Tugas Kelompok – Progress Project Pertama

Nama Kelompok :

1. Teguh Agung Prabowo (512121230007)

2. Much Etrik Wijanarko (20190801178)

3. Candra (20190801153)

4. Dhea Devita Oktaryana (20190801135)

5. Yayan Afriyanto (30818059)

1. Tentukan Scope Dari Project :
2. Uraikan Konsep Overview Project Apa Yang Akan dihasilkan !

WeOrganize merupakan aplikasi yang digunakan sebagai sarana untuk mengikuti *event* dan menyelenggarakan *event* seperti seminar *online* maupun *offline*. WeOrganize bersifat *general* dimana semua *user* dapat menyelenggarakan maupun mengikuti *event* yang ada. *Event-*event ini dapat diselenggarakan tidak hanya dari satu universitas ataupun suatu lembaga saja tapi dapat diselenggarakan oleh perusahaan juga. Semua *user* dapat mengikuti *event-event* yang diselenggarakan dalam WeOrganize.

WeOrganize juga membantu penyelenggaraan pada *event* seperti seminar karena dapat menyelesaikan permasalahan seputar pelaksanaan *event* yang masih banyak menggunakan sistem manual sehingga dalam penelitian ini dapat memberikan solusi terhadap masalah yang terjadi pada penyelenggara dan peserta agar dapat memberikan informasi seputar pengumuman *event* dan pendaftaran peserta secara *online*, dapat menghindari absensi fiktif, karena absen kehadiran peserta menggunakan QRCode yang dikirim ke *email* masing-masing, serta peserta dapat bertanya langsung melalui aplikasi tanpa menyela isi materi yang sedang disampaikan oleh pembicara.

1. Tentukan Fuctional Requirement Buat Dalam Bentuk Tabel !

Kebutuhan fungsional adalah suatu kebutuhan berupa fungsi atau fitur – fitur yang harus ada dalam sistem yang akan dibuat dan berkaitan langsung dengan studi kasus yang dipilih. Daftar kebutuhan fungsional sistem akan ditampilkan pada tabel berikut ini.

|  |  |  |
| --- | --- | --- |
| No | Kebutuhan Fungsional | User |
| 1 | Sistem dapat menampilkan verifikasi *login* untuk EO | EO |
| 2 | Sistem dapat menampilkan *form* konfirmasi data bagi EO | EO |
| 3 | Sistem dapat menampilkan data *Participant* yang mengikuti *event* | EO |
| 4 | Sistem dapat menampilkan data *profile EO* | EO |
| 5 | Sistem dapat menampilkan halaman beranda | EO |
| 6 | Sistem dapat menampilkan menu membuat *event* | EO |
| 7 | Sistem dapat menampilkan menu *edit event* | EO |
| 8 | Sistem dapat menampilkan verifikasi *login* untuk *Participant* | Participant |
| 9 | Sistem dapat menampilkan *form* konfirmasi data bagi *Participant* | Participant |
| 10 | Sistem dapat menampilkan data *history event* | Participant |
| 11 | Sistem dapat menampilkan data *profile Participant* | Participant |
| 12 | Sistem dapat menampilkan halaman beranda | Participant |
| 13 | Sistem dapat menampilkan menu *scan qrcode* untuk *check-in event* | Participant |
| 14 | Sistem dapat menampilkan menu daftar *event* | Participant |
| 15 | Sistem dapat menampilkan notifikasi setelah berhasil *check-in* | Participant |

1. Tentukan Non Fungsional Requirement Buat Dalam Bentuk Diagram !

Kebutuhan non fungsional merupakan kebutuhan yang menitikberatkan pada properti yang dimiliki oleh sistem atau yang diperlukan di luar kebutuhan fungsional. Adapun kebutuhan non fungsional yang dibutuhkan adalah :

1. Karakteristik Sistem

|  |  |  |
| --- | --- | --- |
| No | Karakteristik Sistem | Keterangan |
| 1 | *Portability* | Aplikasi dapat diakses dimana saja dan kapan saja asalkan terhubung dengan internet. |
| 2 | *User Friendly* | Penggunaan fitur serta navigasi aplikasi yang mudah. |
| 3 | *Ergonomy* | Menjamin kenyamanan pemakaian antarmuka sistem dengan pengguna. |
| 4 | *Responsive* | *Responsive* merupakan suatu teknik agar tampilan aplikasi dapat menyesuaikan dengan layar *device* yang mengaksesnya. |

1. Kebutuhan Perangkat Lunak (*Software)* dan Perangkat Keras (*Hardware)*

|  |  |
| --- | --- |
| Kebutuhan Perangkat Lunak | Kebutuhan Perangkat Keras |
| 1. *Java Development Kit* versi 8   Digunakan untuk mengkompilasi kode program, dan *java* *development kit* ini kompatibel dengan spesifikasi laptop yang digunakan. | 1. Processor Intel® Core™ i3-3217U CPU @ 1.80GHz (4 CPUs), ~1.8GHz |
| 1. *Java Runtime*   Merupakan *platform* untuk menjalankan sistem, ini digunakan karena kompatibel dengan *Eclipse* *Juno*. | 1. RAM 2GB DDR3 |
| 1. Android Studio   Digunakan untuk membangun aplikasi *client* pada android. Aplikasi ini digunkan karena mudah diinstal dan digunakan dalam pembuatan *java script* dan juga *free*. | 1. HDD 500GB |
| 1. Android SDK,   Untuk pengembangan aplikasi android. | 1. Smartphone Android |
| 1. *Android Development Tools, plugin android* pada *android studio.* |  |
| 1. *Android Virtual Device,*   *Emulator* untuk menjalankan sistem android.  *Android SDK, Android Development Tools, dan Android Virtual Device* menggunakan tools bawaan dari *Eclipse Juno* yang dapat digunakan dengan mudah. |  |
| 1. Minimal Sistem Operasi Windows 7   Sistem Operasi Windows digunakan untuk proses *development* aplikasi ini. |  |
| 1. Android OS   Sistem Operasi Android ini dipilih karena sistem operasi ini bersifat *open source*, sehingga dapat dikembangkan siapa saja yang nantinya mau mengembangkannya. |  |

1. Buatlah Tabel Pengerjaan Project, dan Jobdesk Dari Masing Masing Anggota !

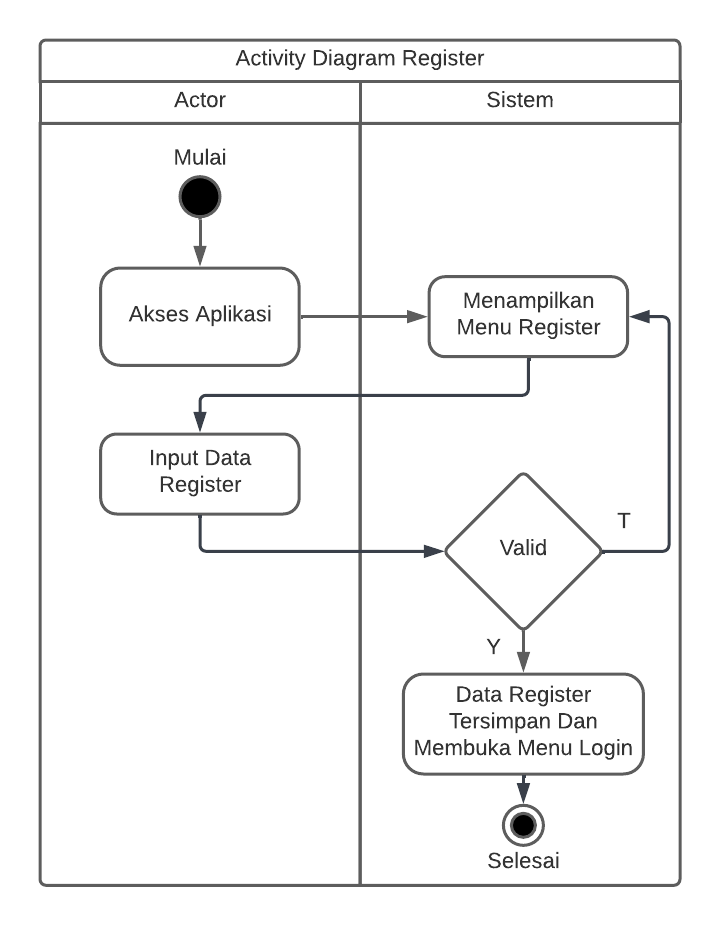
|  |  |  |
| --- | --- | --- |
| No | Nama Anggota | Jobdesk |
| 1 | Teguh Agung Prabowo dan Yayan Afriyanto | 1. Membuat *Activity Diagram* (Sistem / Proses Bisnis) 2. Membuat *Class Diagram* (Untuk Perancangan Database) 3. Membuat *Use Case Diagram* Untuk Model Sistem 4. Membuat *Deployment Diagram* Untuk Skema Arsitektur Aplikasi |
| 2 | Dhea Devita Oktaryana dan Much Etrik Wijanarko | 1. Membuat Uraian Konsep *Overview Project* Apa Yang Akan dihasilkan 2. Menentukan *Fuctional Requirement* Buat Dalam Bentuk Tabel 3. Menentukan *Non Fungsional Requirement* Buat Dalam Bentuk Diagram 4. Membuat Tabel Pengerjaan Project, dan *Jobdesk* Dari Masing Masing Anggota |
| 3 | Candra dan Much Etrik Wijanarko | 1. Membuat Desain *User Interface* Aplikasi *Mobile* dari studi kasus yang telah ditentukan, yang dibuat meliputi MENU (*FORM, BUTTON, TOGGLE, HEADER* Dan Lain Sebagainya). |

1. Konsep dan Alur Aplikasi :
2. *Activity Diagram* (Sistem / Proses Bisnis)

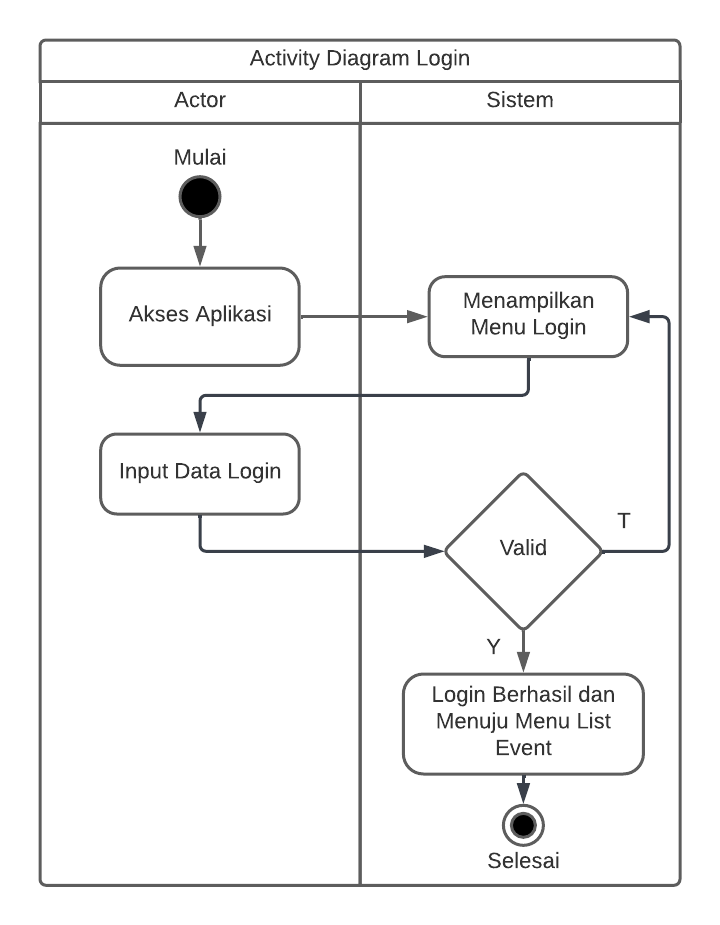
*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case.* Berikut ini *activity diagram* dari Perancangan

Aplikasi WeOrganize :

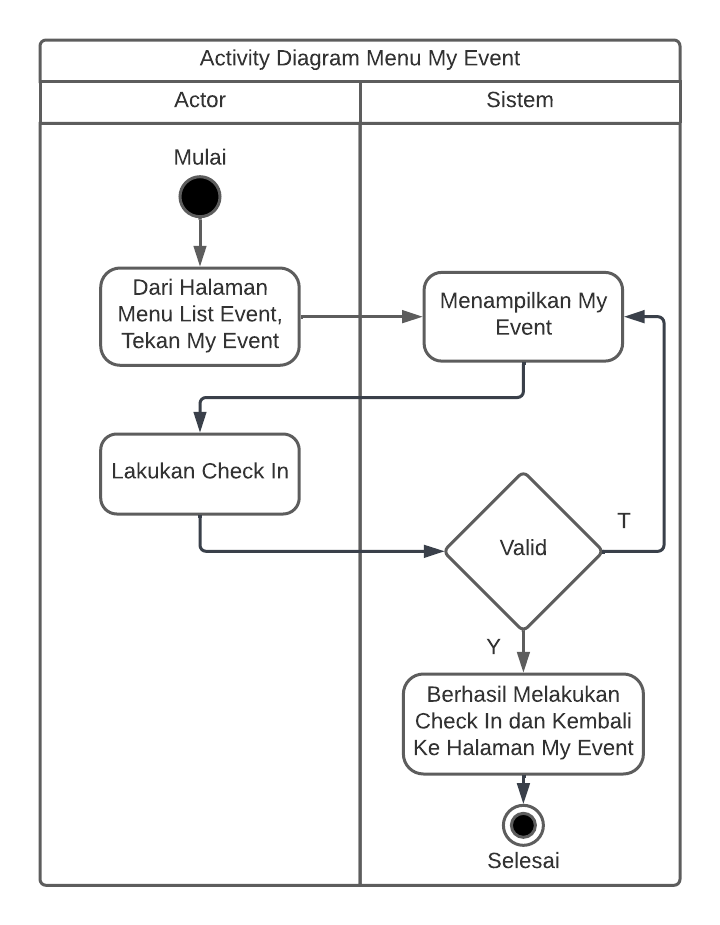
1. Register



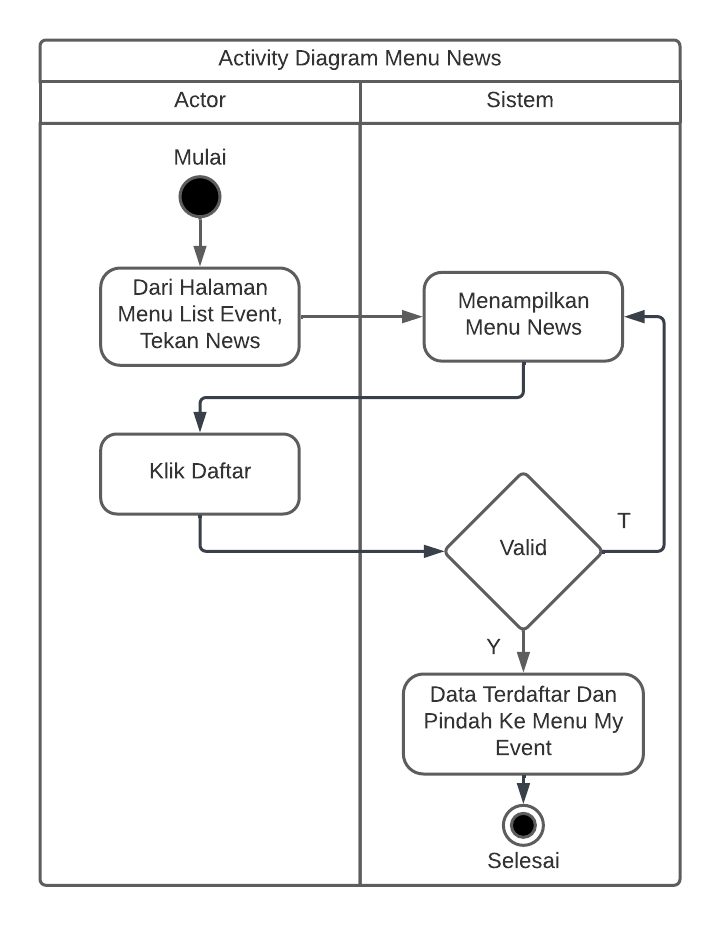
1. Login



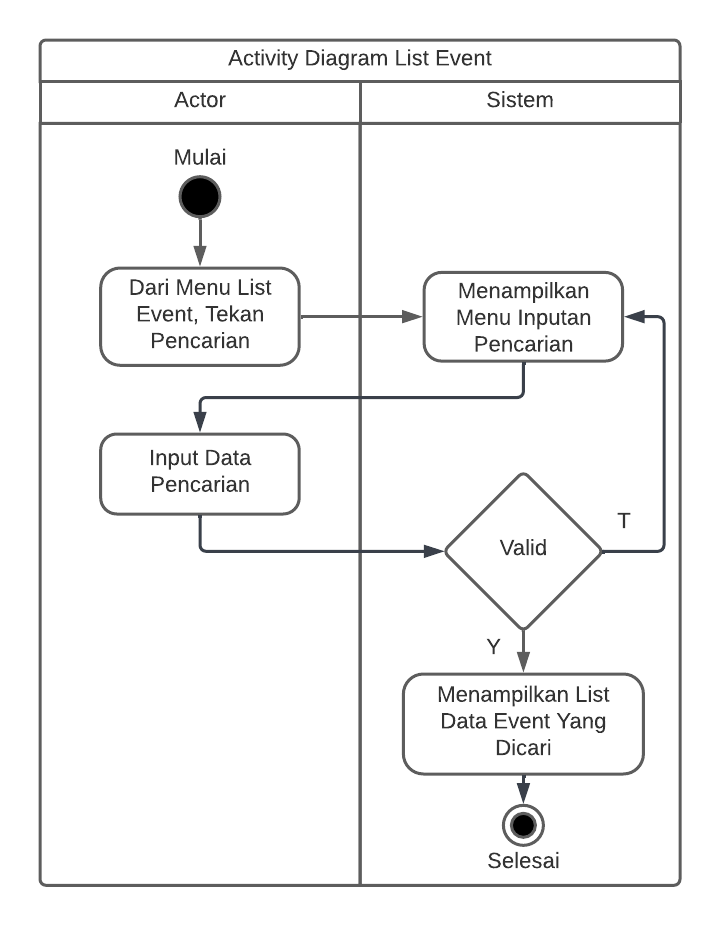
1. My Event



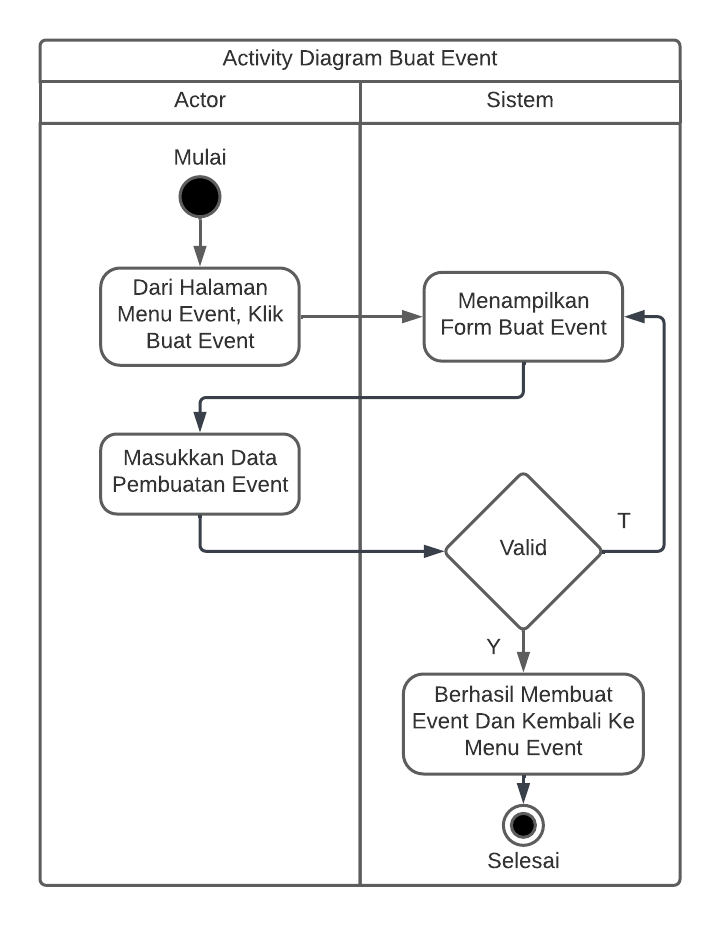
1. News



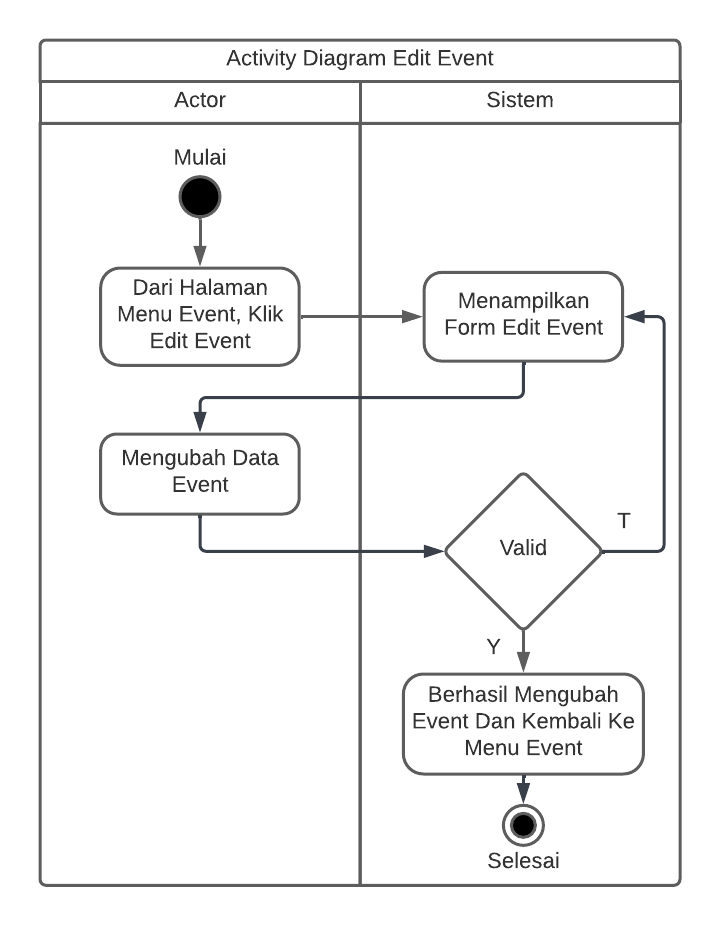
1. Event List



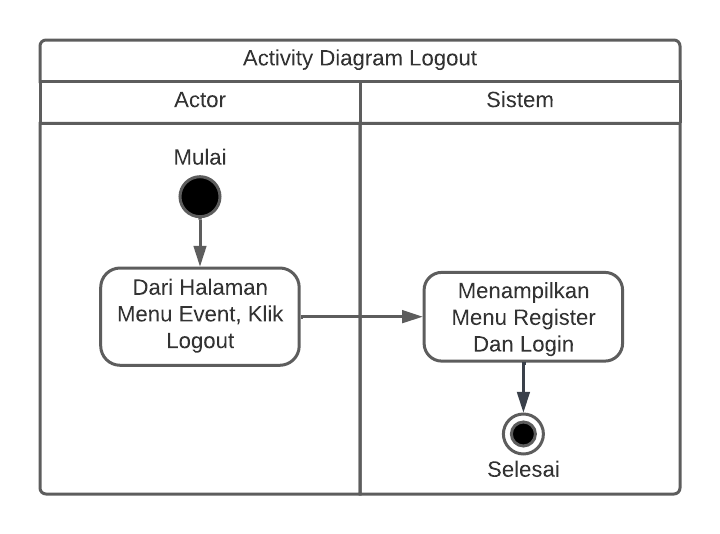
1. Create Event



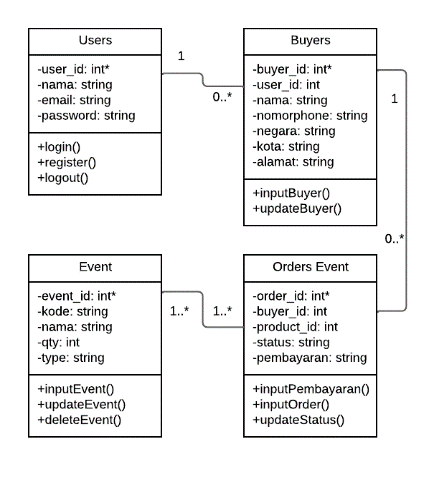
1. Edit Event



1. Logout

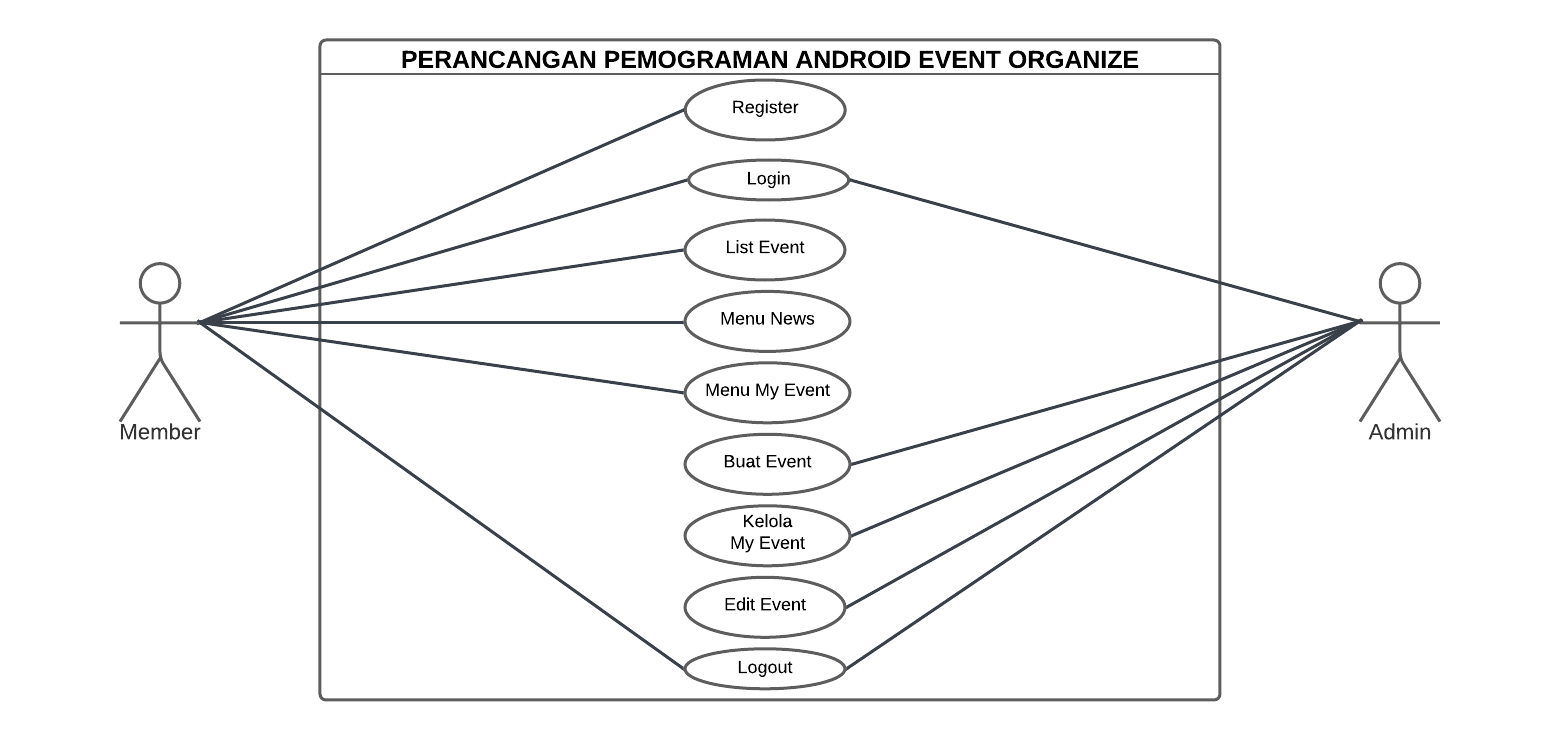


1. *Class Diagram* (Untuk Perancangan Database)  
   *Class Diagram* adalah model statis yang menggambarkan struktur dan deskripsi *class* serta hubungannya antara *class*. *Class Diagram* mirip *ER-Diagram* pada perancangan database, bedanya pada *ER-Diagram* tidak terdapat operasi/metode tapi hanya atribut. *Class* terdiri dari nama kelas, atribut dan operasi/metode. Berikut ini *Class Diagram* dari Perancangan Aplikasi WeOrganize :

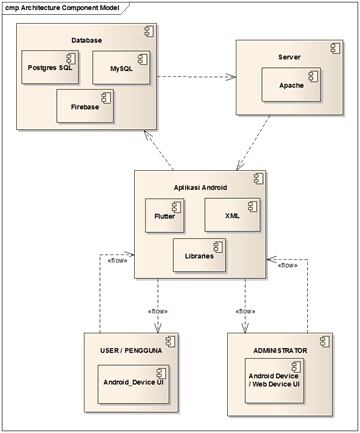


1. *Use Case Diagram* Untuk Model Sistem

*Use Case* *Diagram* merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use Case Diagram* bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat*. Use Case Diagram* juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem.

Berikut ini *Use Case Diagram* dari Perancangan Aplikasi WeOrganize :  


1. *Deployment Diagram* Untuk Skema Arsitektur Aplikasi

*Deployment Diagram* digunakan untuk memvisualisasikan hubungan antara *software* dan *hardware*. Secara spesifik *Deployment Diagram* dapat membuat *physical model* tentang bagaimana komponen perangkat lunak digunakan pada komponen perangkat keras, yang dikenal sebagai node. Berikut ini *Deployment Diagram* dari Perancangan Aplikasi WeOrganize :  
  


1. Desain User Interface Aplikasi We Organize

Pada kali ini kita mengerjakan desain UI/UX langsung pada aplikasi androidnya, desain langsung dituangkan pada aplikasi jadi, mengingat efisiensi waktu sangat penting,

Berikut link aplikasi jadi (file apk) serta penjelasan tiap interface yang telah kita buat :

1. Link

<https://drive.google.com/drive/folders/1hCQUulc8wTgR_-euxzReONwH76oT0QIp?usp=sharing>

1. Penjelasan :
   * Splash Screen

Adalah halaman atau layar yang muncul saat setelah aplikasi dijalankan, berikut tampilanya :

Text

Description automatically generated with low confidence

* + Login/Register

Menggunakan Google sebagai autentikasinya memudahkan kita memanage akun dari banyak pengguna serta akun google adalah hal wajib bagi pengguna android, berikut tampilanya :

A picture containing graphical user interface

Description automatically generated

* + Setup

Berisi halaman dimana pengguna bisa memutuskan *role* apa yang dia kehendaki, baik itu sebagai penyelenggara atau peserta, berikut tampilanya :

Text

Description automatically generated with medium confidence

* + Konfirmasi data

Adalah halaman dimana pengguna bisa menambahkan informasi tambahan atau sekedar mengkonfirmasi dara yang akan terisi dari akun googlenya, jika tersedia, berikut tampilanya :

A picture containing graphical user interface

Description automatically generated A picture containing diagram

Description automatically generated

* + Halaman Utama

Merupakan tampilan setelah berhasil login/membuat akun, berisi infromasi berupa acara yang akan datang yang dikemas dengan *carousel widget* dan lainnya,

Graphical user interface, text, application

Description automatically generated

Untuk sementara itulah *progress* dari aplikasi yang kelompok kami kerjakan, dimana akan ditambahkan dan dirampungkan fungsi dan fitur dari aplikasi secara utuh, dan juga mungkin akan ditambahkan fitur lainnya.