**DPPL-09**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

(Tik-Et)

Dipersiapkan oleh:

Akbar Taufiqurrahman (1301174602)

Adriansyah Dwi Rendragraha (1301174385)

Azha Alvin Rahmansyah (1301174149)

Fahlar Reva Fauzi (1301170446)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-09* | | *<#>/<jml #* |
| Revisi | *-* | *Tgl: -* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

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

Daftar Halaman Perubahan

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

Daftar Isi

[1. Pendahuluan 8](#_Toc23814467)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc23814468)

[1.2 Lingkup Masalah 8](#_Toc23814469)

[1.3 Definisi dan Istilah 8](#_Toc23814470)

[1.4 Referensi 9](#_Toc23814471)

[1.5 Sistematika Pembahasan 9](#_Toc23814472)

[2 Deskripsi Perancangan Global 10](#_Toc23814473)

[2.1 Deskripsi Arsitektural 10](#_Toc23814474)

[2.2 Deskripsi Komponen 10](#_Toc23814475)

[3 Perancangan Rinci 11](#_Toc23814476)

[3.1 Realisasi Use Case 11](#_Toc23814477)

[3.1.1 Use Case Registrasi 11](#_Toc23814478)

[3.1.1.1 Identifikasi Kelas 11](#_Toc23814479)

[3.1.1.2 Sequence Diagram 11](#_Toc23814480)

[3.1.1.3 Diagram Kelas 12](#_Toc23814481)

[3.1.2 Use Case Login 13](#_Toc23814482)

[3.1.2.1 Identifikasi Kelas 13](#_Toc23814483)

[3.1.2.2 Sequence Diagram 13](#_Toc23814484)

[3.1.2.3 Diagram Kelas 14](#_Toc23814485)

[3.1.3 Use Case Lihat Acara 14](#_Toc23814486)

[3.1.3.1 Identifikasi Kelas 14](#_Toc23814487)

[3.1.3.2 Sequence Diagram 15](#_Toc23814488)

[3.1.3.3 Diagram Kelas 16](#_Toc23814489)

[3.1.4 Use Case Lihat Pesanan atau Tiket 17](#_Toc23814490)

[3.1.4.1 Identifikasi Kelas 17](#_Toc23814491)

[3.1.4.2 Sequence Diagram 17](#_Toc23814492)

[3.1.4.3 Diagram Kelas 18](#_Toc23814493)

[3.1.5 Use Case Pemesanan 18](#_Toc23814494)

[3.1.5.1 Identifikasi Kelas 18](#_Toc23814495)

[3.1.5.2 Sequence Diagram 19](#_Toc23814496)

[3.1.5.3 Diagram Kelas 20](#_Toc23814497)

[3.1.6 Use Case Verifikasi Pembayaran 20](#_Toc23814498)

[3.1.6.1 Identifikasi Kelas 20](#_Toc23814499)

[3.1.6.2 Sequence Diagram 21](#_Toc23814500)

[3.1.6.3 Diagram Kelas 22](#_Toc23814501)

[3.1.7 Use Case Pengelolaan Acara 22](#_Toc23814502)

[3.1.7.1 Identifikasi Kelas 22](#_Toc23814503)

[3.1.7.2 Sequence Diagram 23](#_Toc23814504)

[3.1.7.3 Diagram Kelas 24](#_Toc23814505)

[3.1.8 Use Case Input Acara 25](#_Toc23814506)

[3.1.8.1 Identifikasi Kelas 25](#_Toc23814507)

[3.1.8.2 Sequence Diagram 25](#_Toc23814508)

[3.1.8.3 Diagram Kelas 26](#_Toc23814509)

[3.1.9 Use Case Edit Acara 27](#_Toc23814510)

[3.1.9.1 Identifikasi Kelas 27](#_Toc23814511)

[3.1.9.2 Sequence Diagram 27](#_Toc23814512)

[3.1.9.3 Diagram Kelas 28](#_Toc23814513)

[3.1.10 Use Case Hapus Acara 29](#_Toc23814514)

[3.1.10.1 Identifikasi Kelas 29](#_Toc23814515)

[3.1.10.2 Sequence Diagram 29](#_Toc23814516)

[3.1.10.3 Diagram Kelas 30](#_Toc23814517)

[3.1.11 Use Case Pengelolaan User 31](#_Toc23814518)

[3.1.11.1 Identifikasi Kelas 31](#_Toc23814519)

[3.1.11.2 Sequence Diagram 31](#_Toc23814520)

[3.1.11.3 Diagram Kelas 32](#_Toc23814521)

[3.1.12 Use Case Lihat User 33](#_Toc23814522)

[3.1.12.1 Identifikasi Kelas 33](#_Toc23814523)

[3.1.12.2 Sequence Diagram 33](#_Toc23814524)

[3.1.12.3 Diagram Kelas 34](#_Toc23814525)

[3.1.13 Use Case Hapus User 35](#_Toc23814526)

[3.1.13.1 Identifikasi Kelas 35](#_Toc23814527)

[3.1.13.2 Sequence Diagram 35](#_Toc23814528)

[3.1.13.3 Diagram Kelas 36](#_Toc23814529)

[3.1.14 Use Case Lihat Semua Pesanan 37](#_Toc23814530)

[3.1.14.1 Identifikasi Kelas 37](#_Toc23814531)

[3.1.14.2 Sequence Diagram 37](#_Toc23814532)

[3.1.14.3 Diagram Kelas 38](#_Toc23814533)

[3.1.15 Use Case Lihat Semua Acara 38](#_Toc23814534)

[3.1.15.1 Identifikasi Kelas 38](#_Toc23814535)

[3.1.15.2 Sequence Diagram 38](#_Toc23814536)

[3.1.15.3 Diagram Kelas 39](#_Toc23814537)

[3.1.16 Use Case Lihat Acara Admin 40](#_Toc23814538)

[3.1.16.1 Identifikasi Kelas 40](#_Toc23814539)

[3.1.16.2 Sequence Diagram 40](#_Toc23814540)

[3.1.16.3 Diagram Kelas 41](#_Toc23814541)

[3.1.17 Use Case Lihat Pesanan Customer 42](#_Toc23814542)

[3.1.17.1 Identifikasi Kelas 42](#_Toc23814543)

[3.1.17.2 Sequence Diagram 42](#_Toc23814544)

[3.1.17.3 Diagram Kelas 43](#_Toc23814545)

[3.2 Perancangan Detil Kelas 43](#_Toc23814546)

[3.2.1 Kelas User 44](#_Toc23814547)

[3.2.2 Kelas Customer 44](#_Toc23814548)

[3.2.3 Kelas Admin 44](#_Toc23814549)

[3.2.4 Kelas SuperAdmin 44](#_Toc23814550)

[3.2.5 Kelas Tiket 45](#_Toc23814551)

[3.2.6 Kelas Acara 45](#_Toc23814552)

[3.3 Diagram Kelas Keseluruhan 47](#_Toc23814553)

[3.4 Algoritma/QueryContoh: 47](#_Toc23814554)

[3.5 Diagram Statechart 52](#_Toc23814555)

[3.5.1 Registrasi Customer 52](#_Toc23814556)

[3.5.2 Registrasi Admin 53](#_Toc23814557)

[3.5.3 Login 53](#_Toc23814558)

[3.5.4 Input Acara 53](#_Toc23814559)

[3.5.5 Input Tiket 53](#_Toc23814560)

[3.5.6 Edit Customer 53](#_Toc23814561)

[3.5.7 Edit Admin 53](#_Toc23814562)

[3.5.8 Edit Acara 54](#_Toc23814563)

[3.5.9 Edit Tiket 54](#_Toc23814564)

[3.5.10 Delete User 54](#_Toc23814565)

[3.5.11 Delete Acara 54](#_Toc23814566)

[3.5.12 Delete Tiket 54](#_Toc23814567)

[3.6 Perancangan Antarmuka 55](#_Toc23814568)

[3.6.1 Menu Login 55](#_Toc23814569)

[3.6.2 Menu Registrasi 55](#_Toc23814570)

[3.6.3 Menu Lihat Acara 56](#_Toc23814571)

[3.6.4 Menu Keranjang 57](#_Toc23814572)

[3.6.5 Menu Pemesanan 58](#_Toc23814573)

[3.6.6 Menu Tiket 58](#_Toc23814574)

[3.6.7 Menu Edit Profil 59](#_Toc23814575)

[3.7 Perancangan Representasi Persistensi Kelas 60](#_Toc23814576)

[3.7.1 ER Diagram 60](#_Toc23814577)

[3.7.2 Skema Relasi 61](#_Toc23814578)

[4 Matriks Kerunutan 61](#_Toc23814579)

[5 Lampiran 61](#_Toc23814580)

[5.1 Rancangan Lingkungan Implementasi 61](#_Toc23814581)

[5.2 Use Case Skenario 62](#_Toc23814582)

[*Usecase Scenario #1* 62](#_Toc23814583)

[*Usecase Scenario #2* 62](#_Toc23814584)

[*Usecase Scenario #3* 63](#_Toc23814585)

[*Usecase Scenario #4* 64](#_Toc23814586)

[*Usecase Scenario #5* 64](#_Toc23814587)

[*Usecase Scenario #6* 65](#_Toc23814588)

[*Usecase Scenario #7* 66](#_Toc23814589)

[*Usecase Scenario #8* 67](#_Toc23814590)

[*Usecase Scenario #9* 68](#_Toc23814591)

[*Usecase Scenario #10* 68](#_Toc23814592)

[*Usecase Scenario #11* 69](#_Toc23814593)

[5.3 Functional Requirements 70](#_Toc23814594)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen DPPL ini memuat Deskripsi Perancangan Perangkat Lunak (DPPL) untuk Aplikasi Sistem Informasi *E-Ticketing*. Dokumen DPPL ini kami susun dengan tujuan-tujuan sebagai berikut.

1. Untuk dijadikan acuan dalam tahap pembangunan perangkat lunak oleh pengembang.
2. Untuk memberikan penjelasan dan gambaran mengenai perancangan perangkat lunak yang akak disusun baik berupa gambaran umum maupun penjelasan detil dan menyeluruh.
3. Menghasilkan analisis sistem yang mencakup sekumpulan proses yang terorganisasi secara hirarkis, dimana proses-proses saling berkomunikasi melalui suatu jalur aliran data.
4. Dengan disusunnya dokumen ini, diharapkan pembangunan Aplikasi Sistem Informasi *E-Ticketing* dapat lebih terarah, lebih efektif, dan lebih efisien.

## Lingkup Masalah

Aplikasi Sistem Informasi *E-Ticketing* ini merupakan aplikasi penjualan tiket dalam bentuk tiket elektroink yang nantinya, tiket tidak lagi berbentuk kertas fisik sebagaimana biasanya, akan tetapi berbentuk surat elektronik yang didalamnya terdapat sebuah QR code, yang mana QR code tersebut dapat di-*scan*. Tiket yang dijual pun merupakan tiket-tiket acara suatu event. Aplikasi ini juga dapat digunakan sebagai sarana validasi tiket masuk acara, yaitu dengan men-*scan* QR code yang ada pada tiket melalui *smartphone* menggunakan *webcam* pada laptop, sehingga dapat mempercepat proses validasi dan pencocokan tiket. Namun dalam hal pembayaran tiket, verifikasi pembayaran masih dilakukan langsung oleh admin yang menjual tiket yang lebih detailnya akan dijelaskan pada bab kedua.

## Definisi dan Istilah

|  |  |  |
| --- | --- | --- |
| No | ISTILAH,  SINGKATAN, AKRONIM | KETERANGAN |
| 1. | DPPL | Deskripsi Perancangan Perangkat Lunak, merupakan deskripsi dari perangkat lunak yang aka dikembangkan. |
| 2. | SKPL | Spesifikasi Kebutuhan Perangkat Lunak, merupakan dokumentasi kebutuhan perangkat lunak. |
| 3. | Algoritma | Langkah langkah yang ditulis untuk menyelesaikan masalah pemrograman komputer. |
| 4. | Query | Bahasa untuk melakukan perintah terhadap database, yang lebih dikenal dengan nama SQL. |
| 5. | Use Case Diagram | Diagram yang menggambarkan actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. |
| 6. | Kelas Diagram | Diagram yang menggambarkan kelas kelas dalam sebuah sistem dan hubungannya antara satu dengan yang kain, serta dimasukkan atribut dan operasinya. |
| 7. | Sequence Diagram | Diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek untuk menunjukkan rangkaian pesan yang dikirim objek dan interaksi antar objek. |
| 8. | Component Diagram | Diagram yang menggambarkan struktur fisik dari sebuah sistem dan digunakan untuk mengiilustrasikan bagaimana kode program dibagi menjadi beberapa komponen dan mendeskripsikan hubungan antar komponen. |
| 9. | Statechart Diagram | Diagram yang menggambarkan *behaver pattern* dari sebuah objek, dari awal objek tersebut di inisialisasi sampa di destroy. |
| 10. | ER Diagram | Model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara peyimpanan. |
| 11. | Skema Relasi | Nama relasi yang diikuti dengan kumpulan atribut atributnya. |

Tabel 1.2.1 Definisi dan Istilah

## Referensi

Referensi yang kami gunakan dalam menyusun dokumen DPPL ini adalah sebagai berikut.

1. Spesifikasi Kebutuhan Perangkat Lunak (SKPL) Sistem Informasi *E-Ticketing*.
2. Modul Praktikum APPL, IF LAB, Fakultas Informatika, Telkom University.

## Sistematika Pembahasan

Lingkup penulisan dokumen ini didasarkan pada Bab-bab yang ada dengan penjelasan singkat sebagaimana dibawah ini.

1. Pada Bab1 Pendahuluan, berisi tentang pengertian dokumen, tujuan penulisan okumen dan ruang lingkup serta referensi.
2. Pada Bab 2 Deskripsi Perancangan Global, berisi perspektif dan fungsi produk, karakteristik pengguna, batasan-batasan, dan asumsi yang digunakan pada perangkat lunak yang akan dibangun.
3. Pada Bab 3 Deskripsi Rincian Kebutuhan, berisi kebutuhan antarmuka eksternal, kebutuhan fungsional, deskripsi data, deskripsi kebutuhan non-fungsional, serta bahasan perancangan.
4. Pada Bab 4 Matriks Kerunutan, verisi mapping use case dengan kelas kelas terkait.
5. Pada Lampiran akan dilampirkan dokumen tambahan yang dinilai perlu untuk ditambahkan pada SKPL ini.

# Deskripsi Perancangan Global

## Deskripsi Arsitektural

A close up of a map

Description automatically generated

Gambar 2.1.1 Diagram Komponen

## Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1. | DB Manager | Database yang menyimpan semua data yang dapat mengakses perangkat lunak |
| 2. | Tambah Customer | Menambah customer baru |
| 3. | Tambah Admin | Menambah admin baru |
| 4. | Tambah Acara | Menambah acara baru |
| 5. | Tambah Tiket | Menambah tiket |
| 6. | Edit Customer | Mengedit data customer |
| 7. | Edit Admin | Mengedit data admin |
| 8. | Edit Acara | Mengedit data acara |
| 9. | Edit Tiket | Mengedit data tiket |
| 10. | Hapus Acara | Mengahpus acara |
| 11. | Hapus Tiket | Menghapus Tiket |
| 12. | Lihat Acara | Melihat acara yang ada |

Tabel 2.2.1 Deskripsi Komponen

# Perancangan Rinci

## Realisasi Use Case

A close up of a map

Description automatically generated

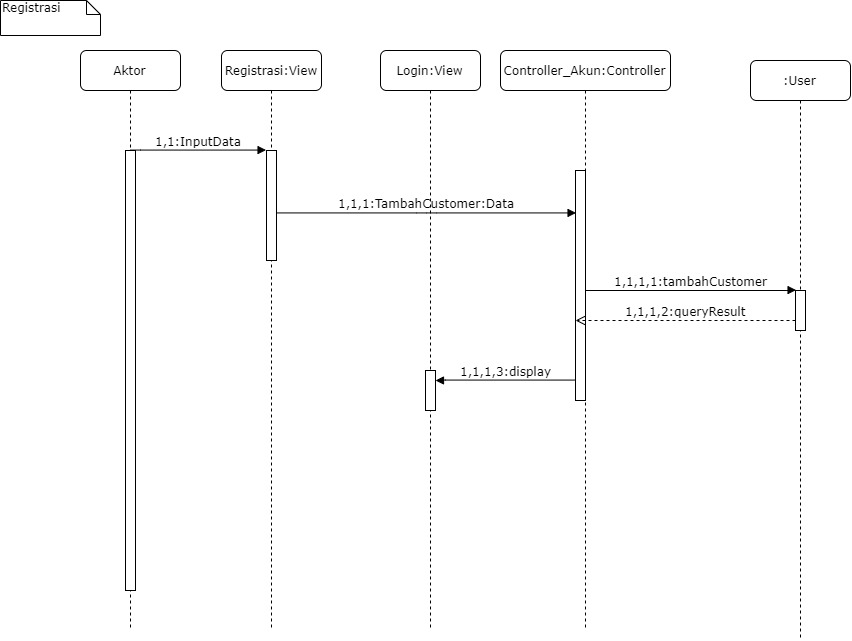
### Use Case Registrasi

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Customer* | *Actor* |
| *2* | *Admin* | *Actor* |
| *3* | *ControllerAkun* | *Controller* |
| *4* | *User* | *Entity* |
| *5* | *V\_registrasi* | *Boundary* |
| *6* | *V\_login* | *Boundary* |

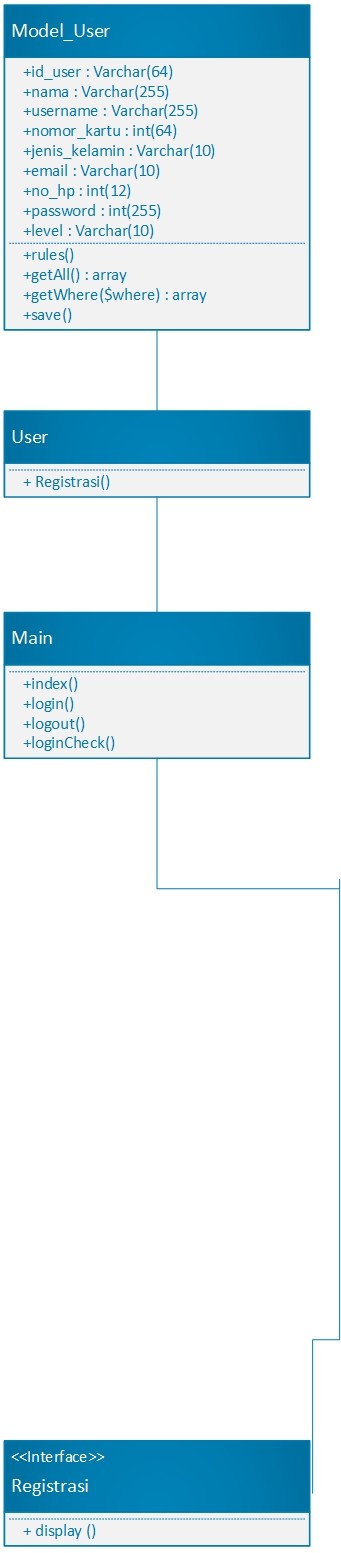
Tabel 3.1.1.1.1 Identifikasi Kelas Use Case Registrasi

#### Sequence Diagram



Gambar 3.1.1.2.1 Seuquence Diagram Use Case Registrasi

#### Diagram Kelas

**

Gambar 3.1.1.3.1 Class Diagram Use Case Registrasi

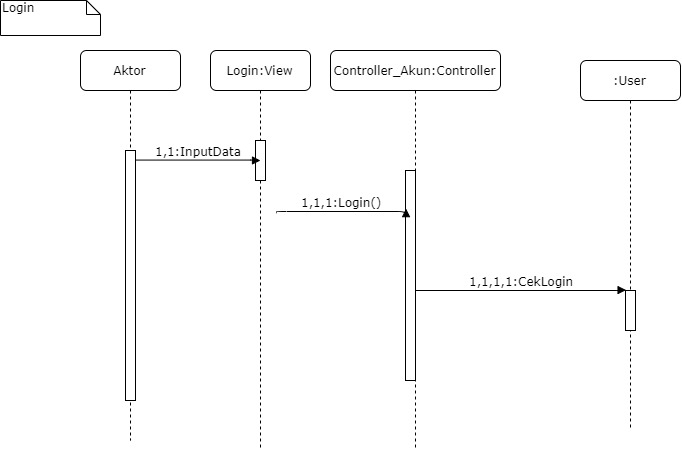
### Use Case Login

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1.* | *Customer* | *Aktor* |
| *2.* | *Admin* | *Aktor* |
| *3.* | *Super\_Admin* | *Aktor* |
| *4.* | *ControllerAkun* | *Controller* |
| *5.* | *v\_login* | *Boundary* |
| *6.* | *User* | *Entity* |

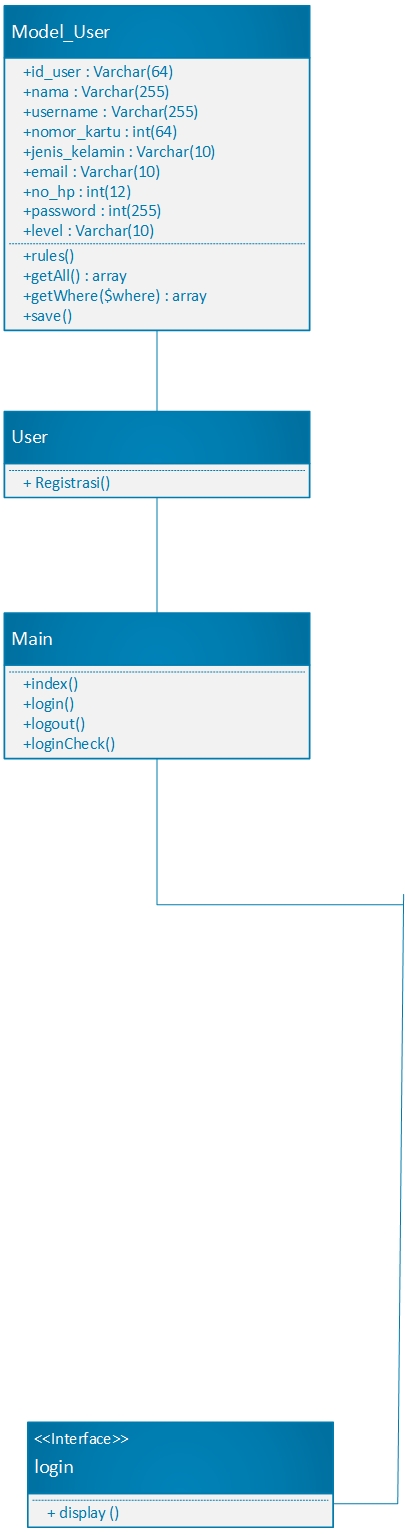
Tabel 3.1.2.1.1 Identifikasi Kelas Use Case Login

#### Sequence Diagram



Gambar 3.1.2.2.1 Sequence Diagram Use Case Login

#### Diagram Kelas



Gambar 3.1.2.3.1 Class Diagram Use Case Login

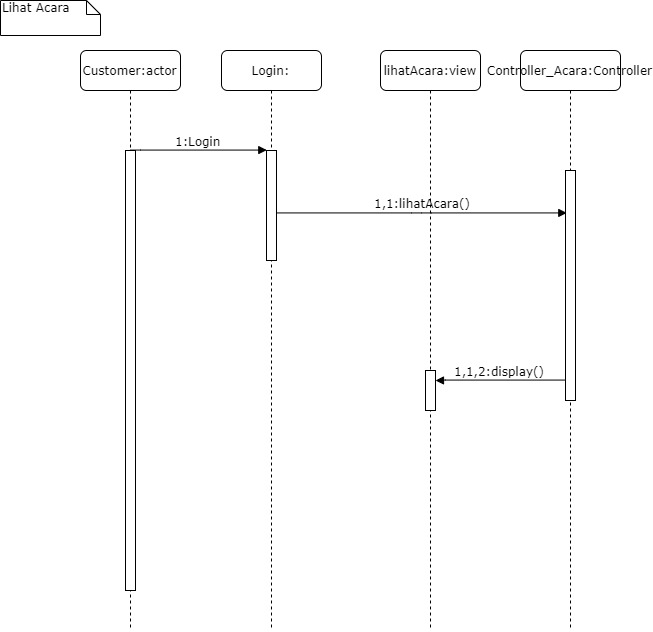
### Use Case Lihat Acara

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *SuperAdmin* | *Actor* |
| *2* | *Admin* | *Actor* |
| *3* | *Customer* | *Actor* |
| *4* | *ControllerAkun* | *Controller* |
| *5* | *ControllerAcara* | *Controller* |
| *6* | *Acara* | *Entity* |
| *7* | *V\_lihatAcara* | *Boundary* |

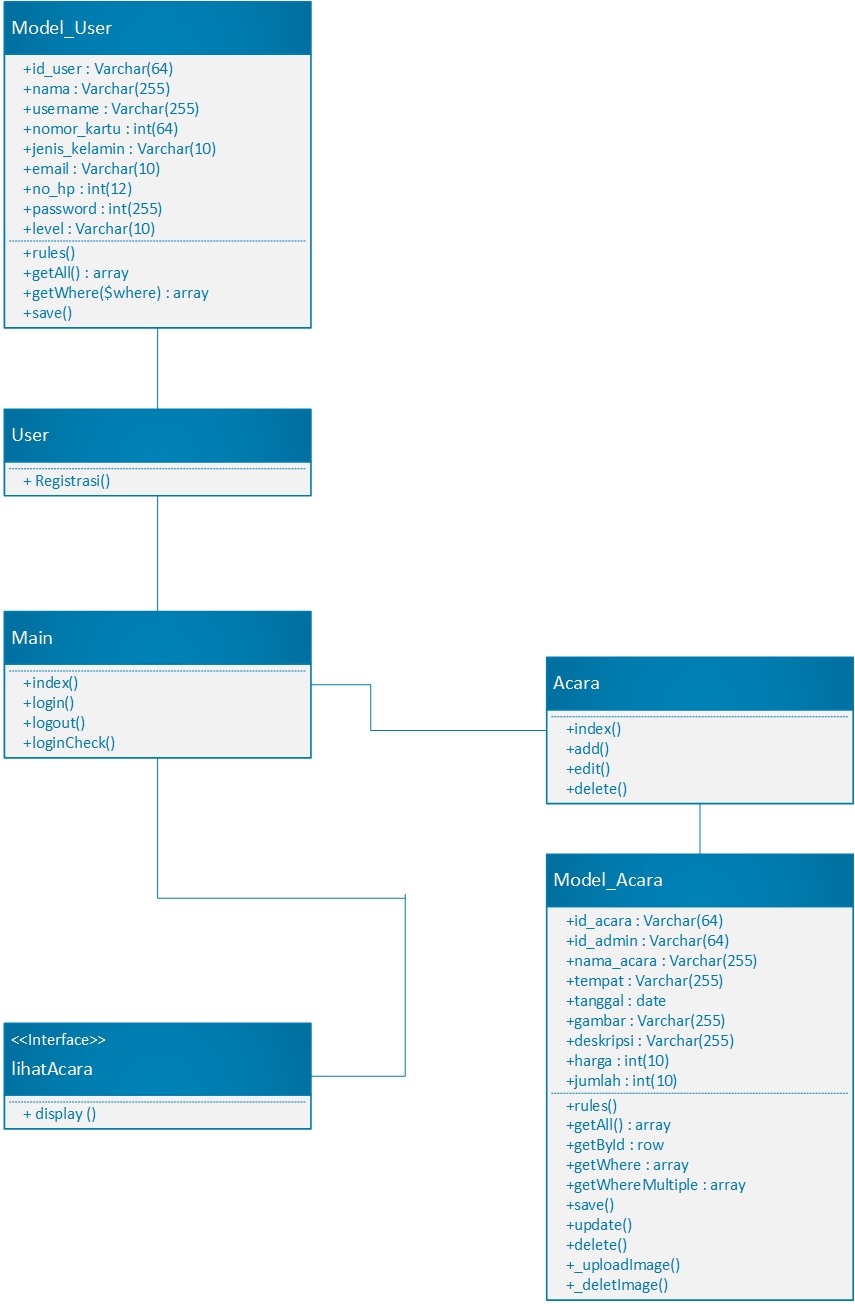
Tabel 3.1.3.1.1 Identifikasi Kelas Use Case Lihat Acara

#### Sequence Diagram



Gambar 3.1.3.2.1 Sequence Diagram Use Case Lihat Acara

#### Diagram Kelas



Gambar 3.1.3.3.1 Class Diagram Use Case Lihat Acara

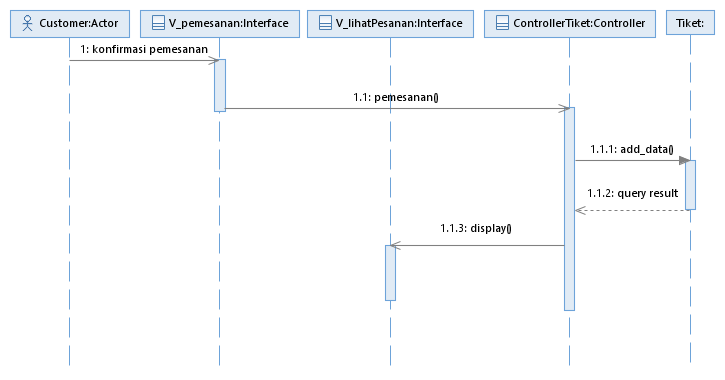
### Use Case Pesanan

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Customer* | *Actor* |
| *2* | *ControllerTiket* | *Controller* |
| *3* | *Tiket* | *Entity* |
| *4* | *Acara* | *Entity* |
| *5* | *V\_pemesanan* | *Boundary* |
| *6* | *V\_lihatPesanan* | *Boundary* |

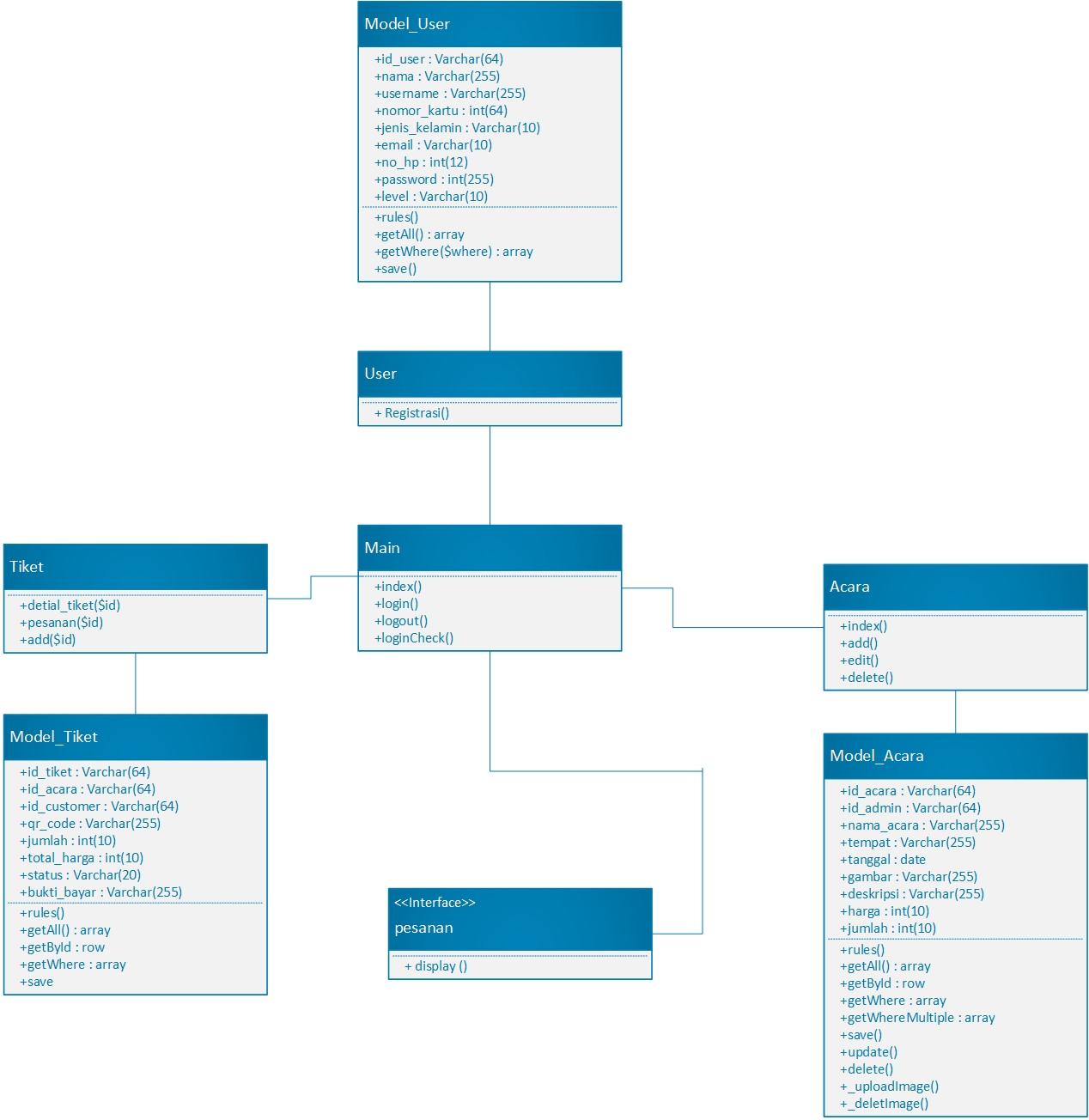
Gambar 3.1.5.1.1 Identifikasi Kelas Use Case Pemesanan

#### Sequence Diagram



Gambar 3.1.5.2.1 Sequence Diagram Use Case Pemesanan

#### Diagram Kelas



Gambar 3.1.5.3.1 Class Diagram Use Case Pemesanan

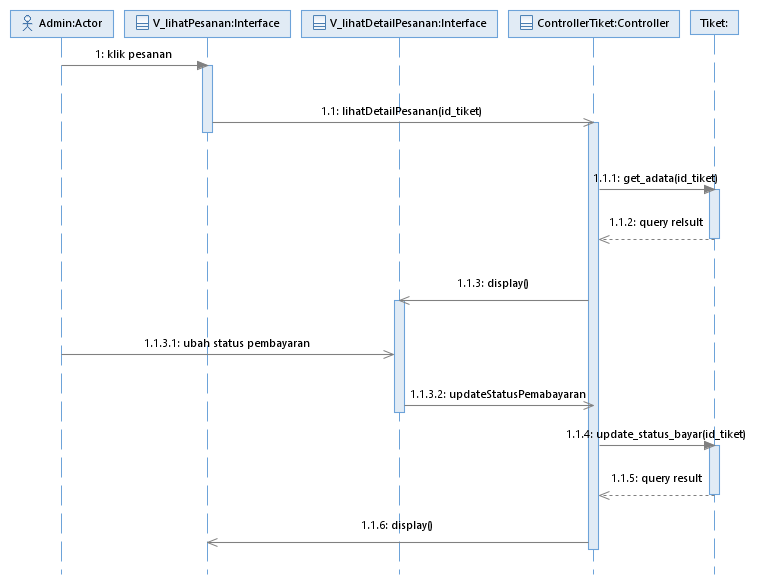
### Use Case Verifikasi Pembayaran

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Admin* | *Actor* |
| *2* | *ControllerTiket* | *Controller* |
| *3* | *Tiket* | *Entity* |
| *4* | *V\_lihatPesanan* | *Boundary* |
| *5* | *V\_lihatDetailPesanan* | *Boundary* |

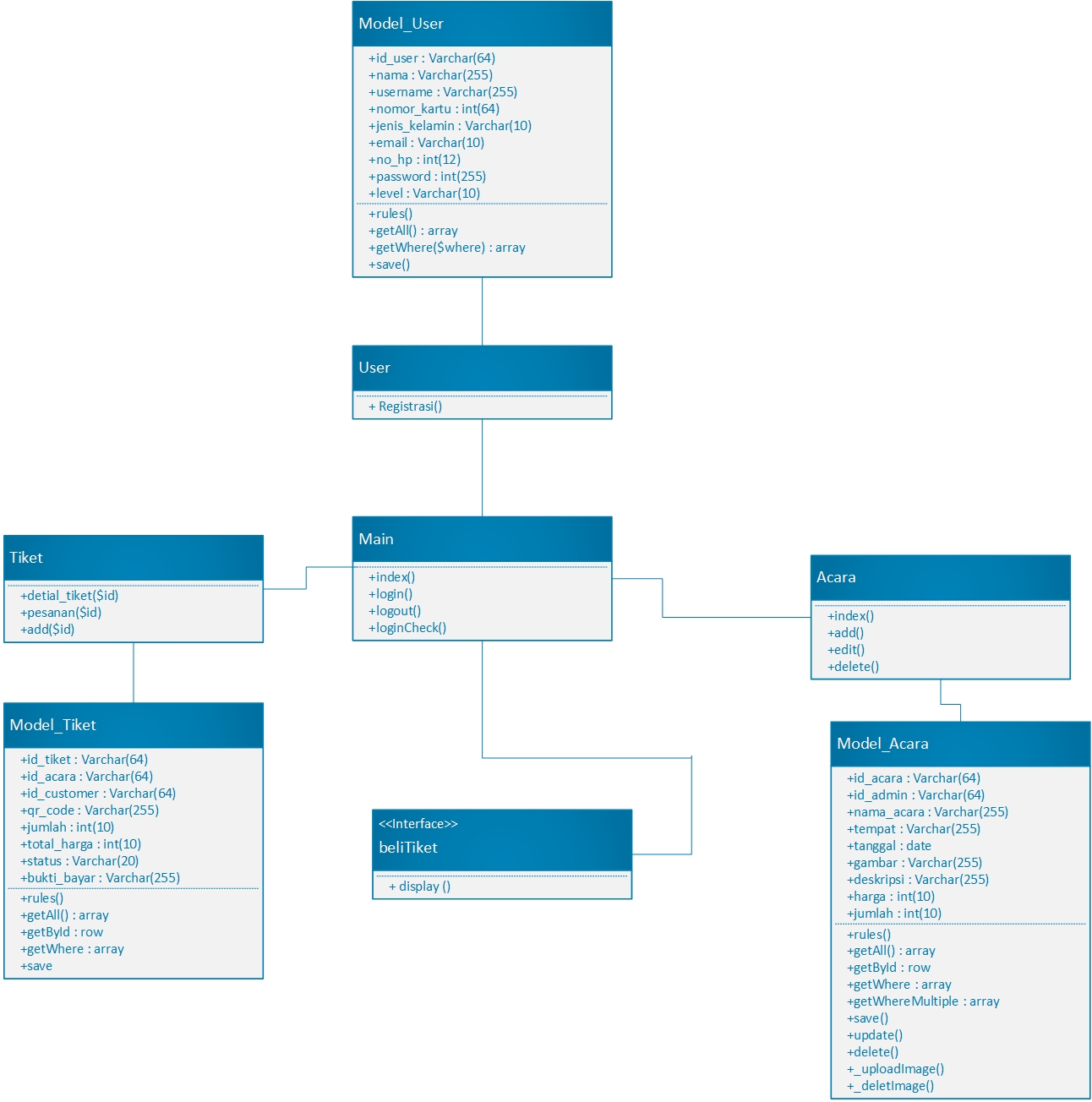
Tabel 3.1.6.1.1 Identifikasi Kelas Use Case Verifikasi Pembayaran

#### Sequence Diagram



Gambar 3.1.6.2.1 Sequence Diagram Use Case Verifikasi Pembayaran

#### Diagram Kelas



Gambar 3.1.6.3.1 Class Diagram Use Case Verifikasi Pembayaran

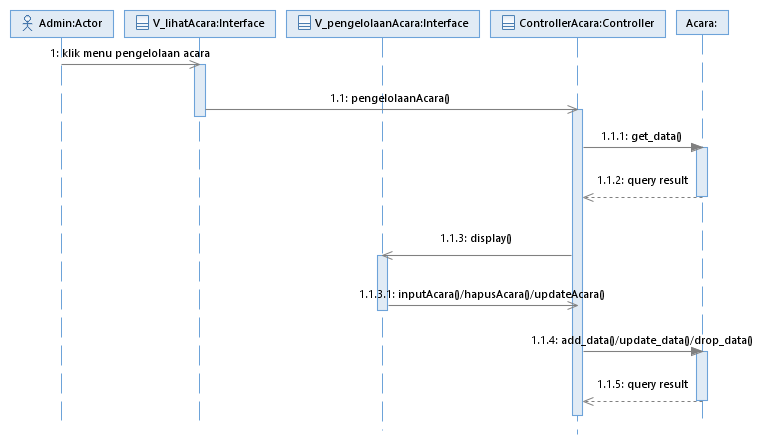
### Use Case Pengelolaan Acara

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Admin* | *Admin* |
| *2* | *ControllerAdmin* | *ControllerAdmin* |
| *3* | *Acara* | *Acara* |
| *4* | *V\_lihatAcara* | *V\_lihatAcara* |
| *5* | *V\_pengelolaanAcara* | *V\_pengelolaanAcara* |

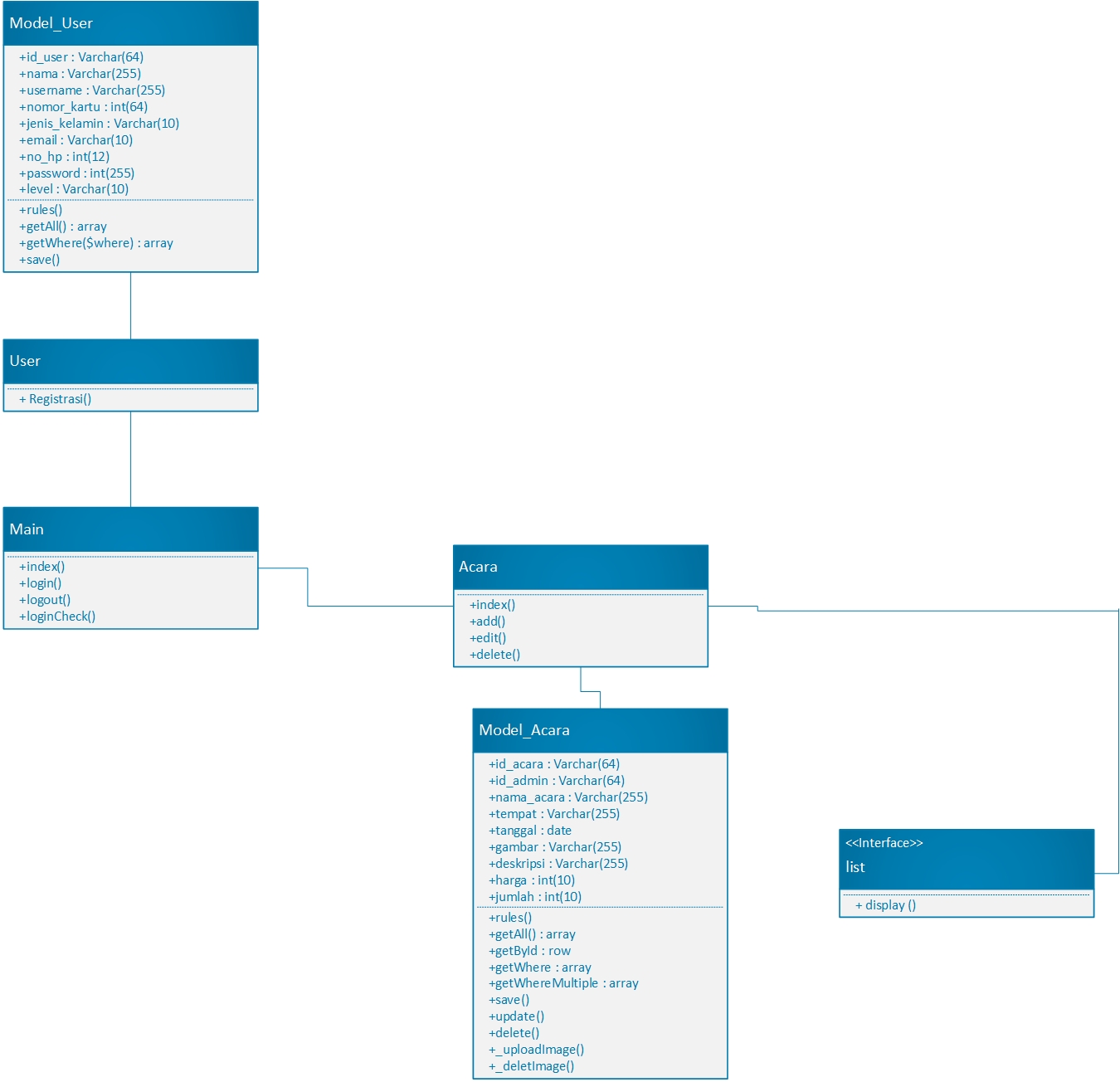
Tabel 3.1.7.1.1 Identifiasi Kelas Use Case Pengelolaan Acara

#### Sequence Diagram



Gambar 3.1.7.2.1 Sequence Diagram Use Case Pengelolaan Acara

#### Diagram Kelas



Gambar 3.1.7.3.1 Class Diagram Use Case Pengelolaan Acara

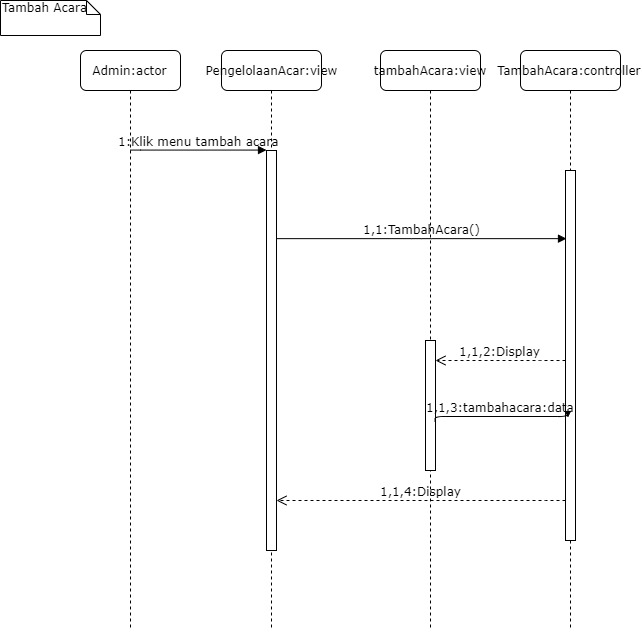
### Use Case Tambah Acara

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Admin* | *Actor* |
| *2* | *ControllerAcara* | *Controller* |
| *3* | *Acara* | *Entity* |
| *4* | *V\_inputAcara* | *Boundary* |
| *5* | *V\_pengelolaanAcara* | *Boundary* |

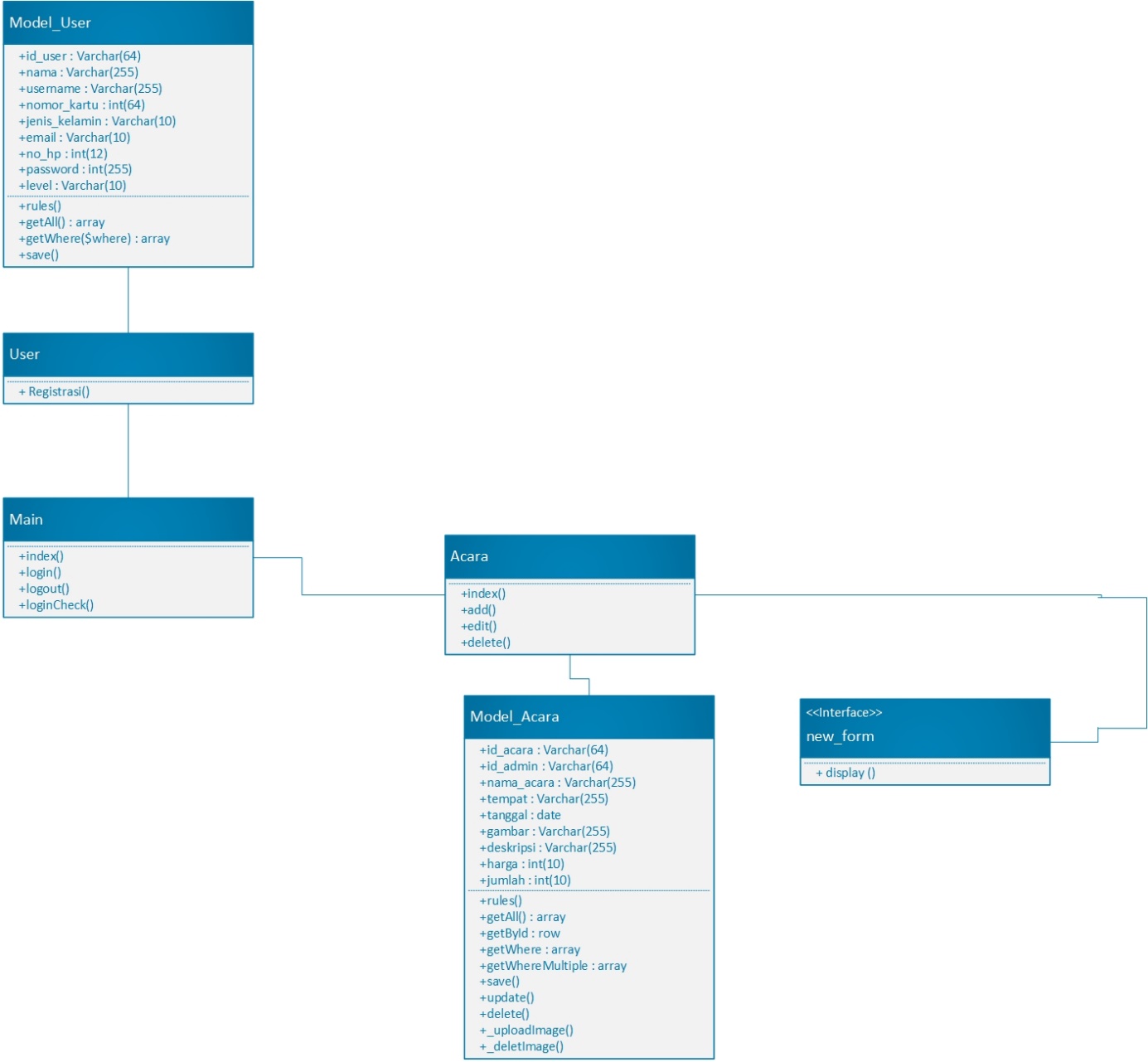
Tabel 3.1.8.1.1 Identifikasi Kelas Use Case Tambah Acara

#### Sequence Diagram



Gambar 3.1.8.2.1 Sequence Diagram Use Case Tambah Acara

#### Diagram Kelas



Gambar 3.1.8.3.1 Class Diagram Use Case Tambah Acara

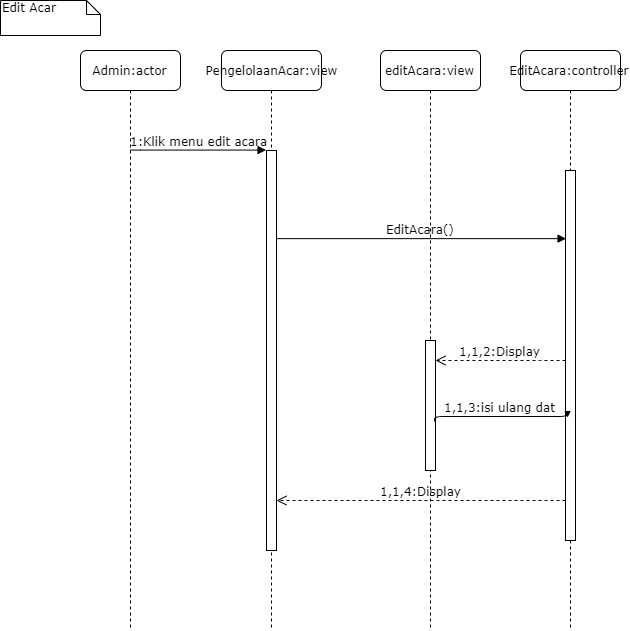
### Use Case Edit Acara

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Admin* | *Actor* |
| *2* | *ControllerAcara* | *Controller* |
| *3* | *Acara* | *Entity* |
| *4* | *V\_updateAcara* | *Boundary* |
| *5* | *V\_pengelolaanAcara* | *Boundary* |

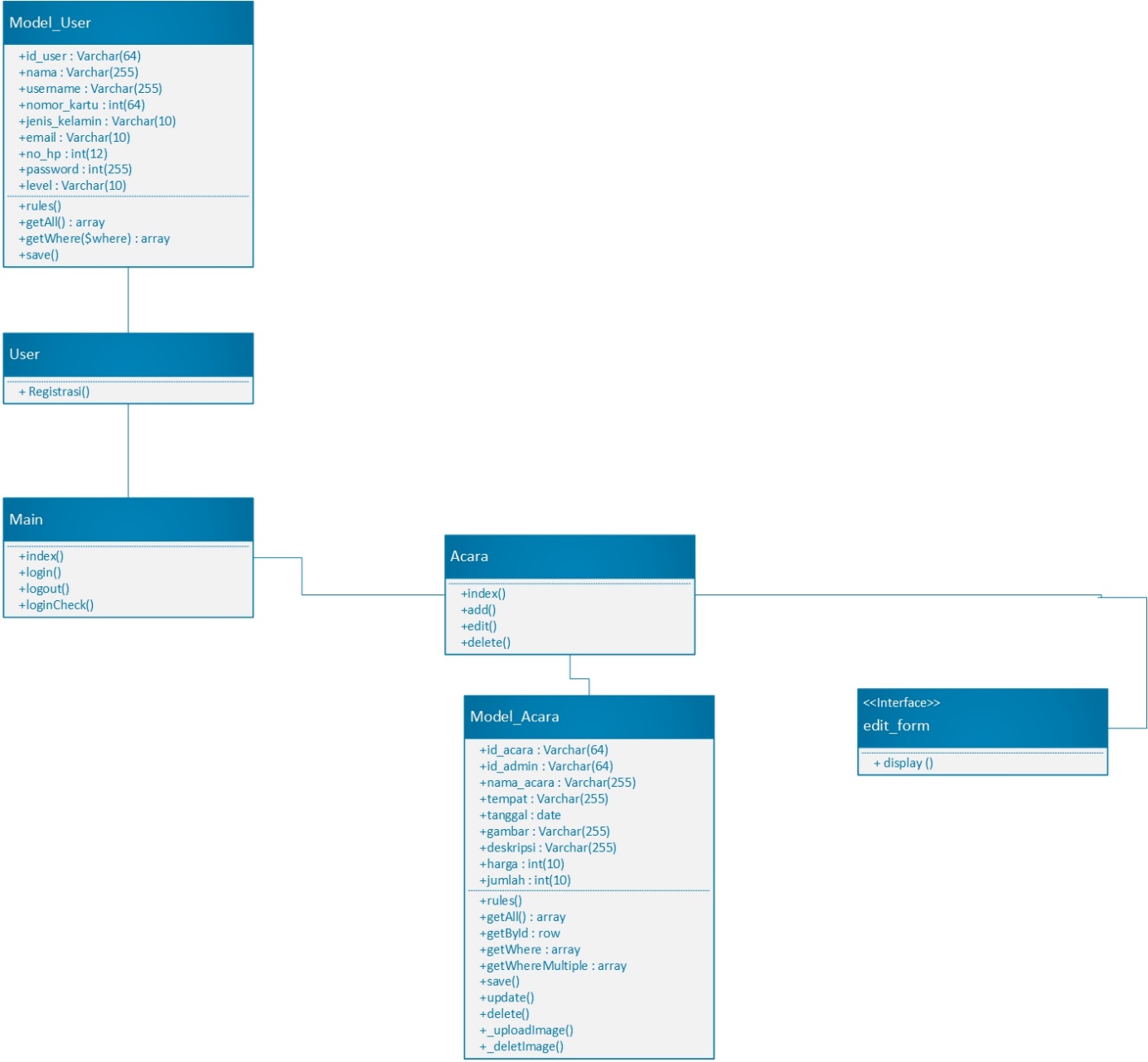
Tabel 3.1.9.1.1 Identifikasi Kelas Use Case Edit Acara

#### Sequence Diagram



Gambar 3.1.9.2.1 Sequence Diagram Use Case Edit Acara

#### Diagram Kelas



Gambar 3.1.9.3.1 Class Diagram Use Case Edit Acara

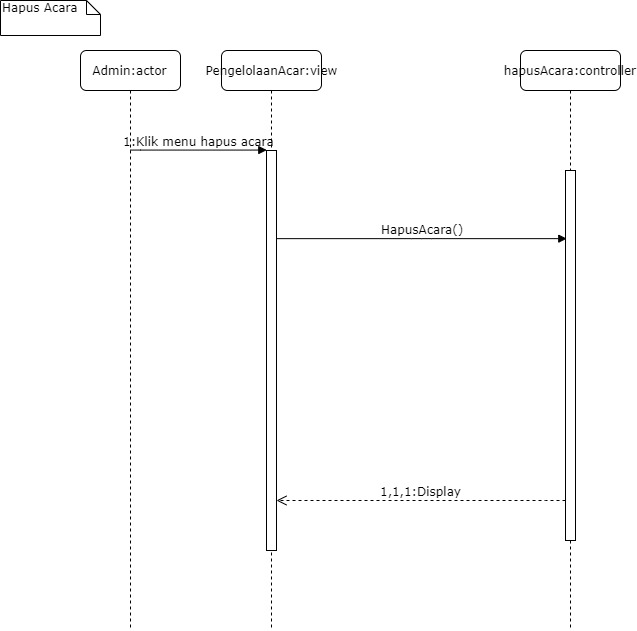
### Use Case Hapus Acara

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Admin* | *Actor* |
| *2* | *ControllerAcara* | *Controller* |
| *3* | *Acara* | *Entity* |
| *4* | *V\_pengelolaanAcara* | *Boundary* |

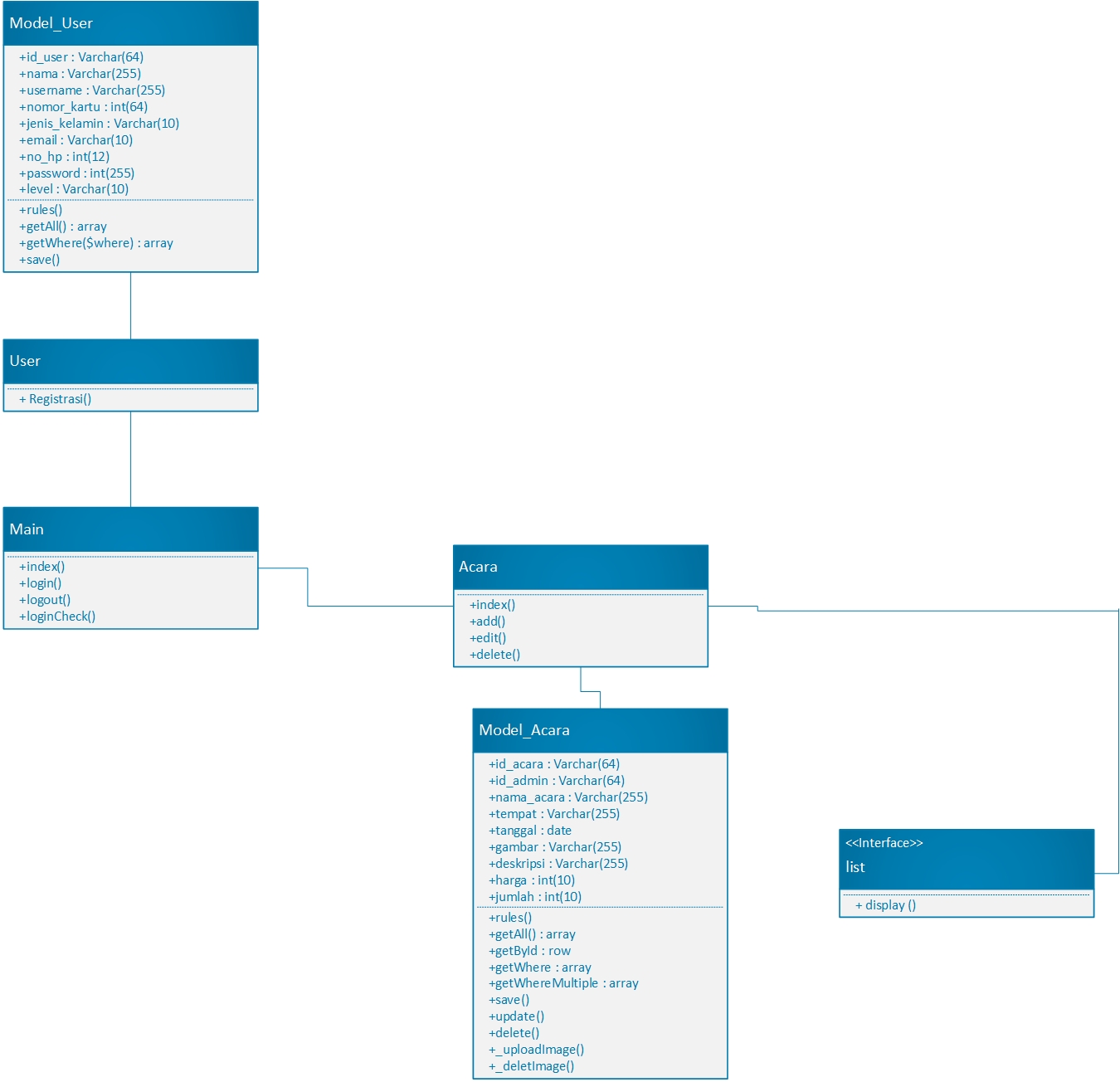
Tabel 3.1.10.1.1 Identifikasi Kelas Use Case Hapus Acara

#### Sequence Diagram



Gambar 3.1.10.2.1 Sequence Diagram Use Case Hapus Acara

#### Diagram Kelas



Gambar 3.1.10.3.1 Class Diagram Use Case Hapus Acara

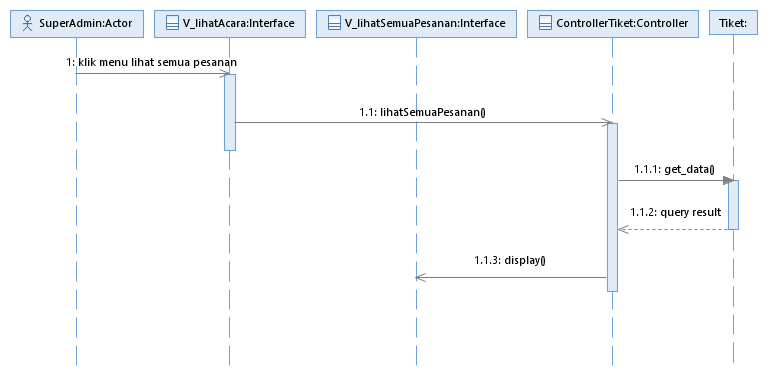
### Use Case Lihat Pesanan

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *SuperAdmin* | *Actor* |
| *2* | *ControllerTiket* | *Controller* |
| *3* | *Tiket* | *Entity* |
| *4* | *V\_lihatAcara* | *Boundary* |
| *5* | *V\_lihatSemuaPesanan* | *Boundary* |

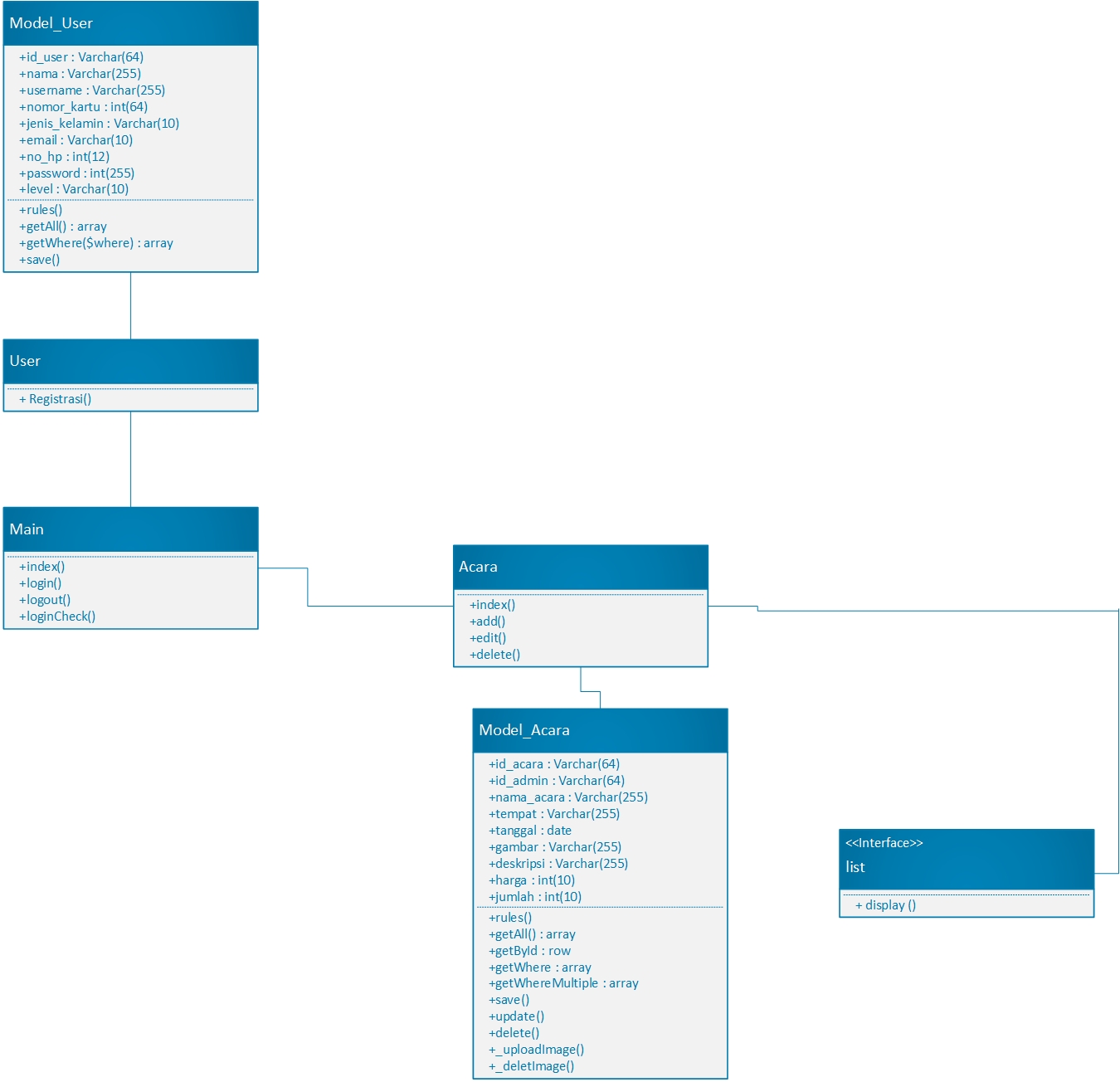
Tabel 3.1.14.1.1 Identifikasi Kelas Use Case Lihat Pesanan

#### Sequence Diagram



Gambar 3.1.14.2.1 Seuqence Diagram Use Case Lihat Pesanan

#### Diagram Kelas

**

Gambar 3.1.14.3.1 Class Diagram Use Case Lihat Pesanan

## Perancangan Detil Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1.* | *Model\_User* | *Model\_User* |
| *2.* | *User* | *User* |
| *3.* | *Main* | *Main* |
| *4.* | *Tiket* | *Tiket* |
| *6.* | *Acara* | *Acara* |
| *7.* | *Model\_Tiket* | *Model\_Tiket* |
| *8.* | *Model\_Acara* | *Model\_Acara* |
| *9.* | *beliTiket* | *beliTiket* |
| *10.* | *lihatAcara* | *lihatAcara* |
| *11.* | *pesanan* | *pesanan* |
| *12.* | *login* | *login* |
| *13.* | *Registrasi* | *Registrasi* |
| *14.* | *detail\_tiket* | *detail\_tiket* |
| *15.* | *edit\_form* | *edit\_form* |
| *16.* | *new\_form* | *new\_form* |
| *17.* | *list* | *list* |

Tabel 3.2.1 Perancangan Detail Kelas

### Kelas Model\_User

*Nama Kelas : Model\_User.*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *rules* | *Public* | *Untuk masuk kedalam sistem* |
| *getAll* | *Public* | *Untuk mendapatkan semua data* |
| *getWhere* | *Public* |  |
| *Save* | *Public* | *Untik mengsave data* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *id\_user* | *Public* | *Varchar* |
| *nama* | *Public* | *Varchar* |
| *username* | *Public* | *Varchar* |
| *nomor\_kartu* | *Public* | *Integer* |
| *jenis\_kelamin* | *Public* | *Varchar* |
| *email* | *Public* | *Varchar* |
| *no\_hp* | *Public* | *Integer* |
| *password* | *Public* | *Integer* |
| *level* | *Public* | *varchar* |

Tabel 3.2.1.1 Kelas Model\_User

### Kelas User

*Nama Kelas : User*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Registrasi* | *Public* | *Untuk user membuat akun pada sistem* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

Tabel 3.2.2.1 Kelas User

### Kelas Main

*Nama Kelas : Main*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *index* | *Public* | *Untuk mendapat index* |
| *login* | *Public* | *Untuk masuk kedalam aplikasi* |
| *logout* | *Public* | *Untuk keluar dari aplikasi* |
| *loginCheck* | *Public* | *Untuk mengecek data pada saat mau login* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

Tabel 3.2.3.1 Kelas Main

### Kelas Acara

*Nama Kelas : Acara*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Index* | *Public* | *Untuk mendapatkan index* |
| *add* | *Public* | *Untuk menambah acara* |
| *edit* | *Public* | *Untuk mengedit acara* |
| *delete* | *Public* | *Untuk menghapus acara* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

Tabel 3.2.4.1 Kelas Acara

### Kelas Tiket

*Nama Kelas : Tiket.*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Detail\_tiket* | *Public* | *Untuk mendapat detail tiket* |
| *pesanan* | *Public* | *Untuk mendapat pesanan* |
| *add* | *Public* | *Untuk menambah Tiket* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

Tabel 3.2.5.1 Kelas Tiket

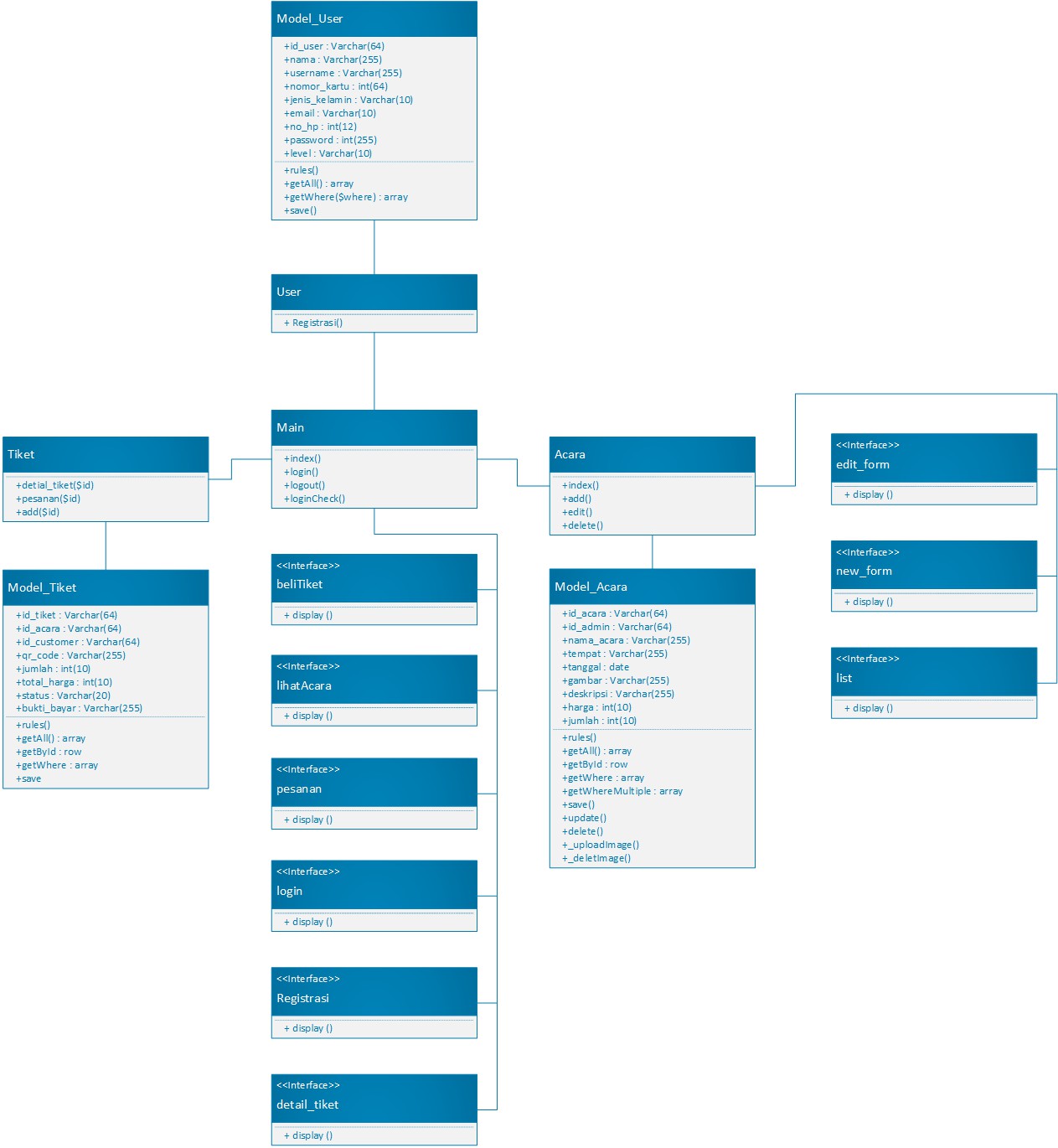
### Kelas Model\_Acara

*Nama Kelas : Model\_Acara*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *rules* | *Public* |  |
| *getAll* | *Public* | *Untuk mendapatkan semua data acara* |
| *getById* | *Public* | *Untuk mendapatkan data acara berdasarkan id* |
| *getWhere* | *Public* |  |
| *save* | *Public* |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *id\_tiket* | *Public* | *Varchar* |
| *id\_acara* | *Public* | *Varchar* |
| *id\_ customer* | *Public* | *Varchar* |
| *qr\_code* | *Public* | *Varchar* |
| *jumlah* | *Public* | *Integer* |
| *total\_harga* | *Public* | *Integer* |
| *Status* | *Public* | *Varchar* |
| *bukti\_bayar* | *Public* | *Varchar* |

Tabel 3.2.6.1 Kelas Acara

## Diagram Kelas Keseluruhan



Gambar 3.3.1 Kelas Diagram Keseluruhan

## Algoritma/QueryContoh:

*Nama Kelas :User*

*Nama Operasi :Login*

*Algoritma : (Algo-001)*

*Input(username, password)*

*Query(Q-001)*

*If(true)*

*Pindahcontroller*

*Else*

*Output(‘Username/password salah’)*

*Nama Kelas : Customer*

*Nama Operasi : Registrasi*

*Algoritma : (Algo-002)*

*Input(username, password, nama, kategori)*

*Query(Q-002)*

*If(username ditemukan dari Q-002)*

*Output(‘Username sudah terdftar’)*

*Else*

*Query(Q-003)*

*Nama Kelas : Customer*

*Nama Operasi : editCustomer*

*Algoritma : (Algo-003)*

*Input(username)*

*Query(Q=002)*

*If(true)*

*Query(Q-004)*

*Else*

*Output(‘Tidak ditemukan’)*

*Nama Kelas : Admin*

*Nama Operasi : registrasi*

*Algoritma : (Algo-004)*

*Input(username, password, nama, kategori)*

*Query(Q-002)*

*If(username ditemukan dari Q-002)*

*Output(‘Username sudah terdftar’)*

*Else*

*Query(Q-003)*

*Nama Kelas : Admin*

*Nama Operasi : edit*

*Algoritma : (Algo-005)*

*Input(username)*

*Query(Q=002)*

*If(true)*

*Query(Q-004)*

*Else*

*Output(‘Tidak ditemukan’)*

*Nama Kelas : Super\_Admin*

*Nama Operasi : getDataCustomer*

*Algoritma : (Algo-006)*

*Input(Kategori)*

*Query(Q=005)*

*If(true)*

*Tampilkan hasil Q-005*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Super\_Admin*

*Nama Operasi : getDataAdmin*

*Algoritma : (Algo-007)*

*Input(Kategori)*

*Query(Q=005)*

*If(true)*

*Tampilkan hasil Q-005*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Super\_Admin*

*Nama Operasi : deleteCustomer*

*Algoritma : (Algo-008)*

*Input(username)*

*Query(Q=006)*

*Nama Kelas : Super\_Admin*

*Nama Operasi : deleteAdmin*

*Algoritma : (Algo-009)*

*Input(username)*

*Query(Q=006)*

*Nama Kelas : Tiket*

*Nama Operasi : getdata*

*Algoritma : (Algo-010)*

*Query(Q=007)*

*If(true)*

*Tampilkan hasil Q-007*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Tiket*

*Nama Operasi : get\_adata*

*Algoritma : (Algo-011)*

*Input(id)*

*Query(Q=008)*

*If(true)*

*Tampilkan hasil Q-008*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Tiket*

*Nama Operasi : add\_data*

*Algoritma : (Algo-012)*

*Input(id\_tiket, id\_customer, id\_admin, tanggal\_pemesanan, jumlah, total\_harga)*

*Query(Q-009)*

*Nama Kelas : Tiket*

*Nama Operasi : edit\_data*

*Algoritma : (Algo-013)*

*Input(id\_tiket)*

*Query(Q=010)*

*If(true)*

*Query(Q-011)*

*Else*

*Output(‘Tidak ditemukan’)*

*Nama Kelas : Tiket*

*Nama Operasi : drop\_data*

*Algoritma : (Algo-014)*

*Input(id\_tiket)*

*Query(Q=012)*

*Nama Kelas : Acara*

*Nama Operasi : getdata*

*Algoritma : (Algo-015)*

*Query(Q=013)*

*If(true)*

*Tampilkan hasil Q-013*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Acara*

*Nama Operasi : get\_adata*

*Algoritma : (Algo-016)*

*Input(id)*

*Query(Q=014)*

*If(true)*

*Tampilkan hasil Q-014*

*Else*

*Output(‘Data Kosong’)*

*Nama Kelas : Acara*

*Nama Operasi : add\_data*

*Algoritma : (Algo-017)*

*Input(id\_acara, id\_admin, nama\_acara, nama\_acara, jenis, tempat\_acara, waktu\_acara, tanggal\_acara, jumlah, harga)*

*Query(Q-015)*

*Nama Kelas : Acara*

*Nama Operasi : edit\_data*

*Algoritma : (Algo-018)*

*Input(id\_acara)*

*Query(Q=016)*

*If(true)*

*Query(Q-017)*

*Else*

*Output(‘Tidak ditemukan’)*

*Nama Kelas : Acara*

*Nama Operasi : drop\_data*

*Algoritma : (Algo-019)*

*Input(id\_acara)*

*Query(Q=018)*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

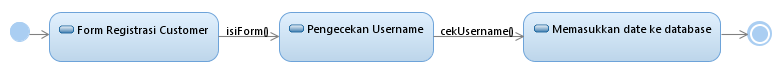
*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* FROM User WHERE username = username AND password = password* | *Mengambil data dari table User yang memiliki username dan password sesuai dengan inputan* |
| *Q-002* | *SELECT \* FROM User WHERE username = username* | *Mengambil data dari User dengan username sesuai dengan inputan* |
| *Q-003* | *INSERT INTO User VALUES (username, password, nama, kategori)* | *Menambah akun kedalam table User* |
| *Q-004* | *UPDATE User SET password = password WHERE username = username* | *Mengubah password User dengan username yang sesuai* |
| *Q-005* | *SELECT \* FROM User WHERE kategori = kategori* | *Mengambil data dari User dengan kategori sesuai dengan inputan* |
| *Q-006* | *DELETE FROM User WHERE username = username* | *Menghapus akun User dari database* |
| *Q-007* | *SELECT \* FROM Tiket* | *Mengambil semua data Tiket yang ada* |
| *Q-008* | *SELECT \* FROM Tiket JOIN Admin USING (id) WHERE id = id* | *Mengambil data tiket yang dinput oleh admin dengan id tertentu* |
| *Q-009* | *INSERT INTO Tiket VALUES (id\_tiket, id\_customer, id\_admin, tanggal\_pemesanan, jumlah, total\_harga)* | *Menambah data Tiket kedalam database* |
| *Q-010* | *SELECT \* FROM Tiket WHERE id\_tiket = id\_tiket* | *Mengambil data tiket dengan id\_tiket yang diinputkan* |
| *Q-011* | *UPDATE Tiket SET ………………………. WHERE id\_tiket = id\_tiket* | *Mengubah………………….. dengan id\_tiket yang sesuai* |
| *Q-012* | *DELETE FROM Tiket WHERE id\_tiket = id\_tiket* | *Menghapus Tiket dari database sesuia dengan id\_tiket* |
| *Q-013* | *SELECT \* FROM Acara* | *Mengambil semua data Acara yang ada* |
| *Q-014* | *SELECT \* FROM Acara JOIN Admin USING (id) WHERE id = id* | *Mengambil data acara yang dinput oleh admin dengan id tertentu* |
| *Q-015* | *INSERT INTO Acara VALUES (id\_acara, id\_admin, nama\_acara, jenis, tempat\_acara, waktu\_acara, tanggal\_acara, jumlah, harga)* | *Menambah acara kedalam database* |
| *Q-016* | *SELECT \* FROM Acara WHERE id\_acara = id\_acara* | *Mengambil data Acara dengan id\_acara yang sesuai* |
| *Q-017* | *UPDATE Acara SET nama\_acara = nama\_acara, jenis = jenis, tempat\_acara = tempat\_acara, waktu\_acara, waktu\_acara, tanggal\_acara = tanggal\_acara, jumlah = jumlah, harga = harga WHERE id\_acara = id\_acara* | *Mengupdate data Acara sesuai dengan id\_acara* |
| *Q-018* | *DELETE FROM Acara WHERE id\_acara = id\_acara* | *Menghapus acara dari database* |

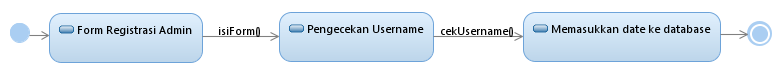
## 

## Diagram Statechart

### Registrasi Customer

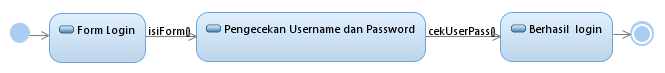


### Registrasi Admin



Gambar 3.5.2.1 Stucture Chart Registrasi Admin

### Login



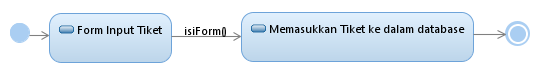
Gambar 3.5.3.1 Stucture Chart Login

### Input Acara



Gambar 3.5.4.1 Stucture Chart Input Acara

### Input Tiket



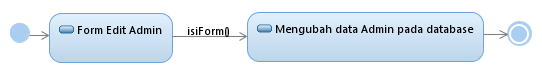
Gambar 3.5.5.1 Stucture Chart Input Tiket

### Edit Customer



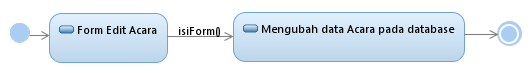
Gambar 3.5.6.1 Stucture Chart Edit Customer

### Edit Admin



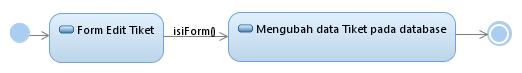
Gambar 3.5.7.1 Stucture Chart Edit Admin

### Edit Acara



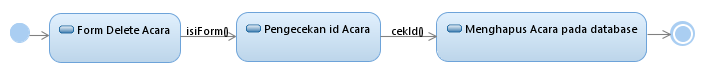
Gambar 3.5.8.1 Stucture Chart Edit Acara

### Edit Tiket



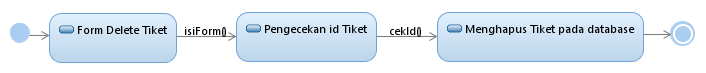
Gambar 3.5.9.1 Stucture Chart Edit Tiket

### Delete Acara



Gambar 3.5.11.1 Stucture Chart Delete Acara

### Delete Tiket



Gambar 3.5.12.1 Stucture Chart Delete Tiket

## Perancangan Antarmuka

### Menu Login

*Antarmuka : 01*

|  |  |
| --- | --- |
| *Screenshot (33)* | *Fungsinya, contoh :*   * *Klik registrasi menuju 02* * *Klik log in menuju 03* |
| *Ukuran 1920 x 10180 Font –24—Roboto Mono—Light, Nama Form: 01* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *textBox1* | *text box* | *Username* | *Wajib diisi username akun yang dimiliki sebagai syarat untuk melakukan log in* |
| *textBox2* | *text box* | *Password* | *Wajib diisi password akun yang dimiliki sebagai syarat untuk melakukan log in* |
| *button1* | *button* | *Registrasi* | *Jika diklik, akan menjalankan fungsi regitrasi() pada ControllerAkun.* |
| *button2* | *button* | *Log in* | *Jika diklik, maka akan menjalankan fungsi cekLogin() pada ControllerAkun.* |

*Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan*

### Menu Registrasi

*Antarmuka : 02*

|  |  |
| --- | --- |
| *Screenshot (32)* | *Fungsinya, contoh :*   * *Klik registrasi, input user, menuju 01* |
| *Ukuran 1920 x 10180 Font –24—Roboto Mono—Light, Nama Form: 02* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *textBox1* | *Text box* | *Username* | *Wajib diisi dengan username akun yang akan dibuat.* |
| *textBox2* | *Text box* | *NIK* | *Wajib diisi dengan NIK pemilik akun yang akan dibuat.* |
| *textBox3* | *Text box* | *No.Handphone* | *Wajib diisi dengan No.Handphone pemilik akun yang akan dibuat.* |
| *textBox4* | *Text box* | *password* | *Wajib diisi dengan password akun yang akan dibuat.* |
| *button1* | *Button* | *Registrasi* | *Jika diklik, akan menjalankan fungsi inpurUser() pada ControllerAkun.* |

*Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan*

### Menu Lihat Acara

*Antarmuka : 03*

|  |  |
| --- | --- |
| *Screenshot (30)* | *Fungsinya, contoh :*   * *Klik icon keranjang menuju 04* * *Klik Pemesanan atau icon dokumen menuju 05* * *Klik tiket menuju 06* * *Klik user -> klik logout menuju 01* * *Klik user -> edit profil menuju 07* * *Klik tambah menuju 04* * *Klik detail menuju 08* |
| *Ukuran 1920 x 10180 Font –24—Roboto Mono—Light, Nama Form: 03* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *button1* | *Button* | *Acara* | *Jika diklik, akan menjalankan fungsi lihatAcara() pada ControllerAcara* |
| *button2* | *Button* | *Keranjang* | *Jika diklik, akan menjalankan fungsi lihatKeranjang(id\_customer) pada ControllerTiket* |
| *button3* | *Button* | *Pemesanan* | *Jika diklik, akan menjalankan fungsi pemesanan(id\_customer) pada ControllerTiket* |
| *button4* | *Button* | *Tiket* | *Jika diklik, akan menjalankan fungsi lihatPesanan(id\_customer) pada ControllerTiket* |
| *button5* | *Button* | *tambah* | *Jika diklik, akan menjalankan fungsi tambahAcara(id\_customer) pada ControllerTiket* |
| *button6* | *Buttone* | *detail* | *Jika diklik, akan menjalankan lihatDetailAcara(id-acara) pada ControllerAcara* |

*Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan*

### Menu Pemesanan

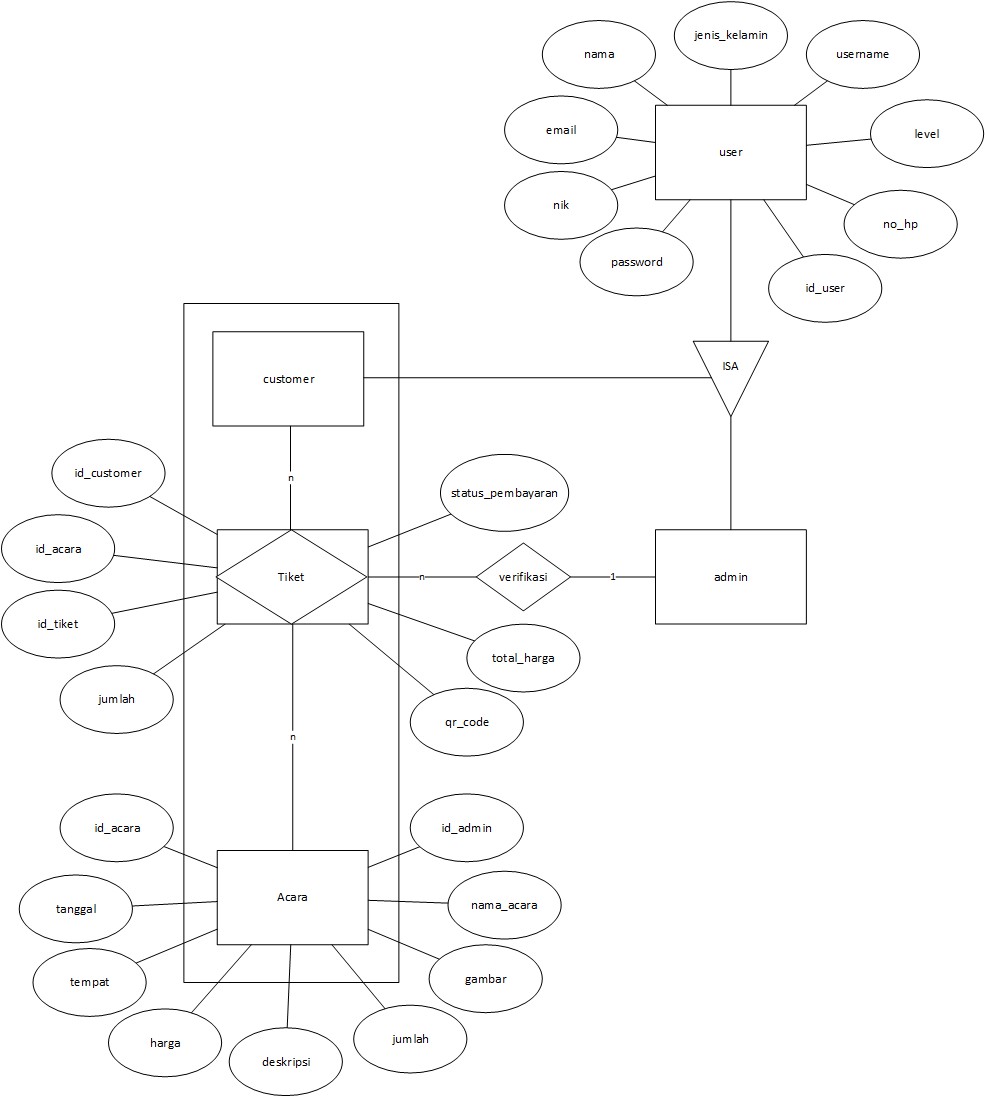
*Antarmuka : 04*

|  |  |
| --- | --- |
| Screenshot (31) | *Fungsinya, contoh :*   * *Klik icon acara menuju 03* * *Klik icon keranjang menuju 04* * *Klik icon tiket menuju 06* * *Klik hapus, delete acara* * *Klik detail, menuju 08* * *Klik user -> klik logout menuju 01* * *Klik user -> edit profil menuju 07* |
| *Ukuran 1920 x 1080 Font –24—Roboto Mono—Light, Nama Form: 05* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *button1* | *Button* | *Acara* | *Jika diklik, akan menjalankan fungsi lihatAcara() pada ControllerAcara* |
| *button2* | *Button* | *Keranjang* | *Jika diklik, akan menjalankan fungsi lihatKeranjang(id\_customer) pada ControllerTiket* |
| *button3* | *Button* | *Pemesanan* | *Jika diklik, akan menjalankan fungsi pemesanan(id\_customer) pada ControllerTiket* |
| *button4* | *Button* | *Tiket* | *Jika diklik, akan menjalankan fungsi lihatPesanan(id\_customer) pada ControllerTiket* |
| *button5* | *Button* | *tambah* | *Jika diklik, akan menjalankan fungsi tambahAcara(id\_customer) pada ControllerTiket* |
| *button6* | *Buttone* | *detail* | *Jika diklik, akan menjalankan lihatDetailAcara(id-acara) pada ControllerAcara* |

## Perancangan Representasi Persistensi Kelas

### ER Diagram



Gambar 3.6.1.1 Entity Relationship Diagram

### Skema Relasi

# Matriks Kerunutan

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Usecase Terkait** | **Kelas** |
| FR-01 | Registrasi | Model\_User, User, Main, Registrasi |
| FR-02 | Login | Model\_User, User,Main, Login |
| FR-03 | Lihat Acara | Model\_User, User, Main, Acara, Model\_Acara, lihatAcara |
| FR-04 | Pesanan | Model\_User ,User, Main, Acara, Model\_Acara, detail\_tiket, beliTiket |
| FR-05 | Verifikasi Pembayaran | Model\_User ,User, Main, Acara, Model\_Acara, Tiket,Model\_Tiket, pesanan |
| FR-06 | Pengelolaan Acara | Model\_User ,User, Main, Acara, Model\_Acara |
| FR-07 | Lihat Acara | Model\_User ,User, Main, Acara, Model\_Acara, list |
| FR-08 | Tambah Acara | Model\_User ,User, Main, Acara, Model\_Acara, new\_form |
| FR-09 | Edit Acara | Model\_User ,User, Main, Acara, Model\_Acara, edit\_form |
| FR-10 | Hapus Acara | Model\_User ,User, Main, Acara, Model\_Acara, list |

Tabel 4.1 Matriks Kerunutan

# Lampiran

## Rancangan Lingkungan Implementasi

Sistem yang akan dibangun diimplementasikan dalam kebutuhan perangkat lunak sebagai berikut :

1. Sistem Operasi : Windows
2. Scripting Languange : HTML, Javascript, CSS dan PHP
3. Web Server : Apache
4. DBMS : MySQL
5. Development tools : Sublime, Visual Basic, phpmyadmin(database)
6. Framework : Code Igniter

## Use Case Skenario

#### *Usecase Scenario #1*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat Acara | |
| Deskripsi | Fungsi ini digunakan oleh user melihat acara yang ada | |
| Pre-Kondisi | Aktor telah memiliki akun dan sudah melakukan login | |
| Post-Kondisi | Aktor dapat melihat acara yang tersedia | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Membuka menu acara |  |
|  | 1. Menampilkan acara yang ada |

Tabel 5.2.1 Use Case Skenario #1

#### *Usecase Scenario #2*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Registrasi | |
| Deskripsi | Fungsi ini digunakan Aktor untuk membuat akun | |
| Pre-Kondisi | Aktor belum memiliki akun | |
| Post-Kondisi | Aktor telah memiliki akun dan bisa login | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Membuka laman web |  |
| 1. Memilih registrasi |  |
| 1. Memasukkan data yang diperlukan |  |
|  | 1. Memverifikasi data yang diinput |
|  | 1. Jika sesuai melanjutkan proses |
|  | 1. Jika tidak sesuai kembali ke proses selanjutnya |
| 1. Langsung masuk ke halaman home aplikasi |  |

Tabel 5.2.2 Use Case Skenario #2

#### *Usecase Scenario #3*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Login | |
| Deskripsi | Fungsi ini agar aktor dapat mengunakan aplikasi | |
| Pre-Kondisi | Aktor telah melakukan registrasi dan belum login | |
| Post-Kondisi | Aktor melakukan login dan dapat masuk ke dalam aplikasi | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor melakukan login |  |
| 1. Aktor mengklik button login |  |
|  | 1. Jika username dan password sesuai, maka lanjut ke tahap berikutnya |
|  | 1. Jika masih salah kembali ke proses satu |
|  | 1. Masuk ke tampilan utama |

Tabel 5.2.3 Use Case Skenario #3

#### *Usecase Scenario #4*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Pesanan | |
| Deskripsi | Fungsi ini digunakan aktor untuk melakukan pemesanan acara | |
| Pre-Kondisi | Aktor belum memesan acara | |
| Post-Kondisi | Aktor telah memesan acara | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor mencari tiket yang dinginkan |  |
| 1. Aktor mengklik button pesan |  |
|  | 1. Melakukan pengelolaan terhadap tiket yang dipesan, dan memberikan kode pembayaran |
| 1. Aktor melakukan pembayaran |  |
| 1. Menginput bukti pembayaran |  |
|  | 1. Memverifikasi bukti pembayaran |
|  | 1. Jika sesuai, lanjut ke proses selanjutnnya |
|  | 1. Jika tidak, kembali ke proses 5 |
|  |  | 1. Pemesanan berhasil, mengirimkan hasil verifikasi |

Tabel 5.2.4 Use Case Skenario #4

#### *Usecase Scenario #5*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Verifikasi Pembayaran | |
| Deskripsi | Aktor melakukan verifikasi terhadap input bukti pembayaran | |
| Pre-Kondisi | Aktor belum melakukan verifikasi | |
| Post-Kondisi | Aktor sudah melakukan verifikasi | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor melihat ke menu pesanan |  |
|  | 1. Menampilkan pesanan yang ada |
| 1. Aktor melakukan verifikasi |  |
|  | 1. Aktor membenarkan verifikasi |  |
|  |  | 1. QR code diberikan ke customer |

Tabel 5.2.5 Use Case Skenario #5

#### *Usecase Scenario #6*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Pengelolaan acara | |
| Deskripsi | Fungsi ini digunakan oleh admin untuk melihat, menambah, hapus dan edit acara | |
| Pre-Kondisi | Admin belum mengupdate acara | |
| Post-Kondisi | Admin sudah mengupdate acara yang ada | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor melakukan melihat, menambah atau menghapus atau edit acara |  |
| 1. Aktor mengklik button submit |  |
|  | 1. Jika sudah sesuai, lanjut ke proses selanjutnya |
|  | 1. Jika belum sesuai, kembali ke proses awal |
|  | 1. Kembali ke menu sebelumnya |

Tabel 5.2.6 Use Case Skenario #6

#### *Usecase Scenario #7*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat Pesanan | |
| Deskripsi | Fungsi ini digunakan oleh admin untuk melihat pesanan yang sudah ada | |
| Pre-Kondisi | Admin belum melihat pesanan | |
| Post-Kondisi | Admin sudah melihat pesanan | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka menu lihat pesanan |  |
|  | 1. Menampilkan menu lihat pesanan |

Tabel 5.2.7 Use Case Skenario #7

#### *Usecase Scenario #8*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Tambah Acara | |
| Deskripsi | Fungsi ini digunakan oleh admin untuk menambah acara. | |
| Pre-Kondisi | Aktor belum menambah acara | |
| Post-Kondisi | Aktor sudah menambah acara | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka menu tambah acara |  |
|  | 1. Sistem menampilkan menu tambah acara |
| 1. Aktor menginput data yang diperlukan |  |
| 1. Klik button |  |
|  | 1. Jika sudah sesuai maka acara akan ditambah dan masuk ke skenario selanjutnya |
|  | 1. Jika belum sesuai, kembali ke view tambah acara |
|  | 1. Jika berhasil, kembali ke menu lihat pesanan |

Tabel 5.2.8 Use Case Skenario #8

#### *Usecase Scenario #9*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Edit Acara | |
| Deskripsi | Fungsi ini digunakan oleh admin untuk mengedit acara yang ada | |
| Pre-Kondisi | Admin belum mengedit acara | |
| Post-Kondisi | Admin sudah mengedit acara | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka menu edit acara |  |
|  | 1. Sistem menampilkan menu edit acara |
| 1. Aktor menginput data yang diperlukan |  |
| 1. Klik button |  |
|  | 1. Jika sudah sesuai maka acara akan ditambah dan masuk ke skenario selanjutnya |
|  | 1. Jika belum sesuai, kembali ke view tambah acara |
|  | 1. Jika berhasil, kembali ke menu lihat pesanan |

Tabel 5.2.9 Use Case Skenario #9

#### *Usecase Scenario #10*

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Hapus Acara | |
| Deskripsi | Fungsi ini digunakan oleh admin untuk menghapus acara yang ada | |
| Pre-Kondisi | Admin belum menghapus acara | |
| Post-Kondisi | Admin sudah menghapus acara | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka menu lihat acara |  |
|  | 1. Sistem menampilkan menu lihat acara |
| 1. Aktor memilih acara yang ingin dihapus. |  |
| 1. Klik button hapus |  |
|  | 1. Jika sudah, acara akan terhapus. |

Tabel 5.2.10 Use Case Skenario #10

## Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Kebutuhan | Fungsi | Deskripsi |
| 1 | FR-01 | Registrasi | Pengguna melakukan pendaftaran. |
| 2 | FR-02 | *Login* | Pengguna dan admin melakukan login untuk masuk ke aplikasi**.** |
| 3 | FR-03 | Lihat Acara | Pengguna dapat melihat acara yang ada. |
| 4 | FR-04 | Pesanan | Pelanggan melakukan pemesanan acara dan menginputkan bukti pembayaran setelah melakukan pembayaran. |
| 5 | FR-05 | Verifikasi Pembayaran | Admin melakukan verifikasi pembayaran terhadap input bukti transfer dari pelanggan. |
| 6 | FR-06 | Pengelolaan Acara | Admin dapat mengelola acara yaitu lihat pesanan, edit acara, tambah acara, hapus acara. |
| 7 | FR-07 | Lihat Pesanan | Admin melihat pesanan yang ada. |
| 8 | FR-08 | Tambah Acara | Admin dapat menambah acara. |
| 9 | FR-09 | Edit Acara | Admin dapat mengedit acara yang sudah ada. |
| 10 | FR-10 | Hapus Acara | Admin dapat menghapus acara yang sudah ada. |