|  |
| --- |
| Untitled  **Software Requirement Spesification**  **e-Reservations**  **Peminjaman Fasilitas UII**  **Disiapkan oleh:**  **Rahmat Nurhabibie (16523029)**  **Ivan Firmansyah (16523036)**  **Fikri Azizi (16523054)**  **Muh Nizomuddin Fauza Sidiq (16523071)**  **Saputro Andi Pamungkas (16523189)**  **FAKULTAS TEKNOLOGI INDUSTRI**  **UNIVERSITAS ISLAM INDONESIA**  **YOGYAKARTA**  **Maret, 2017** |

DAFTAR ISI

[**DAFTAR ISI 1**](#_gjdgxs)

[**BAB I.**](#_30j0zll) **Deskripsi Sistem dan Perangkat Lunak 3**

[**1.1**](#_1fob9te) **Nama Perangkat Lunak 3**

[**1.2**](#_3znysh7) **Deskripsi Perangkat Lunak 3**

[**1.3**](#_2et92p0) **Domain dan Lingkungan Sistem 3**

[**BAB II.**](#_tyjcwt) **Analisis Kebutuhan 5**

[**2.1**](#_3dy6vkm) **Kebutuhan Fungsionalitas 5**

[**2.2**](#_2s8eyo1) **Kebutuhan Non Fungsionalitas 5**

[**2.3**](#_3rdcrjn) **Identifikasi Aktor 6**

[**2.4**](#_35nkun2) **Identifikasi *Use Case* 6**

[***2.5***](#_2jxsxqh) ***Use Case Diagram* 7**

[**BAB III.**](#_1y810tw) **Permodelan Analisis 9**

[**3.1**](#_4i7ojhp) **Skenario Aktivitas 9**

[**3.1.1**](#_2xcytpi) **PUC-01 *Booking* Ruangan 9**

[**3.1.2**](#_1ci93xb) **PUC-02 Mengecek Data Ruangan dan Fasilitas 10**

[**3.1.3**](#_3whwml4) **PUC-03 Memberikan Kritik dan Saran 10**

[**3.1.4**](#_qsh70q) **PUC-04 Mengelola Data Ruangan 11**

[**3.1.5**](#_3as4poj) **PUC-05 Mengelola Data Pengguna 12**

[**3.2**](#_1pxezwc) **Diagram Kelas Sederhana 13**

[**3.2.1**](#_49x2ik5) **PUC-01** [**Memboking Ruangan**](#_2xcytpi) [**13**](#_49x2ik5)

[**3.2.2**](#_2p2csry) **PUC-02** [**Mengecek Data Ruangan dan Fasilitas**](#_1ci93xb) [**13**](#_2p2csry)

[**3.2.3**](#_147n2zr) **PUC-03** [**Memberikan Kritik dan Saran**](#_3whwml4) [**13**](#_147n2zr)

[**3.2.4**](#_23ckvvd) **PUC-04** [**Mengelola Data Ruangan**](#_qsh70q) [**14**](#_23ckvvd)

[**3.2.5**](#_ihv636) **PUC-05 Mengelola Data Pengguna 14**

# Deskripsi Sistem dan Perangkat Lunak

Pembangunan perangkat lunak yang dijelaskan pada bab ini merupakan proses rekayasa perangkat lunak (RPL) terhadap *software* Prototipe Peminjaman Fasilitas UII yang dikembangkan pada penelitian ini. Dalam pelaksanaanya, metodologi pembangunan perangkat lunak yang digunakan adalah metodologi *waterfall* sedangkan pendekatan yang digunakan adalah pendekatan berorientasi obyek. Dengan demikian, dokumentasi yang dihasilkan ini diharapkan dapat menyerupai hasil *generate* dari *software* Prototipe Peminjaman Fasilitias UII setelah digunakan.

## Nama Perangkat Lunak

“**e-Reservations**. Perangkat Lunak untuk *booking* Ruangan dan fasilitas di lingkup UII.”

## Deskripsi Perangkat Lunak

**e-Reservations,** atau cukup disebut **e-Reserv** adalah sebuah *software* yang akan memudahkan *users* untuk melakukan *booking* ruangan dan fasilitas di UII. *Users*  yang dimaksudkan di sini adalah *Civitas Akademika UII* ataupun *Guest* yang biasanya kesulitan dalam memesan ruangan dan fasilitas di UII . Dengan *softwarei* ini, para *users* dapat melakukan *booking* ruangan dan fasilitas dengan mudah dan cepat. Untuk kemudahan *Guest* dan *Civitas akademika UII*, **e-Reservations** dikembangakan sebagai sebuah *software* berbasis *desktop*.

## Domain dan Lingkungan Sistem

Perangkat lunak **e-Reservations** dibangun dalam domain pendidikan. Lingkungan sistem dari perangkat lunak **e-Reservations** adalah Universitas Islam Indonesia.

## Pemangku Kepentingan

Pemangku kepentingan adalah pihak-pihak yang memiliki kepentingan di dalam proyek perangkat lunak ini, contoh nya adalah pemilik sistem, pemilik modal, pelanggan, users dan lain-lain.

## Proses Bisnis

Suatu kegiatan yang memiliki rantai kegiatan terstruktur, untuk mengatur dan mengorganisasi segala aktifitas dan digunakan untuk meningkatkan hubungan timbal balik antar pihak yang terlibat dalam proses bisnis (produsen, konsumen, distributor).   
 Proses bisnis adalah banyaknya aktifitas yang akan mengkoordinasikan teknik pengorganisasian untuk mencapai suatu tujuan yang di setiap prosesnya akan memiliki aturan sendiri yang sudah terstruktur, tetapi tidak menutup kemungkinan bahwa proses tersebut akan berpengaruh terhadap proses yang lain.

# Analisis Kebutuhan

## Kebutuhan Fungsionalitas

Dari analisis permasalahan yang telah dilakukan, maka diperoleh enam hal utama yang dibutuhkan dalam sebuah *software* e-Reservations. Tabel 1 memperlihatkan kebutuhan fungsioanlitas dari *software* e-Reservations, sebuah *software* peminjaman fasilitas UII.

**Tabel 1. Kebutuhan fungsionalitas *software* e-Reservations**

|  |  |
| --- | --- |
| **Kode** | **Deskripsi** |
| PKF-01 | *Software* harus dapat mempermudah *users* dalam mereservasi ruangan di UII |
| PKF-02 | *Software* harus menyediakan fasilitas untuk mengecek data fasilitas ruangan. |
| PKF-03 | *Software* harus dapat menyimpan data ruangan yang tersedia dan tidak tersedia maupun yang sedang digunakan. |
| PKF-04 | *Software* harus menyediakan fasiltas untuk melakukan review terhadap fasilitas ruangan yang telah dibuat sebelumnya. |
| PKF-05 | *Software* harus menyediakan fasilitas untuk memberikan kritik atau saran jika diperlukan. |
| PKF-06 | *Software* harus dapat mengelola pengguna, mengelompokkannya sebagai dosen ataupun sebagai mahasiswa beserta mengelola hak aksesnya. |

## Kebutuhan Non Fungsionalitas

Tabel 2 menunjukkan kebutuhan non fungsionalitas dari software e-Reservations ini. Empat kebutuhan pendukung *software* peminjaman fasilitas UII ini juga diperoleh dari hasil analisis permasalahan sebelumnya.

**Tabel 2. Kebutuhan non fungsionalitas *software* e-Reservations**

|  |  |
| --- | --- |
| **Kode** | **Deskripsi** |
| PKN-01 | *Software* harus memiliki usabilitas yang tinggi sehingga mudah digunakan oleh *users*. |
| PKN-02 | *Software* harus memiliki tampilan yang sederhana, dengan form isian yang ada juga sederhana sehingga penggunaan dan akses dapat dilakukan dengan cepat. |
| PKN-03 | *Software* harus menyediakan informasi yang jelas terkait data ruangan yang diperlukan untuk mempermudah *users* menentukan ruangan yang dibutuhkan. |
| PKN-04 | *Software* harus dapat membatasi hak akses setiap aktor denga proses autentifikasi. |

## Identifikasi Aktor

Dari kebutuhan fungsionalitas yang telah disampaikan sebelumnya, maka diidentifikasi empat aktor dalam *software* e-Reservations. Tabel 3 menjelaskan keempat aktor dalam *software* peminjaman fasilitas UII ini.

**Tabel 3. Daftar aktor *software* e-Reservations**

|  |  |
| --- | --- |
| **Nama Aktor** | **Deskripsi** |
| *Users* | Pengguna yang diberi hak untuk *booking* dan mengecek ruangan beserta fasilitas didalamnya. |
| Admin | Pengguna yang mengelola data ruangan , data *users* dan mengelola kritik dan saran. |
| Rektorat | Pengguna yang mengelola kritik dan saran |

## Identifikasi *Use Case*

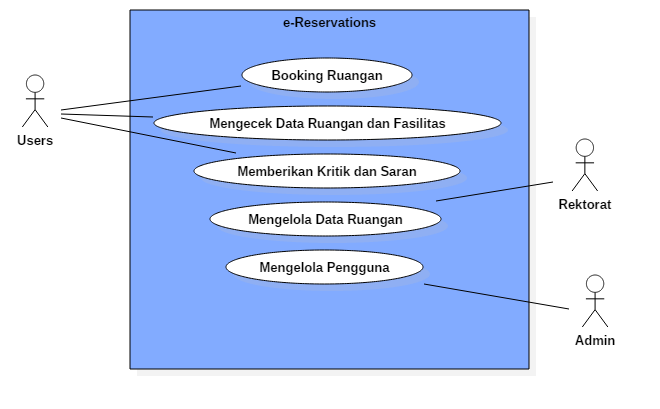
Tabel 4 menjelaskan lima *use case* dari *software* **e-Reservations**. Lima *use case* tersebut dipetakan dari enam kebutuhan fungsionalitas software peminjaman fasilitas UII ini.

**Tabel 4. Daftar *use case software* e-Reservations**

|  |  |
| --- | --- |
| **Kode** | **Nama & Deskripsi** |
| PUC-01 | *Booking* ruangan  **Deskripsi**: *Users* dapat melakukan *booking* ruangan dan fasilitas yang tersedia di UII. |
| PUC-02 | Mengecek fasilitas ruangan  **Deskripsi**: *Users* dapat mengecek informasi ruangan dan fasilitas yang ada di UII , sehingga *users* dapat menentukan ruangan yang akan di-*booking*. |
| PUC-03 | Memberikan kritik dan saran  **Deskripsi**: *Users* dapat memberikan kritik dan saran terhadap ruangan dan fasilitas yang ada di UII. |
| PUC-04 | Mengelola data ruangan  **Deskripsi:** Admin dapat mengubah data ruangan jika sewaktu-waktu fasilitas yang ada di ruangan tersebut di tambahkan maupun di kurangi. |
| PUC-05 | Mengelola pengguna  **Deskripsi**: Admin dapat mengelola pengguna *software*, termasuk menambah, mengubah, menghapus dan menentukan jenis pengguna. |

## *Use Case Diagram*

Dari keempat aktor dan kelima *use case* yang telah diidentifikasi sebelumnya, maka dapat dimodelkan sebuah diagram *use case* yang memperlihatkan interaksi antara aktor dan *use case* dalam sistem e-Reservations. Gambar 3 memperlihatkan diagram *use case* untuk *software* peminjaman fasilitas UII ini.



**Gambar 3. *Use case diagram software* e-Reservations**

# Permodelan Analisis

## Skenario Aktivitas

Berikut skenario aktivitas yang akan terjadi dalam setiap *use case* (UC) pada *software* pemijaman fasilitas UII:

### PUC-01 *Booking* Ruangan

Aktor: *Users*

Prerequisit: Aktor sudah memilih menu *booking* ruangan dan fasilitas.

1. Sistem menampilkan syarat peminjaman ruangan dan fasilitas dari basis data.
2. Aktor mengisi form verifikasi syarat peminjaman ruangan dan fasilitas.
3. Aktor memilih tombol lanjut
4. Sistem mengirimkan data verifikasi ke dalam basis data.
5. Sisten menampilkan halaman informasi bahwa peminjaman sedang di proses.
6. Sistem memberikan notifikasi adanya permintaan peminjaman ruangan dan fasilitas ke Admin.

### PUC-02 Mengecek Data Ruangan dan Fasilitas

Aktor: *Users*

Prerequisit: Aktor sudah memilih menu informasi ruangan dan fasilitas

1. Sistem mengambil data ruangan dan fasilitas dari basis data
2. Sistem menampilkan daftar informasi ruangan dan fasilitas ke layar

### PUC-03 Memberikan Kritik dan Saran

Aktor: *Users*

Prerequisit: Aktor sudah memilih menu kritik dan saran.

1. Sistem menampilkan *form* kritik dan saran.
2. Aktor mengisi *form* kritik dan saran.
3. Aktor menekan tombol *Submit*.
4. Sistem menyimpan data kritik dan saran tersebut ke basis data.
5. Sistem mengirim notifikasi kritik dan saran baru ke Admin.

### PUC-04 Mengelola Data Ruangan

Aktor: Admin

Prerequisit: Aktor sudah login

1. Sistem mengambil data ruangan dan fasilitas dari basis data.
2. Sistem menampilkan daftar data ruangan dan fasilitas yang ingin di-*booking* oleh *user* ke layar.
3. Admin memberikan persetujuan jika syarat peminjaman dari *user* sudah terpenuhi
4. Sistem menampilkan status *booked* di bagian status ruangan
5. Admin bisa mengubah data fasilitas yang ada dalam ruangan jika sewaktu-waktu ada perubahan fasilitas di dalam ruangan.
6. Sistem menampilkan data ruangan dan fasilitas dari basis data
7. Admin menekan tombol *edit*
8. Admin akan menindaklanjutilaporan yang masuk
9. Admin melakukan perubahan status pada ruangan yang telah di *booking*
10. Admin menekan tombol *update*
11. Sistem menampilkan data ruangan dan fasilitas yang telah di-*update*

### PUC-05 Mengelola Pengguna

Aktor: Admin

Prerequisit: Aktor telah *login* dan masuk ke menu mengelola pengguna

1. Sistem mengambil data pengguna dari basis data.
2. Sistem menampilkan daftar pengguna ke layar.
3. Sistem menampilkan fom Pengguna Baru ke layar.
4. *Guests* mengisi form pengguna baru.
5. *Guests* menekan tombol ok.
6. Sistem menyimpan data pengguna baru ke basis data.
7. Sistem menampilkan halaman pemesanan berhasil.

## *Diagram Kelas Sederhana*

Dari skenario-skenario tiap *use case* yang telah disampaikan pada subbab 2.5, diidentifikasi ada beberapa obyek yang dapat menjadi dasar pembuatan diagram kelas. Obyek-obyek dalam setiap *use case* tersebut yaitu:

### PUC-01 Booking Ruangan

Dari *use case* pertama yakni *booking* ruangan akan didapatkan sebuah obyek **Pemesanan**. **Pemesanan** tersebut adalah sebuah obyek berisi dokumen pemesanan dan verifikasi terhadap peminjaman sebuah ruangan. Obyek **Pemesanan** digunakan oleh **Pengguna** untuk mengirimkan permintaan peminjaman sebuah ruangan.

### PUC-02 Mengecek Data Ruangan dan Fasilitas

*Use case* kedua ini akan menampilkan obyek-obyek **Ruangan** yang ada.

### PUC-03 Memberikan Kritik dan Saran

Pada *use case* ketiga, akan terdapat obyek **Komentar**. obyek **Komentar** akan memiliki satu **Ruangan** yang dituju dan beberapa identitas **Pengguna**, sedangkan **Ruangan** akan memiliki beberapa **Komentar**,

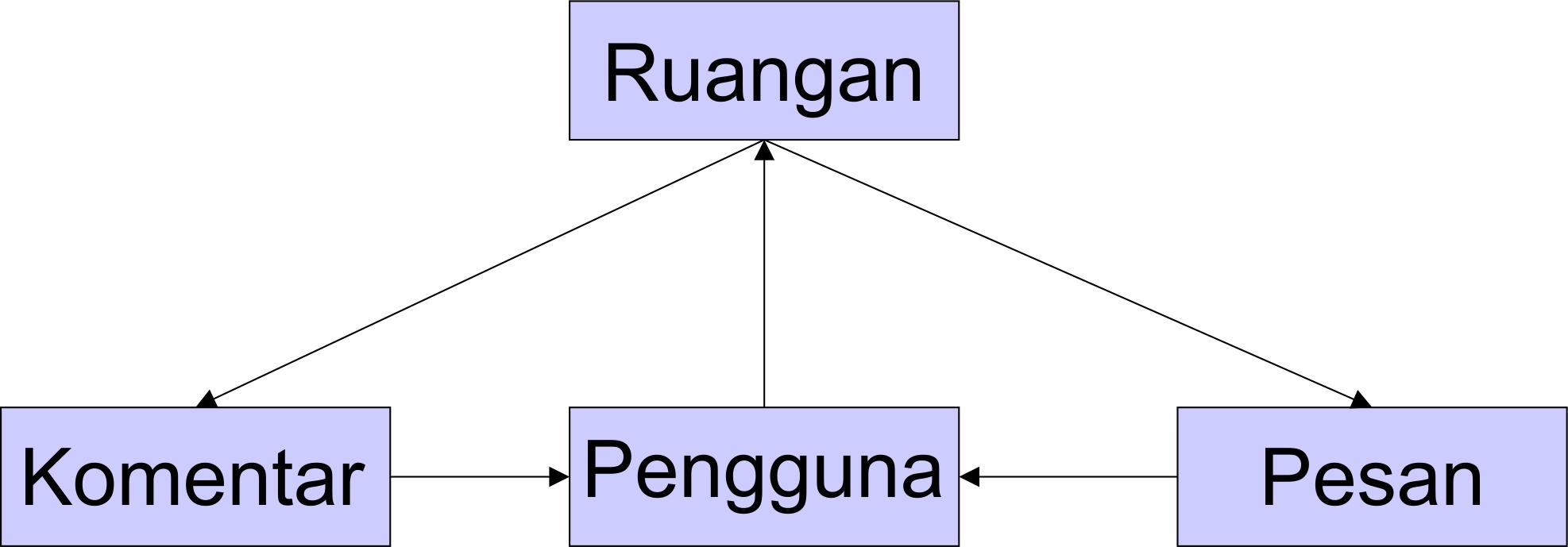
### PUC-04 Mengelola Data Ruangan

Pada *use case* kelima, *obyek* yang ada adalah obyek **Pemesanan** dan obyek **Ruangan.** Obyek **Pemesanan** akan ditindaklanjuti oleh **Pengguna** dalam level admin, sehingga **Ruangan** dapat di-*edit*statusnya. **Pengguna** dalam level admin dapat menindaklanjuti **Komentar** yang masuk. Obyek **Ruangan** juga dapat ditambah, di*-edit*, maupun dihapus dalam *use case* ini.

### PUC-05 Mengelola pengguna

Dalam *use case* terakhir yaitu mengelola pengguna, hanya terdapat obyek **Pengguna** yang terkait. Obyek **Pengguna** ini akan ditambahkan, diubah dan juga dihapus dalam *use case* ini.

Dari deskripsi obyek-obyek yang telah dijelaskan tersebut, dapat disusun sebuah diagram kelas yang terdiri dari 4 kelas seperti diperlihatkan Gambar 10.



**Gambar 10. PCA, Sketsa awal *class diagram* tahap analisis**