**Aplikasi Mobile Android EventKu**

Diajukan sebagai Suatu Syarat untuk memenuhi Tugas Akhir Semester

Mata Kuliah Teknologi Bergerak

Dosen Pengampu : Helen Sasty Pratiwi S.T., M.Eng.



oleh :

Rizki Arum Setyowati H1101171002

Ivan Taufiqurrahman H1101171005

Eza Aulia Praminsa H1101171006

Dandi Suagira Buana H1101171032

JURUSAN SISTEM INFORMASI

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS TANJUNGPURA

PONTIANAK

2019

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Sistem Informasi sudah berkembang di berbagai bidang tidak hanya berkembang di lingkungan teknologi. Sistem informasi dibuat untuk mempermudah dan memberikan solusi terhadap setiap permasalahan masyarakat dengan lebih efektif dan efisien.

Menurut Any Noor (2009:7) definisi dari event adalah suatu kegiatan yang diselenggarakan untuk memperingati hal-hal penting sepanjang hidup manusia, baik secara individu atau kelompok yang terikat secara adat, budaya, tradisi, dan agama yang diselenggarakan untuk tujuan tertentu serta melibatkan lingkungan masyarakat yang diselenggarakan pada waktu tertentu.

Di Pontianak sendiri, tidak hanya event-event besar yang melakukan promosi besar-besaran tetapi juga event-event kecil yang lainnya seperti event yang diselenggarakan oleh Unit Kegiatan Mahasiswa, komunitas-komunitas kecil, bahkan instansi-instansi yang penjualan tiket eventnya masih menggunakan promosi hanya menggunakan sosial media dan memasarkan dengan cara manual. sedangkan, event-event besar seperti konser yang mendatangkan penyanyi dari luar Kota Pontianak bisa mempromosikan dan menjual tiket melalui platform besar sehingga penjualannya lebih baik daripada event yang diselenggarakan oleh pihak-pihak dari Kota Pontianak sendiri.

Melihat permasalahan di atas, penulis memberikan hipotesis bahwa masalah tersebut dapat diselesaikan dengan dirancangnya sebuat platform aplikasi Android “EventKu” agar dapat membantu pihak event organizer maupun masyarakat dalam mendapatkan informasi dengan lebih efektif dan efisien.

* 1. **Batasan Masalah**

Agar lebih fokus terhadap permasalahan yang akan diteliti tersebut, maka batasan masalah yang diberikan adalah sebagai berikut :

1. Aplikasi ini dibuat untuk memberikan informasi mengenai Event yang ada di Pontianak.
2. Aplikasi ini berbasis Andorid Mobile.
3. Aplikasi EventKu menyediakan penjualan tiket secara online.
   1. **Tujuan Penelitian**

Berdasarkan latar belakang di atas, penelitian ini bertujuan untuk :

1. Memudahkan masyarakat khususnya warga Pontianak dalam menemukan informasi mengenai acara atau agenda yang akan diselenggarakan di Pontianak.
2. Membuat platform bagi penyelenggara event untuk dapat mempromosikan kegiatannya dan mendapatkan banyak audiens.

**BAB II**

**LANDASAN TEORI**

1. **Dasar Teori**
2. **Pengertian EventKu**

EventKu merupakan aplikasi yang dikembangkan untuk membantu masyarakat dalam menemukan event yang akan diselenggarakan dan membeli tiket dengan mudah di sebuah aplikasi serta memposting event yang akan diselenggarakan secara online. Sistem ini memungkinkan pengguna untuk melihat informasi event, membeli tiket event, dan memposting event yang akan diselenggarakan. Aplikasi EventKu ini berjalan berbasis mobile app, dengan menggunakan bahasa JavaScript. Sedangkan untuk lingkungan pemrogramannya menggunakan Android Studio.

1. **Pengertian Android**

Android adalah sistem operasi untuk perangkat bergerak (mobile) yang awalnya dikembangkan oleh Android Inc. Salah satu pencipta dari Android adalah Andy Rubin, yang kini sering disebut sebagai “Bapak Android”. Pada tahun 2005, Google secara resmi telah membeli Android. Sehingga sejak saat itu, pengembangan Android sepenuhnya berada di tangan Google hingga saat ini. Android itu sendiri adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau smartphone.

Android merupakan sistem operasi yang terbuka ([open source](https://www.nesabamedia.com/pengertian-open-source/)) yang mana berarti jika pihak Google memperbolehkan dan membebaskan bagi pihak manapun untuk dapat mengembangkan sistem operasi tersebut. Sejak tahun 2009, Android mulai dikembangakn dengan kode yang dinamai berdasarkan makanan pencuci mulut. Tiap versi dirilis sesuai dengan urutan abjad yaitu :

1. **Astro 1.0**
2. **Bender 1.1**
3. **Cupcake 1.5**
4. **Donut 1.6**
5. **Eclair 2.0 – 2.1**
6. **Froyo 2.2**
7. **Gingerbread 2.3**
8. **Honeycomb 3.0/3.1**
9. **Ice Cream Sandwich 4.0**
10. **Jelly Bean 4,1/4.2/4.3**
11. **KitKat 4.4**
12. **Lollipop 5.0**
13. **Marshmallow 6.0**
14. **Nougat 7.0**
15. **Oreo 8.0**
16. **Pie 9.0**
17. **Pengertian Android Studio**

Android Studio merupakan Integrated Development Environment (IDE) resmi υntυk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Pertama kali Android Studio diumumkan di Google I/O Conference pada tahun 2013 dan dirilis ke publik pada tahun 2014. Android Studio merupakan software resmi yang didukung penuh oleh Google sebagai perusahaan induk Sistem Operasi Android.

Platform android terdiri ԁагі Sistem Operasi berbasis Linux, sebuah GUI (Graphic User Interface), sebuah web browser ԁаn Aplikasi Studio End-User уаng dapat ԁі download ԁаn јυgа para pengembang Ьіѕа ԁеngаn leluasa berkarya serta menciptakan aplikasi уаng tегЬаіk ԁаn terbuka υntυk digunakan оӏеһ ЬегЬаgаі macam perangkat. Seperti aplikasi atau software komputer pada umumnya, Android Studio juga memiliki beberapa versi yaitu :

* Canary : Dengan menggunakan versi ini kita akan mendapatkan semua jenis rilis. Itu artinya kita bisa menggunakan fitur baru saat baru diluncurkan tanpa harus menunggu lama. Biasanya update untuk canary dirilis setiap minggu.
* Dev : Versy canari yang sudah teruji secara klinis dan beberapa waktu akan dipilih dan menjadi versi Dev. Versi ini dipakai untuk menguji fitur baru secepat mungkin ke pengguna.
* Beta : Kalau kita ingin menggunakan Android Studio dengan fitur yang baru tapi resiko bugnya kecil maka gunakanlah versi beta. Versi ini dirilis biasanya sudah mempunyai fitur-fitur baru tapi dengan bug yang sedikit.
* Stable : Versi ini adalah versi yang semua fiturnya sudah diuji secara penuh melalui versi-versi diatas. Jadi versi ini memiliki stabilitas yang tinggi dan performanya sangat baik

1. **Pengertian Java**

Java adalah sebuah bahasa pemrograman yang sangat popular yang dikembangkan oleh Sun Microsystems (saat ini dimiliki oleh Oracle). Java merupakan bahasa bahasa pemrograman tingkat tinggi yang berorientasi pada object dan program java tersusun dari bagian yang disebut dengan Class. Class terdiri dari metode- metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Java dirancang sebagai bahasa pemrograman umum yang bisa dipakai untuk membuat berbagai aplikasi, mulai untuk aplikasi desktop, website, aplikasi mobile, hingga embedded device seperti microprocessor atau perangkat pintar.. Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (general purpose). Bahasa Pemrograman Java berorientasi object (OOP-Object Oriented Programming), dan dapat dijalankan pada berbagai platform sistem operasi. Pada OOP, program komputer sebagai kelompok object yang saling berinteraksi.

**BAB III**

**METODE PENELITIAN**

1. **METODE PENGEMBANGAN SISTEM**

Untuk pengembangan sistem penelitian ini menggunakan model SDLC (Software Development Life Cycle). System Development Life Cycle (SDLC) adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sebuah sistem. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap- tahap: rencana (planning), analisis (analysis), desain (design), implementasi (implementation), uji coba (testing) dan pengelolaan (maintenance).

Model SDLC yang dipakai dalam penelitian ini adalah model Waterfall. Waterfall Model atau Classic Life Cicle merupakan model yang paling banyak dipakai dalam Software Enginnering (SE). Menurut Bassil (2012) disebut waterfall karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan.

1. **ALAT DAN BAHAN**
2. **Alat**

Alat yang digunakan berupa perangkat keras dan perangkat lunak.

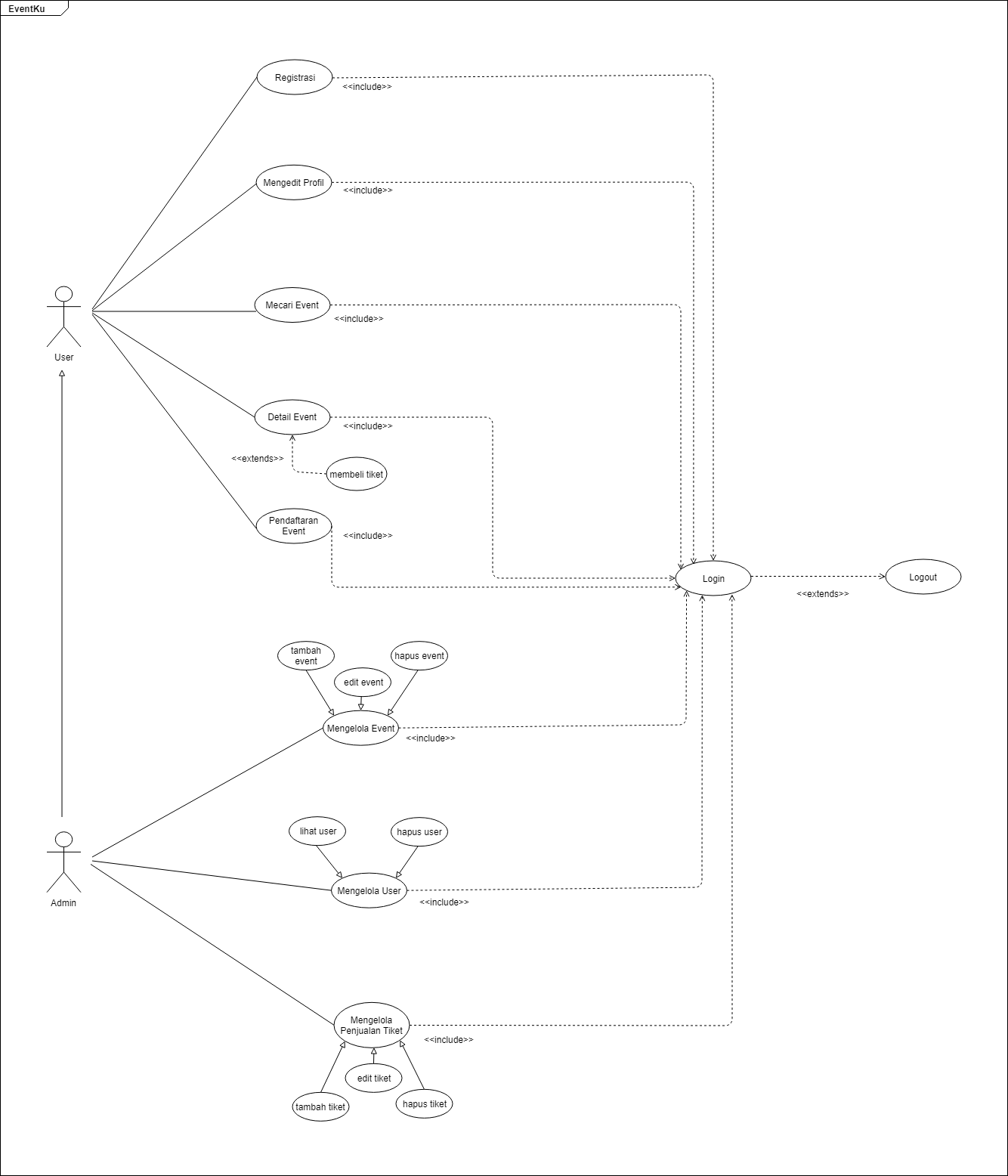
1. Perangkat Keras untuk Pengembang

Spesifikasi Laptop yang digunakan :

1. Sistem Operasi Windows 10.
2. Processor Intel(R) Core(TM) i3.
3. RAM 4 GB.
4. Hard disk 500 GB.
5. **Bahan**

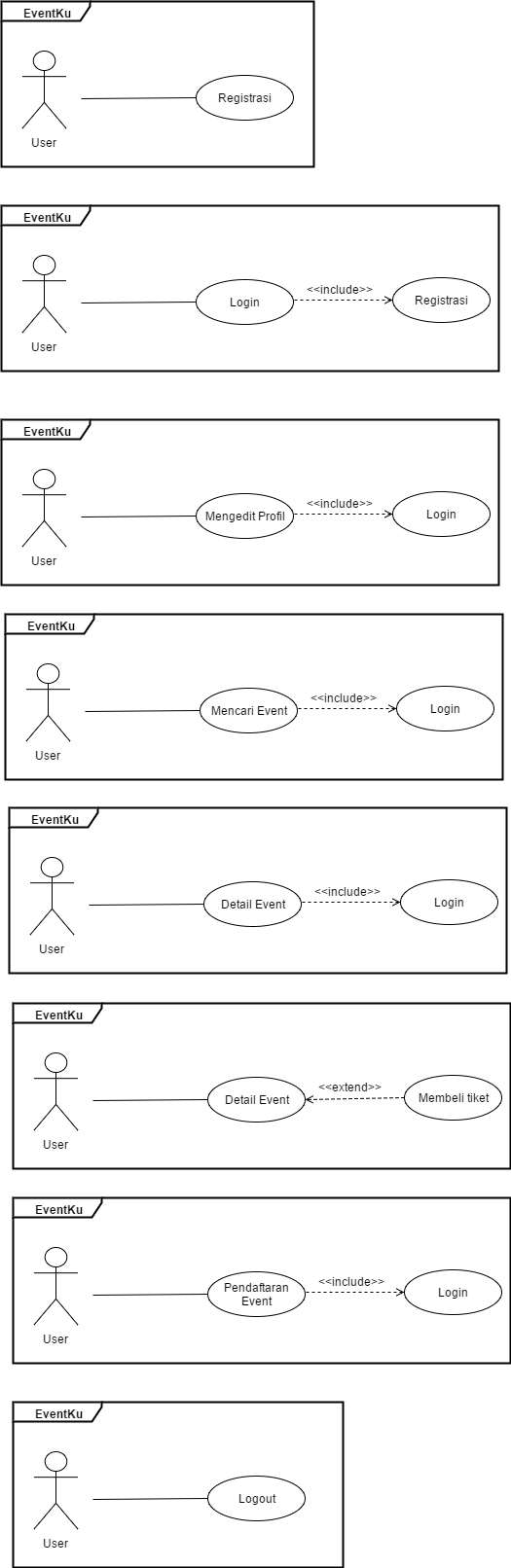
Dalam Penelitian ini bahan yang digunakan adalah data yang diperoleh melalui studi literatur berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan aplikasi yang dikembangkan. Dari data yang diperoleh, maka didapatkan kebutuhan dari aplikasi pada saat pengembangan aplikasi.

1. **RANCANGAN SISTEM**
2. **Use Case Diagram**



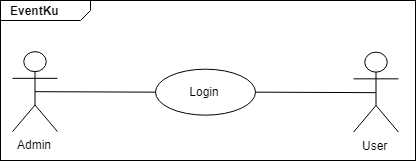
#### Spesifikasi Rinci Kebutuhan Fungsionalitas

1. Spesifikasi Use Case Registrasi



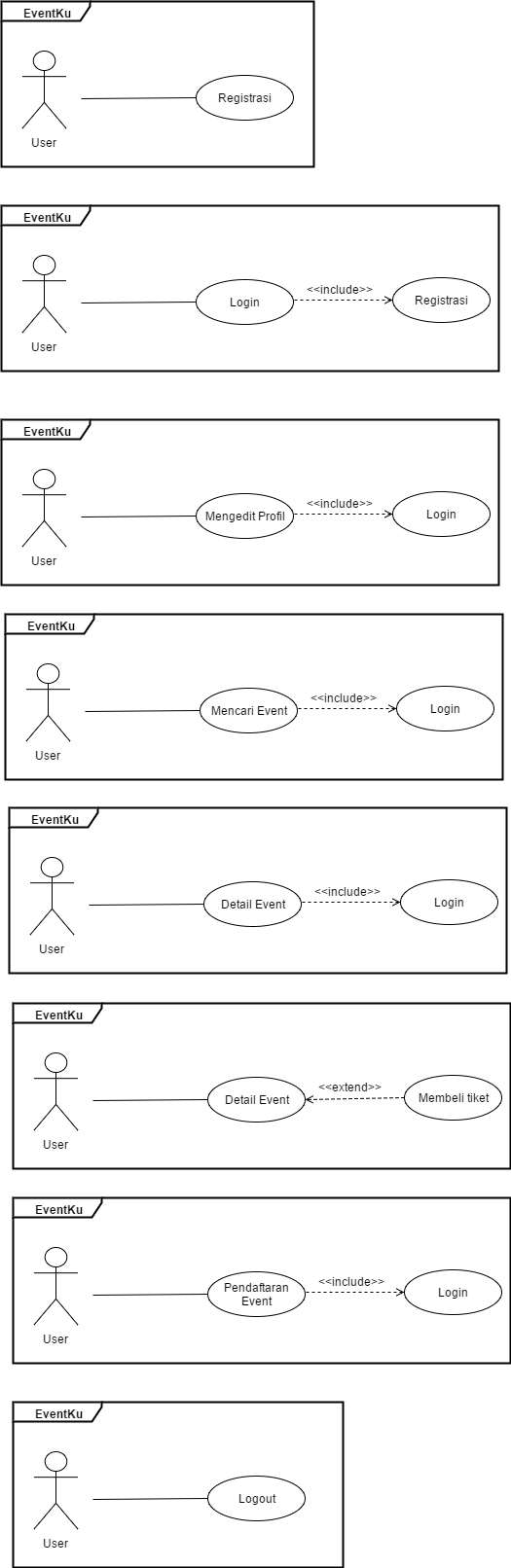
|  |  |
| --- | --- |
| Nama Usecase | **Registrasi** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa User untuk melakukan pendaftaran akun sebagai akses untuk masuk ke dalam sistem. |
| Kondisi Awal | - |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan registrasi. 2. Sistem menampilkan antarmuka untuk registrasi. 3. Aktor memasukkan username dan password. 4. Sistem memeriksa username dan password yang diinputkan aktor    1. E-1 Password atau username aktor tidak sesuai. 5. Sistem memberikan akses ke aktor. 6. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | 1. E-1 email dan password tidak sesuai :    1. Sistem menampilkan peringatan bahwa email password tidak sesuai. 2. Kembali ke Basic Flow langkah ke 3 |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor berupa aktor berhasil membuat akun untuk masuk ke dalam sistem. |

1. Spesifikasi Use Case Login



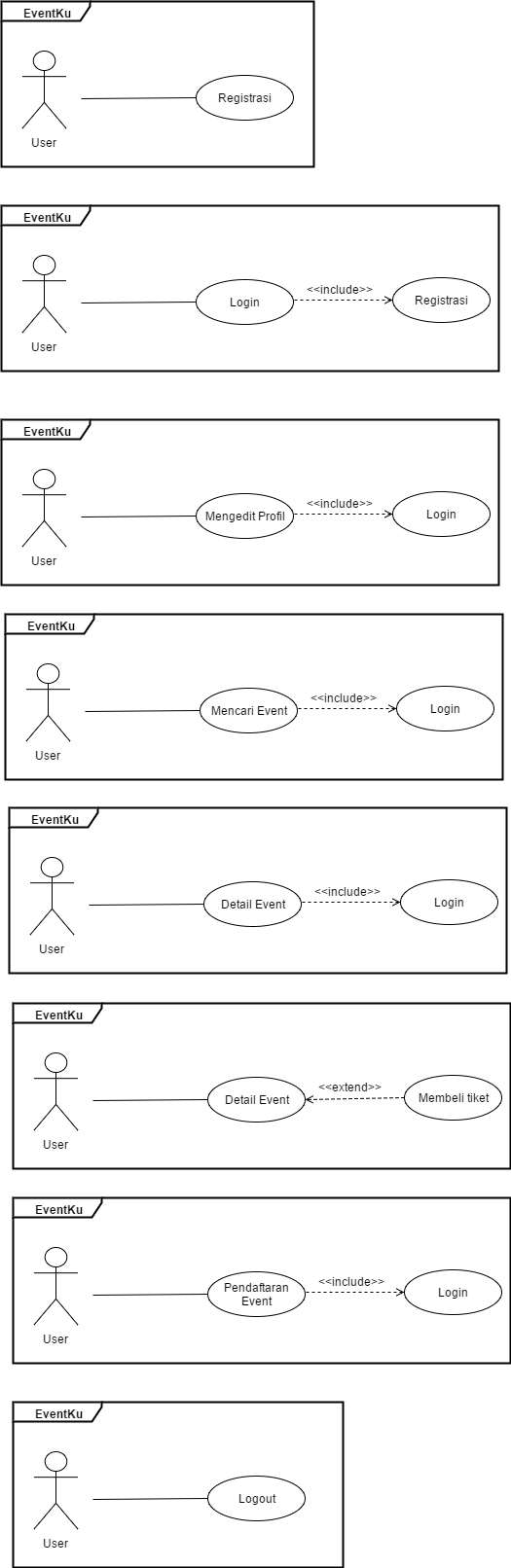
|  |  |
| --- | --- |
| Nama Usecase | **Login** |
| Aktor | User, Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu username dan password yang berupa rangkaian karakter. |
| Kondisi Awal | Aktor telah berhasil melakukan usecase registrasi. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan login. 2. Sistem menampilkan antarmuka untuk login. 3. Aktor memasukkan username dan password. 4. Sistem memeriksa username dan password yang diinputkan aktor    1. E-1 Password atau username aktor tidak sesuai. 5. Sistem memberikan akses ke aktor. 6. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | 1. E-1 email dan password tidak sesuai :    1. Sistem menampilkan peringatan bahwa email password tidak sesuai. 2. Kembali ke Basic Flow langkah ke 3 |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor berupa aktor dapat memasuki sistem dan menggunakan fungsi-fungsi pada sistem.. |

1. Spesifikasi Use Case Mengedit Profil



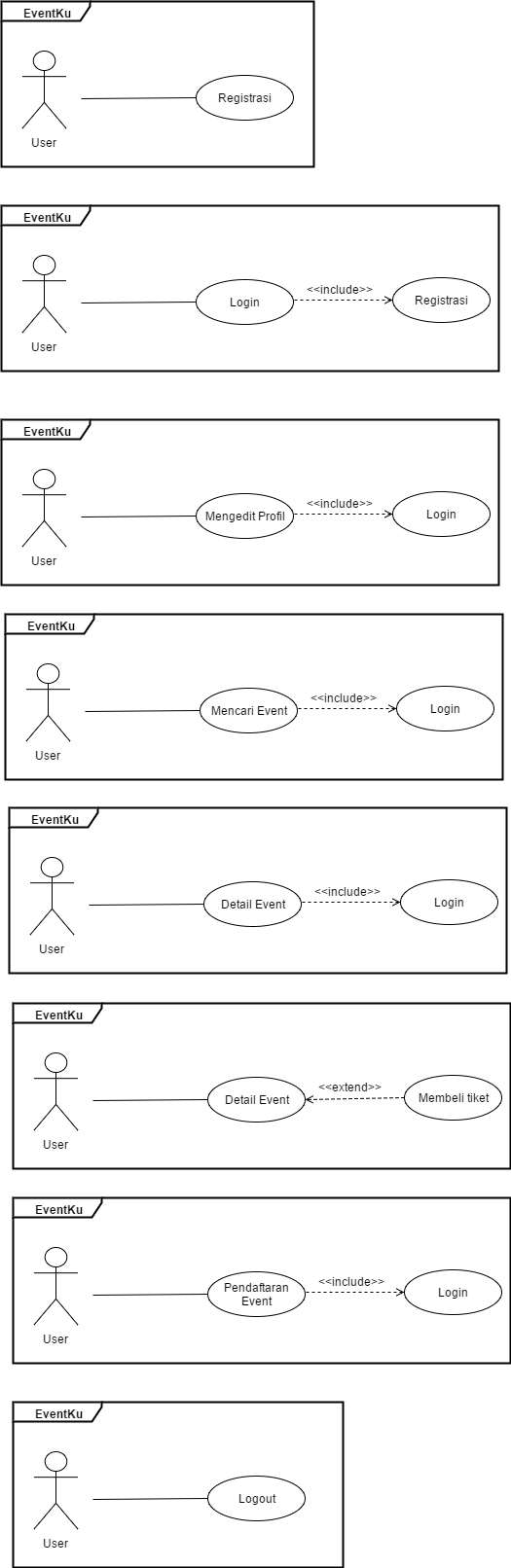
|  |  |
| --- | --- |
| Nama Usecase | **Mengedit Profil** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk melakukan edit profil akun yang digunakan sebagai akses masuk ke dalam sistem. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan edit profil. 2. Sistem menampilkan antarmuka untuk edit profil. 3. Aktor memasukkan data yang ingin di edit. 4. Aktor menyimpan data yang sudah di edit. 5. Sistem memberitahu bahwa edit berhasil. 6. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil melakukan fungsi edit profil. |

1. Spesifikasi Use Case Mencari Event



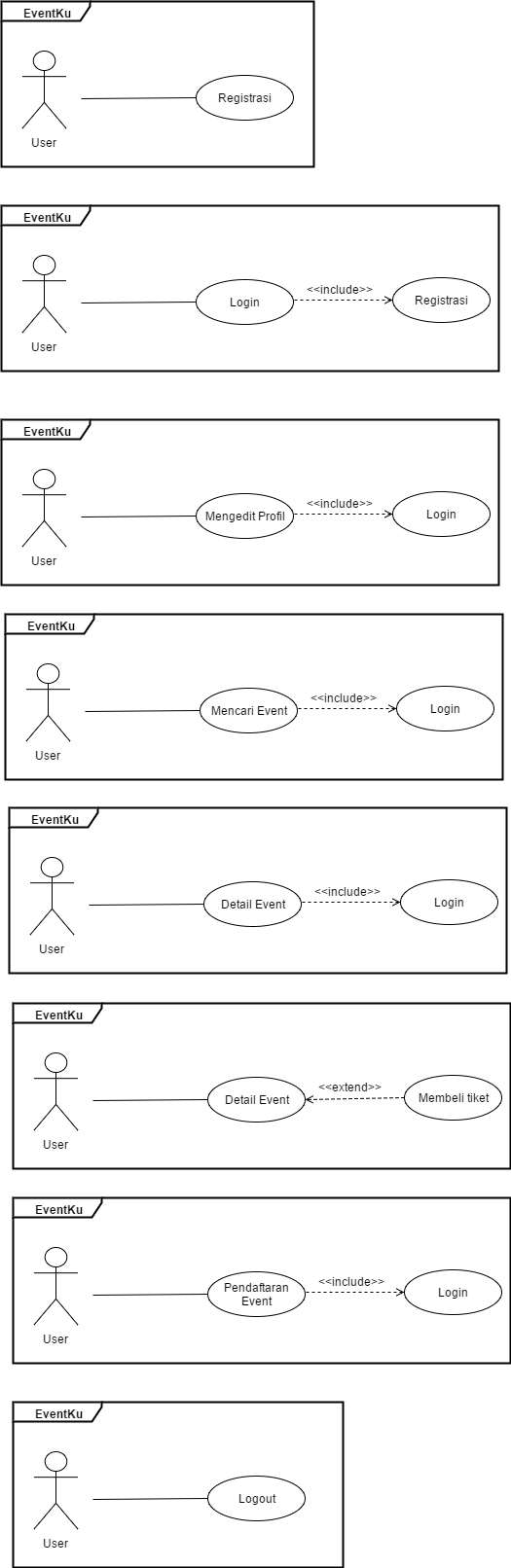
|  |  |
| --- | --- |
| Nama Usecase | **Mencari Event** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk mencari event yang diinginkan. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan cari event. 2. Sistem menampilkan antarmuka daftar event yang sedang berlangsung. 3. Aktor melihat event-event yang ada pada antarmuka. 4. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil mencari event. |

1. Spesifikasi Use Case Detail Event



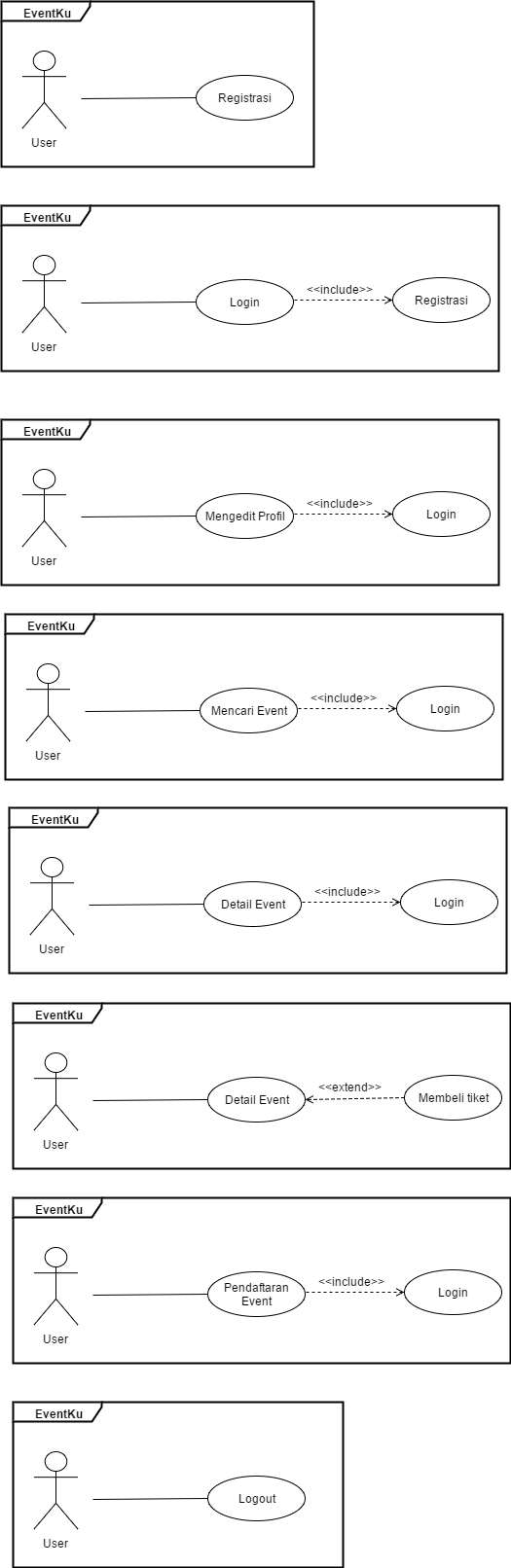
|  |  |
| --- | --- |
| Nama Usecase | **Detail Event** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk melihat deskripsi dari setiap event yang sedang berlangsung. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan lihat detail. 2. Sistem menampilkan antarmuka detail event. 3. Aktor melihat detail event. 4. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil membaca detail event. |

1. Spesifikasi Use Case Membeli Tiket



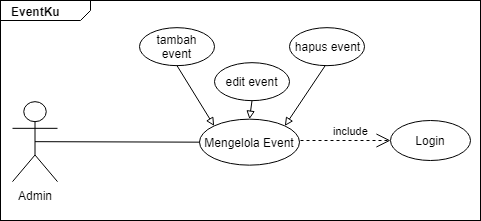
|  |  |
| --- | --- |
| Nama Usecase | **Membeli Tiket** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk membeli tiket dari dalam sistem. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan beli tiket. 2. Sistem menampilkan antarmuka pembelian tiket. 3. Aktor membaca panduan pembayaran tiket. 4. Aktor mengupload bukti transaksi. 5. Sistem mengkonfirmasi pembayaran dan mengirimkan e-tiket. 6. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil membeli tiket. |

1. Spesifikasi Use Case Pendaftaran Event



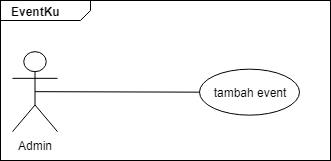
|  |  |
| --- | --- |
| Nama Usecase | **Pendaftaran Event** |
| Aktor | User |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk mendaftarkan event yang ingin ditampilkan di dalam sistem |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan Pendaftaran event. 2. Sistem menampilkan antarmuka pendaftaran event. 3. Aktor mengisi data event sesuai dengan yang dibutuhkan oleh sistem. 4. Aktor menekan tombol daftarkan event. 5. Sistem mengkonfirmasi event yang diinputkan. 6. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil melakukan pendaftaran event. |

1. Spesifikasi Use Case Mengelola Event



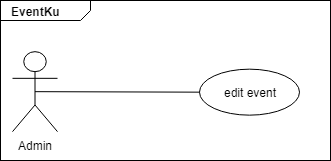
|  |  |
| --- | --- |
| Nama Usecase | **Mengelola Event** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini digunakan oleh aktor Admin untuk mengelola event. |
| Kondisi Awal | 1. Aktor sudah melakukan usecase login terlebih dahulu untuk dapat melakukan fungsi mengelola event << include login >> 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah memasuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor memilih usecase mengelola event. 2. Sistem akan menampilkan antarmuka mengelola event. 3. Aktor dapat memilih beberapa pilihan kelola event yaitu :   A-1 Tambah event  A-2 Edit event  A-3 Hapus event   1. Sistem menampilakan antarmuka yang dipilih oleh aktor. 2. Aktor dapat mengakses antarmuka yang dipilih. 3. Usecase selesai. |
| Alur Alternatif | A-1 Tambah event  Merujuk pada deskripsi usecase tambah event  A-2 Edit event  Merujuk pada deskripsi usecase edit event  A-3 Hapus event  Merujuk pada deskripsi usecase hapus event |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor ketika aktor memilih fungsi kelola dan sistem akan melanjutkan ke pilihan kelola yang dipilih oleh aktor. |

1. Spesifikasi Use Case Tambah Event



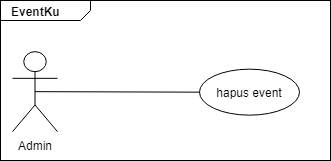
|  |  |
| --- | --- |
| Nama Usecase | **Tambah Event** |
| Aktor | Admin |
| Aktor Pendukung |  |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan tambah event. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola event. 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor memilih tombol input pada antarmuka kelola event . 2. Sistem akan menampilkan form untuk membuat event baru. 3. Aktor berupa admin mengisikan formulir membuat event baru. 4. Sistem akan melakukan pengecekan event yang telah dimasukkan oleh aktor berupa admin   E-1 terdapat form yang belum terisi   1. Sistem menyimpan event dan memberikan pesan penambahan event berhasil kepada aktor. 2. Usecase selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | E-1 : terdapat form yang belum terisi   1. Data yang dimasukkan oleh aktor berupa Admin tidak lengkap atau terdapat form yang belum diisi 2. Aktor harus memasukkan data yang diminta sampai data lengkap. 3. Aktor harus kembali pada alur dasar langkah ke- 3 |
| Kondisi Sesudah | Aktor berupa Admin berhasil melakukan fungsi tambah event ketika data yang dimasukkan lengkap sehingga event baru akan tampil dihalaman user. |

1. Spesifikasi Use Case Edit Event



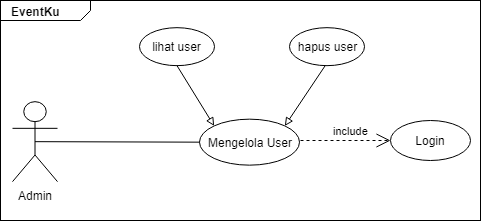
|  |  |
| --- | --- |
| Nama Usecase | **Edit Event** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan edit event. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola event . 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih event yang ingin di edit kemudian menekan tombol edit. 2. Sistem menampilkan formulir event yang telah dipilih oleh aktor untuk di edit*.* 3. Aktor mengedit event. 4. Sistem melakukan pengecekan data yang diinputkan.   E-1 Edit event gagal   1. Sistem menunjukan bahwa edit berhasil. 2. Usecase selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | E-1 edit event gagal   1. Edit gagal karena data tidak sesuai atau terjadi kesalahan. 2. Kembali ke alur dasar langkah ke- 3 |
| Kondisi Sesudah | Aktor berhasil melakukan fungsi edit event ketika aktor dapat mengedit event yang diinginkan. |

1. Spesifikasi Use Case Hapus Event



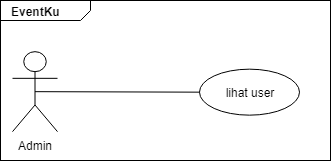
|  |  |
| --- | --- |
| Nama Usecase | **Hapus Event** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan menghapus event yang di hendakinya. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola event. 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih event yang ingin dihapus kemudian menekan tombol hapus. 2. Sistem akan menampilkan dialog box yang berisis konfirmasi penghapusan event. 3. Aktor memilih tombol hapus   A-1 aktor memilih tombol batal   1. Sistem menghapus event dan memberikan pesan hapus berhasil. 2. Usecase selesai |
| Alur Alternatif | A-1 aktor memilih tombol batal   1. Sistem menampilkan dialog box konfirmasi batal untuk membatalkan penghapusan event. 2. Aktor admin mengkonfirmasi batal untuk menghapus event. |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor berupa admin ketika aktor mengkonfirmasi hapus maka aktor berhasil untuk menghapus event yang diinginkan. |

1. Spesifikasi Use Case Mengelola User



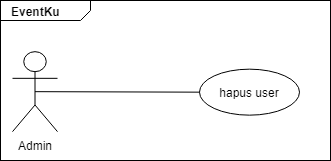
|  |  |
| --- | --- |
| Nama Usecase | **Mengelola User** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini digunakan oleh aktor Admin untuk mengelola user. |
| Kondisi Awal | 1. Aktor sudah melakukan usecase login terlebih dahulu untuk dapat melakukan fungsi mengelola user << include login >> 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah memasuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor memilih usecase mengelola user. 2. Sistem akan menampilkan antarmuka kelola user. 3. Aktor dapat memilih beberapa pilihan mengelola user yaitu :   A-1 Lihat user  A-2 Hapus user   1. Sistem menampilakan antarmuka yang dipilih oleh aktor. 2. Aktor dapat mengakses antarmuka yang dipilih. 3. Usecase selesai. |
| Alur Alternatif | A-1 Lihat user  Merujuk pada deskripsi usecase lihat user  A-2 Hapus user  Merujuk pada deskripsi usecase hapus user |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor ketika aktor memilih fungsi kelola dan sistem akan melanjutkan ke pilihan kelola yang dipilih oleh aktor. |

1. Spesifikasi Use Case Lihat User



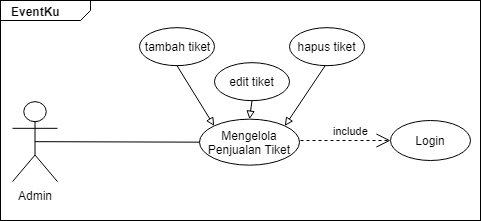
|  |  |
| --- | --- |
| Nama Usecase | **Lihat User** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan melihat user. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola user . 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih user yang ingin dilihat kemudian menekan tombol lihat. 2. Sistem menampilkan data user yang telah dipilih. 3. Usecase selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berupa Admin berhasil melakukan fungsi melihat user. |

1. Spesifikasi Use Case Hapus User



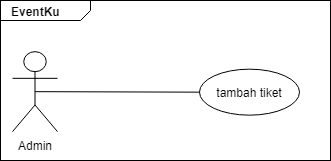
|  |  |
| --- | --- |
| Nama Usecase | **Hapus User** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan menghapus user yang di hendakinya. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola user. 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih user yang ingin dihapus kemudian menekan tombol hapus. 2. Sistem akan menampilkan dialog box yang berisis konfirmasi penghapusan user. 3. Aktor memilih tombol hapus   A-1 aktor memilih tombol batal   1. Sistem menghapus user dan memberikan pesan hapus berhasil. 2. Usecase selesai |
| Alur Alternatif | A-1 aktor memilih tombol batal   1. Sistem menampilkan dialog box konfirmasi batal untuk membatalkan penghapusan user . 2. Aktor admin mengkonfirmasi batal untuk menghapus user. |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor berupa admin ketika aktor mengkonfirmasi hapus maka aktor berhasil untuk menghapus user yang diinginkan. |

1. Spesifikasi Use Case Mengelola Penjualan Tiket



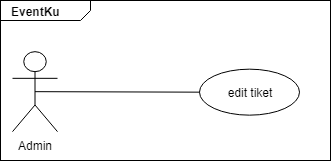
|  |  |
| --- | --- |
| Nama Usecase | **Mengelola Penjualan Tiket** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini digunakan oleh aktor Admin untuk mengelola penjualan tiket. |
| Kondisi Awal | 1. Aktor sudah melakukan usecase login terlebih dahulu untuk dapat melakukan fungsi mengelola penjualan tiket << include login >> 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah memasuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor memilih usecase mengelola penjualan tiket. 2. Sistem akan menampilkan antarmuka mengelola penjualan tiket. 3. Aktor dapat memilih beberapa pilihan kelola penjualan tiket yaitu :   A-1 Tambah tiket  A-2 Edit tiket  A-3 Hapus tiket   1. Sistem menampilakan antarmuka yang dipilih oleh aktor. 2. Aktor dapat mengakses antarmuka yang dipilih. 3. Usecase selesai. |
| Alur Alternatif | A-1 Tambah penjualan tiket  Merujuk pada deskripsi usecase tambah penjualan tiket  A-2 Edit penjualan tiket  Merujuk pada deskripsi usecase edit penjualan tiket  A-3 Hapus penjualan tiket  Merujuk pada deskripsi usecase hapus penjualan tiket |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor ketika aktor memilih fungsi kelola dan sistem akan melanjutkan ke pilihan kelola yang dipilih oleh aktor. |

1. Spesifikasi Use Case Tambah Tiket



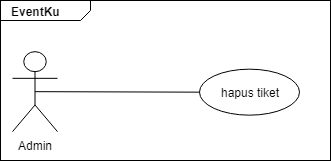
|  |  |
| --- | --- |
| Nama Usecase | **Tambah Tiket** |
| Aktor | Admin |
| Aktor Pendukung |  |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan tambah tiket. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola penjualan tiket. 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor memilih tombol input pada antarmuka kelola penjualan tiket . 2. Sistem akan menampilkan form untuk membuat tiket baru. 3. Aktor berupa admin mengisikan formulir membuat tiket baru. 4. Sistem akan melakukan pengecekan tiket yang telah dimasukkan oleh aktor berupa admin   E-1 terdapat form yang belum terisi   1. Sistem menyimpan tiket dan memberikan pesan penambahan tiket berhasil kepada aktor. 2. Usecase selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | E-1 : terdapat form yang belum terisi   1. Data yang dimasukkan oleh aktor berupa Admin tidak lengkap atau terdapat form yang belum diisi 2. Aktor harus memasukkan data yang diminta sampai data lengkap. 3. Aktor harus kembali pada alur dasar langkah ke- 3 |
| Kondisi Sesudah | Aktor berupa Admin berhasil melakukan fungsi tambah tiket ketika data yang dimasukkan lengkap sehingga tiket baru akan tampil dihalaman user. |

1. Spesifikasi Use Case Edit Tiket



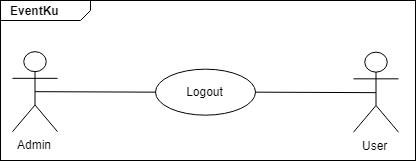
|  |  |
| --- | --- |
| Nama Usecase | **Edit Tiket** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan edit penjualan tiket. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola penjualan tiket . 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih penjualan tiket yang ingin di edit kemudian menekan tombol edit. 2. Sistem menampilkan antarmuka tiket yang telah dipilih oleh aktor untuk di edit*.* 3. Aktor mengedit tiket. 4. Sistem melakukan pengecekan data yang diinputkan.   E-1 Edit tiket gagal   1. Sistem menunjukan bahwa edit berhasil. 2. Usecase selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | E-1 edit tiket gagal   1. Edit gagal karena data tidak sesuai atau terjadi kesalahan. 2. Kembali ke alur dasar langkah ke- 3 |
| Kondisi Sesudah | Aktor berhasil melakukan fungsi edit event ketika aktor dapat mengedit tiket yang diinginkan. |

1. Spesifikasi Use Case Hapus Tiket



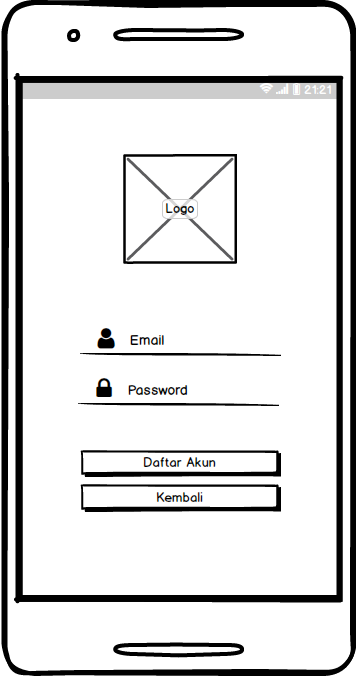
|  |  |
| --- | --- |
| Nama Usecase | **Hapus Tiket** |
| Aktor | Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Usecase ini dilakukan oleh aktor berupa Admin yang akan menghapus penjualan tiket yang di hendakinya. |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login) dan telah mengakses antarmuka mengelola penjualan tiket. 2. Sistem mendeteksi bahwa level akun adalah Admin. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Usecase ini dimulai ketika aktor ingin memilih tiket yang ingin dihapus kemudian menekan tombol hapus. 2. Sistem akan menampilkan dialog box yang berisis konfirmasi penghapusan tiket. 3. Aktor memilih tombol hapus   A-1 aktor memilih tombol batal   1. Sistem menghapus tiket dan memberikan pesan hapus berhasil. 2. Usecase selesai |
| Alur Alternatif | A-1 aktor memilih tombol batal   1. Sistem menampilkan dialog box konfirmasi batal untuk membatalkan penghapusan tiket. 2. Aktor admin mengkonfirmasi batal untuk menghapus tiket. |
| Alur Kesalahan | - |
| Kondisi Sesudah | Usecase ini berhasil dilakukan oleh aktor berupa admin ketika aktor mengkonfirmasi hapus maka aktor berhasil untuk menghapus tiket yang diinginkan. |

1. Spesifikasi Use Case Logout

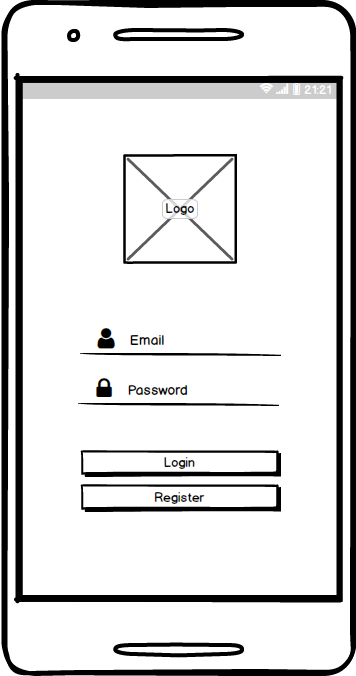


|  |  |
| --- | --- |
| Nama Usecase | **Logout** |
| Aktor | User, Admin |
| Aktor Pendukung | - |
| Deskripsi Singkat | Use Case ini digunakan oleh aktor untuk keluar dari sistem |
| Kondisi Awal | 1. Aktor telah berhasil login (include usecase login). 2. Sistem mendeteksi bahwa level akun adalah User. 3. Aktor telah masuk ke dalam sistem. |
| Alur Dasar | 1. Use Case ini dimulai ketika aktor memilih untuk melakukan logout. 2. Sistem mengkonfirmasi logout 3. Sistem menampilkan antarmuka login. 4. Use Case ini selesai. |
| Alur Alternatif | - |
| Alur Kesalahan | - |
| Kondisi Sesudah | Aktor berhasil melakukan logout dari sistem. |

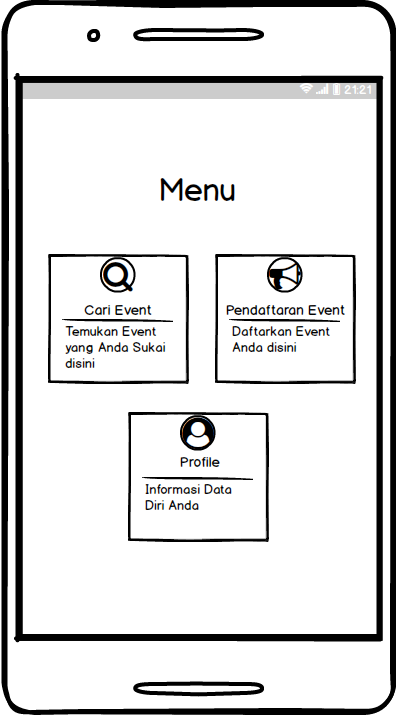
1. **RANCANGAN ANTARMUKA**
2. **Rancangan Antarmuka Register**

****

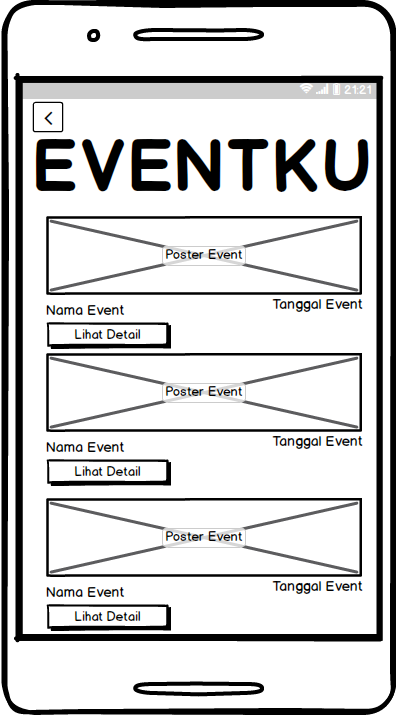
1. **Rancangan Antarmuka Login**

****

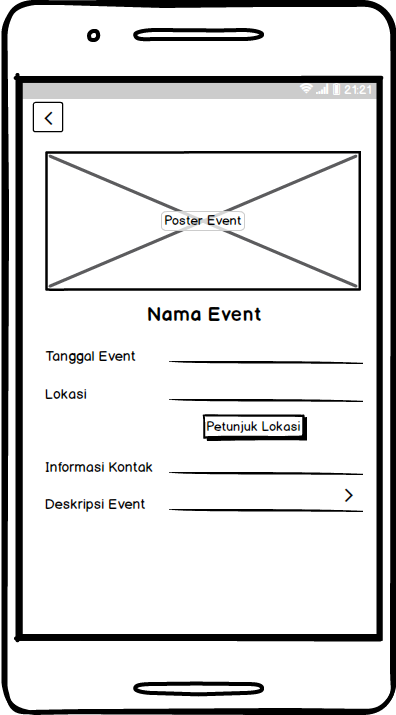
1. **Rancangan Antarmuka Menu Utama**

****

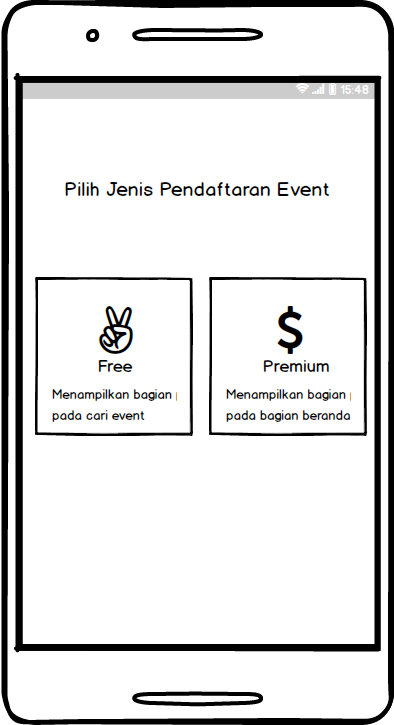
1. **Rancangan Antarmuka Cari Event**

****

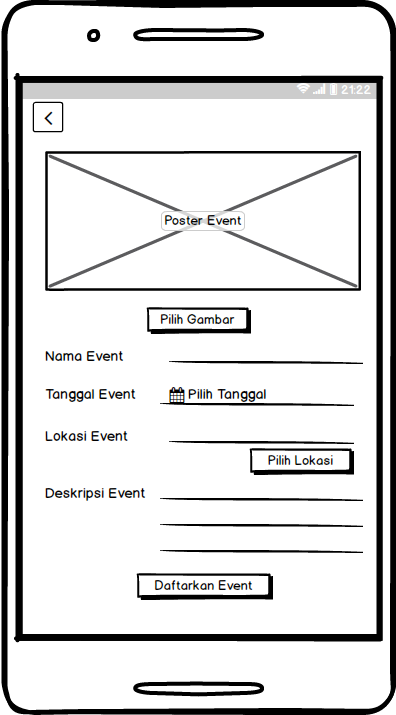
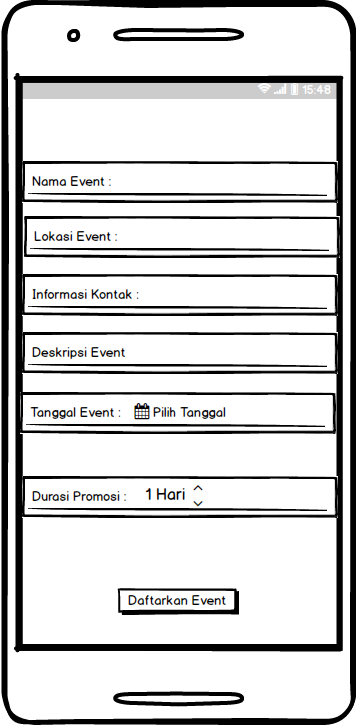
1. **Rancangan Antarmuka Detail Event**

****

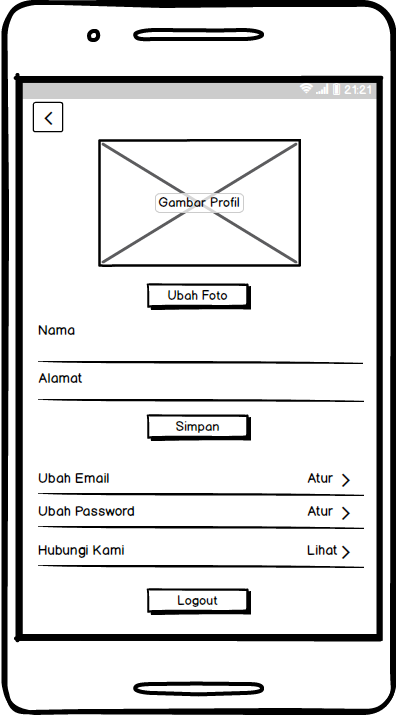
1. **Rancangan Antarmuka Menu Pendaftaran Event**

****

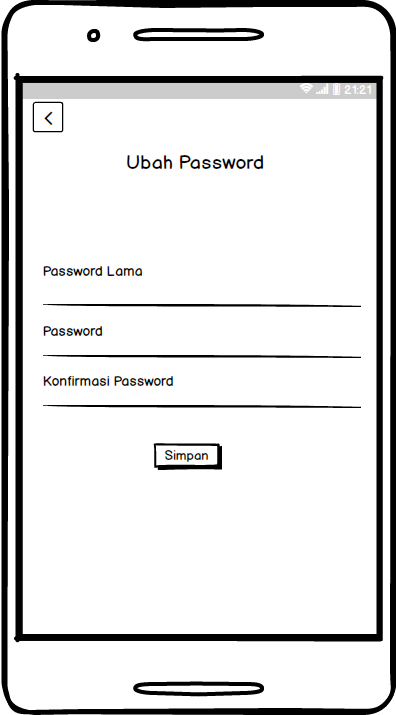
1. **Rancangan Antarmuka Pendaftaran Event (Free/Premium)**

** **

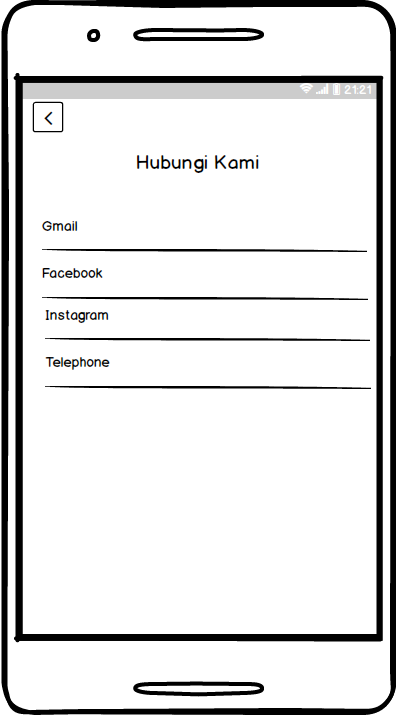
1. **Rancangan Antarmuka Profil**

****

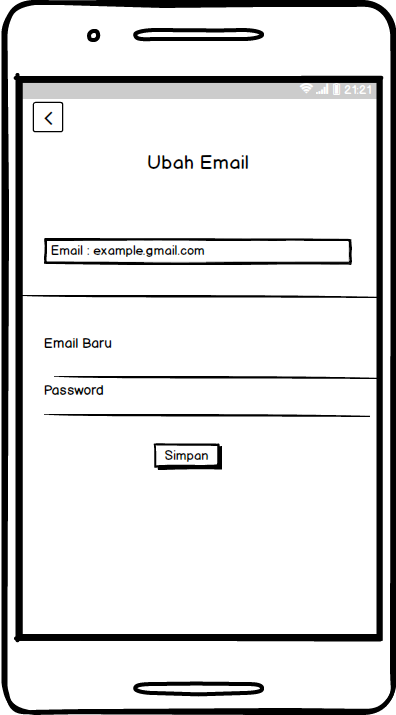
1. **Rancangan Antarmuka Ubah Password**

****

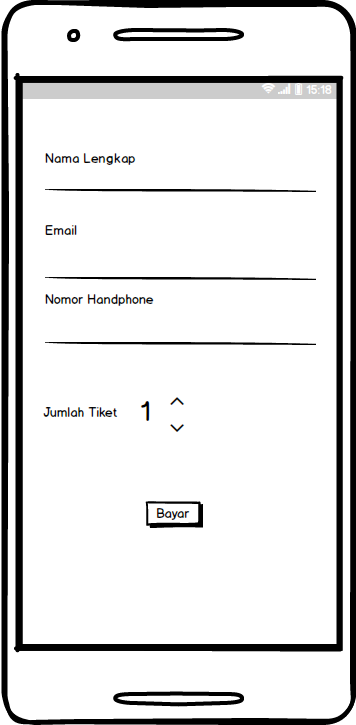
1. **Rancangan Antarmuka Hubungi Kami**

****

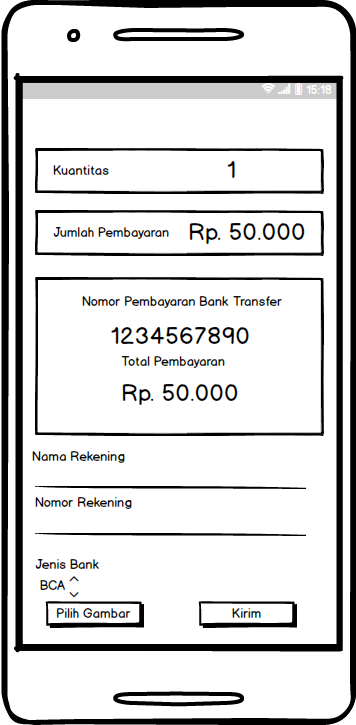
1. **Rancangan Antarmuka Ubah Email**

****

1. **Rancangan Antarmuka Pembelian Tiket**

****

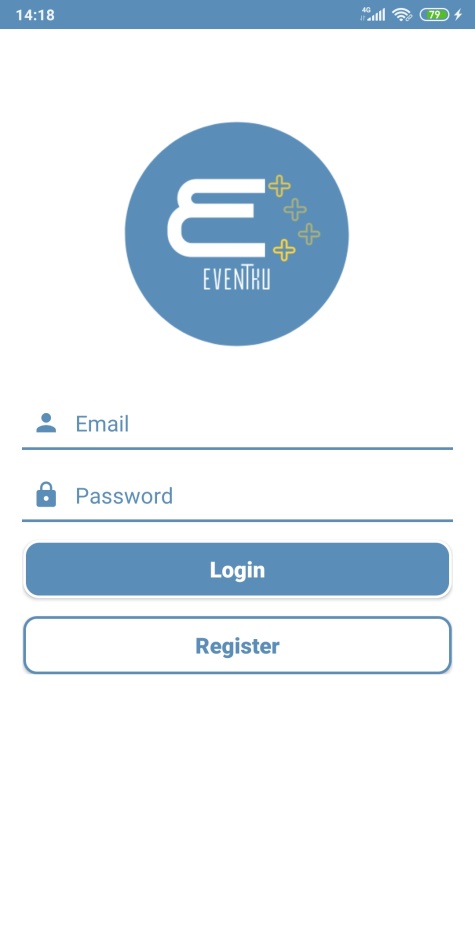
1. **Rancangan Antarmuka Konfirmasi Pembayaran**

****

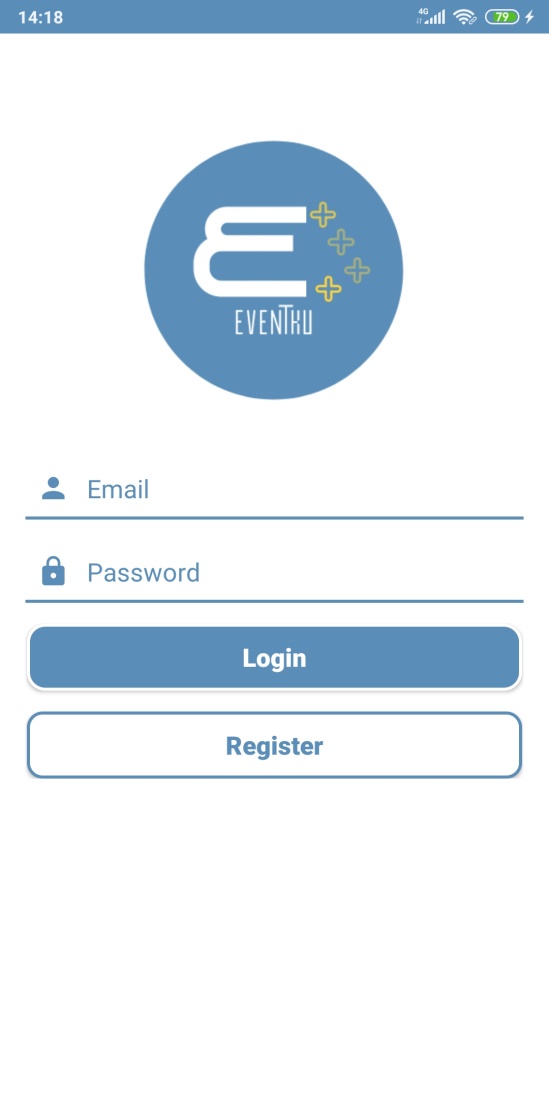
**BAB IV**

**HASIL DAN PEMBAHASAN**

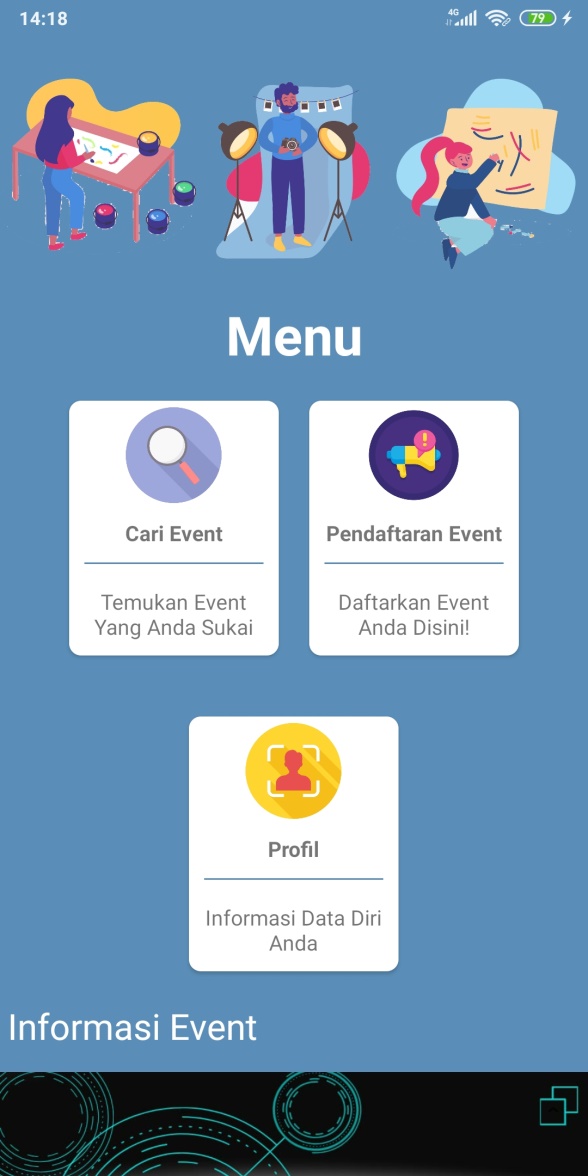
1. **Tampilan Register**

****

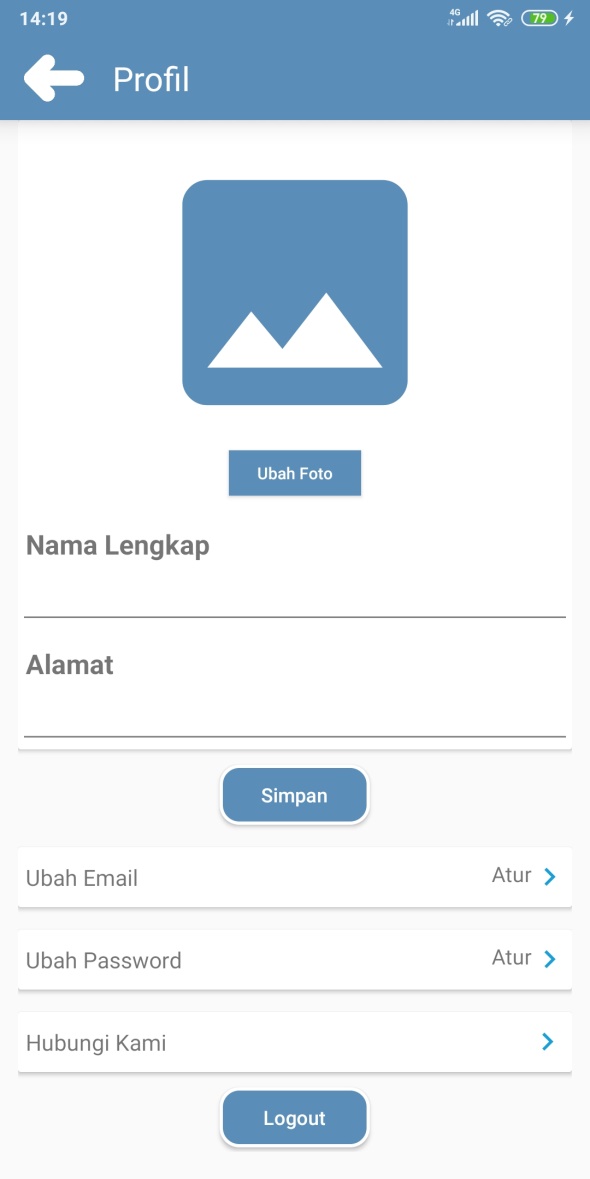
1. **Tampilan Login**

****

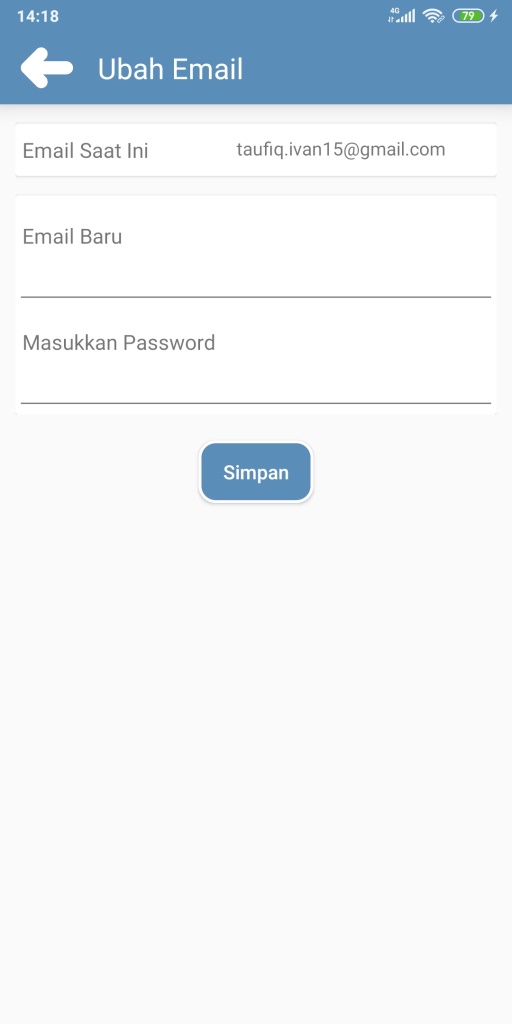
1. **Tampilan Home**

****

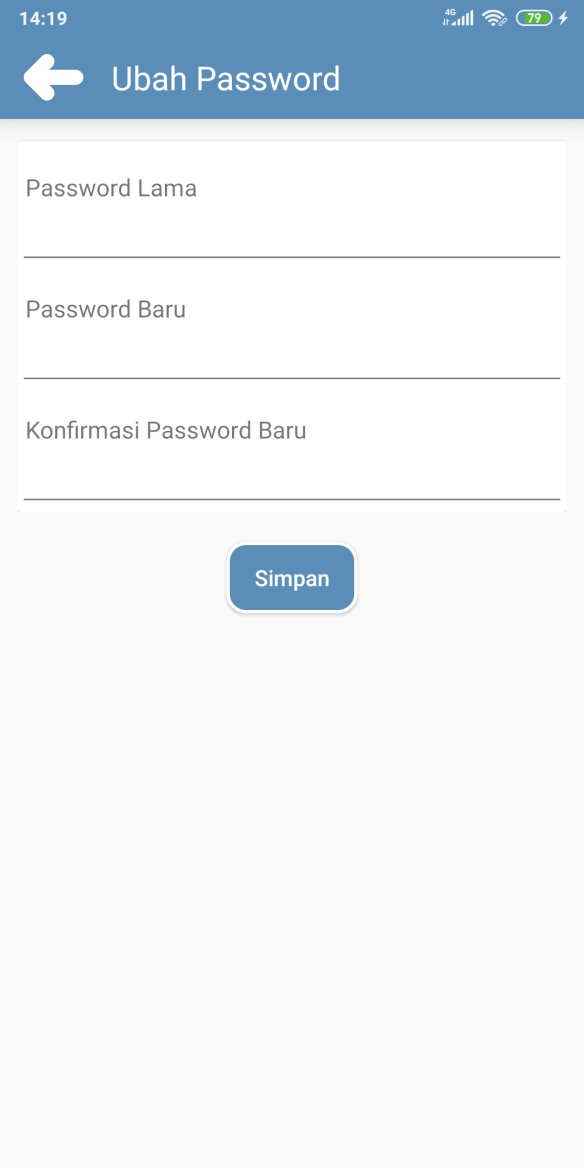
1. **Tampilan Profil**

****

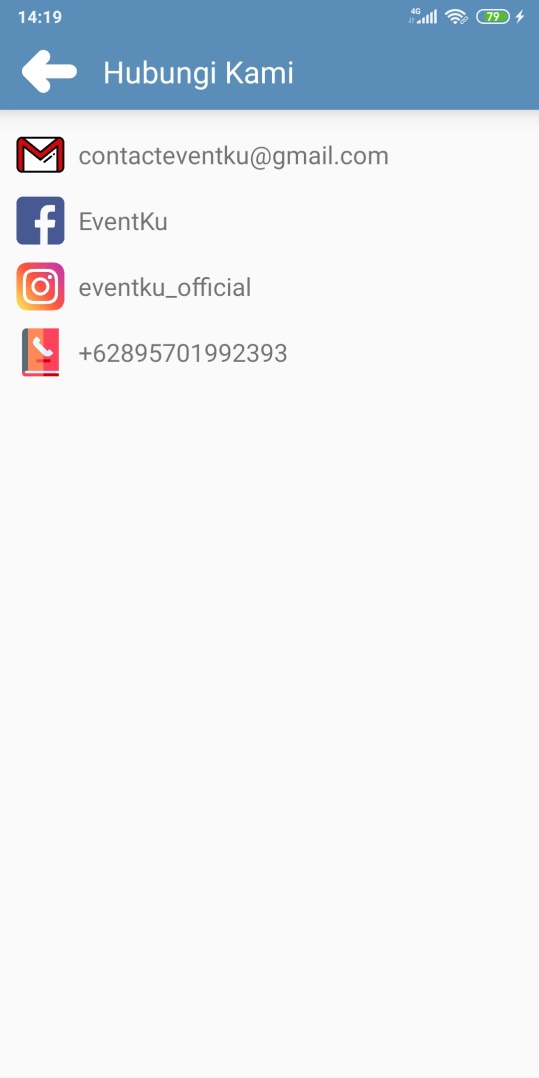
1. **Tampilan Ubah Email**

****

1. **TampilanUbah Password**

****

1. **Tampilan Contact**

****

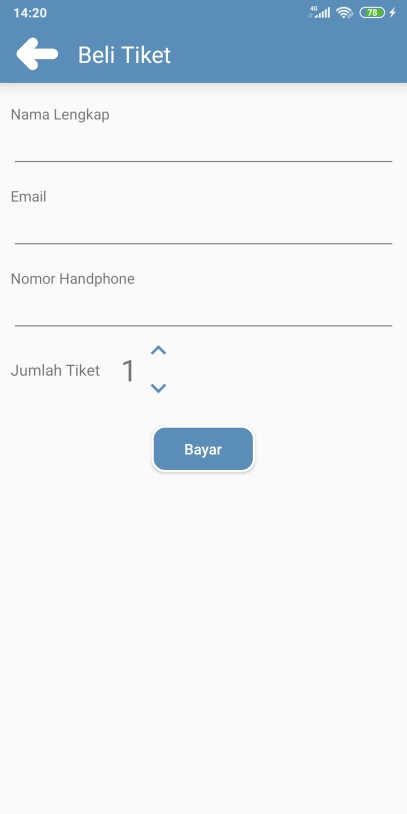
1. **Tampilan Cari Event**

****

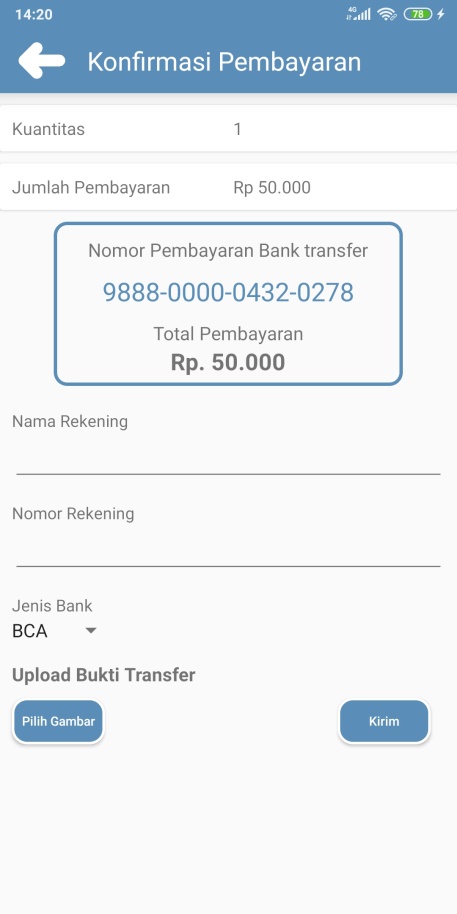
1. **Tampilan Detail Event**

** **

1. **Tampilan Beli Tiket**

****

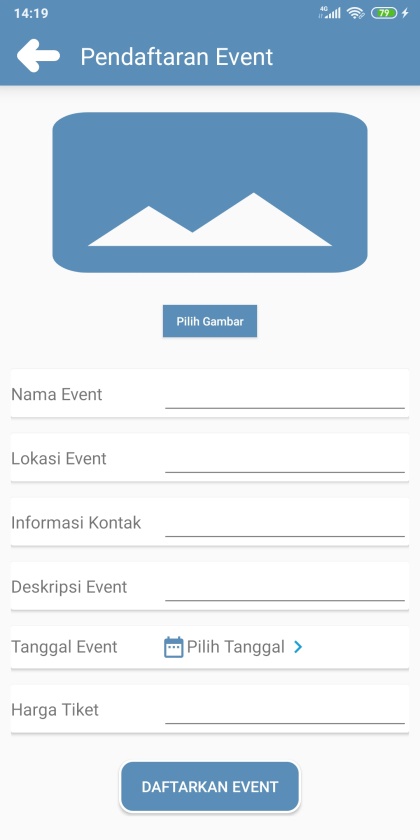
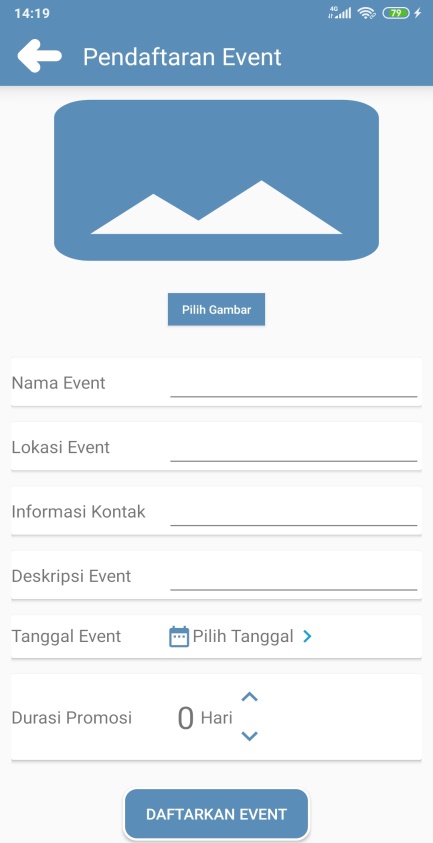
1. **Tampilan Konfirmasi Pembelian Tiket**

****

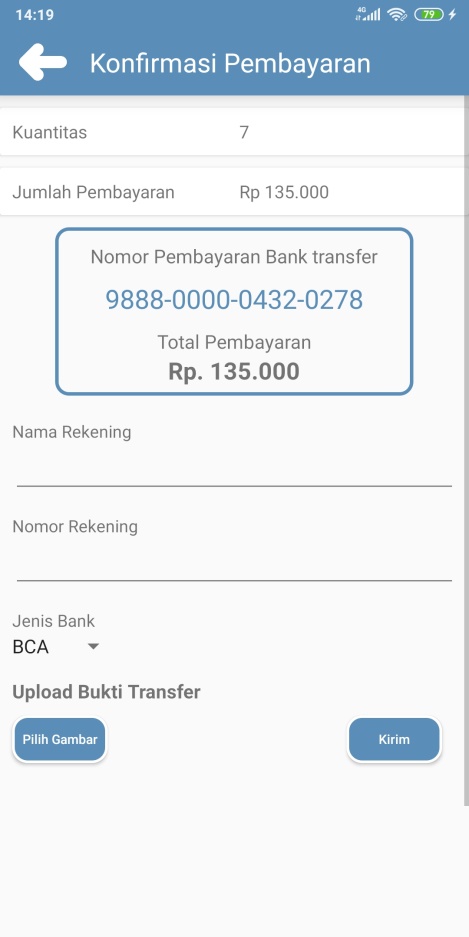
1. **Tampilan Menu Pendaftaran Event**

****

1. **Tampilan Pendaftaran Event (Free/Premium)**

** **

1. **Tampilan Konfirmasi Pembayaran Pendaftaran Event (Premium)**

****

**BAB V**

**PENUTUP**

1. **Kesimpulan**

EventKu merupakan aplikasi penyedia informasi event, workshop, seminar, gathering komunitas dan berbagai agenda lainnya dimana masyarakat bisa membeli tiket dan memberikan komentar terhadap event yang akan diselenggarakan. Tidak hanya mempromosikan event-event besar, tetapi juga event-event yang biasanya diselenggarakan oleh Unit Kegiatan Mahasiswa, komunitas-komunitas, maupun instansi-instansi terkait yang masih dalam ruang lingkup Kota Pontianak.

Tujuan dari aplikasi platform EventKu berbasis android ini dikembangkan adalah Memudahkan masyarakat khususnya warga Pontianak dalam menemukan informasi mengenai acara atau agenda yang akan diselenggarakan di Pontianak dan membuat platform bagi penyelenggara event untuk dapat mempromosikan kegiatannya dan mendapatkan banyak audiens.

EventKu juga menyediakan tiket event secara online yang dapat membantu para event organizer untuk mempromosikan eventnya. EventKu dibangun dengan menggunakan Android Studio dan bahasa pemrograman Java.

1. **Saran**

Sistem ini dibangun dinamis, untuk itu peneliti selanjutnya diharapkan dapat mengembangkan sistem informasi dengan menambahkan fitur-fitur lain.

**DAFTAR PUSTAKA**

Nesabamedia.com, Pengertian Java, Website: <https://www.nesabamedia.com/pengertian-java/>, diakses tanggal : 19 Desember 2019

Duniailkom.com, 2019, Tutorial Belajar Java : Pengertian Bahasa Pemrograman Java, Website: <https://www.duniailkom.com/tutorial-belajar-java-pengertian-bahasa-pemrograman-java/>, diakses tanggal : 19 Desember 2019

Medium.com, 2018, Pengertian Android Studio dan Macam Bagiannya, Website: <https://medium.com/@tulisanndeso/pengertian-android-studio-dan-berbagai-macam-bagiannya-a21db3160a6e>, diakses tanggal : 19 Desember 2019

Badoystudio.com, Android Studio, Website: <https://badoystudio.com/android-studio/#more-5409>, diakses tanggal : 19 Desember 2019

IDCLOUDHOST.com, Mengenal Apa Itu Android Studio, Fungsi, Manfaat dan Cara Installasinta, Website: <https://idcloudhost.com/mengenal-apa-itu-android-studio-fungsi-manfaat-dan-cara-installasinya/>, diakses tanggal : 19 Desember 2019