**PENGEMBANGAN APLIKASI MANAJEMEN SEMINAR *(DIGIMICE)* UNTUK PANITIA BERBASIS ANDROID**

**LAPORAN PROYEK AKHIR**



**Oleh :**

**ANGGA DWI SAPUTRA**

**NIM. 361755401181**

**PROGRAM STUDI DIPLOMA III**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANYUWANGI**

**2020**

*--- Halaman ini sengaja dikosongkan ---*

**PENGEMBANGAN APLIKASI MANAJEMENSEMINAR *(DIGIMICE)* UNTUK PANITIA BERBASIS ANDROID**

# HALAMAN JUDUL

**LAPORAN PROYEK AKHIR**



Proyek Akhir ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan Program Studi Diploma III Teknik Informatika dan Mencapai

Gelar Ahli Madya (A. Md)

**Oleh :**

**ANGGA DWI SAPUTRA**

**NIM. 361755401181**

**PROGRAM STUDI DIPLOMA III**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANYUWANGI**

**2020**

*--- Halaman ini sengaja dikosongkan ---*

# PERSEMBAHAN

Segala puji syukur kepada Allah SWT yang telah memberikan rahmat serta hidayah-Nya, sehingga Proyek Akhir ini dapat diselesaikan.

Proyek Akhir ini saya persembahkan kepada :

1. Ibunda “Ervayanti”, Ayahanda “Wageono” dan keluarga penulis tercinta atas doa dan dukungan yang diberikan kepada penulis selama proses penulisan proyek akhir.
2. Bapak Devit Suwardiyanto, S.Si., M.T. selaku Dosen Pembimbing I dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom. selaku Dosen Pembimbing II yang telah membimbing dengan baik selama proses penulisan proyek akhir selesai.
3. Bapak Herman Yuliandoko, S.T., M.T. selaku Dosen Penguji I dan Bapak Sepyan Purnama Kristanto, S.Kom., M.Kom. selaku Dosen Penguji II yang telah meluangkan waktunya untuk menguji dan membimbing dengan baik sampai penulisan proyek akhir selesai.
4. Bapak dan Ibu Dosen Teknik Informatika Politeknik Negeri Banyuwangi yang telah membantu membimbing penulis menuntut ilmu selama perkuliahan berlangsung yang berperan atas berjalannya proyek akhir ini.
5. Sahabat dan teman-teman penulis atas bantuan, motivasi, dan waktu untuk berdiskusi. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu yang selalu mendukung penulis.

Penulis menyadari bahwa proyek akhir ini masih terdapat banyak kekurangan. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun sehingga proyek akhir ini dapat bermanfaat.

*Halaman ini sengaja dikosongkan ---*

# MOTTO

“Teruslah berjuang jangan cepat menyerah dan

selalu sabar dengan segala cobaan yang diterima”

*--- Halaman ini sengaja dikosongkan ---*

# PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Angga Dwi Saputra

NIM : 361755401181

Menyatakan dengan sesungguhnya bahwa proyek akhir yang berjudul “Pengembangan Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android“ adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya, serta bukan karya jiplakan/plagiat. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

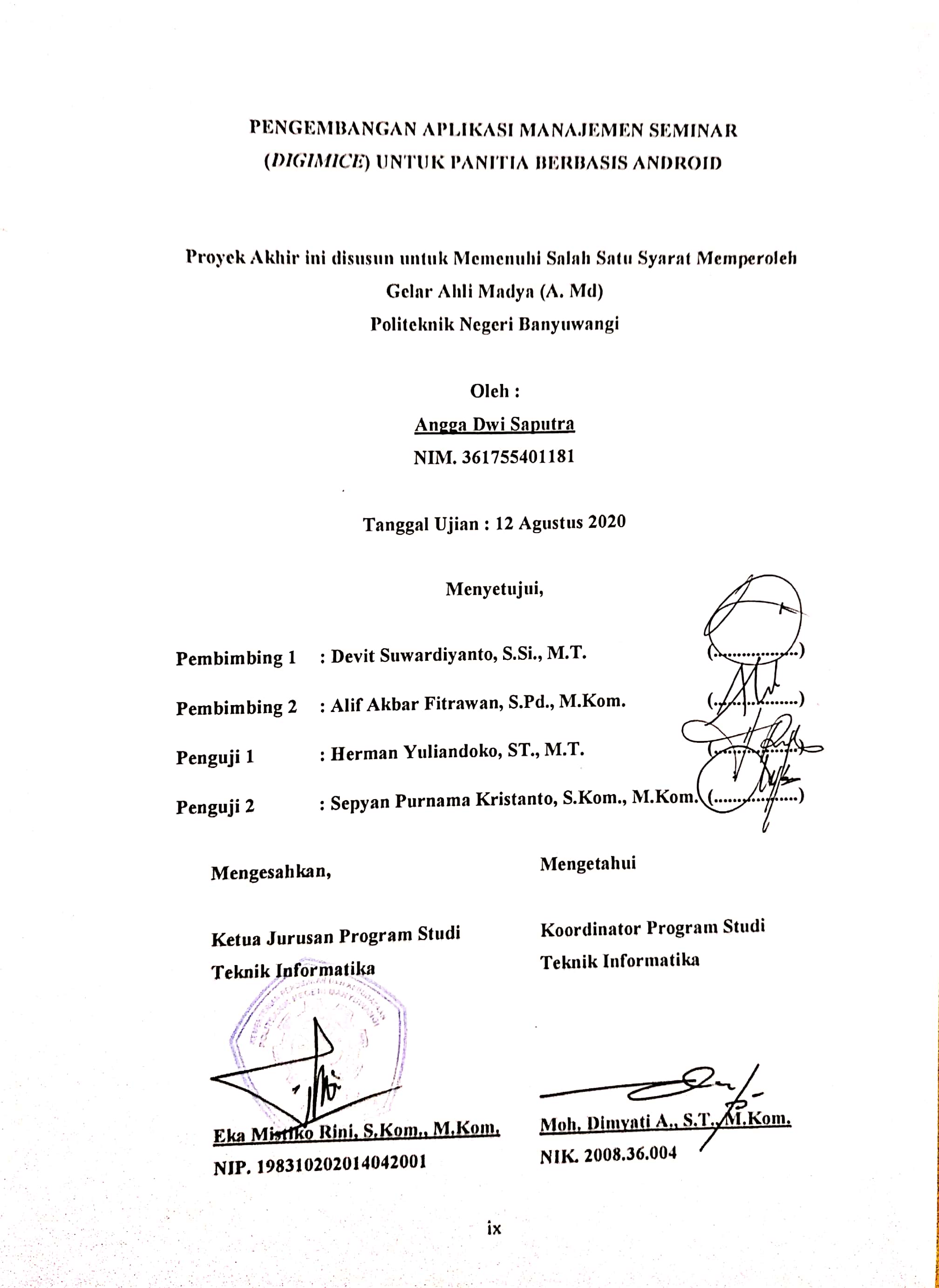
Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Banyuwangi, 12 Agustus 2020

Angga Dwi Saputra

NIM. 361755401181

*--- Halaman ini sengaja dikosongkan ---*

**PENGEMBANGAN APLIKASI MANAJEMEN SEMINAR**

**(*DIGIMICE*) UNTUK PANITIA BERBASIS ANDROID**

# PENGESAHAN

**Proyek Akhir ini disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Ahli Madya (A. Md)**

**Politeknik Negeri Banyuwangi**

**Oleh :**

**Angga Dwi Saputra**

**NIM. 361755401181**

**Tanggal Ujian : 12 Agustus 2020**

**Menyetujui,**

**Pembimbing 1 : Devit Suwardiyanto, S.Si., M.T. (..................)**

**Pembimbing 2 : Alif Akbar Fitrawan, S.Pd., M.Kom. (..................)**

**Penguji 1 : Herman Yuliandoko, ST., M.T. (..................)**

**Penguji 2 : Sepyan Purnama Kristanto, S.Kom., M.Kom. (..................)**

**Mengesahkan, Mengetahui**

**Ketua Jurusan Program Studi Koordinator Program Studi**

**Teknik Informatika Teknik Informatika**

**Eka Mistiko Rini, S.Kom., M.Kom. Moh. Dimyati A., S.T., M.Kom.**

**NIP. 198310202014042001 NIK. 2008.36.004**

*--- Halaman ini sengaja dikosongkan ---*

**PENGEMBANGAN APLIKASI MANAJEMEN SEMIAR**

**(*DIGIMICE*) UNTUK PANITIA BERBASIS ANDROID**

Nama Mahasiswa : Angga Dwi Saputra

NIM : 361755401181

Dosen Pembimbing : 1. Devit Suwardiyanto, S.Si., M.T.

2. Alif Akbar Fitrawan, S.Pd., M.Kom.

# ABSTRAK

*MICE* merupakan kepanjangan dari *Meeting*, *Incentive*, *Conference* dan *Exhibition*. *MICE* merupakan wisata konvensi yang bergerak di bidang pariwisata. *MICE* diselenggarakan oleh sekelompok panitia, salah satu kegiatan *MICE* adalah kegiatan seminar. Kegiatan seminar berupa membahas suatu topik dan mencari solusinya. Sebelum menjalankan suatu kegiatan, panitia selalu menyiapkan beberapa hal untuk mendukung jalannya kegiatan. Pada kegiatan seminar yang akan diselenggarakan, panitia menyiapkan beberapa hal seperti menyiapkan formulir pendaftaran, mengkonfirmasi pembayaran, bukti pembayaran, membuat absensi peserta, dan menyebarkan informasi kegiatan seminar melalui media sosial atau media cetak. Pada kegiatan seminar berlangsung panitia harus melakukan menyiapkan materi seminar, absensi peserta, dan merekap hasil absensi peserta yang sudah hadir dalam acara. Cara tersebut masih dianggap manual yang dapat memakan waktu yang lama dan rentan terhadap kesalahan data. Untuk mengatasi hal tersebut, dibuatlah Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android. Aplikasi ini dibuat berbasis Android dikarenakan penggunaan *smartphone* Android yang sangat banyak. Aplikasi ini merupakan aplikasi pengembangan dari aplikasi berbasis Web yang bernama *digiMICE*. Aplikasi ini dapat mempermudah panitia dalam mengelola kegiatan seminar yang akan diselenggarakan dengan cepat dan terhindar dari kesalahan data.

Kata kunci : Seminar, Panitia, Android, Pariwisata, *MICE*

*--- Halaman ini sengaja dikosongkan ---*

***DEVELOPMENT MANAGEMENT APPLICATION OF SEMINAR (DIGIMICE) FOR COMMITE BASED ON ANDROID***

*By* : Angga Dwi Saputra

*Student Identify Number* : 361755401181

*Supervisor* : 1. Devit Suwardiyanto, S.Si., M.T.

2. Alif Akbar Fitrawan, S.Pd., M.Kom.

# *ABSTRACT*

*MICE is a abbreviation for Meeting, Incentive, Conference and Exhibition. MICE is convention tourism in the tourism sector. MICE was held by a commite, one of the activities on MICE is a seminar event. The seminar event discuses about one of topics and search the solution. Before the commite actualize a seminar event, they always prepare some thing to support seminar event. When the seminar event will be held, the commite prepare some thing like make registration form, verify registration payment, make proof of payment, make presence form of the participant and share on social media or print media. Since the seminar event is running, the commite must serve the seminar materials, presence of the participant and recap the attending of the participant on the event. The method was assumed still manually which it can take a long time and vulnerable the data does not match. Management Application of Seminar (digiMICE) for Commite has been build to solve the problem. The application based on Android because so many people use the smartphone of Android. The application was developed from a web-based digiMICE. It can be easy to manage a seminar event which the seminar event will be held quickly and avoid from data does not match.*

*Keyword : Seminar, Commite, Android, Tourism, MICE*

*--- Halaman ini sengaja dikosongkan ---*

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat taufik dan hidayah-Nya sehingga laporan proyek akhir ini yang berjudul “Pengembangan Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android” ini dapat terselesaikan. Penulis menyadari bahwa proyek akhir ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Allah SWT yang selalu memberikan kemudahan dalam setiap proses penulisan proyek akhir ini.
2. Kedua Orang Tua dan keluarga penulis atas doa dan dukungan yang diberikan kepada penulis selama proses penulisan proyek akhir.
3. Bapak Son Kuswadi Dr. Eng selaku Direktur Politeknik Negeri Banyuwangi.
4. Ibu Eka Mistiko Rini, S.Kom., M.Kom. selaku Ketua Jurusan Teknik Informatika.
5. Bapak Moh. Dimyati A. S.T., M.Kom. selaku Koordinator Program Studi Teknik Informatika.
6. Bapak Devit Suwardiyanto, S.Si., M.T. selaku Dosen Pembimbing I dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom. selaku Dosen Pembimbing II yang telah membimbing dengan baik selama proses penulisan proyek akhir selesai.
7. Bapak Herman Yuliandoko, S.T., M.T. selaku Dosen Penguji I dan Bapak Sepyan Purnama Kristanto, S.Kom., M.Kom. selaku Dosen Penguji II yang telah meluangkan waktunya untuk menguji dan membimbing dengan baik sampai penulisan proyek akhir selesai.
8. Bapak dan Ibu Dosen Teknik Informatika Politeknik Negeri Banyuwangi yang telah membantu membimbing penulis menuntut ilmu selama perkuliahan berlangsung yang berperan atas berjalannya proyek akhir ini.

*--- Halaman ini sengaja dikosongkan ---*

1. 
2. Sahabat dan teman-teman penulis atas bantuan, motivasi, dan waktu untuk berdiskusi. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu yang selalu mendukung penulis.

Penulis menyadari bahwa proyek akhir ini masih terdapat banyak kekurangan. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun sehingga proyek akhir ini dapat bermanfaat.

Banyuwangi, 12 Agustus 2020

Angga Dwi Saputra

NIM. 361755401181

*--- Halaman ini sengaja dikosongkan ---*

# DAFTAR ISI

Halaman

[HALAMAN JUDUL i](#_Toc48765041)

[PERSEMBAHAN iii](#_Toc48765042)

[MOTTO v](#_Toc48765043)

[PERNYATAAN vii](#_Toc48765044)

[PENGESAHAN ix](#_Toc48765045)

[ABSTRAK xi](#_Toc48765046)

[*ABSTRACT* xiii](#_Toc48765047)

[KATA PENGANTAR xv](#_Toc48765048)

[DAFTAR ISI xix](#_Toc48765049)

[DAFTAR TABEL xxv](#_Toc48765050)

[DAFTAR GAMBAR xxix](#_Toc48765051)

[BAB 1 PENDAHULUAN 1](#_Toc48765052)

[1.1. Latar Belakang 1](#_Toc48765053)

[1.2. Perumusan Masalah 2](#_Toc48765054)

[1.3. Tujuan 3](#_Toc48765055)

[1.4. Manfaat 3](#_Toc48765056)

[1.5. Batasan Masalah 3](#_Toc48765057)

[BAB 2 TINJAUAN PUSTAKA 5](#_Toc48765058)

[*2.1.* *MICE* 5](#_Toc48765059)

[2.2. Seminar 6](#_Toc48765060)

[2.3. *Xendit* 8](#_Toc48765061)

[2.4. Web *API* 8](#_Toc48765062)

[*2.5.* *Lumen* 9](#_Toc48765063)

[*2.6.* *PHP* 10](#_Toc48765064)

[*2.7.* *MySQL* 11](#_Toc48765065)

[*2.8.* *Java* 11](#_Toc48765066)

[2.9. Android 13](#_Toc48765067)

[2.10. Android Studio 13](#_Toc48765068)

[*2.11.* Metodologi *eXtreme Programming* *(XP)* 14](#_Toc48765069)

*--- Halaman ini sengaja dikosongkan ---*

[*2.12.* *UML* *(Unified Modeling Language)* 14](#_Toc48765070)

[2.13. Pengujian *Blackbox Testing* 17](#_Toc48765071)

[2.14. Penelitian Terdahulu 19](#_Toc48765072)

[BAB 3 METODE PENELITIAN 23](#_Toc48765073)

[3.1. Waktu Pelaksanaan Penelitian 23](#_Toc48765074)

[3.2. Tempat Pelaksanaan Penelitian 23](#_Toc48765075)

[3.3. Jadwal Pelaksanaan Penelitian 23](#_Toc48765076)

[3.4. Metode Penelitian 23](#_Toc48765077)

[3.5. Gambaran Umum Sistem 25](#_Toc48765078)

[3.5.1. Sistem yang Sedang Berjalan 25](#_Toc48765079)

[3.5.2. Sistem yang Diusulkan 27](#_Toc48765080)

[3.6. Rancangan Sistem 29](#_Toc48765081)

[3.6.1. Spesifikasi Sistem 29](#_Toc48765082)

[*3.6.2.* *Use Case Diagram* 32](#_Toc48765083)

[3.6.3. *Activity* Diagram 43](#_Toc48765084)

[3.6.4. Struktur Menu 53](#_Toc48765085)

[3.6.5. *Mockup* Aplikasi 54](#_Toc48765086)

[*3.6.6.* *Entity Relational Diagram* 68](#_Toc48765087)

[BAB 4 HASIL DAN PEMBAHASAN 75](#_Toc48765088)

[4.1. Hasil 75](#_Toc48765089)

[4.1.1. *Login* Aplikasi 75](#_Toc48765090)

[4.1.2. *Register* Panitia 76](#_Toc48765091)

[4.1.3. Membeli Paket *Event* 77](#_Toc48765092)

[4.1.4. Mengelola *Event* 80](#_Toc48765093)

[4.1.5. Mengelola Daftar Sesi 81](#_Toc48765094)

[4.1.6. Mengelola Agenda 83](#_Toc48765095)

[4.1.7. Mengungah Materi 85](#_Toc48765096)

[4.1.8. Menambah Pemateri 86](#_Toc48765097)

[4.1.9. Menambah Anggota Panitia 87](#_Toc48765098)

[4.1.10. Mengelola Peserta 88](#_Toc48765099)

[4.1.11. Mengabsensi Peserta 89](#_Toc48765100)

[4.1.12. Melihat Rekapitulasi Peserta 90](#_Toc48765101)

*--- Halaman ini sengaja dikosongkan ---*

[4.1.13. *Logout* Aplikasi 91](#_Toc48765102)

[4.1.14. Mengubah Profil 92](#_Toc48765103)

[4.2. Pengujian 92](#_Toc48765104)

[4.3. Pembahasan 106](#_Toc48765105)

[BAB 5 PENUTUP 109](#_Toc48765106)

[1.1. Kesimpulan 109](#_Toc48765107)

[1.2. Saran 109](#_Toc48765108)

[DAFTAR PUSTAKA 111](#_Toc48765109)

[LAMPIRAN 115](#_Toc48765110)

[Lampiran 1. *Source Code* Pembuatan *Invoice* pada *payment gateway Xendit.* 115](#_Toc48765111)

[Lampiran 2. *Source Code* PembuatanNotifikasi *Email.* 119](#_Toc48765112)

[Lampiran 3. Biodata Penulis 121](#_Toc48765113)

*--- Halaman ini sengaja dikosongkan ---*

# DAFTAR TABEL

[Tabel 2. 1 *Use Case Diagram* 16](#_Toc34210909)

[Tabel 2. 2 *Activity Diagram* 17](#_Toc34210910)

[Tabel 2. 3 Penelitian Terdahulu 19](#_Toc34210911)

[Tabel 3. 1 Jadwal Penelitian Proyek Akhir 23](#_Toc45440653)

[Tabel 3. 2 Definisi Aktor 32](#_Toc45440654)

[Tabel 3. 3 Keterangan *Use Case Login* 33](#_Toc45440655)

[Tabel 3. 4 Keterangan *Use Case* Membeli Paket *Event* 33](#_Toc45440656)

[Tabel 3. 5 Keterangan *Use Case* Mengubah *Event* 35](#_Toc45440657)

[Tabel 3. 6 Keterangan *Use Case* Menambah Agenda 35](#_Toc45440658)

[Tabel 3. 7 Keterangan *Use Case* Melihat Daftar *Event* 36](#_Toc45440659)

[Tabel 3. 8 Keterangan *Use Case* Mengubah Agenda 36](#_Toc45440660)

[Tabel 3. 9 Keterangan *Use Case* Melihat Daftar Agenda 36](#_Toc45440661)

[Tabel 3. 10 Keterangan *Use Case* Menambah Anggota Panitia 37](#_Toc45440662)

[Tabel 3. 11 Keterangan *Use Case* Melihat Daftar Peserta 38](#_Toc45440663)

[Tabel 3. 12 Keterangan *Use Case* Mengabsensi Peserta 38](#_Toc45440664)

[Tabel 3. 13 Keterangan *Use Case* *Logout* 39](#_Toc45440665)

[Tabel 3. 14 Keterangan *Use Case* Mengunggah Materi 39](#_Toc45440666)

[Tabel 3. 15 Keterangan *Use Case* Menambah Pemateri 40](#_Toc45440667)

[Tabel 3. 16 Keterangan *Use Case* Melihat Sesi 41](#_Toc45440668)

[Tabel 3. 17 Keterangan *Use Case* Menambah Sesi 41](#_Toc45440669)

[Tabel 3. 18 Keterangan *Use Case* Mengubah Sesi 41](#_Toc45440670)

[Tabel 3. 19 Keterangan *Use Case* Mengubah Profil 42](#_Toc45440671)

[Tabel 3. 20 Keterangan *Use Case* *Register* 42](#_Toc45440672)

[Tabel 3. 21 Tabel *event\_type* 70](#_Toc45440673)

[Tabel 3. 22 Tabel *team* 70](#_Toc45440674)

[Tabel 3. 23 Tabel *user* 71](#_Toc45440675)

[Tabel 3. 24 Tabel *regencies* 71](#_Toc45440676)

[Tabel 3. 25 Tabel *participant\_group* 71](#_Toc45440677)

[Tabel 3. 26 Tabel *participant*\_*order*\_*history* 71](#_Toc45440678)

[Tabel 3. 27 Tabel *event*\_*presensi* 72](#_Toc45440679)

*--- Halaman ini sengaja dikosongkan ---*

[Tabel 3. 28 Tabel *event\_agenda* 72](#_Toc45440680)

[Tabel 3. 29 Tabel *event\_session* 72](#_Toc45440681)

[Tabel 3. 30 Tabel *event\_paket* 72](#_Toc45440682)

[Tabel 3. 31 Tabel *event\_order\_history* 73](#_Toc45440683)

[Tabel 3. 32 Tabel *event\_material* 73](#_Toc45440684)

[Tabel 3. 33 Tabel *regencies* 73](#_Toc45440685)

[Tabel 3. 34 Tabel *event* 73](#_Toc45440686)

[Tabel 3. 35 Tabel *participant* 74](#_Toc45440687)

[Tabel 3. 36 Tabel *provinces* 74](#_Toc45440688)

[Tabel 3. 37 Tabel *speakers* 74](#_Toc45440689)

[Tabel 3. 38 Tabel *participant\_refund\_order* 74](#_Toc45440690)

[Tabel 4. 1 Tabel *Blackbox Testing* 93](#_Toc46346969)

*--- Halaman ini sengaja dikosongkan ---*

# DAFTAR GAMBAR

[Gambar 3. 1 Metodologi *eXtreme Programming* 24](#_Toc45440695)

[Gambar 3. 2 Sistem yang Sedang Berjalan 25](#_Toc45440696)

[Gambar 3. 3 Sistem yang Diusulkan 27](#_Toc45440697)

[Gambar 3. 4 *Use Case* Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia 32](#_Toc45440698)

[Gambar 3. 5 *Activity* Diagram *Login* 43](#_Toc45440699)

[Gambar 3. 6 *Activity* Diagram *Register* 44](#_Toc45440700)

[Gambar 3. 7 *Activity* Diagram *Edit* Profil 44](#_Toc45440701)

[Gambar 3. 8 *Activity* Diagram Membeli Paket *Event* 45](#_Toc45440702)

[Gambar 3. 9 *Activity* Diagram Mengubah *Event* 46](#_Toc45440703)

[Gambar 3. 10 *Activity* Diagram Menambah Agenda 46](#_Toc45440704)

[Gambar 3. 11 *Activity* Diagram Melihat Daftar *Event* 47](#_Toc45440705)

[Gambar 3. 12 *Activity* Diagram Mengubah Agenda 47](#_Toc45440706)

[Gambar 3. 13 *Activity* Diagram Melihat Daftar Agenda 48](#_Toc45440707)

[Gambar 3. 14 *Activity* Diagram Menambah Anggota Panitia 48](#_Toc45440708)

[Gambar 3. 15 *Activity* Diagram Mengabsensi Peserta 49](#_Toc45440709)

[Gambar 3. 16 *Activity* Diagram Melihat Daftar Peserta 50](#_Toc45440710)

[Gambar 3. 17 *Activity* Diagram Mengunggah Materi 50](#_Toc45440711)

[Gambar 3. 18 *Activity* Diagram 51](#_Toc45440712)

[Gambar 3. 19 *Activity* Diagram *Logout* 51](#_Toc45440713)

[Gambar 3. 20 *Activity* Diagram Melihat Sesi 52](#_Toc45440714)

[Gambar 3. 21 *Activity* Diagram Menambah Sesi 52](#_Toc45440715)

[Gambar 3. 22 *Activity* Diagram Mengubah Sesi 53](#_Toc45440716)

[Gambar 3. 23 Struktur Menu 53](#_Toc45440717)

[Gambar 3. 24 Halaman *Login* 54](#_Toc45440718)

[Gambar 3. 25 Halaman *Register* 55](#_Toc45440719)

[Gambar 3. 26 Halaman Membeli Paket *Event* 55](#_Toc45440720)

[Gambar 3. 27 Halaman Detail Paket 56](#_Toc45440721)

[Gambar 3. 28 Halaman Kelola Paket 56](#_Toc45440722)

[Gambar 3. 29 Halaman Pembayaran 57](#_Toc45440723)

[Gambar 3. 30 Halaman *Checkout Detail* 58](#_Toc45440724)

*--- Halaman ini sengaja dikosongkan ---*

[Gambar 3. 31 Halaman Menambah Anggota Panitia 58](#_Toc45440725)

[Gambar 3. 32 Halaman Mengunggah Materi 59](#_Toc45440726)

[Gambar 3. 33 Halaman Detail *Upload* Materi 59](#_Toc45440727)

[Gambar 3. 34 Halaman Kelola *Event* 60](#_Toc45440728)

[Gambar 3. 35 Halaman *Edit* *Event* 60](#_Toc45440729)

[Gambar 3. 36 Halaman Daftar Agenda 61](#_Toc45440730)

[Gambar 3. 37 Halaman Tambah Agenda 61](#_Toc45440731)

[Gambar 3. 38 Halaman *Edit* Agenda 62](#_Toc45440732)

[Gambar 3. 39 Halaman Daftar Sesi 62](#_Toc45440733)

[Gambar 3. 40 Halaman Tambah Sesi 63](#_Toc45440734)

[Gambar 3. 41 Halaman *Edit* Sesi 63](#_Toc45440735)

[Gambar 3. 42 Halaman Daftar Melihat Daftar Pesertar 64](#_Toc45440736)

[Gambar 3. 43 Halaman Rekap Peserta 64](#_Toc45440737)

[Gambar 3. 44 Halaman Rekapitulasi Peserta 65](#_Toc45440738)

[Gambar 3. 45 Halaman Kelola Peserta 65](#_Toc45440739)

[Gambar 3. 46 Halaman Presensi Peserta 66](#_Toc45440740)

[Gambar 3. 47 Halaman *Scan QR Code* *Smartpone* 67](#_Toc45440741)

[Gambar 3. 48 Halaman *Logout* 67](#_Toc45440742)

[Gambar 3. 49 Halaman *Edit* Profil 68](#_Toc45440743)

[Gambar 3. 50 *Entity Relational Diagram* 69](#_Toc45440744)

[Gambar 4. 1 Tampilan Halaman *Login* 75](#_Toc46347063)

[Gambar 4. 2 Halaman Utama Ketua dan Anggota Panitia 76](#_Toc46347064)

[Gambar 4. 3 Tampilan Halaman *Register* 76](#_Toc46347065)

[Gambar 4. 4 Tampilan Halaman Pilih Paket 77](#_Toc46347066)

[Gambar 4. 5 Tampilan Halaman Detail Paket 77](#_Toc46347067)

[Gambar 4. 6 Tampilan Halaman Kelola Paket 78](#_Toc46347068)

[Gambar 4. 7 Tampilan Halaman Pembayaran 79](#_Toc46347069)

[Gambar 4. 8 Tampilan Halaman *Checkout Detail* 79](#_Toc46347070)

[Gambar 4. 9 Tampilan Halaman *Event* 80](#_Toc46347071)

[Gambar 4. 10 Tampilan Halaman *Edit Event* 81](#_Toc46347072)

[Gambar 4. 11 Tampilan Halaman Daftar Sesi 82](#_Toc46347073)

[Gambar 4. 12 Tampilan Halaman Tambah Sesi 82](#_Toc46347074)

*--- Halaman ini sengaja dikosongkan ---*

[Gambar 4. 13 Tampilan Halaman *Edit* Sesi 83](#_Toc46347075)

[Gambar 4. 14 Tampilan Halaman Daftar Agenda 83](#_Toc46347076)

[Gambar 4. 15 Tampilan Halaman Tambah Agenda 84](#_Toc46347077)

[Gambar 4. 16 Tampilan Halaman *Edit* Agenda 85](#_Toc46347078)

[Gambar 4. 17 Tampilan Halaman *Upload* Materi 85](#_Toc46347079)

[Gambar 4. 18 Tampilan Halaman Detail *Upload* Materi 86](#_Toc46347080)

[Gambar 4. 19 Tampilan Halaman Tambah Pemateri 87](#_Toc46347081)

[Gambar 4. 20 Tampilan Halaman Tambah Anggota 87](#_Toc46347082)

[Gambar 4. 21 Tampilan Halaman Peserta 88](#_Toc46347083)

[Gambar 4. 22 Tampilan Halaman Kelola Peserta 89](#_Toc46347084)

[Gambar 4. 23 Tampilan Halaman *Scan* *QR Code* 89](#_Toc46347085)

[Gambar 4. 24 Tampilan Halaman *Set QR Code* 90](#_Toc46347086)

[Gambar 4. 25 Tampilan Halaman Rekap Peserta 90](#_Toc46347087)

[Gambar 4. 26 Tampilan Halaman Rekapitulasi Peserta 91](#_Toc46347088)

[Gambar 4. 27 Tampilan Halaman Akun 91](#_Toc46347089)

[Gambar 4. 28 Tampilan Halaman *Edit* Profil 92](#_Toc46347090)

*--- Halaman ini sengaja dikosongkan ---*

# BAB 1 PENDAHULUAN

## Latar Belakang

Kepanjangan *MICE* sebagai *Meeting, Incentive, Conferent,* and *Exhibition* yang telah dikenal secara luas didunia dan menjadi istilah umum dalam industri pariwisata*.* Istilah *MICE* di Indonesia dikenal juga dengan nama wisata konvensi, kegiatan wisata konvensi ini merupakan bagian dari kegiatan pariwisata, karena banyak sekali menggunakan fasilitas pariwisata dalam pelaksanaannya, sehingga kegiatan ini merupakan kegiatan yang berkarakteristik padat karya, memberikan kontribusi yang baik dari sisi penyediaan tenaga kerja maupun dalam memberikan devisa negara (Noor, 2015). *MICE* merupakan salah satu bisnis dibidang pariwisata yang sangat menjanjikan.

Upaya pemerintahan dalam mengedepankan sektor pariwisata dinilai akan lebih pesat perkembangannya bila berbasis pada *MICE*. Kepala Badan Perencanaan Pembangunan Nasional (Bappenas) merujuk data perhelatan *World Economic Forum* di Davos tahun 2001, dimana ada 4600 peserta dan pengunjung serta diisi dengan 200 pertemuan. Dampak ekonomi langsung *World Economic Forum* terhadap Davos tercatat sebesar 15,7 juta *franc* (*₣*) dan dampak ekonomi tidak langsung sebesar 7 juta *franc* (*₣*). Faktanya biaya pengeluaran penyelenggaraan *World Economic Forum* sebesar 2 sampai 2,5 juta *franc* (*₣*) sehingga Davos mendapat untung hampir sekitar 20 juta *franc* (*₣*) (Putera, 2018).

Manajemen seminar adalah kegiatan mengatur dan merencanakan penyelenggaraan kegiatan seminar yang dilakukan oleh setiap organisasi atau kelompok. Menurut Kamus Besar Bahasa Indonesia seminar merupakan pertemuan atau persidangan untuk membahas suatu masalah dibawah pimpinan ahli (guru besar, pakar, dan sebagainya). Seminar dirancang dan diselenggarakan oleh suatu organisasi atau kelompok yang bertujuan untuk membahas suatu topik masalah beserta mencari solusinya dengan bentuk kegiatannya berupa presentasi yang diakhiri dengan sesi tanya jawab. Kegiatan seminar pada umumnya diikuti oleh moderator sebagai orang yang memimpin jalannya seminar, penyaji atau pemateri sebagai orang ahli atau narasumber yang memberikan pemaparan suatu topik pada seminar dan peserta sebagai pendengar dan penanya pada seminar.

Namun proses pengelolaan pada penyelenggaraan kegiatan seminar biasanya masih dilakukan secara manual. Bagi panitia menyediakan lembaran formulir pendaftaran, menyiapkan lembaran bukti pendaftaran, mengumpulkan bukti pembayaran, membuatkan lembaran absensi peserta dari data hasil pengumpulan bukti pembayaran. Proses manajemen yang dilakukan secara manual tersebut akan akan memakan waktu lama bagi panitia serta panitia rentan melakukan kesalahan saat mengelola penyelenggaraan seminar seperti kurang teliti dan kesalahan ketik data diri peserta.

Berdasarkan perhitungan dengan Laju Pertumbuhan Majemuk Tahunan *(Compound Annual Growth Rate)* atau disingkat *CAGR* pengguna Smartphone diperkirakan meningkat menjadi 140,4 Juta di tahun 2019 (*CAGR* 26,17 persen) (Permana, 2019). Melihat dari data tersebut, upaya untuk mengatasi masalahnya dengan dibuatkan Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android untuk mempermudah mengelola penyelenggaraan kegiatan seminar. Aplikasi ini merupakan aplikasi pengembangan dari aplikasi *digiMICE* berbasis web yang berfokus pada manajemen *MICE*. Aplikasi *digiMICE* dapat diakses melalui alamat *digimice.id*.

## Perumusan Masalah

Berdasarkan penjelasan dari latar belakang, maka permasalahan yang akan dikaji dapat dirumuskan sebagai berikut :

1. Bagaimana panitia mengelola penyelenggaraan seminar dengan menggunakan aplikasi?
2. Bagaimana menggunakan pembayaran *online* menggunakan *payment gateway* pada aplikasi manajemen seminar?

## Tujuan

Adapun tujuan dari masalah yang sudah dirumuskan sebagai berikut :

1. Membuat aplikasi manajemen seminar yang dapat mengelola kegiatan seminar baru, pendaftaran peserta, presensi peserta dan pembayaran penyelenggaraan seminar untuk panitia.
2. Membuat aplikasi manajemen seminar yang dapat melakukan pembayaran *online* menggunakan *payment gateway*.

## Manfaat

Adapun beberapa manfaat yang diperoleh :

1. Mempermudah panitia mengelola penyelenggaraan seminar dengan meninggalkan penggunaan cara manual pada proses pendaftaran, absensi dan pembayaran yang dapat mempercepat proses penyelenggaraan dan mengurangi kesalahan dalam penulisan data seminar.
2. Membantu beberapa panitia, kelompok atau organisasi dalam menyelenggarakan seminar dengan cara yang lebih efisien.

## Batasan Masalah

Adapun beberapa batasan masalah yang ada yang bertujuan agar supaya memfokuskan bagian-bagian yang akan dikerjakan. Diantaranya sebagai berikut :

1. Aplikasi ini menggunakan platform Android sehingga dapat dijalankan dengan perangkat yang menggunakan sistem operasi Android.
2. Aplikasi ini dikembangkan dari aplikasi berbasis web yang bernama *digiMICE* yang dapat diakses melalui alamat *digimice.id* dengan dibuat menggunakan *tools IDE Android Studio*.
3. Aplikasi ini dapat dijalankan dengan sistem operasi Android dengan minimal Android versi 5.0 *(Lolipop)* agar dapat berjalan dengan baik.
4. Aplikasi ini hanya mengelola penyelenggaraan kegiatan seminar.
5. Aplikasi ini dapat digunakan oleh dua pengguna yaitu ketua panitia dan anggota panitia.

*--- Halaman ini sengaja dikosongkan ---*

# BAB 2 TINJAUAN PUSTAKA

## *MICE*

*MICE* merupakan industri yang baru dan sedang berkembang dengan angka pertumbuhan yang cepat. Industri ini berasal dari Eropa dan Amerika Utara, dan sekarang menjadi industri internasional yang ditandai dengan adanya investasi lintas benua. Dengan perkembangan ilmu pengetahuan dan teknologi yang semakin cepat, kegiatan *MICE* sangat diperlukan oleh banyak perusahaan untuk mengembangkan usahanya yang dapat dilakukan dengan cara bertukar informasi atau saling bekerja sama yang ditempuh dengan cara pertemuan-pertemuan yang banyaknya pertemuan-pertemuan dilakukan antar negara, maka munculah bisnis *MICE* dan semakin meluas keseluruh dunia kerena adanya permintaan yang tinggi. Permintaan tempat penyelenggaraan ini mulai banyak dibangun di hampir setiap negara karena secara ekonomi memberikan kontribusi yang besar terhadap penyelenggara kegiatan ini.

Istilah *MICE* di Indonesia dikenal juga dengan nama wisata konvensi, kegiatan wisata konvensi ini merupakan bagian dari kegiatan pariwisata, karena banyak sekali menggunakan fasilitas pariwisata dalam pelaksanaannya, sehingga kegiatan ini merupakan kegiatan yang berkarakteristik padat karya, memberikan kontribusi yang baik dari sisi penyediaan tenaga kerja maupun dalam memberikan devisa negara (Noor, 2015).

Kepanjangan *MICE* sebagai *Meeting, Incentive, Conference,* and *Exhibition* yang telah dikenal secara luas didunia dan menjadi istilah umum dalam industri pariwisata. Beberapa definisi untuk *MICE* diberikan oleh para ahli (Noor, 2015) :

1. *Meeting*

*Meeting* berarti rapat, pertemuan, atau persidangan yang diselenggarakan oleh kelompok orang yang bergabung dalam suatu asosiasi, pekumpulan atau serikat dengan adanya tujuan yang hendak dicapai dalam organisasi (Pendit, 1999).

1. *Incentive*

Istilah Insentif diberikan oleh pemerintah Indonesia yang tercantum dalam Surat Keputusan Menparpostel Nomor KM.108/HM.703/MPPT-91, Bab 1 Pasal 1 Ayat b :

Perjalanan Insentive merupakan suatu kegiatan perjalanan yang diselenggarakan oleh suatu perusahaan untuk para karyawan dan mitra usaha sebagai imbalan penghargaan atas prestasi mereka dalam kaitan penyelenggaraan konvensi yang membahas perkembangan kegiatan perusahaan yang bersangkutan.

1. *Conference*

*Conference* atau dalam bahasa Indonesia dikenal dengan istilah Konferensi, didefinisikan oleh pemerintah yang tertuang dalam Surat Keputusan Menteri Pariwisata Pos dan Telekomunikasi Nomor : KM.108/HM.703/MPPT-91 sebagai Konferensi, Kongres, atau Konvensi merupakan suatu kegiatan berupa pertemuan sekelompok orang (negarawan, usahawan, dan sebagainya) untuk membahas masalah-masalah yang berkaitan dengan kepentingan bersama.

1. *Exhibition*

Menurut Surat Keputusan Menteri Pariwisata Pos dan Telekomunikasi Nomor yang sama diatas, Pasal 1c tentang *Exhibition* atau Pameran : merupakan suatu kegiatan untuk menyebarluaskan infomasi dan promosi yang ada hubungannya dengan penyelenggaraan konvensi atau yang ada kaitannya dengan pariwisata.

## Seminar

Seminar (*seminarium* – bahasa latin – tanah tempat menanam benih) merupakan suatu pertemuan ilmiah yang membahas suatu masalah yang diikuti banyak peserta dan mereka yang ahli di bidangnya yang pada akhirnya akan diperoleh suatu rumusan yang disepakati bersama.

Berikut penjelasan waktu, tempat, penyelenggaraan, dan ruang lingkup Seminar (Darsana, 2016) :

1. Waktu Seminar

Pertemuan ilmiah seperti seminar, waktunya tidak terlalu lama, maksimal pelaksanan dua hari, selain itu waktu pelaksanaan juga harus mempertimbangkan faktor-faktor seperti cuaca dan musim.

1. Tempat Seminar

Secara internal, seminar dapat dilakukan di dalam kampus – kampus perguruan tinggi dengan memanfaatkan auditorium, gedung serbaguna, atau pada aula di sekolah-sekolah. Untuk seminar yang berskala besar, seminar juga biasa diselenggarakan di Pusat Konvensi yang memang dibangun untuk keperluan seminar. Lokasi tempat seminar, untuk efektifnya sebaiknya diadakan ditengah atau dalam kota, tapi dengan pertimbangan tertentu, seminar juga dapat diselenggarakan di tempat yang jauh dari keramaian kota, misalnya ditempat-tempat peristirahatan, seperti daerah pegunungan atau daerah pantai.

1. Penyelenggaraan Seminar

Ada tiga pilihan untuk Penyelenggara Seminar, pilihan pertama adalah dibentuk sebuah panitia yang khusus untuk keperluan penyelenggaraan seminar. Pilihan kedua adalah menyerahkan penyelenggaraan seminar kepada perusahaan khusus bergerak dalam bidang penyelenggaraan seminar, atau yang lebih dikenal dengan *event organizer* *(EO).* Pilihan ketiga, dapat dibentuk panitia gabungan antara panitia lokal atau internet dengan panitia dari *event organizer*, ini dimaksud untuk pembelajaran bagi penyelenggaraan oleh panitia lokal pada masa-masa yang akan datang.

1. Ruang Lingkup Seminar

Dari ruang lingkup sebuah seminar kita dapat membedakan atas tiga tingkatan, yaitu:

* Seminar Lokal

Seminar ini mempunyai cakupan kecil, baik jumlah peserta maupun materi yang diseminarkan, dan biasanya dilaksanakan di kampus-kampus perguruan tinggi dengan pemakalah lokal dan pesertanya para mahasiswa perguruan tinggi tersebut.

* Seminar Nasional

Seminar ini cakupannya lebih luas dibandingkan dengan seminar lokal, pesertanya berasal dari berbagai temapt atau daerah, materinya yang tentunya lebih berbobot dan tempat penyelenggarannya biasanya dilakukan di hotel besar atau balai sidang.

* Seminar Internasional

Seminar ini dhadiri oleh peserta yang berasal dari dalam maupun luar negeri, membahas isu-isu global. Penyelenggaraan seminar internasional agak sedikit merepotkan dibandingkan semniar nasional, dimana harus menyediakan fasilitas seminar yang tidak diperlukan dalam seminar nasional, seperti penerjemah dan petugas yang bisa berbahasa asing.

## *Xendit*

*Xendit* adalah perusahaan penyedia layanan pembayaran di Indonesia yang membuat semua jenis transaksi menjadi mudah. *Xendit* memungkinkan pelanggan aktivitas untuk dapat menerima dan mengirim dana dengan proses integrasi yang sederhana. Setiap dan semua transaksi dapat dengan mudah ditemukan dan dicatat

dengan sistem deteksi penerimaan atau pengeluaran dana yang otomatis dan pemberitahuan yang instan.

*Xendit* memungkinkan suatu bisnis untuk menerima pembayaran melalui Virtual *Account* dan Kartu Kredit. *Xendit* juga menyediakan modul pengiriman yang memungkinkan suatu bisnis dapat mengirimkan dana dalam jumlah berapapun ke semua bank di Indonesia melalui *API* atau secara sederhana dengan *upload* sebuah file excel dengan format *CSV*. *Xendit* menjadi perusahaan teknologi pembayaran pertama di Indonesia dengan teknologi pengenalan otomatis, misi *Xendit* adalah untuk menyediakan solusi infrastruktur pembayaran otomatis untuk semua jenis bisnis (Xendit, n.d.).

## Web *API*

*API* adalah singkatan dari *Application Programming Interface* dan digunakan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols* dan *tools* lainnya yang memungkinkan developer untuk membuat aplikasi.

Tujuan pengunaan *API* adalah untuk mempercepat proses development dengan menyediakan *function* secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, dimana tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya, integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, library dan web. Program web *API* ini berbentuk alamat-alamat *URL* yang ketika diakses akan menampilkan data dalam format *JSON*.

*JSON* adalah singkatan untuk *Javascript Object Notation*, yang artinya adalah sebuah format untuk berbagi data. Seperti dapat kita lihat dari namanya, *JSON* diturunkan dari bahasa pemrograman *JavaScript*, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk *Phyton, Ruby, PHP dan Java. JSON* biasanya dilafalkan seperti nama “Jason”.

*JSON* menggunakan ekstensi *.json* saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam *.html*), ia dapat tampil didalam tanda petik sebagai *JSON* *string*, atau ia dapat dimasukkan ke dalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar server web dengan klien atau browser. Karena sangat mudah dibaca dan ringan, *JSON* memberikan alternatif lebih baik dari *XML* dan membutuhkan *formatting* yang tidak banyak (Yudhanto & Wijayanto, 2019).

## *Lumen*

Menurut Fauzi (2017) *Lumen* merupakan *Micro* *Framework* berbasis *PHP* yang juga diciptakan oleh Taylor Otwell seperti *Laravel*. *Lumen* dibuat khusus untuk akomondasi kebutuhan pengembang yang ingin membuat aplikasi dengan skala yang lebih kecil dari *Laravel* oleh karena itu banyak *library* yang dihilangkan. Tidak hanya itu, *Lumen* juga dapat digunakan untuk membuat *RESTful* *API*. Lebih menariknya jika *Lumen* dibandingkan dengan *Micro Framework* yang dikhususkan untuk *RESTful* *API* pada *PHP* seperti *Slim Framework* dan *Silex* hasil tes *Lumen* paling tinggi dengan hasil maksimal 1900 *request* (Maulidy & Nudin, 2019).

Untuk lebih jelasnya berikut fitur-fitur yang ada di *Lumen* (Maulidy & Nudin, 2019).

1. *Caching*, seperti halnya *laravel* yang memiliki *“.env”* untuk *caching* sistem untuk menentukan *driver* *cache* mana yang ingin digunakan secara *default* di seluruh aplikasi.
2. *Queues*, antrian dapat mengatur *API* sehingga memungkinkan menunda pemrosesan tugas yang memakan waktu, seperti melakukan tugas di server jauh, sampai waktu yang secara drastis mempercepat permintaan web ke aplikasi.
3. *Validation*, *Lumen* menggunakan *ProvidesConvenienceMethods* sebagai metode yang mudah digunakan untuk memvalidasi permintaan *HTTP* masuk dengan berbagai aturan validasi yang kuat.
4. *Routing*, Pengaturan terhadap *URL* yang terdiri dari beberapa *method* dan bisa menyematkan/mengambil parameter pada *URL* sesuai dengan aturan penulisannya.
5. *Middleware*, digunakan untuk memfilter permintaan *HTTP* yang masuk ke aplikasi. Misalnya, *Lumen* menyertakan *middleware* yang memverifikasi pengguna aplikasi Anda telah diautentikasi. Jika pengguna tidak diautentikasi, *middleware* akan mengalihkan pengguna ke layar masuk. Namun, jika pengguna diautentikasi, *middleware* akan mengizinkan permintaan untuk melangkah lebih jauh ke aplikasi.

## *PHP*

*PHP* singkatan dari *Hypertext Preprocesor* yang merupakan *server-side* programing, yaitu bahasa pemrograman yang diproses di sisi server. Fungsi utama *PHP* dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diubah, dihapus, dan ditampilkan pada *website* yang diatur oleh *PHP.* (Abdulloh, 2016).

Beberapa kelebihan *PHP* dari bahasa pemrograman web, antara lain (Lavarino & Yustanti, 2016) :

1. Bahasa pemrograman *PHP* adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunanya.
2. *PHP* memiliki tigkat akses yang lebih cepat.
3. *PHP* memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkemangan teknologi internet.
4. *PHP* juga mendukung akses ke beberapa *database* yang sudah ada baik yang bersifat *free*/gratis ataupun komersial. *Database* itu antara lain : *MySQL*, *PostgresSQL*, *infomix*, dan *MicrosoftSQL* Server. Web server yang mendukung *PHP* dapat ditemukan dimana mana dari mulai *Apache*, *IIS*, *AOserver, phttp. Fhttp. PWS, Lighttpd* hingga *Xitami* dengan konfigurasi yang relative mudah.

## *MySQL*

*MySQL* adalah sistem manajemen *database* *SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* *MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi-user* dan *SQL* *database* *managemen* *system* *(DBMS)*. *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan. Beberapa kelebihan dari menggunakan *MySQL* (MADCOMS, Divisi LITBANG, 2016) adalah :

1. *Source* *MySQL*  mudah dan gratis.
2. Sintaksnya lebih mudah dipahami.
3. Didukung program-program umum seperti *C, C++, Java, Perl, PHP* dan sebagainya.
4. Bekerja pada berbagai platform.
5. Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.
6. Mendukung *ODBC (Open Database Conectivity)* untuk *Windows*.

## *Java*

*Java* merupakan pemrograman yang sangat populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari kompter hingga smartphone. Bahasa pemrograman *Java* dikembangkan pertama kali oleh Sun Microsystem yang dimulai oleh James Gosling dan dirilis pada tahun 1995. Saat ini Sun Microsystem telah diakusisi oleh Oracle Corporation. Apabila sudah terbiasa dengan bahasa *C* dan *C++,* bahasa *Java* dapat dipelajari dengan cepat.

*Java* bersifat *Write One, Run Anywhere* (program yang ditulis satu kali dan dan dapat berjalan pada banyak platform). Dengan demikian tidak mengherankan apabila aplikasi yang dibuat menggunakan bahasa pemrograman *Java* bisa ditemukan di lingkungan komputer dan *smartphone* tanpa perbedaan.

*Java* merupakan bahasa pemrograman paling populer dan secara resmi dipelajari dibangku perkuliahan. Java memiliki berbagai kelebihan yang dirinci dalam setiap fiturnya. Berikut ini fitur-fitur *Java* (Jubilee Enterprise, 2015) :

* Berorientasi Objek : dalam *Java*, semua adalah Objek.
* Bersifat *Platform* *Independent* : *Java* di-*compile* dalam *bit* kode *platform* independen. Itu artinya, aplikasi yang ditulis menggunakan *java* dapat dengan mudah dibuka menggunakan berbagai sistem komputer dan arsitektur komputer.
* Sederhana : *Java* didesain untuk dapat dengan mudah dipelajari sehingga penetrasi pemrograman ini cukup tinggi dikalangan para pelajar.
* Aman : dengan fitur kemamanan *Java*, sehingga pengembang dapat membuat sistem yang bebas virus dan powerful.
* Bersifat *Architectural*-*neutral* : *Compiler* *Java* membuat format *file* objek yang *architectural-neutral*, yang membuat kode yang di-*compile* dapat dieksekusi pada berbagai jenis prosesor yang memiliki sistem *runtime* *Java*.
* *Portable* : *Java* bersifat portable karena adanya fitur *platform* *independent* dan *architectural-neutral*.
* Kuat dan powerful : Java mengeliminasi *error* dengan menjalankan pengecekan pada waktu *compile* dan *runtime*.
* *Multithreaded* : dengan adanya fitur *multithread*, pengembang dapat membuat program yang dapat mengerjakan banyak tugas sekaligus.
* Terinterpretasi : kode *bit* *Java* ditranslasi secara langsung pada instruksi mesin dan tidak disimpan.
* Performa tinggi : *Java* memiliki performa yang tinggi karena menggunakan *compiler* langsung.
* Terdistribusi : *Java* didesain untuk lingkungan distribusi internet.
* Dinamis : *Java* lebih dinamis dari *C* dan *C++* karena *Java* didesain untuk beradaptasi dengan lingkungan pengembangan.

## Android

Memilih sistem operasi Android untuk perangkat mobile adalah langkah yang tepat karena produk ini dibawah naungan *Google* sehingga pasti terus menerus dikembangkan. Sebagai developer bisa secara bebas membuat aplikasi berbasis Android. Ada juga sistem operasi lain seperti *iOS, Linux, Blackberry,* *Symbian* dan juga *Windows Mobile (WM)*. Semua memiliki kelebihan dan kekurangan masing-masing akan tetapi yang paling banyak dimiliki manusia di dunia saat ini adalah Android (Yudhanto & Wijayanto, 2019).

## Android Studio

*Android* *Studio* merupakan sebuah perangkat lunak *tools Integrated Development Environment (IDE)* untuk platform Android. ini diluncurkan pada tanggal 16 Mei 2013 pada Konferensi *Android Studio* *Google I/O* oleh Produk Manajer *Google*, Ellie Powers. Android Studio ini bersifat gratis di bawah *Apache License 2.0*. *Android Studio* awalnya dimulai dengan versi 0.1 pada bulan mei 2013, kemudian dibuat versi *beta* 0.8 yang rilis pada bulan juni 2014. Yang paling terbaru dirilis *v.3* pada bulan Oktober 2017. Berdasarkan *JetBrainns’ IntelliJ IDEA*, Studio didesain khusus untuk pengembangan Android. Sekarang ini *Android Studio* sudah bisa diunduh untuk *Windows, Mac OS X* dan *Linux* (Yudhanto & Wijayanto, 2019).

Berikut ini adalah beberapa hal yang akhirnya banyak mengundang banyak developer untuk melirik Android Studio sebagai software *pengembang* (Firly, 2019):

1. Dukungan dari *C++,* *NDK* dan sekarang *Kotlin.*
2. Perkembangan yang *up to date.*
3. Sistem berbasis *Gradle* yang dinilai fleksibel.
4. Lingkungan yang mencakup seluruh perangkat Android.
5. *Emulator* yang cepat dan kaya akan fitur.
6. Alat pengujian dan kerangka yang juga ekstensif.
7. *Instant Run* dan dukungan *Google Cloud Platform.*

## Metodologi *eXtreme Programming* *(XP)*

Menurut Suryantara (2017). *XP* merupakan salah satu metodologi dalam rekayasa perangkat lunak dan juga merupakan satu dari beberapa *agile software development methodologies* yang berfokus pada *coding* (penulisan kode program) sebagai aktivitas utama di semua tahap pada siklus pengembangan perangkat lunak *(software development lifecycle).* Metodologi ini mengedepankan proses pengembangan yang lebih *responsive* terhadap kebutuhan customer *(agile)* dibandingkan dengan metode-metode tradisional sambil membangun suatu aplikasi dengan kualitas yang lebih baik. Metodologi *XP* dapat diterapkan bila :

1. Adanya perubahan – perubahan yang sangat cepat.
2. Memiliki resiko tinggi pada aplikasi yang dibuat.
3. Adanya tantangan baru dalam mengembangkan aplikasi.
4. Dengan tim sedikit programmer (2 – 10) orang.
5. Keterlibatan/peran serta pelanggan secara langsung.

Metode ini menjadi sebuah pendekatan untuk pengembangan awal sebuah aplikasi, dengan melihat batas waktu pengerjaannya. Metode *XP* dikembangkan dengan tujuan untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif. *XP* juga bertujuan untuk mengurangi biaya selama ada perubahan dalam pengembangan perangkat lunak menggunakan siklus (tahapan) pengembangan perangkat lunak yang singkat. *XP* sangat cocok untuk pengembangan proyek yang memerlukan adaptasi cepat dalam perubahan-perubahan yang terjadi selama pengembangan aplikasi. *XP* juga cocok untuk anggota tim yang tidak terlalu banyak dan berada pada lokasi yang sama dalam pengembangan sistem.

## *UML* *(Unified Modeling Language)*

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML).* *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (S. & Shalahuddin, 2016).

Adapun kelebihan dan kekurangan dalam menggunakan *UML* antara lain sebagai berikut (Agarina & Jamil, 2018) :

* Kelebihan *UML* sebagai yaitu :

1. *Uniformity*

Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan.

1. *Understandability*

Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah untuk dipahami.

1. *Stability*

Kode program yang dihasilkan relatif stabil sepanjang waktu, karena mendekati permasalahan yang sesungguhnya.

1. *Reusability*

Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode sehingga pada akhirnya akan sangat mempercepat waktu pengembangan perangkat lunak.

* Kekurangan dari *UML* sebagai yaitu :

1. Pada awal desain, sistem mungkin akan sangat simple
2. Lebih fokus pada coding
3. Tidak menekankan pada kinerja team.
4. Tidak mudah untuk mendefinisikan class dan obyek yang dibutuhkan sistem.
5. *Use Case Diagram*

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakukan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang dibuat (S. & Shalahuddin, 2016).

Berikut adalah simbol-simbol yang ada pada diagram *use case* yang ditunjukkan pada Tabel 2.1 (S. & Shalahuddin, 2016).

Tabel 2. 1 *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| No | Simbol | Deskripsi |
| 1 | *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case.* |
| 2 | Aktor / *actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase. |
| 3 | Asosiasi / *association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
| 4 | Ekstensi / *extend* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek, *biasanya use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. |
| 5 | Generalisasi / *generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6 | Menggunakan / *include / uses* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. |

1. *Activity Diagram*

Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (S. & Shalahuddin, 2016).

Berikut adalah simbol-simbol yang ada pada diagram aktivitas yang ditunjukkan pada Tabel 2.2 (S. & Shalahuddin, 2016).

Tabel 2. 2 *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Deskripsi** |
| 1 | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2 | Aktivitras  aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 | Percabangan / decision | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 | Penggabungan / join | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 | Swimlane   |  | | --- | | Nama swimlane | |  | | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

## Pengujian *Blackbox Testing*

*Black Box Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Keuntungan dalam penggunaan metode *Blackbox Testing* adalah penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, pengujian yang dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan, dan *programmer* dan *tester* keduanya saling bergantung sama lain. Sedangkan kekurangan dari metode ini adalah uji kasus sulit didesain tanpa spesifikasi yang jelas, kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*, dan beberapa bagian *back end* tidak diuji sama sekali. (Jaya, 2018)

*Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. (Mustaqbal et al., 2015) *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan - pertanyaan berikut:

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
2. Masukan seperti apa yang dapat menjadi bahan kasus uji yang baik?
3. Apakah sistem sensitif pada masukan tertentu?
4. Bagaimana sekumpulan data dapat diisolasi?
5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Black Box Testing*, antara lain:

1. *Equivalence Partitioning*
2. *Boundary Value Analysis/Limit Testing*
3. *Comparison Testing*
4. *Sample Testing*
5. *Robustness Testing*
6. *Behavior Testing*
7. *Requirement Testing*
8. *Performance Testing*
9. Uji Ketahanan (*Endurance Testing*)
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*)

## Penelitian Terdahulu

Tabel 2. 3 Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| Unsur Perbandingan | (Nugroho, 2016) | (Adumy et al., 2019) | (Fatmawatie et al., 2016) |
| Judul | Aplikasi Web Informasi Dan Registrasi Peserta Seminar, Workshop, Talkshow Pada Acara Seminar Nasional Pengamplikasian Telematika (Sinaptika) | Sistem Absensi Seminar Menggunakan *QR Code* | Aplikasi Informasi Kegiatan Seminar Nasional untuk Dosen Universitas Telkom Berbasis Android dan Web |
| Fitur | Aplikasi untuk pendaftaran peserta seminar, workshop dan talkshow pada acara seminar nasional | Sistem untuk absensi peserta seminar dengan menggunakan *QR Code* | Aplikasi untuk menyediakan Informasi kegiatan seminar nasional untuk dosen Universitas Telkom yang berbasis Android dan web |
| Teknologi | Web, Boostrap, Google Form | Web, *QR Code* | Web, Android |
| Cakupan Sistem | Pendaftaran peserta dapat dilakukan melalui website dengan Google Form. | Sistem absensi peserta seminar menggunakan aplikasi web dengan *QR Code.* | Aplikasi penyedia informasi kegiatan seminar nasional untuk dosen berbasis Android dan web yang diakses dengan jaringan *local* di universitas telkom |
| Kelebihan | Mempermudah pendaftaran peserta pada acara seminar nasional | Mempermudah proses absensi peserta dengan cepat dengan menggunakan sistem absensi yang otomatis dengan menggunakan *QR Code* | Mempermudah untuk memberikan informasi kegiatan seminar nasional untuk dosen universitas telkom |
| Kekurangan | Hanya masih menggunakan website statis dengan menggunakan boostrap dan pengelolaan data menggunakan Google Form yang berbeda aplikasi. | Hanya digunakan pada proses absensi seminar untuk peserta seminar dengan menggunakan *QR Code* | Hanya digunakan untuk menampung segala informasi kegiatan seminar nasional untuk dosen universitas telkom dan di akses di jaringan *local* |
| Hasil | * Pada web Sinaptika pemakalah sudah dapat melihat tanggal penting yang isinya adalah kapan tanggal acara Sinaptika, waktu mulai acara, tempat pelaksanaan acara serta pembayaran yang harus dikeluarkan oleh pemakalah maupun peserta seminar Sinaptika. * Peserta seminar Sinaptika maupun peserta pada kegiatan acara Sinaptika dapat mengetahui namanya sudah terdaftar atau belum, selain itu peserta seminar juga dapat mengetahui informasi konfirmasi pembayaran baik melalui *whats app* ataupun melalui e-mail yang sudah diberikan informasinya pada *google form*. * Pihak panitia juga dapat langsung menghitung pengeluaran dan pemasukan dari pendaftaran pesertayang melakukan pendaftaran pada *google form*. | Sistem Absensi Seminar Berbasis *QR Code* ini  hanya bisa diakses oleh jaringan lokal, dan peserta akan mendapatkan kode unik setiap melakukan pendaftaran, yang selanjutnya kode unik tersebut bisa digunakan untuk authentikasi kehadiran peserta pada acara seminar. Jika peserta lupa dengan kode unik tersebut peserta bisa authentikasi menggunakan email dan password yang telah didaftarkan pada saat pendaftaran seminar. | * Aplikasi yang dibangun dapat memberikan informasi kegiatan seminar nasional berbasis android dan website * Aplikasi yang dibangun memberikan fasilitas kepada penyelenggara internal mengelola kegiatan seminar nasional * Aplikasi yang dibangun dapat memberikan notifikasi * Aplikasi yang dibangun memberikan fasilitas agar dosen dapat saling berbagi informasi kegiatan seminar nasional |

Tabel 2.3 merupakan beberapa contoh dari penelitian terdahulu yang menjadi referensi dalam pembuatan proyek akhir ini yang berjudul “Pengembangan Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android. Berdasarkan Tabel 2.3 terdapat beberapa parameter sebagai unsur pembanding dengan proyek akhir yang akan dibuat. Beberapa parameter tersebut diantaranya Fitur, Teknologi, Cakupan Sistem, Kelebihan, Kekurangan dan Hasil.

Berdasarkan data dari Tabel 2.3 judul pertama memiliki kekurangan aplikasi hanya digunakan untuk pendaftaran peserta seminar, judul kedua memiliki kekurangan sistem yang digunakan hanya untuk presensi peserta seminar dan judul ketiga memiliki kekurangan aplikasi yang digunakan hanya untuk penyedia informasi kegiatan seminar. Melihat dari kekurangan-kekurangan tersebut, penulis mengusulkan untuk membangun sebuah aplikasi yang dapat melakukan pengelolaan pada suatu kegiatan seminar untuk panitia. Pengelolaan yang dilakukan panitia dapat berupa mengelola absensi, pendaftaran dan pembayaran peserta seminar.

*--- Halaman ini sengaja dikosongkan ---*

# BAB 3 METODE PENELITIAN

## Waktu Pelaksanaan Penelitian

Penelitian ini dijadwalkan akan dilaksanakan selama kurang lebih 6 bulan (enam bulan) yang terhitung mulai bulan Januari 2020 s/d Juni 2020

## Tempat Pelaksanaan Penelitian

Tempat pembuatan aplikasi ini di Politeknik Negeri Banyuwangi yang mengambil studi kasus dari kegiatan seminar Surabaya Primary Care Forum.

## Jadwal Pelaksanaan Penelitian

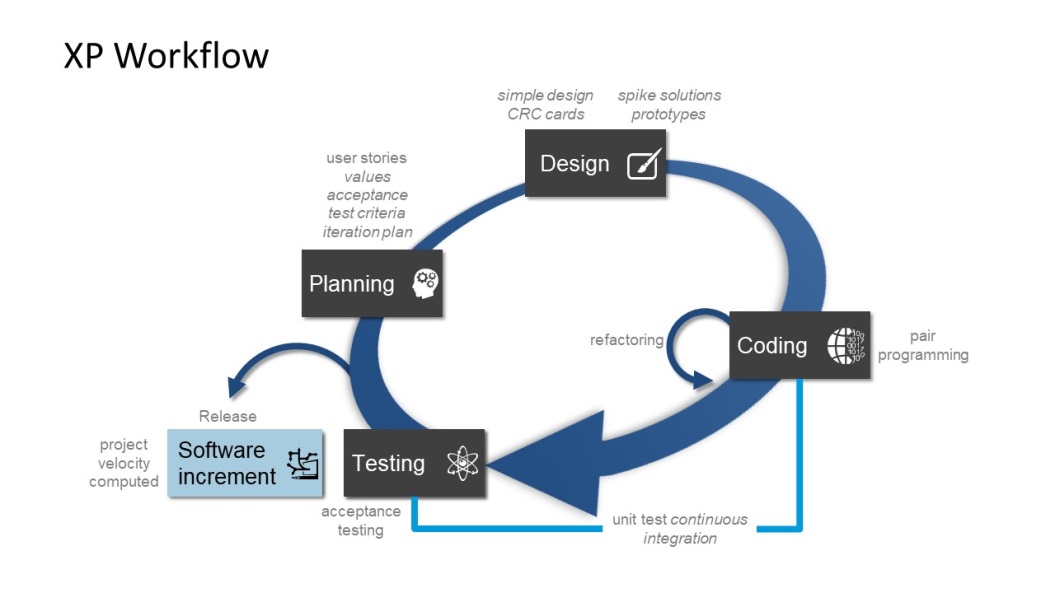
Metode penelitian ini dikarenakan menggunakan metodologi *eXtreme Programming*. Berikut adalah jadwal penelitian yang akan dilakukan sesuai waktu pelaksanaan dan mencakup tahapan perencanaan, desain sistem dan pengujian, rincian jadwal ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Jadwal Penelitian Proyek Akhir

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jenis Kegiatan | Tahun 2020 | | | | | |
| Januari | Februari | Maret | April | Mei | Juni |
| 1 | Perencanaan *(Planning)* |  |  |  |  |  |  |
| 2 | Desain Sistem *(Design)* |  |  |  |  |  |  |
| 3 | Pembuatan Aplikasi *(Coding)* |  |  |  |  |  |  |
| 4 | Pengujian *(Testing)* |  |  |  |  |  |  |

## Metode Penelitian

Proyek akhir ini dalam pelaksanaannya menggunakan salah satu metode *Agile* yaitu *eXtreme Programming (XP)*. Alur kerangka kerja metode ini dapat dilihat pada Gambar 3.1.



Sumber : Slidemodel.com

Gambar 3. 1 Metodologi *eXtreme Programming*

Pada Gambar 3.1 menunjukkan kerangka kerja dari metode *XP* ini yaitu :

* 1. *Planning* / Perencanaan

Pada tahap ini dalam pembuatan Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia dilakukan pemahaman konteks aplikasi melalui gambaran umum sistem, menganalisa spesifikasi sistem dengan menentukan fitur-fitur*,* fungsi aplikasi yang dibuat, penentuan waktu, serta alur proses pengembangan aplikasi.

* 1. *Design* / Desain

Pada tahap ini melakukan desain aplikasi yaitu *layout* aplikasi Android untuk merancang tampilan aplikasi yang akan dibuat agar mempermudah dalam proses *coding* aplikasi. Selain layout perancangan aplikasi digambarkan dengan membuat spesifikasi sistem, skenario sistem, *use case*, *activity diagram*, rancangan *database* *mysql* aplikasi web *digiMICE* untuk penyedia data di aplikasi Android dengan bantuan web *API*.

* 1. *Coding* / Penulisan Kode Program

Tahap ini melakukan proses pembuatan aplikasi berdasarkan *planning* dan *design* yang telah dilakukan. Proses *coding* aplikasi ini menggunakan bahasa *Java* dengan menggunakan *tools IDE Android Studio* dan pembuatan web *API* dengan menggunakan *micro-framework Lumen* untuk penyedia data yang akan diolah pada aplikasi android dari *database* aplikasi web *digiMICE* dimana akan menghasilkan sebuah aplikasi yang siap untuk dilakukan pengujian.

* 1. *Testing* / Percobaan dan Pengujian

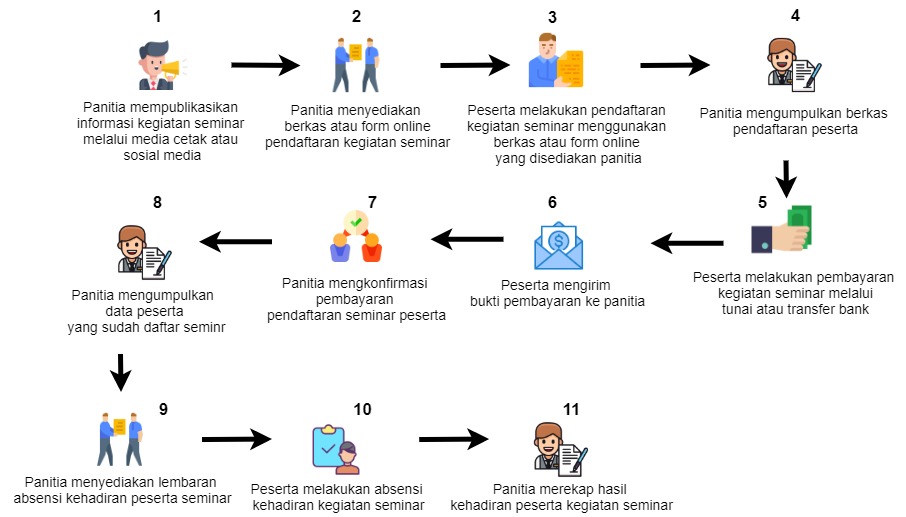
Tahap ini melakukan proses pengujian aplikasi yang difokuskan pada fitur-fitur yang telah dibuat agar mengetahui apakah fitur-fitur tersebut sudah berjalan dengan baik sehingga tidak terjadi kesalahan. Pengujian pada aplikasi ini menggunakan pengujian *blackbox testing,* selain itu diuji dengan digunakan oleh pengguna yang menyelenggarakan kegiatan seminar.

* 1. *Software Increement*

Pada tahap ini merupakan proses produksi aplikasi atau merilis aplikasi. Aplikasi ini akan dipublikasikan ke khalayak umum dengan mengunduhnya dengan melalui *Google Play Store*.

## Gambaran Umum Sistem

### Sistem yang Sedang Berjalan

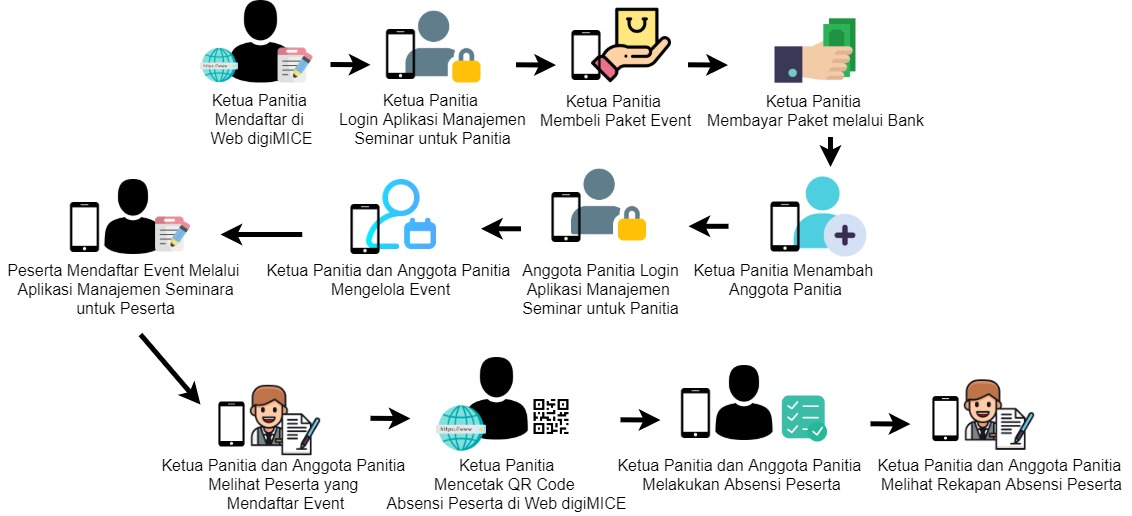
**

Gambar 3. 2 Sistem yang Sedang Berjalan

Pada Gambar 3.2 menunjukkan penjelasan langkah-langkah panitia dalam kegiatan seminar, Pada langkah nomor 1 menjelaskan ketika panitia sudah merencanakan kegiatan seminar, panitia menyebarkan informasi kegiatan seminar melalui media cetak seperti poster, pamflet atau *banner* atau melalui media sosial seperti *facebook*, *whatsapps* atau *instagram*. Kemudian langkah nomor 2 panitia menyediakan berkas-berkas yang dibutuhkan oleh peserta yang ingin mendaftar pada kegiatan seminar, berkas berupa lembaran *form* atau melalui *form* *online*. Pada nomor 3 peserta melakukan pendaftaran dengan mengisi lembaran berkas atau melalui *link form online* yang disediakan oleh panitia dan dikumpulkan ke panitia. Pada nomor 4 panitia mengumpulkan data peserta yang telah mendaftar melalui lembaran berkas atau *form online*. Setelah melakukan pendaftaran peserta pada nomor 5 melakukan pembayaran untuk tiket masuk kegiatan seminar melalui tunai atau transfer bank ke rekening panitia penyelenggara seminar yang tertera di informasi kegiatan seminar. Selanjutnya pada nomor 6 bukti pembayaran tersebut dikirimkan ke panitia, agar panitia dapat mengetahui peserta sudah membayar. Bukti berupa kwitansi atau struk pembayaran. Lalu pada nomor 7 panitia menerima bukti pembayaran peserta dan mendata peserta yang sudah membayar untuk mengetahui peserta yang pasti ikut kegiatan seminar. Setelah bukti pembayaran peserta terkumpul, pada nomor 8 panitia mengumpulkan data peserta yang sudah daftar dan membayar untuk mengikuti kegiatan seminar. Kemudian di nomor 9 panitia membuatkan lembaran absensi kehadiran peserta dari data peserta yang sudah daftar dan membayar. Berikutnya nomor 10 peserta datang ke kegiatan seminar lalu melakukan absensi kehadiran yang disediakan panitia dengan tanda tangan di lembaran absensi. Terakhir pada nomor 11 setelah kegiatan seminar selesai panitia merekap kedatangan peserta dari hasil absensi peserta yang sudah tanda tangan.

Melihat dari Gambar 3.2 gambaran sistem yang sedang berjalan, dapat disimpulkan langkah-langkah tersebut masih dianggap manual yang memerlukan waktu lebih lama dan kurang efisien dan masih rentan terjadinya kesalahan atau kehilangan berkas.

### Sistem yang Diusulkan



Gambar 3. 3 Sistem yang Diusulkan

Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android ini dibuat berfungsi untuk mempermudah panitia dalam mengelola penyelenggaraan kegiatan seminar dan mengurangi terjadinya kesalahan seminar hanya dengan menggunakan aplikasi yang dapat dilakukan dimana saja, cepat dan lebih sederhana.

Cara penggunaan aplikasi ini, pengguna harus membuka aplikasi dan akan diarahkan ke halaman *login*. Proses *login* untuk pada aplikasi ini akan membedakan hak akses pengguna sebagai ketua panitia atau anggota panitia ketika sudah masuk pada aplikasi, dalam proses *login* ini menggunakan akun ketua panitia yang sudah didaftarkan melalui aplikasi website *digiMICE.* Setelah melakukan *login* pengguna sebagai ketua panitia akan masuk ke halaman menu Panitia, dalam menu panitia ini terdapat halaman pilih paket *event*. Halaman ini berfungsi untuk memilih dan membeli paket penyelenggaraan kegiatan seminar berdasarkan isi paket yang dibutuhkan berdasarkan jumlah maksimal peserta. Setelah memilih paket, ketua panitia diarahkan ke halaman Detail Paket, lalu menekan tombol Kelola Paket akan diarahkan ke halaman Kelola Paket untuk menambahkan data *event* atau kegiatan seminar. Berikutnya setelah selesai mengisi data *event* ketua panitia menekan tombol Pilih Paket dan akan muncul pertanyaan yang menanyakan apakah yakin informasi yang ditambahkan sudah benar, jika dipilih “Iya” akan menuju halaman Pembayaran, jika dipilih “tidak” tetap di halaman Kelola Paket. Ketua panitia dapat langsung membayarkan tagihan yang harus dibayar sesuai yang ada pada halaman menu Pembayaran pada *tab* Belum Dibayar lalu pilih daftar pembayaran yang belum dibayar dan menuju halaman *Checkout* Detail. Pembayaran akan otomatis dibatalkan ketika panitia tidak membayar sesuai waktu yang telah ditentukan.

Panitia dapat melihat status pembayaran melalui menu Pembayaran disana terdapat tiga *tab* navigasi diantaranya *tab* Belum Dibayar, Batal dan Lunas. *Tab* Belum Dibayar berfungsi untuk menampilkan pembayaran yang belum dibayarkan, *tab* Batal berfungsi untuk menampilkan pembayaran yang dibatalkan dikarenakan ketua panitia tidak membayar pembayaran sesuai waktu yang telah ditentukan, jika proses pembayaran dibatalkan, *event* atau kegiatan seminar tidak dapat digunakan peserta untuk mendaftar dan *tab* Lunas berfungsi untuk menampilkan pembayaran yang berhasil dibayarkan dan *event* akan muncul di halaman *event*.

Setelah selesai melakukan pembayaran ketua panitia menambahkan anggota panitia pada halaman kelola *event* yang diakses melalui halaman *event*. Anggota panitia ditambahkan menggunakan *email user* peserta yang telah didaftarkan melalui pendaftaran dari aplikasi peserta atau menggunakan *email* aktif jika *email* belum terdaftar pada aplikasi. Setelah berhasil menambahkan anggota panitia, maka anggota panitia ketika *login* menggunakan akunnya akan diarahkan ke halaman seperti ketua panitia tetapi anggota panitia tidak bisa membeli paket untuk menambahkan *event* atau kegiatan seminar dan tidak bisa melakukan pembayaran paket serta tidak bisa melihat status pembayaran paket.

Kemudian menu Kelola *Event*, didalam menu ini terdapat halaman Daftar *Event* yang berfungsi untuk menampilkan *event* yang sudah ditambahkan dan ketika menekan tombol Kelola *Event* akan diarahkan ke halaman Kelola *Event* untuk mengubah data *Event* jika terjadi kesalahan ketika pembuatan *Event*. Pada halaman *Kelola* *Event* terdapat beberapa *tab* navigasi yaitu *tab* Edit *Event* untuk mengubah data *Event* yang salah dan *tab* Daftar Agenda untuk melihat daftar agenda yang sudah ditambahkan. *Tab* Daftar Sesi berfungsi untuk menambahkan pilihan sesi pada data agenda. Penambahan Agenda *Event* dapat dilakukan pada *tab* Daftar Agenda kemudian menekan tombol “+” pada aplikasi kemudian akan diarahkan ke Halaman tambah Agenda untuk menambahkan agenda. Pengubahan data Agenda ketika terjadi kesalahan data dapat dilakukan melalui *tab* Daftar Agenda lalu memilih agenda yang ingin diubah kemudian akan diarahkan ke halaman Edit Agenda. Lalu *tab* navigasi Upload Materi berfungsi untuk menambahkan materi kegiatan seminar dari setiap agendanya, di dalam *tab* navigasi Upload Materi berisi daftar *event* di dalamnya berisi daftar agenda event yang berisi beberapa materi kegiatan yang sudah ditambahkan. *Tab* navigasi Tambah Pemateri berfungsi untuk menambahkan akun pemateri kegiatan seminar.

Selanjutnya menu Kelola peserta, didalam menu ini terdapat dua *tab* navigasi yaitu *tab* Presensi Peserta dan Rekap Peserta. *Tab* Presensi Peserta berisi daftar *Event* yang ditambahkan berfungsi untuk melakukan absensi peserta ketika kegiatan peserta berlangsung. Setelah memilih event akan diarahkan ke halaman Detail *Event* yang didalamnya berisi jadwal agenda yang akan diselenggarakan pada *event*. Absensi dilakukan berdasarkan tiap sesi dilaksanakan, sesi memiliki beberapa agenda. Untuk melakukan Absensi dengan menekan tombol *Scan* *QR* dari tiap sesi kemudian akan diarahkan ke halaman *Scan QR*, di halaman *Scan QR* aplikasi meminta izin *smartphone* untuk mengakses kamera untuk membaca *QR Code* yang telah disediakan panitia kepada peserta. *Tab* Peserta berfungsi untuk melihat peserta yang sudah absensi dan sudah membayar tiket seminar serta hasil perekapan peserta seminar. Didalam *tab* Peserta terdapat hasil perekapan peserta seminar dan daftar peserta seminar.

Data dalam Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia ini disediakan oleh web *API* yang dirancang dengan menggunakan *micro-framework Lumen* yang berfungsi untuk menghubungkann *database* dari aplikasi web *digiMICE* ke Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android*.*

## Rancangan Sistem

### Spesifikasi Sistem

Spesifikasi sistem terdiri dari analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

* + - 1. **Analisis Kebutuhan Fungsional**

Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia ini memiliki fungsi mengelola penyelenggaraan kegiatan seminar yang akan diselenggarakan oleh panitia. Aplikasi ini dibuat untuk dua pengguna, yaitu pengguna pertama sebagai ketua panitia dan yang kedua sebagai anggota panitia. Aplikasi ini berfungsi agar mempercepat proses pengelolaan kegiatan seminar dan mengurangi kesalahan ketika proses pengelolaan serta aplikasi isi dapat digunakan oleh beberapa panitia penyelenggara kegiatan seminar, dimana aplikasi ini memiliki beberapa fitur utama yaitu kelola *event*, kelola panitia, kelola peserta, pembayaran, masuk dan keluar aplikasi.

1. Fitur kelola *event* digunakan untuk pengelolaan panitia dalam kegiatan seminar yang akan diselenggarakan panitia yang berisi informasi seperti nama kegiatan seminar, deskripsi, tempat dan waktu pelaksanaan serta gambar kegiatan seminar. Setelah mengelola *event* menambahkan informasi agenda-agenda seperti nama agenda, waktu dan deskripsi yang diselenggarakan pada kegiatan seminar. Seluruh data *event*, agenda, dan data lainnya disimpan dan dikelola pada web *API* yang terhubung pada *database* dari aplikasi web *digiMICE* untuk diolah oleh aplikasi Android. Fitur ini tersedia bagi pengguna sebagai ketua panitia dan anggota panitia
2. Fitur kelola panitia digunakan untuk mengelola urusan panitia sebagai ketua panitia. Urusan panitia yang hanya dikelola ketua panitia berupa menambahkan anggota panitia untuk membantu proses mengelola penyelenggaraan kegiatan seminar dan melakukan pemilihan paket seminar yang digunakan untuk membuat atau menambahkan kegiatan seminar yang akan diselenggarakan. Fitur ini tidak tersedia bagi anggota panitia.
3. Fitur mengunggah materi kegiatan seminar yang dapat diunduh dari aplikasi seminar untuk peserta.
4. Fitur menambah akun pemateri kegiatan seminar.
5. Fitur pembayaran digunakan untuk melakukan pembayaran pembuatan kegiatan seminar baru yang akan diselenggarakan dari hasil pemilihan paket yang dilakukan oleh ketua panitia. Untuk proses pembayaran *online* menggunakan *payment* *gateway Xendit* agar pembayaran otomatis dikonfirmasi oleh sistem. Fitur ini tidak tersedia bagi anggota panitia.
6. Fitur kelola peserta digunakan untuk proses absensi peserta ketika kegiatan seminar berlangsung dan melihat daftar peserta yang sudah terdaftar disetiap *event* beserta rekapitulasi data peserta dimasing-masing *event*.
7. Fitur masuk aplikasi digunakan untuk proses masuk (*login*) ke dalam aplikasi yang akan membedakan jenis pengguna sebagai ketua panitia atau anggota panitia.
8. Fitur keluar aplikasi digunakan untuk proses keluar (*logout*) dari aplikasi jika tidak ingin menggunakan aplikasi agar terhindar dari penyalahgunaan akun *user* yang digunakan.
   * + 1. **Analisa Kebutuhan Non Fungsional**
9. Analisis Perangkat Keras

Perangkat keras yang digunakan adalah *smartphone* yang akan menjalankan Aplikasi Manajemen Seminar (*digiMICE)* untuk panitia dengan spesifikasi berikut :

* Sistem Operasi : Minimal Android 5.0 *(Lolipop).*
* Kamera : Minimal 5 *MP*.
* *RAM* : Minimal 1 *Gb*
* *Free* *Storage* : 100 *Mb*

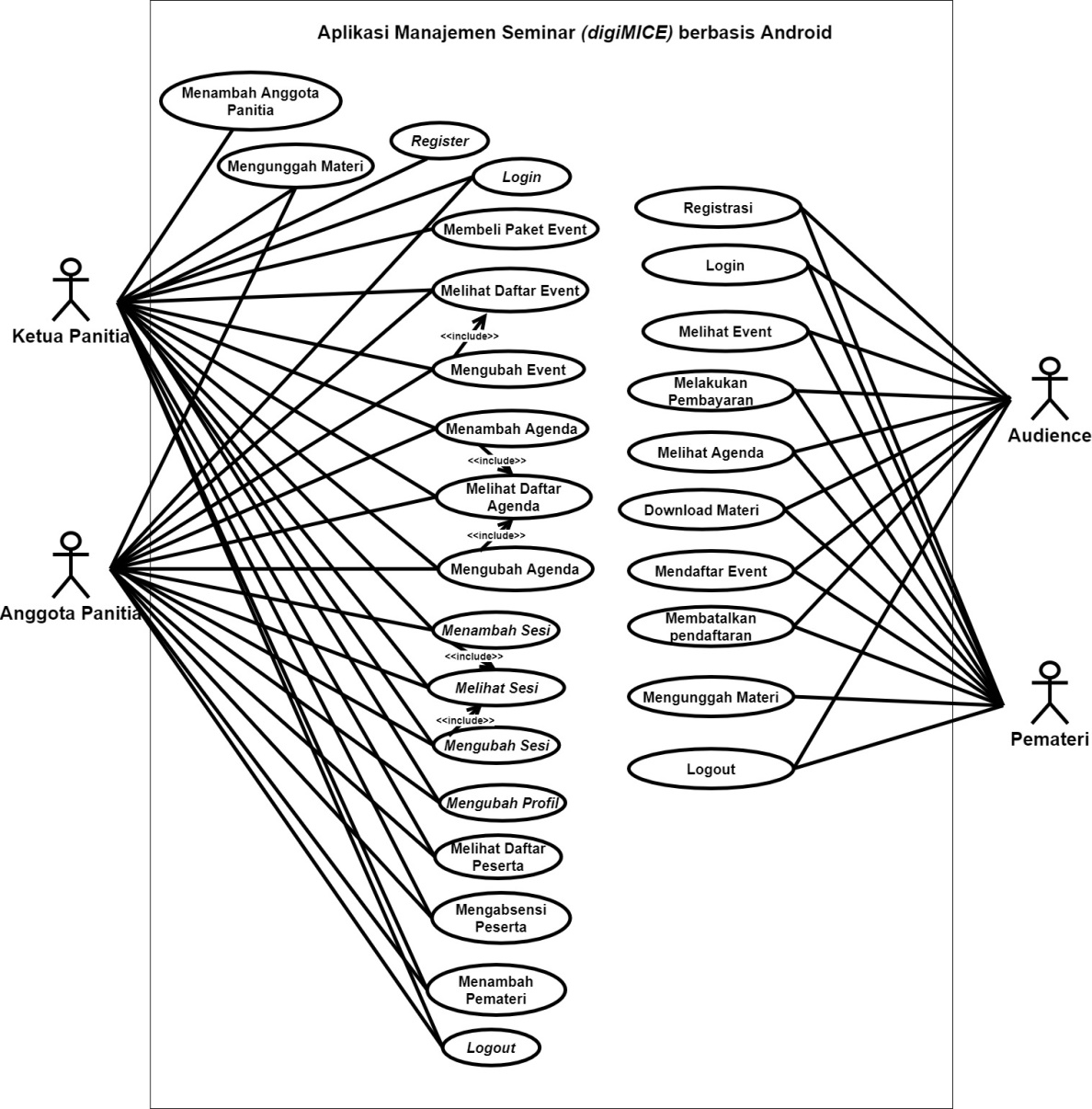
1. Analisis Pengguna

Aplikasi akan sangat digunakan jika memiliki target pengguna yang jelas, berikut target pengguna :

* Pengguna dapat mengoperasikan *smartphone* yang memiliki sistem operasi Android.
* Pengguna yang berkecimpung dalam penyelenggaraan seminar sebagai panitia.
* Pengguna dengan usia yang sudah termasuk usia produktif.
* Pengguna yang memiliki akses internet.

### *Use Case Diagram*

Pada Gambar 3.3 merupakan pemodelan sistem *use case diagram* Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia. Pada tabel 3.2 merupakan tabel definisi aktor pada *use case diagram* dan penjelasan tiap *use case* ditunjukkan pada tabel 3.3, tabel 3.4, tabel 3.5, tabel 3.6, tabel 3.7, tabel 3.8, tabel 3.9, tabel 3.10, tabel 3.11, tabel 3.12, tabel 3.13, tabel 3.14, tabel 3.15, tabel, 3.16, tabel 3.17, tabel 3.18, tabel 3.19 dan tabel 3.20.



Gambar 3. 4 *Use Case* Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia

Tabel 3. 2 Definisi Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Ketua Panitia | Ketua Panitia merupakan *user* yang memiliki hak akses membeli paket *event*, melihat status pembayaran, menambah anggota panitia, mengunggah materi, menambah pemateri, mengelola *event* dan mengelola peserta. |
| 2 | Anggota Panitia | Anggota Panitia merupakan *user* yang memiliki hak akses mengelola *event,* mengunggah materi, menambah pemateri dan mengelola peserta. |

Tabel 3. 3 Keterangan *Use Case Login*

|  |  |
| --- | --- |
| *Use case name* | *Login* |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor masuk ke aplikasi untuk menggunakan aplikasi sesuai hak akses aktor. |
| *Normal Course* | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman *login* 3. Aktor memasukkan *email* dan *password* di *form* *login* yang sudah terdaftar pada sistem 4. Aktor melakukan *login* 5. Sistem memasukkan aktor ke dalam aplikasi sesuai hak akses aktor. |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor memiliki *email* dan *password* |
| *Post-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |

Tabel 3. 4 Keterangan *Use Case* Membeli Paket *Event*

|  |  |
| --- | --- |
| *Use case name* | Membeli Paket *Event* |
| *Actor* | Ketua Panitia |
| *Description* | Aktor membeli paket *event*  untuk menambahkan *event* baru |
| *Normal Course* | 1. Aktor memilih menu panitia 2. Sistem menampilkan halaman pilih paket 3. Aktor memilih paket *event.* 4. Sistem menampilkan detail paket dan tombol kelola paket untuk menambahkan *event* baru 5. Aktor menekan tombol kelola paket 6. Sistem menampilkan halaman tambah *event* berisi *form* tambah *event* baru. 7. Aktor memasukkan data *event* baru 8. Aktor menekan tombol pilih paket 9. Sistem menampilkan pertanyaan “apakah sudah yakin” dengan pilihan “ya” dan “tidak” 10. Aktor memilih “ya” 11. Sistem menampilkan halaman pembayaran. 12. Aktor memilih *tab* belum dibayar 13. Sistem menampilkan daftar pembayaran yang belum dibayarkan oleh aktor 14. Aktor memilih salah satu dari daftar pembayaran yang belum dibayar 15. Sistem menampilkan halaman *checkout* detail 16. Aktor melihat isi halaman *checkout* detail 17. Aktor melakukan pembayaran 18. Aktor menekan tombol kembali 19. Sistem menampilkan halaman pembayaran pembayaran 20. Aktor memilih *tab* batal 21. Sistem menampilkan daftar pembayaran yang dibatalkan dikarenakan tidak membayar sesuai dihalaman *checkout* detail 22. Aktor melihat daftar pembayaran yang dibatalkan 23. Aktor memilih *tab* lunas 24. Sistem menampilkan daftar pembayaran yang sudah dibayarkan aktor 25. Aktor melihat daftar pembayaran yang sudah dibayarkan |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat daftar *event* yang sudah dibayar |

Tabel 3. 5 Keterangan *Use Case* Mengubah *Event*

|  |  |
| --- | --- |
| *Use case name* | Mengubah *Event* |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melakukan perubahan terhadap data *event* yang sudah dibayarkan jika ada kesalahan. |
| *Normal Course* | 1. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 2. Sistem menampilkan halaman kelola *event.* 3. Aktor memilih *tab* edit *event* yang berisikan *form* edit *event* 4. Sistem menampilkan halaman edit *event* 5. Aktor memasukkan dan mengirim data *event* 6. Sistem menyimpan perubahan data *event* dan menampilkan *alert* berhasil |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor melihat daftar *event* |
| *Post-Condition* | Aktor dapat melihat perubahan data *event* pada daftar *event* |

Tabel 3. 6 Keterangan *Use Case* Menambah Agenda

|  |  |
| --- | --- |
| *Use case name* | Menambah Agenda |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor menambahkan data agenda baru pada *event* |
| *Normal Course* | 1. Aktor menambahkan agenda baru dengan menekan tombol “+”. 2. Sistem menampilkan halaman *form* tambah agenda 3. Aktor memasukkan dan mengirim data agenda 4. Sistem menyimpan data agenda baru dan menampilkan alert berhasil |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor melihat daftar agenda |
| *Post-Condition* | Aktor dapat melihat data agenda baru pada daftar agenda |

Tabel 3. 7 Keterangan *Use Case* Melihat Daftar *Event*

|  |  |
| --- | --- |
| *Use case name* | Melihat Daftar *Event* |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melihat daftar *event* yang sudah ditambahkan |
| *Normal Course* | 1. Aktor memilih menu *event* di halaman utama 2. Sistem menampilkan daftar *event* |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat mengelola event seperti mengubah data *event* |

Tabel 3. 8 Keterangan *Use Case* Mengubah Agenda

|  |  |
| --- | --- |
| *Use case name* | Mengubah Agenda |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor mengubah data agenda yang sudah ditambahkan jika terjadi kesalahan |
| *Normal Course* | 1. Aktor mengubah data agenda dengan memilih agenda dari daftar agenda 2. Sistem menampilkan halaman *form* edit agenda 3. Aktor memasukkan dan mengirim data agenda 4. Sistem menyimpan perubahann data agenda dan menampilkan alert berhasil. |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor melihat daftar agenda |
| *Post-Condition* | Aktor dapat melihat data agenda yang diubah pada daftar agenda |

Tabel 3. 9 Keterangan *Use Case* Melihat Daftar Agenda

|  |  |
| --- | --- |
| *Use case name* | Melihat Daftar Agenda |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melihat daftar agenda yang sudah ditambahkan |
| *Normal Course* | 1. Aktor memilih menu kelola *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event.* 5. Aktor memilih *tab* daftar agenda 6. Sistem menampilkan halaman daftar agenda |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat mengelola agenda seperti menambah dan mengubah data agenda |

Tabel 3. 10 Keterangan *Use Case* Menambah Anggota Panitia

|  |  |
| --- | --- |
| *Use case name* | Menambah Anggota Panitia |
| *Actor* | Ketua Panitia |
| *Description* | Aktor menambahkan *user* baru sebagai anggota panitia untuk membantu aktor dalam mengelola penyelenggaraan seminar. |
| *Normal Course* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* tambah anggota 6. Sistem menampilkan halaman *form* tambah anggota 7. Aktor memasukkan dan mengirim data anggota 8. Sistem menyimpan data anggota dan menampilkan *alert* berhasil |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | *User* sebagai anggota panitia dapat melakukan *login* sesuai hak akses |

Tabel 3. 11 Keterangan *Use Case* Melihat Daftar Peserta

|  |  |
| --- | --- |
| *Use case name* | Melihat Daftar Peserta |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melihat data peserta dan rekapitulasi peserta *event* atau kegiatan seminar yang diselenggarakan |
| *Normal Course* | 1. Aktor memilih menu peserta 2. Sistem menampilkan halaman kelola peserta. 3. Aktor memilih *tab* rekap peserta 4. Sistem menampilkan daftar sesi. 5. Aktor memilih salah satu sesi. 6. Sistem menampilkan halaman rekapitulasi peserta berisi rekapitulasi peserta dan daftar peserta seminar. 7. Aktor mengamati rekapitulasi peserta dan daftar peserta |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat hasil rekapitulasi peserta seminar |

Tabel 3. 12 Keterangan *Use Case* Mengabsensi Peserta

|  |  |
| --- | --- |
| *Use case name* | Mengabsensi Peserta |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melakukan absensi peserta ketika *event* atau kegiatan seminar diselenggarakan |
| *Normal Course* | 1. Aktor memilih menu peserta 2. Sistem menampilkan halaman kelola peserta. 3. Aktor memilih *tab* presensi peserta 4. Sistem menampilkan jadwal sesi dan agenda *event.* 5. Aktor menekan tombol *scan QR.* 6. Sistem menampilkan halaman presensi peserta. 7. Aktor memilih *tab Scan QR Code* 8. Sistem menampilkan halaman *scan QR Code.* 9. Aktor memindai *QR Code* peserta untuk absensi. 10. Jika peserta menggunakan kartu aktor memilih *tab* *Set Qt Code.* 11. Sistem menampilkan halaman *set QR Code*. 12. Aktor memindai *QR Code* kartu peserta dan menekan tombol *set QR.* 13. Sistem menyimpan data absensi peserta 14. Sistem merekap data absensi peserta. |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat daftar peserta seminar |

Tabel 3. 13 Keterangan *Use Case* *Logout*

|  |  |
| --- | --- |
| *Use case name* | *Logout* |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor keluar dari aplikasi untuk tidak menggunakan hak aksesnya lagi |
| *Normal Course* | 1. Aktor memilih menu akun 2. Sistem menampilkan halaman akun yang berisikan informasi akun dan tombol *logout.* 3. Aktor menekan tombol *logout* 4. Sistem mengeluarkan aktor dari aplikasi dan menampilkan halaman *login* |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor sudah keluar dari hak akses dan berada di halaman *login* |

Tabel 3. 14 Keterangan *Use Case* Mengunggah Materi

|  |  |
| --- | --- |
| *Use case name* | Mengunggah Materi |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor menambahkan materi kegiatan seminar pada tiap-tiap agenda. |
| *Normal Course* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab Upload* Materi 6. Sistem menampilkan daftar agenda *event* 7. Aktor memilih daftar agenda *event* 8. Sistem menampilkan daftar materi agenda *event* 9. Aktor mengunggah materi agenda *event* 10. Sistem menyimpan materi agenda *event* |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Peserta mengunduh materi kegiatan seminar |

Tabel 3. 15 Keterangan *Use Case* Menambah Pemateri

|  |  |
| --- | --- |
| *Use case name* | Menambah Pemateri |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor menambahkan pemateri kegiatan seminar yang dilaksanakan |
| *Normal Course* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* TambahPemateri 6. Sistem menampilkan halaman tambah pemateri 7. Aktor memasukkan dan mengirim data pemateri 8. Sistem menyimpan data pemateri |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Pemateri melakukan login melalui aplikasi peserta |

Tabel 3. 16 Keterangan *Use Case* Melihat Sesi

|  |  |
| --- | --- |
| *Use case name* | Melihat Sesi |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor melihat sesi kegiatan seminar yang dilaksanakan |
| *Normal Course* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* Daftar Sesi 6. Sistem menampilkan halaman daftar sesi |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat, mengubah, dan menambah data sesi |

Tabel 3. 17 Keterangan *Use Case* Menambah Sesi

|  |  |
| --- | --- |
| *Use case name* | Menambah Sesi |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor menambahkan sesi kegiatan seminar yang dilaksanakan |
| *Normal Course* | 1. Aktor menekan tombol “+” 2. Sistem menampilkan halaman tambah sesi 3. Aktor memasukkan dan mengirim data sesi 4. Sistem menyimpan data sesi |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor melihat sesi |
| *Post-Condition* | Aktor dapat melihat data sesi baru pada halaman daftar sesi |

Tabel 3. 18 Keterangan *Use Case* Mengubah Sesi

|  |  |
| --- | --- |
| *Use case name* | Mengubah Sesi |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor mengubah sesi kegiatan seminar yang dilaksanakan |
| *Normal Course* | 1. Aktor memilih data sesi 2. Sistem menampilkan halaman edit sesi 3. Aktor memasukkan dan mengirim data sesi 4. Sistem menyimpan data sesi |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor melihat sesi |
| *Post-Condition* | Aktor dapat melihat data sesi yang diubah pada halaman daftar sesi |

Tabel 3. 19 Keterangan *Use Case* Mengubah Profil

|  |  |
| --- | --- |
| *Use case name* | Mengubah Profil |
| *Actor* | Ketua Panitia dan Anggota Panitia |
| *Description* | Aktor mengubah profil *user* |
| *Normal Course* | 1. Aktor memilih menu akun 2. Sistem menampilkan halaman akun 3. Aktor menekan tombol edit profil 4. Sistem menampilkan halaman edit profil 5. Aktor mengisi dan mengirim data profil yang akan diubah. 6. Sistem menyimpan data profil yang diubah |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor masuk dalam aplikasi sesuai hak akses |
| *Post-Condition* | Aktor dapat melihat data profil yang diubah pada halaman akun |

Tabel 3. 20 Keterangan *Use Case Register*

|  |  |
| --- | --- |
| *Use case name* | *Register* |
| *Actor* | Ketua Panitia |
| *Description* | Aktor membuat akun baru |
| *Normal Course* | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman login 3. Aktor menekan tombol register 4. Sistem menampilkan halaman register panitia 5. Aktor mengisi dan mengirim data akun panitia. 6. Sistem menyimpan data akun panitia |
| *Alternative Course* | - |
| *Pre-Condition* | Aktor belum memiliki akun panitia untuk login aplikasi |
| *Post-Condition* | Aktor melakukan login dengan akun terdaftar |

### *Activity* Diagram

*Activity* diagram dari Aplikasi Manajemen Seminar (*digiMICE)* untuk Panitia berbasis Android berfungsi untuk menggambarkan alur dalam proses mengelola penyelenggaraan kegiatan seminar dari *use case* yang telah dibuat. Berikut penjelasan dari tiap-tiap *activity* diagram aplikasi dibawah ini :

1. *Activity* Diagram *Login*

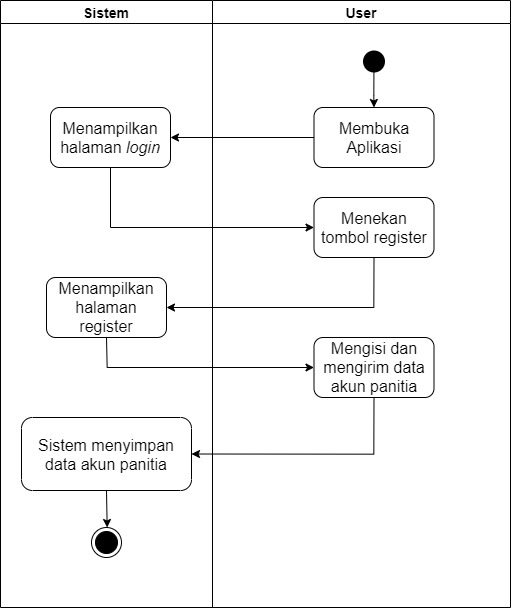
Pada Gambar 3.5 merupakan *acitivity* diagram yang menjelaskan aktivitas proses login.



Gambar 3. 5 *Activity* Diagram *Login*

1. *Activity* Diagram *Register*

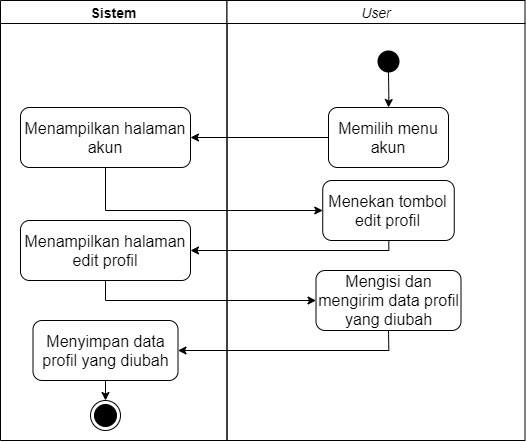
Pada Gambar 3.6 merupakan *activity* diagram yang menjelaskan aktivitas proses *register* akun panitia sebagai ketua panitia



Gambar 3. 6 *Activity* Diagram *Register*

1. *Activity* Diagram *Edit* Profil

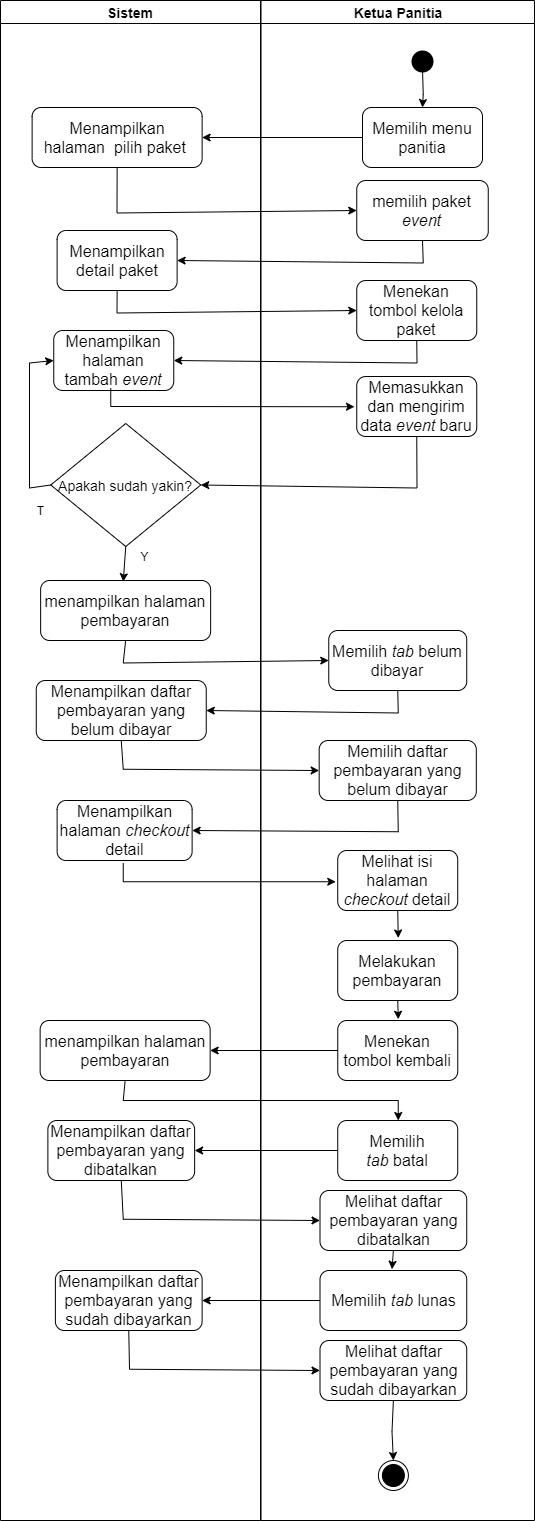
Pada Gambar 3.7 merupakan *activity* diagram yang menjelaskan aktivitas proses edit profil.

****

Gambar 3. 7 *Activity* Diagram *Edit* Profil

1. *Activity* Diagram Membeli Paket *Event*

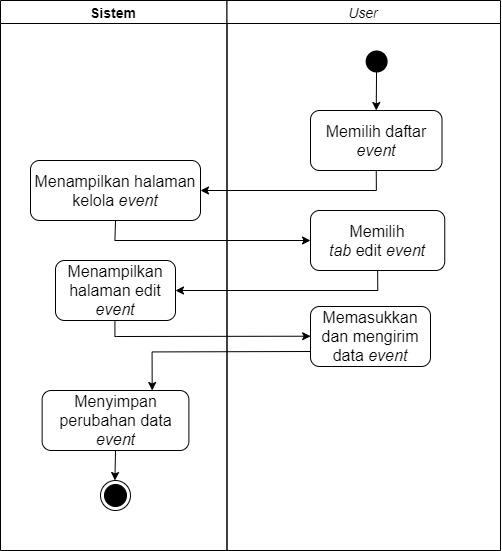
Pada Gambar 3.8 merupakan *activity* diagram yang menjelaskan proses membeli paket event untuk menambahkan event baru..



Gambar 3. 8 *Activity* Diagram Membeli Paket *Event*

1. *Activity* Diagram Mengubah *Event*

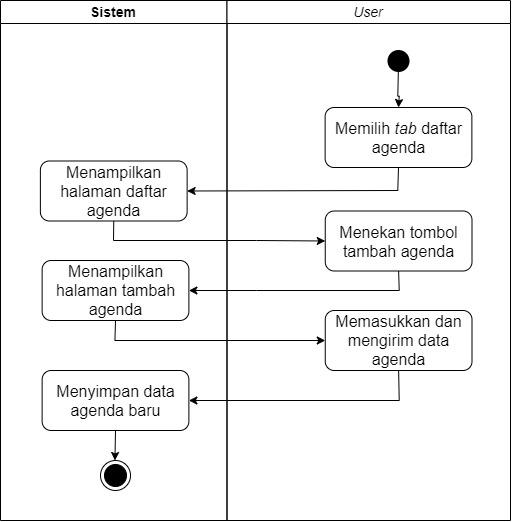
Pada Gambar 3.9 merupakan *activity* diagram yang menjelaskan proses aktivitas mengubah *event*.



Gambar 3. 9 *Activity* Diagram Mengubah *Event*

1. *Activity* Diagram Menambah Agenda

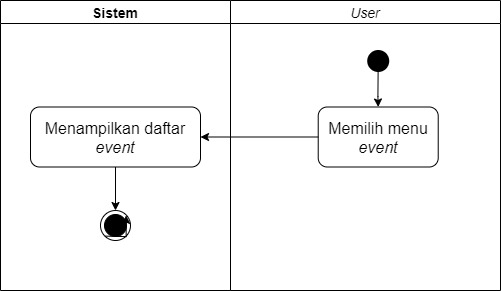
Pada Gambar 3.10 merupakan *activity* diagram yang menjelaskan aktivitas proses menambah agenda *event*.



Gambar 3. 10 *Activity* Diagram Menambah Agenda

1. *Activity* Diagram Melihat Daftar *Event*

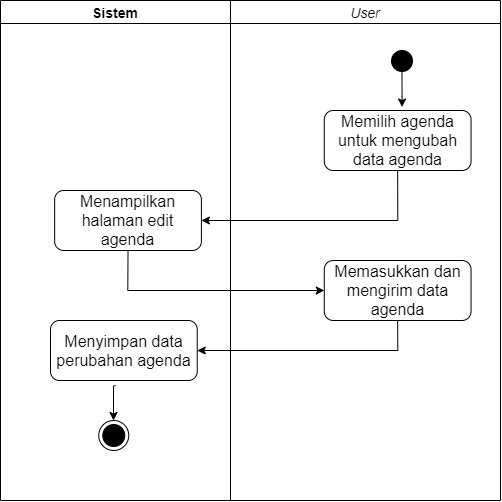
Pada Gambar 3.11 merupakan *activity* diagram yang menjelaskan aktivitas proses melihat daftar *event*.



Gambar 3. 11 *Activity* Diagram Melihat Daftar *Event*

1. *Activity* Diagram Mengubah Agenda

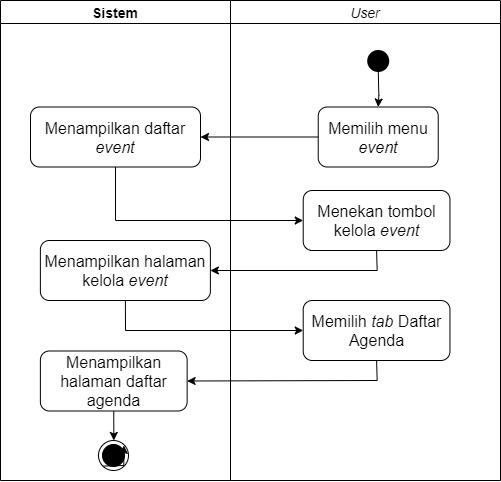
Pada Gambar 3.12 merupakan *activity* diagram mengubah agenda.



Gambar 3. 12 *Activity* Diagram Mengubah Agenda

1. *Activity* Diagram Melihat Daftar Agenda

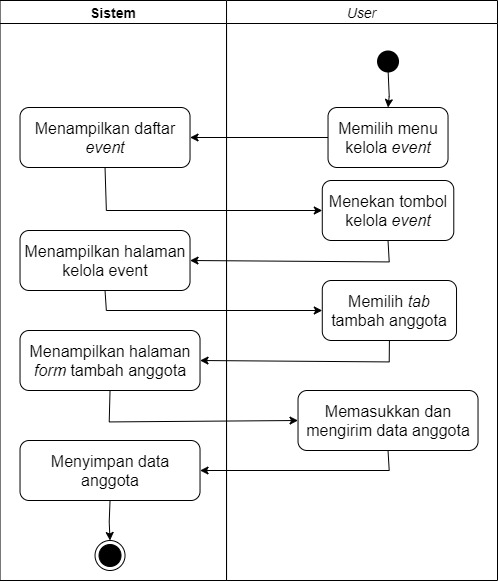
Pada Gambar 3.13 merupakan *activity* diagram yang menjelaskan aktivitas proses melihat daftar agenda



Gambar 3. 13 *Activity* Diagram Melihat Daftar Agenda

1. *Activity* Diagram Menambah Anggota Panitia

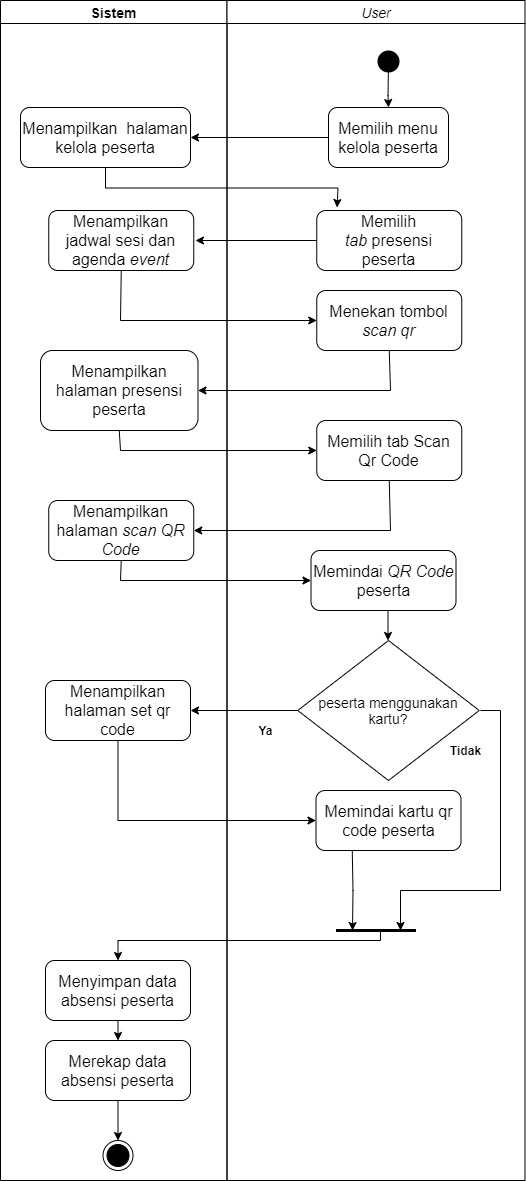
Pada Gambar 3.14 merupakan *activity* diagram yang menjelaskan aktivitas proses menambah anggota panitia

****

Gambar 3. 14 *Activity* Diagram Menambah Anggota Panitia

1. *Activity* Diagram Mengabsensi Peserta

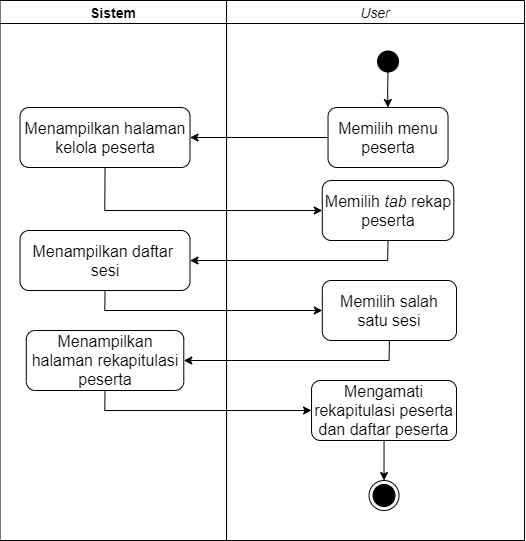
Pada Gambar 3.15 merupakan *activity* diagram yang menjelaskan aktivitas proses mengabsensi peserta.



Gambar 3. 15 *Activity* Diagram Mengabsensi Peserta

1. *Activity* Diagram Melihat Daftar Peserta

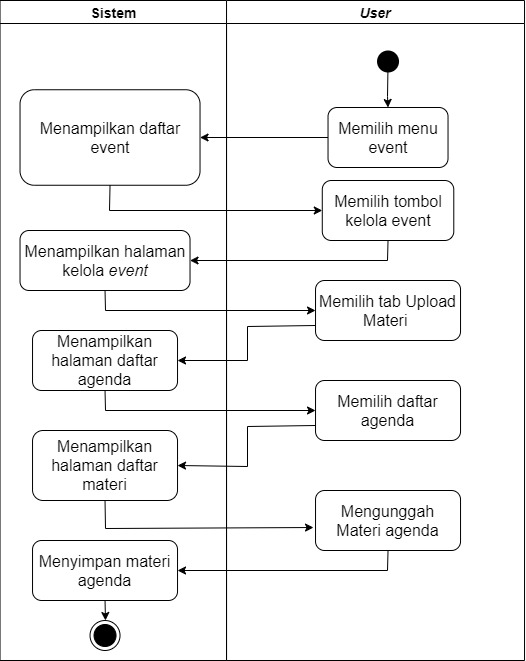
Pada Gambar 3.16 merupakan *activity* diagram yang menjelaskan aktivitas proses melihat daftar peserta.

****

Gambar 3. 16 *Activity* Diagram Melihat Daftar Peserta

1. *Activity* Diagram Mengunggah Materi

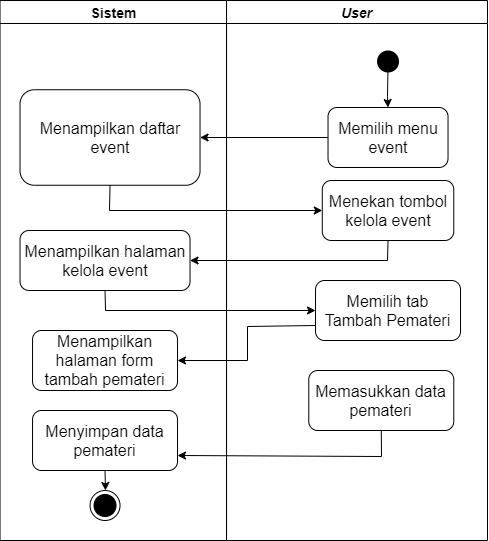
Pada Gambar 3.17 merupakan *activity* diagram yang menjelaskan aktivitas proses mengunggah materi kegiatan seminar.



Gambar 3. 17 *Activity* Diagram Mengunggah Materi

1. Activity Diagram Menambah Pemateri

Pada Gambar 3.18 merupakan *activity* diagram yang menjelaskan aktivitas proses penambahan akun pemateri.



Gambar 3. 18 *Activity* Diagram

1. *Activity* Diagram *Logout*

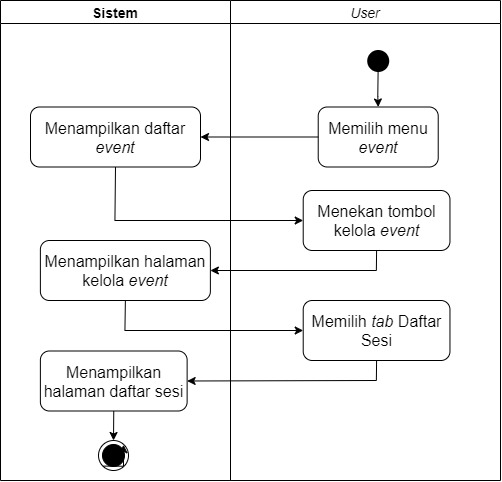
Pada Gambar 3.19 merupakan *activity* diagram yang menjelaskan aktivitas proses *logout*.



Gambar 3. 19 *Activity* Diagram *Logout*

1. *Activity* Diagram Melihat Sesi

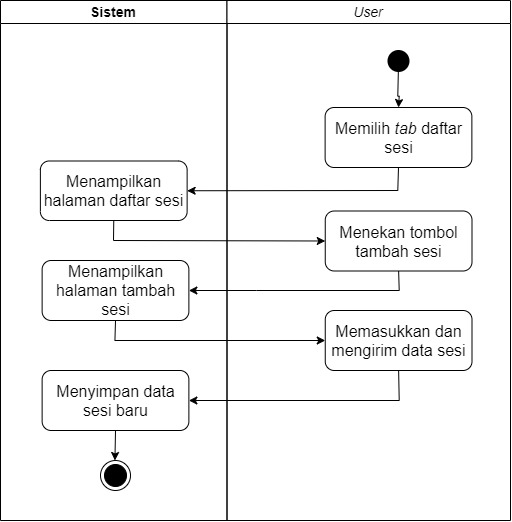
Pada Gambar 3.20 merupakan *activity* diagram yang menjelaskan aktivitas proses melihat sesi.



Gambar 3. 20 *Activity* Diagram Melihat Sesi

1. *Activity* Diagram Menambah Sesi

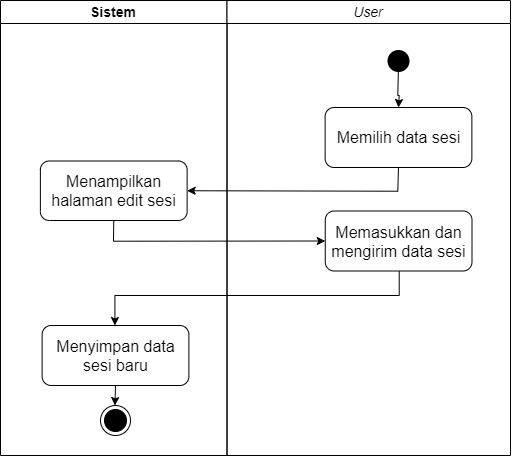
Pada Gambar 3.21 merupakan *activity* diagram yang menjelaskan aktivitas proses menambah sesi.



Gambar 3. 21 *Activity* Diagram Menambah Sesi

1. *Activity* Diagram Mengubah Sesi

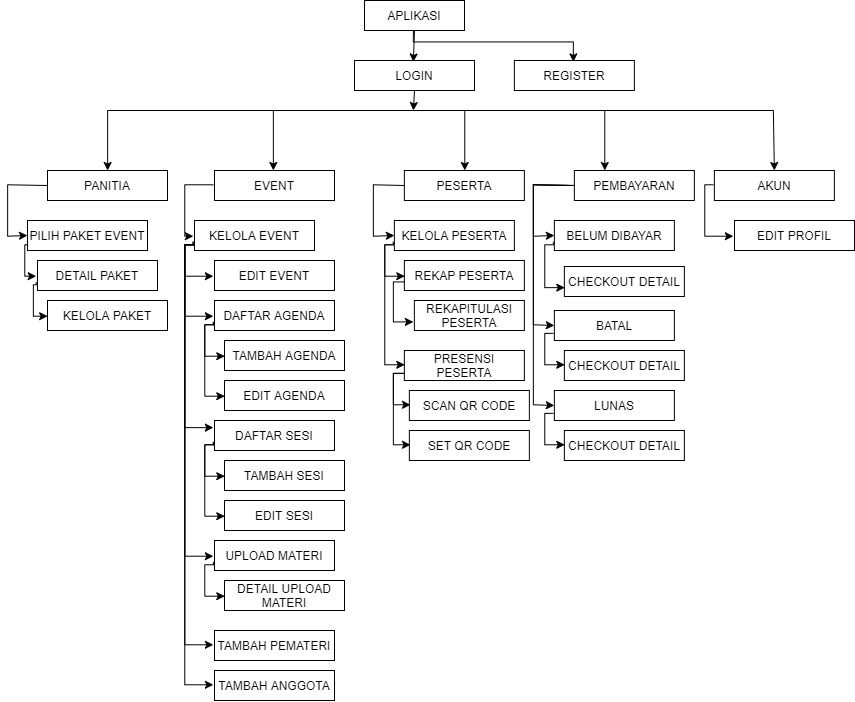
Pada Gambar 3.22 merupakan *activity* diagram yang menjelaskan aktivitas proses mengubah sesi.



Gambar 3. 22 *Activity* Diagram Mengubah Sesi

### Struktur Menu

Pada Gambar 3.23 merupakan struktur menu yang terdapat pada Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android.



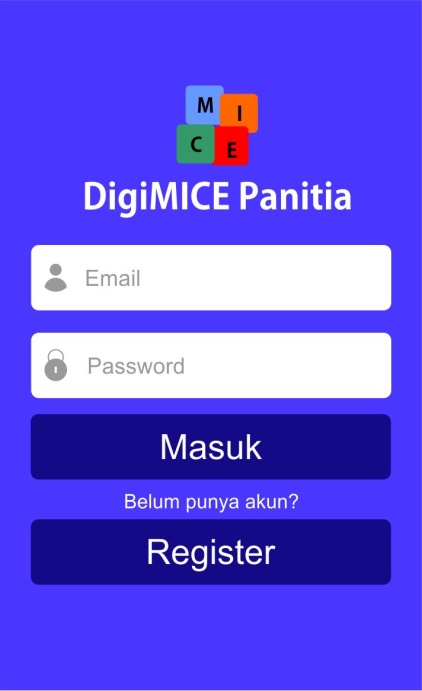
Gambar 3. 23 Struktur Menu

### *Mockup* Aplikasi

*Mockup* aplikasi merupakan rancangan *layout* atau tampilan aplikasi yang dibuat pada Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android, penjelasan rancangan *layout* aplikasi sebagai berikut :

1. Halaman *Login* dalam *Login*

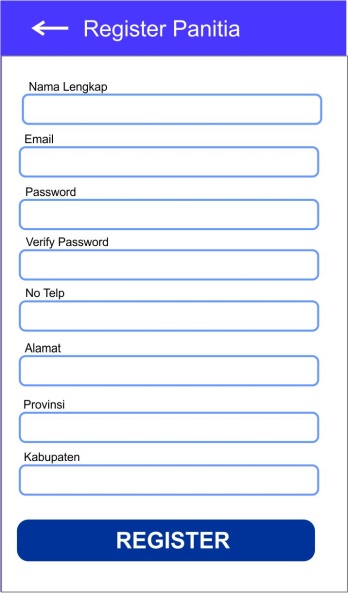
Pada Gambar 3.24 merupakan tampilan halaman *login* yang berfungsi untuk masuk dalam aplikasi setelah pengguna memasukkan *email* dan *password* yang sudah terdaftar pada sistem.



Gambar 3. 24 Halaman *Login*

1. Halaman *Register* dalam *Register*

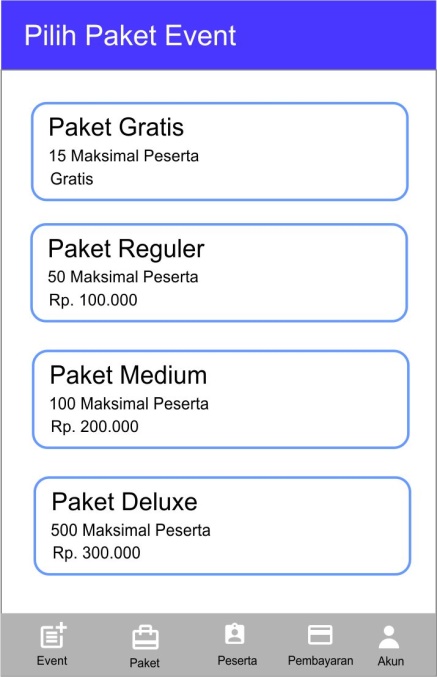
Pada Gambar 3.25 merupakan tampilan halaman *register* yang berfungsi untuk menambahkan akun baru panitia.



Gambar 3. 25 Halaman *Register*

1. Halaman Pilih Paket *Event* dalam Membeli Paket *Event*

Pada Gambar 3.26 merupakan tampilan halaman membeli dengan memilih paket *event* atau seminar yang akan dipilih oleh ketua panitia untuk membuat *event* baru sesuai paket yang dipilih.



Gambar 3. 26 Halaman Membeli Paket *Event*

1. Halaman Detail Paket dalam Membeli Paket *Event*

Pada Gambar 3.27 merupakan halaman detail paket setelah pengguna memilih paket seminar yang telah dipilih yang berisikan detail isi paket. Detail paket ini berfungsi untuk memberikan informasi paket kepada pengguna sebelum melakukan pembelian paket yang akan dipilih.



Gambar 3. 27 Halaman Detail Paket

1. Halaman Kelola Paket dalam Membeli Paket *Event*

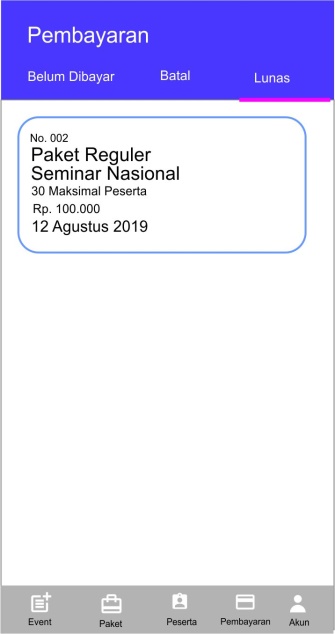
Pada Gambar 3.28 merupakan tampilan halaman kelola paket seminar setelah memilih menekan tombol kelola paket. Pada halaman ini ketua panitia menambahkan data *event* baru yang akan dibuat dan ketika menambahkan data event selesai aplikasi akan menanyakan untuk menyakinkan data sudah benar.

Gambar 3. 28 Halaman Kelola Paket

1. Halaman Pembayaran dalam Membeli Paket *Event*

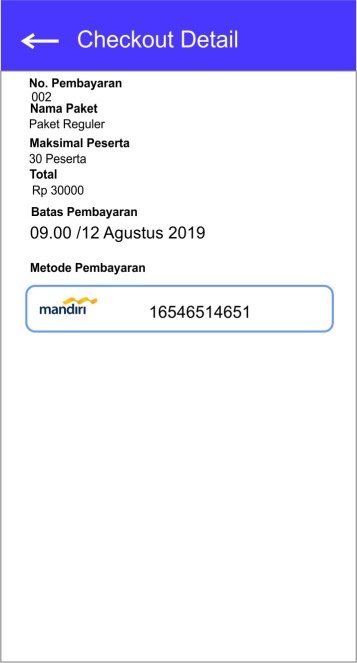
Pada Gambar 3.29 merupakan tampilan halaman pembayaran paket pada ketua panitia. Pembayaran yang masih belum dibayarkan akan masuk pada *tab* navigasi Belum Dibayar dimana untuk melakukan pembayaran dengan memilih daftar pembayaran yang belum dibayar dan akan diarahkan ke halaman *checkout detail*, pembayaran yang tidak dibayarkan sesuai batas waktu yang ditentukan akan masuk pada *tab* Batal dan pembayaran yang sudah dibayarkan akan masuk pada *tab* Lunas.

Gambar 3. 29 Halaman Pembayaran

1. Halaman *Checkout* *Detail* dalam Membeli Paket *Event*

Pada gambar 3.30 merupakan tampilan halaman *checkout* *detail* yang menampilkan rincian pembayaran yang harus dibayarkan oleh ketua panitia. Pembayaran akan dibatalkan ketika melebihi batas waktu yang sudah ditentukan oleh aplikasi.



Gambar 3. 30 Halaman *Checkout Detail*

1. Halaman Menambah Anggota Panitia dalam Menambah Anggota

Pada Gambar 3.31 merupakan tampilan halaman tambah anggota panitia dimana anggota panitia yang akan membantu ketua panitia mengelola penyelenggaraan kegiatan seminar. Ketua panitia menambahkan *email* anggota panitia yang sudah terdaftar sebagai *user* peserta atau *email* aktif yang belum terdaftar pada aplikasi dari aplikasi manajemen seminar untuk peserta.



Gambar 3. 31 Halaman Menambah Anggota Panitia

1. Halaman Mengunggah Materi dalam Mengunggah Materi

Pada Gambar 3.32 merupakan tampilan halaman *upload* materi kegiatan seminar, halaman ini berisi daftar agenda kegiatan seminar yang akan ditambahkan materi di dalamnya.



Gambar 3. 32 Halaman Mengunggah Materi

1. Halaman Detail *Upload* Materi dalam Mengunggah Agenda

Pada Gambar 3.33 merupakan halaman detail upload materi agenda yang berisi daftar materi yang sudah ditambahkan. Pada halaman ini juga dapat menambahkan materi dengan menekan tombol “+”.



Gambar 3. 33 Halaman Detail *Upload* Materi

1. Halaman *Event* dalam Melihat Daftar *Event*

Pada Gambar 3.34 merupakan halaman *event* dimana berisi daftar *event* yang sudah ditambahkan oleh ketua panitia. Halaman ini berfungsi untuk mengelola data *event* seperti mengubah data *event*, melihat daftar agenda, menambah agenda dan mengubah agenda *event*.



Gambar 3. 34 Halaman Kelola *Event*

1. Halaman *Edit* *Event* dalam Mengubah *Event*

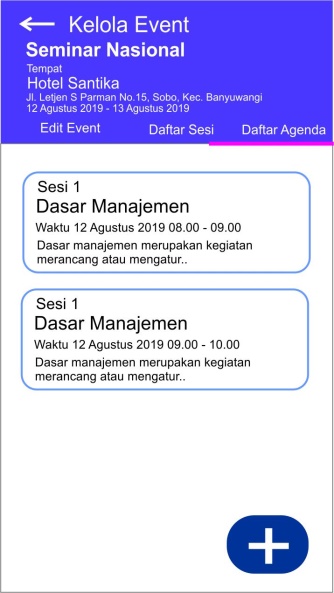
Pada Gambar 3.35 merupakan tampilah halaman edit *event*, halaman ini berfungsi untuk mengubah data *event* yang sudah ditambahkan oleh ketua panitia dikarenakan terjadi kesalahan ketika menambahkan atau memperbarui data *event*.



Gambar 3. 35 Halaman *Edit* *Event*

1. Halaman Daftar Agenda dalam Melihat Daftar Agenda

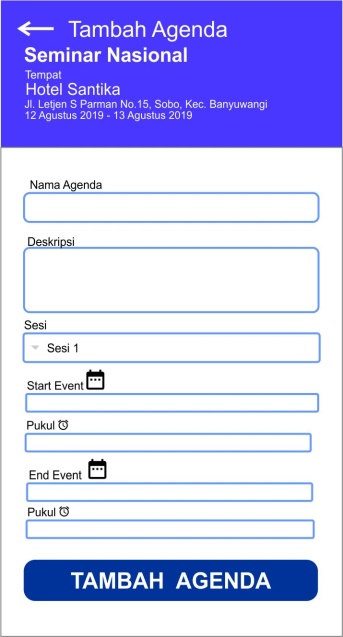
Pada Gambar 3.36 merupakan halaman daftar agenda yang menampilkan agenda-agenda yang sudah ditambahkan. Halaman ini berfungsi untuk menambah kan agenda baru dengan menekan tombol ” + “ dan untuk mengubah data agenda dengan memilih salah satu dari daftar agenda.



Gambar 3. 36 Halaman Daftar Agenda

1. Halaman Tambah Agenda dalam Menambah Agenda

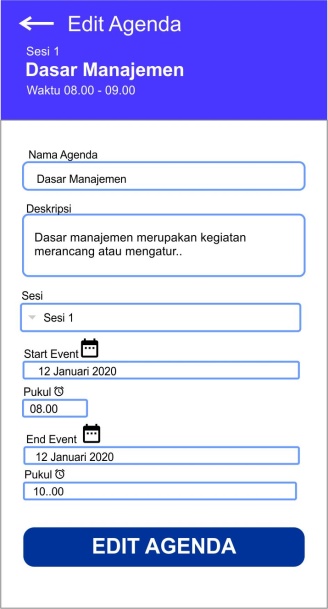
Pada Gambar 3.37 merupakan halaman tambah agenda baru yang berisikan *form* untuk menambahkan data agenda. Data yang sudah ditambah akan ditampilkan pada daftar agenda.



Gambar 3. 37 Halaman Tambah Agenda

1. Halaman Edit Agenda dalam Mengubah Agenda

Pada Gambar 3.38 merupakan halaman edit agenda, halaman ini berisi *form* yang sama dengan tambah agenda yang berfungsi untuk mengubah data agenda. Data agenda yang diubah akan ditampilkan sistem pada daftar agenda.



Gambar 3. 38 Halaman *Edit* Agenda

1. Halaman Daftar Sesi dalam Melihat Sesi

Pada Gambar 3.39 merupakan halaman yang menampilkan daftar sesi yang sudah ditambahkan. Pada halaman ini berfungsi untuk membedakan jenis absensi event berdasarkan sesi atau bebas. Perbedaannya jika jenis absensi bebas maka tombol “+” tidak akan muncul.



Gambar 3. 39 Halaman Daftar Sesi

1. Halaman Tambah Sesi dalam Menambah Sesi

Pada Gambar 3.40 merupakan halaman yang menampilkan *form* untuk menambahkan data sesi yang baru. Data sesi yang baru akan diisi untuk agenda kegiatan seminar yang akan diselenggarakan.



Gambar 3. 40 Halaman Tambah Sesi

1. Halaman Edit Sesi dalam Mengubah Sesi

Pada Gambar 3.41 merupakan halaman yang menampilkan halaman *form* edit sesi untuk mengubah data sesi jika terjadi kesalahan.



Gambar 3. 41 Halaman Edit Sesi

1. Halaman Peserta dalam Melihat Daftar Peserta

Pada Gambar 3.42 merupakan halaman yang menampilkan daftar peserta berdasarkan pada setiap *event*. Halaman ini berfungsi untuk melihat peserta yang sudah terdaftar pada setiap *event* dan rekapitulasinya.



Gambar 3. 42 Halaman Daftar Melihat Daftar Pesertar

1. Halaman Rekap Peserta dalam Melihat Daftar Peserta

Pada Gambar 3.43 merupakan halaman yang menampilkan daftar sesi pada kegiatan seminar. Daftar sesi tersebut di dalamnya berisikan rekapitulasi dari peserta seminar.



Gambar 3. 43 Halaman Rekap Peserta

1. Halaman Rekapitulasi Peserta dalam Melihat Daftar Peserta

Pada Gambar 3.44 merupakan halaman yang menampilkan rekapitulasi data peserta pada kegiatan seminar dan daftar peserta yang sudah terdaftar. Halaman ini berfungsi untuk melihat rekapitulasi data peserta yang hadir atau belum hadir dan lunas atau belum lunas serta melihat daftar peserta yang sudah terdaftar.



Gambar 3. 44 Halaman Rekapitulasi Peserta

1. Halaman Presensi Peserta pada Kelola Peserta dalam Mengabsensi Peserta

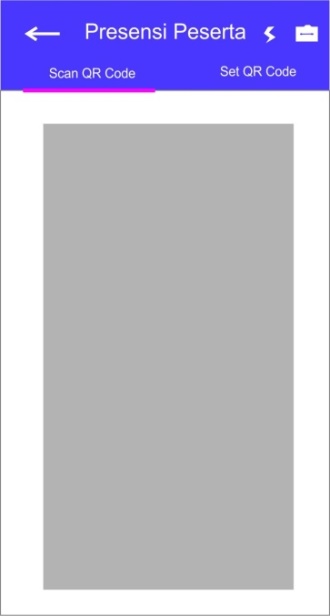
Pada Gambar 3.45 merupakan halaman yang menampilkan daftar presensi peserta dari tiap *event* dalam bentuk daftar yang digunakan untuk menuju halaman presensi peserta. Halaman ini berfungsi untuk melakukan absensi peserta seminar yang sudah mendaftar pada saat kegiatan seminar diselenggarakan.



Gambar 3. 45 Halaman Kelola Peserta

1. Halaman *Scan QR Code* pada Presensi Peserta dalam Mengabsensi Peserta Seminar

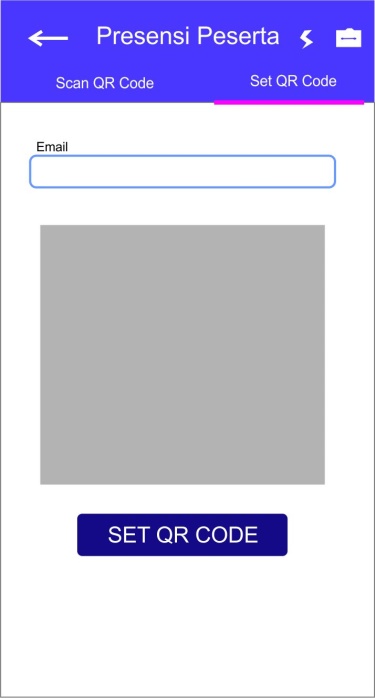
Pada Gambar 3.46 merupakan halaman untuk melakukan presensi peserta seminar, proses absensi seminar dilakukan setiap sesi kegiatan seminar yang berlangsung dengan membaca *QR Code* dari peserta seminar.



Gambar 3. 46 Halaman Presensi Peserta

1. Halaman *Set* *QR Code* dalam Mengabsensi Peserta Seminar

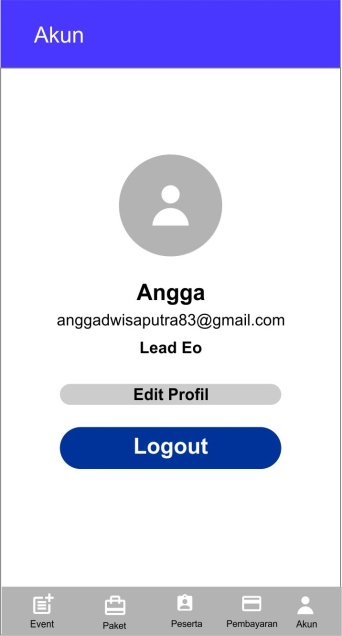
Pada Gambar 3.47 merupakan halaman proses mengabsensi dengan membaca *QR Code* peserta seminar dari kartu peserta untuk pertama kali pada *tab set QR Code*. Peserta yang menggunakan kartu peserta ketika absensi, kartu peserta harus di atur terlebih dahulu dengan membaca *QR Code* kartu dan menambahkan alamat *email* peserta yang digunakan untuk mendaftar agar dapat digunakan untuk absensi di sesi selanjutnya. Setelah membaca *QR Code* peserta aplikasi akan memberikan tampilan berhasil yang menandakan proses absensi peserta berhasil.



Gambar 3. 47 Halaman *Scan QR Code* *Smartpone*

1. Halaman Akun dalam Proses *Logout*

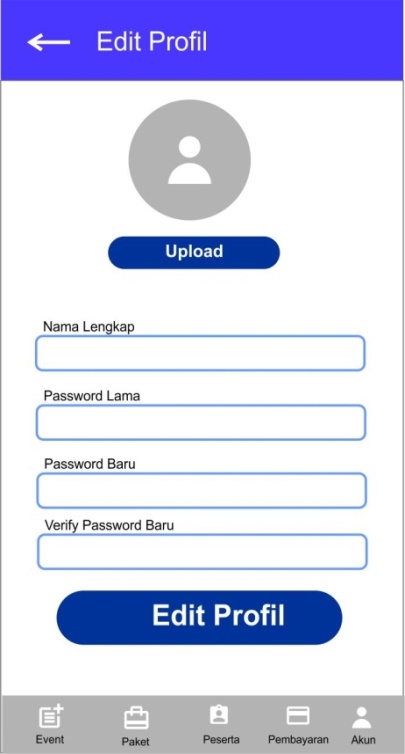
Pada Gambar 3.48 merupakan tampilan halaman akun pada aplikasi, didalam halaman ini terdapat informasi akun yang berisikan nama pengguna, email, status kepanitiaan dan nama kelompok atau organisasi. Tombol *logout* merupakan tombol yang berfungsi untuk pengguna keluar dari aplikasi jika sudah tidak menggunakan aplikasi. Untuk menggunakan kembali aplikasi, pengguna dapat melakukan *login* untuk masuk dalam aplikasi.



Gambar 3. 48 Halaman *Logout*

1. Halaman *Edit* Profil dalam mengubah data profil

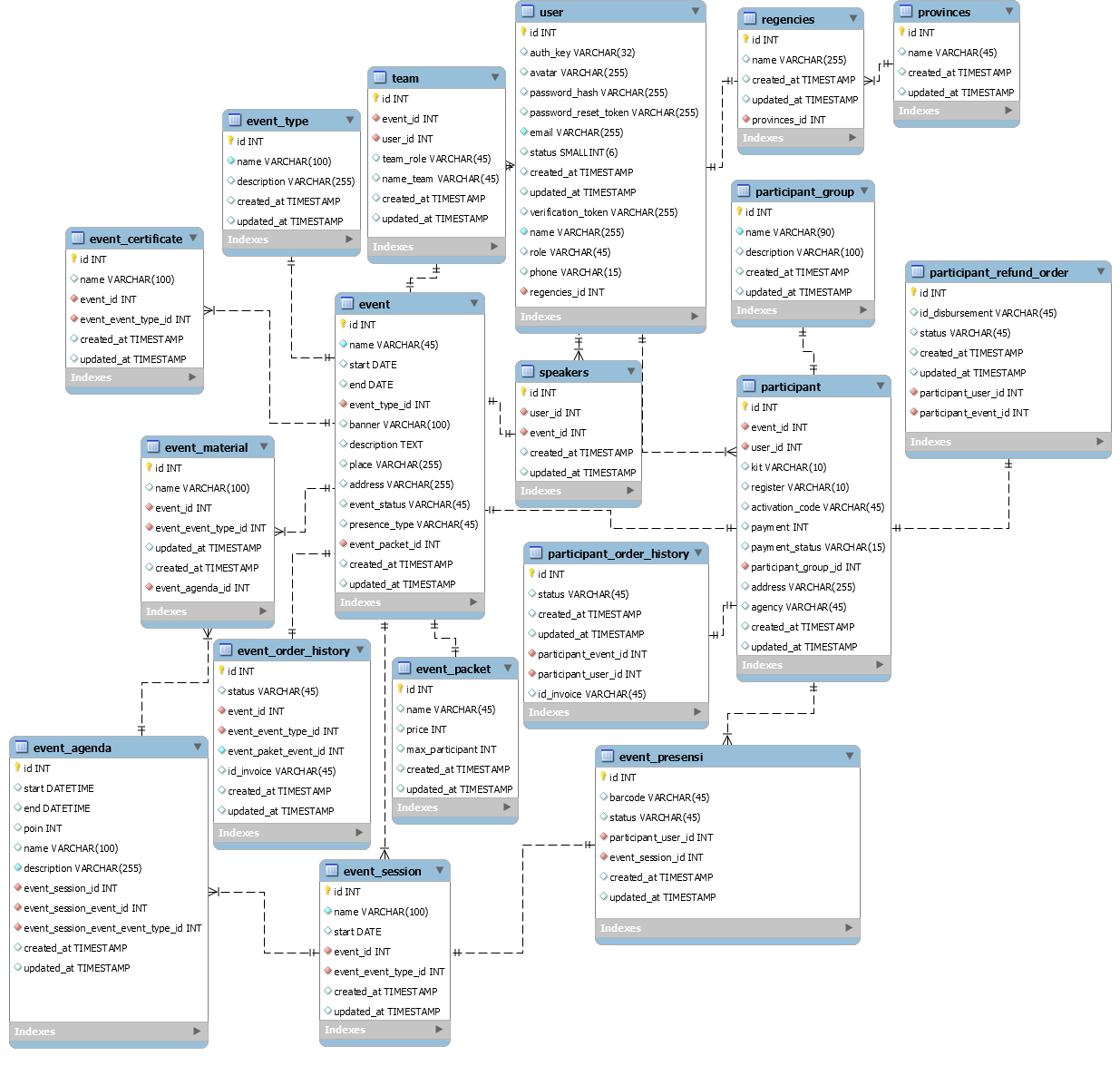
Pada Gambar 3.49 merupakan halaman *edit* profil yang berfungsi untuk melakukan pembaruan data profil *user.* Pada halaman ini dapat memperbarui data profil *user* seperti foto profil, nama lengkap profil dan memperbarui *password* lama dengan *password* yang baru.



Gambar 3. 49 Halaman *Edit* Profil

### *Entity Relational Diagram*

Pada Gambar 3.50 merupakan relasi antar tabel *database* dari aplikasi *digiMICE* berbasis Web yang diterapkan pada web *API* yang digunakan untuk mengelola data pada Aplikasi Manajemen Seminar (*digiMICE)* untuk Panitia berbasis Android.



Gambar 3. 50 *Entity Relational Diagram*

Pada Gambar 3.50 terdapat tabel-tabel yang saling berelasi satu sama lain, pertama tabel *event\_type* berelasi dengan tabel *event* untuk menyimpan data tipe *event*, tabel *team* berelasi dengan tabel *user* dan *event* untuk menyimpan data panitia, tabel *user* berelasi dengan tabel *team*, tabel *regencies* dan tabel *participant* untuk menyimpan data pengguna aplikasi, tabel *regencies* berelasi dengan tabel *user* untuk menyimpan data kabupaten, tabel *provinces* berelasi dengan tabel *regencies* untuk menyimpan data provinsi, tabel *event\_certificate* berelasi dengan tabel *event* dan tabel agenda untuk menyimpan data sertifikat, tabel *event* *material* berelasi dengan tabel *event* dan tabel agenda untuk menyimpan berkas yang diperlukan ketika seminar, tabel *participant\_group* berelasi dengan tabel *participant* untuk menyimpan jenis peserta, tabel *participant\_order\_history* berelasi dengan tabel *participant* untuk menyimpan hasil pembelian tiket, tabel *event* presensi berelasi dengan tabel *participant* untuk presensi peserta, tabel event agenda berelasi dengan tabel *event*\_*session* untuk menyimpan data agenda, tabel *event*\_*session* berelasi dengan tabel *event*, tabel *event\_presensi* dan tabel *event\_agenda* untuk menyimpan data sesi, tabel *even\_packet* berelasi dengan tabel *event* untuk menyimpan paket *event*, tabel *event\_order\_history* berelasi dengan tabel *event* untuk menyimpan hasil pembelian paket panitia, tabel *event* berelasi dengan tabel *team*, tabel *participant*, tabel *event\_session*, tabel *event\_packet*, tabel *event\_order\_history*, tabel *event\_material*, tabel *event\_certificate* dan tabel *event\_type* untuk menyimpan data *event*, tabel *speakers* berelasi dengan tabel *event* dan tabel *user* untuk menyimpan data pemateri dan tabel *participant* berelasi dengan tabel *user*, tabel *event*, tabel *event\_presensi*, tabel *participant\_order\_history*, tabel *participant\_refund\_order* berelasi dengan tabel *participant* untuk menyimpan data pengembalian tiket event, dan tabel *participant\_gruop* untuk menyimpan data peserta yang daftar seminar. Penjelasan tabel terdapat pada Tabel 3.21, Tabel 3.22, Tabel 3.23, Tabel 3.24, Tabel 3.25, Tabel 3.26, Tabel 3.27, Tabel 3.28, Tabel 3.29, Tabel 3.30, Tabel 3.31, Tabel 3.32, Tabel 3.33, Tabel 3.34, Tabel 3.35, Tabel 3.36, Tabel 3.37 dan Tabel 3.38.

Tabel 3. 21 Tabel *event\_type*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | varchar(50) | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | varchar(100) | *Not Null* |
| 3 | description | varchar(255) | *Not Null* |
| 4 | created\_at | timestamp | *Null* |
| 5 | updated\_at | timestamp | *Null* |

Tabel 3. 22 Tabel *team*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | user\_id | int | *Foreight Key , Not Null* |
| 2 | team\_role | varchar(45) | *Not Null* |
| 3 | name\_team | varchar(45) | *Null* |
| 4 | event\_id | Int | *Foreight Key , Not Null* |
| 4 | created\_at | timestamp | *Null* |
| 5 | updated\_at | timestamp | *Null* |

Tabel 3. 23 Tabel *user*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | Int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | auth\_key | varchar(32) | *Null* |
| 3 | password\_hash | varchar(255) | *Not Null* |
| 4 | password\_reset\_token | varchar(255) | *Null* |
| 5 | email | varchar(255) | *Not Null* |
| 6 | status | smallint(6) | *Not Null* |
| 7 | verification\_token | varchar(255) | *Null* |
| 8 | name | varchar(255) | *Not Null* |
| 9 | role | varchar(45) | *Not Null* |
| 10 | phone | varchar(15) | *Not Null* |
| 11 | regencies\_id | int | *Foreight Key* |
| 12 | create\_at | timestamp | *Null* |
| 13 | updated\_at | timestamp | *Null* |

Tabel 3. 24 Tabel *regencies*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Atribut** | | **Tipe Data** | | **Keterangan** | |
| 1 | id | | int | | *Primary Key , Not Null, Auto\_Increment* | |
| 2 | name | | varchar(255) | | *Not Null* | |
| 4 | created\_at | timestamp | | *Null* | |
| 5 | updated\_at | timestamp | | *Null* | |
| 6 | province\_id | int | | *Not Null, Foreight Key* | |

Tabel 3. 25 Tabel *participant\_group*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Atribut** | | **Tipe Data** | | **Keterangan** | |
| 1 | id | | int | | *Primary Key , Not Null, Auto\_Increment* | |
| 2 | name | | varchar(255) | | *Not Null* | |
| 3 | description | | varchar(100) | | *Not Null* | |
| 4 | created\_at | timestamp | | *Null* | |
| 5 | updated\_at | timestamp | | *Null* | |

Tabel 3. 26 Tabel *participant\_order\_history*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | status | varchar(45) | *Not Null* |
| 3 | created\_at | timestamp | *Null* |
| 4 | updated\_at | timestamp | *Null* |
| 5 | participant\_event\_id | int | *Not Null, Foreight Key* |
| 6 | partcipan\_user\_id | int | *Not Null, Foreight Key* |
| 7 | id\_invoice | varchar(45) | *not null* |

Tabel 3. 27 Tabel *event\_presensi*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | barcode | varchar(45) | *Null* |
| 2 | Status | varchar(45) | *Not Null* |
| 3 | participant\_user\_id | int | *Not Null, Foreight Key* |
| 4 | event\_session\_id | int | *Not Null, Foreight Key* |
| 5 | created\_at | timestamp | *Null* |
| 6 | updated\_at | timestamp | *Null* |

Tabel 3. 28 Tabel *event\_agenda*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | start | varchar(45) | *Not Null* |
| 3 | end | int | *Not Null* |
| 4 | point | int | *Not Null* |
| 5 | name | varchar(100) | *Not Null* |
| 6 | description | varchar(255) | *Not Null* |
| 7 | event\_session\_id | int | *Not Null, Foreight Key* |
| 8 | event\_session\_event\_id | int | *Not Null, Foreight Key* |
| 9 | event\_session\_event\_type\_id | int | *Not Null, Foreight Key* |
| 10 | event\_material\_id | int | *Not Null, Foreight Key* |
| 11 | created\_at | timestamp | *Null* |
| 12 | updated\_at | timestamp | *Null* |

Tabel 3. 29 Tabel *event\_session*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | varchar(45) | *Not Null* |
| 3 | start | date | *Not Null* |
| 4 | event\_id | int | *Not Null, Foreight Key* |
| 5 | event\_event\_type\_id | int | *Not Null, Foreight Key* |
| 6 | created\_at | timestamp | *Null* |
| 7 | updated\_at | timestamp | *Null* |

Tabel 3. 30 Tabel *event\_packet*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | varchar(45) | *Not Null* |
| 3 | price | int | *Not Null* |
| 4 | max\_participant | int | *Not Null* |
| 5 | created\_at | timestamp | *Null* |
| 6 | updated\_at | timestamp | *Null* |

Tabel 3. 31 Tabel *event\_order\_history*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | status | varchar(45) | *Not Null* |
| 3 | created\_at | timestamp | *Null* |
| 4 | updated\_at | timestamp | *Null* |
| 5 | event\_id | int | *Not Null, Foreight Key* |
| 6 | event\_event\_type\_id | int | *Not Null, Foreight Key* |
| 7 | event\_paket\_event\_id | int | *Not Null, Foreight Key* |

Tabel 3. 32 Tabel *event\_material*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1. | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2. | name | varchar(100) | *Not Null* |
| 3 | event\_id | int | *Not Null, Foreight Key* |
| 4 | event\_event\_type\_id | int | *Foreight Key, not Null* |
| 5 | created\_at | timestamp | *Null* |
| 6 | updated\_at | timestamp | *Null* |

Tabel 3. 33 Tabel *regencies*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | int | *Not Null* |
| 3 | created\_at | timestamp | *Null* |
| 4 | updated\_at | timestamp | *Null* |
| 5 | provinces\_id | int | *Not Null, Foreight Key* |

Tabel 3. 34 Tabel *event*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | varchar(100) | *Not Null* |
| 3 | start | date | *Not Null* |
| 4 | end | date | *Not Null* |
| 5 | banner | varchar(100) | *Null* |
| 6 | description | text | *Not Null* |
| 7 | place | varchar(255) | *Not Null* |
| 8 | address | varchar (255) | *Not Null* |
| 9 | event\_status | varchar(45) | *Not Null* |
| 10 | event\_type\_id | int | *Not Null, Foreight Key* |
| 11 | event\_packet\_id | int | *Not Null, Foreight Key* |
| 12 | created\_at | timestamp | *Null* |
| 13 | updated\_at | timestamp | *Null* |

Tabel 3. 35 Tabel *participant*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | event\_id | int | *Not Null, Foreight Key* |
| 2 | user\_id | int | *Not Null, Foreight Key* |
| 3 | kit | varchar(10) | *Not Null* |
| 4 | register | varchar(10) | *Not Null* |
| 5 | activation\_code | varchar(45) | *Null* |
| 6 | payment | int | *Not Null* |
| 7 | payment\_status | varchar(15) | *Not Null* |
| 8 | participant\_group\_id | int | *Not Null, Foreight Key* |
| 9 | created\_at | timestamp | *Null* |
| 10 | updated\_at | timestamp | *Null* |
| 11 | address | varchar(255) | *Not Null* |
| 12 | agency | varchar(45) | *Null* |

Tabel 3. 36 Tabel *provinces*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Primary Key , Not Null, Auto\_Increment* |
| 2 | name | int | *Not Null* |
| 3 | created\_at | timestamp | *Null* |
| 4 | updated\_at | timestamp | *Null* |

Tabel 3. 37 Tabel *speakers*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | user\_id | int | *Not Null, Foreight Key* |
| 2 | event\_id | int | *Not Null, Foreight Key* |
| 3 | created\_at | timestamp | *Null* |
| 4 | updated\_at | timestamp | *Null* |

Tabel 3. 38 Tabel *participant\_refund\_order*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Data** | **Keterangan** |
| 1 | id | int | *Not Null, Foreight Key* |
| 2 | id\_disbursement | varchar(50) | *Not Null, Foreight Key* |
| 3 | status | varchar(25) | *Not Null* |
| 4 | participant\_user\_id | int | *Not Null* |
| 5 | participant\_event\_id | int | *Not Null* |
| 6 | created\_at | timestamp | *Null* |
| 7 | updated\_at | timestamp | *Null* |

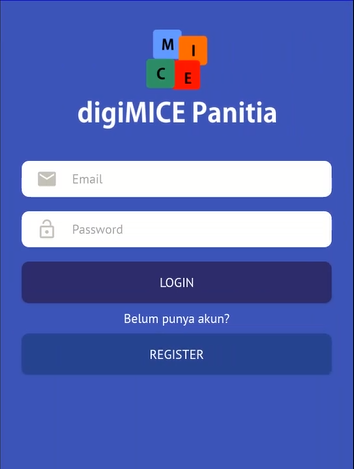
# BAB 4 HASIL DAN PEMBAHASAN

## Hasil

Berikut merupakan penjelasan hasil Aplikasi Manajemen Seminar *(digiMICE)* untuk Panitia berbasis Android yang telah dibuat.

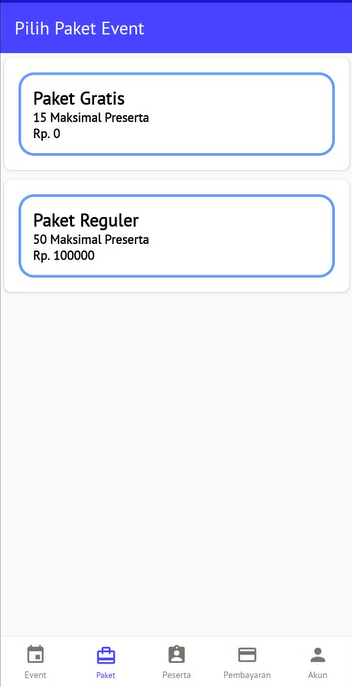
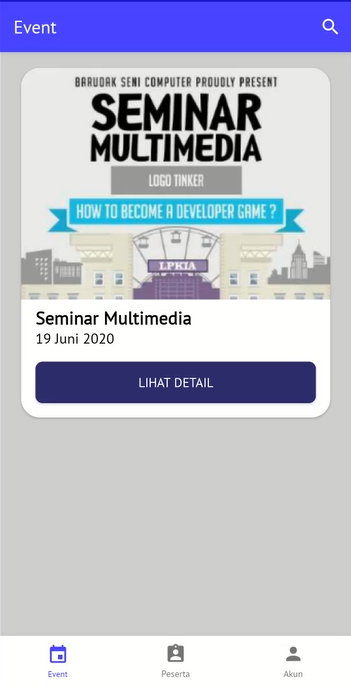
### *Login* Aplikasi

Pada tampilan halaman *login* merupakan tampilan awal ketika aplikasi pertama kali dibuka oleh panitia untuk masuk pada aplikasi sebagai ketua atau anggota panitia dengan memasukkan *email* dan *password* yang sudah terdaftar. Halaman *login* ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Tampilan Halaman *Login*

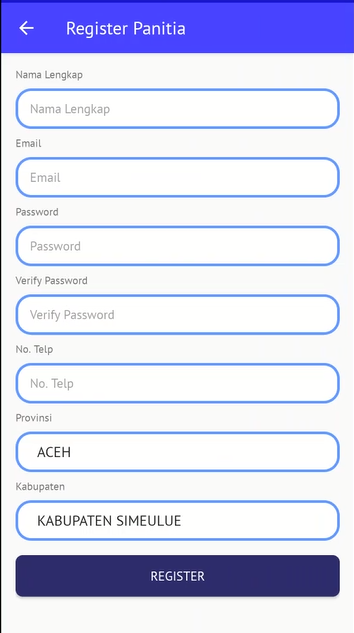
Setelah berhasil *login* panitia akan menuju halaman utama aplikasi berdasarkan jenis hak akses *user* sebagai ketua panitia atau anggota panitia. Perbedaan dari masing-masing halaman utama hak akses *user* adalah pada halaman utama ketua panitia terdapat menu *Event*, Paket, Peserta, Pembayaran dan Akun sedangkan anggota panitia terdapat menu *Event*, Peserta dan Akun. Tampilan halaman utama ketua dan anggota panitia terdapat pada Gambar 4.2.

**** 

Gambar 4. 2 Halaman Utama Ketua dan Anggota Panitia

### *Register* Panitia

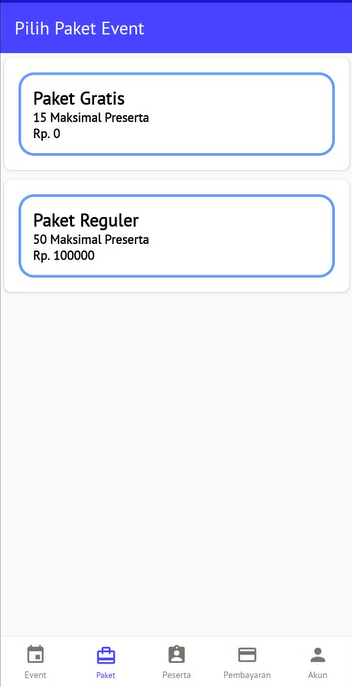
Ketika panitia belum memiliki akun untuk masuk pada aplikasi, panitia dapat melakukan registrasi untuk membuat akun baru dengan menekan tombol *register* pada halaman login. Halaman register terdapat pada Gambar 4.3, pada halaman ini panitia harus mengisi beberapa *field* untuk mengisikan data panitia untuk keperluan registrasi panitia.



Gambar 4. 3 Tampilan Halaman *Register*

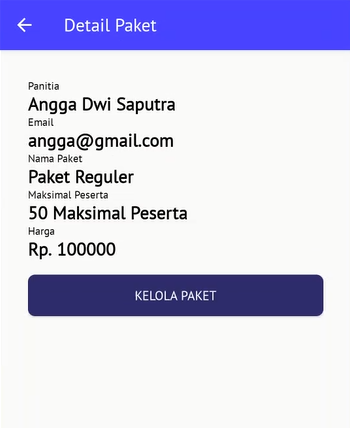
### Membeli Paket *Event*

Setelah melakukan *login* sebagai ketua panitia akan diarahkan ke halaman utama. Halaman yang pertama kali ditampilkan adalah halaman pilih paket *event* yang terdapat pada Gambar 4.4. Halaman ini berfungsi untuk membeli paket *event* yang akan diselenggarakan.

****

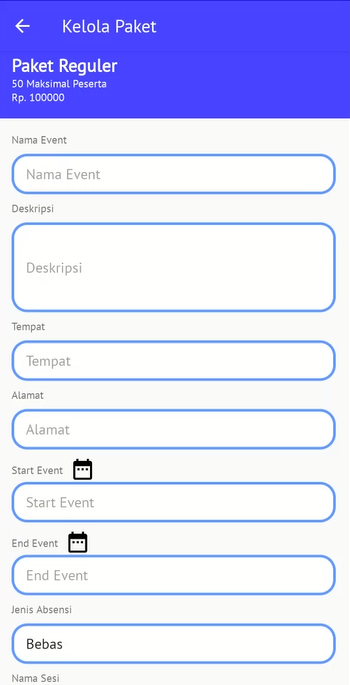
Gambar 4. 4 Tampilan Halaman Pilih Paket

Ketika paket *event* dipilih akan diarahkan ke halaman detail paket yang terdapat pada Gambar 4.5. Halaman ini menampilkan rincian paket yang akan dibeli oleh panitia penyelenggara seminar.



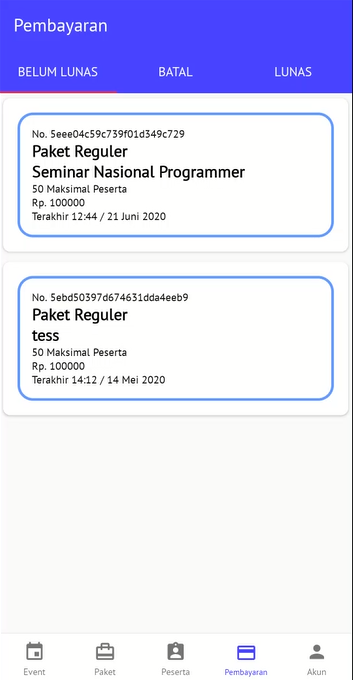
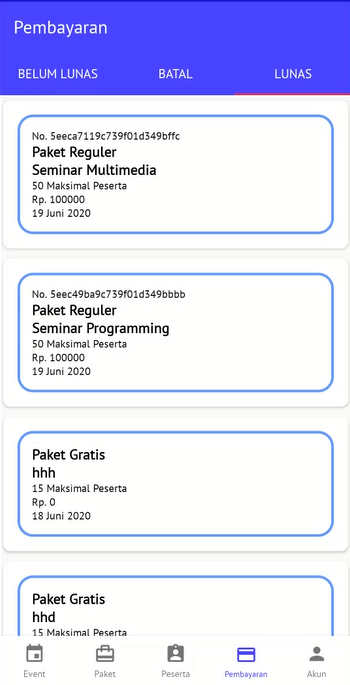
Gambar 4. 5 Tampilan Halaman Detail Paket

Lalu pada halaman kelola paket yang tampil ketika menekan tombol kelola paket pada halaman detail. Halaman kelola paket terdapat pada Gambar 4.6 yang berfungsi untuk menambahkan data-data *event* yang akan diselenggarakan oleh panitia.

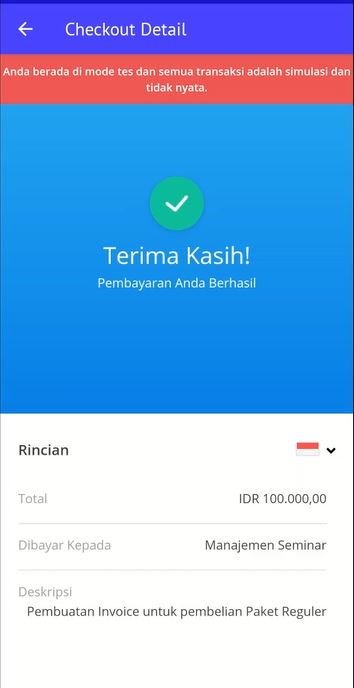
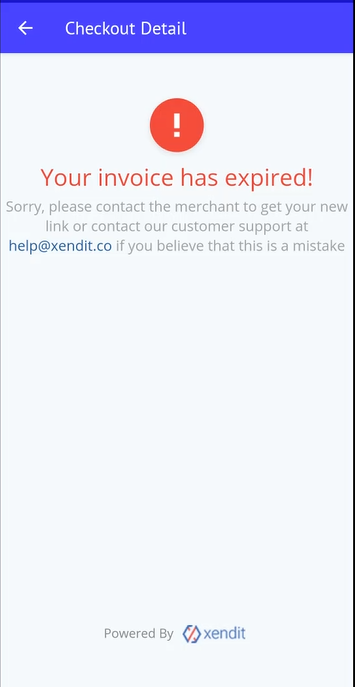
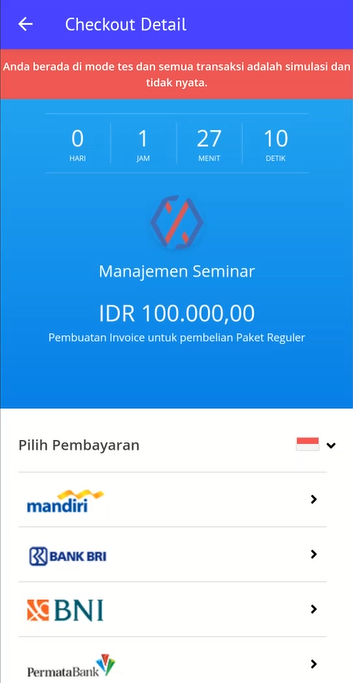
Gambar 4. 6 Tampilan Halaman Kelola Paket

Setelah mengisi data-data *event* yang diselenggarakan panitia akan diarahkan ke halaman utama ketua panitia kemudia panitia memilih navigasi pembayaran, dimana panitia harus segera membayarkan paket yang sudah dibeli. Halaman pembayaran pada Gambar 4.7 terdapat 3 *tab* menu yaitu Belum Lunas, Lunas, dan Batal yang masing-masing berfungsi untuk menampilkan data pembayaran sesuai dengan kategori *tab* menu tersebut.

Gambar 4. 7 Tampilan Halaman Pembayaran

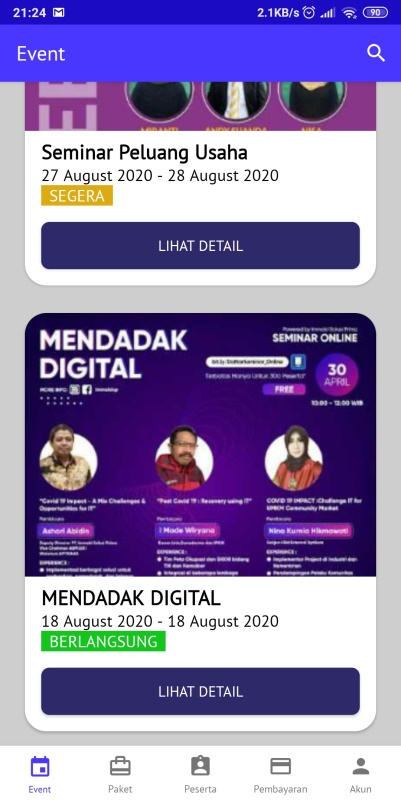
Kemudia halaman *checkout detail* yang terdapat pada Gambar 4.8 merupakan halaman yang muncul setelah menekan salah satu daftar pembayaran yang terdapat pada masing-masing tab menu Belum Lunas, Lunas, dan Batal. Halaman ini berfungsi untuk melakukan pembayaran dan menampilkan secara detail status pembayaran.



Gambar 4. 8 Tampilan Halaman *Checkout Detail*

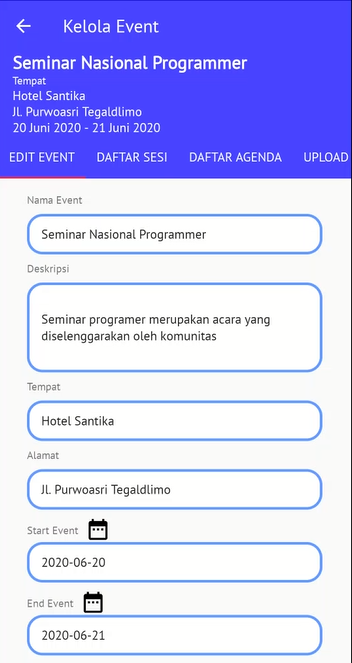
### Mengelola *Event*

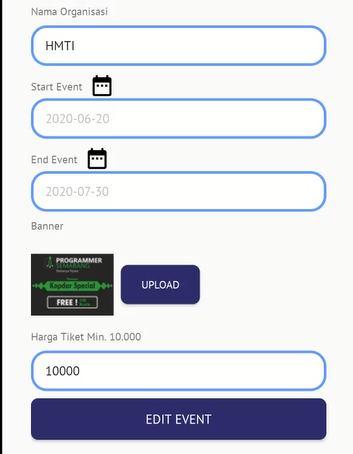
Setelah pembayaran selesai dibayarkan *event* akan muncul pada halaman event yang terdapat pada Gambar 4.9 dan siap untuk dikelola oleh panitia. Halaman ini berfungsi untuk menampilkan *event-event* yang sudah dibayarkan dan siap dikelola.



Gambar 4. 9 Tampilan Halaman Event

Untuk mengelola data *event* yang sudah dibayarkan, panitia menekan tombol lihat detail pada halaman *event* dan diarahkan ke halaman kelola *event* yang terdapat pada Gambar 4.10*.* Halaman ini memiliki beberapa *tab-tab* navigasi untuk mengelola data tentang keperluan *event* yang diselenggarakan. *Tab-tab* navigasinya seperti *Edit* *Event*, Daftar Sesi, Daftar Agenda, *Upload* Materi, Tambah Pemateri, dan Tambah Anggota Panitia, namun halaman yang ditampilkan pertama kali adalah halaman *edit* *event* pada *tab* navigasi *Edit* *Event*. Halaman *edit* *event* berfungsi untuk mengubah data *event* jika ada yang harus diganti.

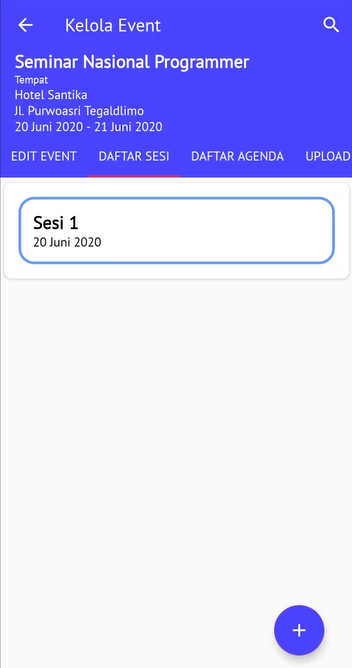
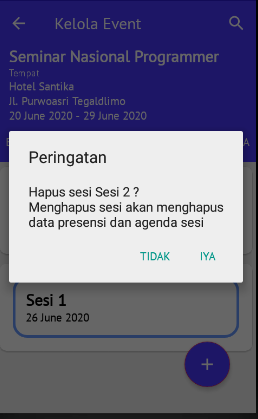




Gambar 4. 10 Tampilan Halaman Edit Event

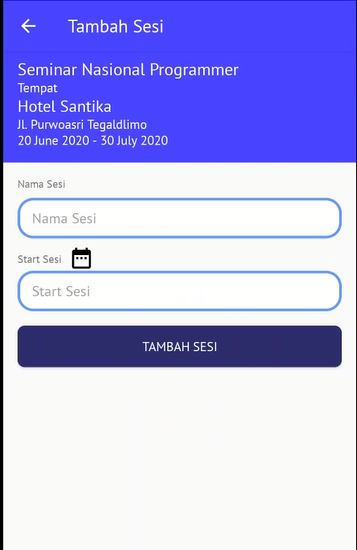
### Mengelola Daftar Sesi

Pada tab daftar sesi terdapat halaman daftar sesi yang berfungsi untuk memuat data sesi pada *event*. Data sesi ditampilkan dengan nama dan tanggal sesi yang sudah ditentukan. Halaman daftar sesi terdapat pada Gambar 4.11, di halaman ini juga dapat melakukan tindakan lanjutan pada data sesi seperti tambah, edit, dan hapus sesi. Ketika jenis presensi berdasarkan sesi tombol “+” akan muncul.

Gambar 4. 11 Tampilan Halaman Daftar Sesi

Pada Gambar 4.12 merupakan halaman tambah sesi yang muncul setelah menekan tombol “+” pada halaman daftar sesi. Halaman ini berfungsi untuk menambah sesi *event* yang akan dibuat.



Gambar 4. 12 Tampilan Halaman Tambah Sesi

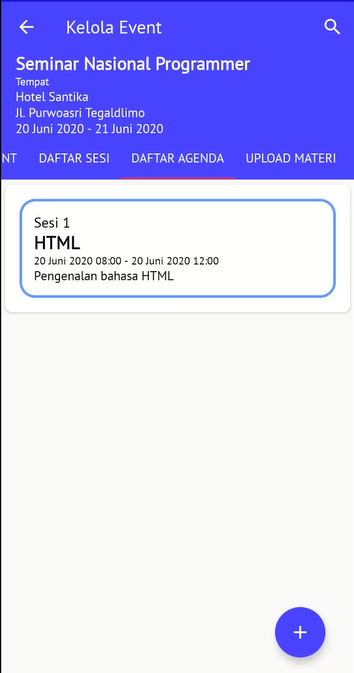
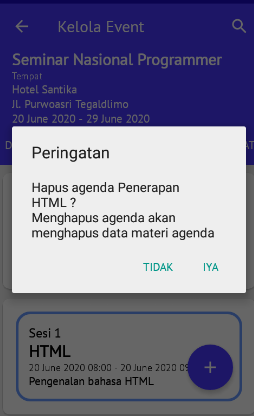
Pada Gambar 4.13 merupakan halaman edit sesi yang muncul setelah menekan salah satu dari data sesi yang ditampilkan. Halaman ini berfungsi untuk mengubah data sesi jika terdapat perubahan data.



Gambar 4. 13 Tampilan Halaman *Edit* Sesi

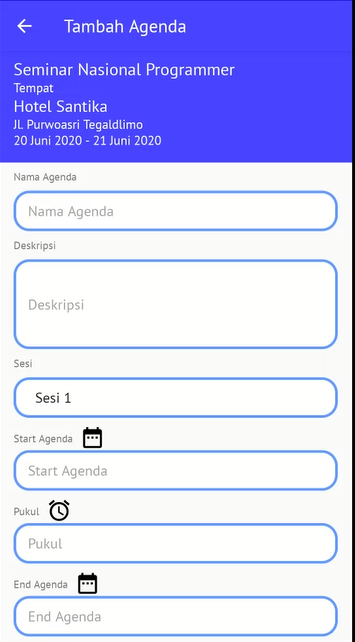
### Mengelola Agenda

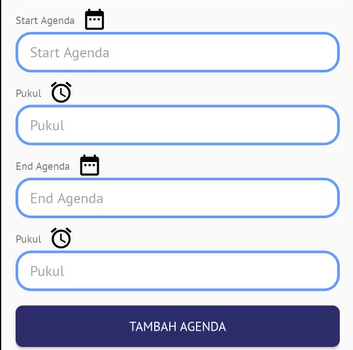
Pada tab navigasi daftar aganda terdapat halaman daftar agenda yang terdapat pada Gambar 4.14. Halaman ini berfungsi untuk menampilkan data-data agenda. Pada halaman ini juga dapat melakukan tindakan lanjutan pada data agenda seperti menambah, mengubah, dan menghapus data agenda.

Gambar 4. 14 Tampilan Halaman Daftar Agenda

Pada halaman tambah agenda yang terdapat pada Gambar 4.15 merupakan halaman untuk menambah data agenda untuk melengkapi data sesi yang sudah ditambahkan. Halaman ini muncul ketika menekan tombol “+” pada halaman daftar agenda.

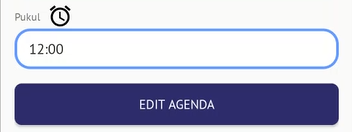




Gambar 4. 15 Tampilan Halaman Tambah Agenda

Pada halaman edit agenda yang terdapat pada Gambar 4.16 merupakan halaman yang berfungsi untuk mengubah data agenda jika terjadi perubahan. Halaman ini muncul ketika memilih salah satu dari data-data agenda yang ada di halaman daftar agenda.

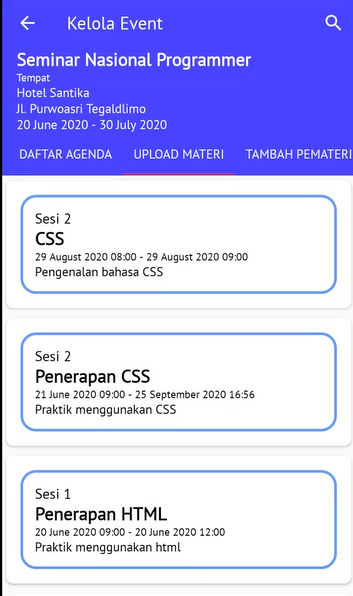




Gambar 4. 16 Tampilan Halaman Edit Agenda

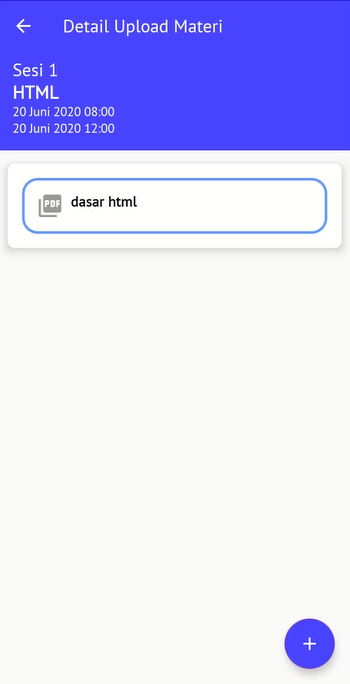
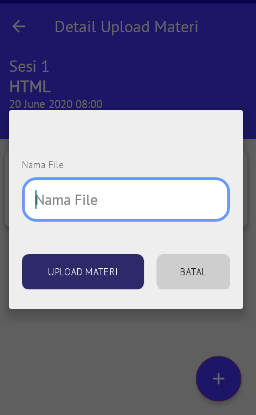
### Mengungah Materi

Pada halaman *upload* materi yang terdapat pada Gambar 4.17 merupakan halaman yang muncul di tab navigasi *upload* materi. Halaman ini berfungsi untuk menambahkan materi *event* yang akan diunduh oleh peserta. Data materi dikategorikan berdasarkan data agenda sehingga jika ingin mengunggah materi harus memilih salah satu data agenda terlebih dahulu.



Gambar 4. 17 Tampilan Halaman *Upload* Materi

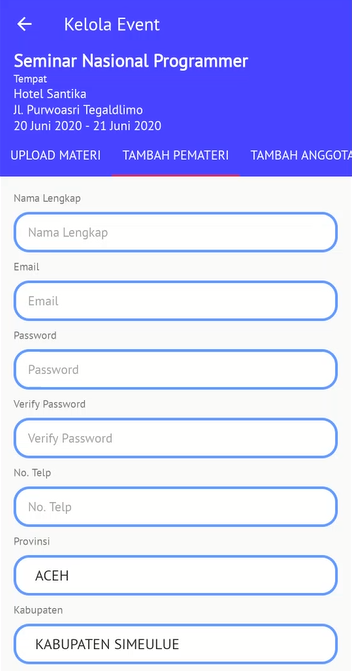
Pada halaman detail *upload* materi yang terdapat pada Gambar 4.18 merupakan halaman yang berfungsi untuk menampilkan data materi yang sudah diunggah. Pada halaman ini dapat melakukan tindakan untuk mengunggah dan menghapus data materi.

Gambar 4. 18 Tampilan Halaman Detail *Upload* Materi

### Menambah Pemateri

Pada *tab* navigasi tambah pemateri terdapat halaman tambah pemateri yang terdapat pada Gambar 4.19. Halaman ini berfungsi untuk menambahkan *user* sebagai pemateri yang akan digunakan pada aplikasi peserta, dimana *user* sebagai pemateri dapat mengunggah materi yang ada pada *event* yang diselenggarakan.

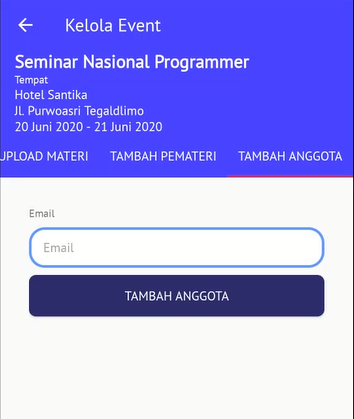




Gambar 4. 19 Tampilan Halaman Tambah Pemateri

### Menambah Anggota Panitia

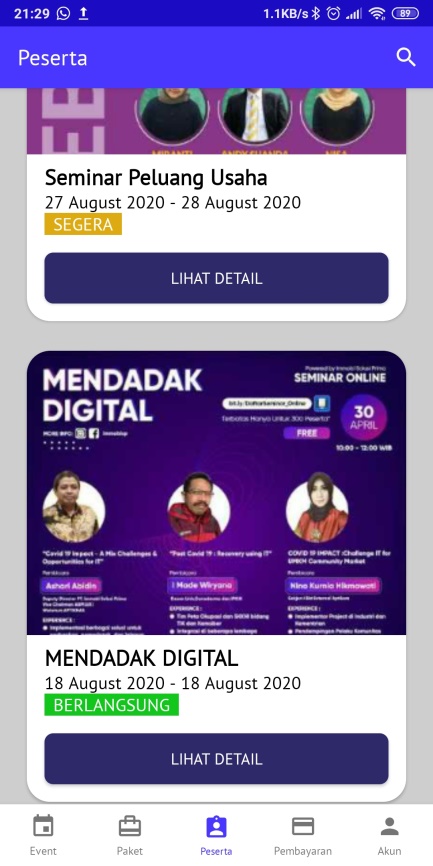
Pada *tab* navigasi tambah anggota terdapat halaman tambah anggota yang terdapat pada Gambar 4.20. Halaman ini berfungsi untuk menambahkan akses pada *user* yang mendaftar melalui aplikasi peserta untuk menjadi anggota panitia membantu ketua panitia mengelola *event* yang akan diselenggarakan dengan memasukkan alamat *email* *user* yang sudah terdaftar sebagai panitia atau alamat *email* yang aktif yang belum terdaftar pada aplikasi.



Gambar 4. 20 Tampilan Halaman Tambah Anggota

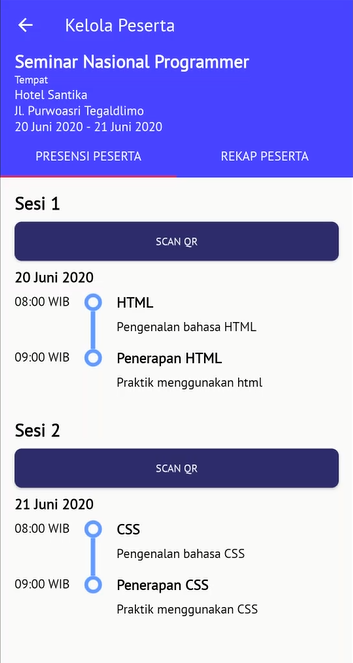
### Mengelola Peserta

Pada *menu* navigasi peserta pada Gambar 4.21 terdapat halaman yang berisikan *event-event* sama seperti halaman *event.* Halaman ini berfungsi untuk mengelola peserta berdasarkan *event* yang diselenggarakan. Pengelolaan peserta seperti absensi dan rekapilulasi peserta.



Gambar 4. 21 Tampilan Halaman Peserta

Pada Gambar 4.22 merupakan tampilan halaman kelola peserta yang memiliki 2 *tab* navigasi yaitu presensi peserta dan peserta. Halaman ini berfungsi untuk mengelola peserta seperti melakukan presensi peserta ketika *event* berlangsung dan rekapitulasi dari presensi peserta. Pada *tab* presensi peserta terdapat halaman jadwal presensi *event* peserta yang menampilkan daftar sesi beserta agendanya.



Gambar 4. 22 Tampilan Halaman Kelola Peserta

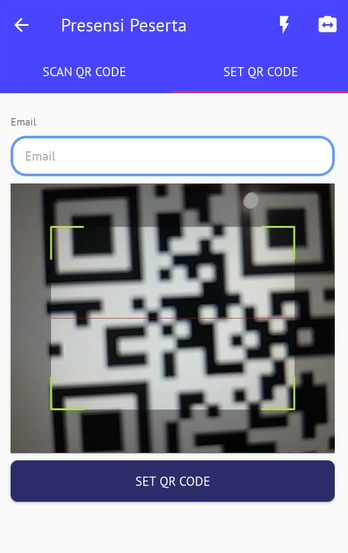
### Mengabsensi Peserta

Setelah menekan tombol *scan QR* pada halaman kelola peserta akan diarahkan ke halaman presensi peserta yang terdapat pada Gambar 4.23. Halaman ini memiliki 2 *tab* navigasi yaitu *scan QR Code* dan *set QR Code.* Pada *tab* *scan QR Code* terdapat halaman *scan QR Code* yang berfungsi untuk melakukan absensi peserta dengan memindai *QR Code* peserta yang sudah ada pada aplikasi peserta.



Gambar 4. 23 Tampilan Halaman *Scan* *QR Code*

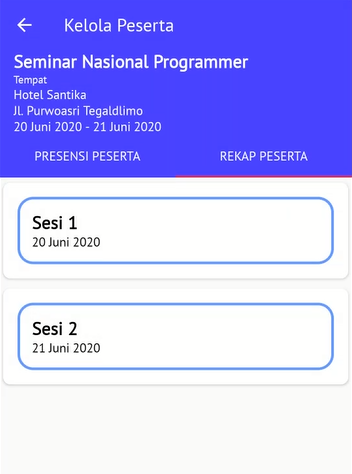
Pada *tab* navigasi *set QR Code* terdapat halaman *set QR Code* yang terdapat pada Gambar 4.24 berfungsi untuk mengatur *QR Code* peserta yang melakukan absensi melalui kartu peserta dengan menyertakan *email* peserta ketika registrasi.



Gambar 4. 24 Tampilan Halaman *Set QR Code*

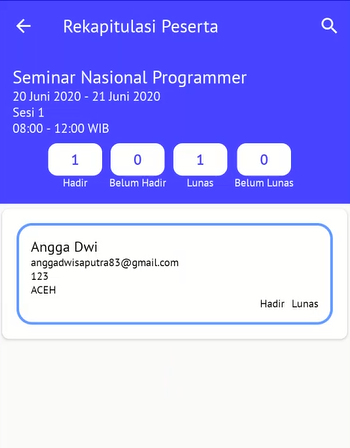
### Melihat Rekapitulasi Peserta

Pada Gambar 4.25 merupakan *tab* navigasi rekap peserta terdapat halaman rekap peserta yang berisi data-data sesi *event.* Data sesi tersebut berisi rekapitulasi peserta yang sudah direkap, hasil rekap dikategorikan berdasarkan data sesi masing-masing.



Gambar 4. 25 Tampilan Halaman Rekap Peserta

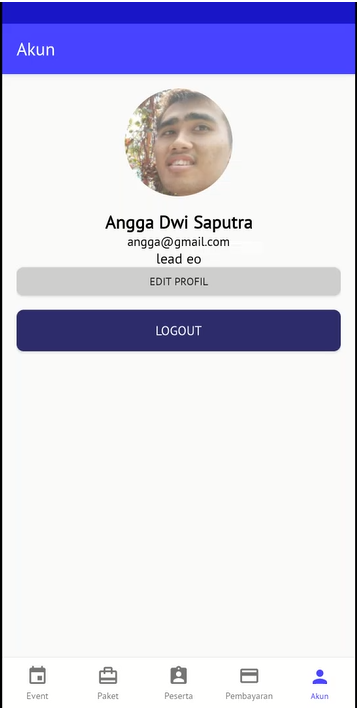
Setelah memilih salah satu data sesi dari halaman rekap peserta, ­*user* akan diarahkan ke halaman rekapitulasi peserta berdasarkan sesi yang terdapat pada Gambar 4.26. Halaman ini berfungsi untuk menampilkan hasil rekapitulasi peserta dalam kategori hadir, belum hadir, lunas dan belum lunas beserta data peserta yang sudah tergabung.



Gambar 4. 26 Tampilan Halaman Rekapitulasi Peserta

### *Logout* Aplikasi

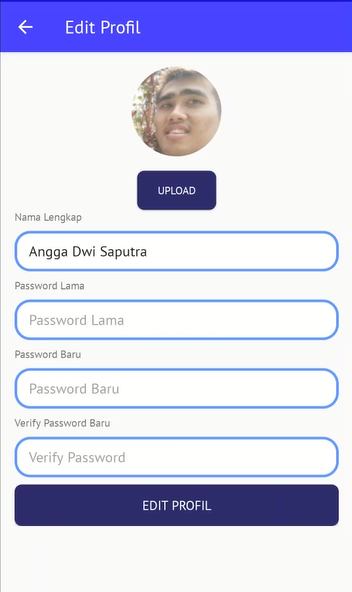
Pada menunavigasi akun terdapat halaman akun yang terdapat pada Gambar 4.27. Halaman ini berfungsi untuk menampilkan profil *user* dan tindakan lanjutan seperti *logout* untuk mengakhiri sesi *login* aplikasi dan *edit* profil untuk mengubah profil.



Gambar 4. 27 Tampilan Halaman Akun

### Mengubah Profil

Setelah menekan tombol *edit* profil akan diarahkan ke halaman *edit* profil yang terdapat pada Gambar 4.28. Halaman ini berfungsi untuk mengubah data profil *user* seperti memperbarui foto profil, nama lengkap, dan mengganti kata sandi.



Gambar 4. 28 Tampilan Halaman *Edit* Profil

## Pengujian

Untuk mengetahui cara kerja aplikasi maka diperlukan pengujian pada Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android agar semua fitur yang dibutuhkan sesuai rumusan masalah tercapai, pengujian aplikasi menggunakan pengujian *blackbox testing* yang akan dijelaskan pada Tabel 4.1 sebagai berikut.

Tabel 4. 1 Tabel *Blackbox Testing*

|  |  |  |  |
| --- | --- | --- | --- |
| *User Story* | *Test Step* | *Expected Result* | *Result* |
| Sukses *Login* | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman *login* 3. Aktor memasukkan *email* dan *password* di *form* *login* yang sudah terdaftar pada sistem 4. Aktor melakukan *login* 5. Sistem memasukkan aktor ke dalam aplikasi sesuai hak akses aktor. | 1. Aktor mendapat pesan berhasil login. 2. Aktor masuk dalam aplikasi sesuai hak akses | *Passed* |
| Gagal *Login* | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman *login* 3. Aktor memasukkan *email* dan *password* di *form* *login* yang belum terdaftar pada sistem 4. Aktor melakukan *login* | 1. Aktor mendapat pesan gagal *login*. | *Passed* |
| Sukses Membeli Paket *Event* | 1. Aktor memilih menu panitia 2. Sistem menampilkan halaman pilih paket 3. Aktor memilih paket *event.* 4. Sistem menampilkan detail paket dan tombol kelola paket untuk menambahkan *event* baru 5. Aktor menekan tombol kelola paket 6. Sistem menampilkan halaman tambah *event* berisi *form* tambah *event* baru. 7. Aktor memasukkan data *event* baru 8. Aktor menekan tombol pilih paket 9. Sistem menampilkan pertanyaan “apakah sudah yakin” dengan pilihan “ya” dan “tidak” 10. Aktor memilih “ya” 11. Sistem menampilkan halaman pembayaran. 12. Aktor memilih *tab* belum dibayar 13. Sistem menampilkan daftar pembayaran yang belum dibayarkan oleh aktor 14. Aktor memilih salah satu dari daftar pembayaran yang belum dibayar 15. Sistem menampilkan halaman *checkout* detail 16. Aktor melihat isi halaman *checkout* detail 17. Aktor melakukan pembayaran 18. Aktor menekan tombol kembali 19. Sistem menampilkan halaman pembayaran pembayaran 20. Aktor memilih *tab* batal 21. Sistem menampilkan daftar pembayaran yang dibatalkan dikarenakan tidak membayar sesuai dihalaman *checkout* detail 22. Aktor melihat daftar pembayaran yang dibatalkan 23. Aktor memilih *tab* lunas 24. Sistem menampilkan daftar pembayaran yang sudah dibayarkan aktor 25. Aktor melihat daftar pembayaran yang sudah dibayarkan | 1. Aktor mendapatkan pesan berhasil. 2. Aktor menuju kembali ke halaman utama untuk melakukan pembayaran. 3. Data paket muncul di halaman pembayaran pada *tab* belum lunas. 4. Aktormendapatkan pesan pembayaran berhasil 5. Paket berhasil ditambahkan 6. Data paket muncul pada *tab* navigasi lunas | *Passed* |
| Gagal Membeli Paket *Event* | 1. Aktor memilih menu panitia 2. Sistem menampilkan halaman pilih paket 3. Aktor memilih paket *event.* 4. Sistem menampilkan detail paket dan tombol kelola paket untuk menambahkan *event* baru 5. Aktor menekan tombol kelola paket 6. Sistem menampilkan halaman tambah *event* berisi *form* tambah *event* baru. 7. Aktor mengisikan tanggal pada *field end event* sebelum tanggal *start event* dan tanggal pada *field start sesi* sebelum tanggal *start event serta* sesudah tanggal *end event.* 8. Aktormemasukkan jumlah tiket kurang dari 10.000 9. Aktor menekan tombol pilih paket 10. Sistem menampilkan pertanyaan “apakah sudah yakin” dengan pilihan “ya” dan “tidak” 11. Aktor memilih “ya” | 1. Aktormendapat pesan pada *field end event* dan *field start sesi* tanggal harus sesudah *start event* serta tanggal harus sebelum *end event*. 2. Aktormendapatkan pesan tiket harus minimal 10.000. | *Passed* |
| Sukses Tambah Akun Anggota Panitia | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* tambah anggota 6. Sistem menampilkan halaman *form* tambah anggota 7. Aktor memasukkan dan mengirim data anggota 8. Sistem menyimpan data anggota dan menampilkan *alert* berhasil | 1. Aktormendapat pesan berhasil. 2. Akun anggota panitia berhasil dibuat. 3. Anggota panitia dapat melakukan login di aplikasi | *Passed* |
| Gagal Tambah Akun Anggota Panitia | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* tambah anggota 6. Sistem menampilkan halaman *form* tambah anggota 7. Aktor memasukkan dan mengirim data anggota yang salah dan sudah pernah dimasukkan. 8. Sistem menyimpan data anggota dan menampilkan *alert* berhasil | 1. Aktormendapatkan pesan *email* sudah terdaftar. 2. Aktormendapatkan pesan *email* belum terdaftar jika belum terdaftar. | *Passed* |
| Sukses Mengubah data *Event* | 1. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 2. Sistem menampilkan halaman kelola *event.* 3. Aktor memilih *tab* edit *event* yang berisikan *form* edit *event* 4. Sistem menampilkan halaman edit *event* 5. Aktor memasukkan dan mengirim data *event* 6. Sistem menyimpan perubahan data *event* dan menampilkan *alert* berhasil | 1. Aktormendapatkan pesan berhasil. 2. Data *event* berhasil diubah. | *Passed* |
| Gagal Mengubah data *Event* | 1. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 2. Sistem menampilkan halaman kelola *event.* 3. Aktor memilih *tab* edit *event* yang berisikan *form* edit *event* 4. Sistem menampilkan halaman edit *event* 5. Aktormengisikan tanggal pada *field end event* sebelum tanggal *start event.* 6. Aktormemasukkan jumlah tiket kurang dari 10.000 7. Aktormenekan tombol edit *event.* | 1. Aktormendapat pesan pada *field end event* dan *field start sesi* tanggal harus sesudah *start event*. 2. Aktormendapatkan pesan tiket harus minimal 10.000. | *Passed* |
| Melihat data *Event* | 1. Aktor memilih menu *event* di halaman utama 2. Sistem menampilkan daftar *event* | 1. Aktor dapat melihat mengelola event seperti mengubah data *event* | *Passed* |
| Melihat data Sesi | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* Daftar Sesi 6. Sistem menampilkan halaman daftar sesi | 1. Aktordapat melihat data sesi. 2. Aktor dapat melihat, mengubah, dan menambah data sesi | *Passed* |
| Sukses Menambah data Sesi | 1. Aktor menekan tombol “+” 2. Sistem menampilkan halaman tambah sesi 3. Aktor memasukkan dan mengirim data sesi 4. Sistem menyimpan data sesi | 1. Aktormendapat pesan berhasil. 2. Data sesi berhasil ditambahkan. | *Passed* |
| Gagal Menambah data Sesi | 1. Aktor menekan tombol “+” 2. Sistem menampilkan halaman tambah sesi 3. Sistem menyimpan data sesi 4. Aktor mengisi tanggal sesi sebelum tanggal *start* *event* dan sesudah tanggal *end event.* 5. Aktor menekan tombol tambah sesi. | 1. Aktormendapatkan pesan tanggal sesi harus sesudah tanggal *start* event. | *Passed* |
| Sukses Mengubah Sesi | 1. Aktor memilih data sesi 2. Sistem menampilkan halaman edit sesi 3. Aktor memasukkan dan mengirim data sesi 4. Sistem menyimpan data sesi | 1. Aktormendapat pesan berhasil. 2. Data sesi berhasil diubah. | *Passed* |
| Gagal Mengubah Sesi | 1. Aktor memilih data sesi 2. Sistem menampilkan halaman edit sesi 3. Aktormengisi tanggal sesi sebelum tanggal *start* *event* dan sesudah tanggal *end event..* 4. Aktormenekan tombol edit sesi | 1. Aktormendapatkan pesan tanggal sesi harus sesudah tanggal *start* event. | *Passed* |
| Melihat data Agenda | 1. Aktor memilih menu kelola *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event.* 5. Aktor memilih *tab* daftar agenda 6. Sistem menampilkan halaman daftar agenda | 1. Aktordapat melihat data agenda. 2. Aktor dapat mengelola agenda seperti menambah dan mengubah data agenda | *Passed* |
| Sukses Menambah data Agenda | 1. Aktor menambahkan agenda baru dengan menekan tombol “+”. 2. Sistem menampilkan halaman *form* tambah agenda 3. Aktor memasukkan dan mengirim data agenda 4. Sistem menyimpan data agenda baru dan menampilkan alert berhasil | 1. Aktormendapat pesan berhasil. 2. Data agenda berhasil disimpan. | *Passed* |
| Gagal Menambah data Agenda | 1. Aktor menambahkan agenda baru dengan menekan tombol “+”. 2. Sistem menampilkan halaman *form* tambah agenda 3. Aktormengisi *field* tanggal *start* agenda sebelum *start event* dan sesudah tanggal *end event.*di halaman tambah agenda. 4. Aktormengisi *field* tanggal *end* agenda sebelum *start agenda* dan sesudah tanggal *end event* di halaman tambah agenda 5. Aktormenekan tombol tambah agenda | 1. Aktormendapat pesan tanggal *start* agenda harus setelah *start* event atau sama. 2. Aktormendapat pesan tanggal *end* agenda harus setelah tanggal *start* agenda | *Passed* |
| Sukses  Mengubah data Agenda | 1. Aktor mengubah data agenda dengan memilih agenda dari daftar agenda 2. Sistem menampilkan halaman *form* edit agenda 3. Aktor memasukkan dan mengirim data agenda 4. Sistem menyimpan perubahann data agenda dan menampilkan alert berhasil. | 1. Aktormendapat pesan berhasil. 2. Data agenda berhasil diubah. | *Passed* |
| Gagal  Mengubah data Agenda | 1. Aktor mengubah data agenda dengan memilih agenda dari daftar agenda 2. Sistem menampilkan halaman *form* edit agenda 3. Aktormengubah *field* tanggal *start* agenda sebelum *start event* dan sesudah tanggal *end event* di halaman edit agenda. 4. Aktormengubah *field* tanggal *end* agenda sebelum *start agenda* dan sesudah tanggal *end event* di halaman edit agenda 5. Aktormenekan tombol edit agenda | 1. Aktormendapat pesan tanggal *start* agenda harus setelah *start* *event*, sebelum *end event* atau sama. 2. Aktormendapat pesan tanggal *end* agenda harus setelah tanggal *start* agenda atau sebelum *end event* | *Passed* |
| Sukses  Mengunggah materi *event* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab Upload* Materi 6. Sistem menampilkan daftar agenda *event* 7. Aktor memilih daftar agenda *event* 8. Sistem menampilkan daftar materi agenda *event* 9. Aktor mengunggah materi agenda *event* 10. Sistem menyimpan materi agenda *event* | 1. Aktormendapat pesan berhasil 2. Materi berhasil di *upload.* 3. Aktor dapat melihat materi. | *Passed* |
| Gagal  Mengunggah materi *event* | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab Upload* Materi 6. Sistem menampilkan daftar agenda *event* 7. Aktor memilih daftar agenda *event* 8. Sistem menampilkan daftar materi agenda *event* 9. Aktor mengunggah materi agenda *event* 10. Aktormencari dan memilih berkas materi yang berformat .*pdf* dengan ukuran lebih dari 2 *mb* | 1. Aktormendapat pesan ukuran berkas melebihi 2 *MB* 2. Aktorharus mencari berkas lain. | *Passed* |
| Proses Berhasil  Menambah akun pemateri | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* TambahPemateri 6. Sistem menampilkan halaman tambah pemateri 7. Aktor memasukkan dan mengirim data pemateri 8. Sistem menyimpan data pemateri | 1. Aktormendapatkan pesan berhasil. 2. Akun pemateri berhasil ditambahkan. | *Passed* |
| Proses Gagal  Menambah akun pemateri | 1. Aktor memilih menu *event* 2. Sistem menampilkan daftar *event* 3. Aktor memilih daftar *event* dengan menekan tombol kelola *event* 4. Sistem menampilkan halaman kelola *event* 5. Aktor memilih *tab* TambahPemateri 6. Sistem menampilkan halaman tambah pemateri 7. Aktormengisi *field email* dengan *email* yang sudah terdaftar, *field password* kurang dari 8 karakter, *field* *password* tidak sama dengan *field verify password.* 8. Aktormenekan tombol tambah pemateri. | 1. Aktormendapat pesan email sudah terdaftar. 2. Aktormendapatkan pesan *password* minimal 8 karaker. 3. Aktormendapatkan pesan isi *field* *password* tidak sama dengan isi *field verify password.* | *Passed* |
| Sukses  Mengabsensi Peserta | 1. Aktor memilih menu peserta 2. Sistem menampilkan halaman kelola peserta. 3. Aktor memilih *tab* presensi peserta 4. Sistem menampilkan jadwal sesi dan agenda *event.* 5. Aktor menekan tombol *scan QR.* 6. Sistem menampilkan halaman presensi peserta. 7. Aktor memilih *tab Scan QR Code* 8. Sistem menampilkan halaman *scan QR Code.* 9. Aktor memindai *QR Code* peserta untuk absensi. 10. Jika peserta menggunakan kartu aktor memilih *tab* *Set Qt Code.* 11. Sistem menampilkan halaman *set QR Code*. 12. Aktor memindai *QR Code* kartu peserta dan menekan tombol *set QR.* 13. Sistem menyimpan data absensi peserta 14. Sistem merekap data absensi peserta. | 1. Aktormendapatkan pesan peserta berhasil absen. 2. Aktormendapatkan pesan berhasil *set QR Code* peserta*.* 3. Peserta sudah terabsen. | *Passed* |
| Gagal  Mengabsensi Peserta | 1. Aktor memilih menu peserta 2. Sistem menampilkan halaman kelola peserta. 3. Aktor memilih *tab* presensi peserta 4. Sistem menampilkan jadwal sesi dan agenda *event.* 5. Aktor menekan tombol *scan QR* yang tidak aktif. 6. Aktor menekan tombol *scan QR* ketika masih aktif*.* 7. Sistem menampilkan halaman presensi peserta. 8. Aktor memilih *tab Scan QR Code* 9. Sistem menampilkan halaman *scan QR Code.* 10. Aktormemindai *QR Code* peserta lebih dari 1 kali secara langsung. 11. Jika peserta menggunakan kartu peserta, u*ser* memilih *tab set QR Code.* 12. Aktormemindai *QR Code* kartu peserta dan memasukkan *email* yang belum terdaftarpeserta. 13. Aktormenetapkan *QR Code* kartu peserta untuk absensi peserta sebanyak lebih 1 kali. 14. Aktormenekan tombol *set QR Code.* | 1. Aktormendapatkan pesan peserta sudah melakukan absen. 2. Aktormendapatkan pesan *QR Code* salah. 3. *QR Code* sudah ditetapkan. 4. Aktor mendapatkan pesan *QR Code* sudah terdaftar jika *QR Code* sudah ditetapkan. 5. Aktor mendapat pesan *email* tidak terdaftar | *Passed* |
| Melihat Daftar Peserta | 1. Aktor memilih menu peserta 2. Sistem menampilkan halaman kelola peserta. 3. Aktor memilih *tab* rekap peserta. 4. Sistem menampilkan daftar sesi. 5. Aktor memilih salah satu sesi. 6. Sistem menampilkan halaman rekapitulasi peserta berisi rekapitulasi peserta dan daftar peserta seminar. 7. Aktor mengamati rekapitulasi peserta dan daftar peserta. | 1. Aktordapat mengamati data hasil rekapitulasi peserta berdasarkan sesi yang dipilih. | *Passed* |
| Sukses  Mengubah Profil | 1. Aktor memilih menu akun 2. Sistem menampilkan halaman akun 3. Aktor menekan tombol edit profil 4. Sistem menampilkan halaman edit profil 5. Aktor mengisi dan mengirim data profil yang akan diubah. 6. Sistem menyimpan data profil yang diubah | 1. Aktormendapatkan pesan berhasil. 2. Data profil *user* berhasil diubah. | *Passed* |
| Gagal  Mengubah Profil | 1. Aktor memilih menu akun 2. Sistem menampilkan halaman akun 3. Aktor menekan tombol edit profil 4. Sistem menampilkan halaman edit profil 5. Aktormengubah *password* dengan memasukkan *password* lama yang salah. 6. Aktor memasukkan *password* kurang dari 8 karakter. 7. Aktormemasukkan *password* baru dengan *verify* *password* tidak sama 8. Aktormenekan tombol edit profil. | 1. Aktormendapat pesan *password* lama tidak sama. 2. Aktormendapat pesan *password* minimal 8 karakter. 3. Aktormendapatkan pesan *password* tidak sama | *Passed* |
| Berhasil *Logout* | 1. Aktor memilih menu akun 2. Sistem menampilkan halaman akun yang berisikan informasi akun dan tombol *logout.* 3. Aktor menekan tombol *logout* 4. Sistem mengeluarkan aktor dari aplikasi dan menampilkan halaman *login* | 1. Aktorberhasil *logout* 2. Aktormenuju halaman *login*. | *Passed* |
| Sukses Register | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman login 3. Aktor menekan tombol register 4. Sistem menampilkan halaman register panitia 5. Aktor mengisi dan mengirim data akun panitia. 6. Sistem menyimpan data akun panitia | 1. Aktormendapat pesan berhasil. 2. Akun panitia berhasil dibuat. 3. Aktor dapat melakukan login di aplikasi sebagai ketua panitia | *Passed* |
| Gagal Register | 1. Aktor membuka aplikasi 2. Sistem menampilkan halaman login 3. Aktor menekan tombol register 4. Sistem menampilkan halaman register panitia 5. Aktormengisi *field email* dengan *email* yang sudah terdaftar, *field password* kurang dari 8 karakter, *field* *password* tidak sama dengan *field verify password.* 6. Aktor menekan tombol register | 1. Aktormendapat pesan gagal. 2. Aktormendapatkan pesan *password* minimal 8 karaker. 3. Aktormendapatkan pesan isi *field* *password* tidak sama dengan isi *field verify password.* |  |

## Pembahasan

Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android berfungsi mengelola kegiatan penyelenggaraan seminar panitia. Dalam proses pengelolaan kegiatan seminar, aplikasi ini mengelola kegiatan seminar seperti mengelola kegiatan seminar, pendaftaran peserta, presensi peserta, dan pembayaran penyelenggaraan kegiatan seminar untuk panitia. Proses mengelola kegiatan seminar panitia harus membuat sebuah *event* baru yang sudah berisi informasi lengkap dan membayar paket *event* baru menggunakan pembayaran *online* dari *payment gateway Xendit*. Kemudian mengelola peserta yang melakukan pendaftaran dengan membayar harga tiket kegiatan seminar melalui pembayaran *online* dari *payment gateway Xendit*, hasil pendaftaran akan tercatat pada aplikasi. Peserta yang sudah terdaftar dapat melakukan presensi kepada panitia menggunakan aplikasi peserta atau kartu peserta. Presensi peserta yang dilakukan panitia dengan memindai *QR Code* yang dimiliki peserta sesuai jadwal kegiatan seminar, hasil presensi akan tercatat pada aplikasi. Ketika mengelola kegiatan seminar panitia sebagai ketua panitia dapat menambahkan anggota panitia untuk membantu ketua panitia mengelola kegiatan seminar. Ketua dan anggota panitia dapat menambahkan materi kegiatan seminar dan menambahkan akun pemateri yang digunakan untuk aplikasi peserta.

*Payment gateway Xendit* membuat sebuah *invoice* sesuai paket yang dipilih, *invoice* dibayarkan berdasarkan bank yang tersedia di *payment gateway Xendit.* Setelah dibayarkan *payment gateway Xendit* mengirimkan *callback* kepada web *API* untuk mengubah dari status pembayaran dari *pending* menjadi *completed* atau *expired*, lalu *event* baru siap di kelola.

Berikut *source code* pembuatan *invoice* pada *payment gateway Xendit* yang digunakan pada Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android sebagaimana terlampir pada Lampiran 1 :

|  |
| --- |
| Xendit::setApiKey(getenv('SECRET\_API\_KEY'));  $params = [  'external\_id' => $name\_packet,  'payer\_email' => $email,  'description' => 'Pembuatan Invoice untuk pembelian '.$name\_packet.' '.$name,  'amount' => $price\_packet,  'should\_send\_email' => true  ];  $createInvoice = \Xendit\Invoice::create($params); |

Berikut *source code* pembuatan notifikasi *email* yang digunakan pada Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android sebagaimana terlampir pada Lampiran 2 :

|  |
| --- |
| $link = getenv('APP\_URL')."/aktivasi/$token";  $data = [  'name' => $name,  'body' => "Selamat akun anda berhasil dibuat silahkan melakukan aktivasi melalui link berikut ".$link  ];  Mail::send('email.mail', $data, function($message) use ($name, $email){  $message->to($email,$name)->subject('Pemberitahuan Akun');  $message->from('admin@apidigimice.me','digiMICE Panitia');  }); |

Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android diuji menggunakan pengujian *blackbox testing*, dimana pengujian ini menguji fitur-fitur aplikasi sesuai rumusan masalah yang dibuat untuk memenuhi tujuan yang dibuat agar tujuan yang dibuat dapat tercapai seperti aplikasi dapat mengelola kegiatan seminar dan menggunakan pembayaran *online* melalui *payment gatewat Xendit*.

# BAB 5 PENUTUP

## Kesimpulan

Berdasarkan proses perancangan Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android dapat disimpulkan sebagai berikut :

* 1. Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android yang merupakan pengembangan dari aplikasi berbasis Web *digiMICE* ini dapat membantu panitia penyelenggara seminar untuk mengelola kegiatan seminar yang akan diselenggarakan seperti mengelola kegiatan baru dengan pembayaran paket, pendaftaran peserta, dan presensi peserta.
  2. Aplikasi ini dapat digunakan oleh lebih dari satu panitia yang berfungsi untuk mempermudah panitia membagi tugas dalam mengelola kegiatan seminar seperti ketua panitia berbagi tugas dengan anggotanya.
  3. Aplikasi ini dapat melakukan pembayaran *online* untuk membeli paket kegiatan seminar dengan menggunakan *payment gateway Xendit.*
  4. Aplikasi ini mengelola data kegiatan seminar yang telah ditambahkan oleh panitia untuk digunakan pada aplikasi peserta seminar.

## Saran

Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android dapat dikatakan masih jauh dari kata sempurna. Pada aplikasi ini masih terdapat beberapa kekurangan dan masih perlu dikembangkan lagi agar sistem lebih kompleks. Berikut beberapa saran untuk mengembangkan Aplikasi Manajemen Seminar (*digiMICE*) untuk Panitia berbasis Android ini.

* + 1. Aplikasi ini masih perlu dikembangkan di sisi tampilan, dikarenakan tampilan masih sederhana.
    2. Aplikasi ini dapat menggunakan dikembangkan di sisi fitur yang masih dapat ditambahkan ketika mengelola kegiatan seminar yang diselenggarakan oleh panitia, seperti penambahan lokasi otomatis dan penambahan dokumentasi kegiatan
    3. Aplikasi ini diharapkan dapat dikembangkan untuk mengelola kegiatan *MICE* mengikuti aplikasi berbasis Web *digiMICE* agar cakupan aplikasi ini menjadi lebih kompleks dengan menyesuaikan permasalahan yang ada.
    4. Aplikasi dapat dikembangkan untuk berjalan di berbagai *platform* *mobile* lain selain Android.

# DAFTAR PUSTAKA

Abdulloh, R., 2016. *Web Programming is Easy*. Jakarta: PT Elex Media Komputindo.

Adumy, D.A., Periyadi & Ismail, S.J.I., 2019. Sistem Absensi Seminar Menggunakan QR Code. *e-Proceeding of Applied Science*, 5, p.1350.

Agarina, M. & Jamil, S.N., 2018. IMPLEMENTASI JAVA REMOTE METHOD INVOCATION (JRMI). *Jurnal Sistem Informasi dan Telematika*, 9, pp.39-44.

Darsana, I.N., 2016. *PANDUAN SEMINAR*. [Online] BALI: UNIVERSITAS UDAYANA Available at: <https://simdos.unud.ac.id/uploads/file_pendidikan_1_dir/a65da35a47e77c6759299b3f80fc0a70.pdf> [Accessed 22 January 2020].

Fatmawatie, B.D., Adittya, B.R. & Sardi, I.L., 2016. Aplikasi Informasi Kegiatan Seminar Nasional untuk Dosen Universitas Telkom Berbasis Android dan Web. *e-Proceeding of Applied Science*, 2, p.308.

Firly, N., 2019. *Create Your Own Android Application*. Jakarta: PT Elek Media Komputindo.

Jaya, T.S., 2018. Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(02), pp.45-46.

Jubilee Enterprise, 2015. *Mengenal Java dan Database dengan Netbeans*. Jakarta: PT Elex Media Komputindo.

Kemendikbud, 2016. *KBBI Daring*. [Online] Available at: <https://kbbi.kemdikbud.go.id/entri/seminar> [Accessed 22 January 2020].

Lavarino, D. & Yustanti, W., 2016. RANCANG BANGUN E – VOTING BERBASIS WEBSITE DI UNIVERSITAS NEGERI SURABAYA. *Jurnal Manajemen Informatika*, 6, pp.72-81.

MADCOMS, Divisi LITBANG, 2016. *Pemrograman PHP dan MySQL untuk Pemula.* Yogyakarta: C.V ANDI OFFSET.

Maulidy, F.R. & Nudin, S.R., 2019. RANCANG BANGUN SIMDEPAD (SISTEM INFORMASI MONITORING DAN EVALUASI. *Jurnal Manajemen Informatika*, 9, pp. 162-168.

Mustaqbal, M.S., Firdaus, R.F. & Rahmadi, H., 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), p.34.

Noor, A., 2015. *Globalisasi Industri MICE*. Bandung: Alfabeta.

Nugroho, A., 2016. Aplikasi Web Informasi Dan Registrasi Peserta Semina, Workshop, Talkshow pada Acara Seminar Nasional Pengamplikasian Telematika (SINAPTIKA) Tahun 2016. *Seminar Nasional Sistem Informasi Indonesia*, 2017.

*--- Halaman ini sengaja dikosongkan ---*

Permana, K.S., 2019. *Ketika Pengguna Internet dan Smartphone Terus Meningkat, Android Dominasi Pasar Indonesia dan Dunia*. [Online] Available at: <https://jabar.tribunnews.com/2019/01/24/ketika-pengguna-internet-dan-smartphone-terus-meningkat-android-dominasi-pasar-indonesia-dan-dunia> [Accessed 22 January 2020].

Putera, A.D., 2018. *Pengembangan Pariwisata Indonesia Lebih Cepat Jika Berbasis MICE*. [Online] Available at: <https://ekonomi.kompas.com/read/2018/09/18/101000026/pengembangan-pariwisata-indonesia-lebih-cepat-jika-berbasis-mice> [Accessed 5 Januari 2020].

S., R.A. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.

Suryantara, I.G., 2017. *Merancang Aplikasi dengan Metodologi Extream Programmings*. Jakarta: PT Elex Media Komputindo.

Xendit, n.d. *Apa itu Xendit*. [Online] Available at: <https://xendit.github.io/xendit-kb/Indonesia/about-us/what-is-xendit/> [Accessed 27 January 2020].

Yudhanto, Y. & Wijayanto, A., 2019. *Yuk Berbisnis dengan Laravel dan Android*. Jakarta: PT Elex Media Komputindo.

*--- Halaman ini sengaja dikosongkan ---*

# LAMPIRAN

## Lampiran 1. *Source Code* Pembuatan *Invoice* pada *payment gateway Xendit.*

|  |
| --- |
| public function create(Request $request){  $name = $request->input('name');  $start = $request->input('start');  $end = $request->input('end');  $event\_type\_id = $request->input('event\_type\_id');  $banner = $request->input('banner');  $description = $request->input('description');  $place = $request->input('place');  $address = $request->input('address');  $event\_status = $request->input('event\_status');  $event\_paket\_id = $request->input('event\_paket\_id');  $presence\_type = $request->input('presence\_type');  $event\_ticket\_price = $request->input('event\_ticket\_price');  $name\_session = $request->input('name\_session');  $name\_packet = $request->input('name\_packet');  $price\_packet = $request->input('price\_packet');    $user\_id = $request->input('user\_id');  $team\_role = $request->input('team\_role');  $name\_team = $request->input('name\_team');  $email = $request->input('email');  $start\_session = $request->input('start\_session');  if(trim($banner) == ''){  $file = "upload/images/blank.jpg";  $data = [  'name' => $name,  'start' => $start,  'end' => $end,  'event\_type\_id' => $event\_type\_id,  'banner' => $file,  'description' => $description,  'place' => $place,  'address' => $address,  'event\_status' => $event\_status,  'event\_paket\_id' => $event\_paket\_id,  'presence\_type' => $presence\_type,  'event\_ticket\_price' => $event\_ticket\_price,  ];  }else{  $target\_dir = "upload/images";    if(!file\_exists($target\_dir)){  mkdir($target\_dir, 0777, true);  }    $file = $target\_dir."/image".time().".jpeg";  $ifp = fopen($file, "wb");  $data2 = explode(',', $banner);  fwrite($ifp, base64\_decode($data2[0]));  fclose($ifp);  $data = [  'name' => $name,  'start' => $start,  'end' => $end,  'event\_type\_id' => $event\_type\_id,  'banner' => $file,  'description' => $description,  'place' => $place,  'address' => $address,  'event\_status' => $event\_status,  'event\_paket\_id' => $event\_paket\_id,  'presence\_type' => $presence\_type,  'event\_ticket\_price' => $event\_ticket\_price,  ];  }  $event = Event::create($data);  $team = Team::where('event\_id',$event->id)->first();  $edit = [  'user\_id' => $user\_id,  'team\_role' => $team\_role,  'name\_team' => $name\_team  ];  $team->update($edit);    Session::create([  'name' => $name\_session,  'start' => $start\_session,  'event\_event\_type\_id' => $event\_type\_id,  'event\_id' => $event->id  ]);  if(!$price\_packet == "0"){  Xendit::setApiKey(getenv('SECRET\_API\_KEY'));  $params = [  'external\_id' => $name\_packet,  'payer\_email' => $email,  'description' => 'Pembuatan Invoice untuk pembelian '.$name\_packet.' '.$name,  'amount' => $price\_packet,  'should\_send\_email' => true  ];  $createInvoice = \Xendit\Invoice::create($params);  $data3 = [  'status' => "PENDING",  'event\_id' => $event->id,  'event\_paket\_event\_id' => $event\_paket\_id,  'id\_invoice' => $createInvoice['id'],  'user\_id' => $user\_id  ];  EventOrderHistory::create($data3);  }else{  $data3 = [  'status' => "PAID",  'event\_id' => $event->id,  'event\_paket\_event\_id' => $event\_paket\_id,  'id\_invoice' => "no invoice",  'user\_id' => $user\_id  ];  EventOrderHistory::create($data3);  $user = User::where('id', $user\_id)->first();  $namePanitia = $user->name;  $dataEmail = [  'name' => $namePanitia,  'body' => "Selamat Event Paket Gratis $name Anda Berhasil ditambahkan"  ];  Mail::send('email.sukses', $dataEmail, function($message) use ($namePanitia, $email){  $message->to($email, $namePanitia)->subject('Pemberitahuan Event');  $message->from('admin@apidigimice.me', 'digiMICE Panitia');  });    }    return "Berhasil ditambahkan";  } |

*--- Halaman ini sengaja dikosongkan ---*

## Lampiran 2. *Source Code* PembuatanNotifikasi *Email.*

|  |
| --- |
| public function registerPanitia(Request $request){  $this->validate($request,[  'email' => 'required|unique:user',  'password' => 'required|min:8'  ]);  $email = $request->input('email');  $password = $request->input('password');  $name = $request->input('name');  $phone = $request->input('phone');  $regencies\_id = $request->input('regencies\_id');  $hasPassword = Hash::make($password);  $token = $this->getRandomString();  $data = [  'email' => $email,  'password\_hash' => $hasPassword,  'name' => $name,  'role' => "lead eo",  'phone' => $phone,  'regencies\_id' => $regencies\_id,  'avatar' => "upload/images/blank.jpg",  'verification\_token' => $token  ];  if(User::create($data)){  $link = getenv('APP\_URL')."/aktivasi/$token";  $data = [  'name' => $name,  'body' => "Selamat akun anda berhasil dibuat silahkan melakukan aktivasi melalui link berikut ".$link  ];  Mail::send('email.mail', $data, function($message) use ($name, $email){  $message->to($email, $name)->subject('Pemberitahuan Akun');  $message->from('admin@apidigimice.me', 'digiMICE Panitia');  });  $pesan = [  'message' => 'register success',  'code' => 201  ];  }else{  $pesan = [  'message' => 'register gagal',  'code' => 404  ];  }  return response()->json($pesan,$pesan['code']);  } |

## Lampiran 3. Biodata Penulis

**BIODATA PENULIS**

**Identitas Diri :**

Nama Lengkap : Angga Dwi Saputra   
Tempat/Tgl. Lahir : Banyuwangi, 16 Desember 1998  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Status : Mahasiswa  
Kewarganegaraan : Indonesia  
Alamat : Dusun Tegalsari Lor Rt 13 Rw 03 Desa Purwoasri

Kecamatan Tegaldlimo

No. Telp/Hp : 082245836528

Email : anggadwisaputra83@gmail.com  
**Riwayat Pendidikan** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SD | SMP | SMK | PERGURUAN TINGGI |
| Nama Institusi | SD Negeri 2 Purwoasri | SMP Negeri 1 Tegaldlimo | SMKN 1 Banyuwangi | Politeknik Negeri Banyuwangi |
| Jurusan | - | - | Multimedia | D3- Teknik Informatika |
| Tahun Masuk- Lulus | 2005 – 2011 | 2011 – 2014 | 2014 – 2017 | 2017 – 2020 |

Biodata penulis ini dibuat sesuai dengan keadaan yang sesungguhnya.

Banyuwangi, 12 Agustus 2020

Penyusun,

Angga Dwi Saputra

NIM. 361755401181

*--- Halaman ini sengaja dikosongkan ---*