang dapat menggunakan merupakan mahasiswa dan pemberi beasiswa yang telah terverifikasi dan terdaftar di *database*.
2. Mahasiswa dapat mencari beasiswa berdasarkan kategori dan waktu.
3. Mahasiswa dapat mendaftar beasiswa secara langsung.
4. Pemberi Beasiswa dapat menginput beasiswa sesuai form dan persyaratan yang dibutuhkan,

Dengan adanya *ScholarSeeker* ini diharapkan mahasiswa dapat mencari dan mendaftar beasiswa dengan mudah. Serta, Pemberi Beasiswa dapat memberikan informasi beasiswanya secara mudah dan gampang.

SchoolarSeeker mempunyai beberapa fungsi diantaranya adalah :

1. Mempermudah mahasiswa/i dalam mencari beasiswa
2. Mempermudah Pemberi Beasiswa untuk memberikan info beasiswanya.
3. Admin bisa melihat berapa pengunjung terbanyak di dalam info beasiswa tersebut.
4. Admin memberikan informasi berita yang bermanfaat (contoh-contoh soal TOEFL ataupun kita-kiat dalam mendapatkan beasiswa).
5. Mempermudah mahasiswa bisa mendapatkan info beasiswa untuk dalam negeri maupun luar negeri.
6. Mempermudah karena mahasiswa dapat menghubungi pemberi beasiswanya melalui email..
7. Terdapat fitur input,edit dan delete untuk pemberi beasiswa dan admin.

## Profil dan Karakteristik Pengguna

Perangkat lunak *SchoolarSeeker* ini merupakan perangkat lunak yang digunakan oleh Mahasiswa untuk mencari beasiswa dengan sangat mudah. Sistem ini berkaitan dengan beberapa entitas luar, yaitu *Admin,* Mahasiswa dan Pemberi beasiswa. Hal – hal yang dilakukan oleh entitas – entitas tersebut adalah:

Admin:

⸰ Dapat melihat laporan Pengunjung

⸰ Dapat melihat laporan pengguna

⸰ Memvalidasi akun mahasiswa

⸰ Dapat mengetahui info beasiswa yang diberikan oleh pemberi beasiswa

⸰ Dapat mengedit data berita

⸰ Dapat menginputkan data berita

⸰ Dapat menghapus data berita

Mahasiswa:

⸰ Dapat melakukan Registrasi Akun

⸰ Dapat melakukan Login akun

⸰ Dapat mendaftar Beasiswa

⸰ Dapat melihat info beasiswa

⸰ Dapat menginputkan data diri

⸰ Dapat mengupdate data diri

⸰ Dapat melihat berita

Pemberi Beasiswa:

⸰ Dapat melakukan Registrasi Akun

⸰ Dapat melakukan login akun

⸰ Dapat mengelola beasiswa

⸰ Dapat mengelola pengajuan beasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Pengguna | Tugas | Hak Akses ke Aplikasi | Kemampuan yang harus dimiliki |
| Admin | Admin dapat menginputkan berita di halaman web, lalu admin juga dapat melihat laporan pengunjung dan laporan pengguna | Admin dapat input,delete dan edit berita.  Admin dapat memverifikasi akun mahasiswa. | Dapat menggunakan komputer beserta website |
| Mahasiswa | Mencari informasi beasiswa sesuai keinginan mahasiswa yang tertera di dalam web, lalu melakukan pengajuan beasiswa. | Mahasiswa dapat menginputkan data diri,Mahasiswa dapat memilih beasiswa sesuai keinginan. | Dapat menggunakan komputer beserta website |
| Pemberi beasiswa | Melakukan penginputkan informasi Beasiswa yang akan diberikan kepada para Mahasiswa | Pemberi Beasiswa dapat upload informasi beasiswa. | Dapat menggunakan komputer beserta website |

## Lingkungan Operasi

Aplikasi ini akan digunakan pada sistem pencarian beasiswa melalui aplikasi berbasis web, sehingga ada banyak yang mengakses aplikasi ini, seperti *admin*, mahasiswa, dan pemberi beasiswa. Aplikasi ini akan dapat diakses menggunakan perangkat seperti komputer jinjing dengan sistem operasi mulai Windows 7 sampai Windows 10, dengan bantuan *browser* Google Chrome atau Mozilla Firefox. Selain itu *SchoolarSeeker* juga bisa diakses melalui gawai berbasis Android atau iOS, dengan bantuan *browser* yang ada di gawai seperti UC Browser atau Chrome dan lain lain.

Batasan Perangkat Lunak / Sistem

Dalam mengimplementasikan perancangan sistem informasi *SchoolarSeeker* ini ada beberapa hal yang menjadi batasan implementasi, diantaranya yaitu:

1. Aplikasi hanya bisa diakses jika ada koneksi internet, Dengan adanya koneksi internet yang bagus maka web akan bekerja dengan semestinya, sedangkan jika tidak mempunyai koneksi internet maka tampilan web pun tidak akan keluar.
2. Aplikasi hanya bisa diakses menggunakan *browser,* SchoolarSeeker tidak ada di playstore ataupun sejenisnya, kita hanya bisa di akses melalui browser laptop ataupun handphone.

## Asumsi dan Dependensi

Asumsi:

* Sistem *SchoolarSeeker* dapat diakses oleh siapapun dan dari mana saja dengan syarat terhubung dengan jaringan internet.
* *Admin* dapat mengakses dan mengelola data mahasiswa, data pemberi beasiswa, dandata *admin*.
* Mahasiswa dan pemberi beasiswa diharuskan *login* untuk dapat mengakses *website*.
* Data informasi dapat diakses secara *real-time*.
* Pemberi beasiswa dapat melihat peminat mahasiswa yang mengunjungi informasi tersebut.

Dependensi:

* *Admin* harus *login* saat hendak menggunakan Sistem.
* Informasi selalu terus terupdate.
* Mahasiswa harus melakukan *login* untuk mengakses pencarian beasiswa.
* Data informasi beasiswa dipengaruhi oleh pemberi beasiswa, setiap data informasi beasiswa terbaru masuk akan mengganti data beasiswa.
* Admin akan menghapus informasi beasiswa yang sudah melewati batas deadline.

# Deskripsi Rinci Perangkat Lunak

Deskripsi Rinci Perangkat Lunak dijelaskan sebagai berikut:

## Deskripsi Kebutuhan

### Kebutuhan Fungsional

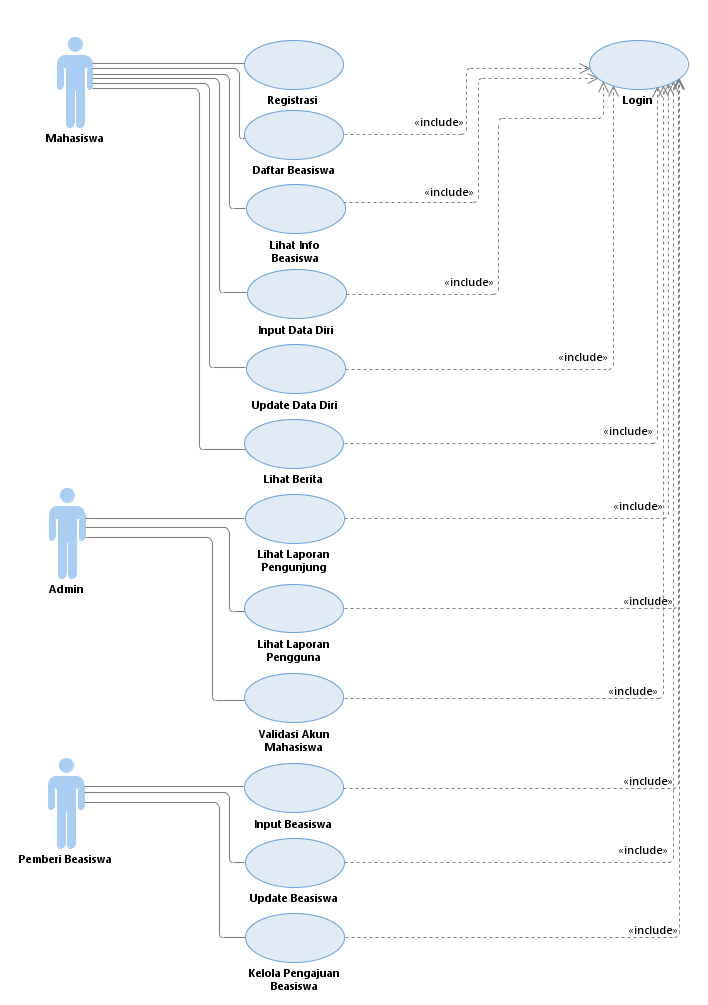
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Kebutuhan | Fungsi | Deskripsi |
| 1. | FR-01 | Info Berita | Fungsi ini digunakan oleh admin untuk menginputkan berita ke sistem |
| 2. | FR-02 | Login | Fungsi ini digunakkan oleh user (mahasiswa, pendonor, admin) untuk masuk ke halaman utama |
| 3. | FR-03 | Register | Fungsi ini digunakan untuk meregistrasi user dan memudahkan saat pelakasaan login |
| 4. | FR-04 | Profil Mahasiswa | Fungsi ini digunakan untuk mahasiswa menjelaskan profil mahasiswa |
| 5. | FR-05 | Profil Penyedia Beasiswa | Fungsi ini digunakan untuk pendonor menambahkan profil penyedia beasiswa |
| 6. | FR-06 | Informasi Beasiswa | Fungsi ini digunakan untuk informasi Beasiswa yang ada di dalam website |
| 7. | FR-07 | Search Beasiswa | Fungsi ini digunakan bagi Mahasiswa yang ingin mencari Beasiswa yang ada di dalam website |
| 8 | FR-08 | Daftar Beasiswa | Fungsi ini merupakan beberapa daftar beasiswa yang sudah ada di dalam website |

### Kebutuhan Non-Fungsional

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Kode Kebutuhan | Deskripsi |
| 1. | Security  Safety | NFR-01 | Fungsi ini digunakan oleh user untuk menginputkan barang ke sistem |
| 2. | Aplikasi Aktif 24/7 | NFR-02 | Aplikasi bisa digunakan jam berapa aja |
| 3. | Responsive | NFR-03 | Aplikasi bersifat responsive dalam penggunaannya |

## Pemodelan Analisis

### Usecase Diagram



#### Usecase Scenario #1

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Registrasi | |
| Aktor | Mahasiswa dan Pendonor | |
| Deskripsi | Mahasiswa dan Pendonor Beasiswa melakukan input data | |
| Pre-Kondisi | Mahasiswa dan Pendonor Beasiswa harus sudah memiliki Email terlebih dahulu | |
| Post-Kondisi | Data direkam dan dimasukan ke DataBase Mahasiswa dan Pendonor Beasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Membuka Website ScholarSeeker |  |
| 2. Mengklik tombol registrasi |  |
|  | 3. Menampilkan form registrasi |
| 4. Mahasiswa mengisi form regist tersebut |  |
| 5. Mengeklik tombol Register |  |
|  | 6. Menyimpan form yang sudah diisi ke database |

#### Usecase Scenario #2

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Daftar Beasiswa | |
| Aktor | Mahasiswa | |
| Deskripsi | Mendaftarkan diri kepada beasiswa tertentu | |
| Pre-Kondisi | Mahasiswa telah login | |
| Post-Kondisi | Mahasiswa telah mendaftar beasiswa dan akan disimpan di database mahasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Membuka halaman beasiswa |  |
|  | 2.Memperlihatkan halaman beasiswa |
| 3.Memilih beasiswa oleh mahasiswa |  |
|  | 3.Mengupdate data beasiswa yang sudah diinput ke database beasiswa |
|  |  |
|  |  |

#### Usecase Scenario #3

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melihat info beasiswa | |
| Aktor | Mahasiswa | |
| Deskripsi | Melihat info beasiswa dengan inputan user | |
| Pre-Kondisi | Jika ingin melihat info beasiswa, mahasiswa sudah melakukan registrasi dan login terlebih dahulu. | |
| Post-Kondisi | Mahasiswa dapat melihat info beasiswa sesuai dengan kata kunci yang diinputkan | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Mahasiswa memasuki halaman web ScholarSeeker |  |
| 2. Mahasiswa memasukan kata kunci yang diinputkan |  |
|  | 3. Memproses pencarian data yang ingin oleh mahasiswa (user) |
|  | 4. Menampilkan beasiswa yang sesuai dengan kata kunci yang diinputkan oleh actor |
| 5. actor dapat melihat hasil pencarian |  |
|  |  |

#### Usecase Scenario #4

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Data Diri | |
| Aktor | Mahasiswa | |
| Deskripsi | Mahasiswa membuka profile lalu melengkapi data diri | |
| Pre-Kondisi | Mahasiswa telah login dan membuka profile mahasiswa | |
| Post-Kondisi | Input data diri disimpan di database Mahasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Mengklik “Profile Mahasiswa” |  |
|  | 2. Memperlihatkan halaman “Profile Mahasiswa” |
| 3. Menginput data sesuai denga form yang muncul lalu mengklik “simpan” |  |
|  | 4. Info data diri akan tersimpan di database Mahasiswa |

#### Usecase Scenario #5

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Update Data Diri | |
| Aktor | Mahasiswa | |
| Deskripsi | Mengupdate data diri | |
| Pre-Kondisi | Mahasiswa telah login dan membuka halaman profile mahasiswa | |
| Post-Kondisi | Data direkam dan mengupdate data di DataBase Mahasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Mengklik “Edit Profile Mahasiswa” pada data mahasiswa yang akan diedit |  |
|  | 2. Memperlihatkan halaman edit |
| 3. Menginput data dan menyelesaikan tindakan dengan mengklik tombol “Edit” |  |
|  | 4. Mengupdate data mahasiswa yang sudah diinput ke database mahasiswa |

#### Usecase Scenario #6

|  |  |  |
| --- | --- | --- |
| Nama Use Case | LihatBerita | |
| Aktor | Mahasiswa | |
| Deskripsi | Melihat daftar berita | |
| Pre-Kondisi | Mahasiswa telah login dan membuka halaman berita | |
| Post-Kondisi | Data direkam di Database Mahasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Mengklik Lihat Berita |  |
|  | 2.Memperlihatkan halaman Lihat Berita |
| 3.Memilih dan membaca berita, dan dapat meng-arcive info beasiswa |  |
|  | 4.Berita yang di-arcive akan disimpan di database mahasiswa |

#### Usecase Scenario #7

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat Laporan Pengunjung | |
| Aktor | Admin | |
| Deskripsi | Admin telah login dan membuka halaman lihat laporan pengunjung dan akan mengirim hasil laporan pengunjung ke pendonor | |
| Pre-Kondisi | Admin telah login | |
| Post-Kondisi | Data direkam dan mengupdate data di Database Admin | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Mengklik halaman lihat laporan pengunjung |  |
|  | 2.Menampilkan halaman lihat pengunjung |
| 3.Mengirim hasil laporan pengunjung ke pendonor |  |
|  | 4.Hasil laporan pengunjung yang dikirim akan disimpan di database |

#### Usecase Scenario #8

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat Laporan Pengguna | |
| Aktor | Admin | |
| Deskripsi | Admin telah login dan membuka halaman lihat laporan pengguna dan akan mengirim hasil laporan pengunjung ke pendonor | |
| Pre-Kondisi | Admin telah login | |
| Post-Kondisi | Data direkam dan mengupdate data di Database Admin | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Mengklik halaman lihat laporan pengguna |  |
|  | 2.Menampilkan halaman lihat pengguna |
| 3.Mengirim hasil laporan pengguna ke pendonor |  |
|  | 4.Hasil laporan pengguna yang dikirim akan disimpan di database |

#### Usecase Scenario #9

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Validasi Akun Mahasiswa | |
| Aktor | Admin | |
| Deskripsi | Admin akan memvalidasi akun sesuai dengan kriteria | |
| Pre-Kondisi | Admin telah melakukan login dan membuka halaman validasi mahasiswa | |
| Post-Kondisi | Data direkam dan mengupdate data di Database Mahasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Memilih halaman validasi mahasiswa |  |
|  | 2.Menampilkan data yang akan divalidasi |
| 3.Mengcheck data mahasiswa yang telah diinput untuk di validasi |  |
|  | 4.Data yang telah divalidasi akan disimpan di database |

#### Usecase Scenario #10

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memasukan data beasiswa | |
| Aktor | Pendonor | |
| Deskripsi | Menambahkan dan menyimpan data informasi beasiswa | |
| Pre-Kondisi | Pemberi Beasiswa harus sudah melakukan registrasi dan login terlebih dahulu | |
| Post-Kondisi | Pemberi beasiswa telah memasukkan informasi beasiwa yang akan ditampilkan di halaman web | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka fitur beasiswa |  |
| 2.Aktor mengklik button tambah beasiswa |  |
|  | 3. Sistem menampilkan form yang harus diisi untuk menambah beasiswa |
| 4. Aktor menginputkan data sesuai dengan form yang telah diberikan oleh sistem |  |
|  | 5. Sistem melakukan pengechekan data dan melakukan validasi data yang telah diinputkan oleh actor |
| 6. Aktor mendapatkan info dari sistema bahwa data informasi beasiswa tervalidasi |  |

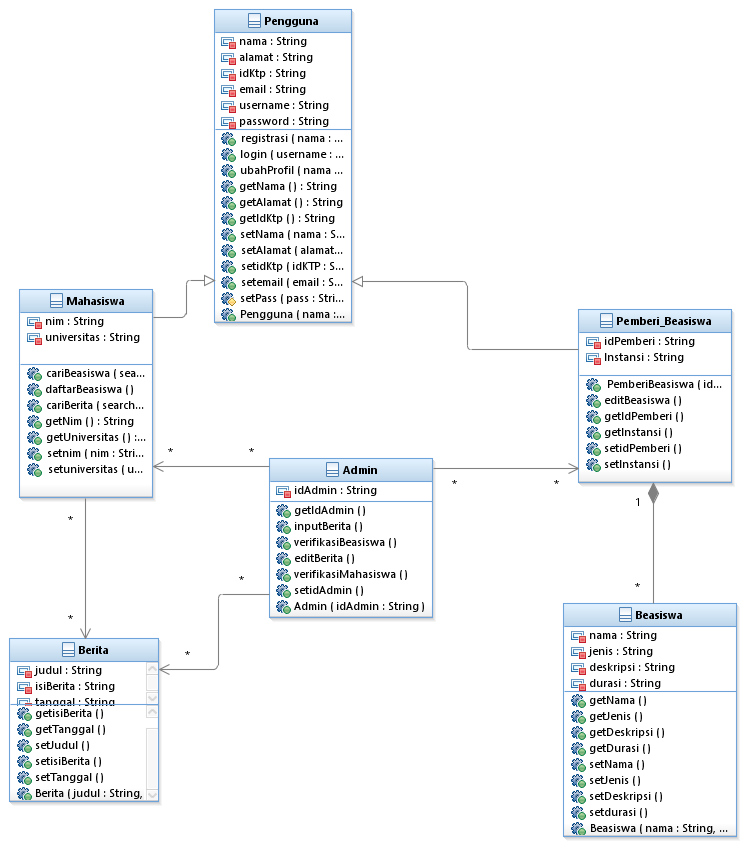
#### Usecase Scenario #11

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Update Beasiswa | |
| Aktor | Pendonor | |
| Deskripsi | Mengedit data beasiswa | |
| Pre-Kondisi | Pendonor telah login | |
| Post-Kondisi | Data direkam dan mengupdate data di DataBase Beasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Mengklik “Edit Beasiswa” pada data beasiswa yang akan diedit |  |
|  | 2.Memperlihatkan halaman edit |
| 3.Menginput data dan menyelesaikan tindakan dengan mengklik tombol “Edit” |  |
|  | 4.Mengupdate data beasiswa yang sudah diinput ke database beasiswa |
|  |  |
|  |  |

#### Usecase Scenario #12

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Kelola Pengajuan Beasiswa | |
| Aktor | Pendonor | |
| Deskripsi | Mengelola Beasiswa | |
| Pre-Kondisi | Pendonor telah login | |
| Post-Kondisi | Data direkam dan mengupdate data di DataBase Beasiswa | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Mengklik “Edit Beasiswa” pada data beasiswa yang akan diedit |  |
|  | 2.Memperlihatkan halaman edit |
| 3.Menginput data dan menyelesaikan tindakan dengan mengklik tombol “Edit” |  |
|  | 4.Mengupdate data beasiswa yang sudah diinput ke database beasiswa |

### Class Diagram:



# Requirements Antarmuka Eksternal

## Antarmuka Pengguna

Aplikasi SchoolarSeeker menggunakan antarmuka berbasis web, dan pengguna dapat mengaksesnya dari gawai maupun laptop yang terhubung dengan jaringan internet. Kebutuhan eksternal dari Sistem SchoolarSeeker ini mencakup Antarmuka pengguna, Antarmuka perangkat keras, dan Antarmuka komunikasi. Antarmuka dari aplikasi SchoolarSeeker mencakup sebagai berikut:

1. Formulir Registrasi Pelamar/Pendaftar Beasiswa

Pelamar dapat melakukan registrasi menggunakan data diri sehingga dapat mengajukan beasiswa yang dipilih.

1. Formulir Login untuk Pengguna(Pelamar, Pendonor dan Pengelola Beasiswa)

Pengguna dapat mengakses aplikasi setelah melakukan login menggunakan *username* dan *password* yang telah dibuat di bagian registrasi.

1. View Beasiswa

Pengguna dapat melihat beasiswa apa saja yang tersedia beserta dengan detail dari beasiswa tersebut dan bisa menggunakan pencarian dengan memasukan kata kunci.

1. View Berita

Pengguna dapat melihat berita tentang layanan pendidikan, beasiswa, dan lomba-lomba di tampilan web bagian berita.

1. CRUD Berita

Terdapat fitur untuk Pengelola sehingga dapat menggunakan Create, Read, Update dan Delete pada berita di dalam web.

1. Formulir pengelolaan Beasiswa

Terdapat fitur untuk Pendonor sehingga dapat mendaftarkan, menghapus dan mengupdate beasiswanya.

1. Formulir pengajuan Beasiswa

Terdapat fitur untuk Pelamar sehingga dapat mendaftarkan diri ke suatu beasiswa dengan mengajukan berkas-berkas yang sudah diisi di registrasi.

1. View statistik pengguna

Terdapat fitur untuk Pengelola sehingga dapat melihat statistik pengguna mulai dari banyak pelamar, penerima beasiswa, pendonor dan pengunjung web.

## Antarmuka Perangkat Keras

Aplikasi SchoolarSeeker service berjalan diatas perangkat keras, seperti:

Laptop atau Komputer :

1. Intel Pentium 4 2,6 GHz, RAM 512MB.
2. Harddisk 80 GB.
3. Windows 2003 Server Professional, PHPTriad 2.21, Apache 1.3.23.

Server database dan server SMS :

1. AMD Athlon XP 2600+ 1,92 GHz, RAM 512MB.
2. Harddisk 120 GB.
3. Windows XP Professional SP 2, Microsoft Visual Studio .NET 2003.
4. DBMS Microsoft SQL Server 2000.

## Antarmuka Perangkat Lunak

Aplikasi SchoolarSeeker yang akan dibangun dengan menggunakan bahasa pemrograman HTML dan CSS yang dapat berjalan di berbagai Web *Browser* mulai dari PC, Laptop ataupun Mobile.

## Antarmuka Komunikasi

SchoolarSeeker merupakan sistem yang terhubung dengan lingkup jaringan internet atau pun intranet yang berbasis protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP). Pengaksesan dapat dilakukan melalui halaman web dengan sarana komputer ataupun gawai yang terhubung dengan jaringan internet atau intranet. Aplikasi ini juga menggunakan *Hypertext Transfer* *Protocol* (HTTP) yang digunakan untuk pengambilan sumber daya yang saling terhubung dengantautan, yang disebut dengan dokumen *Hypertext* yang selanjutnya akan membentuk *World Wide* *Web* atau yang sering disebut dengan WWW.

# Requirements Lain

***Database***

*Database* digunakan untuk menyimpan data pesanan, data *owner*, data kurir, data bahanbaku, dan data produk. *Database* memainkan peran penting dalam sistem ini karena *owner* perlu melihat data pesanan untuk membuat produk sesuai pesanan.

**My SQL**

Untuk mengatur *database*, kita menggunakan RDBMS MySQL karena MySQL merupakan perangkat lunak sumber terbuka (*open-source*). Selain itu MySQL memiliki portabilitas yang cukup baik sehingga bisa digunakan di berbagai sistem operasi seperti Windows, Linux, MacOS, dan Andorid.

**Apache**

Kita menggunakan web server apache karena Apache mengikuti standar protokol HTTP dan memiliki beberapa fitur seperti pesan kesalahan yang dapat diatur, autentifikasi berbasis basis data, dan didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI).

Data diri pelamar dari kampus masing-masing diperlukan untuk memverifikasi keaslian dan ketepatan info agar meminimalisir data palsu.

# Lampiran A: Daftar Kata-Kata Sukar

|  |  |
| --- | --- |
| Istilah,Akronim dan Singkatan | Keterangan |
| SKPL | Spesifikasi Kebutuhan Perangkat Lunak Merupakan dokumen hasil analisis yang berisi kebutuhan user. |
| DFD | Data Flow Diagram , Diagram yang menggambarkan aliran data dan proses yang terjadi di dalam sistem. |
| Admin | Merupakan seseorang yang memiliki serta mengatur fasilitas-fasilitas yang ada. |
| Mahasiwa | Orang yang akan mengakses web *SchoolarSeeker* yang dimana untuk mencari informasi beasiswa. |
| Pemberi Beasiswa | Orang yang mana akan memberikan informasi beasiswa dan akan di tampilkan di halaman web *SchoolarSeeker* |
| Use Case Diagram | Diagram yang mendeskripsikan aksi yang dapat dilakukan oleh sistem dari sudut pandang user |
| Class Diagram | Merupakan diagram yang menggambarkan kelas-kelas yang berada di sistem dan relasi-relasi yang terjadi di kelas tersebut. |

# Lampiran B: Analysis Models

