# DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**I-TAK**

## Dipersiapkan oleh:

Anbi Alif Rahman (1301174646)

Wilrades Christofel Purba (1301170753)

Rafif Ihdar Milzam (1301174633)

Program Studi Informatika Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

1. **Pendahuluan**

### Tujuan Penulisan Dokumen

Tujuan dengan dibuatnya dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini adalah untuk memberikan penjelasan mengenai perangkat lunak yang akan kami baik dengan gambaran umum ataupun penjelasan detil dan menyeluruh mengenai spesifikasi dari sistem informasi Itak.Kemudian yang akan menggunakan dokumen ini ialah pengembang dari sistem informasi perangkat lunak dan dan orang-orang lainya yang membutuhkan sebagai referensi dalam proses pengembangan sistem informasi Itak ini. Dengan adannya dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini diharapkan proes pengembangan sistem berjalan lancar dan lebih terarah selain itu dapat menghasilkan efisiensi dalam waktu,tenaga, dan biaya dalam proses pengembangan sistem informasi .

### Lingkup Masalah

Nama dari perangkat lunak/software yang akan kami buat adalah I-TAK, software

tersebut merupakan software yang berbasis web yang akan mempertemukan Event Organizer

yang akan mengadakan event dengan mahasiswa universitas telkom.

Tujuan utama dari software ini adalah memberikan informasi event kepada mahasiswa Universitas Telkom.

### Definisi dan Istilah

1. SKPL

Spesifikasi Kebutuhan Perangkat Lunak

Merupakan dokumen hasil analisis yang berisi spesifikasi

kebutuhan user

2.IEEE

Institute of Electrical Electronics Engineers

Merupakan standar international untuk pengembangan dan

rancangan perangkat lunak.

4.ERD

Entity Relational Diagram;Diagram yang menggambarkan relasi antara entitas dan atribut

dari masing-masing entitas.

5.Admin

Pemimpin di bidang pelaksanaan peraturan, prosedur, dan kebijakan.

6.User

Seseorang yang menggunakan produk

7.Database

Kumpulan dari beberapa data yang saling berhubungan satu

dengan yang lainnya

### Referensi

Sumber referensi dalam pembuatan SKPL ini adalah sebagai berikut:

1. IEEE std 830-1993, IEEE Recommended Practice for Software Requirement Spesification.

2. Universitas Telkom,”KR.297/KMHS7/BKA/2014 Tentang Transkrip Aktivitas

Kemahasiswaan (TAK) Universitas Telkom” 2014 [online]. Available:

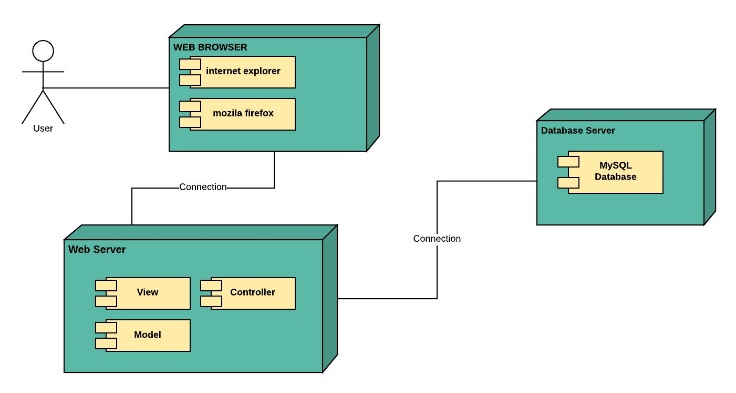
https://drive.google.com/file/d/1qABybenfuMLlbZX37U9amydtQKpH2o\_-/view

[Accessed 16 Maret 2020].

1. **Deskripsi Perancangan Global**

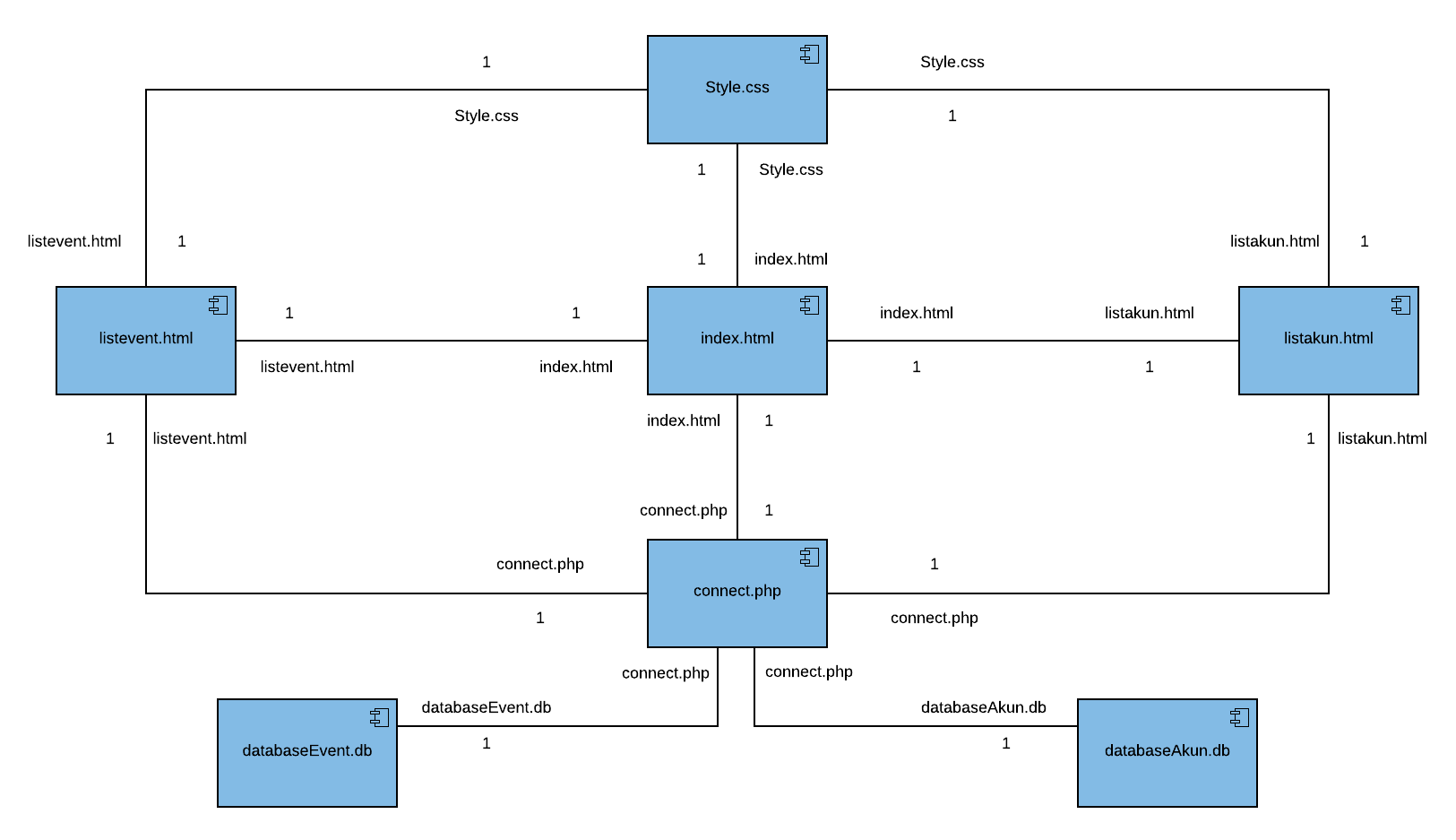
### Deskripsi Arsitektural

Perangkat lunak ini dibangun menggunakan web service yang menggunakan html, css, dan php dan penyimpanan menggunakan database yaitu database event dan akun, sebagai penghubung antara detil info event yang tersedia dengan menu utama yang merupakan fungsional sistem. Didalam menu utama terdapat beberapa komponen yang bergantung padannya yaitu daftar akun, daftar panitia event, daftar event, Harga event, komentar event.

****

### Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
|  | Web service | Penghubung entitas peserta dan event organizer dengan menu utama i-TAK |
|  | List Akun | Kumpulan kelas List akun, sebagai akses utama untuk masuk ke dalam sistem |
|  | List Event | Kumpulan kelas daftar event. Dependen terhadap menu utama |
|  | Database Event | Kumpulan kelas database event, yang berisi data event yang tersedia |
|  | Database Akun | Kumpulan kelas akun, yang berisi, data user akun yang telah mendaftarkan diri didalam sistem |



1. **Perancangan Rinci**
   1. ***Realisasi Use Case***

##### Berisi TABEL USE CASE sebagai berikut :

|  |  |  |
| --- | --- | --- |
| No | Nama Use Case | Deskripsi Use Case |
| 1 | Registrasi | Fungsi ini digunakan oleh user untuk daftar sebagai anggota ke system |
| 2 | Kelola Data Akun | Fungsi ini digunakan oleh user untuk mengedit data akun dari sistem |
| 3 | View Info Event | Fungsi ini digunakan oleh peserta untuk melihat semua info yang berkaitan dengan event dari sistem |
| 4 | Input Data Event | Fungsi ini digunakan oleh event organizer untuk menginputkan semua info tentang event ke system |
| 5 | Delete Data Event | Fungsi ini digunakan oleh admin dan EO untuk menghapus data event yang telah habis masa waktunya dari event |
| 6 | View Data User | Fungsi ini digunakan oleh admin untuk melihat semua data member dari system |
| 7 | Login | Fungsi ini dapat digunakan oleh user untuk melakukan login ke system |
| 8 | Input komentar | Fungsi ini dapat digunakan oleh user untuk menginputkan komentar di event section |
| 9 | Delete komentar | Fungsi ini dapat digunakan oleh user untuk menghapus komentar di event section |

* + 1. **Use Case Login**

Skenario Use Case Login :

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Login | |
| Input | Username, password | |
| Output | Autentifikasi Login | |
| Aktor | User/Admin | |
| Pre-Kondisi | User/admin ingin masuk kedalam program halaman utama tapi belum login | |
| Post-Kondisi | user/admin telah memasukkan username dan password yang disubmit dan diproses ke database | |
| Deskripsi | User/admin memasukkan data username dan password agar bisa masuk kedalam program utama yang setelah diinput maka masuk kedalam autentifikasi  untuk diproses ke data base | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Actor membuka program |  |
|  | 1. Program terbuka dan menampilkan halaman login. |
| 1. Actor dapat memilih ingin login sebagai admin,user,atau EO 2. Actor memasukkan data username dan password ke text-field username dan password 3. actor mengklik login |  |
|  | 1. program memproses dengan memautentifikasi inputan user 2. Jika benar maka akan masuk autentifikasi berhasil dengan sebuah keterangan 3. Jika tidak maka akan muncul autentifikasi salah setelah login tersebut |

Primary Flow :

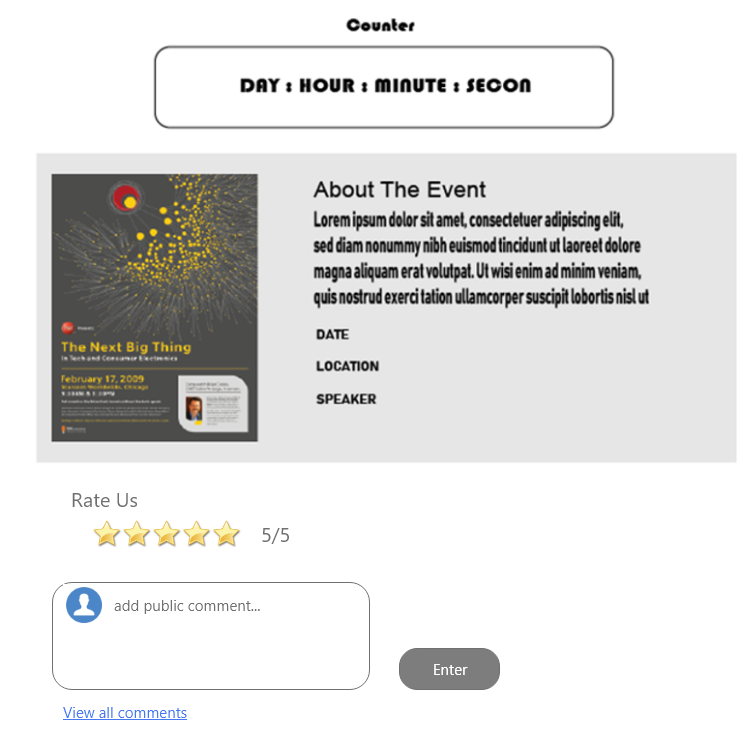
1. *Actor memilih ingin login sebagai admin,user,atau EO*
2. Setelah user menginputkan data pada textbox
3. User masuk ke halaman utama
4. User bisa mengubah data diri
5. Semua data tervalidasi setelah user menekan tombol ‘login’

Alternate Flow : Buat akun apabila login tidak berhasil

* + - 1. **Perancangan Antarmuka Usecase Login**

#### Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.

#### 



* + - * 1. **Tabel Identifikasi Antarmuka / Layar / Page**

*Semua gambar/layar prototipe antarmuka per Use Case dimasukkan ke dalam tabel di bawah ini.*

|  |  |  |
| --- | --- | --- |
| **ID.**  **LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| Ly-01 | Login | Page dimana user yang sudah teregistrasi bisa log in ke dalam sistem. |

##### UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil

Antarmuka XXX: {diisi dengan no. layar atau no gambar rancangan antarmuka} Page Sign In

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *Signin\_Button1* | *Button* | *LOGIN* | *Jika di-klik dan semua text sudah terisi dengan benar maka akan masuk kedalam halaman utama web* |
| *Check\_BoxButton* | *Checkbox* | *CheckBox* | *Jikan diCentang maka untuk first akun yang melakukan login dan belum melakukan sign out akan langsung direct kemenu utama tanpa ada login terebih dulu* |
| *Signin\_Button2* | *Button* | *SIGN UP* | *Berpindah ke tampilan SIGN UP* |
| *Signin\_Button3* | *Button* | *USER* | *Jika di klik maka halaman page akan berpindah untuk login user* |
| *Signin\_Button3* | *Button* | *Admin* | *Jika di klik maka halaman page akan berpindah untuk login admin* |
| *Signin\_Button4* | *Button* | *EO* | *Jika di klik maka halaman page akan berpindah untuk login Event Organizer* |

### Identifikasi Object Baru

Identifikasi object yang terkait dengan use case tersebut.

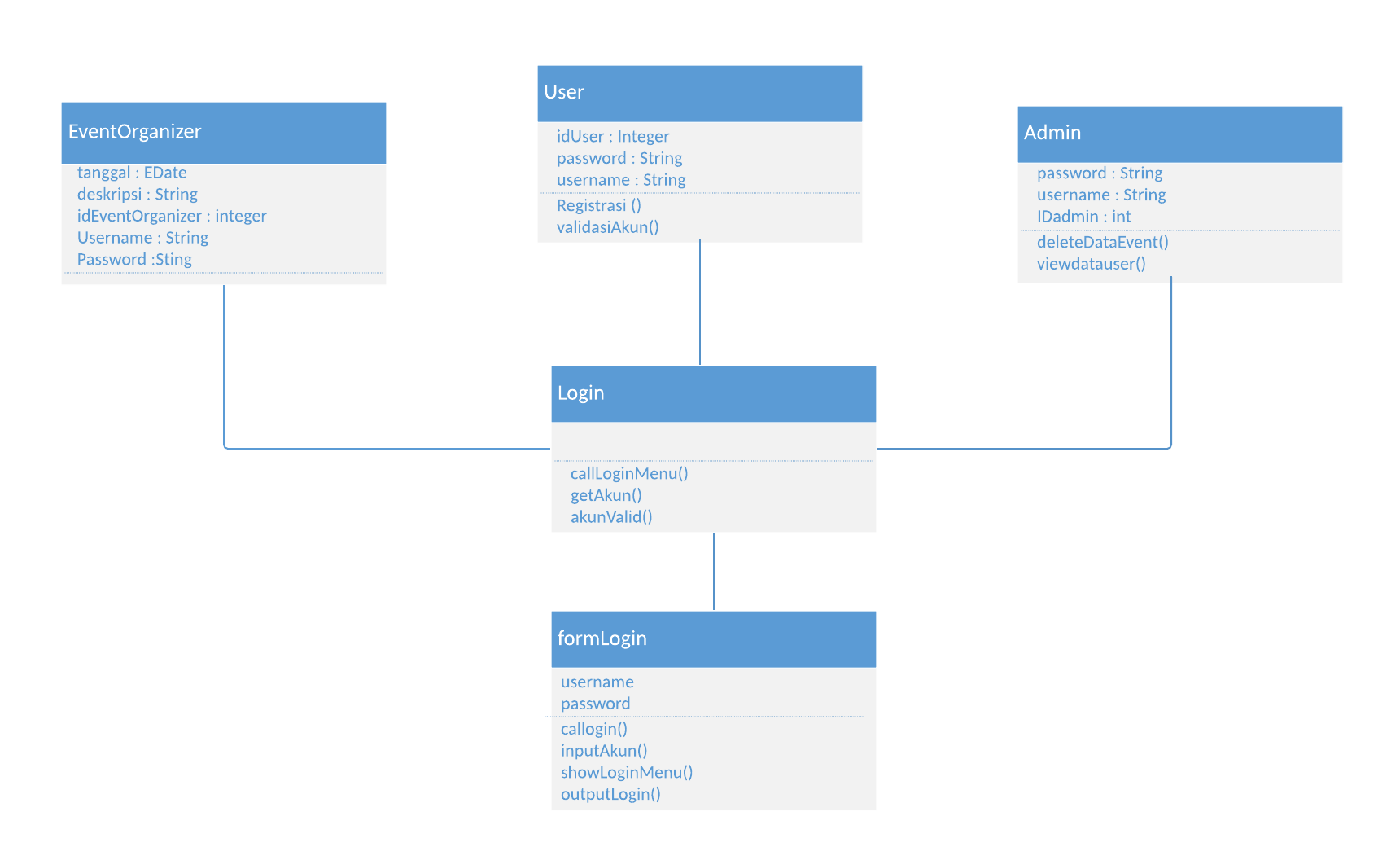
Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis. Gunakan tabel di bawah:

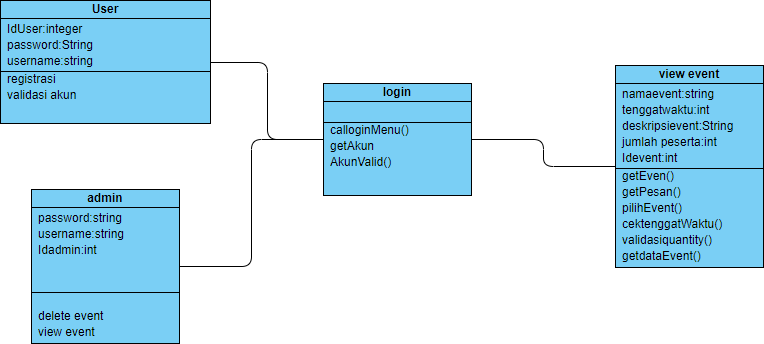
TABEL OBJECT PERANCANGAN

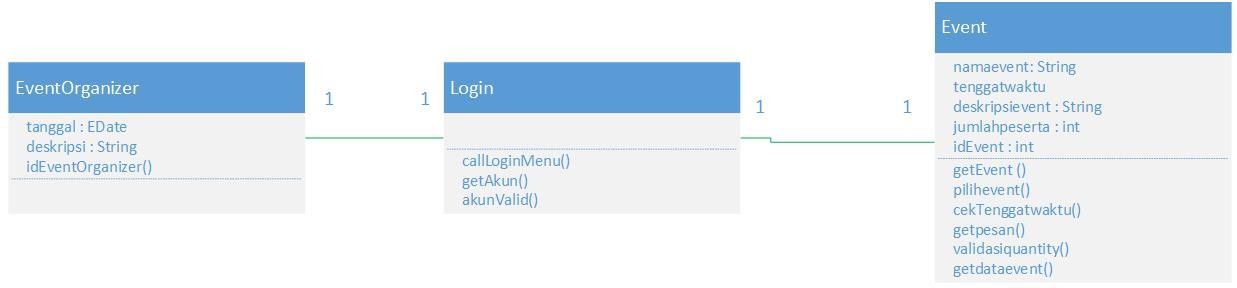
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
|  | User | User |
|  | Info Event | Info Event |
|  | Event | Event |

### Robustness DiagramLogin :

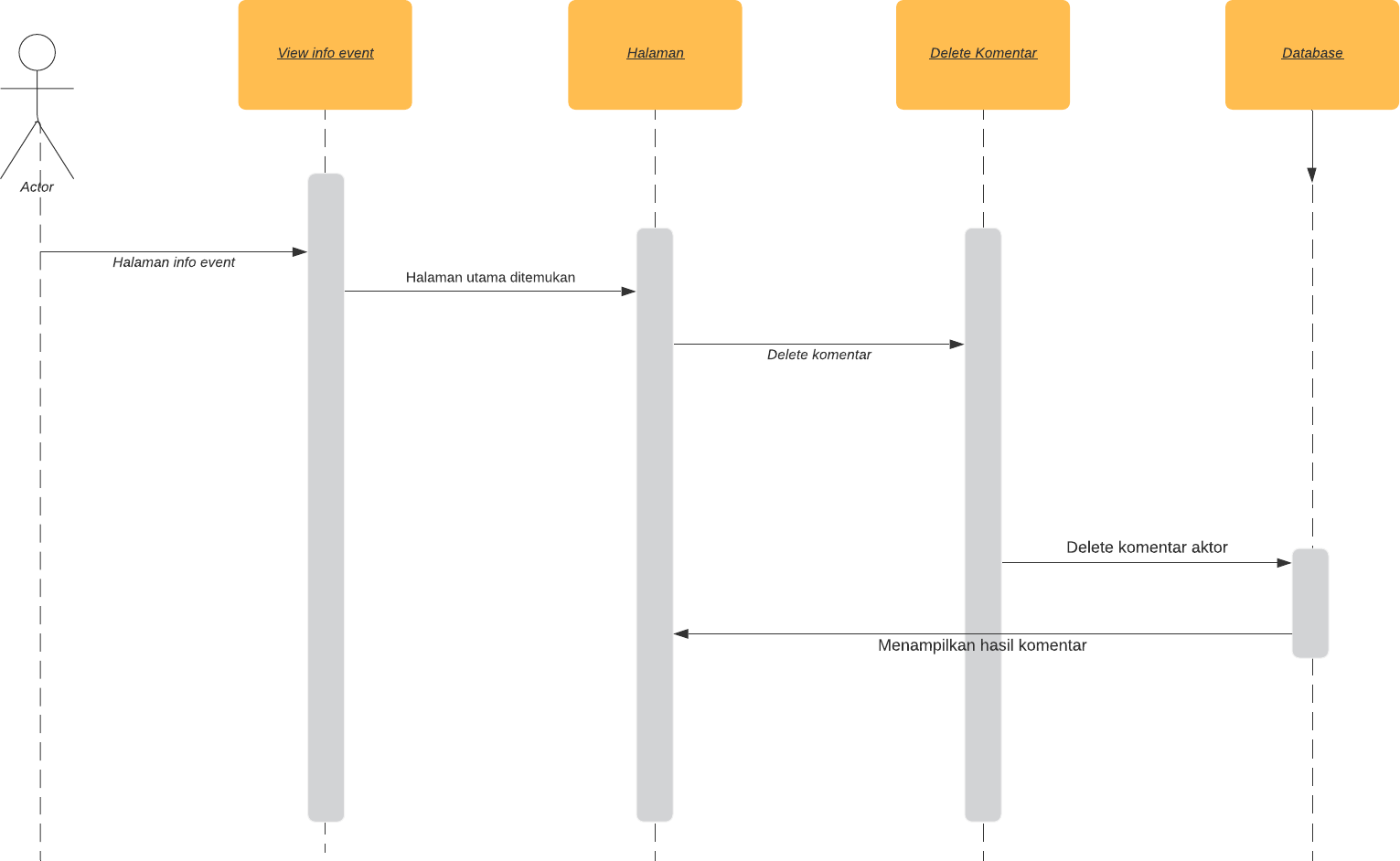
#### 

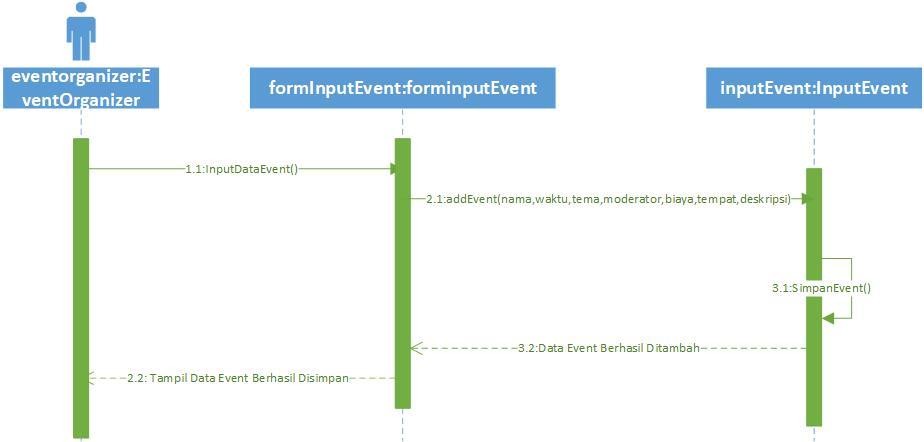
* + - 1. **Diagram Kelas**





### Sequence Diagram





1. **Perancangan Detil**

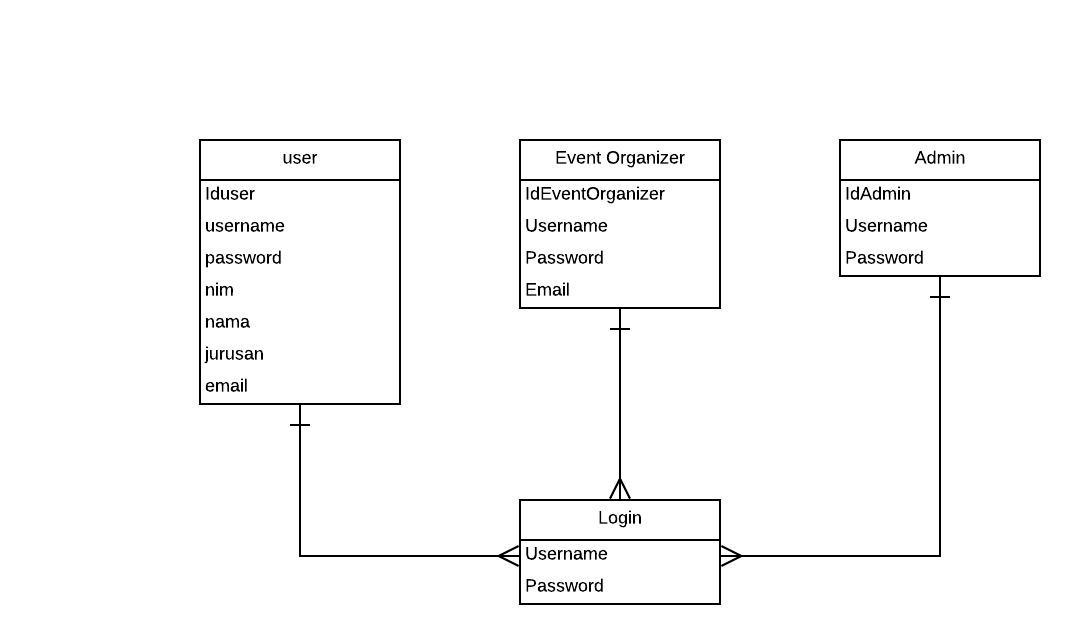
### Perancangan Detil Kelas

### TABEL KELAS :

|  |  |  |  |
| --- | --- | --- | --- |
| **ID**  **Kelas** | **Nama Kelas Perancangan** | **Atribute (visibility)** | **Method / Operation** |
| 001 | User | Private | Construct pembenetukan objek user |
| 002 | Login | Public | Untuk login user |

### Perancangan Kelas Persistensi

Skema Relasi database yang digunakan



### Perancangan Algoritma

### 4.3.1 Algoritma login

Contoh:

Nama Kelas : user Nama Operasi : login() Algoritma :

*Input username(username)*

*Input password(password)*

*if (username != db[ usename]) then*

*Output(“username salah!”)*

*Else*

*If(password != db[password]) then*

*Output(“Password salah”)*

*Else*

*Output(“Login Berhasil”)*

### Perancangan Query

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | Insert INTO user (Nama, Nim, Email, Jurusan, Username, Password) values (‘nama’, ‘nim’, ’email’, ’jurusan’ ,‘username’, ‘password’); | *Memasukkan data user kedalam database user* |
| *Q-002* | SELECT Judul, harga,tanggal,komentar,deskripsi,image\_url FROM DataEvent | Melihat data Event yang terdapat dalam database DataEvent |
| Q-003 | Insert INTO EventOrganizer (Nama\_Organisasi, Email, Username, Password) values (‘Nama\_Organisasi ‘,’email’,‘username’, ‘password’); | *Memasukkan data EventOrganizer kedalam database EnventOrganizer* |

**Matriks Kerunutan (Requirement Traceability Matrix)**

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Functional Requirement** | **Nama Use Case** |
| FR-02 | User dan Event Organizer | Login |