LAPORAN TUGAS PROYEK AKHIR

PERANCANGAN ”WEORGANIZE” PORTAL EVENT APP

Logo, company name

Description automatically generated

Disusun oleh :

|  |  |
| --- | --- |
| **Nama** | **NIM** |
| Candra | 20190801153 |
| Dhea Devita Oktaryana | 20190801135 |
| Much Etrik Wijanarko | 20190801178 |
| Teguh Agung Prabowo | 512121230007 |
| Yayan Afriyanto | 30818059 |

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS ESA UNGGUL KEBON JERUK**

**TAHUN 2022**

# DAFTAR ISI

[DAFTAR ISI 3](#_Toc108554504)

[1 PENDAHULUAN 5](#_Toc108554505)

[2 LANDASAN TEORI 5](#_Toc108554506)

[3 RUANG LINGKUP PROYEK SISTEM INFORMASI 8](#_Toc108554507)

[3.1 Definisi Masalah dan Ruang Lingkup 8](#_Toc108554508)

[3.2 Kebutuhan Sistem Fungsional 8](#_Toc108554509)

[3.3 Kebutuhan Sistem Non Fungsional 8](#_Toc108554510)

[4 METODOLOGI PENGEMBANGAN SISTEM INFORMASI 10](#_Toc108554511)

[5 PERANCANGAN APLIKASI 11](#_Toc108554512)

[*5.1* *Use Case Diagram* 11](#_Toc108554513)

[*5.2* *Activity Diagram* (Sistem / Proses Bisnis) 11](#_Toc108554514)

[*5.3* *Class Diagram* (Untuk Perancangan Database) 14](#_Toc108554515)

[*5.4* *ERD (Entity Relationship Diagram)* 14](#_Toc108554516)

[*5.5* Kamus Data 15](#_Toc108554517)

[6. MODEL ARSITEKTUR APLIKASI 17](#_Toc108554518)

[6.1 *Deployment Diagram* Untuk Skema Arsitektur Aplikasi 17](#_Toc108554519)

[7. UI / UX APLIKASI 18](#_Toc108554520)

[8. LINK 21](#_Toc108554521)

[8.1. Source Code 21](#_Toc108554522)

[8.2. Link Apk 21](#_Toc108554523)

[8.3. Demo Youtube 21](#_Toc108554524)

[8.4. Demo Instagram 21](#_Toc108554525)

[9. DAFTAR PUSTAKA 21](#_Toc108554526)

# PENDAHULUAN

Dalam menyelenggarakan kegiatan-kegiatan yang bersifat kajian ilmiah, manajerial dan operasional pada umumnya tidak terlepas dari penyelenggaraan kegiatan seminar dan workshop. Penyelenggaran kegiatan tersebut tentunya untuk membantu kemampuan sumber daya manusia dalam meningkatkan pengetahuan dan keterampilan, kegiatan ini biasanya diadakan oleh kampus-kampus dan perkantoran untuk mengadakan kegiatan kajian ilmiah, pelatihan manajerial dan operasional, kegiatannya dapat dilaksanakan satu bulan sekali atau sesuai kesepakan yang ditentukan.

Biasanya seminar dipromosikan melalui poster dan pamflet yang ditempelkan di tempat pengumuman kampus ataupun kantor. Namun di masa pandemi ini, kebanyakan mahasiswa maupun karyawan kesulitan untuk menemukan wadah yang membagikan kegiatan bermanfaat seperti *event* dan seminar yang dapat menambah wawasan. Maka dari itu, dirancanglah suatu aplikasi yang dapat mengatasi masalah tersebut, yakni WeOrganize.

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.

# LANDASAN TEORI

* 1. Seminar

Seminar merupakan suatu pertemuan yang diikuti oleh sekelompok orang untuk membahas suatu topik tertentu dan mencari solusi mengenai suatu tema atau masalah dengan menghadirkan seorang mentor yang ahli dibidang tersebut untuk menjelaskan solusi masalah yang dibahas.

Dari segi etimologi kata, istilah seminar berasal dari bahasa Latin, yaitu seminarium yang artinya adalah tanah tempat menanan benih. Kalimat tersebut tidak untuk dipahami secara harafiah, melainkan sebuat bentuk kalimat konotasi, dimana artinya lebih kepada bentuk pengajaran akademis maupun pembentukan diri yang dilaksanakan oleh suatu komunitas maupun organisasi pendidikan.

Tujuan utama dilangsungkannya kegiatan seminar adalah agar informasi maupun buah pikiran yang terbaru dapat tersampaikan kepada orang banyak. Sehingga informasi tersebut bisa dengan lebih mudah lagi dikembangkan kepada masyarakat yang lebih luas.

* 1. UML (*Unified Modelling Language*)

Merupakan suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997.

UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint*sebuah *software.*

UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability, robustness, security*, dan sebagainya.

Tools yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*, merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol simbol yang terdapat pada use case diagram yaitu :
   1. Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan antar unit dengan aktor dan dinyatakan dengan kata kerja.
   2. Actor adalah abstraksi dari orang atau sistem lain yang mengaktifkan fungsi dari target sistem.
   3. Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang meminta interaksi secara langsung.
2. Diagram Aktivitas (*Activity Diagram*), Menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam Activity Diagram yaitu :
   1. Start Point, yang merupakan awal aktivitas,
   2. End Point, akhir aktivitas,
   3. Activities, menggambarkan proses atau kegiatan bisnis
   4. Percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau menggabungkan dua kegiatan menjadi satu yaitu penggabungan.
   5. Decision Points, menggambarkan pilihan untuk pengambilan keputusan.
   6. Swimlane, pembagian activity diagram untuk menunjukkan stakeholder yang melakukan aktivitas.
3. *Class Diagram* (Diagram Kelas),

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas yang juga memperlihatkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Class diagram juga menunjukkan atribut dan operasi dari sebuah kelas dan batasan yang berhubungan dengan objek yang dikoneksikan. Class diagram meliputi : kelas, relasi, atribut, operasi dan visibility. Hubungan antar kelas mempunyai keterangan disebut dengan Kardinalitas.

1. *Deployment Diagram,*

Deployment diagram merupakan salah satu model diagram dalam UML untuk mengerahkan artifact dalam node. *Deployment diagram* digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis *Object Oriented* yang akan dibangun. Tujuan atau fungsi dari *deployment diagram* yaitu untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/*software*.

Node dalam UML merepresentasikan *hardware* atau *software execution environments*. Node bisa terhubung melalui *communication path* untuk membuat sistem jaringan dari *arbitrary complexity*. Artefak dalam UML mempresentasikan spesifikasi dari bentuk *physic* informasi yang digunakan atau dihasilkan development *process*.

* 1. Android

Android merupakan sebuah sistem operasi seluler yang didasarkan pada versi modifikasi dari kernel Linux dan perangkat sumber terbuka lainnya. Android dirancang untuk perangkat seluler terutama layar sentuh seperti smartphone dan tablet. Sistem operasi ini pertama kali diluncurkan pada bulan September 2008, di mana Android dikembangkan oleh Open Handset Alliance yang disponsori secara komersial oleh Google.

Android bersifat open source atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat open source perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi alias gratis.

Begitupun dengan para pembuat aplikasi, mereka bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan google. Dengan seperti itu android memiliki jutaan support aplikasi gratis/berbayar yang dapat diunduh melalui google play.

# RUANG LINGKUP PROYEK SISTEM INFORMASI

# Definisi Masalah dan Ruang Lingkup

Laporan ini dibatasi oleh beberapa ruang lingkup agar pembahasan terhadap masalah yang ada dapat tersampaikan dengan jelas. Batasan-batasan tersebut diantaranya adalah:

1. Mendaftarkan diri sebagai *Organizer* ataupun *Partisipant*
2. Menampilkan *event*
3. Menampilkan *history*
4. Menampilkan *profile*
5. Melakukan *scan QR Code*

# Kebutuhan Sistem Fungsional

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 |

# Kebutuhan Sistem Non Fungsional

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. |  |

# METODOLOGI PENGEMBANGAN SISTEM INFORMASI

Model proses pengembangan dari sistem informasi yang akan kami buat menggunakan Model Waterfall. Menurut Pressman (2012), Model Waterfall (model air terjun) merupakan suatu model pengembangan secara sekuensial. Model Waterfall bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan. Model pengembangan waterfall memiliki beberapa kelebihan, antara lain: dapat mudah dipahami dan dapat diterapkan dalam proses pengembangan perangkat lunak.

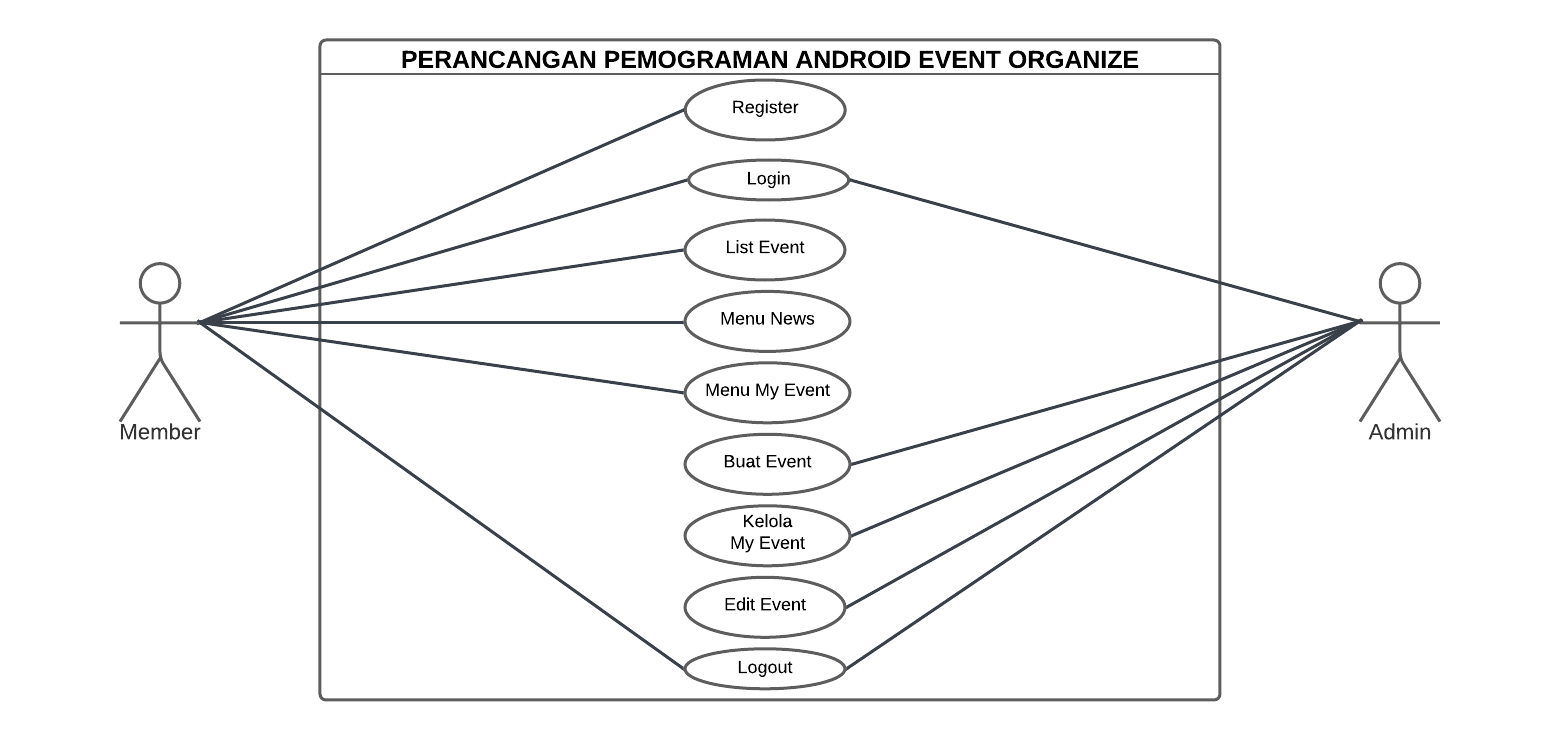
Tahapan yang harus dilakukan sesuai dengan model waterfall yaitu :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Communication | | | | | |
| Instrumen | | Langkah-langkah | | Luaran | |
| Observasi, wawancara | | Identifikasi masalah | | Informasi yang diperlukan, masalah yang terjadi, dan  solusi yang akan diberikan | |
| Observasi, wawancara | | Analisis kebutuhan pengguna | | Kebutuhan pengguna | |
| Observasi, wawancara | | Identifikasi data | | Data yang diperlukan dalam pembuatan aplikasi | |
|  | | | | | |
| Planning | | | | | |
| Instrumen | | Langkah-langkah | | Luaran | |
| - | | Pembuatan jadwal kerja  penelitian | | Jadwal kerja penelitian | |
|  | |  | |  | |
| Modelling | | | | | |
| Instrumen | | Langkah-langkah | | Luaran | |
| Draw.io, Astah UML | | Perancangan desain proses fungsional | | 1. Use case diagram 2. Activity diagram 3. Use case description 4. Class diagram 5. Deployment diagram | |
| Draw.io | | Perancangan antarmuka  pengguna | | Desain antarmuka pengguna | |
| Microsoft Word | | Perancangan uji coba | | Dokumen perancangan uji  coba |
| Construction | | | | |
| Instrumen | Langkah-langkah | | Luaran | |
| Android studio | Pembuatan program (coding) | | Aplikasi | |
| Android studio | Uji coba program | | Hasil uji coba | |

# PERANCANGAN APLIKASI

# *Use Case Diagram*

*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.

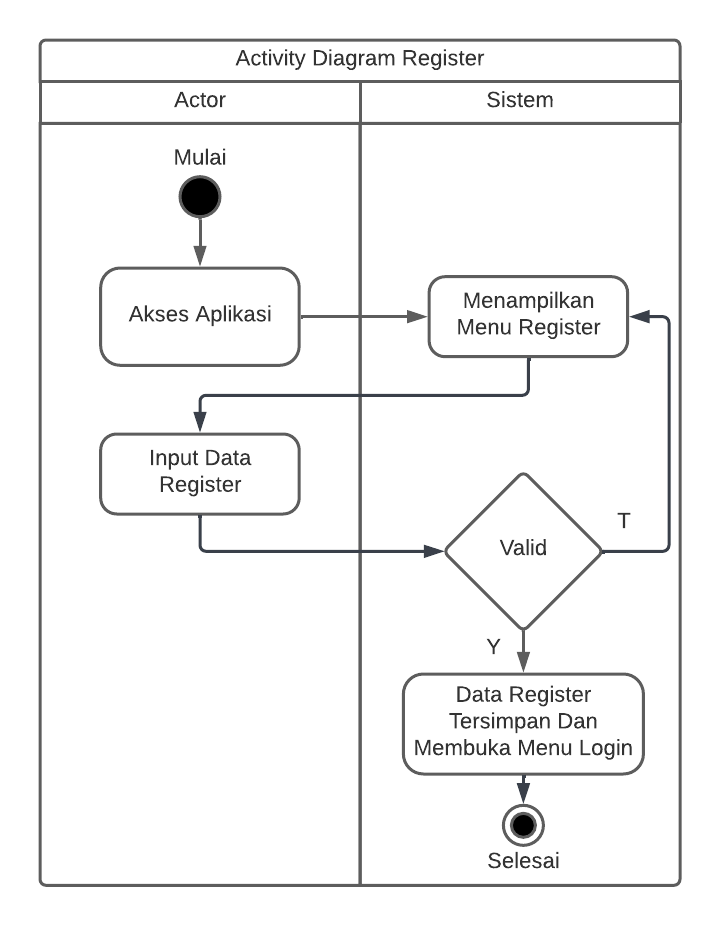
Berikut ini *Use Case Diagram* dari Perancangan Aplikasi WeOrganize :  


# *Activity Diagram* (Sistem / Proses Bisnis)

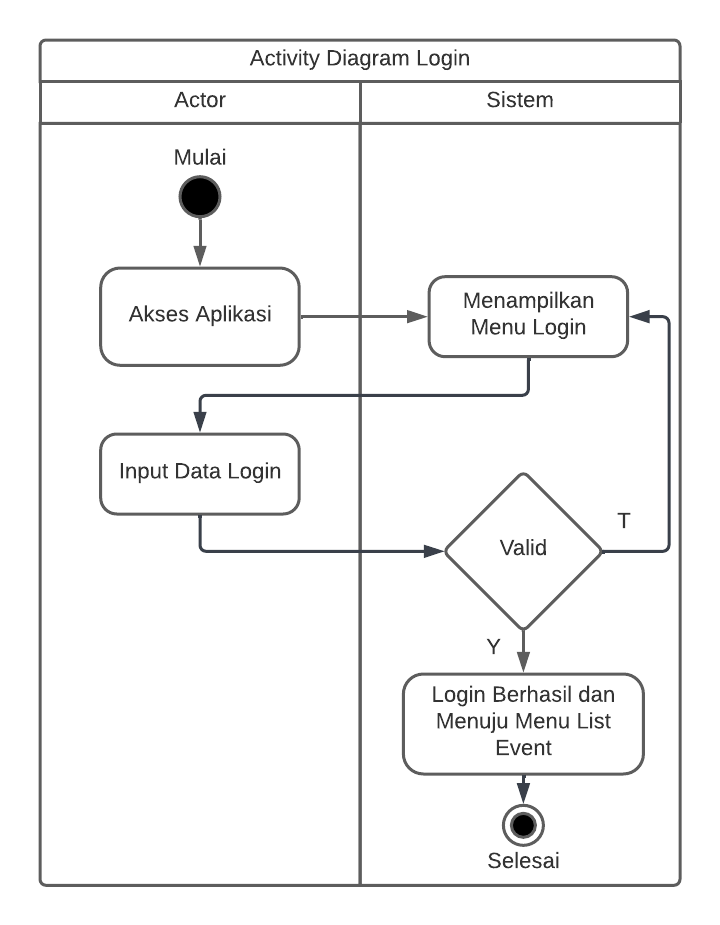
*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case.* Berikut ini *activity diagram* dari Perancangan

Aplikasi WeOrganize :

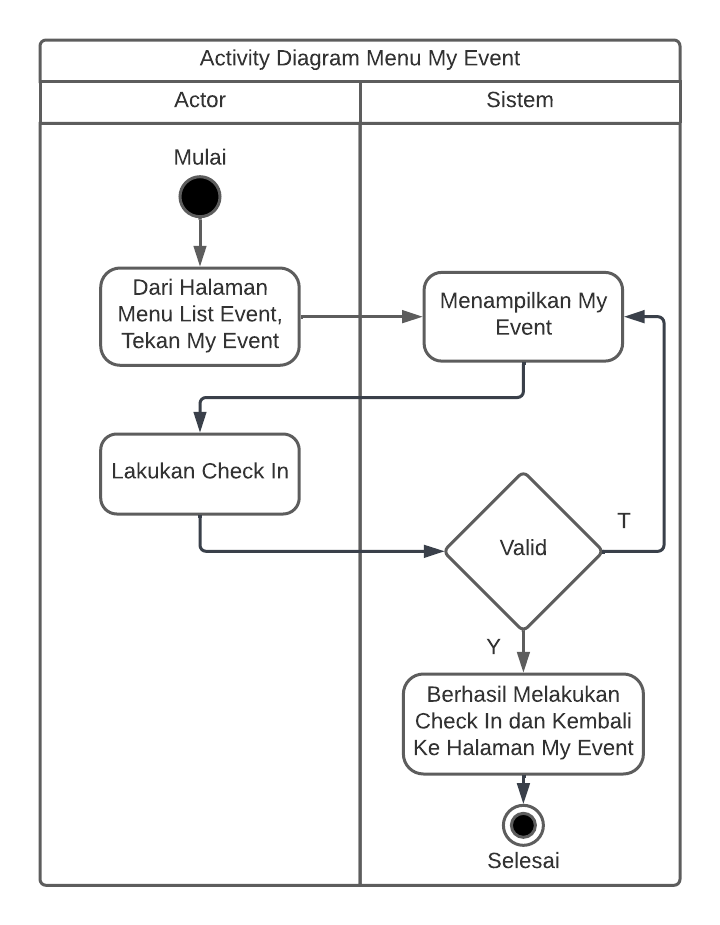
* 1. Register



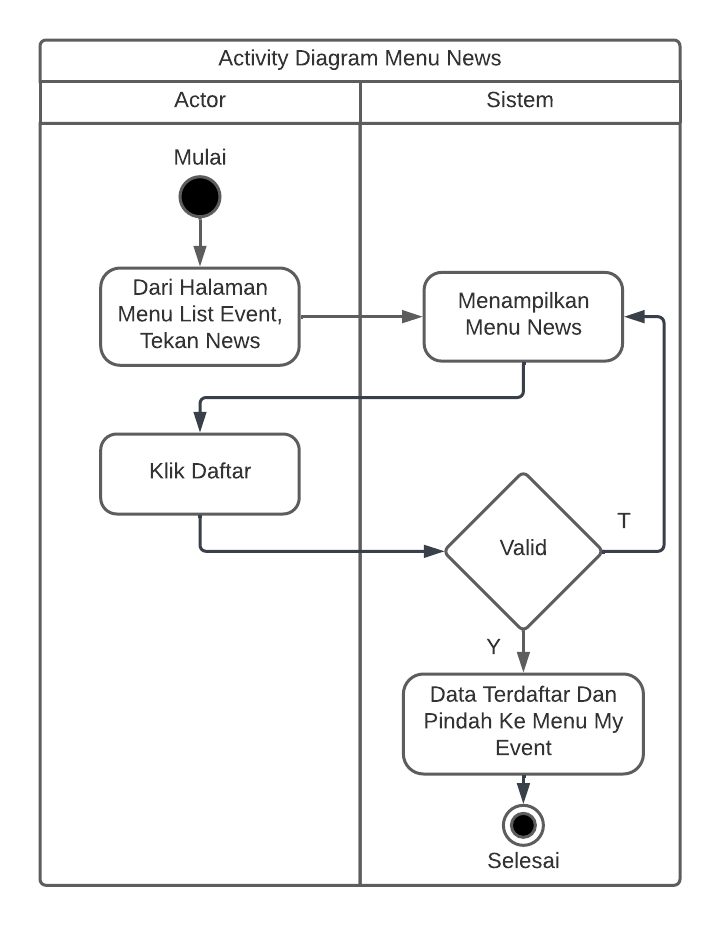
* 1. Login



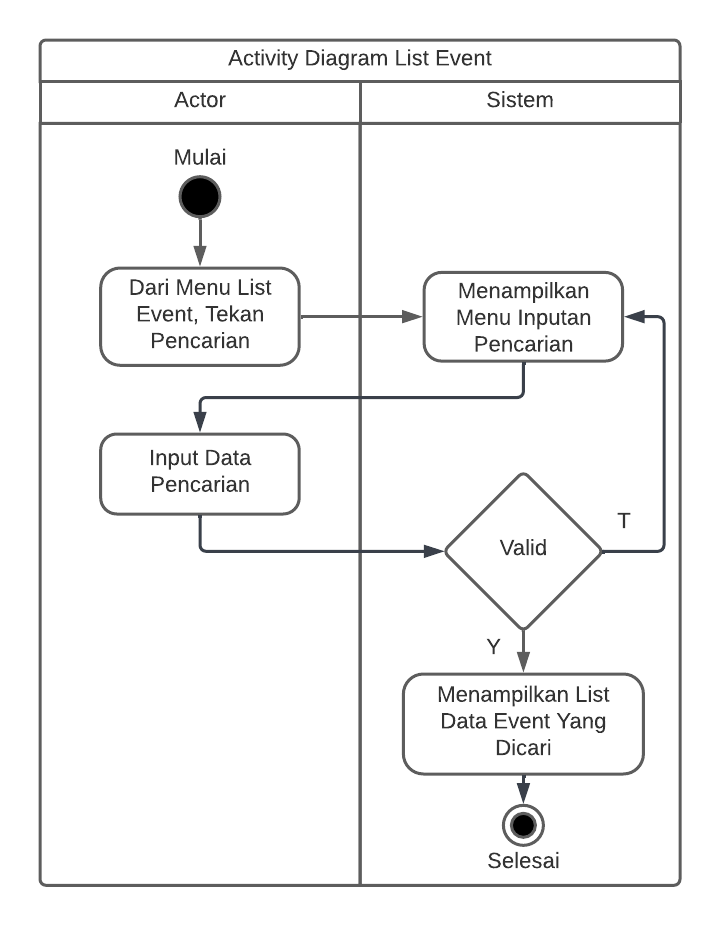
* 1. My Event



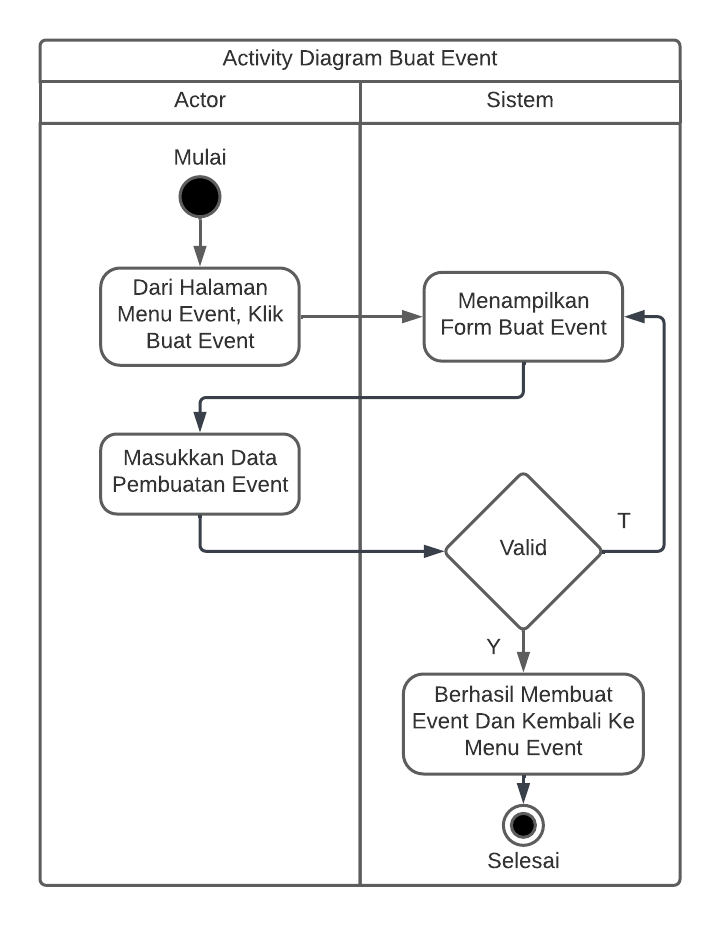
* 1. News



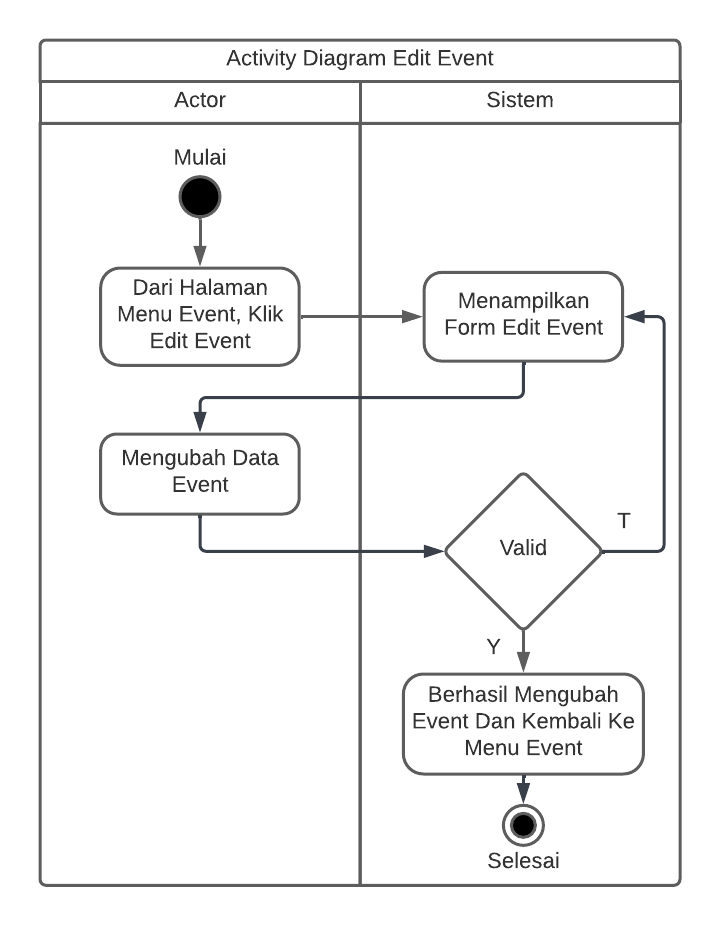
* 1. Event List



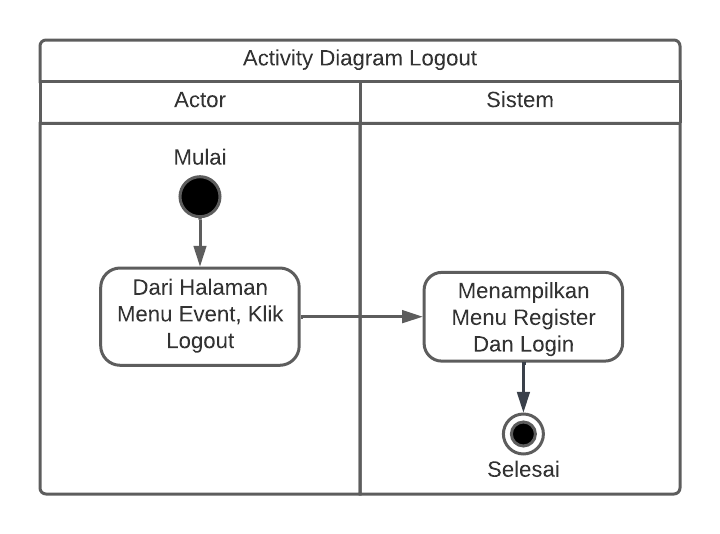
* 1. Create Event



* 1. Edit Event

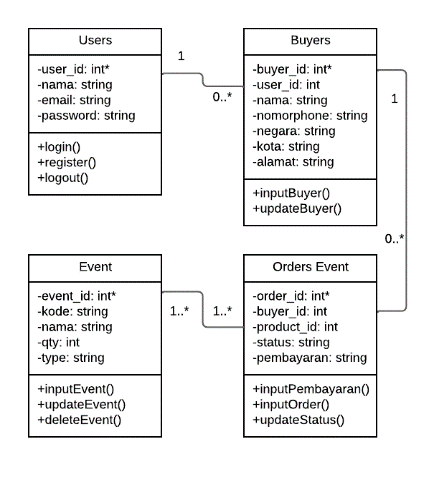


* 1. Logout



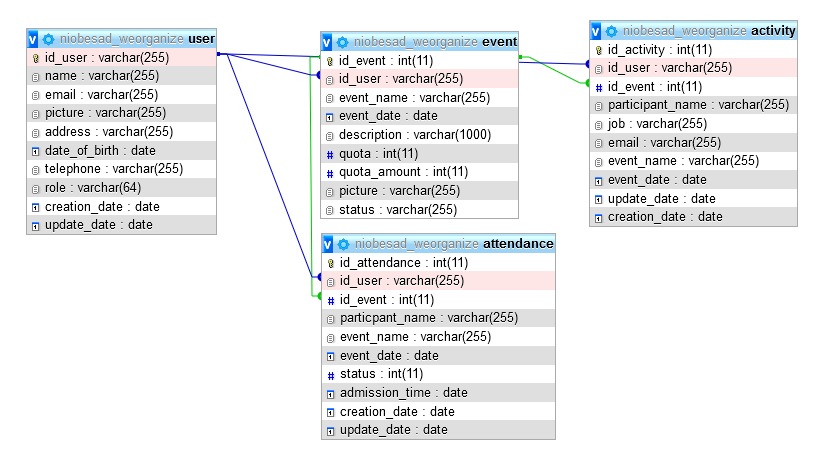
# *Class Diagram* (Untuk Perancangan Database)

*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 :



# *ERD (Entity Relationship Diagram)*

Berikut dibawah ini relasi antar tabel di database weorganize :



# Kamus Data

Berikut ini akan diuraikan struktur table-tabel dalam WeOrganize :

1. Nama Tabel : activity

Fungsi Tabel : Untuk menyimpan aktivitas dari setiap event yang diselenggarakan

|  |  |
| --- | --- |
| Nama | Type |
| id\_activity | int (11) |
| id\_user | int (11) |
| id\_event | int (11) |
| participant\_name | varchar(255) |
| job | varchar(255) |
| email | varchar(255) |
| event\_name | varchar(255) |
| event\_date | date |
| update\_date | date |
| creation\_date | date |

1. Nama Tabel : attendance

Fungsi Tabel : Untuk menyimpan daftar hadir dari peserta event

|  |  |
| --- | --- |
| Nama | Type |
| id\_attendance | int(11) |
| id\_user | int(11) |
| id\_event | int(11) |
| particpant\_name | varchar(255) |
| event\_name | varchar(255) |
| event\_date | Date |
| status | int(11) |
| admission\_time | Date |
| creation\_date | Date |
| update\_date | Date |

1. Nama Tabel : event

Fungsi Tabel : Untuk menyimpan event-event yang akan diselenggarakan

|  |  |
| --- | --- |
| Nama | Type |
| id\_event | int(11) |
| id\_user | int(11) |
| id\_event | int(11) |
| oclock | varchar(255) |
| event\_name | varchar(255) |
| event\_date | Date |
| description | int(11) |
| quota | int(11) |
| quota\_amount | int(11) |
| last\_quota | int(11) |
| picture | varchar(255) |
| status | int(11) |
| creation\_date | Date |
| update\_date | Date |

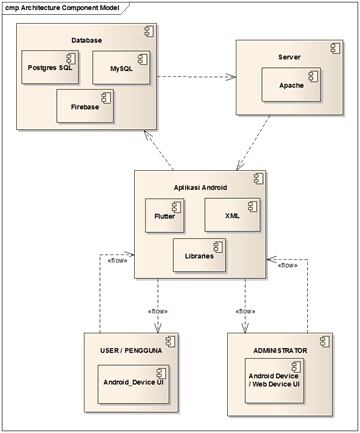
1. Nama Tabel : user

Fungsi Tabel : Untuk menyimpan data user dan organize

|  |  |
| --- | --- |
| Nama | Type |
| id\_user | int(11) |
| name | varchar(255) |
| email | varchar(255) |
| address | varchar(255) |
| date\_of\_birth | Date |
| telpone | int(11) |
| password | varchar(255) |
| job | varchar(255) |
| is\_event | int(11) |
| status | int(11) |
| creation\_date | Date |
| update\_date | Date |

# MODEL ARSITEKTUR APLIKASI

# *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 : 

# UI / UX APLIKASI

Implementasi UI dari aplikasi kami, dimana terdapat tampilan-tampilan berikut

1. Login

Tampilan login kami menggunakan *firebase authentication* sebagai *back-end*nya, dan dengan fitur *google sign-in* jadi tak perlu lagi mengingat *password* dan *email*.

A picture containing graphical user interface

Description automatically generated Graphical user interface, text, application

Description automatically generated

1. Memilih *role*

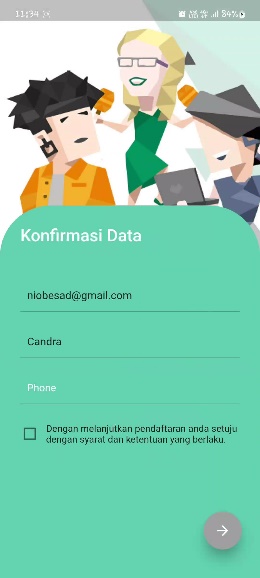
Aplikasi kita bisa digunakan tak hanya oleh *participant* saja, tetapi juga oleh para *event organizer,* dimana pada tampilan awal, kita hanya perlu memilih *role* setelah *sign in* pada aplikasi kami.

Text

Description automatically generated

1. Tampilan Konfirmasi Data

Setelah memilih *role,* maka akan ada tampilan untuk konfirmasi data. Sebagian data sudah terisi karena kita menarik data dari *back-end* dalam hal ini *firebase*, perbedaan antara 2 *role* ini adalah untuk *organizer* wajib menyatakan instansi terkait.

 A close up of a calculator

Description automatically generated with medium confidence

1. Tampilan Awal *Role*

Pada tampilan awal *role*, untuk *role organizer* sama dengan *role participant*. Yang membedakan hanya pada tombol *floating-button* untuk membuat *event* baru yang hanya ada pada *role organizer*. Pada tampilan ini, *participant* dapat memilih *event-event* yang menarik untuk mereka ikuti.

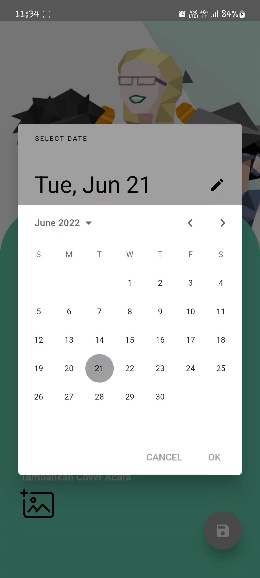
Timeline

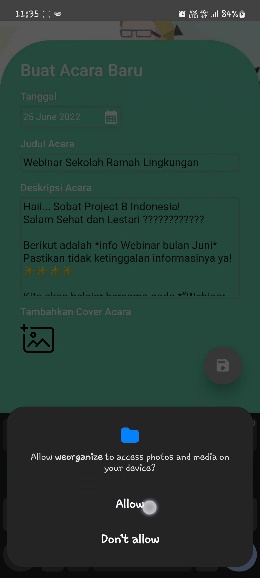
Description automatically generated Graphical user interface, text, application

Description automatically generated

1. Tampilan membuat *Event*

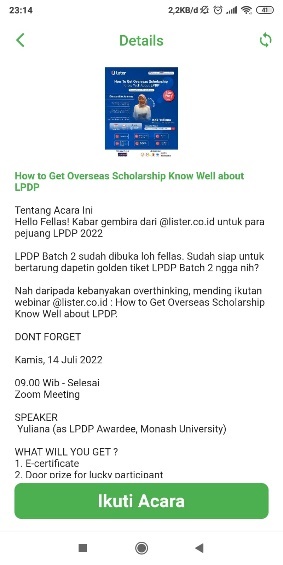
Pada tampilan ini *user* dengan *role organizer* bisa membuat *event* baru dengan bebas memilih tanggal, judul, deskripsi bahkan gambar, berikut tampilannya;

 Diagram

Description automatically generated 

1. Tampilan *Event*

Pada tampilan *event,* terdapat beberapa *event* yang akan berlangsung, dimana kita dapat meng-klik *event* tersebut untuk membaca detailnya, dan dapat juga mendaftar untuk mengikuti *event* tersebut.

1. Tampilan *History*

Pada tampilan *History,* terdapat *event-event* yang sudah pernah kita ikuti. Kita dapat melihat kembali detail dari *event-event* tersebut.



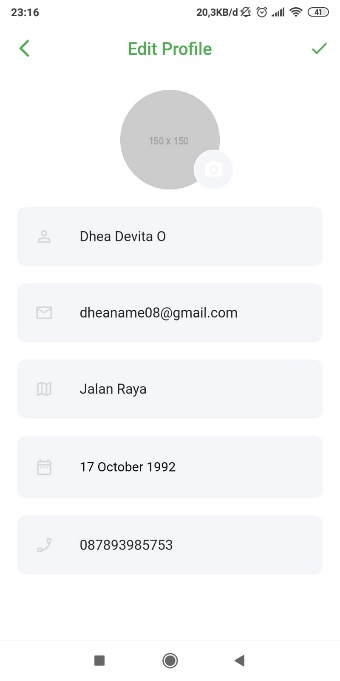
1. Tampilan Edit *Event*

Pada tampilan ini, *event organizer* dapat mengedit detail dari *event* yang akan mereka laksanakan, apabila ada detail yang kurang jelas, ataupun yang belum tercantum dari detail yang sebelumnya telah mereka *upload.*



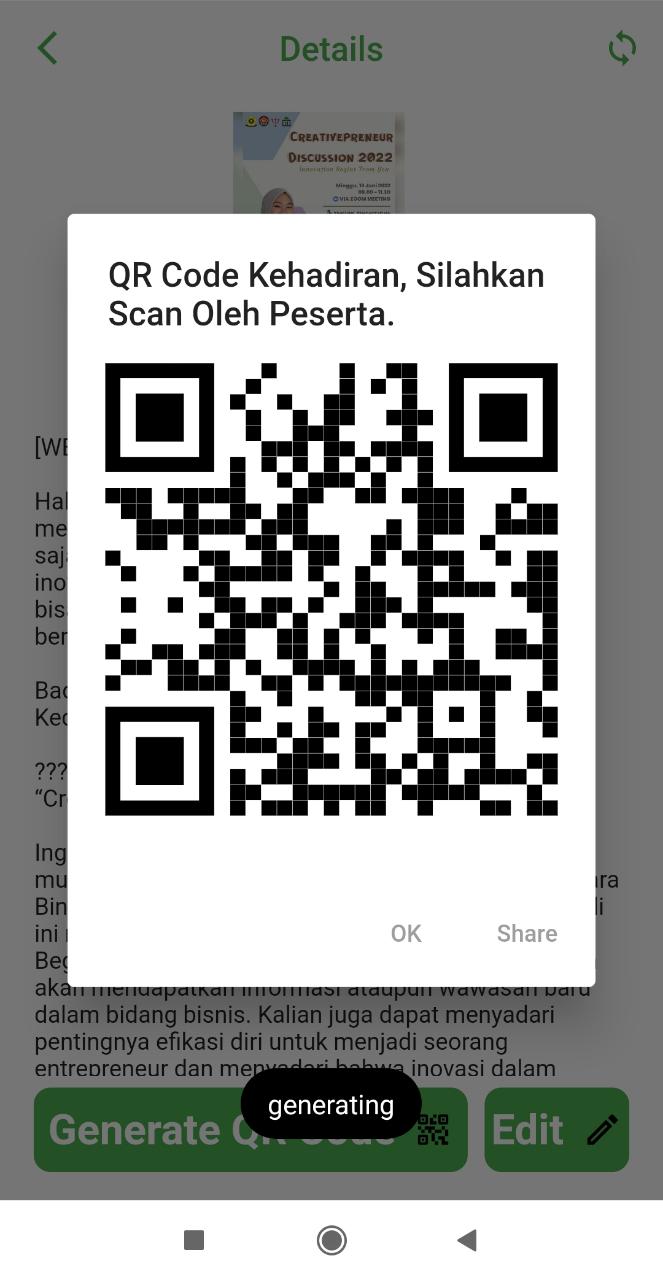
1. TampilanEdit *Profile*

Pada tampilan ini, *participant* ataupun *event organizer* dapat mengubah data yang telah mereka isi sebelumnya.



1. Tampilan QR Code

Pada tampilan ini, *organizer* atau pelaksana *event* men-*generate* *QR Code* untuk mendata kehadiran peserta atau *participant* dari *event* yang kemudian akan di*scan* oleh peserta atau *participant.*



# LINK

# Source Code

<https://github.com/sadandsleepy/weorganize>

# Link Apk

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

# Demo Youtube

<https://youtu.be/_ECdVvzAqw8>

# Demo Instagram

https://www.instagram.com/reel/Cf6oM2MA\_xk/?igshid=MDJmNzVkMjY=

# DAFTAR PUSTAKA

“Apa itu UML? Beserta Pengertian dan Contohnya”. dicoding.com. 12 Mei 2021. Diakses 11 Juli 2022. <https://www.dicoding.com/blog/apa-itu-uml/>

“UML Use Case Diagram Tutorial”. Lucidchart.com. Diakses 9 Juli 2022. <https://www.lucidchart.com/pages/uml-use-case-diagram>

“Contoh Collaboration Diagram”. Guratgarut.com. 12 Juli 2021. Diakses 9 Juli 2022. <https://guratgarut.com/contoh-collaboration-diagram/>

“Pengertian Seminar Adalah : Arti, Ciri-Ciri, Tujuan, Fungsi dan Pihak yang Terlibat Dalam Seminar”. Saturadar.com. 3 November 2019. Diakses 11 Juli 2022. <https://www.saturadar.com/2019/11/Pengertian-Seminar.html>

“PENGERTIAN ANDROID: Sejarah, Kelebihan & Versi Sistem Operasi”. salamadian.com. 23 Oktober 2019. Diakses 12 Mei 2022. <https://salamadian.com/pengertian-android/>