*Software Requirements Specification (Requirement Analysis Document) and Software Design Description*

for

Aplikasi Kalkulasi Indeks Pembangunan Inklusif (IPI APPS)

Version 2 approved

25 November 2019

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Revisi Minor | 25-11-19 | Permintaan Client | 2.1 |
| Revisi Minor | 25-11-19 | Permintaan Client | 2.2 |
|  |  |  |  |
|  |  |  |  |

**Table Of Contents**

[1. Pendahuluan 4](#_Toc666940907)

[1.1. Tujuan Penulisan Dokumen 4](#_Toc1134171110)

[1.2. Batasan Produk 4](#_Toc1148175658)

[1.3. Definsi dan Istilah 4](#_Toc1608571589)

[2. Deskripsi Keseluruhan 5](#_Toc392854367)

[2.1. Deskripsi dan Fungsi Produk 5](#_Toc1565844898)

[2.2. Penggolongan Karakteristik Pengguna 5](#_Toc1513001197)

[2.3. Lingkungan Operasi 5](#_Toc673641382)

[2.4. Batasan Desain dan Implementasi 6](#_Toc1110313547)

[3. Kebutuhan Antarmuka Eksternal 7](#_Toc1593766257)

[3.1. User Interfaces 7](#_Toc1500725084)

[3.2. Hardware Interfaces 7](#_Toc593186097)

[4. Feature 8](#_Toc727822030)

[5. Software Requirement Spesification 9](#_Toc562494354)

[5.1. Functional Requirements 9](#_Toc424351408)

[5.2. Non Functional Requirements 10](#_Toc230054537)

[6. Software Design Description 11](#_Toc1149110551)

[6.1. Use Case Diagram 11](#_Toc1275712649)

[6.2. Spesifikasi Use Case Scenario 12](#_Toc724918350)

[6.2.1. Use Case Scenario Update Data 12](#_Toc529877897)

[6.2.2. Use Case Scenario Input Tahun 13](#_Toc261882101)

[6.2.3. Use Case Scenario HapusTahun 14](#_Toc440189060)

[6.2.4. Use Case Scenario Melihat](#_Toc432364884) *[Report](#_Toc432364884)* [Data IPI Asli 15](#_Toc432364884)

[6.2.5. Use Case Scenario Tambah User 16](#_Toc1135190290)

[6.2.6. Use Case Scenario Ubah User 17](#_Toc695064961)

[6.2.7. Use Case Scenario Hapus User 19](#_Toc32528232)

[6.3. Class Diagram 21](#_Toc836827600)

# Pendahuluan

## Tujuan Penulisan Dokumen

Tujuan dokumen SRS (Software Requirement Specification) dan RAD (*Requirement Analysis Document)* ini dibuat untuk:

1. Mendefinisikan dan memperjelas spesifikasi kebutuhan dan ruang lingkup operasional aplikasi perhitungan Indeks Pembangunan Inklusif (IPIAPPS) yang akan dikembangkan.
2. Menggambarkan dan mendefinisikan secara global aplikasi perhitungan Indeks Pembangunan Inklusif (IPIAPPS) yang akan dikembangkan.
3. Mempermudah dalam proses pengembangan lanjutan aplikasi perhitungan Indeks Pembangunan Inklusif (IPIAPPS).

Oleh karena tujuan tersebut, maka pihak yang berhak menggunakan dokumen ini adalah:

1. Pengembang aplikasi yang akan menggunakan dokumen ini sebagai pedoman dalam

pengembangan aplikasi perhitungan Indeks Pembangunan Inklusif (IPIAPPS).

1. Pengguna aplikasi yang akan menggunakan dokumen ini untuk memvalidasi kebutuhan

yang akan diimplementasikan oleh pengembang aplikasi.

## Batasan Produk

Batasan yang digunakan pada pengembangan sistem ini adalah sebagai berikut:

1. Sistem hanya dapat dijalankan dan ditujukan pada daerah Kabuptan Gresik
2. Sistem yang dikembangkan berbasis web
3. Sistem hanya dapat melakukan perhitungan Indeks Pembangunan Inklusif

## Definsi dan Istilah

* SRS : *Software Requirements Specification*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* RAD : *Requirement Analysis Document*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* SDD : *Software Design Description,*  atau

Deskripsi Perancangan Perangkat Lunak (DPPL)

* RDBMS : *Relational Database Management System*
* RAM : *Random Access Memory*
* OS : *Operating System*
* Mysql : *My Structured Query Languages*

# Deskripsi Keseluruhan

## Deskripsi dan Fungsi Produk

Indeks Pembangunan Ekonomi Inklusif digunakan sebagai alat untuk mengukur tingkat inklusivitas pembangunan di suatu daerah. IPI APPS merupakan sebuah aplikasi yang mengkalkulasi Indeks Pembangunan Inklusif sesuai data yang ada per tahun dan menampilkannya dalam bentuk Tabel sekaligus Grafik. Sistem ini ditujukkan di suatu daerah tertentu, dan untuk studi kasus ini ditujukkan untuk Kabupaten Gresik. Sistem ini akan dikembangkan dalam basis web yang dapat diakses melalui berbagai perangkat seperti PC, laptop, dan smartphone. Sistem ini diharapkan dapat memudahkan untuk menghitung Indeks Pembangunan Inklusif.

## Penggolongan Karakteristik Pengguna

**Tabel 2.1. Daftar Tipe Pengguna Beserta Karakteristiknya**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Tipe Pengguna** | **Karakteristik** | **Pengguna** |
| 1. | Pengguna Standar | * Pegawai pemerintahan di divisi utama yang bekerja dengan standar yang jelas. * Menggunakan teknologi dalam menyelesaikan tugasnya. | Administrator |

**Tabel 2.2. Daftar Kategori Pengguna Beserta Tugas dan Hak Akses**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kategori Pengguna** | **Tugas** | **Hak Akses Ke Aplikasi** |
| 1. | Administrator | * Menambahkan dan menghapus data * Melihat hasil *report* data asli * Menambahkan dan menghapus tahun perhitungan * Menambah, Menghapus, dan Mengedit user | *Insert, Delete, Update, Read* |

## Lingkungan Operasi

IPI APPS dapat berjalan dengan baik dan stabil pada lingkungan operasi yang mendukung pada sisi server dan client dengan spesifikasi sebagai berikut.

1. Spesifikasi Server
   1. Server: Mysql, Apache Web Server
   2. OS Server: Linux
   3. RDBMS: Mysql
   4. RAM : 8 GB
2. Spesifikasi Client
   1. Client: Web Browser (Google, Mozilla Firefox, Opera Mini, UC Browser atau lainnya)
   2. OS Client: Laptop (Windows atau Linux), Handphone (Android)

## Batasan Desain dan Implementasi

Batasan dalam desain dan implementasi sistem IPIAPPS adalah sebagai berikut:

* Bahasa pemrograman yang digunakan adalah CSS, Mysql, PHP, HTML, Java Script
* Web server yang digunakan Apache web server
* Protokol komunikasi yang digunakan HTTPS

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Sistem IPI APPS membutuhkan interaksi dengan pengguna. Oleh karena itu, agar pengguna dapat berinteraksi dengan sistem IPIAPPS ini dibutuhkan beberapa perangkat yang dapat memfasilitasi pengguna dalam memberi input kepada aplikasi dan dapat menampilkan output dari proses aplikasi. Perangkat yang dibutuhkan sebagai berikut:

* + - 1. Keyboard dan Mouse

Perangkat ini digunakan pengguna untuk memberi input kepada sistem berupa data. Spesifikasi perangkat ini adalah jenis yang kompatibel untuk sistem operasi Microsoft Windows98/XP.

2. Monitor

Perangkat ini digunakan sebagai sarana untuk menampilkan aplikasi kepada pengguna. Spesifikasi untuk perangkat ini adalah mampu menampilkan grafis dengan kualitas warna yang baik agar dapat menampilkan grafik.

## Hardware Interfaces

Sistem IPIAPPS membutuhkan antarmuka perangkat keras sebagai berikut:

Graphic card

Kapasitas yang dibutuhkan untuk perangkat ini adalah sebesar 4 Mbyte agar perangkat lunak dapat berjalan dengan baik.

Ethernet card dan modem

Perangkat ini digunakan untuk menghubungkan laptop yang menjalankan perangkat lunak dengan jaringan internet

# Featu**re**

**Tabel 4.1 Fitur IPI APPS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode Fitur** | **Nama Fitur** | **Deskripsi** | **Prioritas** |
| FTR-01 | Data IPI | Sistem dapat memfasilitasi dalam mengubah data IPI | MO |
| FTR-02 | Tahun | Sistem dapat memfasilitasi dalam menambahkan, serta menghapus Tahun | MO |
| FTR-03 | *Report* Data | Sistem dapat memfasilitasi untuk menampilkan *report* data asli | MO |
| FTR-04 | *Availabilty* | Sistem harus dapat beroperasi tujuh hari perminggu, 24 jam per hari tanpa gagal | MO |
| FTR-05 | *Portability* | Sistem dapat berjalan pada platform atau sistem operasi yang mendukung aplikasi berbasis web | MO |
| FTR-06 | Data Grafik | Sistem harus dapat menampilkan data grafik pada hasil *re-scale* IPI | MO |
| FTR-07 | Manajemen User | Sistem harus dapat melakukan lihat, tambah, hapus dan edit user | MO |

# Software Requirement Spesification

## 5.1. Functional Requirements

**Tabel 5.1 Kebutuhan Fungsional Sistem**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode Fitur** | **Kode Fungsional Dasar** | **Kode Lengkap Fungsi** | **Nama Fungsi** | **Deskripsi** |
| FTR-01 | IPIAPPS -FL-01 | IPIAPPS -FL-01-01 | Mengubah Data IPI | Sistem mampu menyediakan fungsi untuk mengubah data IPI |
| FTR-02 | IPIAPPS -FL-02 | IPIAPPS -FL-02-01 | Menambahkan Tahun | Sistem mampu menyediakan fungsi untuk menambahkan tahun |
| FTR-02 | IPIAPPS -FL-03 | IPIAPPS -FL-03-01 | Menghapus Tahun | Sistem mampu menyediakan fungsi untuk menghapus tahun |
| FTR-03 | IPIAPPS-FL-04 | IPIAPPS-FL-04-01 | Melihat *Report* Data IPI Asli | Sistem mampu menyediakan fungsi untuk menampilkan *report* data IPI asli |
| FTR-06 | IPIAPPS-FL-05 | IPIAPPS-FL-05-01 | Melihat data grafik | Sistem harus dapat menampilkan data re-scale dalam bentuk grafik batang |
| FTR-07 | IPIAPPS-FL-06 | IPIAPPS-FL-06-01 | Melihat daftar user | Sistem harus dapat menampilkan daftar user dalam bentuk tabel |
| FTR-07 | IPIAPPS-FL-07 | IPIAPPS-FL-07-01 | Menghapus data user | Sistem harus dapat menghapus data user |
| FTR-07 | IPIAPPS-FL-08 | IPIAPPS-FL-08-01 | Memperbaharui data user | Sistem harus dapat memperbaharui data user |
| FTR-07 | IPIAPPS-FL-09 | IPIAPPS-FL-09-01 | Menambah data user | Sistem harus dapat menambahkan data user |

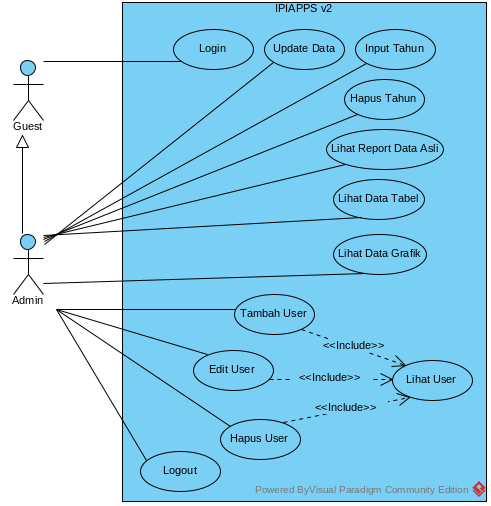
## 5.2. Non Functional Requirements

**Tabel 5.2 Kebutuhan Non Fungsional Sistem**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode Fitur** | **Kode Fungsional Dasar** | **Kode Lengkap Fungsi** | **Nama Fungsi** | **Deskripsi** |
| FTR-04 | IPIAPPS-NFL-01 | IPIAPPS-NFL-01-01 | *Availability* | Sistem harus dapat beroperasi tujuh hari perminggu, 24 jam per hari tanpa gagal |
| FTR-05 | IPIAPPS-NFL-02 | IPIAPPS-NFL-02-01 | Portability | Sistem dapat berjalan pada platform atau sistem operasi yang mendukung aplikasi berbasis web |

# Software Design Description

## 6.1. Use Case Diagram



**Gambar 6.1 Use Case Diagram IPI Apps**

## 6.2. Spesifikasi Use Case Scenario

### 6.2.1. Use Case Scenario Update Data

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-01 |
| ***Use Case Name*** | Update Data |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses peng*update*-an data yang dilakukan admin. |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS 2. Aktor telah masuk ke dalam sistem IPIAPPS |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu [Input Data] 2. Sistem menampilkan halaman Input Data 3. Aktor memilih menu [Dimensi], [Sub Dimensi], [Indikator], dan [Tahun] (Dalam bentuk *dropdown*) 4. Aktor menginputkan data 5. Aktor mengganti nominal pada kolom [Nilai] yang ditampilkan dari hasil penginputan data Dimensi, Sub Dimensi, dan Indikator 6. Aktor menekan tombol [Perbaharui Data] |
| ***Alternative Flows*** | * *Alert*(‘Harap memasukkan nilai data), jika data kosong tidak diinputkan * Jika tidak ada input dimensi, sub dimensi, indikator dan tahun, menampilkan *alert(*‘harap tentukan pilihan dimensi, sub dimensi, indikator dan tahun’) |
| ***Sub-Flow*** | Tidak ada |
| ***Post-Condition*** | 1. Sistem menampilkan *alert* bahwa data berhasil ditambahkan |

**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Input Data] |  |
|  | 1. Menampilkan halaman Input Data |
| 1. Menginputkan data dari *dropdown* yang diberikan |  |
|  | 1. Menampilkan nilai dari hasil dropdown yang diinputkan |
| 1. Mengubah nominal [Nilai] |  |
| 1. Menekan [Perbaharui Data] |  |
|  | 1. Menyimpan dan memperbaharui data |
|  | 1. Menampilkan Flash Message “Data Berhasil Diperbaharui” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 3.a. Data input tidak lengkap |  |
|  | 7.a. Data tidak tersimpan |

### 6.2.2. Use Case Scenario Input Tahun

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-02 |
| ***Use Case Name*** | Input Tahun |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses penambahan Tahun untuk input data IPI yang dilakukan admin. |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS 2. Aktor telah masuk ke dalam sistem IPIAPPS |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu [Input Data] 2. Aktor menekan tombol Tahun di halaman Input Data 3. Sistem menampilkan *popup* / *modal* untuk input tahun 4. Aktor memasukkan nilai tahun 5. Sistem menambahkan tahun ke dalam sistem |
| ***Alternative Flows*** | * *Alert*(“Harap memasukkan nilai Tahun”), jika data kosong tidak diinputkan |
| ***Sub-Flow*** | Tidak ada |
| ***Post-Condition*** | 1. Sistem menampilkan alert bahwa tahun telah ditambahkan |

**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Input Data] |  |
|  | 1. Menampilkan halaman Input Data |
| 1. Menekan [+ Tahun] |  |
|  | 1. Menampilkan modal Tambah Tahun |
| 1. Memasukkan [Tahun] |  |
| 1. Menekan [Tambah Tahun] |  |
|  | 1. Menambahkan tahun ke dalam sistem |
|  | 1. Menampilkan Flash Message “Tahun Berhasil Ditambahkan” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 5.a. Form tahun tidak diisi |  |
|  | 8.a. Menampilkan alert “Harap masukkan nilai tahun” |

### 6.2.3. Use Case Scenario HapusTahun

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-03 |
| ***Use Case Name*** | Hapus Tahun |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses penghapusan Tahun untuk input data IPI yang dilakukan admin. |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS |
| ***Basic Flow/Flow of Events*** | 1. Aktor membuka halaman [Input Data] 2. Sistem menampilkan halaman Input Data 3. Aktor menekan tombol [Hapus Tahun] 4. Sistem menampilkan *popup* / *modal* untuk input tahun 5. Aktor memilih nilai tahun untuk dihapus dari *dropdown* 6. Sistem menghapus tahun dan seluruh data IPI dalam tahun tersebut |
| ***Alternative Flows*** | * *Alert* (“Harap memasukkan nilai tahun”), jika data kosong tidak diinputkan |
| ***Sub-Flow*** | Tidak ada |
| ***Post-Condition*** | 1. Sistem menampilkan alert bahwa tahun telah dihapus |

**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Input Data] |  |
|  | 1. Menampilkan halaman Input Data |
| 1. Menekan [Hapus Tahun] |  |
|  | 1. Menampilkan modal Hapus Tahun |
| 1. Memilih Tahun dari *dropdown* |  |
| 1. Menekan [Hapus Semua di Tahun] |  |
|  | 1. Menghapus tahun dan seluruh data IPI dalam tahun tersebut |
|  | 1. Menampilkan Flash Message “Tahun Berhasil Dihapus” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 5.a. Form tahun tidak diisi |  |
|  | 8.a. Menampilkan alert “Harap masukkan nilai tahun” |

### 6.2.4. Use Case Scenario Melihat *Report* Data IPI Asli

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-04 |
| ***Use Case Name*** | Melihat *Report* Data IPI Asli |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses melihat *report* keseleruhan  data IPI |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu ‘Report’ 2. Sistem menampilkan seluruh *report* data IPI |
| ***Alternative Flows*** |  |
| ***Sub-Flow*** | Tidak Ada |
| ***Post-Condition*** | Sistem menampilkan seluruh *report* data IPI |

**Main Flow**

**Melihat Report Data IPI Asli**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu ‘Report’ |  |
|  | 1. Menampilkan halaman Report |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
|  |  |

### 6.2.5. Use Case Scenario Tambah User

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-05 |
| ***Use Case Name*** | Tambah User |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses menambah user IPIAPPS |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS 2. Aktor telah masuk ke dalam sistem IPIAPPS |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu [Manajemen User] 2. Sistem menampilkan halaman Manajemen User 3. User memilih [Tambah User] 4. Sistem menampikan halaman Tambah User 5. User mengisi [Username], [Password], [Masukkan ulang password], dan [Status user] 6. User menekan [+ Tambah User] 7. Sistem menyimpan data user baru |
| ***Alternative Flows*** | * *Alert* “(Harap mengisi field ini”) bila ada field yang tidak diisi |
| ***Sub-Flow*** | Tidak Ada |
| ***Post-Condition*** | Sistem menampilkan alert bahwa user telah ditambahkan |

**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Management User] |  |
|  | 1. Menampilkan halaman Management User |
| 1. Menekan [+ Tambah User] |  |
|  | 1. Menampilkan halaman Tambah User |
| 1. Mengisi [Username], [Password], [Masukkan ulang password], dan [Status user] |  |
| 1. Menekan [+ Tambah User] |  |
|  | 1. Menyimpan data user baru |
|  | 1. Menampilkan alert “User berhasil ditambahkan” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 5.a. Form tidak lengkap |  |
|  | 8.a. Menampilkan alert “Harap mengisi field berikut” |

### 6.2.6. Use Case Scenario Ubah User

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-06 |
| ***Use Case Name*** | Hapus User |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses mengedit/mengubah akun user IPIAPPS |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS 2. Aktor telah masuk ke dalam sistem IPIAPPS 3. Sistem memiliki minimal satu akun pengguna |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu [Manajemen User] 2. Sistem menampilkan halaman Manajemen User 3. User memilih [Ubah] pada salah satu akun 4. Sistem menampikan halaman Ubah User 5. User mengisi [Username], [Password], [Masukkan ulang password], dan [Status user] 6. User menekan [+ Edit User] 7. Sistem mengupdate data user baru |
| ***Alternative Flows*** | * Tidak ada |
| ***Sub-Flow*** | *Alert* “(Harap mengisi field ini”) bila ada field yang tidak diisi |
| ***Post-Condition*** | Sistem menampilkan alert bahwa user telah diubah |

**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Management User] |  |
|  | 1. Menampilkan halaman Management User |
| 1. Menekan [Ubah] |  |
|  | 1. Menampilkan halaman Ubah User |
| 1. Mengisi [Username], [Password], [Masukkan ulang password], dan [Status user] |  |
| 1. Menekan [+ Edit User] |  |
|  | 1. Menupdate data user baru |
|  | 1. Menampilkan alert “User berhasil diubah” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 5.a. Form tidak lengkap |  |
|  | 8.a. Menampilkan alert “Harap mengisi field berikut” |

### 6.2.7. Use Case Scenario Hapus User

|  |  |
| --- | --- |
| ***Use Case Code*** | IPIAPPS -FL-07 |
| ***Use Case Name*** | Hapus User |
| ***Actor*** | Admin |
| ***Brief Description*** | *Use case* ini menggambarkan proses menghapus akun user IPIAPPS |
| ***Pre-Condition*** | 1. Aktor telah melakukan *login* ke sistem IPIAPPS 2. Aktor telah masuk ke dalam sistem IPIAPPS 3. Sistem memiliki minimal satu akun pengguna |
| ***Basic Flow/Flow of Events*** | 1. Aktor memilih menu [Manajemen User] 2. Sistem menampilkan halaman Manajemen User 3. User menekan [Hapus] pada salah satu akun 4. Sistem menampikan modal konfirmasi 5. User memilih [Hapus] 6. Sistem menghapus akun |
| ***Alternative Flows*** | * Tidak ada |
| ***Sub-Flow*** | Tidak Ada |
| ***Post-Condition*** | Sistem menampilkan alert bahwa user telah dihapus |

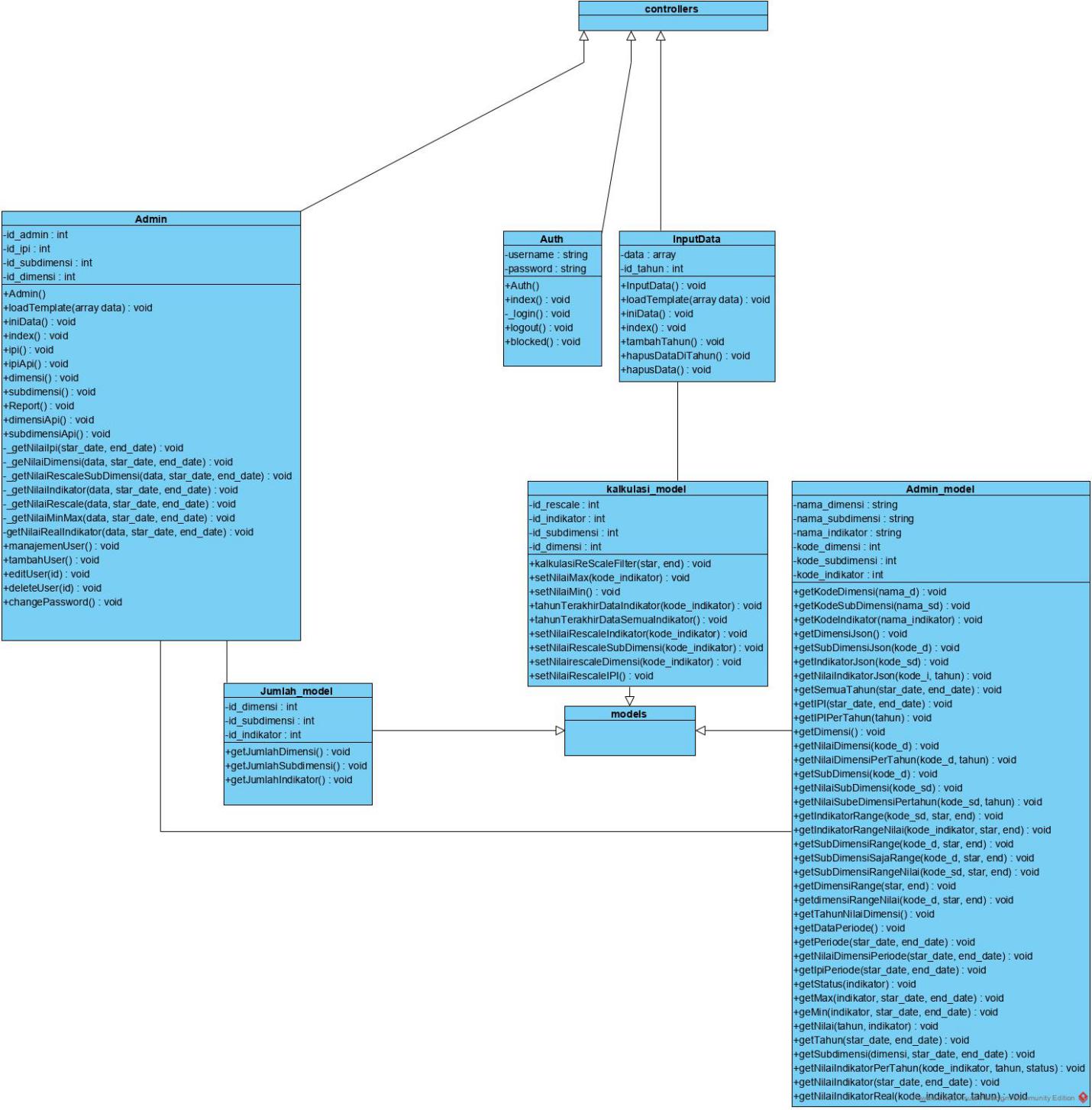
**Main Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Memilih menu [Management User] |  |
|  | 1. Menampilkan halaman Management User |
| 1. Menekan [Hapus] |  |
|  | 1. Menampilkan modal konfirmasi |
| 1. Menekan [Hapus] |  |
|  | 1. Sistem menghapus data user |
|  | 1. Menampilkan alert “User berhasil dihapus” |

**Alternative Flow**

|  |  |
| --- | --- |
| Action by user | Response from system |
|  |  |

## 6.3. Class Diagram



**Gambar 6.2 Class Diagram Sistem**